



Diseño e implementación de un sistema descentralizado remoto, basado en tecnologías IoT y de código abierto, para la adquisición de variables eléctricas y control de un motor de inducción trifásico.

Samir Correa Esquivia

Tesis de maestría presentada para optar al título de Magíster en Ingeniería

Directora

Melisa de Jesús Barrera Durando, Doctor (PhD)

Codirector

Johnatan Mauricio Rodríguez Serna, Doctor (PhD)

Universidad de Antioquia

Facultad de Ingeniería

Maestría en Ingeniería

Medellín, Antioquia, Colombia

2023

Cita	Samir Correa E [1]
Referencia Estilo IEEE (2020)	[1] Samir Correa E, "Diseño e implementación de un sistema descentralizado remoto, basado en tecnologías IoT y de código abierto, para la adquisición de variables eléctricas y control de un motor de inducción trifásico", Tesis de maestría, Maestría en Ingeniería, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2024.



Maestría en Ingeniería,
 Grupo de Investigación Manejo Eficiente de la Energía (GIMEL).



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Agradecimientos

Agradezco en primer lugar a Dios, quien me ha dado la fuerza y perseverancia para enfrentar las pruebas que cada día la vida nos plantea.

De igual forma agradezco a mi madre Nery Isabel Esquivia Hernández, que siempre ha estado para darme los mejores consejos y apoyo incondicional, a mi padre José Miguel Correa López, quien hoy ya no está presente con nosotros, y quien siempre me alentó y confió en mis capacidades de superación. A ellos agradezco lo que soy hoy en día.

Agradezco grandemente a los profesores Nelson Londoño Ospina, Andrés Sánchez Prisco, Johnatan Mauricio Rodríguez Serna y Melisa de Jesús Barrera Durango, quienes me guiaron y asesoraron con su conocimiento en el transcurso de este sueño que se materializa llamado maestría.

Contenido

Resumen	6
Abstract	7
Objetivos	8
Objetivo general	8
Objetivos específicos	8
Introducción	9
Capítulo 1: Estado del arte	10
Capítulo 2: Marco teórico	17
2.1 Internet de las Cosas (IoT)	18
2.2 MQTT	19
2.3 Node-red	20
2.4 MariaDB	20
2.5 Interfaz de usuario	21
2.6 Sistemas embebidos	21
2.7 Red inalámbrica	23
2.8 Buses de campo	24
2.9 Motores Eléctricos	25
Capítulo 3: Descripción general del sistema propuesto	29
3.1 Implementación del sistema propuesto	30
3.1.1 Tarjeta de adquisición de datos de temperatura y velocidad	31
3.1.2 Etapa de polarización	32
3.1.3 Etapa de conexión de sensores de velocidad y temperatura	32
3.1.4 Etapa de comunicación	33
3.1.5 Etapa de notificación de estados	34
3.1.6 Etapa de control	35
3.1.7 Tarjeta de circuito impreso	35
3.1.8 Descripción de <i>firmware</i> de las tarjetas	37
3.1.9 Medición de variables eléctricas	38
3.1.10 Etapa de control del motor de inducción	39
Capítulo 4: Servidor local	40
4.1 Descripción del software	41

4.1.1 Descripción de <i>firmware</i> servidor local.....	41
4.1.2 Almacenamiento en la nube.....	43
4.1.3 Seguridad SSL Node-red	47
4.1.3 Motor de base de datos	50
4.1.4 Interfaz de usuario	51
Capítulo 5: Pruebas y resultados	55
5.1 Calibración de la tarjeta de adquisición de temperatura	55
5.2 Calibración de la tarjeta de adquisición de velocidad.....	56
5.3 Descripción de la prueba	57
5.4. Análisis y resultados	61
6. Conclusiones.....	66
7. Trabajos futuros	68
8.Anexos	69
8.1 Código de programación para la tarjeta de temperatura.....	69
8.2 Código de programación para la tarjeta de velocidad.....	73
8.3 Configuraciones de instalación general del servidor en la nube	76
9.Referencias	85

Resumen

En el presente trabajo se presenta el desarrollo de un sistema descentralizado basado en tecnologías Internet de las Cosas (IoT- Por sus siglas en inglés) para la adquisición y control de variables eléctricas en un motor de inducción trifásico de forma remota e inalámbrica.

El sistema se implementó haciendo uso de software de código abierto, y la elección de dispositivos y componentes se realizó teniendo en cuenta criterios tales como: disponibilidad en el mercado local, costo, dispositivos IoT, compatibilidad, fiabilidad, flexibilidad, entre otros. El servidor IoT ofrece un servicio en la nube que tiene como función principal el monitoreo y control de variables de forma remota, sin embargo, entre otras funcionalidades podemos encontrar: visualización de datos en tiempo real (*dashboard*), generación de alarmas (notificación vía mensaje de texto), entre otras.

El sistema ha sido probado en el laboratorio de Máquinas Eléctricas de la Universidad de Antioquia, bajo condiciones similares a las encontradas en aplicaciones industriales, dentro de las cuales se realizaron pruebas simulando fallas en los motores de rotor devanado.

Esta implementación permite el control remoto inalámbrico del motor, donde es posible enviar la orden de marcha y parada del motor, variación de velocidad en su marcha e inversión del sentido de giro y además brinda la posibilidad de generar alarmas cuando las variables medidas presenten un comportamiento por fuera de lo establecido. El sistema emplea una red inalámbrica, que permite almacenar la información monitoreada en una base de datos en la nube, los cuales pueden ser consultadas por el usuario ingresando a dicha base de datos en el servidor o realizar la visualización de la información que se está mostrando en tiempo real por medio de la interfaz *web*, lo que le permitirá implementar tareas de mantenimiento, elaborar historiales de uso y seguimiento a condiciones de fallas que se presenten durante el funcionamiento del motor.

El sistema diseñado e implementado podrá aplicarse a máquinas eléctricas de funcionamiento similar y podrá usarse en entornos industriales y académicos, teniendo en cuenta que se implementó un sistema (prototipo) cuya madurez tecnológica puede clasificarse en una categoría mayor o igual a la TRL (Nivel de Transferencia Tecnológica) 4.

Palabras claves: Motor de inducción, sistemas embebidos, red inalámbrica, IoT, monitoreo, código abierto, sensores, actuadores, servidor, *dashboard*.

Abstract

This work presents the development of a decentralized system based on Internet of Things (IoT) technologies for the acquisition and control of electrical variables in a three-phase induction motor remotely and wirelessly. The system was implemented using open-source software, and the choice of devices and components was made considering criteria such as: availability in the local market, cost, IoT devices, compatibility, reliability, flexibility, among others. The IoT server is a cloud service that has as its main function the monitoring and control of variables remotely, however, among other functionalities we can find Data visualization in real time (Dashboard) and alarm generation (notification via text message). The system has been tested in the laboratory of electrical machines of the University of Antioquia, under conditions like those found in industrial applications, in which tests will be carried out simulating failure conditions in the rotor windings of the motor under study. This implementation allows the wireless remote control of the motor, where it is possible to implement the order to start and stop the motor, speed variation and inversion of the rotation direction and provides the possibility of generating alarms when measured variables exhibit abnormal behavior. The system has a wireless network, which allows storing the monitored information in a database in the cloud, which can be consulted by the user by accessing the database on the server or viewing the information being displayed in real time through the web interface, which it will allow the user to implement maintenance tasks, develop usage reports and monitoring of fault conditions that occur during motor operation. The designed and implemented system could be applied to electrical machines of similar operation and could be used in industrial and academic environments, considering that a system (prototype) was implemented that could fit in a category greater than or equal to Technology Transfer Level (TRL) 4.

Keywords: Induction motor, embedded systems, wireless network, IoT, monitoring, open source code, sensors, actuators, server, dashboard.

Objetivos

Objetivo general

Implementar un sistema descentralizado remoto, basado en tecnologías IoT y de código abierto para la adquisición y control de variables eléctricas en un motor de inducción trifásico.

Objetivos específicos

- ✓ Determinar los requerimientos del sistema descentralizado propuesto y las componentes de hardware y software basados en código abierto necesarios para su implementación.
- ✓ Diseñar e implementar las componentes de hardware y realizar la programación basada en código abierto de los dispositivos esclavos y el controlador maestro.
- ✓ Implementar un sistema de comunicación inalámbrica para el control y monitoreo de variables desde un servidor IoT en una plataforma de código abierto.
- ✓ Realizar la validación del sistema implementando los ajustes necesarios y publicar resultados obtenidos.

Introducción

Actualmente la automatización y control son líneas de profundización en Ingeniería Eléctrica muy importantes, que han tomado fuerza con el acelerado avance tecnológico [1]. Un ejemplo de ello es el Internet de las Cosas (IoT), que ha evolucionado de manera acelerada, dado el gran avance de la Internet y la optimización de procesos para control, monitoreo y supervisión de sistemas remotos. El mundo está cambiando de forma acelerada, anteriormente existía la conexión entre objetos que solamente se comunicaban para reportar informes de sus sensores, pero con el paso del tiempo, estos objetos tienen la posibilidad de comunicarse entre ellos, permiten el intercambio de información y la toma de acciones dependiendo de la información recibida [2].

Los motores eléctricos de inducción son de gran uso en los diferentes procesos implementados en las industrias, no solo por la confiabilidad, eficiencia y capacidad de trabajo en ambientes pesados [3], sino también por las facilidades en su control obtenidas con los distintos avances tecnológicos de la electrónica de potencia [4].

Es importante realizar el monitoreo de los parámetros y variables eléctricas de estos equipos; como lo son las tensiones y corrientes de alimentación, velocidad angular del rotor y la temperatura de operación, los cuales son fundamentales en los procesos de producción, para garantizar que las condiciones de operación sean seguras, confiables y eficientes. Existen dispositivos comerciales que permiten el monitoreo de las variables eléctricas de motores eléctricos de inducción en tiempo real, dentro de los cuales se pueden mencionar los analizadores de redes Fluke 438 II [5], Metrel MI 2892 [6], e-MCM - Artesis [7], Hioki 3198-01 [8], entre otros. Estos equipos, si bien permiten medir en tiempo real las condiciones operativas de los motores, no siempre permiten realizar control sobre el motor de manera sencilla y directa, ni posibilitan la integración con otros equipos de automatización. Además, los anteriores equipos tienen una característica que les limita significativamente el escalamiento y la flexibilidad de los esquemas de control y automatización, y es que son desarrollados mediante protocolos propietarios.

Por esta razón se hace necesario contar con un sistema de control y monitoreo de variables eléctricas en motores de inducción con protocolos de código abierto que permitan una interoperabilidad e integración con diferentes sistemas y dispositivos, y que, además, posibiliten la gestión de las variables eléctricas y la estimación de las condiciones de operación según los requerimientos definidos por el usuario.

Tomando en cuenta estas necesidades, se desarrolló este trabajo de investigación, en el cual, se implementó un sistema basado en protocolos de código abierto adaptable a diferentes tipos de dispositivos para el monitoreo de las variables eléctricas y control de un motor eléctrico, de manera remota e inalámbrica usando IoT.

Capítulo 1: Estado del arte

A continuación, se hace una reseña de los estudios encontrados en la bibliografía relacionados con el monitoreo de variables eléctricas y control de motores de inducción, donde se enfatiza en el tipo de tecnología utilizada, protocolos de comunicación y el software. Finalmente, se presenta una sinopsis de dicha revisión.

En [9] se presenta el diseño e implementación de una herramienta para realizar el monitoreo del comportamiento de un motor eléctrico de inducción trifásico, por medio del análisis de las corrientes eléctricas en sus fases, para lo cual desarrolló un algoritmo que permitiera la adquisición, procesamiento y envío de los datos tomados a la nube. Para esto se realizó el diseño de la arquitectura necesaria para el monitoreo de las corrientes en cada una de las fases; ésta cuenta con una componente física, compuesta por tarjetas de adquisición de datos y los sensores y un componente lógico, compuesto por el programa LabVIEW de *National Instruments* (NI). La herramienta tiene como ventaja su conexión inalámbrica, pero no permite el control sobre las variables eléctricas ni el motor.

En [10] se presenta un sistema de monitoreo, para el análisis dinámico de un motor eléctrico de inducción trifásico de 3 hp, en el Laboratorio de Control Electrónico de la Universidad de las Fuerzas Armadas ESPE, de Latacunga Ecuador, el cual consiste en la implementación de un sistema para la adquisición de datos, que permite almacenar las señales obtenidas en tiempo real por los diferentes sensores eléctricos y mecánicos, que miden: velocidad, corriente y tensión y que están instalados en el motor de inducción. Estas señales son analizadas posteriormente, por medio de aplicaciones de programación gráfica, logrando así tener el comportamiento de corrientes de arranque en función del tiempo, corriente de arranque en función de la velocidad, potencia generada y torque.

La adquisición de dichas señales electromecánicas es transmitida por medio de una interfaz entre la tarjeta de adquisición DAQ USB 6210-250 – 250 ks/s y el computador, que tendrá la función de medio de comunicación y análisis de la información capturada. Se utiliza LabVIEW, el cual es licenciado y permite observar el momento de arranque, inversión de giro, cambios de arranque de alta a baja tensión y arranque con cargas a los que se encuentra sometido el motor para las pruebas de monitoreo. No permite la interacción con otros dispositivos para realizar control sobre las variables que están siendo almacenadas.

En [11] se implementó una red GSM (Sistema Global para las Comunicaciones Móviles, por sus siglas en inglés) para el monitoreo de variables eléctricas en un banco de variadores de frecuencia que permiten el control de motores eléctricos de inducción en el Laboratorio de Control y Manipulación Automática de la Escuela de Ingeniería de Mantenimiento, en Riobamba en Ecuador. Este sistema tiene el fin de medir variables eléctricas tales como: tensión, corriente, potencia activa, reactiva y aparente, utilizando comunicación con tecnología GSM, haciendo necesaria la configuración para el envío y recepción de mensajes de texto entre un dispositivo móvil y un PLC S7 – 1200 (SIEMENS) y el software TIA PORTAL. Este sistema

permite la notificación a grandes distancias, pero no brinda la posibilidad de ejercer un control remoto.

En [12] se implementa un módulo para la adquisición de variables eléctricas de motores trifásicos asíncronos, el cual permite llevar a cabo la medición de cada parámetro en valores eficaces, presentarlos en una pantalla y subirlos a una plataforma del internet de las cosas, desde donde se puede monitorear la información de la red en la que está conectado el motor. Para la adquisición y comunicación se utilizó la placa de desarrollo de IoT ESP32, que envía los datos recolectados por cada sensor a la plataforma Ubidots, haciendo uso de un protocolo de comunicación. Ubidots es una plataforma habilitadora del Internet de las Cosas que permite a miles de ingenieros, desarrolladores y empresas, desplegar soluciones IoT y en la que se desarrolló el entorno gráfico que permite la visualización del comportamiento de cada variable. El sistema si bien es inalámbrico, solo permite el monitoreo de un motor a la vez, lo que se convierte en una limitación si el objetivo es hacer medidas en diferentes equipos.

En [13] se implementa un sistema que permite realizar la medición, monitoreo y análisis de variables eléctricas en motores eléctricos, con el objetivo de detectar perturbaciones en su funcionamiento normal, permitiendo realizar el diagnóstico de eficiencia y detección de fallas en las máquinas eléctricas. Para lo anterior, se creó una aplicación web que permite controlar el modo de trabajo de la máquina por medio de Internet. Este sistema permite el análisis de datos y la toma de decisiones basadas en dicho análisis. Está limitado en cuanto a escalabilidad e interconexión con otros sistemas.

En [14] se realiza la implementación de un sistema de adquisición de datos, el cual es implementado con una tarjeta Arduino UNO y el software LabVIEW, permitiendo obtener de forma automática las variables como tensión y velocidad, necesarias para caracterizar un motor eléctrico. Este sistema no transmite la información recolectada de forma inalámbrica, lo cual es una limitante para el momento de realizar el análisis de los datos recolectados.

En [15] se diseñó un prototipo que monitorea la temperatura en los motores eléctricos industriales, haciendo uso de sensores de temperatura DS18B20, el cual es compatible con la tarjeta Arduino y con módulos XBEE S2C. Los datos son retransmitidos a través de una red ZigBee, reportando en tiempo real el estado de temperatura que presenta el motor trifásico, activando alarmas en caso de que la temperatura sobrepase los valores normales estipulados. Es un sistema que funciona de forma inalámbrica, pero está limitado al monitoreo de una variable, que en este caso es la temperatura y, además, no permite realizar control sobre el motor.

En [16] se desarrolló un sistema de monitoreo predictivo, el cual es aplicado en motores de inducción trifásicos. Este permite realizar una planeación de los mantenimientos que requiere el motor y así evitar paradas no programadas y reducir costos. El sistema está constituido por los nodos de sensores, el Data-Logger y el software que permite realizar el control y análisis. Este sistema permite monitorear

las variables de interés, las cuales son enviadas de forma inalámbrica y posteriormente analizadas para hacer una toma de decisiones que permitan implementar un plan de mantenimiento adecuado de los motores. La interfaz es desarrollada en MATLAB ®, el cual es un software licenciado.

En [17] se presenta el diseño e implementación de un sistema de monitoreo en tiempo real de condiciones de motores de baja tensión industrial, haciendo uso de IoT. El sistema puede registrar y monitorear las condiciones de vibración y temperatura de un motor industrial y transmitir los datos a través de una red inalámbrica a un centro de registro de datos. El prototipo fue desarrollado usando código abierto. Este permite enviar un mensaje de texto cuando un motor se acerca a una condición anormal de funcionamiento, pero no permite al usuario realizar un control sobre el motor.

En [18] se implementó una arquitectura de adquisición y transmisión de datos bajo protocolo Modbus RTU utilizando Raspberry Pi3, la placa de adquisición se implementó con Arduino e Internet de las cosas (IoT). Las variables medidas son la tensión, la corriente y los estados de apertura/cierre de interruptores automáticos en motores eléctricos. Estos datos se envían a la Raspberry Pi y se almacenarán en servidores remotos donde el usuario puede acceder a ellos a través de una interfaz web. El sistema permite realizar control de forma remota y control sobre los interruptores que manejan las cargas conectadas a estos, por ejemplo, un motor eléctrico.

En [19] se presentan los resultados de un proyecto enfocado en las mediciones de vibraciones, temperatura, corriente y tensión de funcionamiento de las máquinas eléctricas para monitorear y analizar su condición de funcionamiento. Los datos de los sensores se transmiten de forma inalámbrica a la nube para ser procesados, evaluados y realizar predicción de fallas que permitan posteriormente implementar el mantenimiento más acorde a la situación. La adquisición de los datos de interés es realizada haciendo uso del software LabVIEW para el cual se hace necesario la compra de una licencia.

En [20] se proporcionan métodos y técnicas para realizar monitoreo de forma inalámbrica de los parámetros de un motor de inducción, utilizando Arduino UNO (para comunicación por cable) y unidad micro controladora Node (Node MCU) con módulo Wi-Fi (para comunicación inalámbrica). El software empleado es MATLAB ®. Los parámetros monitoreados son: forma de onda de la tensión, temperatura, valor rms de la tensión y corriente, valores pico a pico de corriente y voltaje, potencia. Se implementan sistemas de gestión de bases de datos gratuito y de código abierto para la gestión de los datos monitoreados, pero no es posible realizar control remoto sobre el motor.

En [21] se trabajó en el monitoreo de los parámetros de un motor trifásico, como lo son: corriente de fase, tensiones de fase, potencia activa, potencia reactiva, temperatura del motor, velocidad del motor y factor de potencia, los cuales son guardados en una base de datos. La interfaz gráfica de usuario fue desarrollada

en MATLAB®. Este desarrollo se implementó haciendo uso de dispositivos ZigBee, medidor de energía ADE7758, los cuales se comunican con MATLAB® GUI por medio de RS232. Este sistema permite encender o apagar el motor ante una falla, pero no es posible intervenir sus condiciones de funcionamiento, como por ejemplo la velocidad a la cual se mueve el rotor.

En [22] se muestra el diseño, implementación y prueba de un sistema industrial basado en IoT, el cual está pensado para realizar el monitoreo en motores eléctricos en tiempo real. El sistema ha sido diseñado utilizando componentes de hardware de bajo costo, un software de código abierto y una versión gratuita de un servicio de análisis de IoT en la nube, donde se almacena toda la información relevante. El módulo recopila datos en tiempo real sobre las vibraciones y la temperatura de un motor eléctrico. No es posible el control del motor, solo es posible el monitoreo de las variables anteriormente mencionadas.

En [23] se desarrolló un sistema que permite realizar monitoreo en tiempo real de la corriente trifásica de un motor eléctrico, por medio de una red inalámbrica. Con la ayuda de este sistema, los datos se pueden monitorear, guardar y luego transmitir a la nube de almacenamiento. El sistema contiene placa Arduino-UNO, sensor de corriente ACS-712, módulo Wi-Fi ESP-8266 que es el encargado de enviar la información a un servicio API IoT THING-SPEAK que se comporta como una nube y guarda la información recolectada por los sensores. No es posible realizar el monitoreo de otras variables ni el control de forma remota del motor.

En [24] se implementa un sistema de control y monitoreo de motores DC, en el cual se utiliza el protocolo ZigBee para transmitir la información generada por un conjunto de sensores utilizados para tomar los parámetros requeridos del motor DC, dentro de los cuales se tiene voltaje, velocidad, temperatura y datos de sobrecorriente. En este proceso es usado un microcontrolador para la recopilación y el almacenamiento de información. De acuerdo con la información monitoreada, el motor DC será detenido o puesto en marcha por la señal de control a través de la interfaz de usuario, cuando se cumple una determinada condición, pero no es posible hacer un control de velocidad o interconexión con dispositivos que permitan realizarlo.

En[25] se propone una implementación para estimar los parámetros del modelo simplificado del generador síncrono a partir de oscilogramas eléctricos bajo escenarios de operación diferentes. Para esto es elegido el modelo de máquina síncrona que es suministrado por las librerías de Sim Power Systems en Simulink del Software MATLAB®. Se realizaron pruebas con en dicho generador con variación de la carga en cada una de sus fases, desconexión de la carga en una de las fases, reducción del 50% en la carga en una de sus fases, obteniendo así los oscilogramas eléctricos que estarían relacionados con cada condición de operación, los cuales serían los utilizados para retroalimentar el módulo de estimación de parámetros ofrecido por Simulink-MATLAB®. Para lograr la estimación de los datos se implementó una serie de dispositivos y sensores controlados por una tarjeta de programación (STM32F411 Nucleo-64) en la plataforma MATLAB®.

En [26] se muestra el diseño e implementación de un sistema de monitoreo y registro de datos aplicado a grupos electrógenos, con el fin de visualizar y almacenar las variables generadas por el grupo electrógeno durante su operación. Para esto se utilizó un PLC S7-1200, pantallas HMI, sensores para la toma de las variables y el software de la National Instruments (NI) LabVIEW.

En la siguiente tabla se resumen los principales hallazgos del estudio del estado del arte.

Tabla 1. Análisis trabajos relacionados con módulos de monitoreo y control de variables eléctricas

Proyecto	Año	Software	Ventajas	Desventajas
Monitoreo de Motores de Inducción Trifásico a Través de las Corrientes Eléctricas de Alimentación	2021	LabVIEW	Cuenta con un software muy estable y de fácil manejo	Software pago el cual debe contar con una licencia que es de costos elevados, a su vez el sistema no permite hacer control sobre las variables ni el motor.
Diseño e implementación de un sistema de monitoreo, para el análisis dinámico de un motor eléctrico asíncrono trifásico de 3 Hp, en el laboratorio de control eléctrico de la Universidad de las Fuerzas Armadas ESPE Latacunga.	2019	LabVIEW	El sistema permite hacer medidas de variables importantes y utiliza software muy estable y de fácil manejo	Software pago el cual debe contar con una licencia que es de costos elevados, a su vez el sistema no permite hacer control sobre las variables ni el motor.
Implementación de una red GSM para monitoreo de variables eléctricas en un banco de variadores de frecuencia en el laboratorio de control y manipulación Automática de la escuela Ingeniería de mantenimiento	2019	TiaPortal	Se implementa el uso de software muy estable y de fácil manejo, el cual controla el PLC que es usado para centralizar el sentido de las variables y la notificación a grandes distancias.	El sistema no permite una escalabilidad o migración a otros dispositivos, al igual que no permite el control de forma remota.
Implementación de un sistema de monitoreo remoto de variables eléctricas para un motor trifásico de inducción en el laboratorio de máquinas eléctricas de la carrera de electricidad	2021	Software de protocolos abiertos	Permite hacer monitoreo de forma inalámbrica de las variables de interés.	Está limitado a un solo dispositivo de estudio, es decir, no permite hacer monitoreo en variables de varios motores el tiempo.
Sistema para medir, monitorear, analizar, y controlar disturbios de calidad, eficiencia y fallas en máquinas eléctricas, uso racional de la energía y el modo de trabajo de una maquina a través de internet.	2018	Software de protocolos abiertos	Permite realizar el monitoreo y análisis de variables eléctricas de un motor asíncrono y la toma de decisiones basado en dicho análisis	No permite realizar una expansión del sistema al igual que el control sobre el motor asíncrono.

