



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE TELECOMUNICACIONES

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

“DISEÑO DE UN SISTEMA INALÁMBRICO DE MONITOREO Y NOTIFICACIÓN DEL NIVEL DE AGUA POTABLE EN LA CAPTACIÓN DE ABASTECIMIENTO DE LAS REDES DE DISTRIBUCIÓN MEDIA Y BAJA DE SAN GABRIEL PROVINCIA DEL CARCHI”

Trabajo de titulación previo a la obtención del título de Ingeniera en Telecomunicaciones.

Línea de investigación: Innovación Tecnológica y de productos

AUTOR:

Lizeth Magali Cárdenas Mallamas

DIRECTOR:

Msc. Jaime Roberto Michilena Calderón

Ibarra, 2025

**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	0402010904		
APELLIDOS Y NOMBRES:	Cárdenas Mallamas Lizeth Magali		
DIRECCIÓN:	San Gabriel - Carchi		
EMAIL:	lmcardenasm@utn.edu.ec		
TELÉFONO FIJO:	XXXXXXXX	TELF. MOVIL	0994809492

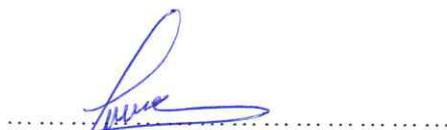
DATOS DE LA OBRA	
TÍTULO:	Diseño de un Sistema Inalámbrico de Monitoreo y Notificación del Nivel de Agua Potable en la captación de abastecimiento de las redes de distribución media y baja de San Gabriel provincia del Carchi
AUTOR (ES):	Cárdenas Mallamas Lizeth Magali
FECHA: AAAAMMDD	10/02/2025
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniera en Telecomunicaciones
DIRECTOR:	Msc. Jaime Roberto Michilena Calderón

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Lizeth Magali Cárdenas Mallamas, con cédula de identidad Nro. 0402010904, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 10 días del mes de febrero de 2025

EL AUTOR:



Cárdenas Mallamas Lizeth Magali

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 10 días, del mes de febrero de 2025

EL AUTOR:

A handwritten signature in blue ink, appearing to read 'Lizeth Magali', is written over a horizontal dotted line.

Cárdenas Mallamas Lizeth Magali

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 10 de febrero de 2025

Msc. Jaime Roberto Michilena Calderón
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



.....
Msc. Jaime Roberto Michilena Calderón
C.C.: 1002198438

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificador del trabajo de Integración Curricular “Diseño de un Sistema Inalámbrico de Monitoreo y Notificación del Nivel de Agua Potable en la captación de abastecimiento de las redes de distribución media y baja de San Gabriel provincia del Carchi” elaborado por Cárdenas Mallamas Lizeth Magali previo a la obtención del título de Ingeniera en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



.....
Msc. Jaime Roberto Michilena Calderón
C.C.: 1002198438



.....
Msc. Luis Edilberto Suárez Zambrano
C.C.: 1002304291

DEDICATORIA

La realización de este trabajo, con todo el proceso que ha implicado y los años de esfuerzo y dedicación que finalmente han dado su fruto, es el resultado de un esfuerzo colectivo y el apoyo invaluable de diversas personas. A ellas debo este logro. Quiero agradecer especialmente a mi familia, mi principal fuente de fortaleza. En particular, a mi madre, Gladys, quien con su amor incondicional ha sido mi mayor inspiración, especialmente en los momentos más adversos de mi vida, cuando las dificultades parecían insuperables. Su confianza inquebrantable, su apoyo constante, y su capacidad de ser no solo mi madre, sino también mi amiga más fiel, han sido cruciales para que hoy pueda estar aquí, celebrando este logro. A ella, que con su lucha diaria ha trabajado incansablemente para brindarnos un futuro mejor, le dedico este esfuerzo. Asimismo, a mi tío, mis tías y mis abuelitos, quienes también han sido un pilar fundamental en este camino, ofreciéndome su apoyo y cariño constante.

De igual forma quiero agradecer a mis amigos, quienes, a lo largo de estos años universitarios, me han demostrado una amistad sólida e inquebrantable. Ellos han sido mi refugio en los momentos de estrés y dificultades, siempre brindándome apoyo en medio de los arduos momentos de trabajo.

Este logro es el reflejo del amor, la confianza y el apoyo de todos los que han estado a mi lado.

AGRADECIMIENTO

Quiero dedicar este agradecimiento a las personas que han sido fundamentales en mi vida durante este proceso. En primer lugar, a mi madre, quien con su amor incondicional ha sido mi pilar más grande. A ella le debo no solo este logro, sino también la fuerza para seguir adelante en los momentos de incertidumbre y dificultad. Su dedicación, paciencia y sacrificio diario me han enseñado el verdadero significado de la perseverancia. A mi familia, por ser siempre mi apoyo inquebrantable, por su aliento constante y por estar a mi lado en cada paso de este camino.

A los ingenieros que me han guiado en este proceso de titulación, el MSc. Jaime Michilena y el MSc. Luis Suárez, por compartir generosamente su vasto conocimiento, por sus valiosos consejos y enseñanzas. Su apoyo y compromiso conmigo han sido fundamentales para que este trabajo vea la luz. Gracias por brindarme las herramientas necesarias para desarrollar y fortalecer mis habilidades, por ser excelentes mentores y por creer en mí incluso cuando las dudas parecían nublar el horizonte.

A la Casona Universitaria de la Universidad Técnica del Norte, por otorgarme la oportunidad de crecer académicamente y por ser el espacio donde he podido desarrollar mis conocimientos y capacidades. Este logro no solo es el resultado de mi esfuerzo individual, sino también de todo el respaldo que he recibido durante mi formación. Mi más sincero agradecimiento a todos los docentes que, con su dedicación, han sido no solo maestros, sino también consejeros, siempre dispuestos a guiarme y a brindarme las bases para llegar hasta aquí.

A mis amigos, quienes, desde el primer día de nivelación hasta este momento de culminación, se han mantenido a mi lado demostrando una amistad sólida e

inquebrantable. Ellos han sido un apoyo constante y, con sus ocurrencias, han logrado hacer mucho más llevadero este camino. Gracias por su compañía, por hacernos reír en los momentos de estrés, y por recordarme que la vida universitaria también se trata de disfrutar y aprender juntos. A (LS), por su generosidad al compartir su tiempo y conocimientos, y por ser un apoyo crucial durante este trabajo de titulación.

Este agradecimiento va también para todos aquellos que, de alguna manera, directa o indirecta, han aportado a mi crecimiento personal y profesional. Gracias a todos por ser parte de esta etapa tan importante de mi vida.

RESUMEN EJECUTIVO

El presente proyecto aborda la problemática del monitoreo manual en la captación de agua potable de las redes de distribución media y baja de San Gabriel, provincia del Carchi, donde deficiencias en el sistema han provocado interrupciones significativas en el suministro, afectando hasta al 70% de los usuarios en eventos críticos como sequías o taponamientos derivados de lluvias intensas; esta situación evidencia la necesidad de modernizar el proceso para garantizar un suministro continuo y confiable. Con este propósito, se planteó como objetivo principal diseñar e implementar un sistema inalámbrico basado en tecnologías IoT, capaz de realizar mediciones del nivel de agua en tiempo real, notificar automáticamente a los operadores en caso de anomalías y proporcionar una plataforma de visualización accesible para optimizar la gestión del sistema. La metodología incluyó un análisis exhaustivo del funcionamiento actual de las redes de distribución, la selección de sensores ultrasónicos y nodos de comunicación LoRa considerando su compatibilidad con las condiciones geográficas y técnicas del entorno, la configuración de una plataforma remota para la recolección y visualización de datos, y la realización de pruebas de campo para verificar el desempeño del sistema en escenarios reales; este enfoque permitió ajustar cada componente a las necesidades del proyecto. Entre los resultados más relevantes se destacan la eliminación de inspecciones manuales frecuentes, el envío inmediato de alertas ante niveles fuera del umbral, la optimización del tiempo del personal encargado y el fortalecimiento del control sobre el recurso hídrico mediante el acceso remoto a la información en tiempo real. El sistema propuesto mejora significativamente la gestión del agua potable en el área de captación, ofreciendo una solución eficiente y sostenible para la región. Asegura una respuesta oportuna ante situaciones críticas y optimiza el tiempo de trabajo del operador. Además,

beneficia a la población de las redes de distribución media y baja, garantizando un suministro confiable.

Palabras clave: Monitoreo Inalámbrico, Nivel de Agua, IoT, LoRa, Captación, Automatización.

ABSTRACT

The present project addresses the problem of manual monitoring of drinking water collection in the medium and low distribution networks of San Gabriel, province of Carchi, where deficiencies in the system have caused significant interruptions in the supply, affecting up to 70% of users in critical events such as droughts or clogging caused by heavy rains; this situation evidences the need to modernize the process to ensure a continuous and reliable supply. With this purpose, the main objective was to design and implement a wireless system based on IoT technologies, capable of measuring the water level in real time, automatically notifying operators in case of anomalies and providing an accessible visualization platform to optimize system management. The methodology included a thorough analysis of the current operation of the distribution networks, the selection of ultrasonic sensors and LoRa communication nodes considering their compatibility with the geographical and technical conditions of the environment, the configuration of a remote platform for data collection and visualization, and the performance of field tests to verify system performance in real scenarios; this approach allowed adjusting each component to the needs of the project. Among the most significant results are the elimination of frequent manual inspections, the immediate sending of alerts when levels exceed thresholds, the optimisation of staff time and the strengthening of control over the water resource through remote access to real-time information. The proposed system significantly improves the management of drinking water in the catchment area, providing an efficient and sustainable solution for the region. It ensures a timely response to critical situations and optimises the operator's working time. It also benefits the population of the medium and low distribution networks, ensuring a reliable supply.

Keywords: Wireless Monitoring, Water Level, IoT, LoRa, Intake, Automation.

LISTA DE SIGLAS

IoT. Internet of Things (Internet de las Cosas)

LoRa. Long Range (Larga Distancia)

LoRaWAN. Long Range Wide Area Network (Red de Área Amplia de Larga Distancia)

GAD. Gobierno Autónomo Descentralizado

API. Application Programming Interface (Interfaz de Programación de Aplicaciones)

ESP. Electronic Serial Protocol (Protocolo Electrónico Serial - aplicado al microcontrolador ESP32)

HTTP. Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto - para comunicación web)

ÍNDICE DE CONTENIDOS

CAPITULO I: ANTECEDENTES	24
1.1 Problema de investigación.....	24
1.2 Justificación	25
1.3 Objetivos	27
1.3.1 Objetivo General.....	27
1.3.2 Objetivos Específicos	27
1.3.3 Alcance	28
CAPÍTULO II. MARCO TEÓRICO	30
2.1. El agua	30
2.1.1. El agua potable	30
2.1.1.1. Sistema de agua potable	31
2.1.2.1. Métodos de Captación del agua.....	31
2.1.2.2. Estructura de potabilización del agua	32
2.2. Internet de las Cosas.....	33
2.2.1. Arquitectura IoT	34
2.2.2. Abordaje del Proyecto Basado en IoT.....	36
2.2.3. Sistemas Embebidos	37
2.2.3.1. Aplicación del Sistema embebido	37
2.2.3.2. Restricciones de un Sistema embebido.....	38
2.2.3.3. Componentes de un sistema embebido.....	40
2.2.4. Red de sensores Inalámbricos (WSN).....	41
2.2.4.1. Elementos de una WSN.....	41
2.2.4.2. Arquitectura de una WSN	41

2.2.4.3. Tipo de topologías de una WSN.....	42
2.2.5. Redes de área amplia de baja potencia (LPWAN)	43
2.2.5.1. Comparación de tecnologías LPWAN.....	44
2.2.5.2. LoRa	45
2.2.6. LoRaWAN.....	46
2.2.6.1. Arquitectura LoRaWAN	46
2.2.6.2. Bandas de frecuencia y Canales LoRa.....	47
2.2.6.3. Modulación LoRa.....	49
2.2.7. Fase de almacenamiento, Gestión y Visualización.....	49
2.2.8 Fase de Notificación	51
2.3. Metodología del diseño del Proyecto.....	52
2.3.1. Metodología en Cascada.....	53
CAPÍTULO III: DISEÑO DEL SISTEMA	55
3.1 Metodología de Diseño	56
3.2 Fase 1: Requerimientos	57
3.2.1 Generalidades	57
3.2.1.1 Escenario Actual	57
3.2.2 Medidas del sistema.....	60
3.2.3 Propósito del sistema	62
3.2.3 Beneficiarios o Stakeholders.....	62
3.2.3.1 Conectividad del operador.....	64
3.2.4 Construcción de atributos de los requerimientos.....	64
3.2.4.1 Nomenclatura de los requerimientos	65
3.2.5 Requerimientos de Stakeholders	65
3.2.6 Requerimientos del Sistema.....	67
3.2.7 Requerimientos de Arquitectura	69

3.2.8 Benchmarking (Selección de Hardware y Software).....	71
3.2.8.1 Hardware.....	71
3.2.8.2 Software	87
3.3. Fase 2: Diseño	92
3.3.1 Arquitectura del sistema	93
3.3.2 Diagrama de Bloques.....	96
3.3.1.1. Diagrama de Flujo del Sistema.....	98
3.3.3 Diagramas de interconexión	99
3.3.3.1. Nodo Sensor.....	100
3.3.4 Armado.....	107
3.3.4.1 Configuración del Nodo Sensor.....	108
3.3.4.2 Configuración Gateway Lora.....	115
3.3.4.4 Configuración ThingSpeak.....	128
3.3.5.5 Configuración de Página Web.....	133
3.3.5.6 Configuración de Notificaciones	137

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DE FUNCIONAMIENTO

.....	140
4.1. Verificaciones previo a la implementación.....	140
4.1.1 Verificación del Nodo Sensor	140
4.1.1.1 Hardware.....	141
4.1.1.2 Software	143
4.1.2 Verificación del Gateway	145
4.1.2.1 Hardware.....	146
4.1.2.2 Software	146
4.1.3 Pruebas de Funcionamiento del Sistema.....	150
4.1.3.1 Recolección y Visualización de Datos del Nodo Sensor	151

4.1.3.2 Recolección y Visualización de Datos de la estación Gateway	152
4.2. Fase 3: Implementación del Sistema	153
4.2.1 Definición de parámetros de pruebas	154
4.2.1.1. Rango de medición.....	154
4.2.1.2. Intervalo de tiempo	155
4.2.1.3. Umbrales de alarma.....	156
4.2.2 Toma de datos simulados previo a implementación.....	159
4.2.2.1 Pruebas de Precisión del Sensor	159
4.2.2.2 Pruebas a Largo Plazo	160
4.2.2.3 Toma de datos Simulada	160
4.2.3 Implementación de nodos del Sistema	162
4.2.4 Recolección de datos	166
4.2.4.1 Frecuencia de medición.....	166
4.2.4.2 Duración del muestreo	167
4.2.4.3 Estructura de Datos	167
4.2.5 Análisis de Datos Lectura de Datos	169
4.2.5.1 Gráficas de Tendencia Lineal	170
4.2.5.2 Histograma de Variación (Umbrales).....	171
4.3. Fase 4: Verificación de Funcionamiento	173
4.3.1. Verificación de datos receptados en plataformas de visualización	173
4.3.1.1. ThinSpeak	173
4.3.1.2. Página Web	174
4.2.3.3. Configuración de Alarma en base a umbrales	176
4.3.2. Certificación por el Operador Autorizado.....	178
4.3.1. Verificación de Frecuencia	181
4.4. Discusión: Comparativa entre inspección manual e inspección automatizada..	182

4.4.1 Descripción del Método de Inspección Manual	183
4.4.2. Sistema Automatizado	184
4.4.3 Ventajas de la Inspección Automatizada.....	184
4.5. Análisis Costo - Beneficio.....	185
4.5.1 Costos de Hardware	186
4.5.2 Costos de Software	187
CONCLUSIONES	188
RECOMENDACIONES	191
REFERENCIAS BIBLIOGRÁFICAS	193
ANEXOS	198
Anexo 1. Requerimientos	198
Anexo 2. Datasheets	203
Anexo 3. Códigos de Programación: Repositorio GitHub	207
Anexo 4. Recopilación de datos no procesados	207
Anexo 5. Guía rápida de Usuario	207

ÍNDICE DE TABLAS

Tabla 1 Tecnologías LPWAN	44
Tabla 2 Bandas de operación de LoRa en cada Región.....	48
Tabla 3 Metodologías más comunes en proyectos IoT	52
Tabla 4 Dimensiones y condiciones del desarenador	61
Tabla 5 Lista de Stakeholders de Sistema de medición de Nivel de Agua.....	63
Tabla 6 Definición de Acrónimos de Nomenclatura del sistema	65
Tabla 7 Requerimientos de Stakeholders	66
Tabla 8 Requerimientos del Sistema	68
Tabla 9 Requerimientos de Arquitectura	69
Tabla 10 Elección del microcontrolador del Nodo.....	73
Tabla 11 Selección de Sensor de Medición de Nivel	75
Tabla 12 Selección de tecnología LPWAN	77
Tabla 13 Selección del Módulo de Tecnología LoRa	78
Tabla 14 Consumo eléctrico en estación nodo	80
Tabla 15 Elección de microcontrolador Gateway	85
Tabla 16 Selección de software de estación de monitoreo	88
Tabla 17 Selección de software de visualización de datos	90
Tabla 18 Selección de software de notificación de datos	92
Tabla 19 Rangos de distancias típicas y de pruebas del sistema de medición de Nivel de Agua.....	155
Tabla 20 Umbrales mínimos y máximos en cm y m ³ del nivel de agua en desarenador	158
Tabla 21 Mediciones del nivel de agua en un entorno simulado.....	161
Tabla 22 Promedios semanales de las mediciones del nivel de agua.....	168
Tabla 23 Promedios mensuales de datos de nivel de agua	168
Tabla 24 Umbrales, manejos de errores y condiciones de alarma del sistema de medición de nivel de agua y alarma	177
Tabla 25 Comparativa de inspección manual vs automatizada del nivel de agua.....	184
Tabla 26 Costo de Hardware del sistema de medición del nivel de agua	186
Tabla 27 Costo de Software del sistema de medición del nivel de agua.....	187

ÍNDICE DE FIGURAS

Figura 1 Arquitectura de 3 capas para sistemas IoT.....	34
Figura 2 Arquitectura de una WSN	42
Figura 3 Arquitectura LoRaWAN	47
Figura 4 Fases de la Metodología en Cascada.....	54
Figura 5 Fases del modelo en cascada del sistema de Medición de Nivel de Agua.....	56
Figura 6 Ubicación de Captación de agua y Zona de Abastecimiento.....	59
Figura 7 Piscinas de contención del agua de captación (desarenador).....	60
Figura 8 Dimensiones del desarenador en cm.....	61
Figura 9 Microcontrolador ESP32 de 38 Pines	74
Figura 10 Sensor Ultrasónico HC-SR04	76
Figura 11 Módulo de Lora RYLR998 del interfaz 868/915 MHz	79
Figura 12 Cargador Solar para los nodos de recolección de datos del sistema de medición de nivel de agua.	82
Figura 13 Microcontrolador Raspberry Pi 4.....	86
Figura 14 Raspberry Pi - Sistema operativo Raspbian y lenguaje Python	90
Figura 15 Arquitectura del sistema de monitoreo y notificación del nivel de agua.	94
Figura 16 Diagrama de bloques basado en la Arquitectura IoT de 3 capas	97
Figura 17 Diagrama de Flujo del Sistema de Medición de Nivel de Agua en General	99
Figura 18 Identificación de Pines del ESP32 de 38 pines	101
Figura 19 Pines del sensor HC-SR04	101
Figura 20 Interconexión del HC-SR04 y el ESP32 de 38 pines.....	102
Figura 21 Identificación de pines del módulo lora RYLR998	103
Figura 22 Interconexión de Módulo LoRa al ESP32	104
Figura 23 Pines de la Raspberry Pi 4 Modelo B	105
Figura 24 Diagrama de Interconexiones del Nodo Gateway del Sistema de Medición de Nivel de Agua	107
Figura 25 Diagrama de Flujo del nodo Sensor	109
Figura 26 Diagrama de Interconexión del Nodo Sensor Completo	111
Figura 27 Configuración Inicial de Parámetros Lora en placa ESP32 del nodo sensor	112

Figura 28 Declaración y Configuración de Variables en placa ESP32 del nodo sensor	112
Figura 29 Configuración Inicial en la Función setup () en placa ESP32 del nodo sensor	113
Figura 30 Configuración Inicial del Módulo LoRa con Comandos AT en placa ESP32 del nodo sensor	114
Figura 31 Medición de Distancia e Indicador LED en placa ESP32 del nodo sensor	114
Figura 32 Envío de Datos y Control de Indicador LED en placa ESP32 del nodo sensor	115
Figura 33 Configuración Inicial de Parámetros, Declaración de Variables en placa ESP32 del nodo Gateway	116
Figura 34 Configuración Inicial del Sistema LoRa y Periféricos en placa ESP32 del nodo Gateway	117
Figura 35 Configuración Inicial en la void loop en placa ESP32 del nodo Gateway	118
Figura 36 Recepción y Procesamiento de Datos de Nodos LoRa con Indicadores Visuales en placa ESP32 del nodo Gateway	119
Figura 37 Envío de Datos de Nodos LoRa y Reinicio de Contador en placa ESP32 del nodo Gateway	120
Figura 38 Importación de librerías e inicio de programación del código Python en Raspberry Pi	121
Figura 39 Código de configuración de puertos seriales y análisis de datos en Raspberry Pi.....	122
Figura 40 Código de recepción e impresión de datos en Raspberry Pi.....	123
Figura 41 Código para establecimiento de conexión y envío de datos a ThingSpeak en Raspberry Pi	124
Figura 42 Código de establecimiento de Umbrales del nivel de agua en Raspberry Pi	125
Figura 43 Código de establecimiento de líneas de errores y reinicio de contadores en Raspberry Pi	126
Figura 44 Establecimiento de excepciones en el código en Raspberry Pi	127
Figura 45 Establecimiento de variables y parámetros para el envío de notificaciones en Raspberry Pi	127
Figura 46 Registro en la plataforma de ThingSpeak	129
Figura 47 Creación de Canal en ThingSpeak	130

Figura 48	Definición de los campos para datos en ThingSpeak	131
Figura 49	Configuración de metadatos en ThingSpeak	132
Figura 50	Obtención de clave API de escritura para envío de datos de la Raspberry Pi 4 a plataforma de ThingSpeak	133
Figura 51	Registro en GoDaddy para la creación de página web	134
Figura 52	Edición de página web de acuerdo con requerimientos	135
Figura 53	Obtención de etiqueta HTML de plataforma ThingSpeak.....	135
Figura 54	Inserción de código HTML en página WEB para visualización de imágenes	136
Figura 55	Publicación y Visualización de página Web.....	137
Figura 56	Dashboard de Sitio Web de Mailgun	138
Figura 57	Código de configuración para envío de correos desde la Raspberry al cliente	139
Figura 58	Conexiones de Hardware de nodo Sensor.....	142
Figura 59	Carcasas de los nodos para exteriores.....	143
Figura 60	Parámetros de configuración en nodo Emisor	144
Figura 61	Comandos para el establecimiento de conexión LoRa	145
Figura 62	Conexión del Hardware del nodo Gateway	146
Figura 63	Selección de parámetros del ESP32 del receptor Gateway.....	147
Figura 64	Configuración de parámetros para el funcionamiento de LoRa	148
Figura 65	Sistema de Medición de nivel de agua en completo en funcionamiento ...	149
Figura 66	Ruta de separación entre nodos principal y secundarios del sistema de Medición de Nivel de agua.....	149
Figura 67	Ruta de separación actualizada entre Nodos de Recolección y Nodo Gateway del Sistema de Medición de Nivel de agua	150
Figura 68	Toma y visualización de datos en nodo sensor.....	151
Figura 69	Recepción de los datos en el nodo Gateway ESP32	152
Figura 70	Recepción y visualización de datos en la terminal de la Raspberry pi 4 ...	153
Figura 71	Medidas del desarenador de agua del área de Captación.....	157
Figura 72	Ubicación de Nodos de Recolección de Datos en Desarenador del área de Captación.....	163
Figura 73	Orificios de separación entre estructura 1 y estructura 2 del Desarenador	164
Figura 74	Ubicación del sensor en superficie de estructura del Desarenador.....	165
Figura 75	Ubicación de Nodo Gateway	166

Figura 76 Gráfica Lineal de promedios semanales del nivel de agua	170
Figura 77 Representación Lineal del promedio mensual del nivel de agua	171
Figura 78 Representación gráfica de días con fallas en cada mes de prueba	172
Figura 79 Dashboard de datos públicos del nivel de agua de ThingSpeak	174
Figura 80 Visualización de Datos del nivel de agua en Página Web	175
Figura 81 Dashboard de Mailgun para la emisión de alertas de correo según umbrales de nivel de agua	176
Figura 82 Recepción de correos de Notificación ante alertas de Umbrales establecidos	178
Figura 83 Certificado de funcionamiento del sistema de medición de nivel de agua por parte del Operador del área de captación.....	179
Figura 84 Manipulación del nodo de recolección de datos de nivel de agua por parte del operador del área de captación	180
Figura 85 Captura de señal de radiofrecuencia entre nodos recolectores de datos y nodo Gateway mediante SDR Sharp	182
Figura 86 Medición Visual del Nivel de Agua en Desarenador del área de Captación	183

CAPITULO I: ANTECEDENTES

Este capítulo presenta los elementos clave que proporcionan una visión panorámica del desarrollo de este proyecto de integración curricular. Se abordan aspectos como la problemática identificada, los objetivos establecidos, así como el alcance y la justificación del estudio.

1.1 Problema de investigación.

El Gobierno Autónomo Descentralizado del Cantón Montufar a través del modelo de gestión de una Dirección, presta los servicios de agua potable al área urbana de la Ciudad de San Gabriel donde un aproximado de 3.800 usuarios se ven beneficiados (El Universo, 2019). Los últimos años se han propiciado mejoras en las redes de distribución de agua potable que sin lugar a duda han dado buenos resultados, pero, se sabe que aún existen ciertas deficiencias en el sistema, sobre todo en el área de captación y almacenamiento de agua que es la primera fase del sistema de distribución.

Según información del comercio La Hora (La Hora, 2022), generalmente las fuertes lluvias de determinadas épocas del año provocan derrumbes en sectores estratégicos de la captación que dejan sin suministro de agua a un 70% de la población de la ciudadanía que se beneficia de esta red de distribución. Actualmente el proceso de monitoreo y control de actuadores en la captación y almacenamiento de agua es llevado a cabo manualmente, lo que implica la constante presencia de un operador y por ende esto le genera pérdidas de tiempo innecesarias, así mismo como una total desinformación en periodos de descanso o la realización de otras actividades de trabajo. En base a lo expuesto, el problema radica en una mala optimización del monitoreo del sistema y la privación del agua potable en situaciones que pueden ser fácilmente solucionadas con rapidez.

Como una mejora de la eficiencia del proceso de monitoreo del nivel de agua potable en el área de la captación de la red de distribución media y baja, se plantea el diseño e implementación de un sistema inalámbrico basado en tecnologías IoT para el monitoreo y notificación del nivel de agua potable; el sistema realizará la recolección de datos en tiempo real, notificará automáticamente a los responsables de la administración del servicio de agua potable en caso de que el nivel de agua esté por encima o por debajo de los límites establecidos y contará con una interfaz de visualización para el público en general.

1.2 Justificación

El agua potable es un recurso indispensable y esencial para la salud, el bienestar, el desarrollo infantil, la productividad y la calidad de vida de las personas. Garantizar un suministro constante y confiable de agua potable es un objetivo prioritario que requiere inversiones, infraestructuras adecuadas y una gestión eficiente del recurso hídrico. Es primordial reconocer la importancia del agua como un derecho humano fundamental y promover su acceso equitativo y sostenible para todas las comunidades. Según lo que se menciona en el Plan Nacional del Buen Vivir “El agua potable es un recurso vital para garantizar la vida digna y la salud de la población. En el marco del Plan Nacional del Buen Vivir, el Gobierno de Ecuador reconoce la importancia de asegurar el acceso equitativo y sostenible al agua potable para todas las personas, especialmente aquellas que se encuentran en situaciones de vulnerabilidad (Consejo Nacional de Planificación (CNP), 2017)” El objetivo estratégico del Gobierno es fortalecer la gestión integral y sustentable del agua potable, asegurando su calidad, disponibilidad y acceso para todos los ciudadanos. Entre las acciones que implican el cumplimiento de este objetivo estratégico está la mejora en la infraestructura que radica dentro los planes a realizar en este proyecto.

Según el Plan de Desarrollo y Ordenamiento Territorial del Cantón Montufar se cuenta con la zona de distribución de Red Media y Zona de distribución de la Red Baja, las cuales son cubiertas por el recurso hídrico proveniente de la Quebrada el Oso ubicada en la comunidad Jesús de Gran Poder, la misma que abastece a un total de usuarios de 2.835, en este contexto el sistema de Tratamiento de agua Potable está compuesto por: captación, desarenador, aereador, planta de tratamiento (GAD Montufar, 2015). Este proyecto se enfoca en el beneficio de los usuarios expuestos y mejorar parte del sistema, enfocándose principalmente en el área de captación y desarenador. El sistema se enfoca en pequeñas mejoras que plantean una mejora funcional de sistema en distintos aspectos.

Se plantea la mejora del control y seguimiento del agua potable, debido a que con ello se garantizará un suministro adecuado y continuo hacia la población. Un sistema inalámbrico de monitoreo permitirá obtener datos precisos y actualizados en tiempo real, lo que mejorará significativamente el control sobre los niveles de agua y evitará desabastecimientos innecesarios.

Permitirá el ahorro de tiempo y recursos, la constante verificación del nivel de agua en la sección de captación y paso al desarenador se realiza de manera manual, lo que implica una inversión considerable de tiempo y recursos por parte del operador. Con la implementación de un sistema inalámbrico de monitoreo, se reducirá la necesidad de realizar mediciones manuales frecuentes, lo que permitirá al operador enfocarse en otras tareas relevantes para el sistema de abastecimiento de agua.

Con la implementación de alertas y recomendaciones automáticas se permitirá la notificación en caso de niveles altos o bajos de agua en el tanque de almacenamiento. Esto facilitará la detección temprana de situaciones anormales y la toma de medidas preventivas o correctivas de manera oportuna, evitando posibles problemas y garantizando un suministro continuo y confiable de agua potable.

Finalmente, el acceso a la información en tiempo real permitirá a diversas partes interesadas, como el personal encargado del control, autoridades locales y población beneficiada, acceder y visualizar de manera fácil y rápida, con lo cual se fomentará la transparencia y la participación ciudadana.

1.3 Objetivos

1.3.1 Objetivo General

Implementar un sistema inalámbrico basado en tecnologías IoT para el monitoreo y notificación del nivel de agua potable en un tanque de almacenamiento del sector Jesús del Gran Poder en la provincia del Carchi, con el fin de mejorar la eficiencia del proceso de monitoreo.

1.3.2 Objetivos Específicos

- Analizar información bibliográfica sobre tecnologías y herramientas disponibles que se adapten a la medición y notificación del nivel del agua, para hacer elección de la opción que mejor se ajuste a las necesidades del proyecto.
- Seleccionar los sensores y dispositivos inalámbricos adecuados para la recolección de datos del nivel del agua en tiempo real de acuerdo con la tecnología planteada.
- Interconectar los nodos de medición al sistema central de monitoreo y notificación automática para alertar al operario encargado sobre niveles excesivos o insuficientes.
- Realizar la instalación del sistema en las celdas de almacenamiento de la captación, verificando el correcto funcionamiento y configurando adecuadamente la escala de los niveles de notificación.

1.3.3 Alcance

El presente proyecto plantea como objetivo la implementación de un sistema inalámbrico de monitoreo del nivel de agua potable en el área de la captación de las redes de distribución media y baja que abastecen la ciudad de San Gabriel, la captación se ubica en la Loma Guagua en la comunidad Jesús del Gran Poder la cual a su vez se encuentra en el cantón Montufar, provincia del Carchi. El sistema por implementar tiene el fin de mejorar el control y seguimiento del nivel de agua, evitando desabastecimientos innecesarios y proporcionando una mejora en el uso del tiempo del operador.

En primera instancia se plantea realizar una investigación exhaustiva sobre las diferentes tecnologías y herramientas disponibles para la medición y notificación del nivel del agua. Se incluye el estudio de sensores, protocolos de comunicación inalámbrica, plataformas IoT, sistemas de notificación y finalmente, el estado y funcionamiento del área de captación.

Basándose en la información recopilada se procederá a la elección de los sensores y dispositivos inalámbricos que sean apropiados para la medición en tiempo real del nivel del agua en el área de captación, de forma específica en la celda de almacenamiento. Estos dispositivos deben ser compatibles con la tecnología inalámbrica y capaces de transmitir los datos de manera confiable. Se hará uso de la tecnología de comunicación LoRa la cual es ideal para redes de amplia cobertura y baja potencia, además de ajustarse perfectamente al perfil geográfico del área de captación ya que permite establecer comunicaciones de larga distancia de manera confiable y segura, esta tecnología de comunicación hace uso del protocolo de red LoRaWan el cual ofrece las características de consumo de energía optimizado, seguridad extremo a extremo y capacidad para admitir una gran cantidad de dispositivos en una red.

Seguidamente, se diseñará el sistema iniciando con la conexión entre los distintos dispositivos electrónicos a usar, la correcta configuración para una buena comunicación entre nodos secundarios y nodo principal, la configuración de plataforma de visualización y emisión de notificación de acuerdo con los datos proporcionados por los nodos secundarios. De acuerdo al uso de la tecnología de comunicación inalámbrica a usar LoRa, se considera la posibilidad de utilizar los componentes y herramientas siguientes: NodeMCU como microcontrolador en cada uno de los nodos, ya que brinda bastas funcionalidades y son compatibles con el módulo de comunicación, módulos LoRa para el establecimiento de la comunicación inalámbrica, sensores de nivel de agua ultrasónicos, antenas para la transmisión y recepción de datos a grandes distancias, baterías para los nodos secundarios y de forma adicional de acuerdo al procesamiento del sistema se plantea evaluar el uso de Raspberry Pi en el nodo central, como un microcontrolador de mayor potencia para el acceso a servicios en la nube.

Implementar los nodos de comunicación en el tanque de almacenamiento de agua potable de la captación. Asegurarse de que los componentes estén conectados y funcionando adecuadamente. Además, configurar de forma correcta la escala de los niveles de notificación para alertar al operario encargado en caso de niveles excesivos o insuficientes de agua. Cabe mencionar que la distancia entre el nodo central y nodos secundarios es de 750 metros aproximadamente. Una vez implementado el sistema se realizará pruebas para verificar el correcto funcionamiento del sistema de monitoreo y la plataforma de visualización, el sistema será puesto en marcha y capacitará sobre la funcionalidad al personal encargado del control como el seguimiento del agua potable de la captación.

CAPÍTULO II. MARCO TEÓRICO

La siguiente fundamentación teórica se centra en tres temas principales. El primero se enfoca en el recurso vital del agua y aborda temas como su consumo, la estructura de los sistemas de agua potable y las técnicas de captación. El segundo, se enfoca en la situación actual del Internet de las cosas (IoT), un fenómeno tecnológico que está cambiando numerosas industrias, como la distribución de agua. Esto incluye la arquitectura de Internet de las cosas, sus aplicaciones para monitorear el suministro de agua y partes particulares como sistemas embebidos, redes de sensores (WSN) y redes de área amplia de baja potencia (LPWAN), con un enfoque en la tecnología LoRaWAN. Finalmente, se analizan las metodologías de diseño de proyectos, como la metodología en cascada, que pueden guiar la implementación efectiva de soluciones tecnológicas en este contexto complejo e interdisciplinario.

2.1. El agua

El agua es esencial para la vida en la Tierra, participando en procesos biológicos vitales y siendo un recurso vital para la agricultura, la industria y el consumo humano. Su ciclo es fundamental para mantener el equilibrio de los ecosistemas terrestres. La gestión sostenible del agua es crucial para asegurar su disponibilidad futura y mantener la salud del planeta y sus habitantes (Saguapac, 2023).

2.1.1. El agua potable

El agua potable es aquella que está destinada para el consumo humano y cumple con los estándares de calidad y seguridad establecidos. Se caracteriza por ser libre de contaminantes y apta para el consumo directo o para uso en procesos industriales y agrícolas. Su importancia radica en su vital papel para mantener la salud y el bienestar de las poblaciones, siendo fundamental para la hidratación, la higiene y el funcionamiento adecuado del organismo humano. La disponibilidad de agua potable de calidad es un

derecho humano básico y un factor clave para el desarrollo sostenible de las comunidades (Salas-Salvadó et al., 2020).

2.1.1.1. Sistema de agua potable

La potabilización del agua es un procedimiento esencial para asegurar su idoneidad para el consumo humano, eliminando contaminantes y microorganismos que puedan representar un peligro para la salud. Este proceso abarca una serie de fases que pueden comprender desde la filtración hasta la desinfección, así como la eliminación de elementos químicos no deseados. La purificación del agua desempeña un papel vital en la prevención de enfermedades transmitidas por el agua y en el cumplimiento de los estándares de calidad establecidos para su consumo. Es un componente importante en la administración del recurso hídrico y en la promoción de la salud pública (Lama, 2023).

2.1.2.1. Métodos de Captación del agua

Los métodos que se utilizan en la captación de agua son los siguientes:

- ***Captación de Agua de Lluvia:*** Recolectar y almacenar agua de las precipitaciones mediante sistemas instalados en techos u otras superficies. Esta técnica dirige el agua hacia dispositivos de almacenamiento, como cisternas o tanques, para su posterior uso en diversas actividades, como domésticas, agrícolas o de riego (Ricardo et al., 2015).
- ***Perforación de Pozos:*** Acceso a aguas subterráneas mediante la excavación de pozos profundos en el suelo hasta alcanzar acuíferos subterráneos que contienen agua. Esta agua subterránea puede ser extraída mediante bombas u otros sistemas de bombeo para usos como el abastecimiento de agua potable, el riego agrícola o aplicaciones industriales.

- ***Captación de Agua de Manantiales:*** Recolectar agua proveniente de fuentes naturales, como manantiales o vertientes, que son afloramientos naturales de agua subterránea. La captación de agua de manantiales puede llevarse a cabo mediante sistemas de canalización que dirigen el agua hacia puntos de almacenamiento o distribución. (Ministerio de Agricultura y Ganadería Costa Rica, 2010).
- ***Captación de agua Superficial:*** La recolección de agua superficial implica la captación y almacenamiento del agua proveniente de fuentes naturales como ríos, quebradas y arroyos. Este proceso se realiza mediante la construcción de estructuras adecuadas para dirigir el flujo de agua hacia dispositivos de almacenamiento, como tanques o embalses.

Basado en los puntos anteriormente mencionados, la captación de agua superficial es la principal forma de captación de agua, según datos proporcionados por (GAD Montufar, 2015) en el cantón Montufar, las precipitaciones generan un flujo de agua a través de las quebradas, originadas por el relieve irregular del territorio que se origina en las zonas altas de los páramos. Esto resulta en la formación de una extensa red de cursos de agua que alimentan importantes cuencas hidrográficas, como los ríos Carchi y Mira, caracterizados por su descarga hacia el mar. Esta red hídrica incluye aproximadamente veinte microcuencas con diversos niveles de complejidad y caudales suficientes para satisfacer las necesidades de agua del cantón Montufar y por ende la Ciudad de San Gabriel en la cual se enfoca el presente proyecto.

2.1.2.2. Estructura de potabilización del agua

Poniendo como referencia el sistema de tratamiento de agua para las redes de distribución media y baja del cantón Montufar, se cuenta con 4 unidades que permiten la purificación del agua:

- **Captación:** Obtención del agua de una fuente natural, como un río o una quebrada, para su tratamiento posterior.
- **Desarenador:** Eliminación de partículas de arena, lodo y otros sedimentos presentes en el agua cruda mediante un proceso de sedimentación o filtración.
- **Aereador:** Introducción de oxígeno al agua para mejorar su sabor, eliminar olores y promover la oxidación de sustancias disueltas como el hierro y el manganeso.
- **Planta de Tratamiento:** Instalación donde se lleva a cabo el proceso de purificación del agua, que puede incluir diversas etapas como la coagulación, la floculación, la sedimentación, la filtración y la desinfección, con el objetivo de eliminar impurezas, microorganismos patógenos y compuestos químicos indeseables para hacerla apta para el consumo humano (GAD Montufar, 2015).

2.2. Internet de las Cosas

El Internet de las Cosas (IoT) se refiere a una red de dispositivos interconectados que pueden comunicarse y compartir datos entre sí a través de Internet. La definición y los principios de funcionamiento del IoT se basan en las capacidades de conectividad y comunicación de diversos dispositivos. Los dispositivos IoT pueden ir desde objetos cotidianos hasta herramientas industriales sofisticadas, todos equipados con sensores y características de conectividad para recopilar e intercambiar datos (Cepeda, 2018).

Los principios de funcionamiento del IoT incluyen:

- **Conectividad:** Los dispositivos IoT están conectados a Internet u otros dispositivos para habilitar el intercambio de datos.
- **Sensores y Recolección de Datos:** Los dispositivos IoT utilizan sensores para recopilar datos del entorno o del propio dispositivo.
- **Procesamiento de Datos:** Los datos recopilados se procesan localmente o en la nube para obtener información o activar acciones.

- **Comunicación:** Los dispositivos IoT se comunican entre sí o con sistemas centralizados para compartir datos y recibir comandos.
- **Automatización:** Los sistemas de IoT a menudo involucran automatización donde los dispositivos pueden actuar según reglas predefinidas o datos de entrada.

2.2.1. Arquitectura IoT

La arquitectura para sistemas de IoT se conceptualiza como una estructura en capas, dividida en diferentes capas funcionales para asignar y distribuir responsabilidades a lo largo del sistema. Esta arquitectura debe satisfacer varios requisitos clave, incluyendo flexibilidad, escalabilidad, seguridad e interoperabilidad, dada la diversidad de dispositivos disponibles en estos sistemas. Además, es necesario que la arquitectura garantice la confiabilidad en el almacenamiento y la calidad del servicio. Como se menciona en (Fernández, 2021), aunque no existe un estándar universal para esta arquitectura, hay propuestas ampliamente aceptadas, como la división en tres capas: la capa de percepción, la capa de red y la capa de aplicación.

Figura 1

Arquitectura de 3 capas para sistemas IoT



Nota. Adaptado de “Utilización de dispositivos móviles como herramienta de sensado en aplicaciones de IoT”, por (Fernández, 2021).

La arquitectura de tres capas en IoT como se muestra en la **Figura 1** se centra en recopilar información del entorno, transmitir los datos a través de internet y permitir el acceso a estos datos por parte de usuarios u otros dispositivos. Aunque es una estructura básica, el rápido crecimiento en el área de IoT ha generado la necesidad de arquitecturas más complejas para abordar nuevos desafíos tecnológicos.

- **La capa de percepción:** Es la primera capa en la arquitectura de sistemas de IoT y se encarga de la digitalización del mundo real. Aquí es donde se encuentran dispositivos físicos como sensores y actuadores, los cuales recolectan datos del entorno, como temperatura, humedad o ubicación. Estos datos son luego digitalizados y enviados a la capa de red para su procesamiento.
- **La capa de red:** Es responsable de transportar los datos recopilados por la capa de percepción hacia la capa de aplicación, así como de gestionar la comunicación entre los dispositivos del sistema IoT. También se encarga del filtrado y la agregación de datos. Los medios de comunicación pueden ser alámbricos o inalámbricos, y las tecnologías comunes incluyen RFID, ZigBee, Wi-Fi, entre otras.
- **La capa de aplicación:** Es la capa superior de la arquitectura y proporciona servicios solicitados por los usuarios. Aquí se definen las aplicaciones que utilizan tecnología IoT, como casas inteligentes, ciudades inteligentes, salud inteligente, entre otras. Esta capa garantiza la integridad, confidencialidad y autenticidad de los datos, y permite la comunicación entre procesos utilizando diversos protocolos como HTTP, MQTT y CoAP.

Esta arquitectura es planteada como la más sencilla, pero se puede dividir en más capas de ser necesario, esto de acorde a las necesidades del proyecto, en este caso este

modelo de capas servirá como una guía para la recolección de la información necesaria que permita comprender los requerimientos del sistema a desarrollar.

