

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Electricidad y Computación**

Diseño de sistema supervisor remoto y diagnóstico predictivo basado en Técnicas de Inteligencia Artificial, para la operación de grúas en un Terminal Portuario

**Proyecto Integrador**

Previo la obtención del Título de:

**Ingeniero en Electrónica y Automatización**

Presentado por:

Emily Vanessa Guaita Mayancela

Ricardo Antonio Muñoz Ochoa

Guayaquil - Ecuador

Año: 2024

## **Dedicatoria**

---

Dedico este trabajo a mi familia, mis padres y hermanas, pilares fundamentales en mi vida, quienes han confiado en mí y me han acompañado durante toda mi formación académica. A mis docentes, amigos y a todos aquellos que estuvieron junto a mí en el proceso. La culminación de esta etapa no hubiera sido posible sin su apoyo incondicional y consejos.

**Emily Guaita Mayancela**

Dedico este trabajo a mi familia, aquella que no es estrictamente de sangre, a mi madre, a mi tía, a quien considero mi hermano, a quien considero mi hijo, a mi figura paterna y por quienes ya no me acompañan, pero siempre me quisieron ver triunfar. A los docentes y compañeros que estuvieron apoyándome durante el proceso.

**Ricardo Muñoz Ochoa**

## Agradecimientos

---

Agradezco a Dios y a la Virgen María por ser mi fuente de fortaleza y alegría, gracias por cuidarme y bendecirme en todas las circunstancias de la vida. A mis padres Gustavo y Ofelia, por su ardua labor y sacrificio, su amor y guía han sido mi refugio en los momentos más difíciles. A mis hermanas Gaby y María José, mi abuelita Rosario, tía Flor, Verónica y Sebas, por su presencia, cariño y predisposición siempre. A Nico, Mickey y Mist, parte integral de mi familia. A mis docentes, por sus conocimientos impartidos. En general, a todas las personas que me han acompañado en este viaje ¡Muchas gracias por todo!

**Emily Guaita Mayancela**

## Agradecimientos

---

Agradezco a Dios y a la Virgen María por ser mi fuente de fortaleza y alegría, gracias por cuidarme y bendecirme en todas las circunstancias de la vida. A mi madre Jenniffer por estar siempre a mi lado, enseñándome como debe ser una persona profesional y que actúa con valores siempre. A mi tía María por siempre alentarme a alcanzar mis metas y no detenerme por las dificultades. A mi bisabuela Dolores, mi bisabuelo Jorge y mi tío Jorge que desde cielo me atestiguan, por en vida siempre demostrar la fe que tenían en que llegaría lejos. Al Ing. Alex Cornejo por apoyarme en mi formación técnica.

**Ricardo Muñoz Ochoa**

## Declaración Expresa

---

Nosotros Emily Vanessa Guaita Mayancela y Ricardo Antonio Muñoz Ochoa acuerdo/acordamos y reconozco/reconocemos que la titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, información no divulgada y cualquier otro derecho o tipo de Propiedad Intelectual que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, incluyendo cualquier derecho de participación de beneficios o de valor sobre titularidad de derechos, pertenecerán de forma total, perpetua, exclusiva e indivisible a LA ESPOL, sin limitación de ningún tipo. Se deja además expresa constancia de que lo aquí establecido constituye un “previo acuerdo”, así como de ser posible bajo la normativa vigente de transferencia o cesión a favor de la ESPOL de todo derecho o porcentaje de titularidad que pueda existir.

Sin perjuicio de lo anterior los alumnos firmantes de la presente declaración reciben en este acto una licencia de uso gratuita e intransferible de plazo indefinido para el uso no comercial de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, sin perjuicio de lo cual deberán contar con una autorización previa expresa de la ESPOL para difundir públicamente el contenido de la investigación, desarrollo tecnológico o invención.

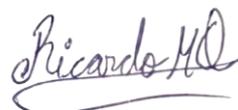
Así también autorizamos expresamente a que la ESPOL realice la comunicación pública de la obra o invento, por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual.

Guayaquil, 20 de Enero del 2024.



Emily Vanessa

Guaita Mayancela



Ricardo Antonio

Muñoz Ochoa

## **Evaluadores**



---

**Dennys Dick Cortez Alvarez**

Profesor de Materia

---

**Douglas Antonio Plaza Guingla**

Tutor de proyecto

## Resumen

En la presente tesis se diseña un sistema predictivo para anticipar las fallas en una grúa STS en el Terminal Portuario de Guayaquil, a través del análisis vibratorio de motores eléctricos incorporados en la grúa usando un modelo de Inteligencia Artificial, y mostrar los resultados de ese modelo en un sistema SCADA. En los aspectos considerados para la determinación de falla de motores, está la revisión de la magnitud de frecuencias según las normas ISO y los armónicos en el espectro de frecuencia para diagnosticar fallo mecánico de los elementos móviles del sistema y estructura mecánica que soporta al mismo. Esto con el fin de que el área de mantenimiento este mejor preparada para preservar la integridad de la grúa y no tenga que detener sus operaciones por un tiempo prolongado, a causa de un desperfecto o avería crítica fuera del plan de mantenimiento actual. Además de generar un registro más eficiente de datos históricos para el seguimiento y pronóstico de fallos de grúas del terminal portuario. Para el diseño se montó una red con equipos Siemens como PLC, variador de frecuencia, módulos punto a punto y motores de inducción, el cual se conectó a un servidor que crea una instancia en Node RED para distribuir los datos de vibración, voltaje, corriente, peso y posición a una red neuronal de regresión para así llegar a la determinación de la probabilidad de una tendencia de falla. De los resultados obtenidos se tuvo que la red reconoce el surgimiento temprano de fallos estructurales e internos de los elementos rodantes progresivamente hasta antes del punto donde generan danos graves en los equipos, notificando la necesidad de mantenimiento preventivo con mayor eficiencia. Con el proyecto se consigue integrar a la grúa a la Industria 4.0, a monitorear continuamente su estado y a proporcionar información útil para preservar la integridad de los equipos esenciales para las tareas diarias.

**Palabras Clave:** redes neuronales, mantenimiento predictivo, histórico de datos, análisis vibratorio.

## **Abstract**

This thesis presents the design of a predictive system to anticipate failures in an STS crane at the Guayaquil Port Terminal. The system uses vibration analysis of electric motors incorporated in the crane through an Artificial Intelligence model and displays the results of that model in a SCADA system. The aspects considered for determining motor failure include the review of the magnitude of frequencies according to ISO standards and harmonics in the frequency spectrum to diagnose mechanical failure of the moving elements of the system and the mechanical structure that supports it. The maintenance department can be better prepared to preserve the integrity of the crane with this analysis and prevent stopping operations for a prolonged period due to a critical failure or breakdown outside of the current maintenance plan. It also aims to generate a more efficient historical data record for monitoring and predicting failures of cranes in the port terminal. For the design, a network was assembled with Siemens equipment such as PLC, VFD, point-to-point modules, and induction motors, which were connected to a server that creates an instance in Node-RED to distribute vibration, voltage, current, weight, and position data to a regression neural network to determine the probability of a failure trend. The results showed that the network recognizes the early onset of structural and internal failures of the rolling elements progressively until before the point where they generate serious damage to the equipment, notifying the need for preventive maintenance with greater efficiency. The project achieves the integration of the crane into Industry 4.0, to continuously monitor its condition and to provide useful information to preserve the integrity of the equipment essential for daily tasks.

**Keywords:** neural networks, predictive maintenance, data history, vibration analysis.

## Índice general

Evaluadores .....	6
Resumen .....	7
Abstract.....	8
Índice general .....	9
Abreviaturas .....	15
Simbología.....	16
Índice de figuras .....	17
Índice de tablas .....	22
Índice De Planos.....	24
Capítulo 1 .....	25
1.1.    Introducción .....	25
1.2.    Descripción del problema.....	25
1.3.    Justificación del problema.....	26
1.4.    Objetivos .....	27
1.4.1.    Objetivo general .....	27
1.4.2.    Objetivos específicos.....	27
1.5.    Marco teórico .....	28
1.5.1    Descripción de una Grúa STS .....	28
Tipos de Grúa Portacontenedores.....	29

Componentes .....	30
Mecanismo de elevación .....	33
Elementos mecánicos y eléctricos .....	34
1.5.2 Mantenimiento Industrial .....	36
Tipos de mantenimiento .....	36
Categorización de equipos en planta. ....	37
Técnicas de Mantenimiento Predictivo. ....	38
1.5.3 Sistema SCADA.....	39
Autómata Programable .....	39
Módulos de entradas y salidas digitales .....	43
Módulo de Entradas y salidas analógicas. ....	44
Variadores de frecuencia. ....	46
Protocolos de Comunicación. ....	52
1.5.4 Crane Monitoring System .....	55
1.5.5 Node RED .....	57
1.5.6 Sistema Motor-rodamiento de Hoist CMS.....	59
Sensor de Vibración: Acelerómetro de velocidad (Current Output). ....	60
Motor de Inducción utilizado para los dos motores Hoist.....	63
Rodamientos .....	66
1.5.7 Inteligencia Artificial .....	67

1.5.6.	Análisis Vibratorio .....	69
	Sistema Resorte-Masa: Masa, Rigidez y Amortiguación.....	71
	Recolección de Señales de Vibración con Transductores de Vibración. ....	72
	Conversión de Vibraciones en Señal Eléctrica.....	74
	Procesamiento de las Señales en Bases de Datos para Motores con Vibraciones.	74
	Analizador de Vibraciones. ....	74
	Transformada de Fourier FFT para Análisis de Vibración .....	76
	Los Armónicos. Límites y Normas de la Vibración.....	76
	Procedimiento de medición. ....	79
	Puntos de medición.....	80
	Condiciones de operación.....	80
	Identificación de causas de vibraciones .....	80
1.5.7.	Teoremas y ecuaciones para Análisis vibratorio.....	84
	Teorema de Nyquist .....	84
	Frecuencias de Rodamientos .....	84
1.6.	Trabajos anteriores .....	86
	1.6.1. “Determinación del estado de falla de motores eléctricos a partir del análisis de ruidos utilizando técnicas de Aprendizaje Profundo” Dávalos y Vásquez (2022).....	86
	1.6.2. “Análisis y detección de fallas en motores eléctricos aplicando algoritmos de inteligencia artificial” Odar Chero (2023).....	87

1.6.3. “Mantenimiento Predictivo en transformadores de potencia y de distribución eléctrica” Jado Moreno (2022).....	88
1.6.4. “Sistema de mantenimiento predictivo para máquinas eléctricas en procesos industriales en el escenario de la industria 4.0” Rodríguez Rivera (2022). .....	89
Capítulo 2 .....	92
2.1. Metodología. ....	92
2.2. Selección de Elementos.....	95
2.2.1. Modbus TCP en lugar de OPC Server: Simplicidad y Compatibilidad ....	95
2.2.2. Nodered en lugar de Matlab: Sistema de Análisis Compacto .....	97
2.2.3. Google Cloud como una nube predilecta para almacenamiento y análisis de datos	98
2.2.4. Consideraciones técnicas del proyecto.....	100
2.3. Flujo del Proyecto .....	104
2.4. Flujos de node red .....	106
2.4.1. Entrenamiento de Modelo (MODEL) .....	106
2.4.2. Lectura y Almacenamiento (DATA y MONITOR).....	107
2.4.3. Histórico del CMS (DATA y MONITOR) .....	108
2.4.4. Presentación de Vibraciones (MODEL) .....	110
2.4.5. Analizador de Vibraciones (ANALYSIS y MONITOR).....	110
2.4.6. Reportes de Google (Reporte y Monitoreo).....	112
2.5. Entrenamiento .....	113

2.6.	Diagnóstico de vibraciones .....	115
Capítulo 3 .....		119
3.1.	Resultados y análisis .....	119
3.1.1.	Integración de Raspberry a la red local para Recolección de Datos .....	119
3.1.2.	Nueva base de datos .....	123
3.1.3.	Entrenamiento modelo Aprendizaje de Maquina.....	124
3.1.4.	Reconocimiento de fallas .....	128
3.1.5.	Reportes con Google .....	137
3.2.	Costos .....	140
Capítulo 4 .....		141
4.1.	Conclusiones y recomendaciones.....	141
4.1.1.	Conclusiones .....	142
4.1.2.	Recomendaciones.....	143
Referencias .....		144
Apéndice A.....		147
Cálculo de Frecuencias de vibraciones en Rodamientos .....		147
Cálculo de Frecuencias de muestreo para los rodamientos.....		149
Cálculo de Velocidad de transmisión de datos .....		149
Apéndice B .....		154
Código de entrenamiento RandomForest.....		154

Fragmento de generación de armónicos.....	156
Código de conexión y generación de queries con Google Cloud .....	157
Programación en TIA Portal .....	161
Apéndice C .....	165
Código de captura del panel CMS .....	165
Código para analizar datos en las capturas del CMS .....	165
Oportunidad de Banco Pruebas a futuro en la ESPOL .....	168
Acelerómetro PCB Piezoeléctrico 352C03 .....	169
Cable BNC Modelo 003C10 PCB .....	170

## Abreviaturas

ESPOL	Escuela Superior Politécnica del Litoral
STS	Ship to Shore (Barco a Tierra)
CMS	Crane Monitoring System (Sistema de Monitoreo de Grúa)
AI	Artificial Intelligence (Inteligencia Artificial)
TCP	Transmission Control Protocol (Protocolo de Control de Transmisión)
SQL	Structured Query Language (Lenguaje de Consultas Estructuradas)
SCADA	Supervisory Control and Data Acquisition (Supervision, Control y Acquisition de Datos)
PLC	Programmable Logic Controller (Controlador Lógico Programable)
VFD	Variable Frequency Drive (Controlador de Frecuencia Variable)

## Simbología

mm/s	Milímetro por segundo
Ton	Tonelada
V	Voltaje
m	Metro
mV	Milivoltio
A	Corriente
Hz	Frecuencia
%	Porcentaje

## Índice de figuras

Figura 1.5.1 <i>Esquema del desarrollo de las grúas STS</i> .....	30
Figura 1.5.2 <i>Caseta de maquinaria de la grúa STS 3 del Terminal Portuario</i> .....	32
Figura 1.5.3 <i>Spreader utilizado en el movimiento Hoist de la grúa</i> .....	33
Figura 1.5.4 <i>Componentes de una grúa STS</i> .....	33
Figura 1.5.5 <i>Imagen del Controlador AC 500</i> .....	41
Figura 1.5.6 <i>Controlador S7-1500</i> .....	43
Figura 1.5.7 <i>Módulos de entradas y salidas digitales para un S7-1500</i> .....	44
Figura 1.5.8 <i>Módulos de salidas digitales para un AC 500</i> .....	44
Figura 1.5.9 <i>Módulos de entradas analógicas para un S7-1500</i> .....	45
Figura 1.5.10 <i>Rampas de Aceleración y Desaceleración en un Variador de Frecuencia</i>	47
Figura 1.5.11 <i>Variador de Frecuencia ACS800</i> .....	48
Figura 1.5.12 <i>Variador de Frecuencia SINAMICS G120</i> .....	49
Figura 1.5.13 <i>Periferia Descentralizada CI502 para AC500</i> .....	51
Figura 1.5.14 <i>Periferia Descentralizada ET200 SP con módulo de interfaz IM 155-6 PN</i> .....	52
Figura 1.5.15 <i>Adición de Servidor Modbus en Automation Builder</i> .....	53
Figura 1.5.16 <i>Adición de Servidor Modbus en TIA Portal</i> .....	53
Figura 1.5.17 <i>Entorno de Node RED</i> .....	58
Figura 1.5.18 <i>Nodos Modbus</i> .....	58

Figura 1.5.19 <i>Panel de Node RED</i> .....	59
Figura 1.5.20 <i>Acelerómetro Modelo: 060 J079 01</i> .....	60
Figura 1.5.21 <i>Dimensiones y código de cableado del Modelo MAV52 Código AG932</i> ...	61
Figura 1.5.22 <i>Motor M3BP 450LB 6</i> .....	63
Figura 1.5.23 <i>Dimensiones de los rodamientos</i> .....	66
Figura 1.5.24 <i>Diagrama de RandomForest</i> .....	69
Figura 1.5.25 <i>Niveles de vibracion segun clase de Motor y su estado</i> .....	79
Figura 1.5.26 <i>Dimensiones de los rodamientos</i> .....	81
Figura 1.5.27 <i>Desalineaciones</i> .....	82
Figura 1.5.28 <i>Espectro de frecuencia de desalineación</i> .....	83
Figura 1.5.29 <i>Espectro de deterioro interno</i> .....	84
Figura 2.1.1 <i>Ilustración de red de Mantenimiento Predictivo</i> .....	93
Figura 2.2.1 <i>Servidor Modbus del PLC S7 1500</i> .....	96
Figura 2.2.2 <i>Instancia de Node-RED en Placa Raspberry</i> .....	98
Figura 2.2.3 <i>Consola de Google Cloud con instancia SQL</i> .....	99
Figura 2.2.4 <i>Instancia de Google abierta desde MySQL Workbench</i> .....	99
Figura 2.4.1 <i>Training Model</i> .....	106
Figura 2.4.2 <i>Flujo Recolección de Datos (CMS to LocalStorage)</i> .....	107
Figura 2.4.3 <i>CMS: Hoist Panel en Dashboard</i> .....	108
Figura 2.4.4 <i>CMS: Hoist Panel en Dashboard</i> .....	108

Figura 2.4.5 <i>CMS: Hoist Historical en Dashboard</i> .....	109
Figura 2.4.6 <i>Flujo Visualización de Predicciones vs Lecturas</i> .....	110
Figura 2.4.7 <i>Flujo Análisis de Datos Diario</i> .....	110
Figura 2.4.8 <i>Panel de niveles de vibración.</i> .....	111
Figura 2.4.9 <i>Panel de espectro de frecuencia.</i> .....	112
Figura 2.4.10 <i>Flujo para conexión con Google</i> .....	112
Figura 2.5.1 <i>Diagrama del sistema Motor-rodamientos-spreader</i> .....	113
Figura 2.5.2 <i>Lecturas exportadas a Schema de MySQL</i> .....	114
Figura 2.5.3 <i>Panel CMS</i> .....	114
Figura 2.6.1 <i>Proyección de la lectura de sensores de vibración de motor Hoist 1 y 2, y del sensor Slack Rope</i> .....	115
Figura 2.6.2 <i>Proyección de los datos de prueba de un motor asíncrono con carga mecánica en estado normal (curva azul) y presentando fallas mecánicas (curva naranja).</i> .....	117
Figura 2.6.3 <i>Espectro de base de datos estado Normal</i> .....	118
Figura 2.6.4 <i>Espectro de base de datos estado Desgastado</i> .....	118
Figura 3.1.1 <i>Servidor Raspberry integrado a red de la planta</i> .....	119
Figura 3.1.2 <i>Reconocimiento de equipo Raspberry por ordenadores de oficina</i> .....	119
Figura 3.1.3 <i>Abertura del panel</i> .....	120
Figura 3.1.4 <i>Cuadro de dialogo para Inicio de Sesión.</i> .....	121
Figura 3.1.5 <i>Bloques recibiendo valores de parámetros.</i> .....	121
Figura 3.1.6 <i>Registro de sostenimiento con valores de parámetros.</i> .....	122

Figura 3.1.7 <i>Gráfico de Posición del Hoist en el histórico.</i> .....	124
Figura 3.1.8 <i>Mejor Predicción de las primeras pruebas.</i> .....	125
Figura 3.1.9 <i>Análisis de diferencias de posición Hoist y Trolley en relación con la vibración.</i> .....	126
Figura 3.1.10 <i>Fragmento del entrenamiento con reacondicionamiento de valores de entrada.</i> .....	127
Figura 3.1.11 <i>Predicción vs Diferencia con precisión del 90%</i> .....	127
Figura 3.1.12 <i>Predicción vs Real Hoist1 con precisión del 90%</i> .....	127
Figura 3.1.13 <i>Predicción vs Real Hoist 2 con precisión del 90%</i> .....	128
Figura 3.1.14 <i>Proporciones de magnitudes de vibración de Motor1 recién reacondicionado</i> .....	128
Figura 3.1.15 <i>Frecuencias de diferencias generadas por Python</i> .....	129
Figura 3.1.16 <i>Comparación señal en estado normal (azul) y señal con amplitudes aumentadas (naranja)</i> .....	129
Figura 3.1.17 <i>Espectro de frecuencia con amplitud aumentada.</i> .....	130
Figura 3.1.18 <i>Proporciones de magnitudes de vibración de Motor1 con emulación de desgaste parcial de soldadura o afloje de los soportes</i> .....	130
Figura 3.1.19 <i>Espectro de Prueba 4</i> .....	131
Figura 3.1.20 <i>Comparación con señal real en Prueba 4</i> .....	131
Figura 3.1.21 <i>Espectro de frecuencia completo en Prueba 4</i> .....	132
Figura 3.1.22 <i>Comparación con señal real en Prueba 8</i> .....	132

Figura 3.1.23 <i>Espectro de frecuencia real en Prueba 8</i> .....	133
Figura 3.1.24 <i>Proporción de Magnitudes de Vibraciones en Rodamiento 1 en Prueba 8</i> .....	133
Figura 3.1.25 <i>Panel de Espectro de Frecuencia en Prueba 8</i> .....	134
Figura 3.1.26 <i>Espectro real de frecuencia en Prueba 7</i> .....	135
Figura 3.1.27 <i>Comparación con señal real en Prueba 7</i> .....	135
Figura 3.1.28 <i>Análisis de niveles de vibración en Prueba 7</i> .....	136
Figura 3.1.29 <i>Espectro de vibración de Motor1 con emulación de desgaste tras un tiempo prolongado.</i> .....	136
Figura 3.1.30 <i>Reportes almacenados en base de datos en la nube.</i> .....	137
Figura 3.1.31 <i>Reporte de decaimiento del estado del Rodamiento 1 del Motor 1 con pruebas iniciales</i> .....	138
Figura 3.1.32 <i>Reporte del sistema después de pruebas variadas</i> .....	138
Figura 3.1.33 <i>Avisos de presencia de deterioro y de estado de fallo en el Rodamiento 1 del Motor1.</i> .....	139

## Índice de tablas

Tabla 1.5-1 <i>Especificaciones técnicas del Controlador AC 500</i> .....	41
Tabla 1.5-2 <i>Especificación S7 1500</i> .....	42
Tabla 1.5-3 <i>Especificaciones técnicas del Variador de Frecuencia SINAMICS G120</i> ....	48
Tabla 1.5-4 <i>Especificaciones técnicas de la Periferia Descentralizada CI502 para AC500</i> .....	50
Tabla 1.5-5 <i>Especificaciones técnicas de ET200 SP con módulo de interfaz IM 155-6 PN</i> .....	51
Tabla 1.5-6 <i>Mapeo Modbus</i> .....	54
Tabla 1.5-8 <i>Especificaciones de Rendimiento para MAV52</i> .....	61
Tabla 1.5-9 <i>Especificaciones Ambientales para MAV52</i> .....	61
Tabla 1.5-10 <i>Especificaciones Eléctricas para MAV52</i> .....	62
Tabla 1.5-11 <i>Especificaciones Mecánicas para MAV52</i> .....	62
Tabla 1.5-12 <i>Generalidades de los dos motores Hoist</i> .....	63
Tabla 1.5-13 <i>Datos de placa del motor para M3BP 450LB 6</i> .....	64
Tabla 1.5-14 <i>Componentes Eléctricos para M3BP 450LB 6</i> .....	64
Tabla 1.5-15 <i>Especificaciones Mecánicas para M3BP 450LB 6</i> .....	65
Tabla 1.5-16 <i>Dimensiones para M3BP 450LB 6</i> .....	65
Tabla 1.5-17 <i>Especificaciones generales del Rodamiento 6326/C3</i> .....	66
Tabla 1.5-18 <i>Especificaciones técnicas del rodamiento 6326/C3</i> .....	66
Tabla 1.5-19 <i>Especificaciones generales del Rodamiento 6322/C3</i> .....	67

Tabla 1.5-20 <i>Especificaciones técnicas del rodamiento 6326/C3</i> .....	67
Tabla 1.5-21 <i>Normas para vibraciones en máquinas reacondicionadas</i> .....	78
Tabla 2.2-1 <i>Tabla de desempeño y rendimiento de protocolos</i> .....	96
Tabla 3.1-1 <i>Fragmento del Excel importado desde la base de datos sts1_db llenada por el flujo de Recolección de Datos</i> .....	123
Tabla 3.2-1 <i>Costos del proyecto</i> .....	140

## Índice De Planos

PLANO 1 Diseño de red de dispositivos ABB con Raspberry integrada.....	151
PLANO 2 Red implementada de dispositivos SIEMENS en el laboratorio de Sistemas Complejos.....	153

# Capítulo 1

## 1.1.Introducción

En un mundo globalizado y orientado al comercio internacional, los terminales portuarios desempeñan un papel vital como puntos de entrada y salida de mercancías a nivel mundial. Estas instalaciones son esenciales para mantener la fluidez de las cadenas de suministro y preservar la competitividad en un entorno de negocios en constante evolución. No obstante, la operación de estas grúas enfrenta diversos desafíos que amenazan su eficiencia y continuidad. El mantenimiento y la gestión de los componentes que impulsan estas máquinas son un reto significativo.

Este problema afecta a múltiples instalaciones en toda Latinoamérica. La falta de una solución integral para el monitoreo y diagnóstico de estos equipos ha llevado a situaciones en las que los fallos y las averías inesperadas son una amenaza constante. Los sistemas SCADA desempeñan un papel esencial en este proceso, aunque la verdadera innovación radica en la aplicación de la inteligencia artificial (IA) para el mantenimiento predictivo.

## 1.2.Descripción del problema

Un terminal portuario emplea una gran cantidad de motores de inducción para diversas aplicaciones dentro su entorno de trabajo, principalmente los tienen incorporados en grúas para movilización de contenedores. Cada uno con su plan de mantenimiento y un monitoreo e histórico por parte de programas propietarios (bajo licencia). Ocurre que ese registro, de los parámetros de funcionamiento de estos motores y lectura de sensores, no es almacenado de forma especializada para integrarse a sistemas SCADA modernos y tampoco es usado para un proceso de análisis sobre las condiciones de trabajo y recursos utilizados. Y es frecuente encontrarse con averías y desgaste

inesperados, aumentando el riesgo de tiempos de inactividad no planificados, costosas reparaciones y el desperdicio de los recursos.

### **1.3. Justificación del problema**

En Latinoamérica, la manufactura y logística son actividades fundamentales en su economía y en cada hora se realizan cientos de procesos clave uno tras otro, pasando por maquinaria en plantas, estaciones y talleres. El mantenimiento es una práctica vital para mantener el funcionamiento óptimo de esa maquinaria, no obstante, con la evolución en la demanda e incremento del trabajo, los periodos de inactividad por fallos o accidentes son factor crítico que conlleva a la necesidad del monitoreo constante y la anticipación de futuros conflictos. Para ellos se usan las tecnológicas de interfaces multiplataformas para difundir ampliamente el estado de los procesos. La Inteligencia Artificial es utilizada para procesamiento a profundidad de los datos recopilados y machine learning para tener información preparada y adaptada a las condiciones de trabajo.

El presente proyecto consiste en diseñar un sistema de mantenimiento predictivo enfocado en el análisis de la data recopilada de trabajo de la grúa, implementando en un sistema SCADA que presentara las variables de la grúa, relacionadas a las fallas frecuentes, almacenadas en una nube, aprovechando redes neuronales para el análisis de datos. Esto permitirá al área de mantenimiento, anticipar los problemas de los equipos y maquinas, alargando su vida útil y minimizando costos por paralización inesperadas.

Al finalizar, este sistema permitirá evaluar el comportamiento de motor y estructura de la grúa registrando un tiempo estimado del mantenimiento, además de generar alertas cuando se detecten tendencias a un desgaste, reduciendo los retrasos por averías, mejorando la eficiencia y transparencia de los procesos

## **1.4.Objetivos**

### ***1.4.1. Objetivo general***

Diseñar un Sistema de Diagnóstico Predictivo basado en el análisis de parámetros de motores eléctricos y lecturas de los sensores en la grúa aplicando de técnicas de inteligencia artificial para el registro, seguimiento y pronóstico de fallos de grúas de contenedores del terminal portuario.

### ***1.4.2. Objetivos específicos***

1. Realizar la programación y configuración de los equipos para adquirir los datos de un motor de inducción a través de TIA Portal
2. Determinar datos de fallo acordes al motor utilizado a partir del estudio de la literatura existente.
3. Definir un sistema de almacenamiento en base de datos usando servidores basados en TCP u OPC junto al ambiente de NODE-RED.
4. Diseñar un sistema SCADA que incorpore las herramientas de mantenimiento predictivo haciendo uso de Redes Neuronales, basado en el ambiente de NODE-RED.
5. Evaluar el funcionamiento del sistema SCADA, los resultados y pronóstico del mantenimiento predictivo, en base a los fallos ya registrado en una base datos y en el comportamiento de la grúa)

## **1.5.Marco teórico**

En este apartado se definirán los conceptos primarios de una grúa portacontenedores tipo STS, sus componentes y mecanismos, así como los tipos de mantenimiento, enfocándose en las técnicas y principios del mantenimiento predictivo. Además, se expondrán las partes principales de un sistema SCADA en relación a las marcas Siemens y ABB utilizadas en el desarrollo del proyecto, las generalidades del Crane Monitoring System y algoritmos de Inteligencia Artificial como vehículo para realizar el tipo de mantenimiento predictivo a los motores de las grúas.

### **1.5.1 Descripción de una Grúa STS**

Una grúa STS (Ship-to-Shore) es un tipo de grúa utilizada para cargar y descargar contenedores de barcos a muelles o viceversa.

Según Bou Porta (2018), la grúa STS se encuentra instalada sobre raíles y se alimenta eléctricamente mediante una fuente de corriente externa. Posee un carro móvil capaz de manipular contenedores, desplazándolos de un extremo a otro de la grúa. La parte de la grúa que mira hacia el mar cuenta con una pluma tipo voladizo que puede elevarse para permitir el atraque de los barcos.

Esta grúa está equipada con motores eléctricos, así como sus correspondientes sistemas de control. Además, Esparza Herrera (2020) señala que esta dispone de maquinaria para elevar y mover cargas, un conjunto spreader para manipular contenedores, cables de elevación, frenos de servicio y de emergencia, una cabina para el operador de la grúa, una caseta de máquinas y un ascensor, entre otros componentes necesarios para su funcionamiento.

**Tipos de Grúa Portacontenedores.** Hoy en día, la capacidad de carga de un barco de contenedores se determina mediante la unidad de medida conocida como TEUs, donde un TEU equivale a un contenedor estandarizado de 20 pies de longitud (aproximadamente 6.1 metros). En la mayoría de los casos, los barcos se cargan con contenedores de 20 y 40 pies, lo que significa que se utilizan 1 TEU y 2 TEUs respectivamente para calcular su capacidad de carga.

Las naves de contenedores se clasifican según su capacidad de carga a lo largo del tiempo.

**Feeder.** La grúa Feeder cuenta con espacio en su ancho para la manipulación de hasta 10 contenedores, tanto en la carga como en la descarga.

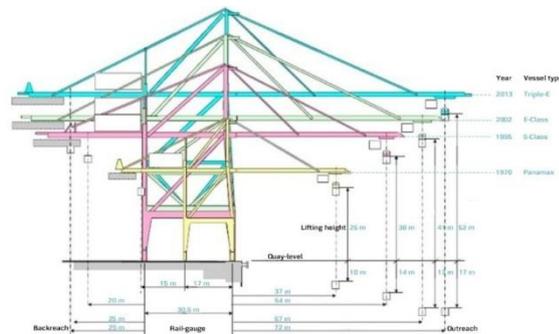
**Panamax.** Originadas a partir de la década de 1970, la grúa Panamax está diseñada para cargar y descargar contenedores de barcos portacontenedores que tienen la capacidad de navegar a través del Canal de Panamá, con espacio para manipular hasta 13 contenedores y con una capacidad de transporte de 4,500 TEUs. Estas embarcaciones poseen dimensiones que alcanzan hasta 290 metros de eslora y 32 metros de manga.

**Post-panamax.** Desarrolladas a partir de la mitad de los años 80 y con una capacidad de transporte que oscila entre 6000 y 8500 TEUs, estas embarcaciones presentan dimensiones aproximadas de 340 metros de eslora y 43 metros de mango. La grúa Post-Panamax, dispone de espacio en su ancho para manipular entre 16 y 20 contenedores y tiene un peso estimado de alrededor de 800-900 toneladas.

**Super post-panamax.** La grúa Super Post-Panamax está diseñada para manejar entre 22 y 24 contenedores y tiene un peso de aproximadamente 1800 toneladas. Estas naves, pertenecientes a una generación desarrollada a partir de la década de los 90, representan las embarcaciones más avanzadas en la actualidad, con una capacidad de transporte que oscila entre 18000 y 20000 TEUs. En la actualidad, estas naves tienen dimensiones aproximadas de 400 metros de longitud y 59 metros de ancho.

**Figura 1.5.1**

*Esquema del desarrollo de las grúas STS*



En la Figura 1.5.1 se muestra la evolución que han tenido las grúas STS a través de las décadas.

### Componentes

**Viga Principal (Boom).** Constituida por una viga metálica, la estructura consta de cuatro patas que se conectan en la parte inferior a las vigas de los extremos en los lados tierra y mar, la cual facilita el desplazamiento del carro, la cabina, el equipo de elevación y, por ende, los contenedores. Es una construcción compuesta por chapas laminadas que se sueldan para formar vigas tipo cajón, y sus partes principales se conectan mediante bridas que se aseguran con pernos de alta resistencia. Bou Porta (2018) sostiene que, a media altura, se refuerzan mediante vigas portal y tubos diagonales en los lados izquierdo y derecho. Además, en la parte superior, se refuerzan con vigas transversales en los lados tierra y mar. La estructura superior está formada por

una estructura en forma de "A" compuesta por dos patas verticales, una viga de cierre superior y dos tubos traseros de refuerzo.

