



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR, PROYECTO DE INVESTIGACIÓN

TEMA:

**“SISTEMA DE MONITOREO CON LÓGICA DIFUSA PARA PREVER EL
DESBORDAMIENTO DEL RÍO COSANGA EN LA PARROQUIA DE COSANGA”**

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cibersecurity (seguridad cibernética)

AUTOR:

Anthony Miguel Panamá Chicaiza

DIRECTOR:

MSc. Fabián Geovanny Cuzme Rodríguez

Ibarra – Ecuador 2025

UNIVERSIDAD TECNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
IDENTIFICACIÓN DE LA OBRA

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

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	150117410-4		
APELLIDOS Y NOMBRES:	Panamá Chicaiza Anthony Miguel		
DIRECCIÓN:	Av. Eugenio Espejo y Tobías Mena		
EMAIL:	ampanamac@utn.edu.ec / anthony123panama@gmail.com		
TELÉFONO FIJO:	No registra	TELF. MOVIL	0967331209

DATOS DE LA OBRA	
TÍTULO:	SISTEMA DE MONITOREO CON LÓGICA DIFUSA PARA PREVER EL DESBORDAMIENTO DEL RÍO COSANGA EN LA PARROQUIA DE COSANGA.
AUTOR (ES):	PANAMÁ CHICAIZA ANTHONY MIGUEL
FECHA: AAAAMMDD	2025/07/21
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	☒ GRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN TELECOMUNICACIONES
DIRECTOR:	MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ
ASESOR:	MSC. LUIS EDILBERTO SUÁREZ ZAMBRANO

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Anthony Miguel Panamá Chicaiza, con cédula de identidad Nro. 150117410-4, 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 21 días del mes de julio de 2025

El AUTOR:

Firma


Nombre: Anthony Miguel Panamá Chicaiza

CONSTANCIAS

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

Ibarra, a los 21 días, del mes de julio de 2025

EL AUTOR:

Firma.....

Nombre: Anthony Miguel Panamá Chicaiza

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR**

Ibarra, 21 de Julio del 2025

MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

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

A handwritten signature in blue ink, appearing to read 'Fabián Cuzme', is written over a horizontal dotted line. To the left of the signature is a small circled letter 'f'.

Msc. Fabián Geovanny Cuzme Rodríguez

C.C.: 1311527012

APROBACION DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “SISTEMA DE MONITOREO CON LÓGICA DIFUSA PARA PREVER EL DESBORDAMIENTO DEL RÍO COSANGA EN LA PARROQUIA DE COSANGA” elaborado por ANTHONY MIGUEL PANAMÁ CHICAIZA, previo a la obtención del título de Ingeniero en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f).....

Ing. FABIÁN GEOVANNY CUZME RODRÍGUEZ

C.C: 1311527012

(f).....

Ing. LUIS EDILBERTO SUÁREZ ZAMBRANO

C.C: 1002304291

DEDICATORIA

Dedico este trabajo de titulación a mis padres, Miguel y María Inés, por su incondicional apoyo desde el inicio de mi carrera. Gracias por impulsarme a seguir adelante, por ser un ejemplo de esfuerzo y perseverancia, por enseñarme el sacrificio y responsabilidad, por palabras de impulso para culminar la carrera, por guiarme en el buen camino, y por estar siempre a mi lado incluso en los momentos más difíciles. Estaré eternamente agradecido.

Quiero dedicar este logro a mi hermana, María José, por su constante apoyo a lo largo de este proceso. Gracias por guiarme en la toma de decisiones, por ayudarme a superar cada obstáculo y por estar siempre presente. Mi gratitud hacia ti será infinita.

AGRADECIMIENTO

En primer lugar, agradezco a la Virgen del Quinche por darme salud y fortaleza para enfrentar cada desafío, por guiarme al camino correcto y poner en mi vida a personas increíbles para mi crecimiento personal y académico.

A mis padres y hermanas, quienes con su apoyo incondicional y sabios consejos han sido pilares esenciales para culminar esta etapa estudiantil.

A mis amigos, quienes han estado presentes desde el inicio de la carrera. No encuentro palabras suficientes para expresar mi gratitud por su compañía, apoyo y por los momentos alegres que compartimos dentro y fuera de la Universidad.

Expreso mi más profundo agradecimiento a mi director Msc. Fabián Cuzme, y a mi asesor Msc. Luis Suárez, por su valiosa orientación en el desarrollo de este trabajo de titulación, por compartir su conocimiento y dedicar su tiempo a mi formación académica. Gracias a ustedes, he adquirido no solo las herramientas necesarias para ser un buen profesional, sino también la importancia de la humildad y el compromiso de ser una persona de bien.

RESUMEN

El presente trabajo plantea un sistema de monitoreo para predecir el riesgo de desbordamiento del río Cosanga en la Parroquia de Cosanga, mediante el uso de sensores IoT, comunicación inalámbrica y lógica difusa. El objetivo principal del sistema de monitoreo es prevenir riesgos y reducir gastos relacionados a los desastres naturales, proporcionando una respuesta adecuada ante posibles desastres naturales. Como resultado, esta solución proporciona una herramienta tecnológica innovadora para la gestión del riesgo de inundaciones, mejorando la resiliencia de la comunidad frente a eventos hidrometeorológicos extremos. En base a la metodología en V, en la primera etapa se definen los requisitos técnicos con los que va a operar el sistema a partir de un análisis del territorio; seguidamente, se realizará el diseño modular que contiene la arquitectura, el funcionamiento y la integración de los nodos para el sistema de monitoreo. El sistema utiliza tres sensores, estos se encargan de medir el nivel de agua, precipitación y lluvia, los cuales monitorean el estado del río dentro de una zona estratégica. Para optimizar el funcionamiento del sistema, se aplicaron 27 reglas difusas basadas en el conocimiento de la comunidad local. Las pruebas realizadas demuestran que el sistema mantiene una comunicación inalámbrica estable y un consumo energético muy bajo, gracias al uso de la tecnología LoRa operando a una frecuencia de 915 MHz. Como resultado, se alcanzó una exactitud del 96% y una precisión del 100%, según los valores obtenidos en la matriz de confusión. Además, se comprobó que el sistema permanece activo y funcional incluso durante eventos de precipitación intensa y aumento del caudal del río, emitiendo notificaciones y alertas sonoras en caso de detectar niveles críticos. El sistema logró demostrar ser una herramienta eficiente dentro del área de alerta temprana, permitiendo a la comunidad y autoridades actuar rápidamente para reducir riesgos y pérdidas económicas. El proyecto contiene bases para aplicar en otras zonas con condiciones climáticas similares, aportando a la adopción en la comunidad ante desastres naturales.

Palabras clave: Lógica difusa, LoRa, desbordamiento de ríos, sistema de alerta temprana, inteligencia artificial, monitoreo ambiental, redes de sensores inalámbricos, IoT.

ABSTRACT

This study proposes a monitoring system designed to predict the risk of overflow of the Cosanga River in the Cosanga Parish, using IoT sensors, wireless communication, and fuzzy logic. The primary objective of the monitoring system is to prevent risks and reduce costs associated with natural disasters by providing an adequate response to potential emergency events. As a result, this solution offers an innovative technological tool for flood risk management, enhancing community resilience to extreme hydrometeorological events. Based on the V-model development methodology, the first stage involves defining the technical requirements for system operation through territorial analysis. Subsequently, a modular design is developed, detailing the system's architecture, functionality, and node integration. The system utilizes three sensors responsible for measuring water level, rainfall, and precipitation, which monitor the river's condition within a strategically selected area. To optimize system performance, 27 fuzzy rules were implemented, developed from local community knowledge. Experimental testing demonstrated that the system maintains stable wireless communication and very low energy consumption, enabled by the use of LoRa technology operating at a frequency of 915 MHz. As a result, the system achieved an accuracy of 96% and a precision of 100%, according to the values obtained in the confusion matrix. Furthermore, the system remained active and functional even during periods of intense rainfall and increased river flow, issuing notifications and triggering audible alerts when critical thresholds were detected. The solution proved to be an efficient early warning tool, allowing the community and authorities to respond promptly to reduce risks and economic losses. The proposed system establishes a replicable foundation for implementation in other regions with similar climatic conditions, contributing to community preparedness and resilience in the face of natural disasters.

Keywords: Fuzzy logic, LoRa, river overflow, early warning system, artificial intelligence, environmental monitoring, wireless sensor networks, IoT.

INDICE DE CONTENIDO

1. CAPITULO I: Antecedentes	22
1.1. Tema.....	22
1.2. Problema.....	22
1.3. Justificación.....	24
1.4. Objetivos.....	25
1.4.1. Objetivo General	25
1.4.2. Objetivos específicos	25
2. Capitulo II: Fundamentos Teóricos.....	27
2.1 Aspecto general de la parroquia Cosanga.....	27
2.1.1. Componente social, económico y político territorial.....	27
2.1.2. Componente Político-Geográfica	28
2.1.3. Clima	29
2.1.4. Hidrográfica menor de rio Cosanga	32
2.1.5. Análisis de riesgos de desastres en Ecuador	33
2.2. Sistema de monitoreo	35
2.2.1. Internet de las cosas	35
2.2.2. Componentes de Hardware	35
2.2.3. Componentes de Software	36
2.2.4. Arquitectura IoT	38
2.2.5. Tecnologías inalámbricas.....	39
2.3. Sistemas Inteligentes.....	43
2.3.1. Aprendizaje Automático	43
2.3.2. Inteligencia Artificial	43
2.3.3. Norma IEEE 29148	44
2.3.4. Lógica difusa.....	44
3. Capitulo III: Diseño del sistema	51
3.1. Metodología.....	51
3.1.1. Modelo en V	51
3.2. Análisis del proyecto.....	53
3.2.1. Situación Actual.....	53
3.2.2. Ubicación.....	54
3.2.3. Características del sistema	56
3.2.4. Método de investigación para la obtención de requisitos.	57
3.3. Descripción general del sistema.....	59
3.4. Requerimientos	60

3.4.1.	Requerimientos de stakeholders	61
3.4.2.	Requerimientos iniciales del sistema	63
3.4.3.	Requerimientos de arquitectura	66
3.4.4.	Aspectos técnicos del sistema de monitoreo.	71
3.5.	Elección de hardware y software para el sistema	71
3.5.1.	Hardware destinado para el sistema	71
3.5.2.	Software destinado para el sistema.....	90
3.6.	Diseño del sistema.....	95
3.6.1.	Diagramas de flujo del sistema	97
3.6.2.	Implementación de Hardware.....	101
3.6.2.1.	Bloque de Percepción.....	101
3.6.3.	Bloque de Red	105
3.6.4.	Bloque de Aplicación y diseño de software	106
3.6.5.	Diseño en 3D de Nodos del Sistema	113
3.6.6.	Diseño de red inalámbrica	117
3.6.7.	Diseño de operación con lógica difusa	128
4.	CAPITULO IV: Implementación y Pruebas de funcionamiento	141
4.1.	Integración de Componentes	141
4.1.1.	Armado de Nodos.....	141
4.2.	Verificación de configuración de elementos del sistema.....	144
4.2.1.	Verificación de configuración Nodo Sensor.....	144
4.2.2.	Verificación de configuración de Nodo Coordinador.....	148
4.2.3.	Verificación de Nodo Actuador	150
4.3.	Entrenamiento de Datos.....	153
4.3.1.	Entrenamiento del modelo.....	153
4.3.2.	Obtención del modelo entrenado	160
4.4.	Instalación del sistema de monitoreo	161
4.4.1.	Instalación de Nodo Sensor	162
4.4.2.	Instalación de Nodo Coordinador.....	163
4.4.3.	Instalación de Nodo Actuador	163
4.5.	Pruebas de funcionamiento del sistema.....	164
4.5.1.	Pruebas de comunicación inalámbrica.....	165
4.5.2.	Pruebas de Funcionamiento de Nodo Sensor	168
4.5.3.	Captura y Visualización de datos del nodo sensor	169
4.5.4.	Prueba de funcionamiento de nodo coordinador	171
4.5.5.	Prueba de funcionamiento de Sitio web y Aplicación móvil.....	172

4.5.6. Funcionamiento del sistema de monitoreo	175
4.6. Análisis de datos	180
4.7. Verificación y presentación del sistema de monitoreo	184
4.7.1. Recepción y visualización de datos en sitio web	184
4.7.2. Verificación de Notificaciones en Aplicación	185
4.8. Rendimiento del sistema	187
4.8.1. Matriz de confusión.....	187
4.9. Medición de Tiempo de Respuesta	191
4.10. Socialización del sistema en la comunidad	192
4.11. Análisis Costo/Beneficio del Sistema.....	194
4.11.1. Costo de Hardware	194
4.11.2. Costo de Software	196
4.11.3. Costo de Ingeniería	197
4.12. Discusión.....	199
Conclusiones y Recomendaciones.....	201
Conclusiones	201
Recomendaciones	203
REFERENCIAS.....	205
ANEXOS.....	208
ANEXO 1. Documento de aceptación y certificado emitido por el GADPR Cosanga.....	209
ANEXO 2. Cuestionario aplicado al presidente de la junta parroquial.....	210
ANEXO 3. Ficha de requerimientos.....	212
ANEXO 4. Diseño en 3D de sistema de monitoreo.....	215
ANEXO 5. Simulación en Radio Mobile.....	218
ANEXO 6. Socialización de sistema de monitoreo.....	223
ANEXO 7. Instalación de nodos.	225
ANEXO 8. Conversión de modelo entrenado.....	228
ANEXO 9. Manual de Usuario.	232

INDICE DE FIGURAS

Figura 1. Temperaturas media y precipitaciones.	30
Figura 2. Temperaturas máximas en la parroquia de Cosanga.	31
Figura 3. Mapa de demarcación hidrográfica parroquia Cosanga.	32
Figura 4. Sistema de proceso básico IoT 38	38
Figura 5. Clasificación de redes inalámbricas. 39	39
Figura 6. Arquitectura básica de la tecnología LoRaWAN..... 42	42
Figura 7. Sistema de control difuso..... 46	46
Figura 8. Función de pertenencia triangular. 48	48
Figura 9. Función de pertenencia Signoidal..... 49	49
Figura 10. Función de pertenencia Trapezoidal. 49	49
Figura 11. Función de pertenencia Gaussiana. 50	50
Figura 12. Río Cosanga cerca de la parroquia Cosanga..... 54	54
Figura 13. Ubicación geográfica del río Cosanga en Google Earth..... 55	55
Figura 14. Cauce del río Cosanga 56	56
Figura 15. Zona seleccionado para recolectar datos con los sensores. 57	57
Figura 16. Arquitectura del sistema de monitoreo..... 95	95
Figura 17. Diagrama de flujo de nodo sensor. 98	98
Figura 18. Diagrama de flujo de nodo coordinador. 99	99
Figura 19. Diagrama de conexión de sensor JSN-SR04T..... 102	102
Figura 20. Diagrama de conexión de sensor FC-37..... 102	102
Figura 21. Diagrama de conexión de sensor infrarrojo..... 103	103
Figura 22. Diagrama de conexión general del nodo sensor. 104	104
Figura 23. Diseño de comunicación inalámbrica de los dos nodos. 105	105
Figura 24. Diagrama de conexiones de nodo sensor..... 106	106
Figura 25. Diagrama de secuencia de recopilación de datos..... 106	106
Figura 26. Diagrama de base de datos NoSQL de Firebase Firestore. 108	108
Figura 27. Diseño de frontend del sistema. 109	109
Figura 28. Diagrama de flujo de notificación del sistema. 110	110
Figura 29. Diseño de interfaz de notificación celular del sistema..... 110	110
Figura 30. Diagrama de flujo de nodo actuador..... 111	111
Figura 31. Diagrama de conexiones de nodo actuador..... 113	113
Figura 32. Diseño en 3D de nodo sensor vista interna. 113	113
Figura 33. Diseño en 3D de nodo sensor vista externa..... 114	114
Figura 34. Diseño en 3D de nodo coordinador vista externa. 115	115

Figura 35. Diseño en 3D de nodo actuador vista interna.	115
Figura 36. Diseño en 3D de nodo actuador vista externa.....	116
Figura 37. Entorno de nodos con conexión inalámbrica.....	117
Figura 38. Distancia de Nodos hacia la parroquia de Cosanga.	123
Figura 39. Resultado de simulación de nodo sensores y nodo coordinador.....	124
Figura 40. Zona geográfica de transmisión entre nodo sensor y nodo coordinador.	125
Figura 41. Arquitectura del sistema difuso.	128
Figura 42. Diagrama de flujo del diseño de operación difusa.....	129
Figura 43. Distribución borrosa del sistema de monitoreo	132
Figura 44. Gráfica de la función de pertenencia de nivel de río.	134
Figura 45. Gráfica de la función de pertenencia de precipitación.	135
Figura 46. Gráfica de la función de pertenencia de velocidad.....	136
Figura 47. Gráfica de la función de pertenencia de grado de riesgo.	137
Figura 48. Armado del bloque de recolección de datos.	142
Figura 49. Armado final de Nodo Sensor.	142
Figura 50. Armado final de Nodo Coordinador.	143
Figura 51. Armado final de Nodo Actuador.	144
Figura 52. Parámetros de inicialización de Cx inalámbrica de nodo sensor.....	145
Figura 53. Configuración de transmisión y recepción de datos en nodo sensor.	146
Figura 54. Configuración de operación de reglas de pertenencia.	147
Figura 55. Configuración del proceso de defuzzificación.....	148
Figura 56. Función de visualización de datos de Rx mediante protocolo MQTT.....	148
Figura 57. Procesamiento de control de IA.....	149
Figura 58. Librerías de Nodo Actuador.....	150
Figura 59. Configuración de conexión a servicio MQTT.	151
Figura 60. Configuración de verificación de acceso a la red WiFi.	152
Figura 61. Configuración de estado de alarma.	153
Figura 62. Carga de Dataset en Google Colab.....	154
Figura 63. Modelo de ingreso de datos.	154
Figura 64. Parámetros de entrenamiento de datos.	155
Figura 65. Configuración de modelo secuencial para el entrenamiento.	157
Figura 66. Precisión del modelo durante el entrenamiento.	158
Figura 67. Comportamiento de precisión de entrenamiento y validación por épocas.	158
Figura 68. Comportamiento pérdida de entrenamiento y validación por épocas.....	159
Figura 69. Modelo Convertido a formato C.	161
Figura 70. Instalación de Nodo Sensor.	162

Figura 71. Instalación de Nodo Coordinador.....	163
Figura 72. Instalación de Nodo Actuador.....	164
Figura 73. Inicialización de nodo coordinador.	166
Figura 74. Prueba de frecuencia LoRa	166
Figura 75. Visualización de espectro LoRa	167
Figura 76. Resultado de bloque de Nodo Sensor.	168
Figura 77. Interfaz de inicio de recolección de datos.	169
Figura 78. Captura y visualización de datos del nodo sensor.....	170
Figura 79. Inicialización de nodo coordinador.	171
Figura 80. Prueba de funcionamiento en aplicación móvil.....	172
Figura 81. Prueba de funcionamiento de sitio web del sistema de monitoreo.....	173
Figura 82. Funcionamiento de Data Picker del sitio web.....	174
Figura 83. Prueba de alerta del sistema.....	175
Figura 84. Prueba de notificación de sistema de monitoreo.	176
Figura 85. Prueba de alerta del sistema de monitoreo.....	177
Figura 86. Prueba de finalización de tiempo de alerta.	178
Figura 87. Prueba de finalización de tiempo de alerta en app.	179
Figura 88. Promedio de nivel de agua de Febrero-Junio 2025.	181
Figura 89. Promedio de nivel de lluvia de Febrero-Junio 2025.....	181
Figura 90. Promedio de nivel de velocidad de Febrero-Junio 2025.....	182
Figura 91. Promedio de total de datos de Febrero-Junio 2025.	183
Figura 92. Sitio web de visualización de datos.....	184
Figura 93. Verificación de notificaciones de alerta de app móvil.	185
Figura 94. Verificación de notificación dentro del monitor serial.....	186
Figura 95. Verificación de datos procesados en app móvil.....	186
Figura 96. Socialización con los vocales del GADPR Cosanga.	193
Figura 97. Presentación de Sistema de monitoreo.....	193

INDICE DE TABLAS

Tabla 1. Arquitectura de cuatro capas	38
Tabla 2 Clasificación de redes inalámbricas	40
Tabla 4 Definición de abreviaturas	61
Tabla 5 Actores involucrados.....	61
Tabla 6 Requerimientos de stakeholders	62
Tabla 7 Requerimientos iniciales del sistema.....	63
Tabla 8 Requerimientos de arquitectura.....	66
Tabla 9 Cuadro comparativo de sensores ultrasónicos.....	73
Tabla 10 Análisis comparativo de sensores ultrasónicos.	73
Tabla 11 Cuadro comparativo de sensores de precipitación.	75
Tabla 12 Análisis comparativo de sensores de precipitación.....	76
Tabla 13 Cuadro comparativo de sensor de rotación.....	78
Tabla 14 Análisis comparativo de sensor de rotación.	78
Tabla 15. Cuadro comparativo de microcontroladores.....	79
Tabla 16. Análisis comparativo de microcontroladores.	80
Tabla 17. Cuadro comparativo de chips LoRa.	82
Tabla 18. Análisis comparativo de Chips LoRa.	83
Tabla 19. Cuadro comparativo de Panel Solar.....	84
Tabla 20. Análisis comparativo de Panel Solar.	85
Tabla 21. Cuadro comparativo de batería.....	86
Tabla 22. Análisis comparativo de batería.	86
Tabla 23. Análisis de consumo eléctrico de nodo sensor.....	87
Tabla 24. Requerimientos de software para el entorno de desarrollo del sistema	90
Tabla 25. Requerimientos de software para la recolección de datos	91
Tabla 26. Requerimientos de software para la interfaz gráfica.....	92
Tabla 27. Requerimientos de software para el entrenamiento de datos.....	93
Tabla 28. Requerimientos de herramienta de notificaciones	94
Tabla 29. Ubicación geográfica de Nodos	119
Tabla 30. Comparación de resultados calculados y simulados.	126
Tabla 31. Relación en función de distancia y RSSI.....	127
Tabla 32. Variables de Universo de discurso	130
Tabla 33. Conjuntos de funciones de pertenencia del sistema.....	132
Tabla 34. Conjuntos de funciones de pertenencia del sistema.....	138
Tabla 35. Lista de pruebas realizadas dentro del sistema.....	164

Tabla 36. Modelo de representación de matriz de confusión.	188
Tabla 37. Predicciones para la matriz de confusión.	188
Tabla 38. Modelo de representación de matriz de confusión del sistema.	191
Tabla 39. Costo de Hardware de nodo sensor y coordinador.	194
Tabla 40. Costo de Hardware de alerta.	195
Tabla 41. Costo de Software para el sistema de monitoreo.	196
Tabla 42. Costo de ingeniería.	197
Tabla 43. Costo general del sistema.	198

1. CAPITULO I: Antecedentes

En este capítulo de antecedentes, se menciona sobre el desarrollo del proyecto de titulación, en donde se presenta el tema, problema, objetivos, alcance y justificación, donde se respalda la importancia de desarrollar el presente proyecto.

1.1. Tema

Sistema de monitoreo con lógica difusa para prever el desbordamiento del río Cosanga en la parroquia de Cosanga.

1.2. Problema

En el Ecuador, la época lluviosa inicia de marzo a julio, donde se esperan fuertes lluvias que provocan el crecimiento de los ríos y desbordamientos, como resultado de las intensas lluvias, se produce pérdidas subsecuentes y vidas humanas. Las inundaciones y las lluvias fuertes son las amenazas más frecuentes y las que mayor incidencia han tenido en el territorio nacional durante los últimos 35 años (correspondiendo al 44 % del territorio nacional), seguidas por los incendios forestales (26 %), los deslizamientos de tierra (11 %), oleajes (9 %), terremotos y tsunamis (8%) y las sequías (2 %). (El Universo, 2023)

La parroquia Cosanga se encuentra en la región de la Cordillera Subandina, que es la parte occidental de la gran llanura amazónica. Esta zona se caracteriza por tener montañas, colinas y paisajes irregulares que son parte de la estribación de la cordillera oriental andina. La parroquia Cosanga presenta un clima mayormente Tropical Mesotérmico Húmedo que cubre un área de 18810,8 ha. que representa el 47,12% del territorio, seguido de un clima Ecuatorial Frío de Alta Montaña (páramo) que cubre un área de 13348,97 ha. que representa el 33,44% del territorio en estudio; el clima Trópico Frio Húmedo abarca un área de 7627,41 ha representado por un 19,10% del territorio de la parroquia, y finalmente tenemos un clima

Tropical Mesodérmico Húmedo que cubre un área de 137,55 que representa un 0,34% del territorio de la parroquia Cosanga. (*PDOT-Cosanga*, 2019)

La población total de la parroquia de Cosanga para el año 2020 es de 556 habitantes, de acuerdo con las proyecciones referenciales de población a nivel cantonal-parroquial, periodo 2010-2020. Las principales actividades económicas de la parroquia rural de Cosanga están divididas en el sector agropecuario o primario; seguido del sector de servicios o terciario. Entre las actividades económicas de mayor representatividad en la parroquia tenemos a la Ganadería, Agricultura y el Turismo. La parroquia Cosanga por su situación geográfica y topografía altamente accidentada es propensa a que se produzcan fenómenos como: inundaciones, deslizamientos, sismos y otros fenómenos de tipo natural. Al tratarse de un área amazónica, no se puede contar con fechas exactas de duración del invierno por lo que, al no contar con registros estadísticos del crecimiento del caudal del río, en la parroquia Cosanga es imposible saber en qué meses se presenta mayor influencia de agua y por medio de ello llevar un respectivo control en los meses de invierno.

El desbordamiento de los ríos provoca inundaciones que perjudican a las personas que tienen viviendas a sus alrededores, poniendo en riesgos sus vidas. En estos momentos no hay una herramienta de bajo costo que les permita monitorear el estado del río, por lo cual no se puede saber en qué condiciones se encuentra y si representa un peligro inminente para las personas que están en sus cercanías. Contar con un sistema de monitoreo basado con lógica difusa, permite llevar un registro del crecimiento del caudal del río y en caso de existir un nivel fuera de lo normal, los residentes pueden desplazarse a áreas resguardadas. De esta forma la implementación de un sistema basado en lógica difusa es de gran ayuda para disminuir los desastres por desbordamiento evitando así la pérdida de vidas humanas, como también pérdidas subsecuentes.

1.3. Justificación

Napo tiene una extensión de 13.294,8 kilómetros cuadrados, está conformada por cinco cantones: Archidona, Carlos Julio Arosemena Tola, El Chaco, Quijos y Tena. Constituyen sus principales atractivos turísticos la Cascada de Hollín, las Termas y Lagunas de Papallacta, el Puerto Misahualli, las Cavernas de Jumandi, el Parque la Islas y el río Jatunyaku.

Napo al ser parte de la iniciativa ODS, contó con un financiamiento de la Unión Europea y tiene una duración de tres años (2017-2019). La iniciativa busco contribuir a la mejora integral de las condiciones y medios de vida en el país a través del cumplimiento de las ODS mediante su incorporación en las políticas públicas nacionales y locales, y en el fortalecimiento de la sociedad civil y de los Gobiernos Autónomos Descentralizados en su implementación y seguimiento. (Andrade & Peña, 2020)

En esta provincia se desarrollan prácticas de turismo comunitario, como la que impulsa la Red Ricancie integrada por 10 comunidades indígenas que se posicionan a través de su auto comercialización de sus destinos y servicios turísticos. (Ministerio de Turismo, 2014)

Con el propósito de fortalecer las capacidades de las Unidades de Gestión de Riesgos de los Municipios de la provincia de Napo y de los Cuerpos de Bomberos, el Servicio Nacional de Gestión de Riesgos y Emergencias (SNGRE) realizó el taller denominado “Flujo de Información durante Emergencias y Desastres”. (López, 2023)

La cuenca del río Cosanga y las subcuencas son considerados aptos para la pesca deportiva, para ello es necesario el contacto con un guía local y el permiso de los propietarios por donde se ingresa. El río Cosanga está considerado uno de los 10 ríos más aptos del país para la práctica de deportes extremos como el tubing, kayak y rafting, categorizado a nivel internacional de clase III que significa un río de caudal con dificultad menor. (Napo Travel, 2022)

La Parroquia Cosanga por su situación geográfica y topografía altamente accidentada es propensa a que se produzcan fenómenos como: inundaciones, deslizamientos, sismos y otros fenómenos de tipo natural, otra de las causas de daños al ambiente es los producidos por un origen antrópica entre los que se destacan, extracción de maderas, intervención y explotación de los bosques para el uso agropecuario, principalmente y entre otras formas de alteración de la zona natural. El impacto que se produce por inundaciones ocasiona ganado enfermo y débil para su producción; otro impacto generado por las inundaciones es los derrumbes, donde provoca pérdida de suelos productivos. (PDOT-Cosanga, 2019)

En la actualidad, resulta ser muy necesario implementar un sistema de monitoreo y alerta temprana en dicho sector, con el fin de recolectar información para identificar las temporadas donde existe más crecimiento del nivel del río Cosanga y alertar a las autoridades sobre cualquier evento de riesgo que se presente hacia la Parroquia.

La implementación de este proyecto permitirá reducir los daños a las viviendas, la pérdida de la producción agrícola y turística, y prevenir pérdidas humanas. Gracias a la rapidez del sistema de alerta temprana, se esperan resultados positivos. Además, las autoridades tendrán el tiempo necesario para tomar decisiones, llevar a cabo acciones y desarrollar planes de prevención de inundaciones en la parroquia de Cosanga.

1.4. Objetivos

1.4.1. Objetivo General

Implementar un sistema de monitoreo, basado en lógica difusa para prevenir y mitigar los efectos de las inundaciones en la parroquia de Cosanga de la provincia de Napo.

1.4.2. Objetivos específicos

- Realizar un diagnóstico territorial de la parroquia en base a los componentes sociales, económicos y físicos.

- Analizar herramientas disponibles que se acoplen a la arquitectura de este proyecto.
- Implementar el sistema de monitoreo inteligente, asegurando la conformidad con los requisitos establecidos mediante la integración eficiente de los elementos de hardware y software.
- Validar la funcionalidad del sistema monitoreo, para evidenciar el envío, recepción y tratamiento de datos mediante lógica difusa.

2. Capítulo II: Fundamentos Teóricos

Dentro de este capítulo, se establecerán fundamentos teóricos para aclarar diversos temas vinculados al proyecto. Se definirán los conceptos que interactúan con las condiciones climáticas actuales de la parroquia, sobre el sistema basado en lógica difusa, las tecnologías inalámbricas, las normas, las placas y los lenguajes de programación que resultarán apropiados para la implementación del proyecto. Se busca proporcionar una comprensión sólida y detallada de los elementos clave que forman parte del desarrollo del proyecto.

2.1 Aspecto general de la parroquia Cosanga

Cosanga, una de las cinco parroquias rurales del cantón Quijos en Napo, abarca 399,25 km² con una topografía irregular y precipitaciones que favorecen la agricultura y la ganadería. Sus dos áreas protegidas, la Reserva Ecológica Antisana y el Parque Nacional Sumaco Napo-Galeras, albergan una gran biodiversidad y ofrecen oportunidades para el ecoturismo. La ganadería lechera es la principal actividad económica, seguida por la agricultura y el ecoturismo emergente. La protección de las áreas protegidas y el desarrollo sostenible de las actividades económicas son claves para el futuro próspero de Cosanga.

2.1.1. Componente social, económico y político territorial

La normativa vigente ha permitido al GAD Parroquial la conformación implementación de los diferentes sistemas y consejos de planificación y participación ciudadana, en el territorio; pero revisado las normativas internas existentes en el GAD Parroquial, se constató que no existen políticas públicas de igualdad para los planes, programas y proyectos establecidos en los PDOT, más bien se las ha realizado de manera empírica sin considerar la normativa o guías pertinentes. En la actual administración se considera trabajar en iniciativas para mejorar la coherencia normativa de la localidad para el desarrollo sostenible, continuando con la participación ciudadana, eje fundamental del desarrollo local de la parroquia. (PDOT-Cosanga, 2019)

Dentro del análisis social de la Parroquia Cosanga se busca fortalecer su identidad cultural y mejorar las condiciones de vida de sus habitantes, especialmente de los grupos prioritarios. Se fortalecerán programas culturales y se mejorará el acceso a servicios básicos para los grupos más vulnerables, promoviendo la equidad e inclusión en el territorio. Además, se busca un impacto positivo en la comunidad y el desarrollo sostenible de Cosanga, construyendo un futuro más próspero y equitativo para todos. Por otro lado, el componente de análisis económico busca comprender los patrones de producción y consumo, la complementariedad y competencia entre sectores productivos, las opciones para el desarrollo de emprendimientos, las potencialidades y recursos disponibles, y los factores de concentración y redistribución de la riqueza.

Dentro de la parroquia se forman diversas actividades económicas abarcando la agricultura y turismo hasta los servicios públicos, construcción y la ganadería. El análisis del componente económico no solo se traduce en la identificación de las fortalezas y debilidades, sino también en la visualización de oportunidades clave en el sistema económico local, impulsando la creación de empleos, la generación de ingresos, la reducción de desigualdades y el bienestar de la población. El objetivo final es construir un futuro más próspero y equitativo para la parroquia. (PDOT-Cosanga, 2019)

2.1.2. Componente Político-Geográfica

Según el documento de la Agenda Zonal de la Secretaría Nacional de Planificación y Desarrollo, declara que la parroquia de Cosanga se encuentra dentro de la Zona 2 de zonas administrativas de planificación del país, la planificación está estrechamente vinculada a la escala y a las características específicas del territorio que se pretende desarrollar y organizar.; este tipo de zona de planificación va conformada por las provincias de Napo, Orellana y Pichincha (excepto el Distrito Metropolitano de Quito). Limita, al norte, con las provincias de Sucumbíos, Imbabura y Esmeraldas (Zona 1); al sur, con Cotopaxi, Tungurahua y Pastaza

(Zona 3); hacia el oeste, con Santo Domingo de los Tsáchilas (Zona 4) y parte de la provincia de Esmeraldas (Zona 1); y, al este, con la República de Perú. Tiene una superficie de 39.542,58 km².(Morona, 2013)

La parroquia de Cosanga es una de las cinco parroquias rurales que conforman el cantón Quijos. Este cantón está situado en las faldas de la cordillera oriental y pertenece a la provincia de Napo. Dada su extensión, Cosanga representa el 25% de la superficie total del cantón, abarcando una extensión de 399,25 kilómetros cuadrados, lo que la convierte en la parroquia más extensa de Quijos.

El Gobierno Autónomo Descentralizado Parroquial Rural (GADPR) de Cosanga posee una extensa red hidrográfica que abarca varios ríos, quebradas y lagunas, distribuida a lo largo de toda la superficie de la parroquia. Las lagunas en Cosanga se encuentran en menor proporción. Estas fuentes de agua han sido beneficiosas para los habitantes de la parroquia, ya que han respaldado el desarrollo de actividades económicas, especialmente en lo que respecta al abastecimiento de agua para consumo humano.(PDOT-Cosanga, 2019)

2.1.3. Clima

Ecuador al encontrarse situado sobre la línea ecuatorial, genera escasa variación estacional a lo largo del año. Se identifican únicamente dos estaciones bien definidas: la estación húmeda o invierno, y la estación seca o verano. La duración de estas estaciones presenta variaciones según la región. En la zona Costa, las lluvias comienzan en diciembre y persisten hasta mayo, mientras que la estación seca abarca de junio a noviembre. En los Andes, la temporada de lluvias se extiende de octubre a mayo, seguida por la estación seca de junio a septiembre. En la región amazónica, existen discrepancias entre el norte y el sur. En la Amazonía Centro Norte (provincia de Napo), la temporada de lluvias abarca de marzo a mayo y en octubre, mientras que la estación seca se registra de diciembre a febrero. En el resto de la

Amazonía, el patrón estacional es similar al de la región Andina. El clima de Ecuador al tener un rango altitudinal de 0 a 6300m, genera temperaturas de 0 a 26°C en promedio anual.(Varela, 2022)

2.1.3.1. Promedio climático de Cosanga

La parroquia de Cosanga contiene un clima ecuatorial, esto quiere decir que en ocasiones el clima va a ser muy caluroso, húmedo y lluvioso durante todo el año. La temperatura promedio anual en Cosanga se sitúa en 17,2° C, con una precipitación anual media de 2624 mm. Se registran alrededor de 14 días al año sin lluvia, manteniendo una humedad promedio del 90%. El Índice UV en esta área alcanza un nivel de 3.

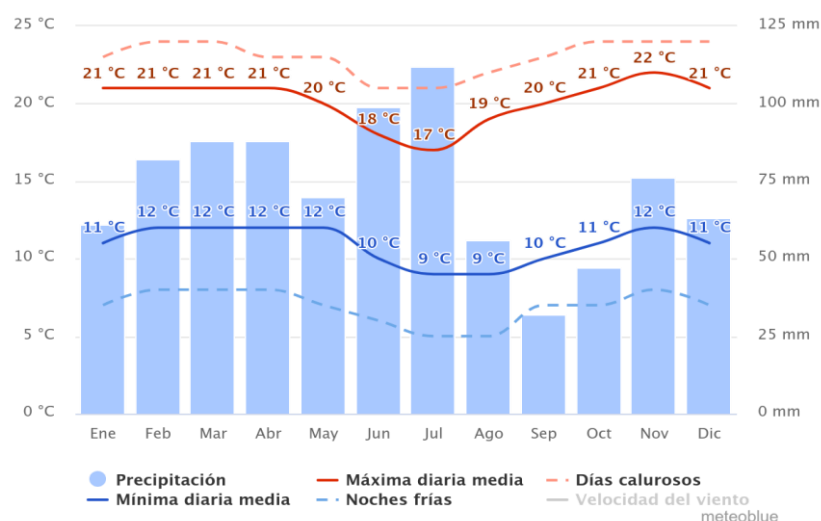
Los promedios climáticos dentro de la parroquia de Cosanga son:

- Temperatura máxima: 12° y 14°C
- Mes más cálido: enero (14°C)
- Mes más frío: marzo (12°C)
- Temperatura nocturna: 4° y 7°C
- Lluvia: llueve 337 días (hasta el día más seco puede tener mucha lluvia).

Dentro de la Figura 1, se puede observar las temperaturas máximas y mínimas, y la cantidad de precipitaciones por mes durante todo el año.

Figura 1.

Temperaturas media y precipitaciones.



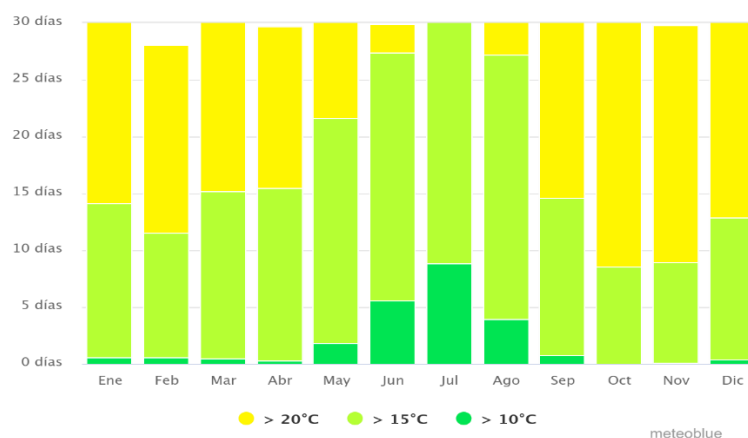
Fuente: (Meteoblue, 2023)

Las temperaturas dentro de la Parroquia de Cosanga pueden variar según las estaciones del año, pero al ser un clima húmedo en un 90%, casi todos los días van a existir lluvia. Dentro de la siguiente imagen se puede visualizar de mejor manera las temperaturas máximas dentro de la parroquia.

En la Figura 2, se puede observar el promedio máximo de las temperaturas de cada mes en la parroquia de Cosanga.

Figura 2.

Temperaturas máximas en la parroquia de Cosanga.



Fuente: (Meteoblue, 2023)

2.1.4. Hidrográfica menor de río Cosanga

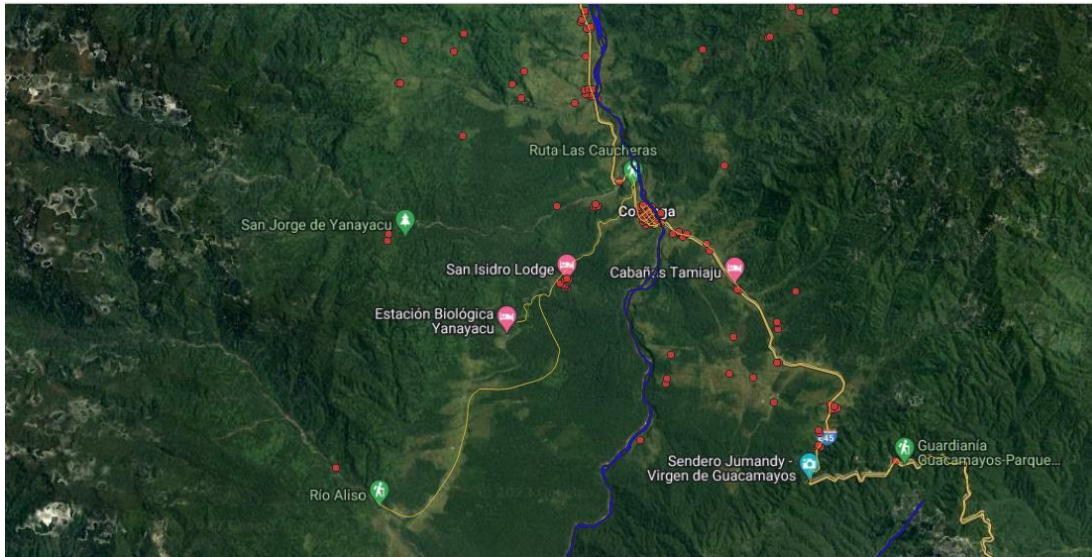
La posición estratégica de la cuenca del río Cosanga adquiere una significativa relevancia para Ecuador, al estar enclavada en dos áreas de gran trascendencia: los páramos del norte, específicamente el páramo del Antisana, y los bosques montanos ubicados en las estribaciones orientales de la Cordillera de los Andes, que comprenden las estribaciones orientales de la cordillera Real y la subcordillera de Guacamayos. Además, destaca la proximidad entre la Parroquia y el Antisana, con una distancia de 32,2 km.

La cuenca del río Cosanga asume un papel hidrográfico crucial, extendiéndose a lo largo de 53,1 km y abarcando un área de 495 km². Esta extensión y su relevancia en el contexto hídrico la posicionan como una cuenca de notable importancia. Adicionalmente, dentro de la parroquia Cosanga, se encuentran afluentes fundamentales que contribuyen a la formación del río Cosanga, tales como río Aliso, Arenillas, Chontas Grande, Yanayacu Grande, y río Bermejo. *(PDOT-Cosanga, 2019)*

En la Figura 3, se presenta un mapa representativo de la ruta del río Cosanga, para lograr visualizar el mapa fue necesario utilizar el software QGIS de manera que se logre visualizar tanto la ruta del río como también los puntos donde existen hogares cercanos al río Cosanga.

Figura 3.

Mapa de demarcación hidrográfica parroquia Cosanga.



Fuente: Tomado de software QGIS

2.1.5. Análisis de riesgos de desastres en Ecuador

Dentro del plan nacional, se centra en desarrollar mecanismos y estrategias que permitan generar una mayor comprensión del riesgo de desastres al que se encuentra expuesto el territorio nacional, así como la vigilancia, monitoreo y alerta temprana ante la inminente ocurrencia de un evento peligroso; dando cumplimiento a la Prioridad 1 del Marco de Sendai: “Comprender el Riesgo de Desastres”, donde la Academia, los organismos técnico científico y el ente rector de la gestión de riesgos deberán considerar los siguientes procesos:

Generación de información para incrementar el conocimiento del riesgo de desastres: Proceso orientado a la aplicación de metodologías y herramientas consolidadas entre la Academia, los organismos técnico científico y el ente rector de la gestión de riesgos, con la finalidad de consolidar la información referente al riesgo de desastres en un sistema nacional de gestión registro de información geográfica de amenazas, vulnerabilidades y riesgos.

Vigilancia y monitoreo de eventos peligrosos: Proceso mediante el cual se busca establecer una red interconectada de vigilancia y monitoreo de eventos peligrosos previsibles,

en la que contribuyan los diferentes organismos técnico-científicos, la academia, el organismo rector de la gestión de riesgos y los GAD en sus diferentes niveles de gobierno, dentro del marco de sus competencias y capacidades territoriales.

Alerta temprana: Tiene por finalidad emitir los avisos respectivos ante la eminente ocurrencia de un evento peligroso, a través de la implementación de un Sistema Nacional de Alerta Temprana Multiamenaza que articule los diferentes actores del SNGRE en su ámbito nacional y local, a través del cumplimiento de estándares para la implementación y funcionamiento de los sistemas de alerta temprana a nivel institucional y comunitario, emitidos por el ente rector de la gestión de riesgos. (SNGR, 2016)

2.1.5.1. Amenazas o peligros

La parroquia Cosanga, debido a su ubicación geográfica y terreno extremadamente accidentado, está susceptible a la ocurrencia de fenómenos naturales como inundaciones, deslizamientos, sismos, y otros eventos de este tipo. Además, se identifican causas de daño ambiental relacionadas con actividades humanas, siendo notables la extracción de maderas, la intervención y explotación de los bosques con fines agropecuarios, principalmente, así como otras formas de alteración del entorno natural. En el análisis del riesgo en la parroquia Cosanga, se ha dado especial atención al estudio del riesgo volcánico, representado en el mapa correspondiente que muestra la vulnerabilidad volcánica presente en la zona. (PDOT-Cosanga, 2019)

Los deslizamientos pueden ser desencadenados por diferentes eventos, como terremotos, erupciones volcánicas, suelos empapados debido a lluvias intensas, el aumento del nivel de aguas subterráneas y la erosión provocada por los ríos. Un sismo en suelos saturados, originado por un terremoto, crea condiciones extremadamente riesgosas. A pesar de que los

deslizamientos se concentran en áreas relativamente pequeñas, su peligrosidad radica en su frecuencia recurrente.

2.2. Sistema de monitoreo

Al utilizar un sistema de monitoreo, hace referencia al conjunto de sistemas y técnicas que posibilitan la vigilancia en tiempo real, la identificación de cambios climáticos, y la completa gestión de los accesos, tanto para personal encargado del monitoreo, en edificios, talleres, plantas industriales o cualquier propiedad que lo requiera.

2.2.1. Internet de las cosas

IoT se entiende como una red vinculada a objetos físicos y virtuales que dependen del acceso a Internet. Estos dispositivos emplean software embebido, donde, no solo tienen acceso a internet, si no también prestan servicios que están basados en acciones dictadas de forma remota, las cuales se pueden originar en eventos particulares e incluso del aprendizaje de datos recibidos. “IoT consiste en desarrollos de software y hardware que cuentan con todas las herramientas necesarias para cumplir tareas muy específicas. Cada uno de los objetos conectados a Internet tiene un número de IP y mediante este puede ser accedido para recibir instrucciones. Asimismo, puede contactarse con un servidor externo y enviar los datos que recoja”. (J. Salazar & Silvestre, 2016)

2.2.2. Componentes de Hardware

Dentro de los equipos principales de hardware para lograr el funcionamiento es necesario: microcontroladores, módulo de comunicación inalámbrica y sensores.

2.2.2.1. LoRa

LoRa, presente en las redes LoRaWAN, constituye la tecnología de modulación que viabiliza a los dispositivos de IoT la transmisión de información a velocidades bajas, con un alcance extenso y un consumo mínimo de energía. En el contexto de LoRaWAN, la

comunicación entre los dispositivos LoRa se establece mediante la dirección de sus canales y parámetros de conexión, incluyendo canal, ancho de banda y cifrado de datos. La eficacia de estas tecnologías, tanto en entornos cerrados como al aire libre, así como en términos de la capa física inalámbrica y la presencia de múltiples gateways, se evaluó en el distrito central de negocios. Como resultado, se constató que esta tecnología forma enlaces seguros, lo que la hace idónea para su aplicación en teledetección a costos reducidos. (Camberos & Heyssen, 2020)

2.2.2.2. *Sensores*

Los sensores son elementos electrónicos que están diseñados para monitorear la presencia de lluvia y medir niveles del agua dentro de cualquier lugar específico. De igual manera, los sensores proporcionan información esencial para la generación de alertas meteorológicas y lograr el seguimiento de patrones climáticos.

2.2.2.3. *Microcontrolador*

Un microcontrolador es un circuito integrado que actúa como el núcleo principal en aplicaciones embebidas. Se asemeja a una pequeña computadora, ya que integra sistemas para gestionar entradas y salidas, un procesador y memoria para almacenar tanto programas como variables. Su propósito es automatizar tareas y procesar información, desempeñando un papel similar al de una mini PC.(Marmolejo, 2019)

2.2.3. Componentes de Software

A continuación, se da a conocer los lenguajes de programación software que van a intervenir dentro del desarrollo del proyecto de monitoreo con lógica difusa, de igual manera se define la comunicación con el servidor que se va a utilizar.

2.2.3.1. *Arduino*

Arduino representa una plataforma electrónica de código abierto que combina hardware y software de manejo sencillo. Las tarjetas Arduino tienen la capacidad de interpretar diversas entradas, como la luz en un sensor, la presión de un botón o un mensaje proveniente de Twitter, transformándolas en salidas específicas, como activar un motor, encender un LED o realizar publicaciones en línea. El usuario puede instruir a la tarjeta mediante un conjunto de órdenes dirigidas al microcontrolador de la placa, utilizando el lenguaje de programación Arduino, basado en Wiring, y el Software Arduino (IDE), fundamentado en Processing.(Arduino, 2018)

2.2.3.2. *Python*

Python se destaca como un lenguaje de programación poderoso y de fácil comprensión. Ofrece estructuras de datos eficientes a nivel superior y un sistema de programación orientado a objetos simple pero efectivo. La sintaxis de Python, su tipado dinámico y su naturaleza interpretada lo posicionan como un lenguaje óptimo para la creación rápida de aplicaciones y scripting en diversas áreas, siendo compatible con la mayoría de las plataformas.(Python, 2023)

Este lenguaje de uso general se aplica en el desarrollo web, la ciencia de datos, el aprendizaje automático y la computación científica. Python es empleado tanto por pequeñas como por grandes empresas, siendo seleccionado por algunas organizaciones prominentes como Google, Netflix y la NASA.(Vargas, 2023)

2.2.3.3. *Lenguaje de programación JSON*

JSON es un formato de texto diseñado para almacenar y transferir datos de forma comprensible tanto para las personas como para los sistemas. Gracias a esta característica, es sencillo de aprender y de depurar. Aunque originalmente fue creado en el contexto de

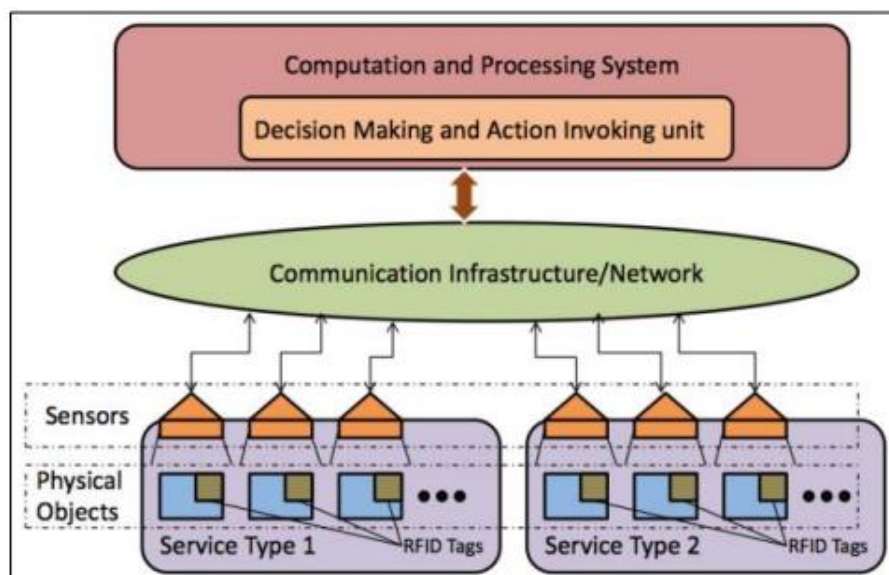
JavaScript, JSON se ha consolidado como un formato versátil que facilita la comunicación de datos entre diferentes lenguajes de programación y plataformas. (Oracle, 2025)

2.2.4. Arquitectura IoT

Para implementar diversas aplicaciones o servicios en un sistema básico basado en IoT, es necesario seguir el procedimiento de trabajo detallado en la Figura 4.

