

**Sistema para la supervisión del nivel de agua en el canal de una
pequeña central hidroeléctrica perteneciente a la Empresa Municipal
de Energía Eléctrica S.A E.S.P.**



Universidad
del Cauca

**Trabajo de grado en Automática.
Modalidad: Práctica profesional.**

Natalia Samboni Noguera

**Director: Oscar Amaury Rojas
Asesor Empresa: Luis Ángel Patiño**

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería en Automática Industrial

Popayán

Octubre 2023

**Sistema para la supervisión del nivel de agua en el canal de una
pequeña central hidroeléctrica perteneciente a la Empresa Municipal
de Energía Eléctrica S.A E.S.P**

Natalia Samboni Noguera

**Trabajo de grado presentado a la Facultad de Ingeniería Electrónica y
Telecomunicaciones de la Universidad del Cauca para la obtención del
título de**

**Ingeniero en:
Automática Industrial**

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Ingeniería en Automática Industrial
Popayán
Octubre 2023

Dedicatoria

A mi padre **Giovanni Samboni Ortiz**, gracias por creer en mi siempre, por tu amor sincero, por motivarme y acompañarme siempre que lo necesité, y sobre todo por enseñarme a ser fuerte como tú, espero estés muy orgulloso de mi.

Agradecimientos

Agradezco a **Dios**, por darme la oportunidad de estudiar y darme sabiduría para poder superar cada obstáculo en mi vida.

A mis padres, **Maria del Pilar Noguera** y **Giovanni Samboni**, por estar siempre para mí, por el amor e incondicionalidad, por apoyarme en todo momento y con todas mis decisiones.

A mis abuelos **Jorge Issac Noguera** y **Virgelina Ramirez**, por darme tanto amor, cuidarme, acompañarme y apoyarme todos los días.

A mis hermanos **Daniela Samboni** y **Luis Fernando Samboni**, por cuidarme siempre, por sus consejos de hermanos mayores y por darme tanto amor.

A mi mejor amiga **Mariana Rendon Leal**, por su apoyo incondicional en todo momento, por ser mi confidente, compañera y por convertirse en parte de mi familia.

A mis compañeros de universidad porque gracias a ellos disfruté este proceso.

A la universidad del Cauca por forjarme como profesional y ser una entidad profesional y excelente.

Tabla de Contenido

Lista de Figuras	V
Lista de Tablas	VII
1. INTRODUCCIÓN	1
2. CONCEPTUALIZACIÓN	3
2.1. Empresa Municipal de Energía Eléctrica S.A E.S.P	3
2.2. Puracé - Coconuco	4
2.3. Internet de las cosas	5
2.4. Modelo OSI	6
2.4.1. Nivel Físico	6
2.4.2. Enlace de datos	7
2.4.3. Nivel de red	7
2.4.4. Nivel de transporte	8
2.4.5. Nivel de Sesión	8
2.4.6. Nivel de Presentación	8
2.4.7. Nivel de Aplicación	9
2.5. Modelos de comunicación de internet de las cosas	9
2.5.1. Comunicación dispositivo a dispositivo	9
2.5.2. Comunicación Dispositivo a la nube	10
2.5.3. Comunicación Dispositivo a puerta de enlace	10
2.5.4. Comunicación intercambio de datos mediante back-end	11
2.6. Protocolo de Comunicación	12
2.6.1. ZigBee	12
2.6.2. LoRa	13

TABLA DE CONTENIDO	III
2.6.3. LoRaWAN	13
2.6.4. Bluetooth	13
2.6.5. Wi-Fi	14
2.6.6. Ethernet	14
2.6.7. Modbus	14
2.7. Microcontroladores	15
2.7.1. Arduino	15
2.7.2. Raspberry Pi	16
2.7.3. ESP32	16
2.7.4. PIC	16
2.8. Sensores	17
2.8.1. Sensor de ultrasonido	17
2.8.2. Sensor de nivel capacitivo	18
2.8.3. Sensor de flotador	19
2.8.4. Sensor de nivel por radar de onda guiada	20
3. CRITERIOS DE SELECCIÓN	22
3.1. Elección Protocolo de comunicación	22
3.1.1. Implementación con LoRa	26
3.2. Elección Microcontrolador	43
3.3. Elección Sensor	46
4. METODOLOGÍA	49
4.1. Diseño	52
4.1.1. Diagrama P&ID	53
4.2. Servidor	55
4.3. Circuito	56
4.4. Conexiones	59
4.5. Implementación	60
4.5.1. Montaje	62
4.6. Adquisición de datos	67

<i>TABLA DE CONTENIDO</i>	IV
4.7. Servidor Implementado	75
5. CONCLUSIONES	78
5.1. Recomendaciones	79
Bibliografía	80
6. ANEXOS	85

Lista de Figuras

2.1. Infraestructura de la planta	4
2.2. Mapa Puracé - Coconuco	5
2.3. Modelo OSI	6
2.4. Comunicación dispositivo a dispositivo	9
2.5. Comunicación nodo a la nube	10
2.6. Comunicación nodo a puerta de enlace	11
2.7. Intercambio de datos mediante back-end	12
2.8. Sensor de ultrasonido	18
2.9. Sensor de nivel capacitivo	19
2.10. Sensor de flotador	20
2.11. Sensor de radar de onda guiada	21
3.1. Ubicación de los puntos de medida.	24
3.2. Sistema propuesto con LoRa.	26
3.3. Cobertura a nivel general	29
3.4. No viabilidad del enlace	29
3.5. Niveles de potencia	30
3.6. Viabilidad de cobertura	31
3.7. Perfil del terreno	32
3.8. Diseño en proteus de placa con LoRa	36
3.9. Placa soldada	37
3.10. Prueba exitosa de envío de datos	40
3.11. Antena instalada en la bocatoma	41
3.12. Prueba fallida en la planta	42
4.1. Bocatoma de la planta	49

4.2. Tanque de carga	50
4.3. Tuberías	50
4.4. Turbina y generador en la planta	51
4.5. Proceso del funcionamiento de la planta	52
4.6. Sistema propuesto	53
4.7. Diagrama P&ID.	54
4.8. Propuesta del intercambio de información	56
4.9. Diseño esquemático del circuito	57
4.10. Diseño de la PCB	58
4.11. PCB impresa	59
4.12. Diseño de conexiones	60
4.13. Sensor instalado en bocatoma	62
4.14. Sensor instalado en tanque de carga	63
4.15. Placa con componentes	64
4.16. Montaje completo	64
4.17. Sistema completo energizado	65
4.18. Configuración de velocidad de cambio en la aplicación Endress+Hauser	66
4.19. Configuración de nivel mínimo y máximo en la aplicación Endress+Hauser	66
4.20. Medición e información correcta en el sistema	67
4.21. Pantalla principal del servidor	76
4.22. Pantalla Sensor 1	76
4.23. Pantalla Sensor 2	77

Lista de Tablas

<u>3.1. Criterios selección Protocolo de comunicación</u>	23
<u>3.2. Primera parte Dispositivos para trabajar con LoRa.</u>	33
<u>3.3. Segunda parte Dispositivos para trabajar con LoRa</u>	34
<u>3.4. Última parte Elementos adquiridos para trabajar con LoRa</u>	35
<u>3.5. Criterios de selección microcontrolador</u>	45
<u>3.6. Criterios de selección sensor de nivel</u>	47
<u>4.1. Elementos necesarios para trabajar con Ethernet</u>	61

Capítulo 1

INTRODUCCIÓN

La empresa Municipal de Energía Eléctrica S.A E.S.P EMEESA, ubicada en el departamento del Cauca, opera desde el año 1927 [1]. Esta entidad hace parte del sector eléctrico, cuenta con una pequeña central hidroeléctrica (PCH) en el kilómetro 21 vía al Huila, la cual tiene una capacidad instalada de 4800 kW. La infraestructura de la planta se compone de una bocatoma, que posibilita el acceso al río Grande, el agua captada inicia su recorrido por un canal hasta encontrarse con 2 tanques de carga, unidos a 2 tuberías que conducen el fluido a la casa de máquinas, donde se encuentran 3 turbinas acopladas a generadores, las turbinas 1 y 2 son turbinas tipo Pelton y la turbina 3 es tipo Francis, las cuales tienen una capacidad de 3.2 kW y 1.6 kW respectivamente. El transporte de la energía se realiza mediante una red de distribución de 15 km de longitud aproximadamente, que inicia desde la planta principal hasta la ciudad de Popayán, existen dos subestaciones las cuales son conocidas como subestación Centro y subestación Tulcán, el servicio es prestado en las zonas de Coconuco, Puracé y Popayán, la energía restante generada se inyecta al Sistema de Transmisión Nacional STN, compuesto por un conjunto de líneas, subestaciones y transformadores operando a tensiones iguales o superiores a 220 kV [2]. La comercialización de la energía con el usuario final, se realiza mediante la aplicación de estrategias de mercadeo y pautas para administrar el sistema de facturación atendiendo así a clientes pertenecientes a la línea de red de EMEESA y también de otros operadores de red convirtiéndose en los representantes de dichos clientes sin importar que su operador de red sea otro agente distribuidor [3].

Actualmente, la verificación del nivel se realiza con ayuda de 2 cámaras instaladas en el canal y en el tanque de carga, las imágenes se visualizan en una pantalla ubicada en el

cuarto de supervisión de la casa de máquinas, de forma paralela, los operarios deben realizar inspecciones directas en el canal, teniendo en cuenta que las altas precipitaciones constantemente causan desplazamientos de tierra y desbordamientos del río [4].

Sin embargo, no se tiene implementado un sistema de medición remoto con señales de alarma, que permita monitorear el comportamiento y la evolución de la variable de nivel. Esta práctica, impide generar acciones correctivas oportunas, cuando los niveles de caudal se encuentran por fuera del rango aceptado en el sistema, ocasionando diferentes accidentes, daños en los equipos que componen la planta y grandes pérdidas económicas, hasta el punto de detener la generación de energía. Frente a la problemática mencionada anteriormente, este proyecto busca implementar una solución tecnológica, que permita medir y monitorear el nivel del canal de la PCH de la Empresa Municipal de Energía Eléctrica S.A. E.S.P.

Objetivo general

Proponer un sistema para la supervisión del nivel de agua en el canal de la PCH perteneciente a la Empresa Municipal de Energía Eléctrica S.A E.S.P.

Objetivos específicos

- Identificar las necesidades y condiciones actuales de la PCH Coconuco adscrita a la empresa EMEESA E.S.P.
- Diseñar un prototipo de medición que permita medir el nivel de agua del canal de la PCH perteneciente a la Empresa Municipal de Energía Eléctrica S.A-E.S.P.
- Desarrollar un servidor Web donde se visualice la información de la variable nivel.

Capítulo 2

CONCEPTUALIZACIÓN

En este capítulo, se introduce a una serie de conceptos fundamentales para la comprensión y desarrollo del proyecto en cuestión. Primero se da a conocer información sobre la empresa donde se elaboró el proyecto y más adelante se abordan temas como el 'Internet de las cosas' (IoT), además, se explora el 'modelo OSI', así mismo, se examinan los 'modelos de comunicación', 'protocolos de comunicación', 'microcontroladores' y 'sensores'.

2.1. Empresa Municipal de Energía Eléctrica S.A E.S.P

Es una empresa de energía eléctrica enfocada en la generación, distribución y comercialización desde el año 1927 en adelante, su central hidroeléctrica está ubicada en el departamento del Cauca en el municipio de Puracé-Coconuco, gran parte de dicha infraestructura se presenta en la figura [2.1], por otro lado sus labores administrativas se llevan a cabo en la ciudad de Popayán. En el año 2018, esta empresa logró destacarse como líder en el mercado generador y distribuidor de energía. Sus principales clientes se encuentran en gran parte de la ciudad de Popayán y Puracé, donde provee servicios eléctricos confiables y de calidad [5].

Es importante destacar que EMEESA E.S.P es parte del Sistema de Interconexión Nacional (SIN) en Colombia. El SIN está compuesto por una red de líneas de transmisión y subestaciones de energía que permiten la interconexión entre las diferentes regiones del país. Gracias a esto, se garantiza que gran parte del territorio colombiano tenga acceso a la energía eléctrica [6].



Figura 2.1: Infraestructura de la planta
Tomado de: [5]

2.2. Puracé - Coconuco

Puracé-Coconuco es una región ubicada en el departamento del Cauca, a una distancia de 26 km de la ciudad de Popayán. Se extiende por un área de 707km^2 y presenta una temperatura media de 16°C . La región experimenta precipitaciones anuales que oscilan entre 1600 y 2500 mm [4]. Los suelos en esta región son de origen volcánico con altos contenidos de gas carbónico, azufre y aluminio, en la figura 2.2 se presenta un mapa perteneciente a esta zona.

La población de Puracé-Coconuco cuenta aproximadamente con 17.272 habitantes [7], su economía se basa principalmente en la agricultura y la ganadería, aprovechando las zonas fértiles de la región. Además, en Puracé se encuentra la mina de azufre de Industrias Puracé, que en el pasado generaba alrededor de 300 empleos directos y 500 empleos indirectos. Sin embargo, debido a las normas de libre comercio, la mina de azufre entró en una crisis económica que llevó a su liquidación en 1996. No obstante, en 1998 se reanudaron las operaciones y actualmente la empresa continúa generando empleo para los habitantes, lo que ha sido beneficioso para la economía local [8].

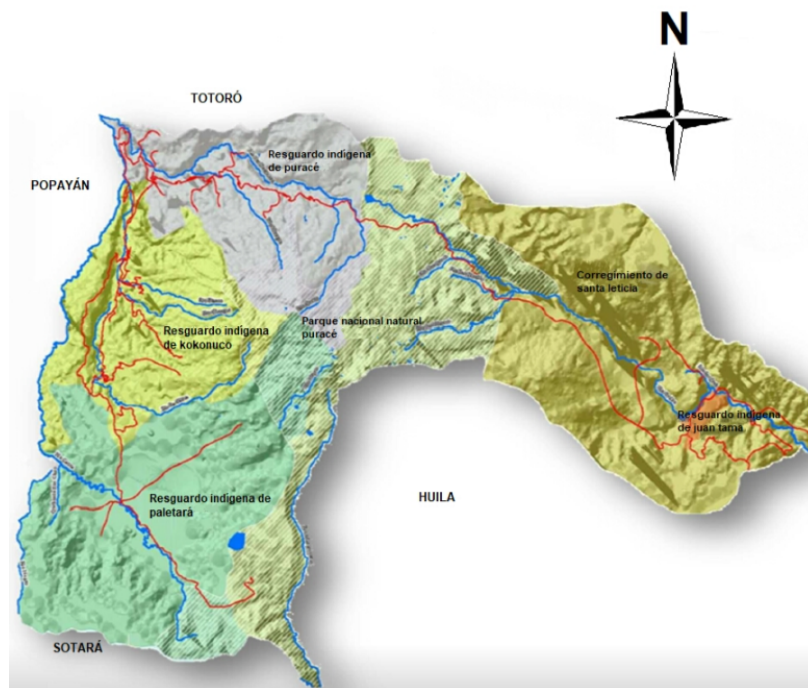


Figura 2.2: Mapa Puracé - Coconuco
Tomado de: [4]

2.3. Internet de las cosas

Este término fue acuñado por primera vez en 1999 por Kevin Ashton, con el propósito de referirse a la capacidad de conectar objetos de uso cotidiano a Internet. En la actualidad, diversos sectores como la salud, la manufactura y la energía están aprovechando los beneficios de la incorporación del IoT en sus operaciones. De acuerdo con las proyecciones de Cisco, el tráfico generado por dispositivos distintos a las computadoras aumentaría del 40 % en 2014 al 70 % en 2019. Esto se debe a que la tecnología del IoT puede aplicarse en una amplia gama de entornos, como el cuerpo humano, el hogar, las oficinas, las fábricas, las obras, los vehículos y los espacios exteriores. Gracias a diversos dispositivos se pueden enviar y recibir datos sin necesidad de intervención humana, lo que permite el control y la monitorización de forma remota desde cualquier ubicación [9].

2.4. Modelo OSI

ISO (international Organization for Standardization) es una organización creada en 1946 en Suiza y establece estándares sobre diversos asuntos entre ellos protocolos de comunicación como fue el caso del modelo de referencia de interconexión de sistemas abiertos (OSI) propuesto en 1978 [10], tiene como objetivo principal que la comunicación entre sistemas incompatibles logren realizar juntos tareas con éxito. Este modelo está dividido en 7 capas las cuales son: Física, enlace, red, transporte, sesión, presentación y aplicación [11], en la Figura 2.3 se muestra el modelo OSI y el modo en como se desplazan los datos desde el emisor al receptor.

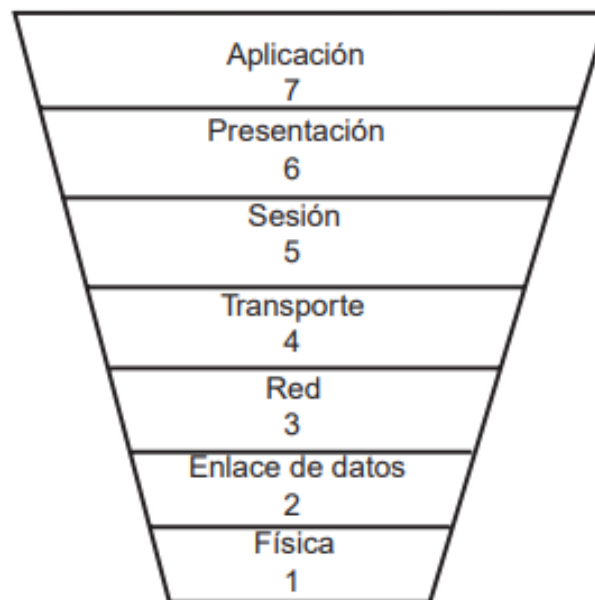


Figura 2.3: Modelo OSI
Tomado de: [11]

2.4.1. Nivel Físico

En este nivel se conectan eléctricamente los dispositivos, se estandariza los conectores y cables, la información que transporta son bits de manera continua y no es legible para el usuario, en este nivel se define diferentes características necesarias para lograr una conexión física [10], entre ellas se encuentran:

- *Eléctrica*: Señala la intensidad, potencial, impedancia y demás características eléctricas que debe tener el interfaz y su entorno eléctrico.
- *Funcional*: Define las operaciones que realiza cada uno de los circuitos.
- *Mecánicas*: Define como son los conectores para los diferentes dispositivos.
- *Procedimiento*: Define el estado de la interfaz en las etapas de la comunicación.

2.4.2. Enlace de datos

La función de esta capa es detectar y corregir posibles errores de los servicios recibidos por el nivel Físico, también es responsable del buen funcionamiento del enlace de datos en toda la comunicación, agrupa los bits recibidos por la capa física en tramas las cuales se enviarán secuencialmente a la siguiente capa. Las redes de difusión manejan funciones especiales para regular el uso del canal de comunicación, por esta razón se divide en dos capas, una de estas se denomina MAC (Media Access Control) resuelve el acceso al medio y la Capa LLC (Logical Link Control) control de enlace lógico [10].

- *Protocolos de la capa de enlace*: CCITT X.25, RDSI, LAP-D.
- *Protocolos de la Subcapa MAC*: Ethernet, Token Ring, FDDI.
- *Protocolos de la subcapa LLC*: IEEE 802.2.

2.4.3. Nivel de red

También conocida como capa IP, desempeña un papel esencial al proporcionar al equipo origen funciones de empaquetado, dirección lógica y física del siguiente salto, así como la capacidad de fragmentación de datos. A través del protocolo IP en esta capa, se lleva a cabo el proceso de direccionamiento, que puede ser de dos tipos: directo, cuando los dos equipos se encuentran en la misma red física, o indirecto, cuando están en redes diferentes. En este nivel se decide la ruta por la cual se enviará la información, la cual se maneja en forma de paquetes [12]

- *Protocolos utilizados:* CCITT X.25 y X.75, Internet Protocol (IP), CCITT/ITU-T Q.931, Q.933, Q.2931 y OSI CLNP.

2.4.4. Nivel de transporte

Se encarga del intercambio de datos entre emisor y receptor, divide correctamente los datos recibidos por la capa sesión con el fin de transmitirlos a la capa de red, es responsable del control de flujo determinando una velocidad óptima para que un emisor rápido no sature a un receptor lento. También realiza un control exhaustivo de errores, asegurándose de que los datos recibidos estén completos. En caso contrario, solicita una retransmisión para mantener la integridad de la información [13].

- *Ejemplo de protocolos para este nivel:* OSI TP4 (Transport Protocol 4), En internet los protocolos son TCP y UDP.

2.4.5. Nivel de Sesión

Establece la comunicación entre el emisor y receptor, se encarga de sincronizar la transmisión de datos. Gracias a esta capa si la comunicación falla cuando se están enviando datos en el momento en el que se restablece la conexión solo se debe enviar la información detrás del último punto de control recibido [14].

2.4.6. Nivel de Presentación

Traduce la información de la capa de aplicación a un formato que el usuario pueda entender, también cifra los datos y los comprime, mejorando así la velocidad y eficiencia en la comunicación minimizando los datos transferidos. Si dos dispositivos conectados utilizan distintos métodos de codificación esta capa traduce los datos del emisor para que así el receptor los comprenda [13].

- *Protocolos:* MPEG, TIFF, GIF.

