



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DISEÑO DE UN SISTEMA DE MEDICIÓN PARA EL NIVEL DE AGUA DEL CAUDAL DE LA PRINCIPAL FUENTE DE CAPTACIÓN DE LA CIUDAD DE TULCÁN UTILIZANDO LA TECNOLOGÍA DE COMUNICACIÓN LORA.

TRABAJO DE GRADO PREVIO A LA OBTENCION DEL TITULO DE INGENIERO EN TELECOMUNICACIONES

AUTOR: PAUL FERNANDO RAMÍREZ ORTEGA

DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA

ASESOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

IBARRA-ECUADOR

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	0402088165		
APELLIDOS Y NOMBRES:	Ramírez Ortega Paul Fernando		
DIRECCIÓN:	Ibarra		
EMAIL:	pframirezo@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0968121564

DATOS DE LA OBRA	
TÍTULO:	Diseño de un sistema de medición para el nivel de agua del caudal de la principal fuente de captación de la ciudad de Tulcán utilizando la tecnología de comunicación LoRa.
AUTOR (ES):	Ramírez Ortega Paul Fernando
FECHA:	25 de mayo de 2023
PROGRAMA:	Pregrado
TITULO POR EL QUE OPTA:	Ingeniero en Telecomunicaciones
ASESOR /DIRECTOR:	MSc. Maya Olalla Edgar Alberto

2. CONSTANCIAS

El autor (es) manifiesta (n) 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 (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 25 días del mes de mayo de 2023.

EL AUTOR:



Paul Fernando Ramírez Ortega

C.I: 0402088165



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN:

MAGÍSTER MAYA OLALLA EDGAR ALBERTO, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que el presente trabajo de Titulación "DISEÑO DE UN SISTEMA DE MEDICIÓN PARA EL NIVEL DE AGUA DEL CAUDAL DE LA PRINCIPAL FUENTE DE CAPTACIÓN DE LA CIUDAD DE TULCÁN UTILIZANDO LA TECNOLOGÍA DE COMUNICACIÓN LORA", ha sido desarrollado por el señor Ramírez Ortega Paul Fernando bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Maya Olalla Edgar Alberto
DIRECTOR

DEDICATORIA

El presente trabajo de titulación está dedicado a mis padres Clara y Fernando, quienes con su esfuerzo y sacrificio me han permitido alcanzar este objetivo tan anhelado, por su apoyo incondicional y a pesar de las dificultades me dieron un gran ejemplo de esfuerzo y perseverancia, siempre confiaron en mis capacidades y me alentaron a nunca rendirme.

A mi segunda madre, Andrea, que a pesar de la distancia siempre estuvo al pendiente de mí y me alentó con sus consejos y oraciones para que pueda lograr con éxitos mis objetivos.

A todas mis hermanas Lisseth, Cinthya y Cristina quienes me apoyaron de forma incondicional para poder continuar con mis estudios universitarios, además también demostraron sacrificio y entrega para que pueda culminar mi carrera universitaria.

A Melannie, por su amor, apoyo y compañía en esta última fase universitaria, me alentó con sus consejos y estuvo siempre presente en toda esta etapa final sin dejarme solo en ningún momento.

Paul Fernando Ramírez Ortega

AGRADECIMIENTO

En primer lugar, quiero agradecer a mis padres, ya que sin ellos no hubiese sido posible el alcance de este objetivo, con su esfuerzo y sacrificio me demostraron cuan importante era yo para ellos y todo el amor que tienen por mí, a lo largo de todo el trayecto de mi vida me han demostrado su apoyo y confianza incondicional. Mi padre y madre, que trabajan fuertemente día a día para brindarme la educación y que a pesar de la distancia nunca me hicieron sentir solo, sé que su bendición me acompañó en todo momento.

Agradezco a mi segunda madre Andrea, así como también a mis hermanas Lisseth, Cinthya y Cristina, quiénes me alentaron a seguir esta carrera universitaria y estuvieron al pendiente de mí, desde el principio me alentaron y estuvieron a mi lado de manera incondicional, demostrando un apoyo invaluable, especialmente en los momentos más difíciles.

Agradezco a mi novia, Melannie Calero, por apoyarme en esta última etapa, su dedicación y disposición para ayudarme han sido mi fuente de inspiración y motivación, gracias por creer y por alentarme a dar lo mejor de mí en cada paso del camino, estoy muy agradecido por tenerte a mi lado.

Agradezco a la Universidad Técnica del Norte por permitirme prepararme como profesional, especialmente al MSc. Edgar Maya, quien con paciencia me guió a lo largo del desarrollo del presente proyecto brindándome sus conocimientos, así como también al MSc. Jaime Michilena que me brindó la asesoría requerida para finalizar con éxito el proyecto.

Paul Fernando Ramírez Ortega

INDICE DE CONTENIDOS

IDENTIFICACIÓN DE LA OBRA	2
CONSTANCIAS	¡Error! Marcador no definido.
CERTIFICACIÓN	¡Error! Marcador no definido.
DEDICATORIA.....	5
AGRADECIMIENTO	6
INDICE DE CONTENIDOS.....	7
INDICE DE FIGURAS	12
INDICE DE TABLAS.....	18
RESUMEN	20
ABSTRACT	22
1. Capítulo I.....	24
1.1. Tema	24
1.2. Problema.....	24
1.3. Objetivos.....	25
1.3.1. Objetivo General.....	25
1.4. Alcance	26
1.5. Justificación.....	28
2. Capitulo II: Fundamentos Teóricos.....	30

2.1.	Internet de las Cosas	30
2.1.1.	Arquitectura IoT.....	30
2.2.	Redes de Sensores Inalámbricos (WSN).....	32
2.2.1.	Elementos de las WSN.....	33
2.2.2.	Arquitectura de las WSN.....	34
2.2.3.	Topologías de Red.....	35
2.2.3.1.	<i>Estrella</i>	35
2.2.3.2.	<i>Árbol</i>	36
2.2.3.3.	<i>Malla</i>	36
2.3.	Tecnologías de Comunicación Inalámbrica	36
2.3.1.	Redes de Baja Potencia y Área Amplia (LPWAN).....	38
2.3.1.1.	SigFox.....	39
2.3.1.2.	LoRa	40
2.4.	Tecnología de Comunicación LoRa	40
2.4.1.	Modulación LoRa.....	42
2.4.1.1.	<i>Factor de Dispersión (SF)</i>	43
2.4.1.2.	<i>Ancho de Banda (BW)</i>	45
2.4.1.3.	<i>Tasa de Codificación (CR)</i>	46
2.4.2.	Radiofrecuencia y Canales	46
2.4.3.	Trama física de LoRa	52

2.4.4.	Tiempo en el aire del paquete (ToA)	53
2.4.5.	Arquitectura de la red LoRa	54
2.5.	Protocolo LoRaWAN	55
2.5.1.	Arquitectura LoRaWAN	55
2.5.2.	Seguridad LoRaWAN	57
2.5.3.	Clases de Dispositivos LoRaWAN	58
2.6.	El Agua.....	60
2.6.1.	Agua Superficial.....	60
2.6.2.	Agua Subterránea	61
2.7.	Captación de Agua.....	61
2.8.	Cambio Climático.....	61
2.9.	Caudales en ríos.....	62
2.10.	Medición del nivel	62
2.10.1.	Medidor de sonda.....	63
2.10.2.	Medidor de cristal.....	63
2.10.3.	Medidor de flotador.....	64
2.10.4.	Medidor manométrico	65
2.10.5.	Medidor de burbujeo	66
2.10.6.	Medidor por radiación.....	67
2.10.7.	Medidor por láser	68

	10
2.10.8. Medidor ultrasónico	69
2.11. Metodología en cascada.....	70
3. Capitulo III: Diseño del Sistema.....	72
3.1. Descripción General del Sistema.....	72
3.2. Situación Actual	74
3.2.1. Ubicación	74
3.2.2. Características	75
3.2.3. Planta de Captación.....	77
3.3. Diseño del Sistema	79
3.4. Diagrama de Bloques del Sistema.....	80
3.5. Requerimientos del Sistema	82
3.5.1. Etapa de sensado	82
3.5.1.1. <i>Análisis del Sensor de Nivel de Agua</i>	82
3.5.1.2. <i>Análisis de las Placas Microcontroladoras</i>	91
3.5.1.3. <i>Análisis de la Comunicación RF mediante LoRa</i>	96
3.5.2. Etapa de control.....	98
3.5.2.1. <i>Análisis de Minicomputadoras para el Nodo Central</i>	99
3.5.3. Etapa de visualización.....	104
3.6. Arquitectura del Sistema	106
3.7. Diagramas de interconexión	107

	11
3.7.1. Nodo Sensor	107
3.7.2. Nodo Gateway LoRa.....	110
3.7.3. Simulación del Armado del Sistema	112
3.8. Diseño del Prototipo	116
3.8.1. Diagrama de flujo del sistema.....	116
3.8.2. Configuración del Nodo Sensor	118
3.8.2.1. Programación ESP32	120
3.8.3. Configuración Gateway LoRa.....	126
3.8.3.1. Programación Arduino UNO	128
3.8.3.2. Programación Raspberry PI 4	133
3.8.4. Configuración ThingSpeak	139
4. Capitulo IV: Implementación y Pruebas de Funcionamiento	143
4.1. Integración de Componentes	143
4.2. Verificación de la Configuración de los Elementos de la Red	146
4.2.1. Verificación del Nodo Sensor	147
4.2.2. Verificación del Gateway LoRa.....	149
4.3. Pruebas de funcionamiento del sistema.....	152
4.3.1. Implementación del Sistema	152
4.3.2. Recolección y Visualización de los Datos del Nodo Sensor.....	154
4.3.3. Recepción de Datos en el Gateway LoRa	158

	12
4.3.4. Envío de Datos a la Plataforma ThingSpeak.....	159
4.4. Verificación y Presentación de los Datos	161
4.4.1. Comunicación entre el Nodo Sensor y el Gateway LoRa.....	161
4.4.2. Recepción y Presentación de los Datos en la Plataforma ThingSpeak	162
4.4.3. Verificación de la Aplicación Móvil ThingSpeak.....	164
4.4.4. Envío del Mensaje de Alerta Mediante Correo Electrónico	166
4.4.5. Patrones de Sequía	169
4.5. Prueba de la comunicación inalámbrica LoRa	176
4.6. Tiempo en el aire (ToA) para la comunicación LoRa	177
CONCLUSIONES Y RECOMENDACIONES	184
Conclusiones.....	184
Recomendaciones	185
REFERENCIAS	186
ANEXOS	190
ANEXO 1 – Código del Nodo Sensor.....	190
ANEXO 2 – Código del Gateway LoRa	192

INDICE DE FIGURAS

Figura 1 <i>Arquitectura IoT</i>	31
Figura 2 <i>Elementos de una WSN</i>	34

	13
Figura 3 <i>Arquitectura de la red WSN</i>	35
Figura 4 <i>Topologías de red WSN</i>	36
Figura 5 <i>Modulación LoRa</i>	43
Figura 6 <i>Plan de canalización para la banda US902-928</i>	52
Figura 7 <i>Formato de la trama física</i>	53
Figura 8 <i>Arquitectura de red LoRa</i>	54
Figura 9 <i>Arquitectura del protocolo LoRaWAN</i>	56
Figura 10 <i>Arquitectura básica de LoRa y LoRaWAN</i>	57
Figura 11 <i>Ranura de recepción de un dispositivo clase A</i>	58
Figura 12 <i>Ranura de recepción de un dispositivo clase B</i>	59
Figura 13 <i>Ranura de recepción de un dispositivo clase C</i>	60
Figura 14 <i>Medidor de sonda</i>	63
Figura 15 <i>Medidor de cristal</i>	64
Figura 16 <i>Medidor de flotador</i>	65
Figura 17 <i>Medidor manométrico</i>	66
Figura 18 <i>Medidor de burbujeo</i>	67
Figura 19 <i>Medidor por radiación</i>	68
Figura 20 <i>Medidor por láser</i>	69
Figura 21 <i>Medidor ultrasónico</i>	70
Figura 22 <i>Metodología en cascada</i>	71

Figura 23 <i>Nodo de monitoreo de nivel de río Negro de la Pontificia Universidad Javeriana</i>	74
Figura 24 <i>Ubicación geográfica de Río Chico en Google Earth</i>	75
Figura 25 <i>Caudal del Río Chico</i>	76
Figura 26 <i>Características de Río Chico</i>	77
Figura 27 <i>Planta de captación de Río Chico</i>	78
Figura 28 <i>Diagrama de bloques del sistema</i>	81
Figura 29 <i>Funcionamiento del Sensor MB7076</i>	89
Figura 30 <i>Funcionamiento del Sensor MB1240 y MB1043</i>	90
Figura 31 <i>Funcionamiento del Sensor HC-RS04</i>	91
Figura 32 <i>Arquitectura del sistema planteado</i>	107
Figura 33 <i>Diagrama de conexión sensor HC-SR04 y ESP32</i>	108
Figura 34 <i>Diagrama de conexión módulo LoRa RFM95 y ESP32</i>	110
Figura 35 <i>Diagrama de conexión del Gateway LoRa</i>	112
Figura 36 <i>Simulación de la disposición de los nodos sensores y el gateway LoRa</i>	113
Figura 37 <i>Vista de los nodos sensores y el gateway LoRa en simulación de Río Chico</i>	114
Figura 38 <i>Simulación de la distribución de los elementos del nodo sensor en el case</i>	115
Figura 39 <i>Simulación de la distribución de los elementos del gateway LoRa en el case</i>	116
Figura 40 <i>Diagrama de flujo del sistema</i>	118
Figura 41 <i>Diagrama de conexión del nodo sensor</i>	119

Figura 42 <i>Comunicación SPI entre el módulo LoRa RFM95 y el microcontrolador ESP32</i>	120
Figura 43 <i>Inclusión de librerías SPI y LoRa</i>	121
Figura 44 <i>Pines de conexión del Módulo LoRa RFM95 a ESP32</i>	122
Figura 45 <i>Variables para la configuración de parámetros de la comunicación LoRa</i>	122
Figura 46 <i>Parámetros de la transmisión LoRa</i>	123
Figura 47 <i>Paquete LoRa para transmitir</i>	123
Figura 48 <i>Pines y variables para el sensor ultrasónico HC-SR04</i>	124
Figura 49 <i>Cálculo de distancia hacia el objeto y valor de nivel</i>	125
Figura 50 <i>Diagrama de flujo del nodo sensor</i>	126
Figura 51 <i>Diagrama de conexión Gateway LoRa</i>	127
Figura 52 <i>Comunicación SPI entre el módulo LoRa RFM95 y Arduino UNO</i>	128
Figura 53 <i>Inclusión de librerías SPI y LoRa</i>	129
Figura 54 <i>Pines de conexión del Módulo LoRa RFM95 a Arduino UNO</i>	129
Figura 55 <i>Variables para la configuración de parámetros de la comunicación LoRa</i>	130
Figura 56 <i>Parámetros de la transmisión LoRa</i>	130
Figura 57 <i>Recepción de los datos de los nodos sensores LoRa</i>	131
Figura 58 <i>Envío por comunicación Serial de los datos receptados por el Gateway LoRa</i>	131
Figura 59 <i>Diagrama de flujo del Gateway LoRa en Arduino UNO</i>	133

Figura 60 <i>Importación de librerías en Python</i>	134
Figura 61 <i>Datos de correo electrónico para envío de alerta</i>	135
Figura 62 <i>Datos para acceso a la plataforma IoT ThingSpeak</i>	135
Figura 63 <i>Lectura y procesamiento de los datos</i>	136
Figura 64 <i>Código para el envío de alertas mediante correo</i>	137
Figura 65 <i>Publicar los datos en ThingSpeak utilizando el protocolo MQTT</i>	138
Figura 66 <i>Diagrama de flujo del envío de datos al servidor ThingSpeak</i>	139
Figura 67 <i>Creación de un nuevo canal en ThingSpeak</i>	140
Figura 68 <i>Creación del dispositivo MQTT</i>	141
Figura 69 <i>Canal y dispositivo MQTT creado en ThingSpeak</i>	142
Figura 70 <i>Nodo LoRa</i>	144
Figura 71 <i>Gateway LoRa</i>	145
Figura 72 <i>Componentes del sistema</i>	146
Figura 73 <i>Selección de la placa en Arduino IDE</i>	147
Figura 74 <i>Inicialización del módulo RFM95 en nodo 1</i>	148
Figura 75 <i>Inicialización del módulo RFM95 en nodo 2</i>	149
Figura 76 <i>Selección de la placa Arduino</i>	150
Figura 77 <i>Inicialización del módulo LoRa en el Gateway</i>	151
Figura 78 <i>Inicialización del módulo LoRa en el Gateway</i>	152
Figura 79 <i>Ubicación y medidas del nodo sensor sobre el río</i>	153

Figura 80 <i>Ubicación del nodo sensor sobre el río</i>	154
Figura 81 <i>Recolección y visualización de los datos del nodo 1</i>	155
Figura 82 <i>Recepción de datos del nodo 1 en el Gateway LoRa</i>	156
Figura 83 <i>Recolección y visualización de los datos del nodo 2</i>	157
Figura 84 <i>Recepción de datos del nodo 2 en el Gateway LoRa</i>	158
Figura 85 <i>Recepción de datos en el Gateway LoRa de los dos nodos</i>	159
Figura 86 <i>Recepción de datos depurados en el Gateway LoRa</i>	160
Figura 87 <i>Envío de datos a la plataforma de ThingSpeak</i>	161
Figura 88 <i>Recepción de datos del nodo sensor en el Gateway</i>	162
Figura 89 <i>Plataforma de visualización</i>	163
Figura 90 <i>Protocolo MQTT</i>	164
Figura 91 <i>Canal en la App Móvil de ThingSpeak</i>	165
Figura 92 <i>Visualización de los datos desde la App móvil de ThingSpeak</i>	165
Figura 93 <i>Visualización del descenso del nivel de agua para envío de alerta</i>	166
Figura 94 <i>Visualización del mensaje de alerta para el descenso del nivel</i>	167
Figura 95 <i>Visualización del aumento del nivel de agua para envío de alerta</i>	168
Figura 96 <i>Visualización del mensaje de alerta para el aumento del nivel</i>	169
Figura 97 <i>Gráfica del nivel del río a lo largo del tiempo en base a datos recolectados</i> .	170
Figura 98 <i>Comparación entre las medidas de incremento en el nivel del río y la medida de nivel referencial</i>	172

Figura 99 <i>Comparación entre las medidas de descenso en el nivel del río y la medida de nivel referencial.</i>	174
Figura 100 <i>Tamaño del payload</i>	178
Figura 101 <i>Calculadora LoRa para el ToA</i>	182

INDICE DE TABLAS

Tabla 1 <i>Factores de propagación de LoRa en la región de América con la banda US902-928 Uplink con ancho de banda de 125 kHz.</i>	44
Tabla 2 <i>Factores de propagación de LoRa en la región de América con la banda US902-928 Downlink con ancho de banda de 500 kHz.</i>	45
Tabla 3 <i>Comparativa de las especificaciones LoRa para Europa y Estados Unidos, regiones en las que se utilizan bandas ISM.</i>	46
Tabla 4 <i>Canales Uplink con ancho de banda de 125 kHz en la banda US902-928.</i>	47
Tabla 5 <i>Canales Uplink con ancho de banda de 500 kHz en la banda US902-928.</i>	50
Tabla 6 <i>Canales Downlink con ancho de banda de 500 kHz en la banda US902-928.</i>	51
Tabla 7 <i>Comparativa sensores industriales para medición de flujo y nivel de caudal</i>	84
Tabla 8 <i>Comparativa entre sensores de flujo y nivel de caudal compatibles con hardware libre</i>	87
Tabla 9 <i>Comparativa entre el microcontrolador ESP32 y Arduino UNO</i>	92
Tabla 10 <i>Comparación entre los módulos LoRa RFM95 y RA02</i>	97
Tabla 11 <i>Comparación entre los LoRa Gateways disponibles en el mercado.</i>	99

Tabla 12 <i>Comparación entre los módulos LoRa RFM95 y RA02</i>	101
Tabla 13 <i>Comparación entre las plataformas IoT disponibles</i>	104
Tabla 14 <i>Detalle de las medidas obtenidas y relación con el patrón de sequía</i>	175
Tabla 15 <i>Comparación distancia vs RSSI</i>	177
Tabla 16 <i>Cálculos teóricos del ToA</i>	180
Tabla 17 <i>Cálculos del ToA mediante la calculadora LoRa</i>	182

RESUMEN

El presente trabajo de titulación está orientado al uso de la tecnología de comunicación inalámbrica LoRa para elaborar una red de sensores que permita la medición del nivel de agua del río, esto permitirá conocer acerca del estado de este e identificar si su nivel incrementa o se reduce mediante el envío de alertas a través del correo electrónico.

El sistema desarrollado posee varios subsistemas acordes a lo planteado en su arquitectura relacionada a las redes IoT, en primer lugar, se tiene los nodos sensores, estos nodos están conformados por el hardware seleccionado utilizando los sensores HC-SR04 para el cálculo del nivel, el módulo LoRa RFM95 para la transmisión inalámbrica utilizando LoRa y estos dos dispositivos se conectan al microcontrolador ESP32 para las tareas de procesamiento. El nodo central, el cual cumple la función de recibir los datos de los nodos, está conformado por un Arduino Uno compatible con el módulo LoRa RFM95 y conectado a una Raspberry PI 4 para el procesamiento y posterior el envío de datos mediante el protocolo MQTT al servidor, con ello se puede acceder a la visualización de la información en la plataforma de ThingSpeak.

Una vez seleccionado el hardware para la elaboración de los diferentes elementos del sistema se realiza la configuración de cada uno de ellos con el software adecuado, para que cumplan con lo propuesto para la red. Finalmente, se elaboran las pruebas para verificar la comunicación que se establece entre los nodos y el Gateway, para lo cual se evidencia que los parámetros de la comunicación LoRa se establecen correctamente en los módulos RFM95 y que los datos se envíen de los nodos al Gateway para la visualización en ThingSpeak, en donde se evidencia que el nivel del río la mayor parte del tiempo tiene un nivel estable entre los 50cm a 60cm en base a las mediciones realizadas por los sensores, por ello, cuando se

detecta un descenso por debajo de los 40cm se envía una alerta por correo indicando este particular, por otro lado, cuando el nivel incrementa se envía una alerta cuando el río sobrepasa los 70cm, teniendo así la información del estado del recurso en el momento.

ABSTRACT

The present degree project is focused on the use of LoRa wireless communication technology to develop a sensor network that allows for the measurement of river water levels. This will provide information about the state of the river and identify whether its level is increasing or decreasing by sending alerts via email.

The developed system consists of various subsystems in accordance with the architecture related to IoT networks. Firstly, there are the sensor nodes, which are composed of selected hardware using HC-SR04 sensors for level calculation, the LoRa RFM95 module for wireless transmission using LoRa technology, and these two devices are connected to the ESP32 microcontroller for processing tasks. The central node, which is responsible for receiving data from the nodes, consists of an Arduino Uno compatible with the LoRa RFM95 module and connected to a Raspberry Pi 4 for processing and subsequent data transmission via the MQTT protocol to the server. This allows access to information visualization on the ThingSpeak platform.

Once the hardware for the different elements of the system is selected, each of them is configured with the appropriate software to meet the proposed requirements for the network. Finally, tests are conducted to verify the communication established between the nodes and the gateway. It is evidenced that the LoRa communication parameters are correctly set in the RFM95 modules and that data is sent from the nodes to the gateway for visualization on ThingSpeak. The measurements obtained from the sensors indicate that the river level remains mostly stable between 50cm to 60cm. Therefore, when a decrease below 40cm is detected, an alert is sent via email indicating this particular situation. On the other hand, an

alert is sent when the river level exceeds 70cm, providing information on the current state of the resource.

1. Capítulo I

1.1. Tema

Diseño de un sistema de medición para el nivel de agua del caudal de la principal fuente de captación de la ciudad de Tulcán utilizando la tecnología de comunicación LoRa.

1.2. Problema

Las fuentes de agua constituyen el principal recurso en el suministro de agua en forma individual o colectiva para satisfacer necesidades de alimentación, higiene y aseo de las personas que integran una localidad. Cabe señalar que es importante seleccionar una fuente adecuada o una combinación de fuentes para dotar de agua en cantidad suficiente a la población. La Empresa Municipal de agua potable de la ciudad de Tulcán capta actualmente agua de las fuentes que pertenecen a la cuenca del río Chico, constituyéndose en la principal fuente de abastecimiento de la ciudad. Según (El Universo, 2021) el descenso de los niveles de los caudales de agua que abastecen a la ciudad se debe a la temporada de sequía en Tulcán y a la afectación en los entornos naturales de estas zonas.

Actualmente, el clima es bastante impredecible y de igual forma no se tiene un monitoreo sobre el nivel de los caudales que abastecen el sistema de agua potable para de esta forma poder tener un límite en el nivel del caudal para un abastecimiento continuo caso contrario poder regular la distribución. La ciudad de Tulcán posee cuatro fuentes de abastecimiento de agua que son Monte Redondo, Cucurucho, Aguas Calientes y río Chico, la principal es río Chico, según (La Hora, 2021) esta fuente de abastecimiento principal tuvo un descenso considerable en su caudal, pasó de 400 a 60 litros por segundo, es un cambio que preocupa tanto a autoridades locales como a los consumidores del servicio.

El monitoreo constante de los caudales que abastecen el sistema de agua potable de la ciudad puede ayudar a obtener información sobre su nivel, por ello, se plantea un diseño de una red de sensores inalámbricos para el monitoreo del nivel de los caudales utilizando transmisión de datos en tiempo real mediante el uso de la tecnología de radio LoRa, que permite transmisiones de radio encriptadas para largas distancias consumiendo poca energía, con ello permitirá indicar los niveles de los caudales para poder tomar decisiones inmediatas en cuanto a reserva del líquido.

El sistema lo que puede proporcionar son datos en tiempo real del nivel del caudal tomados mediante una red de sensores inalámbricos, los cuales utilizan tecnología de radio LoRa para transmisiones seguras y de largas distancias, de esta forma la entidad encargada del abastecimiento y distribución del agua pueda accionar medidas para no desabastecer en gran medida a la población.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar un sistema de medición del nivel de agua del caudal de la principal fuente de captación de la ciudad de Tulcán mediante el uso de redes de sensores inalámbricos utilizando la tecnología de comunicación LoRa para la recolección y transmisión de datos con el fin de identificar patrones de sequía a lo largo del tiempo.

1.3.2. Objetivos Específicos

- Analizar mediante la recopilación de información bibliográfica acerca de los sistemas de medición del nivel de agua de caudales y la tecnología de comunicación LoRa, para determinar las bases teóricas del tema.

- Diseñar una red de sensores inalámbricos utilizando la tecnología LoRa y la plataforma de visualización para la recolección y transmisión de información del nivel de agua del caudal.
- Implementar la red de sensores inalámbricos utilizando el hardware y software más adecuado en base a los requerimientos del sistema para obtener información del estado del recurso.
- Realizar pruebas de funcionamiento del sistema de medición y de la red de comunicación implementados con la tecnología LoRa para la corrección de errores del sistema.

1.4. Alcance

El presente proyecto tiene como objetivo el desarrollo de un sistema de medición del nivel del caudal de Rio Chico, principal fuente de captación de agua de la ciudad de Tulcán, el sistema permitirá realizar lecturas del nivel de agua del caudal, con los datos obtenidos a lo largo del tiempo se podrá identificar un patrón de sequía. La información obtenida de la red de sensores será accesible mediante plataformas y herramientas de presentación, con el libre acceso a esta información, quienes hacen uso del recurso pueden tomar conciencia y racionar el uso de agua potable.

Para el proyecto se utilizará el modelo en cascada, el cual es un procedimiento lineal caracterizado por dividir los procesos de desarrollo en sucesivas fases de proyecto. Las fases consisten en el análisis, diseño, implementación, verificación y servicio. (IONOS Digitalguide, 2022)

En la fase de análisis, se recopilará información bibliográfica acerca de los diferentes sistemas existentes para la medición del nivel de agua de caudales, de esta forma se podrá comprender los diferentes métodos de medición y su respectivo funcionamiento, con ello se aplicará al sistema el método más adecuado con el que se podrá medir el nivel de agua del caudal. Además, para determinar los elementos y herramientas tecnológicas que serán mejores y eficientes para el sistema planteado, es necesario realizar un estudio acerca de la tecnología de comunicación, en este caso la tecnología de comunicación LoRa que utiliza el protocolo de red LoRaWAN para redes de baja potencia y área amplia, entonces se determinará los requerimientos de hardware y software acorde a las necesidades del tema planteado.

En la fase de diseño, se desarrollará un sistema de monitoreo que consiste en una red de sensores inalámbricos para la recopilación y transmisión de información acerca del nivel del caudal de agua, este utilizará la tecnología de comunicación determinada en base a la ubicación geográfica y situación actual del terreno, teniendo en cuenta que se requiere transmisión a largas distancias para poder llevar la información hacia un nodo central. Los sensores formarán parte de una red de sensores inalámbricos que se conectaran a una placa controladora, la cual procesará la información y la enviará a un nodo central el cual se encargará de la gestión para su posterior visualización en una plataforma.

En la fase de implementación, se pretende desplegar la red de sensores inalámbricos que utilizan la tecnología de comunicación LoRa y su protocolo de red LoRaWAN en la ubicación geográfica correspondiente en la cual se requiere realizar la obtención de datos acerca del nivel de agua del caudal. Se realizará la respectiva configuración de los sensores conectados a un NodeMCU en donde se incorporará un módulo LoRa para tener la capacidad

de utilizar esta tecnología de comunicación inalámbrica, posteriormente el nodo central consistirá en un Gateway LoRa conformado por el dispositivo Raspberry PI y un módulo LoRa, al cual se enviarán los datos recolectados por los sensores, este se encargará de dirigir la información hacia la nube, en este caso hacia una plataforma IoT, la cual mediante sus herramientas pueda realizar el procesamiento de los datos recolectados y permita la visualización para el acceso de los usuarios finales a la información.

En la fase de verificación y servicio del sistema, se pretende realizar las respectivas pruebas de funcionamiento de la red de sensores inalámbricos desplegada y de la tecnología de comunicación utilizada, en este caso LoRa, de esta forma se podrá identificar fallos e inconvenientes que se puedan presentar debido a las condiciones geográficas del área donde se pretende desplegar el sistema, con ello se podrá mejorar la eficiencia y funcionalidad del sistema planteado.

1.5. Justificación

Los cambios climáticos, el consumo inadecuado en los hogares, el aumento y desarrollo de la población, son algunos de los factores que causan un mayor gasto del agua en la actualidad. El impacto en los ecosistemas a consecuencia del cambio climático alrededor de las fuentes de captación de agua para el servicio de la ciudadanía se ha visto afectado, ya que el descenso en el nivel del caudal de abastecimiento ha sido notable y un claro factor para el desabastecimiento del recurso hídrico. (Granados Quitian & Pérez Ombita, 2017)

En la actualidad, la necesidad de contar con un sistema que permita monitorear y visualizar el nivel de agua del caudal puede resultar de gran ayuda para identificar épocas de sequía. En algunos casos no se tiene conocimiento del nivel del caudal de la fuente de

abastecimiento, por ello, no se puede conocer ni prevenir el impacto en la distribución del recurso hídrico debido a su descenso. Este proyecto realizará el estudio del nivel del caudal de agua de Rio Chico, el cual permitirá identificar patrones de sequía, con ello, se podría ayudar al ente encargado del servicio de distribución de agua y a la ciudadanía a conocer acerca de esto para la toma de decisiones y la concientización acerca del uso de este recurso.

Por lo expuesto, el objetivo de este proyecto es contar con un sistema de medición del nivel del caudal de agua en la fuente Rio Chico, el cual permitirá el proceso de recolección de información de las mediciones, para informar de forma automática acerca del nivel del caudal. Adicionalmente, se presentarán las mediciones realizadas en tiempo real, para de esta forma tener un conocimiento al instante de dicha variable.

2. Capítulo II: Fundamentos Teóricos

El presente capítulo consiste en el análisis de los conceptos teóricos a partir de los cuales se realiza el desarrollo de esta investigación.

2.1. Internet de las Cosas

“Una red abierta y completa de objetos inteligentes conectados a Internet que tienen la capacidad de auto-organizarse, compartir información, datos y recursos, reaccionar y actuar frente a situaciones y cambios en el entorno” (Aguilar Alvarez & Hinojosa Altamirano, 2019). Debido al alto crecimiento del uso de internet las aplicaciones y beneficios que ha generado este ha sido considerable, en este caso con el Internet de las Cosas o IoT, cuyo objetivo principal es permitir la interconexión entre diferentes dispositivos, los cuales intercambian información según los requerimientos del usuario, los dispositivos que pueden utilizarse para ello pueden ser: sensores, computadoras, teléfonos inteligentes o cualquier dispositivo capaz de acceder a la red y generar algún tipo de información. Otro punto importante en el internet de las cosas es que se busca que cualquier objeto pueda integrarse a la red y transmita información de forma automatizada sin necesidad de la intervención humana, permitiendo de esta forma automatizar procesos y mantener el control de los dispositivos u objetos obteniendo datos continuos acerca de estos.

2.1.1. Arquitectura IoT

A continuación, se presenta la arquitectura IoT que consiste en cuatro capas, las cuales ayudan a cumplir requerimientos para que esta tecnología pueda operar de forma correcta y permitir que los objetos puedan interactuar entre sí, además de permitir escalabilidad, flexibilidad, robustez, eficiencia y seguridad.

La arquitectura IoT consta de cinco capas las cuales son:

- Capa de aplicación.
- Capa de Nube/Centro de Datos.
- Capa de red.
- Capa de Gateway.
- Capa de Dispositivos.

Figura 1

Arquitectura IoT



Fuente. (Benítez, D. & Anias, C., 2016)

- **Capa de Dispositivos:** se encuentra en la base de la arquitectura y es la encargada de varias funciones, relacionadas fundamentalmente con el medio físico en el cual se encuentran los dispositivos IoT. Es la capa física, que incorpora sensores para detectar y recopilar información sobre el entorno. Detecta algunos parámetros físicos o reconoce otros objetos inteligentes en el entorno. Entre sus funciones está la de sensar, actuar, generar peticiones, responder a peticiones, entre otras.

- **Capa de Gateways:** está formada por los nodos IoT que intervienen en la comunicación entre algunos dispositivos y la Capa de Red, en la capa de gateways se soportan dispositivos conectados mediante diferentes tipos de tecnologías alámbricas e inalámbricas.
- **Capa de Red:** se consideran las diferentes redes que dan soporte a los sistemas IoT, en este caso lo fundamental es el internet, ya que es la red que soportará todo el sistema IoT, y es la responsable de la conexión de las cosas inteligentes, dispositivos de red y servidores.
- **Capa de Nube/Centro de Datos:** esta capa incluye varias funciones relacionadas con el procesamiento de los datos, esta capa cumple con diferentes funciones relacionadas con los datos obtenidos de las capas inferiores, se realiza la validación de los datos como medida de integridad, transformación de los datos, los cuales se almacenan y preparan para la capa aplicación, filtrado de datos que corresponde a la separación de los datos acorde a su utilidad y el procesamiento y almacenamiento con el fin de generar información útil.
- **Capa Aplicación:** brinda servicios específicos de la aplicación al usuario, esta capa contiene las aplicaciones IoT.

2.2. Redes de Sensores Inalámbricos (WSN)

“Una red de sensores inalámbricos (WSN) consiste en dispositivos autónomos que se encuentran para monitorear condiciones físicas o ambientales” (Santos Benavides, 2019). Las Wireless Sensor Networks o Redes de Sensores Inalámbricos, actualmente se han convertido en una tecnología útil que ayuda en la recolección de ciertas variables en un entorno determinado, en este caso con la recolección de datos se puede obtener información

relevante acerca de un evento en particular. Este tipo de redes constan de elementos que se interconectan utilizando tecnologías inalámbricas, en este caso un Gateway y nodos distribuidos en la zona.

Las redes de sensores inalámbricos se forman por dispositivos autónomos, los cuales se distribuyen en una zona determinada y como se había mencionado anteriormente, principalmente permiten la recolección de información en base al monitoreo y medición de parámetros físicos o ambientales tales como: temperatura, sonido vibración, gases, agentes contaminantes, niveles de líquidos, distancias, entre otros. (Camacho , Rodriguez, Bajaña, Jonathan, & William, 2019). Las WSN se conforman de un gran número de nodos de sensores los cuales funcionan de manera autónoma, ya que acompañados con un microcontrolador programable pueden ejecutar algoritmos para realizar el proceso de obtención de datos en base al entorno de despliegue y, por otra parte, el Gateway es aquel dispositivo al cual todos los nodos sensores, haciendo uso de una tecnología inalámbrica, envían los datos recogidos para su posterior procesamiento.

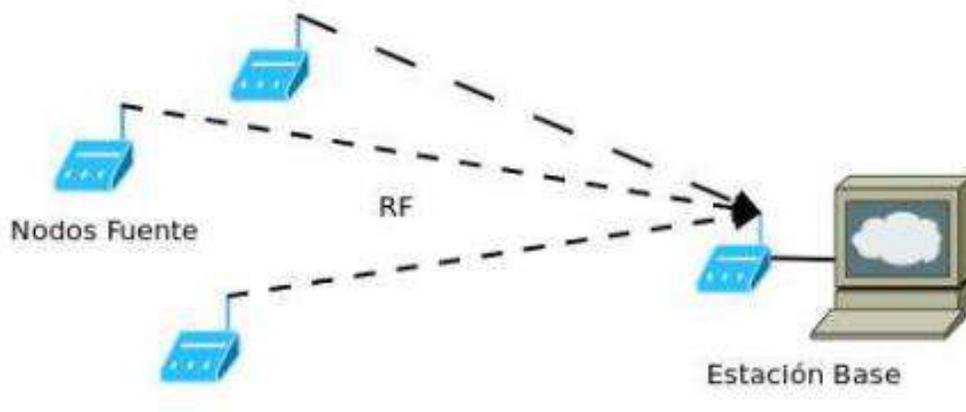
2.2.1. Elementos de las WSN

Las redes de sensores inalámbricos están compuestas por diferentes elementos, entre estos están los nodos, los cuales corresponden a dispositivos como microcontroladores, sensores, que a su vez utilizan algún tipo de conexión. El elemento principal de la red WSN se denomina estación, es el elemento al cual se comunican todos los nodos, recibe información de ellos y se encarga de coordinar el funcionamiento de la red. Finalmente, en una maquina o sistema embebido los datos recolectados, se almacenan, se envían hacia internet y son mostrados al usuario final. (Gordillo Gordillo, 2021)

En la Figura 2, se puede observar una red de sensores inalámbricos con sus respectivos elementos, que corresponden a los nodos sensores y la estación base.

Figura 2

Elementos de una WSN



Fuente. (Iacono, Godoy, Marianetti, Garcia Garino, & Párraga, 2012)

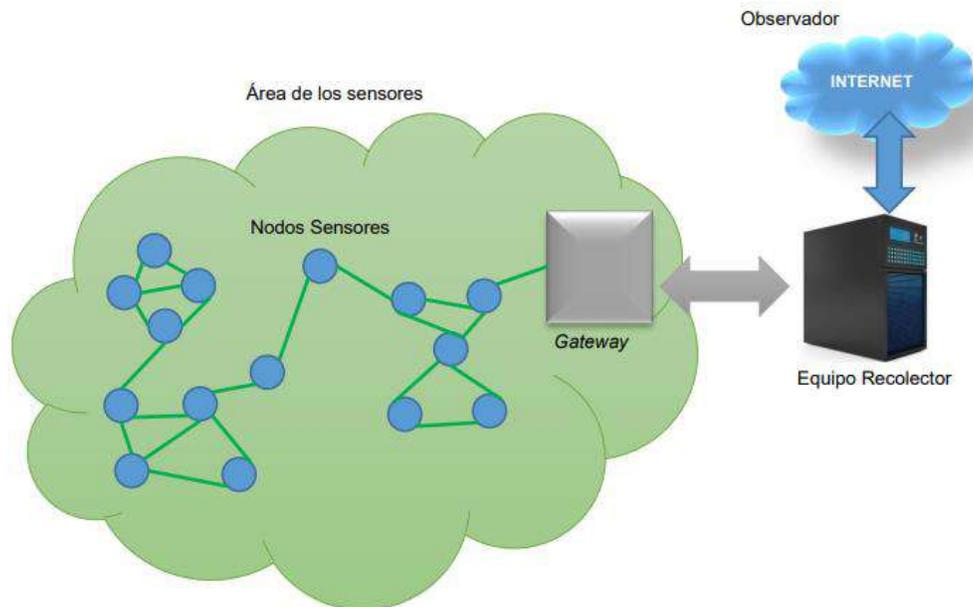
2.2.2. Arquitectura de las WSN

Las redes de sensores inalámbricos corresponden a un conjunto de nodos, los cuales se despliegan en una zona específica en la cual se requiera atención o monitoreo, los nodos sensores recolectan información acerca de una variable física medible de su entorno y esta se envía a través de los nodos para posteriormente llegar al nodo Gateway que finalmente se conecta al equipo recolector, en donde la información recibida es tratada y mediante una conexión a internet, puede llevar dicha información a un sistema mucho más grande o simplemente presentarla de forma visual. (Muñoz & Morales, 2018)

En la Figura 3, se presenta la arquitectura de una red de sensores inalámbrica que se compone por los nodos sensores, Gateway, servidor, internet y usuario final.

Figura 3

Arquitectura de la red WSN



Fuente. (Muñoz & Morales, 2018)

2.2.3. Topologías de Red

Las topologías de red consisten en la forma de agrupación de los nodos con el objetivo de buscar la mejor ubicación de estos, para así poder enviar y recibir información de la red de la mejor manera. Las redes WSN tienen tres tipos de topología: estrella, árbol y malla.

2.2.3.1. Estrella

En este tipo de topología los nodos sensores se ubican de tal forma que todos se conectan a un único elemento central conocido como Gateway, computadora o conmutador, el cual se encarga de recibir la información. Los nodos se consideran clientes y el elemento central servidor.

2.2.3.2. *Árbol*

En esta topología, cada uno de los nodos se conectan a un nodo de mayor categoría, posteriormente llegan a la comunicación con el Gateway, es decir, los nodos más generales son los únicos encargados de comunicarse con el elemento central de la red.