Figura 4.

Sistema de proceso básico IoT



Fuente: Tomado de (J. Salazar & Silvestre, 2016)

Dentro del proyecto, los dispositivos realizan acciones como recolectar datos o enviar señales, proporciona servicios inteligentes, además, va incorporado un sistema para monitorear el estado actual de la zona, facilitando datos sobre el estado actual y ejecutar acciones ante cualquier amenaza climática, esto se logra con su arquitectura que va incorporada.

Tabla 1.

Arquitectura de cuatro capas

Capa de sensorización	Está conformada por sensores, variables físicas y la recopilación de datos.
Capa de transmisión	Se encarga de enviar la información utilizando la tecnología de comunicación empleada en el sistema.
Capa de integración de datos	Su propósito es procesar la información proporcionada por la tecnología de comunicación, filtrar los datos no deseados y consolidar información para los usuarios finales y servicios.
Capa de servicio de aplicación	Muestra el contenido a los usuarios finales a través de interfaces.

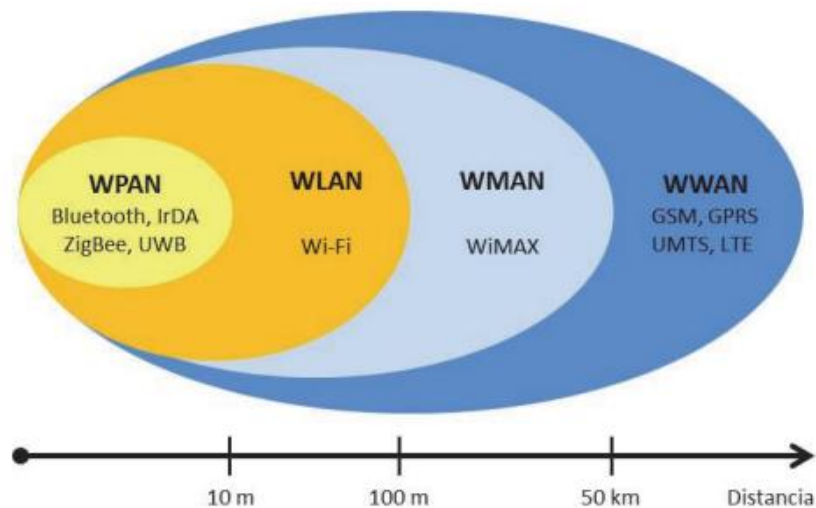
Fuente: Elaborado por el autor.

2.2.5. Tecnologías inalámbricas

Las redes inalámbricas son sistemas que emplean señales de radio para establecer conexiones entre dispositivos, prescindiendo de la necesidad de utilizar cables de cualquier índole. Los dispositivos que con frecuencia emplean redes inalámbricas abarcan laptops, computadoras de escritorio, netbooks, asistentes personales digitales (PDA), teléfonos móviles, tablets y dispositivos de localización. Aunque las redes inalámbricas operan de manera análoga a las redes con cable, es necesario que transformen las señales de información en un formato apropiado para la transmisión a través del medio de aire. En la figura 5 se muestra la clasificación de las tecnologías inalámbricas.(J. Salazar, 2019)

Figura 5.

Clasificación de redes inalámbricas.



Fuente (J. Salazar, 2019)

2.2.5.1. Clasificación de tecnologías inalámbricas

Las tecnologías inalámbricas van de la mano de distintos protocolos, certificaciones, zona de cobertura, frecuencia, rango de alcance, entre otros aspectos; las características de las cuatro tecnologías inalámbricas se indican a continuación:

Tabla 2

Clasificación de redes inalámbricas

Tipo de Red	WPAN	WLAN	WMAN	WWAN
Estándar	IEEE 802.15	IEEE 802.11	IEEE 802.16	GSM/GPRS/UMTS
Certificación	Bluetooth, ZigBee	Wi-Fi	WiMAX	
Velocidad	55 Mbps	600 Mbps	128 Mbps	42 Mbps
Frecuencia	2.4 GHz	2.4 a 5 GHz	2 a 66 GHz	0.8/1.8/2.1 GHz
Rango	1-100m	30 a 150m	1 a 50Km	35Km
Técnica de radio	FHSS	FHSS, DSSS, OFDM	Varias	Varias

Itinerancia (Roaming)	NO	NO	Si (802.16e)	SI
--------------------------	----	----	--------------	----

Fuente: Adaptado de (Díaz, 2022)

2.2.5.2. *LPWAN*

Las necesidades dentro de una red de sensores inalámbricos es ideal mantener una comunicación entre componentes físicos de baja potencia, menor costo y mayor rango de cobertura. Las redes LPWAN (Low Power Wide Area Networks) constituyen tecnologías de comunicación inalámbrica que facilitan la transmisión de datos entre un dispositivo y una estación base o gateway, con distancias que pueden alcanzar cientos de metros o kilómetros, todo ello con un consumo energético considerablemente bajo.

Estas tecnologías, debido a sus características particulares, están posibilitando la implementación de las principales iniciativas actuales en el Internet de las cosas (IoT). Al ser diseñadas específicamente para este entorno, permiten la instalación de decenas o incluso centenares de nodos distribuidos en una amplia área. Estos nodos pueden ser alimentados con baterías de larga duración, sin necesidad de grandes infraestructuras o costosas instalaciones de cableado. (Becolve, 2020)

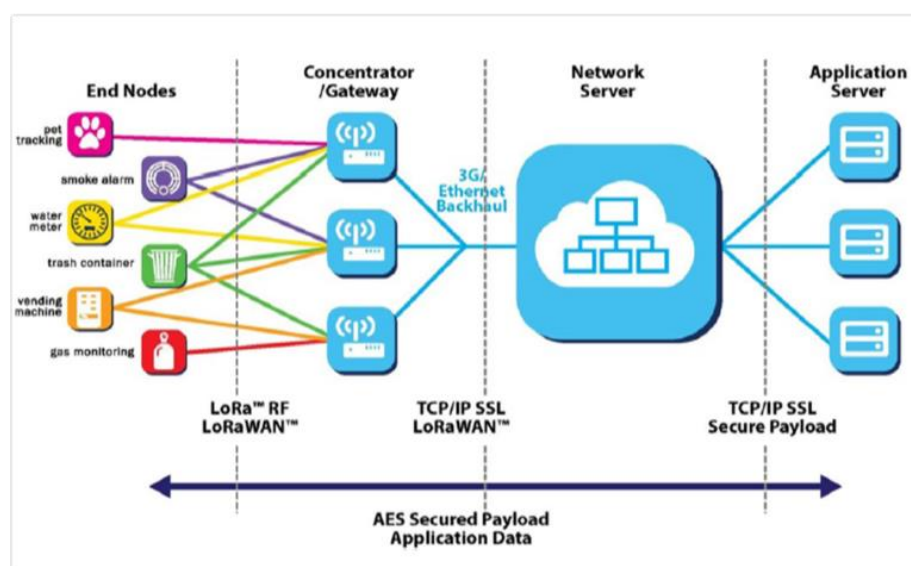
LoRaWAN. La Alianza LoRa define LoRaWAN como un protocolo de comunicación de baja potencia diseñado para conectar dispositivos alimentados por batería de manera inalámbrica a Internet, abarcando redes regionales, nacionales o globales. Este protocolo responde a las necesidades clave de la Internet de las cosas (IoT), como la comunicación bidireccional, la seguridad integral, la movilidad y los servicios de localización.(LoRa, 2020)

LoRaWAN opera sobre la tecnología LoRa, una forma de comunicación de radiofrecuencia inalámbrica que utiliza un espectro sin licencia. La especificación de LoRaWAN está disponible para que cualquier persona configure y opere una red LoRa. LoRa

emplea la modulación de espectro amplio y se caracteriza por su capacidad para lograr comunicaciones de largo alcance con un ancho de banda reducido. Al utilizar una forma de onda de banda estrecha y una frecuencia central para la transmisión de datos, se vuelve resistente a las interferencias. La Figura 6 muestra como está compuesto por una serie de elementos la arquitectura LoRaWAN.

Figura 6.

Arquitectura básica de la tecnología LoRaWAN.



Fuente. (Marino, 2023)

LoRaWAN para Ecuador. Dentro de la información de LoRa Alliance se identifica las frecuencias de canales AU915-928 utilizadas en la comunicación LoRa para Ecuador. Los dispositivos deben alternar entre canales de subida y bajada para garantizar eficiencia y evitar interferencias. (LoRa, 2020)

LoRa es una tecnología inalámbrica patentada que utiliza un método específico de modulación en radiofrecuencia. En América, opera principalmente en las bandas de 915 a 928 MHz y también en 2.4 GHz. (ARCOTEL, 2022)

2.3. Sistemas Inteligentes

Los sistemas inteligentes se fundamentan en la integración de diversos procesos industriales para crear un sistema global de operación y supervisión. Estos sistemas están presentes en diversas áreas industriales, siendo particularmente reconocidos en el control de centros de distribución de grandes empresas minoristas, compañías logísticas o infraestructuras como puertos, donde gestionan actividades como el control de inventario, recepción, almacenamiento y distribución de mercancías. Los sistemas inteligentes desempeñan un papel crucial en la cuarta revolución industrial al posibilitar la integración de tecnologías avanzadas para coordinar los sistemas de control de producción y cadena de suministro, permitiendo así la automatización completa de los procesos comerciales. (UNIR, 2023)

2.3.1. Aprendizaje Automático

El proceso de aprendizaje abarca diversas situaciones en las cuales el aprendiz amplía sus conocimientos o habilidades para llevar a cabo una tarea específica. En estas instancias, el aprendizaje implica realizar inferencias sobre la información disponible con el objetivo de construir una representación adecuada de algún aspecto relevante de la realidad o de un proceso determinado. Una metáfora común en el ámbito del aprendizaje automático, dentro de la Inteligencia Artificial, consiste en considerar la resolución de problemas como un tipo de aprendizaje. Este enfoque implica que, una vez que se resuelve un tipo de problema, el sistema puede reconocer situaciones problemáticas similares y reaccionar utilizando la estrategia aprendida. (Moreno et al., 1994)

2.3.2. Inteligencia Artificial

La inteligencia artificial (IA) implica la capacidad de las máquinas para aplicar algoritmos, aprender de los datos y emplear esos conocimientos en la toma de decisiones, imitando así el proceso humano. A diferencia de las personas, los dispositivos basados en IA no requieren descanso y tienen la capacidad de analizar grandes cantidades de información

simultáneamente. Además, la proporción de errores en las máquinas al realizar tareas comparables a las humanas es considerablemente menor. (Rouhiainen, 2018)

2.3.3. Norma IEEE 29148

La Norma ISO 29148 representa un protocolo internacional que establece los fundamentos y directrices para la definición de los requisitos de software. Este estándar delinea los procedimientos para la documentación precisa de los requisitos, delineando la información necesaria y los métodos de gestión a lo largo del ciclo de vida del software. (IEEE, 2018)

2.3.4. Lógica difusa

La lógica difusa ofrece una herramienta para conservar la noción de vaguedad en lugar de eliminarla mediante la imposición arbitraria de afirmaciones verdaderas o falsas, como lo hace la lógica bivalente. Gran parte del razonamiento humano no se basa en extremos de blanco y negro, ceros y unos, sino en tonalidades de grises y valores intermedios. Esto se refleja en el hecho de que la experiencia humana, que los sistemas expertos buscan transferir a las máquinas, a menudo depende del entorno, es incompleta y tiene poca confiabilidad. En consecuencia, muchas decisiones en el mundo real se toman en un entorno donde los objetivos, restricciones y consecuencias de las acciones posibles no se conocen con precisión. Para abordar la imprecisión de manera cuantitativa, a menudo se recurre a conceptos y técnicas de la teoría de probabilidades, así como a las herramientas proporcionadas por la teoría de decisiones, la teoría de control y la teoría de la información. Sin embargo, al hacer esto, se asume incorrectamente que la imprecisión, sea cual sea su naturaleza, puede reducirse a procesos aleatorios desde la perspectiva de la lógica difusa. (Perez & Leon, 2007)

2.3.4.1. *Ventajas de la lógica difusa*

Entre las principales ventajas que brinda la lógica difusa son las siguientes:

- Su adaptabilidad, permitiendo ajustes según las necesidades.

- Su sencillez en la construcción.
- Ofrece soluciones directas para problemas complejos.
- Maneja múltiples entradas de información simultáneamente y toma decisiones precisas.
- Necesita menos datos para su codificación.
- Trabaja eficazmente con información incompleta o incierta.
- Flexibilidad para representar la ambigüedad y relaciones entre datos.
- Capacidad para modelar conocimiento experto en un sistema formal.
- Habilidad para tomar decisiones basadas en diversos criterios.
- En caso de fallo, es fácil de corregir o reprogramar.

2.3.4.2. *Sistemas de control difuso*

Los sistemas expertos difusos o Fuzzy Expert System (FES) son una categoría de sistemas de inteligencia artificial que combinan la teoría de conjuntos difusos con la teoría de sistemas expertos para desarrollar un sistema de razonamiento capaz de manejar la incertidumbre y la vaguedad de los conceptos. En los FES, los conjuntos difusos se emplean para representar los conceptos y variables del sistema, mientras que los sistemas expertos se utilizan para procesar la información y realizar inferencias. Esta combinación permite modelar sistemas complejos y no lineales, siendo especialmente útil en aplicaciones donde la incertidumbre y la variabilidad son factores críticos.

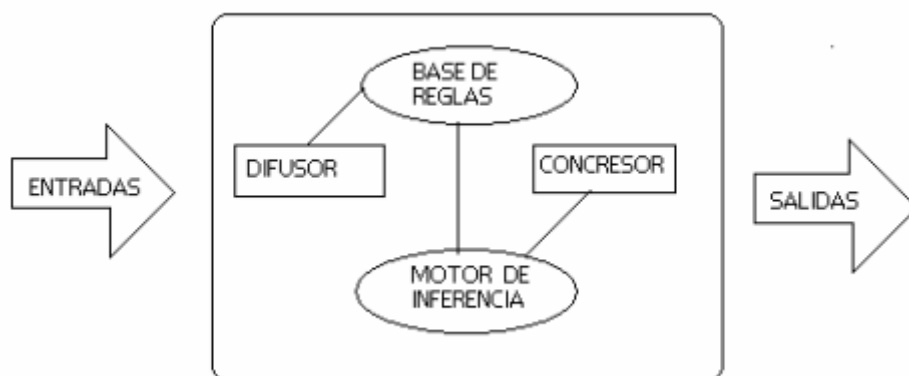
La arquitectura de los FES consta de tres componentes principales: la base de conocimiento, el motor de inferencia y la interfaz de usuario. La base de conocimiento es una base de datos que contiene información sobre el dominio del problema y las reglas de inferencia. El motor de inferencia procesa la información de la base de conocimiento y realiza inferencias utilizando la lógica difusa y las reglas de inferencia. La interfaz de usuario

proporciona una forma para que los usuarios interactúen con el sistema y realicen consultas. El proceso de aprendizaje en los FES se lleva a cabo mediante el uso de algoritmos de optimización, como el algoritmo de retropropagación difusa y el algoritmo de evolución diferencial, los cuales ajustan los parámetros del modelo para minimizar una función de costo que mide el error entre la salida del sistema y la salida deseada. Dentro de la Figura 7, se muestra el proceso que debe cumplir el control difuso.

Los sistemas de control difuso están compuestos de cuatro módulos:

Figura 7.

Sistema de control difuso.



Fuente: (Kassir, 2023)

Difusor (fuzificación): Se trata de recibir valores numéricos y asignarles predicados imprecisos mediante conjuntos borrosos.

Base de reglas: Se trata de una base de datos originada a partir de experiencia, sentido común o intuición, empleada para dirigir el comportamiento del sistema mediante estructuras condicionales del tipo SI – ENTONCES.

Motor de inferencia: Analiza qué regla o reglas del sistema se pondrán en marcha frente a un valor específico de entrada.

Concesor (defuzificación): Implica obtener un valor numérico para cada una de las salidas del sistema basándose en los conjuntos borrosos a los que están asociadas.

2.3.4.3. *Conjuntos difusos y conjuntos clásicos*

Los conjuntos difusos y los conjuntos clásicos son dos herramientas matemáticas distintas empleadas para la representación de conjuntos de objetos o fenómenos. A pesar de compartir la noción fundamental de conjunto, presentan notables diferencias entre sí. Ambos tipos de conjuntos presentan capacidades distintas para abordar la subjetividad y la incertidumbre. Los conjuntos difusos ofrecen la posibilidad de representar la imprecisión y la ambigüedad en procesos de toma de decisiones y clasificación de objetos o fenómenos. En contraste, los conjuntos clásicos carecen de la capacidad para representar la subjetividad y la incertidumbre, siendo más apropiados para situaciones donde la clasificación es precisa y clara. Esta diferencia es crucial, ya que los conjuntos difusos demuestran ser más flexibles y adaptables a diversas situaciones y contextos, a diferencia de los conjuntos clásicos que son más rígidos y menos adaptables. En los conjuntos difusos, la función de membresía, que define el grado de pertenencia, puede ajustarse y adaptarse según las necesidades del problema, permitiendo una mayor precisión y flexibilidad en la toma de decisiones. En cambio, en los conjuntos clásicos, la definición del conjunto es inmutable, limitando así la precisión y flexibilidad de la clasificación. (Alaminos, 2023)

2.3.4.4. *Característica de Conjuntos difusos*

Los conjuntos difusos tienen ciertas características importantes que los hacen útiles en la representación de la incertidumbre. Por ejemplo, la propiedad de continuidad indica que

pequeñas variaciones en la función de membresía no alteran significativamente la forma del conjunto, lo que facilita su adaptación a diferentes tipos de datos. Otra propiedad relevante es la inclusión, que establece que, si un conjunto difuso **A** está contenido en un conjunto difuso **B**, entonces todos los elementos que pertenecen a **A** también pertenecen a **B**. Esta propiedad es útil para realizar operaciones como la unión, la intersección y el complemento con conjuntos difusos. De esta manera, las operaciones básicas que se pueden realizar con conjuntos difusos son la unión, la intersección y el complemento, las cuales se definen a través de la función de membresía del conjunto difuso.

2.3.4.5. Funciones de pertenencia

Las funciones de pertenencia, también llamadas de membresía, indican el grado en que un elemento pertenece a un conjunto difuso, con valores comprendidos entre 0 y 1. La elección de una función de pertenencia depende del contexto en el que se aplique. Generalmente, se prefieren funciones simples, ya que facilitan los cálculos, optimizan la computación y mantienen la precisión al definir el conjunto difuso correspondiente. (ALBA, 2019)

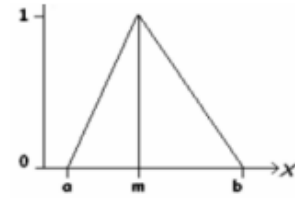
Existen diversas funciones de pertenencia, entre las más utilizadas se encuentran las de tipo triangular, trapezoidal, gaussiana y sigmoideal.

Función Triangular. Esta función se caracteriza por estar delimitada por un límite inferior (a) y un límite superior (b), y se evalúa con un valor (m) que cumple la condición $a < m < b$.

Figura 8.

Función de pertenencia triangular.

$$A(x) \begin{cases} 0 & \text{si } x \leq a \\ (x-a)/(m-a) & \text{si } x \in (a, m) \\ (b-x)/(b-m) & \text{si } x \in (m, b) \\ 1 & \text{si } x \geq b \end{cases}$$



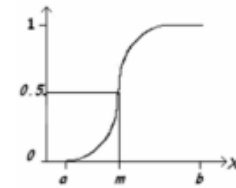
Tomado de: (Zimmermann, 2001)

Función Sigmoidal. Esta función se define mediante un límite inferior (a) y un límite superior (b), y se evalúa en su punto de inflexión, cumpliendo la condición $a < m < b$. El crecimiento de la función es gradual cuando la distancia entre a y b es grande.

Figura 9.

Función de pertenencia Sigmoidal.

$$A(x) \begin{cases} 0 & \text{si } x \leq a \\ 2\{(x-a)/(b-a)\}^2 & \text{si } x \in (a, m) \\ 1 - 2\{(x-a)/(b-a)\}^2 & \text{si } x \in (m, b) \\ 1 & \text{si } x \geq b \end{cases}$$



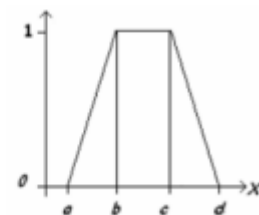
Tomado de: (Zimmermann, 2001)

Función Trapezoidal. Esta función se caracteriza por tener límites inferiores (a) y (d), así como límites superiores (b) y (c), cumpliendo la condición $a < b < c < d$.

Figura 10.

Función de pertenencia Trapezoidal.

$$A(x) \begin{cases} 0 & \text{si } (x \leq a) \text{ o } (x \geq d) \\ (x-a)/(b-a) & \text{si } x \in (a, m) \\ 1 & \text{si } x \in (m, b) \\ (d-x)/(d-c) & \text{si } x \in (b, d) \end{cases}$$

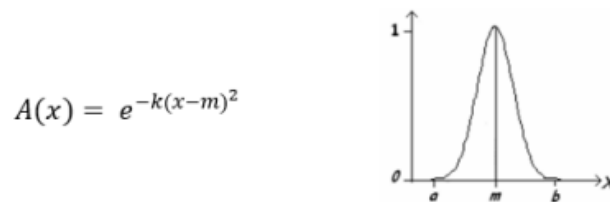


Tomado de: (Zimmermann, 2001)

Función Gaussiana. Esta función se define mediante un valor medio (m) y una desviación estándar $k > 0$, cumpliéndose cuando k tiene un valor reducido.

Figura 11.

Función de pertenencia Gaussiana.



Tomado de: (Zimmermann, 2001)

2.3.4.6. Inferencia difusa

La inferencia difusa constituye el procedimiento de razonamiento empleado en la lógica difusa para derivar conclusiones a partir de conjuntos y reglas difusas. Este proceso se fundamenta en la fusión de información difusa procedente de diversas fuentes, con el propósito de obtener respuestas más precisas y adaptadas a la incertidumbre y ambigüedad inherentes a los conceptos en el contexto del mundo real. En la inferencia difusa, se aplican diversas operaciones lógicas para combinar la información difusa de conjuntos y reglas difusas. Estas operaciones engloban la unión difusa, la intersección difusa, la negación difusa, la implicación difusa, la composición difusa, la competencia difusa, la agregación difusa, la defuzzificación, entre otras.(Alaminos, 2023)

La inferencia difusa se desglosa en dos fases: la inferencia hacia delante y la inferencia hacia atrás. La inferencia hacia delante implica el empleo de las reglas y conjuntos difusos de entrada para generar un conjunto difuso de salida. Por otro lado, la inferencia hacia atrás es el proceso de utilizar el conjunto difuso de salida y las reglas difusas para identificar los conjuntos difusos de entrada que contribuyeron a su formación.

3. Capítulo III: Diseño del sistema

Este capítulo aborda los aspectos relacionados con la metodología utilizada para la implementación del sistema de monitoreo con lógica difusa para prever el desbordamiento del río Cosanga. Incluye la integración de la lógica difusa en el control, la organización de las zonas en el área de estudio, y la evaluación de los requisitos esenciales que permiten identificar los componentes de hardware y software necesarios para construir un prototipo funcional.

3.1. Metodología

El presente proyecto está orientado al desarrollo de un sistema de monitoreo para alertar sobre posibles desbordamientos, utilizando lógica difusa para identificar situaciones de riesgo y notificar a los habitantes de la Parroquia de Cosanga. Se emplea el "Modelo en V" para guiar el diseño e implementación del sistema, abarcando la selección de hardware, software y los requisitos del sistema y usuarios. Además, se toma en cuenta la norma IEEE 29148 para garantizar la correcta gestión y definición de los requisitos durante todas las fases del desarrollo. A continuación, se indican los siguientes procesos del modelo.

3.1.1. Modelo en V

El modelo en V es popular debido a que ofrece claridad y transparencia en cada etapa del desarrollo, gracias a la definición detallada de los procesos del sistema. El modelo divide el desarrollo de software en tres fases principales: diseño, implementación y pruebas de integración. La estructura en forma de "V" simboliza visualmente el flujo del desarrollo, representando la conexión entre las etapas de diseño y pruebas, asegurando que cada etapa de diseño tenga su correspondiente fase de validación, garantizando un proceso estructurado y comprensible. Este modelo se basa en la ejecución de una serie de pasos organizados de manera secuencial, los cuales se detallarán en la siguiente sección.(Oppermann, 2024)

Estudio de los requisitos: Se lleva a cabo un estudio para definir la funcionalidad del sistema, estableciendo los requisitos funcionales y no funcionales. Los requisitos funcionales especifican las funciones que el sistema ejecutará sobre las entradas para generar salidas, es decir, el "qué" del sistema. Por otro lado, los requisitos no funcionales describen las características esperadas por los interesados en cuanto al comportamiento del sistema.

Diseño del sistema: Implica definir la tecnología que será utilizada dentro del sistema de monitoreo. Se diseña detalladamente, especificando las funciones del sistema, los elementos de la interfaz de usuario y las estructuras de datos. Además, se preparan los elementos necesarios para realizar las pruebas del sistema, garantizando que todo esté alineado con los requisitos y el diseño previamente definidos.

Diseño de la arquitectura: Se define el funcionamiento de los componentes del sistema, así como sus interfaces y las dependencias entre ellos. El diseño arquitectónico facilita la organización de las interacciones entre las distintas soluciones técnicas, permitiendo una integración eficiente entre cada parte del sistema. Esto garantiza que los componentes trabajen de manera coordinada para cumplir con los requisitos funcionales y no funcionales previamente establecidos, asegurando que cada solución esté claramente alineada con el propósito general del proyecto.

Diseño modular: Se detalla cada módulo del sistema, especificando su lógica interna y la interfaz que describe la API. El diseño detallado facilita el posterior desarrollo del software, permitiendo programar cada componente de manera efectiva. Además, se realizan las pruebas para cada módulo, ya que se cuenta con la especificación de la interfaz y una descripción funcional clara de los componentes, lo que asegura que las interacciones entre ellos se verifiquen adecuadamente durante el proceso de desarrollo.

Programación: La fase de programación consiste en la escritura del código fuente utilizando un lenguaje de programación determinado. En este proceso, los desarrolladores siguen las especificaciones definidas en las fases anteriores de diseño y planificación. Durante esta etapa, se traduce el diseño técnico en un software funcional, asegurándose de que el código cumpla con los requisitos establecidos previamente y se prepare para su integración y pruebas posteriores.

3.2. Análisis del proyecto

Este proyecto se fundamenta en la revisión de información científica y territorial relacionada con la prevención de desbordamientos. Para ello, se establece un vínculo dentro de la junta parroquial de Cosanga, con el propósito de recopilar datos clave para el desarrollo de la investigación. En el **ANEXO 1** se observa el documento otorgado por el GADPR Cosanga.

3.2.1. Situación Actual

Para llevar a cabo este trabajo de manera efectiva, es fundamental comprender la situación actual de la zona donde se implementará el proyecto. Esto permitirá definir con precisión los requisitos necesarios para el desarrollo e implementación del sistema de monitoreo basado en lógica difusa, que se instalará en la parroquia de Cosanga.

La ubicación geográfica y el relieve montañoso de la parroquia de Cosanga la hacen especialmente vulnerable a diversos fenómenos naturales, como inundaciones, deslizamientos de tierra y actividad sísmica, entre otros.

En cuanto a su economía, las principales actividades productivas se dividen en el sector agropecuario o primario, seguido por el sector de servicios o terciario. Destacan especialmente la ganadería, la agricultura y el turismo como las actividades económicas más representativas de la parroquia. (PDOT-Cosanga, 2019)

Figura 12.

Río Cosanga cerca de la parroquia Cosanga.



Fuente: Tomado de (*PDOT-Cosanga*, 2019)

Actualmente no existe un sistema en esta zona que se encargue de monitorear las diversas condiciones del río en sus proximidades. Esta carencia subraya la necesidad urgente de implementar una red de sensores y módulos inalámbricos, diseñada para proporcionar información inmediata sobre el comportamiento del río. El sistema propuesto permitirá monitorear el nivel del caudal, detectar cambios en la velocidad del río, y registrar las variaciones en los niveles de precipitación, tanto en su aumento como en su disminución. Este enfoque integral no solo facilitará una mejor comprensión de las dinámicas del río, sino que también proporcionará datos críticos para anticipar posibles riesgos de desbordamiento, contribuyendo significativamente a la seguridad y protección de la comunidad local.

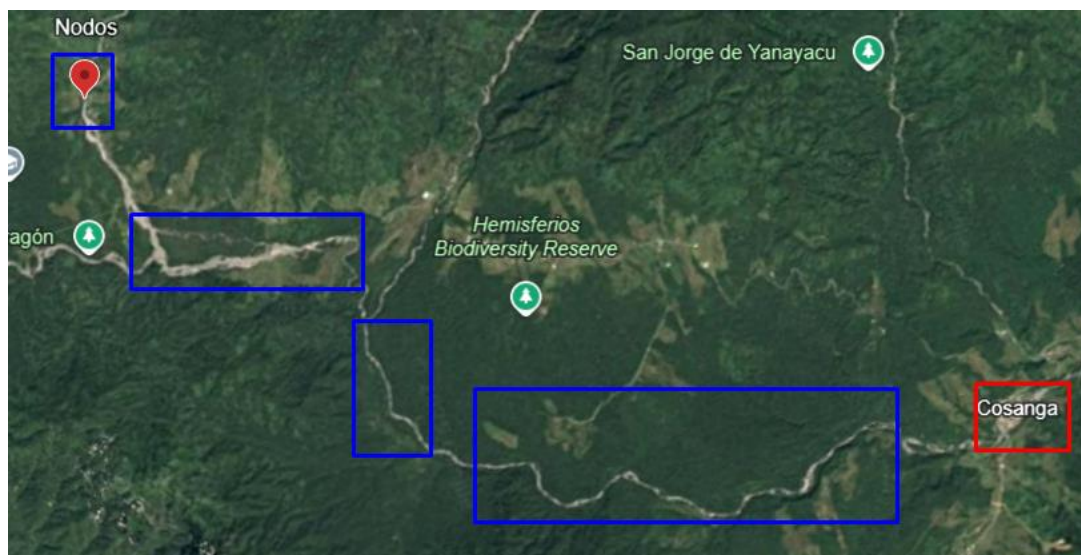
3.2.2. Ubicación

El presente proyecto se desarrollará en el río Cosanga, ubicado en la parroquia de Cosanga, perteneciente al cantón Quijos, en la provincia de Napo. Este río es reconocido principalmente por ser un destino popular para la práctica de deportes extremos, como el kayak y el rafting, debido a sus aguas rápidas y desafiantes. Además de su importancia para el turismo de aventura, el río Cosanga juega un papel crucial en la agricultura local, proporcionando agua esencial para el riego de cultivos en las comunidades circundantes.

Dentro de la Figura 13 obtenida de Google Earth, se puede observar la confluencia de los ríos Aragón y Aliso, que da origen al río Cosanga. Esta formación fluvial se destaca en el recuadro de color azul, permitiendo visualizar claramente el curso inicial del río. Además, en el recuadro de color rojo, se muestra la ubicación de la Parroquia de Cosanga, destacando su proximidad al río, el cual pasa a una distancia aproximada de 11 metros de la comunidad.

Figura 13.

Ubicación geográfica del río Cosanga en Google Earth.



Fuente. Google Earth

3.2.3. Características del sistema

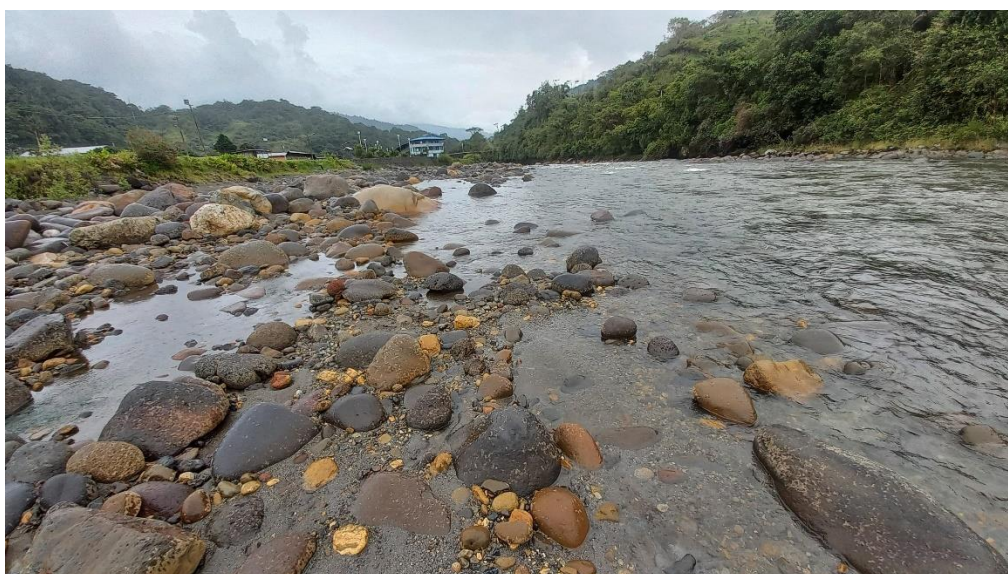
El sistema de monitoreo a emplear en esta investigación está destinado para ayudar a prevenir desbordamientos dentro de la parroquia de Cosanga, esto con el objetivo de obtener información en tiempo real como el nivel del río, la velocidad y la precipitación. Para lograr todo el proceso del sistema, se implementa una metodología basada en el uso de sensores, comunicación LoRa y procesamiento de datos mediante lógica difusa.

Es fundamental garantizar una transmisión de datos confiable entre el nodo sensor y el nodo coordinador, ya que posibles interferencias o pérdidas de señal podrían comprometer el funcionamiento del sistema. Para mitigar cualquier riesgo, se realizan pruebas de cobertura y estabilidad de la red LoRa dentro de la zona donde se va a colocar los nodos.

Dentro de la Figura 14, se da a conocer el sitio en donde se va a planear desarrollar el proyecto, se logra visualizar el cauce del río, este río tiende a variar su nivel y velocidad, los cambios que ocurren dentro del río van acorde a las temporadas de lluvia.

Figura 14.

Cauce del río Cosanga



Fuente: Autoría.

En la Figura 15, se presenta la zona seleccionada para la instalación del nodo sensor. Esta ubicación ha sido seleccionada estratégicamente debido a las características específicas del río en este punto, ya que se encuentra a aproximadamente 17 km de la Parroquia de Cosanga. Esta distancia asegura un tiempo óptimo para que las notificaciones generadas lleguen oportunamente a la parroquia, permitiendo una rápida toma de decisiones y una eficiente implementación de medidas preventivas frente a riesgos asociados al río.

Figura 15.

Zona seleccionado para recolectar datos con los sensores.



Fuente: Autoría.

3.2.4. Método de investigación para la obtención de requisitos.

Se emplea una técnica de investigación para recopilar los requisitos del sistema, con el propósito de realizar una indagación precisa y objetiva. Esta técnica permite identificar de manera clara las necesidades y expectativas de los usuarios y partes interesadas, garantizando que el diseño y desarrollo del sistema se alinee con los requisitos establecidos, promoviendo así la efectividad del proyecto y asegurando que las decisiones tomadas se basen en información fundamentada.

3.2.4.1. Cuestionario aplicado a la parroquia

Realizar un cuestionario dentro del marco de la tesis tiene como objetivo recopilar información valiosa y específica directamente de la Parroquia de Cosanga. Esto permite identificar las necesidades, opiniones y percepciones relacionadas con el sistema de monitoreo propuesto, asegurando que las soluciones planteadas sean relevantes y adaptadas al contexto local. Además, las preguntas sirven como una herramienta para validar hipótesis, orientar el diseño del sistema y garantizar que el proyecto tenga un impacto práctico y significativo en la Parroquia.

A continuación, se indica las preguntas que fueron planteadas al señor presidente de la Junta Parroquial de Cosanga, el formato se detalla en el **ANEXO 2**:

1. ¿Qué tan importante considera usted el monitoreo del río en la parroquia de Cosanga?
2. ¿En su experiencia, cuál es el mayor impacto de las lluvias intensas en la parroquia?
3. ¿Qué medidas considera que serían más efectivas para prevenir daños causados por el río?
4. ¿Le parece útil implementar un sistema de monitoreo que alerte sobre el riesgo de desbordamiento?
5. ¿Qué medio prefiere para recibir alertas sobre posibles desbordamientos?
6. ¿Estaría dispuesto a colaborar en actividades de mantenimiento del sistema de monitoreo?
7. ¿Cómo describiría los cambios en el nivel o comportamiento del río durante las temporadas de lluvia?

8. ¿Qué tan informada considera que está la comunidad sobre los riesgos de desbordamiento del río?
9. ¿Qué tan confiable considera Ud que un sistema de monitoreo puede trabajar sin la presencia de un operador?
10. ¿Con cuánta anticipación te gustaría recibir una alerta sobre un posible desbordamiento del río?
11. ¿Cuánto tiempo estimas que tomaría evacuar tu hogar en caso de emergencia?
12. ¿Consideras necesario un plan de evacuación comunitario en caso de inundaciones?
13. ¿Qué otras ideas o sugerencias tienen Ud para mejorar la seguridad y prevenir desbordamientos en la parroquia?

3.3. Descripción general del sistema

El sistema de monitoreo que se implementará en la Parroquia de Cosanga está diseñado para prever el desbordamiento del río Cosanga mediante el uso de tecnologías avanzadas de sensores y lógica difusa. Este sistema tiene como objetivo principal proporcionar alertas tempranas que permitan a las autoridades locales y a la comunidad tomar medidas preventivas, minimizando así el riesgo de pérdidas humanas y materiales debido a inundaciones.

El sistema consta de una red de sensores distribuidos estratégicamente. Estos sensores son responsables de medir variables críticas como la cantidad de lluvia, el nivel del agua en el río y la velocidad del flujo. Estos datos son fundamentales para evaluar las condiciones que podrían llevar a un desbordamiento. Los sensores utilizados son altamente sensibles y están diseñados para operar en entornos difíciles, como las zonas montañosas y de alta pluviosidad que caracterizan a la región de Cosanga.

Para la transmisión de los datos recolectados, se utilizan módulos WiFi LoRa 32, que permiten una comunicación eficiente y de largo alcance entre el nodo sensor y nodo coordinador. El nodo sensor recibe la información de los sensores y la envía de manera inalámbrica al nodo coordinador, que se encuentra a una distancia aproximada de 62 m. El nodo coordinador a su vez, es responsable de recibir y procesar los datos, evaluando el riesgo de desbordamiento en tiempo real.

Los datos procesados mediante lógica difusa se evalúan tomando decisiones mediante la IA implementada. Esta tecnología permite manejar la incertidumbre inherente a las condiciones climáticas y del río, ofreciendo una estimación adecuada del riesgo de desbordamiento. Las reglas de la lógica difusa consideran múltiples factores simultáneamente, como un alto nivel de agua, alta velocidad del río y fuertes precipitaciones, para determinar si existe un alto riesgo de desbordamiento. Si el sistema detecta condiciones peligrosas, se genera una alerta automática que puede ser comunicada rápidamente a las autoridades locales y a la comunidad.

Este sistema de monitoreo es autónomo y se actualiza continuamente, lo que garantiza que la información proporcionada sea siempre relevante y oportuna. El sistema al iniciar su operación funcionará de manera independiente, proporcionando una capa adicional de seguridad para la población de Cosanga.

3.4. Requerimientos

El diseño del sistema se enfoca en la monitorización exhaustiva de las condiciones físicas del río Cosanga, abarcando cada aspecto relevante para su análisis. El sistema ha sido estructurado en varias etapas clave, cada una de ellas desempeñando un papel fundamental en la captura, procesamiento y transmisión de los datos críticos sobre el río. Es esencial destacar

los componentes involucrados en cada fase del proceso, ya que estos elementos permiten la recopilación precisa de información sobre el nivel del agua, velocidad del flujo y precipitaciones. En base al **ANEXO 3** se realizó un análisis de requerimientos esenciales para el sistema.

A continuación, se observa la Tabla 4 en donde se da a conocer los acrónimos que van a ser utilizados dentro del proyecto para la selección de requerimientos.

Tabla 4

Definición de abreviaturas

Abreviatura	Descripción
StRS	Requerimientos de Stakeholder
SyRS	Requerimientos del Sistema
SRSR	Requerimientos de Arquitectura

Fuente: Adaptado de (IEEE, 2018)

3.4.1. Requerimientos de stakeholders

Basándose en el análisis de la situación actual y considerando el estándar ISO/IEEE 29148, que establece directrices específicas para el desarrollo de procesos y productos en la ingeniería, se destacan aspectos importantes. Entre ellos, la identificación de los Stakeholders, quienes representan a las personas o elementos que tienen un impacto significativo en los resultados y metas del proyecto de grado. Este estándar internacional proporciona un enfoque unificado para los procesos y productos necesarios en la definición de requisitos de ingeniería a lo largo del ciclo de vida de un sistema y software.

Tabla 5

Actores involucrados

Función	Nombre
Tutor	Ing. Fabián Cuzme
Asesor	Ing, Luis Suárez
Desarrollador	Sr. Anthony Panamá
Ingeniero a cargo del GAD parroquial	Ing. Gonzalo Morales

Fuente: Autoría.

Antes de proceder con la configuración del sistema que apoyará la investigación en curso, es fundamental considerar los requisitos específicos de los stakeholders, así como los requisitos funcionales del sistema. A continuación, se presenta una tabla que detalla estos requisitos enfocados en las necesidades de los usuarios.

Tabla 6

Requerimientos de stakeholders

Tabla de requerimientos StRS				
#	Requerimientos	Prioridad		
		Alta	Media	Baja
StRS1	El sistema debe ser completamente sellado y operar en entornos con alta humedad.	X		
StRS2	El sistema debe estar alimentado por una fuente eléctrica estable y confiable	X		
StRS3	Automatización del sistema para la recolección y transmisión de datos.	X		
StRS4	Asegurar la comunicación inalámbrica para la transmisión de datos.	X		
StRS5	El sistema debe operar de manera continua a lo largo del día.	X		

StRS6	El sitio web y el sistema de alerta deben mantenerse en funcionamiento durante el máximo tiempo posible.	X
StRS7	El nodo coordinador debe mantener conexión a Internet todo el tiempo.	X
REQUERIMIENTOS DE USUARIO		
StRS8	El sistema de monitoreo debe alertar al usuario por medio del sitio web, mensajería y una alarma comunitaria sobre las condiciones que presenta el río.	X
StRS9	El sistema debe indicar los datos través de una página web.	X
StRS10	Contar con una base de datos que permita almacenar los datos y garantizar un acceso inmediato.	X

Fuente: Elaborado por el autor

3.4.2. Requerimientos iniciales del sistema

Dentro de la siguiente tabla se da a conocer los requerimientos más relevantes dentro del sistema.

Tabla 7

Requerimientos iniciales del sistema

FUNCIONALIDADES REQUERIDAS SySR
--

#	REQUERIMIENTOS	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS DE USO					
SySR1	Los sensores utilizados para medir las variables del nivel de agua, precipitación y velocidad tengan un margen de error mínimo.	X			
SySR2	El nodo sensor debe recolectar datos de forma automática y ser enviados al nodo coordinador. Implementar el algoritmo de lógica difusa en el sistema para analizar los datos recolectados.	X			StRS3
SySR3	Los datos recibidos por el nodo coordinador serán almacenados en una base de datos.	X			
SySR4	Facilidad de instalación y mantenimiento	X			
SySR5	Los datos recopilados por el nodo sensor deben visualizarse en el sitio web.		X		StRS9
REQUERIMIENTOS DE PERFORMANCE					
SySR7	Margen de error mínimo en la recolección de datos de sensores.	X			SySR1

SySR8	Capacidad de expandir el sistema para incrementar el número de sensores.	X
SySR9	Gestión equilibrada de recursos energéticos.	X
SySR10	Fiabilidad en condiciones climáticas adversas.	X
SySR11	El sitio web y la aplicación debe generar una alerta por un desbordamiento y el tiempo para que la comunidad evacue.	X
SySR12	Tiempos de análisis de datos y generación de alertas adecuados.	X
SySR13	El sistema debe tener una disponibilidad de acceso el mayor tiempo posible en el día.	X

REQUERIMIENTOS DE INTERFACES

SySR14	Interfaz de usuario adecuada para la visualización de datos.	X
SySR15	Las interfaces utilizadas para el sistema de alerta deben ser las adecuadas (aplicación móvil y sirena).	X
SySR16	Diseño flexible para acceso desde múltiples tipos de dispositivos como smartphones, tablets y PC	X

REQUERIMIENTOS FISICOS

SySR17	El nodo sensor y coordinador deben contar con una estructura sellada y ser capaz de soportar la exposición al agua.	X
SySR18	Colocación estratégica de los sensores para evitar daños y garantizar su funcionamiento continuo.	X
SySR19	Protección de los dispositivos contra daños físicos.	X
SySR20	Fácil integración de panel solar para el respaldo de energía del sistema.	X

Fuente: Elaborado por el autor

3.4.3. Requerimientos de arquitectura

Para garantizar la funcionalidad efectiva del sistema electrónico que se propone, resulta crucial especificar detalladamente tanto el hardware requerido como el software adecuado que permitirá su operación exitosa. La selección de componentes hardware debe alinearse con los requisitos de rendimiento del sistema, incluyendo la capacidad de procesamiento, la durabilidad en condiciones ambientales adversas y la eficiencia energética. Asimismo, el software debe ser capaz de manejar eficientemente la recopilación de datos, el procesamiento en tiempo real y la comunicación segura entre dispositivos.

Tabla 8

Requerimientos de arquitectura

SRSH
REQUERIMIENTOS DE ARQUITECTURA

#	REQUERIMIENTOS	PRIORIDAD			Relación
		Alta	Media	Baja	
REQUERIMIENTOS DE DISEÑO					
SRSH1	Diseño flexible para acceso desde múltiples tipos de dispositivos como smartphones, tablets y PC.	X			SySR13
SRSH2	El nodo sensor debe estar diseñado con un recubrimiento sellado que garantice la protección de todos sus componentes internos.	X			
	Software compatible,	X			
SRSH3	escalable y de código abierto.				
	Hardware asequible, de	X			
SRSH4	bajo consumo de energía y alta durabilidad.				X
	El sitio web tiene una opción en la que solo el	X			
SRSH5	usuario con la contraseña puede ingresar para configuraciones de datos.				

SRSH6	El nodo coordinador debe ubicarse dentro del área de alcance que se encargue de receptar y enviar los datos mediante internet.	X
REQUERIMIENTOS LÓGICOS		
SRSH7	Disponibilidad de mantenimiento y mejoras.	X
SRSH8	El lenguaje de programación deberá ser compatible con todos los sensores y los microcontroladores utilizados.	X
SRSH9	La detección de alertas se basará en reglas difusas aplicadas dentro del sistema.	X
SRSH10	La red inalámbrica debe operar dentro de los parámetros establecidos mediante cálculos y simulaciones realizadas por software.	X
REQUERIMIENTOS DE SOFTWARE		

	Compatibilidad con	X
SRSH11	librerías de sensores con el microcontrolador	
SRSH12	Implementar las reglas de lógica difusa.	
SRSH13	Uso de licencia gratuita y código abierto	X
	Es necesario utilizar el puerto serial para la captura de datos, por lo que	X
SRSH14	el IDE empleado para la programación debe incluir funcionalidades de lectura serial.	
SRSH15	Flexibilidad de conexión de sensores y obtener su lectura	X
REQUERIMIENTOS DE HARDWARE		
	El sistema requiere de dos microcontroladores para	X
SRSH16	recopilar datos, transmitir y crear el acceso para visualizar los datos y alertas.	

SRSH17	Las placas del sistema	X	
	deben incluir puertos de entrada tanto analógicos como digitales para permitir la conexión de sensores.		
SRSH18	Los sensores por utilizar	X	
	deben existir en el mercado y compra accesible.		
SRSH19	Estabilidad de	X	StRS4
	comunicación inalámbrica entre los nodos.		
SRSH20	Garantizar poca latencia en	X	
	la transmisión de datos desde los sensores hacia la estación de monitoreo.		

Fuente: Elaborado por el autor

Es fundamental considerar meticulosamente las especificaciones técnicas y características lógicas de los componentes que conformarán la estructura del sistema propuesto. Cada elemento seleccionado desempeña un papel crítico en la configuración general, impactando directamente en el rendimiento y la eficiencia operativa del sistema. Los detalles técnicos, como la capacidad de procesamiento, la memoria, la resistencia a condiciones ambientales adversas, y la compatibilidad con otros dispositivos, deben ser evaluados y alineados con los objetivos globales del proyecto. Asimismo, las características lógicas, que

incluyen el diseño del software, los algoritmos de procesamiento y las metodologías de integración, son esenciales para asegurar que el sistema funcione de manera cohesiva y eficaz.

3.4.4. Aspectos técnicos del sistema de monitoreo.

- La fuente de energía debe proveer la cantidad de corriente eléctrica a cada elemento dentro de cada nodo.
- Los sensores utilizados deben ser calibrados previamente para minimizar el margen de error y garantizar la precisión de los datos, asegurando un funcionamiento confiable del sistema en tiempo real.
- Se deben utilizar librerías adecuadas que tengan capacidad de compatibilidad entre software y hardware del sistema.

3.5. Elección de hardware y software para el sistema

En esta parte del proyecto se enumeran los recursos disponibles, tales como herramientas, equipos y tecnologías. Luego, se procede a seleccionar los más adecuados según las prioridades establecidas en los requisitos definidos anteriormente. La selección se realiza de manera estratégica para asegurar que cada recurso cumpla con las especificaciones y necesidades del proyecto, priorizando aquellos que permitan un mejor desempeño y cumplimiento de los objetivos del sistema de monitoreo propuesto.

3.5.1. Hardware destinado para el sistema

La arquitectura del sistema se compone de una red inalámbrica y una red de sensores integrados. El nodo sensor incluye un sensor de nivel de agua, un sensor de precipitación y un sensor de velocidad del río, y para lograr la comunicación inalámbrica se utiliza dos módulos WiFi LoRa 32. La selección de estos componentes se realizó considerando los requisitos previamente establecidos, asegurando su adecuación para satisfacer las necesidades del sistema planteado.

3.5.1.1. *Análisis de sensor de nivel del agua en el río.*

Para monitorear el nivel de caudal en un río, se pueden utilizar distintos dispositivos, entre los que resaltan los sensores ultrasónicos. Estos equipos están diseñados para resistir condiciones climáticas extremas y, además, existen modelos especializados para medir niveles de agua en ríos. Los sensores ultrasónicos disponibles en el mercado ofrecen diversas funcionalidades que aseguran un desempeño confiable, incluso en ambientes complejos, lo que los posiciona como una alternativa óptima para ser integrados en el proyecto.

A continuación, en la Tabla 9 se describen las características más importantes de los sensores ultrasónicos para la medición del nivel del río.

Tabla 9

Cuadro comparativo de sensores ultrasónicos.

Parámetros Técnicos	JSN-SR04T	HC-SR04	MB1240	MB7076
Figura				
Tipo de sensor	Ultrasónico	Ultrasónico	Ultrasónico	Ultrasónico
Alimentación	3.3V – 5V	3.3V – 5V	5V	3V – 5V
Corriente	30 mA	15 mA	3.4 mA	3.4 mA
Frecuencia	40 kHz	42 kHz	40 kHz	-
Rango de detección	25cm - 450cm	2cm a 450cm	20 a 765 cm	Hasta 1068 cm
Tamaño del sensor	40 x 28 mm	45 x 20 x 15 mm	22 x 19.9 x 25.11 mm	64 x 43.8 x 43.8 mm
Dispositivos compatibles	Arduino, ESP32, Raspberry Pi	Arduino, ESP32, ESP8266	Arduino, ESP32, Raspberry Pi	Arduino, ESP32
Costo	\$20	\$2.50	\$50	\$132

Fuente: Tomado de (Ramírez, 2023)

Tabla 10

Análisis comparativo de sensores ultrasónicos.