2.4.7. Nivel de Aplicación

Concede las herramientas al usuario, los servicios de las aplicaciones y acceso general a la red, suministra la transferencia de archivos, mensajes y base de datos, entre estos servicios está la Web y el correo electrónico [11].

- *Un protocolo de esta capa:* TELnet.

2.5. Modelos de comunicación de internet de las cosas

Los modelos de comunicación desempeñan un papel fundamental en la mejora de la expansión, gestión y control de datos en diferentes procesos. Además de optimizar estos aspectos, estas estructuras también tienen la tarea de definir los elementos necesarios para lograr una efectiva interconexión y transferencia de datos. En el contexto de la comunicación en la Internet de las Cosas (IoT), se identifican cuatro tipos de modelos [15], los cuales son:

2.5.1. Comunicación dispositivo a dispositivo

Representa la conexión y comunicación directa entre varios dispositivos que no se conectan por medio de un servidor de aplicaciones, esto se puede hacer mediante redes IP, internet o con diferentes protocolos como lo son Bluetooth, LoRa, ZigBee, LTE. Este tipo de transmisión se utiliza principalmente en sistemas que requieren transmitir una cantidad reducida de datos [15], en la figura 2.4 se ilustra como funciona este tipo de comunicación.

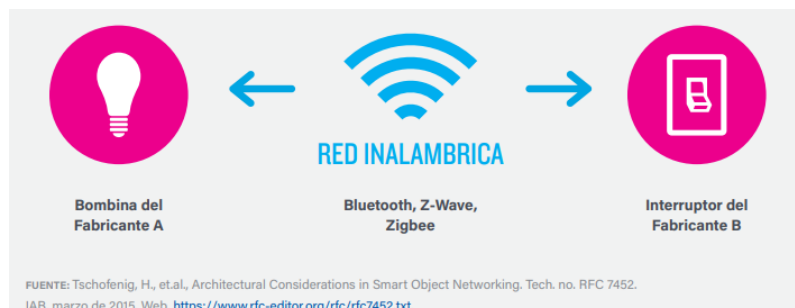


Figura 2.4: Comunicación dispositivo a dispositivo

Tomado de: [9].

2.5.2. Comunicación Dispositivo a la nube

En la figura 2.5 se observa como la comunicación de los dispositivos se lleva a cabo a través de Wi-Fi o Ethernet, mientras se conecta a un proveedor de servicios en la nube. Este proveedor se encarga del envío de datos y del control del tráfico de mensajes. Además, aloja la interfaz web que permite al usuario acceder a la información almacenada en dicho proveedor [16].

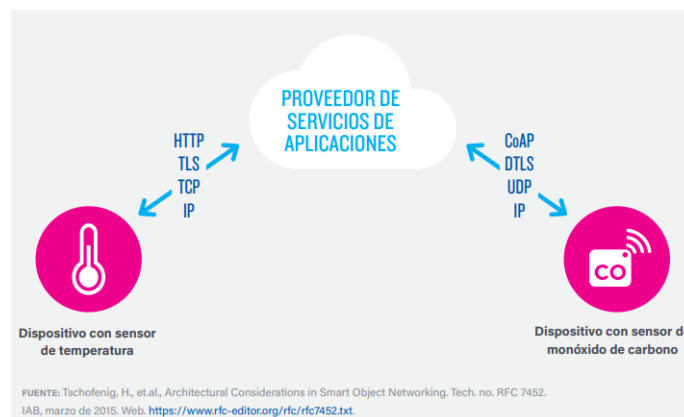


Figura 2.5: Comunicación nodo a la nube

Tomado de: [9].

2.5.3. Comunicación Dispositivo a puerta de enlace

En este modelo, el dispositivo IoT se conecta a un intermediario que facilita el acceso a los servicios en la nube. Esta forma de comunicación requiere un software de aplicación que actúa como puerta de enlace entre el dispositivo y el servicio en la nube. La función de este software es proporcionar seguridad y realizar la traducción de protocolos necesaria. Sin embargo, es importante tener en cuenta que este tipo de comunicación conlleva costos elevados y requiere un trabajo complejo para desarrollar tanto el software como el sistema de la puerta de enlace [17]. En la figura 2.6 se ilustra el funcionamiento de este modelo.

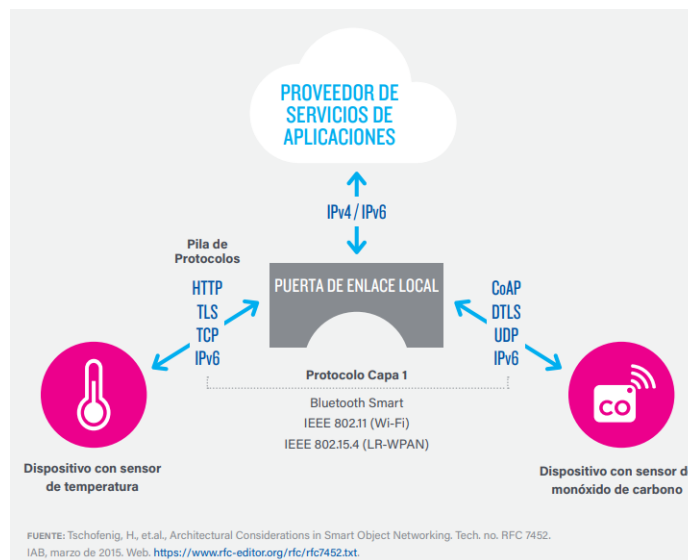


Figura 2.6: Comunicación nodo a puerta de enlace
Tomado de: [9].

2.5.4. Comunicación intercambio de datos mediante back-end

Esta forma de comunicación se presenta como una extensión del modelo 'Dispositivo a puerta de enlace', ya que permite la exportación de datos adquiridos a través de dispositivos inteligentes desde un servicio en la nube. Esta información puede ser analizada por terceros, lo que implica que ya no se presentan los denominados 'silos de datos' [18], los cuales son conjuntos de información aislados e inaccesibles para toda la jerarquía de una empresa. Al eliminar estos silos, el acceso a la información se vuelve instantáneo, esto contribuye a una toma de decisiones más acertada y oportuna [19]. A continuación, en la figura 2.7 se indica una imagen que complementa lo explicado anteriormente.

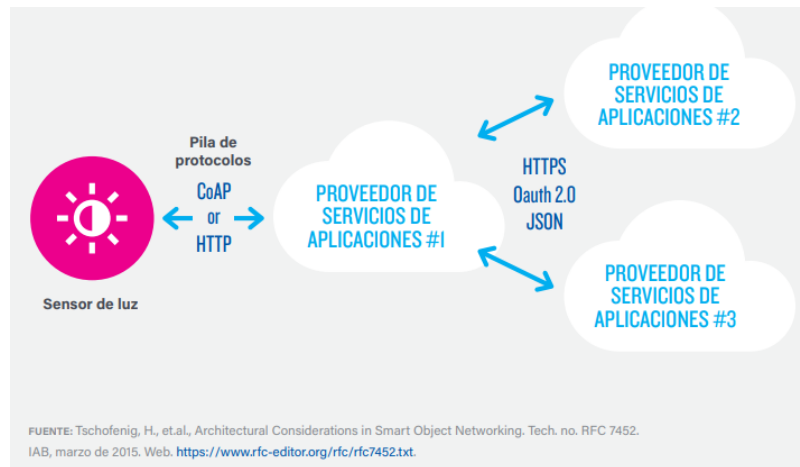


Figura 2.7: Intercambio de datos mediante back-end

Tomado de: [9].

2.6. Protocolo de Comunicación

Los protocolos de comunicación permiten que dos o más dispositivos establezcan una interacción para transmitir información. Esto es gracias a la existencia de un conjunto de reglas y normas previamente establecidas [20].

2.6.1. ZigBee

Es una tecnología inalámbrica basada en Radiofrecuencia que se originó en la alianza HomeRF. Se caracteriza por su bajo consumo de energía y su velocidad de transmisión relativamente baja, que puede oscilar entre 20 y 250Kbps. Además, tiene un alcance de distancia que va desde 10 hasta 75 metros. Una de las ventajas de Zigbee es su eficiencia energética, ya que puede mantenerse en un estado latente cuando no está transmitiendo datos. En términos de frecuencias de operación, Zigbee utiliza la banda de 868 MHz en Europa, 915 MHz en Estados Unidos y la banda global de 2.4GHz con una potencia de transmisión de 1mW. Esta tecnología se utiliza principalmente en aplicaciones de domótica, donde permite la comunicación inalámbrica y el control de dispositivos en el hogar [21].

2.6.2. LoRa

Fue creada por Semtech en 2014, es una tecnología que utiliza el aire como medio de transmisión. Se destaca por su largo alcance, que puede ser desde 2 km hasta 8 km, dependiendo del entorno en el que se utilice.

Al igual que ZigBee, LoRa se caracteriza por su bajo consumo de energía, lo que la hace ideal para dispositivos de baja potencia y aplicaciones de larga duración. Su banda de frecuencia varía según la región geográfica, utilizando los 433MHz en Asia, los 868MHz en Europa y los 915MHz en América [22].

2.6.3. LoRaWAN

Es un protocolo de red desarrollado por LoRa Alliance en 2015, basado en la tecnología LoRa. Es ampliamente reconocido en la industria IoT por su notable alcance y eficiencia energética. En 2021, LoRaWAN fue aprobado como estándar para redes de área amplia de baja potencia (LPWAN) por la Unión Internacional de Telecomunicaciones. Una de las principales ventajas de LoRaWAN es su bajo consumo de energía, lo que permite una vida útil prolongada de las baterías de los dispositivos conectados. Además, ofrece un alcance considerable, que puede abarcar distancias de 3 a 10 km, dependiendo del entorno y las condiciones específicas de implementación. Otra característica destacada de LoRaWAN es el uso de espectro sin licencia. Esto significa que no requiere el pago de licencias para implementar una red LoRaWAN, lo que lo convierte en una opción más accesible y económicamente viable para desplegar infraestructuras de comunicación IoT [23].

2.6.4. Bluetooth

Es un protocolo de comunicación inalámbrica de corto alcance, que generalmente abarca desde los 10 metros y puede llegar hasta los 100 metros con un uso adecuado de amplificadores. Este protocolo está diseñado para su uso en entornos personales, y fue creado con el propósito de permitir la comunicación sin cables entre dispositivos móviles y computadoras. La banda de frecuencia asignada a este protocolo abarca desde los 2400 MHz hasta

los 2483 MHz. Una de las ventajas de Bluetooth es que no es necesario que los dispositivos conectados estén en línea de vista directa, lo que significa que la comunicación puede establecerse incluso cuando hay obstáculos físicos entre ellos [24].

2.6.5. Wi-Fi

Es un modelo de comunicación inalámbrica que ofrece una amplia capacidad de transmisión con tasas de velocidad que varían entre 11 y 54 Mbps [25]. Su principal objetivo es permitir que múltiples dispositivos accedan a Internet. Para lograr esto, los dispositivos deben contar con un adaptador de red, el cual suele estar integrado en los equipos. Esta tecnología se encuentra bien establecida en el mercado, lo que significa que los precios son bajos y hay una amplia oferta disponible [26].

2.6.6. Ethernet

Ethernet es una red de comunicación no determinista que utiliza una topología en forma de bus y es capaz de intercambiar grandes volúmenes de información. Su protocolo, denominado CSMA/CD, permite a los nodos verificar si el bus está libre antes de enviar datos. Si más de un nodo intenta enviar información simultáneamente, se produce una colisión, lo que requiere que los nodos reintenten el envío después de un período de tiempo aleatorio [27]. Inicialmente, la capacidad de velocidad de Ethernet fue de 10 Mbps. Sin embargo, en 1992 surgió la necesidad de mejorar la velocidad de la red, lo que condujo al desarrollo de Fast Ethernet. Posteriormente, en 1995, se creó Gigabit Ethernet para satisfacer la demanda de velocidades aún más altas, alcanzando una velocidad de 10 Gbps. Este incremento en la velocidad ha permitido la integración de Ethernet en aplicaciones de control, donde se requiere una transmisión rápida y confiable de datos [28].

2.6.7. Modbus

El modelo de comunicación MODBUS es un sistema serial maestro/esclavo en el que cada solicitud enviada por el maestro es atendida de manera independiente por el esclavo [29]. Este protocolo permite la comunicación de equipos industriales a través de un mismo canal

y es ampliamente conocido en la industria, lo que facilita encontrar equipos con interfaces compatibles [30]. Existen dos variantes principales de MODBUS: MODBUS RTU, que utiliza representación binaria de la información y puede resultar difícil de comprender para los usuarios; y MODBUS ASCII, que utiliza una representación más comprensible para los usuarios. Si se requiere una conexión a través de TCP/IP, está disponible MODBUS TCP. Este protocolo permite el uso de mensajes MODBUS en un entorno de intranet o internet, aprovechando los modelos TCP/IP [31].

2.7. Microcontroladores

Es un circuito integrado altamente configurable que se adapta a diversas tareas según las necesidades específicas sin la necesidad de elementos externos aparte del cristal y oscilador. Soporta la conexión de sensores y diferentes dispositivos necesarios para efectuar un proceso. En la actualidad, su uso se ha extendido ampliamente en diferentes campos debido a su efectividad, accesibilidad y tamaño compacto, se ha distribuido de forma exponencial. Estos controladores se encuentran presentes en una amplia variedad de aplicaciones, ya que son capaces de integrar inteligencia a cualquier objeto. [32]

2.7.1. Arduino

Arduino fue desarrollado en el Instituto IVREA en 2005 para satisfacer la necesidad de los estudiantes, ya que las placas disponibles en ese momento eran costosas [33]. Se caracteriza por ser un sistema de hardware y software de código abierto, lo que significa que tanto el diseño de la placa como el software son accesibles y modificables por los usuarios. El software de Arduino es gratuito y utiliza un entorno de desarrollo integrado basado en el lenguaje de programación Processing [34].

En cuanto a la comunicación, Arduino utiliza principalmente la comunicación serial, donde los datos se envían bit a bit a través de un único canal. Sin embargo, también es posible la comunicación en paralelo, que requiere una menor latencia debido a la mayor cantidad de buses de datos utilizados para transmitir la información. A pesar de esto, la comunicación

serial ha ganado popularidad debido a su capacidad para minimizar la latencia y permitir la miniaturización de los componentes [35].

2.7.2. Raspberry Pi

Es un ordenador de bajo costo y alta versatilidad que ofrece diversas funciones, desde navegar por Internet hasta jugar. Fue desarrollado en el Reino Unido por la organización Raspberry Pi Foundation, cuyo principal objetivo es fomentar el interés en la informática y la programación en niños y jóvenes. El sistema operativo utilizado en este dispositivo es GNU/Linux, un sistema operativo de código abierto. Raspberry Pi es compatible con diferentes distribuciones de Linux, pero la más común y optimizada es Raspbian OS, que está diseñada específicamente para aprovechar al máximo el potencial de Raspberry Pi [36].

2.7.3. ESP32

El ESP32 es un SoC (System on Chip) fabricado por Espressif Systems que incorpora funciones de conectividad como WiFi y Bluetooth, con una frecuencia de operación de 2.4GHz. Este dispositivo utiliza un sistema operativo en tiempo real y es compatible con FreeRTOS [37], un sistema operativo en tiempo real de código abierto [38]. Al trabajar con FreeRTOS, se simplifica la migración y portabilidad de datos. Una característica destacada del ESP32 es su versatilidad en cuanto a los lenguajes de programación compatibles. Entre ellos se encuentra el IDE de Arduino, lo que facilita el aprovechamiento eficiente y rápido de los recursos del microcontrolador [37].

2.7.4. PIC

El microcontrolador PIC ha alcanzado un notable éxito debido a su manejo sencillo y su bajo costo en comparación con otras placas de desarrollo. Se caracteriza por su alta velocidad de funcionamiento. Así mismo, presenta un bajo consumo de energía, lo que resulta beneficioso en términos de eficiencia y duración de la batería. Su tamaño reducido lo hace adecuado para proyectos con limitaciones de espacio. El microcontrolador PIC se encuentra disponible

en cuatro diferentes gamas: enana, baja, media y alta. Es responsabilidad del ingeniero de diseño seleccionar la gama que mejor se ajuste a las especificaciones de su proyecto [39].

2.8. Sensores

Los dispositivos de medición, conocidos como sensores, proporcionan información precisa sobre una variable medida con el fin de procesarla y generar órdenes de acuerdo a esta [40]. Existen diferentes formas de clasificarlos, una de ellas se basa en el principio de transducción utilizado, que se refiere a cómo el sensor convierte la variable medida en una señal. Sin embargo no siempre es práctica, ya que puede ser difícil tener información clara sobre el tipo de variable que un sensor específico puede medir, por otro lado se tiene los sensores clasificados según la variable medida, siendo estos los más utilizados. Los sensores pueden medir variables como temperatura, presión, nivel, humedad, luz, entre muchas otras. Estos dispositivos tienen características estáticas y dinámicas que los distinguen entre sí. Las características estáticas son aquellas que no cambian con el tiempo y proporcionan información sobre la sensibilidad del sensor, su rango de medición, precisión, exactitud, línea estática, offset, resolución y error estático. Por otro lado, las características dinámicas son variables que cambian en función del tiempo. Estas características incluyen el tiempo de respuesta del sensor, como lo son: la histéresis, la línea dinámica y el error dinámico [41].

2.8.1. Sensor de ultrasonido

Los sensores ultrasónicos son dispositivos utilizados para medir distancias sin contacto físico, aprovechando la reflexión de ondas de ultrasonido. Estas ondas están fuera del rango de audición humana, se emiten periódicamente a alta frecuencia, generalmente entre 40 kHz y 400 kHz, y se propagan a través del aire. Cuando la vibración ultrasónica incide en un objeto, parte de ella se refleja y regresa al sensor. Utilizando el tiempo transcurrido entre el envío y la recepción de la onda reflejada, el sensor ultrasónico calcula la distancia entre el dispositivo y el objeto. Una de las ventajas de estos sensores es que no se ven afectados por condiciones como suciedad, niebla o polvo, lo que los hace adecuados para diversas aplicaciones en entornos desafiantes [42]. En la figura 2.8 tomada de [43] se observa

como funciona este tipo de sensores.

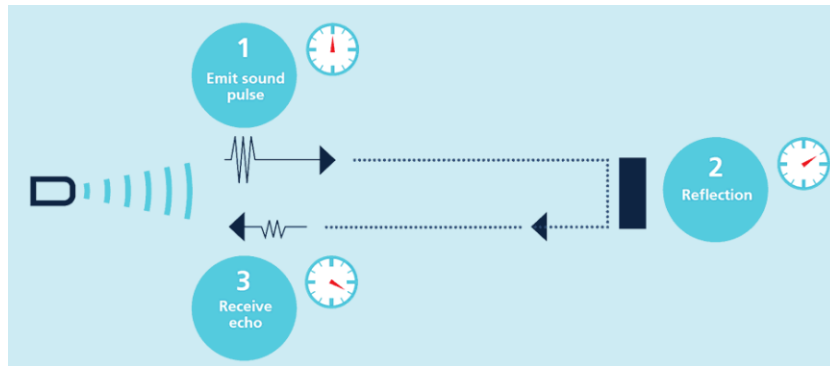


Figura 2.8: Sensor de ultrasonido
Tomado de: [43].

2.8.2. Sensor de nivel capacitivo

El sensor de nivel de líquido basado en capacitancia está compuesto por un condensador y su medición se fundamenta en el cambio de la capacitancia del sistema. En este tipo de sensor, la placa del condensador está formada por un electrodo, mientras que las paredes del contenedor actúan como la otra placa del condensador. La variación de la capacitancia depende del líquido presente, el cual actúa como el material dieléctrico en el condensador. De esta manera, el sensor puede determinar el nivel del líquido basándose en los cambios en la capacidad eléctrica. Es importante destacar que este tipo de sensor necesita estar en contacto directo con el líquido para lograr una medición precisa [44]. En la figura 2.9 se visualiza el funcionamiento de este sensor.

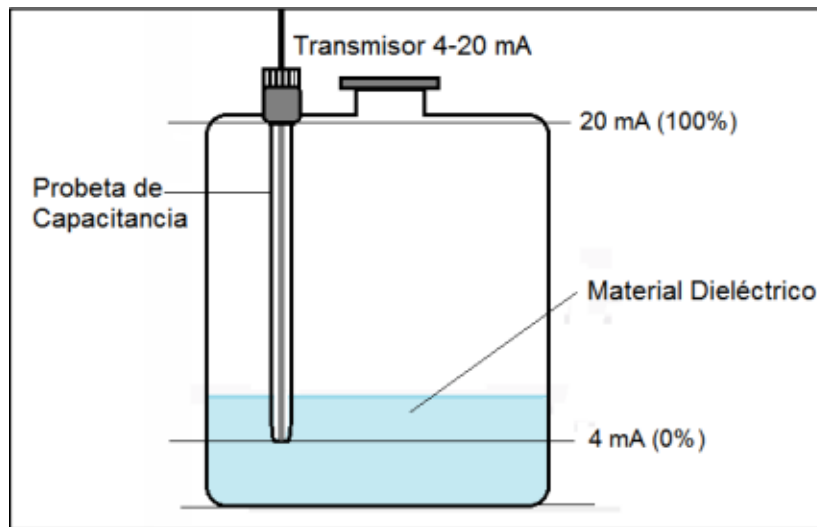


Figura 2.9: Sensor de nivel capacitivo

Tomado de: [44].

