

UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA AGRÍCOLA



**PUESTA EN MARCHA Y EVALUACIÓN DE UN SISTEMA DE MEDICIÓN
DE EFICIENCIA ENERGÉTICA EN SISTEMAS DE BOMBEO.**

IGNACIO LEÓN SEPÚLVEDA FERNÁNDEZ.

PROYECTO DE TÍTULO
PRESENTADO A LA FACULTAD DE
INGENIERÍA AGRÍCOLA DE LA
UNIVERSIDAD DE CONCEPCIÓN,
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL AGRÍCOLA.

CHILLÁN-CHILE

2025

**PUESTA EN MARCHA Y EVALUACIÓN DE UN SISTEMA DE MEDICIÓN
DE EFICIENCIA ENERGÉTICA EN SISTEMAS DE BOMBEO.**

Aprobado por:

Christian Correa Farías
Ingeniero Civil Agrícola, Dr.
Profesor Asociado.

Profesor Guía

Wilson Esquivel Flores
Ingeniero Civil Mecánico, Dr.
Profesor Asistente.

Profesor Asesor

David Lara Castells
Ingeniero Civil Eléctrico, Dr.
Profesor Asociado.

Profesor Asesor

Christian Correa Farías
Ingeniero Civil Agrícola, Dr.
Profesor Asociado.

Director de Departamento

María Eugenia González Rodríguez
Ingeniero Agrónomo, Ph. D.
Profesor Asociado.

Decana

INDICE DE MATERIAS.

	Página
RESUMEN	1
ABSTRACT.....	2
1. INTRODUCCIÓN.....	3
2. OBJETIVOS.	5
2.1. Objetivo General.....	5
2.2. Objetivos específicos.....	5
3. ANTECEDENTES GENERALES.....	6
3.1. Eficiencia energética en sistemas bombeo.....	6
3.1.1. Selección de componentes.....	7
3.1.2. Optimización del diseño.....	7
3.1.3. Manejo y mantenimiento.....	8
3.1.4. Sistemas de Control.....	8
3.2. Tecnologías basadas en IoT en eficiencia energética.....	9
4. METODOLOGIA.....	14
4.1. Sección 1: Construcción banco de pruebas de bombeo.	14
4.1.1. Arquitectura del sistema de bombeo.....	14
4.1.2. Selección de componentes.....	15
4.1.3. Diseño del sistema de bombeo.....	17
4.1.4. Instalación del Sistema de bombeo.	19
4.1.5. Instalación eléctrica del sistema de bombeo.....	20
4.1.6. Configuración de los variadores.....	20
4.2. Sección 2: Definición de variables y criterios de evaluación.....	21
4.2.1. Variables de medición.....	22
4.2.2. Indicadores de rendimiento y criterio de evaluación.....	22
4.2.3. Escenarios de prueba.....	25
4.3. Sección 3: Implementación del sistema de medición.....	26
4.3.1. Arquitectura del sistema de medición.....	26
4.3.2. Componentes de Sensorización y Software.....	27
4.3.3. Diagrama de flujo y esquemático del circuito electrónico.....	31

4.3.4.	Configuración Plataforma Thingsboard.....	32
5.	RESULTADOS.....	33
5.1.	Construcción banco de pruebas de bombeo.....	33
5.2.	Implementación del sistema de medición.....	36
6.	DISCUSIÓN.....	44
6.1.	Comportamiento del sistema de bombeo.....	44
6.2.	Comportamiento del sistema de monitoreo.....	45
6.3.	Comparativa de los escenarios de prueba.....	47
7.	CONCLUSIONES.....	50
8.	LITERATURA CITADA.....	51
9.	APENDICES Y ANEXOS.....	54

INDICE DE TABLAS.

En el texto	Página
Tabla 1. Criterio de configuración de los variadores de frecuencia (VDF).....	21
Tabla 2. Criterios seleccionados para la evaluación de la eficiencia energética en el sistema de bombeo.....	24
Tabla 3. Escenario de pruebas a realizar en el sistema de bombeo.....	25
En el Apéndice	Página
Tabla 4. Componentes del banco de pruebas.	54
Tabla 5. Accesorios utilizados en la construcción del banco.....	55
Tabla 6. Componentes del sistema de monitoreo.	56
Tabla 7. Eficiencia estándar según la norma IEC 60034-30-1 para la clase IE ₂	65
Tabla 8. Datos de rendimiento proporcionados por el fabricante.	66
Tabla 9. Obtención de P ₁ y el criterio PEE.	66
Tabla 10. Datos de tensión y consumo de corriente de las bombas, con el S obtenido.....	67
Tabla 11. Obtención del criterio FP.....	67
Tabla 12. Obtención del criterio Rc.	67
Tabla 13. Obtención de la velocidad de giro por punto de operación de la bomba CP190.....	69

INDICE DE FIGURAS.

En el texto	Página
Figura 1. Consumo eléctrico estimado de motores por diferentes sectores.....	6
Figura 2. Visualización de la eficiencia de bombeo del día anterior y la media de los últimos siete días	11
Figura 3. Parámetros de la página web.....	12
Figura 4. Interfaz gráfica de parámetros en tiempo real.	13
Figura 5. Curvas de prestaciones de las bombas utilizadas con su punto máximo de eficiencia, en contraste con la curva característica del sistema en paralelo.....	18
Figura 6. Organización de la arquitectura del sistema.....	27
Figura 7. Diagrama de flujo de datos de ThingsBoard.....	30
Figura 8. Interfaz IDE Arduino.....	30
Figura 9. Esquema del banco de pruebas propuesto.....	33
Figura 10. Representación tridimensional del banco de pruebas..	34
Figura 11. Instalación completa del banco de pruebas.....	35
Figura 12. Arquitectura del sistema.....	36
Figura 13. Recepción de datos en Thingsboard.....	37
Figura 14. Panel de visualización creado para la monitorización.....	38
Figura 15. Monitoreo de la presión y caudal en la plataforma web.....	38
Figura 16. Notificación de alarma en la plataforma móvil.....	39
Figura 17. Datos obtenidos en la medición de los indicadores de rendimiento durante el Escenario 1.....	41
Figura 18. Datos obtenidos en la medición de los indicadores de rendimiento durante el Escenario 2.....	43
Figura 19. Comparativa entre las curvas presión - caudal teóricas de las bombas y el sistema en paralelo con las curvas obtenidas en el Escenario 1.....	45

En el Apéndice	Página
Figura 20. Vista frontal del diseño.....	57
Figura 21. Vista trasera del diseño.....	57
Figura 22. Vista superior del diseño.....	58
Figura 23. Vista lateral del diseño.....	58
Figura 24. Dimensiones de la base de los caudalímetros.....	58
Figura 25. Dimensiones de la base de las bombas.....	59
Figura 26. Dimensiones de la estructura de soporte del estanque.....	59
Figura 27. Estructuras de soporte instaladas en el banco de pruebas.....	60
Figura 28. Vista lateral del sistema de bombeo construido.....	60
Figura 29. Vista trasera y frontal del sistema de bombeo construido.....	60
Figura 30. Conexión eléctrica del sistema de bombeo.....	61
Figura 31. Conexión estrella de bomba CP190 y especificaciones técnicas.....	61
Figura 32. Conexión estrella de bomba CP170 y especificaciones técnicas.....	62
Figura 33. Conexión eléctrica de las bombas a la caseta de bombeo.....	62
Figura 34. Diagrama esquemático del cable de conexión UWR00468-2.....	63
Figura 35. Cable de conexión PC – Variador UWR00468-2 construido.....	63
Figura 36. Configuración realizada en los VDF.....	65
Figura 37. Circuito electrónico del sistema de medición.....	70
Figura 38. Diagrama de Flujo del código fuente.....	72
Figura 39. Creación del dispositivo y su token de acceso.....	73
Figura 40. Creación del panel de visualización de datos.....	74
Figura 41. Configuración de widgets de visualización.....	76
Figura 42. Configuración y visualización de widget de tabla.....	77
Figura 43. Configuración widget de gráfico.....	78
Figura 44. Visualización widget de gráfico.....	78
Figura 45. Configuración y visualización de widget led indicador.....	79

Figura 46. Creación de reglas de alarma en la plataforma..... 80

INDICE DE ECUACIONES.

	Página
Ecuación 1. Eficiencia Energética de bombeo.....	23
Ecuación 2. Ratio de Consumo.....	23
Ecuación 3. Factor de Potencia.....	23
Ecuación 4. Eficiencia del motor.....	65
Ecuación 5. Potencia Aparente.....	65
Ecuación 6. Relación de semejanza de bombas centrifugas.....	67

GLOSARIO.

Eficiencia Energética	Capacidad de un sistema para realizar una tarea utilizando la menor cantidad de energía posible, maximizando el rendimiento y minimizando el consumo energético.
IoT	(Internet de las cosas) red de dispositivos conectados que intercambian datos a través de Internet, permitiendo la automatización y el control remoto.
MODBUS	Protocolo de comunicación utilizado en sistemas de automatización industrial para la transmisión de datos entre dispositivos electrónicos, como PLCs, sensores y actuadores, permitiendo la comunicación maestro-esclavo.
WSN	(Wireless Sensor Network) red de sensores inalámbricos que monitorean condiciones ambientales y transmiten datos a un sistema central para su análisis.

PUESTA EN MARCHA Y EVALUACIÓN DE UN SISTEMA DE MEDICIÓN DE EFICIENCIA ENERGÉTICA EN SISTEMAS DE BOMBEO

COMMISSIONING AND EVALUATION OF AN ENERGY EFFICIENCY MEASUREMENT SYSTEM IN PUMPING SYSTEMS

Palabras claves: Eficiencia energética, IoT, bombeo, rendimiento.

RESUMEN

El objetivo principal de esta tesis fue el desarrollo e implementación de un sistema de medición de eficiencia energética en sistemas de bombeo. Para ello, se utilizó un Microcontrolador ESP32 en conjunto con la plataforma IoT Thingsboard, permitiendo el monitoreo en tiempo real de variables críticas e indicadores de rendimiento. La metodología empleada incluye la adquisición de datos a través de sensores que se comunican mediante el protocolo MODBUS, los cuales se envían en tiempo real utilizando el protocolo MQTT vía Wifi. El sistema fue implementado en un banco de pruebas con dos bombas en paralelo, modelos CP 190 y CP 170, y se evaluó en dos condiciones experimentales: Escenario 1, donde las bombas operaron a su frecuencia nominal (50 Hz), y Escenario 2, en el que se ajustó la frecuencia de la bomba CP190. Los resultados obtenidos demostraron que el ajuste de frecuencia incrementó la eficiencia energética del sistema, mejorando el PEE en un 1,7% y reduciendo el gasto energético 0,7 kW-h por metro cúbico bombeado en el punto más eficiente. En conclusión, el uso del sistema de monitoreo permitió optimizar el rendimiento del sistema de bombeo, reduciendo el consumo energético y promoviendo una mayor sostenibilidad operativa.

COMMISSIONING AND EVALUATION OF AN ENERGY EFFICIENCY MEASUREMENT SYSTEM IN PUMPING SYSTEMS

Keywords: Energy efficiency, IoT, pumping, performance.

ABSTRACT

The main objective of this thesis was the development and implementation of an energy efficiency measurement system for pumping systems. For this purpose, an ESP32 Microcontroller was used in conjunction with the IoT Thingsboard platform, allowing real-time monitoring of critical variables and performance indicators. The methodology employed includes data acquisition through sensors that communicate via MODBUS protocol, which are sent in real time using the MQTT protocol via Wifi. The system was implemented in a test bench with two pumps in parallel, models CP 190 and 170, and was evaluated in two experimental conditions: Scenario 1, where the pumps operated at their nominal frequency (50 Hz), and Scenario 2, where the frequency of the CP190 pump was adjusted. The results obtained showed that the frequency adjustment increased the energy efficiency of the system, improving the PEE by 1,7% and reducing the energy expenditure by 0,7 kW-h per cubic meter pumped at the most efficient point. In conclusion, the use of the monitoring system allowed optimizing the performance of the pumping system, reducing energy consumption and promoting greater operational sustainability.

1. INTRODUCCIÓN.

Los recursos hídricos ofrecen una importante gama de servicios que son cruciales para un desarrollo sostenible. El consumo de energía del sector del agua a nivel mundial representó 120 Mtep¹ en 2014, principalmente en forma de electricidad, lo que corresponde al 4% del consumo total mundial de electricidad (Luna *et al.*, 2019). Según el informe de la Agencia Internacional de la Energía (IEA, 2016, citado en Luna *et al.*, 2019) y estudios como el de Lam *et al.* (2017, citado en Luna *et al.*, 2019), el consumo eléctrico relacionado con los sistemas de bombeo representa la mayor parte del consumo energético en todo el sector del agua, llegando a alcanzar hasta el 90% (Grundfos, 2004, citado en Luna *et al.*, 2019).

La modernización de los equipos de bombeo mediante soluciones más eficientes, como la incorporación de motores con sistemas de velocidad variable y controles avanzados, puede generar un ahorro de energía significativo. En este contexto, la integración de tecnologías avanzadas, junto con la monitorización constante del rendimiento, contribuye de manera significativa a mejorar la eficiencia energética. Tecnologías emergentes como el Internet de las cosas (IoT) y las redes de sensores inalámbricos (WSN, por sus siglas en inglés) han surgido como soluciones clave para abordar los desafíos relacionados con el uso eficiente del agua y la energía.

¹ Millones de toneladas equivalentes al petróleo

El éxito de los sistemas de monitoreo actuales, en las redes de distribución de agua, radica en la adopción de tecnologías avanzadas, especialmente aquellas vinculadas al IoT. Esta tecnología permite la conectividad inalámbrica entre los diferentes componentes del sistema, facilitando la recopilación y procesamiento de datos en tiempo real para optimizar la toma de decisiones y mejorar la eficiencia operativa (Perez-Padillo *et al.*, 2020).

La implementación del IoT, en gestión energética, permite el monitoreo en tiempo real y control eficiente en los sistemas de bombeo convencionales, aumentando la capacidad de detectar y abordar rápidamente problemas como: fugas, mal funcionamiento de equipos y/o anomalías en patrones de uso. La digitalización del ecosistema, a través de IoT, mejora la visibilidad de los activos, optimiza el control del consumo de la energía y facilita la generación de ahorros económicos en el área (López-Morales *et al.*, 2021).

En este contexto, y con el objetivo de evaluar el impacto de estas tecnologías en la eficiencia operativa de los sistemas de bombeo, esta tesis se centró en el diseño e implementación de un sistema de monitoreo energético basado en IoT.

2. OBJETIVOS.

2.1. Objetivo General.

Implementar un sistema de medición de eficiencia energética en sistemas de bombeo.

2.2. Objetivos específicos.

- Construir un banco de pruebas que permita medir la eficiencia energética en sistemas de bombeo.
- Configurar un sistema de medición que sea capaz de recopilar y analizar datos de manera inalámbrica.
- Implementar indicadores y criterios estándar de evaluación para medir la eficiencia energética en sistemas de bombeo.

3. ANTECEDENTES GENERALES.

3.1. Eficiencia energética en sistemas bombeo.

La energía es un recurso fundamental tanto para el sector doméstico como para el industrial. El alarmante ritmo de aumento de la población mundial y de la economía ha tenido un impacto enorme en el consumo de energía. Según la Agencia Internacional de la Energía (IEA, 2011), los motores eléctricos consumen el 46% de la electricidad generada en el mundo y representan casi el 70% del consumo total de electricidad en las industrias. Según el informe elaborado por la Comisión Europea (EC, 2001) el sistema de bombeo representa casi el 22% de la energía suministrada por motores eléctricos en el mundo, como se aprecia en la Figura 1.

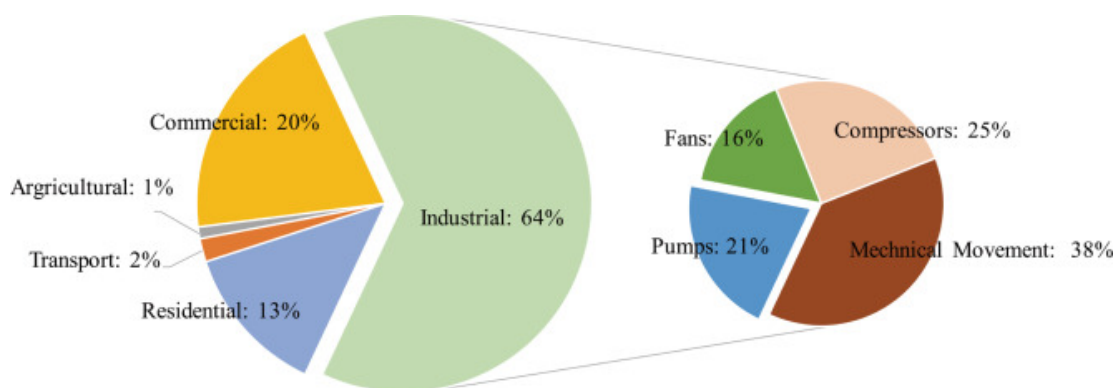


Figura 1. Consumo eléctrico estimado de motores por diferentes sectores. (Abdelaziz *et al.*, 2011)

Los sistemas de bombeo son un conjunto de equipos que utilizan energía mecánica para mover líquidos a través de tuberías, generando presión y caudal. Son esenciales en diversas aplicaciones, como el abastecimiento de agua y la industria. Su funcionamiento se basa en la conversión de energía para transportar fluidos de un lugar a otro.

Dado el papel fundamental de los sistemas de bombeo en diversas aplicaciones, optimizar su eficiencia energética se ha convertido en una prioridad tanto económica como ambiental. En este contexto, las oportunidades para mejorar la eficiencia energética están presentes en: optimización del diseño, selección de componentes, manejo y mantenimiento y los sistemas de control.

3.1.1. Selección de componentes.

El diseño del sistema define los límites de funcionamiento de un sistema de bombeo, estableciendo valores de caudal y presión de operación. De acuerdo con esto se seleccionan los componentes que puedan soportar dicha ejecución, en especial las bombas, dado que son las principales influyentes en la eficiencia energética. Una elección adecuada de esta reduce los costos operativos, logrando una reducción del consumo de energía del 2% al 10 % reemplazando las bombas existentes con bombas eficientes.

3.1.2. Optimización del diseño.

En cuanto al diseño, existen algunas medidas que pueden disminuir el gasto energético en la distribución del agua, tales como: el aumento del diámetro en

sistemas de tuberías, para reducir la fricción, disminución de márgenes de seguridad, con el objeto de acercar el punto de funcionamiento, al punto máximo de eficiencia y finalmente reducir el número de accesorios.

3.1.3. Manejo y mantenimiento.

Por otro lado, el manejo y mantenimiento permiten corregir problemáticas que reducen la eficiencia en los equipos de bombeo. El mantenimiento preventivo incluye acciones como la limpieza de filtros, verificación del desgaste de sellos mecánicos y alineación de componentes para minimizar pérdidas por fugas, prolongando la vida útil del sistema y asegurando un funcionamiento eficiente y sostenible (Dynopro, 2024). Por su parte, el mantenimiento predictivo optimiza la gestión operativa al monitorear en tiempo real parámetros clave, como vibraciones, temperatura y consumo energético. A través de sensores y análisis de datos, se anticipan fallos, reduciendo tiempos de inactividad y costos de reparación.

3.1.4. Sistemas de Control.

Por último, los métodos de control que se han incorporado en los sistemas de bombeo se basan en la utilización de variadores de frecuencia (VFD). La potencia eléctrica consumida por un motor depende de la carga que este debe mover y de la velocidad a la que opera. Al disminuir la frecuencia, la velocidad del motor se reduce, y como muchas cargas industriales (como bombas) tienen un comportamiento proporcional a la ley de los cubos, la potencia

requerida disminuye significativamente con la reducción de la velocidad. Es decir, si la velocidad se reduce al 50%, la potencia consumida puede disminuir hasta un 12,5% ($0,5^3$). Por lo tanto, al variar la frecuencia y reducir la velocidad del motor, el consumo de potencia eléctrica disminuye considerablemente, lo que mejora la eficiencia energética del sistema. Su uso habilita la posibilidad de ejercer cargas variables, implicando altos ahorros de energía, en comparación con otros métodos de control convencionales, como válvulas de estrangulamiento y sistemas de control de derivación.

3.1.5. Tecnologías basadas en IoT en eficiencia energética.

Un ecosistema de IoT está compuesto por dispositivos inteligentes habilitados para la web, que utilizan sistemas integrados, como procesadores, sensores y hardware de comunicación, para recopilar, transmitir y actuar sobre los datos que obtienen. Estos dispositivos IoT comparten los datos recopilados mediante una puerta de enlace IoT o dispositivos perimetrales, que luego envían la información a la nube o a un servidor local para su análisis. En ocasiones, estos dispositivos también se comunican entre sí, utilizando la información intercambiada para realizar acciones específicas (Laghari *et al.*, 2023).

La implementación de esta tecnología, en la optimización de la eficiencia energética, en sistemas de bombeo, supone ventajas comparativas respecto a los métodos convencionales, como monitoreo en tiempo real, programación de mantenimientos preventivos, aprendizaje automático, gestión remota, almacenamiento de datos en la nube, entre otros beneficios. En la literatura,

existen diversos trabajos, ejemplificando el funcionamiento de IoT en este campo y sus beneficios, los cuales se describen a continuación:

El proyecto de López-Morales *et al.* (2021) se centra en el diseño e implementación de una plataforma tecnológica basada en IoT para optimizar la eficiencia energética en sistemas de bombeo de agua utilizados en riego agrícola. Con el objetivo de abordar los altos costos asociados al consumo energético en pozos de riego, la investigación propone un modelo de gestión integral que combina datos heterogéneos como sensores de campo, tarifas eléctricas dinámicas y características técnicas de las bombas.

El sistema incorpora indicadores clave como el indicador de eficiencia energética del bombeo (EEP, por sus siglas en inglés), que mide el desempeño operativo de las bombas, y el costo de bombeo por metro cúbico, para evaluar la relación entre consumo energético y volumen de agua extraída. Además, integra alertas preventivas basadas en umbrales críticos para anticipar fallos y propone medidas correctivas, como la optimización de horarios de operación según tarifas eléctricas y la instalación de equipos adicionales, como convertidores de frecuencia.

La plataforma (Figura 2) fue implementada en una comunidad de riego en Murcia, España, demostrando su efectividad al reducir los costos energéticos en un 10% y lograr un ahorro económico anual de 43500 €. Así, este proyecto demuestra que las tecnologías basadas en IoT no solo permiten monitorear en tiempo real, sino que tomar decisiones de operación y actuar los sistemas de forma remota.



Figura 2. Visualización de la eficiencia de bombeo del día anterior y la media de los últimos siete días. (López-Morales *et al.*, 2021)

Otra experiencia, realizada en México, desarrolló un sistema para evaluar la eficiencia electromecánica de equipos de bombeo agrícolas, integrando sensores avanzados para medir parámetros eléctricos (como voltaje y corriente) e hidráulicos (caudal y presión), junto con una Unidad Terminal Remota (RTU) que procesa los datos y calcula la eficiencia en tiempo real. Este sistema, diseñado como un equipo portátil y robusto con protección IP67, facilita su implementación en diferentes sitios y permite la supervisión continua a través de una interfaz web, como se muestra en la Figura 3. Durante las pruebas realizadas en el laboratorio de hidráulica del IMTA, se obtuvo una eficiencia promedio del 52%, con una potencia eléctrica medida de 26 kW y una potencia hidráulica proporcionada de 13,5 kW. Estos resultados evidencian el potencial del sistema para optimizar la gestión del agua y la

energía en zonas agrícolas, promoviendo una operación más sostenible y eficiente (Ruiz Carmona., 2014). De este modo, este proyecto demuestra la factibilidad de monitorear vía IoT parámetros eléctricos que permiten medir la eficiencia energética de un sistema de bombeo.

