



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Identificación de anomalías de máquinas asíncronas por medio de un monitoreo no intrusivo con sensores de vibración

INFORME DE PROYECTO INTEGRADOR

Precio a la obtención del Título de:

INGENIERO EN ELECTRICIDAD

Jorge Leonardo Morejón Aucancela

Joshua Alberto Choez Rodríguez

GUAYAQUIL – ECUADOR

AÑO: 2023

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por la vida y la salud, por ser mi guía y mi fortaleza en cada paso de este camino académico. Mi agradecimiento se extiende a los profesores de la ESPOL, cuyas enseñanzas han sido fundamentales en mi crecimiento y desarrollo que, con sus lecciones fueron pilares importantes en mi formación. Un especial agradecimiento al Máster Eduardo Altamirano Ochoa, CEO de MOVEC, por ser clave en el desarrollo de mi proyecto de materia integradora. Usted no solo fue un mentor, sino el arquitecto intelectual detrás de todo, su tiempo y apoyo han sido invaluable. Agradezco a la Ingeniera Jenny Maribel Aucancela, por su creencia en mis capacidades que, han sido un faro de motivación y confianza para este largo viaje. Asimismo, extendo mi gratitud al Ingeniero Jorge Morejón Noboa, cuya disposición y apoyo dio un gran soporte para la culminación del proyecto de materia integradora. De la misma manera, me gustaría agradecer a todas aquellas personas que formaron parte de mi vida académica, aquellos que compartieron su conocimiento conmigo, experiencias y sabiduría, no solo me enseñaron aspectos técnicos de mi carrera si no que me formaron como profesional y como persona. Sin su consideración este logro no habría sido posible.

-Jorge Leonardo Morejón Aucancela-

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa a la realización de este proyecto. En primer lugar, agradezco a los profesores de la ESPOL por su orientación durante mi trayectoria académica, su conocimiento y apoyo fueron fundamentales para el desarrollo de este trabajo. Asimismo, quiero reconocer a los ingenieros con quienes tuve el privilegio de colaborar durante mis prácticas preprofesionales, quienes generosamente compartieron su experiencia y conocimientos en el campo de la electricidad. Por supuesto, no puedo pasar por alto el apoyo incondicional de mis padres, cuyo sacrificio y aliento constante han sido mi mayor inspiración a lo largo de estos años de estudio.

-Joshua Alberto Choez Rodríguez-

DEDICATORIA

Con mucho amor y gratitud, dedico este trabajo a los pilares fundamentales de mi vida, a mi querida madre, Ingeniera Jenny Aucancela Yambay, por sus enseñanzas, por su amor y sacrificios, por su guía, por ser mi inspiración para ir tras mis sueños y mi refugio incondicional, eres la razón de mi perseverancia. A mi abuelita, Elvira Yambay Bonifaz, por su ternura y gran apoyo, por enseñarme la bondad, la compasión y por ser mi guía espiritual. En memoria de mi abuelito, Luis Alberto Aucancela Soria, quien dejó un legado invaluable, sus valores de honestidad, trabajo duro y dignidad que siempre intento honrar siguen vivos en mí, y fueron importantes en mi camino académico y continúan siendo en mi vida personal. A mi padre, por su apoyo y enseñanzas que, alimentaron mi pasión por la investigación y el descubrimiento. Agradezco a Milena P. Suarez, por su amor, motivación y apoyo, siendo mi roca en momentos difíciles y una de mis más grandes alegrías en mis momentos de éxito. Agradezco a mis compañeros de universidad, por hacer de este viaje una aventura compartida, celebrando juntos nuestros éxitos y compartiendo nuestras experiencias. A cada uno de ustedes por ser parte de mi historia, les agradezco. Han sido el impulso detrás de cada triunfo hasta esta parte de mi vida.

-Jorge Leonardo Morejón Aucancela-

Dedico esta tesis a mis padres, Kleber Choez López y Mónica Rodríguez Malavé, cuyo amor y constante aliento han sido la base de mi educación. A mis hermanos, Melanie e Isaí, por su apoyo incondicional. A mis amigos de la universidad, por alentar el estudio dentro y fuera de las aulas de clase. A mis amigos de la infancia, por su apoyo en mi vida personal. Y a los ingenieros, futuros colegas, que me han apoyado profesionalmente, su experiencia ha sido de gran ayuda.

-Joshua Alberto Choez Rodríguez-

TRIBUNAL DE EVALUACIÓN

RECALDE LINO ANGEL ANDRES

PROFESOR DE MATERIA
INTEGRADORA

UGARTE VEGA LUIS FERNANDO

TUTOR ACADÉMICO

DECLARACIÓN EXPRESA

“La responsabilidad y la autoría del contenido de este Trabajo de Titulación nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



**JORGE LEONARDO
MOREJÓN AUCANCELA**



**JOSHUA ALBERTO
CHOEZ RODRIGUEZ**

RESUMEN

Este estudio trata el problema de asegurar el óptimo funcionamiento de las máquinas asíncronas, que son fundamentales para diversos procesos industriales. La investigación respalda el empleo de un sistema de análisis de vibraciones para prevenir de manera temprana cualquier anomalía en estas máquinas, resaltando la relevancia de un monitoreo que no altere su funcionamiento habitual. Se pretende realizar un tablero con pantalla HMI para exhibir y examinar datos, junto con una comunicación eficiente entre sensores de vibración y un procesador de datos para análisis en tiempo real.

Luego de obtener los datos de vibración se transformó al dominio de la frecuencia para generar gráficos y valores que diferenciaron diferentes tipos de fallas.

La motivación para abordar este problema se basa en la necesidad de mantener la eficiencia y calidad en los procesos de producción industriales, donde las máquinas asíncronas implican una inversión significativa. El sistema recomendado ofrece una solución eficaz y económica al permitir prever problemas antes de que repercutan en el rendimiento, otorgando una ventaja estratégica al aplicar medidas preventivas. Además de disminuir costos vinculados con el tiempo de parada y reparaciones, se resalta la preservación de la calidad de la producción, satisfacción del cliente y reputación empresarial en un entorno competitivo.

Palabras clave:

Máquinas asíncronas

Análisis de vibraciones

Datos de vibración

Frecuencia

Espectro

ABSTRACT

This study discusses the problem of ensuring the optimal operation of asynchronous machines, which are essential for various industrial processes. The research supports the use of a vibration analysis system to prevent early anomaly in these machines, highlighting the relevance of a monitoring that does not alter their normal operation. It is intended to make a dashboard with HMI screen to display and examine data, together with an efficient communication between vibration sensors and a data processor for real time analysis.

After obtaining the vibration data, it was transformed to frequency domain to generate graphs and values that differentiated different types of failures.

The motivation to address this problem is based on the need to maintain efficiency and quality in industrial production processes, where asynchronous machines involve a significant investment. The recommended system offers an effective and economical solution by making it possible to anticipate problems before they impact performance, providing a strategic advantage by applying preventive measures. In addition to reducing costs related to downtime and repairs, the preservation of production quality, customer satisfaction and business reputation in a competitive environment is highlighted.

Keywords:

Asynchronous machines

Vibration analysis

Vibration data

Frequency

Spectrum

INDICE GENERAL

TRIBUNAL DE EVALUACIÓN.....	4
RESUMEN.....	VI
ABSTRACT.....	VII
INDICE GENERAL.....	VIII
INDICE DE FIGURAS.....	IX
1. INTRODUCCIÓN.....	1
1.1 Descripción del problema.....	1
1.2 Justificación del Problema	2
1.3 Objetivos.....	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos Específicos	3
1.4 Marco Teórico	4
1.4.1 Espectros vibratorios	4
1.4.2 Motores de inducción.....	4
1.4.3 Fallas en motores	6
2. METODOLOGÍA.....	7
2.1. Desarrollo de códigos de fallas	7
2.2. Interfaz y creación del HMI	13
2.3 Pasos para la instalación del sistema operativo en el dispositivo	16
2.3.1. Primer método: Sistema Operativo Windows 10	17
2.3.2 Segundo Método: Sistema Operativo Raspbian	23
2.4. Instalación y actualización de los componentes necesarios para el programa desarrollado.....	31
3. RESULTADOS.....	36
3.1. Desarrollo y Aplicación de Código de las fallas en motores	36
3.2 Pruebas en los motores usando el sensor de vibración y el monitor de vibraciones.	50
4. Conclusiones y recomendaciones	64
Conclusiones.....	64
Recomendaciones.....	65

INDICE DE FIGURAS

Figura 1. Sensor de vibración WTVB01-485	10
Figura 2. Espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada	11
Figura 3. Segundo resultado de los espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada	12
Figura 4. Tercer resultado de los espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada.	12
Figura 5. Interfaz del programa realizado en Python para el análisis de vibraciones.	13
Figura 6. Resultados de los espectros de vibración del sensor y a la derecha de la imagen, los espectros característicos de fallas en máquinas asíncronas	14
Figura 7. Microprocesador Raspberry Pi 3	15
Figura 8. Tarjeta de memoria MicroSD Sandisk 64 GB	17
Figura 9. Página web UUP Dump	18
Figura 10. Selección del arm64 de Win10	18
Figura 11. Selección del Idioma para el sistema operativo	19
Figura 12. Activación de ambas ediciones del sistema operativo Windows 10	19
Figura 13. La opción Download and Convert to ISO está marcada	20
Figura 14. Archivo .ZIP dentro de una carpeta sin espacios dentro del disco E:	20
Figura 15. Mensaje de alerta del procesador de comando de Windows	21
Figura 16. Configuración en el programa Windows on Raspberry	22
Figura 17. Montando la Imagen ISO en la tarjeta MicroSD	22
Figura 18. Instalación del sistema operativo en la MicroSD como sistema Bootable	23
Figura 19. Página de descarga del instalador de Raspbian	24
Figura 20. Ventana del instalador para el sistema operativo Raspbian	24
Figura 21. Programa instalador de la imagen para el sistema operativo Raspbian	25
Figura 22. Selección de opción Raspberry en posesión.	25
Figura 23. Sistemas Operativos a seleccionar.	26
Figura 24. Elección Sistema Operativo Raspberry pi Os (Legacy, 32-bit) Full.	26
Figura 25. Ventana Final de la instalación del sistema operativo en la tarjeta MicroSD	27
Figura 26. Primera imagen del dispositivo Raspberry pi 3 con el sistema operativo.	27
Figura 27. Bienvenida del Software Raspbian	28
Figura 28. Configuración de idioma y parámetros básicos iniciales.	28
Figura 29. Configuración del usuario para la cuenta en el dispositivo.	29
Figura 30. Actualizaciones necesarias para el sistema operativo del dispositivo.	29
Figura 31. Ultima ventana antes del reinicio del sistema operativo con las actualizaciones.	30
Figura 32. Bienvenida Sistema Operativo Raspberry pi Os	30
Figura 33. Pantalla de Escritorio del Sistema Operativo Raspbian	31
Figura 34. Actualización de las librerías y componentes del sistema operativo.	32
Figura 35. Actualización del paquete del sistema operativo.	32
Figura 36. Librerías necesarias para el correcto funcionamiento del programa desarrollado.	34
Figura 37. Primera visualización del programa desarrollado en el sistema operativo Raspbian.	34
Figura 38. Sistema de Monitoreo Vibracional con HMI.	35
Figura 39. Código de la falla como señal de desbalance	36
Figura 40. Código de la Transformada de Fourier para la falla de desbalance	37
Figura 41. Código para hallar el espectro de falla de desbalance	37
Figura 42. Vibraciones en el dominio del tiempo y espectro de frecuencia de la falla de desbalance	38
Figura 43. Espectro de frecuencia de la falla de desbalance	38
Figura 44. Código de la falla de excentricidad y de la Transformada de Fourier	39
Figura 45. Código para hallar el espectro de la falla de excentricidad	39
Figura 46. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de excentricidad	40
Figura 47. Espectro de frecuencia de la falla de excentricidad	40
Figura 48. Código de la falla de holguras y de la Transformada de Fourier	41

Figura 49. Código para hallar el espectro de la falla de holguras _____	41
Figura 50. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de holguras _____	42
Figura 51. Espectro de frecuencia de la falla de excentricidad _____	42
Figura 52. Código de la falla como señal de problemas en rodamientos _____	43
Figura 53. Código de la transformada de Fourier de la falla en rodamientos _____	43
Figura 54. Código para hallar el espectro vibratorio de la falla en problemas en rodamientos _____	43
Figura 55. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de rodamientos _____	44
Figura 56. Espectro de frecuencia en problemas de rodamientos _____	45
Figura 57. Código de la falla como señal de problemas eléctricos y la transformada de Fourier de la señal completa _____	45
Figura 58. Código del espectro de problemas eléctricos _____	46
Figura 59. Vibraciones en el dominio del tiempo y el espectro de frecuencia de problemas eléctricos _____	46
Figura 60. Espectro de frecuencia en problemas eléctricos _____	47
Figura 61. Grupo de Bombeo 9H y placa del motor. _____	48
Figura 62. Resultados de vibración en el eje X del grupo de bombeo 9H en La Toma-INTERAGUA _____	48
Figura 63. Pruebas de comparación del sensor WTVB01-485 y el sensor del analizador CAT _____	50
Figura 64. Espectro vibratorio de falla de desbalance _____	50
Figura 65. Espectro vibratorio de la falla de desalineación _____	51
Figura 66. Espectro vibratorio de la falla de holguras _____	51
Figura 67. Espectro vibratorio de la falla de rodamientos _____	52
Figura 68. Espectro vibratorio de la falla de problemas eléctricos _____	52
Figura 69. Ubicación del sensor de vibración en el motor para las pruebas de funcionamiento. _____	53
Figura 70. Estado Vibracional de la máquina Asíncrona en operación normal en el eje X _____	54
Figura 71. Estado Vibracional de la máquina Asíncrona en operación normal en el eje "y", el eje "z" _____	55
Figura 72. Rangos de la Norma ISO 2372 _____	55
Figura 73. Espectro de Frecuencia del eje X _____	56
Figura 74. Espectro de Frecuencia del Eje Y _____	57
Figura 75. Espectro de Frecuencia del Eje Z _____	57
Figura 76. Pernos de la base del motor sueltos para las pruebas de funcionamiento _____	58
Figura 77. Pernos desajustados en la base del motor para pruebas de funcionamiento _____	59
Figura 78. Estado vibracional en el dominio del tiempo del eje X cuando se altera la estructura del motor _____	60
Figura 79. Estado vibracional en el dominio del tiempo del eje Y y Z cuando se altera la estructura del motor _____	60
Figura 80. Espectro de frecuencia del eje X _____	62
Figura 81. Espectro de frecuencia del eje Y _____	62
Figura 82. Espectro de frecuencia del eje X _____	63
Figura 83. Sensor es de vibración WTVB01-485 _____	66
Figura 84. Pruebas realizadas con el sensor en una máquina de clase I y clase III según la norma ISO 2372 _____	66
Figura 85. Primer código utilizado para comunicar computador con programa Python y sensor con convertidor RS485 a USB. _____	67
Figura 86. Resultados obtenidos con el código actualizado para presentar gráficas en tiempo real con Matplotlib _____	68
Figura 87. Pruebas realizadas en un sistema de bombeo en INTERAGUA - LA TOMA _____	68
Figura 88. Ficha Técnica del motor utilizado para monitoreo de vibraciones del sistema de bombeo. _____	69
Figura 89. Pruebas de vibración con una maleta de pruebas con motor de 0.5 HP y un variador de frecuencia _____	69
Figura 90. Resultados del monitoreo vibracional obtenido con la maleta de pruebas _____	70
Figura 91. Espectros de frecuencia de vibración en tiempo real de la maleta de pruebas en los tres ejes _____	70
Figura 92. Resultados obtenidos con la actualización de la interfaz del programa con sistema de monitoreo y espectro vibracional en tiempo real _____	71
Figura 93. Pruebas realizadas con sensor profesional CAT _____	71
Figura 94. Nueva interfaz gráfica desarrollada para el programa con botones, checkbox y ventanas extra para presentar datos _____	72

<i>Figura 95. Pruebas de compatibilidad realizadas en un sistema operativo basado en LINUX con el sensor de vibración</i>	72
<i>Figura 96. Tarjeta STM32MP135F-DK</i>	73
<i>Figura 97. Raspberry Pi 3 como consola emuladora de juegos</i>	73
<i>Figura 98. Integración pantalla HMI 5" con Raspberry Pi 3 y sistema operativo WIN10</i>	74
<i>Figura 99. Creación de caja para tablero de monitoreo vibracional</i>	74
<i>Figura 100. Tablero monitor de vibraciones con sistema operativo Raspbian en un Raspberry Pi 3 en pruebas de funcionamiento con maleta de pruebas</i>	75

CAPITULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

Las máquinas asíncronas trifásicas son un elemento importante en las compañías donde se tienen producción, por lo que, grandes empresas afrontan el desafío de garantizar su rendimiento constante sin interrupciones, una falla en estas máquinas significa parar la producción, lo que conllevaría a pérdidas económicas y afectar de manera negativa la reputación de la empresa porque estas máquinas operan en áreas de largos y arduos procesos, tal es la razón por la que tienen que estar en correcto funcionamiento y deben ser monitoreadas constantemente.

Las restricciones identificadas durante la investigación incluyen la necesidad de que el sistema sea económicamente viable y fácilmente integrable en entornos industriales existentes. Además, se enfatiza la importancia de presentar los resultados del análisis de vibraciones de manera clara y accesible a través de la interfaz humano-máquina.

Las variables de interés, según la investigación, se centran en la detección temprana de fallas en las máquinas asíncronas. Esto implica la recopilación precisa de datos de vibración, su transformación del dominio del tiempo al dominio de la frecuencia, y la generación de gráficos que identifiquen distintos tipos de fallas. La efectividad del sistema se mide en su capacidad para alertar al operador ante posibles problemas, reduciendo así el tiempo de inactividad y los costos de reparación. Es por tal razón que se propone la implementación de un sistema de monitoreo de vibraciones como una solución para identificar de manera temprana cualquier anomalía en las máquinas asíncronas trifásicas. Por lo que se necesita describir los requerimientos para resolver este problema, tales como: el desarrollo de un sistema de monitoreo con sensores de vibración que pueda detectar cualquier anomalía que influya con el funcionamiento normal de los motores asíncronos.

1.2 Justificación del Problema

La justificación para abordar esta problemática se basa en la necesidad de las grandes industrias de mantener la eficiencia y calidad en sus procesos de producción. Se conoce que las máquinas asíncronas son la mayor inversión de muchas operaciones industriales y cualquier fallo en su funcionamiento puede tener consecuencias económicas y de reputación significativas.

La solución que se propone está basada en un sistema de monitoreo no intrusivo mediante análisis de vibraciones. Donde la capacidad de prever problemas antes de que afecten el rendimiento del motor otorga a las empresas una ventaja estratégica para aplicar medidas preventivas antes que correctivas. Además, esta estrategia no solo reduciría los costos relacionados con el tiempo de inactividad y reparaciones en casos que se requiera, sino que también preserva la calidad de la producción, la satisfacción del cliente y la reputación de la empresa.

En un panorama industrial altamente competitivo, donde la eficiencia es fundamental, se respalda que la implementación de este sistema de monitoreo no intrusivo no solo resuelve un problema crítico, sino que también proporciona una ventaja estratégica a aquellas empresas que lo adoptan. Al aplicar un enfoque de mantenimiento preventivo basado en monitoreo de datos, se garantiza la continuidad y la eficiencia en los procesos industriales, creando el camino hacia un futuro donde la maquinaria asíncrona opera de manera óptima y sin contratiempos, siendo este enfoque una oportunidad para impulsar la competitividad y la sostenibilidad de las empresas en el ámbito industrial.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema de monitoreo no intrusivo utilizando sensores de vibración y análisis de datos mediante Python para identificar de manera efectiva las anomalías en máquinas asíncronas.

1.3.2 Objetivos Específicos

- Implementar un dispositivo de adquisición de datos con sensores de vibración en máquinas asíncronas para recopilar información relevante sobre su comportamiento.
- Desarrollar y optimizar algoritmos de procesamiento de datos utilizando Python para analizar las mediciones de vibración y generar espectros vibratorios conforme a la norma ISO 10816-3.
- Validar la eficacia del sistema propuesto mediante pruebas y comparaciones con estudios de casos de anomalías en máquinas asíncronas, evaluando la precisión y fiabilidad de las detecciones realizadas.

1.4 Marco Teórico

1.4.1 Espectros vibratorios

El análisis de vibraciones es una de las técnicas fundamentales de monitorización del estado para el mantenimiento de maquinaria y el diagnóstico de fallos, además de que, entre los diversos métodos de control de estado, los métodos basados en la corriente y las vibraciones son los preferidos debido a su fiabilidad, no intrusividad y fácil medición [1], además, el efecto de corriente desequilibrada y la vibración/ruido en los motores de inducción son cada vez más importantes [2] por lo que se debe considerar un estudio respecto a estos datos.

Los espectros vibratorios son señales gráficas que muestran las diferentes vibraciones que emiten las máquinas industriales donde pueden existir vibraciones que son causa de diferentes eventos que estén sucediendo en la máquina ya sea por desbalance, desalineación, desgaste, fricción, entre otras. Existen varias técnicas convencionales de análisis de vibraciones y corrientes mediante las cuales se pueden identificar y reparar determinados fallos en maquinaria rotativa antes de que se produzcan fallos catastróficos en un componente [3]. Normalmente, los analistas de vibraciones utilizan la transformada rápida de Fourier (FFT) para procesar e interpretar datos de vibraciones de motores y otros sistemas rotativos [4]. El análisis de vibraciones es una de las técnicas fundamentales de monitorización para el mantenimiento de maquinaria. Las características específicas de los espectros de vibración y corriente que se asocian a los estados de avería más comunes son bien conocidas [5] [3]. En el caso de las máquinas eléctricas, las normas ISO 7919 e ISO 10816 se centran en el comportamiento de las vibraciones in situ, sin embargo, la diferencia es la ubicación de los dispositivos de medición, a diferencia de la norma ISO 7919, la norma 10816 se basa en una evaluación de la vibración en piezas no giratorias [6] por lo que es la que se utilizará para el proyecto.

1.4.2 Motores de inducción

Se conoce que el motor de inducción trifásico fue inventado a mediados del siglo XIX por el ingeniero ruso Dolivo-Dobrovolsky. Sin embargo, los motores de inducción se han utilizado en muchas aplicaciones industriales desde hace ya 100 años, y su uso

se ha ido extendiendo en aplicaciones de velocidad variable. La vibración y el ruido de los motores no sólo dependen de la fuente de las fuerzas de vibración, que son principalmente electromagnéticas, mecánicas y de fricción del aire, sino también de la estructura y las características mecánicas de la máquina. Es bien sabido que se generarán grandes vibraciones cuando la frecuencia natural de la máquina coincida con la frecuencia de la fuerza aplicada al estator o esté próxima a la fuerza aplicada al estator [7] [8]. Es por tal razón que su uso industrial se ve complementado por los diferentes dispositivos a los que se encuentre relacionado, pudiendo afectar en su funcionamiento de manera positiva o negativa a lo largo de su vida útil.

Los motores de inducción de jaula de ardilla se han usado en diferentes áreas en la industria debido a su bajo costo, bajo mantenimiento y una eficiencia que pocas veces se ve afectada en cuanto a factores externos [9], estas han sido las que más atención han atraído en la investigación del diagnóstico de fallos de máquinas eléctricas debido a su uso generalizado a pesar de que se están utilizando cada vez más otros tipos de máquinas eléctricas rotativas, como las máquinas síncronas de imanes permanentes, las máquinas de rotor bobinado, las máquinas multifásicas, y las máquinas de reluctancia conmutada. Además, los motores multifásicos pueden aplicarse a vehículos eléctricos, aeronaves, propulsión naval, equipos militares y vagones de ferrocarril. Se han investigado métodos de detección de fallos en un inversor de fuente de tensión de cinco fases, accionamientos eléctricos, control de ruido y vibraciones, variación de velocidad y par, vector de Park y método de corriente continua normalizada [10] [11]. A pesar de que los MIJA tienen una construcción robusta, pueden averiarse debido a una instalación incorrecta, defectos durante el proceso de fabricación, carga no uniforme del motor, degradación del aislamiento, etc. Los fallos de los MIJA son de naturaleza incipiente y el motor funciona casi con normalidad. Si el fallo no se detecta, puede provocar el fallo completo del motor y la pérdida de producciones [12] [13].

Cuando existe un fallo en componentes rotativos de sistemas electromecánicos o mecánicos como motores eléctricos, engranajes, ejes y cojinetes, los impulsos periódicos producirán y modificarán amplitudes de modulación inherentes a las señales de corriente, flujo y vibración [14] [15]. Cada falla debe detectarse lo antes posible para evitar consecuencias perjudiciales, así como reducir pérdidas financieras y mejorar el nivel de seguridad [16].

1.4.3 Fallas en motores

En ciertas ocasiones los fallos en los motores no son visibles a simple vista por lo que se debe tener en cuenta ciertos aspectos preventivos para evitar eventos en las máquinas eléctricas. La monitorización del estado y el diagnóstico de fallos de los motores de inducción basados en la técnica del análisis de vibraciones se utilizan ampliamente en la industria moderna. La precisión del diagnóstico de maquinaria realizado mediante el análisis de vibraciones depende de la capacidad de analizar correctamente los datos de vibraciones, especialmente los espectros de frecuencia [17] [18]. Los motores eléctricos en general, y los motores de inducción en particular, pueden tener fallos eléctricos o mecánicos. Los fallos eléctricos pueden estar en las piezas del motor (estator y rotor) o en la alimentación [19].

Se conoce además que el MIJA vibra con una frecuencia específica para cada falla y los movimientos fundamentales describen la dinámica de los rodamientos [20] por lo que se necesita conocer los tipos de fallas que suele tener un motor de inducción, no solo de jaula de ardilla sino también los diferentes motores que se encuentren en el sector industrial. [21]

Entre las fallas que se pueden encontrar en los motores se presentan los siguientes,

- Los fallos en los rodamientos pueden deberse a una lubricación insuficiente, así como a fuertes tensiones radiales y axiales debidas a la desviación del eje, montaje, alineación o cimentación inadecuados.
- Los fallos del bobinado del estator pueden ser fallos del bobinado entre vueltas causados por la rotura del aislamiento.
- Los fallos del rotor se producen principalmente por roturas en las uniones entre las barras y los anillos extremos, normalmente como resultado de una carga pulsante o de arranques directos en línea [22] [23].

Es de mencionar que, cualquier falla asociada con los devanados del estator o del rotor, incluidas las variaciones del entrehierro, producirá cambios en estas magnitudes, por lo que las corrientes de línea y la vibración radial del núcleo del estator pueden utilizarse para supervisar las condiciones del motor [24].

Una vez se detectan problemas relacionados con las vibraciones del motor, es importante utilizar un enfoque sistemático para resolver el problema, enumerando

cualquier tipo de variación en la frecuencia de vibración que se ha identificado en diferentes condiciones de funcionamiento [25].

CAPITULO 2

2. METODOLOGÍA

2.1. Desarrollo de códigos de fallas

Las mediciones de las vibraciones se las determinará por un sensor de vibración, el procesamiento de estos se realizará en lenguaje de Python a código abierto ya sea para pruebas en tiempo real donde el sensor se encuentre en operación recopilando las mediciones, así como para pruebas offline donde no se necesite que el sensor se encuentre conectado en estado de operación sino al tener los resultados en un documento compatible.

