

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN SERVICIOS BIBLIOTECARIOS						 <small>SC 7384-1</small>	 <small>SA-CERE 597526</small>	 <small>OS-CER 597555</small>	
CARTA DE AUTORIZACIÓN										
CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 2			

Neiva, Huila, 18 de enero de 2022

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Nicolas Francisco Liscano Caldas , con C.C. No. 1107102300

Luis Fernando Mosquera Imbachi, con C.C. No. 1075303936

Autor(es) de la tesis y/o trabajo de grado titulado DESARROLLO DE UN GATEWAY LPWAN PARA EL CONTROL ON/OFF Y MONITOREO DE UN CULTIVO DE TOMATE presentado y aprobado en el año 2022 como requisito para optar al título de INGENIERO ELECTRÓNICO ;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permite la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN SERVICIOS BIBLIOTECARIOS

CARTA DE AUTORIZACIÓN



CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2
--------	--------------	---------	---	----------	------	--------	--------

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Firma: Nicolas Francisco Liscano Caldas

EL AUTOR/ESTUDIANTE:

Firma: Luis Fernando Mosquera Imbachí



UNIVERSIDAD SURCOLOMBIANA GESTIÓN
SERVICIOS BIBLIOTECARIOS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 4
--------	--------------	---------	---	----------	------	--------	--------

TÍTULO COMPLETO DEL TRABAJO: DESARROLLO DE UN GATEWAY LPWAN PARA EL CONTROL ONOFF Y MONITOREO DE UN CULTIVO DE TOMATE AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Liscano Caldas	Nicolas Francisco
Mosquera Imbachí	Luis Fernando

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Quintero Polanco	Jesús David

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
Quintero Polanco	Jesús David

PARA OPTAR AL TÍTULO DE: INGENIERO ELECTRÓNICO

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA GESTIÓN
SERVICIOS BIBLIOTECARIOS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 4
--------	--------------	---------	---	----------	------	--------	--------

FACULTAD: INGENIERÍA

PROGRAMA O POSGRADO: INGENIERÍA ELECTRÓNICA

CIUDAD: NEIVA, HUILA

AÑO DE PRESENTACIÓN: 2022

NÚMERO DE PÁGINAS: 110

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados
Láminas Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros

SOFTWARE requerido y/o especializado para la lectura del documento: Microsoft Word o Adobe PDF

MATERIAL ANEXO: Ninguno

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

	Español	Inglés		Español	Inglés
1.	LoRa	LoRa		6. ChirpStack	ChirpStack
2.	LPWAN	LPWAN		7. ADC	ADC
3.	Gateway	Gateway		8. Modulacion	Modulation
4.	TTGo t-beam	TTGO t-Beam		9. IoT	IoT
5.	Sevidor de red	Network Server		10. RSSI	RSSI

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA GESTIÓN SERVICIOS BIBLIOTECARIOS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 4
--------	--------------	---------	---	----------	------	--------	--------

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

En este documento se detalla el diseño de una red LPWAN de bajo costo, basada en la modulación LoRa. Esto se logra, usando una Raspberry PI-4 junto con el concentrador RAK2287 como puerta de enlace. En los nodos, se utilizó el módulo TTGO T- Beam basado en el microcontrolador ESP32, y el controlador Dragino LT-33222L. El primer nodo, fue el responsable de enviar datos de humedad y temperatura. El segundo se encargó, de accionar el sistema de riego cuando los requerimientos de humedad así lo requirieron, y adicionalmente, permitió el llenado del tanque de agua utilizado en el cultivo.

Para la plantación de tomate, se construyó dos huertas de madera de dimensiones necesarias para su correcto crecimiento y desarrollo. La temperatura y humedad fue monitoreada en dichas huertas, sin embargo la huerta 1 correspondió al cultivo que implementó el sistema de riego por goteo que funcionó bajo el control ON/OFF de los actuadores, mientras que la huerta 2 fue el cultivo que se regó de forma manual mediante riego con pulverizador. En el transcurso de 3 meses, el cultivo presentó ciertas enfermedades y plagas, que se trataron de manera natural o con el uso de pesticidas. Utilizando el servicio de ChirpStack, se creó una interfaz amigable para el usuario, que permitió la visualización en tiempo real de humedad y temperatura, y la operación a distancia del sistema de riego por goteo.

ABSTRACT: (Máximo 250 palabras)

This document details the design of a low-cost LPWAN network, based on LoRa modulation. This is achieved, using a Raspberry PI 4 together with the RAK2287 hub as a gateway. In the nodes, the TTGO T-Beam module based on the ESP32 microcontroller, and the Dragino LT-33222-L controller were used. The first node was responsible for sending humidity and temperature data. The second oversaw the activation of the irrigation system when the humidity requirements required it, and additionally, it allowed the filling of the water tank used in the crop.

For the tomato plantation, a two wooden garden was built of the necessary dimensions for its correct growth and development. The temperature and humidity were monitored in these wooden gardens, however, the

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA GESTIÓN
SERVICIOS BIBLIOTECARIOS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	4 de 4
--------	--------------	---------	---	----------	------	--------	--------

wooden garden 1 corresponded to the crop that implemented the drip irrigation system that worked under the ON / OFF control of the actuators, while the wooden garden 2 was the crop that was watered by manually by sprayer irrigation. Over 3 months, the crop presented numerous diseases and pests, which were treated naturally or with the use of pesticides. Using ChirpStack, a user-friendly interface was created, which allowed real-time display of humidity and temperature, and remote operation of the irrigation system.

APROBACION DE LA TESIS Nombre Jurado:

Yamil Armando Cerquera Rojas

Firma:

Nombre Jurado: Martín Diomedes Bravo Obando

Firma:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



Universidad Surcolombiana
Facultad de Ingeniería
Programa de Ingeniería Electrónica

**DESARROLLO DE UN GATEWAY LPWAN PARA EL CONTROL ON/OFF Y
MONITOREO DE UN CULTIVO DE TOMATE**

Modalidad:
Proyecto de grado

Línea de investigación:
Telecomunicaciones

2021



Universidad Surcolombiana

**DESARROLLO DE UN GATEWAY LPWAN PARA EL CONTROL ON/OFF Y
MONITOREO DE UN CULTIVO DE TOMATE**

Proyecto de grado para obtener el título de

Ingeniero Electrónico

Autor(es):

Nicolas Francisco Liscano Caldas

Luis Fernando Mosquera Imbachí

Director:

Jesús David Quintero Polanco

Ingeniero Electrónico y de Telecomunicaciones

Neiva – 2021

Agradecimientos

Agradezco a mis padres por el apoyo incondicional en esta etapa de mi vida. A mi tía, por hacer posible esta meta que hoy me encuentro cumpliendo. Y finalmente, al programa de Ingeniería Electrónica de la Universidad Surcolombiana, por brindarme todas las herramientas necesarias para culminar este proceso.

Nicolas Francisco Liscano Caldas

Agradezco a mi familia y mis amigos por brindarme su apoyo constante a lo largo de estos años, a mis compañeros de estudio, docentes y director Ing. Jesús David Quintero Polanco por proporcionar espacios para el aprendizaje, así como su conocimiento.

Luis Fernando Mosquera Imbachí

Resumen

En este documento se detalla el diseño de una red LPWAN de bajo costo, basada en la modulación LoRa. Este proyecto es fruto de una base de datos de proyectos que tiene el ingeniero Jesús David Quintero Polanco, docente de la Universidad Surcolombiana. Esto se logra, usando una Raspberry PI-4 junto con el concentrador RAK2287 como puerta de enlace. En los nodos, se utilizó el módulo TTGO T- Beam basado en el microcontrolador ESP32, y el controlador Dragino LT-33222-L. El primer nodo, fue el responsable de enviar datos de humedad y temperatura. El segundo se encargó, de accionar el sistema de riego cuando los requerimientos de humedad así lo requirieron, y adicionalmente, permitió el llenado del tanque de agua utilizado en el cultivo.

Para la plantación de tomate, se construyó dos huertas de madera de dimensiones necesarias para su correcto crecimiento y desarrollo. La temperatura y humedad fue monitoreada en dichas huertas, sin embargo la huerta 1 correspondió al cultivo que implementó el sistema de riego por goteo que funcionó bajo el control ON/OFF de los actuadores, mientras que la huerta 2 fue el cultivo que se regó de forma manual mediante riego con pulverizador. En el transcurso de 3 meses, el cultivo presentó ciertas enfermedades y plagas, que se trataron de manera natural o con el uso de pesticidas. Utilizando el servicio de ChirpStack, se creó una interfaz amigable para el usuario, que permitió la visualización en tiempo real de humedad y temperatura, y la operación a distancia del sistema de riego por goteo.

Abstract

This document details the design of a low-cost LPWAN network, based on LoRa modulation. This project is the result of a database of projects owned by the engineer Jesús David Quintero Polanco, a professor at the South Colombian University. This is achieved, using a Raspberry PI 4 together with the RAK2287 hub as a gateway. In the nodes, the TTGO T-Beam module based on the ESP32 microcontroller, and the Dragino LT-33222-L controller were used. The first node was responsible for sending humidity and temperature data. The second oversaw the activation of the irrigation system when the humidity requirements required it, and additionally, it allowed the filling of the water tank used in the crop.

For the tomato plantation, a two wooden garden was built of the necessary dimensions for its correct growth and development. The temperature and humidity were monitored in these wooden gardens, however, the wooden garden 1 corresponded to the crop that implemented the drip irrigation system that worked under the ON / OFF control of the actuators, while the wooden garden 2 was the crop that was watered by manually by sprayer irrigation. Over 3 months, the crop presented numerous diseases and pests, which were treated naturally or with the use of pesticides. Using ChirpStack, a user-friendly interface was created, which allowed real-time display of humidity and temperature, and remote operation of the irrigation system.

Tabla de Contenido

1. INTRODUCCIÓN	16
2. JUSTIFICACIÓN	17
3. ANTECEDENTES	18
4. OBJETIVOS	20
4.1 Objetivo General.....	20
4.2 Objetivos Específicos	20
5. MARCO TEÓRICO	21
5.1 <i>Técnicas de Modulación de Espectro Ensanchado</i>	21
5.1.1. <i>Frecuencia Modulada Pulsada</i>	22
5.1.2. <i>Modulación LoRa</i>	23
5.1.2.1. <i>Características intrínsecas</i>	23
5.1.2.2. <i>Factor de Dispersión (SF)</i>	24
5.2. <i>Diferencias entre LoRa y LoRaWAN</i>	25
5.3. <i>Tecnología LPWAN y por qué LoRa</i>	25
5.4. <i>Arquitectura de una Red LPWAN basada en el Protocolo LoRaWAN</i>	26
5.4.1. <i>Puertas de Enlace</i>	27
5.4.2. <i>Clases de Nodos</i>	28
5.4.2.1. <i>Clase A</i>	28
5.4.2.2. <i>Clase B</i>	29
5.4.2.3. <i>Clase C</i>	29
5.4.3. <i>Payload</i>	30
5.4.4. <i>Uplink y Downlink</i>	30
5.4.5. <i>Data Rate</i>	31
5.5. <i>Generalidades del cultivo de tomate</i>	31
5.5.1. <i>Requerimientos Edafoclimáticos</i>	31
5.5.2. <i>Nutrición en tomate</i>	32
5.5.3. <i>Enfermedades y Plagas comunes</i>	32
5.5.3.1. <i>Tizón Temprana</i>	33
5.5.3.2. <i>Mosquita Blanca de los Invernaderos</i>	33
5.5.3.3. <i>Polilla del Tomate</i>	33
5.6. <i>Sistema de Riego</i>	33

5.6.1. Riego por goteo	34
5.6.2. Riego manual con pulverizador.....	34
6. METODOLOGÍA.....	35
6.1. Dispositivos (Hardware y Software).....	35
6.2. Caracterización del Sensor Capacitivo de Humedad	45
6.3. Acondicionador de Señal.....	46
6.4. Configuración TTGO T-Beam.....	49
6.5. Geolocalización del proyecto.....	51
6.6. Configuración Dragino LT33222-L.....	52
6.7. Creación del Servidor LoRa.....	53
 6.7.1. Instalación del Gateway Bridge.....	53
 6.7.2. Instalación del Servidor de Red	54
 6.7.3. Instalación del Servidor de Aplicación.....	55
6.8. Integración con ChirpStack.....	56
6.9. Construcción del Gateway.....	59
6.10. Adaptación de Equipos y Sensores para Exteriores.....	65
6.11. Construcción de las Huertas.....	67
6.12. Control de los Actuadores Mediante Mensajes Downlink.....	72
 6.12.2. Integración con ThingsBoard.....	74
7. RESULTADOS	78
7.1. Cultivo de Tomate	78
7.2. Visualización de datos en ChirpStack	80
7.3. Integración con ThingsBoard.....	81
8. ANÁLISIS DE RESULTADOS	83
8.1. Promedio de Humedad y Temperatura.....	83
 8.1.1. Suministro de Agua	85
8.2. Nodo Clase A y Clase C.....	87
8.3. Comportamiento de los dispositivos LoRa.....	88
 8.3.1. Paquetes recibidos TTGO t-Beam	89
 8.3.2. Paquetes recibidos Dragino LT33222-L	89
 8.3.3. Indicador de Intensidad de Señal Recibida (RSSI).....	90
 8.3.4. Relación Señal a Ruido (SNR)	92
8.4. Efecto de la Distancia en los Parámetros de Comunicación.....	94

9. CONCLUSIONES.....	98
BIBLIOGRAFÍA.....	100
ANEXOS.....	103

Lista de Figuras

Figura 1 Comparación de una señal en banda estrecha con una señal modulada en secuencia directa	21
Figura 2 Modulación Chirp.....	22
Figura 3 Representación de las señales chirp que cambian linealmente cerca de una frecuencia central en la modulación LoRa.....	23
Figura 4 Características principales de las diferentes tecnologías LPWAN para IoT	26
Figura 5 Arquitectura de la red LoRa.....	27
Figura 6 Esquema representativo de la conexión del Gateway.....	27
Figura 7 Proceso de envío para nodo clase A.....	28
Figura 8 Proceso de envío para nodo clase B.....	29
Figura 9 Proceso de envío para nodo clase C.....	29
Figura 10 Estructura de paquetes LoRaWAN.....	30
Figura 11 Representación del riego por goteo.....	34
Figura 12 Raspberry Pi 4 4gb.....	35
Figura 13 Controlador de serie I/O Dragino LT33222-L	36
Figura 14 Módulo conversor adaptador Serial Usb Ttl CP2102.....	36
Figura 15 Módulo de desarrollo IoT inalámbrico.....	37
Figura 16 Concentrador LPWAN RAK2287	37
Figura 17 Módulo RAK2287 PI Hat	38
Figura 18 Sensor de humedad de suelo.....	38
Figura 19 Sensor de temperatura y humedad SHT10	39
Figura 20 Bomba de agua sumergible HJ-411	39
Figura 21 Válvula solenoide 12V 3/4	40
Figura 22 Fuente de alimentación utilizada	40
Figura 23 Sensor de nivel de líquido	41
Figura 24 Módulo regulador LM2596.....	41
Figura 25 Logotipo ChirpStack	42
Figura 26 Arquitectura de componentes conectado en ChirpStack.....	42
Figura 27 Esquema antes de la conversión.....	45
Figura 28 Divisor de tensión	46
Figura 29 Circuito acondicionador de señal	47
Figura 30 Esquema del circuito acondicionador de señal diseñado	48
Figura 31 Circuito acondicionador de señal construido	48
Figura 32 Circuito acondicionador de señal construido conectado al regulador de voltaje y sus baterías de alimentación.....	49
Figura 33 Identificación Dragino LT33222-L en ChirpStack	50
Figura 34 Ubicación del módulo en Google Maps de acuerdo con las coordenadas suministradas por la TTGO t-Beam.....	51
Figura 35 Conexión Dragino LT33222-L con Putty.....	52
Figura 36 Valores de configuración Dragino LT33222-L	52

Figura 37 Intervalo de transmisión	53
Figura 38 Verificación del funcionamiento de la instalación del Gateway Bridge	54
Figura 39 Verificación del funcionamiento de la instalación del Network Server.	55
Figura 40 Verificación del funcionamiento de la instalación del servidor de aplicaciones.....	56
Figura 41 Ventana de inicio de sesión en ChirpStack	56
Figura 42 Página de inicio del servidor de aplicación ChirpStack	57
Figura 43 Configuración del perfil en ChirpStack	57
Figura 44 Configuración del servidor de red en ChirpStack	58
Figura 45 Configuración del Gateway en ChirpStack	58
Figura 46 Ventana Device-profiles en ChirpStack	59
Figura 47 Perfil del dispositivo Dragino LT33222-L.....	59
Figura 48 Acople a la Raspberry Pi y vista interna del Gateway	60
Figura 49 Elaboración final del Gateway	60
Figura 50 Visualización de la línea de comando al ingresar a la configuración del Gateway	61
Figura 51 Opciones de configuración del Gateway	61
Figura 52 ID único del Gateway	62
Figura 53 Opciones de servidor LoRa	63
Figura 54 Elección del plan de configuración del Gateway	63
Figura 55 Opciones de configuración de Wifi	64
Figura 56 Información del Gateway suministrada en ChirpStack	64
Figura 57 Aislamiento e impermeabilización de los sensores de humedad capacitivos.....	65
Figura 58 Conexión entre el módulo TTGO t-Beam y el sensor SHT10, aislamiento e impermeabilización.....	65
Figura 59 Conexión sensores capacitivos y sensor SHT10 con la TTGO y el dispositivo acondicionador de señal.....	66
Figura 60 Impermeabilización de las conexiones electrónicas de la electroválvula	66
Figura 61 Impermeabilización de las conexiones electrónicas: TTGO y circuito acondicionador de señal	66
Figura 62 Huertas rectangulares para el cultivo de tomate (construidas a partir de madera)	67
Figura 63 Acople realizado para el riego de la huerta 1	68
Figura 64 Esquema de conexión válvula solenoide, sensor de nivel flotador y fuente de alimentación	69
Figura 65 Conexión de la válvula solenoide	69
Figura 66 Esquema de conexión bomba de agua y Dragino LT33222-L.....	70
Figura 67 Cultivo de tomate sembrado en el germinadero	70
Figura 68 Vista completa del cultivo sembrado para el desarrollo del proyecto ...	71
Figura 69 Plántulas trasplantadas a su respectiva huerta	71
Figura 70 Ubicación de sensores en las huertas.....	72

