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

INFORME FINAL DEL PROYECTO DE INTEGRACIÓN CURRICULAR, PROPUESTAS TENOLÓGICAS

TEMA:

"SISTEMA DE ALERTA CONTRA HELADAS PARA LA PROTECCIÓN DE CULTIVOS VULNERABLES UTILIZANDO ARQUITECTURA IOT Y TÉCNICAS DE IA EN LA PARROQUIA RURAL OLMEDO – PESILLO PERTENECIENTE AL CANTÓN CAYAMBE"

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

AUTORA:

Guatemal Neppas Katherine Mishell

DIRECTOR:

Msc. Luis Edilberto Suárez Zambrano

Ibarra, 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 DEL CONTACTO			
CÉDULA DE IDENTIDAD	1004351258		
APELLIDOS Y NOMBRES	Guatemal Neppas Katherine Mishell		
DIRECCIÓN	Ibarra, Luis Eduardo Dávila y Ricardo Sánchez		
E-MAIL	kmguatemaln@utn.edu.ec kmguatemaln@gmail.com		
TELÉFONO FIJO		**TELÉFONO MÓVIL**	0994106572

DATOS DE LA OBRA	
TÍTULO	Sistema de alerta contra heladas para la protección de cultivos vulnerables utilizando arquitectura IoT y Técnicas de IA en la parroquia rural Olmedo – Pesillo perteneciente al cantón Cayambe.
AUTOR	Guatemal Neppas Katherine Mishell
FECHA	15/02/2025
PROGRAMA	■ GRADO ☐ POSGRADO
TÍTULO	Ingeniera en Telecomunicaciones
DIRECTOR	Msc. Suárez Zambrano Luis Edilberto

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Katherine Mishell Guatemal Neppas, con cédula de identidad Nro. 1004351258, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 15 días del mes de febrero de 2025

EL AUTOR:

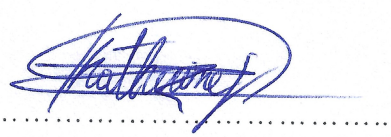
Guatemal Neppas Katherine Mishell

CONSTANCIAS

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

Ibarra, a los 15 días del mes de febrero de 2025

EL AUTOR



.....

Guatemal Neppas Katherine Mishell

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 14 de febrero. de 2025

Magíster Luis Edilberto Suárez Zambrano
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

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

MSc. Luis Edilberto Suárez Zambrano.

C.C.: 1002304291

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular "Sistema de alerta contra heladas para la protección de cultivos vulnerables utilizando arquitectura IoT y técnicas de IA en la Parroquia Rural Olmedo – Pesillo perteneciente al Cantón Cayambe" elaborado por Katherine Mishell Guatemal Neppas, previo a la obtención del título del Ingeniero en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f): .....
Ing. Luis Edilberto Suárez Zambrano, Msc.
C.C.: 1002304291

(f): .....
Ing. Carlos Alberto Vásquez Ayala, Msc.
C.C.: 1002424982

DEDICATORIA

A mis padres, Eloisa Neppas y Jorge Guatemal, por su amor incondicional, sus enseñanzas y sacrificios que me han permitido alcanzar esta meta tan anhelada. Su confianza y apoyo han sido mi mayor motivación para seguir adelante.

A esa persona incondicional A.S., por estar siempre a mi lado, brindándome su cariño, paciencia y apoyo. Su compañía ha sido un refugio en los momentos difíciles y una fuente de inspiración en este camino.

A todos aquellos que confiaron en mis capacidades y me brindaron su apoyo, les dedico este trabajo con todo mi corazón como muestra de gratitud y reconocimiento.

.

Katherine Mishell Guatemal Neppas

AGRADECIMIENTO

Al concluir esta importante etapa de mi vida, mi primer agradecimiento es para Dios, quien ha sido mi fuente inagotable de fortaleza y guía en cada paso del camino. Su presencia me ha dado la fuerza y la sabiduría necesarias para superar cada obstáculo, permitiéndome avanzar con fe y determinación.

A mis padres que fueron la base esencial de mi vida, apoyándome y brindándome su amor. Gracias por creer en mí, por sus sacrificios y por ser la fuerza que me motiva a continuar. Siempre les estaré profundamente agradecida.

A mi familia: Rosita, Piedad, Aleluya, Lidia, Eloisa, Cesar y primos por acompañarme en cada etapa del camino, por sus palabras de motivación en las dificultades y por confiar en mí. Gracias por su paciencia y comprensión.

A la segunda familia que la vida me regaló: Carmita, Carlitos C., Patricia, Diego C., Gustavo, Alexandra, Carlita, Javier, Lizbeth, Anahí, Nayeli, Alexis, Dieguito, Jeremy y Carlitos X., quienes, con sus palabras de aliento, fueron una fuente de motivación en este camino. Gracias a ustedes, nunca me rendí y encontré la fuerza para afrontar los obstáculos.

A mis amigos y compañeros, especialmente a Luis, Pablo, Raúl, Anderson, Edin, Fernanda y su hijo Aarón, quienes han sido mi apoyo, compartiendo conmigo alegrías, desafíos y largas jornadas de estudio. Su compañía y ánimo han hecho que este proceso sea más llevadero.

Expreso mi gratitud a la Universidad Técnica del Norte por brindarme la oportunidad de formarme como profesional. En especial, agradezco al Msc. Luis Suárez, Msc. Carlos Vásquez y Msc. Jaime Michilena quienes me han inculcado valores para ser un profesional con ética, además su paciencia, orientación y conocimientos compartidos fueron la clave para culminar de manera exitosa este trabajo.

Este trabajo es el resultado del esfuerzo, la perseverancia y el apoyo de cada uno de ustedes. A todos, gracias de corazón.

Katherine Mishell Guatemal Neppas

RESUMEN

El presente trabajo de titulación describe el diseño e implementación de un sistema de alerta contra heladas basado en una arquitectura IoT con integración de Inteligencia Artificial, permitiendo el monitoreo de variables climáticas como temperatura, humedad, velocidad del viento e índice UV para la detección temprana de este fenómeno y la generación de alertas preventivas. El estudio se basa en la necesidad de proteger los cultivos vulnerables a las heladas, optimizando la decisión oportuna de los agricultores para reducir pérdidas económicas.

El desarrollo del sistema se llevó a cabo utilizando la metodología Action Research, cumpliendo con sus etapas principales; en la fase de análisis, se realizó una investigación detallada sobre los conceptos clave del fenómeno y las tecnologías a implementar; en la fase de planificación, se validaron los requerimientos del sistema, incluyendo necesidades de los stakeholders, especificaciones de hardware y software, y la arquitectura general de la solución. Durante la fase de implementación, se desarrolló e instaló el sistema en el terreno con la presencia del cultivo, asegurando su funcionalidad en condiciones de campo.

Finalmente, se realizó las pruebas de validación para verificar el cumplimiento del objetivo planteado, obteniendo como resultado un sistema confiable, preciso y adaptable, capaz de generar alertas oportunas basadas en datos en tiempo real.

Los resultados demuestran que la solución desarrollada es efectiva para mitigar el impacto de las heladas, permitiendo la prevención de daños en los cultivos y promoviendo el uso de tecnología avanzada en el sector agrícola.

Palabras clave: Alerta contra heladas, árbol de decisiones, monitoreo climático, fenómeno climático, LPWAN, inteligencia artificial.

ABSTRACT

This thesis describes the design and implementation of a frost alert system based on an IoT architecture integrated with Artificial Intelligence, allowing the monitoring of climatic variables such as temperature, humidity, wind speed, and UV index for the early detection of this phenomenon and the generation of preventive alerts. The study is based on the need to protect crops vulnerable to frost, optimizing timely decision-making by farmers to reduce economic losses.

The system was developed using the Action Research methodology, following its main stages. In the analysis phase, a detailed investigation was conducted on the key concepts related to the phenomenon and the technologies to be implemented. In the planning phase, the system requirements were validated, including stakeholders' needs, hardware and software specifications, and the overall system architecture. During the implementation phase, the system was developed and installed in the field alongside the crops, ensuring its functionality under real farming conditions.

Finally, validation tests were conducted to verify the achievement of the proposed objective, resulting in a reliable, precise, and adaptable system, capable of generating timely alerts based on real-time data.

The results demonstrate that the developed solution is effective in mitigating the impact of frost, enabling the prevention of crop damage and promoting the use of advanced technology in the agricultural sector.

Keywords: Frost alert, decision tree, climate monitoring, climatic phenomenon, LPWAN, artificial intelligence.

LISTA DE SIGLAS

ABP. Activation By Personalization o Activación Por Personalización

OTAA. Over The Air Activation o Activación Por Aire

ADC. Convertidor Analógico a Digital

MQTT. Transporte de Telemetría en Cola de Mensajes o Message Queuing Telemetry Transport

RSSI. Indicador de Intensidad de Señal Recibida

ToA. Tiempo en el Aire

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

CAPÍTULO I. ANTECEDENTES

1.1. Problema de investigación

Las heladas representan una de las mayores amenazas para la seguridad alimentaria y la estabilidad económica de los agricultores en diversas regiones del Ecuador, según el Instituto Nacional de Meteorología e Hidrología (INAMHI, 2010), el sector agrícola en Ecuador es uno de los más susceptibles a las consecuencias del cambio climático. Estos eventos climáticos pueden desencadenar sequías prolongadas, inundaciones repentinas y variaciones bruscas en las temperaturas. Además, los vientos, cuya dirección y fuerza varían según su origen geográfico, ejercen un impacto diferencial en las diferentes regiones naturales de Ecuador.

En particular, la región de la Sierra, caracterizada por su topografía montañosa y altitudes elevadas, enfrenta desafíos derivados de los fenómenos climáticos como las heladas, que pueden intensificar o mitigar los efectos del clima en la agricultura y producción de tubérculos, verduras, legumbres, hortalizas, granos, frutas, áreas de pastizales entre otros cultivos locales. (MIES, 2022)

La Parroquia Rural Olmedo – Pesillo perteneciente al Cantón Cayambe en la Región Sierra no es una excepción; por el contrario, debido a su ubicación geográfica, enfrenta mayores riesgos de sufrir la presencia de heladas. El impacto de las heladas se traduce en pérdidas significativas en la producción de las especies locales. No obstante, este fenómeno climático no ha sido ampliamente investigado, lo que ha generado la ausencia de un control específico para mitigar sus efectos, impactando directamente a los agricultores. La carencia de información detallada sobre las heladas dificulta la implementación de medidas preventivas adecuadas para proteger los cultivos ya que no cuentan con sistemas de alertas.(Cordero, 2013)

A pesar de que el Instituto Nacional de Meteorología e Hidrología (INAMHI) ofrece valiosa información sobre el clima, esta información no siempre llega a la población agrícola,

lo que resulta en una falta de prevención contra las heladas en los cultivos locales y por otro lado la implementación de sistemas de detección o prevención de heladas implica una gran inversión para los agricultores, quienes a menudo continúan utilizando métodos y técnicas antiguas de forma reactiva. (Sencrop, 2023).

Para abordar esta problemática y garantizar la protección y producción efectiva de los cultivos vulnerables, Se plantea la creación de un prototipo de sistema de alerta contra heladas, empleando componentes más asequibles que faciliten la recopilación automatizada de datos. Este sistema se basará en tecnología inalámbrica de largo alcance para proporcionar alertas precisas y oportunas a los agricultores, permitiéndoles tomar medidas preventivas de manera proactiva.

1.2. Justificación

Es fundamental señalar que la helada es un fenómeno que impacta a las plantas de forma física e irreversible. Sus efectos adversos se originan por la congelación del agua dentro de la planta, lo que puede provocar la formación de hielo tanto en el interior como en el exterior de las células. La formación de hielo fuera de las células resulta en una deshidratación de la planta y daño en el tejido, que puede ser permanente durante su crecimiento, también puede causar la muerte de algunas hojas, he incluso puede llevar a la muerte de la planta. Por esta razón, los cultivos pueden experimentar daños catastróficos si no se implementan alertas a tiempo, que permitan al agricultor tomar medidas preventivas de manera oportuna.(Bravo et al., 2020)

Las heladas representan un desafío significativo para la seguridad económica de Ecuador, en especial para la agricultura. En las últimas décadas, el país ha experimentado impactos devastadores debido a eventos climáticos extremos, como El Niño, heladas y sequías. Estos eventos han resultado en pérdidas sustanciales en términos de vidas humanas, infraestructura y actividad económica. Por ejemplo, las heladas y sequías ocurridas en el año 2005 también causaron graves daños económicos, especialmente en las zonas agrícolas de la

sierra ecuatoriana. Estos eventos climáticos adversos han afectado particularmente a los productores agrícolas minoritarios, quienes son propensos a ser vulnerables a estas condiciones. (MAE, 2014). Se estima que las pérdidas anuales podrían situarse entre 927 y 3.300 millones de dólares estadounidenses. Además, estos efectos adversos no solo afectan la economía del país, sino que también tienen un impacto profundo en la inseguridad alimentaria afectando a la calidad de vida y el bienestar de la población ecuatoriana en general.(MAATE, 2023)

En los últimos años, hemos sido testigos de un avance significativo en el desarrollo de sistemas tecnológicos aplicados al ámbito agrícola. La incorporación de tecnologías como el Internet de las cosas (IoT) y la inteligencia artificial (IA) han revolucionado la forma de afrontar problemáticas en la agricultura. Estos avances permiten la recolección de datos y el análisis de los mismos en tiempo real, lo que brinda a los agricultores información detallada sobre las condiciones climáticas, la salud de los cultivos y el rendimiento del suelo, entre otros aspectos clave.

Toda esta innovación ayuda a contribuir con el objetivo 5 del Plan de Desarrollo para un Nuevo Ecuador (2024 - 2025), “Fomentar de manera sustentable la producción mejorando los niveles de productividad” En resumen, este objetivo pretende proteger la industria agroalimentaria y pesquera nacional mediante la aplicación de estrategias financieras preferenciales y la promoción de tecnologías y prácticas ecológicas en la producción agrícola y ganadera. (SNP, 2024)

Además, este proyecto tiene un impacto significativo en varios objetivos de la Agenda 2030: Objetivos de Desarrollo Sostenible (ODS) aprobados por la ONU en 2015 (ONU, 2015). En primer lugar, contribuye al logro del ODS 2: Hambre Cero, al salvaguardar la seguridad alimentaria mediante la protección de los cultivos vulnerables contra las heladas, lo que garantiza la disponibilidad de alimentos nutritivos para las comunidades agrícolas. Asimismo,

aborda el ODS 13: Acción por el Clima, al emplear tecnologías innovadoras para mitigar los efectos del cambio climático en la agricultura, protegiendo los cultivos y fortaleciendo la resiliencia frente a eventos climáticos extremos. Por último, también contribuye al ODS 9: Industria, Innovación e Infraestructura, al impulsar el desarrollo y la implementación de tecnologías avanzadas en el sector agrícola, fomentando la innovación y la construcción de infraestructuras para una agricultura más sostenible y eficiente.

Por lo tanto, el desarrollo de un sistema de alerta basado en tecnología inalámbrica de largo alcance y algoritmos de aprendizaje supervisado permitirá a los agricultores recibir alertas precisas y oportunas sobre la posibilidad de heladas, lo que les permitirá tomar medidas preventivas de manera proactiva y proteger sus cultivos de manera efectiva. Esto ayudará a reducir las pérdidas en la producción agrícola y contribuirá al desarrollo sostenible del sector agrícola en el Ecuador.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema de alerta contra heladas para la protección de cultivos vulnerables utilizando arquitectura IoT y técnicas de IA en la Parroquia Rural Olmedo – Pesillo perteneciente al Cantón Cayambe.

1.3.2. Objetivos Específicos

- Elaborar un estado del arte sobre el proceso de los cultivos, la identificación y las consecuencias de los fenómenos meteorológicos, así como la arquitectura IoT, técnicas de inteligencia artificial y demás dispositivos a utilizar.
- Determinar los requerimientos de software y hardware del sistema, considerando las necesidades del usuario y el entorno de implementación, para garantizar su correcto funcionamiento.
- Desarrollar el sistema de alerta contra heladas, incluyendo el diseño y la programación de la red de sensores, así como la implementación Inteligencia Artificial.
- Evaluar el sistema propuesto realizando pruebas de funcionamiento para verificar su eficacia y correcto desempeño.

CAPÍTULO II. FUNDAMENTACIÓN TEÓRICA

Este capítulo proporciona una visión detallada de los aspectos fundamentales relacionados con la agricultura y su impacto socioeconómico. Se examinan las características geográficas y climáticas particulares de la región, así como los cultivos principales. Se estudian los desafíos que enfrenta la agricultura en esta área, con un enfoque especial en las amenazas derivadas de los cambios climáticos y la variabilidad meteorológica, especialmente las heladas. Además, se lleva a cabo una investigación sobre la tecnología de redes de sensores inalámbricos (WSN) y las redes de baja potencia y área amplia (LPWAN), resaltando su importancia en la recopilación de datos agrícolas y la transmisión de información esencial para la gestión de cultivos. Esta recopilación bibliográfica establecerá una base sólida para comprender la importancia y la necesidad de desarrollar un sistema de alerta contra heladas basado en tecnologías emergentes como el IoT y la inteligencia artificial.

2.1. Introducción a la Agricultura

Indudablemente la agricultura desempeña un papel fundamental en la vida humana al satisfacer una variedad de necesidades esenciales para la supervivencia. Además, la agricultura interviene en procesos clave como los ciclos del agua y de nutrientes, el control de la temperatura y de la erosión, entre otros aspectos cruciales para el equilibrio ecológico y la sostenibilidad del medio ambiente. (Sarandón, 2020).

Ecuador alberga una amplia gama de cultivos, desde los tropicales en las regiones costeras hasta los de alta montaña en la región andina los cuales son importantes para la seguridad alimentaria y sostenibilidad económica del país.

2.1.1. Características geográficas y climáticas de la Sierra

La región Sierra del Ecuador se extiende a lo largo de unos 800 km y abarca una amplia gama de paisajes, desde montañas hasta valles y páramos. En esta área, la Cordillera de los Andes tiene un gran trayecto, con la cordillera Occidental al oeste y la Oriental al este,

presentando elevaciones notables como los volcanes Imbabura y Cayambe. La región es conocida por su diversidad climática, que varía desde tropical en las áreas bajas hasta fría en los páramos de alta montaña, con estaciones lluviosas alternas entre marzo-abril y octubre-noviembre (Varela & Ron, 2018). La agricultura en esta región se ve influenciada por factores geográficos y climáticos únicos, que van desde la fertilidad de los suelos hasta la disponibilidad de agua y la altitud.

2.1.2. Importancia económica y social de la agricultura de la región

La agricultura es vital para la economía, es reconocida como un pilar esencial para el progreso de la región, proporcionando empleo a una gran parte de la población y contribuyendo significativamente al ingreso nacional representando en promedio un 8.5% del PIB del país. Además, fomenta el intercambio comercial con otros países y produce un exceso de productos que se venden localmente, lo que ayuda al desarrollo económico. (Andrade, 2017).

2.1.3. Proceso de Cultivos

Cada cultivo sigue una serie de procedimientos para lograr un crecimiento eficiente y exitoso de las plantas. Todo comienza con la preparación del suelo, que implica ararlo para aflojar la tierra y eliminar las hierbas no deseadas. Luego, se abona el terreno para enriquecerlo, utilizando abonos orgánicos como el estiércol o productos químicos preparados. A continuación, se preparan surcos en la tierra, conocidos como guachos, que sirven como guía para plantar las semillas previamente seleccionadas.

Durante el crecimiento de la planta, es crucial mantener el área libre de malezas y realizar fumigaciones para prevenir plagas y enfermedades. A medida que la planta madura, comienza a producir frutos, que se cosechan según el tipo de cultivo y el momento óptimo de consumo.

2.1.4. Principales Cultivos de la Región

Los productos que se siembran en la región Sierra son muy diversos como las papas, el maíz, la quinua, la cebada, el trigo, las hortalizas, las frutas y las flores. (MIES, 2020). Estos cultivos son cosechados en diferentes áreas de la región, aprovechando las condiciones climáticas y geográficas variadas que ofrece, lo que permite una diversificación y especialización en la producción agrícola.

2.1.5. Cultivos Vulnerables e Identificación.

Los cultivos vulnerables en la región de la Sierra ecuatoriana son aquellos que están expuestos a riesgos climáticos y fenómenos meteorológicos extremos, como las heladas. Se consideran vulnerables y se pueden identificar aquellos cultivos que tienen una baja tolerancia a las bajas temperaturas y son propensos a sufrir daños durante eventos de heladas. Entre los cultivos identificados como vulnerables se encuentran las papas, el maíz, la quinua y otras hortalizas de ciclo corto (El Universo, 2022).

2.2. Amenazas y Desafíos para la Agricultura

La agricultura enfrenta varios desafíos que dependen de diversos factores como el entorno geográfico, climático y económico. (Chávez, 2021) Tomando, así como las principales amenazas al cambio climático y variabilidad meteorología los cuales pueden afectar al crecimiento de los cultivos. Además, la presencia de plagas y otras enfermedades representan una amenaza constante para los cultivos.

2.2.1. Cambios Climáticos y variabilidad meteorológica

Estos eventos climáticos pueden presentarse de diversas formas, que van desde el aumento de la temperatura media hasta modificaciones en los ciclos de lluvia, pasando por fenómenos extremos como sequías, inundaciones, tormentas y heladas, entre otros (Toulkeridis et al., 2020). Estas variaciones pueden tener efectos adversos en los cultivos, afectando su rendimiento, calidad y disponibilidad.

2.2.2. Heladas

Una helada meteorológica se produce cuando la temperatura ambiental desciende a 0 °C o menos, resultando en daños para los cultivos. Desde una perspectiva agrometeorológica, se define como el cambio abrupto de temperatura que afecta los tejidos de las plantas. La magnitud de los daños depende de la intensidad y duración de las bajas temperaturas, así como de factores como la topografía y el tipo de cultivo. Incluso la helada agronómica, que no siempre alcanza los 0 °C, puede ocasionar daños considerables en los cultivos (Bravo et al., 2020).

2.2.3. Clasificación de las Heladas

Las heladas pueden clasificarse de diversas maneras según diferentes criterios, como su intensidad, duración, momento de ocurrencia y condiciones atmosféricas asociadas. Una de las clasificaciones más utilizadas se fundamenta en la temperatura del aire y su correspondencia con el punto de congelación. Esta categorización se presenta en la Tabla 1.

Tabla 1

Clasificación de las Heladas

Tipos de Helada	Descripción
Advección	Se caracterizan por la entrada de aire frío con temperaturas bajo cero. El viento es importante y no hay inversión térmica, lo que significa que la temperatura disminuye con la altitud. Además, los vientos suelen superar los 15 km/h.
Radiación	Ocurren debido a la variación de temperatura entre las capas inferiores y superiores de la atmósfera. Se dan en condiciones de poco viento y cielos despejados, permitiendo la pérdida de calor desde la superficie.
Evaporación	Surgen cuando la humedad se reduce y el rocío se evapora de la superficie de la planta, lo que provoca su enfriamiento. Y esta puede diferenciarse en dos categorías.

	• Heladas blancas: Se distinguen por la formación de hielo en las plantas y suelen presentarse en condiciones de aire húmedo, vientos calmados y cielos despejados. • Heladas negras: ocurren cuando el aire está seco. La presencia de nubes en el cielo o una ligera turbulencia en las capas bajas de la atmósfera favorecen su aparición.

Fuente: Adaptado de (Sencrop, 2022)

2.2.4. Efectos de las Heladas

Las heladas pueden dañar los cultivos directa e indirectamente, afectando negativamente en el proceso de floración y fructificación, así como aumentar la susceptibilidad de las plantas a enfermedades y plagas. Estos efectos se detallan en la Tabla *2*.

Tabla 2

Efectos de las Heladas en los Cultivos

Efectos	Descripción
Consecuencias de Impacto	Externos: provoca la muerte de hojas, tallos y una parte importante de flores y frutos en desarrollo, puede causar la muerte total de la planta. Internos: la deshidratación provocada por el crecimiento de cristales de hielo dentro de la célula resulta en la ruptura de las membranas celulares.
Por tiempo de exposición	Inmediatos: sus consecuencias incluyen la deshidratación y la ruptura de la membrana de la planta. Acumulativos: Si se repite, puede intoxicarla con sales minerales y cuanto más tiempo, más daño puede tener la planta.

Fuente: Adaptado de (PROAIN, 2020)

2.2.5. Mitigación de Heladas

Son medidas tomadas para reducir o prevenir los efectos negativos de las heladas en los cultivos y en el entorno agrícola en general. Esto puede implicar el uso de técnicas como el riego por aspersión continua, la aplicación de mallas térmicas, el uso de sistemas de calefacción, entre otros métodos (Netafim, 2023). Estas se aplican con el fin de mantener las plantas a una temperatura adecuada y protegerlas de los daños causados por las heladas.

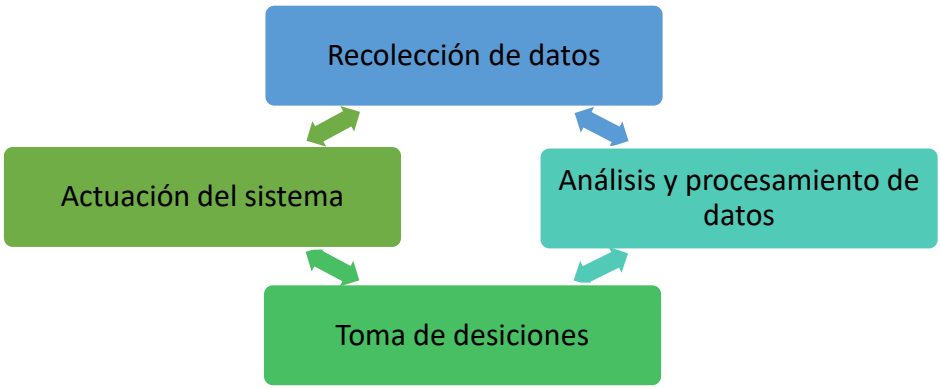
2.3. Agricultura de Precisión

Es la agrupación de técnicas agrícolas que emplean tecnologías de información y sistemas de información geográfica (SIG), además de sistemas de posicionamiento global (GPS) que buscan mejorar la eficiencia y el rendimiento en la producción agrícola. El objetivo es permitir la aplicación precisa de productos como agua, fertilizantes y pesticidas, y técnicas de protección contra fenómenos o plagas reduciendo el desperdicio y mejorando la calidad de los cultivos. (BBVA, 2023)

Se fundamenta en recopilar y el analizar datos sobre las condiciones del suelo, las plantas y las condiciones climáticas para tomar decisiones personalizadas en cada etapa del proceso agrícola. Las etapas para la agricultura de precisión pueden variar según el enfoque y la aplicación específica, pero generalmente incluyen las etapas de la Figura 1.

Figura 1

Etapas Agricultura de precisión



Fuente: Adaptado de (GRAP, 2022)

2.3.1. Sistema de Medición Meteorológica

El sistema de medición meteorológica comprende una serie de instrumentos que recopilan datos sobre las condiciones atmosféricas y climáticas en un área específica (Palaguachi, 2018). Estos dispositivos pueden abarcar termómetros para la temperatura, barómetros para la presión atmosférica, anemómetros para medir la velocidad viento, y pluviómetros para la cantidad de lluvia, entre otros. Estos datos son cruciales para comprender y prever el clima y las condiciones atmosféricas.

2.3.2. Factores o Variables por Medir

Existen varias variables que influyen en la generación de heladas, como se detallan en la Tabla 3. Estas variables serán consideradas para desarrollar el prototipo. Es crucial definir rangos específicos para cada una de estas variables, lo que indicará las condiciones propicias para la formación de heladas.

Tabla 3

Variables por medir y rangos de intensidad en heladas

Intensidad de Helada	**Variable**			
	Temperatura del aire	**Humedad relativa**	**Velocidad del viento**	**Radiación UV**
Alta	-6°C a -3°C	Menor a 20%	Mayor o igual a 15 km/h	Menor a 0.1
Moderada	-3°C a 0°C	Entre 20% y 40%	7 a 14 km/h	1.5 a 3
Baja	0°C a 3°C	Entre 40% y 60%	4 a 6 km/h	3 a 5
Muy baja	3°C a 6°C	Mayor a 60%	Menor a 4 km/h	5 a 7

Fuente: Adaptado de (Xunta de Galicia, 2019)

2.4. Internet de las Cosas (IoT)

IoT hace referencia a la tendencia de utilizar dispositivos que se conectan a un sistema para recopilar información de forma automática y sin intervención humana. Estos dispositivos, integrados con tecnología informática, pueden transferir datos a través de redes inalámbricas. El IoT ofrece mejoras en la eficiencia y la conectividad, pero también plantea desafíos como la gestión del gran volumen de datos generados (big data) y la seguridad de la información. (Flores & Gonzalo, 2021)

En la actualidad, IoT tiene diversas aplicaciones en campos como la agricultura, educación, salud, entre otros. Esta tecnología representa tanto el presente como el futuro, mejorando la calidad de vida y simplificando las actividades cotidianas al ofrecer comodidades. Se espera que, en el futuro, ciudades enteras formen parte de una extensa red de información basada en el IoT.

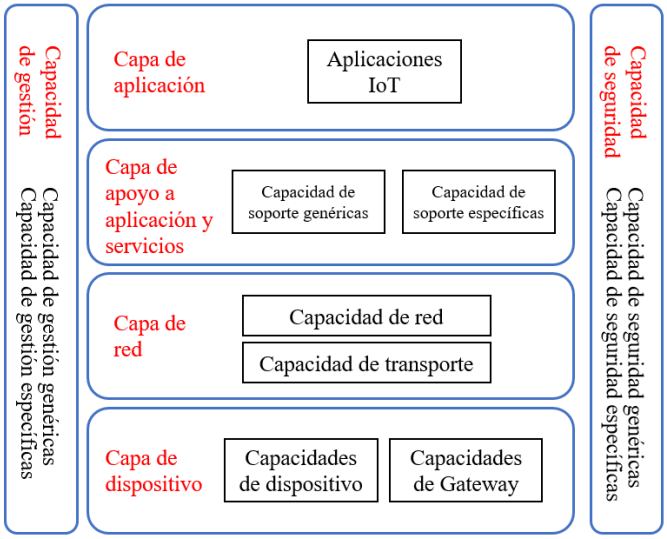
2.4.1. Arquitectura de Internet de las cosas (IoT)

Este ofrece varias arquitecturas adaptables entre la más común esta especificada la de 4 capas, como ejemplificación de la Figura 2. Cada una de las capas cumple con una función específica que complementa al sistema y según Eterovic et al. (2019) describe cada una de las capas como:

- Capa de Dispositivo: Recolecta datos del entorno a través de sensores y dispositivos embebidos.
- Capa de Red: Transmite datos a través de redes inalámbricas como Bluetooth, Wifi o celulares.
- Capa de apoyo a aplicación y servicios: Procesa datos utilizando algoritmos para extraer información relevante.
- Capa de Aplicación: Utiliza los resultados del análisis para tomar decisiones y ofrecer servicios a los usuarios finales.

Figura 2

Modelo de Arquitectura IoT



Fuente: Obtenido de (UIT, 2012)

2.4.2. Protocolos para IoT

Los protocolos de comunicación en IoT tienen un papel fundamental en garantizar la seguridad de la información entre dispositivos conectados. Se dividen en dos categorías: los protocolos de red, tales como HTTP, Bluetooth, ZigBee y LoRaWAN, que posibilitan la transmisión de datos en la red, y los protocolos de datos, como MQTT y CoAP, diseñados específicamente para dispositivos de baja potencia (Libson, 2022).

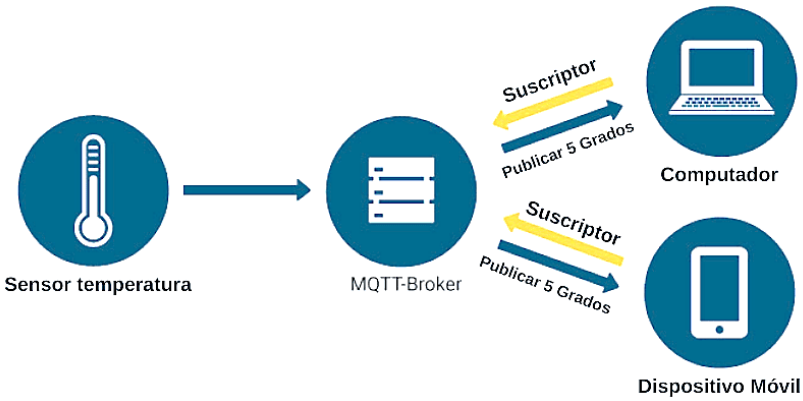
2.4.2.1. Protocolo MQTT

MQTT, acrónimo de MQ Telemetry Transport, ha ganado popularidad en el ámbito del IoT gracias a su sencillez y eficiencia para dispositivos con recursos limitados. Funciona mediante TCP/IP y adopta un modelo de publicación/suscripción (pub-sub), en el que los dispositivos se conectan a un servidor central llamado bróker, como se ilustra en la Figura 3. Los mensajes se organizan en temas (topics), lo que permite que los dispositivos publiquen y suscriban mensajes según sus necesidades (Llamas, 2019). La conexión entre dispositivos y broker permanece abierta, empleando mensajes como CONNECT, PUBLISH, SUBSCRIBE y

UNSUBSCRIBE para la comunicación, y se incluyen mensajes PINGREQ y PINGRESP para mantener la conexión activa (Bassi, 2021).

Figura 3

Esquema de conexión básico de MQTT



Fuente: Obtenido de (Solutec, 2020).

2.4.3. Plataformas de Servicio IoT

La plataforma IoT es un sistema que recopila y procesa datos mediante dispositivos integrados y protocolos de comunicación IoT, basándose en el concepto de inteligencia ambiental (AmI). Esto facilita la creación de sistemas de monitoreo en paneles de control que permiten el análisis y control de eventos para gestionar infraestructuras. Esta tecnología es útil tanto en entornos industriales como en situaciones que requieren la vigilancia de personas.(Ferrandez et al., 2021).

2.5. Sistemas Embebidos

Son sistemas electrónicos diseñados para funciones específicas, a diferencia de las computadoras generales. Están optimizados para tareas particulares y ofrecen un rendimiento superior en su entorno previsto. Se utilizan en diversas aplicaciones, como dispositivos médicos, automotrices y domésticos inteligentes, impulsando la innovación y la eficiencia en varias industrias (Chetto & Queudet, 2020).

2.5.1. Componentes de un sistema Embebido

Los sistemas embebidos integran una variedad de componentes esenciales, tales como microcontroladores, memoria e interfaces de entrada/salida, entre otros, que colaboran para llevar a cabo funciones específicas con eficiencia y fiabilidad. La selección y configuración correctas de estos elementos son críticas para garantizar el éxito en el diseño de sistemas embebidos. A continuación, se nombran algunos de estos componentes.

2.5.1.1. Interfaces de entrada/salida (E/S):

Estos componentes facilitan la interconexión entre el sistema embebido y su entorno externo, posibilitando la comunicación con sensores, actuadores y diversos dispositivos periféricos. Su correcto funcionamiento garantiza una integración efectiva y un intercambio fluido de datos dentro del sistema.

2.5.1.2. Microcontroladores y Microprocesadores

Son dispositivos de versión compacta de una computadora, integra una CPU, RAM y ROM, desempeñando un papel crucial en la automatización industrial y diversas aplicaciones cotidianas. (SIISA GLOBAL, 2021). Estos son los encargados de ejecutar las instrucciones y controlar las operaciones. Algunas placas se describen a continuación:

- **ESP32:** Es un sistema incorporado de bajo consumo. Incorpora un microprocesador y se caracteriza por su eficiencia energética y su avanzada integración con componentes. (Bruno, 2019)
- **Arduino Uno:** Facilita el desarrollo de hardware de código abierto. Cuenta con entradas y salidas digitales y analógicas que permiten la conexión y control de dispositivos y sensores (Freire, 2023).
- **Rasperry Pi:** Placa ideal para diversas actividades como navegación web, programación y creación de dispositivos. A pesar de su tamaño reducido, es potente y puede realizar la mayoría de las tareas de un PC estándar. (Halfacree, 2020)

2.5.1.3. Fuente de alimentación

Esencial para el funcionamiento del sistema, la alimentación proporciona la energía necesaria a través de baterías o fuentes externas. Garantizar una alimentación estable y adecuada es crucial para el rendimiento óptimo del sistema embebido.

2.6. Redes de Sensores Inalámbricos (WSN)

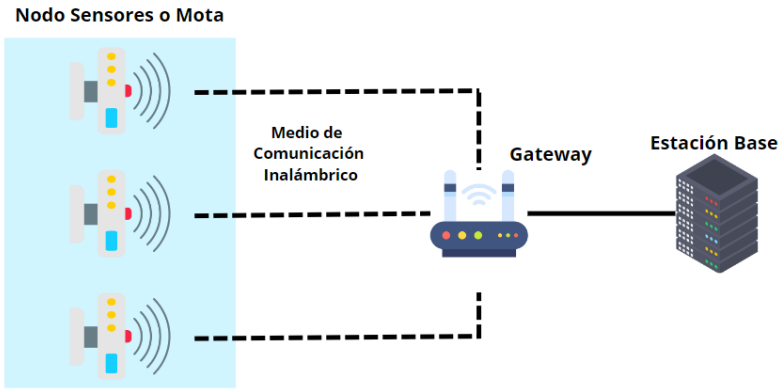
Son sistemas compuestos por dispositivos sensoriales que se destacan por su versatilidad, rendimiento energético y aptitud para medir parámetros físicos en tiempo real, además de poder comunicarse a corta distancia. Son muy adecuadas para recolectar información en entornos específicos (Cajas, 2022).