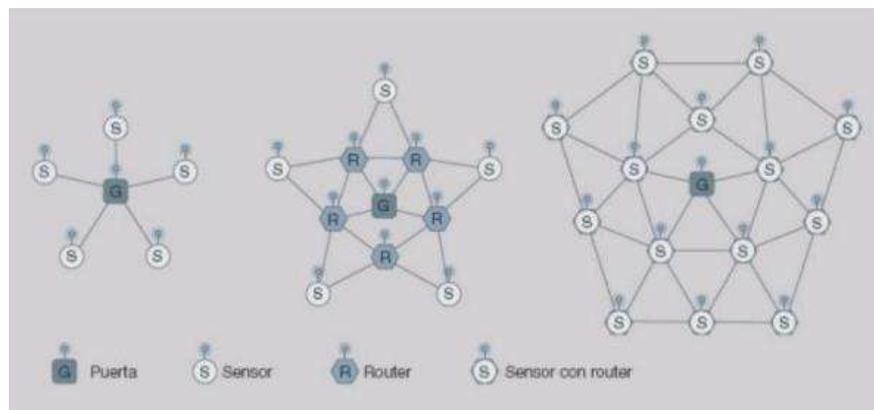
2.2.3.3. *Malla*

En este tipo de topología, cada uno de los elementos de la red están conectados a todos los demás elementos entre sí. Una de sus características principales es que un paquete puede tomar distintos caminos para llegar a su destino y no puede haber ninguna interrupción en la comunicación debido a que está completamente conectada. (Gordillo Gordillo, 2021)

En la Figura 4, se muestran cada una de las topologías de red mencionadas, en el siguiente orden: estrella, árbol y malla.

Figura 4

Topologías de red WSN



Fuente. (Gordillo Gordillo, 2021)

2.3. Tecnologías de Comunicación Inalámbrica

“El uso de las comunicaciones inalámbricas se ve reflejado en el avance tecnológico que ha tenido la sociedad” (Ortiz Soza, 2020, pág. 27). La comunicación inalámbrica ha estado

presente desde la creación del telégrafo, aún más fácil de recordar con la utilización de dispositivos como la radio y televisión, con el auge del internet y desarrollo de tecnologías de comunicación actuales, la comunicación inalámbrica ha presentado un desarrollo y constituye actualmente una tecnología muy empleada e importante dentro de las telecomunicaciones.

En la comunicación inalámbrica, la comunicación entre dispositivos se da a través del medio aire, de esta forma no se requiere el despliegue de cableado para interconectar los equipos, pero es necesario hacer uso del espectro electromagnético, en algunos casos se utiliza el espectro licenciado y en otros el espectro libre, esto depende de los requerimientos y la tecnología utilizada.

Las tecnologías de comunicación inalámbrica han incrementado su uso y posición dentro de la industria, pero esto depende de los requerimientos y necesidades de la red (Garrido Terán, 2021). Dentro de las opciones comerciales más utilizadas se pueden encontrar las tecnologías ZigBee, GSM/GPRS, WiFi, WiMax, entre otras.

En la actualidad, el internet de las cosas ha tomado gran relevancia, ya que este promete un mundo completamente automatizado, permitiendo la conexión a la red de miles de dispositivos capaces de recolectar información, además involucra la reducción de costos, tamaños y consumo de energía. (Maldonado Borda & Pineda Cusba, 2016). En sistemas actuales se implementan tecnologías inalámbricas tales como: las redes celulares, ya sea 3G o 4G, las que tienen una amplia cobertura, pero su utilización implica elevados costos, por otra parte, tecnologías como Bluetooth y Wi-Fi presentan un uso libre, pero tiene un limitado alcance, ya que trabajan a altas frecuencias. La tecnología para destacar es LoRa, la cual es una tecnología reciente y cumple los requerimientos en cuanto a las redes de sensores

inalámbricos, presenta bajo consumo de energía, costos reducidos y una capacidad de comunicación a largas distancias encontrándose así dentro de las Low Power Wide Area Network (LPWAN), por ello se ha tomado en cuenta esta tecnología en la realización del presente proyecto. (Maldonado Borda & Pineda Cusba, 2016)

2.3.1. Redes de Baja Potencia y Área Amplia (LPWAN)

En este caso, centrándonos en las necesidades de comunicación de una red de sensores inalámbricos, hay que tener en cuenta las tecnologías que permiten mantener una comunicación entre elementos físicos con baja potencia, bajos costos y alto rango de cobertura. “Las LPWAN (acrónimo en inglés por Low Power Wide Area Network) son un grupo de tecnologías inalámbricas utilizadas para ofrecer eficientemente conectividad IoT a bajo coste en amplias áreas geográficas” (Wireless Logic., 2022). Las LPWAN o redes de baja potencia y área amplia, como su nombre lo indica, son redes las cuales están orientadas al despliegue de soluciones IoT, poseen un ancho de banda menor y son utilizadas principalmente para la baja transmisión de datos a largas distancias, como lo es en las redes de sensores inalámbricos, donde se requiere que los dispositivos tengan un diseño simplificado, bajo costo y permitan una larga duración de sus baterías debido al consumo energético menor.

“LPWAN se concentra en la conectividad IoT para dispositivos (finales) que consumen poca energía, envían y reciben mensajes cortos a velocidades bajas y cuentan con ciclos de trabajo cortos” (Cantuña Escalante & Satián Guamán, 2019, pág. 20). LPWAN provee un bajo consumo de energía y un rango de cobertura de entre 5 a 40 km, además de los costos reducidos de sus equipos, entre las tecnologías LPWAN que pueden brindar sus servicios en el espectro sin licencia se encuentran SigFox y LoRa.

Las dos tecnologías como SigFox y LoRa, mencionadas anteriormente, tienen su funcionamiento en las bandas sin licencia, las cuales están definidas mediante el Instituto Europeo de Normas de Telecomunicaciones (ETSI) y la Unión Internacional de Telecomunicaciones, entonces, estas bandas se definen como bandas ISM, que hacen referencia a bandas industriales, científicas y médicas, por lo que están reservadas a nivel mundial para su uso no comercial, es decir, no requieren ningún pago por su uso y están abiertas a todo el mundo. (Nrváez Narvárez & Contreras Pérez, 2020)

Las redes LPWAN suelen estar diseñadas para conectar dispositivos con los siguientes requerimientos: (Durand, Visagie, & Booyesen, 2019)

- Comunicación unidireccional (dispositivo a gateway).
- Bajo rendimiento de datos, normalmente del orden de 10 bytes por hora.
- Bajo uso de energía. Estos dispositivos normalmente funcionan con baterías, con una vida útil esperada de la batería de diez años.
- Bajo costo, para permitir la implementación a gran escala.

2.3.1.1. SigFox

La tecnología SigFox corresponde al desarrollo de un grupo de Francia, quienes se centran en una red dedicada para comunicaciones de baja velocidad, además aporta en la reducción del consumo de energía de los dispositivos y el coste de estos. “Los dispositivos SigFox también permiten demodular la señal con una pequeña potencia recibida, la sensibilidad es de hasta -142 dBm y permite un alcance de 30 a 50 km en espacios abiertos” (Nrváez Narvárez & Contreras Pérez, 2020, pág. 35). En cuanto a la topología, la red SigFox permite una configuración de dispositivos en forma de estrella, lo que reduce los gastos de

implementación, de igual forma funciona en el espectro de frecuencias no licenciado o bandas ISM.

2.3.1.2. LoRa

La tecnología LoRa fue creada e impulsada por la asociación LoRa Alliance, la cual tiene como objetivo las comunicaciones de baja potencia y transmisión de datos. Además, debido a las técnicas de capa física que utiliza es posible cubrir una zona amplia en términos de kilómetros. (Narváez Narváez & Contreras Pérez, 2020)

En el presente proyecto es la tecnología seleccionada para el diseño de la red de sensores inalámbricos, ya que ofrece ciertas ventajas frente a su principal competidor, que es SigFox, a diferencia de SigFox, en LoRa sus módulos estándar pueden ser tanto transmisores como receptores y no tienen restricción en la transferencia de datos.

2.4. Tecnología de Comunicación LoRa

“LoRa es una tecnología inalámbrica (al igual que WiFi, Bluetooth, LTE, SigFox o Zigbee) que emplea un tipo de modulación en radiofrecuencia patentado por Semtech, una importante empresa fabricante de chips de radio” (CATSENSORS, 2021). La empresa encargada del desarrollo de esta tecnología se denomina Semtech, y está relacionada con la fabricación de los chips que funcionan con esta tecnología.

La abreviatura LoRa, corresponde a largo alcance, lo que nos indica una de sus principales características, esto se logra haciendo uso de una técnica de modulación de espectro ensanchado denominada la tecnología chirp de espectro ensanchado o CSS. (SEMTECH, 2022)

La tecnología LoRa hace uso del espectro de frecuencias de banda libre, es decir, no supone costo alguno el uso de la banda de frecuencias en las que trabaja, esta banda es la denominada Banda ISM, la cual es de libre acceso al nivel mundial y en las que trabajan tecnologías como Bluetooth y Wi-Fi. Principalmente, LoRa trabaja en la banda sub-GHz y juntamente con la modulación CSS le permite alcanzar altos rangos de cobertura, mayores a 10 kilómetros.

La tecnología LoRa es ideal para trabajar en redes de sensores inalámbricos y el internet de las cosas, debido a que permite implementar una red de área amplia y baja potencia, por lo que es posible mantener cientos de dispositivos conectados mediante enlace inalámbrico hacia un concentrador o Gateway, este es el dispositivo encargado de comunicarse con todos los nodos de la red y gestionar el tráfico proveniente de los mismos, de esta forma se puede compartir la información hacia el destino requerido. (Maldonado Borda & Pineda Cusba, 2016)

LoRa permite el tráfico de datos de tres maneras distintas: (Ortiz Soza, 2020)

- Simplex: Se trata de una comunicación unidireccional, donde un radio en la red transmite datos y los otros N radios reciben los datos.
- Half dúplex: Los radios de la red pueden transmitir y recibir datos, pero nunca de manera simultánea.
- Full dúplex: Los radios de la red pueden transmitir y recibir datos, de manera aleatoria o simultánea.

2.4.1. Modulación LoRa

“LoRa utiliza espectro ensanchado de chirp (CSS) como modulación para transferir datos del transmisor al receptor. CSS utiliza una señal sinusoidal (chirps), que tiene una variación lineal en frecuencia a lo largo del tiempo, para codificar datos” (Durand, Visagie, & Booyesen, 2019, pág. 2).

La tecnología LoRa utiliza una modulación especial, la cual se considera la variación de la modulación DSSS o espectro ensanchado por secuencia directa, esta modulación es la denominada Chirp Spread Spectrum o CSS que proporciona señales robustas ante ruido e interferencia. (Ortiz Soza, 2020)

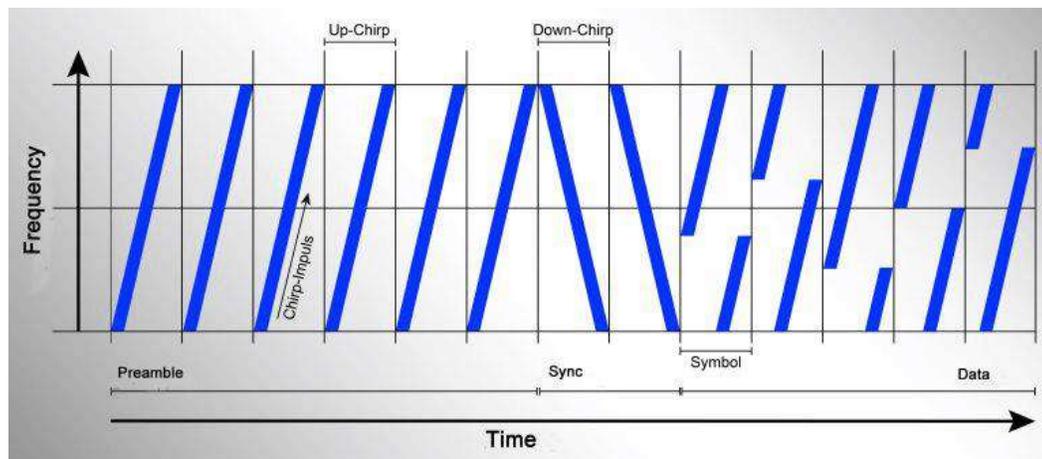
La modulación Chirp Spread Spectrum (CSS), que permite la comunicación de datos de largo alcance. CSS es una técnica de espectro ensanchado que utiliza pulsos de chirp modulados en frecuencia lineal de banda ancha para codificar la información. Las señales chirp tienen amplitud constante con una frecuencia variable. Si la frecuencia cambia de menor a mayor, se denomina up-chirp y si la frecuencia cambia de mayor a menor, se denomina down-chirp.

Los pulsos chirp se envían como símbolos, que aumentan o disminuyen la frecuencia de LoRa de forma continua a lo largo del tiempo. Cuando se procesa un mensaje LoRa, se transmite el preámbulo largo de chirrido constante para lograr la detección de la señal. El final del preámbulo es seguido por dos chirridos inversos utilizados para la sincronización de tiempo y posteriormente se transmite los datos mediante señales moduladas que saltan entre frecuencias.

A continuación, en la Figura 5 se observa la modulación LoRa representada en tiempo y frecuencia en donde se observan los pulsos de chirp, que son diferentes para la transmisión del preámbulo, la sincronización y la transmisión de los datos.

Figura 5

Modulación LoRa



Fuente. (Ortiz Soza, 2020)

El rango de cobertura de LoRa depende altamente de la configuración de la modulación que se defina, ya que esta posee tres parámetros principales los cuales pueden ser configurados por el transmisor, estos son: el factor de dispersión (SF), ancho de banda (BW) y la tasa de codificación (CR). (Ortiz Soza, 2020)

2.4.1.1. Factor de Dispersión (SF)

Establece el número de bits utilizados para codificar un símbolo. La velocidad de transmisión se encuentra relacionada con el factor de dispersión de forma directa, de manera que si el valor de este incrementa se logrará un máximo alcance con baja velocidad. (Ortiz Soza, 2020)

La modulación LoRa tiene posee un total de seis factores de dispersión o SF, que van desde SF7 a SF12, cuanto mayor sea el valor del factor de dispersión utilizado la señal podrá viajar más lejos y ser recibida en el receptor sin errores, pero esto implica que el tiempo en el aire del paquete se incremente, teniendo así una velocidad de transmisión menor.

En la Tabla 1, se muestra para los valores de spreading factor, el respectivo payload máximo y bitrate para el enlace Uplink, en la región en la que se trabaja, que corresponde a América, haciendo uso de la banda US902-928.

Tabla 1

Factores de propagación de LoRa en la región de América con la banda US902-928

Uplink con ancho de banda de 125 kHz.

Spreading Factor	Ancho de Banda	Bit Rate	Tamaño Máximo Payload	Tiempo en el aire
SF7	125 kHz	5470 bps	242 bytes	399.6 ms
SF8	125 kHz	3125 bps	125 bytes	399.9 ms
SF9	125 kHz	1760 bps	53 bytes	390.1 ms
SF10	125 kHz	980 bps	11 bytes	370.7 ms
SF8	500 kHz	12500 bps	242 bytes	178.77 ms

Fuente. (LoRa Alliance®, 2017)

En la Tabla 2, se muestra para los valores de spreading factor, el respectivo payload máximo y bitrate para el enlace Downlink haciendo uso de la banda US902-928.

Tabla 2

Factores de propagación de LoRa en la región de América con la banda US902-928

Downlink con ancho de banda de 500 kHz.

Spreading Factor	Ancho de Banda	Bit Rate	Tamaño Máximo Payload	Tiempo en el Aire
SF7	500 kHz	21900 bps	222 bytes	92.2 ms
SF8	500 kHz	12500 bps	222 bytes	164 ms
SF9	500 kHz	7000 bps	222 bytes	292.1 ms
SF10	500 kHz	3900 bps	222 bytes	533 ms
SF11	500 kHz	1760 bps	109 bytes	554 ms
SF12	500 kHz	980 bps	33 bytes	493.6 ms

Fuente. (LoRa Alliance®, 2017)

Al incrementar el valor del factor de dispersión se puede alcanzar distancias más largas, pero, por el contrario, se reduce la velocidad de transmisión ya que se incrementa el tiempo en el aire del paquete, por lo tanto, un SF10 permitirá alcanzar mayores distancias que un SF7, pero SF7 tendrá una mayor velocidad de transmisión.

2.4.1.2. Ancho de Banda (BW)

Corresponde al rango de frecuencias utilizado para transmisión y se relaciona con la velocidad y sensibilidad, a mayor ancho de banda mayor velocidad, pero se incrementa el ruido. (Ortiz Soza, 2020)

2.4.1.3. Tasa de Codificación (CR)

Relación de corrección de errores hacia adelante que ofrece protección contra interferencias, se relaciona con la velocidad, cuanto mayor es la tasa de codificación, menor es la velocidad de datos. (Ortiz Soza, 2020)

2.4.2. Radiofrecuencia y Canales

LoRa opera en el espectro de frecuencias no licenciado, es por ello, que hace uso de la banda ISM. Principalmente, LoRa utiliza frecuencias por debajo de 1 GHz, lo que se refiere a que trabaja en el rango de sub-GHz, es por ello, por lo que permite la comunicación a grandes distancias. En la Tabla 3, puede observarse la comparativa en cuanto a las características de frecuencia, canal y ancho de banda para las regiones de Europa y América.

Tabla 3

Comparativa de las especificaciones LoRa para Europa y Estados Unidos, regiones en las que se utilizan bandas ISM.

	Europa	Estados Unidos
Banda de Frecuencia	867-869 MHz	902-928 MHz
Canales	10	64+8+8
Canal banda ancha ascendente	125/250 kHz	125/500 kHz
Canal banda ancha descendente	125 kHz	500 kHz
Tx encendido	+14 dBm	+20 dBm tip (+30 dBm permitidos)
Tx desconectar	+14 dBm	+27 dBm
SF Up	7-12	7-10

Velocidad de Datos	250 bps – 50 kbps	980 bps – 21.9 kbps
Link Budget Up	155 dB	154 dB
Link Budget Dn	155 dB	157 dB

Fuente. (Pickering, 2017)

LoRa utiliza diferentes frecuencias definidas para cada región, estas frecuencias principalmente se encuentran dentro de la banda ISM, conocida como una banda libre para su uso, ya que no posee restricciones ni costo para su uso, en América latina y en Ecuador se utiliza la banda US902-928 MHz, el mismo que para Estados Unidos. La banda US902-928 se divide en el siguiente plan de canalización:

- En uplink: 64 canales de 125 kHz de ancho de banda, iniciando desde 902,3 MHz con un incremento lineal de 200 kHz hasta 914,9 MHz. En la Tabla 4, se presenta el plan de canalización para uplink, se muestra el número de canal, el ancho de banda y la frecuencia para cada uno.

Tabla 4

Canales Uplink con ancho de banda de 125 kHz en la banda US902-928.

Canal	Tipo	Ancho de banda (kHz)	Frecuencia (MHz)
0	Uplink	125	902,3
1	Uplink	125	902,5
2	Uplink	125	902,7
3	Uplink	125	902,9
4	Uplink	125	903,1

5	Uplink	125	903,3
6	Uplink	125	903,5
7	Uplink	125	903,7
8	Uplink	125	903,9
9	Uplink	125	904,1
10	Uplink	125	904,3
11	Uplink	125	904,5
12	Uplink	125	904,7
13	Uplink	125	904,9
14	Uplink	125	905,1
15	Uplink	125	905,3
16	Uplink	125	905,5
17	Uplink	125	905,7
18	Uplink	125	905,9
19	Uplink	125	906,1
20	Uplink	125	906,3
21	Uplink	125	906,5
22	Uplink	125	906,7
23	Uplink	125	906,9
24	Uplink	125	907,1
25	Uplink	125	907,3
26	Uplink	125	907,5
27	Uplink	125	907,7

28	Uplink	125	907,9
29	Uplink	125	908,1
30	Uplink	125	908,3
31	Uplink	125	908,5
32	Uplink	125	908,7
33	Uplink	125	908,9
34	Uplink	125	909,1
35	Uplink	125	909,3
36	Uplink	125	909,5
37	Uplink	125	909,7
38	Uplink	125	909,9
39	Uplink	125	910,1
40	Uplink	125	910,3
41	Uplink	125	910,5
42	Uplink	125	910,7
43	Uplink	125	910,9
44	Uplink	125	911,1
45	Uplink	125	911,3
46	Uplink	125	911,5
47	Uplink	125	911,7
48	Uplink	125	911,9
49	Uplink	125	912,1
50	Uplink	125	912,3

51	Uplink	125	912,5
52	Uplink	125	912,7
53	Uplink	125	912,9
54	Uplink	125	913,1
55	Uplink	125	913,3
56	Uplink	125	913,5
57	Uplink	125	913,7
58	Uplink	125	913,9
59	Uplink	125	914,1
60	Uplink	125	914,3
61	Uplink	125	914,5
62	Uplink	125	914,7
63	Uplink	125	914,9

Fuente. (LoRa Alliance®, 2017)

- En uplink: 8 canales de 500 kHz de ancho de banda, iniciando desde 903 MHz con un incremento lineal de 1.6 MHz hasta 914.2 MHz. En la siguiente tabla, se muestra el número de canales para uplink, el ancho de banda y la frecuencia para cada uno.

Tabla 5

Canales Uplink con ancho de banda de 500 kHz en la banda US902-928.

Canal	Tipo	Ancho de banda (kHz)	Frecuencia (MHz)
--------------	-------------	---------------------------------	-----------------------------

64	Uplink	500	903
65	Uplink	500	904,6
66	Uplink	500	906,2
67	Uplink	500	907,8
68	Uplink	500	909,4
69	Uplink	500	911
70	Uplink	500	912,6
71	Uplink	500	914,2

Fuente. (LoRa Alliance®, 2017)

- En downlink: 8 canales de 500 kHz de ancho de banda, iniciando desde 923.3 MHz con un incremento lineal de 600 kHz hasta 927.5 MHz. En la Tabla 6, se muestra el número de canal, el ancho de banda y la frecuencia para downlink.

Tabla 6

Canales Downlink con ancho de banda de 500 kHz en la banda US902-928.

Canal	Tipo	Ancho de banda (kHz)	Frecuencia (MHz)
0	Downlink	500	923,3
1	Downlink	500	923,9
2	Downlink	500	924,5
3	Downlink	500	925,1
4	Downlink	500	925,7
5	Downlink	500	926,3

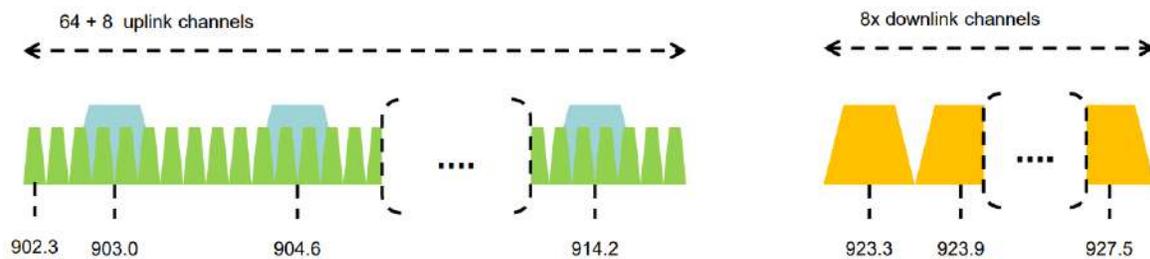
6	Downlink	500	926,9
7	Downlink	500	927,5

Fuente. (LoRa Alliance®, 2017)

En la Figura 6, se observa en forma gráfica la distribución de frecuencias con la canalización en el espectro de frecuencias, esto se visualiza para los canales de Uplink y Downlink, cada canal está asociado a su respectiva frecuencia y se verifica con las tablas anteriormente mostradas .

Figura 6

Plan de canalización para la banda US902-928



Fuente. (LoRa Alliance®, 2017)

2.4.3. Trama física de LoRa

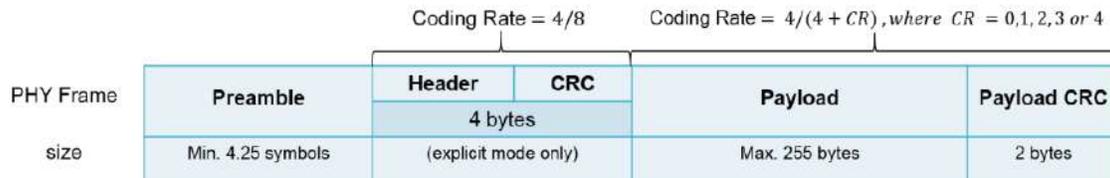
En la Figura 7, se visualiza el formato de la trama física de LoRa, el primer campo observable corresponde al preámbulo de sincronización, este define el esquema de modulación y sus parámetros, por lo que permite ajustar el rendimiento y consumo energético en base a la transmisión para cada paquete. Este campo termina con un byte de sincronización que le permite diferenciar su red de otras redes LoRa cercanas. (Nrváez Narváez & Contreras Pérez, 2020)

El tamaño máximo del payload es de 255 bytes y es suficiente para la transmisión de datos como la geolocalización, estado del dispositivo, datos de sensores, entre otros. El

campo PHYPayload contienen la trama de la capa de acceso al medio LoRaWAN. (Narváez Narváez & Contreras Pérez, 2020)

Figura 7

Formato de la trama física.



Fuente: (Narváez Narváez & Contreras Pérez, 2020, pág. 37)

- **Preámbulo:** Una secuencia de chirps (variación de frecuencia) de tamaño variable que se utiliza para sincronización de tiempo y frecuencia.
- **Cabecera:** Campo que contiene la longitud de la carga útil, la velocidad de datos utilizada, un bit que indica la presencia de una verificación de redundancia cíclica (CRC) de la carga útil y una suma de verificación de encabezado de 1 byte.
- **Payload:** Campo de longitud variable que contiene los datos transmitidos.
- **CRC:** es opcional y comprende redundancia cíclica bytes de verificación (CRC) para la protección contra errores para la carga útil.

2.4.4. Tiempo en el aire del paquete (ToA)

Este parámetro corresponde al tiempo empleado para la transmisión de un paquete, es decir, es el valor de tiempo que demora el paquete desde el momento en que sale del transmisor y hasta que es recibido por el receptor. Este valor de tiempo depende de la configuración que se realice, tomando en cuenta los parámetros de SF, BW y CR. La duración del paquete en el aire se puede obtener mediante la calculadora que ofrece como herramienta la empresa responsable del desarrollo de LoRa, denominada Semtech.

El tiempo del paquete en el aire o ToA, según Semtech, se puede obtener mediante la siguiente ecuación:

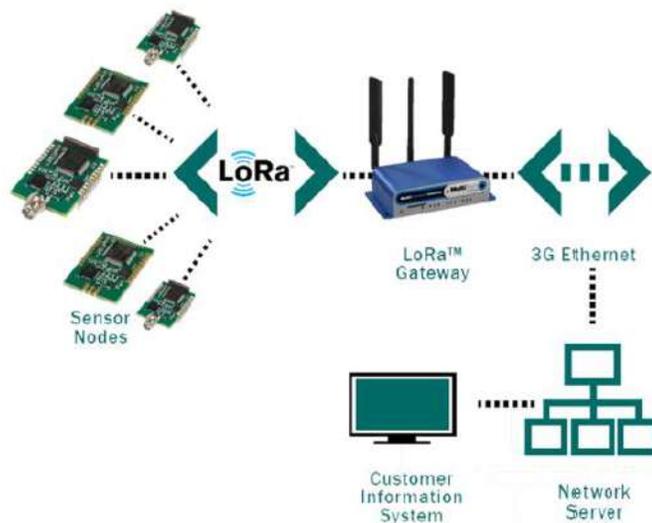
$$ToA = T_{preambulo} + T_{payload}$$

2.4.5. Arquitectura de la red LoRa

La red LoRa está basada en una topología estrella, en la cual los dispositivos sensores se comunican con uno o varios Gateway, los cuales son los encargados de enviar la información recibida de los nodos hacia un servidor o destino requerido como se observa en la siguiente figura.

Figura 8

Arquitectura de red LoRa



Fuente. (Maldonado Borda & Pineda Cusba, 2016, pág. 19)

A continuación, se presentan los diferentes elementos pertenecientes a la red:

- **Nodos o sensores:** Consisten en varios elementos, sean sensores o actuadores, los cuales se despliegan en un área geográfica definida y permiten la recolección de datos del entorno y lo envían hacia el Gateway.

- **Gateway:** Es el elemento central de la red, es decir, todos los nodos envían la información recogida y estos la reenvían hacia un servidor o destino requerido.
- **Servidores de red:** Son elementos encargados de la recepción y procesamiento de la información proveniente del Gateway.

2.5. Protocolo LoRaWAN

“LoRaWAN es un estándar abierto que define el protocolo de comunicaciones para la tecnología LPWAN basada en los chips LoRa” (Maldonado Borda & Pineda Cusba, 2016, pág. 19). LoRaWAN es el protocolo de red que utiliza LoRa, está diseñado para permitir la conexión inalámbrica de dispositivos que trabajan a base de baterías, es decir, dispositivos que poseen limitado recurso energético, con ello se hace referencia a las LPWAN, redes diseñadas para este tipo de características.

“El protocolo incluye funciones que admiten comunicaciones bidireccionales de bajo costo, móviles y seguras para IoT, máquina a máquina (M2M), ciudad inteligente y aplicaciones industriales” (Cantuña Escalante & Satián Guamán, 2019, pág. 26).

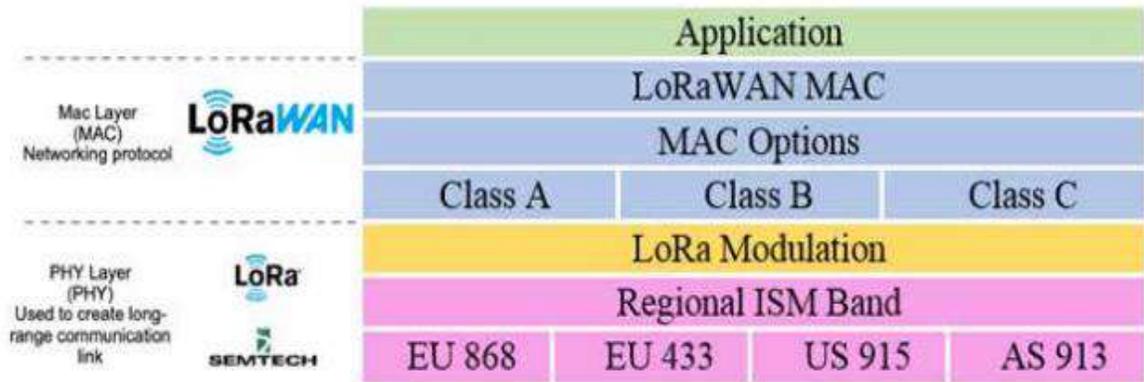
LoRaWAN es un protocolo de comunicación inalámbrica, el cual utiliza como capa física a la tecnología LoRa. Mediante LoRaWAN se define la capa MAC y se administra y comunica los dispositivos LoRa logrando conexiones seguras, móviles y de bajo consumo y coste. (Ortiz Soza, 2020)

2.5.1. Arquitectura LoRaWAN

LoRaWAN está definida dentro de la capa de acceso al medio, se visualiza en la Figura 9 las diferentes capas de esta tecnología y su interacción, desde la capa física con las bandas de frecuencias y la capa MAC con la clase de dispositivos para el protocolo LoRaWAN.

Figura 9

Arquitectura del protocolo LoRaWAN



Fuente. (Pérez, 2019)

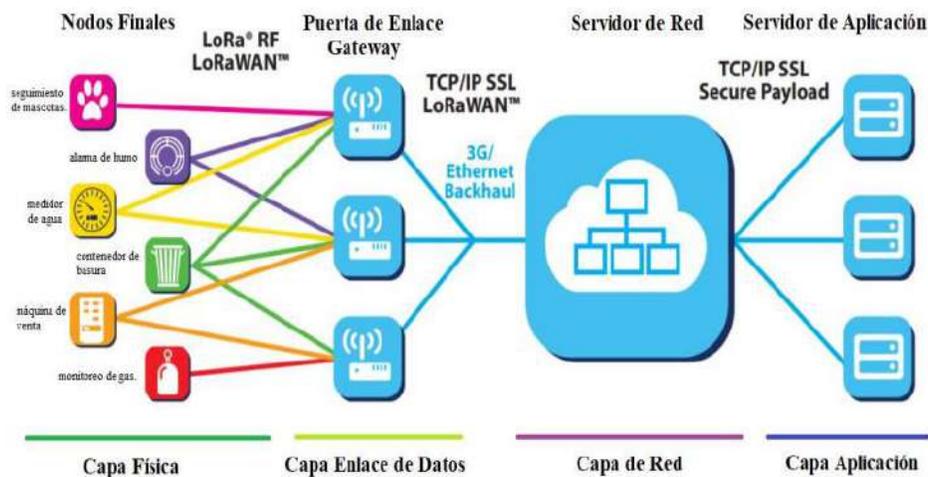
La arquitectura LoRa y LoRaWAN contiene 4 capas, las cuales se estructuran de la siguiente manera:

- **Capa Física:** Conformada por los nodos finales o nodos sensores que recolectan la información del medio.
- **Capa Enlace de Datos:** Conformada por la puerta de enlace que recibe la información de todos los nodos implementados en la red.
- **Capa de Red:** Conformada por los medios de transmisión como tecnologías 3G, Ethernet, Wifi etc. Comunica la capa de enlace de datos con la capa aplicación o la nube.
- **Capa Aplicación:** Dirigida a los servicios o aplicaciones en la Nube, almacenamiento, presentación, seguridad y tratamiento de la información mediante el uso de protocolos TCP/IP.

Lo mencionado anteriormente, puede visualizarse de forma gráfica en la siguiente figura, donde se observan los elementos de la red que componen la arquitectura de una red LoRa.

Figura 10

Arquitectura básica de LoRa y LoRaWAN



Fuente. (LoRa Alliance, 2015)

2.5.2. Seguridad LoRaWAN

La seguridad en LoRaWAN puede obtenerse en dos niveles, la primera corresponde a la encriptación de paquetes a nivel de capa red y la segunda a nivel de capa aplicación. En cuanto al primer nivel, se puede proteger la red de dispositivos maliciosos capaces de acceder a la red y suplantar la identidad de uno de los nodos de la red, por otra parte, en capa aplicación la seguridad garantiza que el operador de la red no tenga la accesibilidad a los datos de la aplicación. Para la seguridad en los dos niveles mencionados se utiliza AES con las claves de 128 bits, las cuales se basan en distintos identificadores de los dispositivos, red y aplicación. (Maldonado Borda & Pineda Cusba, 2016)

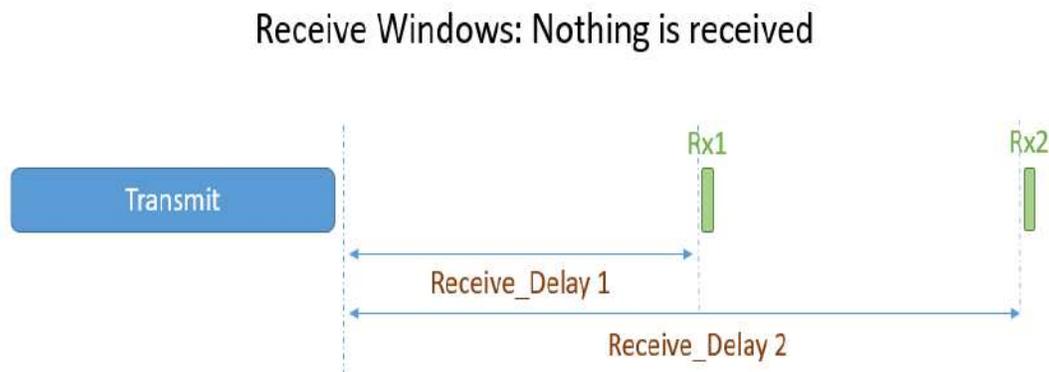
2.5.3. Clases de Dispositivos LoRaWAN

LoRaWAN se encuentra definida dentro de la capa de acceso al medio, por lo que permite la conexión de tres clases de diferentes dispositivos:

- **Clase A:** En este tipo de dispositivos se requiere una menor cantidad de energía para la operación, ya que especifica un punto final bidireccional que permite dos ventanas de recepción luego de una transmisión hacia el Gateway, por lo que es ideal para el ahorro de energía en dispositivos dependientes de una batería. (Pérez, 2019)

Figura 11

Ranura de recepción de un dispositivo clase A

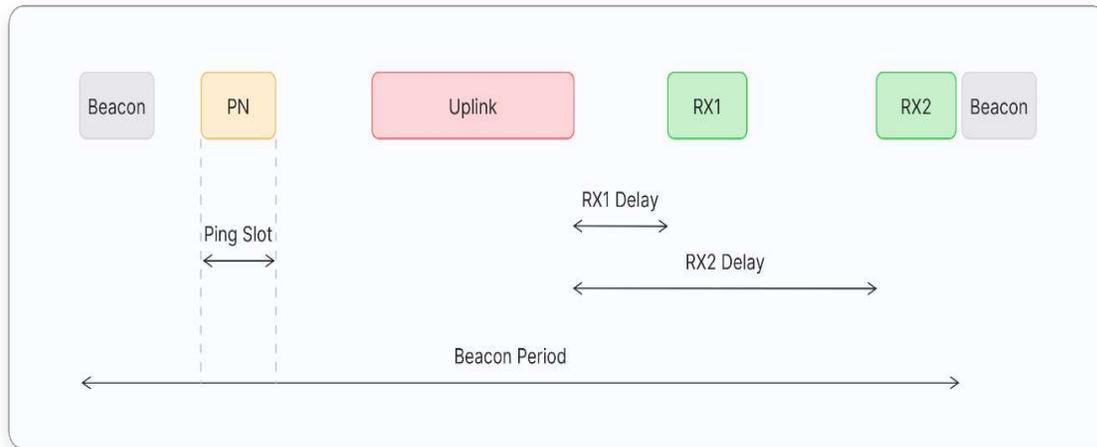


Fuente. (LoRa Alliance®, 2022)

- **Clase B:** Estos dispositivos pueden recibir datos sin necesidad de enviar paquetes, permitiendo así a la aplicación enviar mensajes programados, un ejemplo claro puede evidenciarse en una red LoRa cuando el Gateway envía mensajes de sincronización periódicamente y negociar tiempos de recepción, esto conlleva a un mayor consumo de energía que los de clase A, debido a los mensajes de sincronización periódicos. (Maldonado Borda & Pineda Cusba, 2016)

Figura 12

Ranura de recepción de un dispositivo clase B

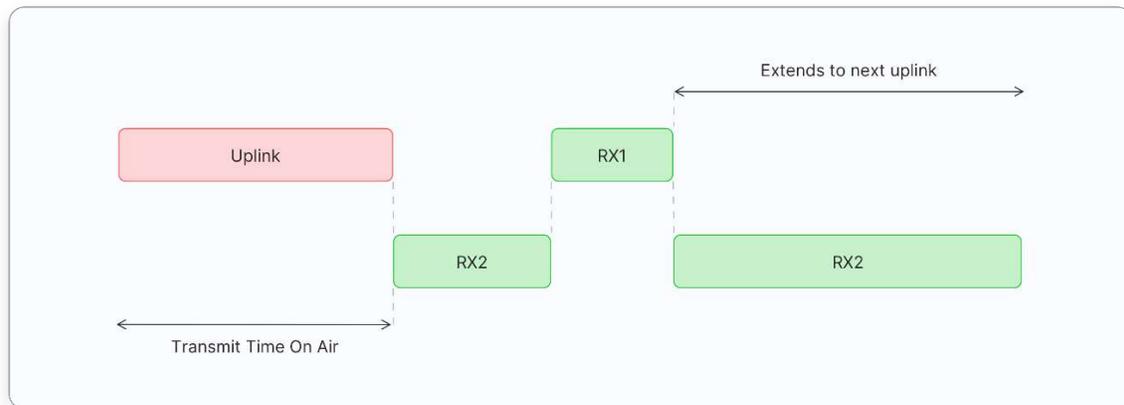


Fuente. (The Things Network, 2022)

- **Clase C:** Estos dispositivos tienen un consumo de energía mucho mayor que los dos anteriores dispositivos, esto debido a que siempre están escuchando en la red y son capaces de recibir datos en cualquier momento, menos cuando estén enviando datos, esto proporciona mejores tiempos de respuesta, pero alto consumo de energía. (Maldonado Borda & Pineda Cusba, 2016)

Figura 13

Ranura de recepción de un dispositivo clase C



Fuente. (The Things Network, 2022)

2.6. El Agua

La Tierra está cubierta en su gran mayoría por agua, pero solo un pequeño porcentaje de esta es apta para el consumo humano, después del proceso de captación y potabilización está disponible para satisfacer las necesidades de consumo de las personas. “Es un recurso natural que predomina en el planeta tierra, fundamental para el desarrollo de los seres humanos ya que se depende de su disponibilidad para el uso doméstico y labores agrícolas” (Garrido Terán, 2021).

2.6.1. Agua Superficial

Como su nombre lo indica, las aguas superficiales son aquellas fuentes de agua que pueden encontrarse sobre la superficie de la Tierra, esto debido a su acumulación sobre la misma a causa de las precipitaciones, de acuerdo con el terreno éstas tienden a formar corrientes como en ríos o arroyos, pero también pueden encontrarse en un solo punto como los lagos o lagunas. (Induanálisis Laboratorio Ambiental, 2019)

2.6.2. Agua Subterránea

El agua subterránea es aquella que puede encontrarse por debajo de la superficie terrestre, esta básicamente es una reserva de agua para ser utilizada como agua potable, ya que al estar debajo de la superficie tiende a tener menos contaminación que las aguas superficiales. Además, muchas de estas suelen desembocar en la superficie para formar manantiales o abastecer el flujo de caudal de muchos ríos en la superficie. (Induanálisis Laboratorio Ambiental, 2019)

2.7. Captación de Agua.

“La captación de aguas superficiales como ríos, lagos y embalses es una estructura a nivel del terreno mediante la cual se hace uso y aprovechamiento del agua de la fuente que corresponda” (Stauffer & Spuhler, 2018). Los sistemas de captación de agua pueden utilizar la captación de agua por gravedad o por bombeo, en este caso, la captación por gravedad se realiza cuando la fuente de agua se encuentra a una altitud mayor que el lugar donde se realiza dicho proceso. La captación de agua es una estructura fundamental para poder proveer de este recurso a una comunidad.