Inicialmente se realizó el estudio del proyecto con base en los procedimientos generales para una correcta medición y una valoración de datos vibratorios en máquinas industriales asíncronas, como se indica en la norma ISO 10816-3, partiendo de ciertos eventos o sucesos que pueden suceder en motores que se los denominó ejemplos, clasificados así, de perturbación, desbalance, excentricidad, ejemplo de holguras, rodamientos y problemas eléctricos. Esto supuso un avance notorio para reconocer los espectros vibratorios de las fallas o ejemplos mencionados proporcionando información sobre la transformada de los datos de tiempo a frecuencia, y sus respectivos gráficos, aunque no se tuvo información, se realizó un análisis intrusivo sobre los eventos que pueden suceder en los motores.

La obtención de los gráficos de los espectros característicos de las respectivas fallas o eventos se realizaron mediante programación en Python, como primer ejemplo de desbalance se realizó el análisis de falla en base al espectro característico presentado como límite permisible según la norma ISO 10816-3 donde se puede identificar el evento cuando aumenta la amplitud de los armónicos en un gráfico de espectro de frecuencia que se obtuvo como resultado tal como se muestra en la figura 42, además, de que se detecta únicamente cuando el sensor recopila datos de manera radial.

Además, se realizó el código para determinar el espectro característico de un evento de desalineamiento, a pesar de que este tipo de falla presenta varias clasificaciones por lo que se lo trató de manera general, es decir, cuando el eje se mueve de su posición normal provocando un comportamiento inusual en el motor. Sin embargo, para efectos de estudio se comprobó que cuando hay falta de alineamiento y es vertical, hay vibraciones radiales monitoreadas por el sensor a frecuencias equivalentes al doble de la frecuencia nominal. Además, cuando la falta de alineamiento sea angular, las vibraciones monitoreadas serán de manera axial pudiendo duplicar o incluso triplicar la velocidad a la que se encuentra girando el rotor de manera nominal. Otra característica sobre la falla de desalineamiento es que cuando el gráfico de espectros presente 3 armónicos significativos dependiendo de su origen de medición (ya sea en el eje de las abscisas, ordenadas y/o axial), si el gráfico tiene tres armónicos significativos y monitoreados en dirección horizontal, suele ocurrir con mucha probabilidad de que la falla sea vertical, lo mismo ocurre cuando se monitorean verticalmente tomando como hipótesis de que existe una falla horizontal, mientras que, si las mediciones son del tipo axial en base al eje de las abscisas.

De la misma manera, cuando existe presencia de excentricidad en el motor la aparición de armónicos aumenta sin importar su amplitud mostrando como amplitud frecuencias de 50 Hz y 200 Hz esto causado por los cambios en la fuerza magnetomotriz ya que se está sufriendo modificaciones en el entrehierro es por tal razón que se monitorea con el sensor y usando el código para obtener el espectro vibratorio correspondiente.

Posteriormente antes de que se modificara el código para poder leer los datos de manera real por lo que se trabajó con el sensor de vibración WTVB01-485 el cual por medio de un convertidor USB de rs482 a USB se logró conectar el sensor y realizar las pruebas por medio de una interfaz propia del dispositivo donde se pudo observar que se tenían parámetros de vibración triaxial en mm/s, desplazamiento, posición en el espacio, temperatura, entre otros parámetros. Una vez que nos aseguramos de que se pudiera exportar los datos medidos, se pudo realizar las pruebas en grandes motores de Interagua y posteriormente se modificó el código en Python para conectar el sensor a la computadora y luego procesar los datos que se estaban recibiendo, luego se llevó el sensor ante un sistema de bombeo para revisar los datos obtenidos y saber si el sensor se encontraba en funcionamiento óptimo. Además, se realizaron

monitoreos y pruebas en motores del laboratorio de maquinaria eléctrica de la ESPOL donde se procedió a tomar los valores de vibración solamente para observar los datos que recibía el sensor junto con el código actualizado y observar las gráficas resultantes de espectros vibratorios como efecto de estudio. Después se consiguió un banco de pruebas con un motor y un variador de frecuencia el cual nos permitió que podamos realizar más pruebas y pulir los resultados obtenidos en el código creado en python, después de haber llegado a un punto donde se presentaban los valores de vibración según la norma con la cual estábamos trabajando se prestó un sensor de vibración profesional de un valor de 17 mil dólares para comparar los resultados obtenidos del sensor de vibración creado por nosotros a uno que fue prestado por un Ingeniero Eléctrico jefe del área de rebobinado de motores en Quito de la empresa Caterpillar lo que dio como resultado valores del sensor de vibración creado por nosotros cercanos al profesional el cual se regía por la norma ISO 2372 la cual fue adoptada posteriormente a eso porque la norma sirve para dar información de las máquinas asíncronas trifásicas dependiendo de su categoría siendo de Clase 1 hasta la clase 3; luego de haber realizado las pruebas se procedió a actualizar el código con las nuevas observaciones como colocar los umbrales correctos de vibración según la norma ISO 2372 dependiendo de la clase de la máquina, así como también una interfaz donde el usuario podría tener una mejor comunicación con la HMI y también varios botones donde el usuario podría iniciar la obtención de datos, detener la compilación, seleccionar la clase deseada y calcular las transformadas de Fourier en los ejes que el usuario seleccionara a su conveniencia lo que después daría como resultado una nueva ventana con las gráficas de las transformadas y también imágenes con las posibles fallas características o fallas principales que un motor asíncrono trifásico podría tener lo que ayudaría que el usuario pueda obtener información del espectro de vibración calculado e interpretar los datos según sea el caso.



Figura 1. Sensor de vibración WTVB01-485

Una de las grandes ventajas de tener este sensor de vibración en específico es que tiene comunicación MODBUS RS485 lo que significa que puede conectarse a otros dispositivos que tengan el mismo protocolo de comunicación como pantallas HMI de la marca Weintek, Delta o Siemens que tienen protocolo de comunicación RS232 pero estas entradas tienen pines específicos para conectar la señal A(+) y B(-) los cuales son puertos de comunicación de los dispositivos para comunicarse mediante RS485 logrando programar las señales y redirigirlas para visualizar sus datos a través de una pantalla.

Además de que la comunicación RS485 puede ser escrita y leída por medio de parámetros hexadecimales, es decir que, conociendo las direcciones hexadecimales de cada variable se podría obtener los parámetros específicos del sensor que el usuario requiera. Para poder comunicar el sensor RS485 a la PC se empleó un convertidor y se instalaron los drivers correspondientes para facilitar la comunicación, luego de comunicar la computadora con el sensor se desarrolló un código de conexión en Python para obtener los datos del sensor en un determinado tiempo configurando en las líneas de código el puerto serial, la velocidad con la que transmitiría la información y los datos de vibración en mm/s del eje x. Esto permitió obtener el parámetro "0x3A" el cual representa el valor de la vibración en el eje X en mm/s.

Gracias a la empresa MOVEC se nos prestó una mochila de pruebas de un motor y variador de frecuencia de la marca WEG lo que permitió realizar un mejor análisis de las vibraciones de una máquina asíncrona por lo que se actualizó el código de Python especificando umbrales de la norma ISO 10816 la cual se divide en categorías para

establecer las zonas de correcto funcionamiento según los datos de vibración que las máquinas de inducción deberían tener para tener un óptimo funcionamiento y se cambió el tiempo de análisis de un minuto a ser en tiempo real.

Esto supuso una mejora notoria para la actualización del código de Python para poder obtener los datos de los tres ejes "X, Y, Z" a la vez con lectura en tiempo real, gracias a estos datos se logró actualizar el código para graficar los espectros de vibración triaxial medidos por el sensor en un punto específico de la máquina asíncrona.

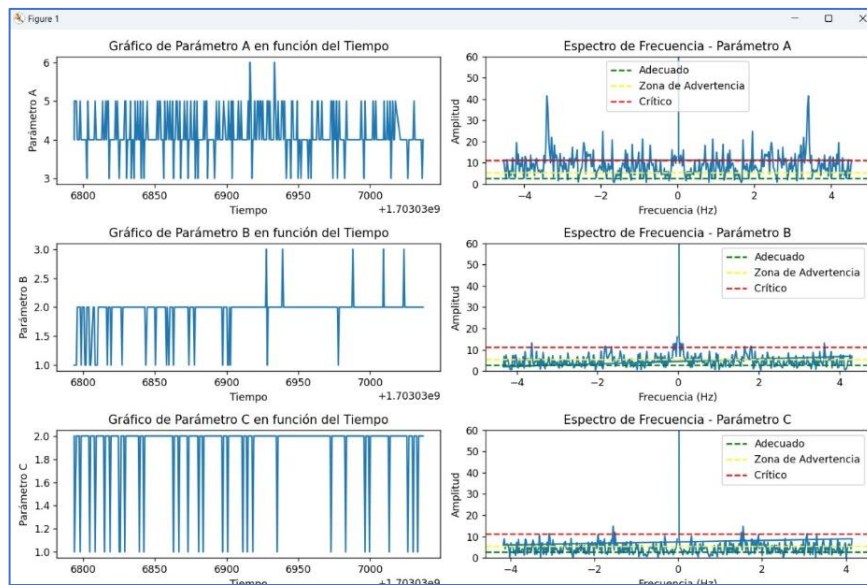


Figura 2. Espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada

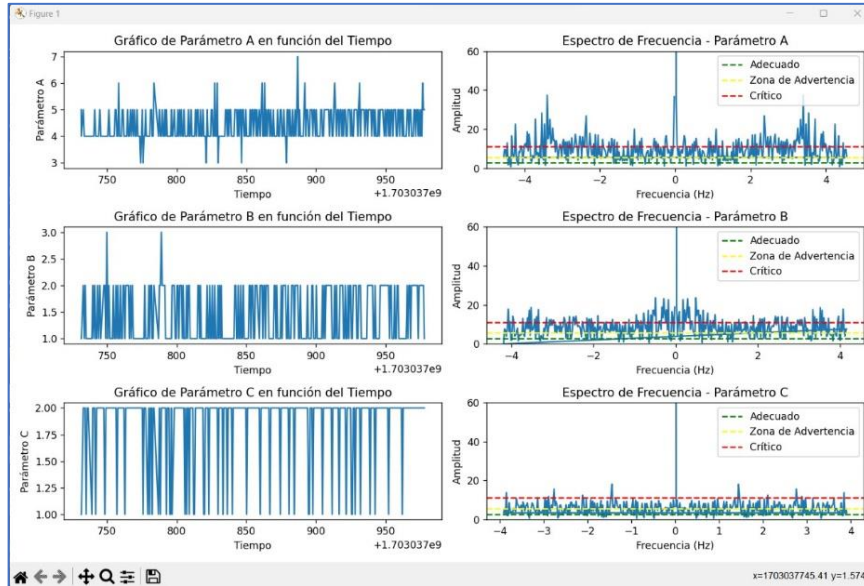


Figura 3. Segundo resultado de los espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada

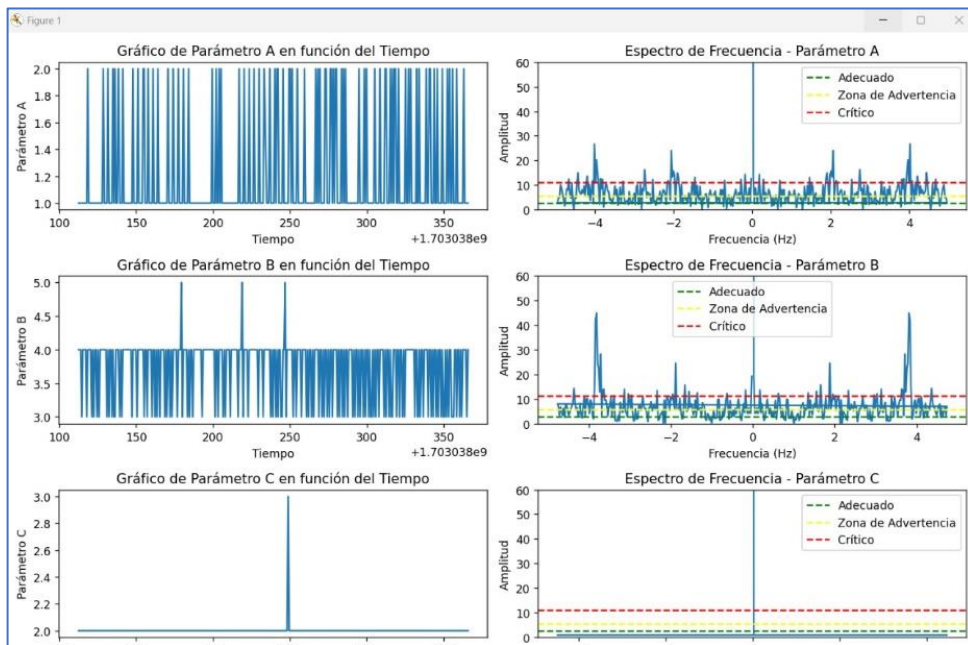


Figura 4. Tercer resultado de los espectros de vibración de los tres ejes "X, Y, Z" en un punto de la máquina asíncrona analizada.

2.2. Interfaz y creación del HMI



Figura 5. Interfaz del programa realizado en Python para el análisis de vibraciones.

La interfaz incluye botones donde el usuario podrá recoger datos en tiempo real según decida si es necesario continuar o no con el análisis de vibración, como elegir entre las Clases I, II y III que categorizan la norma ISO 2372, que depende del motor a utilizar. Una vez que se realiza la recopilación de datos se puede dar clic al botón de “Calcular la transformada de Fourier” para desplegar una nueva ventana que visualiza las gráficas de los espectros de vibración de cada uno de los ejes analizados o marcar las casillas de “Selección de ejes para TF” donde podrá especificar una gráfica de espectro de vibración en específico acompañada de imágenes de los espectros característicos de fallas en motores asíncronos.

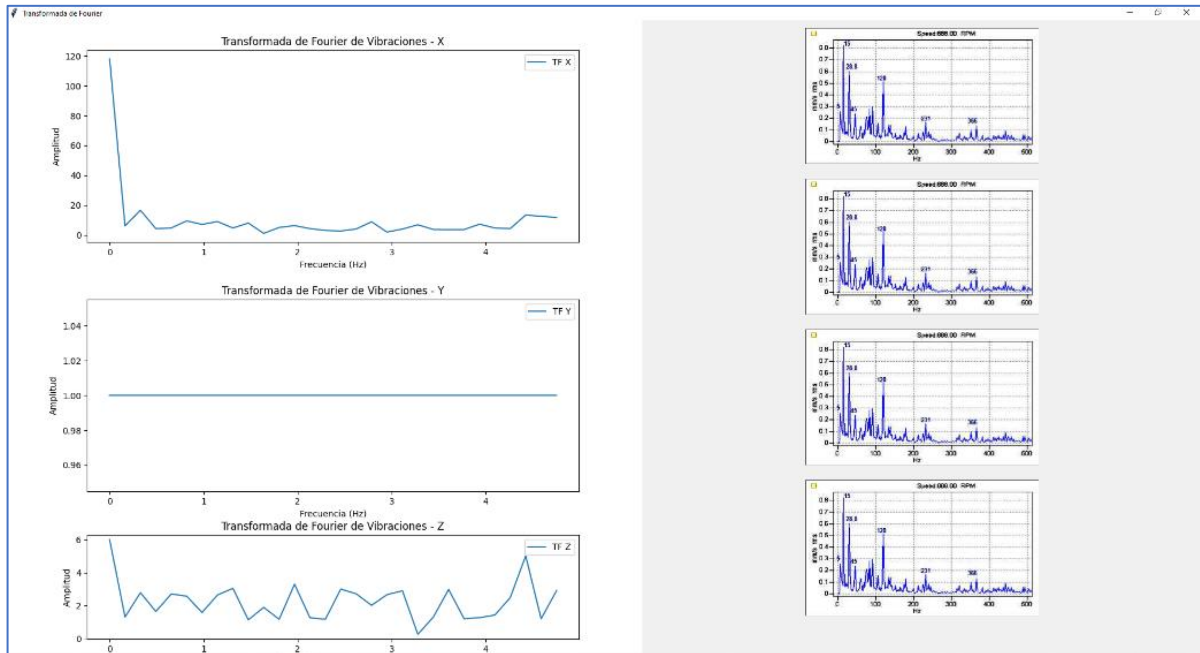


Figura 6. Resultados de los espectros de vibración del sensor y a la derecha de la imagen, los espectros característicos de fallas en máquinas asíncronas

Una vez desarrollado el programa en Python que realiza lo requerido para el análisis de datos y la interfaz hombre-máquina que ayudará a que el usuario pueda comunicarse de manera efectiva con el dispositivo e interpretar los datos, continuamos con la segunda parte que es compactar lo desarrollado en una computadora y otros dispositivos en un pequeño tablero apto para el ambiente industrial el cual puede ser utilizado en las empresas para el análisis de vibración en sus motores en operación normal o previo a un mantenimiento preventivo.

Los elementos pensados para realizar el tablero de prototipo son: un microprocesador capaz de procesar los datos del sensor, una pantalla HMI táctil, el sensor de vibración utilizado en las pruebas de funcionamiento y al ser un prototipo no se utilizará un breaker para la protección de la persona y los equipos y se utilizará una fuente enchufable dentro del tablero donde se conectarán los dispositivos.

Para empezar, se eligió un microcontrolador Raspberry Pi 3 que, tras no ser un dispositivo enfocado al ámbito industrial, puede realizar un análisis de datos porque es capaz de manejar Python 3 en su sistema, lo que lo hace perfecto para el desarrollo del tablero como prototipo, pero más importante aún es que el microcontrolador puede tener Windows 10 como sistema operativo, lo que lo hace más conveniente porque

todo lo realizado en la programación de Python y el desarrollo de la aplicación fue realizado en el entorno y junto a las librerías de Win10. El microcontrolador Raspberry pi 3 se compone de varias partes los cuales son:

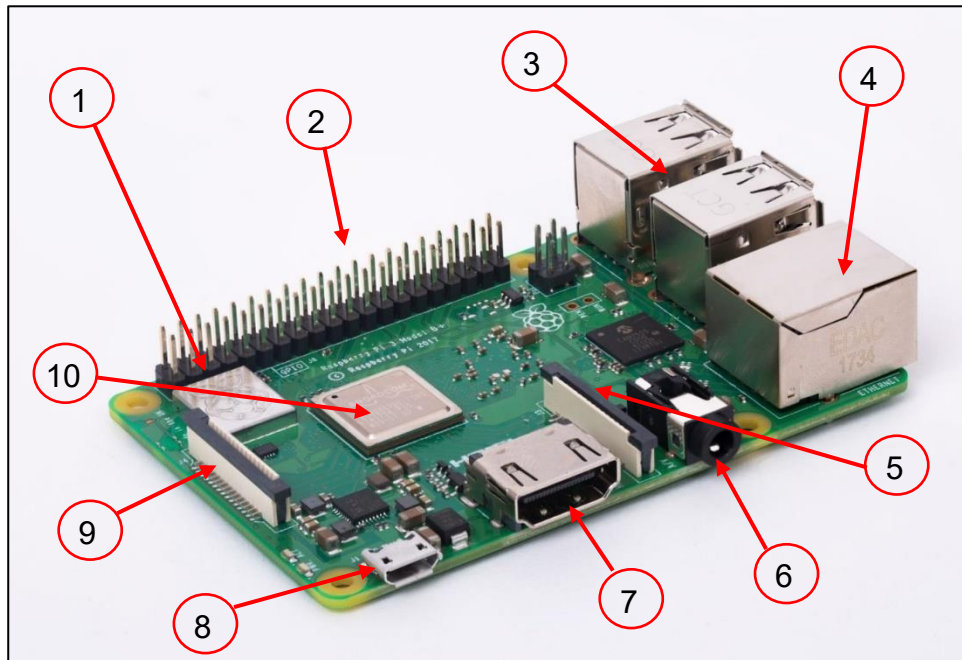


Figura 7. Microprocesador Raspberry Pi 3

1. 2.4/5 Ghz 802.11 b/g/n/ac WiFi y, Bluetooth 4.2, BLE
2. 40-Pin GPIO
3. 4 x USB 2.0
4. Puerto Ethernet
5. Puerto para cámara MIPI CSI
6. Salida de Audio
7. Puerto HDMI
8. Entrada alimentación USB 5V /2.5A
9. Puerto display
10. Cortex-A53 1.4 GHz 64-bits.

Estos puertos ayudan a su comunicación y posterior programación porque en ellos se puede conectar un teclado, un mouse mediante USB y una pantalla de monitor o cualquier dispositivo LCD que tenga una HDMI. Se mencionó en párrafos anteriores

que el microcontrolador tendrá el sistema operativo Win10 por lo que en los siguientes apartados se explicará paso a paso como realizar este proceso.

2.3 Pasos para la instalación del sistema operativo en el dispositivo

La aplicación del proyecto puede ser abierta en varios sistemas operativos tanto en Windows como aquellos basados en Linux, la condición de esto es que deben tener como mínimo requisito Python 3.9 y todas las librerías que se requieren importar para que la interfaz gráfica pueda presentar los datos procesados en la aplicación en tiempo real como:

- Tkinter: Usado para desarrollar la interfaz gráfica en el entorno de programación Python.
- Matplotlib: Librería utilizada para realizar la presentación de las gráficas de curvas.
- Pillow: Librería utilizada para importar imágenes a la interfaz gráfica.
- Numpy: Librería utilizada para realizar el análisis y operaciones matemáticas necesarias para el proyecto.

Como existen varios sistemas operativos en los cuales el programa puede ser adaptado, se describirá los pasos para trabajar con ambos según sea el caso y la conveniencia del usuario empezando por el sistema operativo Windows 10 y terminando como segunda opción en un entorno de programación basado en Linux como lo es “Raspbian”, un sistema operativo para Raspberry pi, para el proyecto realizado se eligió la segunda opción.

2.3.1. Primer método: Sistema Operativo Windows 10

Para iniciar, se debe tener una tarjeta MicroSD que tenga mínimo 32 GB de memoria y tenga una velocidad de envío y recepción de datos igual o mayor a 120 MB/s, esto se debe a que el sistema operativo Win10 tiene peso y puede llegar a tener lentitud en ciertos aspectos por no ser una versión oficial si no una versión adaptada para este tipo de dispositivos, en este caso se utilizará una tarjeta de memoria MicroSD de 64 GB Sandisk de 120 MB/s. La tarjeta de memoria cumplirá la función de “Disco Duro” y ayudará en la memoria virtual para la memoria RAM del dispositivo.



Figura 8. Tarjeta de memoria MicroSD Sandisk 64 GB

Para realizar la instalación de Win10 en Raspberry pi 3 se pueden tomar varios métodos, la manera descrita a continuación es la cual se decidió para obtener la imagen de Windows en ARM quien es el único tipo de archivo capaz de ejecutar el sistema operativo. Para obtener la imagen ISO sin inscribirse al programa de Windows Insider, el cual es una suscripción que otorga ediciones especiales del software a desarrolladores y programadores haciéndose miembro del programa para evitar esta opción se puede descargar la imagen ISO requerida por medio de una página web alternativa, la página UUP dump permite descargar versiones alternativas del sistema operativo.

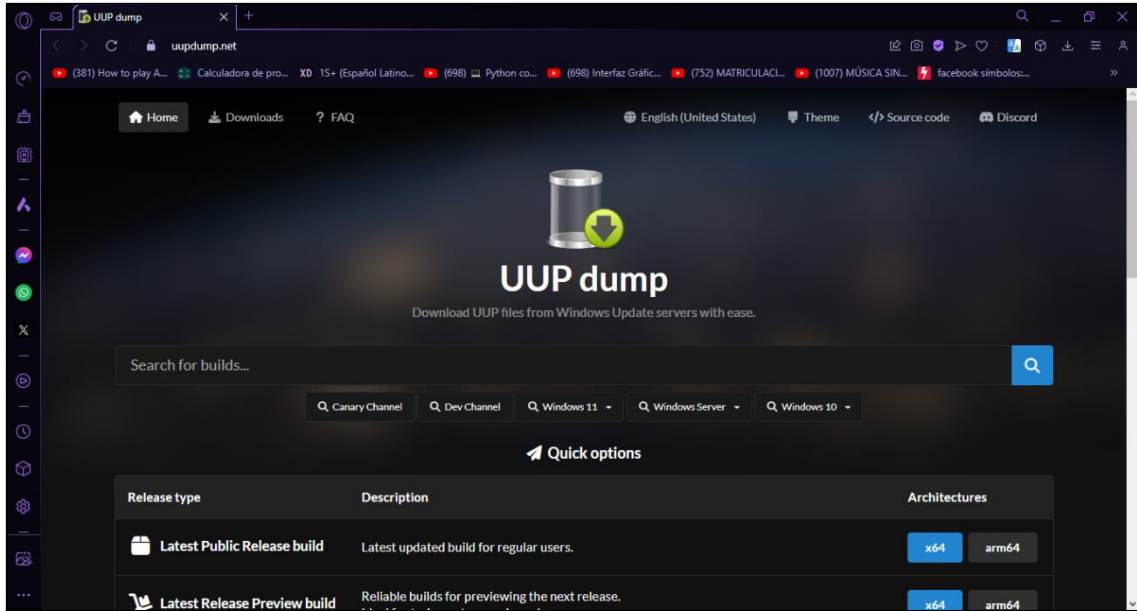


Figura 9. Página web UUP Dump

Como primer paso entramos a la página web y seleccionamos la opción “Windows 10” luego seleccionamos la arquitectura “arm64”, lo que nos mostrará solo un resultado donde permanece la versión estable y disponible del Win10 para procesadores ARM como lo es el microprocesador Raspberry pi 3, en esta ocasión se seleccionará la imagen ISO “Feature update to Windows 10, version 22H2 (19045.3996) arm64”, al hacerlo nos redirige a una página donde debemos seleccionar el idioma al cual queremos el sistema operativo, en nuestro caso “Spanish (Mexico)” por ser un español latinoamericano y le damos clic en “Next”.