2.2.2. Abordaje del Proyecto Basado en IoT

En el panorama actual, el uso del Internet de las Cosas (IoT) está transformando rápidamente la manera en que se gestionan y utilizan los recursos hídricos, especialmente en el ámbito del agua potable. El IoT ofrece una variedad de tecnologías y aplicaciones innovadoras que van desde la optimización del consumo hasta el monitoreo en tiempo real del agua. Algunas de las aplicaciones más relevantes del IoT en el contexto del agua potable.

- ***Lectura Eficiente de Contadores:*** En diversas áreas de países de América Latina, se ha implementado el uso de aplicaciones móviles para la lectura de contadores de agua potable. Estas aplicaciones permiten a los operadores registrar los consumos de agua con mayor precisión y eficiencia, lo que reduce errores y optimiza el proceso de lectura (Orozco, 2023).
- ***Diversificación del Suministro:*** En regiones con escasez de agua, se está empleando la tecnología IoT para analizar y diversificar las fuentes de suministro de agua. Esto implica combinar fuentes como aguas superficiales, trasvasadas y tratadas para garantizar un suministro sostenible y seguro para la población (Bernabé-Crespo, 2022).
- ***Medición de Gases Contaminantes:*** En varios países se ha comenzado a implementar sistemas IoT para la medición de gases contaminantes en el agua y en el aire. Estos sistemas permiten monitorear y reducir la contaminación ambiental, lo que es especialmente importante en áreas urbanas donde la calidad del aire y del agua puede verse afectada por diversas actividades industriales y de transporte (Barrera & Leonardo, 2018).

- **Monitoreo del Nivel del Agua:** El IoT también se utiliza para monitorear el nivel del agua de manera continua en diferentes entornos, lo que resulta crucial para la gestión eficiente de los recursos hídricos y la prevención de posibles problemas de abastecimiento. Sensores estratégicamente ubicados pueden medir el nivel del agua en embalses, tanques de almacenamiento, ríos, y otros cuerpos de agua, proporcionando información en tiempo real sobre los cambios en los niveles de agua (Abraham & Huaman, 2023).

Estas aplicaciones muestran cómo el IoT se utiliza actualmente en diversos países para mejorar la gestión del agua potable, desde la lectura precisa de contadores hasta el control de la calidad del recurso y la reducción de impactos ambientales.

2.2.3. Sistemas Embebidos

Los sistemas embebidos desempeñan un papel importante en el ámbito del Internet de las Cosas (IoT), donde se integran en dispositivos conectados a la red para facilitar la comunicación de datos y la automatización de procesos. En el contexto de IoT, estos sistemas son esenciales para la recopilación eficiente, el procesamiento y la transmisión de información, así como para el control de dispositivos y la toma de decisiones (Delgado & Banguera, 2019).

2.2.3.1. Aplicación del Sistema embebido

Los sistemas embebidos tienen una importancia en el suministro de agua potable al posibilitar el control, monitoreo y gestión eficiente de los recursos hídricos. En este contexto, se pueden identificar diversas aplicaciones para estos sistemas:

- **Gestión Eficiente de Recursos:** Estos sistemas contribuyen a una gestión más efectiva de los recursos hídricos al automatizar procesos como la distribución de agua y el control de presión en las redes de suministro. Esto ayuda a optimizar el

uso del agua, reducir pérdidas y mejorar la eficiencia operativa de los sistemas de abastecimiento (Romero, 2021).

- **Monitoreo de la Calidad del Agua:** Los sistemas embebidos pueden ser utilizados para supervisar la calidad del agua en tiempo real, aspecto fundamental para asegurar su aptitud para el consumo humano. Integrando sensores para medir parámetros como temperatura, pH y presencia de contaminantes, permiten una vigilancia continua y la detección temprana de posibles riesgos (Bravo et al., 2023)
- **Optimización de Sistemas de Abastecimiento:** Mediante la implementación de sistemas embebidos, es posible evaluar y mejorar los sistemas de suministro de agua potable para comunidades, como se ha evidenciado en el caso de la comunidad Caiza D en Bolivia. Estos sistemas permiten determinar aspectos clave como las necesidades de potencia de las bombas, el diámetro de las tuberías y la capacidad de los tanques, garantizando así un suministro adecuado de agua potable (Romero, 2021).

El presente proyecto se enfoca en la Optimización de Sistemas de Abastecimiento ya que la medición del nivel de agua con su respectiva notificación de anomalías al operador es necesario mantener un nivel de agua específico ayudando al correcto funcionamiento del sistema de distribución.

2.2.3.2. Restricciones de un Sistema embebido

En base a diversas investigaciones un sistema embebido puede presentar varias restricciones en un sistema de abastecimiento de agua potable, las cuales se presentan a continuación:

- **Recursos hídricos disponibles:** La calidad y cantidad de agua disponibles pueden restringir el diseño del sistema de abastecimiento. La presencia de contaminantes

o la baja calidad del agua puede requerir tratamientos adicionales, aumentando así los costos y la complejidad del sistema embebido (Salas & Martinez, 2022).

- **Infraestructura preexistente:** La infraestructura ya existente en el área de implementación puede influir en el diseño del sistema embebido. Aspectos como la disponibilidad de energía eléctrica, la distancia entre puntos de captación y distribución, y las características de las tuberías y bombas pueden limitar las opciones de diseño (Llashac, 2017).
- **Costos:** Los costos asociados con la instalación y mantenimiento del sistema embebido pueden ser una restricción significativa. Es esencial considerar los costos de adquisición, instalación, operación y mantenimiento para garantizar la viabilidad y accesibilidad del sistema.
- **Regulaciones y normativas:** Las normativas locales y nacionales relacionadas con el suministro de agua potable deben tenerse en cuenta durante el diseño del sistema embebido (Salguero, 2020).
- **Disponibilidad de tecnología:** La disponibilidad de tecnología en el mercado puede limitar las opciones de diseño y la implementación del sistema embebido.

En base a esta información es importante mencionar que la principal restricción que se encuentra en la ejecución del proyecto es la infraestructura preexistente, esto debido a que el lugar no cuenta con energía eléctrica, lo cual hace necesario el uso de baterías, perjudicando así de cierta manera la vida útil del sistema, de igual forma una de las restricciones en caso de proyectos futuros es la rusticidad de los actuadores con los que cuenta el sistema, para lo cual es netamente necesario la intervención humana.

2.2.3.3. Componentes de un sistema embebido

Según información proporcionada en (Revelo & Ordoñez, 2012) los elementos de un sistema embebido pueden variar según la aplicación específica, pero en términos generales, un sistema embebido incluye los siguientes componentes:

- **Unidad de procesamiento:** Es la parte central del sistema, responsable de ejecutar instrucciones y controlar las operaciones. Puede ser un microprocesador o un microcontrolador, según las necesidades del sistema.
- **Memoria:** Almacena los datos y las instrucciones necesarias para el funcionamiento del sistema, como memoria flash, RAM o EEPROM.
- **Entradas y salidas:** Permiten la interacción del sistema con el entorno, pudiendo ser analógicas o digitales tanto para entrada como para salida.
- **Interfaz de comunicación:** Facilita la comunicación del sistema con otros dispositivos o sistemas, a través de interfaces como serial, USB, Ethernet, Wi-Fi o Bluetooth.
- **Fuente de energía:** Suministra la energía necesaria para el funcionamiento del sistema, como baterías, adaptadores de corriente o generadores.
- **Software:** Conjunto de instrucciones que controla el sistema, programado en lenguajes de bajo o alto nivel como ensamblador, C o Python.
- **Sensores:** Detectan y miden variables del entorno, como temperatura, presión o humedad.
- **Actuadores:** Permiten al sistema interactuar con el entorno, como motores, luces o pantallas.

En lo que respecta a la selección de cada componente a usar en el diseño del sistema del proyecto, se presentará una sección dentro del capítulo 3 en el cual se realizará una breve descripción de cada componente principal con el cual se trabajará.

2.2.4. Red de sensores Inalámbricos (WSN)

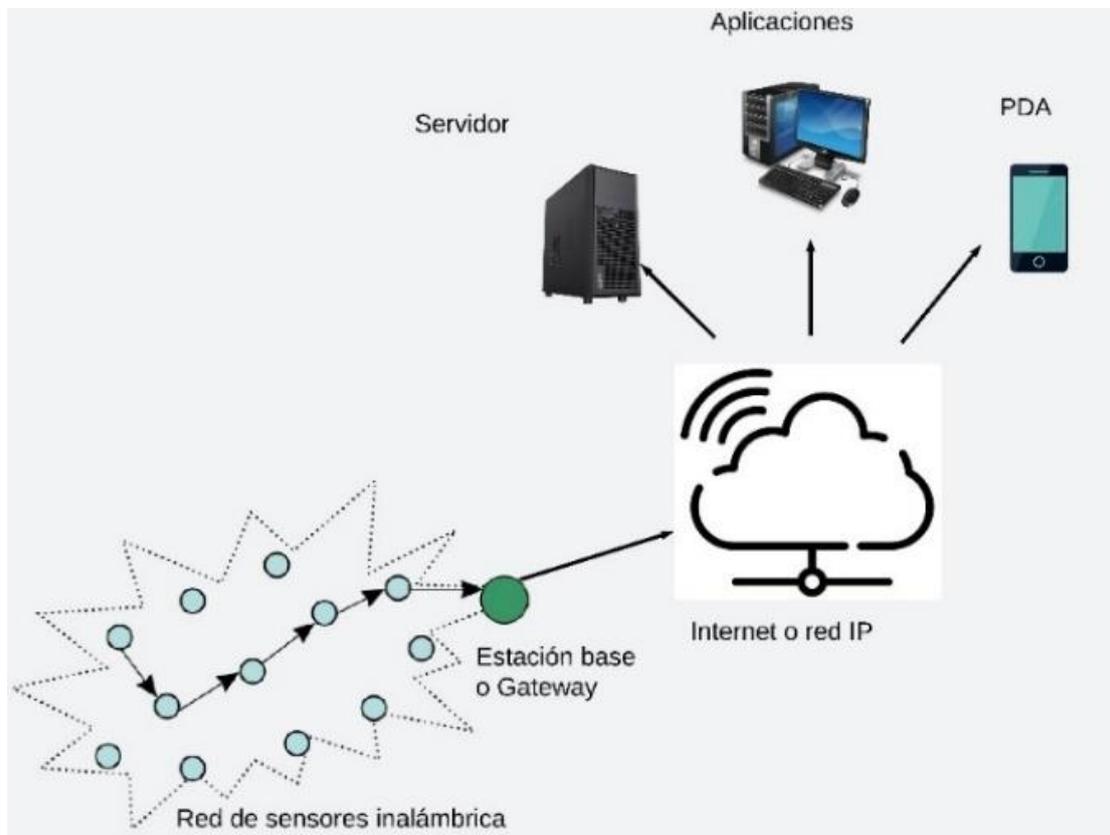
Las Redes de Sensores Inalámbricos (WSN) son sistemas que emplean múltiples nodos de sensores para recolectar información de un entorno específico y transmitirla de manera inalámbrica a un punto central para su análisis. Estas redes son ampliamente utilizadas en diversos campos, como la industria petrolera, la agricultura de precisión y el monitoreo ambiental, entre otros (Dender-Zurita et al., 2018; López et al., 2022).

2.2.4.1. Elementos de una WSN

Los elementos esenciales de una Red de Sensores Inalámbricos (WSN) incluyen nodos de sensores, estaciones base y protocolos de comunicación inalámbrica. Los nodos de sensores recopilan datos ambientales y los transmiten mediante tecnología inalámbrica, equipados con sensores, procesadores y fuentes de energía. Las estaciones base gestionan la recolección y procesamiento de datos. Los protocolos de comunicación inalámbrica, como facilitan una transmisión eficiente de datos en la red. En conjunto, estos elementos forman una infraestructura que permite la adquisición, transmisión y procesamiento de información en tiempo real dentro de la WSN (Philco & Orlando, 2023).

2.2.4.2. Arquitectura de una WSN

Las Redes de Sensores Inalámbricos (WSN) están conformadas por nodos de sensores distribuidos en el entorno, estaciones base encargadas del procesamiento de datos y una infraestructura de comunicación inalámbrica que enlaza los nodos con las estaciones base como se muestra en la **Figura 2**. Estos nodos recolectan información ambiental, la cual es transmitida a las estaciones base para su análisis, utilizando protocolos de comunicación inalámbrica como Zigbee, Bluetooth o LoRaWAN para una transmisión eficiente de datos. La arquitectura de una WSN se enfoca en la captura de datos, su procesamiento centralizado y la comunicación inalámbrica, todo ello de gran importancia para un funcionamiento preciso y efectivo en diversos sectores.

Figura 2*Arquitectura de una WSN*

Nota. Adaptado de “Sistema de Red de Sensores Inalámbricos (WSN) portátil aplicado al monitoreo ambiental en una ciudad, como complemento a las estaciones fijas” por (Valencia, 2019).

2.2.4.3. Tipo de topologías de una WSN

Según se menciona en (Pérez et al., 2014) las disposiciones de los nodos de sensores dentro de una Red de Sensores Inalámbricos (WSN), conocidas como topologías de red, delinean cómo están organizados y conectados en la red. Algunas configuraciones comunes son las siguientes:

- **Topología de Estrella:** En esta estructura, todos los nodos de sensores se enlazan a una estación base central. Aunque es sencilla de administrar, puede presentar limitaciones en términos de alcance y escalabilidad.

- **Topología de Malla:** En esta disposición, los nodos de sensores se entrelazan entre sí, proporcionando múltiples rutas de comunicación redundantes. Esto aumenta la resistencia de la red, ya que los datos pueden encontrar diversos caminos para llegar a su destino.
- **Topología de Árbol:** En esta configuración, los nodos de sensores se estructuran jerárquicamente como las ramas de un árbol, con un nodo raíz que se comunica con nodos secundarios. Aunque eficiente en el consumo de energía, puede tener limitaciones en términos de escalabilidad y resistencia.

2.2.5. Redes de área amplia de baja potencia (LPWAN)

Como se menciona en el artículo de (Montes & Camacho, 2020), las LPWAN (Redes de Área Amplia de Baja Potencia) son utilizadas en la comunicación inalámbrica para conectar dispositivos IoT (Internet de las Cosas) en aplicaciones que demandan un consumo mínimo de energía y una prolongada duración de la batería. Estas redes se dividen en dos categorías: licenciadas y no licenciadas.

Las LPWAN licenciadas, como EC-GSM-IoT y NB-IoT, emplean espectro asignado por reguladores como la FCC, ofreciendo amplia cobertura y capacidad, pero requieren una inversión inicial más alta. Por otro lado, las LPWAN no licenciadas, como LoRa y Sigfox, utilizan espectro no asignado, siendo más económicas inicialmente, pero con capacidades y cobertura limitadas. Sigfox, con su modulación de banda ultra estrecha, es adecuada para aplicaciones simples como alarmas, mientras que LoRa, con su espectro ensanchado, es ideal para distancias largas y manejo de múltiples dispositivos. Aunque el alcance de las LPWAN puede variar según tecnología y entorno, es importante considerar despliegues específicos y factores externos que puedan influir.

2.2.5.1. Comparación de tecnologías LPWAN

En la **Tabla 1** se expone una tabla comparativa de las tecnologías disponibles y más comunes, en esta se expone aspectos como la calidad del servicio, la duración de la batería, la latencia, la escalabilidad, la capacidad de carga útil, la cobertura, el alcance, la implementación y el costo, esto permitirá conocer de mejor forma las ventajas de la tecnología seleccionada frente a las otras existentes (Montes & Camacho, 2020).

Tabla 1

Tecnologías LPWAN

Tecnología/Parámetro	LoRa	SigFox	NB-IoT
Rango	5Km (Urbano) 20Km (Rural)	10Km (Urbano), 40Km (Rural)	1 Km (Urbano), 10Km (Rural)
Modulación	CSS	BPSK	QPSK
Cobertura	157dB	153dB	164dB
Max. velocidad de datos	50Kbps	100bps	250Kbps
Ancho de banda	250KHz y 125 KHz	100Hz	200KHz
Potencia Max. Transmitida	14dB	20dB	23dB
Max. Longitud de carga útil	243 bytes	12 bytes (Subida), 8 bytes (Bajada)	1600 bytes
Frecuencia	ISM-no licenciado 433Mhz, 868Mhz, 915MHz	ISM-no licenciado 868-869 MHz 902-928 MHz	Licenciado, banda LTE, banda de guarda
Max mensajes/día	Sin limites	140 (Subida), 4 (Bajada)	Sin Limites
Inmunidad de interferencia	Muy Alta	Muy Alta	Baja
Autenticación y Cifrado	AES 128 bit	No soportado	Encriptación LTE

Nota. Adaptado de (Montes & Camacho, 2020).

Para el presente proyecto LoRa sería preferible en un sistema de monitoreo de nivel de agua potable debido a su mayor rango de cobertura tanto en entornos urbanos como rurales (5km y 20km respectivamente), lo que garantiza una comunicación

confiable incluso en áreas remotas. Además, su capacidad para manejar cargas útiles más grandes (243 bytes) permite la transmisión de datos más detallados y complejos sobre el nivel de agua y su calidad. También ofrece una mayor inmunidad a la interferencia y soporte para autenticación y cifrado AES de 128 bits, lo que garantiza la seguridad de los datos transmitidos en el sistema de monitoreo. Estas características se adecúan de manera precisa a las necesidades del proyecto.

2.2.5.2. LoRa

LoRa, abreviatura de Long Range, es una tecnología de comunicación inalámbrica de largo alcance que no consume mucha energía y está diseñada para aplicaciones en las redes de sensores y en Internet de las cosas (IoT). La capacidad de transmitir datos a largas distancias destaca esta tecnología, incluso en entornos rurales o remotos donde la conectividad tradicional es limitada (Dunford Pabst, 2022).

Esta tecnología cuenta con ciertos principios de funcionamiento los cuales son:

- La modulación de espectro ensanchado se utiliza para enviar señales a través del aire por LoRa.
- Funciona en bandas de frecuencia no licenciadas, lo que lo hace más fácil de usar en varios lugares.
- Permite una comunicación eficiente y de largo alcance utilizando la técnica de modulación LoRa (Long Range).
- La tecnología LoRa mejora la eficiencia energética al transmitir señales en sub-bandas de frecuencia estrechas y de baja potencia.
- Los dispositivos LoRa pueden comunicarse directamente entre sí o a través de una red LoRaWAN, una infraestructura de red de área amplia y baja potencia diseñada para soportar dispositivos IoT (Luis et al., 2022).

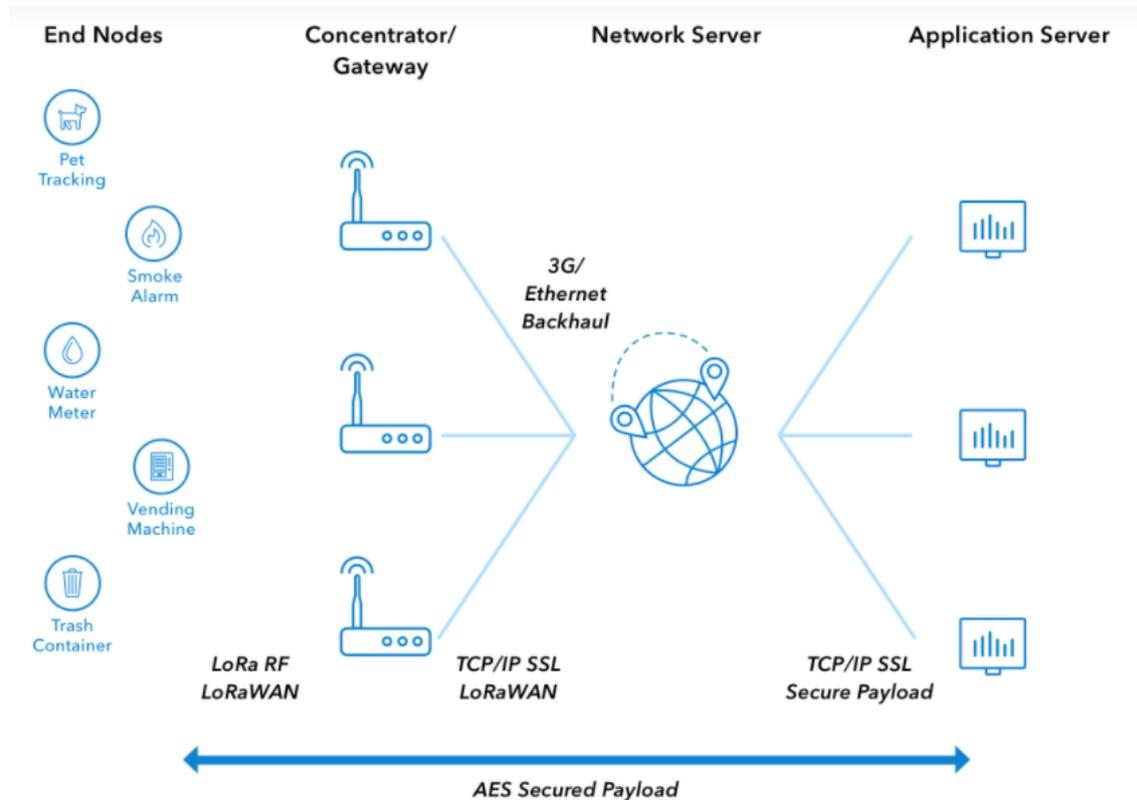
2.2.6. LoRaWAN

El protocolo LoRaWAN es una especificación desarrollada por LoRa Alliance que se basa en el esquema de modulación LoRa. Las redes LoRaWAN suelen tener una topología en estrella, donde los dispositivos finales se comunican con un gateway central, que a su vez retransmite los mensajes hacia un servidor central. Los dispositivos finales establecen una conexión inalámbrica LoRa con el gateway, el cual transmite la información hacia otros elementos que lo soliciten a través de conexiones IP.

En LoRaWAN, los dispositivos finales cambian de canal de forma pseudoaleatoria utilizando velocidades de datos disponibles que oscilan entre 0.3 kbps y 50 kbps, optimizando así el rango de comunicación y la duración del mensaje. Además, LoRaWAN implementa un esquema de Adaptive Data Rate (ADR) que permite gestionar la velocidad de datos en cada dispositivo final de manera adaptativa. Esta tecnología facilita la creación de redes eficientes y de larga duración para aplicaciones de Internet de las Cosas (IoT), permitiendo una comunicación inalámbrica confiable y de bajo consumo de energía (Palanca, 2019).

2.2.6.1. Arquitectura LoRaWAN

La arquitectura LoRaWAN se basa en un modelo llamado Star of Stars, que permite una comunicación eficaz entre dispositivos y un bajo consumo de energía en los dispositivos finales. Esta estructura consta de cuatro componentes principales: dispositivos finales, nodos o motes, Gateway y servidores de red y de aplicación tal como se evidencia en la **Figura 3**. Los nodos recopilan información de sensores y la envían a los Gateways siguiendo el protocolo LoRaWAN, los Gateways transmiten la información sin procesar a un servidor de red a través de una interfaz de back-haul de alto rendimiento, como Ethernet o 3G/4G, el servidor de red recibe los datos, los decodifica, aplica protocolos de seguridad y los hace disponibles para las aplicaciones (Idrovo, 2021).

Figura 3*Arquitectura LoRaWAN*

Nota. Tomado de (Idrovo, 2021) .

2.2.6.2. Bandas de frecuencia y Canales LoRa

Las comunicaciones inalámbricas, especialmente en el ámbito del Internet de las Cosas (IoT), dependen en gran medida de las bandas de frecuencia asignadas para garantizar una transmisión eficaz y sin interferencias. En diferentes partes del mundo, se han designado rangos específicos de frecuencia para este propósito. En la **Tabla 2** que se muestra a continuación, se detalla las bandas de frecuencia asignadas para China, Europa y Estados Unidos y América, lo que proporciona una visión general de las frecuencias disponibles para el despliegue de tecnologías inalámbricas como LoRa, Sigfox y otros protocolos de comunicación IoT.

Tabla 2

Bandas de operación de LoRa en cada Región.

Región	Banda
China	De 779 MHz hasta 787 MHz
Europa	De 863 MHz hasta 870 MHz
Estados Unidos y América	De 902 MHz hasta 928 MHz

Nota. Adaptado de (Bertoletti, 2019).

LoRa, como tecnología de comunicación inalámbrica para IoT, opera en diferentes frecuencias que están definidas para cada región. Estas frecuencias se encuentran principalmente dentro de la banda ISM, que es conocida como una banda libre para su uso, ya que no posee restricciones ni costos asociados para su utilización.

En América Latina, y específicamente en Ecuador, se utiliza la banda US902-928 MHz, la cual es la misma que se emplea en Estados Unidos. Esta banda, US902-928, está especialmente reservada para aplicaciones de IoT y se divide en un plan de canalización específico. El plan de canalización que se describe a continuación define cómo se organizan las frecuencias y los canales dentro de la banda US902-928 MHz. Esta subdivisión es para garantizar una comunicación eficiente y sin interferencias entre los dispositivos LoRa en esta región, esto según datos proporcionados por (LoRa Alliance, 2020).

- **Uplink:** hay 64 canales numerados del 0 al 63 que utilizan LoRa con un ancho de banda de 125 kHz y variaciones de DR0 a DR3. Estos canales inician en 902.3 MHz y aumentan linealmente en 200 kHz hasta alcanzar 914.9 MHz.
- **Uplink:** se incluyen 8 canales numerados del 64 al 71 que utilizan LoRa con un ancho de banda de 500 kHz en DR4 o LR-FHSS con un ancho de banda de 1.523 MHz en DR5-DR6. Estos canales inician en 903.0 MHz y aumentan linealmente en 1.6 MHz hasta llegar a 914.2 MHz.

- **Downlink:** se encuentran 8 canales numerados del 0 al 7 que utilizan LoRa con un ancho de banda de 500 kHz en DR8 a DR13. Estos canales inician en 923.3 MHz y aumentan linealmente en 600 kHz hasta 927.5 MHz (LoRa Alliance, 2020)

2.2.6.3. Modulación LoRa

La modulación LoRa, desarrollada por Semtech, se basa en Chirp Spread Spectrum (CSS), una técnica que proporciona señales robustas ante interferencias y ruidos. Esta modulación utiliza tres parámetros clave que afectan tanto a la robustez como a la velocidad de bits: el Spreading Factor (SF), el Ancho de Banda (BW) y la Tasa de Codificación (CR) (Ortiz, 2020).

- **El Spreading Factor (SF):** determina la cantidad de datos repetidos que se envían durante la transmisión. Un SF más alto resulta en un mayor alcance, pero a costa de una velocidad de transmisión más baja.
- **El Ancho de Banda (BW):** se refiere al rango de frecuencias utilizado en la transmisión. Un mayor ancho de banda permite velocidades de datos más altas, pero también disminuye la sensibilidad de la red debido al aumento del ruido.
- **La Tasa de Codificación (CR):** representa la relación de corrección de errores hacia adelante, proporcionando protección contra interferencias. Una tasa de codificación más alta ofrece una mayor protección, pero a expensas de una menor velocidad de datos.

Estos parámetros son necesarios para ajustar el rendimiento de una red LoRa según las necesidades específicas de la aplicación, ya que permiten equilibrar la velocidad de datos, el alcance y la robustez contra interferencias y ruidos.

2.2.7. Fase de almacenamiento, Gestión y Visualización

Una plataforma integral de datos IoT es una solución diseñada para gestionar eficientemente los datos generados por dispositivos del Internet de las Cosas (IoT),

recopilando, procesando y almacenando información de manera óptima para análisis y uso futuro. Además de almacenar datos, estas plataformas ofrecen capacidades avanzadas de análisis y visualización, permitiendo a los usuarios obtener información significativa y tomar decisiones informadas. Actúan como el núcleo central para administrar todo el flujo de información dentro de un ecosistema IoT, garantizando su disponibilidad, seguridad y utilidad para aplicaciones y usuarios finales.

Algunas de las plataformas comerciales que cumplen con los requisitos de visualización, gestión y almacenamiento son las siguientes:

- **ThingSpeak:** Una plataforma de análisis de IoT que simplifica la agregación, visualización y análisis de flujos de datos. Es idónea para proyectos que requieren una monitorización continua de datos.
- **Ubidots:** Permite una visualización intuitiva de la información proveniente de sensores conectados a dispositivos IoT en la red. Ofrece una versión experimental gratuita por 30 días.
- **Emoncms.org:** Una plataforma de código abierto diseñada para monitorizar y visualizar datos relacionados con energía y agua. Es especialmente útil en proyectos de eficiencia energética.
- **ThingWorx:** Proporciona herramientas para el desarrollo, implementación y gestión de aplicaciones IoT. Ofrece funciones avanzadas de visualización y notificación.
- **Thingier.io:** Plataforma de código abierto que facilita la conexión de dispositivos IoT y la creación de paneles personalizados para la visualización de datos en tiempo real(ESGEEKS, 2023).

En lo que respecta al proyecto se plantea el uso de ThingSpeak para el envío de datos desde los sensores de medición del nivel de agua que se procesen en el

microcontrolador Raspberry Pi se hará uso del protocolo HTTP o MQTT, seguidamente se proporcionará canales donde se pueda enviar los datos de los sensores, se puede configurar campos específicos en estos canales para representar los datos del nivel de agua potable. De igual forma esta plataforma ofrece la posibilidad de almacenar datos históricos que será de gran importancia si se requiere acceder a ellos a futuro.

Una vez que los datos estén siendo enviados y almacenados en ThingSpeak, se puede utilizar las herramientas de visualización integradas en la plataforma para crear gráficos, tablas y visualizaciones personalizadas de los datos de nivel de agua potable.

ThingSpeak permite crear paneles de control personalizados donde se puede organizar y visualizar los datos de manera intuitiva.

2.2.8 Fase de Notificación

Para enviar correos electrónicos desde una Raspberry Pi, hay varias herramientas y servicios útiles aparte de Mailgun, cada una con características particulares que las hacen adecuadas para diferentes escenarios:

- **SendGrid:** Proporciona una API robusta ideal para aplicaciones que requieren el envío de grandes volúmenes de correos electrónicos, como notificaciones automáticas.
- **SMTP2GO:** Este servicio ofrece un servidor SMTP fácil de integrar con dispositivos como la Raspberry Pi, facilitando el envío de correos con un enfoque en la entrega y la monitorización.
- **Amazon SES (Simple Email Service):** Ofrecido por AWS, este servicio es escalable y eficiente para aplicaciones embebidas en la nube, permitiendo un manejo extensivo de correos electrónicos a bajo costo.

En este contexto, **Mailgun** se destaca y se elige por su accesibilidad y facilidad de uso, especialmente para desarrolladores que trabajan con Python y Raspberry Pi. Mailgun no solo permite una integración sencilla gracias a su bien documentada API, sino que también ofrece un plan gratuito que permite el envío de hasta 100 correos diarios, lo que es suficiente para muchos proyectos de pequeña y mediana escala. Esto lo hace ideal para proyectos que requieren funcionalidad de correo electrónico sin un alto costo operativo, convirtiéndolo en la herramienta preferida para enviar correos electrónicos desde dispositivos como la Raspberry Pi.

2.3. Metodología del diseño del Proyecto

En el desarrollo de proyectos de IoT (Internet de las cosas), la elección de la metodología adecuada es necesaria para garantizar el éxito del proyecto. Las metodologías de desarrollo de software ofrecen enfoques estructurados para planificar, ejecutar y controlar el proceso de desarrollo.

En la **Tabla 3** se expone una comparativa de las metodologías más utilizadas para proyectos de IoT:

Tabla 3

Metodologías más comunes en proyectos IoT

Metodología	Descripción	Ventajas	Desventajas
Cascada	Enfoque secuencial con fases lineales: análisis, diseño, implementación, pruebas y mantenimiento.	Estructura clara y definida. Adecuado para proyectos con requisitos estables y claros.	Poca flexibilidad para cambios de requisitos. Riesgo de errores en etapas posteriores si los requisitos no están bien definidos al principio.

Agile	Enfoque iterativo e incremental, enfocado en la colaboración entre equipos multifuncionales.	Flexibilidad para adaptarse a cambios. Mayor involucramiento del cliente durante el desarrollo.	Requiere una comunicación y colaboración sólida entre equipos. No es ideal para proyectos con requisitos muy definidos desde el inicio.
Scrum	Marco de trabajo ágil que se enfoca en iteraciones cortas (sprints) y roles definidos.	Mayor transparencia y visibilidad del progreso. Adaptabilidad a cambios en los requisitos.	Requiere un equipo bien entrenado en el marco de trabajo Scrum. No adecuado para proyectos donde los requisitos son altamente

Nota. Adaptado de (VanZandt, 2023)

2.3.1. Metodología en Cascada

La metodología en cascada es una opción sólida para proyectos de IoT con requisitos estables y predecibles. Su enfoque secuencial permite una planificación precisa y ejecución eficiente, especialmente cuando los requisitos del sistema están claramente definidos desde el principio. Como lo muestra la **Figura 4**, esta metodología ofrece una estructura clara que facilita la gestión del proyecto y permite un seguimiento detallado del progreso en cada etapa para la coordinación efectiva entre hardware y software en proyectos de IoT. Organizada en fases como análisis, diseño, implementación, verificación y mantenimiento, proporciona una guía clara para el desarrollo de sistemas IoT. (Ionos, 2019).

Figura 4

Fases de la Metodología en Cascada



Nota. Tomado de (Ionos, 2019).

CAPÍTULO III: DISEÑO DEL SISTEMA

En este proyecto, se emplea la metodología en cascada para guiar el proceso de diseño del sistema inalámbrico de monitoreo y notificación del nivel de agua potable. La metodología en cascada, también conocida como modelo de desarrollo secuencial, se caracteriza por dividir el proceso de desarrollo en fases lineales y secuenciales, donde cada fase depende de la finalización exitosa de la fase anterior. En este contexto, la Metodología de Diseño sigue una secuencia lógica y predefinida, comenzando con la fase de "Requerimientos" y progresando hacia la fase de "Diseño".

En la fase 1, "Requerimientos", se lleva a cabo un análisis exhaustivo de los requisitos del sistema, incluyendo la identificación de los Stakeholders, la definición de los requerimientos funcionales y no funcionales, y la selección de hardware y software mediante benchmarking. Estos requisitos son documentados de manera detallada y sirven como base para la siguiente fase del diseño. En la fase 2, "Diseño", se utiliza la información recopilada en la fase de requerimientos para desarrollar la arquitectura del sistema y elaborar un diagrama de bloques que represente la estructura y la interacción de los componentes del sistema. Además, se detalla el proceso de armado del sistema, incluyendo la configuración y programación de los dispositivos hardware y software necesarios para su funcionamiento. Al seguir la metodología en cascada, se busca asegurar una planificación rigurosa y una secuencia ordenada de actividades, lo que contribuye a minimizar los riesgos y a garantizar la coherencia y la calidad del diseño final del sistema.

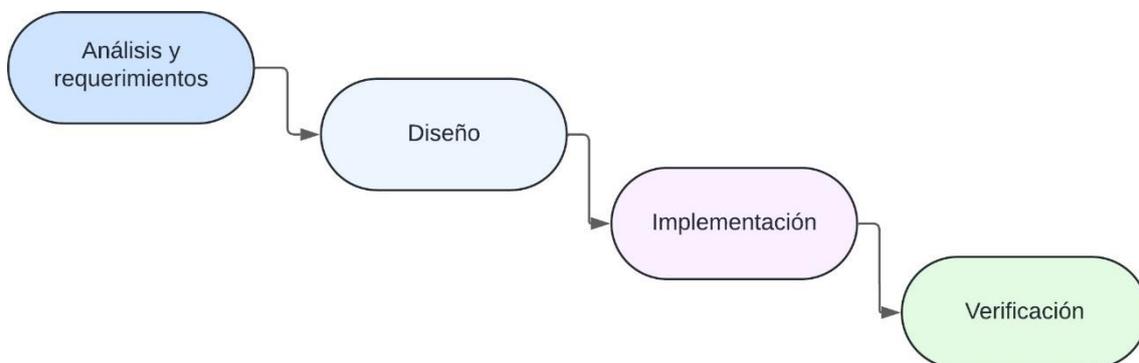
3.1 Metodología de Diseño

Una metodología de desarrollo de proyectos es un enfoque estructurado utilizado para gestionar y controlar el desarrollo de un proyecto. Proporciona un marco organizado de procesos, prácticas y directrices que guían desde la planificación inicial hasta la finalización del proyecto. Estas metodologías son fundamentales para asegurar la eficiencia, minimizar los riesgos y garantizar que se cumplan los objetivos del proyecto en términos de calidad, tiempo y alcance.

El proyecto mencionado seguirá una metodología de desarrollo basada en el modelo en cascada, que se extiende a lo largo de los cuatro capítulos del documento. Este modelo, elegido por su enfoque secuencial, proporciona una planificación estructurada y una implementación por medio de fases definidas. Una metodología sirve como un marco de orientación y guía para las actividades del proyecto, organizando el proceso mediante la definición de pautas a seguir y restricciones a cumplir. Las cuatro fases del modelo en cascada se detallan en la **Figura 5** y serán seguidas rigurosamente para garantizar el cumplimiento de los objetivos planteados.

Figura 5

Fases del modelo en cascada del sistema de Medición de Nivel de Agua



Nota. Adaptado de “El modelo en cascada: desarrollo secuencial de software” por (Ionos, 2019).

3.2 Fase 1: Requerimientos

Una de las primeras fases de la metodología en cascada que se usa en este proyecto es la de los requerimientos, aquí se comprenden las necesidades y especificaciones funcionales y no funcionales que deben ser cumplidas para satisfacer los objetivos del sistema. Estos incluyen aspectos como el escenario actual del abastecimiento de agua potable en la zona de San Gabriel, provincia del Carchi, el propósito del sistema, los Stakeholders involucrados, los atributos específicos del sistema, tales como la precisión y frecuencia de los datos, así como la selección de hardware y software adecuados para el proyecto. Los requerimientos también abordan aspectos relacionados con la arquitectura del sistema, la configuración de los dispositivos y la integración con la plataforma de gestión de datos.

3.2.1 Generalidades

Las generalidades en este proyecto se refieren a la comprensión del contexto y las condiciones actuales del abastecimiento de agua potable en la zona de San Gabriel, provincia del Carchi. Esto incluye la descripción detallada del entorno en el que se implementará el sistema de monitoreo, considerando factores como la infraestructura existente, los problemas y desafíos actuales en el suministro de agua potable, así como las necesidades y expectativas de los usuarios y Stakeholders involucrados en el proyecto. Las generalidades proporcionan un marco de referencia para la identificación de los requerimientos y la planificación del diseño del sistema.

3.2.1.1 Escenario Actual

El proyecto se desarrolla en la ciudad de San Gabriel, provincia del Carchi, donde se aborda el sistema de abastecimiento de agua potable. Se identifican problemas como la reducción del nivel de agua debido a condiciones climáticas extremas, como lluvias intensas o temporadas de sequía. Estos eventos pueden afectar la captación de agua,

especialmente en la Quebrada del Oso en el sector del Chamizo, zona de enfoque del proyecto.

La importancia del proyecto radica en garantizar un suministro estable de agua potable a la comunidad, considerando que el 60.91% del agua proviene de la red pública y el 36.89% de fuentes naturales como ríos y canales. A pesar de una alta cobertura de acceso al servicio de agua potable (98%), existen desafíos relacionados con el caudal de agua disponible y la demanda de los usuarios. De igual forma según información proporcionada por (GAD Montufar, 2015) Al examinar la infraestructura a nivel de la cabecera cantonal, se observa que la cobertura del acceso al servicio de agua potable es del 98%, lo que resulta en un déficit mínimo del 2%. En cuanto al suministro de agua en la ciudad de San Gabriel, este se realiza mediante dos puntos de captación que abastecen a las tres redes de distribución. En particular, la captación que atiende a la red media y baja se encuentra en la Quebrada del Oso, en el sector del Chamizo. Esta fuente de agua provee a 2.875 usuarios y tiene un caudal nominal de $Q=30-55$ l/s, aunque está diseñada para soportar un caudal máximo de $Q=77$ l/s.

Las zonas de distribución incluyen áreas como San Antonio, San Vicente, San José y parte del barrio 27 de septiembre en la red media, y sectores como Santa Rosa, San Andrés y Jardín del Norte en la red baja. La calidad del servicio de agua potable es controlada mediante un sistema de monitoreo que incluye análisis físicos, químicos y bacteriológicos mensuales, así como la realización de sondeos diarios de cloro residual en toda la ciudad.

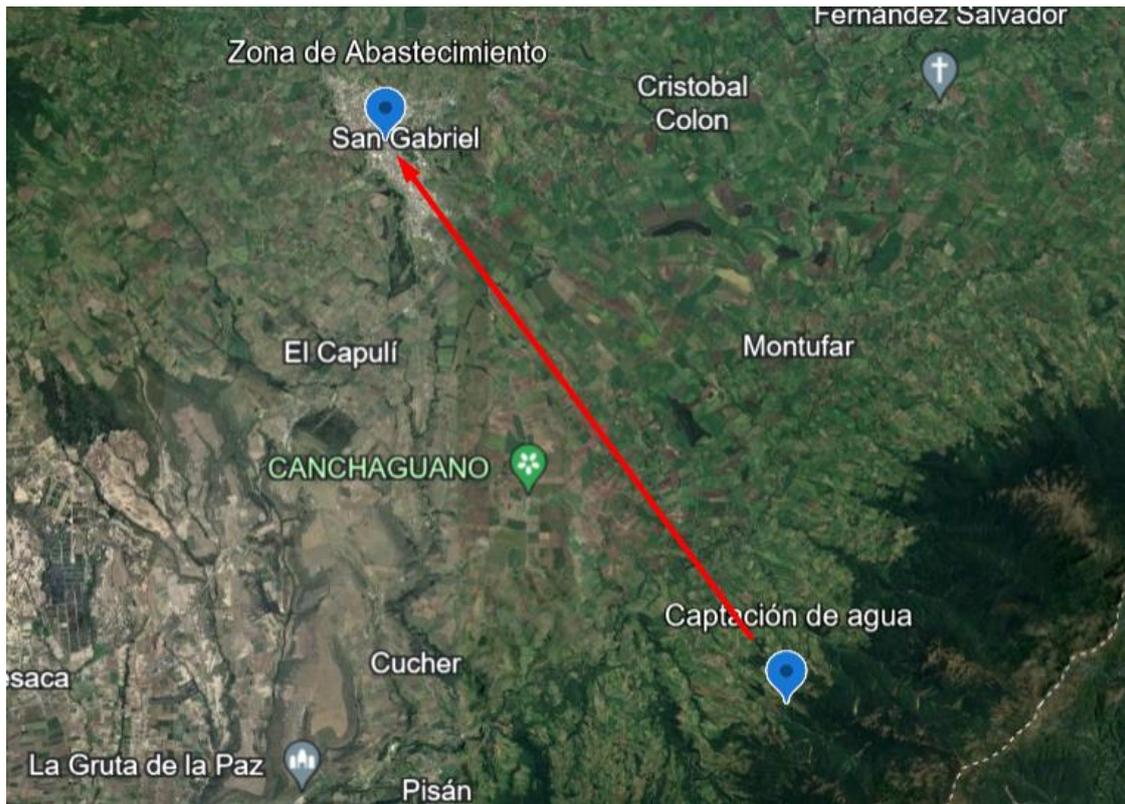
3.1.1.2 Ubicación del sistema

La **Figura 6** muestra la ubicación geográfica de la zona de captación y la zona de abastecimiento de agua potable en el proyecto de monitoreo y notificación del nivel

de agua. Se destaca que existe una separación de aproximadamente 10 kilómetros entre ambas ubicaciones. La captación se encuentra aproximadamente en las coordenadas $0^{\circ}30'50''\text{N}$ y $77^{\circ}46'28''\text{W}$."

Figura 6

Ubicación de Captación de agua y Zona de Abastecimiento



Fuente. Autoría

El sistema de monitoreo estará equipado con nodos de recolección de datos estratégicamente colocados en las piscinas de contención de agua (desarenador), situadas antes del desarenador. Actualmente, se utiliza un indicador de nivel visual, consistente en un letrero, para observar el estado del nivel del agua como se puede apreciar en la **Figura 7**.