Monitoreo y obtención de la caracterización de un motor DC mediante el desarrollo de un sistema de adquisición de datos en LabView para prácticas de estudiantes de electrónica del ISTCT	2021	LabVIEW	Permite realizar el monitoreo y análisis de variables eléctricas de un motor asíncrono y la toma de decisiones basado en dicho análisis	No permite realizar una expansión del sistema al igual que el control sobre el motor asíncrono.
Diseño de un sistema de monitoreo y alerta de temperatura para motores industriales utilizando tecnología ZIGBEE	2019	No especifica	Es un sistema que permite el monitoreo de temperatura de forma inalámbrica.	No permite tener control sobre el motor al que se le está monitoreando la temperatura.
Monitoreo, detección y diagnóstico de fallos en motores asíncronos trifásicos	2020	LabVIEW	Permite realizar el monitoreo y análisis de variables eléctricas de un motor asíncrono, para después implementar un plan de mantenimiento del motor.	No permite tener control sobre el motor al que se le está monitoreando la temperatura.
A Real-Time Condition Monitoring and Maintenance Management System for Low Voltage Industrial Motors Using Internet-of-Things	2020	No especifica	Se implementa el uso de software de código abierto.	No es posible realizar ningún tipo de control sobre el motor.
A secure wireless control of Remote Terminal Unit using the Internet of Things in smart grids	2018	No especifica	permite implementar control de forma remota en una red eléctrica que está alimentando cargas determinadas como motores.	No permite realizar un control en parámetros específicos del motor cuando está alimentado por la red eléctrica que se controla.
Condition Monitoring of Electrical Machines with Internet of Things	2018	LabVIEW-MATLAB	Se realiza el monitoreo de variables de forma inalámbrica en varios motores asíncronos	No existe la posibilidad de hacer control de variables ni del motor.
Data Acquisition System & Real Time Monitoring of the Parameters of Induction Motor via Wireless Communication	2020	MATLAB	Se implementan sistemas de gestión de bases de datos gratuito y de código abierto para la gestión de los datos monitoreados	Se usa software en los cuales es necesario la compra de licencia, además no es posible realizar ningún control sobre el motor.
Induction Motor Parameter Monitoring System using Zig bee Protocol & MATLAB GUI	2018	MATLAB GUI	Permite realizar el monitoreo de variables eléctricas del motor y el control del encendido o apagado ante una falla.	No permite control de condiciones de funcionamiento como la velocidad o conexión con otros dispositivos de control.

Low-cost real-time monitoring of electric motors for the Industry 4.0	2019	No especifica	Se logra hacer monitoreo de temperatura y vibraciones, haciendo uso de software de código abierto	Es imposible realizar control del motor e interconexión con dispositivos para este fin.
Real-Time Wireless Monitoring for Three Phase Motors in Industry: A Cost-Effective Solution using IoT	2020	No especifica	Se logra hacer monitoreo de la corriente trifásica del motor de forma inalámbrica.	El sistema no permite realizar control del motor e interconexión con dispositivos para este fin.
ZigBee-based Parameter Monitoring and Controlling Scheme for Multiple DC Motors	2020	No especifica	Con este sistema es posible realizar el monitoreo de variables como tensión, corriente, temperatura, entre otras, de forma inalámbrica.	El sistema no permite realizar control del motor e interconexión con dispositivos para este fin.
Estimación de los parámetros del generador síncrono del módulo TNA a partir de oscilogramas eléctricos	2020	Simulink-MATLAB	Software muy estable y de fácil manejo	Software pago y de costos elevados, no permite el acceso remoto.
Sistema de monitoreo de grupos electrógenos	2019	LabVIEW	Software muy estable y de fácil manejo	Software pago y de costos elevados, no permite el acceso remoto.
Diseño e implementación de un banco de pruebas para un generador de corriente continua, aplicando cargas variables para la visualización de parámetros eléctricos de funcionamiento a través de software LABVIEW.	2016	LabVIEW	Software muy estable y de fácil manejo	Software pago y de costos elevados, no permite el acceso remoto.

De la sinopsis realizada en la Tabla 1 se evidencia que la mayoría de los trabajos relacionados con la medición de variables eléctricas y el control remoto de los motores trifásicos de inducción se ha desarrollado usando herramientas de software comerciales que tienen costos significativos y aplicaciones restringidas. En su mayoría los desarrollos fueron implementados haciendo uso de LabVIEW y MATLAB los cuales no permiten hacer control sobre las variables monitoreadas y el motor, tampoco ofrecen flexibilidad en cuanto a la implementación de los dispositivos, y no se evidencia la posibilidad de que los sistemas sean escalables. Todo esto representa una desventaja, debido a que siempre se requiere personal en el lugar donde se encuentra el equipo.

Por lo anterior es importante mencionar nuevas tendencias de las tecnologías basadas en IoT, como lo es también el IIoT (Internet Industrial de las Cosas) el cual es una evolución del Internet de las Cosas que se enfoca específicamente en aplicaciones industriales y empresariales. El IIoT se centra en la conectividad de dispositivos y sistemas industriales para mejorar la eficiencia, la productividad y la toma de decisiones en entornos de fabricación y operaciones empresariales. Este

tipo de sistemas proporciona grandes ventajas debido que aprovecha tecnologías de conectividad avanzadas, como 5G y redes de baja potencia, para conectar una amplia variedad de dispositivos industriales, que pueden incluir sensores, máquinas, equipos de producción y sistemas de control, que es lo implementado en el desarrollo de esta tesis.

Capítulo 2: Marco teórico

Como ya se ha mencionado, se realizó el diseño e implementación de un sistema de monitoreo y control de variables en motores eléctricos de inducción de forma remota, basado en un protocolo abierto. Este enfoque permitió realizar una implementación de bajo costo y a su vez, la interacción con diferentes tecnologías, logrando así un sistema escalable y flexible. Como un complemento a la implementación de protocolo abierto, se implementa la arquitectura mostrada en la siguiente figura, que incluye una serie de sensores y equipos que hacen más fácil la escalabilidad del sistema implementado.

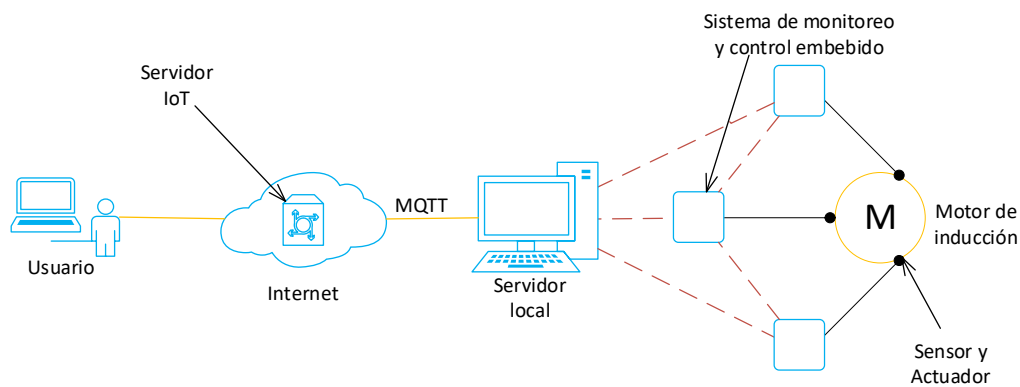


Figura 1: Arquitectura general del sistema propuesto.

En la Figura 1 se observa que el usuario puede acceder de forma remota al sistema, para controlar y hacer medición de variables eléctricas, mediante una conexión a Internet. Cada una de las variables obtenidas por las tarjetas y equipos de adquisición de datos son enviadas a un servidor local, donde son gestionadas y enviadas vía MQTT al servidor IoT en la red, el cual realiza su almacenamiento en una base de datos y permite generar un registro histórico, que posteriormente podría ser usado para gestión de mantenimiento y análisis de fallas de los motores de inducción.

A continuación, se definen y describen los principales conceptos, utilizados en la construcción de la arquitectura presentada en la Figura 1.

2.1 Internet de las Cosas (IoT)

El Internet de las cosas (IoT) hace referencia a la conexión de objetos de uso cotidiano a Internet, permitiendo intercambiar, agregar y procesar información relacionada con su entorno físico, logrando así proporcionar servicios de valor añadido a los usuarios que hacen uso de estos objetos. La conexión a Internet se logra por medio de redes guiadas o inalámbricas, permitiendo una nueva forma de interacción en el mundo físico, basado en la idea de ubicuidad y apoyado por el desarrollo y avance de las Tecnologías de Información y Comunicación (TIC) [27]. Este tipo de tecnologías también permite realizar el reconocimiento de eventos o cambios que se presentan, logrando reaccionar de forma automática, eficaz y adecuada ante dicha incidencia. Su objetivo es brindar una infraestructura que permita superar la barrera existente entre los objetos en el mundo físico y su representación en los sistemas de información.

Dentro de los componentes principales del Internet de las cosas se pueden mencionar los siguientes:

- ✓ **Dispositivos IoT:** Estos son los objetos físicos que están equipados con sensores, actuadores y conectividad a Internet. Pueden ser una amplia variedad de dispositivos, como sensores de temperatura, cámaras de seguridad, electrodomésticos, automóviles conectados, *wearables* y mucho más.
- ✓ **Sensores y actuadores:** Los sensores recopilan datos del entorno, como temperatura, humedad, luz, movimiento, etc. Los actuadores son componentes que permiten que los dispositivos realicen acciones en función de la información que reciben, como encender o apagar una luz, abrir una cerradura o ajustar una temperatura.
- ✓ **Conectividad:** Los dispositivos IoT se comunican a través de diferentes tecnologías de conectividad, como Wi-Fi, Bluetooth, Zigbee, LoRa, 4G, 5G, entre otras. La elección de la tecnología depende de la aplicación y los requisitos de la red.
- ✓ **Plataformas IoT:** Estas son infraestructuras de software que gestionan la conectividad, el almacenamiento de datos, el procesamiento y la gestión de dispositivos IoT. Las plataformas IoT permiten a las empresas, usuarios en general y los desarrolladores administrar y analizar datos de dispositivos, así como crear aplicaciones y servicios basados en IoT.
- ✓ **Nube:** La nube es fundamental para el almacenamiento y el procesamiento de grandes cantidades de datos generados por dispositivos IoT. La información recopilada se envía a la nube para su análisis y almacenamiento seguro, lo que permite extraer información valiosa y tomar decisiones basadas en datos.
- ✓ **Aplicaciones y servicios:** Las aplicaciones y servicios IoT permiten a los usuarios interactuar con los dispositivos y acceder a los datos recopilados. Estas aplicaciones pueden ser aplicaciones móviles, aplicaciones web o software específico que aprovecha los datos de IoT para proporcionar valor a los usuarios finales.

- ✓ **Seguridad:** La seguridad es un componente crítico en el IoT, ya que los dispositivos están conectados a Internet y pueden ser vulnerables a amenazas cibernéticas. Se requieren medidas de seguridad robustas, como autenticación, cifrado y actualizaciones de firmware, para proteger la integridad de los datos y la privacidad de los usuarios.

Estos son los componentes principales del Internet de las Cosas que trabajan en conjunto para crear un ecosistema en el que los objetos y dispositivos pueden comunicarse, recopilar datos y ofrecer servicios inteligentes a los usuarios [27].

2.2 MQTT

El Transporte de Telemetría usando Mensajes por Cola (MQTT- Por sus siglas en inglés) es un protocolo de mensajería estándar que se emplea en la comunicación Máquina a Máquina (M2M – Por sus siglas en inglés) en los sistemas que usan IoT. Se orienta principalmente a la comunicación de sensores debido al bajo consumo de banda ancha y mínima sobrecarga, lo que permite que sea utilizado en dispositivos que poseen pocos recursos [28].

Dentro de sus principales características se encuentran las siguientes:

- ✓ Protocolo de mensajería de publicación / suscripción ligera implementándose en servidores bróker
- ✓ Diseñado para ser abierto, sencillo, ligero y fácil de implementar
- ✓ Ideal para uso en entornos restringidos, pero no por ello sólo útil en estos entornos
- ✓ Buen uso en dispositivos con procesador limitado o memoria
- ✓ Buen uso cuando la red es costosa tiene poco ancho de banda o no es fiable
- ✓ Uso de TCP/IP para proveer una básica conectividad de red
- ✓ Entre otras.

MQTT se basa en un modelo de publicación/suscripción (Pub/Sub), que permite a los dispositivos enviar y recibir datos de manera eficiente. En este modelo, los dispositivos pueden ser "editores" (publishers) que envían mensajes o "suscriptores" (subscribers) que reciben y procesan esos mensajes. Los dispositivos no necesitan estar directamente conectados entre sí; en su lugar, se comunican a través de un intermediario llamado "broker MQTT."

El broker MQTT es el núcleo del sistema y actúa como un intermediario que gestiona la distribución de mensajes entre los editores y suscriptores. Los editores envían mensajes al broker, y este los reenvía a los suscriptores apropiados. El broker es responsable de mantener un registro de todos los temas (topics) a los que están suscritos los clientes y de garantizar que los mensajes se entreguen de manera confiable.

Los mensajes MQTT se organizan en torno a temas (topics), el cual es una cadena de caracteres que actúa como un identificador para un cierto tipo de datos o canal de comunicación. Los suscriptores pueden registrarse para recibir mensajes

relacionados con un tema específico, lo que les permite filtrar los mensajes y recibir solo los que les interesan [28].

2.3 Node-red

Para la implementación del servidor local y el servidor en la nube se usó Node-red, el cual es una herramienta de programación visual que permite crear aplicaciones IoT (Internet de las cosas) y de automatización del hogar, monitoreo y control de sensores, dispositivos, integración de servicios web y procesamiento en tiempo real.

Node-red utiliza una interfaz de usuario gráfica basada en navegador web utilizando un enfoque basado en nodos, que permite a los usuarios arrastrar y soltar nodos y conectarlos visualmente para crear flujos de trabajo. Los nodos son bloques de construcción de la aplicación, y cada uno tiene una función específica, como leer sensores, analizar datos o enviar alertas.

Node-red se basa en Node.js, entorno de código abierto multiplataforma lo que significa que puede ser ejecutado en cualquier plataforma que soporte Node.js, como una Raspberry Pi, un servidor en la nube o una computadora personal.

En resumen, se puede decir que el software Node-red es una herramienta de programación visual que permite crear soluciones IoT y automatización de procesos de manera rápida y sencilla, con una interfaz gráfica intuitiva y una amplia comunidad de desarrolladores que contribuyen con nodos personalizados para integrar con diferentes dispositivos y servicios [29].

2.4 MariaDB

MariaDB es un sistema de gestión de bases de datos relacionales de código abierto, que se ha convertido en una alternativa popular y confiable frente a otros sistemas de bases de datos como MySQL y Oracle. Esta plataforma de código abierto cuenta con una comunidad activa y una gran cantidad de recursos disponibles en línea, lo que la hace fácil de usar e implementar para una amplia gama de aplicaciones. Además, cuenta con reconocidos usuarios como Google y WordPress, lo que demuestra su gran eficacia en entornos empresariales de alto rendimiento.

MariaDB Server está disponible en una amplia variedad de sistemas operativos, incluyendo Linux, Windows y Mac OS. Proporciona una gama de características avanzadas, incluyendo soporte para transacciones de alta disponibilidad, análisis de datos y una variedad de herramientas y aplicaciones de terceros. Fue creada por los desarrolladores originales de MySQL, después de que esta última fuera adquirida por Oracle Corporation. Desde entonces, ha ganado una gran cantidad de seguidores gracias a su enfoque de código abierto y su compromiso con la compatibilidad con MySQL. La plataforma cuenta con una gran cantidad de recursos

y soporte, lo que la hace una excelente opción para empresas y desarrolladores en todo el mundo [30].

2.5 Interfaz de usuario

La interfaz de usuario o *dashboard*, es una herramienta de gestión de información que permite monitorear, analizar y visualizar la información monitoreada por medio de cuadros de mando interactivos, los cuales están constituidos por gráficos y botones que enlazan sensores y actuadores, que son el origen de los datos mostrados. Estos componentes muestran los datos de forma compacta y visual, lo que puede ayudar a los usuarios de cuadros de mandos a ver las tendencias y tomar decisiones [31].

Para este proyecto la interfaz de usuario fue implementada usando Node-red Dashboard que es una herramienta de visualización web que se integra con la plataforma de programación visual Node-red. Permite crear paneles interactivos y personalizados para mostrar datos y controlar dispositivos, todo desde una interfaz gráfica fácil de usar. Los paneles se pueden acceder desde cualquier navegador web, lo que los hace ideales para monitorear y controlar sistemas remotos o dispositivos conectados a Internet de las cosas (IoT).

Además, proporciona una variedad de *widgets* y plantillas predefinidas que pueden ser personalizadas de acuerdo con las necesidades del usuario. Los *widgets* incluyen gráficos de barras, gráficos de líneas, medidores, interruptores, botones, cajas de selección y mucho más. Además, se pueden integrar con otros nodos de Node-red para agregar lógica personalizada o manipulación de datos antes de que se visualicen en el panel.

Node-red Dashboard se instala como un paquete de Node.js y se integra con Node-red de forma sencilla. La documentación detallada y la comunidad activa también hacen que tenga un aprendizaje rápido y que sea fácil utilizar, incluso para aquellos con poca experiencia en programación.

Dicho lo anterior se puede decir que Node-red Dashboard es una herramienta poderosa y versátil para la visualización de datos y el control de dispositivos a través de la web. Con una interfaz gráfica intuitiva y una gran cantidad de *widgets* personalizables, es una excelente opción para la creación de paneles de control en tiempo real para una variedad de aplicaciones [32].

2.6 Sistemas embebidos

Son un conjunto de componentes electrónicos que permiten realizar funciones dedicadas, constan de un procesador central, que tiene como función recibir, analizar y procesar la información recolectada por los diferentes sensores, y posteriormente transmitir señales a los actuadores que son los encargados de

desarrollar las funciones específicas requeridas, normalmente forman parte de otros sistemas de mayor envergadura. La característica principal es que emplea para ello uno o varios procesadores digitales en formato microprocesador, microcontrolador o procesador de señales digitales (DSP por sus siglas en inglés) lo que le permite aportar 'inteligencia' al sistema anfitrión al que ayuda a gobernar y del que forma parte [33].

Dentro de sus características se nombran las siguientes:

- ✓ Procesador central: Estos módulos centrales para el procesamiento pueden estar conformados por varios procesadores digitales (CPUs) en formato microcontroladores, microprocesadores, FPGAS, DPS o una combinación de estos, permitiendo aportar 'inteligencia' en el sistema que ayuda a gobernar
- ✓ Son sistemas confiables
- ✓ No exigen un mantenimiento dedicado
- ✓ Cuentan con niveles de seguridad importantes
- ✓ Son diseñados para realizar actividades de propósito específico, y pueden contar con una interfaz de uso específico para la interacción con los usuarios.

En la siguiente Figura se muestra un ejemplo de la arquitectura para los sistemas embebidos [33].

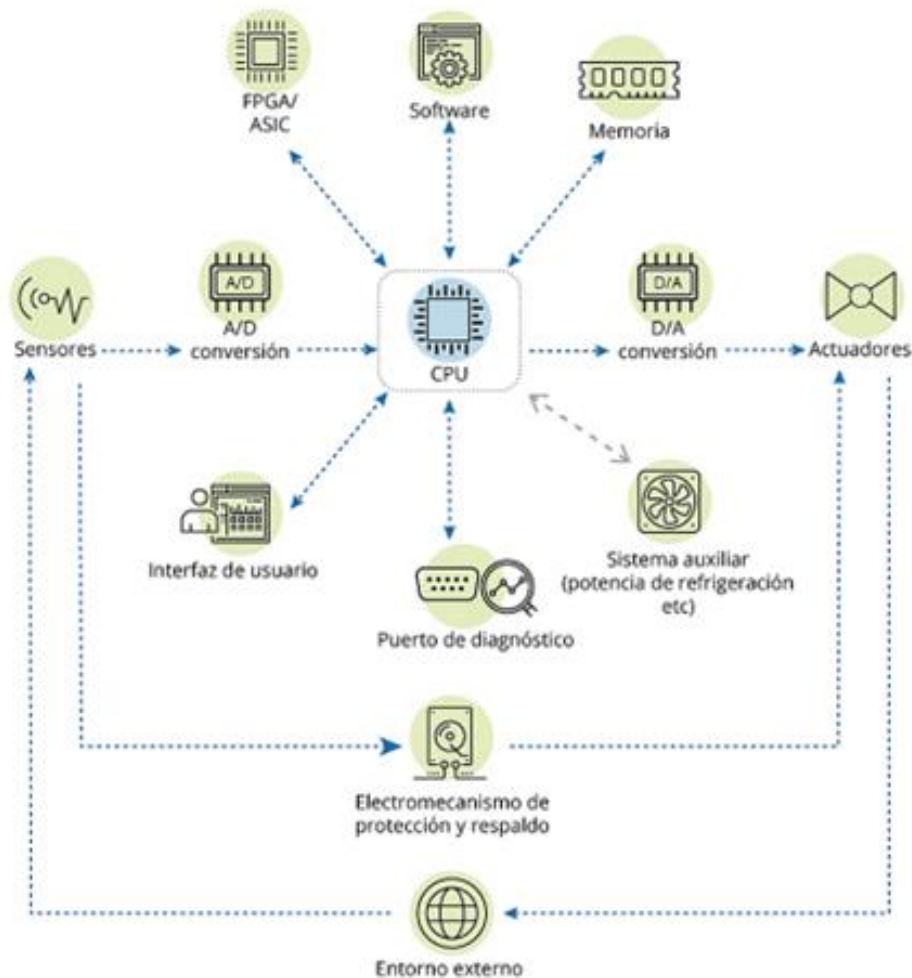


Figura 2: Arquitectura de los sistemas embebidos [33].

2.7 Red inalámbrica

Una red inalámbrica es la que usa las ondas de radio para hacer la conexión de dispositivos sin la utilización de cables. Estas funcionan de forma semejante a una red cableada, sin embargo, debe realizar la conversión de las señales de información a la forma adecuada para la transmisión a través del aire. Este tipo de redes tienen diversas aplicaciones, dentro de las cuales se puede mencionar la sustitución de las redes cableadas para acceder a datos recolectados por sensores que están en ubicaciones de difícil acceso o en entornos donde la movilidad es una necesidad, como en redes de dispositivos móviles o en aplicaciones de Internet de las cosas.

Las redes inalámbricas utilizan una variedad de tecnologías, como Wi-Fi, Bluetooth, Zigbee, LoRa, y otras formas de comunicación inalámbrica, para permitir la comunicación entre dispositivos. Estas tecnologías proporcionan diferentes niveles

de alcance, velocidad, seguridad y otras características, lo que las hace adecuadas para diferentes aplicaciones y entornos.

Además de su capacidad de proporcionar conectividad sin cables, las redes inalámbricas también pueden ser más fáciles de configurar y mantener que las redes con cableado físico. Las redes inalámbricas también permiten la expansión fácil y la integración con otras redes existentes, lo que las hace muy populares en entornos empresariales y de consumo.

Por lo anterior, es posible decir que las redes inalámbricas son una forma conveniente y eficaz de conectar dispositivos y proporcionar conectividad en una amplia gama de entornos. A medida que la tecnología continúa avanzando, se espera que las redes inalámbricas sean cada vez más ubicuas y estén presentes en una variedad de aplicaciones y dispositivos [34].