2.8.3. Sensor de flotador

Este sensor utiliza un flotador que se encuentra sumergido en el líquido a medir, lo que implica que debe estar en contacto directo con la variable medida. El flotador puede ser conectado al exterior de diferentes maneras [45], entre las cuales se encuentran las siguientes:

- *Conexión Directa*: Esta es la conexión más común y antigua. En este método, el flotador está conectado mediante un cable que se desliza a través de poleas. Esto permite que el cable se mueva a medida que el flotador cambia de posición, lo que indica el nivel del líquido. La posición del índice en el extremo del cable proporciona una indicación visual del nivel de líquido [45].
- *Conexión Magnética*: En este método, el tanque contiene un flotador que está conectado a un tubo guía. El flotador se desliza verticalmente a lo largo de este tubo. En el tubo, hay una pieza magnética que está unida al cable de medición. La pieza magnética sigue el movimiento del flotador y transmite esa información al cable. El valor medido se puede visualizar en la parte superior del tanque utilizando un índice de medición que varía según el movimiento del flotador [45].

La figura 2.10 ilustra el funcionamiento de este tipo de sensor y sus diferentes conexiones.

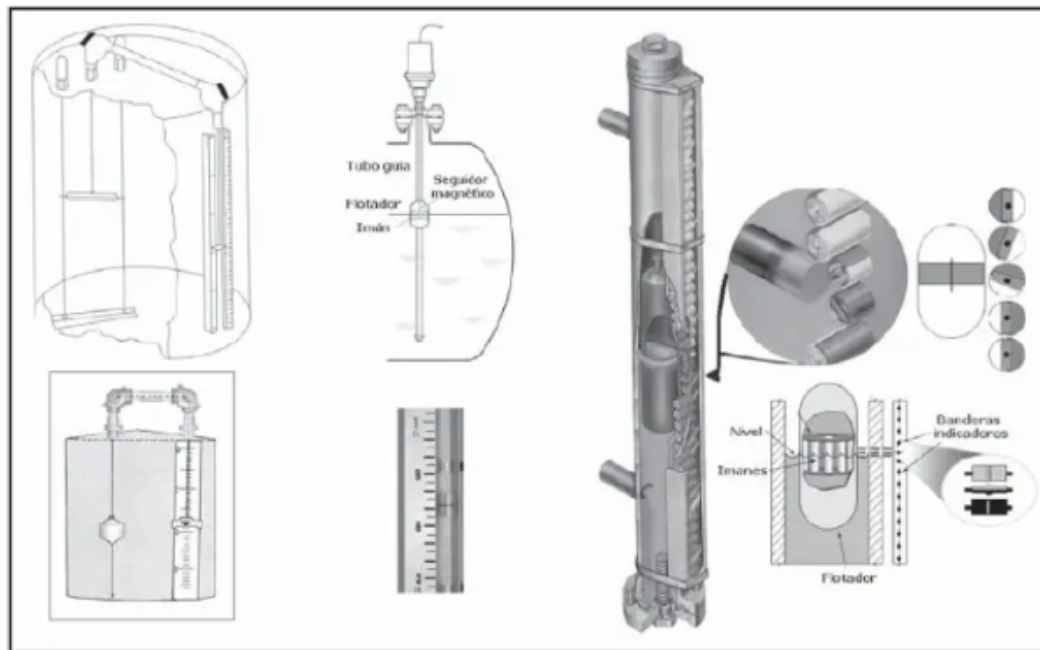


Figura 2.10: Sensor de flotador

Tomado de: [45].

2.8.4. Sensor de nivel por radar de onda guiada

Utiliza pulsos de microondas que son emitidos hacia la superficie del agua. La distancia entre el sensor y el líquido se calcula midiendo el tiempo que tarda el sensor en recibir un eco del pulso después de que la antena haya enviado el pulso de microondas. Las ondas microondas utilizadas en este tipo de sensor viajan a la velocidad de la luz y no requieren un medio de propagación específico. Esto significa que el sensor puede realizar mediciones de nivel sin verse afectado por factores como polvo, viento o temperatura [46].

En la figura 2.11 se ilustra como puede ubicarse este tipo de sensores para realizar su correcta medición y como se reflejan sus ondas por medio de flechas [47].

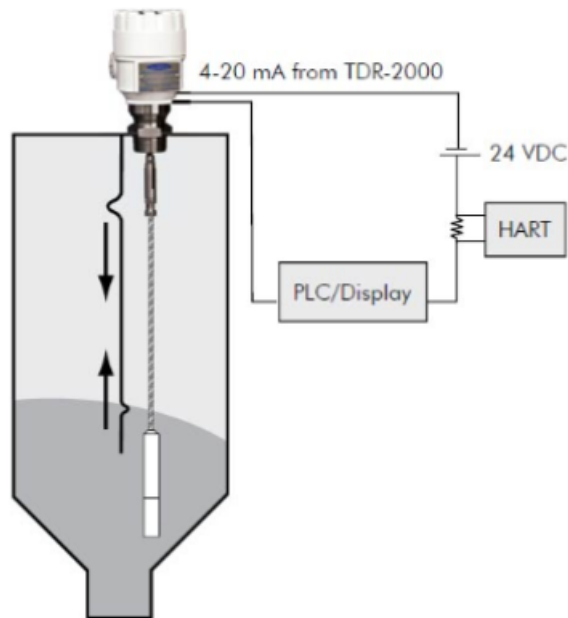


Figura 2.11: Sensor de radar de onda guiada
Tomado de: [47].

En el próximo capítulo, se establecen los criterios que orientaron la selección de los elementos utilizados en la realización de este proyecto. Se analiza en detalle los factores que influyeron en la elección de los componentes y tecnologías específicas. Además, se identifica y destaca aquellos elementos que se han demostrado como los más adecuados para satisfacer las necesidades y los objetivos específicos de este proyecto.

Capítulo 3

CRITERIOS DE SELECCIÓN

En este capítulo, se definen los criterios que han orientado la selección de tres componentes fundamentales en la concepción y desarrollo del presente proyecto.

El objetivo primordial consiste en realizar un prototipo de medición de nivel que sea económico para la empresa sin sacrificar la robustez necesaria para garantizar un funcionamiento efectivo.

Se abordarán los aspectos clave que han influido en la elección de estos componentes, destacando la importancia de conciliar la eficiencia económica con la durabilidad y confiabilidad del sistema. La búsqueda de una solución que combine un costo reducido con la capacidad de resistir las demandas operativas inherentes al entorno de aplicación es esencial para llevar a cabo la elección de los componentes como lo son: El protocolo de comunicación, el tipo de sensor y el microcontrolador.

3.1. Elección Protocolo de comunicación

En la tabla [3.1](#) se evalúan los criterios para la selección del protocolo de comunicación, para esto es necesario calificar cada requisito junto con cada opción que se tiene para realizar el proyecto. El rango de puntuación va desde 0-10, después de evaluar todos los protocolos se suma el resultado para cada uno de ellos y aquel que obtenga la puntuación más alta es el que mejor se adapta al proyecto.

CRITERIO	Zig-Bee	Ethernet	Lora	Wi-Fi	Modbus	LoraWan	Bluetooth
ALCANCE: Es necesario que tenga un alcance mínimo de 2.3 km de distancia.	4	10	10	3	10	10	3
COSTO: La implementación del protocolo de comunicación debe ser de bajo costo.	10	4	10	4	4	5	10
INTEROPERABILIDAD: El protocolo debe ser compatible con diferentes plataformas.	10	10	10	10	10	10	3
PENETRACIÓN DE OBSTÁCULOS: El lugar donde se desarrolla el proyecto es transitado por carros grandes, además cuenta con grandes árboles y montañas por lo tanto debe ser capaz de penetrar los obstáculos que se presenten.	6	10	8	10	8	8	2
INFORMACIÓN EN LA WEB: Es necesario que en la Web se encuentre información para el uso correcto del protocolo.	10	10	10	10	10	10	8
LATENCIA: Uno de los requisitos que tiene la empresa es que la información sea indicada al usuario en tiempo real.	7	10	7	10	8	8	3
Total:	47	54	55	47	50	51	29

Tabla 3.1: Criterios selección Protocolo de comunicación

Elaboración propia.

Siguiendo los criterios de selección mencionados anteriormente, el protocolo más adecuado para la ejecución de este proyecto es LoRa. Esta tecnología se destaca por su largo alcance, su enfoque en la transmisión de paquetes pequeños de datos y su eficiente consumo de energía. LoRa opera en diversas bandas de frecuencia, como 433, 868 y 915 MHz, lo que brinda flexibilidad en términos de configuración [22].

Las características esenciales que debe cumplir la tecnología para el proyecto incluye un alcance de 2.3 km, debido a la ubicación del punto de medición (Bocatoma). Es crucial que los módulos sean económicos y accesibles en el mercado para posibilitar la creación de un prototipo de bajo costo para la empresa. Además, se valora la disponibilidad de información detallada en la web sobre la configuración y programación del módulo, agilizando así el desarrollo del proyecto. Estos requisitos son clave para garantizar la eficiencia y el éxito del prototipo y su implementación.

Los puntos claves para implementar el prototipo diseñado son: la bocatoma, un punto intermedio del canal y el tanque de carga, según se ilustra en la figura 3.1.

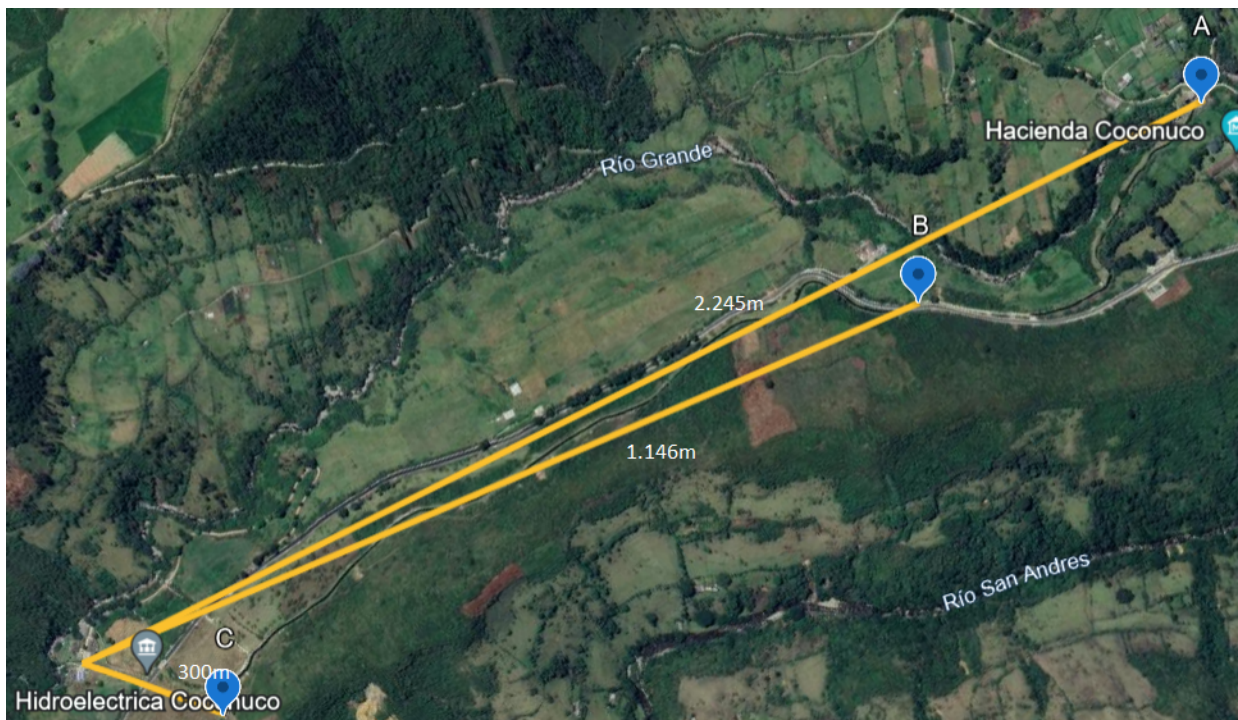


Figura 3.1: Ubicación de los puntos de medida.

Fuente: Google Earth.

Se propone la implementación de un sistema de medición mediante la utilización de la tecnología LoRa presente en la figura [3.2](#). En este sistema, se realiza la medición del nivel en puntos específicos a través de sensores dedicados. Cada punto de medición cuenta con un sensor cuya señal se transmite a un microcontrolador el cual cuenta con un módulo LoRa E32-433T30D, encargado de emitir una señal de radiofrecuencia hacia un módulo LoRa E32-433T30D ubicado en la planta, actuando como el nodo maestro del sistema. Este nodo maestro se encarga de recibir la información proveniente de todos los módulos esclavos distribuidos en los diferentes puntos de medición.

Con el propósito de la centralización y visualización eficientemente de los datos recopilados, la información obtenida en cada punto de medición será transmitida al nodo maestro y, posteriormente, cargada en un servidor en la web. Este enfoque permitirá que el operario tenga acceso a la información en todo momento, facilitando así la supervisión y gestión del sistema de medición en tiempo real.

La distancia de cada módulo frente a la planta son las siguientes: El módulo de la bocatoma se encuentra a 2.245 metros alejado de la planta, el punto medio en el canal donde se desea realizar la medición está a 1.146 metros y por último el tanque se encuentra a 300 metros de la planta.

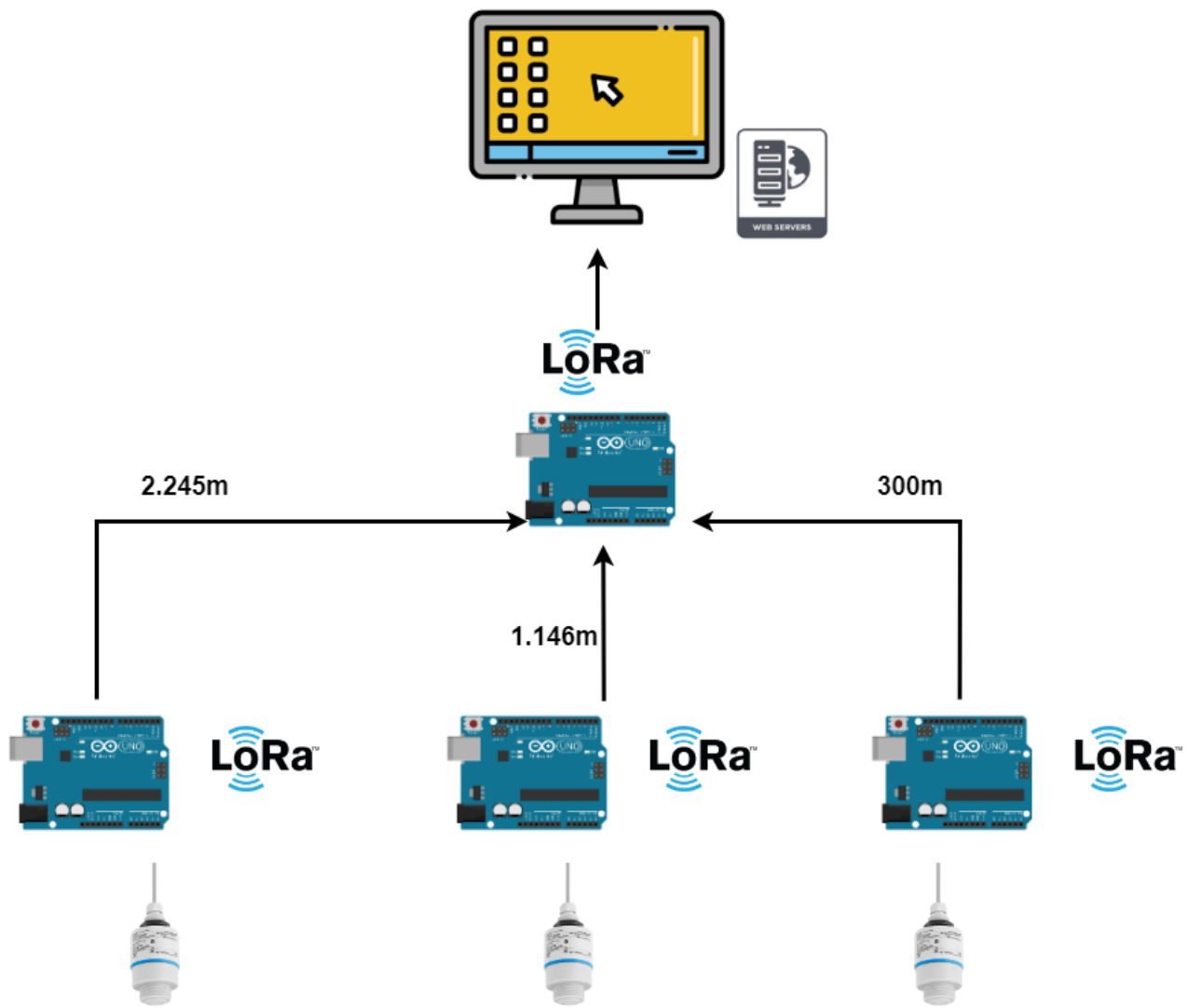


Figura 3.2: Sistema propuesto con LoRa.
Elaboración propia.

3.1.1. Implementación con LoRa

Antes de trabajar con este protocolo de comunicación se recibió una asesoría con el ingeniero Victor Quintero el cual proporcionó la simulación en la herramienta Matlab de la cobertura del sistema teniendo en cuenta las coordenadas de los puntos donde se desea instalar los equipos con el fin de saber si los datos pueden ser enviados desde el punto de medición hasta la planta donde se encuentra el módulo principal encargado de recibir la información de los otros dispositivos LoRa.

A continuación se evidencia el código proporcionado por el ingeniero Victor Quintero donde se establece la frecuencia, potencia y sensibilidad del módulo, las coordenadas del punto de medición, el tipo de antena conectada al LoRa, su medida en metros, y además se ingresan las coordenadas de la planta donde se reciben todos los datos.

```

1 clear all;
2 close all;
3 % sitio de transmisi n
4 fq = 900e6; % 150 M
5 tx = txsite("Name","Sensor", ...
6     "Latitude",2.356860, ...
7     "Longitude",-76.494779, ...
8     "Antenna",design(monopole,fq), ...
9     "AntennaHeight",1.5, ... % Units: meters
10    "TransmitterFrequency",fq, ... % Units: Hz
11    "TransmitterPower",1); % Units: Watts %30 dBm y hasta 23 dBi de ganancia
12 %tx.Antenna = monopole; %antena monopolo
13
14 %Define Receiver Sites
15 % Define receiver sites in several surrounding towns and cities
16 rxNames = ["Planta"];
17 rxLocations = [2.376922 -76.492336]; % Planta
18
19 % Define receiver sensitivity. Sensitivity is the minimum signal strength in
20 % power that is necessary for the receiver to accurately detect the signal.
21 rxSensitivity = -110; % Units: dBm
22 rxs = rxsite("Name",rxNames, ...
23     "Latitude",rxLocations(:,1), ...
24     "Longitude",rxLocations(:,2), ...
25     "Antenna",design(monopole,tx.TransmitterFrequency), ... %antena directiva en recepci n tipo Yagi
26     "ReceiverSensitivity",rxSensitivity); % Units: dBm

```

Después de ingresar los datos necesarios, el siguiente paso consiste en representar gráficamente la cobertura en función de la información proporcionada. El código necesario para visualizar esta información de manera clara y detallada se indica más adelante donde se establece un mapa base con el comando `viewer.Basemap`, se calcula la cobertura de red con el modelo de propagación 'Longley-Rice' considerando los niveles de intensidad de la señal con `"SignalStrengths",rxSensitivity'`, por otro lado el comando `'link'` representa la comunicación entre los transmisores (tx) con los receptores(rxs), finalmente se configura un mapa de

cobertura contorneada indicando los niveles de intensidad de la señal con 'coverage()'.

```
1 % Tilt antenna to direct radiation in XY-plane (i.e., geographic azimuth)
2 %Show Sites on a Map
3 viewer = siteviewer;
4 show(tx);
5 show(rxs);
6 viewer.Basemap = "openstreetmap";
7 %pattern(tx); %mostrar patr n de radiaci n
8 % % Display Coverage Map using monopole
9 coverage(tx,"Longley-rice", ...
10         "SignalStrengths",rxSensitivity)
11 %Plot Communication Links using Monopole Antenna
12 viewer = siteviewer;
13 show(tx);
14 show(rxs);
15 viewer.Basemap = "openstreetmap";
16 sc = [0 0.3 0];
17 link(rxs,tx,"Longley-rice","SuccessColor",sc)
18 % %Display Contoured Coverage Map using Multiple Signal Strengths
19 %
20 % % Define signal strengths from sensitivity to -60 dB
21
22 viewer = siteviewer;
23 show(tx);
24 show(rxs);
25
26 viewer.Basemap = "openstreetmap";
27 sigStrengths = rxSensitivity:5:-60;
28 % % Update coverage map
29 coverage(tx,"longley-rice","SignalStrengths",sigStrengths)
```

En la primera ventana la cual se indica en la figura [3.3](#), se proporciona una representación visual de la cobertura en un nivel general, se observa de manera gráfica la ubicación del punto emisor y el punto receptor.