Parámetros	JSN-SR04T	HC-SR04	MB1240	MB7076
SySR1	1	0	1	1
SySR17	1	0	0	1
SRSH8	1	1	1	1
SRSH11	1	1	1	0
SRSH18	1	1	1	0
Puntuación	5	3	4	3

Cumple “1”

No cumple “0”

Fuente: Elaborado por el autor.

Tras verificar las opciones disponibles, se determinó que el sensor más adecuado para el sistema de monitoreo es el modelo JSN-SR04T. Esta elección se fundamenta no solo en las especificaciones técnicas presentadas en la comparativa, sino también en sus atributos físicos, como su capacidad de resistencia al agua y su cableado de 2.50 metros de longitud. Estas ventajas permiten posicionar la parte del sensor más próxima al agua, específicamente la bocina, a una distancia ideal, mientras que los demás componentes pueden instalarse en un área segura, lejos de posibles riesgos.

Sensor ultrasónico JSN-SR04T. Es un dispositivo de medición de distancia basado en la emisión y recepción de ondas ultrasónicas dentro de un rango operativo de 25 a 450 cm. Su diseño destaca por su resistencia al agua, dimensiones compactas, bajo consumo de energía y alta precisión, lo que lo hace adecuado para entornos expuestos a condiciones de humedad o agua. Este sensor es ampliamente utilizado en aplicaciones técnicas, como sistemas automotrices de asistencia al aparcamiento y detección de colisiones. (Mechatronics, 2023)

3.5.1.2. Análisis de sensores de precipitación

Para monitorear la precipitación en una determinada zona, se pueden utilizar diversos sensores diseñados para medir la cantidad de lluvia. Entre estos destacan los pluviómetros electrónicos y sensores de lluvia resistivos, los cuales están preparados para operar en condiciones climáticas adversas. Además, existen modelos especializados que permiten obtener datos en tiempo real sobre la intensidad y duración de la lluvia. Los sensores de precipitación disponibles en el mercado ofrecen diversas funcionalidades que garantizan una medición precisa y confiable, incluso en entornos desafiantes, lo que los convierte en una opción ideal para ser incorporados en el proyecto.

A continuación, se detallan algunas de las características principales que poseen los sensores de precipitación:

Tabla 11

Cuadro comparativo de sensores de precipitación.

Parámetros	YL-83	FC-37	Hydreon RG-11
Técnicos			
Figura			
Tipo de sensor	Analógico/Digital	Analógico/Digital	Óptico
Alimentación	3.3V – 5V	3.3V – 5V	12V
Corriente	15 mA	15 mA	20 mA
Tamaño del dispositivo	54 x 40 mm	50 x 40 mm	121 x 71 x 56 mm
Costo	\$ 4	\$ 3.25	\$60

Fuente: Elaborado por el autor.

Tabla 12

Análisis comparativo de sensores de precipitación.

Parámetros	YL-83	FC-37	Hydreon RG-11
SySR1	1	1	1
SySR17	1	1	1
SRSH8	1	1	0
SRSH11	1	1	1

SRSH18	1	1	0
Puntuación	5	5	3

Cumple “1”

No cumple “0”

Fuente: Elaborado por el autor.

Sensor de precipitación FC-37. Este sensor es capaz de detectar tanto la presencia como la ausencia de lluvia. Está compuesto por una placa con trazos metálicos separados por una distancia específica que impide su contacto directo. Al entrar en contacto con el agua, se forma una conexión entre estos trazos, lo que genera un valor de conductividad que se traduce en dos tipos de salida: un valor analógico entre 0 y 4095, y una señal digital que puede ser alta (HIGH) o baja (LOW). Un valor analógico cercano a 0 indica la presencia de lluvia, mientras que un valor próximo a 4095 señala la ausencia de esta. En la salida digital, un valor alto representa la presencia de lluvia y un valor bajo indica su ausencia. La sensibilidad del sensor se puede ajustar mediante un potenciómetro integrado en su circuito.

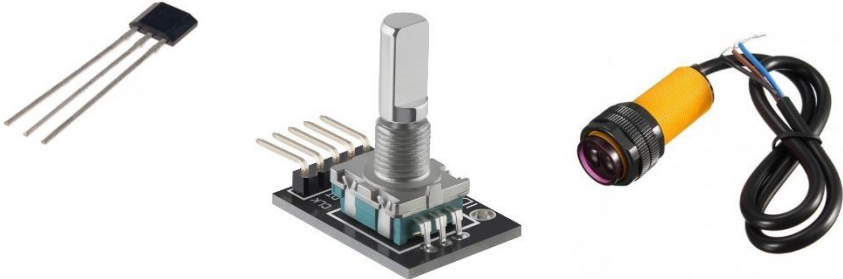
3.5.1.3. Análisis de módulos de rotación.

Los sensores de rotación son fundamentales para medir el movimiento angular de un objeto con alta precisión. Estos dispositivos se emplean en diversas aplicaciones, desde la automatización industrial hasta la robótica, permitiendo detectar y controlar la velocidad de rotación con un margen de error reducido. Su integración en sistemas tecnológicos mejora la precisión y eficiencia en procesos que requieren un control preciso del movimiento, optimizando el rendimiento en entornos industriales y automatizados.

A continuación, en la Tabla 13 se describen algunas de las características presentes en los sensores de rotación:

Tabla 13

Cuadro comparativo de sensor de rotación.

Parámetros técnicos	Sensor hall SS49E	Encoder rotativo KY-040	Sensor infrarrojo E18-D80NK
Figura			
Tipo de sensor	Efecto hall	Mecánico	Infrarrojo
Alimentación	4.5V - 6V	5V	5V
Corriente	10 mA	10 mA	20 mA
Tamaño del dispositivo	-	20 x 30 x 30mm	18 x 50mm
Costo	\$2.00	\$ 3,50	\$10,00

Fuente: Elaborado por el autor

Tabla 14

Análisis comparativo de sensor de rotación.

Parámetros	Sensor hall SS49E	Encoder rotativo KY-040	Sensor infrarrojo E18-D80NK
SySR1	0	1	1
SySR17	0	1	1
SRSH8	1	1	1
SRSH11	1	1	1
SRSH18	1	1	1
Puntuación	3	5	5
Cumple “1”			

No cumple “0”

Fuente: Elaborado por el autor.

Sensor infrarrojo E18-D80NK. El sensor de proximidad fotoeléctrico E18-D80NK permite la detección de objetos sin contacto en un rango máximo de 80 cm. Su integración con sistemas digitales como Arduino, PICs o PLCs es sencilla, ya que solo requiere conectar su salida a una entrada digital del microcontrolador. Este tipo de sensores opera midiendo variaciones en la intensidad de la luz para detectar la presencia de objetos. En particular, el modelo E18-D80NK es un sensor fotoeléctrico de tipo difuso, lo que significa que incorpora tanto el emisor como el receptor de luz en un solo dispositivo, eliminando la necesidad de un reflector adicional. (Mechatronics, 2023)

3.5.1.4. Análisis de placas microcontroladores.

La selección de un microcontrolador adecuado es esencial para garantizar un funcionamiento eficiente y rentable del sistema de monitoreo. Un diseño compacto facilita su integración en la estructura general del sistema, mientras que un bajo consumo de energía es crucial para prolongar la vida útil de la batería y minimizar los costos operativos.

En la Tabla 15, se realiza un análisis comparativo de las opciones disponibles en el mercado para identificar el microcontrolador más apropiado para el sistema de monitoreo. Este proceso de selección asegura que el dispositivo elegido cumpla con los requisitos específicos del proyecto, contribuyendo así al desarrollo de un sistema confiable y eficiente para la gestión del riesgo de inundaciones.

Tabla 15.

Cuadro comparativo de microcontroladores.

Parámetros técnicos	ESP32	WiFi LoRa 32 V3	ESP8266	Particle Photon
Figura				
Alimentación	5V	3.3V – 6V	5V	3.6V – 5V
Corriente	20 mA	>150 mA	80 mA	80 mA
Memoria Flash	16 MB	8 MB	4 MB	1 MB
Memoria RAM	520kB	512 kB	96 kB	128 kB
Frecuencia	Hasta 240 MHz	470-510MHz, 863-928MHz	80 MHz	120 MHz
Pines digitales I/O	36	26	17	18
Pines de entrada analógica	15	9	1	6
Wi-Fi	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Bluetooth	Si	Si	No	No
Costo	\$ 14	\$ 62	\$ 12	\$ 40

Fuente: (Ramírez, 2023)

Tabla 16.

Análisis comparativo de microcontroladores.

Parámetros	ESP32	ESP32 LoRa	ESP8266	Particle
		V3		Photon
SRSH8	1	1	1	1
SRSH16	1	1	1	1
SRSH17	1	1	1	1
SRSH19	0	1	0	0
Puntuación	3	4	3	3
Cumple “1”				
No cumple “0”				
Fuente: Elaborado por el autor.				

WiFi LoRa 32 V3. Es una placa de desarrollo avanzada que incorpora un microcontrolador ESP32 de doble núcleo con una velocidad de hasta 240 MHz y un rendimiento de 600 DMIPS. Este dispositivo ofrece conectividad WiFi, Bluetooth BLE y tecnología LoRa en el rango de 868 a 915 MHz. Además, cuenta con una pantalla OLED integrada para la visualización de datos, un puerto de carga para baterías LiPo y una interfaz USB a Serial CP2102 que facilita la programación mediante el entorno Arduino IDE.

Por su parte, el módulo WiFi LoRa 32 se presenta como una solución integral para proyectos de IoT, destacándose como una opción ideal para aplicaciones en ciudades inteligentes, agricultura de precisión, automatización del hogar y desarrollos relacionados con el Internet de las cosas.

3.5.1.5. *Análisis de Chips LoRa*

Es crucial contar con información detallada sobre los diversos chips LoRa para maximizar el rendimiento y la flexibilidad de los sistemas de comunicación inalámbrica en aplicaciones de IoT. Entender sus diferencias facilita la elección del chip más apropiado, alineado con las necesidades específicas del proyecto.

A continuación, se puede observar la Tabla 17 en donde se compara los diferentes chips LoRa dentro del mercado:

Tabla 17.

Cuadro comparativo de chips LoRa.

Parámetros	SX1262	SX1278	RFM95W	RAK3172
técnicos				
Figura				
Alimentación	1.8V – 3.7V	1.8V – 3.7V	1.8V – 3.7V	2.0V – 3.6V
Frecuencia	150 MHz – 960 MHz	433 MHz, 868 MHz y 915 mHz	915 Mhz	915 MHz – 928 MHz
Potencia	+22 dBm	+20 dBm	20 dBm	20 dBm
Modulación	LoRa, FSK, GFSK, MSK, GMSK.	LoRa, FSK, GFSK, MSK, GMSK.	FSK, GFSK, MSK, GMSK,	LoRaWAN, LoRa, FSK, GFSK.

LoRaTM y				
OOK.				
Módulos compatibles	ESP32	RA-02 Ai-Thinker	ESP32	ESP32

Fuente: Elaborado por el autor.

Tabla 18.

Análisis comparativo de Chips LoRa.

Parámetros	SX1262	SX1278	RFM95W	RAK3172
SySR2	1	1	1	1
SRSH8	1	0	1	1
SRSH10	1	1	1	1
SRSH16	1	1	1	1
SRSH18	1	1	0	0
Puntuación	5	4	4	4
Cumple “1”				
No cumple “0”				

Fuente: Elaborado por el autor.

El chip SX1262 es una de las opciones más adecuadas para el WiFi LoRa 32, ya que permite establecer comunicaciones inalámbricas de largo alcance con un bajo consumo energético. Su compatibilidad con la tecnología LoRaWAN y su amplio rango de frecuencias de 150 MHz a 960 MHz lo convierten en una solución ideal para redes IoT. En comparación con otros chips, el SX1262 ofrece mejor cobertura y mayor eficiencia en entornos con poca señal. Además, cuenta con una interfaz SPI de alta velocidad, lo que facilita su integración con el ESP32 y optimiza el desarrollo de aplicaciones en el ámbito IoT.

3.5.1.6. *Análisis de Panel Solar.*

Dentro del nodo sensor es necesario colocar un panel solar, ya que es muy útil para que los dispositivos dentro del nodo sigan operando. Además, al colocar un panel solar representa otro método de obtener energía en lugares en donde no existe el acceso directo a la energía.

A continuación, dentro de la Tabla 19, se observa un cuadro comparativo de diferentes paneles solares que son compatibles con el nodo sensor.

Tabla 19.

Cuadro comparativo de Panel Solar.

Parámetro	Panel solar 6v	Panel solar	Panel solar 12v
s técnicos	200mah	6v 140mah	
Figura 			
Voltaje	6V	6V	12V
Capacidad nominal	200 mA	140 mA	100 mA
Tamaño	120x90mm	120x60mm	120x90mm
Costo	\$9.50	\$4.00	\$10.00

Fuente: Autoría.

Tabla 20.

Análisis comparativo de Panel Solar.

Parámetros	Panel solar 6v 200mah	Panel solar 6v 140mah	Panel solar 12v
StRS2	1	1	1
StRS5	1	1	1
SySR9	1	1	1
SySR20	1	0	1
SRS4	1	0	0
Puntuación	5	3	4
Cumple “1” No cumple “0”			

Fuente: Autoría.

El panel solar de 6v 200mA resultó cumplir con los requerimientos del sistema. Este panel solar contiene material de silicio monocristalino. Este tipo de panel solar son utilizados para recargar baterías, lo cual resulta muy útil para este proyecto.

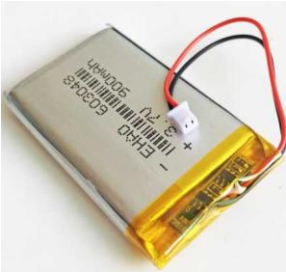
3.5.1.7. Análisis de baterías para el sistema.

En el sistema de monitoreo es adecuado utilizar baterías, esto es fundamental para garantizar el funcionamiento eficiente y continuo del sistema. Las baterías deben ser adecuadas para el uso en sistemas inteligentes, unidades de monitoreo y dispositivos electrónicos. Dado que se tiene un sistema inalámbrico, contar con una fuente de energía permite mantener el sistema activo y sin interrupciones.

A continuación, en la Tabla 21, se realizará una comparación de los tipos de baterías que existen dentro del mercado y que son compatibles con el sistema.

Tabla 21.

Cuadro comparativo de batería.

Parámetros técnicos	PL 706090	PK LP803860	EHA0 603048
Figura			
Voltaje	3.7 V	3.7 V	3.7 V
Capacidad nominal	5000 mah	2600 mah	900 mah
Cargable	Si	Si	Si
Tamaño	9x6x0.7cm	2.7x2.5x2cm	2.4x1.8x1.8cm
Costo	\$15.00	\$8.50	\$18.00

Fuente: Elaborado por el autor.

Tabla 22.

Análisis comparativo de batería.

Parámetros	PL 706090	PK LP803860	EHA0 603048
StRS2	1	1	1
SrRS4	1	1	1
StRS5	1	1	1

SySR20	1	1	1
SRSH4	1	1	0
Puntuación	5	5	4
Cumple “1”			
No cumple “0”			
Fuente: Elaborado por el autor.			

La elección de la batería es un aspecto crucial en el sistema de monitoreo, por lo que se eligió el modelo PK LP803860 debido a su capacidad para garantizar un rendimiento continuo, especialmente en entornos sin acceso a la red eléctrica. Esta batería destaca por su fiabilidad y eficiencia, asegurando que el sistema pueda capturar y transmitir datos de manera ininterrumpida.

3.5.1.8. *Análisis de fuente de alimentación.*

Los componentes electrónicos del sistema requieren tener una fuente eléctrica para lograr un buen funcionamiento, en especial el nodo sensor, que se encarga de recolectar datos y es el nodo que debe permanecer en constante actividad. El sistema contiene un apartado para ahorrar energía y aumentar el rendimiento de los elementos electrónicos. Dado que el sistema utiliza energía de manera considerada, fue necesario realizar un análisis del consumo energético de cada componente electrónico, esto con el fin de adquirir una batería recargable adecuada para el sistema.

Tabla 23.

Análisis de consumo eléctrico de nodo sensor.

Componente	Consumo en	Modo Sleep
electrónico	funcionamiento	

Sensor ultrasónico		
JSN-SR04T	30mA	2mA
Sensor de		
precipitación FC-	15mA	---
37		
Sensor infrarrojo		
E18-D80NK	20mA	---
WiFi LoRa 32	150mA	1mA
TOTAL	215mA	3mA

Fuente: Autoría.

Al obtener un amperaje del nodo sensor de 215 mA, es necesario realizar el cálculo para conocer el tipo de batería que se requiere para el sistema. Para llevar a cabo el cálculo se utiliza la ecuación (1).

$$\text{Consumo de batería} = \frac{(T_{cf} * I_{cf}) + (T_{ms} * I_{ms})}{T_{cf} + T_{ms}} \quad (1)$$

Donde:

T_{cf} : Tiempo de consumo en funcionamiento.

I_{cf} : Intensidad de corriente en funcionamiento.

T_{ms} : Tiempo en modo sleep.

I_{ms} : Intensidad de corriente en modo sleep.

A continuación, se realiza el cálculo con los datos necesarios para la ecuación.

$$T_{cf}: 10 \text{ min} = 600 \text{ seg.}$$

$$I_{cf}: 215 \text{ mA}$$

$$T_{ms}: 30 \text{ min} = 1800 \text{ seg.}$$

$$I_{ms}: 3 \text{ mA}$$

$$\text{Consumo de batería} = \frac{(600 \cdot 215) + (1800 \cdot 3)}{600 + 1800}$$

$$\text{Consumo de batería} = 56 \text{ mA}$$

Al resultado obtenido de consumo de batería fue necesario utilizar una batería de 3.7v con 2600 mAh recargable para brindar mayor durabilidad al nodo sensor.

Al obtener el consumo total de la batería, se realiza el cálculo de duración de la batería dentro del nodo sensor, para el cálculo se utiliza la ecuación (2).

$$\text{Duración: } \frac{\text{Capacidad de batería mAh}}{\text{Consumo de batería}} \quad (2)$$

$$\text{Duración} = \frac{2600 \text{ mAh}}{56 \text{ mA}}$$

$$\text{Duración} = 46.4 \text{ h}$$

El nodo sensor contiene un panel solar que se encarga de ayudar con la carga de la batería, para realizar el proceso de la carga, se lleva a cabo el siguiente cálculo.

Tomando en cuenta que el valor de 0.8 corresponde al 80% de eficiencia de la batería ya que puede existir pérdidas de energía por diferentes ocasiones.

$$\text{Corriente} = 200 \text{ mA} \times 0.8 = 160 \text{ mA}$$

$$\text{Tiempo de carga} = \frac{2600 \text{ mAh}}{160 \text{ mA}} = 16 \text{ h de carga del } 100 \%$$

La batería utilizada dentro del sistema cuenta con 300 ciclos, esto indica que puede durar hasta 8 meses y al momento de cumplir con los ciclos de la batería, se mantendrá al menos un 75-80% de la capacidad original.

A continuación, se realiza el cálculo de consumo energético durante la noche aproximadamente 12 h, con un consumo de 56mA y con la batería de 2600mAh.

$$\text{Corriente consumida} = 56 \text{ mA} \times 12 \text{ h} = 672 \text{ mAh}$$

$$\text{Porcentaje de consumo} = \frac{672 \text{ mAh}}{2600 \text{ mAh}} = 0.25\% = 25\%$$

El resultado indica el 25% de consumo de la batería durante la noche.

3.5.2. Software destinado para el sistema

El software seleccionado debe estar alineado con los requisitos establecidos en la etapa inicial del proyecto. Esto asegura un rendimiento óptimo, una integración eficiente entre los componentes lógicos y físicos del sistema, y aprovecha las capacidades ofrecidas por cada plataforma tecnológica evaluada.

Tabla 24.

Requerimientos de software para el entorno de desarrollo del sistema

Entorno de desarrollo	Requerimientos				Valoración
	SRSH3	SRSH7	SRSH8	SRSH12	total
IDE					
ARDUINO	1	1	1	1	4
PYTHON	1	1	0	1	3
JAVA	1	0	0	1	2

Cumple “1”
No cumple “0”

Fuente: Elaborado por el autor

A partir de la Tabla 24, se concluye que el software que satisface la mayor cantidad de requisitos necesarios para el desarrollo del sistema es el IDE Arduino. Por esta razón, se optó por seleccionarlo.

3.5.2.1. *Software de programación para el bloque de recolección de datos.*

La selección del software destinado para la recopilación de datos, junto con el análisis comparativo de hardware de la Tabla 15, facilita la identificación del software más adecuado para tener un mejor rendimiento con el WiFi LoRa 32. La Tabla 27 muestra la elección del software adecuado para el bloque de recopilación de datos.

Tabla 25.

Requerimientos de software para la recolección de datos

Parámetro	Lenguaje de programación	
	IDE Arduino	Python
SySR2	1	0
SRSH3	1	1
SRSH7	1	1
SRSH13	1	0
Puntuación	4	2

Cumple “1”

No cumple “0”

Fuente: Elaborado por el autor.

La elección del software Arduino IDE es ideal para realizar el proceso de recolección de datos, hay que tomar en cuenta que se utiliza un WiFi LoRa 32 y el entorno de Arduino es ideal para este tipo de microcontroladores.

3.5.2.2. *Software destinado para el desarrollo de la interfaz gráfica*

La funcionalidad del sistema se reflejará en el diseño de una interfaz gráfica adecuada, robusta y con rápido acoplamiento. Esta garantizará un alto desempeño y aprovechará herramientas que potencien el hardware. Actualmente, existen numerosas plataformas que permiten desarrollar interfaces gráficas robustas y funcionales. Las opciones disponibles para la creación de aplicaciones gráficas se presentan en la Tabla 26.

Tabla 26.

Requerimientos de software para la interfaz gráfica.

Herramienta	Requerimientos			Valoración total
	SRSH1	SRSH4	SRSH6	
ANGULAR	1	1	1	3
REACT	0	1	1	2
NATIVE				
Cumple “1”				
No cumple “0”				

Fuente: Elaborado por el autor

A partir de la Tabla 26, se concluye que el software que satisface la mayor cantidad de requisitos necesarios para el desarrollo del sistema es Angular. Por esta razón, se decidió seleccionarlo para el proyecto.

Angular es una plataforma de desarrollo basada en TypeScript que permite construir aplicaciones web escalables mediante un enfoque basado en componentes. Incluye una colección integrada de bibliotecas que abarcan funcionalidades como enrutamiento, gestión de formularios y comunicación cliente-servidor. También proporciona herramientas para desarrollar, compilar, probar y actualizar aplicaciones. Es un framework de JavaScript de código abierto diseñado para crear aplicaciones de una sola página (SPA). Google se encarga de su mantenimiento, garantizando actualizaciones frecuentes y mejoras continuas. (Goncalves, 2024)

3.5.2.3. *Software compatible para el entrenamiento de datos*

La plataforma en la que se llevara a cabo el entrenamiento debe cumplir con todos los requerimientos establecidos en la Tabla 27. Es fundamental que la plataforma debe permitir automatizar trabajos complejos, monitorear el rendimiento del modelo y visualizar de mejor manera los resultados, con el fin de lograr con precisión el desarrollo de soluciones basadas en IA.

En la Tabla 27, se realiza la comparación de las diferentes plataformas que permiten el entrenamiento de datos.

Tabla 27.

Requerimientos de software para el entrenamiento de datos.

Parámetro	Plataforma de entrenamiento		
	Google Colab	Kaggle	Amazon SageMaker
StRS3	1	1	1
SySR3	1	1	1
SRSH3	1	1	0
SRSH7	1	1	1

SRSH8	1	1	0
SRSH9	1	1	1
SRSH12	1	1	1
SRSH13	1	0	1
Puntuación	8	7	6
Cumple “1”			
No cumple “0”			
Fuente: Elaborado por el autor.			

3.5.2.4. *Plataforma de notificación.*

Para seleccionar la plataforma de notificación sobre el nivel de riesgo, se han considerado dos opciones de notificación ampliamente utilizadas en la actualidad, junto con una alternativa desarrollada específicamente para el sistema mediante el software Ionic, una extensión de Angular diseñada para la creación de aplicaciones móviles.

Tabla 28.

Requerimientos de herramienta de notificaciones

Herramienta	Requerimientos		Valoración
	StRS8	SySR12	total
WhatsApp	1	0	1
Telegram	1	1	2
App Sistema	1	1	2
Cumple “1”			
No cumple “0”			
Fuente: Elaborado por el autor			

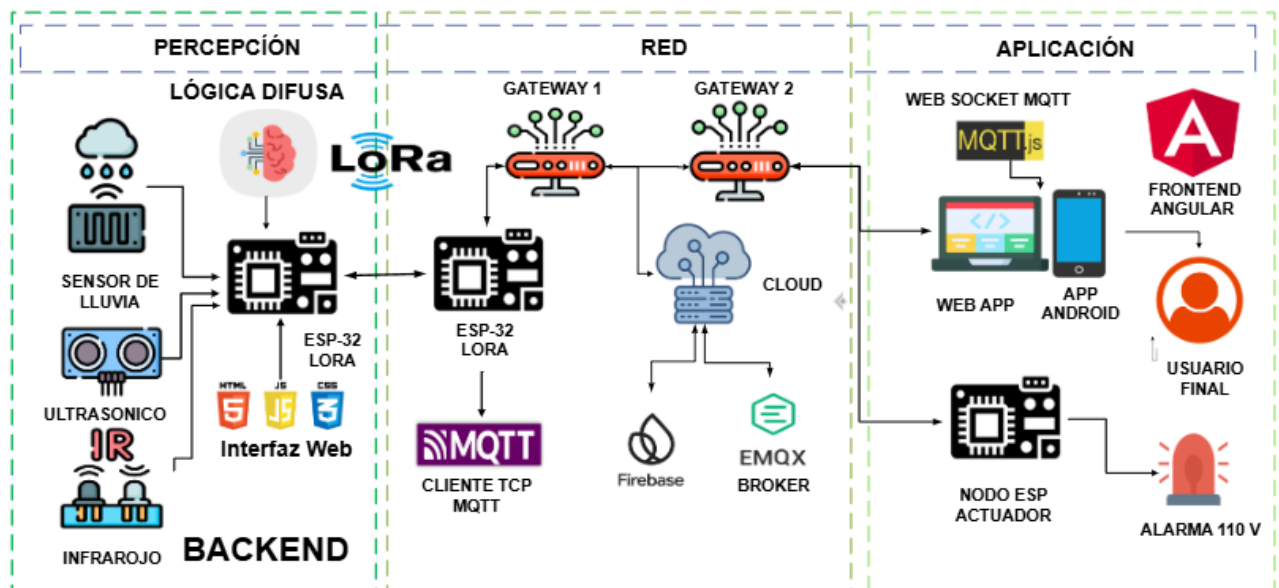
Para lograr la elección de la herramienta de notificaciones sobre los tipos de alertas que envía el sistema de monitoreo se establece dos herramientas de mensajería, estas son muy populares actualmente y una que fue desarrollada para el sistema. A partir de la Tabla 28, se opta por la aplicación propia para el sistema ya que cumple con los requisitos necesarios para incorporar la opción de Bot y a la vez de uso gratuito para implementar dentro del diseño del sistema de monitoreo.

3.6. Diseño del sistema

El esquema de este sistema se desarrolla utilizando el modelo en V, el cual permite identificar de manera técnica y precisa los coeficientes críticos para el análisis de este estudio. Esto asegura la selección adecuada de herramientas y componentes para gestionar eficientemente el sistema, logrando que el monitoreo basado en lógica difusa opere de manera autónoma. Tras la selección minuciosa de los elementos de hardware y software, se procede a integrar el prototipo, garantizando su funcionalidad y compatibilidad técnica en el contexto del diseño propuesto.

Figura 16.

Arquitectura del sistema de monitoreo.



Fuente: Elaborado por el autor.

En la Figura 16, se logra visualizar la arquitectura en donde se compone por tres bloques: percepción, red y aplicación. El sistema de monitoreo parte desde el bloque de percepción, en donde, los datos son recolectados directamente desde el entorno mediante los sensores. Se utiliza un sensor de precipitación, un sensor ultrasónico para medir el nivel de agua del río y un sensor que se encargara de medir la velocidad de flujo del río. Estos sensores están conectados al módulo WiFi LoRa 32. Adicionalmente, se emplea una interfaz web establecida con tecnologías como son HTML, CSS y JavaScript para lograr visualizar los valores capturados en tiempo real. Los datos recolectados en el nodo sensor son aplicados lógica difusa para ser enviados al siguiente nodo mediante tecnología LoRa. Dentro del bloque de etapa de red, ocurre el proceso de envío de los datos a través de una red LoRa que actúa como receptor y transmisor de los datos, este nodo está conectado a un gateway. Este nodo se encarga de reenviar los datos a un broker MQTT en línea, para este proceso se utiliza la herramienta EMQX para la gestión de los mensajes, adicionalmente se consume una API proporcionada por Google, la cual permite acceder al servicio de Firebase.

Se aplica un cliente TCP MQTT para afirmar una comunicación robusta y confiable entre los dispositivos LoRa. En la tercera y última etapa del sistema, es en la que interactúa el usuario final. A través del broker utilizado, varios usuarios pueden observar los datos, de igual manera, en esta etapa se encuentra el sitio web y aplicación del sistema de monitoreo desarrolladas en Angular y utilizando MQTT.js permitiendo observar los datos procesados y el estado de alerta. Estos datos generados pueden ser guardados gracias a la comunicación con el servicio de Firebase. Por último, dentro de esta etapa se encuentra el nodo actuador que se encarga de estar a la espera del mensaje que indica que los niveles de alerta son altos y se activará una sirena para avisar a la gran mayoría de la Parroquia.

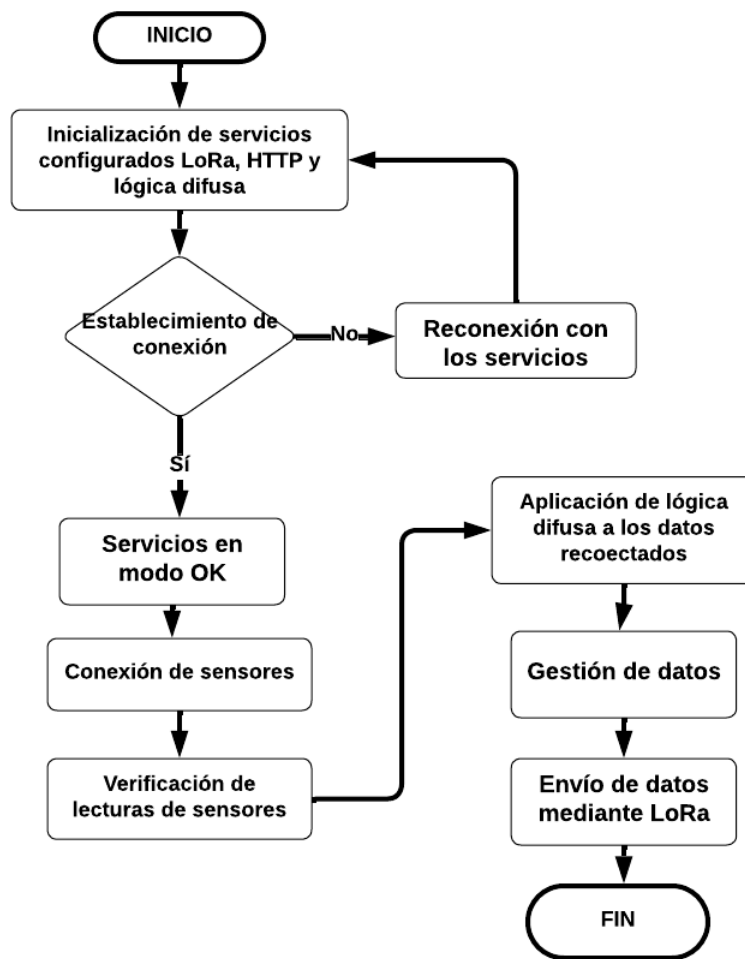
3.6.1. Diagramas de flujo del sistema

En los diagramas de flujo del sistema consiste en dar a conocer cómo funciona cada etapa que forma parte del sistema de monitoreo. Este tipo de diagrama se encarga del proceso lógico de inicio, configuración y operación para tomar y enviar datos. El nodo sensor inicializa los procesos de variables de control, configuración e inicio de interfaz inalámbrica, inicio de servidor web asíncrono, inicio de interfaz LoRa e inicio de lógica difusa. Al completar el proceso de inicialización, el nodo inicia el proceso de tomar datos de cada sensor, estos datos pueden ser vistos mediante la red que contiene el nodo desde cualquier dispositivo. Una vez obtenidos los datos, comienzan a ser procesados aplicando la lógica difusa, la cual transforma los datos numéricos en variables cualitativos para facilitar la toma de decisiones. Los datos al pasar por el proceso de fusificación están listos para ser enviados hacia el nodo coordinador.

Dentro de la siguiente figura 17 se indica el diagrama de flujo del nodo sensor aplicado dentro del sistema.

Figura 17.

Diagrama de flujo de nodo sensor.



Fuente: Elaborado por el autor

Por otro lado, se da a conocer cómo se lleva a cabo todo el proceso el nodo coordinador. Al iniciar el nodo coordinador, empieza a inicializar todas las configuraciones establecidas como son las variables de control, servicio de MQTT, servicio a Firebase, servicio web asíncrono, inicio de interfaz LoRa y el servicio de IA. Al iniciar todos los procesos, se procede a configurar el acceso a internet y se establece la sincronización con el nodo sensor utilizando la tecnología LoRa. Al completar la inicialización con distintos servicios y configuraciones, el nodo coordinador comienza a

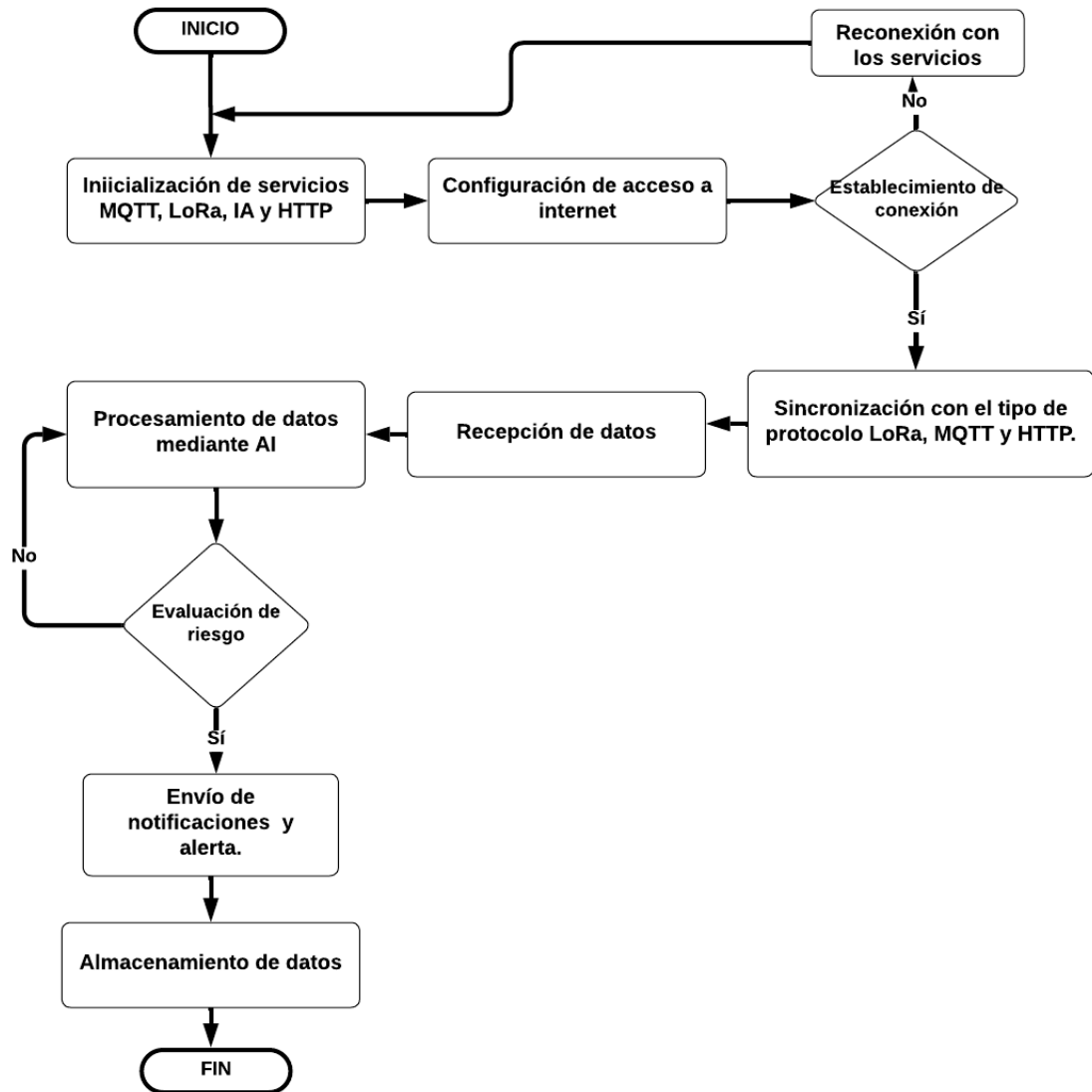
recibir los datos que son transmitidos desde el nodo sensor. Estos datos recibidos son procesados mediante la IA implementada, si se detecta un nivel de riesgo, automáticamente se envía una notificación de alerta.

Por otro lado, si las lecturas de los sensores se mantienen normales, el nodo coordinador continúa procesando los datos junto con la IA, de esa manera el tiempo de toma de datos se ajustará. Este ajuste busca optimizar recursos energéticos y evitar la saturación del sistema con datos innecesarios en intervalos de tiempo muy cortos. En caso de detectar niveles de lectura altos, el sistema responde de forma autónoma reduciendo el tiempo de captura de datos, con el fin de detectar rápidamente cualquier anomalía que puede significar un riesgo.

A continuación, dentro de la Figura 18 se da a conocer el diagrama en general del proceso que cumple el nodo coordinador dentro del sistema.

Figura 18.

Diagrama de flujo de nodo coordinador.



Fuente: Elaborado por el autor.

3.6.2. Implementación de Hardware

En cuanto a la implementación del hardware, se lleva a cabo el proceso de conexión eléctrica de los distintos sensores y el microcontrolador WiFi LoRa 32.

A continuación, se detallan las conexiones correspondientes a cada uno de los bloques descritos en la Figura 16.

3.6.2.1. Bloque de Percepción

En esta sección se encarga de capturar magnitudes físicas y transformarlas en señales eléctricas mediante el uso de sensores específicos para cada una de las variables establecidas. Se conecta al sistema de recopilación de datos a través del microcontrolador WiFi LoRa 32, y las conexiones eléctricas correspondientes se especifican en el diagrama del circuito para cada sensor.

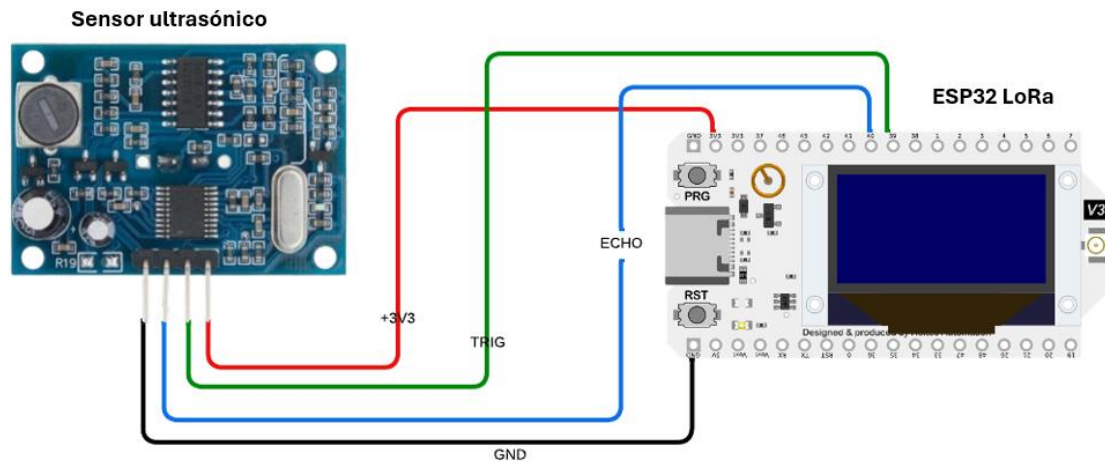
La principal función de este nodo es la recolección de datos del nivel del caudal del río, intensidad de precipitación y la velocidad a la que se encuentra el río, y mediante la tecnología LoRa se establece la comunicación entre los dos nodos.

Sensor JSN-SR04T. Este sensor consta de cuatro terminales (GND, Vcc, TRIGGER, ECHO), es un dispositivo que calcula distancias empleando tecnología de ultrasonido, cuyos terminales se conectan directamente al WiFi LoRa 32, tal como se muestra en la Figura 19.

- Voltaje – 3v3 (cable rojo)
- Negativo – GND (cable negro)
- Echo – pin 40 (cable celeste)
- Trigger – pin 39 (cable verde)

Figura 19.

Diagrama de conexión de sensor JSN-SR04T.



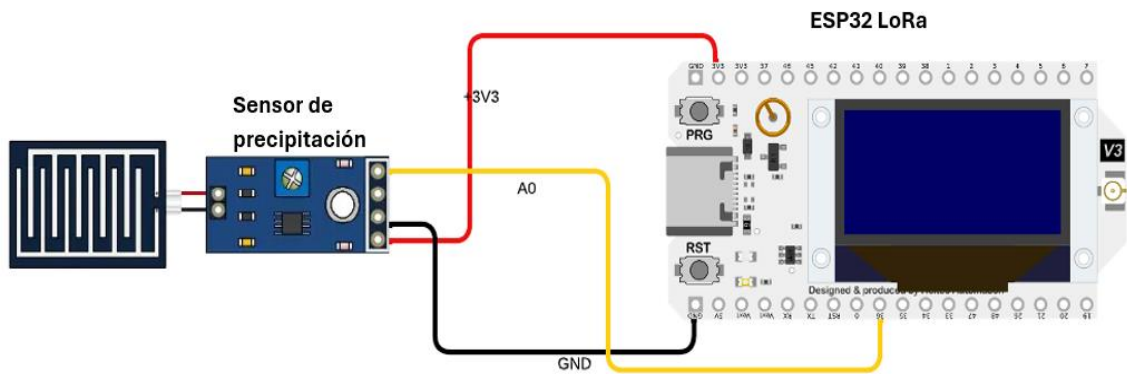
Fuente: Elaborado por el autor.

Sensor FC-37. Este sensor consta de cuatro terminales: GNS, Vcc, Salida digital y analógica. Dentro del proyecto se utiliza el terminal de salida analógica para recibir los datos de precipitación, como se puede ver en la Figura 20.

- Voltaje – 3V3 (cable rojo)
- Negativo – GND (cable negro)
- A0 – pin 36 (cable amarillo)

Figura 20.

Diagrama de conexión de sensor FC-37.



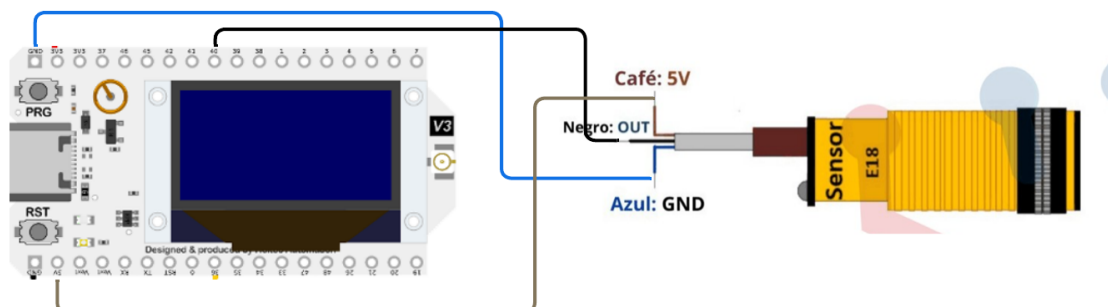
Fuente: Elaborado por el autor.

Sensor Infrarrojo. Este dispositivo cuenta con dos terminales: Vcc, GND y OUT. Para el desarrollo del proyecto, el terminal OUT se encarga de recibir las lecturas del molino, de igual manera, los terminales VCC y GND corresponden a la fuente de 5v y neutro. La Figura 21 ilustra la conexión con mayor claridad.

- Voltaje – 5V (cable café)
- Negativo – GND (cable azul)
- OUT – pin 40 (cable negro)

Figura 21.

Diagrama de conexión de sensor infrarrojo.

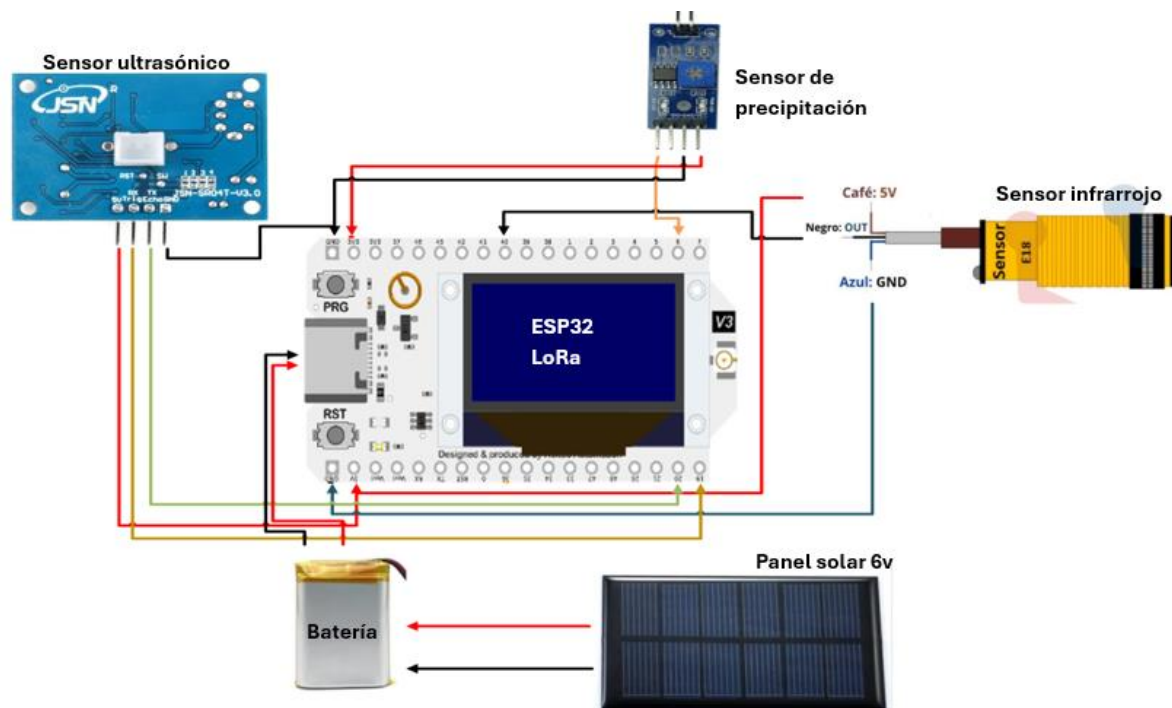


Fuente: Elaborado por el autor.

A continuación, se visualiza en la Figura 22 el diagrama de conexión general del nodo sensor, en donde, se logra observar las conexiones de todos los componentes que lleva el nodo sensor.

Figura 22.

Diagrama de conexión general del nodo sensor.



Fuente: Elaborado por el autor.

Diseño de comunicación Inalámbrica

Se emplean los módulos WiFi LoRa 32, ya que incorporan el chip LoRa SX1262, lo que los convierte en microcontroladores versátiles con diversas funcionalidades para la comunicación inalámbrica. Además, se utiliza el software Arduino IDE para la configuración del módulo y la asignación de su modo de operación, permitiendo definirlo como transmisor, coordinador, entre otros.

En la Figura 23, se presenta el diseño de los dos nodos conectados mediante la tecnología LoRa, operando a una frecuencia de 915MHz.

Figura 23.

Diseño de comunicación inalámbrica de los dos nodos.



Fuente: Elaborado por el autor.

3.6.3. Bloque de Red

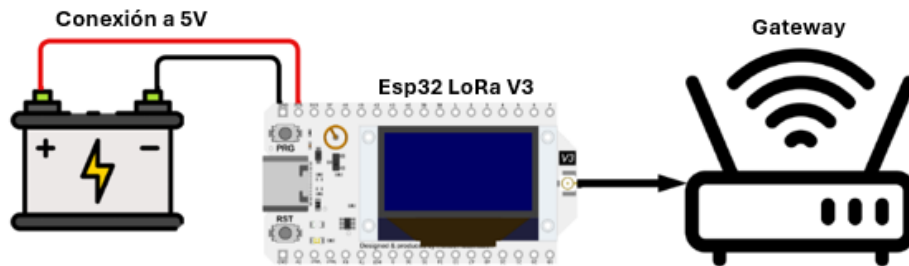
En este bloque se implementa la comunicación mediante la tecnología LoRa, permitiendo la recepción y transmisión de información a largas distancias con un bajo consumo de energía. Además, se incluye el nodo coordinador, que funciona como un transmisor, encargado de enviar los datos recopilados junto con el Gateway para su posterior procesamiento y distribución.

3.6.3.1. Diagrama de conexión de nodo coordinador

El nodo coordinador está encargado de la recepción y procesamiento de los datos entre el WiFi LoRa 32 y el gateway. La conexión se logra visualizar dentro de la Figura 24.

Figura 24.

Diagrama de conexiones de nodo sensor.



Fuente: Elaborado por el autor

Cuando el nodo coordinador envía la información a través del gateway, la retransmite a un Broker MQTT en línea mediante el protocolo MQTT. Esto permite que otros dispositivos puedan acceder a los datos en tiempo real y visualizarlos de manera remota.

3.6.4. Bloque de Aplicación y diseño de software

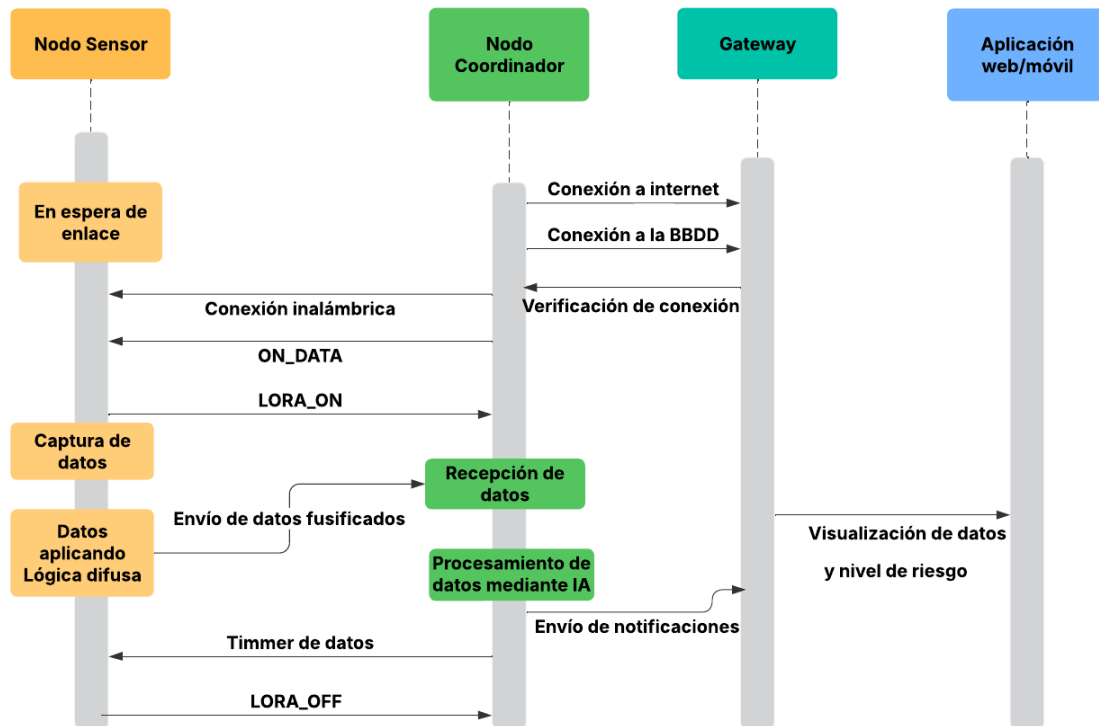
En base a la información anterior de implementación de hardware en el sistema de monitoreo, dentro de esta sección se desarrollará el diseño de software que fue formado en cada etapa del sistema.

