

## **Sistema de Monitoreo de Variables de un Grupo Electrónico**

John Alberto Grimaldo González

Juan Carlos Garay Lozano

Asesor:

Sandra Liliana Castellanos Franco

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Ingeniería Electrónica

2024

### **Agradecimiento**

A Dios por brindarnos la oportunidad de tener una educación de calidad y por ser apoyo en tiempo de agotamiento, a nuestra familia y amigos que mostraron empatía en el proceso educativo y fueron fuente de inspiración para poder lograr nuestro sueño.

## Resumen

Un Grupo Electrónico (GE) se puede definir como un sistema de emergencia utilizado como fuente de energía eléctrica alterna, por lo general, para contrarrestar fallas o interrupciones de energía en las redes de distribución comercial, especialmente en aquellos casos en los que la falta de esta puede traer como consecuencia daños importantes, o en lugares en los que las redes eléctricas son insuficientes, no están disponibles o son poco rentables. (Francisco et al., 2007)

Un GE entra en funcionamiento cuando existe una interrupción en el suministro de energía eléctrica, mediante la conversión de energía primaria utilizando un motor diésel y un generador eléctrico, este sistema auxiliar cuenta con la capacidad de compensar la falta de energía proveniente de la red y repartirla entre las cargas que tienen mayor importancia. (Angulo Hernández & Yarleque Chunga, 2018) Debido a su naturaleza, estos sistemas requieren un mantenimiento periódico que garantice su correcto funcionamiento en todo momento, puesto que una falla operativa del sistema de emergencia puede ocasionar pérdidas incalculables. El presente proyecto aplicado plantea el diseño de un sistema de monitoreo remoto que permita medir variables críticas en un Grupo Electrónico, con el fin de detectar de manera temprana anomalías en el funcionamiento del generador contribuyendo a minimizar el riesgo de fallas y pérdidas económicas asociadas.

El sistema de monitoreo diseñado mostró capacidad para realizar un seguimiento preciso y programado de diferentes variables; la integración de sensores industriales con un dispositivo embebido de bajo costo y el uso de tecnologías IoT proporcionaron una solución efectiva para mejorar la eficiencia operativa y garantizar el rendimiento óptimo del Grupo Electrónico; se logró que el sistema mantuviera una comunicación confiable por medio de la red Wi-Fi haciendo uso del protocolo API HTTP(s) y se consiguió visualización en tiempo real de las variables

medidas mediante un dashboard a través de un aplicativo Web de uso gratuito. Los resultados obtenidos en la etapa de pruebas y validación del sistema de monitoreo no solo llevan al cumplimiento de los objetivos planteados para el proyecto, además de esto, se consigue mantener una baja inversión económica para la construcción del dispositivo.

***Palabras clave:*** Grupo Electrónico, Industria 4.0, Internet de las Cosas, Monitoreo de variables.

### **Abstract**

A generator set (GE) can be defined as an emergency system used as a source of alternative electrical energy, generally, to counteract power failures or interruptions in commercial distribution networks, especially in those cases in which the lack of energy can result in significant damage, or in places where electrical networks are insufficient, unavailable or unprofitable. (Marielys et al., 2007)

A GE comes into operation when there is an interruption in the supply of electrical energy, by converting primary energy using a diesel engine and an electric generator. This auxiliary system has the capacity to compensate for the lack of energy coming from the network and distribute it among the loads that are most important. (Angulo Hernández & Yarleque Chunga, 2018) Due to their nature, these systems require periodic maintenance to guarantee their correct functioning at all times, since an operational failure of the emergency system can cause incalculable losses. This applied project proposes the design of a remote monitoring system that allows measuring critical variables in a Generator Set, in order to early detect anomalies in the operation of the generator, contributing to minimizing the risk of failures and associated economic losses.

The designed monitoring system showed the ability to carry out precise and scheduled monitoring of different variables; The integration of industrial sensors with a low-cost embedded device and the use of IoT technologies provided an effective solution to improve operational efficiency and ensure optimal performance of the Generator Set; It was possible for the system to maintain reliable communication through the Wi-Fi network using the HTTP(s) API protocol and real-time visualization of the measured variables was achieved through a dashboard through a Web application that was free to use. The results obtained in the testing and validation stage of

the monitoring system not only lead to the fulfillment of the objectives set for the project, in addition to this, it is possible to maintain a low economic investment for the construction of the device.

***Keywords:*** Generator Set, Industry 4.0, Internet of Things, Variable Monitoring.

## Tabla de Contenido

Introducción.....	14
Justificación .....	17
Planteamiento del Problema .....	19
Objetivos.....	21
Objetivo General.....	21
Objetivos Específicos .....	21
Alcance .....	22
Marco Conceptual.....	23
Generación Distribuida.....	23
Grupo Electrógeno.....	23
<i>Mantenimiento Preventivo y Correctivo</i> .....	25
Industria 4.0.....	25
<i>Cadenas de Suministro Inteligentes</i> .....	27
<i>Monitoreo de Equipos</i> .....	27
<i>Colaboración Hombre-Maquina:</i> .....	28
<i>Internet de las Cosas</i> .....	30
<i>Comunicación Machine to Machine – M2M</i> .....	35
Metodología.....	37
Análisis del Problema y Definición de Requisitos .....	37
Selección de Componentes y Tecnologías .....	38
<i>Tarjeta de Desarrollo</i> .....	39
<i>Sensores</i> .....	47

Diseño del Sistema .....	57
<i>Implementación del Hardware del sistema</i> .....	58
<i>Desarrollo del Software</i> .....	73
Integración del Sistema .....	83
Pruebas y Validación del Sistema .....	86
Resultados.....	88
Conclusiones.....	96
Referencias Bibliograficas.....	99
Apéndices .....	103

## Lista de Tablas

<b>Tabla 1</b> <i>Características de los Sensores</i> .....	32
<b>Tabla 2</b> <i>Comparativo entre las Tarjetas de Desarrollo Arduino, NodeMCU, ESP8266 y Raspberry</i> .....	46
<b>Tabla 3</b> <i>Características Sensor FZ0403</i> .....	48
<b>Tabla 4</b> <i>Características del Sensor de Presión VDO 10</i> .....	50
<b>Tabla 5</b> <i>Características del Sensor de Nivel 100TECH</i> .....	51
<b>Tabla 6</b> <i>Características Sensor de Temperatura DS18B20</i> .....	53
<b>Tabla 7</b> <i>Características del Módulo de Relé</i> .....	55
<b>Tabla 8</b> <i>Caracterización del Sensor de Voltaje</i> .....	61
<b>Tabla 9</b> <i>Caracterización del Sensor de Presión VDO 10</i> .....	65
<b>Tabla 10</b> <i>Caracterización del Sensor de Combustible 100TECH</i> .....	68
<b>Tabla 11</b> <i>Validación de Datos Obtenidos por los Sensores</i> .....	92

## Lista de Figuras

<b>Figura 1</b> <i>Las Cuatro Etapas de la Revolución Industrial</i> .....	26
<b>Figura 2</b> <i>Pilares de la Industria 4.0</i> .....	29
<b>Figura 3</b> <i>IoT Industrial</i> .....	31
<b>Figura 4</b> <i>Diagrama de Bloques Sistema</i> .....	38
<b>Figura 5</b> <i>Tarjetas de Desarrollo Arduino</i> .....	39
<b>Figura 6</b> <i>Prototipo Implementado con Arduino</i> .....	40
<b>Figura 7</b> <i>Configuración de Pines ESP8266 NodeMCU</i> .....	42
<b>Figura 8</b> <i>Configuración de Pines ESP32</i> .....	43
<b>Figura 9</b> <i>Circuito de Prueba Implementado con ESP32</i> .....	44
<b>Figura 10</b> <i>Tarjeta de Desarrollo Raspberry Pi</i> .....	45
<b>Figura 11</b> <i>Sensor de Voltaje FZ0430</i> .....	48
<b>Figura 12</b> <i>Sensor de Presión de Aceite VDO10</i> .....	49
<b>Figura 13</b> <i>Sensor de Nivel Agua o Combustible 100TECH</i> .....	51
<b>Figura 14</b> <i>Sensor de Temperatura DS18B20</i> .....	52
<b>Figura 15</b> <i>Módulo de Relé</i> .....	54
<b>Figura 16</b> <i>Pantalla LCD 20X4 I2C</i> .....	56
<b>Figura 17</b> <i>Componentes Seleccionados para el Hardware del Sistema</i> .....	57
<b>Figura 18</b> <i>Diagrama de Conexiones Caracterización Sensor de Voltaje</i> .....	59
<b>Figura 19</b> <i>Código para Lectura del Sensor de Voltaje</i> .....	60
<b>Figura 20</b> <i>Caracterización del Sensor de Voltaje</i> .....	60
<b>Figura 21</b> <i>Gráfico de Caracterización del Sensor de Voltaje</i> .....	62
<b>Figura 22</b> <i>Circuito de Acople para el Sensor de Presión</i> .....	63

<b>Figura 23</b> <i>Diagrama de Conexiones Caracterización Sensor de Presión</i> .....	63
<b>Figura 24</b> <i>Código para Lectura del Sensor de Presión</i> .....	64
<b>Figura 25</b> <i>Gráfico de Caracterización del Sensor de Presión</i> .....	65
<b>Figura 26</b> <i>Diagrama de Conexiones Caracterización Sensor de Nivel de Combustible</i> .....	66
<b>Figura 27</b> <i>Código para Lectura del Sensor de Nivel de Combustible</i> .....	67
<b>Figura 28</b> <i>Caracterización del Sensor de Nivel de Combustible</i> .....	67
<b>Figura 29</b> <i>Gráfico de Caracterización del Sensor de Nivel de Combustible</i> .....	69
<b>Figura 30</b> <i>Tarjeta de Expansión de la Placa de Desarrollo ESP 32</i> .....	70
<b>Figura 31</b> <i>Diagrama de Conexión de Pantalla LCD</i> .....	71
<b>Figura 32</b> <i>Código para Pantalla LCD</i> .....	72
<b>Figura 33</b> <i>Plano Esquemático General del Sistema</i> .....	73
<b>Figura 34</b> <i>Dashboard con la Plataforma Blynk</i> .....	75
<b>Figura 35</b> <i>Diagrama de Flujo del Software del Sistema</i> .....	76
<b>Figura 36</b> <i>Inclusión de Librerías en el Código Fuente</i> .....	77
<b>Figura 37</b> <i>Definición de Pines de Entrada y Salida</i> .....	78
<b>Figura 38</b> <i>Definición de Puertos Virtuales Blynk</i> .....	79
<b>Figura 39</b> <i>Función Void setup() en el Código Fuente</i> .....	80
<b>Figura 40</b> <i>Fragmento de Código Ciclo Void loop()</i> .....	81
<b>Figura 41</b> <i>Impresión de Variables por Consola</i> .....	81
<b>Figura 42</b> <i>Código para Visualización de Variables en la Pantalla LCD</i> .....	82
<b>Figura 43</b> <i>Escritura de Puertos Virtuales de Blynk</i> .....	83
<b>Figura 44</b> <i>Control Remoto del Relé</i> .....	83
<b>Figura 45</b> <i>Simulación del Circuito en Wokwi</i> .....	84

<b>Figura 46</b> <i>Plano Impreso y Tarjeta para Circuitos de Acople</i> .....	85
<b>Figura 47</b> <i>Caja de Almacenamiento del Dispositivo Electrónico</i> .....	86
<b>Figura 48</b> <i>Prototipo Finalizado</i> .....	88
<b>Figura 49</b> <i>Visualización Local de Variables</i> .....	89
<b>Figura 50</b> <i>Visualización Remota Mediante Blynk</i> .....	90
<b>Figura 51</b> <i>Monitoreo Local para Soporte Técnico</i> .....	91
<b>Figura 52</b> <i>Comparativo entre Valores Ideales y Reales para el Voltaje Medido</i> .....	93
<b>Figura 53</b> <i>Comparativo entre Valores Ideales y Reales para la Presión Medida</i> .....	93
<b>Figura 54</b> <i>Comparativo entre Valores Ideales y Reales para el Nivel de Combustible</i> .....	94
<b>Figura 55</b> <i>Comparativo entre Valores Ideales y Reales para la Temperatura Medida</i> .....	94

## Lista de Apéndices

<b>Apéndice A</b> <i>Código Fuente</i> .....	103
--	-----

## Introducción

La cuarta revolución industrial hace referencia a la digitalización del sector de la manufactura, anidada a ella se encuentran una serie de conceptos enmarcados en los avances tecnológicos que se presentan de la mano de la conectividad, la potencialización de sistemas computacionales y el tratamiento de grandes volúmenes de datos. (Peralta-Abarca et al., 2020). Por su parte, (Basco et al., 2018) considera que la industria 4.0 se enmarca en la convergencia de tecnologías preexistentes como la robótica, la fabricación aditiva, la realidad aumentada, el almacenamiento y tratamiento de datos y la simulación; sin embargo, es la combinación de estas tecnologías en el ámbito industrial lo que genera una disrupción significativa. (p. 26)

A nivel mundial, la industria busca la adopción de estas nuevas tecnologías con el objetivo de llegar a las denominadas fábricas inteligentes, donde los procesos se desarrollan de manera autónoma, a partir de la aplicación de tecnología avanzada y donde la intervención del ser humano se reduce a la supervisión de procesos (Lasi et al., 2014 citado por Peralta-Abarca et al., 2020); en Colombia, aunque el desarrollo se ha llevado a cabo de manera tardía, no se encuentra del todo descontextualizado. La transformación digital de las empresas trae consigo diferentes beneficios, entre ellos la optimización en tiempos de producción, la disminución de costos en mano de obra y un mayor cumplimiento de los objetivos empresariales propuestos. (Cárdenas Vidal, 2022)

Para las pymes la incursión en la Industria 4.0 ha propuesto importantes desafíos, debido a diversos factores como la falta de madurez tecnológica (Castillo-Vergara, 2024), la limitación de recursos para la implementación de nuevas tecnologías (Fernández Ladino et al., 2022) y el poco apoyo de entes gubernamentales en esta transición (Cárdenas Vidal, 2022), no obstante, aunque la tarea es compleja y el avance lento, los gerentes y propietarios de pymes le han

apostado a la capacitación como fuente principal de apropiación tecnológica. Según los estudios realizados por (Fernández Ladino et al., 2022) entre las temáticas más abordadas se encuentran el IoT, seguido del Big Data y la Computación en la Nube. (p. 10) Esta combinación de tecnologías impulsa la automatización y la transferencia de datos a través de la red global, creando sistemas inteligentes capaces de operar de manera autónoma y eficiente en la producción.

Los sistemas de control y monitoreo en tiempo real son importantes en múltiples áreas y aplicaciones, además de lograr monitorear la eficiencia de los sistemas, permite diagnosticar errores y fallas presentes en estos. (Escobar-Gallardo & Villazón, 2018) El monitoreo de variables físicas como la temperatura, el nivel de líquidos, la presión y el voltaje, entre otras, inmerso en un ecosistema IoT representa un papel fundamental en la tecnología industrial contemporánea, esto permite la interacción remota con las máquinas, obteniendo información en tiempo real y enviando señales de control a diferentes dispositivos de acuerdo con el entorno de trabajo.

El presente trabajo tiene como objetivo el desarrollo de un sistema de monitoreo remoto de variables críticas en un Grupo Electrónico, las magnitudes físicas seleccionadas para este monitoreo son esenciales para la detección temprana de anomalías en el funcionamiento del generador. El sistema propuesto recopila datos a través de sensores analógicos, los cuales son procesados y analizados por un dispositivo embebido programado específicamente para esta tarea. Posteriormente, la información es presentada en una aplicación web de uso libre, facilitando su visualización y análisis en tiempo real.

En este documento se detalla el desarrollo metodológico del proyecto, partiendo de un marco conceptual que fundamenta la investigación, resaltando las características principales de las tecnologías disponibles y justificando la selección de los dispositivos utilizados en el diseño.

A continuación, se describe el diseño de los distintos módulos que componen el sistema, así como la programación de las rutinas en el microcontrolador.

Finalmente, se presentan los resultados obtenidos tras la implementación del prototipo en un entorno controlado, de esta manera, se destacan las principales ventajas de la implementación del sistema y se identifican las áreas de mejora para su optimización. Las conclusiones reflejan el proceso de diseño e implementación llevado a cabo, subrayando la importancia de este tipo de sistemas para el control energético en diversos campos de aplicación.

## **Justificación**

Los Grupos Electrógenos suelen utilizarse principalmente para tres aplicaciones: como fuente primaria, donde actúan como la única fuente de energía disponible y operan de manera continua para abastecer la demanda energética; como fuente de cogeneración, en cuyo caso se utiliza para disminuir los picos de máxima demanda, optimizando el uso de la energía disponible y como fuente de emergencia, donde el grupo entra en operación únicamente cuando se produce una interrupción en la red de suministro principal, garantizando así la continuidad del fluido eléctrico. (Francisco et al., 2007)

Como sistemas de respaldo, los GE son componentes críticos en numerosos entornos, ya que se encargan de mantener el suministro eléctrico esencial, este tipo de sistemas son utilizados en distintos sectores de la industria para sostener su operatividad de manera continua. En este tipo de sistemas la falta de mantenimiento periódico puede llevar a fallas imprevistas, agotando la vida útil del equipo y generando pérdidas económicas significativas. De acuerdo a lo que menciona Power Rent (2020) en su blog, las fallas más importantes en los grupos electrógenos suelen diagnosticarse en el sitio, lo que resulta en la necesidad de múltiples visitas por parte del personal técnico para identificar y reportar las incidencias; entre las fallas más recurrentes se encuentran el agotamiento de la batería, falta de combustible, bajos niveles de refrigerante, altas temperaturas y sobrecargas eléctricas, que pueden provocar bloqueos en el sistema y el apagado del equipo.

Mantener el suministro eléctrico constante y confiable se convierte cada día en una necesidad latente para empresas de diferentes sectores económicos, es así como, bancos, estaciones de gasolina, hospitales, hoteles, conjuntos residenciales, empresas de TI, entre otros,

requieren que el suministro eléctrico sea confiable e ininterrumpido, por lo cual la disponibilidad e integridad del grupo electrógeno son factores críticos.

Con la implementación de un sistema de monitoreo de variables en tiempo real para grupos electrógenos, se garantiza la supervisión continua de este tipo de equipos, generando confianza en los sistemas eléctricos de empresas industriales y de servicios, este proyecto permitirá la integración de sensores de tipo industrial con sensores de bajo costo, asegurando su compatibilidad a través del sistema desarrollado; el microcontrolador centralizado es el encargado de integrar y gestionar las diferentes variables monitoreadas, proporcionando una solución eficiente y adaptable para diversas aplicaciones.

## **Planteamiento del Problema**

Los grupos electrógenos son componentes esenciales en múltiples sectores económicos, incluyendo los comerciales, industriales, de servicios y de salud. Su función principal es proporcionar un respaldo eléctrico confiable durante interrupciones del suministro de energía, garantizando así la continuidad de operaciones críticas y protegiendo tanto la seguridad humana como los intereses económicos. Sin embargo, a pesar de su importancia, la falta de monitoreo constante y mantenimiento adecuado de estos equipos se ha identificado como una de las causas más comunes de fallas en su funcionamiento, tal como lo indica. (Gestor, 2021)

En sectores como la salud, donde el suministro ininterrumpido de energía es vital para la integridad de los pacientes y la operatividad de los servicios médicos, las fallas en los grupos electrógenos pueden tener consecuencias catastróficas. Un fallo en estos sistemas puede provocar interrupciones en áreas críticas como quirófanos, unidades de cuidados intensivos y otros departamentos esenciales. A pesar de contar con departamentos dedicados al mantenimiento, muchas veces estos enfrentan desafíos para proporcionar una cobertura de monitoreo constante, lo que aumenta el riesgo de fallos inesperados y pone en peligro la seguridad de los pacientes y la operatividad del hospital.

En el sector industrial, la necesidad de mantener una operación continua es igualmente crítica. Empresas que dependen de grupos electrógenos para evitar interrupciones en el suministro eléctrico no pueden permitirse fallos, ya que esto podría resultar en pérdidas económicas significativas, daños a equipos sensibles y, en casos extremos, poner en riesgo la seguridad de los trabajadores. A pesar de la importancia de estos equipos, muchas veces el monitoreo es insuficiente, especialmente en empresas que operan en entornos remotos o con recursos limitados para la supervisión continua.

Asimismo, en otros sectores como el financiero y el de combustibles, la confiabilidad del suministro eléctrico es fundamental. En bancos, la seguridad es prioritaria, y la falla de los sistemas de respaldo puede impedir la grabación o reporte de incidentes críticos, aumentando los riesgos de seguridad y potenciales pérdidas financieras. De manera similar, en las estaciones de servicio de combustible, la continuidad del suministro eléctrico es vital para mantener las operaciones, y las interrupciones pueden afectar gravemente su economía y operación. En muchas estaciones de servicio en Colombia, especialmente en áreas remotas, el monitoreo de los grupos electrógenos es deficiente, incrementando la vulnerabilidad ante cortes eléctricos.

La problemática es clara y latente, la necesidad de un sistema que permita monitorear de manera remota y continua las variables críticas de los grupos electrógenos, asegurando su operatividad y reduciendo los riesgos asociados a fallas inesperadas que afectan diferentes sectores de la industria. A pesar de que existen diversas tecnologías y soluciones para el monitoreo de equipos, muchas de estas son costosas o no se adaptan fácilmente a las necesidades específicas de cada sector o localización geográfica.