2.8 Buses de campo

Son sistemas conformados por dispositivos de campo (sensores y actuadores) que transmiten información entre ellos, lo cual permite simplificar notablemente el uso en máquinas y equipos industriales usados en diversos procesos de producción. Los buses de campo se pueden clasificar como propietarios y abiertos [35].

- ✓ **Buses propietarios:** Son aquellos que pertenecen a compañías, los cuales requieren para ser utilizados la adquisición de una licencia, la cual normalmente es de un precio considerable.
- ✓ **Buses abiertos:** Son aquellos en los cuales no es necesario obtener licencias pagas para hacer uso de estos, sus especificaciones suelen ser de uso público y sus componentes críticos también se encuentran disponibles, al igual que los procesos de verificación y validación.

Dentro de las características principales que debe cumplir un bus de campo, relacionadas a la conexión de dispositivos, se pueden mencionar las siguientes [35]:

- ✓ **Interconectividad:** radica en que deben permitir la conexión de dispositivos de diversos fabricantes, siempre y cuando estos sean compatibles con el protocolo que se esté utilizando en las aplicaciones para las que estos se requieren.
- ✓ **Interoperabilidad:** tienen la capacidad de intercambiar información y a su vez usar la información intercambiada.
- ✓ **Intercambiabilidad:** hace referencia a que los dispositivos utilizados pueden ser reemplazados por otros de diferentes fabricantes, y que sean equivalentes.

Los buses de campo al ser debidamente escogidos para una aplicación determinada permiten grandes ventajas como:

- ✓ **Flexibilidad:** esta radica en que, para realizar el uso de un nuevo dispositivo en el sistema, solo es necesario una conexión eléctrica a este y la posterior configuración que puede ser de forma remota.

- ✓ **Reducción de la complejidad del sistema de control en términos de hardware, en cuanto a:**
 - Reducción drástica del cableado, tiempo de instalación y por ende la cantidad de personal calificado para dicha actividad.
 - Se mitiga la necesidad de hacer uso de grandes montajes de tableros de control.
- ✓ **Facilidad de mantenimiento:** permite que la tarea de mantenimiento sea mucho más fácil, debido a que se pueden identificar de forma remota los dispositivos que no están dentro de su funcionamiento normal, lo que permite reducir tiempos de parada y por ende, pérdidas en la producción [35].

2.9 Motores Eléctricos

Los motores eléctricos son máquinas rotativas que tienen la capacidad de transformar la potencia eléctrica en potencia mecánica. Existen diferentes tipos de motores eléctricos trifásicos de inducción, dentro de los cuales se pueden mencionar los motores asíncronos de rotor bobinado y motores asíncronos de rotor jaula de ardilla. Estos motores son utilizados con gran frecuencia en los procesos industriales, ya que son de construcción simple, robustos y no requieren de un mantenimiento riguroso [36].

Motor de inducción con rotor jaula de ardilla

Este motor es también llamado motor con rotor en cortocircuito, es el más usado actualmente en los procesos industriales, su nombre hace alusión a que su rotor está construido por un conjunto chapas de acero al silicio laminadas, ver Figura 3, en cuyo interior se disponen unas barras, generalmente de aluminio moldeado a presión [36].

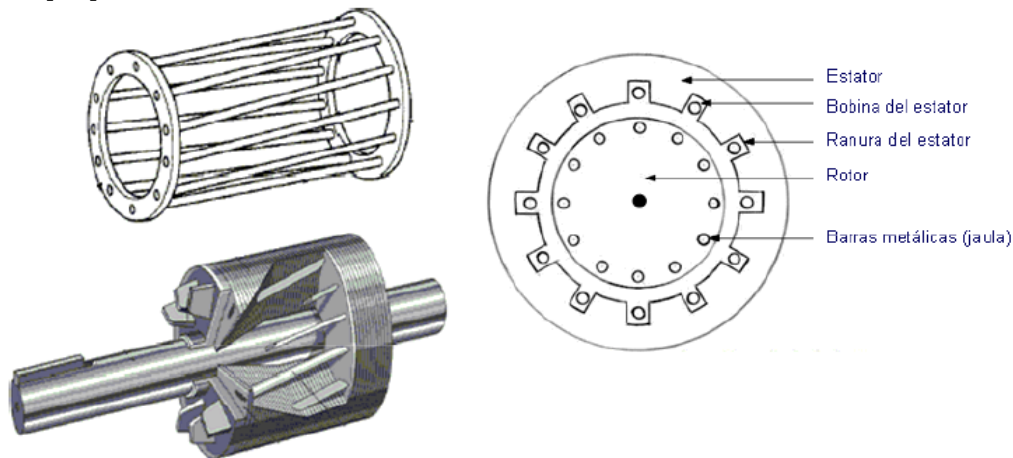


Figura 3: Rotor jaula de ardilla o en cortocircuito [36].

Motor de inducción de rotor devanado

Este tipo de motor se caracteriza por tener en su rotor un devanado trifásico de cobre, el cual está dispuesto en las ranuras de rotor, que va conectado a tres anillos metálicos por uno de sus extremos, y en el otro extremo están conectados en estrella [36].

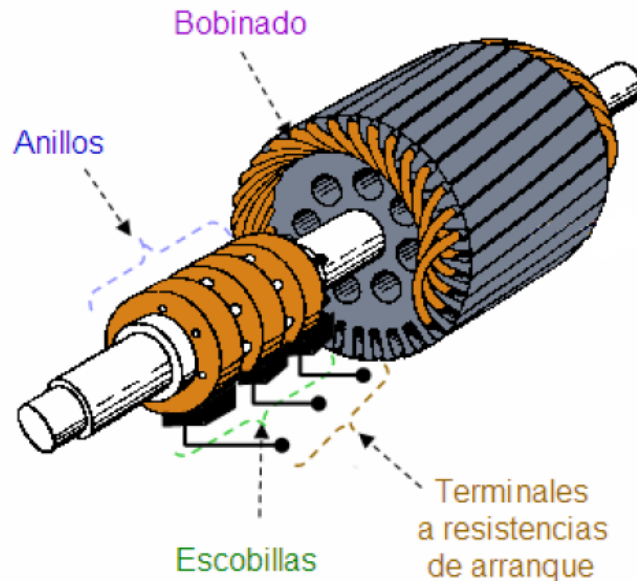


Figura 4: Rotor jaula de ardilla o en cortocircuito [36].

Previamente se describieron las principales componentes, de hardware y software, del sistema propuesto. A continuación, se describe una de las principales aplicaciones de los datos medidos mediante el sistema propuesto, enfocada al análisis de fallas y diagnóstico de las condiciones de operación de los motores de inducción.

Fallas en los motores de inducción

Un motor de inducción es un sistema electromecánico, por lo que pueden presentarse fallas de naturaleza eléctrica y mecánica. De acuerdo con estudios estadísticos, se reporta que los porcentajes de fallas en máquinas eléctricas corresponden en un 40% a fallas de cojinetes, 38% fallas del estator, 12% fallas del rotor y 10% a otro tipo de fallas [36]. A continuación, se mencionan algunas de las causas de las fallas.

Causas de fallas en motores de inducción

- ✓ **Efecto electromagnético:** Las fuerzas electromagnéticas son proporcionales al cuadrado de la corriente, siendo unidireccionales y tienden a dislocar las barras produciendo una deflexión en las mismas. El mayor esfuerzo sobre las barras ocurre durante el arranque del motor, debido a las altas corrientes generadas [16].
- ✓ **Arranques cíclicos:** Al someter el motor a numerosos arranques en un corto periodo de tiempo, se genera un aumento de la temperatura en los bobinados, lo cual es consecuencia de las elevadas corrientes de arranque.
- ✓ **Obstrucción del sistema de ventilación:** El sistema de ventilación del motor es el encargado de disipar el calor generado en el rotor y estator, y si por alguna razón este sistema no está funcionando de forma óptima, se generará un sobrecalentamiento que finalmente afectará el bobinado del estator.
- ✓ **Sobretensiones de corta duración:** Estas ocurren con frecuencia durante el uso del motor, lo que genera la degradación del aislamiento del bobinado y por ende una reducción en la vida útil del motor, dando lugar a fallas prematuras.
- ✓ **Sobrecorriente:** Las corrientes excesivas, ya sea debido a problemas en la red eléctrica o en el propio motor, pueden dañar el devanado del motor y sus componentes.
- ✓ **Fallas en el Aislamiento:** La degradación del aislamiento eléctrico puede causar cortocircuitos y daños en los devanados. Esto puede ser causado por sobretensiones, humedad, calor excesivo o envejecimiento del material aislante.
- ✓ **Sobrecargas operativas:** Estas se producen cuando el motor está sometido a una carga excesiva, generando un consumo elevado de corriente, par insuficiente para mover la carga y aumento considerable de la temperatura, este aumento de temperatura es una de las causas principales de fallo del motor [36].
- ✓ **Desequilibrio de tensión:** Puede generar esfuerzos en los circuitos de cada fase del sistema eléctrico, creando flujos de corriente con niveles excesivos en dichas fases, trayendo como consecuencia el aumento de temperatura interna de los bobinados, lo que lleva a un deterioro del aislamiento [36].

Control de motores de inducción

Los sistemas de control implementados para el manejo de motores de inducción son sistemas diseñados para controlar y gestionar la operación de estos. Estos sistemas de control están diseñados para lograr una operación precisa y eficiente del motor, y pueden variar en complejidad según las necesidades específicas de la aplicación. A continuación, se proporciona una descripción general de los componentes y conceptos clave de estos sistemas:

- ✓ **Variadores de frecuencia (VFD):** Los VFD son dispositivos electrónicos que controlan la velocidad y el par de un motor de inducción al variar la frecuencia y la tensión suministrada al motor. Esto permite una regulación precisa de la velocidad y el torque, lo que a su vez proporciona un mayor control sobre la aplicación. Los VFD son esenciales en sistemas de control de motores de inducción.

- ✓ **Controlador de motor:** Un controlador de motor es una parte integral del sistema de control. Puede ser un microcontrolador, un PLC (Controlador Lógico Programable) u otro dispositivo que ejecuta algoritmos de control para gestionar el funcionamiento del motor. El controlador toma entradas de sensores y comandos del usuario para ajustar la velocidad, dirección y otras características de funcionamiento del motor.
- ✓ **Sensores:** Los sensores se utilizan para proporcionar retroalimentación al sistema de control. Los sensores comunes incluyen encoders para medir la velocidad y la posición del rotor, sensores de corriente, voltaje y temperatura para monitorear cada una de estas variables del motor.
- ✓ **Interfaz de usuario:** Muchos sistemas de control de motores de inducción incluyen una interfaz de usuario que permite a los operadores supervisar y ajustar parámetros de funcionamiento, como la velocidad, el sentido de giro y la potencia. Esto se puede hacer a través de una pantalla táctil, una aplicación de software o paneles de control físicos.
- ✓ **Algoritmos de control:** Los algoritmos de control son programas o lógica incorporada en el controlador del motor que determinan cómo el motor debe responder a las entradas y las condiciones del sistema. Estos algoritmos pueden incluir control de lazo cerrado para mantener una velocidad o posición deseada, control de torque, y protección contra condiciones de funcionamiento anormales.
- ✓ **Protección y seguridad:** Los sistemas de control de motores de inducción también incorporan características de protección y seguridad. Esto incluye la capacidad de detectar sobrecorriente, sobrecalentamiento y otras condiciones anormales que podrían dañar el motor.
- ✓ **Comunicación y supervisión:** En aplicaciones industriales, es común que los sistemas de control de motores de inducción estén conectados a redes de comunicación para la supervisión remota y la recopilación de datos. Esto permite el monitoreo en tiempo real y la solución de problemas a distancia.

Estos sistemas de control de motores de inducción son esenciales en una amplia gama de aplicaciones, desde maquinaria industrial y sistemas de bombeo hasta sistemas de climatización y transporte. La elección de los componentes y la complejidad del sistema dependen de los requisitos específicos de la aplicación [36].

Capítulo 3: Descripción general del sistema propuesto

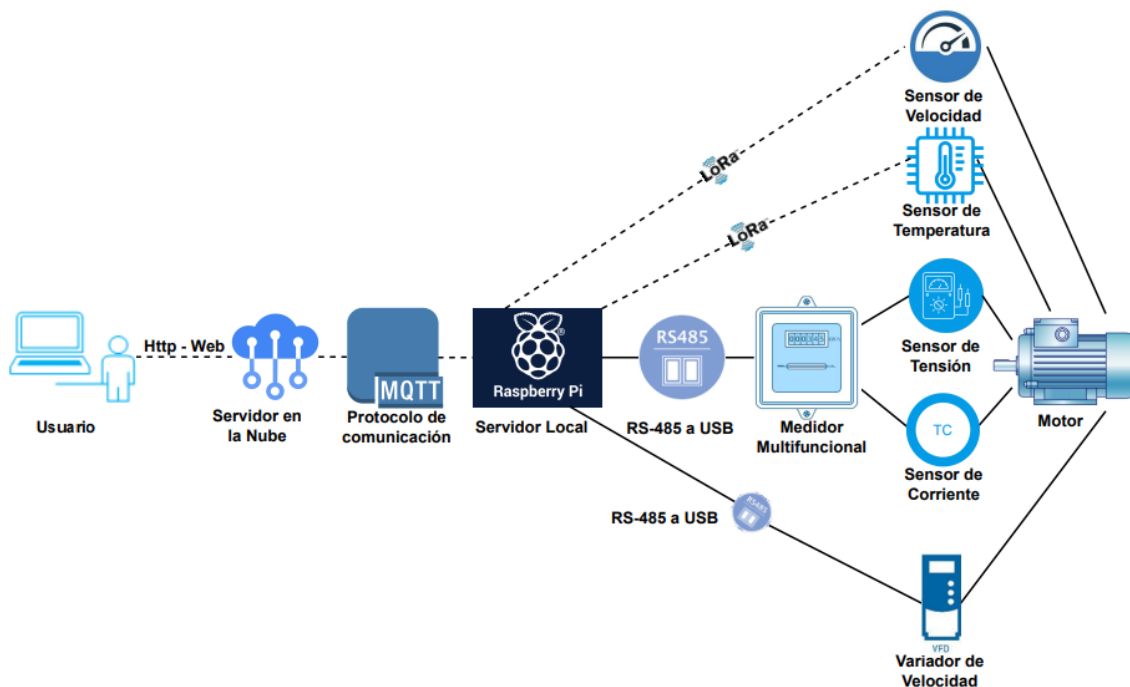


Figura 5: Sistema implementado.

En esta sección se describen en detalle las diferentes etapas que conforman el sistema descentralizado remoto, basado en tecnologías IoT y de código abierto, para la adquisición de variables eléctricas de un motor de inducción trifásico.

Como se observa en Figura 5, el sistema está constituido por una serie de módulos y dispositivos, que permiten el monitoreo de velocidad angular del motor, temperatura en las espiras del bobinado estatórico, tensión, corriente, potencia de alimentación y control de marcha y parada del motor.

1. Para la medición de velocidad y temperatura se diseñó una tarjeta, que entre sus componentes cuenta con un microcontrolador PIC16F1827; un dispositivo de comunicación de largo alcance (LoRa - Long Range Por sus siglas en inglés) para la comunicación inalámbrica con el servidor local y un DIP Switch, el cual permite realizar la asignación del ID a cada una de las tarjetas que componen el sistema general. Para la medición de la velocidad se instaló un encoder en el motor de inducción, que es el encargado de enviar los pulsos que posteriormente son procesados e interpretados por medio de la tarjeta de adquisición de datos, lo que permite calcular la velocidad en RPM a la que gira el eje del motor. La información procesada es enviada a un servidor en la nube, donde se almacena en una base de datos y posteriormente es presentada en la interfaz de usuario.

2. Por su parte, las variables eléctricas en las líneas de alimentación del motor de inducción; tensión, corriente y potencia, entre otras, son adquiridas a través del medidor multifuncional EasyLogic PM2120 de Schneider Electric [4], que cuenta con comunicación modbus RS485, permitiendo de esta manera que la información de las variables medidas sea recopilada y enviada al servidor local, el cual las publica posteriormente en el servidor IoT que se encuentra en la nube.
3. El servidor local se implementó con una SBC (Single Board Computer- Por sus siglas en inglés), que para este caso es una Raspberry Pi 4 modelo B, que funciona con el sistema operativo Raspbian. En esta SBC se realiza la instalación del software Node-red por medio del cual se implementa el código de programa respectivo que permite la recepción de los datos que envían los transductores o equipos de adquisición de datos. Esta información se envía posteriormente al servidor en la nube que se encarga de almacenarla y realizar la debida gestión para reportar por medio de alarmas (mensajes de texto) al usuario los funcionamientos anormales que se puedan presentar, de igual forma aloja la configuración de la interfaz de usuario.
4. La gestión de la información en la nube se realiza mediante el servidor Mosquitto, el cual es un popular servidor MQTT de código abierto que se utiliza para implementar el protocolo MQTT (Message Queuing Telemetry Transport). MQTT es un protocolo de mensajería ligero diseñado para la comunicación entre dispositivos en redes con ancho de banda limitado y alta latencia, lo que lo hace ideal para aplicaciones IoT (Internet of Things) y M2M (Machine-to-Machine).

3.1 Especificación de los requerimientos técnicos del sistema

Cada uno de los elementos que conforman el sistema fueron escogidos con base en las especificaciones técnicas que presentan y que se ajustan a los parámetros o características técnicas de funcionamiento del motor de inducción utilizado. A continuación, se describen las razones que determinaron la escogencia de cada uno de estos elementos.

- ✓ **Medida de temperatura:** el objetivo es medir la temperatura en los devanados de un motor de inducción. Por lo tanto, para determinar el sensor a utilizar se debe tenerse en cuenta el acceso al devanado, dicho sensor debe ser compacto, que no afecte o sea afectado por los campos Electromagnéticos presentes en el devanado, la temperatura máxima de funcionamiento del motor la cual es de 115°C debido a que el aislamiento de este es clase F. Debido a lo anterior se optó por el sensor de temperatura análogo MCP9700, el cual tiene un rango de temperatura -40°C a 150°C, presenta un bajo consumo de energía, lo que lo hace adecuado para aplicaciones de bajo consumo y con restricciones de energía.
- ✓ **Medida de velocidad:** para determinar la velocidad angular del rotor del motor de inducción, el cual cuenta con una velocidad nominal de 1680 RPM,

se hizo uso del encoder E6B2-CWZ3E de la marca Omron, el cual es un encoder rotativo incremental de 2000 P/R, lo que le permite contar con una alta sensibilidad. Estos pulsos se utilizan para determinar la dirección y la magnitud del movimiento, cuenta con una velocidad máxima de 6000 RPM y está diseñado con protección contra polvo y humedad para su uso en entornos industriales.

- ✓ **Medición de variables eléctricas:** dado que, entre las características técnicas de funcionamiento del motor de inducción este requiere una alimentación de 220/380V y una corriente de 12.8/6.8A, dependiendo si la conexión es estrella o delta (Δ/Y). por lo anterior se escogió el medidor multifuncional EasyLogic PM2120 de Schneider Electric, el cual es un dispositivo que sus rangos de medición permiten sensar voltajes en un rango que va desde 100 a 600 V en tensiones línea-línea y des 57 a 347 V en tensiones Línea-neutro. Pueden medir corrientes nominales bajas en el rango de miliamperios (mA) o microamperios (μA) hasta 20A de corriente nominal. Es importante destacar que, en aplicaciones donde se esperan corrientes muy altas que superan la capacidad máxima de medición del medidor, se suelen utilizar transformadores externos, por lo que son una herramienta importante para el desarrollo del presente sistema.
- ✓ **Control de velocidad y sentido de giro:**
Para el control de velocidad y sentido de giro, se utiliza un variador de velocidad altivar 58, el cual fue conectado a un PLC siemens S1200 y que a su vez es controlado por una Raspberry Pi 4 modelo B, que funciona como servidor local, a la cual llegan los comandos por MQTT, que son posteriormente convertidos por medio de la librería o plugin Snap 7 para Node-red a protocolos compatibles con el PLC. El control se puede realizar directamente con el variador, si este soporta comunicación ethernet o par diferencial. Para este caso fue necesario el uso del PLC debido a que el variador no cuenta con un periférico de comunicación.

3.2 Implementación del sistema propuesto

En este capítulo se describe de manera detallada el diseño de la arquitectura implementada en el sistema que permite la adquisición de temperatura, velocidades, variables eléctricas y control de un motor de inducción.

3.2.1 Tarjeta de adquisición de datos de temperatura y velocidad

se presentan las diferentes etapas y componentes que hacen parte del diseño de las tarjetas electrónicas de adquisición de temperatura y velocidad en el motor de inducción, cuyas etapas se describen a continuación.

3.2.2 Etapa de polarización

Esta etapa es la encargada de la energización de la tarjeta, está conformada por una conexión tipo bornera, que permite la conexión de la fuente de referencia HLK – PM01 a una red eléctrica de corriente alterna, 50 – 60 Hz, tensión de 100 – 240 VAC. El consumo de esta fuente es de 0,2 A, mientras que la salida es una señal de corriente directa de 5 VDC con una corriente máxima 0,6 A. Entre las terminales +5V y AGND se conecta en paralelo un condensador de desacople encargado de realizar el filtrado de las señales parasitas que puedan generar un funcionamiento inadecuado del sistema, en especial el microcontrolador. En la Figura 6 se muestra el esquemático de la etapa de polarización de la tarjeta.

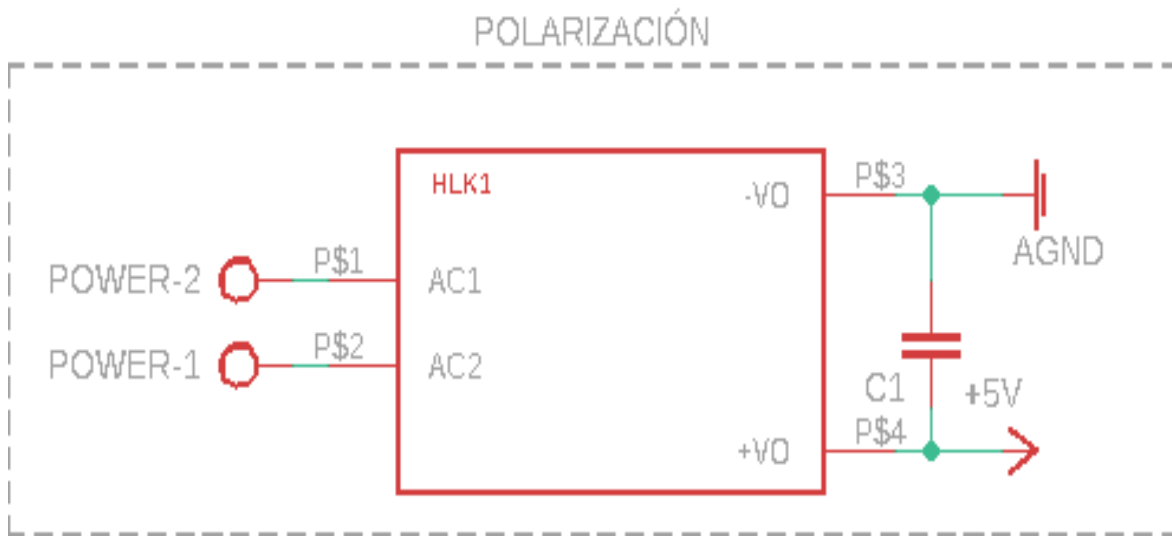


Figura 6: Etapa de polarización de la tarjeta de adquisición de temperatura y velocidad.

3.2.3 Etapa de conexión de sensores de velocidad y temperatura

El circuito de conexión de los sensores encargados de tomar los datos tanto de temperatura como de velocidad consta de una bornera de conexión de tres entradas en la que se realiza la conexión de la sonda de temperatura o el encoder, ver Figura 7. La terminal marcada como SENSOR-1 es la encargada de la polarización de los sensores de velocidad y temperatura, la terminal SENSOR-3 es la referencia o conexión a tierra, mientras que por medio de la SENSOR-2 ingresa la señal de pulsos del encoder o señal de temperatura. Es de aclarar que esta etapa solo puede hacer la medición de temperatura o velocidad, dependiendo de la previa programación realizada.