Figura 71 Configuración de parámetros para enlace Downlink	73
Figura 72 Creación del tipo de integración	74
Figura 73 Creación del decodificador de mensajes Uplink.....	74
Figura 74 Creación y definición de la cadena de reglas (rule chain)	76
Figura 75 Definición de los parámetros de control ON/OFF mediante el envío de mensajes downlink	76
Figura 76 Creación del mensaje de alarma.....	77
Figura 77 Crecimiento y floración de las plantas	78
Figura 78 Primeros frutos en las plantas	78
Figura 79 Efecto del tizón temprano en algunas plantas del huerto 1	79
Figura 80 Presencia de la mosquita blanca de los invernaderos en plantas del huerto 2	79
Figura 81 Presencia de larvas (Polilla en tomate) en planta del huerto 2	79
Figura 82 Esquema de conexión con ChirpStack.....	80
Figura 83 Registro de tramas, enlace LoRaWAN (ChirpStack)	81
Figura 84 Información detallada el estado activo del nodo TTGO en ThingsBoard	81
Figura 85 Implementación de widgets para la visualización de parámetros en ThingsBoard	82
Figura 86 Antenas de telecomunicación situadas alrededor de la localización del proyecto	92
Figura 87 Ubicación de los puntos de prueba en el municipio de Neiva, Huía. Colombia (Google Maps)	95

Lista de Gráficas

Gráfica 1 Promedio temperatura mensual registrada durante los meses de trabajo.	83
Gráfica 2 Promedio humedad mensual registrada durante los meses de trabajo	83
Gráfica 3 Humedad de Suelo: configuración Dragino en Clase A	84
Gráfica 4 Humedad de Suelo: configuración Dragino en Clase C	85
Gráfica 5 Consumo de agua promedio	86
Gráfica 6 Paquetes recibidos por la TTGO t-Beam	89
Gráfica 7 Paquetes recibidos Dragino	90
Gráfica 8 Promedio diario de RSSI durante los meses de trabajo.....	91
Gráfica 9 Promedio diario SNR durante los meses de trabajo	93
Gráfica 10 Parámetro RSSI y SNR vs Distancia	96

Lista de Tablas

Tabla 1 Descripción de costos de equipos y materiales utilizados en el desarrollo del proyecto	44
Tabla 2 Descripción de costos del software y/o aplicativos webs utilizado en el desarrollo del proyecto	45
Tabla 3 Representación del payload	50
Tabla 4 Registro de información suministrada por el módulo GPS de la TTGO t-Beam ...	51
Tabla 5 Direcciones de memoria Dragino LT33222-L	72
Tabla 6 Función Byte	73
Tabla 7 Cantidad de agua suministrada a la Huerta 1	86
Tabla 8 Cantidad de agua suministrada a la Huerta 2.....	86
Tabla 9 Tiempos de activación del sistema de riego	88
Tabla 10 Valor típico de RSSI en enlaces LoRaWAN	91
Tabla 11 SNR en función del SF en enlaces LoRaWAN.....	93
Tabla 12 Resultados obtenidos por el enlace LoRaWAN utilizando SF7	96
Tabla 13 Resultados obtenidos por el enlace LoRaWAN utilizando SF10.....	96

Lista de Anexos

Anexo A Código para realizar prueba del módulo GPS de la TTGO t-Beam	103
Anexo B Código ABP para la TTGO t-Beam	103
Anexo C Código Upnlink Dragino LT33222-L implementado en Chirpstack	108
Anexo D Código Uplink TTGO t-Beam implementado en Chirpstack.....	110

Lista de Símbolos y abreviaturas

Abreviatura	Termino
ABP	Activation By Personalization.
bps	Bits por segundo.
dB	Decibel.
dBm	Decibelio milivatio.
DHCP	Protocolo de Configuración Dinámica de Host
DNS	Sistema de Nombres de Dominio
EC	Concentrado Emulsionable
E/S	Entrada/Salida.
EU	Europeo
EUI	Extended Unique Identifier
GPS	Sistema de Posicionamiento Global
HF	Alta frecuencia.
ICM	Bandas Industriales, Científicas y Medicas
I/O	Input/Output.
IoT	Internet of things.
IP	Protocolo de Internet
IPV4	Protocolo de Internet Versión 4
ISO	Organización Internacional de Normalización
kbps	Kilo bytes por segundo.
LF	Baja frecuencia.
LoRa	Long Range.
LPWAN	Low Power Wide Area Network.
MHz	Mega Hertz.
mPCIe	Mini PCI Express
OTAA	Over the Air Activation.
RF	Radio frecuencia.
RH	Humedad relativa.
SD	Secure Digital
SO	Sistema Operativo.
SSH	Secure Shell.
TCP	Protocolo de Control de Transmisión.
US	Americano
VAC	Voltaje corriente alterna.
VDC	Voltaje corriente continua.
Wi-Fi	Wireless Fidelity
WG	Gránulos Dispersable en Agua

1. INTRODUCCIÓN

Durante estos últimos años y debido al gran desarrollo a nivel tecnológico, Internet de las cosas (IoT) se ha transformado en una de las tecnologías más importantes, permitiendo una fácil comunicación entre personas, procesos y dispositivos. Mediante informática de bajo costo, sistemas basados en la nube, big data e incluso cualquier tipo de tecnología móvil se puede adquirir y compartir información con mínimo esfuerzo humano.

Ahora, uno de los aspectos de las soluciones IoT el cual es evidentemente es la conectividad. En zonas donde existe una amplia gama de opciones para conectarse a Internet, los requerimientos de energía y alcance de la red tal vez no sean un problema. Sin embargo, donde la cobertura de comunicaciones es escasa o incluso inexistente y donde no siempre hay acceso a la energía eléctrica, las tecnologías que pueden ayudar a sobrellevar estas carencias son bienvenidas.

En este sentido, la tecnología LoRa soluciona tanto el alcance de la red, como la falta de energía eléctrica. Es que LoRa permite comunicar dispositivos hasta 30 kms en campo abierto, con un mínimo consumo de energía en los nodos remotos. LoRa es una tecnología de modulación del tipo spread spectrum (amplio espectro). Esto le permite tolerar ruido, caminos múltiples de señal y el efecto Doppler, mientras mantiene muy bajo el consumo de energía. El costo de lograr esto es el ancho de banda, que es muy bajo comparado con otras tecnologías inalámbricas. La tecnología LoRa está adquiriendo mucho impulso en su uso a través de LoRaWAN, que es un estándar de control de acceso al medio. Mediante esta especificación, es posible gestionar la comunicación de dispositivos inalámbricos que utilizan distintas tecnologías en la capa física, entre una de ellas LoRa.

2. JUSTIFICACIÓN

Los productos agrícolas de mayor exportación en el país son el café, el banano y las flores. Sin embargo, el país tiene toda la capacidad de ampliar este espectro de productos de exportación debido a sus ventajas en la agricultura, por tanto, es de suma importancia buscar nuevos métodos que permitan monitorear y administrar algunas o todas las variables a las cuales son sensibles cada tipo de cultivo, entre las cuales se pueden mencionar la humedad, la acidez del suelo, la temperatura, el clima, entre otras. Con el fin de resolver este problema se vienen estudiando diferentes sistemas de gestión para sistemas de riego, a los cuales no se les ha realizado gran difusión y su presencia en el sector agrícola es muy bajo comparado con las técnicas tradicionales que se siguen manejando, por esto es necesario un sistema que además de cumplir con su tarea principal sea adaptable a las necesidades del tipo de cultivo y de fácil operación para todo tipo de usuario con el fin de fomentar la tecnificación del campo en los pequeños productores.

Tradicionalmente en Colombia, la forma en que se mantienen los cultivos agrícolas pasa de generación en generación. Y son muy pocos los productores que hacen lecturas en tiempo real de las variables físicas más importantes para el correcto desarrollo de las plantaciones. Este problema es aún mayor en nuestro departamento donde tener acceso a este tipo de tecnología es más un privilegio que un estándar. Por este motivo el presente trabajo se centra en brindar solución a una problemática que involucra a gran parte del sector agrícola del país. Ya que, sus beneficios no se centran únicamente en mejorar la calidad de los cultivos, sino también ahorrar en recursos naturales como el agua.

3. ANTECEDENTES

Actualmente el agro colombiano tiene la difícil tarea de competir contra productores agrícolas internacionales que poseen mayor inversión en infraestructuras de automatización y control. Por ende, usualmente no pueden competir ni en precio ni en volumen contra dichas empresas. A pesar de esto, el grupo empresarial BIOS junto con otras pocas compañías colombianas son las únicas que van a la vanguardia en lo que automatización y asistencia técnica se refiere. Debido a que el sector agrícola no solo está en manos de grandes corporaciones, sino que también incluye campesinos con pequeñas y medianas aglomeraciones de tierra, nace la necesidad de mecanismos que hagan más eficiente la producción a un costo accesible.

Normalmente las variables físicas que más se utilizan en el sector agropecuario son la humedad, temperatura, Ph y radiación UV. Por lo tanto, es de vital importancia monitorear y controlar dichas variables para así disminuir costos a largo plazo atribuidos a la mala optimización de los recursos. Mediante el uso del espectro radio eléctrico es posible supervisar todas estas variables, sin embargo, debido a las grandes extensiones de tierra que abarcan los cultivos es sumamente costoso utilizar transmisores de radio convencionales, ya que la potencia necesaria para que la comunicación se realice de manera íntegra tiene que ser muy alta. Además, debido a la poca cobertura que tienen las redes 3G y 4G en entornos rurales no es viable el uso de esta tecnología. Internacionalmente hay ciertas bandas de uso libre denominadas ICM, que en su mayoría se utilizan en aplicaciones industriales, científicas, y médicas. Cada país tiene regulaciones propias en cuanto a que frecuencias y que potencia se puede utilizar en estas bandas.

Debido a todos estos inconvenientes nace LoRaWAN, que es una especificación para redes de baja potencia y área amplia, diseñada específicamente para dispositivos de bajo consumo de alimentación, que operan en redes de alcance local, regional, nacionales o globales (Technical Marketing Workgroup 1.0, 2015).

El estándar de red LoRaWAN apunta a requerimientos característicos de Internet de las Cosas, tales como conexiones bidireccionales seguras, bajo consumo de energía, largo alcance de comunicación, bajas velocidades de datos, baja frecuencia de transmisión, movilidad y servicios de localización. Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones locales complejas, y además otorga amplia libertad de uso al usuario final, al desarrollador y a las empresas que quieran instalar su propia red para Internet de las Cosas (Alliance, 2021).

La presente propuesta de grado toma como precedentes el trabajo realizado por Yi- Wei Ma y Jiann-Liang Chen en Toward Intelligent Agriculture Service Platform with LoRa-Based Wireless Sensor Network (Ma & Chen, 2018), el cual busca disminuir el fallo de las comunicaciones a grandes distancias, a la vez que se mantiene el consumo energético al mínimo. En este trabajo se implementa una red de sensores que permite crear una plataforma para desarrollar la denominada agricultura inteligente. Gracias a la infraestructura basada en LoRa es posible la integración de la agricultura al Internet de las Cosas.

Para la plantación y la elaboración del sistema de riego se tomó como referencia el “Manual de cultivo del tomate al aire libre” del Instituto de Desarrollo Agropecuario del gobierno de Chile (Guzmán L, y otros, 2017). El cual es un documento que contiene todos los requerimientos edafoclimáticos, de suelo y nutricionales del tomate. Además, brinda información de todo el contenido nutricional necesario para cada etapa de desarrollo, así como las principales enfermedades que afectan a la planta. Y finalmente, aporta información fundamental sobre el monitoreo y control de riego en este tipo de cultivos.

En el panorama nacional la tesis de grado “Prototipo de solución IOT con tecnología “LoRa” en monitoreo de cultivos agrícolas” (Triana Useche & Rodriguez Leguizamo, 2018). Aquí el objetivo es la adquisición de datos provenientes de los sensores de temperatura, humedad, radiación y Ph, que tienen comunicación directa con el sistema Arduino + LoRa shield, el cual envía los datos a través de comunicación LoRa hacia el respectivo Gateway (concentrador). Posteriormente el Gateway envía los datos de los nodos hacia un servidor, y mediante una aplicación web que posee una interfaz agradable, el usuario puede visualizar la información de los sensores de cada nodo, las estadísticas en tiempo real y alertas cuando las medidas no están dentro de los parámetros que se pueden establecer mediante el mismo aplicativo.

Y finalmente, desde nuestra al mater, “Implementación wplan para el envío de información remota en aplicaciones IOT” (Rivera Puentes, 2020) el cual se centra en la integración de LoRa a un sistema M-BUS para el envío de información a grandes distancias, con alta fidelidad y bajo consumo. Por lo tanto, nuestra propuesta busca evaluar el costo y la efectividad de una red LoRaWAN diseñada a partir de una Raspberry Pi 4 (Gateway), un controlador I/O, sensores, actuadores y una aplicación web, en una plantación casera de tomates. También va a ser objeto de estudio la aplicación de esta red en una plantación de cielo abierto en un entorno rural.

4. OBJETIVOS

4.1 Objetivo General

- Diseñar e implementar una red basada en LoRaWAN para el monitoreo de temperatura, humedad del aire y suelo de un cultivo de tomate cuyos actuadores son accionados mediante un aplicativo web.

4.2 Objetivos Específicos

- Elaborar dos huertas caseras para el cultivo de tomate al aire libre con su sistema de riego (por goteo y manual) respectivamente, para el abono e irrigación del cultivo.
- Implementar actuadores necesarios para realizar el control ON/OFF de la humedad del suelo mediante el sistema de riego por goteo.
- Establecer comunicación entre los nodos y el Gateway utilizando diferentes factores de dispersión, y distintas clases (Clase A y Clase C).
- Adquirir y procesar datos mediante una aplicación web para realizar el monitoreo de las variables del cultivo y control de los actuadores.
- Comparar el cultivo donde se llevó a cabo el control del sistema de riego por goteo, con otra plantación tratada con riego manual.

5. MARCO TEÓRICO

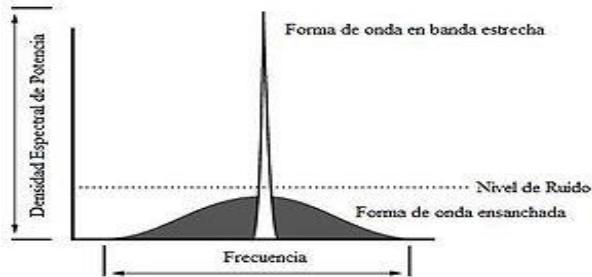
5.1 Técnicas de Modulación de Espectro Ensanchado

El Espectro Ensanchado se basa en la expansión del espectro de la señal a transmitir mediante el uso de secuencias ortogonales. Lo que hace posible que el receptor solo pueda demodular la señal si conoce la secuencia que se ha utilizado en su expansión; esto implica que los demás transmisores que utilicen la misma banda ven la señal básicamente como ruido.

Los sistemas de comunicación de espectro ensanchado son útiles para suprimir la interferencia, lo que dificulta la detección y la seguridad de las comunicaciones proceso, acomodando canales de desvanecimiento y multirayecto, y proporcionando capacidad de acceso múltiple. A pesar de que las técnicas de espectro ensanchado en un principio se utilizaron para fines militares aproximadamente por medio siglo, actualmente se ha reconocido que pueden proporcionar características que mejoran los sistemas de radioenlace; a continuación se presenta la Figura 1, una comparativa entre una señal de banda estrecha frente a una señal modulada en secuencia directa.

Figura 1

Comparación de una señal en banda estrecha con una señal modulada en secuencia directa



Nota: La señal en banda estrecha se suprime al transmitir el espectro ensanchado. Adaptada de Espectro ensanchado, 2021 (https://es.wikipedia.org/wiki/Espectro_ensanchado) CC BY 3.0

No existe una barrera teórica fundamental para la efectividad de las comunicaciones de amplio espectro. Este hecho notable no es inmediatamente evidente ya que el mayor ancho de banda de una señal de espectro extendido necesita un filtro de recepción que pasa más potencia de ruido al demodulador. Sin embargo, cuando cualquier señal y blanco el ruido gaussiano se aplica a un filtro adaptado a la señal, la salida del filtro muestreado tiene una relación señal-ruido que depende únicamente de la relación energía-densidad-ruido. Por lo tanto, el

ancho de banda de la señal de entrada es irrelevante y las señales de espectro ensanchado no tienen limitaciones inherentes.

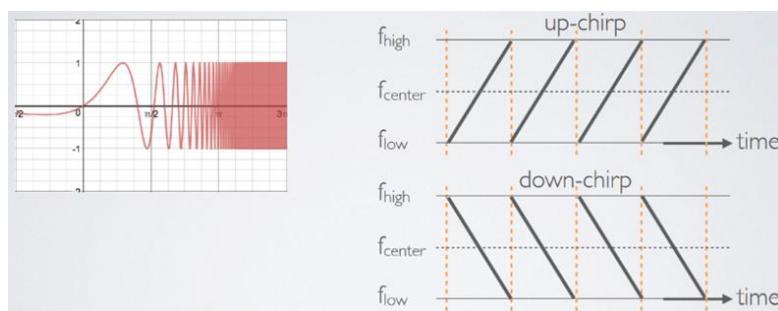
La modulación implica la adición directa de una secuencia de expansión de alta velocidad a la secuencia de código-símbolo de menor velocidad, lo que da como resultado una señal transmitida con un ancho de banda amplio. La eliminación de la secuencia de propagación en el receptor provoca una contracción del ancho de banda que se puede explotar aplicando un filtrado adecuado para eliminar una gran parte de la interferencia.

5.1.1. Frecuencia Modulada Pulsada

Chirp Spread Spectrum se desarrolló para aplicaciones de radar en la década de 1940. Usado tradicionalmente en un número de aplicaciones de comunicaciones militares y seguras; durante los últimos veinte años esta modulación la técnica ha experimentado una mayor adopción en una serie de aplicaciones de comunicaciones de datos debido a sus requisitos de potencia de transmisión relativamente bajos y robustez inherente a la degradación del canal. Mecanismos tales como interferencias de interferencia multirayos, desvanecimiento, Doppler y en banda.

Figura 2

Modulación Chirp



Nota: Los llamados “chirridos” hace referencia una rampa con valores flow a fhigh (chirrido hacia arriba) o de fhigh a flow (chirrido hacia abajo). Adaptado de LoRa, 2018 (<https://lora.readthedocs.io/en/latest/>) CC BY 3.0

La Figura 2 representa el método Chirp Spread Spectrum, el cual, corresponde a una técnica de espectro ensanchado la cual implementa pulsos llamados “chirp” modulados de frecuencia lineal de banda ancha para conseguir codificar la información.

El IEEE adoptó un CSS PHY para las redes de área personal inalámbricas de baja velocidad (LR-WPAN) estándar 802.15.4 para aplicaciones que requieren mayor alcance y movilidad que el que se puede lograr con el modo OQPSK DSSS PHY.

