



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL
MONITOREO ENERGÉTICO EN ESTACIONES DE BOMBEO EN EL
SECTOR CAMARONERO MEDIANTE TELEMETRÍA.**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica

AUTORES: JUAN DIEGO FLORES SOLÓRZANO
JOHN ANDRÉS COELLO JARA

TUTOR: ING. VÍCTOR DAVID LARCO TORRES, MSc

Guayaquil – Ecuador
2025 – 2026

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, Juan Diego Flores Solórzano con documento de identificación N° 0932581366
y John Andrés Coello Jara con documento de identificación N° 0940812985;
manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de
lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de
manera total o parcial el presente trabajo de titulación.

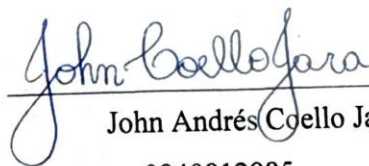
Guayaquil, 15 de enero del año 2026

Atentamente,



Juan Diego Flores Solórzano

0932581366



John Andrés Coello Jara

0940812985

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Juan Diego Flores Solórzano con documento de identificación N° 0932581366 y John Andrés Coello Jara con documento de identificación N° 0940812985, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño e implementación de un prototipo para el monitoreo energético en estaciones de bombeo en el sector camaronero mediante telemetría”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 15 de enero del año 2026.

Atentamente,



Juan Diego Flores Solórzano
0932581366



John Andrés Coello Jara
0940812985

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Víctor Larco con documento de identificación N° 0923270136, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Diseño e implementación de un prototipo para el monitoreo energético en estaciones de bombeo en el sector camaronero mediante telemetría realizado por Juan Diego Flores Solórzano con documento de identificación N° 0932581366 y John Andrés Coello Jara con documento de identificación N° 0940812985, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 15 de enero del año 2026.

Atentamente,



Ing. Víctor Larco, MSc

0923270136

DEDICATORIA

A Dios, por sostenerme siempre, por darme fuerza en cada prueba y recordarme que no debo bajar la guardia.

A mis padres, por su amor, sacrificio y ejemplo. Quiero que sientan con orgullo que tienen un hijo que siempre verá por ustedes.

A mi novia, por su paciencia y su fe en mí; gracias por acompañarme en estos meses duros, cuando pensé rendirme, y por ayudarme a seguir.

A mis abuelos, por la sangre y el pulso firme que me heredaron; por enseñarme que escribir y servir es una misma misión. Que este trabajo sea también una manera de llevar ayuda, de corazón, a quien la necesite.

A todas las personas que, con un consejo de vida durante mis años como estudiante, dejaron luz en mi camino: sus palabras habitan estas páginas.

Con gratitud eterna,

John Andrés Coello Jara

DEDICATORIA

Quiero dedicar este trabajo a mis padres, por su apoyo incondicional, su esfuerzo y dedicación, sus enseñanzas, sus consejos y hacerme sentir respaldado en todo momento en cada uno de los procesos de mi vida, me siento orgulloso de tenerlos y sobre todo se sientan orgullosos de que todo lo que consiga siempre será para y por ustedes.

A mi abuela y a toda mi familia, por estar siempre presentes en todo ámbito y dificultad que atravesé.

A las personas que pasaron por mi vida y las que están presentes aun acompañándome en este proceso de estudiante y sobre todo como amigo viviendo experiencias inolvidables que siempre recordaré.

A Dios, por guiarme en este camino, dame esa luz y fuerza para poder atravesar cualquier obstáculo que se aparezca y demostrarme que con esfuerzo y dedicación todo es posible.

Con gratitud eterna

Juan Diego Flores Solórzano

AGRADECIMIENTO

A Dios, por su guía constante, por darme fuerza en cada jornada y recordarme que la perseverancia abre caminos.

A mis padres, por su amor incondicional y su ejemplo diario; deseo que se sientan orgullosos de este logro y seguros de que siempre veré por ustedes.

A mi novia, por su apoyo firme y paciente en los meses más duros, por sostenerme cuando quise bajar la guardia y animarme a continuar.

A mis abuelos, por el carácter y la templanza que me heredaron; por enseñarme que escribir y servir es también una forma de amar a los demás.

A quienes me regalaron un consejo de vida durante mis años como estudiante, compañeros y amigos que caminaron conmigo y me tendieron la mano cuando fue necesario.

A mi tutor Víctor Larco, por su orientación académica y su confianza en este trabajo; y a los docentes y autoridades de UPS, por su formación exigente y humana.

John Andrés Coello Jara

AGRADECIMIENTO

Mi gratitud infinita es para mis padres. Gracias por ser el pilar fundamental de mi vida, por sus enseñanzas y por el respaldo inquebrantable que me brindaron en cada una de las etapas de mi vida.

Este logro es el fruto de su esfuerzo y sacrificio, me llena de orgullo ser su hijo y mi mayor deseo es que este triunfo sea una recompensa a su entrega.

A mi abuela y a toda mi familia, gracias por su apoyo constante. Su presencia en los momentos difíciles y su compañía en las alegrías han sido un refugio indispensable para mí; gracias por creer siempre en mi capacidad.

A mi tutor Víctor Larco, a mis compañeros, amigos y las personas que pasaron por mi vida. Agradezco a quienes siguen a mi lado y a quienes han dejado una huella, gracias por esas experiencias inolvidables, por compartir el estudio y por hacer de este proceso una etapa memorable.

A Dios por ser mi guía y fortaleza en cada paso de este camino. Gracias por iluminar mi mente en los momentos de incertidumbre y por darme la resiliencia necesaria para superar los obstáculos, demostrándome que, con fe y perseverancia, todo propósito es alcanzable.

Juan Diego Flores Solórzano

RESUMEN

AÑO	ALUMNOS	DIRECTOR DEL PROYECTO	TEMA DE TRABAJO DE TITULACIÓN
2026	JUAN DIEGO FLORES SOLÓRZANO JOHN ANDRÉS COELLO JARA	ING. VÍCTOR DAVID LARCO TORRES, MSc	DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL MONITOREO ENERGÉTICO EN ESTACIONES DE BOMBEO EN EL SECTOR CAMARONERO MEDIANTE TELEMETRÍA.

En este Trabajo de Titulación se diseña e implementa un prototipo de monitoreo energético para estaciones de bombeo en el sector camaronero utilizando técnicas de telemetría. El alto consumo eléctrico dentro de los procesos de recambio de agua y aireación representa uno de los mayores costos operativos en la industria acuícola, esto se ve gravemente afectado por la ubicación remota de las estaciones, y la dificultad para poder supervisar estas situaciones de variables eléctricas en proceso. La falta de acceso en tiempo real a los datos dificulta la detección de ineficiencias que se presenten, por ello se han realizado estudios dentro del sector, el cual indica que aproximadamente el 30% de la energía que se consume en estas instalaciones, se desperdicia en su mayoría debido a las sobrecargas operativas o fallas críticas que resultan en paradas del proceso no programadas, donde también, se utiliza de manera ineficiente los recursos.

La telemetría industrial y el uso de controladores lógicos programables como el PLC, de gama avanzada como el Siemens S7-1500 permiten superar la barrera de la distancia,

facilitando la transmisión de datos desde el campo hasta centros de control o dispositivos remotos. La implementación de esta arquitectura permite una adquisición continua de ciertos parámetros indicados por el usuario, en este caso serían parámetros eléctricos, también pueden ser como el cálculo de indicadores de consumo, y una visualización del estado en el que están operando las bombas, sin necesidad de estar físicamente.

El desarrollo del prototipo se divide en varias etapas que incluyen: la investigación de protocolos de comunicación industrial, la integración de medidores de energía y variadores de frecuencia bajo una red PROFINET, la programación del PLC para el procesamiento de señales y el diseño de una interfaz gráfica para la visualización telemétrica de los perfiles de carga.

Palabras Clave: Telemetría, Eficiencia Energética, PLC S7-1500, Sistemas de Bombeo, Monitoreo Remoto.

ABSTRACT

YEAR	STUDENTS	PRJ. DIRECTOR	SUBJECT
2026	JUAN DIEGO FLORES SOLÓRZANO JOHN ANDRÉS COELLO JARA	ING. VÍCTOR DAVID LARCO TORRES, MSc	DESIGN AND IMPLEMENTATION OF A PROTOTYPE FOR ENERGY MONITORING IN PUMPING STATIONS IN THE SHRIMP FARMING SECTOR USING TELEMETRY

This thesis designs and implements an energy monitoring prototype for pumping stations in the shrimp farming sector using telemetry techniques. High electricity consumption in water replacement and aeration processes represents one of the highest operating costs in the aquaculture industry, a problem that is exacerbated by the remote location of the stations and the difficulty of monitoring electrical variables on site. The lack of access to real-time data prevents the timely detection of inefficiencies. Studies in the sector indicate that approximately 30% of the energy consumed in these facilities is wasted due to operational inefficiencies, overloads, and critical failures, resulting in unscheduled downtime and inefficient use of resources.

Industrial telemetry and the use of advanced programmable logic controllers (PLCs) such as the Siemens S7-1500 overcome the barrier of distance, facilitating the transmission of data from the field to control centers or remote devices. The implementation of this architecture allows for the continuous acquisition of electrical parameters, the calculation of consumption indicators, and the visualization of the operational status of the pumps without the need for physical presence at the station.

The development of the prototype is divided into several stages, including: research into industrial communication protocols, the integration of energy meters and frequency converters, and the design of the control system.

Key words: Telemetry, Energy Efficiency, PLC S7-1500, Pumping Systems, Remote Monitoring.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
2. PROBLEMA	2
2.1 IMPORTANCIA DE LA PROBLEMÁTICA	4
3. OBJETIVOS	5
3.1. OBJETIVO GENERAL.	5
3.2. OBJETIVOS ESPECÍFICOS.	5
4. FUNDAMENTOS TEÓRICOS	6
4.1. PLC	6
4.1.1 PLC Siemens Modelo: S7-1500	6
4.2 HMI	7
4.2.1 Pantalla HMI Modelo: KTP 700	8
4.3 Lenguaje de programación Ladder o escalera	9
4.4 Motores eléctricos	10
4.5 Variador de Frecuencia G120x	11
4.6 Protocolo de comunicación Modbus	12
4.7 Tecnologías de transmisión Inalámbrica de largo alcance	13
4.8 Módulos de desarrollo y Nodos IoT	14
4.9 Protocolos de comunicación Industrial y redes de datos	15
4.9.1 Protocolo Profinet IO	16
4.9.2 Protocolo Modbus TCP/IP	17
4.10 Software TIA Portal V18	18
4.11 Funcionalidad del servidor Modbus en siemens (MB_SERVER)	19
4.12 Telegrama Estándar 20 (Standard Telegram 20)	20
4.13 Bloque de función SINA SPEED	21
4.14 Heltec Wifi Lora 32 (V3)	22
4.15 Fundamentos de la Modulación Lora	23
4.16 Arquitectura de comunicación punto a punto	24
4.17 Software Arduino IDE	25
5. MARCO METODOLÓGICO	26
5.1 Descripción del proyecto	26
5.2 Tipo de metodología	27

5.3	<i>Etapas del prototipo</i>	27
5.3.1	<i>Etapa 1: Programación del PLC y Adquisición de datos</i>	27
5.3.2	<i>Etapa 2: Interfaz HMI</i>	37
5.3.3	<i>Etapa 3: Arquitectura de Interconexión y Protocolo Modbus TCP/IP</i>	43
6.	<i>RESULTADOS</i>	49
6.1	<i>Programación del PLC</i>	49
6.2	<i>Interfaz HMI</i>	53
6.3	<i>Comunicación Heltec</i>	55
6.4	<i>Dashboard</i>	57
6.5	<i>Validación experimental y comparativa de datos</i>	59
6.6	<i>Implementación</i>	60
7.	<i>CRONOGRAMA DE ACTIVIDADES</i>	64
8.	<i>PRESUPUESTO</i>	65
9.	<i>CONCLUSIONES</i>	66
10.	<i>RECOMENDACIONES</i>	67
11.	<i>REFERENCIAS BIBLIOGRÁFICAS</i>	68
12.	<i>ANEXOS</i>	70

ÍNDICE DE FIGURAS

Figura 1 PLC S7-1500.....	6
Figura 2 Diseño de pantalla HMI	7
Figura 3 Pantalla HMI KTP 700	8
Figura 4 Programación Ladder	9
Figura 5 Motor eléctrico inducción	10
Figura 6 Variador de Frecuencia G120X.....	11
Figura 7 Protocolo de comunicación Modbus	12
Figura 8 Tecnologías de transmisión de largo alcance	13
Figura 9 Módulo de desarrollo y Nodos Iot	14
Figura 10 Protocolo de comunicación industriales	15
Figura 11 Protocolo Profinet Io	16
Figura 12 Modbus TCP/IP	17
Figura 13 Tia Portal V18.....	18
Figura 14 Bloque MB_SERVER	19
Figura 15 Telegrama 20.....	20
Figura 16 Bloque SINA SPEED TLG20	21
Figura 17 Diagrama de imagen de microcontrolador Heltec V3.....	22
Figura 18 Modulación de comunicación Lora	23
Figura 19 Arquitectura de comunicación punto a punto	24
Figura 20 Software Arduino.....	25
Figura 21 Diagrama de proceso de comunicación Modbus TCP/IP.....	26
Figura 22 Creación de nuevo proyecto TIA Portal.....	28
Figura 23 Dirección IP del PLC S71500.....	28
Figura 24 Selección de telegrama 20 en la configuración del dispositivo.....	29
Figura 25 Bloque de programación SINA SPEED TLG20	30
Figura 26 Bloque de programación SINA SPEED TLG20	31
Figura 27 Lógica Sencilla en Ladder.....	31
Figura 28 Imagen del DB config para programar bloque MB_SERVER	32
Figura 29 Bloque de datos	33
Figura 30 Configuración del puntero para el db data esp32.....	33
Figura 31 Lógica de transferencia de datos con bloque MOVE	34
Figura 32 Verificación de variables y de DB data esp32	35
Figura 33 Bloque de Programación CALCULATE.....	35
Figura 34 Variables y referencias del bloque CALCULATE	36
Figura 35 Interfaz HMI	37
Figura 36 Interfaz HMI con cada uno de los elementos	37
Figura 37 I/O fields en la interfaz HMI.....	38
Figura 38 Gráfica implementada dentro de la interfaz HMI.....	38
Figura 39 Variable colocada en el botón Start	39
Figura 40 Variable colocada en el botón Stop	39
Figura 41 Interfaz HMI con gráfica	40
Figura 42 Programación de las validaciones	40
Figura 43 Interfaz HMI prueba 1	41
Figura 44 Imágenes, márgenes y mejoras HMI.....	41
Figura 45 Pantalla e interfaz HMI.....	42
Figura 46 Diagrama de flujo proceso de comunicación.....	43

Figura 47 Vista Frontal del variador G120 Siemens.....	44
Figura 48 Router TP Link.....	45
Figura 49 Punto del sistema Inalámbrico.....	45
Figura 50 Lista del DB ESP32.....	46
Figura 51 Código y registros encapsulados.....	46
Figura 52 Diagrama de flujo de receptor.....	47
Figura 53 Dashboard de la página THING SPEAK.....	48
Figura 54 Arranque simple ladder.....	49
Figura 55 Funcionamiento del bloque SINA SPEED.....	50
Figura 56 Funcionamiento del bloque SINA SPEED.....	50
Figura 57 Bloque de datos en funcionamiento.....	51
Figura 58 Bloque MOVE en funcionamiento.....	51
Figura 59 Bloque MB SERVER en funcionamiento.....	52
Figura 60 Interfaz completa del HMI.....	53
Figura 61 HMI con un SP 1200.....	53
Figura 62 Prueba HMI con un valor de 1615 rpm.....	54
Figura 63 Tx Heltec V3.....	55
Figura 64 Rx Heltec V3.....	55
Figura 65 Tx Heltec V3 y Rx Heltec V3.....	56
Figura 66 Voltaje con un valor de 1200 rpm.....	57
Figura 67 Corriente con un valor de 1200 rpm.....	57
Figura 68 Potencia con un valor de 1200 rpm.....	58
Figura 69 Voltaje con un valor de 1200 rpm.....	58
Figura 70 Comparación de registros.....	59
Figura 71 Implementación en el módulo de prácticas.....	60
Figura 72 Parametrización del variador.....	61
Figura 73 Programación del TIA Portal.....	62
Figura 74 Datos del DB y HMI.....	63
Figura 75 Vista de Heltec cargada la programación.....	63
Figura 76 Conexiones de los módulos y elementos a utilizar.....	70

ÍNDICE DE TABLAS

Tabla 1 Cronograma de Actividades	64
Tabla 2 Presupuesto	65

1. INTRODUCCIÓN

En el sector industrial acuícola, las estaciones de bombeo son muy importantes para diversos procesos productivos, el cual garantiza la circulación de agua y el oxígeno en los estanques de cultivo. Por ello, es necesario que su operación cumpla con altos estándares de eficiencia energética para evitar costos operativos excesivos y asegurar que no se acorte la vida útil de la maquinaria electromecánica por falta de supervisión.

La etapa de monitoreo es fundamental en el funcionamiento del sistema, ya que este permite controlar la velocidad y el consumo del motor mediante el uso de tecnología de variación de frecuencia, la cual optimiza la conversión de energía eléctrica a mecánica. Al incorporar sistemas de microcontroladores y comunicación inalámbrica, se permite expandir las capacidades del control industrial tradicional, es decir, brinda una transmisión de datos telemétrica más accesible y versátil hacia diferentes puntos remotos.

Este Trabajo de Titulación tiene como objetivo principal diseñar e implementar un prototipo de monitoreo energético utilizando un PLC S7-1500, una pantalla HMI y dispositivos de comunicación como un router TP-Link para la gestión de red. Con este propósito se abordan aspectos como la configuración del variador G120, la programación de adquisición de datos en el PLC, su visualización local en la HMI y la transmisión inalámbrica de la información mediante los módulos Heltec.

La finalidad de este trabajo es mostrar la integración de tecnologías de automatización robusta con soluciones IoT y las ventajas que esta convergencia ofrece. El usar equipos industriales junto a microcontroladores en un proceso de control real ayuda a los estudiantes a tener una visión más técnica de la telemetría moderna y conocer el funcionamiento práctico de estos sistemas híbridos.

2. PROBLEMA

El cultivo de camarón se convirtió en una de las actividades acuícolas de mayor crecimiento en el mundo durante las últimas décadas, registrando una tasa de crecimiento anual promedio del 5.3% que contrasta con el estancamiento de la pesca tradicional. Este auge, impulsado por la tecnificación de sistemas de engorde en ambientes controlados y la expansión de la demanda, ha posicionado al camarón como el producto pesquero más valioso del comercio internacional, representando cerca del 16% del valor total de las exportaciones mundiales de productos del mar (FAO, 2024).

A nivel nacional, el Ecuador se consolidó como uno de los protagonistas de esta dinámica debido a sus condiciones costeras favorables y a la tecnificación intensiva de sus procesos productivos. Las exportaciones de camarón para el año 2021 superaron los USD 5 300 millones FOB, cifra que evidencia la magnitud del sector dentro de la economía nacional. Entre 2018 y 2024, Ecuador alcanzó exportaciones de camarón por valor estimado en USD 4 616,73 millones, situándose entre los cinco mayores exportadores mundiales (Panorama acuicola, 2024).

Este crecimiento acelerado posicionó al país como un actor clave en la exportación de camarón, convirtiendo esta actividad en una fuente relevante de divisas y empleo hasta la fecha de hoy (Velázquez, 2024).

Dada su ubicación estratégica costera, la infraestructura camaronera incluyendo estaciones de bombeo, estanques de cultivo y sistemas de alimentación ha acompañado el crecimiento de la industria. No obstante, dicho crecimiento ha generado también desafíos relacionados con la gestión energética, los costos operativos y la eficiencia productiva. En el año 2023 el sector camaronero enfrentó una caída de las exportaciones del 6 % con pérdidas superiores a USD 1 500 millones, lo que refleja la vulnerabilidad de la industria frente al entorno global y la necesidad de mejoras de eficiencia (Ecuador, CNA, 2024).

Debido a dicho crecimiento, los márgenes de rentabilidad se estrechan, optimizar el consumo eléctrico y el control de estaciones de bombeo se presenta como un factor clave para sostener la competitividad de los productores. En el sector camaronero, el

bombeo es un proceso industrial que depende estrictamente de variables eléctricas y de control, como el voltaje y la corriente eléctrica consumida por el motor de la bomba; esta vinculación técnica es la razón por la cual la ausencia de monitoreo deriva en situaciones graves.

Al no supervisar estos parámetros en tiempo real, los equipos trabajan forzados sin que el operador lo note, provocando que cerca del 75% de las fincas operen con una eficiencia energética real de apenas el 50%. Este problema operativo ante las variables eléctricas genera sobrecostos por desperdicio de energía y desgaste de maquinaria que pueden superar los USD 70 000 anuales en una piscina de camarón promedio de 100 hectáreas, afectando directamente la sostenibilidad del negocio (Alliance, 2023).

Los autores realizaron, en la Universidad Politécnica Salesiana sede Guayaquil, una inspección o verificación en los laboratorios de la carrera, donde se evidenció que no se cuenta con un prototipo que permita analizar este proceso, y así permita a los estudiantes poder estudiar, analizar, y comprender como funciona este procedimiento del monitoreo de un sistema energético a través de la telemetría. De una estación de bombeo para una piscina camaronera.

2.1 IMPORTANCIA DE LA PROBLEMÁTICA

El sector camaronero se constituye como un pilar fundamental de la economía no petrolera del Ecuador. Sin embargo, la contracción de exportaciones del 6% y las pérdidas millonarias registradas en 2023 evidencian que el modelo de crecimiento por expansión se está agotando, siendo importante cambiar hacia un modelo de crecimiento por eficiencia. La implementación de sistemas de monitoreo energético resulta crítica para proteger los márgenes de rentabilidad, dado que se podría cambiar la ineficiencia del 50% en las estaciones de bombeo y evitar pérdidas operativas significativas es vital para mantener la competitividad del camarón ecuatoriano en el mercado internacional (Ecuador, CNA, 2024).