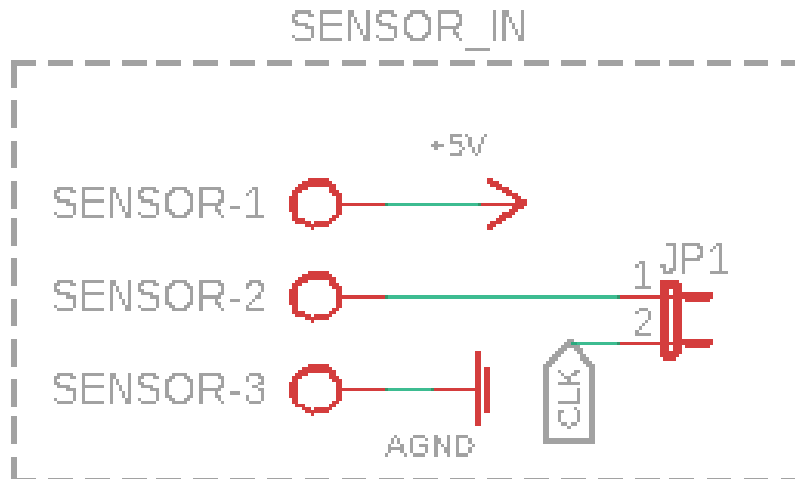


Figura 7: Etapa conexión sensores de velocidad y temperatura.

3.2.4 Etapa de comunicación

La etapa de comunicación está conformada por un dispositivo de comunicación LoRa (acrónimo de Long Range) E49 ver Figura 8, el cual es un tipo de tecnología inalámbrica de largo alcance, baja potencia, bajo costo, fácil configuración, con alta inmunidad al ruido que es utilizada para la transmisión de datos en redes de sensores y otros dispositivos de IoT. La tecnología LoRa se basa en una modulación de espectro ensanchado y utiliza frecuencias de radio no licenciadas para permitir la comunicación a través de largas distancias con un bajo consumo de energía.

Los dispositivos de comunicación LoRa se utilizan comúnmente en aplicaciones de monitoreo ambiental, sistemas de control de la calidad del aire, medición de la humedad del suelo, sistemas de seguridad y alarmas, entre otros. La tecnología LoRa es especialmente útil para aplicaciones que requieren una larga duración de la batería y una conectividad fiable a largas distancias.

El dispositivo LoRa E49 implementado en esta aplicación cuenta con un puerto de comunicaciones uart (rs232 ttl) que permite la configuración de los parámetros y el envío de tramas a un dispositivo específico. En línea de vista el fabricante asegura una distancia de transmisión de 2,4 km. La estructura de la trama considera los siguientes campos:

- ✓ Canal: Identifica la frecuencia en la que el dispositivo se va a identificar.
- ✓ Identificador de dispositivo: Asocia el identificador que el dispositivo tiene dentro de la red.
- ✓ Datos: en total se pueden enviar hasta 54 bytes de forma continua, o en caso de ser enviado un menor número de estos, el sistema cuenta con un tiempo de espera de forma que los datos sean enviados de forma automática.

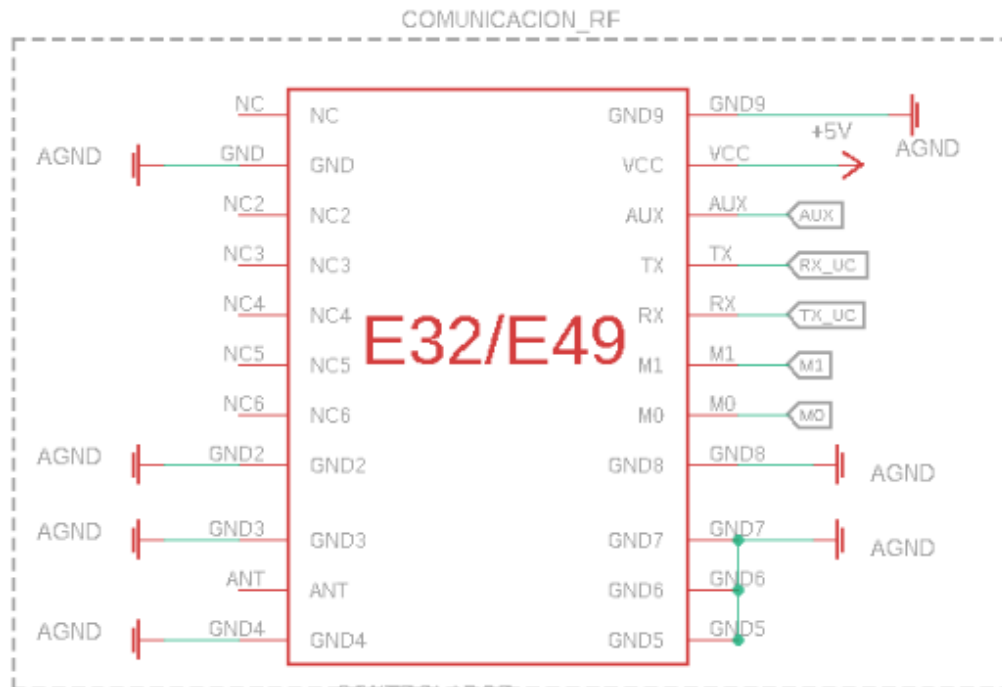


Figura 8: Etapa de comunicación.

3.2.5 Etapa de notificación de estados

Esta etapa está conformada por dos indicadores led, como se observa en la Figura 9, de los cuales uno indica que la tarjeta está conectada a la red eléctrica, mientras que el otro permite verificar el estado de conexión de la comunicación.

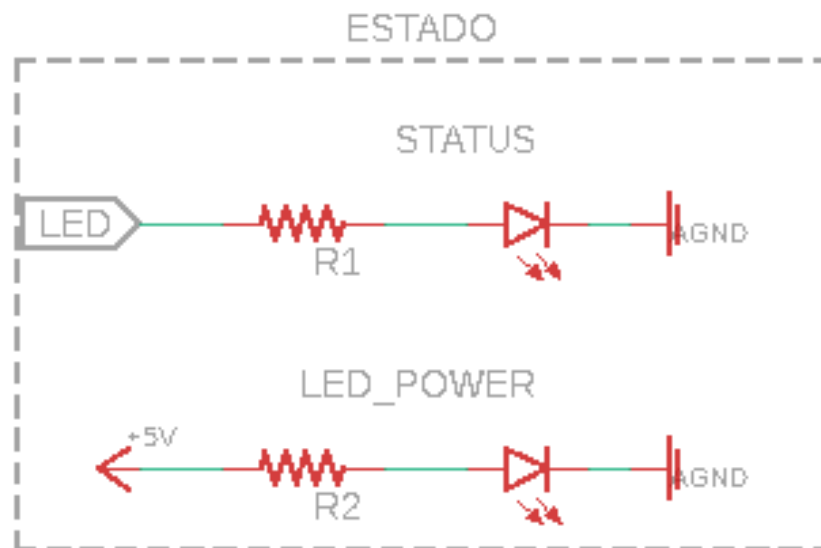


Figura 9: Etapa de notificación de estados.

3.2.6 Etapa de control

Para la etapa de control se cuenta con un microcontrolador pic 16f1827, Figura 10, el cual puede ser programado mediante un puerto ICSP (In Circuit Serial Programming). Adicionalmente, se incluyó un Dip Swicht que permite gestionar la dirección del dispositivo de modo que, al ser polarizada la tarjeta, el microcontrolador se encarga de configurar el módulo LoRa con este parámetro. Otra de las tareas del controlador consiste en la digitalización de la información proporcionada por los sensores, de modo que pueda ser enviada de forma inalámbrica hasta el dispositivo maestro o servidor local.

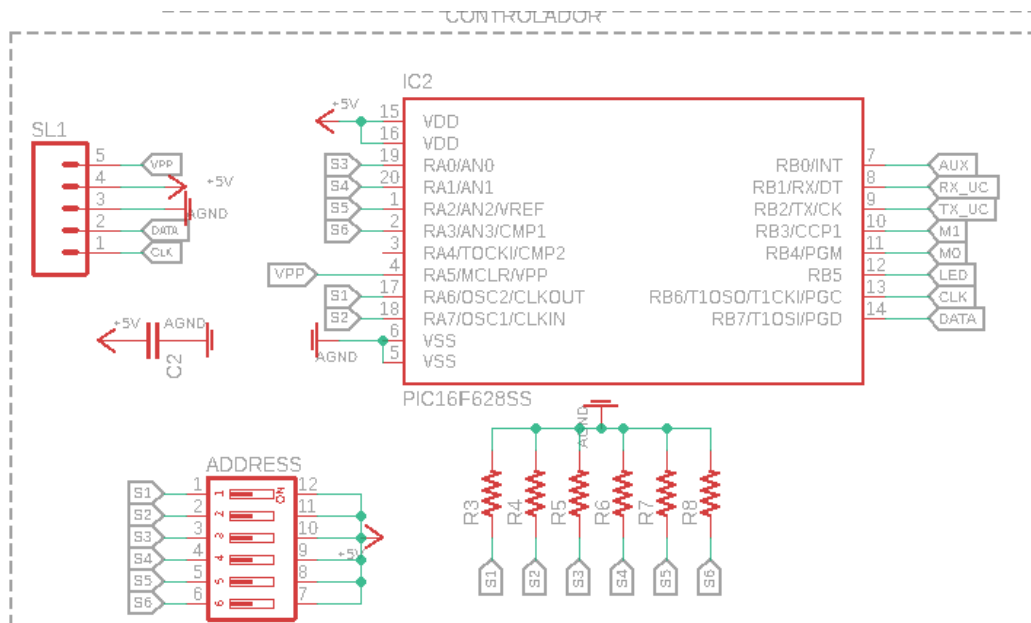


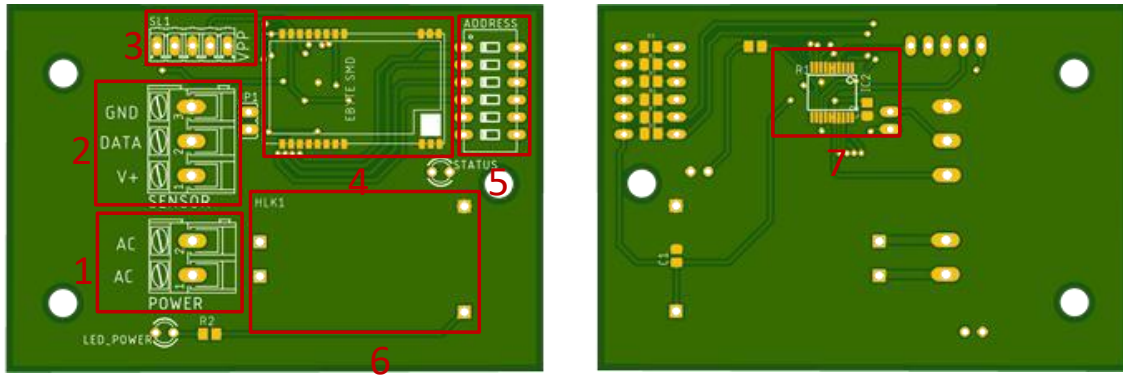
Figura 10: Etapa de control.

3.2.7 Tarjeta de circuito impreso

Se realizó el diseño de una tarjeta de circuito impreso (PCB, por sus siglas en inglés) para la implementación física del circuito que permite la toma de los datos de velocidad angular y temperatura de los bobinados del estator.

Para el diseño se consideraron las limitaciones de espacio, de manera que el circuito obtenido es de dimensiones reducidas lo que facilita su instalación y posterior uso.

Esta tarjeta debe ser configurada para uno de los dos fines, medición de temperatura o de velocidad, ya que no es posible medir las dos variables de manera simultánea con una única tarjeta. En la Figura 11 se presenta la tarjeta diseñada.



(a)

(b)

Figura 11: Tarjeta diseñada para medición de velocidad y temperatura. (a), vista superior y (b), vista inferior.

Las componentes de la tarjeta de medición de temperatura y velocidad que se muestran en la Figura 11, encerradas con rectángulos de color rojo, se listan a continuación:

1. Borneras de alimentación en corriente alterna para el suministro de energía del sistema;
2. Bornera de conexión de los sistemas de toma de datos, puede ser un encoder para la medición de velocidad angular del rotor o un sensor de temperatura tipo sonda, para la medición de temperatura en las bobinas del estator;
3. Bornera de conexión para realizar la programación del microcontrolador que controla la tarjeta;
4. Dispositivo de comunicación LoRa, el cual permite realizar la comunicación con el servidor local, y, por ende, el envío de la información capturada;
5. DIP Switch, el cual permite realizar la asignación de la dirección a cada una de las tarjetas que componen el sistema general;
6. Fuente AC/DC para alimentación de los componentes, y
7. El microcontrolador PIC16F1827, que es el procesador o cerebro de la tarjeta.

La PCB fabricada y ensamblada se muestra en la Figura 12.



Figura 12: Tarjeta diseñada para medición de velocidad y temperatura. (a), vista superior y (b), vista inferior.

3.2.8 Descripción de *firmware* de las tarjetas

Para el desarrollo del *firmware* para el microcontrolador PIC16F1827, que controla cada una de las tarjetas de adquisición de velocidad y temperatura, se utilizó el software de programación de microcontroladores llamado PIC C Compiler, el cual cuenta con una gran gama de dispositivos PIC compatibles con dicho Software. El lenguaje de programación empleado es C y la herramienta para la carga del *firmware* en el microcontrolador es el PICKIT3.

A continuación, en la Figura 13, se presenta el diagrama de bloques que hace referencia al funcionamiento de cada uno de los códigos de programación de las tarjetas de adquisición de datos implementadas para la medición de velocidad angular del rotor del motor asíncrono y la temperatura en los bobinados estáticos de este.

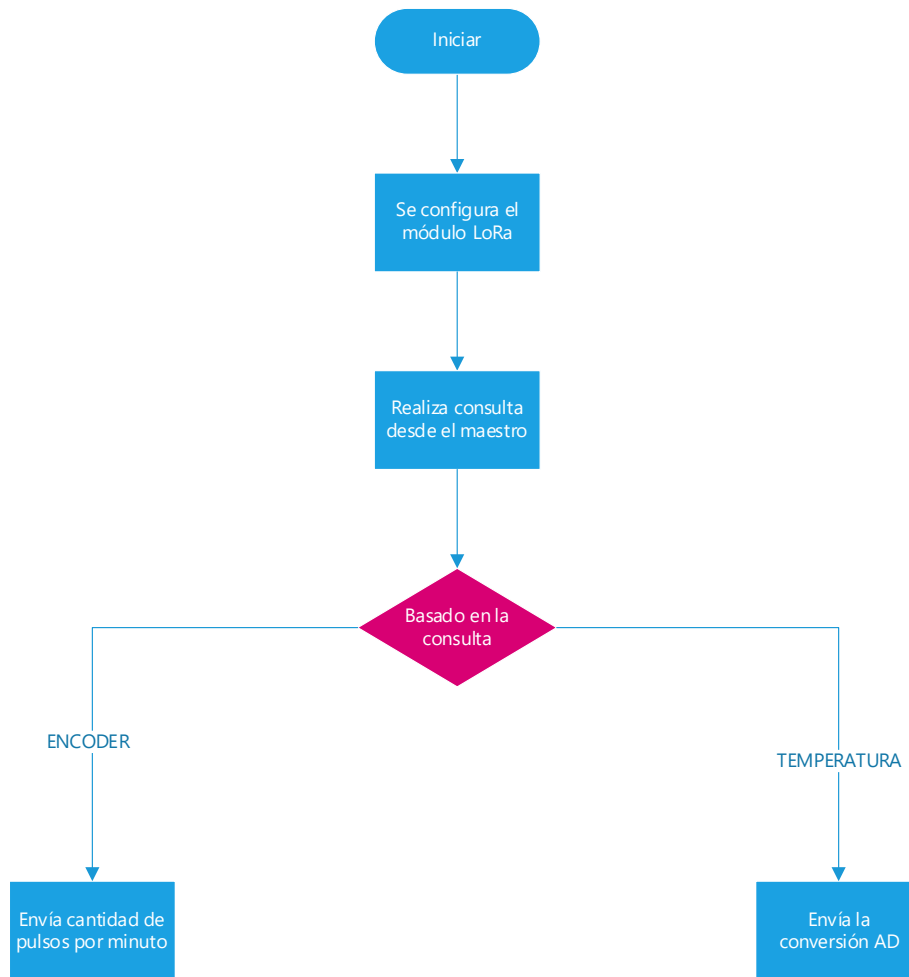


Figura 13: Diagrama de bloques del firmware de las tarjetas

En los anexos **8.1** y **8.2** se detalla el código de programación de estos dispositivos.

3.2.9 Medición de variables eléctricas

Para realizar la medición de las variables eléctricas en las líneas de alimentación del motor de inducción, tales como corrientes, THD y factor de potencia, se hace uso de un medidor de energía multifuncional de tablero EasyLogic PM2120 de Schneider Electric. Estos dispositivos cumplen con los estándares de precisión de Clase 1 o Clase 0.5S, y pueden ser utilizados como unidad independiente o como esclavo, lo que permite una gran versatilidad y optimización de sus funciones.

Este equipo permite establecer comunicación serial a través de un puerto de comunicación modbus RS-485 y se pueden conectar hasta 32 dispositivos a un único bus RS-485. De este modo se realiza la conexión con el servidor local y se extraen las tramas con la información de las variables de interés, las cuales posteriormente son enviadas al servidor en la nube desde el cual se gestionan para

su publicación a través de la interfaz de usuario y posterior almacenamiento en la respectiva base de datos.

En la Figura 14 se muestra una imagen del panel frontal del medidor multifuncional.



Figura 14: Medidor multifuncional, vista frontal [37].

3.2.10 Etapa de control del motor de inducción

Para realizar el control remoto del motor de inducción, se usó el protocolo Modbus, mediante el cual se le indica a un variador de velocidad cual será la velocidad, dirección del sentido de giro arranques y paradas del motor. Por otro lado, debido a la flexibilidad que presenta el sistema implementado, este control también puede realizarse haciendo uso de un Controlador Lógico Programable (PLC- por sus siglas en inglés) o arrancadores suaves con conexión modbus, debido a que Node-red cuenta con librerías que permiten realizar dicha conexión y hacer la modificación de las variables internas de los mismos.

En la Figura 15, se muestra el diagrama de bloques que representa la interacción entre el usuario, a través de la interfaz con el servidor local, y el motor. Esto le permite al usuario realizar el cambio de velocidad, arranque y parada del motor de inducción.



Figura 15: Etapa de control del motor.

Capítulo 4: Servidor local

El servidor local está implementado en una tarjeta electrónica SBC Raspberry Pi 4 modelo B, creada en Reino Unido por la Fundación Raspberry PI, que funciona como control maestro y se encarga de la recopilación y gestión de los datos entregados por cada una de las tarjetas, equipos y dispositivos de adquisición de datos. Los datos obtenidos son enviados posteriormente por WiFi, a un servidor en la nube, donde podrán ser observados y analizados por el usuario o el personal encargado.

Para implementar el servidor local, se seleccionó una computadora de placa reducida SBC (Single Board Computer- Por sus siglas en inglés), que se encuentra en una sola placa de circuito impreso, luego de haber analizado sus prestaciones y características. A diferencia de las computadoras de escritorio o portátiles, que están compuestas por varias partes conectadas entre sí, las SBC integran todos los componentes esenciales en una sola placa. Esto incluye el procesador, memoria RAM, almacenamiento de datos, puertos de entrada y salida, y otros componentes como adaptadores de red, controladores de pantalla, etc.

Los computadores de placa reducida son generalmente más compactos, livianos y confiables que los computadores de múltiples tarjetas, debido a que integran una mayor cantidad de componentes y conectores en una sola placa. Además, estos computadores suelen ser más eficientes en el manejo de la energía eléctrica, gracias a su diseño de bajo consumo y se caracterizan por sus dimensiones reducidas y su bajo costo.



Figura 16: Raspberry Pi 4 modelo B [38].

La Raspberry Pi 4 modelo B contiene un procesador de cuatro núcleos con arquitectura de 64bits, frecuencia de 1,5 GHz, con conectividad por medio de

Ethernet, Wifi y Bluetooth, y se hace ideal para la implementación del servidor local debido a su bajo consumo de energía, tamaño reducido, facilidad de instalación y su bajo costo.

4.1 Descripción del software

4.1.1 Descripción de *firmware* servidor local

Para la implementación del firmware del servidor local, se usó la herramienta Node-red, el cual es una plataforma libre de fácil implementación que integra varios protocolos de comunicación multi plataforma. Este permite realizar la programación haciendo uso de una serie de nodos o bloques, los cuales son configurados con base en la actividad que se desea implementar. La configuración se divide en tres etapas, las cuales son descritas a continuación.

Primera etapa

Es la etapa encargada del sincronismo y de enviar la trama de consulta a los módulos inalámbricos. Para esto se hace uso de un nodo **inject** que envía datos periódicos a una función personalizada en el nodo **function**, que se encarga de hacer el envío de las tramas de consulta a los dispositivos inalámbricos, que, para este caso, son las tarjetas de adquisición de temperatura en el estator y de velocidad del motor de inducción.

En la Figura 17 se muestran los bloques de Node-red que conforman la primera etapa.

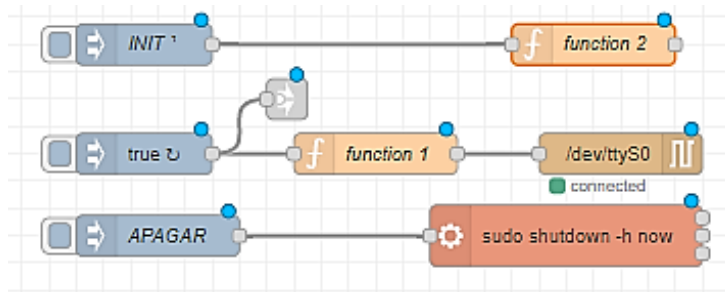


Figura 17: Etapa encargada del sincronismo y de enviar mensajes de consulta

Segunda etapa

Una vez implementada la primera etapa, entra en funcionamiento una segunda etapa que se encarga de hacer la decodificación de la trama enviada por cada uno de los esclavos o tarjetas de adquisición de datos, para luego independizar las variables medidas, que en este caso son la velocidad y temperatura. Para esto se hace uso de un nodo **serial**, que se encarga de comunicarse con el módulo LoRa y el nodo **function**, que se encarga de independizar los datos para cada una de las variables, de igual forma se hace uso del nodo **Smooth**, el cual es una herramienta

que se utiliza para suavizar los datos de entrada a través de un algoritmo de promediado móvil. Este se caracteriza por ser especialmente útil para reducir el ruido o las fluctuaciones en los datos de sensores o dispositivos que pueden afectar el rendimiento del sistema o la precisión de las mediciones, además permite que se pueda configurar para ajustar la cantidad de suavizado que se aplica a los datos de entrada. Esto permite a los usuarios adaptar el bloque a diferentes tipos de datos y aplicaciones, como en este caso, los datos que ingresan de la tarjeta de adquisición de temperatura [39].

En la Figura 18 se muestran los bloques que conforman la etapa dos.

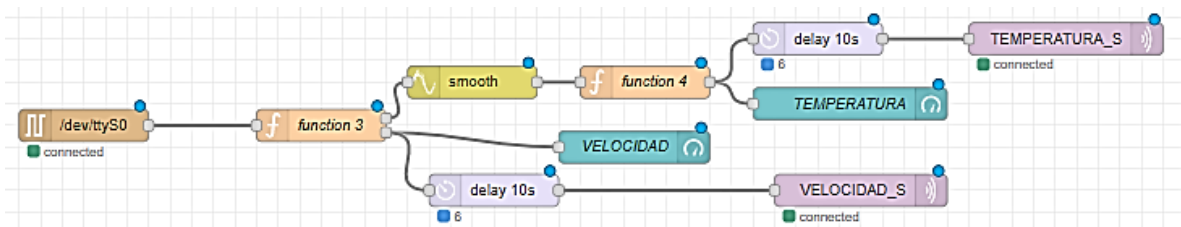


Figura 18: Etapa encargada de la decodificación de tramas.