2.6.1. Arquitectura de la WSN

La estructura de una red de que integra sensores se fundamenta en cuatro componentes: los nodos sensores, las puertas de enlace, el medio de comunicación y la estación base. La exigencia de hardware y software de estos elementos varía según la aplicación específica de la red de sensores inalámbricos. Esta arquitectura se representa en la Figura 4.

Figura 4

Arquitectura WSN



Fuente: Adaptado de (Rodríguez, 2023)

Rodríguez (2023) describe los componentes de la red WSN de la siguiente manera:

2.6.1.1. Nodos Sensores:

Estos dispositivos electrónicos integran funciones de procesamiento, almacenamiento, sensores, comunicación inalámbrica y fuente de energía. Algunos de estos nodos recolectan información del medio, como en el caso de:

- Sensor de Humedad y temperatura: Detectan y miden tanto la temperatura como la humedad relativa del entorno donde está el sensor.
- Sensor de radiación UV: Detecta y mide la radiación ultravioleta en el ambiente, utilizado en diversas aplicaciones como la monitorización ambiental.
- Sensor de Velocidad del Viento: Utiliza diferentes tecnologías, como anemómetros de copa, ultrasónicos o de hélice, para registrar la velocidad del viento en unidades como (m/s) o (km/h).

2.6.1.2. Gateway:

La puerta de enlace conecta una Red WSN con una red de área extensa (WAN), permitiendo la transferencia de datos para su acceso por aplicaciones y usuarios (Romero, 2024).

2.6.1.3. Medio de comunicación

Se hace referencia a la tecnología que emplea señales electromagnéticas, como radiofrecuencias, infrarrojos o microondas, para transferir información entre dispositivos sin requerir una conexión física directa.

2.6.1.4. Estación base

La información procedente de la puerta de enlace se guarda y procesa en un ordenador o sistema embebido, lo que posibilita su visualización y el acceso remoto o local para supervisar la red de sensores inalámbricos.

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

Es una tecnología de comunicación inalámbrica de largo alcance, desarrollada para conectar dispositivos con bajo consumo energético y capacidad de transmisión limitada a distancias significativas. Estas redes están especialmente diseñadas para aplicaciones de comunicación entre máquinas (M2M) y el Internet de las Cosas (IoT). A diferencia de las redes móviles tradicionales, las LPWAN destacan por su eficiencia energética y costos operativos reducidos, permitiendo además la conexión de un gran número de dispositivos al mismo tiempo en áreas extensas. (Qin et al., 2021)

2.7.1. Tecnología LoRa

Es una tecnología inalámbrica que se fundamenta en un sistema de radiofrecuencia, dando origen al protocolo LORA-WAN. Sus aspectos esenciales incluyen una baja demanda energética, costos de instalación económicos, autonomía de dispositivos, una arquitectura de implementación sencilla, seguridad en la transmisión de datos y una modulación adecuada para evitar interferencias (Chiriboga, 2020).

2.7.2. LoRaWAN

Es un protocolo de red de baja potencia y amplio alcance (LPWA) creado para vincular dispositivos alimentados por batería a Internet en redes de alcance regional, nacional o global, centrándose en aspectos fundamentales de IoT como la conectividad bidireccional, la seguridad integral, la movilidad y la localización. (LoRa Alliance®, 2022)

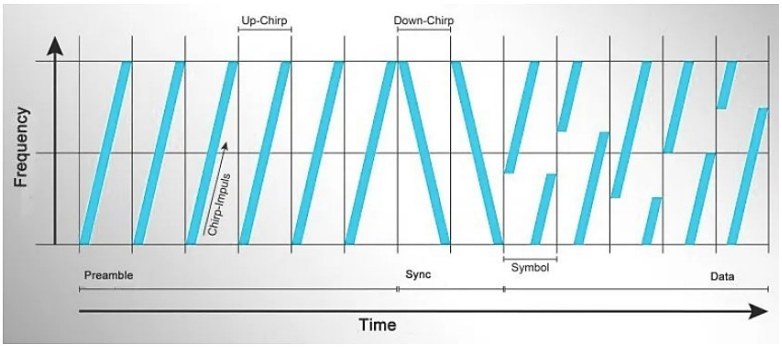
2.7.3. Modulación LoRa

Esta tecnología inalámbrica, que emplea la modulación Chirp Spread Spectrum (CSS), facilita la transmisión de datos a largas distancias mediante pulsos de chirp, los cuales se caracterizan por tener una amplitud constante y una frecuencia en continuo cambio. Dependiendo de la dirección de variación de la frecuencia, estos pulsos se clasifican en up-chirp y down-chirp, lo que permite la alteración progresiva de la frecuencia en LoRa. Como se

muestra en la Figura 5, durante la emisión de un mensaje LoRa, primero se genera un preámbulo prolongado con un chirp constante para detectar la señal, seguido de chirps inversos que sincronizan el tiempo de transmisión. Finalmente, la información es enviada a través de señales moduladas que alternan entre diferentes frecuencias (Ramírez, 2023).

Figura 5

Modulación LoRa



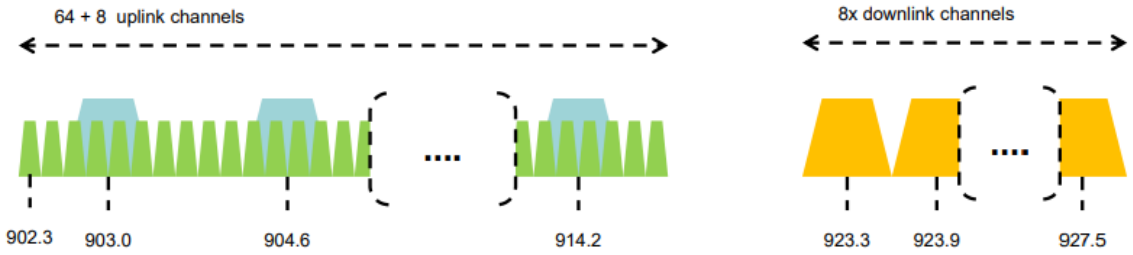
Fuente: Obtenido de (Panza, 2024)

2.7.4. Radiofrecuencia y canales

LoRa funciona en el espectro sin licencia, empleando la banda ISM, mayormente por debajo de 1 GHz, lo que facilita la comunicación a grandes distancias. Las frecuencias particulares difieren según la región; por ejemplo, en América Latina y Ecuador se emplea la banda US902-928 MHz. Esta banda se segmenta en canales específicos, como 64 canales de 125 kHz de ancho en uplink y 8 canales de 500 kHz de ancho en downlink (LoRaWAN®, 2022; Ramírez, 2023). La Figura 6 presenta de forma visual canalización en el espectro y la distribución de frecuencias, mostrando los canales de subida (uplink) y bajada (downlink) junto con sus respectivas frecuencias.

Figura 6

Plan de canalización para la banda US902-928



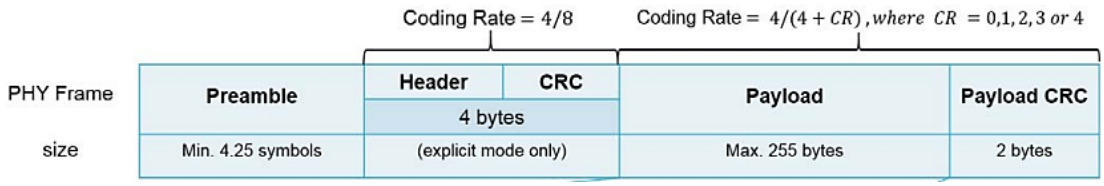
Fuente: Obtenido de (LoRaWAN®, 2022)

2.7.5. Trama Física de la tecnología LoRa

La estructura de LoRa apreciada en la Figura 7, inicia con un preámbulo que contiene una secuencia constante de upchirps que abarcan todo el rango de frecuencia, seguido por upchirps que contienen información sobre la palabra de sincronización, la cual distingue entre las redes LoRa que comparten las mismas bandas de frecuencia. Posteriormente, se presentan downchirps que señalan el término del preámbulo. Después, puede incluirse un encabezado opcional que transmite detalles sobre el tamaño de la carga útil, la velocidad de código y la presencia de un CRC de 16 bits. La extensión de la carga útil se almacena en un byte, limitándola a 255 bytes.(Cevallos & Rubio, 2021; Narváez & Contreras, 2020)

Figura 7

Trama Física LoRa



Fuente: Obtenido de (Narváez & Contreras, 2020)

2.7.6. Tiempo en el Aire del Paquete

Es el valor que indica el tiempo necesario para completar la transmisión de un paquete, en otras palabras, el período que transcurre desde su envío hasta su recepción. Su determinación

está influenciada por parámetros como BW (Bandwidth), CR (Coding Rate) y SF (Spreading Factor). Para calcular con precisión la duración de la transmisión, se puede utilizar la herramienta de estimación desarrollada por Semtech, empresa responsable de la tecnología LoRa (Ramírez, 2023). Se determina mediante la ecuación (1):

$$ToA = T_{Preambúlo} + T_{Payload} \quad (1)$$

2.7.7. Clases de dispositivos

Según Cadena (2020) las clases de dispositivos son los siguientes: Clase A tiene comunicación bidireccional asimétrica; pueden enviar datos en cualquier momento, pero solo recibir después de una transmisión, seguido de un período de recepción. Clase B contiene horarios de recepción programados para recibir datos en intervalos específicos, sincronización con transmisiones periódicas del servidor para comunicación más predecible y por ultimo los de Clase C que permiten recibir datos del servidor en cualquier momento excepto durante la transmisión, ideal para alta disponibilidad de datos, pero con mayor consumo energético.

2.7.8. Seguridad y Modos de Activación

Existen dos métodos de acceso, ABP y OTAA, que difieren en la forma de establecer claves de sesión y la dirección del nodo. Ambos requieren el almacenamiento de claves como AppKey, DevAddr, NwkSkey y AppSkey. Para la seguridad, LoRaWAN utiliza AES-128, con Network Session Key y Application Session Key para proteger las comunicaciones de red y de extremo a extremo, respectivamente; la Application Key se usa en despliegues OTAA (González, 2019).

2.8. Lenguajes de Programación

Son agrupaciones de reglas y símbolos que permiten instruir a las computadoras de manera precisa. Estos lenguajes difieren en su grado de abstracción, organización y reglas sintácticas, y se emplean en el desarrollo de diversas aplicaciones, que van desde sistemas operativos hasta aplicaciones web y móviles.

2.8.1. Python

Es un lenguaje de programación con características de alto nivel y multiparadigma. Reconocido por su versatilidad, facilidad de aprendizaje y la gran cantidad de bibliotecas y frameworks disponibles que facilitan el desarrollo de aplicaciones complejas. (Londaña, 2023).

2.9. Base de Datos

Es un sistema centrado y programado para recopilar, almacenar y gestionar datos de manera adecuada, con el fin de facilitar su posterior consulta, actualización y análisis. Se utiliza en aplicaciones y sistemas informáticos para almacenar información de manera eficiente y segura, permitiendo a los usuarios recuperar y manipular los datos según sea necesario (Microsoft, 2021). Actualmente, existen varias bases de datos gratuitas relacionadas con las plataformas de IoT.

2.10. Inteligencia Artificial

Se basa en como enseñarle a una computadora a pensar y tomar decisiones como lo haría un ser humano. Esto implica que la computadora puede aprender de la información que recibe, resolver problemas, reconocer patrones y hasta entender y responder preguntas en lenguaje humano (Reventós, 2019). La IA abarca diferentes de técnicas, incluyendo el aprendizaje automático, la visión por computadora, entre otros.

2.10.1. Aprendizaje Supervisado y no Supervisado

En el aprendizaje supervisado, se entrena un modelo utilizando datos que están etiquetados, lo que significa que cada muestra tiene un identificativo que indica la salida deseada. Esto permite que el modelo aprenda cómo relacionar las entradas con las salidas conocidas y hacer predicciones en nuevos datos (MathWorks, 2023). En contraste, el aprendizaje no supervisado utiliza datos sin etiquetar para entrenar el modelo, permitiéndole identificar patrones o relaciones ocultas dentro de la información sin depender de etiquetas predefinidas como referencia (INESDI, 2022).

2.10.1.1. Técnicas de Aprendizaje Supervisado

Los diferentes algoritmos son entrenados según la necesidad y perspectiva del proyecto en cuestión. (AWS, 2023) describe algunos de estos algoritmos de la siguiente manera:

- Regresión lineal: Es un método estadístico que permite modelar la relación entre variables dependiente y variables independientes, ajustando una línea recta que representa la tendencia de los datos.
- Árboles de decisión: Este algoritmo divide en conjuntos muestras creando ramificaciones, lo que permite tomar decisiones basadas en reglas simples.
- Redes neuronales: Modelan relaciones complejas entre las características de entrada y la salida utilizando una estructura de red de neuronas interconectadas.

2.10.2. Métricas de desempeño.

Para evaluar la efectividad de los algoritmos de aprendizaje en un conjunto de datos particular, resulta crucial emplear una gama diversa de métricas. Estas métricas pueden abarcar la exactitud, la exhaustividad, la precisión, el área bajo la curva ROC (Receiver Operating Characteristic), la matriz de confusión, entre otras. Cada métrica ofrece detalles específicos sobre el rendimiento del modelo en cuanto a su capacidad predictiva y de generalización (Singh, 2023).

2.11. Sistemas de Alertas

Un sistema de alerta es una llamada al usuario mediante procesos y tecnologías diseñadas para notificar tiempo real y donde se advierte al usuario algún evento en el cual pueda tomar medidas adecuadas, mitigar cualquier riesgo o impacto negativo asociado con el evento o la situación.

2.11.1. Método de envío de alerta

Los métodos de envío de alertas son diferentes formas de notificar a los usuarios sobre eventos importantes o situaciones críticas. Algunos de estos métodos incluyen:

Correo Electrónico: El envío de alertas a través de mensajes enviados a las direcciones especificadas este es un servicio en línea, teniendo una red que permite el envío y recepción. (INCIBE, 2021)

- Mensajes de Texto (SMS): Son una forma rápida y directa de comunicarse con los usuarios, sobre todo cuando hay que informarles inmediatamente.
- Notificaciones Push: Estas aparecen directamente en la pantalla del dispositivo del usuario, sin necesidad de abrir una aplicación específica. Son muy efectivas para captar la atención del usuario de manera inmediata. (IBM, 2021)

CAPÍTULO III. DESARROLLO DE LA PROPUESTA

Este capítulo se centra en el desarrollo de la metodología elegida para la construcción del proyecto, donde se exploran en detalle los requisitos del sistema y se realiza un análisis de los interesados involucrados, aquellos actores que tienen una relación con la implementación y resultados del sistema. Además, se incluye una investigación de la situación actual del escenario donde se va a implementar el sistema, con el fin de obtener una comprensión completa del entorno y las condiciones específicas que influirán en su funcionamiento. Así mismo, se selecciona los diferentes componentes disponibles para el proyecto, evaluando sus características y funcionalidades. Por último, se proporciona una documentación detallada del proceso de diseño del prototipo e implementación del sistema de alerta contra heladas.

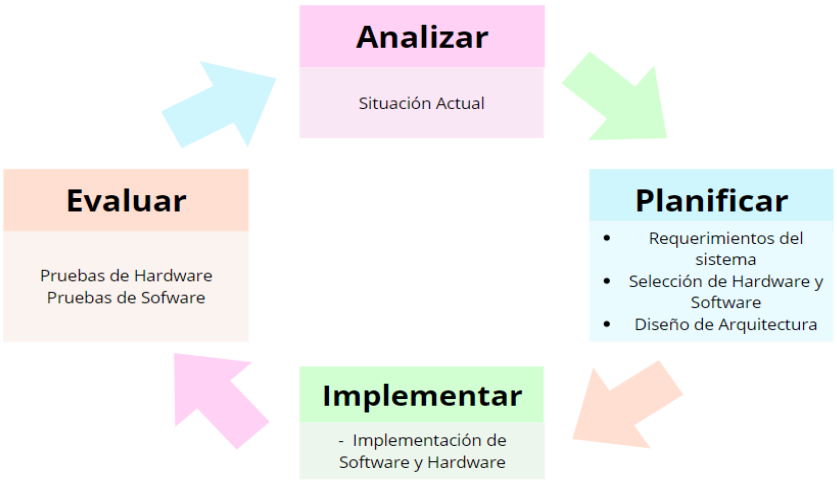
3.1. Metodología

En este proyecto, se utiliza la metodología de investigación-acción, también conocida como "Action Research", esta permite una colaboración estrecha y continua entre estos actores, Lo que permite reconocer dificultades y optimizar el proceso de decisión informadas y la adaptación ágil a las necesidades cambiantes del entorno. Su objetivo es generar conocimiento práctico y soluciones efectivas mediante un ciclo de análisis, planificación, implantación y evaluación de resultados (Wiater, 2022).

El propósito central de este proyecto es desarrollar un dispositivo capaz de identificar la presencia de heladas en áreas de cultivo susceptibles mediante la implementación de una red con infraestructura basada en IoT y el uso de técnicas avanzadas de IA para el análisis y clasificación de datos recopilados. La Figura 8 exhibe detalladamente las distintas etapas de la metodología adoptada al proyecto y descrita a lo largo del presente documento.

Figura 8

Metodología "Action Research"



Fuente: Adaptado de (Wiater, 2022)

Este ciclo de cuatro pasos se ajusta perfectamente al proyecto, facilitando el logro de los objetivos establecidos. Cada uno de estos pasos se integra de la siguiente manera:

- Analizar: Simplifica la definición del tema, lo que lleva al desarrollo de la idea principal mediante un análisis de investigaciones relacionadas con el objetivo compartido. Esta etapa refuerza la obtención de resultados y ayuda al cumplimiento de los objetivos 1 y 2 al tratar la problemática y comprender la situación actual.
- Planificar: Esta fase implica seguir un conjunto de pasos sistemáticos para cumplir con el objetivo 2 y el 3. Se exploran diversas opciones para establecer los requisitos del sistema que guiarán la selección de los componentes necesarios para el funcionamiento del proyecto, priorizando su orden de acuerdo con las necesidades de este.
- Implementar: Se ejecuta las acciones planificadas y diseñadas en las etapas anteriores. Esto incluiría la construcción y configuración del sistema de alerta contra heladas utilizando la arquitectura IoT y las técnicas de inteligencia artificial, además de la incorporación de los elementos requeridos para su funcionamiento.

- Evaluar: Se lleva a cabo pruebas para garantizar el correcto funcionamiento y desempeño del sistema en situaciones reales, Además, se efectúan ajustes según sea necesario para optimizar su rendimiento.

3.2. Etapa de Análisis

En esta etapa, se profundiza en el desarrollo de la investigación realizada en el capítulo anterior, llevando a cabo un análisis de la situación actual. Este proceso no solo busca comprender la solución el proyecto, sino también establecer claramente su alcance y objetivos. De esta manera, se sientan las bases necesarias para avanzar de manera efectiva hacia las siguientes etapas del proyecto.

3.2.1. Situación actual

En la agricultura actual, los cambios climáticos representan un desafío significativo, ya que pueden ocasionar retrasos en el crecimiento de las plantas y disminuir la productividad de los cultivos. Sin embargo, Uno de los inconvenientes más relevantes es la ausencia de alertas para los agricultores ante la amenaza de las heladas, lo que dificulta la adopción de medidas preventivas.

Este sistema de alertas se desarrolla en la parroquia se Olmedo – Pesillo perteneciente al cantón Cayambe de la provincia Pichincha ubicado a 3600 m.s.n.m., Se distingue por un ambiente frío y con alta humedad, donde las temperaturas oscilan entre los 4° y 15 °C. Este entorno climático presenta desafíos significativos para la agricultura local, especialmente para los cultivos vulnerables a las heladas. El sistema quedará desplegado en una superficie de 3 hectáreas, con solo $1000m^2$ destinados al cultivo visualizados en la Figura 9.

Figura 9

Área de terreno de aplicación

Nota: Se selecciona la zona de cultivo de 1000m^2 representada por el cuadro azul para la implementación del sistema. **Fuente:** Adaptado de Google Earth

En la representación gráfica proporcionada por la Figura 10, se delinean las zonas de interconexión del sistema, las cuales han sido cuidadosamente seleccionadas en función de la extensión del cultivo y la distancia al Gateway, con el objetivo de optimizar la transmisión de datos y minimizar posibles interferencias. En este contexto, la implementación de una única estación de sensores se presenta como la opción más eficiente, ya que permite centralizar la recopilación de información, reduciendo la redundancia de datos y mejorando la precisión de las mediciones. Además, esta configuración facilita el mantenimiento y la calibración de los sensores, asegurando un monitoreo continuo y confiable de las variables ambientales que influyen en el desarrollo del cultivo.

Figura 10

Ubicación de la estación y la zona del Gateway

Fuente: Adaptado de Google Earth

3.3. Etapa de Planificar

Se inicia con el establecimiento de la nomenclatura y los requisitos necesarios para cubrir las necesidades del sistema y usuario. Este establecimiento se basa en la norma ISO/IEC/IEEE 29148 la cual proporciona directrices para la especificación de requisitos de software y sistemas, incluyendo el establecimiento de nomenclatura y requerimientos.

Posteriormente, se procede a seleccionar los elementos que serán utilizados en el sistema, basándose en estos requisitos. Este proceso de planificación se lleva de forma sistemática cumpliendo con el objetivo 2.

3.3.1. Determinación de Stakeholders

Es crucial considerar las necesidades y expectativas de los involucrados en cada etapa del proyecto, desde la definición de requisitos hasta su ejecución y evaluación, con el objetivo de garantizar su satisfacción y el éxito del mismo. Los stakeholders tienen un papel esencial en todas las etapas del proyecto, asegurando la pertinencia y eficacia de la solución propuesta. Se

ha elaborado un listado detallado de estos participantes, Incluye a aquellos individuos con un interés específico en el desenlace del proyecto, tal como se evidencia en la Tabla 4.

Tabla 4

Lista de Stakeholders

N°	Descripción	
1	**Usuario Directo**	Sr. Jorge Neppas
2	**Usuario Indirecto**	Sr. Mauricio Neppas
3	**Tutor**	Msc. Luis Suárez
4	**Asesor**	Msc. Carlos Vásquez
5	**Desarrollador**	Srta. Katherine Guatemal

Fuente: Autor

3.3.2. Nomenclatura de Requerimientos

Establecer estos requisitos es crucial para garantizar el funcionamiento del sistema y cumplir con las sugerencias del usuario final. Por ello en la Tabla *5* se presentan los acrónimos y abreviaturas con el objetivo de hacer más accesible la interpretación de los términos utilizados, donde se engloban los requisitos esenciales del sistema, los requisitos de Stakeholders y los requisitos de Hardware y Software.

Tabla 5

Nomenclatura de Requerimientos

N°	Abreviatura	Descripción
1	**StSR**	Requerimientos de Stakeholders
2	**SySR**	Requerimientos del Sistema
3	**SRSH**	Requerimientos Arquitectura

Fuente: Adaptado de (IEEE/ISO/IEC, 2018)

Otro aspecto crucial para tener en cuenta es la priorización de los requisitos del sistema, es crucial para asignar los recursos de manera efectiva y abordar inicialmente las funciones críticas. Por ende, establecer un proceso transparente y bien definido de priorización resulta fundamental para entregar un producto que satisfaga las expectativas del cliente. La estrategia de priorización se encuentra detallada en la Tabla 6, donde se especifica el nivel de cada requisito.

Tabla 6

Priorización de los requerimientos del sistema

Prioridad	Descripción
Alta	Se considera fundamental para asegurar el rendimiento óptimo del sistema.
Media	Pueden ser dejados de lado en circunstancias extremas, pero su exclusión tendría un impacto parcial en el sistema.
Baja	Pueden ser descartados sin afectar significativamente en el funcionamiento del sistema.

Fuente: Adaptado de (Chacua, 2019)

3.3.3. Requerimientos de Stakeholders

El sistema a desarrollar debe ajustarse a los requerimientos de los interesados, que incluyen las necesidades de los usuarios. Estos requisitos se detallan en la Tabla 7 y se centran en los aspectos operativos y las demandas del usuario. Estos requisitos son fundamentales para evaluar el dispositivo y determinar si es adecuado para resolver el problema planteado.

Tabla 7

Requerimientos de Stakeholders

StSR				
Requerimientos de Stakeholders				
#	**Requerimiento**	**Prioridad**		
		Alta	**Media**	**Baja**
REQUERIMIENTOS DE FUNCIONAMIENTO				
StSR1	El sistema debe instalarse en un terreno expuesto a diversas condiciones climáticas.	X		
StSR2	El dispositivo debe recopilar datos sobre las variables ambientales para el entrenamiento de la de IA.	X		
StSR3	El sistema debe asegurar una conexión estable desde el nodo principal o Gateway hacia Internet	X		
StSR4	El sistema debe hacer uso de tecnologías inalámbricas.	X		
StSR5	Es necesario que el sistema sea energéticamente eficiente			X
StSR6	El sistema debe de enviar alertas en el caso de que los índices sean críticos.	X		
REQUERIMIENTOS DE USUARIO				
StSR7	El sistema debe mostrar de forma clara, precisa los datos al usuario.	X		
StSR8	El sistema debe operar de manera rentable desde una perspectiva financiera.		X	
StSR9	Los dispositivos de hardware deben estar disponibles en el mercado.		X	
StSR10	Es fundamental que los equipos utilizados en el terreno sean adaptables al entorno y se puedan transportar con facilidad.	X		

Fuente: Autor

3.3.4. Requerimientos del Sistema

Comprenden diversas áreas, tales como el uso, el rendimiento, las interfaces y los aspectos físicos, detallados en la Tabla 8. Estos proporcionan una guía sobre el comportamiento y las capacidades esenciales del sistema. Tiene como objetivo cubrir los requerimientos de los usuarios y lograr las metas establecidas en el proyecto, asegurando al mismo tiempo su adecuado funcionamiento.

Tabla 8

Requerimientos del Sistema

SySR				
Requerimientos del Sistema				
#	**Requerimiento**	**Prioridad**		
		Alta	**Media**	**Baja**
REQUERIMIENTOS DE USO				
SySR1	El sistema deberá ser fácil de usar e implementar en un terreno destinado al cultivo.	X		
SySR2	Los sensores deben realizar mediciones precisas de la temperatura y humedad ambiental, velocidad del viento y la intensidad de los rayos UV en el área de cultivo.	X		
SySR3	El nodo sensor y Gateway deben estar en comunicación constante.	X		
SySR4	El sistema debe permitir monitoreo remoto en tiempo real para verificar el estado y detectar posibles fallos		X	
REQUERIMIENTOS DE PERFOMANCE				
SySR5	El nodo sensor debe establecer comunicación inalámbrica con el Gateway utilizando tecnologías de LPWAN.	X		
SySR6	El nodo sensor debe enviar la información recopilada al nodo principal.	X		

SySR7	El Gateway debe recibir los datos y mostrar los valores registrados por el nodo sensor.	X	
SySR8	El sistema tiene la función de avisar al usuario final mediante una alerta, con un mensaje de presencia de helada.	X	
SySR9	El sistema debe proporcionar información ambiental en tiempo real.	X	
	REQUERIMIENTOS DE INTERFACES		
SySR10	El diseño del sistema embebido debe incluir pines que permitan recibir y enviar datos de manera eficiente.	X	
SySR11	Se debe garantizar la adecuada incorporación de cada sensor y módulos de transmisión.	X	
SySR12	La comunicación inalámbrica debe cumplir con las regulaciones y frecuencias asignadas por las autoridades competentes	X	
	REQUERIMIENTOS FÍSICOS		
SySR13	El nodo sensor debe tener dimensiones adecuadas que faciliten su manejo durante la instalación y extracción del área de cultivo.		X
SySR14	Carcasa protectora que se adapta a cualquier ambiente para asegurar su operatividad.	X	
SySR15	Los sensores y módulos deben ubicarse estratégicamente para mejorar la captura y transmisión de datos.	X	
SySR16	Ejecutar pruebas de operatividad y depuración de errores previo a la implementación	X	
SySR17	Terreno y área de cultivo listos para la instalación del sistema.	X	
	REQUERIMIENTOS DE MODOS DE ESTADO		
SySR18	El nodo gateway debe alertar si no recibe datos del nodo sensor durante un tiempo configurable, indicando posible falla o pérdida de alimentación.		X

SySR19	El nodo sensor reiniciarse automáticamente cuando se restaure la fuente alimentación	X

Fuente: Autor

3.3.5. Requerimientos de Arquitectura

Esta información facilitará la selección y adquisición de los elementos adecuados que se ajusten de manera coherente a las necesidades del proyecto. Los requerimientos de arquitectura se detallan en la Tabla 9, que describe los requisitos necesarios para el funcionamiento óptimo del sistema, abordando aspectos como el software, hardware y componentes eléctricos.

Tabla 9

Requerimientos de Arquitectura

SRSH				
Requerimientos del Arquitectura				
#	**Requerimiento**	**Prioridad**		
		Alta	**Media**	**Baja**
REQUERIMIENTOS LÓGICOS				
SRSH1	El nodo sensor debe contar con entradas lógicas y digitales para la conexión de los diferentes componentes	X		
SRSH2	Comunicación entre el nodo sensor y la puerta de enlace o Gateway	X		
SRSH3	Calibración de cada uno de los sensores	X		
SRSH4	Utilizar protocolo MQTT para envió de datos	X		
REQUERIMIENTOS DE DISEÑO				
SRSH5	Resistencia a condiciones climáticas adversas	X		
SRSH6	Los sensores y módulos de transmisión deben ser integrados en un único dispositivo	X		

SRSH7	Elegir la LPWAN con mejor desempeño según la ubicación geográfica.	X
SRSH8	Portabilidad del equipo	X
	REQUERIMIENTOS DE HARDWARE	
SRSH9	Cobertura inalámbrica estable de al menos 1000 metros	X
SRSH10	Dispositivos resistentes a condiciones ambientales externas.	X
SRSH11	Estación central para recolección y procesamiento de datos	X
SRSH12	Contar con antena que trabaje en los subGHz	X
	REQUERIMIENTOS DE SOFTWARE	
SRSH13	Librerías para la utilización de sensores y módulos de transmisión y recepción de datos.	X
SRSH14	Utilizar lenguajes de programación de uso libre y con flexibilidad de uso.	X
SRSH15	Agente de alerta para notificar al usuario	X
SRSH16	Es necesario que el software sea compatible con el procesamiento en tiempo real utilizando IA.	X
	REQUERIMIENTOS ELÉCTRICOS	
SRSH17	Contar con alimentación eléctrica y un respaldo de baterías para los nodos y Gateway	X

Fuente: Autor

3.3.6. Selección de Hardware

Una vez que se han establecido los requisitos del sistema, se inicia el proceso de selección del hardware, que abarca la elección de componentes como sensores, microprocesadores, microcontroladores y módulos de comunicación. Estos elementos deben ser compatibles y satisfacer los requisitos específicos del proyecto. La selección cuidadosa del hardware facilita la convergencia de todos los elementos del prototipo, asegurando su adecuado funcionamiento en conjunto.

3.3.6.1. Selección de Sensores

La selección de sensores implica identificar y elegir aquellos dispositivos que sean adecuados para medir las variables específicas requeridas por el sistema o proyecto. El nodo destinado al área de cultivo cuenta con tres tipos de sensores disponibles: para medir la humedad y temperatura ambiente, el índice UV y la velocidad del viento. Se realizará un análisis comparativo de los dispositivos disponibles en el mercado, teniendo en cuenta criterios como la disponibilidad, accesibilidad de costos, estabilidad y resistencia a condiciones externas.

- **Sensor de Humedad y Temperatura Ambiente**

Un sensor de temperatura y humedad ambiente es fundamental para detectar la posibilidad de heladas. En la Tabla 10 se presentan tres tipos de sensores disponibles en el mercado con el propósito de identificar cuál se ajusta adecuadamente a las necesidades y requerimientos del proyecto.

Tabla 10

Selección de Sensor de humedad y temperatura ambiental

Hardware	**Requerimientos**						**Valoración Total**
	StSR2	**StSR8**	**StSR9**	**SySR2**	**SySR11**	**SRSH3**	
LM35	1	1	1	0	0	1	4
DHT11	1	1	1	1	1	1	6
SHT11	1	1	1	0	0	1	4
DHT22	1	0	1	1	1	1	5
Elección	DHT11						

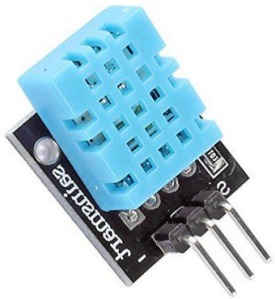
Fuente: Autor

Después de analizar los datos presentados en la Tabla 10, se determinó que el sensor DHT11 será la opción más adecuada para medir la temperatura y la humedad ambiente en este proyecto. Como se ilustra en la Figura 11, el sensor DHT11 proporciona lecturas digitales, lo

que garantiza una mayor precisión en comparación con el sensor LM35. Además, su asequibilidad en el mercado lo hace una elección favorable en comparación con el sensor DHT22 que, aunque ofrece características similares, es más costoso. Aunque el sensor SHT11 también ofrece alta precisión y fiabilidad, su mayor costo y menor disponibilidad lo hacen menos ideal para este proyecto. Adicionalmente, el sensor DHT11 destaca por su compatibilidad con una amplia gama de placas de procesamiento, lo que lo hace aún más versátil para su integración en el sistema.

Figura 11

Sensor DHT11



Fuente: (Kuongshun, 2023)

- **Sensor de radiación solar UV**

Este sensor debe ser capaz de medir la radiación ultravioleta (UV) con alta precisión, ya que esta variable juega un papel crucial en la evaluación de las condiciones climáticas y su impacto en el cultivo. La elección del sensor adecuado es fundamental para garantizar lecturas confiables que permitan un monitoreo eficiente y la toma de decisiones basadas en datos precisos. Para ello, se han explorado diversas opciones disponibles en el mercado, considerando factores como el rango de medición, la sensibilidad, la estabilidad a largo plazo y la compatibilidad con el sistema de monitoreo implementado. En la Tabla 11 se presentan algunos de los sensores especializados en la medición de la radiación solar, con el fin de facilitar la selección del dispositivo más adecuado para las necesidades del proyecto.

Tabla 11

Selección del sensor de radiación solar UV

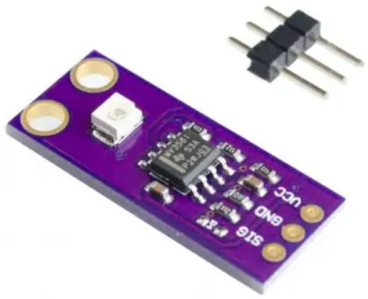
Hardware	Requerimientos						Valoración
	StSR2	StSR8	StSR9	SySR2	SySR11	SRSH3	Total
GUVA-T11GD	1	0	1	0	0	1	3
GUVA-T12SD	1	1	1	1	1	1	6
Veml6070	1	0	0	1	1	1	4
Elección	GUVA-T12SD						

Fuente: Autor

Después de revisar los datos presentados en la Tabla 11, se ha decidido seleccionar el sensor GUVA-T12SD para la medición de la radiación ultravioleta (UV). Este sensor, como se muestra en la Figura 12, ofrece lecturas precisas de la radiación UV, lo que es crucial para comprender y monitorear la exposición a esta forma de energía solar. Aunque existen otras opciones, como el sensor GUVA-T11GD y el VEML6070, se ha preferido el GUVA-T12SD por su capacidad para proporcionar mediciones precisas y su compatibilidad con una variedad de placas de procesamiento además de su consumo energético.

Figura 12

Sensor GUVA-S12SD



Fuente: (GNS Components, 2023)

- **Sensor de velocidad del viento o anemómetro**

La importancia del sensor de velocidad del viento, también llamado anemómetro, radica en su papel fundamental para detectar heladas en los cultivos. La velocidad del viento juega un papel crucial en la formación y severidad de las heladas, ya que puede evitar que el aire frío se asiente sobre los cultivos. Por lo tanto, es esencial realizar una comparación entre varios sensores para elegir el más adecuado para el proyecto, como se detalla en la Tabla 12.

Tabla 12

Selección del anemómetro

Hardware	**Requerimientos**							**Valoración**
	StSR2	**StSR8**	**StSR9**	**SySR2**	**SySR11**	**SRSH3**	**SRSH10**	**Total**
HANG FENG	1	0	1	1	0	1	1	5
VBESTLIFE	1	0	0	1	0	1	1	4
PCE-ADL 11	1	0	0	1	0	1	1	4
Terrific	1	1	1	1	1	1	1	7
Elección	Terrific							

Fuente: Autor

Después de evaluar diversas opciones de anemómetros, se decidió optar por el modelo de la marca Terrific, que se muestra en la Figura 13. Este sensor destaca por sus excepcionales prestaciones y su fácil integración con los demás componentes del sistema. En comparación con las marcas HANG FENG, VBESTLIFE y PCE-ADC 11, el anemómetro Terrific se distingue por su capacidad para proporcionar lecturas de la velocidad del viento con mayor precisión y consistencia. Además, Terrific ofrece una construcción robusta que garantiza una larga vida útil incluso en condiciones climáticas adversas, una amplia disponibilidad en el

mercado que facilita su adquisición y mantenimiento, y un excelente soporte técnico. Por otro lado, Hang Feng presenta una precisión moderada y una integración limitada, Vbestlife tiene una precisión aceptable pero menos consistente, y PCE-ADC 11, aunque con buena precisión y disponibilidad, no iguala la robustez y el soporte técnico de Terrific. Por estas razones, el anemómetro Terrific es la elección más adecuada para nuestras necesidades.