Figura 7

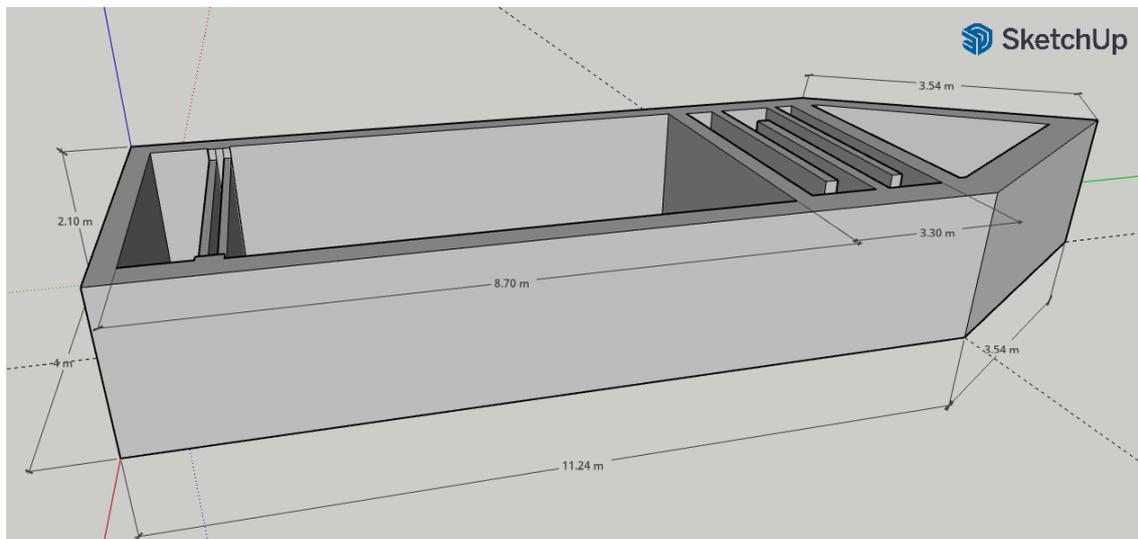
Piscinas de contención del agua de captación (desarenador)



Fuente. Autoría

3.2.2 Medidas del sistema

La **Figura 8** muestra el diseño de un desarenador de agua potable. Las dimensiones globales del desarenador son 13 metros de largo, 4 metros de ancho y 2.10 metros de alto. La estructura es de tipo rectangular y presenta varios compartimientos internos para la separación y depuración del agua. En la parte superior se encuentra una estructura de forma triangular de medidas de 4m por cada lado que es la parte principal del sistema de desarenador, donde llega toda el agua captada de la Quebrada del Oso.

Figura 8*Dimensiones del desarenador en cm*

En la **Tabla 4** se muestra la información estructurada en forma de tabla sobre el desarenador, incluyendo los detalles sobre los niveles de agua en diferentes condiciones climáticas:

Tabla 4*Dimensiones y condiciones del desarenador*

Descripción	Medida
Largo total del desarenador	13 m
Ancho total del desarenador	4 m
Altura total del desarenador	2.10 m
Altura normal del agua en invierno	1.95 m
Altura normal del agua en verano	1.83 m
Condiciones	Comentario
Niveles alertantes de bajo nivel	<180 cm
Niveles altos de rebote	No problemático, debido a orificios de desagüe

Esta tabla ofrece una visión clara y estructurada de las características operativas del desarenador bajo diferentes condiciones climáticas, facilitando su comprensión y seguimiento para mantenimiento y operaciones.

3.2.3 Propósito del sistema

El propósito del sistema de monitoreo y notificación del nivel de agua potable se fundamenta en la necesidad crítica de garantizar un acceso constante y seguro al recurso hídrico, en consonancia con los principios del Plan Nacional del Buen Vivir y el Plan de Desarrollo y Ordenamiento Territorial del Cantón Montufar. En este contexto, el proyecto se centra en mejorar la gestión integral y sustentable del agua potable, especialmente en las áreas de captación y desarenador, con el fin de beneficiar a los 2.835 usuarios abastecidos por la Quebrada el Oso. Las pequeñas mejoras planteadas en el sistema representan un avance significativo hacia una mayor eficiencia y funcionalidad, permitiendo un control y seguimiento más efectivo del suministro de agua potable.

La implementación de un sistema inalámbrico de monitoreo ofrecerá numerosos beneficios, incluida una mayor precisión en la medición de los niveles de agua, la optimización de los recursos y el tiempo dedicado a la verificación manual, y la capacidad de detectar y responder rápidamente a situaciones anormales mediante alertas automáticas. Esto no solo mejorará la calidad del servicio para los usuarios finales, sino que también aumentará la transparencia y la participación ciudadana al proporcionar acceso a la información en tiempo real a diversas partes interesadas. En última instancia, el propósito del sistema es contribuir a la mejora continua del suministro de agua potable, promoviendo así la salud, el bienestar y el desarrollo sostenible de la comunidad.

3.2.3 Beneficiarios o Stakeholders

En este proyecto, los beneficiarios Stakeholders son las personas, grupos o entidades que se verán afectados por el diseño, implementación y resultados del sistema de monitoreo y notificación del nivel de agua potable. Se incluye a:

- **Usuarios finales:** La ciudad de San Gabriel en la provincia del Carchi, que se beneficia directamente del suministro de agua potable y cuya calidad de vida y bienestar están estrechamente ligados a la disponibilidad y fiabilidad del agua, esta está representada por la Dirección de Agua Potable.
- **Autoridades locales:** Como el Gobierno Municipal y otras entidades responsables de la gestión del agua, que tienen un interés en asegurar un suministro adecuado y sostenible de agua potable para la población.
- **Operadores del sistema:** El personal encargado de operar y mantener el sistema de abastecimiento de agua, quienes se beneficiarán de la mejora en la eficiencia y la facilidad de monitoreo proporcionada por el nuevo sistema.

Básicamente los beneficiarios Stakeholders son todas aquellas partes interesadas que tienen un interés directo o indirecto en el funcionamiento y los resultados del proyecto, y cuyas necesidades y expectativas deben ser consideradas durante su desarrollo (Ayda et al., 2021), la **Tabla 5** muestra detalladamente la lista de Stakeholders y el rol que desempeñan.

Tabla 5

Lista de Stakeholders de Sistema de medición de Nivel de Agua

Lista de Stakeholders	Rol
Ing. Sigifredo Ruales Garcés	Director de agua potable y alcantarillado GADM-M
Roberto Cárdenas	Operador de la captación de agua potable
MSc. Jaime Michilena	Tutor de Trabajo de Titulación
MSc. Luis Suárez	Asesor de Trabajo de Titulación
Lizeth Cárdenas	Desarrollador de Trabajo de Titulación

Fuente. Autoría

3.2.3.1 Conectividad del operador

El sistema utiliza una conexión de internet constante, disponible en la ubicación del desarenador, para enviar actualizaciones y alertas críticas directamente al correo electrónico del responsable. La elección del correo electrónico como método principal de comunicación se debe a las siguientes razones:

- **Conectividad Constante:** El operador cuenta con acceso permanente a internet, lo que facilita la recepción inmediata de las notificaciones sin depender de la señal de telefonía móvil.
- **Independencia de la Señal Móvil:** En la región donde se ubica el desarenador, la falta de internet es generalmente síntoma de ausencia total de señal de telefonía. Por lo tanto, optar por notificaciones por mensaje de texto (SMS) no sería efectivo, ya que, en situaciones de falta de señal, tanto el internet como los SMS serían inaccesibles.
- **Eficiencia y Alcance:** El correo electrónico permite enviar información detallada y extensa, incluyendo datos, gráficos y alertas sin las limitaciones de longitud de un SMS. Además, permite que múltiples destinatarios reciban simultáneamente las notificaciones, asegurando que toda la información relevante llegue a todos los interesados de manera eficaz y oportuna.

3.2.4 Construcción de atributos de los requerimientos

Abordar la definición de las características de los atributos de los requisitos, basándose en los Stakeholders, el sistema y la arquitectura, es esencial para el éxito del proyecto. Al comprender las necesidades de las partes interesadas, la naturaleza del sistema y su arquitectura, los requisitos pueden ser definidos de manera coherente y realista. Esta alineación garantiza que los requisitos sean factibles de implementar y

satisfagan las expectativas de los involucrados, lo que contribuye a una ejecución exitosa del proyecto.

3.2.4.1 Nomenclatura de los requerimientos

La **Tabla 6** presenta la nomenclatura de los requerimientos utilizada en el proyecto, identificando las abreviaturas correspondientes a cada tipo de requerimiento. Estas abreviaturas se utilizarán para clasificar y organizar los requisitos según su origen y propósito dentro del proceso de desarrollo. Esta estructura facilitará la comprensión y gestión de los requerimientos por parte de los Stakeholders y los equipos de desarrollo, asegurando una implementación efectiva y satisfactoria del proyecto.

Tabla 6

Definición de Acrónimos de Nomenclatura del sistema

Abreviatura	Requerimientos
RDS	Requerimientos de Stakeholders
RDA	Requerimientos de Arquitectura
RDST	Requerimientos de sistema

Fuente. Autoría

3.2.5 Requerimientos de Stakeholders

Los requisitos por cumplir se definieron en una reunión con los principales interesados en el proyecto, incluyendo a la persona encargada del diseño y al operador de la zona de captación de agua. Durante este encuentro, realizado el viernes 26 de abril de 2024 a las 3 p.m., se abordaron aspectos clave como el monitoreo del nivel de agua

potable, los dispositivos de notificación necesarios y otros requisitos técnicos y funcionales del sistema inalámbrico de monitoreo.

El operador de la zona aportó su experiencia y conocimientos sobre los desafíos del suministro de agua en San Gabriel, provincia del Carchi. El objetivo fue establecer los requisitos en cada fase del sistema, permitiendo un análisis detallado para seleccionar el mejor hardware y software, garantizando así un desempeño óptimo. Para más detalles, se puede consultar el anexo, donde se documenta la reunión y los temas tratados.

El éxito del proyecto de diseño del sistema inalámbrico de monitoreo y notificación del nivel de agua potable en San Gabriel, provincia del Carchi, depende en gran medida de la identificación y satisfacción de los requerimientos de los Stakeholders involucrados en su implementación y uso. En la **Tabla 7** se presentan los Requerimientos de Stakeholders (RDS), organizados en dos categorías principales: Requerimientos de Funciones y Requerimientos de Usuarios. Estos requerimientos reflejan las necesidades y expectativas de los diferentes actores del proyecto, desde la optimización de la distribución de dispositivos hasta la accesibilidad de los datos de monitoreo y la automatización del proceso de notificación, priorizando su importancia en función de su impacto en la eficacia y utilidad del sistema.

Tabla 7

Requerimientos de Stakeholders

Requerimientos de Stakeholders RDS		Prioridad		
	Requerimientos de funciones	Alta	Media	Baja
RDS 1	Estrategias de alimentación eléctrica para el sistema	X		
RDS 2	Distribución para que el Gateway y los Nodos Sensores funcionen de manera más eficiente	X		

RDS 3	Condiciones de acceso a Internet necesarias para el funcionamiento de los dispositivos.	X
RDS 4	Asegurar que los dispositivos del sistema operen de manera ininterrumpida.	X
RDS 5	Captura y transferencia ininterrumpida de datos.	X

Requerimientos de Usuarios

RDS 6	Utilización de recursos de hardware y software accesibles en costos.	X
RDS 7	Habilitación de acceso a los datos de monitoreo tanto de manera local como remota.	X
RDS 8	Presentación actualizada de datos de nivel de agua	X
RDS 9	Notificaciones ante variaciones relevantes en el nivel de agua.	X
RDS 10	Disponibilidad ágil y optimizada de los datos de monitoreo.	X
RDS 11	Automatización del proceso de monitoreo y notificación	X
RDS 12	Notificación automática a través de un correo.	X

Fuente. Autoría

3.2.6 Requerimientos del Sistema

El diseño del presente sistema de monitoreo del nivel de agua potable implica considerar diversos factores, como la precisión del monitoreo, la autonomía energética, la capacidad de comunicación inalámbrica y la facilidad de integración con sistemas de gestión existentes. Además, se debe garantizar la fiabilidad y la robustez del sistema para asegurar su funcionamiento continuo en condiciones adversas.

A continuación, se presenta **Tabla 8** donde se describe los requerimientos del sistema, detallando sus características y funcionalidades clave para satisfacer las necesidades de monitoreo y notificación en las redes de distribución de agua potable en San Gabriel, provincia del Carchi.

Tabla 8*Requerimientos del Sistema*

Requerimientos del Sistema RDST		Prioridad		
Requerimientos de Interfaces		Alta	Media	Baja
RDST 1	Interfaz para visualizar datos y generar alertas en tiempo real.	X		
RDST 2	Interfaz optimizada para la transmisión de datos entre nodos, Gateway y dispositivos finales.	X		
RDST 3	Interfaz de alimentación eléctrica que garantice la operatividad continua y el funcionamiento de los dispositivos.	X		
RDST 4	Asegurar la disponibilidad y accesibilidad de los puertos de conexión entre nodos sensores y módulos LPWAN, evitando bloqueos o interferencias en la comunicación.	X		
Requerimientos de Uso				
RDST 5	Implementación del sistema con un enfoque seguro y estratégicamente planificado.	X		
RDST 6	Obtención de datos precisos sobre el nivel de agua mediante sensores.	X		
RDST 7	Someter a pruebas previas del software cargado en los dispositivos (nodos sensores y Gateway) antes de la implementación.	X		
RDST 8	Comunicación permanente y operación ininterrumpida entre la estación Gateway y los nodos.	X		
Requerimientos de Estados				
RDST 9	Conexión permanente entre dispositivo transmisor (TX) y receptor (RX).	X		
RDST 10	Funcionamiento continuo de los dispositivos del sistema.	X		
RDST 11	Operatividad constante de la plataforma de gestión y emisión de alarma.	X		
Requerimientos Físicos				

RDST 12	Alineación óptima y ausencia de interferencias en la comunicación entre nodos sensores y el Gateway.	X
RDST 13	Ejecución de pruebas operativas y depuración de errores previas a la implementación.	X
RDST 14	Carcasa impermeable ajustada a las condiciones del entorno.	X
RDST 15	Supervisión y control para prevenir daños o pérdidas de los nodos sensores.	X
RDST 16	Autorización de uso de instalaciones para la instalación de los nodos sensores.	X

Fuente. Autoría

3.2.7 Requerimientos de Arquitectura

En la **Tabla 9** se cuenta con los requerimientos de arquitectura para el proyecto, el cual, demanda una cuidadosa planificación arquitectónica que garantice su eficiencia, confiabilidad y adaptabilidad a las condiciones específicas de la región. En la siguiente tabla se presentan los Requerimientos de Arquitectura (RDA), clasificados en diferentes categorías como Hardware, Eléctricos, Software, y Lógicos y de Diseño. Cada requerimiento se evalúa en términos de prioridad, desde alta hasta baja, considerando su importancia para el funcionamiento integral del sistema y su impacto en la eficacia del monitoreo del suministro de agua potable.

Tabla 9

Requerimientos de Arquitectura

Requerimientos de Arquitectura RDA		Prioridad		
Requerimientos de Hardware		Alta	Media	Baja
RDA 1	Accesibilidad económica del Sistema.	X		
RDA 2	Sensores y dispositivos diseñados para operar en exteriores bajo diversas condiciones ambientales.	X		

RDA 3	Bajo requerimiento de energía eléctrica.	X
RDA 4	Alta eficiencia en el procesamiento de datos.	X
RDA 5	Suministro eléctrico y sistema de respaldo con baterías para los nodos y el Gateway.	X
RDA 6	Estructura minimizada de los dispositivos del sistema.	X
RDA 7	Disposición de dispositivos operando en la misma banda de frecuencia de MHz o GHz.	X
Requerimientos Eléctricos		
RDA 8	Suministro eléctrico con respaldo de baterías para garantizar funcionamiento en caso de cortes de energía o agotamiento de las baterías de los nodos.	X
Requerimientos de Software		
RDA 9	Configuración de un servidor web en la nube y una plataforma de gestión para la notificación.	X
RDA 10	Instalación simplificada del sistema Operativo en las placas.	X
RDA 11	Garantizar las bibliotecas y herramientas del lenguaje de programación de Python	X
RDA 12	Procesamiento de gran cantidad de datos en tiempo real.	X
RDA 13	Sistema para visualizar y gestionar los datos recopilados.	X
Requerimientos Lógicos y de Diseño		
RDA 14	Integración óptima entre hardware y software para un rendimiento eficiente del sistema.	X
RDA 15	Compatibilidad del Gateway con tecnologías de comunicación inalámbrica.	X
RDA 16	Costo reducido para la implementación del sistema.	X
RDA 17	Determinación de la tecnología LPWAN más adecuada según las condiciones y demandas del área geográfica.	X

Fuente. Autoría

3.2.8 Benchmarking (Selección de Hardware y Software)

El benchmarking, especialmente en la etapa de selección de hardware y software, es de gran importancia para el diseño del sistema inalámbrico de monitoreo y notificación del nivel de agua potable en San Gabriel, provincia del Carchi. Esta práctica permite comparar el desempeño, características y costos de diferentes opciones disponibles en el mercado, lo que facilita la toma de decisiones informadas y la elección de las soluciones más adecuadas para las necesidades específicas del proyecto.

En el contexto de selección de hardware, el benchmarking permite evaluar la capacidad de procesamiento, consumo energético, tamaño y durabilidad de los dispositivos, asegurando la elección de aquellos que mejor se adapten a las condiciones geográficas y operativas de la región. Asimismo, en cuanto a la selección de software, el benchmarking proporciona información importante sobre la compatibilidad, eficiencia y funcionalidades de las diferentes plataformas y herramientas disponibles, garantizando la elección de aquellas que permitan una gestión eficaz de los datos de monitoreo y una presentación clara y accesible de la información para los usuarios finales (Sergio & Romo Picazo, 2000).

3.2.8.1 Hardware

Elegir los dispositivos de hardware adecuados es clave para el diseño del sistema inalámbrico de monitoreo y notificación del nivel de agua potable. Estos dispositivos conforman la infraestructura física del sistema y su elección influye directamente en su rendimiento, confiabilidad y capacidad para cumplir con los objetivos del proyecto.

Es fundamental que los dispositivos seleccionados puedan medir, procesar y transmitir los datos con precisión y eficiencia, además de ser duraderos y compatibles con el entorno y las condiciones climáticas de la región. Una elección bien pensada no solo mejora el desempeño del sistema, sino que también reduce costos de mantenimiento y garantiza su funcionamiento óptimo a largo plazo.

- **Selección de Hardware del nodo de monitoreo**

En lo que respecta a la elección de Hardware para el nodo de monitoreo se cuenta con dos elementos que son el microcontrolador y el sensor, ya que el nodo de monitoreo actúa como el corazón del sistema, garantizando la captura precisa de datos y la transmisión eficiente de información para una gestión efectiva del recurso hídrico, se debe seleccionar adecuadamente los elementos a utilizar para un eficiente rendimiento del sistema.

a) Microcontrolador

El microcontrolador determina la capacidad de procesamiento de datos, la eficiencia energética y la capacidad de integración de diferentes componentes del nodo (Alvear & Freire, 2001). Una elección acertada del microcontrolador garantiza que el nodo pueda manejar de manera eficiente las tareas de medición, procesamiento y transmisión de datos en tiempo real. Además, un microcontrolador adecuado facilita la implementación de algoritmos de control y gestión que optimizan el funcionamiento del sistema y mejoran su capacidad de respuesta ante cambios en los niveles de agua.

Para lograr una buena selección del microcontrolador a usar, se hará uso de los Requerimientos de Stakeholders (**3.2.5 Requerimientos de Stakeholders**), Requerimientos del Sistema (**3.2.6 Requerimientos del Sistema**) y Requerimientos de Arquitectura (**3.2.7 Requerimientos de Arquitectura**), tratados en los puntos anteriores, los

microcontroladores a considerarse serán: Arduino Nano, Arduino Uno, Arduino Mega y el ESP32.

Tabla 10

Elección del microcontrolador del Nodo

Hardware	Requerimientos							Valoración Total
	RDS4	RDST3	RDST4	RDA1	RDA4	RDA6	RDA15	
Arduino Nano	0	1	1	1	1	1	0	5
Arduino Uno	1	1	1	1	1	0	0	5
Arduino Mega	1	1	1	0	1	0	0	4
ESP32-38 pines	1	1	1	1	1	1	1	7

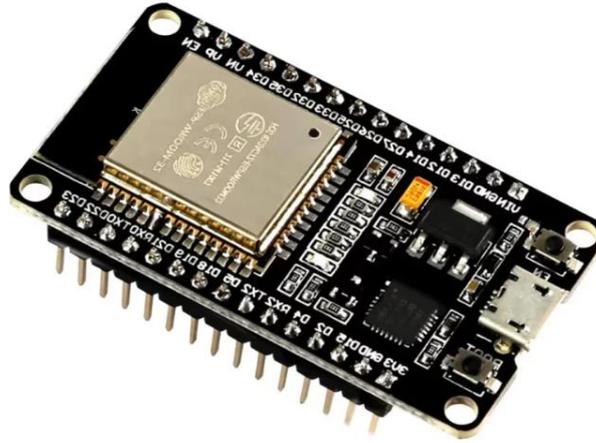
1 – Cumple
0 – No cumple
Elección: Microcontrolador ESP32-38 pines

Fuente. Autoría

En la **Tabla 10** de evaluación, el ESP32 obtuvo una puntuación total de 7, lo que indica que cumple con la mayoría de los requisitos especificados. Este microcontrolador satisface las necesidades identificadas en los Requerimientos de Diseño de Stakeholders (RDST), como la estrategia de alimentación eléctrica (RDST3) y la garantía del funcionamiento continuo de los dispositivos del sistema (RDST4). Además, cumple con los Requerimientos de Arquitectura (RDA) relacionados con la accesibilidad económica del sistema (RDA1), la capacidad de procesamiento rápido de datos (RDA4) y el tamaño reducido de los dispositivos (RDA6). En sí, la elección del microcontrolador ESP32-38 pines que se visualiza en la **Figura 9** se justifica por su capacidad para cumplir con los requisitos del proyecto y proporcionar una base sólida para el desarrollo y la implementación exitosa del sistema de monitoreo y notificación del nivel de agua potable.

Figura 9

Microcontrolador ESP32 de 38 Pines



Nota. Tomado de APM Electrónica, por (APM ELECTRONICS, 2024)

Algunas características con las que cuenta son las siguientes:

- Microcontrolador: Dual-core Xtensa 32-bit LX6, hasta 240 MHz.
- Memoria RAM: 520 KB.
- Memoria Flash: 4 MB.
- Conectividad: Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BLE.
- GPIO: 38 pines disponibles.
- Interfaces: SPI, I2C, UART, ADC, DAC, PWM, etc.
- Consumo de energía: Depende de la configuración y las tareas realizadas.

b) Sensor

El sensor de nivel de agua es el componente clave que permite medir con exactitud los niveles de agua en las captaciones de abastecimiento. Una elección acertada del sensor garantiza una detección precisa y constante de los cambios en el nivel de agua, lo que ayuda para la toma de decisiones informadas en la gestión del suministro de agua potable. Además, un sensor de nivel de agua adecuado proporciona una respuesta rápida y estable

ante variaciones en las condiciones ambientales, asegurando una supervisión continua y fiable del recurso hídrico.

Tabla 11

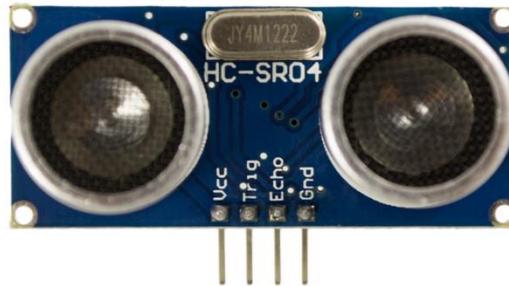
Selección de Sensor de Medición de Nivel

Hardware	Requerimientos						Valoración Total
	RDS4	RDS6	RDA1	RDA2	RDA14	RDST6	
Sensor Ultrasónico MB7076	1	0	0	1	0	1	3
Sensor Ultrasónico HC-SR04	1	1	1	0	1	1	5
Sensor Efecto Hall EFS-04P	0	0	1	0	0	0	1
Sensor Ultrasónico MB1240	1	1	0	1	1	1	5

1 – Cumple
0 – No cumple
Elección: Sensor Ultrasónico HC-SR04

Fuente: Autoría

Basándose en la evaluación detallada de los sensores y sus calificaciones en relación con los requisitos de hardware establecidos en la **Tabla 11** indican que el Sensor Ultrasónico HC-SR04 es la mejor opción, este se puede evidenciar en la **Figura 10**. Este sensor cumple con los requisitos más importantes, logrando una valoración total de 5 puntos. En comparación con los otros sensores evaluados, el HC-SR04 cumple con el funcionamiento continuo del sistema (RDS4), la accesibilidad en costo (RDS6), la accesibilidad económica (RDA1), la compatibilidad de hardware y software (RDA14), y la precisión en las mediciones (RDST6). Aunque no cumple con la resistencia a la intemperie (RDA2), su alto desempeño en los demás criterios lo hace el más adecuado para el uso en el nodo secundario.

Figura 10*Sensor Ultrasónico HC-SR04*

Nota. Tomado de Sensor de distancia ultrasónico, por (Amazon, 2024d)

Algunas de las características que lo definen son las siguientes:

Claro, aquí tienes una lista de las características técnicas del Sensor Ultrasónico HC-SR04:

- Voltaje de Operación: 5V DC
- Corriente de Operación: 15 mA
- Frecuencia de Operación: 40 kHz
- Rango de Medición: 2 cm a 400 cm
- Precisión: ± 3 mm
- Ángulo de Medición: 15 grados
- Tiempo de Ciclo de Medición: 50 ms
- Señal de Trigger: Pulso de 10 μ s de alto nivel
- Señal de Echo: Señal de pulso proporcional a la distancia medida
- Dimensiones:
 - Longitud: 45 mm
 - Ancho: 20 mm
 - Altura: 15 mm

- Temperatura de Operación: -15°C a 70°C
- Interfaz de Comunicación: Pines VCC, GND, Trigger, Echo

Estas características técnicas hacen del HC-SR04 una opción eficiente y precisa para aplicaciones de medición de distancia y nivel de agua.

- **Selección de la tecnología Inalámbrica de Baja Potencia (LPWAN)**

La selección de la tecnología inalámbrica de baja potencia (LPWAN) es un paso crítico en el diseño de sistemas de monitoreo y notificación, especialmente en entornos donde la eficiencia energética y la cobertura de larga distancia son fundamentales. LPWAN ofrece una gama de opciones que incluyen LoRaWAN, Sigfox y NB-IoT, entre otras, cada una con sus propias características y beneficios. La elección adecuada de la tecnología LPWAN depende de diversos factores, como el alcance requerido, la duración de la batería, la capacidad de penetración en interiores y exteriores, así como los costos de implementación y operativos. En la **Tabla 12** se tratará los aspectos clave para seleccionar la tecnología LPWAN más adecuada para un sistema de monitoreo y notificación, proporcionando una base sólida para tomar decisiones informadas y maximizar la eficacia del sistema en su conjunto.

Tabla 12

Selección de tecnología LPWAN

LPWAN	Requerimientos						Valoración Total
	RDST9	RDA3	RDA6	RDA7	RDA16	RDA17	
LoRa	0	1	1	1	1	1	5
ZigBee	1	1	1	0	0	0	3
Sigfox	0	1	1	1	0	0	3
NB-IoT	1	1	1	0	0	0	3

1 – Cumple

0 – No cumple
Elección: LoRa

Fuente: Autoría

c) Selección del Módulo LoRa

Tras la elección de la tecnología LPWAN LoRa como la más adecuada, se realiza un minucioso examen de los diferentes módulos LoRa disponibles en el mercado. Este proceso busca identificar el módulo que mejor se ajuste a las exigencias del proyecto de diseño de monitoreo y notificación del nivel de agua potable. Para ello, se ha recabado información sobre módulos que operan en la banda de frecuencia específica de la región, como se había mostrado en la **Tabla 2**, donde se establece un intervalo de 433 y 915 MHz, algunos de los módulos LoRa que cumplen con este rango de frecuencias son los que se puede evidenciar en la **Tabla 13** que se expone a continuación.

Tabla 13

Selección del Módulo de Tecnología LoRa

Módulo LoRa	Requerimientos					Valoración Total
	RDST3	RDST9	RDA3	RDA6	RDA16	
PID 3072 RFM95W	1	0	0	0	1	2
RYLR998	1	1	1	1	1	5
RYLR896	1	0	1	1	1	4

1 – Cumple

0 – No cumple

Elección: RYLR998

Fuente: Autoría

El módulo RYLR998 que se evidencia en la **Figura 11** ha sido seleccionado para el diseño del sistema, debido a su destacada calificación en diversas características clave, como bajo consumo eléctrico y costo de implementación reducido, además de su capacidad para operar dentro del rango de frecuencias de 433 y 915 MHz necesarias para

la región. Además, el hecho de que el transmisor y el receptor puedan estar conectados continuamente añade un valor significativo al sistema, asegurando una comunicación constante y fiable entre los dispositivos en todas las condiciones operativas. Esta elección garantiza un rendimiento óptimo del sistema en términos de eficiencia energética, costos y fiabilidad de la comunicación, lo que es necesario para el monitoreo eficaz del suministro de agua potable en la región especificada.

Figura 11

Módulo de Lora RYLR998 del interfaz 868/915 MHz



Nota. Tomado de Sensor de distancia ultrasónico, por (Amazon, 2024c)

Algunas de las características con las que cuenta son:

- Frecuencia: 433 MHz - 915 MHz.
- Alcance: Hasta 10 km en línea de visión.
- Potencia de transmisión: Hasta 100 mW.
- Modulación: LoRa (spread spectrum).
- Interfaces: UART, SPI.
- Consumo de energía: Variable según la configuración de potencia de transmisión.

- **Selección de Batería o fuente de Alimentación del Sistema**

La selección de una batería o fuente de alimentación es fundamental para garantizar el funcionamiento óptimo y la durabilidad del sistema. Considerar factores como la capacidad, la eficiencia y la compatibilidad con los componentes es esencial para mantener el rendimiento constante del sistema.

a) Estación de monitoreo

La selección de la fuente de alimentación en un sistema electrónico influye en el consumo eficiente de energía. En una estación nodo, utilizada en sistemas de monitoreo o control, la elección de la batería o fuente de energía afecta directamente su autonomía y eficiencia.

La siguiente tabla muestra el consumo de corriente en modo activo y en reposo de los principales componentes de una estación nodo, como el microcontrolador ESP32, el módulo LoRa RYLR998 y el sensor ultrasónico HC-SR04. Con estos datos, se puede analizar y determinar la opción de alimentación más adecuada para optimizar el rendimiento y prolongar la vida útil del sistema.

Tabla 14

Consumo eléctrico en estación nodo

Elemento	Consumo Estado Sleep (mA)	Consumo estado Activo (mA)
ESP32-38 Pines	0.05	60
RYLR998	0.05	10
Sensor Ultrasónico HC-SR04	0.01	5
Total	0.11 mA	75 mA

Fuente: Autoría

De acuerdo con la **Tabla 14**, Se determina que la corriente requerida para el funcionamiento de las estaciones de monitoreo es de 75 mA, lo que corresponde al consumo en estado activo de la placa del microcontrolador ESP32 y los sensores conectados a las estaciones. Para estimar el consumo de la batería, se aplica la siguiente ecuación (1).

$$\text{Consumo Total} = \frac{(T_{cn} * I_{cn}) + (T_{cd} * I_{cd})}{(T_{cn} * T_{cd})} \quad (1)$$

En donde:

T_{cn}: tiempo de consumo en estado activo

T_{cd}: tiempo de consumo en estado de reposo

I_{cn}: Consumo de energía en estado activo

I_{cd}: Consumo de energía en estado de reposo

Con base en los parámetros definidos para el tiempo de operación y el modo de reposo de las estaciones, se determinan los siguientes valores:

T_{cn} como 1 minuto (equivalente a 60 segundos)

I_{cn} de 75 mA

T_{cd} como 5 minutos (equivalente a 300 segundos)

I_{cd} de 0.11 mA

Sustituyendo estos valores en la ecuación (1),

$$\text{Consumo Total} = \frac{(60 * 75) + (300 * 0.11)}{(75 * 0.11)} \quad (2)$$

$$\text{Consumo Total} = 61.39. \text{mA}$$

Para los cálculos realizados se ha optado por elegir un cargador solar de 20000 mAh con puertos USB de 5 V y resistente a exteriores como se muestra en la **Figura 12**.

Figura 12

Cargador Solar para los nodos de recolección de datos del sistema de medición de nivel de agua.



Nota. Tomado de Power-Bank-Solar-Charger por (Amazon, 2024a)

Las características y funciones principales del dispositivo son las siguientes:

- **Batería:** 20000 mAh, esta es la capacidad total de la batería. Indica cuánta energía puede almacenar.
- **Carga Solar:** 5V/200 mAh, esto significa que el panel solar puede generar una corriente de 200 mA a un voltaje de 5V bajo condiciones óptimas de luz solar.
- **Linterna LED:** 4W, la potencia de la linterna LED es de 4 vatios. Esto indica cuánta energía consume cuando está encendida.

- **Entrada/Salida:** DC 5V/1A, la batería puede ser cargada y puede cargar otros dispositivos a una corriente de 1A y un voltaje de 5V.

Tiempo de Funcionamiento para el consumo total calculado en la ecuación (2)

$$\textit{Tiempo de vida útil} = \frac{\textit{Capacidad de bateía}}{\textit{Consumo Total}} \quad (3)$$

$$\textit{Tiempo de vida útil} = \frac{20000 \textit{mAh}}{61.39 \textit{mA}} \quad (4)$$

$$\textit{Tiempo de vida útil} = 325.82 \textit{ horas} \quad (5)$$

Este resultado que se ha obtenido en el cálculo de la ecuación 3 es el tiempo de consumo sin que el dispositivo se cargue ya sea de forma tradicional o mediante el panel solar.

Cuando se utiliza el panel solar para recargar la batería, la duración puede extenderse dependiendo de cuánta energía se genere y se almacene en la batería. Bajo condiciones óptimas de luz solar, el panel solar genera:

$$\textit{Energía generada por hora} = 5V \times 0.2A = 1W \quad (7)$$

Si la batería se está descargando a una tasa de 61.39 mA, se está consumiendo energía a una tasa de:

$$\textit{Consumo de energía por hora} = 5V \times 0.06139A = 0.30695W \quad (8)$$

La energía neta ganada por hora con carga solar sería:

$$\textit{Energía neta por hora} = 1W - 0.30695W = 0.69305W \quad (9)$$

- **Tiempo de Carga Solar Completa**

Para cargar completamente la batería de 20000 mAh utilizando solo el panel solar (sin consumo), el tiempo sería:

$$Tiempo\ de\ carga\ completa = \frac{20000\ mAh \times 5V}{1W} = \frac{100000\ mWh}{1W} = 100\ horas \quad (10)$$

- **Carga Tradicional**

El dispositivo también puede cargarse de forma tradicional utilizando una entrada de 5V/1A. El tiempo para cargar completamente la batería de esta manera sería:

$$Tiempo\ de\ carga\ completa = \frac{20000\ mAh}{1000\ mA} = 20\ horas \quad (11)$$

b) Estación Gateway

En el contexto de la estación Gateway del proyecto, que actúa como punto central en el sistema de comunicación inalámbrica, se espera que esté conectada directamente a la red eléctrica, eliminando así la necesidad de utilizar baterías para su alimentación. Dado que la estación Gateway está diseñada para gestionar un mayor volumen de datos y realizar funciones más complejas que una estación nodo, el consumo de energía, especialmente del microcontrolador, será significativamente mayor. Por lo tanto, la opción de utilizar baterías sería menos práctica debido al consumo más rápido de energía, haciendo que la alimentación eléctrica directa sea la elección más adecuada para garantizar un funcionamiento continuo y confiable del sistema.

- **Selección de Hardware para la estación Gateway**

La selección del hardware para la estación Gateway es un paso crítico en el diseño del sistema, ya que determinará la capacidad de procesamiento, la conectividad y la fiabilidad de la estación central. Se debe considerar cuidadosamente la elección de componentes como el microcontrolador, módulos de comunicación y sensores

adicionales para asegurar un funcionamiento eficiente y una comunicación estable con los nodos de monitoreo.

a) Microcontrolador Gateway

La **Tabla 15** muestra la comparativa de microcontroladores, evaluando sus capacidades en función de los requerimientos del sistema definidos previamente. Los microcontroladores considerados son la Raspberry Pi 4, ESP32 y Arduino MKR WAN 1300, los cuales se analizarán en cuanto a su conectividad a internet, funcionamiento continuo, compatibilidad con el lenguaje de programación Python, facilidad de instalación de sistemas operativos y eficiencia en la alimentación eléctrica. Cada microcontrolador será calificado con 1 si cumple con el criterio y con 0 si no lo cumple, lo que permitirá determinar la opción más adecuada para el diseño del sistema inalámbrico de monitoreo y notificación del nivel de agua.

Tabla 15

Elección de microcontrolador Gateway

Hardware	Requerimientos					Valoración Total
	RDS3	RDS4	RDST3	RDA10	RDA11	
Raspberry Pi 4	1	1	1	1	1	5
ESP32	1	1	1	0	1	4
Arduino MKR WAN 1300	0	1	1	1	0	3

1 – Cumple

0 – No cumple

Elección: Raspberry Pi 4

Fuente: Autoría

Con base en la evaluación realizada se ha tomado a la Raspberry Pi 4 como la mejor opción para este proyecto, este se evidencia en la **Figura 13**. La Raspberry Pi 4 obtuvo la valoración total más alta debido a su capacidad de cumplir con todos los

requerimientos críticos del sistema: conectividad a internet, funcionamiento continuo, compatibilidad con el lenguaje de programación Python, facilidad de instalación de sistemas operativos y eficiencia en la alimentación eléctrica. Estos atributos la hacen particularmente adecuada para la implementación del sistema a desarrollar.

Figura 13

Microcontrolador Raspberry Pi 4



Nota. Tomado de Sensor de distancia ultrasónico, por (Amazon, 2024a).

Las características que presenta este microcontrolador son:

- CPU: Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.
- Memoria RAM: 2GB, 4GB o 8GB LPDDR4-3200 SDRAM (dependiendo del modelo), en este caso se hará uso de una RAM de 2GB.
- Conectividad: Wi-Fi 802.11ac, Bluetooth 5.0, Gigabit Ethernet.
- Puertos: 2x USB 3.0, 2x USB 2.0, HDMI, 2x micro HDMI, GPIO, CSI camera port, DSI display port, etc.
- Alimentación: USB-C (5V/3A).

- Sistema Operativo: Compatible con varios sistemas operativos, incluyendo Raspbian (basado en Debian).

3.2.8.2 Software

El software determina la capacidad de procesamiento, la eficiencia en la gestión de datos y la funcionalidad del sistema en su conjunto. Por lo tanto, es importante elegir plataformas, protocolos y herramientas que sean compatibles, confiables y adecuadas para las necesidades específicas del proyecto. Una selección acertada de software garantiza la integración efectiva de los dispositivos, la seguridad de la información, y facilita la presentación y análisis de datos para los usuarios finales. Además, un software bien seleccionado permite la escalabilidad del sistema, asegurando su adaptabilidad a futuras necesidades y tecnologías emergentes.

- **Estación de monitoreo**

Se debe considerar la selección de software que ofrezca funcionalidades robustas para la gestión de datos, la configuración de alertas y la generación de informes.

a) Software de Gestión

En la **Tabla 16** se evalúan tres lenguajes de programación utilizados para flashear y programar los microcontroladores ESP32 en las estaciones de monitoreo o nodos del sistema: C/C++ (Arduino IDE), MicroPython y ESP-IDF. La evaluación se basa en cinco características clave: interfaz de visualización amigable con el usuario, instalación segura y estratégica del sistema operativo en placa, buenas pruebas de funcionamiento, instalación de forma sencilla y conocimiento del lenguaje. Cada lenguaje se califica con 1 si cumple con la característica y con 0 si no lo hace, proporcionando una visión clara de cuál es la opción más adecuada para el desarrollo de este proyecto.

Tabla 16*Selección de software de estación de monitoreo*

Software	Requerimientos					Valoración Total
	RDST1	RDST5	RDST13	RDA10	RDA11	
C/C++ (Arduino IDE)	1	1	1	1	1	5
MicroPython	1	1	1	1	1	5
ESP-IDF	0	1	1	0	0	2

1 – Cumple
0 – No cumple
Elección: C/C++ (Arduino IDE)

Fuente: Autoría

Después de una cuidadosa evaluación de los lenguajes de programación disponibles para el desarrollo de las estaciones de monitoreo o nodos del sistema, se ha tomado la decisión de seleccionar C/C++ utilizando el entorno de desarrollo Arduino IDE como la opción preferida. Esta elección se basa en su capacidad para cumplir con la mayoría de las características requeridas para el proyecto, incluyendo una interfaz de visualización amigable con el usuario, una instalación segura y estratégica del sistema operativo en la placa, buenas pruebas de funcionamiento, una instalación sencilla además de tener conocimientos previos sobre el lenguaje.

- **Estación Gateway**

La elección del software para el microcontrolador de la estación gateway juega un papel fundamental en el éxito y la eficiencia del proyecto de monitoreo y notificación del nivel de agua potable. El software seleccionado no solo debe ser capaz de recibir y procesar los datos provenientes de los nodos de medición de manera confiable y precisa, sino que también debe ofrecer herramientas adecuadas para la visualización de estos datos de manera clara y comprensible. Un software bien elegido garantiza una gestión eficiente

de los recursos hídricos al permitir una monitorización en tiempo real, detección temprana de cambios bruscos en los niveles de agua y una respuesta rápida ante cualquier eventualidad. Además, una interfaz de visualización intuitiva y amigable facilita la interpretación de los datos por parte de los usuarios finales, permitiendo una toma de decisiones informada y oportuna.

a) Software de recepción y procesamiento de datos

Para la estación Gateway, se ha decidido utilizar el sistema operativo Raspbian, basado en Debian, debido a su amplia compatibilidad y estabilidad en entornos de desarrollo y proyectos de IoT. Raspbian ha sido optimizado específicamente para funcionar en las placas Raspberry Pi, lo que garantiza un rendimiento óptimo y una configuración sencilla para diversas aplicaciones. Este sistema operativo ofrece una amplia gama de herramientas y recursos que facilitan el desarrollo de proyectos, incluyendo una gran cantidad de paquetes preinstalados y una interfaz de usuario amigable.

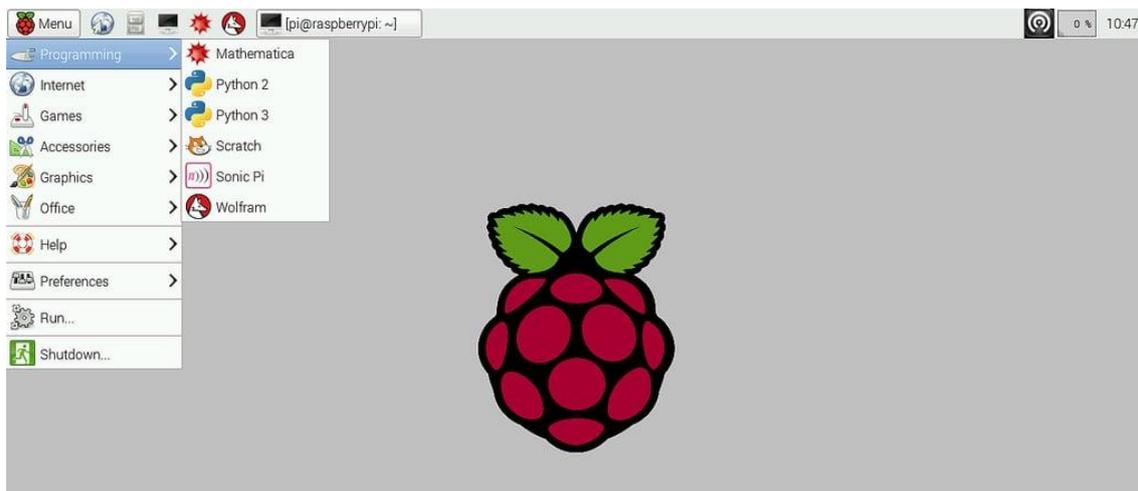
Python se ha elegido el lenguaje de programación principal para el desarrollo del software de la estación Gateway debido a su simplicidad, versatilidad y amplio soporte en la comunidad. Python es conocido por su sintaxis limpia y legible, lo que facilita la creación de código claro y conciso. Además, cuenta con una gran cantidad de bibliotecas y módulos especializados para el procesamiento de datos, la comunicación con dispositivos externos y la integración con servicios en la nube.

La combinación de Raspbian como sistema operativo y Python como lenguaje de programación ofrece una plataforma robusta y flexible para el desarrollo de la estación Gateway. Permite una fácil integración con otros dispositivos y servicios, así como una rápida implementación de nuevas funcionalidades y actualizaciones. Además, la comunidad activa de Raspberry Pi (**Figura 14**) y Python proporciona un amplio soporte

y recursos adicionales que pueden ser aprovechados para optimizar y expandir el sistema según sea necesario.