Tercera etapa

Esta etapa es la encargada de realizar las consultas Modbus al medidor de energía multifuncional de tablero EasyLogic PM2120, accediendo a los diferentes registros internos. El periodo de acceso a los registros está determinado por el mismo periodo utilizado para realizar las consultas inalámbricas de las tramas a las tarjetas de adquisición de datos de velocidad y temperatura. En esta etapa se utiliza el nodo **Modbus** de Node-red, el cual es una herramienta utilizada para la comunicación entre dispositivos a través del protocolo Modbus. Modbus es un protocolo de comunicación ampliamente utilizado en la industria para la transmisión de datos entre dispositivos electrónicos, como controladores lógicos programables (PLC), sensores y actuadores. Este permite a los usuarios leer y escribir datos en dispositivos que utilizan el protocolo Modbus, puede ser configurado para conectarse a dispositivos en una red local o remota a través de una variedad de interfaces, incluyendo Ethernet, RS-485, RS-232 y TCP/IP.

Una vez se logra establecer la correcta conexión con un dispositivo vía Modbus, el bloque **Modbus** de Node-red puede leer y escribir datos utilizando los registros de entrada y salida del dispositivo como lo es en este caso el medidor multifuncional. Los registros de entrada se utilizan para leer datos del dispositivo, mientras que los registros de salida se utilizan para escribir datos en el dispositivo, logrando de esta forma obtener las tramas con la información de las variables adquiridas [40].

En la Figura 19 se muestran los bloques que conforman la etapa tres.

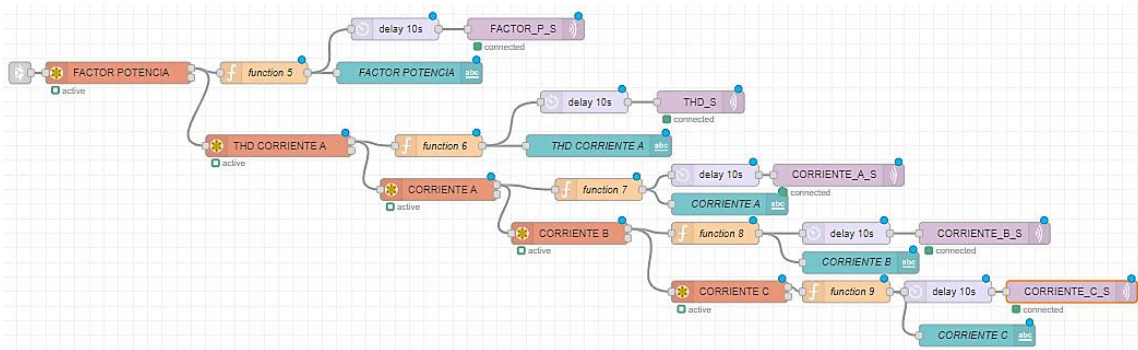


Figura 19: Etapa es la encargada de realizar las consultas modbus al medidor de energía multifuncional.

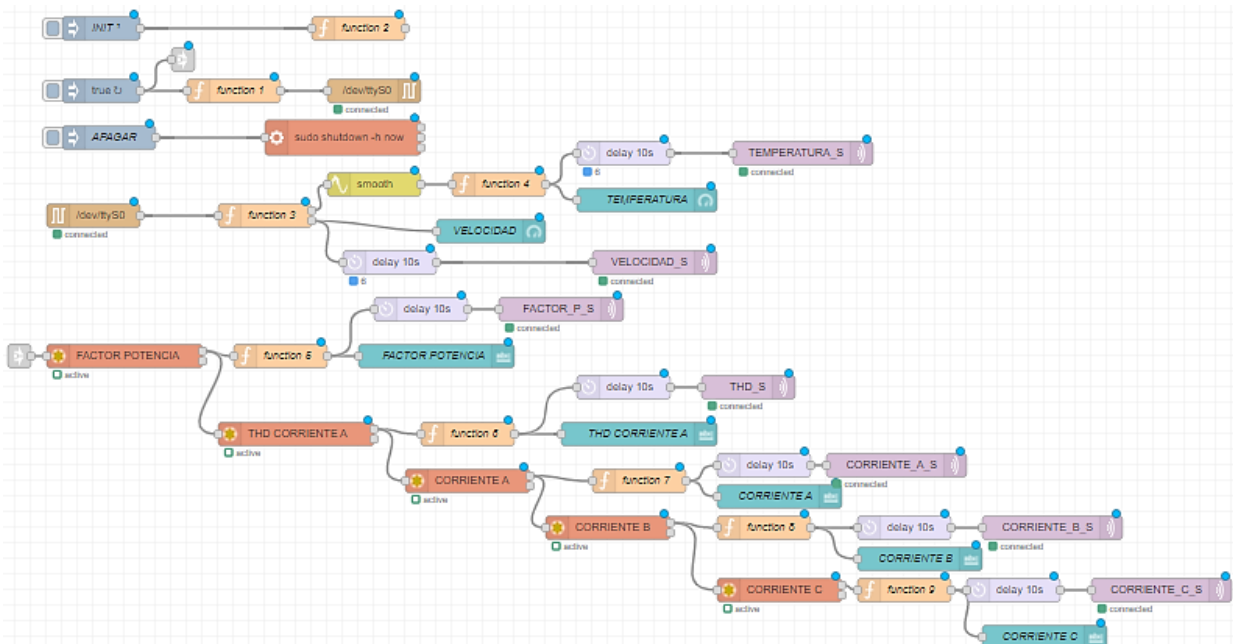


Figura 20: firmware servidor local

Finalmente, en la Figura 20, se muestra la estructura completa de bloques que conforman el firmware del servidor local.

4.1.2 Almacenamiento en la nube

Una vez capturados los datos de las variables de interés como corrientes en la línea de alimentación del motor, THD, factor de potencia, velocidad y temperatura, es necesario almacenar dicha información. En la Figura 21 se muestran los tópicos que hacen referencia a los datos adquiridos, los cuales son recibidos por medio del Nodo

mqtt in de Node-red, que es el encargado de conectarse a un agente MQTT, posteriormente se suscribe a mensajes del tema especificado, que para este caso son las variables de interés.



Figura 21: Tópicos almacenados.

Los nodos **function** que son los permiten escribir funciones personalizadas en JavaScript y ejecutarlas en el flujo de Node-red.

Para usar el nodo "Function", se arrastra y se suelta desde la paleta de nodos en la barra lateral izquierda a la zona de trabajo. Al hacer doble clic en el nodo, se abrirá un editor de código que permitirá escribir una función personalizada en JavaScript. La función debe aceptar una entrada (payload) y puede generar una salida, que se enviará al siguiente nodo en el flujo.

La función personalizada puede realizar cualquier tarea que se pueda programar en JavaScript, como procesar datos, realizar cálculos, conectarse a una API, etc. Por lo que fue necesario implementar una función personalizada en cada nodo **function** por cada una de las variables medidas como velocidad, temperatura factor de potencia etc, en la Figura 22 se muestran los bloques **function** creados para cada una de estas.

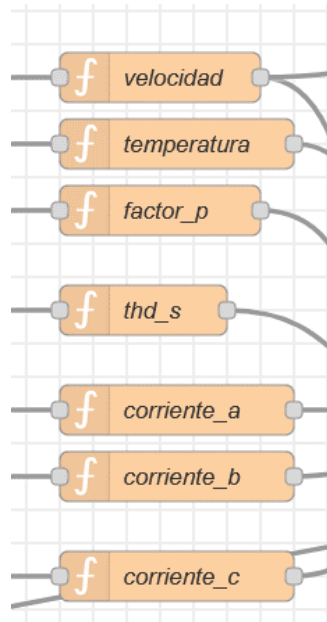


Figura 22: Nodos de funciones personalizadas para cada variable.

Posteriormente, los datos de cada una de las variables entregados a la salida de los respectivos nodos de funciones personalizadas son ingresados en nodos tipo **switch**, los cuales se encargan de tomar los valores medidos y enrutarlos en función de sus valores de propiedad, es decir es en este punto donde se definen los límites de los valores de funcionamiento normal en cada una de las variables monitoreadas, por ejemplo en el caso de la velocidad, se define un valor máximo de velocidad en RPM, el cual una vez superado dará origen a una alarma, de igual forma se establecen los valores límite para temperatura, factor de potencia, THD y corrientes de alimentación. En la Figura 23 se muestran los diferentes bloques que hacen referencia a cada una de las variables monitoreadas.

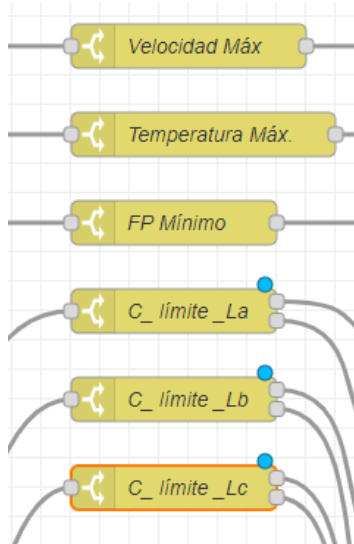


Figura 23: Nodos switch encargados de enrutar los datos de cada variable.

Una vez se superan los límites definidos en el nodo **switch**, esta información se hace llegar al nodo **exec**, que se encarga de ejecutar el comando del sistema en el servidor en la nube, lo que permite generar la alarma que para este caso es un mensaje de texto enviado al destinatario registrado en el **script** de envío de mensajes, permitiendo de esta forma que el usuario esté enterado del funcionamiento anormal o de las alarmas que se están generando. Cada una de las variables monitoreadas, cuenta con un nodo **exec**, como se ve en la Figura 24.



Figura 24: Nodos **exec** encargados de ejecutar el envío de mensajes.

De la salida de los bloques **function** de cada una de las variables monitoreadas, se toma dicha información y se ingresa en el bloque **change** que se encarga de crear una variable de flujo con la información de los datos adquiridos, este flujo de datos es ingresado en un bloque **join**, que se encarga de convertir el flujo de mensajes que le es ingresado en un vector. En el bloque **function** que fue renombrado con la etiqueta de nombre **lista**, se realiza la asignación de la posición de cada una de las variables en el vector, esta información es ingresada en un nodo **function** y posteriormente enviada al nodo **MySQL** que es una herramienta utilizada para conectarse a bases de datos y realizar operaciones en ellas. Este bloque permite al usuario leer y escribir datos en una base de datos **MySQL** desde un flujo de trabajo de Node-red. Puede configurarse para conectarse a la base de datos en una red local o remota, como es el caso de este proyecto. Una vez que se ha establecido una conexión por medio de una IP fija por el puerto 1880 ingresando usuario y contraseña, los usuarios pueden realizar una variedad de operaciones, incluyendo la lectura y escritura de datos, la ejecución de consultas SQL y la creación o eliminación de tablas.

En la Figura 25 se presenta la configuración del servidor para el almacenamiento en la nube, donde se pueden evidenciar cada uno de los bloques utilizados para realizar dicha implementación.

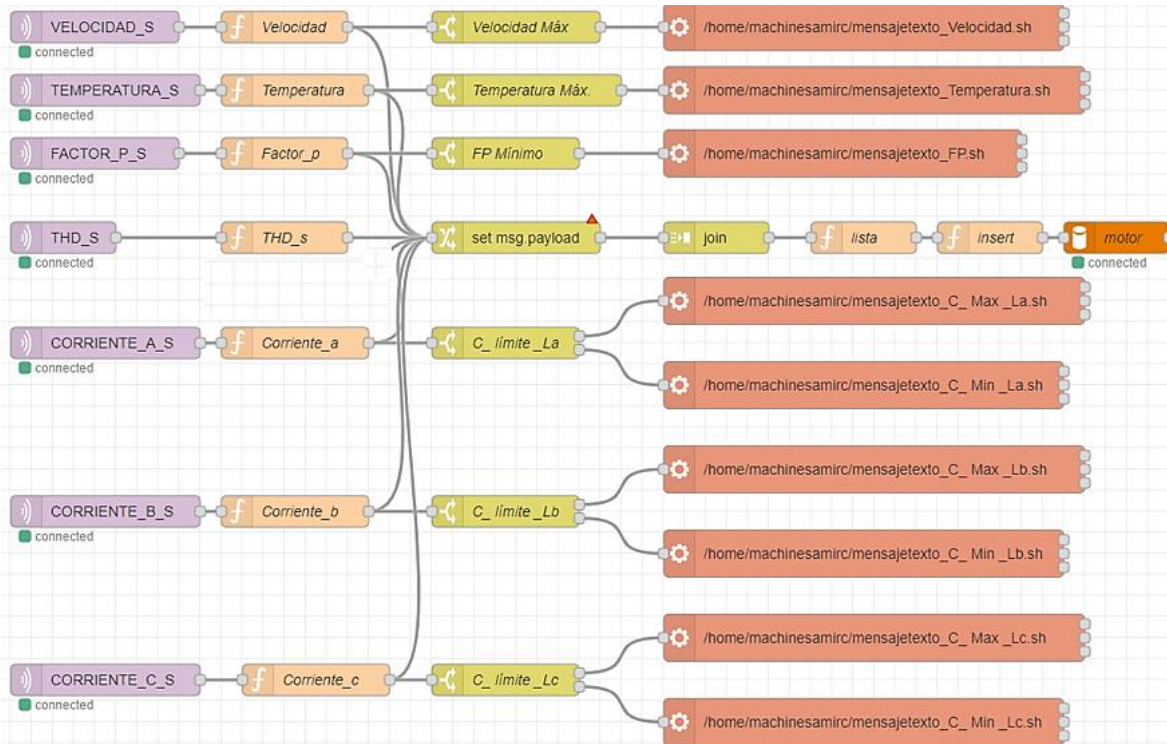


Figura 25: Nodos para la configuración del servidor en la nube.

4.1.3 Seguridad SSL Node-red

Se realizó la configuración de Node-red con certificado (Secure Sockets Layer) SSL, que se aplica para proteger la información transmitida entre un usuario y un servidor en Internet. SSL es una tecnología de cifrado que asegura que los datos compartidos entre el navegador del usuario y el servidor web estén protegidos contra la interceptación, el espionaje y la manipulación por parte de terceros no autorizados. Para garantizar la seguridad de la información que es transmitida desde el servidor local hasta el servidor externo o nube, con la ayuda del WAF de Cloudflare se obtuvo un certificado SSL y un dominio <https://samircorrea.optimalsoft.com.co/>. A continuación, se detallan algunas de las razones clave por las que se aplica la seguridad SSL:

- ✓ **Confidencialidad de los Datos:** SSL cifra los datos transmitidos entre el navegador del usuario y el servidor web, lo que significa que, si alguien intercepta la comunicación, no podrá entender ni acceder a la información confidencial. Esto es crucial para proteger datos sensibles como contraseñas, números de tarjetas de crédito y otra información personal.

- ✓ **Integridad de los Datos:** SSL garantiza que los datos transmitidos no sean modificados ni alterados durante la transferencia. Si alguien intenta manipular los datos en tránsito, la conexión SSL se romperá, lo que alertará al usuario y al servidor de que la comunicación no es segura.
- ✓ **Autenticación del Servidor:** SSL permite la autenticación del servidor, lo que significa que el navegador del usuario puede verificar que está conectado al sitio web correcto y no a un sitio web falso o malicioso. Esto ayuda a prevenir ataques de intermediario (hombre en el medio) y phishing.
- ✓ **Protección contra Ataques:** SSL ayuda a proteger contra ataques como la suplantación de identidad, la interceptación de información y la inyección de contenido malicioso en la comunicación entre el navegador y el servidor.

A continuación, se muestran las líneas de código para la instalaciones y configuraciones de software necesarias para el desarrollo del proyecto.

//se editan líneas de configuración Node-red para activar el certificado SSL
nano /usr/lib/node_modules/Node-red/node_modules/Node-red-admin/settings.js

//donde los archivos .key y .pem son los que se genera en cloudflare
https: {
key: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft.key'),
cert: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft.pem')
},

//Asegurar logueo de Node-red

nano /root/.Node-red/settings.js

*/** To password protect the Node-red editor and admin API, the following*
** property can be used. See <http://nodered.org/docs/security.html> for details.*
**/*
adminAuth: {
type: "credentials",
users: [{
username: "admin",
password: "\$2a\$08ssdf34Z6dNbM6tl8sJogENOMcxWV9DN.",
*permissions: "**"*
}]
},

//Configuración del servidor

-Firewall en servidor

apt-get install ufw -y
sudo ufw status numbered

//puertos mariadb


```
sudo ufw allow out 3346/tcp
sudo ufw allow in 3346/tcp
```

//habilitacion de puertos Node-red

```
sudo ufw allow 1880
sudo ufw allow out 1880/tcp
sudo ufw allow in 1880/tcp
```

-Configurar zona horaria

```
dpkg-reconfigure tzdata
```



Figura 26: Configuración de la zona horaria.

-instalacion java

```
sudo apt install openjdk-17-jre openjdk-17-jdk
```

-instalacion mariadb

```
sudo apt -y install mariadb-server mariadb-client
```

```
sudo mysql_secure_installation
```

//creacion de usuario

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'claveusuario' WITH GRANT OPTION;
```

permisos de acceso puertos mariadb

```
sudo ufw allow out 3306/tcp
sudo ufw allow in 3306/tcp
```

-instalacion de node red