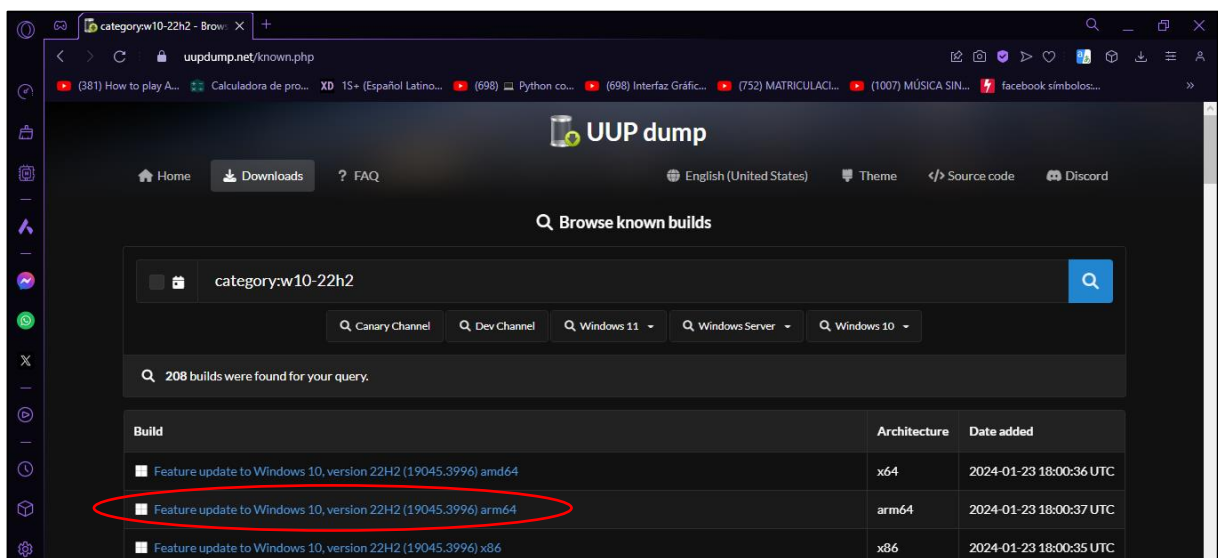


Figura 10. Selección del arm64 de Win10

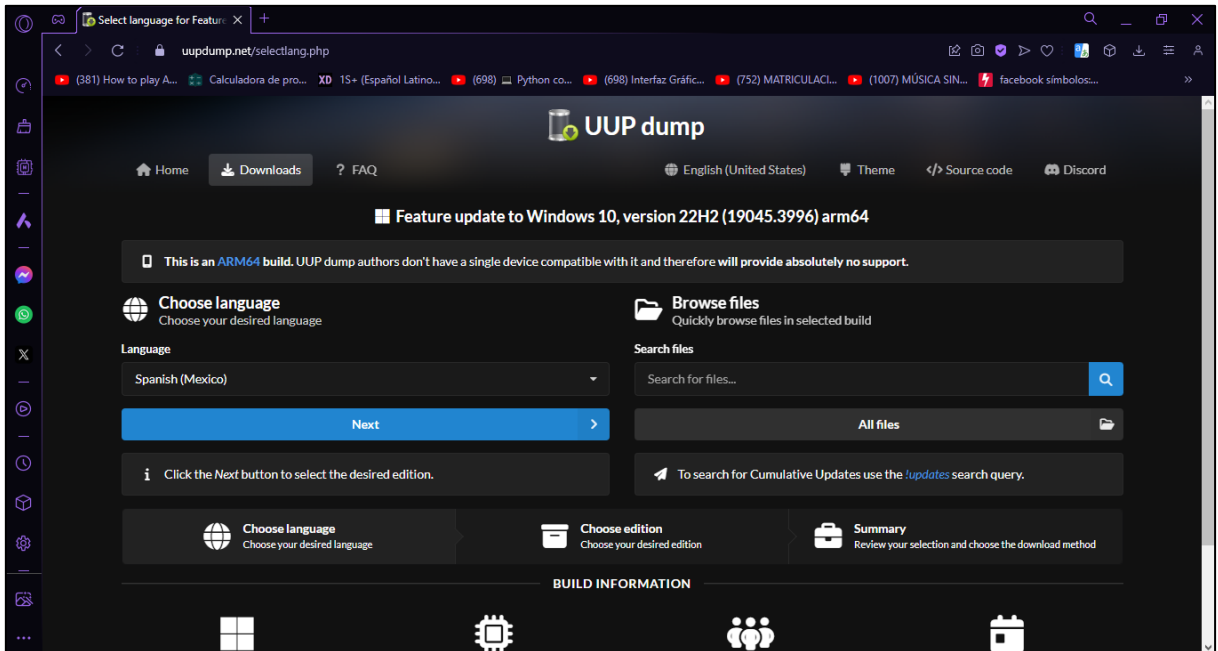


Figura 11. Selección del Idioma para el sistema operativo

Luego de realizar los pasos anteriores la página redirige hacia una nueva pestaña donde para continuar se debe dejar seleccionados los vistos en las opciones “Windows Home” y “Windows Pro”, luego de asegurarnos que se mantenga esto activado se debe dar clic en “Next”.

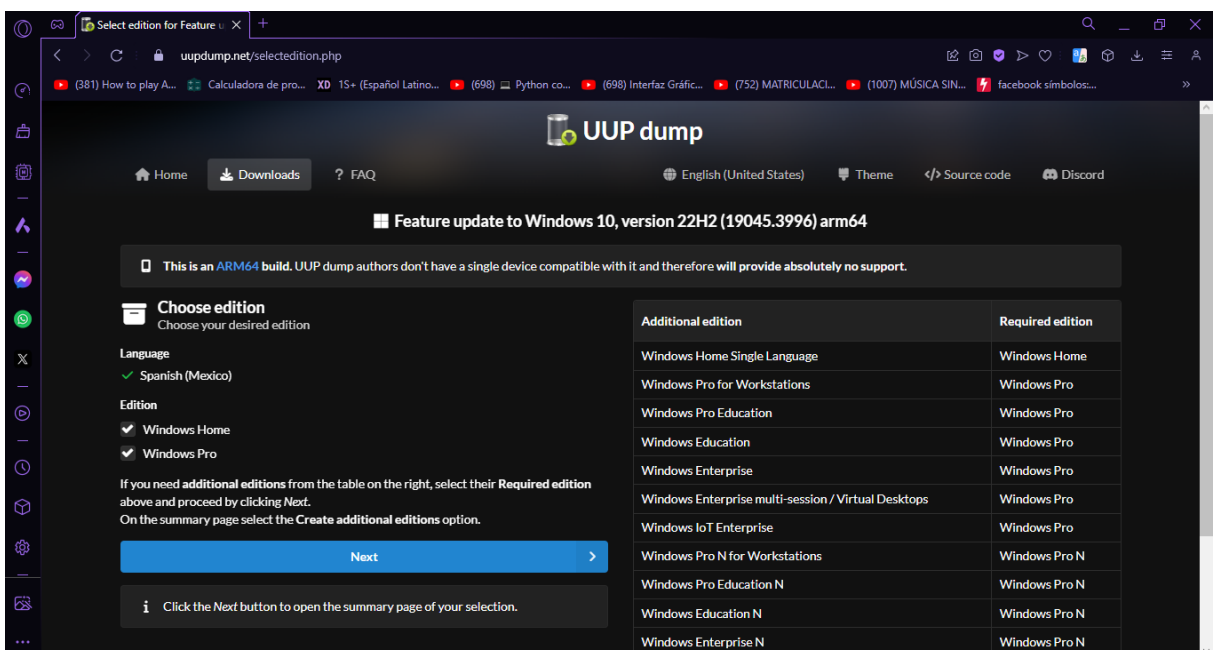


Figura 12. Activación de ambas ediciones del sistema operativo Windows 10

Como penúltimo paso antes de la descarga se debe asegurar que se encuentra seleccionada la opción “Download and Convert to ISO” para que se descargue el sistema operativo y poder utilizarlo en nuestra Raspberry con el software de Windows.

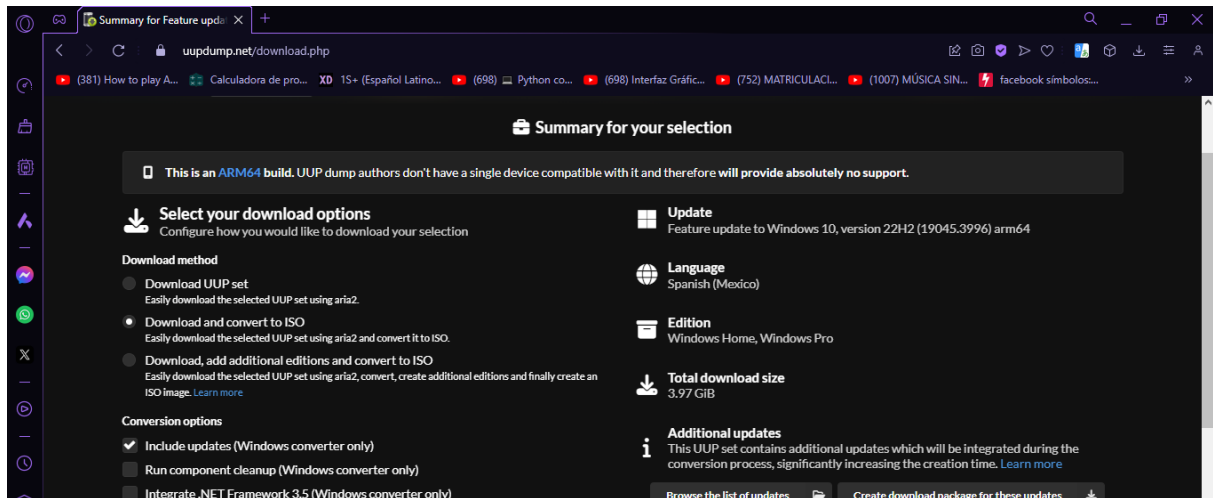


Figura 13. La opción Download and Convert to ISO está marcada

Luego de haber realizado la descarga se mueve el archivo .zip a una carpeta raíz del disco duro, puede ser en el disco C: como en algún otro ingresado, en este caso se ha copiado dentro de una carpeta llamada Win10 del disco E:

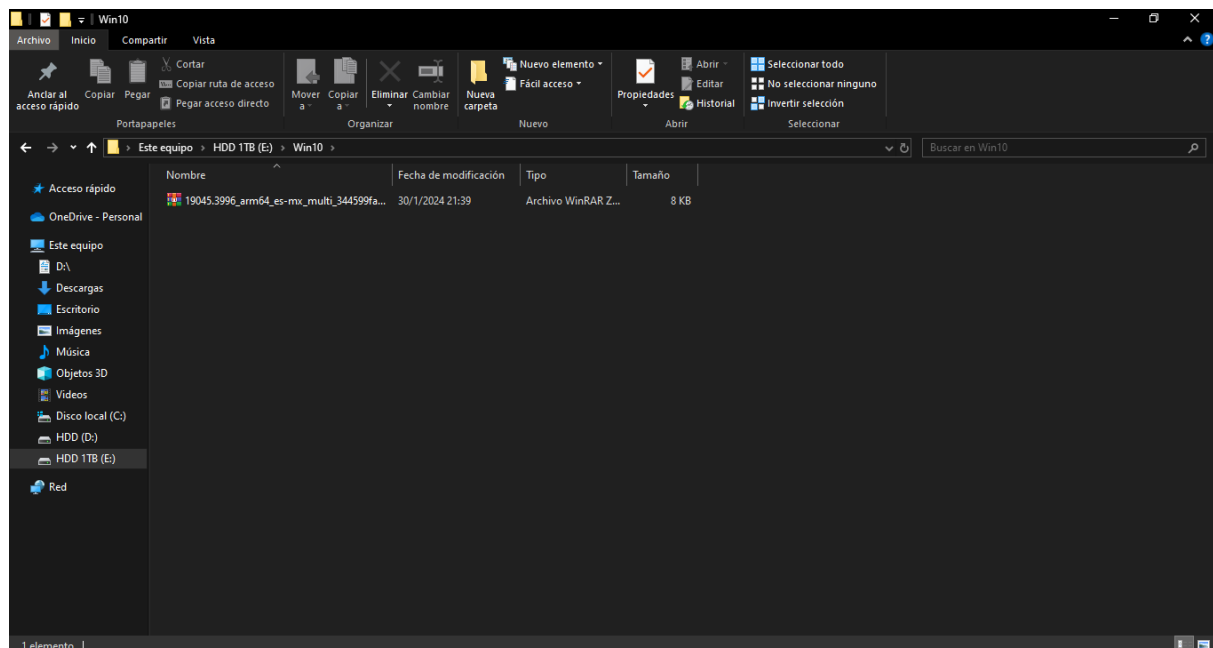


Figura 14. Archivo .ZIP dentro de una carpeta sin espacios dentro del disco E:

Se descomprime el archivo en la misma carpeta, en este caso dentro de la carpeta Win10 y ejecutamos “uup_download_windows.cmd” haciendo doble clic, como

siguiente paso puede aparecer una ventana preguntando si quieres ejecutar el programa para realizar cambios en Windows.

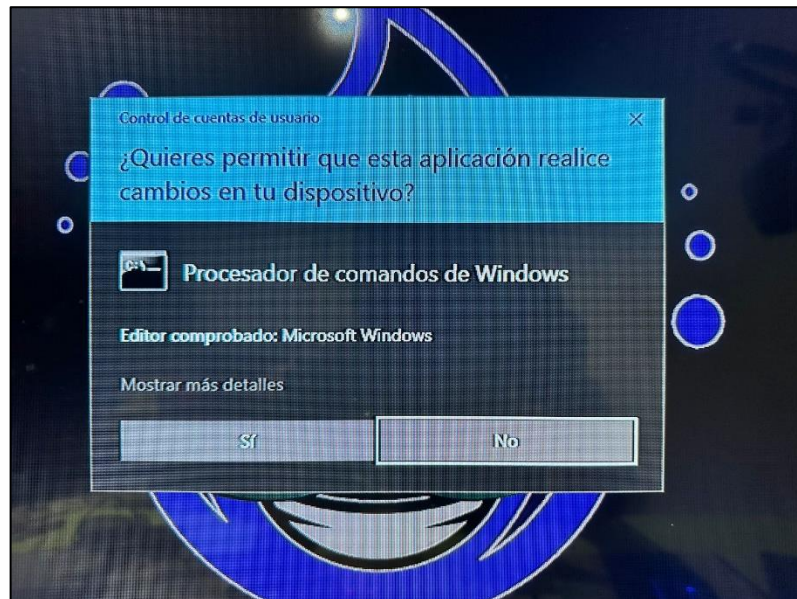


Figura 15. Mensaje de alerta del procesador de comando de Windows

O también, puede aparecer una ventana con un mensaje de Windows que nos impide ejecutar el archivo. Debemos dar clic en “Más información” y luego clic en “Ejecutar de todas formas”. Luego de realizar este proceso, aparecerá una ventana .cmd que empezará la descarga de archivos, cuando todo esto termine, se deberá pulsar el “0” y luego de esto podremos encontrar la imagen ISO del sistema operativo Win10 en la misma carpeta, en nuestro caso Win10.

Para poder instalar el sistema operativo en la memoria es necesario tener el programa “Windows on Raspberry” el cual se puede descargar desde la página web “worproject.com” en el apartado de descargas con el nombre de “Windows on Raspberry imager” descargando la versión actual del software el cual vamos a utilizar en la siguiente parte. Al tener listo el archivo de imagen ISO el siguiente paso es instalar Windows 10 en el microprocesador Raspberry Pi 3, por lo que se tiene que conectar la tarjeta MicroSD en la ranura de la PC y borrar (Formatear) el contenido que se tenga almacenado en la memoria.

Abrimos el programa WoR.exe, seleccionamos el idioma y daremos clic en “Siguiente”, luego de esto, seleccionaremos la unidad donde se encuentra la MicroSD y también se seleccionará el modelo del dispositivo, en nuestro caso un Raspberry Pi 3, posteriormente daremos clic en “Siguiente”.

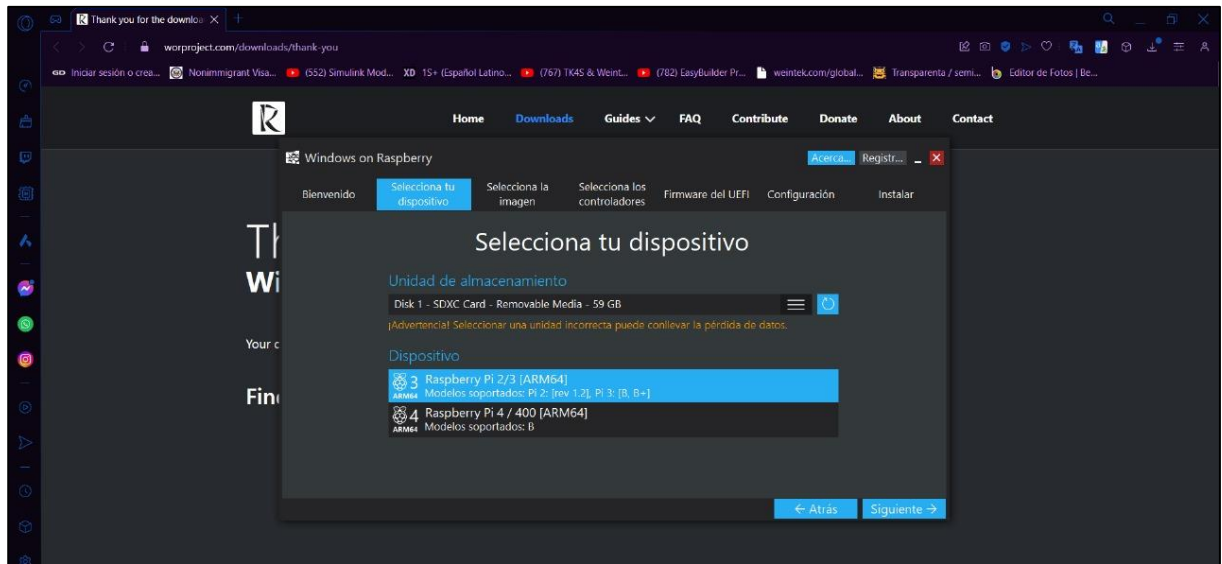


Figura 16. Configuración en el programa Windows on Raspberry

Luego de esto, seleccionaremos la imagen ISO de Win10 que descargamos y seleccionamos la edición “Home”, luego daremos clic en “Siguiente” y aceptamos los permisos.

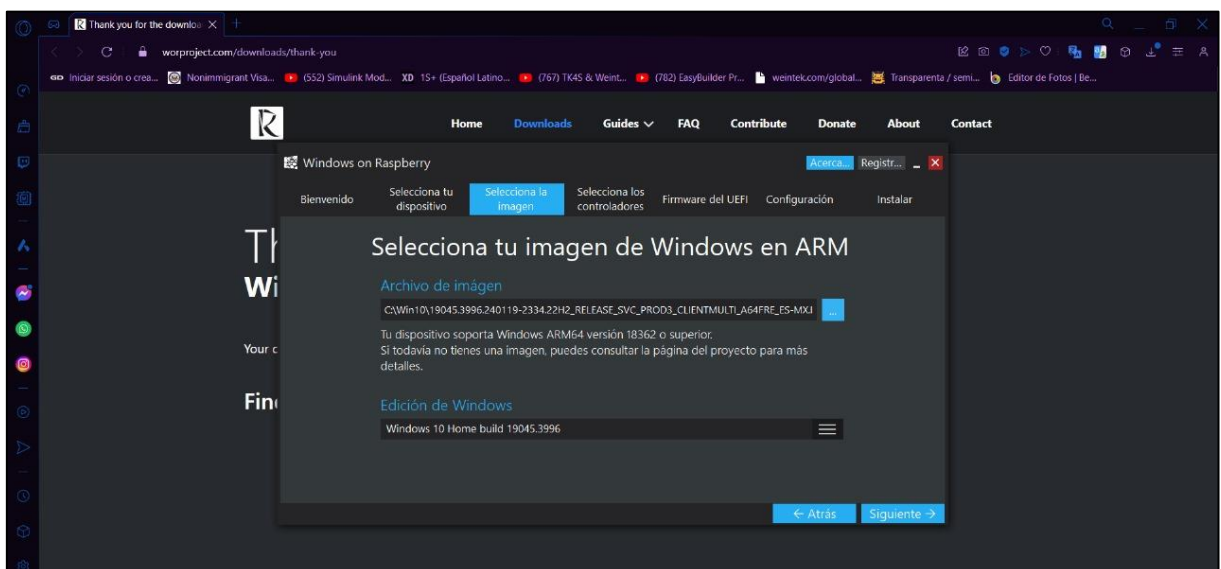


Figura 17. Montando la Imagen ISO en la tarjeta MicroSD

Como último paso se da clic en “Instalar” y se espera hasta que el programa haya instalado la Imagen del sistema operativo en la tarjeta MicroSD, la cual estará lista para ser colocada en el microcontrolador Raspberry Pi 3 y posteriormente configurar Win10.

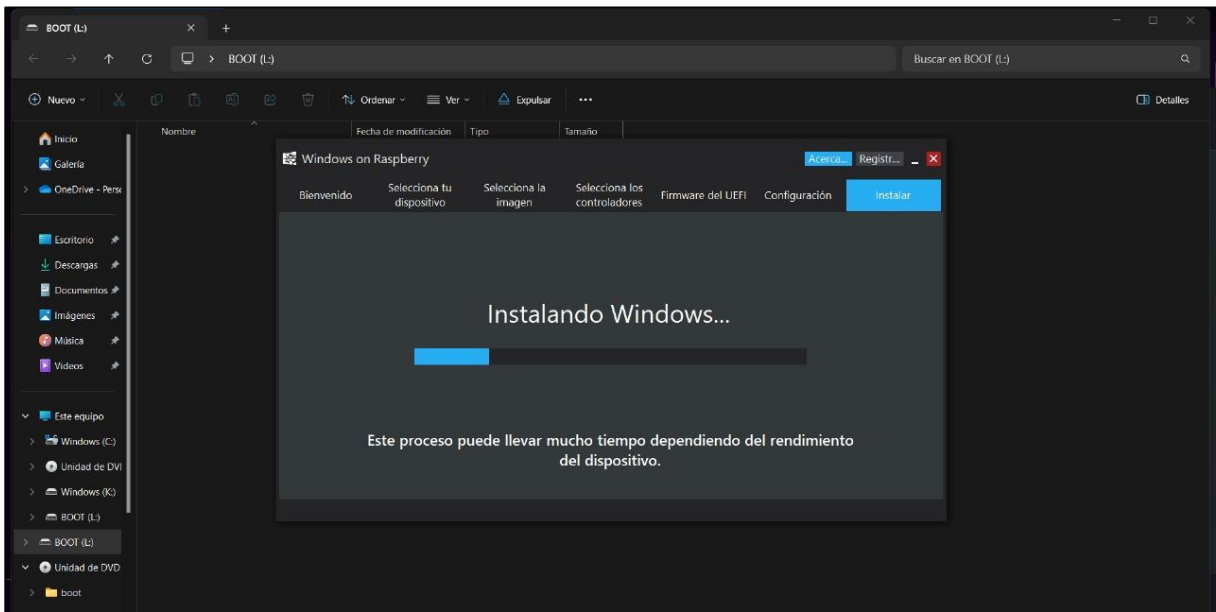


Figura 18. Instalación del sistema operativo en la MicroSD como sistema Booteable

2.3.2 Segundo Método: Sistema Operativo Raspbian

Se eligió la segunda opción como sistema operativo base porque Raspbian tiene los requisitos mínimos para que la aplicación desarrollada funcione sin ningún problema, ya que, el sistema base de Linux tiene el controlador CH340-1 el cual es un requisito necesario para que el sensor de vibración se conecte al dispositivo, el lenguaje de programación Python es uno de los elementos que contienen este sistema por lo que se optó como una opción viable para la implementación.

Para instalar Raspbian en el dispositivo Raspberry Pi 3 se debe navegar por la página principal del dispositivo con el link: www.raspberrypi.org donde el usuario deberá dirigirse a la pestaña de “Software” y seleccionar el botón “Download for Windows”, si el usuario está utilizando el sistema operativo Windows o “Download for MacOs”, si el usuario está utilizando el sistema operativo diseñado por Apple o “Download for Ubuntu for x86”, si el usuario está utilizando un sistema operativo basado en Linux.

Para el caso del proyecto se está utilizando Windows 10 como sistema operativo por lo que al dar clic en descargar se bajará un archivo “.Zip” con el nombre de “imager 1.8.5” el cual es una herramienta de instalación del sistema operativo Raspbian.

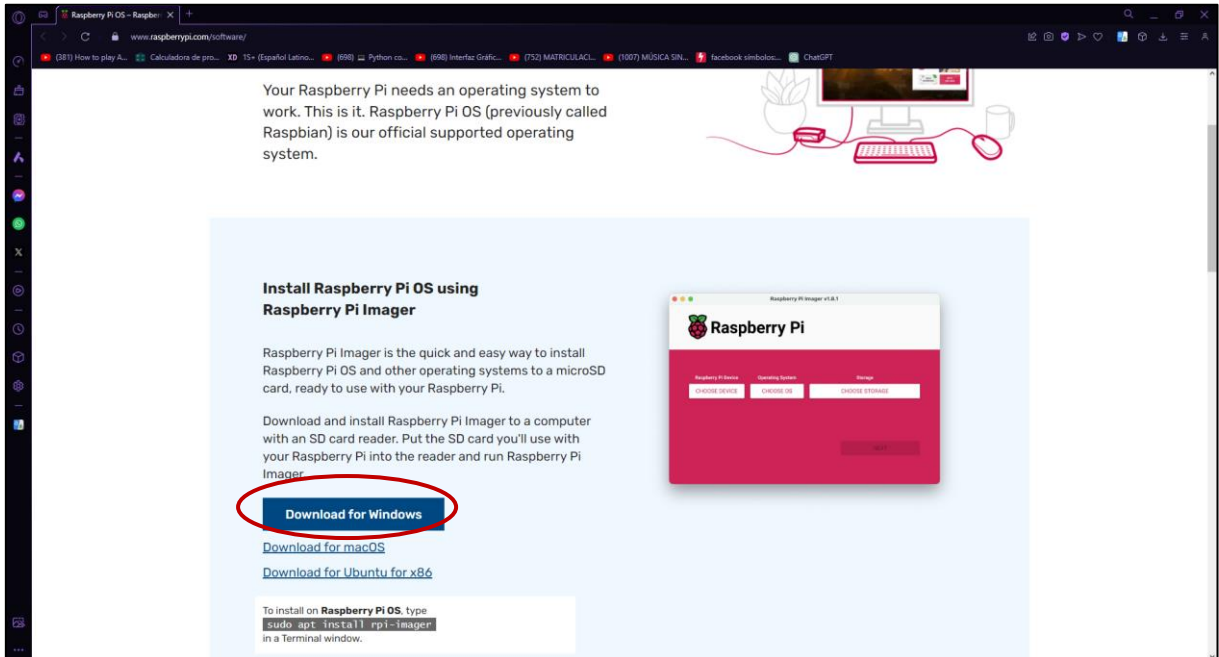


Figura 19. Página de descarga del instalador de Raspbian

Luego de la descarga aparecerá un .exe, el cual es para poder instalar el sistema operativo en el dispositivo, luego de darle doble clic aparecerá una ventana de instalación como la presentada a continuación:



Figura 20. Ventana del instalador para el sistema operativo Raspbian

Después de darle clic en “install” se procederá a realizar la instalación del programa, lo que tardará un momento para luego ejecutarlo y proceder a los siguientes pasos de la instalación:



Figura 21. Programa instalador de la imagen para el sistema operativo Raspbian

Luego de ejecutar el programa se abre una ventana como la **Figura 24** donde se puede observar tres opciones: Dispositivo Raspberry Pi, Sistema Operativo y Almacenamiento; al dar clic a la primera opción se desplegará una pequeña ventana donde podremos seleccionar el tipo de dispositivo Raspberry que tenemos en posesión, en este caso se seleccionará “Raspberry Pi 3”.