Figura 14

Raspberry Pi - Sistema operativo Raspbian y lenguaje Python



Nota. Tomado de Sensor de distancia ultrasónico, por (Mitchell, 2018).

b) Software de visualización de datos

En el siguiente análisis se presenta la **Tabla 17** que muestra la comparativa de software de visualización, evaluando tres plataformas populares: Ubidots, ThingSpeak y Adafruit IO. Cada plataforma se ha evaluado en función de cinco requisitos clave, designados como RDS6 a RDS10. La valoración se basa en si cada plataforma cumple (1) o no cumple (0) con cada uno de estos requisitos. Al final de la tabla, se encuentra la valoración total para cada plataforma, reflejando la cantidad de requisitos que cumplen. Tras un análisis exhaustivo, se ha tomado una decisión y se presenta la elección del software de visualización que mejor se adapta a los requerimientos del proyecto.

Tabla 17

Selección de software de visualización de datos

Requerimientos

Software Visualización	RDS6	RDS7	RDS8	RDS9	RDS10	Valoración Total
Ubidots	1	1	1	1	1	5
ThingSpeak	1	1	1	1	1	5
Adafruit IO	1	1	1	1	1	5

1 – Cumple
0 – No cumple
Elección: ThingSpeak

Fuente: Autoría

Después de un análisis detallado de las tres plataformas de visualización evaluadas: Ubidots, ThingSpeak y Adafruit IO, se ha llegado a la conclusión de que ThingSpeak es la opción más adecuada para este proyecto. Aunque todas las plataformas evaluadas cumplen con los mismos requisitos y reciben la misma valoración total, ThingSpeak fue seleccionada debido a su familiaridad y popularidad en la comunidad de IoT, así como a su integración fluida con otros servicios y dispositivos. Además, ThingSpeak ofrece una interfaz intuitiva y fácil de usar, así como una amplia gama de herramientas para la visualización de datos en tiempo real y la configuración de alertas. A pesar de que todas las opciones evaluadas son sólidas y pueden cumplir con los requisitos del proyecto, ThingSpeak se destaca como la elección preferida en este caso particular debido a su robustez, versatilidad y facilidad de uso.

c) Software de Notificación de datos

En el siguiente análisis se presenta la *Tabla 18* que muestra la comparativa de software de visualización, evaluando tres plataformas populares: SendGrid, SMTP2GO y Mailgun. Cada plataforma se ha evaluado en función de cinco requisitos clave, designados como RDA1, RDA9 a RDA12. La valoración se basa en si cada plataforma cumple (1) o no cumple (0) con cada uno de estos requisitos. Al final de la tabla, se

encuentra la valoración total para cada plataforma, reflejando la cantidad de requisitos que cumplen.

Tabla 18

Selección de software de notificación de datos

Software Visualización	Requerimientos					Valoración Total
	RDA9	RDA10	RDA11	RDA12	RDA1	
SendGrid	1	0	1	1	1	4
SMTP2GO	1	0	1	1	1	4
Mailgun	1	1	1	1	1	4

1 – Cumple

0 – No cumple

Elección: Mailgun

Fuente: Autoría

La elección de Mailgun como herramienta preferida, según la tabla proporcionada, se justifica por su completo cumplimiento en todos los Requerimientos de Software evaluados: RDA9 (configuración de servidor web en la nube para notificaciones), RDA10 (instalación simplificada del sistema operativo en las placas), RDA11 (garantizar las bibliotecas y herramientas del lenguaje de programación de Python), RDA12 (procesamiento de gran cantidad de datos en tiempo real) y RDA1 (accesibilidad económica del sistema). Aunque SendGrid y SMTP2GO también obtuvieron puntuaciones altas, Mailgun se destaca al satisfacer específicamente el requisito RDA10, mostrando una ventaja en la integración y facilidad de uso en plataformas como Raspberry Pi.

3.3. Fase 2: Diseño

Esta segunda etapa correspondiente al modelo en cascada conocida como diseño dentro del diseño del sistema comprende la creación de un diagrama de arquitectura que

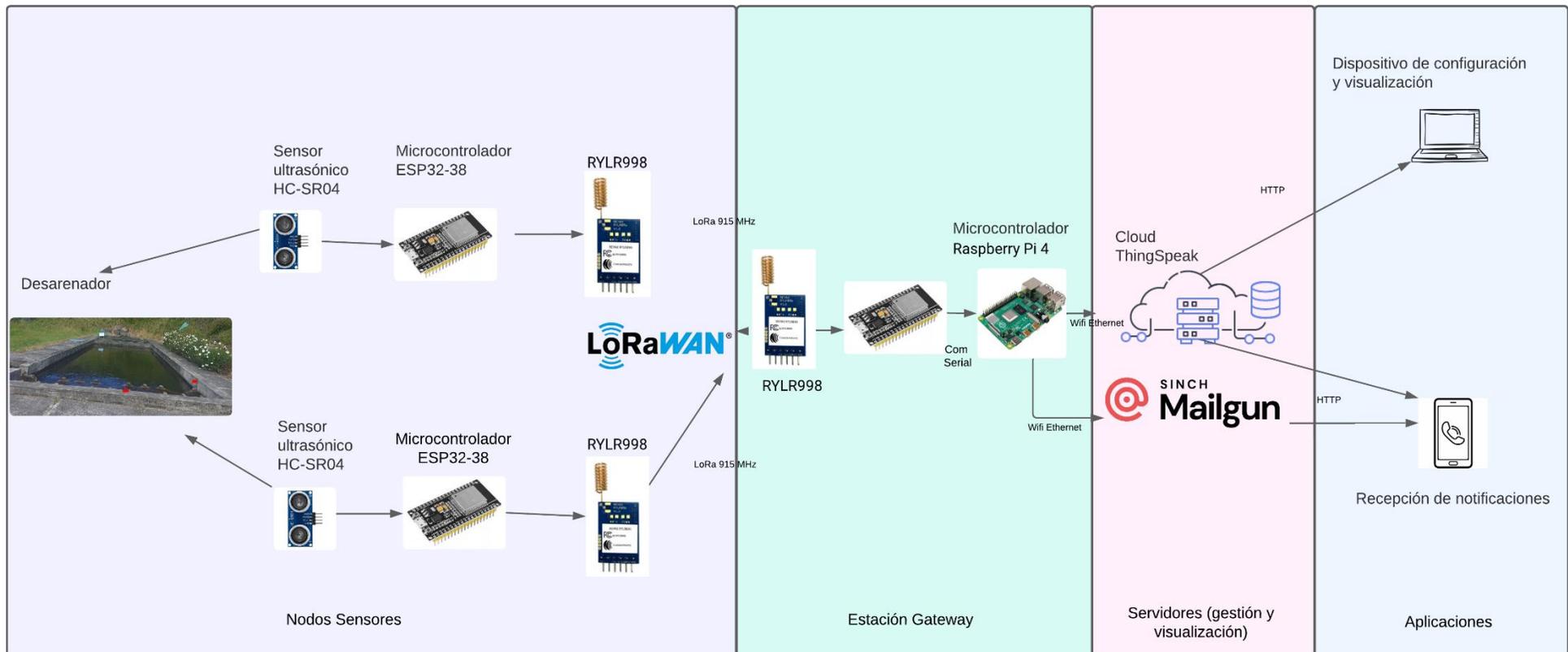
determinará la estructura general del sistema, incluyendo la interconexión de los diferentes componentes y la comunicación entre ellos. Además, se elaborará los respectivos diagramas de bloques que descompondrán el sistema en módulos más pequeños, facilitando así la comprensión de su funcionamiento y la identificación de posibles puntos de fallo. Una vez completados estos diagramas, se procederá al armado del sistema, donde se integrarán físicamente todos los componentes y se configurarán para trabajar de manera conjunta.

3.3.1 Arquitectura del sistema

El diseño de la arquitectura del proyecto se ha desarrollado con un enfoque estratégico para garantizar su funcionalidad y eficiencia. Se ha tomado en cuenta cada componente y su interacción dentro del sistema, logrando una integración coherente de los dispositivos previamente analizados. Como se muestra en la **Figura 15**, esta estructura permite un monitoreo y gestión óptimos del nivel de agua potable. Además, su solidez y escalabilidad aseguran un desempeño confiable, alineado con los objetivos del proyecto y preparado para una implementación efectiva.

Figura 15

Arquitectura del sistema de monitoreo y notificación del nivel de agua.



Fuente: Autoría

En el diseño del sistema, se ha planificado utilizar una combinación de hardware específico para cumplir con las necesidades de monitoreo y gestión del nivel de agua potable. Para los nodos de monitoreo, se emplearán sensores ultrasónicos HC-SR04, que permiten una medición precisa del nivel de agua. Estos sensores estarán conectados a microcontroladores ESP32 de 38 pines, los cuales ofrecen potencia de procesamiento y capacidad de conectividad adecuadas para la recopilación y procesamiento de datos en tiempo real. Además, se integrarán módulos LoRa RYLR998 en cada nodo para la transmisión de datos de forma inalámbrica a la estación base.

En cuanto a la estación base, se ha seleccionado el Raspberry Pi 4 debido a su versatilidad y capacidad de procesamiento. Este dispositivo actuará como el punto central de recepción de datos, donde se recibirán las señales de los nodos de monitoreo a través de los módulos LoRa. Para la comunicación con la plataforma de gestión y notificación, se configurará otro módulo LoRa RYLR998 en la estación base, permitiendo así el envío de datos a la plataforma de manera eficiente.

Para conectar el Raspberry Pi a la plataforma de ThingSpeak, se puede hacer uso de diversos protocolos que ofrece la plataforma, brindando flexibilidad en la forma en que se envían los datos. Estos protocolos incluyen HTTP, MQTT y WebSockets, entre otros. Por ejemplo, mediante HTTP, se puede enviar datos utilizando solicitudes GET o POST, lo que garantiza una comunicación confiable y simple. Por otro lado, MQTT ofrece una opción liviana y eficiente para la transmisión de datos en tiempo real, ideal para aplicaciones que requieren una comunicación rápida y de baja latencia. Además, WebSockets proporciona una conexión bidireccional que permite una comunicación en tiempo real y una actualización instantánea de los datos. Al aprovechar estos protocolos, se facilita la conexión del Raspberry Pi con ThingSpeak, permitiendo una comunicación eficiente y confiable para el monitoreo y la gestión de datos de manera efectiva.

3.3.2 Diagrama de Bloques

El diagrama de bloques presentado en la **Figura 16** ilustra la arquitectura de un sistema IoT (Internet de las Cosas) de 3 capas, que abarca desde la captura de datos hasta la interacción con el usuario final. La capa de percepción es la encargada de capturar los datos del entorno, y en este caso, se utiliza un sensor ultrasónico HC-SR04, el cual mide el nivel de agua en un entorno determinado. Este sensor es conectado a una placa ESP32, que actúa como el microcontrolador que recopila y envía los datos. Para la transmisión de estos datos de forma inalámbrica, se utiliza un módulo LoRa RYLR998, que permite la transmisión de datos a larga distancia a través de radiofrecuencia. Esta capa está destinada a la recolección de datos, actuando como el nodo final de captura.

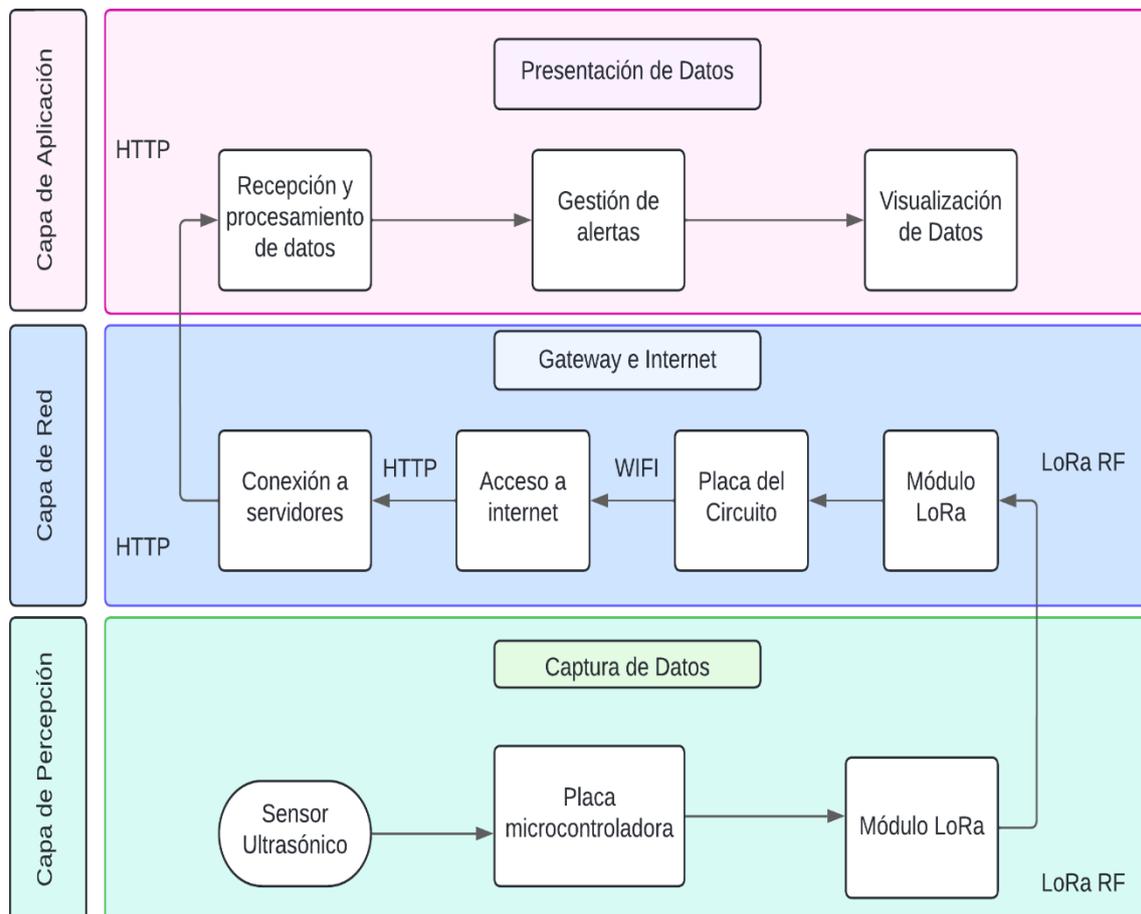
La capa de red es la que gestiona la transmisión y comunicación de los datos entre los nodos de percepción y el servidor que procesa la información. En este caso, se emplea otro ESP32 con un módulo LoRa, que recibe los datos del nodo de recolección. Estos datos son enviados a través de una interfaz serial a una placa Raspberry Pi 4, que actúa como el gateway. La Raspberry Pi procesa la información recibida y la sube a Internet utilizando el protocolo HTTP, conectándose a la plataforma ThingSpeak, que es utilizada para almacenar y visualizar los datos en tiempo real.

Finalmente, la capa de aplicación se ocupa de la interacción con el usuario y de la visualización de los datos. En este caso, los datos procesados en la Raspberry Pi son enviados a través de Internet a la plataforma de visualización, que es una página web alojada en los servidores de GoDaddy. Esta página permite visualizar los niveles de agua en tiempo real. Además, el sistema incluye una función de alerta mediante el servicio de Mailgun, que envía correos electrónicos automáticos en caso de que se superen ciertos umbrales de nivel de agua, alertando al usuario sobre posibles problemas.

El sistema está diseñado para ser completamente automatizado, y la ejecución del programa en la Raspberry Pi se inicia automáticamente al encenderse el dispositivo, sin necesidad de intervención manual. De esta forma, se garantiza que el sistema opere de manera continua y confiable, con el fin de proporcionar monitoreo en tiempo real del nivel de agua, así como alertas cuando sea necesario. Este enfoque de tres capas (percepción, red y aplicación) asegura una arquitectura escalable y eficiente para aplicaciones IoT.

Figura 16

Diagrama de bloques basado en la Arquitectura IoT de 3 capas



Fuente: Autoría

3.3.1.1. Diagrama de Flujo del Sistema

El diagrama de bloques de la **Figura 17** describe un sistema de medición de nivel de agua y alerta mediante notificaciones, que también permite la visualización de datos en una página web. El proceso comienza con la conexión de los módulos LoRa RYLR998 a la placa ESP32, seguida de la programación de componentes y la inicialización de dispositivos. Este paso asegura que todos los componentes funcionen correctamente y estén listos para la comunicación.

Una vez que los dispositivos están programados e inicializados, se establece una conexión entre el nodo sensor y el gateway. Si la conexión es exitosa, se procede a conectar el sensor de ultrasonido HC-SR04, que es responsable de medir el nivel de agua. El siguiente paso implica la programación del sensor y el envío de los datos recogidos a la Raspberry Pi 4 (RP4).

En la RP4, se verifica si se están recibiendo los datos del sensor. Si los datos se reciben correctamente, estos se procesan y se envían a la plataforma ThingSpeak para su almacenamiento y análisis. Esta plataforma permite establecer umbrales y enviar alertas en caso de que los niveles de agua superen ciertos límites predefinidos.

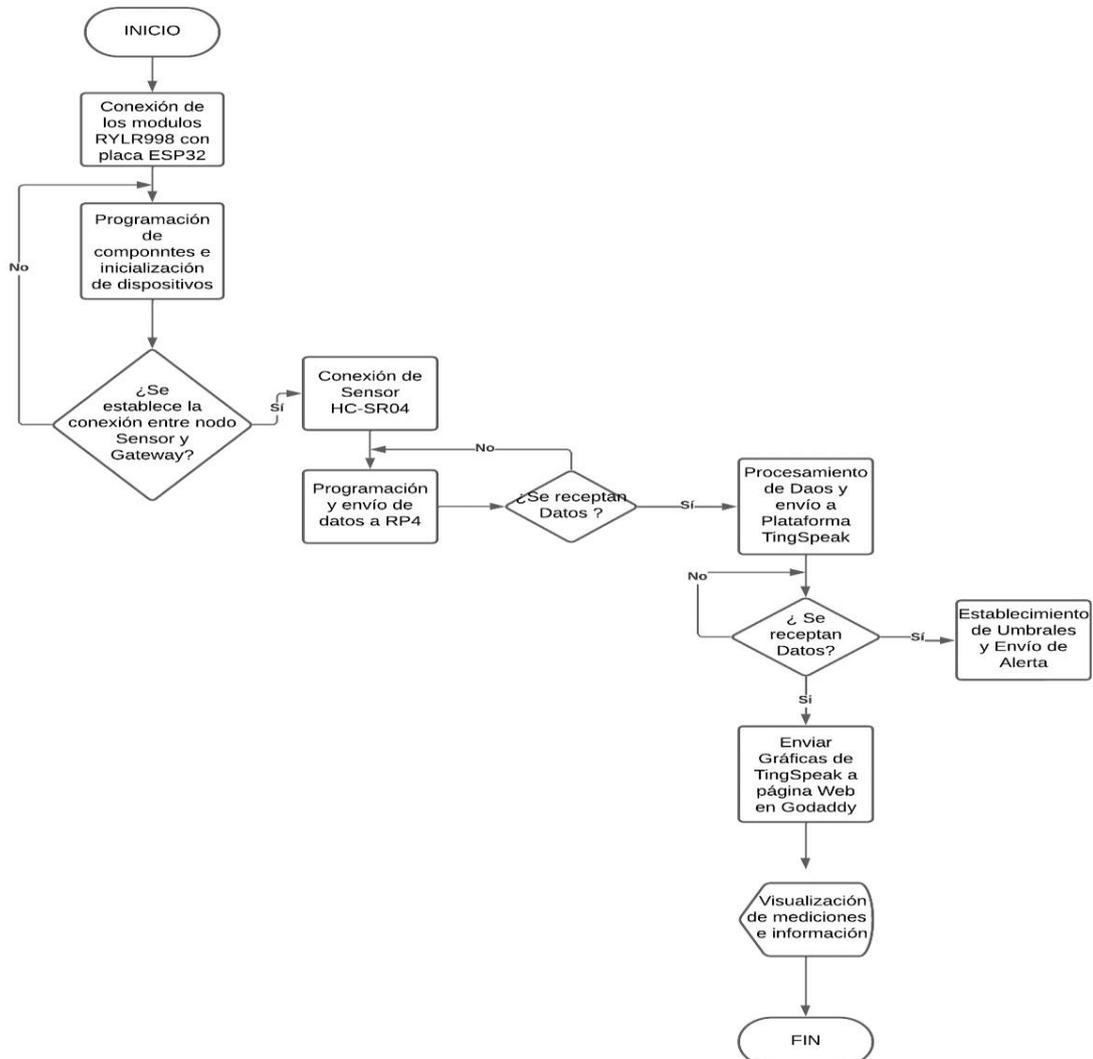
Una vez que los datos son procesados y almacenados, se envían gráficos de ThingSpeak a una página web alojada en GoDaddy. Esto permite la visualización de las mediciones y la información de manera accesible y en tiempo real, finalizando así el proceso.

El establecimiento de un diagrama de flujo como este es importante porque proporciona una visión clara y estructurada del proceso completo. Ayuda a identificar todos los pasos necesarios, las posibles decisiones y los puntos de verificación, asegurando que el sistema funcione de manera eficiente y efectiva. Además, facilita la

identificación de posibles problemas y mejoras en el proceso, contribuyendo al desarrollo de un sistema robusto y confiable.

Figura 17

Diagrama de Flujo del Sistema de Medición de Nivel de Agua en General



Fuente: Autoría

3.3.3 Diagramas de interconexión

Los diagramas de interconexión son fundamentales, ya que proporcionan una representación visual detallada de las conexiones del sistema, ayudan a reducir errores

durante la implementación, sirven como documentación importante para futuras referencias y mejoran la comunicación entre los miembros del equipo de desarrollo. Tras seleccionar el hardware, se presentan los diagramas de conexión para asegurar el funcionamiento del nodo sensor y el Gateway LoRa.

3.3.3.1. Nodo Sensor

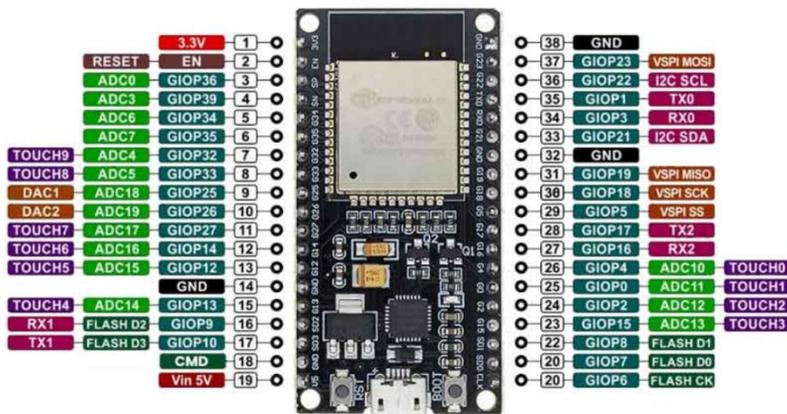
El nodo sensor de este proyecto involucra tres elementos principales: un microcontrolador ESP32 de 38 pines (**Figura 18**), un sensor ultrasónico HC-SR04 (**Figura 19**) y un módulo de transmisión LoRa para el envío de datos del sensor de forma inalámbrica al nodo gateway. En este primer diagrama de conexión, se explicará de forma detallada la conexión entre el ESP32 de 38 pines y el sensor ultrasónico HC-SR04, esto se puede evidenciar en la **Figura 20**. El ESP32 actuará como el cerebro del nodo sensor, manejando la recolección de datos del sensor ultrasónico y, posteriormente, enviando esta información a través del módulo LoRa en configuraciones avanzadas. Por ahora, se centrará en cómo conectar y configurar correctamente el ESP32 y el HC-SR04 para asegurar una lectura precisa de las distancias detectadas por el sensor ultrasónico.

Pasos generales para la conexión entre el HC-SR04 al ESP32 (38 Pines)

- a) Reunión componentes Necesarios
- b) Identificación de pines de cada componente

Figura 18

Identificación de Pines del ESP32 de 38 pines



Nota. Tomado de ESP32 38 Pines Bluetooth + WiFi, por (Tostratonic, 2024)

Figura 19

Pines del sensor HC-SR04



Nota. Tomado de Componentes electrónicos, por (Cuartero, 2017)

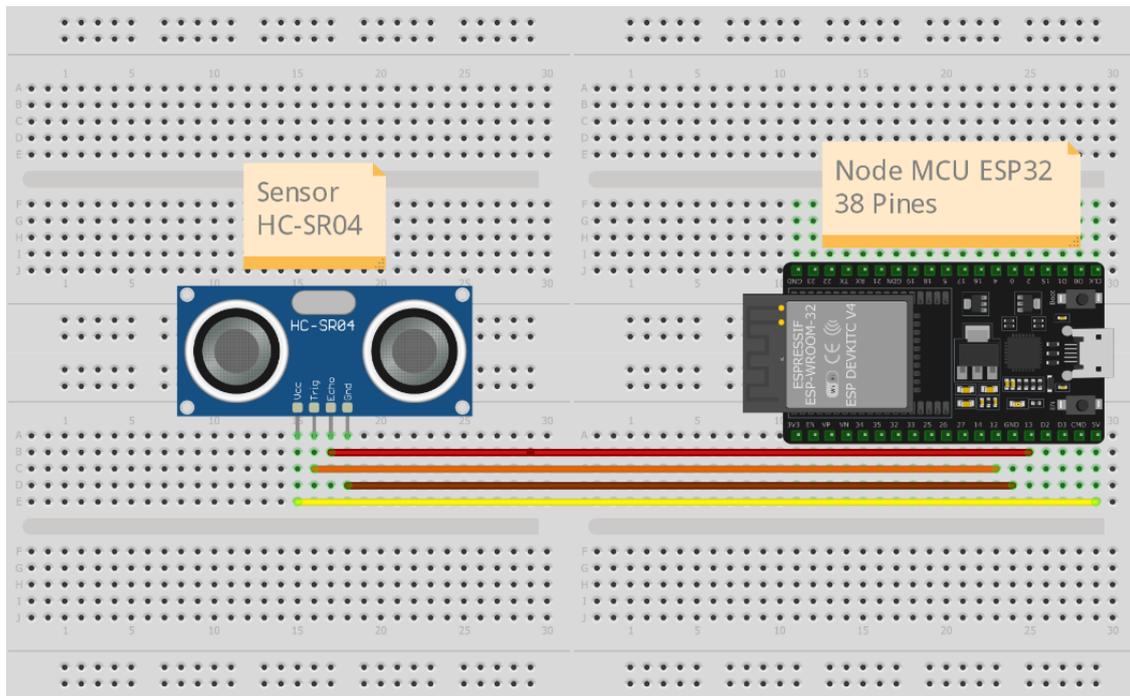
c) Conexiones eléctricas:

- Conectar el pin VCC del HC-SR04 al pin 5V del ESP32.
- Conectar el pin GND del HC-SR04 a uno de los pines GND del ESP32.
- Conectar el pin Trig del HC-SR04 al pin GPIO12 del ESP32.
- Conectar el pin Echo del HC-SR04 al pin GPIO13 del ESP32

d) Configuraciones de Software

Figura 20

Interconexión del HC-SR04 y el ESP32 de 38 pines



Fuente: Autoría

Ahora, el siguiente paso es la conexión del módulo LoRa RYLR998 al nodo sensor para el envío de los datos hacia el nodo gateway, que será el receptor de la información. El módulo LoRa RYLR998 se integrará con el ESP32 para permitir la transmisión de datos de forma inalámbrica a largas distancias. Este proceso implica conectar los pines de transmisión (TX) y recepción (RX) del RYLR998 a los pines UART correspondientes del ESP32, asegurando que ambos dispositivos compartan una línea común de alimentación (VCC) y tierra (GND). Una vez realizada esta conexión física, se configurará el módulo LoRa a través del ESP32 utilizando comandos AT específicos, los cuales establecerán parámetros como la frecuencia de operación, la tasa de datos y el modo de transmisión. Con esta configuración, el nodo sensor podrá enviar las lecturas del sensor ultrasónico al nodo gateway de manera eficiente, donde los datos serán procesados o enviados a un

servidor para su análisis, los pines pueden ser evidenciados en la **Figura 21**, la interconexión del módulo LoRa se puede apreciar en la **Figura 22**.

Pasos generales para la conexión entre el módulo lora RYLR998 al ESP32 (38 Pines)

- e) Reunión componentes Necesarios
- f) Identificación de pines de cada componente

Figura 21

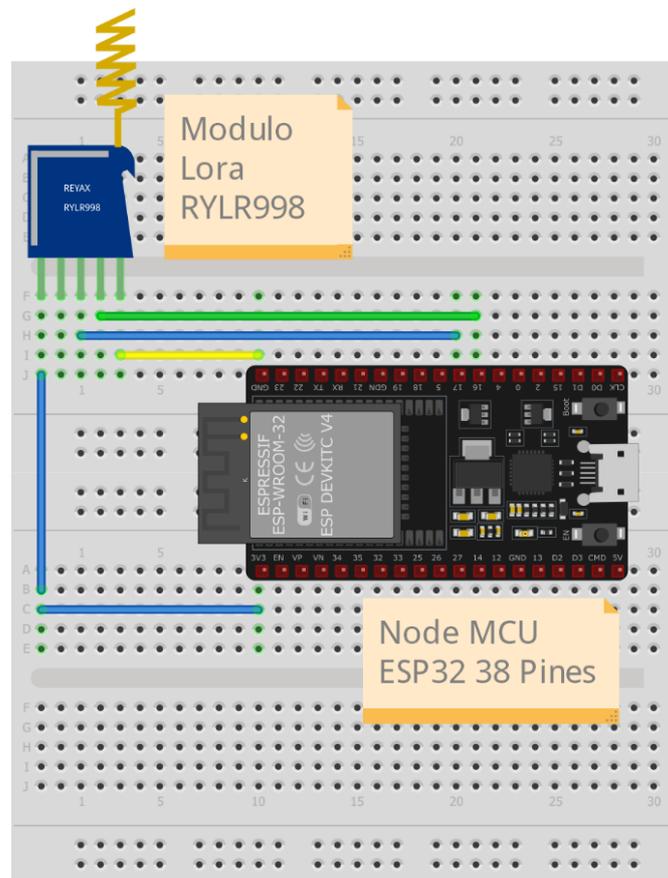
Identificación de pines del módulo lora RYLR998



Fuente: Autoría

g) Conexiones eléctricas

- Conectar el pin VCC del módulo LoRa al pin 3.3V del ESP32.
- Conectar el pin GND del módulo LoRa a uno de los pines GND del ESP32.
- Conectar el pin TX del módulo LoRa al pin RX2 (GPIO16) del ESP32.
- Conectar el pin RX del módulo LoRa al pin TX2 (GPIO17) del ESP32.

Figura 22*Interconexión de Módulo LoRa al ESP32**Fuente: Autoría*

3.3.3.2. Nodo Gateway LoRa

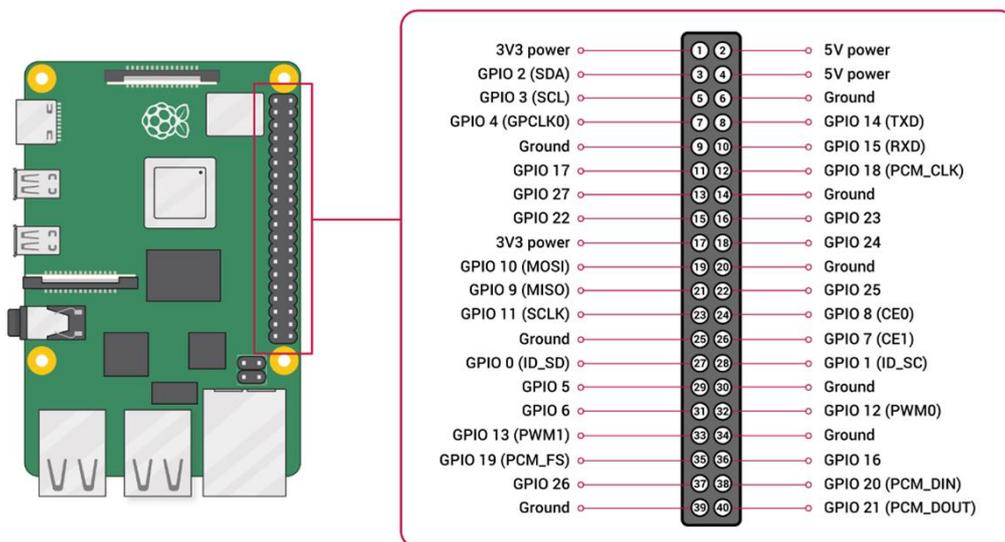
La conexión de pines entre el Nodo Sensor, que utiliza un microcontrolador ESP32, y el Nodo Gateway, que está basado en una Raspberry Pi 4, establece una comunicación efectiva entre ambos dispositivos en una red de sensores. Mediante una interfaz adecuada, se establecerán conexiones físicas y lógicas entre los pines GPIO (General Purpose Input/Output) de ambos dispositivos, permitiendo la transmisión y recepción de datos de los sensores hacia la Raspberry Pi 4 para su posterior procesamiento o envío a sistemas de gestión, los pines de la Raspberry pi 4 se puede evidenciar en la **Figura 23**.

Pasos generales para la conexión entre el ESP32 (38 Pines) y el microcontrolador Raspberry Pi 4 Modelo B

- a) Reunión componentes Necesarios
- b) Identificación de pines de cada componente

Figura 23

Pines de la Raspberry Pi 4 Modelo B



Nota. Tomado de Pines GPIO y su programación, por (Solectro, 2024).

- c) Conexiones eléctricas

- Pin 2 (3.3V) de la Raspberry Pi 4B (RPi4):
Conectado a la línea de alimentación positiva de la breadboard.
- Pin 6 (GND) de la Raspberry Pi 4B (RPi4):
Conectado a la línea de tierra de la breadboard.

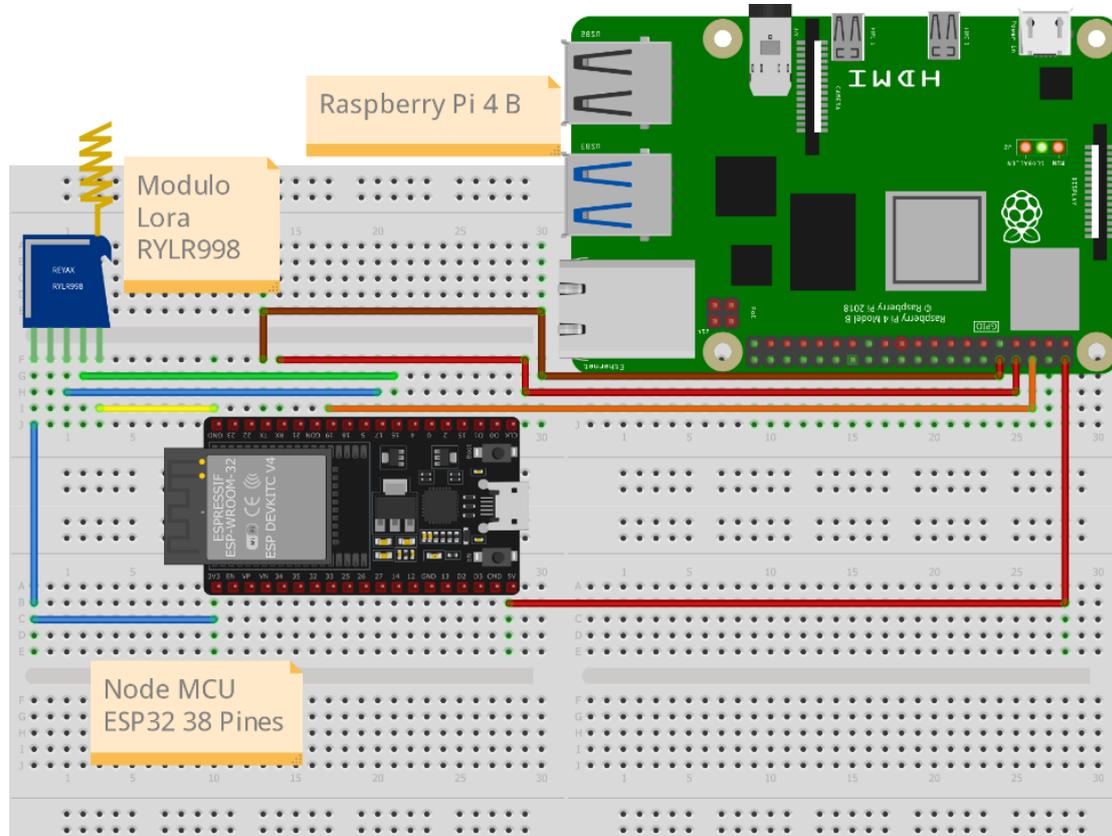
- Pin 8 (GPIO 14 / TXD) de la Raspberry Pi 4B (RPi4): Conectado a un pin en el ESP32 (probablemente RX).
- Pin 10 (GPIO 15 / RXD) de la Raspberry Pi 4B (RPi4): Conectado a un pin en el ESP32 (probablemente TX).

En lo que respecta a su alimentación, la Raspberry Pi 4 (RPi4) proporciona una alimentación de 3.3V a la breadboard y conecta su línea de GND a la línea de tierra de la breadboard. El ESP32 inicialmente toma su alimentación de las líneas de 3.3V y GND de la breadboard. Sin embargo, dado que el ESP32 requiere un voltaje de 5V para un funcionamiento óptimo, debe alimentarse por separado directamente con 5V en lugar de utilizar los 3.3V proporcionados por la RPi4.

La Raspberry Pi 4 (RPi4) y el ESP32 están conectados para comunicación serial. En esta configuración, el pin 8 (TXD) de la RPi4 está conectado al pin GPIO3 (RX) del ESP32, y el pin 10 (RXD) de la RPi4 está conectado al pin GPIO1 (TX) del ESP32. Esta disposición permite la transmisión y recepción de datos entre ambos dispositivos de manera eficiente, con el pin de transmisión de la RPi4 conectado al pin de recepción del ESP32 y viceversa, asegurando una comunicación serial bidireccional adecuada, esto se puede verificar gráficamente en la **Figura 24**.

Figura 24

Diagrama de Interconexiones del Nodo Gateway del Sistema de Medición de Nivel de Agua



Fuente: Autoría

En lo que respecta a la conexión del Módulo Lora es la misma que se ha realizado en los nodos sensores.

3.3.4 Armado

En el proceso de armado del sistema de medición de nivel de agua y alerta mediante notificaciones, se llevarán a cabo varias etapas importantes. Primero, se realizará la configuración del Nodo Sensor, que incluye la creación de un diagrama de flujo específico para el nodo y un diagrama de interconexión final para asegurar todas las conexiones necesarias. A continuación, se procederá con la programación del ESP32, un componente esencial para la gestión y

transmisión de datos del sensor. Posteriormente, se configurará el Gateway LoRa para facilitar la comunicación de largo alcance entre el nodo sensor y la central de procesamiento. La Raspberry Pi 4 también será programada para recibir, procesar y enviar los datos recopilados a la plataforma ThingSpeak. Finalmente, se llevará a cabo la configuración de ThingSpeak, que permitirá el almacenamiento, análisis y visualización de los datos en tiempo real, además de enviar alertas cuando los niveles de agua superen los umbrales predefinidos. Este proceso integral asegura un sistema robusto y eficiente para la monitorización y alerta del nivel de agua.

3.3.4.1 Configuración del Nodo Sensor

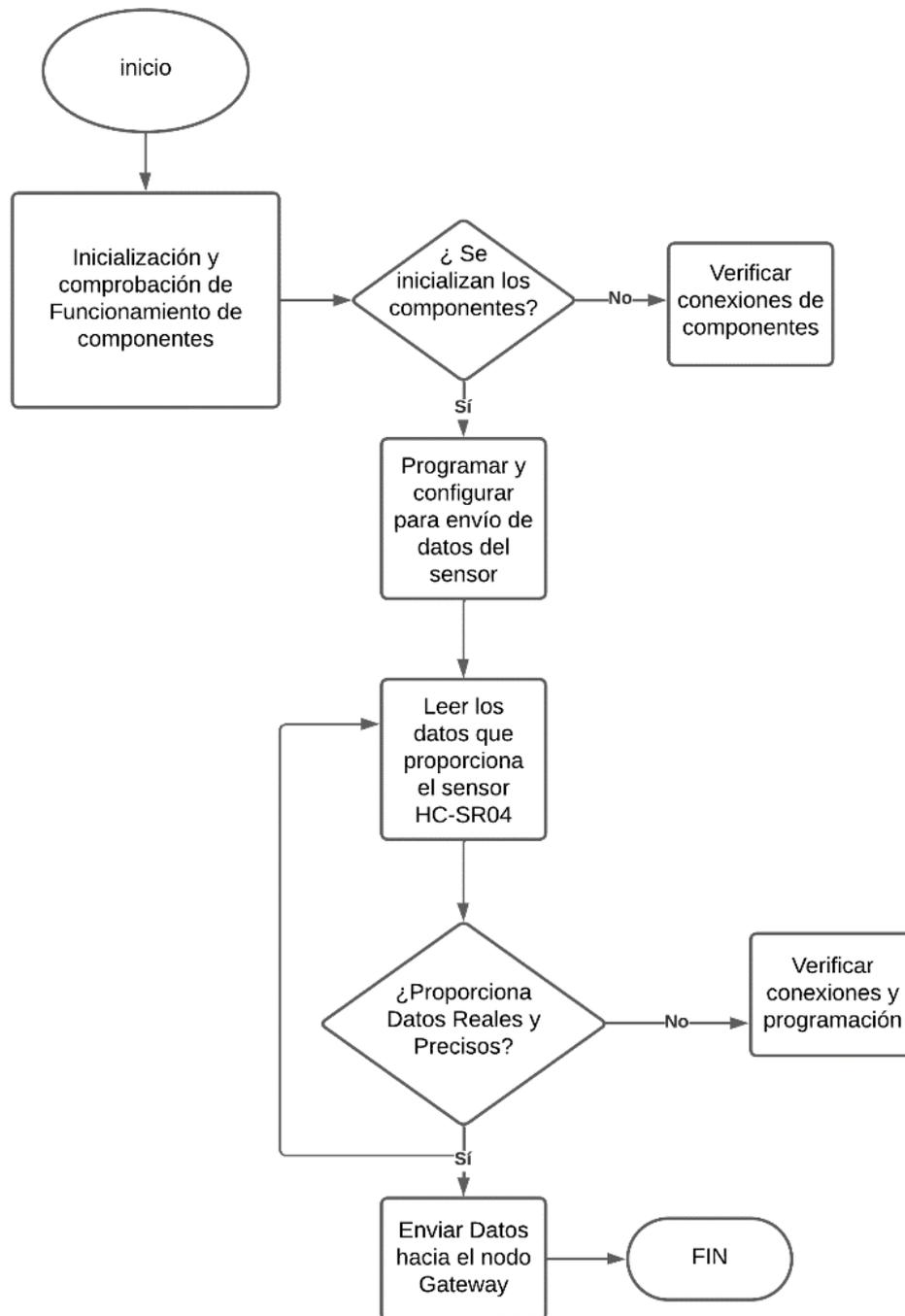
La configuración del Nodo Sensor incluye la creación de un diagrama de flujo para definir el funcionamiento lógico del sensor y un diagrama de interconexión final que asegura todas las conexiones físicas necesarias. Se conectará el sensor de ultrasonido HC-SR04 al ESP32 y se programará para medir los niveles de agua y transmitir los datos al Gateway LoRa. Esta configuración es importante para garantizar la precisión y la fiabilidad de las mediciones en el sistema de monitoreo.

- **Diagrama de Flujo del Nodo Sensor**

El proceso del diagrama de flujo del Nodo Sensor comienza con la inicialización y comprobación del funcionamiento de los componentes. Si los componentes se inicializan correctamente, se procede a programar y configurar el sensor para el envío de datos. Luego, se leen los datos proporcionados por el sensor HC-SR04. Si los datos son reales y precisos, se envían hacia el nodo Gateway. En caso de que los componentes no se inicialicen o los datos no sean precisos, se verifica la conexión y la programación de los componentes para solucionar los problemas antes de continuar con el proceso, se puede apreciar todo el proceso en el diagrama de flujo de la **Figura 25**.