***Caseta de maquinaria.*** La sala de máquinas de una grúa STS es un espacio cerrado ubicado sobre la viga principal de la grúa. Este espacio está dividido en tres zonas principales:

- **Sala mecánica:** Se encuentran los tambores enrolladores de cable del sistema de traslación del carro, el cable usado en el levantamiento de la pluma y el cable utilizado en el sistema de elevación y descenso de la carga. También se encuentran los motores eléctricos que hacen funcionar estos tambores, junto con sus reductores, frenos y motores auxiliares.
- **Sala de control:** Se encuentran todos los dispositivos de control de la grúa, excepto el de izado de la pluma, que a menudo se encuentra en otra cabina cerca de la zona articulada de la pluma.
- **Sala eléctrica:** Se encuentra todo el sistema eléctrico y electrónico de la grúa, incluyendo variadores, tarjetas de control, contactores de potencia y maniobra, relés de protección térmicos, relojes de medida analógicos, etc.

Fernández Pérez (2009) señala que, además de estas tres zonas principales, la sala de máquinas también puede incluir una zona de suministro eléctrico, que contiene el transformador principal, un desconector de alto voltaje y un panel de corrección de la potencia. La construcción de la sala de máquinas está hecha de un recubrimiento de acero de lana mineral que actúa como aislante eléctrico.

**Figura 1.5.2**

*Caseta de maquinaria de la grúa STS 3 del Terminal Portuario*



***Cabina del operador.*** La cabina del gruista es un espacio ubicado en el carro de la grúa que permite al operador tener una visión adecuada del campo de trabajo. La cabina está construida de chapa de acero y está perfectamente aislada para proteger al operador de las condiciones climáticas. Los mandos de la cabina permiten al operador realizar las operaciones de elevación, movimiento del carro y del pórtico.

***Carro.*** El carro se utiliza para transportar contenedores a lo largo de la grúa. No tiene motores, y su movimiento se consigue mediante un sistema funicular. Está conectado al bastidor superior del spreader mediante un sistema de cables, lo que permite la elevación y el descenso de las cargas.

Para Bou Porta (2018), el carro está equipado con un sistema tensor del cable del carro operado hidráulicamente. Este sistema mantiene la tensión adecuada del cable durante el trabajo normal del carro y permite aflojar el cable al levantar la pluma. Cuando se levanta la pluma, los controles eléctricos actúan automáticamente para reducir la presión del sistema tensor. Esto evita que los cables del carro se tensen demasiado.

**Spreader Telescópico.** El spreader es un dispositivo que se utiliza para levantar y bajar contenedores. Según Bou Porta (2018), el spreader consta de un bastidor principal central, dos pares de vigas telescópicas, vigas laterales, brazos extensibles y un sistema de cerrojos giratorios. La longitud del spreader se puede ajustar para adaptarse al tamaño de distintos contenedores.

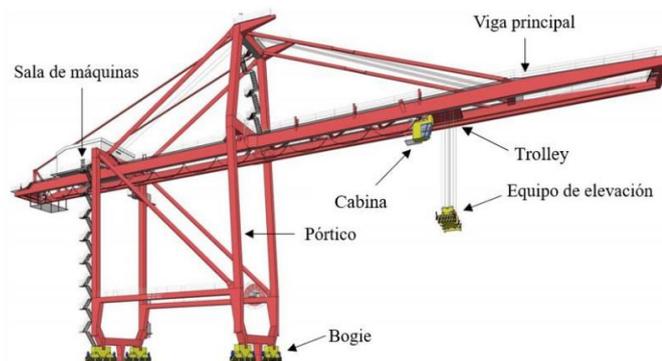
**Figura 1.5.3**

*Spreader utilizado en el movimiento Hoist de la grúa*



**Figura 1.5.4**

*Componentes de una grúa STS*



### **Mecanismo de elevación**

Para poder realizar la operación de elevación, las siguientes condiciones deben de ser cumplidas:

- La grúa rearmada
- La planta hidráulica en marcha
- Los ventiladores de los motores de elevación en marcha
- El PLC en marcha

### **Elementos mecánicos y eléctricos**

El equipo mecánico de elevación este situado en la sala de máquinas.

*Acoplamiento electromagnético.* Cuando no hay corriente el acoplamiento electromagnético está acoplado por la fuerza de los muelles. El acoplamiento se abre, cuando se opera la nivelación. Al girar el selector de nivelación fuera de la posición cero, la excitación eléctrica del acoplamiento activa los imanes y la fuerza magnética, siendo mayor a la de los muelles, obliga la abertura del acoplamiento.

*Tambores de cable.* Los tambores de cable sirven para almacenar los cuatro cables de elevación. Unas puntas de todos los cables están fijadas en el tambor, las otras puntas están fijadas en la punta de la pluma. Los cuatro cables son soltados o recogidos en el mismo tambor y por lo tanto su movimiento es totalmente sincronizado

*Cables de elevación.* Los cables de elevación suelen ser de acero trenzado de 28 mm de diámetro.

**Frenos de elevación.** Los dos frenos de elevación funcionan a falta de corriente, son de disco de dos zapatas y su funcionamiento es paralelo. Un muelle mantiene los frenos siempre cerrados, cuando la elevación no está operada o la grúa está parada.

La abertura del freno se efectúa con un actuador electrohidráulico. Cuando el operador opera la palanca de elevación, los frenos son abiertos por el actuador electrohidráulico después de activar el campo magnético de los motores de elevación, pero antes de arrancarse los motores. Esto impide que cese la carga suspendida durante el intervalo de abrir el freno y arrancar el motor.

**Motor Hoist:** El motor de elevación, también conocido como motor de cabrestante, es el motor que se encarga de elevar la carga de una grúa STS. Se encuentra ubicado en la parte superior de la grúa, en la pluma. El motor de elevación está conectado a la pluma a través de un sistema de engranajes y correas.

El motor de elevación suele ser un motor eléctrico de alta potencia, capaz de levantar cargas de hasta 20 toneladas. El motor está controlado por un sistema de control electrónico que permite realizar movimientos precisos y suaves.

**Motor Trolley.** El motor de carro, también conocido como motor de traslación, es el motor que se encarga de mover el carro de una grúa STS. Se encuentra ubicado en la parte inferior de la grúa, en la estructura del carro. El motor de carro está conectado al carro a través de un sistema de engranajes y correas.

El motor de carro suele ser un motor eléctrico de mediana potencia, capaz de mover cargas de hasta 10 toneladas. El motor está controlado por un sistema de control electrónico que permite realizar movimientos precisos y suaves.

### 1.5.2 *Mantenimiento Industrial*

De acuerdo con Vázquez (2014), el mantenimiento es un conjunto de acciones planificadas y ejecutadas para conservar en buen estado los equipos y maquinarias, con el objetivo de garantizar su funcionamiento óptimo y evitar averías.

Los objetivos del mantenimiento son:

- Disponibilidad: Garantizar que los equipos estén disponibles para su uso cuando se necesiten.
- Funcionamiento: Mantener los equipos en buen estado para que funcionen correctamente.
- Calidad: Obtener productos finales de calidad.
- Eficiencia: Evitar pérdidas económicas.
- Durabilidad: Prolongar la vida útil de los equipos.

#### **Tipos de mantenimiento**

*Mantenimiento Correctivo.* Torres (2005) define al mantenimiento correctivo como un tipo de mantenimiento que se realiza después de que se ha producido una falla. Su objetivo es reparar el equipo para que vuelva a funcionar correctamente.

*Mantenimiento Preventivo.* El mantenimiento preventivo es un tipo de mantenimiento que se realiza con el objetivo de evitar que se produzcan fallas. Para Pinto (2002), su objetivo es inspeccionar y reparar los equipos antes de que se produzcan averías.

***Mantenimiento Predictivo.*** De acuerdo con Torres (2005), el mantenimiento predictivo es un tipo de mantenimiento que se basa en el análisis de datos para predecir cuándo se producirá una falla. Su objetivo es intervenir antes de que se produzca la falla para evitar que se produzcan daños mayores.

***Mantenimiento Proactivo.*** El mantenimiento proactivo es un tipo de mantenimiento que va más allá del mantenimiento preventivo. Su objetivo es mejorar el rendimiento y la eficiencia de los equipos mediante la identificación y eliminación de las causas de las fallas (Vázquez, 2014).

**Categorización de equipos en planta.** El mantenimiento de máquinas se puede clasificar según su criticidad, es decir, según la importancia que tiene la máquina para el funcionamiento de la empresa o instalación.

El mantenimiento crítico es el que se realiza en máquinas que son esenciales para el funcionamiento de la empresa o instalación. La falla de estas máquinas puede provocar un paro total o parcial de la producción, lo que puede tener un impacto negativo significativo en la empresa (Pinto, 2002).

El mantenimiento esencial es el que se realiza en máquinas que son importantes para el funcionamiento de la empresa o instalación, pero no son esenciales. La falla de estas máquinas puede provocar un impacto negativo en la empresa, pero no un paro total o parcial de la producción (Torres, 2005).

El mantenimiento de propósito general es el que se realiza en máquinas que no son esenciales ni importantes para el funcionamiento de la empresa o instalación. La falla de estas máquinas puede provocar un impacto negativo menor en la empresa (Vázquez, 2014).

**Técnicas de Mantenimiento Predictivo.** Las técnicas de mantenimiento predictivo se pueden clasificar en dos categorías principales:

- **Técnicas de análisis de la condición:** Estas técnicas se basan en el análisis de las condiciones físicas de la máquina para predecir cuándo se producirá una falla.
- **Técnicas de análisis de la salud:** Estas técnicas se basan en el análisis de los datos de funcionamiento de la máquina para predecir cuándo se producirá una falla.

Algunos ejemplos de técnicas de mantenimiento predictivo:

- **Análisis de vibraciones:** Esta técnica se utiliza para detectar cambios en las vibraciones de la máquina que pueden indicar una falla inminente.
- **Análisis de aceite:** Esta técnica se utiliza para detectar partículas de desgaste en el aceite que pueden indicar una falla inminente.
- **Termografía:** Esta técnica se utiliza para detectar cambios en la temperatura de la máquina que pueden indicar una falla inminente.
- **Análisis de corriente:** Esta técnica se utiliza para detectar cambios en la corriente que fluye a través de la máquina que pueden indicar una falla inminente.
- **Análisis de ultrasonido:** Esta técnica se utiliza para detectar defectos en los componentes de la máquina que pueden indicar una falla inminente.

### 1.5.3 Sistema SCADA

**Autómata Programable.** Según la NEMA (National Electrical Manufacturers Association), un PLC es un dispositivo electrónico digital operado digitalmente que usa una memoria programable para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas, lógica, secuenciación, temporización, conteo y control de funciones de movimiento.

El funcionamiento de un PLC se puede dividir en las siguientes etapas:

- **Adquisición de datos:** El PLC recibe señales de entrada de los sensores. Estas señales pueden ser analógicas o digitales.
- **Procesamiento de datos:** El PLC procesa las señales de entrada de acuerdo con el programa de control.
- **Emisión de señales de salida:** El PLC emite señales de salida a los actuadores. Estas señales pueden ser analógicas o digitales.

El programa de control de un PLC se puede escribir en un lenguaje de programación específico para PLC. Los lenguajes de programación de PLC más comunes son: Ladder diagram (LD), Structured text (ST), Function block diagram (FBD), Instruction list (IL)

Los PLC se pueden clasificar en tres categorías principales:

- **Compactos:** Son los PLC más pequeños y sencillos. Están diseñados para aplicaciones sencillas, como la automatización de máquinas.

- Semi compactos: Son más grandes y complejos que los PLC compactos. Están diseñados para aplicaciones más exigentes, como la automatización de procesos industriales.
- Modulares: Son los PLC más grandes y complejos. Están diseñados para aplicaciones muy específicas, como la automatización de plantas de producción.

A continuación, se presentan las generalidades de las dos marcas utilizadas para este proyecto, ABB como proveedor en la mayoría de los equipos utilizados en el Terminal Portuario y SIEMENS como equivalente a utilizar en el Laboratorio de Sistemas Complejos de la ESPOL, donde se realizará la programación, parametrización y adquisición de datos como parte de la propuesta a presentar al cliente.

De la marca ABB se tiene a la CPU AC500-eCo y tipo PM564-R-ETH-AC, programable con la plataforma Automation Builder y con las siguientes características:

- Es un modelo modular con posibilidad de expansión con hasta 10 módulos.
- Tiempos de ciclo mínimo para 1 instrucción: bit: 0.08 $\mu$ s, palabra (Word): 0.1 $\mu$ s y coma flotante (DWord): 1.2 $\mu$ s
- Posibilidad de configuración de las entradas/salidas integradas en la CPU: entradas de interrupción, contaje rápido y salidas PWM.
- Tarjeta SD opcional y estándar para la descarga del programa de usuario, actualización del firmware y data logger. x Posibilidad de tener hasta dos puertos serie (1 puerto integrado y otro opcional) para programación y comunicación Modbus RS485. Interfaces/Protocolos

- La interfaz de Ethernet puede ser usada para programación y comunicación (protocolos SNTP, DHCP, UDP, Modbus TCP, HTTP y FTP).
- La interfaz serie COM1 (RS485) puede ser usada para programación, comunicación o como E/S remotas (protocolos MODBUS RTU, maestro CS31 para E/S remotas o ASCII).
- La interfaz serie COM2 (RS485) opcional puede ser usada para programación o comunicación (protocolos MODBUS RTU o ASCII). (ABB, 2016)

**Tabla 1.5-1**

*Especificaciones técnicas del Controlador AC 500*

Tipo	Memoria de Programa	E/S Integradas DI/D0/AI/A0	Tipos de salida	Comunicación integrada	Alimentación	Peso Kg
PM595-ETH-AC	128kB	6/6/2/1	relé	Ethernet	100-240VCA	0.400

Nota: Información obtenida en catalogo ABB.

**Figura 1.5.5**

*Imagen del Controlador AC 500*



De la marca SIEMENS se tiene la SIMATIC S7-1500 CPU compacta CPU 1516-3 PN/1 DP, programable con la plataforma TIA Portal y con las siguientes características:

- Módulo central con memoria principal 250 KB para programa y 1 MB para datos.
- Tiene 32 entradas digitales, 32 salidas digitales, 5 entradas analógicas, 2 salidas analógicas, 6 contadores rápidos, 4 salidas rápidas para PTO/PWM/salida de frecuencia interfaz.
- La gama incluye por un lado CPUs estándares, en parte con la posibilidad de integrar código en C/C++.
- El Advanced Controller SIMATIC S7-1500 se configura con el software SIMATIC STEP 7 Professional.
- Para el funcionamiento de este módulo es necesaria una SIMATIC Memory Card.

**Tabla 1.5-2**

*Especificación S7 1500*

Voltaje de alimentación	24 VDC
Tiempo de ejecución de operaciones con bits	48 ns
Tiempo de ejecución de operaciones con palabras	58 ns
Tiempo de ejecución de operaciones con aritméticas	77 ns
Tiempo de ejecución de operaciones con aritméticas con decimales	307 ns
Interfaz Profinet	2
Memoria de trabajo para programa	250 kB
Memoria de trabajo para datos	1MB
Contadores rápidos	6
Generadores de impulsos (PWM,PTO)	4

SIMATIC Memory Card necesaria	Si
Servidor Web Integrado	Si

**Figura 1.5.6**

*Controlador S7-1500*



**Módulos de entradas y salidas digitales.** Las entradas y salidas digitales se refieren a variables booleanas, lo que significa que solo pueden tener dos estados posibles.

- Las entradas digitales son señales discretas que cambian su estado cuando detectan o no tensión. Esto implica que pueden tomar valores de 1 o 0. Algunos autómatas pueden trabajar con entradas digitales de 24VDC, mientras que otros funcionan con tensiones de 110-220V AC.
- Las salidas digitales tienen como principal objetivo activar o desactivar actuadores, como relés, contactores o lámparas. En los autómatas, el funcionamiento de las salidas digitales puede variar; algunos envían una señal de 24V a los actuadores, mientras que otros disponen de relés integrados como salidas digitales.

**Figura 1.5.7**

*Módulos de entradas y salidas digitales para un S7-1500*



**Figura 1.5.8**

*Módulos de salidas digitales para un AC 500*



Fuente: (Group, ABB, 2016)

**Módulo de Entradas y salidas analógicas.** Las entradas y salidas analógicas generalmente se utilizan para monitorear sensores o establecer referencias para controlar equipos, y en su mayoría transmiten datos en forma de valores de corriente o voltaje.

- En cuanto a las entradas analógicas, operan detectando tensiones o corrientes, y normalmente provienen de sensores. Los valores capturados se utilizan en su mayoría para compararlos con consignas definidas por el usuario. Estas entradas requieren sensores analógicos específicos que sean compatibles con las señales que el sistema automatizado maneja. Existen ciertos estándares para asegurar que los dispositivos funcionen adecuadamente. En términos de voltaje, el estándar más común es de 0 a 10V DC, aunque

también se utilizan rangos como -10 a +10V o 2 a 10V. En lo que respecta a la corriente, el estándar más extendido es de 4 a 20mA, pero existen otras variantes como 0 a 20mA, 1 a 5mA, y 0 a 5mA.

- En el caso de las salidas analógicas, los módulos de salida se encargan de adaptar la señal a los estándares convencionales. Para las conexiones entre módulos, se recomienda el uso de cables apantallados para prevenir interferencias y perturbaciones en la señal, lo que contribuye a garantizar una señal más fiable y, por ende, un mejor rendimiento.

El módulo analógico utilizado en ABB es el AI561. Este módulo consta de:

- 4 entradas analógicas que pueden ser configurables individualmente con señales de 4...20mA, 0...+10V, 0... +5V, entre otras.
- La resolución de entradas depende del tipo de señal, para 4...20mA es de 12 bits.
- Su alimentación es de 24 Vdc y puede ser tomada desde la fuente interna de la CPU o con la ayuda de una fuente externa. (ABB, 2016)

**Figura 1.5.9**

*Módulos de entradas analógicas para un S7-1500*



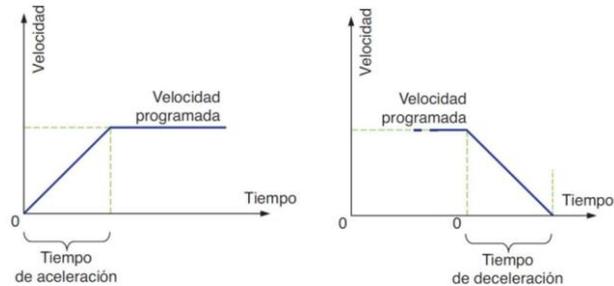
Fuente: (Group, ABB, 2016)

**Variadores de frecuencia.** Un variador de velocidad, también conocido como inversor de frecuencia o drive, es un dispositivo electrónico utilizado en sistemas de control de motores eléctricos para regular y ajustar la velocidad y la potencia del motor. Este dispositivo es capaz de variar la frecuencia y la tensión suministrada al motor, lo que permite un control preciso de la velocidad de rotación y, en consecuencia, del rendimiento del motor. En cuanto a sus principales características se encuentran:

- **Ahorro de Energía:** Los variadores de velocidad pueden optimizar el consumo de energía al ajustar la velocidad del motor según la demanda. Esto puede conducir a un ahorro significativo de energía en comparación con motores que funcionan a velocidad constante.
- **Arranque Suave:** Los variadores permiten un arranque suave del motor, lo que reduce el estrés mecánico y eléctrico en el equipo, prolongando así su vida útil.
- **Control de Par:** Además del control de velocidad, los variadores de velocidad también pueden regular el par del motor, lo que es valioso en aplicaciones donde se requiere un control preciso del esfuerzo de torsión.
- **Rampa de Aceleración y Desaceleración:** Una característica importante de los variadores de velocidad es la capacidad de programar rampas de aceleración y desaceleración. Estas rampas permiten que el motor aumente o disminuya gradualmente su velocidad, evitando picos de corriente y reduciendo el desgaste mecánico.

**Figura 1.5.10**

*Rampas de Aceleración y Desaceleración en un Variador de Frecuencia*



- **Protección del Motor:** Los variadores de velocidad incluyen características de protección, como la detección de sobre corriente y sobrecalentamiento, que protegen el motor contra condiciones adversas y evitan daños.

Con respecto a las marcas manufactureras de ABB y Siemens, la primera tiene la línea de variadores para motores de inducción ACS y la segunda los G120, con sus respectivos modelos de unidad de control y potencia.

En ABB se manufacturan los convertidores de frecuencia únicos ACS800.

- Grados de protección IP21 e IP55
- Control directo de par (DTC) de serie
- Filtro EMC, reactancia y chopper de frenado
- Admite una amplia gama de protocolos de bus de campo
- Opciones flexibles de E/S y de encoder

**Figura 1.5.11**

*Variador de Frecuencia ACS800*



En SIEMENS se manufacturan los SINAMICS G120 Control Units CU250S-2

**Tabla 1.5-3**

*Especificaciones técnicas del Variador de Frecuencia SINAMICS G120*

Entrada	Valor	Salidas Digitales	Valor
Numero de Fases	3 AC	Numero de conmutados de Relé	1
Tensión Red	390 .. 480V	Salida (carga resistiva)	DC 30V,1A
Frecuencia de red	47 .. 63Hz	Numero de Transistor	1
Intensidad de Entrada	18,20 A	Salida (carga resistiva)	DC 30V,1A
Salida		Entradas analógicas/digitales	
Numero de Fases	3 AC	Cantidad	1 diferencial
Tensión Asignada	400V	Salidas Analógicas	
Potencia Asignada	5.50 kW	Cantidad	1 no aislada
Intensad Asignada	12,50A	Comunicación	
Frecuencia de Pulsación	4 KHz	Puerto	Profinet

**Figura 1.5.12**

*Variador de Frecuencia SINAMICS G120*



***Periferia Descentralizada.*** La periferia descentralizada, en el contexto de la automatización industrial, se refiere a un enfoque en el que los dispositivos y sistemas periféricos se ubican físicamente cerca de la maquinaria, equipos o procesos que controlan, en lugar de estar centralizados en un único punto de control. Esto implica la distribución de la inteligencia y la capacidad de procesamiento en el entorno industrial, lo que permite una mayor flexibilidad y eficiencia en la gestión de sistemas automatizados. En cuanto a sus principales características se encuentran:

- **Mayor Flexibilidad:** Al tener dispositivos periféricos inteligentes, es más sencillo adaptar y reconfigurar sistemas automatizados para nuevas tareas o cambios en la producción. Esto se logra sin requerir modificaciones extensas en un controlador central.
- **Reducción de Cables:** La periferia descentralizada permite la reducción de cables largos y costosos, ya que los dispositivos periféricos pueden comunicarse de forma inalámbrica o a través de redes de campo, como Profibus o Profinet.

- **Menor Latencia:** Al descentralizar la inteligencia, se reduce la latencia en la toma de decisiones, lo que es crítico en aplicaciones en tiempo real, como robótica y sistemas de control de procesos.
- **Facilita la Ampliación:** La periferia descentralizada es escalable y facilita la ampliación de sistemas, ya que nuevos dispositivos periféricos pueden integrarse de manera eficiente sin alterar el sistema existente.
- **Mayor Robustez:** Los sistemas descentralizados pueden ser más robustos frente a interferencias electromagnéticas y condiciones ambientales adversas, ya que los componentes periféricos están más cerca de los sensores y actuadores.

Para ABB se tiene los modelos AC500 CI y para Siemens se usan los equipos ET200, con sus respectivos módulos de comunicación y módulos de entradas y salidas.

***AC500 CI502 PROFINET communication interface module***

**Tabla 1.5-4**

*Especificaciones técnicas de la Periferia Descentralizada CI502 para AC500*

<b>Parámetro</b>	<b>Valor</b>
Voltaje de Alimentación	24 VDC
Máxima Carga	10A
Protección contra voltaje inverso	Si
Máxima potencia de disipación	6W
Entradas digitales	8
Salidas digitales	8
Entradas/salidas digitales configurables	8
Tamaño de datos de entrada	12 bytes
Tamaño de datos de salida	20 bytes

Diagnósticos	Visible en pantalla
Indicadores de operación y error	34 LEDs

**Figura 1.5.13**

*Periferia Descentralizada CI502 para AC500*



**ET200 SP Módulo de interfaz IM 155-6 PN.** El módulo de interfaz SIMATIC ET 200SP IM 155-6 PN BA - 6ES7155-6AR00-0AN0 de Siemens está diseñado para aplicaciones PROFINET dentro de sistemas con hasta 12 módulos periféricos (hasta 192 señales de entradas/salidas) y 32 byte cada entrada/salida. El módulo se monta sobre carril DIN. El módulo de interfaz SIMATIC ET 200SP IM 155-6 PN BA con Switch de dos puertos integrado y módulo servidor es la solución más rentable para instalaciones sencillas

**Tabla 1.5-5**

*Especificaciones técnicas de ET200 SP con módulo de interfaz IM 155-6 PN*

Módulo de interfaz	IM 155-6 PN ST
Referencia	6Es7155-6AA01-0BN0
Firmware	v4.1
Tensión de alimentación	24 VDC
Límite inferior de alimentación	19.2V
Límite superior de alimentación	28.8V
Intensidad consumida nominal	450 mA

Intensidad consumida máxima	550 Ma
Potencia del bus	4.5 W
Numero de módulos	32
Numero de puertos	2 puertos RJ45
Método de transferencia	Profinet 100 Mbp/s
LED arranque	Verde
LED error	Rojo
LED mantenimiento	amarillo

**Figura 1.5.14**

*Periferia Descentralizada ET200 SP con módulo de interfaz IM 155-6 PN*



**Protocolos de Comunicación.** Los protocolos de comunicación son conjuntos de reglas y estándares que gobiernan la transmisión de datos entre dispositivos electrónicos o sistemas informáticos. Estas reglas determinan cómo se estructuran, transmiten, reciben y procesan los datos, asegurando una comunicación eficiente y confiable entre los dispositivos interconectados.

***Modbus TCP: Protocolo de comunicación de datos.***

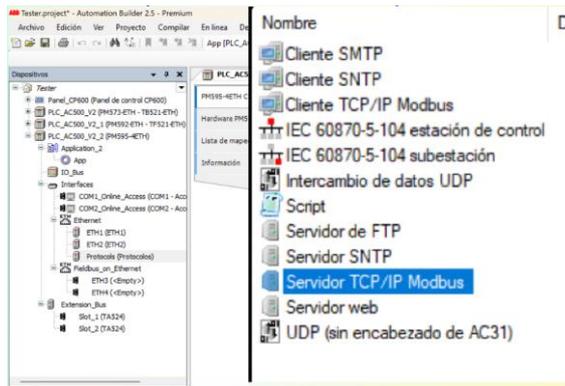
- Basado en TCP/IP, se utiliza comúnmente en aplicaciones industriales para la comunicación entre dispositivos de automatización.

- Es simple y ampliamente adoptado en sistemas de control industrial.
- Utiliza modos de lectura/escritura (Read/Write) para acceder a datos en dispositivos Modbus.
- Puede ser implementado en redes Ethernet, lo que facilita su integración en sistemas modernos.

Para los PLC de ABB y SIEMENS, la integración de un protocolo Modbus es mediante una instancia de servidor Modbus, que se consigue mediante el uso de las librerías de sus respectivas plataformas como se parecía en la Figura 1.5.15 y 1.5.16.

**Figura 1.5.15**

*Adición de Servidor Modbus en Automation Builder*



**Figura 1.5.16**

*Adición de Servidor Modbus en TIA Portal*

Communication	
Name	Description
57 communication	
Open user communicati...	
WEB Server	
Others	
MODBUS TCP	
MB_CLIENT	Communicate via PROFINET as Modbus TCP client
MB_SERVER	Communicate via PROFINET as Modbus TCP server
Communication processoi	

En ambos casos, se hace uso de las direcciones Modbus para que el cliente pueda referirse a los datos en los registros para las tareas de lectura o escritura, como se describe en la Tabla

**Tabla 1.5-6**

*Mapeo Modbus*

Direcciones de Registro Modbus	Direcciones de Data	Acción	Área
1-9999	0000 a 270E	Lectura/Escritura	Salidas Discretas
10001-19999	0000 a 270E	Solo Lectura	Entradas Discretas
30001-39999	0000 a 270E	Solo Lectura	Entradas Analógicas
40001-49999	0000 a 270E	Lectura/Escritura	Registro de Sostenimiento

***Profibus DP (Decentralized Peripherals): Protocolo de campo industrial (fieldbus).***

- Diseñado para la comunicación entre dispositivos de automatización y control, como sensores y actuadores, en aplicaciones industriales.
- Proporciona alta velocidad y capacidad de datos para aplicaciones en tiempo real.
- Adecuado para aplicaciones de control y monitorización de procesos industriales.
- Utiliza el formato de trama maestra/esclavo para la comunicación.

***Profibus PA (Process Automation): Protocolo de campo industrial (fieldbus).***

- Diseñado específicamente para la industria de procesos, como la química y petroquímica.
- Proporciona comunicación digital entre sensores y actuadores en entornos potencialmente explosivos.

- Ofrece una alta inmunidad a interferencias eléctricas y permite la transmisión de datos críticos de proceso.

***Profinet: Protocolo de comunicación industrial basado en Ethernet.***

- Combina las ventajas de Ethernet con la comunicación en tiempo real requerida en aplicaciones industriales.
- Proporciona alta velocidad y rendimiento para aplicaciones de automatización industrial avanzadas.
- Permite la integración de sistemas de control y sistemas de TI (Tecnología de la Información).
- Es altamente flexible y escalable, lo que lo hace adecuado para una amplia gama de aplicaciones en la industria.

#### **1.5.4 Crane Monitoring System**

El Crane Monitoring System (Sistema de Monitoreo de Grúas) es un sistema de control y supervisión utilizado en aplicaciones industriales para gestionar y monitorear grúas y otros equipos de manejo de materiales. El CMS se encarga de garantizar la seguridad y eficiencia de las operaciones de elevación y manejo de cargas en entornos industriales. En cuanto a sus características principales se encuentran:

- Supervisión en tiempo real: El CMS permite el monitoreo continuo de la grúa y sus operaciones, lo que facilita la detección de posibles problemas o condiciones de riesgo.
- Seguridad: Proporciona sistemas de seguridad, como límites de carga y alertas de sobrecarga, para prevenir accidentes.

- Registro de datos: Almacena información detallada sobre el funcionamiento de la grúa en un formato especial para visualización en la interfaz gráfica CMS.
- Interfaz de usuario: Ofrece una interfaz gráfica que permite a los operadores supervisar y controlar la grúa de manera intuitiva.
- Comunicación: Puede estar integrado en sistemas de control más amplios, lo que facilita la comunicación con otros dispositivos y sistemas en la planta.

Los sistemas de monitoreo de grúas suelen generar dos tipos de datos .cms:

- Datos de eventos: Los datos de eventos registran los eventos que ocurren en la grúa, como el inicio del movimiento, el fin del movimiento y la sobrecarga.
- Datos de estado: Los datos de estado registran el estado actual de la grúa, como la posición de la carga, la velocidad de la grúa y la temperatura del motor.

La tarjeta de control CMS (Crane Monitoring System) es un módulo que se utiliza para recopilar datos de los sensores de la grúa y para controlar la grúa. La tarjeta de control CMS suele estar integrada en el PLC de la grúa.

En los tableros de automatización, la tarjeta de control CMS se utiliza para controlar y monitorear los movimientos de la grúa. La tarjeta recopila datos de los sensores de la grúa, como los sensores de carga, los sensores de temperatura y los sensores de movimiento. Estos datos se utilizan para controlar la grúa y para generar informes de seguridad.

Las tarjetas de control CMS suelen tener las siguientes características:

- Entradas y salidas digitales y analógicas: Las entradas digitales se utilizan para recibir señales de los sensores de la grúa. Las salidas digitales se utilizan para controlar los actuadores de la grúa. Las entradas y salidas analógicas se utilizan para recopilar datos de los sensores analógicos de la grúa.
- Protocolos de comunicación: Las tarjetas de control CMS suelen soportar una variedad de protocolos de comunicación, como Modbus TCP, Profibus y Profinet.
- Software de programación: Las tarjetas de control CMS suelen venir con un software de programación que permite al usuario configurar la tarjeta y generar el código de control.

Existen fabricantes como Siemens y ABB que incluyen este tipo de tarjetas:

- Siemens incluye las tarjetas de control CMS integradas en su sistema de automatización industrial SIMATIC y CCU (Crane Control Unit)
- ABB proporciona tarjetas que están diseñadas para trabajar con los sistemas de control que se integran en aplicaciones que involucran grúas y manejo de materiales, como en CCM (Crane Control Module)

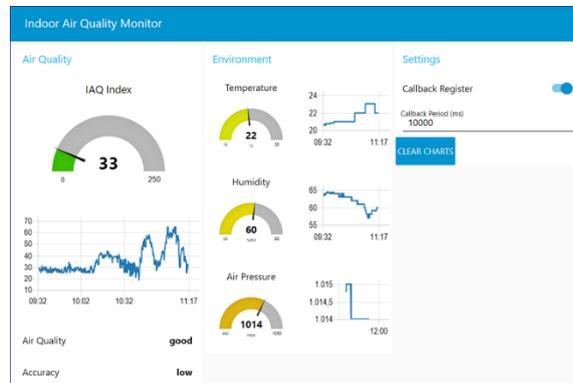
### ***1.5.5 Node RED***

Node-RED es una plataforma de código abierto desarrollada para la creación de aplicaciones IoT (Internet de las cosas) de forma visual. Utiliza un enfoque basado en flujos de trabajo (flows) para conectar dispositivos y servicios, permitiendo la integración de sistemas de manera sencilla y eficiente. Opera sobre Node.js, un entorno de ejecución de JavaScript del lado del servidor. La arquitectura de Node-RED se compone de nodos que representan funcionalidades específicas y se conectan mediante flujos. Los nodos pueden ser desde simples funciones hasta integraciones complejas con servicios externos.



**Figura 1.5.19**

*Panel de Node RED*



Se accede desde cualquier navegador con la dirección localhost:1880/ui.