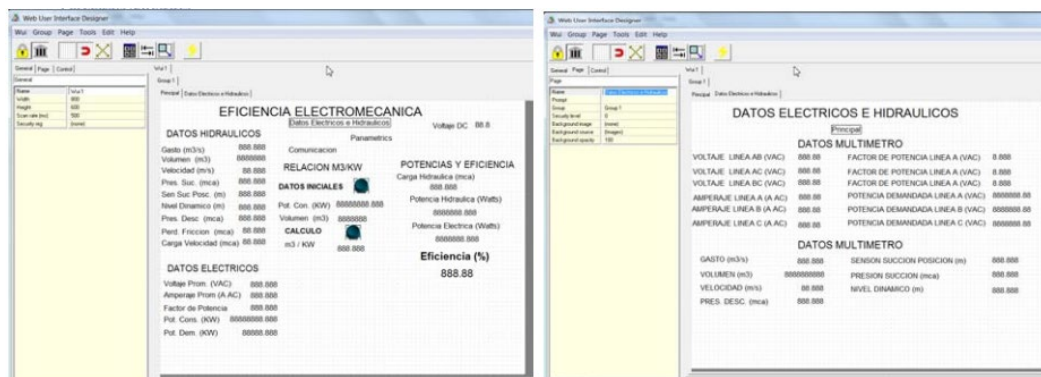


Figura 3. Parámetros de la página web. (Ruiz Carmona., 2014)

Por último, el proyecto realizado por Gallego (2020) implementa un sistema avanzado basado en IoT para optimizar la eficiencia de instalaciones de bombeo de aguas subterráneas. Mediante la sensorización de parámetros clave y utilizando una pasarela Siemens Simatic IoT2040 con Node-RED, el sistema recopila y procesa datos en tiempo real, almacenándolos en una base de datos InfluxDB. Los indicadores de desempeño utilizados fueron: eficiencia hidráulica de pozo, rendimiento de bomba, ratio de consumo y desviación de rendimiento, parámetros que son visualizados a través de una plataforma web como se muestra en la Figura 4. La validación del sistema se realizó en dos casos de estudio: el Sondeo Villamera (1400 habitantes), donde se detectó un elevado gasto eléctrico y disminución en la extracción de agua, y el Sondeo Catogrí (25000 habitantes), donde se identificó un incremento en el costo de

extracción por reducción del caudal. En ambos casos, los resultados teóricos coincidieron con los valores reales, demostrando la efectividad del sistema para la evaluación y mejora de la eficiencia operativa, así como su potencial para generar ahorros energéticos significativos en el marco de la industria 4.0.

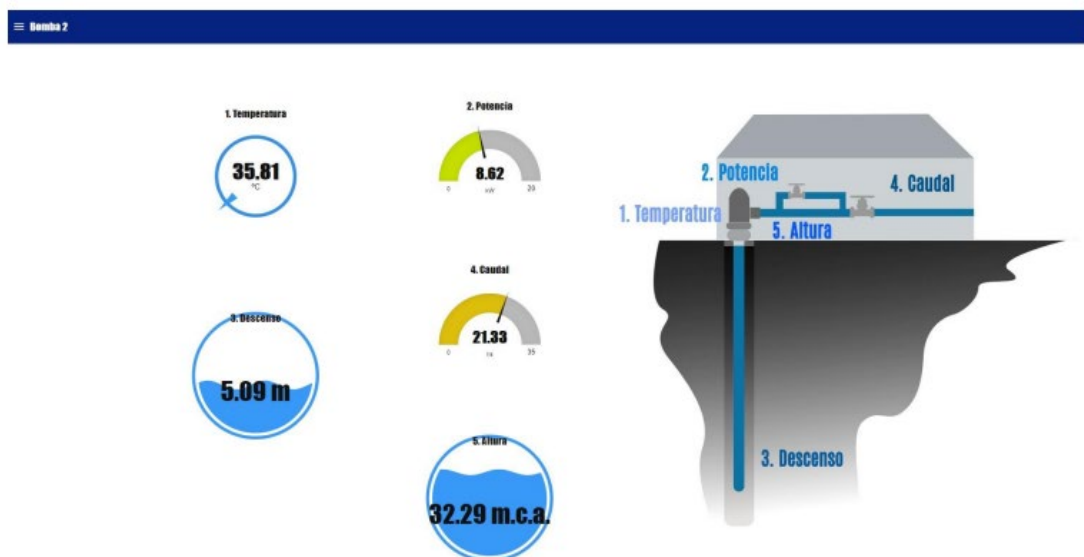


Figura 4. Interfaz gráfica de parámetros en tiempo real. (Gallego, 2020)

Los estudios presentados evidencian el impacto positivo del IoT en sistemas de bombeo, mejorando la eficiencia energética, reduciendo costos y facilitando el mantenimiento predictivo. El monitoreo en tiempo real permite identificar problemas y aplicar soluciones efectivas, optimizando el rendimiento de los equipos. Su validación, en entornos reales, confirma su aplicabilidad y potencial de ahorro, mejorando la gestión operativa y promoviendo un uso más eficiente de los recursos. Desarrollar un sistema de medición de eficiencia energética basado en IoT representa una oportunidad clave para optimizar el funcionamiento de los equipos de bombeo y maximizar su desempeño.

4. METODOLOGIA.

El procedimiento realizado en este proyecto fue dividido en tres secciones, esto es: Construcción banco de pruebas de bombeo; Definición de variables y criterios de evaluación; Implementación del sistema de medición.

4.1. Sección 1: Construcción banco de pruebas de bombeo.

En esta sección se presenta todos los pasos que se llevaron a cabo para la fabricación del banco de pruebas de bombeo, desde la definición de la esquemática, la selección de los componentes y el diseño de este.

4.1.1. Arquitectura del sistema de bombeo.

El esquema realizado fue ideado para formar un sistema de bombeo en paralelo, teniendo como régimen principal la utilización de equipos que se encuentren disponibles. Con esto, se plantearon algunos principios para formar la esquemática del sistema propuesto, estos son: conexión de bombas en paralelo con un acceso directo por parte de las bombas a la fuente de agua, ubicación de los variadores lo más cercana a las bombas posible, ubicación de caudalímetros y manómetros antes de la suma de los caudales individuales por bomba, posicionamiento del filtro al final del sistema en conjunto con una válvula de estrangulamiento. Estos requisitos proporcionaron los límites y condiciones para el trazado del esquema.

4.1.2. Selección de componentes.

Un sistema de bombeo estándar contiene elementos claves como son las bombas, sistemas de filtrado, caudalímetros, entre otros, que permiten un correcto funcionamiento del transporte de agua. Los equipos descritos a continuación, fueron seleccionados en base a la disponibilidad de los componentes en el Laboratorio de Robótica de la Universidad de Concepción y a su compatibilidad entre ellos. Los componentes principales son:

Bombas Centrifugas CP: con el objeto de recrear funcionamientos típicos de sistemas de bombeo en paralelo, fueron elegidas dos bombas centrifugas de la marca Pedrollo. Estas bombas pertenecen a los modelos CP170 y CP190, las cuales no son idénticas, por lo que su uso está condicionado a la utilización de variadores de frecuencia.

Variador de frecuencia IQ Pump Micro: Con el propósito de operar un sistema en paralelo con bombas no idénticas, fue necesario la incorporación de variadores de frecuencia en este proyecto. Sujeto a la disponibilidad, se agregaron 2 variadores de la marca Yaskawa, diseñados especialmente para sistemas de bombeo. Estos equipos ajustan automáticamente las condiciones de funcionamiento de la bomba a los cambios del proceso, manteniendo una protección constante de la bomba y el sistema. Además, ofrecen facilidad en la configuración con su software integrado.

Caudalímetros: Para obtener las mediciones de caudal, es necesario contar con un sensor estable, que ofrezca una precisión adecuada y sea resistente.

De acuerdo con la disponibilidad de equipos se utilizaron dos caudalímetros, uno de tipo Tangencial y el otro de tipo Woltman, ambos con diámetro nominal de 50 mm y un emisor de pulsos electromagnético, el cual emite un pulso cada 100 litros, dato clave para la determinación del caudal.

Filtro AZUD Helix Automatic 201DLP MG: la incorporación de un filtro al sistema de bombeo mantiene la calidad y limpieza del fluido circulante, lo que conlleva la prevención de daños a otros equipos. En base a esto, se escogió un filtro de discos maniobrado con válvulas de 2", que posee un sistema de autolimpieza eficiente de baja presión (1,5 bar) permitiendo un máximo ahorro de agua.

Manómetros de glicerina: conocer el estado de la presión en el sistema de bombeo es primordial, dado que ayuda a la prevención de fugas o subidas de presión, como a la mantención de un rendimiento óptimo. Por ende, fueron seleccionados tres manómetros de glicerina, los cuáles fueron ubicados dos a las salidas de las bombas y el último en el filtro. De esta manera se monitorea la presión in situ y se conocen las pérdidas de carga generadas con el diferencial de presión.

Estanque de agua: Para proporcionar agua al sistema se necesita de un estanque resistente que puede aportar un caudal de forma continua al sistema, por lo que se utilizó un depósito de agua que está hecho de acero inoxidable y posee una forma rectangular. Sus dimensiones son de 50 cm x 72 cm de base con una altura de 60 cm.

4.1.3. Diseño del sistema de bombeo.

El banco de pruebas se diseñó en base a una estructura metálica móvil, distribuyendo los equipos según la arquitectura del sistema definida. Dado su propósito experimental, se priorizó optimizar la ubicación de los componentes, de tal forma que se garantice un flujo continuo y sin interrupciones, evitando problemas hidráulicos y cumpliendo con los protocolos de instalación establecidos (Anexo A). La representación del modelo obtenido se realizó a través del software de diseño Sketchup®, en el cual se implementaron las medidas descritas, obteniendo una configuración eficiente y funcional del banco de pruebas.

La selección de la tubería se realizó de acuerdo con la medida máxima compatible con los equipos y una presión nominal que soporte la presión ejercida por las bombas. Seleccionando tuberías de PVC, con diámetro de 63 mm y una presión nominal de 6 bar (PN6). Además, se consideró que las pérdidas de carga serían mínimas y no afectarían significativamente el rendimiento del banco de pruebas. Este enfoque permitió centrarse en otros parámetros críticos del sistema, alineándose con los objetivos del proyecto.

El comportamiento del banco de pruebas en relación con las curvas presión – caudal, se definió bajo los rangos presentados en la curva de prestaciones de las bombas, en conjunto con la curva resultante del sistema en paralelo.

La Figura 5, presenta la curva característica del sistema en paralelo, la cual consta de 2 partes; la parte A, donde solo opera la bomba CP190, y la parte

B, donde operan ambas bombas. Ya que se quiere operar el sistema en paralelo, se definió que el sistema opere solo en la sección B. De acuerdo con esto, se estableció un rango operativo de presión hasta 40 metros (4 bar) y el intervalo de caudales entre 70 y 300 L min⁻¹.

Los puntos eficientes teóricos por bomba también se presentan en la Figura 5, y estos se utilizaron como referencia al evaluar el comportamiento del Escenario 1, dado que esta prueba funciona a frecuencia nominal (50 Hz), que es lo indicado por el fabricante. De esta forma, se pudo determinar si las condiciones presentadas en el sistema de bombeo muestran un comportamiento similar o diferente a lo que indica el fabricante.

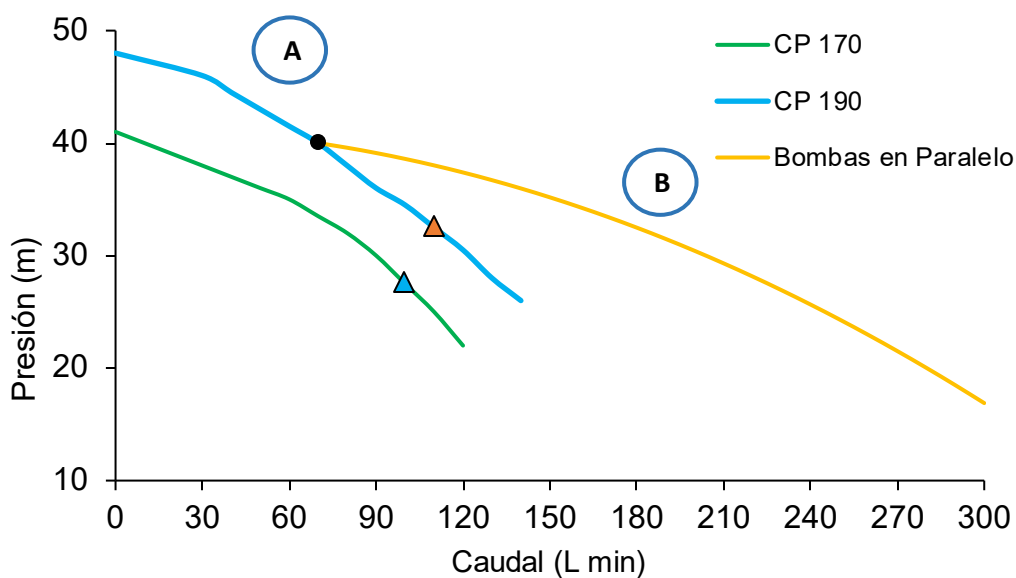


Figura 5. Curvas de prestaciones de las bombas utilizadas con su punto máximo de eficiencia (representados por marcadores triangulares Δ), en contraste con la curva característica del sistema en paralelo.

4.1.4. Instalación del sistema de bombeo.

El armado del banco de pruebas se llevó a cabo considerando algunas etapas previas en relación con el acondicionamiento de la estructura móvil. Estas etapas consistieron en el pulido de los perfiles metálicos, la soldadura de estructuras de soporte y el pintado de todo el armazón metálico.

Se decidió la incorporación de 3 estructuras de soporte que contengan a los equipos de mayor envergadura cuales son; bombas, caudalímetros y estanque. La adecuación de la estructura móvil finalizó con el pintado de esta, utilizando un color azul brillante.

El acoplamiento hidráulico del sistema se realizó considerando tanto el modelo propuesto en la sección de diseño, como los aspectos técnicos de instalación recomendados por los fabricantes. De este modo, se seleccionó, los *fittings* hidráulicos, elementos de fijación mecánicos (pernos, tuercas, golillas), tuberías, pegamento para PVC y teflón de acuerdo con estos criterios.

El APENDICE A1 detalla todos los *fittings* utilizados y cantidad de tubería en la construcción del banco de pruebas. Por otro lado, en el APENDICE A2 se pueden encontrar los diseños realizados en las estructuras de soporte, como las vistas obtenidas del diseño 3D y obtención final del sistema de bombeo propuesto.

4.1.5. Instalación eléctrica del sistema de bombeo.

La conexión eléctrica, se realizó en el gabinete de control, donde se encontraban los variadores de frecuencia (VDF). Las entradas disponibles de conexión utilizadas fueron las siguientes: alimentación trifásica de 380 VAC, Bomba 1 a 380 VAC, Bomba 2 a 380 VAC y transmisores de presión.

De acuerdo con los diagramas de cableado sugeridos por el fabricante del VDF (Yaskawa), se seleccionó los cables a utilizar, los cuales fueron calibre 6 mm² con rangos de 600 VAC y soporte de temperatura hasta 105 °C, utilizando terminales de ojo AWG 10 para la conexión de las bombas y terminales de punta AWG 10 para las conexiones dentro de la caseta de control.

El enlazamiento de las bombas a la red se realizó mediante conexión estrella a 400 VAC (recomendado por el fabricante), con una salida de 3 fases más tierra física. En cuanto a los transmisores de presión, la conexión fue mediante el pin de alimentación de 24 VCC y el pin de salida de corriente de 4 – 20 mA a su lugar definido en la caseta.

En el APENDICE A3 se presenta el diagrama de cableado eléctrico obtenido, como las conexiones físicas de las bombas a la red.

4.1.6. Configuración de los variadores.

Para configurar los variadores de frecuencia (VDF) según las especificaciones de las bombas y los métodos de control, se utilizó el software proporcionado por la marca Yaskawa denominado DriveWizard iQpump, el cual admite una

conexión alámbrica mediante un computador con el variador, permitiendo cambiar parámetros, realizar configuraciones y monitorear el sistema.

La configuración se realizó respetando los valores de referencia entregados por los fabricantes, en conjunto con los puntos de control definidos según el diseño creado (describiendo los criterios en la Tabla 1). Por lo que se utilizó la información proveniente de la ficha técnica de las bombas, el manual de usuario del variador y la información técnica de los transductores de presión.

Tabla 1. Criterio de configuración de los variadores de frecuencia (VDF).

Configuración	Parámetros	Origen de la información.
Variables Eléctricas	Voltaje nominal, corriente nominal, frecuencia nominal, voltaje máximo, potencia nominal.	Placa de datos de cada bomba.
Presión	Definir unidades de presión (bar), escalar el rango de presión del transductor, definir PID de control.	Información técnica de transductores de presión.
Comunicación MODBUS	Darle una dirección MODBUS a cada variador.	Información técnica de los variadores.

Las configuraciones específicas realizadas por VDF se pueden encontrar en APENDICE A4, el cual detalla la conexión realizada y los parámetros configurados por variador.

4.2. Sección 2: Definición de variables y criterios de evaluación.

En esta segunda sección se determinó las variables y los parámetros de rendimiento que permiten la evaluación del sistema de bombeo propuesto, definiendo criterios de evaluación acordes al sistema.

4.2.1. Variables de medición.

La eficiencia energética del sistema depende directamente de las siguientes variables; presión, caudal y eléctricos. La medición de estas variables permite caracterizar el comportamiento del sistema de bombeo, por lo que cada variable se monitoreó en los siguientes puntos estratégicos:

Presión: zona de descarga de cada bomba individual, cercana al tablero eléctrico para una conexión expedita a VDF.

Caudal: zona de descarga de cada bomba individual con un tramo recto en la entrada y salida de cada medidor de flujo, para proporcionar una medición correcta, según restricciones descritas por el fabricante.

Datos eléctricos: directamente de VDF, donde se mide voltaje, corriente, frecuencia y potencia activa.

4.2.2. Indicadores de rendimiento y criterio de evaluación.

Las variables que agrupan la presión, el caudal y los datos eléctricos determinan el funcionamiento de manera eficiente de un sistema de bombeo en términos de energía, por lo que las variaciones de estos alteran el desempeño del sistema de manera crítica. Existen diversos indicadores de desempeño, que permiten conocer el estado en que se encuentre el banco de pruebas, de entre los cuales, se seleccionó aquellos basados en parámetros eléctricos e hidráulicos. En específico:

Eficiencia energética de bombeo (PEE): es la relación entre potencia hidráulica suministrada (P_h) y potencia activa (P_1) de la estación de bombeo.

$$PEE (\%) = \frac{P_h \text{ (kW o HP)}}{P_1 \text{ (kW o HP)}} \quad [1]$$

Interpretar el PEE implica evaluar cuánta de la energía eléctrica se está utilizando efectivamente para bombear el fluido deseado. Una eficiencia más alta significa que la bomba está utilizando la energía de manera más efectiva, lo cual se traduce en menores costos operativos y menor impacto ambiental.

Ratio de consumo (R_c): es el cociente entre la potencia activa medida en la bomba (P_1) y el caudal bombeado medido (Q).

$$R_c \text{ (kWh m}^{-3}\text{)} = \frac{P_1 \text{ (kW)}}{Q \text{ (m}^3\text{h}^{-1}\text{)}} \quad [2]$$

El R_c indica si hay sobreconsumo por un bajo rendimiento total, si hay problemas con el sistema electromecánico. Un valor más bajo muestra que se está utilizando menos potencia para alcanzar el caudal necesario, lo que generalmente se asocia con una mayor eficiencia energética.

Factor de potencia (FP): es la relación entre la energía que extrae la red (S) y la energía útil que obtenemos de su funcionamiento (P_1).

$$FP = \frac{P_1 \text{ (kW)}}{S \text{ (kVA)}} \quad [3]$$

El FP refleja la eficiencia o rendimiento eléctrico de un sistema eléctrico. Un factor cercano a 1 indica un uso eficiente de la energía, mientras que un valor bajo indica que hay una mayor cantidad de energía desperdiciada.

El criterio de evaluación, definido en la Tabla 2, se estableció según las condiciones operativas del sistema, es decir, en base a los datos técnicos de las bombas. Por lo que se empleó la información entregada por el fabricante, estableciendo los límites entre un funcionamiento óptimo o insuficiente.

Tabla 2. Criterios seleccionados para la evaluación de la eficiencia energética en el sistema de bombeo.

Indicador	Criterios de evaluación		
	PEE teórico (%)	Rc (kW h m ⁻³)	FP
Bomba CP190	≥ 25,48 %	< 0,40	≥ 0,70
Bomba CP170	≥ 25,50 %	< 0,35	≥ 0,71
Sistema	≥ 25,50 %	< 0,38	-

El criterio de evaluación para PEE y FP se realizó estimando su valor numérico teórico en el punto más eficiente, según la curva de prestaciones. Estableciendo dicho valor como límite entre un comportamiento óptimo o insuficiente. El criterio de Rc se determinó con el promedio entre los puntos de menor caudal y el más eficiente de cada bomba, según la curva de prestaciones. Estableciendo un valor crítico que muestra el límite entre un funcionamiento correcto y un comportamiento insuficiente.

En APENDICE B1 se presenta el procedimiento realizado para la obtención de los criterios de evaluación definidos en la Tabla 2.

4.2.3. Escenarios de prueba.

Se estableció dos escenarios de prueba (véase Tabla 3) para el sistema de bombeo, con el fin de observar su comportamiento bajo distintas condiciones de operación.

Tabla 3. Escenario de pruebas a realizar en el sistema de bombeo.

Escenario	Frecuencia de giro (Hz)	Estrangulamiento
1	Ambas bombas a 50 Hz	Múltiples puntos de presión
2	Ajuste de la bomba CP190	Múltiples puntos de presión

El Escenario 1 simuló el uso del banco de pruebas a una frecuencia nominal de 50 Hz, operando en diversos puntos de presión y caudal mediante el estrangulamiento de una válvula, con el objetivo de observar el rendimiento del sistema al funcionar con dos bombas diferentes en paralelo.

El Escenario 2 ajustó la frecuencia de la bomba de mayor potencia (CP190) para que opere de manera similar a su contraparte (CP170), también operando en diversos puntos de presión y caudal mediante el estrangulamiento de una válvula. Todo esto se realizó con el fin de observar el rendimiento del sistema tras el ajuste, esperando un comportamiento similar al de bombas idénticas conectadas en paralelo.

El ajuste de la bomba CP190 se realizó por medio de las ecuaciones de semejanza en bombas presión - velocidad de giro (H-N), en donde se definió un nuevo valor de la frecuencia a operar de 45 Hz. El cálculo realizado mostró un error cuadrático medio (RMSE) de 1,43 m, que reflejó una estimación

bastante cercana al comportamiento de la bomba CP170. El procedimiento realizado se puede encontrar en APENDICE B2.

4.3. Sección 3: Implementación del sistema de medición.

En esta última sección, se presenta los procesos efectuados al implementar el sistema de medición, desde la arquitectura del sistema de medición, hasta la visualización de los indicadores en la nube.