Figura 25

Diagrama de Flujo del nodo Sensor



Fuente: Autoría

- **Diagrama de Interconexión Final del nodo sensor**

El diagrama de la **Figura 26** muestra la interconexión del nodo Sensor incluye un sensor de ultrasonido HC-SR04, un módulo LoRa RYLR998, un LED con una

resistencia y una placa NodeMCU ESP32 de 38 pines. A continuación, se describe la conexión general:

El sensor HC-SR04 cuenta con las siguientes conexiones:

- a) VCC: Conectado a la línea de alimentación positiva del ESP32.
- b) Trig: Conectado a un pin digital del ESP32 para enviar la señal de disparo.
- c) Echo: Conectado a otro pin digital del ESP32 para recibir la señal de eco.
- d) GND: Conectado a la línea de tierra de la breadboard.

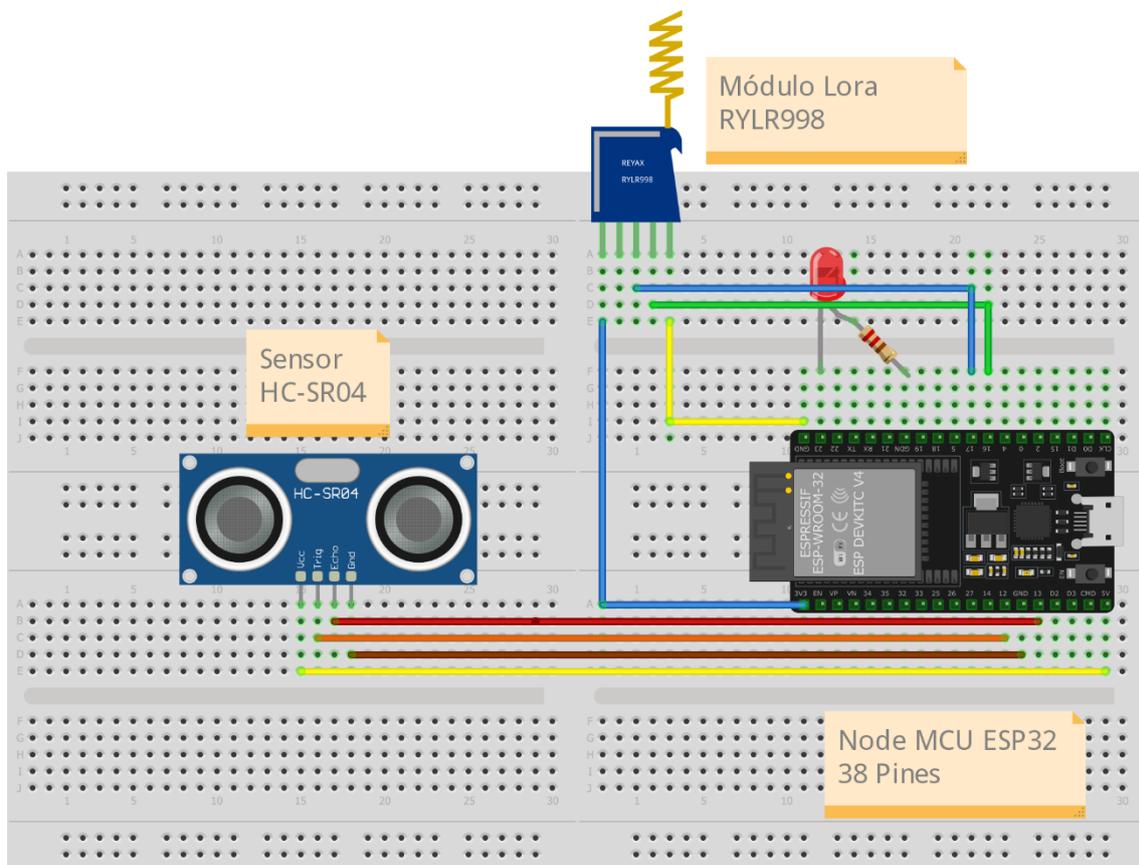
Módulo LoRa RYLR998:

- VCC: Conectado a la línea de alimentación positiva de la breadboard.
- GND: Conectado a la línea de tierra de la breadboard.
- TX: Conectado al pin RX del ESP32.
- RX: Conectado al pin TX del ESP32.

El LED está conectado al ESP32 y se utiliza para visualizar cuando hay salida o envío de datos. Esto proporciona una indicación visual inmediata de la actividad del sistema, ayudando a los usuarios a confirmar que los datos están siendo transmitidos correctamente.

Figura 26

Diagrama de Interconexión del Nodo Sensor Completo



Fuente: Autoría

- **Programación del ESP32**

En el ESP32, se debe realizar la programación necesaria para enviar los datos del sensor utilizando la tecnología LoRa. Esto incluye configurar la comunicación serial con el módulo RYLR998, leer los datos del sensor HC-SR04 y transmitir estos datos de forma inalámbrica, asegurando una transmisión eficiente y de largo alcance.

El código en su primera parte (**Figura 27**) está escrito en lenguaje C++, diseñado para ejecutarse en una placa ESP32. Es importante instalar las librerías adecuadas para el ESP32 en el entorno de desarrollo (como Arduino IDE) y configurar correctamente el hardware. Los prerequisites incluyen tener el módulo LoRa RYLR998 correctamente

conectado al ESP32 y asegurarse de que los pines de conexión (TX, RX, trig, eco, y LED) estén configurados correctamente.

Figura 27

Configuración Inicial de Parámetros Lora en placa ESP32 del nodo sensor

```
EmisorLora.ino
1 String lora_band = "915000000";
2 String lora_networkid = "18";
3 String lora_address = "1";
4 String lora_RX_address = "2";
-
```

Fuente: Autoría

La variable `lora_band` establece la banda de frecuencia en Hz, que debe ajustarse según el país o región para cumplir con las regulaciones locales. La variable `lora_networkid` define la identificación de la red LoRa, permitiendo que los dispositivos en la misma red se reconozcan entre sí. La variable `lora_address` asigna una dirección específica al módulo LoRa en el ESP32, mientras que `lora_RX_address` define la dirección del módulo receptor LoRa con el que se establecerá la comunicación (**Figura 28**).

Figura 28

Declaración y Configuración de Variables en placa ESP32 del nodo sensor

```
6 int led = 22; //
7 int trig = 12;
8 int eco = 13;
9 int duracion;
10 int distancia;
11 int tanque;
```

Fuente: Autoría

En estas líneas de código de la **Figura 29** se tienen varias declaraciones y configuraciones de variables: “`int led = 22;`” define que un LED está conectado al pin

D22. “int trig = 12;” y “int eco = 13;” configuran los pines 12 y 13 como el transmisor y receptor del sensor de ultrasonido HC-SR04, respectivamente. “int duracion;” se utiliza para almacenar la duración del pulso del eco recibido, y “int distancia;” se utiliza para calcular la distancia basada en esa duración. Finalmente, “int tanque;” es una variable que parece estar destinada a algún uso relacionado con el tanque, aunque su propósito específico no se detalla en este fragmento de código.

Figura 29

Configuración Inicial en la Función setup () en placa ESP32 del nodo sensor

```

13 void setup() {
14   pinMode(trig, OUTPUT);
15   pinMode(eco, INPUT);
16   Serial2.begin(115200, SERIAL_8N1, 16, 17); // F
17   pinMode(led, OUTPUT); // Configura el pin del L
18   delay(1500);

```

Fuente: Autoría

Estas líneas de código de la **Figura 30** se encuentran dentro de la función “setup ()”, que se ejecuta una vez al inicio del programa para configurar los pines y la comunicación. “pinMode (trig, OUTPUT);” y “pinMode(eco, INPUT);” configuran los pines del sensor de ultrasonido HC-SR04 como salida y entrada, respectivamente. “Serial2.begin(115200, SERIAL_8N1, 16, 17);” inicializa la comunicación serial en el puerto serie 2 del ESP32 con una velocidad de 115200 baudios, utilizando los pines 16 (RXD) y 17 (TXD). “pinMode (led, OUTPUT);” configura el pin del LED como salida. Finalmente, “delay (1500);” introduce una pausa de 1500 milisegundos para permitir que el sistema se estabilice antes de continuar con la ejecución del programa.

Figura 30

Configuración Inicial del Módulo LoRa con Comandos AT en placa ESP32 del nodo sensor

```

20 // Configuración inicial del módulo LoRa usando com
21 Serial2.println("AT+BAND=" + lora_band); // Confi
22 delay(500);
23 Serial2.println("AT+ADDRESS=" + lora_address); //
24 delay(500);
25 Serial2.println("AT+NETWORKID=" + lora_networkid);
26 delay(1500);
27 }

```

Fuente: Autoría

Estas líneas de código de la **Figura 31** configuran el módulo LoRa mediante comandos AT enviados a través del puerto serial del ESP32. Primero, se establece la banda de frecuencia utilizando el comando “AT+BAND”, seguido de una pausa de 500 milisegundos para asegurar su procesamiento. Luego, se configura la dirección del módulo con el comando “AT+ADDRESS” y otra pausa de 500 milisegundos. Finalmente, se establece la identificación de la red con “AT+NETWORKID”, seguida de una pausa de 1500 milisegundos para asegurar que todos los comandos se apliquen correctamente.

Figura 31

Medición de Distancia e Indicador LED en placa ESP32 del nodo sensor

```

29 void loop() {
30   digitalWrite(trig, HIGH);
31   delay(10);
32   digitalWrite(trig, LOW);
33   duracion = pulseIn(eco, HIGH); //
34   distancia = duracion / 59; //Ca
35   String datos = String(distancia);
36   digitalWrite(led, HIGH); // Encien
37   delay(2000); // Espera 5 segundos

```

Fuente: Autoría

Estas líneas de código dentro de la función “loop ()” de la **Figura 32** se realizan el ciclo de medición y envío de datos de distancia. Primero, se envía una señal de disparo al sensor de ultrasonido HC-SR04 activando el pin “trig” por 10 milisegundos. Luego, se

mide la duración del pulso recibido en el pin “eco”, que corresponde al tiempo que tarda la señal en regresar al sensor. Esta duración se convierte en una distancia en centímetros dividiendo por 59. La distancia se convierte en una cadena de texto y se almacena en la variable “datos”. El LED conectado al pin “led” se enciende para indicar que se está procesando la medición, y luego se introduce una pausa de 2000 milisegundos.

Figura 32

Envío de Datos y Control de Indicador LED en placa ESP32 del nodo sensor

```

39 // Envío de Datos y Control de Indicador LED
40 Serial.println("AT+SEND=" + lora_RX_address + "," + String(datos.length()) + "," + datos);
41 Serial.println("Enviando mensaje: " + datos);
42 digitalWrite(led, LOW); // Apaga el LED indicador
43 delay(1000); // Espera x segundos antes de enviar el siguiente mensaje
44 }
45

```

Fuente: Autoría

Estas líneas de código dentro de la función “loop ()” manejan el envío de datos medidos a través del módulo LoRa y proporcionan una indicación visual. La línea 40 envía un comando AT al módulo LoRa para transmitir los datos de distancia al receptor especificado por “lora_RX_address”. Luego, la línea 41; imprime en el monitor serial del ESP32 el mensaje enviado para verificación. El LED conectado al pin led se apaga para indicar que el envío se ha completado, seguido de una pausa de 1000 milisegundos antes de la próxima medición y envío.

3.3.4.2 Configuración Gateway Lora

La configuración del nodo gateway, que incluye un ESP32 y una Raspberry Pi 4 (RPi4), es necesaria para el funcionamiento eficiente del sistema de monitoreo y alerta. El ESP32 actúa como el punto de recolección de datos de los sensores, utilizando la tecnología LoRa para recibir y transmitir información a larga distancia. La RPi4, por su parte, se encarga de procesar estos datos y enviarlos a la plataforma ThingSpeak para su almacenamiento y análisis. La programación del ESP32 asegura que los datos sean

recogidos y enviados correctamente, mientras que la programación de la RPi4 garantiza que estos datos se procesen y transmitan de manera efectiva a ThingSpeak, permitiendo así la visualización en tiempo real y la generación de alertas basadas en los datos recibidos.

- **Programación de ESP32**

Para la recepción de datos en el Gateway Lora se ha hecho uso de otra placa ESP32 la cual realizará la recepción de los datos y luego los enviará a la Raspberry pi 4 para el respectivo procesamiento de los mismos.

A continuación, se presentará el código utilizado para la recepción de los datos tanto del nodo 1 como del nodo 2 a través de los módulos LoRa usados.

Figura 33

Configuración Inicial de Parámetros, Declaración de Variables en placa ESP32 del nodo Gateway

```

1 String lora_band = "915000000"; // Banda de frecuencia (Hz) dependiendo del país/región
2 String lora_networkid = "18"; // Identificación de la red LoRa
3 String lora_address = "2"; // Dirección del módulo LoRa
4
5 int led = 22; // LED conectado al pin D22 del ESP32
6 int ledb = 23; // LED conectado al pin D23 del ESP32
7 int vn1;
8 int vn2;
9 String sd;
10 int cont = 0;
11
12 String textoEntrada; // Almacena los datos recibidos del emisor
13

```

Fuente: Autoría

Como se muestra en la **Figura 33** se configura un sistema basado en un microcontrolador para recibir datos de nodos LoRa. Se definen varios parámetros como la banda de frecuencia, la identificación de la red LoRa y la dirección del módulo de comunicación. También se asignan pines para controlar dos LEDs, los cuales sirven como indicadores visuales. Se crean variables para almacenar los valores de los nodos, un

contador para llevar la cuenta de las lecturas, y otra variable para almacenar los datos recibidos desde los nodos. El propósito de este código es establecer la comunicación LoRa y visualizar los datos mediante los LEDs.

Figura 34

Configuración Inicial del Sistema LoRa y Periféricos en placa ESP32 del nodo Gateway

```

14 void setup() {
15     pinMode(led, OUTPUT); // Configura el pin del LED como salida
16     pinMode(ledb, OUTPUT); // Configura el pin del LED como salida
17     Serial2.begin(115200, SERIAL_8N1, 16, 17); // Inicializa el puerto serie 2 (pin 16 RXD, 17 TXD)
18     Serial.begin(115200); // Inicializa el puerto serie a 115200 bps
19     delay(1500);
20     Serial2.println("AT+BAND=" + lora_band); // Configura la banda de frecuencia del módulo LoRa
21     delay(500);
22     Serial2.println("AT+NETWORKID=" + lora_networkid); // Configura la identificación de la red LoRa
23     delay(500);
24     Serial2.println("AT+ADDRESS=" + lora_address); // Configura la dirección del módulo LoRa
25     delay(1000);
26
27     Serial.println("Inicialización completa");
28 }

```

Fuente: Autoría

En este bloque de código de la **Figura 34**, se configura el sistema para iniciar la comunicación con el módulo LoRa y los periféricos del microcontrolador. Primero, se definen los pines para los LEDs como salidas. Luego, se inicializan dos puertos serie: uno para la comunicación con el módulo LoRa y otro para la depuración en el monitor serie. Después de un breve retraso, se envían comandos al módulo LoRa para configurar parámetros como la banda de frecuencia, la identificación de la red y la dirección del módulo. Finalmente, se imprime un mensaje en el monitor serie para indicar que la inicialización se ha completado correctamente.

Figura 35

Configuración Inicial en la void loop en placa ESP32 del nodo Gateway

```
30 void loop() {
31     if (Serial2.available()) { // Verifica si hay datos disponibles en el puerto serie 2
32         textoEntrada = Serial2.readString(); // Lee el string recibido desde el transmisor
33         textoEntrada.trim(); // Elimina espacios en blanco al inicio y al final del string
34
35         int primeraComa = textoEntrada.indexOf(',');
36         int segundaComa = textoEntrada.indexOf(',', primeraComa + 1);
37         int tercerComa = textoEntrada.indexOf(',', segundaComa + 1);
38
39         // Extrae el ID del emisor
40         String emisorID = textoEntrada.substring(0, primeraComa);
41         // Extrae el substring correspondiente al número
42         String valorString = textoEntrada.substring(segundaComa + 1, tercerComa);
43         // Convierte el substring a un número entero
44         int valor = valorString.toInt();
45     }
```

Fuente: Autoría

En el bloque de código proporcionado en la **Figura 35** se aprecia El loop () se encarga de leer los datos que llegan desde el módulo LoRa conectado al puerto serie 2 del ESP32. Primero, verifica si hay datos disponibles en el puerto serie y, si es así, lee un string completo. Luego, elimina los espacios en blanco al inicio y al final del string. Después, localiza las posiciones de las comas dentro del string para separar la información: extrae el ID del emisor antes de la primera coma y obtiene el valor numérico entre la segunda y la tercera coma, convirtiéndolo a un número entero. Esta información será utilizada posteriormente para procesar o mostrar los datos provenientes del sensor.

Figura 36

Recepción y Procesamiento de Datos de Nodos LoRa con Indicadores Visuales en placa ESP32 del nodo Gateway

```
46 // Imprime el valor obtenido con el ID del emisor
47 if (emisorID == "+RCV=1") {
48     vn1 = valor;
49     digitalWrite(led, HIGH); // Enciende el LED del Emisor 1
50     delay(200); // Pequeña pausa para que el LED se note
51     digitalWrite(led, LOW); // Apaga el LED del Emisor 1
52     cont++;
53
54 } else if (emisorID == "+RCV=3") {
55     vn2 = valor;
56     digitalWrite(ledb, HIGH); // Enciende el LED del Emisor 2
57     delay(200); // Pequeña pausa para que el LED se note
58     digitalWrite(ledb, LOW); // Apaga el LED del Emisor 2
59     cont++;
60 }
```

Fuente: Autoría

El código de la **Figura 36** evalúa el valor recibido de un identificador de emisor para determinar qué acción realizar según el emisor que haya enviado los datos. Si el identificador corresponde a un emisor específico, el valor se guarda en una variable, se activa un LED para indicar que se ha recibido información de ese emisor, se realiza una breve pausa para que el LED sea visible, y luego se apaga. Además, se incrementa un contador para llevar un registro de la cantidad de datos procesados. Este proceso permite visualizar mediante los LEDs cuál emisor está enviando datos y seguir un conteo de las recepciones.

Figura 37

Envío de Datos de Nodos LoRa y Reinicio de Contador en placa ESP32 del nodo Gateway

```
62     sd = "N1: " + String(vn1) + ", N2: " + String(vn2);
63     if (cont == 3) {
64         Serial.println(sd);
65         cont = 0;
66     }
67     //Serial.println(String(cont));
68 }
69 }
```

Fuente: Autoría

Este código que evidencia la **Figura 37** forma parte del proceso de impresión y envío de los datos recibidos de los dos nodos LoRa. Se crea una cadena de texto sd que combina los valores de los nodos (vn1 y vn2) con etiquetas que indican su origen, y se asigna a la variable sd. Luego, si el contador cont ha alcanzado el valor 3, se imprime la cadena sd en el puerto serie, mostrando los datos de los nodos, y se reinicia el contador a cero para volver a contar las siguientes mediciones. El contador cont se utiliza para limitar la frecuencia de los mensajes enviados al puerto serie.

- **Programación Raspberry PI 4**

La Raspberry Pi 4 debe recibir los datos enviados desde el ESP32 a través de una conexión serial y, a continuación, transmitir estos datos a la plataforma ThingSpeak. Esto permite el análisis de datos en tiempo real, facilitando la monitorización y visualización remota de la información recolectada.

Figura 38

Importación de librerías e inicio de programación del código Python en Raspberry Pi

```
1 import serial
2 import time
3 import os
4 import re
5 import requests
6
7 def send_email(api_key, domain, from_email, to_email, subject, text):
8     url = f"https://api.mailgun.net/v3/{domain}/messages"
9     auth = ("api", api_key)
10    data = {
11        "from": from_email,
12        "to": to_email,
13        "subject": subject,
14        "text": text
15    }
16    response = requests.post(url, auth=auth, data=data)
17    if response.status_code == 200:
18        print("Correo electrónico enviado con éxito.")
19    else:
20        print(f"Error al enviar el correo: {response.status_code}, {response.text}")
```

Fuente: Autoría

En la **Figura 38** se implementa el envío de correos electrónicos utilizando el servicio Mailgun. Para ello, se incluyen librerías que permiten gestionar datos seriales, tiempos, operaciones del sistema, expresiones regulares y solicitudes web. En la función encargada del envío, se configuran los detalles necesarios, como la dirección del servicio, las credenciales de autenticación y los datos del mensaje (emisor, receptor, asunto y contenido). El correo se envía mediante una solicitud HTTP POST, y en caso de errores, el sistema muestra un mensaje indicando el problema y la respuesta de la API.

Figura 39

Código de configuración de puertos seriales y análisis de datos en Raspberry Pi

```

21
22 def open_serial():
23     return serial.Serial(
24         port='/dev/serial0', # Puerto serie de la Raspberry Pi 4
25         baudrate=115200, # Velocidad de baudios
26         parity=serial.PARITY_NONE,
27         stopbits=serial.STOPBITS_ONE,
28         bytesize=serial.EIGHTBITS,
29         timeout=1
30     )
31
32 def parse_line(line):
33     match = re.search(r"N1: (\d+), N2: (\d+)", line)
34     if match:
35         n1 = int(match.group(1))
36         n2 = int(match.group(2))
37         return n1, n2
38     match = re.search(r"1: (\d+), N2: (\d+)", line)
39     if match:
40         n1 = int(match.group(1))
41         n2 = int(match.group(2))
42         return n1, n2
43     return None, None
44

```

Fuente: Autoría

En el código que se evidencia en la *se implementa* el envío de correos electrónicos utilizando el servicio Mailgun. Para ello, se incluyen librerías que permiten gestionar datos seriales, tiempos, operaciones del sistema, expresiones regulares y solicitudes web. En la función encargada del envío, se configuran los detalles necesarios, como la dirección del servicio, las credenciales de autenticación y los datos del mensaje (emisor, receptor, asunto y contenido). El correo se envía mediante una solicitud HTTP POST, y en caso de errores, el sistema muestra un mensaje indicando el problema y la respuesta de la API.

se configura una Raspberry Pi 4 para recibir datos a través de un puerto serie, utilizando una velocidad específica de 115200 baudios. También incluye una función que analiza los mensajes recibidos, buscando dos números etiquetados como "N1" y "N2". Si encuentra estos números en el mensaje, los extrae, los convierte en valores que se pueden usar y los devuelve como una pareja de números. Si no encuentra estos números, devuelve

dos valores vacíos. Este proceso es útil para recibir y organizar datos de otros dispositivos conectados a la Raspberry Pi.

Figura 40

Código de recepción e impresión de datos en Raspberry Pi

```

45 def main():
46     ser = None
47     error_count = 0
48     empty_line_count = 0
49     cont = 175
50     contb =175
51     # Cambiar permisos del puerto serie
52     try:
53         os.system('sudo chmod -R 777 /dev/serial0')
54     except Exception as e:
55         return
56     try:
57         ser = open_serial()
58         while True:
59             try:
60                 if ser.in_waiting > 0:
61                     line = ser.readline().decode('utf-8').rstrip()
62                     if line:
63                         print(line)
64                         n1, n2 = parse_line(line)
65                         if n1 is not None and n2 is not None:
66                             #print(f"N1: {n1}, N2: {n2}")
67                             print(f"conttador bajo: {cont},conttador alto: {contb}")

```

Fuente: Autoría

Este fragmento de código de la **Figura 40** gestiona la comunicación con un dispositivo conectado a un puerto serie en una Raspberry Pi. Primero, establece algunos contadores y cambia los permisos del puerto para garantizar el acceso. Luego, intenta abrir el puerto serie y, si tiene éxito, entra en un bucle infinito donde continuamente lee los datos que llegan por el puerto. Si recibe una línea de texto, la decodifica y la procesa para extraer dos números específicos. Si los números son válidos, imprime los valores de los contadores. El proceso sigue repitiéndose, permitiendo recibir y mostrar datos del dispositivo conectado en tiempo real.

Figura 41

Código para establecimiento de conexión y envío de datos a ThingSpeak en Raspberry Pi

```

67     print(f"conttador bajo: {cont},conttador altb: {contb}")
68     t1 = 210 - n1
69     t2 = 210 - n2
70     print(f"t1: {t1},t2: {t2}")
71     # Enviar a ThingSpeak
72     requests.get('https://api.thingspeak.com/update?api_key=JAR16GKQFIV73PSS&field1=' + str(t1) + '&field2=' + str(t2))
73     # Enviar correo si t1 o t2 son menores a 180
74     if t1 < 180 or t2 < 180:
75         cont += 1
76         if cont ==180: # tiempo de reenvio en segundos
77             send_email(
78
79                 API_KEY, DOMAIN, FROM_EMAIL, TO_EMAIL, SUBJECT,
80                 f"Alerta uno de los tanques tiene BAJO nivel de agua. Valores actuales: t1={t1}, t2={t2} REVISE LA PAGINA WEB
81                 )
82                 cont = 0
83                 contb = 175

```

Fuente: Autoría

El código de la **Figura 41** imprime los valores de dos contadores, luego calcula dos variables (t1 y t2) restando los valores de otros dos números de 210. Después, envía estos valores a la plataforma ThingSpeak a través de una solicitud web. Si alguno de los valores t1 o t2 es menor a 180, incrementa un contador en 100. Cuando el contador alcanza 180, envía un correo de alerta para notificar que uno de los tanques tiene un nivel de agua bajo, incluyendo los valores de t1 y t2. Finalmente, reinicia los contadores a cero. Sin embargo, el código presenta algunos errores que deben corregirse, como la forma en que se concatenan las cadenas y el operador usado en la condición que compara el contador.

Figura 42

Código de establecimiento de Umbrales del nivel de agua en Raspberry Pi

```

84     else:
85         cont = 175
86
87     if t1 > 200 or t2 > 200:
88         contb += 1
89         if contb == 180:#tiempo de reenvio en segundos
90             send_email(
91                 API_KEY, DOMAIN, FROM_EMAIL, TO_EMAIL, SUBJECT,
92                 f"Alerta uno de los tanques tiene ALTO nivel de agua. Valores actuales: t1={t1}, t2={t2} REVISE LA PAGINA WEB
93             )
94             contb = 0
95             cont = 175
96     else:
97         contb = 175
98
99     error_count = 0 # Reiniciar el contador de errores tras una lectura exitosa
100    empty_line_count = 0 # Reiniciar el contador de lineas vacias tras una lectura exitosa
101    else:
102        empty_line_count += 1
103        if empty_line_count >= 1:
104            raise OSError(5, 'Demasiadas cadenas vacias consecutivas')
105

```

Fuente: Autoría

El código de la **Figura 42** realiza varias acciones dependiendo de los valores de t1 y t2. Si ambos son mayores que 200, incrementa un contador. Cuando este contador alcanza 180, envía un correo de alerta para notificar que uno de los tanques tiene un nivel de agua alto, incluyendo los valores actuales de t1 y t2. Luego, reinicia los contadores a cero. Si t1 y t2 no son mayores a 200, reinicia el contador a un valor específico. Además, si se recibe una lectura exitosa, restablece los contadores de errores y de líneas vacías. Si detecta demasiadas líneas vacías consecutivas, lanza un error. Sin embargo, el código contiene errores de sintaxis, como una incorrecta concatenación de cadenas y manejo erróneo de las variables.

Figura 43

Código de establecimiento de líneas de errores y reinicio de contadores en Raspberry

Pi

```

105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

except serial.SerialException:
    error_count += 1
except OSError as e:
    if e.errno == 5:
        error_count += 1
        if empty_line_count >= 1:
            time.sleep(1) # Espera antes de intentar de nuevo
    else:
        break
except Exception:
    break

if error_count >= 1:
    os.system('sudo chmod -R 777 /dev/serial0')
    ser.close()
    time.sleep(2) # Espera antes de reabrir la conexión
    ser = open_serial()
    error_count = 0
    empty_line_count = 0

```

Fuente: Autoría

Este código gestiona diferentes tipos de errores durante la ejecución del programa. Si ocurre un error relacionado con la comunicación serie, se incrementa un contador de errores. Si se detecta un error específico de tipo `OSError`, también se suma al contador. Si se reciben demasiadas líneas vacías consecutivas, el código espera un segundo antes de intentar leer nuevamente. Si no hay errores, el programa termina el bucle. Si se producen múltiples errores, el código cambia los permisos del puerto serie, cierra la conexión y espera dos segundos antes de reabrirlo. Finalmente, los contadores de errores y de líneas vacías se reinician. Esto se puede evidenciar en la **Figura 43** que se presenta a continuación.

Figura 44

Establecimiento de excepciones en el código en Raspberry Pi

```

126 except serial.SerialException:
127     pass
128 except Exception:
129     pass
130 finally:
131     if 'ser' in locals() and ser.is_open:
132         ser.close()

```

Fuente: Autoría

El código de la **Figura 44** se maneja dos tipos de excepciones. Si ocurre una excepción relacionada con la comunicación serie, simplemente se ignora y el programa continúa. Si se presenta cualquier otro tipo de error, también se ignora. Al final del bloque de código, en la sección `finally`, que siempre se ejecuta, se verifica si el puerto serie está abierto. Si es así, se cierra la conexión del puerto serie para asegurarse de que se cierre correctamente, incluso si ocurrió un error durante la ejecución del programa.

Figura 45

Establecimiento de variables y parámetros para el envío de notificaciones en Raspberry Pi

```

133
134 # Datos de la cuenta de Mailgun y del correo
135 API_KEY = 'b9981370e34c950f65da3b62ea5eb036-0f1db83d-2e4080c9'
136 DOMAIN = 'sandbox66a6f378995e4e23890087f6dbd1725e.mailgun.org'
137 FROM_EMAIL = 'gladyscardenas172@gmail.com'
138 TO_EMAIL = 'lizeth.c2698@gmail.com'
139 SUBJECT = 'Notificacion nivel de agua de tanques en CHAMIZO'
140 TEXT = 'Este es el cuerpo del correo electrónico'
141
142 if __name__ == "__main__":
143     while True:
144         try:
145             main()
146         except Exception as e:
147             print(f"El script ha fallado con un error: {e}")
148             time.sleep(5) # Espera antes de reiniciar el script
149

```

Fuente: Autoría

Finalmente se cuenta con la **Figura 45** donde se define las variables necesarias para enviar un correo electrónico utilizando el servicio Mailgun, incluyendo la clave API, el dominio, las direcciones de correo de envío y recepción, el asunto y el cuerpo del mensaje. Luego, dentro de un bloque condicional, se ejecuta un bucle infinito que llama a la función principal. Si ocurre algún error durante la ejecución de esta función, el script captura la excepción y muestra un mensaje de error. Después, espera 5 segundos antes de intentar reiniciar el script, lo que asegura que el programa continúe funcionando incluso si ocurre un error. Sin embargo, hay un error de sintaxis en la asignación de una de las variables, donde la comilla final es incorrecta.

El código visto permite de forma general la toma de datos a través de un puerto serial en una Raspberry Pi, procesando los valores recibidos y enviándolos a la nube a través de ThingSpeak para su monitoreo. Además, gestiona el envío de alertas por correo electrónico utilizando la API de Mailgun si los valores de los datos superan o caen por debajo de ciertos umbrales, activando así una alarma para notificar de posibles condiciones anormales, como niveles bajos o altos de agua en tanques.

La programación de cada uno de los códigos utilizados se podrá encontrar en el **Anexo 3. Códigos de Programación: Repositorio GitHub**.

3.3.4.4 Configuración ThingSpeak

Para la configuración de ThingSpeak se tiene una serie de configuraciones las cuales se enumeran a continuación.

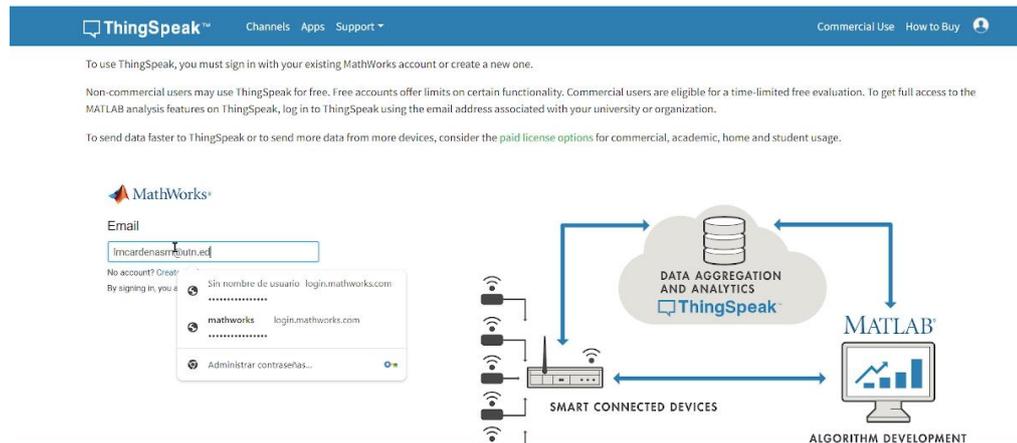
- **Creación de la cuenta**

El primer paso por realizar es la creación de una cuenta en ThingSpeak, como se observa en la **Figura 46**, se ha hecho uso del correo institucional ya que este puede

ofrecer varias ventajas como la mejora en la seguridad y una mejor gestión dentro de la plataforma.

Figura 46

Registro en la plataforma de ThingSpeak



Fuente: Autoría

- **Creación de un Canal**

Como siguiente paso está la creación de un Canal, para ello, en el dashboard de ThingSpeak, seleccionar 'Channels', luego 'My Channels', y hacer clic en 'New Channel'. Llenar los apartados que se muestran en la imagen.

En plataformas como ThingSpeak, un canal es una estructura para organizar y almacenar datos. Un canal es esencialmente un contenedor donde se recogen, almacenan y acceden los datos enviados desde dispositivos conectados, como sensores en un Raspberry Pi o cualquier otro dispositivo de Internet de las Cosas (IoT).

A continuación, se detalla algunas funcionalidades que ofrece un canal de ThingSpeak.

- **Almacenamiento de Datos:** Cada canal en ThingSpeak permite almacenar datos provenientes de múltiples fuentes. Por ejemplo, si estás monitoreando condiciones

ambientales, puedes tener un canal donde un campo almacena datos de temperatura, otro campo almacena datos de humedad, y así sucesivamente.

- **Organización y Segmentación:** Los canales permiten segmentar y organizar los datos según el proyecto o la aplicación. Esto es especialmente útil cuando se manejan múltiples proyectos o dispositivos, ya que cada canal puede dedicarse a un dispositivo específico o a un tipo específico de datos.
- **Visualización y Análisis:** Dentro de un canal, puedes utilizar las herramientas de ThingSpeak para visualizar y analizar los datos almacenados. Esto incluye la creación de gráficos en tiempo real, la realización de análisis estadísticos y el empleo de MATLAB para operaciones de procesamiento de datos más complejas.
- **Compartir y Privacidad:** Los canales pueden configurarse como públicos o privados. Un canal público permite que cualquier persona vea los datos, lo cual es útil para proyectos de código abierto o educativos. Un canal privado restringe el acceso a usuarios autorizados, para datos sensibles o privados, en la **Figura 47**, se evidencia la creación de un canal.

Figura 47

Creación de Canal en ThingSpeak

The screenshot shows the 'Create Channel' form on the ThingSpeak website. The form is titled 'How are you planning to use ThingSpeak?' and includes the following fields and options:

- How are you planning to use ThingSpeak?**
 - Commercial work (including research)
 - Government work (including research)
 - Personal, non-commercial projects
 - Student use, Teaching, or Research in academia
- What is the name of your University?***
- What best describes your current role?***
 - Student
 - Professor
 - Researcher
- What is the name of your Course or Project?***
- Tell us something about your project (optional)**

An 'OK' button is visible at the bottom right of the form.

Fuente: Autoría

- **Definición de los Campos**

Se deberá completar la información necesaria como lo es el nombre del canal, la descripción y los campos para datos, cada canal puede tener hasta 8 campos para datos, esto se evidencia en la **Figura 48**. Durante la creación del canal, se define cuántos campos se necesita y qué representará cada uno.

Figura 48

Definición de los campos para datos en ThingSpeak

The screenshot shows the 'New Channel' page in the ThingSpeak interface. The 'Name' field is 'NodoGateway'. The 'Description' is 'visualización de los datos sobre el nivel del agua potable por medio de los nodos sensores haciendo uso de la tecnología LoRa'. There are three data fields: 'Field 1' with label 'Nodo 1', 'Field 2' with label 'Field Label 2', and 'Field 3' which is currently empty. The checkboxes for Field 1 and Field 2 are checked, while Field 3's is not. To the right, a 'Help' section titled 'Channel Settings' provides instructions: 'Percentage complete' is calculated from channel data; 'Channel Name' must be unique; 'Description' is for channel identification; and 'Field#' must be checked and named, with up to 8 fields allowed.

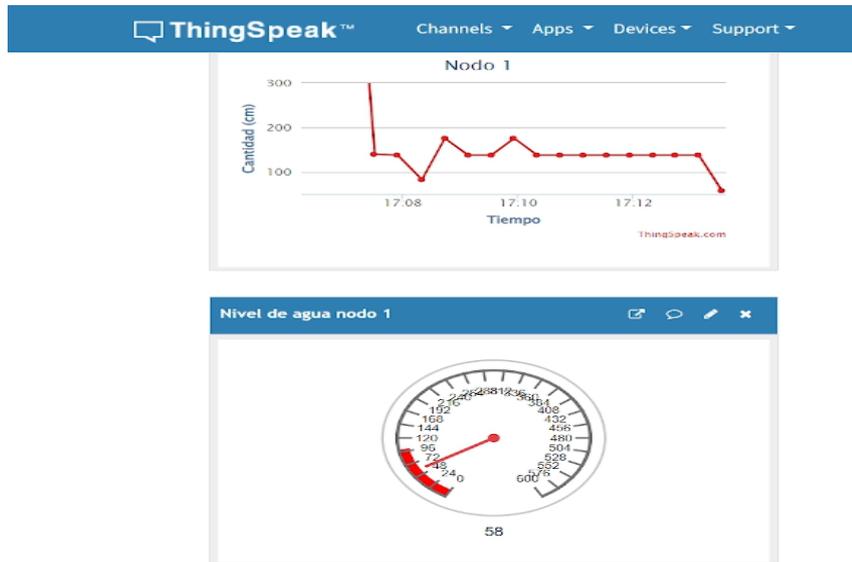
Fuente: Autoría

- **Configuración de Metadatos**

Puedes añadir metadatos al canal, como una descripción, la ubicación (si es relevante) y otros detalles que ayuden a identificar y entender el propósito del canal como se muestra la **Figura 49**.

Figura 49

Configuración de metadatos en ThingSpeak



Fuente: Autoría

- **Obtención de Claves API**

ThingSpeak proporciona claves API para cada canal, que son necesarias para enviar y acceder a los datos. La clave API de escritura que se muestra en la **Figura 50** permite enviar datos al canal, mientras que la clave API de lectura permite recuperar los datos.

Figura 50

Obtención de clave API de escritura para envío de datos de la Raspberry Pi 4 a plataforma de ThingSpeak

The screenshot shows the 'API Keys' tab in the ThingSpeak interface for a channel named 'NodoGateway'. The channel ID is 2583487, the author is mwa0000022632540, and the access is public. The page title is 'Visualización de los datos sobre el nivel del agua potable por medio de los nodos sensores haciendo uso de la tecnología LoRa.' The navigation menu includes 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys' (highlighted with a red box), and 'Data Import / Export'.

Under the 'Write API Key' section, there is a text input field containing the key 'BC9W5KTRBRRUJTJ' and a yellow button labeled 'Generate New Write API Key'.

Under the 'Read API Keys' section, there is a text input field containing the key 'FGRBNBZW1QNA2EVK' and a 'Note' text area.

On the right side, there is a 'Help' section with the text: 'API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.' Below this is the 'API Keys Settings' section with three bullet points:

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

At the bottom right, there is an 'API Requests' section with a sub-section 'Write a Channel Feed' containing a red-bordered input field with the value '65'.

Fuente: Autoría

3.3.5.5 Configuración de Página Web

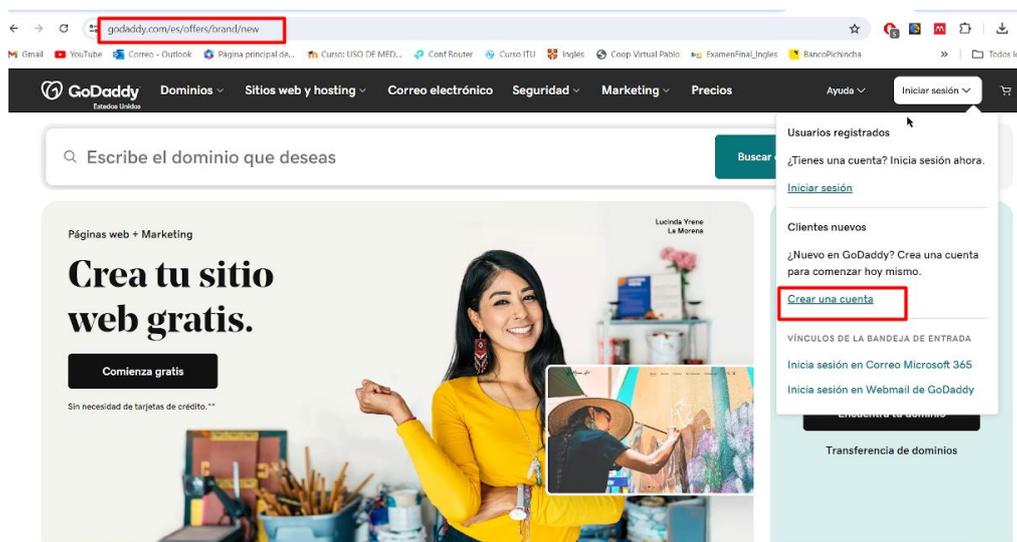
La implementación de una página web para la visualización de datos procedentes de sensores de medición en un desarenador de agua potable ofrece una herramienta importante para la gestión y el mantenimiento de estas instalaciones. Esta plataforma en línea permite el acceso en tiempo real a información detallada sobre los niveles de agua, esto para detectar y responder a anomalías o descensos en los niveles del desarenador. Con gráficos y alertas visuales, la página web permite visualizar al operario cualquier cambio significativo, permitiendo una rápida intervención para investigar y resolver problemas, asegurando así la continuidad y eficiencia del tratamiento del agua potable. Este sistema centralizado no solo mejora la operatividad del desarenador, sino que también facilita una gestión proactiva y una respuesta eficiente ante potenciales incidencias.

- **Creación de página web**

Para crear una página web gratuita en GoDaddy, primero se debe crear una cuenta en la plataforma. Accediendo a la página web de GoDaddy como se evidencia en la **Figura 51**, el usuario se registra proporcionando información básica como nombre, correo electrónico y una contraseña segura. Una vez registrada la cuenta, se selecciona el producto Website Builder de GoDaddy, optando por la versión gratuita del servicio que permite crear una página básica sin costo.

Figura 51

Registro en GoDaddy para la creación de página web

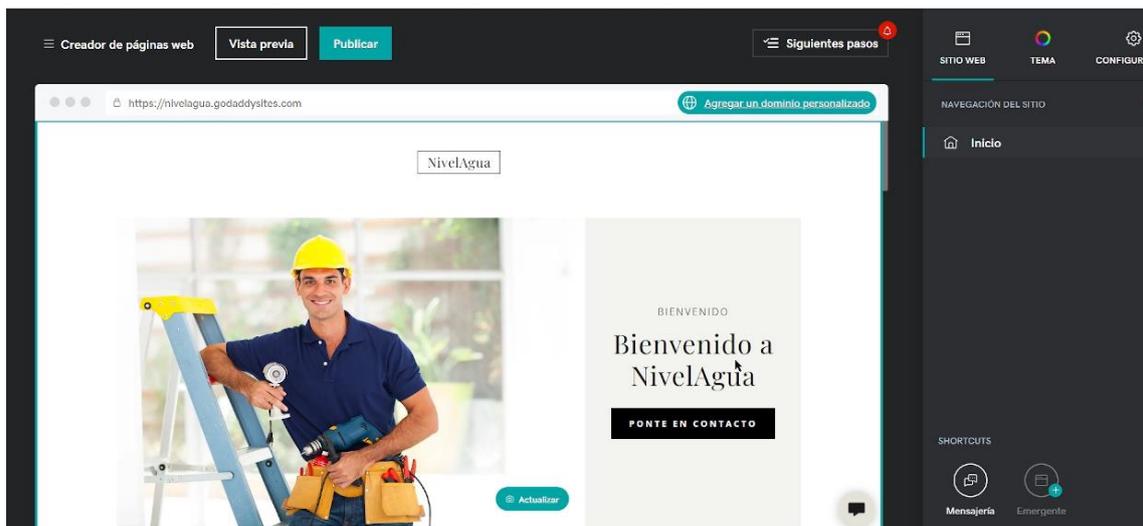


Fuente: Autoría

El siguiente paso es la selección de un tema adecuado para el sitio web. GoDaddy ofrece una variedad de temas prediseñados que el usuario puede elegir de acuerdo con la naturaleza de su negocio o proyecto personal. Una vez seleccionado el tema, el usuario procede a personalizar su sitio web como se muestra en la **Figura 52**. Utilizando el editor de arrastrar y soltar que GoDaddy proporciona, se pueden añadir texto, imágenes y otros elementos, así como personalizar colores y fuentes para adaptar el sitio a las preferencias personales o de marca.

