



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

**IMPLEMENTACIÓN DE SISTEMA DE ALERTA TEMPRANA FRENTE A  
INUNDACIONES PROVOCADAS POR EL DESBORDE DEL RÍO BLANCO EN LA  
PARROQUIA DE VALLE HERMOSO.**

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

**AUTOR: NAVARRETE GARCÍA JESÚS ALBERTO**

**DIRECTOR: JARAMILLO VINUEZA EDGAR DANIEL**

**IBARRA-ECUADOR**

**2024**



# 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			
<b>CÉDULA DE IDENTIDAD:</b>	1724688419		
<b>APELLIDOS Y NOMBRES:</b>	NAVARRETE GARCÍA JESÚS ALBERTO		
<b>DIRECCIÓN:</b>	Ibarra, Av. 13 de Abril y Tulcán.		
<b>EMAIL:</b>	janavarreteg@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	062558174	<b>TELÉFONO MÓVIL:</b>	0983422812

DATOS DE LA OBRA	
<b>TÍTULO:</b>	IMPLEMENTACIÓN DE SISTEMA DE ALERTA TEMPRANA FRENTE A INUNDACIONES PROVOCADAS POR EL DESBORDE DEL RÍO BLANCO EN LA PARROQUIA DE VALLE HERMOSO.
<b>AUTOR (ES):</b>	Navarrete García Jesús Alberto
<b>FECHA DE APROBACIÓN: DD/MM/AAAA</b>	15/10/2024
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniero en Telecomunicaciones
<b>ASESOR /DIRECTOR:</b>	SUAREZ ZAMBRANO LUIS EDILBERTO/ JARAMILLO VINUEZA EDGAR DANIEL

## 2. CONSTANCIAS

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

Ibarra, a los 15 días del mes de Octubre de 2024

### EL AUTOR:

(Firma).....

Nombre: Jesús Alberto Navarrete García

## **AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Navarrete García Jesús Alberto, con cédula de identidad Nro. 1724688419 , en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 16 días del mes de septiembre de 2024

### **EL AUTOR:**

.....

Navarrete García Jesús Alberto



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN:**

MAGÍSTER JARAMILLO VINUEZA EDGAR DANIEL, DIRECTOR DEL PRESENTE

TRABAJO DE TITULACIÓN CERTIFICA:

Que el presente trabajo de Titulación IMPLEMENTACIÓN DE SISTEMA DE ALERTA TEMPRANA FRENTE A INUNDACIONES PROVOCADAS POR EL DESBORDE DEL RÍO BLANCO EN LA PARROQUIA DE VALLE HERMOSO, ha sido desarrollado por el señor Navarrete García Jesús Alberto bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. JARAMILLO VINUEZA EDGAR DANIEL  
**DIRECTOR**

## **APROBACIÓN DEL COMITÉ CALIFICADOR**

El Comité Calificado del trabajo de Integración Curricular “IMPLEMENTACIÓN DE SISTEMA DE ALERTA TEMPRANA FRENTE A INUNDACIONES PROVOCADAS POR EL DESBORDE DEL RÍO BLANCO EN LA PARROQUIA DE VALLE HERMOSO.” elaborado por Jesús Alberto Navarrete García, previo a la obtención del título del Ingeniero en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

MSc. Jaramillo Vinueza Edgar Daniel

MSc. Luis Edilberto Suárez Zambrano



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### DEDICATORIA

*El presente trabajo está dedicado a los ciudadanos de Valle Hermoso, en reconocimiento a su esfuerzo y colaboración en cada etapa del proceso, y por su incansable lucha por convertir esta parroquia en un lugar digno y seguro para todos. Sin su participación activa y compromiso, este proyecto no habría sido posible. Este logro es un reflejo de su dedicación hacia el bienestar de su comunidad.*



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### AGRADECIMIENTO

A lo largo de este proyecto, he tenido la suerte de contar con el apoyo de diversas personas e instituciones que han sido fundamentales para su desarrollo y éxito.

Quiero expresar mi más profundo agradecimiento a la Universidad Técnica del Norte por brindarme la oportunidad de formarme y crecer en un entorno académico de excelencia. A los ingenieros que me guiaron durante este proceso, por su paciencia, sus valiosos conocimientos y su constante disposición para orientarme en cada desafío.

A mis compañeros de estudio, quienes no solo han sido colegas, sino amigos con los que compartí largas jornadas de trabajo, reflexión y aprendizaje mutuo. Su apoyo fue clave en momentos difíciles.

A mi familia, que siempre ha estado a mi lado brindándome su amor y apoyo incondicional. Sin ellos, este proyecto no habría sido posible.

Finalmente, quiero extender un agradecimiento especial a las familias que, a lo largo de este recorrido, me acogieron como uno de los suyos. Su hospitalidad y generosidad me hicieron sentir en casa, y por ello estaré eternamente agradecido.

## RESUMEN

El proyecto se centra en el diseño e implementación de un sistema de alerta temprana para mitigar los efectos de inundaciones provocadas por el desbordamiento de ríos en áreas vulnerables. El sistema utiliza una combinación de sensores de monitoreo continuo que miden el nivel del agua en puntos estratégicos y una infraestructura tecnológica que permite la transmisión y procesamiento de datos en tiempo real. Cuando se detectan niveles de agua que exceden los límites preestablecidos, el sistema genera alertas automáticas que son enviadas a la comunidad a través de canales de comunicación, como mensajes de texto o aplicaciones de mensajería instantánea. Estas alertas brindan tiempo valioso para que las autoridades locales y los ciudadanos puedan tomar medidas preventivas, como la evacuación o la activación de planes de emergencia.

El enfoque del sistema es la prevención y la reducción del riesgo, proporcionando una respuesta rápida ante posibles desastres naturales. El proyecto destaca la importancia de la tecnología como una herramienta clave para mejorar la resiliencia de las comunidades frente a eventos climáticos extremos, promoviendo una gestión proactiva del riesgo y la seguridad.

**Palabras clave:** Alerta temprana, inundaciones, monitoreo de nivel de agua, gestión de riesgo, sistema de alerta, desbordamiento de ríos, prevención de desastres, tecnología para la resiliencia, comunicación de emergencias, evacuación preventiva.

## ABSTRACT

This project focuses on the development and implementation of an early warning system aimed at reducing the impact of floods caused by river overflows in vulnerable areas. The system integrates continuous monitoring sensors that measure water levels at strategic points and a technological infrastructure that enables real-time data transmission and processing. When water levels exceed predefined thresholds, the system automatically sends alerts to the community through communication channels such as text messages or instant messaging apps. These alerts provide critical time for local authorities and citizens to take preventive measures, such as evacuation or activating emergency plans.

The system emphasizes prevention and risk reduction, offering a rapid response to potential natural disasters. The project highlights the role of technology as a crucial tool in enhancing community resilience against extreme weather events, promoting proactive risk management and safety.

**Keywords:** Early warning, floods, water level monitoring, risk management, alert system, river overflow, disaster prevention, resilience technology, emergency communication, preventive evacuation.

## INDICE DE CONTENIDOS

Capítulo I: Introducción .....	1
1.1 Tema.....	1
1.2 Problema de investigación .....	1
1.3 Objetivos .....	2
Objetivo General.....	2
Objetivos Específicos.....	2
1.4 Alcance del proyecto.....	2
1.5 Justificación.....	4
Capítulo II: Fundamentos teóricos.....	5
1.1. Gestión de riesgos.....	5
1.1.1. Riesgo .....	6
1.1.2. Desastre natural.....	6
1.1.3. Inundaciones .....	7
1.1.4. Sistema de alerta temprana (SAT).....	7
1.1.5. Componentes de un sistema de alerta temprana ante inundaciones (SATI)..	9
1.2. Situación actual de la parroquia de Valle Hermoso .....	11

1.3.	Internet de las cosas .....	12
1.3.1.	Arquitectura .....	14
1.3.2.	Protocolos de capa aplicación .....	16
1.3.3.	Tecnologías de red .....	17
1.4.	LoRaWAN (Long Range Wide Area Network) .....	19
1.4.1.	Arquitectura de red .....	20
1.4.2.	Tipos de dispositivos.....	22
1.4.3.	Resumen regional de LoRaWAN™ .....	23
1.4.4.	LoRaWAN™ para Ecuador AU915-928.....	25
Capítulo III: Materiales y Métodos.....		26
2.1.	Organización Comunitaria.....	29
2.2.	Reconocimiento de la Cuenca Menor.....	30
2.2.1.	Identificación de cuerpos de agua.....	30
2.3.	Medición de Lluvia y Nivel de Agua de los Ríos.....	32
2.4.	Funcionamiento del Sistema de Alerta.....	35
2.4.1.	Etapa de toma de datos .....	36
2.4.1.1.	Tipos de sensores .....	36
2.4.1.2.	Elección del sensor de nivel.....	37
2.4.1.3.	Microcontroladores.....	39
2.4.1.4.	Tecnología inalámbrica.....	41

2.4.1.5.	Modulo LORA .....	43
2.4.1.6.	Diseño del sistema de toma de datos .....	45
2.4.1.7.	Configuración de transmisor lora.....	47
2.4.1.8.	Programación del Emisor.....	50
2.4.2.	Procesamiento de datos.....	51
3.4.2.1.	Receptor LORA .....	52
3.4.2.2.	Elección de plataforma IOT.....	56
3.4.2.3.	Diseño del sistema de procesamiento de datos .....	58
3.4.2.4.	Programación del sistema receptor .....	61
3.4.2.5.	Programación del sistema de proceso de datos.....	62
3.4.3.	Plataforma IoT .....	64
3.4.3.1.	Dispositivos.....	65
3.4.3.2.	Data Buckets .....	71
3.4.3.3.	Dashboards.....	76
3.4.4.	Etapa de difusión de la información .....	80
3.4.4.1.	Telegram .....	82
3.4.4.2.	Aplicación de Thinger.io .....	84
3.4.5.	Sistema final.....	89
3.5.	Evaluación de la Situación, Difusión de la Alerta y Plan de Emergencia.....	93
Capítulo IV:	Resultados y análisis .....	94

4.1.	Instalación de los dispositivos .....	94
4.2.	Transmisión de datos .....	96
4.3.	Análisis de datos en temporada de riesgo alto.....	102
4.4.	Análisis de datos en temporada de riesgo bajo.....	107
4.5.	Medición de tiempo de respuesta. ....	111
	Conclusiones y recomendaciones .....	113
	Conclusiones .....	113
	Recomendaciones.....	114
	Bibliografía .....	116
	ANEXOS .....	119
	Anexo 1: Código del módulo Emisor.....	119
	Anexo 2: Código del módulo Receptor.....	122
	Anexo 3: Código del sistema de procesamiento de datos. ....	124
	Anexo 4: Código del servicio de subida de datos. ....	133
	Anexo 5: Manual para el ingreso al sistema de alertas. ....	143
	Introducción .....	145
	Generalidades.....	146
	Resumen del sistema.....	148
	¿Como acceder a la función de alerta? .....	148
	Instalar la Aplicación de Telegram: .....	148

Crear una Cuenta de Telegram: .....	149
Acceder al Canal de Alertas:.....	160
Unirse al Canal:.....	160
Revisar Mensajes del Canal: .....	160
Anexo 6: Planificación de simulacro. ....	161
1.    Introducción .....	164
2.    Objetivos .....	165
3.    Cronograma.....	166
<b>Figura 1.</b> <i>Sistema Comunitario de Alerta Temprana ante Inundaciones.</i> .....	11
<b>Figura 2.</b> <i>Arquitecturas IOT.</i> .....	14
<b>Figura 3.</b> <i>Arquitectura de Red.</i> .....	21
<b>Figura 4.</b> <i>Clases de dispositivos.</i> .....	22
<b>Figura 5</b> <i>Imagen de la zona del satélite Sentinel s1.</i> .....	31
<b>Figura 6</b> <i>Mapa hidrográfico.</i> .....	32
<b>Figura 7</b> <i>Imagen satelital del satélite Chirps</i> .....	33
<b>Figura 8</b> <i>Precipitación por fechas.</i> .....	34
<b>Figura 9</b> <i>Diagrama de conexiones</i> .....	46
<b>Figura 10</b> <i>Diseño de placa.</i> .....	47
<b>Figura 11</b> <i>Datos de configuración.</i> .....	48
<b>Figura 12</b> <i>Diagrama de conexiones.</i> .....	59

<b>Figura 13</b> <i>Diseño de PCV.</i> .....	60
<b>Figura 14</b> <i>Conexión de la raspberry con el ESP32.</i> .....	61
<b>Figura 15</b> <i>Página principal de Thinger.io.</i> .....	65
<b>Figura 16</b> <i>Opción Add Devices.</i> .....	66
<b>Figura 17</b> <i>Creación de nuevo dispositivo.</i> .....	66
<b>Figura 18</b> <i>Dispositivo creado.</i> .....	67
<b>Figura 19</b> <i>Página principal del dispositivo.</i> .....	68
<b>Figura 20</b> <i>Menú de APIs</i> .....	68
<b>Figura 21</b> <i>Datos obtenidos.</i> .....	69
<b>Figura 22</b> <i>Menú de tokens.</i> .....	70
<b>Figura 23</b> <i>Opción Add.</i> .....	70
<b>Figura 24</b> <i>Creación de Tokens.</i> .....	71
<b>Figura 25</b> <i>Opción de Data Buckets.</i> .....	72
<b>Figura 26</b> <i>Opción Add Bucket.</i> .....	72
<b>Figura 27</b> <i>Creación de Bucket.</i> .....	73
<b>Figura 28</b> <i>Buckets creados.</i> .....	74
<b>Figura 29</b> <i>Información de bucket.</i> .....	75
<b>Figura 30</b> <i>Exportación de bucket.</i> .....	76
<b>Figura 31</b> <i>Opción de Dashboards.</i> .....	77
<b>Figura 32</b> <i>Opción Add Dashboard.</i> .....	77
<b>Figura 33</b> <i>Creación de dashboard.</i> .....	78
<b>Figura 34</b> <i>Dashboard creado.</i> .....	78
<b>Figura 35</b> <i>Opción Add Widget.</i> .....	79

<b>Figura 36</b> <i>Creación de widget.</i> .....	79
<b>Figura 37</b> <i>Configuración de widget.</i> .....	80
<b>Figura 38</b> <i>Widgets creados.</i> .....	80
<b>Figura 39</b> <i>Comparación entre tipos de endpoints.</i> .....	81
<b>Figura 40</b> <i>Creación de un bot de telegram.</i> .....	82
<b>Figura 41</b> <i>Obtención del ID del chat.</i> .....	83
<b>Figura 42</b> <i>Opción Endpoints.</i> .....	83
<b>Figura 43</b> <i>Configuración de Endpoint.</i> .....	84
<b>Figura 44</b> <i>Plataformas soportadas por la aplicación.</i> .....	85
<b>Figura 45</b> <i>Opción de tokens.</i> .....	85
<b>Figura 46</b> <i>Creación de tokens.</i> .....	86
<b>Figura 47</b> <i>Obtención del token.</i> .....	86
<b>Figura 48</b> <i>Inicio de la aplicación de thinger.io.</i> .....	87
<b>Figura 49</b> <i>Opción Qr scanner.</i> .....	88
<b>Figura 50</b> <i>Escaneo de QR.</i> .....	88
<b>Figura 51</b> <i>Dispositivo al que se accedió mediante el token.</i> .....	89
<b>Figura 52</b> <i>Datos de los sensores.</i> .....	89
<b>Figura 53</b> <i>Arquitectura IOT.</i> .....	91
<b>Figura 54</b> <i>Diagrama de flujo.</i> .....	92
<b>Figura 55</b> <i>Planificación de acciones en conjunto.</i> .....	93
<b>Figura 56</b> <i>Instalación del sensor 1.</i> .....	94
<b>Figura 57</b> <i>Instalación del sensor 2.</i> .....	95
<b>Figura 58</b> <i>Posicionamiento de sensores.</i> .....	96

<b>Figura 59</b> <i>Captura de datos de ejemplo.</i> .....	97
<b>Figura 60</b> <i>Directorio principal.</i> .....	98
<b>Figura 61</b> <i>Logs del servicio sensor.</i> .....	99
<b>Figura 62</b> <i>Logs del servicio satvh.</i> .....	100
<b>Figura 63</b> <i>Archivo JSON.</i> .....	100
<b>Figura 64</b> <i>Simulación de Alerta.</i> .....	101
<b>Figura 65</b> <i>Grafica de precipitaciones de Junio.</i> .....	102
<b>Figura 66</b> <i>Gráfica de promedio diario de nivel de agua de Junio.</i> .....	103
<b>Figura 67</b> <i>Promedio diario del nivel de agua del mes de Junio.</i> .....	105
<b>Figura 68</b> <i>Comparación de crecimiento de agua y precipitación.</i> .....	106

## INDICE DE TABLAS

<b>Tabla 1.</b> <i>Arquitectura LoRaWAN.</i> .....	20
<b>Tabla 2.</b> <i>Resumen regional de LoRaWAN.</i> .....	24
<b>Tabla 3.</b> <i>Configuración AU915-928.</i> .....	25
<b>Tabla 4.</b> <i>Requerimientos del sistema.</i> .....	27
<b>Tabla 5.</b> <i>Configuración de filtros de sensores.</i> .....	30
<b>Tabla 6</b> <i>Comparación de tecnologías de sensor.</i> .....	36
<b>Tabla 7</b> <i>Tabla comparativa de sensores de ultrasonido.</i> .....	38
<b>Tabla 8</b> <i>Comparación entre microcontroladores.</i> .....	40
<b>Tabla 9</b> <i>Comparación entre tecnologías inalámbricas.</i> .....	41
<b>Tabla 10</b> <i>Comparación entre módulos LORA.</i> .....	43
<b>Tabla 11</b> <i>Conexiones del Emisor.</i> .....	45

<b>Tabla 12</b> <i>Comparación de computadoras.</i> .....	53
<b>Tabla 13</b> <i>Comparación de plataformas IOT.</i> .....	56
<b>Tabla 14</b> <i>Conexiones del módulo receptor.</i> .....	58

## **Capítulo I: Introducción**

### **1.1 Tema**

Implementación de sistema de alerta temprana frente a inundaciones provocadas por el desborde del río blanco en la parroquia de Valle Hermoso.

### **1.2 Problema de investigación**

En Ecuador, los meses de enero a junio se consideran de alto riesgo debido a las fuertes lluvias que causan crecidas en los ríos y desbordamientos, agravados por el fenómeno del Niño, lo cual provoca inundaciones y ocasiona pérdidas materiales y humanas.

Valle Hermoso es una parroquia situada en la provincia de Santo Domingo de los Tsáchilas. Se encuentra atravesada por dos ríos, el río Blanco y el río Cristal. Además de ser utilizados para el riego de las plantas, estos ríos también cuentan con balnearios. Las principales actividades económicas de la parroquia son la agricultura y el turismo.

La población de Valle Hermoso es de aproximadamente 10,000 habitantes, según el Censo de Población, Densidad Poblacional y Superficie de Ecuador realizado en 2022 (Valle Hermoso - Censo de Población, Densidad Poblacional y Superficie de Ecuador | Dateas.Com, 2022). En los últimos años, la parroquia ha sufrido varias inundaciones debido a las crecidas de los ríos y al fenómeno del Niño. El 18 de marzo de 2023, se produjo un desborde que afectó a 75 familias, dejándolas damnificadas (El Valle Hermoso Quedó Destrozado Tras La Inundación, n.d.). Asimismo, el 17 de febrero de 2021, ocurrió otra crecida que dejó a 300 familias de la parroquia de Valle Hermoso incomunicadas (Las Lluvias Provocan Daños En La Zona Urbana y Rural de Santo Domingo - El Comercio, n.d.).

A raíz de este último suceso en el año 2023, es imprescindible contar con un sistema de alerta contra inundaciones que permita minimizar los daños materiales y, sobre todo, las pérdidas humanas.

### **1.3 Objetivos**

#### ***Objetivo General***

Implementar un sistema de alerta temprana (SAT) contra inundaciones, basándose en la arquitectura IoT, para mejorar el tiempo de respuesta ante futuras emergencias.

#### ***Objetivos Específicos***

- Consultar las bases de conocimiento necesarias para el diseño del sistema y los principales parámetros físico-hidrológicos a medir.
- Analizar la situación actual del sector, para establecer puntos estratégicos para la ubicación de los módulos de comunicación, así como los requerimientos necesarios para el funcionamiento del sistema.
- Desplegar el SAT para validar el funcionamiento del sistema, así como los tiempos adecuados de respuesta que permitan la evacuación oportuna del sector.

### **1.4 Alcance del proyecto**

Se plantea la implementación de un sistema de alerta temprana en la cuenca del río Blanco, este sistema constará de múltiples sensores de nivel de agua los cuales se comunicarán mediante un módulo LORA controlado por un esp32, a su vez se establece un módulo LORA configurado en modo de Gateway para recibir los datos y pasarlos a una placa de desarrollo de

hardware libre para su salida a internet. Todo este proceso se llevará a cabo siguiendo los principios de la arquitectura de Internet de las Cosas (IoT).

Para el desarrollo del marco de investigación se usará la metodología propuesta por el programa regional para el fortalecimiento de los servicios meteorológicos, hidrológicos, climáticos y el desarrollo (PRASDES), con énfasis en el enfoque mostrado por la Organización de Estados Americanos (OEA), el cual propone:

1. Investigación sobre el sector en donde se implementará el sistema de alerta temprana, para conocer los datos políticos, económicos y culturales de la parroquia.
2. Reconocimiento de la cuenca menor: En este apartado se realiza un mapa de la cuenca menor en la que se contempla: ríos y otros cuerpos de agua, partes altas y bajas, ubicación de comunidades, vías, casas y otros equipamientos comunitarios), también es en este apartado donde se localizan las zonas vulnerables.
3. Medición del nivel de los ríos: Se realiza una investigación y comparación de los instrumentos de medición, como limnómetros, para evaluar el que mejor se adapte a las necesidades del sistema.
4. Funcionamiento del sistema de alerta: Se consulta algoritmos que permitan un óptimo tratamiento de los datos que se pueden obtener del punto anterior para lograr obtener información útil de estos que permita dar un aviso certero a la población.
5. Evaluación de la situación, difusión de alertas y plan de respuesta: Mediante una investigación de la zona, se realiza una evaluación de la gestión de riesgos actual, obteniendo la información del Centro de operaciones de emergencias zonal (COE).

Previo a la implementación del sistema, se debe hacer un análisis exhaustivo de la zona para encontrar los puntos óptimos en los que se pueden colocar los sensores y obtener la mejor señal

posible para la comunicación de los módulos que se usarán. El análisis se llevará a cabo mediante mapas de cobertura de las empresas disponibles, mapas hidrográficos de la zona y evaluación del tiempo de respuesta.

Una vez que se tengan todos los datos necesarios, se procederá a la implementación del sistema en la cuenca menor que se haya identificado en el punto anterior y se analizarán los requerimientos de los sensores y módulos elegidos para obtener su mejor rendimiento, sin que estos puedan ser afectados a corto plazo por la intemperie.

### **1.5 Justificación**

El sector de Valle Hermoso se ha visto repetidamente afectado por inundaciones, que han causado importantes daños materiales y pérdidas de vidas humanas. La última inundación destruyó muros de contención y sistemas de alerta previamente instalados, dejando a la parroquia en un estado vulnerable.

Las inundaciones en Valle Hermoso suelen ocurrir repentinamente, y la mayoría se producen por la noche, cuando los residentes duermen. Esto, combinado con la presencia de ancianos y niños en la comunidad, ha creado la necesidad de una respuesta rápida y eficaz para proteger las vidas y la seguridad.(Víctor Manuel Quirola Maldonado, 2015)

Las previsiones meteorológicas para este año indican que tanto la estación invernal como el fenómeno de El Niño serán más intensos que en años anteriores. Así lo corroboran las empresas privadas que vigilan el río para el turismo de deportes acuáticos, que han señalado que en años anteriores aún se podían realizar este tipo de actividades en esta época del año(Thielen et al., 2023).

Dada la creciente amenaza de futuras inundaciones, es esencial proporcionar a la población información que le permita reaccionar a tiempo y buscar refugio antes de que se desencadene la tragedia. El dispositivo de alerta temprana propuesto será una herramienta vital en la gestión de

riesgos, proporcionando a los residentes de Valle Hermoso la capacidad de recibir notificaciones y alertas tempranas sobre la subida del nivel del río y otros indicadores clave de peligro. Esta información permitirá a la población tomar las medidas preventivas adecuadas, como la evacuación temprana y la salvaguarda de sus pertenencias, con el objetivo de salvaguardar su bienestar físico y minimizar los daños materiales.

Además de su función de alerta, se espera que este dispositivo se convierta también en una fuente fiable de información para la comunidad en general. Se proporcionarán datos actualizados sobre las condiciones hidrometeorológicas, lo que permitirá a los residentes mantenerse informados y tomar decisiones fundamentadas sobre sus actividades cotidianas, especialmente durante los picos de lluvias intensas.

La implementación de un dispositivo de alerta temprana en el sector de Valle Hermoso es una medida urgente y necesaria para hacer frente a la amenaza recurrente de inundaciones, al proporcionar información oportuna y precisa, se espera que esta iniciativa reduzca los riesgos asociados a los desastres naturales y proteja la vida y el bienestar de los habitantes de la parroquia. Además, contribuirá a la concientización de la comunidad sobre la importancia de estar preparados ante eventos extremos y fomentará una cultura de prevención en la zona afectada.

## **Capítulo II: Fundamentos teóricos**

### **2.1. Gestión de riesgos**

La gestión de riesgos es el proceso de identificar, analizar, evaluar y responder a los riesgos que pueden afectar a una organización. El objetivo de la gestión de riesgos es reducir la probabilidad de que ocurran eventos adversos y mitigar sus consecuencias (Rodríguez, 2008).

### ***2.1.1. Riesgo***

El riesgo es la posibilidad de que ocurra un evento adverso. Este evento puede causar daños o pérdidas, ya sean materiales, personales o morales. El riesgo se puede definir de dos maneras: probabilidad y consecuencia, o incertidumbre. Los riesgos pueden ser de naturaleza diversa, y pueden afectar a cualquier persona, organización o sistema (Rodríguez, 2008).

### ***2.1.2. Desastre natural***

Un desastre natural es un evento o fenómeno natural que causa graves daños o pérdidas a la vida humana, la propiedad y el medio ambiente. Los desastres naturales pueden ser causados por una variedad de factores, como terremotos, inundaciones, tsunamis, deslizamientos de tierra, erupciones volcánicas, sequías, incendios forestales y tormentas.

Los desastres naturales son una amenaza global que afecta a todos los países, independientemente de su nivel de desarrollo. Cada año, millones de personas son afectadas por desastres naturales, y miles de ellas pierden la vida.

La definición de desastre natural es compleja y puede variar según la organización o el contexto. Sin embargo, en general, un desastre natural se puede definir como un evento o fenómeno natural que:

- Causa daños o pérdidas significativas.
- Ocurre de forma repentina o inesperada.
- Supera la capacidad de respuesta de la comunidad afectada.

Los desastres naturales pueden tener un impacto devastador en las personas, las comunidades y los ecosistemas. Pueden causar la pérdida de vidas, la destrucción de la propiedad, la interrupción de los servicios esenciales y la degradación del medio ambiente.

La gestión de los desastres naturales es un desafío complejo que requiere la participación de todos los niveles de la sociedad. Las medidas de mitigación y preparación pueden ayudar a reducir el impacto de los desastres naturales, pero no pueden eliminarlos por completo. (Salud, 2000)

### **2.1.3. Inundaciones**

Una inundación se define como la ocupación temporal por parte del agua de terrenos que normalmente no están cubiertos por ella, como consecuencia de la aportación inusual y más o menos repentina de una cantidad de agua superior a la que puede drenar el propio cauce del río.

Las inundaciones pueden ser causadas por una variedad de factores, incluyendo:

- Lluvias torrenciales: Las lluvias torrenciales son las principales causas de las inundaciones en la mayoría de los casos.
- Deshielo: El deshielo de la nieve y el hielo puede provocar inundaciones, especialmente en zonas montañosas.
- Ruptura de presas o diques: La rotura de presas o diques puede liberar grandes cantidades de agua que pueden provocar inundaciones.
- Subidas de las mareas: Las subidas de las mareas pueden provocar inundaciones en zonas costeras.
- Huracanes y otros fenómenos meteorológicos extremos: Los huracanes y otros fenómenos meteorológicos extremos pueden provocar inundaciones por lluvias torrenciales, mareas altas y otros factores.

### **2.1.4. Sistema de alerta temprana (SAT)**

La gestión efectiva de situaciones de emergencia y desastres naturales requiere la implementación de sistemas de alerta temprana (SAT) que integren políticas, estrategias y acciones específicas.

Según la Estrategia Internacional para la Reducción de Desastres (EIRD), un sistema de alerta

temprana abarca necesariamente cinco elementos fundamentales: conocimiento del riesgo, observancia y monitoreo, análisis y pronóstico de las amenazas, comunicación de alertas y avisos, y capacidades locales para responder.

En primer lugar, el conocimiento del riesgo es fundamental para comprender la naturaleza y la magnitud de las amenazas. Este componente implica una evaluación exhaustiva de los posibles peligros que podrían afectar a una comunidad. Sin una comprensión profunda de estos riesgos, la efectividad de cualquier sistema de alerta temprana se ve comprometida.

La observancia y monitoreo constante constituyen el segundo elemento esencial. La vigilancia continua de las condiciones que podrían desencadenar eventos peligrosos es crucial para anticipar y responder de manera efectiva. La tecnología moderna desempeña un papel vital en este aspecto, permitiendo la recopilación y el análisis de datos en tiempo real para identificar cambios significativos en las condiciones ambientales.

El análisis y pronóstico de las amenazas conforman el tercer componente. Evaluar de manera precisa y anticipada la posible evolución de las amenazas permite una preparación más efectiva y la toma de decisiones informadas. La combinación de modelos científicos y datos históricos contribuye a prever con mayor precisión la ocurrencia de eventos peligrosos.

La comunicación o difusión de alertas y avisos representa el cuarto pilar esencial. La efectividad de un SAT depende en gran medida de la capacidad para transmitir información crítica de manera clara y oportuna a la población afectada. La tecnología de comunicación, desde sistemas de mensajes de texto hasta redes sociales, desempeña un papel crucial en esta fase, permitiendo una respuesta rápida y coordinada.

El quinto y último elemento, las capacidades locales para responder frente a la alerta recibida, destaca la importancia de empoderar a las comunidades para actuar rápidamente. La preparación

y la capacidad de respuesta a nivel local son fundamentales para minimizar los daños y las pérdidas potenciales. La participación activa de la comunidad, la formación y los simulacros son aspectos clave para fortalecer estas capacidades.

Es fundamental reconocer que estos cinco elementos están interconectados y su integración es esencial para el éxito del sistema. Una debilidad en cualquiera de ellos puede comprometer la eficacia general del SAT. La expresión "sistema de alerta de principio a fin" subraya la necesidad de abordar todas las etapas, desde la detección de una amenaza hasta la respuesta comunitaria coordinada.

La sostenibilidad operativa a largo plazo de un SAT requiere un compromiso político y una capacidad institucional duradera. La concientización pública y la apreciación de los beneficios de un SAT efectivo son pilares clave para garantizar el apoyo continuo. Además, es esencial comprender que un SAT va más allá de ser simplemente un instrumento de medición o comunicación; debe ser visualizado como un sistema de información diseñado para facilitar la toma de decisiones y capacitar a sectores vulnerables y grupos sociales (Americanos O. d., 2010).

#### ***2.1.5. Componentes de un sistema de alerta temprana ante inundaciones (SATI)***

El Manual para el Diseño e Implementación de un Sistema de Alerta Temprana de Inundaciones en Cuencas Menores, OEA 2001, propone cinco pasos fundamentales para el diseño y operación efectiva de un Sistema de Alerta Temprana (SATI).

El primer paso, la organización comunitaria, destaca la importancia de la participación activa de la comunidad en la gestión del riesgo. La conciencia colectiva y la colaboración son pilares esenciales para establecer una red sólida de comunicación y coordinación ante situaciones de emergencia. Además, la exploración de la cuenca menor, segundo paso propuesto, subraya la

necesidad de comprender a fondo las características geográficas y morfológicas de la región, permitiendo una evaluación precisa del riesgo de inundación.

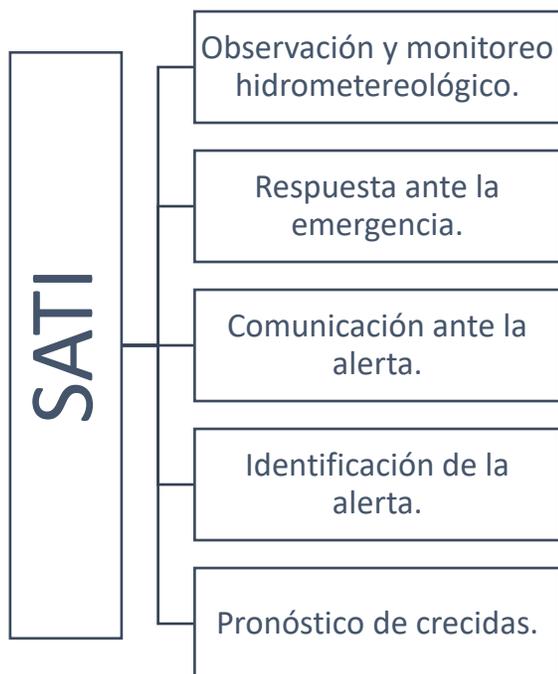
La medición de datos de lluvia y niveles de agua en los ríos, tercer paso, representa la base técnica del SATI. La instrumentación y el monitoreo hidrometeorológico son esenciales para recopilar información en tiempo real, facilitando la toma de decisiones informadas. Esta etapa técnica se complementa con la operación del sistema de alerta, el cuarto paso, que implica la aplicación efectiva de la información recopilada para activar alertas tempranas y ejecutar planes de emergencia.

La implementación del SATI se divide en dos fases: una técnica y otra social. La fase técnica aborda la instrumentación, medición hidrometeorológica y pronóstico, destacando la importancia de la tecnología en la gestión del riesgo. Por otro lado, la fase social resalta la necesidad de una organización comunitaria sólida para garantizar la efectividad del sistema. Ambas fases son interdependientes, y su integración adecuada es esencial para la operación eficiente del SATI.