4.3.1. Arquitectura del sistema de medición.

Para definir la arquitectura del sistema de medición, se consideró varios criterios clave que guiaron la composición de esta. Se priorizó la selección de sensores que ofrezcan alta precisión y fiabilidad de acuerdo con la disponibilidad existente, asegurando su compatibilidad con el microcontrolador ESP32 y el protocolo de comunicación MODBUS RTU. Se buscó garantizar una conexión WiFi estable y de alta disponibilidad, seleccionando el protocolo MQTT por su eficiencia en la transmisión de datos y seguridad de la información transmitida. Se definió la utilización de una interfaz de usuario intuitiva capaz de procesar y visualizar datos en tiempo real, optando por la plataforma Thingsboard, la cual se adecua a estas características siendo flexible, escalable y de código abierto.

De acuerdo con estos principios, se estructuró la arquitectura del sistema de medición como muestra la Figura 5, por medio de 3 capas que agrupan cada elemento seleccionado según su función específica y su rol dentro del sistema.



Figura 6. Organización de la arquitectura del sistema.

Capa de percepción: agrupa los sensores y actuadores utilizados tales como, contadores de pulsos, variadores de frecuencia, sensores de presión. Estos se conectan al dispositivo maestro (ESP32), permitiendo la recopilación de la información por comunicación MODBUS RTU mediante la interfaz RS-485.

Capa de red: permite la conexión vía wifi del microcontrolador ESP32 con Thingsboard, en conjunto con la administración de los datos mediante el protocolo de comunicación MQTT, todo esto codificado en el IDE Arduino.

Capa de aplicación: ofrece servicios y funciones para la visualización de los datos recopilados mediante Thingsboard, permitiendo el procesamiento y control de los datos medidos vía remota y en tiempo real.

4.3.2. Componentes de Sensorización y Software.

Se realizó la descripción brevemente de los componentes seleccionados que cumplen los requisitos planteados en la arquitectura del sistema y se encuentran disponibles para su uso, los cuales son:

Transmisores de presión: se utilizó transductores de presión analógicos con salida de corriente de 4 a 20 mA, alimentados con un voltaje entre 12 a 36 VDC. El rango de medición de presión es de 0 a 6 bar, siendo la salida de corriente lineal respecto a la presión.

Contador de pulsos: este contador funciona con un voltaje entre 12-36 VDC, contando de manera precisa y fiable los pulsos percibidos. Presenta una batería interna la cual guarda el último valor percibido. Se comunica mediante MODBUS a través de la interfaz RS-485.

Fuente de poder: El DRC-40A es una fuente de 40 W con salida única y función de cargador de batería (UPS). Ofrece un voltaje de salida de 13,8 VDC y una corriente nominal de 1,9 A, con un rango de corriente de 0 a 2,9 A. El rango de voltaje de entrada es de 90 a 264 VAC, y también puede aceptar 127 a 370 VDC. Esta fuente de alimentación está diseñada para aplicaciones de respaldo de energía, proporcionando energía confiable durante interrupciones de AC.

Convertidor de voltaje: este convertidor de voltaje de corriente continua de la marca CPT recibe un voltaje de entrada de 12 VDC, transformando la salida de este a un voltaje de 5 VDC con 3 A como corriente máxima, ofreciendo una potencia de salida de 15 W.

Microcontrolador ESP32: creado por Espressif Systems, es un microcontrolador de bajo costo y consumo, con capacidades WiFi y Bluetooth de modo dual. Incorpora un procesador Tensilica Xtensa LX6 de uno o dos núcleos, con una velocidad de hasta 240 MHz. Diseñado para dispositivos

móviles, portátiles y aplicaciones IoT, destaca por su eficiencia energética, gracias a características como activación de reloj de resolución fina, múltiples modos de energía y escalado dinámico de potencia. Lo que lo hace adecuado para enviar información de sensores a la nube de forma continua.

Conversor serial TTL a RS-485: permite la comunicación entre dispositivos con señales TTL y redes RS485, proporcionando una conversión bidireccional de niveles de voltaje. Se utiliza para ampliar la distancia de transmisión y mejorar la robustez de la comunicación en aplicaciones industriales. A través de este equipo llegan los mensajes de los sensores al microcontrolador ESP32.

THINGSBOARD: es una plataforma de IoT de código abierto que permite la recopilación, procesamiento, visualización y gestión de datos de dispositivos IoT (Figura 7). Se utiliza para conectar y gestionar dispositivos, recopilar y visualizar datos, procesar y reaccionar a los datos, y controlar dispositivos de manera remota. Es compatible con protocolos estándar como MQTT, CoAP y HTTP, y puede desplegarse tanto en la nube como en instalaciones locales.

Dentro de este marco, un *dispositivo* en ThingsBoard se define como cualquier objeto IoT capaz de generar datos de telemetría, como sensores de temperatura o dispositivos inteligentes. Por su parte, un *activo* representa una entidad física o lógica, como un edificio o una máquina, que puede estar asociada a varios dispositivos, pero no necesariamente genera datos de telemetría por sí misma. Para facilitar la visualización y el monitoreo de estos datos, ThingsBoard incluye *dashboards*, paneles de control altamente

personalizables que permiten analizar la información en tiempo real, apoyando la toma de decisiones y la gestión eficiente de los recursos.

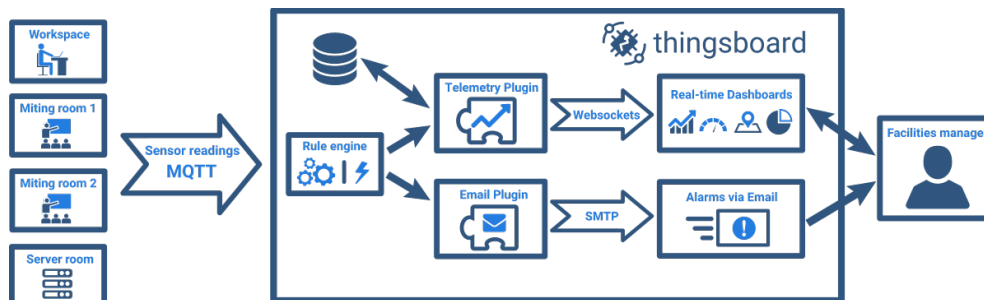


Figura 7. Diagrama de flujo de datos de ThingsBoard.

IDE Arduino: (Integrated Development Environment) es una herramienta de desarrollo que permite escribir, editar, depurar y grabar código en un entorno sencillo, utilizando un lenguaje basado en C++ (Figura 7). Es compatible con ESP32 y cuenta con una comunidad amplia, ofreciendo numerosos recursos y ejemplos para IoT, lo que motivó su selección.

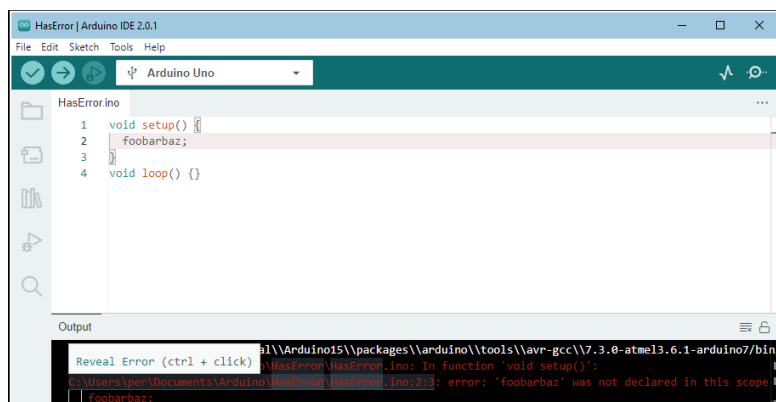


Figura 8. Interfaz IDE Arduino.

4.3.3. Diagrama de flujo y esquemático del circuito electrónico.

Para realizar el esquemático del circuito se prestó atención a las posiciones de los componentes, buscando un diseño ordenado y lógico que facilite la comprensión y el mantenimiento del circuito electrónico. Se aseguró que todos los elementos fueran compatibles entre sí, tanto en términos de voltaje como de corriente, para evitar problemas de funcionamiento. Se seleccionó una fuente de poder que pudiera alimentar adecuadamente todos los componentes del circuito, garantizando un suministro estable y suficiente. Por último, se seleccionó cables de conexión que soporten las interferencias generadas por bombas y variadores de frecuencia, de tal manera que el ruido eléctrico no interfiera en la interfaz RS485.

Para la creación del diagrama de flujo, se identificó los procesos clave del programa, incluyendo la inicialización de comunicaciones, la conexión a redes y plataformas, y la consulta y procesamiento de datos. La lógica que se diseñó consulta constantemente los valores de presión, como condicional para saber si las bombas están prendidas o apagadas. Si una bomba o las dos están prendidas, comienza con el cálculo del caudal, para luego, estimar y enviar a Thingsboard las variables y parámetros de rendimiento.

Tanto el esquema del circuito electrónico, como el diagrama de flujo del programa se encuentra en APENDICE C1, mientras que el código fuente del programa está en APENDICE C3.

4.3.4. Configuración Plataforma Thingsboard.

Se configuró la plataforma Thingsboard para recibir y visualizar los datos enviados por el sistema. Esto incluyó la creación de un *dispositivo* denominado “MONITOREO ENERGÉTICO” y su correspondiente credencial de acceso. También se creó un *activo* denominado “Banco de pruebas”, al cual se asocia este *dispositivo*.

Por último, se configuró un *dashboard*, en el cual se puede monitorear en tiempo real las variables críticas medidas como; presión, caudal, potencia, corriente, voltaje, frecuencia, en conjunto con los indicadores de rendimiento.

La descripción detallada de Thingsboard se puede encontrar en APENDICE C2, donde se explica cómo se configuró todos los *widgets* utilizados en el panel de visualización (*dashboard*).

5. RESULTADOS.

5.1. Construcción banco de pruebas de bombeo.

En la Figura 9 se muestra un esquema del sistema. En él, se dispuso las bombas en paralelo con un ensamble de la succión directa a la fuente de agua, posee además una conexión eléctrica con los variadores de frecuencia para controlar el flujo. Cada bomba, en la descarga, tiene instalado un manómetro de glicerina en conjunto con un transductor de presión, esto con objeto de monitorear las variables mencionadas. También cada bomba posee un caudalímetro para llevar el conteo del flujo, confluyendo estos caudales en una línea única para dirigirse hacia el sistema de filtrado, el cual posee otro manómetro de glicerina dedicado a observar la pérdida de presión del sistema. Una llave de paso regula el punto de operación deseado y el flujo circulante regresa nuevamente al estanque.

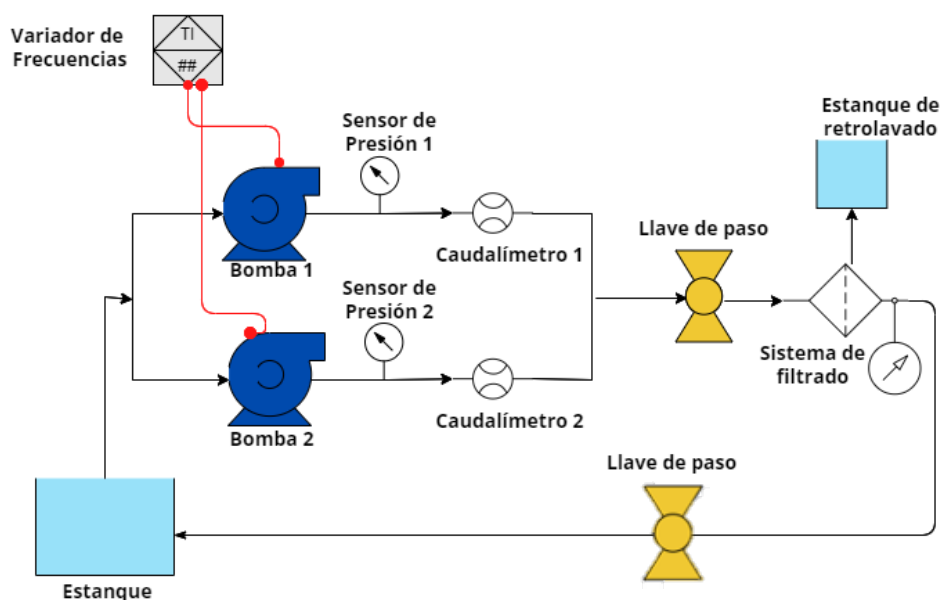


Figura 9. Esquema del banco de pruebas propuesto.

La Figura 10 muestra el diseño del sistema de bombeo, el cual se creó estratégicamente para optimizar el rendimiento y el uso del espacio. Las bombas están ubicadas justo frente al estanque para evitar una altura dinámica en la succión, asegurando un flujo constante y eficiente. La conexión eléctrica se instaló sobre las bombas debido a la cercanía, facilitando el cableado y el acceso. Los caudalímetros fueron posicionados en la parte superior de las tuberías, no solo para aprovechar el espacio disponible y permitir la instalación de una base firme, sino también porque requieren un tramo recto específico para medir con precisión. Finalmente, el filtro se encuentra al final del sistema, principalmente por razones de espacio, garantizando que la disposición general sea compacta y funcional.

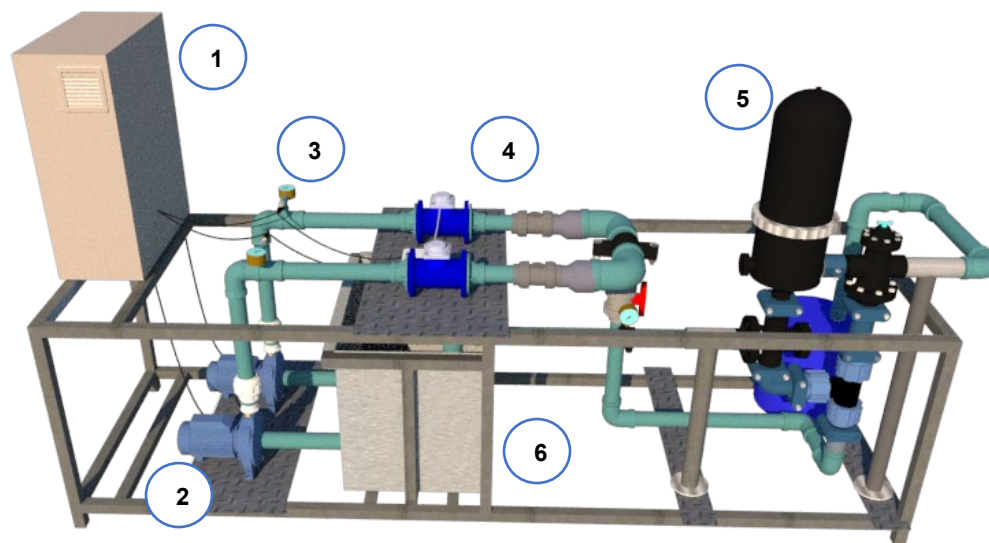


Figura 10. Representación tridimensional del banco de pruebas. Donde (1) Tablero eléctrico, (2) Bombas, (3) manómetros y transductores de presión, (4) Caudalímetros, (5) Sistema de filtrado y (6) Estanque de agua.

La instalación se llevó a cabo con un enfoque en la eficiencia operativa y la facilidad de mantenimiento. Como resultado de una instalación adecuada, se obtuvo el banco de pruebas mostrado en la Figura 11. Para mitigar las vibraciones generadas por las bombas, se incorporaron gomas anti vibratorias, prolongando la vida útil de los componentes. Se instalaron válvulas antirretorno junto a un *manifold* para evitar el cruce de caudales y asegurar que el flujo se mantenga en la dirección correcta. Además, se utilizaron uniones americanas en puntos estratégicos, permitiendo un desmontaje rápido en caso de fallas o ajustes, sin afectar el resto del sistema. El diseño incluye soportes móviles para facilitar el transporte, manómetros para monitorear las presiones en zonas clave, y una disposición compacta de las tuberías que optimiza el uso del espacio.



Figura 11. Instalación completa del banco de pruebas.

5.2. Implementación del sistema de medición.

El sistema de monitoreo, ilustrado en la Figura 12, funciona mediante emisores de pulsos y sensores de presión que envían la información recogida a sensores inteligentes. En este caso, los VDF traducen el valor de presión actual, mientras que los contadores de pulsos registran la cantidad acumulada. El microcontrolador ESP32 consulta estos sensores, a través de MODBUS-RTU, para calcular indicadores de rendimiento, los cuales son transmitidos a ThingsBoard mediante Wifi utilizando el protocolo MQTT. En la plataforma, las variables e indicadores se visualizan en tiempo real a través de *dashboards* interactivos, y se configuran alarmas que notifican bajo rendimiento o falla detectada.

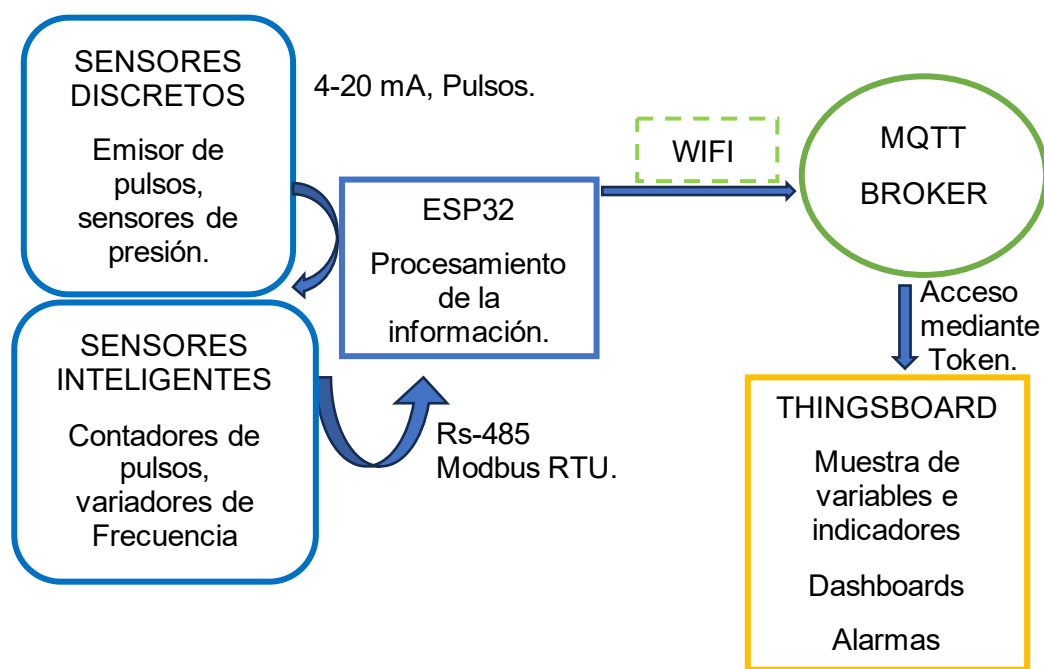


Figura 12. Arquitectura del sistema.

Se realizaron pruebas de envío y recepción de datos inalámbricos entre el microcontrolador ESP32 y la plataforma Thingsboard, siendo estas exitosas. Observando en el buzón de última telemetría (véase en Figura 13) la llegada de cada dato enviado.



The screenshot shows the 'Monitoreo Energético' interface with the 'Última telemetría' tab selected. The table below shows the received data points:

<input type="checkbox"/>	Hora de última actualizaciClave ↑	Valor	
<input type="checkbox"/>	2024-10-14 17:42:40	caudal2	147.96
<input type="checkbox"/>	2024-10-14 19:00:58	corriente1	3.6
<input type="checkbox"/>	2024-10-14 17:42:40	corriente2	3.2

Figura 13. Recepción de datos en Thingsboard.

El panel de visualización obtenido en este proyecto (Figura 14) muestra variables hidráulicas como eléctricas que son importantes en ámbitos energéticos, como también los indicadores de rendimientos definidos en los criterios de evaluación, todo esto en tiempo real. También presenta *widget* de control que informan si las bombas están encendidas, en conjunto con alarmas generadas.

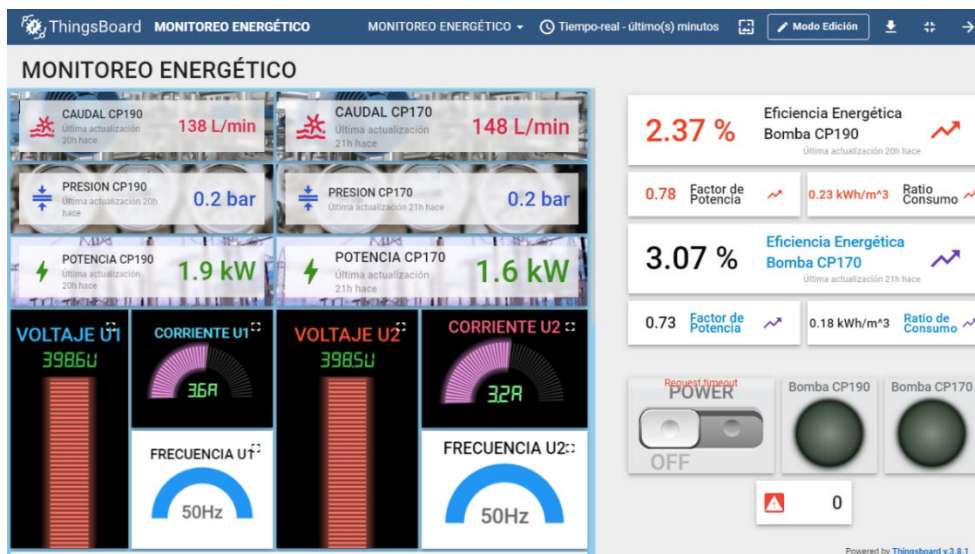


Figura 14. Panel de visualización creado para la monitorización.

Dentro de esta plataforma se pueden visualizar distintos gráficos de variables o parámetros de rendimiento en función del tiempo, pudiendo agrupar estos datos en el tiempo que se desee observar. Un ejemplo de esto es la Figura 15 que muestra el comportamiento de la presión y caudal en un tiempo dado.

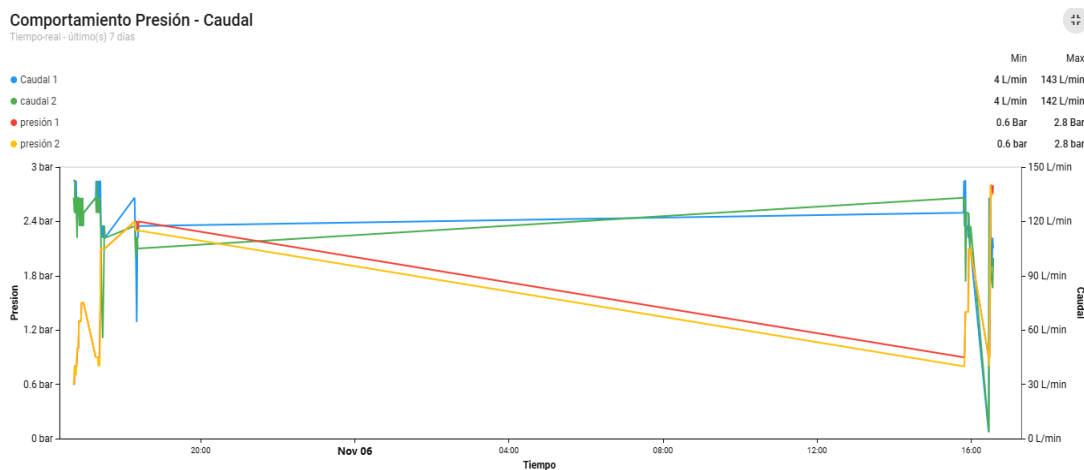


Figura 15. Monitoreo de la presión y caudal en la plataforma web.

La Figura 16 muestra la llegada de las notificaciones de alarma que llegan al celular mediante la aplicación móvil de ThingsBoard, activándose según los criterios definidos en su configuración (descrita en APENDICE C2).