En el ámbito técnico, la modernización de la infraestructura mediante la telemetría es indispensable. Actualmente, el control manual dentro de este tipo de procesos y la dependencia de este, más la ausencia de la supervisión de dichas variables eléctricas, generan una ceguera operativa, que deriva en el desgaste prematuro de la maquinaria o fallas en el proceso. La propuesta de este trabajo plantea un cambio de registros hacia el monitoreo basado en la condición, utilizando la tecnología de controladores lógicos programables para visibilizar el comportamiento real de los elementos ubicados en zonas remotas, optimizando así la gestión de mantenimiento y operación.

Finalmente, en el ámbito académico, el desarrollo de este proyecto posee una relevancia institucional para la Universidad Politécnica Salesiana. Las inspecciones realizadas en los laboratorios de la carrera de Electrónica y Automatización evidenciaron la carencia de bancos de pruebas que simulen este proceso industrial específico, La implementación de este prototipo proporcionará una plataforma didáctica capaz de validar tecnologías de telemetría en escenarios industriales.

3. OBJETIVOS

3.1. OBJETIVO GENERAL.

Implementar un prototipo para el monitoreo energético en estaciones de bombeo en el sector camaronero mediante telemetría.

3.2. OBJETIVOS ESPECÍFICOS.

- Realizar la programación del controlador lógico programable de la etapa de una estación de bombeo en una piscina acuícola.
- Diseñar las interfaces gráficas de operación en el panel HMI permitiendo la visualización de los parámetros y consumo en una estación de bombeo.
- Establecer el protocolo de comunicación entre la unidad de procesamiento y el módulo de telemetría.

4. FUNDAMENTOS TEÓRICOS

4.1. PLC

Un PLC se puede definir como una computadora o autómatas programables, diseñado para trabajar en ambientes industriales. Al poseer una memoria programable se puede almacenar diferentes instrucciones para el control lógico o secuencial, de esta manera se puede manejar máquinas y procesos utilizando las entradas y salidas que ofrece el PLC las cuales pueden ser analógicas o digitales. Su funcionamiento consiste en un bucle repetitivo, que consiste en leer las entradas, la ejecución del programa y por último actualizar las salidas (Elecproy, 2024).

4.1.1 PLC Siemens Modelo: S7-1500

El controlador lógico programable de la serie SIMATIC S7-1500 es la última generación estándar propuesta por Siemens para aplicaciones de automatización de alta complejidad. La arquitectura modular del dispositivo y la capacidad de ejecutar instrucciones de bits en menos de 1 ns aseguran su operación determinista en tareas críticas (Elecproy, 2024).

La pantalla frontal, integrada en la CPU, como se puede ver en la figura 1, tiene la ventaja de que el personal técnico puede ver los estados operativos, diagnósticos de fallos y direcciones IP en el hardware sin la necesidad de conectar una computadora externa. Desde el punto de vista de la conectividad, este autómatas gestiona de forma nativa el estándar PROFINET, facilitando su integración en redes industriales avanzadas bajo el entorno de ingeniería TIA Portal (Siemens, 2025).

Figura 1

PLC S7-1500



Nota. Se caracteriza por su velocidad de procesamiento y sus capacidades Profinet integradas. Tomado de (Siemens, 2025).

4.2 HMI

El HMI es una interfaz Hombre-Máquina, es una herramienta muy utilizada por el usuario, dentro del ámbito industrial, este permite controlar el proceso automatizado, ya que posee una pantalla la cual, se puede visualizar los datos del sistema en tiempo real y como se observa en la figura 2, además de que el usuario también pueda tomar acciones directamente desde la pantalla, al incluir botones, indicadores, o alarmas (Logicbus, 2024).

Figura 2

Diseño de pantalla HMI



Nota. Diseños de pantallas HMI donde se observa el proceso que está realizando. Tomado de (Ignition, 2025).

4.2.1 Pantalla HMI Modelo: KTP 700

Para la interfaz de operación local se ha seleccionado el panel SIMATIC HMI KTP700 Basic, un equipo con una pantalla táctil de 7 pulgadas de alta resolución, que permite reemplazar los tableros de mando convencionales, además de ser un dispositivo de visualización robusta lo cual se aprecia en la figura 3. Está diseñado específicamente para tareas de monitoreo al pie de la máquina, su comunicación se realiza a través de PROFINET, actuando como un cliente que interactúa directamente con las variables del PLC para la gestión de alarmas y recetas en tiempo real (Siemens, 2023).

Figura 3

Pantalla HMI KTP 700



Nota. Pantalla táctil de 7 pulgadas de la serie Basic Panel de Siemens, ideal para tareas de control y visualización. Tomado de (Siemens, 2023).

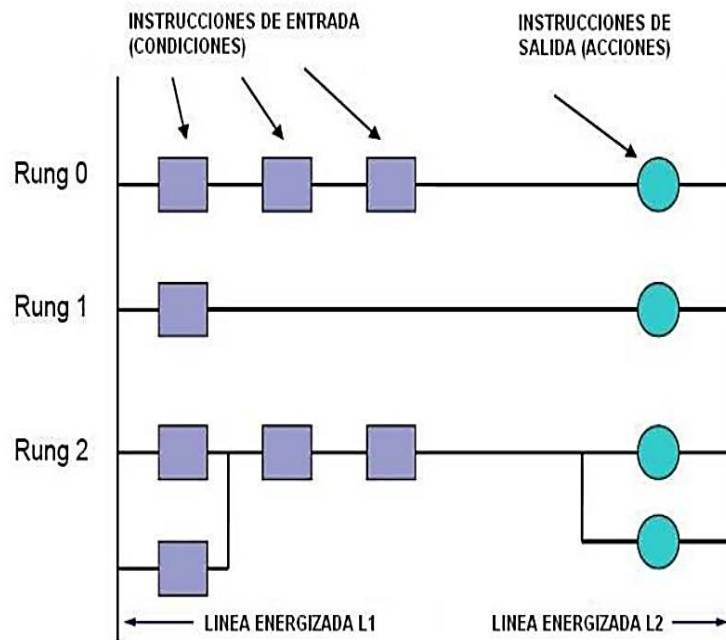
4.3 Lenguaje de programación Ladder o escalera

Es uno de los lenguajes de programación más utilizados para la programación de PLCs, por medio de símbolos gráficos y en forma de escalera, cada segmento tiene su propia lógica, consta de dos rieles encargados de simbolizar la alimentación, los cuales son de forma vertical y líneas horizontales que simbolizan los escalones en los cuales se insertan los contactos eléctricos y se va formando la secuencia lógica (Alvarado, 2022).

Como se observa en la Figura 4 la programación Ladder o escalera se inicia de izquierda a derecha, en el lado izquierdo se colocan las entradas las cuales se pueden utilizar ya sean en serie o paralelo para establecer una conexión (Alvarado, 2022).

Figura 4

Programación Ladder



Nota. Programación Ladder o también llamada programación en escalera. Tomado de (Alvarado, 2022).

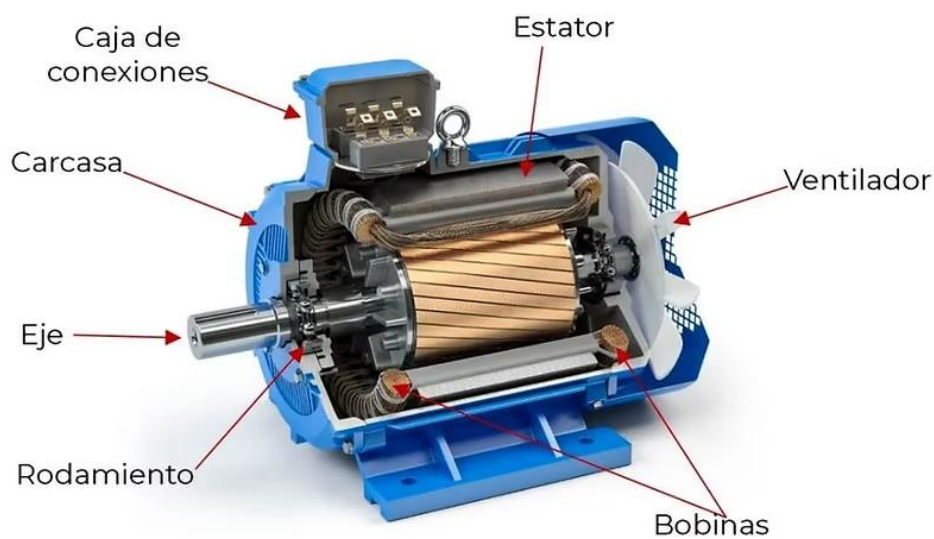
4.4 Motores eléctricos

La función de los motores es convertir la energía eléctrica en energía mecánica, basándose en la Ley de Lorentz, esta ley indica que, al existir una corriente eléctrica a la cual se aplica una fuerza estando dentro del campo magnético, ocasiona el movimiento rotatorio. Estos motores funcionan tanto con corriente continua como con corriente alterna (Promelsa, 2023).

Un motor de inducción como se puede observar en la figura 5 es un tipo de motor de corriente AC asincrónico. El cual está compuesto por dos elementos importantes, tanto como el estator, y el rotor, el estator encargado del paso de la corriente AC, para generar un campo magnético giratorio, y el rotor el cual produce la fuerza magnética (Aude, 2024).

Figura 5

Motor eléctrico de inducción



Nota. Partes de un motor de inducción de corriente alterna, sus elementos principales el estator y el rotor. Tomado de (Aude, 2024).

4.5 Variador de Frecuencia G120x

Es un convertidor de frecuencia modular desarrollado por Siemens específicamente para aplicaciones de infraestructura en los sectores de agua y aguas residuales. Este equipo está optimizado para el control de bombas y ventiladores, destacándose por su capacidad de integración nativa en redes industriales a través de protocolos como PROFINET, lo cual facilita la comunicación directa con el controlador S7-1500 sin requerir módulos de interfaz adicionales. Su diseño robusto otorga una alta resistencia frente a ambientes agresivos (Siemens, 2025).

El dispositivo opera en un rango de tensión trifásica de 380 V a 480 V AC y cubre un espectro de potencias desde 0.75 kW hasta 630 kW, siendo compatible con motores de inducción y síncronos de reluctancia con una eficiencia superior al 98%. Como se puede observar en la figura 6, el variador dispone de un panel de operador inteligente y terminales de conexión accesibles que facilitan su puesta en marcha (Siemens, 2025).

Figura 6

Variador de Frecuencia G120x



Nota. Vista frontal del variador SINAMICS G120X con puertos integrados para comunicación PROFINET. Tomado de (Siemens, 2025).

4.6 Protocolo de comunicación Modbus

La comunicación entre los equipos de campo y el controlador se basa en el protocolo Modbus, es un protocolo estándar en la comunicación industrial.

Como se puede observar en la figura 7. Su arquitectura se basa en el modelo Maestro-Esclavo, donde el PLC, se encarga de gestionar el intercambio de los datos, solicitando la lectura de cada uno de los registros específicos en cada uno de los elementos que se encuentren dentro del proceso. La versatilidad de este estándar permite que la estructura del mensaje se mantenga intacta independientemente del medio físico, ya sea a través de cables seriales RS-485 o mediante redes Ethernet modernas (Aula21, 2025).

Figura 7

Protocolo de Comunicación Modbus



Nota. Arquitectura de comunicación Modbus. Este protocolo estándar opera bajo un modelo Maestro-Esclavo. Tomado de (Aula21, 2025).

4.7 Tecnologías de transmisión Inalámbrica de largo alcance

Las tecnologías de comunicación inalámbrica relacionadas a la telemetría se basan en la modulación de señales de radiofrecuencia para así poder establecer datos a grandes distancias, con un bajo consumo de energía. Lo cual lo diferencia de las redes de alto ancho de banda como el Wifi, ya que estos sistemas utilizan esquemas de modulación de espectro ensanchado, como los CSS, que distribuyen la señal en un ancho de banda mucho mayor, para así se aumente la resistencia, en contra las interferencias y el ruido que se puede generar (Digikey, 2025).

Para establecer enlaces de datos en entornos donde el cableado físico es inviable, se recurre a tecnologías de comunicación inalámbrica orientadas a la telemetría, cuya diversidad de aplicaciones se ilustra en la figura 8 (Digikey, 2025).

Figura 8

Tecnologías de transmisión de largo alcance.



Nota. Esquema de transmisión inalámbrica de largo alcance. Tomado de (Digikey, 2025).

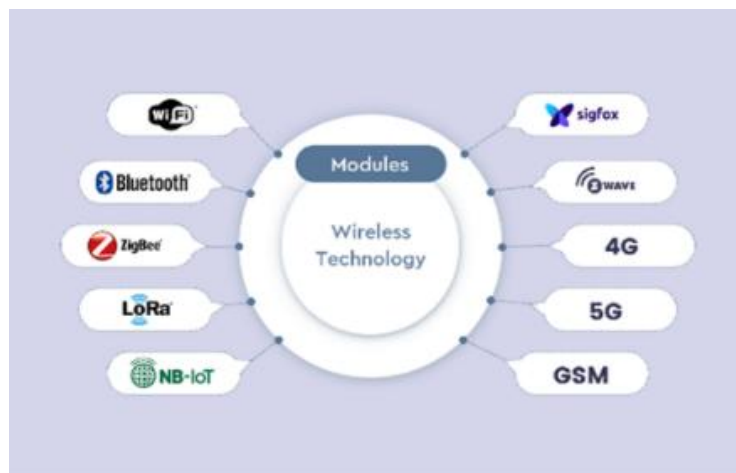
4.8 Módulos de desarrollo y Nodos IoT

En el contexto de la electrónica embebida, los nodos de desarrollo para el Internet de las Cosas son plataformas de hardware los cuales integran un solo circuito esquemático impreso, un microcontrolador y los respectivos periféricos de comunicación. Estos poseen un chip de 32 bits, que combinan la capacidad de procesamiento y transceptores de radiofrecuencia (Smart, 2024).

Como se logra observar en la figura 9, la convergencia entre la automatización industrial y la digitalización se materializa a través de plataformas de hardware como nodos IoT. Estos dispositivos integran en un solo circuito impreso un microcontrolador de bajo consumo y transceptores de comunicación, actuando como pasarelas ágiles que tienen la capacidad de gestionar la transmisión de datos hacia la interfaz de supervisión (Smart, 2024).

Figura 9

Módulo de desarrollo y Nodos IoT



Nota. Módulo de desarrollo IoT esquema de componentes. Tomado de (Smart, 2024).

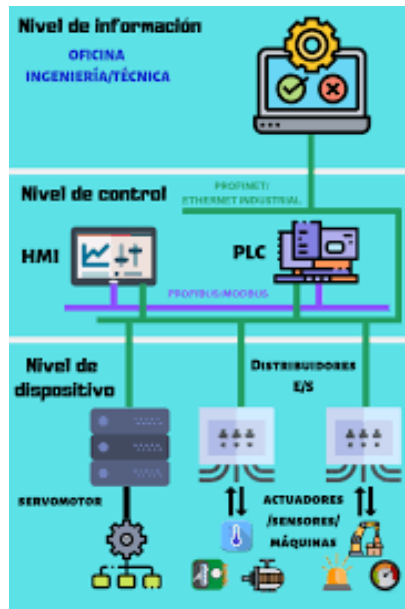
4.9 Protocolos de comunicación Industrial y redes de datos

El estándar de Ethernet Industrial se ha consolidado como la base normativa para así poder transmitir los datos en planta, adoptando así, la tecnología 802.3 para entornos que requieren alta inmunidad al ruido. A diferencia de las redes ofimáticas, las redes industriales suelen implementarse bajo arquitecturas estructuradas como la topología en estrella, la cual se puede apreciar en la figura 10, esta configuración se centraliza las conexiones en un switch – conmutador, lo que facilita la transmisión de paquetes y gestionar el tráfico que ese produce, evitando colisiones de datos y asegurando que la desconexión física en un nodo no interrumpa la comunicación del resto del sistema de control (Exertherm, 2024).

Sobre esta capa física opera el protocolo PROFINET IO, un estándar abierto definido en la norma IEC 61158 diseñado para el intercambio cíclico de datos entre un controlador – maestro. Este protocolo se distingue por su arquitectura de canales divididos: Un canal de Tiempo Real (RT) para señales críticas de control que requieren latencias mínimas e invariables, y un canal estándar TCP/IP para la transmisión de datos acíclicos, como parámetros de configuración o diagnóstico (Exertherm, 2024).

Figura 10

Protocolos de comunicación Industriales



Nota. Arquitectura de comunicación PROFINET IO con canales de Tiempo Real (RT) y TCP/IP. Tomado de (Aula21, 2025).

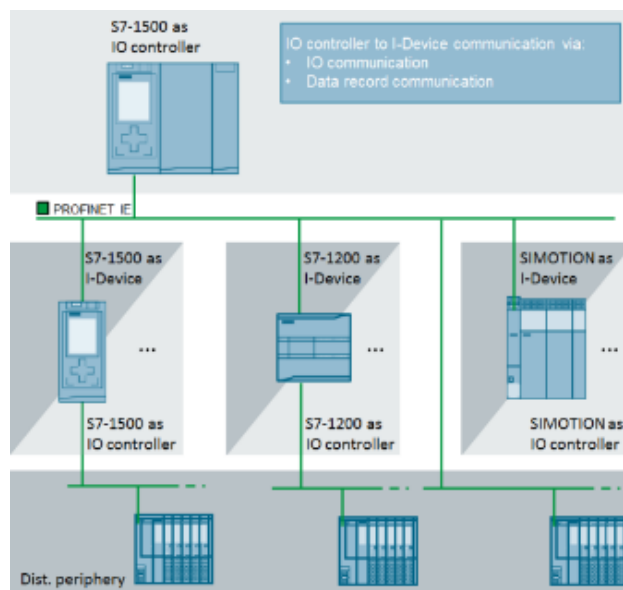
4.9.1 Protocolo Profinet IO

El protocolo PROFINET IO - Process Field Network Input/Output es un estándar de comunicación industrial abierto, basado en Ethernet y normado bajo el estándar internacional IEC 61158, está diseñado para el intercambio cíclico de datos entre sistemas de control y dispositivos de campo, su arquitectura operativa se fundamenta en el modelo proveedor-consumidor donde los dispositivos publican sus estados de entrada y adquieren las órdenes de salida enviadas desde un controlador central (Siemens, 2023).

Como se ilustra en la figura 11, la principal fortaleza técnica de PROFINET IO radica en su capacidad para manejar canales de comunicación paralelos sobre el mismo cable físico. Para las tareas críticas, utiliza el canal de Tiempo Real (RT), el cual omite las capas superiores del modelo TCP/IP para reducir la latencia, logrando tiempos de ciclo deterministas en el rango de 1 a 10 ms, ideales para la regulación de velocidad del sistema de bombeo (Siemens, 2023).

Figura 11

Protocolo Profinet IO



Nota. Arquitectura Maestro-Esclavo en red PROFINET IO. Tomado de (Siemens, 2023).

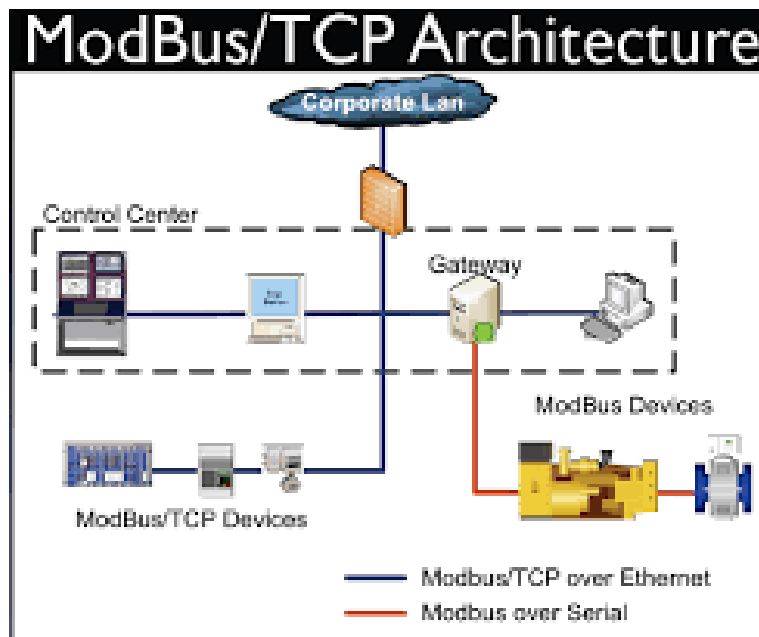
4.9.2 Protocolo Modbus TCP/IP

Modbus TCP/IP es la evolución del protocolo Modbus, es decir, este opera sobre redes Ethernet modernas, el cual combina la simplicidad del modelo original, con la versatilidad del protocolo TCP/IP. Este funciona bajo una arquitectura Cliente-Servidor, donde el cliente es el quien solicita los datos, indica una conexión en el puerto 502 hacia el servidor, donde este permitirá una comunicación abierta y estandarizada entre equipos diferentes. Su amplia adopción en la industria se debe a que utiliza la infraestructura de red convencional (routers y switches), facilitando la integración de sistemas de telemetría y IoT industrial (Aula21, 2025).

Como se observa en la figura 12, permite gestionar múltiples solicitudes simultáneas dentro de la misma red sin que se mezclen las respuestas. Esta estructura de trama eficiente es lo que hace posible que microcontroladores y PLCs intercambien grandes volúmenes de datos de manera fiable y rápida (Aula21, 2025).

Figura 12

Modbus TCP/IP



Nota. Estructura de trama Modbus TCP/IP con encabezado MBAP. Tomado de (Uhu, 2024).