3.6.4.1. *Recopilación de datos*

Dentro de este apartado, se define un diagrama de secuencia en el cual se describe el proceso final de recopilación y transmisión de datos de forma general en la última etapa del sistema. De esta manera, se comprende el flujo de datos desde los nodos hasta la interfaz final del cliente.

Figura 25.

Diagrama de secuencia de recopilación de datos.



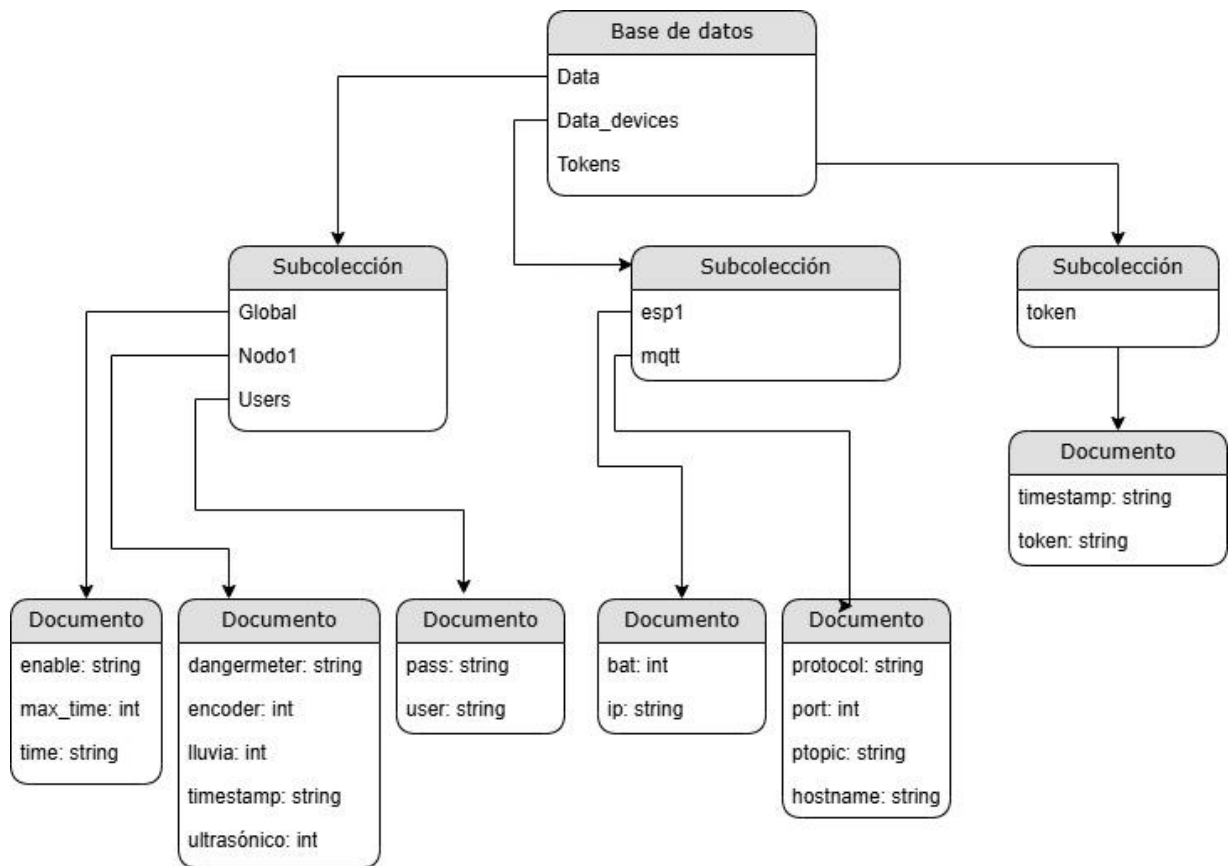
Fuente: Elaborado por el autor.

3.6.4.2. *Diseño de base de datos*

La base de datos desempeña un papel fundamental en el almacenamiento y gestión de la información recopilada. Su principal función es registrar de manera estructurada y en tiempo real los datos relacionados con el nivel del agua en el río, la precipitación y la velocidad del caudal. Esto permite un monitoreo eficiente y oportuno de las condiciones del río, facilitando la toma de decisiones en situaciones de posible riesgo. Al utilizar la plataforma de Firebase como alojamiento de los datos, es importante tomar en cuenta que es una base de datos NoSQL basada en la nube. Al ser NoSQL, permite trabajar con lenguajes JavaScript y ayuda al usuario a especificar mediante colecciones y documentos una estructura de datos con la que va a funcionar el sistema. En la Figura 26, se ilustra el proceso de funcionamiento de la plataforma Firebase, destacando su organización y accesibilidad para su posterior análisis.

Figura 26.

Diagrama de base de datos NoSQL de Firebase Firestore.



Fuente: Elaborado por el autor

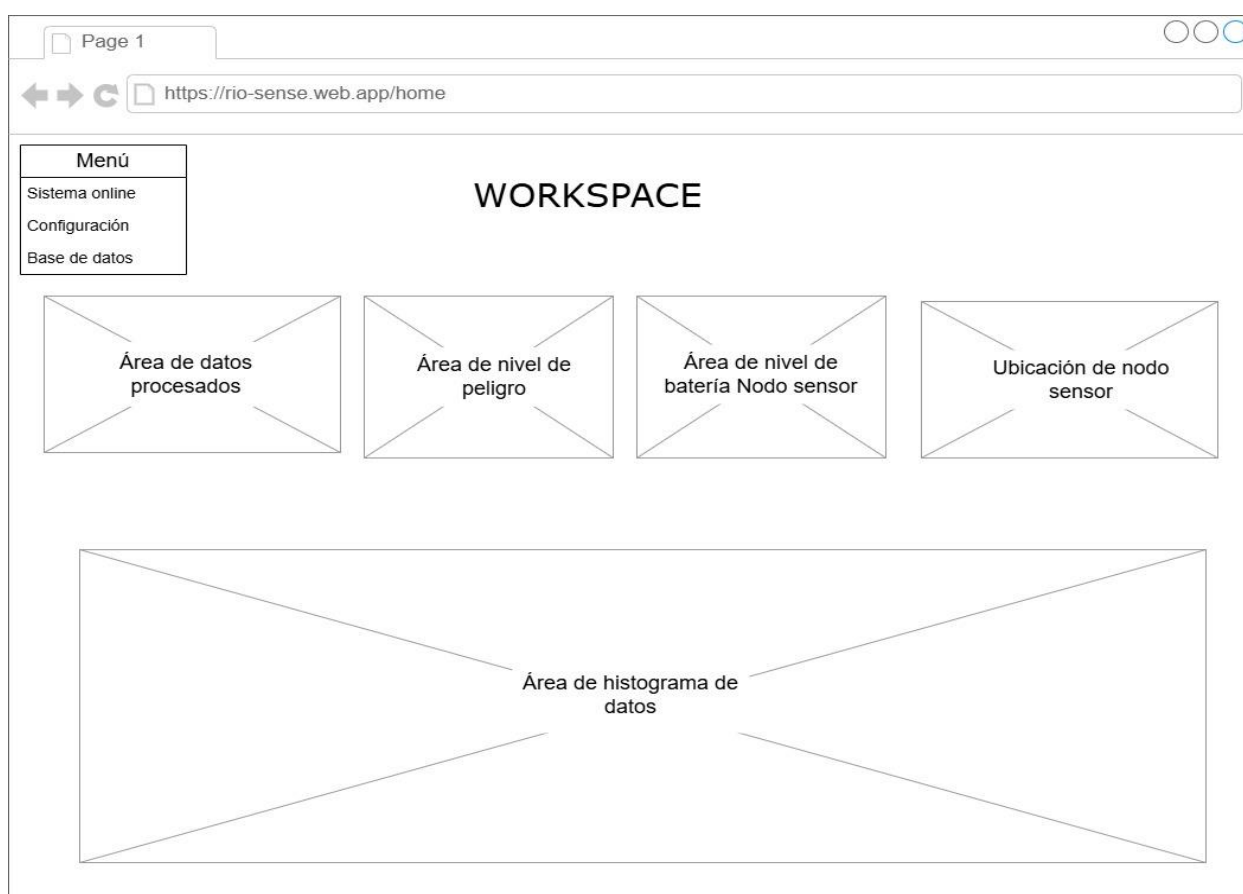
3.6.4.3. *Desarrollo de sitio web*

La página web desarrollada utiliza JavaScript como lenguaje base y está diseñada para crear una interfaz dinámica dentro de una sola página. Su objetivo principal es permitir el monitoreo en tiempo real de los datos obtenidos del nodo de sensores a través de internet.

Dentro de la Figura 27, se da a conocer cómo debe estar diseñada la interfaz para ser intuitiva y funcional, integrando la configuración, monitoreo y visualización de los datos dentro de un solo sitio.

Figura 27.

Diseño de frontend del sistema.



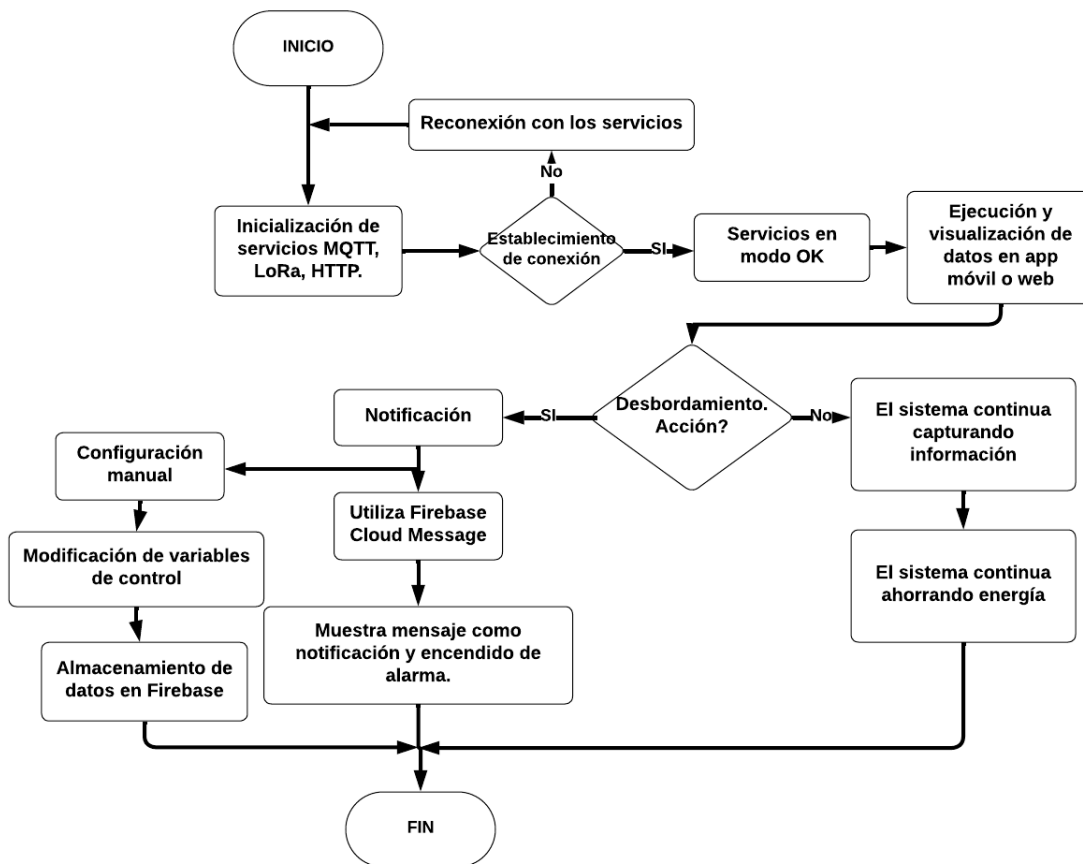
Fuente: Elaborado por el autor

3.6.4.4. Proceso de enlace del sistema a la aplicación móvil

El sistema de monitoreo implementado establece una comunicación eficiente con la aplicación móvil mediante el protocolo MQTT, permitiendo la transmisión de los datos obtenidos por los sensores. La información capturada es enviada desde el nodo sensor hacia el nodo coordinador, el cual la procesa y reenvía a la aplicación móvil para su visualización. Esto garantiza una respuesta rápida ante condiciones críticas y facilita la toma de decisiones. En la Figura 28 se presenta el diagrama de flujo del proceso de funcionamiento y transmisión de datos hacia la aplicación móvil.

Figura 28.

Diagrama de flujo de notificación del sistema.



Fuente: Elaborado por el autor.

En la Figura 29 se presenta el diseño de la interfaz de la aplicación móvil, la cual ha sido concebida para ser intuitiva y funcional. Este diseño integra de manera eficiente las funciones de configuración, monitoreo y visualización de datos dentro de la app.

Figura 29.

Diseño de interfaz de notificación celular del sistema.



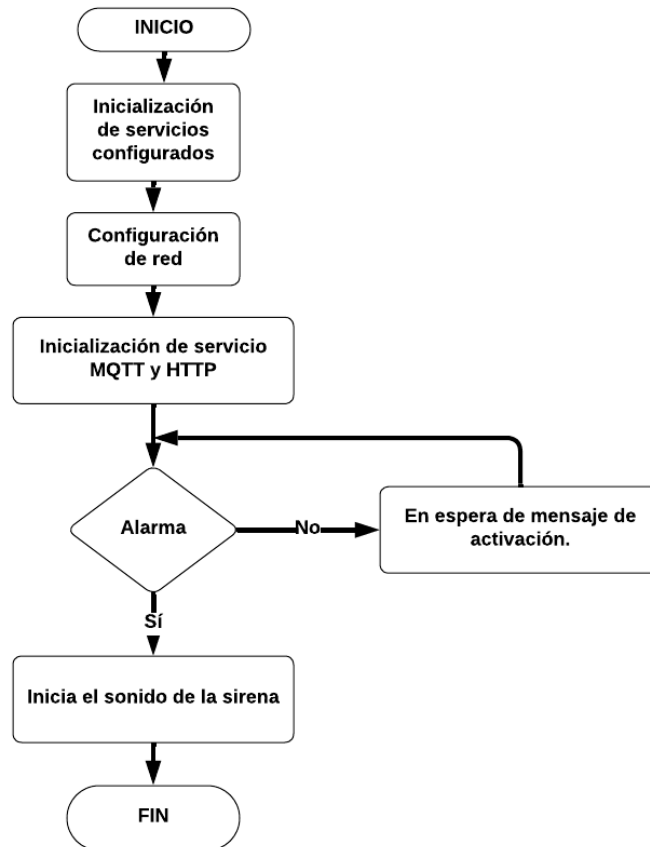
Fuente: Elaborado por el autor.

3.6.4.5. *Nodo Actuador*

En el bloque de aplicación se encuentra el nodo actuador, cuya función principal es activarse únicamente cuando recibe el mensaje de confirmación de activación de la sirena. En la Figura 30, se observa el diagrama de funcionamiento del nodo actuador, desde su activación hasta emitir la alerta.

Figura 30.

Diagrama de flujo de nodo actuador.



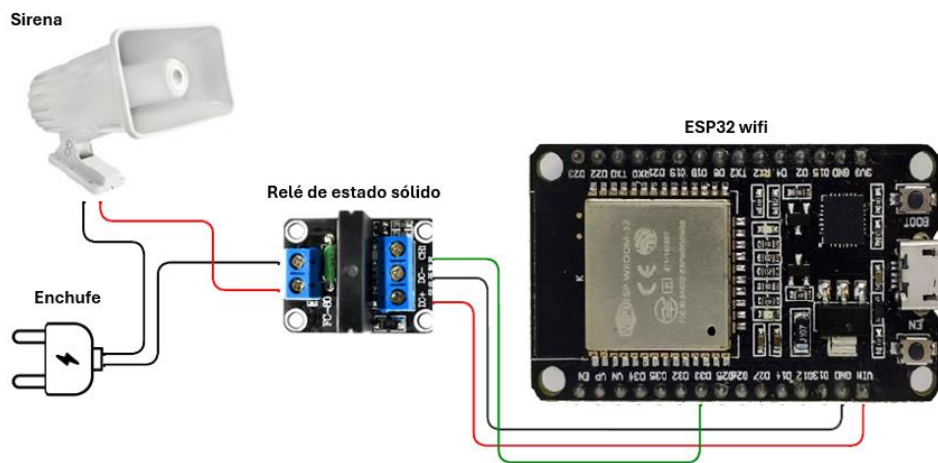
Fuente: Elaborado por el autor.

En la Figura 31, se visualiza las conexiones del nodo actuador. Este nodo contiene tres componentes como la sirena, el relé de estado sólido y el esp32. A continuación, se dará a conocer los pines de conexión.

- Voltaje – 5V (Cable tipo C)
- Voltaje de salida – 5V al pin DC+ del relé (cabe rojo)
- GND de salida – cable negro al pin DC- del relé
- Pin D33 del esp32 conectado al CH1 del relé (cable verde)
- Pin SW1 del rele conectado a la sirena (cable rojo)
- Pin GND conectado al enchufe (cable negro)
- El cable de salida de la sirena es de colo negro (actua como switch)

Figura 31.

Diagrama de conexiones de nodo actuador.



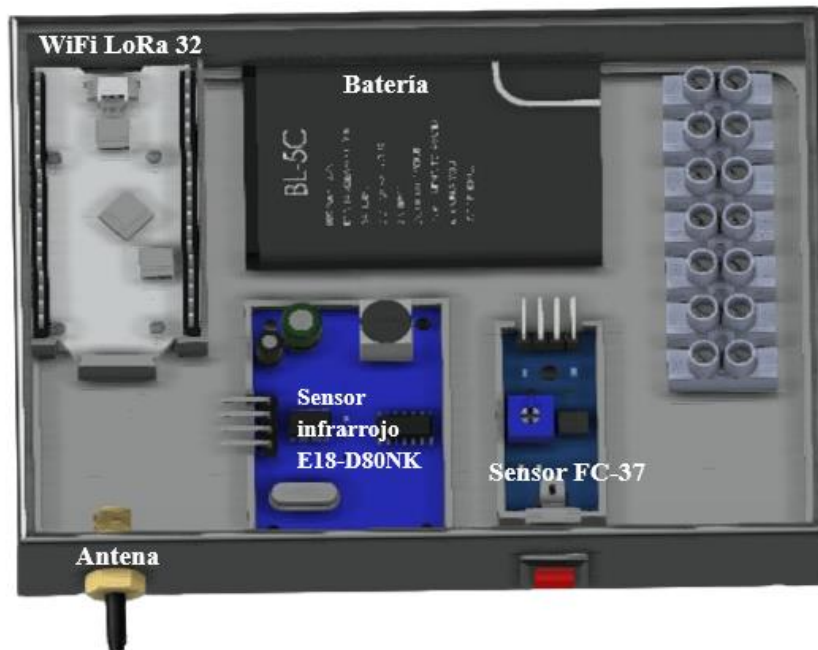
Fuente: Elaborado por el autor.

3.6.5. Diseño en 3D de Nodos del Sistema

Los componentes electrónicos de cada nodo deben ser adecuadamente ubicados dentro del diseño en 3D, asegurando un uso eficiente del espacio. Este diseño fue desarrollado dentro del software Fusion 360 desarrollado por Autodesk, es fundamental para prevenir filtraciones de agua y garantizar la resistencia frente a las variaciones climáticas en el área donde se encuentra el nodo sensor. En el **ANEXO 4** se observa desde otros ángulos el diseño de los nodos.

Figura 32.

Diseño en 3D de nodo sensor vista interna.



Fuente: Elaborado por el autor

La Figura 33 muestra una vista externa del diseño en 3D, donde el único componente ubicado externamente es el panel solar de 6V, responsable de suministrar energía al nodo sensor.

Figura 33.

Diseño en 3D de nodo sensor vista externa.



Fuente: Elaborado por el autor

Dentro de la Figura 34, se logra observar el diseño realizado para el nodo coordinador. Este diseño resulta ser un diseño básico ya que solamente se encarga de

proteger al microcontrolador y ayudar a que no afecte el rendimiento del componente electrónico.

Figura 34.

Diseño en 3D de nodo coordinador vista externa.



Fuente: Autoría.

Por otro lado, en la Figura 35 se realizó el diseño para el nodo actuador, este nodo es muy importante ya que dentro ira implementado un microcontrolador y un relé en donde son útiles para activar la sirena solamente cuando lo permita el sistema.

Figura 35.

Diseño en 3D de nodo actuador vista interna.

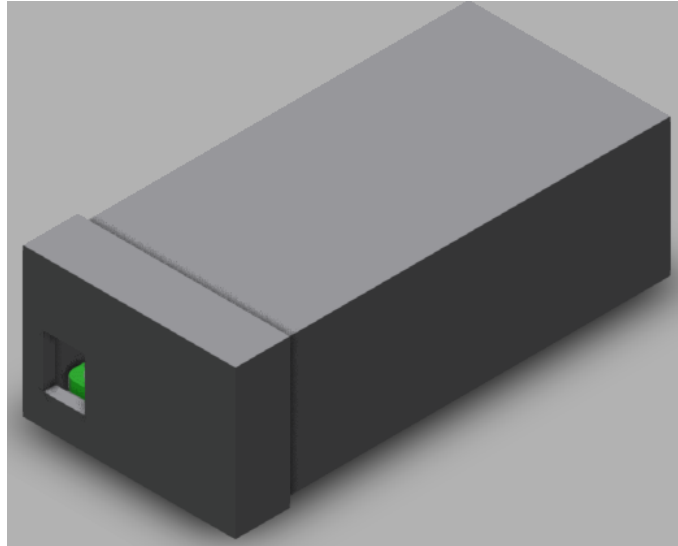


Fuente: Elaborado por el autor.

En la Figura 36 se logra observar la vista externa del diseño en 3D para el nodo actuador. Estos diseños son realizados con el fin de proteger a los componentes electrónicos y su rendimiento no se vea afectado.

Figura 36.

Diseño en 3D de nodo actuador vista externa.



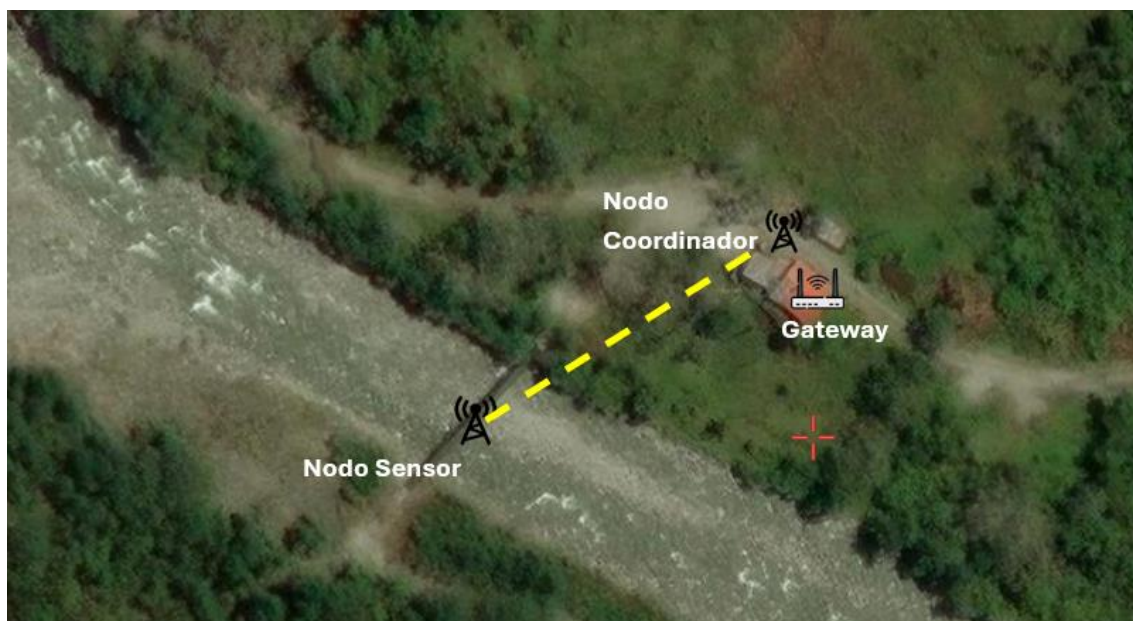
Fuente: Elaborado por el autor.

3.6.6. Diseño de red inalámbrica

En este apartado, se detalla el diseño del esquema de comunicación entre los dos nodos utilizando la tecnología LoRa, desarrollado con base en los requerimientos establecidos para el sistema. La Figura 37 ilustra el escenario donde se encuentran ubicados los nodos, así como la funcionalidad de la comunicación inalámbrica.

Figura 37.

Entorno de nodos con conexión inalámbrica.



Fuente: Elaborado por el autor.

En la Figura 37, se observa la ubicación de los nodos en donde se realizaron las pruebas de funcionamiento y recolección de datos. Inicialmente, se planificó instalar el nodo sensor a una distancia más extensa río arriba, con el fin de incrementar el tiempo para tomar datos y, en caso de que los niveles del río aumenten, el sistema debe generar las alertas tempranas e inicio a protocolos de evacuación. Sin embargo, por motivos de nuevos proyectos de agricultura en el área prevista para la instalación del nodo sensor, se optó por reubicar hacia la zona del puente. Una ventaja de la ubicación del nodo sensor es el acceso para la instalación y el funcionamiento adecuado, gracias a esto se reduce el riesgo por la humedad y la zona inestable del sitio previsto inicialmente. Por otro lado, la zona en donde se colocaron los nodos para recolectar los datos corresponde a una finca privada, mayormente por estas zonas solamente son fincas en donde no existe ningún riesgo de perder los componentes electrónicos ya que únicamente los dueños de la finca tienen acceso y tienen conocimiento del sistema.

Tabla 29.

Ubicación geográfica de Nodos

Nodo	Latitud	longitud	Altura
Nodo Sensor	-0,6633069	-77,9324761	2277
Nodo Coordinador	-0,6630682	-77,9319710	2280

Fuente: Elaborado por el autor.

Cálculo de distancia: Para calcular la distancia entre los dos nodos, se emplea la fórmula de distancia euclidiana, la cual se basa en las coordenadas geográficas especificadas en la Tabla 29.

$$d = \sqrt{[111.32 * (LaGW - LaSENS)]^2 + [111.32 * (LoGW - LoSENS)]^2 + \left(\frac{HeGW - HeSENS}{1000}\right)^2} \quad (1)$$

Para obtener el cálculo entre el nodo sensores (SENS) y nodo coordinador (GW) se utiliza la ecuación (1).

$$\begin{aligned}
 d &= \sqrt{[111.32 * (-0,6630682 - (-0,6633069))]^2 + [111.32 * (-77,9319710 - (-77,9324761))]^2 + \left(\frac{2280 - 2277}{1000}\right)^2} \\
 d &= \sqrt{(0.026572)^2 + (0.056227)^2 + (0.003)^2} \\
 d &= \sqrt{0.0007060 + 0.003161 + 0.000009} \\
 d &= \sqrt{0.003876} \\
 d &= 0.062km \\
 d &= 62 m
 \end{aligned}$$

Ángulo de azimuth. El ángulo de azimuth corresponde a la orientación de la antena en el plano horizontal, tomando como referencia un meridiano, que generalmente es el norte. Para determinar este ángulo, se emplean las ecuaciones (2) y (3).

$$azimut = \tan^{-1}(\theta) \quad (2)$$

$$\theta = \frac{\Delta longitud}{\Delta latitud} \quad (3)$$

Donde θ ; representa la diferencia entre los cambios de longitud y latitud entre los puntos, tal como se describe en la ecuación (3).

$$\theta = \frac{LoSENS-LoGW}{LaSENS-LaGW} \quad (4)$$

Cálculo del ángulo de azimut entre Nodo Sensor y Nodo coordinador. Mediante la ecuación (4), se realiza el cálculo del ángulo de azimut entre el nodo sensor (SENS) y nodo coordinador (GW).

$$\begin{aligned} \theta &= \frac{-77.9324761 + 77.9319710}{-0.6633069 + 0.6630682} \\ azimuth &= \tan^{-1}(2.1160) \\ azimuth &= 64.7050^\circ \end{aligned}$$

Ángulo de elevación. El ángulo de elevación corresponde al ángulo que se forma entre la trayectoria de una onda emitida por una antena y el plano horizontal. Para determinar este ángulo, se utilizan las ecuaciones (5) y (6).

$$elevación = \tan^{-1}(a) \quad (5)$$

$$\alpha = \frac{AltGW-AltSENS}{d} \quad (6)$$

Donde α : representa la variación de altura y distancia entre los puntos de cada nodo.

Cálculo de ángulo de elevación entre Nodo sensor y nodo coordinador. A continuación, se lleva a cabo el cálculo matemático del ángulo de elevación entre el nodo sensor (SENS) y el nodo coordinador (GW).

$$\alpha = \frac{(2280 - 2277)}{62} [m]$$

$$\alpha = 0.04838$$

$$elevación = \tan^{-1}(0.04838)$$

$$elevación = 2.77^\circ$$

Cálculo de Atenuación en el espacio libre. La atenuación en el espacio libre se calcula a partir de la relación entre la potencia transmitida y la disminución de potencia que experimenta la señal al ser recibida, tal como se describe en la ecuación (7).

$$L_{bf}|_{dB} = 32.5 + 20\log [f(MHz)] + 20\log [d(km)] \quad (7)$$

Cálculo de Atenuación entre nodo coordinador y nodo sensor. A partir de la ecuación (7), se determina la atenuación en la comunicación entre el nodo coordinador (GW) y el nodo sensor (SENS). Estos cálculos se llevan a cabo considerando una frecuencia de 915 MHz.

$$L_{bf} = 32.5 + 20 \log(915) + 20\log (0.062)$$

$$L_{bf} = 67.57 \text{ dB}$$

Cálculo de Atenuación por pérdidas de vegetación. A través de la ecuación (8), se determina la pérdida o atenuación causada por la vegetación, representada como $L_v(\text{dB})$, de acuerdo con la recomendación UIT-R P.833. Este cálculo aplica a las bandas de frecuencia UHF, que abarcan el rango de 300 MHz a 3 GHz.

$$L_{v-ITU}(\text{dB}) = 0.2f^{0.3}d^{0.2} \quad (8)$$

$$L_t = l_{bf}(\text{dB}) + l_v(\text{dB}) \quad (9)$$

Cálculo de Atenuación nodo sensor y nodo coordinador. Utilizando la ecuación (9), se lleva a cabo el cálculo de la atenuación entre el nodo coordinador y el nodo sensor.

$$L_{v-ITU}(dB) = 0.2f^{0.3}d^{0.2}$$

$$L_{v-ITU}(dB) = 0.2(915)^{0.3}(62)^{0.2}$$

$$L_{v-ITU}(dB) = 3.53dB \Rightarrow \text{atenuación adicional por vegetación}$$

$$L_t = l_{bf}(dB) + l_v(dB)$$

$$L_t = 67.57 + 3.53$$

$$L_t = 71.1 \text{ dB}$$

Cálculo de potencia de Rx. Para determinar la potencia de recepción, se lleva a cabo un análisis técnico de las especificaciones del módulo WiFi LoRa 32. Además, se aplica la ecuación (10) para efectuar el cálculo matemático correspondiente.

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_t \quad (10)$$

Donde:

- Potencia de transmisión: 21 [dBm]
- Ganancia de antena de transmisión: 3 [dBi]
- Ganancia de antena de recepción: 3 [dBi]
- Frecuencia de operación: 915 [MHz]

Cálculo de recepción nodo coordinador y nodo sensor. A partir de la ecuación 10, se determina la potencia de recepción entre el nodo coordinador y nodo sensor, a una distancia de 62 m.

$$P_{RX}(dBm) = G_{TX}(dBi) + G_{RX}(dBi) - L_t(dB)$$

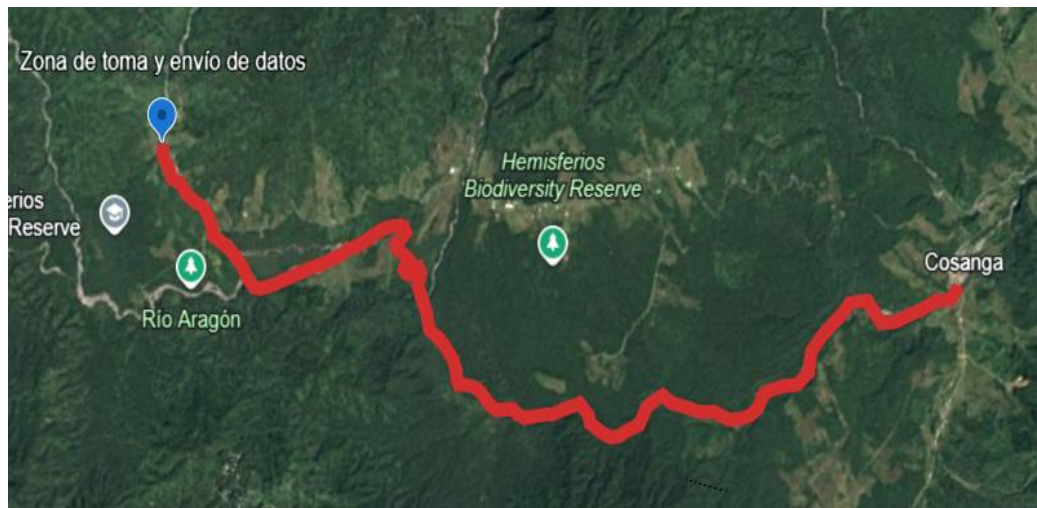
$$P_{RX}(dBm) = 21 + 3 + 3 - 71.1$$

$$P_{RX}(dBm) = -44.1 \text{ dBm}$$

En la Figura 38, se utiliza la herramienta de Google Earth para determinar con precisión la distancia entre la ubicación de los nodos y la parroquia de Cosanga, la cual es de aproximadamente 16.45 km. Este parámetro fue clave para calcular el tiempo estimado que tomaría un posible desbordamiento en alcanzar la parroquia. Además, esta información es fundamental para activar las alertas y notificar oportunamente a los habitantes, permitiéndoles evacuar de manera segura y a tiempo.

Figura 38.

Distancia de Nodos hacia la parroquia de Cosanga.



Fuente: Google Earth

3.6.6.1. Simulación de Red Inalámbrica.

El software Radio Mobile permite simular perfiles geográficos del terreno y combinarlos con los parámetros técnicos del protocolo de comunicación. Entre estos parámetros se incluyen las pérdidas de transmisión, la potencia de transmisión, el ángulo de elevación, distancia, el azimut y las características de las antenas, facilitando así el análisis del enlace de comunicación. Dentro del rango de frecuencia establecido se

encuentra 915 MHz ya que es la frecuencia central para estaciones móviles que recomienda la ARCOTEL. (ARCOTEL, 2017)

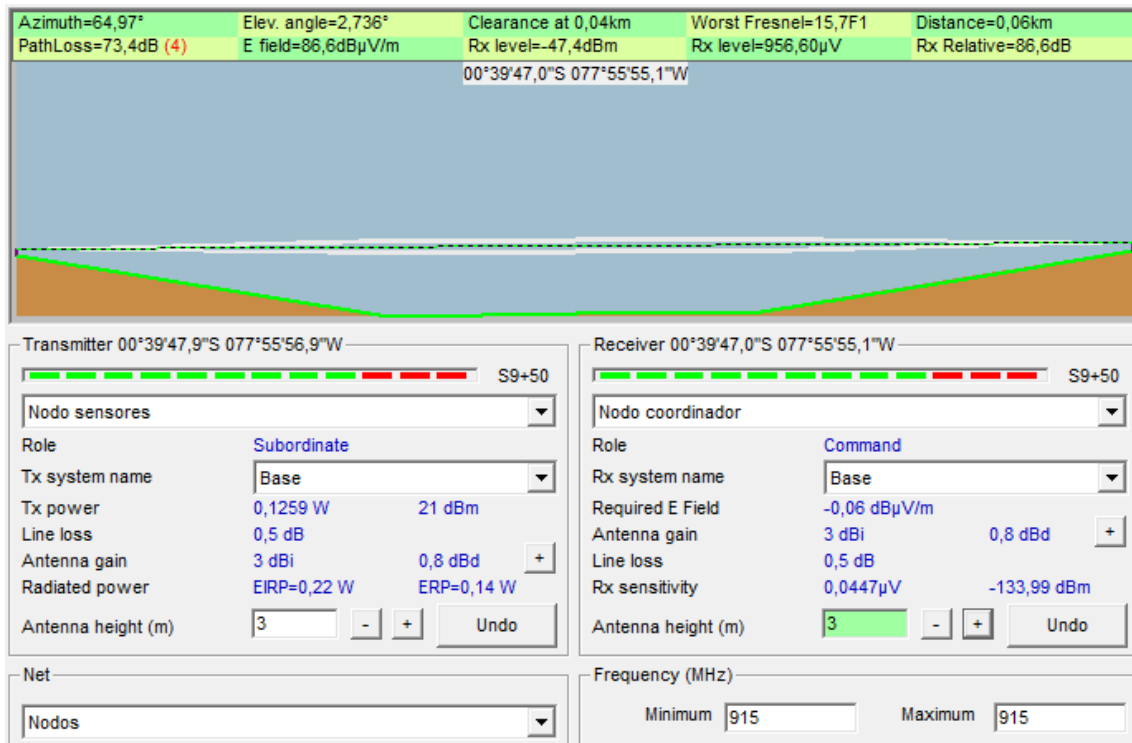
Parámetros de configuración de WiFi LoRa 32 V3. La configuración de los parámetros de red al utilizar módulos WiFi LoRa 32 es fundamental para establecer una comunicación inalámbrica eficiente y confiable. Estos parámetros determinan el alcance, la estabilidad y el rendimiento del enlace, optimizando la transmisión de datos en aplicaciones IoT y redes de largo alcance.

- Frecuencia de operación: 915 [MHz]
- Polarización: Horizontal
- Potencia de transmisión: 21 [dBm]
- Umbral de recepción: -134 [dBm]
- Ganancia de la antena: 3 [dBi]

Resultados de simulación. A continuación, se observan los resultados obtenidos mediante la simulación en el software Radio Mobile, donde se especifican aspectos como la distancia entre los nodos, los ángulos de elevación y azimut, así como las pérdidas en el espacio libre, entre otros parámetros, tal como se muestra en la Figura 39. Todo el proceso de la simulación se encuentra detallado en **ANEXO 5**.

Figura 39.

Resultado de simulación de nodo sensores y nodo coordinador.

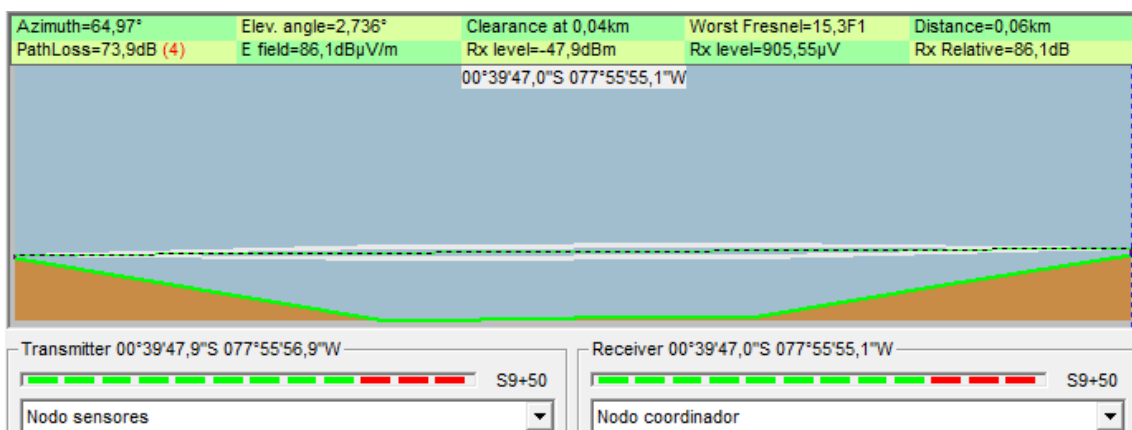


Fuente: Adaptado al simulador Radio Mobile

Zona geográfica del terreno. Los resultados de la zona geográfica de la simulación de la red entre el nodo sensores y nodo coordinador. Se visualiza en la Figura 39, en la que se muestra entre los dos nodos una distancia aproximada de 62m.

Figura 40.

Zona geográfica de transmisión entre nodo sensor y nodo coordinador.



Fuente: Adaptado al simulador Radio Mobile

3.6.6.2. Análisis comparativo de resultados

Dentro de la Tabla 30 se presentan varios parámetros que intervienen en el diseño de la red inalámbrica. Los valores mostrados fueron determinados mediante procedimientos matemáticos y simulaciones llevadas a cabo utilizando el software Radio Mobile. Al realizar la comparación entre ambos conjuntos de resultados, se observa que la atenuación calculada por Radio Mobile supera a la obtenida por cálculos manuales, con una diferencia aproximada de 2 dB en pérdidas.

Tabla 30.

Comparación de resultados calculados y simulados.

Parámetros	Nodo Sensor – Nodo Coordinador	
	Cálculo manual	Radio Mobile
Distancia Nodos	0.062 [km]	0.06 [km]
Azimuth	64.7050°	64.97°
Angulo de elevación	2.77°	2.736°
Atenuación [Hz]	71.10 [dB]	73.4 [dB]
Nivel Rx [dBm]	-44.1 [dBm]	-47.4 [dBm]

Fuente: Elaborado por el autor.

Los resultados obtenidos de manera manual y mediante la simulación en Radio Mobile fueron analizados entre los dos nodos. Los datos tienen un margen similar, tomando en cuenta que pueden existir factores como relieves o vegetación entre los dos nodos, estos son factores tomados dentro de la simulación, pero no dentro de los cálculos manuales. Los valores más destacados dentro de los resultados son la atenuación y el nivel de recepción. La atenuación tiene una diferencia entre cálculo y simulación de 2.3 dB, indicando que se encuentra dentro de un margen estable. Por otro lado, el nivel de señal recibida tiene una diferencia de 3.3 dBm, indicando una variación considerable, tomando en cuenta que se mantiene en un rango adecuado para garantizar el envío de datos.

A pesar de las variaciones de los resultados, esto indica que existe un enlace adecuado con una señal de recepción que se encuentra dentro del rango adecuado, estos datos sirven para validar la funcionalidad del sistema de monitoreo.

3.6.6.3. *Indicador de intensidad de señal recibida (RSSI)*

El indicador de intensidad de señal recibida es un valor negativo que se expresa en dBm. Este indicador refleja la potencia de la señal en función de la distancia entre dos nodos. El RSSI indica que, cuanto más sea un valor a cero, mayor será la potencia y una estabilidad de la señal, por otro lado, si existen valores altos negativos señala que la señal será débil.(Gordillo, 2021)

- Si RSSI = -30dBm: señal fuerte.
- Si RSSI= -120 dBm: señal débil.

Dentro de la Tabla 31, se presentan los valores de distancia, potencia recibida (dBm) y el indicador si recibe señales o no. Basándose en el datasheet del módulo Heltec WiFi LoRa V3, la sensibilidad mínima de la antena de recepción es de -134 dBm. Las distancias evaluadas van desde los 5m hasta los 70m.

Tabla 31.

Relación en función de distancia y RSSI.

Distancia (m)	RSSI (dBm)	Recepción
5	-91	Si
10	-104	Si

15	-107	Si
20	-109	Si
30	-111	Si
50	-114	Si
60	-117	Si
70	-119	Si

Fuente: Autoría.

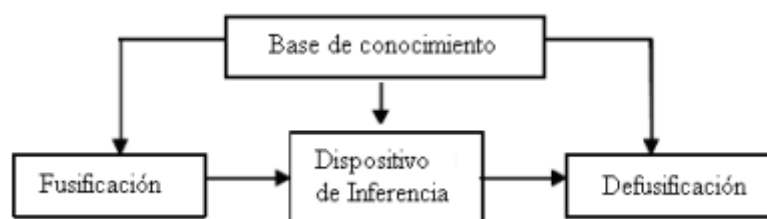
A partir de los valores de RSSI obtenidos en función de la distancia, se observa que la intensidad de la señal va disminuyendo a medida que se aumenta la distancia entre los dos nodos. Sin embargo, la comunicación sigue estable ya que se mantiene dentro del rango de potencia de la señal. Por otro lado, si la distancia supera el límite de recepción de la señal, pueden ocurrir pérdidas de paquetes y la comunicación inalámbrica se verá afectada.

3.6.7. Diseño de operación con lógica difusa

La arquitectura del sistema difuso basado en el modelo de Mamdani consta de tres componentes principales: la fuzzificación, la base de conocimiento con su sistema de inferencia y la defuzzificación, como se logra observar en la Figura 41.(Díaz et al., 2024) (L. Salazar et al., 2020)

Figura 41.

Arquitectura del sistema difuso.

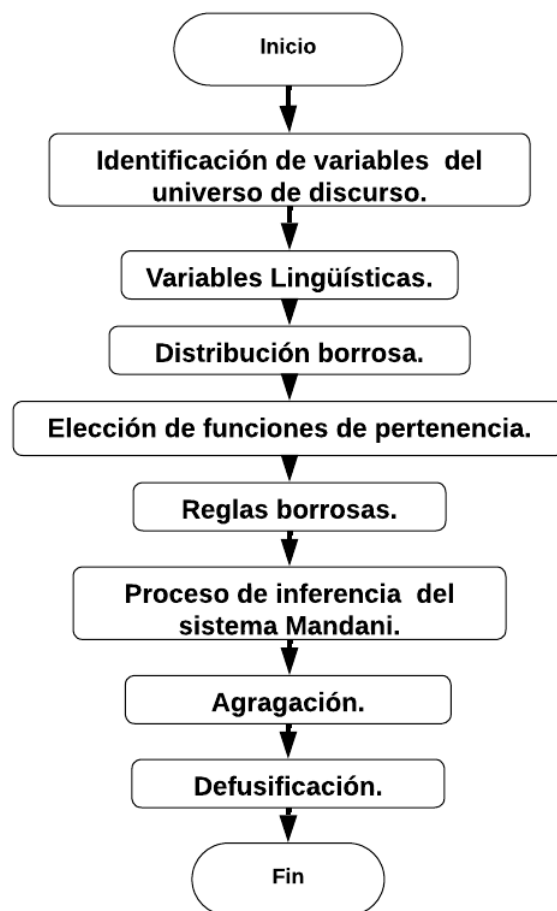


Tomado de: (Carvajal et al., 2010)

A continuación, para la implementación del sistema de control, se define un diagrama de flujo que establece los pasos adecuados para la configuración de variables y parámetros, tal como se muestra en la Figura 42.

Figura 42.

Diagrama de flujo del diseño de operación difusa.



Fuente: Elaborado por el autor.

3.6.7.1. *Identificación de variable del Universo de discurso*

Para definir un universo de discurso, es necesario establecer las variables del sistema, especificando de manera clara los conjuntos de entrada y salida que se interrelacionan mediante un controlador difuso.

Tabla 32.

Variables de Universo de discurso

Tipo de conjunto	Conjunto
Entrada	Nivel del agua en el río Nivel de precipitación Velocidad del flujo del río
Salida	Grado de riesgo

Fuente: Elaborado por el autor

En la Tabla 32, se presentan los conjuntos de entrada y salida junto con sus variables correspondientes. La variable que indica el nivel del agua en el río corresponde al sensor JSN-SR04T, el cual se encarga de medir la altura del agua. En esta área, los niveles de agua oscilan entre 1 y 4 metros.

De manera análoga, la variable de precipitación está asociada al sensor FC-37, el cual mide la cantidad de lluvia con valores analógicos. Los valores medidos van desde 0 cuando la placa está totalmente húmeda, a 4095 cuando la placa está totalmente seca.

Por último, la variable velocidad del flujo del río se mide con un sensor infrarrojo, que determina la rapidez del flujo del agua en m/s. Finalmente, el conjunto de salida obtiene la variable de estado, se determina mediante lógica difusa y puede tomar valores como bajo, medio y alto, dependiendo de las condiciones de las variables de entrada.

3.6.7.2. *Variables Lingüísticas*

Las variables lingüísticas se determinan en base a los conjuntos de entrada y salida con un rango específico. Este proceso se desarrolla a partir de las experiencias obtenidas en el proceso del desarrollo del sistema.

$$N = [0, 4] \Rightarrow \{x \in N\} \Rightarrow cm$$

$$P = [0, 1024] \Rightarrow \{x \in P\} \Rightarrow Valor Analógico (V)$$

$$V = [0, 100] \Rightarrow \{x \in V\} \Rightarrow m/s$$

$$R = [0, 10] \Rightarrow \{x \in R\} \Rightarrow Grado de riesgo$$

Donde:

N [Nivel de agua en el río]; P [Precipitación]; V [Velocidad de flujo del río]; R [Grado de riesgo]

Cada una de estas magnitudes corresponde a las variables de los sensores del sistema de monitoreo, estableciendo los límites de operación según la función de pertenencia.

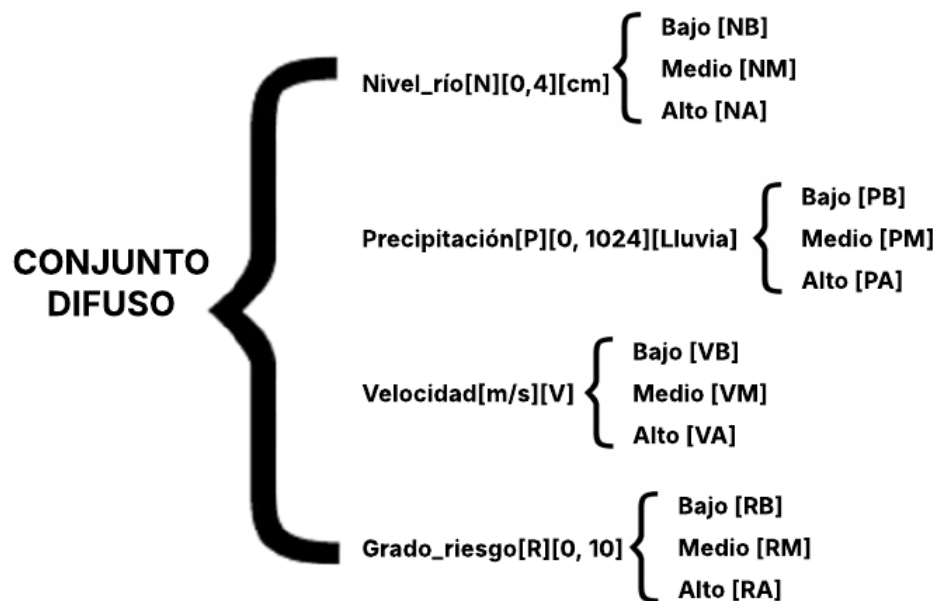
3.6.7.3. *Distribución Borrosa*

A cada variable se le asignan distintos términos lingüísticos, que se definen como subconjuntos dentro de un universo de discurso. Cada uno de estos términos tiene un nombre y una estructura sintáctica. En la Figura 43, se muestra el conjunto difuso compuesto por tres subconjuntos para el nivel del río, tres para la precipitación, tres para la velocidad del río y tres para el grado de riesgo.

Cada una de estas características se representa mediante valores aproximados e imprecisos, lo que permite su análisis mediante lógica difusa.

Figura 43.