Este trabajo de grado busca responder a la pregunta: ¿Cómo se puede implementar un sistema de monitoreo remoto y permanente para las variables críticas de un grupo electrógeno, que sea eficiente, económico y adaptable a diferentes entornos y necesidades?

Este proyecto propone desarrollar una solución de monitoreo remoto para grupos electrógenos que integre sensores de bajo costo con tecnologías IoT para recopilar datos en tiempo real, procesarlos y presentarlos en una plataforma accesible para los usuarios. De esta manera, se espera mejorar la eficiencia en el mantenimiento y operación de los grupos electrógenos, asegurando su funcionalidad y reduciendo significativamente el riesgo de fallos en aplicaciones críticas.

## **Objetivos**

### **Objetivo General**

Desarrollar un sistema electrónico para el monitoreo de variables en un grupo electrógeno, mediante la obtención de datos en tiempo real y el visualización de información remota usando estándares de comunicación WI-FI.

### **Objetivos Específicos**

Seleccionar las variables críticas para el monitoreo de un Grupo Electrónico que permitan aumentar los niveles de confiabilidad en estos equipos.

Realizar el diseño de los circuitos electrónicos para la obtención de señales análogas, la centralización de datos en la tarjeta de desarrollo y el envío de señales de visualización y control para el sistema propuesto.

Desarrollar los códigos de programación requeridos para la lectura, análisis y monitoreo de variables para la tarjeta de desarrollo ESP32 mediante la plataforma Arduino.

Integrar el Aplicativo Web para monitoreo remoto al sistema de adquisición de datos, configurando una interfaz gráfica intuitiva para la visualización y lectura de variables.

Validar el funcionamiento del sistema desarrollado con un grupo electrógeno de prueba simulando condiciones operativas reales.

### **Alcance**

El presente proyecto abarca el diseño, desarrollo e implementación de un sistema electrónico para el monitoreo de cuatro variables críticas para un grupo electrógeno, voltaje de la batería, temperatura del motor, presión de gases y compresión interna del motor y nivel de combustible; mediante el uso de sensores y circuitos de acople para adquisición de datos en tiempo real y visualización remota a través de estándares de comunicación Wi-Fi.

El sistema integra tanto sensores de tipo industrial como sensores de bajo costo, centralizando las señales en un microcontrolador ESP32 programado mediante la plataforma Arduino. Se desarrollaron los circuitos electrónicos necesarios para la adquisición de señales analógicas, así como los códigos de programación que permitirán el procesamiento, análisis y monitoreo de las variables seleccionadas. Además, se llevará a cabo la configuración de un aplicativo web de uso gratuito, que permitirá la visualización remota de las variables monitoreadas, facilitando una interfaz gráfica intuitiva para el usuario final.

El sistema será validado mediante pruebas que simulen el funcionamiento de un grupo electrógeno bajo condiciones operativas estándar, asegurando su funcionalidad y efectividad en aplicaciones reales.

Este proyecto se limita al desarrollo de un prototipo funcional, con la posibilidad de futuras mejoras y adaptaciones para su implementación en un entorno industrial real.

## **Marco Conceptual**

### **Generación Distribuida**

Las instalaciones tradicionales de generación, transmisión y distribución de energía enfrentan desafíos en sostenibilidad, eficiencia y escalabilidad. La creciente preocupación por el medio ambiente ha impulsado el desarrollo de tecnologías de generación de energía a pequeña escala, enfocándose en fuentes renovables y limpias como la energía fotovoltaica y energía eólica; la primera convierte la luz solar en electricidad, mientras que la segunda aprovecha la fuerza del viento. (Francisco et al., 2007)

En la actualidad, se promueve la integración energética, que busca desarrollar infraestructuras que optimicen el uso de diferentes fuentes de energía y fomenten tecnologías alternativas mediante políticas regulatorias. La generación distribuida (GD) es una de estas integraciones, que utiliza unidades modulares de generación eléctrica instaladas de manera aislada o interconectadas, es una estrategia destacada para diversificar la matriz energética, reducir el uso de combustibles fósiles y minimizar el impacto ambiental. La GD necesita una serie de adaptaciones para lograr un acople efectivo con la red eléctrica, aunque ofrece beneficios como la reducción de pérdidas de energía, mejoras en el perfil de tensión y descongestión en las líneas de distribución es indispensable identificar que una mala ubicación o dimensionamiento puede tener efectos negativos en la red. (Rojas Suarez et al., 2021)

### **Grupo Electrónico**

Dentro de la Generación Distribuida, existe un pequeño grupo que se enfoca en los procesos industriales. Los grupos electrónicos (GE) se utilizan para suministrar energía de manera continua cuando hay interrupciones en la red pública, en el contexto industrial, la

generación distribuida emplea GE de baja y media potencia para proporcionar energía de forma descentralizada. (Fernandez-Morales & Duarte, 2015) Muchos sistemas de generación eléctrica no centralizados carecen de monitoreo y visualización de datos clave como temperatura, corriente, voltaje y presión, lo que dificulta el mantenimiento y la detección de fallas.

Los GE constan de un motor y un generador eléctrico acoplados, y están equipados con diversos sistemas para operación, control y monitoreo. Funcionan según la ley de Faraday, que requiere movimiento entre un campo magnético y un conductor para generar electricidad. Los GE se dividen en gama baja (menos de 50 KW) para viviendas y pequeñas empresas, y gama alta (más de 50 KW) para industrias, con potencias que pueden llegar hasta 3 MW. Aunque muchos generadores modernos tienen módulos de visualización y adquisición de datos, estos suelen ser costosos y limitados a alarmas.

Un generador eléctrico diésel es un sistema autónomo que emplea un motor diésel para producir electricidad, basándose en el principio de conversión de energía mecánica en energía eléctrica. El motor convierte la energía química del combustible en energía mecánica, que es utilizada para generar electricidad a través de un alternador. (EMSA, s. f.)

El funcionamiento del grupo electrógeno diésel se divide en cuatro etapas clave: la primera consiste en el suministro del combustible diésel al motor a través de un sistema de alimentación, que lo filtra y bombea para ser inyectado al motor; la segunda etapa es la combustión, donde el combustible se quema dentro del motor generando calor y presión, controlados mediante un sistema de encendido y regulación de aire; en la tercera etapa, la presión y el calor movieron el pistón, cuyo movimiento se transmite al cigüeñal, que a su vez acciona el alternador. Finalmente, el alternador genera electricidad a través de inducción electromagnética, la cual puede ser utilizada para alimentar cargas o almacenada en baterías.

Este trabajo propone desarrollar un dispositivo accesible para monitorear y almacenar datos de GE, facilitando el mantenimiento preventivo y correctivo y la estimación del funcionamiento de los generadores. (Larco-Montenegro, 2019)

### ***Mantenimiento Preventivo y Correctivo***

Debido a la naturaleza y funcionalidad de los grupos electrógenos, el mantenimiento tanto preventivo como correctivo es crucial para este tipo de equipos. Se entiende que el mantenimiento es un conjunto de recursos (físicos, financieros, humanos, tecnológicos) destinado a mejorar la eficiencia, reducir paros, aumentar la confiabilidad y garantizar la seguridad y rentabilidad del sistema. Para un grupo electrógeno, contar con un plan de mantenimiento se traduce en la reducción de fallas inesperadas, más aún cuando los dispositivos están altamente automatizados. El mantenimiento debe enfocarse en la predicción, más que en la corrección, de esta manera, se logra que los sistemas mantengan la capacidad de generación, disponibilidad, respaldo, continuidad, flexibilidad y seguridad que requiere un GE. Existen varios tipos de mantenimiento, para estos sistemas es muy importante contar con el mantenimiento predictivo, que busca anticipar fallas basándose en factores externos y cálculos matemáticos relacionados con las horas de operación.

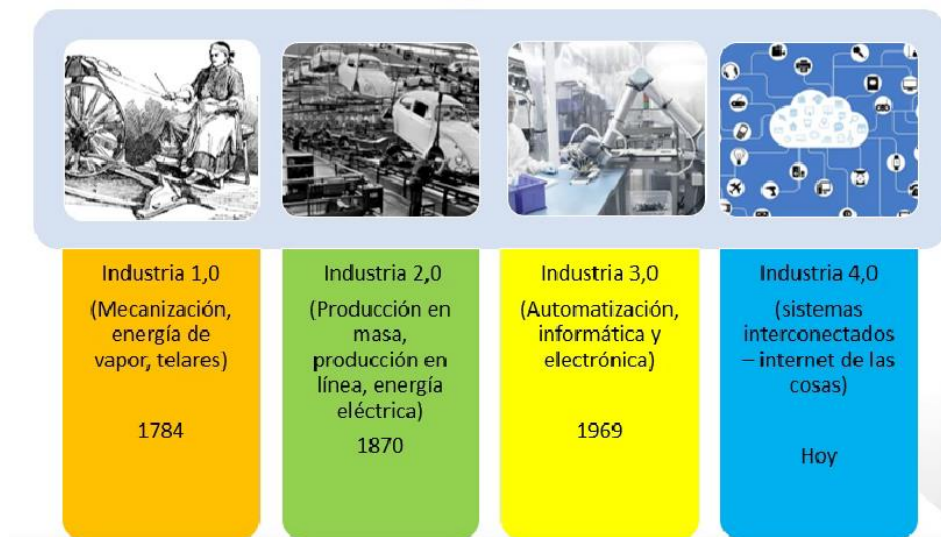
### **Industria 4.0**

El termino Industria 4.0 hace referencia a una nueva fase de la revolución industrial, caracterizada por la digitalización de sistemas y procesos y su interconexión mediante el Internet de las Cosas (IoT); busca la transformación de las empresas en organizaciones inteligentes para lograr procesos flexibles e individualizados mediante una mayor automatización, conectividad y globalización. (Ovalle Castro, 2019)

Joyanes (2017) se refiere a la Cuarta Revolución Industrial como una evolución significativa en los procesos históricos de las revoluciones industriales anteriores que ha sido caracterizada por la convergencia e integración de diversas tecnologías avanzadas, dando lugar a la creación de sistemas ciberfísicos. Estos sistemas tienen la capacidad de recopilar y procesar grandes volúmenes de información en tiempo real, tomar decisiones de manera inteligente, y ejecutar tareas de forma autónoma en entornos dinámicos y cambiantes. Este nuevo paradigma transforma la manera en que se diseñan, producen y operan los sistemas industriales, llevando la automatización y la interconexión a niveles sin precedentes. (pp. 1-5)

### Figura 1

#### *Las Cuatro Etapas de la Revolución Industrial*



*Nota.* Esquema evolutivo de tecnificación industrial, se evidencia un progresivo cambio tecnológico y en la última etapa la interconexión de tecnologías preexistentes. Adaptado de La Industria 4.0. Tecnología en Gestión Industrial. Ovalle, R. (2019).

El concepto de esta cuarta revolución se aplica a la industria con las llamadas fábricas inteligentes, donde los procesos productivos se llevan a cabo de manera autónoma, mediante el uso de tecnología avanzada, y la intervención humana se limita a la supervisión de procesos. (Peralta-Abarca et al., 2020) Este modelo se ha adaptado satisfactoriamente a los sistemas productivos actuales, entre sus principales aplicaciones se destacan:

### ***Cadenas de Suministro Inteligentes***

Una cadena de suministro está organizada por proveedores, facilitadores, logística, fabricación y canales de distribución. En la nueva revolución industrial, estos procesos se automatizan, permitiendo realizar pedidos con requerimientos específicos a los proveedores en tiempo real o al cumplirse la orden programada, y realizando un seguimiento en tiempo real del proceso de fabricación, la aplicación del Blockchain permite el control del producto en cada eslabón de la cadena. Estos sistemas gestión de la cadena de suministro se componen de sitios web alojados en servidores de la compañía donde se encuentran alojados los datos clave en relación con los procesos para cada etapa de la cadena de suministro. (Belalcazar Diaz & Quintero Valero, s. f.)

### ***Monitoreo de Equipos***

La recolección de datos proporciona información valiosa sobre diversos aspectos operativos, como la vida útil de los equipos, el nivel de carga, la detección de fallos, y las condiciones ambientales en las que operan. Al disponer de esta información, es posible monitorear continuamente el estado de las máquinas o equipos, lo que permite anticipar posibles fallos y optimizar el mantenimiento preventivo. En cuanto a la autogestión y la toma de decisiones automatizadas, los sistemas pueden ejecutar órdenes en tiempo real basadas en los

datos recopilados. (Peralta-Abarca et al., 2020) Por ejemplo, un tanque de combustible equipado con sensores y algoritmos de control podría ordenar un autoabastecimiento automáticamente cuando su nivel de combustible descienda por debajo del 30%. Esto no solo asegura la continuidad operativa, sino que también optimiza los recursos y reduce los costos operativos al evitar interrupciones inesperadas. Además, al integrar estos sistemas con plataformas de análisis de datos más avanzadas, se pueden realizar diagnósticos predictivos que anticipen necesidades futuras, mejorando así la eficiencia y la productividad en la gestión de activos industriales.

### ***Colaboración Hombre-Maquina:***

En la nueva industria se habla de sistemas ciberfísicos, que no son más que la integración de un de sistemas para el monitoreo y control de procesos físicos realizada a través de sistemas embebidos (Chandy, 2010); su peculiaridad radica en la integración del trabajo colaborativo con operadores humanos. Es así como la robótica ha ido migrando hacia la creación de robots colaborativos, compuestos por sensores de diversos tipos que permiten que operadores humanos intervengan en procesos a la par con robots; algunas decisiones no dependen de los sistemas automatizados y es importante que expertos puedan también decidir dentro del ecosistema. Es en este punto donde la interacción hombre-máquina se hace imprescindible para la toma de decisiones.

Dentro de los pilares tecnológicos de la Industria 4.0 se encuentran: la Fabricación aditiva, el Big Data y análisis de datos, la Computación en la nube, la Simulación, la Ciberseguridad, la Realidad aumentada, los Sistemas de integración, la Robótica y el Internet de las Cosas; si bien, estas tecnologías existían con anterioridad, es su integración lo que genera una disrupción significativa para la industria actual.

**Figura 2***Pilares de la Industria 4.0*

*Nota.* Industria 4.0 y la sinergia de las tecnologías emergentes. Tomado de Industria 4.0, Fabricando el Futuro (pp 29). por Basco, A.I, Coats, G y Garnero, P. (2018).

<https://play.google.com/books/reader?id=geiGDwAAQBAJ>

Adicional a los pilares mencionados anteriormente, para las diferentes aplicaciones industriales, se integran otros sistemas tecnológicos que complementan las aplicaciones propias en cada campo, entre ellos se pueden encontrar: drones, sensores inteligentes, plataformas electrónicas abiertas, sistemas de localización, sistemas de autoidentificación y Blockchain, entre otros. (Basco et al., 2018)

### *Internet de las Cosas*

La electrónica y las comunicaciones han transformado objetos y personas, dotándolos de una nueva dimensión digital mediante la incorporación de sensores avanzados que permiten la comunicación autónoma entre dispositivos y sistemas de información. Estos sensores embebidos recopilan y transmiten grandes cantidades de datos sin necesidad de intervención humana, lo que posibilita la monitorización y operación en tiempo real de diversos procesos. Esta capacidad de comunicación y análisis instantáneo no solo facilita la toma de decisiones informadas, sino que también mejora significativamente la eficiencia operativa, al optimizar recursos y reducir tiempos de respuesta en entornos industriales. (nodoCAST, 2016)

El Internet de las Cosas (IoT) se refiere a la interconexión de dispositivos físicos como sensores, actuadores y dispositivos de red mediante un sistema embebido y un software integrado para recopilar y compartir datos a través de internet, permitiendo su monitoreo, control y automatización remotos. En la industria se integran diferentes actores como son los dispositivos de medición y control, el monitoreo local y remoto y la interacción hombre – máquina, a lo que se llama IoT Industrial (IIoT).

A nivel de hardware, una solución de Internet de las Cosas (IoT) se compone de varios elementos fundamentales que trabajan en conjunto para cumplir su propósito. Primero, está el conjunto de sensores, que actúan como dispositivos de entrada, capturando datos del entorno o del sistema en el que están implementados. Luego, los actuadores funcionan como dispositivos de salida, realizando acciones físicas o mecánicas basadas en las instrucciones que reciben. El microcontrolador es el componente central que recibe los datos de los sensores, los analiza y, en función de los resultados, decide cuáles datos deben enviarse a la nube para almacenamiento o procesamiento adicional. Dependiendo de la aplicación específica, el microcontrolador también

controla directamente la operatividad de los actuadores, ajustando su comportamiento para garantizar que el sistema responda de manera óptima a las condiciones detectadas. (Quiñones Muñoz, 2019)

### Figura 3

#### *IoT Industrial*



*Nota.* Integración del IoT en la Industria. Tomado de TyN Magazine, Industrial Internet Consortium publica un marco de referencia para la IIoT, por Staff, (2016).

<https://tynmagazine.com/industrial-internet-consortium-publica-un-marco-de-referencia-para-la-iiot/>

**Sensores.** Los sensores son dispositivos diseñados con elementos sensibles capaces de detectar cambios en diferentes magnitudes físicas, químicas o biológicas, transformando esas variaciones en señales eléctricas que pueden ser interpretadas por sistemas de medición y control. Estos dispositivos tienen la capacidad de recibir información del entorno y convertirla en una señal eléctrica cuantificable, permitiendo su monitoreo y manipulación. La función principal de los sensores es detectar cambios en el entorno del objeto de medición y enviar información en tiempo real, facilitando una respuesta rápida y precisa. (Quiñones Muñoz, 2019)

Dependiendo de la variable que se quiera medir, existen diversos tipos de sensores, entre ellos se encuentra los sensores de contacto, ópticos, térmicos, de humedad, magnéticos, infrarrojos, entre otros; cada tipo de sensor está diseñado para detectar una magnitud específica, lo que los hace indispensables en una amplia gama de aplicaciones industriales y tecnológicas. La selección de un sensor para una aplicación específica debe realizarse no basados en la magnitud a medir, adicional a ello es necesario evaluar sus características de funcionamiento, entre ellas se encuentran rango, amplitud, exactitud, precisión, sensibilidad, linealidad, histéresis, resolución, reproductibilidad, repetibilidad y tiempo de respuesta. En la Tabla 1 se relacionan las características de un sensor y su definición.

**Tabla 1**

*Características de los Sensores*

Características	Definición
Rango	Espacio entre el valor mínimo y valor máximo que puede medir el sensor.
Amplitud	Diferencia entre los valores máximos y mínimos de entrada.
Exactitud	Diferencia entre el valor medido y el valor real.
Precisión	Cercanía de la medida entregada a entradas del mismo valor.
Sensibilidad	Cambio mínimo de entrada requerido para producir un cambio en la salida.
Linealidad	Máxima desviación entre los valores medidos de un sensor de la curva ideal.
Histéresis	Es la diferencia en la salida cuando la entrada varía de dos maneras, aumentando y disminuyendo.
Resolución	Cambio mínimo en la entrada que puede ser detectado por el sensor.
Reproducibilidad	Se define como la capacidad del sensor de producir la misma salida cuando se aplica la misma entrada.

---

Repetibilidad	Capacidad del sensor de producir la misma salida cada vez que se aplica la misma entrada y todas las condiciones físicas y de medición se mantienen iguales, incluyendo el operador, el instrumento, las condiciones ambientales, etc.
Tiempo de respuesta	Tiempo requerido para que la salida del sensor cambie del estado anterior a un valor final establecido.

---

*Nota.* La tabla contiene la definición de las principales características de los sensores. Adaptado de Sensores y Actuadores, Ramírez L. Jiménez, G. y Carreño J. (2014)

[https://books.google.com.co/books?id=wMm3BgAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.co/books?id=wMm3BgAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

**Actuadores.** Ramírez et al. (2014) define a los actuadores como dispositivos diseñados para convertir diferentes formas de energía, como la hidráulica, neumática o eléctrica, en movimiento mecánico o en la activación de un proceso, con el propósito de generar un efecto sobre un elemento externo. Estos dispositivos reciben órdenes de un controlador, regulador o plataforma de control y, en función de dichas órdenes, activa un elemento final de control. (p. 25) Por ejemplo, un actuador puede abrir o cerrar una válvula, poner en marcha un ventilador, o activar un sistema de riego. Los actuadores son esenciales en sistemas automatizados, ya que permiten la ejecución de acciones precisas basadas en las señales recibidas, contribuyendo así a la eficiencia y eficacia de los procesos industriales.