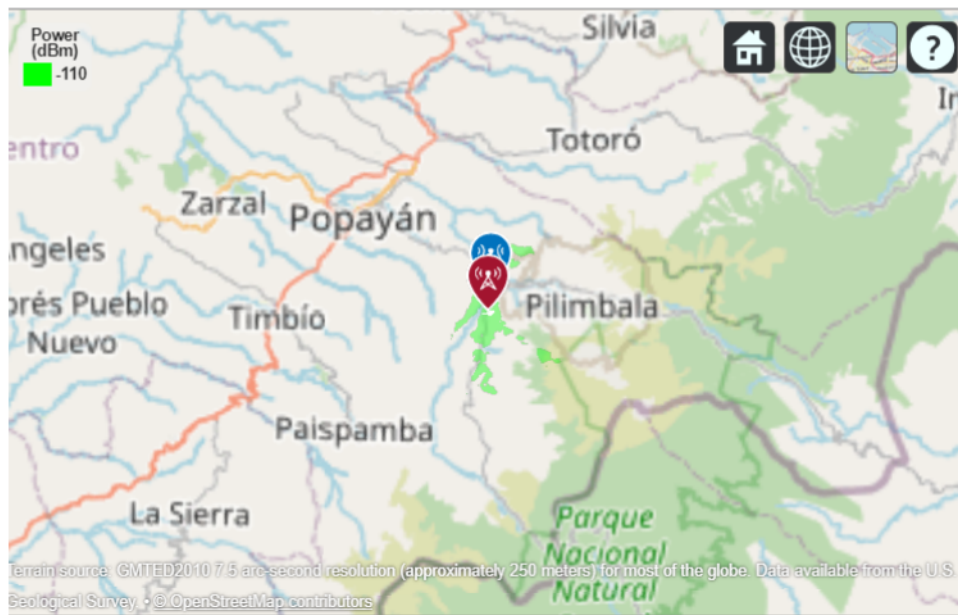


Figura 3.3: Cobertura a nivel general
Elaboración propia.

En la segunda ventana representada en la figura [3.4](#), se realiza una evaluación para determinar la viabilidad del enlace. La línea trazada en color rojo indica que el enlace no es posible, mientras que si aparece en verde, significa que la transmisión de información es viable.

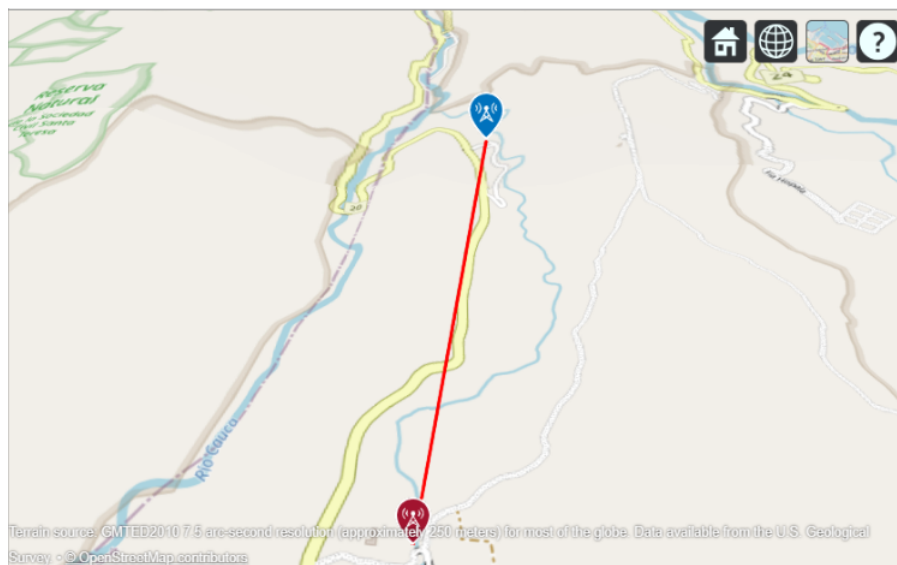


Figura 3.4: No viabilidad del enlace
Elaboración propia.

En la última visualización, se representa la cobertura utilizando colores de intensidad variable para indicar diferentes niveles de potencia. Los colores más intensos representan un mayor nivel de potencia en la cobertura, esto se puede observar en la figura 3.5.

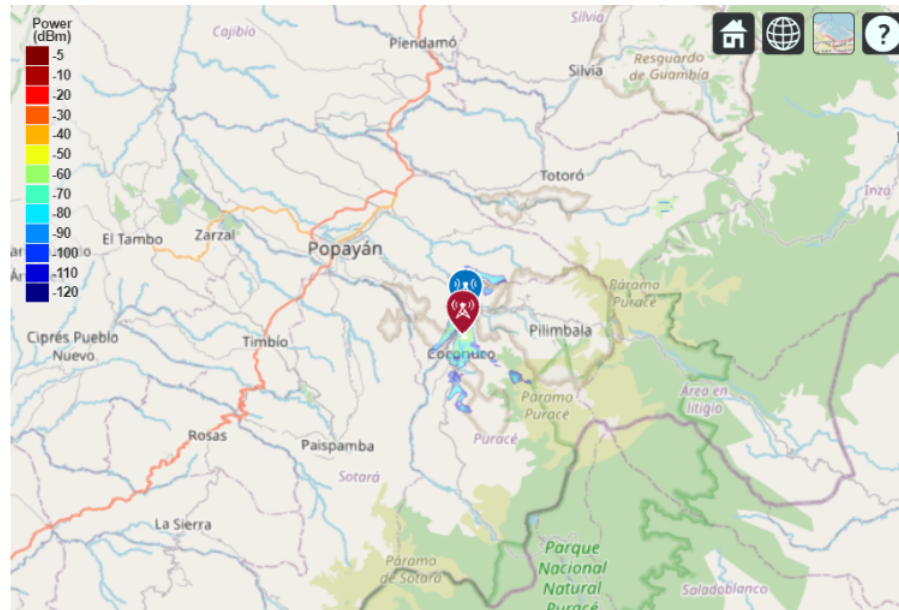


Figura 3.5: Niveles de potencia
Elaboración propia.

De acuerdo a la información previamente obtenida, es posible concluir que los módulos con una frecuencia de 900 MHz, una potencia de 1 dBm y una sensibilidad de -110 dBm no son capaces de establecer una comunicación efectiva y, por lo tanto, no son adecuados para llevar a cabo la transmisión de datos.

Debido a esta limitación, se procedió a ingresar diferentes valores con el propósito de identificar las características necesarias en los módulos LoRa que garanticen una cobertura óptima para su efectiva transmisión de datos.

Después de realizar las simulaciones, se concluyó que la frecuencia de 433MHz es la adecuada para lograr una transmisión de datos exitosa. Esto se evidencia en el hecho de que, al compilar el código con este valor, la línea que indica la posibilidad de conexión se presenta en color verde, así como se ilustra en la figura 3.6.

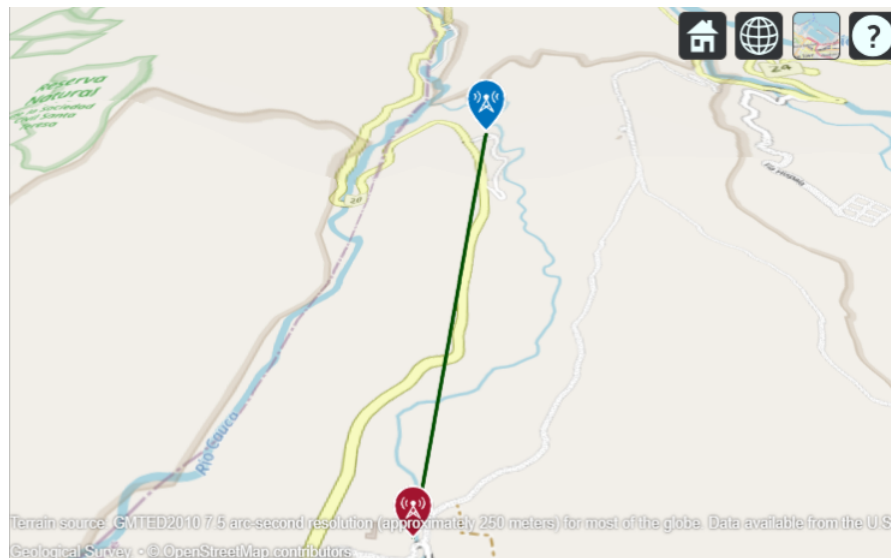


Figura 3.6: Viabilidad de cobertura
Elaboración propia.

Después de completar la simulación, se procedió a utilizar la herramienta Google Earth con el propósito de obtener un perfil detallado del terreno. El objetivo principal de esta etapa es evaluar el tipo de obstrucción presente en el área. Para llevar a cabo esta tarea, fue necesario recopilar con precisión las coordenadas geográficas tanto de la planta como del punto de medición del nivel del agua, en este caso es la bocatoma. Los resultados obtenidos se pueden observar a continuación en la figura [3.7](#), donde se observa una obstrucción significativa en la línea de visión.

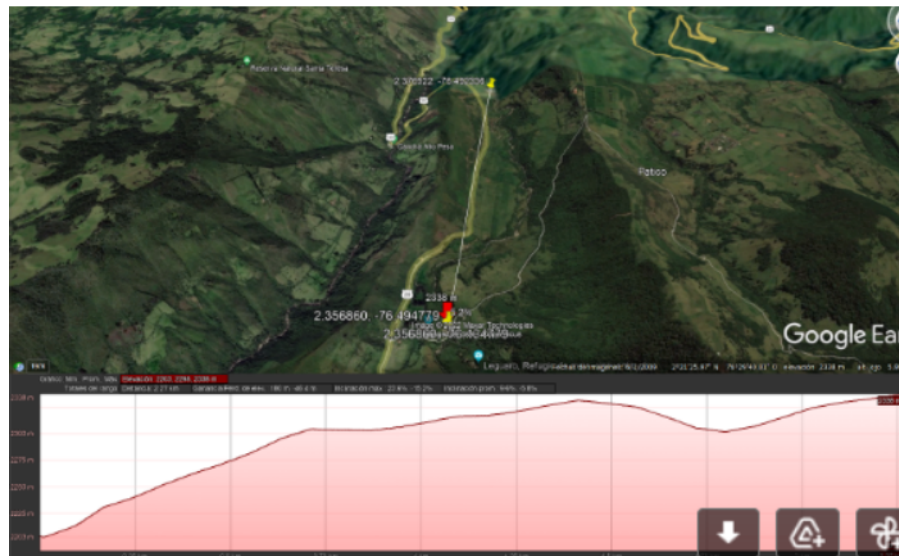


Figura 3.7: Perfil del terreno

Fuente: Google Earth.

Aún con los resultados obtenidos anteriormente con Google Earth se tomó la decisión de trabajar con este protocolo de comunicación ya que la simulación realizada en MATLAB fue exitosa.

Con el fin de llevar a cabo las pruebas necesarias, se procedió a adquirir los distintos dispositivos esenciales los cuales se pueden observar en la tabla [3.2](#), [3.3](#), [3.4](#). Además, se desarrolló la Placa de Circuito Impreso (PCB) en la herramienta Proteus, la cual permitió la interconexión de los elementos necesarios para la realización de las pruebas, el diseño se muestra en la figura [3.8](#).

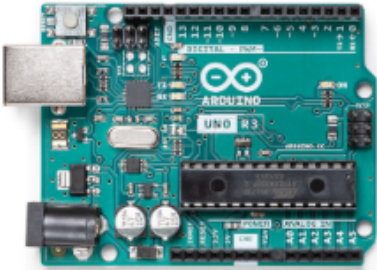


Compo.	Descripción	Cant	Imagen
Arduino UNO	Es una placa con un microcontrolador Atmega 328, cuenta con 14 pines digitales donde 6 de estos se pueden usar como PWM, también tiene 6 pines analógicos, el voltaje de trabajo es de 5V, tiene una memoria EEPROM de 1 KB, se puede conectar a un computador haciendo uso de un cable USB o se puede alimentar con un cargador o una batería, este tipo de microcontrolador está siendo de los más usados por arduino por sus capacidades y su pequeño tamaño, sus medidas son 68.6 mm de longitud y 53.4 mm de ancho [48].	3	 <p>Tomado de: [48].</p>
HW-685	Este módulo transforma la corriente en voltaje, cuenta con dos potenciómetros con los cuales se puede calibrar el valor mínimo y máximo donde generalmente 4mA significa que la variable medida está en 0 y 20mA la variable se encuentra en su valor máximo, su pequeño tamaño (42mm x 25mm x 10mm) permite un fácil manejo por parte del usuario [49].	3	 <p>Tomado de: [49].</p>
LCD 16X2	Es una pantalla LCD que se utiliza en diferentes aplicaciones, cuenta con diferentes pines los cuales se deben manipular para controlar la pantalla y así escribir en ella lo que se desea mostrar al usuario, también se puede configurar el contraste de pantalla con el pin Vo, además tiene sus pines de alimentación 5V y GND. [50].	3	 <p>Tomado de: [50].</p>

Tabla 3.2: Primera parte Dispositivos para trabajar con LoRa.

Elaboración propia.

Compo.	Descripción	Cant	Imagen
DS3231	<p>Es un reloj en tiempo real (RTC), soporta un bus I2C bidireccional, cuenta con una entrada para una pila lo cual permite que cuando la alimentación se interrumpa la hora no se reinicie, en este reloj se tiene la información de los minutos, horas, segundos, día,año y mes, funciona con 3.3V [51].</p>	3	 <p>Tomado de: [51].</p>
LM2596	<p>Este módulo es un reductor de voltaje, permite una entrada de voltaje de 0-40V, cuenta con un potenciómetro el cual se debe calibrar para configurar el voltaje de salida, la corriente máxima es de 3A, es un componente de bajo costo pero aún así su conversión es eficiente con un promedio de 88 % de satisfacción [52].</p>	3	 <p>Tomado de: [52].</p>
FMR10	<p>Este sensor de nivel de radar de onda guiada es fabricado por Endress + Hauser, permite la medición de nivel sin contacto, este dispositivo cuenta con bluetooth, la configuración de los parámetros de este instrumento se debe hacer por medio de la aplicación denominada “Endress+Hauser SmartBlue” y por esta misma se puede observar el valor medido. Puede estar en temperaturas de -40 a -60 °C, el valor máximo de medición es de 12m, su precisión es de ±5 mm y su frecuencia de medición es de 26GHz [53].</p>	3	 <p>Tomado de: [53].</p>

Tabla 3.3: Segunda parte Dispositivos para trabajar con LoRa
Elaboración propia.

Compo.	Descripción	Cant	Imagen
MDR-60-24	Es una fuente de alimentación industrial en riel la cual cuenta con protección para cortocircuito, sobrecarga y sobretensión, esta versión de MDR-60 permite un voltaje de salida de 24V y una corriente de salida de 2.5A, su potencia es de 60W, cuenta con un LED el cual indica si la fuente está encendida [54].	3	 <p>Tomado de: [54].</p>
E32-433T30D	Este módulo se basa en el chip RF SX1278 de SEMTECH y opera en un rango de frecuencia de 410-441 MHz. Tiene un alcance de distancia de hasta 8 kilómetros, lo que lo hace ideal para aplicaciones de larga distancia. Tiene una potencia de transmisión máxima de 1W, se caracteriza por un bajo consumo de energía, lo que lo convierte en una opción eficiente en términos de consumo energético [55].	4	 <p>Tomado de: [55].</p>
Antena Mono-polo	Las antenas monopolo tienen una amplia aplicación en sistemas de comunicación y radiodifusión. Derivadas del diseño del dipolo, estas antenas se componen de un único brazo rectilíneo que es responsable de propagar las ondas electromagnéticas. Su diseño simplificado permite una movilidad conveniente, lo que las convierte en una opción versátil que puede ser desplegada en diversas ubicaciones. [56].	4	 <p>Elaboración propia.</p>

Tabla 3.4: Última parte Elementos adquiridos para trabajar con LoRa
Elaboración propia.

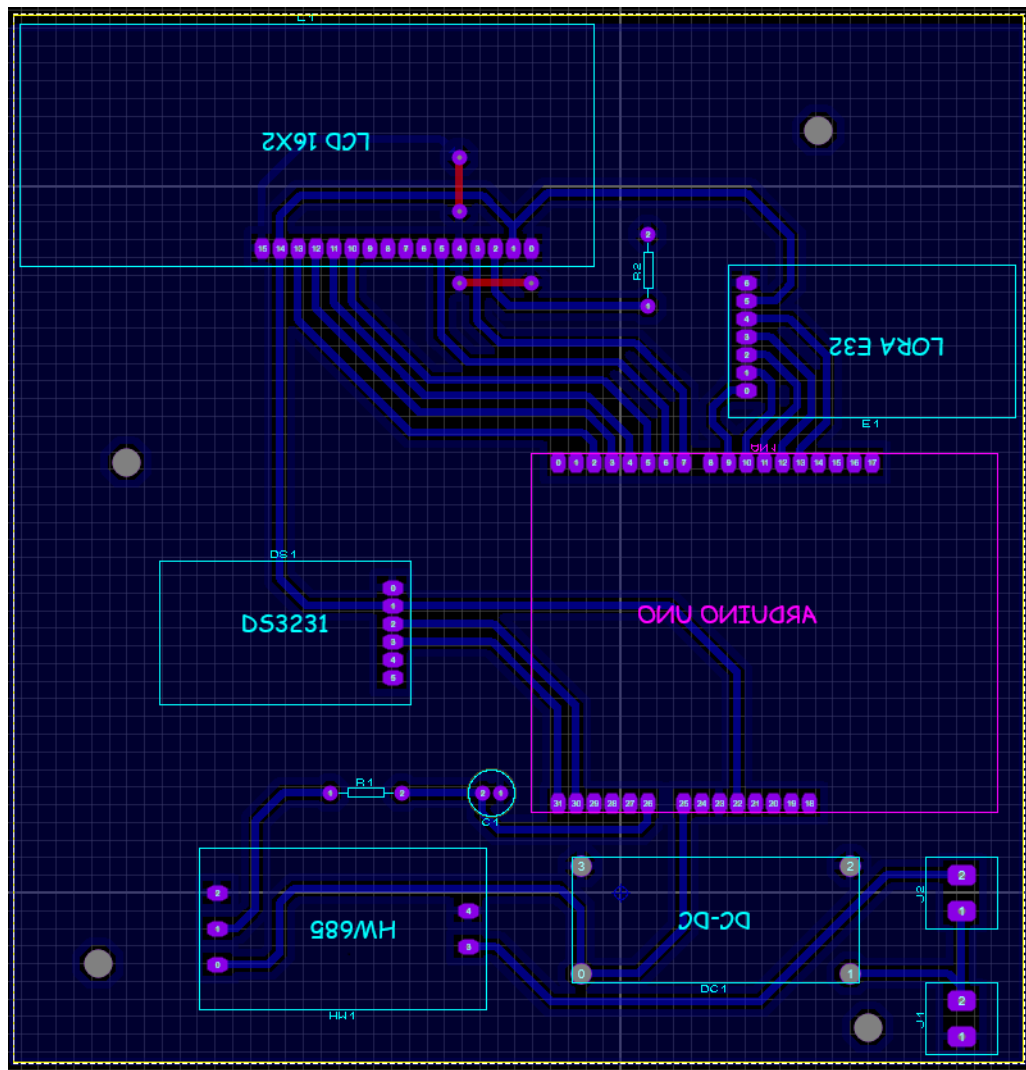


Figura 3.8: Diseño en proteus de placa con LoRa
Elaboración propia.

Las placas se imprimieron en el laboratorio de la Universidad del Cauca en la Facultad de ingeniería Electrónica. Para utilizarlas de manera efectiva, se llevó a cabo una soldadura inicial que incluyó tres resistencias, un capacitor ceramico tipo lenteja, así como pines hembra y macho para ensamblar la LCD, el arduino, el E32-433T30D, el HW685, el LM2596 y el DS3231, así como se muestra en la figura [3.9](#). Esto se hizo con el propósito de simplificar la conexión y desconexión de los módulos asociados.

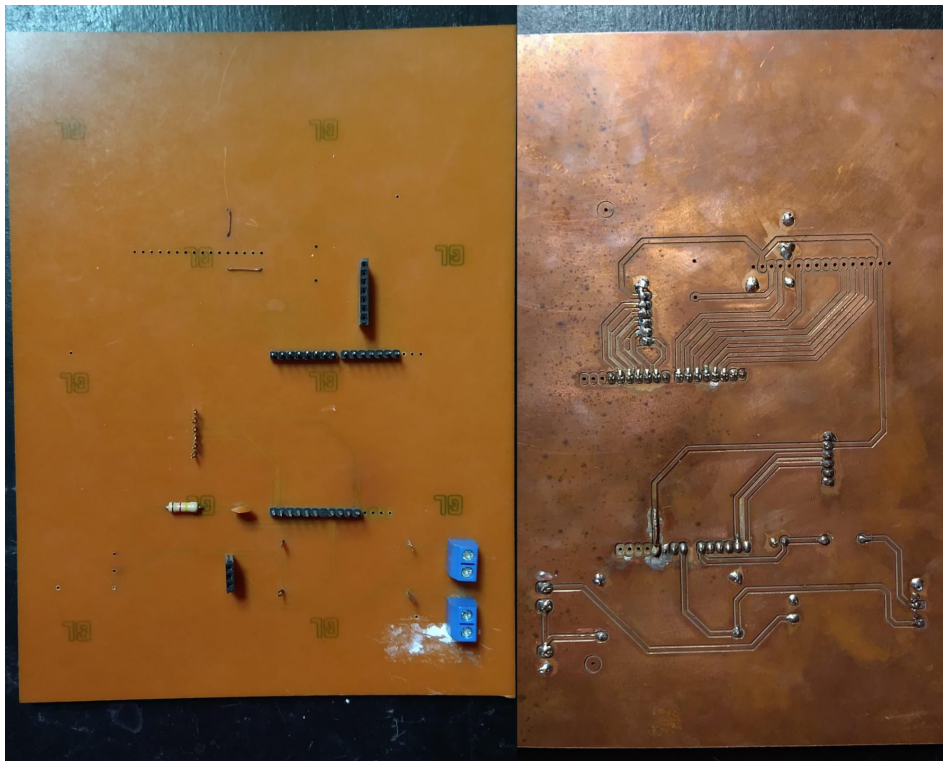


Figura 3.9: Placa soldada
Elaboración propia.