Distribución borrosa del sistema de monitoreo



Fuente: Elaborado por el autor

3.6.7.4. Elección de función de pertenencia

Cada término lingüístico está asociado a un conjunto difuso, definido por una función de pertenencia que incluye su expresión matemática y representación gráfica. La Tabla 33, muestra los conjuntos de funciones de pertenencia del sistema de monitoreo, indicando los términos lingüísticos y las funciones de pertenencia, como la trapezoidal y la triangular, junto con sus valores dentro del universo de discurso.

Tabla 33.

Conjuntos de funciones de pertenencia del sistema.

Termino Lingüístico	Función de membresía	Universo de discurso
Nivel del río		
NB	Trapezoidal	[0, 100, 200]

NM	Triangular	[200, 300, 400]
NA	Trapezoidal	[300, 350, 450]
Precipitación		
PB	Trapezoidal	[0, 500, 1500]
PM	Triangular	[1500, 2500, 3500]
PA	Trapezoidal	[2500, 3500, 4095]
Velocidad		
VB	Trapezoidal	[0, 5, 20]
VM	Triangular	[20, 30, 40]
VA	Trapezoidal	[30, 35, 100]
Grado de Riesgo		
RB	Trapezoidal	[0, 2.5, 3.5]
RM	Triangular	[3.5, 6, 7.5]
RA	Trapezoidal	[7.5, 10]

Fuente: Elaborado por el autor.

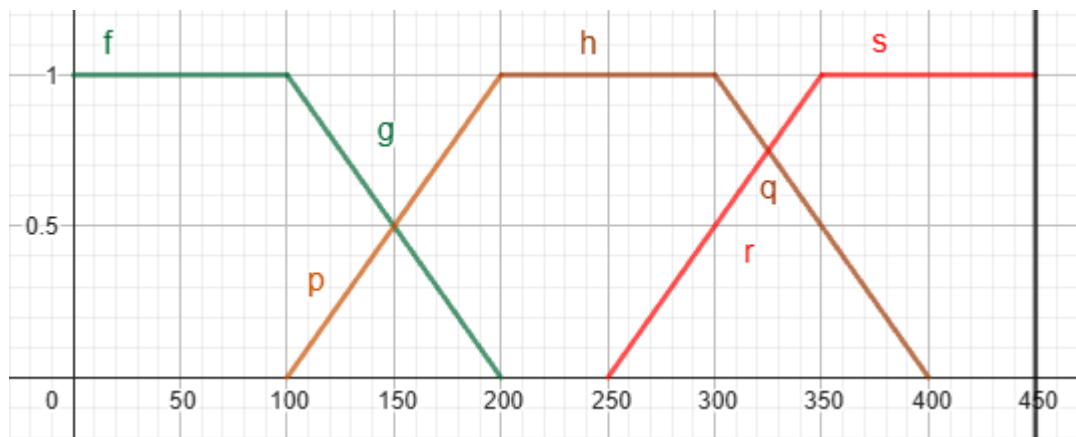
En base al apartado **2.3.4.5**, se consideran diversas funciones de pertenencia, destacando que, para el diseño de este sistema de control difuso, se han utilizado las funciones de membresía trapezoidal y triangular. Con el uso del software matemático Geogebra se realizó los cálculos de cada variable de entrada y sus respectivas gráficas.

Conjunto de nivel de río. Es fundamental evaluar el nivel de agua en el río y definir el ámbito de estudio que será representado mediante expresiones matemáticas, como las funciones trapezoidal y triangular.

$$Nivel_río[0, 450][cm] \left\{ \begin{array}{l} Baja[PB]; [0, 100]; \{1\} si (0 \leq x \leq 100) \\ [100, 200] \left\{ \frac{200 - x}{100} \right\} si (100 < x \leq 200) \\ Media[PM][100, 200] \left\{ \frac{x - 100}{100} \right\} si (100 < x \leq 200) \\ [200, 300]; \{1\} si (200 \leq x \leq 300) \\ [300, 400] \left\{ \frac{-x + 400}{100} \right\} si (300 < x \leq 400) \\ Alta[PA][250, 350] \left\{ \frac{x - 250}{100} \right\} si (250 < x \leq 350) \\ [350, 450] \{1\} si (350 < x < 450) \end{array} \right.$$

Figura 44.

Gráfica de la función de pertenencia de nivel de río.

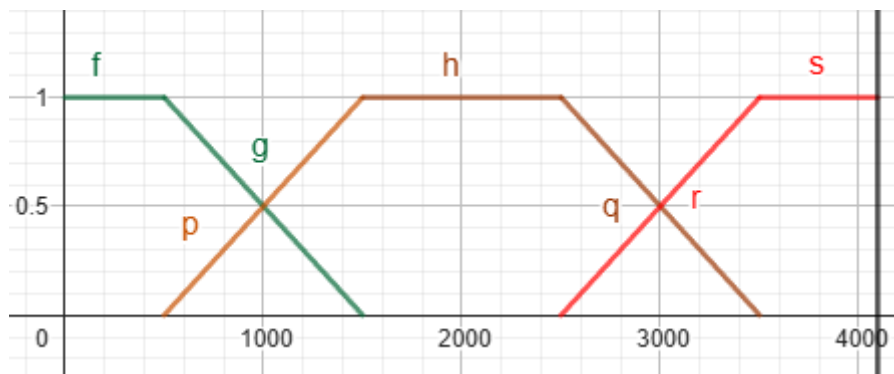


Conjunto de precipitación. De manera similar, se determinan los posibles niveles de la variable de precipitación y se define un universo de discurso mediante la función de pertenencia, utilizando las siguientes expresiones matemáticas.

$$Precipitación[0, 4095][P] \left\{ \begin{array}{l} Baja[PB]; [0, 500]; \{1\} si (0 \leq x \leq 500) \\ [500, 1500] \left\{ \frac{1500 - x}{1000} \right\} si (500 < x \leq 1500) \\ Media[PM][500, 1500] \left\{ \frac{x - 500}{1000} \right\} si (500 < x \leq 1500) \\ [1500, 2500]; \{1\} si (1500 < x \leq 2500) \\ [2500, 3500] \left\{ \frac{-x + 3500}{1000} \right\} si (2500 < x \leq 3500) \\ Alta[PA][2500, 3500] \left\{ \frac{x - 2500}{1000} \right\} si (2500 < x \leq 3500) \\ [3500, 4095] \{1\} si (3500 < x \leq 4095) \end{array} \right.$$

Figura 45.

Gráfica de la función de pertenencia de precipitación.

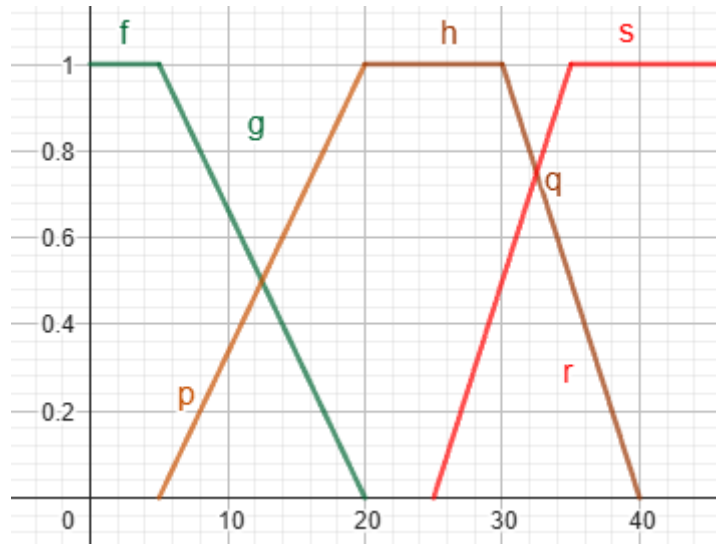


Conjunto de velocidad. De forma similar, se establecen los valores que puede asumir la variable de velocidad y se define el universo de discurso mediante las ecuaciones trapezoidal y triangular, junto con sus respectivas variables lingüísticas.

$$Velocidad[0, 100][rpm] \left\{ \begin{array}{l} Baja[PB]; [0, 5]; \{1\} si (0 \leq x \leq 5) \\ [5, 20] \left\{ \frac{20-x}{15} \right\} si (5 < x \leq 20) \\ Media[PM][5, 20] \left\{ \frac{x-5}{15} \right\} si (5 < x \leq 20) \\ [20, 30]; \{1\} si (20 \leq x \leq 30) \\ [30, 40] \left\{ \frac{-x+40}{10} \right\} si (30 < x \leq 40) \\ Alta[PA][25, 35] \left\{ \frac{x-25}{10} \right\} si (25 < x \leq 35) \\ [35, 100] \{1\} si (35 < x < 100) \end{array} \right.$$

Figura 46.

Gráfica de la función de pertenencia de velocidad.



Conjunto de grado de riesgo. En la posición final se encuentra el conjunto que representa el grado de riesgo, donde se definen los valores correspondientes a cada una de las variables lingüísticas, utilizando tanto funciones trapezoidales como triangulares.

$$Grado_riesgo[0, 10][R] \left\{ \begin{array}{l} Baja[RB]; [0, 2.5]; \{1\} si (0 \leq x \leq 2.5) \\ [2.5, 3.5] \left\{ \frac{3.5 - x}{1} \right\} si (2.5 < x \leq 3.5) \\ Media[RM][2.5, 3.5] \left\{ \frac{x - 2.5}{1} \right\} si (2.5 < x \leq 3.5) \\ [3.5, 6]; \{1\} si (3.5 < x \leq 6) \\ [6, 7.5] \left\{ \frac{-x + 7.5}{1.5} \right\} si (6 < x \leq 7.5) \\ Alta[RA][6, 7.5] \left\{ \frac{x - 6}{1.5} \right\} si (6 < x \leq 7.5) \\ [7.5, 10] \{1\} si (7.5 < x \leq 10) \end{array} \right.$$

Figura 47.

Gráfica de la función de pertenencia de grado de riesgo.



3.6.7.5. Reglas Borrosas

En esta fase del diseño, se emplea el método de Ebrahim Mandani para establecer las reglas basadas en todas las combinaciones posibles de los subconjuntos difusos, considerando las entradas de los términos lingüísticos y las salidas de las reglas de control. Esto da como resultado un total de 27 reglas difusas, como se muestra en la Tabla 34. Cada una de estas reglas se compone de conjuntos de pertenencia y utiliza el operador lógico AND de Zadeh, conocido como intersección.

La selección de 27 reglas difusas se basa en la definición de los subconjuntos de pertenencia. Para el subconjunto del nivel del río, se consideran 3 variables lingüísticas, con un universo de discurso que va de 0 a 450 cm. De manera similar, para el conjunto

de precipitación, se establecen 3 variables lingüísticas, con un universo de discurso que abarca valores analógicos de 0 a 4095. Finalmente, en el subconjunto de la velocidad del caudal del río, se definen 3 variables lingüísticas, con un universo de discurso que varía de 0 a 100 m/s.

Tabla 34.

Conjuntos de funciones de pertenencia del sistema.

#	REGLAS DIFUSAS									
1	If	NB	and	PB	and	VB	entonces	R	es	RB
2	If	NB	and	PB	and	VM	entonces	R	es	RB
3	If	NB	and	PB	and	VA	entonces	R	es	RB
4	If	NB	and	PM	and	VB	entonces	R	es	RB
5	If	NB	and	PM	and	VM	entonces	R	es	RB
6	If	NB	and	PM	and	VA	entonces	R	es	RM
7	If	NB	and	PA	and	VB	entonces	R	es	RB
8	If	NB	and	PA	and	VM	entonces	R	es	RM
9	If	NB	and	PA	and	VA	entonces	R	es	RM
10	If	NM	and	PB	and	VB	entonces	R	es	RB
11	If	NM	and	PB	and	VM	entonces	R	es	RB
12	If	NM	and	PB	and	VA	entonces	R	es	RM
13	If	NM	and	PM	and	VB	entonces	R	es	RB
14	If	NM	and	PM	and	VM	entonces	R	es	RM
15	If	NM	and	PM	and	VA	entonces	R	es	RA
16	If	NM	and	PA	and	VB	entonces	R	es	RM
17	If	NM	and	PA	and	VM	entonces	R	es	RA

18	If	NM	and	PA	and	VA	entonces	R	es	RA
19	If	NA	and	PB	and	VB	entonces	R	es	RM
20	If	NA	and	PB	and	VM	entonces	R	es	RM
21	If	NA	and	PB	and	VA	entonces	R	es	RA
22	If	NA	and	PM	and	VB	entonces	R	es	RM
23	If	NA	and	PM	and	VM	entonces	R	es	RA
24	If	NA	and	PM	and	VA	entonces	R	es	RA
25	If	NA	and	PA	and	VB	entonces	R	es	RA
26	If	NA	and	PA	and	VM	entonces	R	es	RA
27	If	NA	and	PA	and	VA	entonces	R	es	RA

Fuente: Elaborado por el autor

3.6.7.6. *Procesos de Inferencia del Sistema Mandani*

El proceso de inferencia basado en el método de Mandani se basa en las reglas presentadas en la Tabla 34, con el objetivo de derivar las conclusiones de las variables lingüísticas de salida a partir de las variables de entrada. A cada una de las 27 reglas se le aplica la ecuación (18), que se describe a continuación.

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

La resolución de este sistema de control se realiza considerando los tres subconjuntos de entrada, utilizando la propiedad distributiva, tal como se expresa en la siguiente ecuación.

$$N \cup (P \cap V) = (N \cap P) \cup (P \cap V)$$

$$\mu_{N \cap P \cap V}(x) = \{\min[\mu_N(x), \mu_P(x)], \min[\mu_P(x), \mu_V(x)]\}$$

3.6.7.7. Agregación

Las reglas establecidas a partir del conocimiento técnico y la experiencia de un experto en piscicultura son fundamentales para guiar la toma de decisiones a través de las reglas del controlador difuso (consulte 3.6.7.5). Durante este proceso de agregación, se combinan todas las reglas del conjunto difuso para generar una única acción de salida como un conjunto difuso consolidado. Este procedimiento se realiza una vez para cada una de las etiquetas correspondientes a la variable de salida.

3.6.7.8. Defusificación

En la etapa final, cuyo propósito es convertir el conjunto difuso generado en la fase de agregación en un valor de salida concreto y aplicable al sistema de monitoreo. Para ello, se emplea el método del centroide, que transforma la salida difusa en un valor numérico real. En este paso, se determina la salida del sistema de monitoreo, la cual informa al usuario sobre el estado del río, utilizando las funciones de membresía y las variables lingüísticas correspondientes.

4. CAPITULO IV: Implementación y Pruebas de funcionamiento

En este capítulo se describe la implementación del proyecto actual y se analiza la eficiencia del sistema mediante diversas pruebas. Primero, se llevan a cabo pruebas individuales en cada uno de los bloques, tanto de hardware como de software. Por otro lado, se ensambla el hardware dentro del diseño realizado en 3D para cumplir con los requisitos del sistema en cuanto a protección contra daños en condiciones climáticas adversas, seguido de una prueba para verificar su funcionamiento adecuado. Por otro lado, los nodos fueron implementados en las zonas establecidas dentro del capítulo III apartado de diseño y finalmente el sistema de monitoreo fue implementado dentro de la Junta Parroquial de Cosanga, cantón Quijos.

4.1. Integración de Componentes

La implementación del sistema de monitoreo se llevó a cabo integrando componentes electrónicos clave, como el sensor ultrasónico JSN-SR04T para la medición del nivel del agua, el sensor de precipitación FC-37, y un sensor infrarrojo E18-D80NK para determinar la velocidad del flujo del río Cosanga, también los microcontroladores de cada nodo. Basándose en las dimensiones y especificaciones técnicas de cada componente, se diseñó un modelo en 3D que se detalla en el **ANEXO 4**. Posteriormente, dichos diseños fueron materializados en una implementación física, asegurando la correcta integración de los sensores, actuador y su funcionalidad dentro del sistema de monitoreo.

4.1.1. Armado de Nodos

De acuerdo con las medidas del diseño en 3D que se muestra en la Figura 32, se decidió por colocar todos los elementos dentro del Case cuyas medidas son 12.3x9.3x3.5 cm, las características del material de construcción de este Case son de alta resistencia,

otra de las ventajas de la caja es que proporciona un excelente sellado para impedir el ingreso de humedad y sea resistente a la lluvia. Estas características son ideales para que el sistema de monitoreo funcione adecuadamente. En la Figura 48, se puede apreciar los dispositivos conectados correctamente.

Figura 48.

Armado del bloque de recolección de datos.

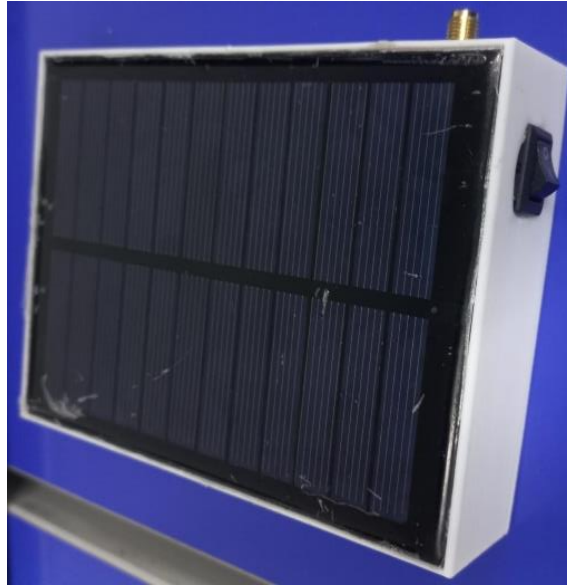


Fuente: Autoría.

Una vez armados y conectados los elementos que del nodo sensor, se procede a sellar la caja por completo para prevenir la filtración de humedad, agua, con el fin de evitar daños dentro de los módulos electrónicos. En la Figura 49 se muestra el armado final del nodo sensor.

Figura 49.

Armado final de Nodo Sensor.



Fuente: Autoría.

Por otro lado, en la Figura 50 se visualiza el nodo coordinador, al que también se le diseñó un Case con la finalidad de hacerlo resistente ante la humedad y evitar la filtración de la lluvia.

Figura 50.

Armado final de Nodo Coordinador.

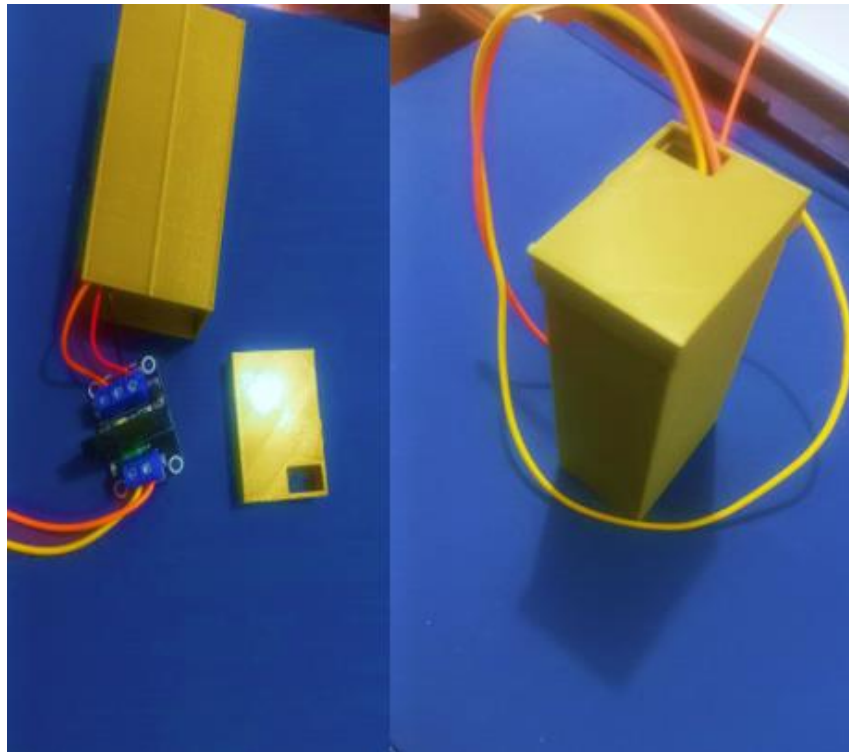


Fuente: Autoría.

Finalmente, en la Figura 51, se logra observar el armado del nodo actuador. El objetivo del diseño es alargar el tiempo de rendimiento de los componentes electrónicos y su funcionalidad.

Figura 51.

Armado final de Nodo Actuador.



Fuente: Autoría.

4.2. Verificación de configuración de elementos del sistema

A continuación, se procede a verificar las configuraciones realizadas dentro de cada uno de los nodos del sistema, se visualizará las partes más importantes que forman el sistema.

4.2.1. Verificación de configuración Nodo Sensor.

Para establecer la configuración con el nodo sensor, se procede a cargar todo el código establecido dentro del IDE de Arduino al microcontrolador WiFi LoRa 32 mediante el puerto COM designado.

A continuación, dentro de la Figura 52 se verifica el proceso de la configuración para establecer la comunicación inalámbrica. Dentro de esta parte del código se representa

la inicialización del nodo sensor, permitiendo estar listo para transmitir o recibir información del nodo coordinador. De igual manera, se definen los parámetros como la frecuencia, el ancho de banda y spreading factor. Estos parámetros son los que se manejarán en los dos nodos al momento de realizar el enlace de comunicación y empezar a intercambiar datos.

Figura 52.

Parámetros de inicialización de Cx inalámbrica de nodo sensor.

```
#include <heltec_unofficial.h>
#include <WiFi.h>
#include <ArduinoOTA.h>
#include <ESPAsyncWebServer.h>
#include <SPIFFS.h>
#include <ArduinoJson.h>

#include "fuzzy.h"

#define VBAT_Read    1
#define ADC_Ctrl     37

AsyncWebServer server(80);

#define FREQUENCY 915
#define BANDWIDTH 125.0
#define SPREADING_FACTOR 12
```

Librerías Implementadas

Acceso al sitio web

Parámetros de Cx LoRa

Fuente: Elaborado por el autor.

En la Figura 53, se puede visualizar la configuración para lograr enviar los datos hacia el nodo coordinador mediante la tecnología LoRa. Dentro del cuadro de color rojo se logra visualizar la configuración para transmitir los datos tomados desde el nodo sensor hacia el nodo coordinador. Por otro lado, dentro del cuadro de color verde se visualiza la configuración para recibir información del nodo coordinador, estos mensajes recibidos pueden ser comandos o confirmación de enlace entre nodos.

Figura 53.

Configuración de transmisión y recepción de datos en nodo sensor.

```
//----- FUNCIONES LORA TX -----  
void LoRaTx(String mensaje) {  
    if (!isTransmitting) {  
        radio.clearDio1Action();  
        isTransmitting = true;  
  
        int status = radio.transmit(mensaje.c_str());  
        if (status == RADIOLIB_ERR_NONE) {  
            display.printf("Tx: %s\n", mensaje.c_str());  
        } else {  
            display.printf("Error de transmisión: %d\n", status);  
        }  
        isTransmitting = false;  
    }  
    radio.setDio1Action(onReceive);  
    radio.startReceive();  
}  
  
//----- FUNCIONES LORA RX -----  
String LoRaRx() {  
    if (rxFlag && isTransmitting == false) {  
        rxFlag = false;  
        String rxdata;  
        int status = radio.readData(rxdata);  
  
        if (status == RADIOLIB_ERR_NONE) {  
            return rxdata;  
        } else {  
            display.printf("Error al recibir: %d\n", status);  
        }  
    }  
    return "";  
}
```

Parámetro para Trasmistir datos mediante LoRa

Envío y verificación del mensaje

Recepción de mensajes del nodo coordinador
// Reiniciar bandera de recepción
// Buffer para datos recibidos
// Leer datos

Fuente: Elaborado por el autor.

En la Figura 54, se visualiza el proceso de lógica difusa. Para empezar, dentro del archivo Fuzzy.h se declaran las funciones de todos los sensores, para este caso como ejemplo se toma las funciones de DistanciaH, DistanciaM y Distancia (indicando los niveles alto, medio y bajo). Estas funciones permiten ser identificadas dentro del archivo Fuzzy.cpp, donde se implementa la lógica de cada función. Dentro de dicho archivo, cada función evalúa el valor de x, aplicando las reglas que corresponden a distintos rangos numéricos asociados a las categorías difusas establecidas. De esta

manera, se establece el grado de pertenencia del valor recibido a una categoría específica, de esta manera se realiza el proceso de los datos mediante lógica difusa.

Figura 54.

Configuración de operación de reglas de pertenencia.

```

30  ///----- Fuzzy.cpp ----- Fuzyficacion Distanicia -----
31
32  float Fuzzy: DistanceH(float x) {
33      if (x > 225) return 1;
34      else if (x >=180 && x <=225) return (x - 180) / 45;
35      else return 0;
36  }
37
38
39  float Fuzzy: DistanceM(float x) {
40      if (x >= 100 && x <=150) return (x - 100) / 50;
41      else if (x > 150 && x <=180) return 1;
42      else if (x > 180 && x <=225) return (x - 225) / -45 ;
43      else return 0;
44  }
45
46  float Fuzzy: Distancel(float x) {
47      if (x>=0 && x <= 100) return 1;
48      else if (x > 100 && x <= 150) return ( - x + 150) / 50;
49      else return 0;
50  }
51  }

```

```

1  #ifndef FUZZY_H
2  #define FUZZY_H
3
4  class Fuzzy {
5  public:
6      Fuzzy();
7
8      float RainH(float x);
9      float RainM(float x);
10     float RainL(float x);
11
12     float DistanceH(float x);
13     float DistanceM(float x);
14     float Distancel(float x);
15
16     float VelocityH(float x);
17     float VelocityM(float x);
18     float VelocityL(float x);
19
20     float Defuzzi(float bajo, float medio, float alto);
21
22 };
23

```

Fuente: Elaborado por el autor.

Dentro de la parte final del archivo Fuzzy.cpp, en la Figura 55 se implementa el proceso de defuzzificación en donde se implementa una salida precisa. Dentro de este apartado, se reciben como parámetros los grados de pertenencia a los conjuntos difusos bajo, medio y alto, a los cuales se les asigna valores representativos como son 1, 5 y 10 respectivamente, que indican la intensidad de la variable evaluada. Seguidamente, se realiza el cálculo de una media ponderada, multiplicando cada valor representativo por su grado de pertenencia y dividiendo la suma de estos productos por la suma total de los grados de pertenencia. Finalmente, el proceso permite obtener un valor numérico preciso, garantizando una respuesta coherente y estable por parte del sistema difuso.

Figura 55.

Configuración del proceso de defuzzificación.

```
float Fuzzy::Defuzzi(float bajo, float medio, float alto) {  
    // Los valores de entrada correspondientes a cada conjunto (se ajustar  
    float xBajo = 1;  
    float xMedio = 5;  
    float xAlto = 10;  
  
    // Numerador: Suma de los valores de pertenencia multiplicados por sus  
    float numerator = (xBajo * bajo) + (xMedio * medio) + (xAlto * alto);  
  
    // Denominador: Suma de los grados de pertenencia  
    float denominator = bajo + medio + alto;  
  
    // Evitar división por cero y devolver el resultado  
    return (denominator != 0) ? numerator / denominator : 0;  
}
```

Fuente: Elaborado por el autor.

4.2.2. Verificación de configuración de Nodo Coordinador.

Dentro de este apartado se da a conocer de forma general las partes más importantes del código establecido dentro del nodo coordinador, con el fin de lograr recibir la información que proviene del nodo sensor.

En la Figura 56, se da a conocer la función que se encarga de recibir la información mediante LoRa hacia el servidor MQTT, prácticamente actúa como un puente para obtener la información del nodo sensor.

Figura 56.

Función de visualización de datos de Rx mediante protocolo MQTT.


```

void send_mqtt_data() {

    if (!LoRaMessage.isEmpty()) {

        String mqttTopic2 = mqttTopic;
        mqttTopic2.replace("#", "ESP_RESP");
        // Procesar el mensaje recibido
        client.publish(mqttTopic2.c_str(), LoRaMessage.c_str());
    }
}

```

Fuente: Autoría.

En este apartado, se realiza el procesamiento e inferencia de datos que provienen del nodo sensor utilizando la tecnología LoRa. Procesa los datos y según la clase inferida como bajo, medio y alto, se ajusta de forma dinámica el intervalo de adquirir datos o a su vez generar una alerta. Este proceso es esencial para gestionar la información ya que controla el estado del nodo sensor para tomar los datos, garantizando una operación eficiente y automática.

Figura 57.

Procesamiento de control de IA.

```

void ai_loop() {
    LoRaMessage = LoRaRx();

    //Comprobacion de mensajes LoRa
    if (LoRaMessage != "") {
        if (lora_resp > 5) {
            lora_resp = 0;
            both.println("LoRA: ON");
        } else if (lora_resp < 5) {
            lora_resp = 0;
        }
        both.printf("Rx: %s\n", LoRaMessage.c_str());

        char buffer[LoRaMessage.length() + 1];
        strcpy(buffer, LoRaMessage.c_str());

        float inputData[3]; // Array para almacenar
        int index = 0;

        // Extraer los valores y almacenarlos en input
        char *token = strtok(buffer, ",");
        while (token != NULL && index < 3) {
            inputData[index++] = atof(token); // Conver
            token = strtok(NULL, ",");
        }

        if(ai_state[1]==10){
            both.println("AI: CHANGE");
            both.println("AI:"+String(ai_state[0])+","+String(ai_state[1]));

            if(ai_state[0]==0){
                lora_sense_time = lora_sense_time+1000;
                both.println("Time:"+String(lora_sense_time));
            }

            if(ai_state[0]==1){
                if(lora_sense_time>1000){
                    lora_sense_time = lora_sense_time-1000;
                }
                both.println("Time:"+String(lora_sense_time));
            }

            if(ai_state[0]==2){
                lora_sense_time = 1000;
                both.println("Peligro:Alto");
                FireService.SendAlert("PELIGRO");
            }
        }
        lora_sense[0]=0;
    }
    xTaskNotifyGive(respTaskHandle);
}

```

Fuente: Autoría.

4.2.3. Verificación de Nodo Actuador

Dentro de este nodo da a conocer la programación establecida para que realice su función de notificación hacia la parroquia. Dentro de la Figura 58, se observa las librerías utilizadas para la comunicación a redes inalámbricas, creación de un sitio web, SPIFFS, ArduinoOTA para realizar modificaciones de firmware y la librería ArduinoJson que se encarga de operar los datos en formato Json.

Figura 58.

Librerías de Nodo Actuador.

```

#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <SPIFFS.h>
#include <ArduinoOTA.h>
#include <PubSubClient.h>
#include <SPIFFS.h>
#include <ArduinoJson.h>

```

Fuente: Autoría.

Dentro de la Figura 59, se observa la configuración de conexión del servicio de MQTT. Como se logra observar en el cuadro rojo, se lanzará un mensaje verificando la conexión y encendiéndose un led como verificación de conexión a MQTT. Por otro lado, en el cuadro verde se observa el proceso de un mensaje de error, a esto se incrementa un contador para que reintente la conexión y lograr la conexión final, de esta manera asegurar los mensajes entrantes de manera correcta.

Figura 59.

Configuración de conexión a servicio MQTT.

```
//----- CONEXION MQTT -----

void eval_mqtt() {
    if (!client.connected()) {
        if (millis() - mqtt_timer >= 1000) {
            mqtt_timer = millis();
            Serial.println("MQTT: Conectando");
            String clientId = "ESP32Client_" + String(random(0xffff), HEX);
            bool status = client.connect(clientId.c_str());
            if (status) {
                Serial.println("Conectado al servidor MQTT!");
                client.subscribe(mqttTopic.c_str());
                Serial.println("MQTT: OK");
                mqtt_tries = 0;
                digitalWrite(2, HIGH);
            }
            else {
                if (mqtt_tries < 3) {
                    Serial.print("Error de conexión, rc=");
                    Serial.print(client.state());
                    Serial.println(" Intentando nuevamente");
                    Serial.println("MQTT: Reconectando");
                    mqtt_tries++;
                }
                else if (mqtt_tries == 3) {
                    Serial.println("MQTT: Fallo Conexión");
                    mqtt_tries = 4;
                }
                digitalWrite(2, LOW);
            }
        }
    }
}
```

Fuente: Autoría.

En la Figura 60, se observa la configuración para gestionar el estado de conexión WiFi del ESP32. Esta parte del código sirve para verificar la conexión WiFi, al lograr conectarse se lanzará una dirección IP y conexión al servicio de MQTT. Por otro lado, si

no se logra la conexión a WiFi, el dispositivo seguirá reintentando por 4 segundos. Al no lograr conexión, cada 3 segundos se envía un mensaje indicando que el dispositivo no se encuentra conectado. Esta parte de la programación es necesaria para visualizar el acceso a internet y lograr que el nodo cumpla su trabajo.

Figura 60.

Configuración de verificación de acceso a la red WiFi.

```
void checkWifiConnection() {  
    if (wifiConnecting) {  
        if (WiFi.status() == WL_CONNECTED) {  
            wifiConnecting = false; // Finaliza el intento de conexión  
            Serial.println("Wifi Conectado !");  
            Serial.println(WiFi.SSID());  
            Serial.println(WiFi.localIP().toString());  
            mqtt_activo = true;  
        } else if (millis() - lastAttemptTime > 4000) { // Timeout de 3  
            wifiConnecting = false;  
            Serial.println("Wifi: No Conectado");  
            WiFi.disconnectAsync(true, false);  
            mqtt_activo = false;  
        } else if (millis() - lastAttemptTime < 4000) {  
            if (millis() - lastPrintTime >= 300) { // Imprimir cada 300 ms  
                digitalWrite(2, aux);  
                aux = !aux;  
                Serial.print(".");  
                lastPrintTime = millis(); // Actualizar el último tiempo de  
            }  
        }  
    }  
    if (WiFi.status() != WL_CONNECTED && millis() - autocon > 5000) {  
        autocon = millis();  
        connect_sta(NULLptr);  
    }  
}
```

Fuente: Autoría.

Dentro de la Figura 61, se observa la configuración establecida para que se reciba mensajes mediante MQTT, se debe recibir mensajes de ON que indica que debe activar la alarma y OFF que indica que debe desactivar la alarma o se debe mantener desactivada.

Figura 61.

Configuración de estado de alarma.

```
void callback(char *topic, byte *message, unsigned int length) {

    String mqttTopic2 = mqttTopic;
    mqttTopic2.replace("#", "ACT/");

    if (String(topic) == mqttTopic2) {

        String mqttTopic2 = mqttTopic;
        mqttTopic2.replace("#", "ACT_RESP");
        String resp = "";

        String messageTemp;
        for (int i = 0; i < length; i++) {
            messageTemp += (char)message[i];
        }

        if (messageTemp == "ON") {
            alarma = 0;
            resp = "Activado";
        } else if (messageTemp == "OFF") {
            alarma = 1;
            resp = "Desactivado";
        }
        else {
            resp = "Comando No reconocido";
        }

        //send_mqtt_data
        client.publish(mqttTopic2.c_str(), resp.c_str());
    }
}
```

Fuente: Autoría.

4.3. Entrenamiento de Datos

En este apartado se da a conocer el proceso de entrenamiento de los datos recolectados del nodo sensor. Para llevar a cabo el entrenamiento se utilizó Google Colab como se especificó en los requerimientos en el apartado 3.5.2, la cual ofrece acceso gratuito a recursos computacionales de alto rendimiento. Google Colab, al ser una plataforma ubicada en la nube, permite ejecutar procesos de gran complejidad para evitar utilizar recursos del equipo físico.

4.3.1. Entrenamiento del modelo

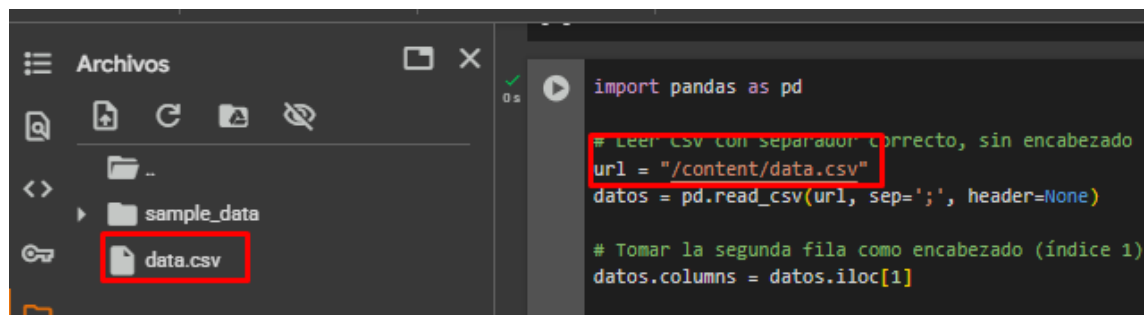
En este apartado se realiza la lectura de los datos recolectados para el entrenamiento en formato CSV, el cual contiene los datos que provienen de los sensores,

y que representan las variables de entrada para el sistema. Finalmente, dentro de cada línea de la representación de datos aparece el nivel de riesgo, este último es la variable de salida, esto es fundamental para dar a conocer el aprendizaje al modelo utilizando un enfoque de lógica difusa.

En la Figura 62, se carga el archivo CSV descargado del sitio web del sistema, seguidamente se indica la ruta para la validación del archivo “*data.csv*”.

Figura 62.

Carga de Dataset en Google Colab.



Fuente: Autoría.

Dentro de la Figura 63, se logra observar los primeros registros ordenados en filas y columnas para verificar que el proceso de visualización de los datos se ha realizado correctamente.

Figura 63.

Modelo de ingreso de datos.

```
url = "/content/datos.csv"
datos = pd.read_csv(url)
datos.head()
```

	lluvia	distancia	velocidad	peligro
0	7.35	6.69	5.78	medio
1	6.09	9.28	5.02	medio
2	7.70	9.61	9.91	alto
3	5.87	4.89	2.08	bajo
4	1.94	8.96	3.25	medio

Fuente: Autoría.

En la Figura 64, se visualiza el bloque de configuración de manera general para entrenar los datos. Se procede a construir el modelo secuencial que va compuesto por varias capas densamente conectadas, en donde se tienen las tres primeras capas con 16 neuronas y la última con 3 neuronas que representan las clases peligro: bajo, medio y alto. Para esta última capa se utiliza la función softmax la cual es ideal para ejecutar tareas de clasificación multiclase. El modelo entrena durante 300 épocas para validar la precisión y la pérdida durante el entrenamiento. Los datos recolectados son fundamentales para lograr entrenar de mejor manera el modelo y lograr predecir adecuadamente el nivel de peligro.

Figura 64.

Parámetros de entrenamiento de datos.

```

import tensorflow as tf
from tensorflow.keras import layers
import numpy as np

# Asegúrate de que X_train y X_test sean de tipo float32
X_train = X_train.astype(np.float32)
X_test = X_test.astype(np.float32)

# Definir el modelo
model = tf.keras.Sequential()

model.add(layers.Dense(16, activation='relu', input_dim=3))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(3, activation='softmax')) # Usar softmax para clasificación

# Configuración del entrenamiento
model.compile(loss='sparse_categorical_crossentropy', # Usar sparse_categorical_crossentropy
              optimizer='adam',
              metrics=['accuracy'])

# Ver el resumen del modelo
model.summary()

# Entrenar el modelo
history = model.fit(X_train, y_train, epochs=300, validation_data=(X_test, y_test))

# Evaluar el modelo en los datos de prueba
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)

# Imprimir la precisión en los datos de prueba
print(f"Test accuracy: {accuracy}")

```

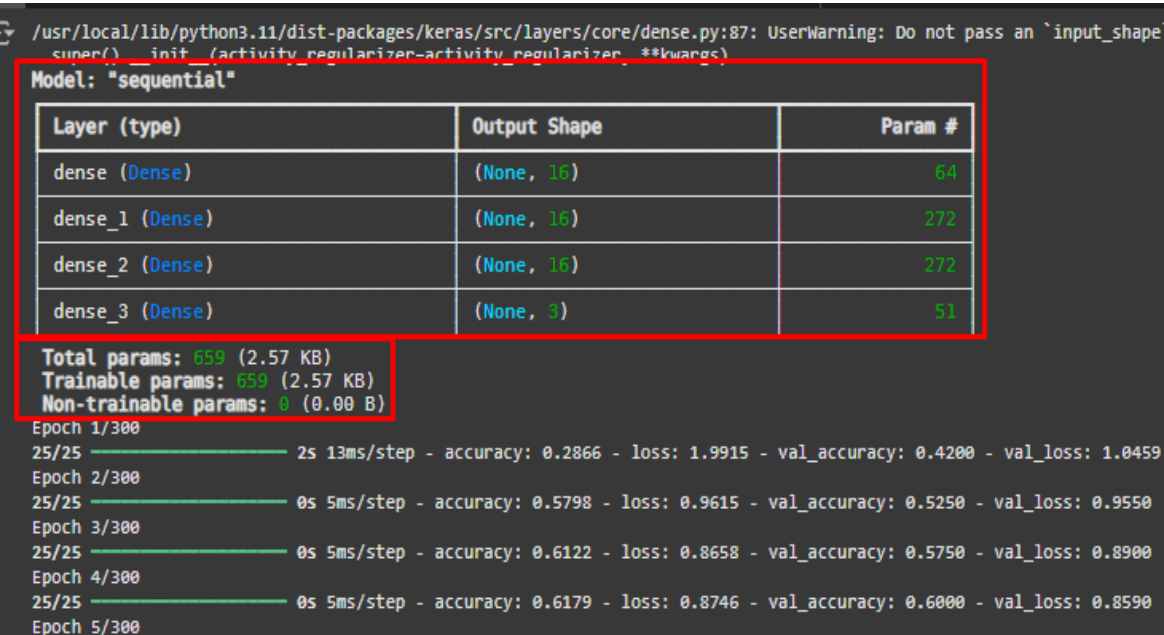
Fuente: Autoría.

En la Figura 65, al ejecutar el proceso de entrenamiento, se logra obtener el modelo secuencial en donde se detalla las cuatro capas densas. Las tres primeras capas se encargan de procesar los datos de entrada y la última capa contiene las clases de salida que indican el nivel de riesgo. En total, se evaluaron 659 parámetros para entrenar el modelo, estos se reparten entre cada capa dependiendo el peso y característica. El modelo que se entrena contiene una arquitectura ligera para acoplarse al microcontrolador, de igual manera, lograr su entrenamiento y la ejecución de la inferencia de manera efectiva. El número de “*Param*”, representa los pesos y sesgos que el modelo utiliza para aprender patrones a partir de los datos cargados. La cantidad de parámetros indica que si hay pocos

puede existir un subajuste, y si hay un valor más alto, puede memorizar los datos sin sobreajuste.

Figura 65.

Configuración de modelo secuencial para el entrenamiento.



Fuente: Autoría.

En la Figura 66, se observa el resultado de entrenamiento durante las 300 épocas, logrando visualizar que, en la última época el modelo arroja un valor de exactitud del entrenamiento de 0.9920% y una precisión de 0.98%, esto indica que el modelo tuvo un entrenamiento exitoso, finalizado el proceso de entrenamiento, se realiza un test de exactitud de manera general y se obtiene como resultado final un 0.98% de precisión, esto indica que el modelo tiene un buen desempeño para el sistema de monitoreo. El valor obtenido en el Test accuracy contiene muchos valores decimales, esto quiere decir que el valor obtenido está siendo presentado con más precisión, pero no implica que el modelo sea más exacto.

Figura 66.

Precisión del modelo durante el entrenamiento.

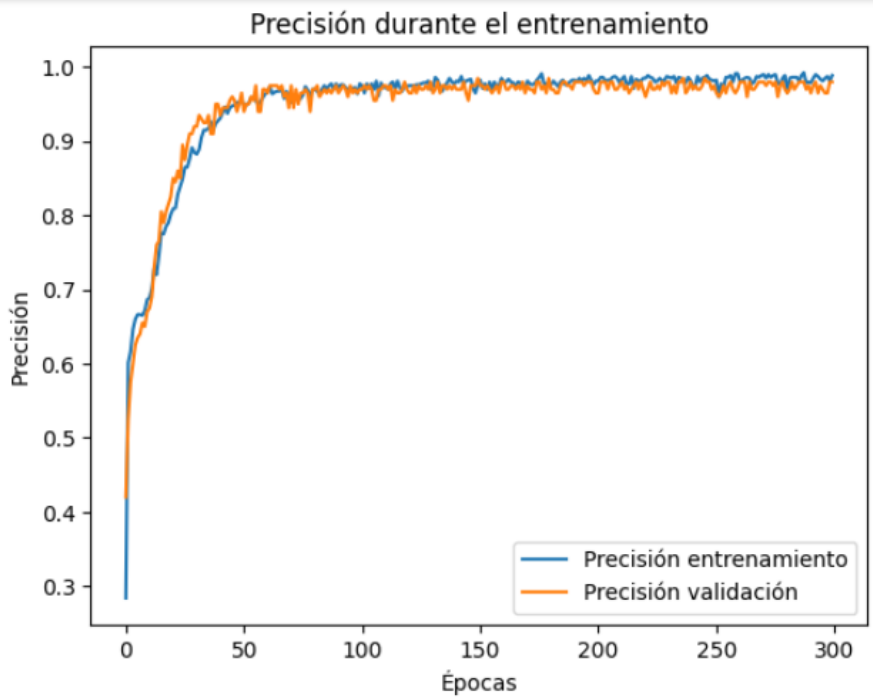
```
Epoch 296/300  
25/25 ————— 0s 6ms/step - accuracy: 0.9827 - loss: 0.0506 - val_accuracy: 0.9700 - val_loss: 0.0872  
Epoch 297/300  
25/25 ————— 0s 5ms/step - accuracy: 0.9876 - loss: 0.0369 - val_accuracy: 0.9650 - val_loss: 0.0828  
Epoch 298/300  
25/25 ————— 0s 6ms/step - accuracy: 0.9871 - loss: 0.0349 - val_accuracy: 0.9650 - val_loss: 0.0965  
Epoch 299/300  
25/25 ————— 0s 5ms/step - accuracy: 0.9832 - loss: 0.0471 - val_accuracy: 0.9800 - val_loss: 0.0605  
Epoch 300/300  
25/25 ————— 0s 6ms/step - accuracy: 0.9920 - loss: 0.0384 - val_accuracy: 0.9800 - val_loss: 0.0673  
Test accuracy: 0.9800000190734863
```

Fuente: Autoría.

Al finalizar el entrenamiento en 300 épocas, se obtiene una gráfica de precisión del modelo como se evidencia en la Figura 67, se logra evidenciar que las curvas tanto de entrenamiento como de validación se estabilizan por encima del 95% de precisión. Esto indica que el modelo ha logrado aprender a clasificar datos con una precisión adecuada. Se logra observar que el modelo no presenta sobreajuste, lo que da una buena señal del aprendizaje. Dentro del comportamiento, a pesar de que el modelo indica una estabilidad a partir de la época 150, se optó por extender el entrenamiento hasta las 300 épocas con la finalidad de comprobar la estabilidad del aprendizaje y confirmar que no existe sobreajuste dentro de las 300 épocas.

Figura 67.

Comportamiento de precisión de entrenamiento y validación por épocas.

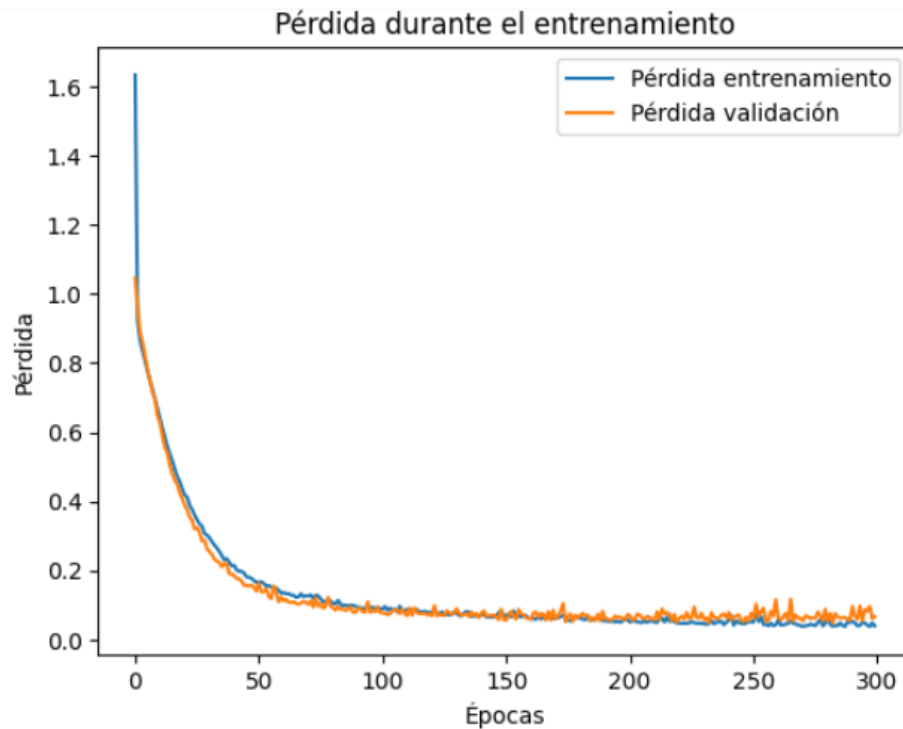


Fuente: Autoría.

En la Figura 68, se da a conocer el resultado del comportamiento de pérdida durante el entrenamiento del modelo. Como se logra observar, la pérdida va disminuyendo de forma rápida durante las primeras épocas y estabilizándose aproximadamente en un valor de 0.1 a partir de la época 100. Los valores de pérdida de entrenamiento y de validación se mantienen valores bajos y estables durante todo el proceso, lo que confirma que el modelo ha logrado minimizar positivamente el error de predicción sin presentar modificaciones.

Figura 68.

Comportamiento pérdida de entrenamiento y validación por épocas.



Fuente: Autoría.

4.3.2. Obtención del modelo entrenado

Al haber finalizado el entrenamiento del modelo y convertir al formato de TensorFlow lite para microcontroladores, se procede a transformar el modelo a formato en lenguaje C, este proceso se encarga de contener los datos en formato hexadecimal. Además, al realizar este tipo de conversión es fundamental para ejecutarlo dentro del WiFi LoRa 32. El proceso de convertir el formato se logra visualizar dentro del **ANEXO 8**.

En la Figura 69, se demuestra el modelo convertido, llevado a cabo desde un entorno desarrollado en Python a una implementación dentro del entorno Arduino IDE, cumpliendo así la eficiencia del modelo y permitiendo realizar inferencias en un tiempo adecuado, logrando obtener una lectura de datos funcionales para el sistema.

Figura 69.

Modelo Convertido a formato C.

```
# Save the file as a C source file
!xxd -i exercise_model.tflite > exercise_model.cc
# Print the source file
!cat exercise_model.cc

unsigned char exercise_model_tflite[] = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
  0x98, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x00, 0x34, 0x0c, 0x00, 0x00,
  0x94, 0x0c, 0x00, 0x00, 0x98, 0x13, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0a, 0x00,
  0x10, 0x00, 0x0c, 0x00, 0x08, 0x00, 0x04, 0x00, 0x0a, 0x00, 0x00, 0x00,
  0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
  0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x90, 0xff, 0xff, 0xff, 0x0d, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x6f, 0x75, 0x74, 0x70,
  0x75, 0x74, 0x5f, 0x30, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x5e, 0xf3, 0xff, 0xff, 0x04, 0x00, 0x00, 0x00,
  0x0c, 0x00, 0x00, 0x00, 0x6b, 0x65, 0x72, 0x61, 0x73, 0x5f, 0x74, 0x65,
  0x6e, 0x73, 0x6f, 0x72, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00,
  0x34, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0xdc, 0xff,
  0x10, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00,
  0x43, 0x4f, 0x4e, 0x56, 0x45, 0x52, 0x53, 0x49, 0x4f, 0x4e, 0x5f, 0x4d,
  0x45, 0x54, 0x41, 0x44, 0x41, 0x54, 0x41, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x08, 0x00, 0x00, 0x00, 0x0f, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00, 0x6d, 0x69, 0x6e, 0x5f,
  0x72, 0x75, 0x6e, 0x74, 0x69, 0x6d, 0x65, 0x5f, 0x76, 0x65, 0x72, 0x73,
  0x69, 0x6f, 0x6e, 0x00, 0x11, 0x00, 0x00, 0x00, 0x90, 0x0b, 0x00, 0x00,
  0x38, 0x0b, 0x00, 0x00, 0xb8, 0x0a, 0x00, 0x00, 0xa8, 0x06, 0x00, 0x00,
  0x98, 0x02, 0x00, 0x00, 0x7c, 0x02, 0x00, 0x00, 0x2c, 0x02, 0x00, 0x00,
  0xdc, 0x01, 0x00, 0x00, 0x8c, 0x01, 0x00, 0x00, 0xbc, 0x00, 0x00, 0x00,
  0xb4, 0x00, 0x00, 0x00, 0xac, 0x00, 0x00, 0x00, 0xa4, 0x00, 0x00, 0x00,
  0x9c, 0x00, 0x00, 0x00, 0x94, 0x00, 0x00, 0x00, 0x74, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x1e, 0xf4, 0xff, 0xff, 0x04, 0x00, 0x00, 0x00,
  0x60, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x0e, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x08, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00,
  0x28, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x08, 0x00, 0x04, 0x00,
  0x06, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
  0xea, 0x03, 0x00, 0x00, 0x0c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00,
```

Fuente: Autoría.