Figura 13

Anemómetro Terrific

Fuente: Autor

3.3.6.2. Selección de Microcontrolador

Para la elección del microcontrolador se tienen en cuenta tres alternativas: ESP32, ESP8266, Arduino Uno y Arduino Mega. Cada una de estas opciones ofrece diferentes capacidades y características que deben ser evaluadas en función de las necesidades específicas del proyecto. Se debe tomar en cuenta que se necesita 2 placas microcontroladoras tanto para el nodo sensor y el Gateway.

- **Microprocesador para el nodo sensor**

La Tabla 13 proporciona una comparación detallada de estas alternativas, lo que facilita la selección del microcontrolador más adecuado para el sistema en cuestión.

Tabla 13

Selección de Microcontrolador para el nodo sensor

Hardware	Requerimientos							Valoración
	StSR8	StSR9	SySR10	SySR11	SySR13	SRSH1	SRSH4	Total
ESP32	1	1	1	1	1	1	1	7
ESP8266	0	1	1	1	0	1	1	5
Arduino Nano	1	1	0	1	0	1	1	5
Arduino Uno	1	1	1	1	0	1	0	5
Arduino Mega	1	1	1	0	0	1	0	4
Elección	ESP32							

Fuente: Autor

Tras una evaluación exhaustiva de los microcontroladores, se decidió optar por la placa ESP32 para el nodo sensor, como se muestra en la Figura 14. Esta decisión se fundamenta en varios factores clave. El ESP32 ofrece un rendimiento de procesamiento superior y una mayor cantidad de pines en comparación con el Arduino Mega, facilitando la integración con dispositivos sensores y módulos LPWAN para una transmisión de datos simplificada. A diferencia del ESP8266, que tiene un rendimiento inferior y menos pines disponibles, el ESP32 proporciona una mayor consistencia y fiabilidad. Comparado con el Arduino Nano y el Arduino Uno, el ESP32 no solo ofrece un rendimiento mejorado, sino también una conectividad más robusta y versátil. Su amplia disponibilidad en el mercado y el sólido soporte técnico también contribuyen a su atractivo, posicionándolo como una opción eficiente y económica frente a los otros microcontroladores evaluados.

Figura 14

Microcontrolador ESP32

Fuente: (Xukyo, 2024)

- **Microprocesador para el Gateway**

La Tabla 14 ofrece una comparación detallada de estas alternativas, lo que simplifica la elección del microcontrolador más adecuado para el sistema en cuestión.

Tabla 14

Selección de microcontrolador para el Gateway

Hardware	Requerimientos						Valoración Total
	StSR8	StSR9	SySR10	SySR12	SySR13	SRSH1	
ESP32	0	1	1	0	0	1	3
Arduino Uno	1	1	1	1	1	1	6
Arduino Mega	0	1	1	1	0	1	4
ESP8266	0	1	1	1	0	1	4
Elección	Arduino Uno						

Fuente: Autor

Después comparación de las opciones de microcontroladores, se seleccionó la placa Arduino Uno para el Gateway que se muestre en la Figura 15, debido a su sólido desempeño, versatilidad y compatibilidad con módulos de LPWAN y microprocesadores. Además, su precio es muy factible a comparación de la ESP32, ESP8266 y Arduino Mega, también se toma en cuenta las necesidades

Figura 15

Microcontrolador Arduino Uno

Fuente: (Arduino.cl, 2023)

3.3.6.3. Selección de Microprocesador

La elección del microprocesador es importante porque ayuda a determinar la capacidad de procesamiento del Gateway, lo que afecta su manejo de grandes volúmenes de datos y la eficiencia en las operaciones de comunicación. Además, debe ser compatible con los protocolos de comunicación del sistema, garantizando una conexión estable con los nodos sensores. Por ello se realiza una comparación entre varias opciones como se ve en la Tabla 15.

Tabla 15

Selección del Microprocesador

Hardware	Requerimientos						Valoración Total
	StSR8	StSR9	SySR10	SySR12	SySR13	SRSH1	
Intel	0	1	1	0	0	1	3
Galileo	0	0	1	1	1	1	6
Rasperry Pi 3	1	1	1	1	1	1	5
Elección	Rasperry Pi 3 Model B						

Fuente: Autor

La Raspberry Pi 3 Model B, representada en la Figura 16, se destaca como una de las mejores opciones debido a sus capacidades avanzadas y facilidad de uso. Este dispositivo incorpora un procesador de alto rendimiento, buena capacidad de memoria RAM y una GPU optimizada, lo que la convierte en una opción ideal para aplicaciones que requieren procesamiento intensivo de datos. Además, su compatibilidad con los microcontroladores seleccionados la hace una plataforma versátil para múltiples implementaciones.

Figura 16.

Raspberry Pi 3 Model B

Fuente: (Raspberry Pi, 2022)

3.3.6.4. Selección de Tecnología Inalámbrica de Baja Potencia (LPWAN)

El proyecto requiere establecer una conexión entre el nodo sensor y el Gateway, por lo que es esencial elegir una tecnología inalámbrica que se adapte al entorno y sea compatible con los demás componentes seleccionados.

Las tecnologías LPWAN son ampliamente empleadas en soluciones IoT dirigidas a la agricultura de precisión debido a sus características como ancho de banda, alcance, velocidad de transferencia de datos y eficiencia energética, entre otras. Por esta razón, se analizan diversas tecnologías de LPWAN en la Tabla 16, teniendo en cuenta los requisitos establecidos para seleccionar la más idónea para este proyecto.

Tabla 16

Selección de LPWAN

Hardware	Requerimientos						Valoración Total
	StSR4	SySR5	SySR13	SRSH2	SRSH7	SRSH9	
Sigfox	1	1	1	1	0	1	5
LoRa	1	1	1	1	1	1	6
ZigBee	1	1	1	1	0	1	5
Elección	LoRa						

Fuente: Autor

La tecnología LoRa destaca sobre otras alternativas como Sigfox y ZigBee debido a su adaptabilidad al entorno y sus características excepcionales que contribuyen al logro de los objetivos del proyecto. Con un alcance de hasta 20 km en áreas abiertas y el uso de una frecuencia de 915 MHz en la región de América, LoRa facilita el establecimiento de una conexión confiable entre el nodo sensor y el Gateway en entornos rurales, cumpliendo así con las necesidades específicas de implementación.

- **Selección del módulo LoRa**

La diversidad de módulos LoRa ofrece una amplia gama de opciones, cada una con características específicas y adaptadas a distintas aplicaciones. Por este motivo, se lleva a cabo una comparación entre varias de estas alternativas disponibles, detalladas en la Tabla 17. Este análisis se enfoca en seleccionar la opción óptima que se ajuste no solo a los parámetros regionales, sino también a los requisitos particulares del sistema, garantizando así una implementación efectiva y eficiente.

Tabla 17

Selección del Módulo LoRa

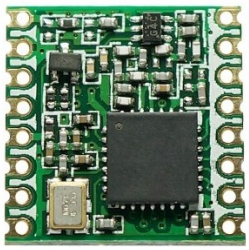
Hardware	Requerimientos						Valoración Total
	StSR9	SySR3	SySR11	SySR12	SRSH9	SRSH12	
RFM95W	1	1	1	1	1	1	6
RYLR896	0	1	1	0	0	1	3
SX1278	1	1	0	0	0	1	3
Elección	RFM95W						

Fuente: Autor

Para este proyecto, se optó por el módulo RFM95W, el cual se muestra en la Figura 17. Esta elección se fundamenta en su capacidad para operar en la banda de frecuencia de 915 MHz, asignada para Latinoamérica. Además de su adecuación a los requisitos regionales, este módulo destaca por su disponibilidad en el mercado y su compatibilidad con los demás componentes, lo que simplifica su integración. Asimismo, su configuración sencilla y bajo consumo energético lo hacen aún más atractivo y viable para el uso prolongado.

Figura 17

Módulo RFM95W



Fuente: (Sigma, 2023)

3.3.6.5. Selección fuente de alimentación

Algunos requisitos del sistema especifican que el nodo sensor debe permanecer en constante movimiento para cumplir con el objetivo principal del proyecto que en resumen es

generar alertas ante la presencia de heladas en los cultivos vulnerables. Dado que estos cultivos rotan por todo el terreno, el nodo sensor debe ser móvil y adaptable. Por lo tanto, el suministro eléctrico se convierte en un desafío en este entorno dinámico. Para abordar este problema, se opta por el uso de baterías recargables que garanticen el funcionamiento continuo del sistema.

La selección de la batería se realiza teniendo en cuenta los dispositivos que se alimentarán y sus requisitos energéticos. En la Tabla 18 se detallará el consumo de cada uno de estos dispositivos, lo que permitirá realizar un cálculo preciso y elegir una batería adecuada para satisfacer las necesidades del sistema.

Tabla 18

Consumo Eléctrico Nodo Sensor

Elemento	Consumo en Funcionamiento	Consumo en modo Sleep
DHT11	1 mA	0.01 mA
UV Guva-s12sd	1.5 mA	0.01 mA
Anemómetro	50 mA	---------
ESP32	50 mA	0.05 mA
RFM95W	40 mA	14.2 mA
Total	**142.5 mA**	**14.27 mA**

Fuente: Autor

De acuerdo con el amperaje para alimentar del nodo sensor es de 142.5 mA, que corresponde a los componentes integrados en dicho nodo. Para el cálculo del consumo de batería se utiliza la ecuación (2).

$$Consumo\ Total = \frac{(T_{cn} * I_{cn}) + (T_{cd} * I_{cd})}{T_{cn} + T_{cd}} \quad (2)$$

Donde:

T_{cn} = Tiempo de consumo en funcionamiento

I_{cn} = Intensidad de consumo de corriente en funcionamiento

T_{cd} = Tiempo de consumo en modo sleep

I_{cd} = Intensidad de consumo de corriente en modo sleep

Los parámetros seleccionados para este nodo se basan en las necesidades específicas del proyecto y se detallan a continuación:

$T_{cn} = 2\,\text{min} = 160\ s$

$I_{cn} = 142{,}5\ mA$

$T_{cd} = 5\ min = 300\ s$

$I_{cd} = 14.27\ mA$

Seguidamente se procede a sustituirlos en la ecuación (2), para calcular el consumo de energía que el nodo necesita en el entorno donde se aplicara.

$$Consumo\ Total = \frac{(160 * 142{,}5) + (300 * 14{,}27)}{160 + 300}$$

$$Consumo\ Total = \mathbf{50,90\ \ mA}$$

Basándonos en el cálculo de consumo energético obtenido de 50.90 mA, se selecciona una batería recargable de 3,7 voltios y 5000 mAh. Utilizando estos datos, calculamos la vida útil de la batería con la fórmula presentada en la ecuación (3):

$$Tiempo\ de\ Vida\ de\ la\ Batería = \frac{Capacidad\ de\ la\ batería}{Consumo\ Total} \quad (3)$$

Donde:

Capacidad de la batería: 5000 mAh

Consumo Total: 50,9 mA

Realizando la sustitución correspondiente en la ecuación (3) se obtiene lo siguiente:

$$Tiempo\ de\ Vida\ de\ la\ Batería = \frac{5000\ mAh}{50{,}9\ mA}$$

$$Tiempo\ de\ Vida\ de\ la\ Batería = \mathbf{98.23\ h}$$

3.3.7. Selección del Software

La elección del software para este proyecto es una decisión crítica que impacta directamente en su éxito y sostenibilidad a largo plazo. Esta selección se fundamenta en criterios esenciales como la compatibilidad con el hardware, la facilidad de uso, la disponibilidad de recursos y el soporte de la comunidad. Una adecuada elección no solo garantiza una integración fluida con los microcontroladores y otros componentes del sistema, sino que también facilita el desarrollo y mantenimiento del código, optimizando el tiempo y los recursos disponibles.

3.3.7.1. Selección de Software para el nodo sensor

El microcontrolador del nodo sensor necesita ser configurado o programado para cumplir con su función de manera eficaz. Para ello, se requiere un software que permita codificar y gestionar los firmwares de estos dispositivos. Después de una evaluación inicial, se toman las opciones de Python y Arduino IDE, ya que ambos cuentan con librerías compatibles con el ESP32 y ofrecen un amplio soporte de la comunidad. Las características y funcionalidades específicas se comparan en la Tabla 19 para tomar la decisión idónea.

Tabla 19

Software para microcontroladores del nodo sensor

Hardware	**Requerimientos**			**Valoración Total**
	SySR11	**SRSH13**	**SRSH14**	
Python	1	0	1	4
Arduino IDE	1	1	1	6
Elección	Arduino IDE			

Fuente: Autor

Se opta por el software de Arduino IDE sobre Python para programar el microcontrolador por que la plataforma de Arduino IDE cuenta con bibliotecas preconfiguradas para el soporte de la ESP32, facilitando su implementación y compatibilidad con los demás componentes como módulos LoRa y sensores. Además, este software es eficiente en la utilización de recursos y tiempo de ejecución, que son fundamentales para asegurar el rendimiento optimo del proyecto.

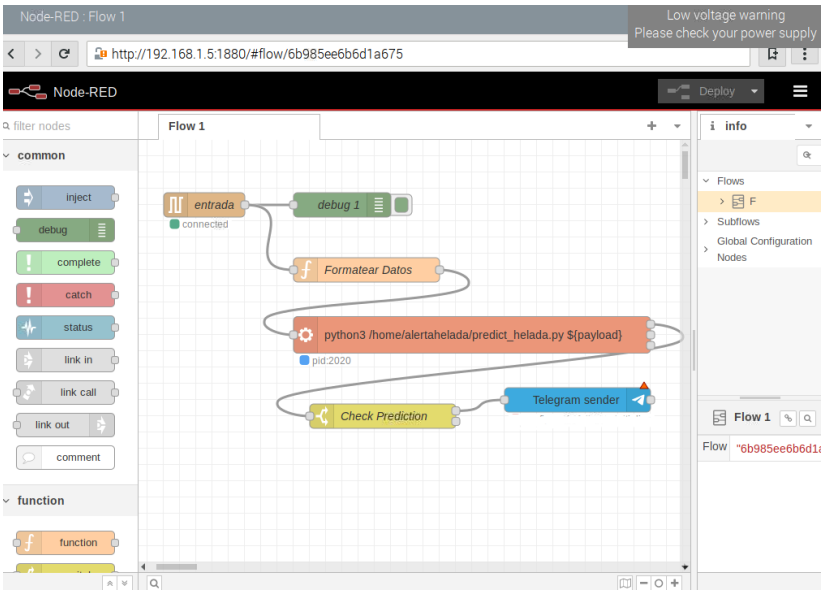
3.3.7.2. Selección de Software para el Gateway

En el nodo Gateway contamos con varios dispositivos programables por lo cual se utilizará diferentes Software que sean compatible con los mismos. Además, se requiere de varios softwares que también son importantes para lograr alcanzar con el objetivo planteado. Para la parte de Arduino Uno se utilizará Arduino IDE, se elige directamente este porque es creado específicamente para programar toda la gama de dispositivos Arduino, por lo cual no se realiza una comparación de software.

En el nodo Gateway se utilizará el sistema operativo Raspbian, previamente instalado en la Raspberry Pi 3 Model. Complementariamente, dentro de este sistema operativo se emplea el software Node-RED, que permite la captura de valores mediante comunicación serial. Además, su entorno gráfico como se observa en la Figura 18, está compuesto por múltiples nodos, facilita la integración de la inteligencia artificial entrenada, siendo así que este software se adapta y tiene compatibilidad con el objetivo planteado.

Figura 18

Entorno Grafico Node-Red



Fuente: Autor

3.3.7.3. Software para el entrenamiento del Árbol de decisión.

En este caso se optó por elegir al software de Visual Studio Code debido a su compatibilidad con el lenguaje de programación de Python, además de contar con las librerías necesarias el entrenamiento de la IA, El entorno de desarrollo de este software cuenta con una gran variedad de extensiones, que facilitan la depuración, corrección de errores y la gestión de entornos virtuales, optimizando así el flujo de trabajo, Visual Studio Code a comparación de otros software intérpretes de Python, cuenta con el soporte para múltiples lenguajes lo que lo convierte en una opción robusta y versátil para el desarrollo de nuestro proyecto.

3.3.7.4. Selección del software de la base de datos

El almacenar de los datos obtenidos para posteriormente visualizarlos es de gran importancia. Por ello, se toma en consideración a dos softwares que pueden ser aptos para este proceso: InfluxDB y la otra DynamoDB. Estas son opciones validas ya que permiten el almacenamiento sin limitaciones, ofreciendo así gran rendimiento con grandes volúmenes de datos.

Sin embargo, se elige implementar InfluxDB porque, al ser un software libre, no genera costos adicionales, ayudando con el objetivo de sustentabilidad del proyecto en general. Además, este se complementa usando el protocolo MQTT mejorando la comunicación en tiempo real y optimizando el almacenamiento de datos.

3.3.7.5. Selección software de Visualización

La visualización de los datos es secundaria porque no es el objetivo principal del proyecto, sin embargo, los agricultores podrían acceder a ellos en tiempo real. Por ello, se elige Grafana para la visualización ya que es compatibilidad con InfluxDB permitiendo acceder a los datos almacenados. Además, el entorno gráfico es amigable para el usuario permitiendo la personalización para mejorar la interpretación de los datos.

3.3.8. Enlace inalámbrico con tecnología LoRaWAN.

Para establecer un enlace inalámbrico con tecnología LoRaWAN, es esencial garantizar una comunicación estable entre el nodo Sensor y el nodo Gateway, permitiendo la transmisión de paquetes a largas distancias de manera estable y confiable. Un aspecto clave en el diseño del enlace es el análisis de las pérdidas de propagación, ya que estas influyen directamente en la calidad de la comunicación.

En entornos rurales, es fundamental conocer y estimar las pérdidas que el enlace experimentará a lo largo de su trayecto. Estas pérdidas pueden ser predecibles y calculables antes de la implementación mediante modelos de propagación. En este caso, se empleará el modelo de Okumura-Hata, el cual permite calcular la atenuación de la señal en diferentes tipos de zonas, incluidas las rurales. Este modelo considera factores clave, permitiendo optimizar la ubicación y configuración de los dispositivos para minimizar las pérdidas y garantizar un rendimiento óptimo de la red LoRaWAN.

3.3.8.1. Cálculo de perdidas en el envío de paquetes con el Modelo de Propagación Okumura-Hata

El Modelo de Propagación Okumura-Hata permite calcular la pérdida de un enlace inalámbrico en diferentes entornos, en caso de las zonas rurales se toma en consideración la atenuación adicional causada por la menor densidad de edificación y obstáculos en comparación con entornos urbanos o suburbanos. La ecuación (4) representa la formula general del modelo Okumura-Hata, que considera factores específicos del entorno, como la frecuencia de operación, la altura de las antenas y la distancia del enlace. Sin embargo, esta puede ser modificada con la finalidad de adaptarse al entorno.

$$L_u = 69.55 + 26.16\,log(fc) - 13.82\,log(hb) - a(H_m) + [44.9 - 6.55log10(hb)]\log(d) \quad (4)$$

Nota: Obtenido de (ITU-R, 2015)

Donde:

L_u =Pérdidas de trayectoria en áreas urbanas. (dB)

fc = Frecuencia de transmisión. (MHz)

hb = Altura de la antena del Gateway. (m)

$a(H_m)$ = factor de corrección de la altura de la antena.

d = Distancia entre el Gateway y nodo sensor. (km).

Una vez definida la ecuación (4) del modelo Okumura-Hata, se reformula con el propósito de derivar la ecuación (5) , la cual facilita el cálculo del factor de corrección de la altura efectiva de la antena.

$$a(H_m) = (1.1\,log\,fc - 0.7)hm - (1.56\log fc - 0.8) \quad (5)$$

Nota: Obtenido de (ITU-R, 2015)

Donde:

$a(H_m) =$ factor de corrección de la altura de la antena.

fc = Frecuencia de transmisión. (MHz)

hm = Altura de la antena del nodo sensor. (m)

Para calcular la perdida de propagación en zonas abiertas o rurales se considera la ecuación (6) la cual se deriva de la ecuación (4). Esta permite estimar con mayor precisión las pérdidas de señal en el sector Pesillo-Olmedo, donde se implementará el proyecto. Esta adaptación es fundamental para optimizar el diseño del enlace inalámbrico, asegurando una cobertura adecuada y una comunicación estable entre los nodos de la red

$$L_o = L_u - 4.78(\log fc)^2 + 18.33 \log fc - 40.94 \quad (6)$$

Nota: Obtenido de (ITU-R, 2015)

Donde:

L_0 =Pérdidas de trayectoria en áreas rurales. (dB)

L_u =Pérdidas de trayectoria en áreas urbanas. (dB)

fc = Frecuencia de transmisión. (MHz)

Una vez establecidas las ecuaciones, se sustituyen los valores correspondientes al enlace inalámbrico del proyecto, considerando los parámetros operativos del módulo LoRa, el cual opera a una frecuencia portadora de 915 MHz y una altura de antena de 3 metros. La correcta aplicación de estos datos en el modelo permite calcular con precisión la pérdida de propagación, garantizando una evaluación realista del rendimiento del enlace. La determinación de la altura de las antenas se realizó considerando las características del cultivo. Los resultados obtenidos se presentan en la Tabla 20.

Tabla 20

Cálculo de pérdidas de propagación con Okumura-Hata

Datos del enlace inalámbrico	**Cálculo de perdida de Propagación siguiendo el modelo Okumura-Hata**
Factor de corrección de la altura efectiva de la antena $\boldsymbol{a(H_m)}$ *fc:915MHz, hm:3 m*	$a(H_m) = (1.1 \log 915MHz - 0.7) * 3m - (1.56 \log 915MHz - 0.8)$ $\boldsymbol{a(H_m) = 3.85m}$
Enlace Inalámbrico *fc:915MHz* *hm:3 m* *hb:3m* *a(hm): 3.85 m* *d: 0.55 Km*	$L_u = 69.55 + 26.16\, log(915MHz) - 13.82\, log(3m) - 3.85m + [44.9 - 6.55log10(3m)] \log(0.55km)$ $\boldsymbol{L_u = 125.73dB}$ $L_o = 125.73dB - 4.78\, (\log 915MHz)^2 + 18.33\, \log 915MHz - 40.94$ $\boldsymbol{L_o = 97.15\, dB}$

Fuente: Autor

Después de calcular la pérdida de propagación del enlace, se procede a realizar la simulación utilizando el software Radio Mobile, permitiendo comparar los resultados obtenidos con los valores presentados en la Tabla 20 para validar la precisión del análisis.

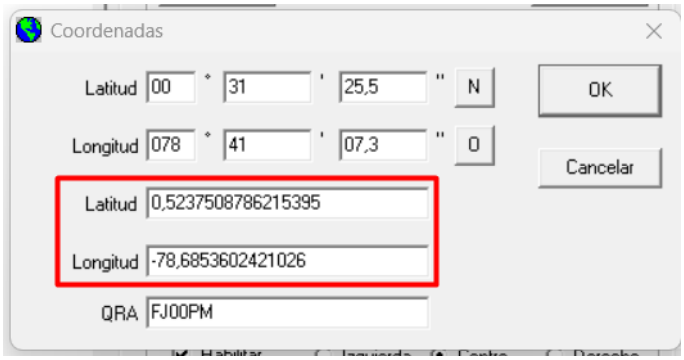
3.3.8.2. Simulación del Enlace Inalámbrico con tecnología LoRaWAN

La simulación del enlace inalámbrico se lleva a cabo en Radio Mobile, un software que permite configurar diversos parámetros, incluyendo coordenadas geográficas reales. Al ingresar estos datos como se observa en la Figura 19, la simulación se desarrolla en un entorno semirealista, proporcionando una representación más precisa del escenario de implementación.

Además, Radio Mobile tiene la capacidad de descargar mapas del área específica, lo que facilita la evaluación del enlace en función de las condiciones del terreno.

Figura 19

Ingreso de coordenadas en Radio Mobile

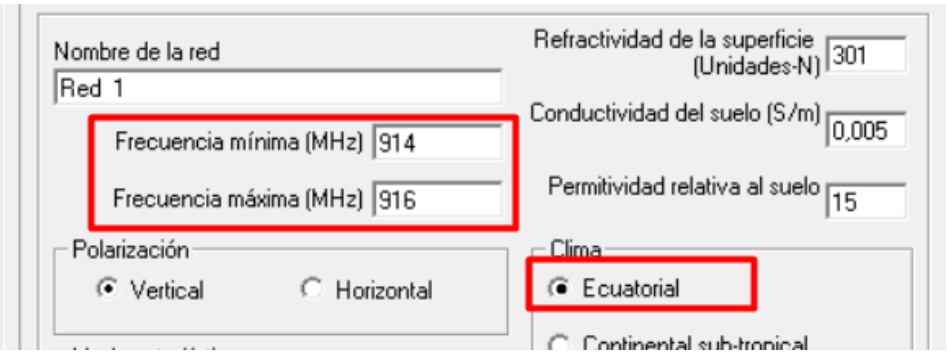


Fuente: Autor

Una vez definida la zona geográfica, se procede a configurar los parámetros del módulo LoRa, estableciendo la frecuencia de operación en 915 MHz. Radio Mobile permite ingresar el rango de frecuencia mínimo y máximo, lo que facilita la selección de la frecuencia exacta requerida, como se muestra en la Figura 20. Además, se especifica la ubicación geográfica, en este caso, Ecuador, para asegurar que la simulación refleje con mayor precisión las condiciones reales del entorno.

Figura 20

Ingreso de Frecuencia mínima y máxima en Radio Mobile

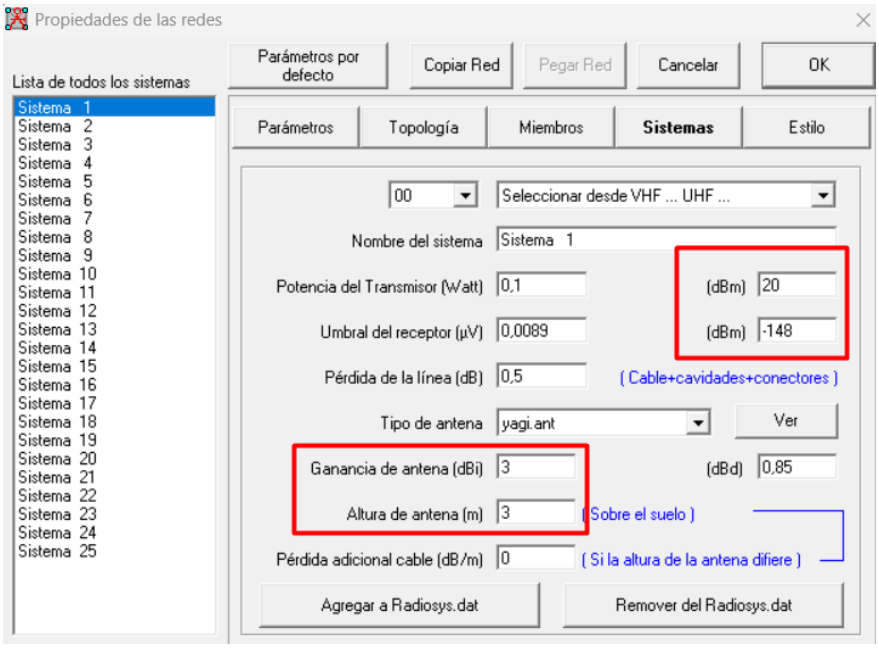


Fuente: Autor

Además, se configura la potencia de transmisión en 20 dBm y la sensibilidad en -148 dBm, valores especificados por el fabricante del módulo RFM95W. Asimismo, considerando la antena utilizada, se obtiene una ganancia aproximada de 3 dBi, y se establece una altura de instalación de 3 m para optimizar la propagación de la señal. Todas estas configuraciones están detalladas en la Figura 21.

Figura 21

Parámetros del módulo RFM95W en radio Mobile

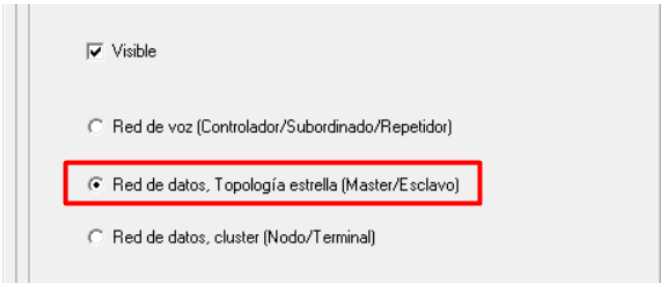


Fuente: Autor

Otro aspecto clave en la configuración es la selección del tipo de red a simular, en este caso, se opta por una red de datos maestro/esclavo como se observa en la Figura 22. Esta configuración permite representar con precisión la comunicación entre el Gateway y el nodo sensor, asegurando una simulación realista del enlace inalámbrico.

Figura 22

Elección del tipo de enlace en Radio Mobile

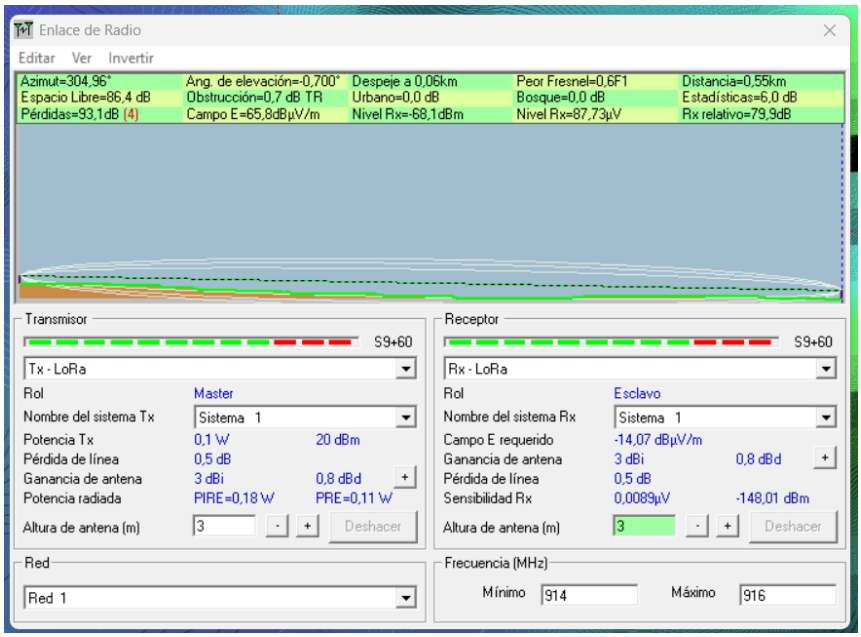


Fuente: Autor

A continuación, se configuran las coordenadas geográficas del Gateway y nodo Sensor, se procede a visualizar los resultados obtenidos, los cuales se detallan en la Figura 23. Se puede observar que la pérdida de propagación del enlace inalámbrico obtenido en la simulación es similar al que se calculó en la Tabla 20.con el modelo de Okumura-Hata siendo este 97.15 dB con una diferencia de -4 dB, así se puede confirmar que el enlace puede llevarse a cabo el sector y este es estable para la transmisión de los datos.

Figura 23

Enlace inalámbrico simulado en Radio Mobile



Fuente: Autor

3.3.8.3. Análisis espectral de la transmisión LoRa

Se considera el análisis espectral de la transmisión, ya que permite observar el comportamiento de la señal. Para ello, se utiliza un analizador de espectro, el cual opera dentro de un rango de frecuencias que incluye la frecuencia central utilizada por el módulo LoRa RFM95W durante la transmisión. El analizador de espectro tiene un rango de frecuencia de 9kHz y 7.0 Ghz.

Los parámetros configurados en el analizador de espectro se detallan en la Tabla 21, donde se especifican tanto la frecuencia inicial como la final, permitiendo un análisis preciso de la señal transmitida y su comportamiento dentro del espectro asignado.

Tabla 21

Parámetros de frecuencia inicial y final

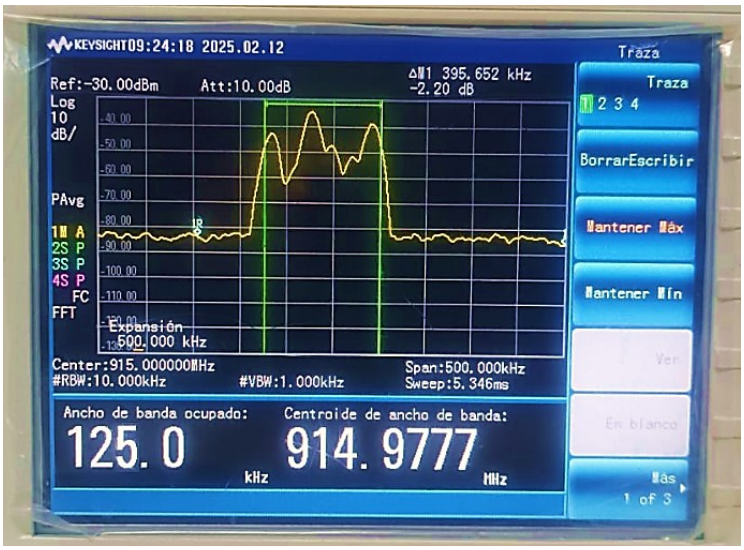
Parámetros	Valor
Frecuencia Inicial	914.90 MHz
Frecuencia Final	915.01 Mhz

Fuente: Autor

Según la Figura 24 el módulo LoRa opera correctamente dentro del rango de frecuencia previsto, aproximadamente 915 MHz, y utiliza un BW de 125 kHz. Esta configuración resulta adecuada para establecer comunicaciones de larga distancia con velocidades de datos reducidas. La potencia medida, de -16.86 dBm, indica que el transmisor está funcionando a niveles intermedios, lo que lo hace apropiado para pruebas en entornos cerrados o distancias cortas.

Figura 24

Espectro de transmisión LoRa



Fuente: Autor

3.4. Etapa de Diseño

Esta etapa empieza describiendo el diseño y armado del prototipo, se presentan los diagramas correspondientes tanto para el nodo sensor y nodo Gateway. Y se describe también cada configuración y códigos utilizados para el desarrollo optimo del prototipo. Esta etapa comprende la parte más importante de todo este proyecto, ya que integra y consolida todos los datos, requerimientos y sugerencias recopilados en las etapas previas, asegurando un desarrollo coherente y funcional.

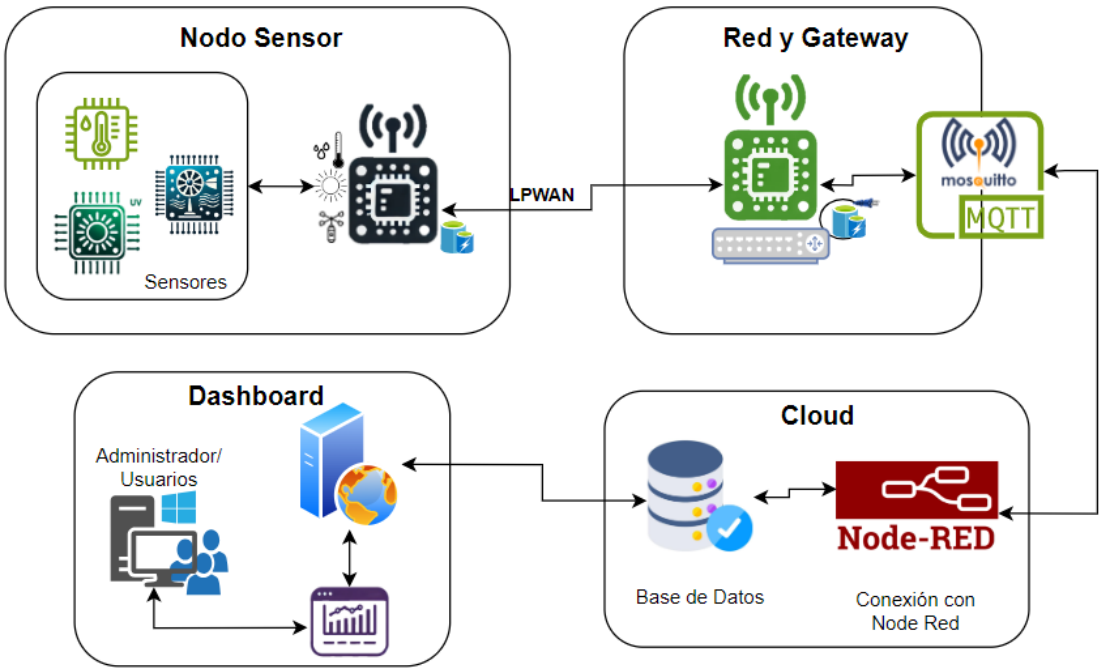
3.4.1. Diseño y Descripción del Sistema de Alerta Contra Heladas

La finalidad de este proyecto es alertar a los agricultores sobre la presencia del fenómeno climático conocido como helada. Al estar informados de su ocurrencia, los agricultores podrán tomar medidas preventivas y de protección para sus cultivos. La

implementación de este sistema permitirá supervisar los cambios climáticos mediante la monitorización de factores clave como la temperatura y humedad ambiental, la radiación UV y la velocidad del viento, todos ellos determinantes en la aparición de heladas. Este sistema no solo asegura la producción eficiente de los cultivos, sino que también contribuye a la sustentación económica de la región y del país, garantizando la producción de alimentos.

Figura 25

Topología del Sistema de Alerta Contra Heladas



Fuente: Autor

En la Figura 25 se ilustra la topología del sistema de alerta contra heladas para cultivos vulnerables. Este sistema adopta una arquitectura IoT que incluye varios componentes clave. Primero, el nodo sensor se encarga de recolectar datos de temperatura, humedad ambiental, radiación UV y velocidad del viento, mediante sensores específicos para cada variable. Estos datos son transmitidos al Gateway utilizando la tecnología LPWAN, conocida por su bajo consumo de energía y amplio alcance, ideal para aplicaciones agrícolas remotas.