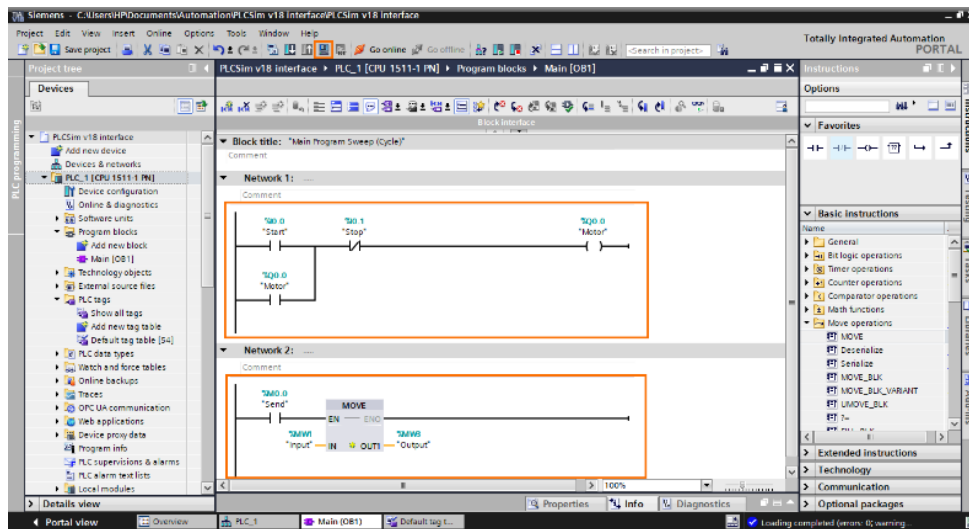
4.10 Software TIA Portal V18

TIA Portal V18 - Totally Integrated Automation es una plataforma de ingeniería unificada por Siemens, está diseñada para poder programar, integrar, y visualizar bajo un mismo entorno procesos industriales. Esta versión permite la optimización del flujo de trabajo, ya que se compone de herramientas como Simatic STEP 7 y WINCC, permitiendo que se pueda lograr la programación lógica de la configuración del hardware del PLC, además del diseño de pantallas de HMI (Siemens, 2024).

Además de la gestión de hardware, también se destaca por su amplia integración de librerías, bibliotecas y su capacidad para el diagnóstico en procesos o sistemas de campo en tiempo real. Como se presenta en la figura 13, el entorno ofrece una visualización de la programación en lenguaje Ladder, gráfica de redes, y topologías, que facilitan la implementación de dichos bloques de función complejos de manera intuitiva. Dentro de este software se gestionan las librerías necesarias para que permita al usuario instanciar instrucciones para la comunicación industrial, ya sea de servidor Modbus o control de movimiento, directamente desde el catálogo de hardware (Siemens, 2024).

Figura 13

TIA PORTAL V18



Nota. Interfaz de configuración de dispositivos y redes en TIA Portal V18. Tomado de (PLC, 2025).

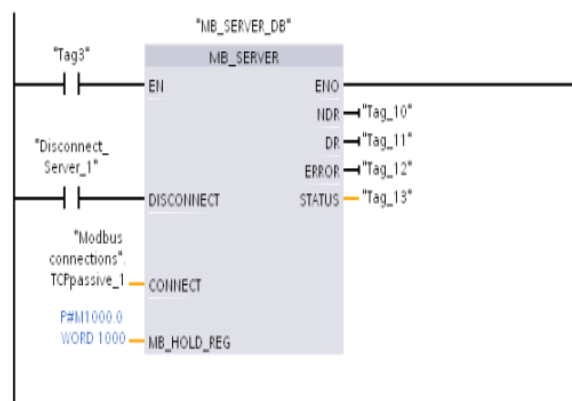
4.11 Funcionalidad del servidor Modbus en Siemens (MB_SERVER)

La instrucción MB_SERVER es un bloque de función estándar integrado en el entorno TIA Portal que permite a los controladores SIMATIC S7-1500 operar como un servidor dentro de una red Modbus TCP. Su función principal es habilitar el PLC para así poder escuchar o entender de manera pasiva, las solicitudes de conexión entrantes, a través de su puerto PROFINET – el puerto 502, y además responder las peticiones de lectura o escritura enviadas por un cliente externo. Esta capacidad es fundamental para la interoperabilidad, ya que permite exponer ciertas áreas de memoria del autómatas como "Holding Registers" estandarizados, facilitando que dispositivos de terceros, como microcontroladores o sistemas SCADA, accedan a los datos del proceso sin requerir hardware de comunicación adicional (Siemens, 2024).

Para su implementación, el bloque requiere de una asignación de punteros, que lo lleve a una estructura de datos que sean específicos los cuales definan parámetros de conexión y almacenamiento. Como se logra observar en la figura 14, la gestión de la información se realiza a través de parámetro MB_HOLD_REG, el cual se está vinculando a un bloque de datos, DB del PLC con el mapa de registros Modbus que es accesible por el cliente. Esta arquitectura de software garantiza que el intercambio de información telemétrica como los estados del variador o lecturas de sensores se realice de manera ordenada y cíclica, manteniendo la integridad de los datos del sistema de control (Siemens, 2024).

Figura 14

Bloque MB SERVER



Nota. Diagrama de bloques de la instrucción MB_SERVER en TIA Portal. Tomado de (Siemens, 2024).

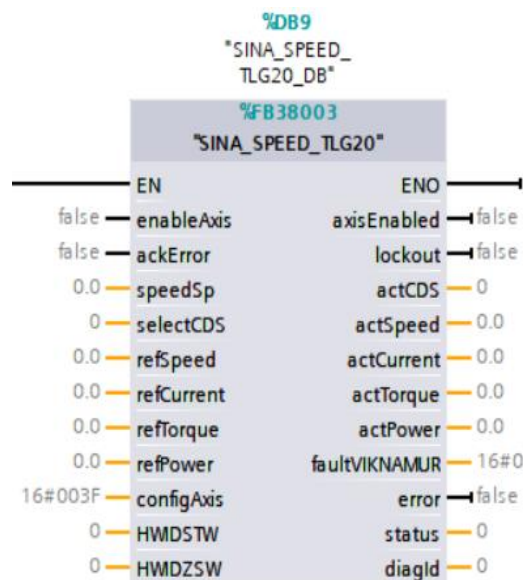
4.13 Bloque de función SINA SPEED

La instrucción SINA_SPEED es un bloque de función estandarizado perteneciente a la librería de TIA Portal, diseñado para simplificar el control cíclico de variadores de la familia SINAMICS. Su función principal es encapsular la complejidad del protocolo PROFIdrive gestionando las secuencias lógicas de control y estado para la operación del motor, esto permite poder manejarlo mediante consignas sin necesidad de programar una lógica, o manualmente, sino directamente mediante la manipulación de cada bit dentro de la palabra de mando del telegrama (Siemens, 2024).

La implementación de este recurso se realiza a través de una interfaz de software que abstrae las señales físicas del variador en variables de fácil manejo. Como se ilustra en la figura 16, el bloque presenta una estructura de entradas y salidas intuitiva para funciones críticas como la habilitación del eje, la referencia de velocidad y el reconocimiento de fallos, facilitando su inserción en lenguajes de programación como Ladder (KOP) (Siemens, 2024).

Figura 16

Bloque SINA SPEED TLG20



Nota. Interfaz de entradas y salidas del bloque de función SINA_SPEED. Tomado de (Siemens, 2024).

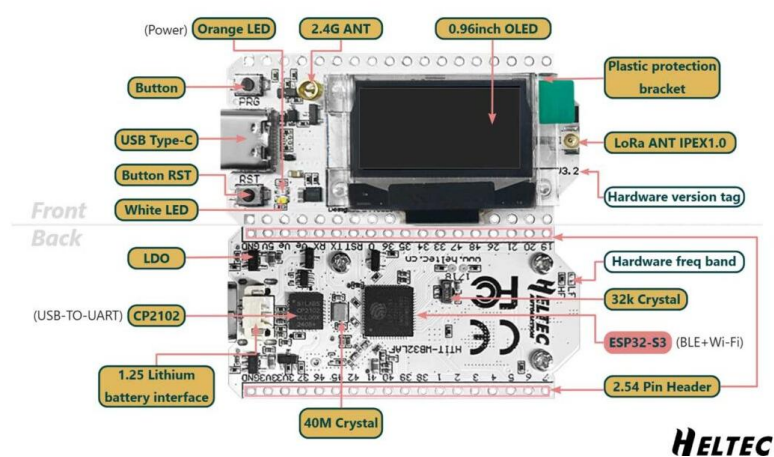
4.14 Heltec Wifi Lora 32 (V3)

El módulo Heltec WiFi LoRa 32 (V3) es una plataforma de desarrollo IoT de alto rendimiento basada en el SoC (System on Chip) ESP32-S3 de Espressif. Este microcontrolador contiene un núcleo doble, que opera a 240 MHz, integra capacidades de comunicación inalámbrica Wifi y bluetooth 5.0, con la tecnología de modulación LoRa a través del chip de radio SX1262. Esta arquitectura permite la integración de nodos de telemetría diseñados para capturar variables localmente y transmitirlos a largas distancias. El prototipo asegura que dichos paquetes se envíen exclusivamente al receptor designado, optimizando el ancho de banda y la seguridad de los datos (Megatronica, 2023).

En cuanto a su arquitectura de hardware, el dispositivo está diseñado para facilitar la implementación rápida de prototipos. Como se observa en la figura 17, la placa incorpora una pantalla OLED de 0.96 pulgadas para la visualización de diagnósticos locales y un sistema de gestión de baterías de litio, lo que le otorga autonomía operativa. Su chip LoRa SX1262 ofrece una transmisión eficiente en comparación con versiones anteriores, garantizando la integridad de los datos en enlaces punto a punto o redes LoRaWAN (Megatronica, 2023).

Figura 17

Diagrama e imagen del microcontrolador Heltec V3



Nota. Diagrama de componentes y distribución de pines del módulo Heltec V3. Tomado de (Automation, 2025).

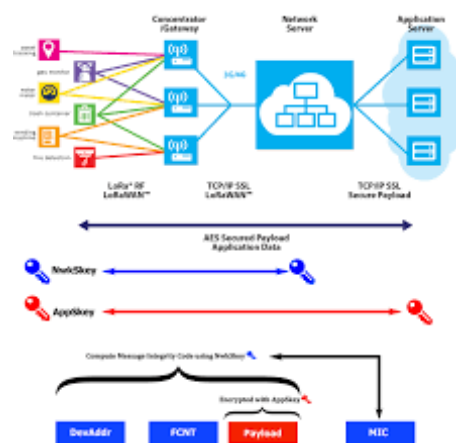
4.15 Fundamentos de la Modulación Lora

La modulación LoRa - Long Range es una tecnología de capa física patentada por Semtech que utiliza la técnica de Espectro Ensanchado por Chirp. A diferencia de las transmisiones de radios convencionales, que modulan dicha amplitud o una fase de cierta onda portadora fija, con la modulación Lora, este codifica la información, mediante pulsos de frecuencia, los cuales son variables y se les denomina ‘chirps’, esta técnica permite que la señal sea modulada incluso cuando exista una potencia inferior al nivel de ruido del ambiente, otorgándole una inmunidad excepcional frente a interferencias electromagnéticas y desfases por rebote de señal (Vencoel, 2025).

Un parámetro crítico en esta tecnología es el Factor de Expansión, el cual es el que define la duración de cada uno de los chips, y como consecuencia define la velocidad de comunicación de los datos. Como se puede observar en la figura 18, se puede ver un esquema de la modulación lora, esta representación de la señal al realizarlo en la práctica existen oscilaciones ascendentes y descendentes, que ocupan todo el ancho de banda asignado, esta robustez espectral permite que el receptor distinga la señal útil de otras emisiones concurrentes. Al aumentar el SF, se incrementa la sensibilidad del receptor y la distancia de cobertura, aunque se reduce la tasa de bits disponible, lo que obliga a diseñar el sistema para transmitir paquetes de datos pequeños y concisos (Vencoel, 2025).

Figura 18

Modulación de comunicación LORA



Nota. La comunicación Lora usa un tipo de modulación de amplio espectro, ideal para tolerar el ruido y para que una señal realice caminos múltiples. Tomado de (Logicbus, 2025).

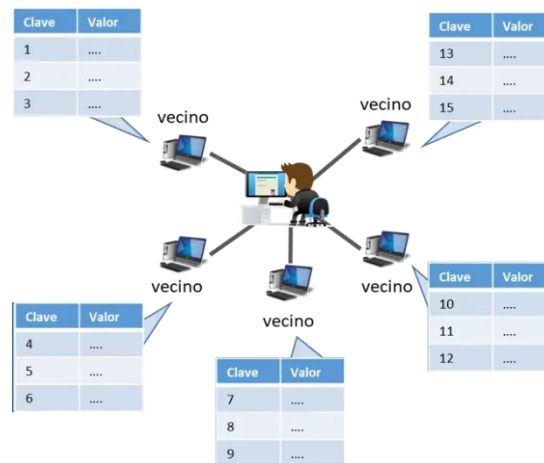
4.16 Arquitectura de comunicación punto a punto

La arquitectura de comunicación Punto a Punto (P2P) establece un enlace directo de transmisión entre dos nodos terminales sin requerir infraestructura intermedia compleja, como gateways o servidores de red. A diferencia de las topologías centralizadas como lo son LoRaWAN, este esquema permite que el dispositivo transmisor envíe la información exclusivamente al receptor designado, eliminando la dependencia de conectividad a internet (Wedbyn, 2024).

Para garantizar el enlace, es indispensable que ambos dispositivos estén sincronizados bajo los mismos parámetros de radiofrecuencia. Como se ilustra en la figura 19, un esquema de cómo es la comunicación punto a punto, además de que el proceso operativo consiste en la encapsulación de la información del proceso en tramas de radio que viajan directamente del emisor al receptor, esto crea un puente de datos que permite extender dichas señales industriales a distancias muy grandes de manera eficiente (Wedbyn, 2024).

Figura 19

Arquitectura de comunicación punto a punto



Nota. Esquema de conexión directa entre nodos en arquitectura P2P. Tomado de (Blancarte, 2025).

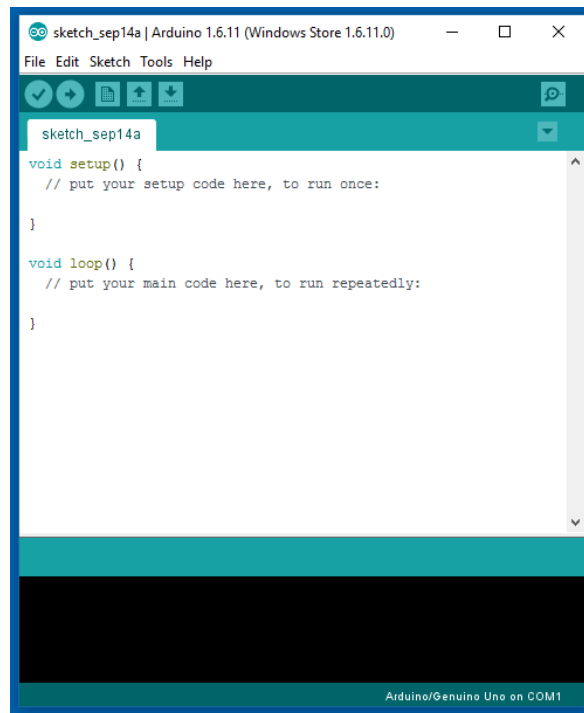
4.17 Software Arduino IDE

El Arduino IDE es una aplicación multiplataforma de código abierto diseñada para facilitar la escritura, compilación y carga de código fuente en microcontroladores y tarjetas de desarrollo compatibles. Este entorno se basa en el lenguaje de programación C++, simplificado mediante una estructura de librerías y funciones predefinidas que abstraen la complejidad de los registros de hardware. Tiene herramientas que permiten la transformación del código de alto nivel en archivos binarios ejecutables por el microprocesador, actuando como el puente fundamental entre la lógica de programación y el hardware físico (Arduino, 2025).

Como se aprecia en la figura 20, la interfaz del entorno permite tanto la estructuración del código como la visualización de varias variables en el monitor serial, facilitando las pruebas de conectividad y validación de datos (Arduino, 2025).

Figura 20

Software Arduino



Nota. Esquema de conexión directa entre nodos en arquitectura P2P. Tomado de (Arduino, 2025).

5. MARCO METODOLÓGICO

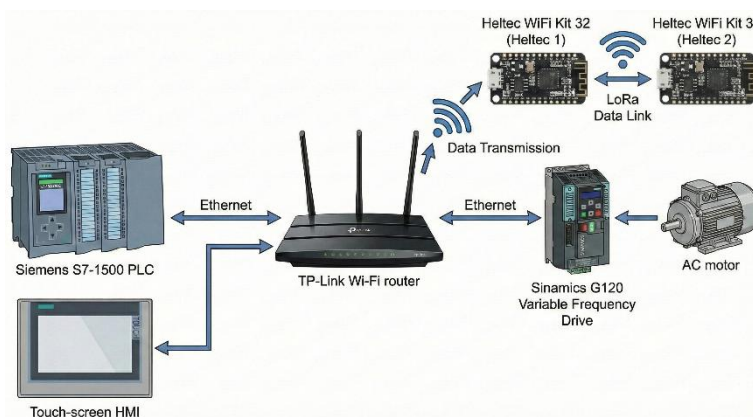
5.1 Descripción del proyecto

La propuesta técnica consiste en el desarrollo integral de un prototipo de monitoreo energético para estaciones de bombeo, diseñado específicamente para las condiciones del sector camaronero. El núcleo de la solución es la integración de los elementos tanto como el controlador industrial PLC S7-1500, una fuente confiable de adquisición de datos eléctricos y se complementa con una arquitectura de telemetría basada en módulos de desarrollo IoT con capacidad de transmisión inalámbrica.

Como se observa en la figura 21, el funcionamiento del prototipo sigue un flujo de datos lineal: el PLC procesa las variables de campo y las entrega a una etapa de transmisión; aquí, mediante tecnología de modulación de radiofrecuencia, la información viaja de forma inalámbrica punto a punto hasta un nodo receptor. Finalmente, el prototipo cierra el ciclo de supervisión exponiendo los valores a través de una interfaz de visualización en el receptor, permitiendo al operador verificar el estado del bombeo en tiempo cercano al real. Esta iniciativa busca validar una alternativa tecnológica que elimine la dependencia de cableado costoso en zonas extensas, ofreciendo una herramienta de diagnóstico accesible y replicable.

Figura 21

Diagrama del proceso de comunicación modbus tcp/ip.



Nota. Diagrama del proceso, el cual se utilizó en la parte de implementación del prototipo.

5.2 Tipo de metodología

Para la ejecución de este trabajo de titulación se ha adoptado una metodología de carácter experimental y aplicada. Es experimental porque el núcleo de la investigación reside en la manipulación deliberada de variables de comunicación como la velocidad de transmisión serial o los parámetros de potencia del enlace con el fin de evaluar su comportamiento, medir el alcance efectivo y determinar la estabilidad de la conexión en un entorno controlado.

Simultáneamente, el enfoque es aplicado y sistemático, ya que el desarrollo no se queda en la teoría, sino que sigue una secuencia ingenieril lógica: diseño de la interfaz, programación del firmware, integración física y validación de campo. Este procedimiento ordenado permite certificar que la integración entre el protocolo industrial Modbus/Profinet y la transmisión inalámbrica es técnicamente viable para solucionar la problemática de supervisión remota en las camaroneras.

5.3 Etapas del prototipo

5.3.1 Etapa 1: Programación del PLC y Adquisición de datos

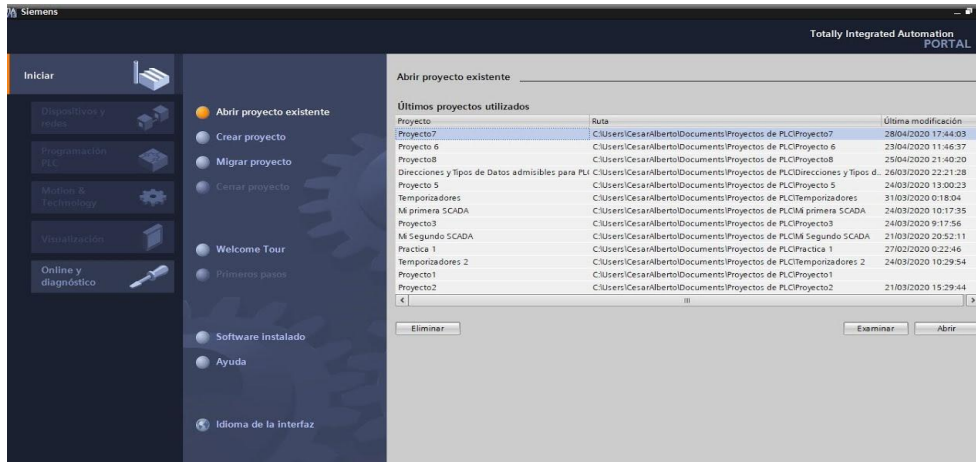
En la fase inicial se diseñó el control dentro del PLC, el cual se utiliza el software TIA Portal, se configurará el PLC S7-1500 para así poder gestionar las variables energéticas fundamentales, además de programar la lógica operativa de los motores en estaciones de bombeo, que abastecen el agua de las piscinas camaroneras. Se estructuraron los bloques de función para gestionar el accionamiento de las bombas y centralizar la información de las variables eléctricas fundamentales, tanto como, el voltaje, el amperaje, y la potencia, dentro de un sistema de bombeo, accediendo a que los datos puedan estar listos para ser detectados.

Creación del Proyecto y Enlace Online

Al abrir el software de TIA Portal, se creó un nuevo proyecto, seleccionando la CPU correspondiente al controlador S7-1500. También, se puede utilizar la función de Online Access, del software, la cual realiza el escaneo de la red mediante la tarjeta de interfaz, logrando así detectar y poder establecer la conexión exitosa tanto como el PLC, como con el panel HMI, como se observa en la figura 22.