4.4. Instalación del sistema de monitoreo

En este apartado se lleva a cabo la instalación del nodo coordinador y el nodo sensor dentro de los sitios estratégicos, con la finalidad de recolectar los datos y lograr transmitir los datos al nodo coordinador.

4.4.1. Instalación de Nodo Sensor

En base a las coordenadas geográficas escritas dentro de la Tabla 15, se coloca el nodo sensor, dentro de este lugar se determina una zona estratégica para instalar el nodo sensor para que se obtenga lecturas correctas sobre el nivel del agua del río, velocidad y precipitación. En la Figura 70 se muestra el punto en donde se encuentra instalado el nodo sensor.

Figura 70.

Instalación de Nodo Sensor.



Fuente: Autoría.

4.4.2. Instalación de Nodo Coordinador

De igual forma, el nodo coordinador es colocado en una zona estratégica, esto con el fin de recibir adecuadamente los datos enviados desde el nodo sensor. Dentro de este nodo solamente se encuentra el módulo WiFi LoRa 32 y cuenta con conexión a internet.

Figura 71.

Instalación de Nodo Coordinador.



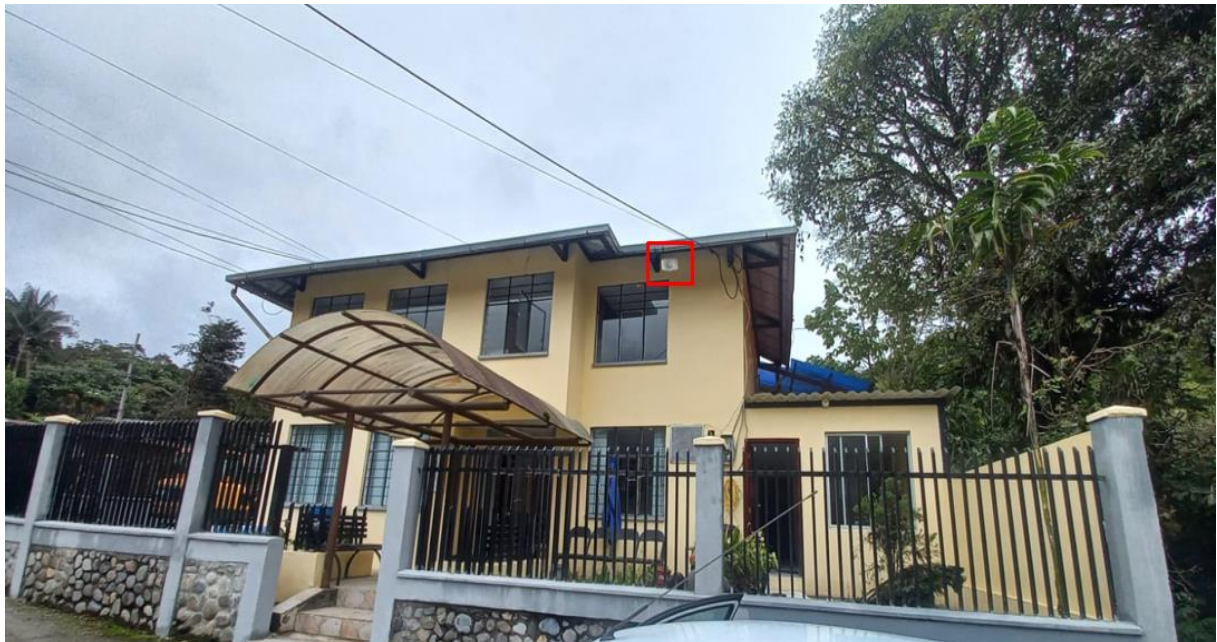
Fuente: Autoría.

4.4.3. Instalación de Nodo Actuador

El nodo actuador fue instalado en la parte externa del GADPR Cosanga con el fin de proteger la sirena de la lluvia para que mantenga su rendimiento. Además, se colocó a una altura para que se logre escuchar su sonido. Hay que tomar en cuenta que la sirena es otro método de alerta para usuarios cuando el nivel de desbordamiento es alto.

Figura 72.

Instalación de Nodo Actuador.



Fuente: Autoría.

4.5. Pruebas de funcionamiento del sistema

A continuación, se definen diversas pruebas funcionales del sistema, las cuales se ejecutan de forma individual en cada uno de los bloques que lo componen según la arquitectura planteada.

Tabla 35.

Lista de pruebas realizadas dentro del sistema.

Test de prueba	Descripción	Resultado Esperado
Test 1: Verificación de comunicación inalámbrica.	Se verifica la inicialización de los parámetros establecidos para que exista la comunicación para el envío de datos entre ambos nodos.	Se visualizará la confirmación de comunicación LoRa
Test 2: Inicialización de configuraciones de nodo sensor.	Se realiza el proceso de inicio de las configuraciones establecidas como la red para el acceso a la interfaz web que contiene el nodo y el proceso de carga de la lógica difusa.	Se verifica que el nodo inicie los servicios para procesar los datos.

Test 3: Captura y visualización de datos de nodo sensor.	Se evalúa los datos que va tomando el nodo sensor y se logra visualizar mediante el sitio web que establece el nodo y de igual manera se logra visualizar los datos de manera asíncrona.	Se observan los datos recolectados y procesados por lógica difusa.
Test 4: Funcionamiento de nodo coordinador.	Se realiza la inicialización de todos los parámetros como el acceso a internet, conexión con el servicio MQTT, inicio con el servicio Firebase, activación de la IA, comunicación LoRa e inicio de servidor web.	Verificación de acceso a los servicios y recepción de datos.
Test 5: Funcionamiento de sitio web y aplicación móvil del sistema.	Se evalúa la precisión al momento de obtener datos muy elevados, verificando de esa manera el envío de una notificación como alerta y visualizando los datos y el porcentaje de riesgo dentro del sitio web y aplicación móvil.	Visualización de datos procesados, porcentaje de alerta, gráficas de datos y notificaciones.
Test 6: Funcionamiento del sistema de monitoreo.	Se evalúa todos los procesos mencionados en los test de prueba anteriores, esto con el fin de evidenciar el funcionamiento de manera general e inducir al sistema a niveles altos de lecturas.	Al inducir al sistema con niveles de lecturas altos se espera los mensajes de alerta temprana como la activación del nodo actuador y notificaciones dando a conocer la alerta y el tiempo de evacuación.

Fuente: Elaborado por el autor.

4.5.1. Pruebas de comunicación inalámbrica

Dentro de este apartado se lleva a cabo pruebas de funcionamiento y transmisión de datos desde el nodo sensor hacia el nodo coordinador. Para comprobar la conexión del microcontrolador, se utiliza la herramienta Arduino IDE, mediante la cual se conecta el WiFi LoRa 32 por comunicación serial COM3 del computador, permitiendo a su vez verificar la estabilidad de comunicación y transmisión de datos.

En la Figura 73, se logra visualizar el estado activo de comunicación inalámbrica mediante LoRa, esto significa que entre el nodo sensor y nodo coordinador ya existe una comunicación estable para iniciar enviar información.

Figura 73.

Inicialización de nodo coordinador.

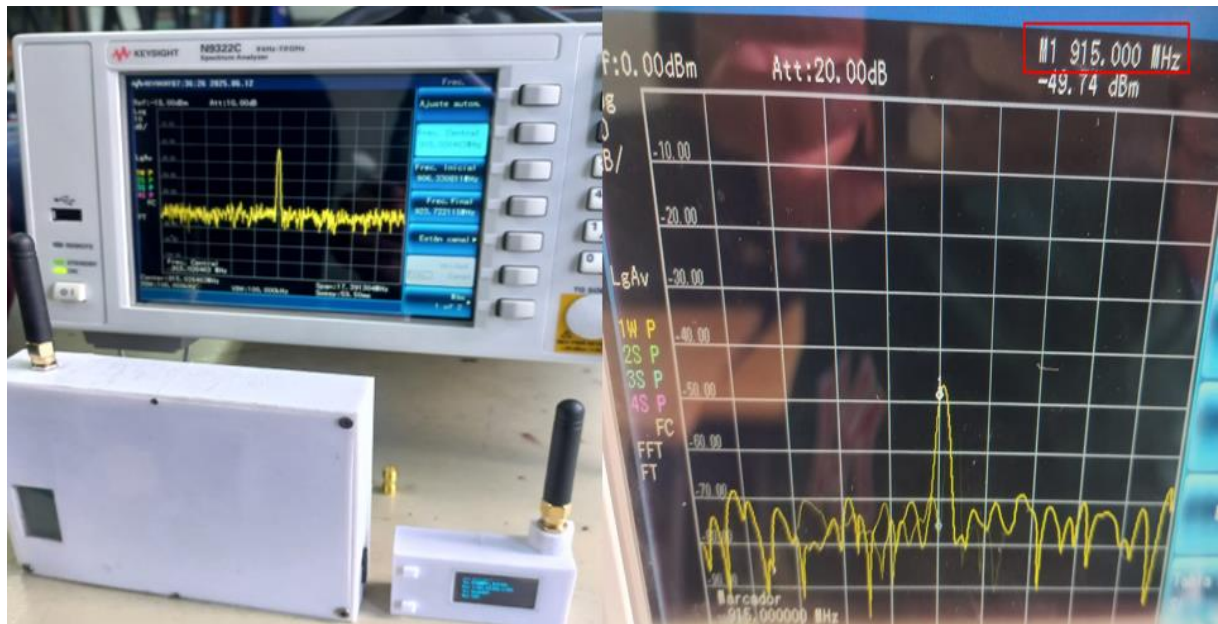
```
Wifi Conectado !  
TONY  
192.168.137.223  
MQTT: Conectando  
Conectado al servidor MQTT!  
MQTT: OK  
Hora:09:46:32  
E-Rx: -7  
E-Rx: -7  
LoRA: ON  
Rx: 5.00,7.36,1.00
```

Fuente: Autoría.

Utilizando en analizador de espectro se logra visualizar la frecuencia con la que está operando el microcontrolador WiFi LoRa 32, como se logra observar en la Figura 74, la frecuencia en la que trabaja el módulo es de 915 MHz.

Figura 74.

Prueba de frecuencia LoRa

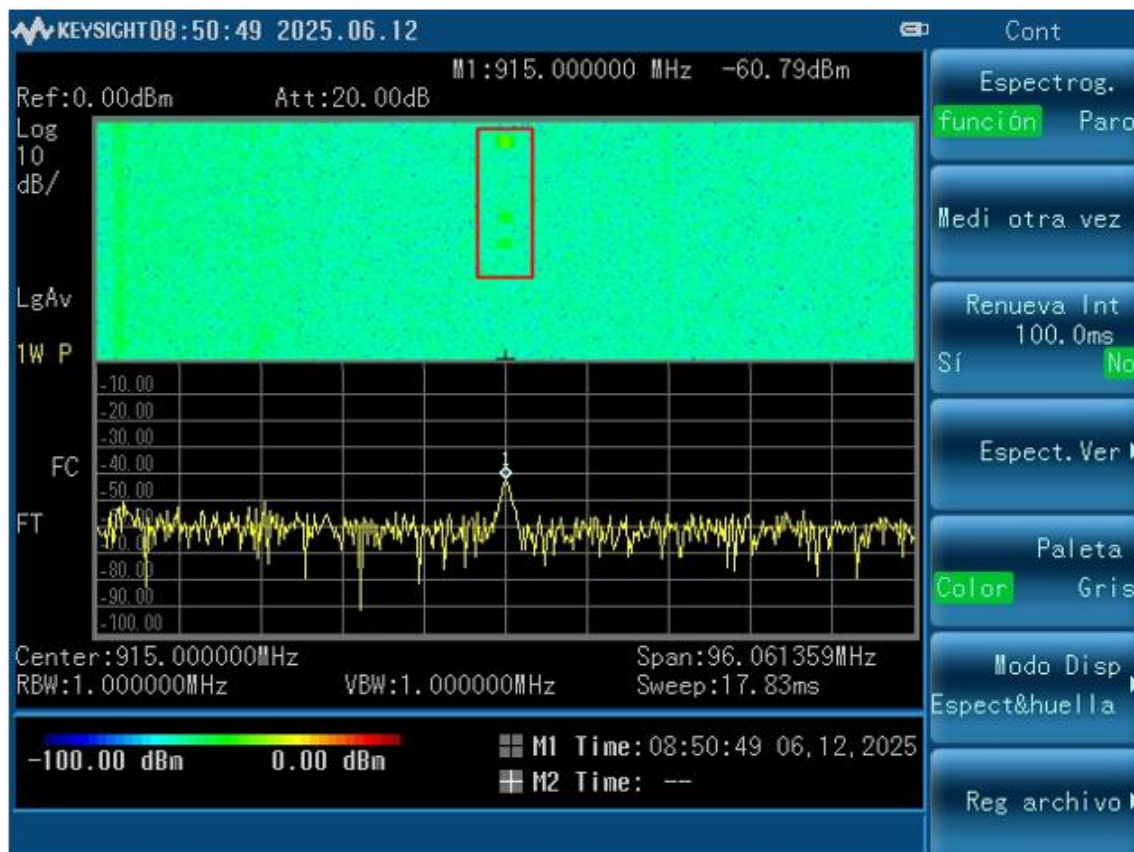


Fuente: Autoría.

En la figura 75, se observa el muestreo del espectro junto con las zonas de color dentro de la zona verde, esto indica pulsos intermitentes que corresponde a los paquetes que se transmiten dentro de la banda de los 915 MHz.

Figura 75.

Visualización de espectro LoRa



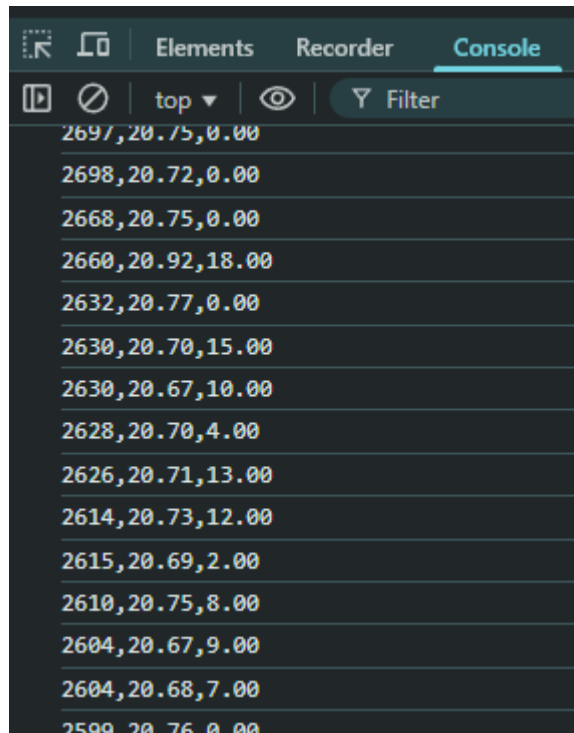
Fuente: Autoría.

4.5.2. Pruebas de Funcionamiento de Nodo Sensor

En el bloque de sensores se realizan las mediciones de las variables: la cantidad de precipitación (V), nivel del agua en el río (cm) y la velocidad del flujo del río (m/s). Los resultados obtenidos, presentados en la Figura 76, confirman que los datos recolectados son correctos.

Figura 76.

Resultado de bloque de Nodo Sensor.



Fuente: Autoría.

4.5.3. Captura y Visualización de datos del nodo sensor

Mediante la interfaz web que contiene el nodo sensor, se da inicio al proceso de toma de datos en cada sensor: precipitación (V), nivel del agua en el río (cm) y velocidad (m/s) respectivamente, esto se observa en la Figura 77.

Figura 77.

Interfaz de inicio de recolección de datos.

192.168.4.1

12

Test BedConfiguración

RAW DATA

Lluvia850

Distancia34.36

Velocidad111

Probar Datos

FUZZY DATA

Lluvia2.3

Distancia10

Velocidad1

Probar Datos

TIMER

Valor [ms]500

Enviar

Reiniciar ESP

Fuente: Autoría.

En la Figura 78 se evidencia los datos almacenados del nodo sensor.

Figura 78.

Captura y visualización de datos del nodo sensor.

Fecha	Ultrasonico	Lluvia	Infrarrojo
02-04-2025 20:25	5	10	2
02-04-2025 20:26	5	10	2
02-04-2025 20:26	10	10	10
02-04-2025 20:28	10	10	10
02-04-2025 20:28	10	10	5
02-04-2025 20:28	5	10	5
02-04-2025 20:29	5	5	5
03-04-2025 11:58	34.29	1288	0
03-04-2025 11:58	34.29	1279	0
03-04-2025 11:58	34.31	1307	45
03-04-2025 11:58	34.29	1320	0
03-04-2025 11:58	34.29	1332	0
03-04-2025 11:59	34.29	1338	0
03-04-2025 11:59	34.29	1337	0
03-04-2025 11:59	34.26	1348	47
03-04-2025 11:59	34.29	1360	0

Fuente. Autoría.

4.5.4. Prueba de funcionamiento de nodo coordinador

Mediante el monitor serial se logra observar la inicialización del nodo coordinador junto con los diferentes parámetros de comunicación establecidos. Además, se evidencia la confirmación de acceso a internet, al servidor web y al broker MQTT.

Figura 79.

Inicialización de nodo coordinador.

```

17:42:18.125 -> SPIFFS: OK
17:42:18.213 -> Server:OK
17:42:18.278 -> Intentando conectar a WiFiWifi Conectado
17:42:18.447 -> Tesis
17:42:18.481 -> 192.168.9.94
17:42:18.662 -> MQTT: Conectando
17:42:19.737 -> Conectado al servidor MQTT!
17:42:23.911 -> Hora:17:42:26
17:45:00.873 -> E (162889) gpio: gpio_isr_handler_remove(542): GPIO isr service
17:47:47.388 -> AI: CHANGE
17:47:47.453 -> AI:0,10
17:47:47.485 -> Time:3000

```

Fuente: Autoría

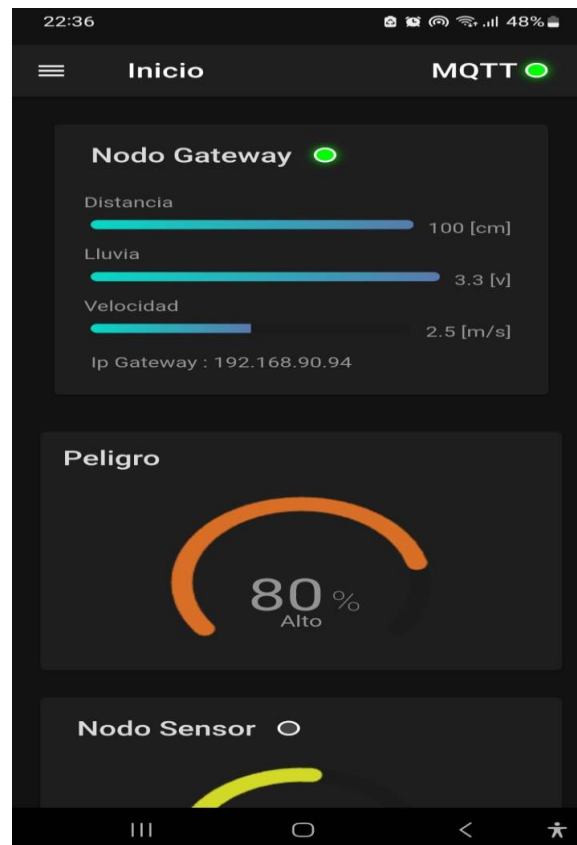
4.5.5. Prueba de funcionamiento de Sitio web y Aplicación móvil

Dentro de este apartado, se lleva a cabo la prueba de funcionamiento de tanto de la aplicación móvil como también del sitio web del sistema, en cada uno se muestra los datos que son procesados desde el nodo sensor hasta que llegue al cliente final. Cada interfaz cuenta con un menú de opciones en donde permite: observar de manera general los datos, acceder a un calendario de datos y finalmente un apartado en donde solo el usuario autorizado podrá acceder para configurar algún parámetro del sistema.

En la Figura 80, se logra observar los datos que son enviados desde el nodo sensor, así como también el porcentaje de riesgo y el nivel de batería de dicho nodo, de igual manera, se logra visualizar la alerta que envía el sistema hacia el dispositivo móvil cuando existe un riesgo muy elevado.

Figura 80.

Prueba de funcionamiento en aplicación móvil.

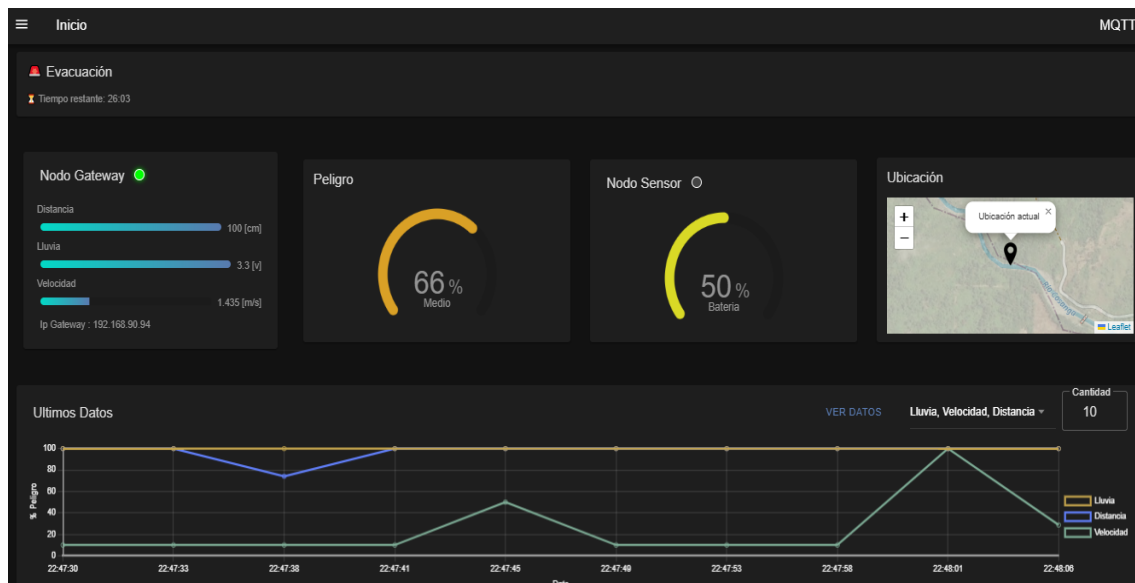


Fuente: Autoría.

De igual manera, dentro de la Figura 81 se logra visualizar el sitio web desarrollado para el sistema, allí se observa los datos procesados de cada sensor y el porcentaje de riesgo como salida de los datos obtenidos. Además, se da a conocer el estado de batería del nodo sensor, las gráficas de cada sensor y un pequeño mapa indicando la zona en donde se realiza el proceso de recolección de datos.

Figura 81.

Prueba de funcionamiento de sitio web del sistema de monitoreo.



Fuente: Autoría.

Por otro lado, en la Figura 82 en el apartado de Data Picker del sistema permite elegir la fecha de inicio y fecha final de las lecturas de los datos que se desean consultar y en la parte inferior visualizar el nivel de peligro que se formaron con los datos recolectados. Además, este apartado es importante ya que, al permitir descargar los datos de las fechas específicas, se puede visualizar y realizar un seguimiento de los cambios del comportamiento del río. De igual manera, estos datos son ideales para realizar un análisis más profundo del comportamiento del río y crear histogramas en donde se vaya dando a conocer el comportamiento de los niveles del río en las fechas seleccionadas.

Figura 82.

Funcionamiento de Data Picker del sitio web.



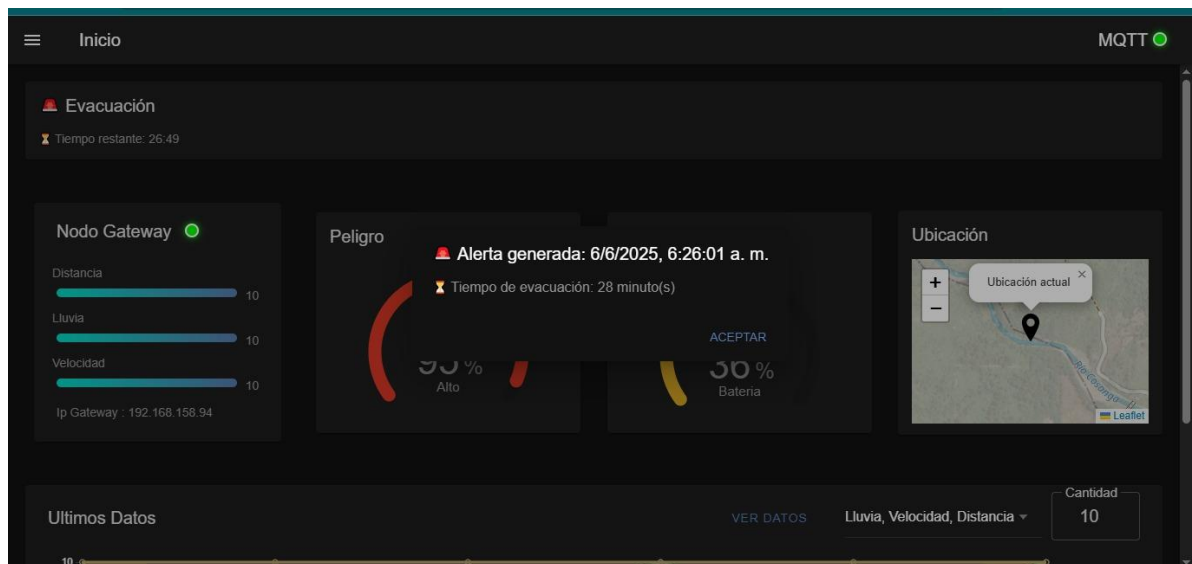
Fuente: Autoría.

4.5.6. Funcionamiento del sistema de monitoreo

En este apartado se observa el funcionamiento del sistema con todos los componentes electrónicos conectados y funcionando respectivamente. Para evaluar el rendimiento del sistema, se llevó a cabo inducir al sistema de monitoreo con lecturas de datos críticos, con el fin de activar las alertas tempranas como el nodo actuador y las notificaciones. Esta prueba se encarga de evidenciar el tiempo que tiene para evacuar el usuario y observar el comportamiento del sistema en casos reales. Es fundamental que el sistema no falle en casos reales y actúe de manera correcta, cumpliendo así la funcionalidad de prevenir y ayudar ante posibles eventos de desbordamiento.

Figura 83.

Prueba de alerta del sistema.

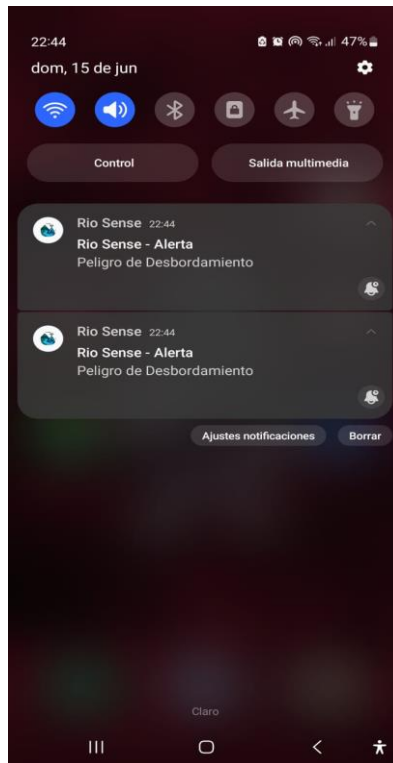


Fuente: Autoría.

En la Figura 84, se logra visualizar la notificación que es emitida desde el sistema de monitoreo. Al inducir el sistema, esto se encarga de que se active todos los procesos que indiquen el aviso de la alerta. En este caso, es necesario abrir la notificación y visualizar los datos que están capturando los sensores y verificar cuanto tiempo queda para evacuar.

Figura 84.

Prueba de notificación de sistema de monitoreo.

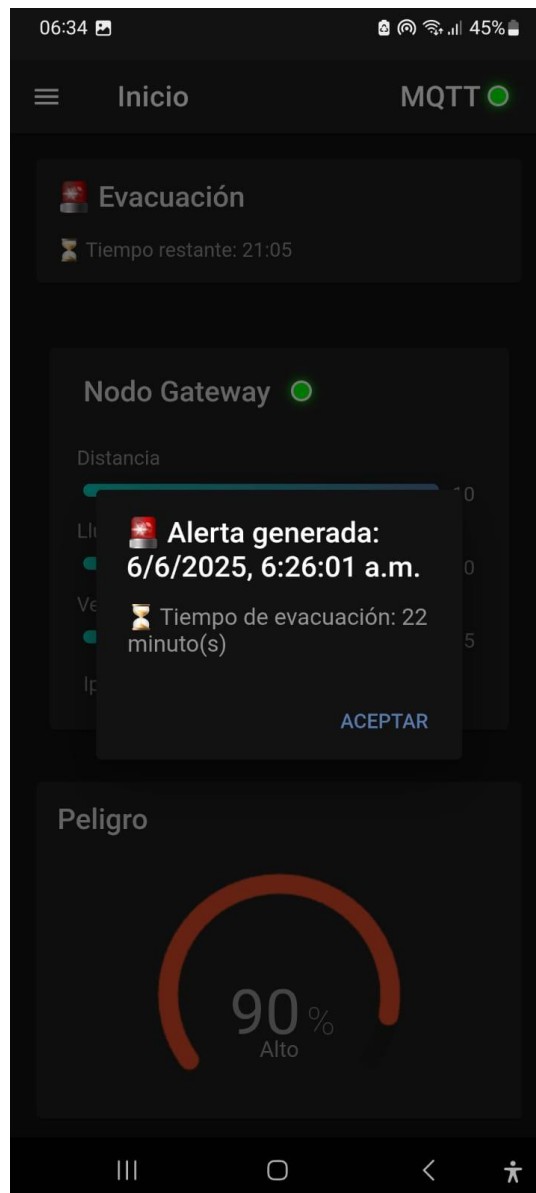


Fuente: Autoría.

Una vez emitida la alerta, dentro de la aplicación o sitio web se visualizará la alerta, indicando cuanto tiempo tiene el usuario para evacuar. Esto es ideal ya que de esa manera se identifica el tiempo de evacuación y las lecturas de los sensores, con el objetivo de lograr ir a zonas seguras.

Figura 85.

Prueba de alerta del sistema de monitoreo.

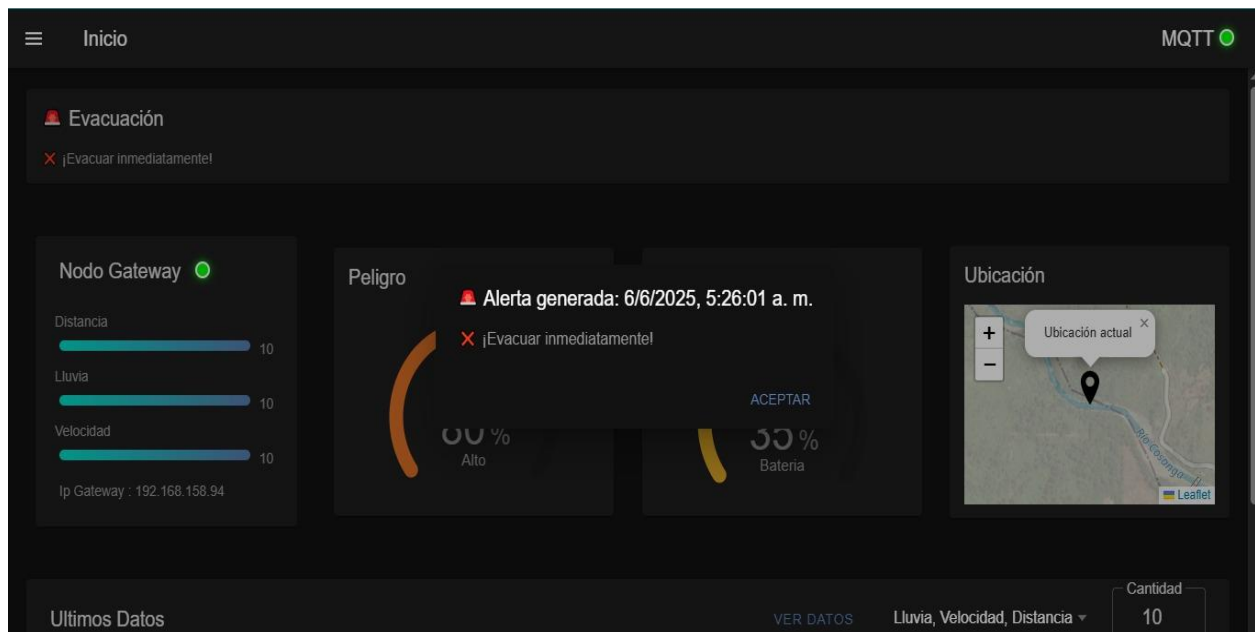


Fuente: Autoría.

Dentro de la Figura 86, se observa cuando el temporizador llega a 0 y se emite un mensaje indicando que la evacuación debe ser inmediata.

Figura 86.

Prueba de finalización de tiempo de alerta.

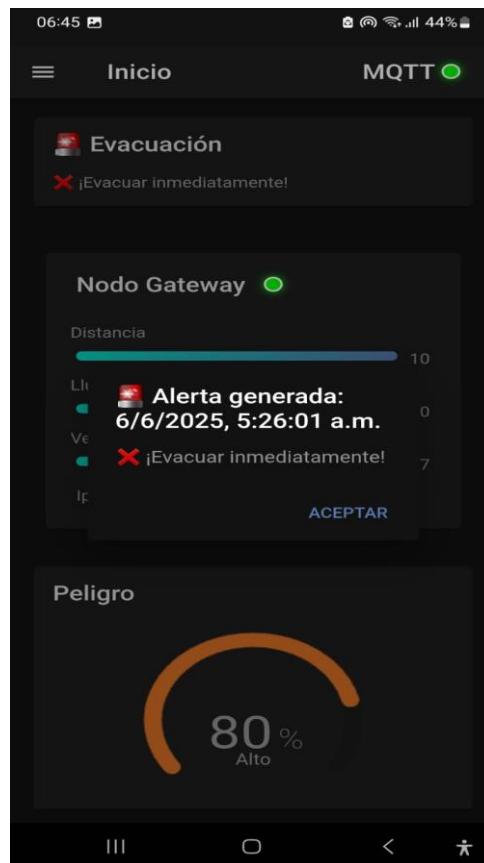


Fuente: Autoría.

De igual manera dentro de la aplicación móvil se observa cuando el temporizador finaliza y se emite el mensaje de evacuación inmediata.

Figura 87.

Prueba de finalización de tiempo de alerta en app.



Fuente: Autoría.

4.6. Análisis de datos

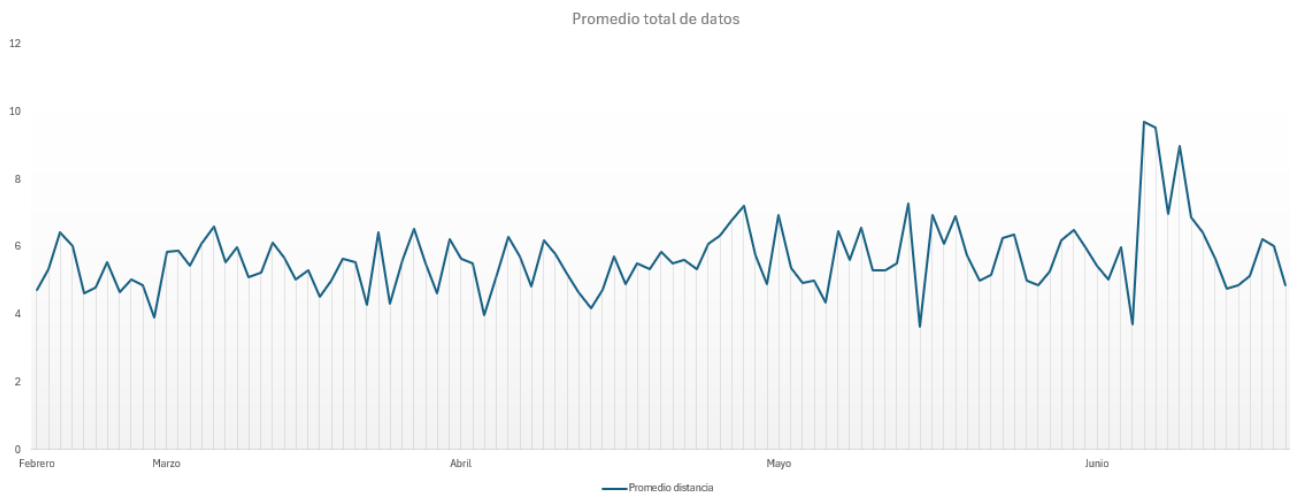
Dentro de la Figura 88, se observa el promedio del nivel del agua en el río, las lecturas son tomadas desde el mes de febrero hasta junio. De manera general, se obtiene un rango de lecturas entre 4 y 6 del sensor de distancia medido en cm.

Según (*PDOT-Cosanga, 2019*), los niveles más altos de precipitación comienzan desde mayo y junio con datos de 223 y 318mm respectivamente, esto provoca que el nivel del río se eleve y se generen lecturas con pico altos cercanos a 10.

Por otro lado, los datos de los meses de febrero, marzo y abril se mantienen en una tendencia con picos moderados entre altos y bajos, esto indica que existieron días con mucha precipitación y días con poca presencia de lluvia.

Figura 88.

Promedio de nivel de agua de Febrero-Junio 2025.



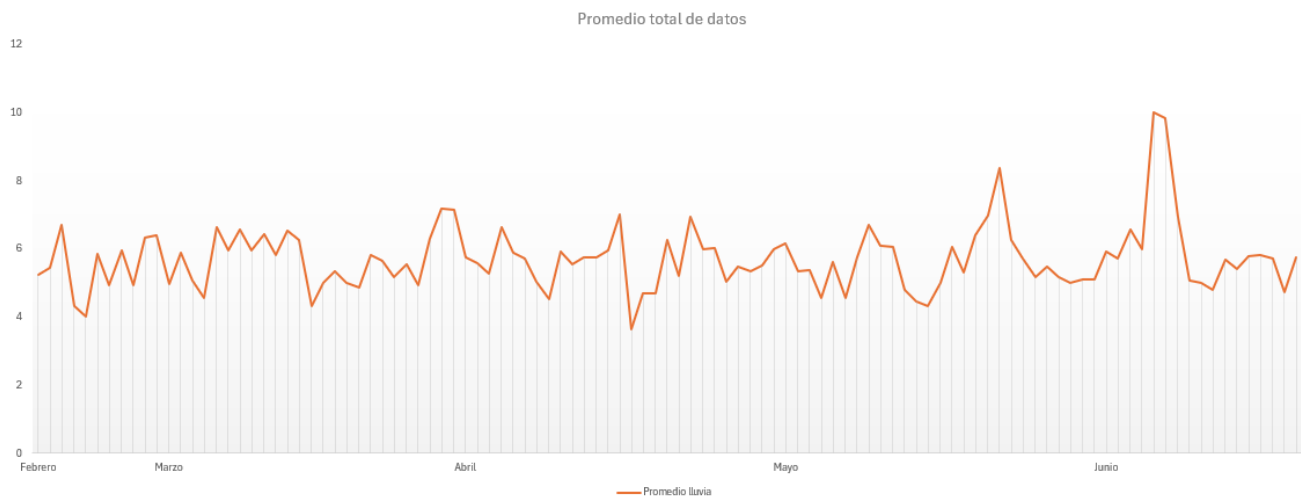
Fuente: Autoría.

La precipitación media anual dentro de toda la parroquia y sus alrededores es entre 1.100 y 3.500 milímetros cúbicos al año. Tomando en cuenta que los meses con más presencia de lluvia es mayo, junio y julio. (*PDOT-Cosanga, 2019*)

Las lecturas tomadas con el sensor de precipitación coinciden con las precipitaciones en los meses de lluvia más alto indicándonos picos elevados y de igual manera picos de bajada. Esto no quiere decir que en los meses restantes no habrá presencia de lluvia, Cosanga al ser una zona tropical megatérmico húmedo, lo que indica que la mayoría del año tendrá presencia de lluvias.

Figura 89.

Promedio de nivel de lluvia de Febrero-Junio 2025.

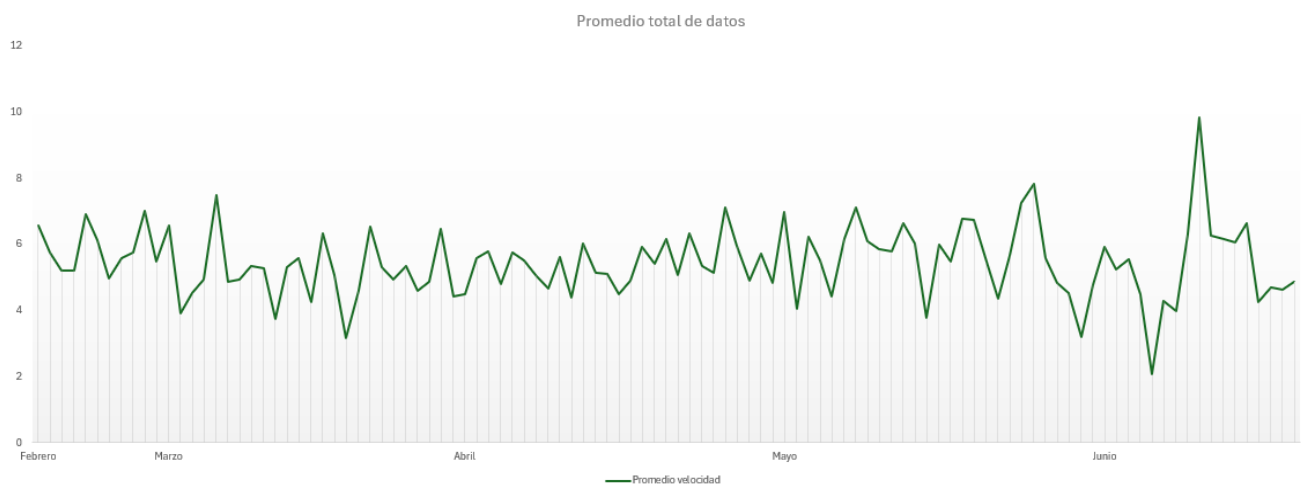


Fuente: Autoría.

En la Figura 90, se observa los datos del sensor de velocidad del río, estos datos pueden aumentar o disminuir debido a varios factores. El factor principal es la presencia de lluvias, en especial los meses con más presencia de lluvia como son mayo, junio y julio. Por otra parte, la velocidad no solamente se basa en la presencia de lluvia, sino también por las pendientes que puede tener el río y también por los derrumbes. Estos ocasionan un impulso para que tenga una mayor velocidad el río y genere picos más elevados en los meses con mayor presencia de lluvia.

Figura 90.

Promedio de nivel de velocidad de Febrero-Junio 2025.



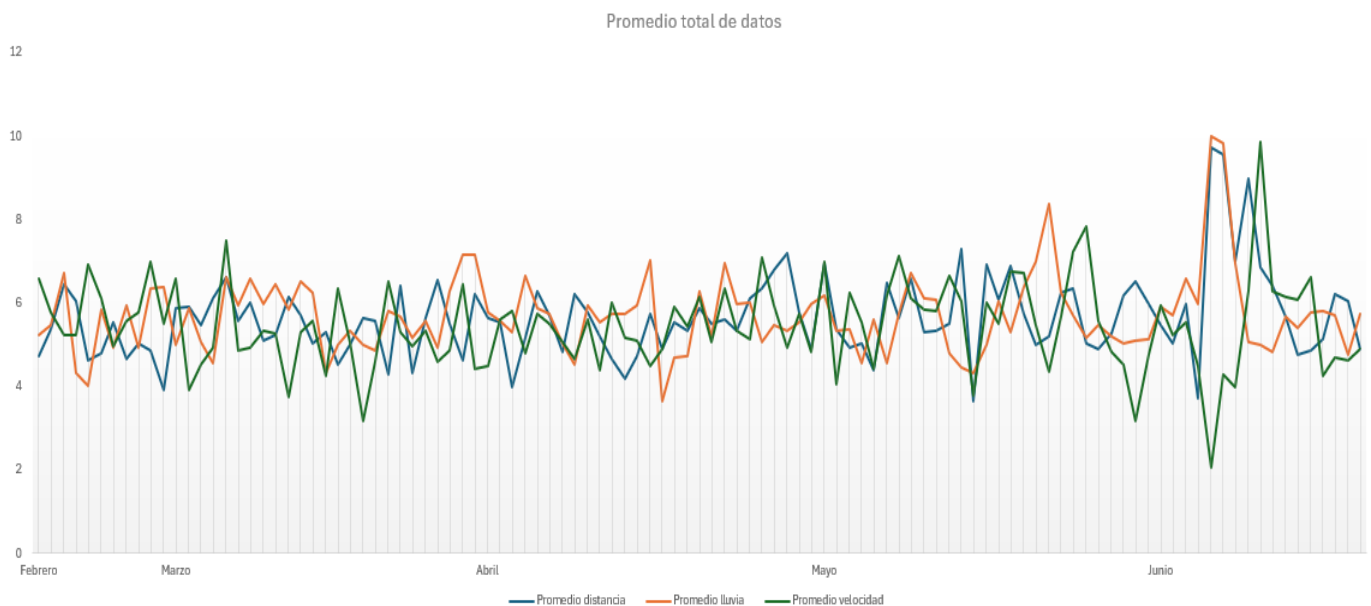
Fuente: Autoría

En la Figura 91, se observa las lecturas de los 3 sensores de distancia, precipitación y velocidad. Se unificaron las lecturas con el fin de identificar coincidencias entre los picos de lecturas altos y bajos registrados.

Los datos permiten visualizar de manera general, el comportamiento del río dentro de la parroquia. Es fundamental ubicar en una zona específica los sensores para lograr obtener datos que coincidan según las temporadas de mayor presencia de lluvia. Por otro lado, se observa la coincidencia de las lecturas en los meses con más presencia de lluvia, indicando lecturas altas y, de igual manera, los meses anteriores se mantiene con lecturas de picos considerables, pero no son mayores que las lecturas de mayor precipitación.

Figura 91.

Promedio de total de datos de Febrero-Junio 2025.



Fuente: Autoría.

4.7. Verificación y presentación del sistema de monitoreo

Al finalizar las pruebas de los dispositivos electrónicos, como el microcontrolador, sensores, sitio web y lógica difusa, se procede a verificar el desempeño en conjunto de cada uno de los nodos que integran el sistema de monitoreo.

4.7.1. Recepción y visualización de datos en sitio web

Dentro de este apartado, mediante el uso del protocolo MQTT se realiza el proceso de enviar datos hacia el sitio web desarrollado únicamente para el sistema de monitoreo. En la Figura 92, se logra observar los datos recolectados como variables de entrada y procesados mediante lógica difusa, de igual manera, se indica el porcentaje de nivel de riesgo, que actúa como la variable de salida. Además, se logra visualizar la gráfica de los datos procesados con su respectivo valor de riesgo.

Figura 92.

Sitio web de visualización de datos.



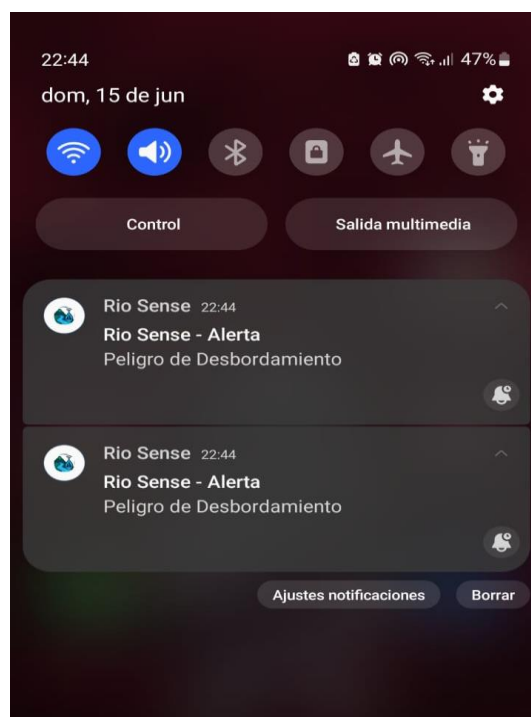
Fuente: Autoría.

4.7.2. Verificación de Notificaciones en Aplicación

La aplicación desarrollada específicamente para el sistema permite la visualización de los datos procesados, indica la gráfica de los datos y el porcentaje de riesgo. En caso de enviarse una alerta, la aplicación activa una notificación dentro del dispositivo móvil e indica el tipo de alerta. Dentro este sistema, solamente se enviará una notificación de alerta indicando que los niveles de los sensores alcanzaron su nivel máximo establecido, indicando así una alerta a todos los dispositivos conectados y la sirena.

Figura 93.

Verificación de notificaciones de alerta de app móvil.

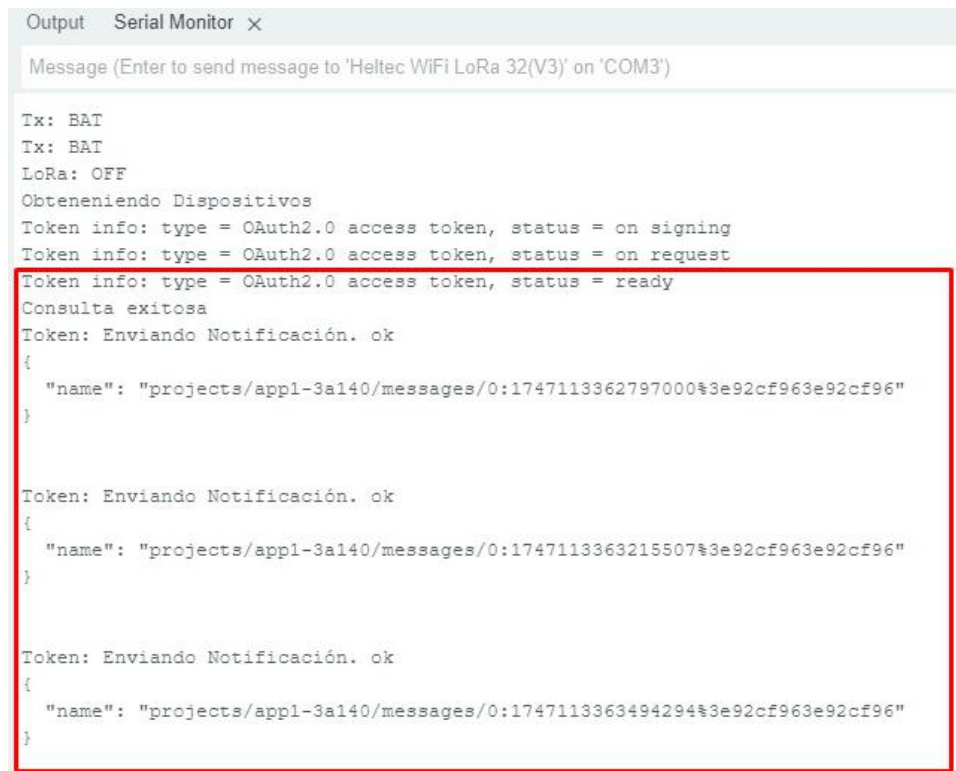


Fuente: Autoría.