El Gateway actúa como un intermediario crucial, gestionando los datos receptados de los nodos sensores. Su función es procesar y enviarlos a la nube para su almacenamiento seguro. En la nube, los datos se almacenan en una base de datos que permite su acceso y gestión eficiente.

En la capa de aplicación, los datos almacenados se visualizan a través de una interfaz gráfica amigable. Esta interfaz permite a los usuarios monitorear en tiempo real las condiciones climáticas que pueden provocar heladas. Mediante las alertas, los agricultores pueden recibir notificaciones tempranas sobre cambios críticos en el clima.

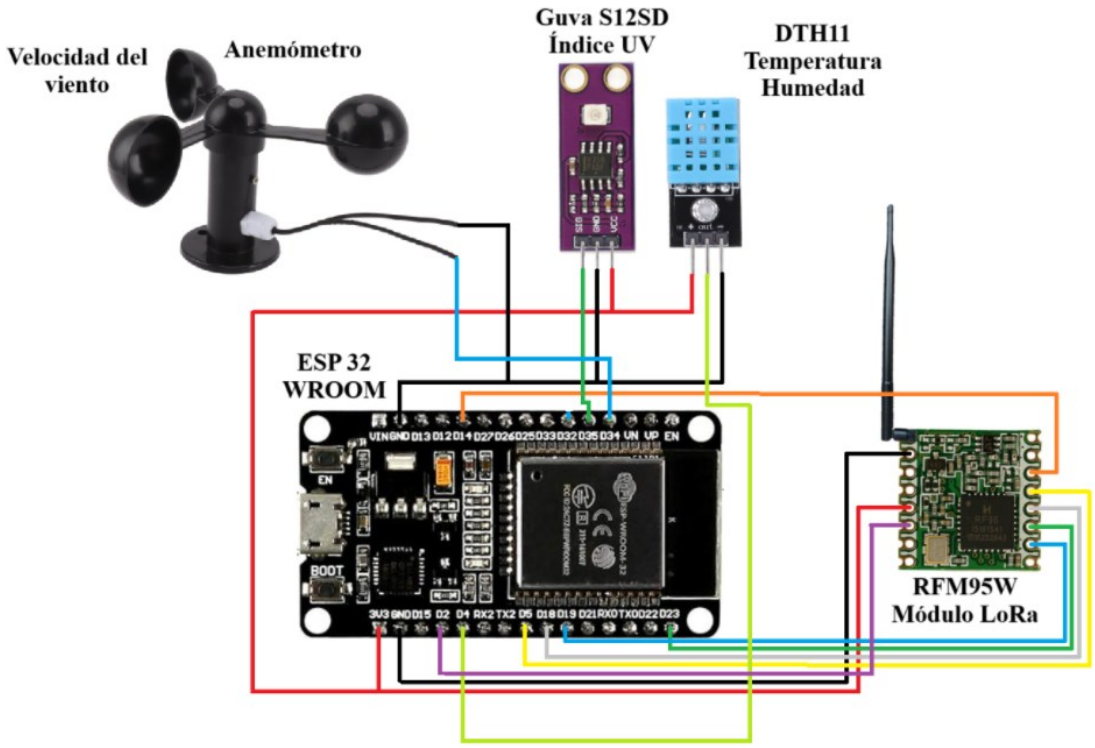
Posteriormente se obtendrá un sistema diseñado para recopilar datos ambientales relevantes a través de sensores, procesar y analizar estos datos utilizando técnicas de inteligencia artificial, y proporcionar alertas precisas y oportunas a los agricultores a través de mensajes de texto programados. El objetivo final es mejorar la protección de los cultivos vulnerables en la Parroquia Rural Olmedo – Pesillo, perteneciente al Cantón Cayambe, mediante la detección de heladas, lo que permitirá a los agricultores tomar medidas preventivas y mitigar los posibles daños en sus cultivos.

3.4.2. Diagrama de conexión del Nodo Sensor

El diagrama de conexión del Nodo Sensor muestra la interconexión entre un ESP32, varios sensores y un módulo LoRa. El ESP32 actúa como el cerebro del sistema, recolectando datos de los sensores y transmitiéndolos a través del módulo LoRa. Los sensores conectados incluyen un anemómetro para medir la velocidad del viento, un sensor de temperatura y humedad (DHT11), y un sensor de radiación ultravioleta (GUVA-S12SD). Los cables de conexión están claramente codificados por colores para facilitar la identificación.

Figura 26

Diagrama de Conexión del Nodo Sensor



Fuente: Autor

Como se observa en la Figura 26, el anemómetro se conecta a través de un cable azul al PIN D34 y el cable negro a GND, el sensor DHT11 a través de cables verde al PIN D4 y cable negro y rojo a GND y 3.3V respectivamente y el sensor GUVA-S12SD a través de cables verde al PIN D35 y de igual manera que el sensor DHT11 el cable negro y rojo a GND y 3.3V. El módulo LoRa está conectado al ESP32 mediante cables diferentes cables que se detallan en la Tabla 22, los cuales son fundamentales ya que permiten la comunicación inalámbrica con otros nodos o Gateway.

Tabla 22

Conexión de pines RFM95W a ESP32 WROOM

Modulo LoRa RFM95W	ESP32 WROOM	Color
VCC	3.3V	Rojo
GND	GND	Negro
DIO0	2	Violeta
RST	14	Naranja
SCK	18	Plomo
NSS (CS)	5	Amarillo
MISO	19	Azul
MOSI	23	Verde

Fuente: Autor

Este montaje es crucial para el proyecto de alerta contra heladas, ya que permite monitorear y transmitir datos ambientales en tiempo real, asegurando una respuesta rápida ante condiciones climáticas adversas.

3.4.3. Diagrama de conexión del Gateway

El diagrama de conexión del Gateway muestra cómo se interconectan un Arduino Uno, una Raspberry Pi y un módulo LoRa para recibir y procesar los datos transmitidos por el nodo sensor. El módulo LoRa está conectado al Arduino Uno, que actúa como intermediario, recibiendo datos del módulo LoRa y enviándolos a la Raspberry Pi 3 Model para su procesamiento y visualización.

Las conexiones representadas en la Figura 27 son detalladas a continuación:

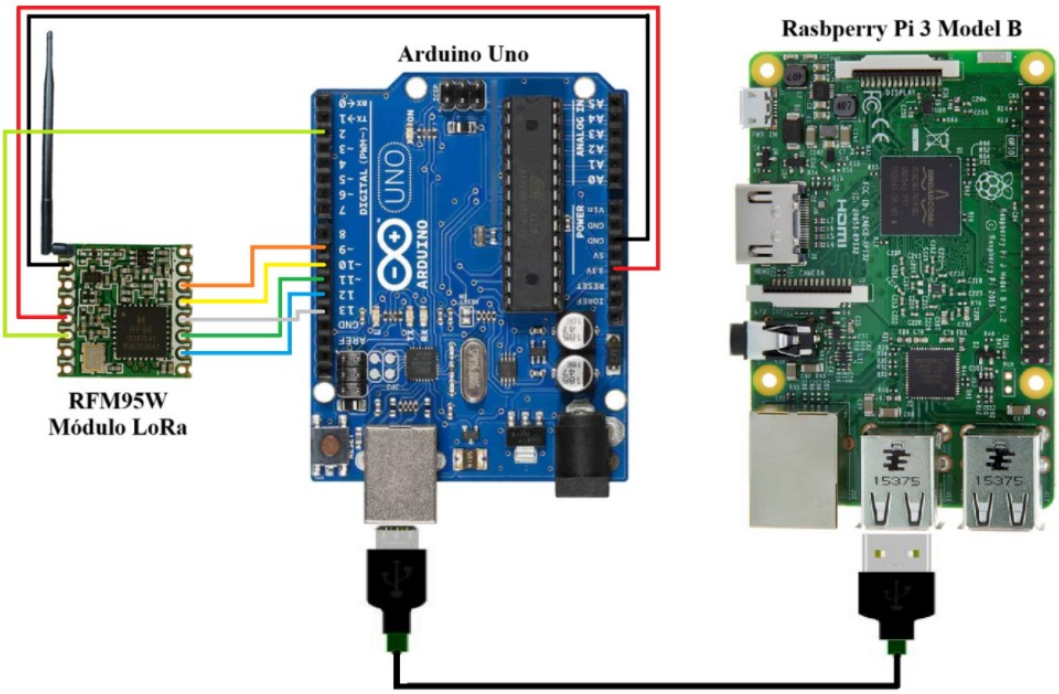
- El módulo LoRa se conecta al Arduino Uno a través de varios pines: el cable naranja de RST va al PIN D10, el cable azul de MISO va al PIN D11, el cable amarillo de NSS

al PIN D12, el cable verde de Mosi al PIN D13, el cable verde claro de Dio0 al PIN D2y los cables rojo y negro a los pines de 3.3V y GND, respectivamente.

- El Arduino Uno se conecta a la Raspberry Pi3 Model mediante un cable USB, permitiendo la transferencia de datos seriales entre ambos dispositivos.

Figura 27

Diagrama de Conexión Gateway



Fuente: Autor

Este montaje es fundamental para el proyecto de alerta contra heladas, ya que el Gateway se encarga de recibir los datos que leyeron los sensores, procesarlos y enviarlos a la nube o a una interfaz de usuario para su monitoreo en tiempo real, asegurando una respuesta rápida y eficiente ante posibles condiciones de heladas.

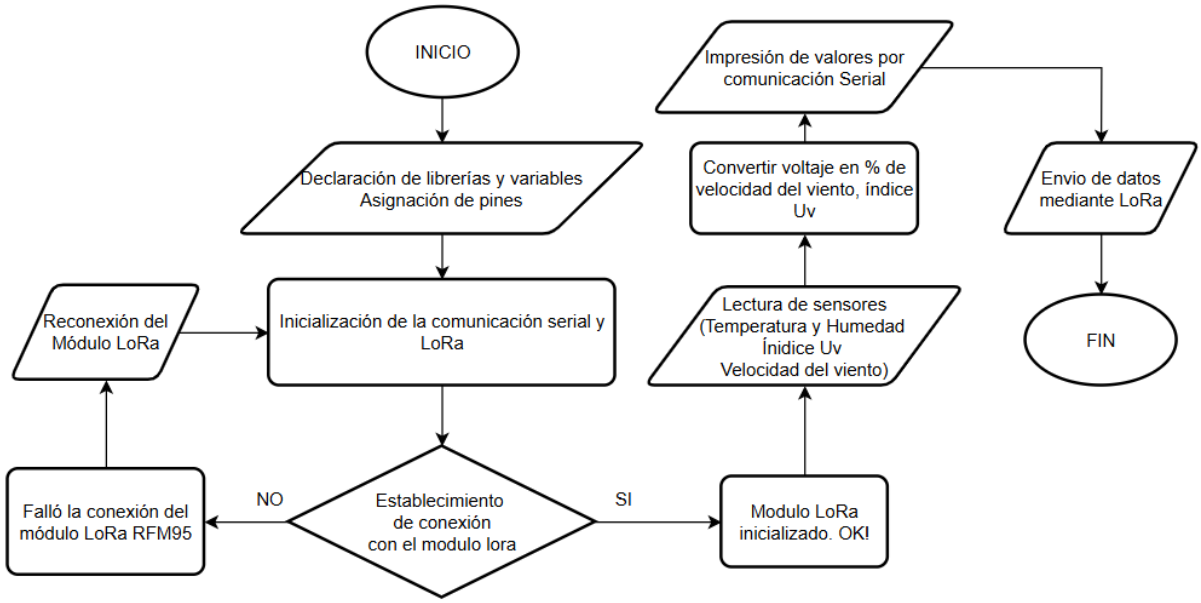
3.4.4. Diagrama de flujo del nodo sensor

El diagrama de flujo del nodo sensor describe el proceso completo, desde la recolección de datos hasta el envío de estos al Gateway. En este sistema, la ESP32 actúa como el núcleo central, comenzando con la declaración de librerías necesarias para la garantizar la

compatibilidad de los componentes del nodo, además de indicar la asignación de pines. A continuación, se inicializan las comunicaciones serial y LoRa. Si la inicialización de la comunicación falla, el sistema retorna al proceso para reintentarlo. En caso de éxito, se procede con la lectura de datos de los sensores DHT11, Guva Sd12s, y el anemómetro. Los datos son procesados por el ESP32, comprimiéndolos y enviándolos a través del módulo LoRa. Además, el flujo incluye la validación de datos, manejo de errores, y el control de la frecuencia de envío

Figura 28

Diagrama de flujo del Nodo Sensor



Fuente: Autor

La describe Figura 28 la programación del nodo sensor mediante un diagrama del flujo, este código es ejecutado por la ESP32 WROOM la cual es la encargada de procesar todos los datos en el Nodo Sensor.

3.4.5. Diagramas de flujo del Gateway

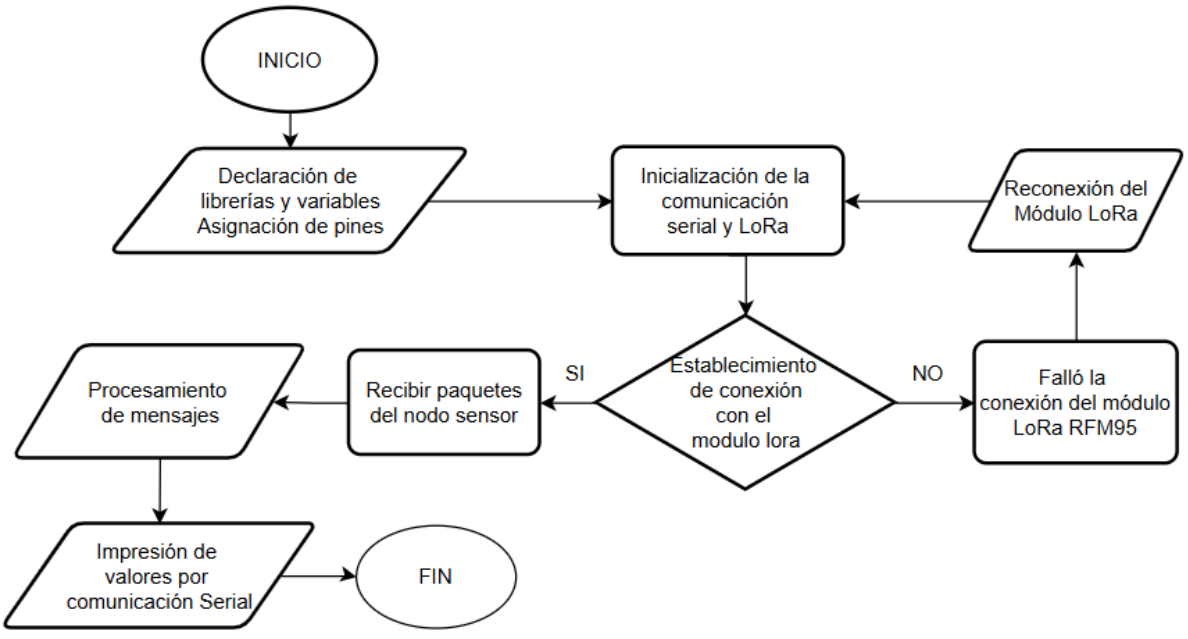
En el Gateway existen dos diagramas de flujo, el primero en el Arduino UNO y el otro la Raspberry Pi porque estos dos elementos se complementan para que el Gateway pueda procesar los datos recibidos a través de la comunicación LoRa. Estos diagramas de flujo del

Gateway muestran cómo se reciben los datos del nodo sensor, se procesan y se envían a la Raspberry Pi. El Arduino Uno recibe los datos del módulo LoRa, los procesa y los envía a la Raspberry Pi mediante conexión USB. La Raspberry Pi, con Node-RED, se encarga de almacenar, visualizar y enviar alertas si se detectan condiciones de helada.

Como se visualiza en la Figura 29 que describe la programación y funcionamiento del Arduino Uno este inicia con la declaración de librerías y variables que aseguren la compatibilidad, posteriormente se configura para metro de LoRA, adicional a esto se inicia la comunicación serial y LoRa, si esta falla se regresa el proceso hasta que se pueda solucionar el problema que posiblemente se manual, si no existe ningún problema se procede a recibir los paquetes enviados por el nodo sensor, estos datos son separados para posteriormente ser enviados de manera serial hacia la Rasbperry Pi.

Figura 29

Diagrama de Flujo en el Arduino UNO (Gateway)



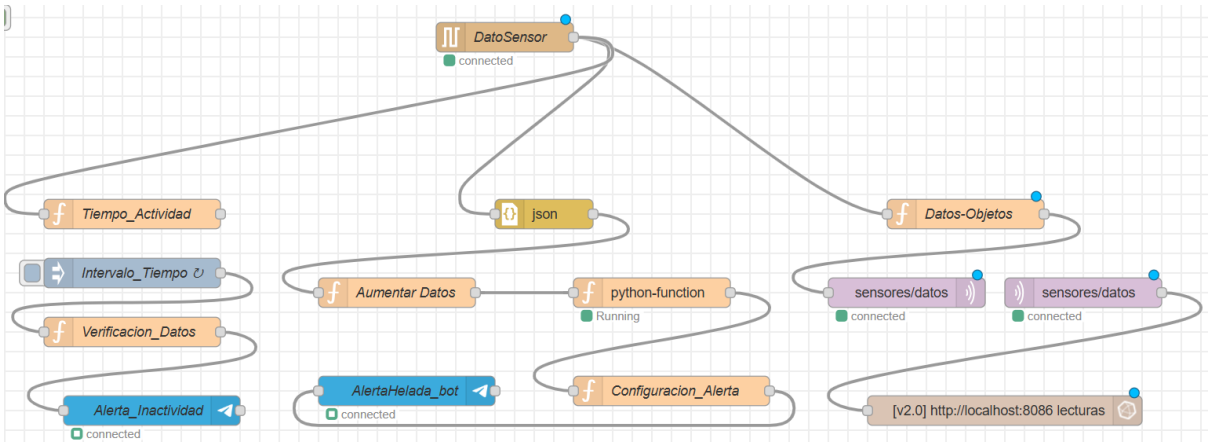
Fuente: Autor

Ahora la Figura 30 de igual manera describe el proceso realizado por la Rasbperry Pi, en este caso directamente se utiliza Node-Red la cual mediante sus nodos recibe los datos

enviados mediante puerto serial desde el Arduino, para luego ingresarlos a otro nodo donde esta previamente cargado el Algoritmo entrenado el cual arroja un resultado de predicción, presencia de helada o no en caso de no serlo, esta está conectada a una extensión que permite la configuración de mensajes preconfigurados a manera de alertas en Telegram. Por otro lado, los datos ingresados son enviados a la base de datos InfluxDB mediante MQTT y posteriormente visualizados.

Figura 30

Diagrama de flujo Gateway en Rasbperry Pi



Fuente: Autor

3.4.6. Calibración y lectura de datos mediante Sensores

Todos los sensores tienen diferente sensibilidad al leer datos correspondientes. Esto depende de su fabricante, las especificaciones que se realizan en su Datasheet son importantes para obtener los valores precisos en cada una de las variables. Estas especificaciones permiten realizar fórmula de cálculo necesarias para convertir las lecturas a unidades del Sistema Internacional de manera adecuada. Además, contar valores precisos que puedan ser comparados con plataformas o datos verídicos permite validar y dar fiabilidad las lecturas de los datos realizadas.

3.4.6.1. Programación y Calibración DHT11

Para la calibración de este sensor es importante describir que cuenta con un microcontrolador interno el cual automáticamente calibra los datos realizando la conversión a unidades estándares en el caso de la temperatura en C° y el de la humedad en %, también trasmite directamente los datos, una de las ventajas de usar el sensor DHT11 es que la precisión de los sensores es segura por su calibración de fábrica.

Figura 31

Código Sensor DHT11

```
#include <DHT.h>
#include <DHT_U.h>

//*************************DHT11*****************************
#define PIN_DHT 4           // Pin conectado al sensor DHT11
#define tipo_DHT DHT11       // Tipo de sensor
DHT dht(PIN_DHT, tipo_DHT);  // Instancia del sensor DHT

void setup() {

  // Inicialización del sensor DHT11
  dht.begin();

}

//*************************LOOP*****************************
void loop() {
  // Leer datos de los sensores
  float humedad = dht.readHumidity();
  float temperatura = dht.readTemperature();

  // Comprobación de datos del sensor de Humedad y Temperatura
  if (isnan(humedad) || isnan(temperatura)) {
    Serial.println("Error al leer del sensor DHT!");
    return;
  }
}
```

Fuente: Autor

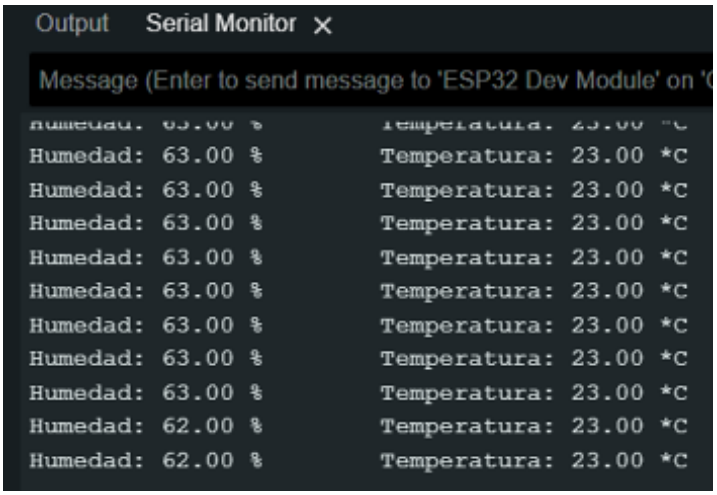
Para poder obtener los valores del sensor incluimos el código de la Figura 31 en la ESP32 el cual utiliza librerías como DHT.h Y DHT_U.h que permiten trabajar con los sensores DTH. Luego se define el PIN al cual está conectado el sensor y el tipo de sensor DHT que se utilizara. También se crea una instancia con las definiciones anteriores, permitiendo la

comunicación con el sensor. Luego se procede a inicializar el sensor para que esté preparado a realizar las lecturas. Se utiliza las funciones de lectura tanto para la humedad como la temperatura y se almacena en variables para posteriormente imprimir los datos y enviarlos.

Y finalmente se agrega un condicional para la comprobación de errores en el cual se verifica si los datos recibidos son números, caso contrario se invalida la lectura, estos errores pueden presentarse debido a conexiones físicas del sensor.

Figura 32

Lectura Sensor DTH11



Fuente: Autor

Los resultados esperados de la lectura de este sensor se muestran en la Figura 32, donde se observa que se obtiene la temperatura ambiental en C° y la humedad ambiental en %.

3.4.6.2. Programación y Calibración Sensor Guva S12SD

El sensor Guva S12SD se conecta a la ESP32 mediante un PIN con convertidor ADC, este valor debe ser convertido a índice UV, por lo cual se aplica diferentes fórmulas teniendo en cuenta las características que ofreces el fabricante en su DataSheet. Este tipo de conversiones de unidades nos ayudan a tener valores precisos. Por ello se agrega el código de la Figura 33 a la programación de la ESP32.

Figura 33

Código Sensor GUVA-S12SD

```
//**********************GUVA S12SD**************************
#define PIN_sensorUV 35       // Pin conectado al sensor UV

void loop() {

  // Lectura y calibración del sensor UV
  int valorAnalogicoUV = analogRead(PIN_sensorUV);
  float voltajeUV = (valorAnalogicoUV * 3.3) / 4095;
  float indiceUV = voltajeUV / 0.1;
```

Fuente: Autor

Primero se define el PIN en el cual está conectado el sensor y esto se puede comprobar en la Figura 26, luego se procede a obtener el valor de ingreso mediante una función de lectura, hay que tomar en cuenta que por el ADC se asigna un valor que varía entre 0 y 4095. Este valor debemos convertirlo a voltaje por lo cual se aplica la formula (7) donde:

$$Voltaje\ de\ Salida = \frac{Valor\ ADC}{4095} * 3.3V \qquad (7)$$

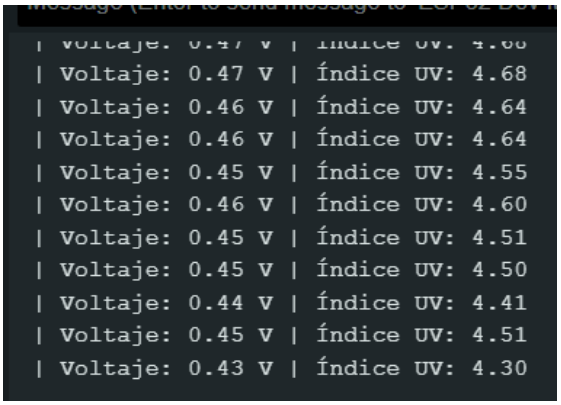
- Voltaje de Salida: es el voltaje que general el sensor al realizar una lectura
- Valor ADC: es el valor asignado por la ESP al convertir el valor de analógico a digital
- Constante 4095: es el valor máximo del valor ADC
- Constante 3.3V: es el valor al cual trabaja el sensor.

Después el voltaje obtenido es calibrado según el Datasheet del fabricante el cual especifica que para cada 0.1V de salida, el índice UV aumenta en 1 por ello el voltaje obtenido se multiplica por 0.1 y se obtiene el índice UV.

El índice UV es almacenado en una variable la cual se puede imprimir y enviar. Los datos esperados son los de la Figura 34. Se puede observar que el índice UV está en un rango moderado.

Figura 34

Lectura Sensor GUVA-S12SD



Fuente: Autor

3.4.6.3. Programación y Calibración Anemómetro

El anemómetro seleccionado utiliza una fórmula concreta que permite calibrar el sensor de manera eficiente, obteniendo valores reales. Esta fórmula se considera para la calibración y programación del sensor que se presenta a continuación.

Figura 35

Código Sensor Anemómetro

```
//**************ANEMÓMETRO**********************************
#define PIN_anemometro 34    // Pin conectado al anemómetro

//*********************************************************

void loop() {

  // Lectura y calibración del anemómetro
  float ValorAnalogicoA = analogRead(PIN_anemometro);
  float voltaje_anemometro = (ValorAnalogicoA * 3.3) / 4095;
  float velocidadKmh = (voltaje_anemometro * 25) * 3.6;
}
```

Fuente: Autor

En primer lugar, como se muestra la Figura 35, el código del anemómetro comienza declarando el PIN al cual está conectado el sensor como se observa en el diagrama de conexión la Figura 26, en este caso el PIN D34. Posteriormente se realiza la lectura del sensor y se

almacena el valor obtenido en la variable designada. A continuación, se realiza la conversión del valor ADC a voltaje como en el sensor anterior utilizando la formula (7).

Una vez obtenido el voltaje de salida del anemómetro, se calcula de la velocidad del viento para lo cual, según especificaciones del fabricante, el voltaje de salida obtenido debe ser multiplicado por un factor predeterminado en este caso 25. Entonces la fórmula aplicar en este caso es la siguiente:

$$Velocidad\ del\ viento\ \left(\frac{m}{s}\right) = (Voltaje\ de\ Salida * 25) * 3.6 \qquad (8)$$

Como se observa en la ecuación (8), el resultado se expresa en (m/s) por lo cual aplicamos una conversión a (km/h), fara facilitar su interpretación.

Figura 36

Lectura Sensor Anemómetro

```
Voltaje (V): 0.00 | Velocidad del viento (km/h): 0.00
Voltaje (V): 0.04 | Velocidad del viento (km/h): 3.34
Voltaje (V): 0.09 | Velocidad del viento (km/h): 8.12
Voltaje (V): 0.13 | Velocidad del viento (km/h): 11.82
Voltaje (V): 0.17 | Velocidad del viento (km/h): 14.87
Voltaje (V): 0.20 | Velocidad del viento (km/h): 17.55
Voltaje (V): 0.22 | Velocidad del viento (km/h): 20.16
Voltaje (V): 0.14 | Velocidad del viento (km/h): 12.55
Voltaje (V): 0.02 | Velocidad del viento (km/h): 1.96
```

Fuente: Autor

Como se muestra en la Figura 36 los datos esperados de la lectura del Anemómetro son variables por que cambia según la ráfaga de viento, de igual manera se observa el voltaje transformado en el cual se puede aplicar la formula del DataSheet del sensor y verificar su calibración.

3.4.7. Programación LPWAN

Los módulos LoRa requieren ser configurados para establecer comunicación y así transmitir y receptar paquetes a larga distancia. La configuración de parámetros de comunicación como frecuencias, transmisión entre otras deben ser definidas tanto en el nodo

Gateway como en el nodo sensor para lo cual se realiza programación similar para cada uno de los microcontroladores que están implicados.

3.4.7.1. Programación LoRa en la ESP32 WROOM

En la Figura 37 se describe la integración de una librería, declaración de pines y además de parámetros que se detallan de mejor manera a continuación. LoRa Se comienza integrando la librería LoRa.h que permite acceder a funciones para configurar módulos LoRa. Luego se declara los pines en los cuales está conectado el módulo en este caso los más importantes como el NSS por el cual se selecciona el módulo LoRa para comunicación SPI, RST que permite reiniciar el módulo LoRa y DIO0 que es el más importante porque a través de este se detectan paquetes LoRa. Además, se crea una variable de identificación de la dirección del nodo, esta servirá para formar el paquete y validar el mismo en el receptor.

Figura 37

Código LoRa ESP32 Variables y Pines.

```
#include <LoRa.h>

//*****************************LoRA*************************************************
#define PIN_nss 5
#define PIN_rst 14
#define PIN_dio0 2

// Configuraciones LoRa
const int sf = 7;            // Spreading Factor (7-12)
const int bw = 125000;       // Bandwidth (Hz)
const int cr = 5;            // Coding Rate
const int nodeAddress = 1;   // Dirección del nodo LoRa
```

Fuente: Autor

Dentro de nuestra función setup como se observa en la Figura 38 configuramos los pines declarados y utilizados por la ESP32 en este caso los mismos declarados en la Figura 37. Después se procede con la inicialización del módulo LoRa, utilizamos un condicional donde se comprueba la inicialización, dentro de este se configura la frecuencia de operación, 915 MHz que solos permitidos en Ecuador. En caso de que esta falle se imprime un mensaje de

fallo y espera 5 segundos para reiniciar el microcontrolador. Si este se inicia bien se procede con la configuración de los parámetros:

- Spreading Factor (SF): Este puede variar entre 7 y 12, representando 7 el valor más bajo con menor sensibilidad y alcance, pero con una velocidad de transmisión alta.
- Ancho de Banda (BW): Entre más alto sea el ancho de banda mayor velocidad de transmisión, pero menor alcance, por esto se selecciona los 125kHz.
- Coding Rate (CR): Los bits utilizados para corrección de errores que se establece como 4/CR donde CR puede varias de 5 a 8. Entre más bajo sea el valor mejor corrección de errores.

También se utiliza funciones como LoRa.enableCrc() la cual activa la detección de errores y LoRa.setSyncWord(0xF3) que activa la sincronización con dispositivos LoRa, para que exista comunicación los dispositivos deben tener el mismo valor de sincronización.

Figura 38

Código LoRa ESP32 Función Setup.

```
void setup() {

  // Configuración de pines del módulo LoRa
  LoRa.setPins(PIN_nss, PIN_rst, PIN_dio0);

  // Inicialización del módulo LoRa
  if (!LoRa.begin(915E6)) {
    Serial.println("Error: Falló la conexión con el módulo LoRa. Reiniciando...");
    delay(5000);
    ESP.restart();
  }

  // Configuración de parámetros LoRa
  LoRa.setSpreadingFactor(sf); // Factor de dispersión (7-12)
  LoRa.setSignalBandwidth(bw); // Ancho de banda (125-250-500 kHz)
  LoRa.setCodingRate4(cr);     // Coding Rate
  LoRa.enableCrc();            // Habilita la comprobación de redundancia cíclica (CRC)
  LoRa.setSyncWord(0xF3);      // Sincroniza con los elementos de la red

  // Impresión de parámetros configurados DE LoRa
  Serial.println("Módulo LoRa inicializado correctamente");
  Serial.print("Spreading Factor (SF): ");
  Serial.println(sf);
  Serial.print("Bandwidth (BW): ");
  Serial.println(bw);
  Serial.print("Coding Rate (CR): 5/");
  Serial.println(cr);
}
```

Fuente: Autor

Dentro del Setup se hace la impresión de estos parámetros mediante comunicación serial para compáralos con el receptor y evitar errores de comunicación que no sen físicas.

Figura 39

Código LoRa ESP32 Función Loop

```
void loop() {

  // Preparar los datos en formato char[]
  char data[128];
  snprintf(data, sizeof(data), "T:%.2f H:%.2f W:%.2f UV:%.2f", temperatura, humedad, velocidadKmh, indiceUV);

  // Imprimir datos en el monitor serial
  Serial.println("Datos a enviar:");
  Serial.println(data);

  // Enviar datos mediante LoRa
  LoRa.beginPacket();         // Iniciar el paquete
  LoRa.write(nodeAddress);    // Dirección del nodo
  LoRa.print(data);           // Agregar datos
  LoRa.endPacket();           // Finalizar el paquete

  Serial.println("Datos enviados con éxito.");
  delay(10000); // Enviar datos cada 10 segundos
}
```

Fuente: Autor

En la función Loop representada en la Figura 39 que se ejecuta en un bucle infinito se prepara los datos obtenidos de los sensores en un arreglo se caracteres con un tamaño de 128 bytes, para posteriormente utilizar este arreglo junto con la función snprintf que permite formatear los datos y almacenarlos en una variable, para verificar si este arreglo está bien armado se imprime a través de comunicación serial donde se esperan los datos como en la Figura 40 donde se observa que el arreglo está bien configurado.

Figura 40

Impresión arreglo a enviar

```
T:25.50 H:65.30 W:12.40 UV:3.20
T:25.50 H:65.30 W:12.40 UV:3.20
T:25.50 H:65.30 W:12.40 UV:3.20
T:25.50 H:65.30 W:12.40 UV:3.20
```

Fuente: Autor

Después se procede con él envió de datos mediante LoRa para lo cual se crea un paquete que contiene: inicio del paquete a través de la función LoRa.beginPacket(), la autenticación del nodo previamente declarada, el arreglo de datos, y la finalización del paquete. Por último, se hace una impresión que confirme el envío de los datos.

3.4.7.2. Programación LoRa en Arduino Uno

En el código de LoRa para el Arduino Uno en la parte de inicialización de la comunicación de la Figura 41 es similar al código de la ESP32, tiene la misma lógica explicada anteriormente.

Figura 41

Código LoRa Arduino Uno Variables y Función Setup

```
#include <SPI.h>
#include <LoRa.h>

// Pines del módulo LoRa
#define PIN_nss 10
#define PIN_rst 9
#define PIN_dio0 2

// Configuración LoRa
const int sf = 7;          // Spreading Factor (7-12)
const long bw = 125E3;     // Bandwidth (Hz)
const int cr = 5;          // Coding Rate
const int expectedNode = 1; // Dirección esperada del nodo transmisor

void setup() {
  // Inicialización del monitor serial
  Serial.begin(9600);
  while (!Serial);
  Serial.println("Nodo Receptor LoRa - Iniciando");

  // Configuración de pines del módulo LoRa
  LoRa.setPins(PIN_nss, PIN_rst, PIN_dio0);

  // Inicialización del módulo LoRa
  if (!LoRa.begin(915E6)) {
    Serial.println("Error: Falló la conexión con el módulo LoRa. Reiniciando...");
    while (1);
  }

  // Configuración de parámetros LoRa
  LoRa.setSpreadingFactor(sf);
  LoRa.setSignalBandwidth(bw);
  LoRa.setCodingRate4(cr);
  LoRa.enableCrc();
  LoRa.setSyncWord(0xF3);
```

Fuente: Autor

Continuamos con la función Loop donde como se observa en la Figura 42 se empieza con la función LoRa.parsePacket() que comprueba si existe algún paquete para ser leído, en caso de haberlo se continua con el condicional dentro del cual lee el primer byte del paquete recibido que es la dirección del nodo y realiza una comparación con la dirección del nodo declarado previamente y esperado, filtrando paquetes desconocidos he imprimiendo un mensaje de paquete no autorizado en caso de no coincidir la dirección del nodo.

Si la dirección del nodo coincide se continua con la lectura del resto del paquete, lee el arreglo de datos enviados y lo coloca en otra variable. Por último, llama a la función de procesar datos la cual esta descrita en la Figura 43.

Figura 42

Código LoRa Arduino Uno Función Loop

```
void loop() {
  // Comprobar si hay un paquete disponible
  int packetSize = LoRa.parsePacket();

  if (packetSize) {
    // Leer la dirección del nodo transmisor
    int senderNode = LoRa.read();
    if (senderNode != expectedNode) {
      Serial.println("Paquete recibido de un nodo no autorizado.");
      return;
    }

    // Leer el resto del paquete
    String data = "";
    while (LoRa.available()) {
      data += (char)LoRa.read();
    }

    // Imprimir el paquete recibido
    Serial.print("Datos recibidos: ");
    Serial.println(data);

    // Procesar los datos
    procesarDatos(data);
  }
}
```

Fuente: Autor

Dentro la función procesar datos se declaran variables en cual almacenar por separado los datos de cada sensor, se extrae las etiquetas del mensaje puestas en el arreglo, valida el

formato del mensaje, si no encuentra alguna etiqueta el arreglo no corresponde al formato esperado, pero si es el formato esperado se procede a almacenar en cada variable el dato y este ya se encuentra para ser enviado por comunicación serial hacia la Rasbperry Pi.