El Manual aborda detalladamente estos aspectos en tres módulos. El primero se centra en la selección y utilización de equipos de observación, el segundo en características morfológicas y umbrales de alerta, y el tercero en la identificación y comunicación de la alerta. Estos módulos proporcionan un enfoque completo que abarca tanto la dimensión técnica como la social, permitiendo una implementación equilibrada y efectiva de los SATI (Americanos O. d., 2010).

**Figura 1.**

*Sistema Comunitario de Alerta Temprana ante Inundaciones.*



Fuente: (Workgroup, 2015)

## **2.2. Situación actual de la parroquia de Valle Hermoso**

Valle Hermoso es una parroquia rural del cantón Santo Domingo, perteneciente a la provincia Santo Domingo De Los Tsáchilas en Ecuador. Se encuentra ubicada a 25 kilómetros de la ciudad de Santo Domingo, en la vía a Esmeraldas.

La parroquia tiene una población de aproximadamente 10.000 habitantes, que se dedican principalmente a la agricultura, la ganadería y el turismo.

Valle Hermoso es un lugar conocido por su belleza natural. La parroquia está rodeada de montañas, ríos y cascadas. El río Blanco, que atraviesa la parroquia, es un destino popular para la pesca y el rafting.

La parroquia también cuenta con una rica cultura tsáchila. Los tsáchilas son un pueblo indígena que vive en la región desde hace siglos. La parroquia cuenta con un centro cultural tsáchila, donde se puede aprender sobre la cultura y las tradiciones de este pueblo.

A continuación, se presentan algunos datos interesantes sobre la parroquia de Valle Hermoso:

- El nombre de la parroquia proviene de los paisajes de la zona, específicamente el río Blanco y sus selvas vírgenes.
- La parroquia fue fundada en 1950.
- La parroquia tiene un clima cálido húmedo, con una temperatura promedio de 25 grados centígrados.

La parroquia cuenta con una variedad de flora y fauna, incluyendo especies endémicas de la región.

Algunos de los atractivos turísticos de la parroquia de Valle Hermoso incluyen:

- El río Blanco
- La cascada de la Virgen
- El centro cultural tsáchila
- El mirador de la Selva
- El sendero de la cascada

La parroquia de Valle Hermoso es un destino turístico popular que ofrece una variedad de actividades y atracciones para los visitantes. (Chávez, Plan de Desarrollo y Ordenamiento Territorial Parroquia Valle Hermoso 2020-2023 )

### **2.3. Internet de las cosas**

El concepto del Internet de las Cosas (IoT) encapsula la convergencia de dispositivos cotidianos con la red digital, estableciendo una red colectiva que redefine la interacción entre el mundo físico y el virtual. En su esencia, el IoT representa un paradigma tecnológico donde objetos

aparentemente inanimados adquieren inteligencia a través de la incorporación de sensores y procesadores, permitiéndoles recopilar datos y responder de manera autónoma a través de la conectividad a la nube.

Desde sus incipientes desarrollos en la década de 1990 hasta su expansión masiva en la actualidad, el IoT ha experimentado una evolución marcada por avances tecnológicos clave. En sus primeras fases, la integración de sensores y procesadores en objetos cotidianos se vio obstaculizada por el tamaño voluminoso de los chips de computadora. No obstante, con la llegada de chips de baja potencia y la mejora en las capacidades de telecomunicaciones, especialmente en términos de ancho de banda, se allanó el camino para la integración masiva de dispositivos de IoT en nuestra vida diaria.

La miniaturización de los chips, acompañada por una mayor eficiencia energética, ha sido esencial para esta proliferación de dispositivos inteligentes. Esta evolución ha transformado objetos comunes, desde cepillos de dientes hasta aspiradoras y coches, en entidades capaces de recopilar datos mediante sensores y de responder de manera inteligente a través de la conexión a la nube. Este fenómeno representa una evolución significativa en la relación entre el ser humano y la tecnología, donde la automatización y la toma de decisiones autónomas son cada vez más prominentes en la vida cotidiana.

La conectividad del IoT no se limita a dispositivos individuales; se extiende a la convergencia de servicios de voz en objetos anteriormente desprovistos de tales capacidades. La integración de servicios como Alexa en interruptores de luz ejemplifica cómo los chips de baja potencia, con menos de 1 MB de RAM integrada, permiten la expansión de la red de dispositivos de IoT incluso en entornos aparentemente simples.

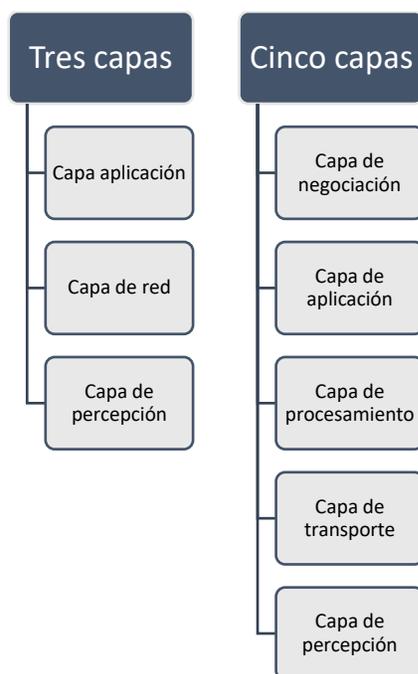
En un plano más amplio, el IoT ha dado lugar a un sector dedicado a llenar nuestros hogares, empresas y oficinas con dispositivos de IoT. Este impulso hacia la conectividad ubicua contribuye a la creación de entornos inteligentes y eficientes, abarcando desde la gestión de la energía hasta aplicaciones prácticas en la salud pública. (Amazon, 2023)

### 2.3.1. Arquitectura

La arquitectura de Internet de las cosas (IoT, por sus siglas en inglés) se refiere al diseño estructurado y organizado de los componentes que componen un sistema de IoT.

#### Figura 2.

*Arquitecturas IOT.*



Fuente: (Workgroup, 2015)

#### Arquitectura de Tres Capas

La arquitectura de tres capas, establece las bases conceptuales del IoT. En esta estructura, se identifican tres capas fundamentales: percepción, red y aplicación.

- La capa de percepción constituye la base física del sistema, empleando sensores para recopilar información sobre el entorno. Esta capa detecta parámetros físicos y reconoce objetos inteligentes en el entorno, proporcionando la entrada crítica para el sistema.
- La capa de red se encarga de la conectividad, facilitando la interacción entre dispositivos inteligentes, nodos de red y servidores. Además de posibilitar la transmisión de datos, esta capa también procesa información sensorial, contribuyendo a la toma de decisiones del sistema.
- La capa de aplicación representa la interfaz directa con el usuario, entregando servicios específicos y definiendo los escenarios de implementación del IoT, como hogares inteligentes, ciudades inteligentes y aplicaciones de salud inteligente.

### **Arquitectura de Cinco Capas**

La arquitectura de cinco capas, una evolución de la estructura de tres capas, incorpora las capas de procesamiento y negocios. Esta ampliación busca abordar las complejidades y los detalles más finos inherentes a la investigación en IoT.

- La capa de transporte facilita el flujo de datos desde la capa de percepción hasta la capa de procesamiento y viceversa, utilizando diversas redes como inalámbricas, 3G, LAN, Bluetooth, RFID y NFC.
- La capa de procesamiento, también conocida como capa intermedia, desempeña un papel crucial al almacenar, analizar y procesar grandes volúmenes de datos generados por la capa de transporte. Utiliza tecnologías avanzadas como bases de datos, computación en la nube y módulos

de procesamiento de grandes datos para ofrecer una gama diversa de servicios a las capas inferiores.

- La capa de negocios gestiona la totalidad del sistema de IoT, incluyendo aplicaciones, modelos de negocios y de beneficios, y la privacidad de los usuarios.

Ambas arquitecturas comparten similitudes en las capas de percepción y aplicación, pero difieren en la incorporación de capas adicionales en la estructura de cinco capas. La adición de las capas de transporte y procesamiento mejora la eficiencia y la capacidad de procesamiento del sistema, lo que es vital para la gestión de grandes volúmenes de datos generados por dispositivos IoT (Sethi, 2017).

### ***2.3.2. Protocolos de capa aplicación***

La capa de aplicación, tradicionalmente basada en HTTP en el Internet convencional, presenta desafíos significativos en entornos IoT con recursos limitados. La naturaleza detallada y verbosa de HTTP resulta ineficiente en términos de sobrecarga de procesamiento, especialmente en dispositivos con restricciones de recursos. Aquí es donde entra en juego la necesidad de protocolos alternativos.

#### **Message Queue Telemetry Transport (MQTT)**

MQTT, desarrollado por IBM, se destaca como un protocolo de publicación/suscripción eficiente que opera sobre TCP. Aunque diseñado inicialmente como un protocolo cliente/servidor, su estructura ligera lo hace adecuado para entornos IoT. Los clientes, actuando como editores o suscriptores, se conectan a un servidor intermediario a través de TCP. Sin embargo, la limitación de MQTT radica en su dependencia de TCP, lo que puede no ser ideal para todos los escenarios de aplicaciones IoT. A pesar de esta limitación, MQTT-S/MQTT-SN ofrece una extensión diseñada específicamente para dispositivos de bajo consumo y bajo costo. Esta extensión conserva

la eficiencia de MQTT al reemplazar los nombres de temas con identificadores de temas, reduciendo así la sobrecarga de transmisión y permitiendo una operación más eficiente en redes de sensores inalámbricos (WSNs).

### **Internet of Things Message Protocol (IOTMP)**

IOTMP (Protocolo de Mensajes de Internet de las Cosas) es un protocolo ligero, orientado a la solicitud-respuesta y al streaming, diseñado por ingenieros del MIT para su uso en redes de IoT. Es un protocolo de mensajería que permite a los dispositivos comunicarse con un servidor central o plataforma, que actúa como el concentrador de la red.

IOTMP es un protocolo cliente-servidor, donde los dispositivos que exponen recursos (es decir, medidas de detección o controles de actuación) se llaman "clientes" y el servidor central se llama el "servidor". Los clientes se conectan al servidor y exponen sus recursos, donde el servidor los pone a disposición a través de una API REST o websockets. Además, los clientes pueden interactuar con recursos del lado del servidor, es decir, para leer configuraciones, interactuar con servicios de terceros (API), activar alarmas, suscribirse a eventos, etc.

Una de las principales ventajas de IOTMP es que está diseñado para ser ligero y eficiente, lo cual es importante en redes de IoT donde los dispositivos pueden tener recursos limitados. También tiene un pequeño tamaño y es fácil de implementar, lo que lo hace adecuado para dispositivos con recursos limitados como sensores o microcontroladores. Actualmente, IOTMP está disponible para diferentes lenguajes de programación, incluyendo Arduino, C++, NodeJs, Python y Javascript para la web.

#### ***2.3.3. Tecnologías de red***

- **6LoWPAN (IPv6 sobre Redes Personales de Área Local de Bajo Consumo):** Es un estándar que facilita la conexión de dispositivos con recursos limitados, como sensores y

actuadores, a través de redes IP de baja potencia. Permite la comunicación eficiente en redes de área local.

- **Bluetooth Low Energy (BLE):** Es una tecnología de comunicación inalámbrica diseñada para dispositivos con bajos requisitos de energía. Se utiliza comúnmente en aplicaciones de conexión entre dispositivos personales, como pulseras de fitness y dispositivos de monitoreo de salud.
- **Near Field Communication (NFC):** Es una tecnología de comunicación inalámbrica de corto alcance que permite la transferencia de datos entre dispositivos cuando están ubicados a poca distancia uno del otro. Se utiliza comúnmente en transacciones de pago y emparejamiento de dispositivos.
- **Wireless Sensor Networks (WSN):** Son redes compuestas por nodos de sensores inalámbricos que recopilan datos del entorno. Se utilizan en diversas aplicaciones, como monitoreo ambiental, salud y control industrial.
- **LoRaWAN (Long Range Wide Area Network):** Es un protocolo de comunicación de área amplia y baja potencia diseñado para la transmisión de datos a larga distancia con bajo consumo de energía. Se utiliza en aplicaciones como el seguimiento de activos y la monitorización remota.
- **Sigfox:** Es una tecnología de comunicación inalámbrica de baja potencia que utiliza bandas de frecuencia no licenciadas para enviar pequeñas cantidades de datos a larga distancia. Se emplea en aplicaciones como la monitorización de dispositivos remotos y el seguimiento de activos.

- **Low-Power WiFi (WiFi de Bajo Consumo):** Se refiere a tecnologías WiFi diseñadas para dispositivos IoT con requisitos de energía reducidos, permitiendo la conexión a Internet con un consumo eficiente de energía.
- **Zigbee:** Es un protocolo de comunicación inalámbrica diseñado para aplicaciones de baja potencia y baja tasa de transmisión, como redes de sensores y control de dispositivos domésticos.
- **Bluetooth Low Energy (BLE):** Es una versión de Bluetooth diseñada para reducir el consumo de energía, siendo ideal para dispositivos que requieren una larga duración de la batería, como dispositivos médicos y wearables.
- **LowPower Wide-Area Networks (LPWAN):** Se refiere a tecnologías de comunicación de área amplia y baja potencia que permiten la transmisión de datos a larga distancia con eficiencia energética, como Narrowband IoT, Sigfox y LoRaWAN.

#### **2.4. LoRaWAN (Long Range Wide Area Network)**

LoRaWAN™ establece el protocolo y la arquitectura para la red, mientras que la capa física LoRa® permite la comunicación de largo alcance. La duración de la batería, la capacidad de la red, la calidad del servicio, la seguridad y la variedad de aplicaciones dependen en gran medida de este protocolo y arquitectura. La tecnología LoRaWAN™ es esencial para habilitar conexiones eficientes y de largo alcance, siendo crucial para sistemas IoT donde la vida útil de la batería y la eficiencia de la red son críticas. (Workgroup, 2015)

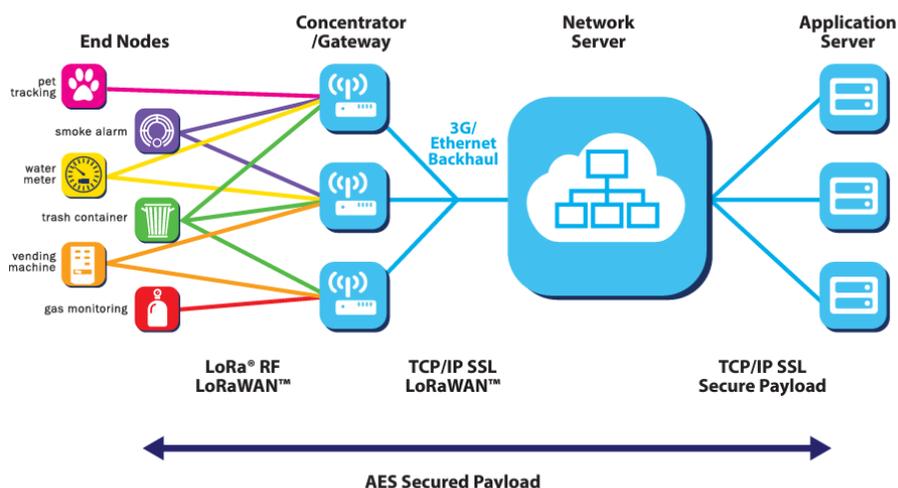
**Tabla 1.***Arquitectura LoRaWAN.*

APLICACIÓN				
LoRa® MAC				
Opciones MAC				
Clase A		Clase B		Clase C
Modulación LoRa®				
Banda ISM regional				
EU 868	EU 433	US 915	AS 430	-

Fuente: (Workgroup, 2015)

**2.4.1. Arquitectura de red**

En redes existentes, la arquitectura de malla implica que los nodos individuales retransmitan información entre sí para ampliar el alcance de la comunicación. Aunque esto extiende el rango, añade complejidad, reduce la capacidad y disminuye la duración de la batería al procesar información potencialmente irrelevante. La alternativa de arquitectura estelar de largo alcance resulta más eficiente, especialmente cuando se busca preservar la duración de la batería en conexiones de largo alcance, evitando los desafíos asociados con las redes de malla. Este enfoque simplificado se destaca en entornos donde la conectividad de larga distancia es alcanzable y crucial (Workgroup, 2015).

**Figura 3.***Arquitectura de Red.*

Fuente: (Workgroup, 2015)

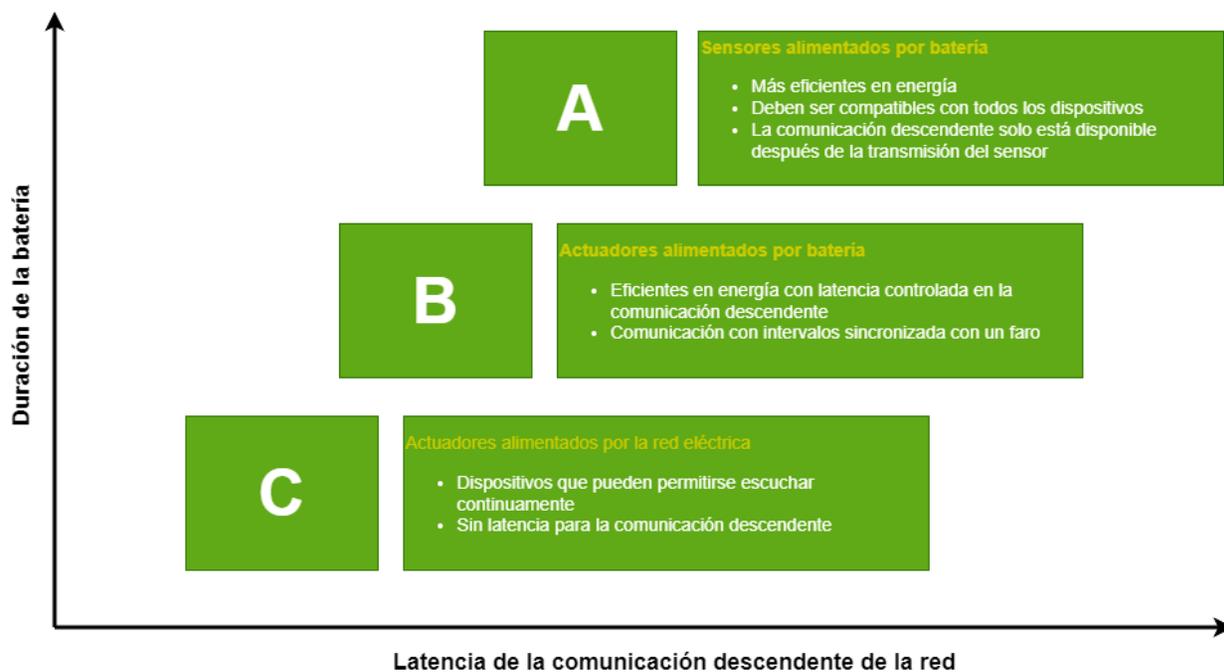
En una red LoRaWAN™, los nodos no están vinculados a una puerta de enlace específica, sino que los datos de un nodo son recibidos por múltiples puertas de enlace. Estas puertas de enlace envían los paquetes al servidor de red basado en la nube a través de diversos enlaces de retorno. La complejidad y la inteligencia se encuentran en el servidor de red, que gestiona la red, filtra datos redundantes, realiza controles de seguridad y adapta la velocidad de datos. Esta arquitectura es especialmente beneficiosa para aplicaciones de seguimiento de activos en IoT, ya que no requiere transferencia entre puertas de enlace, facilitando la movilidad del nodo (Workgroup, 2015).

### 2.4.2. Tipos de dispositivos

LoRaWAN™ adapta diferentes clases de dispositivos para satisfacer diversas aplicaciones y sus requisitos específicos. Estas clases buscan optimizar perfiles de aplicaciones finales al equilibrar la latencia en la comunicación descendente de la red con la duración de la batería. En aplicaciones que involucran control o actuadores, donde la latencia en la comunicación descendente es crítica, se prioriza la eficiencia en este aspecto. Este enfoque permite ajustar la red LoRaWAN™ para atender de manera óptima una amplia gama de aplicaciones, considerando las necesidades individuales de cada dispositivo final.

#### Figura 4.

*Clases de dispositivos.*



Fuente: (Workgroup, 2015)

La Clase A de dispositivos finales permite comunicaciones bidireccionales con transmisiones ascendentes seguidas de dos breves ventanas de recepción descendente. La programación de la ranura de transmisión se basa en las necesidades de comunicación del dispositivo, con variaciones mínimas basadas en un protocolo tipo ALOHA. Esta operación de Clase A es ideal para aplicaciones que solo requieren comunicación descendente poco después de la transmisión ascendente.

La Clase B, además de las ventanas de recepción aleatorias de la Clase A, abre ventanas adicionales en momentos programados. Para abrir la ventana de recepción en el tiempo programado, el dispositivo recibe una baliza sincronizada desde la puerta de enlace. Esto permite al servidor saber cuándo el dispositivo está listo para recibir, optimizando así las comunicaciones.

La Clase C presenta dispositivos finales con ventanas de recepción casi continuamente abiertas, cerrándose solo durante la transmisión. Este enfoque permite una comunicación bidireccional más constante, pero con mayor consumo de energía en comparación con las Clases A y B. Las ventanas de recepción máximas ofrecen flexibilidad, ya que están abiertas la mayor parte del tiempo.

Las clases A, B y C proporcionan opciones para diversas necesidades de comunicación y eficiencia energética. La elección entre ellas dependerá de los requisitos específicos de la aplicación, como la frecuencia de las comunicaciones descendentes, la sincronización y el consumo de energía. Estos enfoques ofrecen soluciones adaptadas a diferentes escenarios, permitiendo una implementación más eficiente y efectiva de las redes de comunicación (Workgroup, 2015).

#### ***2.4.3. Resumen regional de LoRaWAN™***

La Alianza LoRa, a través de su documento de Aviso y Divulgación “RP002-1.0.2 LoRaWAN® Regional Parameters”, informa a los usuarios de varios países sobre el uso correcto de la tecnología

para garantizar su compatibilidad con las regulaciones locales. (Alliance, 2020) A continuación, se presenta los primeros datos de la tabla de referencia cruzada para facilitar esta compatibilidad:

**Tabla 2.**

*Resumen regional de LoRaWAN*

<b>País (Código alfa-2)</b>	<b>Bandas/Canales</b>	<b>Plan de canal</b>
Islas Aland (AX)	433.05-434.79 MHz, 863-870 MHz	EU433, EU863-870
Albania (AL)	433.05-434.79 MHz, 863-873 MHz, 915-918 MHz	EU433, EU863-870, AS923-3
Samoa Americana (AS)	902-928 MHz	US902-9281
Andorra (AD)	433.05-434.79 MHz, 863-870 MHz	EU433, EU863-870
Angola (AO)	915-928 MHz <sup>2</sup>	AU915-9283
Argentina (AR)	915-928 MHz <sup>2</sup>	AU915-928
Armenia (AM)	433.05-434.79 MHz, 863-870 MHz	EU433, EU863-870
Australia (AU)	915-928 MHz	AS923-1, AU915-928
Austria (AT)	433.05-434.79 MHz, 863-870 MHz	EU433, EU863-870
Azerbaiyán (AZ)	868-868.6 MHz, 868.7-869.2 MHz	-
Bahamas (BS)	902-928 MHz	US902-928
Bahrein (BH)	433-434 MHz, 863-870MHz	EU433, EU863-870
Bangladesh (BD)	433.05-434.79 MHz, 866-868 MHz	EU433, -
Barbados (BB)	902-928 MHz	US902-928
Bielorrusia (BY)	863-870 MHz	EU863-870
Bélgica (BE)	433.05-434.79 MHz, 863-870 MHz	EU433, EU863-870
Belice (BZ)	902-928 MHz	US902-928
Benín (BJ)	921-928 MHz	AU915-928
Bermudas (BM)	915-928 MHz	US902-928
Bolivia (BO)	921-928 MHz	AU915-928
Bosnia y Herzegovina (BA)	433.05-434.79 MHz	EU433
Botswana (BW)	863-870 MHz	EU863-870
Brasil (BR)	902-928 MHz	US902-928
Ecuador (EC)	902 - 928 MHz	AU915-928

Fuente: (Alliance, 2020)

#### 2.4.4. LoRaWAN™ para Ecuador AU915-928

Las recomendaciones de la LoRaWAN Alliance se basan en el Plan Nacional para el Espectro Radioeléctrico, específicamente en el capítulo "Notas al Cuadro Nacional de Atribución de Bandas de Frecuencias". Según la nota número 5.150, la banda de 915 MHz está designada para aplicaciones industriales, científicas y médicas (ICM). Esto asegura que la tecnología LoRaWAN opere de manera compatible y conforme a las regulaciones establecidas para el uso del espectro radioeléctrico en aplicaciones específicas. (Alliance, 2020)

Para la configuración en AU915-928 se hace referencia a la siguiente tabla.

**Tabla 3.**

*Configuración AU915-928.*

Data Rate	Configuration	Indicative physical bit rate [bit/sec]
0	LoRa: SF12/125 kHz	250
1	LoRa: SF11/125 kHz	440
2	LoRa: SF10/125 kHz	980
3	LoRa: SF9/125 kHz	1760
4	LoRa: SF8/125 kHz	3125
5	LoRa: SF7/125 kHz	5470
6	LoRa: SF8/500 kHz	12500
7	LR-FHSS CR1/3: 1.523 MHz OCW	162
8	LoRa: SF12/500 kHz	980
9	LoRa: SF11/500 kHz	1760
10	LoRa: SF10/500 kHz	3900
11	LoRa: SF9/500 kHz	7000
12	LoRa: SF8/500 kHz	12500
13	LoRa: SF7/500 kHz	21900

Fuente: (Alliance, 2020)

### **Capítulo III: Materiales y Métodos**

Este capítulo se centrará en el manual para el diseño e implementación de un sistema de alerta temprana de inundaciones en cuencas menores, proporcionado por la Organización de los Estados Americanos (OEA) en colaboración con el gobierno de Irlanda. Este manual consta de nueve requerimientos fundamentales que se detallan a continuación.

#### **1. Organización Comunitaria:**

- El sistema debe ser accesible y fácil de usar para que la comunidad pueda operarlo y mantenerlo.
- La interfaz debe ser intuitiva, permitiendo a los miembros de la comunidad designar un comité responsable del mantenimiento y operación del sistema.
- Capacitación y recursos deben estar disponibles para asegurar que los usuarios puedan gestionar el sistema de manera efectiva.

#### **2. Reconocimiento de la Cuenca Menor:**

- Se debe desarrollar un mapa detallado que incluya las zonas vulnerables, ríos, casas, otros cuerpos de agua, y diferencias de elevación (zonas altas y bajas).
- El mapa debe indicar claramente las ubicaciones óptimas para la instalación de los sensores, asegurando la cobertura de todas las áreas críticas.

#### **3. Medición de las Lluvias y Nivel de los Ríos:**

- Los sensores ultrasónicos y de lluvia deben proporcionar datos precisos y en tiempo real sobre la precipitación y el nivel de los ríos.
- Estos datos son esenciales para predecir inundaciones y evaluar los riesgos actuales.

#### 4. **Funcionamiento del Sistema:**

- El sistema debe ser capaz de medir con precisión las variables físicas (nivel del agua y precipitación).
- Además, debe interpretar estos datos, hacer pronósticos y transmitir alertas oportunas.
- La integración con una plataforma en la nube para procesamiento y análisis de datos es crucial para la operatividad continua y efectiva.

#### 5. **Evaluación de la Situación, Difusión de Alerta y Plan de Respuesta:**

- Un plan de emergencia debe ser desarrollado y acordado por la comunidad, detallando las acciones a seguir en caso de alerta.
- El sistema debe ser capaz de evaluar la situación actual y activar protocolos de respuesta mediante la difusión de alertas a través de aplicaciones como Telegram.
- La difusión de alertas debe ser rápida y efectiva, asegurando que toda la comunidad esté informada y preparada para actuar. (Americanos S. G., 2001)

A continuación, se presenta una tabla con los requerimientos del sistema siguiendo los requerimientos tanto de los stakeholders como de la metodología planteada.

**Tabla 4.**

*Requerimientos del sistema.*

<b>Componente</b>	<b>Requerimiento</b>	<b>Prioridad</b>
Sensores y Medición		
Tecnología de Sensores	Instalación de sensores ultrasónicos para medir el nivel de los ríos	Alta
	Implementación de monitoreo de lluvia	Alta
	Calibración y mantenimiento regular de los sensores	Media

Procesamiento y Análisis de Datos		
Medición de Variables Físicas	Medición precisa del nivel del agua y precipitación	Alta
Algoritmos de Análisis	Implementación de algoritmos para interpretar datos, realizar pronósticos y evaluar riesgos	Alta
Plataforma en la Nube	Integración con una plataforma en la nube para procesamiento y almacenamiento continuo de datos	Alta
	Implementación de endpoints para alertas	Alta
Interfaz de Usuario		
Interfaz Intuitiva	Implementación de una interfaz de usuario amigable y accesible para la comunidad	Alta
	Diseño adaptable para acceso desde diversos dispositivos (smartphones, tablets, PC)	Alta
Evaluación y Difusión de Alertas		
Evaluación en Tiempo Real	Evaluación de la situación actual en tiempo real para activar protocolos de respuesta	Alta
Difusión de Alertas	Utilización de aplicaciones como Telegram para la difusión rápida de alertas	Alta
	Envío de alertas mediante SMS y correos electrónicos como métodos alternativos	Media
Mantenimiento y Operación		
Designación de Comité	Facilitar la designación de un comité responsable de la operación y mantenimiento del sistema	Alta
Capacitación Técnica	Implementación de programas de capacitación para el manejo y mantenimiento del sistema	Alta
Requerimientos Físicos		
Resistencia a Condiciones Climáticas	Sensores y equipos deben ser resistentes a condiciones climáticas adversas (lluvia intensa, viento, temperaturas extremas)	Alta
Instalación Segura	Ubicación segura de los sensores para prevenir daños y asegurar el funcionamiento continuo	Alta
Alimentación Eléctrica	Fuente de energía confiable para los sensores y equipos, incluyendo opciones de energía solar o baterías de respaldo	Alta

Protección Contra Inundaciones	Protección física de los equipos y sensores para evitar daños durante inundaciones	Alta
Mantenimiento Accesible	Diseño de los equipos para facilitar el acceso y mantenimiento regular sin necesidad de herramientas especializadas	Media
Durabilidad	Equipos diseñados para tener una larga vida útil y resistir el desgaste por el uso continuo	Media
Desempeño del Sistema		
Escalabilidad	Capacidad de escalar el sistema para cubrir más áreas o aumentar la cantidad de sensores	Media
Latencia Baja	Asegurar una baja latencia en la transmisión de datos desde los sensores hasta la plataforma de análisis	Alta
Integración y Compatibilidad		
Interoperabilidad	Compatibilidad con otros sistemas y plataformas de monitoreo de desastres naturales	Media
API Abierta	Implementación de una API abierta para facilitar la integración con aplicaciones y servicios externos	Media
Monitoreo y Soporte Técnico		
Monitoreo 24/7	Implementación de un sistema de monitoreo 24/7 para asegurar el funcionamiento continuo	Alta

## 2.1. Organización Comunitaria

El funcionamiento continuo del sistema requiere de ciertos conocimientos por parte de la comunidad, por lo que es imperativo formar un comité voluntario compuesto por personas interesadas. En este caso, el comité estará liderado por ciudadanos directamente afectados, con el propietario de una empresa de deportes acuáticos a la cabeza. Este individuo está actualmente a cargo del monitoreo del río.

## 2.2. Reconocimiento de la Cuenca Menor

Para lograr un análisis mejor de la situación geográfica se debe realizar un mapa de la cuenca menor la cual es una sección de la cuenca de un río, en la que se identifican, cuerpos de agua, casas, plantaciones y zonas en riesgo.

### 2.2.1. Identificación de cuerpos de agua

Lo primordial en la identificación de la cuenca menor es localizar todos los cuerpos de agua, dado que estos constituirán los elementos a monitorear con mayor atención. Dentro de los cuerpos de agua identificados, se encuentran diversos riachuelos y tres ríos específicos que serán objeto de estudio: el río Blanco, el río Toachi y el río Meme.