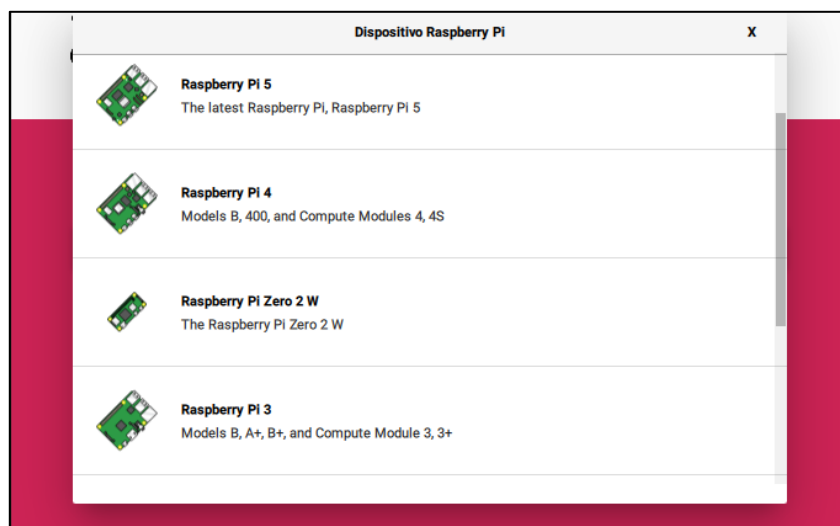


Figura 22. Selección de opción Raspberry en posesión.

Como siguiente punto se elige el sistema operativo que se utilizará, como se puede visualizar en la **Figura 26** se recomienda un sistema operativo llamado “Raspberry Pi Os (Legacy, 32-bit), pero como recomendación para el dispositivo que se tiene y para el uso que se le dará es mejor seleccionar la opción “Raspberry Pi Os (other) en el cual se podrá observar otros sistemas operativos Raspbian de 32 hasta 64 bits y con sus versiones tanto lite como full; para el siguiente proyecto se seleccionar la opción de “Raspberry Pi Os (Legacy, 32-bit) Full”.

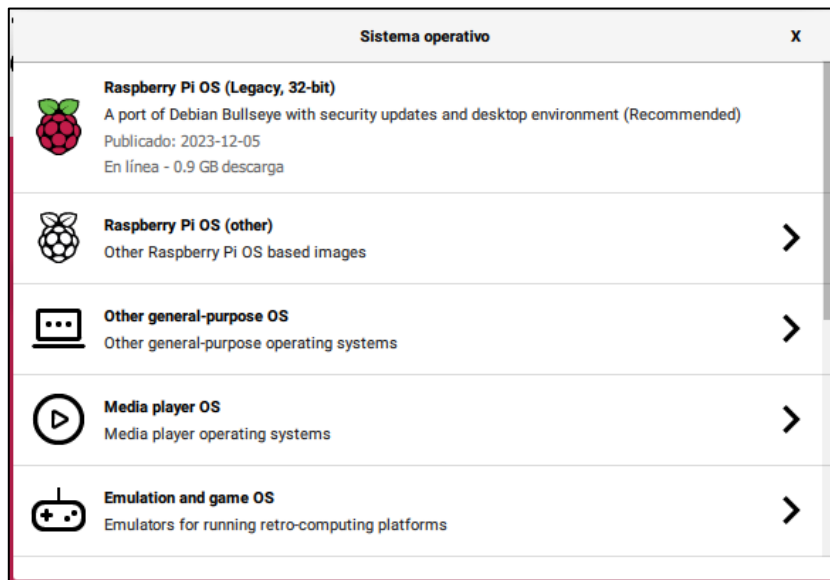


Figura 23. Sistemas Operativos a seleccionar.

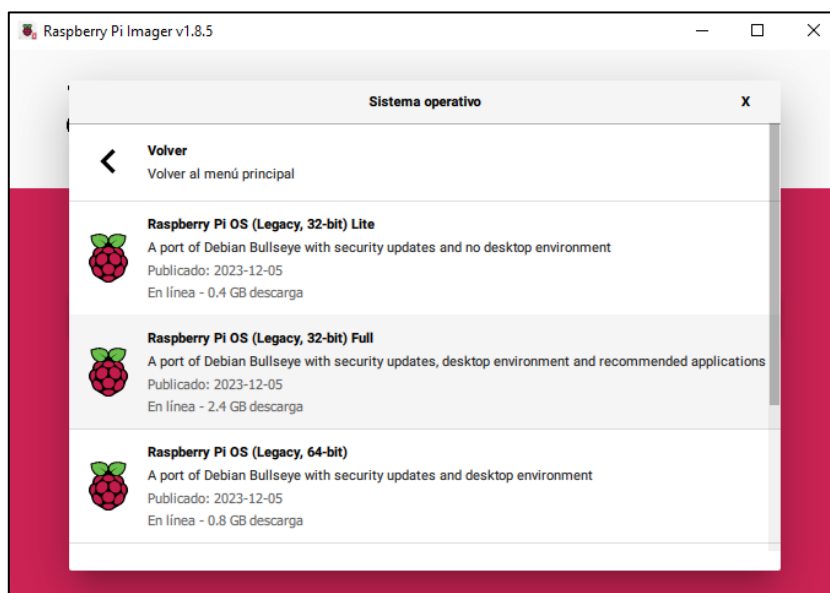


Figura 24. Elección Sistema Operativo Raspberry pi Os (Legacy, 32-bit) Full.

Como última parte se ingresa la tarjeta MicroSD en la computadora y se verifica el nombre del dispositivo de almacenamiento para luego elegirlo en el programa de instalación del sistema operativo Raspbian y posteriormente dar clic en “Siguiete” donde aparecerá una ventana nueva como “Opciones Avanzadas” las cuales se puede colocar el nombre al dispositivo local, guardar y posteriormente “instalar”.

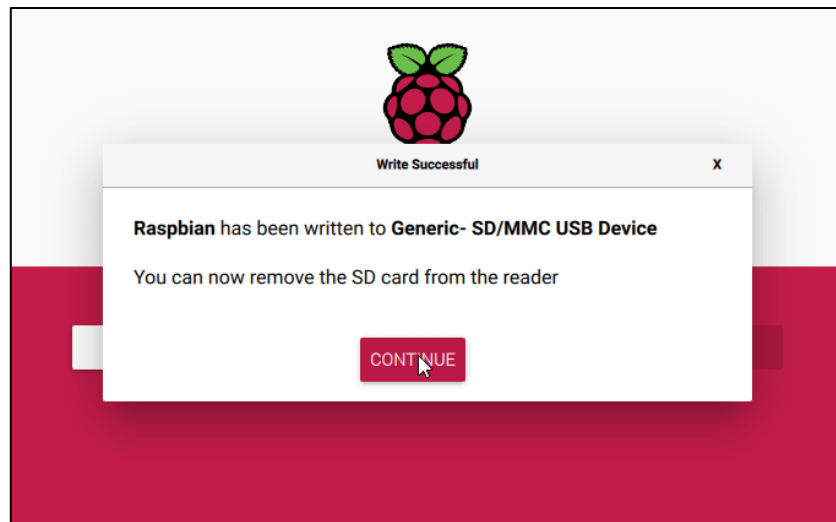


Figura 25. Ventana Final de la instalación del sistema operativo en la tarjeta MicroSD

Luego de realizar la instalación del sistema operativo en el dispositivo de almacenamiento se procede a colocar la tarjeta MicroSD dentro del dispositivo y se enciende el Raspberry pi 3 obteniendo la primera imagen en la pantalla HMI.

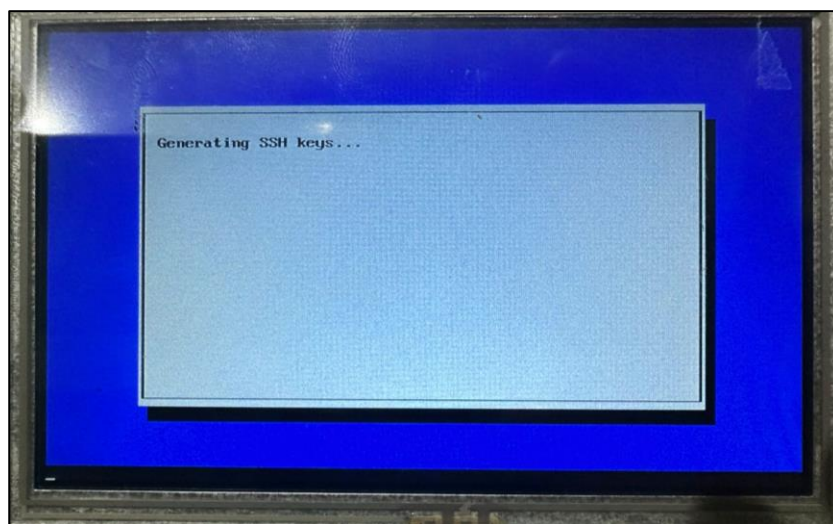


Figura 26. Primera imagen del dispositivo Raspberry pi 3 con el sistema operativo.

Luego de esperar unos segundos a que cargue el software en el dispositivo nos da como primera imagen una ventana que da la bienvenida al sistema operativo antes de su configuración como se visualiza en la **Figura 30**.

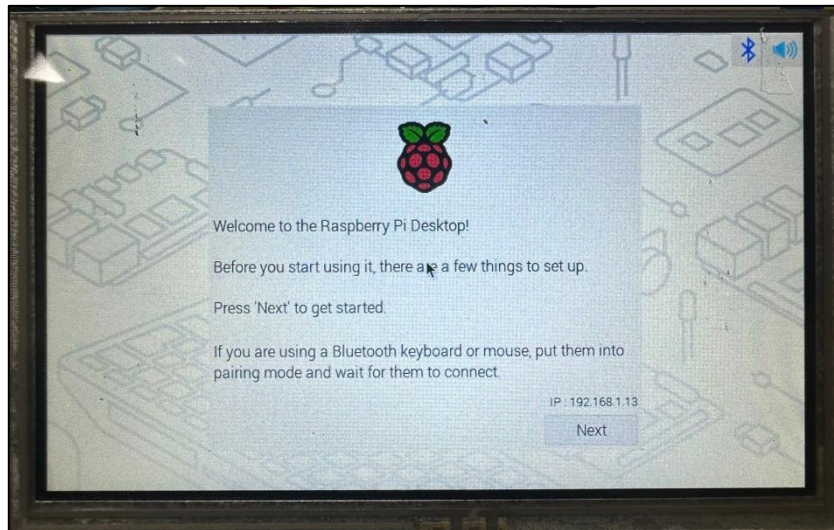


Figura 27. Bienvenida del Software Raspbian

Al dar clic en el botón "Next" se ingresa a una nueva ventana donde se debe configurar: el país donde se encuentra el usuario, el lenguaje de preferencia y la zona horaria seleccionándolo por una lista de opciones.

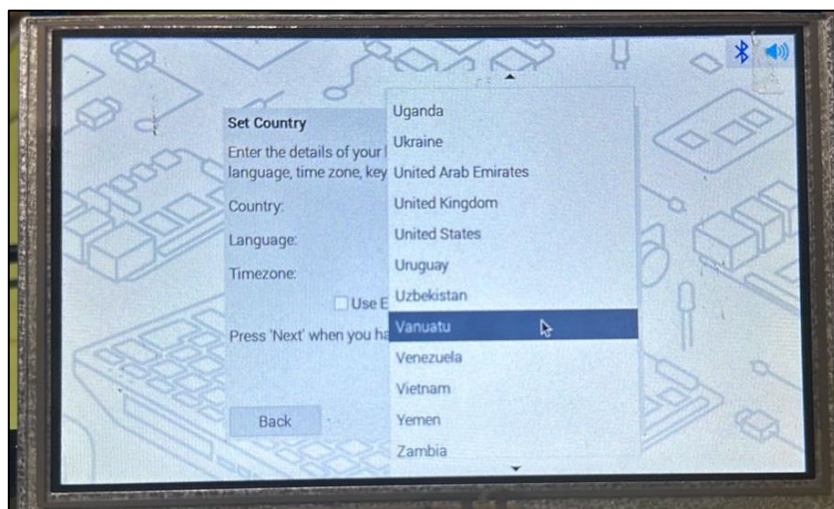


Figura 28. Configuración de idioma y parámetros básicos iniciales.

Luego de realizar la configuración es necesario crear un usuario para la cuenta del sistema operativo, en este caso se ingresó el nombre de usuario: "jorlemor" y como contraseña se utilizó caracteres básicos y fáciles de recordar como: "L12345", luego de realizar este paso el usuario debe dar clic en "Next".

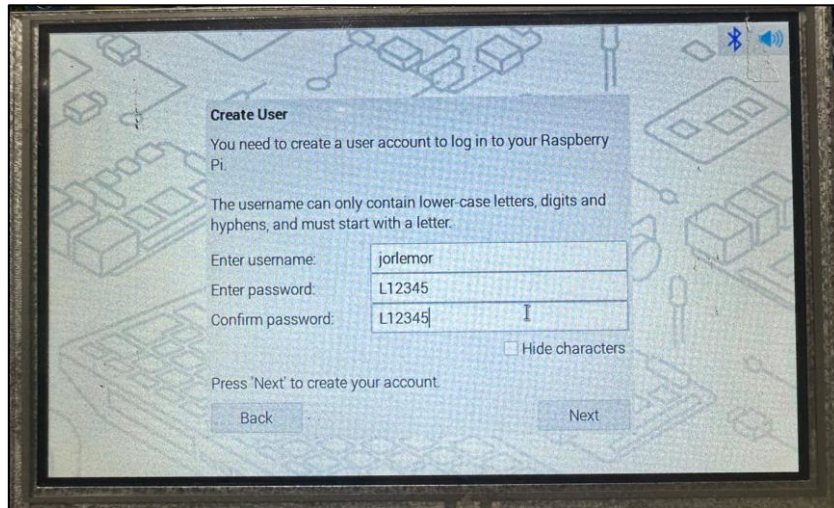


Figura 29. Configuración del usuario para la cuenta en el dispositivo.

Como siguiente punto se debe configurar el wifi del dispositivo, el cual se puede saltar con el botón "Skip" o continuar a la siguiente configuración donde se colocará la contraseña de la conexión a internet elegida. Este paso es necesario para descargar e instalar las actualizaciones en el sistema operativo.

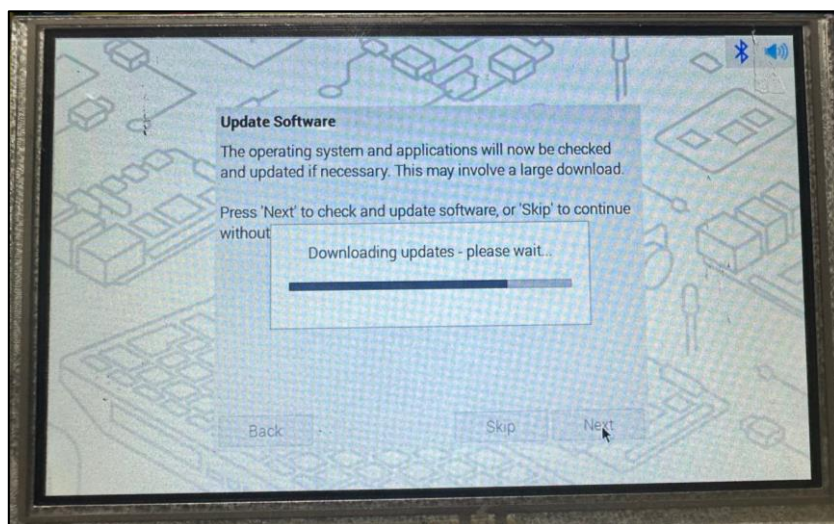


Figura 30. Actualizaciones necesarias para el sistema operativo del dispositivo.

Después de actualizar el sistema operativo del dispositivo, aparecerá una mini ventana con el texto "System is up to date" donde se dará clic en "OK" y pasaremos a la siguiente ventana con el título "SETUP COMPLETE" el cual nos dará dos botones de opciones: "Back" y "Restart", se dará clic en el segundo botón de opción como se visualiza en la **Figura 34**.

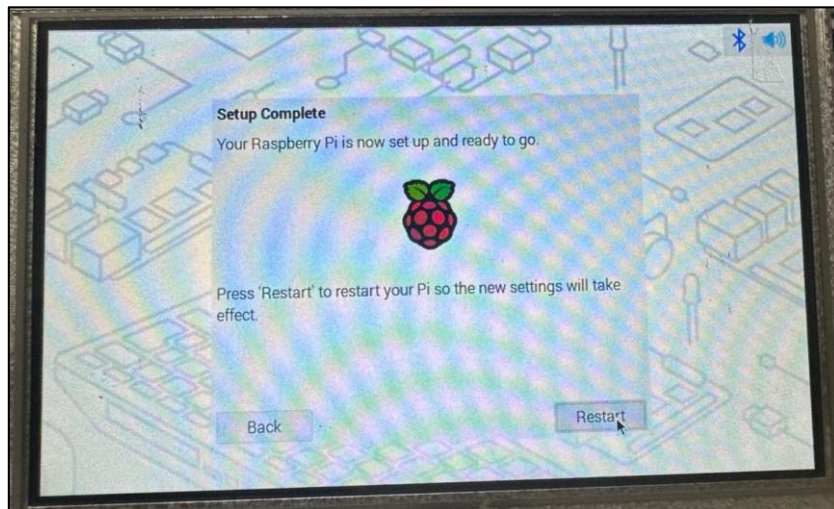


Figura 31. Última ventana antes del reinicio del sistema operativo con las actualizaciones.

Luego de que el dispositivo se reiniciará aparecerá una ventana que da la bienvenida al sistema operativo Raspberry Pi OS (32-bit).

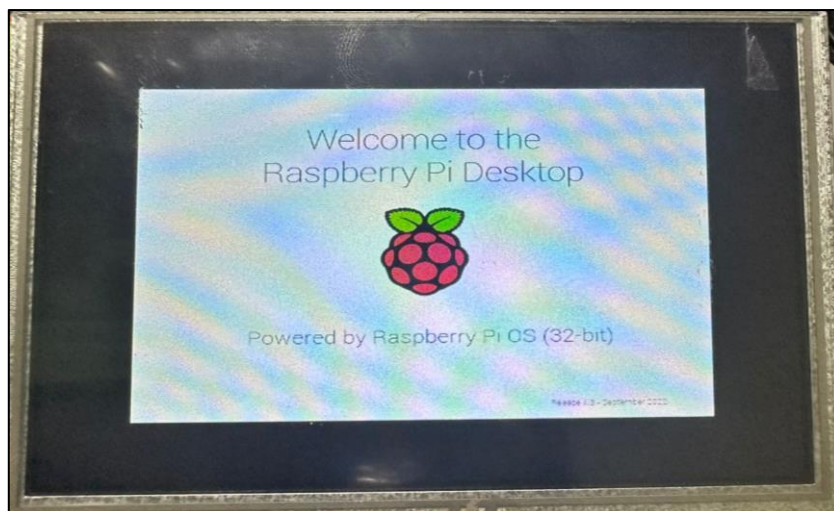


Figura 32. Bienvenida Sistema Operativo Raspberry pi Os

Finalmente aparecerá la pantalla de “Escritorio” del sistema operativo instalado el cual estará listo para realizarle la instalación y/o actualización de las librerías y otros componentes necesarios para el programa desarrollado.



Figura 33. Pantalla de Escritorio del Sistema Operativo Raspbian

2.4. Instalación y actualización de los componentes necesarios para el programa desarrollado.

En primer lugar, se verifica que el sistema operativo cuente con las librerías y los controladores necesarios para el proyecto como el lenguaje de programación Python y el controlador CH340-1, para esto es importante que el usuario abra el “Terminal” del dispositivo y digite las líneas:

“sudo apt-get-update”

Lo cual nos ayudará a actualizar todas las librerías y componentes que integran el sistema operativo Raspbian, posteriormente a eso se procede a verificar la versión del lenguaje de programación Python con las líneas en el terminal:

“Python3 –version”

El dispositivo debe regresar un mensaje de resultado con la versión de Python preinstalada en el sistema operativo, para el caso del proyecto la versión instalada es Python 3.9 cual cumple con los requisitos mínimos necesarios para correr el programa desarrollado.

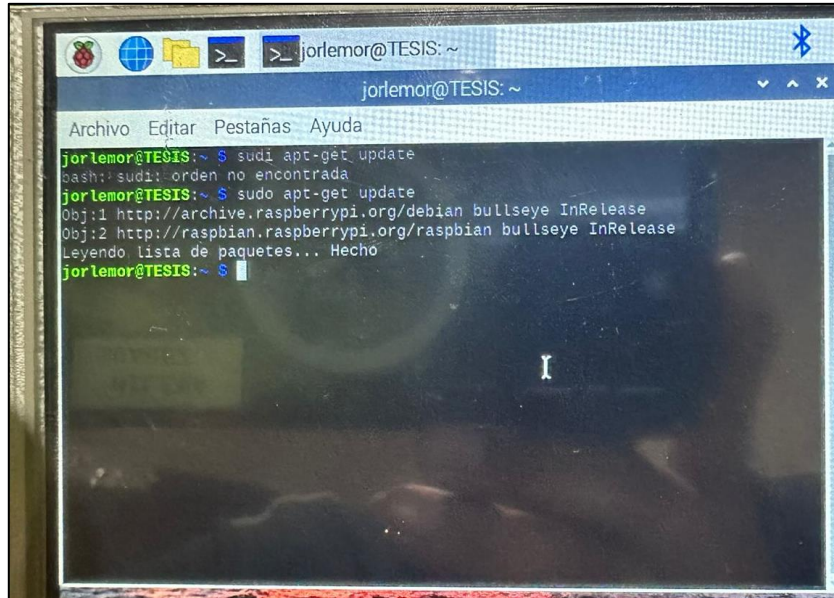


Figura 34. Actualización de las librerías y componentes del sistema operativo.

Finalmente se procede a guardar los datos de todas las librerías actualizadas y cambios realizados los cuales fueron llamados anteriormente mediante el comando:

“sudo apt-get upgrade”

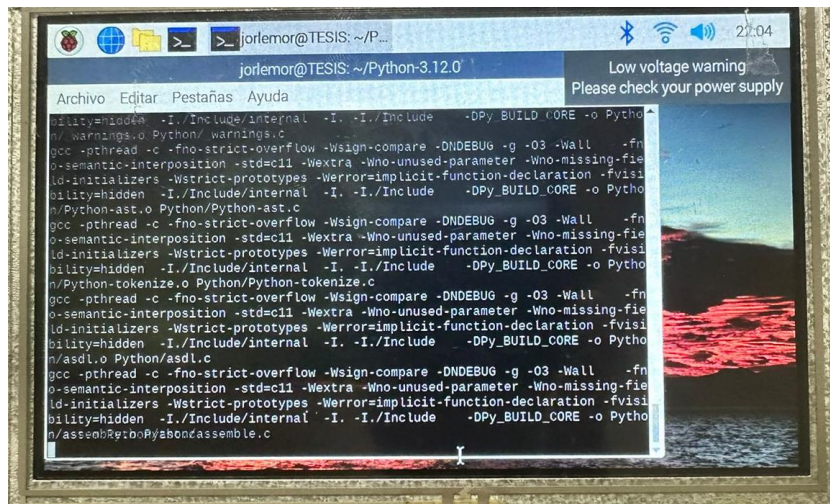


Figura 35. Actualización del paquete del sistema operativo.

Luego de seguir los pasos para actualizar el sistema operativo es necesario instalar las librerías de Python requeridas para que el programa desarrollado no tenga inconvenientes y pueda funcionar en el entorno de Raspbian porque, el programa fue

desarrollado en Python en el entorno base de Windows 10 por lo que es necesario realizar cambios en la programación para que este se adapte a la pantalla de 5" HDMI colocada para el tablero de monitor de vibraciones.

Para instalar las librerías de Python necesarias se procede a instalarlas desde el terminal de Raspbian, tal y como se procedería a realizar en el sistema operativo de Win10, es decir, utilizando la herramienta "pip" para la instalación o actualización de las nuevas librerías digitando el código a continuación:

- Para la librería **Numpy**:
"pip install numpy"
- Para la librería **Matplotlib**:
"pip install matplotlib"
- Para la librería **Pillow**:
"pip install pillow"
- Para la librería **MinimalModbus**:
"pip install minimalmodbus"
- Para la librería **Tkinter**:
"pip install tkinter"

Las librerías adicionales que se encuentran dentro de la programación de Python del programa desarrollado ya vienen por defecto por esa razón no se importan desde el terminal como las anteriores mostradas, a menos que, se necesiten actualizar. Todas las líneas con las que comienza el código de Python las cuales comienzan con "import" deben ser instaladas porque esas son las librerías que necesita el programa desarrollado para funcionar sin problemas.


```

import minimalmodbus
import tkinter as tk
from tkinter import ttk
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
import numpy as np
import threading
import time
from PIL import Image, ImageTk

```

Figura 36. Librerías necesarias para el correcto funcionamiento del programa desarrollado.

Finalmente, luego de terminar la instalación, verificación de librerías y actualización de estas se procede a migrar el programa desarrollado en el ambiente de programación en el sistema operativo Windows 10 al sistema operativo Raspbian, por el cual se tiene el primer aspecto del programa como se visualiza en la **Figura 40**.

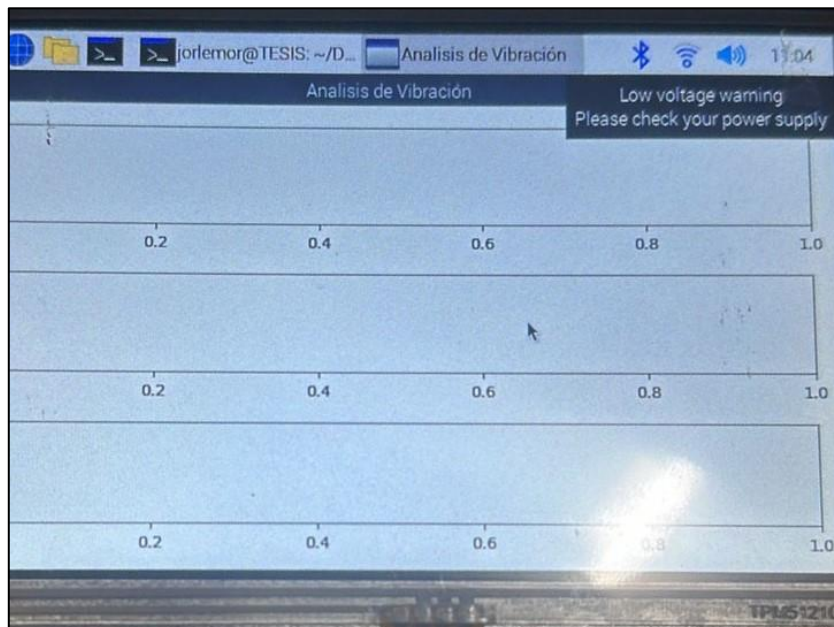


Figura 37. Primera visualización del programa desarrollado en el sistema operativo Raspbian.

Como existe gran diferencia entre las pantallas del sistema operativo y el HMI utilizado para el proyecto se procede por conveniencia a modificar la interfaz gráfica para que

se pueda manejar de mejor forma dentro de la pantalla de cinco pulgadas con resolución de 800x480, quedando como se muestra en la **Figura 41** donde se puede visualizar los botones de: “Iniciar Recopilación”, “Detener Recopilación”, “Calcular transformada de Fourier”, “Reiniciar”, “Seleccionar la Clase” y la “Selección de los ejes para la transformada de Fourier” en la parte superior de las gráficas.