Figura 43

Código LoRa Arduino Uno Función Procesar Datos

```
void procesarDatos(String data) {
  // Variables para almacenar los datos procesados
  float temperatura = 0.0;
  float humedad = 0.0;
  float velocidadKmh = 0.0;
  float indiceUV = 0.0;

  // Buscar y extraer cada valor del mensaje recibido
  int tempIndex = data.indexOf("T:");
  int humIndex = data.indexOf("H:");
  int windIndex = data.indexOf("W:");
  int uvIndex = data.indexOf("UV:");

  if (tempIndex == -1 || humIndex == -1 || windIndex == -1 || uvIndex == -1) {
    Serial.println("Error: Formato de datos no válido.");
    return;
  }

  // Extraer los valores usando substring
  temperatura = data.substring(tempIndex + 2, humIndex).toFloat();
  humedad = data.substring(humIndex + 2, windIndex).toFloat();
  velocidadKmh = data.substring(windIndex + 2, uvIndex).toFloat();
  indiceUV = data.substring(uvIndex + 3).toFloat();

  // Imprimir los datos procesados
  Serial.println("Datos Procesados:");
  Serial.print("Temperatura: ");
  Serial.print(temperatura);
  Serial.println(" °C");
```

Fuente: Autor

3.4.8. Programación y Entrenamiento de la IA

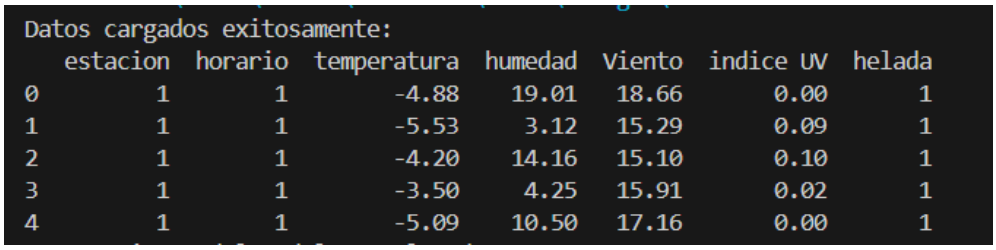
Para el entrenamiento de una IA es necesario acudir a un método de entrenamiento en nuestro caso optamos el entrenamiento por programación. En la cual se eligió la técnica de IA del algoritmo árbol de decisiones el cual especifica plantear varios escenarios o ramificaciones de situaciones que tienen una misma raíz, esta técnica se adapta de manera eficiente al objetivo del proyecto. Se presenta a continuación el proceso para el entrenamiento de la IA.

3.4.8.1. Creación de Base de Datos para el entrenamiento de IA

El entrenamiento de un IA implica varios factores como el uso de condiciones según la investigación realizada y para aumentar su eficiencia la utilización de una base de datos robusta, esta base de datos debe contar con todo tipo de escenarios para poder abrir más ramificaciones y tener precisión en la predicción del algoritmo. Por ello se conformó una base de datos con etiquetas de entrada como: estación, horario, temperatura. humedad, velocidad del viento, índice UV y helada, presentadas en la Figura 44.

Figura 44

Fragmento de Base de Datos para el entrenamiento.



Datos cargados exitosamente:

	estacion	horario	temperatura	humedad	Viento	indice UV	helada
0	1	1	-4.88	19.01	18.66	0.00	1
1	1	1	-5.53	3.12	15.29	0.09	1
2	1	1	-4.20	14.16	15.10	0.10	1
3	1	1	-3.50	4.25	15.91	0.02	1
4	1	1	-5.09	10.50	17.16	0.00	1

Fuente: Autor

Estación: esta solo puede variar en 1 cuando representan la estación de invierno que implica en los meses de junio a septiembre y 0 cuando representan la estación de verano perteneciente a los meses de octubre a mayo. Este dato se toma en cuenta por que según la investigación realizada en el capítulo anterior la presencia frecuente de helada es en la estación de invierno, sin embargo, al ser un fenómeno natural no se puede asegurar la presencia solo por la estación.

El horario: de igual manera que la etiqueta anterior esa varía en función de la hora, si es la noche/madrugada esta tiene un valor de 1 pero si se la lectura se encuentra en el día su valor será 0.

Temperatura ambiental, humedad ambiental, velocidad del viento y el índice UV: dependen de la lectura de los sensores y la calibración de estos. Estos datos representan en su

mayoría la base de datos, pero otros fueron tomados como referencia de datos históricos de eventos pasados.

Con todos estos datos recolectados, simulados e históricos se obtiene una base sólida de 8240 datos individuales que representan 1030 posibilidades de entrenamiento.

3.4.8.2. Código de Entrenamiento

El código de entrenamiento tiene una lógica básica desde el entorno resumido, esta carga la base de datos exportada en un archivo .csv, luego prepara y divide los datos en entrenamiento y prueba. Para luego proceder a entrenar el árbol de decisión, luego se evalúa el modelo, se guarda el modelo entrenado en un archivo. pkl y se visualiza gráficamente el funcionamiento del árbol de decisiones. Sin embargo, tiene más complejidad de la mencionada por lo cual se describe las partes del código a continuación.

Figura 45

Código de Entrenamiento Importación de Librerías

```
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier, export_text
import joblib
import matplotlib.pyplot as plt
```

Fuente: Autor

Como se observa en la Figura 45 se utiliza variar librerías las cuales permiten la manipulación y análisis de la base de datos, crear funciones para el entrenamiento, comparación de eficiencia y permiten guardar el algoritmo entrenado, por último, también se importa la librería para la generación de graficas.

Figura 46

Código de Entrenamiento Importación de Base de Datos

```
file_path = r'C:\Users\Kate G\Downloads\TESIS\Datos\casosheladas.csv'  # Ca

try:
    # Leer el archivo con punto y coma como delimitador
    data = pd.read_csv(file_path, delimiter=';', encoding='utf-8')
    print("Datos cargados exitosamente:")
    print(data.head())
except FileNotFoundError:
    print(f"Error: El archivo no fue encontrado en la ruta: {file_path}")
    exit()
except Exception as e:
    print(f"Ocurrió un error al cargar el archivo: {e}")
    exit()
```

Fuente: Autor

En la Figura 46 se muestra otro fragmento del código de entrenamiento de la IA, donde se realiza la carga de la base de datos. Además, se visualiza una parte de los datos para verificar que coincidan con los valores esperados. Finalmente, se implementa un manejo de errores, permitiendo detectar y notificar posibles inconvenientes con el archivo que contiene los datos de entrenamiento

Figura 47

Código de Entrenamiento separación de datos y entrenamiento.

```
# 2. Separar características (X) y etiqueta (y)
X = data[['estacion', 'horario', 'temperatura', 'humedad', 'viento', 'indice_UV']]
y = data['helada']

# 3. Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Entrenar el Árbol de Decisión
clf = DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)  # Puedes ajustar max_depth
clf.fit(X_train, y_train)
print("Entrenamiento del modelo completado.")
```

Fuente: Autor

La Figura 47 se aprecia la parte más importante de todo el código donde se definen los datos de entrada como X, los cuales incluyen las variables: estación, horario, temperatura,

humedad, viento e índice UV. La variable objetivo o etiqueta de salida, denominada Y, corresponde a la ocurrencia de helada.

Seguidamente, los datos se dividen en dos grupos, de entrenamiento y prueba, asignando 80% para el entrenamiento del modelo y 20% para su evaluación. Durante el entrenamiento del Árbol de Decisión, el modelo busca agrupar los datos en conjuntos con características similares, permitiendo mejorar la precisión de las predicciones y minimizar errores. Para lograr esta agrupación, se utiliza un criterio matemático denominado entropía, el cual mide el nivel de incertidumbre en los datos. El árbol explorará diferentes divisiones hasta encontrar la óptima, asegurando que la entropía se reduzca progresivamente en cada nivel.

Para controlar el crecimiento del árbol y evitar problemas como el sobreajuste, se establece un límite en la profundidad máxima del modelo, es decir, el número de niveles desde la raíz hasta las hojas. Este parámetro es crucial, ya que afecta la complejidad y el desempeño del modelo. En este caso, se utilizó una profundidad de 5, lo que permite un equilibrio entre precisión y generalización. Finalmente, una vez entrenado el modelo, se realizan pruebas utilizando el 20% de los datos reservados, permitiendo evaluar la precisión del árbol y validar la efectividad de las divisiones generadas.

Figura 48

Código de Entrenamiento evaluación de precisión y guardar el modelo

```
# 5. Evaluar el modelo
accuracy = clf.score(X_test, y_test)
print(f"Precisión del modelo en datos de prueba: {accuracy * 100:.2f}%")

# Validación cruzada
scores = cross_val_score(clf, X, y, cv=5)
print(f"Precisión media con validación cruzada: {scores.mean() * 100:.2f}%")

# 6. Guardar el modelo
joblib.dump(clf, 'arbol_heladas.pkl')
print("Modelo guardado como 'arbol_heladas.pkl'.")
```

Fuente: Autor

Luego de entrenar el Árbol de Decisión, es fundamental evaluar su precisión. Para ello, se utilizan diversas métricas, entre ellas accuracy, que mide el porcentaje de predicciones correctas con relación al total de muestras evaluadas, como se muestra en la ecuación (9). Además, se aplica validación cruzada con 5 particiones, lo que permite evaluar el modelo en diferentes subconjuntos de datos, obteniendo así una medición más confiable y robusta de su desempeño.

$$Accuracy = \frac{Número\ de\ predicciones\ correctas}{Total\ de\ predicciones\ realizadas} \quad (9)$$

Finalmente, como se observa en la Figura 48, para facilitar su uso en futuras predicciones, el modelo entrenado se guarda en un archivo con extensión .pkl, lo que permite cargarlo sin necesidad de volver a entrenarlo.

Figura 49

Código de Entrenamiento Visualización del árbol de decisiones.

```
# 7. Visualizar el Árbol de Decisión
# Exportar las reglas del árbol
tree_rules = export_text(clf, feature_names=['estacion', 'horario', 'temperatura', 'humedad', 'viento', 'indice_UV'])
print("Reglas del Árbol de Decisión:")
print(tree_rules)

# Visualizar el árbol gráficamente
from sklearn.tree import plot_tree

plt.figure(figsize=(20, 10))
plot_tree(clf, feature_names=['estacion', 'horario', 'temperatura', 'humedad', 'viento', 'indice_UV'],
          class_names=['No Helada', 'Helada'], filled=True, rounded=True)
plt.show()
```

Fuente: Autor

La última parte del código es opcional y permite interpretar visualmente la estructura del Árbol de Decisión, incluyendo su raíz, nodos intermedios y hojas. Para ello, se emplean diversas funciones que facilitan la representación gráfica del modelo entrenado, proporcionando una mejor comprensión de su funcionamiento. Esta visualización se muestra en la Figura 49.

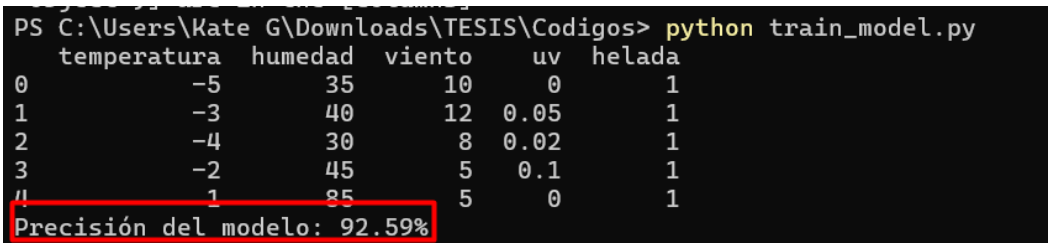
3.4.8.3. Árbol de Decisiones Entrenado.

Al ejecutar el código descrito en la Sección 3.4.8.2, se espera confirmar el proceso de entrenamiento y visualizar la precisión obtenida. Un modelo es más efectivo cuanto más se acerca al 100% de precisión, ya que indica una mayor capacidad de predicción. Se llevaron a cabo varios entrenamientos, obteniendo inicialmente una precisión del 92.59% como se observa en la Figura 50, un resultado altamente aceptable para un modelo de esta complejidad. Sin embargo, la precisión se mejoró al incrementar la cantidad de datos de entrenamiento y ampliar las variables consideradas.

En el primer entrenamiento, solo se tomaron en cuenta los datos recopilados por los sensores, como temperatura, humedad, velocidad del viento e índice UV. Posteriormente, se optimizó el modelo al incluir variables adicionales, como la estación y el horario, además de aumentar el volumen de datos disponibles para el entrenamiento.

Figura 50

Precisión del modelo primer entrenamiento

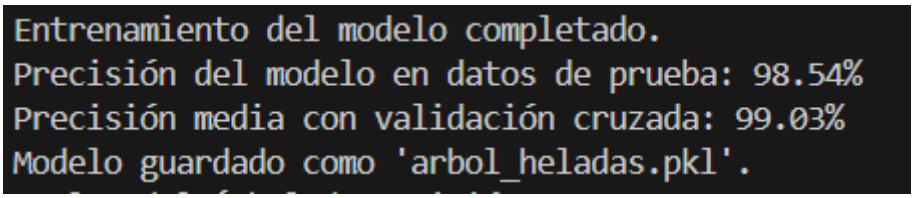


Fuente: Autor

Estos ajustes contribuyeron a mejorar significativamente la precisión del modelo, haciéndolo más robusto y confiable en la predicción de heladas. En el entrenamiento final de la Figura 51, se logró una precisión del 98.54%, mientras que la validación cruzada arrojó un 99.03%, lo que confirma la alta capacidad del modelo para generalizar y predecir con precisión. Finalmente, el modelo fue guardado correctamente, asegurando su disponibilidad para futuras predicciones sin necesidad de volver a entrenarlo.

Figura 51

Precisión del modelo entrenamiento final



Fuente: Autor

Con la opción de visualización del modelo, es posible interpretar su lógica de decisión, como se muestra en la Figura 52, donde se aprecian las reglas aprendidas por la IA entrenada. La raíz del árbol representa una condición lógica inicial basada en las características del conjunto de datos, mientras que las hojas (al final de cada rama) indican la clase predicha.

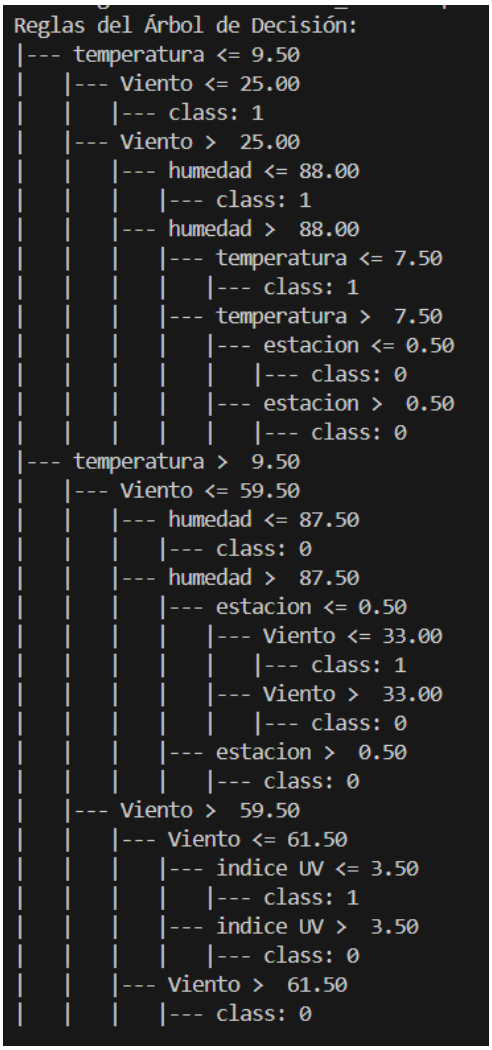
En este caso, la raíz evalúa si la temperatura es mayor o menor a 9.50°C, dividiendo el árbol en dos ramas principales. A partir de estas divisiones, se generan más ramas considerando otras condiciones, como si la velocidad del viento es mayor o menor a 25 km/h, seguido por variables adicionales como la humedad. Estas divisiones continúan hasta llegar a las hojas, que determinan la clase de predicción:

- Clase 1: Indica la presencia de helada.
- Clase 0: Indica la ausencia de helada.

Además, se puede observar que la temperatura es el factor más relevante, ya que aparece en el primer nivel jerárquico del árbol, lo que refleja su influencia predominante en la predicción. Este enfoque jerárquico facilita la interpretación del modelo y permite entender cómo las diferentes características interactúan para llegar a una decisión final.

Figura 52

Visualización reglas del árbol de decisión entrenado



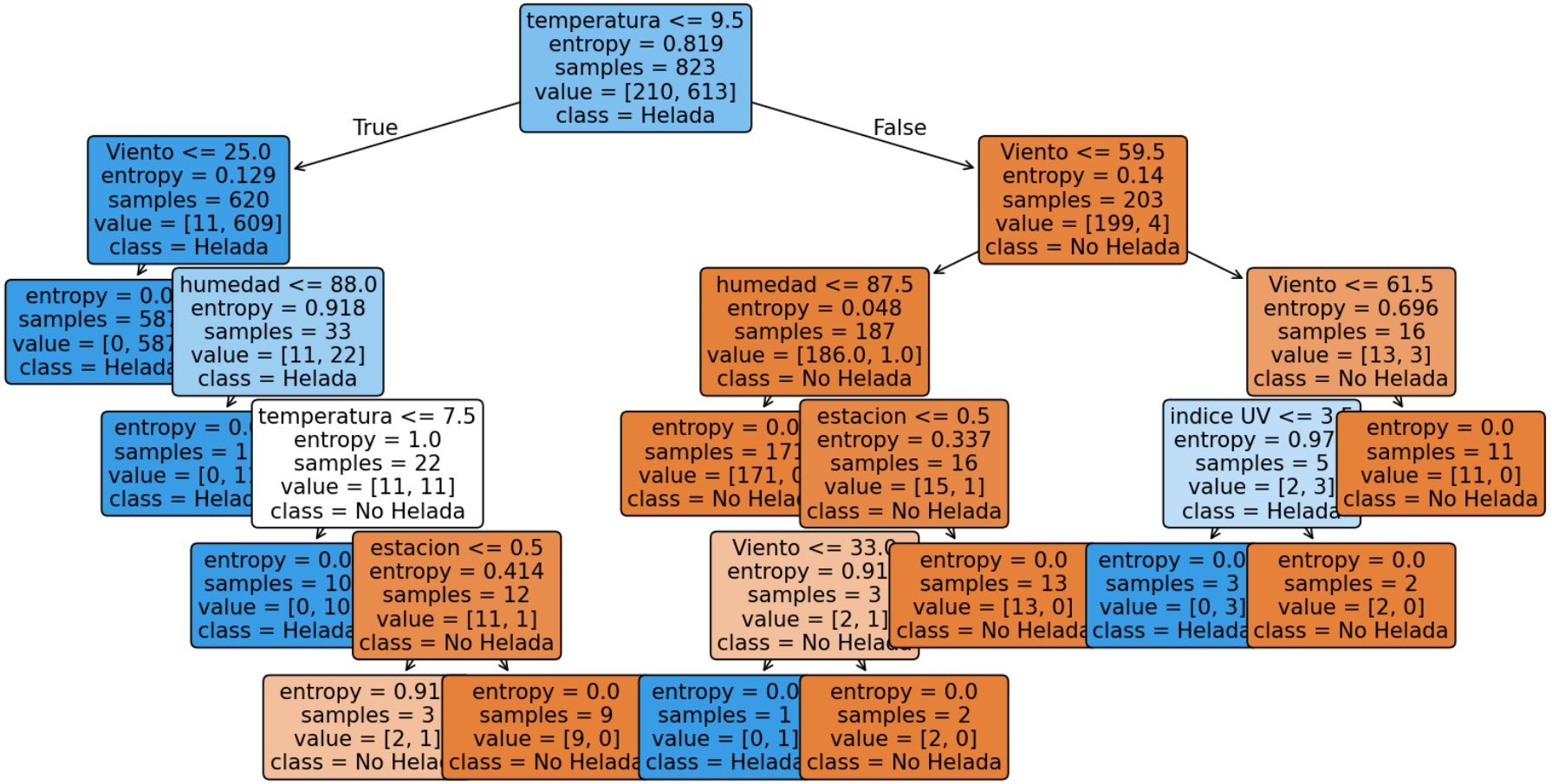
Fuente: Autor

Al igual que en la interpretación anterior, la Figura 53 permite observar la entropía utilizada en la formación de cada rama del Árbol de Decisión. Esta gráfica también presenta información adicional clave en cada nodo:

- Entropía: Indica el nivel de incertidumbre en el nodo (menor entropía significa mayor homogeneidad).
- Samples: Representa el número total de muestras evaluadas en ese nodo o rama.
- Value: Muestra la distribución de casos en cada clase (por ejemplo, helada o no helada).
- Clase predominante: Indica la clase que domina en ese nodo (helada o no helada).

Figura 53

Visualización árbol de decisión entrenado



Fuente: Autor

Además, los colores de los nodos ayudan a interpretar visualmente el resultado:

- Azul: Representa nodos que terminan en una hoja que predice helada.
- Naranja: Representa nodos que terminan en una hoja que predice que no habrá helada.

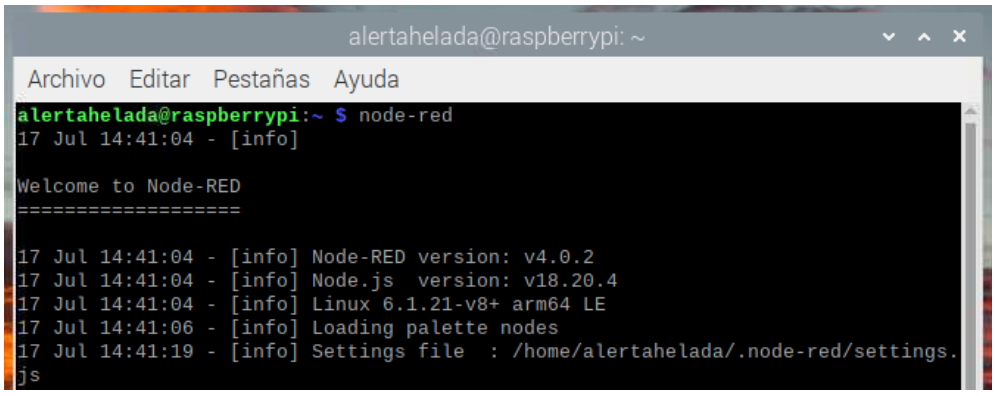
Esta visualización facilita la comprensión de cómo el modelo toma decisiones basadas en las características de los datos, mostrando claramente las divisiones y predicciones en cada rama del árbol.

3.4.9. Plataforma de Gestión Node Red

En nuestra Raspberry Pi, instalamos el servicio Node-RED, una herramienta que permite crear diagramas de flujo, cuenta con una amplia variedad de librerías y complementos que se pueden integrar para ajustarse a los objetivos específicos del desarrollo. El servicio se inicia con el comando node-red visualizado en la Figura 54, lo que permite acceder a la interfaz para configurar y personalizar los flujos. Un aspecto importante a tener en cuenta es que, una vez completada la configuración, es necesario habilitar el servicio para que se ejecute al encender la Raspberry Pi, garantizando así que el sistema funcione de manera continua.

Figura 54

Activación del servicio de Node-Red



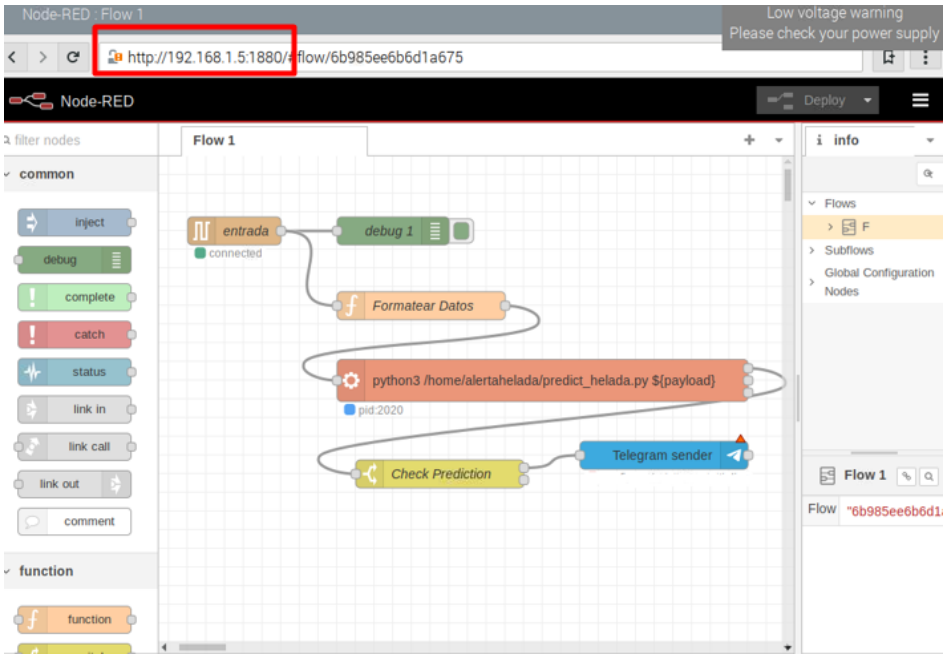
Fuente: Autor

3.4.9.1. Ingreso de interfaz

Para acceder a la interfaz de Node-RED, se puede utilizar cualquier navegador web y escribir la dirección IP del localhost, seguida del puerto en el que opera el servicio. En este caso, el puerto es el 1880, como se muestra en la Figura 55. Una vez ingresada la dirección, se despliega la interfaz gráfica, que estará lista para ser configurada según las necesidades del proyecto.

Figura 55

Ingreso a la Interfaz Node-Red



Fuente: Autor

3.4.9.2. Enlace de parámetros adicionales

Los datos recibidos en la Raspberry Pi a través de la comunicación serial corresponden a las mediciones capturadas por los sensores conectados al sistema. Sin embargo, el entrenamiento de nuestra IA incluye dos parámetros adicionales: la estación del año y el horario del día. Para incorporar estos datos, se utiliza un nodo de función que ejecuta el código descrito en la Figura 56

Figura 56

Generación de variables adicionales

```
let now = new Date(); // Obtener la fecha y hora actual

// Calcular horario (0 o 1)
let hora = now.getHours();
let minutos = now.getMinutes();
if ((hora > 5 && hora < 17) || (hora === 17 && minutos <= 30)) {
    msg.payload.horario = 0; // 6:00 AM a 5:30 PM
} else {
    msg.payload.horario = 1; // 5:31 PM a 5:59 AM
}

// Calcular estación (0 o 1)
let mes = now.getMonth(); // 0 = Enero, 1 = Febrero, ..., 11 = Di
if (mes >= 5 && mes <= 8) { // Junio (5) a Septiembre (8)
    msg.payload.estacion = 0;
} else {
    msg.payload.estacion = 1; // Octubre (9) a Mayo (4)
}

// Preparar la entrada para el modelo
let datosModelo = {
    estacion: msg.payload.estacion,
    horario: msg.payload.horario,
    temperatura: msg.payload.temperatura,
    humedad: msg.payload.humedad,
    viento: msg.payload.velocidad,
    indice uv: msg.payload.uv
};

// Actualizar msg.payload con los datos listos para el modelo
msg.payload = datosModelo;

return msg;
```

Fuente: Autor

Este código realiza una consulta de la fecha y hora actual mediante la red y utiliza condicionales para clasificar los valores: Horario:

- De 6:00 AM a 5:30 PM, se asigna un valor de 0, que representa el día.
- De 5:31 PM a 5:59 AM, se asigna un valor de 1, que representa la noche/madrugada.

Estación del año:

- Si el mes actual está entre junio y septiembre, se asigna un valor de 0, que representa el verano.
- Si el mes está entre octubre y mayo, se asigna un valor de 1, que representa el invierno.

Finalmente, el código prepara los datos procesados, combinándolos con los valores recibidos desde los sensores por el puerto serial, y los envía al modelo de IA para su análisis y predicción.

3.4.9.3. Integración del árbol de decisiones a Node-Red

Existen diversas formas de integrar el modelo entrenado; sin embargo, se optó por utilizar un nodo de función, donde se ingresa el código mostrado en la Figura 57, esta forma de integración permite la optimización de recurso de la Rasbperry Pi. Este código realiza las siguientes tareas clave:

- Especifica la ruta del modelo entrenado y lo carga en memoria, permitiendo que esté listo para ser utilizado en predicciones.
- Prepara y acomoda los datos de entrada, asegurando que tengan el mismo formato y estructura que el modelo requiere para procesarlos correctamente.
- Realiza la predicción utilizando el modelo entrenado, generando la etiqueta correspondiente (por ejemplo, "Helada" o "No Helada").
- Envía la predicción al siguiente nodo para su procesamiento o visualización en el flujo de trabajo.

Este enfoque garantiza que el modelo sea integrado de manera eficiente, manteniendo la coherencia en los datos y la funcionalidad dentro del sistema.

Figura 57

Código de integración del árbol de decisión a Node-Red

```
import sys
import json
import joblib
import pandas as pd

# Cargar el modelo entrenado
modelo_path = '/home/alertahelada/modelo_heladas.pkl'  # Ruta del modelo
clf = joblib.load(modelo_path)

# Leer los datos de entrada
def main():
    try:
        # Leer los datos JSON desde la entrada estándar (stdin)
        datos = json.loads(sys.stdin.read())

        # Crear un DataFrame con nombres de las características
        entrada = pd.DataFrame([[
            datos['estacion'],
            datos['horario'],
            datos['temperatura'],
            datos['humedad'],
            datos['viento'],
            datos['indice_UV']
        ]], columns=['estacion', 'horario', 'temperatura', 'humedad', 'viento', 'indice_UV'])

        # Realizar la predicción
        prediccion = clf.predict(entrada)[0]
        resultado = "Helada" if prediccion == 1 else "No Helada"

        # Imprimir el resultado en formato JSON
        print(json.dumps({"prediccion": resultado}))

    except Exception as e:
```

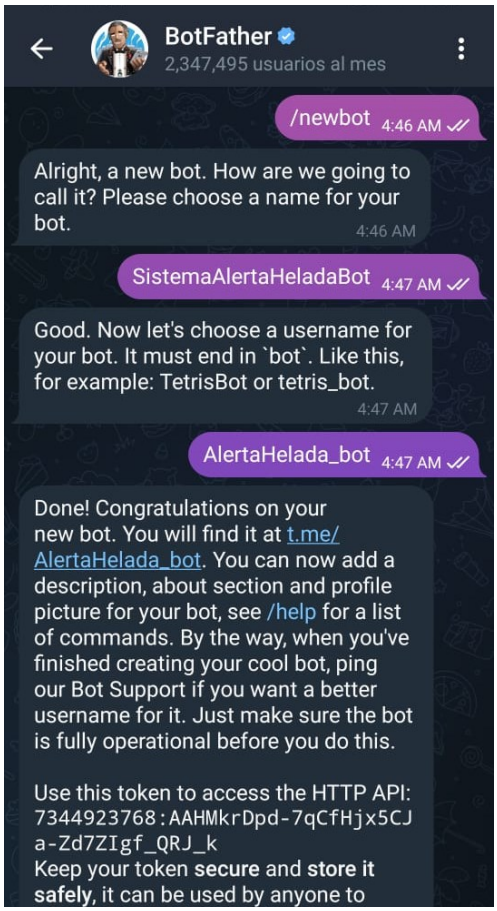
Fuente: Autor

3.4.9.4. Configuración de Alerta de Helada

Para la configuración de la alerta, se decidió utilizar Telegram, ya que cuenta con una librería compatible con Node-RED que facilita la integración. Antes de configurar Node-RED, es necesario crear un bot en Telegram y habilitar los permisos para recibir mensajes. El proceso comienza creando un bot con el nombre SistemaAlertaHeladaBot. Una vez creado, se solicita al bot un token de autenticación, que será utilizado para establecer la conexión con Node-RED. Este token, proporcionado por Telegram, se copia y guarda para usarlo en configuraciones posteriores, como se muestra en la Figura 58.

Figura 58

Creación del Bot de Telegram

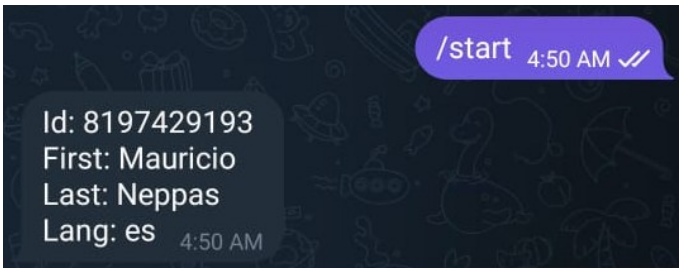


Fuente: Autor

Luego, se requiere el ID de usuario de Telegram, un identificador único que permite enviar notificaciones personalizadas a cada usuario. Para obtenerlo, se utiliza el bot @userinfobot, que proporciona este número al ser iniciados en Telegram como se observa en la Figura 59. Este ID debe registrarse en Node-RED para asegurar que las alertas se envíen al destinatario correcto, optimizando la efectividad del sistema de monitoreo ante posibles heladas.

Figura 59

ID Telegram



Fuente: Autor

A continuación, se configura un nodo de función donde se ingresa el ID de usuario obtenido en la Figura 59. En este nodo, se personaliza el mensaje de alerta a enviar, adaptándolo según las condiciones establecidas. A través de una estructura condicional, se define una salida específica, de modo que, si el modelo predice la ocurrencia de una helada, el mensaje se envía al siguiente nodo para su procesamiento y posterior notificación, este código se lo realiza mediante un nodo función obteniendo así la Figura 60.

Figura 60

Configuración ID Telegram en Node-Red

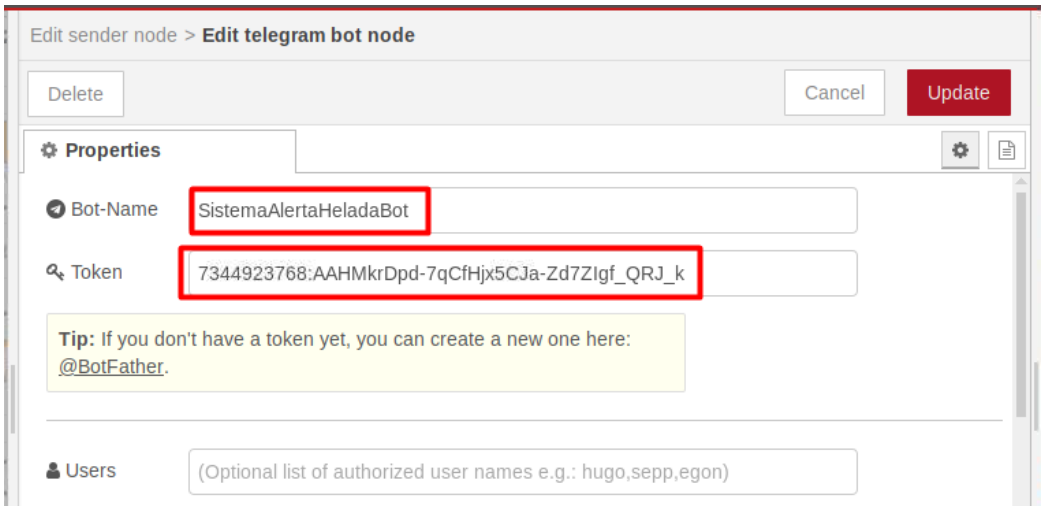
```
// Configuración del mensaje de Telegram
const chatId = "8197429193";
const alertMessage = "⚠️ Alerta de helada detectada en el cultivo. Tome precauciones.";

// Si la predicción es "Helada", envía un mensaje a Telegram
if (msg.payload.prediccion === "Helada") {
    return [
        {
            payload: {
                chatId: chatId,
                type: "message",
                content: alertMessage
            }
        },
        null
    ]; // Envía el mensaje a la primera salida
} else {
    return [null, msg]; // Envía el mensaje a la segunda salida
}
```

Fuente: Autor

Finalmente, en el nodo send Telegram, se configura el token obtenido en la Figura 61, permitiendo que servicios externos, como Node-RED, se conecten con Telegram y gestionen el envío automatizado de mensajes. Esta integración facilita la comunicación eficiente entre el sistema y la plataforma de mensajería.

Figura 61

Configuración Token en Node-Red



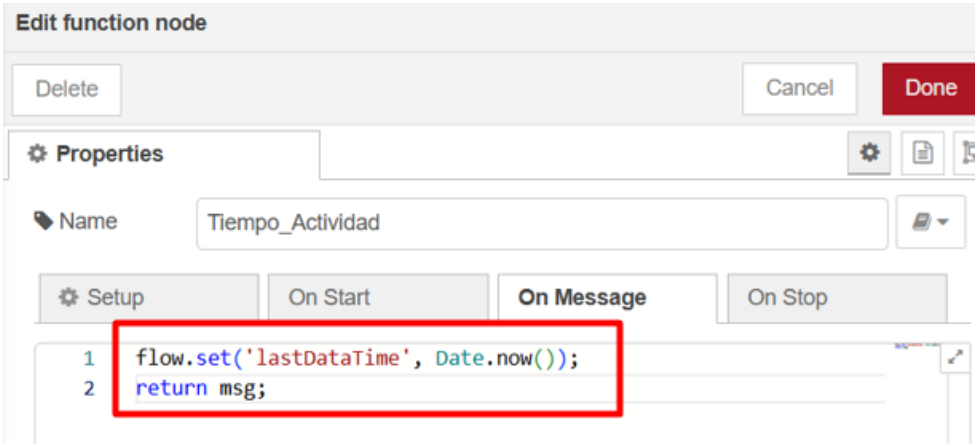
Fuente: Autor

3.4.9.5. Integración de alerta de Inactividad del nodo sensor

Uno de los requerimientos de modo de estado es que el nodo Gateway debe generar una alerta al no recibir datos desde el nodo sensor por lo cual se implementa un sistema para detectar la inactividad en la comunicación serial. Para ello empezamos configurando un nodo de función el cual guardara la hora del último dato recibido como se observa en la Figura 62. Este nodo se conecta directamente con el nodo serial in.