5.1.2. Modulación LoRa

Long Range Modulation o simplemente LoRa es una modulación de espectro expandido que tiene origen de la modulación de tipo Chirp. Diseñada para tener baja transferencia de datos, alta cobertura y bajo consumo energético, implementa tasas de trasferencias ajustables utilizando un factor de ensanchamiento ortogonal (SF) permitiendo que el usuario escoja transferencias de datos específicas de acuerdo con el rango o consumo, manteniendo el ancho de banda constante con el fin de optimizar el rendimiento de la red.

LoRa fue patentada por Semtech y corresponde a la capa física de LoRaWAN, una tecnología emergente.

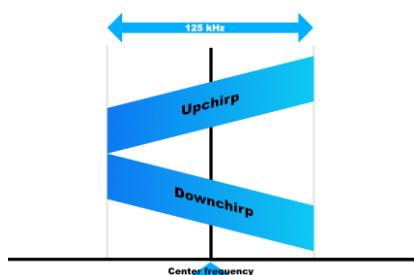
5.1.2.1. Características intrínsecas

LoRa utiliza una técnica llamada Chirp o también conocida como Espectro ensanchado por Chirp (CSS). Esta técnica modula el mensaje por medio de una señal de “chirrido” que varía continuamente en frecuencia. Como se mencionó anteriormente inicialmente fue desarrollada para aplicaciones de radares alrededor del año 1940, por lo que al principio era utilizada para fines militares.

Figura 3

Representación de las señales chirp que cambian linealmente cerca de una frecuencia

central en la modulación LoRa



Nota: Comportamiento señal chirp en la modulación LoRa. Adaptado de Inmunidad al efecto Doppler en enlace satelital (LEO) con modulación LoRa, 2018, (<https://medium.com/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora-652c6880e30c>) CC BY 3.0

Debido a que, en CSS los desplazamientos de tiempo y frecuencia terminan siendo equivalentes entre el transmisor y receptor, la complejidad del diseño del receptor se reduce en gran medida. Y se debe tener en cuenta que el ancho de banda de frecuencia de este “chirrido” es equivalente al ancho de banda espectral de la señal LoRa modulada, obsérvese la Figura 3.

Actualmente debido a su robustez dicha técnica sobresale frente a otras técnicas de espectro ensanchado como por ejemplo el Espectro ensanchado por secuencia directa (DSSS); puesto que no requiere insertar ningún elemento de pseudoruido para ayudar al receptor a distinguirlo del ruido de canal, por lo que es más resistente frente a factores de ruido como el “efecto Doppler” (típico en aplicaciones de radio). Como requiere un bajo nivel de potencia de transmisión, esta técnica se ha vuelto popular en dispositivos que funcionen con baterías y que su aplicación tenga como objetivo la confiabilidad de transmisión y el bajo consumo de energía.

5.1.2.2. Factor de Dispersión (SF)

Corresponde a uno de los parámetros importantes que influye sobre el rendimiento físico, hace referencia a la relación entre el ancho de banda de la señal y la velocidad de símbolos, permite establecer: tasa de datos, la banda de envío y la distancia óptima de transmisión. Es una de las características ajustables dentro de las tecnologías y redes enfocadas a LoRa/LoRaWAN.

Se debe tener claro que un factor con mayor difusión genera que la tasa de bits se reduzca por lo que la duración de la batería también disminuye y esto provoca un incremento en el tiempo de transmisión.

De acuerdo con Semtech (dueños de la patente de LoRa) en su documento titulado “LoRa Modulation Basics”, la tasa de bits se define a partir de un factor de dispersión y el ancho de banda:

$$R_b = SF \times \frac{1}{2^{SF}} \frac{\text{bits/sec}}{\text{BW}}$$

Donde:

- SF corresponde al factor de dispersión.
- BW corresponde al ancho de banda.

LoRa permite seis factores de dispersión: desde el valor mínimo 7 hasta el valor máximo 12 es decir (SF7 – SF12), el factor de dispersión también determina la relación señal ruido por lo que afecta también en la sensibilidad del receptor.

5.2. Diferencias entre LoRa y LoRaWAN

Con la llegada de la cuarta revolución industrial se da origen al internet de las cosas o de forma abreviada IoT, esta tecnología necesita que las redes de comunicación presenten gran eficiencia y LoRaWAN es una de estas. LoRaWAN funciona mediante la tecnología de modulación que presenta LoRa, establecida específicamente para IoT y eso que no es la única puesto que existen más como lo es el caso de Sigfox.

Por lo tanto, al hablar de LoRa y LoRaWAN no se refiere a lo mismo. En pocas palabras LoRa es el tipo de modulación en radiofrecuencia que fue patentando por Semtech mientras que LoRaWAN es el protocolo de red que hace uso la tecnología LoRa para establecer la comunicación y administración dispositivos LoRa los cuales se componen de principalmente Gateway (puertas de enlace) y Nodos.

Una red de tipo LoRa tiene la finalidad de conectar objetos y su tecnología de modulación permite reducir costos y que funcione con un consumo mínimo de energía de los dispositivos conectados, por lo que el tiempo de duración de la batería de estos es amplio (entre 3 y 5 años). La señal que emite la red LoRa cuenta con la capacidad de atravesar objetos (edificios, montañas, entre otros) así como llegar a lugares profundos como sótanos y garajes, esto es resulta ser muy útil porque en aplicaciones industriales permite localizar y posicionar dispositivos de gran valor en áreas de difícil acceso.

Por otra parte, LoRaWAN es un protocolo de control de acceso a medios (MAC) para redes de área amplia. Y de acuerdo con (The Things Network, 2021) “está implementado sobre la modulación LoRa o FSK en bandas de radio industriales, científicas y médicas (ISM). Y sus protocolos están definidos por LoRa Alliance y formalizados en la Especificación LoRaWAN los cuales se pueden descargar en el sitio web de LoRa Alliance”.

El protocolo LoRaWAN no es adecuado para ser usado siempre o en todos los casos puesto que posee ciertas limitaciones, (The Things Network, 2021) presenta los casos útiles o no de este protocolo.

5.3. Tecnología LPWAN y por qué LoRa

En los últimos años se ha vuelto común mencionar el término “redes LPWAN” (Low-Power Wide Area Network) o redes de bajo consumo y área extensa, esto hace referencia a un grupo de redes de distinta denominación, en donde cada una implementa diferentes frecuencias en su transmisión y emplea estándares propios. Este tipo de tecnologías facilita el desarrollo en zonas donde la cobertura de comunicación sea escasa o incluso nula y no se cuente regularmente con acceso a energía eléctrica.

Para este caso práctico y de acuerdo con los requerimientos de una red de comunicación IoT, es decir baja velocidad de datos, baja frecuencia de transmisión, movilidad y servicios de localización, conexiones bidireccional segura, bajo consumo de energía y largo alcance de comunicación, el factor económico en sus dispositivos y el hecho de que la tecnología LoRa/LoRaWAN pueda ser utilizada por clientes que no son operadores móviles y teniendo a ChirpStack como iniciativa para creación de este tipo de red, se optó por implementar este tipo de tecnología en este proyecto, ya que fue la opción que más se adaptó a nuestra necesidad.

A continuación la Figura 4 presenta las características de las diferentes tecnologías LPWAN para IoT:

Figura 4

Características principales de las diferentes tecnologías LPWAN para IoT

SIGFOX		LoRa		Short-range		NB-IoT Rel. 13		eMTC Rel. 13		EC-GSM Rel. 13		5G (under definition) 5G	
SIGFOX													
Range MCL	<12km 160 dB	< 10km 157 dB		10cm to 200m		<15km 164 dB		< 10km 156 dB		< 15km 164 dB		<15km 164 dB	
Spectrum Bandwidth	Unlicensed 900MHz 100Hz	Unlicensed 900MHz <500kHz		Unlicensed 2.4 GHz		Licensed IMT 200 kHz shared		Licensed IMT 1.4 MHz shared		Licensed 8-900MHz shared		Licensed IMT shared	
Data rate	<100 bps	<10 kbps		<100s Mbps		<67 kbps		<1 Mbps		<70 kbps		<1 Mbps	
Use case	Smart Grid/City/ Monitoring	Smart Grid / City/ Monitoring		Smart home/factory		Smart Grid/City/ Monitoring		Smart Grid / City / Monitor./ vehic.		Smart Grid / City / Monitor./ vehic.		Smart Grid / City / Monitor./ vehic.	
Module cost	4.00\$ (2015) 2.64\$ (2020)	4.00\$ (2015) 2.64\$ (2020)		Not available		4\$ (2016) 2-3\$ (2020)		5.00\$ (2016) 3.30\$ (2020)		4.55 (2016) 2.97\$ (2020)		<\$2	
Network cost, US example* (cost drivers)	\$10/year/km ² >\$80M/year (HW+SW+Service)	\$10/year/km ² >\$80M/year (HW+SW+Service)		Not available		\$1/year/km ² <\$7M/year (SW upgrade)		\$1/year/km ² <\$7M/year (SW upgrade)		\$1/year/km ² <\$7M/year (SW upgrade)		Included in 5G deployment	

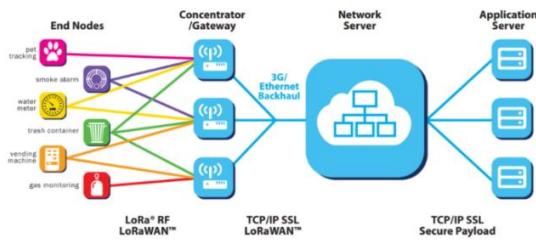
Nota: Adaptado de IoT (Internet-of-Things) – El Internet de las “Cosas”, 2018, (<http://micm.es/noticias/iot-el-internet-de-las-cosas/>) CC BY 3.0

5.4. Arquitectura de una Red LPWAN basada en el Protocolo LoRaWAN

La red LPWAN está basada en la topología tipo estrella como se observa en la Figura 5. Muchos dispositivos finales, que comúnmente se conocen como nodos, se comunican de forma directa con el dispositivo central o Gateway, el cual brinda las funciones de enrutamiento para direccionar la información entregada por los nodos, a una estación base o servidor (en este caso práctico), dicha estación o servidor ofrece la capacidad al usuario de monitorear, instrumentar, y ejecutar acciones (en tiempo real) conforme a los eventos que se estén presentando.

Figura 5

Arquitectura de la red LoRa



Nota: Corresponde a una topología estrella-estrella que facilita retransmitir datos a un servidor central a través de puertas de enlace (Gateways). Adaptado de Tecnología LoraWan, 2019, (<http://www.dset-energy.com/2019/07/12/descripcion-lorawan/>) CC BY 3.0

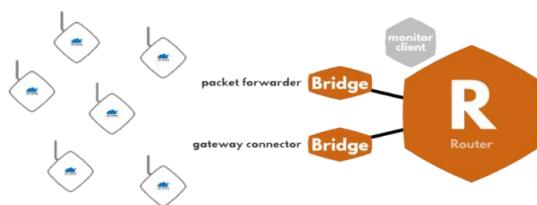
5.4.1. Puertas de Enlace

El Gateway establece una especie de puente entre los dispositivos y ChirpStack, dichos dispositivos implementan la red de baja potencia como LoRaWAN para así conectarse al Gateway, entretanto este último (Gateway) hace uso de redes alto ancho de banda como lo es, Ethernet, celular o red WiFi para establecer conexión con ChirpStack. Cada una de las puertas de enlace que esté cerca de un dispositivo va a recibir los mensajes de este y serán reenviados a ChirpStack.

Cabe mencionar que un solo Gateway puede ofrecer el servicio a múltiples dispositivos como se observa en la Figura 6.

Figura 6

Esquema representativo de la conexión del Gateway



Nota: El Gateway corresponde a un enrutador equipado con un concentrador LoRa , lo que le permite recibir paquetes LoRa. Adaptado de Gateways, 2021, (<https://www.thethingsnetwork.org/docs/gateways/index.html>) CC BY 3.0

Actualmente se pueden encontrar dos tipos de Gateway, el primer tipo es el Gateway que puede ejecutarse sobre un firmware mínimo por lo que genera que sea de bajo costo y fácil de implementar haciendo uso solo del software para el reenvío de paquetes de información. Un ejemplo de este tipo de puerta de enlace es el The Things Gateway. El segundo tipo es el Gateway que ejecuta un sistema operativo, aquí el software de reenvío de paquetes de información funciona como un programa que se mantiene en segundo plano por lo que brinda que el usuario del Gateway cuente con mayor libertad para instalar y ejecutar su propio software, así como administrar su propia puerta de enlace.

Puesto que LoRaWAN funciona en bandas sin licencia, en la mayoría de los países se puede ejecutar el Gateway ya que es legal, aunque como cada país tiene sus propias normas puede existir cierta restricción a nivel de instalación de antenas, entre otros.

5.4.2. Clases de Nodos

Existen tres clases de dispositivos que se denominan nodos, estos permiten que se efectúe la comunicación bidireccional y están clasificados en Clase A, Clase B y Clase C. Estos dos últimos son extensiones de la especificación de los dispositivos de Clase A.

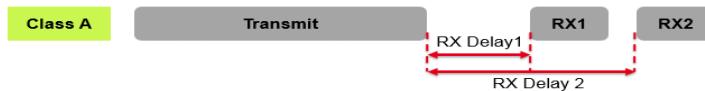
5.4.2.1. Clase A

El dispositivo LoRaWAN Clase A consume menos energía y permite comunicación bidireccional entre un dispositivo y una puerta de enlace, es de tipo asincrónico. El envío de mensajes de enlace ascendente es decir desde el dispositivo hasta el servidor pueden ser enviados en cualquier momento.

El proceso de envío de mensajes se representa en la Figura 7.

Figura 7

Proceso de envío para nodo clase A



Nota: Despues de que el terminal envía una señal de transmisión de enlace ascendente, el servidor puede comunicarse con el enlace descendente muy rápidamente. Adaptado de Performance Evaluations of LoRa Wireless Communication in Building Environments (p. 4), por Liang, Liang Zhao, Peng Wang, 2020, Sensors

El nodo final permite la comunicación desde el servidor mediante dos breves ventanas de recepción las cuales se abren por un corto período de tiempo luego realizar la transmisión de enlace ascendente (envío de información enviada desde una estación en tierra hasta un receptor, para este caso es desde el nodo hacia la puerta de enlace). Cabe mencionar que el servidor puede responder en la primera o segunda ventana, pero no puede hacer uso de ambas al mismo tiempo.

5.4.2.2. Clase B

Amplían la Clase A al agregar ventanas de recepción extra programadas para los mensajes de enlace descendente desde el servidor. Este nodo utiliza un contador sincronizado en el tiempo transmitido por el Gateway por lo que los dispositivos abren de forma periódica ventanas de recepción.

El proceso de envío de mensajes se representa en la Figura 8.

Figura 8

Proceso de envío para nodo clase B



Nota: Adaptado de Performance Evaluations of LoRa Wireless Communication in Building Environments (p. 4), por Liang, Liang Zhao, Peng Wang, 2020, Sensors

5.4.2.3. Clase C

Este dispositivo al igual que el anterior amplían la Clase A proporcionando ventanas de recepción que permanecen casi continuamente abiertas, dichas ventanas se cierran solo durante la transmisión de punto final y permite comunicación de baja latencia, pero en consecuencia consume más energía que el dispositivo de Clase A. Este dispositivo es adecuado cuando se necesita recibir gran cantidad de datos. El proceso de envío de mensajes se representa en la Figura 9.

Figura 9

Proceso de envío para nodo clase C



Nota: Adaptado de Performance Evaluations of LoRa Wireless Communication in Building Environments (p. 4), por Liang, Liang Zhao, Peng Wang, 2020, Sensors

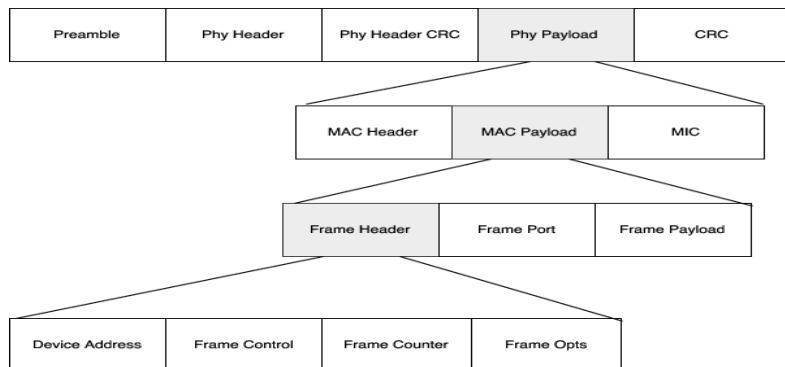
5.4.3. Payload

Hace referencia al mensaje que se quiere transmitir y se obtiene excluyendo el resto de la trama que incorpora el protocolo LoRaWAN. En esencia son los datos útiles que requiere la aplicación final del sistema (SCADA, BMS, GMAO, etc.). Se debe tener en cuenta que la velocidad de datos junto con el tamaño máximo de paquete para carga útil (payload) de la aplicación, dependerá aproximadamente de la distancia a la que se encuentre la puerta de enlace más cercana, así como también del tipo de datos que serán enviados.

A continuación se presenta la estructura de paquetes LoRaWAN:

Figura 10

Estructura de paquetes LoRaWAN



Nota: La carga útil de la capa física está formada por encabezados de capa MAC, encabezados de trama, carga útil y código de integridad del mensaje. Adaptado de Exploring The Security Vulnerabilities of LoRa, 2017, (https://www.researchgate.net/publication/318575428_Exploring_the_Security_Vulnerabilities_of_LoRa#pf4) CC BY 3.0

5.4.4. Uplink y Downlink

Estos términos hacen referencia a la dirección del mensaje, concretamente cuando se habla de mensajes Uplink, se indica que es el nodo quien envía un mensaje hacia el Gateway. Por otra parte, en los mensajes de tipo Downlink es el Gateway el que enviará la información hacia el nodo; la Figura 11 refleja gráficamente este proceso.

5.4.5. Data Rate

LoRaWAN emplea un esquema Adaptive Data Rate (ADR) o Tasa de datos adaptable que optimiza dinámicamente la velocidad de datos, el tiempo aire y el consumo de energía.

La adaptación de la velocidad de datos en un enlace LoRaWAN permite fácil escalabilidad de la red mediante la adición de Gateways. Además, el uso de ADR produce que se aumente de forma significativa la capacidad de dicha red, puesto que paquetes de datos que se transmiten implementando diferentes factores de dispersión (SF) que son ortogonales y pueden transmitirse de forma simultánea.

Sin embargo, este algoritmo debe implementar con cuidado pues la probabilidad de colisión, que influye directamente en el rendimiento del radio enlace, se ve afectada por el cambio en SF.