Para llevar a cabo la identificación de estos cuerpos de agua, se emplea el software Google Earth Engine, una herramienta poderosa para el análisis geoespacial. Se accede al repositorio del satélite Sentinel S1, el cual proporciona datos satelitales de alta resolución. Utilizando una combinación de filtros, se seleccionan los sensores B12, B11 y B4, los cuales corresponden a longitudes de onda específicas, según se detalla en la siguiente tabla. Este proceso permite discriminar con precisión los cuerpos de agua dentro de la cuenca, facilitando así su posterior monitoreo y análisis. (Argote, 2023)

**Tabla 5.**

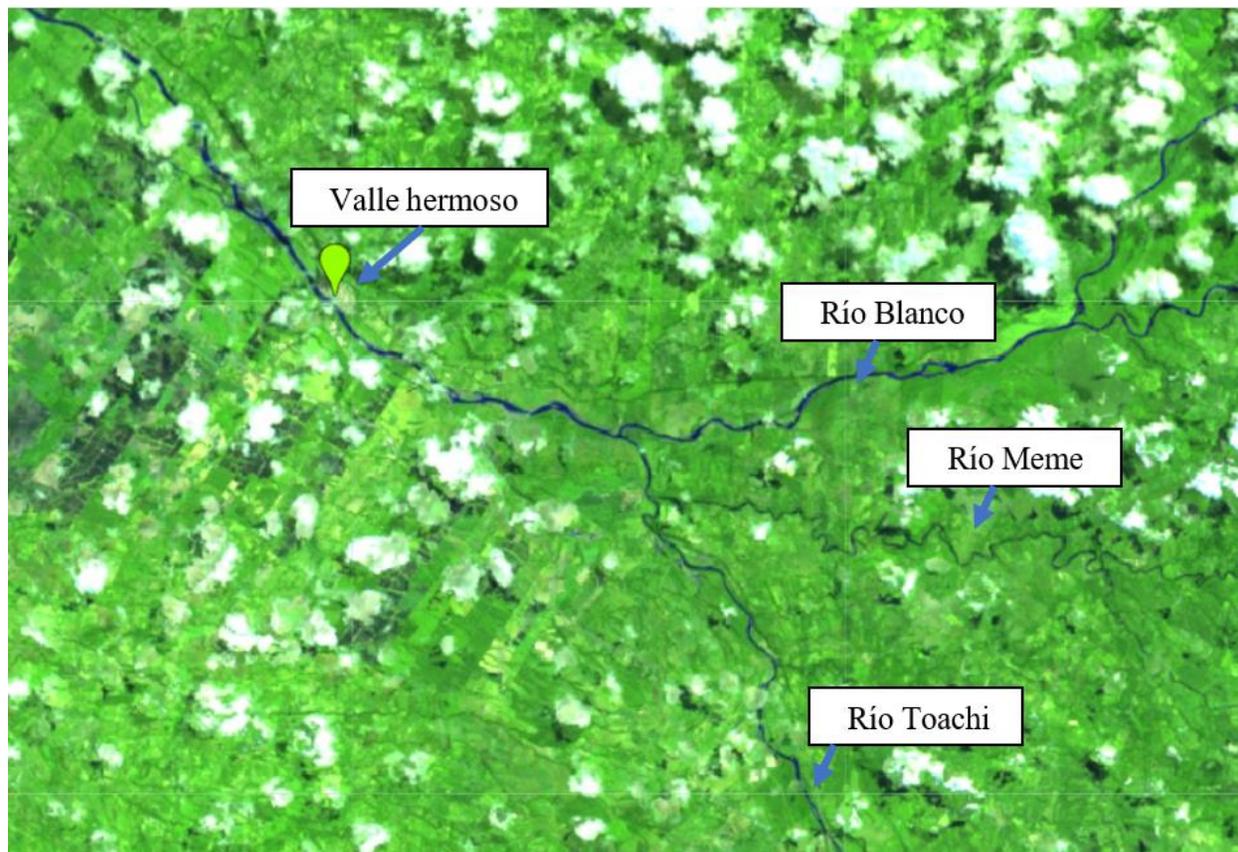
*Configuración de filtros de sensores.*

Nombre	Escala	Tamaño del pixel	Longitud de onda	Descripción
<b>B4</b>	0.0001	10 metros	664.5nm (S2A) / 665nm (S2B)	Rojo
<b>B11</b>	0.0001	20 metros	1613.7nm (S2A) / 1610.4nm (S2B)	SWIR 1
<b>B12</b>	0.0001	20 metros	2202.4nm (S2A) / 2185.7nm (S2B)	SWIR 2

Como resultado, se obtienen colores brillantes en los que resalta notablemente la vegetación y los cuerpos de agua, como se puede apreciar en la figura adjunta. Este contraste vívido facilita la identificación y delimitación precisa de los elementos naturales dentro de la cuenca.

### Figura 5

*Imagen de la zona del satélite Sentinel s1.*



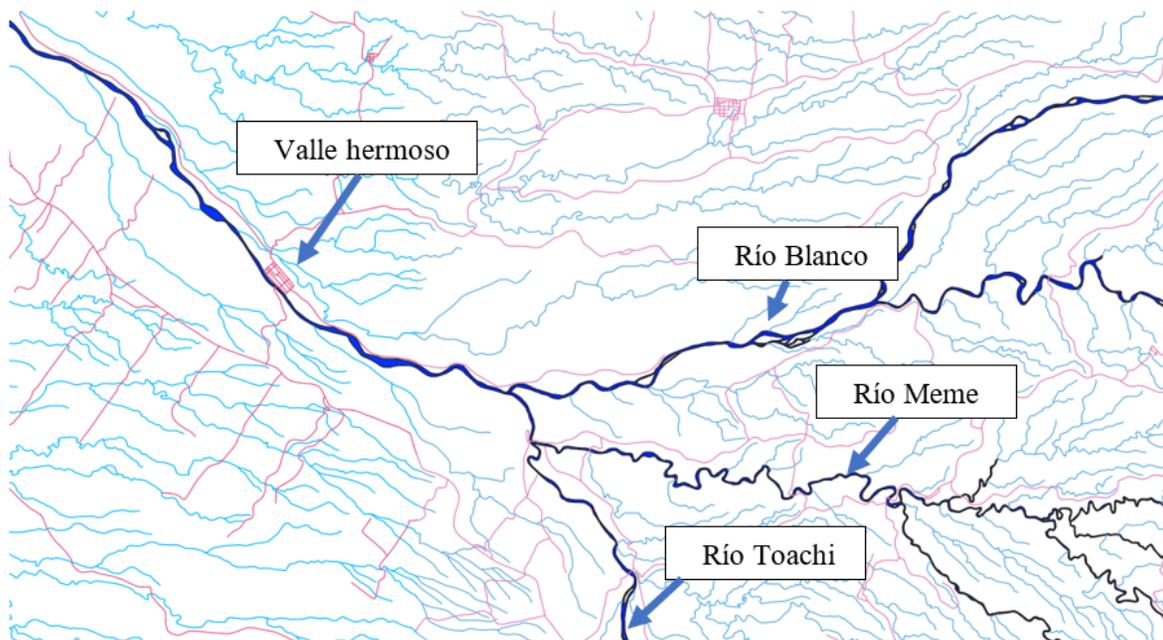
Fuente: (Europea, 2024)

Para obtener una visualización más detallada del terreno, se procede a realizar una extracción de polígonos. Estos polígonos se obtienen del Instituto Geográfico Militar del Ecuador, lo cual garantiza una representación más precisa y confiable del área de interés. Esta información resulta fundamental para facilitar una mejor manipulación del mapa a través de QGIS.

Este proceso brinda la oportunidad de analizar con mayor profundidad la configuración geográfica de la zona, lo que permite identificar con claridad los límites y características de los cuerpos de agua, así como otros elementos relevantes del paisaje. (Abad, 2013)

### Figura 6

*Mapa hidrográfico.*



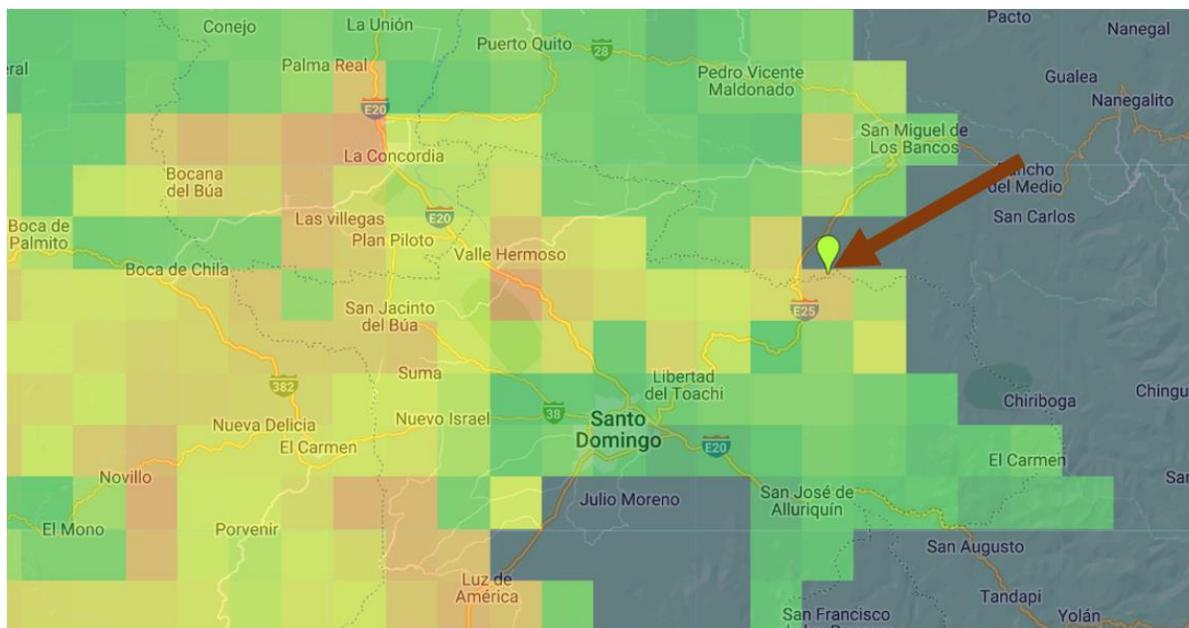
Fuente: (Abad, 2013)

### 2.3. Medición de Lluvia y Nivel de Agua de los Ríos

Para la medición de la lluvia, nuevamente se empleó el software Google Earth Engine. En esta ocasión, se hizo uso del satélite CHIRPS, que registra la cantidad de agua caída diariamente y proporciona imágenes con una resolución de 11km x 11km. Esta tecnología permite una observación detallada de los patrones de precipitación a lo largo del tiempo y en diferentes regiones geográficas. La información recopilada a través de CHIRPS es fundamental para el monitoreo de fenómenos climáticos. (Argote, 2023)

## Figura 7

### *Imagen satelital del satélite Chirps*

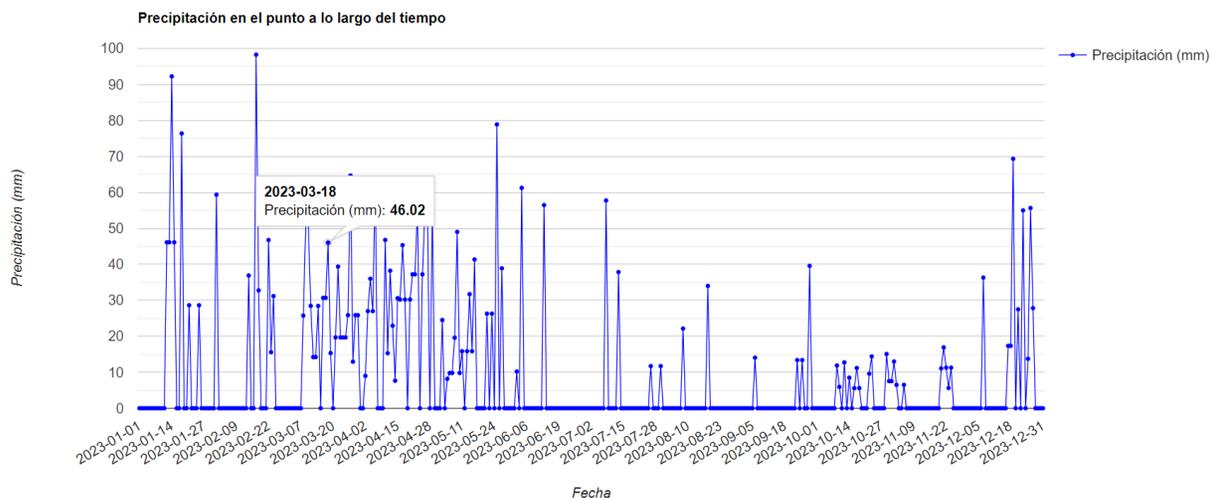


Fuente: (California, 2023)

En la imagen se observa una representación obtenida a partir de los datos proporcionados por el satélite CHIRPS entre el 17 y 18 de marzo del 2023. Esta fecha fue elegida debido a que coincide con la ocurrencia de la última inundación, como se mencionó en el primer capítulo. El punto seleccionado y marcado en la imagen fue elegido porque, según los datos del satélite, esta fue la sección del río que experimentó la mayor precipitación. Específicamente, las coordenadas exactas son Longitud:  $-78.97675000018374$  y Latitud:  $-0.10394996656794489$ . La precipitación promedio registrada en esta ubicación en la fecha seleccionada fue de 30.67 milímetros.

## Figura 8

### *Precipitación por fechas.*



Fuente: (California, 2023)

En un análisis por fechas, se observa que, en el punto seleccionado, ese día en concreto, se registraron 46.02 mm de precipitación. Aunque se pueden identificar picos más altos en otros momentos, es importante destacar que estos picos suelen ocurrir después de períodos extensos sin precipitaciones significativas. Por el contrario, en la fecha del 18 de marzo, se observa que habían transcurrido varios días con lluvias constantes antes de la fecha seleccionada.

Este patrón sugiere una diferencia en la dinámica de la precipitación entre los distintos eventos observados. Mientras que en el caso de los picos más altos se experimentó una precipitación intensa en un solo día, con períodos de sequía intercalados. El 18 de marzo estuvo marcado por una precipitación más constante y sostenida a lo largo de varios días. Este análisis detallado de los datos puede proporcionar información valiosa para comprender la variabilidad climática y sus posibles impactos en la hidrología local y los riesgos de inundaciones.

Para complementar el sistema, se ha optado por integrar el sistema de OpenWeather debido a su capacidad para proporcionar información precisa en aplicaciones de tiempo real. OpenWeather ofrece una amplia gama de datos meteorológicos, incluyendo pronósticos actuales y futuros, condiciones climáticas en tiempo real, y datos históricos. Su API robusta y fácil de usar permite la integración fluida con el sistema, lo que nos permite acceder rápidamente a la información meteorológica relevante para la aplicación. Con la inclusión del sistema de OpenWeather, se puede mejorar la precisión y relevancia de los datos recopilados. (Openwather, 2024)

#### **2.4. Funcionamiento del Sistema de Alerta**

El funcionamiento del sistema de alerta temprana de inundaciones es similar al de otros sistemas de alerta temprana y consta de cuatro capas fundamentales.

En la primera capa, se lleva a cabo la captura de los niveles físicos mediante sensores. Esta medición proporciona información crucial para evaluar el riesgo de inundación y permite monitorear los cambios en el nivel del agua en tiempo real.

La segunda capa implica el procesamiento de los datos recopilados para obtener la información necesaria. Los datos capturados por los sensores ultrasónicos son enviados a una computadora, que actúa como la unidad de procesamiento central. Aquí, se realiza un análisis detallado de los datos, interpretando las mediciones y determinando si se alcanzan niveles críticos de inundación que requieran emitir alertas.

La tercera capa hace uso de una interfaz en la nube, que cumple varias funciones críticas:

- **Análisis y Procesamiento de Datos:** La nube recibe los datos y utiliza algoritmos específicos para analizarlos en tiempo real, detectando patrones y anomalías.
- **Almacenamiento de Datos:** Los datos son almacenados de manera segura y redundante, permitiendo el acceso y análisis a largo plazo.

- **Interconexión entre Dispositivos:** La nube facilita la comunicación entre diferentes dispositivos del sistema, asegurando que todos los componentes estén sincronizados y operen de manera coordinada.

Por último, la cuarta etapa involucra la transmisión de información a los usuarios finales mediante aplicaciones específicas.

Este sistema integral asegura que la información relevante sobre el riesgo de inundaciones sea capturada, procesada, analizada y transmitida de manera eficiente, permitiendo a los usuarios tomar decisiones informadas y actuar rápidamente en caso de emergencias.

#### **2.4.1. Etapa de toma de datos**

##### **2.4.1.1. Tipos de sensores**

Para el sistema de alerta temprana, la adquisición de datos se realiza mediante sensores conectados a una placa de desarrollo. En este caso, se dispone de diversas opciones, incluyendo sensores de nivel, sensores de caudal, sensores ultrasónicos y tecnología de visión artificial. A continuación, se presenta una tabla comparativa para analizar detalladamente cuál sería la mejor alternativa para el sistema:

**Tabla 6**

*Comparación de tecnologías de sensor.*

<b>Tipo de Sensor</b>	<b>Características</b>	<b>Ventajas</b>	<b>Desventajas</b>
Sensores de Nivel	Detectan el nivel del agua en un punto específico.	Precisos en mediciones puntuales.	Requieren instalación precisa y mantenimiento.
Sensores de Caudal	Miden la cantidad de agua que fluye a través de un punto.	Proporcionan información sobre el flujo de agua en tiempo real.	Requieren una instalación y calibración precisas.

Sensores Ultrasonicos	Utilizan ondas ultrasónicas para medir la distancia al agua.	No requieren contacto directo con el agua.	Pueden ser afectados por condiciones climáticas adversas.
Visión Artificial	Utiliza cámaras para analizar imágenes y detectar agua.	Puede proporcionar información detallada sobre la extensión de inundaciones.	Requiere iluminación adecuada y procesamiento computacional.

Basándose en un análisis exhaustivo de cada alternativa, se llega a la conclusión de que los sensores ultrasónicos son la mejor opción para el sistema de alerta temprana de inundaciones. Aunque los sensores de nivel son precisos, su instalación dentro del agua puede resultar complicada dada la dimensión del río. De manera similar, los sensores de caudal también enfrentan desafíos logísticos debido a su ubicación en el agua. Por otro lado, aunque la visión artificial recopila una cantidad significativa de información, requiere sistemas más complejos que pueden aumentar los costos y reducir la eficiencia del sistema.

#### ***2.4.1.2. Elección del sensor de nivel***

Los sensores ultrasónicos, a pesar de ser susceptibles a condiciones climáticas adversas, cuentan con modelos específicamente diseñados para aplicaciones como la medición de niveles de agua en ríos y cuerpos de agua. Estos modelos están equipados con características que les permiten funcionar de manera confiable incluso en entornos desafiantes, lo que los convierte en la opción más adecuada para el sistema de alerta temprana de inundaciones.

**Tabla 7**

*Tabla comparativa de sensores de ultrasonido.*

Parámetros Técnicos	JSN-SR04T	MB1240	HC-SR04	MB7076
Figura				
Tipo de sensor	Ultrasónico	Ultrasónico	Ultrasónico	Ultrasónico
Alimentación	3.3V – 5V	3.3V – 5V	5V	3V – 5V
Corriente	5 mA	3.4 mA	5 mA	3.4 mA
Frecuencia	40 kHz	42 kHz	40 kHz	
Rango de detección	Hasta 6 m	20cm 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	D25*L19 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	\$8	\$50	\$2.50	\$132

Fuente: (Ramirez, 2022)

Basándose en un análisis exhaustivo de cada alternativa, se llega a la conclusión de que el sensor elegido para el sistema de alerta temprana de inundaciones es el JSN-SR04T. Este sensor se seleccionó no solo por sus características mostradas en la tabla comparativa, sino también por sus

cualidades físicas, como su resistencia al agua y su longitud de cableado de 2.48 metros. Estas características permiten instalar la parte del sensor más cercana al agua, es decir, la bocina, a una distancia óptima, mientras que el resto de los componentes pueden ubicarse alejados de la zona de riesgo.

El JSN-SR04T es especialmente adecuado para entornos donde se requiere una medición precisa y confiable de los niveles de agua, ya que su diseño resistente al agua y su capacidad de cableado lo hacen ideal para aplicaciones en ríos y cuerpos de agua. Esta elección garantiza un funcionamiento eficiente y confiable del sistema de alerta temprana de inundaciones, incluso en condiciones climáticas adversas y entornos desafiantes.

Para la siguiente etapa del sistema, se utiliza un microcontrolador para la captación, procesamiento y envío de datos. En el mercado, existe una amplia gama de opciones disponibles; sin embargo, en este caso se analizan específicamente aquellas que cumplen con criterios de tamaño compacto, bajo consumo de energía y bajo coste.

#### ***2.4.1.3. Microcontroladores***

La selección de un microcontrolador adecuado es crucial para garantizar un funcionamiento eficiente y económico del sistema de alerta temprana de inundaciones. Los requisitos de tamaño compacto son importantes para facilitar la integración del microcontrolador en el diseño general del sistema, mientras que el bajo consumo de energía es fundamental para prolongar la vida útil de la batería y reducir los costes operativos. Además, el bajo coste es un factor determinante para mantener el proyecto dentro de un presupuesto razonable.

Al considerar estas características, se llevará a cabo un análisis detallado de las opciones disponibles en el mercado para seleccionar el microcontrolador más adecuado para el sistema de alerta temprana de inundaciones. Se evaluarán aspectos como la potencia de procesamiento, la

facilidad de programación, la compatibilidad con otros componentes del sistema y la disponibilidad de soporte técnico y documentación. Este proceso de selección garantizará que el microcontrolador elegido cumpla con los requisitos específicos del proyecto y contribuya al desarrollo de un sistema eficiente y confiable para la gestión de inundaciones.

**Tabla 8**

*Comparación entre microcontroladores.*

Modelo	ESP32	ESP8266	Particle Photon	Thing Plus	Artemis
Figura					
Voltaje de operación	5V	5V	3.6V – 5V	2.3V – 3.6V	5V
Consumo de corriente	20 mA	80 mA	80 mA	20 mA	-
Memoria Flash	16 MB	4 MB	1 MB	16 MB	1 MB
Memoria RAM	520 kB	96 kB	128 kB	520 kB	384 kB
Velocidad de reloj	Hasta 240 MHz	80MHz	120 MHz	240 MHz	48 MHz
Pines digitales I/O	36	17	18	21	24

Pines de entrada analógica	15	1	6	6	10
Wi-Fi	802.11 b/g/n	802.11b/g/n	802.11b/g/n	802.11 b/g/n	No
Bluetooth	Si	No	No	Si	Si
Costo	\$14	\$10	\$40	\$40	\$35

Fuente: (Ramirez, 2022)

#### 2.4.1.4. Tecnología inalámbrica.

Para la elección de la tecnología de envío de datos, se consideraron varias tecnologías inalámbricas, las cuales fueron comparadas para determinar cuál se ajusta mejor al proyecto. Este análisis comparativo permitirá seleccionar la opción más adecuada en términos de alcance, consumo de energía, velocidad de transmisión, costos y facilidad de implementación.

Entre las tecnologías inalámbricas evaluadas se encuentran en la siguiente tabla:

**Tabla 9**

*Comparación entre tecnologías inalámbricas.*

<b>Tecnología</b>	<b>Características</b>	<b>Limitaciones</b>
<b>Wi-Fi</b>	Ampliamente disponible, alta velocidad de transmisión. Requiere infraestructura estable, consumo energético alto.	Infraestructura necesaria, consumo energético elevado.
<b>Bluetooth</b>	Corto alcance, consumo de energía moderado, bajo costo.	Limitado en alcance y velocidad de transmisión.

<b>LoRaWAN</b>	Alcance extendido, bajo consumo de energía.	Velocidad de transmisión más baja, infraestructura específica necesaria.
<b>Sigfox</b>	Alcance y consumo de energía similar a LoRaWAN. Cobertura limitada, posibles costos asociados.	Cobertura limitada, posibles costos de red.
<b>Narrowband IoT</b>	Buen equilibrio entre alcance, consumo de energía y velocidad de transmisión. Diseñado para IoT.	Requiere suscripciones de datos, acceso a red celular.

Al evaluar estas tecnologías, se considerarán factores como la ubicación del sistema, los requisitos de alcance y cobertura, la disponibilidad de infraestructura de red, el consumo de energía y los costos asociados. Esta comparación detallada permitirá tomar una decisión informada y seleccionar la tecnología de envío de datos que mejor se adapte a las necesidades específicas del proyecto de alerta temprana de inundaciones.

La elección de LoRaWAN como tecnología de envío de datos para el proyecto se basa en una evaluación cuidadosa de las opciones disponibles y sus características. Aunque una de las desventajas de LoRaWAN es su velocidad de transmisión más baja en comparación con otras tecnologías, esta limitación no afecta significativamente al proyecto, ya que no se requieren altas velocidades de transmisión para la recolección y envío de datos relacionados con la alerta temprana de inundaciones.

En cambio, LoRaWAN ofrece importantes ventajas que se alinean con los requisitos del proyecto. Su bajo consumo de energía y su capacidad para transmitir datos a largas distancias son características esenciales para un sistema de monitoreo de inundaciones, donde se requiere una operación continua y de larga duración, así como la capacidad de cubrir áreas extensas.

Además, la infraestructura de red LoRaWAN es relativamente fácil de implementar y puede proporcionar una cobertura confiable incluso en áreas remotas o con acceso limitado a la

infraestructura de red tradicional. Esto hace que LoRaWAN sea una opción sólida y adecuada para el proyecto de alerta temprana de inundaciones, donde la fiabilidad y la eficiencia energética son consideraciones clave.

#### **2.4.1.5. Módulo LORA**

La selección del módulo se realizará en base a su compatibilidad con el sistema y su alcance. Con el fin de elegir de manera más precisa, se llevará a cabo una comparación detallada de varios modelos disponibles en el mercado.

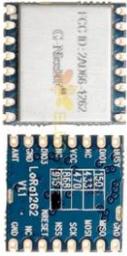
Se evaluarán aspectos como la frecuencia de operación, la potencia de transmisión, la sensibilidad del receptor, la capacidad de penetración en obstáculos y la interoperabilidad con otros dispositivos. Además, se considerarán factores como el tamaño físico, el consumo de energía y el costo del módulo.

La elección final se basará en encontrar el equilibrio adecuado entre estas características para garantizar un rendimiento óptimo del sistema de alerta temprana de inundaciones. Es importante seleccionar un módulo que ofrezca una comunicación confiable a larga distancia y que sea compatible con la infraestructura LoRaWAN existente, asegurando así una integración sin problemas en el sistema global de monitoreo.

**Tabla 10**

*Comparación entre módulos LORA.*

<b>Parámetros</b>	<b>LoRa1262</b>	<b>RYLR99</b>	<b>E32-TTL-100</b>	<b>Módulo</b>	<b>Ra-07H</b>
<b>Técnicos</b>		<b>8</b>		<b>RFM95</b>	

<b>Figura</b>					
<b>Modulación</b>	LoRa	LoRa	LoRa	LoRa	LoRa
<b>Frecuencia de operación</b>	433/470/868/915 MHz	868 MHz - 915 MHz	410 MHz - 441MHz	868 MHz - 915 MHz	803 MHz - 930 MHz
<b>Velocidad de bits</b>	0.018-62.5 Kbps	300 kbps	19.2 kbps	Hasta 300 kbps	Hasta 50 kbps
<b>Tensión de funcionamiento</b>	3.3V – 3.7V	1.8 V – 3.7 V	2.3V – 5.5V	1.8 V – 3.7 V	3.3V
<b>Sensibilidad</b>	-148 dBm	-129 dBm	-130 dBm	Hasta -148 dBm	-137dBm
<b>Costo</b>	\$17	\$15	\$17	\$10	\$13

Fuente: (Ramirez, 2022)

El módulo RYLR998 es la opción adecuada para el sistema de alerta temprana de inundaciones. Sus características, que incluyen un alcance de hasta 8 km con línea de vista y funciones de ahorro de energía inteligente, son particularmente adecuadas para las necesidades del proyecto en Ecuador. Además, su diseño compacto lo hace altamente compatible con el sistema global de monitoreo.

La capacidad del módulo RYLR998 para alcanzar distancias de transmisión significativas, junto con sus características de ahorro de energía, son factores clave para garantizar una comunicación confiable y de larga duración en áreas remotas o de difícil acceso. Esto es especialmente relevante para un proyecto de alerta temprana de inundaciones, donde la capacidad de transmitir datos de manera eficiente y sin interrupciones es fundamental para la efectividad del sistema.

Además, la compatibilidad del módulo con la reglamentación de Ecuador garantiza su idoneidad para el entorno local, lo que contribuye a la fiabilidad y eficacia del sistema en la detección y alerta temprana de posibles inundaciones.

#### **2.4.1.6. Diseño del sistema de toma de datos**

Para el diseño del sistema de toma de datos, se tienen en cuenta todos los componentes elegidos previamente. El sensor ultrasónico JSN-SR04T se conecta directamente a los puertos 22 y 23 del microcontrolador ESP32, así como a los puertos de tierra y 3.3 voltios para la alimentación adecuada del sensor. El microcontrolador recibe los datos del sensor y los procesa para proporcionar información en centímetros sobre la distancia hasta la superficie del agua.

Los datos procesados son enviados por los puertos de comunicación serial 16 y 17 del ESP32. Estos puertos están conectados a los puertos seriales del módulo LoRa RYLR998. El módulo LoRa utiliza comandos AT para su configuración y envío de datos a través de la red LoRaWAN.

**Tabla 11**

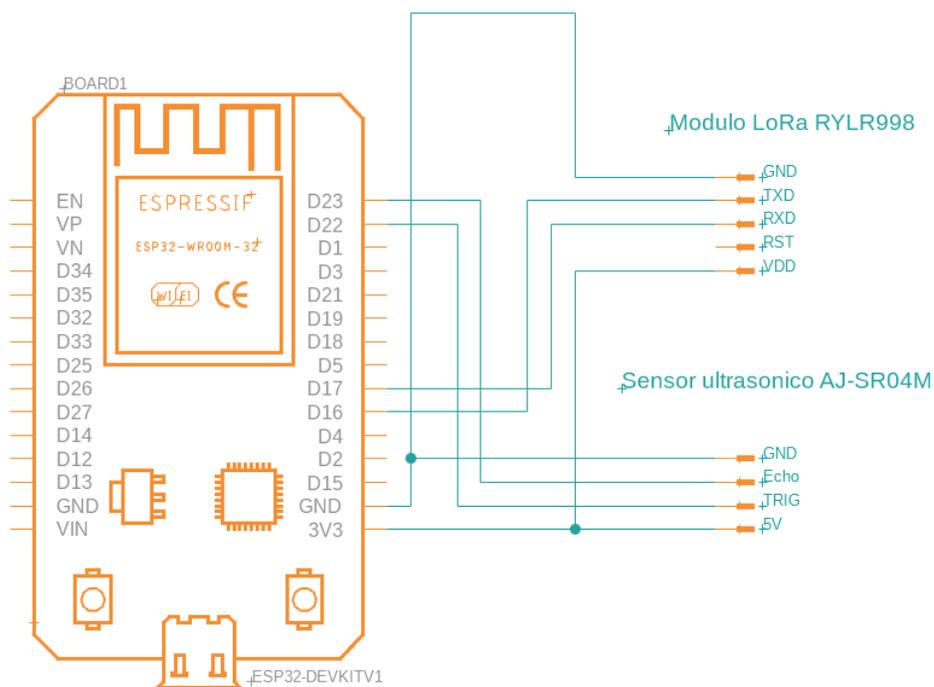
*Conexiones del Emisor.*

<b>Conexiones del sensor</b>	
<b>ESP32</b>	<b>AJ-SR04M</b>
3V3	5V
D22	TRIG

D23	Echo
GND	GND
Conexiones del módulo LoRa	
ESP32	RYLR998
GND	GND
D16	TXD
D17	RDX
-	RST
3V3	VDD

**Figura 9**

*Diagrama de conexiones*



Una vez que se tiene el esquema de conexiones y se han definido los componentes clave del sistema, se procede a realizar el diseño del PCB del sistema.

## Figura 10

*Diseño de placa.*



### *2.4.1.7. Configuración de transmisor lora*

En el caso del módulo LoRa RYLR998, las configuraciones se realizan mediante comandos AT de acuerdo con las especificaciones del fabricante. Estas configuraciones deben definirse conforme a la tabla AU915, detallada en el capítulo 2, y bajo las capacidades del dispositivo, con el objetivo de alcanzar la máxima distancia posible sin pérdida de datos.

Para lograr este objetivo, se elige el máximo spread factor que el fabricante permite, con un ancho de banda de 125 kHz. En este caso, se selecciona un spread factor SF9, ya que maximiza el alcance de la señal sin comprometer la integridad de los datos. El resto de los parámetros se mantienen predefinidos, ya que cumplen con las directrices establecidas por la LoRa Alliance para Ecuador.

Para comprobar la validez de las configuraciones se realizan los siguientes cálculos.

**Datos****Figura 11***Datos de configuración.*

<b>Mensaje:</b> "+RCV=50, 5, 600, -99, 40"
<b>Banda :</b> 915MHz
<b>UART :</b> 115200
<b>Spreading Factor :</b> SF9
<b>Bandwidth :</b> 125kHz
<b>Coding Rate :</b> 1
<b>Preamble Length :</b> 12

**Tamaño del mensaje**

$$26 \text{ caracteres} \times \frac{8 \text{ bits}}{\text{caracter}} = 208 \text{ bits}$$

**Data rate**

$$R = \frac{SF \times BW}{2^{SF}} \times \frac{4}{4 + CR}$$

**Tasa de símbolo**

$$R_s = \frac{BW}{2^{SF}} = \frac{125 \times 10^3}{2^9} = \frac{125000}{512} = 244.14 \text{ simbolo/segundo}$$

**Tasa de símbolo ajustada a la tasa de codificación**

$$R = R_s \times \frac{4}{4+1} = 244.14 \times \frac{4}{5} = 244.14 \times 0.8 = 196 \text{ bits/segundo}$$

**Tiempo de transmisión**

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

**Tiempo de preambulo**

$$T_{preamble} = (N_{preamble} + 4.25) \times T_s$$

$$T_{preamble} = (12 + 4.25) \times \frac{2^{SF}}{BW} = 16.25 \times \frac{2^9}{125000} = 16.25 \times 0.004096$$

$$T_{preamble} = 0.06656 \text{ segundos}$$

**Tiempo de payload**

$$T_{payload} = T_s \times \left( 8 + \left( \left\lceil \frac{8 \times PL - 4SF + 28 + 16CR - 20H}{4 \times (SF - 2DE)} \right\rceil \times (CR + 4) \right) \right)$$

donde:

- PL = longitud del payload en bytes = 26 bytes
- H = 0 para encabezado explícito, 1 para implícito
- DE = 1 si SF ≥ 11, 0 si SF < 11

$$payloadSymbNb = 8 + \left( \left\lceil \frac{8 \times 26 - 4 \times 9 + 28 + 16 \times 1 - 20 \times 0}{4 \times (9 - 2 \times 0)} \right\rceil \times (1 + 4) \right)$$

$$= 8 \left( \left\lceil \frac{208 - 36 + 28 + 16}{28} \right\rceil \times 5 \right)$$

$$= 8 \left( \left\lceil \frac{216}{28} \right\rceil \times 5 \right)$$

$$= 8 + (8 \times 5)$$

$$= 48 \text{ Símbolos}$$

Entonces

$$T_{payload} = 48 \times T_S = 48 \times 0.04096 = 0.196608 \text{ segundos}$$

$$T_{oA} = T_{preamble} + T_{payload} = 0.06656 + 0.196608 = 0.263168 \text{ segundos}$$

El tiempo total de transmisión (ToA) del mensaje es aproximadamente 0.263 segundos. Esto es mucho menor que 1 segundo, por lo que es posible transmitir el mensaje "+RCV=50, 5, 600, -99, 40" cada segundo con los parámetros (SF=9, BW=125 kHz, CR=1, preámbulo de 12) en la banda de 915 MHz.