Figura 62

Guardar hora del último dato recibido

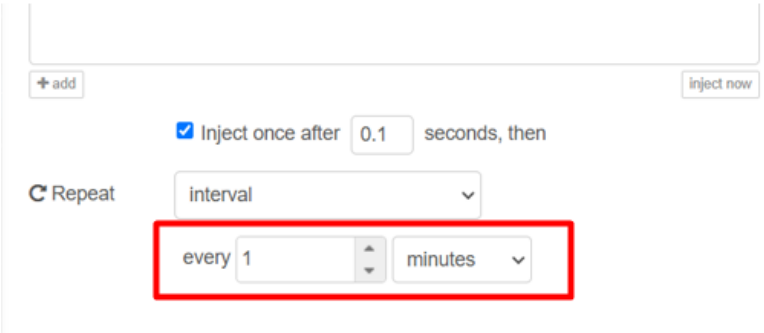


Fuente: Autor

Luego procedemos a configurar un nodo inject el cual permite generar un chequeo, este se configura según el tiempo que se requiere, en este caso como nuestros datos son enviados cada 30 segundos, configuramos este chequeo en 1 min para que no existan falsas alarmas. Esta configuración se observa en la Figura 63.

Figura 63

Generación de Chequeo de inactividad



Fuente: Autor

Luego para la verificación de inactividad se utiliza otro nodo de fuction en el cual se compara el último dato guardado con el tiempo actual teniendo la diferencia entre estos para

evaluar si sobrepasa el tiempo estimado de 1 min y generar la Alerta mediante Telegram. Todo el código de este nodo se aprecia en la Figura 64

Figura 64

Configuración de Alerta de Inactividad

Fuente: Autor

Para enviar a Telegram de igual manera que en la alerta de helada se utiliza un nodo sender de Telegram el cual se configura con el mismo bot y token que el anterior. Como resultado esperado al no recibir datos desde el nodo sensor se genera una alerta de inactividad, en la Figura 65 se observa el mensaje, el cual especifica que el nodo sensor ha dejado de funcionar y que necesita ser revisado.

Figura 65

Alerta de Inactividad de Telegram

Fuente: Autor

3.5. Etapa de visualización

La visualización de los datos recopilados por los sensores representa una fase esencial dentro del sistema de monitoreo climático, ya que proporciona a los agricultores acceso inmediato a información clave sobre las condiciones ambientales en tiempo real. Este proceso no solo permite un análisis más preciso de las variables climáticas, también facilita la actuación de estratégias para la gestión agrícola, optimizando el uso de recursos y mejorando la respuesta ante posibles cambios en el entorno.

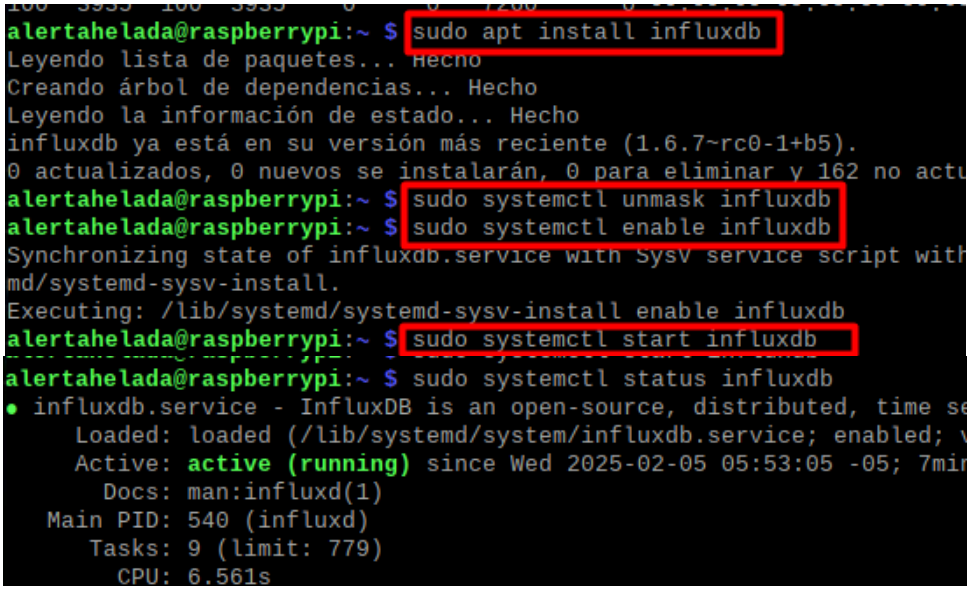
3.5.1. Envió de datos a InfluxDB

Para integrar el envío de datos a InfluxDB, se lleva a cabo la instalación del servicio en la Raspberry Pi junto con todos los complementos necesarios para su correcto funcionamiento. Una vez completada la instalación, se procede a activar el servicio y verificar el estado del servicio y la confirmación de que responde correctamente a las solicitudes. En la Figura 66 se

muestra un ejemplo de esta verificación, lo que asegura que la configuración está lista para recibir y gestionar los datos enviados.

Figura 66

Instalación y comprobación del servicio InfluxDB

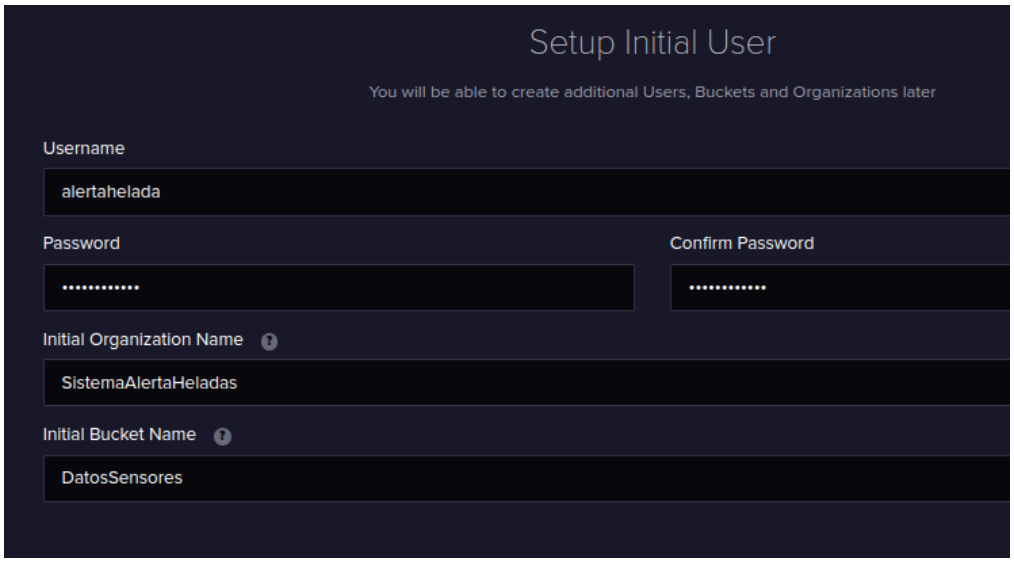


Fuente: Autor

Es importante destacar que, para garantizar una mejor compatibilidad con Node-RED, se utiliza la versión InfluxDB v2. Para acceder a la interfaz gráfica, se ingresa a través del navegador utilizando la dirección IP del servidor y el puerto 8086. En la Figura 67 se ilustran las configuraciones iniciales realizadas al acceder a la plataforma. Durante este proceso, se establece un usuario y contraseña, se crea una organización y se define un bucket para el almacenamiento de los datos.

Figura 67

Configuraciones de InfluxDB

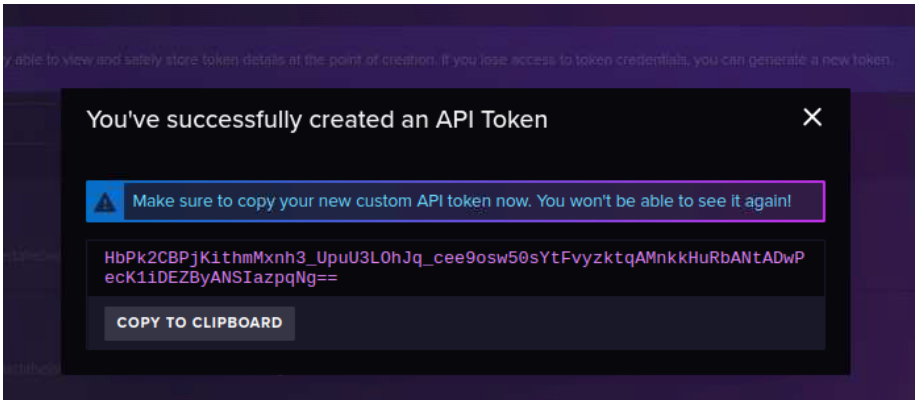


Fuente: Autor

Además, en pasos posteriores, como se muestra en la Figura 68 se genera un token para establecer la comunicación segura entre los servicios y el servidor de InfluxDB. Este token actúa como una credencial única que permite autenticar y autorizar el envío de datos desde diferentes aplicaciones, como Node-RED. Es fundamental copiar y guardar el token generado para utilizarlo posteriormente.

Figura 68

Generación del token InfluxDB



Fuente: Autor

En Node-RED, los datos se procesan y preparan teniendo en cuenta que deben ser enviados en formato de objeto, lo cual es fundamental para garantizar una comunicación estructurada y compatible la base de datos. En la Figura 69 se muestra un ejemplo de cómo se estructuran y envían estos datos

Figura 69

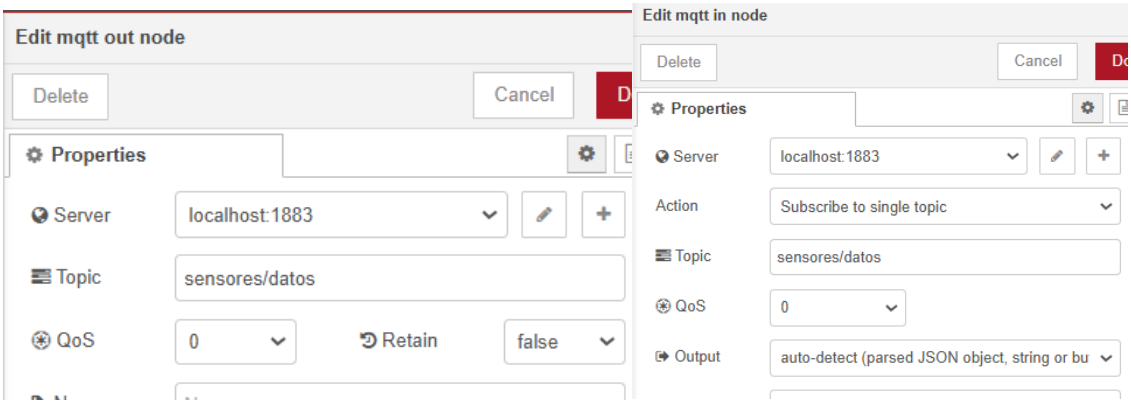
Formato de datos enviados a InfluxDB

Fuente: Autor

Para realizar el envío de los datos, se utiliza MQTT. Para ello, previamente se instaló el bróker Mosquitto, que actúa como intermediario en la comunicación. En Node-RED, se emplean los nodos mqtt out para enviar los datos y mqtt in para recibirlos y posteriormente subirlos a InfluxDB. Estos nodos se configuran con parámetros clave, como la dirección IP del servidor, el puerto (generalmente 8083) y el Topic correspondiente, asegurando así una comunicación eficiente y organizada entre los servicios, estas configuraciones están evidenciadas en la Figura 70.

Figura 70

Configuración de nodos MQTT

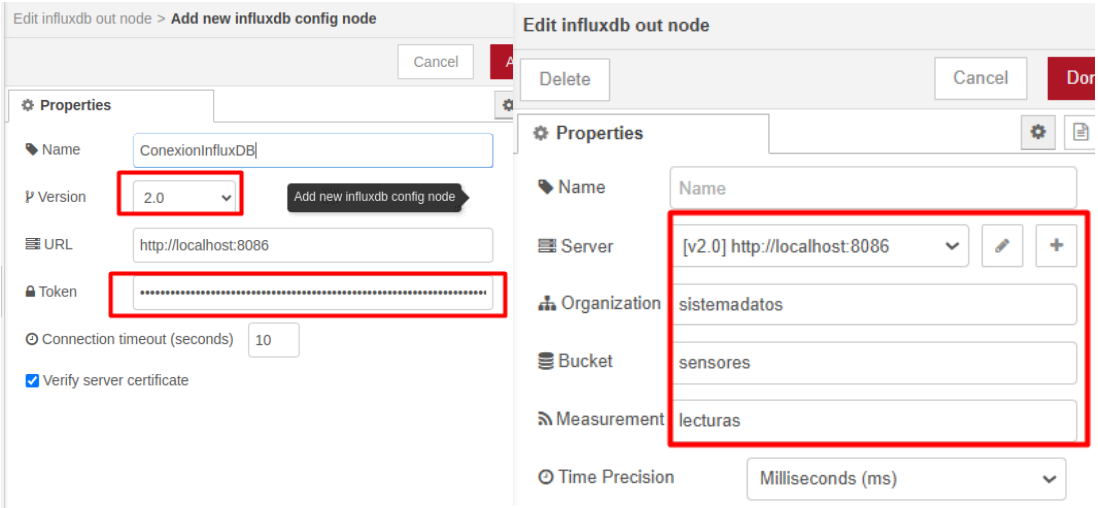


Fuente: Autor

Por último, se configura el nodo influxdb out dentro de Node-RED para establecer la conexión con el servidor. Durante esta configuración, se selecciona la versión de InfluxDB utilizada, se ingresa el token previamente generado y copiado, y se especifican los parámetros necesarios, como la dirección del servidor, el puerto 8086, la organización, el bucket y un topic correspondiente. Esta configuración garantiza que los datos se envíen correctamente al servidor InfluxDB, como se ilustra en la Figura 71.

Figura 71

Nodo InfluxDB

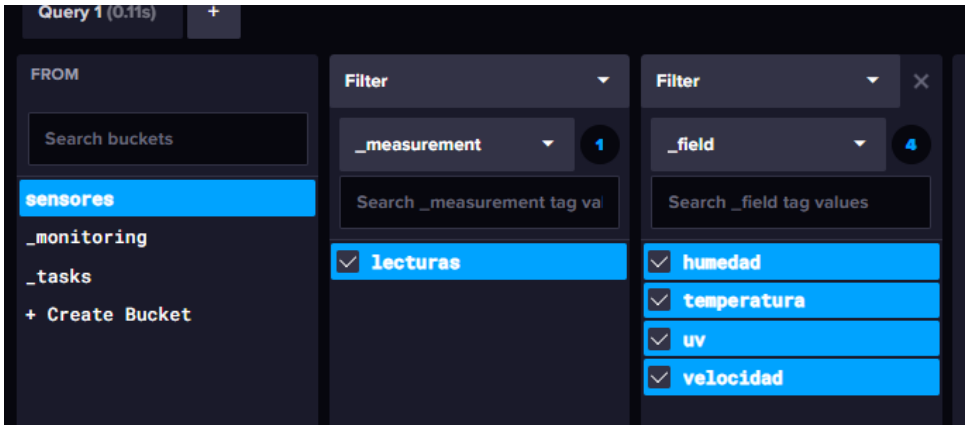


Fuente: Autor

Para verificar que los datos se envian correctamente, se accede a la interfaz gráfica de InfluxDB, en la Figura 72 donde se observa que las etiquetas se han generado automáticamente. Esto confirma que la conexión ha sido establecida con éxito y que los datos están siendo registrados en el sistema de manera adecuada.

Figura 72

Confirmación de conexión InfluxDB y Node-Red



Fuente: Autor

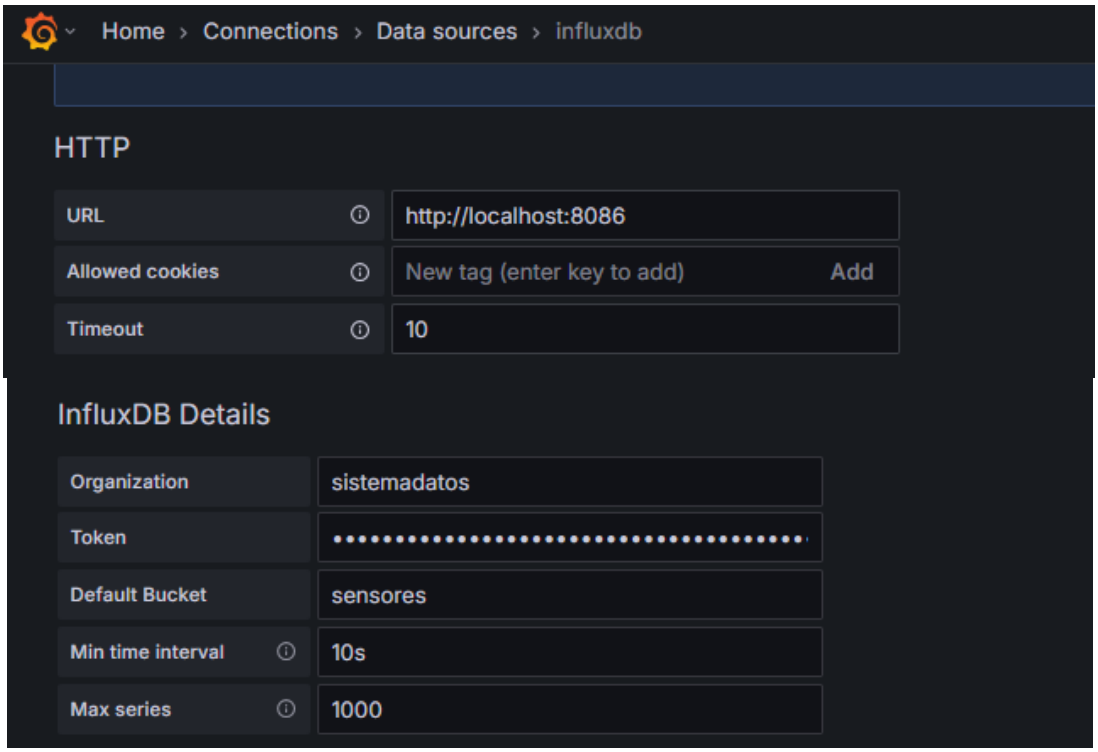
3.5.2. Visualización de Datos mediante Grafana

Al igual que con los servicios anteriores, se lleva a cabo la instalación y activación previa del servicio. Una vez completado este proceso, se accede a la interfaz gráfica utilizando la dirección IP del servidor y el puerto correspondiente, que en este caso es el 3000. Al ingresar por primera vez, se utiliza la autenticación predeterminada con el usuario admin y la contraseña admin, tras lo cual el sistema redirige automáticamente a la configuración de cambio de contraseña para reforzar la seguridad del acceso.

Inmediatamente, se procede a configurar la conexión con InfluxDB agregando una nueva fuente de datos. Para ello, se especifica la dirección y el puerto del servidor, se introduce el token de autenticación para establecer la conexión y se define el bucket al que se desea acceder. En la Figura 73 se muestra el proceso de configuración, asegurando que la integración con InfluxDB se realice de manera correcta y eficiente.

Figura 73

Configuración de conexión InfluxDB y Grafana



Fuente: Autor

A continuación, se configura un dashboard en el que se agregan diferentes gráficos según el tipo de datos a visualizar. Por ejemplo, para la temperatura y la humedad se utiliza el gráfico Gauge, ya que permite una interpretación rápida e intuitiva de los valores. Por otro lado, para la velocidad del viento y el índice UV se emplea el gráfico Stat, debido a su claridad al mostrar valores numéricos directos. Estas visualizaciones han sido elegidas con el objetivo de facilitar la interpretación de los datos.

Es importante destacar que, para acceder a la base de datos, se debe seleccionar InfluxDB como la fuente de datos. En la Figura 74 se muestra el resultado de estas configuraciones, así como el panel personalizado con los ajustes aplicados. Finalmente, se guarda el dashboard para conservar la configuración realizada.

Figura 74

Visualización Grafana

Fuente: Autor

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

El capítulo presenta el contenido de la cuarta fase de la metodología Action Research, la cual consiste en la fase de evaluación de los resultados de proyecto. Esta sección aborda la integración del hardware y software en un solo nodo respectivamente, que permita la comunicación de los nodos mediante la tecnología inalámbrica en conjunto con la publicación de los datos recopilados por estos, de modo que el análisis a realizar identifica el desempeño del sistema mediante pruebas de operación.

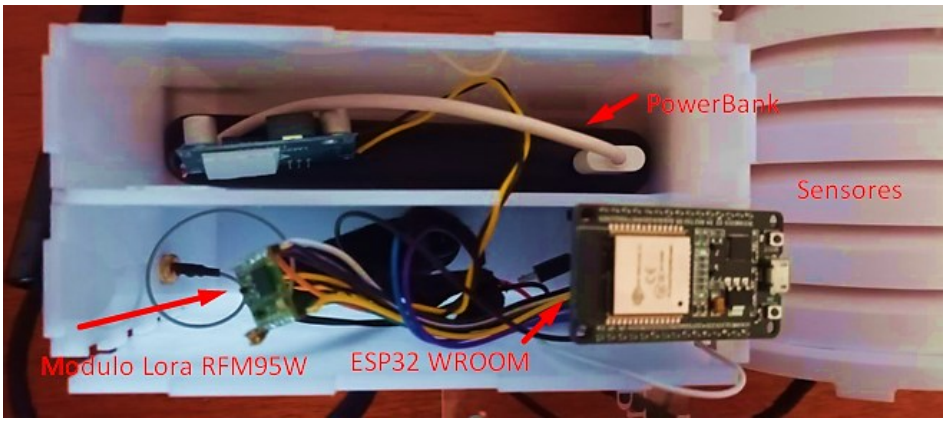
4.1. Integración de Componentes

La integración de los componentes corresponde a todas las conexiones físicas del hardware del nodo transmisor y receptor, de modo que se garantice su correcto funcionamiento dentro del sistema, Esta etapa implica la instalación, interconexión y validación de los dispositivos que conforman la infraestructura de la red de comunicación.

En el caso del nodo sensor, que desempeña la función de transmisor dentro de la red, la Figura 75 correspondiente ilustra sus conexiones principales. Destaca la presencia de una placa base ESP32 WROOM, la cual se encuentra integrada con tres sensores: DHT11, anemómetro y GUVA S12SD. Además, incorpora el módulo LoRa RFM95, encargado de facilitar la comunicación entre ambas estaciones, lo que conlleva a que todo el sistema sea alimentado por un power bank con una salida de 3V, asegurando un suministro de energía estable y continuo

Figura 75

Integración de componentes Nodo Sensor

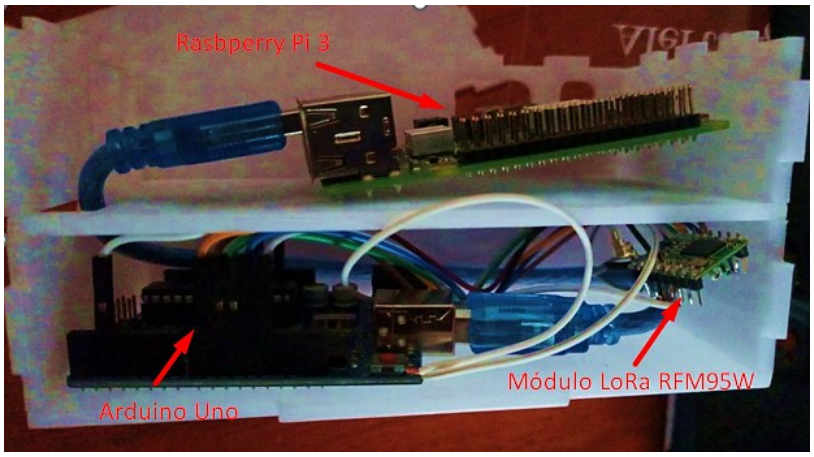


Fuente: Autor

Para el nodo Gateway, encargado de recibir los datos a través de la comunicación LoRa, en un inicio establece un enlace serial entre el microcontrolador Arduino UNO y la Raspberry Pi3. En este proceso, el módulo RFM95 recepta los datos transmitidos, que son recopilados mediante el Arduino UNO, para posteriormente enviarlos a la Raspberry Pi 3 para su procesamiento mediante Node-Red y posterior envío a InfluxBD. De modo que la implementación final del nodo Gateway LoRa es representado en la Figura 76.

Figura 76

Integración de componentes Nodo Gateway



Fuente: Autor

Una vez la implementación de los nodos ha sido completada respecto a su interconexión, ubicación y configuración dentro del prototipo, como se muestra en la Figura 77, el resultado del hardware obtenido corresponde a un nodo encargado de medir niveles de temperatura/humedad relativa mediante el sensor DHT11, la velocidad del viento en el anemómetro y la intensidad de radiación ultravioleta a través del sensor GUVA S12SD, para utilizar la comunicación LoRa y enviarlos al nodo receptor para su posterior procesamiento y publicación.

Figura 77

Nodos Integrados

Fuente: Autor

4.2. Verificación de configuración de los nodos

La verificación de la configuración en cada uno de los componentes del sistema es un paso fundamental para garantizar el funcionamiento de la red. Este proceso permite comprobar que la configuración se ha realizado exitosamente y que cada nodo opera según los parámetros establecidos. En particular, es crucial confirmar que la comunicación inalámbrica LoRa se inicia correctamente tanto en el nodo de transmisión como en el nodo Gateway.

Además, se verifican aspectos clave como la sincronización de los dispositivos. Este proceso de verificación no solo confirma el correcto establecimiento de la comunicación, sino

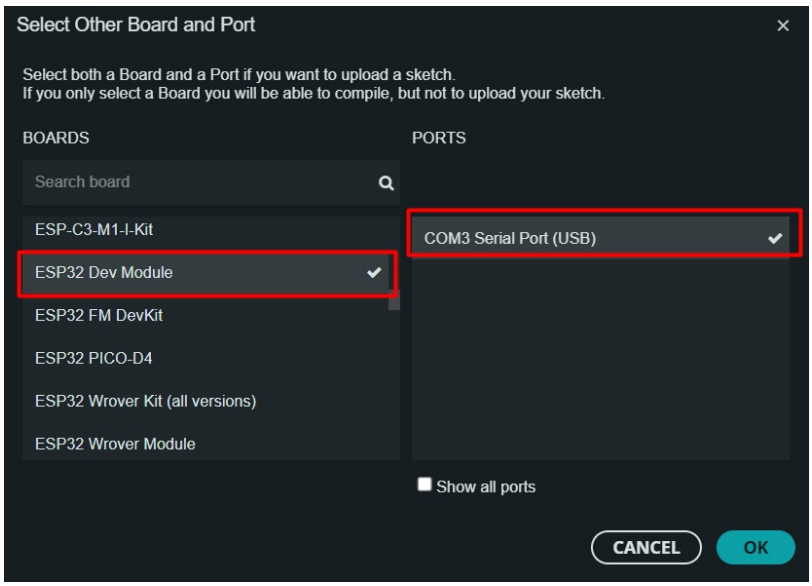
que también permite anticipar posibles problemas operativos, asegurando la fiabilidad del sistema antes de su despliegue en condiciones reales.

4.2.1. Verificación Nodo sensor

La programación del nodo sensor, basado en la placa ESP32, se lleva a cabo mediante el entorno de desarrollo Arduino IDE. En este dispositivo se integran el módulo LoRa RFM95 junto con los sensores DHT11, anemómetro y GUVA S12SD. Para asegurar su correcto funcionamiento, el código es escrito y transferido desde una computadora a la placa, estableciendo la conexión a través del puerto serial, identificado como "COM" dentro del entorno de desarrollo. Durante este procedimiento, se configuran parámetros clave, como la selección de la placa, la asignación del puerto de comunicación y la velocidad de transmisión de datos como se observa en la Figura 78, con el fin de garantizar una comunicación estable y eficiente en la red de sensores.

Figura 78

Selección de placa y puerto ESP32 WROOM



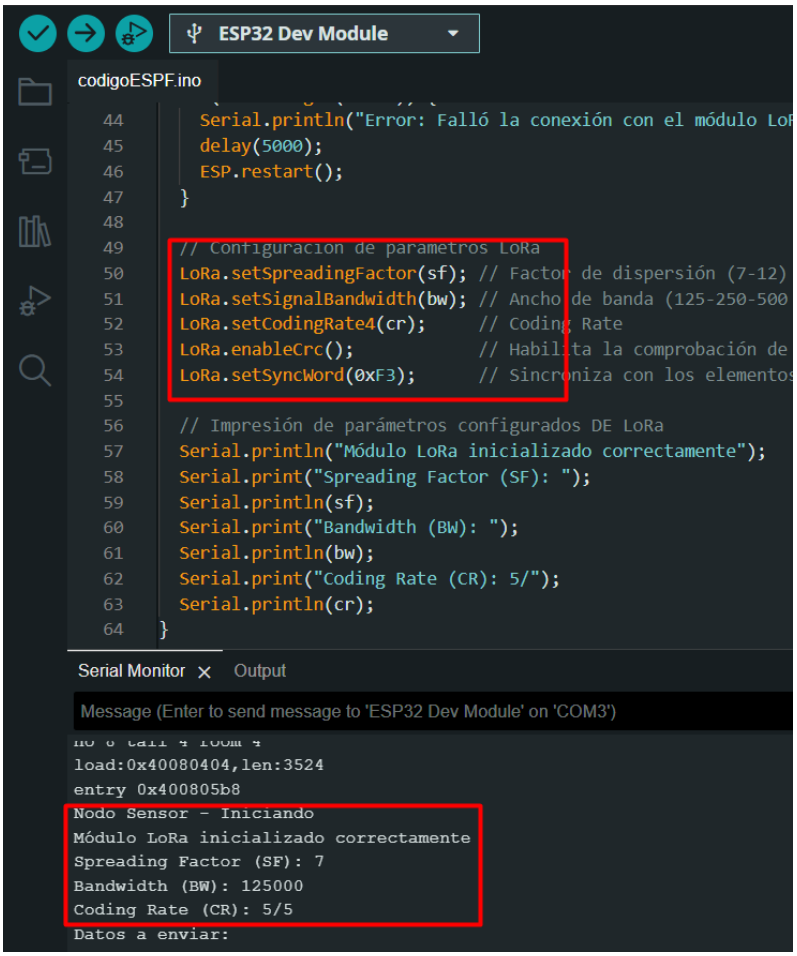
Fuente: Autor

Utilizando el monitor serial, se documenta el proceso de inicio del nodo sensor, y la inicialización del módulo LoRa. En este procedimiento, se define una frecuencia de operación

de 915 MHz, un factor de propagación (SF) de 7, un ancho de banda (BW) de 125 kHz y una tasa de corrección de errores (CR) de 4/5. Estos ajustes aseguran una transmisión eficiente y confiable de los datos, como se ilustra en la Figura 79, Comprobando así que la Placa ESP32 está configurada correctamente.

Figura 79

Inicialización LoRa Nodo Sensor



Fuente: Autor

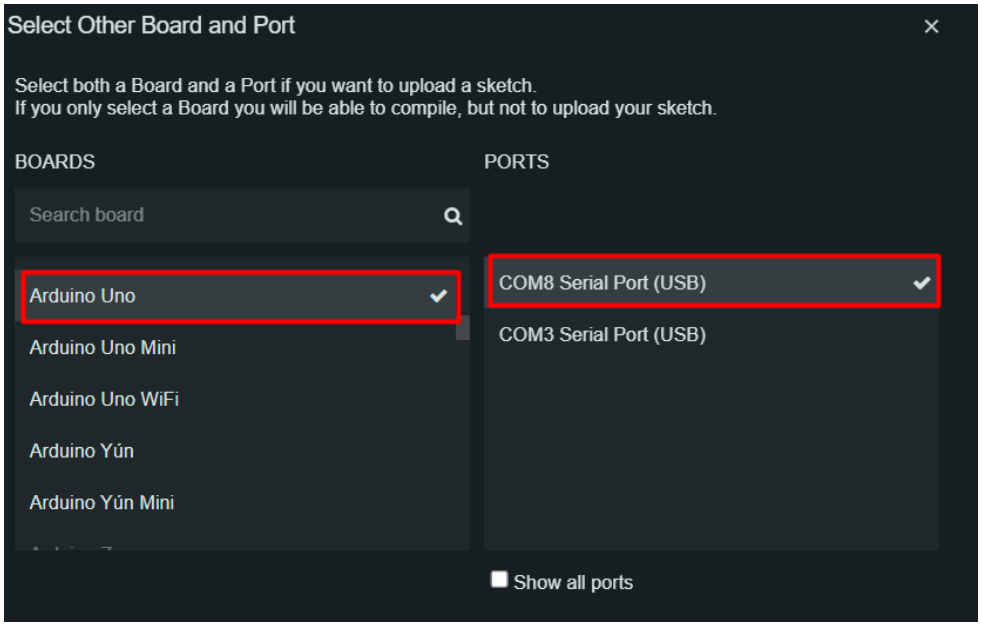
4.2.2. Verificación Nodo Gateway

La configuración del módulo LoRa en el Gateway inicia con la elección de la placa y la identificación del puerto COM asociado a la placa Arduino, que cumple la función de receptor. Como se muestra en la Figura 80 este procedimiento se lleva a cabo desde una

computadora, estableciendo la comunicación con la placa mediante el puerto, lo que facilita la adecuada recepción de los datos transmitidos.

Figura 80

Selección de placa y puerto Arduino

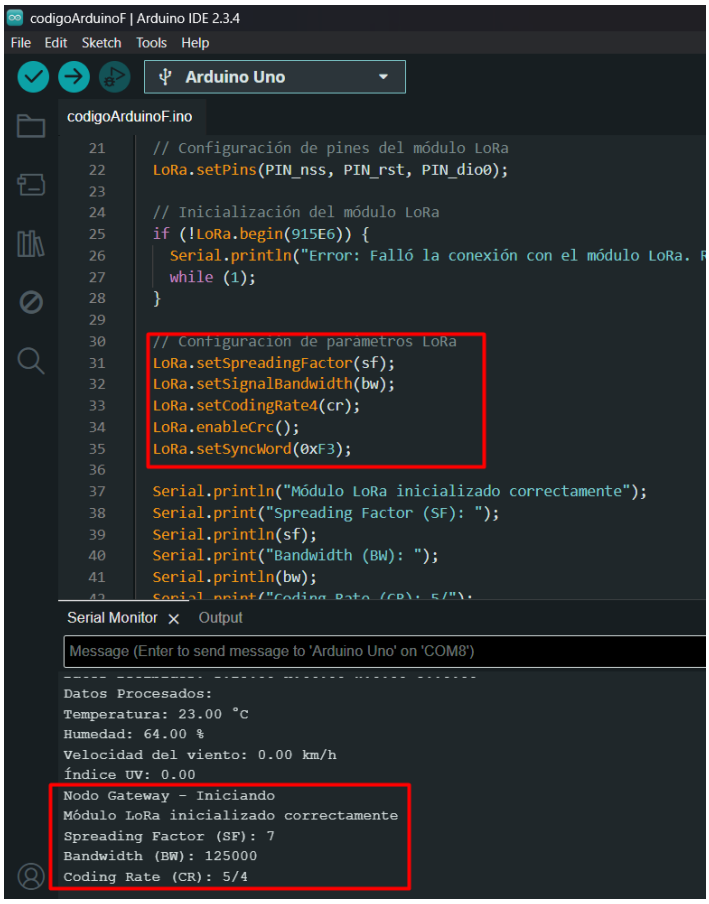


Fuente: Autor

De manera similar, el proceso avanza mediante el monitor serial, donde se registra la activación del módulo LoRa. En la Figura 81 se presenta la visualización de los parámetros configurados para la comunicación del módulo receptor. Los parametros de configuracion son identigos al nodo sensor asegurando una transmisión eficiente de los datos.

Figura 81

Inicialización LoRa Nodo Gateway



Fuente: Autor

4.3. Pruebas de funcionamiento de los nodos

Las pruebas de funcionamiento del sistema son fundamentales, ya que permiten verificar el desempeño de cada componente. Se inicia con la recolección de datos provenientes de los sensores DHT11, anemómetro y GUVA S12SD, asegurando su correcto funcionamiento. Además, se comprueba que el módulo LoRa RFM95W opere con los parámetros configurados. Asimismo, se verifica la recepción de datos por parte del Gateway, junto con el procedimiento de conexión a la base de datos y su correcta visualización.

4.3.1. Implementación del sistema

La implementación del sistema se lleva a cabo en la parroquia rural Olmedo – Pesillo, perteneciente al cantón Cayambe. En este sector, el nodo sensor se instalará en un cultivo de

papas, mientras que el Gateway se ubicará en la casa del agricultor, permitiendo así la activación del sistema de alerta contra heladas. Para la correcta colocación de los nodos, se consideran factores clave como la orientación de las antenas, la altura óptima de instalación y las conexiones necesarias para la alimentación del sistema.

La instalación del nodo sensor se realiza en el cultivo seleccionado, asegurando una línea de vista directa hacia el nodo Gateway o, en su defecto, una trayectoria con mínima interferencia para garantizar una comunicación eficiente, que no se obstruya el funcionamiento del sensor anemómetro y GUVA S12SD. Para su fijación, se construye una estructura de madera que se ancla firmemente en el suelo, permitiendo estabilidad y resistencia a las condiciones ambientales, como se muestra en la Figura 82.