Dentro de la Figura 94, se puede observar dentro del monitor serial los tokens correspondientes de cada dispositivo móvil conectado al sistema. El token se asigna al momento de enlazarse con Firebase para lograr visualizar los datos del sistema.

Figura 94.

Verificación de notificación dentro del monitor serial.



```
Output Serial Monitor x
Message (Enter to send message to 'Heltec WiFi LoRa 32(V3)' on 'COM3')

Tx: BAT
Tx: BAT
LoRa: OFF
Obteniendo Dispositivos
Token info: type = OAuth2.0 access token, status = on signing
Token info: type = OAuth2.0 access token, status = on request
Token info: type = OAuth2.0 access token, status = ready
Consulta exitosa
Token: Enviando Notificación. ok
{
  "name": "projects/appl-3a140/messages/0:1747113362797000%3e92cf963e92cf96"
}

Token: Enviando Notificación. ok
{
  "name": "projects/appl-3a140/messages/0:1747113363215507%3e92cf963e92cf96"
}

Token: Enviando Notificación. ok
{
  "name": "projects/appl-3a140/messages/0:1747113363494294%3e92cf963e92cf96"
}
```

Fuente: Autoría.

Dentro de la Figura 95, se logra visualizar de manera general lo que se implementó dentro del sistema para que se logre observar la ubicación, porcentaje de riesgo, porcentaje de batería del nodo coordinador, los datos procesados y su gráfica.

Figura 95.

Verificación de datos procesados en app móvil.



Fuente: Autoría.

4.8. Rendimiento del sistema

En este apartado del sistema de monitoreo, se presentan las reglas difusas que se establecieron previamente en la sección 3.6.7.5. A partir de las reglas implementadas, se determina la matriz de confusión para el sistema y así lograr evaluar el rendimiento del sistema.

4.8.1. Matriz de confusión

La matriz de confusión son una de las diferentes métricas de evaluación que miden el rendimiento de un modelo de clasificación. (Hand & Christen, 2018). La matriz de confusión contiene los valores de decisiones sobre las acciones de notificar según los valores recolectados del nodo sensor.

La Matriz de Confusión es una de las métricas más intuitivas y sencillas que se utiliza para encontrar la precisión y exactitud del modelo. Se utiliza para el problema de clasificación donde la salida puede ser de dos o más tipos de clases.(Gonzalez, 2019)

Tabla 36.

Modelo de representación de matriz de confusión.

PREDICCIÓN			
		Si Notifica	No Notifica
Valores Reales	Si Notifica	VP	FN
	No Notifica	FP	VN

Fuente: Autoría.

El número de predicciones representadas en la matriz de confusión corresponden al número de reglas borrosas establecidas en la sección 3.7.5, siendo un total de 27 reglas, como se puede observar en la Tabla 37.

Tabla 37.

Predicciones para la matriz de confusión.

Número	Inferencia	Decisión	Verdadero	Falso
1	Cuando P es 800, N es 50cm y V es 10km/h	No Notifica	X	
2	Cuando P es 800, N es 50cm y V es 25km/h	No Notifica	X	
3	Cuando P es 800, N es 50cm y V es 40km/h	No Notifica	X	

4	Cuando P es 250, N es 120cm y V es 5km/h	No notifica	X
5	Cuando P es 250, N es 180 y V es 8km/h	No notifica	X
6	Cuando P es 1500, N es 182 y V es 38km/h	No Notifica	X
7	Cuando P es 2113, N es 190 y V es 5km/h	No Notifica	X
8	Cuando P 600, N es 210 y V es 3km/h	No notifica	X
9	Cuando P es 2000, N es 190 y V es 40km/h	No Notifica	X
10	Cuando P es 500, N es 120 y V es 13km/h	No Notifica	X
11	Cuando P es 700, N es 150 y V 11km/h	No Notifica	X
12	Cuando P es 600, N es 120 y V es 35km/h	No Notifica	X
13	Cuando P es 1000, N es 120cm y V es 10km/h	No Notifica	X
14	Cuando P es 800, N es 180 cm y V es 15km/h	No Notifica	X
15	Cuando P es 1500, N es 130 cm y V es 35km/h	Notifica	X

16	Cuando P es 1000, N es 180 y V es 15km/h	No notifica	X
17	Cuando P es 2000, N es 150 y V es 20km/h	No notifica	X
18	Cuando P es 2000, N es 160cm y V es 25km/h	No Notifica	X
19	Cuando P es 500, N es 80 y V es 8km/h	No Notifica	X
20	Cuando P es 800, N es 90cm y V es 20km/h	No Notifica	X
21	Cuando P es 500, N es 80cm y V es 35km/h	No notifica	X
22	Cuando P es 650, N es 50 y V es 10km/h	No notifica	X
23	Cuando P es 1000, N es 80 cm y V es 18km/h	Notifica	X
24	Cuando P es 1000, N es 80cm y V es 35km/h	No Notifica	X
25	Cuando P es 3000, N es 100 y V es 10km/h	No Notifica	X
26	Cunado P es 4000, N es 80 y V es 18km/h	Notifica	X
27	Cuando P es 4000, N es 60cm y ves 35km/h	Notifica	X

Fuente: Autoría.

Gracias a los resultados obtenidos dentro de la matriz de confusión se puede realizar los cálculos de precisión del controlador difuso del sistema desarrollado.

Tabla 38.

Modelo de representación de matriz de confusión del sistema.

		PREDICCIÓN	
		Si Notifica	No Notifica
Valores reales	Si Notifica	26	1
	No Notifica	0	27

Fuente: Autoría.

Con los resultados obtenidos se procede a utilizar la ecuación de exactitud y precisión, estas ecuaciones surgen de la necesidad de evaluar el rendimiento de un modelo de clasificación.

$$exactitud = \frac{VP + VN}{VP + VN + FN + FP} = \frac{26 + 27}{26 + 1 + 0 + 27} = 0.96 = 96\%$$

$$precisión = \frac{VP}{VP + FP} = \frac{26}{26 + 0} = 1 = 100\%$$

4.9. Medición de Tiempo de Respuesta

El tiempo de respuesta ha sido evaluado mediante cálculos basados en la velocidad del río, tomando en cuenta también los datos de los sensores de nivel del agua en el río y precipitación. La distancia en donde se encuentran los nodos hacia la Parroquia de Cosanga es de aproximadamente 16.45km. La velocidad del río fue medida en 25km/h durante su época de riesgo alto y entre 10 a 15km/h en la época más baja.

A continuación, se establece el cálculo matemático establecido para obtener el tiempo para evacuar la parroquia en caso de una alerta de posible desbordamiento.

Formula general para convertir km/h a m/s.

$$m/s = \frac{\frac{km}{h} * 1000}{3600} = \frac{25 \times 1000}{3600} = \frac{25000}{3600} = 6.94$$

$$Tiempo = \frac{16.450m}{6.94m/s} = 2.370seg.$$

$$Tiempo = \frac{2.370s}{60s} = 39.5 minutos$$

Con el resultado obtenido, se estima que el agua tardaría aproximadamente 39.5 minutos en llegar desde los nodos hacia la parroquia de Cosanga en caso de de desbordamiento. Para obtener un mejor resultado de tiempo de evacuación, se retira un período de 7 minutos ya que ese tiempo corresponde al nodo sensor para obtener datos reales y procesarlos. Al realizar este ajuste, se obtiene un tiempo aproximado de 32.5 minutos para evacuar y las personas permanezcan en un lugar seguro.

$$Tiempo total de evacuación = 32.5 minutos$$

4.10. Socialización del sistema en la comunidad

Para dar a conocer cómo se desarrolló el sistema, sus funciones, tiempo de evacuación, elementos utilizados y otros aspectos importantes, todo esto se presentó a las autoridades del GADPR Cosanga. Durante la presentación, se detalló todo el proceso que se aplicó al sistema, desde su problemática hasta la aplicación del prototipo dentro de la

parroquia. Esto con el fin de dar a conocer todas las etapas del proyecto y la funcionalidad hacia la parroquia.

Figura 96.

Socialización con los vocales del GADPR Cosanga.



Fuente: Autoría.

Dentro de la presentación se dio a conocer todos los elementos que fueron utilizados, permitiendo a todo el personal observar físicamente los nodos y como está armado cada uno. De igual manera, se dio a conocer el costo de todo el sistema y la viabilidad de dicho sistema dentro de la parroquia. Finalmente, se dio la respuesta a varias preguntas que tenía el personal y futuros proyectos que se pueden aplicar dentro de la parroquia.

Figura 97.

Presentación de Sistema de monitoreo.



Fuente: Autoría.

4.11. Análisis Costo/Beneficio del Sistema

Para lograr el desarrollo del sistema de monitoreo utilizando lógica difusa para prever el desbordamiento del río Cosanga, fue necesario adquirir varios componentes entre hardware y software.

4.11.1. Costo de Hardware

Dentro de este apartado se detalla los elementos que componen el hardware del sistema de monitoreo. En la Tabla 38, se logra observar el costo del nodo sensor. Cabe recalcar que dentro de los precios también se agrega el nodo coordinador en el WiFi LoRa 32 ya que dentro de ese nodo únicamente se utilizó dicho microcontrolador.

Tabla 39.

Costo de Hardware de nodo sensor y coordinador.

Elemento	Cantidad	Precio Unitario	Precio Total
WiFi LoRa 32	2	\$ 60.00	\$ 120.00
Sensor de precipitación FC-37	1	\$ 4.00	\$ 4.00
Sensor de nivel JSN-SR04T	1	\$ 21.50	\$ 21.50
Sensor de velocidad E18-D80NK	1	\$ 12.50	\$ 12.50
Batería modelo PK LP803860	1	\$ 8.50	\$ 8.50
Panel Solar 6v 200mA	1	\$ 9.50	\$ 9.50
Estructura física de Nodo sensor	1	\$ 10.00	\$10.00
TOTAL		\$ 113.50	\$ 186.00

Fuente: Autoría.

Dentro de la Tabla 40, se representa el costo del nodo actuador, el cual se activa cuando existe un nivel de alerta muy alto.

Tabla 40.

Costo de Hardware de alerta.

Elemento	Cantidad	Precio Unitario
Sirena 110V	1	\$ 23.00
Modulo Esp32	1	\$ 14.00
Relé de estado sólido	1	\$ 9.50
Estructura física de Nodo Actuador	1	\$10.00
TOTAL		\$ 56.50

Fuente: Autoría.

4.11.2. Costo de Software

Dentro de la Tabla 41 se detallan las plataformas de software que fueron utilizadas para el desarrollo del sistema. Cada plataforma que fue seleccionada para el sistema es de uso gratuito, esto permite que el desarrollo de un sistema similar no genere gastos por licencia. Esto promueve a desarrollar soluciones económicas y accesibles ante recursos limitados.

Tabla 41.

Costo de Software para el sistema de monitoreo.

Elemento	Descripción	Costo
Google Colab	Plataforma de computación en la nube utilizado para el entrenamiento de los datos ofreciendo procesamiento de alta capacidad.	\$ 0.00
Geogebra	Herramienta matemática para modelar las gráficas de manera dinámica las variables de entrada de los sensores.	\$ 0.00
Angular	Software utilizado para crear el sitio web en donde se visualizará los datos procesados, gráficas, configuración del sistema y visualización de datos almacenados.	\$ 0.00
Ionic	Software utilizado para desarrollar la aplicación móvil del sistema.	\$ 0.00
IDE Arduino	Plataforma utilizada para programar de manera general el sistema de monitoreo que sea compatible con el microcontrolador WiFi LoRa 32.	\$ 0.00
Radio Mobile	Herramienta utilizada para realizar la simulación de la zona de los dos nodos y analizar sus resultados.	\$ 0.00
Firebase	Plataforma en donde permite observar los datos en tiempo real, almacenar datos procesados y notificaciones de alerta.	\$ 0.00
Fusión 360	Plataforma utilizada para elaborar el diseño en 3D e imprimir la estructura de los nodos y el sistema de alerta.	\$ 30.00
TOTAL		\$ 30.00

Fuente: Autoría.

4.11.3. Costo de Ingeniería

Actualmente, la junta parroquial de Cosanga no cuenta con información sobre los costos que maneja un tipo sistema de monitoreo o la información correspondiente a lo que implica desarrollar un sistema de monitoreo. Debido a esto, se llevó a cabo colocar el costo de hardware, software y el proceso de desarrollo que lleva armar el sistema. La información obtenida proviene del uso de un sitio web que funciona como fuente informativa de las medias salariales que existe dentro del país relacionado a la parte de programación. Hay que tomar en cuenta que en Ecuador el horario laboral de los programadores es de 8 horas diarias durante los 5 días de la semana laboral.(Computrabajo, 2025)

De acuerdo con los salarios sectoriales obtenidos en el documento del IESS, el salario mínimo que tarte temas relacionados a este proyecto es de \$495,20. (IESS, 2024)

Tabla 42.

Costo de ingeniería.

Descripción	#Días	Costo
Diseño de estructura de nodos e impresión en 3D	1	\$ 30.00
Simulación de Red en Radio Mobile	1	\$ 10.00
Simulación en herramienta matemática Geogebra	1	\$ 10.00
Armado de elementos en cada nodo	1	\$ 10.00
Programación en Arduino	8	\$ 300.00
Programación de sitio web y app	4	\$ 120.00

Programación de entrenamiento	2	\$ 60.00
Instalación del sistema	1	\$ 25.00
Mantenimiento del sistema	2	\$ 50.00
TOTAL		\$ 615.00

Fuente: Autoría.

Al concluir el costo de lo que lleva todo el sistema de monitoreo, en la Tabla 43 se especifica el costo total de hardware, software e implementación.

Tabla 43.

Costo general del sistema.

Costo general del Sistema		
Tipo de Costo	Costo de Mercado	Costo Actual
Costo de Hardware	\$ 186.00	\$ 186.00
Costo de Sistema de Alarma	\$ 56.50	\$ 56.50
Costo de Software	\$ 30.00	\$ 30.00
Costo de Ingeniería	\$ 615.00	\$ 0.00
TOTAL	\$ 887.50	\$ 272.50

Fuente: Autoría.

Desde el ámbito económico, el sistema implementado contiene un presupuesto rentable frente a un costo que se manejaría para este tipo de sistemas de monitoreo. La rentabilidad se logra al optimizar recursos, principalmente en la parte de programación, contratar servicios de plataformas IoT o colocar equipos que contengan altos precios. Se ha optado por hardware y software que resulte ideal para la Parroquia de Cosanga. Además, gracias a este sistema se obtiene el beneficio de evitar pérdidas agrícolas, daños a infraestructuras, daños en viviendas y costos por vigilar el sistema de monitoreo de

manera manual. De manera general, el sistema contiene un gran beneficio con una inversión mínima, sin requerir elementos con precios elevados.

4.12. Discusión

Los resultados del proyecto indican que el sistema de monitoreo, basado en sensores IoT, tecnología de comunicación inalámbrica y lógica difusa, es capaz de dar a conocer cambios críticos en los niveles del río Cosanga y enviar alertas tempranas utilizando los medios de comunicación como notificaciones y la alarma. La distancia de la zona en donde se encuentran los Nodos y la parroquia es aproximadamente 17km. Al demostrar la incorporación de la lógica difusa permite entender datos imprecisos o variables, formando así estimaciones más precisas que los sistemas tradicionales de umbrales fijas. Un estudio similar como el de (Navarrete, 2024), quien plantea el uso de una red de sensores remotos junto con el uso de comunicación inalámbrica para obtener respuestas oportunas ante posibles inundaciones. Por otro lado, el sistema desarrollado en este proyecto utiliza la tecnología LoRa con una frecuencia de 915 MHz y contiene un enfoque autónomo que se adapta a las variaciones del río, logrando así reducir falsas alertas en comparación con sistemas tradicionales.

Es importante tomar en cuenta que, estos sistemas pueden tener limitaciones, como la necesidad de energía solar para mantener activo los nodos recolectores de datos en épocas en donde la lluvia es muy prolongada, la vida útil de la batería y calibrar correctamente los sensores para evitar datos imprecisos. En la parte práctica, el sistema aporta al área de monitoreo ambiental en zonas rurales, demostrando que la lógica difusa es una herramienta capaz de adaptarse ante desastres en comunidades vulnerables. Por otro lado, los resultados del sistema pueden servir para implementar en otras parroquias con condiciones geográficas similares. Para trabajos futuros, se propone investigar la

integración de algoritmos de aprendizaje automático que ayuden a perfeccionar los rangos difusos mediante datos históricos. Esto implica que, al obtener registros anteriores, el sistema de monitoreo podría adaptar de manera más precisa los valores que indican los niveles de riesgo y reducir posibles errores de alertas. Seguidamente, el sistema debe contener paneles solares de mayor capacidad y dispositivos electrónicos de mayor calidad, con el fin de garantizar el funcionamiento a largo plazo e incluso en épocas de lluvia muy prolongadas. Por último, se sugiere extender la red de sensores para incluir otras variables hidrometeorológicas, lo cual permitiría una predicción más precisa del comportamiento del río.

Conclusiones y Recomendaciones

Conclusiones

Al finalizar el desarrollo del prototipo del sistema de monitoreo con lógica difusa para prever el desbordamiento del río Cosanga, se comprobó el funcionamiento de todos los componentes electrónicos y sus estructuras físicas. El diseño físico que contiene el sistema es implementado para recolectar información y proteger el rendimiento de cada componente electrónico.

Durante el proceso de pruebas de funcionamiento del sistema de monitoreo, se llevaron a cabo dos escenarios de pruebas, en el primero escenario se utilizaron componentes electrónicos básicos para observar el comportamiento de cada elemento y corregir fallas o actualizaciones que requiere el sistema. En el segundo escenario, se implementó nuevos componentes electrónicos que van acorde al proyecto requerido, de igual manera en este escenario se realiza depuraciones del sistema.

El sistema utiliza tecnología LoRa, esta tecnología permite establecer comunicaciones inalámbricas a más de 1km, considerando que los nodos dentro del sistema se encuentran a 62 m con una frecuencia de 915 MHz, esto indica que, el nodo sensor puede cambiar su ubicación a una zona más elevada, con la finalidad de obtener más tiempo de detección. Esta tecnología es muy utilizada en redes de sensores inalámbricos debido a su capacidad de interconexión de dispositivos a largas distancias ya sea con obstáculos o una línea de vista libre, esta tecnología resulta ser ideal para este tipo de proyectos y es una gran ventaja respecto a otras tecnologías de comunicación inalámbrica.

Mediante las 27 reglas difusas, descritas a partir del conocimiento de la comunidad y la conducta del río, se logró una exactitud del 96% y una precisión de 100% según el modelo de la matriz de confusión. De esa manera, se valida los valores de decisiones que contiene el sistema de monitoreo.

La toma de datos en intervalos de tiempo permite al sistema detectar con mayor precisión los cambios que ocurre en el río, esto gracias a que el sistema funciona de manera automática. Con el mayor número de lecturas ayuda a interpretar correctamente las variaciones de las lecturas y evitar falsas alarmas, Por esta razón, es fundamental contar con lecturas frecuentes para que no pase por alto niveles peligrosos.

El consumo energético del sistema determinó un aproximado de 56 mA cuando el sistema está totalmente activo, lo que permitió que una batería de 2600 mAh proporcione 46 horas de autonomía. Además, el sistema contiene un panel solar que permite cargar el 100% en 16 horas de exposición al sol. Tomando en cuenta que, en la noche se consume aproximadamente el 25% de la batería, el panel solar en el día recolectará el 25% consumido e incluso genera un margen adicional para que el sistema se mantenga activo.

Recomendaciones

Al manipular los dispositivos, debe ser de manera cuidadosa. Es recomendable apagar los nodos en caso de realizar un cambio en las conexiones de los componentes electrónicos, con el fin de evitar posibles daños tanto a los elementos electrónicos como también a los microcontroladores, esta medida ayuda a evitar gastos innecesarios.

Al realizar mejoras al sistema o trabajos relacionados al tema, es recomendable optimizar recursos económicos para dar a conocer la viabilidad del sistema y la accesibilidad en cualquier lugar que se implemente el sistema.

Para garantizar que la comunicación inalámbrica se mantenga estable y segura, es adecuado utilizar los parámetros predeterminados de fábrica de la tecnología LoRa que contienen los microcontroladores. Esta configuración se encarga de evitar crear valores de frecuencia u otros parámetros para la comunicación inalámbrica.

Es muy importante utilizar buenas prácticas de programación, con el fin de realizar modificaciones como herramientas de gestión, administración y visualización del prototipo. Con el uso adecuado de la programación, se impulsa en beneficio del sistema o modificaciones en proyectos relacionados al tema.

Se recomienda establecer el mantenimiento preventivo del sistema al menos dos veces al año, con el fin de asegurar el funcionamiento adecuado del sistema de monitoreo. También, es importante realizar cambios de dispositivos electrónicos que presenten deterioro, ya que estos elementos son fundamentales para que el sistema recolecte información.

Se recomienda desarrollar a futuro un sistema de monitoreo más amplio en donde no solo se contemple el comportamiento del río, sino que también se tome en cuenta un análisis de las zonas geográficas con alto índice de deslizamientos de tierra. Esta

integración permite obtener un análisis más completo de eventos relacionados a los riesgos producidos por lluvias intensas, incorporando sensores de humedad, estaciones meteorológicas, imágenes satelitales o drones que se encarguen de detectar áreas inestables y propensas a deslizamientos. Al integrar toda la información mediante modelos predictivos, se lograría anticipar crecidas peligrosas del río y deslizamientos de tierra que afecten comunidades y caminos cercanos al cauce del río. De esta manera, se fortalecería el enfoque preventivo ante emergencias y se optimiza la toma de decisiones en comunidades vulnerables en entornos rurales.

REFERENCIAS

- Alaminos, A. (2023). *Introducción a la teoría de conjuntos difusos y sus aplicaciones en investigación social e IA*.
- ALBA, E. (2019). *UNIVERSIDAD NACIONAL DEL CALLAO ESCUELA DE POSGRADO TESIS*.
- Andrade, Á., & Peña, C. (2020). *P U B L I C A C I Ó N A N U A L*.
- ARCOTEL. (2017). *Bandas de frecuencia en Ecuador*. https://www.arcotel.gob.ec/wp-content/uploads/2017/09/Informe-Audiencias-P%C3%BAblicas_.pdf
- ARCOTEL. (2022). *REFORMAS NORMATIVAS PARA LA IMPLEMENTACIÓN Y APLICACIÓN DEL NUEVO SERVICIO DE CONECTIVIDAD DE INTERNET DE LOS OBJETOS IoT*. https://www.arcotel.gob.ec/wp-content/uploads/2023/05/informe_it-crds-gr-2022-046.pdf
- Arduino. (2018). *Arduino*. <https://www.arduino.cc/en/Guide/Introduction>
- Becolve. (2020). *LPWAN | Becolve Digital*. <https://becolve.com/blog/lpwan-que-son-y-para-que-se-utilizan/>
- Camberos, C., & Heyssen, F. (2020). *DESARROLLO DE UNA RED PUNTO A PUNTO CON TECNOLOGÍA LORA Y SERVIDOR WEB*.
- Carvajal, L., Colina, E., & Fernández, N. (2010). *Sistema Difuso Tipo Mamdani para la Determinación Genérica de la Calidad del Agua. Mandani Fuzzy System for General Water Quality Determination*. 8(1), 2010.
- Computrabajo. (2025). *Salario de Desarrollador programador en Ecuador 2025 | Computrabajo Ecuador*. <https://ec.computrabajo.com/salarios/desarrollador-programador>
- Díaz, J., Cuzme, F., Michilena, J., Domínguez, M., & Suárez, L. (2024). *Intelligent Monitoring System for Fry Rearing Using Fuzzy Logic. Lecture Notes in Networks and Systems, 871 LNNS*, 282–292. https://doi.org/10.1007/978-3-031-52090-7_27
- Domingo Pérez. (2024). *Seguridad y Alta Disponibilidad | TIC – IES DOMINGO PÉREZ MINIK*. <https://www3.gobiernodecanarias.org/medusa/ecoblog/flopmarl/seguridad-y-alta-disponibilidad/>
- El Universo. (2023). *Inundaciones Ecuador durante los últimos cinco años | Informes | Noticias | El Universo*. <https://www.eluniverso.com/noticias/2020/07/12/nota/7903011/inundaciones-ecuador-2020-agua-lluvias-cambio-climatico/>
- GoDaddy. (2021). *Qué es la interfaz de usuario (UI): Conceptos y mejores prácticas*. <https://www.godaddy.com/resources/latam/desarrollo/que-es-interfaz-usuario-ui>
- Goncalves, M. (2024). *hiberus*. <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>

- Gonzalez, L. (2019, May 17). *Matriz de Confusión - Aprende IA*. <https://aprendeia.com/matriz-de-confusion-machine-learning/>
- Gordillo, C. (2021). *“RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO PARA MONITOREO AMBIENTAL EN INVERNADEROS.”*
- Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3), 539–547. <https://doi.org/10.1007/S11222-017-9746-6>
- IEEE. (2018). *IEEE/ISO/IEC 29148-2018*. <https://standards.ieee.org/ieee/29148/6937/>
- IESS. (2024). *Salarios sectoriales*. https://www.iessec.gob.ec/documents/10162/25743050/SALARIOS_SECTORIALES_2024.pdf
- Kaspersky. (2025). *Internet y la privacidad personal: cómo protegerte y proteger tus datos*. https://latam.kaspersky.com/resource-center/threats/internet-and-individual-privacy-protection?srltid=AfmBOozUiaVN5zKBLiaTVmttHCaww4Alq_21dxkf7xxuHLocNxbMQL
- Kassir, E. E. (2023). *SISTEMAS DE CONTROL DIFUSO*.
- klippa. (2025). *Automatización de Datos: Beneficios, Uso y Softwares*. <https://www.klippa.com/es/blog/informativo/automatizacion-datos/>
- López, V. (2023). *Unidades de Gestión de Riesgos cantonales de Napo se capacitaron en el taller sobre flujo de información durante emergencias y desastres. – Secretaría de Gestión de Riesgos*. <https://www.gestionderiesgos.gob.ec/unidades-de-gestion-de-riesgos-cantones-de-napo-se-capacitaron-en-el-taller-sobre-flujo-de-informacion-durante-emergencias-y-desastres/>
- LoRa, A. (2020). *LoRaWAN*. https://lora-alliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf
- Marino, D. M. (2023). *LoRaWAN*. <https://www.innovaciondigital360.com/iot/lorawan-que-es-para-que-sirve-por-que-merece-la-pena-utilizarla/>
- Marmolejo, R. (2019). *Microcontrolador - qué es y para que sirve - HeTPro-Tutoriales*. https://hetpro-store.com/TUTORIALES/microcontrolador/?srltid=AfmBOoropAiqafsgRmtDAwJ3ZDt11FnkMAOs1C1D4CYNFu-nV_1e-YnU
- Mechatronics, N. (2023). *Sensor Ultrasonido JSN-SR04T*. <https://naylorlampmechatronics.com/sensores-proximidad/326-sensor-ultrasonido-jsn-sr04t.html>
- Meteoblue. (2023). *Datos climáticos y meteorológicos Cosanga*. https://www.meteoblue.com/es/tiempo/historyclimate/climatemodelled/cosanga_ecuador_3658789
- Ministerio de Turismo. (2014). *Napo, potencia turística de la región amazónica – Ministerio de Turismo*. <https://www.turismo.gob.ec/napo-potencia-turistica-de-la-region-amazonica/>

- Monolitic. (2025). *¿Cómo puede el IoT mejorar la gestión de los recursos energéticos en Smart Buildings?* | Monolitic. <https://www.monolitic.com/como-puede-iot-mejorar-la-gestion-en-smart-building/>
- Moreno, A., Béjar, A.-J., Belanche, L., Cortés, U., Gavalda, R., Manuel, J., López, G.-B., Martín, M., & Sánchez, M. (1994). *Aprendizaje automático*. www.edicionsupc.es
- Morona, S. (2013). *Agenda Zonal ZONA 1-Norte*.
- Naciones Unidas Ecuador. (2024). *Ecuador redobla su compromiso con alertas tempranas para proteger a su población* | Naciones Unidas en Ecuador. <https://ecuador.un.org/es/273023-ecuador-redobla-su-compromiso-con-alertas-tempranas-para-proteger-su-poblaci%C3%B3n>
- Napo Travel. (2022). *Río Cosanga*. <https://www.napo.travel/rio-cosanga/>
- Navarrete, J. (2024). *UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA PROGRAMA: X PREGRADO POSGRADO TITULO POR EL QUE OPTA: Ingeniero en Telecomunicaciones*.
- Oppermann, A. (2024). *What Is the V-Model? (Definition, Examples)* | Built In. <https://builtin.com/software-engineering-perspectives/v-model>
- Oracle. (2025). *¿Qué es JSON?* | Oracle España. <https://www.oracle.com/es/database/what-is-json/>
- Oracle MX. (2025). *¿Qué es una base de datos?* | Oracle México. <https://www.oracle.com/mx/database/what-is-database/>
- PDOT-Cosanga. (2019). <https://sil.napo.gob.ec/wp-content/uploads/2022/09/PDOT-Cosanga.pdf>
- Perez, I., & Leon, B. (2007). *Logica difusa para principiantes* - Google Libros. 173. <https://books.google.com.co/books?id=iCiyyCFZuIC&pg=PA12&lpg=PA12&dq=La+I%C3%B3gica+difusa+provee+una+herramienta+para+preservar+el+concepto+de+vaguedad+en+vez+de+eliminarlo+mediante+la+imposici%C3%B3n+arbitraria+de+sentencias+ciertas+o+falsas&source=bl&>
- Python. (2023). *Documentación de Python*. <https://docs.python.org/es/3/tutorial/>
- Ramírez, P. (2023). *UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA*.
- Rouhiainen, L. (2018). *INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO INTELIGENCIA ARTIFICIAL*. www.planetadelibros.com
- Salazar, J. (2019). *REDES INALÁMBRICAS*. <http://www.techpedia.eu>
- Salazar, J., & Silvestre, S. (2016). *INTERNET DE LAS COSAS*. <http://www.techpedia.eu>
- Salazar, L., Cuzme, F., Maya, E., Domínguez, M., & Zambrano, M. (2020). Design of an Intelligent Irrigation System Based on Fuzzy Logic. *Communications in Computer and Information Science*, 1194 CCIS, 386–399. https://doi.org/10.1007/978-3-030-42520-3_31
- SNGR. (2016). *Secretaría de Gestión de Riesgos*. <https://www.gestionderiesgos.gob.ec/ecuador-mejor-preparado-para-enfrentar-el-fenomeno-de-el-nino/>

- UNIR. (2023). *Sistemas inteligentes / UNIR*. <https://www.unir.net/ingenieria/revista/sistemas-inteligentes/>
- Varela, A. (2022). *Geografía y clima*. <https://bioweb.bio/fungiweb/GeografiaClima/>
- Vargas, D. (2023). *Java vs Python*. https://www.hostinger.es/tutoriales/java-vs-python?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=9076633&gad_source=1&gclid=CjwKCAiA4smsBhAEEiwAO6DEjbvPVirZzdmPYTB8W12dl8CCykDtpBMf1t27Ne9TUjHj5dPL_OJR7BoCONYQAvD_BwE
- Zimmermann, j. (2001). *Fuzzy Set Theory-and Its Applications, Fourth Edition*. <https://doi.org/10.1007/978-94-010-0646-0>

ANEXOS

ANEXO 1. Documento de aceptación y certificado emitido por el GADPR Cosanga.



**GAD PARROQUIAL
RURAL DE COSANGA**

Oficio N° 0105 -GADPRC-2023-P
Cosanga, 15 de noviembre de 2023

Ing. Jaime Michilena
COORDINADOR DE LA CARRERA DE TELECOMUNICACIONES
UNIVERSIDAD TÉCNICA DEL NORTE
Presente.

De mi consideración:

Hago llegar un afectuoso saludo en nombre de quienes hacemos el *Gobierno Autónomo Descentralizado Parroquial Rural de Cosanga*, deseando el mayor de los éxitos en cada una de las actividades encomendadas.

Por medio del presente informo que el estudiante **ANTHONY MIGUEL PANAMÁ CHICAIZA** con cédula de identidad N° 1501154104, va a realizar el tema de TESIS denominado **"SISTEMA DE MONITOREO CON LÓGICA DIFUSA PARA PREVER AL DESBORDAMIENTO DEL RÍO COSANGA EN LA PARROQUIA DE COSANGA"**, perteneciente al Cantón Quijos, Provincia de Napo.

Con sentimiento de distinguida consideración.

Atentamente,

Ing. Gonzalo Morales Ordoñez, Mgtr.

PRESIDENTE DEL GADPR DE COSANGA

CI: 1500756885

Correo: juntapacosanga15@yahoo.es

Telf: 0998135532 – 063061670

GAD PARROQUIAL RURAL
Adm. 2º
Cosanga
Presidente Ing. Gonzalo Morales

06 3 061 670
juntapacosanga15@yahoo.es
av. Roberto Erazo y vía Tena





GAD PARROQUIAL
RURAL DE COSANGA

Cosanga, 26 de mayo del 2025

CERTIFICADO

Yo, Mgr. Wilmer Gonzalo Morales Ordoñez, con cédula de identidad número 150075688-5, en mi calidad de Presidente del Gobierno Autónomo Descentralizado Parroquial Rural de Cosanga, certifico que el señor Anthony Miguel Panamá Chicaiza, con cédula de identidad número 1501174104, realizó las pruebas de su proyecto de titulación titulado **"Sistema de monitoreo con lógica difusa para prever el desbordamiento del río Cosanga en la parroquia Cosanga"**, cuyos resultados fueron satisfactorios para el uso del GADPR de Cosanga.

Es todo cuanto puedo certificar.



WILMER GONZALO
MORALES ORDOÑEZ

Mgr. Gonzalo Morales O

PRESIDENTE GADPR COSANGA

CI: 1500756885

Correo: juntapacosanga15@yahoo.es

Telf: 0998135532 – 063061670



Gonzalo Morales
PRESIDENTE

ANEXO 2. Cuestionario aplicado al presidente de la junta parroquial.

1. ¿Qué tan importante considera usted el monitoreo del río en la parroquia de Cosanga?
 - ☐ Muy importante
 - ☐ Importante
 - ☐ Poco importante
 - ☐ No es importante
2. ¿En su experiencia, cuál es el mayor impacto de las lluvias intensas en la parroquia?

- ☐ Daños en viviendas
 - ☐ Problemas de transporte
 - ☐ Daños en cultivos
- 3. ¿Qué medidas considera que serían más efectivas para prevenir daños causados por el río?
 - ☐ Sistemas de alerta temprana
 - ☐ Obras de infraestructura
 - ☐ Educación comunitaria
- 4. ¿Le parece útil implementar un sistema de monitoreo que alerte sobre el riesgo de desbordamiento?
 - ☐ Sí
 - ☐ No
 - ☐ No estoy seguro
- 5. ¿Qué medio prefiere para recibir alertas sobre posibles desbordamientos?
 - ☐ Mensajes SMS
 - ☐ Llamadas telefónicas
 - ☐ Radio comunitaria
- 6. ¿Estaría dispuesto a colaborar en actividades de mantenimiento del sistema de monitoreo?
 - ☐ Sí
 - ☐ No
 - ☐ Depende de la actividad
- 7. ¿Cómo describiría los cambios en el nivel o comportamiento del río durante las temporadas de lluvia?
 - ☐ Muy riesgoso
 - ☐ Poco riesgoso
 - ☐ Nada riesgoso
- 8. ¿Qué tan informada considera que está la comunidad sobre los riesgos de desbordamiento del río?
 - ☐ Muy bien informados
 - ☐ Poco informados
 - ☐ No existe ninguna información
- 9. ¿Qué tan confiable considera Ud que un sistema de monitoreo puede trabajar sin la presencia de un operador?
 - ☐ Muy confiable
 - ☐ Poco confiable
 - ☐ Nada confiable

10. ¿Con cuánta anticipación te gustaría recibir una alerta sobre un posible desbordamiento del río?
- ☐ Más de 1 hora
 - ☐ Entre 30 minutos y 1 hora
 - ☐ Menos de 30 minutos
11. ¿Cuánto tiempo estimas que tomaría evacuar tu hogar en caso de emergencia?
- ☐ Menos de 15 minutos
 - ☐ Entre 15 y 30 minutos
 - ☐ Más de 30 minutos
12. ¿Consideras necesario un plan de evacuación comunitario en caso de inundaciones?
- ☐ Sí
 - ☐ No
 - ☐ No estoy seguro
13. ¿Qué otras ideas o sugerencias tiene Ud para mejorar la seguridad y prevenir desbordamientos en la parroquia?
- ☐ Comentario

ANEXO 3. Ficha de requerimientos.

Proyecto de titulación: “Sistema de monitoreo con lógica difusa para prever el desbordamiento del río Cosanga en la parroquia de Cosanga”

Objetivo de análisis: Comprender de manera precisa todos los requisitos que debe incluir el Esquema de Autenticación para garantizar seguridad y alcanzar los objetivos establecidos en el proyecto de investigación.

Lista de Abreviaturas

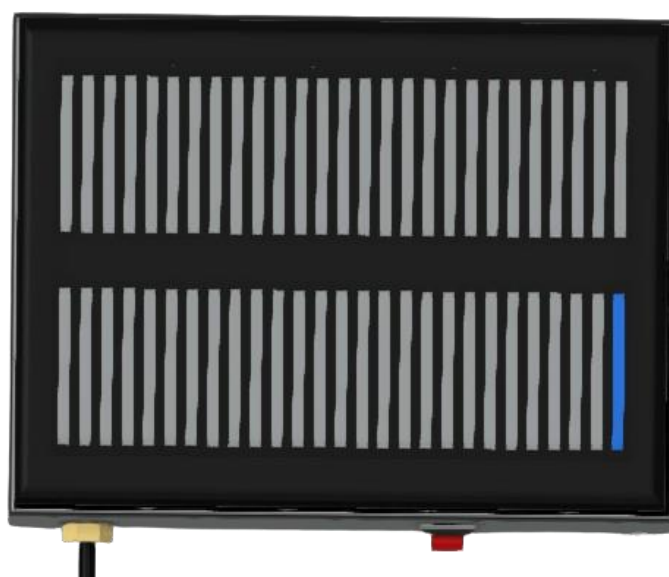
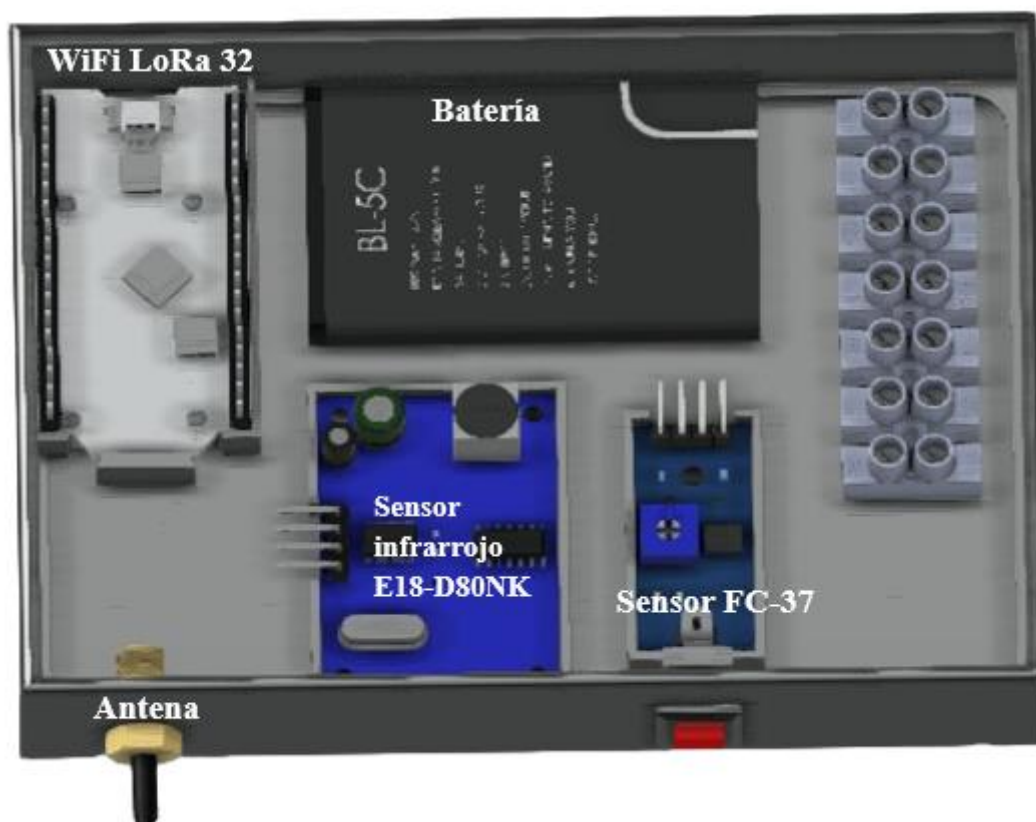
Abreviatura	Descripción
StRS	Requerimientos de Stakeholder
SyRS	Requerimientos del sistema
SRSR	Requerimientos de Software y Hardware

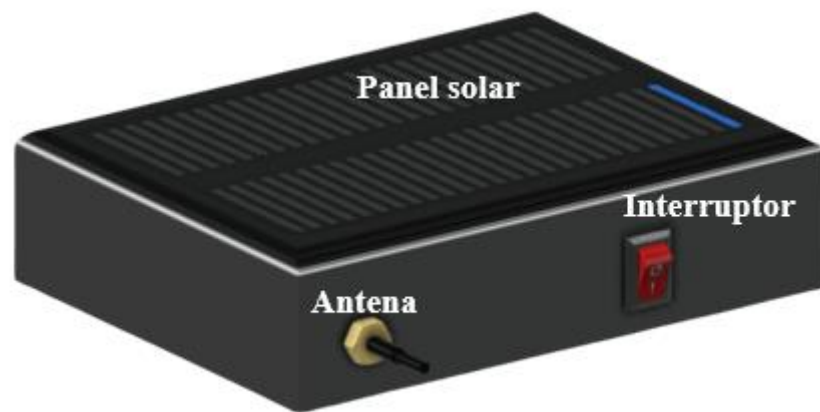
Requerimientos de Stakeholders			
Abreviatura	Requerimiento	Descripción	Prioridad
StRS3	Automatización del sistema para la recolección y transmisión de datos.	La automatización de datos, también denominada automatización del origen de datos consiste en emplear tecnología para recopilar, procesar y administrar información de manera automática. Este proceso minimiza la intervención humana y optimiza la precisión de los datos.(klippa, 2025)	Alta
StRS4	Asegurar la comunicación inalámbrica para la transmisión de datos	Las redes inalámbricas facilitan la conexión de dispositivos remotos sin importar si están a corta o larga distancia. Además, eliminan la necesidad de perforar paredes para instalar cables o conectores. Esta característica ha impulsado su popularidad y rápida expansión. (J. Salazar, 2019)	Alta
StRS8	El sistema de monitoreo debe alertar al usuario por medio del sitio web, mensajería y una alarma comunitaria sobre las condiciones que presenta el río.	Según el representante de las naciones unidas del ecuador, los sistemas de alerta temprana se constituyen en la base para la implementación de acciones de reducción de riesgo de desastres, pues no existe mejor medida que evitar que los desastres ocurran, y no solo esperar que sucedan para dar una adecuada respuesta. (Naciones Unidas Ecuador, 2024)	Alta
StRS10	Contar con una base de datos que permita almacenar los datos y garantizar un acceso inmediato.	La base de datos autogestionada está diseñada para potenciar significativamente estas habilidades. Al automatizar procesos manuales que suelen ser costosos y laboriosos, estas bases de datos permiten que los usuarios empresariales adopten un enfoque más proactivo en el manejo de sus datos. Al otorgarles la capacidad de crear y gestionar bases de datos directamente, los usuarios adquieren mayor control e independencia, sin comprometer los estándares esenciales de seguridad. (Oracle MX, 2025)	Alta

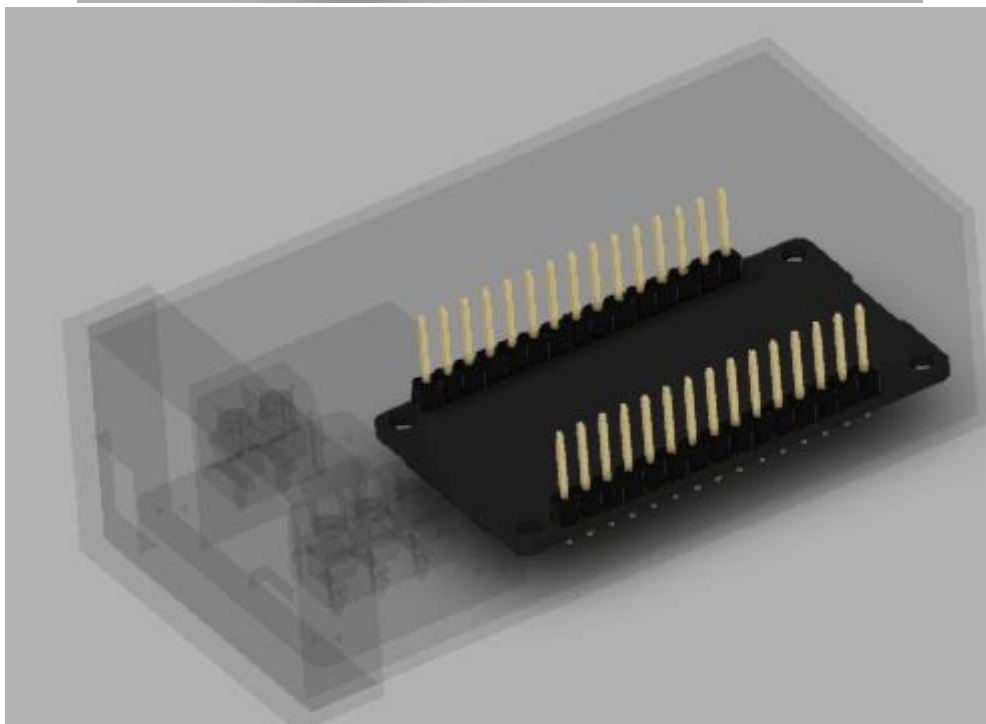
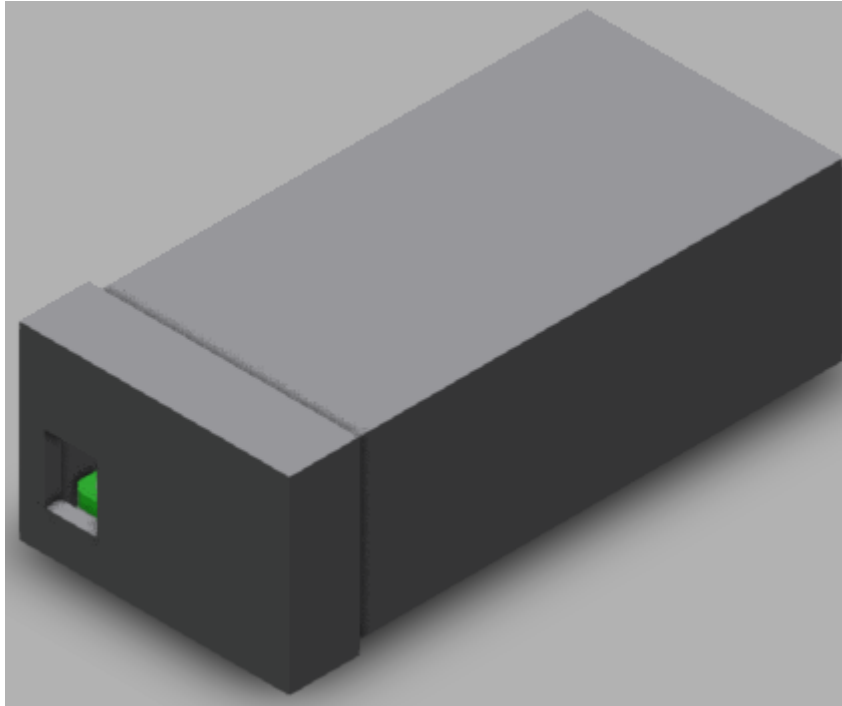
Requerimientos iniciales del sistema			
Abreviatura	Requerimiento	Descripción	Prioridad
SySR9	Gestión equilibrada de recursos energéticos.	Las soluciones de Internet de las Cosas (IoT) representan una excelente oportunidad para mejorar los servicios de gestión energética en infraestructuras o instalaciones que integren sistemas eléctricos y de control de clima.(Monolitic, 2025)	Alta
SySR13	El sistema debe tener una disponibilidad de acceso el mayor tiempo posible del día.	Se refiere a la habilidad de un servicio, conjunto de datos o sistema para estar disponible y ser utilizado por usuarios autorizados en el momento que lo soliciten. Esto implica que la información pueda ser recuperada cuando sea necesaria, previniendo su pérdida o inaccesibilidad.(Domingo Pérez, 2024)	Alta
SySR14, SySR15	Interfaz de usuario adecuada para la visualización de datos.	La interfaz de usuario representa el medio de interacción entre una persona y un sistema, dispositivo o aplicación. Está conformada por componentes visuales y de control que facilitan una experiencia de uso eficiente y agradable para el usuario al interactuar con la tecnología.(GoDaddy, 2021)	Alta

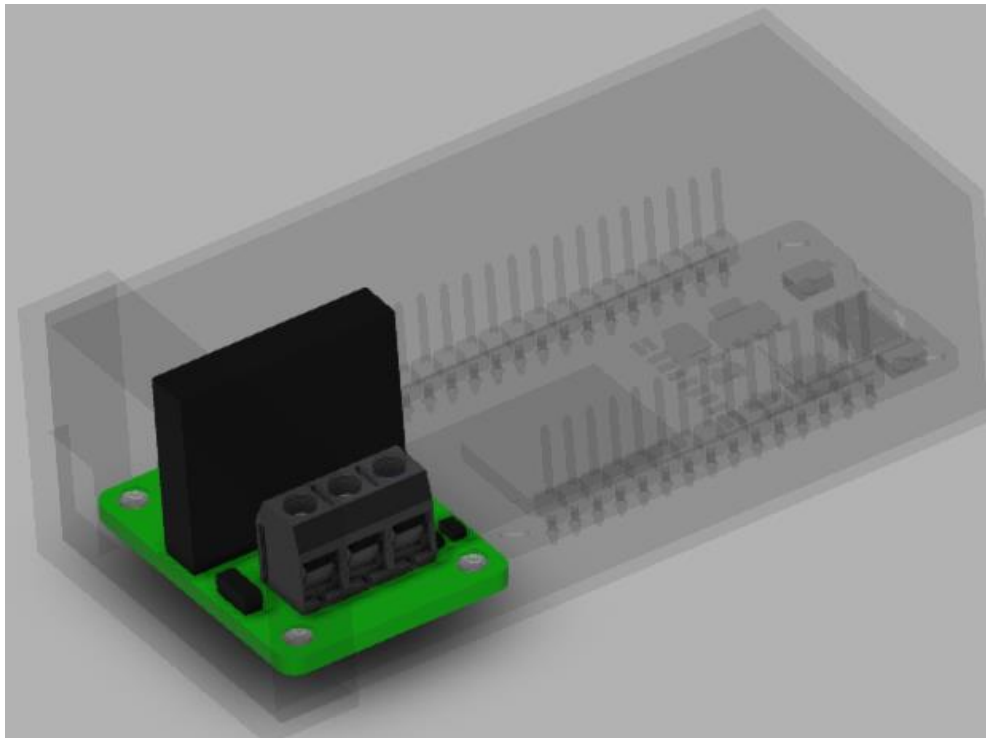
Requerimientos de Arquitectura			
Abreviatura	Requerimiento	Descripción	Prioridad
SRS14	El sitio web tiene una opción en la que solo el usuario con la contraseña puede ingresar para configuraciones de datos.	Las violaciones de privacidad en Internet representan riesgos significativos y tangibles. Es fundamental implementar medidas de protección, como el uso de contraseñas robustas, autenticación de dos factores y cifrado de datos, para minimizar estos riesgos y salvaguardar tu información en línea.(Kaspersky, 2025)	Alta

ANEXO 4. Diseño en 3D de sistema de monitoreo.