#### 2.4.1.8. Programación del Emisor

La programación del sistema, referenciado en el **Anexo 1**, se llevó a cabo utilizando el Entorno de Desarrollo Integrado (IDE) de Arduino, aprovechando la versatilidad que ofrece para proyectos de este tipo. Se utilizó la librería específica para ESP32 para la integración del microcontrolador con el resto del hardware. Asimismo, se empleó una librería compatible con el sensor ultrasónico para facilitar su funcionamiento y la adquisición precisa de datos.

En cuanto al módulo LoRa, su comunicación se establece a través del puerto serial2 del ESP32. Es importante destacar que, a diferencia de otros componentes, no fue necesario utilizar librerías adicionales para su configuración y operación ya que esta se realiza a través de comandos AT. Esto simplifica el proceso de programación y garantiza una comunicación eficiente entre dispositivos.

En primer lugar, se definen dos variables de tipo String para almacenar las direcciones del módulo LoRa y del módulo receptor LoRa. Luego, se definen dos constantes para los pines de trigger y echo del sensor ultrasónico. Posteriormente, se declaran tres variables para almacenar la duración

del pulso ultrasónico, la distancia medida y el texto de entrada recibido por el módulo LoRa receptor.

En la función de configuración (`setup()`), se inicializa la comunicación serial para el módulo LoRa con una velocidad de 115200 baudios y un formato de 8 bits, sin paridad y 1 bit de stop. También se configuran los pines del sensor ultrasónico, el pin de trigger como salida y el pin de echo como entrada. Además, se inicia la comunicación serial para la monitorización a una velocidad de 115200 baudios y se envía un comando AT para establecer la dirección del módulo LoRa. Después de cada comando AT, se agrega un retraso de 1.5 segundos.

En el bucle principal (`loop()`), se realiza la medición de distancia utilizando el sensor ultrasónico. Se envía un pulso al pin de trigger del sensor y se mide la duración del pulso de eco utilizando la función `pulseIn()`. Luego, se calcula la distancia en centímetros utilizando una fórmula de conversión y se convierte este valor a una cadena de caracteres. Se envía la distancia al módulo LoRa receptor utilizando comandos AT y también se imprime en el monitor serial local. A continuación, se lee cualquier dato de entrada recibido del módulo LoRa receptor y se muestra en el monitor serial. Finalmente, se agrega un retraso de 1 segundo antes de repetir el bucle. (REYAX, 2021)

#### ***2.4.2. Procesamiento de datos.***

Para la etapa de procesamiento de datos, es crucial que esta capa actúe como un intermediario eficaz entre la capa de toma de datos y la capa en la nube. Esta etapa debe cumplir con los siguientes requisitos:

1. Interfaz para la Obtención del Canal LoRa:

- El sistema necesita una interfaz para recibir los datos del canal LoRa. Esto implica el uso de un módulo LoRa compatible, como el RYLR998, que se conectará al dispositivo de procesamiento.
2. Interfaz para la Salida a Internet:
    - Se requiere una interfaz de salida para enviar los datos procesados a la nube. Esto podría ser mediante Ethernet, Wi-Fi, o una conexión móvil, dependiendo de la disponibilidad de infraestructura en el área.
  3. Capacidades Computacionales Elevadas:
    - Es esencial utilizar un dispositivo con capacidades computacionales suficientes para procesar los datos de manera adecuada. Un ejemplo adecuado para este propósito es la Raspberry Pi, que ofrece una combinación de capacidad de procesamiento y flexibilidad.

#### ***3.4.2.1. Receptor LORA***

Utilizar el mismo microcontrolador ESP32 y módulo LoRa tanto en el emisor como en el receptor, siguiendo los lineamientos de hardware libre propuestos para el proyecto, es una decisión estratégica que ofrece numerosas ventajas. Además de garantizar una máxima compatibilidad entre los dispositivos, simplifica significativamente tanto la configuración como el montaje de los dispositivos. Algunas de las razones clave para adoptar este enfoque son:

- **Compatibilidad y Coherencia:** Al utilizar el mismo hardware en ambos extremos de la comunicación, se establece una base sólida para una comunicación coherente y confiable. Esto minimiza las posibles discrepancias y reduce la complejidad en la configuración del sistema.
- **Simplicidad en la Configuración:** Al tener un conjunto uniforme de componentes en el emisor y el receptor, se simplifica en gran medida el proceso de configuración. No es

necesario ajustar configuraciones o realizar adaptaciones adicionales para garantizar la interoperabilidad.

- **Facilita el Desarrollo y Depuración:** La consistencia en el hardware facilita tanto el desarrollo como la depuración del sistema. Los problemas relacionados con la comunicación o la compatibilidad pueden identificarse y abordarse más rápidamente gracias a la uniformidad en los componentes utilizados.
- **Flexibilidad y Escalabilidad:** A pesar de utilizar componentes idénticos, este enfoque aún permite flexibilidad y escalabilidad en el diseño. Se pueden realizar ajustes y mejoras según sea necesario, manteniendo la base de compatibilidad entre el emisor y el receptor.

Si bien el sistema ESP32 con el módulo LoRa RYLR998 ofrece una solución sólida para la comunicación inalámbrica, es cierto que puede carecer de la potencia computacional necesaria para el procesamiento de datos y la subida a la nube. Por lo tanto, es necesario agregar un tercer componente que cumpla con estos requerimientos, para esto se hace uso de la siguiente tabla comparativa para poder elegir un equipo con la potencia necesaria para cumplir con los requerimientos.

**Tabla 12**

*Comparación de computadoras.*

Parámetros	Raspberry Pi	Orange Pi	ASUS Tinker Board S	Rock64 Media Board	Banana Pi M64
Técnicos	3	PLUS			

Figura					
CPU	Procesador de cuatro núcleos a 1,4 GHz Cortex-A53	H3 Quad-core Cortex-A7	Cortex-A17 Quad-core 1.8GHz	Rockchip RK3328	Amlogic S905X3
GPU	VideoCore VI	Mali400MP2 @ 600MHz Soporta OpenGL ES 2.0	ARM Mali-T764 - Compatible con 4K	Mali 450 MP2	Mali-G31
Núcleos	4	4	4	4	4
Memoria	1/2/4GB LPDDR3 RAM	DDR3 de 1GB (compartido con GPU)	Dual-CH LPDDR3 2GB	1, 2 y 4 GB de RAM LPDDR3	4 GB de LPDDR4
Conectividad	802.11ac Wi-Fi / Bluetooth 5.0, Gigabit Ethernet	Realtek RTL8189ETV, IEEE 802.11 b/g/n	Wi-Fi 802.11 b/g/n, Bluetooth 4.0, Gigabit Ethernet	Gigabit Ethernet	10/100/1000 Mb/s de Ethernet

Video	HDMI	HDMI	HDMI	HDMI 2.0a, audio/micrófono de 3,5 mm, GPIO de 40 clavijas, GPIO de 22 clavijas	HDMI 2.0, audio/micrófono de 3,5 mm, 28 x GPIO, UART, I2C, SPI, PWM
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	2 x USB 2.0, 1 x USB 3.0	USB 3.0, 1GbE puerto LAN
Alimentación	5V/2A vía USB-C, 5V vía cabezal GPIO	La entrada de CC, la entrada USB OTG no suministra energía	microUSB o por pines	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	\$50	\$81,99	\$129,22	\$85,92	\$75,18

*Fuente: (Ramirez, 2022)*

En términos de precio y capacidad de procesamiento, la Raspberry Pi 3 sigue siendo una opción sólida a pesar de pertenecer a una generación anterior. Esta pequeña computadora de placa única ofrece las conexiones necesarias para integrarse perfectamente en un sistema de recepción de datos. Además de su atractivo precio, la Raspberry Pi 3 es una opción viable gracias a su

versatilidad en cuanto a conectividad. No solo cuenta con puertos de conexión alámbrica, sino que también dispone de opciones inalámbricas que permiten una conexión estable a internet. Esta capacidad dual de conexión asegura que se cumplan con los requisitos necesarios para el sistema de recepción y procesamiento de datos.

Además de sus características de conectividad, la Raspberry Pi 3 ofrece un rendimiento suficiente para manejar las tareas de procesamiento de datos requeridas por el sistema. Su procesador de cuatro núcleos y su capacidad de memoria permiten ejecutar aplicaciones y realizar cálculos sin problemas, lo que la convierte en una opción confiable para las necesidades pertinentes.

La comunicación entre el ESP32 y la raspberry se la realiza mediante conexión serial en los puertos USB de los dispositivos con un cable USB C a USB.

#### ***3.4.2.2. Elección de plataforma IOT.***

Para poder realizar la conexión entre el receptor LoRa y la plataforma en la nube, primero se debe elegir el servicio que más convenga para el proyecto ya que cada plataforma tiene distintos protocolos y procesos para el envío de datos para esto se hará uso de la siguiente tabla.

**Tabla 13**

*Comparación de plataformas IOT.*

<b>Características</b>	<b>Microsoft Azure</b>	<b>AWS</b>	<b>Thingier.io</b>	<b>ThingSpeak</b>	<b>Google Firebase</b>
<b>Protocolos</b>	HTTP, AMQP, MQTT	HTTP, WebSockets, MQTT	HTTP, MQTT, IotMp	HTTP, MQTT, CoAP,	HTTP, MQTT
<b>Hardware Certificado</b>	Intel, Raspberry, FreeScale, Texas Instruments	Broadcom, Marvell, Texas Instruments, Intel.	ESP32, Sistemas Linux.	Intel, Raspberry, FreeScale, Texas Instruments	Broadcom, Marvell, Texas Instruments, Intel.

<b>Lenguajes</b>	.Net, Java, NodeJS	Java, C, NodeJS	C#, C++, Phyton, Java, NodeJS	Phyton, Java, NodeJS.	C#, C, Phyton, Java, NodeJS
<b>Precio</b>	Pagado, depende de la suscripción.	Pagado, depende de la suscripción	Ofrece una versión gratuita no comercial	Ofrece una versión gratuita no comercial	Ofrece una versión gratuita no comercial

*Fuente: (Ramirez, 2022)*

Al evaluar diferentes opciones de plataformas en la nube, se decidió descartar AWS y Azure debido a sus políticas de pago que pueden resultar costosas una vez se superen ciertas cantidades de datos y dispositivos. Además, ThingSpeak presentaba limitaciones en cuanto al tiempo de muestreo y obtención de datos, lo cual podría ser perjudicial en aplicaciones que requieran datos en tiempo real. Aunque Firebase ofrece una amplia gama de aplicaciones y funcionalidades, está más orientado hacia el desarrollo de aplicaciones móviles, por lo que no era la mejor opción para este proyecto.

La plataforma elegida fue thinger.io, que cuenta con una amplia variedad de herramientas para el Internet de las Cosas (IoT). thinger.io ofrece aplicaciones móviles tanto para iOS como para Android, paneles de control personalizables, endpoints y bases de datos integradas. Lo más destacado de thinger.io es su propio protocolo para IoT, que simplifica la configuración y comunicación con la plataforma, al tiempo que sigue siendo compatible con otros protocolos estándar. Esta plataforma proporciona una solución completa y robusta para la recepción, procesamiento y almacenamiento de datos de manera eficiente y rentable. (Bustamante, 2024)

### 3.4.2.3. *Diseño del sistema de procesamiento de datos*

En el diseño del sistema de procesamiento de datos debe tener en cuenta 3 fases, la primera en la que se reciben los datos de los sensores, la segunda en la que se procesan y la tercera en la que se envían a la nube.

Para la primera fase, la conexión se realiza utilizando la placa ESP32 y el módulo LoRa RYLR998. Debido a que la comunicación es de tipo serial, es esencial conectar correctamente los puertos del módulo LoRa con los de la ESP32. En este caso, el puerto RDX del módulo RYLR998 se debe conectar al puerto TDY (D17) del ESP32, y el puerto TDY del módulo LoRa debe conectarse al puerto RDX (D16) del ESP32. Esta configuración asegura una comunicación eficiente y estable entre los dispositivos, permitiendo la correcta recepción de datos de los sensores.

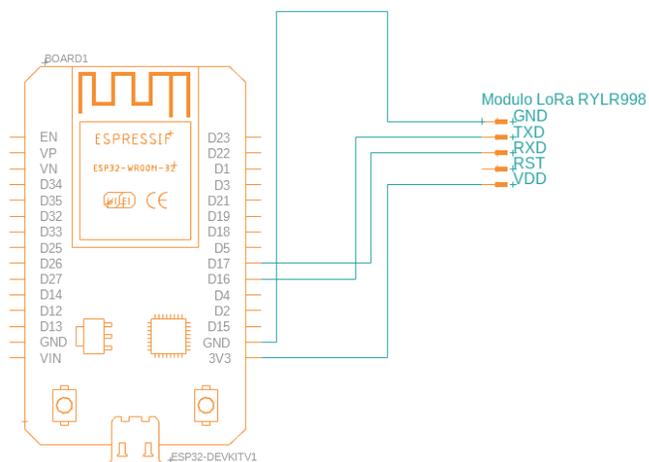
**Tabla 14**

*Conexiones del módulo receptor.*

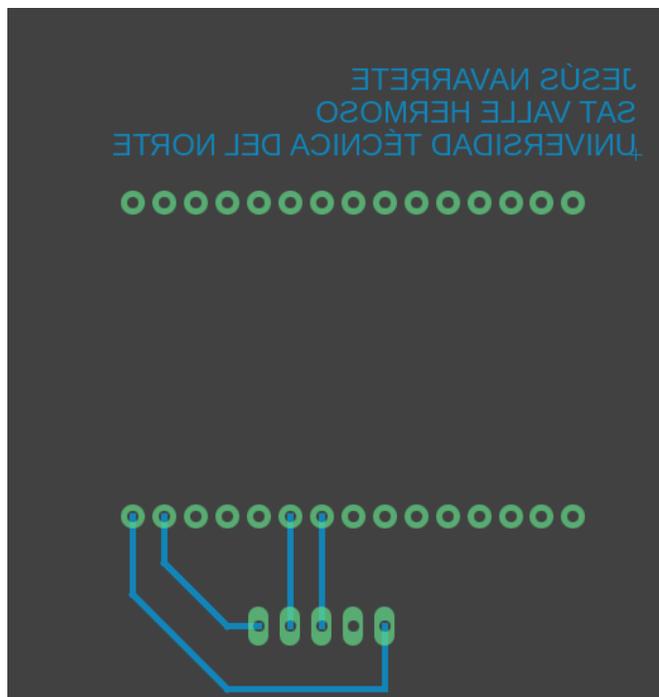
<b>Conexiones del módulo LoRa</b>	
<b>ESP32</b>	<b>RYLR998</b>
GND	GND
D16	TXD
D17	RDX
-	RST
3V3	VDD

**Figura 12**

*Diagrama de conexiones.*



Ya que se tienen las conexiones se realiza el diseño del PCV.

**Figura 13***Diseño de PCV.*

La conexión del Esp32 a la Raspberry se realiza mediante el cable USB para aprovechar la comunicación serial de ambos dispositivos.

## Figura 14

*Conexión de la raspberry con el ESP32.*



Fuente: (Rdagger, 2017)

### ***3.4.2.4. Programación del sistema receptor***

Al tener dos placas, se requiere un código para cada una de ellas, se realizará el análisis del script referenciado en el **Anexo 2**, usado para la transferencia de los datos entre el ESP32 a la raspberry. El código establece la comunicación con un módulo LoRa utilizando un microcontrolador ESP32. En la función `setup()`, se inicializa la comunicación serie tanto para comunicarse con la raspberry (Serial) como para el módulo LoRa (Serial2). Se envía un comando AT para establecer la dirección del módulo LoRa. La función `loop()` se encarga de recibir y procesar los datos entrantes desde el módulo LoRa. Los datos se almacenan en un buffer hasta que se recibe un mensaje completo, indicado por un carácter de nueva línea. Dependiendo del contenido del mensaje, se imprime en el monitor serie. El código utiliza pausas para asegurar que los comandos enviados al controlador central sean procesados correctamente.

### ***3.4.2.5. Programación del sistema de proceso de datos***

Para la programación del sistema se opta por usar el repositorio de Github para clientes Linux proporcionado por la plataforma thinger.io el cual usa el principal promotor del protocolo IOTMP que será el protocolo usado por el sistema.

Se modifica el repositorio descargado para que funcione como servicio de Linux y se adapte al sistema de alerta temprana, el código se divide en dos partes, el primero recibe, procesa los datos y los guarda en un archivo JSON, el segundo lee los archivos generados y los envía a thinger.io, también es el encargado de las condicionales para enviar las alertas.

A continuación, se expone el algoritmo del servicio “sensor”, (**referenciado en el Anexo 3**) encargado de leer y procesar los datos:

#### 1. Inicialización:

- Configura la velocidad del puerto serial y las rutas de los archivos JSON donde se almacenarán los datos.
- Define los parámetros específicos para los receptores rcv1 y rcv3 como altura máxima, cambio máximo permitido y altura de referencia.
- Inicializa variables globales para manejar los datos iniciales, referencia, últimos datos válidos y últimos crecimientos registrados.

#### 2. Encontrar Puerto Serial:

- Busca y prueba puertos seriales disponibles hasta encontrar uno que responda con datos esperados (+RCV=).
- Devuelve el puerto serial válido.

#### 3. Leer Datos del Puerto Serial:

- Abre el puerto serial y comienza a leer datos en un bucle continuo.

- Procesa cada línea de datos recibida del puerto serial.
4. Procesar Datos:
    - Divide la línea de datos recibida en partes.
    - Identifica el tipo de receptor (rcv1 o rcv3) y el valor de distancia.
    - Calcula el crecimiento del río para rcv1 y valida el cambio brusco.
    - Valida el crecimiento del río para rcv3 y detecta cambios bruscos.
    - Guarda los datos válidos en archivos JSON.
  5. Guardar Datos en Archivos JSON:
    - Abre y lee el archivo JSON correspondiente, si existe.
    - Añade el nuevo dato a la lista de datos existente.
    - Escribe la lista actualizada de vuelta en el archivo JSON.

Ahora se expone el algoritmo y el código del servicio “satvh”, (**Referenciado en el Anexo 4**) encargado de la comunicación con la nube.

1. Inicialización:
  - Define las constantes y variables globales necesarias, como las credenciales de Thinger.io, las rutas de los archivos JSON y las variables para almacenar las distancias y la precipitación.
  - Declara las funciones de callback, lectura de JSON, y envío de datos a Thinger.io.
2. Función enviar\_precipitacion\_periodicamente:
  - Obtiene datos de precipitación desde OpenWeatherMap y los envía a Thinger.io en intervalos regulares.
3. Función leer\_json:
  - Abre y lee el contenido de un archivo JSON específico.

- Si el archivo no está vacío y contiene datos válidos, extrae el último valor de crecimiento.
4. Función leer\_datos\_json\_continuamente:
- Lee continuamente los archivos JSON para rcv1 y rcv3.
  - Si se detectan cambios en las distancias, envía estos datos a Thinger.io.
  - Verifica si se cumplen las condiciones de alerta y llama al endpoint de Telegram si es necesario.
5. Función main:
- Inicializa la conexión con Thinger.io.
  - Crea y lanza hilos separados para el envío de datos de precipitación y la lectura continua de los archivos JSON.
  - Mantiene el bucle principal ejecutando thing.handle() para mantener la conexión con Thinger.io.

### ***3.4.3. Plataforma IoT***

En la capa anterior se eligió la plataforma a la que el sistema se va a conectar, es crucial analizar detalladamente todas las configuraciones que esta nos proporciona. Estas configuraciones serán fundamentales para garantizar que las opciones seleccionadas en la etapa anterior puedan ser procesadas de manera correcta y eficiente.

Se llevará a cabo un análisis exhaustivo de las configuraciones ofrecidas por la plataforma seleccionada, thinger.io, para asegurar que se puedan aprovechar al máximo todas las funcionalidades disponibles. Esto incluirá la revisión de opciones como la configuración de dispositivos, la definición de endpoints, la configuración de bases de datos, la personalización de paneles de control, y la configuración de reglas y notificaciones.

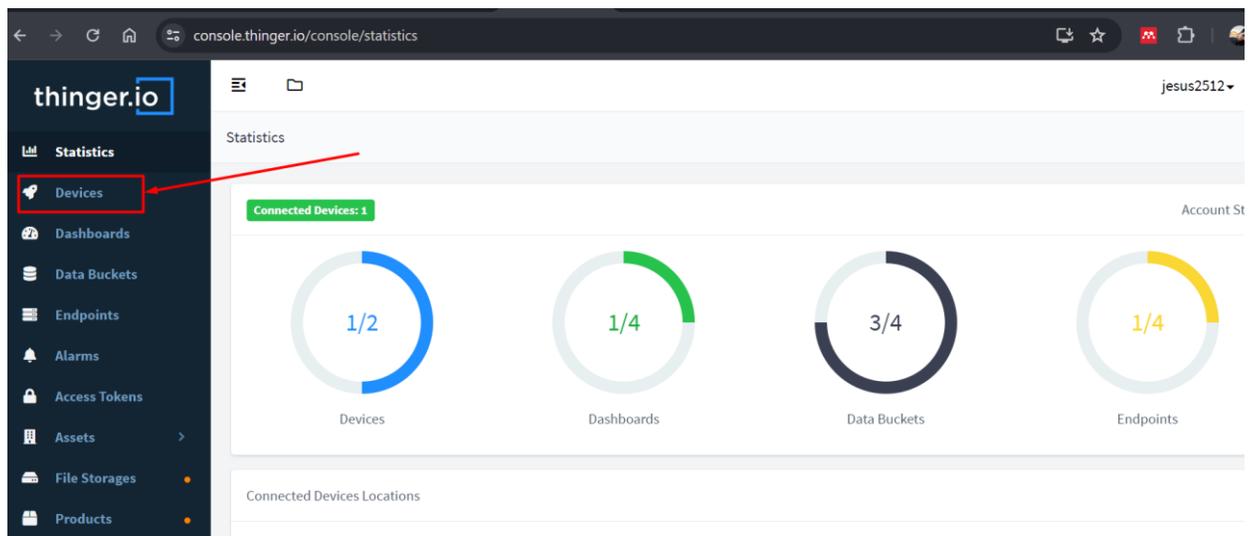
### 3.4.3.1. Dispositivos

En el apartado dispositivos se configuran las credenciales que va a usar el sistema para poder establecer la conexión cliente servidor, los pasos a seguir para la configuración correcta se detallan a continuación.

1. Se accede a la plataforma y en la pantalla principal se accede al apartado de “Devices”.

## Figura 15

*Página principal de Thinger.io.*



2. Dentro del apartado de “Devices”, se elige la opción de “Add Device”.

## Figura 16

*Opción Add Devices.*



3. En este apartado se tienen algunas opciones.

- En Device Type se puede elegir entre varios protocolos compatibles con la plataforma, los cuales incluyen, IOTMP, HTTP, MQTT y NB-IoT, en el caso del proyecto se elige IOTMP.
- En device Id se escribe el nombre del cliente o dispositivo que se quiera colocar.
- Para el apartado de Device credentials, se puede escribir una contraseña elegida por el usuario o se le puede solicitar a la plataforma una contraseña randomica.
- En “Device Name” se coloca el nombre del dispositivo que se quiere que se muestre en la plataforma.
- A continuación, se le coloca una descripción al dispositivo y se da clic en el botón de Add Device.

## Figura 17

*Creación de nuevo dispositivo.*

- El dispositivo debe aparecer creado en la pantalla principal de la sección de “Devices”, si el dispositivo fue configurado correctamente al ejecutar el código el indicador aparecerá en verde, caso contrario, en rojo.

## Figura 18

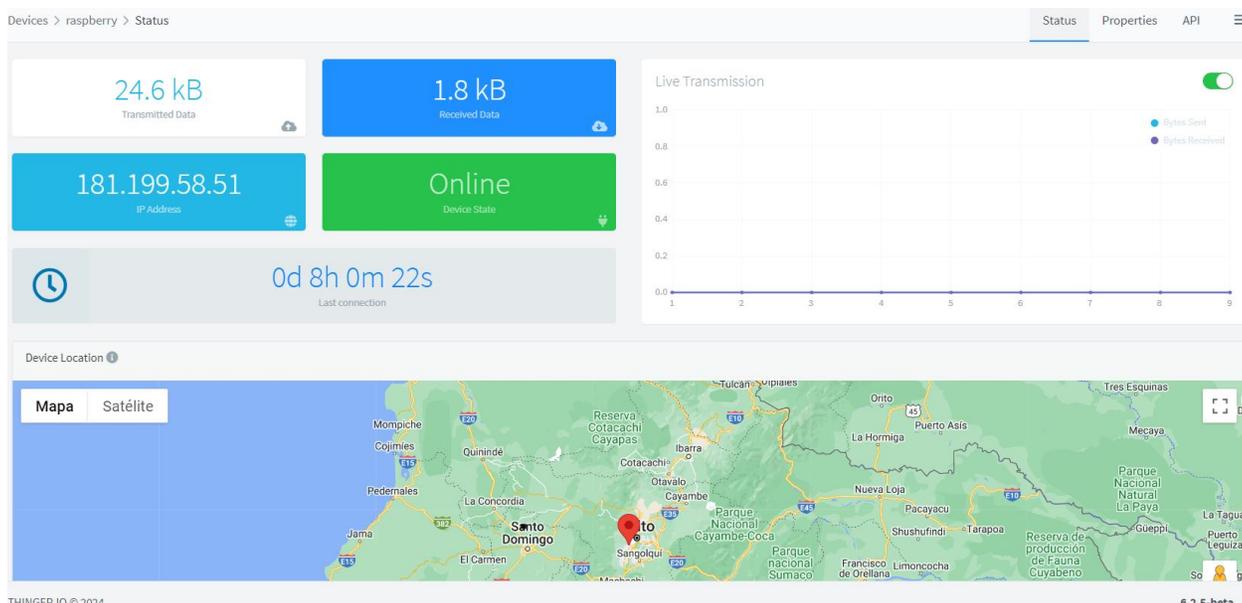
*Dispositivo creado.*

Device	Description	Protocol	Last Connection	Type	Group	Project
● raspberr raspberr	Sistema de alerta temprana	IOTMP	7h			

- Al acceder a este se visualizan dashboards de la cantidad de información que este ha transmitido y unas pestañas donde se pueden realizar configuraciones, si el dispositivo ya está configurado.

## Figura 19

*Página principal del dispositivo.*



6. Los valores transmitidos aparecerán en la pestaña de API.

## Figura 20

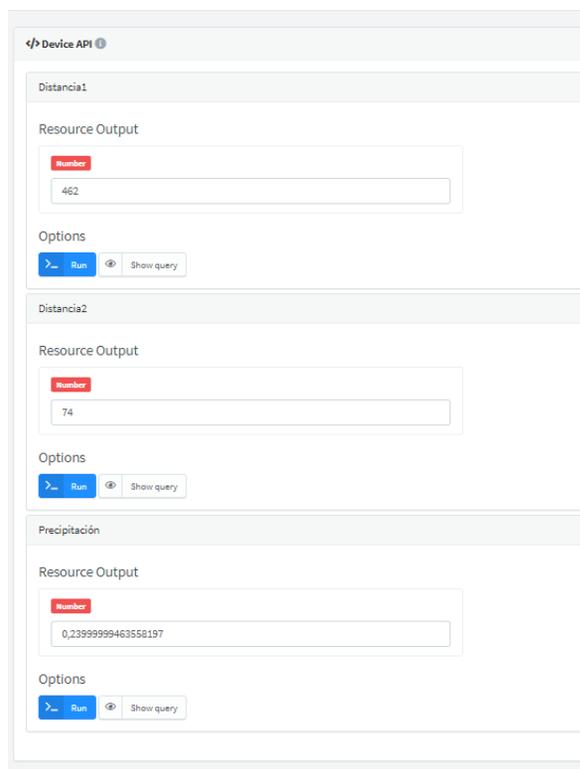
*Menú de APIs*



7. De igual manera estos se pueden abrir para ver más detalles.

**Figura 21**

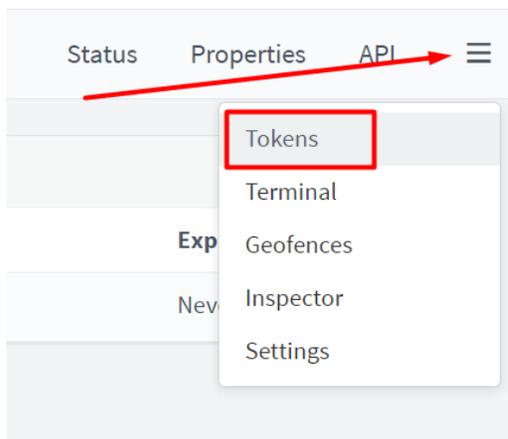
*Datos obtenidos.*



8. Para poder visualizar los datos desde la aplicación móvil se debe crear un token desde el propio apartado de tokens.

**Figura 22**

*Menú de tokens.*



9. Se da clic en el botón de “+Add”

**Figura 23**

*Opción Add.*



10. Para configurar únicamente añadimos un nombre y se le configura los accesos que las personas con el token obtendrán y el tiempo.

**Figura 24**

*Creación de Tokens.*

**Add Device Token**

**Token Name** ⓘ

Sistema de alerta temprana

**Token Access** ⓘ

Allow accessing all device resources with this token

Limit the resource access for this token

**Token Expiration** ⓘ

This token will never expire

Limit the token lifetime on some date

× Cancel Save

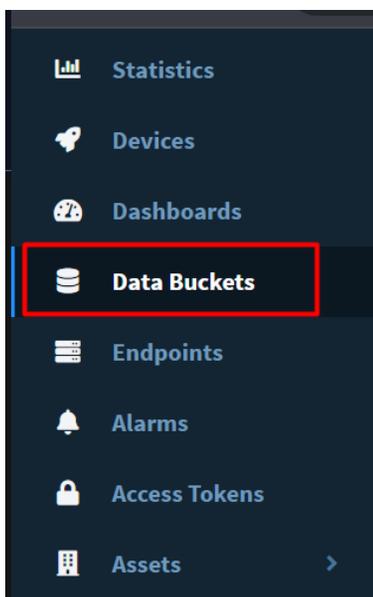
### ***3.4.3.2. Data Buckets***

Thinger.io cuenta con su propio apartado para almacenar información que puede ser útil para futuros análisis y el proceso para configurarlos es el siguiente.

1. Se ingresa al apartado de Data Buckets en el panel de la plataforma.

**Figura 25**

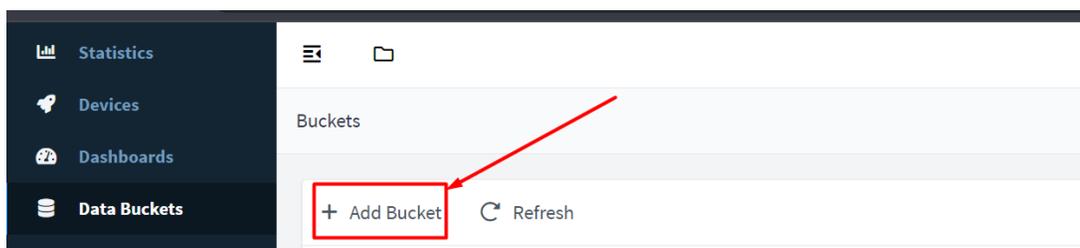
*Opción de Data Buckets.*



2. Se da clic en Add Bucket.

**Figura 26**

*Opción Add Bucket.*



3. Dentro de este apartado se tienen varios campos que deben ser llenados.

- En el campo de Bucket Id se coloca el nombre que tendrá esta base.
- Para el Bucket Name se elige el nombre que se mostrará en la pantalla principal.

- En el siguiente campo se coloca una descripción detallada de lo que se trata la información que se va a almacenar.
- El campo de data source se elige de donde se van a tomar los datos, en este caso se elige From Device Resource, esta opción habilitará más campos donde se puede elegir el dispositivo y el recurso de este.
- Otro campo que se habilita es el de Refresh mode, en este caso se pueden elegir varias formas para el modo en el que se toman los datos, para el proyecto se elige la opción de sampling cada un minuto. Por último se da clic en el botón de Add bucket.

## Figura 27

### Creación de Bucket.

The image shows a web form for creating a bucket. It is organized into several sections:

- Bucket Information:** Includes fields for 'Bucket Id' (placeholder: 'Enter a bucket id'), 'Bucket Name' (placeholder: 'Bucket name'), and 'Bucket Description' (placeholder: 'Optional bucket description').
- Bucket Configuration:** Includes a toggle for 'Enabled' (checked), a 'Data Source' dropdown (selected: 'From Device Resource'), a 'Select Device' dropdown (selected: 'raspberry'), a 'Select Resource' dropdown (selected: 'Distancia1'), and a 'Refresh Mode' dropdown (selected: 'Sampling Interval' with a value of '1').
- Advanced Options:** Includes three checkboxes for 'Asset Type', 'Asset Group', and 'Product', each with a 'Select...' dropdown.

At the bottom of the form is a blue button with a checkmark and the text 'Add Bucket'.

- Una vez creados los buckets se podrán visualizar en la pantalla principal de la siguiente manera.

## Figura 28

*Buckets creados.*

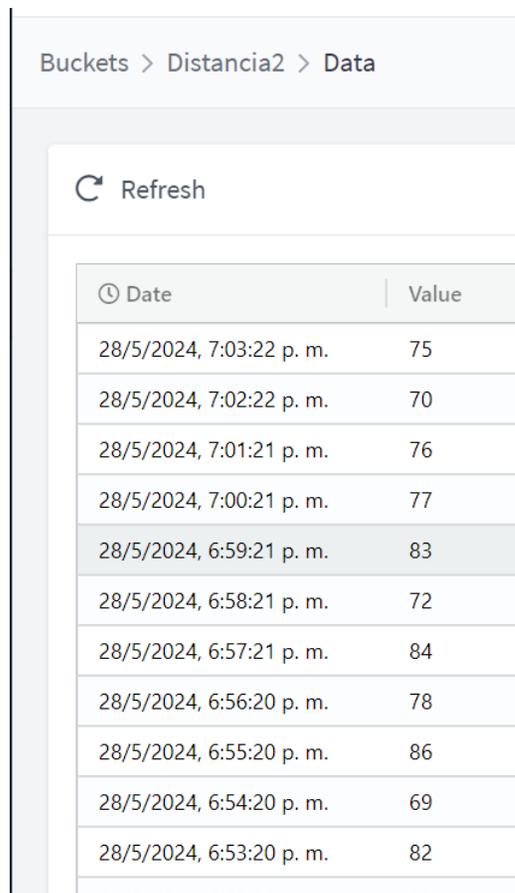


The screenshot displays a user interface for managing buckets. At the top, there are two buttons: '+ Add Bucket' and 'Refresh'. A search bar labeled 'Search Bucket' is positioned on the right. Below these elements is a table with the following columns: 'Bucket', 'Description', 'State', 'Type', 'Group', and 'Project'. The table contains three entries, each with a checkbox on the left and a 'Normal' status indicator on the right.