Figura 52

Edición de página web de acuerdo con requerimientos

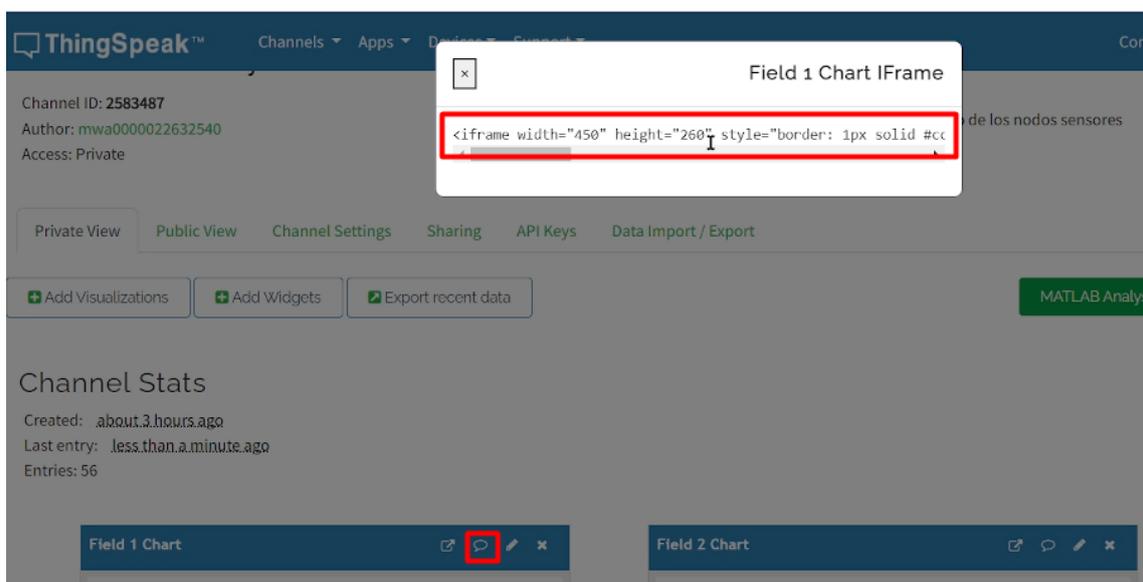


Fuente: Autoría

Para el envío de las imágenes en tiempo real desde la plataforma de ThingSpeak a la página web se debe copiar la etiqueta HTML como se aprecia en la **Figura 53** que ofrece cada imagen de metadato presente en ThingSpeak.

Figura 53

Obtención de etiqueta HTML de plataforma ThingSpeak



Fuente: Autoría

Seguidamente se debe pegar este código en la sección de código personalizado de la página y finalmente se debe guardar los cambios y publicar pudiéndose apreciar las gráficas de nivel de agua de los nodos sensores como se aprecia en la **Figura 54**.

Figura 54

Inserción de código HTML en página WEB para visualización de imágenes

The screenshot shows a web editor interface. On the left, a line graph titled 'Nodo 1' displays 'Cantidad (mm)' on the y-axis (ranging from 0 to 300) and 'Tiempo' on the x-axis (ranging from 17:10 to 17:25). The graph shows a fluctuating line that starts around 100mm, peaks near 200mm, and then stabilizes around 100mm. Below the graph, there is a section titled 'LA EMPRESA' with an image of a hand holding a yellow tool. On the right side of the editor, there is a 'Código personalizado' (Custom Code) section with a text area containing the following HTML code:

```

<div style="border: 1px solid #23d620; padding: 5px; width: fit-content; margin: 0 auto; text-align: center;">
  <img alt="Gráfico de nivel de agua de un nodo sensor" data-bbox="173 356 391 428" style="width: 100%; height: auto; border: 1px solid #23d620; background-color: #f9f9f9; margin: 0 auto;"/>
</div>

```

The code is highlighted with a red box in the image. The editor interface also shows buttons for 'Vista previa', 'Publicar', and 'Sigüientes pasos'.

Fuente: Autoría

Mediante el link que se proporciona en la parte superior de la edición se tendrá acceso a toda la información que se haya incorporado en la página como se evidencia en la **Figura 55**.

Figura 55

Publicación y Visualización de página Web



Fuente: Autoría

3.3.5.6 Configuración de Notificaciones

Ya que ThingSpeak o cuenta con la funcionalidad de generar alertas mediante el envío de correos electrónicos o de otro tipo se ha hecho uso de Mailgun el cual está especializado en la gestión y automatización de envíos de correos electrónicos. Ya que el sistema necesita una solución robusta y escalable para enviar notificaciones por correo electrónico. Mailgun proporciona una API poderosa que facilita la integración con otros sistemas y aplicaciones. Si el sistema que estás diseñando necesita integrarse con otros servicios o manejar datos de una manera que ThingSpeak no ofrece.

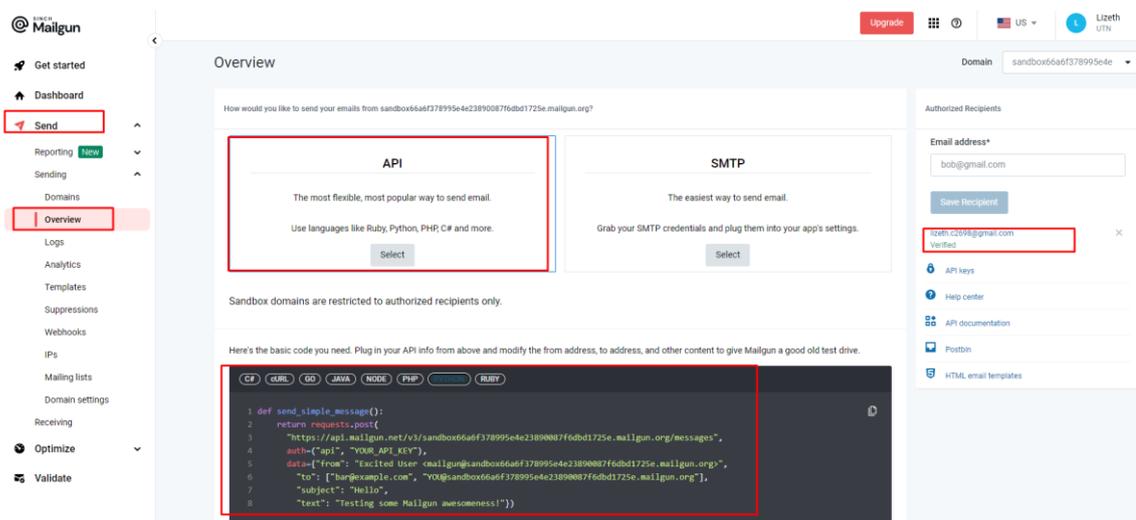
Para el uso y configuración de mailgun se debe seguir los siguientes pasos:

- Crear la cuenta haciendo uso de un correo personal en el sitio oficial de mailgun, en este caso se hará uso de un correo de gmail.
- Como se puede observar en la **Figura 56**, en el menú de la izquierda se puede apreciar el apartado de envíos, se deberá ingresar en esta parte y dirigirse a

sending, en donde se configurará el correo que va a recibir las notificaciones como el código PHP de la API para incorporarlo en la Raspberry Pi y hacer las respectivas modificaciones de códigos para el establecimiento de Umbrales.

Figura 56

Dashboard de Sitio Web de Mailgun



Fuente: Autoría

El código del que se hace uso ya modificado en la Raspberry Pi puede ser apreciado en la **Figura 57**, este código es un script en Python diseñado para ejecutarse en una Raspberry Pi y utiliza la API de Mailgun para enviar correos electrónicos automáticamente. En la sección de datos de cuenta, se especifican la clave API, el dominio y la dirección de correo electrónico desde la cual se envían los mensajes. Los detalles del correo incluyen el remitente, el destinatario, el asunto y el cuerpo del mensaje. El script está configurado para ejecutarse continuamente dentro de un bucle `while True`, y utiliza manejo de excepciones para capturar y manejar errores durante su ejecución, intentando reanudar la operación después de un breve retardo en caso de fallos.

Figura 57

Código de configuración para envío de correos desde la Raspberry al cliente

```
int.py x  mailn.py x  fin.py x
125  except serial.SerialException:
126      pass
127  except Exception:
128      pass
129  finally:
130      if 'ser' in locals() and ser.is_open:
131          ser.close()
132
133  # Datos de la cuenta de Mailgun y del correo
134  API_KEY = 'b9981370e34c950f65da3b62ea5eb036-0f1db83d-2e4080c9'
135  DOMAIN = 'sandbox66a6f378995e4e23890087f6dbd1725e.mailgun.org'
136  FROM_EMAIL = 'gladyscardenas172@gmail.com'
137  TO_EMAIL = 'lizeth.c2698@gmail.com'
138  SUBJECT = 'Notificacion nivel de agua de tanques en CHAMIZO'
139  TEXT = 'Este es el cuerpo del correo electrónico'
140
141  if __name__ == "__main__":
142      while True:
143          try:
144              main()
145          except Exception as e:
146              print(f"El script ha fallado con un error: {e}")
147              time.sleep(5) # Espera antes de reiniciar el script
148
```

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DE FUNCIONAMIENTO

Antes de adentrarse en los detalles técnicos y evaluativos del sistema de medición de nivel de agua potable, es importante asegurarse de comprender bien el proceso que se describe en este capítulo. Se cuenta con un proceso que va desde la verificación de la configuración de los elementos de red hasta el análisis del rendimiento y el costo del sistema. Cada paso ha sido diseñado para garantizar que el sistema no solo cumple con los requisitos técnicos, sino que también mejora la eficiencia operativa en comparación con las prácticas actuales. Las pruebas del sistema incluyen desde la simulación inicial de toma de datos hasta la implementación y toma de datos reales, asegurando así la fiabilidad y precisión del sistema ante condiciones diversas. Además, se aborda la integración de la plataforma tecnológica, que permite la visualización de datos y el envío de notificaciones.

4.1. Verificaciones previo a la implementación

Un paso antes de pasar a la fase de implementación es una verificación previa del funcionamiento de los componentes, este paso es importante para asegurarse de que todos los componentes estén funcionando adecuadamente. Se presta especial atención a la configuración de los módulos LoRa en cada nodo, ya que esto es clave para asegurar que la transmisión y recepción de los datos sea efectiva. Además, se verifica que cada componente esté cumpliendo correctamente con la función asignada dentro del sistema.

4.1.1 Verificación del Nodo Sensor

La verificación del nodo sensor, tanto en hardware como en software, es clave para garantizar la precisión y fiabilidad del sistema de monitoreo. En el hardware, se debe asegurar que todas las conexiones sean correctas y que los componentes funcionen

correctamente para evitar fallos en la recolección de datos. En el software, es importante revisar el código y las configuraciones para confirmar que el nodo opere según los parámetros establecidos, asegurando una buena transmisión de datos.

4.1.1.1 Hardware

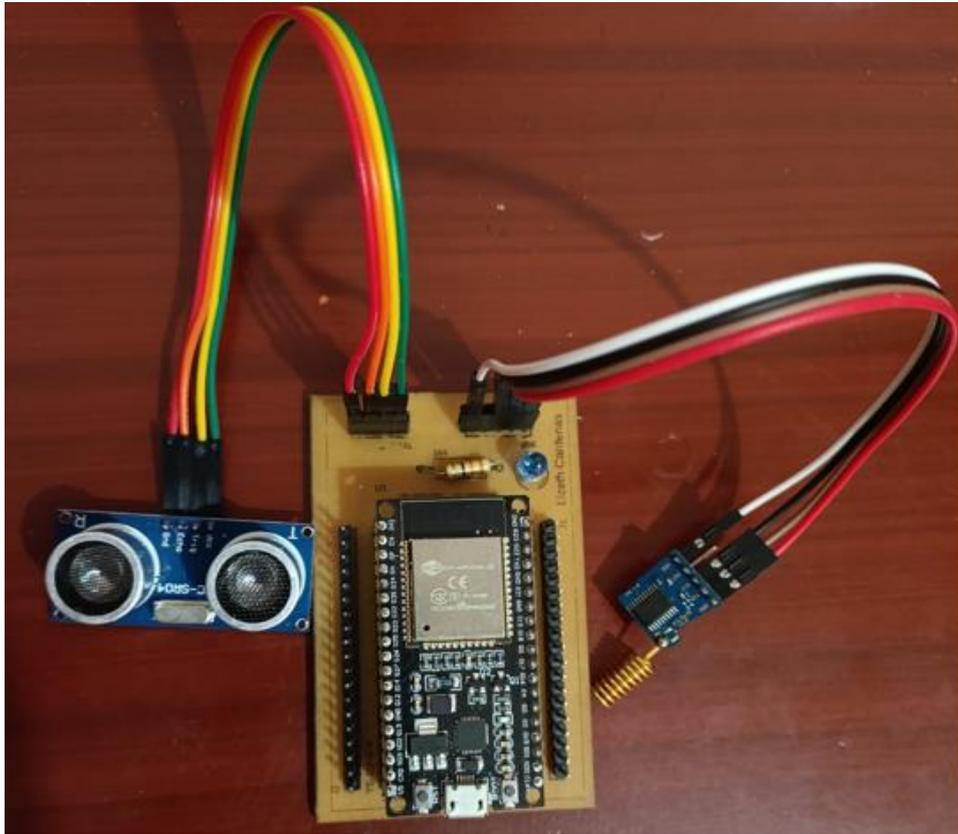
Una vez seleccionado el hardware adecuado para cada componente de la red, se procede a realizar las conexiones físicas de los equipos, siguiendo las configuraciones establecidas en el capítulo anterior.

Para el nodo sensor, se conectan los dispositivos periféricos a la placa ESP32. En la **Figura 58**, se muestra la implementación del nodo sensor, que incluye un sensor ultrasónico HC-SR04 conectado a la placa ESP32, un módulo LoRa RYLR998 conectado al ESP32 y alimentado por un banco de energía con cargador solar, con una entrada de 5V para garantizar el funcionamiento autónomo del nodo.

Se debe verificar que no haya errores en las conexiones de todos los componentes del nodo sensor para asegurar su correcto funcionamiento. Esta verificación es importante, ya que cualquier conexión defectuosa podría comprometer la precisión de las mediciones y la transmisión de datos.

Figura 58

Conexiones de Hardware de nodo Sensor

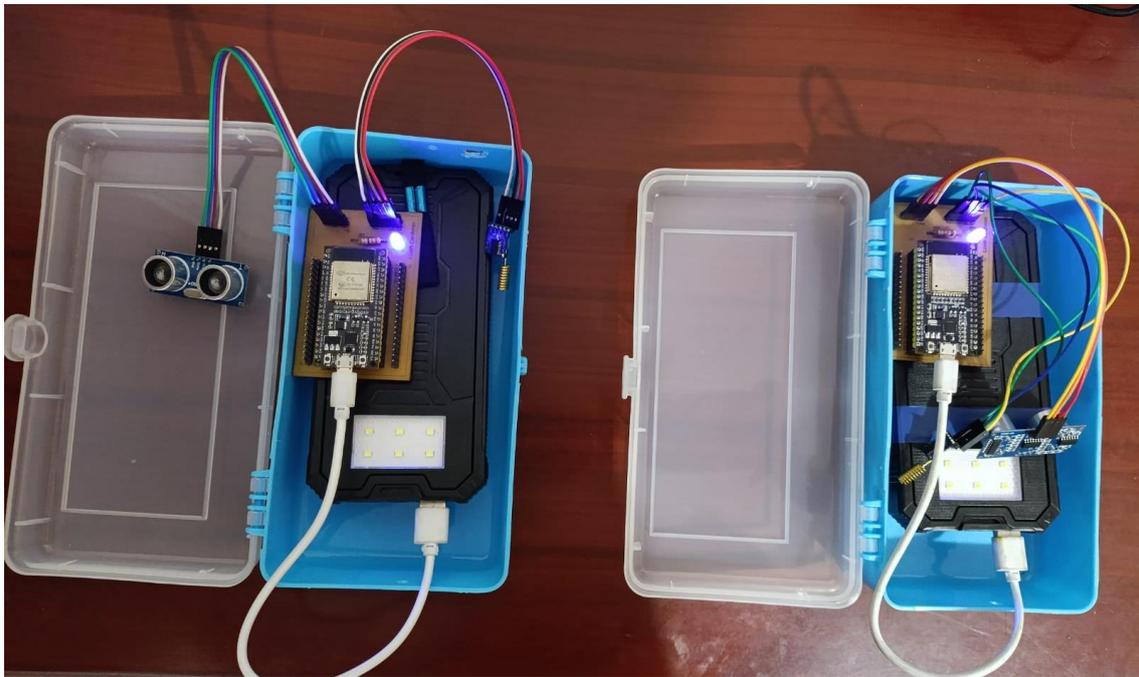


Fuente: Autoría

En cuanto a los nodos, se dispone de dos unidades del mismo tipo, ubicadas en diferentes posiciones del desarenador. A pesar de estar en posiciones distintas, su estructura y funcionamiento son idénticos. Para estos nodos, se ha seleccionado un tipo de carcasa que se adapta a las necesidades tanto del entorno como del sistema. En la **Figura 59** se muestran las carcasas utilizadas para los nodos.

Figura 59

Carcasas de los nodos para exteriores



Fuente: Autoría

El material de las carcasas utilizadas es de plástico resistente, están diseñadas para adaptarse al entorno en el que serán instaladas. Durante el proceso de instalación, todas las aperturas de la caja se sellan herméticamente para garantizar su protección contra factores externos como polvo, humedad o cambios climáticos. Además, los cables se recubren con materiales como cinta aislante y tubo termorretráctil, lo cual previene posibles daños en los componentes del sistema.

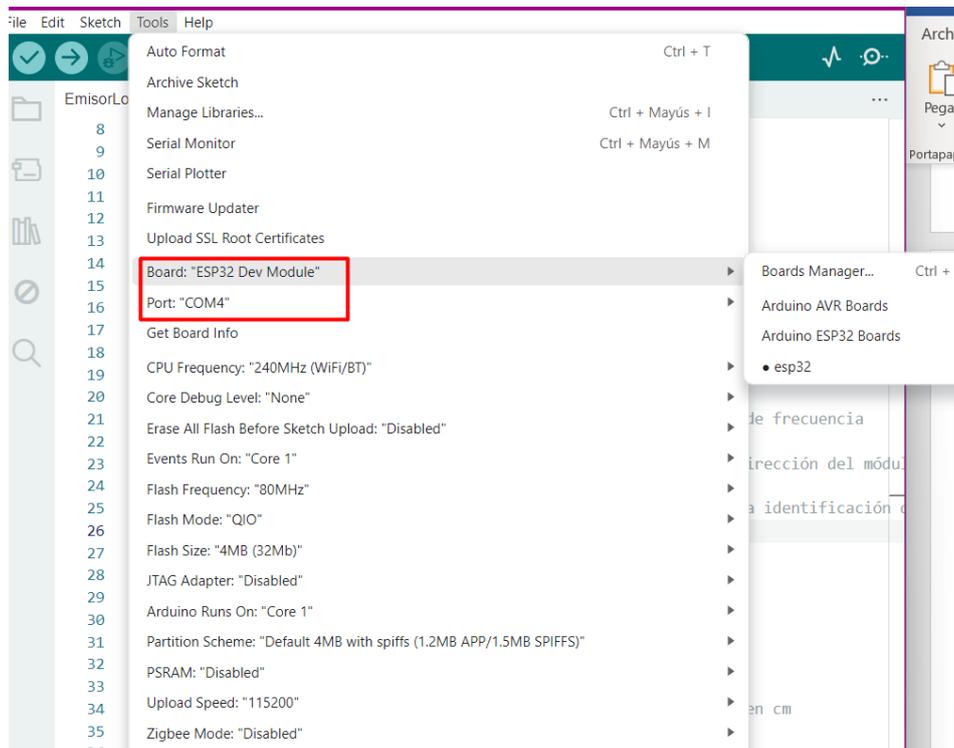
4.1.1.2 Software

Para configurar el nodo sensor, se carga el programa desde Arduino IDE a la placa ESP32. Se selecciona la opción "Dev Module", a la cual están conectados el módulo LoRa RYLR998 y el sensor ultrasónico HC-SR04. La configuración se lleva a cabo a través de la PC, utilizando el puerto COM asignado, que en este caso es el COM4, como se muestra

en la imagen. Se elige el tipo de placa, el puerto correspondiente y la velocidad de carga, esto se puede evidenciar en la **Figura 60**.

Figura 60

Parámetros de configuración en nodo Emisor



Fuente: Autoría

En la **Figura 61** se muestran las primeras cuatro líneas del código establecen las variables para configurar la comunicación entre módulos LoRa en un proyecto de red de sensores. Cada variable representa un aspecto de la configuración necesaria para asegurar una comunicación efectiva.

En esta parte del sistema, se envían instrucciones al módulo utilizando comandos AT a través de un puerto de comunicación llamado Serial2. Al configurar estos parámetros mediante comandos AT, se inicializa el módulo LoRa con los valores necesarios para que opere correctamente dentro de la red configurada. Este paso asegura que la comunicación entre los módulos sea coherente con las especificaciones

establecidas, esencial para la fiabilidad y efectividad del sistema de comunicación en el proyecto.

Figura 61

Comandos para el establecimiento de conexión LoRa

```

EmisorLora.ino
1 String lora_band = "915000000"; // Banda de frecuencia (Hz) dependiendo del país/re
2 String lora_networkid = "18"; // Identificación de la red LoRa
3 String lora_address = "1"; // Dirección del módulo LoRa
4 String lora_RX_address = "2"; // Dirección del módulo receptor LoRa
5
6 int led = 22; // LED conectado al pin D22 del ESP32
7 int trig = 12; //Transmisor
8 int eco = 13; //Receptor
9 int duracion; //Variable para duracion del pulso
10 int distancia; //Variable para hallar la distancia
11 int tanque;
12
13 void setup() {
14   pinMode(trig, OUTPUT);
15   pinMode(eco, INPUT);
16   Serial2.begin(115200, SERIAL_8N1, 16, 17); // Puerto serie 2 (pin 16 RXD, 17 TXD
17   pinMode(led, OUTPUT); // Configura el pin del LED como salida
18   delay(1500);
19
20   // Configuración inicial del módulo LoRa usando comandos AT
21   Serial2.println("AT+BAND=" + lora_band); // Configura la banda de frecuencia
22   delay(500);
23   Serial2.println("AT+ADDRESS=" + lora_address); // Configura la dirección del mód
24   delay(500);
25   Serial2.println("AT+NETWORKID=" + lora_networkid); // Configura la identificación
26   delay(1500);
27 }

```

Fuente: Autoría

4.1.2 Verificación del Gateway

La verificación del funcionamiento del hardware y software del Gateway asegura la estabilidad y eficacia de cualquier sistema de comunicación basado en red. Esta verificación garantiza que todos los componentes estén funcionando correctamente y que el software esté libre de errores. Asimismo, permite identificar y resolver problemas potenciales antes de que afecten el rendimiento del sistema o causen interrupciones en la red. Una cuidadosa revisión y prueba de la estación Gateway mejora la interoperabilidad dentro de la red, optimiza la recolección de datos y asegura que la información procesada y enviada a plataformas como ThingSpeak sea fiable y oportuna.

4.1.2.1 Hardware

Para el Gateway LoRa, se implementó una comunicación serial entre el ESP32 y la Raspberry Pi 4. Esta configuración permite que los datos capturados por el módulo RYLR998 en el ESP32 sean transmitidos y procesados en la Raspberry Pi 4, la conexión física puede ser evidenciada en la **Figura 62**. Posteriormente, estos datos son enviados a la plataforma ThingSpeak para su análisis y visualización. Adicionalmente, el Gateway se ha alojado en una carcasa que se encuentra diseñada para exteriores protegiendo la estación de daños físicos.

Figura 62

Conexión del Hardware del nodo Gateway



Fuente: Autoría

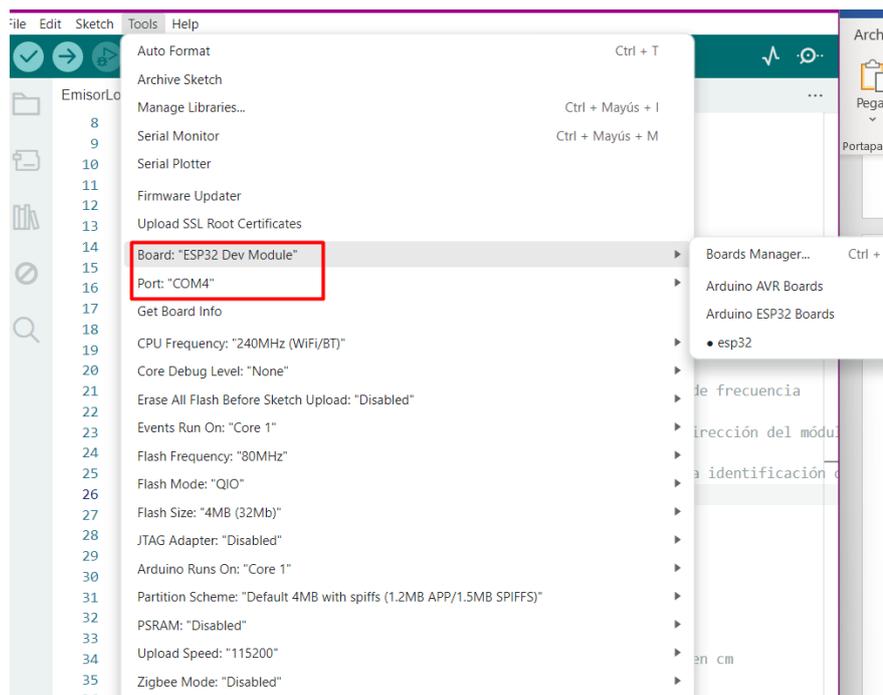
4.1.2.2 Software

Para verificar la inicialización del módulo LoRa en el Gateway, se selecciona la placa y el puerto COM adecuados, en este caso, el COM5, tal como se muestra en la configuración para la placa ESP32 que forma parte del Gateway LoRa. Además, se

configura otros parámetros que deben coincidir exactamente con los establecidos en la programación del nodo sensor, los parámetros se pueden evidenciar en la **Figura 63**. Así se garantiza una comunicación coherente y efectiva entre el nodo sensor y el Gateway.

Figura 63

Selección de parámetros del ESP32 del receptor Gateway



Fuente: Autoría

Para el correcto funcionamiento del módulo LoRa en Ecuador, se ha establecido la banda de frecuencia adecuada de acuerdo con las regulaciones del país. Además, se ha identificado y configurado la red LoRa, asignando una dirección específica al módulo, siguiendo el mismo procedimiento utilizado en el nodo emisor. Cada uno de estos parámetros se configuran para garantizar su operatividad dentro del sistema de comunicación LoRa. La **Figura 64** muestra el código que evidencia la configuración de estos parámetros.

Figura 64

Configuración de parámetros para el funcionamiento de LoRa

```

ReceptorLora | Arduino IDE 2.3.3-nightly-20240615
File Edit Sketch Tools Help
SparkFun ESP32 Thing
ReceptorLora.ino
1 String lora_band = "915000000"; // Banda de frecuencia (Hz) dependiendo del país/región
2 String lora_networkid = "18"; // Identificación de la red LoRa
3 String lora_address = "2"; // Dirección del módulo LoRa
4
5 int led = 22; // LED conectado al pin D22 del ESP32
6
7 String textoEntrada; // Almacena los datos recibidos del emisor
8
9 void setup() {
10   pinMode(led, OUTPUT); // Configura el pin del LED como salida
11   Serial2.begin(115200, SERIAL_8N1, 16, 17); // Inicializa el puerto serie 2 (pin 16 RXD
12   Serial.begin(115200); // Inicializa el puerto serie a 115200 bps
13
14   delay(1500);
15   Serial2.println("AT+BAND=" + lora_band); // Configura la banda de frecuencia del módul
16   delay(500);
17   Serial2.println("AT+NETWORKID=" + lora_networkid); // Configura la identificación de l
18   delay(500);
19   Serial2.println("AT+ADDRESS=" + lora_address); // Configura la dirección del módulo Lo
20   delay(1000);

```

Fuente: Autoría

El sistema en general se encuentra funcionando como se muestra en la **Figura 65**, la estación Gateway y los nodos sensores se encuentran con un led que muestra visualmente el correcto funcionamiento del sistema, ya que estos se encienden cuando se encuentra enviando los datos del emisor al receptor correctamente, de igual forma en la plataforma de ThingSpeak se tiene las imágenes que evidencian gráficamente la llegada de datos de los sensores, lo que indica que la Raspberry pi está enviando los datos correctamente.

Figura 65

Sistema de Medición de nivel de agua en completo en funcionamiento

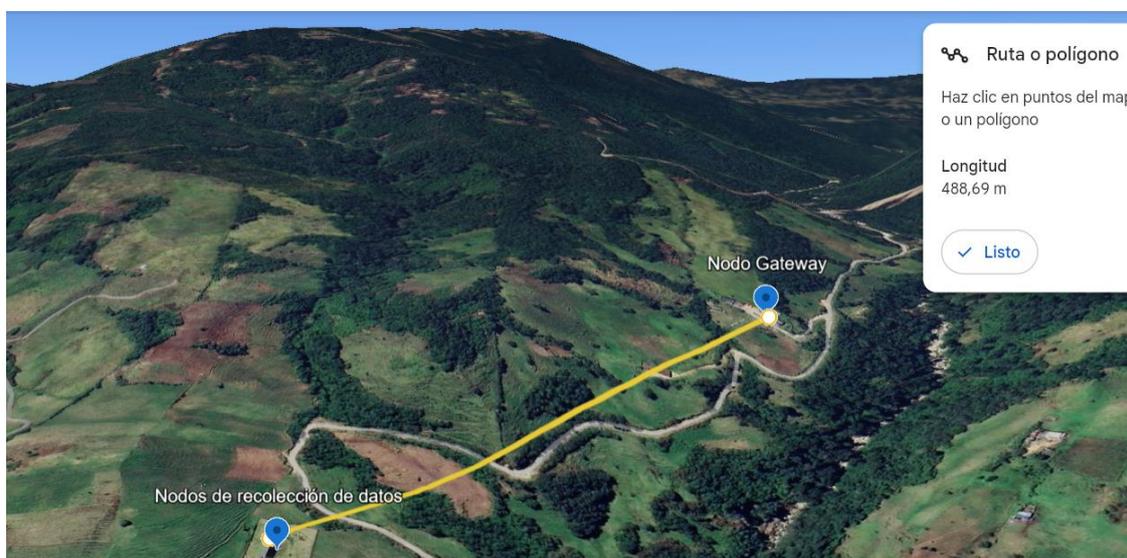


Fuente: Autoría

Como se muestra en la **Figura 66** se establecen tres nodos, dos identificados como "Nodos de Recolección de Datos" y el otro "Estación Gateway". La distancia de separación entre los nodos recolectores y el Gateway es de aproximadamente 488 metros.

Figura 66

Ruta de separación entre nodos principal y secundarios del sistema de Medición de Nivel de agua



Fuente: Autoría

La ubicación del nodo Gateway ha sido cambiada debido a pequeñas interferencias causadas por el relieve del terreno, se decidió reubicar el nodo Gateway a un lugar más cercano que se ubica a 300 metros de los nodos de recolección de datos como se evidencia en la **Figura 67**. Esta nueva ubicación no presenta inconvenientes, ya que el sistema es completamente autónomo, por lo que no se requiere interacción directa, y el operador puede verificar los datos desde cualquier ubicación. Además, el nuevo sitio es seguro y adecuado para el funcionamiento del sistema.

Figura 67

Ruta de separación actualizada entre Nodos de Recolección y Nodo Gateway del Sistema de Medición de Nivel de agua



Fuente: Autoría

4.1.3 Pruebas de Funcionamiento del Sistema

Las pruebas de funcionamiento en el sistema aseguran la integridad y eficiencia de la red de comunicaciones. Estas pruebas permiten verificar que la recolección y visualización de datos desde el nodo sensor hasta la estación Gateway operan correctamente. Es importante que el microcontrolador de la Raspberry Pi 4 maneje

adecuadamente los datos recibidos, procesándolos y enviándolos a plataformas de visualización como ThingSpeak. Estas pruebas no solo ayudan a detectar y corregir errores en las etapas tempranas, sino que también garantizan que el sistema pueda funcionar de manera continua y fiable bajo diferentes condiciones operativas, maximizando así la utilidad y el valor de los datos recogidos.

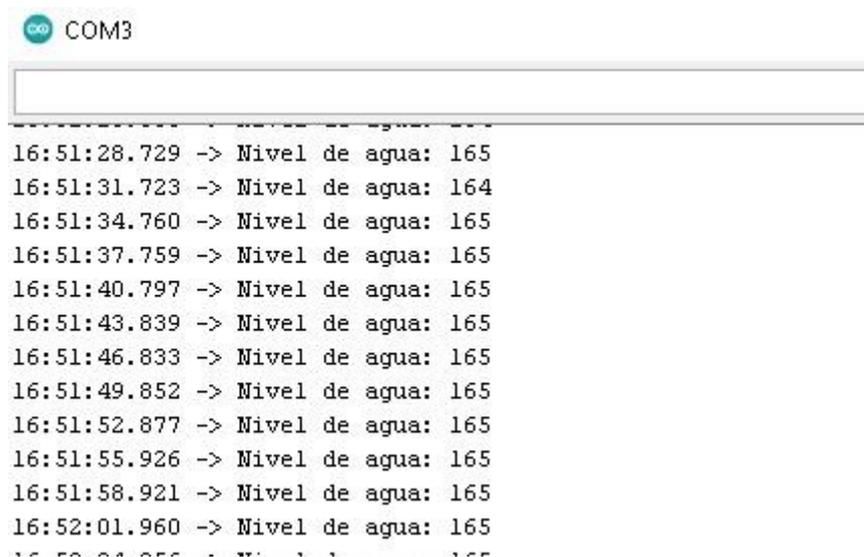
4.1.3.1 Recolección y Visualización de Datos del Nodo Sensor

La **Figura 68** muestra el monitor serial de Arduino (puerto COM3), registrando las lecturas de un sensor de nivel de agua en tiempo real. Cada línea incluye una marca de tiempo en formato hh:mm:ss y un mensaje que indica el nivel de agua detectado, con valores numéricos que oscilan entre 164 y 165. Las lecturas se realizan aproximadamente cada segundo, evidenciando un sistema de monitoreo continuo.

Una vez obtenidos los datos como mediciones precisas, están listos para ser enviados a la estación Gateway. Esto permite el procesamiento y análisis posterior en la plataforma de ThingSpeak y Mailgun.

Figura 68

Toma y visualización de datos en nodo sensor



```
COM3
16:51:28.729 -> Nivel de agua: 165
16:51:31.723 -> Nivel de agua: 164
16:51:34.760 -> Nivel de agua: 165
16:51:37.759 -> Nivel de agua: 165
16:51:40.797 -> Nivel de agua: 165
16:51:43.839 -> Nivel de agua: 165
16:51:46.833 -> Nivel de agua: 165
16:51:49.852 -> Nivel de agua: 165
16:51:52.877 -> Nivel de agua: 165
16:51:55.926 -> Nivel de agua: 165
16:51:58.921 -> Nivel de agua: 165
16:52:01.960 -> Nivel de agua: 165
```

Fuente: Autoría

4.1.3.2 Recolección y Visualización de Datos de la estación Gateway

Una vez configurado adecuadamente el nodo emisor y efectuado el envío de datos a la estación Gateway, que en este caso es el ESP32, es posible verificar la correcta recepción de los datos en el nodo receptor como lo muestra la **Figura 69**; en esta instancia, solo se requiere la llegada del valor específico que será procesado y reenviado, el ESP32 procede a transmitir los datos de manera serial al microcontrolador Raspberry Pi 4 para su procesamiento.

Figura 69

Recepción de los datos en el nodo Gateway ESP32



```

21   delay(500),
22   Serial2.println("AT+NETWORKID=" + lora_networkid); // Configura la identificac
23   delay(500);
24   Serial2.println("AT+ADDRESS=" + lora_address); // Configura la dirección del m
25   delay(1000);
26
27   Serial.println("Iniciación completa");
28 }
29
30

```

Output Serial Monitor x

Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM4')

N1: 13, N2: 33
N1: 12, N2: 36
N1: 12, N2: 33
N1: 13, N2: 36
N1: 12, N2: 33
N1: 12, N2: 36
N1: 12, N2: 36
N1: 13, N2: 33
N1: 13, N2: 33
N1: 13, N2: 36
N1: 13, N2: 33
N1: 12, N2: 33
N1: 13, N2: 33
N1: 12, N2: 33

Fuente: Autoría

Para la recepción de datos desde el ESP32 hacia la Raspberry Pi 4 se hace uso de un script en Python mediante el script de Gary. Gary es una herramienta o script personalizado destinado a procesar los datos recibidos. En la **Figura 70**, se puede observar que los datos son recibidos de manera consistente cada 15 segundos, intervalo

4.2.1 Definición de parámetros de pruebas

La definición precisa de los parámetros de prueba es necesaria para asegurar el correcto funcionamiento y la eficacia del sistema de monitoreo de nivel de agua. Estos parámetros actúan como las bases sobre las cuales se establecerá la toma de datos reales y simulados. Entre los más importantes se encuentran el rango de medición, el intervalo de tiempo entre mediciones y los umbrales de alarma.

4.2.1.1. Rango de medición

El alcance del sistema se compara entre los valores teóricos proporcionados por la tecnología LoRa y los resultados obtenidos mediante pruebas prácticas, para garantizar que cumpla con las necesidades del sistema. El rango de medición es crítico, ya que, si se excede el alcance previsto, los datos pueden volverse intermitentes, es decir, algunos pueden ser recibidos correctamente y otros no.

- **Alcance General Teórico de LoRa**
 - a) **Entornos Urbanos:** En zonas urbanas, donde los edificios y otras estructuras pueden interferir con la señal, LoRa generalmente alcanza entre 2 y 5 kilómetros. Sin embargo, la presencia de obstáculos urbanos como edificios altos y densidad de infraestructuras puede reducir este rango.
 - b) **Entornos Rurales:** En áreas rurales, donde hay menos obstrucciones, el alcance de LoRa puede extenderse hasta 15 kilómetros o más, dependiendo de las condiciones del terreno y la línea de vista directa.

- **Mediciones Realizadas con el Sistema**

Las pruebas específicas realizadas con nuestro sistema han demostrado resultados alentadores:

- a) **Entornos Urbanos:** Se logró un alcance de 200 metros en áreas con obstáculos.

b) Entornos Rurales: En condiciones ideales, se probado un alcance de 2 kilómetros, los cuales satisfacen las necesidades del proyecto.

- **Requerimientos Específicos de Implementación**

Aunque el sistema ha demostrado tener un alcance considerablemente mayor en pruebas, para la implementación del sistema específico que se está considerando, se requiere un rango de 300 metros para asegurar una comunicación efectiva y fiable.

a) Tabla de Resultados de Pruebas

Aquí se presenta la *Tabla 19* que incluye tanto los rangos típicos de LoRa como los obtenidos en las pruebas:

Tabla 19

Rangos de distancias típicas y de pruebas del sistema de medición de Nivel de Agua

	Alcance General Teórico	Mediciones Realizadas	Requerimientos Específicos
1	Entornos Urbanos (2-5km)	Entorno Urbano (200 m)	N/A
2	Entornos Rurales (hasta 15 km)	Entorno Rural (2 km)	Estructura de área de captación y estación Gateway (300 m)

La tabla demuestra que la tecnología se adapta a las necesidades del sistema de monitoreo y notificación del nivel de agua.

4.2.1.2. Intervalo de tiempo

Para las pruebas simuladas del sistema, se ha establecido un intervalo de tiempo de 15 segundos. Este intervalo permite observar de manera eficiente el comportamiento

y la funcionalidad del sistema, facilitando una rápida verificación del procesamiento de datos en la estación Gateway. Durante este período, se han recopilado y analizado los datos que se envían y reciben a través de la plataforma ThingSpeak. Los resultados obtenidos han ofrecido información clave sobre la capacidad de respuesta del sistema y su eficiencia al manejar datos en tiempo real.

En lo que concierne a la toma de datos para el funcionamiento del sistema en condiciones reales se realizará en un lapso de 10 minutos

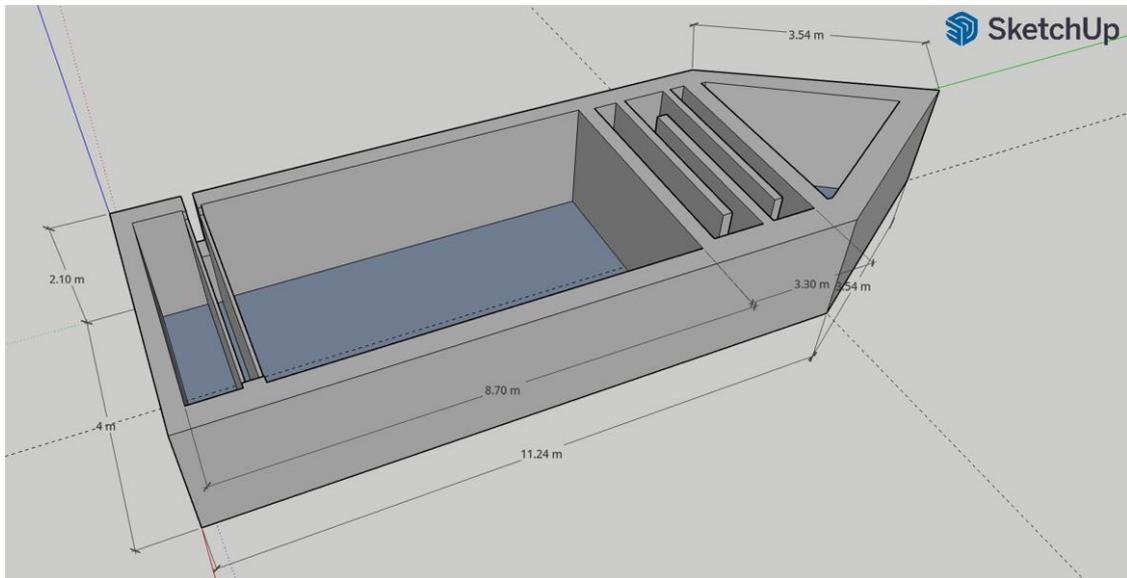
4.2.1.3. Umbrales de alarma

Establecer estos umbrales permite al sistema alertar a los operadores ante condiciones potencialmente críticas o anormales.

En el sistema desarrollado, se emplea el sensor ultrasónico HC-SR04 que brinda mediciones de distancia en centímetros. Esta medida refleja la proximidad del agua al sensor ubicado en la parte superior del desarenador. Para determinar el volumen total de agua, se realiza una operación matemática básica al inicio del código del nodo emisor. Se conoce que la profundidad total del desarenador es de 2.10 metros como se muestra en la **Figura 71**. Al restar la distancia medida por el sensor desde la superficie del agua hasta la parte superior del tanque a la profundidad total, se obtiene la altura del agua en el desarenador. Con este dato, y considerando las dimensiones totales del desarenador, se calcula el volumen total de agua. Además, se establecen umbrales críticos tanto en centímetros como en volumen de agua, basados en estos cálculos, para asegurar un monitoreo efectivo y preciso del sistema.

Figura 71

Medidas del desarenador de agua del área de Captación



Fuente: Autoría

A continuación, se establece la ecuación para calcular la altura del agua en el desarenador usando un sensor ultrasónico HC-SR04:

$$\text{Altura del Agua} = \text{Profundidad Total del Desarenador} - \text{Distancia Medida por el Sensor} \quad (12)$$

Donde:

- a) Profundidad Total del Desarenador: es la profundidad máxima del desarenador, que en este caso es de 2.10 metros.
- b) Distancia Medida por el Sensor: es la distancia desde la parte superior del desarenador hasta la superficie del agua, medida por el sensor.

Por lo tanto, la ecuación para calcular la altura del agua en el desarenador es:

$$\text{Altura del Agua} = 2.10 \text{ m} - \text{Distancia Medida por el Sensor (en metros)} \quad (13)$$

Esta ecuación proporcionará la altura del agua en metros con la cual se realizará el cálculo del volumen del agua en el desarenador haciendo uso de la siguiente ecuación:

$$\text{Volumen del agua} = \text{Largo} \times \text{Ancho} \times \text{Altura del Agua} \quad (14)$$

Donde:

- a) **Largo** es el largo del tanque (13 m).
- b) **Ancho** es el ancho del tanque (4 m).
- c) **Altura del Agua** es el nivel del agua medido por el sensor convertido de cm a metros para mantener la consistencia de las unidades.