Luego de la soldadura inicial, se procedió a la inserción de los componentes anteriormente mencionados. Una vez que la placa contiene todos los elementos necesarios, se subió el código en el Arduino para comprobar la comunicación y la estructura se puede observar a continuación. En dicho código, se incluyeron las librerías 'SoftwareSerial.h' y 'EByte.h', se definieron las variables correspondientes como una estructura 'DATA' conformada por 'id' y 'Conta'. Además, se declararon los pines para la transmisión (Tx) y recepción (Rx) de LoRa y se define la variable 'millis_tiempo'.

```
1 #include <SoftwareSerial.h>
2 #include "EByte.h"
3 struct DATA { // trama de comunicacion en el uplink
4     unsigned long Conta; int id; };
5 DATA MyData;
6 SoftwareSerial ESerial(12, 11); //pines de Tx y Rx
```

```
7 EBYTE Transceiver(&ESerial, 9, 10, 13);
8 unsigned long millis_tiempo;
```

Después se Configuran todos los parámetros relacionados con LoRa, incluyendo su inicialización, potencia, velocidad, tasas de datos y modo de comunicación entre los módulos.

```
1 void setup() {
2   pinMode(9, OUTPUT); //pin M0 lora
3   pinMode(10, OUTPUT); //pin M1 lora
4   pinMode(13, INPUT); // pin AUX lora
5   Serial.begin(9600); //inicializa serial para imprimir
6   ESerial.begin(9600); //inicializa serial lora
7   Serial.println("Starting Reader"); //imprime
8   Transceiver.init(); //inicializa la tarjeta lora
9   Transceiver.SetMode(MODE_PROGRAM); // modo programado
10  Transceiver.SetTransmitPower(OPT_TP24); // pot. transmisi n
11  Transceiver.SetUARTBaudRate(UDR_9600); // Tasas de datos UART
12  Transceiver.SetAirDataRate(ADR_2400); //Velocidad de datos
13  Transceiver.SetAddressH(0);
14  Transceiver.SetAddressL(0);
15  Transceiver.SetChannel(13); // define canal 13
16  Transceiver.SaveParameters(PERMANENT);
17  Transceiver.Reset();
18  ESerial.flush();
19 }
```

Se crean dos funciones denominadas 'transmitir' la cual cuenta con un contador y un id y se llama el método para enviar la información, estos datos se pueden observar en el serial, y la función 'recibir' está diseñada para recibir la información previamente enviada por la función 'transmitir', estos datos recibidos se imprimen en el serial.


```
1 void transmitir() {
2   MyData.Conta++; // aumentar el contador de mensajes
3   MyData.id = 1; // id de la estacion movil
4   Transceiver.SendStruct(&MyData, sizeof(MyData)); //enviar datos
5   Serial.print("Enviando:id= "); Serial.println(MyData.id); Serial.
   print(", "); Serial.print("dato= "); Serial.print(MyData.Conta)
   ;
6   return;}
7 void recibir() {
8   if (ESerial.available()) { //lee el puerto serial
9     Transceiver.GetStruct(&MyData, sizeof(MyData));
10    Serial.println("Recibiendo: id= ");
11    Serial.print(MyData.id); Serial.print(", "); Serial.print("dato
   = "); Serial.print(MyData.Conta); return;}
12   else { // sino hay datos para recibir, quedese escuchando
13     delay(500);
14     //Serial.println("<----Escuchando---->");
15     return;}}
```

La programación de la configuración de variables y el setup previamente descrita es esencial tanto en el módulo emisor como en el receptor. Cada uno desempeña un papel específico: mientras el módulo emisor está encargado de transmitir información, el módulo receptor se dedica a recibir dicha información. Es por esto que en el void loop del módulo emisor, se implementa la función 'transmitir', la cual envía datos en intervalos de 2 segundos.

```
1 void loop() {
2   if (millis() > millis_tiempo) {
3     millis_tiempo = millis() + 2000ul;
4     transmitir();
5   }}
```

Y en el modulo receptor se hace el llamado de la función 'recibir' pero antes se abre la comunicación con el serial de LoRa.

```
1 void loop() {  
2   ESerial.listen();//abre serial de lora  
3   recibir();  
4 }
```

Los resultados obtenidos durante la prueba realizada en laboratorio fueron satisfactorios, dado que los mensajes originados en el punto A fueron transmitidos con éxito y recibidos de manera adecuada en el punto B. Esta observación positiva señala la efectividad de la comunicación de datos en esta configuración particular, lo que respalda el funcionamiento exitoso del sistema, en la figura 3.10 se muestra como los datos fueron enviados y recibidos.

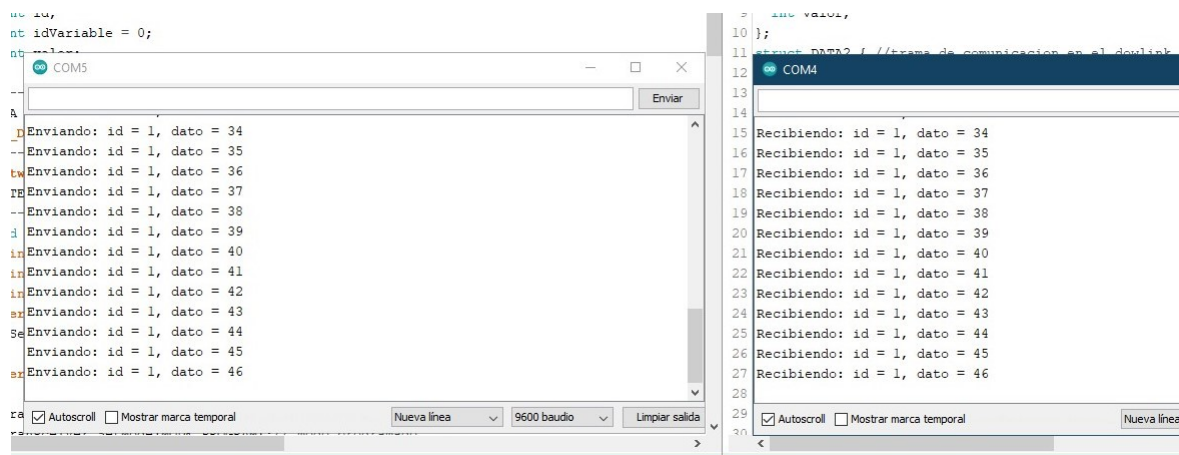


Figura 3.10: Prueba exitosa de envío de datos
Elaboración propia.

Luego, se decidió llevar a cabo el desplazamiento hasta las instalaciones de la empresa municipal de energía eléctrica S.A E.S.P. Esto con el fin de confirmar la capacidad de envío de datos en el entorno real donde se ubicaría el sistema, en la figura 3.11 se puede observar la prueba que se realizó en la planta.



Figura 3.11: Antena instalada en la bocatomá
Elaboración propia.

Al llevar a cabo pruebas en distintas ubicaciones, se tuvo como resultado que el sistema es capaz de alcanzar una distancia máxima de 500 metros para la transmisión de datos, la figura [3.12](#) muestra que la comunicación es intermitente ya que no se reciben los mensajes cuando los puntos A y B están muy separados, superando la distancia de 500 metros, pero se reestablece cuando los puntos están nuevamente cerca. Como resultado de este análisis, se ha tomado la decisión de descartar la posibilidad de utilizar la tecnología LoRa en el desarrollo del proyecto, dado que esta distancia no cumple con los requisitos de cobertura necesarios para la exitosa implementación.

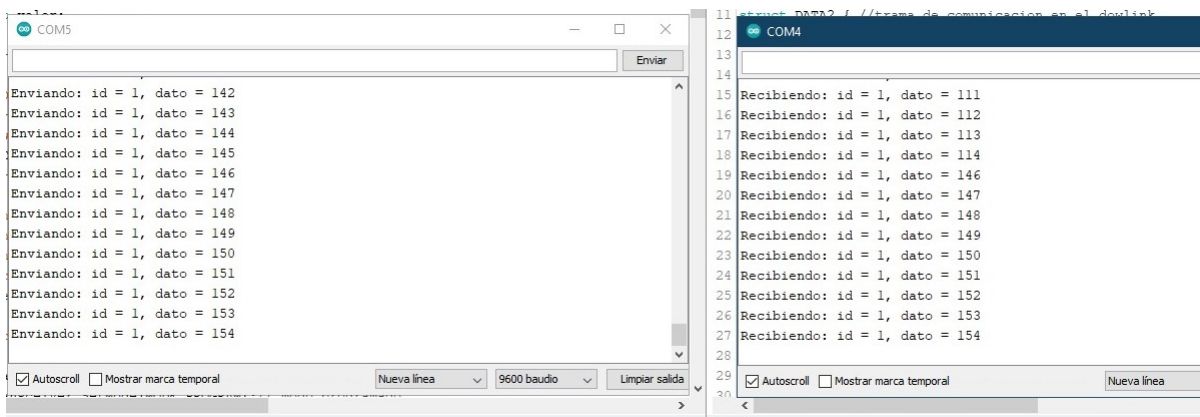


Figura 3.12: Prueba fallida en la planta
Elaboración propia.

Es importante mencionar que en la configuración de los dispositivos no existe una línea de visión directa entre ellos, lo que da lugar a la formación de una zona conocida como la 'Zona de Fresnel'. La cual comprende un espacio alrededor de la línea de visión y cualquier obstrucción que se encuentre dentro de esta área tiene un efecto perjudicial en la señal transmitida, lo que a su vez impacta negativamente en la calidad de la comunicación. Esta información es esencial para comprender las limitaciones y desafíos asociados con la comunicación en este contexto [57].

Luego de las pruebas fallidas con LoRa, se decidió implementar Ethernet como la tecnología de elección para este proyecto ya que se tuvo en cuenta la tabla de criterios de selección donde se establece que Ethernet se ubicó como el segundo protocolo que mejor se adapta a las necesidades específicas del proyecto.

La decisión de optar por Ethernet después de descartar LoRa implica utilizar una tecnología que opera en la capa física, donde se manejan las señales y los componentes físicos necesarios para la transmisión de datos. Ethernet reconoce streams de bits y, en la capa de enlace de datos, su subcapa de Control de Acceso al Medio (MAC) se preparan los datos para su transmisión. Esto incluye la construcción de tramas antes de la transmisión, donde se agrega un encabezado y un tráiler a la Unidad de Datos del Protocolo (PDU) de la capa 3.

La encapsulación de datos en Ethernet también permite el direccionamiento correcto, ya que

los encabezados contienen la dirección física, lo que garantiza que el mensaje sea enviado al destino correcto. Además, se puede verificar la integridad de la trama mediante la Comprobación Cíclica de Redundancia (CRC). Esta técnica implica que el receptor genere una CRC y la compare con la CRC recibida en la trama. Si los valores coinciden, se considera que la trama se ha recibido sin errores.

Finalmente, la subcapa Control de Enlace Lógico (LCC) en Ethernet se encarga de conectarse con las capas superiores y toma los datos del protocolo de red, agregando la información necesaria para que los datos sean entregados con éxito a su destino. La elección de Ethernet implica un enfoque más robusto en la comunicación, que es especialmente adecuado cuando se necesita una mayor confiabilidad en la transmisión de datos.

Ethernet ha evolucionado en el tiempo para adaptarse a las necesidades que han ido surgiendo, en un inicio transportaba datos a 3Mbps y en la actualidad lo puede hacer a 10Gbps, además se ha implementado a través de varias tecnologías tanto en cobre como en fibra óptica, convirtiéndolo en el protocolo Red de área local (LAN) que mas se está utilizando a nivel mundial.

Este medio de comunicación ha tenido mucho éxito ya que su mantenimiento es fácil, simple, confiable y de bajo costo para su instalación y actualización.

El tiempo necesario para que un bit se detecte en el medio, en Ethernet de 10 Mbps es de 100ns, de 100 Mbps requiere 10ns y 1000Mbps solo es necesario 1ns para transmitir un bit, se tiene una medida aproximada de 20.3 centímetros por nanosegundo para conocer el retardo en la transmisión con un cable UTP [58].

3.2. Elección Microncontrolador

Para la selección del microcontrolador que mejor se ajuste a las necesidades del proyecto, se llevó a cabo una evaluación considerando varios criterios clave. Entre estos se tiene la robustez, el costo, la disponibilidad de documentación, y el rendimiento del procesador, a cada uno de ellos se le otorga una puntuación de 0-10 y al final se suma el total de cada microcontrolador, el dispositivo con la puntuación más alta es la opción más óptima para

el prototipo. Esta decisión se fundamenta en su capacidad para satisfacer de manera más efectiva las necesidades específicas del proyecto, garantizando así un rendimiento robusto y eficiente, el resultado se observa en la tabla [3.5](#).

CRITERIO	Arduino	ESP32	Raspberry PI	PIC
ROBUSTEZ: Es necesario un dispositivo robusto ya que el armario donde se conectará el microcontrolador es al aire libre y estará funcionando todo el tiempo.	8	6	9	10
COSTO: Debe ser un microcontrolador de bajo costo en el mercado y aún así sea capaz de realizar las tareas necesarias del proyecto.	10	10	6	8
DOCUMENTACIÓN: El proyecto se quedará en la empresa por lo tanto debe ser fácil de interpretar y manipular por otra persona si es necesario, esto implica que en la Web se encuentre información suficiente sobre su configuración y programación, además que en el mercado se puedan encontrar diferentes módulos que permita diversas funcionalidades.	10	8	10	8
PROCESADOR: Aunque el procesador no ejecutará tareas intensivas si se requiere que sea un buen procesador capaz de soportar las actividades a desarrollar.	8	10	10	9
TOTAL:	36	34	35	35

Tabla 3.5: Criterios de selección microcontrolador
Elaboración propia.

De acuerdo con la tabla previamente analizada, el microcontrolador que mejor se adecua a las necesidades del proyecto es Arduino. Esta elección se fundamenta en su robustez, su costo asequible, la capacidad de su procesador para manejar las tareas requeridas, su carácter de código abierto y la abundante documentación disponible en cuanto a su programación, configuración y adaptabilidad a otros módulos. La combinación de estas cualidades hace que Arduino sea una opción sólida y versátil, permitiendo no solo cumplir con los requisitos actuales, sino también brindando la flexibilidad necesaria para posibles modificaciones y mejoras futuras a cargo del siguiente encargado del proyecto.

3.3. Elección Sensor

La selección del sensor para llevar a cabo este proyecto se basa en la consideración de cuatro criterios esenciales: la capacidad de operar sin contacto directo con el líquido, la precisión en la medición, el costo y la sensibilidad del sensor. Se tuvieron en cuenta cuatro tipos de sensores diferentes para ver cual de estos se adapta más a las necesidades, las opciones fueron: ultrasonido, radar de onda guiada, flotador y capacitivo, el puntaje utilizado es de 0-10, cuando se tiene toda la calificación se suma el total de cada sensor y el de mayor puntuación es elegido para ser utilizado en la realización del proyecto. Esta elección se basa en su capacidad para satisfacer de manera óptima las necesidades específicas del proyecto, garantizando mediciones precisas y confiables sin contacto directo con el líquido.

SENSOR				
CRITERIO	Ultrasonido	Radar onda guiada	Flotador	Capacitivo
NO CONTACTO: La condición más importante en la elección del sensor es que este no necesite contacto directo con el agua para realizar su medición ya que el líquido es captado de un río y trae consigo residuos que pueden dañar al sensor.	10	10	0	0
PRECISIÓN: Debe ser un sensor con una alta precisión capaz de proporcionar una medida confiable ya que con el valor obtenido se toman decisiones en el proceso.	8	10	5	7
PRECIO: El sensor adquirido no debe tener un costo elevado para de esta manera generar un prototipo de bajo presupuesto.	8	7	10	9
SENSIBILIDAD: El sensor debe ser capaz de detectar cambios pequeños en la medida y que esta capacidad no se vea afectada por condiciones ambientales ya que estará expuesto a dichas condiciones como viento, lluvia, polvo y otros factores.	8	10	10	9
TOTAL:	34	37	23	24

Tabla 3.6: Criterios de selección sensor de nivel
Elaboración propia.

Aunque se evaluaron cuatro tipos de sensores, se descartaron los sensores tipo flotador y capacitivo debido a un requisito fundamental: no debe estar en contacto directo con el líquido para realizar las mediciones necesarias. Con este criterio en mente, el sensor que mejor se ajusta a las necesidades del proyecto es el radar de onda guiada. Este sensor no requiere contacto con el agua para medir su nivel, es altamente preciso y su sensibilidad no se ve afectada por agentes externos.

En el próximo capítulo, se aborda la metodología que se empleó para el desarrollo exitoso del proyecto. Se detallan los procesos, las estrategias y las acciones implementadas para cumplir con los objetivos planteados inicialmente.

Capítulo 4

METODOLOGÍA

Para el proceso de generación de energía, la empresa cuenta con una planta ubicada a 26 km de la ciudad de Popayán. Esta planta está compuesta por una serie de componentes que permiten la transformación del agua en energía eléctrica.

El proceso comienza con una bocatoma ilustrada en la figura [4.1](#). Su función es proporcionar acceso al río Grande para obtener agua la cual es utilizada para la generación de energía.



Figura 4.1: Bocatoma de la planta
Elaboración propia.

El agua captada en la bocatoma comienza a recorrer un canal hasta llegar a dos tanques de

carga como lo indica la figura 4.2. Estos tanques están conectados a través de dos tuberías que conducen el fluido hasta la casa de máquinas.



Figura 4.2: Tanque de carga
Elaboración propia.

En la figura 4.3 se presenta las tuberías por donde desciende el agua y llega hasta las turbinas para convertir la energía cinética en mecánica.



Figura 4.3: Tuberías
Elaboración propia.

En la casa de máquinas se encuentran tres turbinas que están acopladas a generadores. Las turbinas convierten la energía cinética del agua en energía mecánica, mientras que los generadores transforman esa energía mecánica en energía eléctrica utilizable [59]. La energía eléctrica generada se transporta a través de la red de transmisión para su distribución, en la figura 4.4 se encuentra una turbina con su generador.

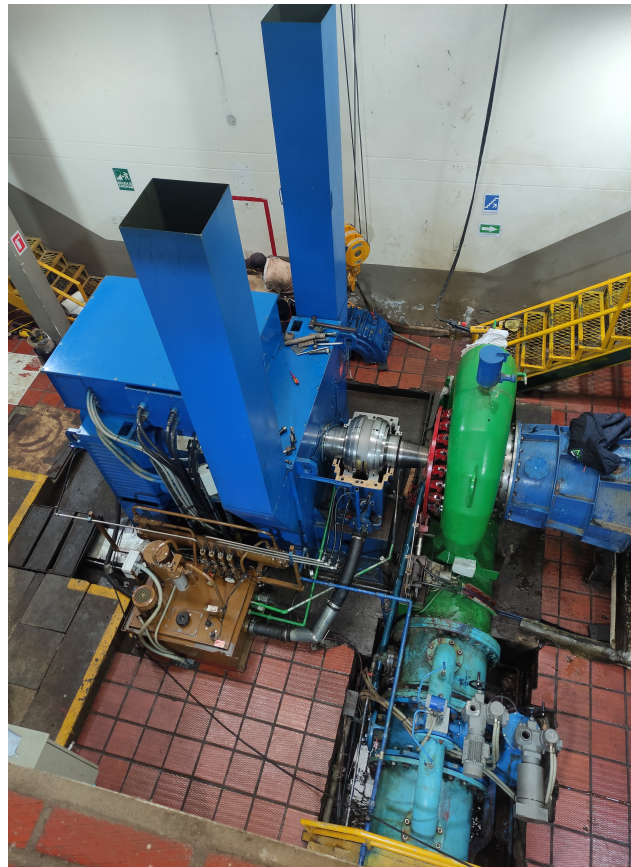


Figura 4.4: Turbina y generador en la planta
Elaboración propia.

A lo largo de este proceso de generación de energía, es necesario monitorear constantemente el nivel del agua presente en el canal y en el tanque de carga. Anteriormente, esta tarea era realizada por los operarios de turno, quienes debían trasladarse hasta los puntos de medición para verificar el nivel del agua. La comunicación de la información relacionada con el agua se realizaba entre operarios a través de dispositivos de radio, como walkie-talkies. Además, la persona presente en el cuarto de supervisión podía visualizar el estado mediante cámaras instaladas en la bocatoma y el tanque de carga.