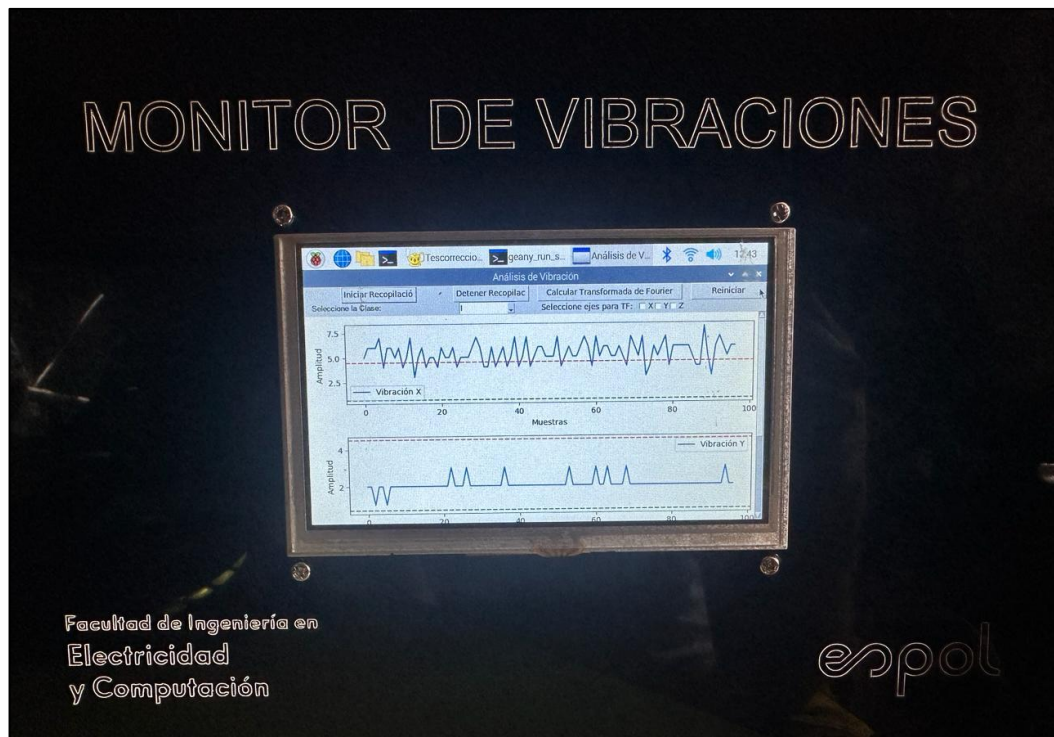


Figura 38. Sistema de Monitoreo Vibracional con HMI.

CAPITULO 3

3. RESULTADOS

3.1. Desarrollo y Aplicación de Código de las fallas en motores

Inicialmente se desarrollaron los códigos para los diferentes eventos mencionados en el apartado anterior, este fue el procesador de datos básico y rudimentario que se tuvo debido a que no se contaba con datos reales ya que era una simulación básica programada en Python. Los datos iniciales estaban dados por valores aleatorios haciendo referencia a datos de vibración con ciertas características, sin embargo, se implementó una perturbación a los valores aleatorios de tal manera que esta perturbación representara la falla.

Falla de desbalance

Como primera falla se tiene al ejemplo de desbalance, el código de esta perturbación consistió en modificar ciertos valores con una amplitud mayor a la que normalmente se simulaba en un rango corto de valores, como valores preliminares se colocó como, velocidad de rotación del motor de inducción de 1800 rpm, frecuencia de desbalance en 30Hz y amplitud de desbalance en 0.5, estos valores se encuentran en el dominio del tiempo, a continuación, se presenta el código utilizado.

```
# Añadir una señal de desbalance a los datos de vibración simulados
# Puedes ajustar los parámetros del desbalance según tus necesidades
amplitud_desbalance = 0.5
frecuencia_desbalance = 30 # Frecuencia de desbalance en Hz
velocidad_rotación = 1800 # RPM
t = np.arange(len(vibraciones_tiempo_real))
desbalance = amplitud_desbalance * np.sin(2 * np.pi * frecuencia_desbalance * t / (60 / velocidad_rotación))
vibraciones_tiempo_real += desbalance
```

Figura 39. Código de la falla como señal de desbalance

Esto dio paso para calcular la transformada de Fourier para llevar los valores aleatorios al dominio de la frecuencia asignándole un muestreo de 1000 datos utilizando la función `np.fft.fftfreq` en Python tal como se muestra en la siguiente imagen.

```

# Calcula la transformada de Fourier de los datos
n = len(vibraciones_tiempo_real)
frecuencia_muestreo = 1000 # Frecuencia de muestreo en Hz

frecuencias = np.fft.fftfreq(n, 1 / frecuencia_muestreo)
espectro_frecuencia = np.abs(np.fft.fft(vibraciones_tiempo_real))

```

Figura 40. Código de la Transformada de Fourier para la falla de desbalance

De la misma manera, se calculó la transformada de Fourier de la perturbación puesto que se necesita para conocer el espectro vibratorio de la falla y poder compararla con los umbrales de la norma ISO 10816-3, el código para realizar la transformada fue el siguiente.

```

# Calcula el espectro de desbalance por separado
n_desbalance = len(desbalance)
frecuencias_desbalance = np.fft.fftfreq(n_desbalance, 1 / frecuencia_muestreo)
espectro_frecuencia_desbalance = np.abs(np.fft.fft(desbalance))

```

Figura 41. Código para hallar el espectro de falla de desbalance

Las primeras representaciones gráficas que se tienen luego de mandar a correr el código que se escribió se muestran a continuación, presentando la onda vibratoria de los valores aleatorios en el dominio del tiempo con desbalance además del espectro de frecuencia completo ya que se observa un pico con amplitud elevada en la parte inicial representando la respectiva balla. De la misma manera, en la segunda figura se muestra el espectro de frecuencia del desbalance donde se observa un pico de mayor amplitud en la frecuencia 0, a diferencia de los demás picos que presentan una amplitud menor, esto representando una perturbación de desbalance en el motor.

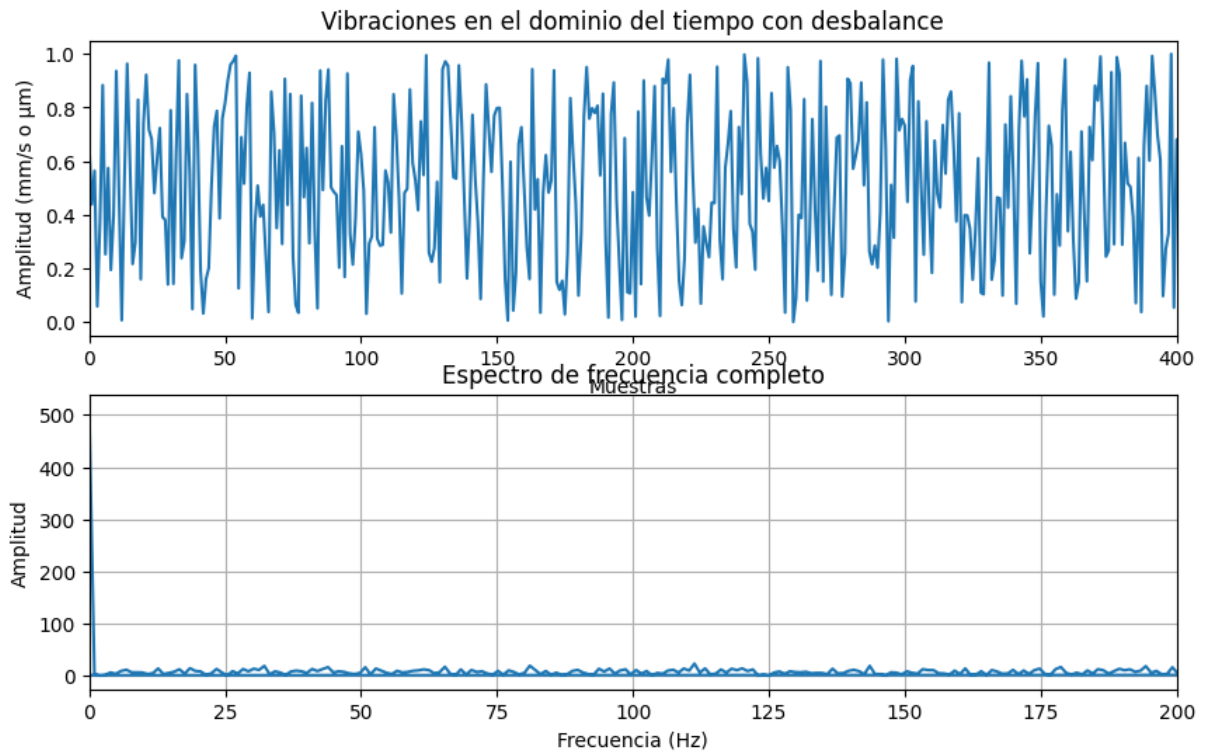


Figura 42. Vibraciones en el dominio del tiempo y espectro de frecuencia de la falla de desbalance

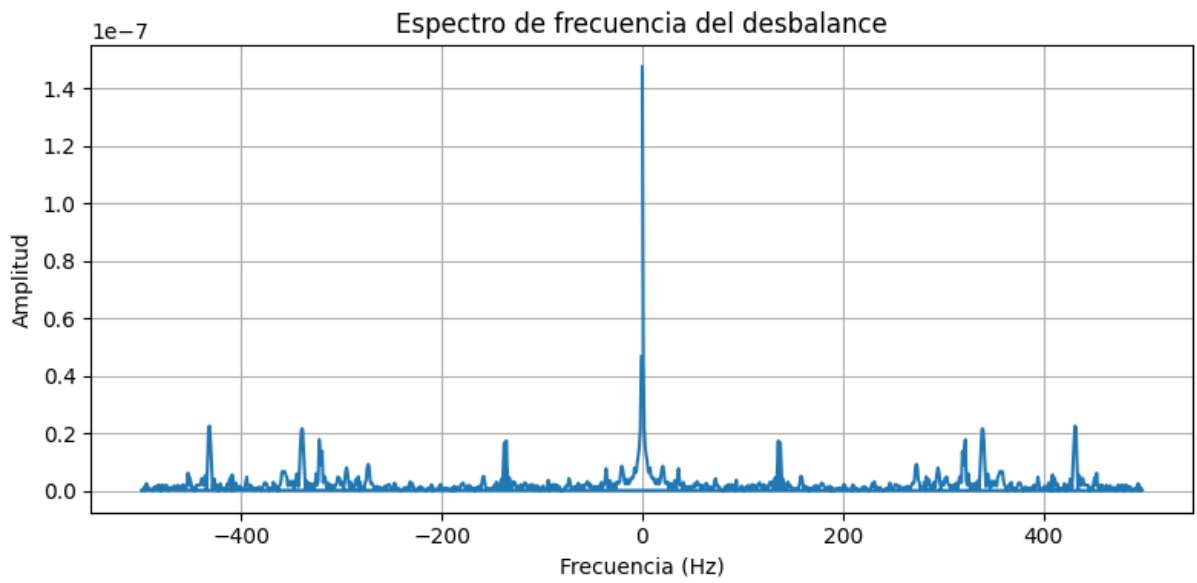


Figura 43. Espectro de frecuencia de la falla de desbalance

Falla de Excentricidad

De la misma manera se realizó el código para una falla de excentricidad con valores aleatorios, en este caso se especificó la amplitud de la excentricidad la cual se parametrizó en 0.3 además de una frecuencia de paso de la máquina de inducción que fue de 5 Hz.

Se calculó también la transformada de Fourier de la señal que formaban los valores aleatorios con la perturbación siendo esta la señal de las vibraciones en el dominio del tiempo con excentricidad tal como se muestra en la gráfica de señales, además del espectro de frecuencia completo que se encuentra debajo de la señal mencionada anteriormente.

```
# Se añade una falla de excentricidad a los datos de vibración simulados
# Puedes ajustar los parámetros de la excentricidad según tus necesidades
amplitud_excentricidad = 0.3
frecuencia_paso = 5 # Frecuencia de paso de la máquina en Hz
t = np.arange(len(vibraciones_tiempo_real))
excentricidad = amplitud_excentricidad * np.sin(2 * np.pi * frecuencia_paso * t)
vibraciones_tiempo_real += excentricidad

# Calcula la transformada de Fourier de la señal completa
n = len(vibraciones_tiempo_real)
frecuencia_muestreo = 1000 # Frecuencia de muestreo en Hz (ajusta según tus datos)

frecuencias = np.fft.fftfreq(n, 1 / frecuencia_muestreo)
espectro_frecuencia = np.abs(fft(vibraciones_tiempo_real))
```

Figura 44. Código de la falla de excentricidad y de la Transformada de Fourier

De la misma manera que el apartado anterior, se realiza el código para calcular el espectro de la falla de excentricidad mediante el rango de datos usados en las variables aleatorias usando la transformada de Fourier, tal como se muestra a continuación.

```
# Calculo del espectro de excentricidad por separado
n_excentricidad = len(excentricidad)
frecuencias_excentricidad = np.fft.fftfreq(n_excentricidad, 1 / frecuencia_muestreo)
espectro_frecuencia_excentricidad = np.abs(fft(excentricidad))
```

Figura 45. Código para hallar el espectro de la falla de excentricidad

Una vez que se tuvo el código se realizó el código utilizando la función *plt.figure* para graficar las señales vibratorias en el dominio del tiempo, así como en el dominio de la frecuencia tanto para los datos totales, así como para hallar el espectro de frecuencia de la falla de excentricidad puesto que se lo necesita para realizar la comparación.

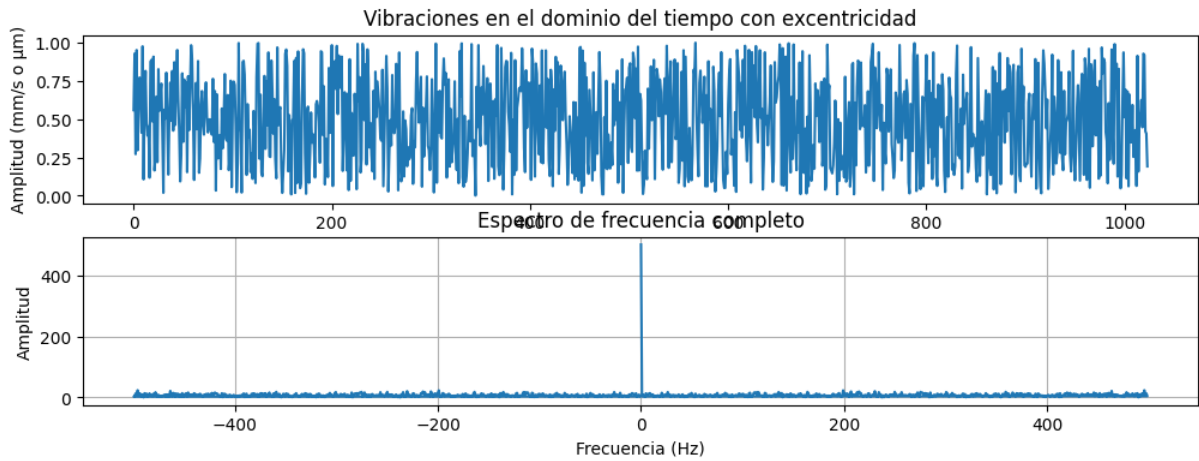


Figura 46. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de excentricidad

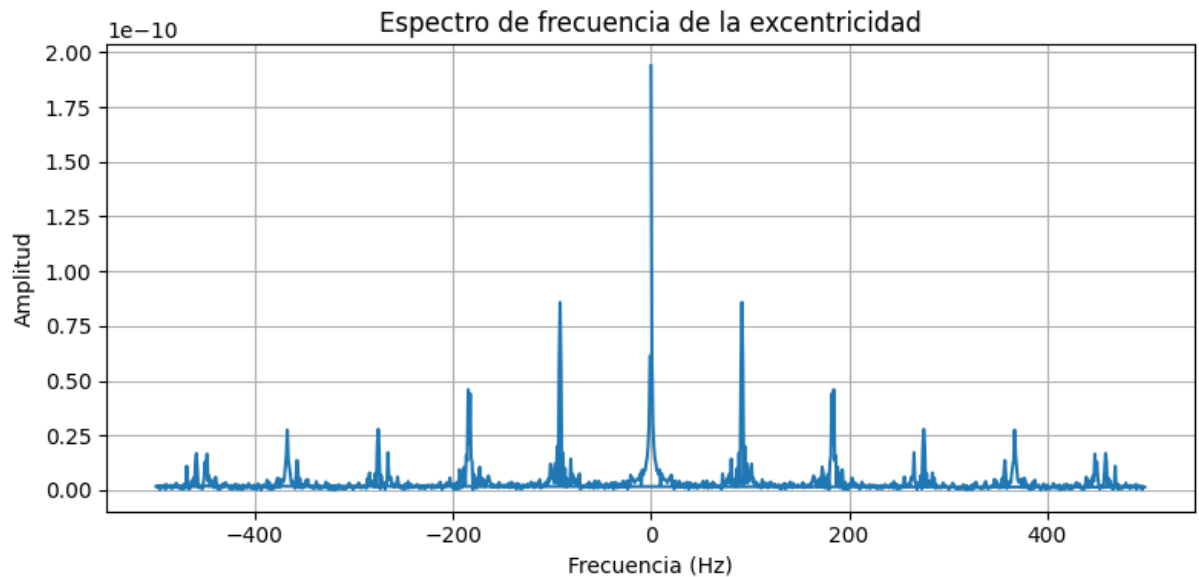


Figura 47. Espectro de frecuencia de la falla de excentricidad

En este caso se observa que existió un pico de frecuencia con una amplitud sumamente mayor a diferencia de los demás picos de frecuencia que son relativamente menores, sin embargo, a diferencia de la falla anterior, los picos de amplitud entre 0.75 y 1 son característicos de este tipo de falla por lo que se puede diferenciar visualmente de la falla anterior, sin embargo, se debe realizar la

comparación con la norma ISO 10816-3, algunos ejemplos podrían ser desviaciones respecto a fabricación que afectan directamente al motor.

Falla de holguras

En este caso para este tipo de perturbación se ajustaron los parámetros como amplitud de frecuencia de holgura siendo este valor de 0,1 de la misma manera se ajustó la frecuencia de esta, siendo de 20Hz acorde a la simulación y datos que se tenían.

Posteriormente se realizó el cálculo de la transformada de Fourier de la señal completa con la falla de holguras tal como se observa en la siguiente imagen.

```
# Añadir una señal de holguras a los datos de vibración simulados
# Puedes ajustar los parámetros de las holguras según tus necesidades
amplitud_holguras = 0.1
frecuencia_holguras = 20 # Frecuencia de las holguras en Hz
t = np.arange(len(vibraciones_tiempo_real))
holguras = amplitud_holguras * np.sin(2 * np.pi * frecuencia_holguras * t)
vibraciones_tiempo_real += holguras

# Calcula la transformada de Fourier de la señal completa
n = len(vibraciones_tiempo_real)
frecuencia_muestreo = 1000 # Frecuencia de muestreo en Hz (ajusta según tus datos)

frecuencias = np.fft.fftfreq(n, 1 / frecuencia_muestreo)
espectro_frecuencia = np.abs(np.fft.fft(vibraciones_tiempo_real))
```

Figura 48. Código de la falla de holguras y de la Transformada de Fourier

De la misma manera, se calculó el espectro de la falla de holguras en el dominio de la frecuencia aplicando la transformada de Fourier a la señal inicial que se encontraba en el dominio del tiempo para conocer el espectro vibratorio de la falla y realizar una posterior comparación con la norma ISO 10816-3.

```
# Calcula el espectro de holguras por separado
n_holguras = len(holguras)
frecuencias_holguras = np.fft.fftfreq(n_holguras, 1 / frecuencia_muestreo)
espectro_frecuencia_holguras = np.abs(np.fft.fft(holguras))
```

Figura 49. Código para hallar el espectro de la falla de holguras

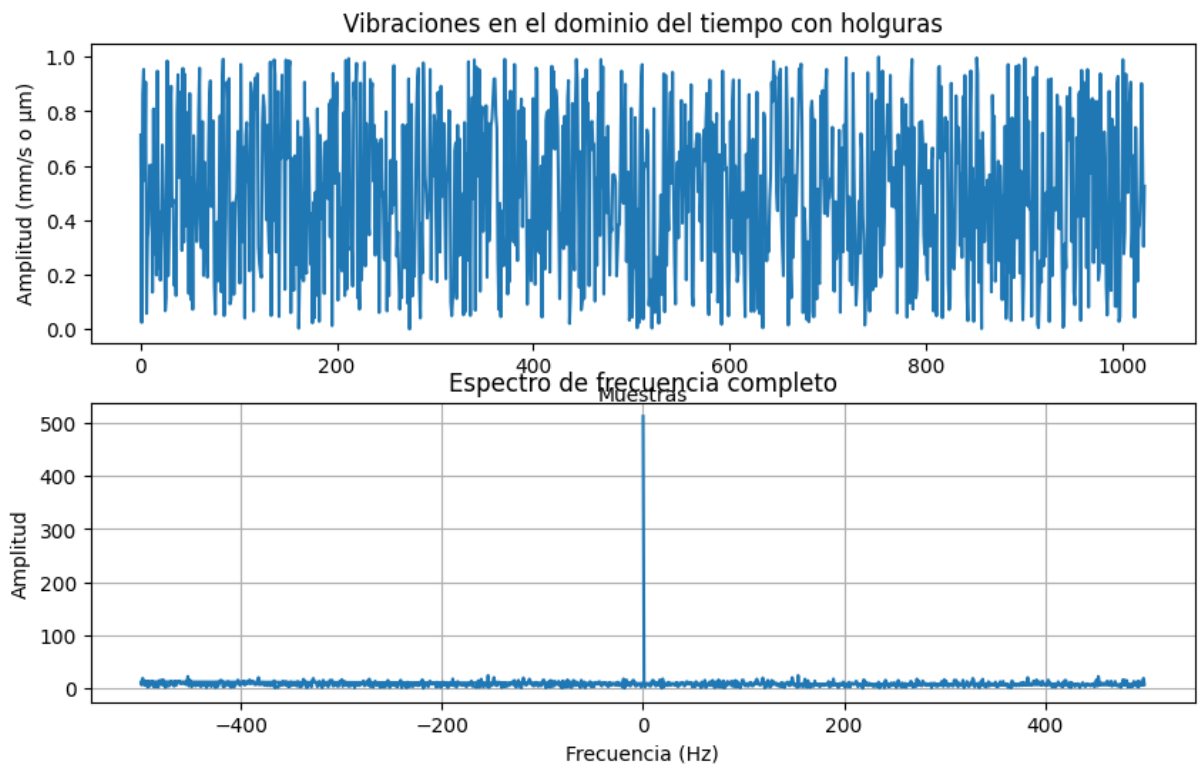


Figura 50. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de holguras

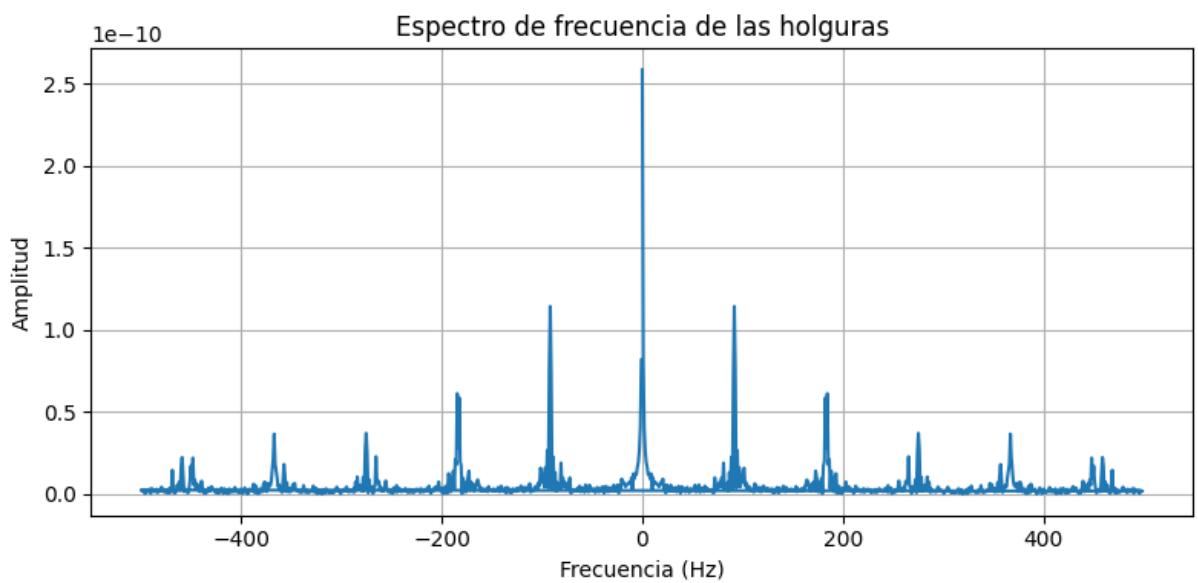


Figura 51. Espectro de frecuencia de la falla de excentricidad

En este tipo de falla, se observó que el espectro de frecuencia de la señal completa presentaba una amplitud sumamente mayor a diferencia de las otras fallas, siendo muy notorio respecto a los demás puntos de las gráficas, en cuanto al espectro de frecuencia de una falla de holguras, se observa el pico de mayor amplitud seguido por 2 picos de menor amplitud, siendo alrededor de 1.0 y 1.5 de amplitud y de 0.5 a 1.0

de amplitud, indicando que es netamente referente a un problema de holguras y estos pueden ser en soportes de amortiguadores, holguras en los pernos y conexiones propias del motor, en cojinetes produciendo vibraciones anormales, desgastes en engranajes, así como en uniones utilizadas para acoplamiento o netamente del motor los cuales pueden evaluarse siguiendo los criterios de la norma ISO 10816-3

Falla de rodamientos

Otro ejemplo de falla es cuando existe problemas en los rodamientos que sostienen el rotor permitiéndolo rotar lo más posible a la par del campo magnético rotativo del estator, en este caso, al añadir la falla en lenguaje de Python se colocó la amplitud de la falla de rodamientos estableciéndolo en 0.2 además de la frecuencia de la falla la cual se estableció en 120 Hz tal como se muestra en la siguiente imagen.

```
# Añadir una señal de problemas en rodamientos a los datos de vibración simulados
# Puedes ajustar los parámetros de los problemas en rodamientos
amplitud_problemas_rodamientos = 0.2
frecuencia_problemas_rodamientos = 120 # Frecuencia de problemas en rodamientos en Hz
t = np.arange(len(vibraciones_tiempo_real))
problemas_rodamientos = amplitud_problemas_rodamientos * np.sin(2 * np.pi * frecuencia_problemas_rodamientos * t)
vibraciones_tiempo_real += problemas_rodamientos
```

Figura 52. Código de la falla como señal de problemas en rodamientos

De la misma manera, se calculó la transformada de Fourier de la señal que se obtuvo para conocer el espectro vibratorio de la señal completa, así como el espectro de la falla de rodamiento con un muestreo de 1000 valores aleatorios.

```
# Calcula la transformada de Fourier de la señal completa
n = len(vibraciones_tiempo_real)
frecuencia_muestreo = 1000 # Frecuencia de muestreo en Hz
```

Figura 53. Código de la transformada de Fourier de la falla en rodamientos

```
# Calcula el espectro de problemas en rodamientos por separado
n_problemas_rodamientos = len(problemas_rodamientos)
frecuencias_problemas_rodamientos = np.fft.fftfreq(n_problemas_rodamientos, 1 / frecuencia_muestreo)
espectro_frecuencia_problemas_rodamientos = np.abs(np.fft.fft(problemas_rodamientos))
```

Figura 54. Código para hallar el espectro vibratorio de la falla en problemas en rodamientos

Una vez realizado el código para este tipo de falla, se procedió a graficarlos utilizando la función `plt.figure()` para obtener las gráficas de la señal en el dominio del tiempo y en el dominio de la frecuencia, de la misma manera, se graficó el espectro de

frecuencia de la falla en rodamientos la cual presenta un pico de amplitud sumamente elevado en comparación a los demás picos donde se observa que el pico de aproximadamente 175 Hz es relativamente más pequeño que el pico de los 420 Hz que posee una amplitud mayor a 1 y menor a 2.