Los actuadores son mecanismos esenciales que aseguran el funcionamiento adecuado de los sistemas de control, ya que permiten la ejecución de acciones específicas sobre las variables de un proceso. La selección del tipo de actuador adecuado depende de las características

específicas de la solución IoT en cuestión, y de cómo este debe integrarse y adaptarse a los diferentes elementos de control. Para la selección de un actuador se debe tener en cuenta el tipo de sistema que se desea controlar, evaluando sus capacidades, cualidades y modos de funcionamiento. Entre las características a evaluar se encuentran: alta fiabilidad, facilidad de uso, baja necesidad de mantenimiento, seguridad y precisión en el posicionamiento, durabilidad, y capacidad de respuesta en ambientes hostiles, entre otros. (Quiñones Muñoz, 2019)

**Dispositivos Embebidos o Microcontroladores.** Un sistema embebido es un sistema diseñado para llevar a cabo una función específica mediante el control de un computador integrado, como un microcontrolador o un microprocesador. El término "embebido" hace referencia a que este computador está incorporado dentro del sistema general, permaneciendo oculto a la vista y formando parte de un conjunto mayor. En estos sistemas, el hardware se encuentra estrechamente integrado con software especializado, lo que permite la ejecución de tareas específicas de manera eficiente y optimizada en el entorno en el que operan. (Pérez A, 2009)

**Periféricos.** Se considera periférico a cualquier unidad o dispositivo de hardware que permite al controlador comunicarse con el mundo exterior o con sistemas de almacenamiento de información, actuando como una memoria auxiliar de la memoria principal. En términos generales, un periférico es un dispositivo auxiliar e independiente conectado al procesador central, como puede ser el caso de un ESP32 u otro entorno de desarrollo. Estos periféricos son fundamentales para ampliar la funcionalidad del sistema embebido, proporcionando interfaces de entrada, salida y almacenamiento.

Actualmente, algunos ejemplos comunes de dispositivos periféricos incluyen pantallas LCD, teclados, memorias externas, indicadores luminosos, pantallas táctiles, displays numéricos,

audífonos, micrófonos, entre otros. Estos dispositivos desempeñan roles esenciales en diversas aplicaciones, permitiendo la interacción del sistema con los usuarios o facilitando el almacenamiento y procesamiento de datos adicionales. (Quiñones Muñoz, 2019)

### ***Comunicación Machine to Machine – M2M***

Las siglas M2M, que significan *Machine to Machine*, se refieren a la comunicación directa entre máquinas, como ordenadores, servidores y dispositivos móviles. (Citelia, 2023) El concepto de *Machine to Machine* forma parte integral del Internet de las Cosas (IoT), en el contexto industrial, también se habla del IIoT, o Internet Industrial de las Cosas, que se centra en la conexión de objetos dentro de la industria.

Las características que destacan la comunicación máquina a máquina están la automatización de tareas y procesos, eficiencia operativa, reducción de intervención humana, disminución de costos, monitoreo remoto, recopilación y análisis de grandes volúmenes de datos para proporcionar información valiosa, otra de las grandes ventajas que tiene, es su aplicación en diversas áreas como el almacenamiento, robótica y transporte. En las redes M2M se utilizan diferentes tipos de dispositivos para permitir la comunicación y transferencia de datos entre máquinas, como sensores que recopilan datos del entorno que son enviados a otros dispositivos para su procesamiento y análisis; actuadores que realizan una acción física en respuesta a las instrucciones o datos recibidos; dispositivos de comunicación encargados de transmitir y recibir datos; dispositivos de control para el monitoreo, y la gestión y control de los sistemas y procesos; dispositivos que almacenan y procesan datos generados por de red M2M y dispositivos finales que interactúan directamente con el usuario o realizan acciones en función de los datos recibidos

como pantallas, paneles de control, dispositivos móviles; aplicaciones o sistemas utilizados para ver y actuar sobre los datos generados.

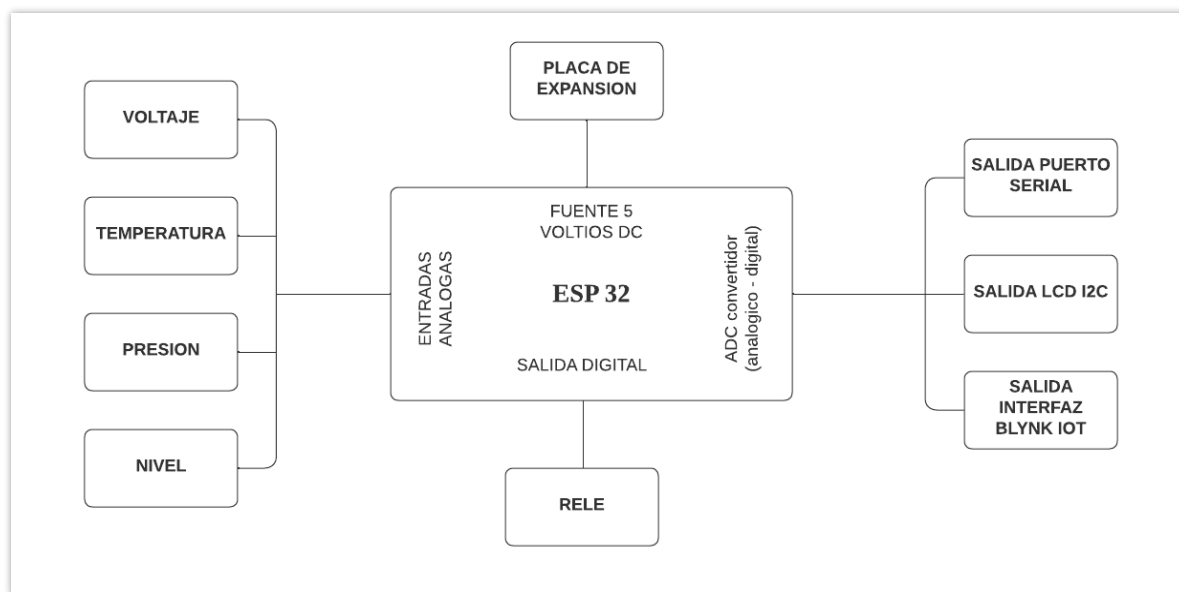
## **Metodología**

Para el desarrollo de este proyecto aplicado se siguió una metodología estructurada que incluyó varias etapas clave: análisis del problema, selección de componentes, diseño del sistema, integración y pruebas. A continuación, se describen cada una de estas etapas en detalle:

### **Análisis del Problema y Definición de Requisitos**

En esta primera etapa, se realizó un análisis profundo del problema, partiendo de la necesidad de monitorear de manera remota las variables críticas de un grupo electrógeno para evitar fallas inesperadas y mejorar la eficiencia en su operación. Se definieron las variables físicas a monitorear: voltaje de la batería, temperatura del motor, presión de gases, compresión interna del motor y nivel de combustible; los requisitos funcionales del sistema como la visualización local de los parámetros medidos y el control de encendido de la planta; las características técnicas que debe cumplir el sistema, como la capacidad monitoreo remoto en tiempo real a través de una plataforma web y la posibilidad de operar en diferentes entornos.

El sistema se basa en un dispositivo embebido, que funciona como el cerebro, encargado de recopilar datos de los sensores y controlar la salida del relé. Las variables físicas son obtenidas mediante cuatro sensores analógicos, cuyos datos son digitalizados mediante un conversor análogo – digital (ADC). El dispositivo embebido se encuentra, a su vez, conectado física y virtualmente a distintos dispositivos de visualización entre los que se encuentran una pantalla LCD y una plataforma de monitoreo IoT.

**Figura 4***Diagrama de Bloques Sistema*

*Nota.* Diagrama de bloques, detalla la interconexión de los diferentes componentes del sistema.

**Selección de Componentes y Tecnologías**

Basado en los requisitos identificados, se seleccionaron los componentes y tecnologías adecuadas para el proyecto. Esto incluyó la elección de sensores específicos para la medición de cada variable crítica, un microcontrolador adecuado para procesar y gestionar los datos y una plataforma web de uso libre para la visualización y análisis de datos en tiempo real. Para la selección de cada parte constitutiva del sistema se tuvieron en cuenta criterios como el costo, la disponibilidad, la compatibilidad con las tecnologías existentes y la facilidad de integración.

### *Tarjeta de Desarrollo*

En cuanto a dispositivos embebidos, en el mercado se encuentran diferentes tecnologías que cumplen con las especificaciones requeridas para el diseño del sistema, a continuación, se presentan las opciones que se evaluaron en el proceso de selección de la tarjeta de desarrollo.

**Arduino.** Puede considerarse como la plataforma más popular para programación de aplicaciones de robótica e IoT. Arduino es una plataforma de hardware y software de código abierto, con diferentes placas de desarrollo que se ajustan al tamaño y las necesidades de cualquier proyecto. Entre las principales características de los módulos de Arduino están la incorporación de microcontroladores ATmega de diferentes referencias, pines análogos y digitales con interfaces UART, PWM, SPI, I2C e I2S variadas según cada modelo.

### **Figura 5**

#### *Tarjetas de Desarrollo Arduino*

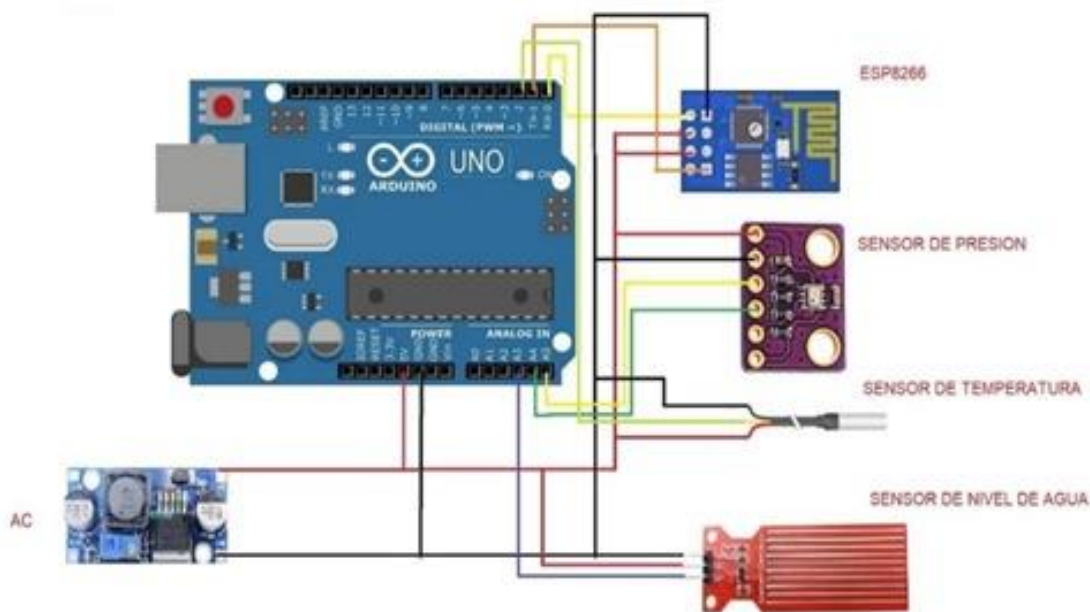


*Nota.* Referencias más representativas de Arduino, entre ellas Arduino nano, Arduino UNO, Arduino Mega y Arduino Leonardo.

El primer prototipo fue implementado haciendo uso de un Arduino UNO, una de las referencias más empleadas en proyectos de pequeño y mediano alcance, debido a su versatilidad, disponibilidad de puerto y capacidad de procesamiento; dadas sus características se hizo necesaria la implementación de un adaptador WiFi que permitiera establecer conectividad a internet, para esta implementación se emplearon sensores de bajo costo compatibles con la tarjeta de desarrollo. Si bien, el diseño permitió la lectura de variables y el monitoreo local y remoto de los datos obtenidos, se identificaron algunas desventajas como la necesidad de módulos adicionales y la incompatibilidad con sensores industriales, el circuito de prueba implementado con Arduino UNO, integró un potenciómetro como variador de voltaje, un sensor de lluvia para determinar el nivel de líquidos y un módulo de temperatura LM35.

### Figura 6

*Prototipo Implementado con Arduino*



*Nota.* Circuito de prueba implementado con Arduino UNO

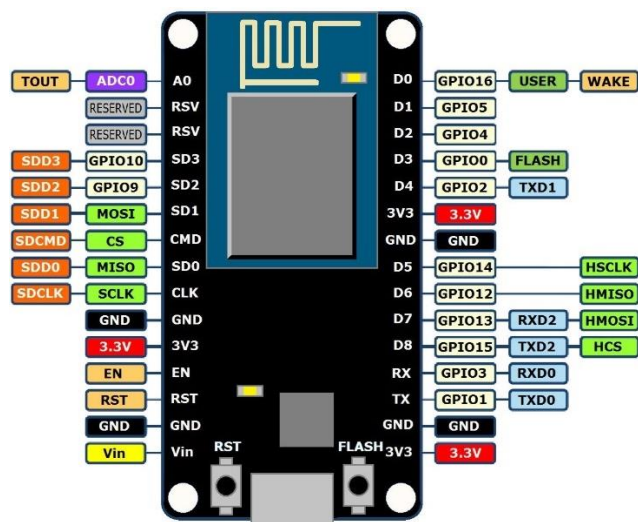
Debido a las limitaciones identificadas, la placa de desarrollo Arduino mostró no ser la adecuada para satisfacer necesidades específicas específicas del sistema, lo que llevó a indagar otras alternativas.

**ESP8266 NodeMCU.** Este dispositivo dispone de potentes capacidades de procesamiento y almacenamiento, lo que facilita la integración con sensores externos conectados a través de sus pines de entrada/salida. Posee 17 pines GPIO, que pueden ser asignados a diversas tareas periféricas: un canal ADC de 10 bits, dos interfaces UART, cuatro pines PWM, interfaces SPI, I2C e I2S; en términos de memoria cuenta con 32 KB de RAM, 80 KB de DRAM y 200 KB de memoria Flash; integra un regulador de 3.3V y 600mA disponible en sus 8 pines de alimentación (GND, 3.3V). El ESP8266 es un dispositivo ampliamente usado en aplicaciones de IoT, recopilando los datos arrojados por los sensores mediante firmware y transfiriéndolos a un servidor local o remoto (IoT), el chip ESP8266, viene equipado con tarjeta de red Wi-Fi integrada; en cuanto a entornos de programación muestra bastante versatilidad admitiendo plataformas como Arduino y Python. (Parihar, 2019)

Para esta tarjeta de desarrollo no se llevó a cabo la implementación de un circuito de prueba, ya que fue descartada por la inoponibilidad de puertos análogos, ya que solo cuenta con un puerto de este tipo y el diseño propuesto requiere un mayor número.

**Figura 7**

*Configuración de Pines ESP8266 NodeMCU*



*Nota.* Identificación de pines de la tarjeta de desarrollo ESP8266. Adaptado de Electronic Lab, Módulo Wifi NodeMCU ESP8266 Esp12e WiFi CP2102. (2024)

<https://electronilab.co/tienda/modulo-wifi-nodemcu-v3-esp8266-esp12e-wifi-cp2102/>

**ESP32.** Del mismo fabricante de su predecesora, la ESP32 representa una versión mejorada de la ESP8266. Con funcionalidades enriquecidas para el desarrollo de aplicaciones IoT gracias a una poderosa combinación de características y capacidades que incluyen conectividad Wi-Fi y Bluetooth integrada, una gran cantidad de pines de entrada/salida de uso general (GPIO) que facilitan la conexión y el control de dispositivos y sensores externos, se puede programar utilizando una variedad de lenguajes y marcos de desarrollo, además, ESP-IDF (Espressif IoT Development Framework) proporciona un conjunto completo de bibliotecas y herramientas específicas para el desarrollo. (Hercog et al., 2023)





puertos USB 3.0 y 2.0, Ethernet Gigabit, dos puertos micro-HDMI con soporte de hasta 4k a 60 Hz, ranura para tarjeta MicroSD y alimentación USB tipo C de 5,1 V 3 A. (Escalas Rodríguez, 2015)

### **Figura 10**

*Tarjeta de Desarrollo Raspberry Pi*



*Nota.* Tarjeta de desarrollo Raspberry Pi. Adaptado de Diseño y desarrollo de un prototipo de riego automático controlado con Raspberry Pi y Arduino. Escalas Rodríguez, G. (2014)

<https://upcommons.upc.edu/handle/2099.1/25074>

Aunque las funcionalidades que presenta la Raspberry Pi son bastantes, tiene dos grandes limitantes para ser utilizada en este proyecto específico, el primero de ellos se relaciona con la disposición de pines de entrada ya que solo cuenta con pines digitales y el segundo es referente a la falta de una tarjeta de red inalámbrica; adicional a ello, por ser un dispositivo más especializado, su costo de es considerablemente superior al de otras opciones que se encuentran en el mercado, por esta razón se descartó su uso y no se llevó a cabo la implementación de un prototipo de prueba.

La selección de la tarjeta de desarrollo se basó en una serie de criterios, que incluyen su potencia de procesamiento, capacidad de memoria, compatibilidad con interfaces analógicas, y soporte para conectividad inalámbrica y costos; estas características son fundamentales para garantizar un monitoreo preciso y confiable de las variables analógicas del grupo electrógeno, así como para garantizar la construcción de un dispositivo asequible al mercado potencial. La Tabla 2 presenta un comparativo de los principales criterios evaluados para las diferentes tarjetas de desarrollo evaluadas.

Tras una evaluación técnica de las diferentes tarjetas de desarrollo se determinó que la ESP32 es el dispositivo idóneo para la puesta en marcha del proyecto, adicional a las capacidades de hardware (Interfaces digitales, ADC, tarjeta WiFi), la ESP32 ofrece una flexibilidad significativa en términos de programación y desarrollo, lo que facilita la implementación de funcionalidades adicionales y la integración con otros dispositivos o sistemas a futuro. Esta tarjeta actuará como el núcleo del sistema, encargado de realizar la lectura, interpretación y transmisión de datos a través de la red global, para su posterior visualización en la aplicación designada.

**Tabla 2**

*Comparativo entre las Tarjetas de Desarrollo Arduino, NodeMCU, ESP8266 y Raspberry*

Características	Arduino	NodeMCU	ESP32	Raspberry Pi
Arquitectura	ATMega328P.	Single-core 32-bit	Dual-core Xtensa 32-bit	ARM Cortex-A7
Velocidad del CPU	16MHz	Hasta 160 MHz	Hasta 240 MHz	Hasta 1.2 GHz
Memoria Flash	32KB	Hasta 16 MB	Hasta 16 MB	Tarjeta SD externa
Memoria RAM	2KB	Hasta 160 KB	Hasta 520 KB	512 MB

Interfaces Analógicas	Si	Sí	Sí	No
Conectividad Wi-Fi	No	Sí	Sí	Requiere modulo adicional
Interfaces Digitales	GPIO, SPI, I2C, UART	GPIO, SPI, I2C, UART	GPIO, SPI, I2C, UART	GPIO, SPI, I2C, UART
Consumo de Energía	Moderado	Moderado	Bajo	Moderado
Costo	\$50.000	\$20.000	\$35.000	\$150.000

*Nota.* Relaciona un comparativo de las principales características de las tarjetas de desarrollo evaluadas.

### ***Sensores***

Dentro de la problemática inicial se definieron cuatro variables críticas a ser monitoreadas, estas son: el voltaje de la batería, la temperatura del motor, la presión de gases y compresión interna del motor y nivel de combustible. Para la adquisición de estas señales se seleccionaron sensores de voltaje, temperatura, presión y nivel de líquidos.

**Sensor de voltaje FZ0430.** Este módulo consiste en un divisor de tensión conformado por dos resistores uno de 7.5K y uno de 30K, esta relación permite reducir el voltaje medido en una razón de 0.2 por lo tanto se puede medir una tensión máxima de 25 V con un procesador de 5V y 16.5 V con un procesador de 3.3V. El módulo FZ0430, contiene 2 bornes para la entrada de voltaje y tres pines que corresponden a la alimentación del sensor y la salida de señal hacia el microcontrolador. (Arrieta, 2019)

**Figura 11***Sensor de Voltaje FZ0430*

*Nota.* Embebido del sensor de voltaje FZ0430. Adaptado de Ingeniería, Informática y Diseño.

Medir voltajes de hasta 25V con Arduino y FZ0430. Llamas, L. (2016)

<https://www.luisllamas.es/medir-voltajes-de-hasta-25v-con-arduino-y-fz0430/>

Este dispositivo se limita a la medición de voltaje DC. El chip cuenta con un CAD de 10 bits, con una resolución es de 0.00489V, por lo cual, el mínimo cambio de voltaje reconocido a la entrada del sensor es de 0.02445V.

**Tabla 3***Características Sensor FZ0403*

Características	
Voltaje de entrada	0 – 25V
Rango de detección de voltaje	0.02445 - 25 V
Voltaje de resolución analógico	0.00489 V
Interfaz de salida	Vin, GND, Señal
Interfaz de entrada	Vcc, GND
Tamaño	27 × 13,5 x 12 mm

*Nota.* Detalla las principales características del sensor FZ0403. Adaptado de Ingeniería,

Informática y Diseño. Medir voltajes de hasta 25V con Arduino y FZ0430. Llamas, L. (2016)

**Sensor VDO 10.** El sensor de presión de aceite VDO 10 está especialmente diseñado para medir la presión de los líquidos que dan funcionamiento a un motor, ideal para plantas eléctricas y sistemas contra incendios. El sensor funciona de la siguiente manera: Cuando el valor de la presión del aceite es 0-10 bares, se abre el interruptor de alarma y la luz de alarma se apaga, cuando el valor de la presión del aceite es  $0,8 \pm 0,2$  bares, el interruptor de alarma se cierra y la luz de alarma se enciende. (Adams, 2024)

Este sensor se caracteriza por una resistencia variable que cambia de 10 a 184 ohmios, cuenta con un voltaje nominal de operación de 6 a 24V y una potencia nominal de 0.5W. La Tabla 4 contiene sus principales características operativas.