Node RED incluso permite usar código Python mediante sus nodos de Python Function Ps. Python es un lenguaje de programación de propósito general de alto nivel, interpretado, con tipado dinámico y multiparadigma. Es uno de los lenguajes de programación más populares del mundo, y se utiliza en una amplia gama de aplicaciones, incluyendo ciencia de datos, aprendizaje automático, ingeniería de software, web development, y más.

Scikit-learn es una biblioteca de aprendizaje automático de código abierto para Python. Es una de las bibliotecas de aprendizaje automático más populares y utilizadas, y ofrece una amplia gama de algoritmos de aprendizaje automático para una variedad de tareas, incluyendo clasificación, regresión, clustering, y reducción de dimensionalidad.

### **1.5.6 Sistema Motor-rodamiento de Hoist CMS**

El sistema de monitoreo de condición (CMS) incorpora una red integral de sensores de vibración, diseñados para capturar y analizar de manera precisa las vibraciones generadas por el motor y los rodamientos. Estos sensores desempeñan un papel fundamental en la evaluación del estado operativo, permitiendo la detección temprana de posibles anomalías. Los sensores de

vibración instalados en el motor registran las oscilaciones y frecuencias asociadas con su funcionamiento, proporcionando datos valiosos sobre la integridad estructural y el rendimiento dinámico. Además, los sensores dedicados a los rodamientos permiten monitorear la salud de estos componentes cruciales, identificando potenciales desgastes o irregularidades en su funcionamiento.

**Sensor de Vibración: Acelerómetro de velocidad (Current Output).** Según su fabricante Honeywell Test & Measurement, el modelo MAV52 es ideal para aplicaciones en las que la velocidad de la vibración es un parámetro más importante para medir y en las que los problemas de ruido o la simplicidad del sistema hacen que la integración de la señal de aceleración en el sensor sea más atractiva que el acondicionamiento de señal.

**Figura 1.5.20**

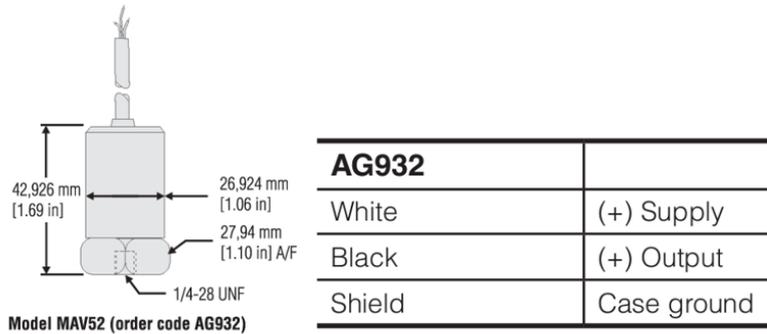
*Acelerómetro Modelo: 060 J079 01*



Entre sus funcionalidades se encuentran: la medición de velocidad, rango de frecuencia usable de 2 Hz a 10 kHz, 4 mA a 20 mA proporcional a la velocidad, opción sumergible, con cable blindado de acero inoxidable integral o conector o una opción intrínsecamente segura.

**Figura 1.5.21**

*Dimensiones y código de cableado del Modelo MAV52 Código AG932*



Es de tipo AG932SE,17B, salida de cable superior, sensibilidad 1 pulg/sec, resonancia 5khz.

**Tabla 1.5-7**

*Especificaciones de Rendimiento para MAV52*

Característica	Medida
Gama dinámica	50 g
Sensibilidad	25.4 mm/s para 4 mA a 20 mA
Sensibilidad Transversal	Menos de 5 %
Sensibilidad a la Temperatura	0,145/°F
Rango de frecuencia	2 Hz a 1 kHz ±10 %
Linealidad de amplitud	Mejor que el 1 % de linealidad
Resonancia base montada	5kHz

**Tabla 1.5-8**

*Especificaciones Ambientales para MAV52*

Característica	Medida
----------------	--------

<b>Temperatura, funcionamiento</b>	-23 °C a 99 °C [-10 °F a 210 °F]
<b>Sellado, AG928 y AG932</b>	IP67/NEMA 4

**Tabla 1.5-9**

*Especificaciones Eléctricas para MAV52*

<b>Característica</b>	<b>Medida</b>
<b>Entrada</b>	Voltaje
<b>Salida</b>	4 mA a 20 mA
<b>Tensión de alimentación</b>	12V a 32V
<b>Rango de corriente</b>	0,5 mA a 8 mA
<b>Tensión de polarización</b>	12 Vdc
<b>Cable. AG932</b>	PTFE blindado de acero inoxidable
<b>Longitud de cable estándar</b>	4,9 m [16 pies]
<b>Ruido eléctrico</b>	0,3 mg máx.
<b>Aislamiento</b>	Base aislada
<b>Conector de acoplamiento</b>	MH008

**Tabla 1.5-10**

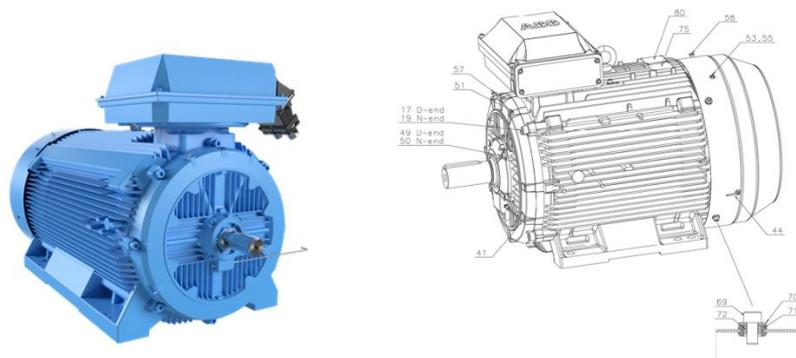
*Especificaciones Mecánicas para MAV52*

<b>Característica</b>	<b>Medida</b>
<b>Peso</b>	0,14 kg [5 onzas]
<b>Material</b>	Acero inoxidable
<b>Par de montaje</b>	6 pies-libra

**Motor de Inducción utilizado para los dos motores Hoist.** El motor M3BP 450LB 6 de ABB es un motor trifásico de jaula de ardilla con una potencia de 710 kW y eficiencia de Clase IE2, lo que indica una alta eficiencia energética. Con 6 polos y un tamaño de bastidor de 450, opera a una tensión de 400V y frecuencia de 50Hz. Este motor está diseñado para resistir condiciones adversas, con un gabinete con clasificación IP55 para protección contra polvo y agua. Su montaje es del tipo IMB3/IM1001, lo que proporciona flexibilidad en la instalación. Es una opción robusta y eficiente para aplicaciones industriales que requieren un rendimiento confiable y económico.

**Figura 1.5.22**

*Motor M3BP 450LB 6*



**Tabla 1.5-11**

*Generalidades de los dos motores Hoist*

<b>Motor</b>	<b>Tipo</b>	(ABB) M3BP450L
	<b>Potencia de Aplicación</b>	2x610Kw
	<b>Velocidad de Aplicación</b>	900/1800 rpm

**Tabla 1.5-12***Datos de placa del motor para M3BP 450LB 6*

Conn	Freq	Volt	Pot	Vel	Corr	FP	Eff	Torque	IS/IN
Y	50Hz	690V	710kW	995RPM	719A	0.850	95%	6814Nm	7.00
D	50Hz	400V	710kW	995RPM	1244A	0.850	95%	6814Nm	7.00
D	50Hz	415V	710kW	995RPM	1203A	0.850	95%	6814Nm	7.80

El principio de funcionamiento de los motores de rendimiento de procesos de bajo voltaje IEC de ABB se basa en la generación de un campo magnético rotativo en el estator, la interacción con el rotor para generar un par electromagnético, y el control de la velocidad y el par mediante la manipulación de la corriente alterna suministrada al motor.

A continuación, se detallan las características eléctricas, mecánicas y dimensionales de este motor.

**Tabla 1.5-13***Componentes Eléctricos para M3BP 450LB 6*

Característica	Medida
<b>Datos de la clase IE (50 Hz)</b>	IE Clase IE2 Carga completa (100%) 95,0 % Carga parcial (75%) 95,0 % Carga parcial (50%) 94,1 %
<b>Clase IC</b>	IC411
<b>Clase IP</b>	IP55
<b>Clase de aislamiento</b>	ICLF
<b>Sentido de giro</b>	Ambos lados
<b>Número de polos (Alto)</b>	6

<b>Temperatura ambiente</b>	40°C
-----------------------------	------

**Tabla 1.5-14**

*Especificaciones Mecánicas para M3BP 450LB 6*

<b>Característica</b>	<b>Medida</b>
<b>Material del bastidor</b>	Hierro fundido
<b>Tamaño del bastidor</b>	450
<b>Cojinete</b>	6326/C3
<b>Cojinetes NDE</b>	6322/C3
<b>Clase IM</b>	IMB3 IM1001
<b>Ubicación de caja de terminadores</b>	Arriba

**Tabla 1.5-15**

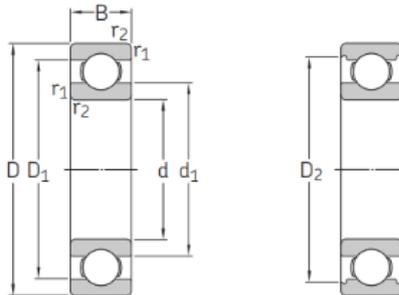
*Dimensiones para M3BP 450LB 6*

<b>Característica</b>	<b>Medida</b>
<b>Embalaje Nivel 1 Unidades</b>	1 palet (elevador)
<b>Largo</b>	2187 mm
<b>Alto</b>	1231 mm
<b>Ancho</b>	966 mm
<b>Peso</b>	4500 mm
<b>Peso bruto</b>	4540 mm

**Rodamientos.** Se utilizan rodamientos rígidos de bolas de una hilera de la marca SKF tanto para la parte delantera (eje de salida): 6326/C3, como para la parte trasera (tapa ventilador): 6322/C3.

**Figura 1.5.23**

*Dimensiones de los rodamientos*



**Tabla 1.5-16**

*Especificaciones generales del Rodamiento 6326/C3*

Característica	Medida
Número de pieza del fabricante	6326/C3
Diámetro interior	130 mm
Diámetro exterior	280 mm
Ancho	58 mm
Unidad de medida	métrico
Sellos	Abierto (Sin Sello)

**Tabla 1.5-17**

*Especificaciones técnicas del rodamiento 6326/C3*

Principales dimensiones	Clasificaciones de carga básicas	Límite de carga	Índices de velocidad	Masa [kg]

[mm]			[kN]		[kN]	[r/min]		
d	D	B	C	Co	Pu	Ref	Lim	-
130	280	58	229	216	6,3	5000	3200	15

**Tabla 1.5-18**

*Especificaciones generales del Rodamiento 6322/C3*

Característica	Medida
Número de pieza del fabricante	6322/C3
Diámetro interior	110 mm
Diámetro exterior	240 mm
Ancho	50 mm
Unidad de medida	métrico
Sellos	Abierto (Sin Sello)

**Tabla 1.5-19**

*Especificaciones técnicas del rodamiento 6326/C3*

Principales dimensiones [mm]			Clasificaciones de carga básicas [kN]		Límite de carga [kN]	Índices de velocidad [r/min]		Masa [kg]
d	D	B	C	Co	Pu	Ref	Lim	-
110	240	50	203	180	5,7	6000	3800	9,65

### 1.5.7 Inteligencia Artificial

La inteligencia se define como un conjunto de atributos que abarcan la capacidad de planificación y resolución de problemas. De manera más sencilla, se puede describir como la

habilidad para tomar decisiones acertadas en función de una serie de datos de entrada y un conjunto de acciones posibles (Jones, 2015).

El concepto de Inteligencia Artificial se ha desarrollado a través del estudio del funcionamiento del cerebro humano durante la resolución de problemas. Finalmente, se crea un sistema de software inteligente basado en los resultados obtenidos al enfrentar problemas. Para implementar esta forma de inteligencia en máquinas, se requiere la creación de un sistema que tenga la capacidad de entender, aprender, razonar y actuar de manera similar a un ser humano (Pathak & Tiwari, 2018).

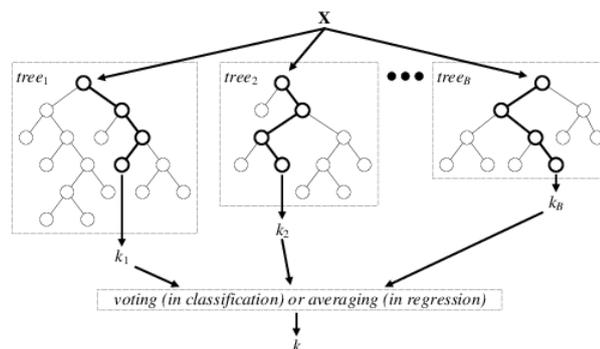
En el ámbito del Machine Learning, se emplean diversos enfoques, como el aprendizaje supervisado y el no supervisado. En el aprendizaje supervisado, el usuario puede definir cuáles respuestas son correctas o incorrectas. Por otro lado, en el aprendizaje no supervisado, no es necesario que el usuario esté presente; el sistema aprende de la información que recopila. Los algoritmos ampliamente utilizados en Machine Learning incluyen árboles de decisión y el aprendizaje basado en ejemplos similares (Jones, 2015).

Las Redes Neuronales, conocidas como ANN (Artificial Neural Network) son modelos computacionales que simulan el funcionamiento de las neuronas biológicas humanas. Esta simulación se basa en el comportamiento de las neuronas reales, aunque se trata de una abstracción debido a la complejidad de las neuronas biológicas. Las ANNs se componen de múltiples entradas, que se categorizan según la intensidad de la señal y se someten a cálculos mediante funciones matemáticas para determinar la activación de las neuronas. Estas ANNs, en esencia, son funciones matemáticas no lineales que combinan neuronas artificiales para procesar información. Una estructura típica de una ANN se muestra en la figura 2.21, donde las capas ocultas son las que aprenden a reconfigurar las entradas (Kulkarni & Joshi, 2015).

Otra opción para la implementación de inteligencia artificial son los árboles de decisión, entre ellos se tiene a RandomForestRegressor, que es un algoritmo de aprendizaje automático supervisado que se utiliza para predecir valores continuos. Es un modelo de ensamble que combina la salida de varios árboles de decisión, cada uno de los cuales se entrena en un subconjunto diferente de los datos de entrenamiento.

**Figura 1.5.24**

*Diagrama de RandomForest*



Para cada árbol, se selecciona un subconjunto aleatorio de los datos de entrenamiento y un subconjunto aleatorio de las características. Luego, el árbol se entrena en el subconjunto de datos seleccionado. Para predecir un valor, el algoritmo primero calcula la predicción de cada árbol de decisión en el conjunto. Luego, la predicción final es el promedio de las predicciones de los árboles.

### 1.5.6. Análisis Vibratorio

El análisis vibratorio, según Girdhar Paresh (2004), puede definirse como "la disciplina que estudia las vibraciones mecánicas de un sistema y su aplicación en la detección temprana de fallos en maquinaria industrial". La importancia del análisis vibratorio en el mantenimiento predictivo ha sido resaltada por Vibration Institute, que sostiene que "la capacidad de monitorear

las vibraciones de los equipos proporciona una ventana única hacia su salud operativa, permitiendo la identificación de posibles problemas antes de que se conviertan en fallas costosas" (Vibration Institute, 2019).

La importancia del análisis vibratorio en el mantenimiento predictivo radica en varios aspectos:

- **Detección temprana de problemas:** El análisis vibratorio permite detectar problemas mecánicos en una etapa temprana, mucho antes de que se vuelvan críticos. Esto proporciona tiempo suficiente para planificar y llevar a cabo actividades de mantenimiento antes de que ocurra una falla.
- **Evita paradas no planificadas:** Al identificar y abordar problemas potenciales antes de que causen una falla completa, el análisis vibratorio ayuda a evitar paradas no planificadas en la producción. Esto es especialmente crucial en entornos industriales donde el tiempo de inactividad puede resultar costoso.
- **Aumenta la vida útil de los equipos:** Al mantener los equipos en condiciones óptimas a través de un monitoreo constante, el análisis vibratorio contribuye a prolongar la vida útil de la maquinaria, reduciendo así los costos asociados con la compra y la instalación de nuevos equipos.
- **Mejora la eficiencia operativa:** Al prevenir fallas inesperadas, el análisis vibratorio contribuye a una mayor eficiencia operativa al garantizar un funcionamiento continuo y confiable de la maquinaria.
- **Reducción de costos de mantenimiento:** El mantenimiento predictivo, facilitado por el análisis vibratorio, permite programar actividades de mantenimiento en momentos óptimos, evitando gastos innecesarios y minimizando el tiempo de inactividad.

- Seguridad laboral: La identificación temprana de problemas mediante el análisis vibratorio también contribuye a mejorar la seguridad laboral al prevenir posibles accidentes asociados con fallas mecánicas.

**Sistema Resorte-Masa: Masa, Rigidez y Amortiguación.** En el ámbito de los conceptos básicos, el sistema resorte-masa se presenta como un modelo fundamental en el análisis vibratorio. Girdhar Pares (2004) destaca que "la masa, rigidez y amortiguación son parámetros clave que definen el comportamiento vibratorio de un sistema mecánico". Estos elementos, como describe Rao (2011) en su obra sobre *Vibrations of Continuous Systems*, son esenciales para comprender la dinámica de las vibraciones en maquinaria.

En el contexto de vibraciones mecánicas, el sistema resorte-masa es un modelo simplificado utilizado para comprender y analizar el comportamiento vibratorio de un sistema mecánico. Este modelo básico consta de tres componentes principales: masa ( $m$ ), rigidez del resorte ( $k$ ), y amortiguación ( $c$ ).

**Masa ( $m$ ).** La masa representa la cantidad de materia en un cuerpo y su medida en kilogramos (kg). En el contexto de vibraciones, la masa se refiere a la masa efectiva del sistema, que incluye la masa física del objeto y cualquier masa adicional que pueda ser influenciada por la vibración.

La masa influye en la frecuencia natural de vibración del sistema. Un sistema con una masa mayor tenderá a tener una frecuencia natural más baja.

**Rigidez del Resorte ( $k$ ).** La rigidez del resorte es una medida de la resistencia de un resorte a la deformación cuando se le aplica una fuerza. Se mide en newtons por metro (N/m).

La rigidez del resorte determina la fuerza restauradora que tiende a devolver el sistema a su posición de equilibrio después de ser perturbado. Una mayor rigidez del resorte resultará en una frecuencia natural más alta del sistema.

**Amortiguación (c).** La amortiguación se refiere a la capacidad del sistema para disipar la energía vibracional. Puede ser de dos tipos: amortiguación viscosa (proporcional a la velocidad) o amortiguación de Coulomb (proporcional a la velocidad y dirección opuesta a la velocidad).

La amortiguación afecta la respuesta del sistema a la excitación. Un sistema con amortiguación adecuada puede disipar la energía de manera controlada, evitando oscilaciones excesivas y posiblemente daño a la maquinaria.

**Recolección de Señales de Vibración con Transductores de Vibración.** La recolección de señales de vibración, llevada a cabo mediante transductores especializados, es una fase crucial en el análisis vibratorio. En palabras de Harris (2002), "los transductores de vibración son dispositivos sensibles que convierten las oscilaciones mecánicas en señales eléctricas para su posterior análisis". Este proceso, resaltado por Girdhar Paresh (2004), asegura una captura precisa de las características vibracionales de la maquinaria.

- Selección del Punto de Medición: Identificar los puntos críticos en la maquinaria donde se deben realizar mediciones. Estos puntos pueden incluir rodamientos, ejes, engranajes u otras áreas susceptibles a vibraciones anómalas.
- Elección del Transductor de Vibración: Seleccionar el transductor de vibración adecuado según el tipo de vibración que se está midiendo (aceleración, velocidad o desplazamiento). Los transductores comunes incluyen acelerómetros para medir

aceleración, velocímetros para medir velocidad y sensores de desplazamiento para medir desplazamiento.

- **Montaje del Transductor:** Montar el transductor de vibración en el punto de medición seleccionado. Asegurarse de que el transductor esté correctamente fijado y en contacto con la superficie de medición.
- **Calibración del Transductor:** Calibrar el transductor para garantizar mediciones precisas. La calibración debe realizarse según las especificaciones del fabricante del transductor.
- **Conexión del Transductor al Instrumento de Medición:** Conectar el transductor al instrumento de medición de vibraciones. Esto puede ser un analizador de vibraciones o un colector de datos. Asegurarse de que la conexión sea firme y esté correctamente cableada.
- **Configuración del Instrumento de Medición:** Configurar el instrumento de medición para el tipo de medición requerido (aceleración, velocidad o desplazamiento) y ajustar la escala de medición según las características de la máquina.
- **Adquisición de Datos:** Iniciar la adquisición de datos. Esto implica registrar las señales de vibración durante un período de tiempo significativo para capturar variaciones y patrones de vibración.
- **Análisis de Datos:** Analizar los datos recopilados utilizando software especializado. Identificar patrones, frecuencias dominantes, armónicos y cualquier comportamiento anómalo que pueda indicar problemas en la maquinaria.
- **Interpretación de Resultados:** Interpretar los resultados del análisis de vibraciones y compararlos con los límites establecidos por normas o recomendaciones del fabricante para determinar la condición de la maquinaria.

- **Acciones Correctivas:** Tomar acciones correctivas según sea necesario. Esto puede incluir programar mantenimiento preventivo, ajustar componentes, o realizar reparaciones para evitar problemas más graves en el futuro.

**Conversión de Vibraciones en Señal Eléctrica.** La conversión de vibraciones en señales eléctricas, como lo describe Kim (2018) en sus investigaciones, es un paso esencial en el análisis vibratorio moderno. Según Girdhar Paresh (2004), "esta conversión facilita la representación y procesamiento de las vibraciones, permitiendo un análisis más preciso". La calidad de esta conversión, influida por la tecnología de los transductores, impacta directamente en la fiabilidad de los resultados obtenidos.

**Procesamiento de las Señales en Bases de Datos para Motores con Vibraciones.** El procesamiento de señales en bases de datos, según Smith et al. (2016), es esencial para gestionar eficientemente la gran cantidad de datos generados por el análisis vibratorio. Girdhar Paresh (2004) destaca que "almacenar las señales en bases de datos permite un acceso rápido y una gestión organizada, facilitando la identificación de patrones y la realización de análisis a largo plazo". Esta estructuración optimizada mejora la capacidad de predecir y prevenir posibles fallos en los motores.

**Analizador de Vibraciones.** Un analizador de vibraciones es un dispositivo que se emplea para evaluar y examinar las oscilaciones presentes en máquinas o estructuras. Estas vibraciones pueden derivar de varios factores, como el desequilibrio en componentes rotativos, la desalineación en acoplamientos, el desgaste o daño en engranajes, el deterioro de rodamientos, fuerzas aerodinámicas o hidráulicas, y problemas eléctricos.

El análisis de vibraciones permite identificar el origen de estas oscilaciones y determinar si representan un problema. Las características fundamentales que se estudian incluyen la frecuencia, desplazamiento, velocidad, aceleración y la energía de impulsos (spike energy).

La frecuencia se define como la cantidad de ciclos completos en un período dado, y se mide en ciclos por minuto (cpm). Cada tipo de anomalía mecánica se manifiesta con una frecuencia de vibración particular. La correlación entre cpm y rpm (revoluciones por minuto) revela la causa y la parte responsable de la vibración, dado que las fuerzas varían según la velocidad de rotación. Problemas diversos se detectan mediante frecuencias que coinciden con la velocidad de rotación o sus múltiplos. La amplitud de la vibración señala la magnitud y gravedad del problema, ofreciendo una idea de la condición general de la máquina. Puede medirse en desplazamiento, velocidad o aceleración.

La velocidad de vibración considera tanto el desplazamiento como la frecuencia, por lo que es un indicador directo de la intensidad de la vibración. La severidad de la vibración se precisa aún más midiendo la velocidad, aceleración o desplazamiento, dependiendo del rango de frecuencias. Para frecuencias bajas (menores a 600 cpm), se toman medidas de desplazamiento; para frecuencias entre 600 y 60,000 cpm, se mide la velocidad; y para frecuencias altas (mayores a 60,000 cpm), se toman mediciones de aceleración. Las vibraciones que tienen lugar entre 600 y 60,000 cpm se analizan teniendo en cuenta el valor de la velocidad. La aceleración está relacionada con la fuerza que provoca la vibración, algunas de ellas se producen a altas frecuencias, aunque velocidad y desplazamiento sean pequeños.

El spike energy o energía de impulsos proporciona información importante a la hora de analizar vibraciones. Este parámetro mide los impulsos de energía de vibración de breve duración y, por lo tanto, de alta frecuencia. Pueden ser impulsos debidos a: Defectos en la superficie de elementos de rodamientos o engranajes. Rozamiento, impacto, 11 contacto entre metal-metal en

máquinas rotativas. Fugas de vapor o de aire a alta presión. Cavitación debida a turbulencia en fluidos.

**Transformada de Fourier FFT para Análisis de Vibración.** Dentro de las pruebas de almacenamiento, la Transformada de Fourier FFT emerge como una herramienta esencial, según González (2015) en sus estudios sobre procesamiento de señales. Girdhar Paresch (2004) destaca que "la FFT descompone las señales complejas en componentes de frecuencia, proporcionando una visión detallada de los modos vibratorios presentes en la maquinaria". Esta técnica matemática avanzada facilita la identificación y el análisis de patrones que podrían indicar posibles problemas en los motores.

Se explora Node RED Contribuit Machine Learning como estrategia específica de entrenamiento para el mantenimiento predictivo de motores. Girdhar Paresch (2004) destaca que "la integración de herramientas de aprendizaje automático en plataformas como Node RED mejora la capacidad de analizar datos de vibración de manera eficiente, permitiendo una identificación más rápida y precisa de posibles fallos en los motores".

**Los Armónicos. Límites y Normas de la Vibración.** La presencia de armónicos en las vibraciones, como señala Brüel & Kjør (2017) en sus investigaciones, puede ser indicativa de problemas mecánicos. Girdhar Paresch (2004) afirma que "la identificación y análisis de armónicos son esenciales para la detección temprana de desalineaciones, desequilibrios y otros fallos". Además, los límites y normas de la vibración, según ISO 10816-1 (International Organization for Standardization, 2019), proporcionan criterios objetivos para evaluar la condición de la maquinaria.

Para evaluar y gestionar las vibraciones en maquinaria, se han establecido límites y normas que definen los niveles aceptables de vibración en diferentes frecuencias. Estas normas son fundamentales en el campo del mantenimiento predictivo y la monitorización de la condición de equipos.

**ISO 10816-1.** Esta norma internacional establece criterios para la evaluación de vibraciones en máquinas industriales. Proporciona límites específicos para diferentes tipos de maquinaria, clasificados según su tamaño y función en términos de aceleración de vibración efectiva. Sin embargo, es importante considerar que esta norma no es exclusiva para motores de inducción y puede requerir interpretación adicional o consideración de otras normas específicas del fabricante para una evaluación más precisa.

Para los motores de inducción, las ubicaciones recomendadas para la medición de la vibración son: Soportes del motor, cojinetes de apoyo y rotor. Los equipos utilizados para medir la vibración deben estar calibrados para garantizar la precisión de los resultados. Proporciona orientación sobre los métodos de análisis de vibraciones que se pueden utilizar para motores de inducción como el análisis de la frecuencia, análisis de la amplitud y análisis del espectro.

**API 670:** Emitida por el American Petroleum Institute (API), esta norma se centra en la monitorización de vibraciones en máquinas rotativas críticas utilizadas en la industria del petróleo y gas. La norma establece directrices para las mediciones de vibración, incluyendo la ubicación de los sensores, los métodos de medición y los límites aceptables de vibración. Estos límites pueden ser aplicables al monitoreo de motores de inducción. Define los niveles de alarma y acciones correctivas basadas en los datos de vibración recopilados.

**VDI 2056:** Desarrollada por la Asociación Alemana de Ingenieros (VDI), esta norma proporciona orientación sobre la evaluación de la vibración en máquinas y plantas industriales, pero no está dedicada exclusivamente a motores de inducción. Es posible que sea necesario combinar las pautas generales de la norma con estándares específicos de la industria o recomendaciones del fabricante para realizar un análisis más completo y preciso de las vibraciones en motores de inducción. La norma puede ofrecer pautas para implementar estrategias de mantenimiento predictivo basadas en análisis de vibraciones, lo que puede ser útil para mantener motores de inducción en condiciones óptimas de funcionamiento.

Muchos fabricantes de equipos también proporcionan límites específicos de vibración para sus productos. Estos límites a menudo se basan en la experiencia y el rendimiento esperado de sus máquinas.

Estas normas proporcionan límites específicos para la evaluación de vibraciones los cuales se basan en las variables frecuencia y amplitud. Para la frecuencia, las vibraciones de los motores de inducción se pueden clasificar en dos categorías principales: vibraciones de baja frecuencia (< 100 Hz) y vibraciones de alta frecuencia (> 100 Hz). La amplitud de la vibración es la magnitud de la vibración. Se mide en unidades de mm/s o  $\mu\text{m/s}$ .

La siguiente tabla proporciona un resumen de los límites de vibración establecidos por estas normas para motores de inducción:

**Tabla 1.5-20**

*Normas para vibraciones en máquinas reacondicionadas.*

Norma	Frecuencia	Amplitud [mm/s]
ISO 10816-1	Baja	0.7
	Alta	0.2

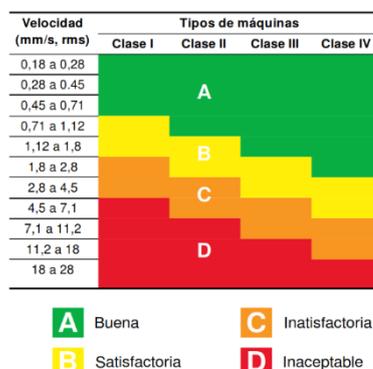
API 670	Baja	0.8
	Alta	0.3
VDI 2056	Baja	0.6
	Alta	0.2

Tabla 1.

En la Tabla 1 se hace referencia a los estándares básicos de cada norma para estructuras y maquinas sometidas a vibraciones. Sin embargo, para este estudio se coge la norma ISO 10816 y se expande por clase de Motor que genera las rotaciones, como se ha mostrado en la descripción del Motor en el Hoist se tiene una máquina de 610Kw por lo que es de clase IV. Entonces, se hace uso de la siguiente tabla.

**Figura 1.5.25**

*Niveles de vibracion segun clase de Motor y su estado*



**Procedimiento de medición.** Antes de iniciar las mediciones, es crucial garantizar que el dispositivo de medición pueda proporcionar el valor RMS de la velocidad vibratoria dentro del rango de frecuencias de 10 a 1000 Hz. Además, se debe verificar la capacidad del instrumento y el sensor de vibraciones para tolerar las condiciones ambientales, tales como temperatura, campos

magnéticos, longitud del cable y la orientación del sensor. Es esencial asegurarse de que, al montar el sensor, su presencia no perturbe las vibraciones naturales de la máquina.

**Puntos de medición.** Hay al menos un punto en la máquina (descansos) donde es importante conocer si existe una vibración significativa. Otro punto importante es en los pies de la máquina, es decir, en los puntos de unión a la fundación. Las mediciones deben ser realizadas sobre cada descanso principal de la máquina, en las direcciones radiales (vertical y horizontal) y en la dirección axial. Se debe asegurar que las medidas representen la vibración de la caja de los descansos y no incluyan una resonancia local.

**Condiciones de operación.** Las mediciones deben realizarse cuando el rotor y los descansos principales han alcanzado sus temperaturas estacionarias de trabajo y con la máquina funcionando bajo condiciones nominales, como voltaje, flujo, presión, carga. En máquinas con velocidad variable, las mediciones deberían realizarse en las condiciones extremas. Los valores medidos máximos serán considerados representativos de la severidad de la vibración

**Identificación de causas de vibraciones.** El diagnóstico de fallos se erige como un componente crítico. Según Girdhar Paresh (2004), "identificar problemas comunes, como desalineación, desequilibrio y desgaste prematuro de componentes, es crucial para prevenir fallas catastróficas y planificar intervenciones de mantenimiento de manera proactiva". La detección temprana y el entendimiento profundo de la naturaleza de los fallos son esenciales para implementar soluciones efectivas.

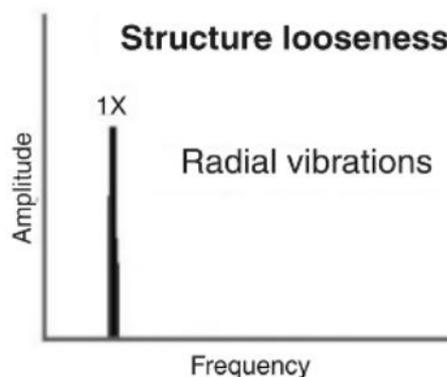
Una vez obtenidos de una forma metódica y precisa los datos de vibraciones de una máquina donde se ha detectado un problema, es necesario identificar cual ha sido su causa y así buscar la forma y momento de reparación más eficiente, es decir, que elimine el fallo y su coste económico sea el mínimo posible.

**Desequilibrio.** Esta es una de las causas más probable de que exista vibración en las máquinas, en casi todos los elementos es fácil encontrar un pico en el gráfico de amplitud frente a frecuencia, que denote un pequeño desequilibrio.

Para conocer la cantidad de desequilibrio hay que encontrar la amplitud de la vibración en la frecuencia igual a 1 x rpm. La amplitud es proporcional a la cantidad de desequilibrio.

**Figura 1.5.26**

*Dimensiones de los rodamientos*

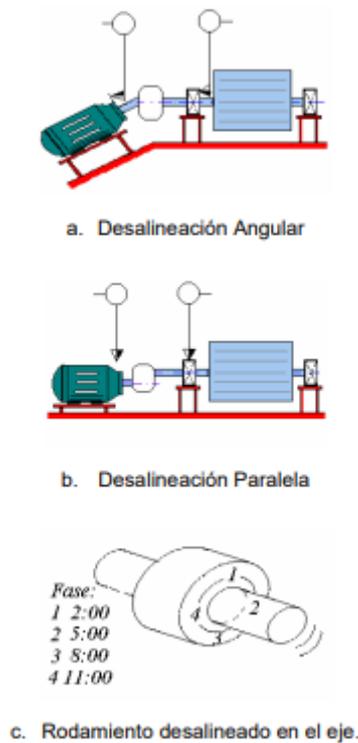


Se debe recalcar que, si en un sistema tiene vibraciones nominales, es decir, que durante su operación normal presentar vibraciones, para poder ser separadas del real desequilibrio mecánico se debe asilar la porción de la amplitud mayor a la usual.