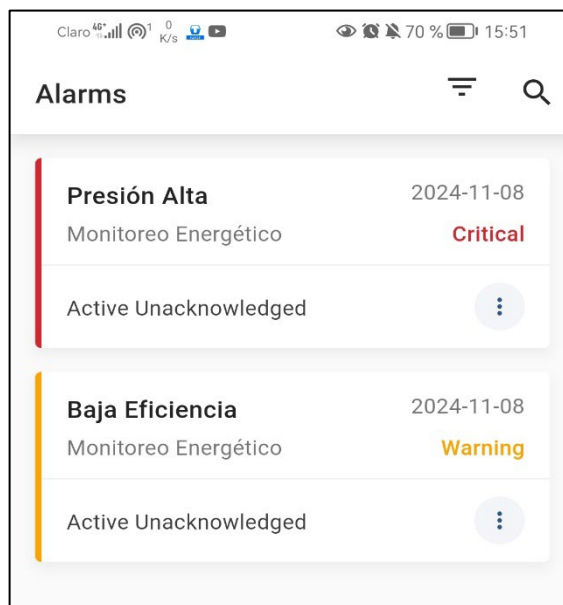


Figura 16. Notificación de alarma en la plataforma móvil.

El sistema de monitoreo se probó en los escenarios propuestos, donde se generó datos a distintos puntos operativos de caudal – presión para su posterior análisis comparativo entre eventos. Los datos obtenidos fueron procesados en Excel para crear gráficos de dispersión (Figuras 17 y 18) y comparar indicadores de rendimiento en función del caudal.

El Escenario 1 reflejó el comportamiento de las bombas operando a una frecuencia nominal de 50 Hz, según la recomendación del fabricante. La curva presión-caudal evidenció diferencias en la capacidad de aporte de caudal entre ambas bombas en distintos puntos de presión operativa, diferencias que se acentuaron a partir de 2 bar, como se observa en la Figura 17(a). En este

rango, la bomba CP190 aportó mayor flujo en la mayoría de los casos en comparación con la bomba CP170.

La Figura 17(b) mostró el valor máximo de PEE alcanzado por cada bomba en distintos puntos de operación. La bomba CP190 registró un PEE de 24,5%, mientras que la CP170 alcanzó 27,3%, lo que indica que, según los criterios establecidos, la CP170 tuvo un rendimiento óptimo, a diferencia de su contraparte. Como resultado, el sistema en conjunto logró un PEE máximo de 25,22%, evidenciando que el funcionamiento en paralelo con bombas no idénticas no fue completamente eficiente.

El ratio de consumo (R_c) presentó un desempeño normal tanto en cada bomba individual como en el sistema en paralelo, como se muestra en la Figura 17(c). Todos los puntos operativos estuvieron por debajo del límite establecido en los criterios de evaluación, y el valor más alto registrado fue $0,34 \text{ kWh m}^{-3}$, lo que indica un desempeño adecuado sin pérdidas de energía significativas que sugieran problemas mayores.

Por otro lado, el factor de potencia (FP) mostró diferencias entre ambas bombas, como se aprecia en la Figura 17(d). Mientras la bomba CP170 presentó valores cercanos a 0,75, la CP190 mantuvo un FP en torno a 0,8. Esta diferencia se debió al manejo de los variadores de frecuencia (VDF), ya que el VDF de la bomba CP190 suministró una corriente de 3,6 A, inferior a la nominal, lo que provocó un aumento en su FP. No obstante, el rendimiento de este parámetro fue óptimo en todos los puntos operativos para ambas bombas.

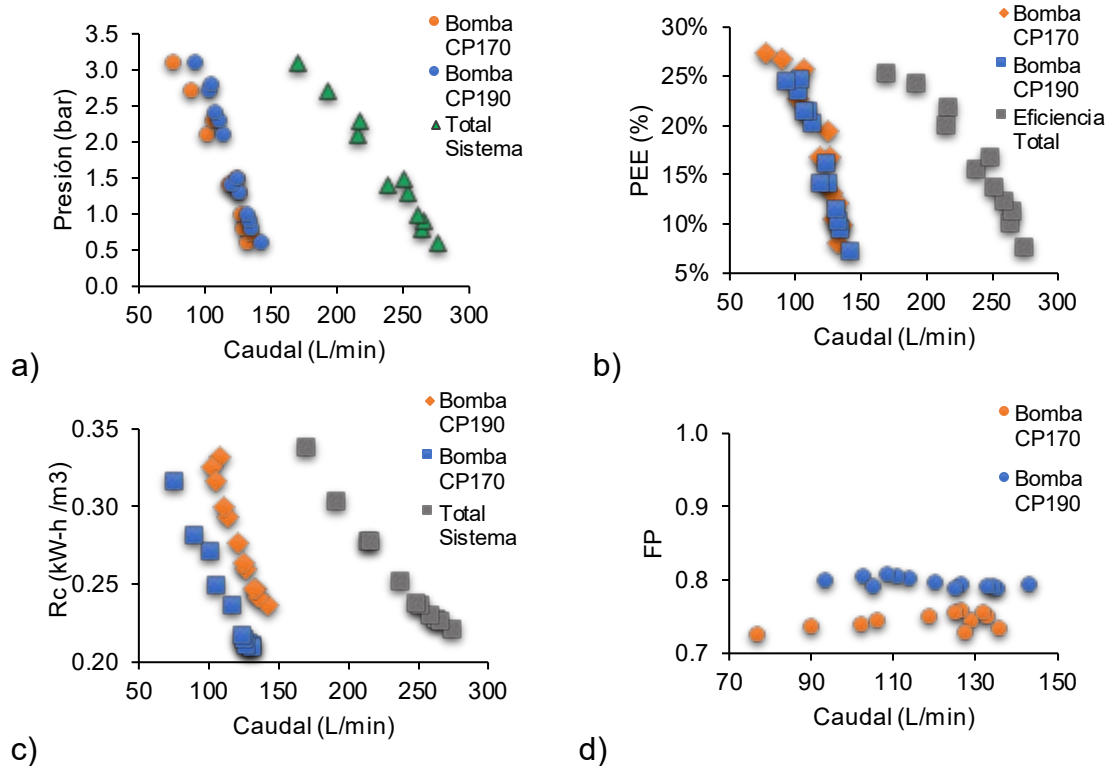


Figura 17. Datos obtenidos en la medición de los indicadores de rendimiento durante el Escenario 1. Donde (a) presenta la curva característica medida, (b) muestra la Eficiencia energética de bombeo (PEE), (c) presenta el Ratio de consumo (Rc), y (d) muestra el Factor de potencia (FP), todos ellos en función del caudal.

En el Escenario 2, se observó un comportamiento similar en la curva presión-caudal para las bombas utilizadas, como se muestra en la Figura 18 (a1). En algunos sectores se produjo una superposición de los puntos pertenecientes a dichas curvas. Por lo tanto, el ajuste realizado a la bomba CP190 tuvo el efecto esperado en el sistema de bombeo, recreando el funcionamiento típico de sistemas de bombeo en paralelo.

El PEE obtenido tuvo una fuerte correlación entre ambas bombas presentando cifras similares por punto operativo, esto se puede observar en la Figura 18 (b1). Los valores máximos obtenidos por las bombas se presentaron a la misma presión operativa (2,6 bar). Estos fueron de 26,2% para la bomba CP190 y de 27,3% para la bomba CP170, teniendo ambas bombas un rendimiento óptimo. En consecuencia, el sistema también tuvo un rendimiento óptimo de un 26,9%.

El Rc se mantuvo con un funcionamiento normal en esta segunda prueba, por lo que todos los puntos medidos se mantuvieron bajo el límite establecido. No obstante, el valor máximo obtenido por parte del sistema disminuyó a un 0,32 kW-h m⁻³, como se muestra en la Figura 18 (c1). Evidenciando que el ajuste a la bomba CP190 redujo el consumo energético del sistema.

El FP en esta evaluación tuvo un comportamiento particular, puesto que los valores medidos por bomba tendieron a igualarse, bajando la cifra obtenida en el primer Escenario por parte de la bomba CP190. Este fenómeno ocurrió por la baja de la frecuencia de trabajo de esta bomba, en donde el VDF disminuyó bastante el voltaje de trabajo rondando 365 VAC. De esta manera, ambas bombas percibieron números cercando al 0,75 respectivamente, tal como se observa en la Figura 18 (d1). En cuanto a su rendimiento, ambas bombas se comportaron de forma normal, con un punto operativo mínimo de 0,73.

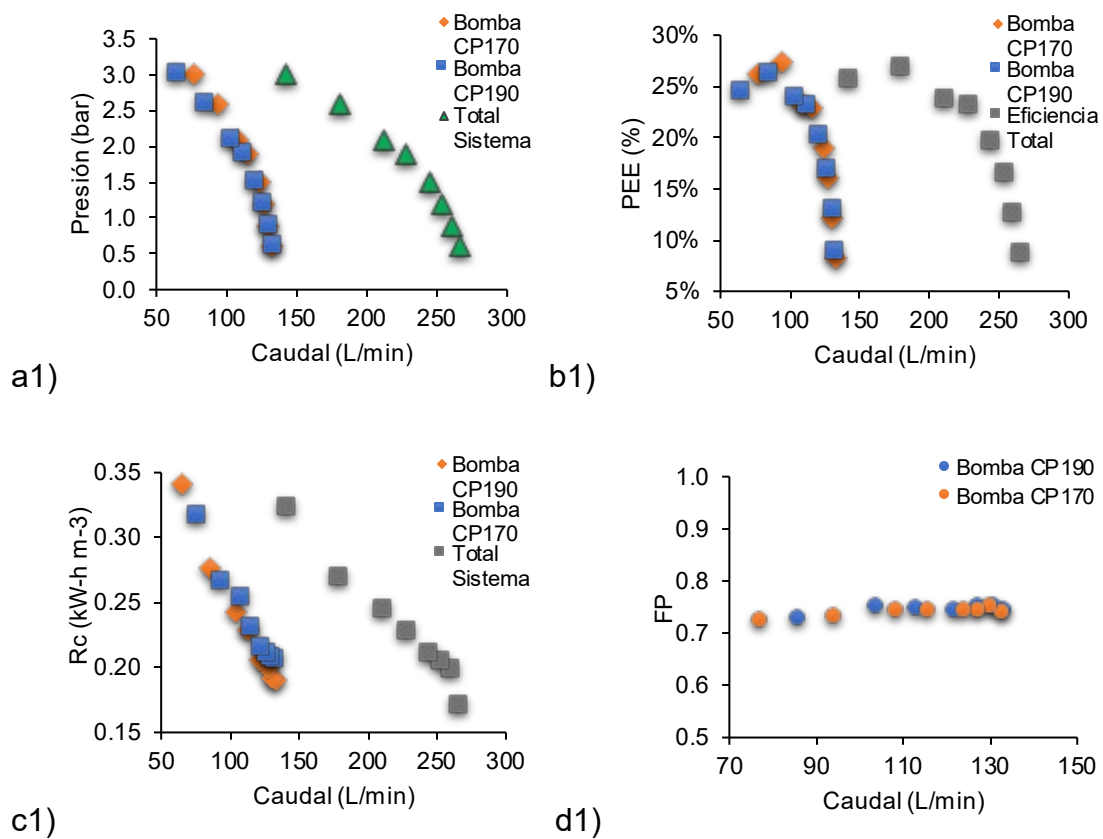


Figura 18. Datos obtenidos en la medición de los indicadores de rendimiento durante el Escenario 2. Donde (a1) presenta la curva característica medida, (b1) muestra la Eficiencia energética de bombeo (PEE), (c1) presenta el Ratio de consumo (Rc), y (d1) muestra el Factor de potencia (FP), todos ellos en función del caudal.

6. DISCUSIÓN.

6.1. Comportamiento del sistema de bombeo.

La construcción del banco de pruebas tuvo algunas dificultades relacionadas con la compatibilidad entre los *fittings* utilizados, estos fueron de la misma medida utilizada en el armado (63 mm), pero al provenir de diferentes marcas se debió realizar algunas intervenciones (cambio de *fittings* y un mayor lijado en las uniones) para que coincidieran entre ellos. Al realizar pruebas de bombeo a diferentes presiones se observó una pérdida de carga mínima en el manómetro ubicado en el sistema de filtrado, registrando una diferencia de presión (perdida de carga) en promedio de 0,3 bar.

El funcionamiento tanto de las bombas individuales como en conjunto no alcanzó los valores entregados en la curva de prestaciones, como se aprecia en la Figura 19 (curvas de prestaciones comparadas con las curvas obtenidas en el Escenario 1). Esto, se reflejó al comparar los puntos más eficientes teóricos con los obtenidos tras la prueba (Escenario 1). En el caso de la bomba CP190, el punto obtenido (105 L min^{-1} , 2,8 bar) se presentó a una presión más baja que la teórica (110 L min^{-1} , 3,3 bar) y con un caudal cercano a la misma.

Por otro lado, la bomba CP170 obtuvo un punto medido (77 L min^{-1} , 3,1 bar) con un valor de presión mayor que la teoría (100 L min^{-1} , 2,7 bar) y en contraste un caudal menor que la mencionada.

Se alcanzó una presión máxima de 3,1 bar (31 metros), estando por debajo del valor establecido en el diseño (4 bar). Los caudales presentados se

mantuvieron dentro del rango establecido. Con estas diferencias respecto a las curvas de prestaciones teóricas del banco de bombeo, de igual manera el sistema construido proporcionó curvas de presión – caudal a distintos puntos operativos, que permitieron la implementación del sistema de monitoreo energético y su respectiva evaluación.

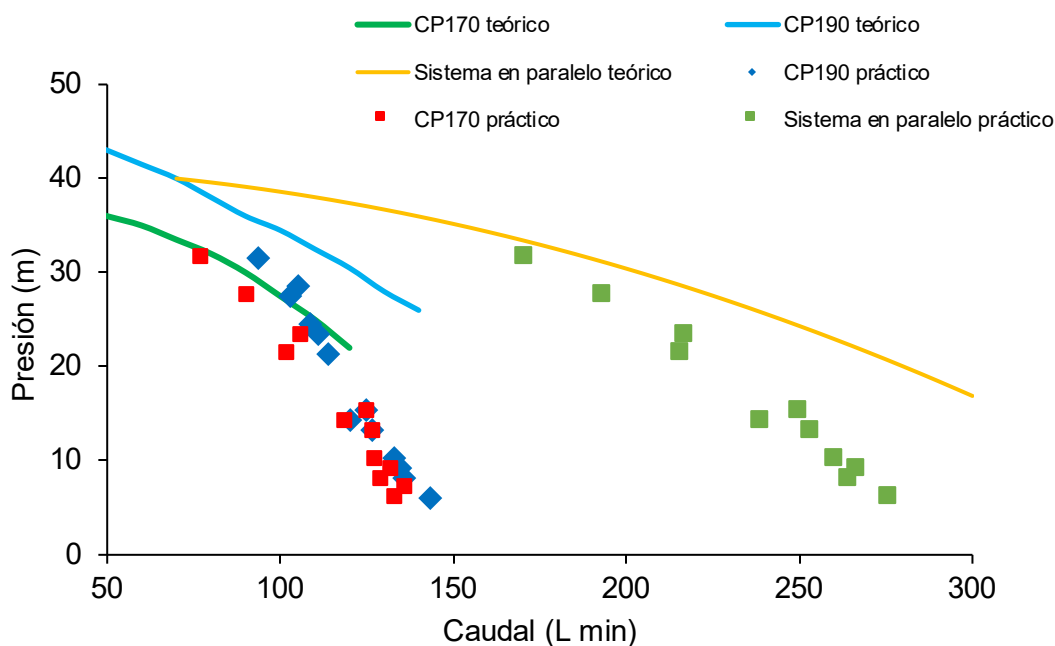


Figura 19. Comparativa entre las curvas presión - caudal teóricas de las bombas y el sistema en paralelo con las curvas obtenidas en el Escenario 1.

6.2. Comportamiento del sistema de monitoreo.

El comportamiento del programa creado dependió directamente de la estimación del caudal, puesto que este valor no se obtuvo directamente al consultar los sensores utilizados, más bien se debió realizar un cálculo extra en el programa dependiente del tiempo en el que cambia el valor de los pulsos registrados en los contadores de pulsos. Esta condición limitó el envío de datos

a Thingsboard de manera constante y expedita, solo enviando la información en base a la disponibilidad del valor de caudal. La utilización de sensores de tipo ultrasónicos o digitales que proporcionen el valor de caudal directamente al sistema de monitoreo hubieran permitido un envío de datos inmediato con mediciones precisas y consistentes, reduciendo la carga de procesamiento al microcontrolador.

En cuanto a la conexión del microcontrolador ESP32 mediante Wifi y MQTT no hubo mayor problema, recibiendo la plataforma ThingsBoard todos los datos enviados correctamente. El panel creado para la visualización del monitoreo energético actualizó rápidamente los valores de las variables recibidas por parte del ESP32, mostrando el tiempo de actualización de dicho valor. También las alarmas incorporadas al sistema se activaron de acuerdo con los criterios configurados en Thingsboard (presión mayor a 3 bar, eficiencia baja), generando una notificación tanto en la página web, como en la aplicación móvil, esto último útil para conocer el estado del sistema de bombeo de manera remota tras supuestos problemas o fallas críticas.

A pesar, de cumplir con la función básica de visualización de variables en función del tiempo, Thingsboard edición comunitaria, se mostró limitado en cuanto a sus capacidades analíticas. La falta de soporte para gráficos de dispersión (necesario para presentar los resultados obtenidos) y la imposibilidad de crear nuevas variables a partir de las existentes obligó a dejar la mayoría de la lógica de procesamiento al microcontrolador, restringiendo la

posibilidad de análisis de datos en línea. Esto sugiere que la utilización de una edición profesional de la plataforma con funcionalidades más avanzadas y un mayor nivel de personalización, habría ofrecido las herramientas necesarias para un análisis más profundo.

6.3. Comparativa de los escenarios de prueba.

Antes de realizar la comparativa de los Escenarios se tuvo que suavizar los datos referidos al caudal, dado que el cálculo de esta variable fue altamente sensible al momento exacto de la medición. Esto se explica por el origen del dato obtenido el cual depende del cambio de pulso, en donde al consultar el programa cada cierto tiempo determinado si ha cambiado el pulso, el tiempo entre este cambio se ve afectado por los demás procesos que realiza simultáneamente el microcontrolador. También, pequeñas turbulencias creadas en el sistema de bombeo provocan inestabilidad en el flujo que refleja un cambio en el tiempo entre pulsos, por consiguiente, mediciones de caudales distintos para una misma presión operativa.

Para suavizar los datos de caudal y obtener una tendencia más precisa y representativa de la realidad, se tomaron más de 10 muestras de caudales por cada presión de trabajo. Posteriormente, se promediaron estos valores, lo que permitió reducir la variabilidad y proporcionar una estimación más confiable del caudal. Esto reflejado en la Figura 17 y Figura 18 que muestran el comportamiento de los escenarios realizados en el capítulo RESULTADOS.

La curva presión – caudal respecto al Escenario 2 respondió de manera positiva al ajuste realizado, generando aportes bastante cercanos de caudal por partes de las bombas y una corrección a la curva del sistema. Por lo que el flujo en el punto de unión de los caudales fue estable y uniforme, mejorando la eficiencia y el rendimiento general del sistema.

El Escenario 2 mejoró el PEE obtenido para la bomba CP190 y el sistema respecto de los resultados obtenidos en el Escenario 1, teniendo el punto operativo más eficiente a la misma presión de trabajo en las bombas y el sistema.

El ratio de consumo en ambos escenarios estuvo bajo los límites establecidos, considerando su comportamiento normal. No obstante, el Escenario 2 disminuyó las cifras de cada bomba y el sistema respecto al Escenario 1, lo que significa un ahorro de energía consumida por el sistema por cada metro cúbico bombeado.

El FP tuvo un comportamiento singular, ya que en el Escenario 2 la bomba CP190 disminuyó su valor respecto al Escenario 1, con un valor medio de 0,74. Asemejándose a los números medidos en la bomba CP170 para ambos Escenarios. Aun así, las cifras registradas en dichas pruebas no superaron el límite establecido, por lo que se considera un rendimiento normal.

En términos generales, el Escenario 2 tuvo una mejora sustantiva en la eficiencia energética del sistema de bombeo evaluado, presentando cambios

positivos en los parámetros de rendimiento PEE y Rc. El valor de PEE paso de un 25,2% a un 26,9% aumentando en un 1,7% la eficiencia del sistema, disminuyendo la pérdida generada entre la potencia Activa (P_1) y la potencia hidráulica final que llega al fluido. En el mismo punto P – Q, el Rc tuvo una disminución entre escenarios de $0,7 \text{ kW-h m}^{-3}$ generando un ahorro energético cercano a 1 kW-h por metro cúbico bombeado, disminuyendo el costo de la facturación eléctrica. No obstante, el parámetro FP de la bomba CP190 específicamente disminuyó de acuerdo con lo obtenido en el Escenario 1, aunque esta disminución no perjudicó a los demás parámetros, y de acuerdo con el criterio establecido se mantuvo dentro de un rango aceptable.

7. CONCLUSIONES.

La implementación de un sistema de medición de eficiencia energética en sistemas de bombeo basados en tecnologías IoT fue exitosa. En efecto, se logró construir un banco de pruebas que permitió evaluar distintos escenarios de operación, lo que facilitó la determinación de la eficiencia energética. A pesar de las discrepancias observadas entre las curvas teóricas y experimentales, se obtuvieron datos válidos para el análisis.

Además, se implementó un sistema de medición IoT para la recopilación y visualización de datos en tiempo real. No obstante, se identificó limitaciones en la precisión de la medición del caudal y en las capacidades de análisis de datos de la plataforma en la nube.

A partir de los datos recopilados, se establecieron indicadores de eficiencia energética, específicamente: Eficiencia Energética de Bombeo (PEE), Ratio de Consumo (Rc) y Factor de Potencia (FP). Los resultados mostraron que en el Escenario 2, la PEE aumentó en un 1,7% y el Rc disminuyó en 0,7 kWh m⁻³ en su punto más eficiente, lo que representa un ahorro energético significativo.

Finalmente, la metodología desarrollada proporciona una base sólida para el análisis y la optimización de la eficiencia energética en sistemas de bombeo. La integración de tecnologías IoT en el sistema de medición ha demostrado que, a partir de la monitorización en tiempo real, es posible mejorar la eficiencia operativa y, en consecuencia, reducir el consumo eléctrico.