### **Figura 12**

*Sensor de Presión de Aceite VDO10*



*Nota.* Encapsulado del sensor VDO10, cuenta con una rosca de 13mm para adaptar el ingreso de aceite. Adaptado de SR Industrial. Sensor de presión de aceite VDO 0 - 10. (2024)

<https://www.srindustrialsa.com/producto/sensor-de-presion-de-aceite/>

**Tabla 4***Características del Sensor de Presión VDO 10*

Características	
Rango de medición	0 – 10 Bar
Unidad de medición	$\Omega$ Ohm
Alarma de presión	1 $\pm$ 0.2Bar
Potencia Nominal	0.5W
Voltaje Nominal	6 – 24V
Temperatura de funcionamiento	25 – 120°C
Seguridad de sobrepresión	30Bar

*Nota.* Principales características del sensor VDO 10. Adaptado de Sensor de presión de aceite VDO 0-10. Rosi Ltda. <https://www.srindustrialsa.com/producto/sensor-de-presion-de-aceite/>

**Sensor de nivel Agua o Combustible 100TECH.** Este sensor de nivel de líquido, utiliza interruptor de láminas y tecnología de campo magnético, el flotador, construido en acero inoxidable, es la única parte móvil del sensor, minimizando así las posibles fallas mecánicas.

El sensor cuenta con una señal de salida de 240 - 30 ohm (240 ohm en vacío, 30 ohm al máximo), un rango de resolución: 0.4 pulgadas y 0.83 pulgadas y temperatura de funcionamiento desde -40 °F a +85 °F. La Tabla 5 relaciona las principales características operativas del sensor.

**Figura 13**

*Sensor de Nivel Agua o Combustible 100TECH*



*Nota.* Sensor de nivel 100TECH, detalla la longitud máxima de sumersión. Adaptado de Yaxa.

100TECH – Unidad Marina de sensor con indicador para nivel de combustible, gas, agua, para lancha, vehículo, camión, caravana, 240-33 ohm. (2024)

<https://colombia.yaxa.co/products/100tech-unidad-marina-de-sensor-con-indicador-para-nivel-de-combustible-gas-agua-para-lancha-vehiculo-camion-caravana-240-33-ohm/>

**Tabla 5**

*Características del Sensor de Nivel 100TECH*

Características	
Rango de medición	165 – 300 mm
Unidad de medición	$\Omega$ Ohm
Voltaje Nominal	9 – 32V

*Nota.* Detalla las principales características del sensor VDO 10. Adaptado de 100TECH – Unidad Marina de sensor con indicador para nivel de combustible, gas, agua, para lancha, vehículo, camión, caravana, 240-33 ohm, Yaxa. (2024) <https://colombia.yaxa.co/products/100tech-unidad->

[marina-de-sensor-con-indicador-para-nivel-de-combustible-gas-agua-para-lancha-vehiculo-camion-caravana-240-33-ohm/](#)

**Sensor de temperatura DS18B20.** El termómetro digital DS18B20 proporciona mediciones de temperatura en grados Celsius de 9 bits a 12 bits y cuenta con una función de alarma superior no volátil programable por el usuario. La lectura del sensor es hasta 125 ° C Debido a que la transmisión de la lectura de temperatura se hace en protocolo one-wire no hay perdida se señal o inclusión de ruido. Su protocolo de comunicación permite la transmisión de la señal a través de una línea de datos, adicional a la alimentación (Vcc y GND) del sensor.

(Mactronica, 2024)

#### Figura 14

*Sensor de Temperatura DS18B20*



*Nota.* Sonda del sensor DS18B20, su recubrimiento especial es óptimo para ser utilizado en condiciones de humedad. Adaptado de Programar Fácil. DS18B20 sensor de temperatura para líquidos con Arduino. Por Del Valle, H. (2017) <https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>

El DS18B20 puede extraer energía directamente de las líneas de datos, eliminando la necesidad de una fuente de alimentación externa. Cada sensor tiene un código de serie único de 64 bits que lo identifica, permitiendo que varios sensores funcionen en el mismo bus de 1 cable utilizando fácilmente un microprocesador para controlar una gran cantidad de DS18B20. (Del Valle - Hernández, 2017)

**Tabla 6**

*Características Sensor de Temperatura DS18B20*

<b>CARACTERÍSTICAS</b>	
Voltaje de alimentación	3V a 5,5V
VDD	voltaje de alimentación
GND	Tierra
DQ	Datos
Rango de temperaturas	-55°C a 125°C
Error (-10°C a 85°C)	±0,5°C
Error (-55°C a 125°C)	±2°C
Resolución programable	9-bit, 10-bit, 11-bit o 12-bit (default)

*Nota.* Detalla las principales características del sensor DS18B20. Adaptado de Programar Fácil, DS18B20 sensor de temperatura para líquidos con Arduino. (2017)

<https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>

Este sensor presenta un error de medida variable, para temperaturas entre -10°C y 85°C se puede tener una variación de ±0,5°C, sin embargo, para el resto de las temperaturas entre -55°C y 125°C el error es de ±2°C. Por otra parte, la resolución del sensor puede ser programada entre 9 bits, 10 bits, 11 bits y 12 bits, el valor predeterminado es de 12 bits.

**Módulo Relé.** El relé es un componente electrónico que aprovecha el principio del electromagnetismo para operar un interruptor de manera mecánica. Es esencialmente un interruptor activado por corriente eléctrica, lo que lo convierte en un método para controlar un circuito mediante otro. El módulo de relé contiene otros componentes además de la unidad de relé, estos incluyen indicadores LED, diodos de protección, transistores y resistencias, entre otras partes.

El voltaje de activación es de 3,3 – 5V DC, pero puede controlar cargas eléctricas del alto voltaje en AC o DC. (William, 2022) La Tabla 7 muestra las principales características operativas de un módulo de relé convencional.

### Figura 15

*Módulo de Relé*



*Nota.* Módulo de relé convencional, se evidencian los pines de alimentación (Vcc, GND y Señal) y los bornes de salida que controlan la carga. Adaptado de Colombianízate. Módulo relé 1 canal 5V - Arduino. (2024) <https://www.colombianizate.com.co/tienda/arduino/arduino-rele-relay/modulo-rele-1-canal-5v-arduino/>

**Tabla 7***Características del Módulo de Relé*

Características	
Voltaje operativo de entrada	3V a 5,5V $\pm$ 10%
Corriente de carga nominal	5A 250V AC / 30V DC
Corriente de carga transitoria	5A
Potencia de la bobina	0.53W
Tiempo de funcionamiento	< 15ms
Tiempo de liberación	<10ms

*Nota.* Detalla las principales características del módulo relé. Adaptado de ¿Qué es un módulo de relé y cómo funciona? (Tipos y usos). Geya. (2022) <https://www.geya.net/es/what-is-a-relay-module-and-what-does-it-do/>

**Pantalla LCD I2C.** Las pantallas LCD son dispositivos de salida, su funcionamiento consiste en recibir instrucciones del texto que se desea mostrar. Una pantalla LCD I2C consiste en una pantalla LCD típica basada en HD44780 y un adaptador I2C. Su conexión es mucho más fácil que la de una pantalla LCD estándar, solo se necesita conectar 4 pines en lugar de 12. Este tipo de dispositivo de salida cuenta con dos pines de alimentación (Vcc y GND) y dos líneas de comunicación serial (SDA y SCL) encargadas de la recepción de datos y la señal de reloj respectivamente. (Paguayo, 2022)

**Figura 16**

*Pantalla LCD 20X4 I2C*



*Nota.* Pantalla LCD 2X16 integrada a un módulo I2C. Adaptado de Mactrónica. LCD 2004 20x4 back light Azul con conversor I2C. (2024) <https://www.mactronica.com.co/lcd-2004-20x4-back-light-azul-con-conversor-i2c>

El resultado final de esta etapa fue la selección de los dispositivos electrónicos que conformarán el Hardware del sistema, la Figura 17 presenta una imagen de los elementos seleccionados.

**Figura 17**

*Componentes Seleccionados para el Hardware del Sistema*



*Nota.* Componentes del hardware del sistema

## **Diseño del Sistema**

Esta etapa se dividió en dos partes: la implementación del hardware y el desarrollo del software. Durante la implementación del hardware se realizó el diseño de los circuitos esquemáticos determinando la conexión de los diferentes dispositivos electrónicos que conforman el sistema, acoplando los sensores, actuadores y periféricos con el microcontrolador, de manera que las señales recibidas y emitidas se procesaran de manera adecuada en cada componente. Durante el desarrollo del software, se definieron las rutinas de programación del microcontrolador para la lectura de sensores, procesamiento de datos, y transmisión de

información a la plataforma web. Además, se modeló visualmente la plataforma web para permitir una interacción clara y efectiva de las variables monitoreadas.

### ***Implementación del Hardware del sistema***

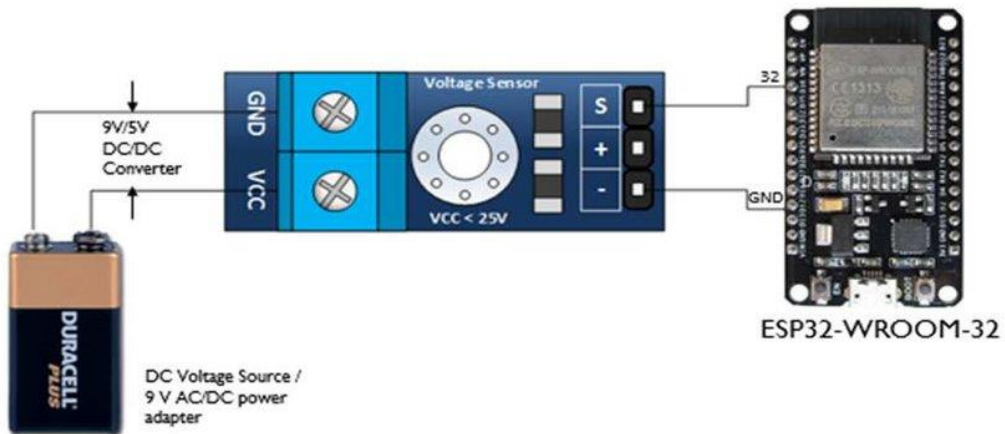
Partiendo del diagrama de bloques propuesto para el sistema, mostrado previamente en la Figura 4, se procedió con el diseño de circuitos de acople para los sensores de manera que se pudieran garantizar los niveles de voltaje y corriente de funcionamiento óptimo; posteriormente se llevó a cabo el desarrollo de códigos de programación para la adquisición de datos por parte de la tarjeta de desarrollo ESP32 y finalmente se realizó la caracterización de los sensores análogos (sensor de voltaje, sensor de presión y sensor de temperatura) determinando tablas de comportamiento y curvas de funcionamiento.

El proceso de caracterización de los sensores involucró una serie de pruebas rigurosas con las que se evaluó la idoneidad de cada sensor, y se generaron tablas de mapeo para entender su comportamiento en entornos reales de operación. Los diseños obtenidos se basaron en fuentes bibliográficas y en prácticas de campo, ya que al optar por sensores industriales que no son inherentemente compatibles con la plataforma Arduino carecen de información detallada. La necesidad de adaptar los sensores industriales para su integración con la ESP32 implicó la creación de diseños personalizados para lograr un acoplamiento y garantizar lecturas precisas.

**Caracterización del Sensor de Voltaje.** Como se explicó anteriormente, este sensor se compone de un circuito divisor de voltaje que se encarga de reducir el voltaje de entrada a un nivel específico más bajo. En este caso, el divisor de voltaje se compone de dos resistencias en serie: una resistencia de 30 k $\Omega$  y otra de 7.5 k $\Omega$ .(Llamas, L. 2016)

**Figura 18**

*Diagrama de Conexiones Caracterización Sensor de Voltaje*



*Nota.* Diagrama de conexiones empleado para la caracterización del sensor de voltaje. Adaptado de Research Gate. Interfacing diagram of the voltage sensor with ESP32. (2021)

[https://www.researchgate.net/figure/nterfacing-diagram-of-the-voltage-sensor-with-ESP32\\_fig4\\_357728653](https://www.researchgate.net/figure/nterfacing-diagram-of-the-voltage-sensor-with-ESP32_fig4_357728653)

Para llevar a cabo la caracterización del sensor fue necesario generar los códigos que permitieran la lectura del sensor por parte de la ESP32, al ser un dispositivo análogo se requiere un mapeo de la entrada que permita convertir el valor en bits adquirido en un valor de voltaje.

## Figura 19

### *Código para Lectura del Sensor de Voltaje*

```

145
146 //.....
147 Serial.println("inicio correcto del votaje de la bateria");
148 float voltaje = 16.5*analogRead(34)/4095;
149 Serial.print("Voltaje bateria = VDC = ");
150 Serial.println(voltaje);
151 delay(500);
152
153 //.....

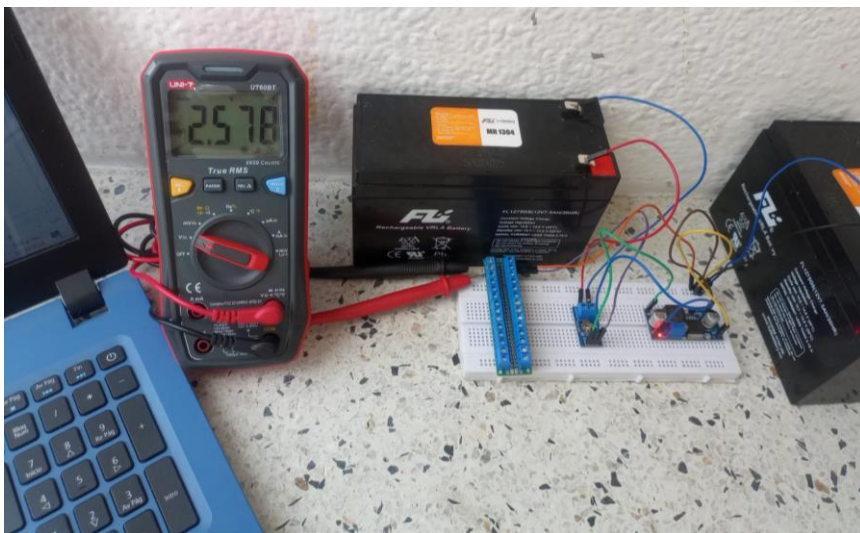
```

*Nota.* Fragmento de código empleado para la caracterización del sensor de voltaje.

El proceso de caracterización se llevó a cabo mediante la conexión del sensor al microcontrolador, y la validación de los valores obtenidos por medio de un multímetro conectado a los bornes de la batería.

## Figura 20

### *Caracterización del Sensor de Voltaje*



*Nota.* Etapa de pruebas y caracterización durante el desarrollo del proyecto

Las lecturas obtenidas durante el proceso de caracterización se consignaron en la Tabla 6, los valores de Vbat corresponden a la medición registrada por el multímetro sobre los bornes de la batería, los valores de Vin corresponden a la lectura arrojada por el puerto análogo y se encuentra en bits y los valores Vout, hace referencia al mapeo obtenido mediante la fórmula de relación implementada por software. Por su parte, la Figura 21 relaciona los valores mapeados de la Tabla 8, allí se evidencia que el sensor pierde linealidad hacia los extremos.

### **Tabla 8**

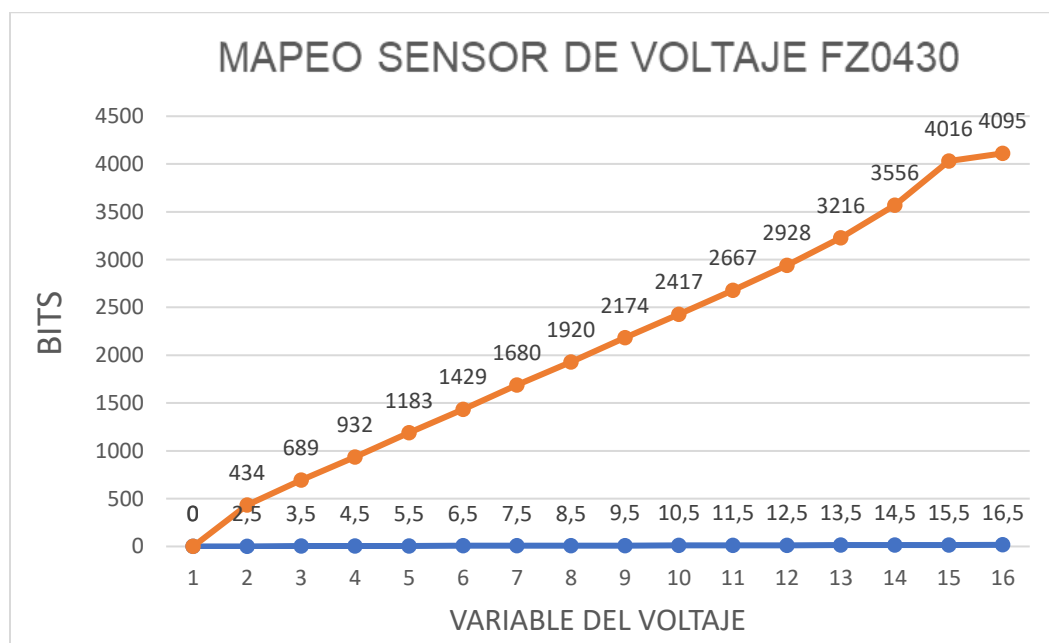
#### *Caracterización del Sensor de Voltaje*

Mapeo Sensor de Voltaje Fz0430		
Vbat	Vin	Vout
2,5v	434	0,434
3,5	689	0,689
4,5	932	0,893
5,5	1183	1,1
6,5	1429	1,293
7,5	1680	1,51
8,5	1920	1,695
9,5	2174	1,891
10,5	2417	2,084
11,5	2667	2,289
12,5	2928	2,489
13,5	3216	2,681
14,5	3556	2,883
15,5	4016	3,089
16,5	4095	3.289

*Nota.* Datos obtenidos durante el mapeo del sensor de voltaje.

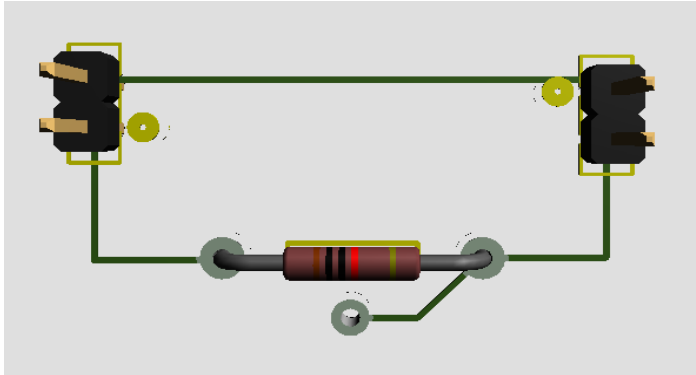
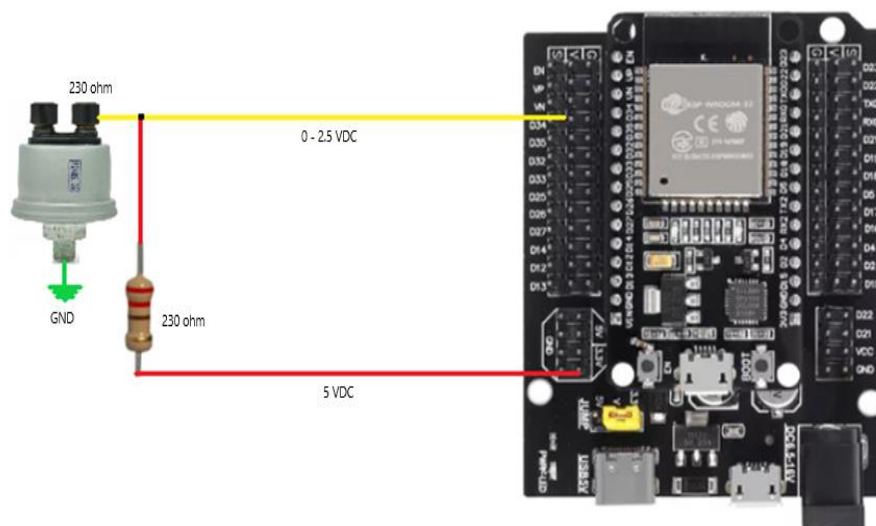
**Figura 21**

*Gráfico de Caracterización del Sensor de Voltaje*



*Nota.* Muestra la relación entre el voltaje medido y su valor de adquisición en bits

**Caracterización del sensor de presión.** De acuerdo con las características operativas, este sensor se comporta como una resistencia variable de 10 – 184  $\Omega$  y cuenta con un voltaje de operación de 6 – 24V. Para la conexión con la ESP 32 es necesario implementar un divisor de voltaje que limite el valor sobre el GPIO del dispositivo. En este caso se estableció como máximo valor de entrada 2.5V, para lo cual se diseñó un circuito de acople basado en un divisor de voltaje, el valor de la resistencia implementada para garantizar el voltaje máximo deseado fue de 230 $\Omega$ .