Figura 22

Creación de nuevo proyecto en TIA Portal



Nota. Selección de documento o creación de nuevo archivo de proyecto y empezar con las configuraciones.

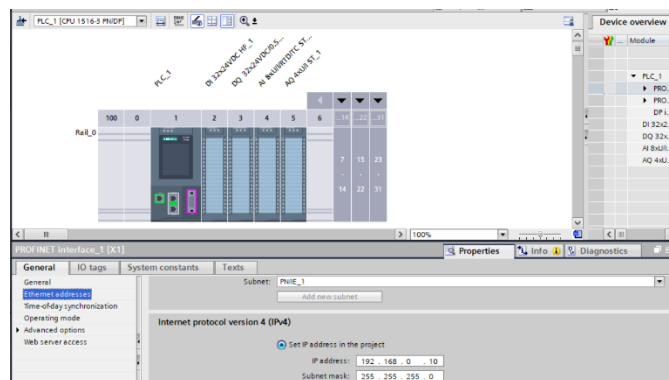
Implementación de la Topología de Red Física

Se diseñó la red física de cada uno de los equipos, tanto como la del PLC, HMI, y laptop, a un router switch TP-Link, a través de los cables Ethernet, estableciendo una topología estrella para poder comenzar con la comunicación.

En la figura 23, se configuró la dirección IP de puerta de enlace del router, que en este caso fue de 192.168.0.20, para el PLC 192.168.0.10, para el HMI 192.168.0.2, y para el variador 192.168.0.3, Se configuran todos en el mismo segmento para así asegurar la visibilidad y evitar conflictos de dirección.

Figura 23

Dirección Ip del PLC s7-1500



Nota. Asignación de direcciones IP estáticas en la configuración de red.

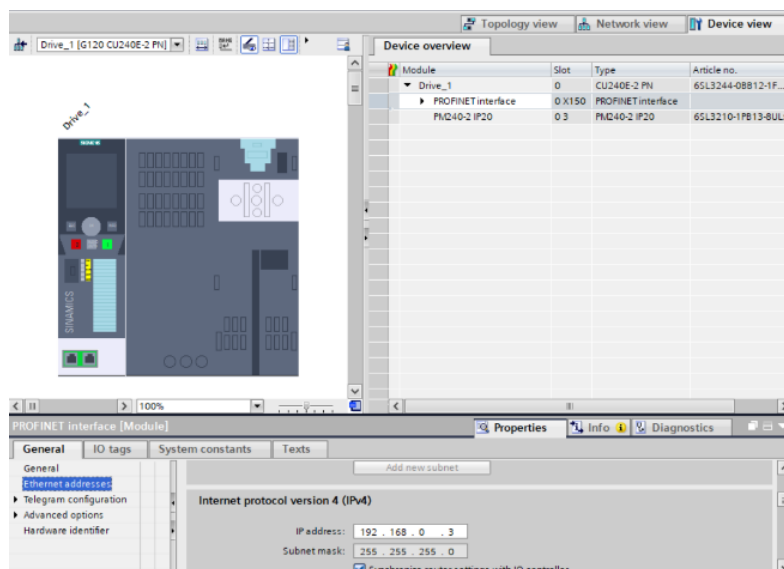
Integración y Parametrización del Variador de Frecuencia

Se integró el variador de frecuencia SINAMICS G120 a la red física, se consolidó mediante la interconexión de los equipos a un switch TP-Link a través de cables Ethernet, estableciendo una topología estrella.

Para el reconocimiento dentro del software TIA Portal, se realizó la investigación e instalación de los archivos de soporte de hardware GSD/HSP. Una vez que reconozca el variador, se configuró el intercambio de datos seleccionando el telegrama estándar 20, el cual es adecuado para la aplicación de bombeo, como se puede observar en la figura 24.

Figura 24

Selección del telegrama 20 en la configuración del dispositivo



Nota. Se estableció el respectivo drive para así poder detectar el variador de frecuencia G120 y empezar a realizar la programación.

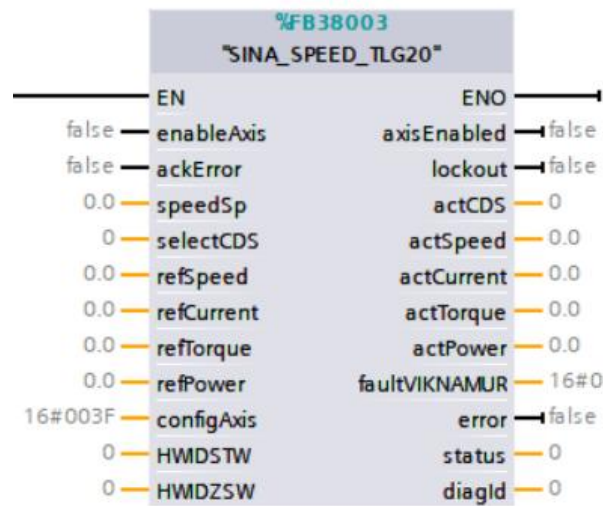
Lógica de Control de Accionamiento (DriveLib)

El accionamiento del control de motor se realizó utilizando una librería extendida, donde, se colocó el bloque de función SINA_SPEED en un segmento network del bloque principal en este caso el main.

Se parametrizó las entradas con los valores nominales del motor ya sea Velocidad, corriente, torque, y potencia, además se asignó las direcciones de hardware correspondientes al telegrama 20, como se puede observar en la figura 25.

Figura 25

Bloque de programación SINA SPEED TLG20



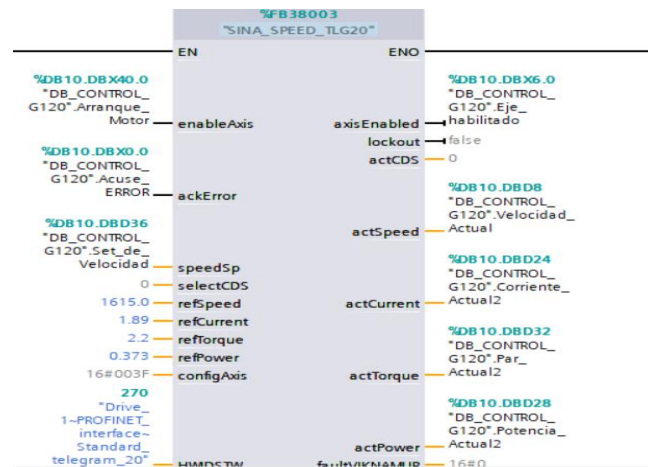
Nota. Este bloque de programación permite habilitar el arranque del variador, otorgándole varios valores.

Una vez implementado este bloque de SINA SPEED, y parametrizado con los valores de referencia, se definió las variables para el intercambio de datos dentro del bloque denominado DB CONTROL G-120, el cual centraliza los parámetros obtenidos del variador para su posterior lectura.

Se realizó la compilación del hardware y software seguida de una simulación de arranque para verificar la comunicación, y se comprobó el accionamiento efectivo del motor y la recepción de los datos en tiempo real, una vez que se compiló, se colocó las variables, como se puede observar en la figura 26.

Figura 26

Bloque de programación SINA SPEED TLG20

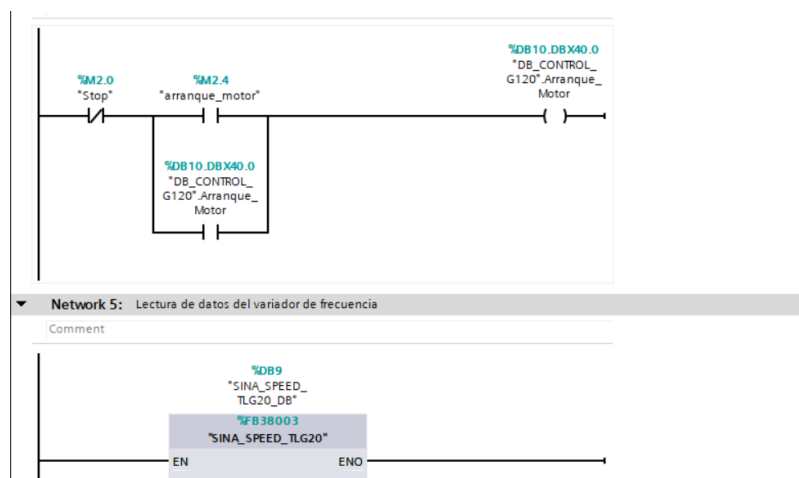


Nota. Variables y definición de parámetros dentro del bloque de programación SINA SPEED TLG20.

Para evitar el forzado manual dentro del bloque para el arranque, lo que se hizo fue una lógica ladder de programación, donde en otro network, se colocó un contacto cerrado para el STOP, un contacto abierto para el Arranque, y la bobina donde se referencia la variable Arranque Motor, la cual está dentro del DB CONTROL G120, además de, un contacto abierto en paralelo con el arranque, de la misma variable del DB CONTROL G120, para así hacer un enclavamiento, como se puede ver en la figura 27.

Figura 27

Lógica sencilla en Ladder.



Nota. Encendido y apagado del motor, llamando a la variable que está colocada en el SINA SPEED.

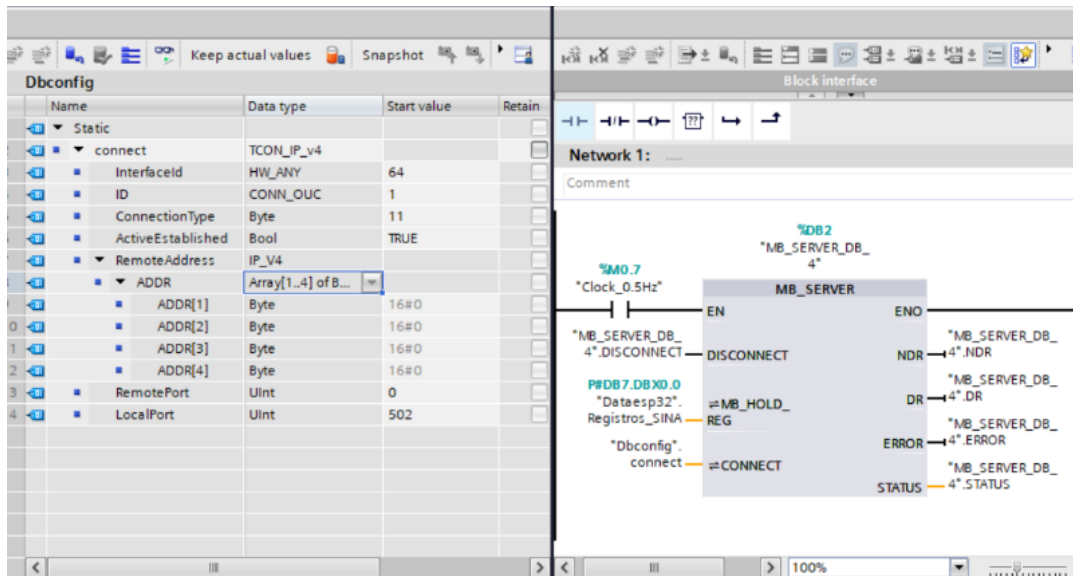
Configuración del Servidor Modbus TCP (MB_SERVER)

Una vez que se estableció este arranque, con el bloque SINA SPEED, se logró establecer la comunicación externa que se implementó, para este caso, esta configuración ayudará al bloque llamado MB_SERVER, que se encuentra en el apartado de comunicación del TIA Portal.

En la figura 28, se puede observar la gestión, Se generó un bloque de datos llamado DB_config para la parametrización de la instrucción MB_SERVER, la cual se estableció, el puerto local 502 para la gestión de solicitudes Modbus TCP/IP.

Figura 28

Imagen del DB_CONFIG para programar bloque MB_SERVER

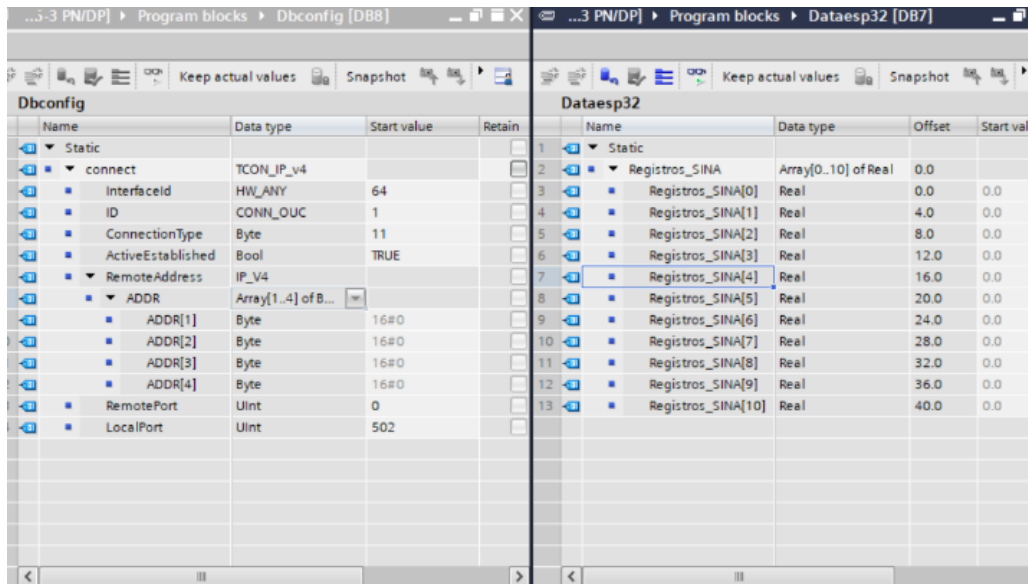


Nota. DB el cual se adjunta al MB_HOLD_REG, debido a que estos serán los datos a trasladar en el heltec.

Una vez que se concluye ese paso, se generó un tercer bloque de datos, denominado Data_ESP32. En ese DB se define un arreglo, el cual va a funcionar como un mapa de registros, 'Holding Registers' donde aquí se van a almacenar los valores de velocidad, voltaje, corriente y potencia, los cuales se encuentran en el bloque DB CONTROL G120, así quedarían listos para ser leídos por el microcontrolador, como se puede observar en la figura 29.

Figura 29

Bloques de datos.

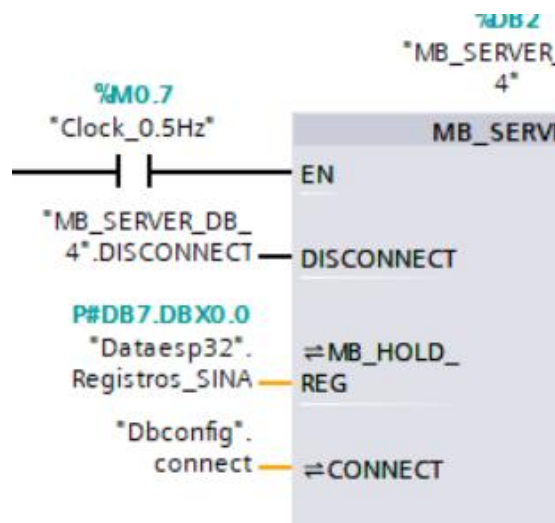


Nota. Configuración y variables para comunicar el programa.

Una vez que se realizó la creación de los DB, se vinculó el puntero del parámetro MB_HOLD_REG, del bloque MB_SERVER, con el DB_dataesp32, que se creó recientemente, habilitando así la zona de memoria que será la que permite la comunicación de estos datos por el cliente Modbus, como se puede observar en la figura 30.

Figura 30

Configuración del puntero para el DB Dataesp32



Nota. Vinculación del puntero dentro del bloque MB_SERVER a su MB_HOLD_REG.

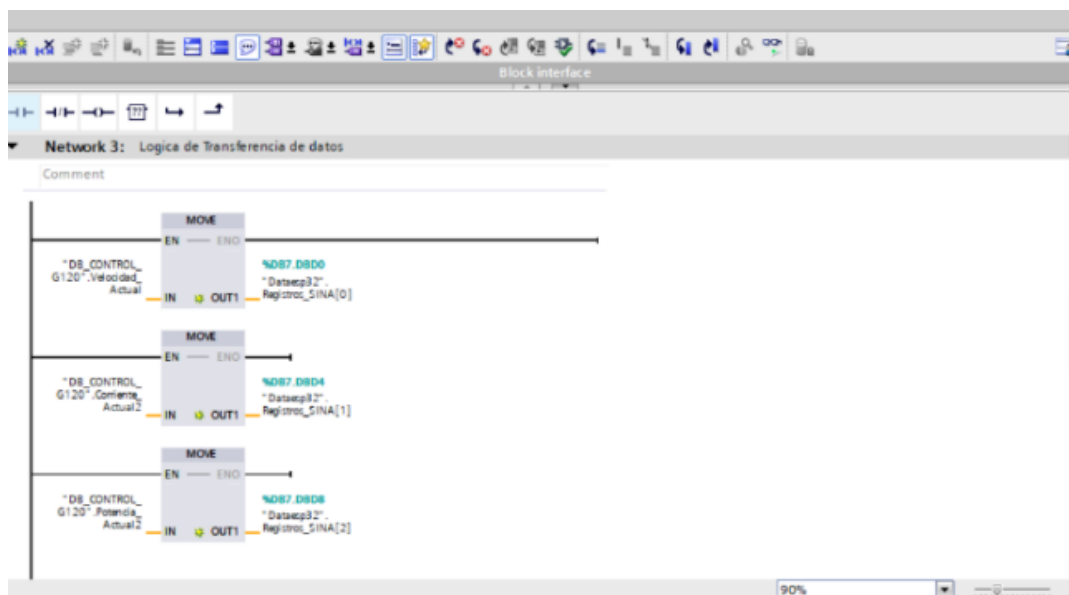
Vinculación de Datos (Bloque MOVE)

Dado que las variables del proceso se obtienen dentro del bloque SINA SPEED, se programó un segmento, es decir un nuevo network donde se colocó el bloque o instrucción MOVE. Como se puede observar en la figura 31, esto permite transferir los valores a leer, ya sea, corriente, potencia, velocidad y voltaje, desde el bloque de control, hacia el arreglo del DB_dataesp32, consolidando así la información en una sola lista de registros.

Al realizar la simulación se pudo detectar que cada uno de estos valores, trasladados al segmento del arreglo, de los DB creados, el registro SINA 0, será para la velocidad, el 1 para la corriente, el 2 para la potencia y el 3, para el voltaje.

Figura 31

Lógica de transferencia de datos con bloque MOVE

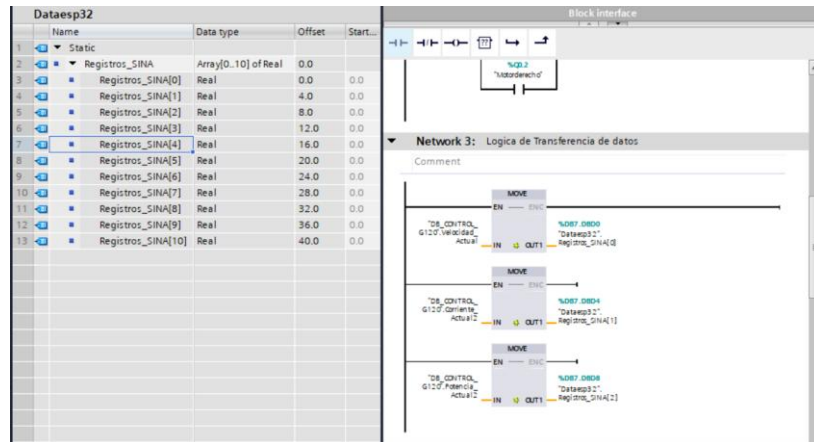


Nota. Bloque MOVE que permite el traslado de las variables de un bloque de datos a otro, en este caso hacia el arreglo.

Finalmente, se realizó pruebas de simulación y monitoreo online para verificar que los datos se actualizan correctamente dentro del registro Modbus, dejando el sistema listo para la integración con los módulos Heltec. Como se puede observar en la figura 32, el acompañamiento y la asignación de las variables en el bloque MOVE, de los Registros SINA [0], [1], [2].

Figura 32

Verificación de variables en el DB Data_ESP32



Nota. Datos del arreglo colocados en la salida del bloque MOVE, ya que en la entrada debe ir los datos del DB CONTROL G120.

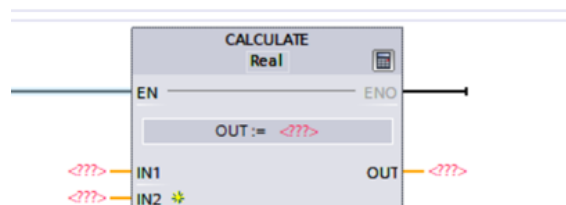
Cálculo para poder adquirir el voltaje del motor.

Ya que las variables están colocadas y listas para comunicar a través del MB_SERVER, ahora hay que calcular el voltaje, debido a que el bloque SINA SPEED, no lo contiene, solo contiene, la velocidad, corriente y potencia, entonces se realizó un proceso para poder adquirir este dato, colocarlo dentro de la programación y así se refleje en el heltec.

Para poder realizar este cálculo, en la figura 33 se logra ver el bloque llamado CALCULATE, el cual se tuvo que requerir de los parámetros del motor para poder establecer este proceso y asignarle las variables.