8. LITERATURA CITADA.

- Abdelaziz, E. A., Saidur, R., & Mekhilef, S. (2011). A review on energy saving strategies in industrial sector. *Renewable and Sustainable Energy Reviews*, 15(1), 150–168. <https://doi.org/10.1016/j.rser.2010.09.003>
- Arun Shankar, V. K., Umashankar, S., Paramasivam, S., & Hanigovszki, N. (2016). A comprehensive review on energy efficiency enhancement initiatives in centrifugal pumping system. *Applied Energy*, 181, 495–513. <https://doi.org/10.1016/j.apenergy.2016.08.070>
- Belaud, G., Mateos, L., Aliod, R., Buisson, M.-C., Faci, E., Gendre, S., Ghinassi, G., Gonzales Perea, R., Lejars, C., Maruejols, F., & Zapata, N. (2020). Irrigation and energy: Issues and challenges. *Irrigation and Drainage*, 69(S1), 177–185. <https://doi.org/10.1002/ird.2343>
- Bolognesi, A., Bragalli, C., Lenzi, C., & Artina, S. (2014). Energy efficiency optimization in water distribution systems. *Procedia Engineering*, 70, 181–190. <https://doi.org/10.1016/j.proeng.2014.02.021>
- Cisternas, I., Velásquez, I., Caro, A., & Rodríguez, A. (2020). Systematic literature review of implementations of precision agriculture. *Computers and Electronics in Agriculture*, 176(105626), 105626. <https://doi.org/10.1016/j.compag.2020.105626>
- Gallego, P. (2020). Sistema de monitorización IoT de datos de una bomba de agua para su mantenimiento predictivo. Trabajo de fin de grado, Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza, Zaragoza. <https://zaguan.unizar.es/record/96574/files/TAZ-TFG-2022106.pdf?version=1>
- Kaya, D., Çanka Kılıç, F., Öztürk, HH (2021). Eficiencia energética en bombas. En: Gestión energética y eficiencia energética en la industria. Energía verde y tecnología. Springer, Cham. https://doi.org/10.1007/978-3-030-25995-2_11
- Hamami, L., & Nassereddine, B. (2020). Application of wireless sensor networks in the field of irrigation: A review. *Computers and Electronics in Agriculture*, 179(105782), 105782. <https://doi.org/10.1016/j.compag.2020.105782>
- International Electrotechnical Commission. (2014). Rotating electrical machines - Part 30-1: Efficiency classes of line operated AC motors (IE code) (IEC 60034-30-1:2014). <https://webstore.iec.ch/publication/136>

- Laghari, A. A., Wu, K., Laghari, R. A., Ali, M., & Khan, A. A. (2023). Retraction note: A review and state of art of internet of things (IoT). *Archives of Computational Methods in Engineering. State of the Art Reviews*, 30(8), 5105–5105. <https://doi.org/10.1007/s11831-023-09985-y>
- López-Morales, J. A., Martínez, J. A., & Skarmeta, AF (2021). Mejorar la eficiencia energética de los pozos de riego mediante el uso de una plataforma basada en IoT. *Electrónica*, 10(3), 250. <https://doi.org/10.3390/electronics10030250>
- Luna, T., Ribau, J., Figueiredo, D., & Alves, R. (2019). Improving energy efficiency in water supply systems with pump scheduling optimization. *Journal of Cleaner Production*, 213, 342–356. <https://doi.org/10.1016/j.jclepro.2018.12.190>
- Marzougui, T., Elghali, B., Doumenc, S. y Outlib, F. (2015). Un análisis del coste energético de una red de riego en Francia. Conferencia de la ICID, *Sesión Especial 'Riego y Energía'*.
- Moreno, M. A., Ortega, J. F., Córcoles, J. I., Martínez, A., & Tarjuelo, J. M. (2010). Análisis energético de sistemas de distribución de riego: seguimiento y evaluación de medidas propuestas para mejorar la eficiencia energética. *Ciencias del riego*, 28(5), 445–460. <https://doi.org/10.1007/s00271-010-0206-8>
- Nogueira Vilanova, M. R., & Perrella Balestieri, J. A. (2014). Energy and hydraulic efficiency in conventional water supply systems. *Renewable and Sustainable Energy Reviews*, 30, 701–714. <https://doi.org/10.1016/j.rser.2013.11.024>
- Pedrollo. (s.f.). Ficha técnica de bombas centrífugas CP. Recuperado de https://www.pedrollo.com/wp-content/uploads/schede-tecniche/ES/CP_ES-ficha-tecnica_50Hz.pdf1.
- Pérez-Padillo, J., García Morillo, J., Ramirez-Faz, J., Torres Roldán, M., & Montesinos, P. (2020). Diseño e implementación de un sistema de monitorización de presión basado en IoT para redes de abastecimiento de agua. *Sensors*, 20 (15), 4247. <https://doi.org/10.3390/s20154247>
- Qin, J., Duan, W., Zou, S., Chen, Y., Huang, W., & Rosa, L. (2024). Global energy use and carbon emissions from irrigated agriculture. *Nature Communications*, 15(1), 1–12. <https://doi.org/10.1038/s41467-024-47383-5>

- Rejeb, A., Rejeb, K., Abdollahi, A., Al-Turjman, F., & Treiblmaier, H. (2022). The Interplay between the Internet of Things and agriculture: A bibliometric analysis and research agenda. *Internet of Things*, 19(100580), 100580. <https://doi.org/10.1016/j.iot.2022.100580>
- Ruiz Carmona, V. (2014). Adaptación y transferencia de tecnología para medición en pozos de bombeo en zonas agrícolas. Instituto Mexicano de Tecnología del Agua, Secretaría de Medio Ambiente y Recursos naturales. <http://repositorio.imta.mx/bitstream/handle/20.500.12013/865/RD1308.1.pdf?sequence=1>

9. APENDICES Y ANEXOS.

9.1.1. APENDICE A1: Componentes del banco de pruebas, accesorios hidráulicos y componentes del sistema de medición.

Tabla 4. Componentes del banco de pruebas.


Componente	Cantidad	Imagen
Bombas Pedrollo línea CP	2	
Variadores de frecuencia IQ Pump Micro	2	
Caudalímetros Tecnidro	2	
Filtro Azud Helix Automatic 201DLP	1	
Manómetros de glicerina	3	
Estanque de agua	1	

Tabla 5. Accesorios utilizados en la construcción del banco.





















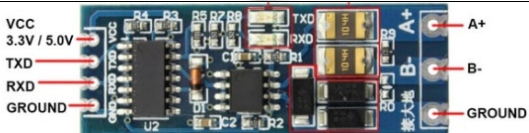


Succión		
Accesorios	Cantidad	Imagen
Unión americana 63 mm PN10	2	
Terminal 63 mm	2	
Reducción 2"x1 1/4"	2	
Longitud Tubería 63 mm	0,44 m	
Descarga		
Unión americana 63 mm PN10	4	
expansión 1" x 2"	2	
codos 90° 63 mm	4	
Tee 63 mm flujo línea recta	2	
bridas de 63 mm	4	
caudalímetro	2	
Válvula antirretorno	2	
Terminal 63 mm	6	
Expansión 63mm x 110 mm	2	
Codo 90° 110 mm	2	
Válvula de compuerta 63 mm	1	
Longitud Tubería 63 mm	4,5 m	
Longitud tubería 110 mm	0,245 m	

Tabla 6. Componentes del sistema de monitoreo.

Componentes	Cantidad	Imagen
Transmisor de presión	2	
Contadores de pulsos	2	
Fuente de poder	1	
Convertor serial TTL a RS-485	1	
Convertidor de voltaje	1	
Microcontrolador ESP32	1	

9.1.2. APENDICE A2: Vistas del diseño y construcción del banco de pruebas.

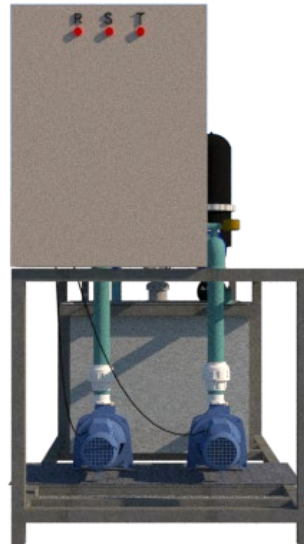


Figura 20. Vista frontal del diseño.



Figura 21. Vista trasera del diseño.

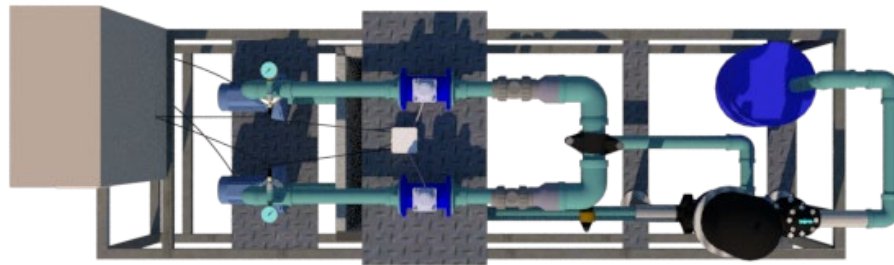


Figura 22. Vista Superior del diseño.

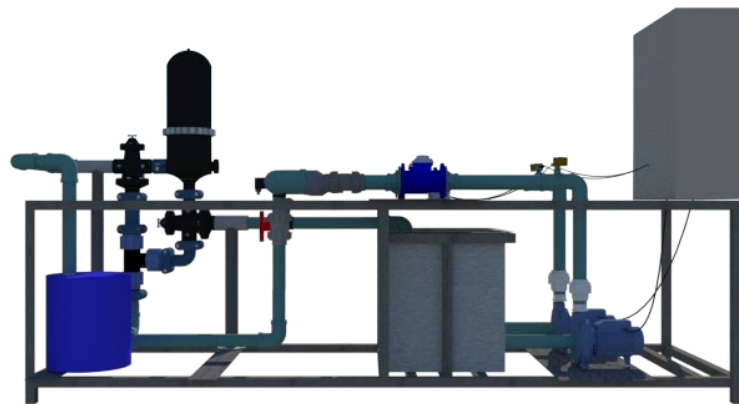


Figura 23. Vista lateral del diseño.

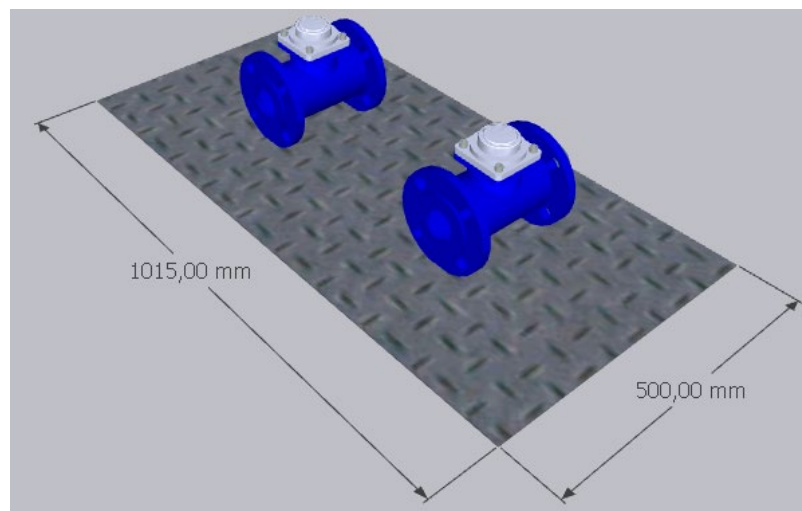


Figura 24. Dimensiones de la base de los caudalímetros.

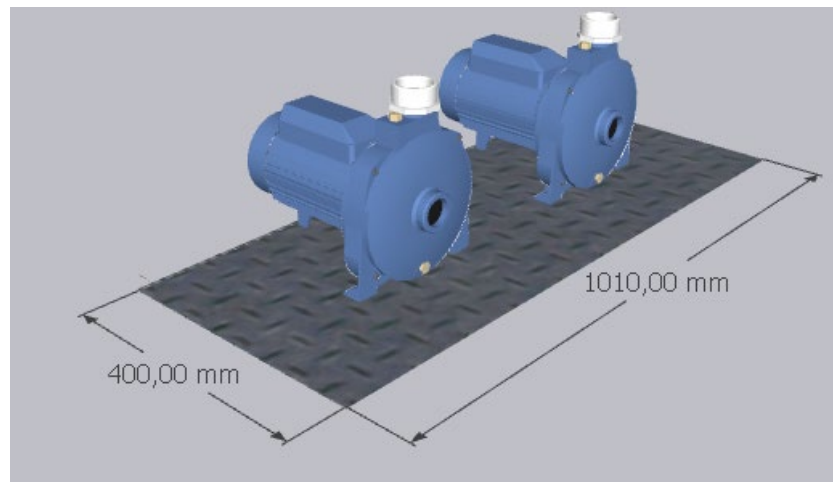


Figura 25. Dimensiones de la base de las bombas.

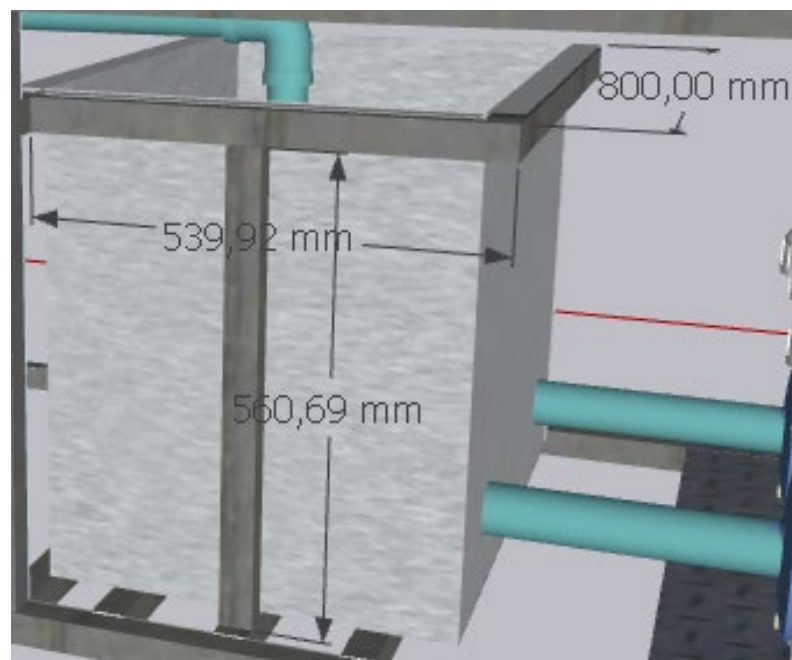


Figura 26. Dimensiones de la estructura de soporte del estanque.



Figura 27. Estructuras de soporte instaladas en el banco de pruebas.



Figura 28. Vista lateral del sistema de bombeo construido.



Figura 29. Vista trasera y frontal del sistema de bombeo construido.

9.1.3. APENDICE A3: Diagrama de cableado y conexión eléctrica.

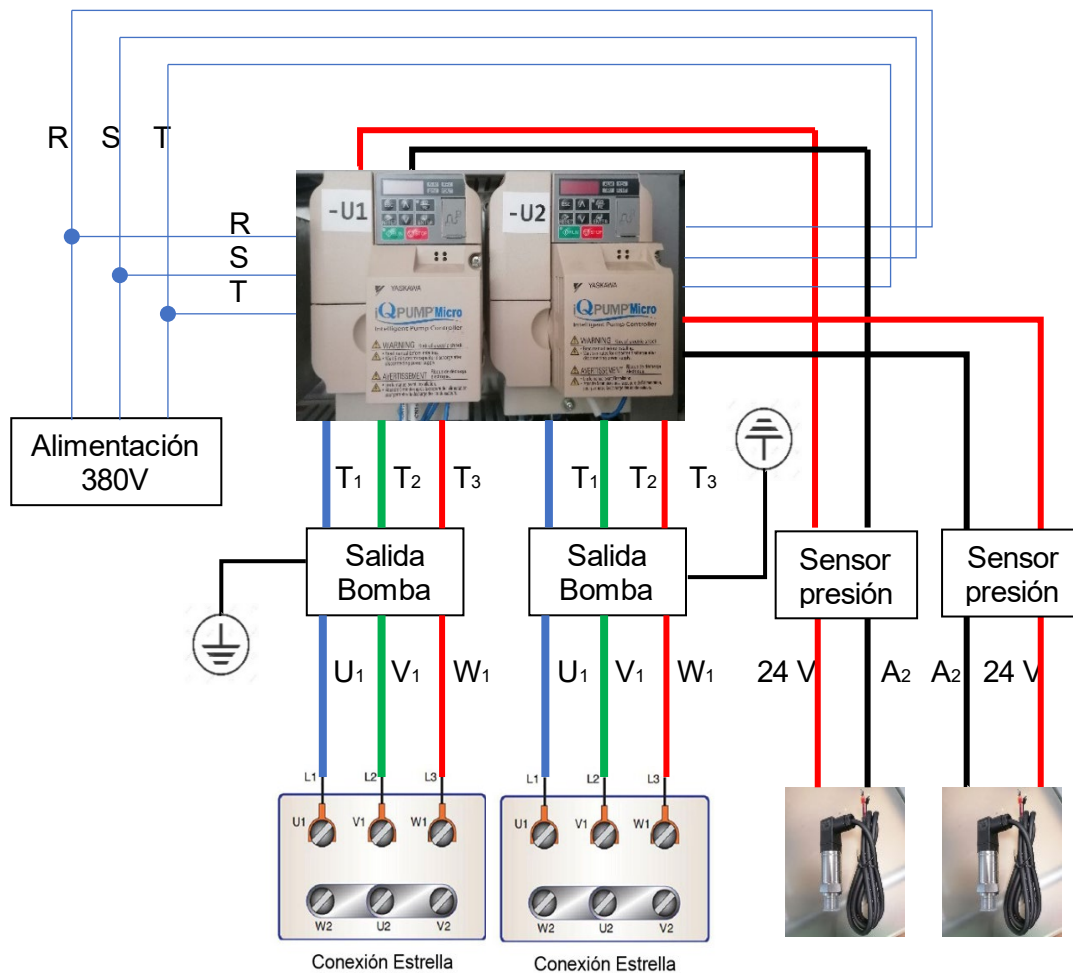


Figura 30. Conexión eléctrica del sistema de bombeo.

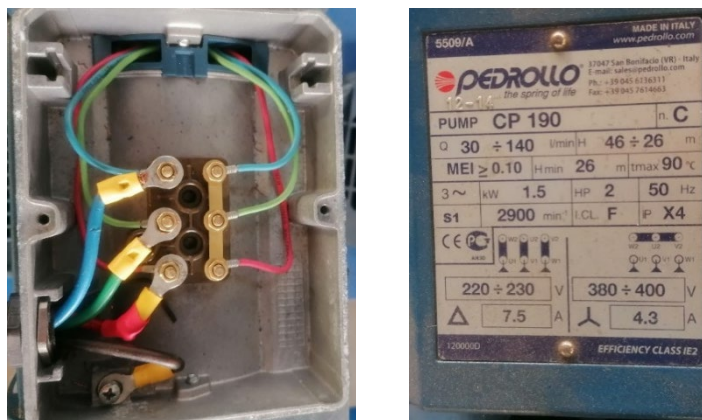


Figura 31. Conexión estrella de bomba CP190 y especificaciones técnicas.

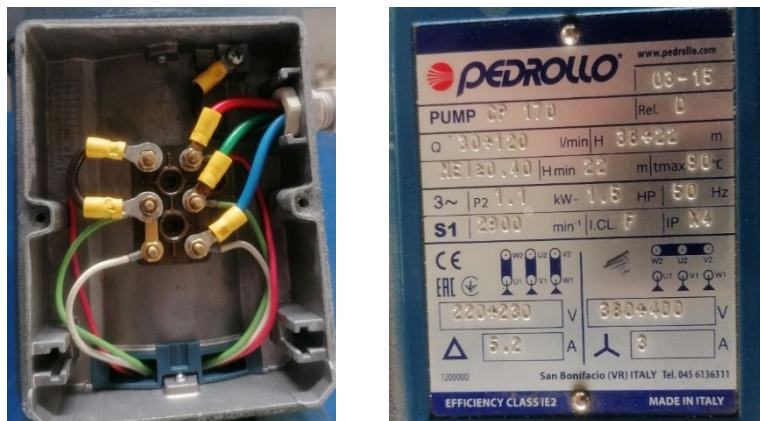


Figura 32. Conexión estrella de bomba CP170 y especificaciones técnicas.

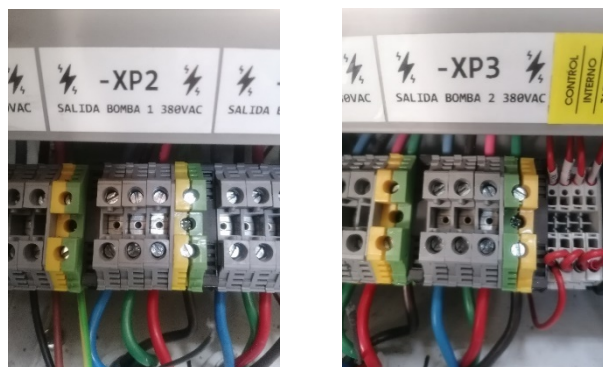


Figura 33. Conexión eléctrica de las bombas a la caseta de bombeo.

9.1.4. APENDICE A4: Construcción del cable de conexión pc – variador UWR00468-2 y configuración de VDF.

Para la configuración de los variadores Yaskawa se construyó un cable de datos que fuera compatible con estos, siguiendo las indicaciones de armado entregadas por la marca en su documento técnico de conexiones (Figura 34). Las especificaciones indicaron la utilización de un conector DB9 hembra en conjunto con un conector RJ45 macho, uniendo pines específicos entre estos conectores. Luego se utilizó un multi- convertor a USB para la conexión con la computadora conectando la entrada DB9 hembra al convertor USB.

WD.DW.05
10/24/2023

**For all 1000-series drives
J7, V7, E7, P7, G7, and F7 Drives**
PC to Drive Connection Cable UWR00468-2

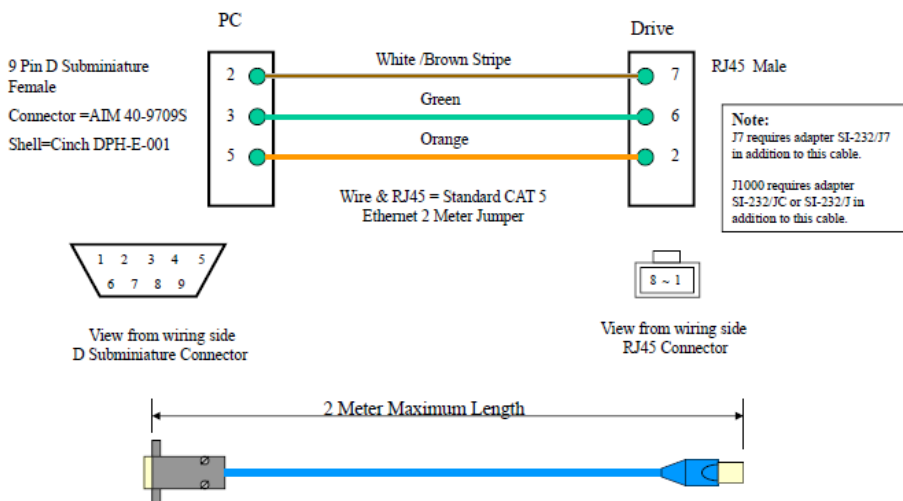


Figura 34. Diagrama esquemático del cable de conexión UWR00468-2.