```
sudo npm install -g --unsafe-perm Node-red
```

4.1.4 Motor de base de datos

Se adquirió un servidor VPS con el proveedor de servicios web Contabo, en este servidor se instalaron las tecnologías necesarias para desarrollar el proyecto. Esta es una plataforma con sede en Alemania, que provee diferentes tipos de alojamiento web incluyendo hosting compartido, servidores VPS (Virtual Private Servers), servidores dedicados y servicios de alojamiento en la nube. También proporcionan servicios de registro de dominios y servidores de juegos. Debido a estas características se optó por el uso de dicho servidor, para realizar la implementación del motor de bases de datos con **MariaDB server**, el cual es un sistema de gestión de bases de datos que cuenta con características importantes como las siguientes:

- ✓ **Compatibilidad con MySQL:** MariaDB se creó como un fork de MySQL y mantiene una alta compatibilidad con MySQL. Esto significa que las aplicaciones y las consultas SQL diseñadas para MySQL generalmente se ejecutarán sin problemas en MariaDB, lo que facilita la migración.
- ✓ **Código Abierto y Gratuito:** MariaDB es un sistema de gestión de bases de datos de código abierto bajo la Licencia Pública General de GNU (GPL). Esto significa que puedes usarlo sin costo y tienes acceso al código fuente para personalizarlo según tus necesidades.
- ✓ **Comunidad Activa y Soporte Empresarial:** A pesar de ser de código abierto, MariaDB cuenta con una comunidad activa de desarrolladores y usuarios que brindan soporte y contribuyen al desarrollo continuo del sistema. Además, existen empresas que ofrecen servicios de soporte y consultoría para MariaDB, lo que puede ser útil en entornos empresariales.
- ✓ **Rendimiento Optimizado:** MariaDB se ha centrado en mejorar el rendimiento en comparación con MySQL. Esto incluye optimizaciones de consultas, mejoras en la gestión de índices y la capacidad de utilizar almacenamiento en caché de consultas en memoria para acelerar las operaciones de lectura.
- ✓ **Almacenamiento de Datos Avanzado:** MariaDB admite una variedad de motores de almacenamiento, incluidos InnoDB (por defecto), Aria, TokuDB y otros. Esto te permite elegir el motor que mejor se adapte a tus necesidades específicas, desde transacciones ACID hasta alta velocidad de lectura/escritura.
- ✓ **Escalabilidad:** MariaDB es escalable y puede gestionar grandes volúmenes de datos y cargas de trabajo. Puedes implementar particiones, replicación y clústeres para escalar horizontal y verticalmente según sea necesario.
- ✓ **Ecosistema de Aplicaciones y Herramientas:** MariaDB es compatible con una amplia gama de aplicaciones, marcos y herramientas de desarrollo, lo que facilita la integración en tu infraestructura existente.

Todo esto le permite estar entre los servidores de bases de datos más populares del mundo.

Una vez instalado **MariaDB server**, se crea una base de datos llamada **motor** y en esta una tabla llamada **datos**, en la cual se almacenará la información de las variables adquiridas por los diferentes sensores por medio del programa Node-red. En la Figura 27 se muestra la interfaz gráfica del **MariaDB server**. Una vez almacenadas en la base de datos, estas son consultadas y mostradas haciendo uso de la UI de Node-red.

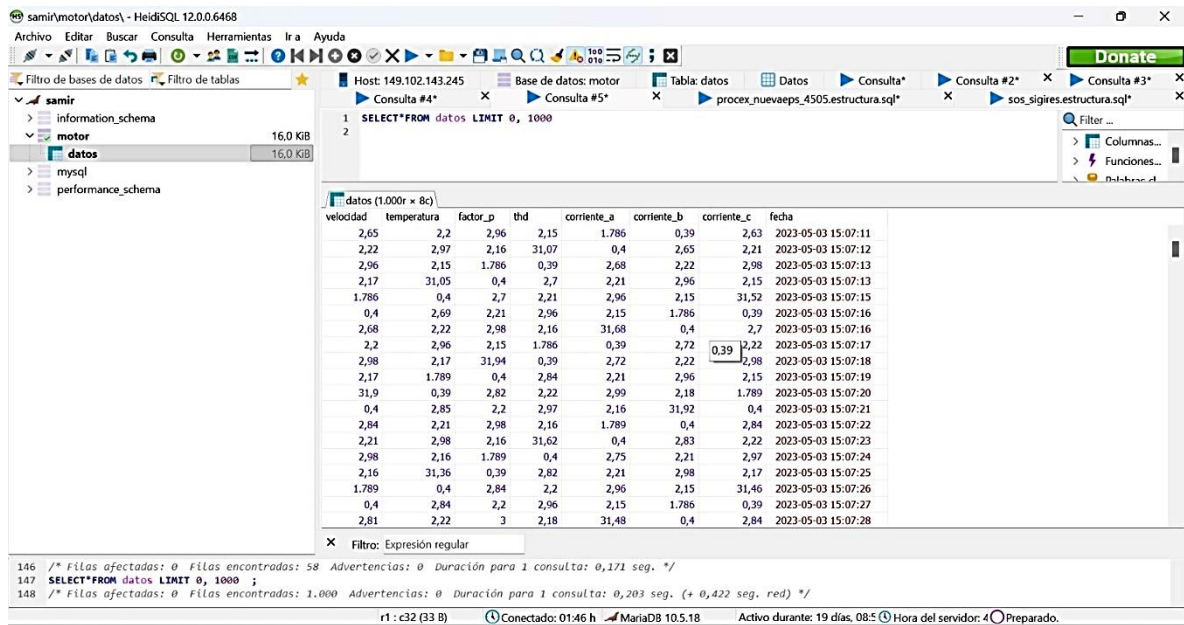


Figura 27: interfaz gráfica del MariaDB server.

4.1.5 Interfaz de usuario

Para implementar la interfaz gráfica de usuario, se usó Node-red Dashboard, que es una herramienta de visualización web que se integra con la plataforma de programación visual Node-red. Los nodos que se utilizaron para la implementación de dicha interfaz fueron, nodo **Gauge**, el cual es una herramienta que se utiliza para crear medidores visuales en un flujo de trabajo. Esta toma los datos que llegan del servidor local vía Wifi permitiendo que sean representados de forma visual. Este bloque es especialmente útil para representar datos de entrada de una manera intuitiva y visualmente atractiva. Se puede configurar para mostrar diferentes tipos de medidores, como medidores de velocidad, de temperatura, de presión, entre otros. Los usuarios pueden ajustar los rangos de los valores de entrada y definir los umbrales para los distintos rangos [41].

Además, se usó el nodo **Chart** para crear gráficos en tiempo real a partir de los mismos datos de entrada, permitiendo al usuario visualizar información compleja de una manera fácil de entender y personalizable. Puede ser configurado para mostrar diferentes tipos de gráficos, como gráficos de líneas, barras, pastel, entre otros. Los

usuarios pueden ajustar los ejes x e y, personalizar la apariencia del gráfico y definir umbrales para los distintos valores de entrada.

Una característica útil del bloque **Chart** de Node-red es su capacidad para actualizar los gráficos en tiempo real, lo que permite a los usuarios monitorear los cambios en los datos de entrada a medida que ocurren. Esto es especialmente útil en situaciones en las que se requiere un monitoreo constante de los datos, como en la monitorización de la temperatura o la presión en una planta industrial [42].

Para realizar el control de velocidad del motor de inducción se utiliza el nodo **Slider** de Node-red el cual permite crear controles deslizantes de entrada en un flujo de trabajo. Este bloque es especialmente útil para ajustar valores de entrada numéricos de manera precisa y eficiente. Se puede configurar para definir un rango de valores, un paso para el incremento o decremento de los valores y la precisión de los valores. Además, los usuarios pueden personalizar la apariencia del control deslizante, como el color y el tamaño [43].

Finalmente se utiliza el nodo **Table** de Node-red que es una herramienta que permite mostrar y manipular datos en forma de tabla en un flujo de trabajo. Este bloque es especialmente útil cuando se requieren visualizar grandes cantidades de datos y realizar operaciones en ellos de manera eficiente, como es el caso de los datos que fueron medidos y enviados desde el servidor local.

Este permite a los usuarios configurar las columnas que se muestran en la tabla, personalizar su apariencia y definir las opciones de ordenamiento y filtrado. Una característica útil del bloque **Table** es su capacidad para trabajar con diferentes fuentes de datos, como bases de datos, archivos CSV o datos en formato JSON. Esto permite importar datos desde diferentes fuentes y manipularlos en la tabla de manera uniforme [44].

En la Figura 28 se muestra la configuración del Dashboard o interfaz de usuario, cuya visualización definitiva se presenta en la Figura 29, donde se pueden observar cada una de las representaciones gráficas de los datos que están siendo medidos por las tarjetas de adquisición y el medidor multifuncional, de igual forma se presenta el control de marcha del motor de inducción y la tabla que está en constante almacenamiento de los datos que son guardados en la base de datos del servidor en la nube, permitiendo así que los usuarios puedan tener acceso de forma visual y a la vez del histórico de la información recolectada.

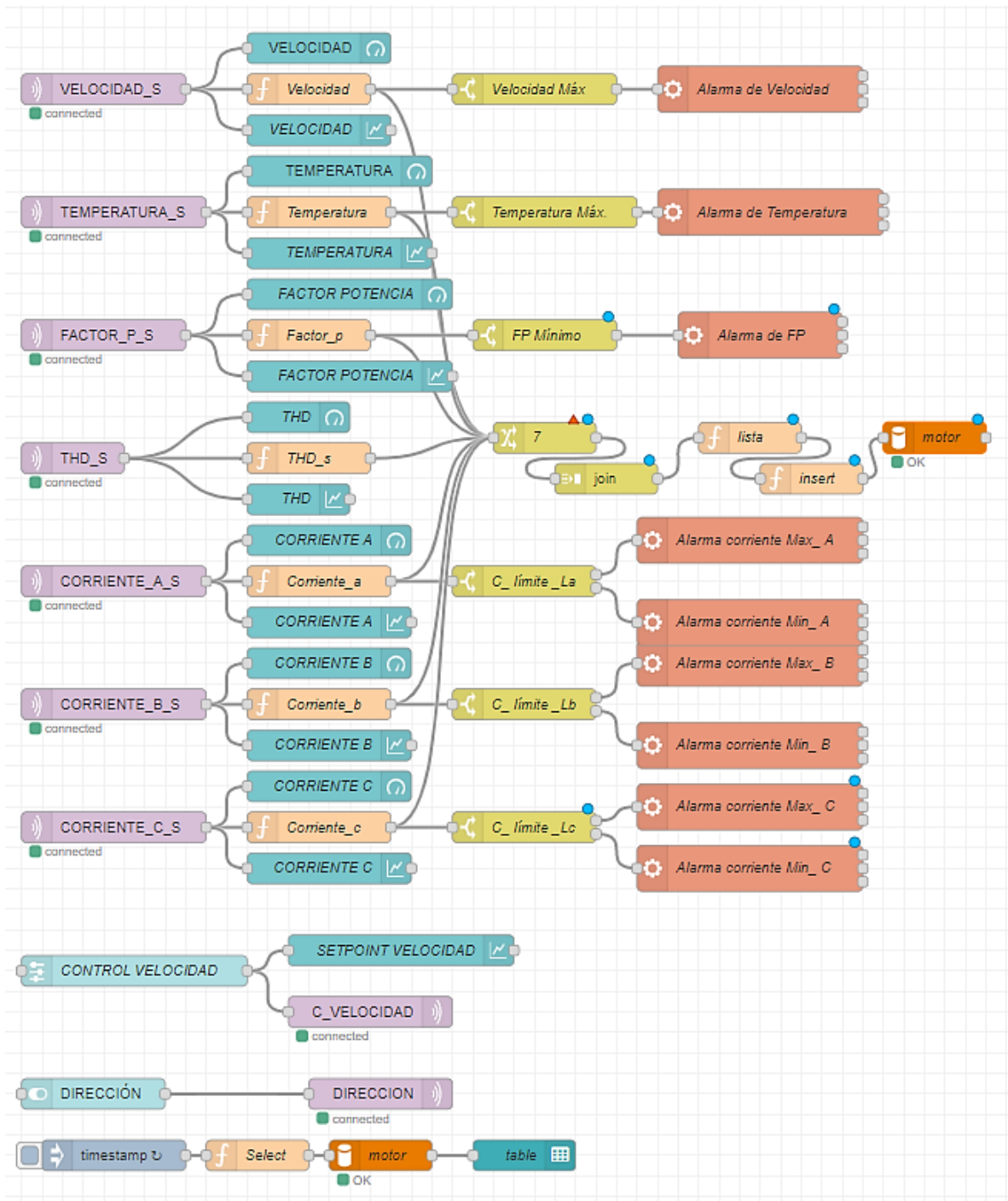


Figura 28: configuración del dashboard o interfaz de usuario.

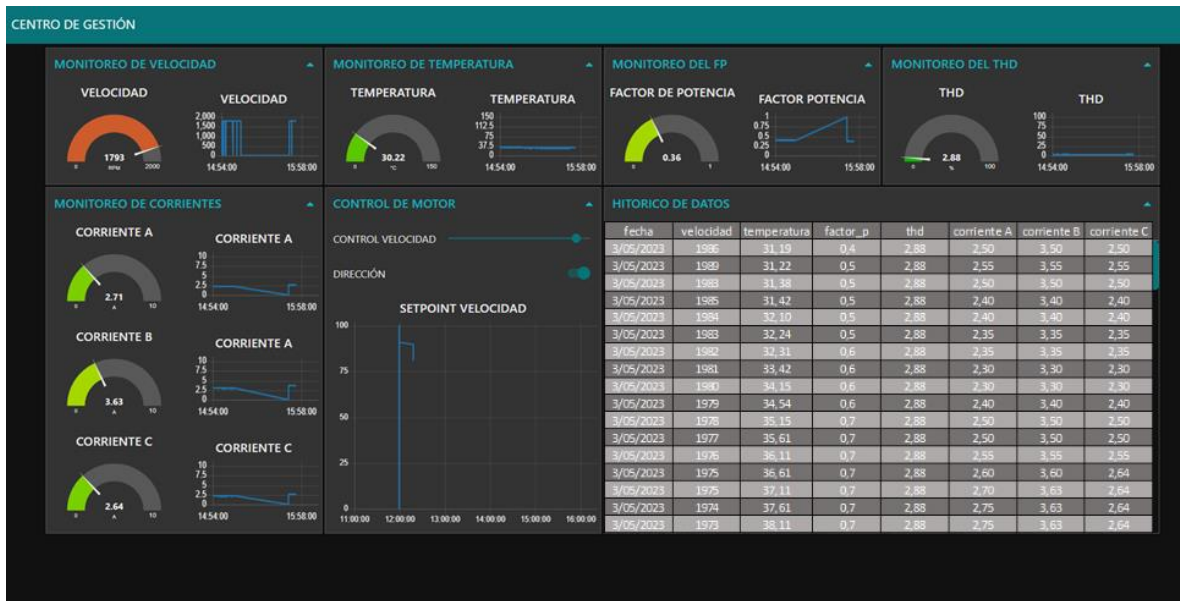


Figura 29: interfaz de usuario.

En la figura 29 se observa la distribución final que le fue asignada a la interfaz de usuario, la cual, le permitirá tener una interacción más amigable con la información adquirida por los distintos equipos de toma de datos.

Capítulo 5: Pruebas y resultados

5.1 Calibración de la tarjeta de adquisición de temperatura

Para verificar el correcto funcionamiento de la tarjeta de adquisición de temperatura, se realizó la calibración de esta haciendo uso de un termómetro, para lo cual, fue necesario realizar la toma de los datos con ambos instrumentos, y realizar la comparación de los diferentes datos, los resultados se muestran en la Tabla 2 y Figura 30.

Tabla 2: Calibración de la tarjeta de adquisición de temperatura.

Calibración de la tarjeta de adquisición de temperatura		
Valor medido por la tarjeta (C°)	Valor medido con termómetro (C°)	Error (%)
31,19	31,20	0,03
31,22	31,23	0,03
31,38	31,4	0,06
31,42	31,45	0,10
32,1	32,1	0,00
32,24	32,25	0,03
32,31	32,33	0,06
33,42	33,42	0,00
34,15	34,15	0,00
34,54	34,55	0,03
35,15	35,25	0,28

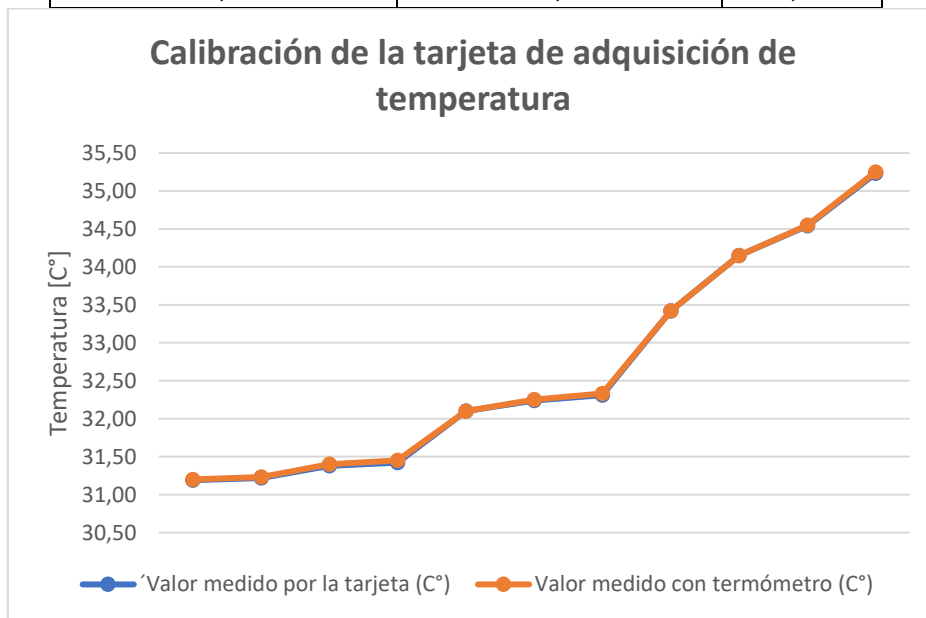


Figura 30: Calibración de la tarjeta de adquisición de temperatura.

Se observa en la Figura 30 que los valores que hacen referencia a cada uno de los datos obtenidos con la tarjeta y el termómetro coinciden con un error no mayor a 0,10 %. Este valor es implementado como factor de corrección en las medidas de temperatura realizadas por la tarjeta de adquisición de datos, para corregir la información suministrada por esta.

5.2 Calibración de la tarjeta de adquisición de velocidad

Para verificar el correcto funcionamiento de la tarjeta de adquisición de velocidad, se realizó la calibración de esta haciendo uso de un tacómetro, para lo cual, fue necesario realizar la toma de los datos con ambos instrumentos mientras se tenía en marcha el motor de inducción. Una vez obtenida la información de interés, se realiza la comparación de los diferentes datos, los resultados se muestran en la Tabla 3 y Figura 31.

Tabla 3: Calibración de la tarjeta de adquisición de temperatura

Calibración de la tarjeta de adquisición de velocidad		
Valor medido por la tarjeta (RPM)	Valor medido con tacómetro (RPM)	Error (%)
1796	1798	0,11
1796	1796	0,00
1798	1798	0,00
1799	1800	0,06
1800	1800	0,00
1795	1797	0,11
1797	1798	0,06
1798	1799	0,06
1799	1800	0,06
1797	1798	0,06
1798	1799	0,06

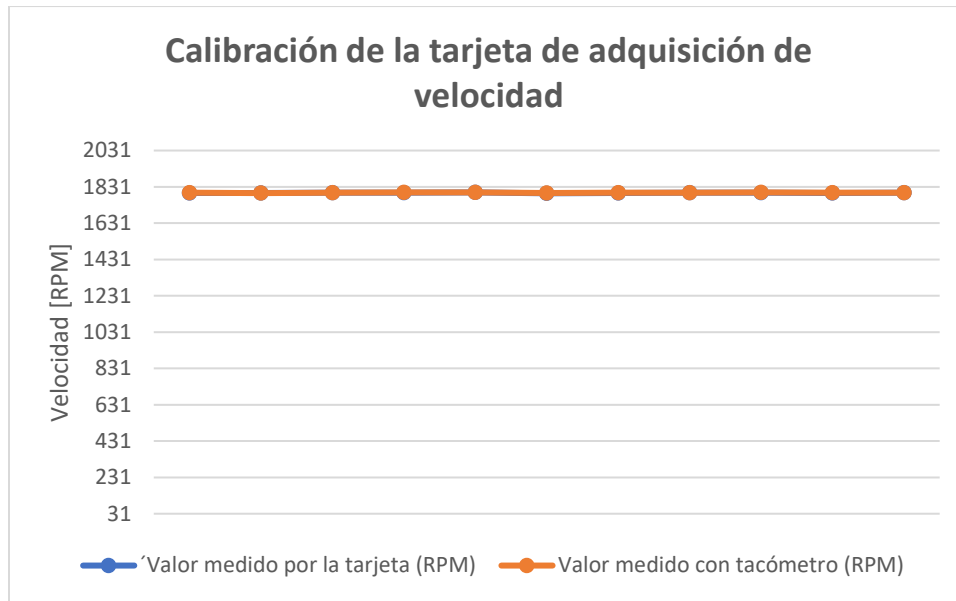


Figura 31: Calibración de la tarjeta de adquisición de velocidad

Se observa en la Figura 31 que los valores que hacen referencia a cada uno de los datos obtenidos con la tarjeta y el tacómetro coinciden con un error no mayor a 0,11 %. Este valor es implementado como factor de corrección en las medidas de velocidad realizadas por la tarjeta de adquisición de datos, para corregir la información suministrada por esta.

A continuación, se muestran los resultados de algunas pruebas realizadas que permiten verificar el funcionamiento general del sistema propuesto, así como sus potencialidades.

5.3 Descripción de la prueba

Para el desarrollo de las pruebas se utilizó un motor de inducción De Lorenzo DL 2053A que cuentan con las características que se muestran en la Figura 32.

DE LORENZO		
20 Viale Romagna - 20089 ROZZANO (Milano) ITALY		
DL 2053A	3 ~ M	S/N 08/00878.1
220 / 380V Δ/Y	12.8 / 6.8A Δ/Y	
3kW	cos φ = 0.85	
1680min ⁻¹	60Hz	
380V	5.5A	
F IEC 34-1	CE	IP 44

Figura 32: Placa de especificaciones del motor probado.

En el motor se simularon fallas en el devanado del rotor y se midieron variables como la temperatura en el bobinado estático (en °C), la velocidad del rotor (en RPM), las corrientes en cada una de las fases que alimentan el estator, y el factor de potencia. Los valores de las variables medidas son comparadas para las condiciones con y sin fallas en el rotor.

Para realizar la medición de temperatura, se instaló un sensor de temperatura análogo MCP9700 de bajo consumo, el cual convierte la temperatura en una tensión analógica. En la Figura 33 se muestra la instalación del sensor de temperatura directamente en el estator de motor bajo estudio, como se puede observar, el sensor está midiendo la temperatura directamente en el bobinado estático, lo cual permite tener información relevante para posteriores análisis del comportamiento del motor durante su uso y de esta forma realizar la gestión de mantenimiento.



Figura 33: Instalación del sensor de temperatura en el devanado del estator.

En la Figura 34 se muestra la topología del sistema implementado, sus diferentes componentes y protocolos de comunicación. En la Figura 35 se evidencian los diferentes componentes físicos que hacen parte de la topología.



Figura 34: Topología del sistema para la adquisición temperatura en el estator del motor



Figura 35: Componentes físicos del sistema para la adquisición temperatura en el estator del motor.

Una vez instalado el sensor de temperatura en el bobinado del estator, se procede a la conexión de este en la tarjeta diseñada para la captura de los datos de temperatura, que es la encargada de recolectar y transmitir los valores tomados cada 0,5 segundos al servidor local, el cual, por medio del protocolo de comunicación MQTT envía la información al servidor en la nube, haciendo uso por de Node-red y se posteriormente se presenta de manera gráfica en la interfaz de usuario.

Para la medición de velocidad angular, se conectó a la tarjeta programada un encoder Omron E6B2 de 2000 P/R, el cual había sido previamente acoplado al eje del motor bajo prueba. En la Figura 36 se muestra la instalación del encoder en el rotor del motor bajo estudio, el cual será el encargado de enviar los pulsos que permitirán cuantificar la velocidad en RPM a la que gira el eje del motor de inducción.



Figura 36: Sistema para la adquisición de velocidad en el eje del motor.

En La Figura 37 se muestra la topología del sistema implementado, sus diferentes componentes y protocolos de comunicación. En la Figura 38 se evidencian los diferentes componentes físicos que hacen parte de la topología.



Figura 37: Topología del sistema para la adquisición velocidad angular del rotor.

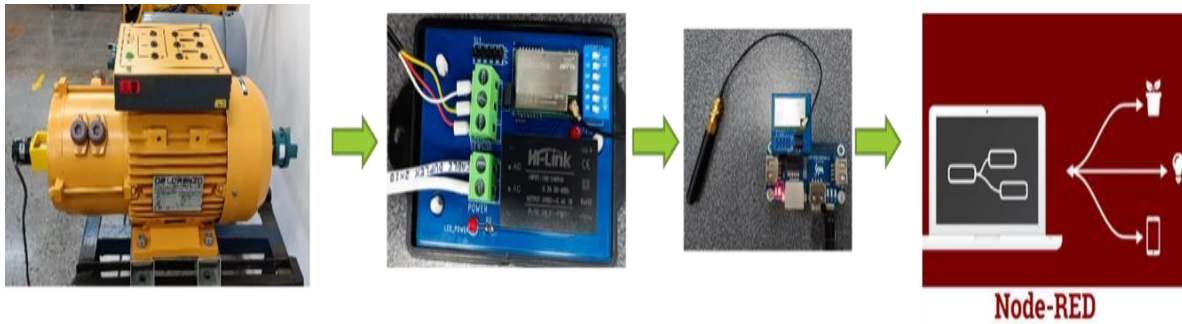


Figura 38: Componentes físicos del sistema para la adquisición velocidad angular del rotor.

En la Figura 39 se muestra la topología del sistema implementado, sus diferentes componentes y protocolos de comunicación. En la Figura 40 se evidencian los diferentes componentes físicos que hacen parte de la topología.

Se presentan los equipos utilizados para realizar la adquisición de las variables eléctricas, donde se logra identificar el medidor de energía multifuncional de tablero EasyLogic PM2120 con sus sensores de corriente y tensión instalados en el motor de inducción, este equipo es el encargado de realizar la adquisición de los parámetros eléctricos de interés, los cuales son enviados mediante comunicación modbus RS-485 al servidor local, y este posteriormente envía la información al servidor en la nube.

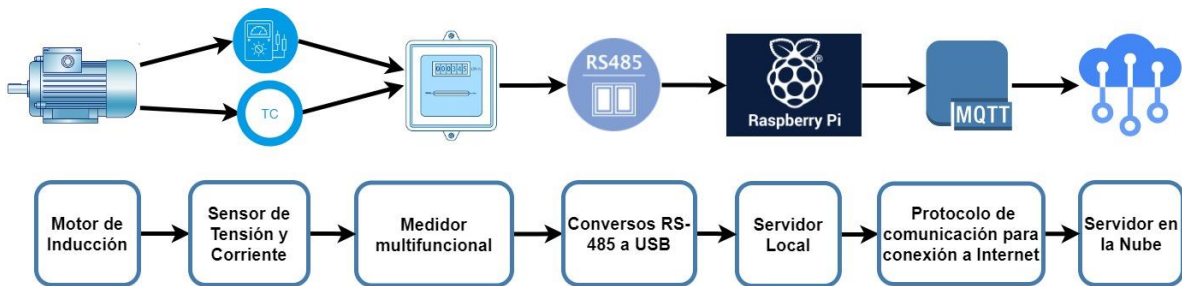


Figura 39: Topología del sistema para la adquisición de variables eléctricas del motor.

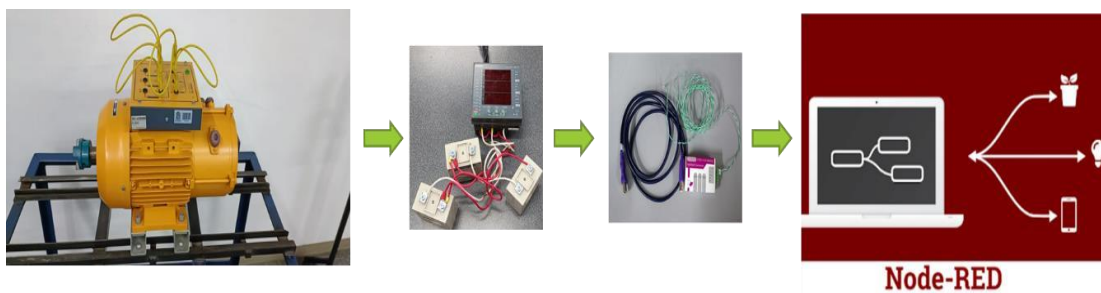


Figura 40: Componentes físicos del sistema para la adquisición de variables eléctricas del motor.

Una vez realizada la adquisición de los datos o variables de interés, esta información es gestionada por el servidor local y enviada al servidor en la nube, donde es almacenada en una base de datos y mostrada de forma gráfica en la interfaz de usuario. Con la información recolectada, es posible implementar informes de los eventos presentados durante el funcionamiento del motor, al igual que un historial con el que se podría implementar analítica de datos y usar herramientas de inteligencia artificial, y de esta forma hacer predicciones del funcionamiento del motor de inducción o identificar las condiciones de fallas más recurrentes para ajustar y/o modificar las estrategias de mantenimiento.

5.4. Análisis y resultados

La prueba realizada consistió en simular una falla en el bobinado del rotor del motor de inducción DE LORENZO DL 2053A, el cual cuenta con tres terminales de conexión marcadas con las letras K, L y M, las cuales deben ser conectadas en cortocircuito (en lo que se denomina operación normal) para iniciar la marcha como si fuera un motor del tipo jaula de ardilla. Inicialmente, se procede a poner en marcha el sistema con dicho motor bajo funcionamiento normal con las tres terminales del rotor en cortocircuito, como se observa en la Figura 41 (a).



(a)

(b)

Figura 41: Motor de rotor devanado en operación normal, (a), y con falla en el rotor, (b).

Una vez el motor se encuentra en marcha, se procede a generar la falla, que consiste en retirar el cortocircuito entre las terminales K y L (lo que se denomina operación en falla). Esto se observa en la Figura 41 (b), generando una serie de comportamientos anormales en la máquina, que se ven reflejados en cada una de las variables que son medidas: temperatura, velocidad, corriente, factor de potencia.

Luego de realizar la adquisición de las variables de interés, se procede a realizar el análisis de los resultados de las mediciones. Donde se definieron como principales variables de interés para la condición de falla estudiada, la temperatura en las bobinas del estator, velocidad del rotor en RPM, factor de potencia y corrientes del

estator. En las magnitudes de estas variables se logra evidenciar una relación directa con el estado de operación del motor de inducción, es decir, permiten inferir el estado de funcionamiento del motor, en operación normal y en operación bajo falla en el rotor.

En la Figura 42 se muestran los resultados de la medición de temperatura. Los datos que se muestran fueron tomados cada 0,5 segundos, donde los valores entre los 0 s y 150 s hacen referencia al estado de funcionamiento normal, mientras que los valores entre el rango de los 150 s y los 320 s hacen referencia al estado de funcionamiento con falla.

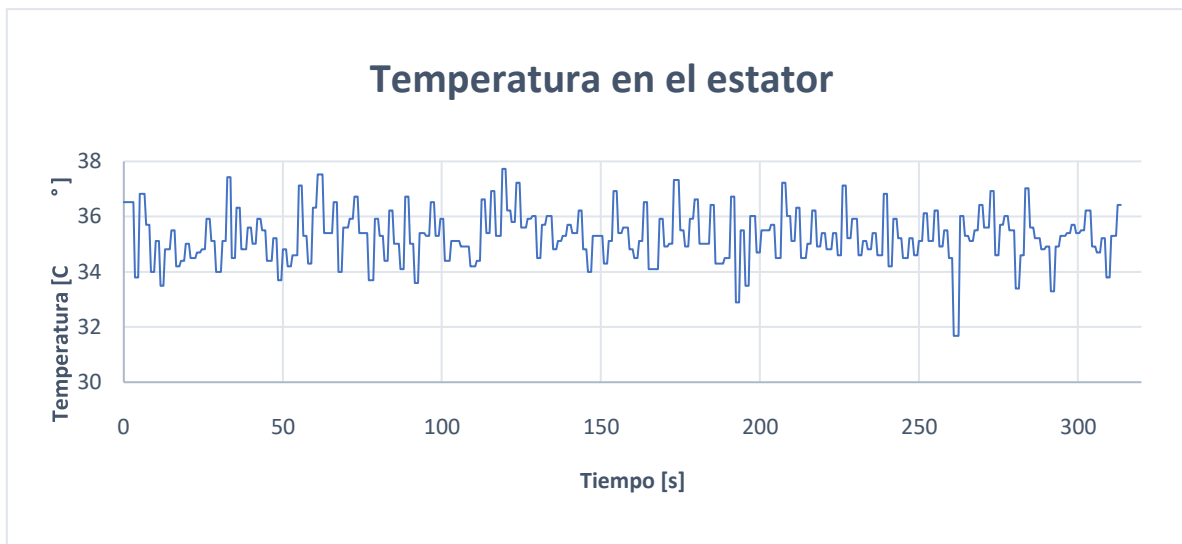


Figura 42: Medida de temperatura

Se logra observar que los valores de temperatura son muy inerciales, y no tienen un comportamiento que permita a primera vista identificar un patrón que brinde la posibilidad de identificar claramente la falla presentada en el rotor. Debido a lo anterior, se realiza el cálculo de los valores máximo, promedio y mínimo, de los datos almacenados, tanto para la etapa de funcionamiento normal como la de funcionamiento bajo falla, obteniéndose los resultados que se presentan en la Tabla 4.

Tabla 4: Medida de temperatura.

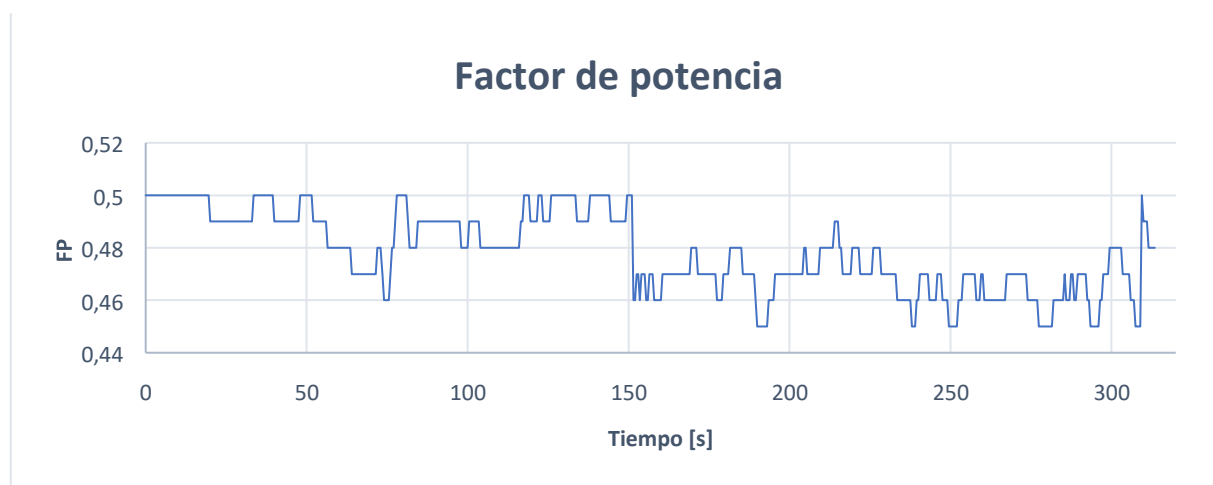
	Máximo			Promedio			Mínimo		
	Sin Falla	Con Falla	Δ (%)	Sin Falla	Con Falla	Δ (%)	Sin Falla	Con Falla	Δ (%)
Temperatura (°C)	37,53	37,73	0,53	35,27	35,29	0,06	33,49	31,68	5,40

En la tabla anterior se observa que existe una variación de los valores de la temperatura relacionados con los dos tipos de operación, es decir se logra evidenciar un aumento en el valor máximo de la temperatura en lo que respecta a la operación con falla de $0,2^{\circ}$ centígrados comparado con el funcionamiento en operación normal, cabe destacar que la prueba se realizó durante 5 minutos con el fin de no poner en riesgo la integridad del motor. Sin embargo, fue posible demostrar la aplicabilidad del sistema diseñado e implementado para la medición de la variación de la temperatura interna del bobinado del estator del motor de inducción debido a la condición de falla y a pesar de la pequeña variación.

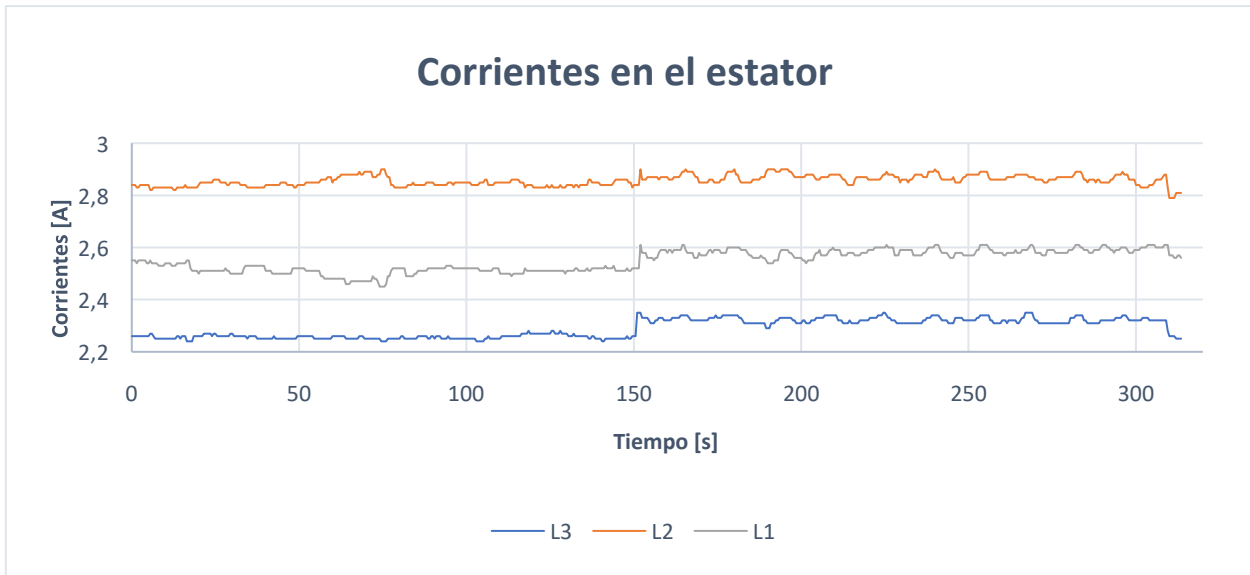
Los resultados de las mediciones de velocidad, factor de potencia y corrientes, para las condiciones de operación normal y con falla se muestran en la Figura 43 (a, b y c) respectivamente.



(a)



(b)



(c)

Figura 43: Histórico de velocidad del rotor (a), factor de potencia (b), corrientes en el estator (c).

Las principales características de los resultados de medición presentados en la Figura 43, se resumen en la Tabla 5, donde IL1, IL2, IL3 y FP corresponden a las corrientes en las líneas 1, 2, 3 que alimentan el estator del motor de inducción y Factor de Potencia trifásico, respectivamente.

Tabla 5: Características de las mediciones de velocidad del rotor, factor de potencia, corrientes en el rotor.

	Máximo			Promedio			Mínimo		
	Sin Falla	Con Falla	Δ (%)	Sin Falla	Con Falla	Δ (%)	Sin Falla	Con Falla	Δ (%)
Velocidad (rpm)	1782	1782	0,00	1780	1772	0,45	1778	1753	1,41
FP (l)	0,5	0,5	0,00	0,49	0,47	4,08	0,46	0,45	2,17
IL1 (A)	2,55	2,61	2,35	2,51	2,56	1,99	2,45	2,45	0,00
IL2 (A)	2,9	2,9	0,00	2,85	2,86	0,35	2,82	2,79	1,06
IL3 (A)	2,27	2,35	3,52	2,26	2,30	1,77	2,24	2,24	0,00

De los resultados presentados en la Figura 43 y la Tabla 5, se logra evidenciar que hay una correlación directa entre el comportamiento de la velocidad, el factor de potencia y las corrientes durante las condiciones de operación con y sin falla. En la

Figura 43 (a), se puede notar que el valor máximo de la velocidad, tanto en funcionamiento normal como con falla conserva el mismo valor, caso contrario a lo ocurrido con los valores mínimos, que presentan una variación significativa de 1,41 %, lo que equivale a una diferencia de 29 RPM, dejando claro que existe una afectación directa muy importante en el funcionamiento del motor de inducción que es claramente evidenciada en la velocidad.

En la Figura 43 (b), que representa el comportamiento del factor de potencia, se puede observar un comportamiento similar, donde el valor máximo se continúa manteniendo tanto para la operación sin falla como durante la operación con falla, pero se evidencia una variación significativa en los valores mínimos del 2,17 %.

En la Figura 43 (c), que muestra el comportamiento de las corrientes que alimentan el estator del motor de inducción, se logra evidenciar un aumento en los máximos de la corriente el cual es del 3,52 % para la línea L3.

A partir de las anteriores pruebas se pudo verificar lo siguiente:

El sistema implementado permite hacer adecuadamente y mediante una interfaz amigable, simple y de manera remota los controles de encendido y parada de un motor de inducción trifásico;

El sistema implementado permite medir y almacenar información relevante de la operación normal del motor de inducción para la gestión energética de las plantas de producción, como lo son el consumo de energía y el factor de potencia;

El sistema implementado permite generar alarmas cuando las condiciones de operación del motor de inducción cambian con respecto a las normales, como en las condiciones de falla;

El sistema implementado es lo suficientemente sensible para determinar variaciones mínimas en las variables medidas que pueden asociarse con condiciones anómalas de operación;

El sistema implementado permite crear históricos para el entrenamiento de sistemas expertos, gestión del mantenimiento e incluso aplicación de herramientas de inteligencia artificial para el diagnóstico y pronóstico de condiciones de falla.

Las tecnologías implementadas demostraron ser estables y confiables en cuanto a sensado de las variables de interés.

El control de marcha y paro del motor respondió de forma correcta y según lo esperado, este fue probado en una banda transportadora industrial en los laboratorios de la Universidad de Antioquia.

Los dispositivos de comunicación mantienen de forma estable la conexión.

El sensor de temperatura soporta de forma correcta los rangos de temperatura que se presentan durante el funcionamiento del motor de inducción.

6. Conclusiones

En este trabajo de maestría se presenta el desarrollo de un sistema descentralizado remoto, basado en tecnologías IoT y de código abierto para la adquisición y control de variables eléctricas en un motor de inducción trifásico.

- ✓ Se determinaron los requerimientos del sistema descentralizado propuesto y las componentes de hardware y software basados en código abierto necesarios para su implementación.
- ✓ Se realizó el diseño e implementación de las componentes de hardware, como también la programación basada en código abierto de los dispositivos esclavos y el controlador maestro.
- ✓ Se implementó un sistema de comunicación inalámbrica para el control y monitoreo de variables desde un servidor IoT en una plataforma de código abierto.
- ✓ Se realizaron las validaciones necesarias del sistema implementando los ajustes necesarios y la publicación de los resultados obtenidos.
- ✓ Se desarrolló una tarjeta electrónica de bajo costo para la medición, adquisición y envío de los datos de temperatura del bobinado del estator en los motores eléctricos, las medidas de temperatura fueron calibradas haciendo uso de un termómetro con certificado de calibración, el cual sirvió como patrón para la calibración de la tarjeta.
- ✓ Se desarrolló una tarjeta electrónica de bajo costo para la medición, adquisición y envío de los datos de velocidad angular del eje de los motores eléctricos, las medidas de velocidad fueron calibradas haciendo uso de un tacómetro con certificado de calibración, el cual sirvió como patrón para la calibración de la tarjeta.
- ✓ Se desarrolló el firmware para las tarjetas de adquisición de datos tanto de velocidad angular como de temperatura en el motor de inducción.
- ✓ Se implementó por medio de comunicación modbus RS-485 la adquisición de las variables eléctricas de interés por medio de un medidor de energía multifuncional de tablero EasyLogic PM2120, donde fue necesario realizar la interpretación de las tramas que envía este equipo al ser solicitadas cada una de las variables de interés.
- ✓ Se implementó una red inalámbrica WiFi para la transmisión de información hacia el servidor central en la nube y el envío de alarmas, en caso de detectar que alguno de los datos se encuentra por fuera de su tendencia de funcionamiento normal.
- ✓ Se desarrolló el software para el servidor local o unidad de control, encargado de realizar el proceso de petición de los datos a las tarjetas de adquisición de datos, analizar los datos y transmitir reportes al servidor en la nube.
- ✓ Se desarrolló el software para el servidor en la nube, el cual es el encargado de realizar el proceso almacenamiento de los datos enviados vía Wifi desde el

servidor local o unidad de control, analizar los datos y transmitir reportes de alarmas.

- ✓ El sistema permite hacer control de velocidad y marcha del motor de inducción de forma remota y ágil.
- ✓ La red inalámbrica realizó mediciones de los datos de interés como velocidad, temperatura, corrientes, Factor de potencia, THD y transmitió reportes hacia el servidor en la nube. En estas mediciones se observó que tanto las tarjetas y equipos de adquisición de datos como el servidor local o unidad de control funcionaron correctamente.
- ✓ El sistema implementado puede realizar la toma de datos de forma remota y en tiempo real, reportando de esta forma la información al servidor y permitiendo al usuario tener acceso ágil y fácil a la información, lo que es una gran ventaja ya que le permite realizar toma de decisiones con mayor rapidez, al tener un monitoreo continuo sobre el motor.
- ✓ Se desarrolló una base de datos que permite el almacenamiento de la información recolectada, para que posteriormente pueda ser consultada por los usuarios.
- ✓ Se implementó la interfaz de usuario o Dashboard, que permite la interacción de forma visual entre el usuario y los datos que están siendo adquiridos por las tarjetas y diferentes dispositivos, control de marcha, velocidad del motor y además muestra la información que se almacena en la base de datos.
- ✓ Debido a su bajo costo, fácil fabricación y la posibilidad de hacer medidas de variables eléctricas en el motor de forma remota e inalámbrica, este sistema es de gran utilidad, ya que puede ser implementado en la industria e incluso en el ámbito académico.
- ✓ El sistema tiene la capacidad de trabajar como esclavo, es decir, se puede conectar a un PLC u otra herramienta que lo controle o de forma independiente.
- ✓ Para el desarrollo del sistema en general, se hace uso de software basados en código abierto y dispositivos de tecnología IoT, lo cual permite que el sistema sea flexible y escalable. Al sistema diseñado se pueden agregar otras tarjetas de medición de variables, como pueden ser de campo magnético, vibraciones y flujo axial, que permitirían un diagnóstico preciso y completo del estado de salud de los motores de inducción.
- ✓ La información recolectada y posteriormente almacenada, permite crear una base de datos que es posible consultar para llevar un historial de uso de los motores, con el cual se puede implementar control de planes de mantenimiento posteriores, con un enfoque principal en el mantenimiento predictivo y preventivo que son de suma importancia para preservar el óptimo funcionamiento de los motores y más aún si estos hacen parte de una línea de producción donde son elementos críticos.
- ✓ El sistema diseñado e implementado podrá aplicarse a máquinas eléctricas de funcionamiento similar y podrá usarse en entornos industriales y académicos, teniendo en cuenta que se implementó un sistema (prototipo) que podría ser clasificado en una categoría mayor o igual a la TRL (Nivel de Transferencia Tecnológica) 4.

7. Trabajos futuros

- ✓ Desarrollar tarjetas que permitan realizar la medición del flujo axial en los motores de inducción y así poder determinar en tiempo real las fallas que se puedan estar presentando por cortocircuitos entre espiras del estator o espiras abiertas en los motores de rotor devanado.
- ✓ Realizar con el historial de la información almacenado, análisis predictivo de posibles fallas en los motores, haciendo uso de Machine Learning. Y de esta manera, el sistema puede ser usado para realizar análisis de diagnóstico y desempeño de los motores de inducción con base en la información de las variables medidas y la creación de historiales y bases de datos de operación. Se pueden usar sistemas de inteligencia artificial para generar alarmas.
- ✓ Implementación del sistema en el ámbito académico para permitir a estudiantes que no estén físicamente presente en un laboratorio de máquinas eléctricas, realizar prácticas de laboratorio por medio de la toma de datos de manera remota.
- ✓ Migrar el sistema a máquinas rotativas de corriente directa y de esta forma poder realizar adquisición de variables de interés y control de estas.
- ✓ Adaptar el sistema para medir presión, temperatura y diversas variables eléctricas de interés.
- ✓ Implementar el sistema en proyectos de generación de energías renovables y de esta forma tener en tiempo real una base de datos que suministre información que sea relevante y de interés para los usuarios.

8.Anexos

8.1 Código de programación para la tarjeta de temperatura