**Desalineación.** Es un problema muy común debido a la dificultad que supone alinear dos ejes y sus rodamientos de forma que no se originen fuerzas que produzcan vibraciones. La forma de vibración de un eje torcido es similar a la de una mala alineación angular.

**Figura 1.5.27**

*Desalineaciones*

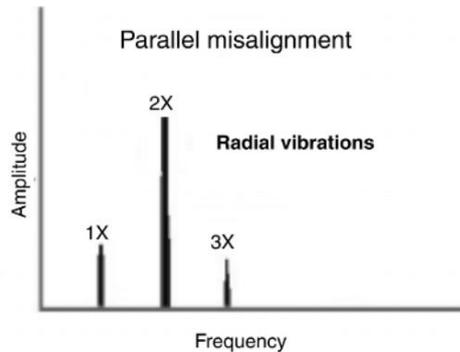


Este fallo puede presentar alta vibración en sentido axial además de radial. Así siempre que exista una alta vibración en axial y radial, y si la axial es mayor que la mitad de la radial puede existir un problema de desalineación o ejes torcidos. Existen tres tipos básicos de desalineación, en paralelo, angular y rodamiento desalineado en el eje.

La falta de alineación en paralelo produce sobre todo vibración en dirección radial con frecuencia igual al doble de la velocidad de giro del eje. La falta de alineación angular da vibración en dirección axial en los dos ejes a una frecuencia igual a 1 x rpm

**Figura 1.5.28**

*Espectro de frecuencia de desalineación*



Los acelerómetros axiales miden la aceleración a lo largo del eje del motor. Los acelerómetros radiales miden la aceleración en un plano perpendicular al eje del motor.

La aceleración axial se puede utilizar para determinar el desequilibrio del motor. El desequilibrio se produce cuando el peso de los componentes del motor no está uniformemente distribuido. El desequilibrio puede causar vibraciones y desgaste prematuro de los componentes del motor.

La aceleración radial se puede utilizar para determinar el desgaste de los rodamientos. El desgaste de los rodamientos puede causar vibraciones y ruido.

**Engranajes.** Desalineación Es un problema muy común debido a la dificultad que supone alinear dos ejes y sus rodamientos de forma que no se originen fuerzas que produzcan vibraciones. La forma de vibración de un eje torcido es similar a la de una mala alineación angular.

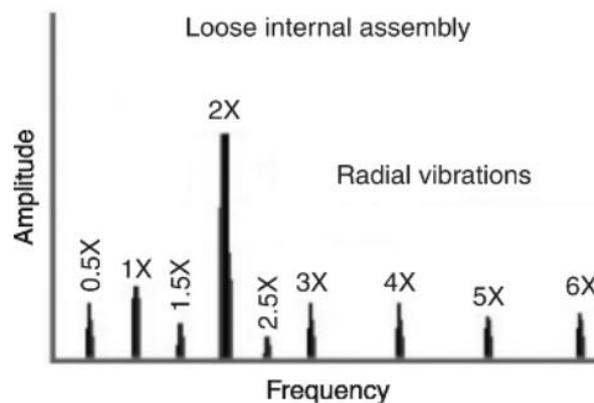
Los problemas de engrane que dan esta vibración son: desgaste excesivo de los dientes, inexactitud de los dientes, fallos de lubricación, elementos extraños entre dientes. Las vibraciones causadas por defectos de engranajes pueden ser detectadas en varios puntos de las máquinas

**Rodamientos.** Fallos en elementos del rodamiento dan vibración a unas frecuencias altas no relacionadas con la velocidad de rotación y de amplitud también aleatoria.

La frecuencia a la que se produce la máxima amplitud puede dar una idea del elemento defectuoso del rodamiento. Los defectos en elementos rodantes, pistas de rodamiento o jaula de retención generan fuerzas que se transmiten al alojamiento y estructura que les rodea.

**Figura 1.5.29**

*Espectro de deterioro interno*



### 1.5.7. Teoremas y ecuaciones para Análisis vibratorio

**Teorema de Nyquist.** Si una señal continua,  $S(t)$ , tiene una banda de frecuencia tal que  $F_n$  sea la mayor frecuencia comprendida dentro de dicha banda, dicha señal podrá reconstruirse sin distorsión a partir de muestras de la señal tomadas a una frecuencia  $F_T$  siendo:

$$F_T > 2 * F_n$$

**Frecuencias de Rodamientos.** La empresa KissSoft manufactura rodamientos y provee softwares para analizar las características mecánicas de las estructuras rodantes, en el 2007 hizo un estudio donde comparo su software con las fórmulas definidas para definir las vibraciones

fundamentales (FTF), de paso bajo y alto de las bolas (BPFO y BPFI), y de frecuencias de deterioro de superficies de elementos rodantes (BSF). Las fórmulas son las siguientes:

$$D_{pw} = D + d - 2B;$$

$$D_{we} = B;$$

$$BPFO = \frac{n}{60} * z * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (1)$$

$$BPFI = \frac{n}{60} * z * \frac{D_{pw} + D_{we} \cos(\alpha)}{2D_{pw}} \quad (2)$$

$$BSF = \frac{n}{60} * \frac{D_{pw}^2 - D_{we}^2 \cos(\alpha)}{D_{we} * D_{pw}} \quad (3)$$

$$BPF = 2 * BSF \quad (4)$$

$$FTF = \frac{n}{60} * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (5)$$

*d = diametro interior de rodamiento en mm*

*D = diametro exterior de rodamiento en mm*

*D<sub>pw</sub> = diametro de encaje de jaula en mm*

*D<sub>we</sub> = B = ancho de rodamiento en mm*

*z = numero de elementos rodantes*

*n = velocidad de rotacion en minutos en RPM*

## **1.6. Trabajos anteriores**

### **1.6.1. “Determinación del estado de falla de motores eléctricos a partir del análisis de ruidos utilizando técnicas de Aprendizaje Profundo” Dávalos y Vásquez (2022).**

El enfoque presentado en este estudio se dirige a la detección temprana de problemas en motores eléctricos mediante la identificación de patrones en señales de audio y vibraciones utilizando modelos avanzados de Aprendizaje Profundo.

Según Dávalos y Vásquez (2022), se creó un conjunto de datos a partir de grabaciones de sonidos y vibraciones de motores en condiciones normales y anómalas procedentes de diversas instalaciones industriales y talleres mecánicos en Guayaquil. La recolección de estos datos se llevó a cabo mediante una aplicación móvil desarrollada específicamente para este propósito. Esta aplicación recopila datos de audio en formato WAV y datos de vibraciones en formato CSV, siendo necesario colocar el teléfono en el motor con la ayuda de un imán para realizar la recolección.

Se obtuvieron grabaciones de 13 motores diferentes, de los cuales 11 estaban en condiciones normales y 2 presentaban anomalías. Se realizaron 21 grabaciones de motores en condiciones normales y 4 de motores anómalos, todas con una duración promedio de 20 segundos.

Se diseñó un módulo de preprocesamiento para transformar las señales en imágenes y aumentar la cantidad de ejemplos de la clase minoritaria. Posteriormente, se desarrolló un módulo de clasificación de imágenes en el que se llevaron a cabo diversos experimentos utilizando arquitecturas como VGG19 y ResNet50. Estas redes fueron entrenadas específicamente para detectar irregularidades en los sonidos de motores, además de implementar una arquitectura tipo ensemble y una red neuronal completamente conectada.

Los resultados de la clasificación demostraron que el uso conjunto de datos multimodales posibilita un diagnóstico eficaz de fallas en motores eléctricos. Se alcanzó una precisión del 93.02% al emplear dos modelos de redes neuronales convolucionales (uno para cada tipo de señal) con una arquitectura tipo ensemble. Por consiguiente, el modelo propuesto se muestra eficiente y preciso para diagnosticar el estado de falla en un motor eléctrico, sin la necesidad de sensores dedicados a cada motor, personal especializado o interrupciones en la producción industrial.

### **1.6.2. “Análisis y detección de fallas en motores eléctricos aplicando algoritmos de inteligencia artificial” Odar Chero (2023).**

El objetivo principal del estudio ha sido anticipar la falla en motores eléctricos, empleando técnicas de inteligencia artificial, particularmente la Red Neuronal de Memoria a Largo Plazo Atencional Multivariante (MA-LSTM). La base de datos consiste en 273 equipos decantadores centrífugos, utilizados para la deshidratación de sólidos en el tratamiento de agua de una empresa pesquera. Aunque estos equipos están instalados en las mismas condiciones y tienen la misma capacidad, varían en la cantidad de datos acumulados debido a su ubicación en diferentes plantas.

Para la recolección de datos se utilizó el sistema WEG Motor Scan, que incluye un sensor, una app móvil y la plataforma web WEG IoT. El sensor, no invasivo y con una vida útil de 3 años gracias a una batería de litio, se instala en el motor y recoge información como vibración, temperatura y tiempo de operación. Además, estima la carga si se trata de un motor eléctrico. La comunicación entre el sensor y el dispositivo móvil se realiza mediante Bluetooth Low Energy, permitiendo una transmisión inalámbrica de datos. Estos datos se envían a la nube y se procesan a través de la plataforma web WEG IoT, donde se pueden descargar en distintos formatos para su análisis posterior. En el desarrollo del modelo, se optó por una partición de datos 80:20 debido a ser un modelo de predicción nuevo y con cantidades limitadas de información.

Tras determinar la configuración óptima del modelo, se llevó a cabo una evaluación exhaustiva de su rendimiento. Se evidenció la eficacia del MA-LSTM en entornos industriales reales, con métricas como una precisión del 98.3% y un AUC de 0.9836. La validación del modelo con datos no utilizados previamente confirmó su solidez y capacidad de generalización, destacando un área bajo la curva ROC de 1 y un F1-score de 0.9565, lo que muestra su precisión y equilibrio entre sensibilidad y especificidad.

Para Odar Chero (2023), los resultados del estudio representan un avance significativo en la detección temprana de fallos en motores eléctricos en entornos con procesos no continuos. De 247 predicciones positivas, solo 4 resultaron ser falsos positivos, y no se identificó ningún falso negativo de 9. Estos datos refuerzan la eficacia y confiabilidad del modelo MA-LSTM.

Sin embargo, se observó que el modelo LSTM mostró un rendimiento inferior cuando se enfrentó a datos de secuencias variables, lo que resalta sus limitaciones en este tipo de desafíos. Esto sugiere que la elección precisa de modelos de inteligencia artificial influye directamente en la eficacia del monitoreo predictivo. El MA-LSTM superó significativamente al LSTM tradicional en la detección de fallos, incluso en pruebas con datos nuevos, con una ventaja de hasta un 44% en precisión.

### **1.6.3. “Mantenimiento Predictivo en transformadores de potencia y de distribución eléctrica” Jado Moreno (2022).**

Este proyecto presentó una herramienta de cálculo diseñada para configurar un sistema de mantenimiento predictivo específicamente para transformadores de potencia, determinando los elementos necesarios según la criticidad de la máquina. Esta herramienta está pensada para ser utilizada por personas no especializadas en el campo.

La recopilación de los principios científicos aplicados a los transformadores se enfocó en comprender las implicaciones del funcionamiento de estos dispositivos, así como los tipos de fallos a los que pueden estar expuestos. Se abordó el estudio del origen y los principios de detección con el propósito de definir la ubicación, cantidad y tipos de sensores a incluir. Se consideraron los desafíos derivados de la integración de energías renovables descentralizadas y su impacto en los transformadores.

Jado Moreno (2022) proporcionó una explicación de los conceptos de predicción, comunicación e inteligencia artificial para comprender el funcionamiento tanto de una red de sensores como de un sistema de mantenimiento predictivo.

Además, se llevó a cabo una evaluación del mercado objetivo y se analizó la regulación legal aplicable. Los apartados finales detallaron las especificaciones de los componentes que podrían formar parte de la red de sensores, el funcionamiento de la calculadora y los criterios de ejecución. Se presentó tanto un manual de uso como la formulación matemática para calcular el período de retorno.

#### **1.6.4. “Sistema de mantenimiento predictivo para máquinas eléctricas en procesos industriales en el escenario de la industria 4.0” Rodríguez Rivera (2022).**

En este proyecto de titulación, se tuvo como objetivo clasificar las fallas más comunes que afectan una bomba centrífuga en un sistema de ósmosis inversa. Esto se logró a través del desarrollo de un modelo de mantenimiento predictivo basado en el análisis de las operaciones del equipo, permitiendo así la detección temprana de anomalías y patrones de fallas. El propósito final fue anticiparse a posibles errores para maximizar el tiempo de operación y reducir la necesidad de mantenimiento correctivo no planificado.

Rodríguez Rivera (2022) dividió el proceso de desarrollo del sistema en cinco etapas: adquisición de datos, procesamiento, identificación de indicadores de condición, entrenamiento del modelo y finalmente su implementación e integración. En la primera etapa se emplearon protocolos de comunicación industrial como Modbus TCP, Profinet, S7 y el protocolo MQTT para la conectividad IoT. Se determinó que la corriente consumida por la bomba era el parámetro más significativo para evaluar las fallas del sistema, debido a su variabilidad en comparación con otros como voltaje de salida, torque, temperatura y velocidad.

En las etapas dos, tres y cuatro se desarrolló el modelo matemático Medium Gaussian SVM para la identificación de fallas en la bomba centrífuga dentro del sistema de ósmosis inversa. Se utilizaron herramientas como DiagnosticFeatureDesigner y Classification Learner en el entorno de Matlab para crear algoritmos de mantenimiento predictivo. Se analizaron las distintas curvas de corriente generadas durante el funcionamiento de la bomba, las cuales mostraron variaciones según si el equipo se encontraba en condiciones normales o presentaba alguna de las fallas previamente identificadas.

La etapa final incluyó la implementación e integración del modelo, presentando una interfaz gráfica de usuario basada en una plataforma web que permitía visualizar en tiempo real los parámetros del proceso, las curvas de las variables y los resultados obtenidos del sistema de predicción aplicado a los datos. Se observó una mejora significativa en el tiempo promedio de detección de fallas: antes de la implementación del sistema era de 24 minutos, mientras que con el sistema implementado se redujo a 8 minutos en promedio, lo que implica una disminución del tiempo de detección de fallos en un tercio.

Para futuras investigaciones, se recomendó la incorporación de dispositivos que permitan obtener datos de vibración y espectro térmico, utilizados para definir la vida útil restante (RUL) del equipo. Además, se propuso la generación de nuevos códigos de falla en bombas, como fallas

por eje roto, vibración en el motor, desalineamiento del eje del rotor, fallas de rodamiento, entre otras.

## Capítulo 2

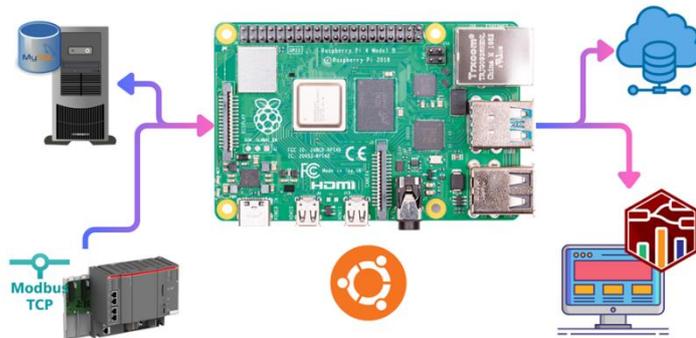
### 2.1. Metodología.

Las redes diseñadas para mantenimiento predictivo siguen el esquema de tener un medio para extraer la información del motor y estado estructural de rodamientos y demás, esto por sensores, hacia un ambiente o plataforma que envíe los datos hacia un almacenamiento en una nube y a otro equipo con algún software que contenga al modelo de predicción y presente los resultados. Con equipos más modernos se opta por el uso de envío de datos por OPC UA, sin embargo, en casos de uso de tecnologías antiguas o no especializadas para servidor OPC, se usan protocolos de comunicación serial como Modbus RTU y un Gateway para enviar datos a la Internet o su variante Modbus TCP/IP en redes LAN que embebe el envío de datos.

Sea por medio de servidores remotos o protocolo MQTT embebidos en los equipos de control PLC o conectándose con dispositivos adicionales como Raspberry o Arduino Node MCU, los proyectos tienden a usar la plataforma de Node RED para recibir los datos y transmitirlos a bases de datos y a otras plataformas e instancias en la internet. Lo que prosigue es el almacenamiento de datos donde las opciones recurrentes son bases de datos SQL o sus variantes locales (DBMaria, PostgreSQL, MySQL) y remotas (Google SQL Cloude, AWS DynamoDB, Microsoft SQL Azure), que siguen un formato muy similar y se diferencian mayormente por sus niveles de seguridad, sus políticas de acceso, sus requerimientos y el contacto con otros servicios, sobretodo en el caso de las bases de datos en la nube que tienen herramientas para Machine Learning y Aplicaciones Web.

**Figura 2.1.1**

*Ilustración de red de Mantenimiento Predictivo*



En la Figura 2.1.1 se muestra el esquema del proyecto que se compone de equipos, protocolos, sistemas OS y códigos involucrados, se asocian de la siguiente manera:

- PLC (AC500 PM595 o S71516 3PN/DP) y Servidor Modbus TCP: La información de datos de operación del motor (voltaje y corriente) es obtenida de los parámetros del equipo VFD, las lecturas de los acelerómetros y demás sensores son recogidas por los módulos de entradas y salidas remotas, y los datos pasan por la red del PLC principal. El protocolo Modbus TCP define al PLC como un servidor de datos con mapeo Modbus Serial, permitiendo definir las variables requeridas como parte del registro de sostenimiento, el cual puede ser revisado por cualquier dispositivo autorizado que cumpla la función de Cliente Modbus.
- Servidor MySQL Local: Para almacenar la nueva base de datos del Crane Monitoring System, considerando la cantidad de gigabytes que se necesiten para tener el histórico de los próximos meses, información valiosa para futuros análisis y monitoreo, el proyecto considera el caso de un ordenador con un disco duro específico para esta data. Esto definido con las consideraciones con el cliente y la red de datos actual donde quisieran incluir al sistema de mantenimiento predictivo.

- Servidor Google MySQL: Para almacenar el histórico de los reportes y alertas del sistema predictivo se propuso y aceptado la idea de usar un servidor en la nube, de donde el cual otras aplicaciones puedan acceder y presentar los datos con otros servicios a más de la red local. Considerando un máximo posible de costos de almacenamiento mensual.
- Cliente Dashboard Local: El ordenador, laptop o servidor perteneciente a la red de oficina donde el usuario tendrá acceso al panel con las secciones de Mediciones de Vibración, Análisis de Fallas y Datos de motores. Estos equipos tendrán una comunicación bidireccional con el Servidor NodeRED.
- Servidor NodeRED: La parte central del análisis vibratorio y las solicitudes de lectura y escritura de las bases de datos usando la programación en Javascript con NodeRED como editor, cuya instancia esta sobre un sistema operativo Ubuntu Linux 22.04 LTS instalado en una placa Raspberry Pi 4 Model B 8GB RAM.

El plano de redes que permite la comunicación entre los dispositivos incluyendo la Raspnberry implementado en el Laboratorio de Sistemas Complejos y esquematizada para ser integrado en caso de implementarse en la grúa STS, es decir, la red de ABB, se puede revisar en el Apéndice A.

## **2.2. Selección de Elementos**

### ***2.2.1. Modbus TCP en lugar de OPC Server: Simplicidad y Compatibilidad***

Modbus TCP y OPC Server son protocolos de comunicación industrial evaluados para un proyecto específico, y la preferencia por Modbus TCP se justifica por su implementación más simple y eficiente. Al operar directamente sobre una red TCP/IP estándar, no requiere software ni hardware adicional, siendo especialmente eficaz en aplicaciones de monitoreo de vibraciones debido a su protocolo de comunicación de solicitud/respuesta, transmitiendo exclusivamente los datos necesarios (Shah y Jain, 2019).

La integración con equipos de monitoreo de vibraciones diseñados para Modbus TCP es más sencilla, reduciendo la necesidad de conversiones o adaptadores adicionales, especialmente con marcas como Siemens y ABB, donde la adaptabilidad de Modbus TCP es crucial. A pesar de sus limitaciones, como la capacidad para manejar grandes volúmenes de datos y la falta de características avanzadas de seguridad presentes en OPC, Modbus TCP se destaca por su eficiencia y facilidad de implementación, siendo confiable y probado en entornos industriales durante muchos años, con bajo costo (Gupta y Singh, 2001).

En comparación, aunque OPC Server ofrece mayor flexibilidad y compatibilidad, su complejidad y costos adicionales en recursos pueden no ser necesarios para todas las aplicaciones de monitoreo de vibraciones, según Barnes (2002). Modbus TCP emerge como una elección acertada en proyectos industriales donde la simplicidad y el costo son factores críticos, como en el análisis de vibraciones, más cuando es necesario que no haya pérdida de datos con la persistencia moderada que tiene Modbus para seguir enviando los paquetes de datos hasta un recibimiento exitoso, como se aprecia en la Tabla 2.2-1. Además, en la red propuesta, la recolección de datos de PLC sería hacia otro elemento (Instancia del Servidor

Node-RED) dentro de la planta (gura STS) y la transmisión de datos hacia la subred de la empresa seguiría siendo por Ethernet, el cual cumple con las capas de protección y seguridad.

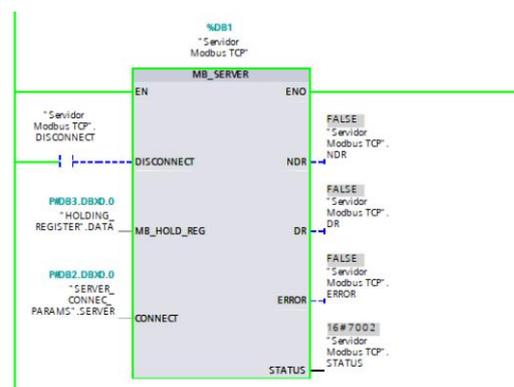
**Tabla 2.2-1**

*Tabla de desempeño y rendimiento de protocolos*

Protocolo	Tiempo para conectarse	Persistencia	Sobreescritura simultanea	Compresión de datos
OPC-UA	Pobre	Pobre	No soportado	Moderado
Modbus	Excelente	Moderado	No soportado	Pobre
HTTP	Excelente	Pobre	Definido por usuario	Definido por usuario
MQTT	Excelente	Buena	Definido por usuario	Definido por usuario
Sparkplug	Buena	Buena	Soportado	Excelente

**Figura 2.2.1**

*Servidor Modbus del PLC S7 1500*



La Figura 2.2.1 muestra al bloque de servidor Modbus de la librería de comunicación Modbus TCP de la plataforma de programación de SIEMENS TIA Portal, este bloque permitió asociar a la base de datos HOLDING REGISTER con direcciones accesibles desde un cliente Modbus remoto, en este caso el servidor Modbus de Node RED. En el apéndice B se tiene la

descripción del código cargado en el PLC s7 1500 para crear la instancia del servidor Modbus con sus bases de datos, parámetros y bloques de la librería SINA SPEED del variador G120 para la lectura de parámetros.

### ***2.2.2. Nodered en lugar de Matlab: Sistema de Análisis Compacto***

Node-RED se destaca por su interfaz gráfica basada en flujos, simplificando la creación de flujos de datos y la integración de dispositivos mediante diagramas de bloques. Su enfoque modular y escalable facilita la adición de funcionalidades y adaptación a cambios en el proyecto. En contraste, MATLAB, aunque poderoso en análisis numérico, tiene una curva de aprendizaje pronunciada y requiere habilidades de programación avanzadas, a veces siendo menos flexible en la integración con hardware y servicios externos.

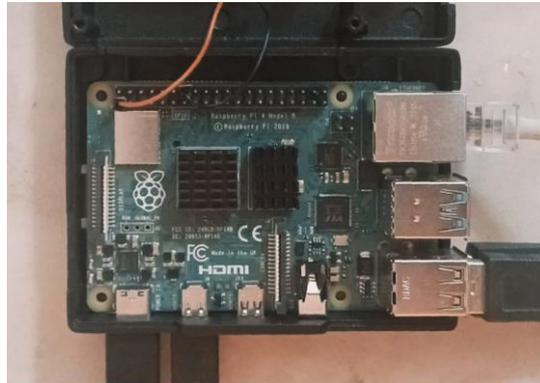
Para diseños y desarrollos ágiles, Node-RED se destaca con su desarrollo visual, siendo de código abierto y gratuito. Permite una integración rápida entre nodos y servicios, agilizando la creación de prototipos. MATLAB, a pesar de su potencia analítica, puede implicar más tiempo de implementación y tiene costos significativos en proyectos más grandes.

En un proyecto de análisis de vibraciones, donde la integración con diversos dispositivos es crucial y se busca una solución ágil con recursos limitados, Node-RED emerge como una opción atractiva en comparación con MATLAB. Su capacidad para analizar datos de vibración, realizar análisis estadísticos, y generar alarmas y notificaciones lo convierte en una elección eficiente y accesible en entornos donde la rapidez y la flexibilidad son fundamentales.

La instancia de Node RED para este proyecto pudo instalarse y programarse en diversos equipos con sistemas operativos basados en Windows, Mac OS y Linux, este ultimo escogido para un aprovechamiento de recursos, y montado en una placa Raspberry Pi 4 model B como se aprecia en la Figura 2.2.1.

### **Figura 2.2.2**

*Instancia de Node-RED en Placa Raspberry.*



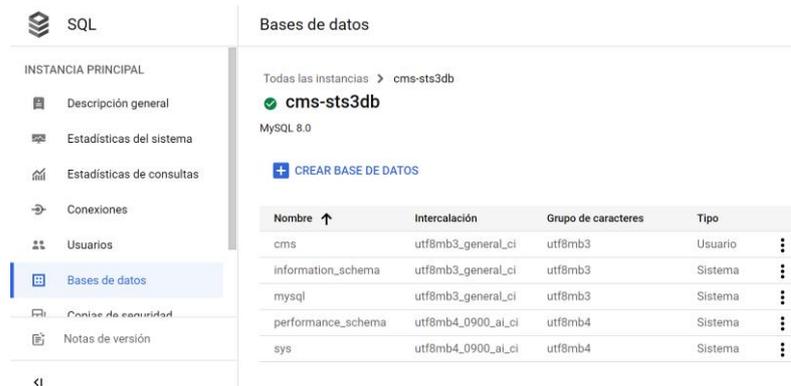
### **2.2.3. Google Cloud como una nube predilecta para almacenamiento y análisis de datos**

Google Cloud proporciona herramientas con alto nivel de seguridad, certificaciones como ISO 27001, SOC 2, y PCI DSS, junto con una garantía de disponibilidad del 99,9%, lo hacen adecuado para el análisis de datos críticos como los diagnósticos de las vibraciones de motores de inducción que son data de sumo interés.

La integración de servicios de colaboración y productividad en GCP permite la colaboración en tiempo real entre equipos geográficamente dispersos. Sin embargo, la elección entre plataformas en la nube depende de los requisitos específicos del proyecto, las habilidades del equipo y la compatibilidad con las tecnologías existentes. Alternativas como AWS y Azure también ofrecen servicios competitivos, y la elección se basa en consideraciones de costos, compatibilidad y versatilidad, haciendo de GCP una opción preferida para este proyecto.

**Figura 2.2.3**

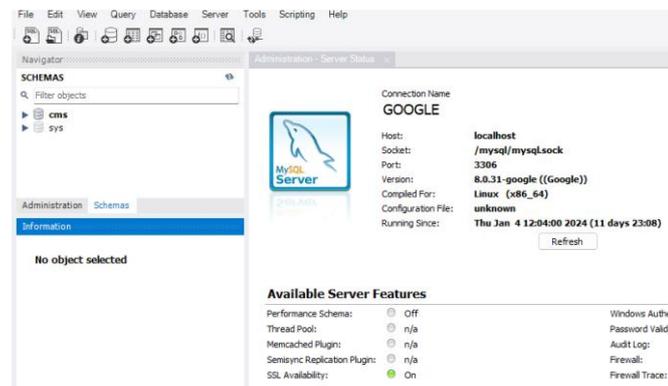
*Consola de Google Cloud con instancia SQL.*



En la Figura 2.2.2 se ve a la instancia SQL creada en la plataforma de la nube de Google, con 0.617 GB de RAM y 10 GB de ROM, con una base de datos añadida para la escritura de los reportes de los análisis del proyecto.

**Figura 2.2.4**

*Instancia de Google abierta desde MySQL Workbench*



En la Figura 2.2.3 se tiene una captura del estado de la conexión SQL creada con el software Banco de Trabajo MySQL con acceso a la base de datos de Google usando su dirección IP pública.

#### 2.2.4. Consideraciones técnicas del proyecto

Para la elección del protocolo de comunicación y del equipo que procesara las tareas de Node RED, se necesitó definir las necesidades de frecuencia de muestreo, velocidad de transmisión de datos e imprecisión del sensor de vibración.

Para el caso de la frecuencia de muestreo se hizo uso de la frecuencia fundamental calculada por las fórmulas usadas en el estudio de KissSoft.

**6326/C3**

$$D = 280 [mm];$$

$$d = 130 [mm];$$

$$B = 58 [mm];$$

$$D_{pw} = D + d - B = 352 [mm];$$

$$D_{we} = B = 58 [mm];$$

$$n = 900 [RPM];$$

$$z = 16;$$

$$\alpha = 30 [grados]$$

*FTF*

$$= \frac{n}{60} * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (1)$$

$$FTF = \frac{900}{60} * \frac{352 - 58 \cos(30)}{2(352)} = 6.425 [Hz]$$

### 6322/C3

$$D = 240 [mm];$$

$$d = 110 [mm];$$

$$B = 50 [mm];$$

$$D_{pw} = D + d - B = 300 [mm];$$

$$D_{we} = B = 50 [mm];$$

$$n = 900 [RPM];$$

$$z = 16;$$

$$\alpha = 30 [grados]$$

*FTF*

$$= \frac{n}{60} * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (1)$$

$$FTF = \frac{900}{60} * \frac{300 - 50 \cos(30)}{2(300)} = 6.41 [Hz]$$

En base a los resultados se tiene una frecuencia fundamental ubicada cerca de los 6.4Hz en ambos rodamientos, esto porque la frecuencia fundamental es principalmente la relación con la frecuencia de revolución por segundo y no considera al número de elementos rodantes. Aplicando el teorema de muestro se tuvo que se necesita una frecuencia de muestreo mayor a 12.8Hz para la respuesta fundamental y mayor a 25.6 para la respuesta del segundo armónico, el cual es el más prominente cuando se origina desperfectos internos en el rodamiento según lo

revisado en el capítulo 1.X.X.X., por lo que, se especificó una frecuencia de 35Hz para mayor precisión.

Como el protocolo Modbus TCP en el servidor de Node RED puede operar hasta 1000Hz, por lo que sí es capaz de trabajar a la frecuencia de muestreo. Pero también se necesitó revisar si cumple con las velocidades requeridas para la transferencia de información desde el servidor Modbus del PLC al servidor de Node RED, el cual está limitado a unos 10 o 100 Mbps, a diferencia de su versión RTU que está limitado a 19.2 Kbps.

Considerando que se enviarán 11 registros: Voltaje del Motor 1, Voltaje del Motor 2, Corriente del Motor 1, Corriente del Motor 2, Posición del Hoist, Peso de contenedor, Posición del Trolley, Vibraciones del Rodamiento 1 del Motor 1, Vibraciones del Rodamiento 2 del Motor 1, Vibraciones del Rodamiento 1 del Motor 2 y Vibraciones del Rodamiento 2 del Motor 2. Se tiene que:

$$Peso_{word} = 16[bits];$$

$$Cantidad\ de\ datos\ por\ segundo = 35 [datos];$$

$$Speed = \#Registros * Peso_{word}$$

$$* Cantidad\ de\ datos\ por\ segundo \quad (1)$$

$$Speed = 6160\ bps = 6kpbs$$

Por lo que, incluso Modbus RTU puede resultar útil.

Por último, se evalúa el rango de imprecisión que pueden tener las lecturas y el cual el modelo debe considerar para evaluar su rendimiento. Según su ficha técnica el sensor tiene un

rango de 0 a 25.4mm/s representados en un intervalo de corriente de 0.5 a 8mA. Con esto se obtiene que:

$$\text{Rango de medicion} = 25.4[\text{mm/s}];$$

$$\text{Rango de senal} = 7.5 [\text{mA}];$$

*resolucion*

$$= \frac{\text{Rango de medicion}}{\text{Rango de senal}} \quad (1)$$

$$\text{resolucion} = 3.387\left[\frac{\text{mm}}{\text{s}} \text{ por cada mA}\right]$$

Ahora para conocer el valor de cambio mínimo de la lectura por mA se toma el porcentaje de sensibilidad transversal mencionada en la ficha de menos de 5%. Por lo que se tiene un error de mínimo a menos de 0.169 mm/s por cada mA. Además, como en el panel CMS se evidencia que el cambio mínimo de 0.1mm/s entonces se estima que el valor de lectura puede variar del valor real a 0.1mm/s.



Como se observa en el diagrama de bloques, la operación comienza en cuanto la grúa comienza a hacer sus actividades usuales, las lecturas del driver y del sensor de vibración se recopilarán en bases de datos del PLC como se estaba ejerciendo hasta el momento, a la vez se registrarán los datos de interés en el Holding Register del servidor modbus. Después la instancia de Node RED se encargará de administrar el almacenamiento para el histórico de la CMS, para el entrenamiento del Modelo ensemble y para su respectivo análisis del comportamiento actual de la grúa. Adicional se presentan los datos actuales del driver y unas alarmas del propio CMS en la ventana principal del Dashboard.