<input type="checkbox"/>	Bucket	Description	State	Type	Group	Project
<input type="checkbox"/>	<b>Precipitacion</b> Precipitacion	Medida de precipitación en milímetros	✓ Normal			
<input type="checkbox"/>	<b>Sensor 2</b> Distancia2	Valores del sensor 2	✓ Normal			
<input type="checkbox"/>	<b>Sensor 1</b> Distancia1	valores del sensor 1	✓ Normal			

Showing 3 buckets

- Se podrá ingresar a cada uno de ellos para visualizar la información.

**Figura 29***Información de bucket.*

The screenshot shows a data table interface with a breadcrumb path 'Buckets > Distancia2 > Data' at the top. Below the path is a 'Refresh' button with a circular arrow icon. The table has two columns: 'Date' and 'Value'. The data rows show a sequence of timestamps and corresponding values, with the row for '28/5/2024, 6:59:21 p. m.' highlighted in grey.

Date	Value
28/5/2024, 7:03:22 p. m.	75
28/5/2024, 7:02:22 p. m.	70
28/5/2024, 7:01:21 p. m.	76
28/5/2024, 7:00:21 p. m.	77
28/5/2024, 6:59:21 p. m.	83
28/5/2024, 6:58:21 p. m.	72
28/5/2024, 6:57:21 p. m.	84
28/5/2024, 6:56:20 p. m.	78
28/5/2024, 6:55:20 p. m.	86
28/5/2024, 6:54:20 p. m.	69
28/5/2024, 6:53:20 p. m.	82

6. De igual manera se puede exportar para hacer un análisis de los datos a futuro.

### Figura 30

#### *Exportación de bucket.*

Buckets > Distancia2 > Export

Data Import **Export** Clear

Bucket Data Export

**Export Format** ⓘ CSV (Comma Separated Values) ▼

**Timestamp** ⓘ ISO Date ▼

**Export Range** ⓘ  Export all data in bucket  
 Specify a custom data range for export the data

**Callback** ⓘ None ▼

Export Data

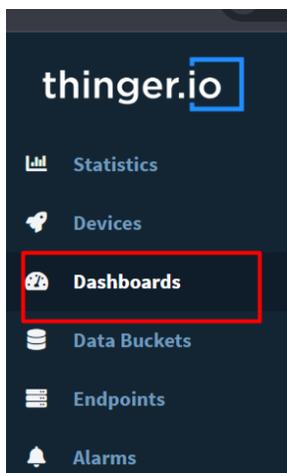
#### **3.4.3.3. Dashboards**

Una vez obtenidos y almacenados los datos, estos se pueden representar gráficamente mediante widgets que estarán en la zona de "Dashboards". Thingiverse.io ofrece una gran variedad de tipos de gráficas y widgets, lo que permite visualizar los datos de manera intuitiva y en tiempo real. Los usuarios pueden personalizar los dashboards para adaptarlos a sus necesidades específicas, creando gráficos de líneas, barras, medidores y más, para monitorear y analizar los datos recolectados de forma eficaz. Esta capacidad de visualización facilita la interpretación de la información y la toma de decisiones basada en datos precisos y actualizados.

1. En el menú lateral se elige la opción de Dashboards para acceder a estos.

**Figura 31**

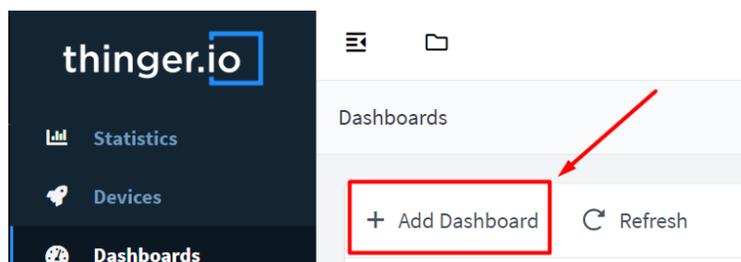
*Opción de Dashboards.*



2. Dentro de este elegimos la opción de Add Dashboard

**Figura 32**

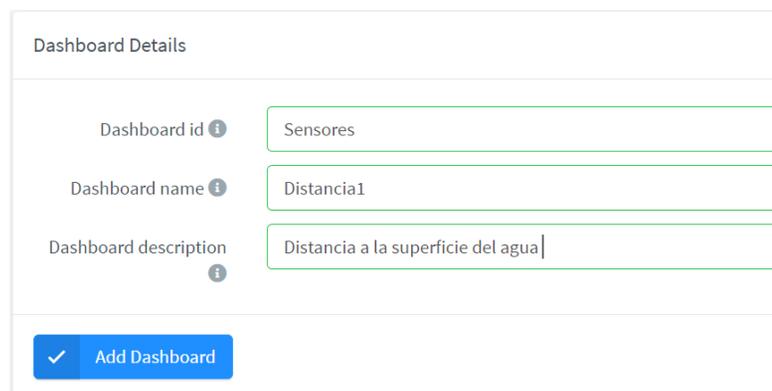
*Opción Add Dashboard.*



3. Para añadir un dashboard únicamente se debe registrar un ID, un nombre y una descripción.

**Figura 33**

*Creación de dashboard.*



Dashboard Details

Dashboard id ⓘ

Dashboard name ⓘ

Dashboard description ⓘ

Add Dashboard

4. Ya agregado el dashboard se lo podrá visualizar en la pantalla principal.

**Figura 34**

*Dashboard creado.*



Dashboards

+ Add Dashboard Refresh Search Dashboard

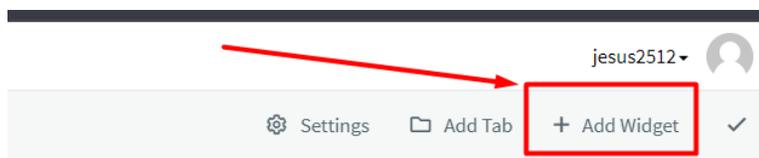
Dashboard	Description	Created	Modified	Project
<input type="checkbox"/> Sensores Distancia1	Distancia a la superficie desde los sensores	16d	2d	

Showing 1 dashboard

5. Dentro del dashboard se da clic en el botón de Add widget para ir agregando elementos a la página.

## Figura 35

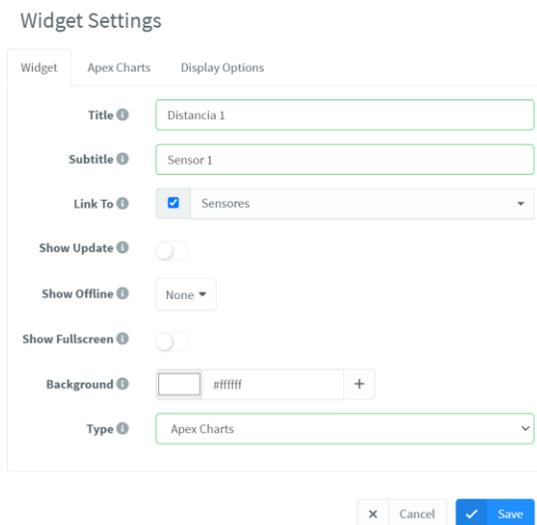
Opción Add Widget.



6. En la página que se despliega se deben agregar el título del widget, el subtítulo, a que datos va a estar atado y el tipo de gráfica que se desea.

## Figura 36

Creación de widget.

A screenshot of a 'Widget Settings' dialog box. The dialog has three tabs: 'Widget', 'Apex Charts', and 'Display Options'. The 'Apex Charts' tab is selected. The settings are as follows: 'Title' is 'Distancia 1', 'Subtitle' is 'Sensor 1', 'Link To' is 'Sensores' (checked), 'Show Update' is a disabled toggle, 'Show Offline' is 'None', 'Show Fullscreen' is a disabled toggle, 'Background' is '#ffffff', and 'Type' is 'Apex Charts'. At the bottom, there are 'Cancel' and 'Save' buttons.

7. Al elegir la opción de Apex charts automáticamente se abrirá una pestaña con las configuraciones de esta gráfica, en este caso se elige de donde se desea que se tomen los datos, en este caso se pueden tomar con un muestreo de los dispositivos o mediante un bucket. Para el proyecto se tomaron los datos del sensor 1 con un muestreo cada 10 segundos en intervalos de 30 minutos.

## Figura 37

Configuración de widget.

The screenshot displays the configuration panel for a widget. At the top, there are tabs for 'Widget', 'Apex Charts', and 'Display Options'. The 'Apex Charts' tab is active. Below it, a section for 'Source 1' is shown with the following settings:

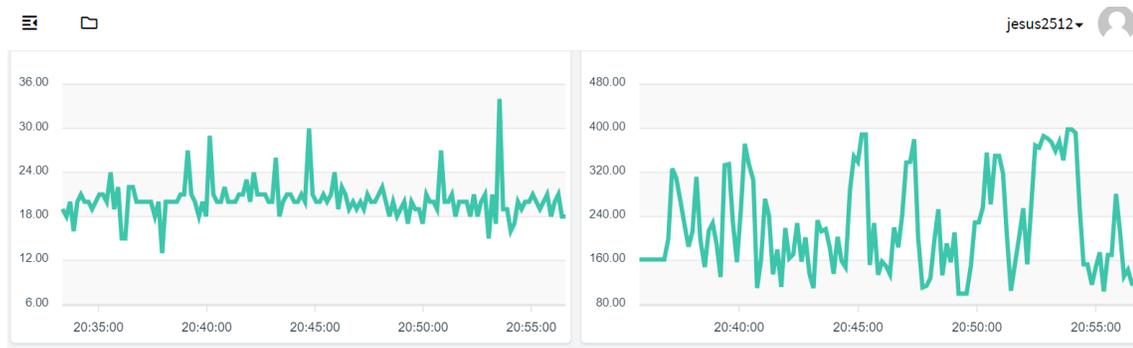
- Name:** Source 1
- Data Source:** From Device Resource
- Select Device:** raspberry
- Select Resource:** Distancia1
- Refresh Mode:** Sampling Interval: 1 minutes
- Time Period:** 30 minutes

At the bottom of the configuration panel, there are three buttons: '+ Add Source', 'Clone Source', and 'Remove Source'.

- En la página principal del dashboard se logra visualizar las gráficas generadas por los sensores.

## Figura 38

Widgets creados.



### 3.4.4. Etapa de difusión de la información

En esta etapa se configuran los endpoints de thinger.io para la difusión de la información y la creación de las alertas para que los usuarios finales puedan monitorear los niveles del río y reciban la alarma lo más pronto posible.

Para esto thinger.io cuenta con una gran variedad de endpoints incluyendo correo electrónico, HTTP request, thinger device call, Telegram, etc. Cada uno de ellos tiene sus propiedades y se detallan a continuación.

### Figura 39

*Comparación entre tipos de endpoints.*

Endpoint	Ventajas	Desventajas	Escenario Ideal
Email	Fácil de configurar. Amplia disponibilidad	Puede ser lento para alertas críticas. Requiere acceso al correo electrónico	Alertas no críticas. Informes periódicos
HTTP Request	Muy flexible. Integración con otros sistemas y servicios web	Requiere configurar un servidor o API receptor	Integración con sistemas web o APIs
IFTTT Webhook Trigger	Amplia integración con diferentes servicios. Fácil configuración de applets	Puede haber algún retraso. Necesidad de crear y gestionar applets en IFTTT	Integración con múltiples servicios
Telegram Send Message	Rápido y fiable. Ideal para notificaciones instantáneas	Requiere configurar un bot de Telegram. Necesita obtener el chat ID	Alertas críticas e instantáneas
Thingier.io Device Call	Control directo de otros dispositivos IoT en el ecosistema Thingier.io. No requiere servicios externos	Limitado a dispositivos registrados en Thingier.io	Control de dispositivos IoT asociados
Telegram message	Similar a Telegram Send Message. Útil para notificaciones rápidas y seguras	Requiere configuración del bot de Telegram	Notificaciones rápidas y seguras

Para el sistema se eligió la aplicación de Telegram para las alertas y thingier.io app para el monitoreo de los sensores, esto ya que Telegram ofrece rapidez, fiabilidad, y seguridad en las notificaciones instantáneas, con fácil configuración y disponibilidad multiplataforma. Por otro lado, Thingier.io proporciona una plataforma integrada y amigable para la gestión y visualización

de datos de dispositivos IoT, con herramientas avanzadas para gráficos y control remoto, facilitando el monitoreo en tiempo real y la gestión de los sensores.

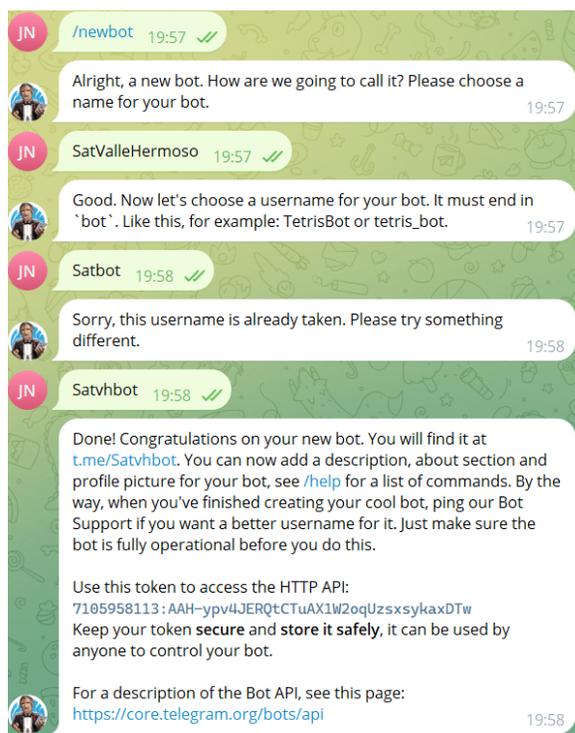
#### 3.4.4.1. Telegram

La sencillez para configurar un chatbot y la alta escalabilidad que tiene la aplicación de Telegram sumada a su integración con las herramientas ya establecidas en el proyecto la volvió el medio de difusión idóneo para la difusión de la alerta. Los pasos para la integración de la aplicación con el sistema se detallan a continuación.

1. Se crea un chatbot mediante la herramienta botfather, únicamente se debe iniciar un chat con el para que este automáticamente vaya guiando al usuario, lo único que el usuario debe proveer es el nombre del chatbot que desea crear.

### Figura 40

*Creación de un bot de telegram.*



2. Con el token se ingresa al siguiente link:

[https://api.telegram.org/bot<YOUR\\_BOT\\_TOKEN>/getUpdates](https://api.telegram.org/bot<YOUR_BOT_TOKEN>/getUpdates)

Donde se encontrará el id del chat de Telegram que está sincronizado con el bot.

## Figura 41

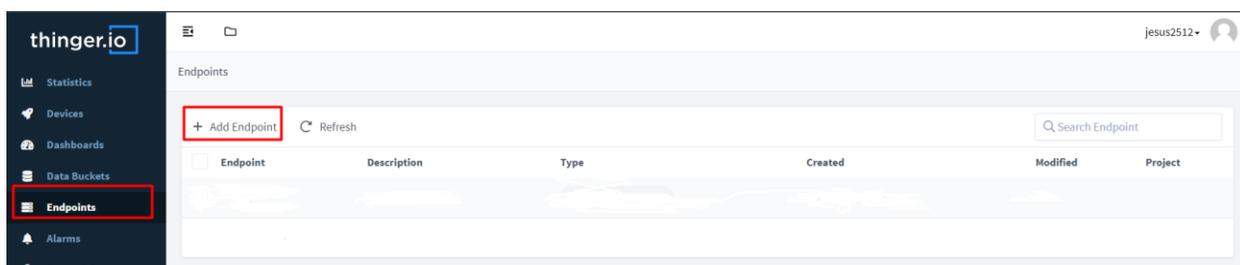
*Obtención del ID del chat.*

```
{
  "ok": true,
  "result": [
    {
      "update_id": 436681279,
      "message": {
        "message_id": 3,
        "from": {
          "id": 847568917,
          "is_bot": false,
          "first_name": "Jes\u00fas",
          "last_name": "Navarrete",
          "username": "h305073256",
          "language_code": "es",
          "chat": {
            "id": 847568917,
            "first_name": "Jes\u00fas",
            "last_name": "Navarrete",
            "username": "h305073256",
            "type": "private",
            "date": 1718153646,
            "text": "/start",
            "entities": [
              {
                "offset": 0,
                "length": 6,
                "type": "bot_command"
              }
            ]
          },
          "update_id": 436681280,
          "channel_post": {
            "message_id": 37908,
            "sender_chat": {
              "id": -1002008262746,
              "title": "SatVH",
              "type": "channel",
              "chat": {
                "id": -1002008262746,
                "title": "SatVH",
                "type": "channel",
                "date": 1718239101,
                "text": "1"
              },
              "update_id": 436681281,
              "message": {
                "message_id": 37986,
                "from": {
                  "id": 777800,
                  "is_bot": false,
                  "first_name": "Telegram",
                  "sender_chat": {
                    "id": -1002008262746,
                    "title": "SatVH",
                    "type": "channel",
                    "chat": {
                      "id": -1002015119736,
                      "title": "SatVhchat",
                      "username": "SatVhchat",
                      "type": "supergroup",
                      "date": 1718239104,
                      "forward_origin": {
                        "type": "channel",
                        "chat": {
                          "id": -1002008262746,
                          "title": "SatVH",
                          "type": "channel",
                          "message_id": 37908,
                          "date": 1718239101,
                          "is_automatic_forward": true,
                          "forward_from_chat": {
                            "id": -1002008262746,
                            "title": "SatVH",
                            "type": "channel",
                            "forward_from_message_id": 37908,
                            "forward_date": 1718239101,
                            "text": "1"
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      ]
    }
  ]
}
```

3. Una vez ya se tienen los datos de Telegram se puede configurar el endpoint en thinger.io en la pestaña de endpoints, pulsando el botón Add Endpoint.

## Figura 42

*Opción Endpoints.*



4. Para crear el endpoint se debe colocar un identificador para la programación y un nombre, en este caso ambos serán “Telegram”. Se coloca una descripción, se elige Telegram send message como Endpoint type y por ultimo se colocan los identificadores del chatbot junto al mensaje que se desea enviar.

### Figura 43

#### Configuración de Endpoint.

Endpoint Details

Endpoint Identifier <i>i</i>	Telegram
Endpoint Name <i>i</i>	Telegram
Endpoint Description <i>i</i>	Alerta de crecimiento del nivel del agua
Enabled <i>i</i>	<input checked="" type="checkbox"/>
Endpoint Type <i>i</i>	Telegram Send Message
Bot Token <i>i</i>	7105958113:AAH-ypv4JERQtCTuAX1W2oqUzsxxykaxDTw
Chat Identifier <i>i</i>	-1002008262746
Chat Message <i>i</i>	Alerta! Crecimiento anormal del nivel del agua

Add Endpoint

Con esto ya quedaría configurado el endpoint y bastaría con realizar un llamado desde la programación del sistema.

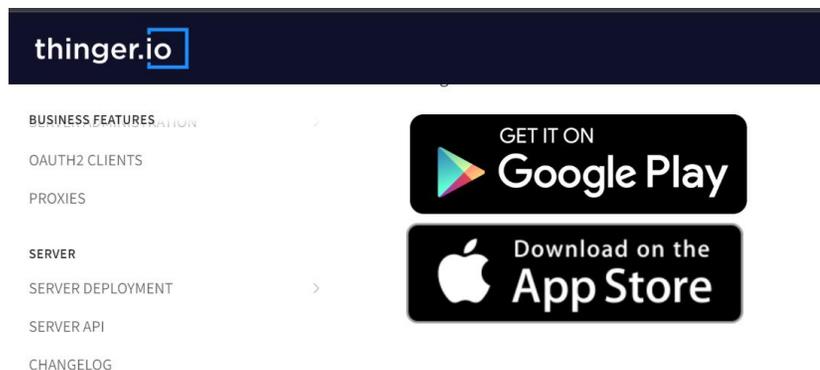
#### 3.4.4.2. Aplicación de *Thinger.io*

La aplicación de *thinger.io* es una herramienta sencilla que ofrece la plataforma desde la cual se pueden hacer consultas de los datos en tiempo real y poder revisar dashboards, esta aplicación no permite editar configuraciones lo que lo vuelve perfecto para que las personas interesadas puedan monitorear los datos proporcionados por los sensores. La configuración se presenta a continuación.

1. Se puede descargar tanto para IOS como para Android sin embargo para Android se necesitará de un apk para su funcionamiento.

## Figura 44

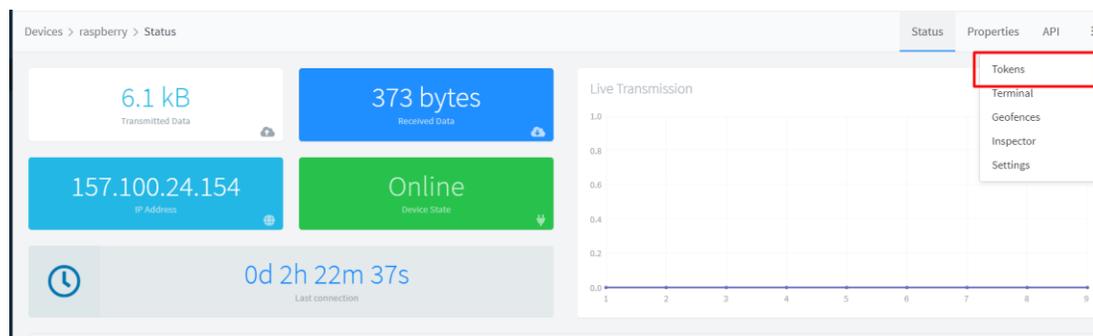
*Plataformas soportadas por la aplicación.*



2. En la plataforma de thinger.io se accede al área de dispositivos y en ella se accede a la opción de tokens.

## Figura 45

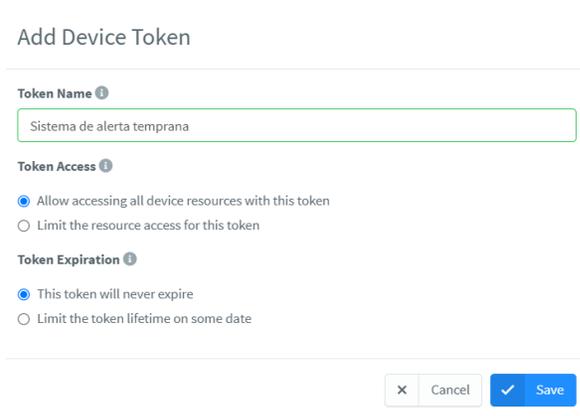
*Opción de tokens.*



3. Ya en el área de tokens se da clic en el botón de “+Add” para agregar uno nuevo.

**Figura 46**

*Creación de tokens.*

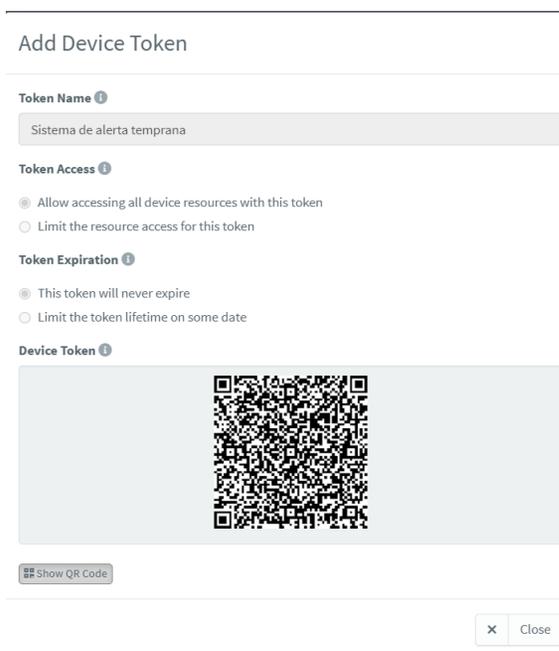


The screenshot shows a web form titled "Add Device Token". It contains three sections: "Token Name" with a text input field containing "Sistema de alerta temprana"; "Token Access" with two radio button options, the first of which is selected; and "Token Expiration" with two radio button options, the first of which is selected. At the bottom right, there are three buttons: "x Cancel", "✓ Save", and a blue "Save" button.

4. Ya teniendo el token creado se accede a él para poder obtener el link o el código QR para poder acceder desde la aplicación.

**Figura 47**

*Obtención del token.*



The screenshot shows the same "Add Device Token" form, but now it displays a QR code under the "Device Token" section. The "Token Name" field is now disabled and greyed out. The "Token Access" and "Token Expiration" sections remain the same. At the bottom right, there is a "x Close" button. A "Show QR Code" button is located below the QR code.

5. Dentro de la aplicación se elige la opción de devices en la parte inferior derecha.

### Figura 48

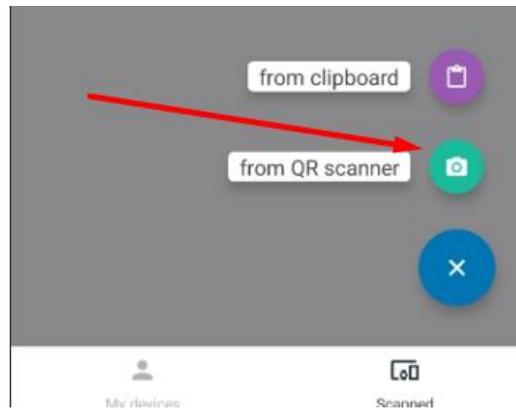
*Inicio de la aplicación de thinger.io.*



6. En devices se pulsa el botón de + para que se despliegue el menú donde se podrá elegir el método que se quiere para acceder al token, en este caso se elige la opción de “from QR scanner” .

**Figura 49**

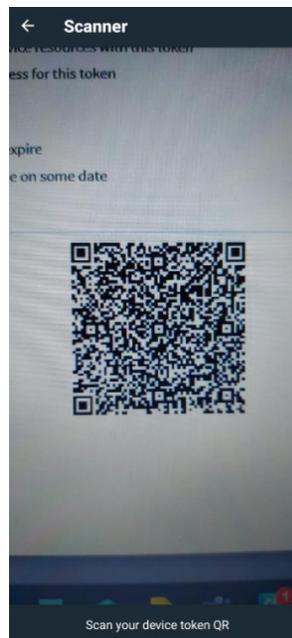
*Opción Qr scanner.*



7. Al seleccionar esta opción se abrirá automáticamente la cámara, acto seguido se debe apuntar esta al código QR del Token generado.

**Figura 50**

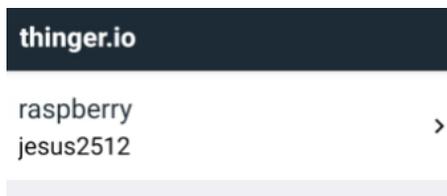
*Escaneo de QR.*



8. Una vez escaneado el dispositivo aparecerá automáticamente en la página principal.

### Figura 51

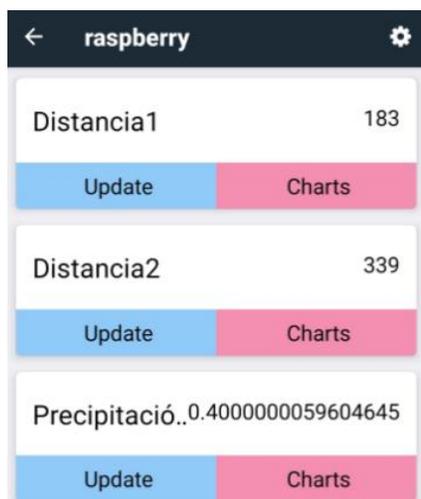
*Dispositivo al que se accedió mediante el token.*



9. Al acceder al dispositivo se obtendrán los datos proporcionados por este en tiempo real.

### Figura 52

*Datos de los sensores.*



#### 3.4.5. Sistema final

La arquitectura final del sistema queda de la siguiente manera:

El sistema se compone de dos módulos emisores que se encargan de leer las variables físicas del nivel del río mediante un sensor ultrasónico JSN-SR04T conectado a un microcontrolador ESP32,

el cual es responsable de transmitir los datos mediante el módulo LORA RYLR998, comunicándose mediante comandos AT enviados por el puerto serial.

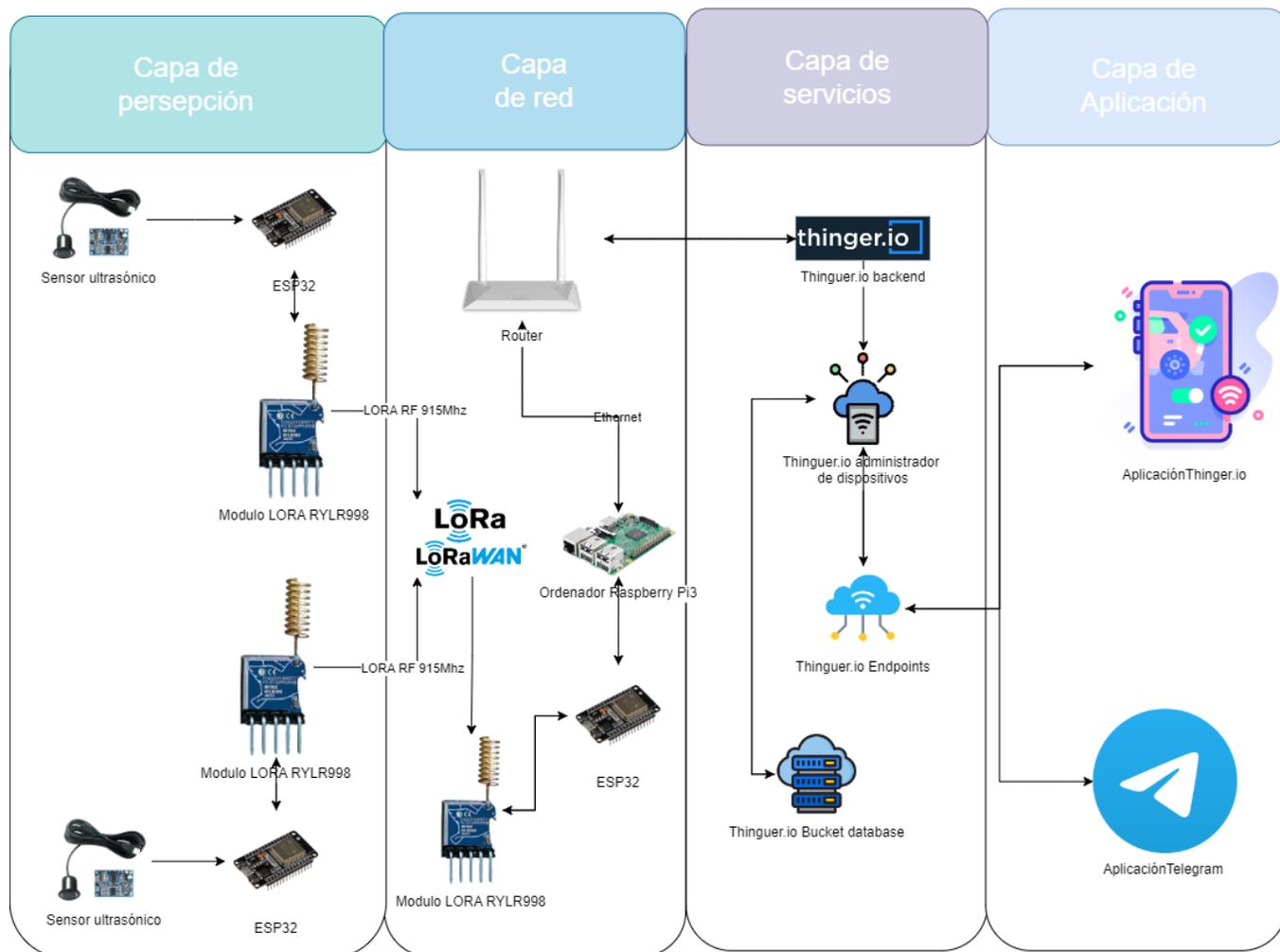
Un módulo receptor, que posee los mismos componentes que los módulos emisores, a excepción del sensor. Este módulo se conecta a una Raspberry Pi 3, que se encarga del procesamiento de los datos y de enviarlos a través de una conexión a internet proporcionada por un router.

Los datos se envían a la plataforma IoT Thinger.io mediante el protocolo MQTT. Esta plataforma es la encargada de almacenar los datos y enviar los mensajes de alerta cuando el computador detecta niveles anormales en las mediciones.

Por último, las alertas se envían a través de la aplicación de Telegram, configurada como endpoint en la plataforma Thinger.io. A su vez, la plataforma cuenta con su propia aplicación, que funciona como una estación de monitoreo portátil desde la cual se pueden visualizar los datos proporcionados.