Figura 33

Bloque de programación Calculate

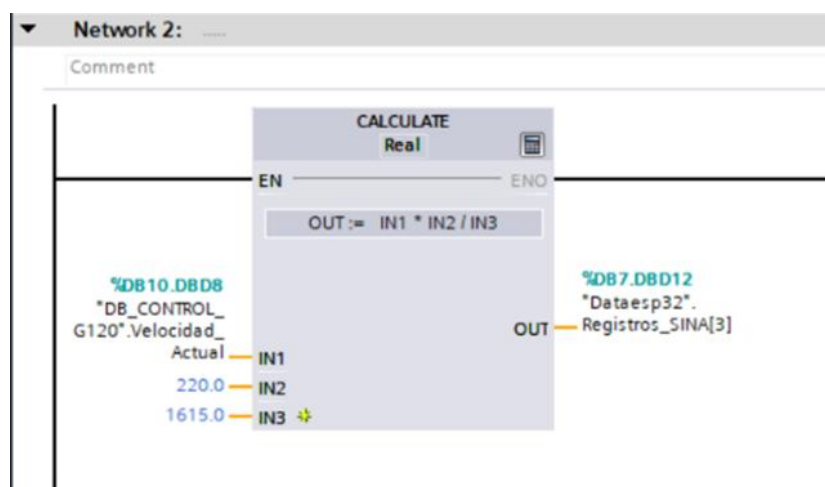


Nota. El bloque de programación calculate adjunta ciertas variables y parámetros para poder adquirir un valor total.

Una vez que se implementó este bloque, se ejecutó el principio de control escalar V/F - Voltaje/Frecuencia, esto también es una interpolación lineal basada en los datos de la placa del motor, estableciendo el voltaje nominal de 220v, y el valor rpm de referencia el cual es 1615, y la velocidad de los datos de DB_CONTROL_G120, que sería 'velocidad Actual' y en el Output se colocó el DB_datosesp32, el cual el valor resultante lo va a tomar un registro SINA y ese valor lo llevará a comunicarse, en la figura 34 se puede observar esta configuración.

Figura 34

Variables y referencias del bloque CALCULATE



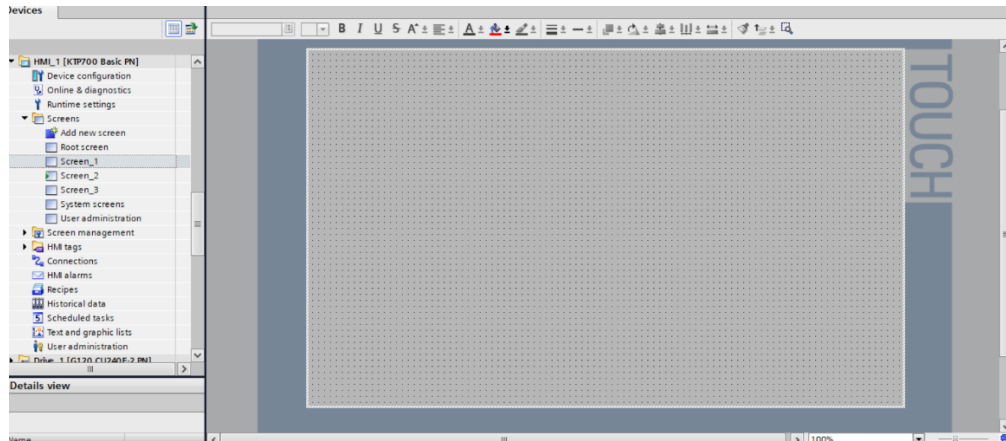
Nota. Validación y simulación de valores ya colocados en el bloque CALCULATE

5.3.2 Etapa 2: Interfaz HMI

Al finalizar con la programación del PLC, para poder controlar y visualizar de mejor manera como se está accionando, que datos se adquieren y trasladan, como se puede ver en la figura 35, se desarrolló una interfaz HMI.

Figura 35

Interfaz HMI



Nota. Se creó una nueva pantalla HMI, la cual se desarrolló para que el usuario pueda visualizar e ingresar datos desde el set point.

Una vez creada la pantalla (Screen1) como se puede observar en la figura 36, se colocó cada uno de los botones, que se van a utilizar.

Figura 36

Interfaz HMI con cada uno de los elementos

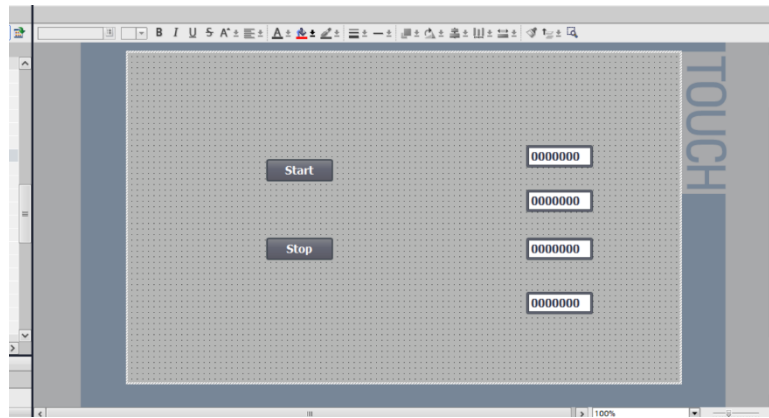


Nota. Pantalla HMI asignada, adjuntando primero los botones para después colocar los elementos con los se pudo interactuar.

Además, como se observa en la figura 37, también se colocó, los I/O Fields, lo que permitió modificar y visualizar valores, ya sea como entrada o salida.

Figura 37

I/O fields en la interfaz HMI

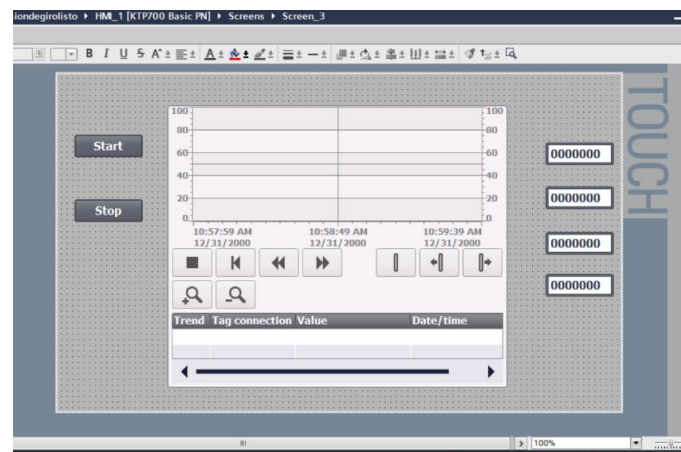


Nota. I/O Fields colocados dentro de la interfaz HMI.

Al colocar los botones y los I/O Fields, se estableció un elemento para visualizar la gráfica de las variables, y cómo se comporta dentro de variaciones, e instrucciones de datos, así se tendrá las oscilaciones del voltaje, corriente, potencia, velocidad y cuando se encuentre estable. Como se puede observar en la figura 38, además se le agregó una imagen de un motor, que va a servir como señalización de cuando esté en funcionamiento y cuando no lo esté.

Figura 38

Grafica implementada dentro de la interfaz HMI

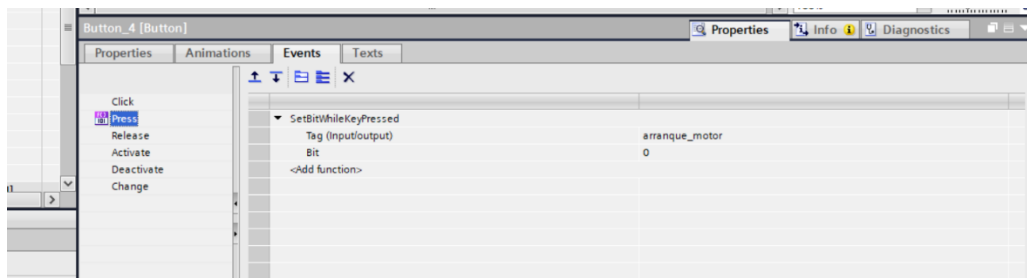


Nota. Gráfica que permite la visualización del comportamiento de cada uno de los datos.

Una vez colocado los botones, I/O fields, y la gráfica, Se planteó cada una de las variables, dentro de cada uno de los elementos. Se estableció la configuración de las variables de los botones, en este caso para el stop, la marca stop del network, así mismo del start, se seleccionó la marca llamada arranque de motor, como se observa en la figura 39.

Figura 39

Variable colocada en el botón Start.

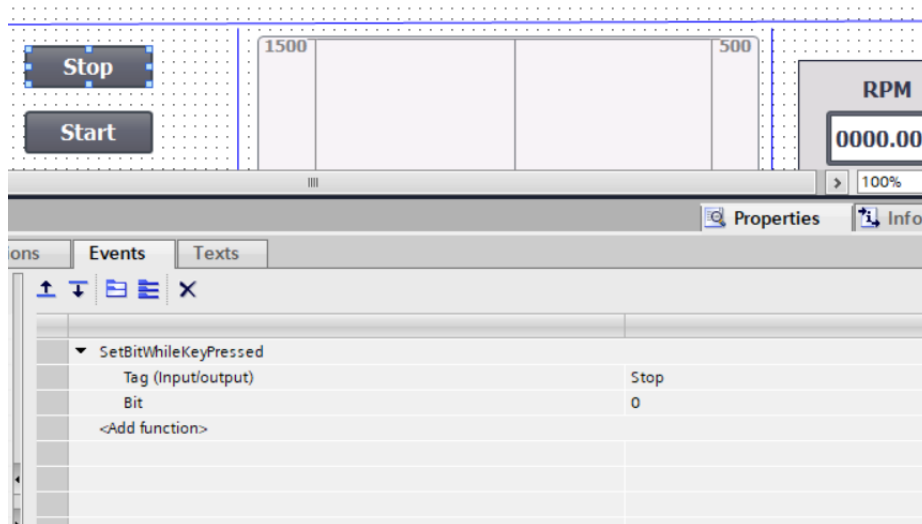


Nota. Variable es configurada para visualizar en el HMI en el programa.

Una vez colocado las variables en el botón start, se colocó la variable en el botón stop, como se puede observar en la figura 40.

Figura 40

Variable colocada en el botón Stop.

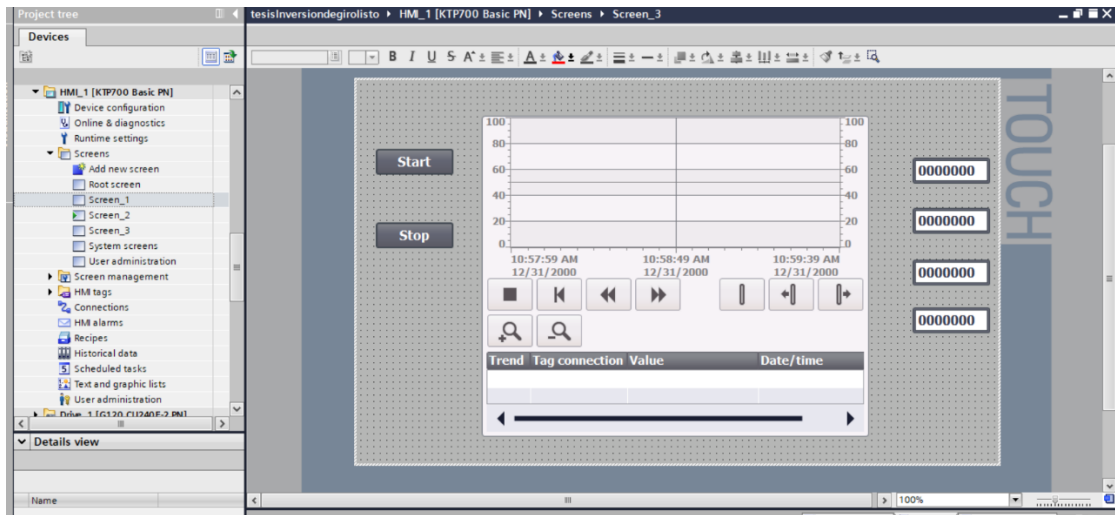


Nota. Variable es configurada para visualizar en el HMI en el programa.

En la figura 41, como se mencionó anteriormente, se colocó la gráfica, la cual va a permitir poder observar el comportamiento de las variables, que se vayan a implementar, así mismo el comportamiento influyente del motor.

Figura 41

Interfaz HMI con gráfica

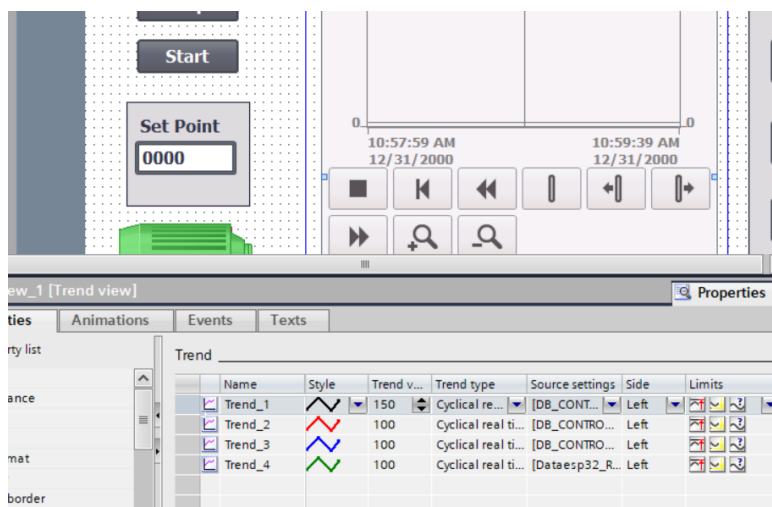


Nota. Trend view del software TIA portal para la observación de los datos.

Una vez colocado en la interfaz HMI, se configuró el trend view, con los siguientes métodos, el cual se va a reflejar la gráfica del comportamiento de cada una de las variables, como se puede observar en la figura 42.

Figura 42

Programación de las validaciones



Nota. Configuración de trend view, para la visualización de las gráficas de las variables.

Ya que se visualiza cada uno de los elementos que se configuraron, se realizó las pruebas de interacción con la programación, visualización y acción del motor, como se observa en la figura 43.

Figura 43

Interfaz HMI prueba 1

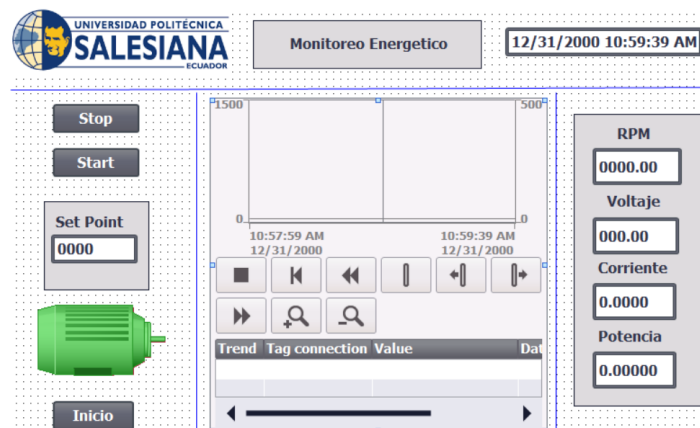


Nota. Se logra observar que el HMI, funciona de manera correcta, con respecto a lo que se solicitó.

Estéticamente como se puede observar se realizó unos cambios para la mejora de la visualización de los datos, se fue colocando ciertos elementos, para una interacción más realista, que se vea todo más organizado y llamativo, la finalización de la configuración HMI vista desde el software, se ilustra en la figura 44.

Figura 44

Imágenes, márgenes y mejoras HMI

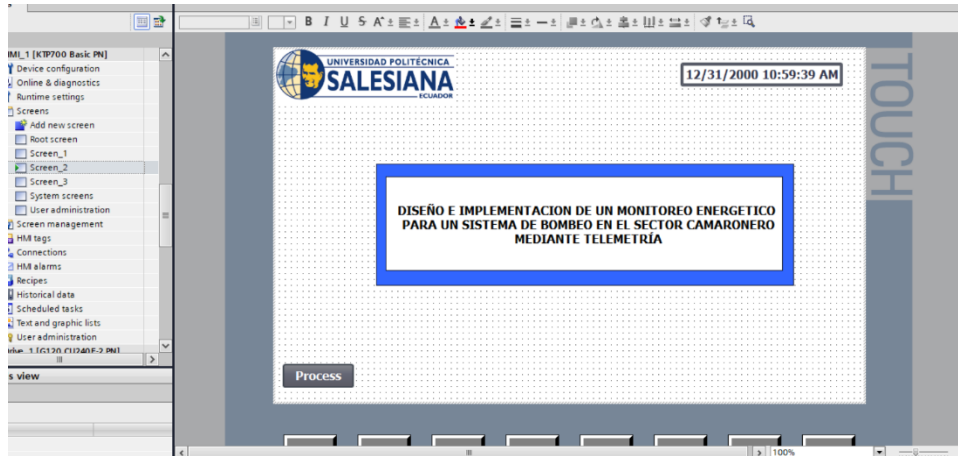


Nota. Configuración del Panel HMI.

Al terminar con la interfaz, se creó una nueva pantalla, donde se observará el título del proyecto, lo que se ilustra en la figura 45.

Figura 45

Pantallas e Interfaz HMI



Nota. Finalización de configuración de HMI.

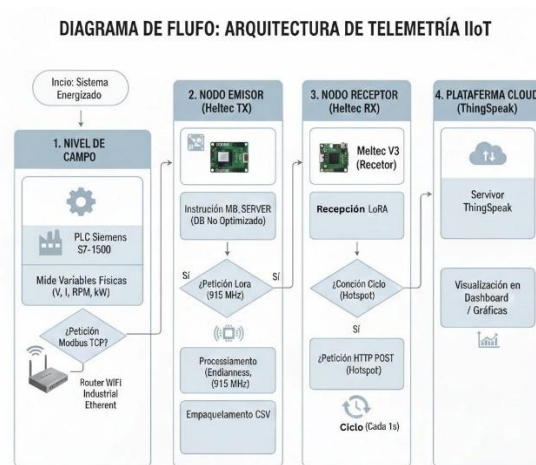
5.3.3 Etapa 3: Arquitectura de Interconexión y Protocolo Modbus TCP/IP

En esta fase el objetivo fue establecer un canal de comunicación entre el entorno de control industrial y el ecosistema de Internet de las Cosas (IoT). Se procedió a realizar dicha estructura de la implementación de una jerarquía de red donde el Heltec V3 actúa como Cliente Maestro y el S7-1500 como Servidor Esclavo, como se observa en la figura 46.

- **Configuración en TIA Portal:** Se utilizó el bloque de función estándar MB_SERVER. Para que los datos se envíen de una forma correcta, primero se crea un Bloque de Datos (DB) de acceso, lo que permitió asignar direcciones físicas fijas (Offsets) a las variables de Velocidad, Corriente, Potencia y Voltaje.
- **Parametrización del Enlace:** Se definió una dirección IP estática para el PLC (192.168.0.10) y se habilitó el puerto 502 para el tráfico Modbus. La conexión se realizó a través de un Router convencional que sirve como puente entre la red Ethernet cableada del PLC y la red Wi-fi del Heltec.
- **MB Server:** Como siguiente paso se realizó la configuración y despliegue del bloque en el entorno SIMATIC, definiendo un área de memoria específica mediante registros de retención (*Holding Registers*).

Figura 46

Diagrama de flujo proceso de comunicación



Nota. Diagrama de flujo del proceso de adquisición de datos y comunicación vía Lora.

Adquisición de Datos en el Variador Siemens G120

En la figura 47 se puede ver que, se continuó con la revisión del variador incluye lectura de los parámetros dinámicos del motor (V, I, P, RPM). Se integró el software Startdrive del TIA Portal, se configuró la transferencia cíclica de datos hacia el PLC mediante el telegrama #20 Profinet. Esto permitió que el PLC reciba en tiempo real los valores de las palabras de estado y los valores reales del motor.

Figura 47

Vista frontal del Variador G120-Siemens



Nota. Módulo de variador de frecuencia G120 del laboratorio de prácticas de Automatización Industrial. Adaptado de Sinamics G120X [Fotografía], por Siemens, 2024, <https://www.siemens.com/global/en.html>

Integración del Router TP-Link como Gateway Local

Este dispositivo actúa como el punto de acceso (Access Point). El PLC está conectado a uno de sus puertos LAN mediante un cable Ethernet, mientras que el Heltec TX se conecta vía Wi-fi. En la figura 48, como se puede visualizar, el Router gestiona el tráfico de datos y asegura que las peticiones Modbus del Heltec lleguen a la dirección IP estática del PLC caso de uso este procedimiento fue la IP: (192.168.0.10) del PLC s7-1500.

Figura 48

Router TP-link



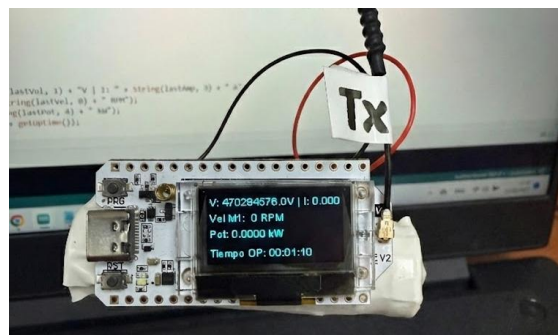
Nota. Se configuró el Router como una red de IP Estática dentro de su interfaz. Adaptado de Router Inalámbrico [Fotografía], por TP Link, 2024, <https://www.tp-link.com/>

Nodo Emisor Heltec V3 (TX) - Cliente Modbus

El código desarrollado utiliza la librería ModbusIP_ESP8266. El Heltec actúa como un Cliente Maestro, solicitando ráfagas de datos cada segundo. Tras recibir los registros, el procesador ESP32-S3 realiza el desempaquetado de los registros de 16 bits para reconstruir las variables de punto flotante de 32 bits antes de prepararlas para el envío por Lora, como se observa en la figura 49.

Figura 49

Punto del sistema inalámbrico



Nota. Microcontrolador Heltec transmisor que contiene el código de prueba 1. Adaptado de Wi-Fi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

Procesamiento de Datos y Modulación de Radiofrecuencia LoRa

El PLC entrega los datos en dos registros de 16 bits cada uno. Para obtener el número real (por ejemplo, 0.543 A), el Heltec TX debe unirlos. Se implementó una estructura de datos tipo unión en el firmware. Esta técnica permite interpretar dos valores

enteros como un solo valor decimal (Float 32 bits) sin perder ni un solo decimal de precisión, asegurando que lo visualizado en TIA Portal sea idéntico a lo reflejado, como se indica en la figura 50.