El proceso de generación de energía se basa en la utilización del agua como materia prima para obtener energía eléctrica. En la figura 4.5, se muestra como era el proceso hasta antes del desarrollo del proyecto.

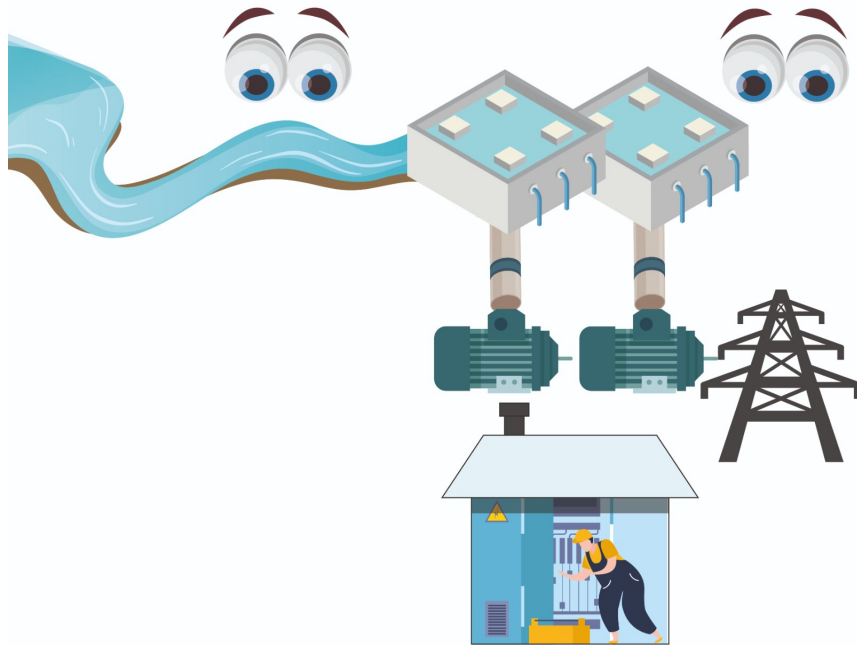


Figura 4.5: Proceso del funcionamiento de la planta
Elaboración propia.

4.1. Diseño

Se propone la implementación de un sistema de medición de nivel remoto utilizando sensores para cada punto de interés donde se desea conocer el nivel de agua. Cada sensor se conecta a un microcontrolador que permite enviar los valores medidos a un servidor web. El cual está encargado de proporcionar una interfaz al usuario que permite visualizar los datos recibidos de manera gráfica. Esta interfaz puede ser utilizada a través de un navegador web mediante una computadora o celular así como se indica en la figura 4.6.

Con este sistema de medición remota, los operarios ya no necesitan desplazarse físicamente hasta cada punto de medición para conocer el nivel de agua. En su lugar, abren la aplicación en su dispositivo móvil o computadora y tienen acceso a la información actualizada del nivel de agua en tiempo real. Esto permite un monitoreo más eficiente y reduce los

desplazamientos innecesarios.

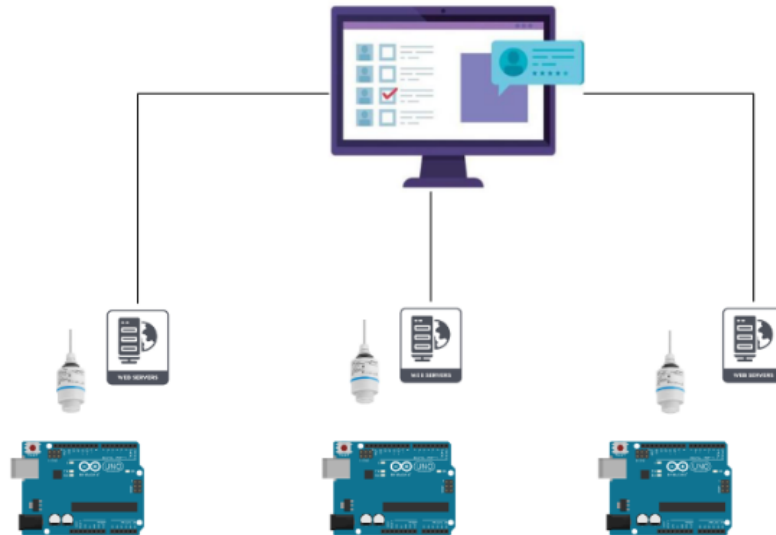


Figura 4.6: Sistema propuesto
Elaboración propia.

4.1.1. Diagrama P&ID

En la figura [4.7](#) se presenta el diseño del P&ID, donde se encuentran dos lazos. En el lazo 100, se realiza la medición del nivel de agua en la bocatoma. Para ello, se emplea un transmisor de nivel que emite una señal de 4-20 mA. Esta señal se convierte a voltaje (0-5V) mediante un convertidor, Finalmente se envía la información a un indicador de nivel permitiendo observar lo datos en una pantalla LCD. Simultáneamente, los datos se transmiten hacia un servidor web para su registro y análisis.

En el lazo 101, se efectúa la medición del nivel del agua en el tanque y al igual que en el lazo 100 la señal recibida del sensor se somete a una conversión y se presenta en una pantalla LCD. Estos datos se envían al mismo servidor web para su posterior procesamiento.

El canal cuenta con dos válvulas manuales las cuales se representan en el P&ID como V1 y V2.

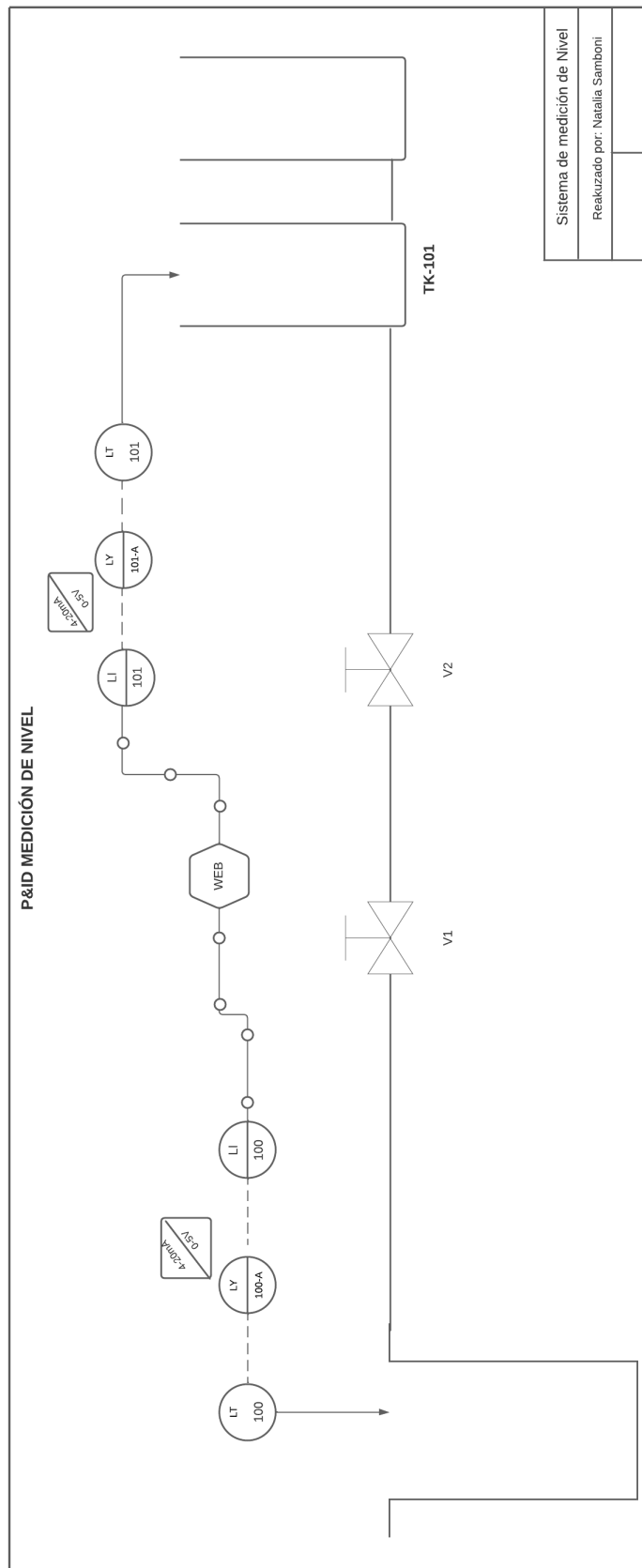


Figura 4.7: Diagrama P&ID.
Elaboración propia.

4.2. Servidor

El servidor cuenta con un sistema de base de datos relacional en este caso se utilizó PostgreSQL quien recibe este nombre desde 1996, está desarrollado en C, es de uso software libre, tiene grandes ventajas frente a otras bases de datos, funciona en diferentes plataformas y es muy estable, cuenta con una gran cantidad de tipo de datos y permite definir una tabla a partir de otra ya definida [60]. El gestor de base de datos en el que se administra es PgAdmin quien otorga una interfaz que permite realizar actividades como crear y modificar las bases de datos, además facilita el uso de Query Tool lo cual permite ejecutar comandos SQL permitiendo analizar la base de datos de una forma visual [61].

Para gestionar el intercambio de información en tiempo real entre la base de datos y el servidor web se hizo uso de una librería denominada “Socket.io” la cual ofrece la función de crear servicios online en la programación, esto permite enviar y recibir información entre procesos del mismo o diferentes sistemas que estén unidos mediante una red, facilitando el acceso a datos centralizados. En TCP/IP los sockets principales son “Stream sockets” donde el protocolo que usan es TCP y por otro lado existe el “Datagram sockets” donde el protocolo utilizado es UDP [62].

El entorno utilizado para gestionar el servidor web, la base de datos junto con el socket es Node.js siendo una de las tecnologías más utilizadas en el momento, el cual está basado en el lenguaje de programación JavaScript y es creado para diseñar aplicaciones escalables, permite múltiples conexiones simultáneamente con un gran rendimiento y también permite enviar archivos pesados [63], esto se resume en la figura 4.8

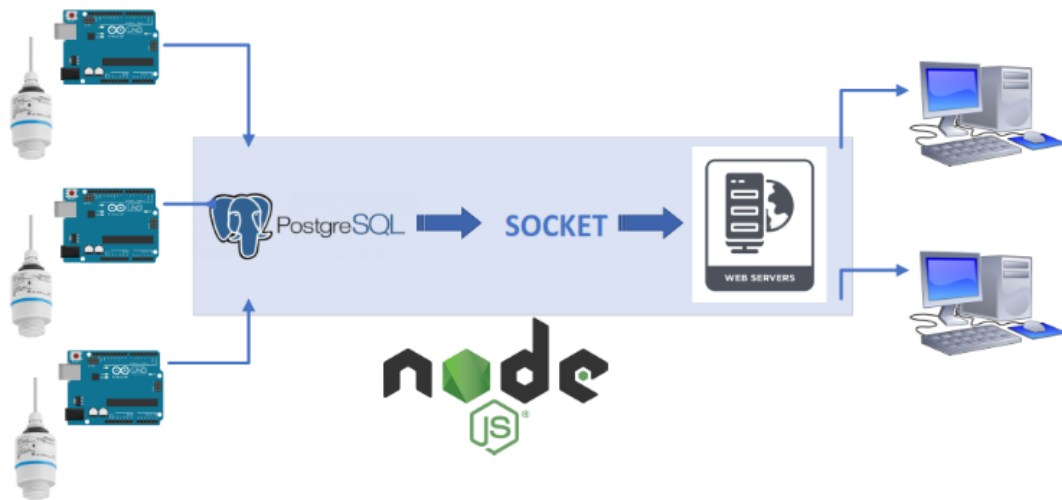


Figura 4.8: Propuesta del intercambio de información
Elaboración propia.

4.3. Circuito

En cada punto de interés donde se requiere la medición de nivel, se lleva a cabo un montaje específico para lograr dicha medición. Para este propósito, se realizó el diseño esquemático del circuito y la PCB utilizando la herramienta Proteus 8 Professional. En la figura [4.9](#) se presentan los diseños realizados para este sistema.

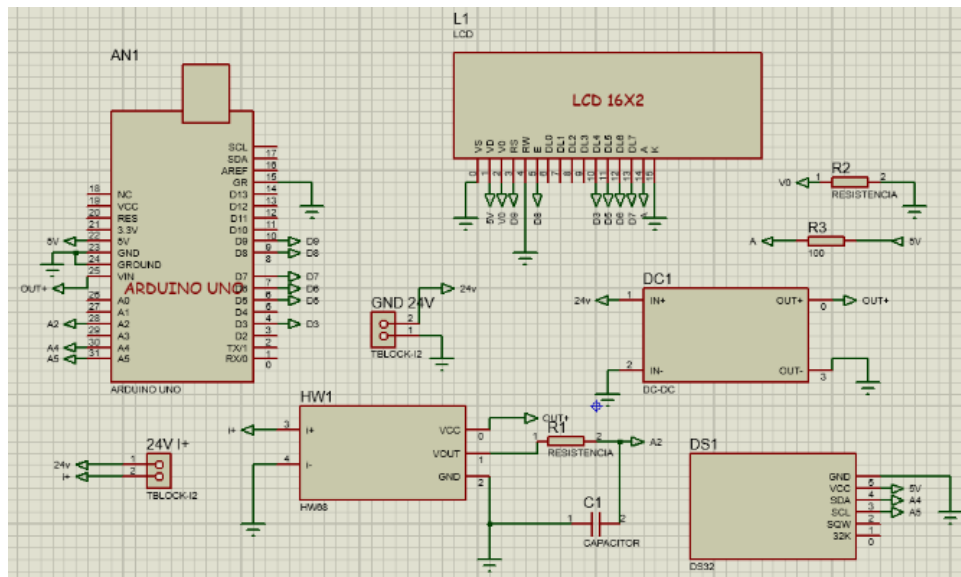


Figura 4.9: Diseño esquemático del circuito
Elaboración propia.

El diseño esquemático del circuito se elaboró teniendo en cuenta los componentes necesarios para la medición de nivel, como el microcontrolador, la LCD, convertidores de señal, el reloj, el capacitor, resistencias y borneras.

Además, se procede a llevar a cabo la elaboración del diseño de la PCB siguiendo las pautas establecidas por las normas IPC. Estas normas representan una herramienta de guía creada por una asociación fundada en 1957, está compuesta por miembros de la industria electrónica a nivel mundial, así como entidades gubernamentales y educativas. La adopción de estas normativas permite alcanzar niveles superiores de calidad en la fabricación de circuitos impresos o PCB [64].

Para llevar a cabo este proceso, se hace uso de la herramienta profesional Proteus 8. Esta herramienta permite el diseño completo de la PCB, donde se incluye la disposición de los componentes en la placa, las pistas de conexión y los puntos de soldadura así como se muestra en la figura 4.10. La PCB permite una implementación compacta y ordenada de los componentes, facilitando su ensamblaje y optimizando el rendimiento del sistema.

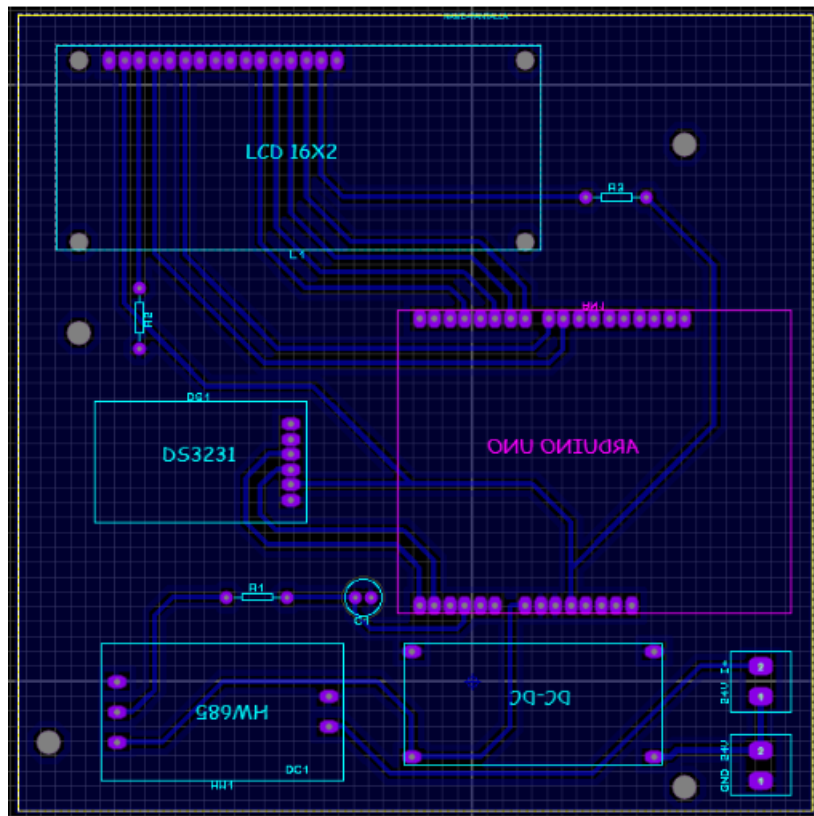


Figura 4.10: Diseño de la PCB
Elaboración propia.

Después de realizar el diseño de la PCB se procede a imprimir la placa en la empresa Colcircuitos S.A.S, donde se dan las especificaciones como el numero de capas, color de antisolder y espesor de la lámina, los resultados finales se indican en la figura [4.11](#).

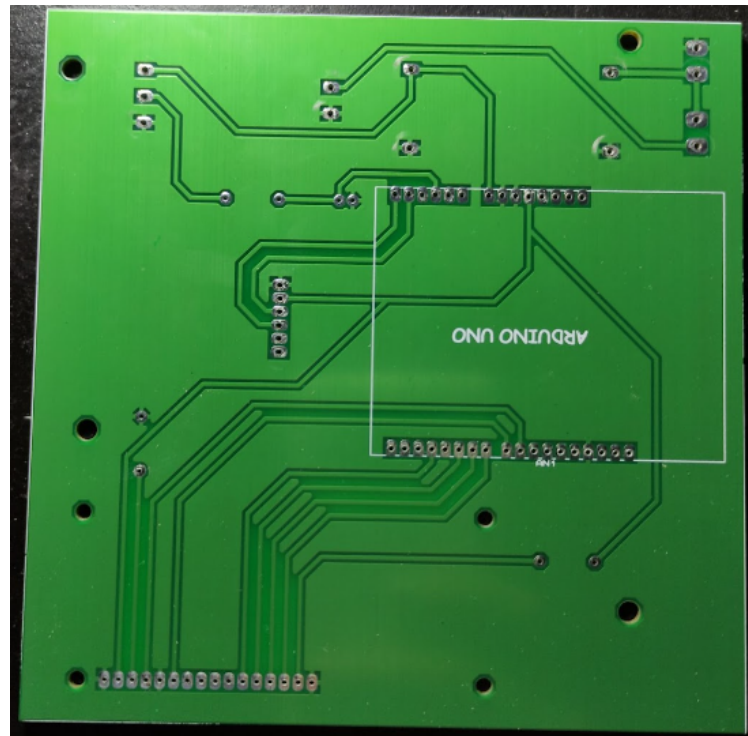


Figura 4.11: PCB impresa
Elaboración propia.

4.4. Conexiones

En la figura [4.12](#) se presentan las conexiones eléctricas del montaje realizado para medir el nivel de agua utilizando un sensor analógico de 4 - 20 mA. Para convertir la señal de corriente a voltaje, se utilizó un convertor de señal llamado HW685. Debido a que el Arduino tiene un rango máximo de voltaje aceptado, fue necesario utilizar un reductor de voltaje para asegurar que la alimentación del Arduino estuviera dentro del rango permitido. Para este propósito, se utilizó un componente llamado LM2596.

Además, se empleó un módulo DS3231 para definir el intervalo de tiempo específico con el cual la información se envía al servidor, dichos valores se pueden visualizar tanto en una aplicación como en una pantalla LCD 16x2.

Para enviar los datos al servidor, se utilizó una tarjeta Ethernet Shield, que se conecta directamente al Arduino y a su vez se conecta a un router por medio de un cable y de esta

manera se tiene acceso a internet, con el fin de enviar los valores obtenidos al servidor.

Todas estas conexiones están alimentadas por una fuente de alimentación con una salida de voltaje de 24V.

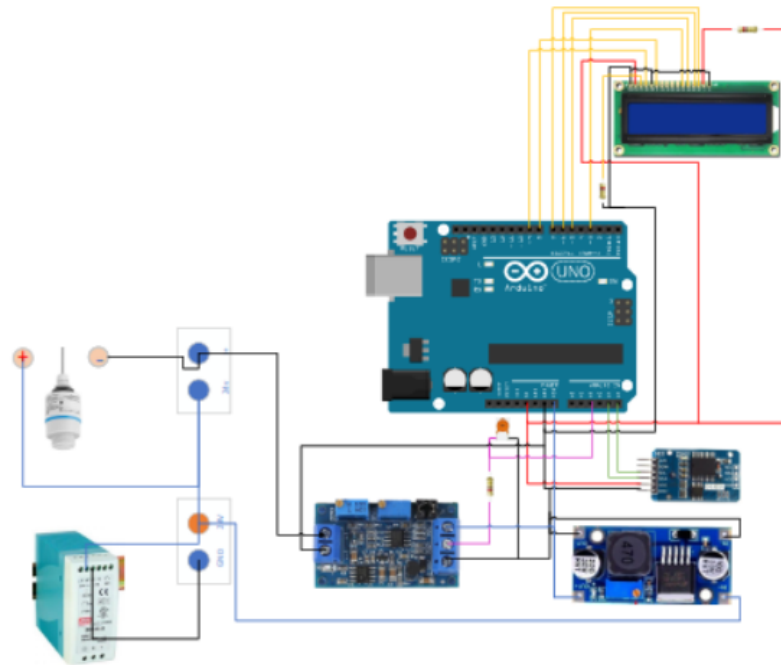


Figura 4.12: Diseño de conexiones
Elaboración propia.