Esta gráfica de espectro de frecuencia es similar a la gráfica de falla de desbalance puesto que un problema de rodamientos puede causar un desbalance dentro del motor, por ende, la gráfica es casi similar, siendo diferentes en las amplitudes de los picos de las frecuencias antes mencionadas.

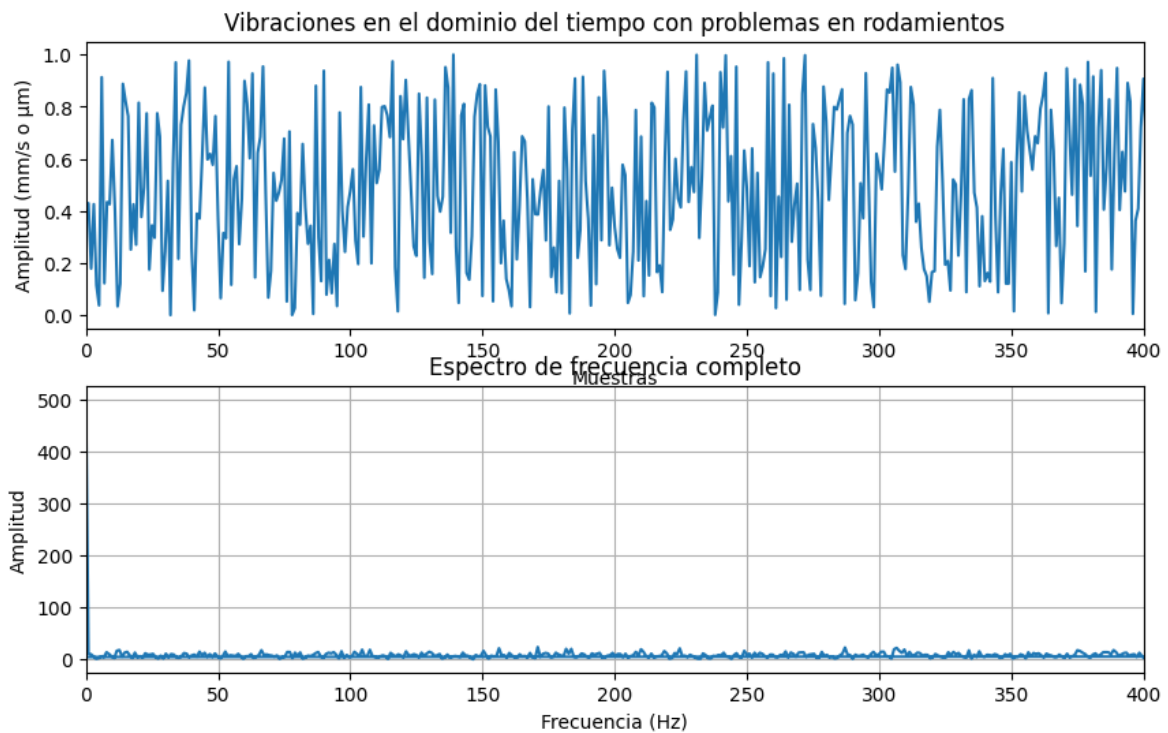


Figura 55. Vibraciones en el dominio del tiempo y el espectro de frecuencia de la falla de rodamientos

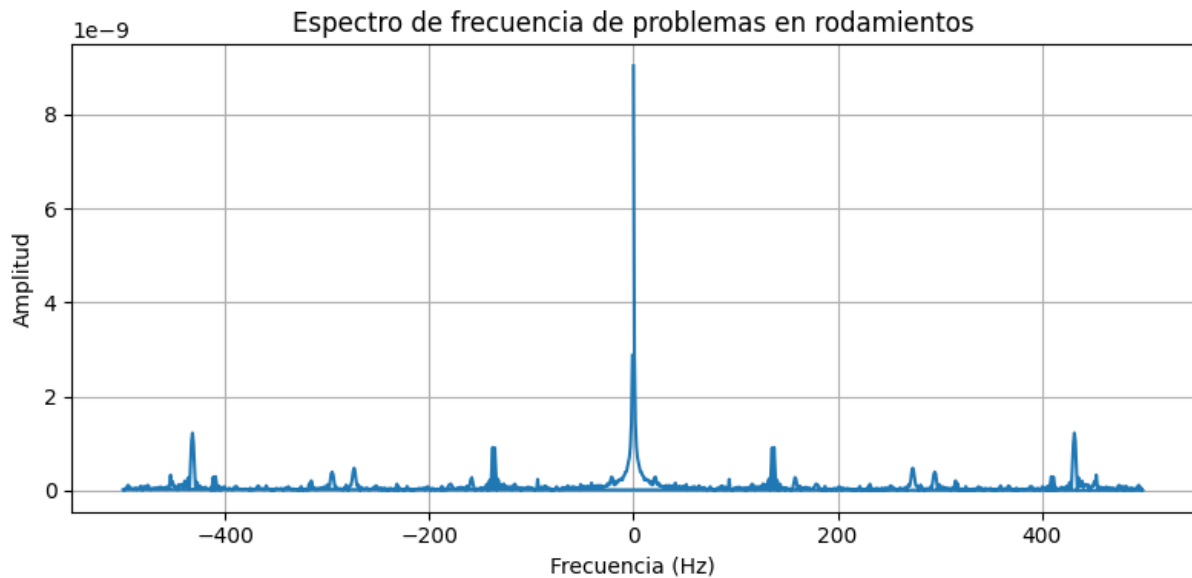


Figura 56. Espectro de frecuencia en problemas de rodamientos

Falla de problemas eléctricos

En este caso, un problema eléctrico como falla puede ser ocasionado por un cortocircuito en las bobinas, ya sea en el interior como en las bornas exteriores, en este caso se realizó la simulación actualizando el código específicamente para este problema, donde se añade la amplitud del disturbio del problema eléctrico en 0,2 además de la frecuencia con la que se está trabajando que es de 60 Hz tal como se observa en el siguiente código.

```
# Añadir una señal de problemas eléctricos a los datos de vibración simulados
# Puedes ajustar los parámetros de los problemas eléctricos según tus necesidades
amplitud_problemas_electricos = 0.2
frecuencia_alimentacion = 60 # Frecuencia de alimentación eléctrica en Hz
t = np.arange(len(vibraciones_tiempo_real))
problemas_electricos = amplitud_problemas_electricos * np.sin(2 * np.pi * frecuencia_alimentacion * t)
vibraciones_tiempo_real += problemas_electricos

# Calcula la transformada de Fourier de la señal completa
n = len(vibraciones_tiempo_real)
frecuencia_muestreo = 1000 # Frecuencia de muestreo en Hz (ajusta según tus datos)

frecuencias = np.fft.fftfreq(n, 1 / frecuencia_muestreo)
espectro_frecuencia = np.abs(np.fft.fft(vibraciones_tiempo_real))
```

Figura 57. Código de la falla como señal de problemas eléctricos y la transformada de Fourier de la señal completa


```

# Calcula el espectro de problemas eléctricos por separado
n_problemas_electricos = len(problemas_electricos)
frecuencias_problemas_electricos = np.fft.fftfreq(n_problemas_electricos, 1 / frecuencia_muestreo)
espectro_frecuencia_problemas_electricos = np.abs(fft(problemas_electricos))

```

Figura 58. Código del espectro de problemas eléctricos

Finalizado el código se realizó la simulación para observar el comportamiento de la señal vibratoria tanto en el dominio del tiempo como en el dominio de la frecuencia, esto se realizó utilizando la función `plt.figure()` tal como se observa la siguiente figura.

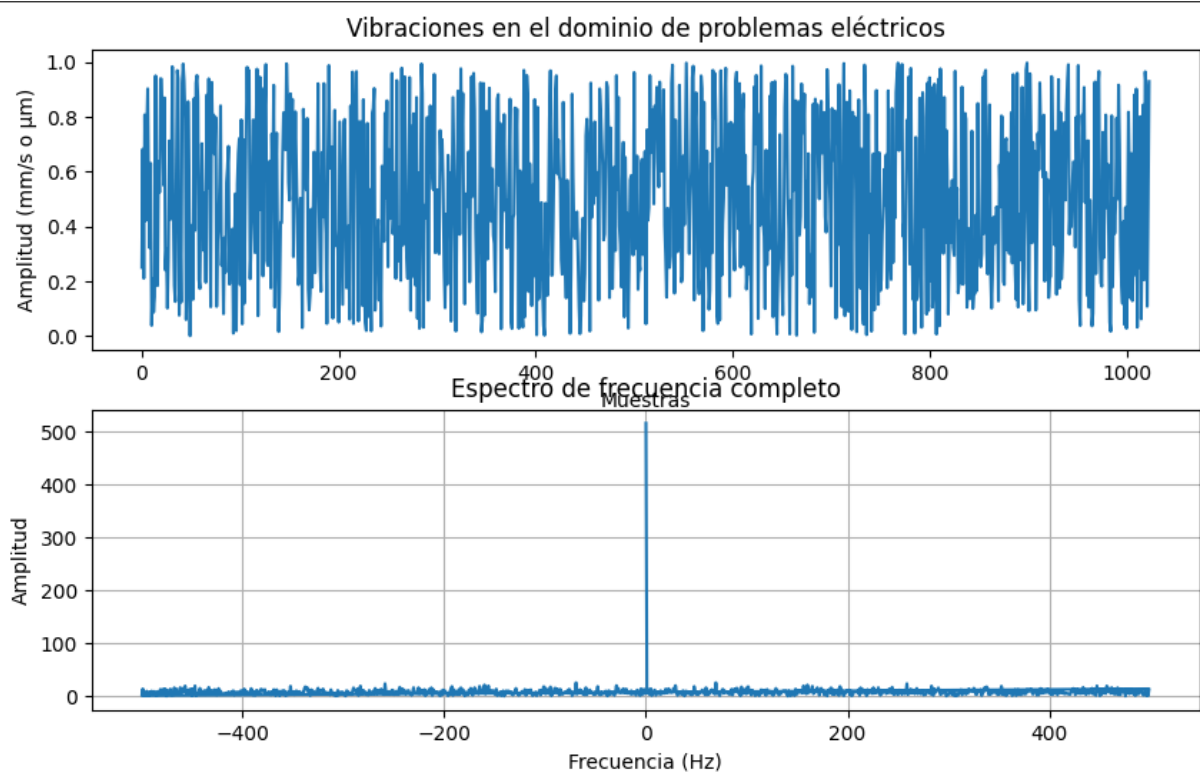


Figura 59. Vibraciones en el dominio del tiempo y el espectro de frecuencia de problemas eléctricos

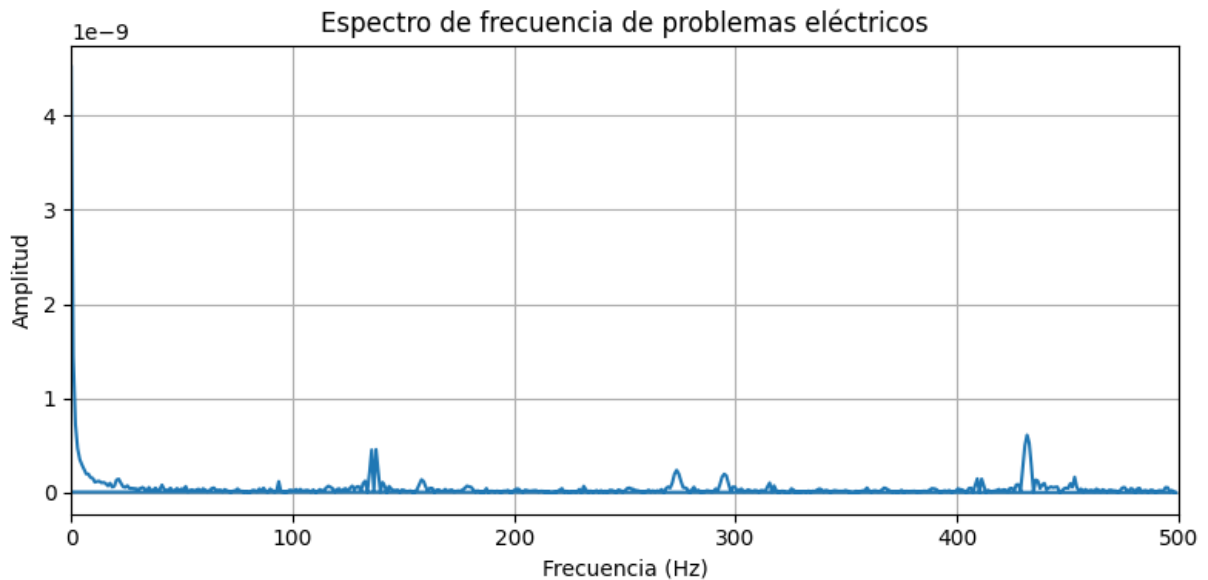


Figura 60. Espectro de frecuencia en problemas eléctricos

En este caso se observa que en el espectro vibratorio propio de la falla de problemas eléctricos existen 2 picos que presentan una amplitud mayor a la que normalmente está operando, este tipo de espectro es similar a las fallas de desbalance y al de excentricidad debido a que cuando ocurren problemas eléctricos se puede ver alterado la fuerza electromagnética que está dentro del motor y como consecuencia de aquello, el rotor puede sufrir alguna desalineación no perceptible a simple vista.

Una vez que se realizaron los códigos de las respectivas fallas de manera offline se realizó la validación en base a los criterios de validación del estándar ISO 10816-3 sin embargo existieron inconvenientes ya que al ser valores aleatorios estos no presentaban un comportamiento uniforme como si fuesen lecturas monitoreadas y medidas propiamente de vibraciones en motores por lo que se requirió de un sistema más robusto para una mejor lectura y toma de datos vibratorios así como un código donde se procesen de mejor manera los valores medidos, tal es la razón por la que se obtuvo el sensor de vibración WTVB01-485 para realizar mediciones de vibraciones cuando el motor esté en funcionamiento.

Este sensor contaba con una interfaz propia del equipo tal como se mencionó en el capítulo 2 por lo que los datos que se exportaron dieron como resultados valores en mm/s complementando con lo explicado en la parte de la metodología. Luego de haber realizado la conexión del sensor con la computadora habiendo acoplado el código en Python, se puso a prueba nuevamente el sensor, esta vez, en los motores del grupo

de bombeo 9H en La Toma – INTERAGUA específicamente a un motor de 1250 HP en operación normal donde se obtuvieron los valores solicitados en un tiempo determinado en un gráfico “plot” de Python como se logra observar en las siguientes ilustraciones.



Figura 61. Grupo de Bombeo 9H y placa del motor.

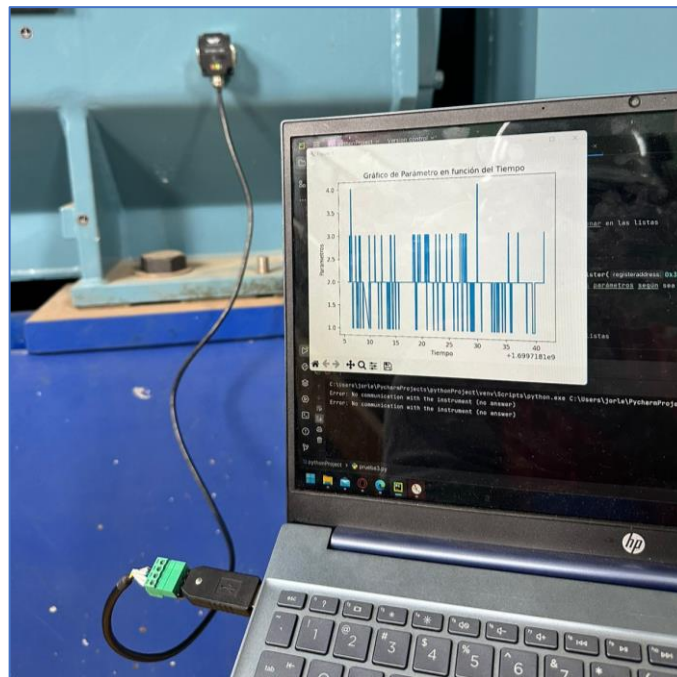


Figura 62. Resultados de vibración en el eje X del grupo de bombeo 9H en La Toma-INTERAGUA

Luego de realizar las pruebas se actualizó el código de Python para calcular la transformada de Fourier en el eje X cuya dirección hexadecimal es “0x3A”, esto para poder obtener los espectros de frecuencia de la máquina asíncrona y con esto comparar los resultados de vibración con los espectros característicos que tiene la

máquina con alguna falla en específico. Gracias a las herramientas de Python se escribió en las líneas del código las operaciones para determinar la transformada de Fourier.

Una vez que se actualizó el código con los datos triaxiales que podía leer el sensor, se realizó la última prueba donde previamente se cambió de ubicación al sensor de donde se encontraba, esta vez ubicándolo en la posición de los ejes X, Y invertidos obteniendo los resultados similares a las gráficas anteriores, pero en esa ocasión se visualizó una reacción más notoria en el: “Espectro de Frecuencia – Parámetro B” que corresponde a la gráfica 2 que toma la forma de las gráficas anteriores del “Espectro de Frecuencia – Parámetro A” como se muestra en la gráfica 4.

Sin embargo, para tener una mejor seguridad de nuestros resultados se pudo obtener, gracias al Ingeniero Eléctrico encargado del área de rebobinado de motores de Caterpillar en Quito, un sensor de vibración 469-1621 de la marca CAT valorado en aproximadamente quince mil dólares con un sensor monoaxial el cual se utiliza para realizar el análisis de vibraciones de los motores y generadores en el área de mantenimiento de máquinas.

Una vez se realizó la prueba, se visualizó que el sensor WTVB01-485 tenía resultados aproximados a las mediciones que tenía el sensor de la marca CAT estandarizada con la norma ISO 2372 por lo que se procedió a cambiar los umbrales de vibración según la norma en el código de Python y por consiguiente se realizó una interfaz utilizando la librería TTKINTER con lo que se pudo programar una HMI sencilla en la que el usuario pudiera iniciar la recopilación de los datos de vibración en mm/s de los tres ejes de la máquina y visualizar si existen anomalías mediante los umbrales de las clases que categorizan la norma ISO 2372 las cuales se dividen por el tipo de potencia de la máquina tal como se observa en la figura 5, mientras que la aplicación del sensor de la marca CAT junto al motor se observan en la siguiente figura.

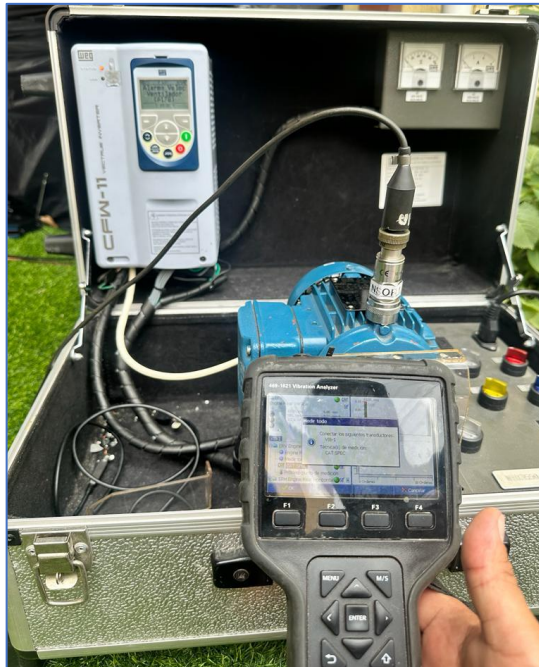


Figura 63. Pruebas de comparación del sensor WTVB01-485 y el sensor del analizador CAT

3.2 Pruebas en los motores usando el sensor de vibración y el monitor de vibraciones.

Previo al desarrollo de las respectivas pruebas, se necesitaba colocar los espectros vibratorios de las gráficas de falla por lo que basados en los criterios de la norma ISO 2372 se colocaron los siguientes espectros de frecuencia como se muestran en las siguientes imágenes.

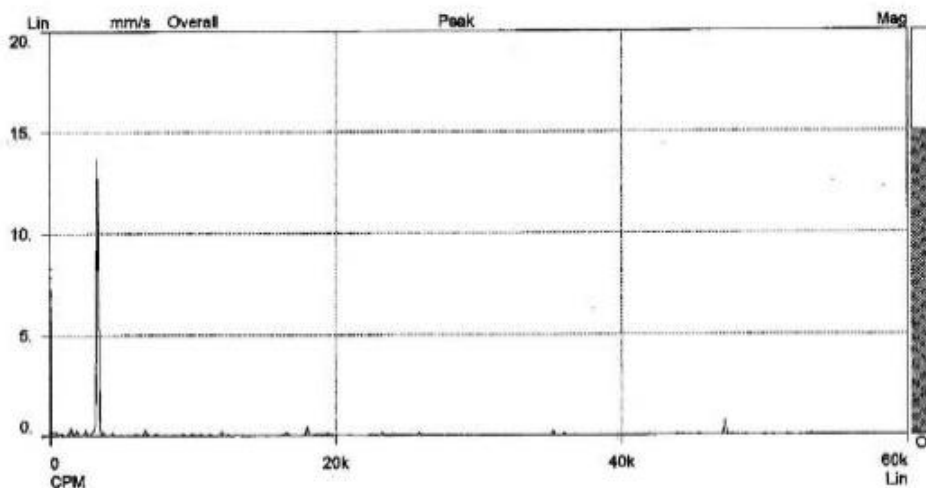


Figura 64. Espectro vibratorio de falla de desbalance

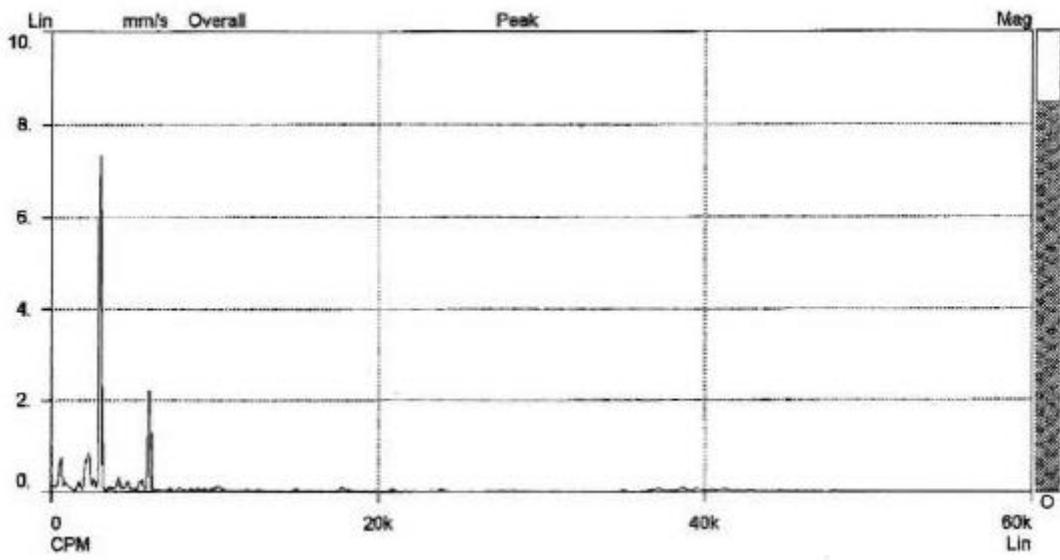


Figura 65. Espectro vibratorio de la falla de desalineación

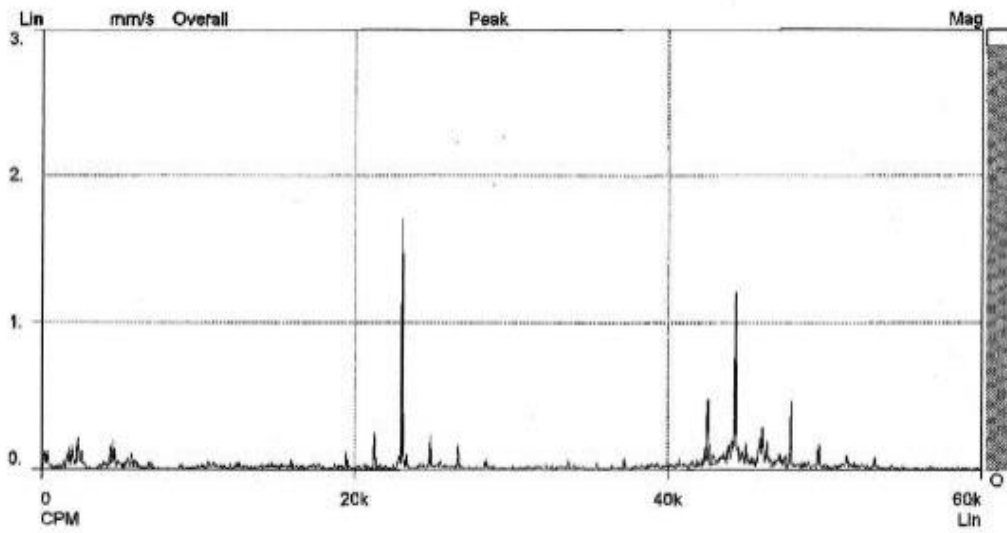


Figura 66. Espectro vibratorio de la falla de holguras

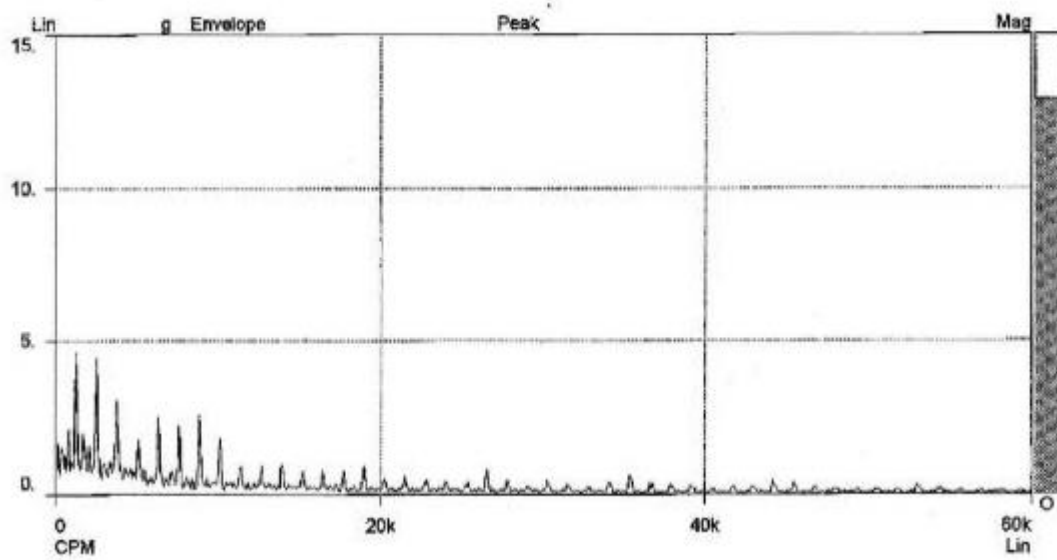


Figura 67. Espectro vibratorio de la falla de rodamientos

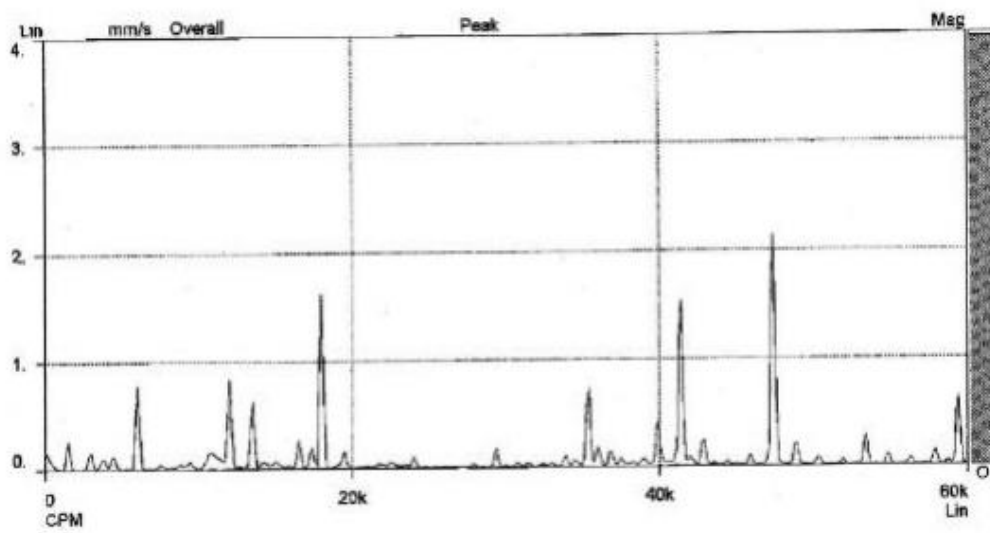


Figura 68. Espectro vibratorio de la falla de problemas eléctricos

Luego de tener el tablero con la pantalla HMI y el microprocesador que analiza los datos recopilados por el sensor de vibración se procede a realizar las pruebas de funcionamiento en la maleta de pruebas que contiene un motor, un variador y la configuración para que el motor llegue a 1800 rpm como dicta la norma ISO 2372 la cual es utilizada para conocer el estado vibratorio de la maquina asínrona trifásica. Se procede a ubicar el sensor en la caja de lateral de conexiones del motor como se visualiza en la siguiente figura.