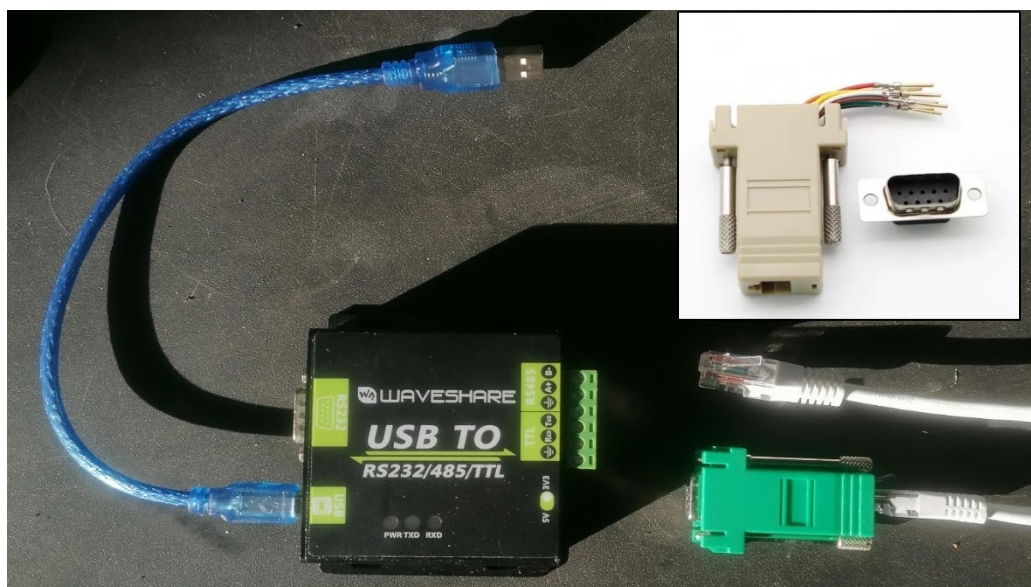


Figura 35. Cable de conexión PC – Variador UWR00468-2 construido.

El adaptador modular DB9 - RJ45 tiene la configuración descrita en la esquemático del cable UWR00648-2, estando conectados solo los pines 2,3 y 5 de la salida DB9 con los pines 7, 6 y 2 de la entrada RJ45 respectivamente.

Luego se probó el prototipo (Figura 35) utilizando el software Drive Wizard IQPump mediante el tipo de conexión serial/ usb teniendo una conexión exitosa entre los variadores y la computadora permitiendo emplear las configuraciones requeridas en el sistema. Estas configuraciones fueron realizadas de acuerdo con los criterios establecidos en la metodología, en donde a cada variador se le modificaron algunos parámetros de acuerdo con las características de cada bomba empleada, como se puede ver en la Figura 36. Los parámetros correspondientes a corriente y potencia nominales son distintos para cada VDF, ya que las bombas son diferentes. Los demás parámetros como frecuencia nominal, voltaje de entrada, máximo voltaje, escala de presión, sistema de unidades de presión o control PID son idénticos en ambos equipos.

Parameter Groups iQpump Micro				
No.	Parameter	Working Value	Info (Working Value)	Default
d1-17	Jog frequency reference	50,00 Hz		6,00 Hz
E1-01	Input voltage setting	380 VAC		460 VAC
E1-05	Max. voltage	415,0 VAC		460,0 VAC
E2-01	Motor rated current	4,30 A		3,10 A
H1-02	Terminal S2 function selection	12	Forward Jog command	0F
H1-03	Terminal S3 function selection	0F	Not used(Unused / when use it as a pass-t	26
H5-01	Station address	03		1F
L2-05	Undervoltage detection level	350 VDC		0 VDC
P1-02	System Units	3	Bar (Bar)	1
P1-03	Feedback Scaling	6,0 Bar		145,0 Bar
P1-11	High Feedback Level	6,0 Bar		155,0 Bar
P5-02	Hand Reference	50,0 Hz		40,0 Hz
P5-02	Hand Reference	50,0 Hz		40,0 Hz
P9-06	Lag Fixed Speed	50,0 Hz		55,0 Hz
P9-25	Highest Node Address	2		4
a) Q1-01	PID Controller Setpoint 1	2,0 Bar		0,0 Bar

Parameter Groups iQpump Micro				
No.	Parameter	Working Value	Info (Working Value)	Default
d1-17	Jog frequency reference	50,00 Hz		6,00 Hz
E1-01	Input voltage setting	400 VAC		460 VAC
E1-05	Max. voltage	415,0 VAC		460,0 VAC
E2-01	Motor rated current	3,00 A		3,10 A
E2-11	Motor rated output	1,10 kW		1,50 kW
H1-02	Terminal S2 function selection	12	Forward Jog command	0F
H1-03	Terminal S3 function selection	0F	Not used(Unused / when use it as a pass-t	26
H5-01	Station address	04		1F
L3-24	Inertia conversion motor acceleration time	0,142 sec		0,166 sec
P1-02	System Units	3	Bar (Bar)	1
P1-03	Feedback Scaling	6,0 Bar		145,0 Bar
P1-11	High Feedback Level	6,0 Bar		155,0 Bar
P5-02	Hand Reference	50,0 Hz		40,0 Hz
P9-06	Lag Fixed Speed	50,0 Hz		55,0 Hz
P9-25	Highest Node Address	2		4
Q1-01	PID Controller Setpoint 1	2,0 Bar		0,0 Bar

b)

Figura 36. Configuración realizada en los VDF donde (a) corresponde al variador de frecuencia 1 y (b) al variador de frecuencia 2.

9.1.5. APENDICE B1: Cálculo de indicadores de rendimiento.

Como la ficha técnica de las bombas no entregó la información correspondiente a la potencia activa y el factor de potencia, se realizó una estimación de estos valores conforme indica el fabricante. Este indicó que las bombas se rigen bajo la norma IEC 60034-30-1 (International Electrotechnical Commission, 2014), que establece las clases de eficiencia para motores de corriente alterna de baja tensión para el año de fabricación de estos equipos. De esta manera se pudo obtener la eficiencia del motor acorde a su potencia P_2 (Potencia mecánica de salida) y su clase IE₂, como muestra la Tabla 7.

Tabla 7. Eficiencia estándar según la norma IEC 60034-30-1 para la clase IE₂.

P_2 (kW)	Polos	Frecuencia (Hz)	Eficiencia del motor (%)
1,1	2	50	75
1,5	2	50	77,2

A través de la curva de prestación de la línea CP se pudo conocer la potencia hidráulica en el punto de máxima eficiencia por bomba (véase Tabla 8).

Tabla 8. Datos de rendimiento proporcionados por el fabricante.

Bomba	Eficiencia Bomba (%)	P ₂ (kW)	Potencia hidráulica (kW)
CP190	33	1,5	0,495
CP170	34	1,1	0,374

De esta forma, se pudo despejar la potencia activa (P₁) con la ecuación 4 respectivamente:

$$\text{Ef. del motor} = \frac{P_2 \text{ (kW)}}{P_1 \text{ (kW)}} \quad [4]$$

Obteniendo el valor de potencia activa teórico por cada bomba, y aplicando la ecuación de PEE, se estableció el valor mínimo para la evaluación del sistema de bombeo, definido en la Tabla 9.

Tabla 9. Obtención de P₁ y el criterio PEE.

Bomba	P ₁ (kW)	PEE calculado (%)
CP190	1,943	25,48
CP170	1,467	25,50

En cuanto al factor de potencia (FP), se debió estimar en primera instancia la potencia aparente (S), representada con la ecuación 5:

$$S \text{ (VA)} = \text{Voltaje (V)} \times \text{Corriente (A)} \times \sqrt{3} \quad [5]$$

Donde los valores de voltaje y corriente fueron adquiridos por la ficha técnica de las bombas utilizadas, mostrando en la Tabla 10 las cifras correspondientes.

Tabla 10. Datos de tensión y consumo de corriente de las bombas, con el S obtenido.

Bomba	Voltaje (V)	Corriente (A)	S (kVA)
CP190	400	4,3	2,98
CP170	400	3,0	2,10

Luego, solo quedó aplicar la Ecuación 3 referida al cálculo del FP en base a S y P_1 estimados por cada bomba, presentando los resultados en la Tabla 11. Con esta metodología se pudo obtener el valor mínimo que requiere el indicador FP para considerar un comportamiento óptimo en la evaluación del sistema de bombeo.

Tabla 11. Obtención del criterio FP.

Bomba	P_1 (kW)	S (kVA)	FP
CP190	1,943	2,98	0,7
CP170	1,47	2,10	0,71

Por último, se estimó el criterio referido al ratio de consumo (R_c) utilizando directamente los valores de caudal (Q) entregados en la curva de prestaciones en conjunto con la potencia activa (P_1) definida anteriormente. Donde se promedia el caudal más bajo con el obtenido en el punto más eficiente teórico, para luego aplicar la Ecuación 2 definida en la metodología.

Tabla 12. Obtención del criterio R_c .

Bomba	P_1 (kW)	Q mínimo ($m^3 h^{-1}$)	Q eficiente ($m^3 h^{-1}$)	Q promedio ($m^3 h^{-1}$)	R_c (kWh m^{-3})
CP190	1,943	3	6,6	4,8	0,4
CP170	1,47	2,4	6	4,2	0,35

9.1.6. APENDICE B2: Ajuste de bomba CP190.

El ajuste de la bomba se realizó en base a las relaciones de semejanza H-N siguiendo la ecuación 6:

$$\frac{H}{H_0} = \left(\frac{N}{N_0}\right)^2 \quad [6]$$

Donde:

H_0 = Presión inicial (bar).

H = Presión final (bar).

N_0 = Velocidad de giro inicial (RPM o Hz).

N = Velocidad de giro final (RPM o Hz).

Se determinó el valor de N para cada punto operativo proporcionado en la ficha técnica de la bomba CP190, utilizando como referencia las alturas de la bomba CP170 para obtener la presión final (H). La velocidad de giro inicial correspondió a la frecuencia nominal de 50 Hz, y la velocidad de giro final obtenida definió la frecuencia utilizada para las pruebas del Escenario 2.

El cálculo de cada N por punto operativo se aprecia en la Tabla 12, dando como resultado valores distintos entre ellos, por lo que se utilizó el valor promedio entre estos, el cuál es de 45 Hz. Se estimó el valor de la presión con esta nueva velocidad de giro por cada punto de la curva, con el objeto de conocer que tan cercana es respecto a las presiones de la bomba CP170,

teniendo un error cuadrático medio (RMSE) de 1,43 m, lo que es aceptable para esta prueba.

Tabla 13. Obtención de la velocidad de giro por punto de operación de la bomba CP190.

Variables	Datos											
H ₀ (m)	48	46	44,5	43	41,5	40	38	36	34,5	32,5	30,5	
N ₀ (Hz)	50	50	50	50	50	50	50	50	50	50	50	
N (Hz)	46	45	46	46	46	46	46	46	45	44	42	
H _{cp170} (m)	41	38	37	36	35	33,5	32	30	27,5	25	22	
H _{45Hz} (m)	39	37	36	35	34	32	31	29	28	26	25	

Este procedimiento buscó lograr un funcionamiento más homogéneo entre ambas bombas y el sistema de bombeo. Esto es comúnmente utilizado en la práctica real al operar sistemas de bombeo conectados en paralelo, donde bombas idénticas proporcionan el mismo caudal y, al sumar estos caudales, la presión se mantiene constante.

9.1.7. APENDICE C1: Circuito electrónico y Diagrama de flujo del Sistema.

La esquemática final del sistema electrónico (Figura 37) contiene una red MODBUS compuesta por 2 variadores de frecuencia, 2 contadores de pulsos, un conversor RS-485 a TTL y el dispositivo maestro ESP32, todo esto facilitado por la interfaz física de transmisión de datos RS-485.

En efecto, el microcontrolador ESP32 envía consultas a través de la red MODBUS a los distintos dispositivos conectados por medio del conversor señalado, el cual habilita esta comunicación. Una regleta bipolar proporciona

un punto común para los terminales A y B de los equipos enlazados, y los pines rx y tx comunican físicamente al conversor con el ESP32.

En cuanto a la alimentación, esta fue cubierta por la fuente de poder seleccionada anteriormente, distribuyendo el voltaje de 13.8 VDC por medio de otra regleta bipolar donde se conectó el positivo y negativo de cada equipo. Con ayuda de un convertidor de voltaje de 5 VDC se alimentó al microcontrolador utilizado, y este último proporcionó el voltaje para el funcionamiento del conversor RS485 a TTL.

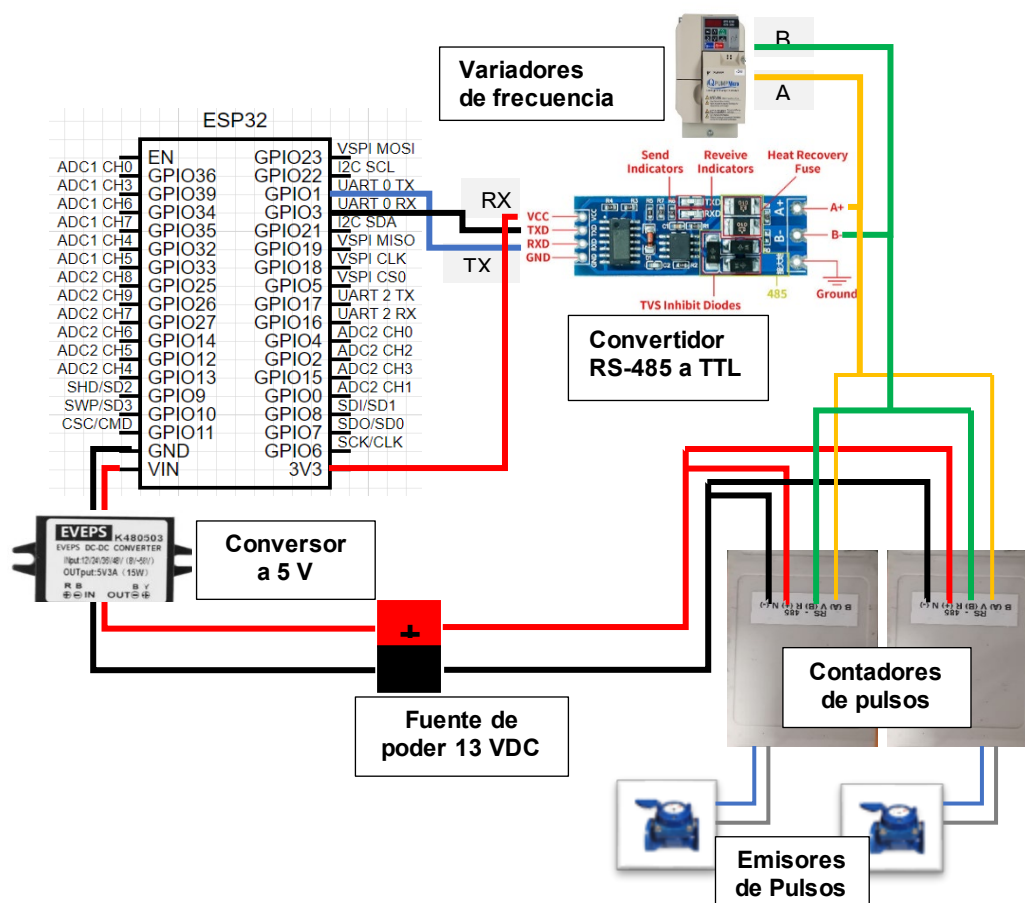


Figura 37. Circuito electrónico del sistema de medición.

El programa creado (Figura 38) comienza con la inicialización de la comunicación serial con el computador (UART0), la comunicación serial con MODBUS (UART2), inicialización de todas las variables, la conexión a WIFI y la conexión a Thingsboard, todo esto contenido en el void setup (). Ya en el bucle principal (void loop) el código aplica un condicional para verificar si existe la conexión a Thingsboard llamando a la función reconnect() y client.loop () para establecer nuevamente la conexión a la plataforma web.

Existiendo la conexión exitosa el programa procede a llamar a la función ConsultarPresion(), esta función envía, recibe y procesa el valor de la presión de cada bomba, definiendo variables booleanas que almacenan el estado de las variables “bomba1Encendida” y “bomba2Encendida”, factor clave para determinar si se procede o no a la consulta de los pulsos. Si el estado de estas últimas variables es “true”, significa que el valor de la presión es mayor que cero bar, entonces el sistema está prendido y hay caudal para medir. La función encargada del siguiente proceso es realizarConsultaYCalcularCaudal (), la cual consulta constantemente el estado de los pulsos para verificar si hay cambio en estos, condicional para proceder a la determinación del caudal por bomba. Teniendo el valor de caudal, se llama a la función consultarVariadores(), la cual proporciona los datos eléctricos faltantes, para que la función calcularRendimientoVariador() genere los parámetros de rendimientos solicitados y el *payload* enviado a la plataforma Thingsboard, reiniciando el ciclo nuevamente.

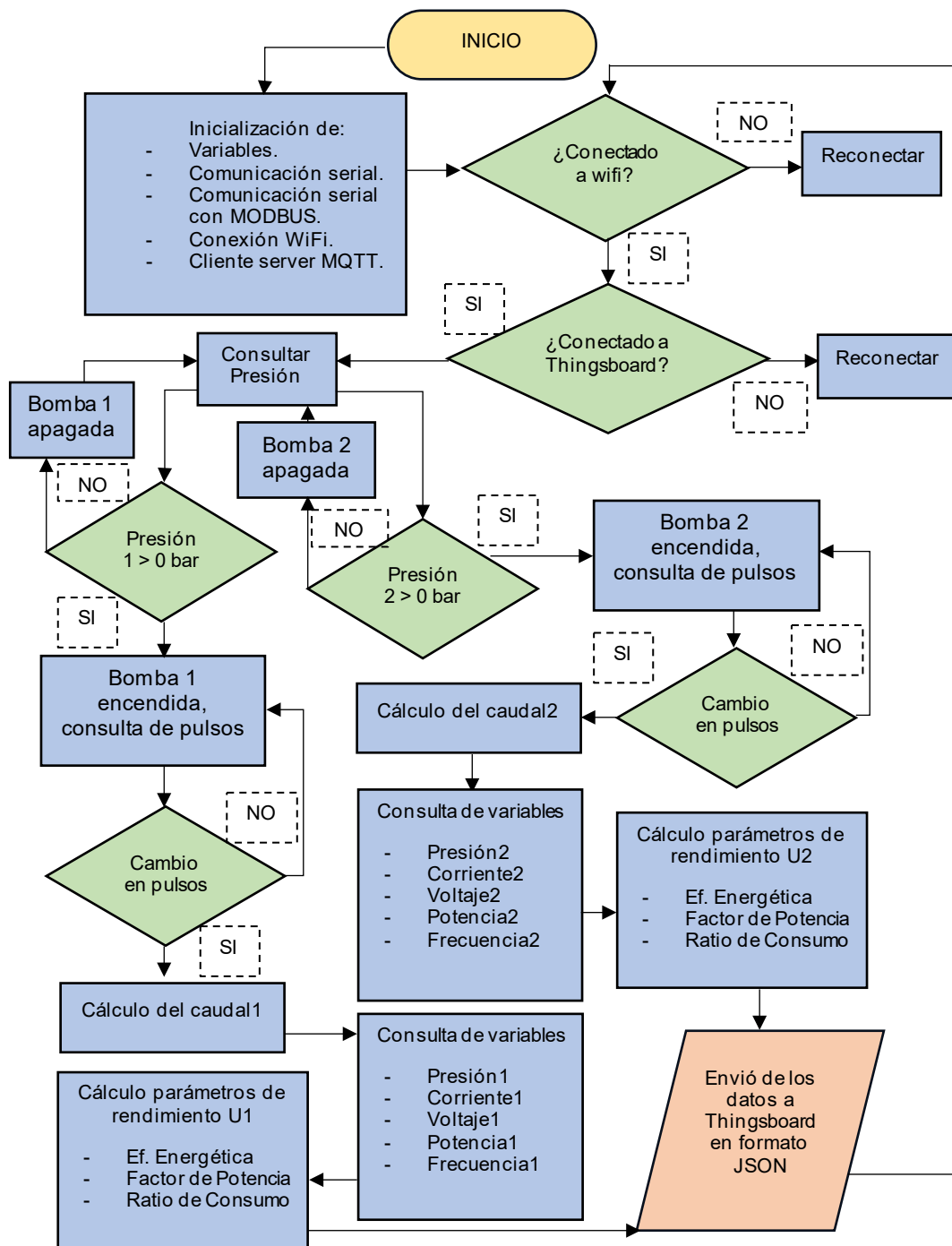


Figura 38. Diagrama de Flujo del código fuente.

9.1.8. APENDICE C2: Configuración completa de la Plataforma Thingsboard.


Para la utilización de ThingsBoard se tuvo que crear una cuenta de usuario en la edición comunitaria (Community Edition), la cuál es gratuita y de código abierto. Al estar suscrito, se creó un dispositivo denominado “Monitoreo Energético”, representado en la Figura 39, al cual deben llegar las variables e indicadores procesadas por el ESP32, ya sean de tipo decimal o booleanos sea el caso. La conexión entre el microcontrolador y Thingsboard está asociada a un token de acceso que permite la conexión mediante MQTT.

The screenshot displays the configuration page for a device named "Monitoreo Energético" in the Thingsboard interface. The page title is "Monitoreo Energético" with the subtitle "Detalles del dispositivo". A modal window titled "Credenciales del dispositivo" is open, showing the configuration for the device's credentials. The modal includes a "Tipo de credenciales" section with two options: "Access token" (selected) and "MQTT Basic". Below this, the "Tóken de acceso*" is displayed as "g96reugwnedb3n3resin". There are "Cancelar" and "Guardar" buttons at the bottom of the modal. The main configuration page also shows fields for "Nombre*" (Monitoreo Energético), "Perfil de dispositivo*" (default), and "Etiqueta". Below the modal, there are sections for "Firmware asignado" and "Software asignado".

Figura 39. Creación del dispositivo y su token de acceso.

Al estar vinculado el programa del microcontrolador con la plataforma en la nube, se hicieron pruebas para corroborar que los datos estaban llegando correctamente visualizando en la sección “última telemetría” si el buzón tenía variables recopiladas, funcionando correctamente el envío de información en formato Json.

Tras las pruebas de envío exitosa, se creó un panel para la visualización (véase Figura 40) de los datos denominado “MONITOREO ENERGÉTICO” en donde se muestran las variables involucradas en el funcionamiento del banco de pruebas, mediante distintos widgets que muestran el estado en tiempo real.



The image shows a configuration interface for a dashboard panel. At the top, there is a dark blue header with the title "MONITOREO ENERGÉTICO" and the subtitle "Detalles del panel". To the right of the header are three icons: a question mark, a close button (X), and a checkmark. Below the header, there are three main sections:

- Título***: A text input field containing "MONITOREO ENERGÉTICO".
- Descripción**: A text area containing "En este panel se podrá observar las variables que influyen en el rendimiento del sistema de bombeo".
- Ajustes de aplicación móvil**: A section with a toggle switch for "Ocultar panel en aplicación móvil" (currently off) and a text input field for "Orden de paneles en aplicación móvil".

Figura 40. Creación del panel de visualización de datos.

La organización del *dashboard* se dividió en dos partes, una movable ubicada al costado izquierdo, y otra estática en el lado derecho. En la sección izquierda se mostraron todas las variables de importancia (Caudal, Presión, Potencia, Voltaje, Corriente, Frecuencia) medidas con los sensores en tiempo real, tanto para la bomba CP190 como para la bomba CP170. Al bajar el panel se agregaron tablas con todas las variables ordenadas por series temporales, en conjunto con gráficos que muestran el comportamiento de algunas variables en el tiempo.

En la sección derecha del *dashboard* se implementaron los indicadores de rendimiento que indican como está funcionando el sistema (Eficiencia Energética de Bombeo, Factor de Potencia, Ratio de consumo), mostrando los valores en tiempo real. También se agregaron widgets de control que permiten conocer el comportamiento del microcontrolador, al igual que el estado de los dispositivos involucrados. En esta línea, se agregó 2 led indicadores que indican si las bombas están prendidas o apagadas de manera independiente, y un *widget* de conteo de alarmas.

Se abrió sesión en la aplicación móvil de Thingsboard con la cuenta creada, para recibir las notificaciones de alarma y visualizar también el comportamiento del monitoreo a través del celular.

Todos los widgets de visualización donde se muestran variables o parámetros de rendimiento en tiempo real se configuraron de la misma forma, seleccionando el dispositivo de donde proviene la información y la variable a mostrar, dándole su respectivo nombre y unidades correspondientes en cada caso.