2.8. Cambio Climático

“Por cambio climático, se entiende un cambio de clima atribuido directa o indirectamente a la actividad humana que altera la composición de la atmósfera mundial y que se suma a la variabilidad natural del clima observada durante periodos de tiempo comparables” (Paspuel Güel, 2020). El cambio climático puede ser un factor que afecte el nivel de caudal de las vertientes que se utilizan para la captación de agua, en Tulcán actualmente según (El Universo, 2021) la falta de precipitaciones en la zona ha provocado

que el nivel de agua de las principales fuentes de captación de la ciudad de Tulcán se vea reducido considerablemente, provocando así la disminución en la cantidad del recurso.

2.9. Caudales en ríos.

“El caudal es la cantidad de agua que lleva una corriente hacia un canal o cañería que llega a los estanques en un cierto tiempo” (Santos Benavides, 2019, pág. 27). La precipitación del agua ya sea en forma líquida como la lluvia o sólida como el granizo o nieve, al llegar al terreno y acumularse tienden a dar origen los ríos, arroyos o vertientes, los cuales de acuerdo con sus características llevan cierto volumen de agua conocido como caudal.

2.10. Medición del nivel

“Para una adecuada forma de medir un fluido, ya sea en reposo o en movimiento, debe tener una medida de referencia, que por lo general es el punto inferior o base en el que el fluido descansa y el vértice superior hasta dónde llega el líquido” (Hoyos Orozco, 2021, pág. 15). La medición del nivel corresponde a una medida tomada desde el fondo del canal o la base sobre la cual el fluido se encuentra, hasta la altura máxima que alcanza el mismo, visto desde la superficie. Conocer el nivel de agua de un canal puede ser muy útil, ya que esta variable ayuda a registrar el nivel y su comportamiento a través del tiempo acorde a los factores climáticos de la zona, como ascensos o descensos en el nivel acorde a mayor o menor precipitación.

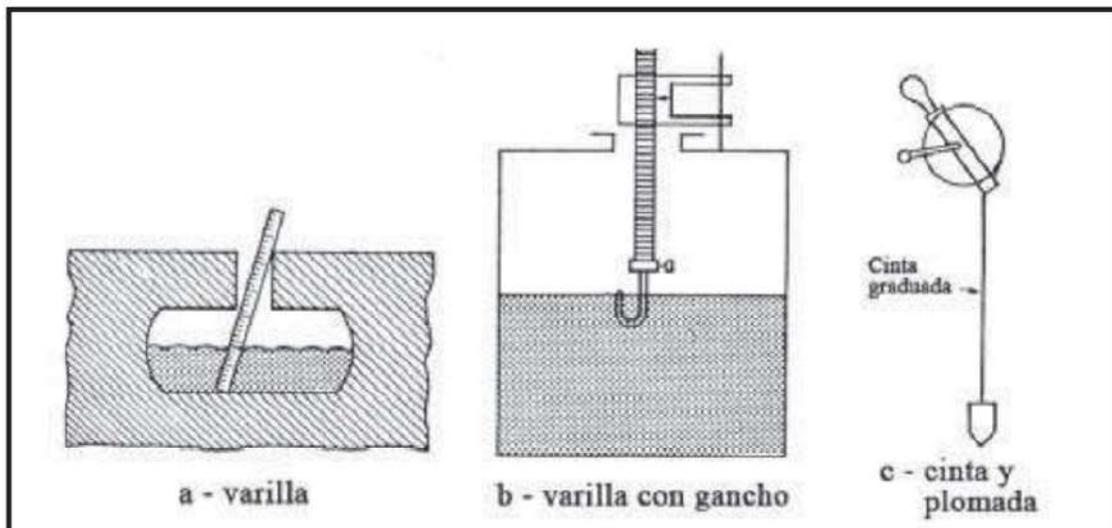
Para la medida del nivel de agua existen varios procedimientos, los cuales pueden llevarse a cabo de diferentes formas como: la medición desde el exterior, interior y desde la superficie, esto depende según los instrumentos de medida que se utilicen y si es necesario entrar en contacto o no con el líquido.

2.10.1. Medidor de sonda

Este medidor consiste en una varilla o regla, la cual previamente debe ser graduada y con el tamaño adecuado puede ser introducida dentro del depósito o caudal que contiene el líquido, de igual forma que el método anterior, su medición debe realizarse observando el objeto en el cual se marcará la información del nivel. “La determinación de nivel se efectúa por lectura directa de la longitud alcanzada por el líquido” (Hoyos Orozco, 2021, pág. 15).

Figura 14

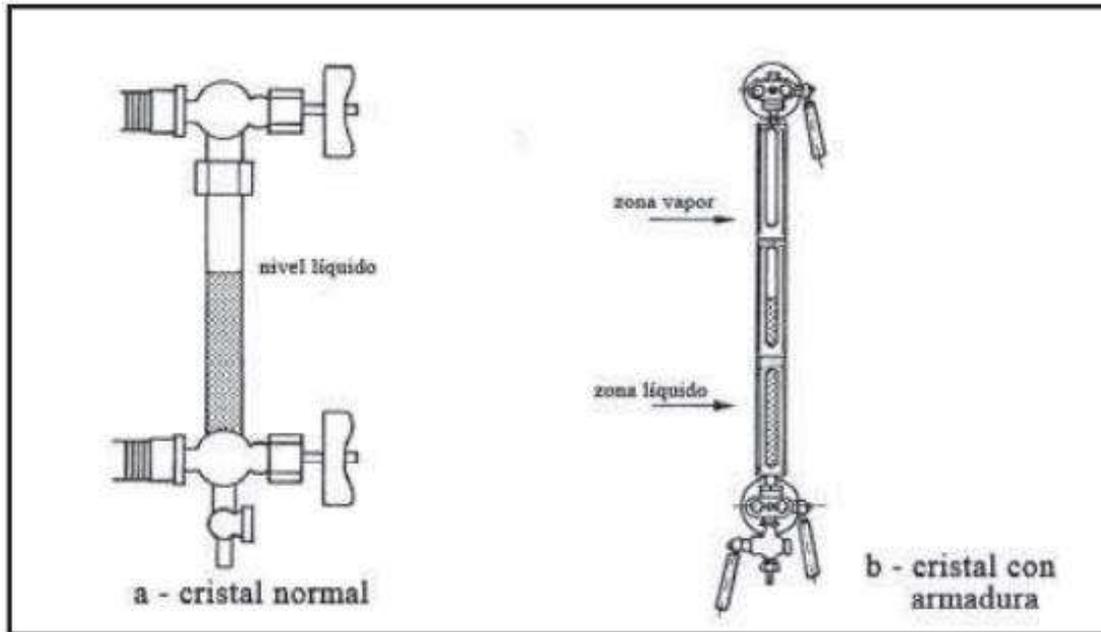
Medidor de sonda



Fuente. (Creus Solé, 2010)

2.10.2. Medidor de cristal

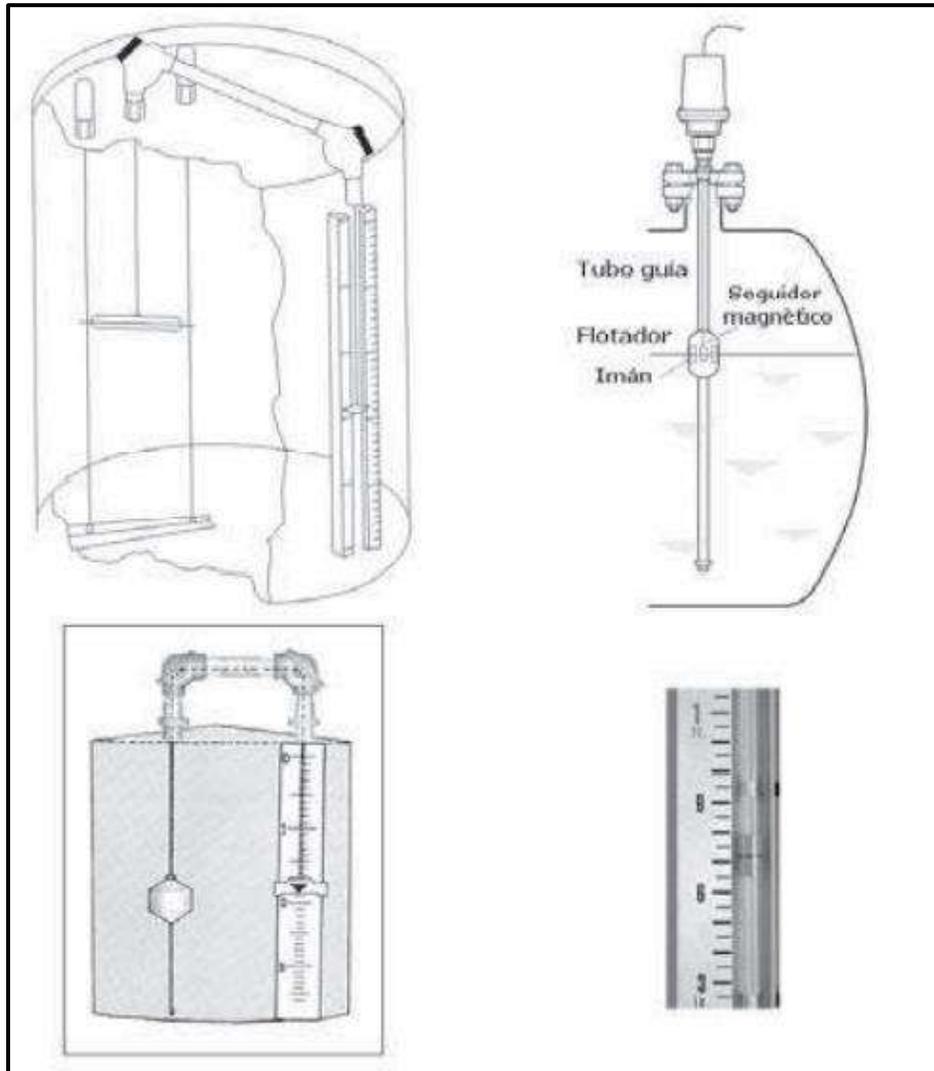
“Consiste en un dispositivo de vidrio, conectado a bloques metálicos cerrados por prensa, los cuales están unidos al recipiente por medio de tres válvulas, dos de cierre de seguridad para evitar escapes y una purga” (Hoyos Orozco, 2021, pág. 16).

Figura 15*Medidor de cristal*

Fuente. (Creus Solé, 2010)

2.10.3. Medidor de flotador

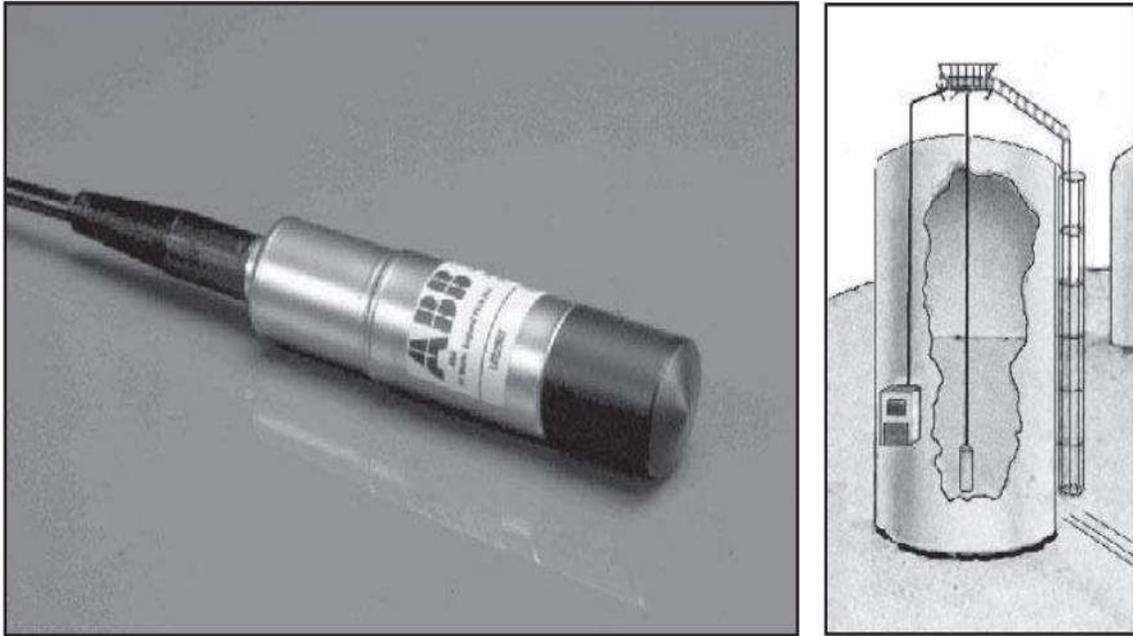
Este tipo de medidor es utilizado dentro de tanques que contengan agua líquida, consiste en un flotador conectado mecánicamente a una regleta graduada la cual mediante la variación del líquido indica el nivel de agua contenida en el recipiente. “Consta de un elemento tipo flotador ubicado en la superficie del líquido y conectado al exterior del recipiente, genera indicaciones de nivel sobre una escala graduada” (Hoyos Orozco, 2021, pág. 16).

Figura 16*Medidor de flotador*

Fuente. (Creus Solé, 2010)

2.10.4. Medidor manométrico

“La lectura del manómetro indica directamente la altura entre el nivel del líquido y el eje del dispositivo” (Hoyos Orozco, 2021, pág. 17). Este medidor consiste en un dispositivo conocido como manómetro, el cual es capaz de medir la altura de un líquido en base a la presión hidrostática que ejerce este sobre el eje del aparato.

Figura 17*Medidor manométrico*

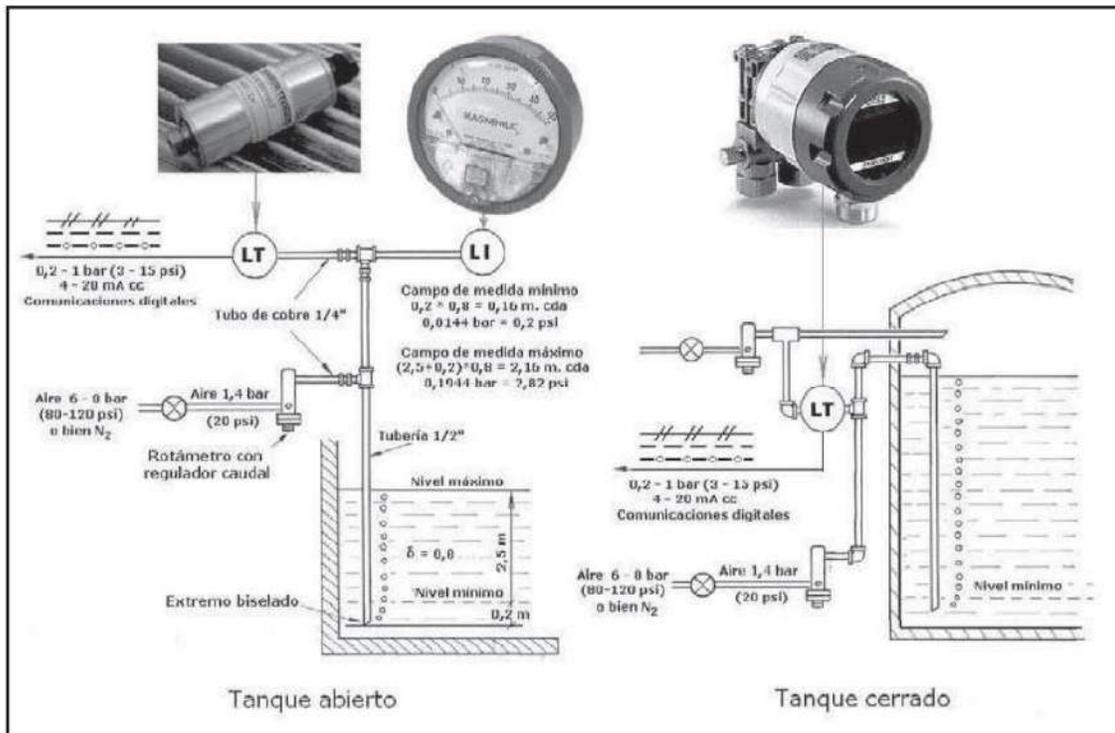
Fuente. (Creus Solé, 2010)

2.10.5. Medidor de burbujeo

En este instrumento de medida se utiliza un regulador de caudal el cual, mediante un tubo colocado hasta el nivel mínimo interior del líquido a medir, envía un flujo de aire o gas inerte, de esta forma se crea un flujo continuo de burbujas, en donde la presión requerida para este flujo corresponde a la medida de nivel del líquido. (Hoyos Orozco, 2021)

Figura 18

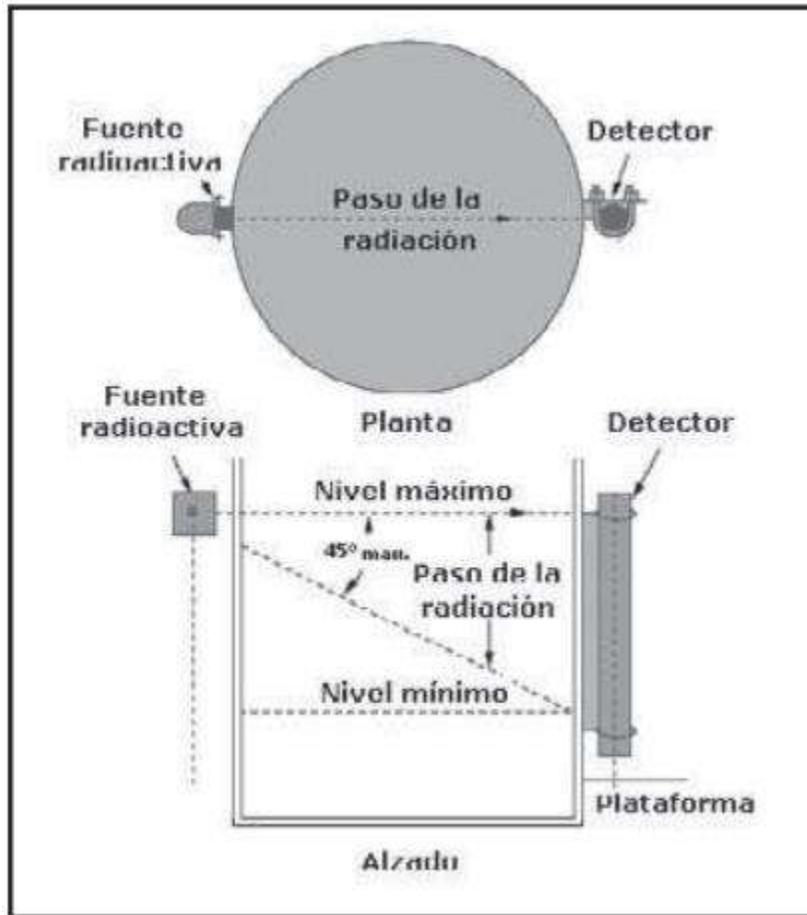
Medidor de burbujeo



Fuente. (Creus Solé, 2010)

2.10.6. Medidor por radiación

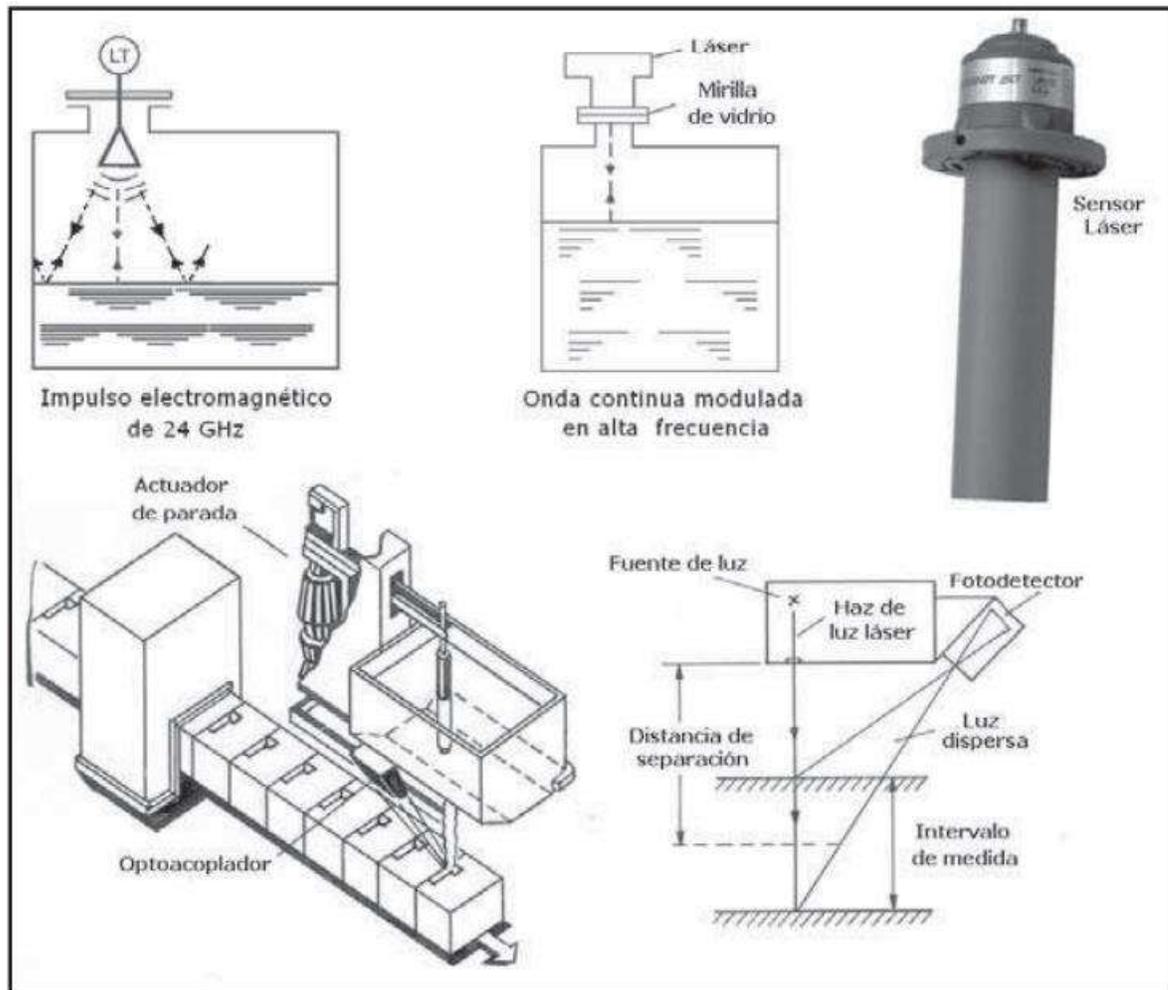
“El sistema de radar de microondas genera una emisión continua de ondas electromagnéticas, típicamente en valores de 10 GHz” (Hoyos Orozco, 2021, pág. 19). Este tipo de medidor se basa en la utilización de ondas electromagnéticas que están en el rango de 10 GHz a 11 GHz, realiza un barrido mediante su sistema de radar sobre el líquido en medición y la diferencia de frecuencias para la señal de transmisión y la señal de recepción es proporcional a al tiempo.

Figura 19*Medidor por radiación*

Fuente. (Creus Solé, 2010)

2.10.7. Medidor por láser

Se utiliza en condiciones de entornos difíciles y donde instrumentos convencionales no funcionarían adecuadamente, esencialmente son utilizados para medir metales en su forma líquida o fundidos, de esta forma se está lo más alejado del objeto de medición y a de las altas temperaturas. “El sistema emite un rayo láser dirigido por reflexión en la superficie del metal fundido, mide el tiempo que transcurre entre el impulso emitido y de retorno, este tiempo es directamente proporcional a la distancia del aparato emisor a la superficie” (Cortés Aguilar, pág. 9).

Figura 20*Medidor por láser*

Fuente. (Creus Solé, 2010)

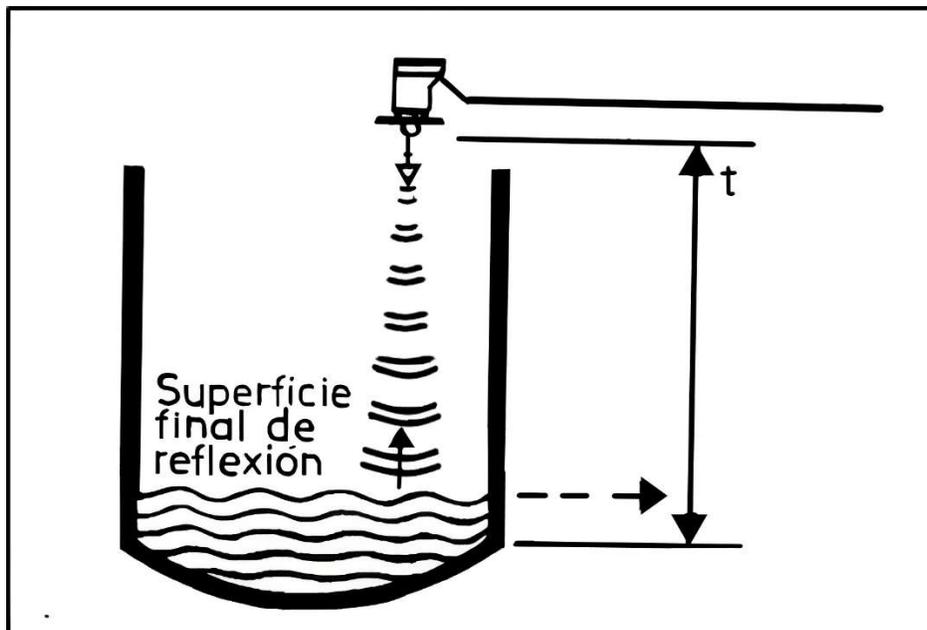
2.10.8. Medidor ultrasónico

Este medidor corresponde a un dispositivo el cual es capaz de enviar señales mediante pulsos ultrasónicos, los cuales se reflejan en una superficie que se encuentre en su trayectoria, el tiempo de ida y de vuelta del dispositivo permitirá obtener una medición de la distancia a la cual se encuentra el objeto a medir, realizando los cálculos necesarios se puede obtener la medida de nivel de un líquido. Por lo general, dispositivos de este tipo trabajan a una

frecuencia de 20 kHz, permitiéndoles atravesar condiciones del medio en donde existan gases o vapores. (Hoyos Orozco, 2021)

Figura 21

Medidor ultrasónico

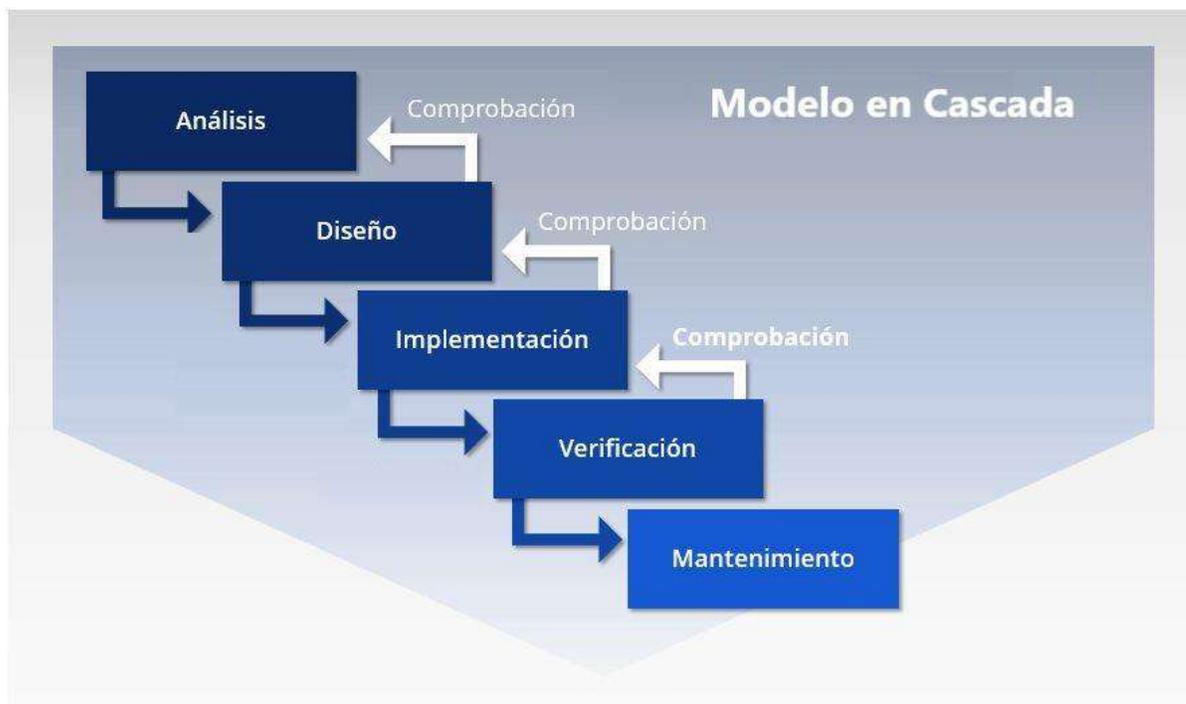


Fuente. (Creus Solé, 2010)

2.11. Metodología en cascada

La metodología en cascada consiste en un proceso secuencial y lineal, esta metodología se aplica a la gestión de proyectos y se divide en distintas fases, en la que cada una debe comenzar y terminar en un tiempo definido, es decir, no se puede iniciar la siguiente fase sin haber terminado la anterior. En cada una de las fases se cuenta con un conjunto de objetivos y actividades definidos para completar lo definido en el desarrollo del proyecto.

En la Figura 22, se puede observar las diferentes fases de la metodología en cascada, éstas siguen una secuencia de forma lineal y cada una da paso a la siguiente, por lo que es necesario cumplir con lo planteado en cada una de ellas.

Figura 22*Metodología en cascada*

Fuente. (IONOS Digital Guide, 2019)

En la fase de análisis se realiza la recopilación de los requerimientos necesarios como bases para el inicio del proyecto, la fase de diseño consiste en formular la solución específica para lo planteado en base a lo realizado en la fase anterior, en la fase de implementación se desarrolla cada elemento del sistema por separado y se ejecuta el funcionamiento de cada uno, para la fase de verificación o también conocida como fase de pruebas, se ejecutan las pruebas necesarias para verificar el funcionamiento del sistema completo aplicado al entorno seleccionado, en la fase final se realiza la entrega del sistema, soporte y mejora en caso de ser necesario.

3. Capítulo III: Diseño del Sistema

El siguiente capítulo, describe el diseño y desarrollo del sistema propuesto para realizar la medición del nivel de agua del caudal de la principal fuente de abastecimiento de la ciudad de Tulcán, en base a la metodología en cascada, específicamente la fase uno y dos, el capítulo estará enfocado en plantear los requerimientos iniciales del sistema y la selección del hardware y software adecuados para su diseño respectivamente.

3.1. Descripción General del Sistema

El presente proyecto tiene como finalidad el desarrollo de un sistema de medición del nivel de agua del caudal de la principal fuente de abastecimiento de la ciudad de Tulcán, este sistema lo que permitirá es identificar los niveles de agua que se presencien en el cauce de dicha fuente, esto para permitir evidenciar altos y bajos en el nivel de agua utilizando una red de sensores inalámbricos conectados mediante la tecnología de comunicación LoRa.

Poder conocer acerca del nivel de agua de una fuente hídrica puede ser muy importante, ya que con el cambio climático se evidencian variaciones en la temperatura ambiental, lo que provoca varianzas en las precipitaciones y en la frecuencia con la que se dan las mismas, entonces al conocer la variable física nivel de agua se puede llegar a asociar las inundaciones o sequías, al incrementar o reducir el nivel de agua. El sensor ultrasónico será el encargado de realizar la medición del nivel del recurso hídrico, este es un elemento no inmersivo, lo cual indica que no entrará en contacto con el agua, la señal recibida de este elemento se procesa mediante una placa microcontroladora la cual incluye un módulo LoRa, lo que permitirá la comunicación a larga distancia con el nodo concentrador o Gateway LoRa, el cual se conformará por una placa que procesará los datos para posteriormente ser enviados a la nube y poder ser accedidos desde internet.

Este sistema realiza un monitoreo diario del nivel de agua del recurso hídrico, con lo que se podrá tener la información de la variación de los niveles frecuentemente, esto permitirá identificar incrementos o descensos en el nivel de agua del caudal.

Para el presente proyecto se toma como referencia el trabajo realizado por parte de la Pontificia Universidad Javeriana, la cual cuenta con un Centro de Excelencia y Apropiación en Internet de las Cosas (CEA-IoT), aquí se realizan varios desarrollos de proyectos relacionados al Internet de las Cosas, en este caso, el proyecto de referencia consiste en una solución IoT para el monitoreo del nivel del caudal del río Negro y Nare. Para el monitoreo del nivel se utilizan sensores ultrasónicos, como comunicación utilizan la red celular y utilizan aplicaciones en la nube para poder mostrar los datos de los sensores, esto según explica el ingeniero Luis Carlos Trujillo.

En la Figura 23, puede observarse el sistema de medición desplegado, este corresponde al nodo de monitoreo de nivel de río Negro de la Pontificia Universidad Javeriana. Se observa una parte del río Negro en donde existe un puente sobre su caudal, se encuentra el nodo sensor anclado mediante una estructura metálica al puente antes mencionado, esto le permite estar sobre el río para poder realizar la medición del nivel utilizando un sensor ultrasónico.

Figura 23

Nodo de monitoreo de nivel de río Negro de la Pontificia Universidad Javeriana



Fuente. (Trujillo, Méndez, & Acosta, 2022)

3.2. Situación Actual

Para poder ejecutar el proyecto de forma adecuada, es necesario conocer la situación actual, esto nos permitirá establecer las bases para definir los requerimientos necesarios los cuales serán considerados para el diseño y desarrollo del sistema planteado.

3.2.1. Ubicación

El presente proyecto se desarrolla en Río Chico, localidad de Tufiño, parroquia correspondiente al cantón Tulcán provincia del Carchi. En Río Chico se cuenta con la afluyente del río que lleva su mismo nombre, el cual es considerado una fuente de captación para la potabilización del agua debido a sus características.

En la siguiente figura obtenida de Google Earth, puede observarse la ubicación de Río Chico, la cual se encuentra en la ubicación geográfica con las siguientes coordenadas 0.78°N , 77.84°O de latitud y longitud respectivamente, el marcador rojo indica la ubicación de Río Chico cerca de la parroquia de Tufiño.

Figura 24

Ubicación geográfica de Río Chico en Google Earth



Fuente. Google Earth

3.2.2. Características

En la siguiente figura, se muestra la localidad en donde se plantea el desarrollo del proyecto, aquí se observa el cauce del río, el cual tiende a variar en su nivel acorde a las temporadas lluviosas o de sequía que influyen en la captación de agua.

Figura 25*Caudal del Río Chico*

Fuente. Autoría

Como es evidente, en esta zona no existe un monitoreo, que permita detectar en qué circunstancias se encuentra el nivel del río cerca de la zona donde posteriormente se realiza la captación de agua. Por ello, es que se plantea la ubicación de una red de sensores inalámbricos, que permita conocer si el nivel del caudal del río aumenta, disminuye o se mantiene constante a lo largo del tiempo.

En la siguiente figura, se observa la zona donde se pretende instalar los nodos sensores, en esta zona el río cuenta con las siguientes características: un ancho de 3.85m, de orilla a orilla y una sección transversal de 0.5m aproximadamente, esta medida corresponde a la tomada desde el fondo hasta la superficie.

Figura 26*Características de Río Chico**Fuente. Autoría*

Debido a que el lecho del río tiende a ser irregular por la acumulación de rocas y otros elementos, se plantea la utilización de dos nodos sensores, los cuales se implementaran en lugares diferentes a lo largo del río específicamente en las zonas donde esté presente una topografía más regular para poder obtener una medida del nivel mucho más estable en la zona externa a la planta de captación.

3.2.3. Planta de Captación

La planta de captación lleva el nombre de planta de captación de Río Chico, en la siguiente figura, se observa la estructura de captación del agua que proviene del río y se conduce a través de este canal, esta zona corresponde a una parte externa de la planta.

Figura 27*Planta de captación de Río Chico*

Fuente. Autoría

Río Chico es la principal fuente de captación de agua para la potabilización, esta planta está conectada a la red de distribución de agua potable de la ciudad de Tulcán la cual cuenta con tres tanques de reserva ubicados en diferentes sitios de la ciudad. Cabe mencionar que el acceso a la planta interna está restringido, por ello, las mediciones se las realiza en la parte externa, justo en el río donde se desea medir el nivel de este para conocer si incrementa o reduce su cauce.

3.3. Diseño del Sistema

Para el diseño del sistema se ha tomado en cuenta tres etapas, las cuales son: la etapa de sensado, la etapa de control y la etapa de visualización. El sistema estará conformado por las tres etapas correspondientes, en las cuales la primera etapa se encarga de la recolección de los datos, en este caso de la variable física de nivel de agua; en la etapa de control se tiene la placa controladora para el procesamiento de los datos; finalmente, se tiene la etapa de visualización donde se observa el detalle de la información de la variable física a medir.

En la etapa de sensado, se tiene los diferentes componentes, que corresponden a un sensor ultrasónico capaz de medir distancias de acuerdo con su principio de funcionamiento, por ende, se utiliza un elemento no inmersivo para obtener el valor de medición del nivel de agua. Además, este estará conectado a una placa microcontroladora para establecer sus funciones y un módulo LoRa para la comunicación con el nodo central.

En la etapa de control, se tiene una placa controladora de circuito impreso y un módulo LoRa para poder establecer la comunicación con los sensores, los cuales enviarán los datos para posteriormente ser procesados por este elemento y enviados a su siguiente fase respectiva.

En la etapa de visualización, se observa las diferentes tomas de medición de nivel obtenidas desde los sensores de ultrasonidos en una página web a la cual se accede desde internet, con ello se puede observar la variación del nivel de agua del caudal y la información que este proporciona.

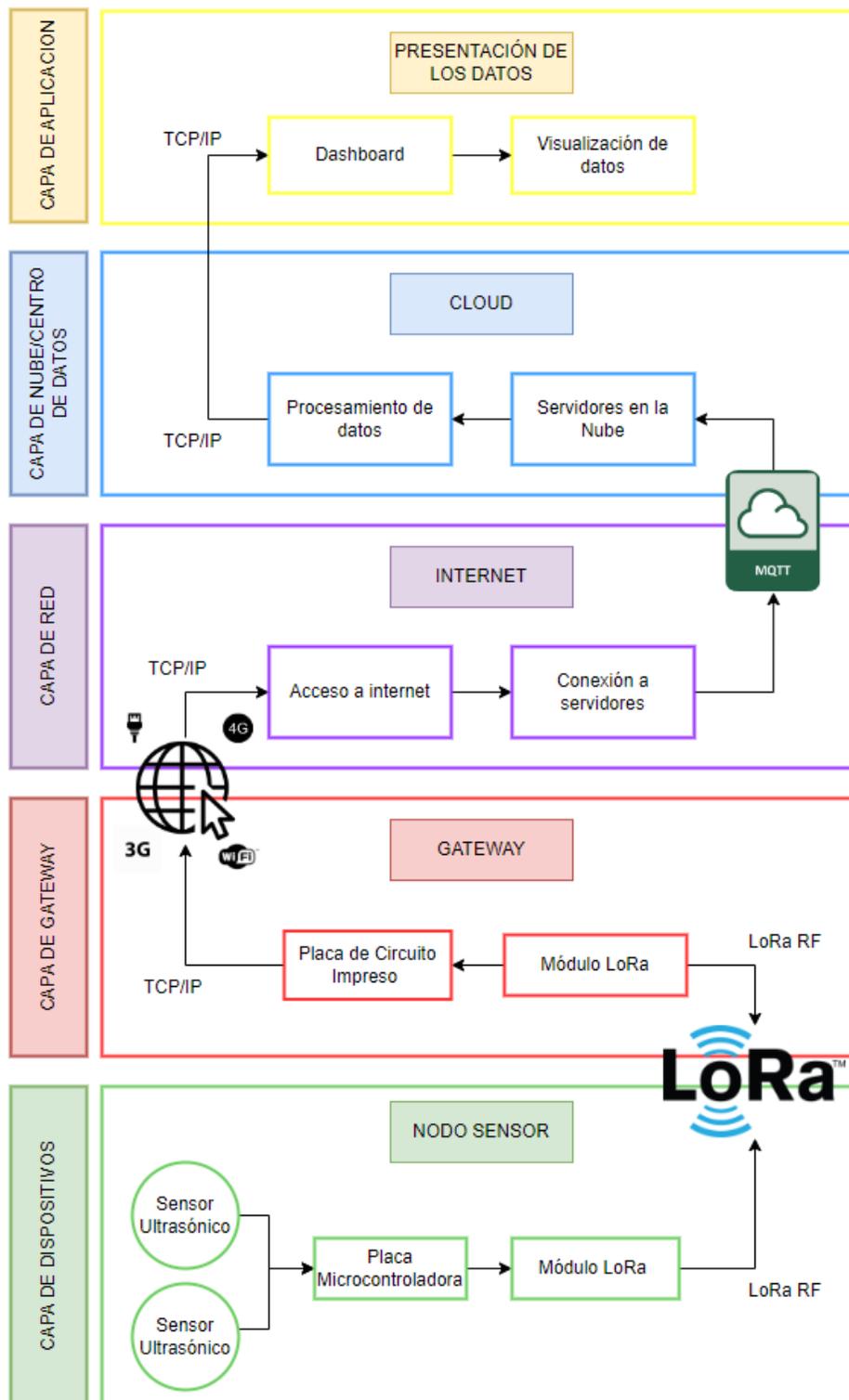
3.4. Diagrama de Bloques del Sistema

A continuación, se presenta el diagrama de bloques del sistema, el cual está conformado por las capas de la arquitectura IoT, en las cuales se detallan de forma general los elementos y requerimientos a utilizar en cada una para el diseño del sistema propuesto.