Figura 50

Lista del DB-ESP32

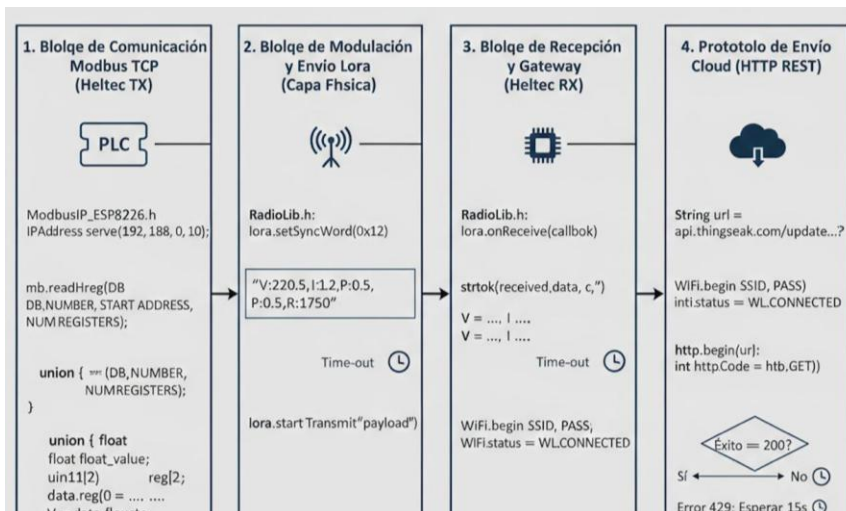
Dataesp32							
Name	Visible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment	
Static							
Registros_SINA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		VELOCIDAD DEL SINA SPEED	
Registros_SINA[1]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		CORRIENTE DEL SINA SPEED	
Registros_SINA[2]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		POTENCIA DEL SINA SPEED	
Registros_SINA[3]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		VOLTAJE DEL SINA SPEED	
Registros_SINA[4]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[5]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[6]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[8]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[9]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Registros_SINA[10]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Nota. Lista de parámetros llamados Holding Registers (SINA1-2-3-4).

Una vez procesados los datos, se configuró los parámetros de radiofrecuencia en el chip integrado. La banda entre Heltec que se usó fue de 915 MHz. Los 4 parámetros (V, I, P, RPM) se empaquetan en una trama de texto separada por comas. Esto optimiza el ancho de banda y reduce el tiempo de aire, lo que se traduce en una mayor fluidez en la actualización de los datos, como se puede observar en la figura 51.

Figura 51

Diagrama de flujo que del código y registros encapsulados



Nota. Los registros encapsulados en dos grupos, el Heltec V3 no procesa 32bits.

Gateway IIoT, Recepción LoRa y Conectividad Cloud

Esta fase se centra en la función del Heltec V3 (Receptor) como una pasarela de datos o Gateway, él debe gestionar la comunicación por radiofrecuencia de largo alcance y simultáneamente mantener un enlace activo con la red global.

Gestión de la Capa Física LoRa y Validación de Datos

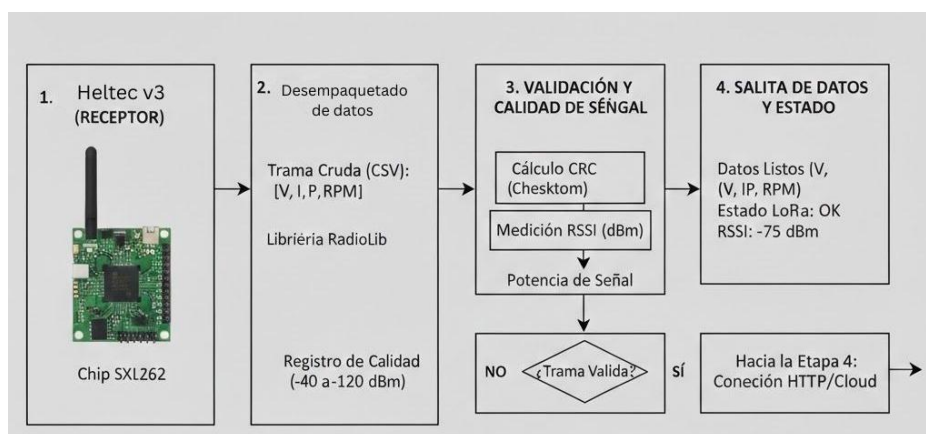
El receptor utiliza la librería RadioLib para controlar el transceptor SX1262, a diferencia de una conexión wifi simple, el protocolo LoRa requiere que el receptor esté en modo de escucha continua.

El firmware calcula el RSSI (Received Signal Strength Indicator), expresado en dBm, esto sirve para documentar la robustez del enlace. Un RSSI de -40 dBm indica una cercanía excelente, mientras que valores cercanos a -120 dBm marcan el límite de sensibilidad del receptor.

En la figura 52 se puede observar, el receptor realiza una comprobación de redundancia cíclica (CRC) automática. Si el paquete llega corrupto debido a interferencias de motores o maquinaria, el Heltec lo descarta automáticamente para evitar que suban valores erróneos a la nube.

Figura 52

Diagrama de flujo del Receptor



Nota. Representación del flujo como se estructuró para su respectivo funcionamiento.

Función de Gateway y Conectividad Hotspot

Debido a que en muchas plantas industriales el acceso a la red corporativa es restringido, se implementó una salida a internet mediante un Hotspot móvil.

El receptor se comprobó en la red de 2.4 GHz. Una vez obtenida la dirección IP interna, el dispositivo queda habilitado para realizar peticiones DNS y alcanzar los servidores de ThingSpeak.

Protocolo de Aplicación (HTTP REST): El Heltec RX actúa como un cliente HTTP. Toma los datos desglosados (V, I, P, RPM) y los concatena en una cadena de consulta. Esta cadena se envía mediante una petición GET a la URL de la API.

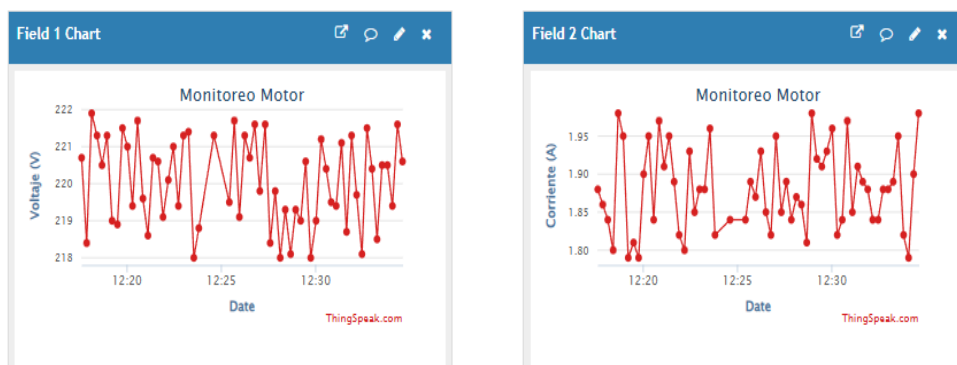
Sincronización y Latencia de Datos

Para evitar saturar el canal y respetar las políticas de uso de ThingSpeak (que permite una actualización cada 15 segundos en su versión gratuita), se programó una lógica de temporización:

El Heltec RX recibe datos LoRa cada segundo para mostrarlos con fluidez en su pantalla OLED integrada. Como se observa en la figura 53.

Figura 53

Dashboard de la página ThingSpeak



Nota. Vista del Hub Principal ThingSpeak con datos Reales.

6. RESULTADOS

En esta sección se registran los resultados de las pruebas realizadas en el módulo que se encuentra en el Laboratorio de Automatización II, se verifica el cumplimiento de cada etapa del proceso de programación, comunicación y el correcto funcionamiento de los elementos establecidos. También se verifica la precisión en el registro de los datos.

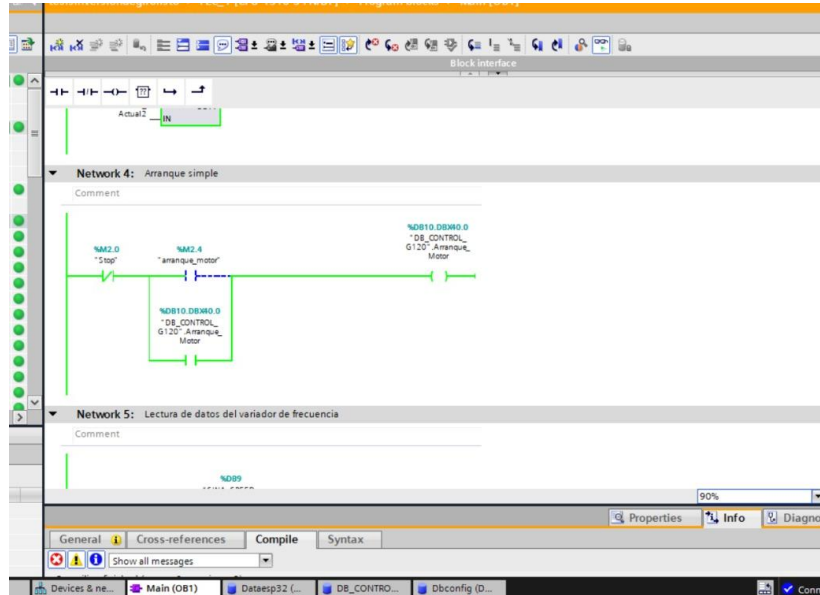
6.1 Programación del PLC

Para la programación del PLC, se validaron los resultados, empezando la simulación, de todo el programa, y cada network, el cual corresponde a la acción que está cumpliendo cada una de la programación.

Al iniciar la simulación de programación SINA SPEED, sin embargo, primero, se habilitó la lógica sencilla, de encendido y apagado del motor que se observa en la figura 54, que permite el arranque también del bloque SINA SPEED.

Figura 54

Arranque simple LADDER

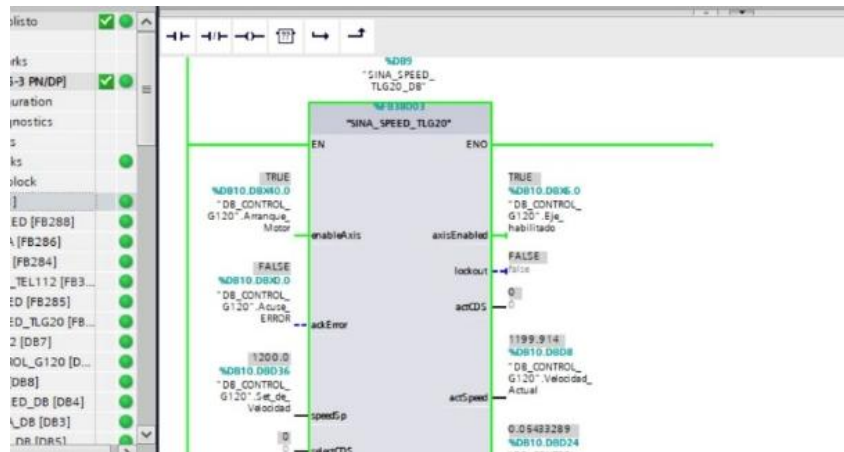


Nota. Lógica mediante el lenguaje de programación LADDER, que realiza el arranque del programa.

Al iniciar el arranque también se activó el bloque SINA SPEED, como se puede observar en la figura 55, se adquirió datos reales del variador a partir de los valores colocados como referencia.

Figura 55

Funcionamiento del Bloque SINA SPEED

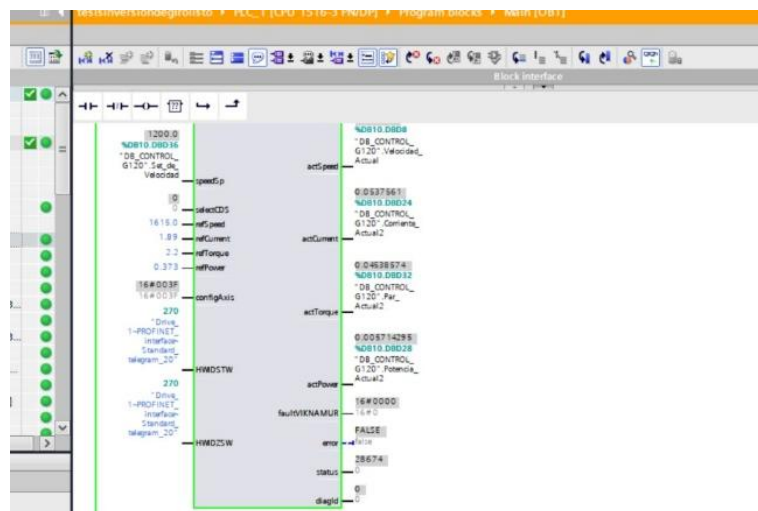


Nota. Bloque SINA SPEED en funcionamiento, para permitir el arranque del motor.

En la figura 56, se ilustra en su totalidad los valores reales de las variables a registrar en la comunicación.

Figura 56

Funcionamiento de Bloque SINA SPEED



Nota. Las variables reales son cambiantes, hasta que llegan a estabilizarse.

En la figura 57, se observó como el DB_CONTROL_G120 al simularlo también coincide con el bloque.

Figura 57

Bloque de datos en funcionamiento

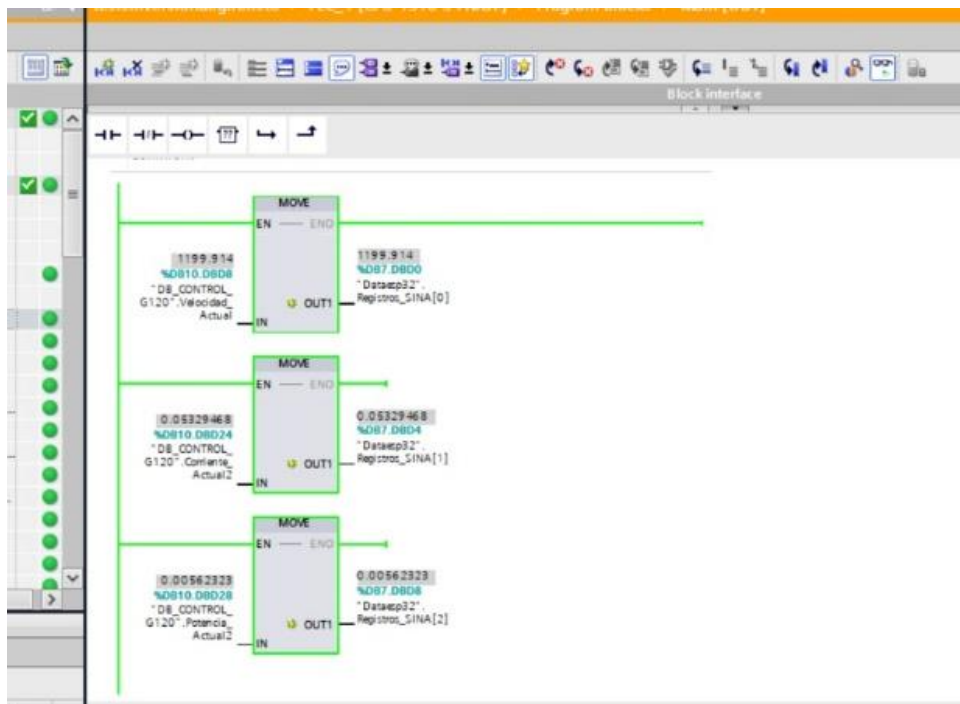
Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
Static									
Acuse_ERROR	Bool	0.0	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Velocidad_referencia	Real	2.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eje_habilitado	Bool	6.0	false	TRUE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Velocidad_Actual	Real	8.0	0.0	1199.914		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Corriente_Actual	Real	12.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Par_Actual	Real	16.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Potencia_Actual	Real	20.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Corriente_Actual2	Real	24.0	0.0	0.05283325		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Potencia_Actual2	Real	28.0	0.0	0.005532166		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Par_Actual2	Real	32.0	0.0	0.04390869		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Set_de_Velocidad	Real	36.0	0.0	1200.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Atrunque_Motor	Bool	40.0	false	TRUE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Voltsje_Calculado	Real	42.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Nota. Variables y valores reales coinciden con el bloque SINA SPEED.

El traslado de estos datos hacia el otro bloque esp32, se observa en la figura 58, mediante el network que tiene la función de mandar estos datos del DB_CONTROL G120, al DB_dataesp32, por medio del bloque MOVE.

Figura 58

Bloque MOVE en funcionamiento

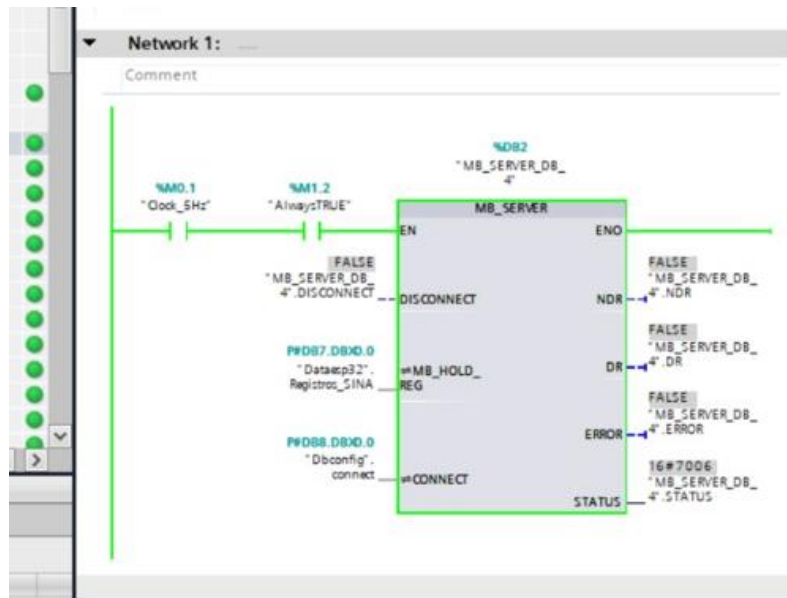


Nota. Traslado de variables mediante el bloque MOVE, para permitir la comunicación.

En la figura 59, se observa como estos datos están siendo comunicados a través del bloque MB_SERVER, para corroborar la correcta vinculación de los datos sí están enlazándose y comunicándose con el heltec, se revisó el status del bloque, el cual indicó que, el valor es de 7006, entonces se tuvo una correcta comunicación.

Figura 59

Bloque MB SERVER en funcionamiento.



Nota. Estableciendo la comunicación a través del bloque MB SERVER.

6.2 Interfaz HMI

Se siguió con la interfaz HMI, la cual, primero se ejecutó una prueba, colocando un valor de 500 en el set point, el encendido y apagado del motor, como se puede observar en la figura 60, está funcionando correctamente.

Figura 60

Interfaz completa del HMI



Nota. En caso de elegir la opción Ethernet se debe colocar la IP de la pantalla que se tenga. Adaptado de HMI KTP 700 [Fotografía], por Siemens, 2024, <https://support.industry.siemens.com/>

Ahora, para la siguiente prueba, se estableció un set point, de 1200, lo cual se refleja en la interfaz, como se puede observar en la figura 61 cambiando así, los valores de los IO fields, la gráfica empieza a interactuar, el motor se coloca de color verde, ya que se encuentra encendido.

Figura 61

HMI con un SP 1200



Nota. Una vez configurada el valor del set point, cambian los valores de las demás variables. Adaptado de HMI KTP 700 [Fotografía], por Siemens, 2024, <https://support.industry.siemens.com/>

Se agregó otro set point con un valor diferente, esta vez era de 1615, lo cual así mismo se puede observar en la figura 62 cambiando así los valores de los IO fields, dentro de la gráfica como hay un pico alto, ya que se aumentó el valor, el motor se ilustra de color verde, indicando un correcto funcionamiento.

Figura 62

Prueba HMI con un valor de 1615



Nota. La calibración es necesaria para asegurar que cada giro sea una vuelta completa. Adaptado de HMI KTP 700 [Fotografía], por Siemens, 2024, <https://support.industry.siemens.com/>

6.3 Comunicación Heltec

Como se observa en la figura 63, el microcontrolador heltec como cliente el cual es el TX, que también va a actuar como transmisor, recibe correctamente los datos que se están visualizando tanto en la simulación del bloque, y en la interfaz HMI.

Figura 63

TX Heltec V3



Nota. Transmisor de Heltec V3 TX con los datos establecidos en el HMI de 1200. Adaptado de WiFi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

El receptor está recibiendo correctamente estos datos, del transmisor Heltec, el cual empezó con la ejecución del primer valor que se asignó que es de 1200, es decir, ya encapsuló estos datos y comenzó con la transmisión del mismo, como se logra observar en la figura 64.

Figura 64

RX Heltec V3

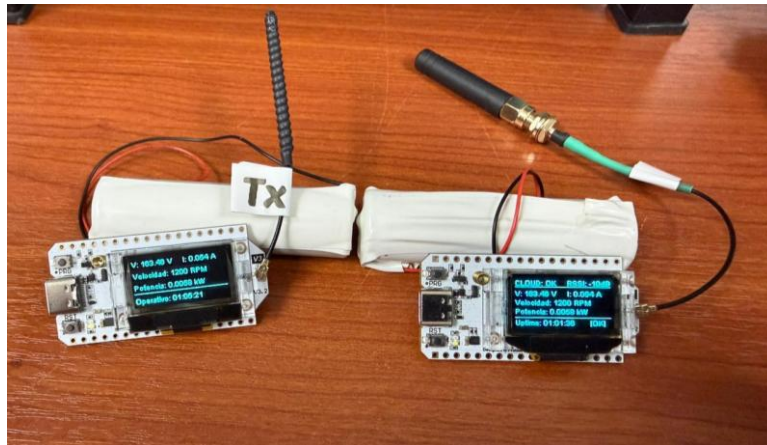


Nota. Receptor de Heltec V3 RX con los datos establecidos en el HMI. Adaptado de WiFi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

Tanto como el transmisor y el receptor tienen los mismos valores que se habían registrado, como se aprecia en la figura 65, esto quiere decir que se está cumpliendo correctamente el funcionamiento de dicho prototipo.