En base a esto se tiene la **Tabla 20** de mediciones mínimas y máximas tanto en centímetros como en metros cúbicos para el nivel del agua del desarenador:

Tabla 20

Umbrales mínimos y máximos en cm y m³ del nivel de agua en desarenador

Descripción	Nivel del Agua (cm)	Nivel del Agua (m)	Volumen (m³)
Mínimo Operativo	180 cm	1.80 m	93.6 m ³
Máximo Operativo	200 cm	2.00 m	104 m ³

Esta tabla facilita la visualización y comprensión de los valores críticos del sistema, permitiéndolo monitorizar eficazmente los niveles y volúmenes de agua dentro del rango operativo del desarenador. El volumen no es un indicador recomendado ya que el operador está familiarizado con el nivel de medición en cm como el actual indicador visual.

4.2.2 Toma de datos simulados previo a implementación

Previo a la implementación, se llevó a cabo diversas pruebas que abarcan desde la verificación de la precisión del sensor hasta evaluaciones a largo plazo del rendimiento del hardware. Estas pruebas son necesarias para garantizar que el sistema cumple con los requisitos operativos y puede manejar diferentes condiciones ambientales y operacionales. A continuación, se describen las pruebas realizadas y los resultados obtenidos.

4.2.2.1 Pruebas de Precisión del Sensor

Para validar la precisión del sensor ultrasónico HC-SR04, se han realizado una serie de pruebas controladas para medir la distancia y evaluar la precisión:

a) Procedimiento:

- Configuración Inicial: El sensor HC-SR04 fue montado en un soporte fijo y se midió una distancia conocida de 100 cm.
- Repetición de Mediciones: Se realizaron 50 mediciones consecutivas de la distancia.
- Registro de Datos: Las mediciones fueron registradas y comparadas con la distancia conocida.

b) Resultados:

- Distancia Conocida: 100 cm
- Mediciones Registradas: 50 mediciones
- Promedio de Mediciones: 100.1 cm
- Margen de Error: ± 0.9 cm

Con estas mediciones se llegó a la conclusión de que el sensor HC-SR04 tiene una precisión de ± 1 cm, cumpliendo con los requisitos de precisión establecidos para el sistema.

4.2.2.2 Pruebas a Largo Plazo

Se realizaron pruebas para evaluar la estabilidad y robustez del sistema durante una semana.

a) Procedimiento:

- Duración de la Prueba: 7 días continuos.
- Frecuencia de Mediciones: Cada 10 minutos.
- Registro de Funcionamiento: Monitoreo continuo del rendimiento de las placas ESP32 y RP4, junto con los módulos RYLR998.

b) Resultados:

- Tiempo Total de Pruebas: 168 horas.
- Mediciones Realizadas: 1008 mediciones.
- Estado de las Placas: Sin fallos ni interrupciones.
- Tasa de Éxito de Transmisión de Datos: 99.8%.

Realizadas estas pruebas se puede verificar que las placas ESP32 y RP4, junto con los módulos RYLR998, demostraron un rendimiento estable y robusto durante el período de prueba de una semana completa.

4.2.2.3 Toma de datos Simulada

A continuación, se presenta la **Tabla 21** de mediciones realizadas cada 15 segundos, simulando variaciones en el nivel del agua en el desarenador.

Tabla 21*Mediciones del nivel de agua en un entorno simulado*

Tiempo(s)	Nivel de Agua (cm)
0	190
15	190
30	190
45	190
60	188
75	185
90	185
105	180
120	177
135	177
150	177
165	180
180	184
195	184
210	196
225	203
240	203
255	203

La tabla muestra un patrón de variación del nivel del agua, comenzando en 190 cm, disminuyendo gradualmente a 177 cm y luego aumentando nuevamente a 203 cm. Estas mediciones simulan el comportamiento esperado del sistema en diferentes

condiciones operativas, validando su capacidad para medir y transmitir datos de manera precisa y confiable, las variaciones del agua se han manipulado gradualmente, comprobando que el sistema funciona de forma eficiente además de dar respuesta automática ante variaciones y sin presentar falsos positivos.

4.2.3 Implementación de nodos del Sistema

Para la implementación del sistema, se seleccionan los lugares más adecuados para instalar las bases de los nodos de recolección de datos del nivel de agua, asegurando la precisión y el buen funcionamiento del sistema. El primer nodo se ubicará en la parte inferior del desarenador, donde previamente se realizaba la inspección manual del nivel de agua. Esta ubicación ha sido elegida por su accesibilidad y buen relieve, permitiendo un monitoreo continuo y preciso de los cambios en el nivel del agua.

El segundo nodo se instalará en la parte superior del desarenador (fosa triangular), área por donde entra el agua captada de la quebrada del Oso. Esto se puede apreciar en la

Figura 72.

Figura 72

Ubicación de Nodos de Recolección de Datos en Desarenador del área de Captación

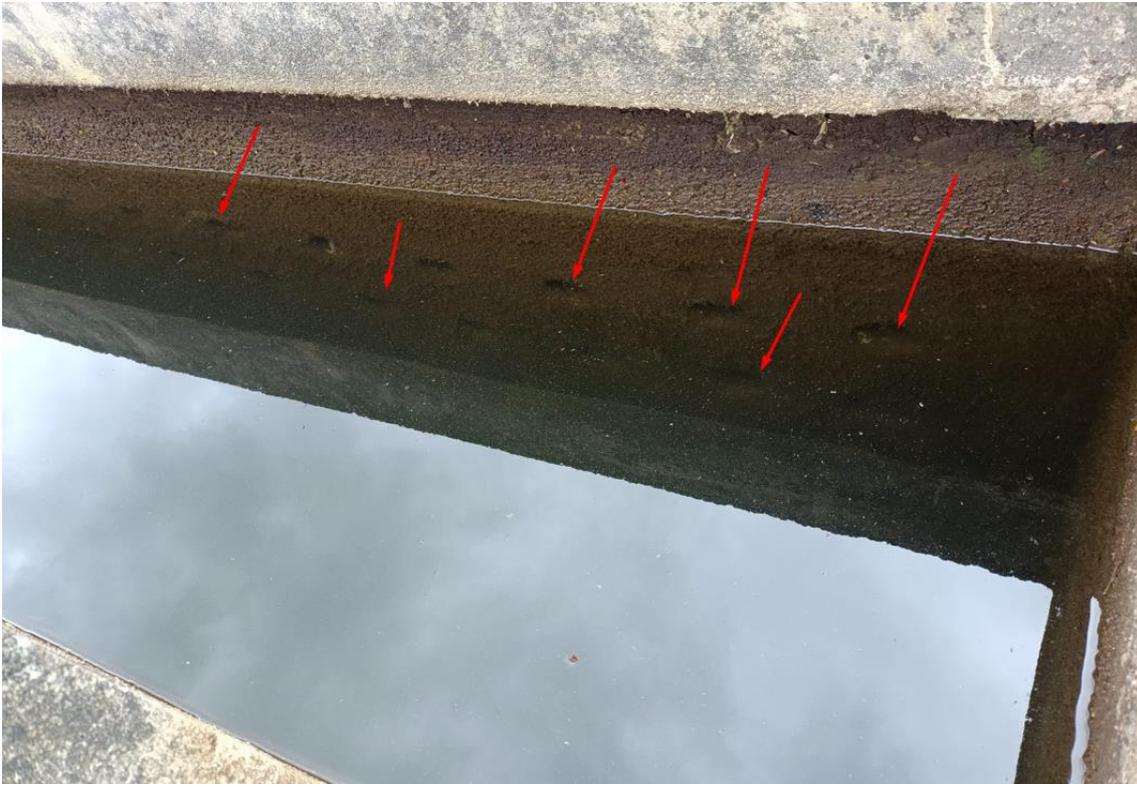


Fuente: Autoría

La ubicación del segundo nodo se debe a que esta estructura triangular constituye la principal entrada de agua, la cual suele llegar con diversas impurezas. En ocasiones, el flujo de agua se ve afectado por la acumulación de estas impurezas o por la formación de nata en la superficie, lo que provoca que el nivel de agua disminuya en la segunda fase del desarenador. El paso del agua hacia la siguiente estructura se realiza a través de pequeños orificios ubicados en la pared que separa las fosas, lo cual se puede observar claramente en la **Figura 73**.

Figura 73

Orificios de separación entre estructura 1 y estructura 2 del Desarenador

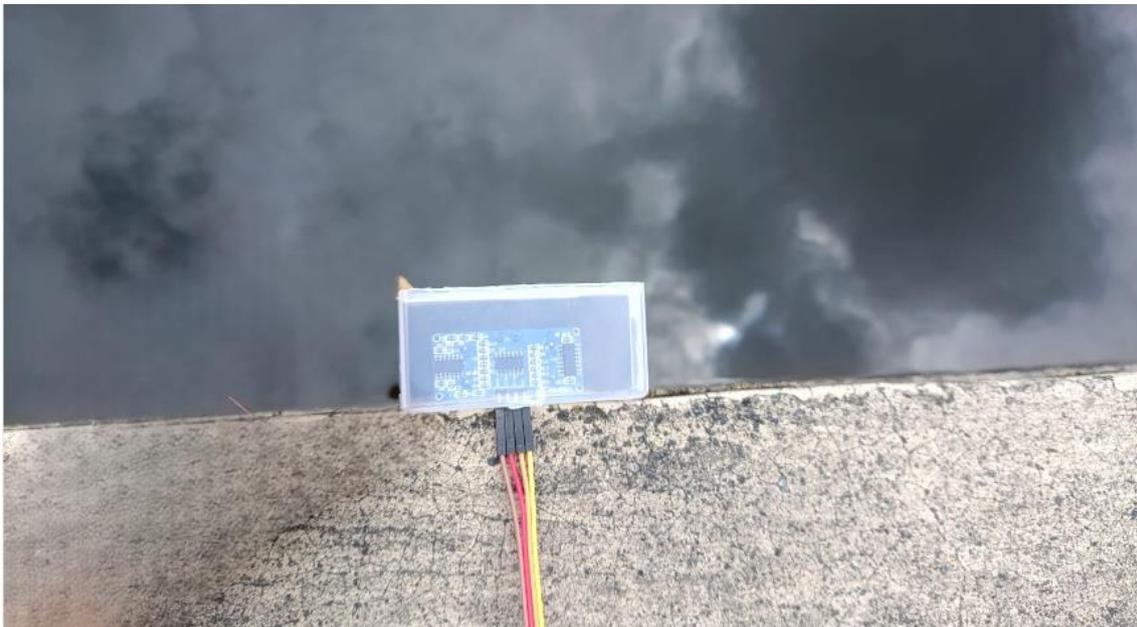


Fuente: Autoría

Los nodos sensores se ubican dentro de una carcasa protectora, ya que se encuentran en un ambiente exterior. Dado que los nodos se colocarán en una posición elevada para facilitar la transmisión a través de los módulos LoRa, los sensores se extienden mediante cables UTP hasta alcanzar el nivel de la superficie de la estructura del desarenador. Para protegerlos de las condiciones ambientales, cada sensor contará con una pequeña carcasa en su parte superior, evitando así que se vean afectados por las lluvias o la humedad. La **Figura 74** muestra la forma en que serán ubicados cada uno de los sensores.

Figura 74

Ubicación del sensor en superficie de estructura del Desarenador



Fuente: Autoría

La Estación Gateway se encuentra ubicada a una distancia de 300 metros de los nodos de recolección, su ubicación puede apreciarse en la **Figura 75**. Su instalación se realiza en el lateral de una vivienda, dispone de una carcasa hermética que protege los componentes electrónicos de factores externos como polvo o humedad. Además, el Gateway cuenta con una fuente de alimentación eléctrica confiable, y la conexión a la red wifi puede realizarse mediante cable ethernet o directamente a través de señal inalámbrica como en este caso.

Figura 75*Ubicación de Nodo Gateway*

Fuente: Autoría

4.2.4 Recolección de datos

La recopilación de datos en el sistema de medición de nivel de agua es el siguiente paso, el cual es importante porque proporciona información precisa y oportuna del funcionamiento del sistema.

4.2.4.1 Frecuencia de medición

Según investigaciones similares como (Angulo & Saavedra, 2018) se ha hecho uso de una frecuencia de 15 minutos para la toma de datos, en este caso fue suficiente. Pero según (Alejandra Sánchez-Hernández et al., 2017) sugiere que la frecuencia de medición debe adaptarse a las características específicas del lugar y a los objetivos del monitoreo, pudiendo variar desde mediciones continuas hasta intervalos más amplios.

Durante las pruebas en el entorno real, se observó que la estructura del desarenador queda completamente vacía en aproximadamente 30 minutos. Para alcanzar

el nivel mínimo de operación toma alrededor 5 minutos. Por ello se ha configurado el sistema para recolectar y procesar los datos cada 10 minutos, ya que esta frecuencia es suficiente para capturar las variaciones relevantes en el nivel de agua y optimizar el manejo de la información.

4.2.4.2 Duración del muestreo

De igual forma que la frecuencia, la duración del tiempo de muestreo depende de acuerdo con los objetivos planteados en cada proyecto y no hay una recomendación que indique algún valor, pero según tesis similares como (Ramirez Fernandez et al., 2024) se toman datos durante un período mínimo de 1 mes a más para identificar y confirmar que las variaciones en los niveles de agua son efectivamente pequeñas y constantes bajo condiciones normales de operación.

Por ello, para contar con un grupo de datos más completo y robusto, se ha optado por realizar pruebas durante 3 meses. Este período es ideal, ya que coincide con la estación de sequía, una situación que afectó directamente los niveles de agua, lo que permitió recopilar información clave para entender el comportamiento del sistema en condiciones críticas. Los datos obtenidos en este período son suficientes y representativos para realizar un análisis detallado.

4.2.4.3 Estructura de Datos

Como datos representativos, se ha elaborado **Tabla 22**, resumen que presenta el promedio semanal de los niveles de agua para cada mes evaluado. Esta información facilita una visión más clara de las tendencias y variaciones registradas durante el período de muestreo.

Tabla 22*Promedios semanales de las mediciones del nivel de agua.*

	Mes	Semana	Nivel_N1 (cm)	Nivel_N2 (cm)
1	9	35	183.06	184.05
2	9	36	183.75	184.71
3	9	37	184.12	185.12
4	9	38	183.81	185.03
5	9	39	184.03	185.02
6	10	40	183.75	184.71
7	10	41	183.83	184.72
8	10	42	184.03	185.01
9	10	43	183.84	184.75
10	10	44	183.98	184.76
11	11	44	184.08	185.15
12	11	45	185.07	185.05
13	11	46	185.19	186.33
14	11	47	185.71	186.27
15	11	48	185.02	186.43

La **Tabla 23** muestra los promedios del nivel de forma mensual.

Tabla 23*Promedios mensuales de datos de nivel de agua*

	Mes	Promedio N1 (cm)	Promedio N2 (cm)
1	9	183.75	184.79
2	10	183.89	184.79
3	11	185.01	185.85

Para una revisión detallada, los datos completos se adjuntan en el **Anexo 4**.

Recopilación de datos no procesados

4.2.5 Análisis de Datos Lectura de Datos

De acuerdo con los datos de promedios semanales y mensuales obtenidos que se muestran en la **Tabla 22** y **Tabla 23**, los niveles de agua en los tanques N1 y N2 muestran una variabilidad mínima durante los meses de septiembre, octubre y noviembre. Esta estabilidad se explica en gran medida porque la entrada de agua está controlada mediante una tubería regulada por una llave que rara vez se ajusta, salvo en situaciones específicas que requieran detener o permitir la entrada de agua. Este control estricto garantiza que los niveles se mantengan dentro de los valores esperados.

Los meses de prueba coinciden con un periodo de sequía significativa en América Latina, causada por condiciones climáticas globales adversas, como las asociadas al fenómeno de El Niño (Organización Meteorológica Mundial, 2024). A pesar de estas condiciones, los niveles de agua registrados no descendieron por debajo del umbral crítico, lo que evitó la necesidad de implementar medidas extremas.

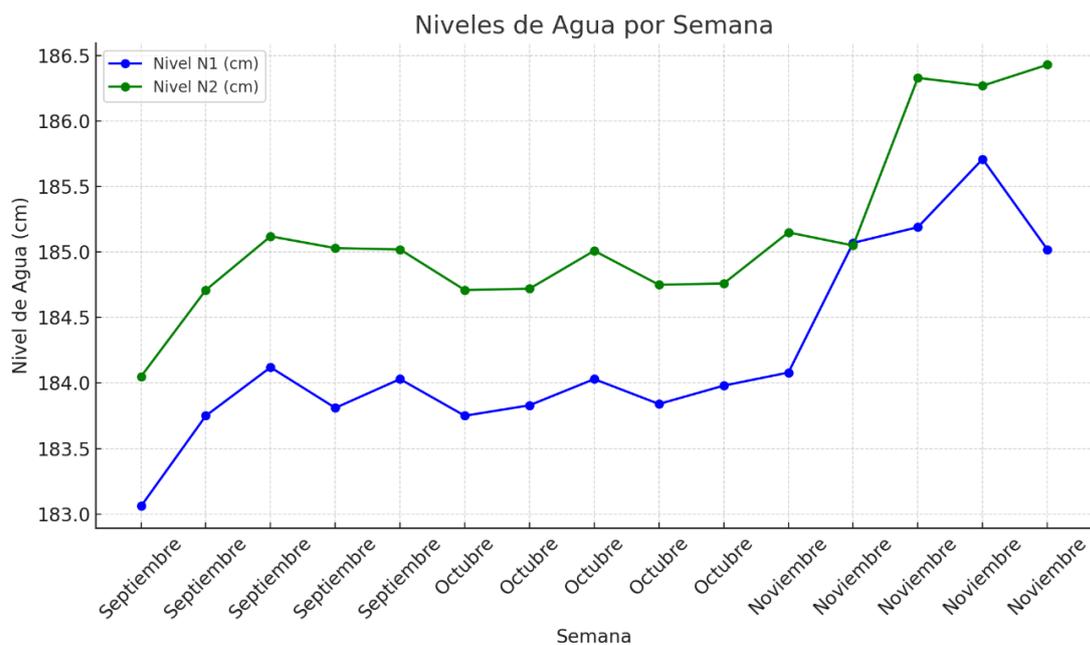
En cuanto a la tendencia mensual, se observa una ligera alza en los niveles de agua durante noviembre, especialmente en el tanque N2, donde el promedio aumentó a 185.85 cm. Aunque este incremento no es muy pronunciado, indica una posible mejora en la disponibilidad de agua hacia el final del periodo de muestreo, posiblemente relacionada con ajustes en el sistema o variaciones locales en las condiciones climáticas.

4.2.5.1 Gráficas de Tendencia Lineal

La **Figura 76** muestra la variación semanal de los niveles de agua en los tanques N1 (línea azul) y N2 (línea verde) durante los meses de septiembre, octubre y noviembre, donde se observa una tendencia general estable con ligeras fluctuaciones. En septiembre, los niveles comienzan más bajos, especialmente en N1, alrededor de 183 cm, mostrando incrementos graduales. En octubre, los niveles tienden a estabilizarse con pequeñas variaciones entre semanas. Finalmente, en noviembre, se presenta un incremento significativo, alcanzando su punto más alto en N2 (aproximadamente 186.5 cm) y un incremento similar en N1. Este comportamiento refleja la recuperación progresiva de los niveles, posiblemente por mejoras en el suministro o ajustes en el sistema, destacando la importancia de continuar monitoreando las semanas críticas donde se registran mayores fluctuaciones.

Figura 76

Gráfica Lineal de promedios semanales del nivel de agua

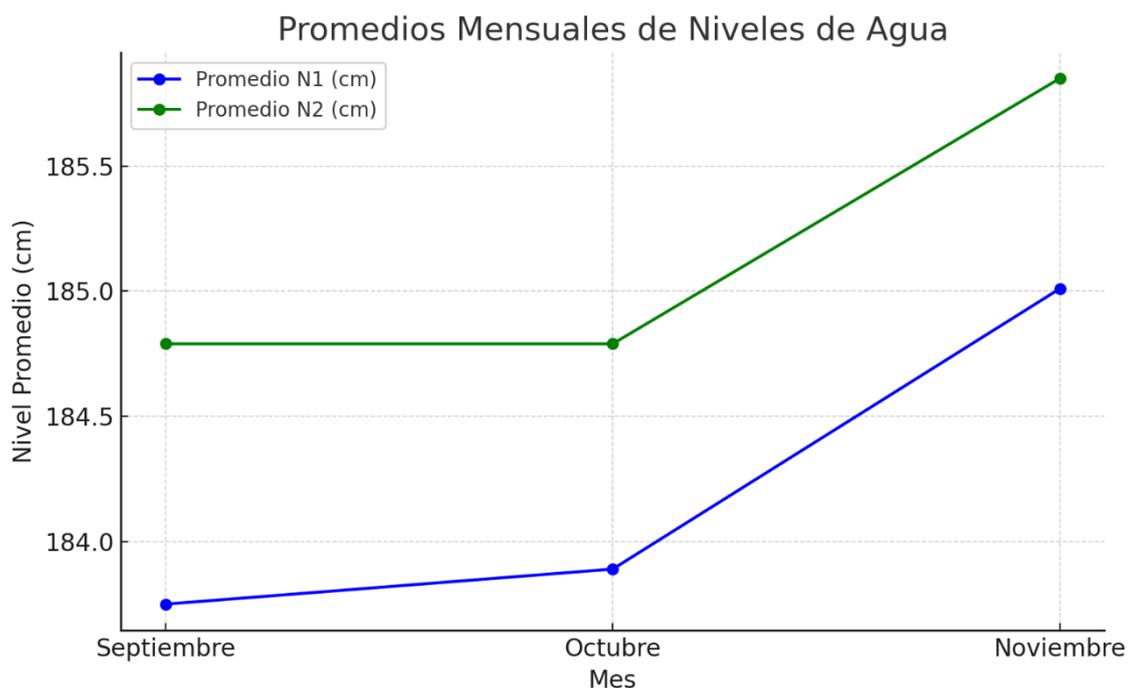


Fuente: Autoría

La **Figura 77** muestra los promedios mensuales de los niveles de agua en los tanques N1 y N2 durante septiembre, octubre y noviembre. Se observa que los niveles de agua en N1 (línea azul) muestran un incremento progresivo, pasando de 183.75 cm en septiembre a 185.01 cm en noviembre. Por otro lado, los niveles de agua en N2 (línea verde) permanecen estables entre septiembre y octubre, con un promedio de 184.79 cm, y experimentan un aumento notable en noviembre, alcanzando 185.85 cm. Este comportamiento sugiere una mejora en el suministro de agua o una recuperación en el sistema durante noviembre, reflejando una tendencia positiva en ambos tanques hacia el final del periodo evaluado.

Figura 77

Representación Lineal del promedio mensual del nivel de agua



Fuente: Autoría

4.2.5.2 Histograma de Variación (Umbrales)

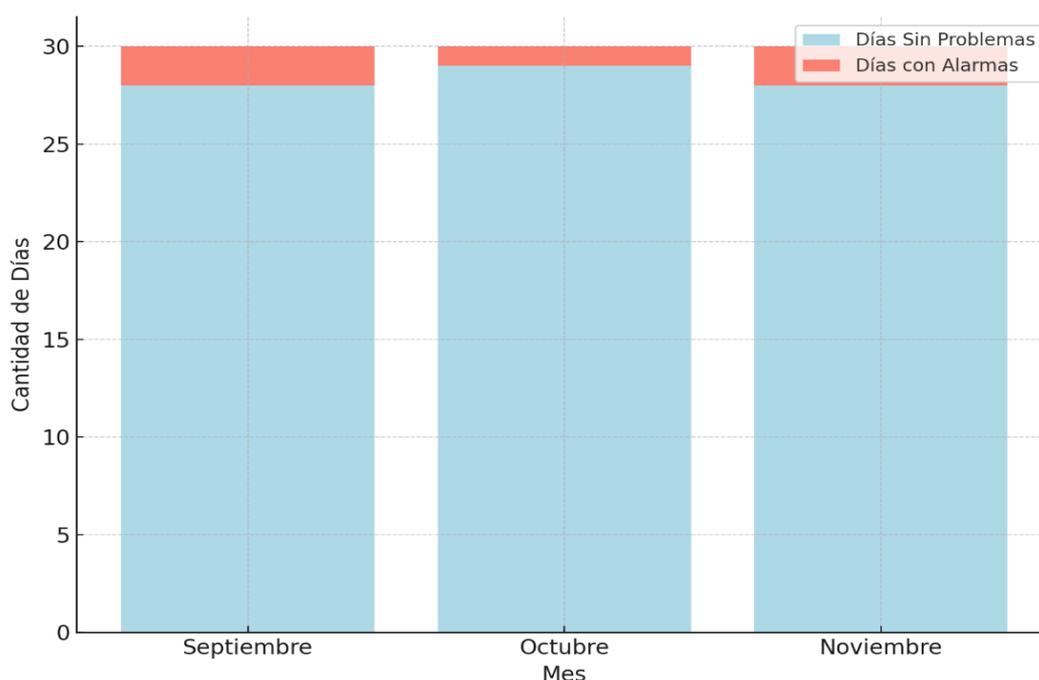
La **Figura 78** muestra claramente la distribución de días con alarmas y días sin problemas para cada mes evaluado:

- **Septiembre:** Hubo 2 días con alarmas por desviaciones fuera del umbral, lo que representa un 6.67% del mes. Los otros 28 días (93.33%) no presentaron problemas, lo que indica una estabilidad predominante.
- **Octubre:** Se registró 1 día con alarma, equivalente a un 3.33% del mes. Los 29 días restantes (96.67%) se mantuvieron dentro del umbral seguro, lo que refleja un mejor desempeño en comparación con los otros meses.
- **Noviembre:** Al igual que en septiembre, hubo 2 días con alarmas, lo que representa un 6.67% del mes. Los 28 días restantes (93.33%) no mostraron desviaciones.

En los 3 meses evaluados, se registraron un total de 5 días con alarmas de 90 días posibles, lo que corresponde a un 5.56% del tiempo con fallos. Por lo tanto, 94.44% de los días transcurrieron sin problemas y los niveles de agua se mantuvieron dentro del umbral.

Figura 78

Representación gráfica de días con fallas en cada mes de prueba



Fuente: Autoría

4.3. Fase 4: Verificación de Funcionamiento

La verificación es el último paso en el modelo en cascada utilizado en este proyecto y su objetivo es asegurarse de que el sistema, ya implementado, funcione correctamente y cumpla con lo que se esperaba. En esta fase, se revisa todo el sistema para confirmar que sus componentes trabajen de manera adecuada y que los resultados sean los esperados. También se comprueba que el sistema cumpla con las necesidades del usuario y que funcione correctamente en condiciones reales.

4.3.1. Verificación de datos receptados en plataformas de visualización

Estas plataformas son las principales herramientas a través de las cuales el usuario final interactúa con el sistema. Por ello, resulta importante asegurar que dichas herramientas ofrezcan información precisa, clara y en tiempo real.

En los siguientes puntos se describen las herramientas empleadas en el sistema de medición de nivel de agua potable, detallando su funcionalidad y el papel que desempeñan en garantizar un rendimiento óptimo del sistema.

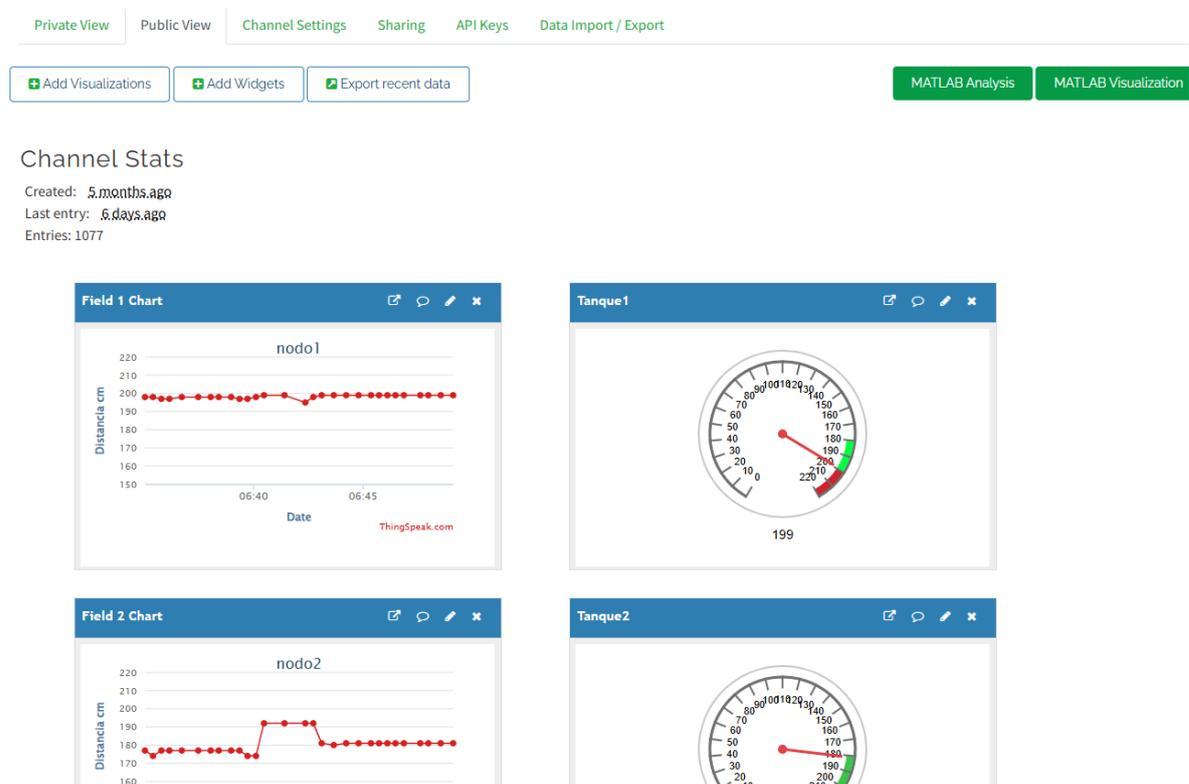
4.3.1.1. ThinSpeak

ThingSpeak es la principal plataforma de visualización de los datos del nivel de agua potable de las captaciones de redes de distribución media y baja de la ciudad de San Gabriel. Esta plataforma permite recopilar, procesar y representar los datos de forma interactiva mediante gráficos y paneles intuitivos. Además, ofrece la capacidad de incrustar estas visualizaciones en una página web, facilitando el acceso público a la información en tiempo real.

Se ha hecho uso de indicadores específicos y gráficos de campo para representar información clave, como niveles de agua, tendencias históricas y alertas críticas como se muestra en la **Figura 79**. Esto no solo mejora la comprensión de los datos, sino que también permite a los usuarios identificar patrones y responder rápidamente ante cualquier situación irregular. En conjunto, la implementación de ThingSpeak contribuye significativamente a una gestión más transparente y eficiente de los recursos hídricos.

Figura 79

Dashboard de datos públicos del nivel de agua de ThingSpeak



Fuente: Autoría

4.3.1.2. Página Web

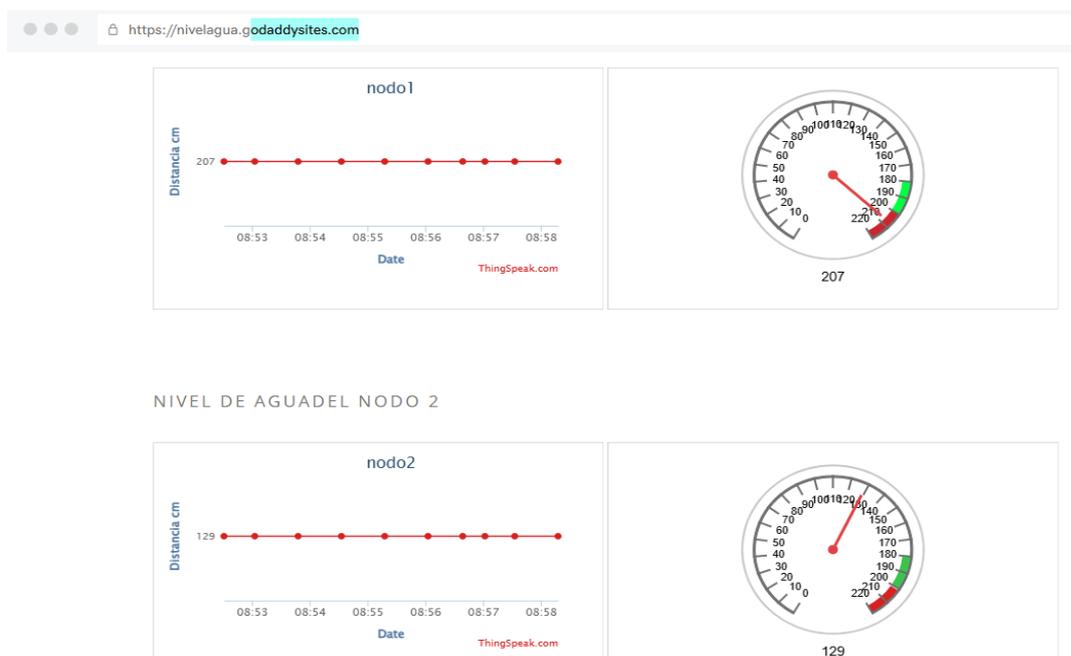
Para el desarrollo de la página web se ha utilizado GoDaddy, una plataforma que permite la creación de sitios con dominios gratuitos que facilitan la publicación de información de manera accesible y pública. En este caso, la página web ha sido diseñada

para mostrar imágenes, detalles del lugar y, principalmente, los datos en tiempo real del nivel de agua de los dos nodos del sistema, se puede evidenciar en la **Figura 80**. La elección de esta plataforma responde a la necesidad de ofrecer información confiable y disponible públicamente, lo cual permite a cualquier usuario consultar los datos en el momento que lo desee.

A diferencia de la plataforma Thingspeak, que limita el acceso a ser privado y requiere credenciales, la página web brinda una solución abierta que facilita la transparencia y accesibilidad de la información. Los datos estarán a disposición del operador y de la dirección de agua potable, quienes podrán utilizarlos y gestionarlos según sus necesidades. Adicionalmente, esta herramienta servirá como una fuente de verificación confiable en situaciones donde el operador necesite confirmar información, ya sea por una notificación recibida por correo o cuando se requiera analizar el funcionamiento del sistema.

Figura 80

Visualización de Datos del nivel de agua en Página Web



Fuente: Autoría

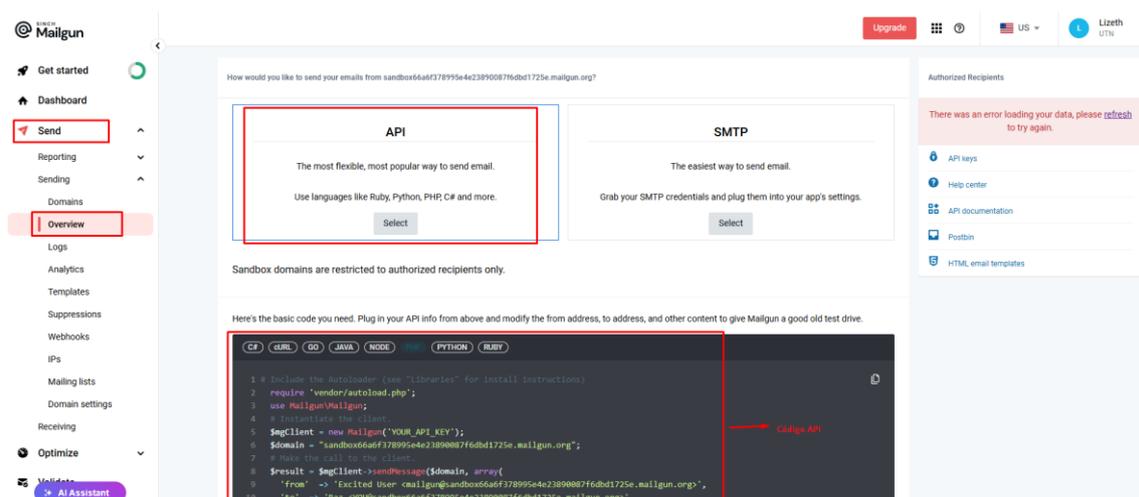
4.2.3.3. Configuración de Alarma en base a umbrales

Para la emisión de las alarmas se ha utilizado Mailgun, un servicio especializado en el envío de correos electrónicos a través de una API. Mailgun facilita la integración de envíos automáticos de correos mediante código, lo que lo convierte en una herramienta eficiente y versátil para la notificación en tiempo real, su dashboard se puede evidenciar en la **Figura 81**.

Mailgun, al utilizar una API, permite realizar envíos rápidos y automatizados con alta fiabilidad, facilitando la entrega de los correos en tiempo real. Además, cuenta con características como la gestión de registros de entrega, monitoreo de correos y verificación de la recepción, lo que asegura que las alertas sean efectivas y cumplan con su propósito.

Figura 81

Dashboard de Mailgun para la emisión de alertas de correo según umbrales de nivel de agua



Fuente: Autoría

En la programación de la Raspberry se monitorea de manera continua los niveles de agua de dos tanques a través de una Raspberry Pi conectada a un dispositivo que envía

los datos. Los valores leídos determinan si los niveles se encuentran dentro de los umbrales seguros o si han alcanzado valores críticos. Se establecen dos límites importantes: un nivel mínimo de 180 centímetros y un nivel máximo de 200 centímetros.

Si el nivel de agua baja de 180 cm, el sistema envía una alerta indicando un nivel demasiado bajo. Por otro lado, si el nivel supera los 200 cm, se emite una alerta de nivel demasiado alto. Para evitar envíos excesivos, una vez que se envía una alerta, el sistema espera 5 minutos (300 segundos) antes de enviar otra notificación similar. El script también cuenta con mecanismos de autor reinicio para solucionar posibles problemas, como fallos en la conexión o falta de datos. Si se detecta algún error, se intenta reiniciar la conexión con el dispositivo, y si persiste, el sistema reinicia la ejecución por completo para garantizar su funcionamiento continuo. La **Tabla 24** muestra cada uno de los aspectos mencionados anteriormente, pero de forma más resumida.

Tabla 24

Umbrales, manejos de errores y condiciones de alarma del sistema de medición de nivel de agua y alarma

Aspecto	Descripción
Umbrales establecidos	Nivel bajo: Menor a 180 cm. - Nivel alto: Mayor a 200 cm.
Acción al detectar alertas	Envía correo electrónico notificando nivel bajo o alto mediante Mailgun.
Frecuencia de alertas	Una vez enviada una alerta, el sistema espera 5 minutos aproximados (300 segundos) antes de enviar otra alerta similar.
Monitoreo en tiempo real	Los niveles se envían a la plataforma ThingSpeak para visualización en gráficos.

Manejo de errores Si hay fallos en la lectura de datos o conexión: - El sistema intenta reiniciar la conexión. - Si falla, reinicia el script automáticamente.

En la **Figura 82** se puede observar la recepción de correos electrónicos, donde se ha aprobado y verificado el correcto funcionamiento. Esto confirma que las notificaciones se emiten de manera adecuada cuando los datos registrados alcanzan o superan los límites configurados, lo cual garantiza la efectividad del sistema de alertas implementado.

Figura 82

Recepción de correos de Notificación ante alertas de Umbrales establecidos

Primary	Promotions	Social
☆ gladyscardenas172 4	Notificación nivel de agua de tanques en CHAMIZO - Alerta uno de los tanques tiene BAJO nivel de agua. Valores actuales: t1=197, t2=177 REVISE LA PAGINA WEB	
☆ gladyscardenas172 7	Notificación nivel de agua de tanques en CHAMIZO - Alerta uno de los tanques tiene BAJO nivel de agua. Valores actuales: t1=168, t2=184 REVISE LA PAGINA WEB	
☆ gladyscardenas172 3	Asunto del correo - Alerta uno de los tanques tiene ALTO nivel de agua. Valores actuales: t1=185, t2=202	
☆ gladyscardenas172 9	Asunto del correo - Alerta uno de los tanques tiene BAJO nivel de agua. Valores actuales: t1=168, t2=169	
☆ Equipo de cuentas M.	Comprobar tu dirección de correo electrónico - Cuenta Microsoft Comprobar tu dirección de correo electrónico Para finalizar la configuración de la cuenta Microsoft, tan solo necesitamos ...	
☆ gladyscardenas172 4	Alerta de Nivel de Agua Bajo - El nivel de agua es bajo: 31. Por favor, verifica el sistema.	
☆ Mailgun	Would you like to receive emails from UTN on Mailgun? - Hi there, Mailgun account "UTN" provided your address to test their integration with Mailgun. Please click the link below if you agree...	

Fuente: Autoría

4.3.2. Certificación por el Operador Autorizado

Una forma de comprobación del sistema de medición de nivel de agua consiste en dar a conocer su funcionamiento al operador responsable del cuidado del sistema, quien además será el principal beneficiario del sistema. Una vez que el operador comprende el sistema y se han recopilado los datos correspondientes sobre el nivel de agua, se procede a certificar su correcto funcionamiento. Este procedimiento se realiza mediante un

certificado emitido por el operador, que documenta y valida que el sistema opera de manera adecuada, como se evidencia en la **Figura 83**.

Figura 83

Certificado de funcionamiento del sistema de medición de nivel de agua por parte del Operador del área de captación

CERTIFICADO DE FUNCIONAMIENTO

Yo, Roberto Cárdenas, en calidad de Operador del Sistema del área de captación del sistema que alimenta a las redes de distribución media y baja, certifico que el sistema denominado:

"DISEÑO DE UN SISTEMA INALÁMBRICO DE MONITOREO Y NOTIFICACIÓN DEL NIVEL DE AGUA POTABLE EN LA CAPTACIÓN DE ABASTECIMIENTO DE LAS REDES DE DISTRIBUCIÓN MEDIA Y BAJA DE SAN GABRIEL, PROVINCIA DEL CARCHI"

ha sido sometido a pruebas exhaustivas y se encuentra funcionando de manera normal, sin presentar ningún tipo de inconveniente. Las pruebas realizadas comprenden todas las funcionalidades principales del sistema, incluyendo:

1. Monitoreo de niveles de agua potable en tiempo real.
2. Notificaciones automáticas en caso de variaciones o niveles críticos.
3. Estabilidad y conectividad de la red inalámbrica utilizada.

El sistema cumple con los estándares establecidos y está operativo para garantizar la eficiencia en la captación y distribución de agua potable en las redes media y baja del sector de San Gabriel, Provincia del Carchi.

Este certificado se emite a los 26 días del mes de diciembre del año 2024 para los fines que se consideren pertinentes.

Atentamente,



Roberto Cárdenas Operador del Sistema Área de Captación

Fuente: Autoría

En la **Figura 84** se puede observar al operador manipulando el sistema, esta manipulación es necesaria, ya que el sistema requiere la recarga de sus baterías al menos una vez cada 13 días para garantizar su funcionamiento continuo y eficiente. Además, de esta interacción también se permite al operador detener el sistema de manera segura, en caso de que sea necesario realizar ajustes, inspecciones o resolver algún inconveniente del área de captación. Este procedimiento asegura que el sistema permanezca en óptimas condiciones, facilitando su uso prolongado y minimizando posibles interrupciones en su desempeño.