```
#include <16f1826.h>
#device adc=10
#fuses intrc_io,protect,wdt,nomclr
#use delay(clock=8M)
#use rs232(baud=9600,uart1)
#use timer(timer=0,tick=1ms)

// Definición canal de comunicación (M1=0,M2=10,M3=20,etc..)
#define CH 10

// Definición de pines de configuración

#define AUX pin_b0
#define LED pin_b5
#define M1 pin_b3
#define M0 pin_b4
#define S1 pin_a6
#define S2 pin_a7
#define S3 pin_a0
#define S4 pin_a1
#define S5 pin_a2
#define S6 pin_a3

int dato=0,canal=CH;
int divisor=0,sec=0;
int1 s=0,t=0,c=0;
int16 conversion=0;
float temperatura=0;

int ID=0;

char config[6]={0xC0, // Encabezado de trama
 0, // Dirección dispositivo parte alta
 0, // Dirección dispositivo parte baja
 0x19, // Configuración serial
 0, // Configuración canal
 0x80
}; // Configuración Fixed Point o comunicación transparente
```

```
#int_rda // Cuando se recibe la petición por parte del maestro, se hace el envío de los datos
```

```
void serial(){  
  dato=getc();  
  switch(sec){  
  case 0:  
    switch(dato){  
      case 'T': // Cuando se haga la petición, se activa la trama de respuesta  
        s=1;  
        set_ticks(0);  
        break;  
      case 'C': // Se verifica la secuencia para cambio de canal  
        sec=2;  
        break;  
    }  
    break;  
  case 2:  
    if(dato=='H'){  
      sec=3;  
    }  
    else{  
      sec=0;  
    }  
    break;  
  case 3:  
    canal=dato;  
    c=1;  
    sec=0;  
    break;  
  }  
}
```

```
#int_timer1  
void tiempo(){ // Cada segundo se hace la lectura del sensor  
  divisor++;  
  if(divisor==4){  
    divisor=0;  
    t=1;  
  }  
}
```

```

void config_channel(){
    output_high(LED);
    delay_ms(100);
    output_low(M0); // Se configura el módulo en modo configuración
    output_high(M1);
    delay_ms(200); // Se calcula la dirección del dispositivo

ID=(int)input(s1)*1+(int)input(s2)*2+(int)input(s3)*4+(int)input(s4)*8+(int)input(s5)*1
6;
    config[2]=ID; // Se modifica la dirección del dispositivo
    config[4]=canal; // Se modifica el canal del dispositivo
    int k=0;
    for(k=0;k<6;k++){
        putc(config[k]);
    }
    output_toggle(LED);
    delay_ms(200);
    output_low(M0); // Se configura en modo transmisión
    output_low(M1);
    output_toggle(LED);
    delay_ms(200);
}

void main(){
    output_low(LED); // Se inicializa el LED apagado
    setup_wdt(WDT_2S);
    setup_adc(adc_clock_internal);
    setup_adc_ports(sAn5);
    set_adc_channel(5);
    setup_vref(VREF_ON|VREF_ADC_2v048);
    setup_adc_reference(VSS_FVR);
    enable_interrupts(global); // Se habilita la máscara de interrupciones
    enable_interrupts(int_rda); // Se habilita la interrupción por puerto serial
    enable_interrupts(int_timer1); // Se habilita interrupción del timer1
    setup_timer_1(T1_internal|T1_div_by_8); // Se configura el timer1 para hacer la
lectura del sensor cada 0.5 segundos
    config_channel();
    for(;;){

        restart_wdt();

        if(s){ // Cuando se recibe la petición se cambia el estado del LED

```

```

    if(get_ticks())>100){ // Se espera un periodo múltiplo de 200ms para enviar la
respuesta
    putc(0x00);
    putc(50);
    putc(canal);
    printf("T%02uN%3.2f\n\r",ID,temperatura);
    s=0; // Se deshabilita la bandera en espera de otra petición.
    output_toggle(LED);
    }
}

```

```

if(c){
    c=0;
    config_channel();
}

```

```

if(t){ // Cada segundo se hace la consulta de la temperatura y se envía en dos
bytes

```

```

    t=0;

```

```

ID=(int)input(s1)*1+(int)input(s2)*2+(int)input(s3)*4+(int)input(s4)*8+(int)input(s5)*1
6;

```

```

    conversion=read_adc();
    //printf("%lu\n\r",conversion);
    //temperatura=((float)conversion-102)*0.488;
    temperatura=((float)conversion-249)*0.199;//MCP9700
    //temperatura=((float)conversion-201)*0.1009;//MCP9701
    // putc(0x00);
    //putc(50);
    //putc(canal);
    //printf("T%02uN%3.2f\n\r",ID,temperatura);
    //output_toggle(LED);
}

```

```

}

```

```

}

```


8.2 Código de programación para la tarjeta de velocidad

```
#include <16f1827.h>
#fuses intrc_io,protect,wdt,nomclr
#use delay(clock=4M)
#use rs232(baud=9600,uart1)
#use timer(timer=2,tick=1ms)

// Definición canal de comunicación
#define CH 0

// Definición de pines de configuración

#define AUX pin_b0
#define LED pin_b5
#define M1 pin_b3
#define M0 pin_b4
#define S1 pin_a6
#define S2 pin_a7
#define S3 pin_a0
#define S4 pin_a1
#define S5 pin_a2
#define S6 pin_a3

int dato=0,canal=CH;
int sec=0;
int1 s=0,p=0,c=0;
int16 conteo,RPM;

int ID=0;

char config[6]={0xC0, // Encabezado de trama
0, // Dirección dispositivo parte alta
0, // Dirección dispositivo parte baja
0x19, // Configuración serial
0, // Configuración canal
0x80
}; // Configuración Fixed Point o comunicación transparente
```

```
#int_rda // Cuando se recibe la petición por parte del maestro, se hace el envío de los datos
```