The screenshot displays the configuration window for a 'Horizontal flow rate card with background' widget. The window title is 'Horizontal flow rate card with background' and it includes tabs for 'Básico' and 'Avanzado'. Below the title bar, there are buttons for 'Previsualizar', 'Descartar', and 'Aplicar'. The main configuration area is divided into sections:

- Origen de dato:** Includes a dropdown for 'Dispositivo' (set to 'Monitoreo Energético') and 'Alias de entidad'.
- Clave de datos*:** A dropdown set to 'caudal1'.
- Auto scale:** A checked toggle switch.
- Etiqueta:** A text input field containing 'CAUDAL CP190'.
- Icono:** A dropdown set to '40' and a unit dropdown set to 'px'.
- Valor:** A unit dropdown set to 'L/min' and a decimal places dropdown set to '0'.
- Fecha:** A dropdown set to 'Última actualización hace N'.

A preview of the widget is shown in the center, displaying the text 'CAUDAL CP170', 'Última actualización 2d hace', and '143 L/min' on a background image of a pump.

Figura 41. Configuración de widgets de visualización.

Este ejemplo (Figura 41) muestra la asignación del dispositivo “Monitoreo Energético”, el cual proporciona la variable “caudal1” correspondiente al caudal de la bomba CP190, se nombró al *widget* “CAUDAL CP190” con sus respectivas unidades en L/min. Se aplicó esta configuración para la muestra

en pantalla de caudal, presión, potencia, voltaje, corriente, frecuencia, eficiencia energética, ratio de consumo y factor de potencia por cada bomba.

De manera similar se configuraron los *widgets* de tabla (Figura 42), donde se asignó el dispositivo que origina los datos y se seleccionaron los valores a mostrar, que en este caso fueron todas las variables y parámetros procesados por bomba, viéndolos en conjunto agrupados por series temporales.

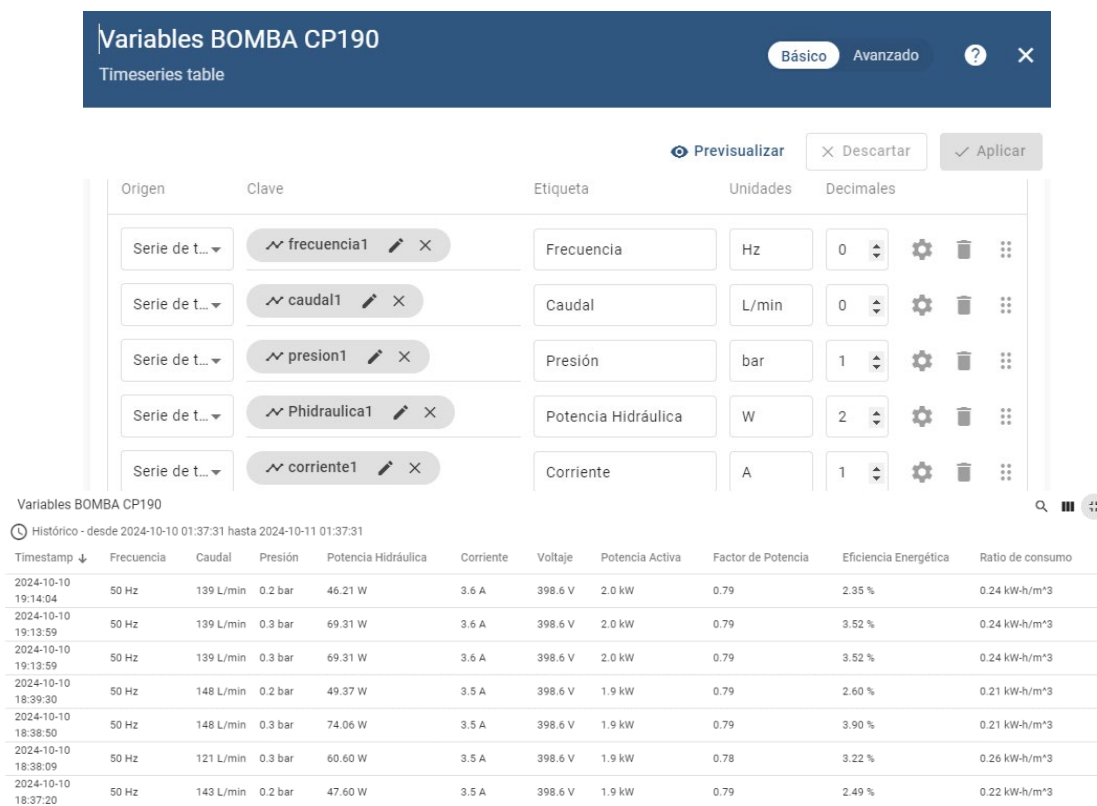


Figura 42. Configuración y visualización de widget de tabla.

La configuración del *widget* de gráfico es similar a la del *widget* de tabla, seleccionando el dispositivo y las variables a mostrar (Figura 43). En este caso se utilizaron las variables de presión y caudal para ambas bombas definiéndolas en el eje Y (puesto que el eje X esta predefinido para el tiempo),

al costado izquierdo esta la métrica de la presión (en bar), y al costado izquierdo la correspondiente al caudal (en L/min), como se muestra en la Figura 44.

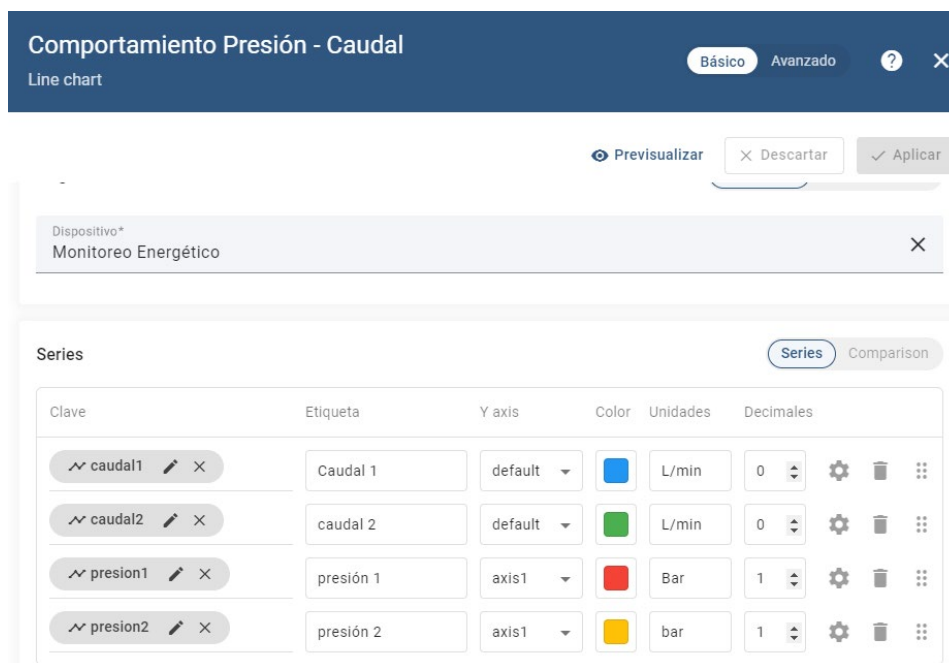


Figura 43. Configuración widget de gráfico.

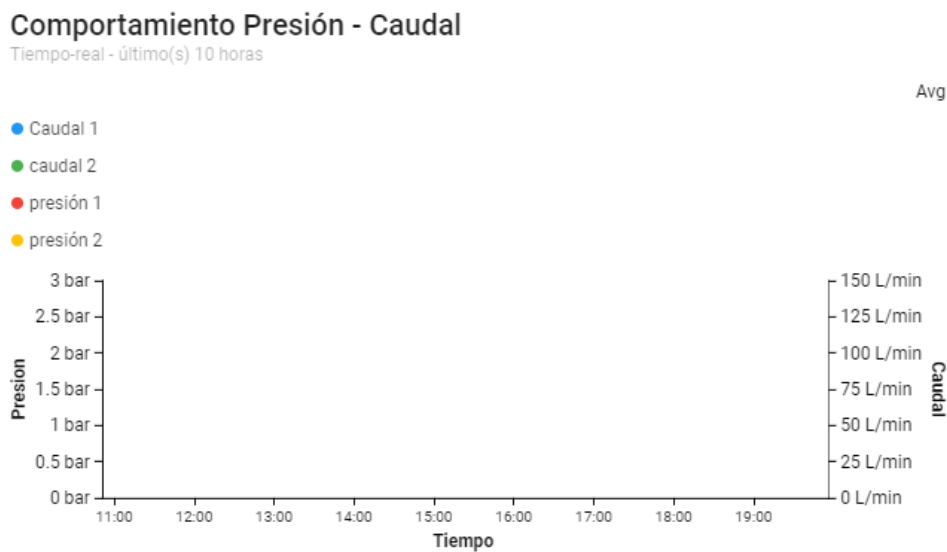


Figura 44. Visualización widget de gráfico.

Los *widgets* de led indicador se configuraron enviando desde el microcontrolador una variable de tipo booleana que entrega un true o un false (Figura 45), lo cual indica si la bomba estaba encendida o apagada. El *widget* se configuró para que reciba el estado de led mediante series temporales en base a la variable bomba1Encendida o bomba2Encendida.



Figura 45. Configuración y visualización de widget led indicador.

Por último, la configuración de alarma se creó mediante el perfil de dispositivo asociado al ESP32 utilizado, denominado en la plataforma “microcontrolador”. En la sección “detalles” existe un apartado el cual permite determinar reglas de alarma, como se puede apreciar en la Figura 46. Donde se realizaron dos alarmas, una relacionada a la presión y otra a la eficiencia.

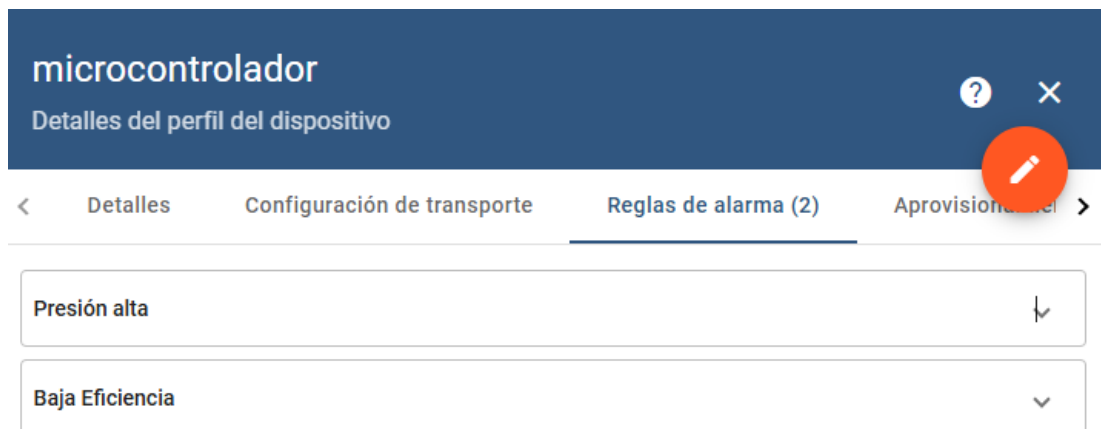


Figura 46. Creación de reglas de alarma en la plataforma.

La alarma denominada “Presión alta” actúa cuando los valores de presión que lleguen al buzón sean mayores de 3 bar, si esto ya no se cumple la alarma se apaga. Por otro lado, la alarma “Baja Eficiencia” se activa cuando lleguen valores de eficiencia bajo un 25,5% y que esto sucede 5 veces seguidas. La visualización de estas alarmas aparece en las notificaciones de la plataforma web como en las notificaciones de la aplicación móvil asociada a Thingsboard.

9.1.9. APENDICE C3: Código Fuente del sistema de monitoreo.

El código creado para este proyecto contó de 10 ficheros, 5 con la extensión .h (para declaraciones de variables y funciones), 4 con la extensión .cpp (desarrollo de la lógica del código) y un archivo .ino (sketch principal del programa). En donde se dividió la lógica según las acciones a tomar por parte del programa.

- main.ino

```
#include "config.h"
#include "modbus.h"
#include "sensores.h"
#include "procesamiento_datos.h"
#include "thingsboard.h"
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>

const unsigned long INTERVALO_CONSULTA_PRESION = 60000;
const unsigned long INTERVALO_CONSULTA_PULSOS = 2000;
// Timing variables
unsigned long ultimoTiempoConsultaPresion = 0;
unsigned long ultimoTiempoConsultaPulsos = 0;
bool primeraDeteccion = true;
// Bandera para verificar la primera detección
void setup () {
  Serial.begin(115200);
  setupWiFi ();
  client.setServer(mqtt_server, mqtt_port);
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
  //Serie para Modbus
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long tiempoActual = millis();

  // Verifique la presión continuamente hasta que cualquiera de las
  bombas esté encendida (primera detección)
  if (primeraDeteccion) {
    consultarPresion(); // Comprobación continua hasta que se detecta
    la bomba
  }
}
```

```

    if (bomba1Encendida || bomba2Encendida) {
        primeraDeteccion = false; // Detenga la verificación continua
una vez que se detecte la bomba
    }
}
//Una vez que cualquiera de las bombas esté encendida, verifique la
presión cada 30 segundos.
else if (tiempoActual - ultimoTiempoConsultaPresion >=
INTERVALO_CONSULTA_PRESION) {
    consultarPresion(); // Verifique la presión para actualizar el
estado de la bomba
    ultimoTiempoConsultaPresion = tiempoActual;

    // si las bombas están apagadas, continuar la consulta de presión
continua
    if (!bomba1Encendida && !bomba2Encendida) {
        primeraDeteccion = true; // Restablecer al modo continuo
    }
}

// Si alguna de las bombas está encendida, calcule el flujo cada 2
segundos.
if ((bomba1Encendida || bomba2Encendida) && (tiempoActual -
ultimoTiempoConsultaPulsos >= INTERVALO_CONSULTA_PULSOS)) {
    realizarConsultaYCalcularCaudal();
    ultimoTiempoConsultaPulsos = tiempoActual;
}
}
}

```

- config.h

```

#ifndef CONFIG_H
#define CONFIG_H
#include <Arduino.h>
#include <HardwareSerial.h>
#include <math.h>
#include <WiFi.h>
#include <PubSubClient.h>

// Pines de Modbus
#define RXD2 16
#define TXD2 17

```

```

// Buffer de respuesta
extern byte ByteArray[250];

// Variables del caudalímetro
extern const float litrosPorPulso;
extern const unsigned long INTERVALO_CONSULTA_PULSOS;

extern unsigned long ultimoTiempoConsultaPulsos;

// Estados de las bombas
extern bool bomba1Encendida;
extern bool bomba2Encendida;

#endif

```

- modbus.h

```

#ifndef MODBUS_H
#define MODBUS_H
#include <Arduino.h>

void enviarTrama(byte trama[], int len);
int recibirRespuesta();
void limpiarBuffer(byte buffer[], int len);

#endif

```

- procesamiento_datos.h

```

#ifndef PROCESAMIENTO_DATOS_H
#define PROCESAMIENTO_DATOS_H
#include <Arduino.h>

void consultarVariadores();
void calcularRendimientoVariador1();
void calcularRendimientoVariador2();

```

```

void procesarRespuestaPresion(byte ByteArray[], int numBytes, byte
direccion);
void procesarRespuestaDatos(byte ByteArray[], int numBytes, byte
direccion);

#endif
/* EN PROCESAMIENTO DE DATOS SE REALIZAN LAS ACCIONES QUE SIGUEN DESPUES
DE TENER UN VALOR DE CAUDAL VALIDO, LO CUAL ES CONSULTAR NUEVAMENTE LA
PRESION ACTUAL, LOS VALORES DE ENERGIA, PROCESAR ESTOS DATOS A ENTEROS O
FLOTANTES Y PODER CALCULAR PARAMETROS DE RENDIMIENTO CON ESTOS, LO CUAL
ES EL OBJETIVO
DE ESTE CODIGO, AQUI AL TENER LOS DATOS FALTA AGREGAR LA CONEXION A
THINGSBOARD PARA QUE SE PUEDAN VISUALIZAR EN LA WEB Y LUEGO DE ESTO
SE REPITA EL CICLO
*/

```

- sensores.h

```

#ifndef SENSORS_H
#define SENSORS_H
#include <Arduino.h>

void consultarPresion();
void realizarConsultaYCalcularCaudal();
int procesarRespuestaPulsos(byte ByteArray[], int numBytes, byte
direccion, byte funcion);
void calcularCaudalDispositivo1(int pulsosActuales);
void calcularCaudalDispositivo2(int pulsosActuales);
#endif
/* EN ESTE SKETCH SE REALIZAN LAS CONSULTAS A LOS SENSORES, PRINCIPALMENTE
A LOS VARIADORES DE FRECUENCIA, CON EL OBJETIVO DE CONOCER EL VALOR DE
PRESION ACTUAL QUE SE ESTA MIDIENDO, TENIENDO COMO FILTRO, EL VALOR DE
LA PRESIÓN, SI LA PRESIÓN VALE 0 BAR, ENTONCES EL SISTEMA DE BOMBEO ESTA
APAGADO Y POR LO TANTO
EL SISTEMA DE MEDICION NO REALIZA LAS ACCIONES, SIGUE CONSULTANDO HASTA
QUE ESTA CONDICION CAMBIE, Y PROCEDE A CONSULTAR A LOS CONTADORES DE
PULSOS EL ESTADO DE LOS PULSOS, SI ESTOS CAMBIAN CALCULA EL CAUDAL, EL
CÓDIGO ACTUA CON LA BOMBA 1 Y 2 DE MANERA INDEPENDIENTE, POR LO QUE SI UNA
ESTA APAGADA MIDE LA OTRA SIN PROBLEMA */

```

- thingsoard.h

```

#ifndef THINGSBOARD_H
#define THINGSBOARD_H
#include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>
#define ssid "Ignacio_wifi" //"LIB-9491272" //"Maxim0"o "Ignacio_wifi"
#define password "12345678" //"MAF632b6bnab" // "974658938" o
#define mqtt_server "demo.thingsboard.io"
#define token "g96reugwnedb3n3resin"
const int mqtt_port = 1883;
extern WiFiClient espClient;
extern PubSubClient client;
void setupWiFi();
void reconnect();
void enviarDatosThingsBoard(String payload);
#endif

```

- modbus.cpp

```

#include "modbus.h"
#include "config.h"
#include <Arduino.h>

byte ByteArray[250]; // Buffer de respuesta

void enviarTrama(byte trama[], int len) { // se envía la trama de la
consulta en hexadecimal a los dispositivos consultados.
    for (int i = 0; i < len; i++) {
        Serial2.write(trama[i]);
    }
    delay(500); // Breve pausa para la respuesta
}

int recibirRespuesta() { // se recibe la respuesta de estos
dispositivos también en hexadecimal, el procesamiento a enteros o
flotantes lo realiza otra función denominada procesar respuesta.
    int a = 0;
    unsigned long timeout = millis() + 1000;
    while (millis() < timeout) {
        if (Serial2.available()) {

```

```

        ByteArray[a] = Serial2.read();
        a++;
    }
}
return a;
}

void limpiarBuffer(byte buffer[], int len) { // limpia el buffer para
que no haya información estancada y existan malas lecturas, se aplica
cada
    for (int i = 0; i < len; i++) {
        // vez que se envía una trama y se recibe la respuesta.
        buffer[i] = 0;
    }
}
}

```

- sensores.cpp

```

#include "sensores.h"
#include "modbus.h"
#include "config.h"
#include "procesamiento_datos.h"
#include <Arduino.h>
#include "thingsboard.h"
// Variables para las bombas
bool bomba1Encendida = false;
bool bomba2Encendida = false;
float caudal1, caudal2;
extern float presion1, presion2;
// Variables para el cálculo del caudal de los dispositivos
int pulsosAnteriores1 = 0, pulsosAnteriores2 = 0;
unsigned long tiempoAnterior1 = 0, tiempoAnterior2 = 0;
bool primeraLectura1 = true, primeraLectura2 = true;
const float litrosPorPulso = 100.0;
extern const unsigned long INTERVALO_CONSULTA_PULSOS;
extern unsigned long ultimoTiempoConsultaPulsos;
void consultarPresion() {
    // Consulta presión de la bomba 1
    byte msgPresionBomba1[] = {0x03, 0x03, 0x00, 0x57, 0x00, 0x01, 0x34,
0x38}; // Trama de presión para bomba 1 (U1)
    enviarTrama(msgPresionBomba1, 8);
    int numBytesPresionBomba1 = recibirRespuesta();
}

```

```

procesarRespuestaPresion(ByteArray, numBytesPresionBomba1, 0x03); // U1

static bool bomba1EstadoAnterior = false;
// Estado anterior de la bomba 1 para detectar cambios

// Verificar si la presión es mayor a 0, indicando que la bomba 1 está
encendida
if (presion1 > 0) {
    if (!bomba1Encendida) { // Si la bomba estaba apagada, ahora se enciende
        bomba1Encendida = true;
        Serial.println("Bomba CP190 encendida, comenzando mediciones de caudal.");
    }

    // Solo enviar a ThingsBoard si ha habido un cambio en el estado de la bomba
    if (bomba1Encendida != bomba1EstadoAnterior) {
        String payload = "{\"bomba1Encendida\": true, \"bomba2Encendida\":
" + String(bomba2Encendida ? "true" : "false") + "}";
        enviarDatosThingsBoard(payload);
    }

    // Enviar el estado de las bombas a ThingsBoard
    bomba1EstadoAnterior = bomba1Encendida;
    // Actualizar el estado anterior
    }
} else {
    if (bomba1Encendida) { // Si la bomba estaba encendida, ahora se apaga
        bomba1Encendida = false;
        Serial.println("Bomba CP190 apagada, deteniendo mediciones.");
    }

    // Solo enviar a ThingsBoard si ha habido un cambio en el estado de la
bomba
    if (bomba1Encendida != bomba1EstadoAnterior) {
        String payload = "{\"bomba1Encendida\": false, \"bomba2Encendida\":
" + String(bomba2Encendida ? "true" : "false") + "}";
        enviarDatosThingsBoard(payload);
    }
}
// Enviar el estado de las bombas a ThingsBoard
bomba1EstadoAnterior = bomba1Encendida;
// Actualizar el estado anterior
}
}

// Consulta presión de la bomba 2 (lógica similar a la de bomba 1)

```

```

    byte msgPresionBomba2[] = {0x04, 0x03, 0x00, 0x57, 0x00, 0x01, 0x35,
0x8F}; // Trama de presión para bomba 2 (U2)
    enviarTrama(msgPresionBomba2, 8);
    int numBytesPresionBomba2 = recibirRespuesta();
    procesarRespuestaPresion(ByteArray, numBytesPresionBomba2, 0x04); // U2

    static bool bomba2EstadoAnterior = false; // Estado anterior de la bomba
2 para detectar cambios

    // Verificar si la presión es mayor a 0, indicando que la bomba 2 está
encendida
    if (presion2 > 0) {
        if (!bomba2Encendida) { // Si la bomba estaba apagada, ahora se enciende
            bomba2Encendida = true;
            Serial.println("Bomba CP170 encendida, comenzando mediciones de caudal.");
        }

        // Solo enviar a ThingsBoard si ha habido un cambio en el estado de la bomba

        if (bomba2Encendida != bomba2EstadoAnterior) {
            String payload = "{\"bomba1Encendida\": " + String(bomba1Encendida
? "true" : "false") + ", \"bomba2Encendida\": true}";
            enviarDatosThingsBoard(payload); // Enviar el estado de las bombas a
ThingsBoard
            bomba2EstadoAnterior = bomba2Encendida; // Actualizar el estado
anterior
        }
    }
} else {
    if (bomba2Encendida) { // Si la bomba estaba encendida, ahora se apaga
        bomba2Encendida = false;
        Serial.println("Bomba CP170 apagada, deteniendo mediciones.");

        // Solo enviar a ThingsBoard si ha habido un cambio en el estado de la
bomba

        if (bomba2Encendida != bomba2EstadoAnterior) {
            String payload = "{\"bomba1Encendida\": " + String(bomba1Encendida
? "true" : "false") + ", \"bomba2Encendida\": false}";
            enviarDatosThingsBoard(payload); // Enviar el estado de las bombas
a ThingsBoard
            bomba2EstadoAnterior = bomba2Encendida; // Actualizar el estado
anterior
        }
    }
}