En la capa de dispositivos, se tiene los sensores ultrasónicos, los cuales son los encargados de obtener los datos de la variable a medir, estos se conectan a una placa microcontroladora para ejecutar la programación respectiva, adicionalmente, se añade un módulo LoRa, el cual permitirá la comunicación inalámbrica mediante dicha tecnología hacia el Gateway LoRa. En la capa de Gateway se encuentra la placa de circuito impreso o placa controladora que posee altas capacidades de procesamiento, además se conecta a un módulo LoRa para permitir la comunicación con los nodos sensores, de esta forma se compone el denominado nodo central o Gateway LoRa, todos los datos recolectados de los sensores son transmitidos hacia este elemento de la red. En la capa de red se establece la conexión a internet, esta conexión se la puede realizar de diferentes formas, ya sea utilizando un medio cableado o inalámbrico, en este caso se utiliza una conexión inalámbrica para de esta forma poder enviar los datos del Gateway LoRa hacia los servidores utilizando el protocolo de conexión MQTT, aquí se tiene la capa de nube o centro de datos, que nos permite almacenar los datos recibidos para posteriormente pasar a la capa final, que corresponde a la capa aplicación, en la cual se visualiza la información mediante herramientas gráficas, por ejemplo un dashboard.

Figura 28

Diagrama de bloques del sistema



Fuente. Autoría

3.5. Requerimientos del Sistema

El diseño del sistema está orientado a la monitorización de la variable física de nivel de agua, por ello se cuenta con diferentes etapas que cumple el sistema, para lo cual es importante señalar los elementos que se utilizan en cada una de ellas.

3.5.1. Etapa de sensado

Esta corresponde a la primera etapa del sistema en la cual se tienen los diferentes sensores, estos se encargan de la adquisición de los datos realizando lecturas de la variable planteada, en este caso, se realiza la medición del nivel de agua del caudal mediante la utilización de sensores ultrasónicos. Las unidades que se requieren medir en esta instancia se programan en el microcontrolador conectado a cada uno de los sensores, este microcontrolador corresponde a la placa ESP32, cada nodo sensor posee un módulo LoRa RFM95, que corresponde al módulo de comunicación inalámbrica que proporciona el medio de transmisión de datos hacia el Gateway LoRa.

3.5.1.1. Análisis del Sensor de Nivel de Agua

Para realizar la medición del nivel de agua de un caudal se puede hacer uso de varios métodos para este fin, se puede utilizar métodos inmersivos, los cuales tienen como característica principal que para realizar la medición se requiere que los instrumentos de medición estén en contacto directo con el agua, en este caso podemos destacar el medidor de sonda, el cual consiste en una varilla o regla graduada con las dimensiones necesarias para poder sumergirlo en el líquido. Por otra parte, se tiene los métodos de medición de nivel de agua no inmersivos, lo que se refiere a que no se necesita estar en contacto con el líquido para obtener la medición de nivel, en este caso se destaca el medidor ultrasónico, ya que utiliza señales que corresponden a pulsos ultrasónicos para medir la distancia hacia una

superficie, de este modo, realizando los cálculos necesarios se puede obtener el nivel de agua de un caudal, este tipo de medición es efectivo y menos complejo en comparación con otros, es por ello que se toma como método de medición para el presente trabajo.

Realizando una investigación de lo existente en el mercado, para la medición de variables en lo que se refiere a monitoreo del agua, encontramos diferentes opciones, en este caso visualizamos una comparativa en lo disponible para el manejo del flujo de un caudal, en el cual, se obtienen los valores correspondientes a la cantidad de agua atraviesa cierto canal o cauce de un río. A continuación, se observa en una tabla comparativa las opciones disponibles en el mercado de sensores para el ámbito industrial:

Tabla 7*Comparativa sensores industriales para medición de flujo y nivel de caudal*

Parámetros	OTT MF	OTT SLD	OTT	OTT C31	OTT C2	Phoenix	BELUGA	Caudalímetro
Técnicos	pro		SVR 100					digital
Figura								
Tipo de sensor	Medidor de caudal magnético-inductivo	Medición Doppler	Sensor radar de velocidad del agua	Molinete, medición de flujo de caudal	Molinete, medición del caudal o del flujo	Caudalímetro RADAR área/velocidad sin contacto	Caudalímetro ultrasónico Doppler	Sonda de flujo de caudal
Tecnología de medición	Electromagnético	Ultrasónico	Ultrasónico	Mecánico	Mecánico	Sensor radar no invasivo	Ultrasónico	Electromagnético

Voltaje de operación	9V	9V a 16V	9V a 27V	9V	9V	4V a 26V	4V a 26V	9V
Consumo de corriente	-	500 mW	<250 mA	-	-	1.38 W	120 mA	-
Precisión	±2%	±1%	±2%	±2%	±2%	±1%	±1%	±2%
Costo	\$7 350	\$7 500	\$10 150	\$6 200	\$5 900	\$8 430	\$7 650	\$800

Fuente. (OTT HydroMet, 2019) & (Halma Water Management, 2017)

Como se puede evidenciar en la tabla expuesta, existen diferentes dispositivos sensores, con los cuales se puede tener una medición tanto del flujo como del nivel del caudal de un río, en este caso estos dispositivos son utilizados en el sector industrial, por ende, su costo representa un alto valor de inversión, además, al ser dispositivos fabricados de manera exclusiva para este uso representan una factibilidad para la medición utilizando diferentes tecnologías como: ultrasonidos, radar y electromagnético. Este tipo de dispositivos cuentan con el software y elementos requeridos para realizar las mediciones, todo esto facilitado por el fabricante. En el caso del presente trabajo, se descarta la utilización de este tipo de sensores, ya que, a pesar de su factibilidad de uso e instalación, representan un alto costo de adquisición.

En el mercado de la electrónica para aplicaciones IoT, existen diferentes dispositivos que permiten realizar mediciones de flujo y nivel del caudal de un río, ya sea utilizando el método inmersivo o no inmersivo. Estos dispositivos son de tipo hardware libre, ya que no cuentan con software propietario, por ende, se puede utilizar libremente y realizar las modificaciones necesarias para el fin que se requiera. En cuanto a la medición para el flujo del caudal se cuenta con sensores, los cuales funcionan bajo el principio electromagnético, estos realizan la medición en base a la cantidad de agua que atraviesa dicho sensor.

Para la medición de nivel del caudal, podemos encontrar diferentes dispositivos, en el caso del método inmersivo se tiene sensores que implementan alambres paralelos expuestos, que al entrar en contacto con el agua provocan una variación en el voltaje, generando así una salida analógica que se puede interpretar para medir el nivel del líquido. Por otra parte, utilizando el método no inmersivo de medidor ultrasónico, se conocen dispositivos como sensores ultrasónicos, los cuales utilizan ondas ultrasónicas con una frecuencia que está en los 20 kHz, estos sensores emiten un pulso ultrasónico que se refleja en una superficie, esta onda reflejada regresa al sensor y mide la distancia a dicho objeto tomando en cuenta el tiempo entre la emisión y la recepción de la onda.

A continuación, se describen algunas de las características presentes en los sensores para medición de flujo y nivel del caudal:

Tabla 8

Comparativa entre sensores de flujo y nivel de caudal compatibles con hardware libre

Parámetros	EFS-04P	YF-S201	MB1240	HC-SR04	MB7076
Técnicos					
Figura					
Tipo de sensor	Efecto hall	Efecto hall	Ultrasónico	Ultrasónico	Ultrasónico
Alimentación	220 V	5V – 18V	3.3V – 5V	5V	3V – 5V
Corriente	3 A	15 mA	3.4 mA	15 mA	3.4 mA
Frecuencia	-	-	42 kHz	40 kHz	
Rango de detección	1 – 3 L/m	1 – 30 L/m	20 cm a 765 cm	2 cm a 4 m	Hasta 1068 cm

Dispositivos compatibles	-	Arduino, ESP32, Raspberry Pi	Arduino, ESP32, Raspberry Pi	Arduino, ESP32, ESP8266	Arduino, ESP32
Tamaño del sensor	2 x 5,2 cm/0,78 x 2,04	61 × 35 × 33 mm	22.1 x 19.9 x 25.11 mm	45 x 20 x 15 mm	64 x 43.8 x 43.8 mm
Costo	\$10	\$8	\$50	\$2.50	\$132

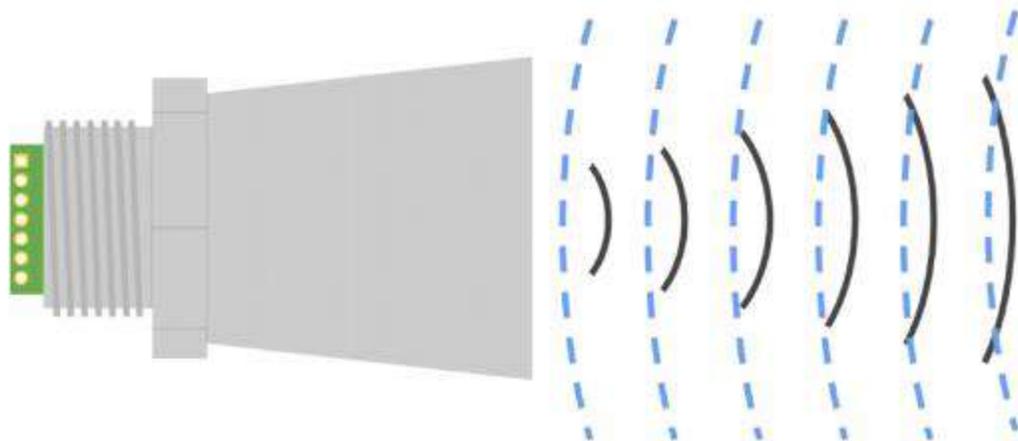
Fuente. (Santos Benavides, 2019) & (MaxBotix Inc., 2022) & (Proyectos con Arduino, 2022)

En la tabla mostrada anteriormente, se tienen diferentes tipos de sensores, los cuales pueden ayudar a la medición del flujo o nivel del caudal. Los sensores de flujo de caudal, los cuales funcionan bajo el principio de efecto hall, consisten en la medición de la cantidad de agua que fluye a través de ellos, obteniendo medidas en litros por minuto, por otra parte, estos sensores se aplican directamente en la tubería por la cual fluye el agua, es por ello, que se descarta el uso de estos sensores, ya que en el presente trabajo se requiere realizar una medición del nivel de agua del caudal del río, mas no del flujo de este.

Ahora, se toma en consideración las otras opciones disponibles, estas opciones nos permiten obtener la medida de nivel, ya que corresponden a sensores ultrasónicos y lo que realizan es mediciones de distancias, entre estos sensores existen diferentes opciones, la primera que se consideró utilizar es el sensor MB7076, el cual, es muy utilizado para aplicaciones de este tipo, en lo que se refiere a mediciones de nivel de agua en ríos y orientado a los sistemas de alerta temprana, para prevenir inundaciones debido al incremento en el nivel del caudal y desborde de los ríos. El sensor MB7076 es compatible con las placas de desarrollo de hardware libre, lo cual es una de sus ventajas, por el contrario, su costo es demasiado elevado y su stock es muy reducido, es por ello, que se descarta su uso en el presente proyecto.

Figura 29

Funcionamiento del Sensor MB7076



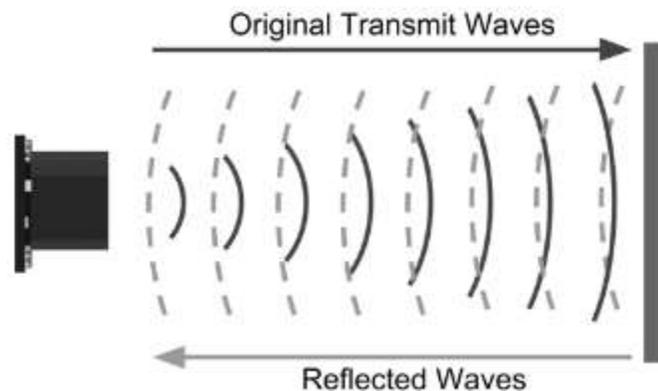
Fuente. (Geek Factory, 2022)

También, se consideran las opciones de los sensores MB1240 y MB1043, los cuales tiene el mismo principio de funcionamiento, la ventaja de estos sensores es que, de igual forma, son compatibles con placas de hardware libre y también que, su rango de medición es

mucho mayor que otros dispositivos, pero representan un costo bastante elevado por lo que no es factible la aplicación al presente proyecto.

Figura 30

Funcionamiento del Sensor MB1240 y MB1043



Fuente. (Geek Factory, 2022)

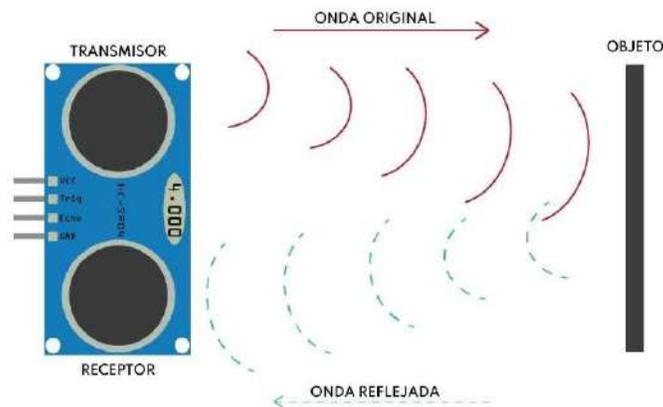
Con todo lo analizado anteriormente, se establece como propuesta para el desarrollo del presente proyecto el uso del sensor HC-SR04, debido a que trabaja bajo el principio de las ondas ultrasónicas, al igual que los sensores anteriormente analizados. Como se observó en la tabla comparativa, este sensor al igual que los otros, es compatible con las placas de hardware libre, las cuales se plantea usar en el proyecto, también, una ventaja sobre los demás es que, su precio es bastante reducido, por ello representa una opción viable, a pesar de que su rango de medición no es tan alto en comparación con los otros, se lo puede adaptar de manera correcta al fin del desarrollo del trabajo, que es medir el nivel de agua del caudal del río.

Funcionamiento del Sensor Ultrasónico HC-SR04. Para el desarrollo del proyecto como sensor se utiliza el sensor ultrasónico HC-SR04, este sensor corresponde a un elemento de bajo costo que permite medir la distancia a un objeto haciendo uso de ondas sonoras en

un rango de frecuencia de 40 kHz, las cuales son emitidas por sus transductores para la detección de objetos, el pulso emitido por el sensor lo realiza un transductor, mientras que el otro se encarga de recibir la señal de retorno que se genera al rebotar sobre una superficie.

Figura 31

Funcionamiento del Sensor HC-RS04



Fuente. (Geek Factory, 2022)

3.5.1.2. Análisis de las Placas Microcontroladoras

Las placas microcontroladoras nos permiten ejecutar instrucciones, las cuales se crean en algún entorno de programación y son escritas en base a un lenguaje de programación, dichas instrucciones se ejecutan en la placa microcontroladora e interactúa con los circuitos de esta. En este caso, se utiliza una placa microcontroladora para conectar los sensores ultrasónicos y para poder ejecutar la función definida de estos, la cual es realizar la medición de nivel de agua del caudal. Todo el procedimiento para que el sensor pueda obtener el valor de medida del nivel de agua se programa y se ejecuta en la placa microcontroladora.

En el mercado de la electrónica existen diferentes opciones para la utilización de hardware libre para placas microcontroladoras. A continuación, se detallan las opciones consideradas:

Tabla 9*Comparativa entre el microcontrolador ESP32 y Arduino UNO*

Parámetros Técnicos	Teensy 3.6	ESP32	ESP8266	Particle Photon	Thing Plus	Netduino N3	Arduino UNO	EFM32 Wonder Geecko	Artemis
Figura									
Voltaje de operación	3.3V	5V	5V	3.6V – 5V	2.3V – 3.6V	5V	5V	1.85V – 3.8V	5V
Consumo o de corriente	20 mA	20 mA	80 mA	80 mA	20 mA	40 mA	40 mA	-	-

Memoria Flash	1 MB	16 MB	4 MB	1 MB	16 MB	2 MB	32 kB	256 kB	1 MB
Memoria RAM	256 kB	520 kB	96 kB	128 kB	520 kB	256 kB	2 kB	32 kB	384 kB
Velocidad de reloj	180 MHz	Hasta 240 MHz	80MHz	120 MHz	240 MHz	180 MHz	16 MHz	48 MHz	48 MHz
Pines digitales I/O	4	36	17	18	21	22	14	10	24
Pines de entrada analógica	25	15	1	6	6	6	6	-	10

Wi-Fi	No	802.11 b/g/n	802.11b/g/ n	802.11b/g/n	802.11 b/g/n	802.11b/g/ n	No	No	No
Bluetooth	No	Si	No	No	Si	No	No	No	Si
Costo	\$52	\$14	\$10	\$40	\$40	\$70	\$16	\$100	\$35

Fuente. (Maker Guides, 2022) & (Texas Instruments Incorporated, 2022) & (ARDUINO®, 2022) & (Espressif Systems, 2023) & (Silicon Laboratories, 2023)

Observamos que cada una de las placas de desarrollo, poseen ciertas características que pueden tomarse como punto de partida para la selección de la placa microcontroladora a utilizar en el presente proyecto. En cuanto al voltaje, su rango de operación para todas las placas está entre los 3.3V hasta los 5V, otro de los parámetros importantes que podemos tomar en cuenta es la cantidad de memoria flash, que es la que nos permite almacenar los datos en la memoria después de apagar el dispositivo, y es donde se carga el programa que ejecuta las instrucciones para la funcionalidad de esta o de los elementos conectados a esta. También se toma en cuenta el

tamaño de memoria RAM, la cual permite una ejecución más rápida del programa cargado, por último, se toma en cuenta el número de pines, tanto analógicos como digitales, entre mayor número de pines, mayor es el número de elementos que pueden conectarse a la placa.

Con todo lo mencionado anteriormente, se tiene diferentes opciones, en primer lugar se toma en cuenta la placa de desarrollo Thing Plus, esta placa cuenta con una capacidad de memoria flash y memoria RAM considerablemente buenas, para el caso de la aplicación a sensores es una herramienta recomendable, además, posee otras tecnologías de comunicación inalámbrica que pueden hacer de la placa una herramienta útil en el campo de IoT.

Ahora, se analiza la placa Arduino UNO y Arduino NANO, estas placas son muy similares, pero poseen baja capacidad de memoria flash y RAM con respecto a otras de las placas mostradas, a pesar de ser una de las placas de desarrollo más utilizadas por los principiantes en el ámbito de la electrónica, es una placa muy simple que no tiene mayores capacidades en cuanto a las demás.

Las placas ESP32 y ESP8266, tienen características técnicas similares, ya que cuentan con la misma cantidad de memoria flash y se diferencian en la capacidad de memoria RAM, ya que la ESP32 cuenta con mayor capacidad. También, en cuanto a los pines que se pueden utilizar, la ESP32 cuenta con mayor número de pines para conectar elementos, a diferencia de la ESP8266.

En base a lo analizado respectivamente a las placas mostradas en la tabla anterior, se puede deducir que, la mejor opción para utilizar en el presente proyecto es la placa ESP32, esto debido a que presenta grandes capacidades respecto a las demás, contando así con alta capacidad de memoria flash y RAM, además, posee tanto tecnología WiFi como Bluetooth,

lo que la hace una placa completa para el desarrollo de aplicaciones IoT, finalmente, por su precio es la mejor opción, ya que es un costo factible y en base a sus características, es mejor que las demás placas mostradas.

3.5.1.3. Análisis de la Comunicación RF mediante LoRa

Para el desarrollo del proyecto, se requiere de comunicación inalámbrica entre los diferentes elementos de la red, como lo son: los sensores y el nodo central, para ello se ha optado por utilizar la tecnología de comunicación LoRa, la cual es utilizada en redes de sensores y en aplicaciones IoT, ya que provee de una baja transmisión de datos a largas distancias, siendo esta una de sus principales características, ya que se puede establecer conexión entre los dispositivos en el rango de kilómetros, utilizando las configuraciones necesarias para ello.

Los nodos sensores se conectarán de forma inalámbrica al nodo central para la transmisión de datos utilizando la tecnología LoRa, para ello, es necesario hacer uso de hardware que permita dicha función, así que, se utiliza módulos LoRa con el fin de añadir este componente a la red.

A continuación, se observa la comparativa entre los módulos LoRa disponibles:

Tabla 10*Comparación entre los módulos LoRa RFM95 y RA02*

Parámetros	LoRa1	RYL	E32-TTL-	Módulo RFM95	Ra-07H	Módulo RA02	RFM96	LLC C68
Técnicos		8	100					
Figura								
Modulación	LoRa	LoRa	LoRa	LoRa	LoRa	LoRa	LoRa	LoRa
Frecuencia de operación	433/470 /868/915 MHz	868 - 915 MHz	410 - 441 MHz	868 - 915 MHz	803 - 930 MHz	433 MHz	433 MHz	433 MHz
Velocidad de bits	0.018-62.5 Kbps	300 kbps	19.2 kbps	Hasta 300 kbps	Hasta 50 kbps	Hasta 300 kbps	Hasta 300 kbps	Hasta 300 kbps
Tensión de funcionamiento	3.3V – 3.7V	1.8 V – 3.7 V	2.3V – 5.5V	1.8 V – 3.7 V	3.3V	1.8 V – 3.7 V	1.8 V – 3.7 V	1.8 V – 3.7 V

ambiente								
o								
Sensibilidad	-148 dBm	-129 dBm	-130 dBm	Hasta -148 dBm	-137dBm	Hasta -148 dBm	-129 dBm	
Costo	\$17	\$15	\$17	\$10	\$13	\$14	\$8	\$30

Fuente. (VISTRONICA S.A.S, 2023)

De acuerdo a los parámetros técnicos observados de cada uno de los módulos LoRa considerados, se obtiene que para la mayoría de módulos, dichos parámetros son bastante similares, ya que son módulos muy económicos y factibles para aplicaciones domóticas, pero en este caso se selecciona el módulo LoRa RFM95, debido a que este permite en la frecuencia de los 915 MHz, lo ideal para su utilización en América, ya que está en la banda de US902-928, por otra parte, el módulo LoRa RA02 trabaja en los 433 MHz al igual que el RYLR998, LoRa1262, RFM96 y LLCC68, que no coincide con la banda para América. También, en cuanto al precio en comparación con los LoRa que, si trabajan en la banda de América, el módulo RFM95 es mejor, ya que ofrece similares características por un costo menor.

3.5.2. Etapa de control

En la etapa de control se realiza la recepción de los datos enviados desde los sensores, estos datos son almacenados y procesados para posteriormente enviarlos a la nube, donde podrán ser accedidos desde internet. En la etapa de control se requiere de un nodo central, al que se conecten los respectivos sensores, formando así una topología tipo estrella, este nodo central corresponde al Gateway LoRa, el cual es el encargado de la recepción,

almacenamiento y envío de los datos hacia los servidores en internet para su posterior visualización.

3.5.2.1. Análisis de Minicomputadoras para el Nodo Central

El Gateway LoRa consiste en un elemento central de la red, el cual es el encargado de recibir los datos provenientes de los nodos sensores conectados a él. En el mercado tecnológico existen dispositivos capaces de cumplir con esta función, estos dispositivos son fabricados y distribuidos por la empresa encargada de LoRa que es SEMTECH, es por ello, que se tiene dispositivos orientados directamente a este fin. A continuación, se detallan las especificaciones de los LoRa Gateway disponibles en el mercado:

Tabla 11

Comparación entre los LoRa Gateways disponibles en el mercado.

Parámetros Técnicos	XHCIOT	DRAGINO LG308N	LPS8V2	LG1301- PF
Figura				
Voltaje de operación	8 – 28V	5V	5V	12V
Frecuencia	220 MHz – 930 MHz	868 – 915 MHz	868 – 915 MHz	433 – 915 MHz

Distancia de comunicación	5 – 10 km	5 km	5 km	5 – 10 km
Protocolos	TCP, UDP, MQTT, HTTP	UDP, MQTT	TCP, UDP, MQTT, HTTP	TCP, UDP, MQTT, HTTP
Costo	\$80	\$70	\$200	\$270

Fuente. (VISTRONICA S.A.S, 2023)

En base a lo analizado en la tabla anterior, se encuentran diferentes modelos de LoRa Gateway que trabajan en varias frecuencias y tiene funcionalidades específicas, debido a que son dispositivos fabricados con el fin de cumplir la función de un LoRa Gateway su precio es bastante elevado por lo que se adquisición no es factible. Entonces, se propone implementar un LoRa Gateway utilizando hardware de uso libre y el módulo LoRa RFM95.

En el mercado tecnológico se encuentran varias opciones para la selección de dicho elemento de la red, existen dispositivos específicos para el uso de la tecnología, fabricados por la empresa a la que pertenece LoRa, en este caso los costos para la adquisición de estos equipos pueden ser considerablemente elevados, como se había mencionado anteriormente. Por otra parte, se puede utilizar hardware libre para la implementación, ya que existen placas como minicomputadoras con herramientas y capacidad de procesamiento eficiente para aplicaciones IoT. A continuación, se establece una tabla de especificaciones técnicas para las opciones consideradas:

Tabla 12*Comparación entre los módulos LoRa RFM95 y RA02*

Parámetros Técnicos	Raspberry Pi 4	Orange Pi PLUS	ASUS Tinker Board S	LePotato	Rock64 Media Board	Banana Pi M64
Figura						
CPU	Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72	H3 Quad-core Cortex-A7	Cortex-A17 Quad-core 1.8GHz	Amlogic S905X SoC	Rockchip RK3328	Amlogic S905X3
GPU	VideoCore VI	Mali400M P2 @ 600MHz Soporta OpenGL ES 2.0	ARM Mali-T764 - Compatible con 4K	Mali-450 MP3	Mali 450 MP2	Mali-G31

Núcleos	4	4	4	4	4	4
Memoria	1/2/4GB LPDDR4 RAM	DDR3 de 1GB (compartido con GPU)	Dual-CH LPDDR3 2GB	1, 2 GB de DDR3 SDRAM	1, 2 y 4 GB de RAM LPDDR3	4 GB de LPDDR4
Conectividad	802.11ac Wi-Fi / Bluetooth 5.0, Gigabit Ethernet	Realtek RTL8189E TV, IEEE 802.11 b/g/n	Wi-Fi 802.11 b/g/n, Bluetooth 4.0, Gigabit Ethernet	100 Mb de Fast Ethernet	Gigabit Ethernet	10/100/1000 Mb/s de Ethernet
Video	HDMI	HDMI	HDMI	HDMI, GPIO, PWM, I2C, SPI, cabezal de audio, cabezal UART,	HDMI 2.0a, audio/micrófono de 3,5 mm, x GPIO de 40 clavijas, GPIO de 22 clavijas	HDMI 2.0, audio/micrófono de 3,5 mm, 28 x GPIO, UART, I2C, SPI, PWM

	receptor					
	IR					
Puertos	2 x USB 3.0, 2 x USB 2.0	3 x USB 2.0 HOST, 1 x USB 2.0 OTG	1 x HDMI de tamaño completo y 4 x USB 2.0	USB 2.0	2 x USB 2.0, 1 x USB 3.0	USB 3.0, 1GbE puerto LAN
Alimentación	5V/3A vía USB-C, 5V vía cabezal GPIO	La entrada de CC, la entrada USB OTG no suministra energía	microUSB o por pines	micro USB de 5 V	5 V a 3 A (a través de puerto de barril de 3,5mm, PoE)	5 V a 2 A a través de UBC C
Costo	\$94	\$81,99	\$129,22	\$ 70,89	\$85,92	\$75,18

Fuente. (VISTRONICA S.A.S, 2023)

En base a los dispositivos analizados en la tabla anterior, se puede destacar que este tipo de placas tienen altas capacidades de hardware y procesamiento en comparación con las placas microcontroladoras, existe una variedad de dispositivos en el mercado que puedan ayudar a cumplir la función que se requiere, los costos varían en cuanto al fabricante y las capacidades de cada placa, la placa seleccionada corresponde a la Raspberry Pi 4, que por su capacidad y precio es la más factible a utilizar, además de ser una de las más utilizadas en el mundo. Esta placa adicionalmente estará conectada a un Arduino Uno, esto debido a que la

Raspberry PI 4 no posee compatibilidad con el módulo LoRa RFM95 y puede presentar problemas con su programación y funcionamiento, mediante comunicación serial entre los dos elementos, el Arduino UNO enviará los datos receptados de los nodos hacia la Raspberry PI 4.

3.5.3. Etapa de visualización

Esta etapa corresponde a la presentación de los datos, para ello se utiliza plataformas disponibles en la nube, que ayudan al procesamiento y almacenamiento de los datos para su posterior presentación, en este caso existen herramientas para la creación de plataformas y dashboard para observar las mediciones realizadas por los sensores.

A continuación, se observa las plataformas disponibles para aplicaciones IoT:

Tabla 13

Comparación entre las plataformas IoT disponibles

Características	Microsoft Azure	AWS	IBM Watson IoT	ThingSpea k	Google Cloud
Protocolos	HTTP, AMQP, MQTT	HTTP, WebSockets , MQTT	HTTP, MQTT	HTTP, MQTT, CoAP,	HTTP, MQTT
Hardware	Intel,	Broadcom,	Arm, Texas	Intel,	Broadcom,
Certificado	Raspberry, FreeScale,	Marvell, Texas	instruments , Raspberry, Arduino.	Raspberry, FreeScale,	Marvell, Texas

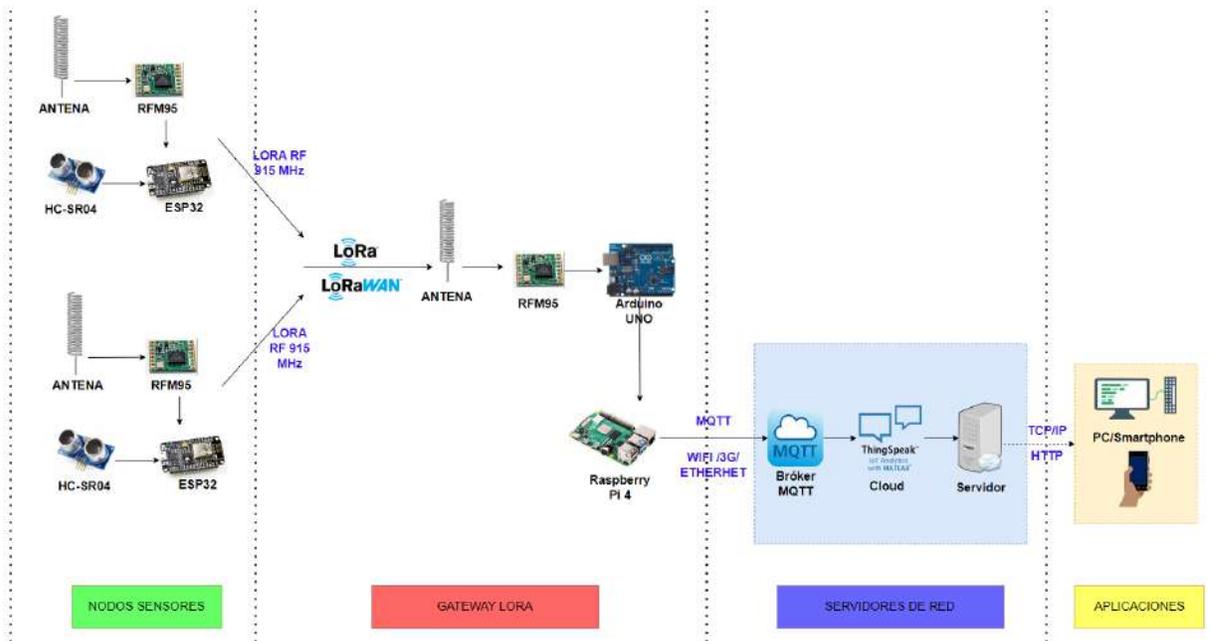
	Texas Instruments	Intel	Texas Instruments	Intel
Lenguajes	.Net, Java, NodeJS	Java, C, NodeJS	C#, C, Phyton, Java, NodeJS	Phyton, C#, C, Phyton, Java, NodeJS
Precio	Pagado, depende de la suscripción	Pagado, depende de la suscripción	Pagado, depende de la suscripción	Ofrece una versión gratuita no comercial

Fuente. (Durand, Visagie, & Booyesen, 2019)

En base a la tabla presentada, se tiene diferentes opciones para seleccionar la plataforma a utilizar, en este caso, todas las plataformas están orientadas a lo que es redes de sensores para aplicaciones en IoT, todas las plataformas cuentan con diferentes características que permiten llevar a cabo la conexión de la red de sensores con la nube, en su gran mayoría ofrecen soporte para la mayoría de los lenguajes de programación y placas de desarrollo. Otro punto importante es la suscripción, ninguna plataforma es gratuita por lo que se requiere de realizar un pago para hacer uso de sus herramientas, es por ello, que en este caso se selecciona ThingSpeak, ya que se puede utilizar la versión no comercial, además ofrece conexión mediante MQTT y presenta facilidad para configuraciones.

3.6. Arquitectura del Sistema

En base a los requerimientos del sistema analizados dónde se seleccionó el hardware y software adecuado para el sistema, en la siguiente figura se representa la arquitectura del sistema, que corresponde a una red en estrella y que consta de los nodos recolectores como primer elemento, los cuales corresponden a sensores ultrasónicos para la medición de nivel de agua, los mismos se conectan a una placa ESP32, el cual hace las veces de microcontrolador que contiene el código de programación para la función de los sensores. Se utiliza la tecnología de comunicación LoRa, para ello se utiliza módulos LoRa RF95, para comunicar los sensores con el Gateway LoRa, que lo conforma un Arduino UNO con una Raspberry Pi 4 que se comunican mediante el puerto serial. Finalmente, se tiene la etapa de procesamiento de datos en la nube y la visualización mediante la plataforma de IoT ThingSpeak para el posterior acceso de los dispositivos finales.

Figura 32*Arquitectura del sistema planteado**Fuente. Autoría*

3.7. Diagramas de interconexión

En base a lo analizado en los requerimientos de hardware, una vez seleccionados los componentes a utilizar para el desarrollo de la red de sensores, se procede a presentar los diagramas de conexión de cada elemento para su correcto funcionamiento en cada uno de los elementos, como el nodo sensor y el nodo Gateway LoRa.

3.7.1. Nodo Sensor

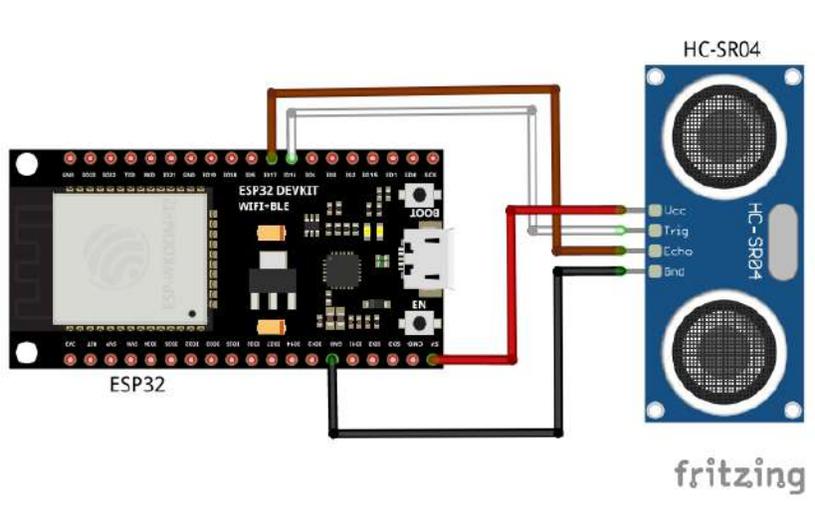
El nodo sensor consiste en una combinación del microcontrolador ESP32 y el sensor ultrasónico, el cual ayuda a la medición de niveles de agua de un caudal, en el diagrama a continuación, se visualiza las conexiones realizadas entre los dos elementos para el respectivo funcionamiento.

A continuación, se detalla la conexión de cada elemento a los pines correspondientes:

- Pin Vcc de alimentación del sensor HC-SR04, se conecta al pin de alimentación de 5V del ESP32.
- Pin Trigger del sensor HC-SR04 se conecta al pin 16 del ESP32.
- Pin Echo del sensor HC-SR-04 se conecta al pin 17 del ESP32.
- Pin GND correspondiente a tierra del sensor HC-SR04, se conecta al ping GND del ESP32.

Figura 33

Diagrama de conexión sensor HC-SR04 y ESP32



Fuente. Autor

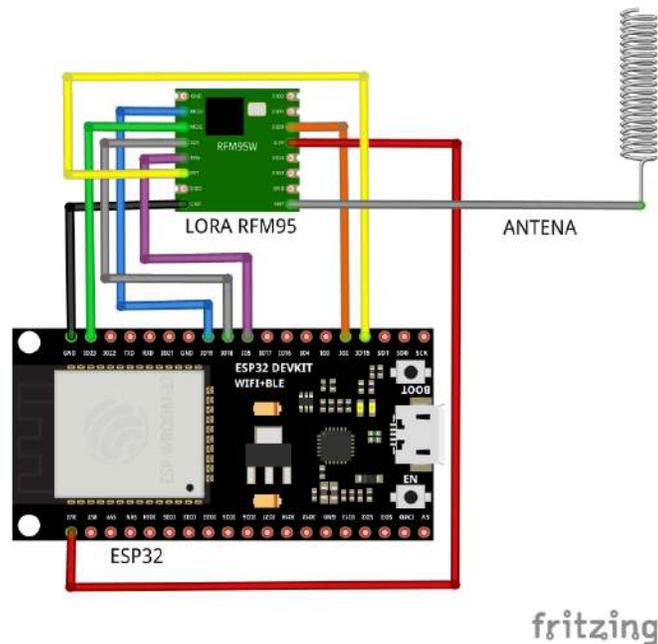
El nodo sensor debe utilizar comunicación inalámbrica para poder conectarse al Gateway LoRa o nodo central, para ello, se requiere del módulo LoRa RFM95, el cual provee la capacidad de utilizar la tecnología inalámbrica LoRa para establecer conexión a grandes distancias, dicho módulo se conecta a la placa microcontroladora ESP32 para enviar las mediciones realizadas por el sensor ultrasónico HC-SR04.

A continuación, se observa el diagrama de conexión realizado para el funcionamiento del módulo RFM95 con la placa ESP32 y se detallan las conexiones realizadas:

- El pin de alimentación del módulo RFM95 se conecta al pin de 3.3V en el ESP32.
- El pin de GND o tierra del módulo RFM95 se conecta al pin de GND de la placa ESP32.
- El pin MISO del módulo RFM95 se conecta al pin IO19 de la placa ESP32.
- El pin MOSI del módulo RFM95 se conecta al pin IO18 de la placa ESP32.
- El pin de SCK correspondiente a la señal de reloj del módulo RFM95 se conecta al pin IO5 de la placa ESP32.
- El pin NSS del módulo RFM95 se conecta al pin IO18 de la placa ESP32.
- El pin RST del módulo RFM95 se conecta al pin IO14 de la placa ESP32.
- El pin DIO0, DIO1 y DIO2 del módulo RFM95 se conecta al pin IO26, IO33 e IO32 respectivamente.
- El pin ANT se conecta a la antena que, en base a su tamaño, funciona a la frecuencia requerida.

Figura 34

Diagrama de conexión módulo LoRa RFM95 y ESP32



Fuente. Autor

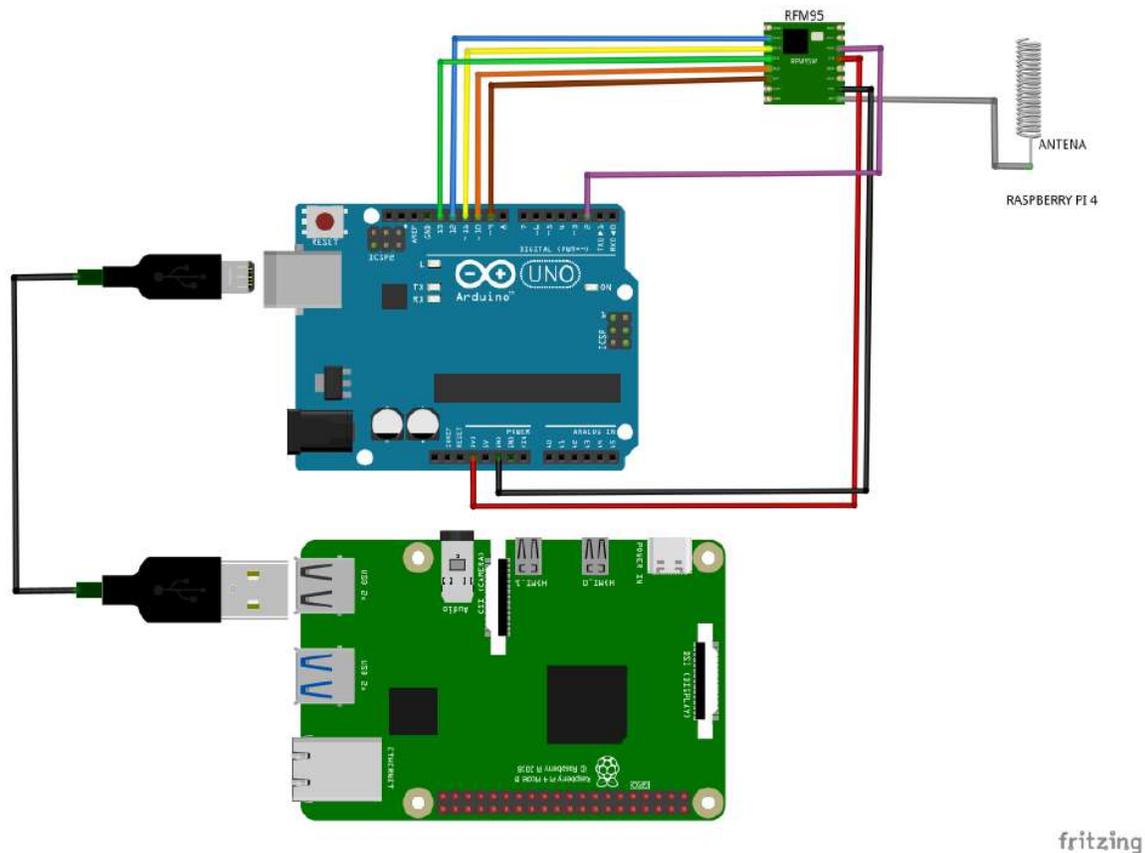
3.7.2. Nodo Gateway LoRa

El nodo central es el elemento de la red a la cual se conectan todos los nodos sensores, este cumple la función de recibir dichos datos obtenidos de los sensores para luego ser enviados a internet, en este caso se elabora un Gateway LoRa utilizando una Raspberry Pi y un Arduino UNO, la Raspberry Pi, tiene capacidades elevadas de procesamiento con referencia a otras placas de desarrollo. Este elemento sirve como Gateway dentro de la red para establecer comunicación con los nodos sensores que requiere de la tecnología de comunicación LoRa, para lo cual se ha realizado la conexión con el módulo RFM95 al Arduino UNO como se observa en el siguiente diagrama y este mediante conexión serial se conecta a la Raspberry Pi, a continuación, se detallan las conexiones realizadas:

- El pin de alimentación de 3.3V del módulo RFM95 se conecta al pin 3.3V Del Arduino UNO.
- El pin GND correspondiente a tierra del módulo RFM95 se conecta a GND del Arduino UNO.
- El pin DIO0 del módulo RFM95 se conecta al pin 2 del Arduino UNO.
- El pin RST del módulo RFM95 se conecta al pin 9 del Arduino UNO.
- El pin NSS del módulo RFM95 se conecta al pin 10 del Arduino UNO.
- El pin MOSI del módulo RFM95 se conecta al pin 11 del Arduino UNO.
- El pin MISO del módulo RFM95 se conecta al pin 12 del Arduino UNO.
- El pin de SCK correspondiente a la señal de reloj del módulo RFM95 se conecta al pin 13 del Arduino UNO.
- El pin ANT se conecta a la antena que, en base a su tamaño, funciona a la frecuencia requerida.
- El Arduino UNO se conecta mediante cable USB hacia la Raspberry PI para leer los datos por comunicación serial.