5.5. Generalidades del cultivo de tomate

El tomate llamado de forma científica *Solanum lycopersicum*, es una planta herbácea y que forma parte de la familia Solanaceae. Su origen se ubica en los Andes sudamericanos.

Es una hortaliza que posee gran diversidad genética por lo que se puede encontrar en gran variedad, que se distinguen de acuerdo con su aspecto, color y sabor. De acuerdo con la conducta en su crecimiento, el tomate puede clasificarse en crecimiento determinado y crecimiento indeterminado, en el primer grupo se encuentran aquellas en las que el tallo termina en ramillete floral el cual indica que en esa parte concluye el crecimiento. En el segundo grupo están aquellas en las que el ápice que es localizado en la parte extrema de su tallo crece de forma indefinida. Las de crecimiento indeterminado son de mayor uso a nivel comercial por lo que ofrecen rendimientos que pueden llegar a 200 t/ha (tonelada por hectárea) incluso en condiciones adversas (Guzmán L, y otros, 2017).

5.5.1. Requerimientos Edafoclimáticos

Suelo

La habilidad del tomate para permanecer en condiciones adversas de crecimiento hace que la planta no presente características exigentes frente a las condiciones de suelo, pero es necesario contar con un buen drenaje. Cabe mencionar que se recomienda un suelo con alto contenido de materia orgánica y para el caso de suelos arcillosos y arenosos esta planta puede desarrollarse con tan solo de 40 cm de profundidad.

El acondicionamiento de la tierra debe comprometer acciones de labranza que permitan un suelo óptimo para instalación de plántulas y así obtener excelente calidad de suelo para favorecer el crecimiento y desarrollo de sus raíces.

Clima

Debido a la habilidad para permanecer en condiciones adversas de crecimiento, esta planta puede adecuarse a múltiples condiciones climáticas que van desde templadas hasta tropicales, cálidas y húmedas.

El tomate forma parte del grupo de las especies de estación cálida y la temperatura para que se desarrolle de forma eficiente se ubica en un rango entre 18 y 30°C, esta es la principal razón por la que en climas templados se realiza su cultivo al aire libre.

Según (Guzmán L, y otros, 2017) con relación a la humedad relativa, para que se desarrolle el tomate necesita que ésta varíe en un rango entre 60 y 80%, teniendo en cuenta que en aquellos lugares donde se tiene presencia de humedades relativas altas se favorece el aumento de enfermedades provocadas por hongos y baterías las cuales impiden la fecundación. En periodos de estrés hídrico con niveles excesivos de humedad en el suelo provocados por el riego se ocasiona un agrietamiento del fruto o lo que se denomina “rajado” (p.17).

La cantidad de luz solar que recae sobre el cultivo de tomate cumple un rol importante que incide en crecimiento vegetativo de la planta, ya que el tomate necesita cerca de 6 horas al día de luz directa para florecer.

5.5.2. Nutrición en tomate

Existen variedad de elementos que son considerados indispensables para obtener buen crecimiento y gran producción de todas las especies cultivadas, entre esas el tomate. A nivel de raíz, tallo, hoja y fruto se tiene que el oxígeno, el hidrógeno y el carbono son esenciales para su desarrollo. Mediante el sistema de riego que de forma constante proporciona agua al cultivo se puede obtener el hidrógeno y oxígeno, mientras que, a través del proceso de fotosíntesis el carbono obtenido por medio de la atmósfera es transformado en carbohidratos.

Cuando la planta requiere y el suelo suministra una cantidad baja de los nutrientes ya mencionados, se debe tener en cuenta que es necesario agregar mediante el proceso fertilización aquellos elementos indispensables para su crecimiento y desarrollo. Aprender sobre del comportamiento nutricional, permite llevar de forma óptima todo el proceso de fertilización, como consecuencia mayor eficiencia y mejor la calidad del fruto y evitar que la planta se desarrolle de forma excesiva.

5.5.3. Enfermedades y Plagas comunes

Para que se inicie una enfermedad es necesario que se presente un ambiente favorable, un agente causal y un huésped apto de forma simultánea; puesto que de lo contrario no podrá generarse dicha enfermedad, si bien, para el tomate el agente causal puede definirse como un hongo o una bacteria que afecta a la planta en sus

diferentes formas de desarrollo y pueden generar daños en el rendimiento, así como su condición comercial.

Cualquier tipo de enfermedad origina síntomas o indicios los cuales pueden reconocerse rápidamente en algunos casos. Cuando el cultivo de tomate se encuentra al aire libre es atacado por enfermedades que provocan una disminución en su nivel de producción, la forma en que estas influyen sobre el cultivo depende del agente causal, del mismo ambiente y del estado en que se encuentre la planta.

A continuación, se presentan las principales enfermedades y plagas que afectan al tomate al aire libre:

5.5.3.1. *Tizón Temprana*

Esta es una de las enfermedades más comunes e importantes que se genera en el cultivo de tomate durante todas las etapas de crecimiento, es provocada por el hongo *Alternaria solani*. Este hongo tiene la capacidad de deteriorar paulatinamente la planta y presenta síntomas que son determinados por manchas circulares de color café en aquellas hojas que están maduras y en ocasiones cuando las circunstancias suelo – climáticas son favorables o se vinculan con otras enfermedades pueden involucrar gran parte de la planta y/o cultivo, dichas manchas o lesiones pueden ser vistas en lugares como tallos y flores (ocasionando su caída).

5.5.3.2. *Mosquita Blanca de los Invernaderos*

La mosquita blanca de los invernaderos ataca en los brotes recién formados, allí las hembras depositan los huevos y el daño directo es provocado por la succión de la savia, y en casos severo debilita la planta.

Según (Guzmán L, y otros, 2017) “El daño indirecto es debido a la abundante mielecilla que la plaga produce, lo que favorece el desarrollo de fumagina en hojas y frutos. El monitoreo de esta plaga se debe hacer durante el desarrollo de todo su ciclo”. Para cultivos al aire libre se sugiere el uso de las llamadas trampas amarillas al principio del cultivo con la intención de localizar a los adultos.

5.5.3.3. *Polilla del Tomate*

Las larvas son el principal agente causante del daño en el cultivo, estas se introducen en los frutos, hojas y tallos alimentándose del tejido interno causando así perforaciones y galerías transparentes. En consecuencia, produce una disminución en el valor comercial del fruto obtenido en el cultivo afectado.

5.6. *Sistema de Riego*

De acuerdo con (Orson W & Vaughn E, 2003), el riego se define como “La aplicación artificial de agua al terreno con el fin de suministrar a las especies vegetales la humedad necesaria para su desarrollo” (p.10). Teniendo en cuenta que dicha aplicación de agua se relaciona directamente con el objetivo de refrigerar el

suelo y la atmósfera, mejorando así las condiciones ambientales para el crecimiento vegetal. Para los efectos prácticos se consideran dos sistemas en específicos, definidos en función del manejo eficiente del agua.

5.6.1. Riego por goteo

El sistema de riego por goteo es quizás el método más eficiente para el suministro de agua y nutrientes en la mayoría de los cultivos agrícolas; mediante una red de tuberías se conduce el agua y de forma directa es aplicada o entregada periódicamente y en pequeños volúmenes de agua (por medio de goteros) en las diferentes zonas radiculares del cultivo tal como se muestra en la Figura 11.

Figura 11

Representación del riego por goteo



Nota: Adaptado de Los sistemas de riego más recomendados para cada tipo de cultivo, 2020,

(<https://citi-sa.com/los-sistemas-de-riego-mas-recomendados-para-cada-tipo-de-cultivo/>) CC

BY 3.0

“Desde el punto de vista agronómico, se denominan riegos localizados porque humedecen un sector de volumen de suelo, suficiente para el buen desarrollo del cultivo. También se le denomina de alta frecuencia, lo que permite regar desde una a dos veces por día, todos o algunos días, dependiendo del tipo de suelo y las necesidades del cultivo” (Liotta , Carrión , Ciancaglini, & Olguín, 2015).

5.6.2. Riego manual con pulverizador

Este sistema de riego tradicional consiste en la aplicación de agua en forma de rocío a tallos y hojas del cultivo, mediante un recipiente con pulverizador. La aplicación de este método implica un leve aumento en el consumo de agua debido las pérdidas por evaporación, como también a la frecuencia con que debe realizarse; además el depósito de agua en hojas puede provocar la proliferación y transporte de patógenos en el cultivo.

6. METODOLOGÍA

Para el desarrollo simplificado del proyecto se plantearon los siguientes pasos:

6.1. Dispositivos (Hardware y Software)

Este equipo fue necesario para la elaboración y funcionamiento del proyecto. A continuación, se presenta la descripción de todos los dispositivos implementados:

Raspberry Pi 4 4gb:

Figura 12

Raspberry Pi 4 4gb



Nota: Adaptado de Elecrow – Raspberry PI 4 Model, 2018, (https://www.elecrow.com/raspberry-pi-4-model-b-4gb-ram.html?gclid=CjwKCAjwmqKJBhAWEiwAMvGt6FUbXbyxdz7hzBEgllkYiIuDQ7k-jRjnWQSUI5x3d6O6Uns7YNHRoC604QAvD_BwE) CC BY 3.0

La Figura 12 representa la Raspberry Pi, una placa ordenador de bajo costo (basadas en hardware y software libre) que brinda un formato compacto y sinfín de funcionalidades. Debido a que cuenta con facilidad para usarse y adaptarse, está destinada para el desarrollo de proyectos y prototipos electrónicos.

Controlador de serie I/O Dragino LT33222-L

Estos dispositivos I/O pertenecientes a la serie Dragino LT son módulos controladores I/O LoRa de gran alcance. Cuentan con diferentes interfaces de I/O como lo es la entrada de corriente analógica, una entrada de voltaje analógico, salida de relé, entrada y salida digitales, etc. Además, estos módulos I/O LT están diseñados para simplificar la instalación de monitoreo de I/O.

A continuación se presenta una imagen del controlador Dragino LT33222-L:

Figura 13

Controlador de serie I/O Dragino LT33222-L



Fuente: Autores

Destinado para aplicaciones profesionales basadas en redes de señales inalámbricas como por ejemplo sistemas de riego, ciudades inteligentes, automatización, entre otros. Estos dispositivos son de fácil instalación y costo bajo debido al uso de la tecnología LoRa.

Módulo conversor TTL CP2102

Este módulo facilita la comunicación entre el PC y un microcontrolador utilizando el protocolo USB, es decir, permite conectar un microcontrolador con puerto UART al pc. Este dispositivo es compatible con cualquier microcontrolador como por ejemplo PIC, Atmel AVR, ESP8266, ESP32 entre otros.

Figura 14

Módulo conversor adaptador Serial Usb Ttl CP2102



Fuente: Autores

Tarjeta WiFi, Bluetooth, GPS, LoRa LILYGO TTGO t-Beam

A continuación se presenta una imagen de la tarjeta TTGO t-Beam:

Figura 15

Módulo de desarrollo IoT inalámbrico



Nota: Adaptado de LILYGO - LILYGO TTGO T-Beam V1.1, 2020,

(http://www.lilygo.cn/claproduct_view.aspx?TypeId=62&Id=1281&FId=t28:62:28) CC BY 3.0

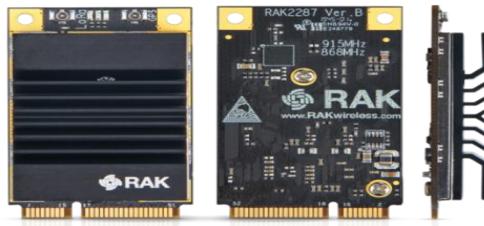
Esta pequeña tarjeta es una de las más completas, puesto que en un mismo modulo, cuenta con LoRa y un ESP32 que brindan funciones de WiFi, Bluetooth y GPS permitiendo ahorrar espacio y evitando la compra de hardware adicional.

Módulo puerta de enlace LPWAN RAK2287

A continuación se presenta una imagen del módulo RAK2287:

Figura 16

Concentrador LPWAN RAK2287



Nota: La Figura 16 corresponde a la forma física del módulo RAK2287. Adaptado de WisLink

RAK2287, 2021, (<https://www.rakwireless.com/en-us/products/lpwan-gateways-and-concentrators/rak2287>) CC BY 3.0

El RAK2287 hace referencia a un módulo concentrador LPWAN con un formato mini-PCIe el cual está basado en Semtech SX1302. Este dispositivo es de

fácil integración en enrutadores u equipos de red que cuenten con capacidades de puertas de enlace LPWAN. Es muy adecuado para aplicaciones de Internet de las cosas (IoT), que requieren una densidad de nodos de hasta 500 nodos por km² en un entorno con interferencia moderada.

Placa convertidora RAK2287 PI Hat

A continuación se presenta una imagen de la placa convertidora:

Figura 17

Módulo RAK2287 PI Hat



Nota: La Figura 17 corresponde a la forma física de la placa RAK2287 Pi Hat. Adaptado de WisHat RAK2287, 2021, (<https://store.rakwireless.com/products/rak2287-pi-hat>) CC BY

Este dispositivo es una placa convertidora la cual ayuda a conectar el módulo concentrador LPWAN RAK2287 ya mencionado con la Raspberry Pi.

Sensor de humedad de suelo capacitivo

El sensor de la Figura 18 permite medir niveles de humedad del suelo por medio de la detección capacitiva. Es fabricado con materiales resistentes a la corrosión, por lo que lo hace un dispositivo con gran vida útil. Debido a que incluye un regulador de voltaje, este sensor tiene un rango de voltaje operativo de 3.3 ~ 5.5 voltios.

Figura 18

Sensor de humedad de suelo



Fuente: Autores

Sensor de temperatura y humedad

A continuación se presenta una imagen del sensor de temperatura y humedad SHT10:

Figura 19

Sensor de temperatura y humedad SHT10



Nota: Adaptado de Didácticas Electrónicas - Sensor de humedad y temperatura industrial SHT10 para exteriores, 2020, (https://www.didacticaselectronicas.com/images/stories/virtuemart/product/SHT15-H_1.jpg) CC BY 3.0

El sensor de la Figura 19 contiene un módulo SHT10 y está capacitado para medir humedad y temperatura. Cuenta con una carcasa por lo que está diseñado para uso exterior o en proyectos y/o aplicaciones donde las mediciones de humedad y temperatura estén sujetas a condiciones adversas.

Actuadores

Bomba de agua sumergible HJ-411

A continuación se presenta una imagen de la bomba de agua sumergible:

Figura 20

Bomba de agua sumergible HJ-411



Fuente: Autores

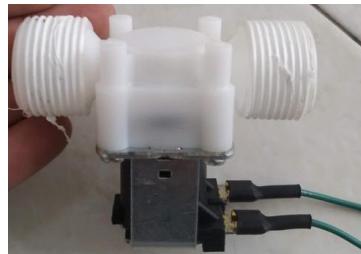
Esta bomba de agua es ideal porque puede emplearse en pequeños recipientes de agua. Cuenta con sistema de filtración.

Válvula solenoide 12v 3/4

A continuación se presenta una imagen de la válvula solenoide:

Figura 21

Válvula solenoide 12V 3/4



Fuente: Autores

Permite controlar el flujo de un fluido mediante un controlador alimentado a 12V. Este tipo de válvulas son un tipo de electroválvula todo/nada o abierto/cerrado; básicamente está formada por dos partes: el solenoide y el cuerpo de plástico.

Fuente switcheada

Es una fuente switcheada que cuenta con tres salidas 12V/10A Figura 22.

Figura 22

Fuente de alimentación utilizada



Fuente: Autores

Sensor de nivel líquido analógico (tipo flotador)

La Figura 23 corresponde a un sensor para medir nivel de líquido, permite realizar la medición del nivel de líquidos en tanques, para identificar si se alcanza o excede un nivel predeterminado.

Figura 23

Sensor de nivel de líquido



Nota: Adaptado de Didácticas Electrónicas - Sensor de nivel de líquido, 2020, (https://www.didacticaselectronicas.com/images/stories/virtuemart/product/Sensor_de_nivel_51b9eb6490023.jpg) CC BY 3.0

Módulo Regulador LM2596 Conversor DC-DC

La función del convertidor de voltaje DC-DC LM2596 (Figura 24) es entregar un voltaje de salida constante inferior al voltaje de entrada frente las variaciones del voltaje de entrada o de carga.

Figura 24

Módulo regulador LM2596



Nota: Adaptado de servoTronik – Módulo LM2596 Regulador de voltaje DC-DC, 2020, (<http://www.servotronik.com.co/index.php/producto/modulo-lm2596-regulador-de-voltaje-dc-dc-buck-1-25v-35v/>) CC BY 3.0

Este módulo puede soportar corrientes de salida de hasta 3A, voltaje de entrada entre 4.5V a 40V y voltaje de salida entre 1.23V a 37V, es último se selecciona mediante un potenciómetro multivuelta.

ChirpStack

El stack open-source llamado ChirpStack (Figura 25), es proveedor de servidor de LoRaWAN seguro de extremo a extremo con una disponibilidad del 99.9%.

Figura 25

Logotipo ChirpStack



Nota: Adaptado de ChirpStack open-source LoRaWAN® Network Server, 2021, (<https://www.chirpstack.io/>) CC BY 3.0

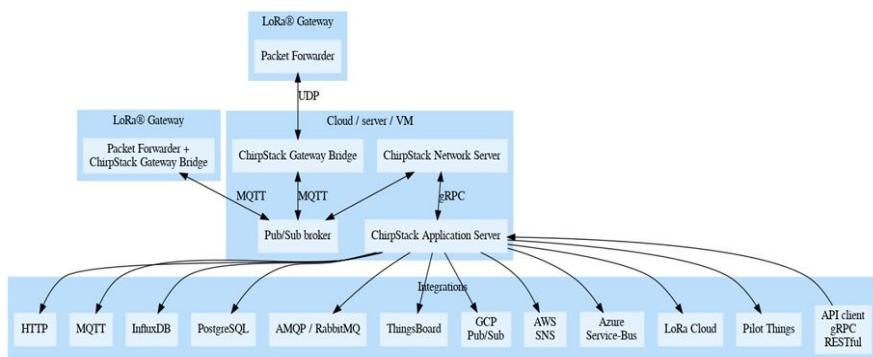
Este servidor brinda recursos necesarios para la solución LoRaWAN, así como también proporciona un conjunto de herramientas abiertas y una red global abierta para construir la aplicación de IoT a bajo costo, con la máxima seguridad mediante un cifrado de extremo a extremo.

ChirpStack hace referencia a un servidor de red LoRaWAN, el cual ofrece componentes de código abierto proporcionando una interfaz web sencilla que permite la administración de dispositivos e integración de APIs.” (ChirpStack, s.f.).