**Figura 53**

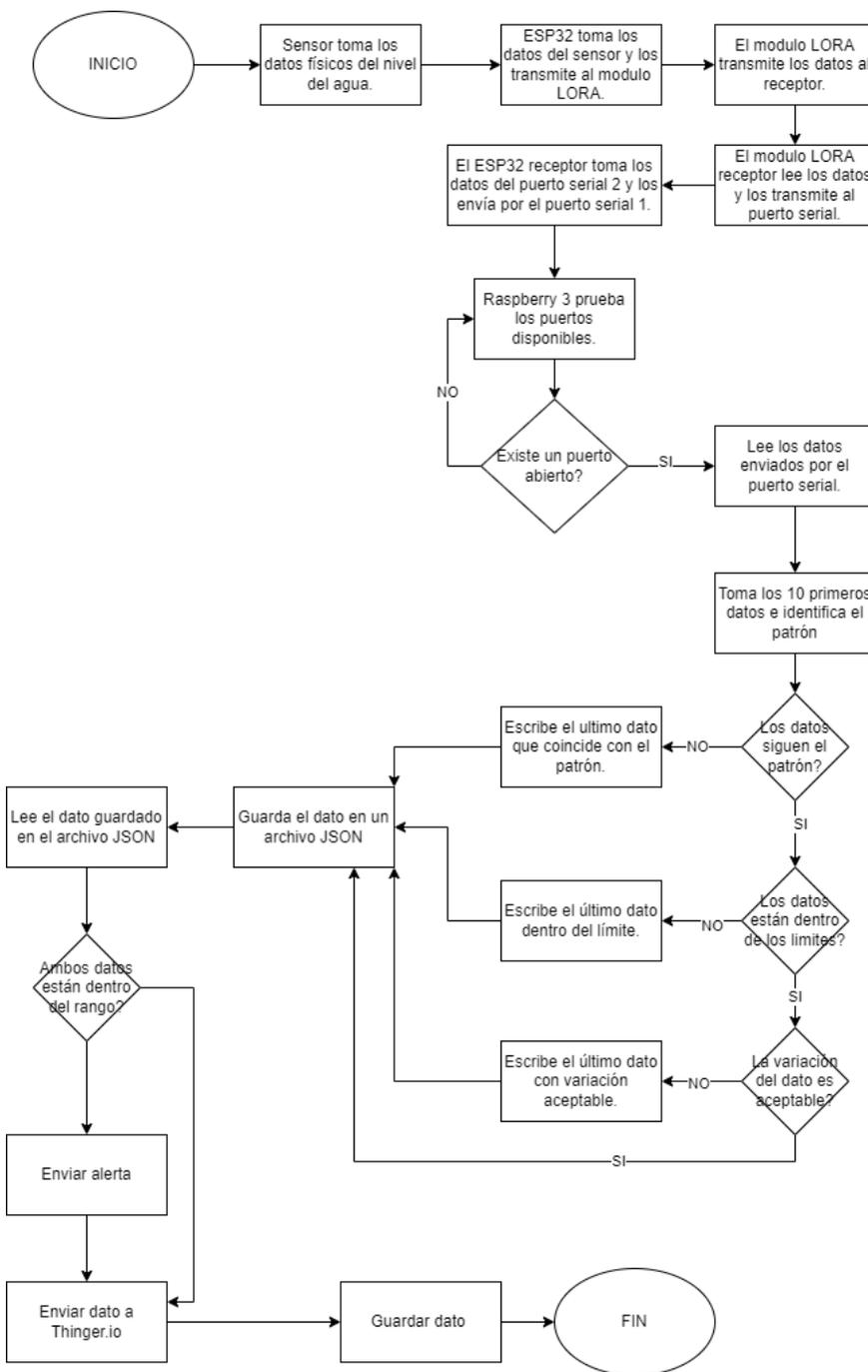
*Arquitectura IOT.*



Para un mayor entendimiento se proporciona un diagrama de flujo el cual se visualiza en la siguiente figura.

**Figura 54**

*Diagrama de flujo.*



### 3.5. Evaluación de la Situación, Difusión de la Alerta y Plan de Emergencia

Para evaluar la situación, se realizó un acercamiento con tres partes interesadas: la empresa de turismo Green Kayak School, con sede en Valle Hermoso; la Cruz Roja, que es la institución con mayor facilidad de acceso en casos de desastres en la zona; y finalmente la televisora Zaracay Televisión, la cual mostró interés en realizar un reportaje sobre el proyecto.

Además, durante estas reuniones, se discutieron los objetivos del proyecto, se identificaron posibles colaboraciones y se establecieron canales de comunicación para una respuesta rápida y eficaz en caso de emergencia. Green Kayak School se comprometió a apoyar con recursos logísticos, la Cruz Roja ofreció su experiencia y equipos para el manejo de emergencias, y Zaracay Televisión proporcionará cobertura mediática para sensibilizar a la comunidad y mantener informada a la población.

#### Figura 55

Planificación de acciones en conjunto.



## Capítulo IV: Resultados y análisis

### 4.1. Instalación de los dispositivos.

El primer sensor (RVC1), se instaló en un brazo del río principal dentro de la propiedad del matrimonio de Carmen Cerón y Hernán Ríos, quienes proporcionaron tanto ayuda logística como mano de obra y acceso a internet. Este dispositivo se instaló a 350 cm del fondo del río, quedando con una medición de nivel de agua normal de 20 cm con variaciones de 15cm debido al mínimo oleaje que se produce en este sector y establecido para que envíe la alerta cuando el nivel supere los 200 cm.

#### Figura 56

*Instalación del sensor 1.*

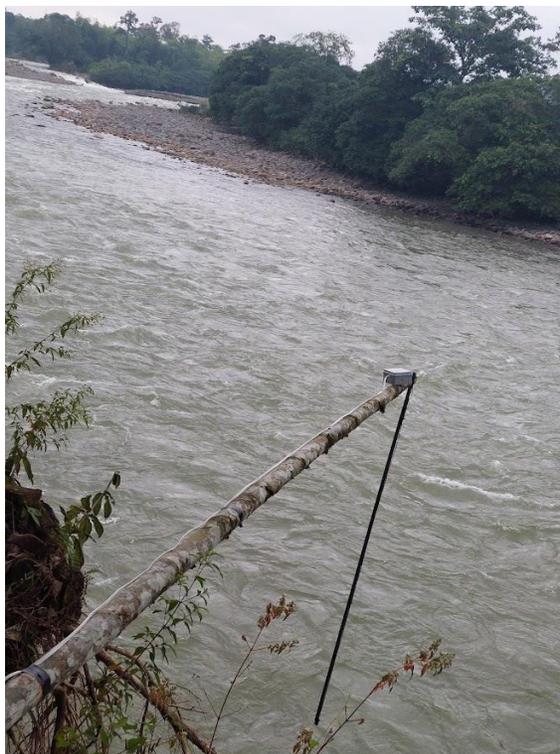


El segundo sensor (RVC3) se instala en la propiedad del señor Marco Carrillo, la cual cuenta con un acantilado que bordea el cauce principal del río Blanco. Este dispositivo se instaló a una altura significativamente mayor que el primer sensor, situándose a una altura de 500 cm sobre el nivel del río. En condiciones normales, este sensor mide un nivel de agua de 250 cm.

Este sensor está diseñado para detectar el crecimiento del nivel del agua y activar alertas cuando el nivel del agua supera los 450 cm. Sin embargo, debido al comportamiento del río en este sector, que incluye una curva pronunciada, las mediciones de este sensor pueden variar hasta 1 metro. Esta variabilidad se debe a la dinámica del flujo de agua en la curva, lo que puede generar fluctuaciones en las lecturas del sensor.

### **Figura 57**

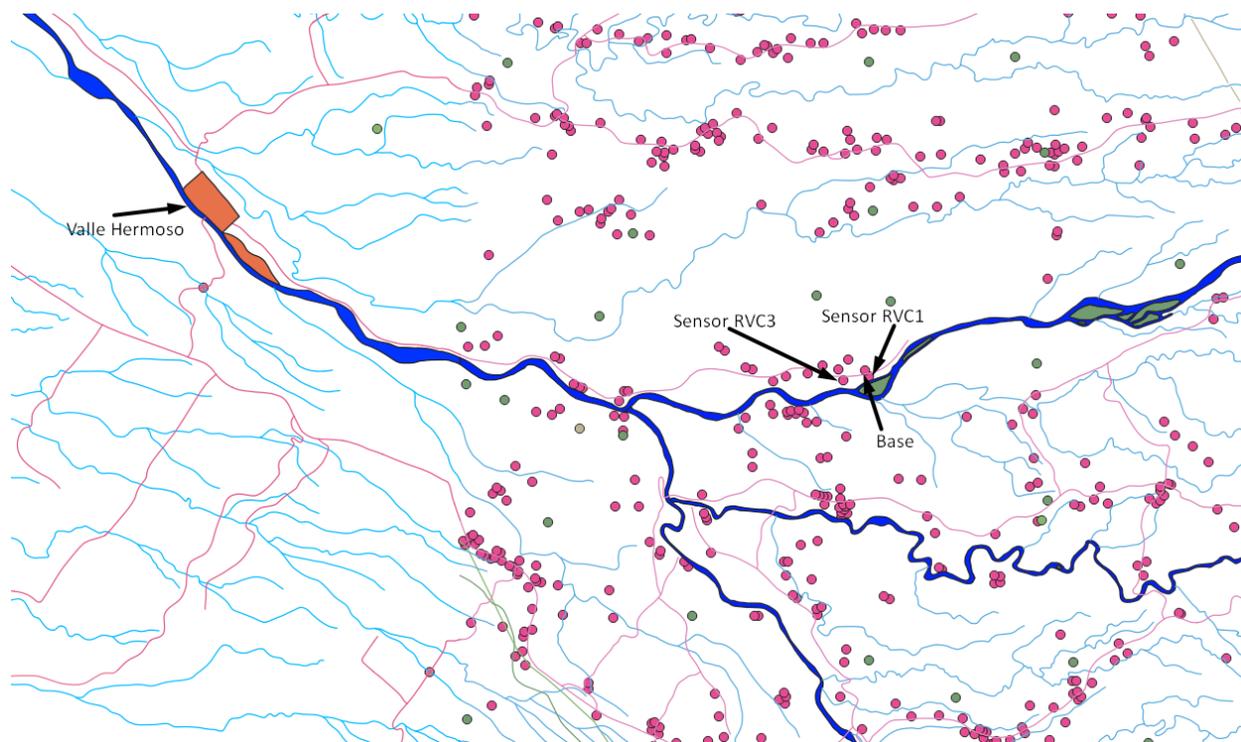
*Instalación del sensor 2.*



La base se instaló en la vivienda de la propiedad donde se instaló el primer sensor ya que los propietarios proporcionaron conexión a internet para el proyecto. La vivienda se encuentra a 100 metros del primer sensor y a 500 metros del segundo.

### Figura 58

*Posicionamiento de sensores.*



### 4.2. Transmisión de datos

Para realizar el análisis de transmisión de datos se usa como guía el manual de uso del módulo LORA RYLR998 el cual da la siguiente información:

- <Address>: El ID de la dirección del transmisor. Este es el identificador del dispositivo que envía los datos.
- <Length>: La longitud de los datos. Esto indica la cantidad de caracteres en el mensaje de datos recibido.

- <Data>: Los datos en formato ASCII. Este es el mensaje o la información que el transmisor está enviando.
- <RSSI>: El indicador de la intensidad de la señal recibida (Received Signal Strength Indicator).
- <SNR>: La relación señal-ruido (Signal-to-Noise Ratio).

A continuación, se toma una muestra de los resultados obtenidos en la base.

### Figura 59

*Captura de datos de ejemplo.*

```
+RCV=1, 2, 18, -101, -2
+RCV=3, 3, 330, -102, -6
+RCV=1, 2, 18, -101, -1
+RCV=3, 3, 350, -102, -6
+RCV=1, 2, 20, -101, -2
+RCV=3, 3, 325, -102, -5
+RCV=1, 2, 21, -101, -2
+RCV=3, 3, 340, -103, -6
+RCV=1, 2, 23, -100, -2
+RCV=3, 3, 330, -103, -7
+RCV=1, 2, 18, -100, -2
+RCV=1, 2, 21, -99, -2
+RCV=1, 2, 18, -101, -2
```

La intensidad mínima para una correcta comunicación del transmisor es de -123dB. En la imagen, se puede observar que el RSSI de ambos sensores, RCV1 y RCV3, están dentro del rango adecuado, presentando valores entre -99dB y -101dB.

En cuanto al SNR, ambos dispositivos muestran una señal de ruido alta, aunque aceptable, lo que permite que la comunicación no se pierda, aunque ocasionalmente se reciben datos erróneos. El SNR del sensor RCV3 es peor, ya que este se encuentra a una mayor distancia de la base en comparación con el sensor RCV1.

### Figura 60

*Directorio principal.*

```
satvallehermoso@raspberrypi:~ $ ls
Bookshelf Desktop Imágenes Linux-Client my_project Plantillas sensor_data_rcv1.json tratamiento
Descargas Documentos IOTMP-Linux Música openweather Público sensor_data_rcv3.json Vídeos
```

Para mitigar esto, se instalaron dos servicios. En la imagen se muestra el directorio donde se encuentran ambos servicios y los documentos que comparten. La primera carpeta es la de "tratamiento", que es el directorio al que accede el servicio "sensor". Este servicio se encarga de recibir los datos, separar los datos del resto de la trama, filtrarlos y escribirlos en un archivo JSON que se encuentra en el mismo directorio mostrado en la figura. Estos datos se escriben en dos archivos por separado: uno para el sensor RCV1 y otro para el sensor RCV3.

La segunda carpeta es "Linux-Client" y es el directorio al que accede el servicio "satvh". Este servicio se encarga de leer los archivos escritos por el servicio "sensor", borrar los datos que ya fueron leídos y enviarlos a la nube de Thingier.io. Además, es responsable de activar las alertas cuando los datos del nivel de agua llegan a ser críticos.

Ambos servicios generan logs que permiten monitorear el funcionamiento del sistema y resolver errores cuando estos se presentan. A continuación, se observan los mensajes recolectados por los logs.

**Figura 61**

*Logs del servicio sensor.*

```

jun 22 19:21:08 raspberrypi python3[1193]: 2024-06-22 19:21:08,958 - WARNING - Crecimiento fuera de rango para rcv3: 472 cm
jun 22 19:21:08 raspberrypi python3[1193]: 2024-06-22 19:21:08,960 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}
jun 22 19:21:10 raspberrypi python3[1193]: 2024-06-22 19:21:10,377 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv1.json: {'ti
po_rcv': '1', 'crecimiento': 22, 'timestamp': '2024-06-22T19:21:10.376203'}
jun 22 19:21:11 raspberrypi python3[1193]: 2024-06-22 19:21:11,123 - WARNING - Crecimiento fuera de rango para rcv3: 608 cm
jun 22 19:21:11 raspberrypi python3[1193]: 2024-06-22 19:21:11,125 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}
jun 22 19:21:12 raspberrypi python3[1193]: 2024-06-22 19:21:12,526 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv1.json: {'ti
po_rcv': '1', 'crecimiento': 18, 'timestamp': '2024-06-22T19:21:12.525171'}
jun 22 19:21:13 raspberrypi python3[1193]: 2024-06-22 19:21:13,289 - WARNING - Crecimiento fuera de rango para rcv3: 608 cm
jun 22 19:21:13 raspberrypi python3[1193]: 2024-06-22 19:21:13,290 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}
jun 22 19:21:14 raspberrypi python3[1193]: 2024-06-22 19:21:14,675 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv1.json: {'ti
po_rcv': '1', 'crecimiento': 18, 'timestamp': '2024-06-22T19:21:14.674121'}
jun 22 19:21:15 raspberrypi python3[1193]: 2024-06-22 19:21:15,445 - WARNING - Crecimiento fuera de rango para rcv3: 472 cm
jun 22 19:21:15 raspberrypi python3[1193]: 2024-06-22 19:21:15,447 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}
jun 22 19:21:16 raspberrypi python3[1193]: 2024-06-22 19:21:16,824 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv1.json: {'ti
po_rcv': '1', 'crecimiento': 18, 'timestamp': '2024-06-22T19:21:16.823068'}
jun 22 19:21:17 raspberrypi python3[1193]: 2024-06-22 19:21:17,610 - WARNING - Crecimiento fuera de rango para rcv3: 608 cm
jun 22 19:21:17 raspberrypi python3[1193]: 2024-06-22 19:21:17,612 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}
jun 22 19:21:18 raspberrypi python3[1193]: 2024-06-22 19:21:18,973 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv1.json: {'ti
po_rcv': '1', 'crecimiento': 17, 'timestamp': '2024-06-22T19:21:18.972028'}
jun 22 19:21:19 raspberrypi python3[1193]: 2024-06-22 19:21:19,776 - WARNING - Crecimiento fuera de rango para rcv3: 608 cm
jun 22 19:21:19 raspberrypi python3[1193]: 2024-06-22 19:21:19,777 - INFO - Datos guardados en /home/satvallehermoso/sensor_data_rcv3.json: {'ti
po_rcv': '3', 'crecimiento': 97, 'timestamp': '2024-06-22T19:21:08.802185'}

```

En la imagen se observa el log del servicio "sensor" cuando detecta que uno de los sensores está enviando datos incorrectos y cuando envían datos correctos. En la imagen, los datos incorrectos los envía el sensor RCV3 y los datos correctos el sensor RCV1. El mensaje indica que el sensor 3 está enviando datos fuera de rango, mientras que para el sensor 1 indica que los datos han sido guardados en el directorio indicado.

**Figura 62**

*Logs del servicio satvh.*

```
satvallehermoso@raspberrypi:~ $ sudo journalctl -u satvh.service -f
jun 22 19:24:59 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:00 raspberrypi bash[1001]: Leído /home/satvallehermoso/sensor_data_rcv1.json: 19 cm
jun 22 19:25:00 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:00 raspberrypi bash[1001]: Enviando Distancia1 a Thingier.io: 19 cm
jun 22 19:25:01 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv1.json
jun 22 19:25:01 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:02 raspberrypi bash[1001]: Precipitación en la última hora: 0.510000 mm
jun 22 19:25:02 raspberrypi bash[1001]: Leído /home/satvallehermoso/sensor_data_rcv1.json: 18 cm
jun 22 19:25:02 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:02 raspberrypi bash[1001]: Enviando Distancia1 a Thingier.io: 18 cm
jun 22 19:25:03 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv1.json
jun 22 19:25:03 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:04 raspberrypi bash[1001]: Leído /home/satvallehermoso/sensor_data_rcv1.json: 18 cm
jun 22 19:25:04 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:05 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv1.json
jun 22 19:25:05 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:06 raspberrypi bash[1001]: Leído /home/satvallehermoso/sensor_data_rcv1.json: 18 cm
jun 22 19:25:06 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
jun 22 19:25:07 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv1.json
jun 22 19:25:07 raspberrypi bash[1001]: El archivo JSON está vacío: /home/satvallehermoso/sensor_data_rcv3.json
```

Por su parte, el log del servicio "satvh" indica que está leyendo los archivos JSON y, si uno de ellos no tiene datos, lo señala con el mensaje "Archivo JSON está vacío" seguido del directorio que está leyendo. Por otro lado, si existen datos en el archivo JSON, este indica el valor del dato leído y si ya fue enviado a Thingier.io.

**Figura 63**

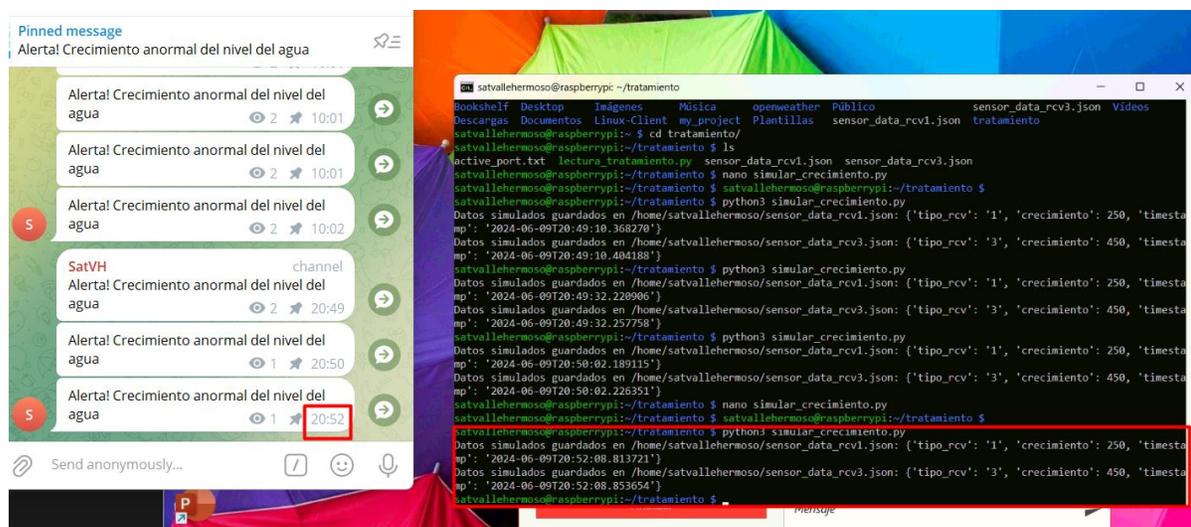
*Archivo JSON.*

```
[
  {
    "tipo_rcv": "3",
    "crecimiento": 241,
    "timestamp": "2024-06-22T19:29:23.820459"
  }
]
```

La figura anterior muestra como el servicio "sensor" imprime un dato en el archivo JSON, este formato ayuda a que ambos servicios puedan entender de mejor manera los datos que se están leyendo.

**Figura 64**

*Simulación de Alerta.*



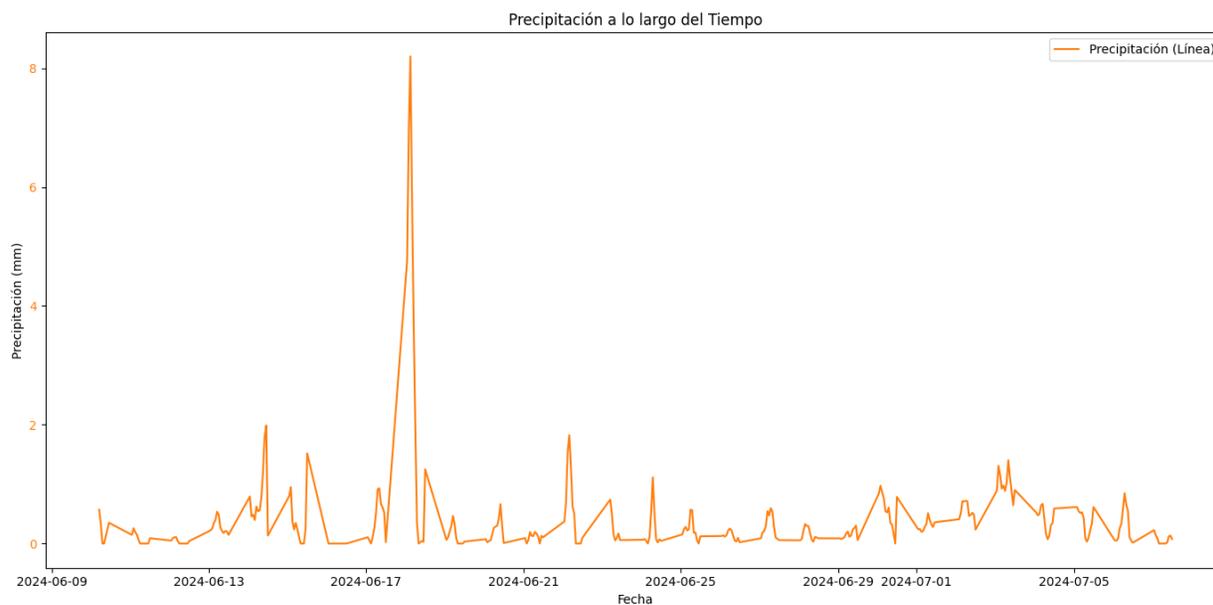
Para realizar una demostración del funcionamiento del sistema se crea el programa “Simular\_crecimiento” el cual escribe valores altos en el archivo JSON lo que obliga al servicio “satvh” a enviar una alerta al Telegram.

Como se observa en la imagen se envía la simulación a las 20:52 y el mensaje de telegram es inmediata por lo que el sistema no ocupa tiempo adicional lo cual es primordial en un sistema de alerta temprana.

### 4.3. Análisis de datos en temporada de riesgo alto.

**Figura 65**

*Grafica de precipitaciones de Junio.*



Para comenzar el análisis del comportamiento del nivel del agua en relación con los datos de precipitación, se observa el comportamiento del río en el mes de junio. Según el Instituto Nacional de Meteorología e Hidrología, la temporada de lluvias se ha extendido hasta junio, siendo julio el comienzo de la época seca.

En la gráfica se pueden observar los datos de precipitación del mes de junio, donde se visualiza un pico de precipitación intensa el día 18 de junio. Sin embargo, esta precipitación se dio en una zona que no afecta al crecimiento del río. A lo largo del mes, se registraron otras precipitaciones que sí influyeron en la subida del nivel del agua.

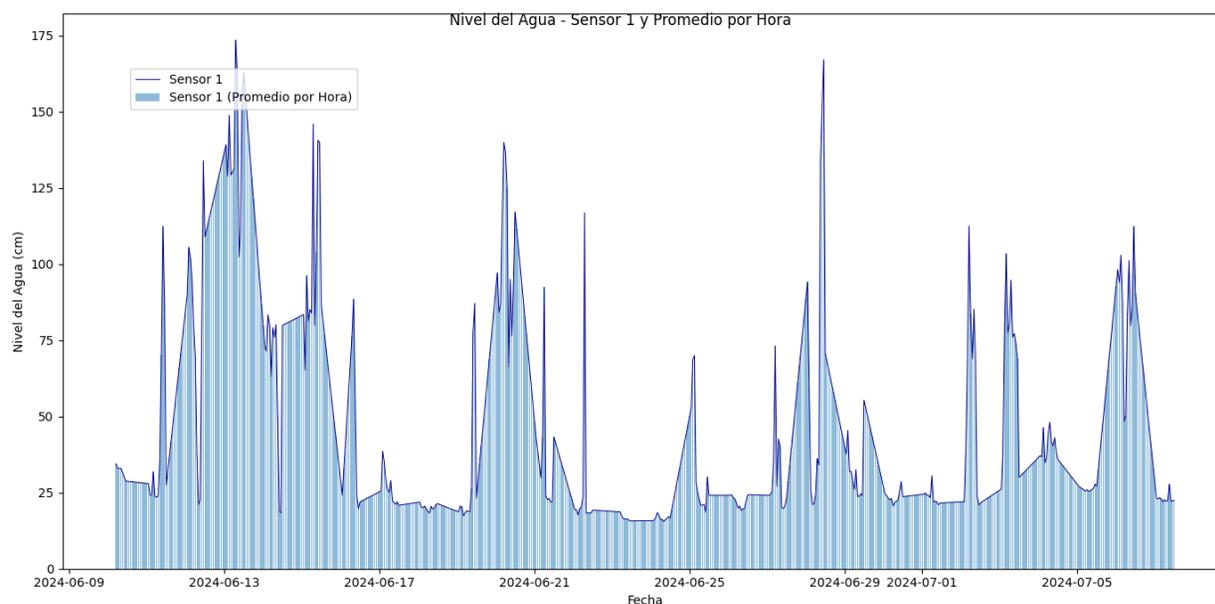
Esta gráfica muestra únicamente la precipitación a lo largo del tiempo. Se observa lo siguiente:

- Picos de Precipitación: Los picos más altos de precipitación ocurrieron alrededor del 17 de junio y nuevamente a finales de junio.
- Tendencia: La precipitación es intermitente, con varios picos menores a lo largo del periodo.

Es importante analizar estos datos para entender cómo las diferentes zonas de precipitación impactan el nivel del río. Identificar los días con lluvias significativas y correlacionarlos con los aumentos en el nivel del agua permitirá establecer patrones y mejorar las predicciones de posibles desbordes.

### Figura 66

*Gráfica de promedio diario de nivel de agua de Junio.*



Para continuar con el análisis de la época crítica, se utilizan los datos recogidos por el sensor 1, el cual está ubicado en un brazo del río a 350 centímetros sobre la superficie del río. Las medidas

muestran en centímetros la distancia desde el lecho del río hasta la superficie del agua. Cabe recalcar que este brazo del río únicamente aparece en épocas críticas.

Esta gráfica se enfoca en el sensor 1, mostrando tanto los niveles de agua horarios como el promedio por hora. Observamos lo siguiente:

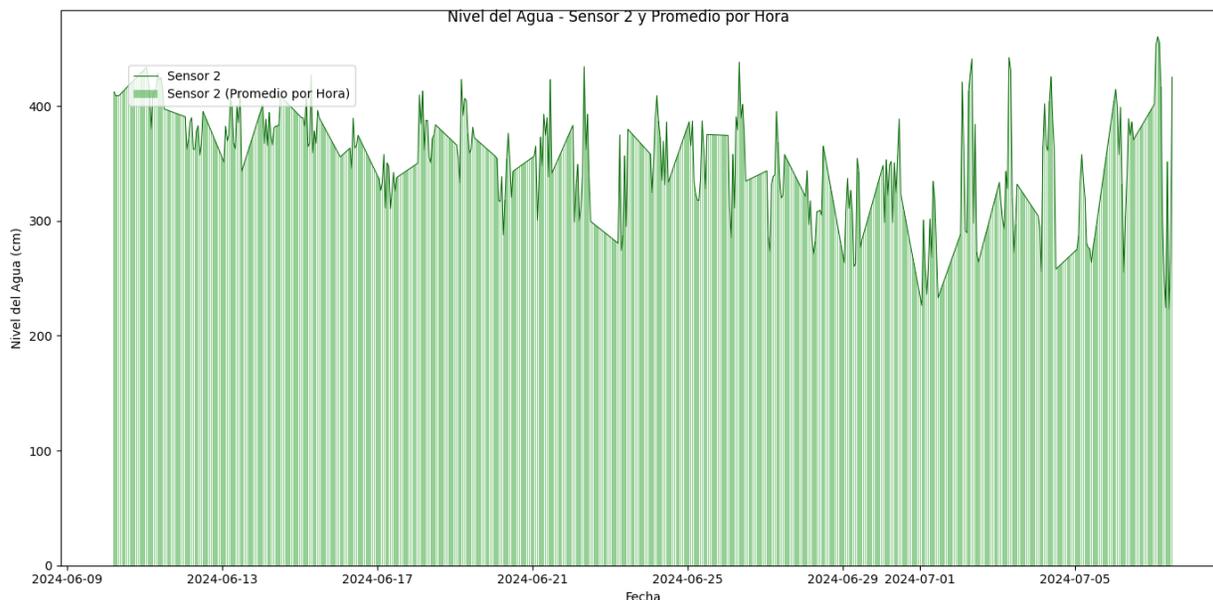
- Picos de Nivel de Agua: Los niveles de agua del sensor 1 tienen picos que coinciden con los picos de precipitación observados en la gráfica de precipitación.
- Consistencia: Los datos muestran una alta variabilidad, pero se observa una tendencia general al alza en los picos después de los eventos de precipitación.

La gráfica muestra crecidas de hasta 175 centímetros, las cuales entran dentro del rango de valores no críticos en la zona. También se observan picos, como el registrado el día 22 de junio, que corresponden a precipitaciones ocurridas en el lugar donde están instalados los sensores.

Este análisis es fundamental para identificar patrones y entender cómo las precipitaciones locales afectan los niveles de agua en diferentes secciones del río. Al correlacionar estos datos con los eventos de precipitación, se pueden ajustar mejor los umbrales de alerta y mejorar la capacidad del sistema para detectar y responder a situaciones críticas. Además, este tipo de información es valiosa para la planificación y gestión de recursos hídricos en la región.

## Figura 67

*Promedio diario del nivel de agua del mes de Junio.*



Para el análisis del sensor dos se toma en cuenta que este se encuentra a seis metros sobre la superficie del cuerpo principal del río blanco, en una curva pronunciada por lo que los datos tienen un rango de variación alto y tienen un valor mayor a los del sensor 1.

El nivel del agua se mide, como en el caso anterior, desde el lecho del río, sin embargo, este dato es estimado por la complejidad del sector.

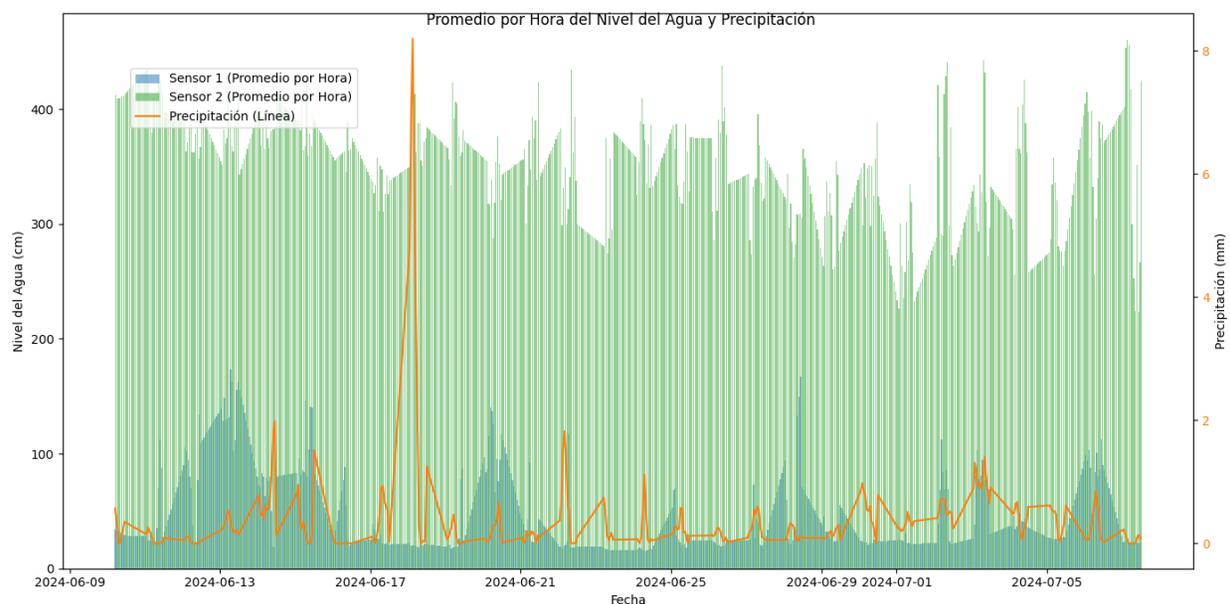
En la gráfica se logra observar cómo los datos, a pesar de ser más variables, mantienen un comportamiento estable debido a que al frente del acantilado existe una playa que estabiliza el nivel del agua en el sector.