4.5. Implementación

El proyecto se llevó a cabo en las instalaciones de la empresa Municipal de Energía Eléctrica S.A - E.S.P, para el desarrollo de este fue necesario utilizar los mismos elementos de las tablas [3.2](#) y [3.3](#), adicional a estos se requiere los dispositivos que se presentan en la tabla [4.1](#).




Compo.	Descripción	Cant	Imagen
MDR-60-24	Es una fuente de alimentación industrial en riel la cual cuenta con protección para corto- circuito, sobrecarga y sobretensión, esta versión de MDR-60 permite un voltaje de salida de 24V y una corriente de salida de 2.5A, su potencia es de 60W, cuenta con un LED el cual indica si la fuente está encendida [55].	3	 <p>Tomado de: [55].</p>
Tarjeta Ethernet Shield	Esta tarjeta permite que el microcontrolador arduino se conecte a internet por medio de una red cableada ethernet con un cable RJ45, el controlador ethernet es un W5500, cuenta con una ranura para una microSD, es compatible con arduino UNO y Mega, se debe utilizar la librería Ethernet, su velocidad de conexión es 10/100M, funciona con 5V los cuales son suministrados por la placa de arduino [65].	3	 <p>Tomado de: [65].</p>
Wlink 4G/3G	Este dispositivo es un enrutador que funciona con una red 4G/3G con el fin de proporcionar una conectividad móvil con soporte VPN, cuenta con una carcasa fuerte lo cual permite usarla para administración remota, telemetría, monitoreo de condición [66].	3	 <p>Tomado de: [66].</p>

Tabla 4.1: Elementos necesarios para trabajar con Ethernet
Elaboración propia.

4.5.1. Montaje

En primer lugar, se ubican los sensores en los puntos de medición designados, que en este caso son la bocatoma y el tanque de carga. El sensor de nivel en la bocatoma permite monitorear el nivel del agua en el punto de entrada del río Grande, proporcionando información sobre la disponibilidad y flujo de agua, la figura [4.13](#) indica su instalación.



Figura 4.13: Sensor instalado en bocatoma
Elaboración propia.

Por otro lado, el sensor de nivel en el tanque de carga permite conocer el nivel del agua almacenada en dicho tanque, lo que resulta crucial para el control y gestión del suministro de agua en el sistema. Una vez que los sensores están correctamente ubicados en los puntos de medición, se encuentran preparados para captar y transmitir la información del nivel de agua en tiempo real, en la figura [4.14](#) se observa el sensor instalado en el tanque.



Figura 4.14: Sensor instalado en tanque de carga
Elaboración propia.

Posteriormente, se procede a ubicar los componentes necesarios en la PCB, tales como Arduino, Ethernet Shield, LCD 16x2, DS3231, HW685, LM2596, capacitor y resistencias, así como se indica en la figura [4.15](#).

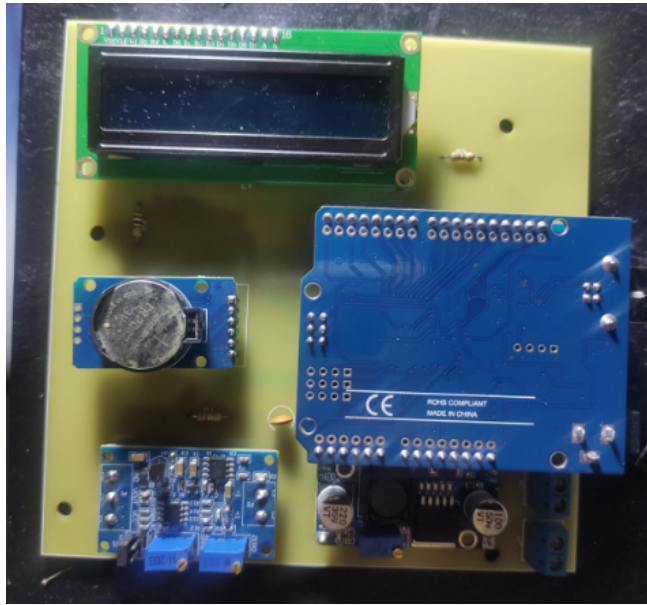


Figura 4.15: Placa con componentes
Elaboración propia.

Una vez que la placa está completamente organizada y soldada, se lleva a cabo el montaje dentro de la caja. En ella se coloca la PCB terminada, la fuente de alimentación y el breaker el cual permite el flujo de energía de manera segura, este montaje se observa en la figura [4.16](#).



Figura 4.16: Montaje completo
Elaboración propia.

Cuando las cajas están completamente terminadas y se han realizado las pruebas correspondientes en un entorno controlado, se procede a instalar los elementos en la planta. Se realizan las conexiones necesarias y se energiza el sistema, como se muestra en la figura

[4.17](#).



Figura 4.17: Sistema completo energizado
Elaboración propia.

Luego, se configura el sensor utilizando la aplicación Endress+Hauser SmartBlue. A través de conexión Bluetooth, se establece comunicación con el sensor. Aunque existen varios parámetros que se pueden configurar, en este caso se establece únicamente el nivel mínimo, nivel máximo y la velocidad de cambio, como se indica en la figura [4.18](#) y [4.19](#)

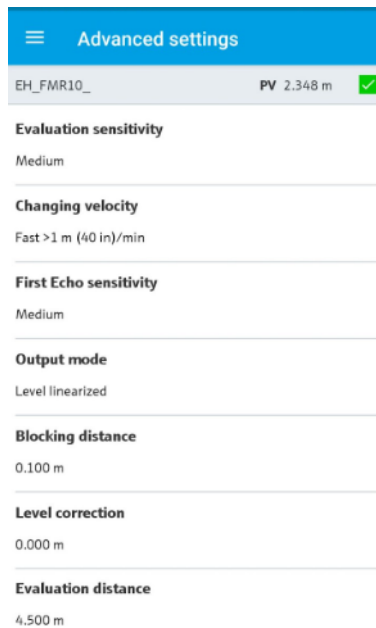


Figura 4.18: Configuración de velocidad de cambio en la aplicación Endress+Hauser
Elaboración propia.



Figura 4.19: Configuración de nivel mínimo y máximo en la aplicación Endress+Hauser
Elaboración propia

Finalmente, una vez configurado el sensor, se realizan pruebas para verificar la correcta medición y asegurar que la información obtenida llegue correctamente al servidor. Se verifica

que el sistema esté funcionando adecuadamente y que los datos del nivel de agua se estén transmitiendo y almacenando correctamente para su posterior análisis y gestión, así como se indica en la figura [4.20](#)



Figura 4.20: Medición e información correcta en el sistema
Elaboración propia.

4.6. Adquisición de datos

Se realiza la programación haciendo uso de Arduino versión 1.8.19. El código es modular y cuenta con 5 Sketch. En el Sketch de la configuración del sensor, el cual se indica a continuación, se define elementos como el pin analógico A2 donde se conecta el sensor, el id del sensor, este permite diferenciar un sensor de otro, los valores mínimos y máximos tanto para configurar la conversión de corriente a voltaje (adc_min, adc_max) como para el rango del nivel, también se declara un valor para un filtro pasa bajo el cual fue necesario usar ya que se tenía mucho ruido al convertir la señal de analógico a digital. Su función principal es seleccionar de la señal de entrada la información que se necesita en la señal

de salida, existen diferentes filtros como lo son pasa baja, pasa alta y pasa banda, depende de las necesidades que se tengan se debe escoger el más conveniente, en este caso fue el 'filtro pasa baja' ya que solo se desea tener la componente de DC para una señal de voltaje [67]. Permite el paso de las frecuencias que están por debajo de un determinado valor [68]. Después de definir las variables necesarias se calcula 'm' y 'b' variables que corresponden a la ecuación de una pendiente en un plano cartesiano, dicha ecuación es la siguiente: $y=mx + b$, donde 'm' es la pendiente de la recta y 'b' es el punto donde la pendiente cruza al eje 'y', los valores 'x' y 'y' son las variables en el plano [69], en este caso vienen siendo Nivel_min, Nivel_max, adc_min y adc_max. El valor recibido del sensor pasa por dos ecuaciones para eliminar el ruido de la señal. Finalmente con este valor, 'm' y 'b' se calcula el valor del nivel del agua en unidades de centímetros.

```
1 #define SenialE A2
2 #define alpha 0.01
3 #define alpha2 0.1
4 #define adc_min 99.84 // Valor min para obtener 4mA en el
   sensor
5 #define adc_max 975.30 //Valor m ximo para obtener 20mA en el
   sensor
6 float NivelMin = 200; // Nivel minimo de agua en cm
7 float NivelMax = 600; // Nivel m ximo de agua en cm
8 float Nivel;
9 float adc_filtro = 100; // Valor filtro para eliminar el
   ruido
10 float adc_filtro2 = 100; // Valorsegundo filtro
11 float m;
12 float b;
13 String id="2";
14 void configurarSensor(){
15     m = 1.0*(NivelMax - NivelMin) / (adc_max - adc_min); // Pendiente
16     b = NivelMin - 1.0*m*adc_min; // punto b de la recta
```

```

17 void MedicionNivel() {
18   int adc_nivel = analogRead(SenialE); // Se almacena el valor del
      sensor
19   adc_filtro = (alpha*adc_nivel)+ ((1-alpha)*adc_filtro); // este
      filtro es utilizado para eliminar el ruido de la se al
20   adc_filtro2 = (alpha2*adc_filtro)+ ((1-alpha2)*adc_filtro2); //
      se hizo uso de dos filtros para poder obtener una se al sin
      ruido
21   Nivel = m*adc_filtro2 + b ; // Se obtiene el Nivel del sensor
      seg n el valor le do por este mismo.}

```

La información obtenida por el sensor debe subirse al servidor web, para lograr esto se tiene el Sketch denominado 'Servidor.h', donde inicialmente se configuran las IPs tanto del servidor, de la tarjeta Ethernet y del modem el cual otorga internet, después de esto se establece conexión ethernet.

```

1 #include <ArduinoHttpClient.h>
2 #include <SPI.h>
3 #include <Ethernet.h>
4 #include "RestClient.h"
5 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // direcci n
      del hardware
6 char server[] = "181.48.232.226"; // IP del servidor (using DNS)
7 IPAddress ip(192, 168, 1, 16); // Ip que se le asigna a la tarjeta
      ethernet shield
8 IPAddress myDns(192, 168, 1, 1); // Dns es la direcci n IP del
      route
9 EthernetClient client;
10 unsigned long beginMicros, endMicros;
11 unsigned long byteCount = 0;
12 bool printWebData = true; // set to false for better speed

```

```
measurement
13 void configurarEthernet() {
14     Serial.println("Initialize Ethernet with DHCP:");
15     if (Ethernet.begin(mac) == 0) { // s Ethernet.begin(mac)
16         Serial.println("Failed to configure Ethernet using DHCP");
17         //Se comprueba si existe la tarjeta ethernet shield
18         if (Ethernet.hardwareStatus() == EthernetNoHardware) {
19             Serial.println("Ethernet shield was not found.
20             while (true) {
21                 delay(1); // se repite el ciclo hasta que se encuentre una
                tarjeta ethernet shield.
22             }
23         }
24         if (Ethernet.linkStatus() == LinkOFF) { //Comprueba cable
                conectado
25             Serial.println("Ethernet cable is not connected.");
26         }
27         Ethernet.begin(mac, ip, myDns); // inicializa la tarjeta
28     } else {
29         Serial.print(" DHCP assigned IP ");
30         Serial.println(Ethernet.localIP());
31     }
```

Ya establecida la conexión Ethernet se procede a hacer envío de los datos, para esto, primero se establece conexión con el servidor, después se crea una cadena 'enviVal' donde se almacena el nivel de agua, el id del sensor, la hora y fecha de cuando se obtuvo el valor de nivel a enviar, finalmente se envían los datos al servidor y sino se logra establecer conexión se muestra un mensaje de error.

```
1     delay(1000);
2     Serial.print("connecting to ");
```



```
3  Serial.print(server);
4  Serial.println(" ...");}
5 void EnvioDato(){
6  if (client.connect(server, 4005)>0) { // si hay conexión (host,
    puerto) si es 1 es exitoso , si es 0 no hay conexión
7      Serial.print("connected to ");
8      Serial.println(client.remoteIP()); // Se escribe la dirección
    IP
9      String enviVal ="{\"valor\":\" + String(Nivel, 2) + ",\"
    id_sensor\":\" + id + \",\" + HoraYFecha()+ "}"; //En enviVal se
    almacena el nivel de sensor, el id y la hora con fecha del dato
    obtenido
10     client.println("POST /prueba HTTP/1.1");
11     client.println("Host: " + String(server));
12     client.println("Content-Type: application/json");
13     client.println("Content-Length: " + String(enviVal.length()));
14     client.println("");
15     client.println(enviVal);
16 } else { // sino tengo conexión con el servidor se envía un
    mensaje que la conexión ha fallado
17     Serial.println("connection failed");
18 }
19 beginMicros = micros();
20 }
```

Este Sketch finaliza con una función 'Respuesta()' donde se lee y procesa la respuesta del servidor. Con 'available()' se verifica la existencia de bytes en el servidor, si se encuentran bytes se actualiza el contador de bytes recibidos. Si el condicional de 'printWebData' es verdadero se imprime los datos en el monitor serie. Después se verifica la conexión con el servidor, si esta se ha desconectado se imprimen ciertos mensajes y se detiene el cliente

con 'client.stop()'.

```
1 void Respuesta() {
2     // Si hay bytes entrantes en el servidor se leen y se imprimen
3     int len = client.available();
4     if (len > 0) {
5         byte buffer[80];
6         if (len > 80) len = 80;
7         client.read(buffer, len);
8         if (printWebData) {
9             Serial.write(buffer, len);
10        }
11        byteCount = byteCount + len;}
12    if (!client.connected()) { // si el servidor no est  conectado
13        se detiene el client
14        endMicros = micros();
15        Serial.println();
16        Serial.println("disconnecting.");
17        client.stop(); // Se detiene el cliente
18        Serial.print("Received ");
19        Serial.print(byteCount);
20        Serial.print(" bytes in ");
21        float seconds = (float)(endMicros - beginMicros) / 1000000.0;
22        Serial.print(seconds, 4);
23        float rate = (float)byteCount / seconds / 1000.0;
24        Serial.print(", rate = ");
25        Serial.print(rate);
26        Serial.print(" kbytes/second");
27        Serial.println();
28    }}
```

En el Sketch reloj.h se configura la comunicación con el reloj, después de establecer la comunicación se configura la hora y fecha, será la misma con la que esté el computador con el cual se programe el arduino.

En la función 'HoraYFecha' se obtiene la hora y fecha actual y se almacena en una variable 'diaHora', se desglosa este dato para así organizar esta información en el formato que acepta la base de datos del servidor, se crea una variable tiempo la cual retorna la fecha y hora organizadas.

```
1 #include <Wire.h> // incluyo la libreria Wire para la comunicaci n
    I2C
2 #include <RTClib.h> // Incluyo la libreria RTClib para manipular el
    modulo DS3231 el cual es para el reloj.
3 RTC_DS3231 reloj;
4 void ConfigurarReloj(void) {
5     if (!reloj.begin()) { //
6         Serial.println("Reloj en falla"); //
7         while (1); // condicional para asegurar que comunicaci n con
            reloj
8     } else { //
9         //Serial.println("Reloj exitoso");
10        //reloj.adjust(DateTime(__DATE__, __TIME__)); //se configura el
            a o , el mes, el dia, la hora, min, seg con la que est
            configurada en el computador.
11    }}
12 String HoraYFecha() {
13    DateTime diaHora = reloj.now(); // se obtiene la hora y fecha
        tipo de dato DateTime
14    String h = String(diaHora.hour()); // Se obtiene la hora
15    String m = String(diaHora.minute()); // Se obtiene los minutos
16    String s = String(diaHora.second()); // Se obtiene los segundos
17    String y = String(diaHora.year()); // // Se obtiene el a o
```

```

18 String mes = String(diaHora.month()); // Se obtiene el mes
19 String d = String(diaHora.day()); // Se obtiene el día
20 String WFecha = d + "/" + mes + "/" + y ; // en la variable
    WFecha se une las variables correspondientes a día, mes y año
21 String WHora = h + ":" + m + ":" + s; // en la variable WHora se
    une las variables correspondientes a hora, minutos y segundos
22 String Tiempo = "\"fecha\":" + WFecha; // creo la variable
    tiempo donde almaceno la variable WFecha
23 Tiempo += "\",\"hora\":" + WHora + "\""; // En la variable
    tiempo donde almaceno WFecha también almaceno WHora.
24 return Tiempo; //retorna tiempo donde esta la fecha y hora}

```

En el Sketch principal se llaman las funciones de configurar la tarjeta Ethernet, el sensor y el reloj, se hace el envío de datos al servidor cada minuto y se muestra en la LCD el valor del nivel del agua, el código completo se visualiza a continuación.

```

1 #include "reloj.h"
2 #include "Sensor.h"
3 #include "Servidor.h"
4 #include "lcd.h"
5 bool    ishora = false;
6 void setup() {
7     Serial.begin(9600); //inicializamos la comunicacion serial
8     lcd.begin(COLS, ROWS);
9     mostrarLCD();
10    delay(2000);
11    ConfigurarReloj();
12    configurarSensor();
13    configurarEthernet();
14    Serial.println("config ok");
15 }

```

```
16 void loop() {
17   MedicionNivel();
18   mostrarLCDval();
19   DateTime diaHora = reloj.now();
20   if (!ishora && diaHora.second() == 0) {
21     ishora = true;
22     Serial.println(Nivel);
23     String enviVal = "{\"valor\":" + String(Nivel, 2) + ",\"
id_sensor\":" + id + ",\"fecha\":" + HoraYFecha() + "}";
24     EnvioDato();
25     Respuesta();
26   }
27
28   if (ishora && diaHora.second() >= 2 ) {
29     ishora = false;
30   }
31 }
```

4.7. Servidor Implementado

Una vez realizadas las conexiones necesarias, la información del nivel de agua medido está disponible en el servidor, el cual cuenta con una interfaz con una pantalla inicial donde se puede visualizar los últimos valores registrados. La información se actualiza automáticamente en tiempo real cada vez que se recibe un nuevo dato del nivel de agua, así como se muestra en la figura [4.21](#).



Figura 4.21: Pantalla principal del servidor
Elaboración propia.

Desde la anterior ventana, los usuarios pueden acceder a dos pantallas adicionales las cuales se ilustran en la figura 4.22 y 4.23, al hacer clic en los botones ‘Sensor 1’ y ‘Sensor 2’. Estas proporcionan un histórico de los datos recibidos de cada sensor, junto con su ubicación correspondiente.



Figura 4.22: Pantalla Sensor 1
Elaboración propia.



Figura 4.23: Pantalla Sensor 2
Elaboración propia.

Capítulo 5

CONCLUSIONES

- En la actualidad se cuenta con diversas alternativas para llevar a cabo la comunicación a distancia. La elección de la opción más óptima depende en gran medida de las necesidades específicas del proyecto, ya que el protocolo de comunicación adecuado debe seleccionarse en función de dichas necesidades.
- Debido a las considerables distancias y obstáculos que existen entre los puntos de medición y la planta, el módulo LoRa E32-433T30D no es efectivo para la elaboración de este proyecto.
- La supervisión a distancia del nivel del agua en el canal reduce de manera significativa el traslado de los operarios, esto también agiliza la toma de decisiones de manera eficiente.
- El uso de un servidor de web permite que tanto el gerente como el operario sepan el estado del nivel del agua en tiempo real y con datos precisos ya que el sensor está midiendo en todo momento, contar con la aplicación brinda un conocimiento a las personas que están lejos de la planta.
- Contar con un registro histórico del nivel del agua en el canal, acompañado de los datos de la energía creada, brinda la oportunidad de llevar a cabo estudios sobre el comportamiento de la generación de energía y evaluar posibles mejoras en el sistema.

5.1. Recomendaciones

- Instalar el sensor número tres el cual se entregó completamente calibrado y organizado ya que el canal es bastante largo y esta información permitiría saber con antelación cuanta agua podría llegar al tanque de carga.
- La empresa tiene como proyecto instalar fibra óptica en la bocatoma y en el tanque de carga por lo tanto sería mejor conectar este punto al arduino y desconectar el modem que se encuentra en el armario de esta manera se tiene una conexión mas estable y no se debe pagar adicional un plan de datos mensual.