Después de una hora se hace uso del modelo para el análisis, en caso de que no esté entrenado pasa a entrenamiento, donde hace uso de la base de datos específica para entrenamiento, durante esta parte del proceso, dependiendo de la complejidad del patrón de relación se tomara un tiempo para alcanzar los estándares necesarios para un score mayor a 90.

Con un modelo entrenado se pasa a usar los datos almacenados para el análisis, el cual se divide en dos partes para diagnosticar el estado de integridad de los rodamientos, por un lado, se cogen los datos de voltaje, corriente, posición y peso para determinar las vibraciones en operación normal supuestas y se contrastan con las lecturas reales, de esta comparación se especifican los niveles o categorías de calidad en que se encuentran las vibraciones guiándose de las normas ISO. Por el otro lado, se realizan los cálculos para el análisis de espectro de frecuencia con las lecturas reales, del todo el conjunto de valores de magnitudes de las frecuencias de Fourier se revise las de mayor magnitud, para revisar si en efecto no hay armónicos dentro de las vibraciones, que son reflejo de problemas en los elementos rodantes. Los resultados de todo el análisis se muestran en ventanas del dashboard. Tras analizar eso se estima un porcentaje de integridad y en base al mismo se especifica el estado de los rodamientos.

Con la integridad definida, se hacen los reportes y se guardan en la nube, de forma consecutiva se hace una lectura de los últimos 10 reportes en la nube para proyectar en otra ventana la evolución del estado de los rodamientos. Se revisa la información del último análisis en el momento de la escritura y si tiene un estado diferente de OK se prepara un mensaje advertencia, caso contrario, se termina al guardar el reporte y se continúa almacenando datos.

## 2.4. Flujos de node red

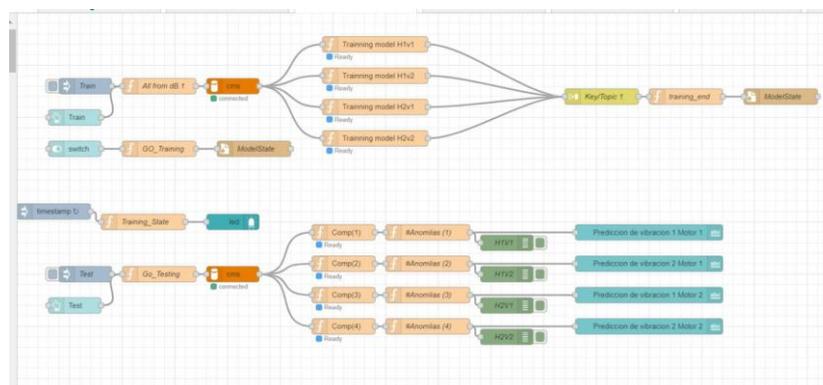
Dentro de los flujos se desarrollan funciones específicas:

- Almacenamiento y gestión de datos (DATA)
- Entrenamiento y testeo del modelo IA (MODELO)
- Mantenimiento predictivo (ANALISIS)
- Almacenamiento y gestión de diagnósticos (REPORTE)
- Monitoreo (MONITOR)

### 2.4.1. Entrenamiento de Modelo (MODEL)

Figura 2.4.1

*Training Model*



Según la figura 2.4.1., se sigue el flujo descrito a continuación:

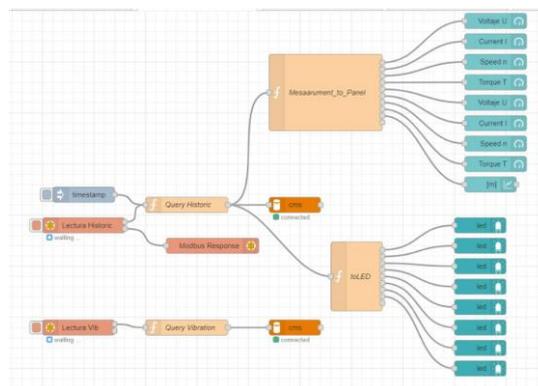
- Hacer una solicitud de lecturas de datos de la base MySQL.

- Se toman las columnas y se crea un arreglo de 5 dimensiones con los datos de voltaje, corriente, posición del hoist, carga y posición del trolley.
- Se usa un código en Python para cargar las librerías de modelos sklearn para entrenar modelos de aprendizaje. Se guarda el modelo en archivo interno de la Raspberry.
- De la misma forma se hace uso de una solicitud para SQL y sacar datos de entrada y reales de un conjunto de vibración de prueba, se hacen comparaciones considerando que el sensor tiene un margen de 0.17 de error y se imprimen los porcentajes de coincidencias.

#### 2.4.2. Lectura y Almacenamiento (DATA y MONITOR)

**Figura 2.4.2**

*Flujo Recolección de Datos (CMS to LocalStorage)*



El flujo de la Figura 2.4.2 sigue el siguiente proceso:

- Obtención de Datos desde el PLC por protocolo Modbus usando un nodo Lectura de Modbus para el histórico (1Hz) y para el análisis vibratorio (35Hz).
- Se desglosan los datos en los telegramas de Modbus para separar las lecturas a presentar en el Panel de Node RED. Dentro del desglose se realizan conversiones para

acondicionar los datos para su lectura y almacenamiento. Sobre todo, en los LEDs de alarmas.

- Solicitud SQL en código javascript a la base de datos cms para guardar los datos de Modbus

**Figura 2.4.3**

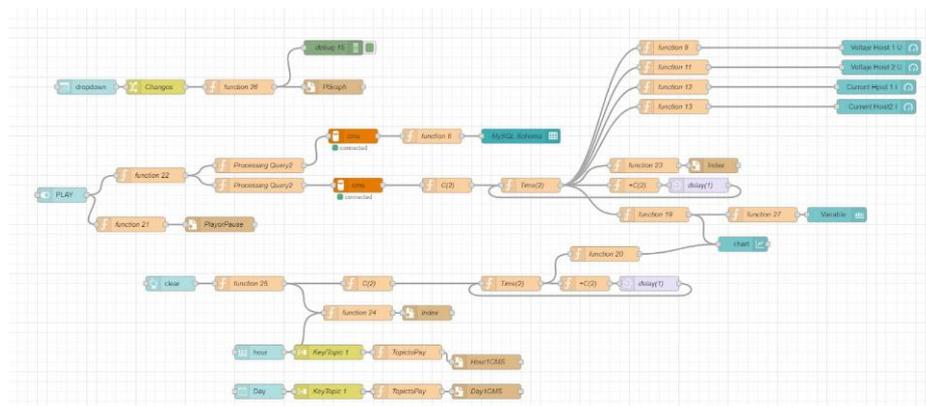
*CMS: Hoist Panel en Dashboard*



### 2.4.3. Histórico del CMS (DATA y MONITOR)

**Figura 2.4.4**

*CMS: Hoist Panel en Dashboard*

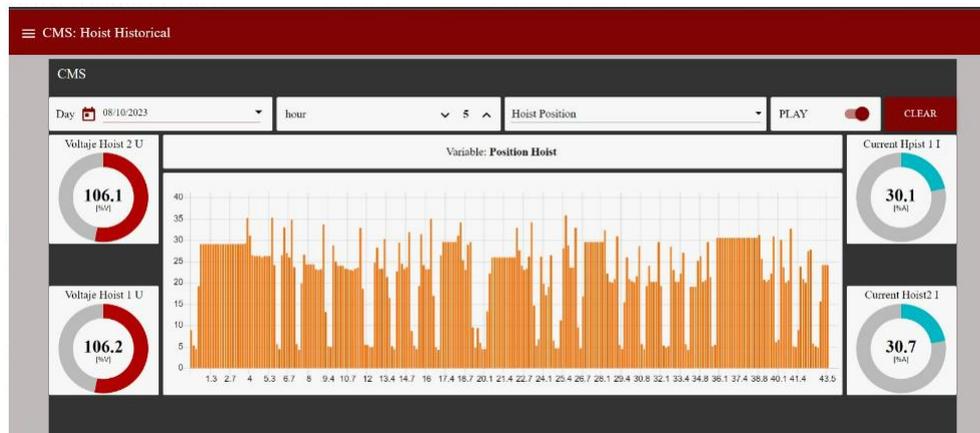


- Los primeros nodos a la izquierda vistos en la Figura 2.4.4 representan los botones, seleccionadores de texto y fecha que recopilan la información definida por el usuario para navegar en el histórico e iniciar la revisión.

- Cuando se especifica fecha, hora y variable a revisar, se hace la solicitud SQL para acceder a la porción de información requerida.
- Los otros nodos se encargan de distribuir las columnas de la solicitud para su respectiva presentación en el panel en la Figura 2.4.5.
- El resto de los nodos ayudan al control y orden de la presentación de los datos como condiciones, delays y guardado de valores.

**Figura 2.4.5**

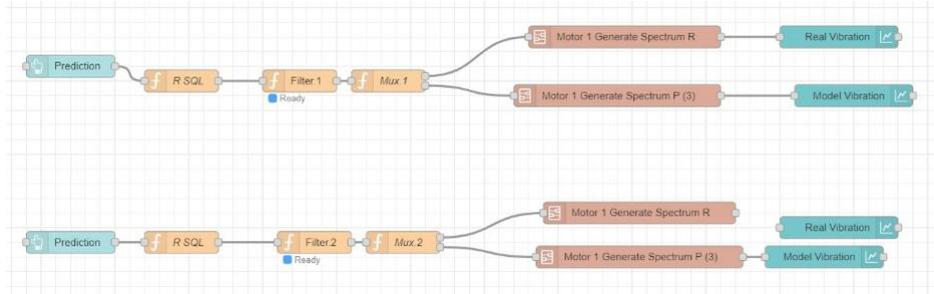
*CMS: Hoist Historical en Dashboard*



### 2.4.4. Presentación de Vibraciones (MODEL)

Figura 2.4.6

Flujo Visualización de Predicciones vs Lecturas

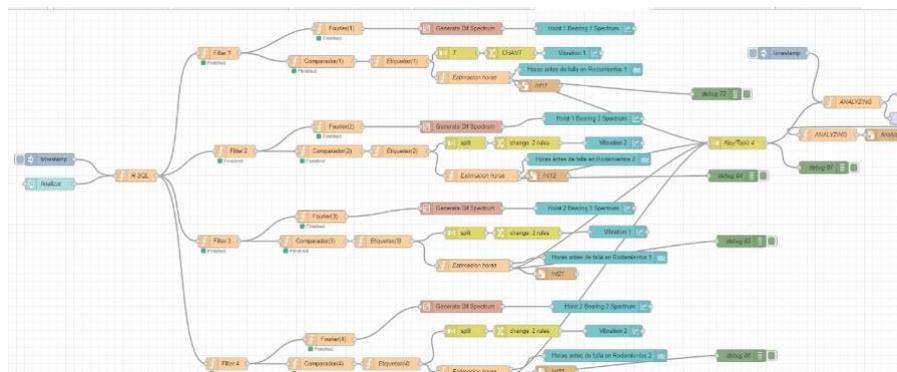


- En la Figura 2.4.6 se continúa con los acondicionamientos de las entradas para ingresar al modelo y generar las predicciones. Seguido se realizan las predicciones y se envían ambas señales.
- Las vibraciones se dividen y van a subflujos que se encargan de ir presentando los valores en las gráficas.

### 2.4.5. Analizador de Vibraciones (ANALYSIS y MONITOR)

Figura 2.4.7

Flujo Análisis de Datos Diario



- Con los primeros nodos en la Figura 2.4.8 se hace la solicitud a la base de datos cms para obtener los datos de vibraciones.

- Se continua con los acondicionamientos de las entradas para generar predicciones. Con los valores reales y predichos se pasan a los dos analizadores.
- Los valores de vibración reales pasan por un código en Python para calcular la transformada de Fourier. Luego por sub-flujo que antes de presentar en gráficos revisa todas las vibraciones en el espectro y selecciona en orden aquellas con magnitudes elevadas en el espectro, para presentarlas en una gráfica en la página de Diagnostico.
- En el otro camino se procede con una comparación del sistema con el modelo para marcar los niveles de cada lectura (A, B, C, D o Anomalía), para su posterior contabilización de los niveles y obtener una proporción relacionada al eficiencia o calidad del sistema real y reflejarlo en la porción del tiempo completo de vida útil de las máquinas y estructuras.
- Las coincidencias con los niveles son presentadas en un diagrama junto al tiempo antes del mantenimiento.

**Figura 2.4.8**

*Panel de niveles de vibración.*

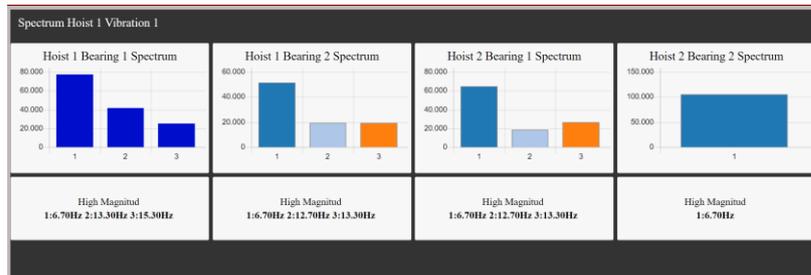


En la Figura 2.4.9 se presenta el panel principal donde están las gráficas con los resultados de la asignación de niveles de vibraciones (A, B, C, D y Anomalía) para definir el

tiempo y el estado de los rodamientos, procedimiento explicado en la sección 2.6 del Capitulo 2.

**Figura 2.4.9**

*Panel de espectro de frecuencia.*

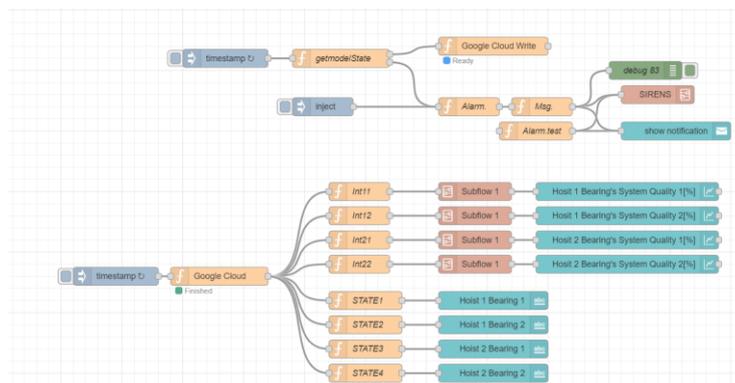


El panel en la Figura 2.4.10 están las magnitudes mas pronunciadas del espectro de frecuencia revisado de 0.1 a 17.5 Hz.

### 2.4.6. Reportes de Google (Reporte y Monitoreo)

**Figura 2.4.10**

*Flujo para conexión con Google*



- Con los nodos superiores del a Figura 2.4.11 se evalúa si se ha realizado un análisis en el momento, con el valor de una variable ANALI, y entonces manda a escribir los resultados del análisis vibratorio que son el estado del rodamiento (OK,

WARNING, FAULT), porcentaje de integridad del rodamiento (0 a 100%) y un mensaje si hay existencias de armónicos (FUNDAMENTAL, HARMONIC).

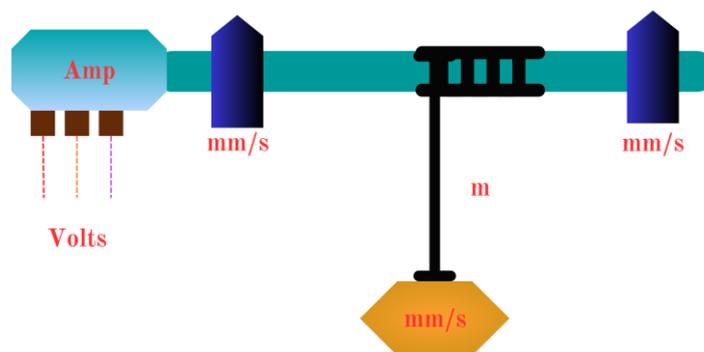
- Otra parte de los nodos superiores se encargan de revisar los resultados y generar mensajes en caso de que el estado del rodamiento no sea OK, este estado cambio si el porcentaje de integridad baja a un 85% para que sea WARNING y FAULT si baja a un 60%.

## 2.5. Entrenamiento

El modelo de Regresión por Búsqueda de Árbol Aleatorio (RandomForestRegressor) se entrenó usando la información de los archivos .cms anotados del panel CMS versión8 correspondiente al voltaje del bus DC, la corriente del motor, la posición del Spreader y las vibraciones de los rodamientos. Las características eléctricas y la posición ayudan al modelo a interpretar como se ve afectada la vibración mientras el Hoist realiza el trabajo de subida y bajada de los contenedores. Respetando los factores presenten en el diagrama del sistema:

**Figura 2.5.1**

*Diagrama del sistema Motor-rodamientos-spreader*



Debido a las referencias que necesita el sistema para entrenarse, guiándose del diagrama en la Figura 2.5.1, se registraron los valores visibles en el panel CMS en una hoja de Excel.

**Figura 2.5.2**

Lecturas exportadas a Schema de MySQL

**Table: sts\_vibone**

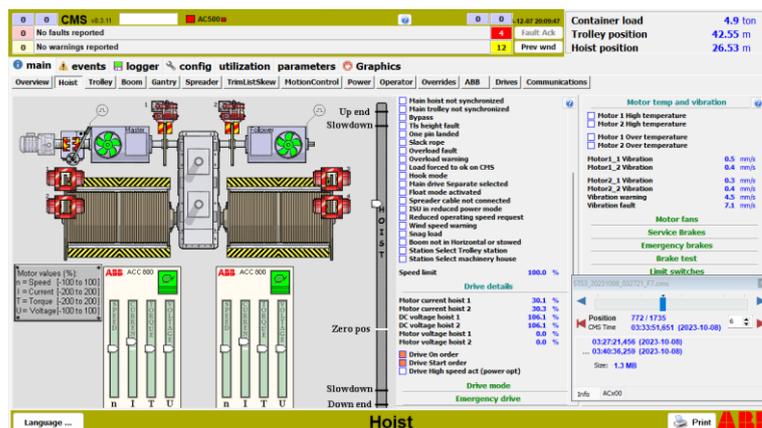
**Columns:**

Column Name	Data Type
Voltage_DC_Motor_1	double
Voltage_DC_Motor_2	double
Current_Motor_1	double
Current_Motor_2	double
Hoist_Position	double
Trolley_Position	double
Load	double
Motor_1_Vibration_1	double
Motor_1_Vibration_2	double
Motor_2_Vibration_1	double
Motor_2_Vibration_2	double

Los datos en el Excel pasaron a ser importados en el Banco de Trabajo de MySQL, como se muestra en la Figura 2.5.2, mediante un código de Python expuesto en Apéndice C.

**Figura 2.5.3**

Panel CMS



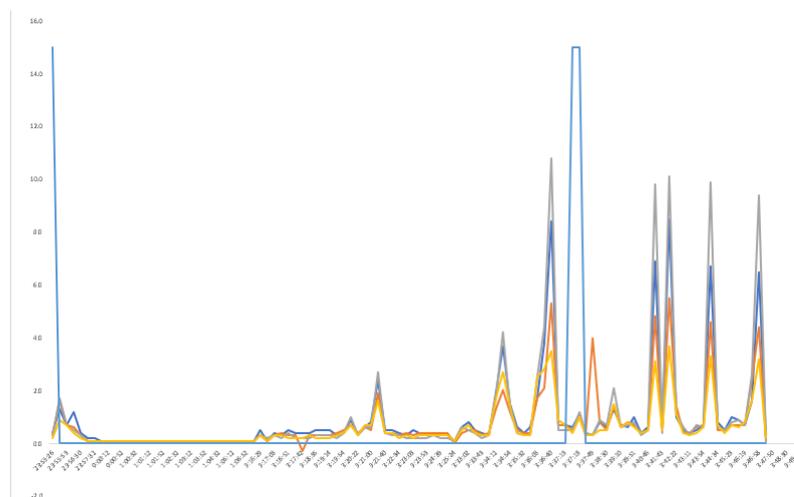
En la Figura 2.5.3 está la ventana del panel gráfico perteneciente al programa CMS versión 8, con la sección Hoist correspondiente a las lecturas y gráficas dinámicas de la cabina del a grúa STS donde trabajan los motores Hoist, sus rodamientos y motores de freno.

## 2.6. Diagnóstico de vibraciones

Según Connelius Scheffer y Paresh Girdhar en su estudio de mantenimiento predictivo por análisis vibratorio, estableció que el espectro de vibración puede entregar información acerca del estado mecánico de las maquinarias y la infraestructura que involucra a los rodamientos. En base a este espectro se complementa el diagnóstico de desbalance, desalineación y demás problemas mecánicos en el conjunto motor-rodamiento, que suele generar que el encoder pierda la posición del eje y desincroniza la manipulación desde la interfaz CMS, y el deterioro del alambre metálico que soporta el peso del spreader al momento de movilizar un contenedor, que culmina con la ruptura de este elemento crucial.

**Figura 2.6.1**

*Proyección de la lectura de sensores de vibración de motor Hoist 1 y 2, y del sensor Slack Rope*



De la figura 2.6.1 se muestran los fragmentos donde las vibraciones han alcanzado un pico alto del resto de valores. Entonces se evidenció que el sensor puede tener lecturas de la

velocidad vibratoria van desde menos de 1 mm/s hasta los 10.5 mm/s, dependiendo de la energía y carga aplicada en los rodamientos. Y se tiene un momento en que las vibraciones incrementan y por un instante la alarma Slack Rope, que normalmente indicaría cuando hay una ruptura en el alambre que soporta al carro, se activa por uno segundos para después desaparecer, lo que indica que el cable fue estirado hasta que la tensión se estabilizo y el cuerpo recupero su estado de equilibrio, aumentando las vibraciones. Estas graficas se obtuvieron con la revisión del histórico de extensión .cms por medio de la interfaz gráfica CMS8, aclarando que estos archivos refieren a que la grúa está trabajando con normalidad, o al menos que no presenta fallas aparentes.

Considerando que el sensor suele recibir valores muy altos pese a estar reacondicionado el sistema motor-rodamiento, y según las normas ISO debería tener casi todas sus vibraciones dentro de los 3.5 mm/s, entonces, el modelo de inteligencia artificial interpretara esos altos que a veces son captados por el sensor y evalúa un sistema de niveles basado en las normas ISO 10816:

- Si el valor de vibración ha aumentado hasta 0.7mm/s, se etiqueta como A y se define como aun en estado de reacondicionamiento reciente,
- Si el valor de vibración ha aumentado hasta 2mm/s, se etiqueta como B y puede operar indefinidamente.
- Si el valor de vibración ha aumentado hasta 5mm/s, se etiqueta como C y puede operar por poco tiempo.
- Si el valor de vibración ha aumentado a más de 5mm/s, se etiqueta como D y estas vibraciones ya pueden producir danos a todos los elementos del sistema motor-rodamiento.

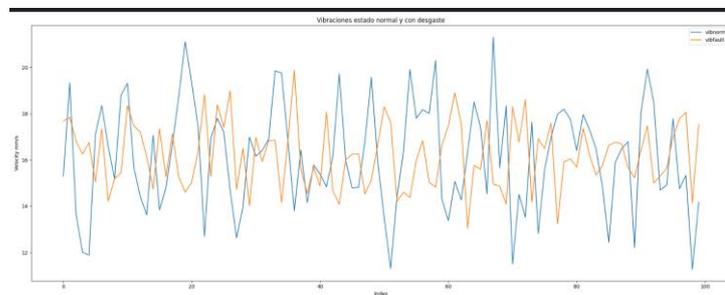
Se considero que cuando los valores se empiezan a etiquetar como C y D, se presenta ya presenta deterioro, hasta entonces, no se reflejara que las horas antes de mantenimiento descenden o su porcentaje de integridad.

Para poder entrar a los modelos de machine learning es necesario tener una referencia de vibraciones en trabajo normal, como la mostrada previamente, y comparar con una serie de datos que involucren fallas, las cuales se pueden definir revisando el resto de históricos CMS, aunque, la revisión de estos no se pudo realizar de forma eficiente o a corto plazo, por lo que se requirió el uso de una base de datos de un estudio de fallas mecánicas y eléctricas para determinar de qué forma se puede probar al programa de Node RED.

La base de datos se descargo de la pagina zenodo.org y fue una contribución de la plataforma ERA-LEARN de Hozion Europe, especializada en la búsqueda de materiales y data para investigaciones del uso de algoritmos para procesamiento de datos como los modelos de inteligencia artificial. En esta base de datos se probaron algunos motores que presentaban desbalance y deterioro en los elementos rodantes, así como problemas eléctricos en los motores.

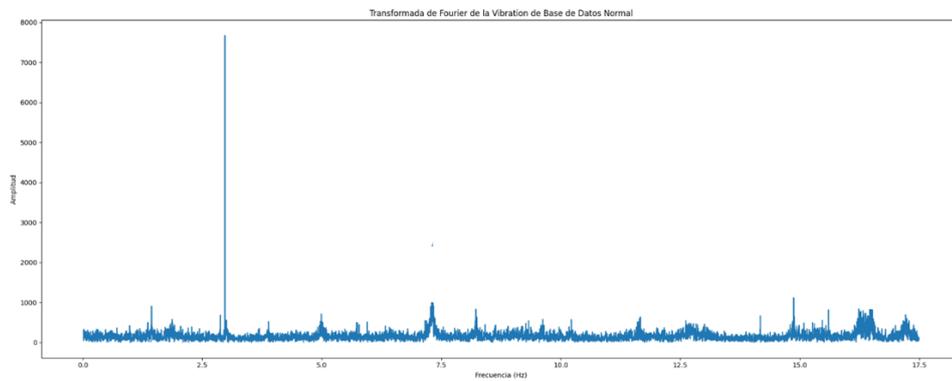
### Figura 2.6.2

*Proyección de los datos de prueba de un motor asíncrono con carga mecánica en estado normal (curva azul) y presentando fallas mecánicas (curva naranjar).*



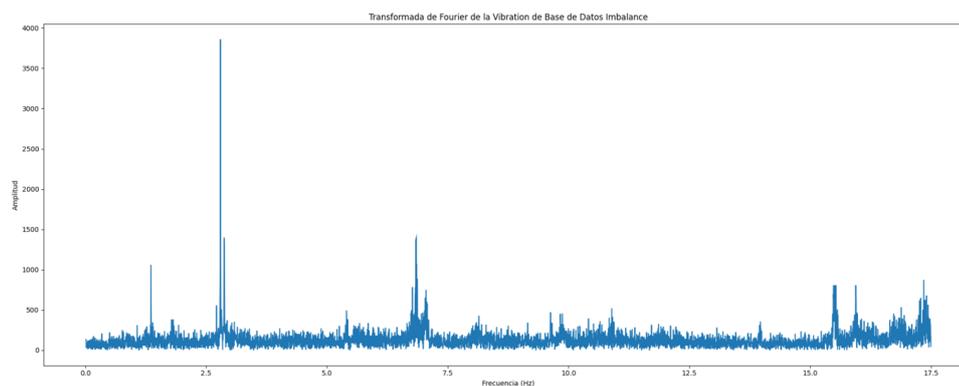
**Figura 2.6.3**

*Espectro de base de datos estado Normal*



**Figura 2.6.4**

*Espectro de base de datos estado Desgastado*



Según lo revisado en la literatura y en estudios realizados de vibración, visibles en la Figura 2.6.2, se tiene que las perturbaciones por fallos mecánicos internos de los elementos que son parte del sistema o por la estructura que soporta a estos, son en esencia aportaciones de armónicos que se presentan cerca de la frecuencia fundamental, a frecuencias menores y sobre todo a frecuencias mayores, visible al comparar la Figura 2.6.3 y 2.6.4. Por ello, para las pruebas se decidió inyectar armónicos de frecuencias dentro de la gama del espectro en la señal de vibración reacondicionada y comparar los resultados de evaluar los niveles de vibración de la señal real y la predicha por el modelo.

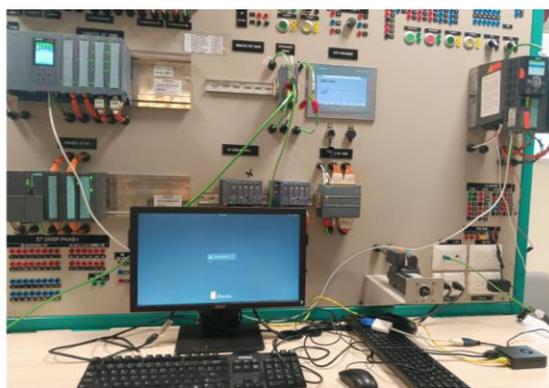
## Capítulo 3

### 3.1. Resultados y análisis

#### 3.1.1. Integración de Raspberry a la red local para Recolección de Datos

Figura 3.1.1

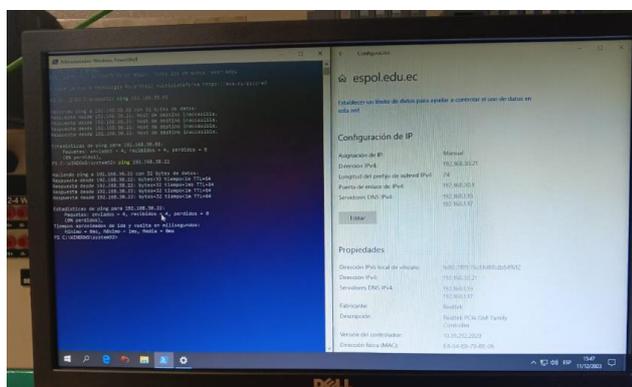
*Servidor Raspberry integrado a red de la planta*



En la Figura 3.1.1 se tiene al Tablero 1 del Laboratorio de Sistemas Complejos con el PLC S7 1516 en la parte superior izquierda y del lado derecho al variador G120 CU250-2S, en la mesa se observa la carcasa de aluminio que contiene a la placa Raspberry con la instancia de Node RED y la pantalla mostrando para iniciar la sesión en Ubuntu.

Figura 3.1.2

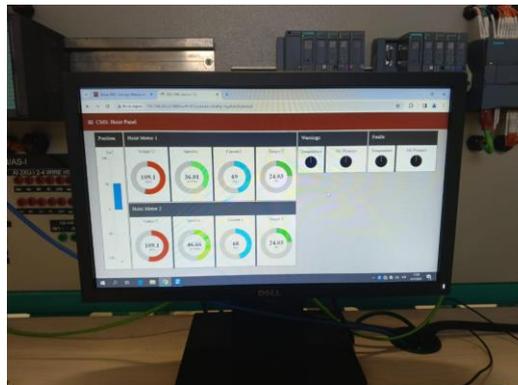
*Reconocimiento de equipo Raspberry por ordenadores de oficina*



La placa Raspberry se conectó a la red LAN mediante su puerto Ethernet, con una IP consistente con la subred local 192.168.30.22 y asociado al proxy de la ESPOL para tener acceso, siendo reconocida por otros ordenadores en la red como se observa en la Figura 3.1.2, y no ser bloqueado por los protocolos de seguridad. Se inicio la instancia de Node RED y a través de un script se abrió la sesión en el ordenador de dirección IP 192.168.30.23. Tal cual se espera usar si fuese montado en la grúa STS.

**Figura 3.1.3**

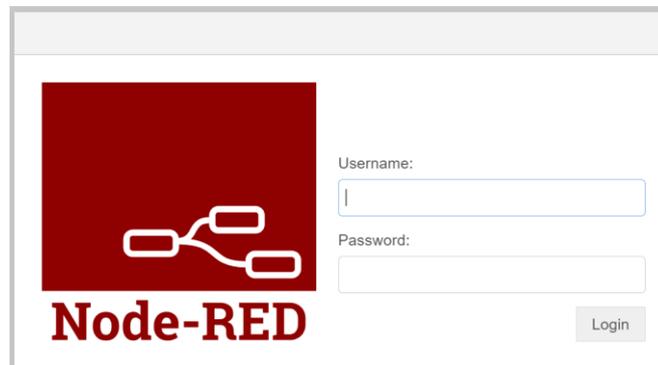
*Abertura del panel*



Con los permisos configurados no hubo problema en que el ordenador (la oficina) pueda abrir la instancia del panel CMS Node RED mostrado en la Figura 3.1.3, sin embargo, el tiempo en que cargo la página fue más lento que en pruebas con red WiFi domiciliaria, a causa de las validaciones del proxy.

**Figura 3.1.4**

*Cuadro de dialogo para Inicio de Sesión.*

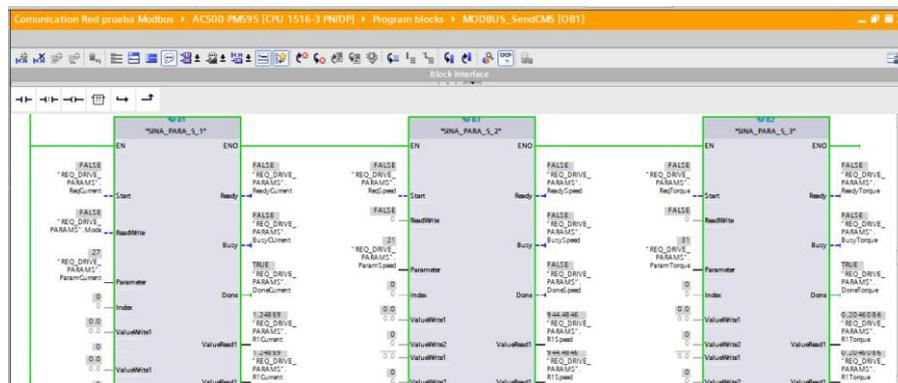


Un problema que se anticipó fue que se podía al panel desde cualquier punto de la red, y peor aún, acceder a la programación de Node RED desde le panel al quitar “/ui” del URL, por lo que, se añadió claves encriptadas para asegurarse de que no se hagan modificaciones ni revisiones por personal no autorizado, como se muestra en la Figura 3.1.4.

Los nodos de Modbus cada 1 segundos recopilaron los datos extraídos por los bosques SINA PARA S, visibles en la Figura 3.1.5, de voltaje, corriente, torque y velocidad, además de datos referentes a posición, peso y las vibraciones, estos últimos generados para concordar con lo que presenta la base de datos de la STS.