Figura 35

Diagrama de conexión del Gateway LoRa



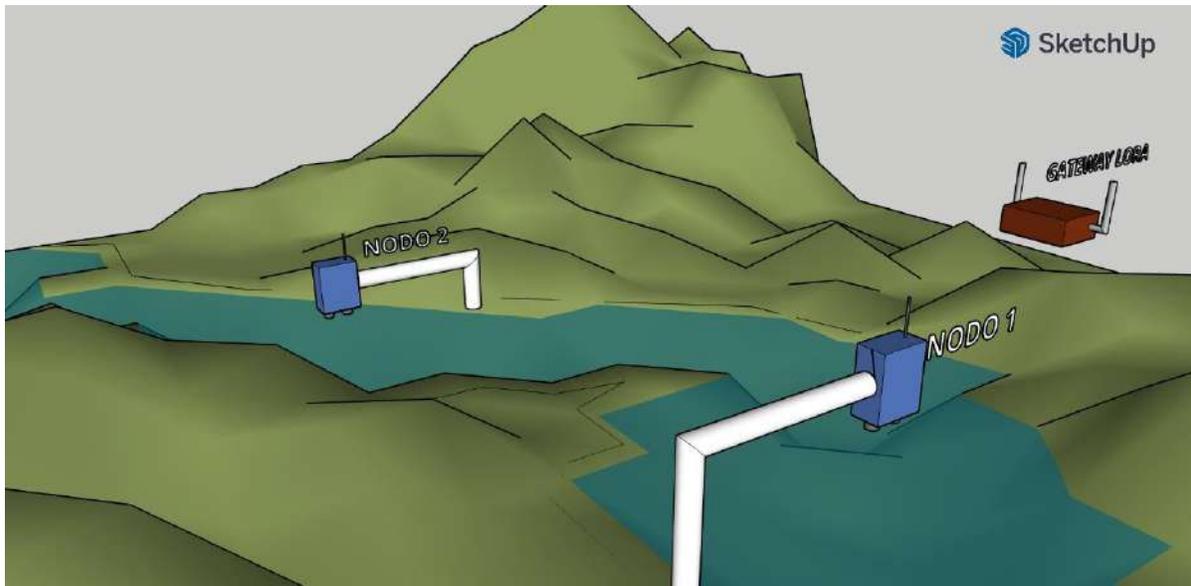
Fuente. Autor

3.7.3. Simulación del Armado del Sistema

En base a todos los requerimientos analizados se procede a mostrar un esquema gráfico de cómo se implementaría en la zona deseada el sistema de monitoreo, para ello se cuenta con dos nodos sensores, los cuales se ubican en dos posiciones diferentes a lo largo del cauce del río, pudiendo así tomar mediciones de dos puntos diferentes del nivel de agua, los dos sensores se conectan a un Gateway para poder enviar y visualizar la información, conformando así una red estrella, que es una arquitectura de red que es soportada por LoRa, en la siguiente figura puede observarse lo mencionado.

Figura 36

Simulación de la disposición de los nodos sensores y el gateway LoRa.

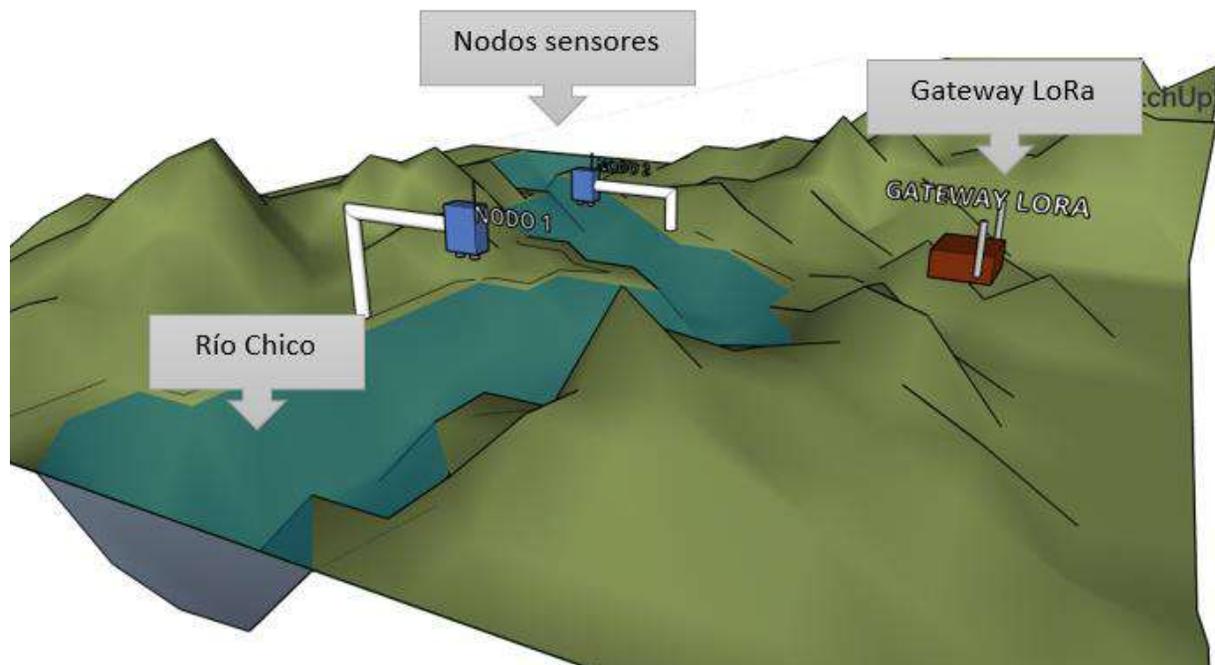


Fuente. Autor

Los dos nodos toman la medición de nivel de agua y lo envían al gateway, el cual es el elemento de la red formando así un topología en estrella para la comunicación inalámbrica utilizando la tecnología LoRa. En la siguiente figura, se tiene otra vista de la posible distribución de los nodos sobre el cauce del río en donde se observa el nodo 1, nodo 2 y el Gateway LoRa.

Figura 37

Vista de los nodos sensores y el gateway LoRa en simulación de Río Chico



Fuente. Autor

Para el nodo sensor, se ha realizado la simulación de la posible distribución de los componentes que lo conforman en base a los requerimientos de hardware seleccionados anteriormente, en la siguiente figura se observa los componentes dentro del case, los cuales corresponden a: microcontrolador ESP32, módulo LoRa RFM95, sensor HC-SR04 y batería de litio.

Figura 38

Simulación de la distribución de los elementos del nodo sensor en el case



Fuente. Autor

Para el Gateway LoRa, se ha realizado la simulación de la posible distribución de los componentes que lo conforman en base a los requerimientos de hardware seleccionados, en la siguiente figura se observa los componentes dentro del case, los cuales corresponden a: placa Raspberry PI 4, microcontrolador Arduino UNO y módulo LoRa RFM95.

Figura 39

Simulación de la distribución de los elementos del gateway LoRa en el case



Fuente. Autor

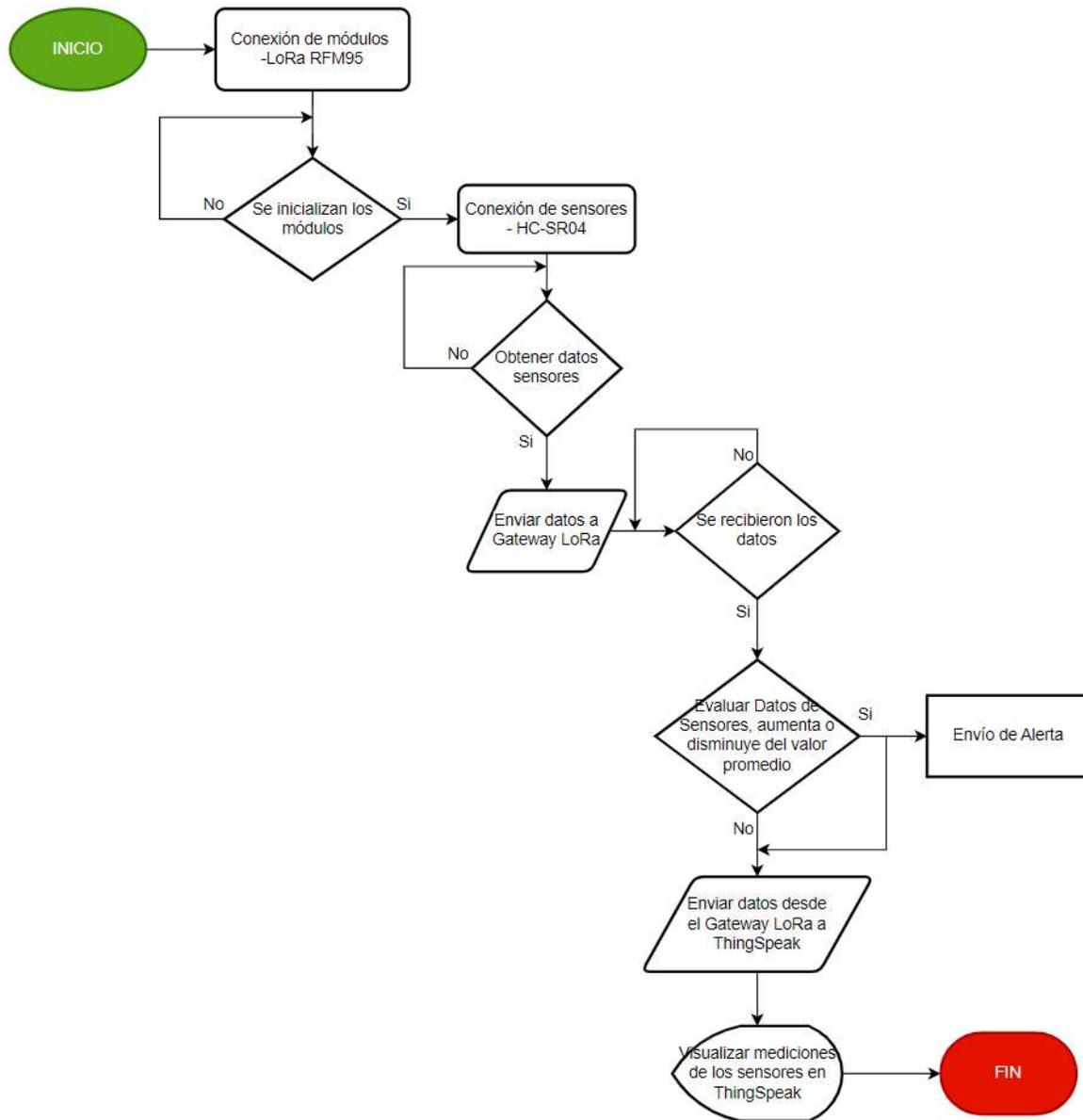
3.8. Diseño del Prototipo

A continuación, se describe el desarrollo completo del sistema en base a los requerimientos analizados anteriormente, se establecen las configuraciones realizadas en cada uno de los equipos utilizados para cada elemento de la red con el fin de implementar la funcionalidad del sistema planteado.

3.8.1. Diagrama de flujo del sistema

El diagrama de flujo del sistema consiste en el proceso que se lleva a cabo para el funcionamiento de este y se presentan los diferentes procesos que cumple. El proceso del sistema inicia con la conexión de los módulos LoRa RFM95 al microcontrolador ESP32 mediante los pines digitales, los módulos permiten establecer la comunicación inalámbrica utilizando la tecnología LoRa, el siguiente proceso es verificar que se hayan inicializado correctamente los módulos, ya que sin ellos no se puede establecer conexión hacia el

Gateway LoRa. A continuación, se establece la conexión de los sensores ultrasónicos mediante los pines del microcontrolador ESP32 para obtener la medida del nivel del caudal del río, se verifica que se obtengan datos de los sensores para posteriormente ser enviados al Gateway LoRa, el cual se compone de una placa Arduino UNO conectado por el puerto serie a la Raspberry Pi 4, se verifica que los datos se reciban en este elemento de la red y se evalúa si el nivel aumentó o disminuyó en relación al valor promedio que se mida, para de esta forma poder enviar una alerta mediante correo electrónico, posteriormente se inicia el proceso de envío de datos a la nube utilizando la plataforma de ThingSpeak, en donde finalmente se podrá acceder a los datos desde internet.

Figura 40*Diagrama de flujo del sistema**Fuente. Autoría*

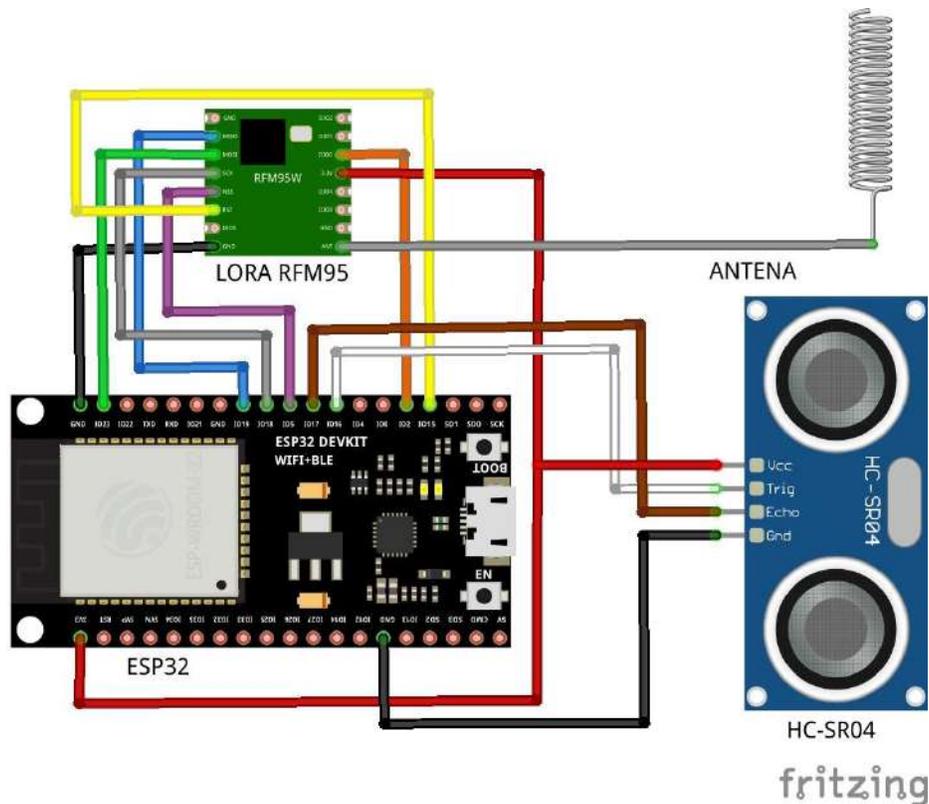
3.8.2. Configuración del Nodo Sensor

El nodo sensor cuenta con la conexión de los diferentes elementos que se muestran en la siguiente figura hacia el microcontrolador ESP32, estos corresponden al módulo LoRa RFM95 y al sensor ultrasónico HC-SR04. Este nodo tiene la funcionalidad de tomar datos

del nivel del caudal del río y mediante el módulo LoRa establece la comunicación de largo alcance con el Gateway LoRa.

Figura 41

Diagrama de conexión del nodo sensor



Fuente. Autoría

Para la conexión del módulo LoRa RFM95 hacia el microcontrolador ESP32 se utiliza la interfaz de comunicación SPI (Serial Peripheral Interface), la cual es una interfaz de comunicación síncrona que conecta varios dispositivos periféricos a los microcontroladores. Para el módulo RFM95, mediante SPI se puede establecer conexión con el microcontrolador ESP32, los pines que están vinculados a esta interfaz son los siguientes:

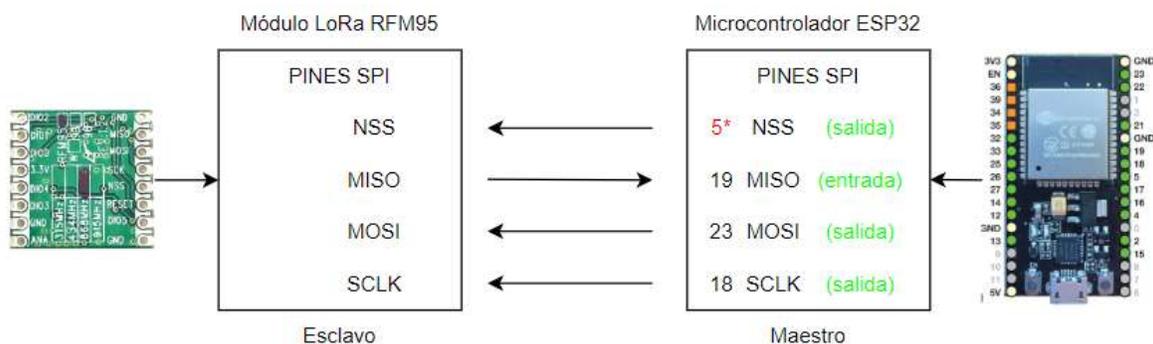
- MOSI (Master Out Slave In): se utiliza para enviar datos desde el maestro (microcontrolador) al esclavo (módulo LoRa).

- MISO (Master In Slave Out): se utiliza para enviar datos desde el módulo LoRa microcontrolador.
- SCLK (Serial Clock): proporciona la señal de reloj de sincronización para la transferencia de datos.
- NSS (Slave Select): se utiliza para seleccionar el dispositivo esclavo al cual el maestro va a comunicarse.

Los pines de comunicación SPI en el microcontrolador viene definidos por el fabricante, de esta forma se debe ser cuidadoso al momento de conectar el módulo RFM95, ya que si no se utiliza los pines correspondientes para cada pin SPI no funcionará correctamente. Todos los pines deben conectarse al pin adecuado a excepción del pin NSS, el cual puede conectarse a cualquier pin del ESP32 y ser configurado por software.

Figura 42

Comunicación SPI entre el módulo LoRa RFM95 y el microcontrolador ESP32



Fuente. Autoría

3.8.2.1. Programación ESP32

Para el nodo sensor se utiliza la placa ESP32 como microcontrolador, este será el que permita la conexión de los dispositivos periféricos, en este caso el módulo LoRa RFM95 y el sensor HC-SR04 para la toma de datos del nivel de caudal del río. Para la funcionalidad

de cada uno se requiere utilizar un programa, el cual se cargue al microcontrolador y permita ejecutar la funcionalidad designada para el nodo sensor, es decir, permita la toma de datos con el sensor HC-SR04 y la comunicación inalámbrica utilizando el módulo LoRa RFM95.

Para la programación se utiliza el IDE Arduino, este nos permite crear un código de programación en C++, con todas las instrucciones a ejecutar. En primer lugar, para el código del nodo sensor se tiene la inclusión de las librerías a utilizar:

Figura 43

Inclusión de librerías SPI y LoRa

```
4 //Inclusión de librerías
5 #include <SPI.h>
6 #include <LoRa.h>
```

Fuente. Autoría

- SPI.h: esta librería corresponde a la interfaz de comunicación para facilitar la conexión entre el microcontrolador y el módulo LoRa como se había explicado anteriormente, cuando se la utiliza hay que conectar cada uno de los pines del módulo al microcontrolador correctamente.
- LoRa.h: esta librería permite establecer la funcionalidad para el módulo LoRa RFM95, las configuraciones necesarias para el módulo se obtienen de la librería haciendo el llamado a cada una de las funciones que contiene dicha librería.

A continuación, se definen los pines NSS, RST y DIO0 del módulo RFM95 que se utilizan para la inicialización de este haciendo uso de la librería LoRa.h.

Figura 44

Pines de conexión del Módulo LoRa RFM95 a ESP32

```

-
8 //Definir los pines para la conexión del módulo RFM95
9 #define nss 5
10 #define rst 15
11 #define dio0 2

```

Fuente. Autoría

Se definen las variables para la configuración de los parámetros de la comunicación LoRa, se toma en cuenta los tres parámetros fundamentales para LoRa, como lo son: factor de dispersión (SF), ancho de banda (BW) y tasa de codificación (CR). Las configuraciones se definen con un valor de 7 para SF, el BW se define en 125 kHz y el CR en 4/5.

Figura 45

Variables para la configuración de parámetros de la comunicación LoRa

```

16 // Variables para configuraciones personalizadas del módulo RFM95 LoRa
17 const int sf = 7; // Spreading Factor (7-12)
18 const int bw = 125000; // Bandwidth (Hz)
19 const int cr = 5; // Coding Rate

```

Fuente. Autoría

Mediante la librería LoRa.h se configuran los pines para el módulo LoRa a los definidos al inicio del código, posteriormente se inicializa LoRa en la frecuencia definida para la banda en la que se desea trabajar, en este caso se define en la frecuencia 915MHz. Se inicializan los parámetros de la comunicación LoRa con las variables anteriormente mencionadas.

Se establece como valor inicial un valor de SF de 7, que indica mayor velocidad de transmisión, menor tiempo en el aire del paquete y menor alcance, un BW de 125kHz que ofrece mayor sensibilidad al receptor y menor ruido, el CR se establece en 4/5 que indica que, por 4 bits de información útil, el codificador genera en total 5 bits y de estos 1 bit es

redundante, esto implica menor sobrecarga. Finalmente, se habilita el CRC para la corrección de errores.

Figura 46

Parámetros de la transmisión LoRa

```

44 //define los pines para el módulo RFM95
45 LoRa.setPins(nss, rst, dio0);
46
47 //Inicializa LoRa en la frecuencia definida
48 while (!LoRa.begin(915E6)) {
49   Serial.println("Falló. Revise las conexiones del módulo LoRa RFM95");
50   delay(500);
51 }
52
53 //Inicialización de parámetros LoRa para la Tx
54 LoRa.setSpreadingFactor(sf); //Factor de dispersión (7-12)
55 LoRa.setSignalBandwidth(bw); //Ancho de banda (125-250-500 kHz)
56 LoRa.setCodingRate4(cr); //Coding Rate
57 LoRa.enableCrc(); // Habilita la comprobación de redundancia cíclica (CRC)

```

Fuente. Autoría

Se inicializa el paquete LoRa con la función `LoRa.beginPacket()`, se envía la dirección del nodo sensor hacia el receptor o nodo central para identificarlo y posteriormente se envía el valor tomado por el sensor, el paquete finaliza con la función `LoRa.endPacket()` que cierra el paquete. Los datos para enviar deben estar dentro de las dos funciones como se muestra en la figura:

Figura 47

Paquete LoRa para transmitir

```

99 //Paquete LoRa con los datos a enviar
100 LoRa.beginPacket(); //inicializa el paquete
101 LoRa.write(nodeAddress); //se envia la dirección del nodo
102 LoRa.print(nivelCm); //se envia el valor medido
103 LoRa.endPacket(); //finaliza el paquete
...

```

Fuente: Autoría

Para el sensor HC-SR04 se inicia definiendo los pines echo, que envía la señal ultrasónica y el pin trigger, que recibe el rebote de la señal enviada. También, se definen las

variables para almacenar los valores que se tomen mediante el sensor y utilizarlos para el cálculo del nivel.

Figura 48

Pines y variables para el sensor ultrasónico HC-SR04

```
22 //Definir los pines para el sensor HC-SR04
23 const int trigPin = 16; //pin trigger
24 const int echoPin = 17; //pin echo
25
26 //variables para almacenar valores medidos por el sensor HCR-SR04
27 long duracion; //duración del pulso
28 float nivelCm; //Distancia medida hacia el objeto y nivel de agua
29 float distanciaMax = 300; //distancia máxima de medición
```

Fuente. Autoría

El sensor calcula la distancia enviando una onda ultrasónica hacia un objeto, esta rebota y se recibe nuevamente, se calcula el tiempo que demora en ir y volver dicha onda, de esta forma se obtiene la distancia hacia dicho objeto. Ahora, con dicho valor de distancia hacia el objeto se ejecuta una función que se realiza hasta tomar un total de 5 muestras, de las cuales se obtiene el promedio y a este valor se lo resta de la distancia máxima, obteniendo así el valor del nivel del agua del río en centímetros, este valor es el que se envía en el paquete LoRa.

Figura 49

Cálculo de distancia hacia el objeto y valor de nivel.

```

86 // Se calcula la distancia y se obtiene el promedio para 5 muestras
87 for(int i = 0; i < muestras; i++){
88     nivelCm = duracion * velSonido/2;
89     promedio = promedio + nivelCm;
90 }
91 //Se obtiene el nivel restando la medida obtenida menos la distancia máxima
92 nivelCm = distanciaMax-(promedio/muestras);

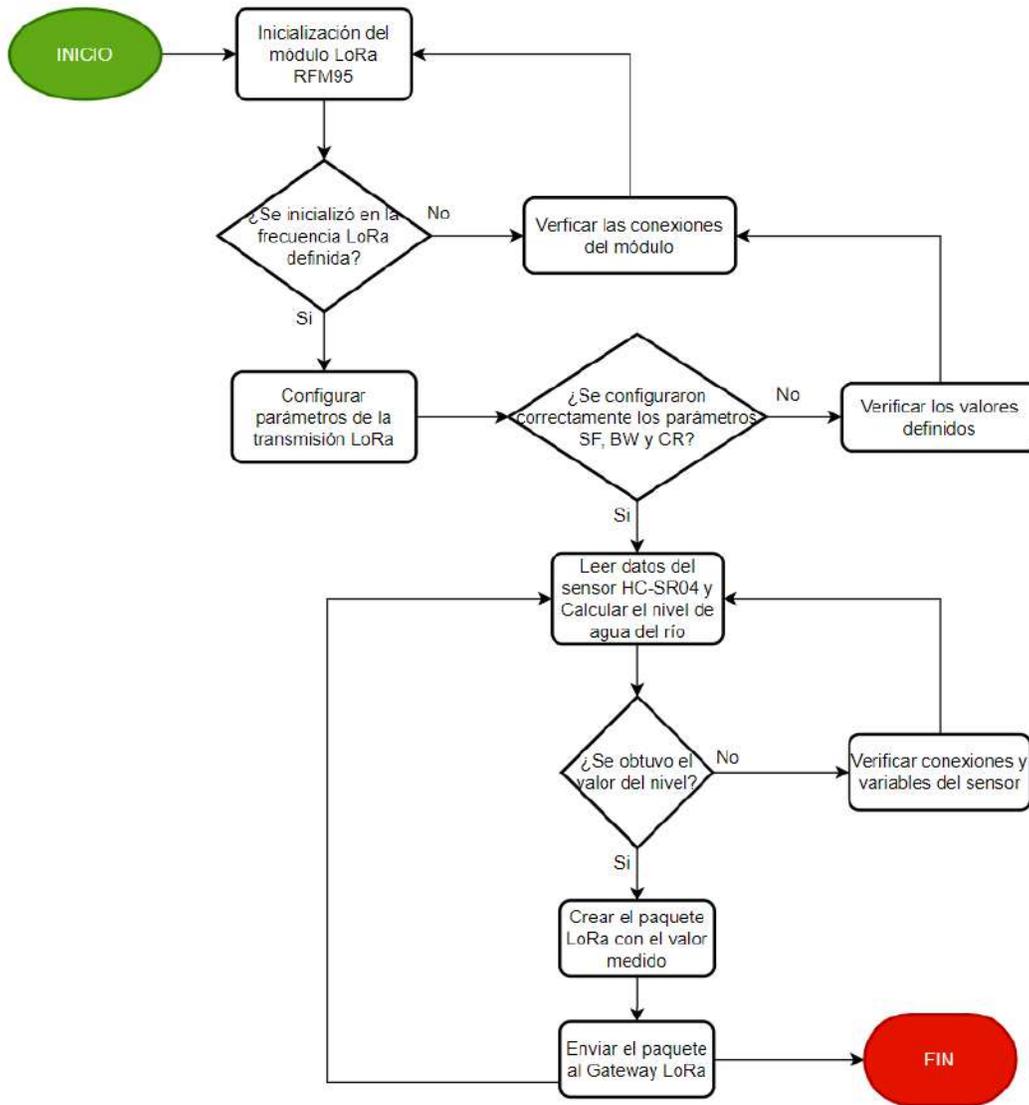
```

Fuente. Autoría

A continuación, se muestra el diagrama de flujo del código para el nodo sensor, el proceso empieza con la inicialización del módulo RFM95, si no se establece en la frecuencia LoRa definida se debe verificar las conexiones del módulo e inicializar el módulo nuevamente, si se establece correctamente se configuran los parámetros para la comunicación LoRa, si no se configuran correctamente se deben verificar e inicializar el módulo nuevamente. Si se configura correctamente el módulo se procede a la captura de datos del sensor, si se obtiene el valor de nivel correctamente se construye el paquete LoRa y se envía al Gateway LoRa correspondiente, este proceso se repite constantemente y envía el dato obtenido cada 10 segundos.

Figura 50

Diagrama de flujo del nodo sensor



Fuente. Autoría

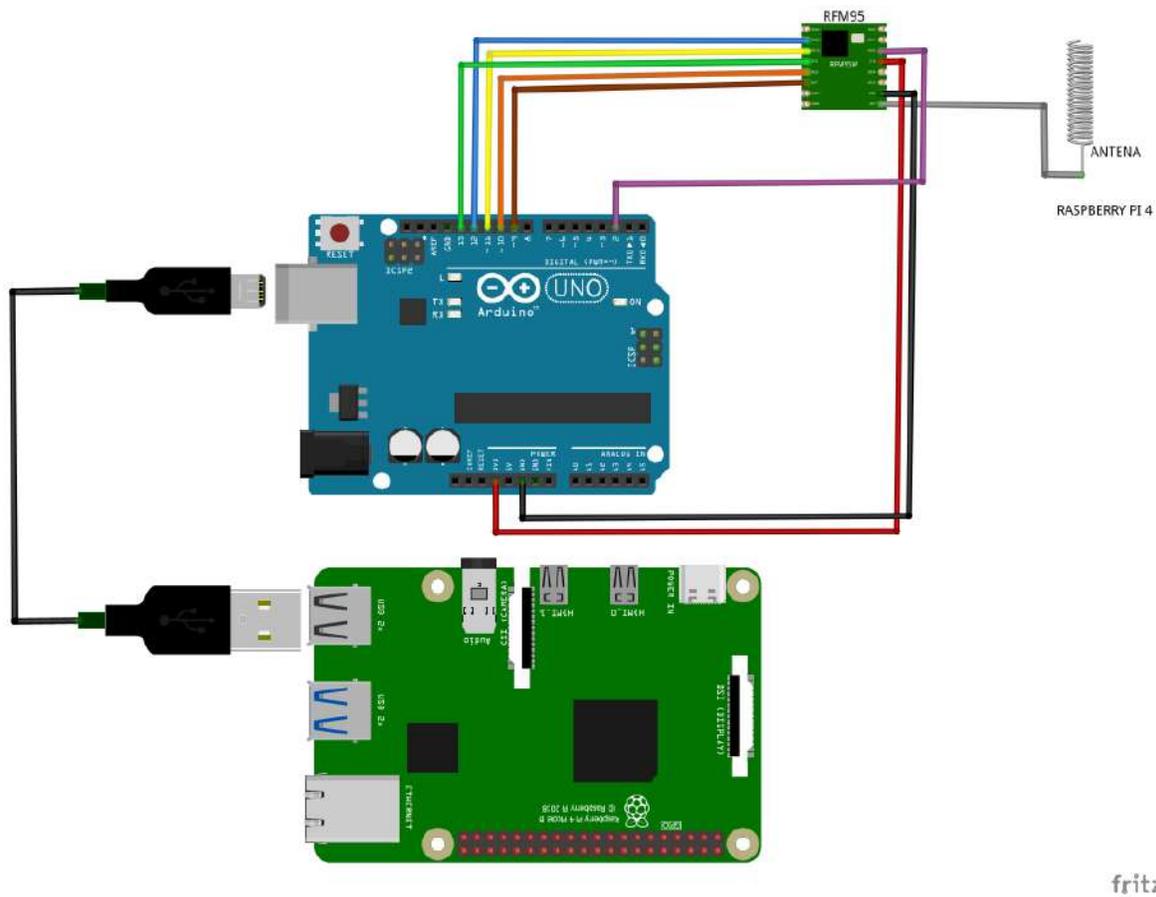
3.8.3. Configuración Gateway LoRa

El nodo central o Gateway LoRa está formado por el módulo RFM95, el cual se conecta al microcontrolador Arduino UNO, ya que de esta forma es mucho más fácil configurarlo debido a las librerías que se utilizan en Arduino. Mediante comunicación serial se conecta el Arduino UNO a la Raspberry PI 4, se utiliza un cable USB para comunicar los

dos elementos, de esta forma se conforma el Gateway LoRa, el cual cumple la función principal de recibir los datos de los nodos sensores, procesarlos y enviarlos a la nube o a la plataforma IoT con la que se trabaja, en este caso ThingSpeak. A continuación, se puede observar la conexión de los dispositivos mencionados:

Figura 51

Diagrama de conexión Gateway LoRa



Fuente. Autoría

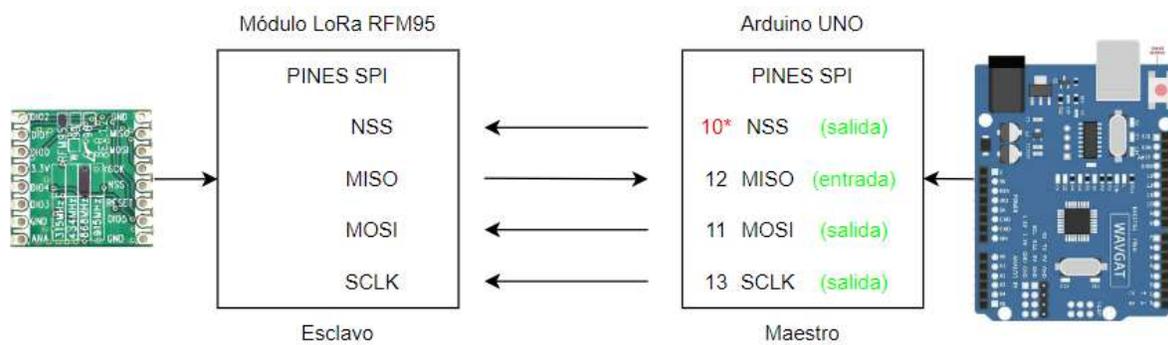
Para la conexión del módulo LoRa RFM95 hacia el Arduino UNO, se utiliza la interfaz de comunicación SPI (Serial Peripheral Interface), la cual es una interfaz de comunicación síncrona que conecta varios dispositivos periféricos a los microcontroladores. Los pines de comunicación SPI en el microcontrolador vienen definidos por el fabricante, de

esta forma se debe ser cuidadoso al momento de conectar el módulo RFM95, ya que si no se utiliza los pines correspondientes para cada pin SPI no funcionará correctamente.

A continuación, se presenta un diagrama que evidencia la conexión SPI entre el módulo RFM95 y Arduino UNO:

Figura 52

Comunicación SPI entre el módulo LoRa RFM95 y Arduino UNO



Fuente. Autoría

3.8.3.1. Programación Arduino UNO

Uno de los componentes utilizados en la construcción del Gateway LoRa, es el microcontrolador Arduino UNO, ya que este nos permite la conexión del módulo LoRa RFM95 con mayor facilidad, ya que incluye las bibliotecas necesarias para su funcionamiento, además la conexión y configuración es mucho más manejable. Mediante el IDE Arduino se programa el microcontrolador para que se puedan ejecutar las funcionalidades del módulo LoRa RFM95, en primer lugar, se inicia con la inclusión de las librerías necesarias:

Figura 53

Inclusión de librerías SPI y LoRa

```
4 //Inclusión de librerías
5 #include <SPI.h>
6 #include <LoRa.h>
```

Fuente. Autoría

- SPI.h: esta librería corresponde a la interfaz de comunicación para facilitar la conexión entre el microcontrolador y el módulo LoRa, hay que conectar cada uno de los pines del módulo al microcontrolador correctamente.
- LoRa.h: esta librería permite establecer la funcionalidad para el módulo LoRa RFM95, las configuraciones necesarias para el módulo se obtienen de la librería haciendo el llamado a cada una de las funciones que contiene esta.

A continuación, se definen los pines que son configurables por software mediante la librería LoRa.h:

Figura 54

Pines de conexión del Módulo LoRa RFM95 a Arduino UNO

```
8 //Definir los pines para la conexión del módulo RFM95
9 #define nss 10
10 #define rst 9
11 #define dio0 2
```

Fuente. Autoría

Para definir los parámetros de configuración de la comunicación LoRa para el módulo RFM95, se definen las variables con el valor de factor de dispersión, ancho de banda y tasa de codificación. Estos parámetros iniciales deben estar configurados de igual forma que los nodos, ya que así se establecerá la comunicación correctamente.

Figura 55

Variables para la configuración de parámetros de la comunicación LoRa

```

13 // Variables para configuraciones personalizadas del módulo RFM95 LoRa
14 const int sf = 7; // Spreading Factor (7-12)
15 const long bw = 125E3; // Bandwidth (Hz)
16 const int cr = 5; // Coding Rate

```

Fuente. Autoría

A continuación, se inicializa el módulo LoRa RFM95 indicando los pines de conexión que utiliza la librería LoRa.h, además, se establece la frecuencia para el funcionamiento del módulo, en este caso se utiliza la frecuencia 915MHz como se observa en la figura. Se inicializa los parámetros para la comunicación LoRa con un SF de 7, que permite menor alcance con mayor velocidad de transmisión, un BW de 125kHz y un CR de 4/5 que permite menor sobrecarga en la comunicación. Finalmente, se inicializa la opción de CRC, que permite la corrección de errores durante la comunicación.

Figura 56

Parámetros de la transmisión LoRa

```

27 //Definir los pines para el modulo LoRa RFM95
28 LoRa.setPins(nss, rst, dio0);
29
30 //Inicializa LoRa en la frecuencia definida
31 while (!LoRa.begin(915E6)) {
32   Serial.println("Falló. Revise las conexiones del módulo LoRa RFM95");
33   delay(500);
34 }
35
36 //Inicialización de parámetros LoRa para la Rx
37 LoRa.setSpreadingFactor(sf); //Factor de dispersión (7-12)
38 LoRa.setSignalBandwidth(bw); //Ancho de banda (125-250-500 kHz)
39 LoRa.setCodingRate4(cr); //Coding Rate
40 LoRa.enableCrc(); // Habilita la comprobación de redundancia cíclica (CRC)

```

Fuente. Autoría

Para la recepción de mensajes de los nodos sensores, se utiliza la función LoRa.parsePacket(), que permite leer el buffer de recepción utilizando la librería LoRa. Cada que un paquete llegue desde los nodos sensores configurados con los mismos parámetros de

comunicación LoRa, se lee el paquete y se obtiene la dirección o identificación de cada nodo además de los datos recolectados por el sensor.

Figura 57

Recepción de los datos de los nodos sensores LoRa

```

46 // Determina si hay un paquete para leer buffer de recepción de la librería LoRa.
47 int packetSize = LoRa.parsePacket();
48 if (packetSize) {
49     //Lee la dirección del nodo que envía el paquete.
50     uint8_t nodeAddress = LoRa.read();
51
52     //Crea la variable message para almacenar el mensaje de recepción LoRa
53     String message = "";
54
55     // Mientras haya paquetes disponibles para leer en el bufer se ejecuta las instrucciones
56     while (LoRa.available()) {
57         message += (char)LoRa.read();//concatena cada mensaje recibido a la cadena message
58     }

```

Fuente. Autoría

Finalmente, una vez leído los datos receptados e identificado el nodo sensor del cual provienen, se construye los datos que se transmiten por comunicación serial a la Raspberry PI 4, siendo estos la dirección del nodo, el nivel medido y el RSSI que corresponde a la intensidad de la señal con la que se comunica el nodo al Gateway.