**Figura 22***Circuito de Acople para el Sensor de Presión**Nota.* Diseño del circuito de acople para el sensor de presión.**Figura 23***Diagrama de Conexiones Caracterización Sensor de Presión**Nota.* Diagrama de conexiones empleado para la caracterización del sensor de presión

Considerando que la variación de voltaje ocurre en el lado negativo y su valor máximo es de 2.5V, se procede con el desarrollo del código para extraer los valores de presión correspondientes de acuerdo con la entrada de voltaje, para ello, al igual que con el sensor FZ0430, fue necesario realizar el mapeo de la señal obtenida a la entrada del microcontrolador.

### Figura 24

*Código para Lectura del Sensor de Presión*

```
Serial.println("inicio correcto de presion");  
float presion = 10*analogRead(33)/2928;  
Serial.print("presion = % = ");  
Serial.println(presion);  
delay(500);
```

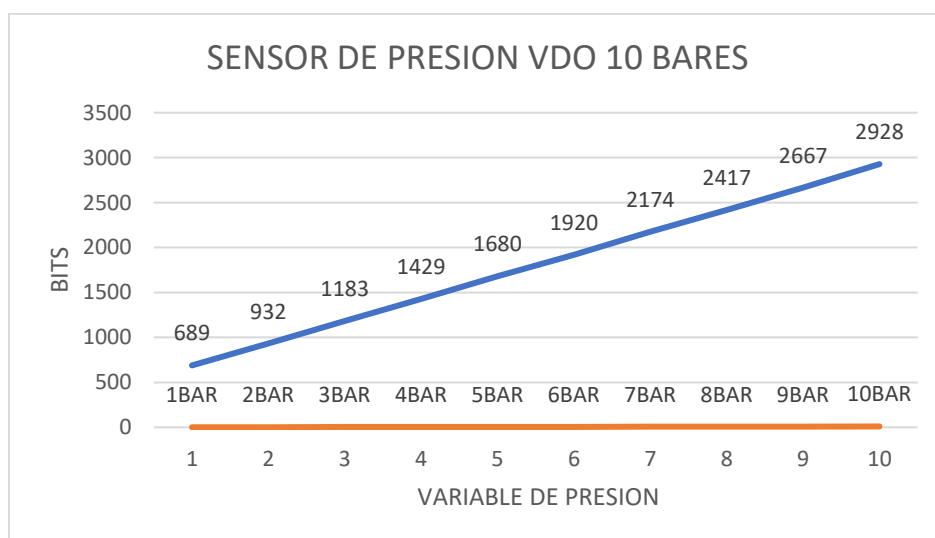
*Nota.* Fragmento de código utilizado la caracterización del sensor de presión

Las pruebas realizadas incluyeron la oscilación del voltaje entre 0 y 2.5V para obtener los resultados de mapeo completo del sensor VDO 10. Estos datos se consignaron en la Tabla 9 donde se puede ver la relación de cambio entre la presión, el voltaje de entrada al microcontrolador y el valor analógico obtenido del ADC en el GPIO. Partiendo de esta tabla, se generó la gráfica del comportamiento del sensor en un rango de operación entre 0 y 10 Bares, mostrada donde se evidencia una relación lineal entre la presión y el voltaje a la entrada del microcontrolador.

**Tabla 9***Caracterización del Sensor de Presión VDO 10*

Mapeo del Sensor de Presion VDO 10		
Presión	Vin	Bits
1	0,689	689
2	0,893	932
3	1,1	1183
4	1,293	1429
5	1,51	1680
6	1,695	1920
7	1,891	2174
8	2,084	2417
9	2,289	2667
10	2,489	2928

*Nota.* Consigna los datos obtenidos durante el mapeo del sensor de presión

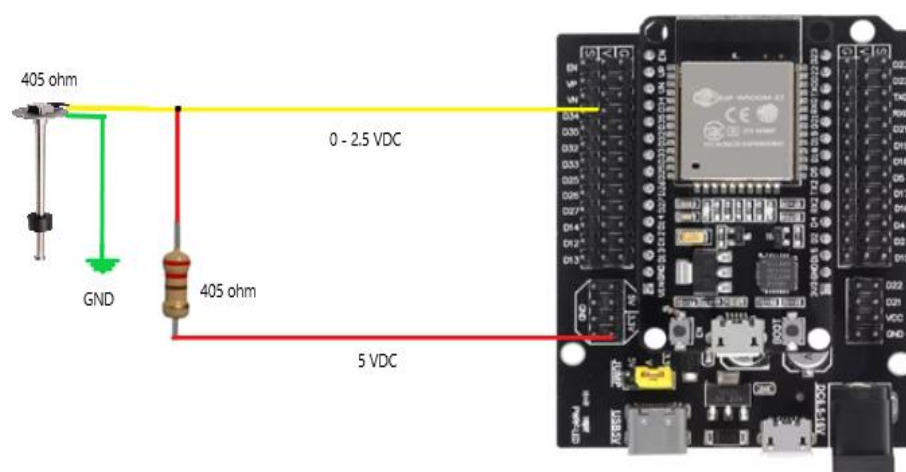
**Figura 25***Gráfico de Caracterización del Sensor de Presión*

*Nota.* Muestra la relación entre la presión medida y su valor de adquisición en bits.

**Caracterización del Sensor de Nivel Combustible.** De acuerdo con sus características operativas, este sensor actúa como una resistencia variable, con un rango de 240 a 30  $\Omega$  y un voltaje de operación entre 9 y 32V. Para conectarlo al ESP32, es necesario implementar un divisor de voltaje que limite el valor aplicado sobre el GPIO del microcontrolador. Al igual que con el sensor VDO 10, se estableció un valor máximo de entrada de 2.5V; para lograrlo, se diseñó el circuito de acople basado en un divisor de voltaje, similar al mostrado anteriormente en la Figura 21, el valor de la resistencia utilizada para asegurar el voltaje máximo deseado fue de 405  $\Omega$ .

**Figura 26**

*Diagrama de Conexiones Caracterización Sensor de Nivel de Combustible*



*Nota.* Diagrama de conexiones utilizado para la caracterización del sensor de presión

Considerando que la variación de voltaje se encuentra entre 0 y 2.5V, se procedió a desarrollar el código necesario para extraer los valores correspondientes al nivel del líquido,

basados en la entrada de voltaje. Al igual que con los sensores previamente utilizados, fue necesario realizar un mapeo de la señal obtenida en la entrada del microcontrolador.

### Figura 27

#### *Código para Lectura del Sensor de Nivel de Combustible*

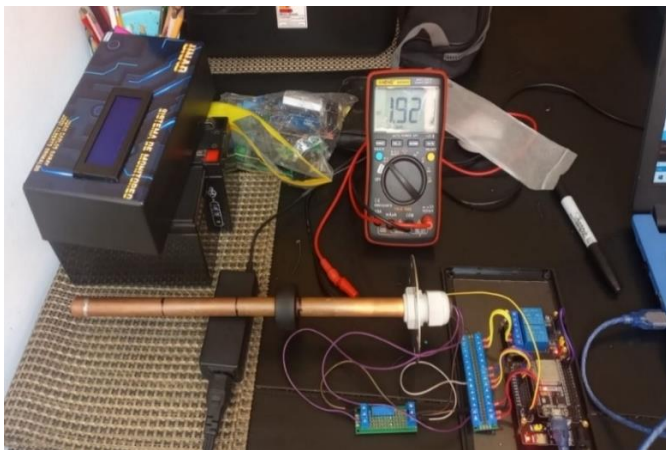
```
Serial.println("inicio correcto de nivel de agua");
float nivel = 100*analogRead(32)/2928;
Serial.print("nivel de combustible = % = ");
Serial.println(nivel);
delay(500);
```

Nota. Fragmento de código implementado para llevar la caracterización del sensor de nivel.

Para las pruebas fue necesario considerar que el sensor ofrece una variación de nivel de 300 mm, lo que limita la dimensión del tanque, igualmente se realizó la oscilación del voltaje entre 0 y 2.5V para obtener los resultados completos del mapeo.

### Figura 28

#### *Caracterización del Sensor de Nivel de Combustible*



Nota. Etapa de pruebas y caracterización durante el desarrollo del proyecto.

Los datos obtenidos se registraron en la Tabla 10, donde se puede observar la relación de cambio entre el porcentaje de llenado del tanque, el voltaje de entrada al microcontrolador y el valor analógico obtenido del ADC en el GPIO. A partir de esta tabla, se generó una gráfica que muestra el comportamiento del sensor en un rango de operación entre 0 y 300mm, donde se evidencia una relación lineal entre el nivel de líquido dentro del tanque y el voltaje a la entrada del microcontrolador.

**Tabla 10**

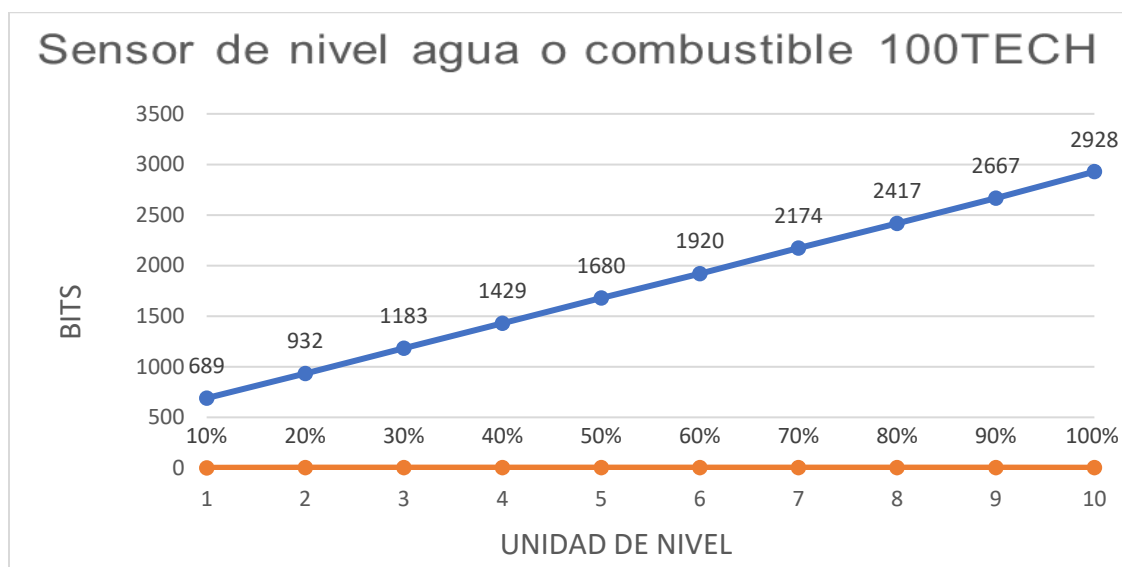
*Caracterización del Sensor de Combustible 100TECH*

Sensor de nivel agua o combustible 100TECH		
Porcentaje de llenado	Vin	ADC
10	0,689	689
20	0,893	932
30	1,1	1183
40	1,293	1429
50	1,51	1680
60	1,695	1920
70	1,891	2174
80	2,084	2417
90	2,289	2667
100	2,489	2928

*Nota.* Consigna los datos obtenidos durante el mapeo del sensor de nivel

**Figura 29**

*Gráfico de Caracterización del Sensor de Nivel de Combustible*

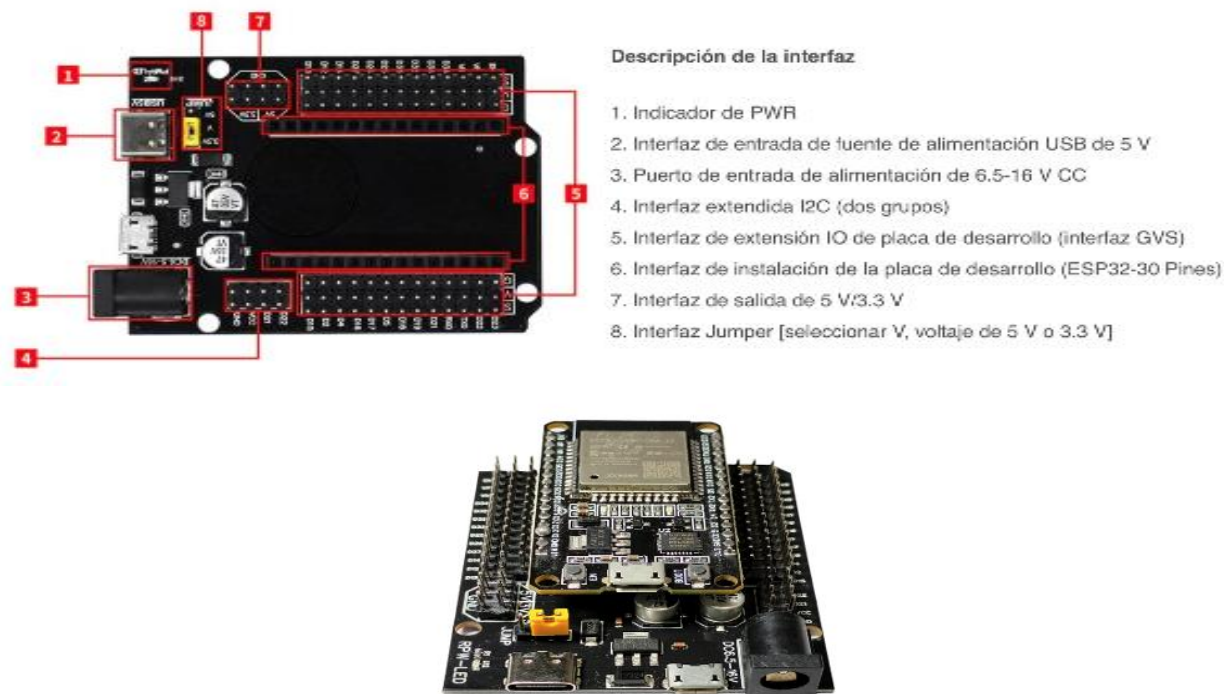


*Nota.* Muestra la relación entre el nivel de agua y su valor de adquisición en bits

Una vez realizada la caracterización de los sensores, se procedió con la conexión de los periféricos y demás componentes del sistema a la tarjeta de desarrollo, para lograr una integración segura de todos los componentes se optó por la implementación de una tarjeta de expansión para la ESP32, la cual ofrece múltiples opciones de conexión, entre ellos fuentes de alimentación de DC (6,5-12 V) o USB (5 V); estandarización de puntos de voltaje estable a 3.3 y 5V lo que simplifica el conexionado y reduce la necesidad de elementos electrónicos adicionales. Esta placa de expansión permite integrar el sistema completo y gestionar los componentes independientemente de su fuente de alimentación.

**Figura 30**

*Tarjeta de Expansión de la Placa de Desarrollo ESP 32*



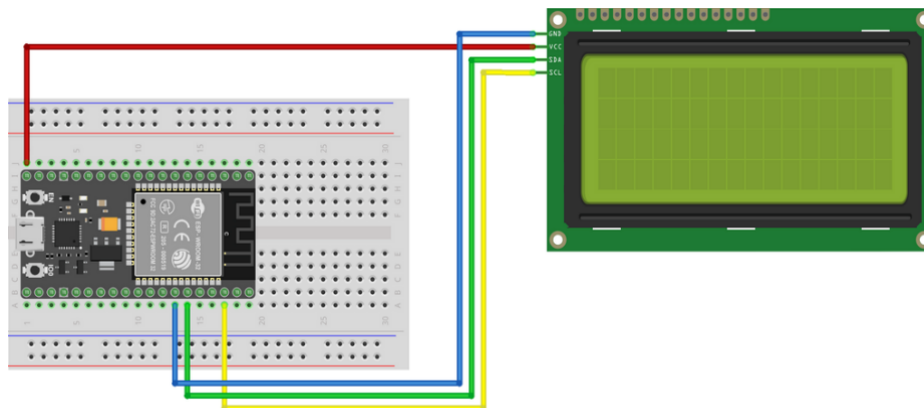
*Nota.* Tarjeta de expansión para ESP32, descripción de pines de conexión. Adaptado de Tettsa, ESP32, 30 pines Placa de Expansión. (2024) <https://tienda.tettsa.gt/producto/esp32-esp32s-30-pines-placa-de-expansion/>

**Conexión de la Pantalla LCD con la Esp32.** Dentro de los periféricos seleccionados para la visualización local se encuentra una pantalla LCD 20x4, donde se muestra el tiempo de actividad del sistema. La capacidad de monitorear el tiempo de actividad es esencial para evaluar el rendimiento y la estabilidad de cualquier dispositivo IoT. La integración con la pantalla LCD permite tener una visualización clara y en tiempo real del tiempo transcurrido desde que la tarjeta

ESP32 fue encendida, proporcionando así información valiosa para el análisis y la gestión del sistema.

### Figura 31

*Diagrama de Conexión de Pantalla LCD*



*Nota.* Esquema de conexión básico de la pantalla LCD usando un módulo I2C con la tarjeta de desarrollo. Adaptado de Talos Electronics, ESP32 y pantalla LCD I2C. (2024)

<https://www.taloselectronics.com/blogs/tutoriales/esp32-y-pantalla-lcd-i2c>

Una vez realizada la conexión de la pantalla a la tarjeta de desarrollo se procedió con el desarrollo de los códigos que permitirán visualizar en pantalla el tiempo de actividad de sistema, para ello se cuenta con librerías disponibles que facilitan la programación del dispositivo embebido

**Figura 32***Código para Pantalla LCD*

```

ESP32_s_LCD20x4.ino

/**
 * Ejemplo ESP32-s y pantalla lcd
 */

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);//crear un objeto lcd (DIRECCIÓN pantalla, Tamaño x,

void setup() {
  lcd.begin();//inicializar la pantalla lcd
  lcd.backlight();//Encender la luz de fondo

  lcd.setCursor(0, 0);//poner el cursor en las coordenadas (x,y)
  lcd.print(" Pantalla lcd 20x4 ");//muestra en la pantalla max 20 caracteres
  lcd.setCursor(0, 1);//poner el cursor en las coordenadas (x,y)
  lcd.print("      esp32-s      ");//muestra en la pantalla max 20 caracteres
}

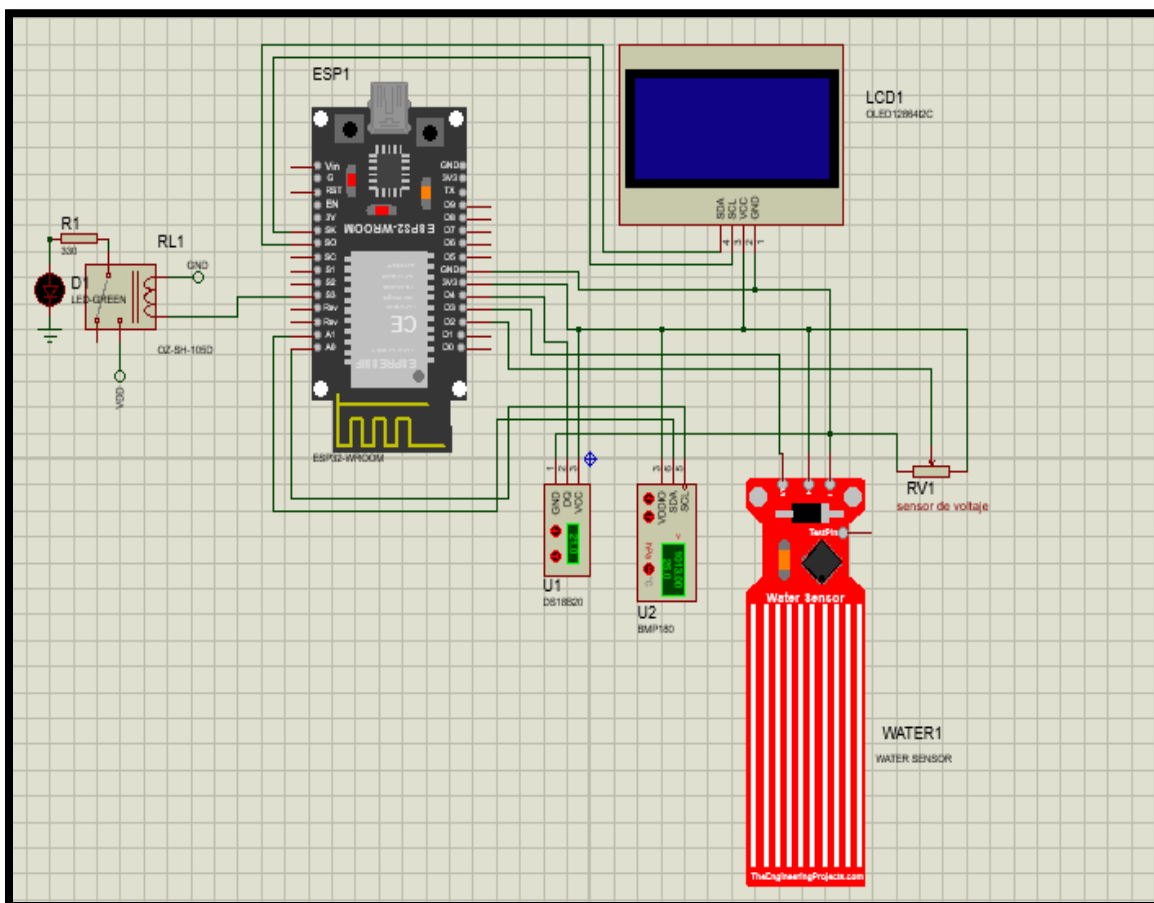
void loop() {
  lcd.setCursor(0, 3);//poner el cursor en las coordenadas (x,y)
  lcd.print("Tiempo activo:");//muestra en la pantalla max 20 caracteres
  //La funcion millis() regresa los ms que lleva encendido
  //Lo dividimos entre 1000 para que nos muestre en segundos.
  lcd.print(millis()/1000);
  lcd.print("s");
  delay(1000);//Esperamos 1 segundo antes de repetir el loop
}

```

*Nota.* Fragmento de código dedicado a la configuración de la pantalla y visualización del tiempo de actividad

El resultado de la etapa de diseño de hardware se ve representado en el plano esquemático general del circuito. En él se integran todos los componentes de hardware que harán parte constitutiva del sistema: cuatro sensores, la pantalla LCD y el relé. Para llevar a cabo este diseño se empleó el software Proteus que permite, además, realizar simulaciones del sistema e identificar posibles fallas de diseño o conexión de dispositivos.