Figura 65

TX Heltec V3 y RX Heltec V3



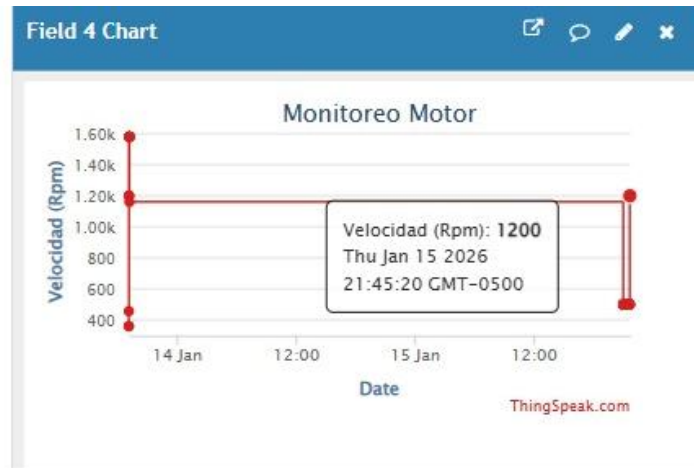
Nota. Receptor de Heltec V3 RX con los datos establecidos en el HMI. Adaptado de WiFi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

6.4 Dashboard

Los datos en la nube se redactaron correctamente, esto se puede ver en la figura 66 y en las gráficas, con un valor de 1200 RPM, el voltaje, corriente, y potencia, permaneció exactamente igual que los valores de PLC, HMI, HELTEC.

Figura 66

Voltaje con un valor de 1200 rpm

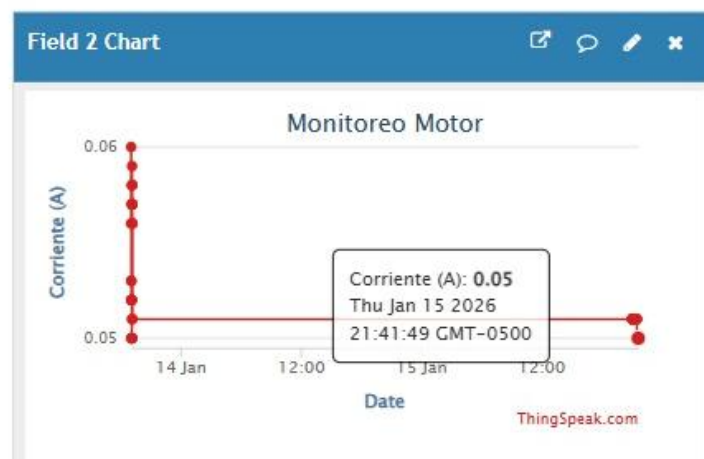


Nota. Valores que se reflejaron en la gráfica mediante la comunicación de los Heltec, en este caso es la velocidad.

Como se observa en la figura 67, se refleja la corriente en la gráfica, mostrando su comportamiento y estableciendo esa comunicación.

Figura 67

Corriente con un valor de 1200 rpm

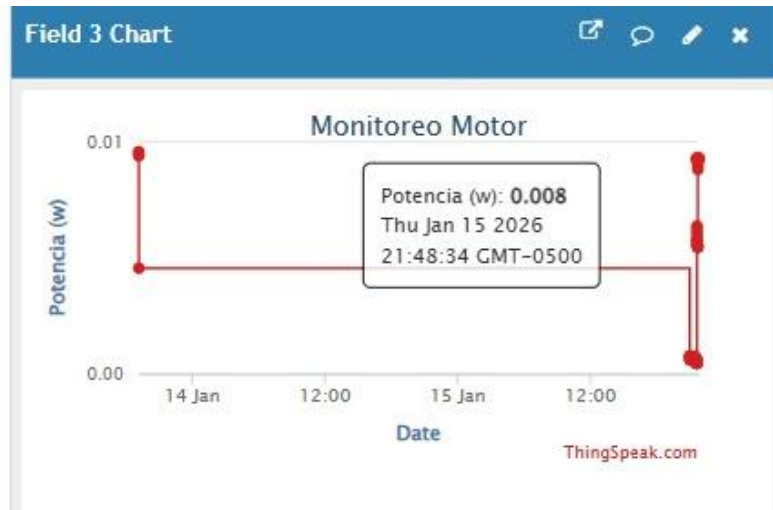


Nota. Se reflejó en la gráfica mediante la comunicación de los heltec, un valor de 0.05 A.

En la figura 68, se observa la potencia reflejada en el dashboard, lo que en comparación a los valores de TIA Portal siguen coincidiendo.

Figura 68

Potencia con un valor de 1200 rpm

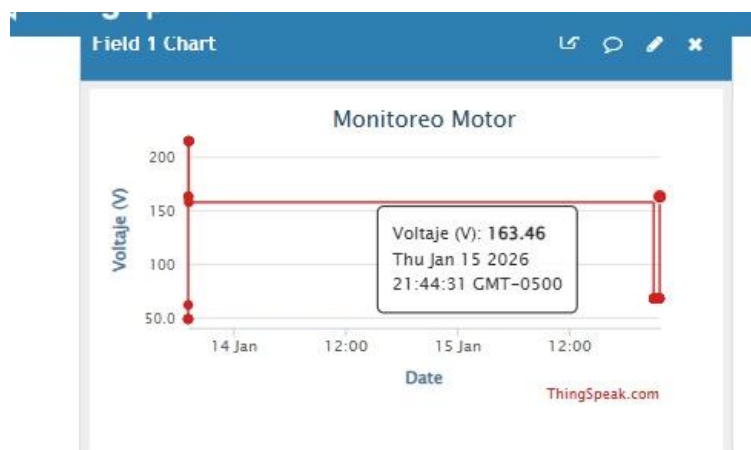


Nota. Se reflejó en la gráfica mediante la comunicación de los heltec, con un valor de 0.008 Watts.

Como se observa la figura 69, el valor del voltaje que fue calculado mediante el bloque CALCULATE del software TIA portal.

Figura 69

Voltaje con un valor de 1200 rpm



Nota. Se reflejó en la gráfica mediante la comunicación de los heltec un voltaje de 163.46V.

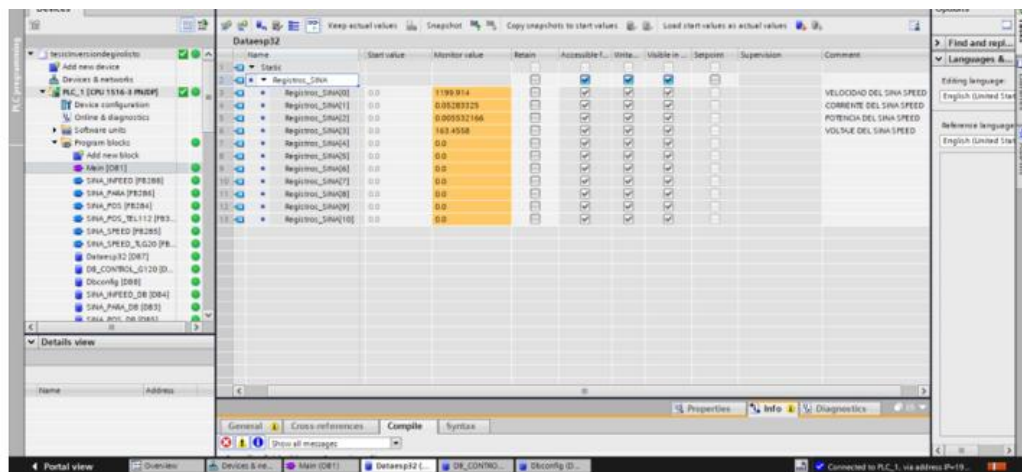
6.5 Validación experimental y comparativa de datos

Para verificar este funcionamiento y determinar que los datos y variables no se vean modificados, o sufran alteraciones, dado que, existen varios puntos por donde pasa esta información, la cual se puede comparar con los datos del TIA Portal, y que coincidan perfectamente, con los datos reflejados en la pantalla oled del heltec.

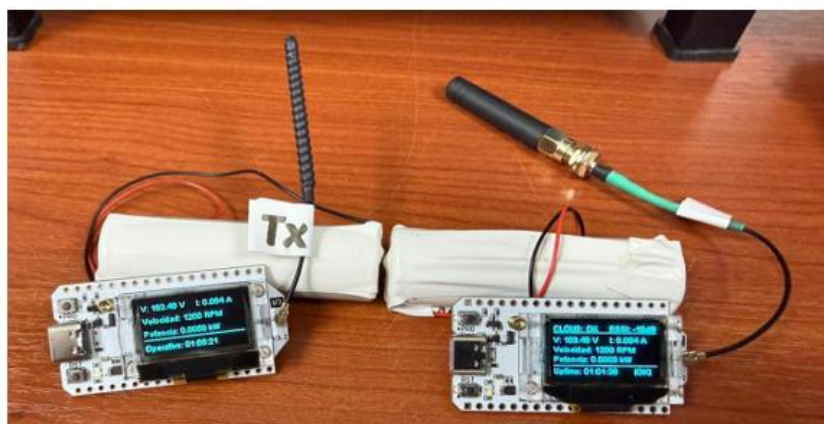
Así, cada uno de los registros a la vez, se comprueba cómo se ve en la figura 70, que no existen alteraciones ni modificaciones de los valores reales.

Figura 70

Comparación de registros



Name	Start value	Monitor value	Retain	Accessible	Write	Visible	Sequences	Supervision	Comment
Registros_SINA[0]	0.0	1199.914							VELOCIDAD DEL SINA SPEED
Registros_SINA[1]	0.0	0.00289323							CORRIENTE DEL SINA SPEED
Registros_SINA[2]	0.0	0.000532166							POTENCIA DEL SINA SPEED
Registros_SINA[3]	0.0	183.4358							VOLTAJE DEL SINA SPEED
Registros_SINA[4]	0.0	0.0							
Registros_SINA[5]	0.0	0.0							
Registros_SINA[6]	0.0	0.0							
Registros_SINA[7]	0.0	0.0							
Registros_SINA[8]	0.0	0.0							
Registros_SINA[9]	0.0	0.0							
Registros_SINA[10]	0.0	0.0							



Nota. Se observa que se cumple cada uno de los valores, y se están comunicando correctamente. Adaptado de WiFi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

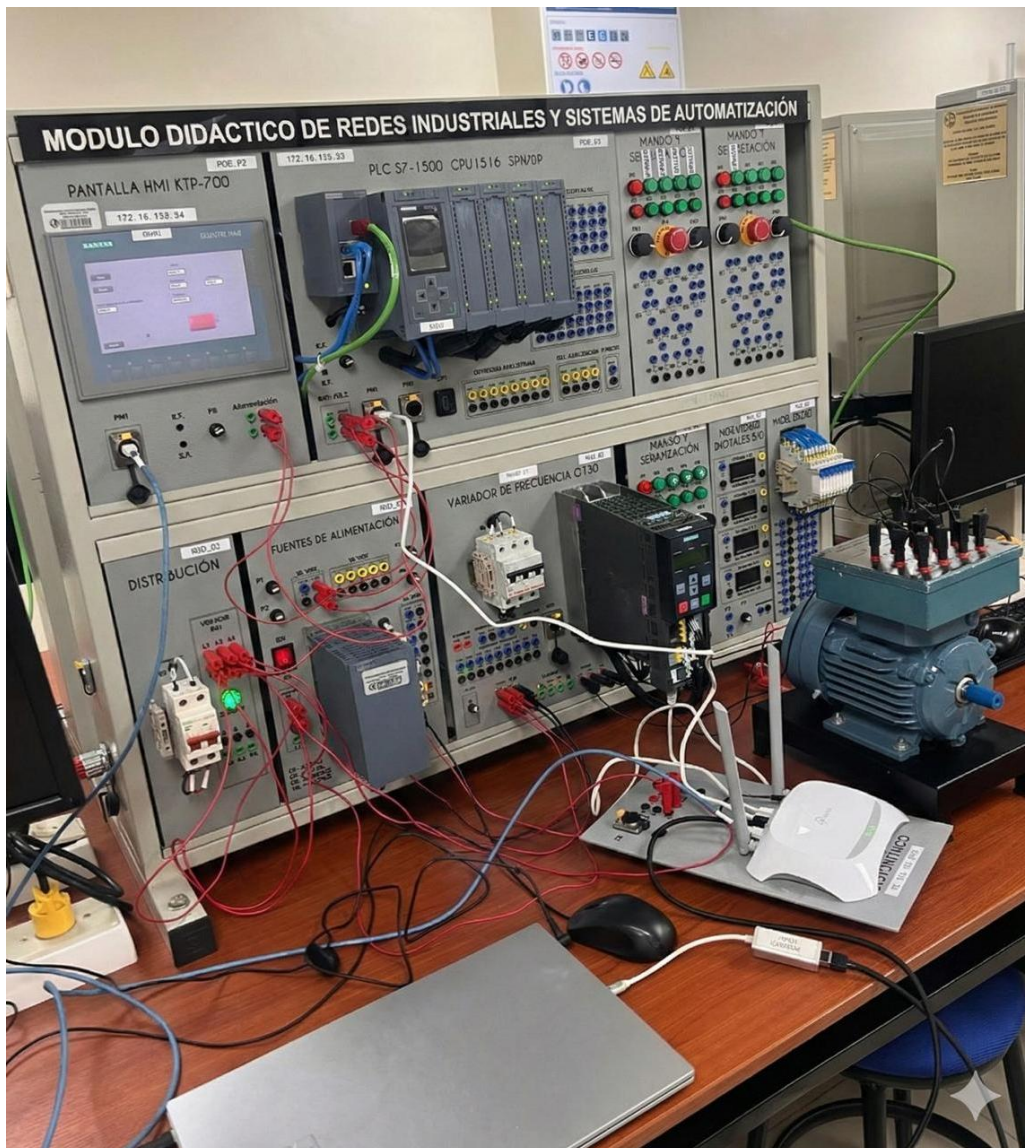
6.6 Implementación

La implementación final en el módulo de prácticas del laboratorio de Automatización Industrial 2, se puede observar en la figura 71.

En el cual se integró: PLC S7 – 1500, Variador de frecuencia G120x, TP LINK, Motor, Heltec V3.

Figura 71

Implementación en el Módulo de prácticas



Nota. Se estableció la comunicación entre el PLC, variador a través de ethernet al TP-Link, y los heltec a través de wifi.

En la figura 72, se realizó la parametrización del variador de frecuencia antes de que este pudiera controlar el motor.

- Se ingresó los datos de la placa del motor, en el asistente de puesta en marcha.
- Se seleccionó en el software del TIA Portal el telegrama de comunicación, el cual fue el número 20 para así realizar el intercambio de datos con el PLC
- Se asignó la IP y el nombre del dispositivo para así el PLC lo reconozca.

Figura 72

Parametrización del variador

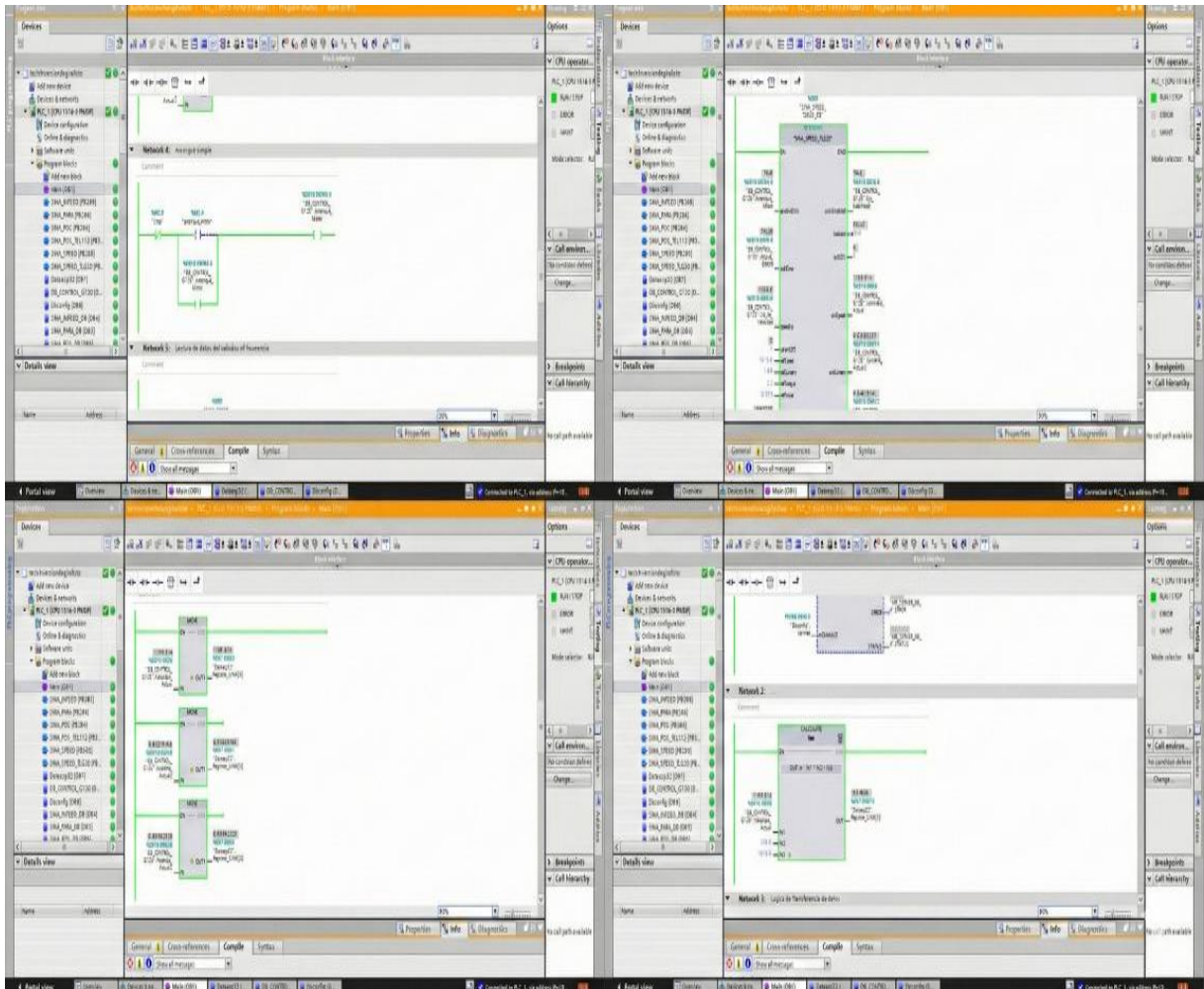


Nota. Se configuró en el variador los datos de la placa de motor, y reconocimiento en el software del TIA Portal. Adaptado de Sinamics G120X [Fotografía], por Siemens, 2024, <https://www.siemens.com>

Como se puede ver en la figura 73, se configuró cada uno de los elementos a necesitar en el software, netamente de la programación, como lo es el bloque SINA SPEED, MOVE, CALCULATE Y MB_SERVER.

Figura 73

Programación TIA Portal.

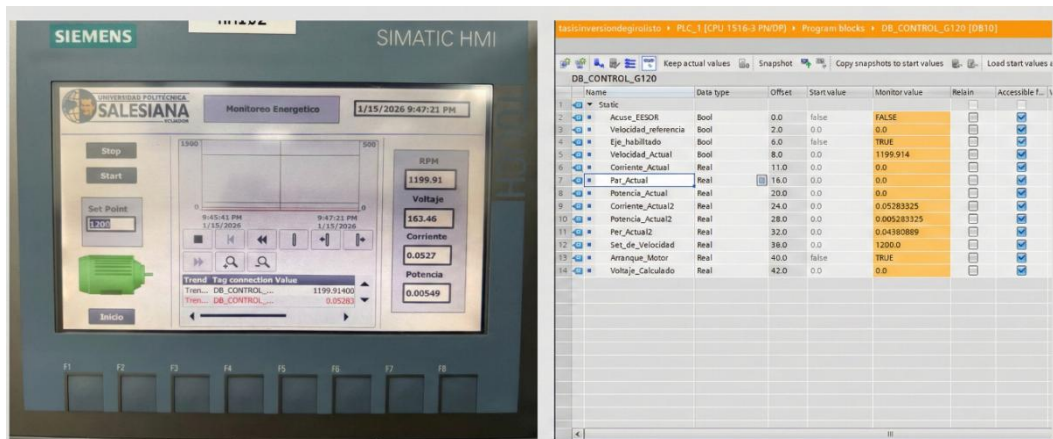


Nota. Programación, configuración, y reconocimiento de cada uno de los elementos en el TIA Portal.

Además de cada uno de los DB que se crearon, para la estructuración de cada elemento, también el HMI, como se puede ver en la figura 74.