Figura 58

Envío por comunicación Serial de los datos receptados por el Gateway LoRa

```

60 //verifica la dirección del nodo transmisor para imprimir el dato correspondiente a cada uno
61 if (nodeAddress == nodo1 || nodeAddress == nodo2){
62     //Se imprimen los datos necesarios para enviar por el puerto serial
63     Serial.print(String(nodeAddress));
64     Serial.print(", " );
65     Serial.print(message);
66     Serial.print(",");
67     Serial.println(LoRa.packetRssi()); //imprime el RSSI del paquete recibido
68 }

```

Fuente. Autoría

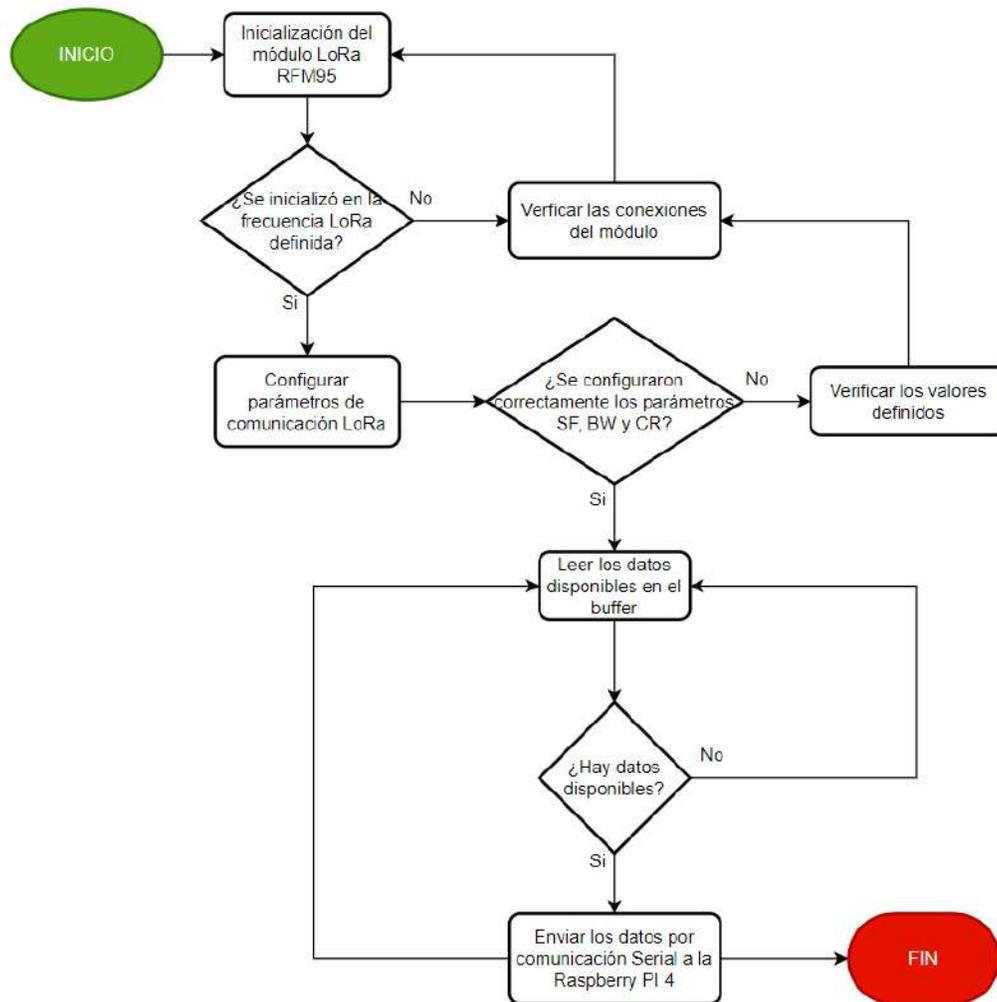
A continuación, se presenta el diagrama de flujo del código que se ejecuta para el Gateway LoRa, el proceso se inicia con la inicialización del módulo RFM95, en donde se

define la frecuencia para la inicialización, si esta se establece correctamente en los 915MHz, el módulo se inicializa y procede a configurar los parámetros de la comunicación LoRa, como lo es el SF, BW y CR, se verifica que estos valores estén correctamente definidos, caso contrario debe verificarse las conexiones y variables definidas para el módulo.

Establecido todo lo anterior, el Gateway LoRa es capaz de recibir los mensajes provenientes de los nodos sensores, procede a leer el buffer de la recepción LoRa, cada vez que exista un paquete disponible lo envía utilizando la comunicación serial para establecer conexión con la Raspberry Pi 4 y proceder al procesamiento y envío de los mismos al servidor IoT.

Figura 59

Diagrama de flujo del Gateway LoRa en Arduino UNO



Fuente. Autoría

3.8.3.2. Programación Raspberry PI 4

La Raspberry PI 4 forma parte del Gateway LoRa, este nos permite leer los paquetes receptados mediante la comunicación LoRa para posteriormente procesarlos y enviarlos a la plataforma IoT correspondiente. Para el procesamiento de los datos en Raspberry PI 4, utilizamos programación en Python utilizando el IDE de Thonny para la creación del script, para ello, se inicia con la importación de las librerías para poder utilizar las diferentes

funcionalidades que requiere el código. También, se inicializan las variables para almacenar el valor del nivel y el RSSI de la recolección de datos de los nodos sensores.

Figura 60

Importación de librerías en Python

```
1 #Importación de librerías
2 import serial
3 import paho.mqtt.publish as publish
4 import time
5 import smtplib
6 from email.mime.text import MIMEText
7
8 # Inicializar las variables de nivel y RSSI
9 nodo1 = 0
10 nodo2 = 0
11 rssi1 = 0
12 rssi2 = 0
```

Fuente. Autoría

- Serial: esta librería permite leer los datos disponibles en el puerto serial de la Raspberry Pi.
- MQTT: esta librería permite la utilización del protocolo MQTT para publicar en la plataforma IoT ThingSpeak.
- Time: esta librería permite trabajar con valores de tiempo para ejecución del programa.
- SMTPLIB: esta librería permite utilizar las funciones para enviar mensajes mediante correo electrónico utilizando el protocolo smtp.

En la siguiente figura, se muestra los datos que se deben incluir dentro del código para generar la alerta mediante correo electrónico, aquí se visualiza que se tiene el correo del remitente, el destinatario y la contraseña de acceso.

Figura 61

Datos de correo electrónico para envío de alerta

```

14 #Datos para alerta gmail
15 remitente = 'ferpaul14@gmail.com'
16 contrasena = 'myiezczkkopqvsuh'
17 destinatario = 'ramirez.ortega.paul@gmail.com'
18

```

Fuente. Autoría

Para el acceso a ThingSpeak, se declara las variables en donde se visualizan los datos de nuestra plataforma creada, estos datos nos permiten el acceso a nuestro Canal en donde se visualizarán los datos obtenidos.

Para el envío de datos se utiliza el protocolo MQTT, se utiliza el broker del servidor de ThingSpeak para realizar la publicación de los datos, se utiliza el protocolo TCP y se accede mediante el puerto 1883.

Figura 62

Datos para acceso a la plataforma IoT ThingSpeak

```

12 # Datos para acceder a ThingSpeak
13 write_api_key = "2B63MJC2V4RDAB8D" #clave para acceso a la API
14 channel_ID = "2050074" #Identificador del canal de thingspeak
15 mqtt_host = "mqtt3.thingspeak.com" # Servidor MQTT de ThingSpeak
16 t_transport = "tcp" #protocolo de transporte
17 t_port = 1883 # Puerto MQTT de ThingSpeak
18 topic = "channels/" + channel_ID + "/publish" #topic para publicar en ThingSpeak
19
20
21 # Credenciales del dispositivo creado para acceder a MQTT de thingspeak
22 mqtt_client_ID = "HwE2Ex8mPQUqEygCNikRLgk" #id del cliente mqtt
23 mqtt_username = "HwE2Ex8mPQUqEygCNikRLgk" #usuario del cliente mqtt
24 mqtt_password = "HJVRJ7Njvyf6sJT8GQGYF43Z" #clave de acceso para el cliente mqtt

```

Fuente. Autoría

Mediante la lectura del puerto serie, estableciendo los baudios para la velocidad de comunicación entre el Arduino UNO y la Raspberry, se obtienen los datos captados de los sensores. Cada vez que se reciba un dato, se realiza el proceso de separación de la cadena de texto, con ello se comprueba a que nodo corresponde la información y se separa dichos

valores en las variables correspondientes para poder construir el payload para enviar mediante el protocolo MQTT a ThingSpeak.

Figura 63

Lectura y procesamiento de los datos

```

26 #inicializacion del puerto serial, se define el puerto USB y el baudrate
27 ser = serial.Serial('/dev/ttyUSB0', 115200)
28
29 while True:
30     if ser.in_waiting > 0: #verifica que haya datos en el puerto serie
31         data = ser.readline().decode('utf-8') #leer los datos y decodificarlos
32         #print(data)
33         datos=data.split(",") #separa la cadena de datos utilizando el identificador ","
34         # Comprobar si los datos contienen la cadena 1 en la posición 0 para identificar al Nodo 1"
35         if "1" in datos[0]:
36             # Si los datos contienen la cadena "1", extrae el valor numérico de la posición correspondiente
37             nodo1 = datos[1]
38             rssi1 = datos[2]
39             print("Nodo 1: " + nodo1 + rssi1) #imprime el valor recibido del nodo 1
40
41         # Comprobar si los datos contienen la cadena 2 en la posición 0 para identificar al Nodo 1"
42         if "2" in datos[0]:
43             # Si los datos contienen la cadena "2", extrae el valor numérico de la posición correspondiente
44             nodo2 = datos[1]
45             rssi2 = datos[2]
46             print("Nodo 2: " + nodo2 + rssi2) #imprime el valor recibido del nodo 2

```

Fuente. Autoría

Para el envío de la alerta mediante correo, se evalúa el valor obtenido de cada sensor, se establece una escala para el nivel, en la cual el valor promedio del nivel del rio en base a las mediciones experimentales realizadas se mantiene oscilando entre los 50cm hasta los 65cm, por lo tanto, se crea alertas tanto para el descenso como para el aumento del nivel. La alerta de descenso del nivel se envía cuando el valor medido está por debajo de los 40cm, para el caso del incremento del nivel, la alerta se envía cuando el valor medido está sobre los 70cm, en la siguiente figura se observa la programación realizada en python para cumplir con dicha función.

Figura 64

Código para el envío de alertas mediante correo

```

47 #Envío de alerta mediante correo
48 if float(nodol)<40 and float(nodol)>0:
49     try:
50         asunto = 'Sistema de Monitoreo del nivel de agua de Rio Chico'
51         mensaje = f'Se detectó un descenso en el nivel.\nEl último valor de nivel medido en el NODO 1 es de: {nodol} cm.'
52         contenido = MIMEText(mensaje)
53
54         # Configurar los detalles del correo electrónico
55         contenido['From'] = remitente
56         contenido['To'] = destinatario
57         contenido['Subject'] = asunto
58
59         # Conectarse al servidor SMTP y enviar el correo electrónico
60         servidor_smtp = smtplib.SMTP('smtp.gmail.com', 587)
61         servidor_smtp.starttls()
62         servidor_smtp.login(remitente, contraseña)
63         servidor_smtp.sendmail(remitente, destinatario, contenido.as_string())
64         servidor_smtp.quit()
65     except:
66         continue
67
68 if float(nodol)>70:
69     try:
70         asunto = 'Sistema de Monitoreo del nivel de agua de Rio Chico'
71         mensaje = f'Se detectó un aumento en el nivel.\nEl último valor de nivel medido en el NODO 1 es de: {nodol} cm.'
72         contenido = MIMEText(mensaje)
73
74         # Configurar los detalles del correo electrónico
75         contenido['From'] = remitente
76         contenido['To'] = destinatario
77         contenido['Subject'] = asunto
78

```

Fuente. Autoría

Se verifica que se obtenga datos para cada uno de los sensores y se procede a construir el payload para enviar a ThingSpeak, en este se deben definir las variables con los datos correspondientes al valor de nivel a la intensidad de RSSI que registra cada nodo al momento del envío del paquete, posteriormente definidos estos campos se procede a utilizar el método publish con las variables correspondientes para el correcto envío de datos a la plataforma.

Figura 65

Publicar los datos en ThingSpeak utilizando el protocolo MQTT

```

61     # Publicar los datos en el canal de ThingSpeak si se han obtenido valores de nodo 1 y nodo 2
62     if nodo1 != 0 and nodo2 != 0:
63         payload = "field1="+str(nodo1)+"&field2="+str(nodo2)+"&field3="+str(rssi1)+"&field4="+str(rssi2)
64         print(payload)
65         publish.single(topic, payload, hostname=mqtt_host, transport=t_transport, port=t_port, client_id
66         nodo1 = 0
67         nodo2 = 0
68         rssi1 = 0
69         rssi2 = 0

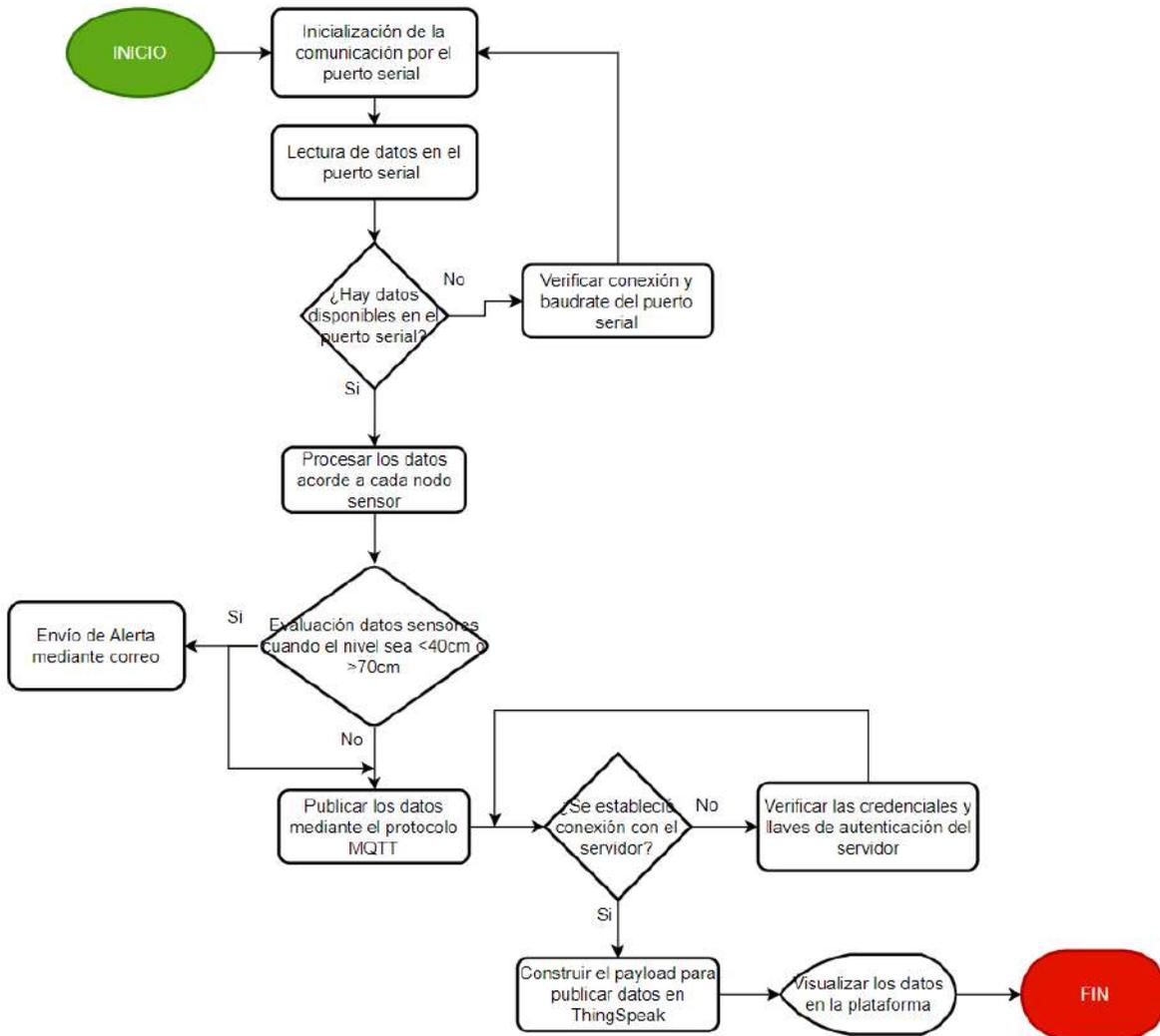
```

Fuente. Autoría

En el siguiente diagrama de flujo se representa el proceso de funcionamiento del código realizado para el procesamiento y envío de datos desde el Gateway LoRa, el proceso inicia con la comunicación mediante el puerto serial donde se conecta el Arduino UNO con la Raspberry PI 4, posteriormente se realiza la lectura de los datos en este puerto, si no se realiza correctamente es necesario verificar las configuraciones y conexiones entre los dispositivos. Si todo lo anterior funciona de manera correcta, se realiza el procesamiento de los datos para poder obtener los valores indicados de cada sensor y ejecutar la evaluación del nivel en el rango establecido para verificar si este aumentó o disminuyó con relación a esto, de esta forma se puede enviar una alerta mediante correo electrónico indicando la variación, luego se crea el payload para enviar mediante el protocolo MQTT al servidor de ThingSpeak, si los datos no se envían correctamente es necesario verificar las credenciales de acceso al servidor y las claves de autenticación del dispositivo MQTT. Si la conexión se establece correctamente con el servidor MQTT, se visualizan los datos con las herramientas de visualización de ThingSpeak.

Figura 66

Diagrama de flujo del envío de datos al servidor ThingSpeak



Fuente. Autoría

3.8.4. Configuración ThingSpeak

Para la visualización de los datos, se utiliza la plataforma ThingSpeak, esta plataforma posee un servidor MQTT mediante el cual se permite la publicación de los mensajes en el topic correspondiente a la identificación del canal creado en dicha plataforma.

Nos dirigimos a la página oficial de ThingSpeak y accedemos mediante una cuenta personal creada con anterioridad. En la parte Channels, se selecciona crear un nuevo canal y se debe definir el nombre del canal, una descripción de este y los campos para las variables que corresponden a cada uno de los datos enviados mediante el protocolo MQTT, estos campos corresponden a la medición de nivel e intensidad de la señal para ambos nodos sensores.

Figura 67

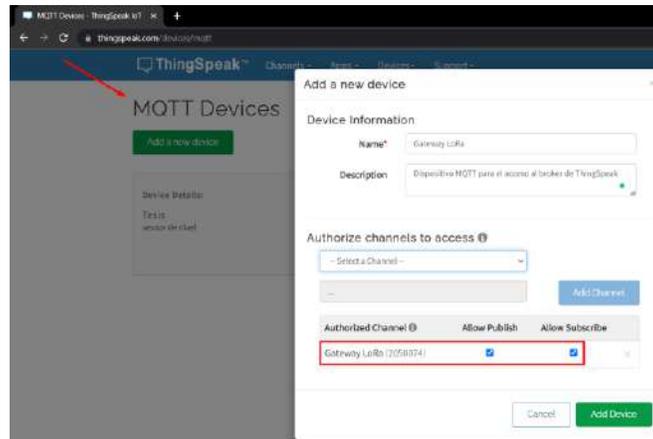
Creación de un nuevo canal en ThingSpeak

The screenshot shows the 'New Channel' page on ThingSpeak. The browser address bar shows 'thingspeak.com/channels/new'. The page title is 'New Channel'. The form includes:

- Name:** Gateway LoRa
- Description:** Este canal permite la visualización de los datos recolectados por los nodos sensores utilizando MQTT.
- Field 1:** Nivel Nodo 1 (checked)
- Field 2:** Nivel Nodo 2 (checked)
- Field 3:** RSSI Nodo 1 (checked)
- Field 4:** RSSI Nodo 2 (checked)

Fuente. Autoría

A continuación, se procede a crear el dispositivo MQTT que nos permitirá el acceso al broker MQTT de ThingSpeak para la suscripción y publicación de los datos en el topic correspondiente. Para ello, nos dirigimos a la parte de Devices y luego a Add new device, se abrirá una ventana para añadir un nuevo dispositivo, se debe ingresar el nombre del dispositivo, una descripción y seleccionar el canal creado anteriormente, es importante verificar que las funciones de publish y subscribe estén habilitadas, finalizado esto se agrega el dispositivo mencionado.

Figura 68*Creación del dispositivo MQTT*

Fuente. Autoría

Finalizadas las configuraciones, se pueden observar los dos elementos creados anteriormente en cada una de sus pestañas, se verifica que está correctamente implementados en la plataforma.

Figura 69

Canal y dispositivo MQTT creado en ThingSpeak

The image shows two screenshots from the ThingSpeak web interface. The top screenshot displays the 'My Channels' page. It features a navigation bar with 'Channels', 'Apps', 'Devices', and 'Support'. Below the navigation bar, there is a 'New Channel' button and a search box labeled 'Search by tag'. A table lists the channel 'Gateway LoRa', created on 2023-03-01 and updated on 2023-03-07 11:21. Below the table are buttons for 'Private', 'Public', 'Settings', 'Sharing', 'API Keys', and 'Data Import / Export'. The bottom screenshot shows the 'MQTT Devices' page. It has an 'Add a new device' button and a table with columns for 'Device Details', 'Authorized Channels and Permissions', and 'MQTT Client ID'. The table contains one entry for 'Gateway LoRa' with a device ID of '2050074', permissions for 'publish' and 'subscribe', and an MQTT Client ID of 'HwE26-smPQvEygGVmRlgk'. There are 'Edit' and 'Delete' buttons for this device.

Fuente. Autoría

Como se pudo observar en la fase de análisis de requerimientos y diseño del sistema, se cumple con el objetivo de plantear el desarrollo del sistema utilizando la tecnología LoRa y la plataforma de visualización para la presentación de la información obtenida del nivel de agua del caudal de Río Chico, de acuerdo al diagrama de bloques y la arquitectura del sistema presentado a lo largo del Capítulo III, el sistema cumple con los requerimientos de hardware y software seleccionados para ejecutar mediciones de nivel, transmisión y visualización de la información además del envío de la alerta.

4. Capítulo IV: Implementación y Pruebas de Funcionamiento

En este capítulo, acorde a la metodología en cascada, específicamente la fase tres y cuatro, se expondrá la integración de los diferentes componentes en base al hardware seleccionado en el capítulo anterior, también se evidenciará el funcionamiento de acuerdo con las configuraciones de software realizadas para la comunicación de la red y para la visualización de los datos. Finalmente, luego de la implementación del sistema, se realizará las pruebas de funcionamiento correspondientes y se evidenciarán los resultados obtenidos.

4.1. Integración de Componentes

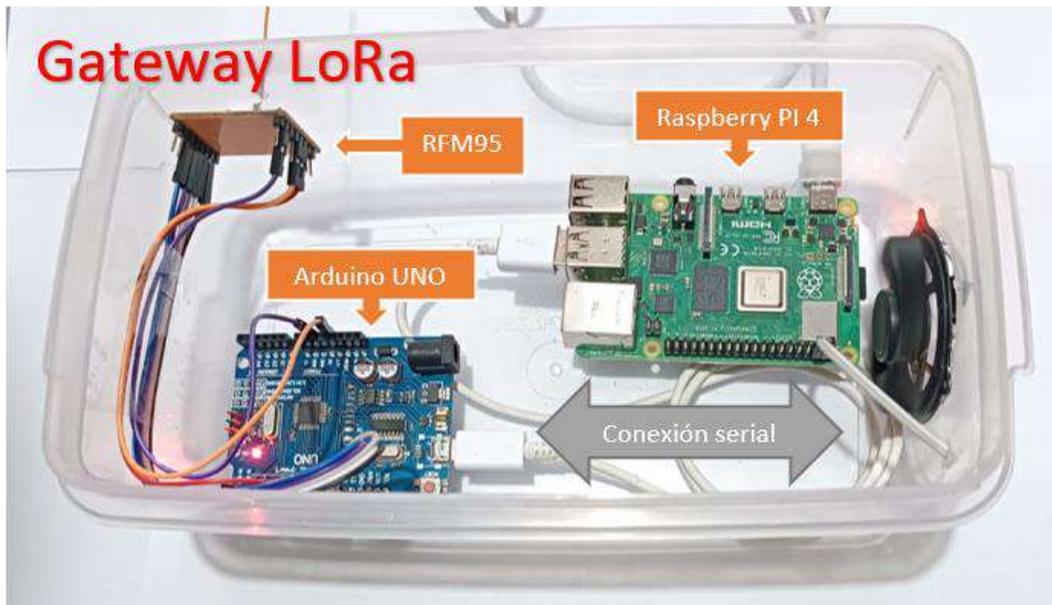
Seleccionado el Hardware correspondiente para cada uno de los elementos de la red, se procede a realizar las conexiones físicas de los equipos en base a lo planteado en las configuraciones establecidas en el capítulo anterior.

Para el nodo sensor, se realiza la conexión de los dispositivos periféricos a la placa ESP32, en la siguiente figura se visualiza como queda la implementación del nodo sensor, el cual consta del sensor ultrasónico HC-SR04 conectado a la placa ESP32, el módulo LoRa RFM95 conectado al ESP32 y alimentado por una batería de litio con entrada de 5V para el funcionamiento autónomo.

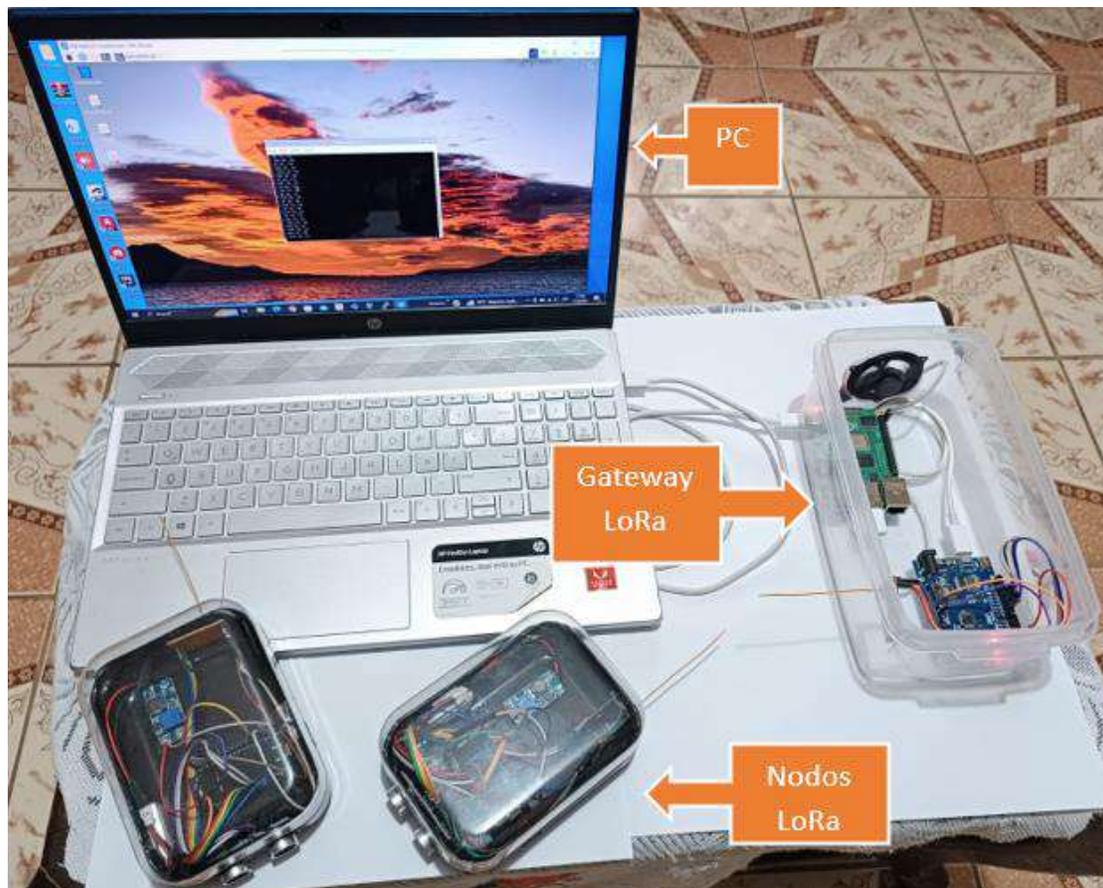
El nodo sensor armado dentro de su case queda de la siguiente manera:

Figura 70*Nodo LoRa**Fuente. Autoría*

Para el Gateway LoRa, se utilizó la comunicación serial entre el Arduino UNO y la Raspberry PI para obtener los datos recolectados mediante el módulo RFM95 al Arduino UNO, los datos se procesan en la Raspberry PI 4 y se envían a la plataforma de ThingSpeak. El nodo central o Gateway LoRa puede verse en su case, ensamblado acorde a lo planteado en la siguiente figura:

Figura 71*Gateway LoRa**Fuente. Autoría*

Finalizada la implementación de cada uno de los nodos se procede a la configuración y a la verificación de la comunicación entre los diferentes nodos, cabe recalcar que se cuenta con dos nodos sensores y un Gateway LoRa.

Figura 72*Componentes del sistema**Fuente. Autoría*

4.2. Verificación de la Configuración de los Elementos de la Red

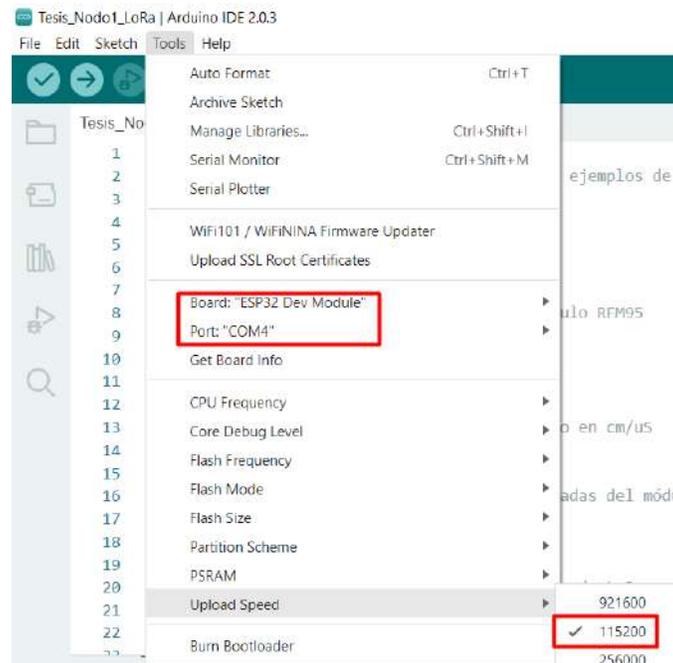
A continuación, se procede a verificar la configuración realizada en cada uno de los elementos del subsistema, se verifica que todo funciona correctamente y que la configuración de los módulos LoRa se establece correctamente en cada nodo para la transmisión y recepción, además de que cada uno de estos cumpla con la función definida.

4.2.1. Verificación del Nodo Sensor

Para establecer la configuración en el nodo sensor, se procede a la carga del programa desde Arduino IDE a la placa ESP32 en la cual se encuentra conectado el módulo LoRa RFM95 y el sensor ultrasónico. Se realiza la configuración a través de la PC utilizando el puerto COM designado, se selecciona el tipo de placa, el puerto y la velocidad de subida.

Figura 73

Selección de la placa en Arduino IDE



Fuente. Autoría

Mediante el monitor serial podemos observar la inicialización del nodo 1, este se establece en los parámetros definidos para la comunicación mediante la tecnología LoRa. Como se puede observar, la frecuencia de trabajo es 915MHz, también se define el SF en un valor de 7, un BW de 125 kHz y un CR de 4/5.

Figura 74

Inicialización del módulo RFM95 en nodo 1

```

Tesis_Nodo1_LoRa.ino
47 //Inicializa LoRa en la frecuencia definida
48 while (!LoRa.begin(915E6)) {
49   Serial.println("Falló. Revise las conexiones de
50   delay(500);
51 }
52
53 //Inicialización de parámetros LoRa para la TX
54 LoRa.setSpreadingFactor(sf); //Factor de dispersi
55 LoRa.setSignalBandwidth(bw); //Ancho de banda (12
56 LoRa.setCodingRate4(cr); //Coding Rate
57 LoRa.enableCrc(); // Habilita la comprobación de
58 LoRa.receive(); // Habilita el modo de recepción
59 LoRa.setSyncWord(0xF3); //sincroniza con los elem
60
61 //Impresión de los parámetros configurados en el
62 Serial.println("Modulo LoRa inicializado. OK!");
63 Serial.print("SF: ");
64 Serial.println(sf);
65 Serial.print("BW: ");
66 Serial.println(bw);
67 Serial.print("CR: 4/");
68 Serial.println(cr);

```

```

Serial Monitor x
Message (Enter to send message to 'ESP32 Dev Module' or
08:36:05.142 -> Nodo LoRa 1
08:36:05.183 -> Modulo LoRa inicializado. OK!
08:36:05.183 -> SF: 7
08:36:05.183 -> BW: 125000
08:36:05.183 -> CR: 4/5
08:36:05.183 -> Nodo 1. Paquete: 0, Nivel: 17
08:36:15.227 -> Nodo 1. Paquete: 1, Nivel: 17
08:36:25.282 -> Nodo 1. Paquete: 2, Nivel: 16

```

Fuente. Autoría

Mediante el monitor serial de Arduino IDE visualizamos la inicialización del módulo RFM95 en el nodo 2, se establece en los parámetros de configuración definidos para LoRa como se observa en la figura: un Sf de 7, un BW de 125 kHz y un CR de 4/5, además se encuentra en la frecuencia de trabajo de los 915 MHz.

Figura 75

Inicialización del módulo RFM95 en nodo 2

The image shows two side-by-side screenshots of the Arduino IDE 2.0.3 interface. The left screenshot displays the sketch 'Tesis_Nodo2_LoRa.ino' with the following code:

```

47 //Inicializa LoRa en la frecuencia definida
48 while (!LoRa.begin(915E6)) {
49   Serial.println("Falló. Revise las conexiones del módulo");
50   delay(500);
51 }
52
53 //Inicialización de parámetros LoRa para la Tx
54 LoRa.setSpreadingFactor(sf); //Factor de dispersión
55 LoRa.setSignalBandwidth(bw); //Ancho de banda (125-
56 LoRa.setCodingRate4(cr); //Coding Rate
57 LoRa.enableCrc(); // Habilita la comprobación de rec
58 LoRa.receive(); // Habilita el modo de recepción
59 LoRa.setSyncWord(0xF3); //sincroniza con los elemnte
60
61 //Impresión de los parámetros configurados en el mód
62 Serial.println("Módulo LoRa inicializado. OK!");
63 Serial.print("SF: ");
64 Serial.println(sf);
65 Serial.print("BW: ");
66 Serial.println(bw);
67 Serial.print("CR: 4/");
68 Serial.println(cr);

```

The right screenshot shows the Serial Monitor output for the same sketch, displaying the following messages:

```

08:30:09.872 -> ets Jun  8 2016 00:22:57
08:30:09.872 ->
08:30:09.872 -> rst:0x1 (POWERON_RESET),boot:
08:30:09.872 -> configsip: 0, SPIWP:0xee
08:30:09.872 -> clk_drv:0x00,q_drv:0x00,d_drv
08:30:09.918 -> mode:DIO, clock div:1
08:30:09.918 -> load:0x3fff0018,len:4
08:30:09.918 -> load:0x3fff001c,len:1216
08:30:09.918 -> ho 0 tail 12 room 4
08:30:09.918 -> load:0x40078000,len:10944
08:30:09.918 -> load:0x40080400,len:6388
08:30:09.918 -> entry 0x400806b4
08:30:10.043 -> Nodo LoRa 2
08:30:10.043 -> Modulo LoRa inicializado. OK!
08:30:10.043 -> SF: 7
08:30:10.043 -> BW: 125000
08:30:10.043 -> CR: 4/5
08:30:10.043 -> Nodo 2. Paquete: 0, Nivel: 27

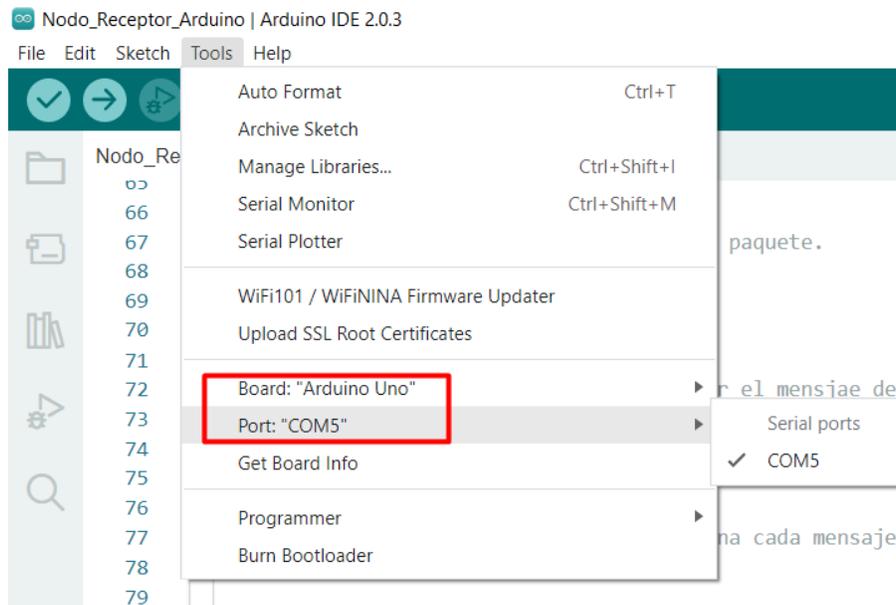
```

Red boxes highlight the initialization code in the sketch and the corresponding serial output messages. A red arrow points from the code to the output.

Fuente. Autoría

4.2.2. Verificación del Gateway LoRa

Para verificar la inicialización del módulo LoRa en el Gateway, se procede a seleccionar la placa y el puerto COM para la placa Arduino, la cual es parte del Gateway LoRa. Esto se realiza utilizando un PC y accediendo a la placa mediante el puerto definido.

Figura 76*Selección de la placa Arduino*

Fuente. Autoría

Mediante el monitor serial, podemos observar que se inicializa el módulo LoRa en el Gateway, de esta forma se observa la configuración de la comunicación LoRa definiendo los parámetros de SF en 7, BW de 125 kHz y un CR de 4/5, además se tiene la frecuencia de trabajo de 915MHz.

Figura 77*Inicialización del módulo LoRa en el Gateway*

```

33 //Inicializa LoRa en la frecuencia definida
34 while (!LoRa.begin(915E6)) {
35   Serial.println("Falló. Revise las conexiones del módulo LoRa");
36   delay(500);
37 }
38
39 //Inicialización de parámetros LoRa para la Rx
40 LoRa.setSpreadingFactor(sf); //Factor de dispersión (7-12)
41 LoRa.setSignalBandwidth(bw); //Ancho de banda (125-250-500 kHz)
42 LoRa.setCodingRate4(cr); //Coding Rate
43 LoRa.enableCrc(); // Habilita la comprobación de redundancia cíclica
44 LoRa.receive(); // Habilita el modo de recepción continua (clasificación)
45 LoRa.setSyncWord(0xF3); // Sincroniza con los elementos de la red
46
47 //Impresión de los parámetros configurados en el módulo
48 Serial.println("Módulo LoRa inicializado. OK!");
49 Serial.print("SF: ");
50 Serial.println(sf);
51 Serial.print("BW: ");
52 Serial.println(bw);
53 Serial.print("CR: 4/");
54 Serial.println(cr);

```

Output Serial Monitor X

```

Message (Enter to send message to 'Arduino Uno' on 'COM5')
08:44:11.761 -> Gateway LoRa
08:44:11.761 -> Módulo LoRa inicializado. OK!
08:44:11.761 -> SF: 7
08:44:11.761 -> BW: 125000
08:44:11.761 -> CR: 4/5

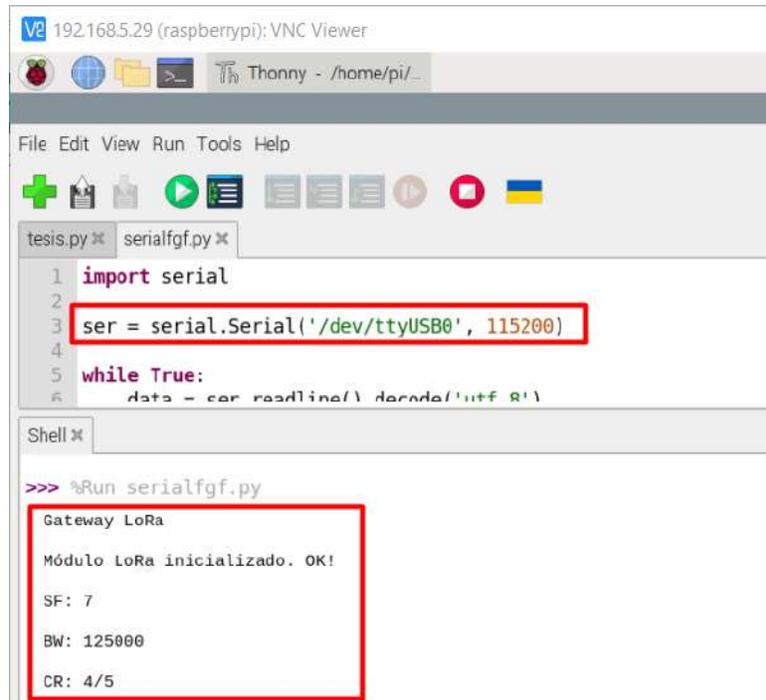
```

Fuente. Autoría

Mediante la lectura del puerto serie en la Raspberry Pi, podemos verificar la inicialización del módulo conectado al Arduino UNO, aquí podemos observar los mismos parámetros de comunicación anteriormente establecidos, como son visibles desde la Raspberry, se tiene comunicación entre estos dos elementos que conforman el Gateway LoRa.

Figura 78

Inicialización del módulo LoRa en el Gateway



```

192.168.5.29 (raspberrypi): VNC Viewer
Thonny - /home/pi/...
File Edit View Run Tools Help
+ [Icons]
tesis.py serialfgf.py
1 import serial
2
3 ser = serial.Serial('/dev/ttyUSB0', 115200)
4
5 while True:
6     data = ser.readline().decode('utf 8')
Shell
>>> %Run serialfgf.py
Gateway LoRa
Módulo LoRa inicializado. OK!
SF: 7
BW: 125000
CR: 4/5

```

Fuente. Autoría

4.3. Pruebas de funcionamiento del sistema

Para las pruebas de funcionamiento del sistema, se realiza la recolección de datos de cada uno de los nodos sensores, de esta forma se verifica que el módulo LoRa RFM95 se inicialice correctamente y que el sensor proporcione el valor de nivel en centímetros. También, se evidencia que los datos se reciben de forma correcta en el Gateway LoRa para su posterior procesamiento y envío a ThingSpeak.