```

```

    }
}

void realizarConsultaYCalcularCaudal() {
    if (bomba1Encendida) {
        // Consulta de dispositivo 1 (pulsos)
        byte msg1[] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x02, 0xC4, 0x38}; // Trama
para el dispositivo 1
        enviarTrama(msg1, 8);
        int numBytes1 = recibirRespuesta();
        int pulsos1 = procesarRespuestaPulsos(ByteArray, numBytes1, 0x01, 0x03);
        if (pulsos1 >= 0) { // Si se recibió una respuesta válida
            calcularCaudalDispositivo1(pulsos1); // Calcular el caudal para el
dispositivo 1
        }
        limpiarBuffer(ByteArray, 250); // Limpia el buffer antes de la siguiente
lectura
    }

    if (bomba2Encendida) {
        // Consulta de dispositivo 2 (pulsos)
        byte msg2[] = {0x02, 0x03, 0x00, 0x00, 0x00, 0x02, 0xC4, 0x38}; // Trama
para el dispositivo 2
        enviarTrama(msg2, 8);
        int numBytes2 = recibirRespuesta();
        int pulsos2 = procesarRespuestaPulsos(ByteArray, numBytes2, 0x02, 0x03);
        if (pulsos2 >= 0) { // Si se recibió una respuesta válida
            calcularCaudalDispositivo2(pulsos2); // Calcular el caudal para el
dispositivo 2
        }
        limpiarBuffer(ByteArray, 250); // Limpia el buffer antes de la siguiente
lectura
    }
}

int procesarRespuestaPulsos(byte ByteArray[], int numBytes, byte direccion,
byte funcion) {
    int pulsos = -1; // Inicializa con -1 para indicar error si no se recibe
un valor válido

    // Verifica que la respuesta es válida

```

```

    if (numBytes >= 7 && ByteArray[0] == direccion && ByteArray[1] == funcion
    && ByteArray[2] == 0x04) {
        pulsos = ByteArray[5] * 256 + ByteArray[6]; // Procesa los pulsos
        (segundo registro)
        Serial.print("Dispositivo ");
        Serial.print(direccion);
        Serial.print(" - Pulsos: ");
        Serial.println(pulsos);
    } else {
        Serial.println("ERROR: Respuesta incorrecta o incompleta.");
    }

    return pulsos;
}

void calcularCaudalDispositivo1(int pulsosActuales) {
    unsigned long tiempoActual = millis(); // Tiempo actual en milisegundos

    // Ignorar la primera lectura
    if (primeraLectura1) {
        pulsosAnteriores1 = pulsosActuales;
        tiempoAnterior1 = tiempoActual;
        primeraLectura1 = false; // Marcar como no es la primera lectura
        Serial.println("Dispositivo 1 - Primera lectura, sin cálculo de
caudal.");
        return;
    }

    // Si los pulsos cambiaron, calcula el caudal
    if (pulsosActuales > pulsosAnteriores1) {
        float tiempoTranscurrido = (tiempoActual - tiempoAnterior1) / 1000.0; //
De ms a segundos
        float tiempoTranscurridoMin = tiempoTranscurrido / 60.0;
        caudal1 = (litrosPorPulso / tiempoTranscurridoMin);

        Serial.print("Dispositivo 1 - Caudal: ");
        Serial.print(caudal1);
        Serial.println(" litros/minuto");
    }
}

```

```

    Serial.print("Dispositivo 1 - Tiempo entre pulsos: ");
    Serial.print(tiempoTranscurrido);
    Serial.println(" segundos");
// Consultar variadores y calcular rendimiento solo después de obtener el
caudal
    consultarVariadores();
    calcularRendimientoVariador1();

    tiempoAnterior1 = tiempoActual;
    pulsosAnteriores1 = pulsosActuales;
} else {
    Serial.println("Dispositivo 1 - No hay cambio en los pulsos.");
}
}

void calcularCaudalDispositivo2(int pulsosActuales) {
unsigned long tiempoActual = millis(); // Tiempo actual en milisegundos

// Ignorar la primera lectura
if (primeraLectura2) {
    pulsosAnteriores2 = pulsosActuales;
    tiempoAnterior2 = tiempoActual;
    primeraLectura2 = false; // Marcar como no es la primera lectura
    Serial.println("Dispositivo 2 - Primera lectura, sin cálculo de caudal.");
    return;
}

// Si los pulsos cambiaron, calcula el caudal
if (pulsosActuales > pulsosAnteriores2) {
    float tiempoTranscurrido = (tiempoActual - tiempoAnterior2) / 1000.0;
// De ms a segundos
    float tiempoTranscurridoMin = tiempoTranscurrido / 60.0;
    caudal2 = (litrosPorPulso / tiempoTranscurridoMin);

    Serial.print("Dispositivo 2 - Caudal: ");
    Serial.print(caudal2);
    Serial.println(" litros/minuto");
}
}

```

```

    Serial.print("Dispositivo 2 - Tiempo entre pulsos: ");
    Serial.print(tiempoTranscurrido);
    Serial.println(" segundos");
// Consultar variadores y calcular rendimiento solo después de obtener el
caudal
    consultarVariadores();
    calcularRendimientoVariador2();

    tiempoAnterior2 = tiempoActual;
    pulsosAnteriores2 = pulsosActuales;
} else {
    Serial.println("Dispositivo 2 - No hay cambio en los pulsos.");
}
}
}

```

- procesamiento_datos.cpp

```

#include "procesamiento_datos.h"
#include "modbus.h"
#include "config.h"
#include <Arduino.h>
#include "thingsboard.h"
#include "sensores.h"
// Declaración de variables globales

extern float caudal1, caudal2;
float presion1, presion2;
float voltaje1, voltaje2;
float corriente1, corriente2;
float potencia1, potencia2;
float frecuencia1, frecuencia2;

void consultarVariadores() {
    if (bomba1Encendida) {
        // Consulta presión y datos de U1
        byte msgU1Presion[] = {0x03, 0x03, 0x00, 0x57, 0x00, 0x01, 0x34,
0x38}; // Trama de presión para U1
        byte msgU1Datos[] = {0x03, 0x03, 0x00, 0x24, 0x00, 0x04, 0x05,
0xE0}; // Trama de frecuencia, voltaje, corriente y potencia para U1
        enviarTrama(msgU1Presion, 8);
    }
}

```

```

int numBytesU1Presion = recibirRespuesta();
procesarRespuestaPresion(ByteArray, numBytesU1Presion, 0x03);
limpiarBuffer(ByteArray, 250); // Limpiar el buffer
enviarTrama(msgU1Datos, 8);
int numBytesU1Datos = recibirRespuesta();
procesarRespuestaDatos(ByteArray, numBytesU1Datos, 0x03);
limpiarBuffer(ByteArray, 250); // Limpiar el buffer
}

if (bomba2Encendida) {
    // Consulta presión y datos de U2
    byte msgU2Presion[] = {0x04, 0x03, 0x00, 0x57, 0x00, 0x01, 0x35,
0x8F}; // Trama de presión para U2
    byte msgU2Datos[] = {0x04, 0x03, 0x00, 0x24, 0x00, 0x04, 0x04,
0x57}; // Trama de frecuencia, voltaje, corriente y potencia para U2
    enviarTrama(msgU2Presion, 8);
    int numBytesU2Presion = recibirRespuesta();
    procesarRespuestaPresion(ByteArray, numBytesU2Presion, 0x04);
    limpiarBuffer(ByteArray, 250); // Limpiar el buffer
    enviarTrama(msgU2Datos, 8);
    int numBytesU2Datos = recibirRespuesta();
    procesarRespuestaDatos(ByteArray, numBytesU2Datos, 0x04);
    limpiarBuffer(ByteArray, 250); // Limpiar el buffer
}
}

void procesarRespuestaPresion(byte ByteArray[], int numBytes, byte
direccion) {
    if (numBytes >= 7 && ByteArray[0] == direccion && ByteArray[1] ==
0x03 && ByteArray[2] == 0x02) {
        int presion = ByteArray[3] * 256 + ByteArray[4];
        float presionBar = presion / 10.0;

        if (direccion == 0x03) {
            presion1 = presionBar; // Asignar a la presión de la bomba 1
(U1)
        } else if (direccion == 0x04) {
            presion2 = presionBar; // Asignar a la presión de la bomba 2
(U2)
        }
    }

    Serial.print("Variador U");
    Serial.print(direccion - 2); // Para mostrar U1 o U2
}

```

```

Serial.print(" - Presión: ");
Serial.print(presionBar);
Serial.println(" bar");
} else {
Serial.print("ERROR: Respuesta incorrecta o incompleta para presión
U");
Serial.println(direccion - 2);
}
}

void procesarRespuestaDatos(byte ByteArray[], int numBytes, byte
direccion) {
if (numBytes >= 13 && ByteArray[0] == direccion && ByteArray[1] ==
0x03 && ByteArray[2] == 0x08) {
// Frecuencia
int frecuencia = ByteArray[3] * 256 + ByteArray[4];
float frecuenciaHz = frecuencia / 100.0;
// Voltaje
int voltaje = ByteArray[5] * 256 + ByteArray[6];
float voltajeV = voltaje / 10.0;
// Corriente
int corriente = ByteArray[7] * 256 + ByteArray[8];
float corrienteA = corriente / 10.0;
// Potencia
int potencia = ByteArray[9] * 256 + ByteArray[10];
float potenciaKW = potencia / 100.0;

// Asignar los valores a las variables correspondientes
if (direccion == 0x03) {
voltaje1 = voltajeV;
corriente1 = corrienteA;
potencia1 = potenciaKW;
frecuencia1 = frecuenciaHz;
} else if (direccion == 0x04) {
voltaje2 = voltajeV;
corriente2 = corrienteA;
potencia2 = potenciaKW;
frecuencia2 = frecuenciaHz;
}

Serial.print("Variador U");
Serial.print(direccion - 2); // Para mostrar U1 o U2
Serial.print(" - Frecuencia: ");

```

```

Serial.print(frecuenciaHz);
Serial.println(" Hz");

Serial.print("Variador U");
Serial.print(direccion - 2);
Serial.print(" - Voltaje: ");
Serial.print(voltajeV);
Serial.println(" V");

Serial.print("Variador U");
Serial.print(direccion - 2);
Serial.print(" - Corriente: ");
Serial.print(corrienteA);
Serial.println(" A");

Serial.print("Variador U");
Serial.print(direccion - 2);
Serial.print(" - Potencia: ");
Serial.print(potenciaKW);
Serial.println(" kW");
} else {
    Serial.println("ERROR: Respuesta incorrecta o incompleta para
datos.");
}
}

void calcularRendimientoVariador1() {
    if (caudal1 > 0 && voltaje1 > 0 && corriente1 > 0 && potencia1 > 0 &&
presion1 > 0) {
        const float raizDeTres = sqrt(3);
        // Imprimir los valores de entrada para verificar que son correctos
        Serial.println("=== Valores de entrada para cálculo eléctrico BOMBA
1 CP190 ===");
        Serial.print("Potencia: ");
        Serial.print(potencia1);
        Serial.println(" kW");
        Serial.print("Voltaje: ");
        Serial.println(voltaje1);
        Serial.println(" V");
        Serial.print("Corriente: ");
        Serial.print(corriente1);
        Serial.println(" A");
        Serial.print("Raíz de tres: ");

```

```

Serial.println(raizDeTres);

// 1. Cálculo del cos(phi)
float cosPhi = (potencia1*1000) / (voltaje1 * corriente1 *
raizDeTres);

// Verificar si cosPhi está dentro del rango permitido
if (cosPhi < -1.0 || cosPhi > 1.0) {
    Serial.println("Error: El valor de cosPhi está fuera de los
límites permitidos (-1 a 1).");
    return;
}

// 2. Cálculo del ángulo phi (en radianes)
float phi = acos(cosPhi);

// 3. Cálculo de la potencia reactiva
float potenciaReactiva1 = voltaje1 * corriente1 * raizDeTres *
sin(phi);

// 4. Cálculo de la potencia aparente
float potenciaAparente1 = sqrt(pow(potencia1*1000, 2) +
pow(potenciaReactiva1, 2));

// Verificar que no haya división por cero en el factor de potencia
if (potenciaAparente1 == 0) {
    Serial.println("Error: Potencia aparente es cero. No se puede
calcular el factor de potencia.");
    return;
}

// 5. Cálculo del factor de potencia
float factorDePotencia1 = (potencia1*1000) / potenciaAparente1;

//6. Cálculo de la Potencia Hidráulica
float PHidraulica_1 = (presion1* 10.2)*1000*9.81*(caudal1/60000.0);
// presion se pasa a metros y caudal a m^3 / s.

//7. Cálculo de la eficiencia energética.
float Efenergetica1 = ( PHidraulica_1/(potencia1*1000))*100 ;

// 8. Cálculo de el Ratio de consumo.

```

```

float Ratio_consumo1 = potencia1 / (caudal1*0.06); // caudal en
m^3/h

// Mostrar los resultados
Serial.print("Cos(phi) U1: ");
Serial.print(cosPhi, 3);
Serial.print( " unidades");
Serial.print("Phi U1: ");
Serial.print(phi, 3);
Serial.println( " rad");
Serial.print("Potencia Reactiva U1: ");
Serial.print(potenciaReactiva1, 3);
Serial.println(" VAR");
Serial.print("Potencia Aparente U1: ");
Serial.print(potenciaAparente1, 3);
Serial.println(" VA");
Serial.print("Factor de Potencia U1: ");
Serial.println(factorDePotencia1, 3);
Serial.print("Potencia Hidráulica U1; ");
Serial.print(PHidraulica_1, 2);
Serial.println(" W");
Serial.print("Eficiencia Energética U1; ");
Serial.print(Efenergetica1, 2);
Serial.println(" % ");
Serial.print("Ratio de Consumo U1: ");
Serial.print(Ratio_consumo1, 2);
Serial.println(" kW-h/m^3 ");

// Enviar los datos de la bomba 1 CP190 a ThingsBoard
if(caudal1<= 150) {
String payload = "{";
payload += "\"frecuencia1\":" + String(frecuencia1) + ",";
payload += "\"caudal1\":" + String(caudal1) + ",";
payload += "\"presion1\":" + String(presion1) + ",";
payload += "\"voltaje1\":" + String(voltaje1) + ",";
payload += "\"corriente1\":" + String(corriente1) + ",";
payload += "\"potencia1\":" + String(potencia1) + ",";
payload += "\"factordepotencia1\":" + String(factorDePotencia1)
+ ",";
payload += "\"Phidraulica1\":" + String(PHidraulica_1) + ",";
payload += "\"Efenergética1\":" + String(Efenergetica1) + ",";
payload += "\"Ratiodeconsumo1\":" + String(Ratio_consumo1);

```

```

    payload += "}";

    // Llamada para enviar el payload a ThingsBoard
    enviarDatosThingsBoard(payload);
    }
    else {
    // Manejar el caso de caudal erróneo
    Serial.println("Caudal fuera de rango: " + String(caudal2));
    }
    }
}

void calcularRendimientoVariador2() {
    if (caudal2 > 0 && voltaje2 > 0 && corriente2 > 0 && potencia2 > 0 &&
presion2 > 0) {
        const float raizDeTres = sqrt(3);
        // Imprimir los valores de entrada para verificar que son correctos
        Serial.println("=== Valores de entrada para cálculo eléctrico BOMBA
2 CP170 ===");
        Serial.print("Potencia U2: ");
        Serial.print(potencia2);
        Serial.println(" kW");
        Serial.print("Voltaje U2: ");
        Serial.print(voltaje2);
        Serial.println(" V");
        Serial.print("Corriente U2: ");
        Serial.print(corriente2);
        Serial.println(" A");
        Serial.print("Raíz de tres: ");
        Serial.println(raizDeTres);

        // 1. Cálculo del cos(phi)
        float cosPhi_2 = (potencia2*1000) / (voltaje2 * corriente2 *
raizDeTres);

        // Verificar si cosPhi está dentro del rango permitido
        if (cosPhi_2 < -1.0 || cosPhi_2 > 1.0) {
            Serial.println("Error: El valor de cosPhi está fuera de los
límites permitidos (-1 a 1).");
            return;
        }

        // 2. Cálculo del ángulo phi (en radianes)

```

```

float phi_2 = acos(cosPhi_2);

// 3. Cálculo de la potencia reactiva
float potenciaReactiva_2 = voltaje2 * corriente2 * raizDeTres *
sin(phi_2);

// 4. Cálculo de la potencia aparente
float potenciaAparente_2 = sqrt(pow(potencia2*1000, 2) +
pow(potenciaReactiva_2, 2));

// Verificar que no haya división por cero en el factor de potencia
if (potenciaAparente_2 == 0) {
    Serial.println("Error: Potencia aparente es cero. No se puede
calcular el factor de potencia.");
    return;
}

// 5. Cálculo del factor de potencia
float factorDePotencia_2 = (potencia2*1000) / potenciaAparente_2;

//6. Cálculo de la Potencia Hidráulica
float PHidraulica_2 = (presion2* 10.2)*1000*9.81*(caudal2/60000.0);
// caudal en m^3/s y presión en m

//7. Cálculo de la eficiencia energética.
float Efenergetica_2 = (PHidraulica_2/ (potencia2*1000))*100 ;

// 8. Cálculo de el Ratio de consumo.
float Ratio_consumo2 = potencia2 / (caudal2*0.06); // caudal en
m^3/h

// Mostrar los resultados
Serial.print("Cos U2(phi): ");
Serial.print(cosPhi_2, 3);
Serial.println(" unidades");
Serial.print("Phi U2: ");
Serial.print(phi_2, 3);
Serial.println(" rad");
Serial.print("Potencia Reactiva U2: ");
Serial.print(potenciaReactiva_2, 3);
Serial.println(" VAR");
Serial.print("Potencia Aparente U2: ");
Serial.print(potenciaAparente_2, 3);

```

```

Serial.println(" VA");
Serial.print("Factor de Potencia U2: ");
Serial.print(factorDePotencia_2, 3);
Serial.println( " unidades");
Serial.print("Potencia Hidráulica U2: ");
Serial.print(PHidraulica_2, 2);
Serial.println(" W");
Serial.print("Eficiencia Energética U2: ");
Serial.print(Efenergetica_2, 2);
Serial.println(" % ");
Serial.print("Ratio de Consumo U2: ");
Serial.print(Ratio_consumo2, 2);
Serial.println(" kW-h/m^3 ");

// Enviar los datos de la bomba 2 CP170 a ThingsBoard
if (caudal2 <= 150) {
String payload = "{";
payload += "\"frecuencia2\":" + String(frecuencia2) + ",";
payload += "\"caudal2\":" + String(caudal2) + ",";
payload += "\"presion2\":" + String(presion2) + ",";
payload += "\"voltaje2\":" + String(voltaje2) + ",";
payload += "\"corriente2\":" + String(corriente2) + ",";
payload += "\"potencia2\":" + String(potencia2) + ",";
payload += "\"factordepotencia2\":" + String(factorDePotencia_2)+
",";
payload += "\"Phidraulica2\":" + String(PHidraulica_2) + ",";
payload += "\"Efenergética2\":" + String(Efenergetica_2) + ",";
payload += "\"Ratiodeconsumo2\":" + String(Ratio_consumo2);
payload += "}";

// Llamada para enviar el payload a ThingsBoard
enviarDatosThingsBoard(payload);
}
else {
// Manejar el caso de caudal erróneo
Serial.println("Caudal fuera de rango: " + String(caudal2));
}
}
}
}

```

- thingsboard.cpp

```
#include "thingsboard.h"
#include <Arduino.h>
#include <ArduinoJson.h>

// Variables globales
WiFiClient espClient;
PubSubClient client(espClient); // Definición única de `client`

// Conectar a WiFi
void setupWiFi() {
    delay(10);
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi conectado.");
    Serial.println("Dirección IP: ");
    Serial.println(WiFi.localIP());
}

// Conectar a ThingsBoard
void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando a ThingsBoard...");
        if (client.connect("ESP32", token, NULL)) {
            Serial.println("Conectado");
            client.subscribe("v1/devices/me/rpc/request/+"); // Suscribirse a
los RPC
            Serial.println("Suscrito a RPC de ThingsBoard.");
        } else {
            Serial.print("Fallo, rc=");
            Serial.print(client.state());

```

```
        Serial.println(" intentando de nuevo en 5 segundos");
        delay(5000);
    }
}
}

void enviarDatosThingsBoard(String payload) {
    client.loop(); // Asegúrate de que la conexión MQTT esté activa

    // Publicar el payload en ThingsBoard
    Serial.print("Enviando datos a ThingsBoard: ");
    Serial.println(payload);

    // Publicar los datos en el tópico adecuado
    client.publish("v1/devices/me/telemetry", payload.c_str());
}
```

9.2. ANEXO A.

PROTOCOLO DE INSTALACIÓN DE LOS EQUIPOS.

Equipos	Protocolo de instalación
Bombas Centrifugas CP	<p>Altura de impulsión menor a 7 m. Bomba lo más cercana a fuente de agua. Entradas con sellado hermético. Base con capacidad de absorber vibraciones. Conexión succión roscada hembra BSP 1” Conexión descarga roscada hembra BSP 1 $\frac{1}{4}$”</p>
Caudalímetros Tecnidro.	<p>Tramos rectos para la medición. Distancia entre singularidad y entrada ≥ 10 diámetros. Distancia entre singularidad y salida ≥ 5 diámetros. Entrada y salida de 50 mm o 2”.</p>
Filtro Azud hélix Automatic 201 DLP	<p>Altura mínima colector de drenaje de 843 mm. Separación entre soportes de 560 mm. Conexión salida roscada macho BSP 2”. Conexión drenaje roscada macho BSP 2”. Conexión entrada de 2” encolar.</p>

DATOS TÉCNICOS BOMBAS CENTRÍFUGAS.

Prestaciones.	CP 170	CP190
Rango de presión	41 – 22 m	48 – 26 m
Rango de Caudal	Hasta 160 L/min	Hasta 160 L/min
Potencia P ₂	1,1 kW	1,5 kW
Frecuencia	50 Hz	50 Hz
Eficiencia	34%	33%

DATOS TÉCNICOS CAUDALIMETROS.

Prestaciones	Serie TW	Serie AW
Tipo	Tangencial	Woltman
Diámetro nominal	50 mm	50 mm
Constante de flujo (K)	100L / Pulso	100L / Pulso
Precisión	Q ₂ ÷ Q ₄ ± 2% Q ₁ ÷ Q ₂ ± 5%	Q ₂ ÷ Q ₄ ± 2% Q ₁ ÷ Q ₂ ± 5%
Normas	ISO 16399	ISO 16399

CURVAS DE PRESTACIONES DE BOMBAS LINEA CP.