Figura 82

Implementación nodo sensor

Fuente: Autor

Por otro lado, el nodo Gateway se instala a una distancia aproximada de 0.55 km del nodo sensor, en la casa del agricultor. Su ubicación estratégica permite una conexión estable tanto con la red de internet como con la fuente de alimentación, asegurando su funcionamiento continuo y eficiente. Además, se verifica que su antena esté correctamente orientada para

optimizar la recepción de los datos enviados por el nodo sensor, como se ilustra en la Figura 83

Figura 83

Implementación nodo Gateway

Fuente: Autor

4.3.2. Recolección y verificación de datos del nodo Sensor

Para verificar que el nodo sensor está funcionando correctamente y realizando lecturas precisas a través de los sensores integrados, se utiliza el monitor serial del entorno de desarrollo Arduino IDE. Esta herramienta permite observar en tiempo real los valores capturados por el DHT11, el anemómetro y el sensor GUVA S12SD, lo que facilita la detección de posibles errores en las mediciones y asegura que cada sensor opere dentro de los parámetros esperados.

Figura 84

Verificación de datos nodo Sensor

Fuente: Autor

Además, se verifica que el módulo LoRa RFM95W esté transmitiendo la información sin interrupciones. Como se muestra en la Figura 84, el envío de datos se realiza de manera exitosa, lo que confirma la correcta integración entre los sensores, el microcontrolador y el módulo de comunicación. Este proceso es fundamental para garantizar que el sistema funcione de manera eficiente antes de su implementación en campo.

4.3.3. Recolección y verificación de datos del nodo Gateway

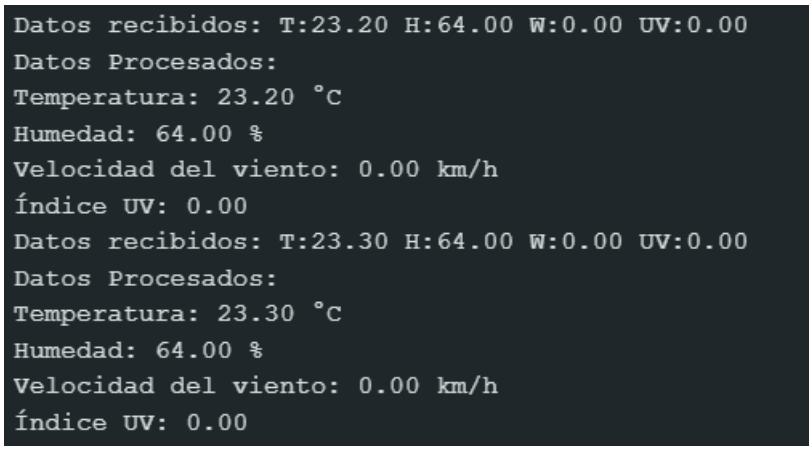
Como se confirmó en la sección anterior que los datos están siendo enviados correctamente desde el nodo sensor, el siguiente paso es verificar la recepción de estos datos en el Gateway. Para ello, se lleva a cabo un proceso de validación en cada uno de los componentes y plataformas que conforman el nodo Gateway, asegurando su correcto funcionamiento**.**

4.3.3.1. Recepción y verificación en Arduino Uno

En primer lugar, el componente inicial en el flujo del Gateway es el Arduino, el cual tiene la función de establecer la comunicación LoRa y recibir los paquetes de datos enviados por el nodo sensor. Una vez que los datos son capturados, el Arduino los procesa y los prepara para su transmisión hacia la Raspberry Pi a través de la comunicación serial.

Figura 85

Verificación de datos en Arduino Uno



Fuente: Autor

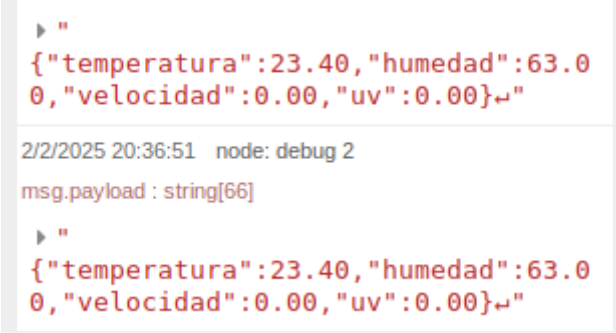
Como se muestra en la Figura 85, se verifica que los datos están siendo correctamente recibidos y procesados, asegurando así la integridad de la información antes de su posterior envío a la base de datos y visualización en la plataforma correspondiente. Esta etapa es crucial, ya que garantiza que la comunicación entre el nodo sensor y el Gateway funcione sin interrupciones.

4.3.3.2. Recepción y verificación en Rasbperry

Los datos son recibidos a través del puerto serial, y para su visualización y verificación se emplea la plataforma Node-RED, una herramienta que facilita la integración de dispositivos y la monitorización de datos en tiempo real. Dentro de esta plataforma, se utiliza el nodo Serial In, encargado de capturar los datos que llegan a través del puerto serial, y se conecta a un nodo Debug, que permite visualizar la información en la consola de depuración de Node-RED.

Figura 86

Recepción y verificación de datos en Rasbperry



Fuente: Autor

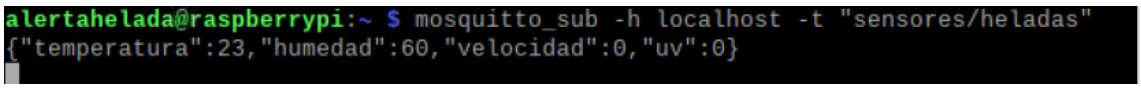
Además, facilita la identificación de posibles errores en la comunicación, como pérdida de paquetes o inconsistencias en los datos transmitidos. Como se muestra en la Figura 86, se comprueba que los datos se estén recibiendo de manera eficiente sin pérdida de datos, es importante recalcar que los datos recibidos está en un formato JSON String.

4.3.3.3. Captura de paquetes enviados por MQTT

Una vez verificada la recepción de los datos en la Raspberry Pi, se procede a validar la conexión con el bróker MQTT, asegurando la correcta transmisión de la información. Posteriormente, se comprueba que los datos sean almacenados adecuadamente en la base de datos InfluxDB. Finalmente, se revisa la interfaz de visualización en Node-RED para confirmar que los valores se actualizan en tiempo real, lo que certifica la integración exitosa del sistema y su correcto funcionamiento. Para ello verificamos que se esté enviando los datos mediante nuestro bróker mosquitto como se muestra en la Figura 87.

Figura 87

Verificación del envío de datos en le Broker Mosquitto

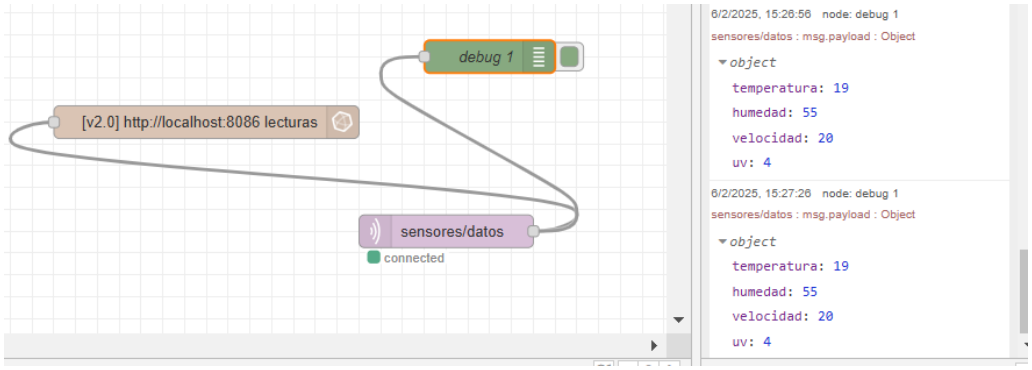


Fuente: Autor

Del mismo modo, es posible verificar el envío de estos datos a través de los nodos MQTT en Node-RED visualizado en la Figura 88, asegurando su correcta recepción y procesamiento en el sistema.

Figura 88

Verificación recepción de Datos nodo MQTT Node-Red

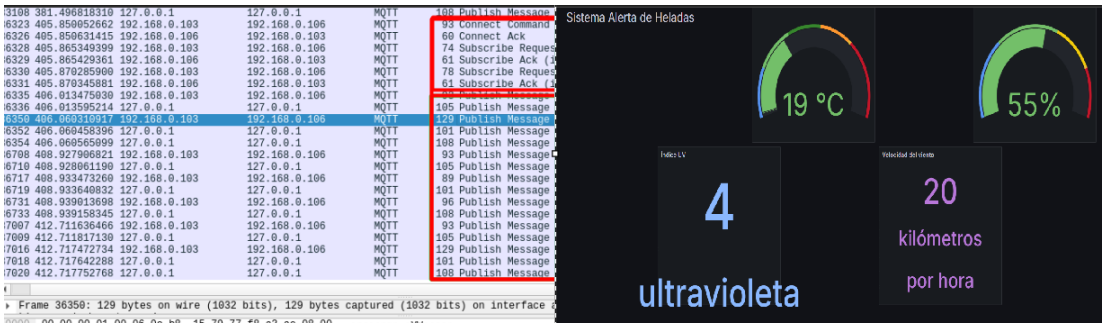


Fuente: Autor

A través de Wireshark, es posible analizar y verificar que los paquetes de datos están siendo transmitidos correctamente mediante el protocolo MQTT. Este análisis permite inspeccionar en detalle la estructura de los paquetes, asegurando que la comunicación entre los dispositivos y el bróker MQTT se lleva a cabo de manera eficiente y sin pérdida de información. Como se explicó anteriormente, los datos generados por los sensores son enviados mediante MQTT hacia la base de datos, donde se almacenan para su posterior procesamiento y visualización. En la Figura 89, se puede observar el flujo de datos desde el nodo sensor hasta la base de datos, lo que confirma la correcta integración y funcionamiento del sistema. Además, este monitoreo en tiempo real facilita la detección de posibles errores en la transmisión, permitiendo realizar ajustes en la configuración si es necesario para optimizar la estabilidad y eficiencia del sistema.

Figura 89

Captura de Wireshark y visualización



Fuente: Autor

4.3.3.4. Recepción y verificación en Influxdb

Como se verificó en el paso anterior, el envío de datos a través de MQTT se realizó correctamente. Ahora, el siguiente paso es comprobar que estos datos están siendo almacenados en la base de datos de manera adecuada. Para ello, es fundamental considerar que la información transmitida mediante MQTT se encuentra en formato JSON, como se muestra en la Figura 90.

Este detalle es crucial, ya que permite asegurar que los datos sean interpretados correctamente y almacenados como valores numéricos, evitando errores en su procesamiento.

Figura 90

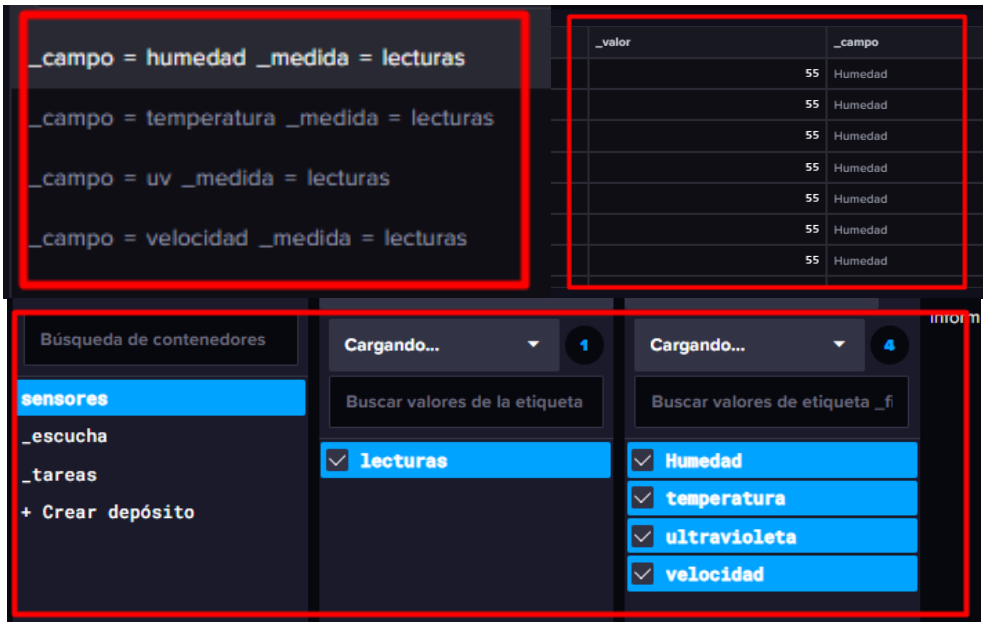
Formato de envío de datos MQTT

Fuente: Autor

Dentro de la plataforma InfluxDB, se puede verificar que las etiquetas han sido creadas correctamente y que los datos están siendo almacenados de manera adecuada. Para facilitar su visualización, se utiliza una tabla simple, la cual permite inspeccionar los valores registrados en tiempo real. Como se muestra en la Figura 91, esta representación gráfica proporciona una manera clara y estructurada de analizar la información almacenada en la base de datos.

Figura 91

Verificación de Datos en Influx DB



Fuente: Autor

4.3.3.5. Recepción y verificación en Grafana

Dado que existe una conexión directa entre la base de datos y la plataforma Grafana, los datos almacenados en InfluxDB se pueden visualizar en tiempo real mediante paneles de monitoreo personalizados. La base de datos ha sido correctamente configurada como fuente de datos en Grafana, permitiendo que la información se actualice de forma automática cada 5 segundos.

Figura 92

Interfaz de Grafana con los datos en tiempo real

Fuente: Autor

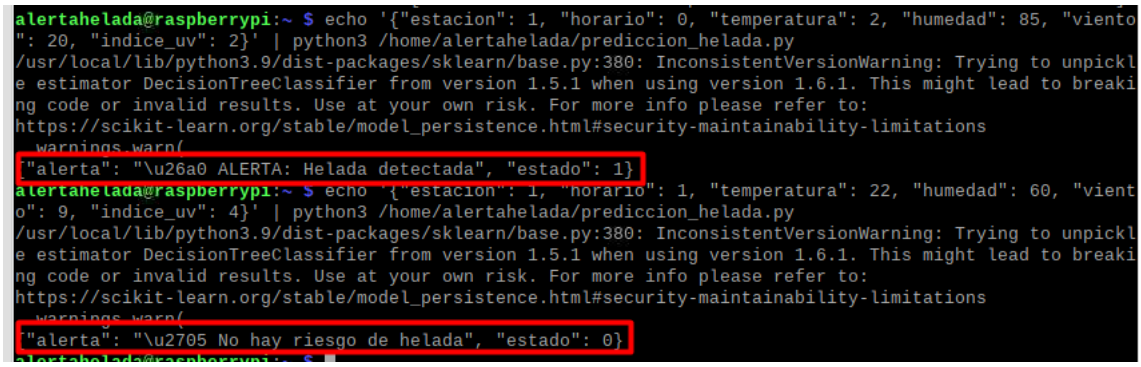
Este proceso garantiza una representación dinámica y precisa de los valores recopilados por los sensores, lo que facilita el análisis y la detección de patrones o anomalías. Además, Grafana permite personalizar las visualizaciones mediante gráficos optimizando así la interpretación de los datos. En la Figura 92, se muestra la interfaz de Grafana con los datos en tiempo real, confirmando que la integración con la base de datos se ha realizado correctamente y que la información se actualiza de manera continua según la configuración establecida.

4.4. Verificación de funcionamiento de la IA

Para verificar que la IA ha sido entrenada correctamente, se realiza una prueba ingresando datos manualmente mediante Python y ejecutándolo. Como resultado, se generan los datos presentados en la Figura 93, donde se puede observar que el árbol de decisiones efectúa una predicción acorde con las variables ingresadas. Esta predicción es validada comparándola con el comportamiento esperado, lo que confirma que el modelo está funcionando correctamente. Se realizan pruebas con diferentes datos simulando ambas predicciones.

Figura 93

Comprobación del funcionamiento de la IA mediante Python

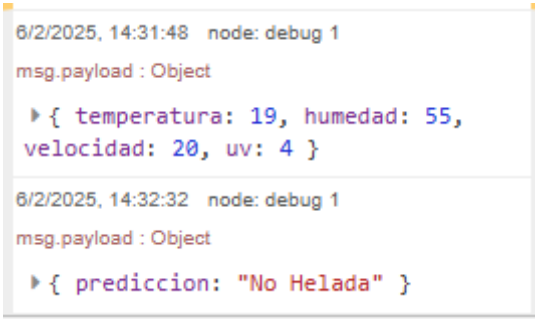


Fuente: Autor

Además, se lleva a cabo una segunda verificación integrando el árbol de decisiones con Node-RED, utilizando datos obtenidos en tiempo real gracias a la lectura de los sensores. En este escenario, el sistema analiza las variables de temperatura y humedad ambiental, velocidad del viento e índice UV para determinar la posible ocurrencia de una helada. En la Figura 94 se muestra que el modelo realiza una predicción precisa, indicando que no se espera una helada dadas las condiciones actuales.

Figura 94

Comprobación de funcionamiento de la IA mediante Node-Red Predicción 1



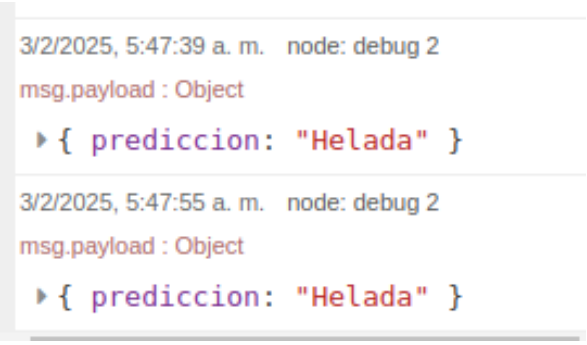
Fuente: Autor

Además, en la Figura 95 se puede observar la predicción de la presencia de helada, ya que los datos obtenidos indicaban condiciones propicias para dicha predicción. Esto refuerza

la efectividad del árbol de decisiones, tanto en pruebas manuales como en su integración con datos en tiempo real, asegurando su fiabilidad para la toma de decisiones automatizada.

Figura 95

Comprobación de funcionamiento de la IA mediante Node-Red Predicción 2



Fuente: Autor

4.5. Comprobación de envió de Alerta en Telegram

Las alertas a través de Telegram se activan únicamente cuando la predicción del árbol de decisiones indica la presencia de una helada, garantizando así que solo se generen notificaciones en casos relevantes. Para ello, se utiliza un Bot automatizado que envía un mensaje predefinido al usuario. El mensaje de alerta como se observa en la Figura 96 incluye un ícono distintivo de advertencia para captar la atención de inmediato y notificar sobre la detección de una helada. Además, se proporciona una breve descripción del evento, indicando la necesidad de tomar precauciones ante las condiciones climáticas adversas.

Figura 96

Notificaciones de Alerta Telegram

Fuente: Autor

Este sistema de notificación en tiempo real permite a los usuarios reaccionar de manera oportuna, implementando medidas preventivas para minimizar los efectos negativos de la helada en los cultivos o infraestructura. Gracias a la integración con Telegram, las alertas pueden ser recibidas en cualquier dispositivo con la aplicación instalada, facilitando la gestión de riesgos en cualquier momento y lugar.

4.6. Condiciones del Sistema para que se origine la Alerta del sistema

Según nuestro algoritmo entrenado existen valores críticos que podrían activar una alerta, como se muestra en la Tabla 23, estos valores representan los valores críticos en cada una de las variables consideradas, los valores son independientes por lo cual podrían generar falsos negativos, pero para evitar estos estas variables se deben relacionar unas con otras para poder activar una alerta lo cual se indica en la Tabla 24

Tabla 23:

Valores picos independientes

Varible	Condición crítica
Temperatura	<= 9.5 °C
Viento	<= 25 Km/h
Huemdad	<=88 %
Indice UV	<= 3.5
Estación	= 1

Fuente: Autor

Tabla 24

Valores picos dependientes

Variable	Condición crítica	Condición de dependencia
Temperatura	<= 9.5 °C	Con humedad > 88.0 %
Viento	<= 25 Km/h	Con temperatura > 9.5 °C
Humedad	<=88 %	Si el viento > 59.5 km/h
Índice UV	<= 3.5	Si el viento > 59.5 km/h
Estación	= 1	En condiciones de alta humedad y bajo viento

Fuente: Autor

Dentro de nuestro árbol de decisiones los datos se han dividido tomando en cuenta la manera jerarquía o priorizando algunas variables más que otras, esto se debe que a que etas variables están presente en la mayoría de los casos de helada, por ello en la Tabla 25 se detalla los posibles picos o ramificaciones que se generaría una alerta.

Tabla 25

Valores picos dependientes priorizando Temperatura

Condición 1	Condición 2	Condición 3	Condición 4
Temperatura ≤ 9.5°C	Viento ≤ 25 km/h	No existe	No existe
Temperatura ≤ 9.5°C	Viento ≤ 25 km/h	Humedad ≤ 88%	No existe
Temperatura ≤ 9.5°C	Viento ≤ 25 km/h	Humedad ≤ 88%	Temp ≤ 7.5°C
Temperatura ≤ 9.5°C	Viento ≤ 25 km/h	Humedad ≤ 88%	Estación ≤ 1
Temperatura ≤ 9.5°C	Viento ≤ 59.5 km/h	No existe	Índice UV ≤ 3.5

Fuente: Autor

También existen otras condiciones que sin priorizar la temperatura se puede notar la presencia de helada, estas se reflejan en la Tabla 26.

Tabla 26

Valores picos dependientes sin priorizar Temperatura

Condición 1	Condición 2	Condición 3
Viento ≤ 25 km/h	Humedad ≤ 88%	No existe
Viento ≤ 25 km/h	Humedad ≤ 88%	Estación ≤ 0.5
Viento > 59.5 km/h	No existe	Índice UV ≤ 3.5

Fuente: Autor

4.7. Análisis del Registro de Datos

El sistema entra en funcionamiento durante todo el período en el que el cultivo se encuentra en el terreno, abarcando desde la fase de germinación hasta la cosecha. Durante este tiempo, se monitorean continuamente las variables ambientales claves para detectar posibles condiciones de helada.

Según los datos presentados en la Tabla 27, durante el día no se registraron valores que indicaran la presencia de una helada. Las mediciones de temperatura, humedad, velocidad del

viento e índice UV se mantuvieron dentro de rangos estables, lo que confirma que, en ese período, las condiciones climáticas fueron favorables para el cultivo y no representaron riesgos de congelación.

Tabla 27

Datos obtenidos en el día.

Fecha y Hora	Temperatura (°C)	Humedad (%)	V. del Viento (Km/h)	Índice UV.
2024/07/17 09:15:15	13	65	35	1
2024/07/17 10:15:15	14	65	0	2
2024/07/17 11:15:15	17	53	0	5
2024/07/17 12:15:15	18	51	49	9
2024/07/17 13:15:15	17	52	30	7
2024/07/17 14:15:15	15	66	0	3
2024/07/17 15:15:15	13	65	51	4
2024/07/17 16:15:15	14	63	0	4
2024/07/17 17:15:15	12	50	0	2

Fuente: Autor

Los valores de la Tabla 27 presentan una muestra representativa de todos los datos recopilados a lo largo del día. Dado que las variaciones en las mediciones no fueron significativas, se optó por realizar un muestreo por horas, permitiendo así una mejor visualización de las tendencias sin perder precisión en el análisis de las condiciones climáticas.

Como se tiene conocimiento, la formación de heladas suele ocurrir durante la noche o madrugada, por lo que las lecturas de datos se realizan principalmente en este período. Sin embargo, es importante recalcar que, al tratarse de un fenómeno natural influenciado por

diversos factores climáticos, su ocurrencia no está estrictamente limitada a estas horas, pudiendo presentarse en cualquier momento del día.

Tabla 28

Datos obtenidos Noche/Madrugada

Fecha y Hora	Temperatura (°C)	Humedad (%)	V. del Viento (Km/h)	Índice UV.
2024/07/16 18:15:15	10	85	29	0
2024/07/16 19:15:15	10	81	0	0
2024/07/16 20:15:15	12	82	8	0
2024/07/16 21:15:15	11	85	30	0
2024/07/16 22:15:15	11	82	30	0
2024/07/16 23:15:15	9	60	0	0
2024/07/17 24:15:15	9	54	20	0
2024/07/17 01:15:15	9	58	0	0
2024/07/17 02:15:15	9	62	0	0
2024/07/17 03:15:15	7	40	19	0
2024/07/17 04:15:15	7	34	0	0
2024/07/17 05:15:15	9	50	0	0
2024/07/17 05:30:00	9	60	28	0

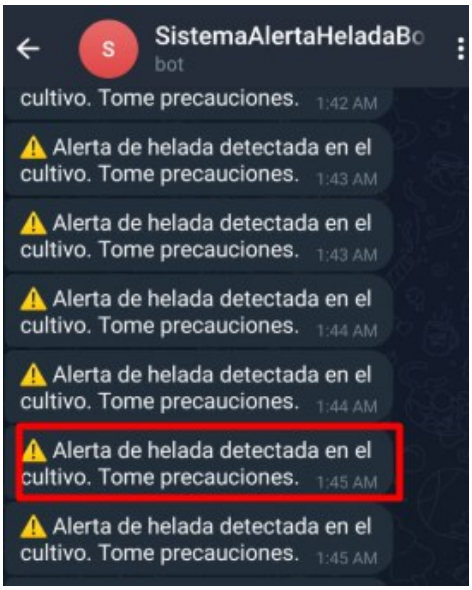
Fuente: Autor

En la Tabla 28, se presentan los datos recopilados en la Noche/Madrugada, con un enfoque especial en los registros que muestran condiciones favorables para la formación de una helada. Este análisis permite identificar patrones de variación en temperatura, humedad y velocidad del viento, facilitando la detección de posibles eventos de heladas y su impacto en los cultivos.

Como se puede observar, la temperatura es uno de los factores clave en la formación de heladas. Durante la madrugada, las temperaturas tienden a descender significativamente, y cuando se combinan con otras variables climáticas, pueden generar condiciones propicias para la ocurrencia de este fenómeno. En estos casos, el sistema activa una alerta de posible helada, permitiendo tomar medidas preventivas. Esto se refleja claramente en la Figura 97, donde se evidencian los registros de la alerta emitida.

Figura 97

Alerta emitida en la madrugada



Fuente: Autor

Otra variable importante que considerar es la disminución significativa de la temperatura durante el invierno, lo que sugiere que el sistema emitirá alertas con mayor frecuencia durante los meses que abarcan esta estación. Este comportamiento es previsible debido a las condiciones climáticas adversas propias del invierno, que aumentan la probabilidad de heladas y, por lo tanto, la necesidad de monitoreo continuo.

4.8. Evaluación de la disminución de pérdidas en cultivos.

El fenómeno de la helada causa daños significativos en los cultivos, afectando directamente la producción estimada por el agricultor y reduciendo las ganancias proyectadas. En los escenarios más extremos, una helada severa puede provocar la pérdida total del cultivo, resultando en ganancias nulas ($0) para el agricultor. Sin embargo, incluso en casos menos drásticos, donde solo el 50% de la producción se ve afectada, las pérdidas económicas pueden ser sustanciales y variar según el tipo de cultivo y su valor en el mercado.

La implementación del Sistema de Alerta contra Heladas reduce significativamente el riesgo de afectación al cultivo al proporcionar información en tiempo real sobre la presencia de condiciones propicias para una helada.

Esto permite que el agricultor tome medidas preventivas oportunas, como la aplicación de riego, el uso de coberturas térmicas o sistemas de calefacción, mitigando así el impacto del fenómeno. Gracias a su eficacia en la detección temprana y la generación de alertas, el sistema representa una herramienta clave para minimizar las pérdidas y mejorar la resiliencia de la producción agrícola.

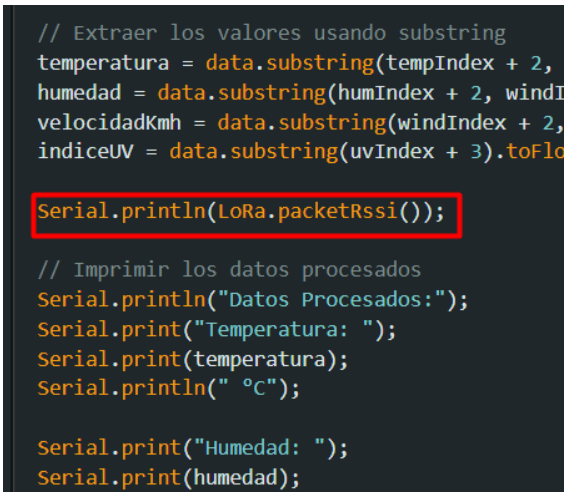
4.9. Comprobación de comunicación de módulos LoRa

Para la prueba de comunicación entre los nodos mediante LoRa, se recopilan datos a diferentes distancias, considerando el RSSI, como uno de los parámetros clave. Este valor permite visualizar la potencia de la señal recibida, lo que es fundamental para evaluar el rendimiento del enlace.

Para obtener el RSSI, se utiliza la función packetRssi() de la biblioteca LoRa, la cual imprime el dato en el receptor, en este caso, el Arduino como se observa en la Figura 98.

Figura 98

Función de impresión de RSSI en el Nodo Gateway



```
// Extraer los valores usando substring
temperatura = data.substring(tempIndex + 2,
humedad = data.substring(humIndex + 2, windI
velocidadKmh = data.substring(windIndex + 2,
indiceUV = data.substring(uvIndex + 3).toFlo

Serial.println(LoRa.packetRssi());

// Imprimir los datos procesados
Serial.println("Datos Procesados:");
Serial.print("Temperatura: ");
Serial.print(temperatura);
Serial.println(" °C");

Serial.print("Humedad: ");
Serial.print(humedad);
```

Fuente: Autor

Basándonos en la escala de interpretación del RSSI, se ha elaborado la Tabla 29, donde se presentan mediciones de señal a distancias que van desde 5 metros hasta 500 metros.

Tabla 29

Relación distancia-RSSI LoRa

Distancia (m)	RSSI Nodo Sensor (dbm)	Interpretación Calidad de la Señal
5	-32	Excelente
10	-35	Excelente
30	-48	Excelente
50	-55	Muy Buena
70	-62	Buena
100	-73	Buena
300	-90	Aceptable
500	-106	Baja Aceptable

Fuente: Autor

A partir del análisis de los valores de RSSI en función de la distancia, se observa que la intensidad de la señal disminuye a medida que la distancia entre los nodos aumenta. Sin embargo, mientras el nodo permanezca dentro de un rango adecuado, la comunicación sigue siendo estable. Si la distancia supera el límite en el que la señal es considerada aceptable, podrían ocurrir pérdidas de paquetes, afectando la conectividad entre el nodo sensor y el Gateway y comprometiendo la transmisión de datos**.**

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

El tiempo que una señal tarda en llegar al receptor se conoce como el Tiempo en el Aire de la transmisión. Este valor puede calcularse utilizando parámetros clave configurados en el nodo sensor, como el factor de dispersión (SF), el ancho de banda (BW) y la corrección de errores (CR). Todos estos parámetros fueron establecidos en el módulo RFM95W para garantizar una transmisión adecuada de los datos.

Para determinar el Tiempo en el Aire, es necesario realizar cálculos previos, como el Tiempo de Símbolo, que representa el tiempo que un único símbolo tarda en ser transmitido. Este valor se calcula mediante la ecuación (10), la cual establece una relación directa entre el factor de dispersión y el ancho de banda configurado. Estos cálculos son esenciales para entender y optimizar el rendimiento de la transmisión en redes LoRa.

$$T_s = \frac{2^{SF}}{BW(khz)} \tag{10}$$

$$T_s = \frac{2^7}{125\ khz}$$

$$T_s = 1.024\ ms$$

Al sustituir los valores en la fórmula correspondiente, se obtiene un Tiempo de Símbolo de 1.024 ms. El siguiente cálculo necesario es el del Tiempo del Preámbulo, que se determina utilizando la ecuación (11). Este cálculo depende del Tiempo de Símbolo previamente obtenido y del número de símbolos del preámbulo, cuyo valor se encuentra especificado en el datasheet

del módulo LoRa RFM95W. En este caso, el número de símbolos del preámbulo está estimado en 8, según las configuraciones estándar del módulo.

$$T_{preamble} = \left(n_{preamble} + 4.25\right) * T_{sym} \quad (11)$$

$$T_{preamble} = (8 + 4.25) * 1.024$$

$$T_{preamble} = 12.544$$

Con los datos previamente calculados, se procede a aplicar la ecuación (12), que permite determinar la longitud de símbolo correspondiente a la carga útil (payload). Este cálculo es fundamental, ya que establece la cantidad de tiempo que cada símbolo de la carga útil ocupará durante la transmisión, lo que impacta directamente en el rendimiento y la eficiencia de la comunicación LoRa.

$$N_{payload} = 8 + \left[\left(\frac{8 * PL - 4 * SF + 28 + 16\text{CRC} - 20 * H}{4(SF - 2 * DE)}\right)(CR + 4)\right] \quad (12)$$

Donde:

PL = Numero de bytes en la carga útil

SF = Factor de dispersión

CRC = Es la comprobación de redundancia cíclica. 1 activado 0 desactivado

H = Cabecera del paquete LoRa. 1 activada 0 desactivada

DE = Optimización de baja velocidad de datos.

CR = Tasa de codificación.

Tabla 30

Parámetros para calcular longitud de símbolo que tendrá la carga útil

Parámetro	Valor
PL	26 bytes
SF	7
CRC	1
H	0
DE	0
CR	1

Fuente: Autor

Los valores utilizados en esta fórmula están detallados en la Tabla 30, los cuales sirven como base para llevar a cabo el cálculo correspondiente. A partir de estos datos, se procede a realizar el cálculo de manera precisa, asegurando la correcta interpretación y aplicación de la fórmula.

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

$$N_{payload} = 48 \text{ símbolos}$$

Otro parámetro que se necesita para el cálculo de ToA es el tiempo de carga útil el cual calculamos mediante la ecuación (13)

$$T_{payload} = N_{payload} * T_s \quad (13)$$

$$T_{payload} = 48 * 1024$$

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

Ahora es posible calcular el Tiempo en el Aire del paquete utilizando la ecuación (1), que establece la relación entre el Tiempo de la Carga Útil y el Tiempo del Preámbulo. Este cálculo

combina ambos valores previamente determinados para obtener el tiempo total que el paquete permanecerá en el aire durante su transmisión.

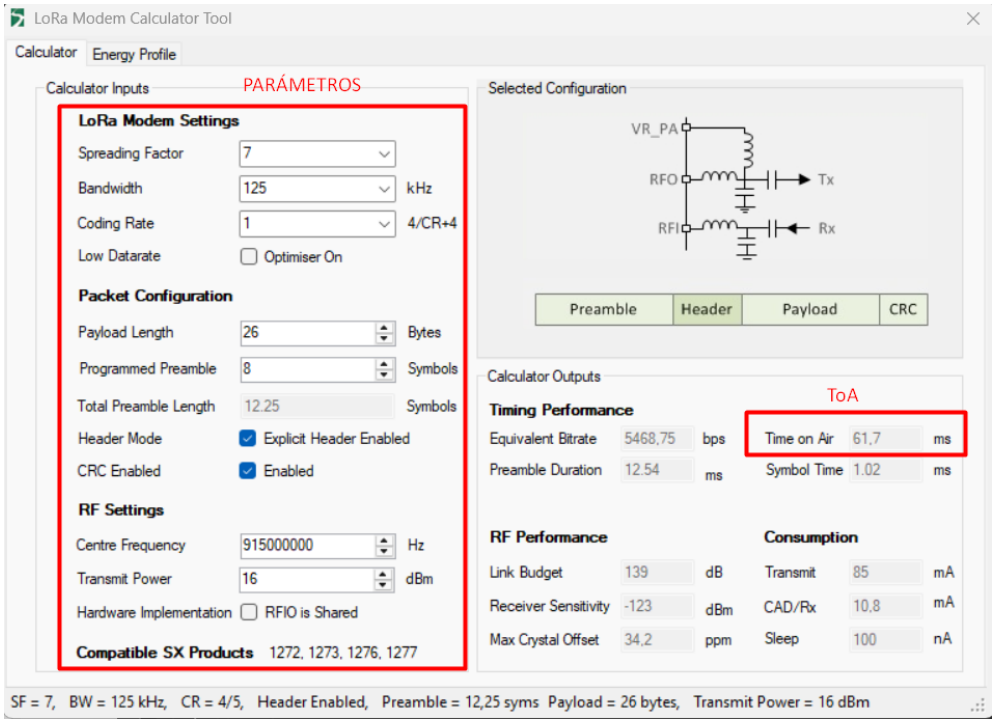
$$ToA = 49.152\ ms + 12.54\ ms$$

$$\boldsymbol{ToA = 61.69\ ms}$$

Para garantizar la precisión de los resultados obtenidos a través de cálculos teóricos, es fundamental contar con herramientas de validación. Una de estas herramientas es la calculadora LoRa desarrollada por Semtech, la cual está diseñada específicamente para calcular de forma automática el ToA en comunicaciones LoRa. Al ingresar los parámetros de configuración se obtiene un valor de ToA que se observa en la Figura 99 de 61.7 ms el cual es un valor similar al calculado teóricamente con una diferencia mínima, garantizando una mayor confiabilidad en los resultados obtenidos, lo que contribuye a un diseño más robusto y eficiente de la red LoRa.