Figura 84

Manipulación del nodo de recolección de datos de nivel de agua por parte del operador del área de captación



Fuente: Autoría

4.3.1. Verificación de Frecuencia

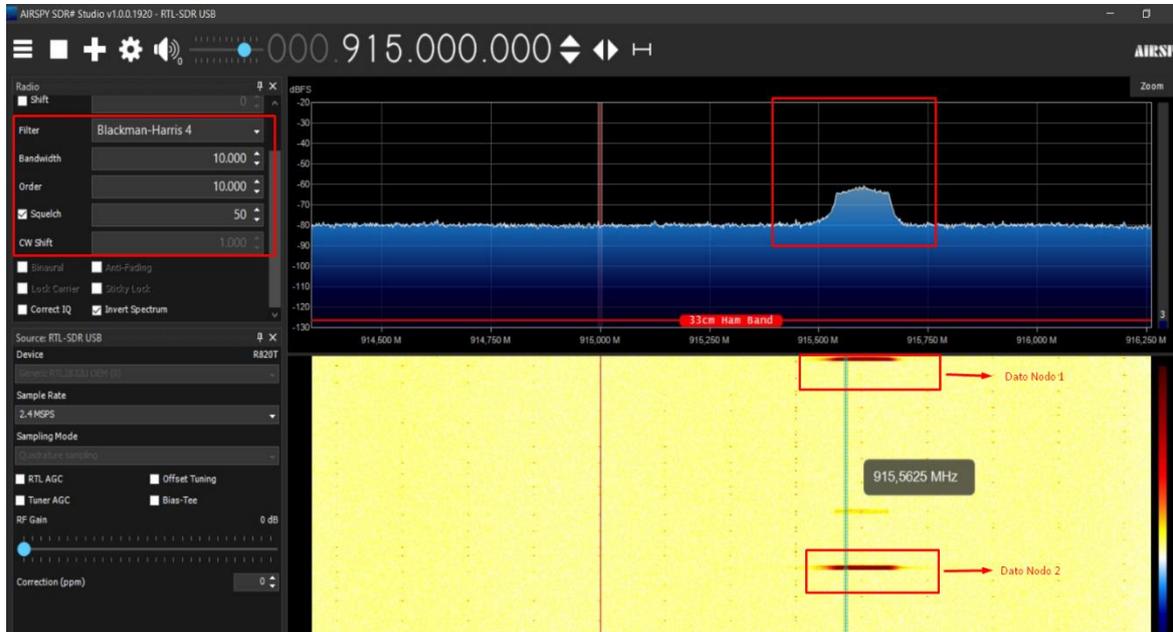
La **Figura 85** corresponde al software SDRSharp, utilizado para la recepción y análisis de señales de radio mediante un receptor RTL-SDR USB con un sintonizador R820T. En este caso, se está trabajando sobre una frecuencia central de 915.000 MHz, donde se observa una señal activa claramente detectada en 915.5625 MHz. Esta frecuencia cae dentro de la banda de 33 cm para radioaficionados, que se utiliza frecuentemente en aplicaciones inalámbricas.

El filtro configurado es de tipo Blackman-Harris 4, con un ancho de banda ajustado a 10.000 kHz (10 kHz), lo que permite aislar la señal deseada al tiempo que reduce las interferencias externas. El squelch está configurado en 50, lo que significa que solo las señales que superen este umbral serán consideradas válidas, ignorando las más débiles.

La visualización de la señal se divide en dos partes. En el gráfico superior, que representa el espectro de potencia en tiempo real, la señal se aprecia con un nivel cercano a -40 dBFS, lo que indica una intensidad relativamente alta en comparación con el ruido de fondo. En el gráfico inferior, conocido como waterfall o cascada, la señal aparece como una franja roja brillante en 915.5625 MHz, lo que confirma su persistencia e intensidad en el tiempo. Los tonos amarillos predominantes en el waterfall representan el ruido de fondo, mientras que los tonos más oscuros y rojizos indican señales más fuertes.

Figura 85

Captura de señal de radiofrecuencia entre nodos recolectores de datos y nodo Gateway mediante SDR Sharp



Fuente: Autoría

El sistema está configurado con una tasa de muestreo de 2.4 MSPS (Megasamples por segundo), lo que determina la cantidad de información de frecuencia capturada en un instante. El RF Gain se encuentra en 0 dB, por lo que no se está aplicando ganancia adicional al receptor. Estas configuraciones aseguran una recepción estable y permiten analizar con precisión la señal detectada en la frecuencia objetivo.

4.4. Discusión: Comparativa entre inspección manual e inspección automatizada

En este apartado se realiza una comparación entre el método tradicional de inspección manual y el nuevo sistema automatizado implementado en el desarenador para la medición del nivel de agua potable. Se demuestra cómo la automatización ha optimizado el proceso, mejorando la eficiencia y efectividad del monitoreo.

4.4.1 Descripción del Método de Inspección Manual

Anteriormente, el monitoreo del nivel de agua en el desarenador dependía únicamente de un medidor visual. Este método implicaba que los operadores debían realizar visitas diarias al sitio (**Figura 86**) para:

- Verificar manualmente los niveles de agua.
- Registrar las mediciones y reportar o resolver cualquier anomalía.

Este proceso demandaba tiempo y esfuerzo continuo, limitando al personal en sus otras responsabilidades y dificultando la identificación oportuna de problemas fuera del horario laboral. Además, en situaciones críticas (sequías o lluvias intensas), los operadores no podían detectar rápidamente problemas como:

- Pérdida de agua durante la sequía.
- Taponamientos del sistema de captación en épocas de lluvia.

Figura 86

Medición Visual del Nivel de Agua en Desarenador del área de Captación



Fuente: Autoría

4.4.2. Sistema Automatizado

Las características más importantes del nuevo sistema son:

- **Medición en tiempo real:** El sistema registra automáticamente los niveles de agua y envía los datos a una plataforma web accesible para el operador.
- **Alertas y notificaciones:** Si los niveles de agua caen fuera del umbral establecido, se envía automáticamente un correo electrónico de alarma, lo que permite actuar de manera rápida y oportuna.
- **Acceso remoto:** El operador puede revisar el sistema desde cualquier lugar mediante la plataforma web, eliminando la necesidad de visitas diarias.

4.4.3 Ventajas de la Inspección Automatizada

A partir de la implementación del sistema, se evidencian varias mejoras, estas se pueden evidenciar de una forma resumida en la **Tabla 25**:

Tabla 25

Comparativa de inspección manual vs automatizada del nivel de agua

Aspecto	Inspección Manual	Inspección Automatizada
Frecuencia de monitoreo	Diaria (con riesgo de retrasos).	Continua y en tiempo real.
Reacción ante problemas	Depende de la visita del operador.	Alertas inmediatas por correo.
Eficiencia del operador	Alto tiempo invertido en visitas.	Optimización del tiempo para labores adicionales.
Accesibilidad	Solo en sitio.	Accesible mediante página web.

Detección de anomalías	Limitada y no siempre oportuna.	Inmediata, con registros automáticos.
------------------------	---------------------------------	---------------------------------------

El operador ha demostrado satisfacción con el nuevo sistema, destacando la optimización de su tiempo y la posibilidad de enfocarse en otras labores de mantenimiento pendientes. El sistema automatizado reduce la necesidad de visitas diarias, ya que solo se requiere acudir al desarenador cuando se reciben alertas de bajo nivel de agua o cuando hay trabajos programados.

Durante el proceso de implementación, se ha entregado a los operadores un manual de usuario (ver **Anexo 5. Guía rápida de Usuario**), el cual incluye:

- Descripción resumida del sistema y su funcionamiento.
- Enlaces de acceso a la plataforma web.
- Contactos de soporte en caso de fallas o inconvenientes técnicos.

4.5. Análisis Costo - Beneficio

Un análisis costo-beneficio es una herramienta que permite evaluar la viabilidad económica de un proyecto al comparar los costos totales de implementación con los beneficios obtenidos. Este tipo de análisis facilita la toma de decisiones al demostrar si los recursos invertidos justifican los resultados esperados, optimizando así el uso del presupuesto asignado.

A continuación, se presentan los costos de hardware (**Tabla 26**) y costos de software (**Tabla 27**) del sistema de monitoreo y notificación de nivel de agua potable. Estas tablas detallan los elementos utilizados, sus precios unitarios, la cantidad requerida y el subtotal final. Además, se incluirán gráficas ilustrativas que permitirán visualizar de

manera clara y estructurada la distribución de costos, facilitando el análisis comparativo y resaltando la eficiencia económica del sistema implementado.

4.5.1 Costos de Hardware

En la *Tabla 26*, se evidencian los elementos de hardware utilizados para la implementación del sistema de monitoreo y notificación del nivel de agua, en donde se podrá visualizar el dispositivo, costo y cantidad.

Tabla 26

Costo de Hardware del sistema de medición del nivel de agua

Material	Precio unitario	Cantidad	Total
Sensores del Nivel de agua (HC-SR04)	\$2,00	2	\$4,00
Microcontrolador Central (Raspberry Pi 4)	\$80,00	1	\$80,00
Módulo de comunicación inalámbrica LoRa (RYLR998)	\$15,00	3	\$45,00
Elementos de alimentación (Power Bank)	\$15,00	2	\$30,00
Componentes de Instalación y Protección	\$10,00	3	\$20,00
Microcontrolador Secundario (ESP32)	\$12,00	3	\$36,00
Placas de circuito	\$10,00	3	\$30,00
Cables de alimentación	\$2,50	4	\$10,00
SUBTOTAL (1)			\$ 255

4.5.2 Costos de Software

En la **Tabla 27**, se evidenciará los elementos en software implementados para el correcto funcionamiento del sistema y sus valores individuales.

Tabla 27

Costo de Software del sistema de medición del nivel de agua

Material	Precio unitario	Cantidad	Total
Desarrollo de software	\$0,00	1	\$0,00
Licencias de Software	\$0,00	1	\$0,00
Plataforma de visualización y Notificación	\$0,00	1	\$0,00
Integración API y servicios externos	\$0,00	1	\$0,00
SUBTOTAL (2)			\$ 0

El análisis costo-beneficio del sistema de monitoreo y notificación del nivel de agua demuestra que la implementación ha sido altamente favorable. Con una inversión total de \$255 destinada al hardware y sin costos adicionales de software, el sistema ha permitido optimizar las tareas de monitoreo, eliminando la necesidad de visitas diarias al desarenador y proporcionando información en tiempo real. Esto ha reducido significativamente el tiempo operativo del personal, permitiendo enfocarse en otras labores importantes. Además, el sistema facilita la detección oportuna de problemas como niveles bajos o taponamientos mediante alertas automatizadas, lo cual minimiza riesgos y evita posibles pérdidas de emisión del agua.

CONCLUSIONES

El análisis de la información bibliográfica sobre tecnologías y herramientas disponibles ha permitido seleccionar las opciones más adecuadas para la medición y notificación del nivel de agua. En lo que respecta a la tecnología de comunicación inalámbrica, se ha seleccionado LoRa debido a su eficiencia en áreas rurales, donde la ausencia de obstáculos físicos permite una transmisión de datos confiable en frecuencias como 915 MHz. Esta tecnología, con su amplio alcance, es ideal para terrenos extensos y asegura una comunicación estable incluso a grandes distancias entre los nodos sensores y el Gateway central. Además, se eligió la metodología en cascada para la implementación del proyecto debido a su enfoque estructurado y secuencial, lo que facilita el control y la gestión del desarrollo de sistemas IoT.

La selección de los componentes electrónicos para este proyecto se realizó considerando los requerimientos específicos del terreno y las características del área de aplicación. A partir de un análisis de campo y un proceso de comparación técnica (benchmarking), se identificaron los dispositivos más adecuados en términos de rendimiento, eficiencia y accesibilidad dentro del mercado nacional. Esto garantizó que tanto los sensores como los sistemas de comunicación inalámbrica sean capaces de llevar a cabo la captura, transmisión y recepción de datos de manera confiable, permitiendo su procesamiento, almacenamiento y visualización en tiempo real.

Con la conexión de los nodos de medición al sistema central de monitoreo y haciendo uso de la tecnología LoRa, los nodos pueden enviar datos en tiempo real al sistema central, lo que permite un monitoreo constante de los niveles de agua. Para facilitar el seguimiento, los datos se visualizan en plataformas especializadas que proporcionan información clara y actualizada. Además, se ha incorporado un sistema de

notificación automática que alerta al operario cuando los niveles de agua son demasiado altos o bajos, lo que mejora la capacidad de respuesta ante cualquier eventualidad y asegura que las medidas correctivas se tomen de manera oportuna.

La instalación del sistema en el desarenador del área de captación asegura que los nodos de medición estén correctamente ubicados y configurados para obtener mediciones precisas. Durante la verificación, se confirma que los componentes del sistema y los umbrales de notificación estén ajustados adecuadamente para alertar al operario en tiempo real sobre los niveles de agua.

La integración de una plataforma de visualización accesible al personal encargado, autoridades locales y partes interesadas ha facilitado el monitoreo remoto y en tiempo real del sistema, optimizando el uso del tiempo y los recursos operativos al eliminar la necesidad de inspecciones físicas constantes. Esta herramienta proporciona información clara y transparente, lo que promueve la confianza y facilita la toma de decisiones informadas ante situaciones críticas, como niveles de agua fuera del umbral. Además, al ofrecer reportes y alertas oportunas, se fomenta la participación activa de las partes interesadas y se garantiza una gestión colaborativa y eficiente, permitiendo una respuesta rápida y efectiva para asegurar la continuidad del suministro de agua potable.

La implementación del sistema inalámbrico basado en tecnologías IoT ha mejorado significativamente la eficiencia en el monitoreo y control del nivel de agua potable en el área de captación. Gracias a la capacidad de realizar mediciones en tiempo real, se ha reducido la necesidad de inspecciones manuales frecuentes, superando las limitaciones de los métodos tradicionales. Además, el sistema facilita la detección temprana de problemas, como niveles bajos por sequías o taponamientos durante épocas de lluvias, mediante la emisión automática de alertas al operador encargado. Esto permite

actuar de manera rápida y oportuna, evitando desabastecimientos y asegurando un suministro continuo y confiable del recurso hídrico.

RECOMENDACIONES

Es recomendable informarse y comprender en detalle el funcionamiento del sistema de distribución de agua potable de las redes de distribución media y baja de la ciudad de San Gabriel antes de proceder con cualquier implementación. Esto permitirá identificar de manera precisa los requerimientos específicos del sistema, como los niveles críticos, puntos de captación y patrones de consumo. Con un conocimiento claro del sistema actual, se podrán tomar decisiones más acertadas respecto a la selección de tecnologías, sensores y métodos de comunicación, garantizando así un enfoque integral y eficiente para abordar las necesidades del sistema de monitoreo y optimizar su desempeño.

Realizar un seguimiento periódico del sistema implementado para garantizar su correcto funcionamiento, identificar posibles fallas técnicas y asegurar la continuidad del monitoreo de los niveles de agua potable. Esto debe complementarse con un plan de mantenimiento preventivo que incluya la revisión regular de los sensores, microcontroladores y módulos de comunicación, asegurando así la prolongación de la vida útil del sistema y evitando interrupciones en la recolección y transmisión de datos.

Es importante capacitar regularmente al personal encargado del sistema en cuanto a su operación, mantenimiento y posibles actualizaciones tecnológicas. Esto garantizará que puedan gestionar adecuadamente cualquier incidencia y optimizar el uso del sistema. Además, se debe incluir en la capacitación el procedimiento para recargar las baterías de los nodos secundarios en caso de ser necesario, asegurando así el correcto funcionamiento continuo del sistema.

Asegurarse de que no existan obstáculos físicos significativos, como vegetación densa, estructuras o terreno irregular, que puedan causar interferencias o comprometer la

línea de visión entre los nodos de recolección y el Gateway. O de otra forma se recomienda que las antenas de los módulos LoRa sean instaladas a una altura adecuada o distanciadas del nodo de recolección hasta un máximo de 10 metros, optimizando así la propagación de la señal y reduciendo las pérdidas. Además, en casos de problemas persistentes, se sugiere utilizar antenas direccionales para mejorar la ganancia de la señal o instalar repetidores intermedios que fortalezcan la conexión, asegurando la estabilidad y fiabilidad del sistema incluso en entornos desafiantes.

REFERENCIAS BIBLIOGRÁFICAS

- Abraham, J., & Huaman, L. (2023). *Escuela Académico Profesional de Ingeniería Industrial* [Escuela Académico Profesional de Ingeniería Industrial].
www.researchgate.net
- Alejandra Sánchez-Hernández, G., Enrique Júnez-Ferreira, H., Alvarez-Robles, W., González-Trinidad, J., Bautista-Capetillo, C., para correspondencia, A., & del Socorro Herrera-Zamarrón, G. (2017). *Optimización del monitoreo del nivel del agua subterránea para una frecuencia fija Resumen*. <https://doi.org/10.24850/j-tyca-2017-05-02>
- Alvear, I., & Freire, M. (2001). *SISTEMA DE DESARROLLO PARA MICROCONTROLADORES PIC DE LA GAMA MEDIA CON CARACTERÍSTICAS ICSP BASADO EN EL MICROCONTROLADOR PIC16F87*.
<https://bibdigital.epn.edu.ec/bitstream/15000/10699/1/T1821pt.1.pdf>
- Amazon. (2024a). *Amazon.com: Power-Bank-Solar-Charger - Cargador portátil de 42800 mAh, banco de energía solar, batería externa 5V3.1A Qc 3.0 cargador rápido linterna súper brillante integrada (naranja) : Celulares y Accesorios*.
<https://www.amazon.com/-/es/Power-Bank-Solar-Charger-Cargador-port%C3%A1til-brillante-integrada/dp/B0C4JTVBBH?th=1>
- Amazon. (2024b). *Amazon.com: Raspberry Pi 4 modelos B 2019 Quad Core 64 bits WiFi Bluetooth (4GB) : Electrónica*. https://www.amazon.com/-/es/Raspberry-modelos-2019-Quad-Bluetooth/dp/B07TC2BK1X/ref=sr_1_3?adgrpid=151058848061&dib=eyJ2IjoiMSJ9.mP4drOfyakW9P2E6ytjWi5Zf7flYea5N2CQDqGfQ1mDzth2zssuL7jmVcQlCvEmm8nai70ukF1495mgtjgAx18e-qyX6ZV86iadP7bl_y0ezNGeuhoL-ILgjlH_n8UEGQHA5dqP5TCZOa5DbOy2CgpMga9I7TRd2R01F4FCWShSJtheLBdccSo4xCh0maMRCmuMeM5iPtJ1EJQXQz_ra5R3E0a3XO5F9KRugsE19DvY.ZBKAJdZm3-7QjxQzAuTnFks_k1KxfyiD2Sw0sOEDqOo&dib_tag=se&gad_source=1&hvadid=690309985572&hvdev=c&hvlocphy=9076633&hvnetw=g&hvqmt=e&hvrnd=8198919513395455487&hvtargid=kwd-297124344473&hydadcr=8150_13695105&keywords=raspberry%2Bpi%2B4&qid=1716205078&sr=8-3&th=1
- Amazon. (2024c). *Amazon.com: REYAX Módulo de Lora del interfaz 868/915 MHz del UART de RYLR998 con la certificación del CE NCC de la FCC de la antena : Electrónica*. <https://www.amazon.com/-/es/M%C3%B3dulo-interfaz-RYLR998-certificaci%C3%B3n-antena/dp/B099RM1XMG>
- Amazon. (2024d). *Amazon.com: Sensor de distancia ultrasónico para Arduino, Raspberry Pi, Robots | MB1240-000 XL-MaxSonar-EZ4 | Rangos de 7.9 in a 301.2 in | Maxbotix Inc. : Industrial y Científico*. <https://www.amazon.com/-/es/distancia->

ultras%C3%B3nico-Raspberry-MB1240-000-XL-MaxSonar-EZ4/dp/B00A7YH1RE

- Angulo, M. A., & Saavedra, O. C. (2018). MODELACIÓN HIDROLÓGICA DE LA CUENCA MAYLANCO UTILIZANDO HEC-HMS. *INVESTIGACION & DESARROLLO*, 18(1), 53–67. <https://doi.org/10.23881/IDUPBO.018.1-4I>
- APM ELECTRONICS. (2024). *ESP32 DE 38 PINES - APM*. <https://apmelectronica.com/producto/esp32-de-38-pines/>
- Ayda, L., Matías, R., Alirio, F., & Rojas, R. (2021). *Los stakeholders, clave para el éxito de la gestión de proyectos*.
- Barrera, V., & Leonardo, C. (2018). *Sistema para la medición de gases de efecto invernadero mediante los principios del internet de las cosas, alineado al cumplimiento de los compromisos de Colombia ante las naciones unidas*. <https://api.semanticscholar.org/CorpusID:172075929>
- Bernabé-Crespo, M. B. (2022). Implicaciones y perspectivas del mix hídrico para el abastecimiento de agua potable en el sureste de España. *Agua y Territorio / Water and Landscape*. <https://api.semanticscholar.org/CorpusID:246726910>
- Bertoleti, P. (2019). *Proyectos con ESP32 y LoRa*. Instituto Newton C Braga. https://books.google.com.ec/books?id=Doi0DwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Bravo, F., Baldeos, L., Lioo, F., Marín, W., Villanueva, D., & Neri, A. (2023, January 12). Sistemas de gestión de calidad en las empresas de servicios de agua potable y alcantarillado en Barranca, Perú. *Konrad Lorenz*. <https://doi.org/10.14349/sumneg/2023.V14.N30.A2>
- Cepeda, J. (2018). *Smart Cities y el Internet de las Cosas, un nuevo paradigma tecnológico Demostración de los beneficios mostrados en el estudio. Definición y estudio de sustentabilidad y factibilidad*. [Universidad de Guayaquil]. <https://repositorio.ug.edu.ec/server/api/core/bitstreams/407b526e-c189-4737-8332-8ec7bb68b739/content>
- Consejo Nacional de Planificación (CNP). (2017). *Plan Nacional Para el Buen Vivir*. <https://www.gobiernoelectronico.gob.ec/wp-content/uploads/downloads/2017/09/Plan-Nacional-para-el-Buen-Vivir-2017-2021.pdf>
- Cuartero, P. (2017). *HC-SR04 – Componentes electrónicos*. <https://componenteselectronicossite.wordpress.com/2017/03/30/hc-sr04/>
- Delgado, M., & Banguera, F. (2019). *Diseño e Implementación de un Sistema de Monitoreo y Adquisición de datos de la Planta Didáctica MPS PA. Compact Workstation de Festo utilizando Hardware libre y Protocolo de Comunicación SMTP Y GSM* [Universidad Politécnica Salesiana]. <https://dspace.ups.edu.ec/bitstream/123456789/17266/1/UPS-GT002586.pdf>

- Dender-Zurita, R. C., Guilcatoma-Moreira, P. M., Argandoña-Moreira, J. G., & Machuca-Avalos, M. P. (2018). Gestión de redes de sensores inalámbricos (WSN) para la industria petrolera. *Polo Del Conocimiento*, 2(12), 15. <https://doi.org/10.23857/PC.V2I12.407>
- Dunford Pabst, A. K. (2022). *Development and testing of LoRa Technology for satellite communications*. <https://repositorio.uc.cl/handle/11534/66333>
- El Universo. (2019, June 23). *El 75% de San Gabriel, en Carchi, sin agua por deslave causado por lluvias | Ecuador | Noticias | El Universo*. <https://www.eluniverso.com/noticias/2019/06/23/nota/7391748/75-san-gabriel-carchi-agua-deslave-causado-lluvias/>
- ESGEEKS. (2023). *Plataformas Online para Internet of Thing (IoT)*. <https://esgeeks.com/plataformas-online-para-iot/>
- Fernández, J. (2021). (PDF) *Utilización de dispositivos móviles como herramienta de sensado en aplicaciones de IoT Trabajo Final presentado para obtener el grado de [Universidad Nacional de La Plata]*. <https://doi.org/10.13140>
- GAD Montufar. (2015). *Plan de Desarrollo y Ordenamiento Territorial Cantón Montufar*. https://gadmontufar.gob.ec/documents/PD%20y%20OT%202015%20MONTUFAR_alineado_al_PND_2017-2021.pdf
- Idrovo, R. (2021). (PDF) *Monitorización en la nube de “livestock” para aplicaciones IoT en el ámbito de “Smart Agriculture.”* <https://doi.org/10.13140>
- Ionos. (2019, March 11). *El modelo en cascada en el desarrollo de software - IONOS*. [https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#:~:text=El%20desarrollo%20en%20cascada%20\(en,ejecuta%20tan%20sola%20una%20vez.](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#:~:text=El%20desarrollo%20en%20cascada%20(en,ejecuta%20tan%20sola%20una%20vez.)
- La Hora. (2022, August 16). *Captación de agua potable de San Gabriel peligra por deslaves – Diario La Hora*. <https://www.lahora.com.ec/imbabura-carchi/deslaves-san-gabriel-cortes-agua-potable/>
- Lama. (2023, June 27). *Potabilización de Agua: Su Importancia y el Proceso*. <https://www.lamastore.es/blog/potabilizacion-de-agua/>
- Llashac, A. (2017). *IMPROVEMENT OF THE CAPTATION CHAMBER, DRIVING LINE AND RESERVOIR STORAGE OF DRINKING WATER OF THE CASERÍO BELLA VISTA, DISTRICT OF CÁCERES DEL PERÚ, PROVINCE OF SANTA, ÁNCASH REGION - 2017*. <https://repositorio.uladech.edu.pe/bitstream/handle/20.500.13032/13140/ARTICULO%20LLASHAC.pdf?sequence=1&isAllowed=y>
- López, Á. H. A., Castro, J. A. Q., Perdomo, E. G., & Zambrano, A. M. M. (2022). WSN REDES DE SENSORES INALÁMBRICOS Y SU APLICACIÓN A LA AGRICULTURA DE PRECISIÓN: UN CASO DE METAANÁLISIS. *ACTAS DEL VII CONGRESO INVESTIGACIÓN, DESARROLLO E INNOVACIÓN DE LA*

- UNIVERSIDAD INTERNACIONAL DE CIENCIA Y TECNOLOGÍA IDI-UNICYT 2022, 245–258. <https://doi.org/10.47300/ACTASIDI-UNICYT-2022-38>
- LoRa Alliance. (2020). *RP002-1.0.2 LoRaWAN® Regional 40 Parameters*. https://loralliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf
- Luis, J., Pérez, V., Manuel, J., & Castro, A. (2022). *Desarrollo de un sistema de comunicación inalámbrico basado en tecnología LoRa para el sensado remoto de variables eléctricas*. Universidad Politécnica Salesiana.
- Ministerio de Agricultura y Ganadería Costa Rica. (2010). *"MANUAL DE ESPECIFICACIONES TÉCNICAS BÁSICAS PARA LA ELABORACIÓN DE ESTRUCTURAS DE CAPTACIÓN DE AGUA DE LLUVIA (SCALL) EN EL SECTOR AGROPECUARIO DE COSTA RICA Y RECOMENDACIONES PARA SU UTILIZACIÓN"*. www.cemedede.una.ac.cr.
- Mitchell, R. (2018). *How to Use Python With Raspberry Pi | Raspberry Pi | Maker Pro*. <https://maker.pro/raspberry-pi/tutorial/how-to-use-python-with-raspberry-pi>
- Montes, A., & Camacho, L. (2020). *Telemetría a través de redes de área extensa de baja potencia (LPWA) y en coexistencia con el internet de las cosas (IoT)*. https://web.archive.org/web/20200319041631/http://repository.udistrital.edu.co/bitstream/11349/16025/1/Telemetria_a_traves_de_redes_de_area_extensa_de_baja_potencia_%28LPWA%29_y_en_coexistencia_con_el_internet_de_las_cosas_%28IoT%29.pdf
- Orozco, J. R. (2023). Análisis del uso de una aplicación móvil para reducir el agua potable no contabilizada. *Agua, Saneamiento & Ambiente*. <https://revistas.usac.edu.gt/index.php/asa/article/view/1526/1022>
- Ortiz, M. (2020). *DESARROLLO DE UNA RED DE SENSORES INALÁMBRICOS UTILIZANDO TECNOLOGÍA LORA PARA EL MONITOREO DE UN SISTEMA [UNIVERSIDAD POLITÉCNICA SALESINA]*. <https://dspace.ups.edu.ec/bitstream/123456789/18469/1/UPS%20-%20ST004461.pdf>
- Palanca, C. M. (2019). *Transmisión de datos mediante LoRa/LoRaWAN*.
- Pérez, J., Urdaneta, E., & Custodio, Á. (2014). Metodología para el diseño de una red de sensores inalámbricos. *Universidad, Ciencia y Tecnología*, 18(70), 12–22. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212014000100002&lng=es&nrm=iso&tlng=es
- Philco, A., & Orlando, L. (2023). *Calidad de sensado y eficiencia energética en redes de sensores inalámbricos (WSN) [Universidad Nacional de La Plata]*. <https://doi.org/10.35537/10915/162474>
- Ramírez Fernández, B., Jahaziel Gerson Br Aguilar Bocanegra, J., Alin Asesor, L., Torres, M., David Co-Asesor, L., Roeder, B., & Antonio, W. (2024). *Automatización de Medición de Nivel en Vertedero Hidráulico: Optimización del Laboratorio de Operaciones Unitarias en la Universidad Nacional de Trujillo*.

- Revelo, C., & Ordoñez, J. (2012). *Diseño e implementación del prototipo de un osciloscopio digital basado en el sistema embebido Mini2440* [Universidad de Nariño]. <https://sired.udenar.edu.co/3304/1/85707.pdf>
- Ricardo, M., Méndez, M., Bonet, C., Sierra, L., & Cutie, V. (2015). *Evaluación de la eficiencia de la captación de agua de lluvia en casas de cultivos*. <https://www.redalyc.org/pdf/5862/586261427001.pdf>
- Romero, Y. (2021, January). *Vista de Evaluación y rediseño de sistemas de abastecimiento de agua potable*. 28–40. <https://doi.org/https://doi.org/10.33996/rebi.v3i1.3>
- Saguapac. (2023, May 24). *¿Qué es el Agua?: Concepto, características y tipos / Saguapac*. <https://www.saguapac.com.bo/que-es-el-agua/>
- Salas, H., & Martínez, O. (2022). Evaluación y determinación del sistema de abastecimiento óptimo de agua potable del barrio Miraflores-Lircay-Angaraes-Huancavelica. *Revista Científica Ciencias Ingenieriles*, 2(1), 11–15. <https://doi.org/10.54943/RICCI.V2I1.215>
- Salas-Salvadó, J., Maraver, F., Rodríguez-Mañas, L., Sáenz de Pipaon, M., Vitoria, I., Moreno, L. A., Salas-Salvadó, J., Maraver, F., Rodríguez-Mañas, L., Sáenz de Pipaon, M., Vitoria, I., & Moreno, L. A. (2020). Importancia del consumo de agua en la salud y la prevención de la enfermedad: situación actual. *Nutrición Hospitalaria*, 37(5), 1072–1086. <https://doi.org/10.20960/NH.03160>
- Salguero, H. (2020). *DISEÑO DEL SISTEMA DE ABASTECIMIENTO DE AGUA POTABLE PARA GERONA, ZONA 8, SAN MIGUEL PETAPA, GUATEMALA* [Universidad de San Carlos de Guatemala]. <http://www.repositorio.usac.edu.gt/13540/1/Hugo%20Roberto%20Salguero%20Hernandez.pdf>
- Sergio, I., & Romo Picazo, H. (2000). *BENCHMARKING: RETOS Y RIESGOS PARA EL INGENIERO INDUSTRIAL*.
- Solectro. (2024). *Pines GPIO y su programación*. <https://solectroshop.com/es/content/60-5-pines-gpio-y-su-programacion>
- Tostratronic. (2024). *ESP32 38 Pines Bluetooth + WiFi – Tostatronic*. <https://www.tostatronic.com/product/esp32-19-pines-bluetooth-wifi/>
- Valencia, O. (2019). *Sistema de Red de Sensores Inalámbricos (WSN) portátil aplicado al monitoreo ambiental en una ciudad, como complemento a las estaciones fijas*. [Universidad de Antioquía]. https://bibliotecadigital.udea.edu.co/bitstream/10495/13218/1/OscarValencia_2019_PIT12624.pdf
- VanZandt, P. (2023, July 25). *Waterfall vs Kanban vs Scrum vs Lean. ¿Cuál es la mejor metodología ágil de gestión de proyectos?* <https://ideascale.com/es/blogs/kanban-vs-caida-vs-escrum-vs-lean/>

ANEXOS

Anexo 1. Requerimientos

En este anexo se incluye una encuesta realizada al operador del sistema de captación de agua potable. Dicha encuesta se llevó a cabo de forma presencial, con el objetivo de resolver todas las dudas existentes y determinar los requisitos de los stakeholders que se utilizarán en el proyecto. A continuación, se presentan las preguntas que ayudarán a los distintos tipos de funciones en las que se agruparán los stakeholders.

Participantes:

Ing. Sigifredo Ruales: Usuario estratégico

Sr. Roberto Cárdenas: Usuario directo

Srta. Lizeth Cárdenas: Desarrolladora

Preguntas de aprobación de ingreso y uso de instalaciones.

Pregunta 1: ¿Es posible implementar un sistema de monitoreo y notificación del nivel de agua en la captación que abastece las redes de distribución media y baja de la ciudad de San Gabriel?

Usuario estratégico: Sí, es posible, ya que cualquier innovación tecnológica destinada a mejorar la gestión y supervisión del sistema de captación es altamente valorada. Además, se cuenta con la disposición del operario local para resolver cualquier consulta técnica relacionada con el funcionamiento del sistema.

Pregunta 2: ¿Es posible obtener un certificado que avale el ingreso y el uso de las instalaciones?

Usuario estratégico: Sí, se puede emitir el certificado solicitado, y estará disponible en los próximos días una vez completado el proceso administrativo correspondiente.

Preguntas de funcionamiento del sistema de distribución de agua potable y sistema a diseñar.

Pregunta 1: ¿Qué necesidades específicas se buscan solucionar con el sistema de monitoreo y control del nivel de agua potable en la captación?

Usuario directo: Notificar de manera constante el nivel de agua potable disponible en las fosas del desarenador de la captación, permitiendo actuar de forma inmediata ante fallas o niveles bajos de agua, que suelen ser los principales problemas identificados. Además, garantizar que estas acciones correctivas puedan realizarse sin afectar el funcionamiento del sistema de distribución en general.

Pregunta 2: ¿Considera necesario medir, además del nivel de agua, otros factores como la oxigenación (oxígeno disuelto), pH, temperatura o turbidez en el sistema de captación?

Usuario directo: No es necesario implementar estas mediciones adicionales, ya que en el barrio Chiles existe un sistema de almacenamiento y distribución de agua potable que ya contempla estos procedimientos y evaluaciones. Lo que se requiere es un sistema de alerta específico para la etapa de captación y recolección inicial de agua, dado que este es un sistema independiente ubicado en un lugar apartado.

Pregunta 3: ¿Cuál de los niveles, altos o bajos, se ha observado con mayor frecuencia en el sistema de captación?

Usuario directo: Siempre se han observado niveles bajos en el sistema de captación, ya que los niveles altos son controlados mediante desfuegos que los evitan. Por el contrario,

los niveles bajos se presentan cuando ocurren taponamientos o daños en la estructura, afectando el flujo normal del agua.

Pregunta 4: ¿Cómo le gustaría recibir las notificaciones del sistema de monitoreo, por ejemplo, mediante mensajes de texto, correo electrónico, una aplicación móvil o algún otro medio?

Usuario directo: Cualquier medio estaría bien, ya sea a través de mensajes de texto o correos electrónicos.

Pregunta 5: ¿Le gustaría contar con una plataforma o página de visualización en tiempo real del nivel de agua potable, que le permita monitorearlo cuando lo desee sin depender de notificaciones?

Usuario directo: Sí, sería ideal, ya que permitiría visualizar el estado del agua sin problemas cuando sea necesario. Además, en caso de recibir una notificación, se podría verificar la información enviada y comprobar el correcto funcionamiento del sistema.

Pregunta 6: ¿Por qué considera necesario implementar un sistema de notificación y visualización para el monitoreo del nivel de agua potable?

Usuario directo: Es necesario porque no siempre es posible estar en el lugar observando el nivel del agua debido a la realización de otras actividades. En ocasiones, el sistema puede presentar fallas durante horas sin que se tenga conocimiento de ello. Esto lleva a que, cuando el sistema de distribución se ve afectado, se reciba un aviso y recién entonces se proceda a solucionar el problema. Además, es importante destacar que el sistema de captación abarca varias áreas, lo que dificulta enfocarse únicamente en una de ellas.

Pregunta 7: ¿Le gustaría recibir capacitación para aprender a manejar el sistema y poder intervenir en caso de que se presente algún fallo?

Usuario directo: No, porque no tengo conocimientos sobre el funcionamiento técnico del sistema implementado, lo que dificultaría su manipulación. Preferiría que el sistema sea completamente automático.

Pregunta 8: ¿El lugar donde se implementará el sistema está en exteriores? Si es así, ¿considera necesario utilizar elementos resistentes a las condiciones ambientales del entorno?

Usuario directo: Sí, es necesario, ya que el lugar está en exteriores y en ciertas temporadas llueve mucho, lo que podría afectar los componentes si no son resistentes.

Pregunta 9: Para las baterías del sistema, se ha planteado utilizar baterías recargables. ¿Estaría dispuesto a encargarse de recargarlas en caso de que se agoten?

Usuario directo: Sí, si el proceso es sencillo, estaría dispuesto a realizar la recarga sin problema.

Pregunta 10: ¿Cuenta con conexión Wi-Fi en su lugar de residencia para facilitar la instalación del nodo gateway?

Usuario directo: Sí, cuento con conexión Wi-Fi en mi lugar de residencia, lo que permitiría instalar el nodo gateway sin inconvenientes. La red tiene buena cobertura y estabilidad, lo que asegura una conexión adecuada para el funcionamiento del sistema.

Pregunta 11: ¿Le gustaría que el sistema realice mediciones precisas del nivel de agua, garantizando estabilidad y evitando variaciones que puedan afectar los datos obtenidos?

Usuario directo: Sí, eso es indispensable, ya que es fundamental contar con mediciones precisas y sin variaciones para garantizar el correcto monitoreo del nivel de agua y tomar decisiones oportunas en caso de cualquier anomalía.

Pregunta 12: ¿Estaría dispuesto a realizar verificaciones periódicas del sistema para garantizar el mantenimiento de su carcasa, revisar posibles desplazamientos de los sensores y, además, tomar medidas para evitar el robo de los elementos?

Usuario directo: Sí, claro, puedo revisar el sistema de vez en cuando para asegurarme de que esté en buen estado, que los sensores no se hayan movido y para evitar que alguien robe algún componente.

Pregunta 13: ¿El lugar cuenta con un suministro de energía eléctrica estable para garantizar el funcionamiento del sistema?

Usuario directo: No, el lugar no cuenta con un suministro de energía eléctrica estable, ya que está ubicado en una zona algo alejada.

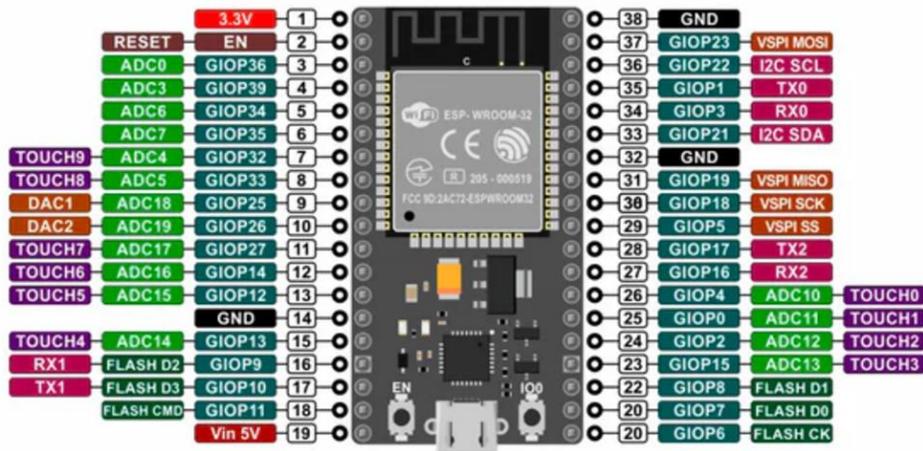
Los demás requerimientos del sistema han sido definidos con base en una investigación exhaustiva, enfocada en garantizar que el sistema cumpla con los estándares necesarios para su buen funcionamiento y operatividad. Las preguntas previamente formuladas son esenciales para establecer los criterios iniciales en la selección de los elementos que compondrán el sistema. Estas interrogantes abarcan aspectos clave como el diseño general, el modo de funcionamiento, y la ubicación específica donde se implementará el sistema. De esta manera, se asegura que cada detalle del sistema sea considerado de forma integral, logrando que las soluciones propuestas sean eficaces y adecuadas para las necesidades identificadas, tanto técnicas como operativas.

Anexo 2. Datasheets

ESP32-38 pines

Información completa en:

https://uelectronics.com/wp-content/uploads/2019/10/esp32_datasheet_en.pdf



2.2 Pin Overview

Table 2-1. Pin Overview

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip shuts down Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIO, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V ~ 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICSO, HS2_CMD, SD_CMD, MTDO

Datasheet Sensor Ultrasónico HC-SR04

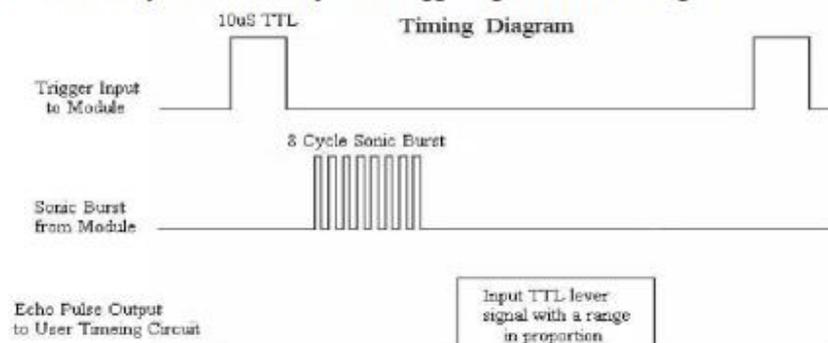
Información completa en:

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ S pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Datasheet Módulo LoRa RYLR998

Información completa en:

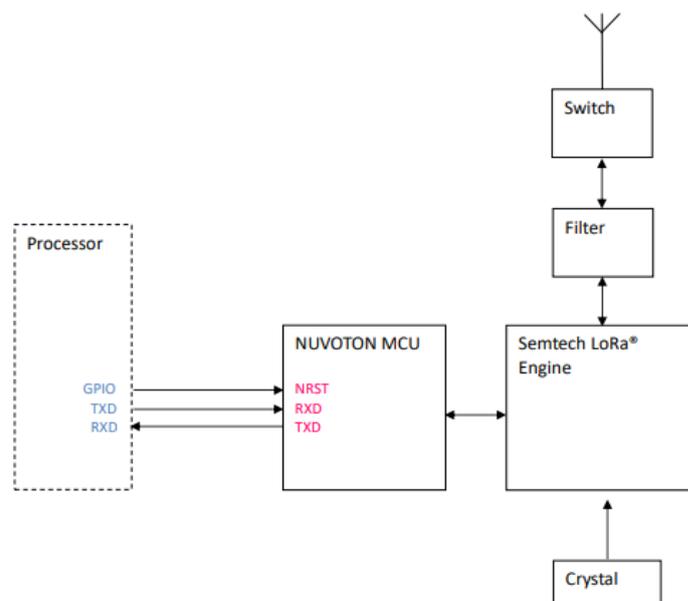
https://reyax.com//upload/products_download/download_file/RYLR998_EN.pdf

RYLR998_M4



Pin	Name	I/O	Condition
1	VDD	I	Power Supply
2	NRST	I	RESET(Active Low) 100K Ω Internal pull up, Pull down at least 100ms
3	RXD	I	UART Data Input
4	TXD	O	UART Data Output
5	GND	-	Ground

BLOCK DIAGRAM

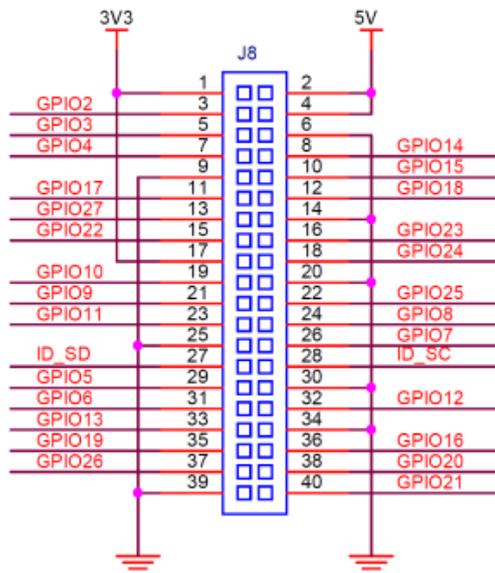


Datasheet Microcontrolador Raspberry PI 4 2GB

Información completa en: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>



5.1.1 GPIO Pin Assignments



ID_SD and ID_SC PINS:

These pins are reserved for HAT ID EEPROM.

At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers).

DO NOT USE these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.

Anexo 3. Códigos de Programación: Repositorio GitHub

<https://github.com/MagalizC/Tesis.git>

Anexo 4. Recopilación de datos no procesados

https://drive.google.com/file/d/1MFpW0kF0NXcP36JjnGAcvJO9cFKwjEXg/view?usp=drive_link

Anexo 5. Guía rápida de Usuario

https://drive.google.com/file/d/17moVDBYG3VpQiNTLl1jn0VUnnqtXwIYD/view?usp=drive_link