```
void serial(){  
  dato=getc();  
  switch(sec){  
  case 0:  
    switch(dato){  
      case 'T': // Cuando se haga la petición, se activa la trama de respuesta  
        s=1;  
        set_ticks(0);  
        break;  
      case 'C': // Se verifica la secuencia para cambio de canal  
        sec=2;  
        break;  
    }  
  break;  
  case 2:  
    if(dato=='H'){  
      sec=3;  
    }  
    else{  
      sec=0;  
    }  
  break;  
  case 3:  
    canal=dato;  
    c=1;  
    sec=0;  
  break;  
  }  
}
```

```
#int_timer0  
void tiempo(){ // Cada segundo se hace la lectura del sensor  
  conteo=get_timer1();  
  set_timer1(0);  
  p=1;  
}
```

```
void config_channel(){  
  output_high(LED);  
  delay_ms(100);  
  output_low(M0); // Se configura el módulo en modo configuración  
  output_high(M1);  
  delay_ms(200); // Se calcula la dirección del dispositivo
```

```

ID=(int)input(s1)*1+(int)input(s2)*2+(int)input(s3)*4+(int)input(s4)*8+(int)input(s5)*1
6;
config[2]=ID; // Se modifica la dirección del dispositivo
config[4]=canal; // Se modifica el canal del dispositivo
int k=0;
for(k=0;k<6;k++){
    putc(config[k]);
}
output_toggle(LED);
delay_ms(200);
output_low(M0); // Se configura en modo transmisión
output_low(M1);
output_toggle(LED);
delay_ms(200);
}

```

```

void main(){
    output_low(LED); // Se inicializa el LED apagado
    setup_wdt(WDT_2S);
    port_b_pullups(0xff);
    enable_interrupts(global); // Se habilita la máscara de interrupciones
    enable_interrupts(int_rda); // Se habilita la interrupción por puerto serial
    //enable_interrupts(int_timer1); // Se habilita interrupción del timer1
    setup_timer_0(T0_INTERNAL|T0_DIV_256);
    enable_interrupts(int_timer0);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_8); // Se configura el timer1 para
hacer la lectura del sensor cada 0.5 segundos
    config_channel();
    set_timer1(0);
    for(;;){

        restart_wdt();

        if(s){ // Cuando se recibe la petición se cambia el estado del LED

            if(get_ticks())>100){ // Se espera un periodo múltiplo de 200ms para enviar la
respuesta
                putc(0x00);
                putc(50);
                putc(canal);
                printf("T%02uN%lu\n",ID,RPM);
                s=0; // Se deshabilita la bandera en espera de otra petición.
                output_toggle(LED);
            }
        }
    }
}

```

```

}

if(c){
    c=0;
    config_channel();
}

if(p){
    p=0;
    RPM=(int16)((float)conteo*3.66);

ID=(int)input(s1)*1+(int)input(s2)*2+(int)input(s3)*4+(int)input(s4)*8+(int)input(s5)*1
6;
}

}

}

```

8.3 Configuraciones de instalación general del servidor en la nube

<https://samircorrea.optimalsoft.com.co/--->>

```
ssh -p 2240 usuario@149.102.143.245
```

```
sudo dpkg-reconfigure tzdata
```

```
sudo ssh-keygen -b 4096 -t rsa
```

```
sudo passwd root
```

```
claveRoot
```

```
Node-red setting
```

```
# npm install -g --unsafe-perm Node-red-admin
```

```
Node-red admin hash-pw
```

```
ClbgkkpSamirr23#
```

```
$2b$08$m87DxFu4051TKbWb/J..OeZjQnIJCCLH.Ad5eWzizonv1V.SCYuHO
```

```
Node-red setting
```

```
/etc/apache2/ssl2023/optimalsoft.key
```

```
/etc/apache2/ssl2023/optimalsoft.pem
```

```
Node-red --settings
```

```
sudo adduser usuarioNuevo
claveUsuario
```

```
sudo nano /etc/sudoers
usuarioNuevo ALL=(ALL:ALL) ALL
usuarioNuevo ALL=(ALL) NOPASSWD:ALL
```

```
apt-get install tmux
```

```
sudo passwd root
claveUsuarioNuevo
```

```
apt-get install ufw -y
sudo ufw status numbered
#sudo ufw enable
#sudo ufw allow out 3346/tcp
#sudo ufw allow in 3346/tcp
sudo ufw allow 80
sudo ufw allow 443
sudo ufw allow int 80/tcp
sudo ufw allow out 80/tcp
sudo ufw allow 2240/tcp
sudo ufw allow ssh
sudo ufw allow 1880
sudo ufw allow out 1880/tcp
sudo ufw allow in 1880/tcp
sudo ufw allow 8443
sudo ufw allow out 8443/tcp
sudo ufw allow in 8443/tcp
```

```
sudo ufw status numbered
sudo ufw delete 2
sudo ufw status
```

```
nano /etc/ssh/sshd_config
```

```
puerto 22 > 2240
```

```
sudo apt install openjdk-17-jre openjdk-17-jdk
```

1. Ocultar version del apache y version del so
apt-get install apache2

sudo nano /etc/apache2/apache2.conf
agregar al final del archivo

```
ServerSignature Off  
ServerTokens Prod
```

cp el default-ssl samircorrea-ssl.conf
sudo a2enmod ssl
sudo a2enmod rewrite

```
#sudo a2ensite default-ssl (este lo deajo sano)  
sudo a2ensite samircorrea-ssl.conf
```

```
#sudo a2dissite 000-default.conf
```

nano 000-default.conf

```
<VirtualHost *:80>  
    RewriteEngine On  
    RewriteCond %{HTTPS} off  
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

por si hay errores de escritura
apachectl configtest

ahora en /etc/apache2/sites_avalibles/samircorrea-ssl.conf

```
ServerAdmin admin@samircorrea.optimalsoft.com.co  
ServerName samircorrea.optimalsoft.com.co:443  
DocumentRoot /var/www/html/samircorrea
```

```
SSLEngine on  
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1  
SSLHonorCipherOrder on  
SSLCipherSuite "EECDH+ECDSA+AESGCM  
EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256  
EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH  
EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS"
```

```
SSLCertificateFile /etc/apache2/ssl2023/optimalsoft.pem  
SSLCertificateKeyFile /etc/apache2/ssl2023/optimalsoft.key
```

```
npm install -g Node-red
```

Node-red

```
nano /usr/lib/node_modules/Node-red/node_modules/Node-red-admin/settings.js
```

```
nano /usr/lib/node_modules/Node-red/node_modules/Node-red-admin/settings.js
```

```
https: {  
  key: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft.key'),  
  cert: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft.pem')  
},
```

cambia puerto 8880 por cloudflare

```
https://samircorrea.optimalsoft.com.co:1880
```

```
sudo nano /etc/systemd/system/Node-red.service
```

```
journalctl -u Node-red -e -f -n 50
```

```
openssl rsa -in optimalsoft.key -text > optimalsoft.pem
```

```
https: {  
  key: require("fs").readFileSync('/etc/ssl/private/selfsigned.key'),  
  cert: require("fs").readFileSync('/etc/ssl/certs/selfsigned.crt')  
},
```

```
/** To password protect the Node-red editor and admin API, the following  
 * property can be used. See http://nodered.org/docs/security.html for details.  
 */
```

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
    password:  
"$2a$08$zZWtXTja0fB1pzD4sHcMyOCMyz2Z6dNbM6tI8sJogENOMcxWV9DN.",  
    permissions: "*"   
  }]  
},
```

```
/** To password protect the Node-red editor and admin API, the following  
 * property can be used. See http://nodered.org/docs/security.html for details.
```

```

*/
adminAuth: {
  type: "credentials",
  users: [{
    username: "admin",
    password:
"$2a$08$zZWtXTja0fB1pzD4sHcMyOCMyz2Z6dNbM6tI8sJogENOMcxWV9DN.",
    permissions: "*"
  },
  {
    username: "invitado",
    password:
"$2a$08$zZWtXTja0fB1pzD4sHcMyOCMyz2Z6dNbM6tI8sJogENOMcxWV9DN.",
    permissions: "read"
  }
  ],
},

```

Node-red -v

```
# vi /etc/systemd/system/Node-red.service
```

```
# To consult the log : journalctl -u Node-red
[Unit]
```

```
Description=Node-red is a tool for wiring together hardware devices, APIs and
online services in new and interesting ways.
```

```
After=syslog.target network.target
```

```
Documentation=http://nodered.org/
```

```
[Service]
```

```
Environment="NODE_OPTIONS="
```

```
Environment="NODE_RED_OPTIONS=-v"
```

```
#Full Path to Node.js
```

```
ExecStart=/usr/bin/Node-red $NODE_OPTIONS $NODE_RED_OPTIONS
```

```
WorkingDirectory=/root/.Node-red/
```

```
# User/Group that launches Node-red (it's advised to create a new user for Node-
red)
```

```
# You can do : sudo useradd Node-red
```

```
# then change the User=root by User=Node-red
```

```
User=root
```

```
Group=root
```

```
Nice=10
```

```
SyslogIdentifier=Node-red
```

```
StandardOutput=syslog
```

```
# Make Node-red restart if it fails
```



```
Restart=on-failure
# Node-red need a SIGINT to be notified to stop
KillSignal=SIGINT
```

```
[Install]
WantedBy=multi-user.target
```

```
# To consult the log : journalctl -u Node-red
```

```
[Unit]
Description=Node-red is a tool for wiring together hardware devices, APIs and
online services in new and interesting ways.
After=syslog.target network.target
Documentation=http://nodered.org/
```

```
[Service]
Environment="NODE_OPTIONS="
Environment="NODE_RED_OPTIONS=-v"
#Full Path to Node.js
ExecStart=/usr/bin/Node-red $NODE_OPTIONS $NODE_RED_OPTIONS
WorkingDirectory=/root/.Node-red/
# User/Group that launches Node-red (it's advised to create a new user for Node-
red)
# You can do : sudo useradd Node-red
# then change the User=root by User=Node-red
User=root
Group=root
Nice=10
SyslogIdentifier=Node-red
StandardOutput=syslog
# Make Node-red restart if it fails
Restart=on-failure
# Node-red need a SIGINT to be notified to stop
KillSignal=SIGINT
```

```
[Install]
WantedBy=multi-user.target
```

Por último recarga los demonios, activa servicio, inícialo y comprueba el estado:

```
# systemctl daemon-reload && systemctl enable Node-red.service && systemctl
start Node-red.service && systemctl status Node-red.service
```

```
-----
Node-red admin hash-pw
clavenoderd#
$2b$08$m87claCLH.Ad5r938378323eWzizonv1V.SCYuHO
```

Node-red setting

nano /root/.Node-red/settings.js

```
/** To password protect the Node-red editor and admin API, the following
 * property can be used. See http://nodered.org/docs/security.html for details.
 */
```

```
adminAuth: {
  type: "credentials",
  users: [{
    username: "admin1",
    password: "$2b$08$dfasdfaeZjQnIJCCLHdfgsdhgfd4nv1V.SCYuHO",
    permissions: ""
  }]
},
```

```
/** Option 1: static object */
```

```
https: {
  key: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft2.pem'),
  cert: require("fs").readFileSync('/etc/apache2/ssl2023/optimalsoft.pem')
},
```

//yo tenia la llave en .key y la tube que pasar a .pem (no se porque pero funciona asi)

//cambie el puerto porque cloudflare waf no me dejo por otros puertos no comunes
uiPort: process.env.PORT || 8443,

debo ejecutarlo desde root, si lo hago desde otro usuario crea otra instancia

//mosquito

//apt-get install mosquito y cliente mosquito

apt-get install mosquito mosquito-client

service mosquito status

/etc/mosquitto/mosquitto.conf

//servidor

mosquitto_sub -h localhost -t tesstyedwin

//cliente

mosquito_pub -h localhots -t resstyedwin -m "hola mundo"

sudo apt install mosquito mosquito-clients

//server

```
mosquitto_sub -h localhost -t test
```

```
//cliente
```

```
mosquitto_pub -h localhost -t test -m "hello world"
```

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd samirc
```

```
Clbgkkpdsamic28
```

```
sudo nano /etc/mosquitto/mosquitto.conf
```

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwd
```

```
sudo systemctl restart mosquitto
```

```
//ahora bien
```

```
mosquitto_sub -h localhost -t testservidorpro -u samirc -P Clbgkkpdsamic28
```

```
mosquitto_sub -h 149.102.143.245 -t testservidorpro -u samirc -P Clbgkkpdsamic28  
-p 1883
```

```
mosquitto_sub -h 149.102.143.245 -t testservidorpro -u samirc -P Clbgkkpdsamic28  
-p 2095
```

```
sudo ufw allow 1883
```

```
cloudflare 2095
```

```
1883: MQTT, no encriptado
```

```
8883: MQTT, encriptado
```

```
8884: MQTT, encriptado, necesario certificado por parte del cliente
```

```
8080: MQTT sobre WebSockets, no encriptado
```

```
8081: MQTT sobre WebSockets, encriptado
```

```
//suscribo //desde el servidor
```

```
mosquitto_sub -h localhost -t testservidorpro -u samirc -P clavemosuitto-p 2095
```

```
-----  
//enel servidor publico
```

```
-----  
//en server publico
```

```
mosquitto_pub -h 149.102.143.245 -t "TEMPERATURA_S" -m "5" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

```
mosquitto_pub -h 149.102.143.245 -t "VELOCIDAD_S" -m "1002" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

```
mosquitto_pub -h 149.102.143.245 -t "FACTOR_P_S" -m "45" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

```
mosquitto_pub -h 149.102.143.245 -t "THD_S" -m "3" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

```
mosquitto_pub -h 149.102.143.245 -t "CORRIENTE_A_S" -m "45" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

```
mosquitto_pub -h 149.102.143.245 -t "CORRIENTE_B_S" -m "57" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095  
mosquitto_pub -h 149.102.143.245 -t "CORRIENTE_C_S" -m "56" -u "samirc" -P  
Clbgkkpdsamic28 -p 2095
```

Configuring MQTT SSL

```
sudo nano /etc/mosquitto/conf.d/default.conf  
mosquitto_pub -h 127.0.0.1 -p 8883 --cafile /etc/mosquitto/ca_certificates/ca.crt -t  
test -m 45  
listener 1883 localhost
```

```
listener 8883  
openssl req -new -x509 -days 1826 -key optimalsoft.key -out ca.crt
```

```
certfile /etc/mosquitto/certs/optimalsoft.pem  
cafile /etc/mosquitto/certs/ca.crt  
keyfile /etc/mosquitto/certs/optimalsoft.pem
```

```
sudo systemctl restart mosquitto
```

```
sudo ufw allow 8883
```

```
mosquitto_pub -h mqtt.example.com -t test -m "hello again" -p 8883 --capath  
/etc/ssl/certs/ -u "samirc" -P Clbgkkpdsamic28  
mosquitto_sub -h localhost -t test -u samirc -P Clbgkkpdsamic28 -p 8883
```

```
//zona horaria  
dpkg-reconfigure tzdata
```

```
sudo apt install openjdk-17-jre openjdk-17-jdk
```

```
sudo apt update && sudo apt -y upgrade  
sudo apt -y install mariadb-server mariadb-client
```

```
sudo mysql_secure_installation
```

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'claveroot' WITH  
GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON *.* TO 'otrousuario'@'%' IDENTIFIED BY  
'claveotrousuario' WITH GRANT OPTION;
```

```
sudo ufw allow out 3306/tcp
```

```
sudo ufw allow in 3306/tcp
```

```
-----
```

9.Referencias

- [1] E. G. Moreno, A. De, and R. J. V. Micó, *Automatización de procesos industriales*. .
- [2] O. C. Novillo-Vicu, Johnny and Rojas, Dixys Hern, ndez and Olivo, Bertha Maz, Jimmy Molina and Villavicencio, *Arduino y el internet de las cosas*. 2018.
- [3] UPME and CORPOEMA, “Procesos Y Tecnologías a Nivel Nacional E Internacional,” p. 191, 2014, [Online]. Available: http://www1.upme.gov.co/DemandaEnergetica/DeterminacionEficiencia/Informe_Final_Volumen_2.pdf.
- [4] T. M. U. W. P. R. Ned Mohan, *Electronica De Potencia Mohan-3Ra-Edicion*. 2009.
- [5] M. Clave, C. Con, and A. Industriales, “Analizador de calidad eléctrica y eficiencia de motores Fluke 438-II.”
- [6] Metrel, “MI 2892 Manual de instrucciones,” 2016.
- [7] A. Condition and M. Systems, “Simpli fi es Predictive Maintenance.”
- [8] A. D. E. Calidad, “Analizador de calidad de energía,” vol. 00, no. 800, pp. 1–13, 2014.
- [9] Y. P. A. Alcazar Oviedo and J. L. Villa Ramirez, “Monitoreo de Motores de Induccion Trifásico a Través de las Corrientes Eléctricas de Alimentación,” 2021.
- [10] C. M. COBA GUANOCHANGA, CARLOS ANDRÉS TENORIO CHANGO, “Diseño e implementación de un sistema de monitoreo, para el análisis dinamico de un motor eléctrico asíncrono trifásico de 3 Hp, en el laboratorio de control eléctrico de la Universidad de las Fuerzas Armadas ESPE Latacunga.” 2019.
- [11] A. Velasco and C. Alexander, *Implemetacion de una red GSM para monitoreo de variables eléctricas en un banco de variadores de frecuencia en el laboratorio de control y manipulación Automática de la escuela Ingeniería de mantenimiento*. 2019.
- [12] RUIZ CARLOS IVAN, “Implementación de un sistema de monitoreo remoto de variables eléctricas para un motor trifásico de inducción en el laboratorio de máquinas eléctricas de la carrera de electricidad,” 2021, [Online]. Available: http://repositorio.utn.edu.ec/bitstream/123456789/11207/2/04_MAUT_142_TRABAJO_GRADO.pdf.
- [13] I. Tecnológico, D. T. Gutiérrez, H. García, R. Antonio, R. A. Carlos, and N. D. E. L. Proyecto, “Sistema para medir, monitorear, analizar, y controlar disturbios de calidad, eficiencia y fallas en máquinas eléctricas, uso racional de la energía y el modo de trabajo de una maquina a través de internet.” 2018.
- [14] C. E. J. Ruiz, F. A. P. León, M. A. R. Blanco, I. Duran, and J. L. V. Ávila, “Automatización De Un Banco De Pruebas Para Extracción De Parámetros De Un Motor De Inducción (Automated Test Bench To Extract the Parameters of an Induction Motor),” *Pist. Educ.*, vol. 40, no. 130, pp. 638–655, 2018, [Online]. Available: <http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/1712>.

- [15] C. A. Villavicencio, "Diseño de un sistema de monitoreo y alerta de temperatura para motores industriales utilizando tecnología ZIGBEE," 2019.
- [16] R. Emanuel and J. Daniel, "Monitoreo , detección y diagnóstico de fallos en motores asíncronos trifásicos," 2020.
- [17] K. Mykoniatis, "A Real-Time Real-Time Condition for Low Low Voltage Voltage Industrial Industrial Motors Motors Using Using Internet-of-Things," *Procedia Manuf.*, vol. 42, no. 2019, pp. 450–456, 2020, doi: 10.1016/j.promfg.2020.02.050.
- [18] A. Lekbich and J. Boukherouaa, "A secure wireless control of Remote Terminal Unit using the Internet of Things in smart grids," *2018 6th Int. Conf. Wirel. Networks Mob. Commun.*, pp. 1–6, 2018.
- [19] H. Barksdale, Q. Smith, and M. Khan, "Condition Monitoring of Electrical Machines with Internet of Things," *SoutheastCon 2018*, pp. 1–4, 2018.
- [20] M. Ali, "Data Acquisition System & Real Time Monitoring of the Parameters of Induction Motor via Wireless Communication.," no. Ieec, pp. 1–6, 2020.
- [21] Z. Protocol and M. Gui, "Induction Motor Parameter Monitoring System using Zig bee Protocol & MATLAB GUI," *2018 Fourth Int. Conf. Adv. Electr. Electron. Information, Commun. Bio-Informatics*, pp. 1–6, 2018.
- [22] L. Magadán, F. J. Suárez, J. C. Granda, and D. F. García, "Low-cost real-time monitoring of electric motors for the Industry 4.0," *Procedia Manuf.*, vol. 42, no. 2019, pp. 393–398, 2020, doi: 10.1016/j.promfg.2020.02.057.
- [23] T. A. Khan, F. A. Shaikh, S. Khan, and M. F. Siddiqui, "Real-Time Wireless Monitoring for Three Phase Motors in Industry: A Cost-Effective Solution using IoT," *2019 IEEE 6th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2019*, no. August, pp. 1–5, 2019, doi: 10.1109/ICSIMA47653.2019.9057343.
- [24] N. Ullah, "ZigBee-based Parameter Monitoring and Controlling Scheme for Multiple DC Motors," *Indian J. Sci. Technol.*, vol. 13, no. 06, pp. 725–734, 2020, doi: 10.17485/ijst/2020/v13i06/149098.
- [25] I. Eléctrica, J. David, C. Bolívar, L. Fernando, and O. Tirano, "Estimación de los parámetros del generador síncrono del módulo TNA a partir de oscilogramas eléctricos," 2020.
- [26] Gandy Larco Montenegro, "Sistema de monitoreo de grupos electrógenos," 2019.
- [27] "Internet de las Cosas - Moisés Barrio Andrés - Google Libros." <https://books.google.com.co/books?id=jF-LDwAAQBAJ&lpg=PA1&dq=internet de las cosas&pg=PA3#v=onepage&q=internet de las cosas&f=false> (accessed Oct. 13, 2021).
- [28] C. Ballesteros de Andrés and D. Sypko, "Sistema IoT para monitorización del estado de un centro de proceso de datos de grandes dimensiones," 2017.
- [29] Node-red, "Node-red." <https://nodered.org/> (accessed May 15, 2023).
- [30] "MariaDB Foundation - MariaDB.org." <https://mariadb.org/> (accessed May 15, 2023).
- [31] SAP Business, "Manual del usuario de Dashboards y Presentation Design," pp. 1–312, 2015.
- [32] GitHub, "GitHub - Node-red/Node-red-dashboard: A dashboard UI for Node-red." <https://github.com/Node-red/Node-red-dashboard> (accessed May 15,

- 2023).
- [33] J. Luis, A. Alarcón, N. Fiorella, and A. Santisteban, *Sistemas embebidos guía metodológica para su desarrollo*. 2020.
 - [34] J. Salazar, *Redes Inalámbricas*, vol. 2. 2012.
 - [35] J. M. Hurtado, "Introducción a las Redes de Comunicación Industrial," p. 19, 2017, [Online]. Available: http://www.infoplcn.net/files/documentacion/comunicaciones/infoPLC_net_introduccion3b3n-a-las-redes-de-comunicacion3b3n-industrial.pdf.
 - [36] M. Castelli Lemez and M. Andrade, "Metodología de monitoreo, detección y diagnóstico de fallos en motores asíncronos de inducción," *Mem. Trab. Difusión Científica y Técnica*, no. 5, pp. 65–76, 2007.
 - [37] "EasyLogic PM2120." <https://www.se.com/co/es/download/document/NHA2779005-00/> (accessed Jun. 09, 2022).
 - [38] "FrontPage - Raspbian." <https://www.raspbian.org/> (accessed Jun. 09, 2022).
 - [39] "Node-red-contrib-smooth." <https://flows.nodered.org/node/Node-red-contrib-smooth> (accessed May 15, 2023).
 - [40] "Node-red-contrib-modbus." <https://flows.nodered.org/node/Node-red-contrib-modbus> (accessed May 15, 2023).
 - [41] "Node-red-dashboard." <https://flows.nodered.org/node/Node-red-dashboard> (accessed May 15, 2023).
 - [42] "Node-red-contrib-ui-chart." <https://flows.nodered.org/node/Node-red-contrib-ui-chart> (accessed May 15, 2023).
 - [43] L.- Node-red, "Node-red-contrib-uibuilder." <https://flows.nodered.org/node/Node-red-contrib-uibuilder> (accessed May 15, 2023).
 - [44]. nodered.org, "Node-red-contrib-ui-table." <https://flows.nodered.org/node/Node-red-contrib-ui-table> (accessed May 15, 2023).