Figura 74

Datos de DB y HMI

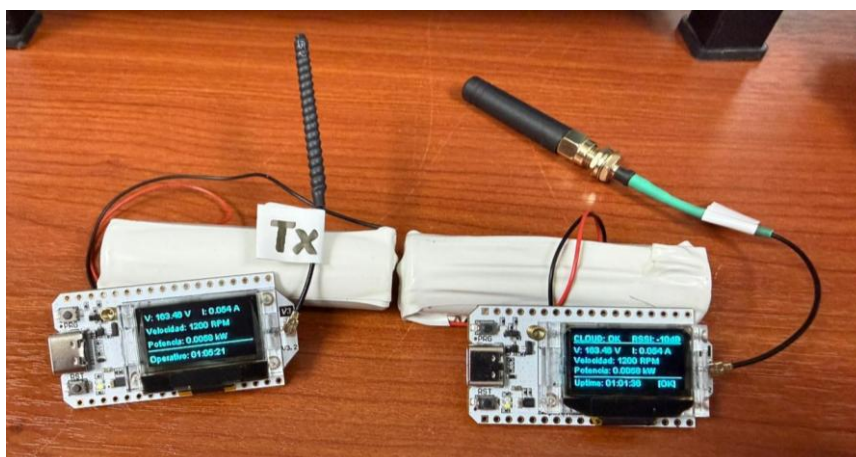


Nota. Valores de las variables coinciden en los valores de la interfaz, dando como conclusión que están correctamente funcionales. Adaptado de HMI KTP 700 [Fotografía], por Siemens, 2024, <https://support.industry.siemens.com/>

En la figura 75, se estableció la programación de cada uno de los microcontroladores, estableciendo uno como transmisor y otro como receptor.

Figura 75

Vista de Heltec cargada la programación



Nota. Se configuró las librerías y parámetros para realizar este prototipo. Adaptado de WiFi LoRa 32(V3) [Fotografía], por Heltec Automation, 2024, <https://heltec.org/project/wifi-lora-32-v3/>

7. CRONOGRAMA DE ACTIVIDADES

Se disponen las actividades a realizar durante los meses de noviembre, diciembre y enero en un período de cuatro semanas por cada mes, en la cual, se logra visualizar cada uno de los pasos que conlleva realizar el desarrollo del prototipo.

Tabla 1

Cronograma de actividades

<i>Actividades</i>	<i>Tiempo de duración</i>												
	<i>Meses</i>	<i>Noviembre</i>				<i>Diciembre</i>				<i>Enero</i>			
	<i>Semanas</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Revisión bibliográfica y definición de variables eléctricas.		X											
Diseño de la arquitectura del sistema y selección de componentes.		X											
Configuración del proyecto en TIA Portal y mapa de memoria del PLC.			X	X									
Desarrollo del firmware para los nodos de transmisión inalámbrica.				X									
Integración de la interfaz de comunicación (PLC - Puente PC - Transmisor).					X								
								X	X	X			
Implementación de la lógica de recepción y visualización de datos.								X	X				
Ejecución de pruebas experimentales de alcance y validación de datos.								X	X				
Validación de funcionalidades y ajustes.									X	X			
Análisis de resultados y redacción del documento final.												X	X

Nota. En esta tabla se desglosa de manera semanal cada actividad que se llevó a cabo durante los tres meses mencionados.

8. PRESUPUESTO

En la Tabla 2 se presenta el presupuesto sobre las horas de ingeniería invertidas al momento de desarrollar el trabajo de titulación, en donde se tuvo en consideración el salario básico unificado del presente periodo 2025.

Tabla 2

Cuadro de presupuesto

<i>DETALLE</i>	<i>CANTIDAD</i>	<i>COSTO UNITARIO</i>	<i>COSTO TOTAL</i>
<i>Horas de ingeniería</i>	<i>160h</i>	<i>\$2.87</i>	<i>\$459,2</i>
<i>Módulos Lora</i>	<i>2 unidades</i>	<i>\$50</i>	<i>\$100</i>
<i>Batería</i>	<i>2 unidades</i>	<i>\$15</i>	<i>\$30</i>
<i>Pantalla oled</i>	<i>1 unidad</i>	<i>\$10</i>	<i>\$10</i>
<i>TOTAL</i>			<i>\$599,2</i>

Nota. Cuadro de presupuesto.

9. CONCLUSIONES

- La programación en TIA Portal permitió el control eficiente del sistema de bombeo mediante el PLC S7-1500, utilizando bloques estandarizados como SINA_SPEED y MB_SERVER para gestionar el variador y la comunicación de datos en modos manual y automático.
- La implementación de la HMI facilitó la supervisión en tiempo real de variables críticas como corriente y velocidad, permitiendo una detección rápida de errores mediante la visualización de alarmas y estados del proceso para una corrección inmediata.
- La integración del PLC, variador G120X y módulos Heltec en una red gestionada por el router TP-Link validó un sistema híbrido capaz de controlar el motor trifásico y transmitir sus datos vía LoRa a puntos remotos, uniendo la robustez industrial con la versatilidad del IoT.

10. RECOMENDACIONES

- Asegurar la compatibilidad e instalación correcta de las librerías tecnológicas, tanto la "DriveLib" en TIA Portal como las librerías de Modbus y LoRa en el entorno del Heltec, ya que el uso de versiones incorrectas o bloques no actualizados impide la compilación del proyecto y bloquea la comunicación inicial entre los equipos.
- Estructurar las variables de proceso utilizando un Bloque de Datos (DB) global, en lugar de variables sueltas, ya que esto facilita enormemente la asignación del puntero en el bloque MB_SERVER y mantiene el mapa de memoria Modbus ordenado para futuras expansiones.
- Realizar pruebas de funcionamiento por etapas independientes, verificando primero que el PLC controle correctamente el variador y después integrando la comunicación con el Heltec, ya que intentar depurar todo el sistema simultáneamente hace muy difícil identificar si una falla proviene de la lógica del PLC, del cableado de red o del código del microcontrolador.

11. REFERENCIAS BIBLIOGRÁFICAS

- Alliance, G. S. (2023, 17 de diciembre). <https://www.globalseafood.org/advocate/energy-efficiency-pumping-systems-aquaculture-farms/>
- Arduino. (2025). <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/>
- Aude, J. (12 de Junio de 2024). *FOCUS*. <https://www.risoul.com.mx/blog/que-es-un-motor-de-induccion-y-como-funciona>
- Aula21. (2025). <https://www.cursosaula21.com/modbus-que-es-y-como-funciona/>
- Automation, H. (2025). *Products*. [//heltec.org/](https://heltec.org/)
- Blancarte, O. (2025). <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/p2p>
- Cardona, S. (06 de Septiembre de 2023). *Servomotores* <https://electrocentercol.com/blog/servomotores-tipos-construccion-trabajo-control-y-aplicaciones>
- Digikey. (2025). <https://www.digikey.es/es/resources/iot-resource-center/long-range-wireless-communication?srsId=AfmBOornxr24JR-2nS9pcRJBu1L5S1PRMmdP2Vn8mOvp20KCd9Q-Qyw>
- Ecuador, Cámara Nacional de Acuicultura. (2024). <https://www.cna-ecuador.com/camaron-cierra-2023-con-cifras-en-rojo-en-materia-economica-y-de-seguridad/>
- Elecproy. (13 de Junio de 2024). <https://elecproy.com/es/blog/plcs-definicion-partes-ventajas/>
- Exertherm. (2024). https://blog.exertherm.com/going-digital-and-future-proofing-electrical-assets?utm_term=industrial%20iot%20solutions&utm_campaign=Going+Digital+and+future+proofing+electrical+assets+-+WT+-+28Nov23+-+Blog&utm_source=adwords&utm_medium=ppc&hsa_acc=353199990
- FAO. (2024). <https://openknowledge.fao.org/items/92319005-6232-450f-8c75-4d4fcf24720d>
<https://www.fortunebusinessinsights.com/integrated-servo-motor-market-110555>
- Ignition. (10 de Abril de 2025). <https://inductiveautomation.com/resources/article/what-is-hmi>
- JRX automation s.r.o. (2025). <https://www.invtcz.cz/eshop/D220-sv-da200-0r4-2-s0>
- Logicbus. (22 de Enero de 2024). *Logicbus*. <https://es.linkedin.com/pulse/qu%C3%A9-es-hmi-definici%C3%B3n-tendencias-y-caracter%C3%ADsticas-clave-que-apuge>
- Logicbus. (2025). <https://www.logicbus.com.mx/Modbus>
- Megatronica. (2023). <https://megatronica.cc/producto/lora32-433mhz-esp32/?srsId=AfmBOoqpo-BILfnpMORkXZ0CoE2v9uUSDt1aSSRtCyOFSRrn-nM-GX1F>
- Muñoz, H. A. (11 de septiembre de 2024). <https://ru.dgb.unam.mx/bitstream/20.500.14330/TES01000857675/3/0857675.pdf>
- Panorama acuicola*. (2024). <https://panoramaacuicola.com/2024/09/11/evaluacion-de-la-ventaja-competitiva-de-la-industria-camaronera-de-ecuador-en-el-mercado-global-utilizando-el-indice-de-balassa-para-el-periodo-2018-2022/>
- PLC, S. (2025). <https://www.solisplc.com/tutorials/siemens-plcsims-new-interface-in-tia-portal-v18>
- Prodel, SA. (2022). *Prodel*. <https://www.prodel.es/subareas/maquinas-electricas/>
- Promelsa. (22 de Noviembre de 2023). <https://www.promelsa.com.pe/blog/post/que-es-motor-electrico-tipos.html?srsId=AfmBOoqJkm-Rt5BYC2Wx1X1OoGHnVwrOZ2jhM0ke6PZI-N2vLi8zIYkG>
- Siemens. (2011). <https://cache.industry.siemens.com/>
- Siemens. (2023). <https://sieportal.siemens.com/en-ww/support>
- Siemens. (2024). <https://docs.tia.siemens.cloud/>
- Siemens. (2024). <https://sieportal.siemens.com/en-ww/support>
- Siemens. (2025). <https://www.siemens.com/es/es/productos/automatizacion/sistemas/simatic/controladores-simatic/simatic-s7-1500.html>

Siemens. (2025). <https://www.siemens.com/es/es/productos/accionamientos/sinamics/baja-tension/sinamics-g/g120x.html>

Siemens. (2025). <https://cache.industry.siemens.com/>

SIMALEC. (2025). <https://simalec.com/product/hmi-modelo-pulgadas-7/>

Smart, M. (2024). <https://www.mokosmart.com/es/types-of-iot-modules/>

Transfer Multisort Elektronik. <https://www.tme.eu/es/news/library-articles/page/59558/campo-magnetico-sus-fuentes-y-propiedades/>

Uhu. (2024). <https://uhu.es/antonio.barragan/content/modbus-tcp>

Vazques y Rodriguez Ltd. (2024). <https://www.vasquezyrodriguez.com/bobinado-de-motores-electricos-errores-comunes-y-como-prevenirlos>

Velázquez, J. C. (2024). <https://www.fao.org/4/ab487s/>

Vencoel. (2025). <https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>

Wedbyn. (2024). <https://www.webdyn.com/es/articulos-de-radio/architecturas-lora/>

Weis, O. (19 de Octubre de 2021). https://www.eltima.com/es/article/rs485-communication-guide/?srsltid=AfmBOop1PACfMQVQonFRygYDC9__HRUkOmis7RncHt7dM6I7Q7pMYTid

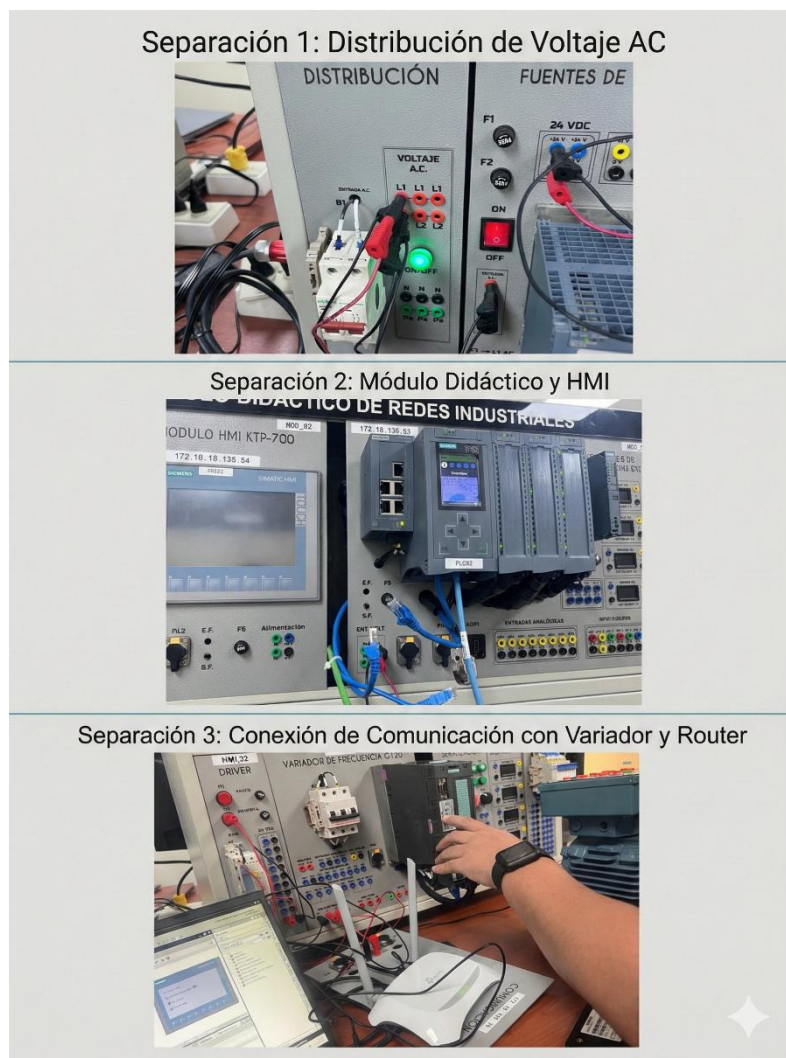
12. ANEXOS

En esta parte se incluyen los anexos que complementan en el desarrollo de la implementación del desarrollo del prototipo para el monitoreo energético, mostrando los elementos y los datos que fueron añadidos al módulo que se encuentra en el Laboratorio de Automatización Industrial II.

La figura 76 se observa la conexión de cada uno de los elementos en el módulo.

Figura 76

Conexiones de los módulos y elementos a utilizar.



Nota. Conexiones para el desarrollo del prototipo, y el correcto funcionamiento de la comunicación.

Código Heltec V3 Emisor.

```
#include <WiFi.h>
#include <ModbusIP_ESP8266.h>
#include <RadioLib.h>
#include <Wire.h>
#include "SSD1306Wire.h"

// --- CONFIGURACIÓN DE RED ---
const char* ssid      = "TP-Link_76F6";
const char* password = "33227856";
IPAddress plcIP(192, 168, 0, 10);

// --- HARDWARE HELTEC V3 ---
SX1262 radio = new Module(8, 14, 12, 13);
SSD1306Wire display(0x3c, 17, 18);
ModbusIP mb;

union FloatConverter {
    uint16_t reg[2];
    float valor;
};

// Función de tiempo de operación (Uptime)
String getUptime() {
    long t = millis() / 1000;
    int h = t / 3600;
    int m = (t % 3600) / 60;
    int s = t % 60;
    char buf[15];
    sprintf(buf, "%02d:%02d:%02d", h, m, s);
    return String(buf);
}

void setup() {
    Serial.begin(115200);

    // Encendido de periféricos V3
    pinMode(36, OUTPUT); digitalWrite(36, LOW);
    pinMode(21, OUTPUT); digitalWrite(21, LOW); delay(50); digitalWrite(21,
HIGH);

    display.init();
    display.flipScreenVertically();

    display.clear();
    display.setFont(ArialMT_Plain_10);
    display.drawString(0, 10, "TESIS: COELLO - FLORES");
    display.drawString(0, 25, "SISTEMA MONITOREO LORA");
}
```

```

display.display();
delay(3000);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    display.clear();
    display.drawString(0, 0, "ENLAZANDO - WiFi...");
    display.display();
}

mb.client();
radio.begin(915.0);
}

void loop() {
    mb.task(); // Mantiene la comunicación fluida

    if (!mb.isConnected(plcIP)) {
        mb.connect(plcIP);
    }
    else {
        static uint32_t lastMillis = 0;
        // Actualización cada 1 segundo para ver las variaciones rápidas
        if (millis() - lastMillis > 1000) {
            lastMillis = millis();

            uint16_t res[8];
            if (mb.readHreg(plcIP, 0, res, 8)) {
                FloatConverter conv;

                // 1. Velocidad (RPM) - registros 0 y 1
                conv.reg[0] = res[1]; conv.reg[1] = res[0];
                float vel = abs(conv.valor);

                // 2. Corriente (Amp) - registros 2 y 3
                conv.reg[0] = res[3]; conv.reg[1] = res[2];
                float amp = abs(conv.valor);

                // 3. Potencia (kW) - registros 4 y 5
                conv.reg[0] = res[5]; conv.reg[1] = res[4];
                float pot = abs(conv.valor);

                // 4. Voltaje (V) - registros 6 y 7
                conv.reg[0] = res[7]; conv.reg[1] = res[6];
                float vol = abs(conv.valor);

                // Filtro de seguridad para voltaje (elimina ruidos de bits)
                if (vol > 1000) vol = 0;
            }
        }
    }
}

```

```

// --- TRANSMISIÓN LORA CON ALTA PRECISIÓN ---
// V(2 dec), I(3 dec), RPM(0 dec), P(4 dec)
String paquete = String(vol,2) + "," + String(amp,3) + "," +
String(vel,0) + "," + String(pot,4);
radio.transmit(paquete);

// --- ACTUALIZACIÓN DE PANTALLA ---
display.clear();
display.setFont(ArialMT_Plain_10);

// Voltaje y Corriente (Precisión Industrial)
display.drawString(0, 0, "V: " + String(vol,2) + " V");
display.drawString(70, 0, "I: " + String(amp,3) + " A");

// Velocidad
display.drawString(0, 16, "Velocidad: " + String(vel,0) + "
RPM");

// Potencia
display.drawString(0, 32, "Potencia: " + String(pot,4) + " kW");

// Tiempo de operación
display.drawHorizontalLine(0, 46, 128); // Línea decorativa
display.drawString(0, 50, "Operativo: " + getUptime());

display.display();
}
}
}
}

```

Código Heltec V3 Receptor

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <RadioLib.h>
#include <Wire.h>
#include "SSD1306Wire.h"

// =====
// 1. CREDENCIALES (Verificadas)
// =====
const char* ssid      = "j*****1";
const char* password  = "1*****8";
const char* apiKey    = "KVX9HMHNRHRAA6Y7G";

// =====
// 2. HARDWARE Y VARIABLES
// =====
SX1262 radio = new Module(8, 14, 12, 13);
SSD1306Wire display(0x3c, 17, 18);

String v = "0.00", i = "0.000", p = "0.0000", vel = "0";
float rssi = 0;
int lastHttpCode = 0;
unsigned long lastCloudUpdate = 0; // Temporizador para ThingSpeak

// Función Tiempo de Operación
String getUptime() {
    long t = millis() / 1000;
    int h = t / 3600;
    int m = (t % 3600) / 60;
    int s = t % 60;
    char buf[15];
    sprintf(buf, "%02d:%02d:%02d", h, m, s);
    return String(buf);
}

// =====
// 3. SETUP
// =====
void setup() {
    Serial.begin(115200);
    pinMode(36, OUTPUT); digitalWrite(36, LOW);
    pinMode(21, OUTPUT); digitalWrite(21, LOW); delay(50); digitalWrite(21,
HIGH);

    display.init();
    display.flipScreenVertically();
    display.setFont(ArialMT_Plain_10);
```

```

// CONEXIÓN WIFI (Hotspot)
WiFi.disconnect(true);
delay(1000);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

display.clear();
display.drawString(0, 0, "Buscando Hotspot...");
display.drawString(0, 15, "Red: " + String(ssid));
display.display();

int intentos = 0;
while (WiFi.status() != WL_CONNECTED && intentos < 20) {
    delay(1000);
    intentos++;
    display.drawString(intentos * 4, 30, ".");
    display.display();
}

radio.begin(915.0); // Frecuencia LoRa
}

// =====
// 4. LOOP (RECEPCIÓN Y NUBE)
// =====
void loop() {
    String paquete;
    int state = radio.receive(paquete);

    if (state == RADIOLIB_ERR_NONE) {
        rssi = radio.getRSSI();

        // DESEMPAQUETAR CSV (V, I, RPM, P)
        int c1 = paquete.indexOf(',');
        int c2 = paquete.indexOf(',', c1 + 1);
        int c3 = paquete.indexOf(',', c2 + 1);

        if (c1 != -1 && c2 != -1 && c3 != -1) {
            v = paquete.substring(0, c1);
            i = paquete.substring(c1 + 1, c2);
            vel = paquete.substring(c2 + 1, c3);
            p = paquete.substring(c3 + 1);
        }

        // ENVÍO A THINGSPEAK (Cada 16 seg para respetar el límite gratuito)
        if (WiFi.status() == WL_CONNECTED && (millis() - lastCloudUpdate >
16000)) {
            lastCloudUpdate = millis();

```

```

    HTTPClient http;
    String url = "http://api.thingspeak.com/update?api_key=" +
String(apiKey) +
        "&field1=" + v + "&field2=" + i + "&field3=" + p +
"&field4=" + vel;

    http.begin(url);
    lastHttpCode = http.GET();
    http.end();
}
}

// --- DISEÑO VISUAL PROFESIONAL ---
display.clear();
display.setFont(ArialMT_Plain_10);

// Encabezado: Estado Red y Señal
display.drawString(0, 0, (WiFi.status() == WL_CONNECTED ? "CLOUD: OK" :
"CLOUD: OFF"));
display.drawString(70, 0, "RSSI: " + String(rssi, 0) + "dBm");
display.drawHorizontalLine(0, 11, 128);

// Parámetros (Misma precisión que Emisor)
display.drawString(0, 14, "V: " + v + " V");
display.drawString(70, 14, "I: " + i + " A");
display.drawString(0, 26, "Velocidad: " + vel + " RPM");
display.drawString(0, 38, "Potencia: " + p + " kW");

// Pie de Pantalla
display.drawHorizontalLine(0, 52, 128);
display.drawString(0, 53, "Uptime: " + getUptime());

// Indicador de éxito ThingSpeak
if (lastHttpCode == 200) display.drawString(100, 53, "[OK]");
else if (lastHttpCode != 0) display.drawString(100, 53, "[ER]");

display.display();
}

```