Figura 33

*Plano Esquemático General del Sistema*

Nota. Diagrama de conexiones completo del sistema

*Desarrollo del Software*

Una vez seleccionado el hardware que se empleará para el sistema, se procede con la elección de una plataforma de programación que permita el desarrollo y la codificación del software de manera eficiente, para esto se seleccionó Arduino como plataforma de código abierto y lenguaje de programación intuitivo, que cuenta con un sinnúmero de librerías prediseñadas para

sensores, actuadores y periféricos, adicional a ello permite la integración de una plataforma prediseñada para el monitoreo remoto de variables.

**Arduino IDE.** Arduino ofrece una plataforma de código abierto que permite programar microcontroladores de manera sencilla, lo que la hace ideal para desarrolladores de todos los niveles. El entorno de Arduino proporciona numerosas bibliotecas y recursos que facilitan la implementación de funcionalidades complejas, como la conectividad Wi-Fi y el manejo de sensores.

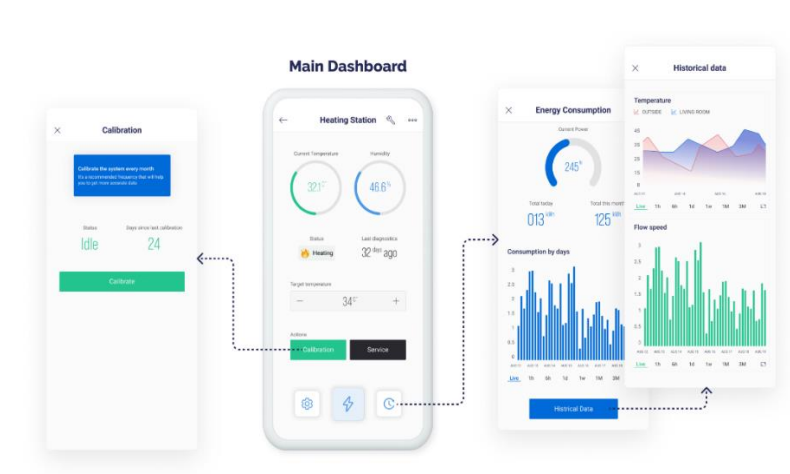
Uno de los principales beneficios del entorno Arduino es su compatibilidad con el ESP32, gracias a esto, se puede acceder fácilmente a herramientas de programación optimizadas para la tarjeta, lo que permite un desarrollo ágil y eficiente. Además, la plataforma de Arduino permite la integración de bibliotecas especializadas, lo que facilita la programación de periféricos y la gestión de las entradas/salidas digitales y analógicas del ESP32. Otro beneficio significativo es la simplicidad del lenguaje de programación basado en C/C++, que reduce la complejidad técnica sin sacrificar el control sobre el hardware. Esto hace que la programación de tareas como la adquisición de datos, el control de dispositivos y la comunicación por Wi-Fi sea accesible para proyectos académicos y profesionales. Al combinar la robustez del hardware del ESP32 con la flexibilidad del entorno de Arduino, se obtiene una solución de desarrollo potente y adaptable a una amplia gama de aplicaciones.

**Blynk IoT.** Esta plataforma con componentes gratuitos fue seleccionada para el proyecto de monitoreo remoto con ESP32 por su excelente compatibilidad y facilidad de uso. Con Blynk, se pudo conectar la ESP32 a la plataforma sin problemas y comenzar a aprovechar todas las características avanzadas que ofrece, como el aprovisionamiento dinámico, las actualizaciones de

firmware OTA y la automatización con notificaciones. Esta integración directa permitió ahorrar una cantidad significativa de tiempo y esfuerzo en el desarrollo del proyecto, ya que no fue necesario destinar recursos adicionales a la programación extensa. En su lugar, se pudo centrar en la creación rápida del producto conectado y en el diseño de una aplicación de firmware integrada que cumpliera con los requisitos de monitoreo remoto de manera efectiva como se muestra en la figura a continuación. (Blynk, 2024)

### Figura 34

*Dashboard con la Plataforma Blynk*



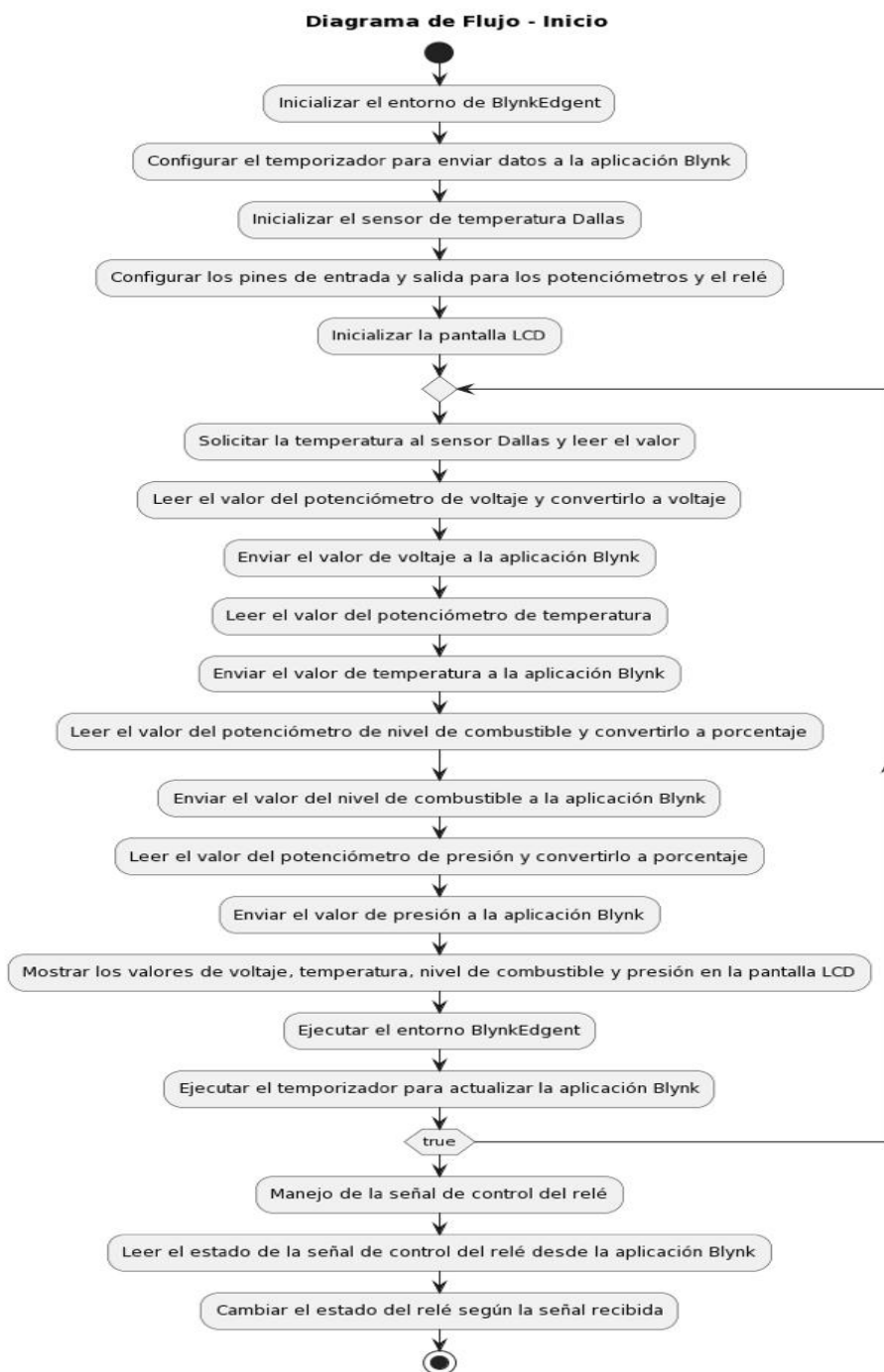
*Nota.* Dashboard construido con la aplicación que permite fácil visualización del estado de variables. Adaptado de Blynk. Components of the Blynk IoT Platform. (2024)

<https://docs.blynk.io/en>

Para iniciar con el desarrollo del software, es indispensable contar con una herramienta que describa el comportamiento general del sistema. La Figura 35 muestra el diagrama de flujo diseñado para el software del sistema, este diagrama determina el procedimiento lógico y ordenado que seguirá el código de manera secuencial en su operación.

Figura 35

Diagrama de Flujo del Software del Sistema



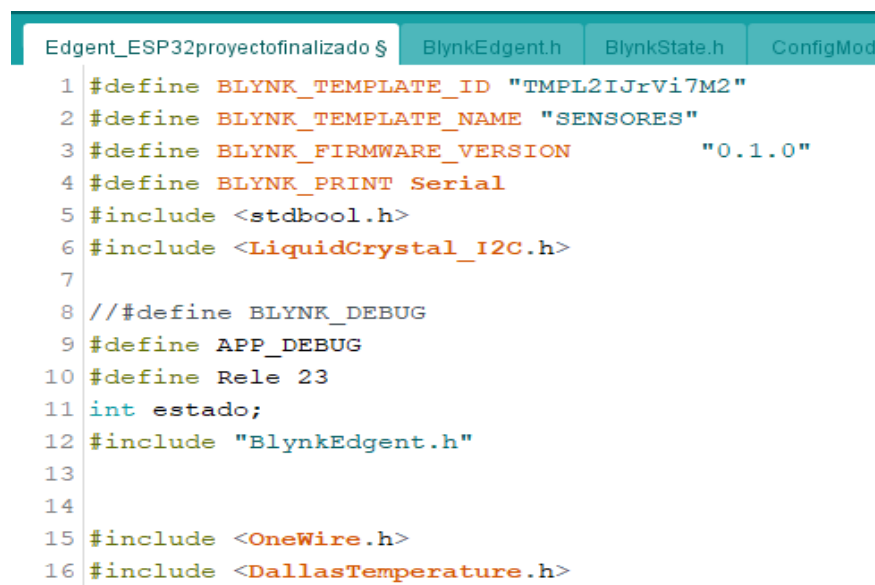
Nota. Diagrama de flujo del sistema, representa el proceso de software programado.

**Código Fuente.** El código se organizó en tres grandes bloques, la primera parte comprende las configuraciones iniciales, la segunda parte se encarga de las rutinas principales de lectura de los sensores y la tercera parte del código está relacionada con la visualización de variables de manera local y remota.

**Configuraciones Iniciales.** Dentro de las configuraciones iniciales se encuentra la inclusión de librerías, entre ellas las librerías de operación de Blynk, la definición de identificadores y configuraciones para el proyecto Blynk; incluyendo el ID del proyecto, el nombre del proyecto y la versión del firmware del dispositivo; así mismo, se incluyen la librería LiquidCrystal para el manejo de la pantalla LCD y las librerías OneWire y Dallas Temperature para la comunicación y lectura con el sensor de temperatura.

### Figura 36

*Inclusión de Librerías en el Código Fuente*



```

Edgent_ESP32proyectofinalizado $ BlynkEdgent.h BlynkState.h ConfigMod
1 #define BLYNK_TEMPLATE_ID "TMPL2IJrVi7M2"
2 #define BLYNK_TEMPLATE_NAME "SENSORES"
3 #define BLYNK_FIRMWARE_VERSION "0.1.0"
4 #define BLYNK_PRINT Serial
5 #include <stdbool.h>
6 #include <LiquidCrystal_I2C.h>
7
8 // #define BLYNK_DEBUG
9 #define APP_DEBUG
10 #define Rele 23
11 int estado;
12 #include "BlynkEdgent.h"
13
14
15 #include <OneWire.h>
16 #include <DallasTemperature.h>

```

*Nota.* Líneas de código que se utilizadas para inclusión de librerías y definición de parámetros de Blynk.

Posteriormente se definen los pines de entrada donde estarán conectados los sensores y salida para el módulo Relé y la pantalla LCD

### Figura 37

#### *Definición de Pines de Entrada y Salida*

```

45 //.....
46 int Sensor = 34;
47 int voltaje = 0;
48
49 int senSor = 35;
50 int tempe = 0;
51
52 int sensor = 32;
53 int combustible = 0;
54
55 int Senso = 33;
56 int presion = 0;
57 //.....

36 OneWire ourWire(25);
37 DallasTemperature sensors(&ourWire);
38 #define COLUMNS 20
39 #define ROWS 4
40 LiquidCrystal_I2C lcd(0x27, COLUMNS, ROWS); //Se inicializa la pantalla LCD de 20X4
41 //.....

```

*Nota.* Líneas de código usadas para la definición de pines

El paso siguiente corresponde con la declaración de los pines virtuales que serán usados por Blynk para la comunicación con la plataforma y visualización remota de variables.

**Figura 38***Definición de Puertos Virtuales Blynk*

```
78 //.....
79 BlynkTimer timer;
80 void myTimer()
81 {
82   Blynk.virtualWrite(V0, PotValor);
83   Blynk.virtualWrite(V1, PotValo);
84   Blynk.virtualWrite(V3, PotVal);
85   Blynk.virtualWrite(V4, PotV);
86 }
87 //.....
```

*Nota.* Líneas de código usadas por Blynk para la comunicación

Por último, la etapa de configuraciones iniciales, culmina con la función Void Setup() donde se inicializa la comunicación serial y la conexión con Blynk; se configura el intervalo de tiempo para la actualización de los valores en Blynk; se inicializa el sensor DS18B20; se configuran los pines de los sensores y el relé y se inicializa la pantalla LCD.

**Figura 39**

*Función Void setup() en el Código Fuente*

```

65
66 void setup()
67 {
68     Serial.begin(115200);
69     BlynkEdgent.begin();
70     timer.setInterval(2000L, myTimer);
71     //.....
72     sensors.begin();
73     //.....
74
75     pinMode(PotPin, INPUT);
76     pinMode(PotPi, INPUT);
77     pinMode(PotP, INPUT);
78     pinMode(Pot, INPUT);
79     pinMode(Rele, OUTPUT);
80
81
82 //.....
83 // inicializar LCD.
84 lcd.init();
85 // encender la luz de fondo.
86 lcd.backlight();
87

```

*Nota.* Configuraciones iniciales en la función *void setup()*

***Rutinas de lectura de Sensores.*** Esta sección se desarrolla dentro de la función *Void loop()*, que lleva a cabo el ciclo de repetición infinita de rutinas para la lectura de sensores, mapeo de valores y almacenamiento de estas cifras en las variables locales y virtuales destinadas para ello. La Figura 40 muestra el fragmento de código correspondiente a la lectura del sensor de temperatura DS18B20 y almacenamiento en la variable *temp* destinada para tal fin; así mismo, dentro de la misma función se realiza la lectura y mapeo de los sensores analógicos (rutinas que se mostraron en las Figuras 18, 23 y 26).

## Figura 40

### *Fragmento de Código Ciclo Void loop()*

```

116 //.....
117 void loop() {
118
119   sensors.requestTemperatures();
120   float temp = sensors.getTempCByIndex(0);
121
122   PotValo = temp;
123   Serial.println(PotValo);
124   delay(1000);
125 //.....

```

*Nota.* Lectura y almacenamiento del valor de temperatura medido

**Visualización de Variables.** Esta última sección, si bien se encuentra ubicada dentro de la función Void loop(), se ha referenciado en un apartado independiente con el ánimo de dividir el código de acuerdo con las funciones que cumple. La visualización de variables se lleva a cabo mediante el puerto serial, la pantalla LCD y el dashboard de la aplicación Blynk.

## Figura 41

### *Impresión de Variables por Consola*

```

137 //.....
138 Serial.println("monitoreo de sensores planta electrica");
139 Serial.println("juan carlos garay _ john grimaldo");
140 delay(500);
141 //.....
142 Serial.println("inicio correcto del votaje de la bateria");
143 float voltaje = 16.5*analogRead(34)/4095;
144 Serial.print("Voltaje bateria = VDC = ");
145 Serial.println(voltaje);
146 delay(500);
147 //.....
148 Serial.println("inicio correcto de temperatura");
149 float tempe = temp;
150 Serial.print("temperatura = °C = ");
151 Serial.println(tempe);
152 delay(500);

```

*Nota.* Impresión de los valores de voltaje y temperatura por consola.

De manera similar se lleva a cabo la visualización de los valores de voltaje, temperatura, presión y nivel de combustible, a través de la pantalla LCD, para ello se envían los datos mediante el uso de funciones de la librería LiquidCrystal.

## Figura 42

### *Código para Visualización de Variables en la Pantalla LCD*

```

172 //.....
173 lcd.setCursor(0,0);
174 lcd.print("Voltaje : ");
175 lcd.print(voltaje,2);
176
177 lcd.setCursor(0,1);
178 lcd.print("Temp   : ");
179 lcd.print(tempe,2);
180
181 lcd.setCursor(0,2);
182 lcd.print("Nivel  : ");
183 lcd.print(nivel,2);
184
185 lcd.setCursor(0,3);
186 lcd.print("Presion : ");
187 lcd.print(presion,2);
188
189 delay(500);

```

*Nota.* Fracción de código encargado de la impresión de datos en la LCD

Finalmente, la visualización de las variables en la plataforma Web se lleva a cabo mediante la escritura de los puertos virtuales, de modo similar, el control del Relé se ejecuta mediante la instrucción BLYNK\_WRITE(V2), cuando, a través de la plataforma, se recibe el comando de activación de la planta. Así mismo, se utiliza la función *myTimer()* junto con las instrucciones para la actualización de puertos virtuales y la visualización remota de variables.

**Figura 43***Escritura de Puertos Virtuales de Blynk*

```

78 //.....
79 BlynkTimer timer;
80 void myTimer()
81 {
82   Blynk.virtualWrite(V0, PotValor);
83   Blynk.virtualWrite(V1, PotValo);
84   Blynk.virtualWrite(V3, PotVal);
85   Blynk.virtualWrite(V4, PotV);
86 }
87 //.....

```

*Nota.* Fracción de código encargada de la escritura de puertos virtuales para Blynk

**Figura 44***Control Remoto del Relé*

```

175 BlynkEdgent.run();
176 timer.run();// runs BlynkTimer
177
178 }
179 BLYNK_WRITE (V2) {
180   estado = param.asInt();
181   digitalWrite (Rele,estado);
182 }

```

*Nota.* Instrucciones de activación del Relé

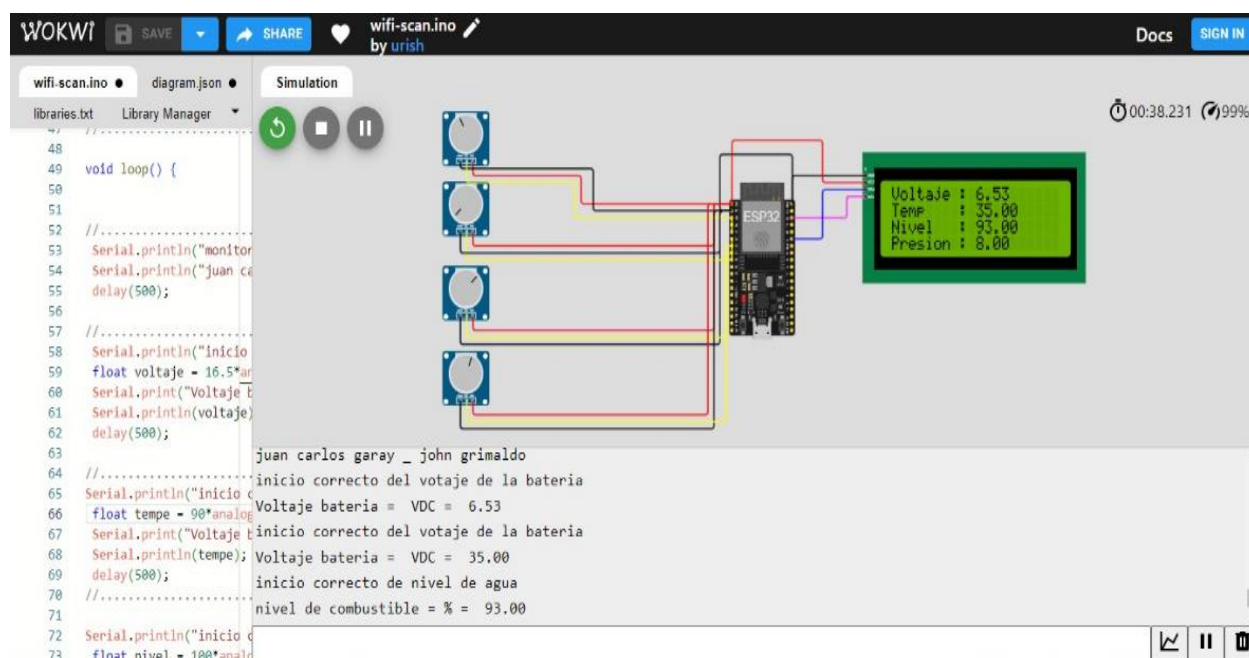
**Integración del Sistema**

Antes de llevar a cabo el montaje final del dispositivo se realizó una simulación del sistema, integrando el circuito propuesto con el código desarrollado. Para ejecutar esta simulación se utilizó la aplicación en línea Wokwi, que permite el uso de diferentes tarjetas de desarrollo, dispositivos electrónicos y la implementación del código para verificar su funcionamiento de manera visual. En este caso se utilizaron tres potenciómetros para simular las

entradas analógicas de los sensores de voltaje, presión y nivel y un sensor digital de temperatura. Los resultados obtenidos mediante el proceso mostraron un correcto funcionamiento del sistema ante la variación de los parámetros de entrada de los sensores. Estas simulaciones fueron fundamentales para validar el diseño antes de pasar a la etapa de implementación física.