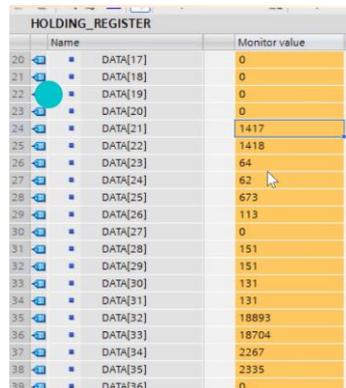
**Figura 3.1.5**

*Bloques recibiendo valores de parámetros.*



**Figura 3.1.6**

*Registro de sostenimiento con valores de parámetros.*



	Name	Monitor value	Re
20	DATA[17]	0	
21	DATA[18]	0	
22	DATA[19]	0	
23	DATA[20]	0	
24	DATA[21]	1417	
25	DATA[22]	1418	
26	DATA[23]	64	
27	DATA[24]	62	
28	DATA[25]	673	
29	DATA[26]	113	
30	DATA[27]	0	
31	DATA[28]	151	
32	DATA[29]	151	
33	DATA[30]	131	
34	DATA[31]	131	
35	DATA[32]	18893	
36	DATA[33]	18704	
37	DATA[34]	2267	
38	DATA[35]	2335	
39	DATA[36]	0	

En la figura 3.1.6 se tiene un vistazo de la base de datos en la programación del PLC que actúa como Registro de Sostenimiento del Servidor Modbus recibiendo los valores pasados a porcentaje de las mediciones de voltaje, corriente, torque, velocidad y lecturas en su unidad de vibraciones y posiciones.

Un conflicto que se presentó fue que los datos de vez en cuando se guardan como 0 a pesar de estar detenido el motor, generados datos erróneos en la base de datos, pero solo fue cuestión de ajustar las líneas de código Ladder que generan las solicitudes para llamar a los bloques de parámetros, dado que no se puede acceder a las lecturas simultáneamente, cosa que no pasa en la programación y bases de datos internas del AC500, por la forma en que el sistema CMS almacena los datos, pero con calibración alcanzo igual el resultado requerido de tener listos los datos al momento de ser leídos por Modbus.

Cabe mencionar que durante las pruebas se evidenció que no se captaba la información completa de vibraciones debido a que se estima que su frecuencia fundamental está a 6Hz, y se comenzó con 1Hz de muestreo, esto buscando tener una gran cantidad de datos que describan el trabajo de la grúa STS día a día, que fueron 100,000 datos por 24 horas, entonces se anexo una lectura Modbus independiente a una frecuencia de 35Hz para captar hasta el segundo

armónico de la frecuencia fundamental de la vibración, para sobrescribirse en una base de datos que alcanza unos 140,000 datos equivalentes a 1 hora.

**Tabla 3.1-1**

*Fragmento del Excel importado desde la base de datos sts1\_db llenada por el flujo de Recolección de Datos*

Fecha	Hora	Voltage_DC_Motor_1	Voltage_DC_Motor_2	Current_Motor_1	Current_Motor_2	Hoist_Position	Trolley_Load	Trolley_Position	Motor_1_Vibration_1	Motor_1_Vibration_2	Motor_2_Vibration_1	Motor_2_Vibration_2	Speed_Motor_1	Speed_Motor_2	Torque_Motor_1	Torque_Motor_2
1/4/2024	16:49:46	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:47	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:49	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:50	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:51	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:53	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:54	105.2	105.2	33.2	32.5	23.6	25	0	0.34	0.29	0.29	0.34	42.14	42.14	50.01	50.01
1/4/2024	16:49:56	105.1	105.1	34	33.3	23.6	18.3	0	0.34	0.29	0.29	0.34	42.14	42.14	36.58	36.58
1/4/2024	16:49:57	105.1	105.1	33.8	33.1	23.6	15.6	0	0.34	0.29	0.29	0.34	42.08	42.08	31.25	31.25
1/4/2024	16:49:58	105.2	105.2	33.8	33.1	30.7	15.3	0	0.73	0.64	0.64	0.73	90.8	90.8	30.54	30.54
1/4/2024	16:50:00	105.2	105.2	28.2	27.6	30.7	11.5	0	0.79	0.69	0.69	0.79	98.15	98.15	22.9	22.9
1/4/2024	16:50:01	105.2	105.2	35.1	34.4	30.7	16.2	0	0.98	0.85	0.85	0.98	70.55	70.55	32.37	32.37
1/4/2024	16:50:02	105.2	105.2	24.4	23.9	30.7	11.3	0	1.2	1.04	1.04	1.2	70.55	70.55	22.55	22.55
1/4/2024	16:50:04	105.2	105.2	24.3	23.8	30.7	11.3	0	1.2	1.04	1.04	1.2	70.55	70.55	22.65	22.65
1/4/2024	16:50:05	106.4	106.4	24.3	23.9	44.2	11	0	1.2	1.04	1.04	1.2	70.55	70.55	22.03	22.03
1/4/2024	16:50:06	106.4	106.4	24.3	23.9	44.2	11	0	1.2	1.04	1.04	1.2	70.55	70.55	22.03	22.03
1/4/2024	16:50:08	106.4	106.4	24.3	23.8	44.2	11.4	0	1.2	1.04	1.04	1.2	70.55	70.55	22.7	22.7
1/4/2024	16:50:09	106.4	106.4	24.3	23.8	44.2	10.7	0	1.2	1.04	1.04	1.2	71	71	21.46	21.46
1/4/2024	16:50:11	106.4	106.4	24.4	23.9	44.2	10.9	0	1.2	1.04	1.04	1.2	71	71	21.87	21.87
1/4/2024	16:50:12	106.4	106.4	24.4	23.9	44.2	12.1	0	1.2	1.04	1.04	1.2	70.23	70.23	24.24	24.24
1/4/2024	16:50:13	369.4	369.6	24.2	23.7	44.2	10.8	0	1.2	1.04	1.04	1.2	70.23	70.23	21.64	21.64

### 3.1.2. Nueva base de datos

La revisión de la información en la base de datos generada por la Raspberry para el monitoreo hora a hora de la grúa STS, permitió exitosamente generar graficas de las variables almacenadas en un día y hora específicos una tras otra, aunque, se tuvieron que adicionar consideraciones como protocolos que impidan el cruce de generación en tiempo real y limpieza del área de gráficos para evitar mezclas entre las lecturas. Con la tabla como complemento para revisar cada punto del dibujo, ofreciendo dinamismo y orden a la información de mejor manera que el panel original saturado de datos. La tarea tarda en realizarse de 3.6 a 76 segundos según se ajuste el tiempo con el que se van presentando los datos en el diagrama, tal cual en la Figura 3.1.7.

**Figura 3.1.7**

*Gráfico de Posición del Hoist en el histórico.*

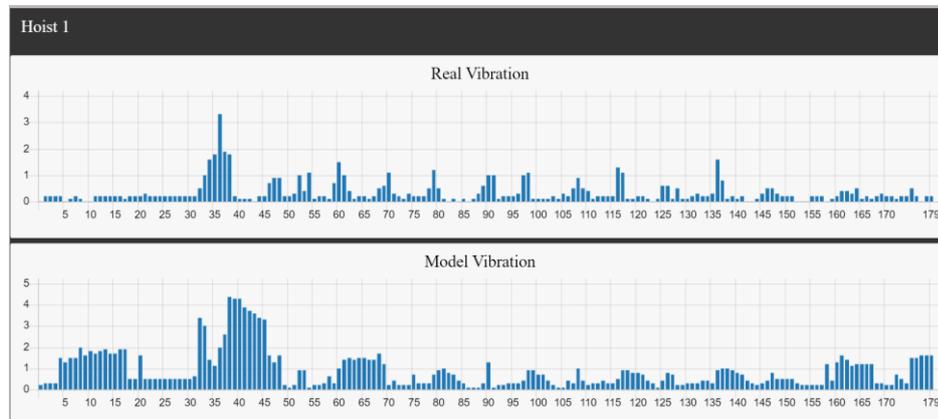


### **3.1.3. Entrenamiento modelo Aprendizaje de Maquina.**

Con 2,000 datos se realizó un entrenamiento con el algoritmo de regresión por Bosque Aleatorio (RandomForestRegressor) donde se revisaron entre 10 y 100 posibles combinaciones de hiper parámetros donde se predominaba profundidades de 100 a 2,000 y estimadores de 100 a 2,000. Consiguiendo raíces de error cuadrático medio de 1.54 mm/s a 1.05 mm/s y errores absolutos medio de 0.95 mm/s a 0.47 mm/s, que, al momento de revisar las coincidencias, considerando que las vibraciones tienen un offset de 0.1 a 0.2 mm/s, se tuvieron hasta 50% de coincidencias de todos esos datos. En la Figura 3.1.8 se tiene un fragmento de las señales generadas por el modelo y las reales.

**Figura 3.1.8**

*Mejor Predicción de las primeras pruebas*

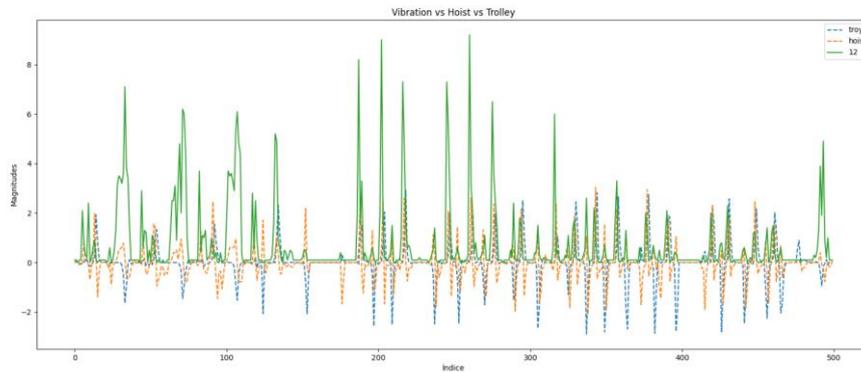


En primera instancia, mientras aún no se contó con el uso del script de Python para capturar datos del panel CMS, se anotaron a mano los valores hasta 2,000 datos. Se usaron estimadores de 1,000 hasta 10,000 y sin límites de profundidad, sin embargo, eso generó un sobreajuste, que en entrenamientos de modelos se definen como la condición en que el modelo se ajusta tanto a los datos de entrenamiento que se desvincula totalmente de nuevos datos, causando errores más grandes.

Tras obtener más de 100,000 datos, se usaron muestras de 4,000, 8,000, 12,000 y 15,000 datos, aunque se seguía obteniendo errores absolutos de 0.41 mm/s o 0.38 mm/s, pero con la cantidad de datos se pudo evaluar con gráficos como se generan los picos de vibración que se presentaron en los datos, de hasta 10.5 mm/s. De esa forma se identificó que se tiene mayor relación con la diferencia de posición del Trolley y el Hoist que sus valores instantáneos.

### Figura 3.1.9

*Análisis de diferencias de posición Hoist y Trolley en relación con la vibración.*



En la Figura 3.1.9 se revisó los cambios de posición a causa del trabajo del Motor Hoist y el Motor Trolley y las influencias en las vibraciones, teniendo una relación y se optó por cambiar las columnas de Hoist\_Position y Trolley\_Position por sus cambios diferenciales y se anexó una columna de oscilación con una función que representa el RMS de los ciclos de una función trigonométrica a 6.6Hz, esto fue con el fin de darle apoyo al modelo para interpretar patrones oscilantes, esto puede revisarse en la Figura 3.1.10. Se obtuvieron mejores resultados con errores absolutos medios menores a 0.06 mm/s con 90% de coincidencias en las comparaciones. Cabe mencionar, que tuvo complicaciones con algunos picos de vibración, que serían ese 10% que no se consiguió predecir con exactitud menor o igual a 0.1 mm/s. Estos resultados son apreciables en las Figuras 3.1.11, 3.1.12 y 3.1.13.

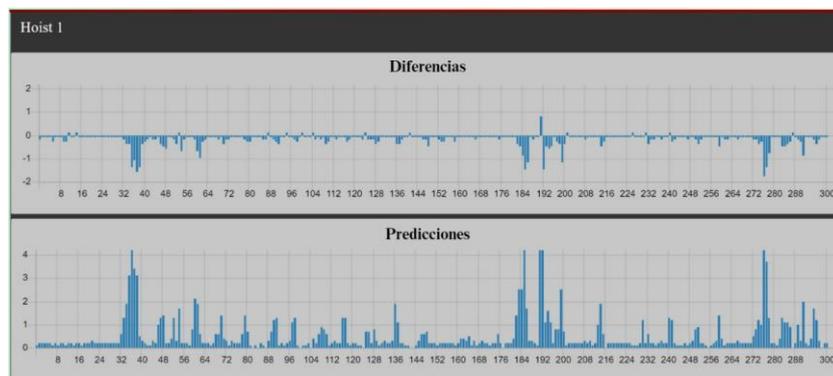
**Figura 3.1.10**

*Fragmento del entrenamiento con reacondicionamiento de valores de entrada.*

```
13 df = pd.DataFrame(msg['payload'])
14 df.loc[df['Trolley_Position'] > 100, 'Trolley_Position'] = 40
15 df.loc[df['Hoist_Position'] > 100, 'Hoist_Position'] = 40
16 df['Hoist_Position'] = df['Hoist_Position'].diff()
17 df['Hoist_Position'].iloc[0] = 0
18 df['Trolley_Position'] = df['Trolley_Position'].diff()
19 df['Trolley_Position'].iloc[0] = 0
20 # Obtener los arreglos A y B desde el DataFrame
21 times = np.arange(0, len(df['Hoist_Position'])/35, 1/35)
22 freqs = 6/2
23 w = 2*np.pi*freqs
24 fsin = np.cos(w*times)
25 velocidad_primitiva = np.sqrt((fsin**2))
26 df['Oscilacion'] = velocidad_primitiva
27
28 entrada = df[['Voltage_DC_Motor_1', 'Current_Motor_1', 'Hoist_Position', 'Load', 'Trolley_Position', 'Oscilacion']]
29
30 hoist1 = entrada.iloc[1:150000,:].to_numpy()
31 #print(hoist1)
32 h1v1 = df['Motor_1_Vibration_1'][1:150000].to_numpy().reshape((-1, 1))
33
```

**Figura 3.1.11**

*Predicción vs Diferencia con precisión del 90%*



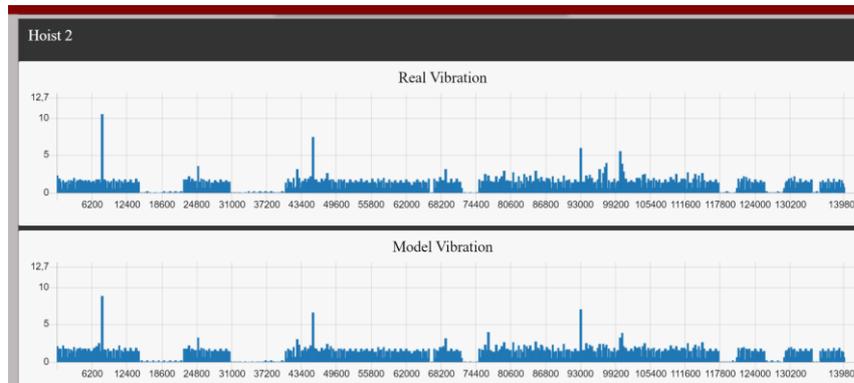
**Figura 3.1.12**

*Predicción vs Real Hoist1 con precisión del 90%*



**Figura 3.1.13**

*Predicción vs Real Hoist 2 con precisión del 90%*



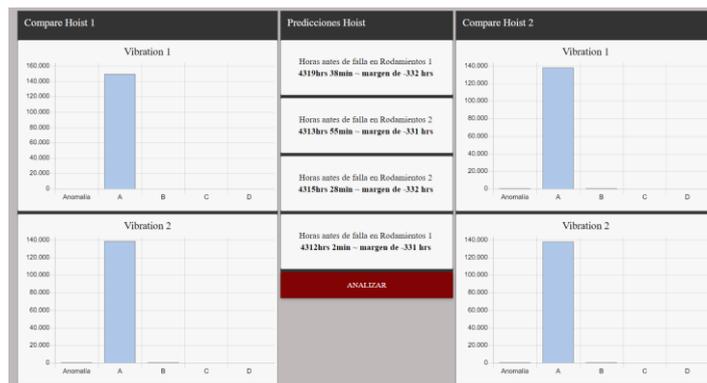
El entrenamiento tarda en realizarse en 5 minutos.

### 3.1.4. Reconocimiento de fallas

Con el resto de los datos considerados de operación normal porque el conjunto motor-rodamiento estaba recientemente reacondicionado, se hizo la revisión de las proporciones de nivel de vibración, visto en la Figura 3.1.14.

**Figura 3.1.14**

*Proporciones de magnitudes de vibración de Motor1 recién reacondicionado*



Al momento de tomar las muestras de la grúa STS recién reacondicionada para el programa de Node RED, se observó que, según el modelo IA, el 100% de las vibraciones se mantienen en nivel A, tal cual se esperaba y se tiene las horas completas de 6 meses para el mantenimiento programado.

Se realizaron algunas pruebas en las que se tomaba a la señal real y se le inyectaban los armónicos, las visualizaciones se hicieron con lenguaje Python y también el conteo respectivo de las diferencias de 0.1 en 0.1 mm/s, recopilando cada diferencia entre cada elemento de ambas señales de forma paralela, haciéndola absoluta y anotando la frecuencia con la que se presenta.

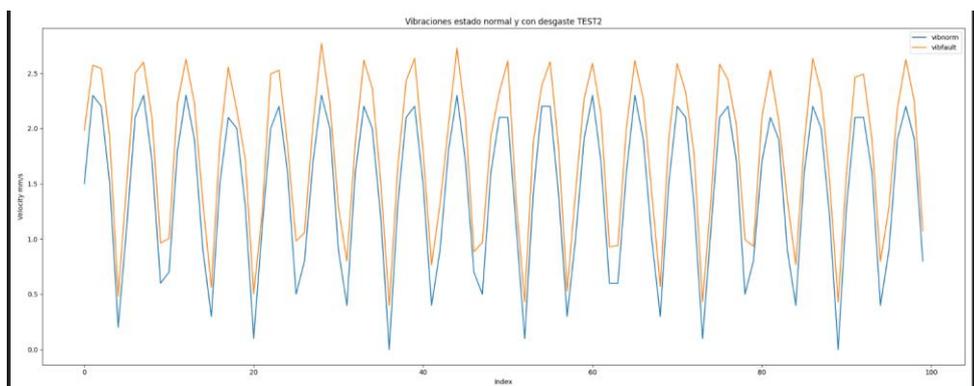
**Figura 3.1.15**

*Frecuencias de diferencias generadas por Python*

```
El número 0.7 : 45920 veces.  
El número 0.6 : 29120 veces.  
El número 0.8 : 23520 veces.  
El número 0.5 : 21840 veces.  
El número 0.4 : 10080 veces.  
El número 0.9 : 9520 veces.
```

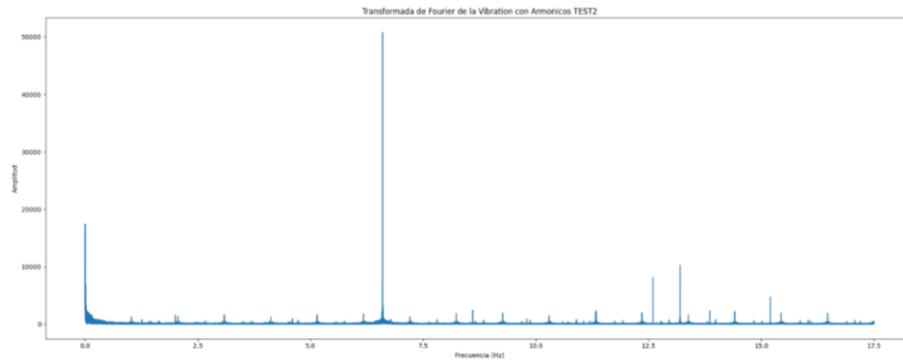
**Figura 3.1.16**

*Comparación señal en estado normal (azul) y señal con amplitudes aumentadas (naranja)*



**Figura 3.1.17**

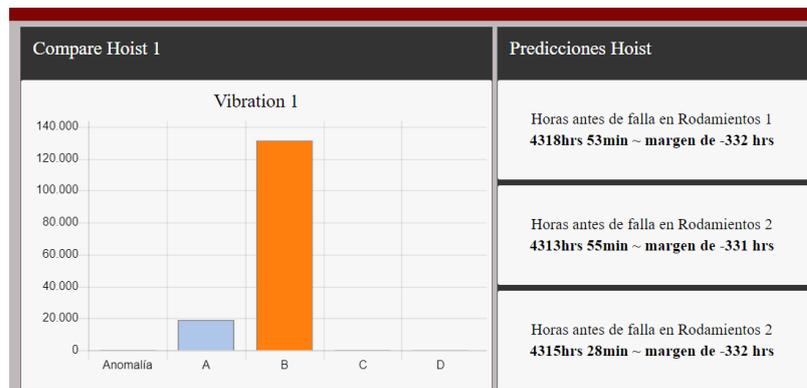
*Espectro de frecuencia con amplitud aumentada.*



Las figuras 3.1.16 y 3.1.17 muestran la inyección de armónicos a muy baja amplitud para mostrar como la fundamental se seguía manteniendo como la frecuencia de mayor a ganancia, y luego se mandó al sistema a analizar usando el modelo, ya que, estas comparaciones iniciales fueron con la señal real. Se tuvo un 100% de las vibraciones en nivel A.

**Figura 3.1.18**

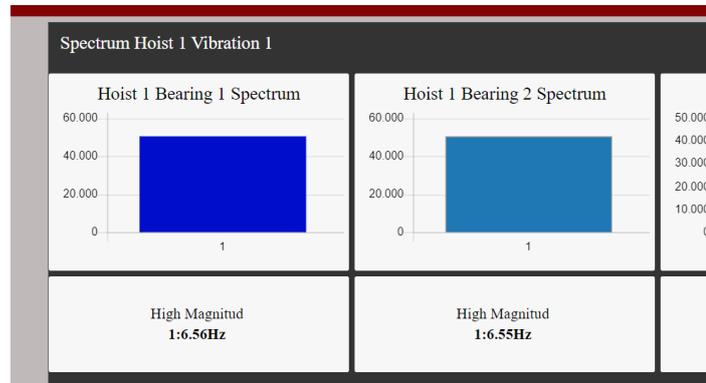
*Proporciones de magnitudes de vibración de Motor1 con emulación de desgaste parcial de soldadura o afloje de los soportes*



Se prosiguió con el aumento de las magnitudes de los armónicos hasta conseguir que la mayoría de las vibraciones pasen al nivel B como se muestra en la Figura 3.1.18, según el modelo el 94% de las vibraciones indican que se puede trabajar indefinidamente, pero es interpreta que ya no está recién reacondicionada.

**Figura 3.1.19**

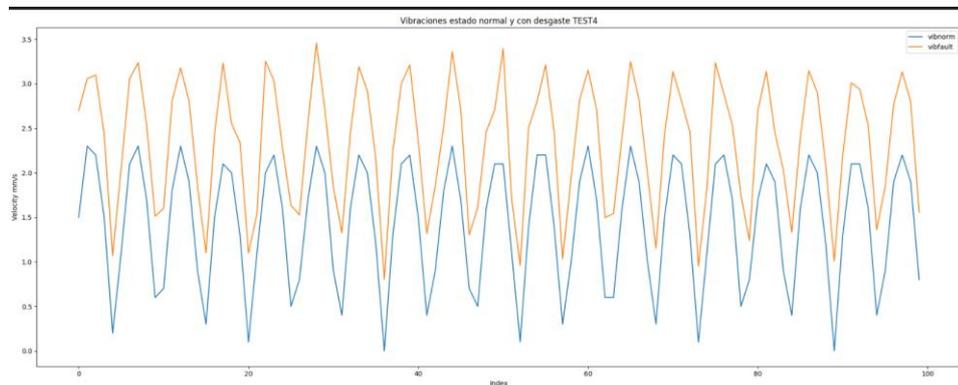
*Espectro de Prueba 4*



En la Figura 3.1.18 esta una ventana con las magnitudes de vibraciones prominentes de la Prueba 4 con la que ya se determinó al sistema motor-rodamiento en nivel B de vibraciones, manteniéndose como la frecuencia fundamental como la única influyente.

**Figura 3.1.20**

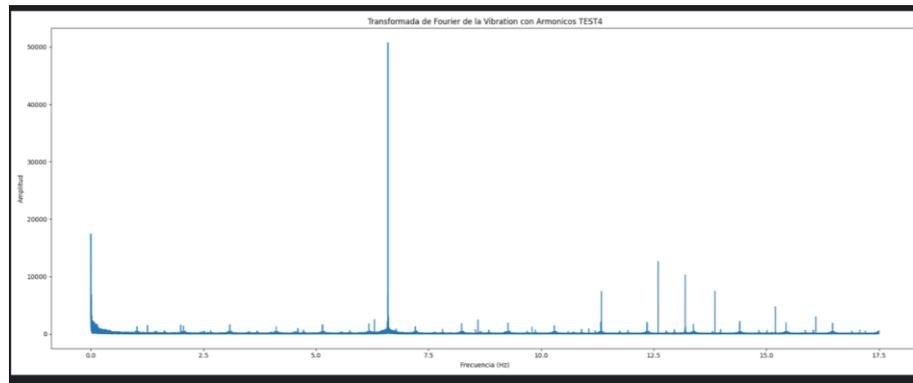
*Comparación con señal real en Prueba 4*



En la figura 3.1.19 se presentan las comparaciones con la señal real con la señal de falla que un 87% de las vibraciones deberían ser etiquetadas como nivel B. Esto quiere decir que hay una diferencia de 7% entre el diagnóstico supuesto y el real con la señal alterada.

**Figura 3.1.21**

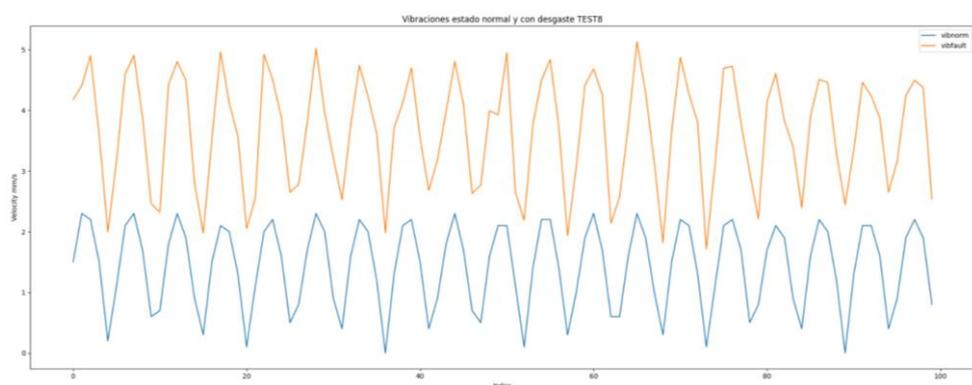
*Espectro de frecuencia completo en Prueba 4*



En la Figura 3.1.21 demuestra que hasta esta Prueba 4 se tuvo al armónico más grande con una magnitud aún menor a la cuarta parte de la magnitud a frecuencia fundamental, no presentando una situación de falla y que puede seguir trabajando. Así como los expresaba la norma ISO y la escala definida para el análisis con Node RED tienen un diagnóstico que coincide con lo visible en el espectro de frecuencia.

**Figura 3.1.22**

*Comparación con señal real en Prueba 8*

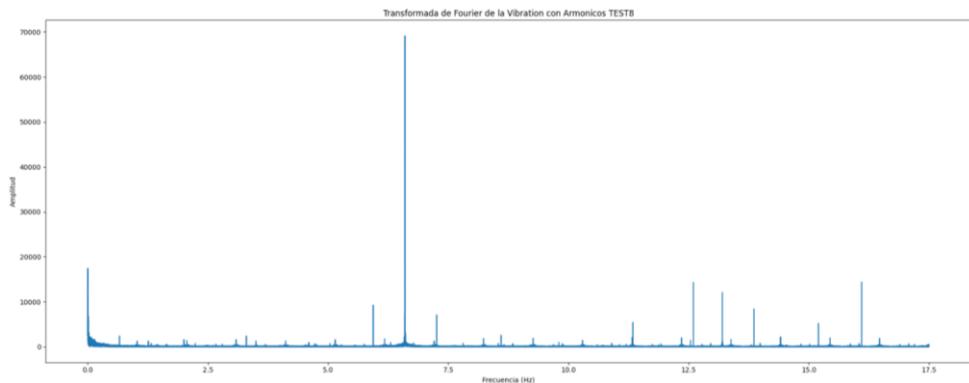


Para la Prueba 8, se inyectaron tanto armónicos en la vecindad de 12Hz como en la vecindad de la fundamental de 6Hz, como parte de la representación de un desgaste en el soporte y en los elementos rodantes. Las dos señales ya comienzan a distanciarse debió al

incremento en la amplitud de las vibraciones, mostrando mas soltura y desplazamiento en la Figura 3.1.22.

**Figura 3.1.23**

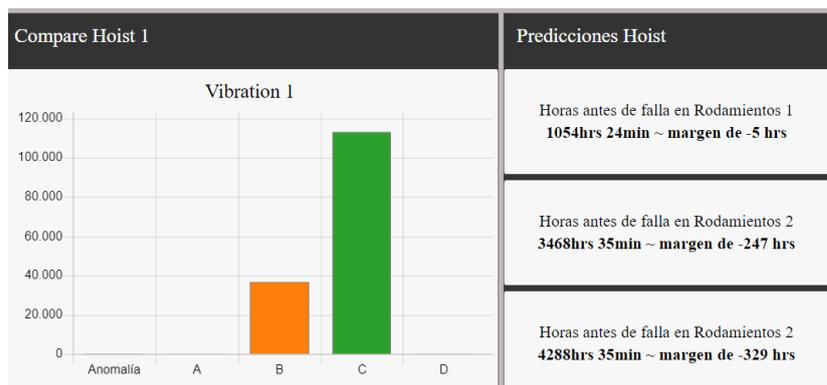
*Espectro de frecuencia real en Prueba 8*



Con la señal real para la Prueba 8 se tuvo que aún no aparecen armónicos relevantes en la Figura 3.1.23 para definir la presencia de una falla que necesite mantenimiento en un tiempo cercano, y se tiene que las vibraciones esta un 28.4% en nivel B y 71.6% en nivel C.

**Figura 3.1.24**

*Proporción de Magnitudes de Vibraciones en Rodamiento 1 en Prueba 8*

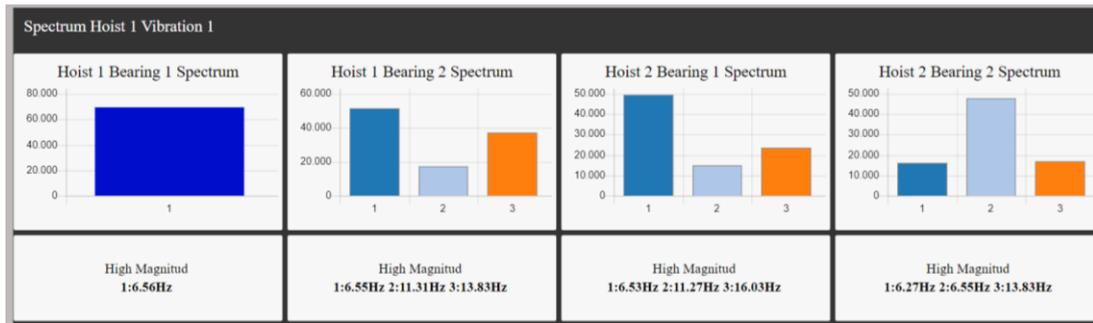


La separación vista con la señal real y la de fallas fue representada en el análisis como el caso de que la mayoría de las vibraciones están en nivel C, como se observa en la Figura 3.1.24, correspondiente al diagnóstico de que el sistema motor-rodamiento ya no podrá operar

por un largo periodo, por ellos de las horas de los 6 meses se pasa a poco más de 1 mes para necesitar mantenimiento.

**Figura 3.1.25**

*Panel de Espectro de Frecuencia en Prueba 8*

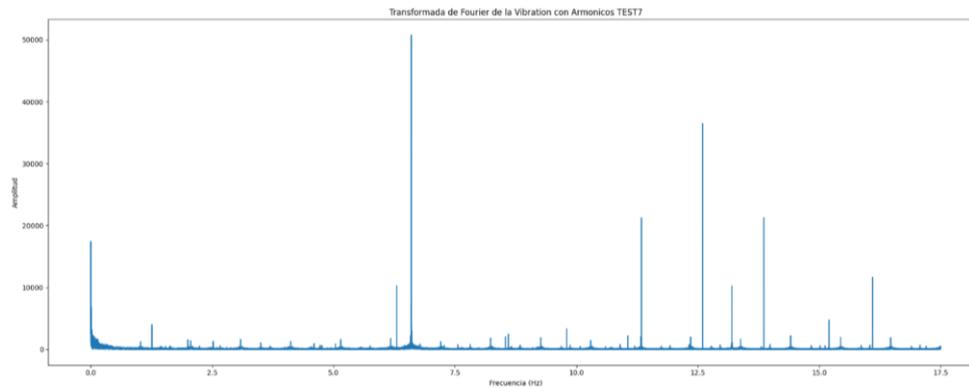


Lo interesante de esta prueba es que, aunque en el Rodamiento 1 del Motor 1 se tiene la necesidad de un mantenimiento en un tiempo cercano, la frecuencia fundamental sigue siendo la más influyente, en esta prueba ocurrió que el mayor daño se presentó a frecuencias cercanas a 6 Hz, es decir, por problemas con la estructura que soporta al sistema motor-rodamiento.

En la respuesta del programa de Node RED en la Prueba 8 se tuvo que un 26.15% este en nivel B y un 73.85% en nivel C, es decir, una diferencia de menos de 3% de error entre el diagnóstico del modelo y la revisión directa de la señal real y la alterada.