La Figura 26 refleja la arquitectura de los componentes conectados a ChirpStack LoRaWAN:

Figura 26

Arquitectura de componentes conectado en ChirpStack



Nota: Adaptado ChirpStack architecture, 2021, (<https://www.chirpstack.io/project/architecture/>) CC BY 3.0

Dispositivos LoRaWAN

Aquellos dispositivos LoRaWAN que no se describen en la Figura 26, hacen referencia a los dispositivos encargados de enviar datos al servidor de red ChirpStack por medio de una o diversas puertas de enlace LoRa. Para este caso, dichos dispositivos corresponden a sensores de temperatura y humedad.

Gateway LoRa

Por lo general el Gateway Lora escucha entre ocho o más canales de forma simultánea y realiza el envío de la información recibida (de sensores, etc.) al servidor de red LoRaWAN, es decir, el servidor de red ChirpStack para este caso práctico.

“El software que se ejecuta en el Gateway LoRa responsable de recibir y enviar se llama Packet Forwarder. Las implementaciones comunes son Semtech UDP Packet Forwarder y Semtech Basic Station Packet Forwarder” (ChirpStack, s.f.).

Gateway Bridge ChirpStack

Se encuentra ubicado entre el reenviador de paquetes y el protocolo MQTT. Es el encargado de transformar el formato Packet Forwarder en el formato de datos que implementa ChirpStack; cabe resaltar que también permite establecer integraciones con plataformas basadas en la nube como GCP Cloud IoT Core.

Servidor de red ChirpStack

Este servidor se encarga de administrar el funcionamiento de la red. Cuenta con un reconocimiento de las activaciones de dispositivos en la red.

Servidor de aplicaciones ChirpStack

Hace referencia al servidor de aplicaciones LoRaWAN, el cual es compatible con el servidor de red ChirpStack; ofrece una interfaz web para la integración de APIs para el control de aplicaciones, dispositivos, usuarios, organizaciones, etc. Y la información de enlace ascendente recibida será reenviada a una o varias integraciones configuradas.

Aplicación final

Recibe la información de los sensores o dispositivos por medio de una de las integraciones que han sido configuradas. “Puede utilizar la API del servidor de aplicaciones ChirpStack para programar una carga útil de enlace descendente a los dispositivos” (ChirpStack, s.f.). El objetivo de esta aplicación es el análisis de información, generación de alertas, acciones de control, entre otros.

A continuación, la Tabla 1 contiene el costo de elementos utilizados en el proyecto:

Tabla 1

Descripción de costos de equipos y materiales utilizados en el desarrollo del proyecto

Equipos							
Descripción	Cant.	Valor monetario (COP)	Valor monetario (US)	Recurso Usco	Recurso Facultad	Recurso compartido	Entidad Financiadora
Raspberry PI 4 4gb	1	\$ 300.000	\$ 78,35	No	No	No	Tesistas
Dragino LT3322-L	1	\$ 210.000	\$ 54,84	No	No	No	Tesistas
LILYGO TTGO t-Beam	1	\$ 140.000	\$ 36,56	No	No	No	Tesistas
RAK2287	1	\$ 360.000	\$ 94,02	No	No	No	Tesistas
RAK2287 PI Hat	1	\$ 140.400	\$ 36,56	No	No	No	Tesistas
Sensor de Humedad	1	\$ 30.000	\$ 7,83	No	No	No	Tesistas
Sensor de temperatura	1	\$ 50.000	\$ 13,06	No	No	No	Tesistas
Bomba de Agua	1	\$ 64.000	\$ 16,71	No	No	No	Tesistas
Electroválvula	1	\$ 20.000	\$ 5,22	No	No	No	Tesistas
LC-LM358-PWM2V	1	\$ 30.000	\$ 7,83	No	No	No	Tesistas
Huerta casera	2	\$ 200.000	\$ 52,23	No	No	No	Tesistas
Insumos extra	N/A	\$ 100.000	\$ 26,12	No	No	No	Tesistas
Total		\$ 1.644.400	\$ 429,33	-	-	-	-

Nota: Esta tabla representa los costos monetarios de cada uno de los materiales y equipos implementados en el desarrollo del proyecto. *Fuente:* Autores.

Tabla 2

Descripción de costos del software y/o aplicativos webs utilizado en el desarrollo del proyecto

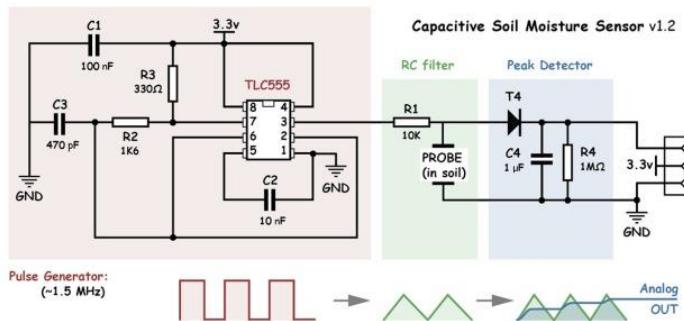
Software y/o aplicativos web						
Nombre	Cant.	Valor monetario (COP)	Valor monetario (US)	Recurso Usco	Recurso Facultad	Recurso compartido
Raspbian	1	\$ 0	\$ 0	No	No	No
Linux	1	\$0 (Versión gratuita)	\$0 (Versión gratuita)	No	No	No
ChirpStack	1	\$0 (Versión gratuita)	\$0 (Versión gratuita)	No	No	No
ThingsBoard	1	\$0 (Versión gratuita)	\$0 (Versión gratuita)	No	No	No
Total		\$ 0	\$ 0	-	-	-

Nota: Esta tabla representa el costo del software y/o aplicativo web que se implementó en el desarrollo del proyecto. *Fuente:* Autores

6.2. Caracterización del Sensor Capacitivo de Humedad

Figura 27

Esquema antes de la conversión



Nota: Adaptado de Capacitive Soil Moisture Sensor (v1.2) for Frequency Output, 2021, (<https://thecavepearlproject.org/2020/10/27/hacking-a-capacitive-soil-moisture-sensor-for-frequency-output/>) CC BY 3.0

La Figura 27 muestra el esquema antes de la conversión. La frecuencia de salida está controlada por la constante de tiempo al cargar y descargar C3 a R2 / R3. Estos sensores usan trazas coplanares para filtrar la salida de alta frecuencia de un oscilador NE555. Esta señal tiene una frecuencia de 370khz, la cual pasa por un filtro RC acoplado a un detector de picos. Cualquier cambio en la capacitancia

del suelo (húmedo o seco) tiene un efecto en la amplitud de la señal, que, gracias al detector de picos, es convertido en una señal DC, que puede ser leído por el ADC del microcontrolador. Para el caso concreto de los sensores adquiridos, el ADC de la TTGO no podía leer los datos correctamente, ya que, el voltaje de salida variaba entre 2.8V y 5V. Debido a esto, fue necesaria la construcción de un circuito acondicionador.

Los sensores de humedad del suelo miden el contenido volumétrico de agua de un suelo indirectamente mediante el uso de alguna otra propiedad, como la resistencia eléctrica o la permitividad dieléctrica.

6.3. Acondicionador de Señal

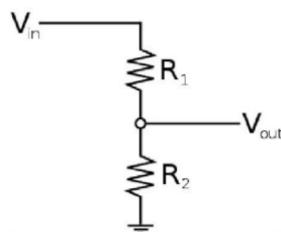
Cuando se requiere adquirir los datos de una señal analógica para luego introducirlos a un circuito digital es necesario contar con un ADC es decir un conversor analógico a digital. Casi siempre se conecta la señal analógica de forma directa a la entrada del conversor, pero en algunos casos es necesario modificar dicha señal antes de ser ingresada en el ADC y así sacar el máximo provecho del conversor.

Cabe resaltar que cada conversor cuenta con parámetros y características que permite diferenciarlo entre sí y lo hace destacar en ciertos casos. El hecho de que tenga un rango de entrada de 0 a 5V significa que la señal ingresada al ADC no puede ser inferior a 0V ni superior a 5V, de forma general esta situación siempre se cumple ya que, si se emplea el ADC para adquirir la lectura de un sensor, la tensión con que es alimentado dicho convertidor casi nunca llegará a superar la tensión con que es alimentado el microcontrolador del sensor.

Parámetros de diseño

Figura 28

Divisor de tensión



Fuente: Autores

$$VR2 = Vin \frac{R2}{R1 + R2} \quad 0V \leq Vin \leq 5V \quad 0V \leq VR2 \leq 3.3V$$

El divisor de tensión se diseña para mantener las proporciones, es decir, que a una entrada de 2.5V (5V/2) debería producirse una salida de 1.65V (3.3V/2).

$$R1, R2 \begin{cases} 3.3V = 5V \frac{R2}{R1 + R2} \\ 1.65 = 2.5V \frac{R2}{R1 + R2} \end{cases}$$

$$R1 \neq 0 \text{ y } R2 = \frac{33}{17} R1$$

$$R1 \neq 0 \text{ y } R2 = \frac{3.3}{1.7} R1$$

Se obtiene una solución infinita de valores no negativos que satisfagan la proporción de R1 y R2.

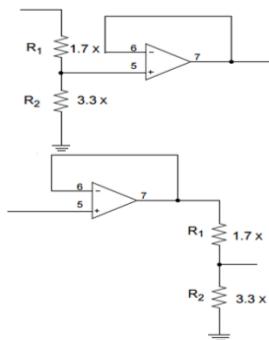
Inicialmente se utilizó un simple divisor de tensión con una relación $R_1 : R_2$ de 1.7: 3.3. Sin embargo, una alta impedancia en el divisor afectaba a los capacitores del ADC. Y si, por el contrario, se usaba un divisor de tensión de baja impedancia, se afectaban las mediciones, ya que, se alteraban los tiempos de carga y descarga de los capacitores presentes en el NE555 integrados en el sensor.

El atenuador puede estar alimentando una carga capacitiva, pero provoca la creación de un filtro paso bajo involuntario. Luego, es posible que el circuito atenuador necesite impulsar una carga de baja impedancia desde una fuente de alta impedancia, siendo así implementar necesariamente un amplificador operacional para almacenar las señales en búfer.

Por lo que, el circuito del amplificador operacional necesario es uno de ganancia unitaria, tal como el que se muestra en la Figura 29:

Figura 29

Circuito acondicionador de señal



Fuente: Autores.

Sin embargo, no fue necesario guardar la proporción 1.7:3.3 ya que la única condición que se debía cumplir era que $VR2 \leq 3.3V$ cuando $Vin = 5V$. Además, tampoco fue necesario estrictamente un amplificador operacional riel a riel, debido a que si se cumplía que $VR2$ no excediese el voltaje máximo que era capaz de proveer, podía usarse un amplificador como el LM358N. Por lo tanto, por simplicidad se escogieron R1 y R2 con valores de $22k\Omega$ y $33k\Omega$ respectivamente.

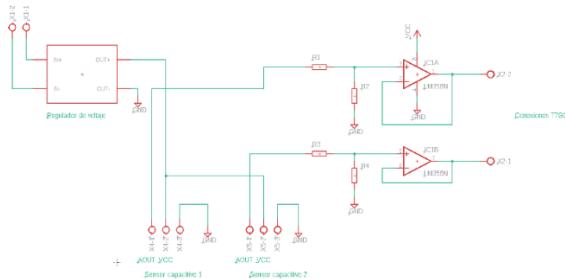
Entonces cuando se tiene una entrada máxima proveniente del sensor (5V), la salida de circuito acondicionador es:

$$VR2 = 5V \frac{33k\Omega}{22k\Omega + 33k\Omega} = 3V$$

A continuación, se presenta el circuito acondicionador de señal diseñado y posteriormente implementado:

Figura 30

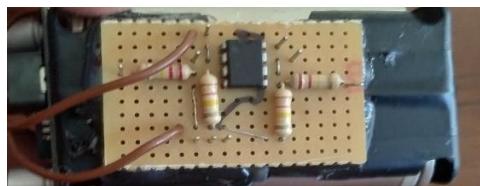
Esquema del circuito acondicionador de señal diseñado



Nota: Circuito amplificador de señal diseñado en Eagle PCB, en el regulador ingresa 7.8V y a su salida se obtiene 5V. *Fuente:* Autores.

Figura 31

Circuito acondicionador de señal construido



Nota: Circuito amplificador de señal elaborado sobre una baqueta universal. *Fuente:* Autores.

Figura 32

Circuito acondicionador de señal construido conectado al regulador de voltaje y sus baterías de alimentación



Nota: Circuito amplificador de señal conectado al regulador de voltaje y sus baterías de litio como fuente de alimentación. *Fuente:* Autores.

6.4. Configuración TTGO T-Beam

Para poder programar la TTGO desde el IDE de Arduino, se tuvo que clonar todos los repositorios del ESP32 (utilizando la línea de comandos del Anexo B), para que fueran compilados en ese entorno de desarrollo.

```
sudo usermod -a -G dialout $USER && \
sudo apt-get install git && \
wget https://bootstrap.pypa.io/get-pip.py && \
sudo python3 get-pip.py && \
sudo pip3 install pyserial && \
mkdir -p ~/Arduino/hardware/espressif && \
cd ~/Arduino/hardware/espressif && \
git clone https://github.com/espressif/arduino-esp32.git esp32 && \
cd esp32 && \
git submodule update --init --recursive && \
cd tools && \
python3 get.py
```

Para leer los datos del sensor SHT10 se utilizó la librería SHT1x-ESP mostrada en el Anexo B, en donde se especificó el voltaje de operación a 3.3V, y se mapearon los pines específicos correspondientes a la TTGO T-Beam v1.1.

En cuanto a los sensores capacitivos de humedad, se diseñó el circuito acondicionador presentado en la Figura 30. Dicho circuito envió una señal analógica proporcional a la humedad en el suelo. Por lo tanto, simplemente se utilizó el código del Anexo D para imprimir los valores de humedad de suelo. Una vez hecho esto, se utilizó la librería MCCI LMIC elaborada por IBM, la cual soporta las especificaciones LoRaWAN 1.0.2/1.0.3 clase A implementadas usando los chips

Semtech SX1272/SX1276. Primero se generó de manera pseudoaleatoria la Network Session Key, la App Session Key y la Device Address con la ayuda de la interfaz gráfica de Chirpstack.

Figura 33

Identificación TTGO t-Beam en ChirpStack

Device address *
01 8f c8 3d
While any device address can be entered, please note that a LoRaWAN compliant device address consists of an AddrPrefix (derived from the NetID) + NwkAddr.
Network session key (LoRaWAN 1.0) *
96 21 d0 a2 12 4b 1f cd d1 8f 80 f7 d0 28 e3 ee
Application session key (LoRaWAN 1.0) *
4f 6e ad ff 60 33 5e 48 c3 c3 08 bd 0f 96 73 15

Nota: La activación de la TTGO t-Beam en ChirpStack generó una clave de sesión de aplicación y una clave de sesión de red. *Fuente:* Autores.

Luego se ingresó esa información teniendo en cuenta el msb en los arreglos pertenecientes al código del Anexo B.

```
static const PROGMEM u1_t NWKSKEY[16] = { };  
static const u1_t PROGMEM APPSKEY[16] = { };  
static const u4_t DEVADDR = ;
```

Finalmente, se integró cada uno de los códigos del Anexo B a la función **void do_send(osjob_t* j)**, que se encargó de llenar el payload byte a byte con la información de temperatura, humedad del aire y humedad de suelo que se envió al servidor de aplicación.

Tabla 3

Representación del payload

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Temperatura		Humedad Relativa		Humedad de Suelo 1		Humedad de Suelo 2	
FLOAT		INT		INT		INT	

Nota: Esta tabla refleja el proceso de llenar el payload byte a byte con la información de los sensores. *Fuente:* Autores.

6.5. Geolocalización del proyecto

Para obtener la ubicación geográfica exacta se utilizó el módulo GPS Ublox NEO-6M-0-001 integrado en la TTGO, en la cual se corrió el código del Anexo A. Los datos obtenidos estaban en formato NMEA-0183 como se muestra en la Tabla 4.

Tabla 4

Registro de información suministrada por el módulo GPS de la TTGO t-Beam

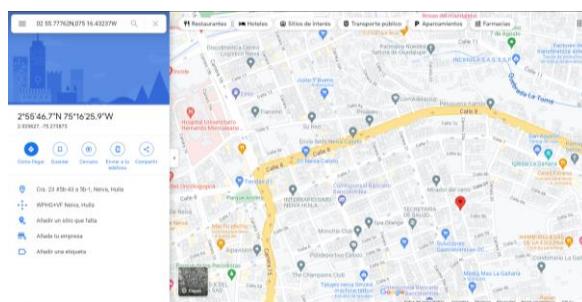
\$GPGGA	0255.77762,N,07516.43237,W,1,04,2.34,464.2M
Latitud (dd mm. mmmmm)	0255.77762N
Longitud (ddd mm.mmmmmm)	07516.43237W
Nº Satélites	04
Altitud (m)	494.2

Nota: Se detallan los valores de latitud, longitud y Nº Satélites y Altitud en formato NMEA-0183. *Fuente:* Autores.

Las coordenadas de la Tabla 4 generan como resultado la ubicación mostrada en la Figura 34.

Figura 34

Ubicación del módulo en Google Maps de acuerdo con las coordenadas suministradas por la TTGO t-Beam



Nota: Esta ubicación contó con 15 metros de error respecto a la dirección del domicilio donde se encontraba el dispositivo. *Fuente:* Autores.

Dicha ubicación se encuentra en la Calle 5B #23-11, La Gaitana. Este domicilio hace parte de la comuna 7 de la ciudad de Neiva. El terreno en esta zona es bastante irregular, con muchas edificaciones, vegetación espesa, y antenas de comunicación aledañas.

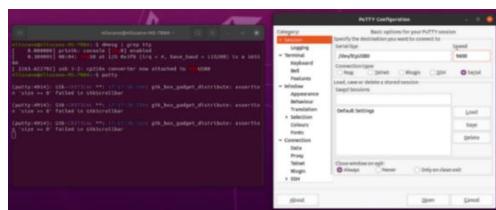
6.6. Configuración Dragino LT33222-L

Gracias a un adaptador serial a TTL se realizó la conexión del Dragino LT33222 al computador para su configuración inicial. Primero se actualizó el firmware a la última versión disponible, en este caso, la 1.5.4. Para iniciar el dispositivo en modo descarga se oprimió el botón PRO y seguidamente el botón RST. Con la ayuda del software Flash Loader Demostrator se cargó la imagen 1.5.4 entre las direcciones de memoria 0X8000000 y 0X800007F.

Una vez terminada la actualización se realizó nuevamente la conexión utilizando el software Putty (Figura 35), y se indicó que el tipo de conexión era serial a 9600 baudios.