### Figura 45

#### *Simulación del Circuito en Wokwi*



#### *Nota. Simulación del circuito*

A continuación, se procedió con la implementación física del sistema, integrando todos los componentes seleccionados y programando el microcontrolador con los códigos desarrollados.

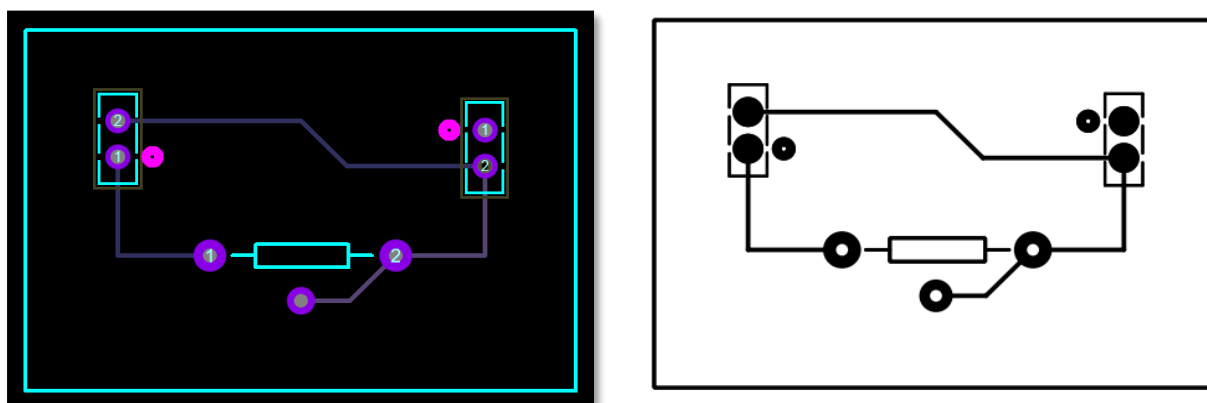
Para la construcción física del sistema se llevaron a cabo procesos como el diseño de circuitos impresos, la elaboración de la PCB a partir de técnicas de maquinado y fresado (proceso

que se contrató de manera externa), la soldadura de componentes, el diseño de un empaque hermético y robusto y la organización de los componentes dentro de este.

Los circuitos impresos fueron diseñados usando el software Proteus, posteriormente los diseños fueron enviados a una empresa especializada para el grabado de las PCB.

### Figura 46

*Plano Impreso y Tarjeta para Circuitos de Acople*



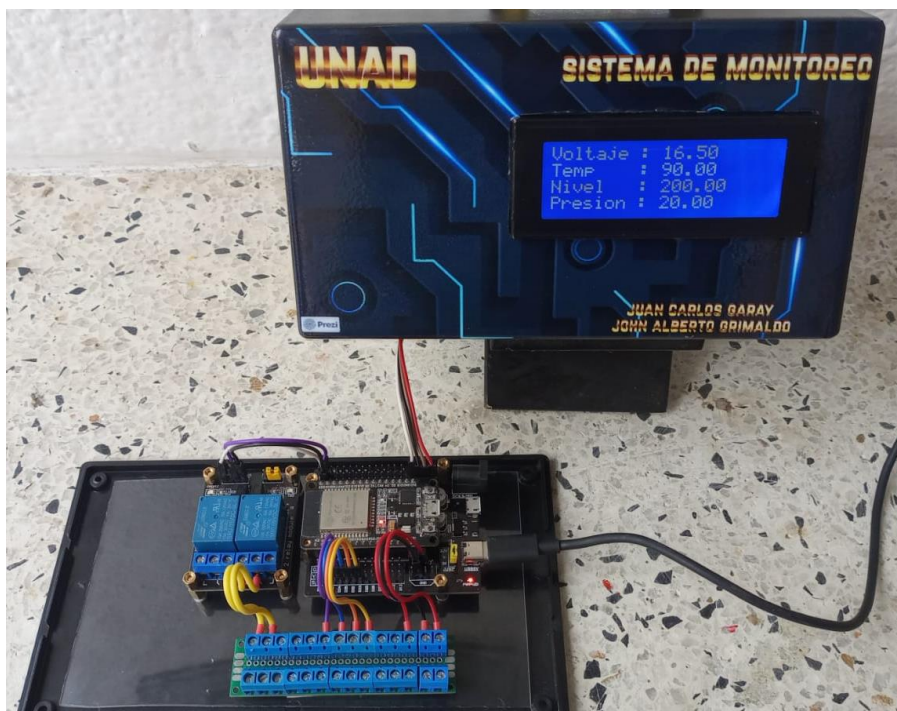
*Nota.* Diseños elaborados para los circuitos de acople de los sensores y el resultado final de la tarjeta terminada

En cuanto a la caja de almacenamiento de los componentes electrónicos, se utilizó una caja plástica de 19cm x 11cm, esta elección se fundamenta en criterios específicos relacionados con el diseño y la funcionalidad del proyecto; las dimensiones ofrecen un espacio adecuado para organizar de manera eficiente todos los elementos del proyecto, además, facilitan un acceso sencillo a los componentes internos para futuras labores de mantenimiento y ajustes. En lo referente al material de la caja, se tuvieron en cuenta aspectos como la resistencia y robustez de manera que se permita la protección de los componentes electrónicos ante condiciones

ambientales como la humedad, la temperatura y el polvo, garantizando así la durabilidad del sistema. También se consideró la estética, asegurando que complemente el diseño general del proyecto y se integre de manera armoniosa en su entorno.

### Figura 47

#### *Caja de Almacenamiento del Dispositivo Electrónico*



*Nota.* Disposición de los elementos al interior y exterior de la caja

### Pruebas y Validación del Sistema

Una vez completada la integración del hardware y el software, se llevaron a cabo pruebas en un entorno controlado para validar el correcto funcionamiento del sistema. Estas pruebas incluyeron la verificación de la precisión de los sensores, la estabilidad de la comunicación

inalámbrica, la respuesta del microcontrolador a los diferentes eventos, y la funcionalidad de la plataforma web.

Las pruebas de campo realizadas buscaron evaluar la exactitud de los sensores, se registraron tablas comparativas para cada una de las variables y se estableció el error estándar en cada sensor. Se realizaron ajustes necesarios basados en los resultados de estas pruebas para asegurar que el sistema cumpliera con los requisitos especificados. Estas pruebas fueron esenciales para garantizar el funcionamiento correcto del sistema y la fiabilidad de las mediciones realizadas.

## Resultados

Con el desarrollo de este proyecto se consiguió desarrollar un sistema de monitoreo remoto eficiente y adaptable para grupos electrógenos, demostrando la importancia de la integración de tecnologías IoT en aplicaciones industriales y resaltando el impacto positivo que estas soluciones pueden tener en la operatividad y mantenimiento de equipos críticos.

El prototipo obtenido se compone de un sistema electrónico centralizado en una tarjeta ESP32, la cual recibe los datos obtenidos por tres sensores analógicos (voltaje, presión y nivel de líquidos) y un sensor digital (temperatura). Las magnitudes medidas corresponden al Voltaje de la batería, la presión en las válvulas del motor, el nivel de combustible de la planta y la temperatura de operación del motor; adicional a ello, el sistema cuenta con un control de encendido y apagado de la planta que se manipula de manera remota a través de la aplicación Web.

### Figura 48

*Prototipo Finalizado*



*Nota.* Prototipo terminado, se evidencian los diferentes componentes del sistema

La visualización de los datos obtenidos por los sensores y procesados mediante código por la tarjeta de desarrollo se lleva a cabo de manera local a través de una pantalla LCD, donde además de los valores de las diferentes magnitudes físicas, se puede verificar el tiempo de operación del sistema, de esta manera obtener datos valiosos para los procesos de mantenimiento preventiva y predicción de posibles fallas del grupo electrógeno.

### Figura 49

#### *Visualización Local de Variables*

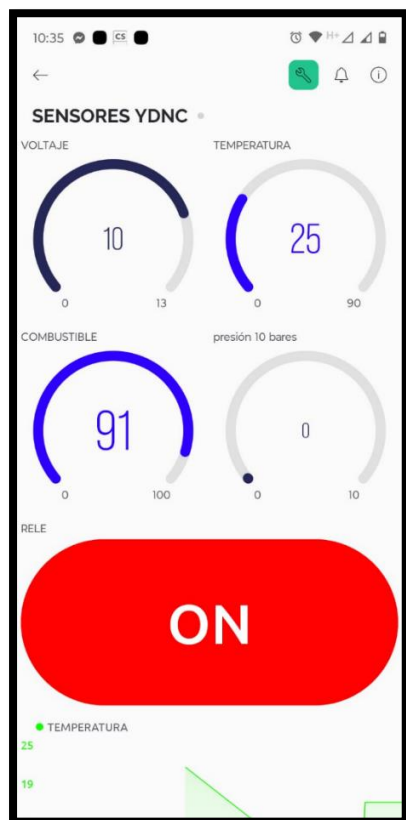


*Nota.* Visualización local de parámetros mediante LCD y consola del computador

Por otra parte, la visualización remota se lleva a cabo mediante la plataforma web Blynk, en la que se creó un Dashboard con cuatro gráficos tipo gauge para representar cada una de las variables medidas y un botón para el control de encendido de la planta.

**Figura 50**

*Visualización Remota Mediante Blynk*



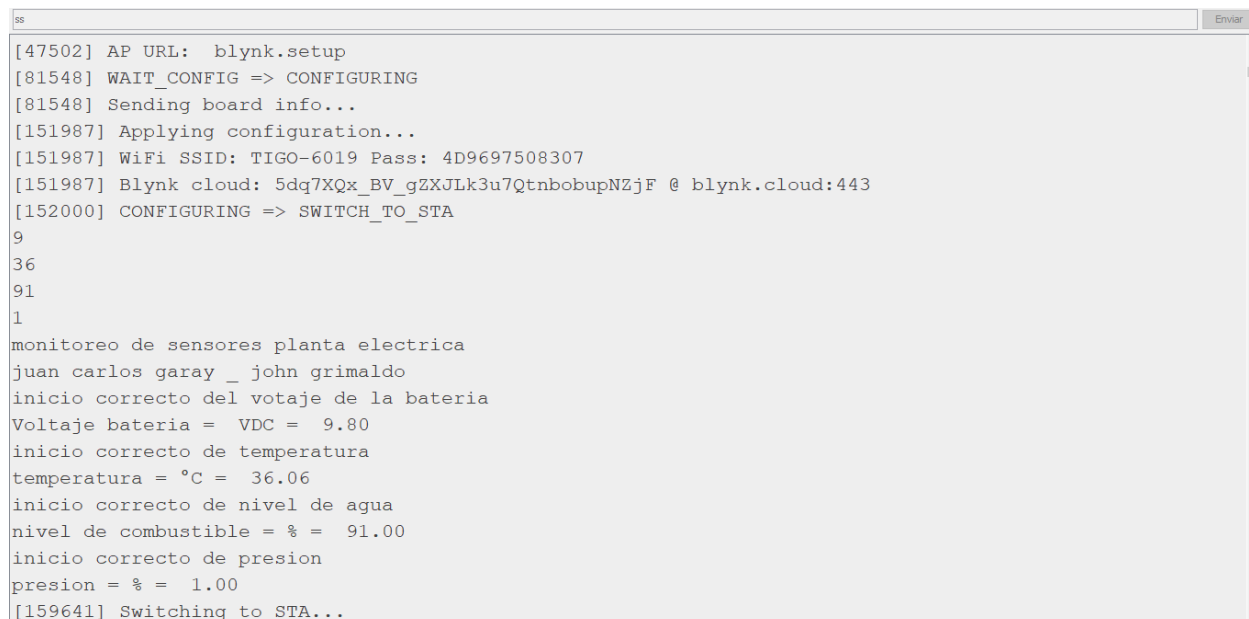
Nota. Dashboard creado en Blynk desde su vista de compatibilidad para dispositivo móvil

Finalmente, como una alternativa para brindar soporte técnico al sistema desarrollado, se integró el monitoreo operativo del prototipo electrónico mediante el envío de datos a través del puerto de comunicación serial de la tarjeta de desarrollo. Para ello se hace necesario conectar el dispositivo a un computador mediante un cable de datos USB y establecer conexión con la consola del IDE de Arduino para la visualización del proceso, allí se puede dar monitorear el funcionamiento de la tarjeta de desarrollo, verificar la conexión a la red Wifi y a la plataforma

Blynk, realizar seguimiento a la operatividad de los sensores y en general, detectar fallas en el hardware del sistema.

## Figura 51

### *Monitoreo Local para Soporte Técnico*



```

[47502] AP URL:  blynk.setup
[81548] WAIT_CONFIG => CONFIGURING
[81548] Sending board info...
[151987] Applying configuration...
[151987] WiFi SSID: TIGO-6019 Pass: 4D9697508307
[151987] Blynk cloud: 5dq7XQx_BV_gZXJLk3u7QtnbobupNZjF @ blynk.cloud:443
[152000] CONFIGURING => SWITCH_TO_STA
9
36
91
1
monitoreo de sensores planta electrica
juan carlos garay _ john grimaldo
inicio correcto del votaje de la bateria
Voltaje bateria = VDC = 9.80
inicio correcto de temperatura
temperatura = °C = 36.06
inicio correcto de nivel de agua
nivel de combustible = % = 91.00
inicio correcto de presion
presion = % = 1.00
[159641] Switching to STA...

```

Nota. Monitoreo de variables por consola para el soporte técnico

Durante la etapa de validación de datos, se evaluó la exactitud de las mediciones obtenidas por los sensores, para este proceso se utilizaron sistemas de medición externos que brindaron información confiable de las magnitudes físicas monitoreadas. La Tabla 11 resume los datos obtenidos durante las pruebas de campo, calculando el error para cada medida realizada. Los cálculos del error fueron establecidos a partir de la aplicación de la siguiente fórmula.

$$\text{Error} = \frac{\text{Valor medido} - \text{Valor esperado}}{\text{Valor medido}} \times 100$$

**Tabla 11***Validación de Datos Obtenidos por los Sensores*

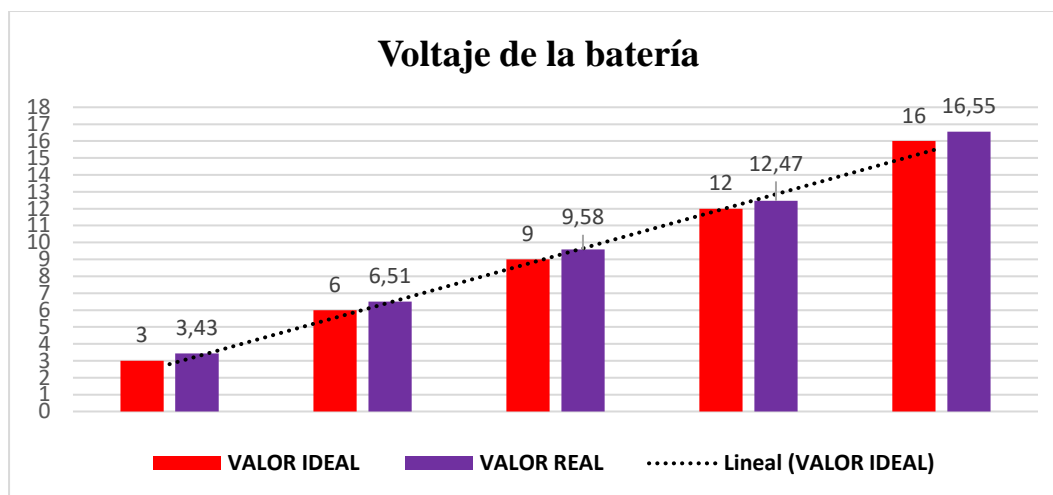
Valor Esperado	Valor Medido	Error
Voltaje de la batería		
3V	3,43V	12,54%
6V	6,51V	7,83%
9V	9,58V	6,05%
12V	12,47V	3,77%
16V	16,55V	3,32%
Presión interna del motor		
1Bar	1,12Bar	10,71%
3Bar	3,27Bar	8,26%
5Bar	5,28Bar	5,30%
7Bar	7,16Bar	2,23%
9Bar	9,13Bar	1,42%
Nivel de combustible		
10%	19%	47,37%
25%	35%	28,57%
50%	59%	15,25%
75%	80%	6,25%
100%	95%	5,26%
Temperatura		
30°C	30,48°C	1,57%
60°C	60,57°C	0,94%
90°C	91,1°C	1,21%
120°C	119,43°C	0,48%
150°C	150,34°C	0,23%

*Nota.* Los datos del Valor Esperado corresponden a las medidas arrojadas por el dispositivo externo de medición, los datos del Valor medido representan las mediciones de los sensores y el error se calculó empleando la fórmula 1.

Se realizaron gráficos para visualizar la relación entre el valor ideal y el valor real para cada una de las variables; la línea de tendencia ilustra la variación entre ambos conjuntos de datos.

**Figura 52**

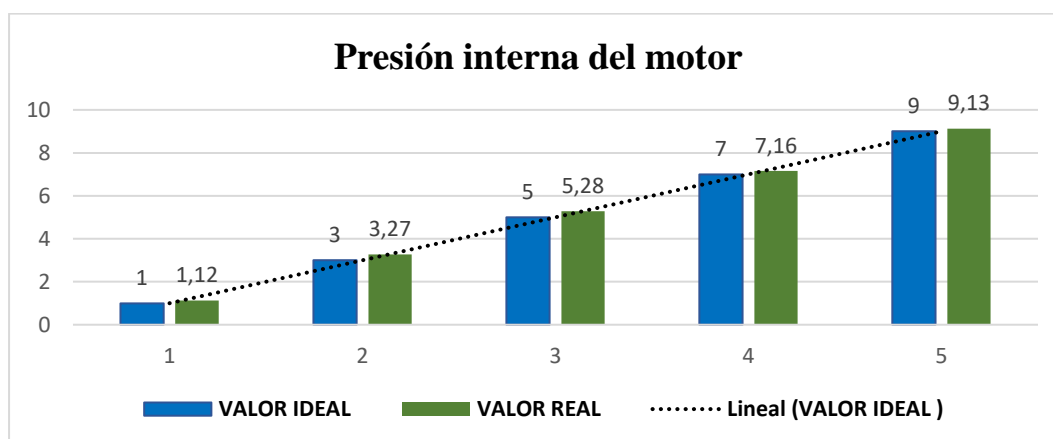
*Comparativo entre Valores Ideales y Reales para el Voltaje Medido*



*Nota.* Relación entre el voltaje esperado y el voltaje adquirido por la tarjeta de desarrollo

**Figura 53**

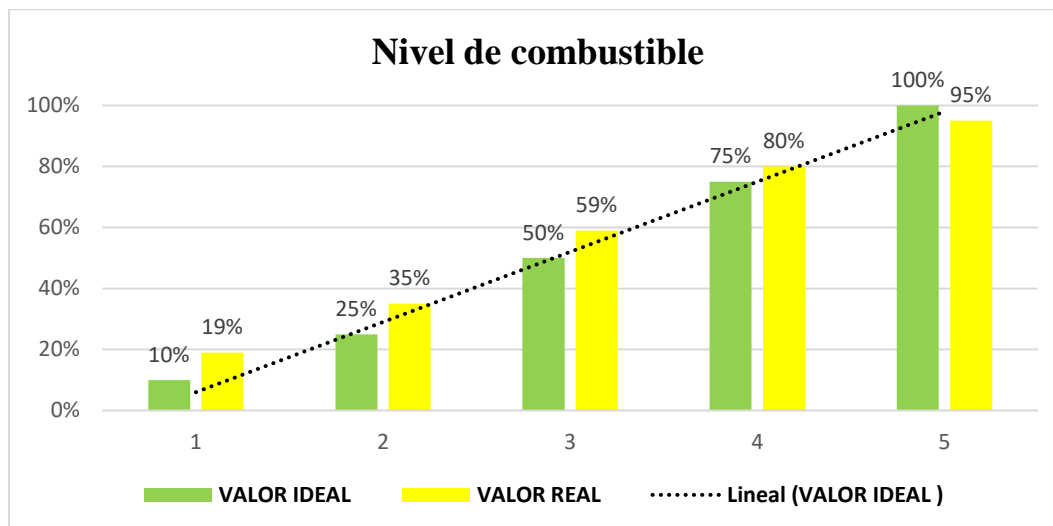
*Comparativo entre Valores Ideales y Reales para la Presión Medida*



*Nota.* Relación entre el valor de presión esperado y el valor adquirido por la tarjeta de desarrollo

**Figura 54**

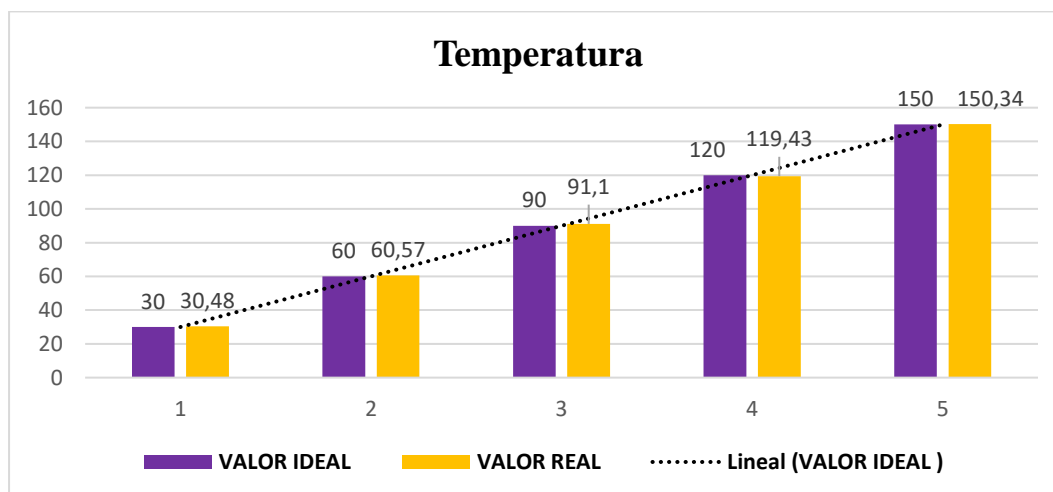
*Comparativo entre Valores Ideales y Reales para el Nivel de Combustible*



*Nota.* Relación entre el nivel de combustible esperado y el valor adquirido por la tarjeta de desarrollo

**Figura 55**

*Comparativo entre Valores Ideales y Reales para la Temperatura Medida*



*Nota.* Relación entre la temperatura esperada y el valor adquirido por la tarjeta de desarrollo.