Esta gráfica se enfoca en el sensor 2, mostrando tanto los niveles de agua horarios como el promedio por hora. Se observa lo siguiente:

- Picos de Nivel de Agua: Similar al sensor 1, el sensor 2 muestra picos en niveles de agua que coinciden con los eventos de precipitación.

## Figura 68

*Comparación de crecimiento de agua y precipitación.*



Por último, se superponen las gráficas de precipitación y de nivel del agua para identificar coincidencias entre ciertos picos de precipitación y las crecidas del río.

Dado que los datos de precipitación representan un promedio general de las precipitaciones en la provincia, es crucial considerar la ubicación específica de las precipitaciones para que coincidan exactamente con los aumentos en el nivel del agua. Las variaciones locales en las precipitaciones pueden influir significativamente en la correlación observada entre los picos de precipitación y las crecidas del río.

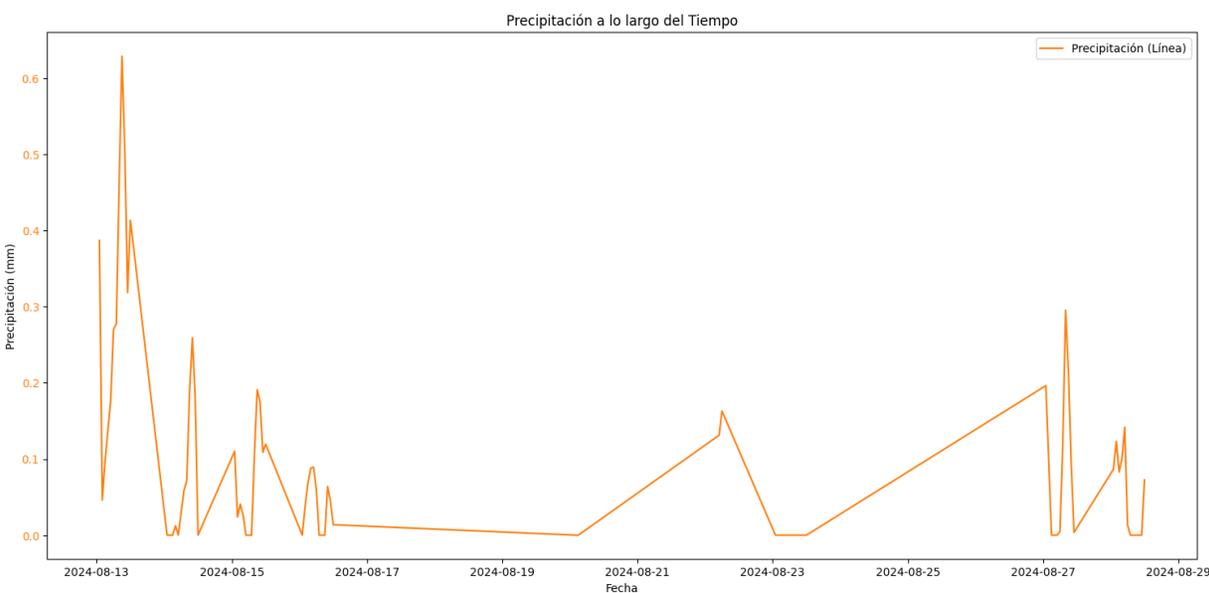
En esta gráfica, se observa los promedios horarios de los niveles de agua de los sensores 1 y 2 (en barras) y la precipitación (en línea) en el mismo periodo. Notamos lo siguiente:

- Sensor 1 y 2: El sensor 2 muestra niveles de agua consistentemente más altos que el sensor 1, lo cual es coherente con la ubicación del sensor 2.
- Precipitación: Hay picos significativos de precipitación que coinciden con aumentos en los niveles de agua de ambos sensores.
- Correlación: Se logran visualizar coincidencias exactas los días 14 , 15, 22 de Junio y 3 de Julio.

#### 4.4. Análisis de datos en temporada de riesgo bajo.

**Figura 69**

*Grafica de precipitaciones de agosto.*



El análisis del sistema en épocas de bajo riesgo se realiza con el objetivo de obtener muestras de control, ya que durante esta temporada hay menos precipitaciones, lo que debería resultar en niveles de agua más estables.

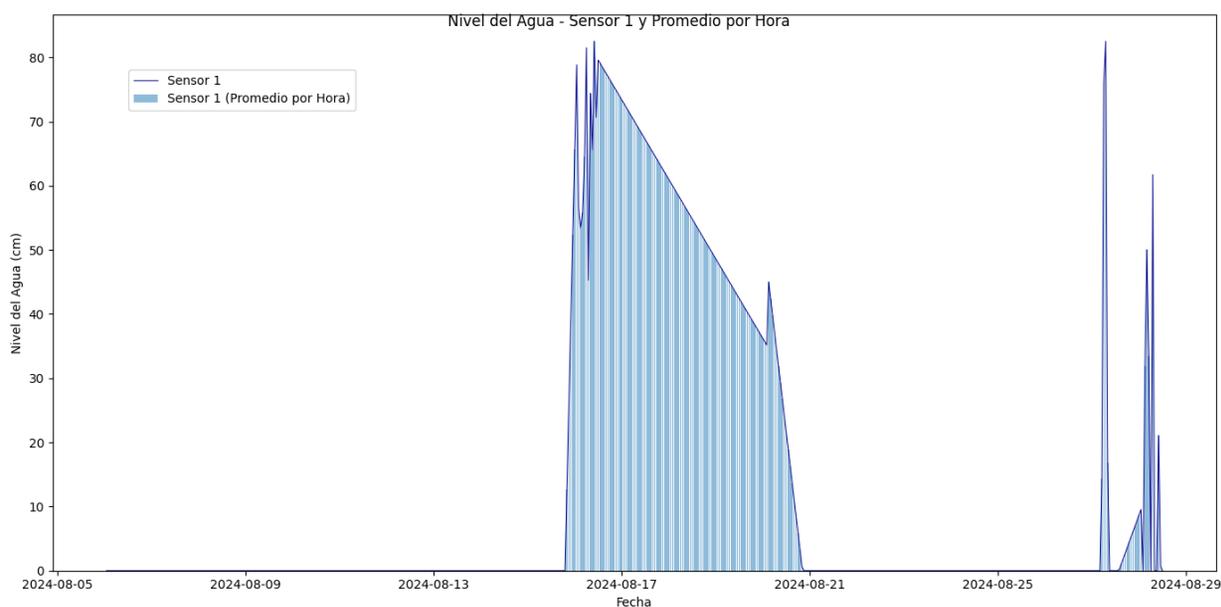
En la figura anterior, se puede observar que las precipitaciones han sido menos frecuentes y de menor intensidad, lo que ha provocado que ciertas zonas del río se sequen. En particular, el brazo

del río donde se instaló el sensor 1 ha quedado completamente seco. Como resultado, el sensor en este periodo solo registra la distancia al lecho del río.

Este análisis en épocas de menor actividad hídrica es crucial para establecer una línea base de comportamiento del río en condiciones normales. Estos datos de control permiten validar el funcionamiento de los sensores y ajustar los umbrales de alerta de manera adecuada. Además, se puede comprobar si los sensores siguen operando correctamente cuando no hay agua, asegurando que estén listos para detectar cambios cuando las condiciones climáticas lo requieran.

### Figura 70

*Gráfica de promedio diario de nivel de agua de agosto.*



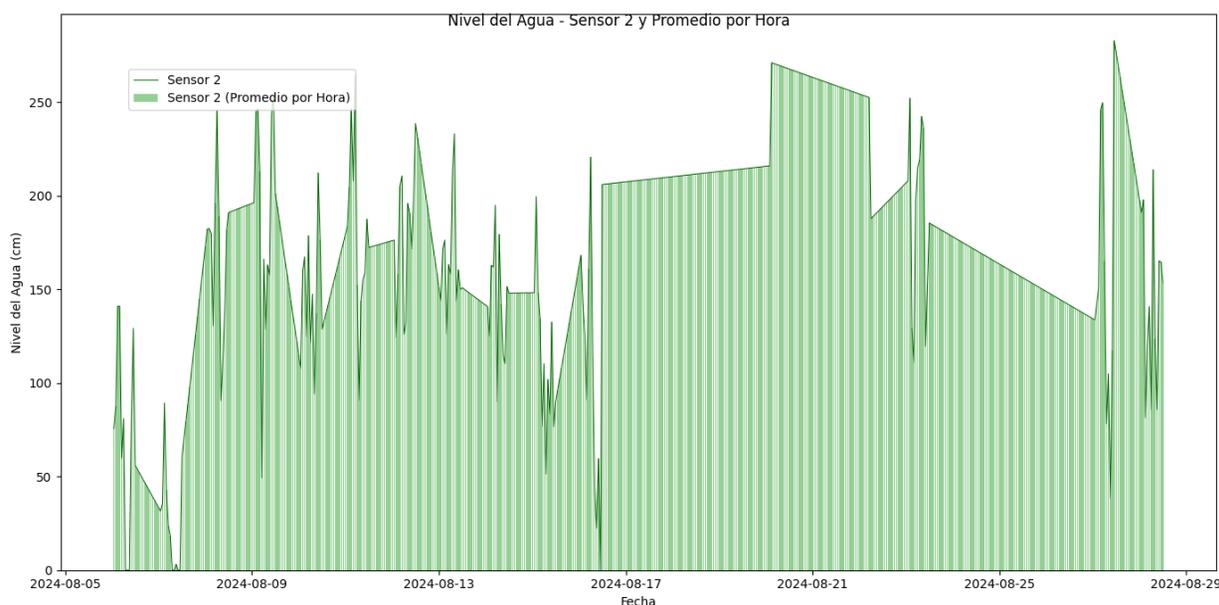
Como se observa en la gráfica, el sensor 1 registra medidas constantes, lo cual se debe a la acumulación de sedimento que quedó cuando el río se secó en esa zona. Este sedimento impide que el sensor detecte variaciones significativas en el nivel del agua, manteniendo las lecturas estables.

Además, se puede visualizar una reducción gradual del nivel del agua a medida que se aproxima la época de bajo riesgo. Este comportamiento es característico de las temporadas secas, donde las precipitaciones disminuyen y, en consecuencia, el flujo del río también se reduce.

En la figura se identifican patrones en los cambios del nivel del agua lo cual garantiza que los sensores continúan funcionando de manera precisa, incluso en condiciones adversas como la presencia de sedimentos. A partir de estos datos, se pueden hacer ajustes en el sistema de monitoreo para mejorar su precisión en futuras épocas críticas y bajas.

### Figura 71

*Promedio diario del nivel de agua del mes de Junio.*



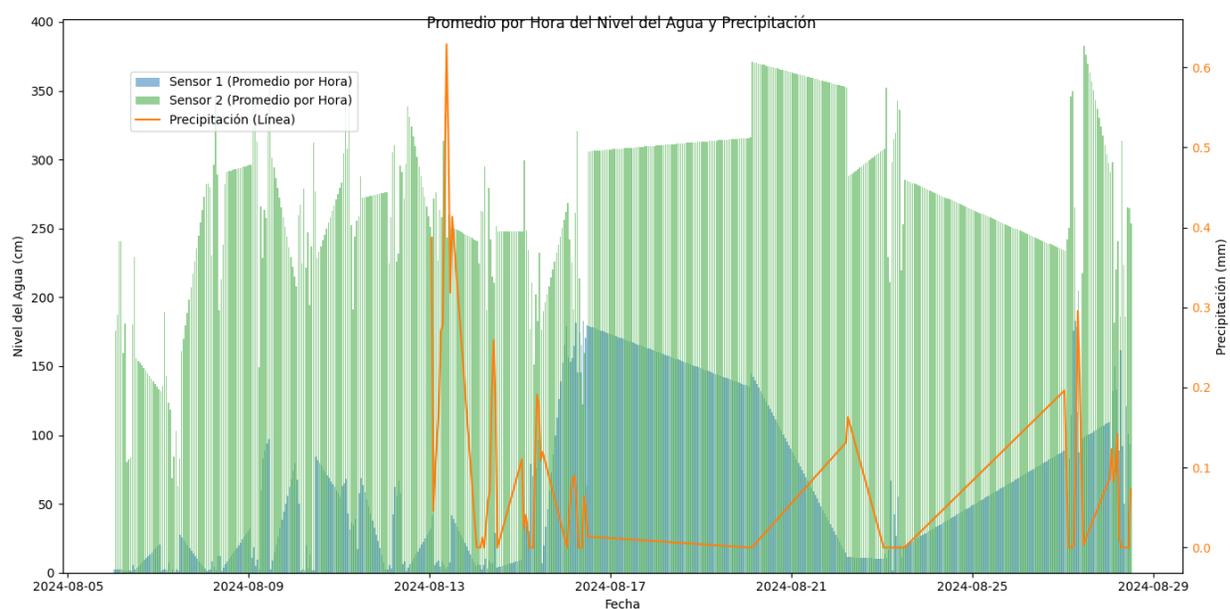
El sensor 2, a pesar de encontrarse en la época seca, continúa registrando altos niveles de agua, ya que está ubicado en el cauce principal del río. Como se mencionó anteriormente, la única manera de que este sensor registre una sequía total sería si el río completo se secara, lo cual es poco probable.

En la gráfica, se observan ciertos picos que corresponden a fallas detectadas en el sensor. Estas irregularidades fueron identificadas y corregidas oportunamente, lo que ha permitido que el sensor proporcione ahora lecturas más precisas y confiables.

Además, se puede visualizar un cambio gradual en el comportamiento del nivel del agua a medida que el río entra en la época de bajo riesgo. Este cambio refleja la disminución del caudal del río debido a la reducción en las precipitaciones, característica de esta temporada. La precisión del sensor 2 es fundamental para garantizar un monitoreo continuo y preciso durante todo el año, especialmente en zonas críticas como el cauce principal del río.

### Figura 72

*Comparación de crecimiento de agua y precipitación.*



Al comparar las gráficas, se puede observar una clara correlación entre las precipitaciones y los niveles de agua medidos por los sensores. Esta correlación confirma que los sensores están cumpliendo su función de manera adecuada y que sus lecturas son confiables para el sistema de alerta temprana.

La consistencia entre los picos de precipitación y los aumentos en los niveles del agua refuerza la precisión de los sensores en el monitoreo del río. Esto garantiza que el sistema de alerta temprana pueda detectar con exactitud situaciones de riesgo, activando las alarmas en el momento oportuno y proporcionando información crítica a los usuarios para tomar decisiones rápidas y efectivas frente a posibles inundaciones.

#### **4.5. Medición de tiempo de respuesta.**

El tiempo de respuesta ha sido evaluado mediante cálculos basados en la velocidad del río, proporcionada por una empresa de turismo especializada en deportes acuáticos. A 13 kilómetros río arriba, donde están instalados los sensores, la velocidad del río fue medida en 18 km/h durante la época de alto riesgo y en 10 km/h durante la época de bajo riesgo.

Tomando en cuenta la velocidad del río en la época de alto riesgo, se estima que una creciente tardaría 43.2 minutos en llegar a la parroquia de Valle Hermoso. Para optimizar el sistema de alerta, se ha establecido un margen de 10 minutos para la detección, envío y recepción de la alerta a la población, lo que deja un tiempo efectivo de 33.2 minutos para que los ciudadanos se pongan a salvo.

**Figura 73**

*Instrucciones dadas por el capitán del cuerpo de bomberos de Santo Domingo*



Para verificar estos cálculos, se realizó un simulacro junto con el cuerpo de bomberos, cuyo objetivo fue medir los tiempos de reacción desde diferentes puntos de la parroquia, como se detalla en el anexo 6. Los resultados mostraron que, al tratarse de una parroquia pequeña, los ciudadanos más alejados del punto de reunión tardaron un máximo de 15 minutos en llegar al punto seguro. Esto proporciona un margen de mejora de 18.2 minutos respecto al tiempo de reacción estimado antes de la instalación de los sensores.

Este ejercicio no solo valida la efectividad del sistema de alerta temprana, sino que también indica áreas donde se puede seguir optimizando el tiempo de respuesta, garantizando así una mayor seguridad para los habitantes de Valle Hermoso frente a posibles inundaciones.

## Conclusiones y recomendaciones

### Conclusiones

El proyecto logró implementar con éxito un sistema de alerta temprana basado en la arquitectura IoT. Este sistema permite monitorear en tiempo real los niveles de agua del río, transmitir los datos mediante módulos LORA, y procesar la información en una Raspberry Pi que se conecta a Thinger.io para la gestión y almacenamiento en la nube. La implementación de este sistema ha demostrado mejorar significativamente el tiempo de respuesta ante posibles emergencias de inundación, proporcionando alertas tempranas a través de Telegram y almacenando los datos de manera segura en la nube para análisis posteriores.

Durante el desarrollo del proyecto, se realizó una extensa revisión de literatura y consulta de bases de conocimiento para determinar los parámetros físico-hidrológicos críticos a medir, como el nivel del agua y la velocidad de flujo. Esto permitió diseñar un sistema robusto y adaptado a las condiciones específicas del río Blanco. La selección de sensores de ultrasonido y la configuración de los módulos LORA se basaron en estos conocimientos, asegurando mediciones precisas y confiables.

Se llevó a cabo un análisis detallado de la situación actual del sector, identificando puntos estratégicos para la instalación de los sensores y módulos de comunicación. La elección de estos puntos se basó en la topografía del área, el comportamiento del río en diferentes condiciones climáticas y la accesibilidad para el mantenimiento. Se instaló el segundo sensor en la propiedad del Sr. Marco Carrillo, aprovechando la ubicación elevada y la proximidad al acantilado, lo que permite una detección temprana de niveles críticos de agua.

El despliegue del sistema de alerta temprana se realizó de manera exitosa, validando su funcionamiento a través de pruebas exhaustivas. Los resultados mostraron que el sistema es capaz de detectar aumentos en los niveles de agua en tiempo real y generar alertas oportunas. Los tiempos de respuesta fueron adecuados para permitir la evacuación segura del sector en caso de emergencia. Las alertas enviadas a través de Telegram y la actualización constante de los datos en Thinger.io aseguraron una comunicación eficaz y una toma de decisiones informada para la gestión de emergencias.

### **Recomendaciones**

Se recomienda extender el sistema de alerta temprana a otras áreas vulnerables a inundaciones dentro de la región. Esto incluiría la instalación de sensores adicionales en puntos críticos identificados durante el análisis de riesgos. La ampliación de la cobertura permitirá un monitoreo más completo y una respuesta más efectiva ante posibles emergencias.

Es crucial implementar un programa de mantenimiento preventivo y regular para los sensores, módulos LORA, y la infraestructura de comunicación. Esto asegurará que todos los componentes del sistema funcionen correctamente y que las lecturas de los sensores sean precisas. Se recomienda realizar inspecciones trimestrales y calibraciones de los sensores según las especificaciones del fabricante.

Realizar evaluaciones periódicas del sistema para identificar áreas de mejora y actualizar los componentes tecnológicos según sea necesario. Esto incluye la evaluación de la eficacia de las alertas, la precisión de los datos, y la velocidad de respuesta del sistema. Las actualizaciones deben basarse en los resultados de estas evaluaciones y en las últimas innovaciones tecnológicas disponibles.

Involucrar a la comunidad en el proceso de monitoreo y respuesta a emergencias mediante programas de educación y concientización. Informar a los residentes sobre cómo funciona el sistema de alerta temprana, qué hacer en caso de una alerta, y cómo pueden colaborar con las autoridades. La participación activa de la comunidad fortalecerá la resiliencia y la capacidad de respuesta ante inundaciones.

Buscar oportunidades de financiación y establecer colaboraciones con instituciones académicas, organizaciones no gubernamentales, y entidades gubernamentales para apoyar el mantenimiento y expansión del sistema. La colaboración con expertos y la obtención de recursos adicionales facilitarán la mejora continua del sistema y su sostenibilidad a largo plazo.

## Bibliografía

- Abad, V. (23 de 01 de 2013). *Instituto geofísico militar del Ecuador*. Obtenido de geoportaligm:  
<https://www.geoportaligm.gob.ec/>
- Alliance, L. (2020). *NOTICE OF USE AND DISCLOSURE*. Fremont: LoRa Alliance, Inc. .
- Amazon. (12 de Febrero de 2023). *Amazon Web Services*. Obtenido de Amazon:  
<https://aws.amazon.com/es/what-is/iot/#:~:text=El%20t%C3%A9rmino%20IoT%2C%20o%20Internet,como%20entre%20los%20propios%20dispositivos.>
- Americanos, O. d. (2010). *Manual para el diseño, instalación, operación y mantenimiento de* .  
 Washington, D.C. : Departamento de Desarrollo .
- Americanos, S. G. (2001). *MANUAL PARA EL DISEÑO E IMPLEMENTACIÓN DE UN*.  
 Washington, D.C.: OEA.
- Argote, A. H. (23 de Diciembre de 2023). *PROGRAMA DE ALTA ESPECIALIZACIÓN EN MONITOREO DE RECURSOS NATURALES*. Obtenido de Escuela global:  
<https://www.especializacionesglobal.net/dashboard/enrolled-courses/>
- Bustamante, P. A. (15 de Marzo de 2024). *thinger.io*. Obtenido de docs.thinger.io:  
<https://docs.thinger.io/about>
- California, U. o. (28 de Marzo de 2023). *ucsb*. Obtenido de chc.ucsb.edu:  
<https://chc.ucsb.edu/data/chirps>
- Chávez, W. M. (Plan de Desarrollo y Ordenamiento Territorial Parroquia Valle Hermoso 2020-2023 ). *Valle Hermosos*. Santo Domingo : GAD Municipal. Obtenido de [www.gadvallehermoso.gob.ec](http://www.gadvallehermoso.gob.ec).

- Europea, U. (15 de 02 de 2024). *copernicus*. Obtenido de [sentiwiki.copernicus.eu](https://sentiwiki.copernicus.eu):  
<https://sentiwiki.copernicus.eu/web/s1-applications>
- Mies Sutorius, S. R. (14 de Diciembre de 2015). La fundamentalidad del derecho al agua en Colombia. *Derecho del Estado*, pág. 2.
- Niño, C. I. (17 de Febrero de 2017). Implementación de Sistemas de Alerta Temprana. *Programa Regional Andino para el Fortalecimiento de los Servicios Meteorológicos, Hidrológicos, Climáticos y el Desarrollo (PRASDES)*, pág. 6.
- Openwather. (05 de Marzo de 2024). *openweathermap*. Obtenido de [openweathermap.org](https://openweathermap.org):  
<https://openweathermap.org/guide>
- Ramirez, P. (2022). *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*. Ibarra: Universidad Técnica del Norte.
- Rdagger. (16 de Julio de 2017). *rototron*. Obtenido de [www.rototron.info](http://www.rototron.info):  
<https://www.rototron.info/raspberry-pi-esp32-micropython-tutorial/>
- REYAX. (2021). *LoRa AT COMAND GUIDE*. Taipéi: Reyax technology corporation.
- Rodriguez, C. (15 de Diciembre de 2008). Guía práctica de gestión de riesgos. *Inteco*, págs. 12-17.
- Salud, O. P. (13 de Octubre de 2000). Publicación Científica No. 575. *Organización Panamericana de la Salud*, págs. 12-17.
- Sethi, P. (2017). Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, 1-2.

Workgroup, T. M. (14 de Febrero de 2015). A technical overview of LoRa® and LoRaWAN™.

*LoRa® Aliance*, págs. 7-10.

## ANEXOS

### Anexo 1: Código del módulo Emisor.

```
// Se definen las direcciones del módulo LoRa

String lora_address = "1"; // Dirección del módulo LoRa

String lora_RX_address = "2"; // Dirección del módulo receptor LoRa

// Se definen los pines para el sensor ultrasónico

#define trigPin 22 // Pin de salida del sensor ultrasónico

#define echoPin 23 // Pin de entrada del sensor ultrasónico

long duration; // Variable para almacenar la duración del pulso

int distance; // Variable para almacenar la distancia medida

String textoEntrada; // Variable para almacenar datos recibidos

void setup() {

  // Se inicia la comunicación serial para el módulo LoRa

  Serial2.begin(115200, SERIAL_8N1, 16, 17);

  // Se configuran los pines del sensor ultrasónico

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  // Se inicia la comunicación serial para depuración
```

```
Serial.begin(115200);  
  
delay(1500);  
  
// Se configura la dirección del módulo LoRa  
Serial2.println("AT+ADDRESS=" + lora_address);  
delay(1500);  
}  
  
void loop() {  
  // Se envía un pulso al sensor ultrasónico  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(5);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  // Se mide la duración del pulso reflejado por el objeto  
  duration = pulseIn(echoPin, HIGH);  
  
  // Se calcula la distancia en centímetros  
  distance = (duration / 2) / 29.1;  
  
  // Se convierte la distancia a cadena de caracteres
```

```
String distanceStr = String(distance);

// Se obtiene la cantidad de caracteres en la distancia medida
int num_caracteres = distanceStr.length();

// Se envía la distancia medida al módulo receptor LoRa
Serial2.println("AT+SEND=" + lora_RX_address + "," + String(num_caracteres) + "," +
String(distance));

// Se muestra la distancia medida por la comunicación serial
Serial.println("AT+SEND=" + lora_RX_address + "," + String(num_caracteres) + "," +
String(distance));

// Se espera a recibir datos del emisor LoRa
textoEntrada = Serial2.readString();

// Se muestra por consola los datos recibidos del emisor
Serial.println(textoEntrada);

// Se espera 1 segundo antes de la próxima iteración
delay(1000);
}
```

**Anexo 2: Código del módulo Receptor.**

```
String lora_address = "2"; // Dirección del módulo LoRa

void setup() {
  Serial.begin(115200); // Inicia la comunicación serial a 115200 baudios
  Serial2.begin(115200, SERIAL_8N1, 16, 17); // Inicia la comunicación serial 2 en el ESP32
  (pin 16 RXD, 17 TXD UART)
  delay(1500);
  Serial2.println("AT+ADDRESS=" + lora_address); // Envía el comando AT para configurar la
  dirección LoRa
  delay(1500); // Espera 1.5 segundos
}

void loop() {
  String data; // Variable para almacenar los datos recibidos
  char c; // Variable para almacenar cada carácter recibido

  while (Serial2.available()) { // Mientras haya datos disponibles en Serial2
    c = Serial2.read(); // Lee el siguiente carácter disponible
    if (c == '+') { // Verifica si es el inicio de un nuevo conjunto de datos
      data = c; // Inicializa el buffer con el primer carácter
    } else if (!data.isEmpty()) { // Si estamos almacenando datos
      data += c; // Agrega el carácter al buffer
    }
  }
}
```

```
if (c == '\n') { // Si es el final del conjunto de datos

    if (data.startsWith("+RCV=1")) { // Si los datos son de la fuente 1

        // Procesa los datos de la fuente 1

        Serial.println(data); // Imprime los datos recibidos por Serial

    } else if (data.startsWith("+RCV=3")) { // Si los datos son de la fuente 3

        // Procesa los datos de la fuente 3

        Serial.println(data); // Imprime los datos recibidos por Serial

    }

    data = ""; // Reinicia el buffer para el próximo conjunto de datos

}

}

}

}
```

**Anexo 3: Código del sistema de procesamiento de datos.**

```
import serial

import json

import time

from datetime import datetime

import serial.tools.list_ports

from statistics import median

# Configuración de la velocidad de baudios

BAUD_RATE = 115200

# Archivos para almacenar los datos

DATA_FILE_RCV1 = '/home/satvallehermoso/sensor_data_rcv1.json'

DATA_FILE_RCV3 = '/home/satvallehermoso/sensor_data_rcv3.json'

# Parámetros específicos para rcv1

ALTURA_MAXIMA_RCV1 = 300 # Máxima altura que puede crecer el río en cm

CAMBIO_MAXIMO_RCV1 = 15 # Máximo cambio permitido entre mediciones consecutivas

en cm

ALTURA_REF = 350 # Altura de referencia para calcular el crecimiento del río en cm

# Parámetros específicos para rcv3

ALTURA_MAXIMA_RCV3 = 500 # Máxima altura que puede crecer el río en cm
```

```
CAMBIO_MAXIMO_RCV3 = 100 # Máximo cambio permitido entre mediciones consecutivas
en cm

# Última distancia registrada para rcv1
datos_rcv1_iniciales = []
referencia_rcv1 = None
ultimo_dato_valido_rcv1 = None

# Último crecimiento registrado para rcv3
ultimo_crecimiento_rcv3 = None
ultimo_dato_valido_rcv3 = None

def encontrar_puerto_serial():
    puertos = serial.tools.list_ports.comports()

    for puerto in puertos:
        try:
            ser = serial.Serial(puerto.device, BAUD_RATE, timeout=1)

            print(f"Probando puerto: {puerto.device}")

            time.sleep(2)

            if ser.in_waiting > 0:
                linea = ser.readline().decode('utf-8').strip()

                if linea.startswith('+RCV='):
                    print(f"Puerto encontrado: {puerto.device}")

                    ser.close()
```

```
        return puerto.device

    ser.close()

except (serial.SerialException, UnicodeDecodeError):

    continue

return None

def leer_datos_serial(puerto_serial):

    try:

        ser = serial.Serial(puerto_serial, BAUD_RATE, timeout=1)

        while True:

            try:

                if ser.in_waiting > 0:

                    linea = ser.readline().decode('utf-8').strip()

                    if linea.startswith('+RCV='):

                        procesar_datos(linea)

            except serial.SerialException as e:

                print(f"Error al leer datos del serial: {e}")

            except UnicodeDecodeError as e:

                print(f"Error al decodificar datos: {e}")

            except Exception as e:

                print(f"Error inesperado: {e}")

    except serial.SerialException as e:

        print(f"No se pudo abrir el puerto serial: {e}")
```

```
except Exception as e:
```

```
    print(f"Error inesperado al iniciar la lectura del puerto serial: {e}")
```

```
finally:
```

```
    if ser.is_open:
```

```
        ser.close()
```

```
        print("Puerto serial cerrado.")
```

```
def procesar_datos(linea):
```

```
    global datos_rcv1_iniciales, referencia_rcv1, ultimo_dato_valido_rcv1,
```

```
ultimo_crecimiento_rcv3, ultimo_dato_valido_rcv3
```

```
    partes = linea.split(',')
```

```
    if len(partes) >= 3:
```

```
        try:
```

```
            tipo_rcv = partes[0].strip()
```

```
            if tipo_rcv.startswith('+RCV='):
```

```
                tipo_rcv = tipo_rcv.split('=')[1]
```

```
                distancia = int(partes[2])
```

```
            if tipo_rcv == '1':
```

```
                # Calcular crecimiento del río para rcv1
```

```
                crecimiento = ALTURA_REF - distancia
```

```
                if crecimiento < 0 or crecimiento > ALTURA_MAXIMA_RCV1:
```

```
                    print(f"Crecimiento fuera de rango para rcv1:
```

```
{crecimiento} cm")  
  
if ultimo_dato_valido_rcv1:  
    guardar_datos(ultimo_dato_valido_rcv1, tipo_rcv)  
  
return
```

```
# Almacenar los primeros 10 datos
```

```
if len(datos_rcv1_iniciales) < 10:  
    datos_rcv1_iniciales.append(crecimiento)  
  
if len(datos_rcv1_iniciales) == 10:  
    referencia_rcv1 = median(datos_rcv1_iniciales)  
    print(f"Referencia inicial calculada para rcv1:  
    {referencia_rcv1} cm")  
  
return
```

```
# Validar cambio brusco después de los primeros 10 datos
```

```
if referencia_rcv1 is not None:  
    cambio = abs(crecimiento - referencia_rcv1)  
  
if cambio > CAMBIO_MAXIMO_RCV1:  
    print(f"Cambio brusco detectado para rcv1:  
    {cambio} cm")  
  
if ultimo_dato_valido_rcv1:  
    guardar_datos(ultimo_dato_valido_rcv1,  
    tipo_rcv)
```

```
        return

referencia_rcv1 = crecimiento

data = {

    'tipo_rcv': tipo_rcv,

    'crecimiento': crecimiento,

    'timestamp': datetime.now().isoformat()

}

ultimo_dato_valido_rcv1 = data

guardar_datos(data, tipo_rcv)

elif tipo_rcv == '3':

    # Validar crecimiento del río para rcv3

    crecimiento = distancia # El valor de distancia ya

    representa el crecimiento

    if crecimiento < 0 or crecimiento > ALTURA_MAXIMA_RCV3:

        print(f"Crecimiento fuera de rango para rcv3:

        {crecimiento} cm")

        if ultimo_dato_valido_rcv3:

            guardar_datos(ultimo_dato_valido_rcv3, tipo_rcv)

        return

if ultimo_crecimiento_rcv3 is not None:

    cambio = abs(crecimiento - ultimo_crecimiento_rcv3)

    if cambio > CAMBIO_MAXIMO_RCV3:

        print(f"Cambio brusco detectado para rcv3:
```

```
        {cambio} cm")

    if ultimo_dato_valido_rcv3:

        guardar_datos(ultimo_dato_valido_rcv3,

            tipo_rcv)

        return

ultimo_crecimiento_rcv3 = crecimiento

data = {

    'tipo_rcv': tipo_rcv,

    'crecimiento': crecimiento,

    'timestamp': datetime.now().isoformat()

}

ultimo_dato_valido_rcv3 = data

guardar_datos(data, tipo_rcv)

else:

    print(f"Tipo de datos desconocido: {tipo_rcv}")

    return

except ValueError as e:

    print(f"Error al convertir los datos: {e}")

except Exception as e:

    print(f"Error inesperado al procesar los datos: {e}")

def guardar_datos(data, tipo_rcv):

    try:
```

```
if tipo_rcv == '1':  
    file_path = DATA_FILE_RCV1  
  
elif tipo_rcv == '3':  
    file_path = DATA_FILE_RCV3  
  
else:  
    print(f"Tipo de datos desconocido: {tipo_rcv}")  
  
    return  
  
try:  
    with open(file_path, 'r') as file:  
        existing_data = json.load(file)  
  
    except (IOError, json.JSONDecodeError):  
        existing_data = []  
  
existing_data.append(data)  
  
    with open(file_path, 'w') as file:  
        json.dump(existing_data, file, indent=4)  
        print(f"Datos guardados en {file_path}: {data}")  
  
except IOError as e:  
    print(f"Error al guardar datos: {e}")  
  
except Exception as e:  
    print(f"Error inesperado al guardar los datos: {e}")
```

```
if __name__ == '__main__':  
    puerto_serial = encontrar_puerto_serial()  
  
    if puerto_serial:  
        leer_datos_serial(puerto_serial)  
  
    else:  
        print("No se encontró un puerto serial válido.")
```