Figura 35

Conexión Dragino LT33222-L con Putty



Nota: Confirmación de la conexión del Dragino LT33222-L mediante Putty. Fuente: Autores.

Seguidamente se digitó en la terminal la contraseña por defecto, que en este caso es 123456, y se pidió la impresión de todos los valores de configuración con el comando AT+CFG (Figura 36).

Figura 36

Valores de configuración Dragino LT33222-L

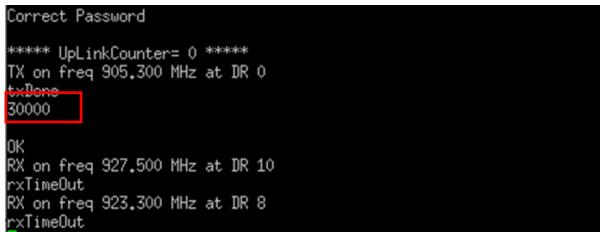


Fuente: Autores

Esta información fue necesaria en el servidor de aplicación para la conexión del nodo al Gateway. Luego se procedió a definir el intervalo de transmisión de datos con el comando AT+TDC=30000, que en este caso es de 30 segundos (Figura 37).

Figura 37

Intervalo de transmisión



```
Correct Password
***** UpLinkCounter= 0 *****
TX on freq 905,300 MHz at DR 0
txDone
30000
OK
RX on freq 927,500 MHz at DR 10
rxTimeOut
RX on freq 923,300 MHz at DR 8
rxTimeOut
```

A terminal window showing the output of a LoRa configuration command. The command 'txDone' is followed by '30000', which is highlighted with a red box. The response 'OK' indicates successful configuration.

Nota: Confirmación exitosa de la configuración del intervalo de transmisión. *Fuente:* Autores.

Finalmente se configuró el dispositivo como **clase C**, para que cualquier mensaje downlink proveniente del servidor fuese escuchado por el nodo tan pronto terminara de transmitir.

6.7. Creación del Servidor LoRa

Para la creación del servidor LoRa privado se instalaron los siguientes componentes necesarios para su funcionamiento:

6.7.1. Instalación del Gateway Bridge

Puesto que ChirpStack Gateway Bridge hace uso de MQTT para la publicación de eventos y recepción de comandos es necesario instalar Mosquitto, el cual es un corredor MQTT muy popular de código abierto. Para su instalación se hizo uso de la siguiente línea de código:

```
sudo apt-get install mosquitto
```

Una vez hecho este paso se procedió activar el repositorio ChirpStack Debian:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00
sudo echo "deb https://artifacts.loraserver.io/packages/stable main" | sudo tee /etc/apt/sources.list.d/loraserver.list
sudo apt-get update
```

Finalmente se instaló el Gateway Bridge y se comprobó su funcionamiento ejecutando el siguiente comando:

```
sudo apt-get install lora-gateway-bridge  
sudo systemctl start chirpstack-gateway-bridge  
sudo systemctl status chirpstack-gateway-bridge
```

Figura 38

Verificación del funcionamiento de la instalación del Gateway Bridge

```
nliscano@nliscano-MS-7884:~$ sudo systemctl status chirpstack-gateway-bridge  
● chirpstack-gateway-bridge.service - ChirpStack Gateway Bridge  
   Loaded: loaded (/lib/systemd/system/chirpstack-gateway-bridge.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2021-09-13 15:20:44 -05; 18h ago  
     Docs: https://www.chirpstack.io/  
       Main PID: 1283 (chirpstack-gate)  
          Tasks: 18 (limit: 9418)  
        Memory: 8.8M  
        CGroup: /system.slice/chirpstack-gateway-bridge.service  
                 └─1283 /usr/bin/chirpstack-gateway-bridge
```

Nota: Confirmación exitosa de la instalación y funcionamiento del Gateway Bridge. *Fuente:*

Autores.

6.7.2. Instalación del Servidor de Red

Para su instalación se ejecutó cada una de las siguientes líneas de código:

```
sudo apt-get install postgresql  
sudo apt-get install redis-server  
sudo -u postgres psql
```

Estando en el indicador de PostgreSQL, se digitó lo siguiente:

```
create role chirpstack_ns with login password 'dbpassword';  
create database chirpstack_ns with owner chirpstack_ns;  
\q
```

Es recomendable verificar si el usuario y la base de datos se configuró de forma correcta, para eso se conectó a estos por medio de:

```
psql -h localhost -U chirpstack_ns -W chirpstack_ns
```

Para la instalación del servidor red de ChirpStack se activó el repositorio Debian del servidor de red ChirpStack ejecutando el siguiente comando:

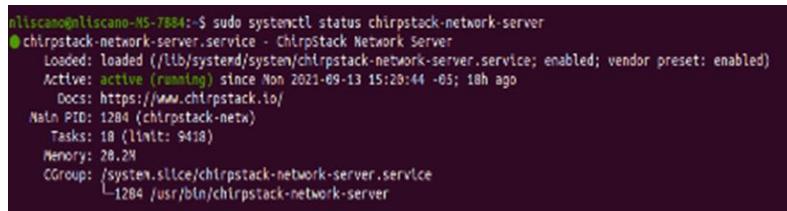
```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
1CE2AFD36DBCCA00  
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable main" | sudo  
tee /etc/apt/sources.list.d/chirpstack.list  
sudo apt update
```

Ahora sí, para la instalación del servidor de red Chirpstack se ejecutó este comando y se comprobó su funcionamiento:

```
sudo apt install chirpstack-network-server  
sudo systemctl start chirpstack-network-server  
sudo systemctl status chirpstack-network-server
```

Figura 39

Verificación del funcionamiento de la instalación del Network Server



A terminal window showing the output of the command `sudo systemctl status chirpstack-network-server`. The output indicates that the service is active and running since Mon 2021-09-13 15:20:44 -05; 10h ago. It shows the Docs URL as <https://www.chirpstack.io/>, Main PID as 1284, Tasks as 18, Memory usage as 28.2M, and the CGroup path as /system.slice/chirpstack-network-server.service - 1284 /usr/bin/chirpstack-network-server.

Nota: Confirmación exitosa de la instalación y funcionamiento del Servidor de Red. *Fuente:*

Autores.

6.7.3. Instalación del Servidor de Aplicación

Para su instalación se ejecutó cada una de las siguientes líneas de código:

```
sudo -u postgres psql  
create role chirpstack_as with login password 'dbpassword';  
create database chirpstack_as with owner chirpstack_ns;  
\c chirpstack_as  
create extension pg_trgm;  
create extension hstore;  
\q
```

Es recomendable verificar si el usuario y la base de datos se configuraron de forma correcta, para eso se conectó a estos por medio de:

```
psql -h localhost -U chirpstack_as -W chirpstack_as
```

Para la instalación del servidor aplicaciones de ChirpStack, se activó el repositorio Debian del servidor de red ChirpStack ejecutando el siguiente comando:

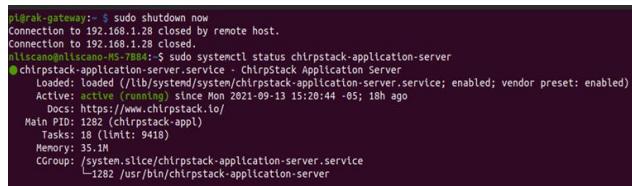
```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
1CE2AFD36DBCCA00  
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable main" | sudo  
tee /etc/apt/sources.list.d/chirpstack.list  
sudo apt update
```

Ahora sí, para la instalación del servidor de aplicaciones ChirpStack se ejecutó este comando y se comprobó su funcionamiento:

```
sudo apt-get install chirpstack-application-server  
sudo systemctl status chirpstack- application-server
```

Figura 40

Verificación del funcionamiento de la instalación del servidor de aplicaciones



```
lirak-pc:~ $ sudo chirtdown now  
Connection to 192.168.1.28 closed by remote host.  
Connection to 192.168.1.28 closed.  
lirak-pc:~ $ sudo systemctl status chirpstack-application-server  
● chirpstack-application-server.service - ChirpStack Application Server  
   Loaded: loaded (/lib/systemd/system/chirpstack-application-server.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2021-09-13 15:20:44 -05; 18h ago  
     Docs: https://www.chirpstack.io/  
       Main PID: 1282 (chirpstack-appl)  
         Tasks: 18 (limit: 9418)  
        Memory: 35.1M  
        CGroup: /system.slice/chirpstack-application-server.service  
                 └─1282 /usr/bin/chirpstack-application-server
```

Nota: Confirmación exitosa de la instalación y funcionamiento del Servidor de Aplicación. *Fuente:*

Autores.

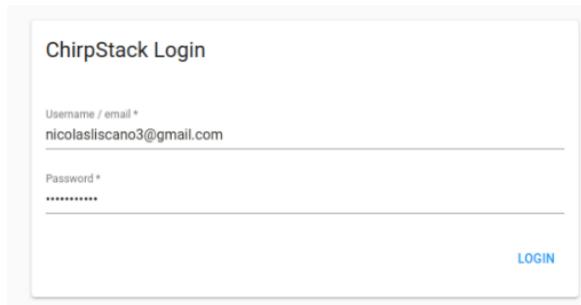
6.8. Integración con ChirpStack

Después de comprobar el funcionamiento del servidor de aplicación mediante la terminal de comando, se procedió a iniciar sesión con el usuario las credenciales predeterminadas, es decir usuario: admin, y contraseña admin, las cuales fueron cambiadas por seguridad.

Como el servidor de aplicación se creó de forma local, para iniciar sesión se escribió en el navegador **http://localhost: 8080** / y luego apreció la siguiente página:

Figura 41

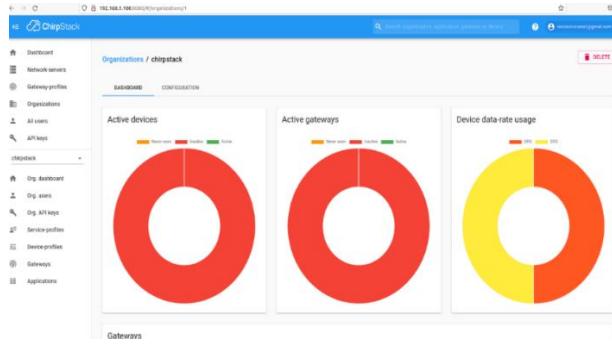
Ventana de inicio de sesión en ChirpStack



Fuente: Autores

Figura 42

Página de inicio del servidor de aplicación ChirpStack



Fuente: Autores

Al ingresar se modificaron los parámetros de usuario y contraseña como ya se había mencionado y luego visualizó el menú de opciones disponible en la parte izquierda. Primero se creó el perfil de servicios, servidor de red y el perfil de puertas de enlace.

En el perfil de servicio se puede configurar los límites mínimos y máximos del ADR, así como permitir que el Gateway envíe los metadatos al servidor, junto con su ubicación (Figura 43).

Figura 43

Configuración del perfil en ChirpStack

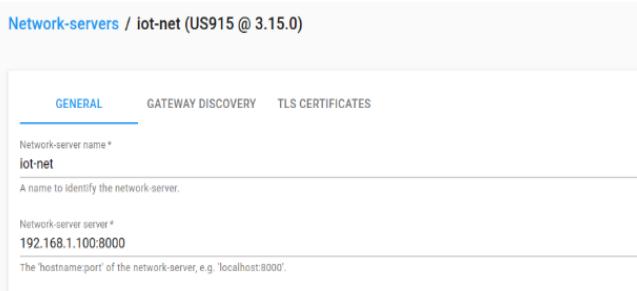
The figure is a screenshot of the ChirpStack configuration interface for a service profile. The title bar says 'Service-profiles / iot-service'. The form fields include: 'Service-profile name*' with 'iot-service' entered; 'Add gateway meta-data' checked; 'Enable network geolocation' checked; 'Device-status request frequency' set to '0'; 'Minimum allowed data-rate*' set to '0'; and 'Maximum allowed data-rate*' set to '5'. There is also a note: 'When enabled, the network-server will try to resolve the location of the devices under this service-profile. Please note that you need to have gateways to support.' and 'Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.'

Nota: Al perfil de servicio se nombró iot-service, se habilitó el envío de metadatos desde el Gateway al servidor y la geolocalización. *Fuente:* Autores.

Luego se configuró el servidor de red apuntando la IP del servidor local, en este caso al puerto 8000.

Figura 44

Configuración del servidor de red en ChirpStack

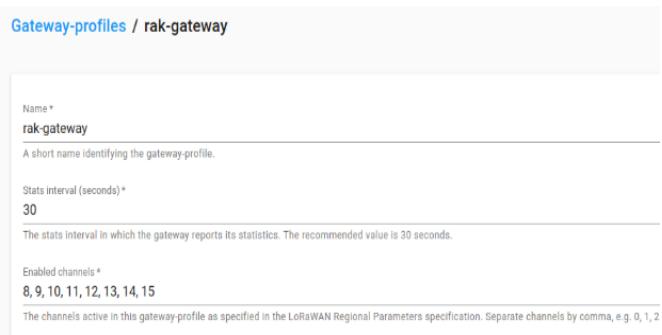


Nota: Al perfil de servicio de red se nombró iot-net y se ingresó la IP local para su configuración habilitando el puerto 8000. *Fuente:* Autores.

En la Figura 45 se definió cada cuento el Gateway debía reportar las estadísticas, y también, los 8 canales que este tiene habilitado para la recepción de datos.

Figura 45

Configuración del Gateway en ChirpStack



Nota: Se estableció intervalos de 30 segundos para el reporte de estadísticas y se habilitaron los canales 8,9,10,11,12,13,14 y 15 para la recepción de información. *Fuente:* Autores

Posteriormente se creó cada uno de los perfiles (Figura 46) para los dispositivos Dragino LT33222 y TTGO T-Beam (nodos finales).

Figura 46

Ventana Device-profiles en ChirpStack

Device-profiles		+ CREATE
Name	Network Server	
dragino-io-controller	iot-net	
ttgo-microcontroller	iot-net	
Rows per page:	10	1-2 of 2 < >

Fuente: Autores

La Figura 47 representa cómo se generó de manera aleatoria el *Device Address*, *Network Session Key* y *Application Session Key*. Estos datos son necesarios para que la TTGO y el Gateway se conecten mediante activación ABP.

Figura 47

Perfil del dispositivo Dragino LT33222-L

Device address *	01 8f c8 3d
While any device address can be entered, please note that a LoRaWAN compliant device address consists of an AddrPrefix (derived from the NetID) + NwkAddr.	
Network session key (LoRaWAN 1.0) *	96 21 d0 a2 12 4b 1f cd d1 8f 80 f7 d0 28 e3 ee
Application session key (LoRaWAN 1.0) *	4f 6e ad ff 60 33 5e 48 c3 c3 08 bd 0f 96 73 15

Fuente: Autores

Por lo tanto, los arreglos que definen estos parámetros quedan de la siguiente manera.

```
static const PROGMEM u1_t NWKSKEY[16] = { 0x96, 0x21, 0xD0, 0xA2, 0x12, 0x4B, 0x1F,  
0xCD, 0xD1, 0x8F, 0x80, 0xF7, 0xD0, 0x28, 0xE3, 0xEE };  
  
static const u1_t PROGMEM APPSKEY[16] = { 0x4F, 0x6E, 0xAD, 0xFF, 0x60, 0x33, 0x5E,  
0x48, 0xC3, 0xC3, 0x08, 0xBD, 0x0F, 0x96, 0x73, 0x15 };  
  
static const u4_t DEVADDR = 0x018fc83d ;
```

6.9. Construcción del Gateway

Realizado este procedimiento se acopló con tornillos la tarjeta mPCIe RAK2287 en la ranura mPCIe del RAK2003 PI HAT asegurándose de que encajaran

perfectamente, luego se conectó las antenas GPS y LoRa en sus puertos correspondientes.

Hecho esto se tomó el RAK2287 PI HAT (ya ensamblado) y se incorporó a la Raspberry Pi (Figura 48) mediante el conector de 40 pines que tienen en común. La tarjeta SD con el firmware ya instalado se insertó en la ranura correspondiente de la Raspberry Pi.

Figura 48

Acople a la Raspberry Pi y vista interna del Gateway



Nota: La imagen de la izquierda corresponde al acople entre el RAK2287 Pi Hat y la raspberry Pi. La imagen de la derecha corresponde al interior del Gateway desarrollado, el cual cuenta además con una antena y un ventilador de 12 mm para la disipación del calor en la placa. *Fuente:* Autores.

Figura 49

Elaboración final del Gateway



Nota: Corresponde al ensamble completo del Gateway y su vista externa. *Fuente:* Autores.

Luego, se conectó a través del puerto Micro USB (Pi3) / puerto USB tipo C (Pi4) el Gateway al tomacorriente y se encendió. Para la su configuración se conectó al pc mediante el puerto Ethernet y se ejecutó la siguiente línea de código:

```
ssh pi@192.168.10.10
```

Al hacerlo solicitó contraseña por lo que se escribió raspberry (contraseña predeterminada) y se inició sesión en la puerta de enlace a través de SSH con éxito como se observa en la Figura 50.

Figura 50

Visualización de la línea de comando al ingresar a la configuración del Gateway



Nota: Este es el mensaje que se visualizó al ingresar a la configuración del Gateway exitosamente.

Fuente: Autores.

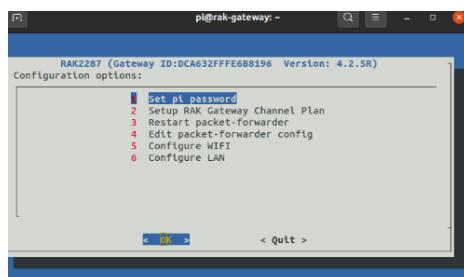
Luego para realizar las configuraciones del Gateway se ingresó la siguiente línea de comando:

```
sudo gateway-config
```

Al hacerlo, inmediatamente se visualizó lo siguiente:

Figura 51

Opciones de configuración del Gateway



Nota: La Figura 53 corresponde a las seis opciones disponibles para la configuración del Gateway.

Fuente: Autores.

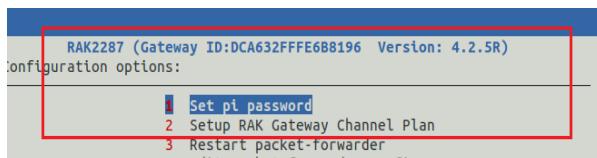
A continuación se explican estas opciones:

- Set pi password: Permite para establecer/cambiar la contraseña del Gateway.
- Set up RAK Gateway LoRa Concentrator: Permite realizar la configuración de la frecuencia en la que va a operar el Gateway y el servidor LoRaWAN.
- Restart packet -forwarder: Se usa para ejercer reinicio en el reenvío de paquetes LoRa.
- Edit packet-forwarder config: Permite abrir el archivo global_conf.json y así editar parámetros de LoRaWAN de forma manual.
- Configure Wifi: Como su nombre lo indica, sirve para configurar los ajustes de Wi-Fi y así conectarse a una red.
- Configure LAN: Permite configurar los ajustes del adaptador Ethernet

La identificación única que se le asignó al Gateway se observó en la parte superior de las opciones de configuración (Figura 52):

Figura 52

ID único del Gateway



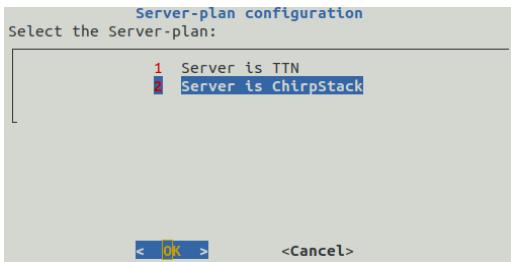
Nota: En la parte superior se indica el ID único del Gateway. *Fuente:* Autores.

Dicha identificación hace referencia al Gateway EUI y fue necesario anotar ese ID porque fue el que permitió hacer el registro del Gateway en el servidor de red LoRa ChirpStack. Como es recomendable cambiar la contraseña entonces se realizó este paso, en el menú de opciones de configuración del Gateway (Figura 51) se seleccionó la número 1, se hizo clic en “yes” y se ingresó la nueva contraseña, al hacerlo se pulsó “Enter” y listo.

Luego se hizo la configuración del concentrador RAK Gateway LoRa mediante la opción 2 donde se escogió uno de los dos servidores LoRa (TTN o ChirpStack). Para este caso práctico **se eligió ChirpStack (Figura 53) puesto que TTN se actualizó a la versión 3 (TTN V3)** y debido a esta actualización, su fase beta presentó ciertos bugs.

Figura 53

Opciones de servidor LoRa

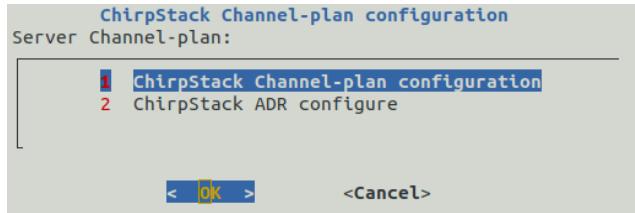


Nota: El servidor TTN hace referencia a The Things Network, pero para este caso se seleccionó ChirpStack. *Fuente:* Autores.

Al elegir ChirpStack se desplegó un nuevo menú donde se visualizó dos opciones y se escogió plan de configuración ChirpStack y luego US_902_928 que hacía referencia al plan disponible en América (Figura 54).

Figura 54

Elección del plan de configuración del Gateway



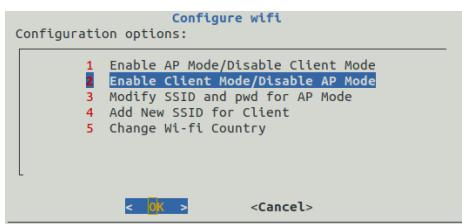
Fuente: Autores.

Como el propósito fue usar ChirpStack de forma independiente, que fuera capaz de implementarse a una red de área local o remota, fue necesario conectar el Gateway a un dispositivo de red el cual permitió conectarse a Internet en este caso mediante Wifi.

Para esto en el menú principal (Figura 51) se seleccionó la opción 5 “Configure WIFI”; como por defecto el RAK2287 WisLink LPWAN funciona en modo Wi-Fi AP se activó la opción 2 “Enable Client Mode/Disable AP mode” (Figura 55) y de esta forma el Gateway se conectó al enrutador.

Figura 55

Opciones de configuración de WiFi



Nota: Estas son las opciones de configuración de WiFi, la opción 2 deshabilita la configuración de fábrica, habilitando el modo cliente. *Fuente:* Autores.

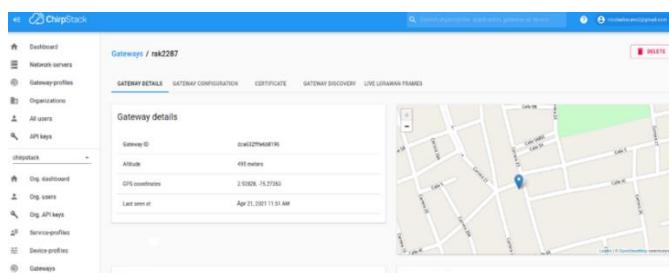
Hecho esto se seleccionó la opción 4 “Add New SSID for Client” del menú, se seleccionó Colombia como país de residencia y se ingresó al SSID y contraseña de la red a la que se conectó.

Después se realizó la conexión al servidor de aplicativo ChirpStack creado anteriormente. Esta plataforma hace posible que los dispositivos de baja potencia hagan uso de las puertas de enlace de largo alcance, así se conecte a una red descentralizada de código abierto y permitir el intercambio de datos con la aplicación. Para llevar a cabo esta tarea se ingresó en el navegador de igual forma que en el apartado 6.9, y se inició sesión.

Para visualizar información adicional del Gateway en ChirpStack, se ingresó a la opción “Gateways” ubicada en el menú y se observó parámetros como ID, localización y última vez (Figura 56).

Figura 56

Información del Gateway suministrada en ChirpStack



Nota: En esta imagen se puede observar información del Gateway adicional como, su ID, geolocalización mediante coordenadas y un mapa, así como también su última conexión. *Fuente:* Autores.

6.10. Adaptación de Equipos y Sensores para Exteriores

Debido a que los sensores, el módulo receptor de datos y el dispositivo de control ON/OFF junto cada uno de sus componentes iban a expuestos a un ambiente húmedo, se asiló cada uno de los cables conductores mediante cintas termo encogibles y/o aislante e impermeabilizar con recipientes de plástico el área donde se localizó cada dispositivo. A continuación, se presentan fotografías de la adaptación realizada:

Figura 57

Aislamiento e impermeabilización de los sensores de humedad capacitivos



Nota: Esta imagen indica el aislamiento de los sensores capacitivos mediante cinta termo encogible y cinta aislante. *Fuente:* Autores.

Figura 58

Conexión entre el módulo TTGO t-Beam y el sensor SHT10, aislamiento e impermeabilización



Fuente: Autores.

Figura 59

Conexión sensores capacitivos y sensor SHT10 con la TTGO y el dispositivo acondicionador de señal



Fuente: Autores.

Figura 60

Impermeabilización de las conexiones electrónicas de la electroválvula



Fuente: Autores.

Figura 61

Impermeabilización de las conexiones electrónicas: TTGO y circuito acondicionador de señal



Nota: La imagen indica la conexión eléctrica (al interior de una botella de plástico) entre la tarjeta Wifi TTGO t-Beam junto con el dispositivo acondicionador de señal. *Fuente:* Autores

6.11. Construcción de las Huertas

Se implementó dos huertas con dimensiones de 0.8m x 2.3m x 0.7m (alto, largo, ancho respectivamente), donde se llevó a cabo el trasplante de plántulas y posteriormente el crecimiento de plantas. Es de resaltar que cada huerta contó con unos soportes de tubo PVC que sirvieron de soporte para las plantas.

La temperatura ambiente y la humedad del sustrato fue monitoreada tanto en las dos huertas, sin embargo la huerta 1 correspondió al cultivo donde se llevó a cabo el control de los actuadores mediante dispositivos LoRa para el riego por goteo, mientras que la huerta 2 hizo referencia al cultivo regado manualmente con pulverizador.

Figura 62

Huertas rectangulares para el cultivo de tomate (construidas a partir de madera)



Fuente: Autores.

Sistema de riego por goteo para la huerta 1.

Este sistema se diseñó de tal manera que las unidades de riego que conformaron esta operación fueran ubicadas en sectores separados en distancias no superiores a los 40 cm, con el fin de equilibrar presiones y dividir los caudales de agua para obtener un menor diámetro en las tuberías, agrupando así todas las plantas cultivadas junto con sus necesidades (hídricas y de fertilización).

Se implementó el sistema por goteo superficial y su ensamble se hizo a partir de:

- 5 metros de manguera de 4/7mm de diámetro
- 1 acople llave 3/4" a manguera de 1/2"
- 1 acople de manguera 1/2" a manguera 4/7mm
- 10 aspersores (goteros), rociadores regulables de 8 chorros
- 10 sujetadores para manguera de 4/7mm
- 10 Tes. Acoples (T) para manguera de 4/7mm

Los goteros implementados son de tipo vortex por lo que “el agua después de atravesar un orificio ingresa a una pequeña cámara en forma tangencial originando un movimiento en espiral que ocasiona una pérdida de carga y luego sale en forma de gota” (Liotta , Carrión , Ciancaglini, & Olguín, 2015).

La manguera de 4/7 mm que se encargó del transporte de agua hacia los goteros se conectó a un acople de manguera de 1/2" a manguera de 4/7 mm y de llave 3/4" a manguera 1/2", dicho acople (Figura 63) se conectó una manguera transparente de 3/8" (de 4 metros de longitud) la cual estaba conectada a la bomba de agua depositada al interior del tanque de 50L.

Figura 63

Acople realizado para el riego de la huerta 1



Nota: La Figura 67 indica el acople entre las dos mangueras requeridas, a la izquierda está la manguera que transporta el agua desde el tanque y a la derecha la que lleva el agua a las plantas. *Fuente:* Autores.

De esta forma, al ejercer acción de control sobre la bomba de agua, se consiguió que el agua suficiente fluyera para alimentar las plantas de tomate de la huerta 1, durante todo el desarrollo del proyecto, mediante riego por goteo. Adicionalmente se utilizó fertilizantes disueltos en agua con el objetivo de regar y abonar al mismo tiempo.

Implementación de actuadores

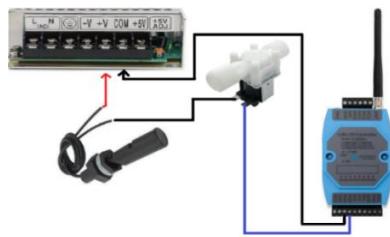
Para el control y suministro de agua en la huerta 1 mediante el sistema de riego por goteo, se implementó una válvula solenoide, que se encargó del paso de agua para el llenado de un tanque de 50L y una bomba de agua, que sirvió para provocar circulación de agua desde el tanque hasta las plantas a través del sistema de riego por goteo. Como se mencionó anteriormente, la necesidad de utilizar una bomba de agua se basó en el hecho de que el tanque no estaba situado a una altura suficiente para que la gravedad provocara dicha circulación.

Uno de los extremos de la válvula, se acopló a una manguera (de 1/2" y 10 metros de longitud) conectada a la llave de paso que proporcionó el agua, en el otro

extremo también se acopló un pedazo de 1 metro de longitud de dicha manguera para depositar el agua en al interior del tanque y llenarlo las veces necesarias para el suministro hídrico en la huerta 1. Para su activación, se realizó la conexión de la Figura 64.

Figura 64

Esquema de conexión válvula solenoide, sensor de nivel flotador y fuente de alimentación



Nota: La activación de la bomba de agua se hizo mediante el uso de los relés RO2.

Fuente: Autores.

Con esto, cuando el nivel de agua del tanque se reducía el flotador activó el estado alto del sensor y se activaba el solenoide de la válvula, el cual se desplazó junto con el diafragma de la válvula, lo que permitió el paso de agua para llenar el tanque y cuando el nivel de agua subía, el flotador activaba en alto al sensor lo que apagaba la electroválvula interrumpiendo el paso de agua. A continuación, se presenta la conexión completa para el suministro de agua para el llenado del tanque.

Figura 65

Conexión de la válvula solenoide



Nota: En esta figura, la manguera que sale del tanque es la principal, permite el llenado del tanque las veces que sea necesaria. *Fuente:* Autores.

La bomba de agua se acopló a una manguera transparente (1/2" diámetro y 4 metros de longitud) y se depositó al interior de tanque de 50L. Para su funcionamiento se realizó la conexión de la Figura 66. De esta forma, al enviar un

mensaje de como downlink al Dragino LT33222-L por medio de ChirpStack, se activó su relé que energizó la bomba, lo que permitió la circulación de agua del tanque hacia la huerta 1.

Figura 66

Esquema de conexión bomba de agua y Dragino LT33222-L



Nota: La activación de la bomba de agua se hizo mediante el uso de los relés RO1.

Fuente: Autores.

Cultivo de tomate

Al inicio las semillas de tomate fueron plantadas en los germinadores (pequeñas materas) Figura 67.

Figura 67

Cultivo de tomate sembrado en el germinadero



Fuente: Autores.

Al completar 30 días después la germinación, las plántulas estaban en condiciones óptimas para ser depositadas en su respectiva huerta, así que se trasplantaron en los dos huertos construidos (Figura 68 y Figura 69).

Figura 68

Vista completa del cultivo sembrado para el desarrollo del proyecto



Nota: Los objetos de color amarillo corresponden a pequeños platos de plástico que tienen la función de actuar como trampas amarillas para la mosquita blanca de los invernaderos *Fuente:* Autores.

Figura 69

Plántulas trasplantadas a su respectiva huerta



Nota: En la parte izquierda se observa la huerta 1 y a la derecha se observa la huerta 2, cada una con sus plántulas transplantadas y trampas amarillas. *Fuente:* Autores.

Figura 70

Ubicación de sensores en las huertas



Nota: El recuadro de color rojo indica la posición del sensor capacitivo en cada huerta, mientras que el recuadro azul indica la posición del sensor SHT10. *Fuente:* Autores.

6.12. Control de los Actuadores Mediante Mensajes Downlink

Las direcciones de memoria del Draguino LT33222 con el prefijo **0x03** modifican el estado de los relés RO1 y RO2. Dependiendo del estado de los bytes los relés tienen los siguientes estados: 01: Cerrado 00: Abierto 11: Sin acción

Tabla 5

Direcciones de memoria Dragino LT33222-L

Código Downlink	RO1	RO2
03 00 11	Abierto	Sin acción
03 01 11	Cerrado	Sin acción
03 11 00	Sin acción	Abierto
03 11 01	Sin acción	Cerrado
03 00 00	Abierto	Abierto
03 01 01	Cerrado	Cerrado
03 01 00	Cerrado	Abierto
03 00 01	Abierto	Cerrado

Nota: Esta tabla indica el código Downlink con la respectiva respuesta de estado en el relé RO1 y RO2. Adaptado de LT Series LoRa IO Controller User Manual, Versión 1.4.2.

6.12.1. Control de Tiempo

Para el control de tiempo de los actuadores se utiliza el prefijo **0x05**, más 4 bytes adicionales. Cada byte hace lo siguiente:

Tabla 6

Función Byte

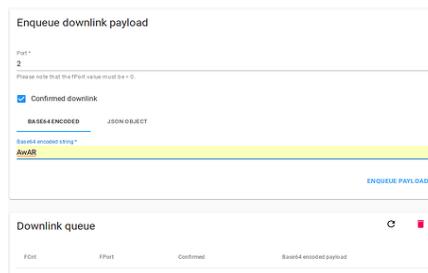
Primer Byte	0x05
Segundo Byte	00: Los relés vuelven a su estado original 01: Los relés cambian a un estado inverso
Tercer Byte	0x11: RO1 y RO2 pasan a estar abiertos 0x10: RO2 pasa a estar abierto y RO1 cerrado 0x01: RO2 pasa a estar cerrado y RO1 abierto 0x00: RO1 y RO2 pasan a estar cerrados 0x20: RO1 sin acción y RO2 pasa a estar cerrado 0x21: RO1 sin acción y RO2 pasa a estar abierto 0x02: RO2 sin acción y RO1 pasa a estar cerrado 0x12: RO2 sin acción y RO1 pasa a estar abierto
Cuarto Byte y Quinto Byte (ms)	Tiempo que dura el latch

Nota: Esta tabla indica la función que tiene cada byte adicional (junto con el prefijo 0x05) para establecer el control de tiempo de los relés. Adaptado de LT Series LoRa IO Controller User Manual, Versión 1.4.2.

ChirpStack codifica los mensajes downlink en base 64, por lo tanto, para enviar el mensaje en un formato apropiado se debe convertir de hexadecimal a base 64 primero. El puerto que utilizo el Dragino LT33222 fue el número 2. Además, ChirpStack ofrece la posibilidad de que el mensaje downlink sea confirmado al llegar al nodo.

Figura 71

Configuración de parámetros para enlace Downlink



Nota: “AwAB” es la notación en base 64 del hexadecimal 03 00 01 la cual activa el relé RO1 y mientras que el relé RO2 mantiene apagado. *Fuente:* Autores

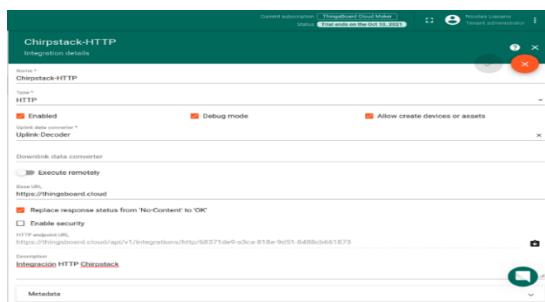
Si se quisiera encender el llenado del tanque y el sistema de riego a la misma vez, el mensaje sería 03 01 01 en hexadecimal; y en base 64 “AwEB”.

6.12.2. Integración con ThingsBoard

Para la configuración de la integración con ThingsBoard se realizó la siguiente secuencia de pasos, que permitieron recibir los datos provenientes del servidor de ChirpStack:

Figura 72

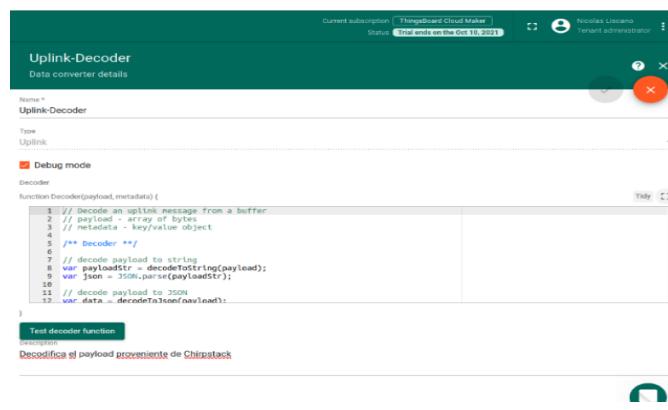
Creación del tipo de integración



Nota: La Figura 76 refleja al interfaz de usuario de la API ThingsBoard para su integración con ChirpStack mediante el protocolo HTTP. *Fuente:* Autores.

Figura 73

Creación del decodificador de mensajes Uplink



Fuente: Autores.

A continuación, se presenta el código referente al decodificador de mensajes Uplink, la función de este código es crear, definir variables y asignar atributos a cada dispositivo