Bibliografía

- [1] E. E.S.P, “Área de generación.”
- [2] C. de regulación de Energía y Gas, “Creg-025-1995,” 1995.
- [3] E. E.S.P, “Área de comercialización.”
- [4] A. de Puracé, “Mi municipio,” 2022.
- [5] E. E.S.P, “Reseña histórica.”
- [6] Celsia, “Sistema interconectado nacional, sin,” 2021.
- [7] A. municipal, “Plan de desarrollo municipal 2020 - 2023.”
- [8] “Plan de desarrollo municipal de puracé 2004 - 2007.” [Online]. Available: <https://repositoriocdim.esap.edu.co/bitstream/handle/123456789/10068/3590-1.pdf?sequence=1&isAllowed=y>
- [9] K. Rose, S. Eldridge, and L. Chapin, “La internet de las cosas - pequeña reseña,” Octubre 2015.
- [10] M. F. Barcell, “Normas y estándares.” [Online]. Available: https://rodin.uca.es/bitstream/handle/10498/16832/temall_normas.pdf
- [11] “El modelo OSI y los protocolos de red.” [Online]. Available: https://blyx.com/public/docs/pila_OSI.pdf
- [12] J. A. P. Martínez, “La capa de red. direccionamiento, máscaras de longitud, variable, mecanismos de transición de ipv4 a ipv6, icmp e igmp.” 2020.
- [13] J. T. Guerra, “Servicios de nivel de transporte en la arquitectura tcp/ip,” 2003.
- [14] CLOUDFLARE, “¿qué es el modelo OSI?” [Online]. Available: <https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>

- [15] E. H. Rabanal, P. C. M. Campos, Neicer and, R. K. Manturano, and M. A. Díaz, "Internet de las cosas (iot) – Ámbito de aplicación y modelos de comunicación," 2022. [Online]. Available: https://laccei.org/LACCEI2022-BocaRaton/full_papers/FP652.pdf
- [16] R. M. I. Urgilés, "Monitorización en la nube de 'livestock' para aplicaciones iot en el ámbito de 'smart agriculture'," 2021.
- [17] H. Tschofenig, J. Arkko, D. Thaler, and D. McPherson, "Consideraciones arquitectónicas en redes de objetos inteligentes," 2015.
- [18] M. MORA and K. URREGO, "Internet de las cosas: Modelos de comunicación, desafíos y aplicaciones," 2018.
- [19] TIBCO, "¿qué es un silo de datos?"
- [20] R. Mclean, A. Sabella, and M. Yannuzzi, "Estándares y mejores prácticas de seguridad e iot," 2019.
- [21] J. M. Moreno and D. Ruiz, "Informe técnico: Protocolo zigbee (ieee 802.15.4)," 2007.
- [22] L. SEMTECH, "What are lora and lorawan?"
- [23] T. things network, "What are lora and lorawan?"
- [24] G. L. Sparacino, "Tecnología inalámbrica bluetooth sobre los servicios de comunicaciones en los ámbitos social y empresarial." 2003.
- [25] G. C. Ayala, J. Sallas Armenta, H. Castillo, A. Rodríguez, S. Lugo Zavala, and M. Ramírez, "Sistema de control inalámbrico: Diseño, construcción y aplicación en caldera industrial," 2014. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=7894051>
- [26] J. A. Carballar, "Wi-fi. lo que se necesita saber," 2010.
- [27] J. Gámiz Caro and J. Gámiz Caro, "Ethernet como soporte de sistemas de control en red," 2011.
- [28] J. Márquez Diaz, K. Pardo Sánchez, and S. Pizarro Valencia, "Ethernet: Su origen, funcionamiento y rendimiento," 2001.

- [29] A. F. Ruiz Olaya, A. Barandica López, and F. G. Guerrero Moreno, "Implementación de una red modbus/tcp," 2004. [Online]. Available: https://revistaingenieria.univalle.edu.co/index.php/ingenieria_y_competitividad/article/view/2277/3027
- [30] J. J. Sotomayor, "Diseño e implementación de una red inalámbrica para un proceso industrial utilizando protocolo modbus," 2014.
- [31] A. R. OYARCE, "Implementación del protocolo modbus sobre una tarjeta de desarrollo para su uso sobre una red gsm con enfoque en telemetría," 2009.
- [32] J. Vesga, "Microcontroladores motorola-freescale: Programación, familias y sus distintas aplicaciones en la industria." 2007.
- [33] Arduino, "Qué es arduino."
- [34] M. S. Massimo Banzi, "Introducción a arduino," 2022.
- [35] J. Novillo Vicuña, D. Hernández, B. Mazón Olivo, J. Molina Ríos, and O. Cárdenas Villavicencio, "Arduino y el internet de las cosas," 2018. [Online]. Available: <https://www.3ciencias.com/wp-content/uploads/2018/10/ARDUINO-Y-EL-INTERNET-DE-LAS-COSAS.pdf>
- [36] C. G. Domínguez, "Aplicaciones orientadas a la domótica con raspberry pi," 2015.
- [37] P. Bertoleti, "Proyectos con esp32 y lora," 2019.
- [38] C. V. Porras, "Diseño y desarrollo de un sistema de precintado remoto sobre freertos para sistemas iot." 2021.
- [39] O. E. Barra Zapata and F. Barra Zapata, "Microcontroladores pic con programación pbp."
- [40] A. Serna Ruiz, F. A. Ros García, and J. C. Rico Noguera, "Guía práctica de sensores."
- [41] L. G. Corona, G. Abarca, and J. Mares, "Sensores y actuadores aplicaciones con arduino," 2014.
- [42] Wenglor, "Funcionamiento y tecnología de un sensor de ultrasonidos."
- [43] microsonic, "Sensores ultrasónicos."

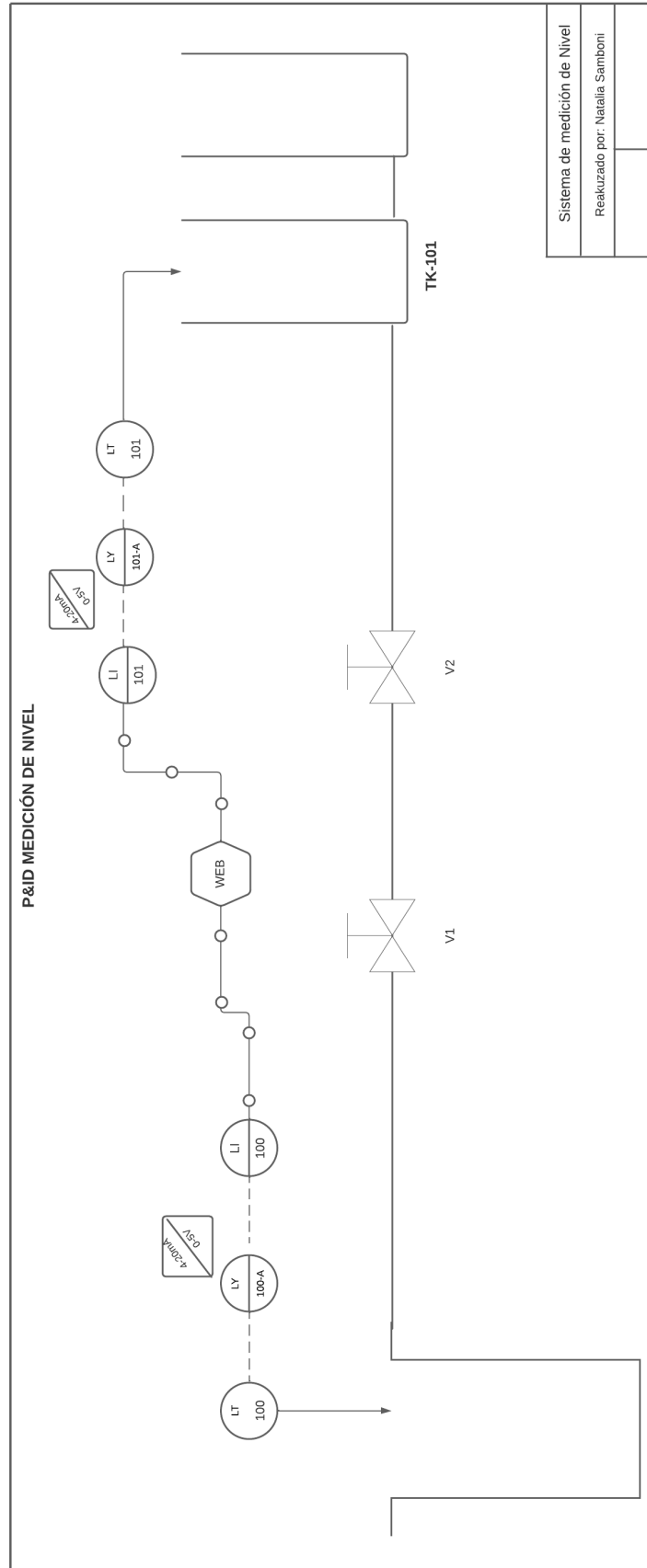
- [44] R. L. Auqui Sayago, "Automatización de los sistemas auxiliares para la planta de tinturado de la empresa lumontex s.a," 2015.
- [45] A. Creus, "Instrumentación industrial," 2011.
- [46] A. S. V. Víctor Brayan Llontop Túllume, "Prototipo de sistema de monitoreo remoto de nivel de agua para un reservorio elevado de la universidad nacional pedro ruiz," 2023.
- [47] genuine Bindicator, "Sensor de radar de onda guiada."
- [48] Arduino, "Arduino uno rev 3."
- [49] A. Mohammed and R. M. Hagem, "Low cost portable system for converting mosul electrical substations to smart one's," 2021.
- [50] Arduino, "Liquid crystal displays (lcd) with arduino."
- [51] W. EWALD, "Ds3231 – real-time clock," 2021.
- [52] Arduino, "Arduino store dc - dc."
- [53] Endress+Hauser, "Radar measurement time-of-flight micropilot fmr10."
- [54] M. Well, "Mdr-60."
- [55] Ebyte, "432-433t30d user manual."
- [56] C. Rodríguez Boada and I. Lavandera Rodríguez, "Antena monopolo parabanda ancha en las frecuencias de hf y vhf." 2017.
- [57] "Lora." [Online]. Available: <https://lora.readthedocs.io/en/latest/#lora>
- [58] "Cisco ccna 1 exploration," 2012.
- [59] E. T. Distribution], "Turbinas hidráulicas, lo que debes saber," 2022. [Online]. Available: <https://epidor.com/blog/turbinas-hidraulicas/#:~:text=La%20funci%C3%B3n%20de%20una%20turbina,generador%20que%20transforma%20esta%20energ%C3%ADa.>
- [60] M. Gibert Ginesta and O. Pérez Mora, "Bases de datos en postgresql."
- [61] C. A. Dorantes, "Descubre postgresql: qué es, cómo funciona y ventajas."
- [62] K. C. Michael Donahoo, "Tcp/ip sockets in c," 2009.

- [63] OpenWebinars, “Qué es nodejs y para qué sirve,” 2019.
- [64] A. delta, “Normas internacionales básicas para diseño de circuitos impresos pcb y productos electrónicos.”
- [65] Arduino, “Arduino ethernet shield.”
- [66] WLINK, “R200 4g/3g router 2xlan.” [Online]. Available: https://www.wlink-tech.com/wlink-r200-4g-3g-router_p0024.html
- [67] A. S. Chipule and R. P. guila, Gerardo, “Diseño de un filtro digital pasa bajas de primer y segundo orden a partir de circuito rc,” 2016.
- [68] E. G. Gutiérrez, “Introducción al filtrado digital,” 2009.
- [69] O. A. M. Herrerías, “Ecuación de la recta conociendo un punto y su pendiente,” 2020.

Capítulo 6

ANEXOS

- P&ID MEDICIÓN DE NIVEL.
- MANUAL DE USUARIO.
- PRESUPUESTO DE EQUIPOS.





EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P

-EMEESA E.S.P-

NIT. 891.500.061-8

VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS

N.U.I.R 219001000-2

MANUAL DE OPERARIO:

La funcionalidad principal de esta aplicación consiste en proporcionar a los operarios información actualizada sobre los niveles de agua en la bocatoma y el tanque de carga. El objetivo es minimizar los desplazamientos innecesarios hasta estos lugares y, al mismo tiempo, agilizar el proceso de toma de decisiones en caso de ser necesario.

Acceso al servidor web:

Los usuarios tienen la capacidad de acceder al servidor web desde cualquier dispositivo, ya sea un teléfono celular o una computadora, en cualquier momento y lugar. Solo necesitan ingresar la dirección web correspondiente, que en este caso es: 181.48.232.226:4005, así como se indica en la figura 1.

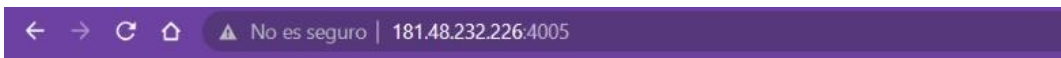


Figura 1. Ingreso dirección web.

Información presentada en el servidor web:

La pantalla principal se muestra en la figura 2, en la cual se representan gráficamente los últimos 15 datos recibidos provenientes de los sensores 1 y 2. La información se proporciona en tiempo real, y además, se incluyen cuadros en los que se puede visualizar la fecha, la hora, el valor y el estado del dato más reciente recibido.

Los sensores tienen dos estados los cuales son: OK o ALERTA esto depende del valor del nivel. Para el sensor 1 el nivel mínimo es 40 centímetros y el nivel máximo es 250 centímetros y para el sensor 2, su valor mínimo es 200 centímetros y su valor máximo es 600 centímetros, si la información recibida está por debajo del mínimo o por encima del máximo se determina su estado como ALERTA.



EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P

-EMEESA E.S.P-

NIT. 891.500.061-8

VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS

N.U.I.R 219001000-2



Figura 2: Pantalla principal.

En esta pantalla se puede ocultar la información del sensor 1 o 2 solo dando clic en el que se quiera esconder, así como se muestra en la figura 3.



Figura 3. Ocultar información de un sensor.

Si se desea ver solo la información del sensor 2 la gráfica se mostrará como se indica en la figura 4.

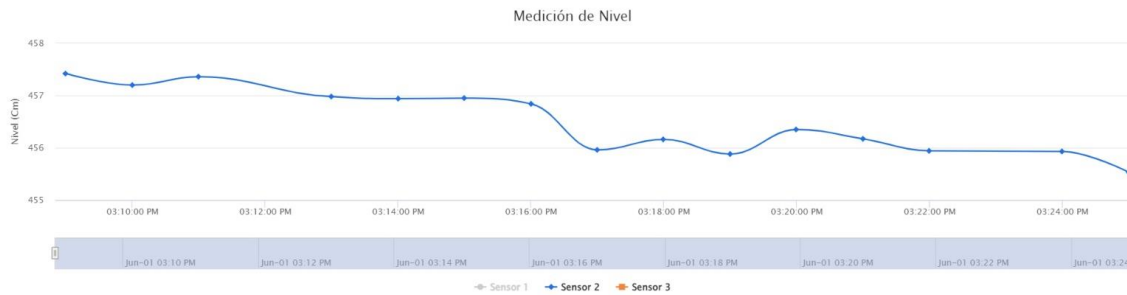


Figura 4. Gráfica del sensor 2 en la pantalla principal.



EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P

-EMEESA E.S.P-

NIT. 891.500.061-8

VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS

N.U.I.R 219001000-2

Y si por el contrario solo se quiere ver la información del sensor 1 la gráfica se verá como se indica en la figura 5.

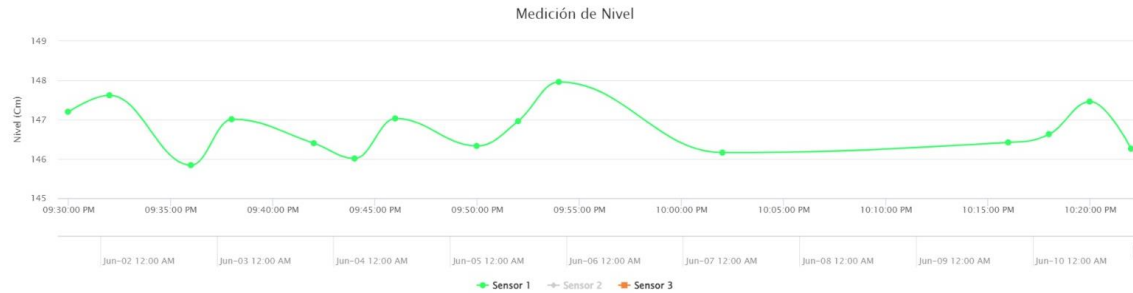


Figura 5. Gráfica del sensor 2 en la pantalla principal.

Además, en esta pantalla, existe la opción de hacer clic en los cuadros correspondientes a cada sensor, así como se indica en la figura 6, esta función permite acceder a una nueva pantalla.



Figura 6. Abrir ventana secundaria.

Si se decide abrir la ventana del sensor 1 la información correspondiente se presenta tal como se ilustra en la figura 7.



Figura 7: Pantalla secundaria sensor 1.



EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P

-EMEESA E.S.P-

NIT. 891.500.061-8

VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS

N.U.I.R 219001000-2

En esta ventana, se presenta un registro histórico en forma de gráfica que refleja los datos recibidos a lo largo del tiempo. Adicionalmente, se incorporan dos recuadros independientes que suministran información adicional. En uno de ellos, se expone la ubicación correspondiente, que en esta instancia es la bocatoma, y en el otro se exhibe el valor más reciente recibido.

La ventana secundaria del sensor 2 se indica en la figura 8 la cual tiene las mismas funciones de la pantalla del sensor 1 pero su diferencia es la información brindada en la ubicación ya que este sensor se encuentra en el tanque de carga.

Visualización Sensor 2



Figura 8: Pantalla secundaria sensor 2.



INFORMACIÓN SOLO PARA LA PERSONA A CARGO.

Si los rangos máximos y mínimos del sensor cambian se debe modificar en el código del servidor web en las variables minX y maxX según corresponda, donde se muestra en la figura 9.

```
angular.module('FlFeederApp', []).
controller('driversController', function($scope,$http) {
  $scope.datos_usuario = "";
  $scope.num="";
  var socket=io.connect('http://181.48.232.226:4005',{ 'forceNew':true});

  socket.on('Nivel', function(data){
    //console.log(data);
    render(data);
  })

  var min1=40;
  var max1=250;

  var min2=200;
  var max2=600;

  var min3=100;
  var max3=150;
```

Figura 9. Código del servidor para cambiar los rangos del nivel.

Y en el código del Arduino se debe cambiar en el Sketch del sensor en la variable NivelMin y NivelMax, así como se muestra en la figura 10.

NivelAgua	Sensor.h	Servidor.h	lcd.h	reloj.h
#define	Seniale	A2		
#define	alpha	0.01		
#define	alpha2	0.1		
#define	adc_min	99.84 // Es el valor minimo para obtener el valor de 4mA en el sensor		
#define	adc_max	975.30 // Es el valor máximo para obtener el valor de 20mA en el sensor		
float	NivelMin	= 200; // Nivel minimo de agua en cm		
float	NivelMax	= 600; // Nivel máximo de agua en cm		
float	Nivel			
float	adc_filtro	= 100; // valor de 100 al filtro para eliminar el ruido		
float	adc_filtro2	= 100; // valor de 100 para el segundo filtro para eliminar el ruido		
float	m;			
float	b;			
String	id	"2";		

Figura 10. Código del Arduino para cambiar los rangos del nivel.

Si se desea cambiar la frecuencia con la que se envía los datos al servidor se debe modificar el código del Arduino en el Sketch principal, así como se indica en la figura 11.



EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P
-EMEESA E.S.P-

NIT. 891.500.061-8
VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS
N.U.I.R 219001000-2

```
void loop() {
  MedicionNivel();

  mostrarLCDval();
  DateTime diaHora = reloj.now();

  if (!ishora && diaHora.second() == 0) {
    ishora = true;
    Serial.println(Nivel);
    String enviVal = {"valor\":" + String(Nivel, 2) + ",\nid_sensor\":" + id + ",\nfecha\":" + HoraYFecha() + "};
    EnvioDato();
    Respuesta();
  }

  if (ishora && diaHora.second() >= 2 ) {
    ishora = false;
  }
}
```

Figura 11. Código para frecuencia de envío de datos al servidor.



EMPRESA MUNICIPAL DE ENERGÍA ELÉCTRICA S.A. E.S.P

-EMEESA E.S.P-

NIT. 891.500.061-8

VIGILADA SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICILIARIOS

N.U.I.R 219001000-2

Presupuesto sistema de medición de nivel del agua.

El presupuesto presentado a continuación es válido solo hasta del 1 de febrero de 2023.

Dispositivo	Proveedor	Cant.	Precio unitario COP	Iva	Precio total COP
FMR10	Endress+Hauser	3	3.084.524	19%	11.011.752
Arduino UNO	MACTRONICA	3	55.882	19%	199.499
Ethernet W5100	MACTRONICA	3	89.916	19%	321.000
HW-685	MACTRONICA	3	10.924	19%	38.999
DS3231	MACTRONICA	3	13.025	19%	46.499
LM2596	MACTRONICA	3	5.882	19%	20.999
LCD 16X2	MACTRONICA	3	10.084	19%	36.000
MDR-60-24	SIGMA ELECTRONICA	3	152.465	19%	544.300
PCB	COLCIRCUITOS	3	94.598	19%	337.715
ROUTER	AMAZON	3	600.000	0%	1.800.000
CAJA ELECTRONICA	TODO ELÉCTRICO	3	150.000	0%	450.000
TOTAL			4.267.300		14.806.763