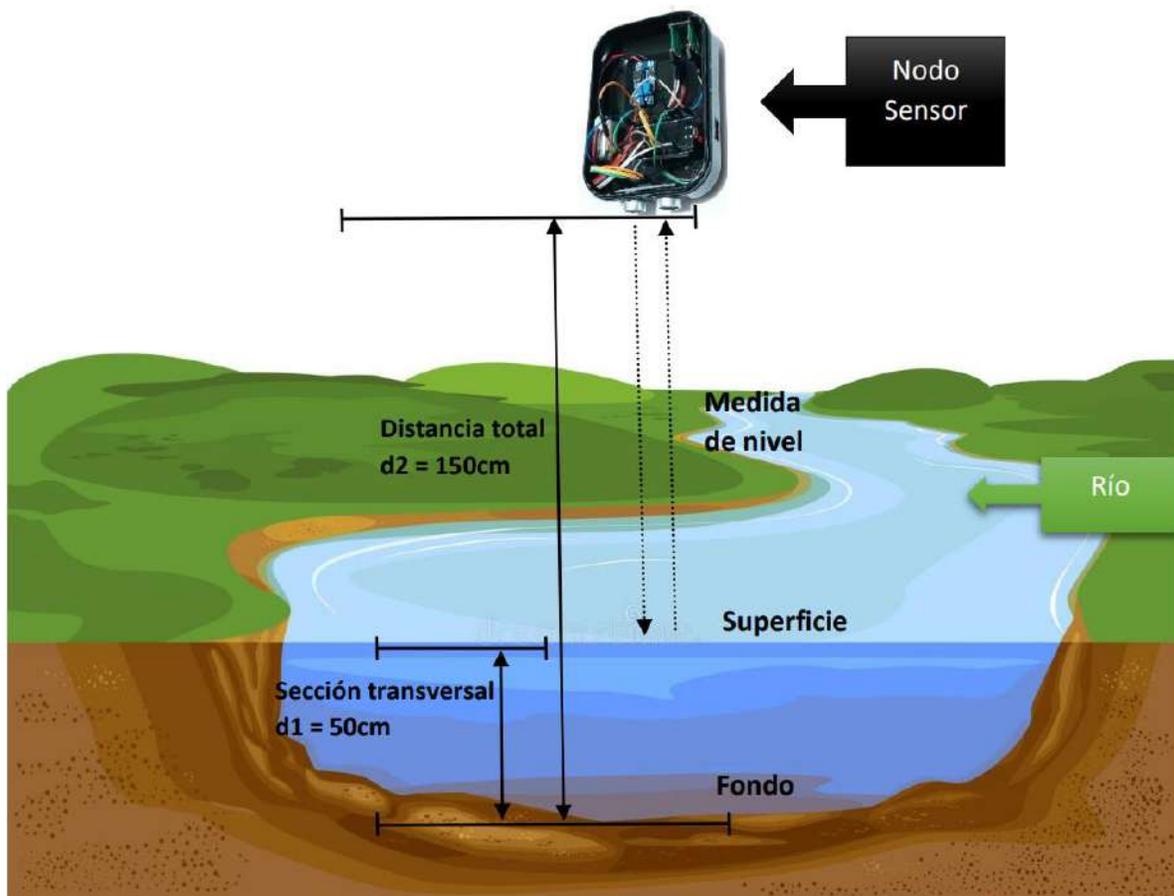
4.3.1. Implementación del Sistema

Como ya se había analizado en la fase de requerimientos y diseño del sistema, el proyecto se desarrolla en Río Chico, sitio en el cual se va a colocar dos nodos sensores para medir en dos puntos diferentes el valor del nivel de agua del río. Para colocar los nodos

sensores se tiene en cuenta que, para obtener el valor del nivel de agua se necesita conocer la medida de la sección transversal del río, es decir, la medida desde el fondo hasta la superficie, además se necesita conocer la medida total, que corresponde a la medida desde el fondo del río hasta el nodo, realizando la resta de estas dos medidas obtenidas el sensor puede calcular el nivel del río en centímetros, en la siguiente figura puede visualizarse lo mencionado.

Figura 79

Ubicación y medidas del nodo sensor sobre el río



Fuente. Autoría

Ahora, se procede a colocar el sensor de forma física en la ubicación seleccionada, tomando en cuenta las medidas vistas en la figura anterior para el correcto funcionamiento del sensor. En la siguiente figura, se observa la colocación del nodo sensor sobre el río, se

utiliza una estructura de tubería de PVC como soporte para el sensor, de esta forma este puede obtener la medida de nivel requerida.

Figura 80

Ubicación del nodo sensor sobre el río



Fuente. Autoría

4.3.2. Recolección y Visualización de los Datos del Nodo Sensor

Mediante el monitor serial podemos observar la inicialización del nodo LoRa con los parámetros de comunicación establecidos, se evidencia la toma de datos mediante el sensor y el envío de paquete LoRa con el valor capturado, en la siguiente figura, se observa el paquete correspondiente al nodo 1:

Figura 81

Recolección y visualización de los datos del nodo 1

The image shows two windows from the Arduino IDE. The left window displays the source code for 'Tesis_Nodo1_LoRa.ino'. The right window shows the 'Serial Monitor' output.

Code Snippets (Left Window):

```

78 delayMicroseconds(2);
79 digitalWrite(trigPin, HIGH);
80 delayMicroseconds(10);
81 digitalWrite(trigPin, LOW);
82
83 // Lee el pin echo, se obtiene el tiempo que viajó la onda
84 duracion = pulseIn(echoPin, HIGH);
85
86 // Se calcula la distancia y se obtiene el promedio para
87 for(int i = 0; i < muestras; i++){
88   nivelCm = duracion * velsonido/2;
89   promedio = promedio + nivelCm;
90 }
91 //Se obtiene el nivel restando la medida obtenida menos
92 nivelCm = distanciaMax-(promedio/muestras);
93
94 //Impresión de los valores del nodo 1
95 Serial.print("Nodo 1. Paquete: ");
96 Serial.print(counter);
97 Serial.print(", Nivel: ");
98 nivelCm = distanciaMax-(promedio/muestras); //se obtiene
99 Serial.println(nivelCm);
100
101 //Paquete LoRa con los datos a enviar
102 LoRa.beginPacket(); //inicializa el paquete
103 LoRa.write(nodeAddress); //se envia la dirección del nodo
104 LoRa.print(nivelCm); //se envia el valor medido
105 LoRa.endPacket(); //finaliza el paquete

```

Serial Monitor Output (Right Window):

```

61 //Impresión de los parámetros configurados
Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')
08:36:05.142 -> Nodo LoRa 1
08:36:05.163 -> Modulo LoRa inicializado. OK!
08:36:05.163 -> SF: 7
08:36:05.163 -> BW: 125000
08:36:05.163 -> CR: 4/5
08:36:05.163 -> Nodo 1. Paquete: 0, Nivel: 172.35
08:36:15.227 -> Nodo 1. Paquete: 1, Nivel: 172.31
08:36:25.282 -> Nodo 1. Paquete: 2, Nivel: 169.80
08:36:35.309 -> Nodo 1. Paquete: 3, Nivel: 169.80
08:36:45.360 -> Nodo 1. Paquete: 4, Nivel: 172.35
08:36:55.409 -> Nodo 1. Paquete: 5, Nivel: 170.24
08:37:05.455 -> Nodo 1. Paquete: 6, Nivel: 168.96
08:37:15.511 -> Nodo 1. Paquete: 7, Nivel: 170.22
08:37:25.549 -> Nodo 1. Paquete: 8, Nivel: 169.37
08:37:35.585 -> Nodo 1. Paquete: 9, Nivel: 172.31
08:37:45.643 -> Nodo 1. Paquete: 10, Nivel: 168.95
08:50:53.152 -> Nodo 1. Paquete: 1, Nivel: 168.59
08:51:03.188 -> Nodo 1. Paquete: 2, Nivel: 169.44
08:51:13.234 -> Nodo 1. Paquete: 3, Nivel: 172.77
08:51:23.278 -> Nodo 1. Paquete: 4, Nivel: 172.38
08:51:33.316 -> Nodo 1. Paquete: 5, Nivel: 168.20
08:51:43.381 -> Nodo 1. Paquete: 6, Nivel: 210.83
08:51:53.385 -> Nodo 1. Paquete: 7, Nivel: 212.09
08:52:03.473 -> Nodo 1. Paquete: 8, Nivel: 210.00

```

Fuente. Autoría

En el Gateway LoRa se reciben los datos correspondientes al nodo 1 como se observa en la siguiente figura, se tiene el identificador del nodo, el valor del nivel medido en centímetros y el RSSI de la intensidad de la señal recibida medido en dBm.

Figura 82

Recepción de datos del nodo 1 en el Gateway LoRa

```
>>> %Run serialfgf.py
Gateway LoRa
Módulo LoRa inicializado. OK!
SF: 7
BW: 125000
CR: 4/5
Paquete recibido: Nodo: 1, Nivel: 56.82, con RSSI: -71
Paquete recibido: Nodo: 1, Nivel: 54.73, con RSSI: -69
Paquete recibido: Nodo: 1, Nivel: 55.15, con RSSI: -73
Paquete recibido: Nodo: 1, Nivel: 55.56, con RSSI: -66
Paquete recibido: Nodo: 1, Nivel: 56.39, con RSSI: -77
Paquete recibido: Nodo: 1, Nivel: 56.39, con RSSI: -68
Paquete recibido: Nodo: 1, Nivel: 56.38, con RSSI: -62
Paquete recibido: Nodo: 1, Nivel: 56.41, con RSSI: -81
Paquete recibido: Nodo: 1, Nivel: 56.82, con RSSI: -87
Paquete recibido: Nodo: 1, Nivel: 53.06, con RSSI: -70
```

Fuente. Autoría

En la siguiente figura, se visualiza la inicialización del nodo 2 y se evidencia la toma de datos, este dato se envía mediante el paquete LoRa como se muestra a continuación:

Figura 83

Recolección y visualización de los datos del nodo 2

The figure consists of two side-by-side screenshots of the Arduino IDE 2.0.3 interface. The left screenshot shows the code for 'Tesis_Nodo2_LoRa.ino'. The code includes comments in Spanish and functions for digital pin control, pulseIn, and LoRa packet handling. Two sections of code are highlighted with red boxes: a loop for calculating distance and a section for printing node data and sending a LoRa packet. The right screenshot shows the 'Serial Monitor' window, which displays the output of the code. The output shows the initialization of the LoRa module and a series of received data packets for 'Nodo 2', each containing a packet number and a level value.

```

// cambia el estado del pin trigger
76 digitalWrite(trigPin, LOW);
77 delayMicroseconds(2);
78 digitalWrite(trigPin, HIGH);
79 delayMicroseconds(10);
80 digitalWrite(trigPin, LOW);
81
82
83 // Lee el pin echo, se obtiene el tiempo que viajó
84 duracion = pulseIn(echoPin, HIGH);
85
86 // Se calcula la distancia y se obtiene el promedio
87 for(int i = 0; i < muestras; i++){
88   nivelCm = duracion * velSonido/2;
89   promedio = promedio + nivelCm;
90 }
91
92 //Impresión de los valores del nodo 2
93 Serial.print("Nodo 2. Paquete: ");
94 Serial.print(counter);
95 Serial.print(", Nivel: ");
96 nivelCm = distanciaMax-(promedio/muestras); //se obti
97 Serial.println(nivelCm);
98
99 //Paquete LoRa con los datos a enviar
100 LoRa.beginPacket(); //inicializa el paquete
101 LoRa.write(nodeAddress); //se envia la dirección del nodo
102 LoRa.print(nivelCm); //se envia el valor medido
103 LoRa.endPacket(); //finaliza el paquete

```

```

08:59:05.748 -> Nodo LoRa 2
08:59:05.769 -> Modulo LoRa inicializado. OK!
08:59:05.769 -> SF: 7
08:59:05.789 -> BW: 125000
08:59:05.789 -> CR: 4/5
08:59:05.789 -> Nodo 2. Paquete: 0, Nivel: 272.04
08:59:15.832 -> Nodo 2. Paquete: 1, Nivel: 272.95
08:59:25.828 -> Nodo 2. Paquete: 2, Nivel: 272.53
08:59:35.906 -> Nodo 2. Paquete: 3, Nivel: 272.53
08:59:45.937 -> Nodo 2. Paquete: 4, Nivel: 273.04
08:59:55.961 -> Nodo 2. Paquete: 5, Nivel: 272.61
09:00:06.022 -> Nodo 2. Paquete: 6, Nivel: 272.61
09:00:16.057 -> Nodo 2. Paquete: 7, Nivel: 273.02
09:00:26.095 -> Nodo 2. Paquete: 8, Nivel: 272.61
09:00:36.125 -> Nodo 2. Paquete: 9, Nivel: 273.02
09:00:46.128 -> Nodo 2. Paquete: 10, Nivel: 273.02
09:00:56.202 -> Nodo 2. Paquete: 11, Nivel: 273.02
09:01:06.247 -> Nodo 2. Paquete: 12, Nivel: 273.02

```

Fuente. Autoría

En el Gateway LoRa se reciben los datos correspondientes al nodo 2 como se observa en la siguiente figura, se tiene el identificador del nodo, el valor del nivel medido en centímetros y el RSSI de la intensidad de la señal recibida medido en dBm.

Figura 84

Recepción de datos del nodo 2 en el Gateway LoRa

```
>>> %Run serialfgf.py
Gateway LoRa
Módulo LoRa inicializado. OK!

SF: 7

BW: 125000

CR: 4/5

Paquete recibido: Nodo: 2, Nivel: 47.06, con RSSI: -67
Paquete recibido: Nodo: 2, Nivel: 51.66, con RSSI: -62
Paquete recibido: Nodo: 2, Nivel: 51.66, con RSSI: -65
Paquete recibido: Nodo: 2, Nivel: 52.09, con RSSI: -64
Paquete recibido: Nodo: 2, Nivel: 50.42, con RSSI: -65
Paquete recibido: Nodo: 2, Nivel: 51.68, con RSSI: -64
Paquete recibido: Nodo: 2, Nivel: 49.59, con RSSI: -67
Paquete recibido: Nodo: 2, Nivel: 52.09, con RSSI: -65
```

Fuente. Autoría

4.3.3. Recepción de Datos en el Gateway LoRa

Para la recepción de datos en el Gateway LoRa se utiliza la lectura mediante el puerto serial, aquí se obtiene la configuración del módulo LoRa inicializado y los paquetes recibidos de cada uno de los nodos, en este caso se los identifica como paquete del nodo 1 y paquete del nodo 2 con los respectivos valores de nivel y RSSI de la señal. En la siguiente figura puede visualizarse esto.

Figura 85

Recepción de datos en el Gateway LoRa de los dos nodos

```
>>> %Run serialfgf.py
Gateway LoRa
Módulo LoRa inicializado. OK!
SF: 7
BW: 125000
CR: 4/5
Paquete recibido: Nodo: 1, Nivel: 56.51, con RSSI: -62
Paquete recibido: Nodo: 2, Nivel: 49.03, con RSSI: -64
Paquete recibido: Nodo: 1, Nivel: 57.31, con RSSI: -62
Paquete recibido: Nodo: 2, Nivel: 51.95, con RSSI: -66
Paquete recibido: Nodo: 1, Nivel: 56.07, con RSSI: -61
Paquete recibido: Nodo: 2, Nivel: 51.95, con RSSI: -67
Paquete recibido: Nodo: 1, Nivel: 56.89, con RSSI: -62
Paquete recibido: Nodo: 2, Nivel: 48.18, con RSSI: -67
Paquete recibido: Nodo: 1, Nivel: 56.46, con RSSI: -62
Paquete recibido: Nodo: 2, Nivel: 50.71, con RSSI: -65
```

Fuente. Autoría

4.3.4. Envío de Datos a la Plataforma ThingSpeak

Para el envío de datos a la plataforma de ThingSpeak se realiza una modificación en el código del receptor, para de esta forma obtener en una cadena de texto los valores receptados por los nodos, en esta cadena únicamente se muestra el identificador del nodo, siendo estos los valores de 1 y 2, el valor del nivel medido del nivel en centímetros y la intensidad de la señal o RSSI medida en dBm. En la siguiente figura se muestra lo obtenido en el Gateway:

Figura 86

Recepción de datos depurados en el Gateway LoRa

```
1 import serial
2
3 ser = serial.Serial('/dev/ttyACM0', 115200)
4
5 while True:
6     data = ser.readline().decode('utf-8')
7     print(data)
8
```

Shell x

```
>>> %Run serialfgf.py
1,56.83,-63
2,51.90,-68
1,56.41,-64
2,50.66,-68
1,56.00,-63
2,49.83,-67
1,56.00,-64
2,49.83,-69
1,56.43,-63
2,48.14,-69
1,56.82,-63
2,48.14,-66
```

Fuente. Autoría

Para enviar los datos obtenidos de los sensores a ThingSpeak, se selecciona en base a la identificación del nodo el valor del nivel y el RSSI, estos se asignan a un campo para la construcción del payload que será enviado utilizando el protocolo MQTT al servidor respectivo.

Figura 87

Envío de datos a la plataforma de ThingSpeak

The screenshot shows the Thonny IDE interface on a Raspberry Pi. The main window displays a Python script with the following code:

```

149 # Publicar los datos en el canal de ThingSpeak si se han obtenido valores de nodo 1 y nodo 2
150 if nodo1 != 0 and nodo2 != 0:
151     payload = "field1="+str(nodo1)+"&field2="+str(nodo2)+"&field3="+str(rssi1)+"&field4="+str(rssi2)
152     print(payload)
153     publish.single(topic, payload, hostname=mqtt_host, transport=t_transport, port=t_port, client_id=
154     nodo1 = 0
155     nodo2 = 0
156     rssi1 = 0
157     rssi2 = 0
  
```

Below the code editor is a terminal window showing the output of the script. The output consists of several lines of data, with two specific lines highlighted by red boxes and arrows:

```

field1=54.76&field2=47.28&field3=-65
&field4=-66
Nodo 2: 51.88-66
field2=51.88&field4=-66
Nodo 1: 56.41-64
field1=56.41&field2=51.88&field3=-64
&field4=-66
Nodo 2: 48.55-66
field2=48.55&field4=-66
Nodo 1: 56.43-64
field1=56.43&field2=48.55&field3=-64
&field4=-66
  
```

Red boxes highlight the lines "Nodo 2: 51.88-66" and "Nodo 1: 56.43-64". Red arrows point from these boxes to the text "Datos del nodo 2" and "Datos del nodo 1" respectively.

Fuente. Autoría

4.4. Verificación y Presentación de los Datos

Una vez establecidas las pruebas de comunicación entre los nodos del sistema, se procede a evidenciar la comunicación y envío de datos de los sensores al Gateway, este es el encargado de procesar los datos acordes a cada nodo y enviar los datos al servidor IoT para su respectiva visualización.

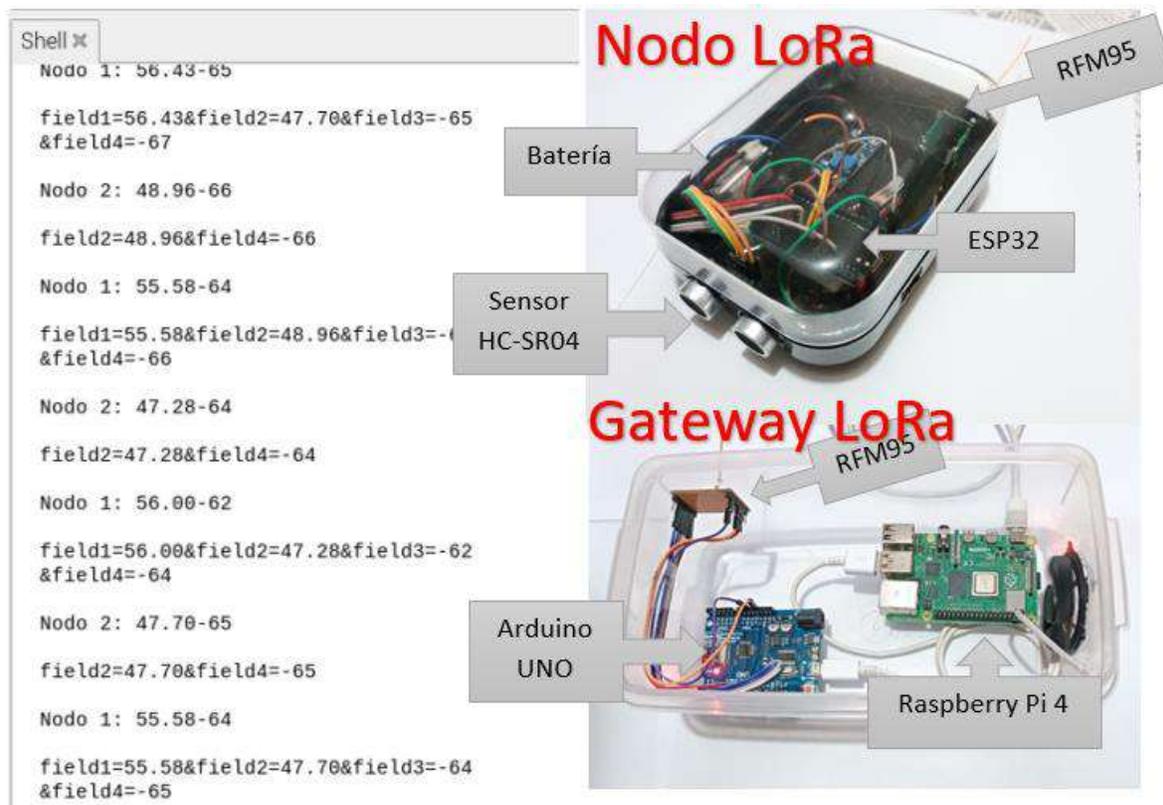
4.4.1. Comunicación entre el Nodo Sensor y el Gateway LoRa

Para esta prueba, el nodo sensor se implementa con una batería de litio, la cual le brinda energía de forma autónoma sin necesidad de conexión mediante cable USB, esto

permite que el nodo tenga movilidad y envíe los datos recolectados en la posición que se encuentre.

Figura 88

Recepción de datos del nodo sensor en el Gateway



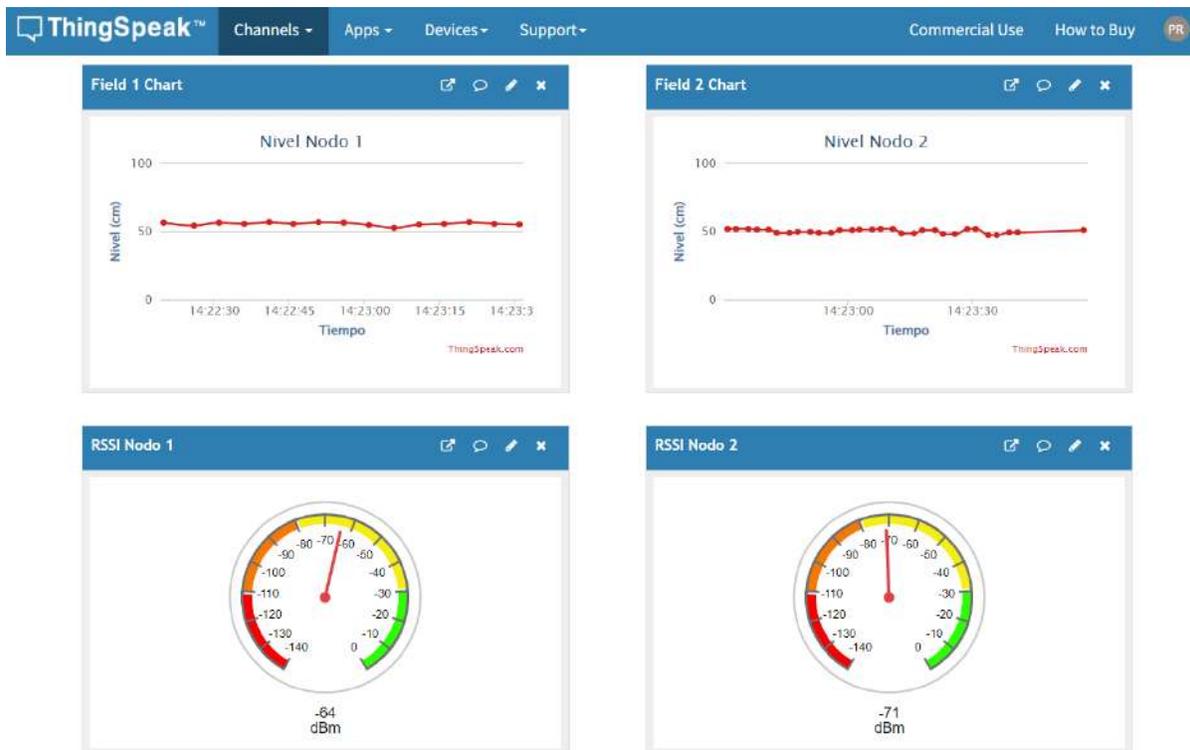
Fuente. Autoría

4.4.2. Recepción y Presentación de los Datos en la Plataforma ThingSpeak

Mediante el protocolo MQTT se envían los datos a la plataforma de ThingSpeak, a continuación, se puede observar los datos en la plataforma, en donde se evidencia las mediciones realizadas del nivel tanto del nodo 1 como del nodo 2, también se observa RSSI de la señal expresado en dBm, para verificar la calidad de la señal en el enlace con los nodos.

Figura 89

Plataforma de visualización



Fuente. Autoría

Los datos son enviados mediante el protocolo MQTT al servidor de ThingSpeak, para verificar esto, realizamos una captura de los paquetes enviados mediante el sniffer Wireshark, en donde se evidencia el protocolo utilizado, el payload correspondiente a los campos para la publicación en el bróker, el puerto TCP utilizado, la dirección del servidor y el tamaño del paquete.

Figura 90

Protocolo MQTT

15449	213.321977867	192.168.5.29	54.81.146.55	MQTT	154 Connect Command
15463	213.716029199	54.81.146.55	192.168.5.29	MQTT	76 Connect Ack
15465	213.716943477	192.168.5.29	54.81.146.55	MQTT	120 Publish Message [channels/2050074/publish]
15466	213.717282211	192.168.5.29	54.81.146.55	MQTT	68 Disconnect Req
15691	213.932334359	192.168.5.29	54.81.146.55	MQTT	154 Connect Command
15797	213.628675503	54.81.146.55	192.168.5.29	MQTT	76 Connect Ack
15799	213.629365341	192.168.5.29	54.81.146.55	MQTT	120 Publish Message [channels/2050074/publish]
15800	213.629579541	192.168.5.29	54.81.146.55	MQTT	68 Disconnect Req

▶ Frame 15799: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface wlan0, id 0
 ▶ Ethernet II, Src: d8:3a:dd:01:2f:75 (d8:3a:dd:01:2f:75), Dst: Routerbo_0b:5e:3e (74:4d:28:0b:5e:3e)
 ▶ Internet Protocol Version 4, Src: 192.168.5.29, Dst: 54.81.146.55
 ▶ Transmission Control Protocol, Src Port: 51799, Dst Port: 1883, Seq: 89, Ack: 5, Len: 54
 ▶ MQ Telemetry Transport Protocol, Publish Message
 ▶ Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
 Msg Len: 52
 Topic Length: 24
 Topic: channels/2050074/publish
 Message: 6669656c64323d3236372e3837266669656c64343d2d33310d0a

```

0000  74 4d 28 8b 5e 3e d8 3a dd 01 2f 75 08 00 45 00  tm(.A>:~/u E.
0010  00 6a 80 60 49 00 40 06 2b e0 c0 a8 05 1d 36 51  -j_@_@+----6Q
0020  92 37 ca 57 07 5b fa 98 6c 09 eb d1 b1 d5 00 18  -7.W[.-l.....
0030  01 f6 1c 88 00 00 01 01 08 0a 80 a0 a2 9b c6 76  .....v
0040  38 55 30 34 00 18 63 68 61 6e 6e 65 6c 73 2f 32  004...channels/2
0050  30 35 30 30 37 34 2f 70 75 62 6c 69 73 68 66 69  050074/publishf
0060  65 6c 64 32 3d 32 36 37 2e 38 37 26 66 69 65 6c  61d2=267.87&file
0070  64 34 3d 2d 33 31 0d 0a                          d4=-31..
  
```

Message (mqtt.msg) 26 bytes

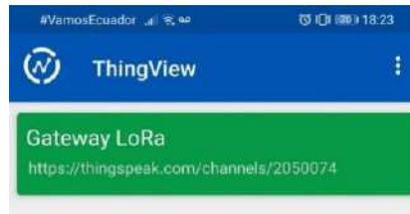
Fuente. Autoría

4.4.3. Verificación de la Aplicación Móvil ThingSpeak

ThingSpeak ofrece una versión móvil para la visualización de los datos obtenidos, para ello, solo debemos acceder mediante el ID del canal creado en la plataforma, con ello, se puede acceder a los recursos de los datos que están siendo enviados a la plataforma como se observa a continuación:

Figura 91

Canal en la App Móvil de ThingSpeak

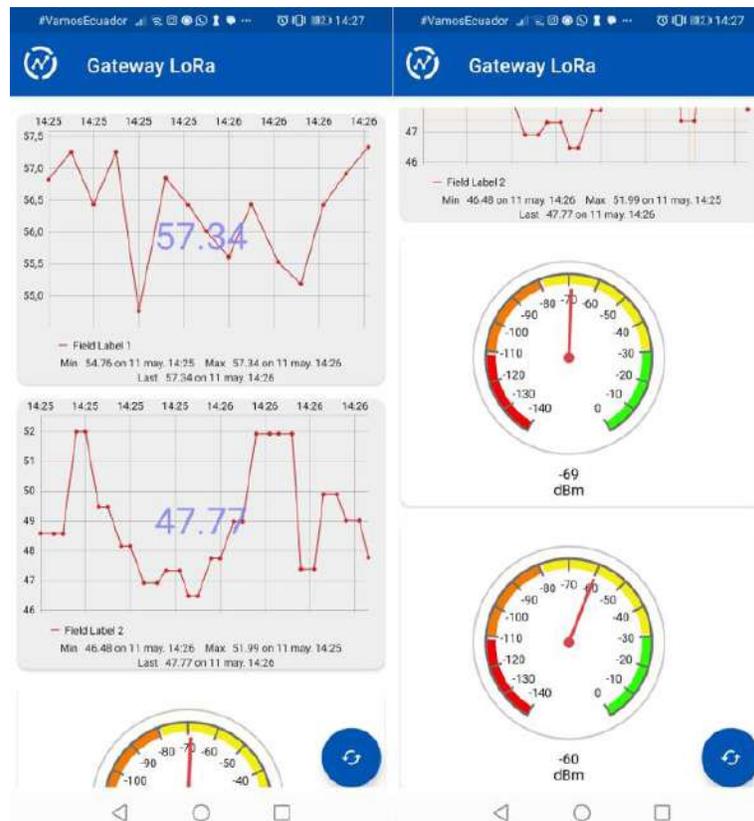


Fuente. Autoría

Una vez ingresados al canal, podemos acceder a los gráficos en donde se evidencia el nivel medido por parte de los sensores, tanto del primer y segundo sensor, por otra parte, también se muestra el valor de la intensidad de la señal recibida con el RSSI de la misma.

Figura 92

Visualización de los datos desde la App móvil de ThingSpeak



Fuente. Autoría

4.4.4. Envío del Mensaje de Alerta Mediante Correo Electrónico

En base a las pruebas realizadas, se logró identificar que el nivel del río se mantiene en un nivel constante de alrededor de 50cm a 60cm cuando el clima por la zona ha permanecido estable, por ello, se toma como valores de referencia estas medidas para la alerta que se enviará por correo electrónico, estableciendo así un rango de valores para los cuales se enviará la alerta en caso de un descenso o un incremento en el nivel.

La alerta se activará únicamente cuando el valor de nivel marque entre los valores establecidos, es decir, para informar acerca de un descenso en el nivel la alerta se enviará si el valor medido es menor a 40cm, el mensaje se enviará por correo electrónico para informar de la situación y además se enviará el último valor medido para conocimiento del mismo.

En la siguiente figura, puede mostrarse como marcaría el nivel el sensor en caso de un descenso, esto visto desde la plataforma de visualización ThingSpeak, el valor es de 39.04cm el cual debería reflejarse en la alerta generada posteriormente.

Figura 93

Visualización del descenso del nivel de agua para envío de alerta



Fuente. Autoría

Como se detecta un descenso en el nivel de agua, el sistema procede a enviar una alerta mediante correo electrónico indicando que se ha detectado un descenso en el nivel y la última lectura refleja el valor del nivel medido que corresponde al mismo visualizado en la figura anterior.

Figura 94

Visualización del mensaje de alerta para el descenso del nivel

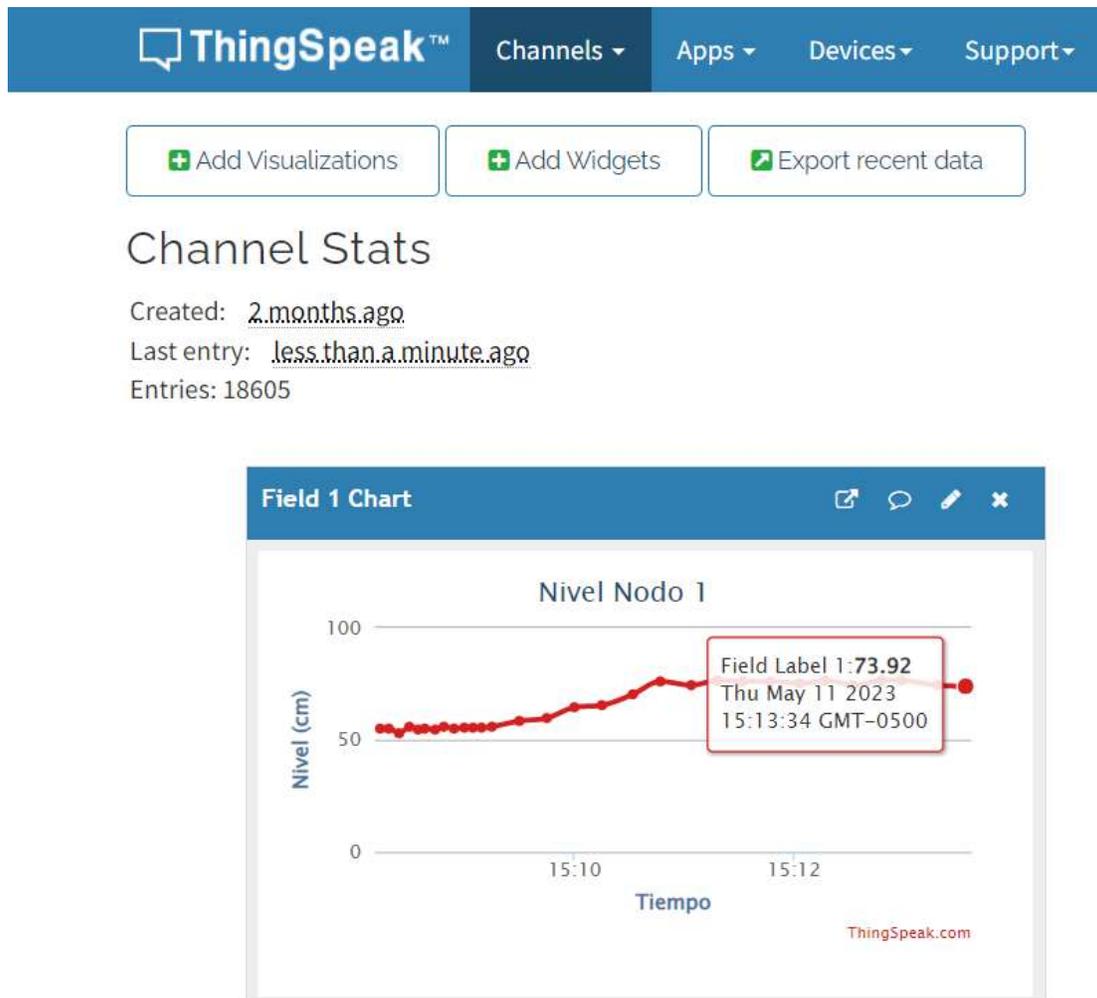


Fuente. Autoría

Por otra parte, para el aumento en el nivel del río se genera una alerta mediante correo cuando este valor ha superado los 70cm, esto puede deberse principalmente a que se presenten elevadas precipitaciones en la zona las cuales incrementan el caudal del río. En la siguiente figura, la gráfica muestra como incrementa el nivel del río a lo largo del tiempo, llegando a marcar así un valor de 73.92cm, lo que activaría la alerta correspondiente.

Figura 95

Visualización del aumento del nivel de agua para envío de alerta

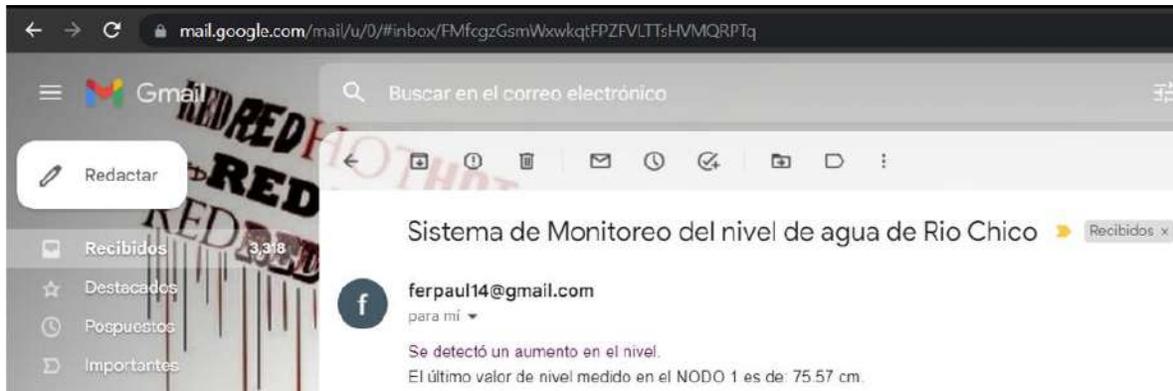


Fuente. Autoría

Como se detecta un aumento en el nivel de agua, el sistema procede a enviar una alerta mediante correo electrónico indicando que se ha detectado un incremento en el nivel y la última lectura refleja el valor del nivel medido que corresponde al mismo visualizado en la figura anterior.

Figura 96

Visualización del mensaje de alerta para el aumento del nivel



Fuente. Autoría

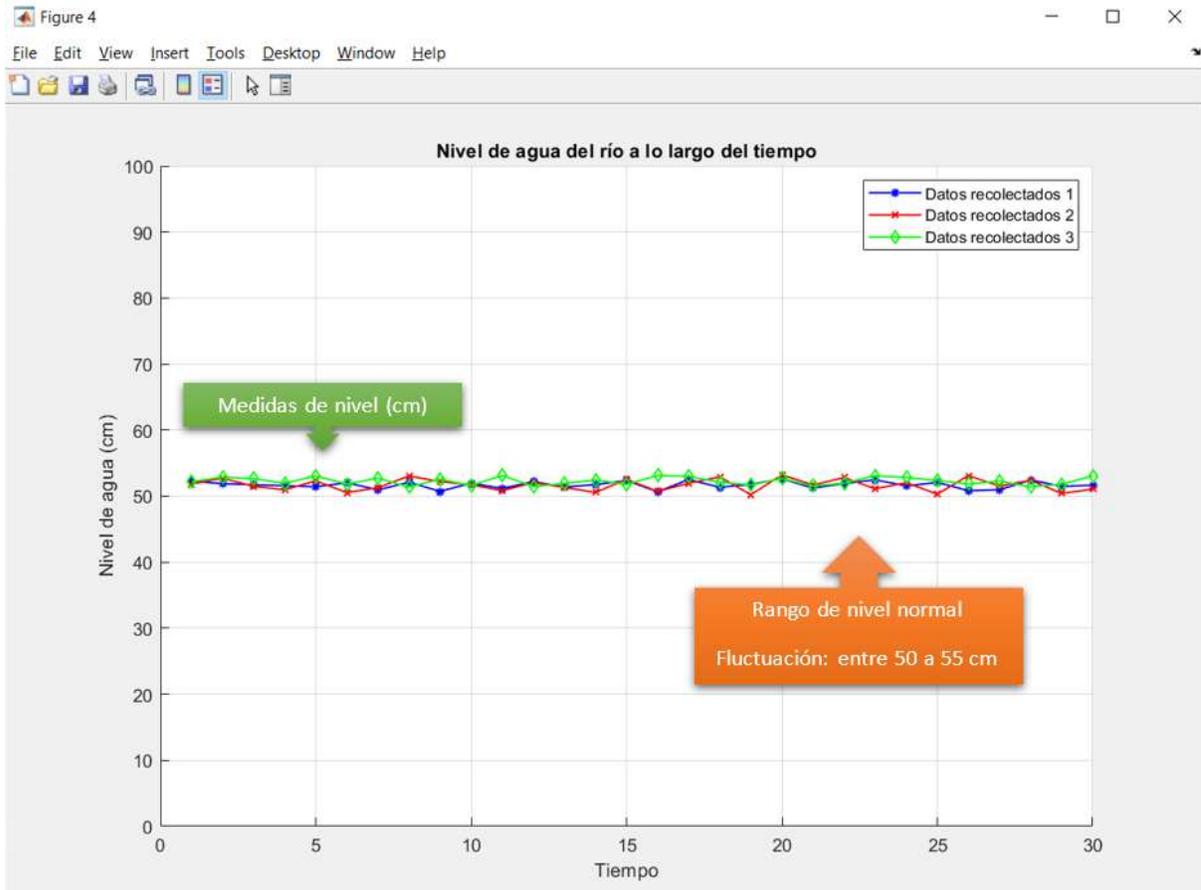
4.4.5. Patrones de Sequía

Para identificar patrones de sequía se procede a realizar un análisis de los datos recolectados a lo largo de las pruebas de medición realizados en Río Chico, para ello, los datos de pruebas se recolectaron a lo largo de diferentes días y diferentes horas de la mañana y tarde.

Utilizando MATLAB, se realizó un análisis gráfico del nivel del río a lo largo del tiempo. Los datos mostraron una variación gradual en el nivel del agua, lo que permitió determinar que las medidas obtenidas representan el nivel normal del río. Se observó que el nivel del río se mantuvo relativamente estable durante el período analizado, indicando una condición climática normal con presencia de precipitaciones moderada, por lo que, el valor del nivel se mantiene alrededor de 50 a 60 cm en estas condiciones.