**Figura 3.1.26**

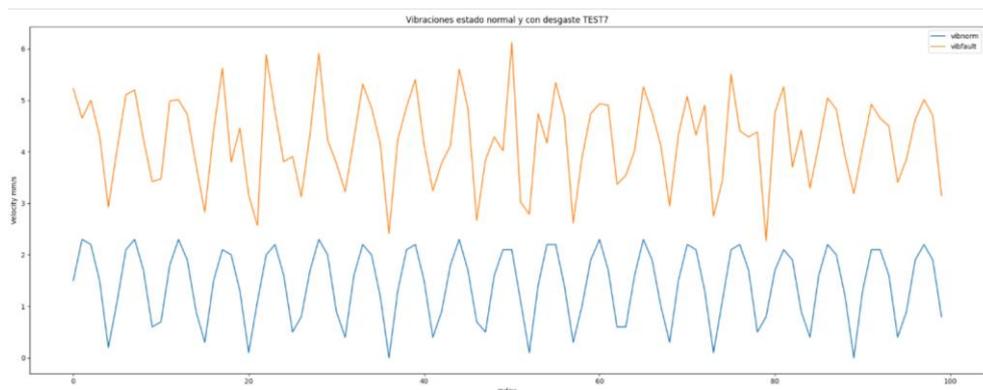
*Espectro real de frecuencia en Prueba 7*



En la Figura 3.1.26 se tiene un espectro que ya mostró armónicos con uno de ellos a una magnitud muy cercana a la fundamental, que representan un problema interno en los rodamientos y aquello corresponde a disparar altera de falla que necesita mantenimiento pronto, antes de que el armónico iguale o supere a la fundamental.

**Figura 3.1.27**

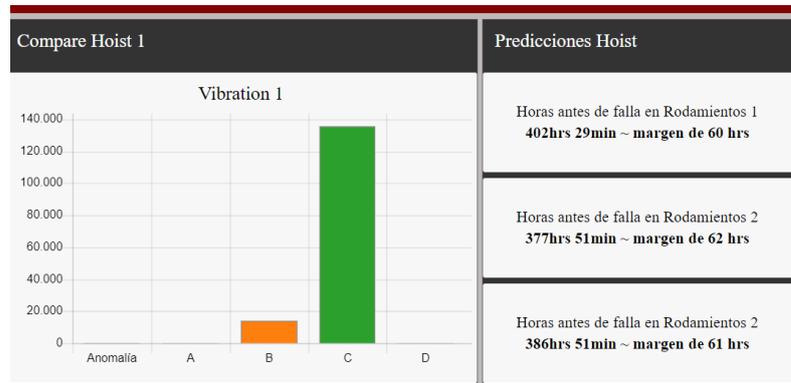
*Comparación con señal real en Prueba 7*



En la Figura 3.1.27 se observó que las vibraciones de falla ya coincidían con ser el doble de las vibraciones en estado óptimo y con perturbaciones notorias a lo largo del gráfico, comenzando a sobrepasar los límites y generar danos en otros elementos del sistema motor-rodamiento.

**Figura 3.1.28**

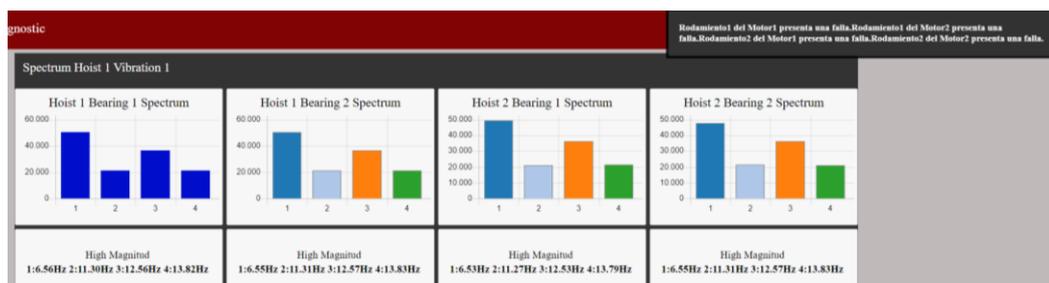
*Análisis de niveles de vibración en Prueba 7*



Para la Prueba 7 entonces se tiene un estado del Rodamiento donde el tiempo para agendar su próximo mantenimiento en unas 2 semanas, como esta en la Figura 3.1.28. A diferencia de la prueba anterior, pese a que son casos similares, la clave está en la proporción del nivel B, porque cuando todas las vibraciones estén en nivel C, ocurrirá que algunas vibraciones llegaran al nivel D progresivamente, y en esa situación, las vibraciones de nivel D pueden provocar otras averías, complicando el trabajo de mantener en buen estado al sistema motor-rodamiento. Por ello es importante anunciar la necesidad de un mantenimiento en un corto tiempo cuando los niveles A y B se quedan sin proporción o cuando los niveles C y D incrementan demasiado.

**Figura 3.1.29**

*Espectro de vibración de Motor1 con emulación de desgaste tras un tiempo prolongado.*



Como se aprecia en la figura 3.1.29, los armónicos generados se pronuncian con ganancias muy cercanas al doble de la frecuencia fundamental, esto sucedió cuando los niveles de vibración aumentaron al nivel C y D, y como se puede apreciar, dado que se dispararon las alteras de fallo del rodamiento 1 del motor 1, demostrando que, siguiendo la norma ISO 10816 de vibraciones si es posible detectar cuando se presentan las fallas, en este caso por problemas mecánicos internos. En esta prueba 7 se tuvo que nivel de 9,98% en B para la comparación real y 9.6% para la comparación con el modelo, mostrando un error del 0.38%, que según lo revisado en las otras pruebas expuestas y demás que se realizaron, un total de 30 se tiene que el modelo se vuelve más preciso mientras mas grande sea el agravio de la falla, pero sin tener mal rendimiento con perturbaciones pequeñas de fallas que van evolucionando a lo largo del tiempo.

### 3.1.5. Reportes con Google

Figura 3.1.30

Reportes almacenados en base de datos en la nube.

	Fecha	Hora	State11	State12	State21	State22	Int11	Int12	Int21	Int22	F11	F12
▶	1/8/24	12:40:45	OK	OK	OK	OK	93	92	90	94	FUNDAMENTAL	FUNDAMENTAL
	1/8/24	16:22:50	OK	OK	OK	OK	93	92	90	94	FUNDAMENTAL	FUNDAMENTAL
	1/8/24	22:00:15	OK	OK	OK	OK	93	92	90	94	FUNDAMENTAL	FUNDAMENTAL
	1/9/24	06:33:22	OK	OK	OK	OK	93	92	90	94	FUNDAMENTAL	FUNDAMENTAL
	2/9/24	02:11:22	OK	OK	OK	OK	92	91	90	94	FUNDAMENTAL	FUNDAMENTAL
	2/22/24	13:08:45	OK	OK	OK	OK	91	91	90	94	FUNDAMENTAL	FUNDAMENTAL
	3/10/24	18:30:22	OK	OK	OK	OK	90	89	90	94	FUNDAMENTAL	FUNDAMENTAL
	5/25/24	09:15:34	OK	WARNING	OK	OK	89	85	90	94	FUNDAMENTAL	1/2-HARMONIC
	6/12/24	12:00:34	OK	WARNING	OK	OK	89	85	90	94	FUNDAMENTAL	1/2-HARMONIC
	7/12/24	12:00:34	WARNING	WARNING	OK	OK	85	84	90	94	1/2-HARMONIC	1/2-HARMONIC
	7/23/24	11:00:34	WARNING	WARNING	OK	OK	85	84	90	94	1/2-HARMONIC	1/2-HARMONIC
	8/1/24	11:00:34	WARNING	WARNING	OK	OK	85	84	90	94	1/2-HARMONIC	1/2-HARMONIC
	8/1/24	11:00:34	WARNING	WARNING	OK	OK	85	84	90	94	1/2-HARMONIC	1/2-HARMONIC
	8/1/24	11:00:34	WARNING	WARNING	OK	OK	85	84	90	94	1/2-HARMONIC	1/2-HARMONIC

Siguiendo el flujo de reportes, después de cada análisis automático o manual, se registra el estado general de los rodamientos, porcentaje de estado y la frecuencia con mayor aporte en el espectro, visible en la Figura 3.1.30. El registro de los reportes fue realizado en cuestión de

un segundo y sin fallo de envíos con la conexión remota a internet, pese a no tener como tal a la Raspberry con acceso constante a la Plataforma de Google Cloud.

**Figura 3.1.31**

*Reporte de decaimiento del estado del Rodamiento 1 del Motor 1 con pruebas iniciales*



**Figura 3.1.32**

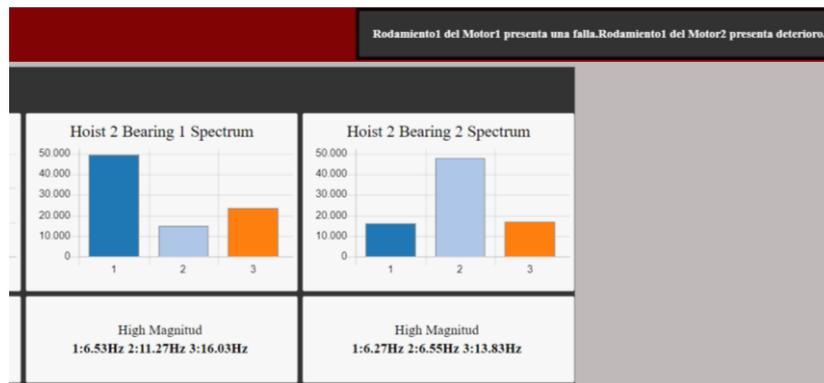
*Reporte del sistema después de pruebas variadas*



El histórico de los reportes, como se muestra en las Figuras 3.1.31 y 3.1.32, de igual forma se actualizo transcurrido de 2 a 5 segundos mientras hace la solicitud y presenta la información en el panel, mostrando efectivamente la evolución del deterioro del estado de los rodamientos registrado por cada análisis.

**Figura 3.1.33**

*Avisos de presencia de deterioro y de estado de fallo en el Rodamiento 1 del Motor1.*



De igual manera, posterior a la escritura y a la par de la lectura de la nube de Google, se tiene la advertencia con el mensaje del diagnóstico actual acompañado de la lectura por voz de este, estos dos avisos aparecieron en todas las ventanas del Panel, como se observa en la Figura 3.1.33.

### 3.2. Costos

Tabla 3.2-1

*Costos del proyecto*

Material/Equipo	Función	Costo
Google MySQL	Almacenamiento de Reportes	\$ 130.12
Raspberry Pi 4 B	Gestión y Predicción de Datos	\$ 100.00
Node RED	Programación y ejecución de Scripts (Licencia Gratuita)	\$ -
Servidor MySQL	Almacenamiento de Variables	\$ 50.00
Fuente de Poder 12V	Fuente de alimentación para Raspberry	\$ 10.00
Carcasa de Aluminio	Protección y Montura de la Raspberry	\$ 25.00
Cable RJ45	Medio de comunicación	\$ 5.00
Pantalla HDMI	Interfaz Gráfica para Raspberry	\$ 50.00
Teclado	Periférico para Raspberry	\$ 12.00
Ratón	Periférico para Raspberry	\$ 7.00
Subtotal		\$ 389.12
Diseño	Programación, Pruebas y Montaje	\$ 1400.00
Inversión		\$ 1789.12

La Tabla 3.2-1 explica el detalle del precio por cada elemento presente en el proyecto, siendo los mas importantes la instancia de Google MySQL, la palca Raspberry Pi 4, el servidor de Node RED y el Servidor MySQL. El costo de la nube de Google es anual.

## Capítulo 4

### 4.1. Conclusiones y recomendaciones

Los resultados del sistema bajo prueba con distintas perturbaciones demostraron que se pueden captar los comportamientos en cuanto a elevación de la amplitud de las vibraciones para definir y la presencia de armónicos como información útil para diagnóstico de fallas y aviso de la necesidad progresiva de un mantenimiento. Adicional, la integración de los elementos IOT que son la instancia de Node RED, el servidor MySQL y el servidor SQL en la nube no tuvo inconvenientes de incompatibilidad, retardos o accesibilidad, cumpliendo con el uso de credenciales y autorizaciones para la seguridad entre redes.

La importancia de este proyecto radica en algunos de los puntos que cimienta la Industria 4.0 como la integración de inteligencia artificial con los modelos ensemble para identificación de patrones de regresión, base de datos en las nubes con Google Cloud SQL, integración de dispositivos IOT para la interconectividad como la placa Raspberry, y la captura, recopilación y análisis de unos datos de interés para apoyar la toma de decisiones con el sistema de Node RED. Otro aspecto es la eficiencia en la utilización de recursos de la planta tiene actualmente, porque el sistema CMS hace un monitoreo masivo de la grúa STS, pero solo presenta información y su histórico no es procesable en otro programa a más de su panel CMS, entonces, el diseño usa los recursos que se usan netamente para visualización para un mejor histórico y un análisis de esos datos. Por último, este diseño pone a cuestionar la idea parcialmente errada de que el mantenimiento predictivo, y en general el análisis de datos, requieren equipos computacionales especiales, de enorme procesamiento y nivel tecnológico, cuando la salida de nuevos dispositivos con su apartado informático expande las posibilidades en el mercado.

#### **4.1.1. Conclusiones**

- Los dos análisis mostrados en el proyecto permiten una interpretación de la evolución del estado del rodamiento con una precisión del más del 90% dependiendo de agravio presente en la falla, es decir, la magnitud de los armónicos.
- El sistema de monitoreo permitió la revisión del estado del motor Hoist y sus magnitudes eléctricas de trabajo en tiempo real y estado de la integridad de sus componentes mecánicos con respuesta eficiente y fiable, sin cruce de información o ralentización en menos de 1 minuto.
- En la revisión de la literatura se lograron definir los indicadores y patrones para la identificación de fallos mecánicos en la estructura que soporta a los elementos rodantes y a estos, conforme se realiza trabajo por un largo periodo, mediante la revisión del espectro de frecuencia y la comparación del rendimiento actual de la maquina con su trabajo cuando es recientemente reacondicionada.
- El protocolo Modbus en su versión TCP demostró ser de respuesta eficiente para un monitoreo continuo a frecuencia de 35Hz para la recolección de datos de la planta sin necesidad de una configuración compleja en la red local, pero limitándose a la cantidad de datos que se maneja para el análisis vibratorio de los rodamientos del Hoist.

#### **4.1.2. Recomendaciones**

- Uso de otros equipos de procesamiento con mayor alcance o flexibilidad para expandir almacenamiento y RAM, si bien se puede recurrir directamente a una PC industrial, es preferible testar las opciones de libre reprogramación en el mercado para la personalización del proyecto según las necesidades del cliente, tipo de planta y el comportamiento captado.
- Adicional al punto anterior, el uso de equipos con mayor rendimiento incluye la posibilidad de aprovechar al máximo las velocidades del protocolo Modbus TCP que en el servidor de Node RED llegaría hasta los 1kHz, ampliando el espectro de frecuencia para diagnósticos de problemas internos más minuciosos como la parte eléctrica del motor.
- El uso de protocolo OPC sería un cambio necesario si el monitoreo se haría principalmente lejos de la planta, el servido OPC que usa Node RED va a velocidad similares a la de Modbus TCP, solo que tendrían que revisarse las condiciones del punto de conexión a internet para el rendimiento.
- Complementar el sistema con análisis de índices de desempeño, si bien la revisión del espectro y amplitudes de la vibración ayuda a comprender los desperfectos mecánicos, hay otros factores como ambientales y eléctricos que pueden revisarse a partir de KPIs para hacer reportes del rendimiento de la panta a la vez que sus tiempos de fallo.

## Referencias

- Vázquez, T. S. (2014). *Lo secreto del mantenimiento industrial*. Palibrio.
- Torres, L. (2005). *Mantenimiento su implementación y gestión*. Universitas, Primera Edición, Libro Electrónico.
- Pinto, A. K. (2002). *Mantenimiento: función estratégica*. Qualitymark Editora Ltda.
- Fernández Álvarez, J. M. (2015). *Técnicas de mantenimiento predictivo: análisis de vibraciones, termografía y análisis de aceite*. Madrid, España: Díaz de Santos.
- García Fernández, F. (2016). *Mantenimiento predictivo: una guía práctica*. Madrid, España: Diaz de Santos.
- NEMA (National Electrical Manufacturers Association). (2022). *NEMA ICS 1-2012: Controladores lógicos programables (PLC)*. National Electrical Manufacturers Association.
- Jones, J. (2015). *Inteligencia: Un enfoque integrador*. Madrid, España: Pearson Educación.
- Millán, J. J. (2023). *Machine Learning para principiantes*. Madrid, España: Marcombo.
- López de Mántaras, R. (2023). *Inteligencia Artificial: Una introducción*. Madrid, España: Cengage Learning.
- Pathak, R., & Tiwari, S. P. (2018). *Inteligencia Artificial: Una visión general*. *International Journal of Computer Applications*, 175(1), 2-6.
- Kulkarni, A., & Joshi, S. (2015). *Inteligencia Artificial: Conceptos y Técnicas*. Madrid, España: Pearson Educación.

- Dávalos, J.; Vásquez, J. (2022). *Determinación del estado de falla de motores eléctricos a partir del análisis de ruidos utilizando técnicas de Aprendizaje Profundo. [Tesis de Grado]. Escuela Superior Politécnica del Litoral*
- Odar, B. (2023). *Análisis y detección de fallas en motores eléctricos aplicando algoritmos de inteligencia artificial (Tesis para optar el título de Ingeniero Mecánico-Eléctrico). Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Mecánico-Eléctrica.*
- Jado Moreno, V. (2022, May 20). *Mantenimiento predictivo en transformadores de potencia y de distribución eléctrica (Treball Final de Grau). UPC, Escola d'Enginyeria de Barcelona Est, Departament d'Enginyeria Elèctrica. Retrieved from <http://hdl.handle.net/2117/370708>*
- Pillajo Ñauñay, Martha Cecilia. Vivanco Correa, Jessica Andrea (2022). *Sistema de mantenimiento predictivo para máquinas eléctricas en procesos industriales en el escenario de la industria 4.0. Maestría en Electrónica y Automatización Mención Redes Industriales. Universidad de las Fuerzas Armadas ESPE. Extensión Latacunga.*
- Scheffer, C., & Girdhar, P. (2004). *Practical machinery vibration analysis and predictive maintenance. Elsevier.*
- Vibration Institute. (2019). *Vibration analysis for predictive maintenance. Vibration Institute.*
- Rao, S. S. (2011). *Vibrations of continuous systems (2nd ed.). New York, NY: Wiley.*
- Harris, C. M. (2002). *Mechanical vibrations: Theory and applications (5th ed.). New York, NY: Wiley.*

- Pinheiro, A. A., Brandao, I. M., & Da Costa, C. (2019). *Vibration analysis in turbomachines using machine learning techniques*. *European Journal of Engineering Research and Science*, 4(2), 12-16. <https://doi.org/10.24018/ejers.2019.4.2.1128>
- Kim, H.-J. (2018). *Vibration analysis of rotating machinery using machine learning techniques*. New York, NY: Springer.
- Smith, R. L., Brown, M. J., & Jones, D. M. (2016). *Vibration analysis for machine condition monitoring: A practical guide*. New York, NY: Springer.
- Grajales, J., Ramírez, J., & Cadavid, D. (2004). *Efectos de los armónicos en los motores de inducción: una revisión*. *Revista Facultad de Ingeniería Universidad de Antioquia*, (31), 116-123.
- González, J. A. (2015). *Procesamiento de señales digitales: teoría y aplicaciones*. Madrid, España: Pearson Educación.
- Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984) *Classification and Regression Trees*. Chapman and Hall, Wadsworth, New York.
- Mendoza Morán, I. O., & Orellana Uruchima, G. X. (2013). *Diseño e implementación de un sistema de control electrónico para una máquina balanceadora Geodyna 40/40 para neumáticos de 12 a 16 pulgadas (Bachelor's thesis)*.
- Organización Internacional de Normalización (ISO). (1990). *ISO 1925:1990 (E/F) - Balancierung von rotierenden Maschinen - Teil 1: Begriffe und Grundsätze*. [En línea]. Recuperado de <https://es.scribd.com/document/638083656/ISO-1925>

## Apéndice A

### Cálculo de Frecuencias de vibraciones en Rodamientos

6326/C3

$$D = 280 \text{ [mm];}$$

$$d = 130 \text{ [mm];}$$

$$B = 58 \text{ [mm];}$$

$$D_{pw} = D + d - B = 352 \text{ [mm];}$$

$$D_{we} = B = 58 \text{ [mm];}$$

$$n = 900 \text{ [RPM];}$$

$$z = 16;$$

$$\alpha = 30 \text{ [grados]}$$

$$BPFO = \frac{n}{60} * z * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (1)$$

$$BPFO = \frac{900}{60} * 16 * \frac{352 - 58 \cos(30)}{2(352)} = 102.8 \text{ [Hz]}$$

$$BPF1 = \frac{n}{60} * z * \frac{D_{pw} + D_{we} \cos(\alpha)}{2D_{pw}} \quad (2)$$

$$BPF1 = \frac{900}{60} * 16 * \frac{352 + 58 \cos(30)}{2(352)} = 137.2 \text{ [Hz]}$$

$$BSF = \frac{n}{60} * \frac{D_{pw}^2 - D_{we}^2 \cos(\alpha)}{D_{we} * D_{pw}} \quad (3)$$

$$BSF = \frac{900}{60} * \frac{(352)^2 - (58)^2 \cos(30)}{58 * 352} = 88.88 \text{ [Hz]}$$

$$BPF = 2 * BSF \quad (4)$$

$$BPF = 2 * 88.88 = 177.77 \text{ [Hz]}$$

$$FTF = \frac{n}{60} * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (5)$$

$$FTF = \frac{900}{60} * \frac{352 - 58 \cos(30)}{2(352)} = 6.425 [Hz]$$

### 6322/C3

$$D = 240 [mm];$$

$$d = 110 [mm];$$

$$B = 50 [mm];$$

$$D_{pw} = D + d - B = 300 [mm];$$

$$D_{we} = B = 50 [mm];$$

$$n = 900 [RPM];$$

$$z = 16;$$

$$\alpha = 30 [grados]$$

$$BPFO = \frac{n}{60} * z * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (1)$$

$$BPFO = \frac{900}{60} * z * \frac{300 - 50 \cos(30)}{2(300)} = 102.6 [Hz]$$

$$BPF1 = \frac{n}{60} * z * \frac{D_{pw} + D_{we} \cos(\alpha)}{2D_{pw}} \quad (2)$$

$$BPF1 = \frac{900}{60} * z * \frac{300 + 50 \cos(30)}{2(300)} = 137.4 [Hz]$$

$$BSF = \frac{n}{60} * \frac{D_{pw}^2 - D_{we}^2 \cos(\alpha)}{D_{we} * D_{pw}} \quad (3)$$

$$BSF = \frac{900}{60} * \frac{(300)^2 - (50)^2 \cos(30)}{50 * 300} = 87.825 [Hz]$$

$$BPF = 2 * BSF \quad (4)$$

$$BPF = 2 * 87.825 = 175.65 [Hz]$$

$$FTF = \frac{n}{60} * \frac{D_{pw} - D_{we} \cos(\alpha)}{2D_{pw}} \quad (5)$$

$$FTF = \frac{900}{60} * \frac{300 - 50 \cos(30)}{2(300)} = 6.41 [Hz]$$

### **Cálculo de Frecuencias de muestreo para los rodamientos**

#### **6326/C3**

$$FTF = 6.425 [Hz];$$

$$FTF_{H2} = 12.85 [Hz];$$

$$F_T > 2 * Fn \quad (1)$$

$$F_T > 2 * FTF_{H2} = 25.7Hz$$

#### **6322/C3**

$$FTF = 6.41 [Hz];$$

$$FTF_{H2} = 12.825 [Hz];$$

$$F_T > 2 * Fn \quad (1)$$

$$F_T > 2 * FTF_{H2} = 25.65Hz$$

### **Cálculo de Velocidad de transmisión de datos**

#### **11 registros a 35Hz**

$$Peso_{word} = 16[bits];$$

$$Cantidad\ de\ datos\ por\ segundo = 35 [datos];$$

$$Speed = \#Registros * Peso_{word} * Cantidad\ de\ datos\ por\ segundo \quad (1)$$

$$Speed = 6160\ bps = 6kpbs$$

#### **20 registros a 1Hz**

$$Peso_{word} = 16[bits];$$

$$Cantidad\ de\ datos\ por\ segundo = 1 [datos];$$

$$Speed = \#Registros * Peso_{word} * Cantidad\ de\ datos\ por\ segundo \quad (1)$$

$$Speed = 320 \text{ bps}$$

**30 registros a 1000Hz**

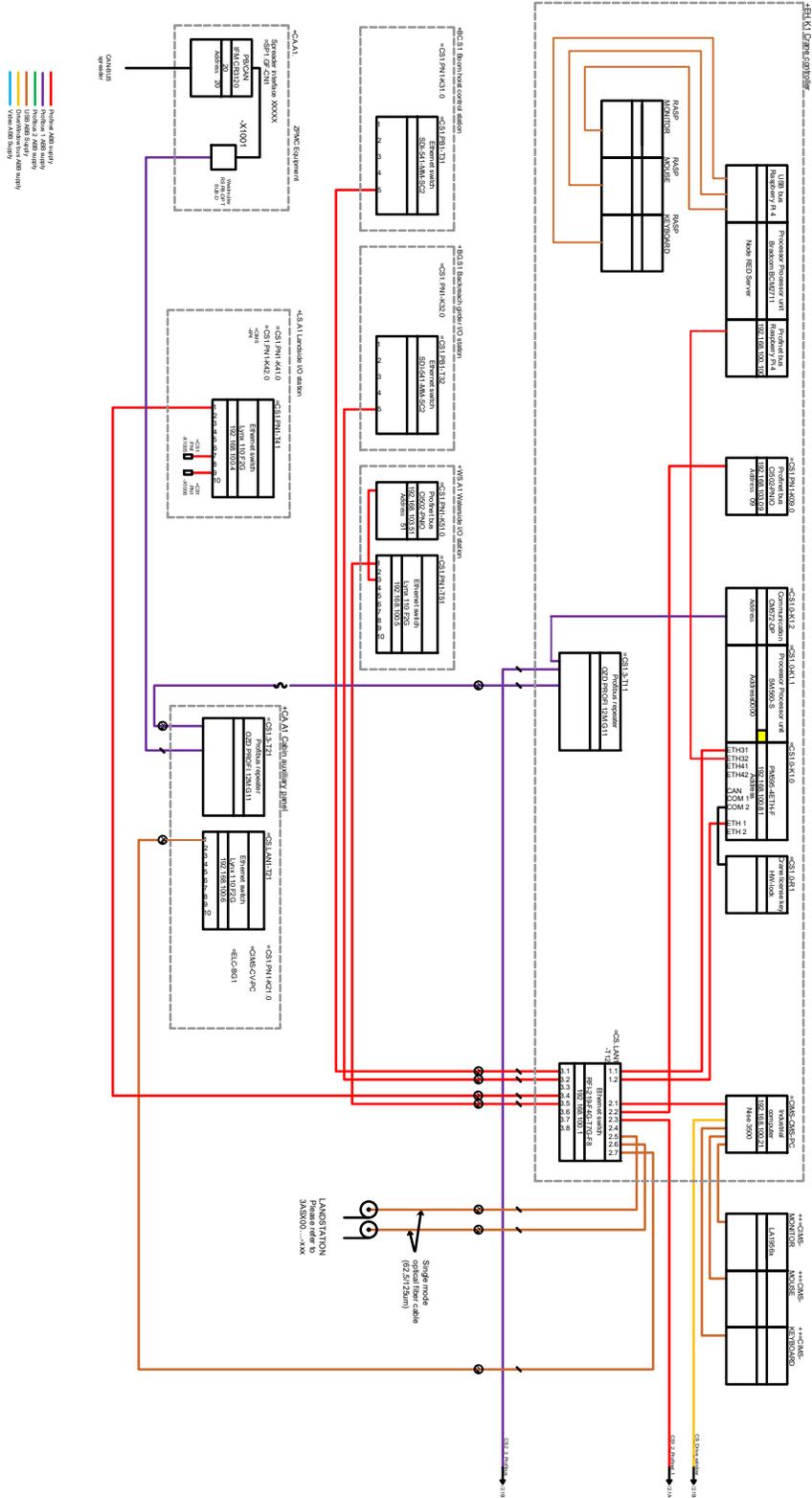
$$Peso_{word} = 16[\text{bits}];$$

$$Cantidad \text{ de datos por segundo} = 1000 [\text{datos}];$$

$$Speed = \#Registros * Peso_{word} * Cantidad \text{ de datos por segundo} \quad (1)$$

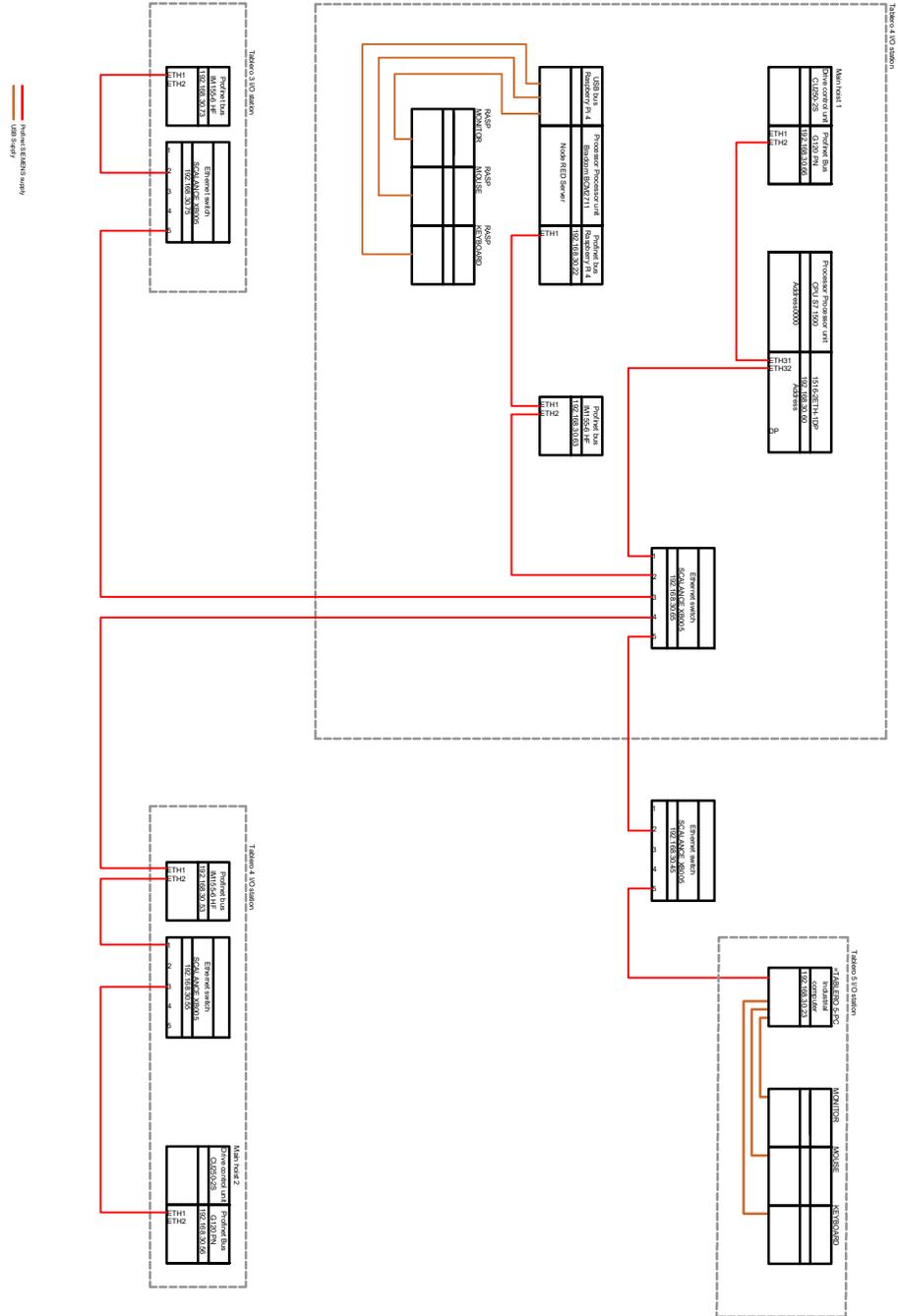
$$Speed = 480000 \text{ bps} = 469 \text{Kpbs}$$

# Diseño de red de dispositivos ABB con Raspberry integrada





# Red implementada de dispositivos SIEMENS en el laboratorio de Sistemas Complejos



## Apéndice B

### Código de entrenamiento RandomForest

```
import random
import math
import numpy as np
import joblib
import pandas as pd

from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, mean_absolute_percentage_error

from sklearn.preprocessing import StandardScaler

df = pd.DataFrame(msg['payload'])
df.loc[df['Trolley_Position'] > 100, 'Trolley_Position'] = 40
df.loc[df['Hoist_Position'] > 100, 'Hoist_Position'] = 40
df['Hoist_Position'] = df['Hoist_Position'].diff()
df['Hoist_Position'].iloc[0] = 0
df['Trolley_Position'] = df['Trolley_Position'].diff()
df['Trolley_Position'].iloc[0] = 0

# Obtener los arreglos A y B desde el DataFrame
entrada = df[['Voltage_DC_Motor_1', 'Current_Motor_1', 'Hoist_Position',
'w', 'Trolley_Position']]
hoist1 = entrada.iloc[1:15000,:].to_numpy()
#print(hoist1)
h1v1 = df['Motor_1_Vibration_1'][1:15000].to_numpy().reshape(-1, 1)

# Comprueba que las longitudes sean las mismas
if len(hoist1) != len(h1v1):
    raise ValueError("Las longitudes de los arreglos de entrada deben ser iguales.")

# Normaliza los datos
```

```

scaler = StandardScaler()
hoist1_n = scaler.fit_transform(hoist1)

# Obtener los datos de entrada (hoist1_n) y la variable objetivo (h1v1)
X = hoist1_n
y = h1v1.flatten() # Asegurarse de que y sea un array 1D

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
param_dist_gb = {
    'n_estimators': np.arange(80, 1000, 5),
    'max_depth': [None,200,1000,50000,100000],
    'min_samples_split': [2,4,6],
    'min_samples_leaf': [1, 2, 4,5,6],
}
mae = 100
porc_coincidentes = 0
scores = 0
while mae >0.05:

    # Inicializar el modelo RandomForestRegressor
    modelo1vib1 = RandomForestRegressor(criterion='squared_error',
warm_start=True)
    random_search_gb = RandomizedSearchCV (
        modelo1vib1, param_distributions=param_dist_gb, n_iter=10,
        scoring='neg_mean_squared_error', cv=5,
    )

    # Entrenar el modelo
    random_search_gb.fit (X_train, y_train)
    coincidentes = 0
    # Realizar predicciones en el conjunto de prueba
    model = random_search_gb. best_estimator_
    y_pred = model. predict(X_test)
    for yp, yt in zip (y_pred, y_test):
        if abs (yp - yt) < 0.2:
            coincidentes += 1

```

```

porc_coincidentes = 100 * coincidentes / len(y_test)

# Calcular el error cuadrático medio (MSE)
mse = mean_squared_error (y_test, y_pred)
mae = mean_absolute_error (y_test, y_pred)
scores = model. score(X_test,y_test)

# Si deseas guardar el modelo entrenado para su uso posterior, puedes
hacerlo con joblib
joblib. dump(model, 'h1v1.joblib')

msg["payload"] = "Modelo de vibracion 1 del motor 1 entrenado con "+
str(porc_coincidentes) +" % de exito"
return msg

```

### Fragmento de generación de armónicos

```

times = np. arange(0,len(df['Hoist_Position'])/35,1/35)
freqs = 6.6/2
w =2*np.pi*freqs
fsin = np.cos(w*times)
velocidad_primitiva = np. sqrt((fsin**2))
df['Oscilacion'] = velocidad_primitiva
entrada = df [['Voltage_DC_Motor_1', 'Current_Motor_1', 'Hoist_Position',
'Load', 'Trolley_Position','Oscilacion']]
times = np. arange(0,len(velocidad_primitiva)/35,1/35)
freqs = 12.6/2
w =2*np.pi*freqs
fsin = 1.0*np.cos (1.5*w*times)
fsin2 =0.8* np.cos (1.1*w*times)
fsin3 =1.0* np.cos(w*times)
fsin4 =0.6* np.cos (0.9*w*times)
fsin5 =0.4* np.cos (0.5*w*times)
armonico =
sqrt((fsin**2)+(fsin2**2)+(fsin3**2)+(fsin4**2)+(fsin5**2))[:])
freqs = 6.6/4
w =2*np.pi*freqs
fsin = 0.2*np.cos (1.5*w*times)

```

```

    fsin2 =0.6* np.cos (1.1*w*times)
    fsin3 =1.0* np.cos(w*times)
    fsin4 =0.7* np.cos (0.9*w*times)
    fsin5 =0.4* np.cos (0.5*w*times)
    armonico                                     +=
sqrt((fsin**2)+(fsin2**2)+(fsin3**2)+(fsin4**2)+(fsin5**2))[:]      np.
    freqs = 6.6/8
    w =2*np.pi*freqs
    fsin = 0.7*np.cos (1.5*w*times)
    fsin2 =0.3* np.cos (1.1*w*times)
    fsin3 =1.2* np.cos(w*times)
    fsin4 =0.4* np.cos (0.9*w*times)
    fsin5 =0.2* np.cos (0.5*w*times)
    armonico                                     +=
sqrt((fsin**2)+(fsin2**2)+(fsin3**2)+(fsin4**2)+(fsin5**2))[:]      np.
    freqs = 15/2
    w =2*np.pi*freqs
    fsin = 0.2*np.cos (1.5*w*times)
    fsin2 =0.1* np.cos (1.1*w*times)
    fsin3 =0.5* np.cos(w*times)
    fsin4 =0.4* np.cos (0.9*w*times)
    fsin5 =0.2* np.cos (0.5*w*times)
    armonico                                     +=
sqrt((fsin**2)+(fsin2**2)+(fsin3**2)+(fsin4**2)+(fsin5**2))[:]      np.