Se logró establecer que el error porcentual de cada medida para los sensores de voltaje, presión y temperatura no superó en ningún momento el 15%, por lo cual se considera que los valores registrados se encuentran dentro de los rangos esperados, con ello se determina que el dispositivo electrónico arroja datos reales y confiables sobre las variables medidas; sin embargo, el sensor de nivel no mostro grandes variaciones en los valores obtenidos, se cree que esto se debe a que el dispositivo empleado para las pruebas es reutilizado, por lo cual esta deteriorado debido al uso. Debido al alto costo de este elemento se hizo imposible realizar un cambio, sin embargo, se espera lograr corregir esta falla en futuras implementaciones. Estas pruebas determinaron que el sistema es confiable en cuanto los datos obtenidos corresponden a la realidad, sin embargo, se recomienda que el dispositivo debe recalibrarse periódicamente, para garantizar la integridad de los datos a lo largo del tiempo.

## Conclusiones

Con el desarrollo de este proyecto se lograron identificar oportunidades de optimización para el funcionamiento de los Grupo Electrógenos, al supervisar de manera constante factores clave como la carga, el consumo energético y el rendimiento del equipo, con su implementación se consigue mejorar la eficiencia y sostenibilidad del sistema. Además, el monitoreo en tiempo real y la posibilidad de acceder a los datos de forma remota ofrecen una visión completa del comportamiento del grupo electrógeno, lo que contribuye a la toma de decisiones informadas y a garantizar la confiabilidad operativa a largo plazo. El sistema desarrollado no solo demuestra la aplicabilidad y versatilidad del monitoreo remoto en grupos electrógenos, sino que también resalta la importancia de este tipo de soluciones tecnológicas para aumentar la eficiencia, confiabilidad y durabilidad del equipo.

El uso del entorno de programación IDE Arduino para la creación del código de la tarjeta ESP32, trajo consigo una serie de beneficios clave para la implementación del sistema de monitoreo remoto de un grupo electrógeno. La flexibilidad y facilidad de uso del IDE, junto con su extensa biblioteca de funciones, permitió una integración eficiente de sensores y actuadores, simplificando el proceso de creación de prototipos. Además, la comunidad activa de desarrolladores y los recursos disponibles facilitaron el diseño de soluciones innovadoras y personalizadas, optimizando el uso del ESP32 en términos de eficiencia energética y procesamiento. El uso del IDE Arduino no solo contribuyó a la rapidez en el desarrollo del proyecto, sino que también garantizó un proceso de programación estructurado y robusto. La capacidad de la plataforma para admitir diferentes entornos de programación hizo posible la adaptación del sistema a los requisitos específicos del proyecto, permitiendo una mayor estabilidad y confiabilidad en el código final. Esto aseguró que el sistema de monitoreo

desarrollado fuera no solo funcional, sino también escalable y adaptable a futuras necesidades o mejoras.

La implementación de una plataforma de visualización a través de un Dashboard facilitó la interpretación de las variables, permitiendo a los usuarios acceder de manera intuitiva a información crucial, el uso de Blynk IoT jugó un papel fundamental al proporcionar un entorno robusto y flexible para el monitoreo remoto, gracias a sus funciones de aprovisionamiento dinámico a través de WiFi, actualizaciones de firmware OTA y automatización con notificaciones, se garantizó la disponibilidad de los datos en tiempo real desde cualquier dispositivo conectado a internet. Esta capacidad no solo facilitó la toma de decisiones informadas, sino que también promovió la colaboración efectiva entre los usuarios y permitió una gestión ágil y eficiente del sistema monitoreado; simplificó el análisis de datos y proporciona una base sólida para actuar de manera oportuna ante posibles fallos o desviaciones en el rendimiento. Al hacerlo, se mejora el proceso de mantenimiento preventivo y se asegura una respuesta más rápida a las necesidades energéticas de las organizaciones que utilicen este tipo de equipos. Con Blynk se logró una integración completa y eficiente del sistema de monitoreo, simplificando considerablemente el proceso de desarrollo, al proporcionar herramientas listas para usar, la plataforma ahorró tiempo y esfuerzo lo que permitió enfocarse en la personalización del producto. Además, la facilidad de uso de Blynk permitió una experiencia fluida para los usuarios finales, fortaleciendo su capacidad para visualizar y reaccionar de manera efectiva ante cambios en el entorno operativo del grupo electrógeno.

La validación del sistema simulando el funcionamiento de un grupo electrógeno resultó fundamental para evaluar su eficacia y eficiencia. Las pruebas realizadas permitieron confirmar que la solución cumple con los requisitos de funcionamiento y desempeño esperados, asegurando

su confiabilidad antes de su implementación en entornos reales. Al simular condiciones operativas controladas, se pudo identificar posibles áreas de mejora y realizar ajustes necesarios, fortaleciendo la robustez del sistema y garantizando un rendimiento óptimo en su uso final. La validación no solo permitió verificar el correcto monitoreo de las variables críticas del grupo electrógeno, sino que también brindó información valiosa para optimizar el diseño y funcionamiento del sistema. Estos datos ayudaron a perfeccionar tanto el hardware como el software implementado, asegurando que el sistema pueda adaptarse a las demandas energéticas de manera eficiente y sostenible. Asimismo, la capacidad de analizar y ajustar el sistema durante esta fase mejoró significativamente su estabilidad y durabilidad, contribuyendo a su éxito en el campo.

### Referencias Bibliográficas

- Adams. (2024). *Sensor de Presión de Aceite VDO Tipo Generac SDMO (2 Vías) Compatibles Para Controles de Plantas de Luz 360-025 – GRUPO ADAMS* [Comercio]. ADAMS. <https://www.plantasdelluz-adams.com/producto/sensor-de-presion-de-aceite-vdo-tipo-generac-sdmo-2-polos/>
- Angulo Hernández, S. H., & Yarleque Chunga, J. A. (2018). *Diseño de un sistema de transferencia automática de energía eléctrica con monitoreo de parámetros eléctricos para el grupo electrógeno del Hospital Regional de Lambayeque*. [Pregrado, Universidad Señor de Sipán]. <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/4867/%C3%81ngulo%20Hernandez%20-%20Yarleque%20Chunga.pdf?sequence=1&isAllowed=y>
- Arrieta, V. (2019, enero 25). Sensor de Voltaje FZ0430 con Arduino [Blog]. *Arca Electrónica*. <https://www.arcaelectronica.com/blogs/tutoriales/sensor-de-voltaje-fz0430-con-arduino>
- Basco, A. I., Beliz, G., Coatz, D., & Garnero, P. (2018). *Industria 4.0: Fabricando el Futuro*. Inter-American Development Bank. <https://play.google.com/books/reader?id=geiGDwAAQBAJ&pg=GBS.PA26>
- Bautista Esquivel, J. E., Zabala Pérez, L. M., & Mendoza Bohórquez, R. H. (2022). *La Revolución Industrial 4.0, sus características e impacto laboral en Colombia*. <https://digitk.areandina.edu.co/server/api/core/bitstreams/690e1f16-07c3-4344-8862-9fd4ff81c29b/content>
- Belalcazar Diaz, C. F. B., & Quintero Valero, I. F. Q. (s. f.). *REVISIÓN SISTEMÁTICA DE LITERATURA: EFECTOS GENERADOS POR LA APLICACIÓN DE LA LOGÍSTICA 4.0*

*EN LA CADENA DE SUMINISTROS DEL SECTOR INDUSTRIA DE ALIMENTOS EN COLOMBIA.*

- Blynk. (2024, abril 15). *Introduction / Blynk Documentation*. <https://docs.blynk.io/en/>
- Castillo-Vergara, M. (2024). Industria 4.0 en la Pyme: Management & Technology Número Especial. *Journal of Technology Management & Innovation*, 19(1), 3-4.
- Chandy, J. C. (2010). Desafíos en el diseño de sistemas Ciber-Físicos. *Ingenierías USBMed*, 1(1), Article 1. <https://doi.org/10.21500/20275846.231>
- Citelia. (2023, diciembre 14). M2M: Qué es y cómo funciona. *Citelia*. <https://citelia.es/blog/que-es-m2m-diferencias-internet-cosas/>
- Del Valle - Hernández, L. (2017, septiembre 12). DS18B20 sensor de temperatura para líquidos con Arduino [Blog]. *Programarfacil*. <https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>
- EMSA, G. (s. f.). *¿Cómo funciona un grupo electrógeno diésel?* EMSA. Recuperado 17 de diciembre de 2024, de <https://www.emsa.gen.tr/es/soporte/preguntas-frecuentes/como-funciona-un-grupo-electrogeno-diesel>
- Escalas Rodriguez, G. (2015). *Diseño y desarrollo de un prototipo de riego automático controlado con Raspberry Pi y Arduino* [Bachelor thesis, Universitat Politècnica de Catalunya]. <https://upcommons.upc.edu/handle/2099.1/25074>
- Escobar-Gallardo, E., & Villazón, A. (2018). SISTEMA DE MONITOREO ENERGÉTICO Y CONTROL DOMÓTICO BASADO EN TECNOLOGÍA “INTERNET DE LAS COSAS”. *INVESTIGACIÓN & DESARROLLO*, 18(1), 103-116.

- Fernández Ladino, J. M., Briceño Barrero, D. L., & Rodríguez Rojas, L. A. (2022). Industria 4.0: El reto para las pymes manufactureras de Bogotá, Colombia. *Revista Mutis*, 12(1), Article 1. <https://doi.org/10.21789/22561498.1784>
- Fernandez-Morales, F. H., & Duarte, J. E. (2015). Automatismo para el monitoreo y control de un grupo electrógeno con arranque eléctrico. *ENTRAMADO*, 11(2), 262-271. <https://doi.org/10.18041/entramado.2015v11n2.22241>
- Francisco, M., Diaz, R., Castro, M., & Costa, Á. (2007). Grupos electrógenos y calidad de la energía. *Ingeniería Energética*, XXVIII(2), 35-44.
- Gestor. (2021, marzo 2). Guía de mantenimiento de Grupos Electrógenos Diésel. *Genesal Energy*. <https://genesalenergy.com/comunicacion/articulos/guia-mantenimiento-grupos-electrogenos-diesel/>
- González Domínguez, C., & Palomo Vázquez, B. (2015). *Aplicaciones orientadas a la domótica con Raspberry Pi*. Universidad de Sevilla.
- Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and Implementation of ESP32-Based IoT Devices. *Sensors*, 23(15), Article 15. <https://doi.org/10.3390/s23156739>
- Joyanes, L. (2017). *Industria 4.0: La cuarta revolución industrial*. Alpha Editorial.
- Larco-Montenegro, G. (2019). *SISTEMA DE MONITOREO DE GRUPOS ELECTRÓGENOS* [Tesis de Maestría, Universidad Técnica de Ambato]. [https://repositorio.uta.edu.ec/bitstream/123456789/29957/1/Tesis\\_1605masc.pdf](https://repositorio.uta.edu.ec/bitstream/123456789/29957/1/Tesis_1605masc.pdf)
- Llamas, L. (2016, diciembre 29). Medir voltajes de hasta 25V con Arduino y FZ0430. *Luis Llamas*. <https://www.luisllamas.es/medir-voltajes-de-hasta-25v-con-arduino-y-fz0430/>
- Mactronica. (2024). *SONDA SENSOR DE TEMPERATURA DS18B20* [Comercio]. MACTRONICA. <https://www.mactronica.com.co/sonda-sensor-de-temperatura-ds18b20>

- nodoCAST (Director). (2016, noviembre 30). *Industria 4.0 Internet de las Cosas y Sistema Embebidos. nodoCAST* [Video recording].  
<https://www.youtube.com/watch?v=xzLsf7yANmg>
- Ovalle Castro, R. A. (2019, diciembre 18). *La Industria 4.0. Tecnología en Gestión Industrial*.  
<http://repository.unad.edu.co/handle/10596/30589>
- paguayo. (2022, diciembre 9). Interfaz de un LCD I2C con Arduino. *MCI Educación*.  
<https://cursos.mcielectronics.cl/2022/12/09/interfaz-de-un-lcd-i2c-con-arduino/>
- Parihar, Y. S. (2019). *Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products*. 6, 1085.
- Peralta-Abarca, J. del C., Martínez-Bahena, B., & Enríquez-Urbano, J. (2020). Industria 4.0. *Inventio*, 16(39), Article 39. <https://doi.org/10.30973/inventio/2020.16.39/4>
- Pérez A, D. A. (2009). Sistemas Embebidos y Sistemas Operativos Embebidos. *Lecturas en Ciencias de la Computación*.
- Power Rent. (2020, diciembre 14). ⚡ Las fallas más comunes de los generadores eléctricos. *Power Rent*. <https://rentadeplantas.com.mx/fallas-comunes-de-los-generadores-electricos/>
- Quiñones Muñoz, O. (2019). *Internet de las Cosas (IoT)*. Ibukku LLC.
- Ramírez, L. G. C., Jiménez, G. S. A., & Carreño, J. M. (2014). *Sensores y Actuadores*. Grupo Editorial Patria.
- Rojas Suarez, J. P., Vergel Ortega, M., & Orjuela Abril, S. (2021). Análisis del potencial uso de la Generación Distribuida en Colombia. *Boletín Redipe*, 10(6), 428-440.
- William. (2022, noviembre 14). ¿Qué es un módulo de relé y cómo funciona? (Tipos y usos) [Blog]. *Geya*. <https://www.geya.net/es/what-is-a-relay-modue-and-what-does-it-do/>

## Apéndices

### Apéndice A

#### *Código Fuente*

```
//.....  
  
/*  
  
* INCLUSION DE IBRERIAS  
  
*/  
  
//.....  
  
#define BLYNK_TEMPLATE_ID "TMPL2IJrVi7M2"  
  
#define BLYNK_TEMPLATE_NAME "SENSORES"  
  
#define BLYNK_FIRMWARE_VERSION "0.1.0"  
  
#define BLYNK_PRINT Serial  
  
#include <stdbool.h>  
  
#include <LiquidCrystal_I2C.h>  
  
//.....  
  
#define APP_DEBUG  
  
#include "BlynkEdgent.h"  
  
#define Rele 23  
  
int estado;  
  
//.....  
  
#include <OneWire.h>  
  
#include <DallasTemperature.h>  
  
OneWire ourWire(25);
```

```
DallasTemperature sensors(&ourWire);
```

```
#define COLUMNS 20
```

```
#define ROWS 4
```

```
LiquidCrystal_I2C lcd(0x27, COLUMNS, ROWS); //Se inicializa la pantalla LCD de 20X4
```

```
//.....
```

```
/*
```

```
Declaración de variables para la lectura en puerto serial
```

```
*/
```

```
//.....
```

```
int Sensor = 34;
```

```
int voltaje = 0;
```

```
int senSor = 35;
```

```
int tempe = 0;
```

```
int sensor = 32;
```

```
int combustible = 0;
```

```
int Senso = 33;
```

```
int presion = 0;
```

```
//.....
```

```
/*
```

```
Declaración De Variables Para La Lectura En Interfaz De La App Blynk
```

```
*/
```

```
//.....
```

```
const int PotPin = 34;// VOLTAJE
```

```
int PotValor = 0;

const int PotPi = 35;// TEMPERATURA

int PotValo = 0;

const int PotP = 32;// COMBUSTIBLE

int PotVal = 0;

const int Pot = 33;// PRESION

int PotV = 0;

//.....

/*

* función para el envío de datos de puertos virtuales a Blynk

*/

//.....

BlynkTimer timer;

void myTimer()

{

  Blynk.virtualWrite(V0, PotValor);

  Blynk.virtualWrite(V1, PotValo);

  Blynk.virtualWrite(V3, PotVal);

  Blynk.virtualWrite(V4, PotV);

}

//.....

/*

* Configuración de parámetros iniciales
```

```
*/  
  
//.....  
  
void setup() {  
  
    Serial.begin(115200);  
  
    BlynkEdgent.begin();  
  
    timer.setInterval(2000L, myTimer);  
  
    sensors.begin();  
  
  
    pinMode(PotPin,INPUT);  
  
    pinMode(PotPi,INPUT);  
  
    pinMode(PotP,INPUT);  
  
    pinMode(Pot,INPUT);  
  
    pinMode(Rele,OUTPUT);  
  
  
    lcd.init();  
  
    lcd.backlight();  
  
}  
  
//.....  
  
/*  
  
* Cuerpo principal del programa  
  
*/  
  
//.....
```

```
void loop() {  
  
    sensors.requestTemperatures();  
  
    float temp = sensors.getTempCByIndex(0);  
  
    PotValor = 16.5*analogRead(34)/4095;  
  
    Serial.println(PotValor);  
  
    delay(1000);  
  
    PotValo = temp;  
  
    Serial.println(PotValo);  
  
    delay(1000);  
  
    PotVal = 100*analogRead(32)/2928;  
  
    Serial.println(PotVal);  
  
    delay(1000);  
  
    PotV = 10*analogRead(Pot)/2928;  
  
    Serial.println(PotV);  
  
    delay(1000);  
  
    //.....  
  
    /*  
  
    * Se envían las lecturas de los sensores al puerto serie para su monitoreo y análisis
```

```
*/  
  
//.....  
  
Serial.println("monitoreo de sensores planta electrica");  
  
Serial.println("juan carlos garay _ john grimaldo");  
  
delay(500);  
  
  
Serial.println("inicio correcto del votaje de la bateria");  
  
float voltaje = 16.5*analogRead(34)/4095;  
  
Serial.print("Voltaje bateria = VDC = ");  
  
Serial.println(voltaje);  
  
delay(500);  
  
  
Serial.println("inicio correcto de temperatura");  
  
float tempe = temp;  
  
Serial.print("temperatura = °C = ");  
  
Serial.println(tempe);  
  
delay(500);  
  
  
Serial.println("inicio correcto de nivel de agua");  
  
float nivel = 100*analogRead(32)/2928;  
  
Serial.print("nivel de combustible = % = ");  
  
Serial.println(nivel);  
  
delay(500);
```

```
Serial.println("inicio correcto de presion");

float presion = 10*analogRead(33)/2928;

Serial.print("presion = % = ");

Serial.println(presion);

delay(500);

//.....

/*

 * Se envían las lecturas de los sensores al LCD para su monitoreo y análisis

 */

//.....

lcd.setCursor(0,0);

lcd.print("Voltaje : ");

lcd.print(voltaje,2);

lcd.setCursor(0,1);

lcd.print("Temp  :");

lcd.print(tempe,2);

lcd.setCursor(0,2);

lcd.print("Nivel  :");

lcd.print(nivel,2);
```

```
lcd.setCursor(0,3);

lcd.print("Presion : ");

lcd.print(presion,2);

delay(500);

//.....

/*

* Función para lectura de datos de Blynk

*/

//.....

BlynkEdgent.run();

timer.run(); // runs BlynkTimer

}

BLYNK_WRITE (V2) {

    estado = param.asInt();

    digitalWrite (Rele,estado);

}
```