Figura 97

Gráfica del nivel del río a lo largo del tiempo en base a datos recolectados



Nota. Cada una de las 3 líneas a lo largo de la gráfica representan las mediciones de nivel de agua del río realizadas en diferentes tiempos. *Fuente.* Autoría

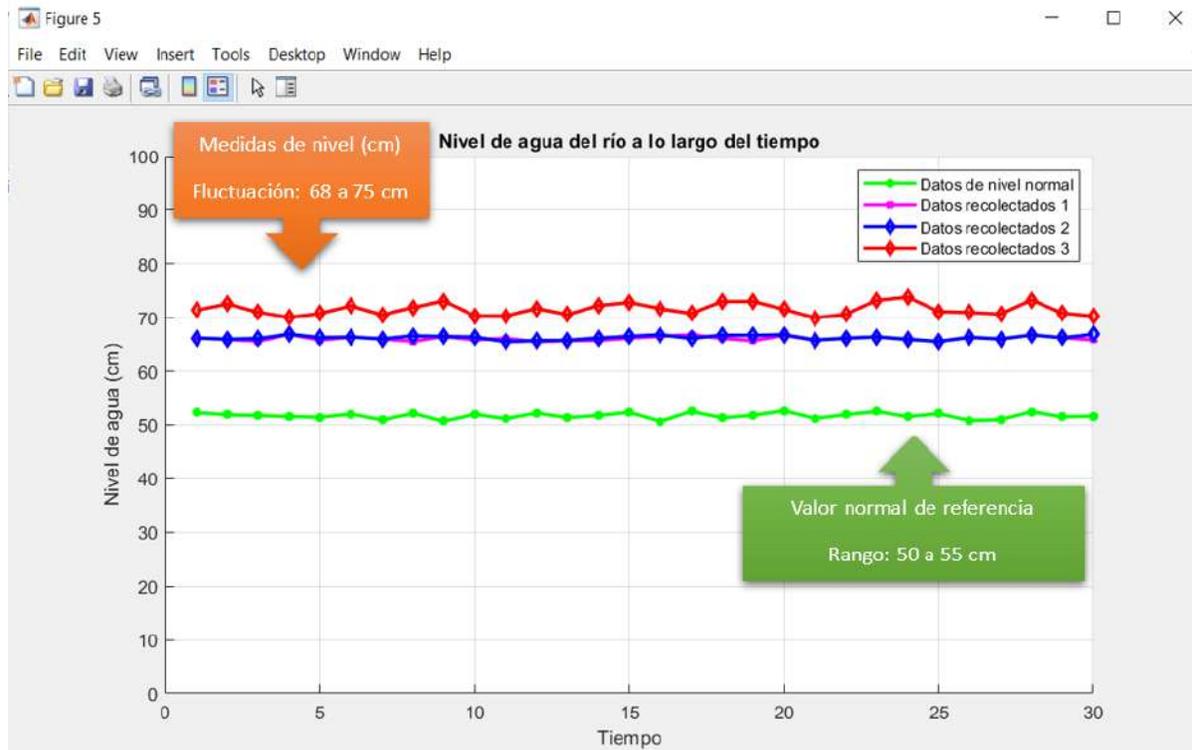
Para el siguiente caso que analizamos, podemos observar que el nivel del río ha experimentado un incremento notable. Este aumento puede atribuirse a la presencia de fuertes precipitaciones, que han generado un mayor flujo de agua en el río. La representación gráfica nos permite apreciar de manera clara y concisa estos cambios en el nivel del agua a medida que transcurre el tiempo, cabe recalcar que se realizaron pruebas de medición del nivel del río en los meses de febrero, marzo, abril y mayo, siendo esta la temporada lluviosa para la

región, pero la cantidad de precipitaciones en la zona puede variar según diferentes factores climáticos de la zona.

En la siguiente figura, se observa que, debido a la elevada cantidad de precipitaciones, las medidas de nivel variaron en relación con las medidas del nivel normal, ya que mediante el periodo analizado se presentaron gran cantidad de lluvias en la zona lo que provocó el incremento en el nivel del río el cual fue captado por los sensores mediante las mediciones realizadas en diferentes horas y días, llegó a incrementarse el nivel de 65 a 73 cm en estas condiciones.

Figura 98

Comparación entre las medidas de incremento en el nivel del río y la medida de nivel referencial.



Nota. La línea de color verde en la gráfica corresponde al valor referencial del nivel río considerado en el rango normal, las siguientes tres líneas corresponden a las medidas obtenidas cuando existe un incremento en el nivel. *Fuente.* Autoría

Para el siguiente análisis, se experimenta cuando el nivel del río ha tenido una reducción significativa. Esta disminución en el nivel del agua puede atribuirse a la baja cantidad de precipitaciones registradas en la zona. Es importante destacar que esta situación puede estar relacionada con la temporada climática en Tufiño, ya que la temporada que suele presentar menor precipitación es el período conocido como la estación seca o la temporada de sequía. Esta temporada, según (El Universo, 2021) generalmente ocurre durante los meses de junio a septiembre. Durante este período, la cantidad de lluvia disminuye

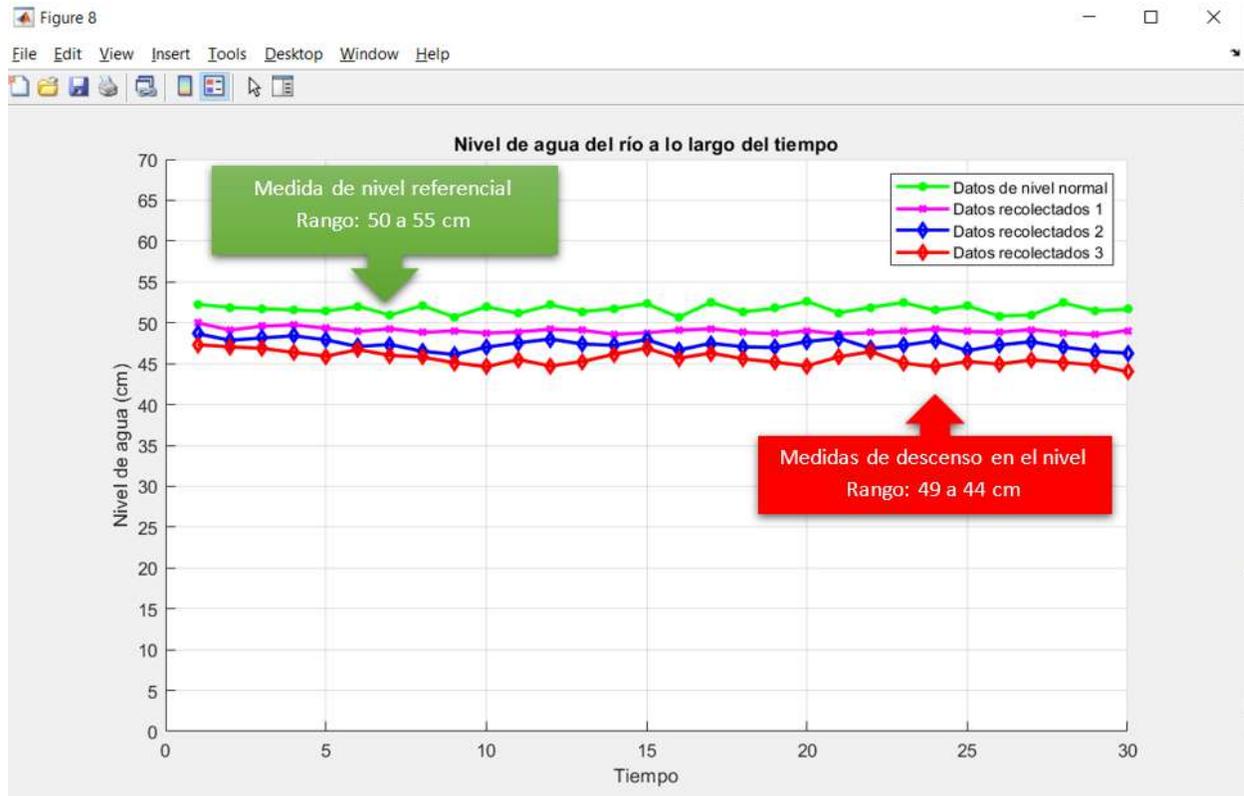
significativamente, lo que resulta en condiciones más secas y menos precipitaciones en comparación con otras épocas del año, aunque las condiciones climáticas pueden variar ligeramente de un año a otro, generalmente, este período se caracteriza por una disminución en la cantidad de precipitaciones en la región.

Los periodos con escasa lluvia pueden influir directamente en el nivel del río, generando una disminución en su caudal y, por ende, en el nivel del agua. Este particular puede tener impactos en el ecosistema local, así como en actividades humanas relacionadas con el río como: la agricultura, el suministro de agua y la navegación.

En la siguiente figura, se muestran los datos obtenidos cuando se presentan menor cantidad de precipitaciones, visualizando así que, en comparación con el nivel normal, representado con la línea de color verde, las mediciones obtenidas se encuentran por debajo de este, lo que indica que el nivel de agua del río ha descendido, lo que en conclusión lo relacionamos con el etapa de sequía en donde se registra menor cantidad de lluvias y posibles descensos en el nivel de agua de Río Chico.

Figura 99

Comparación entre las medidas de descenso en el nivel del río y la medida de nivel referencial.



Nota. La línea de color verde en la gráfica corresponde al valor referencial del nivel río considerado en el rango normal, las siguientes tres líneas corresponden a las medidas obtenidas que denotan un descenso en el nivel. *Fuente.* Autoría

En conclusión, en base a los datos de nivel obtenidos y considerando el factor climático de la zona, se pueden identificar ciertos patrones de sequía o incremento de precipitaciones. Durante el período analizado, se observa una disminución gradual en los niveles de agua registrados. Esto sugiere una posible escasez de precipitaciones en la región, lo cual puede ser indicativo de un período de sequía, aunque durante las pruebas no nos encontramos en la temporada seca, relacionamos los descensos registrados en el nivel del

río con este particular, ya que en la temporada mencionada el nivel de precipitaciones es mucho menor que en la temporada lluviosa

Por otra parte, se identifica que durante la temporada lluviosa los niveles de agua del río incrementan, las medidas obtenidas se recopilaron durante esta temporada, ya que se realizó entre los meses de febrero a mayo y la temporada lluviosa es de octubre a mayo, por lo que se encuentra dentro de este rango. En consecuencia, se puede decir que se observa una tendencia ascendente en los niveles de agua a lo largo del tiempo en la mayoría de los casos, ya que esto se relaciona con la temporada lluviosa, caso contrario, se puede decir que los descensos registrados corresponden a la estación de sequía, ya que se relacionan con la temporada en la que existe menor cantidad de precipitaciones, así que, el patrón de sequía registrado está relacionado con el factor climático de la zona.

En la siguiente tabla, se detallan las mediciones obtenidas a lo largo del período de pruebas y además se relacionan con la temporada seca y lluviosa para denotar el patrón de sequía en la zona.

Tabla 14

Detalle de las medidas obtenidas y relación con el patrón de sequía

Fecha	Nivel (cm)	Precipitaciones	Estación	Rango de meses
07/02/2023	48.67	Menores	Seca	Junio - Septiembre
07/02/2023	47.87	Menores	Seca	Junio - Septiembre
07/02/2023	48.15	Menores	Seca	Junio - Septiembre
07/02/2023	47.35	Menores	Seca	Junio - Septiembre
14/03/2023	52.19	Moderada	Lluviosa	Octubre - Mayo

14/03/2023	52.65	Moderada	Lluviosa	Octubre - Mayo
14/03/2023	51.95	Moderada	Lluviosa	Octubre - Mayo
14/03/2023	53.07	Moderada	Lluviosa	Octubre - Mayo
06/05/2023	66.85	Abundante	Lluviosa	Octubre - Mayo
06/05/2023	65.99	Abundante	Lluviosa	Octubre - Mayo
06/05/2023	66.56	Abundante	Lluviosa	Octubre - Mayo
06/05/2023	65.48	Abundante	Lluviosa	Octubre - Mayo

Fuente. Autoría

Basándonos en los datos obtenidos, podemos observar que, durante el primer período analizado, se presenta un patrón de sequía en la zona de Rio Chico en los meses correspondientes a la estación seca de Tufiño, que según el INAMHI corresponde a los meses de junio a septiembre. Los niveles de agua registrados se encuentran por debajo del nivel normal, que es cuando existe precipitación moderada con valores que fluctúan entre 50 a 55 cm.

En relación a la temporada lluviosa, es posible inferir que en base a las medidas obtenidas, los meses de octubre a mayo son la temporada en la cual se presenta el mayor porcentaje de precipitaciones, lo que influye directamente en el nivel de agua del río, por lo que el patrón de sequía se asocia a las dos estaciones que tiene la región sierra.

4.5. Prueba de la comunicación inalámbrica LoRa

Para verificar la comunicación entre los nodos de la red con el Gateway LoRa, se realizó pruebas de comunicación en diferentes distancias, con lo que se obtenía el dato enviado por el sensor más el RSSI de la señal. A continuación, se muestra una tabla en la

cual se realiza la comparación de las intensidades de la señal recibida para diferentes distancias con la configuración: SF de 7, BW de 125 kHz y CR de 4/5.

Tabla 15

Comparación distancia vs RSSI

Distancia (m)	RSSI Nodo 1 (dBm)	RSSI Nodo 2 (dBm)
5	-59	-61
10	-64	-72
20	-76	-83
30	-84	-86
40	-89	-88
50	-90	-94
70	-110	-117

Fuente. Autoría

Como se observa en la tabla anterior, se tiene que la intensidad de la señal se vuelve más débil conforma se incrementa la distancia de los nodos al Gateway, para distancias mayores la señal será más débil e incluso puede perdersela conexión de los nodos con el nodo central.

4.6. Tiempo en el aire (ToA) para la comunicación LoRa

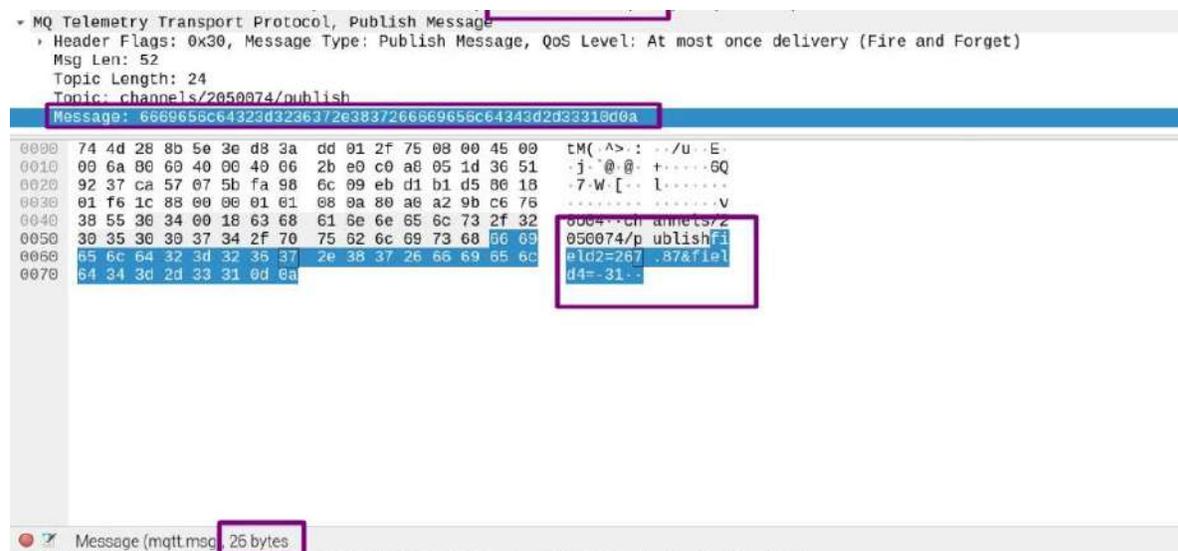
El tiempo en el aire nos permite conocer el valor de tiempo que demora un paquete LoRa en el aire entre su transmisión y recepción, es decir, es el tiempo cuando se envía una señal desde el nodo LoRa hasta que es recibida en el Gateway LoRa. El tiempo en el aire o ToA considera diferentes parámetros para su cálculo e incluye diferentes fórmulas para su

cálculo. A continuación, se realiza el cálculo teórico del ToA, teniendo en cuenta los parámetros configurados para el módulo RFM95.

Para determinar el ToA es necesario conocer los parámetros de la comunicación LoRa, debido a las configuraciones realizadas en el módulo se conoce cuáles son los parámetros utilizados a excepción del valor de payload o carga útil. Para conocer el valor de la carga útil podemos recurrir a la herramienta de Wireshark para determinar el tamaño del payload que se tiene, como se observa en la siguiente figura:

Figura 100

Tamaño del payload



Fuente. Autoría

Una vez obtenido este valor, se procede a determinar los parámetros necesarios para determinar el cálculo:

- Longitud de la carga útil (PL): 26 bytes
- Ancho de banda (BW): 125 kHz
- Spreading Factor (SF): 7

- Tasa de codificación (CR): 1
- Preámbulo ($n_{preambulo}$): Por defecto 8 bytes
- Encabezado (Header): 0 – cuando está habilitado
- Optimización de datos de baja velocidad (DE): 0 - deshabilitada

Inicialmente, se debe calcular el tiempo de símbolo con la siguiente ecuación:

$$T_s = \frac{2^{SF}}{BW}$$

$$T_s = \frac{2^7}{125 \text{ kHz}}$$

$$T_s = \frac{2^7}{125 \text{ kHz}}$$

$$T_s = 1.024 \text{ ms}$$

A continuación, se obtiene el valor de los números de símbolos para transmitir la carga útil (payloadSymbNb) con la siguiente ecuación:

$$payloadSymbNb = 8 + \left(\left(\frac{8 * PL - 4 * SF + 28 + 16 - 20 * H}{4(SF - 2 * DE)} \right) (CR + 4) \right)$$

$$payloadSymbNb = 8 + \left(\left(\frac{8 * 26 - 4 * 7 + 28 + 16 - 20 * 0}{4(7 - 2 * 0)} \right) (1 + 4) \right)$$

$$payloadSymbNb = 48 \text{ símbolos}$$

Ahora se realiza el cálculo del tiempo del preámbulo mediante la siguiente ecuación:

$$T_{preambulo} = (n_{preambulo} + 4.25) * T_s$$

$$T_{preambulo} = (8 + 4.25) * 1.024 \text{ ms}$$

$$T_{preambulo} = 12.544 \text{ ms}$$

Finalmente, se realiza el cálculo del tiempo de la carga útil mediante la siguiente ecuación:

$$T_{payload} = payloadSymbNb * T_s$$

$$T_{payload} = 48 * 1.024 \text{ ms}$$

$$T_{payload} = 49.152 \text{ ms}$$

Ahora, se obtiene el ToA mediante la siguiente ecuación:

$$ToA = T_{preambulo} + T_{payload}$$

$$ToA = 12.544\text{ms} + 49.152\text{ms}$$

$$\mathbf{ToA = 61.696 \text{ ms}}$$

A continuación, se observa en la tabla los distintos valores del ToA para diferente configuración de SF:

Tabla 16

Cálculos teóricos del ToA

SF	Tiempo de símbolo (ms)	Símbolos de la carga útil	Tiempo del preámbulo (ms)	Tiempo de la carga útil (ms)	ToA – Teórico (ms)
7	1.024	48	12.544	49.152	61.696
8	2.048	42.375	25.088	86.784	111.872
9	4.096	38	50.176	155.648	205.824

10	8.192	34.500	100.352	282.624	382.976
-----------	-------	--------	---------	---------	---------

Fuente: Autoría

Se puede evidenciar que al utilizar un valor mayor de SF el tiempo del aire se incrementa, ya que este parámetro permite una velocidad de datos mayor en SF7 que implica un menor ToA pero con un alcance menor, a medida que incrementa se reduce la velocidad de datos, aumenta el ToA, pero se incrementa el alcance.

Para la verificación de los valores obtenidos mediante los cálculos teóricos, se procede a utilizar una herramienta elaborada por Semtech, la cual es una calculadora LoRa para el ToA, que permite ingresar los parámetros de configuración y obtener dicho valor más rápidamente.

Figura 101

Calculadora LoRa para el ToA

Fuente. Autoría

A continuación, se presenta una tabla del ToA para los diferentes valores de SF obtenidos en la calculadora:

Tabla 17

Cálculos del ToA mediante la calculadora LoRa

SF	Tiempo de símbolo (ms)	Tiempo del preámbulo (ms)	ToA – Calculadora LoRa (ms)
7	1.02	12.54	61.70

8	2.05	25.09	113.15
9	4.10	50.18	205.82
10	8.19	100.35	411.65

Fuente. Autoría

Mediante el uso de la calculadora se pudo verificar que el tiempo en el aire también se incrementa conforme aumenta el SF, además los valores tanto calculados mediante las fórmulas son similares a los obtenidos en la calculadora, lo que indica que se realizó correctamente en base a los parámetros definidos.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El sistema desarrollado responde a las necesidades planteadas del proyecto, ya que permite la lectura de datos mediante los sensores para obtener el nivel de agua, y mediante la tecnología de comunicación inalámbrica LoRa se realiza la transmisión de estos, además en base a los datos de las mediciones realizadas, se pudo identificar la variación en el nivel de agua y relacionarlo con las condiciones climáticas de la zona.

Los nodos sensores transmiten el valor del nivel del río constantemente para ser visualizada en la plataforma, por lo que, con la recolección de estos datos es posible determinar si el nivel se mantiene o si existe aumento o disminución, con ello, al analizar estos datos se puede determinar si hay presencia de precipitaciones o existe sequía en la zona.

En base a las pruebas realizadas, se determina que los módulos se configuraron de manera correcta en base a los requerimientos LoRa, ya que así se puede realizar la transmisión de los datos recolectados de los nodos sensores al Gateway para su procesamiento y visualización.

La tecnología de comunicación LoRa, es una tecnología muy utilizada en redes de sensores inalámbricos, permite la interconexión de dispositivos que se encuentren a distancias más largas, por lo que, es una de las ventajas respecto a otras tecnologías de comunicación inalámbrica.

Recomendaciones

Para la conexión de los módulos se debe verificar que las placas de desarrollo tengan compatibilidad con estos, ya que, caso contrario no se podrán configurar de manera correcta, además se debe realizar la conexión adecuada de cada uno de los pines del módulo RFM95 a la placa revisando la funcionalidad de cada uno de los pines en cada elemento.

Para una comunicación más estable y confiable, debe configurarse los módulos LoRa RFM95 en los mismos parámetros de comunicación LoRa, siendo así que deben trabajar a la misma frecuencia, tener el mismo SF, BW y CR.

Los módulos RFM95 tiene un alcance teórico de hasta 3km, pero realizando las pruebas se tuvo una distancia de comunicación mucho menor, esto puede deberse a las configuraciones establecidas, ya que un SF7 permite alcanzar menores distancias que un SF10, aunque también puede deberse al entorno.

Los dispositivos deben manipularse cuidadosamente, es recomendable apagar cada uno de los subsistemas para realizar cualquier configuración física, esto con el fin de evitar daños o cortos en los elementos electrónicos.

REFERENCIAS

- Aguilar Alvarez, S. N., & Hinojosa Altamirano, R. M. (2019). *Desarrollo de un prototipo para dispensar alimento y agua para mascotas con tecnología IoT monitoreado y controlado a través de un dispositivo móvil*. Quito: Universidad Politécnica Nacional.
- ARDUINO®. (11 de Abril de 2022). *Arduino Hardware: ARDUINO®*. Obtenido de ARDUINO®: <https://www.arduino.cc/en/hardware>
- Camacho , A., Rodriguez, E., Bajaan, J., Jonathan, P., & William, C. (2019). *Redes WSN para el monitoreo de las riberas del Rio San Pablo en el Cantón La Maná*. La Maná, Cotopaxi: Universidad Técnica de Cotopaxi.
- Cantuña Escalante, K. L., & Satián Guamán, D. F. (2019). *Análisis de rendimiento de la tecnología LoRaWAN aplicado al desarrollo de un sistema de monitoreo de calidad del aire en la Unidad Educativa Santo Tomás Apóstol Riobamba*. Riobamba: Escuela Superior Politecnica de Chimborazo.
- CATSENSORS. (2021). *Tecnología LoRa y LoRaWAN*. Obtenido de CATSENSORS: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>
- Cortés Aguilar, A. (s.f.). *Instrumentos de medición de nivel*. Instituto Tecnológico.
- Creus Solé, A. (2010). *Instrumentación Industrial, Octava Edición*. Barcelona: Marcombo S.A.
- Durand, T., Visagie, L., & Booyesen, M. (2019). Evaluation of next-generation low-power communication technology to replace GSM in IoT-applications. *ResearchGate*, 1-8.
- El Universo. (30 de Septiembre de 2021). *En Tulcán, sequía y afectación de entornos naturales reducen la distribución del agua*. Obtenido de El Universo: <https://www.eluniverso.com/noticias/ecuador/en-tulcan-sequia-y-afectacion-de-entornos-naturales-reducen-la-distribucion-del-agua-nota/>
- Espressif Systems. (2023). *Development Boards: Espressif Systems*. Obtenido de Espressif: <https://www.espressif.com/en/products/devkits>
- Garrido Terán, B. S. (2021). *Diseño de un sistema inteligente de medición del consumo de agua potable para la junta de agua " El Dique" de la parroquia de San Antonio de Ibarra*. Ibarra: Universidad Técnica del Norte.
- Geek Factory. (2022). *HC-SR04 sensor ultrasónico de distancia: Geek Factory*. Obtenido de Geek Factory – Tienda de Electrónica en Línea: <https://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-ultrasonico-de-distancia/>

- Gordillo Gordillo, C. A. (2021). *Red de sensores inalámbricos bajo protocolo LoRa y gestión de proceso para la analítica de datos mediante meta sistema operativo para monitoreo ambiental en invernaderos*. Ibarra: Universidad Técnica del Norte.
- Halma Water Management. (2017). *Caudalímetros: HWM LATINOAMÉRICA*. Obtenido de HWM LATINOAMÉRICA: <https://hwm-latinoamerica.com/producto/caudalimetro-ultrasonico-beluga-copy/#>
- Hoyos Orozco, J. P. (2021). *Sistema de medición y monitoreo del nivel en canales de captación de la central hidroeléctrica Florida II*. Popayán: Uniautónoma del Cauca.
- Iacono, L., Godoy, P., Marianetti, O., Garcia Garino, C., & Párraga, C. (2012). Estudio de la Integración entre WSN y redes TCP/IP . *Ingeniería*, 57-68.
- Induanálisis Laboratorio Ambiental. (17 de Mayo de 2019). *Agua Subterránea y Superficial*. Obtenido de Induanalisis, Laboratorio, monitoreo, consultoría y equipo. Bucaramanga - Col.: https://www.induanalisis.com/publicacion/detalle/agua_subterranas_y_superficial_29
- LoRa Alliance®. (2017). *LoRaWAN™ 1.1 Regional Parameters*. LoRa Alliance®.
- LoRa Alliance®. (2022). *What are LoRa® and LoRaWAN®?* Obtenido de SEMTECH.COM. LoRa Alliance®: <https://loradevelopers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- Maker Guides. (18 de Mayo de 2022). *Comparación de la velocidad de ESP32 vs Arduino: Maker Guides*. Obtenido de Maker Guides: <https://www.makerguides.com/es/esp32-vs-arduino-speed-comparison/>
- Maldonado Borda , C. A., & Pineda Cusba, L. (2016). *Diseño de un red inalámbrica par la gestión y control de un sistema integral de sistemas de riego*. Bogotá: Universidad Distrital Francisco José de Caldas.
- MaxBotix Inc. (2022). *High Performance Ultrasonic Rangefinder: MaxBotix Inc*. Obtenido de MaxBotix Inc.: https://www.maxbotix.com/ultrasonic_sensors/mb1043.htm
- Mobilefish. (2017). *Build LoRa node using Arduino Uno and HopeRF RFM95 LoRa transceiver module: Mobilefish*. Obtenido de Mobilefish: https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_build_lora_no_de_rfm95_arduino_uno.html
- Muñoz, C., & Morales, G. (2018). *Diseño e implementación de una red de sensores inalámbricos para las condiciones ambientales físicas de un ambiente de trabajo industrial*. Popayán: Universidad del Cauca.
- Nravéaz Narvéaz, K. S., & Contreras Pérez, V. A. (2020). *Diseño y desarrollo de un prototipo de red de sensores iot utilizando tecnología LoRaWAN para el monitoreo*

- de parámetros ambientales en interiores y exteriores*. Guayaquil: Universidad Politécnica Salesiana. Sede Guayaquil.
- Ortiz Soza, M. G. (2020). *Desarrollo de una red de sensores inalámbricos utilizando tecnología LoRa para el monitoreo de un sistema*. Quito: Universidad Politécnica Salesiana - Sede Quito.
- OTT HydroMet. (2019). *Flujo y caudal: OTT HydroMet*. Obtenido de OTT HydroMet: <https://www.ott.com/es-es/aplicaciones/flujo-y-caudal-36/>
- Paspuel Güel, S. F. (2020). *Conflictos socio ambientales por derechos de agua en un contexto de cambio climático: caso Comuna La Esperanza, en la parroquia Tufiño, cantón Tulcán, provincia del Carchi*. Quito: Facultad Latinoamericana de Ciencias Sociales, FLACSO Ecuador.
- Pérez, D. (2019). Implementación de Lora y Lorawan como escenario futuro de la industrias 4.0 en el sector agroindustrial peruano. *Campus*, 133-147.
- Pickering, P. (29 de Junio de 2017). *Digi-Key Electronics. Desarrollar con LoRa para aplicaciones IoT de baja tasa y largo alcance*. Obtenido de Digi-Key Electronics. : <https://www.digikey.com/es/articles/develop-lora-for-low-rate-long-range-iot-applications>
- Proyectos con Arduino. (2022). *Sensores de Flujo de Agua para Arduino: Proyectos con Arduino*. Obtenido de Proyectos con Arduino: <https://proyectosconarduino.com/sensores/flujo-caudalimetro/>
- Santos Benavides, P. G. (2019). *Red Inalámbrica de Sensores (WSN) de Monitoreo de la Calidad del Agua para Estanques de Truchas*. Ambato: Universidad Técnica de Ambato 47.
- SEMTECH. (2022). *¿Qué es LoRa?* Obtenido de SEMTECH: <https://www.semtech.com/lora/what-is-lora>
- Silicon Laboratories. (2023). *Microcontrollers (MCUs): SILICON LABS*. Obtenido de SILICON LABS: <https://www.silabs.com/mcu>
- Stauffer , B., & Spuhler, D. (01 de Octubre de 2018). *Captación de ríos, lagos y embalses (reservorios)*. Obtenido de SSWM - Find tools for sustainable sanitation and water management!: <https://sswm.info/es/gass-perspective-es/tecnologias-de/tecnologias-de-abastecimiento-de-agua/captacion/captaci%C3%B3n-de-r%C3%ADos%2C-lagos-y-embalses-%28reservorios%29>
- Texas Instruments Incorporated. (2022). *MSP-EXP430G2ET: Texas Instruments Incorporated*. Obtenido de Texas Instruments : <https://www.ti.com/tool/MSP-EXP430G2ET>

The Things Network. (2022). *LoRaWAN®. Clases de dispositivos. The Things Network*. Obtenido de The Things Network: <https://www.thethingsnetwork.org/docs/lorawan/classes/>

Trujillo, L. C., Méndez, D., & Acosta, J. C. (2022). *An IoT-based scalable river level monitoring platform*. Bogotá: International Journal of Sensor Networks (IJSNET).

VISTRONICA S.A.S. (2023). *MODULO TRANSCCEPTOR LORA RFM95: VISTRONICA S.A.S*. Obtenido de VISTRONICA S.A.S: <https://www.vistronica.com/comunicaciones/wifi/modulo-transceptor-lora-rfm95-868mhz-detail.html>

Wireless Logic. (2022). *Redes de baja potencia y área amplia (LPWAN)*. Obtenido de Wireless Logic.: <https://www.wirelesslogic.com/es/conectar/conectividad-sim/redes-de-baja-potencia-y-area-amplia-lpwan/>

ANEXOS

ANEXO 1 – Código del Nodo Sensor

```

/*****
  Código adaptado y modificado basado en los ejemplos de Arduino LoRa
  library
  *****/
//Inclusión de librerías
#include <SPI.h>
#include <LoRa.h>

//Definir los pines para la conexión del módulo RFM95
#define nss 5
#define rst 15
#define dio0 2

//Definir el valor de la velocidad del sonido en cm/uS
#define velSonido 0.034

// Variables para configuraciones personalizadas del módulo RFM95 LoRa
const int sf = 7; // Spreading Factor (7-12)
const int bw = 125000; // Bandwidth (Hz)
const int cr = 5; // Coding Rate
const int nodeAddress = 1; // Dirección del nodo LoRa

//Definir los pines para el sensor HC-SR04
const int trigPin = 16; //pin trigger
const int echoPin = 17; //pin echo

//variables para almacenar valores medidos por el sensor HCR-SR04
long duracion; //duración del pulso
float nivelCm; //Distancia medida hacia el objeto y nivel de agua
float distanciaMax = 300; //distancia máxima de medición

//contador de paquetes enviados
int counter = 0;

void setup() {
  //Inicializar el monitor serial
  Serial.begin(115200);
  while (!Serial);
  Serial.println("Nodo LoRa 1");//imprime el nombre del nodo

```

```

//define el modo de los pines del sensor HC-SR04
pinMode(trigPin, OUTPUT); // define el pin como salida
pinMode(echoPin, INPUT); // define al pin como entrada

//define los pines para el módulo RFM95
LoRa.setPins(nss, rst, dio0);

//Inicializa LoRa en la frecuencia definida
while (!LoRa.begin(915E6)) {
  Serial.println("Falló. Revise las conexiones del módulo LoRa RFM95");
  delay(500);
}

//Inicialización de parámetros LoRa para la Tx
LoRa.setSpreadingFactor(sf); //Factor de dispersión (7-12)
LoRa.setSignalBandwidth(bw); //Ancho de banda (125-250-500 kHz)
LoRa.setCodingRate4(cr); //Coding Rate
LoRa.enableCrc(); // Habilita la comprobación de redundancia cíclica (CRC)
LoRa.receive(); // Habilita el modo de recepción continua (clase C)
LoRa.setSyncWord(0xF3); //sincroniza con los elemnetos de la red

//Impresión de los parámetros configurados en el módulo
Serial.println("Modulo LoRa inicializado. OK!");
Serial.print("SF: ");
Serial.println(sf);
Serial.print("BW: ");
Serial.println(bw);
Serial.print("CR: 4/");
Serial.println(cr);
}

void loop() {
  //Definición de variables para obtener el valor promedio de 5 medidas
  float promedio = 0;
  int muestras = 5;

  // cambia el estado del pin trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Lee el pin echo, se obtiene el tiempo que viajó la onde en us
  duracion = pulseIn(echoPin, HIGH);

```

```

// Se calcula la distancia y se obtiene el promedio para 5 muestras
for(int i = 0; i < muestras; i++){
  nivelCm = duracion * velSonido/2;
  promedio = promedio + nivelCm;
}
//Se obtiene el nivel restando la medida obtenida menos la distancia
máxima
nivelCm = distanciaMax-(promedio/muestras);

//Impresión de los valores del nodo 1
Serial.print("Nodo 1. Paquete: ");
Serial.print(counter);
Serial.print(", Nivel: ");
nivelCm = distanciaMax-(promedio/muestras); //se obtiene el nivel restando
la medida obtenida menos la distancia máxima
Serial.println(nivelCm);

//Paquete LoRa con los datos a enviar
LoRa.beginPacket(); //inicializa el paquete
LoRa.write(nodeAddress); //se envia la dirección del nodo
LoRa.print(nivelCm); //se envia el valor medido
LoRa.endPacket(); //finaliza el paquete

counter++; //incrementa el contador

delay(10000); //espera 10 segundos para enviar un nuevo dato
}

```

ANEXO 2 – Código del Gateway LoRa

Arduino UNO

```

/*****
  Código adaptado y modificado basado en los ejemplos de Arduino LoRa
  library
  *****/
//Inclusión de librerías
#include <SPI.h>
#include <LoRa.h>
#include <avr/wdt.h>

//Definir los pines para la conexión del módulo RFM95
#define nss 10

```

```

#define rst 9
#define dio0 2

// Variables para configuraciones personalizadas del módulo RFM95 LoRa
const int sf = 7; // Spreading Factor (7-12)
const long bw = 125E3; // Bandwidth (Hz)
const int cr = 5; // Coding Rate

//Variables para definir las direcciones de los nodos Tx LoRa
const int nodo1 = 1;
const int nodo2 = 2;

void setup() {
  //Inicializar el monitor serial
  Serial.begin(115200);
  while (!Serial);
  Serial.println("Gateway LoRa");//imprime el nombre del nodo

  //Definir los pines para el modulo LoRa RFM95
  LoRa.setPins(nss, rst, dio0);

  //Inicializa LoRa en la frecuencia definida
  while (!LoRa.begin(915E6)) {
    Serial.println("Falló. Revise las conexiones del módulo LoRa RFM95");
    delay(500);
  }

  //Inicialización de parámetros LoRa para la Rx
  LoRa.setSpreadingFactor(sf); //Factor de dispersión (7-12)
  LoRa.setSignalBandwidth(bw); //Ancho de banda (125-250-500 kHz)
  LoRa.setCodingRate4(cr); //Coding Rate
  LoRa.enableCrc(); // Habilita la comprobación de redundancia cíclica (CRC)
  LoRa.receive(); // Habilita el modo de recepción continua (clase C)
  LoRa.setSyncWord(0xF3);//sincroniza con los elemnetos de la red

  //Impresión de los parámetros configurados en el módulo
  Serial.println("Módulo LoRa inicializado. OK!");
  Serial.print("SF: ");
  Serial.println(sf);
  Serial.print("BW: ");
  Serial.println(bw);
  Serial.print("CR: 4/");
  Serial.println(cr);
}

```

```

void loop() {
  // Determina si hay un paquete para leer buffer de recepción de la
  librería LoRa.
  int packetSize = LoRa.parsePacket();

  //Si se ha obtenido un paquete leer su contenido
  if (packetSize) {

    //Lee la dirección del nodo que envía el paquete.
    uint8_t nodeAddress = LoRa.read();

    Serial.print("Paquete recibido: ");

    //Crea la variable message para almacenar el mensaje de recepción LoRa
    String message = "";

    // Mientras haya paquetes disponibles para leer en el bufer se ejecuta
    las instrucciones
    while (LoRa.available()) {
      message += (char)LoRa.read();//concatena cada mensaje recibido a la
      cadena message
    }

    //verifica la dirección del nodo transmisor para imprimir el dato
    correspondiente a cada uno
    if (nodeAddress == nodo1 || nodeAddress == nodo2){
      //Se imprimen los datos para enviar por el puerto serial
      Serial.print("Nodo: ");
      Serial.print(String(nodeAddress));
      Serial.print(", Nivel: " );
      Serial.print(message);
      Serial.print(", con RSSI: ");
      Serial.println(LoRa.packetRssi()); // imprime el RSSI del paquete
      recibido
    }
  }
  wdt_reset(); // Reiniciar el WDT
}

```

Raspberry Pi 4

```
#Importación de librerías
```

```

import serial
import paho.mqtt.publish as publish
import time

# Inicializar las variables de nivel y RSSI
nodo1 = 0
nodo2 = 0
rssi1 = 0
rssi2 = 0

# Datos para acceder a ThingSpeak
write_api_key = "2B63MJC2V4RDAB8D" #clave para acceso a la API
channel_ID = "2050074" #Identificador del canal de thingspeak
mqtt_host = "mqtt3.thingspeak.com" # Servidor MQTT de ThingSpeak
t_transport = "tcp" #protocolo de transporte
t_port = 1883 # Puerto MQTT de ThingSpeak
topic = "channels/" + channel_ID + "/publish" #topic para publicar en
ThingSpeak

# Credenciales del dispositivo creado para acceder a MQTT de thingspeak
mqtt_client_ID = "HwE2Ex8mPQUqEygCNikRLgk" #id del cliente mqtt
mqtt_username = "HwE2Ex8mPQUqEygCNikRLgk" #usuario del cliente mqtt
mqtt_password = "HJVRJ7Njvyf6sJT8GQGYF43Z" #clave de acceso para el
cliente mqtt

#inicializacion del puerto serial, se define el puerto USB y el baudrate
ser = serial.Serial('/dev/ttyUSB0', 115200)

while True:
    if ser.in_waiting > 0: #verifica que haya datos en el puerto serie
        data = ser.readline().decode('utf-8') #leer los datos y
        decodificarlos
        #print(data)
        datos=data.split(",") #separa la cadena de datos utilizando el
        identificador ","
        # Comprobar si los datos contienen la cadena 1 en la posición 0
        para identificar al Nodo 1"
        if "1" in datos[0]:
            # Si los datos contienen la cadena "1", extrae el valor
            numérico de la posición correspondiente
            nodo1 = datos[1]
            rssi1 = datos[2]
            print("Nodo 1: " + nodo1 + rssi1) #imprime el valor recibido
            del nodo 1

            # Comprobar si los datos contienen la cadena 2 en la posición 0
            para identificar al Nodo 1"
            if "2" in datos[0]:
                # Si los datos contienen la cadena "2", extrae el valor
                numérico de la posición correspondiente
                nodo2 = datos[1]
                rssi2 = datos[2]

```

```

        print("Nodo 2: " + nodo2 + rssi2) #imprime el valor recibido
del nodo 2

    # Construir la cadena de carga útil cuando no hay datos en el
nodo 1
    if nodol == 0:
        payload = "field2="+str(nodo2)+"&field4="+str(rssi2) #datos
del nodo 2
        #publicar los datos por mqtt
        publish.single(topic, payload, hostname=mqtt_host,
transport=t_transport, port=t_port, client_id=mqtt_client_ID,
auth={'username':mqtt_username,'password':mqtt_password})
        print(payload)
    # Construir la cadena de carga útil cuando no hay datos en el
nodo 2
    if nodo2 == 0:
        payload = "field1="+str(nodol)+"&field3="+str(rssi1) #datos
del nodol
        #publicar los datos por mqtt
        publish.single(topic, payload, hostname=mqtt_host,
transport=t_transport, port=t_port, client_id=mqtt_client_ID,
auth={'username':mqtt_username,'password':mqtt_password})
        print(payload)

    # Publicar los datos en el canal de ThingSpeak si se han obtenido
valores de nodo 1 y nodo 2
    if nodol != 0 and nodo2 != 0:
        payload =
"field1="+str(nodol)+"&field2="+str(nodo2)+"&field3="+str(rssi1)+"&field4
="+str(rssi2)
        print(payload)
        publish.single(topic, payload, hostname=mqtt_host,
transport=t_transport, port=t_port, client_id=mqtt_client_ID,
auth={'username':mqtt_username,'password':mqtt_password})
        nodol = 0
        nodo2 = 0
        rssi1 = 0
        rssi2 = 0

```