```

## Código de conexión y generación de queries con Google Cloud

```

import pandas.
from google.cloud.sql.connector import Connector
import sqlalchemy.
import pymysql.
from google.oauth2 import service_account
#from google.cloud import spanner
import json.
dict_credentials = {
    "type": "service_account",
    "Project_id": "rising-artifact-403901",
    "Private_key_id": "79482940f6b11700c9fd54fb4c5003dbfe102d07",

```

```
        "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEVaIBADANBgkqhkiG9w0BAQEFAASCbKYwggSiAgEAAoIBAQCCLwGtel9gIAL6x\nnaim9u0AfQhJz4
meB+ZEUGrV7hFr2/AkngtovAwo/bu5MyTVKI7iThQ8ADWGHFI1d\n2n9lC32Rps8zMH7ZJdQS2QWPhARX
Gr9l4PGrIYHOw5A8Ycih8V2zztBr4XZ9SsHb\nnst1J1BCnvxmuElXoPPTwZTaIcOT+ucCD218DLF31btA
iKLw6f29wCMEi7bdaHZmc\nnpAvbgt2FJ8FuzK3+gS/49HrYAg9G1MX2utNZs2DH9Rr+j7n5xWnYR6Ym0h
kRaL1A\nnYqYJec7Xg8xXp0cug3qL+xPSMwdoNbvKQxJ8CMVrTU9MK2HTBujk3cWhh4obNxYc\nngEV6byg
JAgMBAAECggEABoUrZ+tqXzKfxmPvSfSI060ddZYqQXcqXKduw4fuUo2p\nnw60o0X9JJXKAgxaez4m2co
jdDUyShfc11/DPdssVcwhq0q/UFRNKQqfvCAVr8Fb\nnYrPJsrP1LCIVrauGQn/8tjHqVC0IdV9dMhSN8
dmHZBGCGjatdBC8TnR6bzKNNzvT\nnugPu7dpPP0o1rqP/4034500khPo35p7TT/JGZxrMldCaKK2wkjzc
mxIN79+PBVaX\nnx1RrsC1xr6h2hmfukJugP2luSMMdH/CIGh1PoiFhE7t1id+mB4IFIiKrYdesfKM\nnw
/Hm+RheS1ypOepvaz8purB4TEUjJC22kCt4NssIcQKBgQDEzRXm73X9tutfQsuM\nnm5PSJt6E8/3TfSae
Oerauy1YLwrtQPdlz/mzrH0AQJHk7r8eX+dZwMCLbLkK+Nhd\nnXzT5g+xk+di01Y0bG/Wqe6K6V8LFxeB
H1kuQVuZ3rCdfGruCiZhUrZ4YIBxw9WUv\nnb44YLnQ43PwjP8L6b0Fu9BXHnQKBgQC1yisAwqXKTNkJfG
aojG3YJ8UePfiTvSUJ\nnKdNXxw2vCQr/cNpXH1y3N0yNuoFBA5VowyuGVqL3+1PN8Y3HWTczfKeUmrTCQ
TBo\nnqK0lfePZ9XXine+mU4FY14J3TdwicP+l220G7t3fd4pzHR3iNSbZ/Yh5FiOTcKzI\nnb9+ABBT0XQ
KBgCg5Trezy+b+gx8mVNo5Vj5GyYZdJE0csE74wYLRbRQwEUiq1aY0\nnmxfZMndhuzASiCealHv71+3HW
sATRzm64uX4y9Fbq6KODVg0H8Mnn/v8QggQSKYG\nnK82lpVdibw2n1h15LBRq+QMPVRuE7piCcYWJwSF6
MWLMQDLVmIcoBmeVAoGAHD8A\nndYrJBejw5DdvIAtJ2gc76pohbie6hHT6QK50Iv9NtSkM0/KEx3cRLwh
movpykZdw\nn6Nu3UL69Y8d/lxT2rcpVo5JzPDUy12A2Hxgzeh9YpNCTdw2MkH5iKDVx2dv6FQRE\nnr6eX
0+wYGBDTHt3RXpe1R9M3zeX5At/XI2X+94UCgYB+VH4z8/FrglydQfyRYk+2\nnpD7bjmpwht43JPxD+Iz
vWN+3XtLIcY6USAkpe0v2LpfoId4nQMXGHxALUUBhTva\nnychkT2N4Y33TIexsvoKvPcVCnXigyYSSkP
gm/j4zjRA/mBaQigCeJUH29QdAkxBW\nn2UJzJAddY4y0PC5/JVQoQ==\n-----END PRIVATE KEY---
--\n",
```

```
        "Client_email": "oficina@rising-artifact-403901.iam.gserviceaccount.com",
        "Client_id": "111386592271922208784",
        "Auth_uri": "https://accounts.google.com/o/oauth2/auth",
        "Token_uri": "https://oauth2.googleapis.com/token",
        "Auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
        "Client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/oficina%40rising-artifact-
403901.iam.gserviceaccount.com",
        "Universe_domain": "googleapis.com"
    }
}
```

```
#cloud_sql_credentials = service_account.
Credentials.from_service_account_file(r"C:\Users\rianm\Downloads\GOOGLE_CLOUD_CRE
DENTIALS\rising-artifact-403901-79482940f6b1.json")
```

```
cloud_sql_credentials = service_account.
Credentials.from_service_account_info(dict_credentials)
```

```
# Initialize Connector object
```

```
connector = Connector(credentials=cloud_sql_credentials)
```

```
# function to return the database connection
```

```
def getconn () -> pymysql.connections.Connection:
```

```
    conn: pymysql. connections.Connection = connector.connect(
```

```

        "Rising-artifact-403901:us-central1:cms-sts3db", # Cloud SQL
Instance Connection Name
        "pymysql",
        user="oficina",
        password="crane2024",
        db="cms",
        # ip_type=IPTypes.PUBLIC # IPTypes.PRIVATE for private IP
    )
    return conn.

```

```

# Create connection pool

```

```

pool = sqlalchemy.create_engine(
    "mysql+pymysql://",
    creator=getconn,
)
maintenance_test_date = msg['payload'] ['fecha']
maintenance_test_time = msg['payload'] ['hora']
maintenance_state11 = msg['payload'] ['state11']
maintenance_state12 = msg['payload'] ['state12']
maintenance_state21 = msg['payload'] ['state21']
maintenance_state22 = msg['payload'] ['state22']
maintenance_por11 = msg['payload'] ['por11']
maintenance_por12 = msg['payload'] ['por12']
maintenance_por21 = msg['payload'] ['por21']
maintenance_por22 = msg['payload'] ['por22']
maintenance_f11 = msg['payload'] ['F11']
maintenance_f12 =msg['payload'] ['F12']
maintenance_f21 = msg['payload'] ['F21']
maintenance_f22 = msg['payload'] ['F22']
msg = {'payload': {'values':{0:0, 1:1, 2:2, 3:3},
                  'state': {0:0, 1:1, 2:2, 3:3},
                  'frec': {0:0, 1:1, 2:2, 3:3}}
      }

```

```

# Insert statement

```

```

insert_stmt = sqlalchemy.text (
        "INSERT          INTO          cms_report          (Fecha,
Hora,State11,State12,State21,State22,Int11,Int12,Int21,Int22,F11,F12,F21,F22)" +

```

```

                                "VALUES"                                (:
Fecha, :Hora, :State11, :State12, :State21, :State22, :Int11, :Int12, :Int21, :Int22, :F11,
:F12, :F21, :F22)"
    )

    dic_stmt = {'Fecha': maintenance_test_date, 'Hora': maintenance_test_time,
               'State11': maintenance_state11, 'State12':
maintenance_state12, 'State21': maintenance_state21, 'State22': maintenance_state22
               , 'Int11': maintenance_por11, 'Int12': maintenance_por12, 'Int21':
maintenance_por21, 'Int22': maintenance_por22,
               'F11': maintenance_f11, 'F12': maintenance_f12, 'F21':
maintenance_f21, 'F22': maintenance_f22}

    with pool. connect() as db_conn:
        # Insert into database
        db_conn. execute(insert_stmt, parameters=dic_stmt)

        # query database
        result = pandas. DataFrame(db_conn.execute(sqlalchemy.text("SELECT *
from cms_report")).fetchall())

        # Commit transaction (SQLAlchemy v2.X.X is commit as you go)
        db_conn. commit()

        arreglo1 = result['Int11'].to_numpy ()

    msg['payload'] ['values'] [0] = arreglo1.tolist()

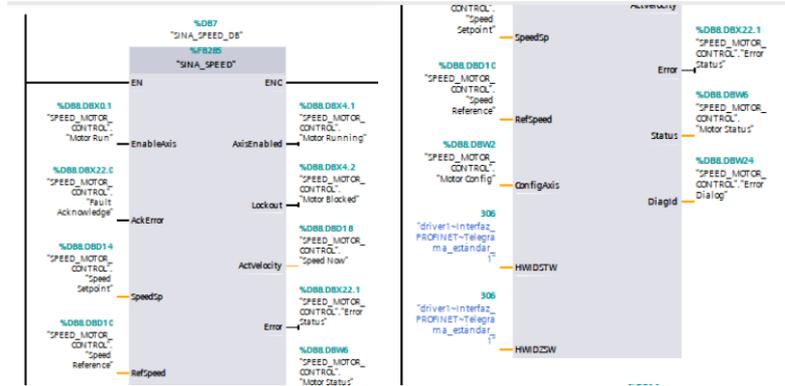
    #print(msg)

    connector. close ()

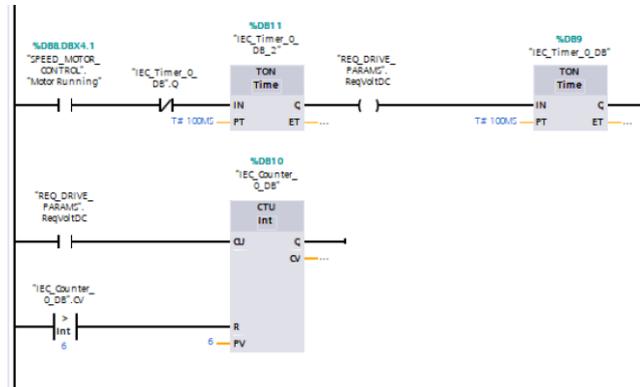
    return msg.

```

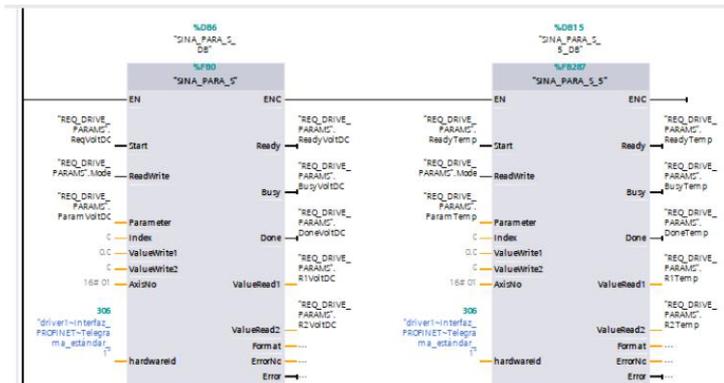
# Programación en TIA Portal

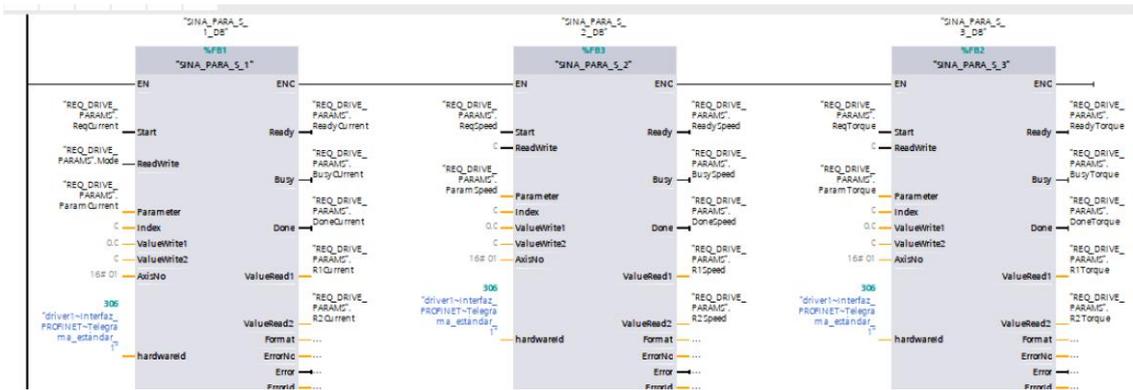


Bloque para controlar el giro del motor.

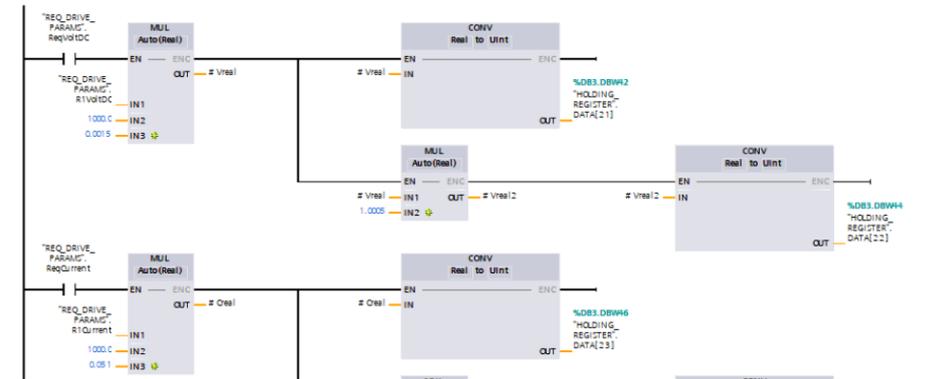


Secuenciador para generar solicitudes de lectura de parámetros.

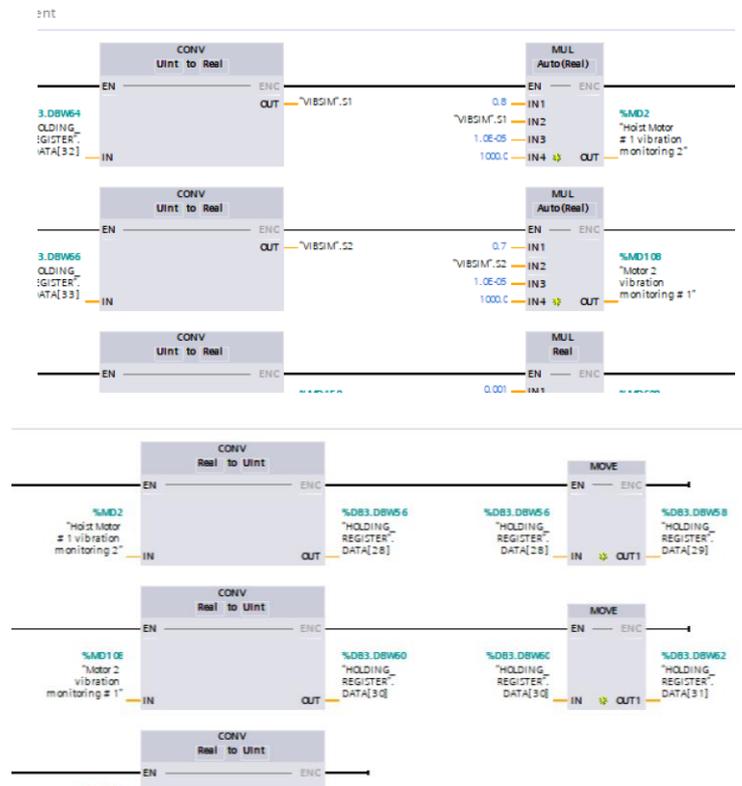




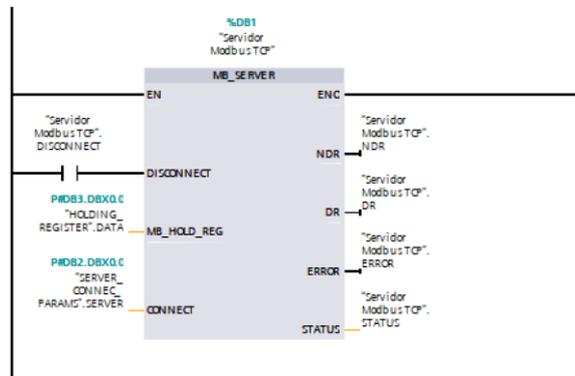
Bloques para extraer datos de parámetros.



Conversión de datos de voltaje, corriente, velocidad y torque a datos porcentuales en el Holding Register.



Generación de vibraciones y envío de estas al Holding Register.



### Servidor Modbus

HOLDING_REGISTER					
	Name	Data type	Offset	Start value	Ret
1	Static				
2	DATA	Array[0..50] of UInt	0.0		
3	DATA[0]	UInt	0.0	0	
4	DATA[1]	UInt	2.0	0	
5	DATA[2]	UInt	4.0	0	
6	DATA[3]	UInt	6.0	0	
7	DATA[4]	UInt	8.0	0	
8	DATA[5]	UInt	10.0	0	
9	DATA[6]	UInt	12.0	0	
10	DATA[7]	UInt	14.0	0	
11	DATA[8]	UInt	16.0	0	
12	DATA[9]	UInt	18.0	0	
13	DATA[10]	UInt	20.0	0	
14	DATA[11]	UInt	22.0	0	
15	DATA[12]	UInt	24.0	0	
16	DATA[13]	UInt	26.0	0	
17	DATA[14]	UInt	28.0	0	
18	DATA[15]	UInt	30.0	0	
19	DATA[16]	UInt	32.0	0	
20	DATA[17]	UInt	34.0	0	
21	DATA[18]	UInt	36.0	0	

### Bases de Datos: Holding Register

REQ_DRIVE_PARAMS				
	Name	Data type	Start value	
1	Static			
2	ReqVoltDC	Bool	false	
3	ReqTorque	Bool	false	
4	ReqCurrent	Bool	false	
5	ReqSpeed	Bool	false	
6	ReqTemp	Bool	false	
7	Mode	Bool	false	
8	ParamVoltDC	Int	79	
9	ParamTorque	Int	31	
10	ParamCurrent	Int	27	
11	ParamSpeed	Int	21	
12	ParamTemp	Int	35	
13	R1VoltDC	Real	0.0	
14	R2VoltDC	Dint	0	
15	R1Current	Real	0.0	
16	R2Current	Dint	0	
17	R1Speed	Real	0.0	
18	R2Speed	Dint	0	
19	R1Temp	Real	0.0	
20	R2Temp	Dint	0	
21	R1Torque	Real	0.0	

### Bases de Datos: Parámetros del G120

SERVER_CONNEC_PARAMS				
	Name	Data type	Offset	Start value
	Static			
2	SERVER	TCON_IP_v4	0.0	
3	InterfaceId	HW_ANY	0.0	64
4	ID	CONN_OUC	2.0	1
5	ConnectionType	Byte	4.0	11
6	ActiveEstablished	Bool	5.0	false
7	RemoteAddress	IP_V4	6.0	
8	ADDR	Array[1..4] of Byte	6.0	
9	ADDR[1]	Byte	6.0	0
10	ADDR[2]	Byte	7.0	0
11	ADDR[3]	Byte	8.0	0
12	ADDR[4]	Byte	9.0	0
13	RemotePort	UInt	10.0	0
14	LocalPort	UInt	12.0	502

### Bases de Dato: Conectividad con Servidor Modbus

SPEED_MOTOR_CONTROL				
	Name	Data type	Offset	Start value
	Static			
1	Motor Start	Bool	0.0	false
2	Motor Run	Bool	0.1	false
3	Motor Stop	Bool	0.2	false
4	Motor Config	Word	2.0	16#003F
5	Motor Enable	Bool	4.0	false
6	Motor Running	Bool	4.1	false
7	Motor Blocked	Bool	4.2	false
8	Motor Status	Word	6.0	16#0
9	Direction Change	Bool	8.0	false
10	Direction left	Bool	8.1	false
11	Direction right	Bool	8.2	false
12	Direction Now	Bool	8.3	false
13	Speed Reference	Real	10.0	1560.0
14	Speed Setpoint	Real	14.0	500.0
15	Speed Now	Real	18.0	0.0
16	Fault Acknowledge	Bool	22.0	false
17	Error Status	Bool	22.1	false
18	Error Dialog	Word	24.0	16#0

### Bases de Datos: Control de velocidad

## Apéndice C

### Código de captura del panel CMS

```
import time
import pyautogui
def tomar_captura(numero_captura, x, y, ancho, alto):
    captura = pyautogui.screenshot(region=(x, y, ancho, alto))
    captura.save(r'C:\Users\rianm\OneDrive\Documentos\Capturas
CMS4\captura%_' + str(numero_captura) + '.png')
def tomar_captura2(numero_captura, x, y, ancho, alto):
    captura = pyautogui.screenshot(region=(x, y, ancho, alto))
    captura.save(r'C:\Users\rianm\OneDrive\Documentos\Capturas
CMS5\capturaton_' + str(numero_captura) + '.png')
if True:
    x_rectangulo = 1200
    y_rectangulo = 0
    ancho_rectangulo = 1400#1400
    alto_rectangulo = 1400

    x_rectangulo2 = 1850
    y_rectangulo2 = 0
    ancho_rectangulo2 = 670
    alto_rectangulo2 = 1300

    numero_captura = 1
    while numero_captura<5000:
        tomar_captura(numero_captura, x_rectangulo, y_rectangulo,
ancho_rectangulo, alto_rectangulo)
        tomar_captura2(numero_captura, x_rectangulo2, y_rectangulo2,
ancho_rectangulo2, alto_rectangulo2)
        numero_captura += 1
        time.sleep(0.5)
```

### Código para analizar datos en las capturas del CMS

```
import os
import cv2
import pytesseract
import re
import pandas as pd

# Configura la ubicación del ejecutable de Tesseract (actualiza con tu ruta
específica)
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-
OCR\tesseract.exe'

hourcms = " "
datecms = " "

def fechayhora(texto):
    # Definir la expresión regular para encontrar la hora y fecha después de
"CMS Time"
    patron = re.compile(r"CMS Time (\d{2}:\d{2}:\d{2},\d{3}) \((\d{4}-\d{2}-
\d{2})\)", re.IGNORECASE)

    # Buscar la coincidencia en el texto
    coincidencia = patron.search(texto)
```

```

# Obtener la hora y fecha si hay coincidencia
if coincidencia:
    hourcms = coincidencia.group(1).split(',')[0]
    datecms = coincidencia.group(2)
    return datecms, hourcms
else:
    print("No se encontró la información de hora y fecha después de CMS
Time.")

def procesar_imagen(imagen_path, texto_inicial, texto_final):
    # Lee la imagen con OpenCV
    imagen = cv2.imread(imagen_path)

    if imagen is not None:
        print("Imagen cargada correctamente:", imagen_path)
    else:
        print("Error al cargar la imagen:", imagen_path)
        return None

    # Utiliza tu función para encontrar texto en la imagen
    texto, fecha, hora = encontrar_texto_en_imagen(imagen, texto_inicial,
texto_final)

    if texto:
        # Utiliza tu función para encontrar valores numéricos en el texto
        resultados = encontrar_valores_numericos(texto)

        return resultados
    else:
        return None

def encontrar_texto_en_imagen(imagen, texto_inicial, texto_final):
    # Convierte la imagen a escala de grises
    gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)

    # Utiliza Tesseract para encontrar texto en la imagen
    texto_completo = pytesseract.image_to_string(gris)
    #print(texto_completo)
    #texto_2 = texto_completo[texto_completo.find("CMS") +
len("CMS"):texto_completo.find("Size")].replace(" ", ',').replace('\n',
').replace(')', ',').split(',')
    #datohora = texto_2[1].replace("Time", '')
    #datofecha = texto_2[3].replace("(", '')
    #datofecha, datohora = fechayhora(texto_completo)
    datohora = ''
    datofecha = ''

    # Encuentra la posición del texto inicial
    inicio_pos = texto_completo.find(texto_inicial)

    # Verifica si el texto inicial fue encontrado
    if inicio_pos != -1:
        # Busca el texto final después del texto inicial
        fin_pos = texto_completo.find(texto_final, inicio_pos)

        # Extrae el fragmento de texto entre el texto inicial y final
        texto_deseado = texto_completo[inicio_pos +

```

```

len(texto_inicial):fin_pos]

    return texto_deseado, datofecha, datohora
else:
    print("Texto inicial no encontrado.")
    return None

def encontrar_valores_numericos(cadena):
    # Definir los nombres específicos
    nombres = ['load', 'Trolley position', 'Hoist position',
               'Motor1_1 Vibration', 'Motor1_2 Vibration', 'Motor2_1
Vibration', 'Motor2_2 Vibration',
               'Motor current hoist 1', 'Motor current hoist 2', 'DC voltage
hoist 1', 'DC voltage hoist 2'
               ]

    valores_numericos = {}

    for nombre in nombres:
        # Utilizar regex para encontrar el valor numérico después del nombre
        match = re.search(rf'{re.escape(nombre)}\s*([-
+]?\d*\.\d+)\s*[%mm/stonm]*', cadena)

        if match:
            valor_numerico = match.group(1)
            if ((nombre=='Motor current hoist 1')or (nombre=='Motor current
hoist 2'))and(float(valor_numerico)>150)):
                valores_numericos[nombre] = float(valor_numerico)/10.0
            elif((nombre=='Motor1_1 Vibration')or (nombre=='Motor1_2
Vibration')or (nombre=='Motor2_2 Vibration')or (nombre=='Motor2_1
Vibration'))and(float(valor_numerico)>=11)):
                valores_numericos[nombre] = float(valor_numerico) / 10.0
            else:
                valores_numericos[nombre] = valor_numerico

        #valores_numericos['Fecha'] = fecha
        #valores_numericos['Hora'] = hora
    return valores_numericos

def obtener_numero_captura(nombre_archivo):
    # Extraer el número "x" de un nombre de archivo como "captura_x"
    return int(nombre_archivo.replace('.', '_').split('_')[1])

# Carpeta que contiene las imágenes
carpeta_imagenes = r'C:\Users\rianm\OneDrive\Documentos\Capturas CMS3'

# Obtener la lista de archivos de imagen filtrando por extensiones válidas
archivos_imagen = [archivo for archivo in os.listdir(carpeta_imagenes)
                    if archivo.lower().endswith(('.png'))]

# Ordenar la lista de archivos utilizando la función de clave personalizada
archivos_imagen_ordenados = sorted(archivos_imagen,
key=obtener_numero_captura)

# Textos de referencia
texto_inicial = '|'
texto_final = 'Snag load'

# Lista para almacenar los resultados de cada imagen

```

```

resultados_totales = []
#print(os.listdir(carpetas_imagenes))

# Seleccionar solo los archivos cada 3
archivos_cada_3 = [archivo for i, archivo in
enumerate(archivos_imagen_ordenados) if (i + 1) % 3 == 0]
# Procesar cada archivo de imagen en la carpeta
for archivo in archivos_imagen_ordenados[1238:3000]:
    if archivo.lower().endswith('.png'):
        ruta_imagen = os.path.join(carpetas_imagenes, archivo)
        resultados = procesar_imagen(ruta_imagen, texto_inicial, texto_final)
        if resultados:
            print("Valores numéricos encontrados:", resultados)
            if len(resultados)==11:
                resultados_totales.append(resultados)

# Crear un DataFrame de Pandas
df = pd.DataFrame(resultados_totales)
print(df)
# Definir el orden de columnas
columnas_ordenadas = ['DC voltage hoist 1', 'DC voltage hoist 2',
'Motor current hoist 1', 'Motor current hoist 2',
'Motor1_1 Vibration', 'Motor1_2 Vibration', 'Motor2_1
Vibration', 'Motor2_2 Vibration',
'Hoist position', 'load', 'Trolley position'
]

# Reorganizar las columnas en el DataFrame
df = df[columnas_ordenadas]
print(df)
# Guardar el DataFrame en un archivo Excel
df.to_excel(r'C:\Users\rianm\OneDrive\Documentos\base_de_datos_W.xlsx',
index=False)

```

## Oportunidad de Banco Pruebas a futuro en la ESPOL

El banco de pruebas utilizado en el laboratorio de instrumentación industrial está conformado por un motor monofásico marca TEKNO con los siguientes datos de placa: 60 HZ, 0.5HP, 110 V, 3450 RPM y 2.1 A.

Tres chumaceras marca KDF con cajera #P205 con sus 5 respectivos rodamientos modelo UC205-16 con graseras incluidas; dos en perfecto estado y uno dañado, un eje de acero inoxidable de 40 cm, un acople flexible con amortiguador marca Lovejoy y un disco de acero inoxidable con agujeros a 90° para simular desbalance y un perno de 5/8 para ajustar el disco y 4 pernos de 5/8 con sus respectivos anillos para la fijación en las chumaceras del banco de pruebas y 4 pernos 5/16

para el anclaje del motor en la base del banco, un cilindro de acrílico para generar falla, un interruptor, un tomacorriente, una extensión, una cinta doble faz y el banco con 91 cm de Alto, 31 cm ancho y 85 cm de largo.



### ***Acelerómetro PCB Piezoeléctrico 352C03***

Acelerómetro de precisión para satisfacer a la medida exacta de la vibración, choque, la aceleración y el movimiento de seguimiento, control y aplicaciones de pruebas.

De uso general, de cerámica del esquileo ICP ® Accel., 10 mV / g, 0,5 a 10k Hz, 10-32 CONN lado.

- Sensibilidad: ( $\pm 10\%$ ) de 10 mV / g (1,02 mV / (m/s<sup>2</sup>))
- Rango de medición:  $\pm 500$  pk g ( $\pm 4,900$  m/s<sup>2</sup>pk)
- Resolución de banda ancha (1 a 10000 Hz) RMS 0,0005 g (0.005 m/s<sup>2</sup> RMS)
- Rango de frecuencia: ( $\pm 5\%$ ) 0,5 a 10000 Hz
- Peso: 0.20 onzas (5.8 gramos)



***Cable BNC Modelo 003C10 PCB***

Un cable en el cual permite conectar el acelerómetro piezoeléctrico con el módulo NI 9234, Tiene bajo nivel de ruido de 10 pies (3 m), conjunto de cables, 10-32 conector coaxial al conector BNC.