Figura 69. Ubicación del sensor de vibración en el motor para las pruebas de funcionamiento.

El sensor se ubicó de tal manera que esté cerca de las partes giratorias de la máquina asínrona, en este caso no se lo puede ubicar directamente sobre el área de los rodamientos por la falta de espacio y esto podría ocasionar que el sensor detecte vibraciones que no corresponden al funcionamiento de la máquina si no, que sean por la vibración entre la pequeña área debajo de la placa del motor y el sensor de vibración. Se identifica los ejes donde fue ubicado el sensor como, por ejemplo: el eje “X” corresponde a la vibración de la maquina en horizontal, es decir, de izquierda a derecha; el eje “Y” corresponde a la vibración de la máquina en vertical, es decir, de

arriba hacia abajo del motor; el eje “Z” corresponde a la vibración de la máquina visto desde el plano superior, desde hacia adentro y fuera del motor.

Pruebas del monitor de vibración con el motor en operación normal sin alterar su estructura.

Teniendo el motor en operación normal y a 1800 rpm se procedió a realizar el monitoreo vibracional de la máquina iniciando la recopilación de los datos de vibración donde se podía ver en tiempo real el estado de la máquina asíncrona, los datos fueron recopilados durante un periodo de tiempo aproximadamente de 4 minutos donde se pudieron recolectar más de 400 muestras del estado vibracional de la máquina en los tres ejes.

Como resultado obtuvimos que el motor se encuentra dentro de los límites de vibración de la norma ISO 2372 en la clasificación de la Clase I, por ser un motor pequeño de hasta 15kW, donde la norma dice que se encuentra dentro de los valores entre 0.71 mm/s y 4.5 mm/s a 1800 rpm estando dentro de los rangos de satisfactorio como se puede visualizar en las gráficas de la siguiente figura.

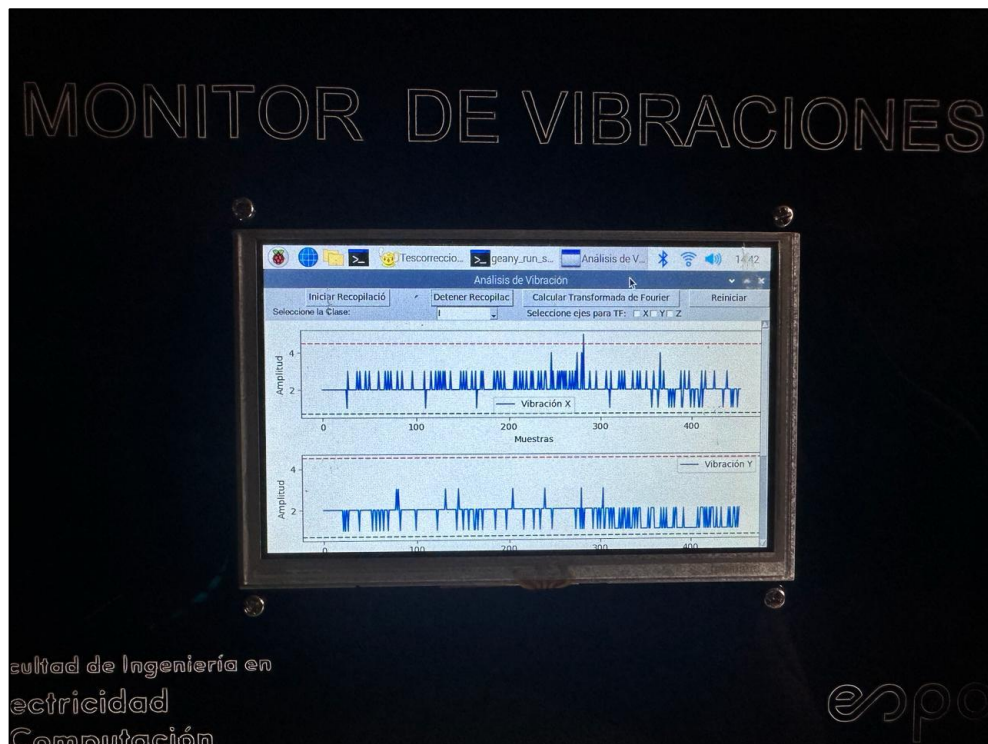
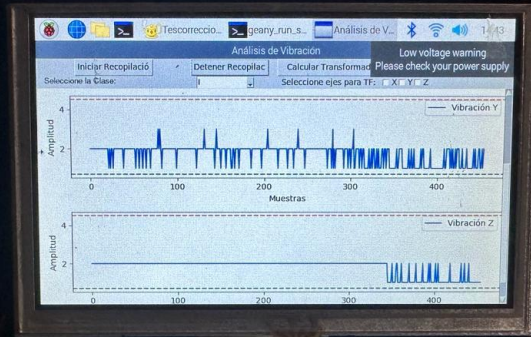


Figura 70. Estado Vibracional de la máquina Asíncrona en operación normal en el eje X

MONITOR DE VIBRACIONES



Facultad de Ingeniería en
Electricidad
y Computación

espol

Figura 71. Estado Vibracional de la máquina Asíncrona en operación normal en el eje "y", el eje "Z"

Velocidad (mm/s, rms)	Tipos de máquinas			
	Clase I	Clase II	Clase III	Clase IV
0,18 a 0,28	A			
0,28 a 0,45				
0,45 a 0,71	B			
0,71 a 1,12				
1,12 a 1,8	C			
1,8 a 2,8				
2,8 a 4,5	D			
4,5 a 7,1				
7,1 a 11,2	D			
11,2 a 18				
18 a 28	D			

Figura 2. 26. Tabla de severidad de vibraciones norma ISO 2372.

Figura 72. Rangos de la Norma ISO 2372

Luego de visualizar que los resultados obtenidos no muestran anomalías de vibración y no se encuentran fuera de los parámetros establecidos por la norma ISO2372 se procedió a calcular la Transformada de Fourier de los tres ejes, lo que se pudo visualizar que no habían espectros característicos de las señales de fallas de: Desbalance, Desequilibrio, Problemas Mecánicos o Problemas eléctricos lo que dio un resultado positivo en el análisis de vibración de la máquina, porque, la maleta didáctica que se prestó para dar las pruebas de funcionamiento del proyecto era relativamente nueva, lo que quiere decir que el motor no había sido utilizado para operar con cargas pesadas o algún funcionamiento similar por lo que se obtuvieron los resultados presentados a continuación de los ejes: “X”, “Y” y “Z” donde sus espectros de vibración en el dominio de la frecuencia no son nada similares a los espectros característicos de las principales fallas que una máquina asíncrona trifásica podría tener.

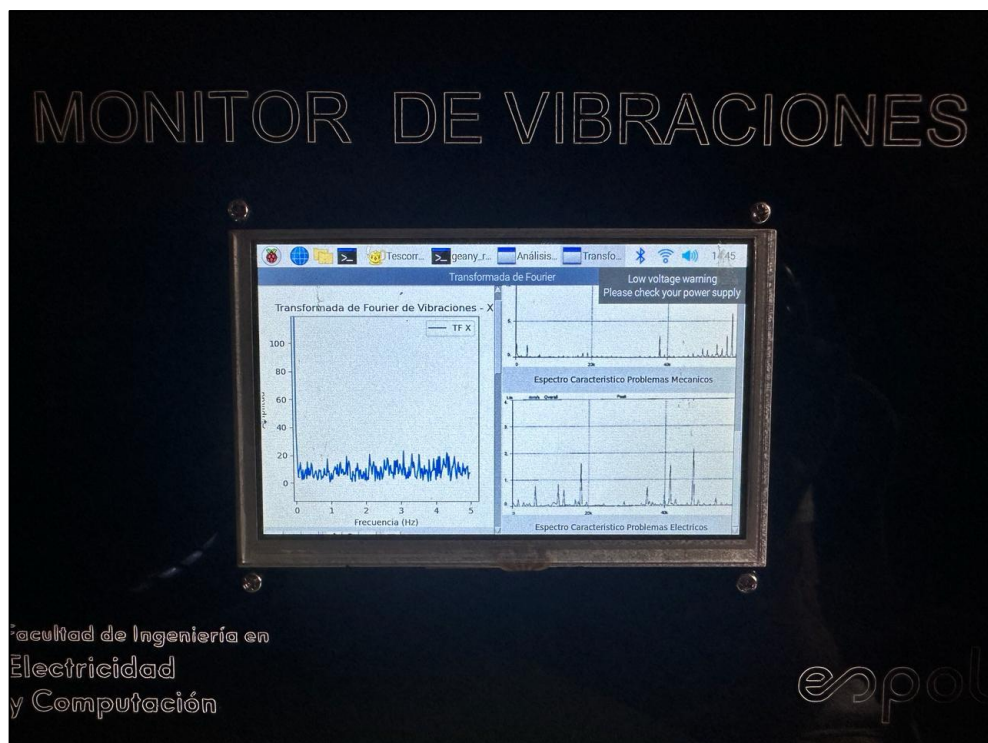
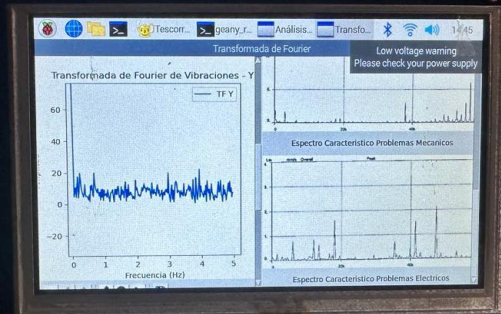


Figura 73. Espectro de Frecuencia del eje X

MONITOR DE VIBRACIONES

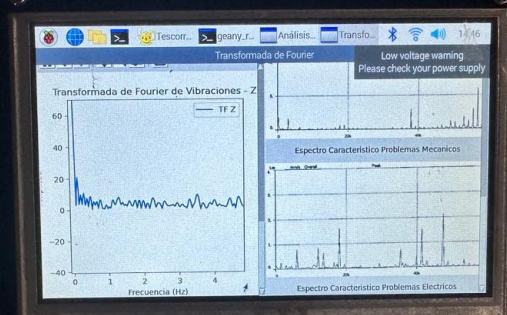


Facultad de Ingeniería en
Electricidad
y Computación

espol

Figura 74. Espectro de Frecuencia del Eje Y

MONITOR DE VIBRACIONES



Facultad de Ingeniería en
Electricidad
y Computación

espol

Figura 75. Espectro de Frecuencia del Eje Z

Pruebas del monitor de vibración con el motor en operación normal alterando su estructura.

Para la siguiente parte, se procedió a retirar y/o aflojar los cuatro pernos de la base donde el motor se situaba para realizar las pruebas de funcionamiento a 1800 rpm en operación normal, esto con el fin de saber qué tipo de resultados obtendríamos en el monitoreo vibracional cuando está ocurriendo este evento en tiempo real, por lo que adicionalmente se ubicó el motor de tal manera en la que podría tener un movimiento oscilatorio provocando que no se encuentre bien fijo y ocasionando que se tengan espectros diferentes que las primeras pruebas realizadas con la expectativa de pasar los umbrales de la norma ISO 2372 que indican cuando una máquina asíncrona se encuentra en correcto funcionamiento.

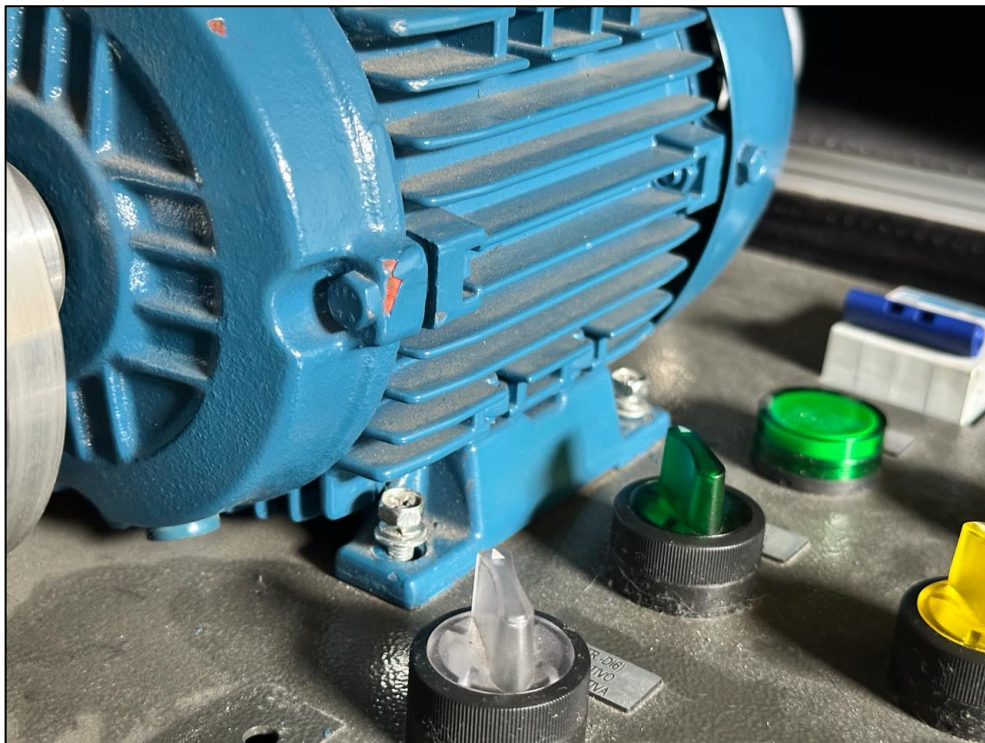


Figura 76. Pernos de la base del motor sueltos para las pruebas de funcionamiento

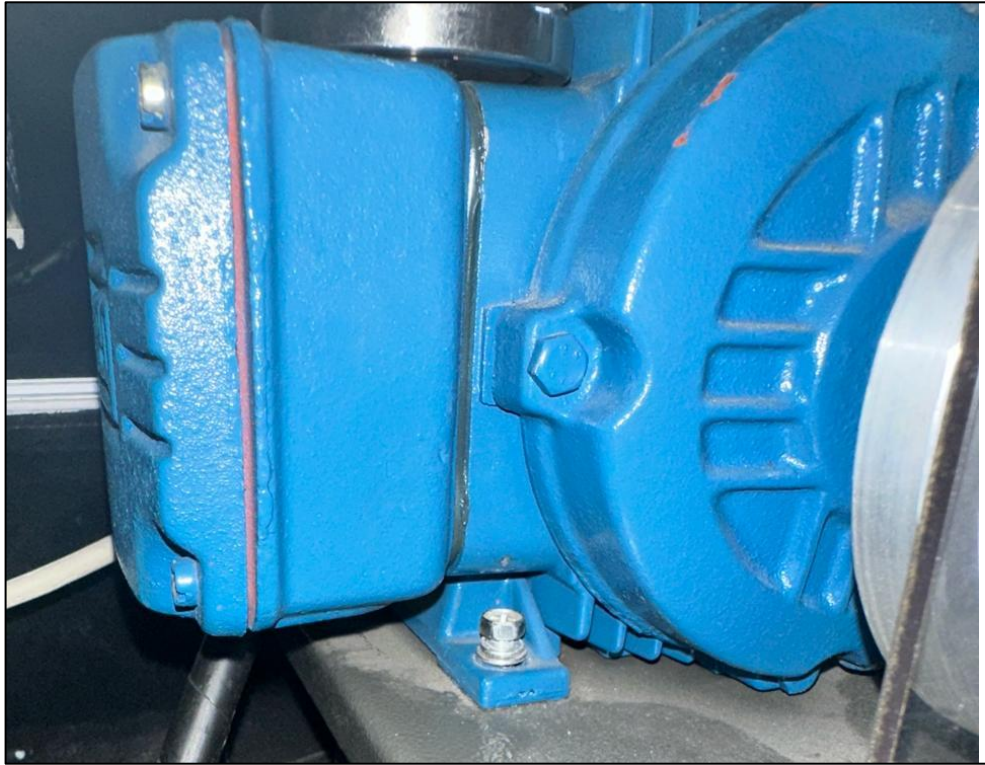


Figura 77. Pernos desajustados en la base del motor para pruebas de funcionamiento

Una vez que se ha realizado las respectivas pruebas utilizando el monitor se puede observar el comportamiento que está ocurriendo en el motor, como ejemplo se muestran en las siguientes imágenes, cuando el motor está operando y se le ha alterado su estructura, esto se proyecta en la pantalla HMI en tiempo real donde se muestra la gráfica en el dominio del tiempo sobre las vibraciones en los diferentes ejes, X, Y y Z como se observa en la **figura 78 y figura 79**, luego en la misma interfaz, de manera didáctica, se pueden realizar el cambio de gráfico hacia el dominio de la frecuencia donde aparece el gráfico comparativo de las vibraciones mediante la Transformada de Fourier y los espectros vibratorios de las fallas en los cuales basados en los criterios de la norma ISO 2372 se puede determinar qué tipo de falla está ocurriendo.

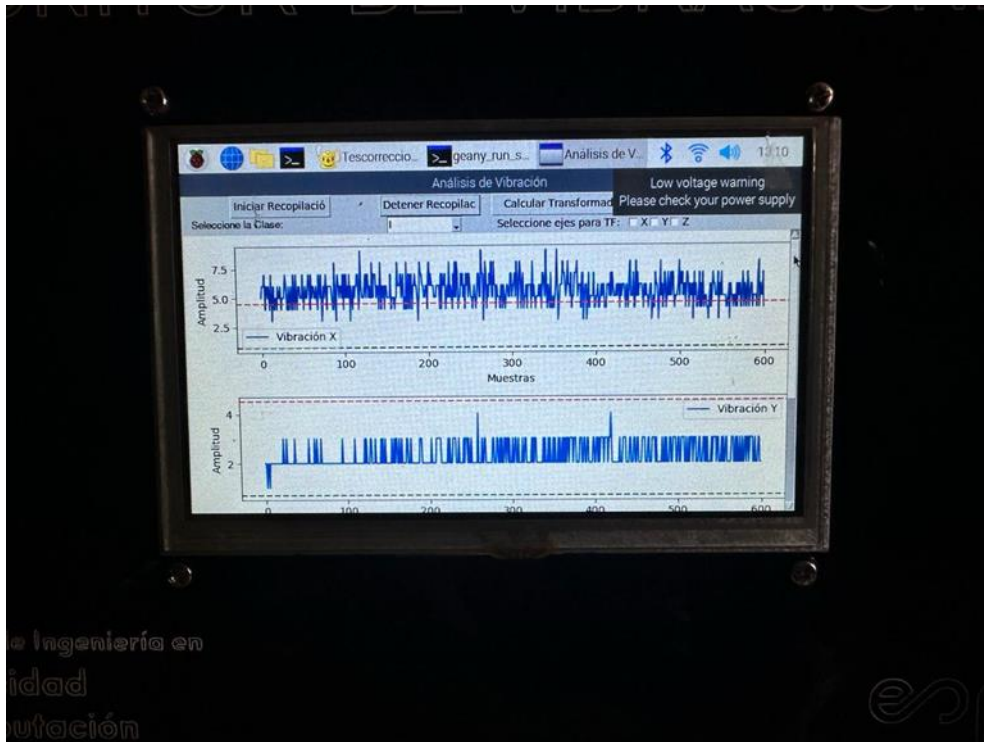


Figura 78. Estado vibracional en el dominio del tiempo del eje X cuando se altera la estructura del motor



Figura 79. Estado vibracional en el dominio del tiempo del eje Y y Z cuando se altera la estructura del motor

Se puede visualizar que, en comparación a los resultados obtenidos en las primeras pruebas realizadas, en esta ocasión el sensor de vibración captó señales vibratorias muy distorsionadas y que llegaron a pasar el umbral de la norma ISO 2372; las gráficas presentan que existe un fenómeno vibratorio anormal en el eje de las "X" donde está ubicado el sensor en el motor, los ejes "Y" y "Z" se encuentran dentro de los parámetros de la máquina pero aun así tienen cierto rango de amplitud mayor a los primeros resultados obtenidos, esto se puede deber a que el movimiento vibracional anormal que está sucediendo está afectando a otras áreas de la máquina por lo que se puede apreciar una forma diferente.

Al hacer clic en "Calcular Transformada de Fourier" se presenta en otra ventana los espectros de vibración en el dominio de la frecuencia lo que en amplitud se presenta como mm/s y el eje de las abscisas en valores de frecuencia; se puede notar que en la **Figura 80** se tiene el espectro de frecuencia del eje de las "X" muy diferente a las pruebas realizadas sin alterar la estructura de la máquina, lo que nos permite garantizar que existe un monitoreo del estado de la máquina efectivo y más aun que el resultado obtenido en el eje X se puede visualizar uno de los espectros característicos que puede tener una máquina asíncrona como lo es "Problemas mecánicos" que tiene que ver con anomalías estructurales de la máquina, como rodamientos o hasta la mala ubicación del motor en su base, lo que se asemeja mucho a lo realizado para los resultados obtenidos.

Se observa en las gráficas de las **Figura 81 y Figura 82** los espectros característicos de la máquina en los ejes "Y" y eje "Z" por lo que se tiene espectros diferentes a los obtenidos en los primeros resultados, con una mayor amplitud y una forma distinta, esto se puede deducir que su comportamiento se debe a que el problema en el eje de las "X" afectó de cierta manera a los otros elementos de la máquina pero que siempre se visualiza el eje que tiene mayor problema en su funcionamiento.