Figura 99

Calculadora LoRa de ToA de Semtech



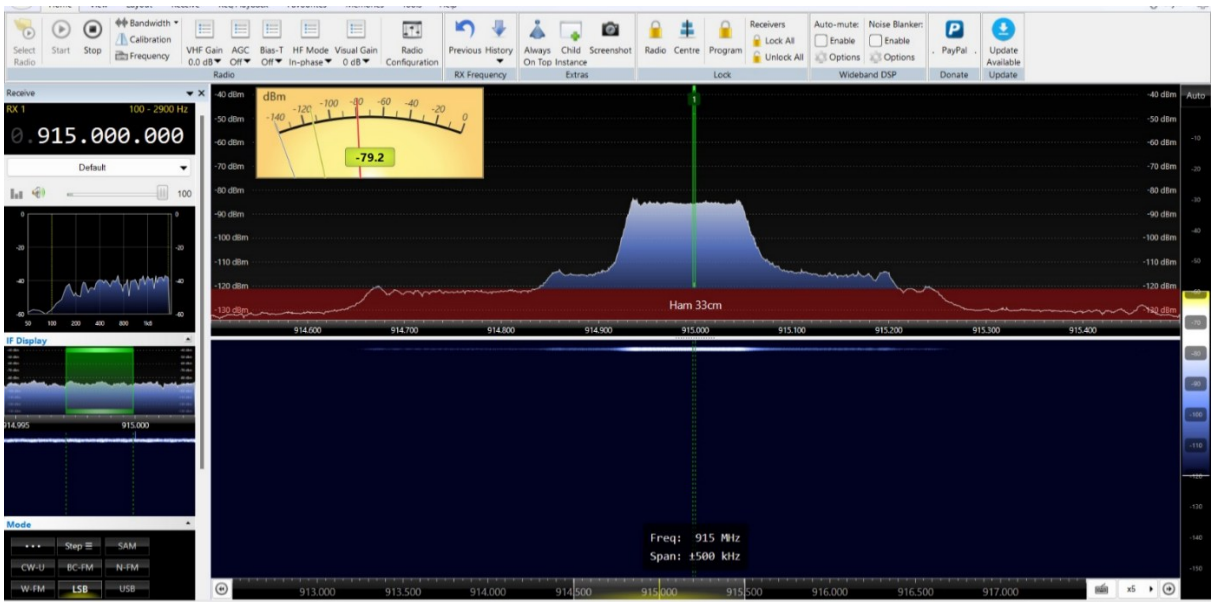
Fuente: Autor

4.11. Análisis de la señal LoRa mediante RTL-SDR

Para garantizar que nuestra comunicación LoRa está transmitiendo los paquetes con los parámetros configurados correctamente, se utiliza el dispositivo RTL-SDR. Este dispositivo permite capturar y analizar las señales de los paquetes que se están enviando en tiempo real. Como se muestra en la Figura 100, el RTL-SDR se configura con la frecuencia central correspondiente a la utilizada en la transmisión, lo que permite sincronizarse con los paquetes emitidos por el nodo sensor.

Figura 100

Captura de paquetes transmitidos mediante RTL-SDR



Fuente: Autor

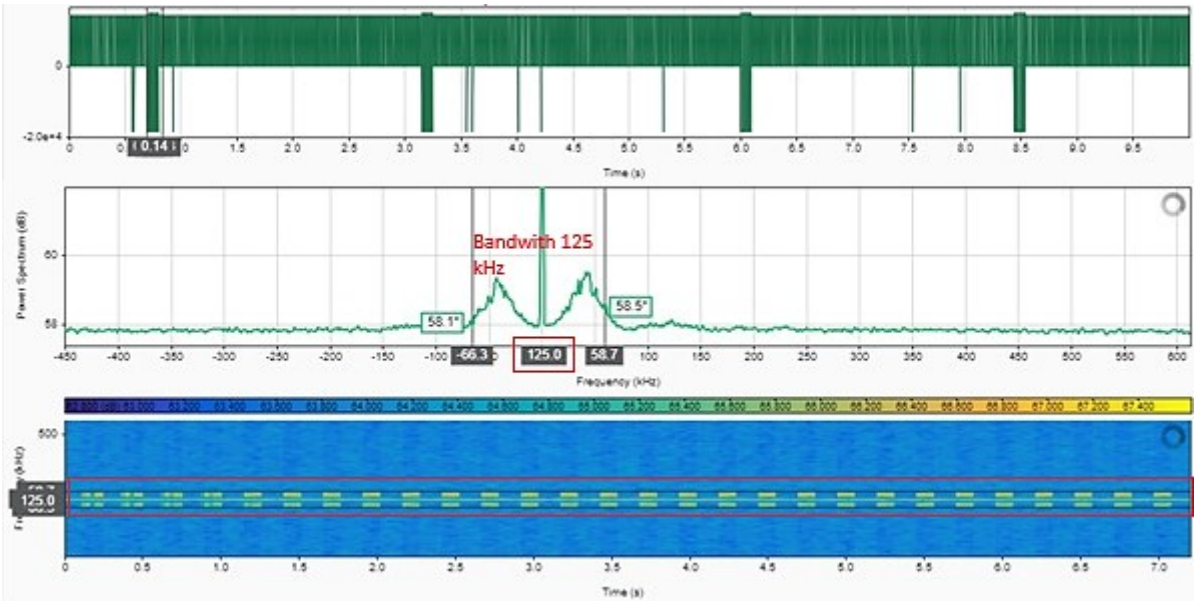
Al realizar esta configuración, el espectrograma, se observa una forma de onda característica de LoRa, lo cual confirma que el dispositivo está transmitiendo datos con este tipo de modulación. La señal está claramente delimitada, lo que sugiere una transmisión estable y bien configurada. Este procedimiento es fundamental para validar la comunicación efectiva entre los nodos sensores y el Gateway, asegurando que los datos enviados cumplen con los parámetros definidos en el sistema.

La señal capturada también se analiza utilizando MATLAB, lo que permite una observación más detallada y precisa de sus características. Al cargar la señal en MATLAB y visualizarla mediante herramientas como el Signal Analyzer, se pueden identificar parámetros clave que describen su comportamiento. Como se muestra en la Figura 101, la señal se encuentra correctamente centrada en 915 MHz, con un ancho de banda de 125 kHz, lo que corresponde a las especificaciones esperadas para una comunicación LoRa.

Otro aspecto importante observado en MATLAB es la ausencia de anomalías importantes en la modulación o transmisión, como interferencias o distorsiones significativas. Esto indica que la configuración del transmisor y el receptor es adecuada, y que el enlace de comunicación es robusto.

Figura 101

Análisis en Matlab de Captura realizada mediante RTL-SDR



Fuente: Autor

4.12. Costos del Sistema

Los costos asociados con la implementación de un Sistema de Alerta contra Heladas se analizan en función del beneficio que este sistema proporciona. Este análisis busca evaluar la

relación costo-beneficio, considerando tanto los gastos involucrados en el desarrollo e instalación del sistema como las ventajas económicas y operativas derivadas de su implementación, por lo cual se detallan todos los costos implicados en las siguientes secciones.

4.12.1. Costo de Hardware

Cada componente del sistema tiene un costo específico en el mercado, el cual se detalla en la Tabla 31 junto con sus valores individuales. Esta tabla proporciona un desglose preciso de los precios, facilitando el análisis del costo total del hardware necesario para la implementación del sistema.

Tabla 31

Costos Hardware

Hardware	**Cantidad**	**Precio unitario**	**Total**
Sensor Guva S12SD	1	$11	$11
Sensor DTH11	1	$4	$4
Sensor Anemómetro	1	$46	$46
Arduino Uno	1	$10	$10
ESP32 WROOM	1	$13.50	$13.50
Módulos LoRa	2	$10	$20
Rasbperry Model 3 Pi	1	$120	$120
PowerBank	1	$12	$12
Otros Elementos			$20
		TOTAL (USD)	$ 256.50

Fuente: Autor

4.12.2. Costos de Software

De manera similar en la Tabla 32, se detallan los costos asociados a los softwares utilizados en el sistema. Estos precios incluyen las licencias, herramientas de desarrollo y cualquier otro software requerido para la implementación y funcionamiento del proyecto. La información se presenta de forma clara para facilitar el análisis del costo total.

Tabla 32

Costos de Software

Software	**Cantidad**	**Precio unitario**	**Total**
Almacenamiento en la Nube	1	0	0
Sistema Operativo Microprocesador	1	0	0
Librerías	1	0	0
Node-Red	1	0	0
		Total (USD)	**0**

Fuente: Autor

Los softwares utilizados no han generado ningún costo, ya que se han empleado herramientas con licencia libre.

4.12.3. Costos de infraestructura

Asimismo, se consideran los costos relacionados con la infraestructura, específicamente los gastos asociados a las carcasas utilizadas en cada nodo del sistema. Los cuales son detallados en la Tabla 33.

Tabla 33

Costos de infraestructura

Infraestructura	**Cantidad**	**Precio unitario**	**Total**
Carcasa Protectora interna Nodo Sensor	1	$6	$6
Carcasa Protectora externa Nodo Sensor	1	$5	$5
Carcasa Protectora Nodo Gateway	1	$5	$5
		Total (USD)	**$16**

Fuente: Autor

4.12.4. Costo total del Sistema

Para determinar el costo total del Sistema de Alerta contra Heladas, se procede a sumar todos los valores correspondientes a los costos de software, hardware e infraestructura como se muestra en la Tabla 34. Este cálculo permite obtener una visión completa de la inversión necesaria para la implementación del sistema.

Tabla 34

Costo total del Sistema

Costo	**Subtotal (USD)**
Hardware	$ 256.50
Software	$0.00
Infraestructura	$16.00
Total (USD)	**272.50**

Fuente: Autor

El costo total de la implementación del sistema es de $272.50 el cual varia si se incrementan más nodos dependiendo de la extensión del cultivo. En base a todo esto se puede deducir que el Sistema de Alerta contra Heladas representa una solución altamente rentable, especialmente en regiones donde las heladas constituyen una amenaza recurrente para la producción agrícola. La capacidad de prever condiciones climáticas adversas y alertar a los agricultores con anticipación permite reducir significativamente las pérdidas en los cultivos, mejorando la estabilidad económica de los productores. La implementación de este sistema no solo protege las cosechas, sino que también impulsa la adopción de tecnologías innovadoras en el sector agrícola, promoviendo una gestión más sostenible de los cultivos.

Si bien la inversión inicial en hardware e infraestructura puede representar un costo significativo, los beneficios a mediano y largo plazo superan ampliamente estos costos. La disminución en la pérdida de cultivos y el aumento en la productividad generan un retorno de

inversión positivo, lo que hace que el sistema sea económicamente viable. Además, el uso de software libre elimina costos adicionales asociados a licencias y suscripciones, permitiendo que la solución sea accesible y escalable para diversas comunidades agrícolas.

CONCLUSIONES

La implementación del Sistema de Alerta contra Heladas ha demostrado ser una herramienta eficaz para mitigar los efectos de este fenómeno, gracias a la integración de una arquitectura IoT con técnicas de Inteligencia Artificial. Su aplicación en el sector agrícola permite un monitoreo en tiempo real de las condiciones climáticas, proporcionando alertas tempranas que ayudan a prevenir pérdidas en los cultivos y optimizar la reacción y acción de los agricultores.

Como parte del desarrollo del proyecto, se realizó una investigación detallada sobre los términos y conceptos relacionados, lo que permitió una comprensión profunda del comportamiento del fenómeno estudiado. Este análisis facilitó la identificación de las tecnologías adecuadas y las técnicas de Inteligencia Artificial más eficientes para garantizar el óptimo funcionamiento del sistema, asegurando así su precisión y efectividad en la detección y mitigación de heladas.

Los requerimientos de software y hardware fueron definidos con precisión a partir de la investigación previa y el análisis de las necesidades planteadas por el agricultor. Este proceso permitió desarrollar una solución efectiva y adaptada al entorno agrícola, garantizando su funcionalidad, fiabilidad y adecuación para solucionar la problemática.

El Sistema de Alerta contra Heladas ha sido desarrollado con éxito, incorporando una red IoT para el monitoreo preciso de variables climáticas y la transmisión de datos a través de comunicación inalámbrica. Además, la integración de Inteligencia Artificial ha optimizado el análisis de la información recopilada, permitiendo una detección del fenómeno y la generación de alertas en tiempo real, mejorando así la capacidad de respuesta y prevención en el sector agrícola.

Las pruebas de funcionamiento realizadas confirman la fiabilidad, precisión y sostenibilidad del sistema. Los resultados evidencian que la solución desarrollada es una

herramienta innovadora y viable para la detección temprana de heladas en los cultivos, impulsando la adaptación de tecnologías que cubran las problemáticas en el sector agrícola y mejorando la capacidad de respuesta ante condiciones climáticas adversas.

RECOMENDACIONES

Para mejorar el alcance del monitoreo, se sugiere evaluar la posibilidad de implementar más nodos sensores en distintos lugares del cultivo en caso de que este sea extenso. Esto permitiría una detección más localizada y precisa de las condiciones climáticas, optimizando la prevención de heladas en áreas con diferentes microclimas.

Se recomienda actualizar periódicamente el entrenamiento de la Inteligencia Artificial utilizando los datos históricos recopilados, con el objetivo de mejorar la precisión en la predicción de heladas. Dado que las condiciones climáticas pueden variar abruptamente, es fundamental ajustar el modelo a los nuevos patrones detectados, permitiendo una adaptación continua y una mayor fiabilidad en la generación de alertas tempranas.

Se recomienda realizar pruebas periódicas para verificar el correcto funcionamiento de cada sensor, ya que, aunque están diseñados para operar en exteriores, la exposición constante a las condiciones ambientales puede afectar su precisión y rendimiento con el tiempo. Estas revisiones permitirán detectar posibles fallos o desviaciones en las mediciones, asegurando la fiabilidad del sistema y la precisión en la detección de heladas

Además de la generación de alertas, se sugiere explorar la posibilidad de integrar mecanismos de respuesta automatizada es decir actuadores, como activación de sistemas de riego o dispositivos de calefacción agrícola, que ayuden a mitigar el impacto de las heladas de manera eficiente y sin intervención manual inmediata.

REFERENCIAS

Amazon Web Services (AWS). (2023). *¿Cuál es la diferencia entre el aprendizaje supervisado y el no supervisado?* https://aws.amazon.com/es/compare/the-difference-between-machine-learning-supervised-and-unsupervised/

Andrade, N. (2017, March 23). *La Importancia de la Agricultura en nuestro país*. https://agropecuaria.utn.edu.ec/?p=1091

Arduino.cl. (2023). *Arduino Uno en Linea*. https://arduino.cl/arduino-uno/

Bassi, A. (2021, February 5). *Introducción a MQTT*. https://www.gotoiot.com/pages/articles/mqtt_intro/index.html

BBVA. (2023, December 21). *¿Qué es la agricultura de precisión? La gestión digital del campo*. https://www.bbva.com/es/sostenibilidad/que-es-la-agricultura-de-precision-la-gestion-digital-del-campo/

Bravo, R., Quintana, J., & Reyes, M. (2020). *Heladas. Factores, tendencias y efectos frutales y vides*.

Bruno, A. (2019). *ESP32 NODE MCU*. https://www.microelectronicash.com/downloads/ESP32_MANUAL.pdf

Cadena, H. (2020). *DISEÑO DE UN SISTEMA PARA EL CONTROL DE RIEGO MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO APLICADA A LA AGRICULTURA DE PRECISIÓN EN LA GRANJA LA PRADERA DE LA UNIVERSIDAD TÉCNICA DEL NORTE*. Universidad Técnica del Norte.

Cajas, K. (2022). *Redes de sensores inalámbricos para IoT automatización de redes inalámbricas de sensores*. Escuela Politécnica Nacinal.

Cevallos, B., & Rubio, S. (2021). *DESARROLLO DE UNA RED IOT CON TECNOLOGÍA LORA PARA GESTIÓN DE INVERNADEROS*. UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO.

Chacua, B. (2019). *Diseño de un sistema prototipo de reconocimiento facial para la identificación de personas en la Facultad de Ingeniería en Ciencias Aplicadas (FICA) de la Universidad Técnica del Norte utilizando técnicas de Inteligencia Artificial*. Universidad Técnica del Norte.

Chávez, J. (2021). *Impacto del Cambio Climático en la Agricultura en los Cantones Cayambe y Pedro Moncayo. Cuestiones Económicas*. *31*. https://doi.org/https://doi.org/10.47550/RCE/MEM/31.63

Chetto, M., & Queudet, A. (2020). *Sistemas de tiempo real autónomos en energía*. ISTE Science Publishing.

Chiriboga, Á. (2020). *Diseño e implementación de una solución con tecnología LORA para el monitoreo de ubicación vehicular con un aplicativo web*. Universidad de las Fuerzas Armadas.

Cordero, A. (2013). *Caracterización, identificación y Evaluación del Fenómeno Meteorologíco la Helada Durante los Años 2005 al 2011, en las Zonas Comprendidas por las Tres Estaciones Metereologícas de: Quimsocha, la Esmeralda y San Gerardo*.

El Universo. (2022, August 8). Cultivos de papa, fresa y maíz están afectados por las heladas en Tungurahua. *El Universo*.

Eterovic, M. J., Cipriano, E. M., Garcia, L. E., & Torres, L. L. (2019). *A Proposal for the Integration of Blockchain with the Internet of Things*. http://reddi.unlam.edu.arPág:1artículooriginal

Ferrandez, F., Maciá, M., Platero, M., & Silveira, D. (2021). *Plataforma IoT basada en paneles de monitorización (Dashboard)*.

Flores, F., & Gonzalo, E. (2021). *Aplicaciones, Enfoques y Tendencias del Internet de las Cosas (IoT): Revisión Sistemática de la Literatura*. *13*(9), 568.

Freire, L. (2023). *Electrónica y sistemas embebidos. Una visión a nivel técnico y tecnológico*. https://doi.org/https://doi.org/10.33996/cide.ecuador.ES2636225

GNS Components. (2023). *CJMCU-GUVA-S12SD Intensidad De La Luz Solar Del Sensor Ultravioleta*. http://es.led-diode.com/development-board/cjmcu-guva-s12sd-sunlight-intensity-of-uv.html

González, L. (2019). *Test y despliegue de tecnología de comunicaciones LoRa para aplicaciones de Internet of Things*. Universidad Politécnica de Madrid.

Grupo de Investigación en Agrítica y Agricultura de Precisión (GRAP). (2022, November 11). *Agricultura de Precisión*.

Halfacree, G. (2020). *La guía oficial de Raspberry Pi para principiantes : cómo usa tu nuevo ordenador*.

IBM. (2021). *¿Qué es una notificación push?* https://www.ibm.com/es-es/topics/push-notifications

IEEE/ISO/IEC. (2018). *International Standard - Systems and software engineering -Std 29148. 104*. https://standards.ieee.org/standard/29148-2018.html

INESDI. (2022, November 2). *¿Qué es el aprendizaje no supervisado y cuándo usarlo?* https://www.inesdi.com/blog/que-es-aprendizaje-no-supervisado/

Instituto Nacional de Ciberseguridad (INCIBE). (2021, September 29). *El correo electrónico: una herramienta básica para comenzar en Internet*. https://www.incibe.es/ciudadania/blog/el-correo-electronico-una-herramienta-basica-para-comenzar-en-internet

ITU-R. (2015). *Propagation data and prediction methods for the planning of terrestrial lineof-sight radio systems operating in the frequency range 30 MHz to 3000 MHz. International Telecommunication Union, Geneva, Switzerland*.

Kuongshun. (2023). *DHT11 Temperatura Y El Módulo De Sensor De Humedad*. https://www.szks-kuongshun.com/uno/uno-sensor/dht11-temperature-and-humidity-sensor-module.html

Libson, S. (2022, March 15). *Guía de protocolos y estándares de IoT*. https://www.kellton.com/kellton-tech-blog/your-complete-guide-to-iot-protocols-and-Standards-2022

Llamas, L. (2019, April 18). *¿Qué es MQTT? Su importancia como protocolo IoT*. https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/

Londaña, P. (2023, April 3). *Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender)*. https://blog.hubspot.es/website/que-es-python

LoRa Alliance®. (2022). *What is LoRaWAN® Specification*. https://lora-alliance.org/about-lorawan/

LoRaWAN®. (2022). *LoRaWAN® Regional Parameters 46 RP002-1.0.4*. https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters

MathWorks. (2023). *Introducción a aprendizaje supervisado*. https://la.mathworks.com/discovery/supervised-learning.html

Microsoft. (2021). *Conceptos básicos sobre bases de datos*. https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204

Ministerio de Inclusión Económica y Social (MIES). (2020). *PRODUCTOS DE LA SIERRA Y DEL SUBTRÓPICO SE INTERCAMBIAN PARA ENTREGAR A FAMILIAS DE SECTORES PRIORITARIOS*. https://www.inclusion.gob.ec/productos-de-la-sierra-y-del-subtropico-se-intercambian-para-entregar-a-familias-de-sectores-prioritarios/#

Ministerio de Inclusión Económica y Social (MIES). (2022). *PRODUCTOS DE LA SIERRA Y DEL SUBTRÓPICO SE INTERCAMBIAN PARA ENTREGAR A FAMILIAS DE SECTORES PRIORITARIOS*. https://www.inclusion.gob.ec/productos-de-la-sierra-y-del-subtropico-se-intercambian-para-entregar-a-familias-de-sectores-prioritarios/#:~:text=Los%20productos%20que%20salen%20de,%2C%20naranjas%2C%20mandarinas%20y%20guan%C3%A1banas.

Ministerio del Ambiente Agua y Transición Ecológica (MAATE). (2023). *PLAN NACIONAL DE ADAPTACIÓN AL CAMBIO CLIMÁTICO DEL ECUADOR 2023 - 2027*. www.ambiente.gob.ec

Ministerio del Ambiente (MAE). (2014). *"GESTIÓN INTEGRADA PARA LA LUCHA CONTRA LA DESERTIFICACIÓN, DEGRADACIÓN DE LA TIERRA Y ADAPTACIÓN AL CAMBIO CLIMÁTICO" CUP Nro. 40400000.0000.376207 SUBSECRETARÍA DE CAMBIO CLIMÁTICO*.

Narváez, K., & Contreras, V. (2020). *Diseño y desarrollo de un prototipo de red de sensores IOT utilizando tecnología Lorawan para el monitoreo de parámetros ambientales en interiores y exteriores*. Universidad Politécnica Salesiana Sede Guayaquil.

Netafim. (2023, January 5). *MITIGACIÓN DE HELADAS: CÓMO PODEMOS HACERLE FRENTE*. https://es.linkedin.com/pulse/mitigaci%C3%B3n-de-heladas-c%C3%B3mo-podemos-hacerle-frente-

Organización de Naciones Unidas (ONU). (2015). *Agenda 2030: Objetivos de Desarrollo Sostenible*. https://www.un.org/sustainabledevelopment/es/

Palaguachi, S. (2018). *Diseño, desarrollo e implementación de una estación meteorológica basada en una red jerárquica de sensores, software libre y sistemas embebidos para la Empresa ELECAUSTRO en la Minicentral Gualaceo utilizando comunicación MQTT y MODBUS*. Universidad Politécnica Salesiana Sede Cuenca.

Panza, G. (2024, January 6). *What LoRaWAN and "Chirp Spread Spectrum" (CSS) technology have in common?* https://www.linkedin.com/pulse/what-lorawan-chirp-spread-spectrum-css-technology-have-giuseppe-panza-ldk4c

PROAIN. (2020, March 10). *EFECTO DE LAS HELADAS EN LA AGRICULTURA*. https://proain.com/blogs/notas-tecnicas/efecto-de-las-heladas-en-la-agricultura

Qin, J., Li, Z., Wang, R., Li, L., Yu, Z., He, X., & Liu, Y. (2021). Industrial Internet of Learning (IIoL): IIoT based pervasive knowledge network for LPWAN—concept, framework and case studies. *CCF Transactions on Pervasive Computing and Interaction*, *3*(1), 25–39. https://doi.org/10.1007/s42486-020-00050-2

Ramírez, P. (2023). *DISEÑO DE UN SISTEMA DE MEDICIÓN PARA EL NIVEL DE AGUA DEL CAUDAL DE LA PRINCIPAL FUENTE DE CAPTACIÓN DE LA CIUDAD DE TULCÁN UTILIZANDO LA TECNOLOGÍA DE COMUNICACIÓN LORA.* Universidad Técnica del Norte.

Raspberry Pi. (2022). *Raspberry Pi 3 Model B*. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

Reventós, V. (2019). *Qué es la inteligencia artificial*.

Rodríguez, A. (2023). *DISEÑO DE UNA RED DE SENSORES SECTORIZADA PARA EL MONITOREO REMOTO DE NUTRIENTES, HUMEDAD Y TEMPERATURA EN CULTIVOS DE VEGETALES Y TUBÉRCULOS A TRAVÉS DE TECNOLOGÍAS LPWAN*.

Romero, M. (2024). *Diseño de un sistema de monitoreo y gestión de humedad en el suelo de cultivo de papa en la finca Villa Lola, utilizando una red de sensores inalámbricos (WSN) y tecnologías LPWAN*. Universidad Técnica del Norte.

Sarandón, S. J. (2020). *El papel de la agricultura en la transformación social-ecológica de América Latina*.

Secretaría Nacional de Planificación (SNP). (2024). *Plan de Desarrollo Para el Nuevo Ecuador (2024 - 2025)*.

Sencrop. (2022, March 28). *Los diferentes tipos de heladas y sus parámetros clave*. https://es.blog.sencrop.com/los-diferentes-tipos-de-heladas-y-sus-parametros-clave/

Sencrop. (2023, January 3). *¿Cómo combatir las heladas?* https://es.blog.sencrop.com/como-combatir-las-heladas/

Sigma. (2023). *Módulo transceptor de largo alcance (LoRa) RFM95W- 915 MHZ*. https://www.sigmaelectronica.net/producto/rfm95w-915mhz/

SIISA GLOBAL. (2021, July 21). *Microcontroladores. ¿Qué son? y su importancia en la industria.* https://es.linkedin.com/pulse/microcontroladores-qu%C3%A9-son-y-su-importancia-en-la-industria-

Singh, N. (2023, September 25). *Métricas De Evaluación De Modelos En El Aprendizaje Automático*. https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

Solutec. (2020, August 6). *¿Qué es el IoT y cómo impacta en la seguridad?* https://solutec-latam.com/el-protocolo-mqtt-en-la-seguridad-y-el-monitoreo-de-alarmas/

Toulkeridis, T., Tamayo, E., Simón, D., Merizalde, M., Reyes, D., Viera, M., & Heredia, M. (2020). Cambio Climático según los académicos ecuatorianos - Percepciones versus hechos. *Ciencias de La Vida*, *31*, 21–46.

UIT. (2012). *UIT-T Rec. Y.2060 Descripción general de Internet de los objetos*. http://handle.itu.int/11.1002/1000/11830-en.

Varela, A. L., & Ron, S. R. (2018). *Geografía y Clima del Ecuador. BIOWEB. Pontificia Universidad Católica del Ecuador*. https://bioweb.bio/fungiweb/GeografiaClima/

Wiater, A. (2022). Metodología de investigación-acción en la didáctica de lenguas extranjeras: sus posibilidades y límites. *Lenguas Modernas*, *59*, 51–63. https://lenguasmodernas.uchile.cl/index.php/LM/article/view/67933

Xukyo. (2024, February 18). *Descripción general del microcontrolador NodeMCU ESP32*. https://www.aranacorp.com/es/descripcion-general-del-microcontrolador-nodemcu-esp32/

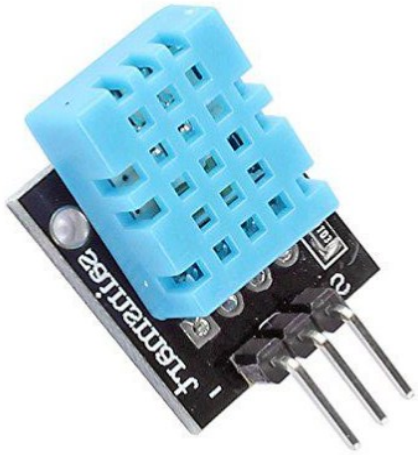
Xunta de Galicia. (2019, June). *Meteogalicia*. https://www.meteogalicia.gal/web/home

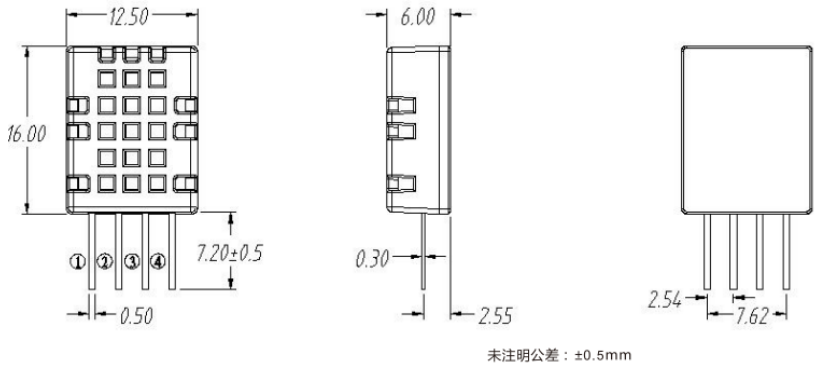
ANEXOS

Anexo 1 – DataSheet DHT11

Dirección de documentación:

https://www.alldatasheet.es/view.jsp?Searchword=DHT11





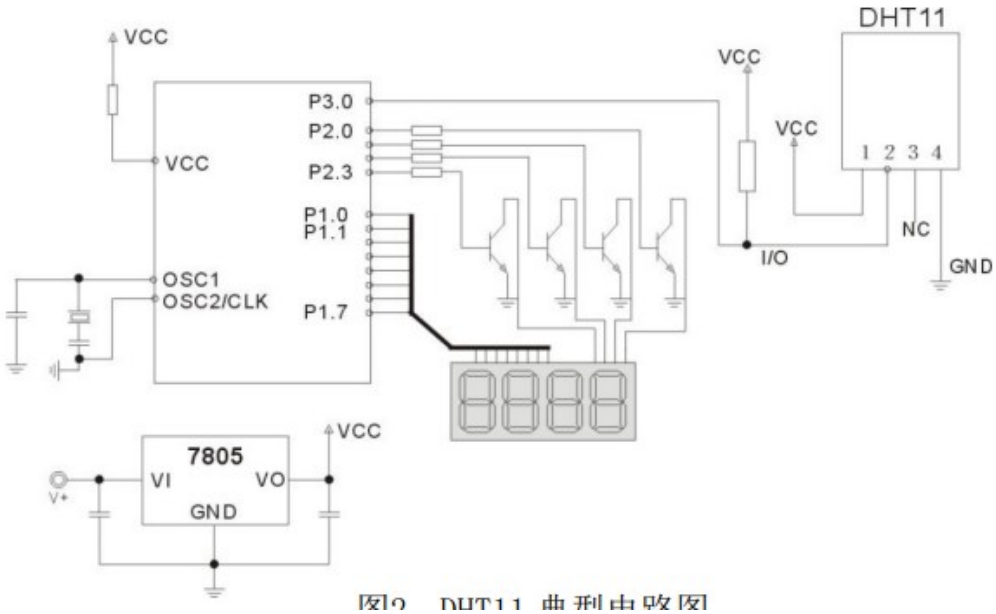
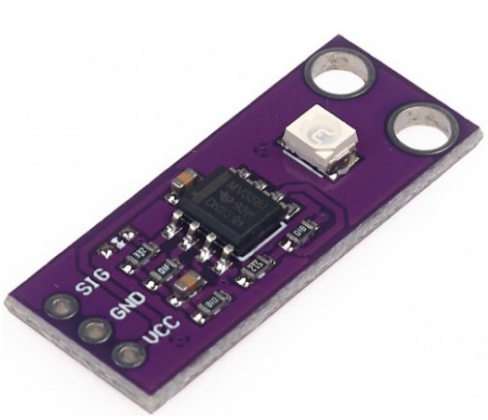


图2 DHT11 典型电路图

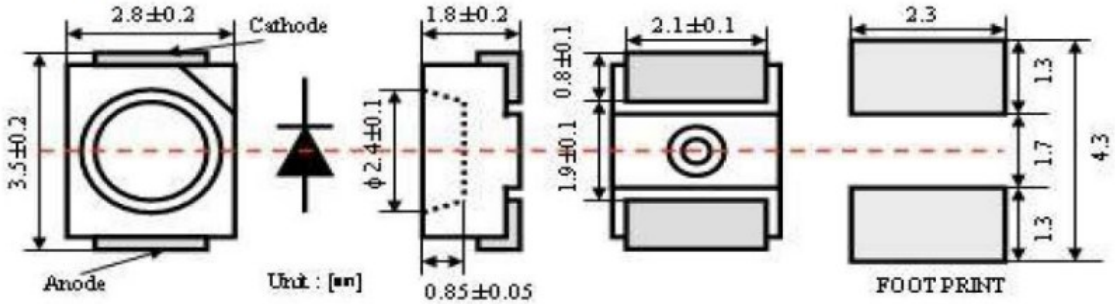
Anexo 2 – DataSheet Guva S12SD

Dirección de documentación:

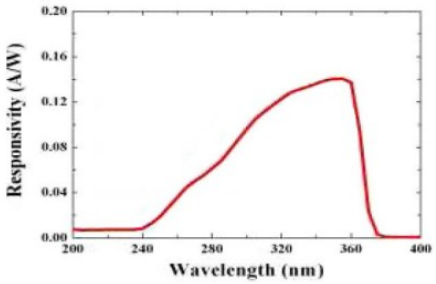
https://www.alldatasheet.es/datasheet-pdf/pdf/712047/ROITHNER/GUVA-S12SD.html



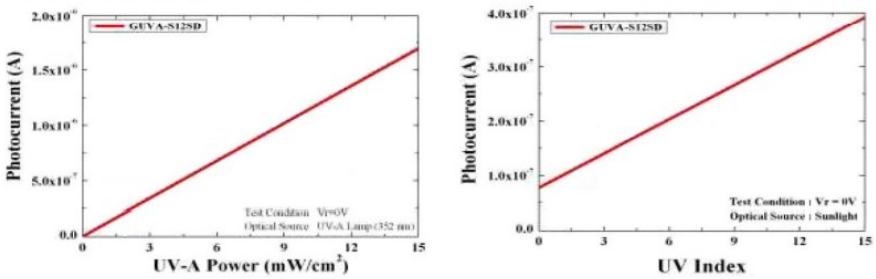
Package Dimension



Responsivity Curve



Photocurrent along UV Power



Anexo 3 – DataSheet módulo RFM95W

Dirección de documentación:

https://www.alldatasheet.com/view.jsp?Searchword=Rfm95w&gad_source=1&gclid=CjwKCAiAh6y9BhBREiwApBLHCyEfrFh6s8YPDyLmm_JcvLWP-a7dyusBxfoHbXQq80o2wzqfxcC3rhoCEsIQAvD_BwE

1.1. Simplified Block Diagram

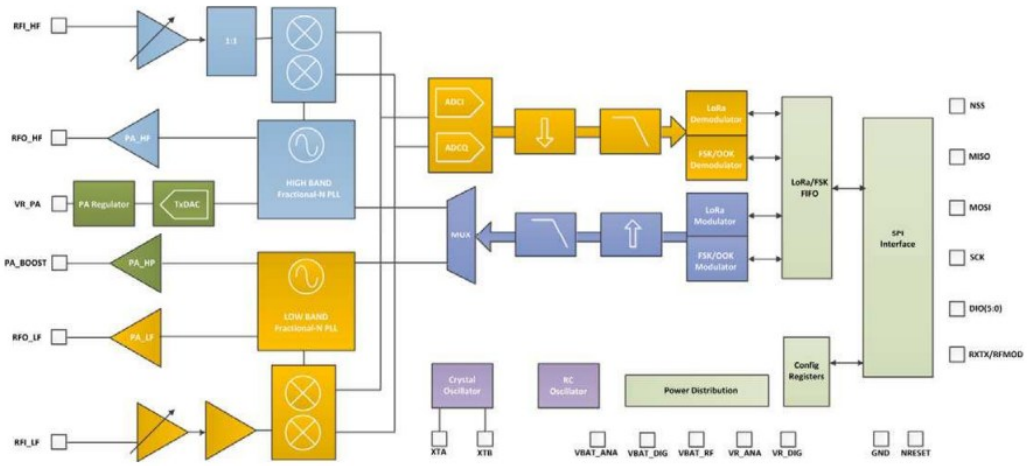


Figure 1. Block Diagram

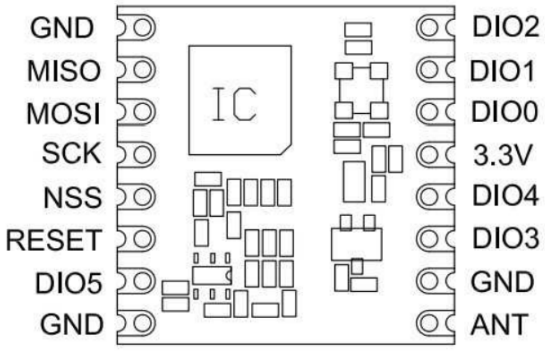


Figure 2. Pin Diagrams

Anexo 4 – Códigos del Sistema

Todos los códigos utilizados para la configuración de los componentes se encuentran en el repositorio GitHub.

https://github.com/Ka635/SISTEMA_HELADAS.git

Anexo 5 – Implementación Sistema Nodo Sensor

Anexo 6 – Implementación Sistema Nodo Sensor Pruebas

Anexo – Implementación Sistema Nodo Sensor Gateway

Anexo 8 – Alertas Emitidas