ANEXO 5. Simulación en Radio Mobile

Para llevar a cabo el proceso de simulación, se utilizó la herramienta de simulación Radio Mobile, dentro de esta herramienta nos permite ingresar coordenadas geográficas del sitio que deseamos observar y los diferentes parámetros técnicos de los dispositivos de comunicación inalámbrica.

Dentro de la siguiente Figura se observa las coordenadas de longitud y latitud del nodo sensor.

Coordinates

Latitude ° ' "

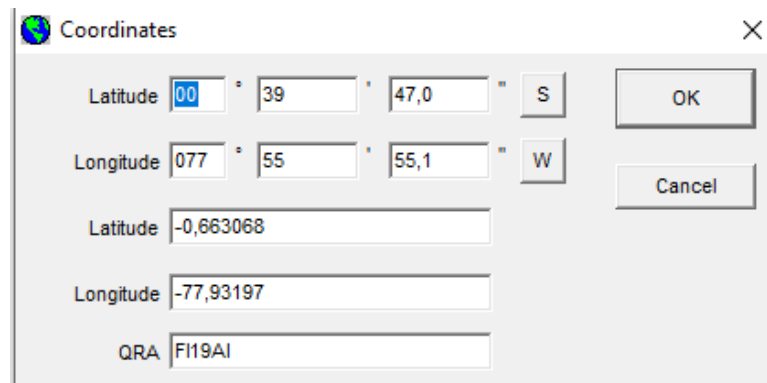
Longitude ° ' "

Latitude

Longitude

QRA

A continuación, en la siguiente Figura se observa las coordenadas del nodo coordinador.



Coordinates

Latitude 00° 39' 47,0" S

Longitude 077° 55' 55,1" W

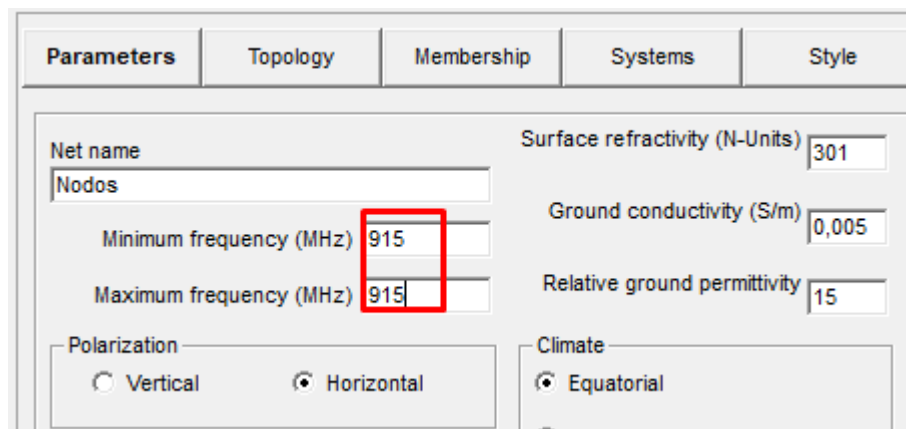
Latitude -0,663068

Longitude -77,93197

QRA F119AI

OK Cancel

Al finalizar ingresar la ubicación de los nodos, se procede a ingresar la frecuencia con la que van a operar los nodos, en este caso se coloca como frecuencia mínima y máxima 915 MHz ya que es con la que se va a operar en este proyecto.



Parameters Topology Membership Systems Style

Net name Nodos

Minimum frequency (MHz) 915

Maximum frequency (MHz) 915

Polarization Vertical Horizontal

Surface refractivity (N-Units) 301

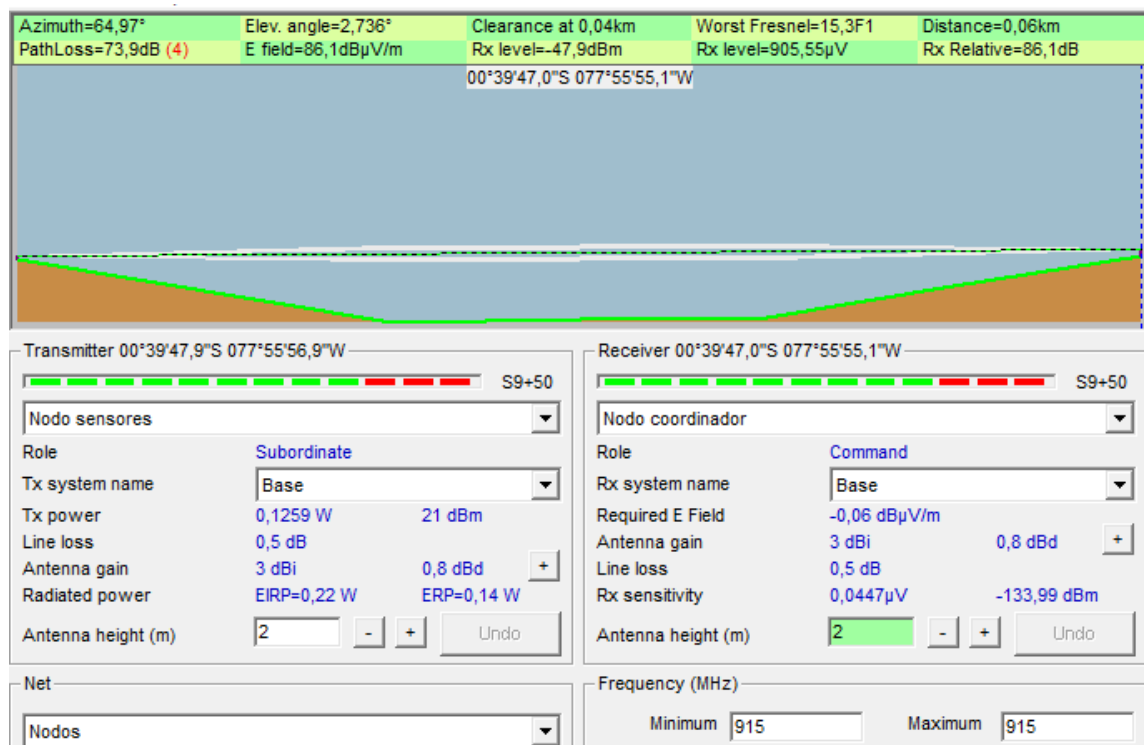
Ground conductivity (S/m) 0,005

Relative ground permittivity 15

Climate Equatorial

Por otro lado, se ingresa los parámetros que vienen dentro del datasheet del WiFi LoRa 32 para la comunicación inalámbrica. La altura de las antenas estan establecidas con la idea de evitar obstáculos, aumentar el alcance y mejorar la calidad de la señal entre ambos nodos.

Parameters	Topology	Membership	Systems	Style
00 Select from Base ... Mobile ...				
System name Base				
Transmit power (Watt)	0,12589254	(dBm)	21	
Receiver threshold (μV)	0,0447	(dBm)	-134	
Line loss (dB)	0,5	(Cable+cavities+connectors)		
Antenna type	omni.ant	View		
Antenna gain (dBi)	3	(dBd)	0,85	
Antenna height (m)	3	(Above ground)		
Additional cable loss (dB/m)	0	(If antenna height differs)		
Add to Radiosys.dat		Remove from Radiosys.dat		



La simulación de red realizada con el software Radio Mobile se basa en dos puntos específicos, definidos por sus coordenadas geográficas, donde se instalarán los dos nodos. La topología de la red correspondiente se presenta en la figura siguiente.



Se presentan los resultados de la simulación de la red inalámbrica entre el nodo de sensores y el nodo coordinador. Estos resultados, generados por el simulador Radio Mobile, incluyen detalles como la distancia entre ambos nodos, el ángulo de azimut y otros parámetros relevantes.

Radio Link
×

Edit View Swap

Distance between Nodo sensores and Nodo coordinador is 0,1 km (0,0 miles)
 True North Azimuth = 64,97°, Magnetic North Azimuth = 69,23°, Elevation angle = 2,7357°
 Terrain elevation variation is 5,3 m
 Propagation mode is line-of-sight, minimum clearance 15,7F1 at 0,0km
 Average frequency is 915,000 MHz
 Free Space = 67,6 dB, Obstruction = -0,8 dB, Urban = 0,0 dB, Forest = 0,0 dB, Statistics = 6,6 dB
 Total propagation loss is 73,4 dB
 System gain from Nodo sensores to Nodo coordinador is 160,0 dB
 System gain from Nodo coordinador to Nodo sensores is 160,0 dB
 Worst reception is 86,6 dB over the required signal to meet 70,000% of situations
 Warning 4

Transmitter 00°39'47,9"S 077°55'56,9"W

S0

Nodo sensores

Role: Subordinate

Tx system name: Base

Tx power: 0,1259 W 21 dBm

Line loss: 0,5 dB

Antenna gain: 3 dBi 0,8 dBd

Radiated power: EIRP=0,22 W ERP=0,14 W

Antenna height (m): 3

Receiver 00°39'47,0"S 077°55'55,1"W

S0

Nodo coordinador

Role: Command

Rx system name: Base

Required E Field: -0,06 dBμV/m

Antenna gain: 3 dBi 0,8 dBd

Line loss: 0,5 dB

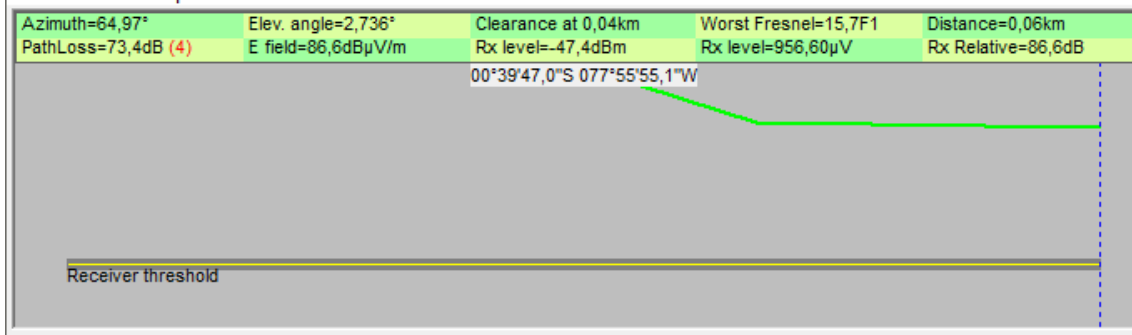
Rx sensitivity: 0,0447μV -133,99 dBm

Antenna height (m): 3

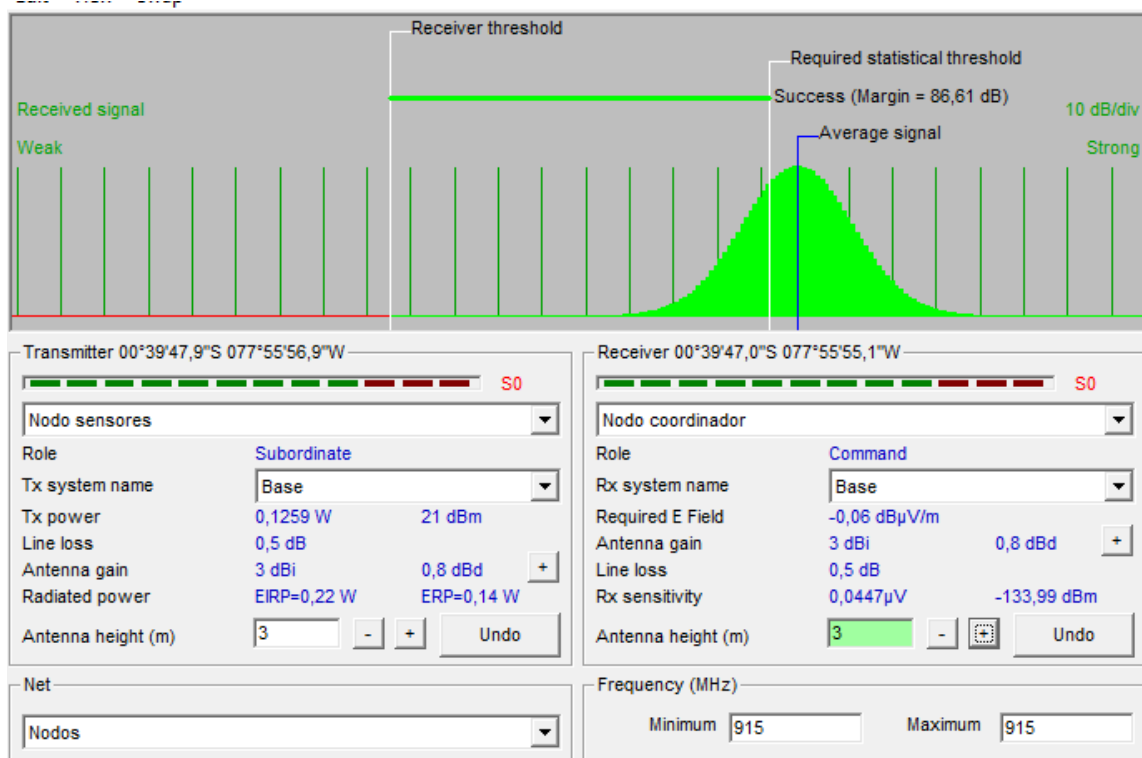
Net: Nodos

Frequency (MHz): Minimum 915 Maximum 915

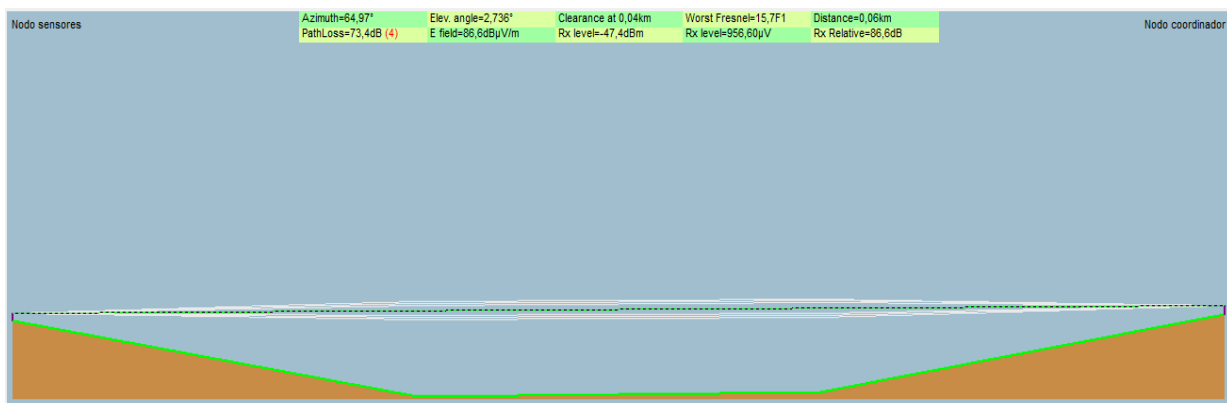
Asimismo, se presenta el umbral de recepción entre el nodo sensor y el nodo coordinador, el cual registra un nivel de recepción de -46.2 [dBm], una recepción relativa de 87.8 [dB] y pérdidas de 82.2 [dB].



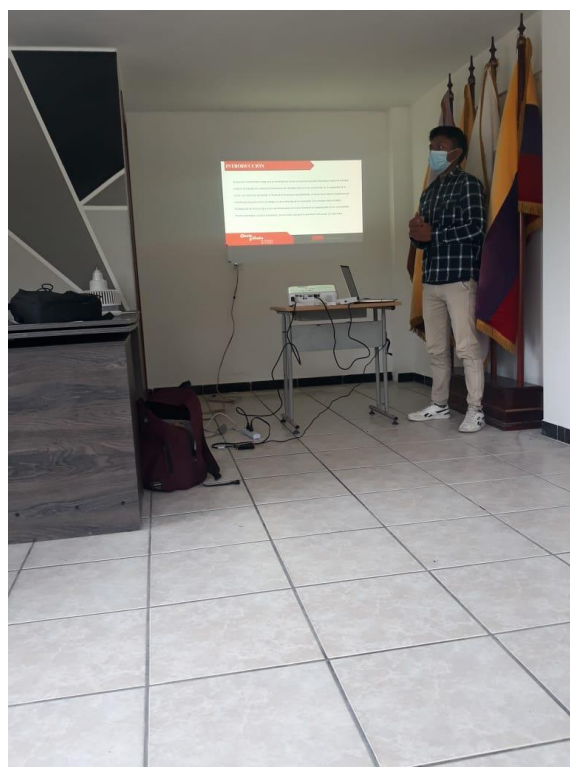
Adicionalmente, se determina el umbral estadístico necesario, junto con los márgenes de éxito para la transmisión y el valor promedio de la señal.



Por último, se analiza desde otra perspectiva la distancia entre los dos nodos y la línea de visión que conecta a cada uno de ellos.



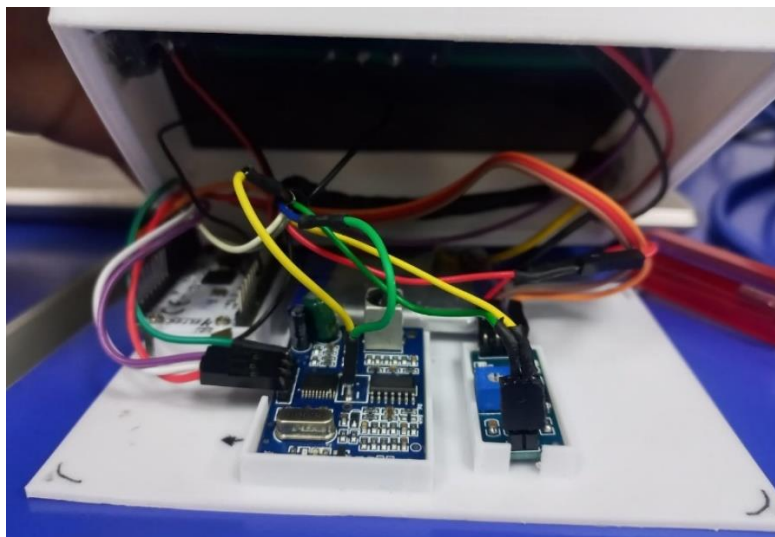
ANEXO 6. Socialización de sistema de monitoreo.



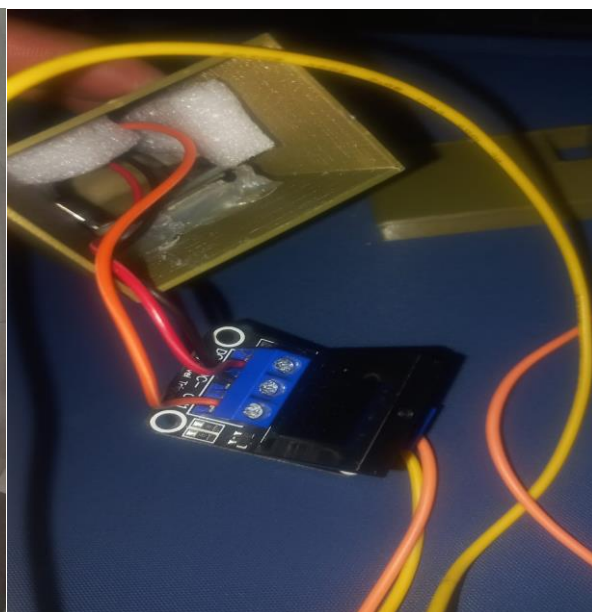




ANEXO 7. Instalación de nodos.









ANEXO 8. Conversión de modelo entrenado.

Dentro del siguiente código se procede a realizar la conversión del modelo entrenado en keras hacia el formato TensorFlow Lite. Para realizar el proceso, se utiliza la herramienta TFLiteConverter para convertir el modelo a formato tflite, este formato resulta ser liviano e ideal para dispositivos con recursos limitados.

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
tflite_model = converter.convert()

# Save the model to disk
open("exercice_model.tflite", "wb").write(tflite_model)
```

En el siguiente apartado, se procede a realizar la evaluación e inferencia del modelo convertido a tflite. Este proceso permite alimentar nuevos datos de entrada para generar predicciones. El resultado corresponde a una métrica de precisión general, que permitió cuantificar la capacidad del modelo TFLite para clasificar de manera adecuada nuevas entradas, de esa manera se garantiza su efectividad.

```
import tensorflow as tf
```



```

import numpy as np

# Asegúrate de que X_test esté en el tipo correcto (float32)
X_test = X_test.astype('float32')

# Cargar el modelo TFLite
interpreter = tf.lite.Interpreter(model_path="exercice_model.tflite")

# Asignar memoria para el modelo
interpreter.allocate_tensors()

# Obtener detalles de los tensores de entrada y salida
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

# Seleccionar un ejemplo de prueba para realizar la predicción (puedes
cambiar el índice)
test_index = 15 # Cambia esto para probar con otros datos de prueba

# Asegúrate de que la entrada esté correctamente estructurada (1, 3)
input_data = X_test[test_index].reshape((1, 3)) # (1, 3) si tu modelo
espera 3 características

# Configurar la entrada del modelo
interpreter.set_tensor(input_details[0]['index'], input_data)

# Ejecutar la inferencia
interpreter.invoke()

# Obtener los resultados de la salida
output_data = interpreter.get_tensor(output_details[0]['index'])

# Imprimir los resultados
print(f"Predicción del modelo: {output_data}")

# Obtener la clase predicha (la clase con la mayor probabilidad)
predicted_class = np.argmax(output_data)
print(f"Clase predicha: {predicted_class}")

# Imprimir la etiqueta real
print(f"Etiqueta real: {y_test[test_index]}")

# Para evaluar el modelo en todo el conjunto de prueba
correct_predictions = 0
total_predictions = len(X_test)

for i in range(total_predictions):
    input_data = X_test[i].reshape((1, 3))
    interpreter.set_tensor(input_details[0]['index'], input_data)
    interpreter.invoke()

    output_data = interpreter.get_tensor(output_details[0]['index'])
    predicted_class = np.argmax(output_data)

    if predicted_class == y_test[i]:
        correct_predictions += 1

```

```
# Calcular la precisión del modelo en los datos de prueba
accuracy = correct_predictions / total_predictions
print(f"Test accuracy: {accuracy}")
```

En la siguiente parte del código se utiliza la función OS para calcular el tamaño final del modelo en formato TensorFlow Lite. La implementación de la función os, es importante cuando se trabaja con dispositivos de bajo consumo ya que permite conocer que el modelo cumple con los requisitos de los dispositivos. El tamaño total del modelo básico es de 5116 bytes.

```
import os
basic_model_size = os.path.getsize("exercice_model.tflite")
print("Basic model is %d bytes" % basic_model_size)
```

En el siguiente apartado se procede a convertir el archivo binario del modelo tflite hacia un archivo en lenguaje C. Para llevar a cabo este proceso, se instala la herramienta “xxd” la cual se encarga de convertir archivos binarios a hexadecimales o arreglos tipo C.

```
# Save the file as a C source file
!xxd -i exercice_model.tflite > exercice_model.cc
# Print the source file
!cat exercice_model.cc
```

```
↩ unsigned char exercice_model_tflite[] = {
    0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
    0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
    0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
    0x98, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x00, 0x34, 0x0c, 0x00, 0x00,
    0x94, 0x0c, 0x00, 0x00, 0x98, 0x13, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
    0x01, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0a, 0x00,
    0x10, 0x00, 0x0c, 0x00, 0x08, 0x00, 0x04, 0x00, 0x0a, 0x00, 0x00, 0x00,
    0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x00,
    0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
    0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
    0x04, 0x00, 0x00, 0x00, 0x90, 0xff, 0xff, 0xff, 0x0d, 0x00, 0x00, 0x00,
    0x04, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x6f, 0x75, 0x74, 0x70,
    0x75, 0x74, 0x5f, 0x30, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
    0x04, 0x00, 0x00, 0x00, 0x5e, 0xf3, 0xff, 0xff, 0x04, 0x00, 0x00, 0x00,
    0x0c, 0x00, 0x00, 0x00, 0x6b, 0x65, 0x72, 0x61, 0x73, 0x5f, 0x74, 0x65,
    0x6e, 0x73, 0x6f, 0x72, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00,
    0x34, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0xdc, 0xff, 0xff, 0xff,
```

En el siguiente apartado se observa el archivo de cabecera Data.h en donde contiene el modelo de TensorFlow Lite convertido a un arreglo en bytes.

```
receptor.ino  data.cpp  data.h  firestore.cpp  firestore.h

// This is a standard TensorFlow Lite model file that has been converted into a
// C data array, so it can be easily compiled into a binary for devices that
// don't have a file system. It was created using the command:
// xxd -i sine_model.tflite > sine_model_data.cc

#ifndef DATA_H_
#define DATA_H_

extern const unsigned char mi_data[];
extern const int mi_data_len;

#endif
```

Finalmente, el modelo entrenado y convertido a un arreglo de bytes expresado en hexadecimal en lenguaje C, permitió la integración dentro del entorno de Arduino IDE. Tomando en cuenta que, inicialmente se obtuvo un archivo .cc, este archivo no va a compilar dentro del Arduino IDE en Windows. Al ser equivalentes los archivos .cc y .cpp, dentro del entorno de Arduino IDE no compilará .cc ya que no valida como una extensión. Por esta razón, se renombró el archivo a .cpp para que el entorno de Arduino IDE compile el archivo como código C++. Al realizar el cambio no altera el contenido del archivo.

```
receptor.ino  data.cpp  data.h  firestore.cpp  firestore.h

// Automatically created from a TensorFlow Lite flatbuffer using the command:
// xxd -i sine_model.tflite > sine_model_data.cc
// See the README for a full description of the creation process.

#include "data.h"

// We need to keep the data array aligned on some architectures.

const unsigned char mi_data[] = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
  0x9c, 0x00, 0x00, 0x00, 0xf4, 0x00, 0x00, 0x00, 0x88, 0x0c, 0x00, 0x00,
  0x98, 0x0c, 0x00, 0x00, 0xc4, 0x13, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
```



Manual de usuario

**"SISTEMA DE MONITOREO CON LÓGICA
DIFUSA PARA PREVER EL
DESBORDAMIENTO DEL RÍO COSANGA EN
LA PARROQUIA DE COSANGA"**

Índice

1. Manual de Usuario	234
1.1. Inicio de interfaz	234
1.1.1. Menú.....	236
1.2. Aplicación Móvil	240

Manual de Usuario

Dentro de este apartado se dará a conocer detalladamente el proceso que se debe llevar a cabo dentro del sistema para visualizar los datos recolectados, entenderlos y en caso de ser necesario, la realización de configuraciones o cualquier mantenimiento del sistema.

Inicio de interfaz

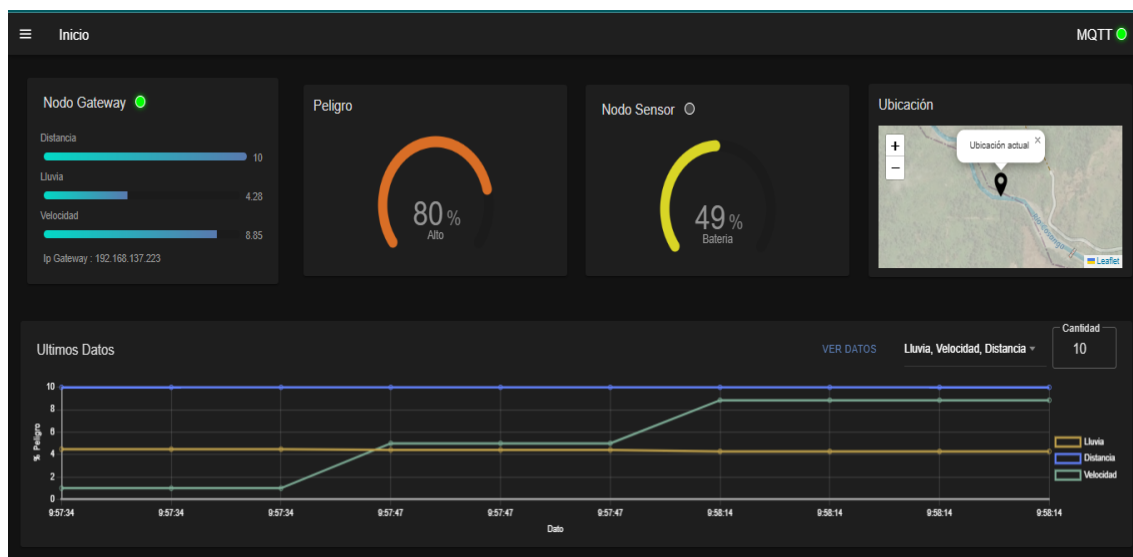
El usuario debe acceder al siguiente link <https://rio-sense.web.app/home> para lograr visualizar el sitio web del sistema.

Cuando se ingresa al sitio web se puede observar directamente los datos procesados, el nivel de riesgo que se obtiene según los datos recolectados, también se logra visualizar el porcentaje de batería del nodo encargado de recolectar datos, el mapa de ubicación del nodo y la gráfica de los datos recolectados. Esta última indica la toma de los 10 primeros datos y va cambiando cada cumple su ciclo, de igual manera se puede modificar para que se visualice los datos en un tramo menor.

Dentro del nivel de peligro se logra visualizar el porcentaje que maneja, para este caso el sistema maneja 4 colores de identificación, verde, amarillo, naranja y rojo, indicando un nivel bajo, medio y los dos últimos colores indican un nivel alto de riesgo.

Figura 1.

Interfaz inicial de sistema de monitoreo.

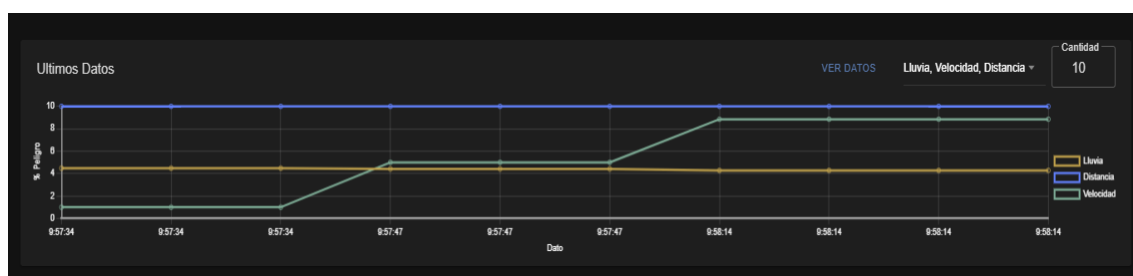


Fuente: Autoría.

El sistema indica de manera gráfica como va cambiando la lectura de los datos, hay que tomar en cuenta para que el usuario que este dentro del sitio web entienda lo que se trata estas señales. En la parte derecha en el eje Y se encuentra el nivel de peligro en donde su valor más alto es 10 indicando que el nivel de peligro es alto y su valor más bajo es 1 indicando que el nivel es bajo. Por otro lado, en el eje X se encuentra el tiempo en el que se está tomando la lectura de los sensores.

Figura 2.

Señales gráficas de datos recolectados.

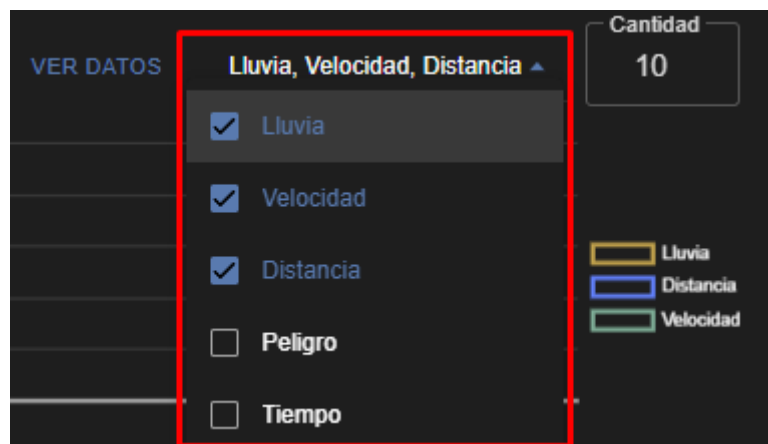


Fuente: Autoría.

Dentro de la gráfica de las lecturas de los sensores, se desliza una barra en donde se elige solamente observar las señales graficas de los sensores o también el nivel de peligro que indica el sistema o también el tiempo que esta tomando datos el sistema.

Figura 3.

Modo de visualización de las señales gráficas.



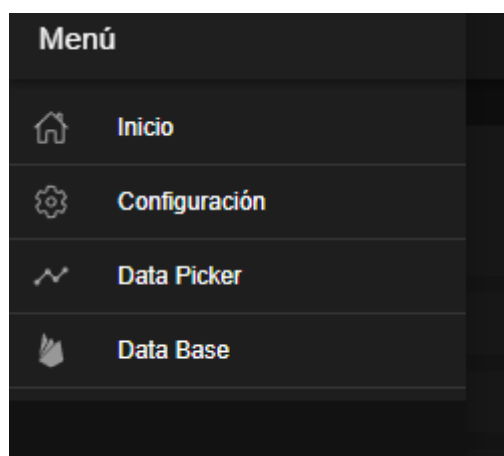
Fuente: Autoría.

Menú

El sistema contiene un pequeño menú en donde se da a conocer la funcionalidad de cada uno. Este menú es ideal para tener una interfaz ordenada que a su vez no pueda confundir al usuario. De esta manera, se optó por un pequeño menú y que aporte de gran ayuda a los usuarios.

Figura 4.

Menú de opciones de sistema de monitoreo.

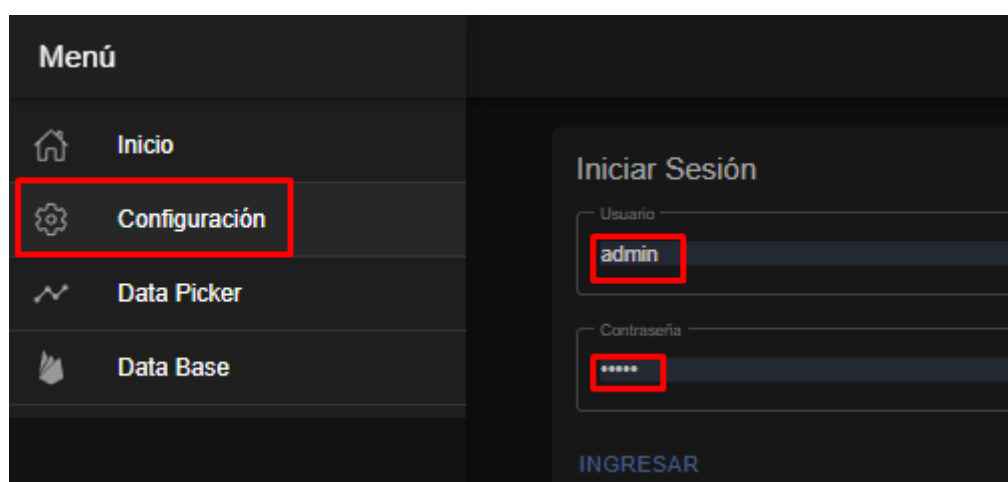


Fuente: Autoría.

Dentro del menú de opciones del sistema, se observa la opción de configuración, allí solamente el usuario con las credenciales puede acceder.

Figura 5.

Menú de opciones de sistema.



Fuente: Autoría.

Al ingresar a las configuraciones, se logra observar el broker con el que trabajar el sistema para recibir y enviar datos, de igual manera, este broker puede ser modificado en caso de ser necesario, recordando que solamente se puede modificar en este sitio y solamente por el usuario que tenga acceso.

A continuación, se explicará el significado de cada botón en este apartado.

Prueba MQTT: La sección se encarga para hacer pruebas para mantenimiento o cambio de broker.

MQTT DATA: Se encarga de recibir datos

LORA RAW: Se encarga de filtrar los datos reales tomados por los sensores sin pasar por el proceso de fusificación.

NOTIFICACIÓN: Este botón es agregado para realizar pruebas de funcionamiento, para alertar a la comunidad de manera manual, también sirve para la comunidad cuando realicen pruebas de evacuación en caso de sismos o algo parecido.

CAPTURAR: Este botón es utilizado para activar y desactivar el almacenamiento de los datos recolectados dentro de la base de datos.

Figura 6.

Visualización de configuraciones de sistema.

The screenshot shows a web application interface for system configuration. The title bar at the top left says 'Configuración'. The main content area is divided into four sections:

- Configuración MQTT:** This section contains four input fields: 'Broker Hostname' (with value 'broker.emqx.io'), 'Port' (with value '8084'), 'Path' (with value '/mqtt'), and 'Tema' (with value 'tony/#'). Below these fields are two buttons: 'CONECTAR' and 'GUARDAR CONFIGURACIÓN'. A red rectangular border highlights the four input fields.
- Prueba MQTT:** This section contains two input fields: 'Tema' (with value 'tony/ESP/') and 'Mensaje' (with value 'Cliente Funcional!'). Below these fields is a button labeled 'PUBLICAR'.
- Comandos MQTT:** This section contains four buttons: 'MQTT DATA', 'AUTO DATA LORA', 'LORA RAW', and 'NOTIFICACIÓN'.
- Capturar Datos:** This section contains a single button labeled 'CAPTURAR'.

Fuente: Autoría

El ultimo icono del menú se encarga de indicarnos los datos que fueron recolectados.

Figura 7.

Datos almacenados del sistema.

Configuración	Ultrasonico	Lluvia	Infrarrojo
Data Picker	2.45	1.11	1.21
Data Base	5.52	9.21	6.74
	3.22	1.52	1.55
	4.42	7.12	5.11
	2.45	1.07	1.63
	2.98	1.44	2.11
	3.77	1.79	2.46
	5.57	1.28	1.54
	2.19	1.71	1.66
	5.57	7.53	5.44
	5.37	1.22	1.72
	3.49	1.74	1.69
	5.53	7.85	6.08

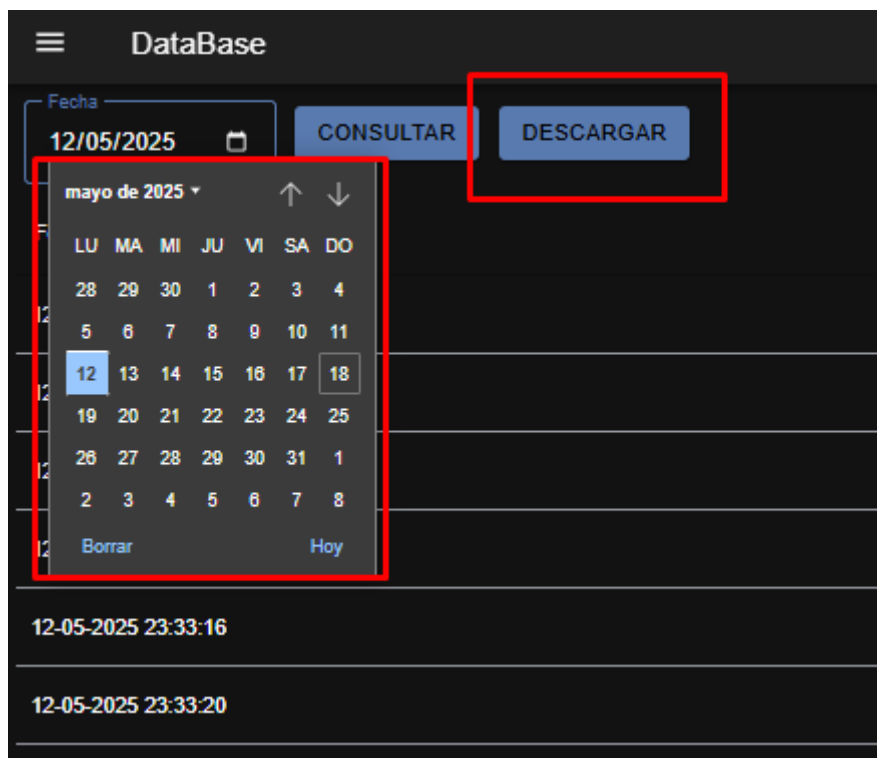
Fuente: Autoría.

Se añadió un calendario en donde aparecerá los datos de los días en donde se a recolectado los datos y son procesados.

Adicionalmente, en este apartado, se puede descargar los datos recolectados de los días que se elijan.

Figura 8.

Calendario de visualización de datos.



Fuente: Autoría.

Aplicación Móvil

Dentro de la aplicación móvil se logra visualizar los datos recolectados y las notificaciones emitidas por el sistema cuando los niveles del río son críticos.

Instalación de aplicación móvil

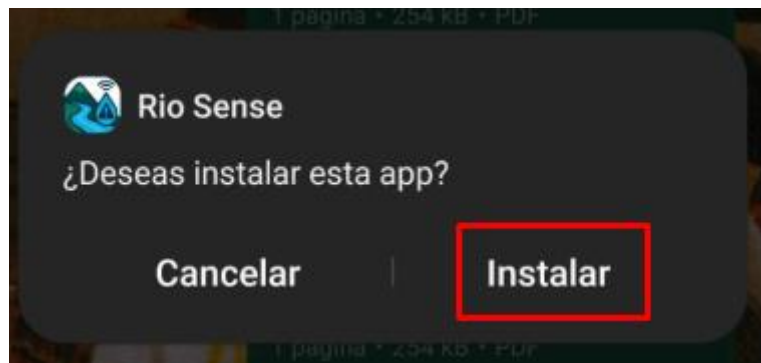
Para iniciar el proceso de instalación de la aplicación web se ingresa al siguiente enlace en donde contiene el archivo apk de la pp.

- [RioSense.apk](#)

Al descargarse el archivo apk se abre una ventana de la aplicación y se inicia la instalación de la app.

Figura 9.

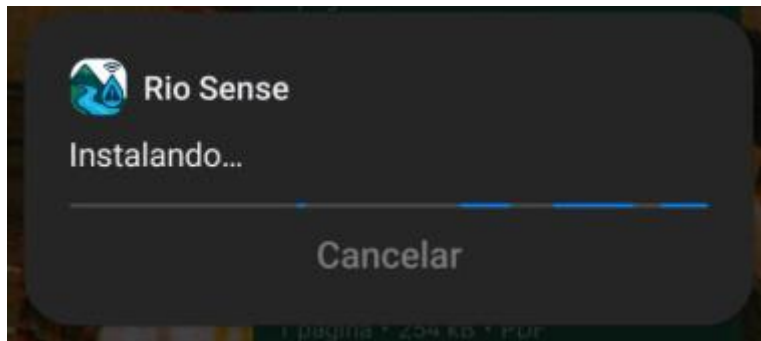
Ventana de inicio de instalación.



Fuente: Autoría.

Figura 10.

Proceso de instalación.

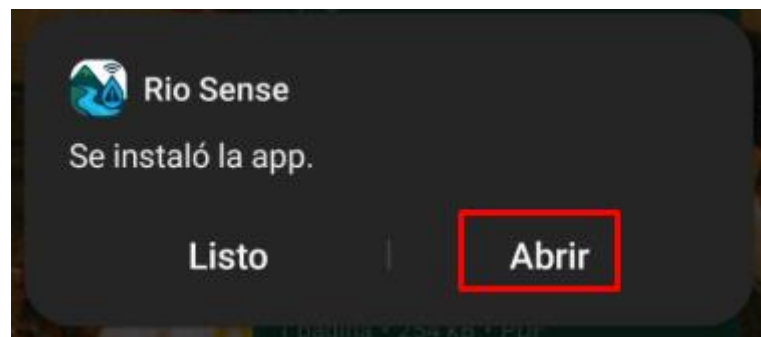


Fuente: Autoría.

Al finalizar la instalación, se procede a abrir la aplicación para generar el token dentro de firebase.

Figura 11.

Ventana de ingreso a la app.

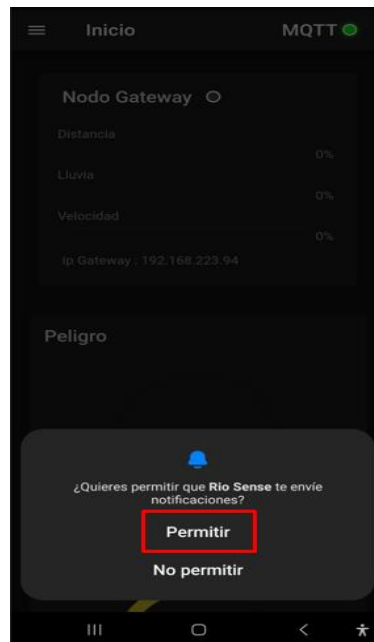


Fuente: Autoría.

Dentro de la aplicación es necesario dar acceso a las notificaciones para recibir las alertas del sistema de monitoreo.

Figura 12.

Acceso a notificaciones del sistema.

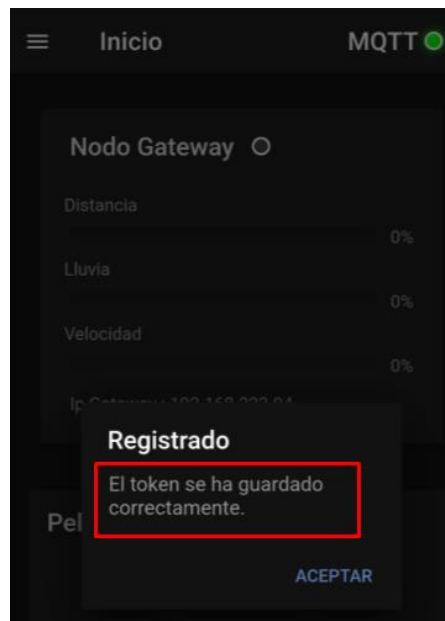


Fuente: Autoría.

Al finalizar el proceso de instalación de la aplicación se crea el token en donde quedará registrado dentro de firebase para que se visualicen los datos y recibir los mensajes de alertas.

Figura 13.

Token de activación de aplicación móvil.

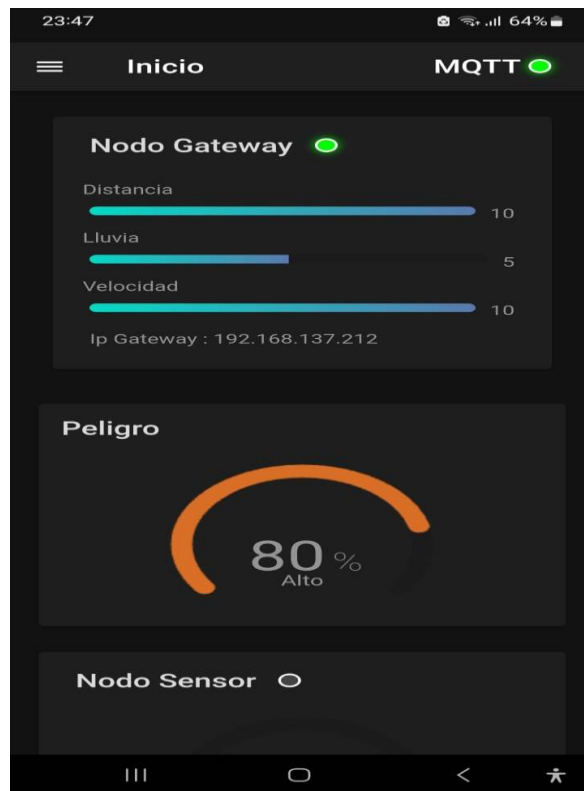


Fuente: Autoría.

Se mantiene un diseño igual al del sitio web, lo cual facilita la familiarización del diseño. La finalidad del diseño del sitio web y aplicación responde a la necesidad de tener acceso inmediato a los datos recolectados ya que se trata de un sistema de monitoreo que puede acceder cualquier persona dentro de la parroquia.

Figura 14.

Interfaz de aplicación móvil.

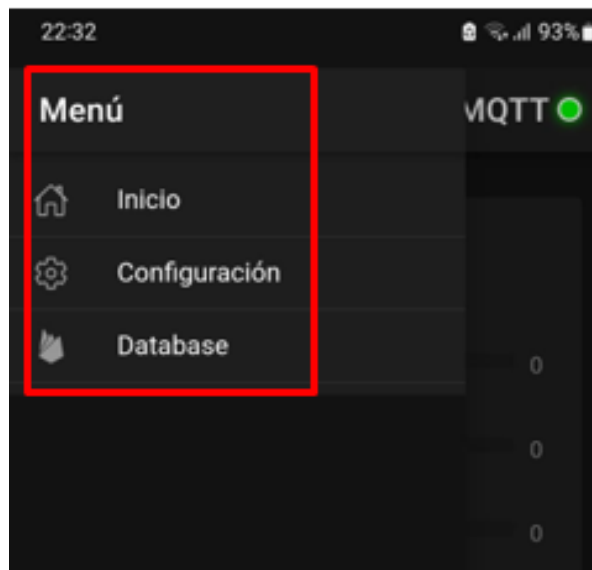


Fuente: Autoría.

De igual manera, en la parte izquierda de la aplicación se encuentra el ícono de menú y se desliza las 3 opciones del sistema, en donde se observa el inicio, configuración y databse.

Figura 15.

Menú de opciones de app móvil.

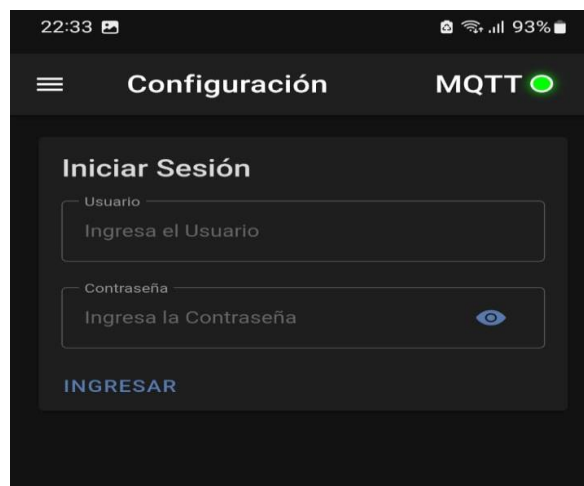


Fuente: Autoría.

Dentro del ícono de configuración solamente el usuario con las credenciales puede acceder para realizar alguna configuración necesaria del sistema o emitir una alerta de manera manual.

Figura 16.

Acceso de usuario en app móvil.



Fuente: Autoría.

El último ícono indica los datos que son recolectados y debidamente procesados, estos datos pueden ser consultados según la fecha de interés, de igual manera se puede

descargar los datos para formar un histograma para futuros proyectos relacionados a la información del río.

Figura 17.

Visualización de datos en app móvil.



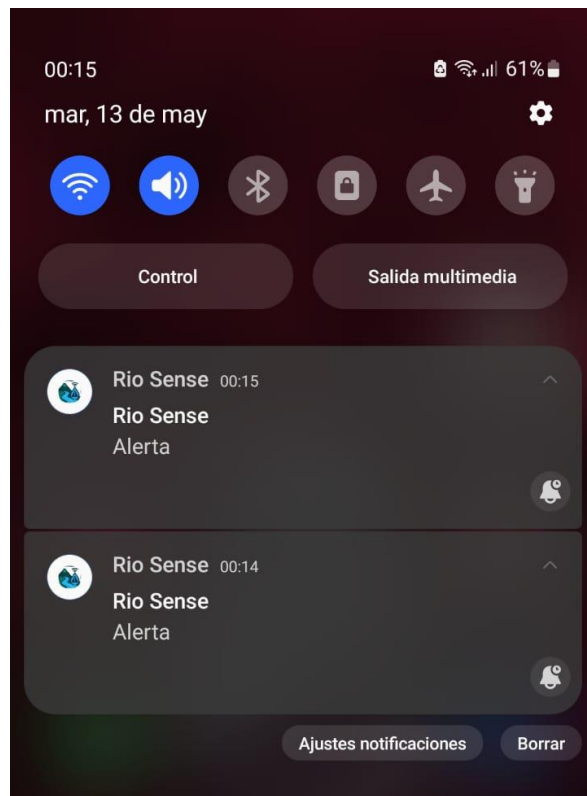
Fecha	Ultrasonico	Lluvia	Infrarrojo
02-04-2025 03:46:28	10	5	3
03-04-2025 03:46:27	10	4	2
04-04-2025 03:46:46	2.3	4	5
12-05-2025 23:32:59	10	5	1
12-05-2025 23:33:04	10	5	10
12-05-2025 23:33:08	10	5	10
12-05-2025			

Fuente: Autoría.

Cuando surge una alerta, el sistema envía una notificación de alerta a los dispositivos móviles conectados al sistema. Esta alerta solamente se envía cuando existe alto riesgo de desbordamiento.

Figura 18.

Verificación de notificación de alerta de sistema.



Fuente: Autoría.