**Anexo 4: Código del servicio de subida de datos.**

```
#include <iostream>

#include <string>

#include <chrono>

#include <thread>

#include <mutex>

#include <fstream>

#include <curl/curl.h>

#include <nlohmann/json.hpp>

#include "thinger/thinger.h"

using json = nlohmann::json;

#define USER_ID      "jesus2512"

#define DEVICE_ID    "raspberrry"

#define DEVICE_CREDENTIAL "raspberrry"

// Variables globales para distancias y precipitación

std::mutex mtx;

int distancia1 = -1;

int distancia2 = -1;

float precipitacion = 0.0;
```

```
bool alerta_enviada = false; // Variable para evitar enviar múltiples alertas para la misma
condición

// Función de callback para manejar los datos recibidos de la API
size_t WriteCallback(void* contents, size_t size, size_t nmemb, void* userp) {
    ((std::string*)userp)->append((char*)contents, size * nmemb);
    return size * nmemb;
}

// Función para obtener datos de precipitación desde OpenWeatherMap
float obtener_precipitacion(const std::string& ciudad, const std::string& apiKey) {
    CURL* curl;
    CURLcode res;
    std::string readBuffer;

    curl_global_init(CURL_GLOBAL_DEFAULT);
    curl = curl_easy_init();
    if (curl) {
        std::string url = "http://api.openweathermap.org/data/2.5/weather?q=" +
std::string(curl_easy_escape(curl, ciudad.c_str(), ciudad.length())) + "&appid=" + apiKey;
        curl_easy_setopt(curl, CURLOPT_URL, url.c_str());
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteCallback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &readBuffer);
    }
}
```

```
res = curl_easy_perform(curl);

if (res != CURLE_OK) {
    std::cerr << "curl_easy_perform() failed: " << curl_easy_strerror(res) << std::endl;
    curl_easy_cleanup(curl);
    curl_global_cleanup();
    return -1;
}

curl_easy_cleanup(curl);
curl_global_cleanup();

// Parsear los datos JSON recibidos
auto jsonData = json::parse(readBuffer);

// Extraer los datos de precipitación
if (jsonData.contains("rain") && jsonData["rain"].contains("1h")) {
    precipitacion = jsonData["rain"]["1h"];
} else {
    std::cout << "No hay datos de precipitación disponibles." << std::endl;
    precipitacion = 0.0; // Asignar 0 si no hay datos disponibles
}

return precipitacion;
```

```
}

curl_global_cleanup();

return -1;
}

void enviar_precipitacion_periodicamente(thinger_device& thing, const std::string& ciudad,
const std::string& apiKey, std::chrono::seconds intervalo) {
    while (true) {
        try {
            obtener_precipitacion(ciudad, apiKey);

            if (precipitacion >= 0) {
                std::cout << "Precipitación en la última hora: " <<
                precipitacion << " mm" << std::endl;

                // Enviar precipitación a través de Thingier.io
                thing["Precipitación"] >> [](pson& out){
                    out = precipitacion;
                };
            }

            std::this_thread::sleep_for(intervalo); // Esperar el intervalo
            especificado antes de la próxima lectura
        }
    }
}
```

```
    } catch (const std::exception& e) {  
        std::cerr << "Error al obtener o enviar datos de precipitación: "  
        << e.what() << std::endl;  
    }  
}  
}  
}  
  
void leer_json(const std::string& file_path, int& distancia) {  
    std::ifstream file(file_path);  
    if (!file.is_open()) {  
        std::cerr << "Error al abrir el archivo JSON: " << file_path <<  
        std::endl;  
        return;  
    }  
  
    // Verificar si el archivo está vacío  
    if (file.peek() == std::ifstream::traits_type::eof()) {  
        std::cerr << "El archivo JSON está vacío: " << file_path << std::endl;  
        file.close();  
        return;  
    }  
  
    json j;
```

```
try {
    file >> j;
} catch (const json::parse_error& e) {
    std::cerr << "Error al parsear el archivo JSON: " << e.what() <<
    std::endl;
    file.close();
    return;
}

if (!j.empty() && j.is_array()) {
    auto last_entry = j.back();
    if (last_entry.contains("crecimiento")) {
        distancia = last_entry["crecimiento"];
        std::cout << "Leído " << file_path << ": " << distancia << " cm"
        << std::endl; // Mensaje de registro
    } else {
        std::cerr << "El archivo JSON no contiene la clave 'crecimiento': " << file_path <<
std::endl;
    }
} else {
    std::cerr << "El archivo JSON no es una lista o está vacío: " <<
    file_path << std::endl;
}
}
```

```
file.close();
}

void leer_datos_json_continuamente(thinger_device& thing) {
    int ultima_distancia1 = -1;
    int ultima_distancia2 = -1;
    while (true) {
        {
            std::lock_guard<std::mutex> lock(mtx);
            leer_json("/home/satvallehermoso/sensor_data_rcv1.json",
                    distancia1);
            leer_json("/home/satvallehermoso/sensor_data_rcv3.json",
                    distancia2);

            if (distancia1 != ultima_distancia1) {
                ultima_distancia1 = distancia1;
                std::cout << "Enviando Distancia1 a Thingier.io: " <<
                    distancia1 << " cm" << std::endl;
                thing["Distancia1"] >> [](pson& out){
                    out = distancia1;
                };
            }
        }
    }
}
```

```
if (distancia2 != ultima_distancia2) {  
    ultima_distancia2 = distancia2;  
  
    std::cout << "Enviando Distancia2 a Thinger.io: " <<  
    distancia2 << " cm" << std::endl;  
  
    thing["Distancia2"] >> [](pson& out){  
        out = distancia2;  
  
    };  
}  
  
// Verificar si `rcv1` sube a 300 cm y `rcv3` sube a 500 cm  
  
if (distancia1 >= 300 && distancia2 >= 500 && !alerta_enviada) {  
    std::cout << "Alert: rcv1 >= 300 cm y rcv3 >= 500 cm. Llamando  
    al endpoint de Telegram..." << std::endl;  
  
    thing.call_endpoint("Telegram");  
  
    alerta_enviada = true;  
  
} else if (distancia1 < 300 || distancia2 < 500) {  
    alerta_enviada = false; // Resetear la alerta si las  
    condiciones ya no se cumplen  
  
}  
  
}  
  
std::this_thread::sleep_for(std::chrono::seconds(1)); // Esperar un
```

```
        segundo antes de la siguiente iteración
    }
}

int main() {
    thinger_device thing(USER_ID, DEVICE_ID, DEVICE_CREDENTIAL);

    // Crear un hilo separado para obtener y enviar datos de precipitación
    periódicamente

    std::string ciudad = "Santo Domingo de los Colorados"; // Ciudad deseada

    std::string apiKey = "b9177434e82c112bb7ae1cc0070689ba"; // Tu clave API

    std::chrono::seconds intervalo(10); // Intervalo de 10 segundos

    std::cout << "Iniciando hilos..." << std::endl;

    std::thread hilo_precipitacion(enviar_precipitacion_periodicamente,
    std::ref(thing), ciudad, apiKey, intervalo);

    std::thread hilo_lectura_json(leer_datos_json_continuamente,
    std::ref(thing));

    try {
        while (true) {
            thing.handle(); // Llamar a thing.handle() para mantener la
            conexión con Thinger.io

            std::cout << "Manejando conexión Thinger.io" << std::endl;
        }
    }
}
```

```
        std::this_thread::sleep_for(std::chrono::seconds(1)); // Esperar
        un segundo antes de la siguiente iteración
    }
} catch (const std::exception& e) {
    std::cerr << "Error: " << e.what() << std::endl;
}

// Unirse a los hilos antes de finalizar
hilo_precipitacion.join();
hilo_lectura_json.join();

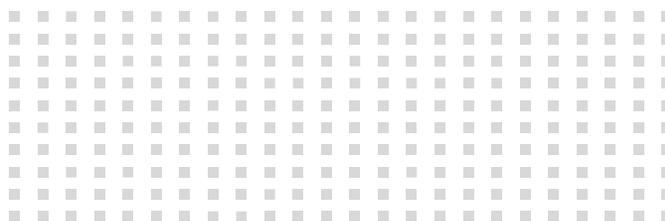
return 0;
}
```

Anexo 5: Manual para el ingreso al sistema de alertas.

# MANUAL SISTEMA DE ALERTA TEMPRANA DE VALLE HERMOSO



Guía para el ingreso y uso  
del sistema de alertas.



---

Jesús Navarrete

[janavarreteg@utn.edu.ec](mailto:janavarreteg@utn.edu.ec)

22 de Junio 2024

Versión 1.0

# MANUAL SISTEMA DE ALERTA TEMPRANA DE VALLE HERMOSO



Guía para el ingreso y uso  
del sistema de alertas.

Jesús Navarrete  
[janavarreteg@utn.edu.ec](mailto:janavarreteg@utn.edu.ec)  
22 de Junio 2024  
Versión 1.0

## Contenido

1. <a href="#">Introducción</a> .....	145
---------------------------------------	-----

2.	<a href="#">Generalidades</a> .....	146
3.	<a href="#">Resumen del sistema</a> .....	148
4.	<a href="#">¿Como acceder a la función de alerta?</a> .....	148
	<a href="#">Instalar la Aplicación de Telegram:</a> .....	148
	<a href="#">Crear una Cuenta de Telegram:</a> .....	149
	<a href="#">Acceder al Canal de Alertas:</a> .....	160
	<a href="#">Unirse al Canal:</a> .....	160
	<a href="#">Revisar Mensajes del Canal:</a> .....	160

### ***Introducción***

Dada la creciente amenaza de futuras inundaciones, es esencial proporcionar a la población información que le permita reaccionar a tiempo y buscar refugio antes de que se desencadene la tragedia. El dispositivo de alerta temprana propuesto será una herramienta vital en la gestión de riesgos, proporcionando a los residentes de Valle Hermoso la capacidad de recibir notificaciones y alertas tempranas sobre la subida del nivel del río y otros indicadores clave de peligro. Esta información permitirá a la población tomar las medidas preventivas adecuadas, como la evacuación temprana y la salvaguarda de sus

pertenencias, con el objetivo de salvaguardar su bienestar físico y minimizar los daños materiales.

Además de su función de alerta, se espera que este dispositivo se convierta también en una fuente fiable de información para la comunidad en general. Se proporcionarán datos actualizados sobre las condiciones hidrometeorológicas, lo que permitirá a los residentes mantenerse informados y tomar decisiones fundamentadas sobre sus actividades cotidianas, especialmente durante los picos de lluvias intensas.

La implementación de un dispositivo de alerta temprana en el sector de Valle Hermoso es una medida urgente y necesaria para hacer frente a la amenaza recurrente de inundaciones, al proporcionar información oportuna y precisa, se espera que esta iniciativa reduzca los riesgos asociados a los desastres naturales y proteja la vida y el bienestar de los habitantes de la parroquia. Además, contribuirá a la concientización de la comunidad sobre la importancia de estar preparados ante eventos extremos y fomentará una cultura de prevención en la zona afectada.

### *Generalidades*

El sistema se basa en el manual para el diseño e implementación de un sistema de alerta temprana de inundaciones en cuencas menores, proporcionado por la Organización de los Estados Americanos (OEA) en colaboración con el gobierno de Irlanda. Este manual consta de nueve requerimientos fundamentales que se detallan a continuación.

#### **1. Organización Comunitaria:**

- El sistema debe ser accesible y fácil de usar para que la comunidad pueda operarlo y mantenerlo.
- La interfaz debe ser intuitiva, permitiendo a los miembros de la comunidad designar un comité responsable del mantenimiento y operación del sistema.
- Capacitación y recursos deben estar disponibles para asegurar que los usuarios puedan gestionar el sistema de manera efectiva.

## **2. Reconocimiento de la Cuenca Menor:**

- a. Se debe desarrollar un mapa detallado que incluya las zonas vulnerables, ríos, casas, otros cuerpos de agua, y diferencias de elevación (zonas altas y bajas).
- b. El mapa debe indicar claramente las ubicaciones óptimas para la instalación de los sensores, asegurando la cobertura de todas las áreas críticas.

## **3. Medición de las Lluvias y Nivel de los Ríos:**

- a. Los sensores ultrasónicos y de lluvia deben proporcionar datos precisos y en tiempo real sobre la precipitación y el nivel de los ríos.
- b. Estos datos son esenciales para predecir inundaciones y evaluar los riesgos actuales.

## **4. Funcionamiento del Sistema:**

- a. El sistema debe ser capaz de medir con precisión las variables físicas (nivel del agua y precipitación).
- b. Además, debe interpretar estos datos, hacer pronósticos y transmitir alertas oportunas.
- c. La integración con una plataforma en la nube para procesamiento y análisis de datos es crucial para la operatividad continua y efectiva.

## **5. Evaluación de la Situación, Difusión de Alerta y Plan de Respuesta:**

- a. Un plan de emergencia debe ser desarrollado y acordado por la comunidad, detallando las acciones a seguir en caso de alerta.
- b. El sistema debe ser capaz de evaluar la situación actual y activar protocolos de respuesta mediante la difusión de alertas a través de aplicaciones como Telegram.

- c. La difusión de alertas debe ser rápida y efectiva, asegurando que toda la comunidad esté informada y preparada para actuar.

### ***Resumen del sistema***

El sistema cuenta con dos sensores que miden el nivel del río en centímetros (cm) en dos puntos estratégicos: uno en una zona con un nivel de agua bajo y otro en una zona con un nivel de agua alto. Cuando el nivel del río está normal, el primer sensor mide aproximadamente  $20\text{ cm} \pm 15\text{ cm}$  y el segundo sensor mide aproximadamente  $350\text{ cm} \pm 100\text{ cm}$ . Ambos sensores están ubicados a 10 km de la Parroquia de Valle Hermoso, lo que proporciona a la población un tiempo de 20 minutos para reaccionar en caso de una alerta de desborde.

Este sistema monitorea el río en tiempo real, lo que significa que las alertas también son inmediatas. Las alertas se envían a los usuarios interesados mediante mensajes de Telegram, utilizando un canal abierto en el que solo el sistema y el administrador pueden enviar mensajes. Esto asegura que las alertas sean claras y no se mezclen con otras comunicaciones.

El sistema envía una alerta cuando ambos sensores detectan una variación de nivel de 300 cm o más. Además, el sistema está integrado con la aplicación de Thinger, permitiendo a los usuarios ver los niveles del agua. Este monitoreo constante y la capacidad de enviar alertas inmediatas a través de Telegram mejoran significativamente la capacidad de respuesta de la comunidad ante posibles desbordes, protegiendo así a la población y sus bienes.

### ***¿Como acceder a la función de alerta?***

#### ***Instalar la Aplicación de Telegram:***

- Si aún no tienes Telegram instalado en tu dispositivo, descárgalo e instálalo desde la tienda de aplicaciones correspondiente:
  - **Android:** Google Play Store. [Descargar Telegram para android](#)
  - **iOS:** App Store. [Descargar Telegram para iPhone](#)
  - **PC/Mac:** Descarga la aplicación desde el sitio web oficial de Telegram (<https://telegram.org>).

La aplicación de Telegram se distingue por contar con el siguiente logo:



*Crear una Cuenta de Telegram:*

1. Al ingresar a la aplicación aparecerá una pantalla de bienvenida en la que se debe pulsar el botón “Empezar a chatear”.



2. La aplicación solicitará un permiso para poder recibir llamadas, al que se le debe dar acceso, para ello se pulsa en el botón de “Continuar”.



3. Inmediatamente aparecerá un aviso para permitir que la aplicación pueda administrar llamadas, se pulsará el botón de "Permitir".



4. En la pantalla siguiente se deberá ingresar el número de teléfono que debe ser el mismo que número del dispositivo donde se está instalando la aplicación, tanto en el país como en el número en sí. Una vez colocado el número se pulsa el botón de siguiente.

**Tu número de teléfono**

Por favor, confirma el código de tu país y pon tu número de teléfono.

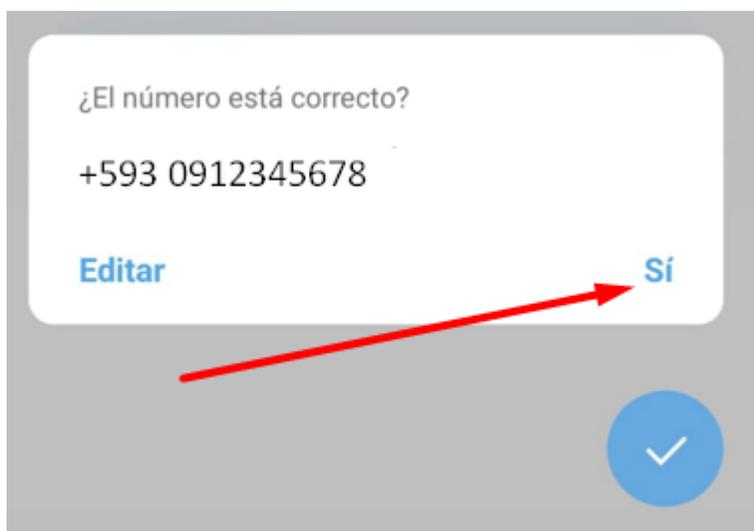
Pais  
Ecuador >

Número de teléfono  
+593 | 09 123 4567 8

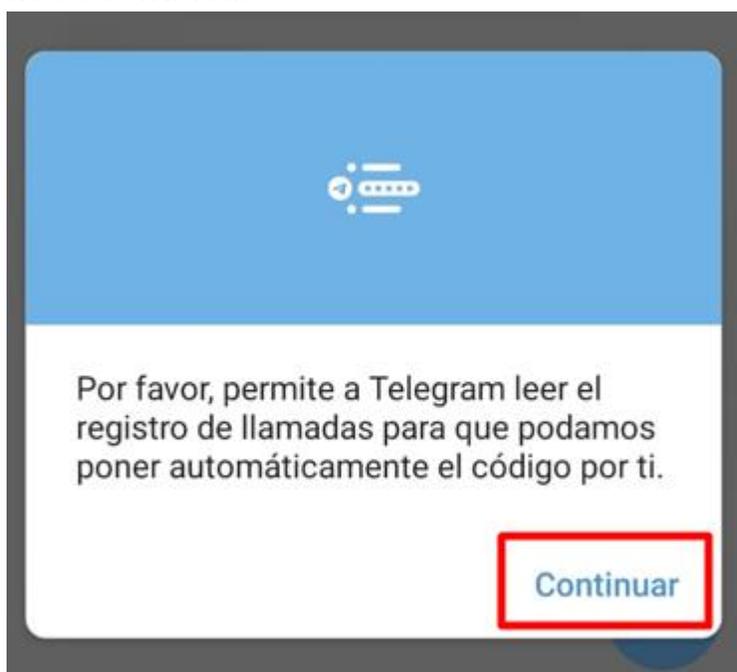
→

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
0 +	⌫	

5. Aparecerá un mensaje donde se debe confirmar si el número que se ingresó es el correcto.



6. A continuación, se presentará un mensaje en el que Telegram solicita un permiso para poder leer el registro de llamadas, ante este mensaje únicamente se continúa.



7. Al continuar, inmediatamente aparece la opción para dar acceso al permiso que solicitó Telegram. Se pulsa la opción de "Permitir" para continuar.



8. Inmediatamente ingresará a la pantalla de verificación, en este punto se debe esperar a que una llamada con número del extranjero ingrese, esta llamada no debe ser contestada.



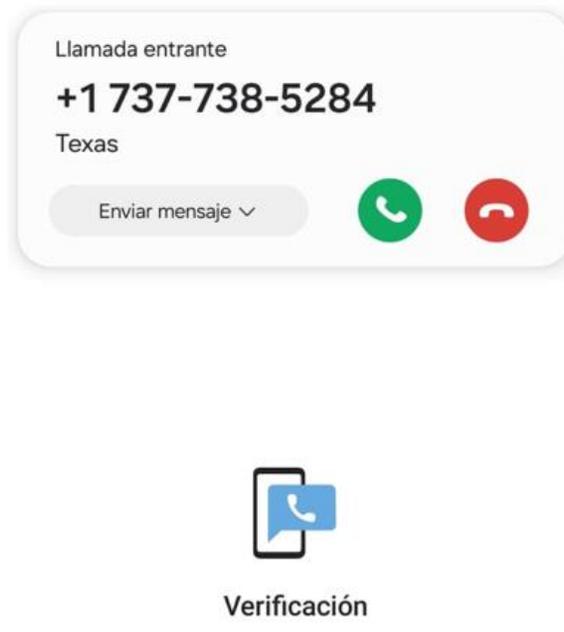
### Verificación

Estamos llamando al número +593 [REDACTED]

No contestes. Telegram hará todo el proceso automáticamente.

Puedes solicitar un SMS en 1:55

9. Al entrar la llamada, el sistema ingresará automáticamente, en caso de no hacerlo proseguir con el siguiente paso.



10. Si la llamada no verifica automáticamente el teléfono se debe pulsar la opción que se encuentra en la parte inferior la cual dice "Recibir código por SMS"



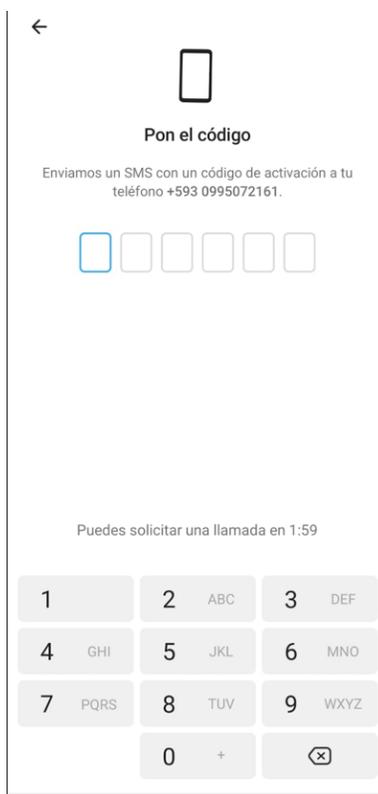
### Verificación

Estamos llamando al número +593 [REDACTED]

No contestes. Telegram hará todo el proceso automáticamente.

[Recibir código por SMS](#)

11. Se cambiará a una pantalla en la que se solicita colocar el código que será enviado por SMS, en este punto únicamente se debe esperar que llegue un mensaje y la propia aplicación se encargará de colocar el código automáticamente.



12. En la imagen a continuación se observa como al llegar el mensaje la aplicación coloca el código automáticamente.

← +7489 074385

\*\*\*

**Pon el código**

Enviamos un SMS con un código de activación a tu teléfono +593 0995072161.

0 7 4 3 8 5

Puedes solicitar una llamada en 1:50

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
0 +	✕	

13. Por último, se solicita que se coloque un nombre y el apellido del usuario.

←



**Información de perfil**

Pon tu nombre y añade una foto de perfil.

Nombre (requerido)

Usuario

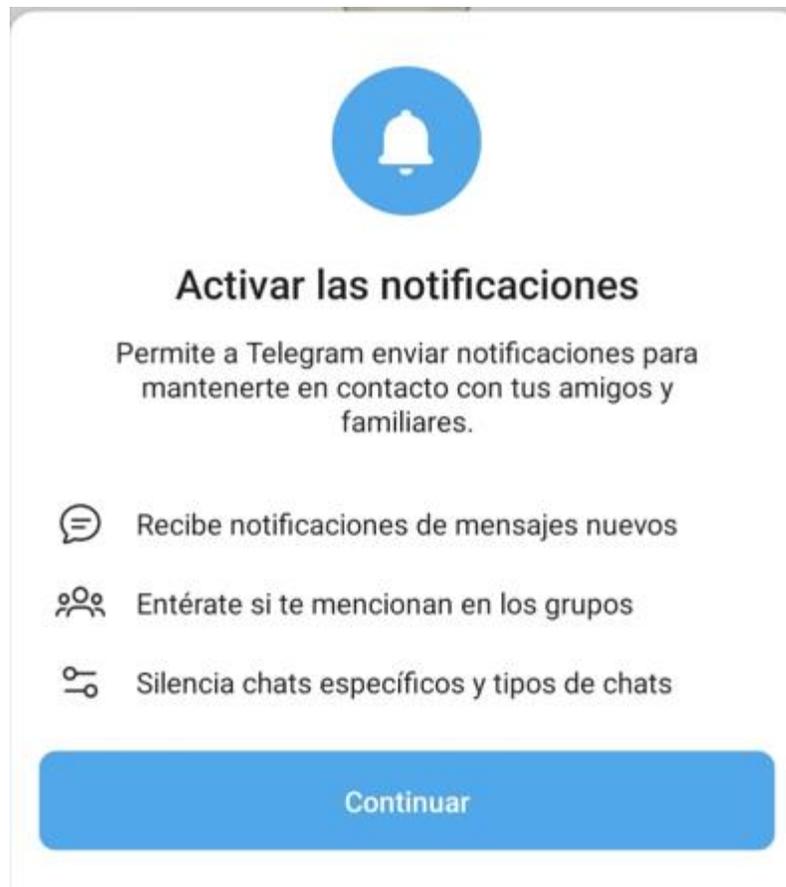
Apellidos (opcional)

Usuario

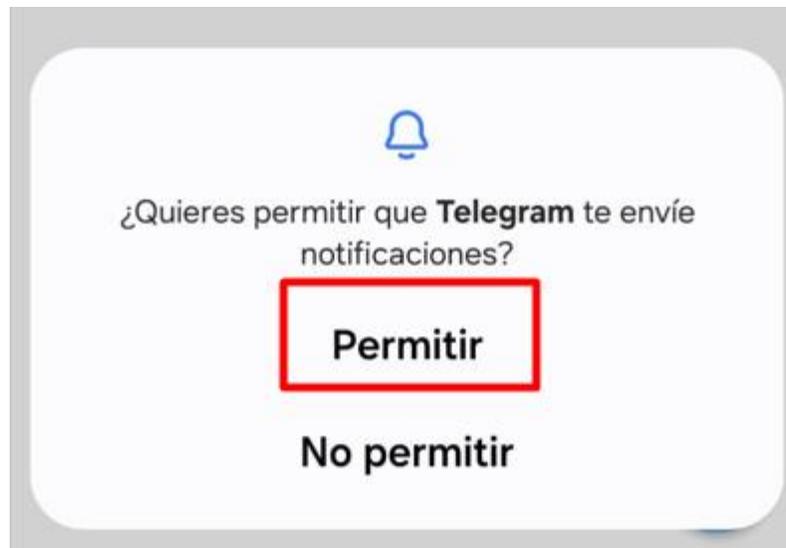
Al registrarte, aceptas los términos de servicio.



14. Ya una vez creada la cuenta e ingresado en la aplicación esta solicitará permisos para activar las notificaciones. Esta parte es importante ya que es lo que el sistema usa para enviar las alarmas por lo que se pulsa el botón de “Continuar”.



15. Nuevamente aparecerá un mensaje en el que se pregunta si se desea que Telegram pueda enviar notificaciones. Se oprime el botón de permitir.



***Acceder al Canal de Alertas:***

- Abre el navegador web en tu dispositivo y accede al siguiente enlace: [Canal de Alertas](#).
- Si el enlace se abre directamente en la aplicación de Telegram, sigue el paso 4. Si se abre en el navegador, selecciona la opción para abrir con Telegram.

***Unirse al Canal:***

- En la página del canal, pulsa en el botón "Unirse" o "Join" que aparece en la parte inferior de la pantalla.

***Revisar Mensajes del Canal:***

- Ahora estarás recibiendo mensajes de alerta en tiempo real. Revisa periódicamente el canal para estar al tanto de cualquier actualización o alerta emitida por el sistema.

Anexo 2:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXYiOiJyYXNwYmVycnkiLCJpYXQiOiJlE3M  
 TY3NDkwNjAsImp0aSI6IjY2NTM4MzA0NzM0NzBhNTUyZDA0NTc4MiIsInN2ciI6InVzLW  
 Vhc3QuYXdzLnRoaW5nZXIuaW8iLCJ1c3liOiJqZXN1czI1MTIifQ.wUb-  
 gX8vfFXK0\_G30NDDyXRulAsbY6n32RPXUOA00mA



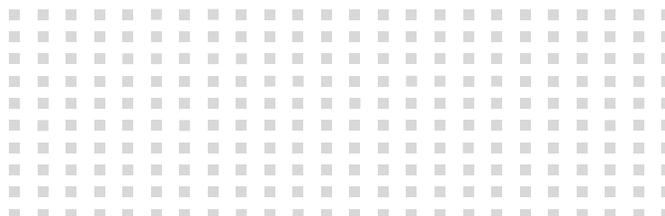
Anexo 6: Planificación de simulacro.



# Simulacro SISTEMA DE ALERTA TEMPRANA DE VALLE HERMOSO



Planificación de  
simulacro .



Jesús Navarrete  
janavarreteg@utn.edu.ec

# <sup>ec</sup>Simulacro

## SISTEMA DE ALERTA TEMPRANA DE VALLE HERMOSO

Planificación de  
simulacro.



Jesús Navarrete  
[janavarreteg@utn.edu](mailto:janavarreteg@utn.edu)

ee

## Contenido

1.....	Introducción 164
2.....	Objetivos 165
3. Cronograma.....	166

## 1. Introducción

El simulacro del **Sistema de Alerta Temprana de Valle Hermoso** tiene como objetivo evaluar la eficiencia del sistema y medir el tiempo de reacción de los usuarios ante una alerta de emergencia. Este ejercicio permitirá poner a prueba el sistema automatizado, que utiliza la plataforma Telegram para enviar notificaciones sobre el crecimiento anormal del nivel del agua en el río Blanco.

Durante el simulacro, los usuarios recibirán un mensaje de alerta que los instruirá a reunirse en el parque central de Valle Hermoso, donde se proporcionarán generalidades del sistema y se impartirán instrucciones de emergencia en coordinación con la Cruz Roja. Esta actividad busca asegurar que la comunidad esté preparada para actuar rápidamente y seguir los procedimientos adecuados en caso de una emergencia real.

El simulacro también permitirá analizar la efectividad del sistema de comunicación, el cumplimiento de los tiempos de reacción establecidos y la coordinación entre los participantes y las autoridades locales.

## **2. Objetivos**

1. Evaluar el tiempo de reacción de los usuarios:
  - Medir el tiempo que transcurre desde el envío de la alerta a través de Telegram hasta la confirmación de recepción por parte de los usuarios.
  - Registrar el tiempo que los usuarios tardan en llegar al parque central de Valle Hermoso tras recibir la alerta.
2. Verificar el correcto funcionamiento del sistema de alerta:
  - Asegurarse de que las alertas se envían correctamente y que los usuarios las reciben sin demoras.
  - Comprobar la funcionalidad del sistema automatizado para enviar mensajes constantes hasta que se reciba una confirmación.
3. Entrenar a la comunidad en los procedimientos de emergencia:
  - Instruir a los usuarios sobre las acciones que deben seguir en caso de una emergencia real.
  - Coordinar con la Cruz Roja la presentación de generalidades del sistema y los pasos a seguir durante una situación de emergencia.
4. Identificar oportunidades de mejora en el sistema de alerta:
  - Detectar posibles fallos o retrasos en el envío de alertas y en la reacción de los usuarios.
  - Recabar comentarios de los participantes para ajustar y optimizar el sistema en futuras pruebas o simulacros.
5. Fomentar la conciencia sobre la importancia de la preparación ante emergencias:

- Crear conciencia en la comunidad sobre la importancia de reaccionar rápidamente ante una alerta de emergencia.
- Reforzar la participación activa de la población en los ejercicios de simulacro para asegurar una mejor preparación en situaciones reales.

### 3. Cronograma

Hora	Actividad	Responsables	Comentarios
9:00 a. m.	Revisión de dispositivos y sistemas	Equipo técnico	Verificar que todos los usuarios estén preparados con Telegram activo.
10:00 a. m.	Preparativos en el parque central de Valle Hermoso	Equipo técnico y Cruz Roja	Asegurar que el equipo de la Cruz Roja esté listo para dar las instrucciones.
10:30 a. m.	Envío de recordatorio a los usuarios de que recibirán la alerta a las 11:00 AM	Sistema de alertas (automático)	Recordatorio para que los usuarios estén atentos a la alerta.
11:00 a. m.	Envío del mensaje de alerta: " <b>Alerta! Crecimiento anormal del nivel del agua.</b> "	Sistema de alertas (automático)	Mensajes enviados constantemente a los usuarios por Telegram.

11:00 a. m.	Registro del tiempo de envío y recepción de la alerta	Sistema de monitoreo	Comenzar a medir el tiempo de reacción desde la recepción del mensaje hasta la respuesta de los usuarios.
11:05 a. m.	Registro de confirmación de la alerta de los usuarios	Sistema de monitoreo	Registrar en la base de datos quién ha respondido y cuánto tardaron en hacerlo.
11:10 a. m.	Primer recordatorio si algún usuario no ha confirmado la alerta	Sistema de alertas (automático)	Enviar otro mensaje de recordatorio si algún usuario no ha confirmado la recepción.
11:15 a. m.	Los usuarios deben comenzar a llegar al parque central	Usuarios	Comenzar a medir el tiempo desde la confirmación hasta la llegada de los usuarios al parque.
11:20 p. m.	Cierre del registro de llegada	Sistema de monitoreo	Finaliza la medición del tiempo de llegada al parque central.
11:30 p. m.	Instrucciones y generalidades del sistema	Equipo técnico	Explicación del funcionamiento del sistema de alerta temprana a los usuarios reunidos.
11:35 p. m.	Instrucciones de emergencia junto con la Cruz Roja	Cruz Roja	Dar instrucciones sobre las acciones a tomar en caso de una emergencia real.

11:45 p. m.	Cierre de la reunión y conclusiones	Equipo técnico y Cruz Roja	Concluir la reunión, preguntas y respuestas de los usuarios.
1:00 p. m.	Análisis de datos: tiempo de reacción y llegada de los usuarios	Equipo técnico	Analizar los datos recopilados y preparar el informe final.
2:00 p. m.	Elaboración del informe final	Equipo técnico	Elaborar un informe con los resultados del tiempo de reacción y las observaciones.