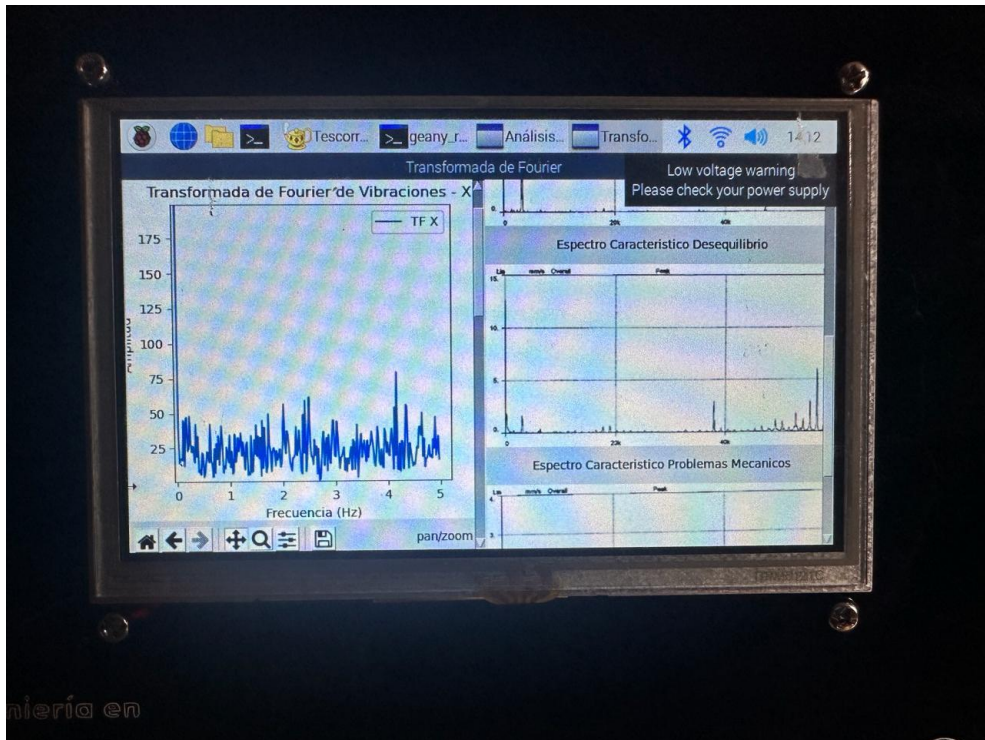


Figura 80. Espectro de frecuencia del eje X

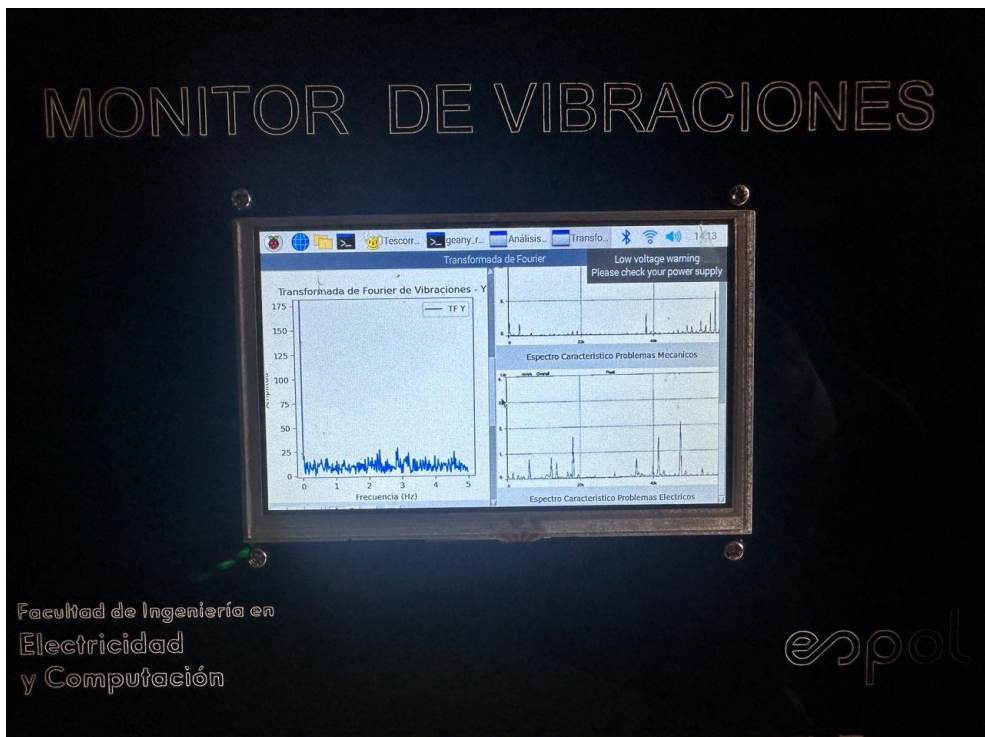


Figura 81. Espectro de frecuencia del eje Y

MONITOR DE VIBRACIONES

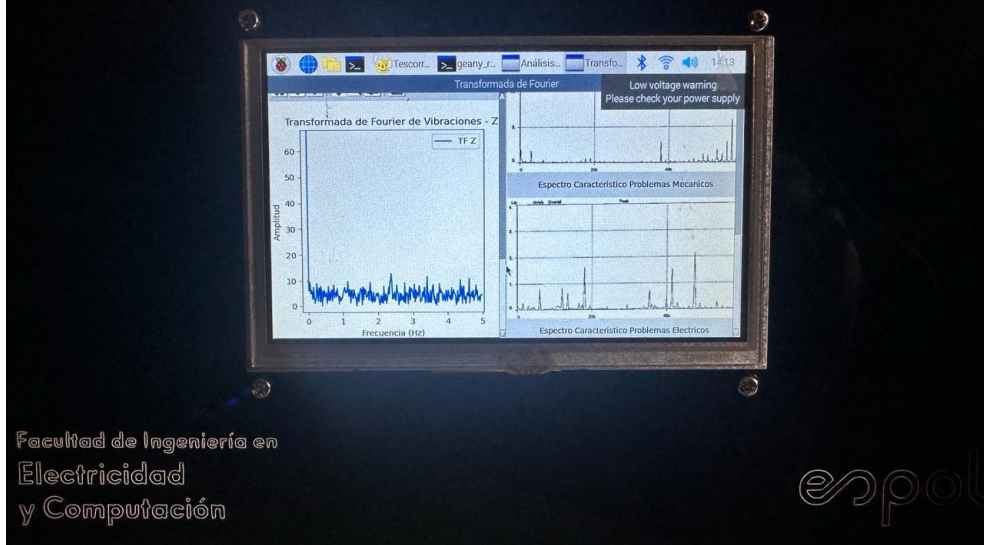


Figura 82. Espectro de frecuencia del eje X

CAPITULO 4

4. Conclusiones y recomendaciones

Conclusiones

- Se realizaron los códigos de lectura en tiempo real en base al procesamiento que se tenía inicialmente donde se logró actualizar el código para graficar la vibración en el dominio del tiempo y posteriormente se realizó la conversión en el dominio de la frecuencia logrando generar así los espectros vibratorios del comportamiento del motor, esto cuando el sensor capturaba datos en los ejes X, Y y Z obteniendo los espectros para cada eje respectivamente donde se sobresalta que la norma ISO 2372 resultó ser la más adecuada para satisfacer las necesidades de la industria en comparación a la norma ISO 10816-3.
- Se creó una interfaz gráfica por medio del lenguaje de programación Python y sus librerías como Tkinter y Matplotlib donde se logró integrar el sensor de vibración en el programa desarrollado y trabajar con la norma ISO 2372 la cual se consideró como apropiada para determinar el funcionamiento de la máquina en el proyecto mediante el análisis de vibraciones y los resultados de sus espectros de frecuencia de vibración.
- Se realizó la implantación del sensor de vibración junto a un Rapsberry pi que contiene el programa desarrollado integrando una pantalla HMI donde se observa el comportamiento del motor mediante vibraciones en tiempo real cuando se están realizando las pruebas permitiendo recopilar sus datos de funcionamiento.
- Se mostró la comparación en el monitor de vibraciones sobre el sistema recopilatorio mediante las pruebas realizadas donde se observaban las anomalías en base a la norma ISO 2372 siendo una innovación para el desarrollo en el mantenimiento preventivo de las máquinas asíncronas industriales contribuyendo significativamente en la reducción de costos en la industria por mantenimiento y monitoreo.

Recomendaciones

Se recomienda actualizar el código en base a las fallas que se determinen en los motores, en el presente proyecto se consideraron 5 tipos de fallas más comunes por lo que el código está hecho para obtener los espectros característicos de estos tipos de fallas y realizar la comparación con los valores medidos utilizando el sensor y el procesamiento de datos.

Se recomienda utilizar la tarjeta de adquisición de datos STM32MP135F en vez de un raspberry pi para darle un mejor uso al sensor y respectiva pantalla HMI, puesto que este tipo de tarjeta es de uso industrial permitiendo una mayor capacidad, estabilidad y mejor procesamiento de datos, mostrando resultados que pueden favorecer grandemente a la industria de motores asíncronos trifásicos.

Se recomienda colocar el sensor de tal manera que no se vea afectado por vibraciones mecánicas externas para tener así una lectura precisa de las vibraciones netamente del motor, esto sirve para una mejor adquisición de datos y evitar frecuencias armónicas que no estén relacionadas a la lectura real.

Anexos



Figura 83. Sensor es de vibración WTVB01-485



Figura 84. Pruebas realizadas con el sensor en una máquina de clase I y clase III según la norma ISO 2372

```

1  import minimalmodbus
2  import time
3
4  # Configuración del dispositivo MODBUS
5  device_address = 1 # Dirección del dispositivo MODBUS
6  baudrate = 9600 # Velocidad de comunicación en baudios
7  serial_port = 'COM3' # Puerto serie en el que está conectado tu dispositivo RS485
8  register_address = 0x3A # Dirección del registro que contiene el valor de vibración
9
10 # Crear un objeto de instrumento MODBUS
11 instrument = minimalmodbus.Instrument(serial_port, device_address)
12 instrument.serial.baudrate = baudrate
13
14 # Tiempo total para leer datos (en segundos)
15 total_time = 10
16
17 # Tiempo de espera entre lecturas (en segundos)
18 read_interval = 1
19
20 # Número total de lecturas
21 num_reads = int(total_time / read_interval)
22
23 try:
24     for _ in range(num_reads):
25         # Leer el valor de vibración del registro 0x3A
26         vibration_value = instrument.read_register(register_address, functioncode=3)
27
28         # Imprimir el valor leído
29         print(f'Valor de vibración: {vibration_value}')
30
31         # Esperar antes de la próxima lectura
32         time.sleep(read_interval)
33
34 except Exception as e:
35     print(f'Error: {e}')
36
37 finally:
38     # Cerrar la conexión al dispositivo
39     instrument.serial.close()

```

Figura 85. Primer código utilizado para comunicar computador con programa Python y sensor con convertidor RS485 a USB.



Figura 86. Resultados obtenidos con el código actualizado para presentar gráficas en tiempo real con Matplotlib

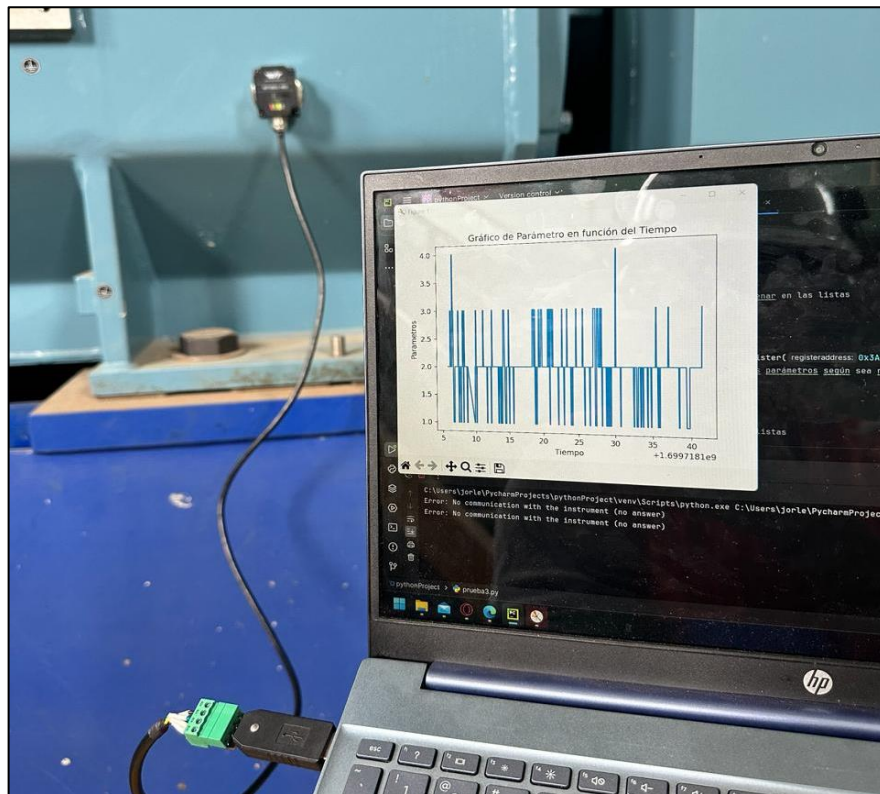


Figura 87. Pruebas realizadas en un sistema de bombeo en INTERAGUA - LA TOMA

ABB		ABB Oy Made in Honkitaie 13, 00990 Helsinki, Finland	
Type	AMI 450L8A BAFH	No	4688563
Year	2021	Phases	3~
Connection	Y	INS	F
ENCL	TEAAC	AMB	40°C
SF	1,15 TEMP RISE 125°C RTD		
Converter	Continuous, CONVERTER SUPPLY, ACS 1000		
Output	0 - 1250	HP	
Voltage	0 - 4160	V	
Current	0 - 168	A	
Power factor	0 - 0,81		
Frequency	0 - 60,5	Hz	
Speed	0 - 900,0		

Figura 88. Ficha Técnica del motor utilizado para monitoreo de vibraciones del sistema de bombeo.



Figura 89. Pruebas de vibración con una maleta de pruebas con motor de 0.5 HP y un variador de frecuencia

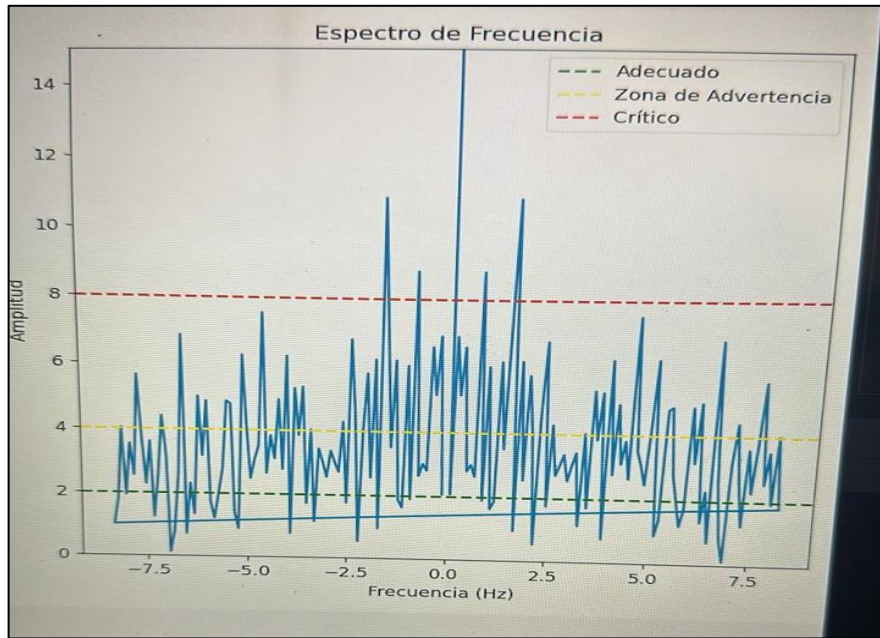


Figura 90. Resultados del monitoreo vibracional obtenido con la maleta de pruebas

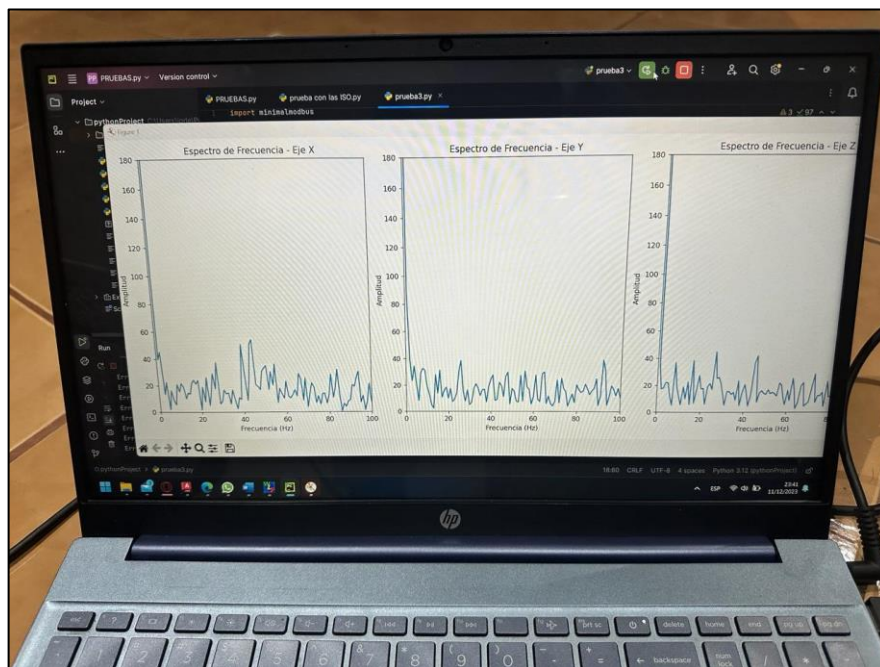


Figura 91. Espectros de frecuencia de vibración en tiempo real de la maleta de pruebas en los tres ejes



Figura 92. Resultados obtenidos con la actualización de la interfaz del programa con sistema de monitoreo y espectro vibracional en tiempo real



Figura 93. Pruebas realizadas con sensor profesional CAT

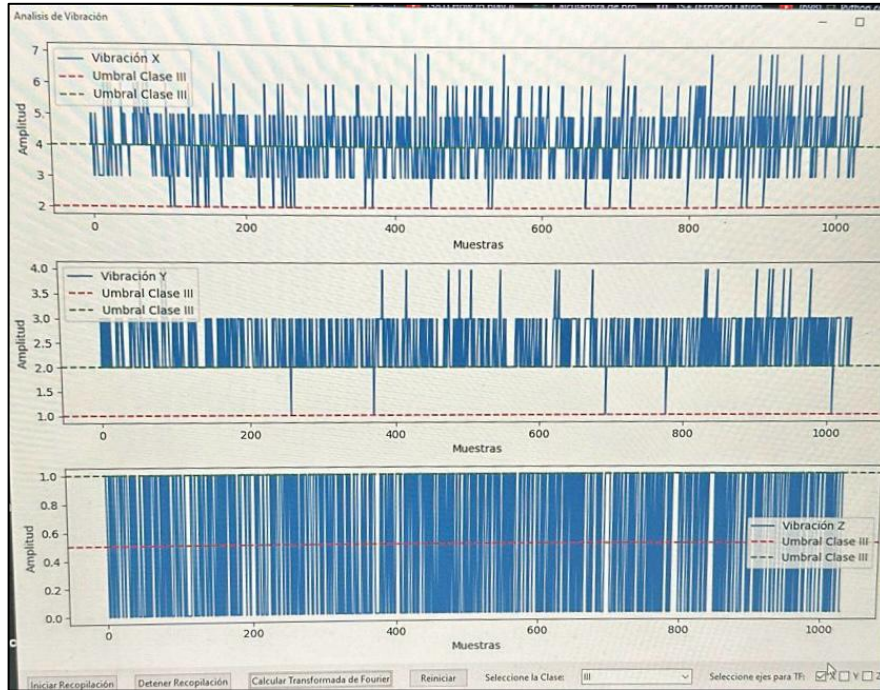


Figura 94. Nueva interfaz gráfica desarrollada para el programa con botones, checkbox y ventanas extra para presentar datos

```

vboxuser@mile: ~/Escritorio
[ 378.566385] usb 1-2: New USB device strings: Mfr=0, Product=2,
[ 378.566387] usb 1-2: Product: USB Serial
[ 378.572233] ch341 1-2:1.0: ch341-uart converter detected
[ 378.585429] usb 1-2: ch341-uart converter now attached to ttyU
vboxuser@mile:~/Escritorio$ cd Escritorio/
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py^C
vboxuser@mile:~/Escritorio$ ^C
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Error: No communication with the instrument (no answer)
vboxuser@mile:~/Escritorio$ python3 pruebalinux.py
Valor de vibración: 0
Valor de vibración: 0
Valor de vibración: 0
Valor de vibración: 3
Valor de vibración: 0
vboxuser@mile:~/Escritorio$

```

Figura 95. Pruebas de compatibilidad realizadas en un sistema operativo basado en LINUX con el sensor de vibración



Figura 96. Tarjeta STM32MP135F-DK



Figura 97. Raspberry Pi 3 como consola emuladora de juegos

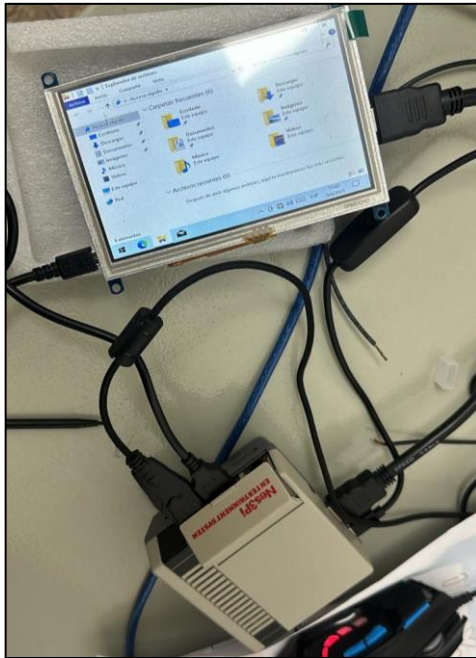


Figura 98. Integración pantalla HMI 5" con Raspberry Pi 3 y sistema operativo WIN10

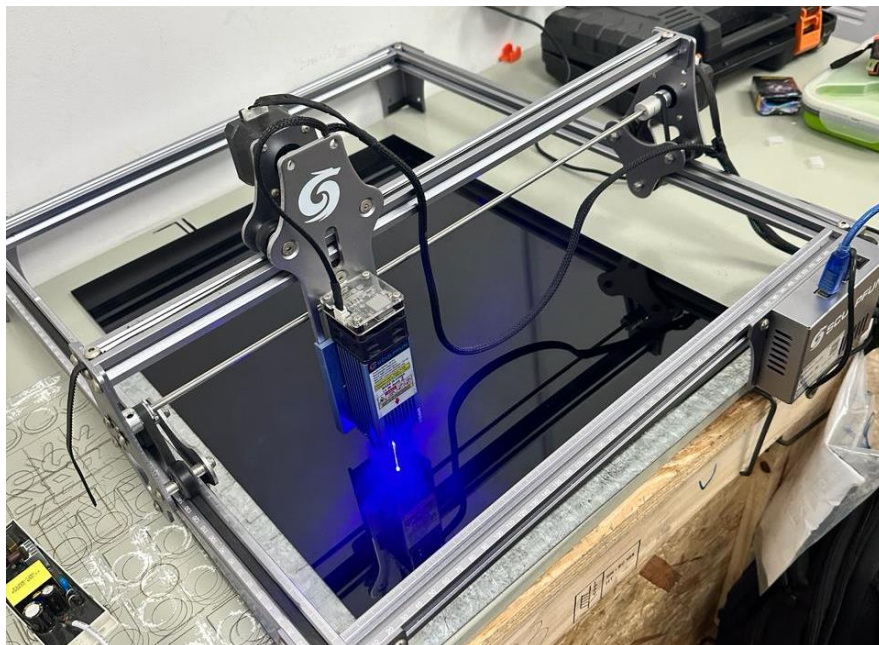


Figura 99. Creación de caja para tablero de monitoreo vibracional



Figura 100. Tablero monitor de vibraciones con sistema operativo Raspbian en un Raspberry Pi 3 en pruebas de funcionamiento con maleta de pruebas

BIBLIOGRAFIA

- [1] M. Moghadasian y S. M. Shakouhi, «Induction motor fault diagnosis using ANFIS based on vibration signal spectrum analysis,» de *3rd International Conference on Frontiers of Signal Processing*, Paris, France, 2017.
- [2] H. Chang-Hung, «Fractional Order PID Control for Reduction of Vibration and Noise on Induction Motor,» *IEEE Transactions on Magnetics*, vol. 55, n° 11, 2019.
- [3] A. Garcia-Perez, R. d. J. Romero-Troncoso, E. Cabal-Yepez y R. A. Osornio-Rios, «The Application of High-Resolution Spectral Analysis for Identifying Multiple Combined Faults in Induction Motors,» *IEEE Transactions on Industrial Electronics*, vol. 58, n° 5, pp. 2002-2010, 2011.
- [4] T. Plante, L. Stanley, A. Nejadpak y C. X. Yang, «Rotating machine fault detection using principal component analysis of vibration signal,» de *IEEE AUTOTESTCON*, Anaheim, 2016.
- [5] Z. Li, X. Jing y J. Yu, «Fault Detection Based on a Bio-Inspired Vibration Sensor System,» *IEEE Access*, vol. 6, pp. 10867-10877, 2018.
- [6] H. Kuemlee, T. Gross y J. Kolerus, «Machine vibrations and diagnostics the world of ISO,» de *Industry Applications Society 60th Annual Petroleum and Chemical Industry Conference*, Chicago, IL, USA, 2008.
- [7] F. Zhang, N. Tong y W. Fengxiang, «Analysis of vibration modes for large induction motor,» de *International Conference on Electrical Machines and Systems*, Nanjing, China, 2005.
- [8] D. Mori y T. Ishikawa, «Force and vibration analysis of induction motors,» *IEEE Transactions on Magnetics*, vol. 41, n° 5, pp. 1948-1951, 2005.
- [9] S. Onodera y K. Yamasawa, «Electromagnetic vibration analysis of a squirrel-cage induction motor,» *IEEE Transactions on Magnetics*, vol. 29, n° 6, pp. 2410-2412, 1993.
- [10] M. Riera-Guasp, J. A. Antonino-Daviu y G.-A. Capolino, «Advances in Electrical Machine, Power Electronic, and Drive Condition Monitoring and Fault Detection: State of the Art,» *IEEE Transactions on Industrial Electronics*, vol. 62, n° 3, pp. 1746-1759, 2015.
- [11] H.-U. Shin, S. K. Baek y K.-B. Lee, «Fault detection and fault-tolerant operation of a five-phase induction motor driving system,» de *IEEE 8th International Power Electronics and Motion Control Conference*, Hefei, China, 2016.

- [12] A. Naha, K. R. Thammayyabbabu, A. Kumar y A. Routray, «Mobile Application to Detect Induction Motor Faults,» *IEEE Embedded Systems Letters*, vol. 9, nº 4, pp. 117-120, 2017.
- [13] M. Kaviarasan, A. TamilSelvan y E. Venugopal, «Fault diagnosis of three phase squirrel cage induction motor due to bearing by using artificial intelligence,» de *International Conference on Emerging Trends in Engineering, Technology and Science*, Pudukkottai, India, 2016.
- [14] F. Bianchini-Batista, P. Monteiro Lamim-Filho, R. Pederiva y V. Diniz-Silva, «An Empirical Demodulation for Electrical Fault Detection in Induction Motors,» *IEEE Transactions on Instrumentation and Measurement*, vol. 65, nº 3, pp. 559-569, 2016.
- [15] N. Q. Hu, L. R. Xia, F. S. Gu y G. J. Qin, «A novel transform demodulation algorithm for motor incipient fault detection,» *IEEE Transaction on Instrumentation and Measurement*, vol. 20, nº 4, pp. 480-487, 2011.
- [16] S. A. Bednarz y M. Dybkowski, «On-Line Detection of the Rotor Faults in the Induction Motor Drive Using Parameter Estimator,» de *International Symposium on Electrical Machines*, Andrychow, Poland, 2018.
- [17] M. Tsytkin, «Induction motor condition monitoring: Vibration analysis technique - diagnosis of electromagnetic anomalies,» *IEEE AUTOTESTCON*, pp. 1-7, 2017.
- [18] M. Tsytkin, «The Origin of the Electromagnetic Vibration of Induction Motors Operating in Modern Industry: Practical Experience - Analysis and Diagnostics,» *IEEE Transactions on Industry Applications*, vol. 53, nº 2, pp. 1669-1676, 2017.
- [19] D. Bhaumik, A. Satta y G. S. Punekar, «Vibration Signal Analysis of Induction Motro Bearing Faults: Some Aspects,» de *International Conference on Smart Systems for applications in Electrical Sciences*, Tumakuru, India, 2023.
- [20] M. R. Barusu y M. Deivasigamani, «Non-Invasive Vibration Measurement for Diagnosis of Bearing Faults in 3-Phase Squirrel Cage Induction Motor Usign Microwave Sensor,» *IEEE Sensors Journal*, vol. 21, nº 2, pp. 1026-1039, 2021.
- [21] J. J. Saucedo-Dorantes, M. Delgado-Prieto, J. A. Ortega-Redondo, R. A. Osornio-Rios y R. D. J. Romero-Troncoso, «Multiple-fault detection methodology based on vibration and current analysis applied to bearings in induction motors and gearboxes on the kinematic chain,» *Shock Vibration*, vol. 2019, pp. 1-13, 2016.
- [22] O. V. Thorsen y M. Dalva, «Failure Identification and Analysis for High-Voltage Induction Motors in the Petrochemical Industry,» *IEEE Transactions on Industry Applications*, vol. 35, nº 4, pp. 810-818, 1999.

- [23] M. G. a. L. Motors, «Tavner, P. J.; Gaydon, B. G.; Ward D. M.,» *Proceedings of the Institution of Electrical Engineerings*, vol. 133, nº 3, pp. 169-180, 1986.
- [24] A. Bellini, C. Concari, G. Franceschini, C. Tassoni y A. Toscani, «Vibrations, currents and stray flux signals to asses induction motors rotor conditions,» de *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, Paris, France, 2006.
- [25] A. Sadda y G. S. Punekar, «Vibration Data and Analysis of A Large rating Induction Motor with Bent Shaft: Some Aspects,» de *International Conference on Power Instrumentation, Control and Computing*, Thrissur, India, 2020.