```

/** Decodificador */
var payloadStr = decodeToString(payload);
var json = JSON.parse(payloadStr);

var datos = decodeToJson(payload);
var humAstr = data.objectJSON[11]+data.objectJSON[12];
var humS1str = data.objectJSON[30]+data.objectJSON[31];
var humS2str = data.objectJSON[48]+data.objectJSON[49];
var tmp = data.objectJSON[64]+data.objectJSON[65]+data.objectJSON[66]...
...+data.objectJSON[67];
var humA = parseFloat(humAstr);
var humS1 = parseFloat(humS1str);
var humS2 = parseFloat(humS2str);
var temp = parseFloat(tmp);

var deviceName = 'ttgo-tbeam';
var deviceType = 'HyT';
var customerName = 'Publico';
var groupName = 'Monitoreo-HyT';

var result = {
    deviceName: deviceName,
    deviceType: deviceType,
    customerName: customerName,
    groupName: groupName,
    attributes: {
        devEUI: datos.devEUI,
        RxInfo: datos.rxInfo,
        TxInfo: datos.txInfo,
        FPort: datos.fPort,
    },
    telemetry: {
        HumedadAire: humA,
        HumedadSuelo1:humS1,
        HumedadSuelo2:humS2,
        Temperatura: temp,
        RawData: datos.data,
    }
};

function decodeToString(payload) {
    return String.fromCharCode.apply(String, payload);
}

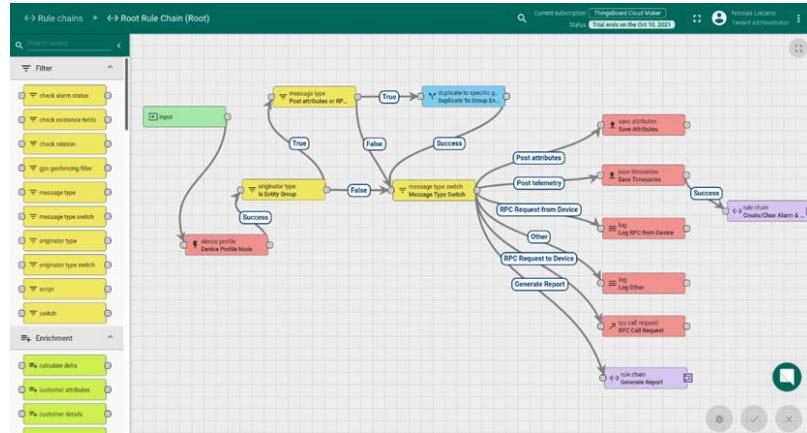
function decodeToJson(payload) {
    var str = decodeToString(payload);
    var datos = JSON.parse(str);
    return datos;
}

return result;

```

Figura 74

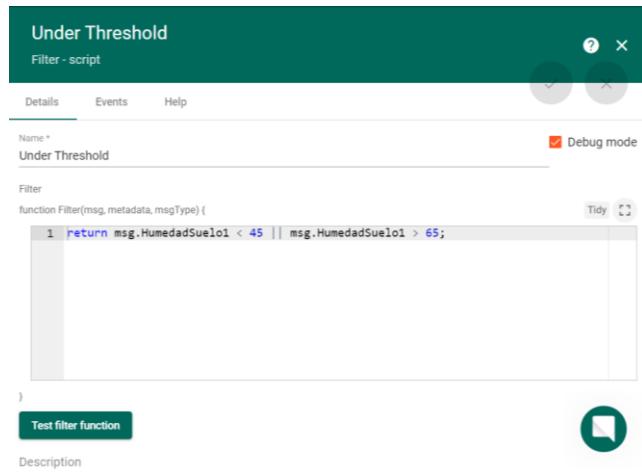
Creación y definición de la cadena de reglas (rule chain)



Nota: Estas etiquetas hacen referencia un grupo lógico de nodos de reglas y relaciones que permiten llevar a cabo el procesamiento de eventos. *Fuente:* Autores.

Figura 75

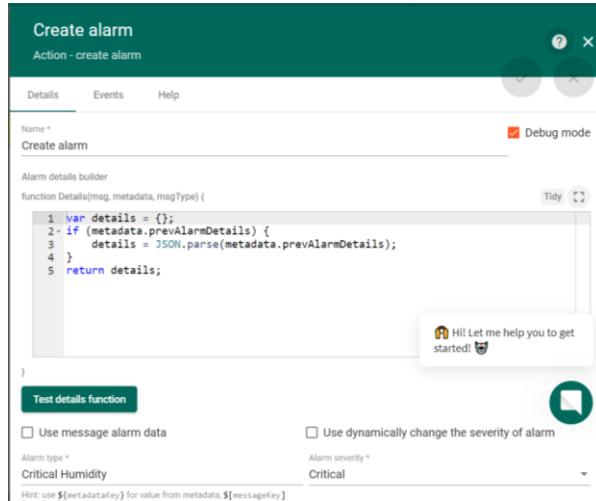
Definición de los parámetros de control ON/OFF mediante el envío de mensajes downlink



Nota: La línea de código descrita en la Figura 75 permite verificar si la humedad del sustrato está un rango esperado, por lo que, si dicha humedad está en este intervalo, el script devolverá False en la interfaz de usuario, de lo contrario, se devolverá True. *Fuente:* Autores.

Figura 76

Creación del mensaje de alarma



Nota: La línea de comando descrita en esta figura, devuelve un True en la interfaz de usuario en el instante en que la humedad del sustrato no esté en el intervalo esperado. *Fuente:* Autores.

7. RESULTADOS

7.1. Cultivo de Tomate

Después de haber realizado el trasplante de plántulas a las huertas 1 y 2, se observó desarrollo floral sobre algunas ramificaciones de las plantas.

Figura 77

Crecimiento y floración de las plantas



Fuente: Autores.

Fruto

Sobre la semana nueve se observó unos pequeños frutos que crecían en ciertas ramificaciones donde se observó buen desarrollo floral como se observa en la Figura 78.

Figura 78

Primeros frutos en las plantas



Fuente: Autores.

El hongo (Tizón Temprano) (Figura 79) apareció aproximadamente a la primera semana después del trasplante de plántulas en la huerta 1 como 2. Debido

al depósito de agua en las hojas de las plantas del huerto 2, a causa del riego manual con pulverizador, la mosquita blanca de los invernaderos tuvo origen sobre la segunda y tercera semana (Figura 80) y sobre la quinta semana se observaron larvas (Polilla del Tomate) (Figura 81).

Figura 79

Efecto del tizón temprano en algunas plantas del huerto 1



Fuente: Autores.

Figura 80

Presencia de la mosquita blanca de los invernaderos en plantas del huerto 2



Fuente: Autores.

Figura 81

Presencia de larvas (Polilla en tomate) en planta del huerto 2



Fuente: Autores.

Para las enfermedades y plagas que se presentaron se llevó a cabo un tratamiento a base de químicos como Amistar 50 WG para el caso del tizón temprano, para la mosquita blanca de los invernaderos se aplicó Citroemulsion 800 EC y para las larvas se aplicó un insecticida llamado Iorsban; cabe resaltar que estos químicos fueron disueltos en agua para su aplicación.

7.2. Visualización de datos en ChirpStack

La interfaz del servidor de aplicaciones ChirpStack es amigable con el usuario y le proporciona visualización de registro de tramas en vivo por el Gateway y los nodos finales. Allí se puede observar todos los metadatos de transmisión y recepción, al igual que el LoRaWAN PHYPayload sin decodificar.

La Figura 86 representa el diagrama de bloques elaborado para indicar el proceso de adquisición y visualización de información.

Figura 82

Esquema de conexión con ChirpStack



Nota: En esta figura se observa que la implementación de ChirpStack permite establecer comunicación bidireccional entre los nodos finales y el Gateway, por lo que además de visualización de información, el usuario establece la toma de decisión de acuerdo con su criterio. *Fuente:* Autores.

A continuación la Figura 83 muestra la cadena de bytes adquirida por la TTGO T-Beam. Esta corresponde a los datos de Humedad del aire, Temperatura y Humedad de suelo 1 y 2. Gracias a la función Decode mostrada en el Anexo D, los datos son mostrados en un objeto JSON con un formato legible, que es enviando mediante el protocolo HTTP a ThingsBoard.

Figura 83

Registro de tramas, enlace LoRaWAN (ChirpStack)

```
▼ txInfo: () 3 keys
  frequency: 904700000
  modulation: "LORA"
  ▼ LoRaModulationInfo: () 4 keys
    bandwidth: 125
    spreadingFactor: 7
    codeRate: "4/5"
    polarizationInversion: false
  adr: true
  dr: 3
  fCnt: 104
  fPort: 1
  data: "KXRClAC4BA=="
  ▼ objectJSON: () 4 keys
    Humedad: 68
    HumedadSuelo1: 70
    HumedadSuelo2: 66
    Temperatura: 28.0625825066
  tags: () 0 keys
  confirmedUplink: false
  devAddr: "018fc83d"
  publishedAt: "2021-05-11T05:10:08.238685321Z"
  ▼ txInfo: () 3 keys
  frequency: 904700000
  modulation: "LORA"
  ▼ LoRaModulationInfo: () 4 keys
    bandwidth: 125
    spreadingFactor: 7
    codeRate: "4/5"
    polarizationInversion: false
  adr: true
  dr: 3
  fCnt: 104
  fPort: 1
  data: "KXRClAA3AA=="
  ▼ objectJSON: () 4 keys
    Humedad: 68
    HumedadSuelo1: 70
    HumedadSuelo2: 65
    Temperatura: 28.6217006521
  tags: () 0 keys
  confirmedUplink: false
  devAddr: "018fc83d"
  publishedAt: "2021-05-11T05:10:15.384930215Z"
```

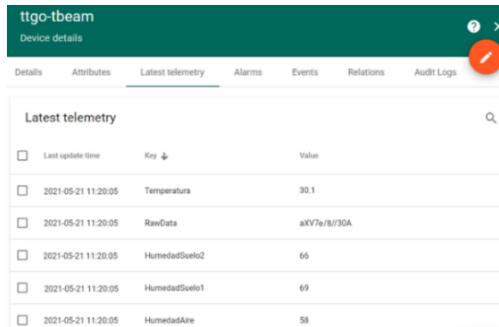
Nota: Además de presentar la información adquirida por los sensores, la plataforma ChirpStack muestra información detallada sobre parámetros de comunicación como ancho de banda, factor de dispersión y el tipo de frecuencia. *Fuente:* Autores.

7.3. Integración con ThingsBoard

La API ThingsBoard permitió observar información en detalle tal como se puede observar en la Figura 84, sin embargo, también es posible configurar varios tipos de indicadores interactivos (widgets) que presentan la información de forma interactiva como se observa en la Figura 85.

Figura 84

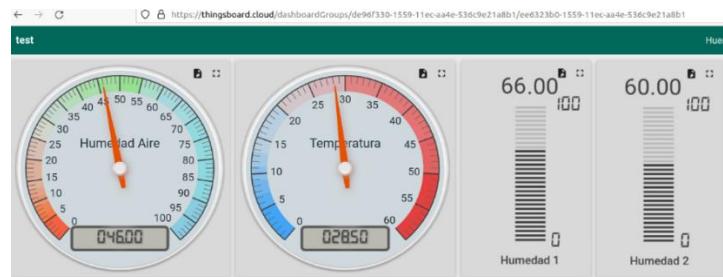
Información detallada el estado activo del nodo TTGO en ThingsBoard



Nota: Esta figura presenta el reporte detallado de las variables monitoreadas por cada uno de los sensores implementados. *Fuente:* Autores.

Figura 85

Implementación de widgets para la visualización de parámetros en ThingsBoard



Fuente: Autores.

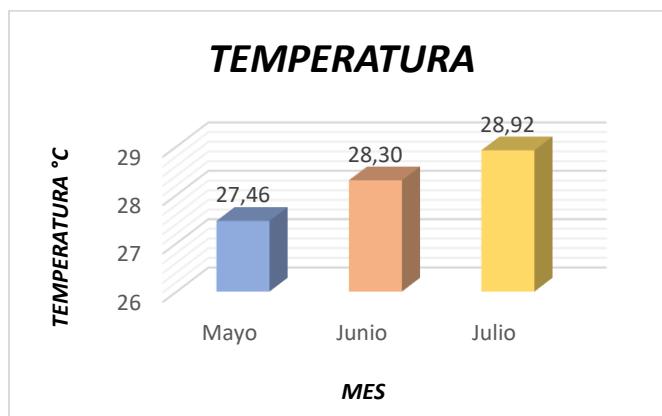
8. ANÁLISIS DE RESULTADOS

8.1. Promedio de Humedad y Temperatura

Durante el desarrollo del proyecto se llevó un registro de la temperatura y humedad del ambiente diario; lo que permitió establecer un promedio mensual de dichos parámetros en las huertas, gracias a esto de evaluó su incidencia sobre el uso del agua respecto a los métodos de riego.

Gráfica 1

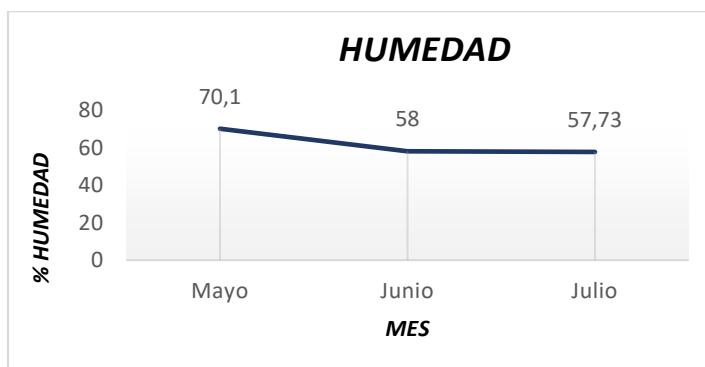
Promedio temperatura mensual registrada durante los meses de trabajo.



Nota: Esta gráfica describe el promedio mensual de la temperatura ambiente en las huertas, registrada con el sensor STH10. *Fuente:* Autores.

Gráfica 2

Promedio humedad mensual registrada durante los meses de trabajo



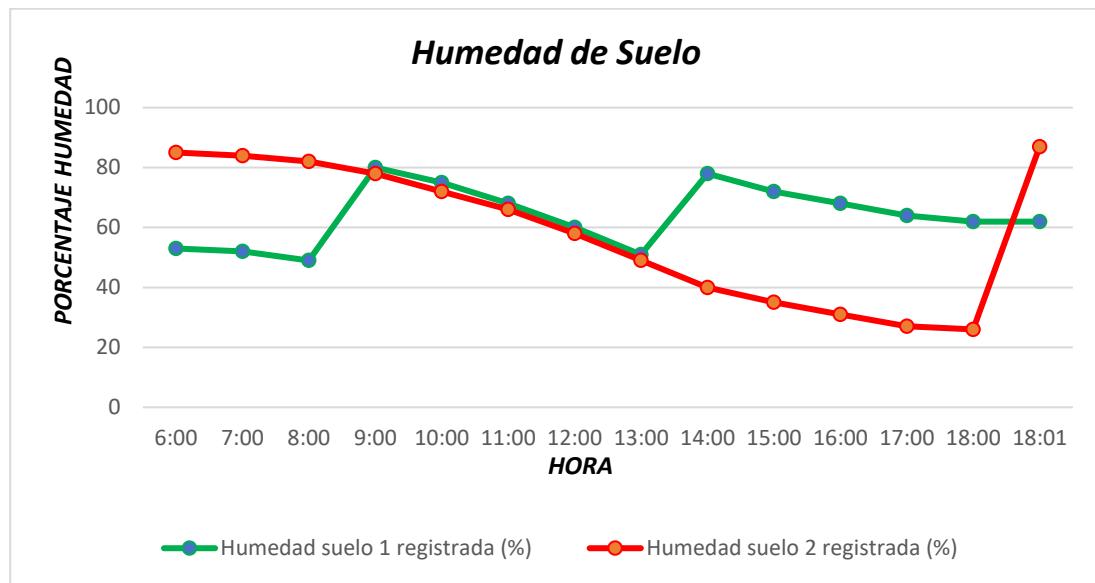
Nota: Esta grafica describe el promedio mensual de la humedad ambiente en las huertas, registradas con el sensor STH10. *Fuente:* Autores.

El registro de la temperatura ambiente y la humedad del sustrato en cada huerta, fue fundamental a la hora de tomar acciones de riego en dichas huertas, en especial la huerta 1, que contó con sistema de riego por goteo manipulado de forma remota mediante el servidor de aplicación ChirpStack y los actuadores, este control ON/OFF fue vital para aprovechar de forma eficiente el recurso hídrico, conforme a intervalos de riego que se llevó a cabo diariamente y también ayudó a reducir el estrés hídrico en las plantas.

A continuación se presenta la humedad del sustrato registrada por los sensores capacitivos en las dos posibles configuraciones del Dragino LT33222-L durante un día con alta radiación solar.

Gráfica 3

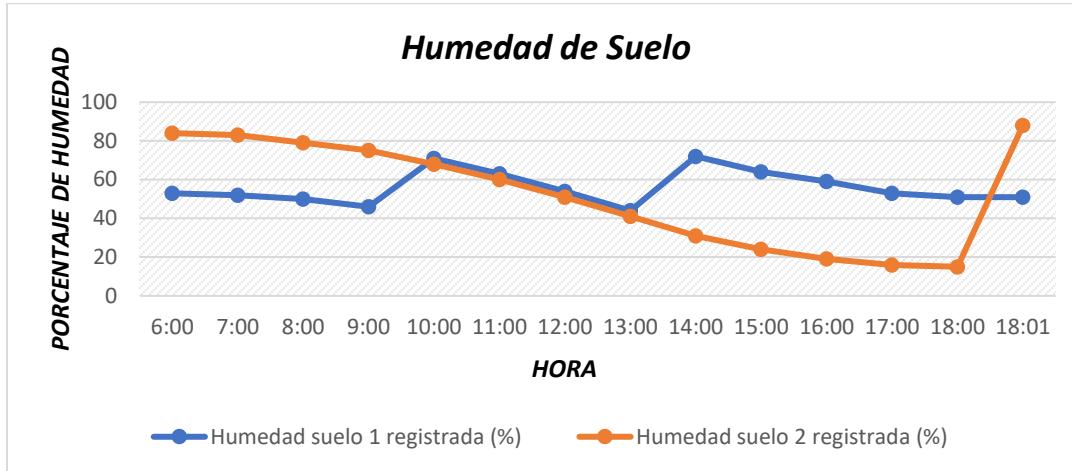
Humedad de Suelo: configuración Dragino en Clase A



Nota: Esta figura muestra el promedio de humedad de suelo obtenido cada hora en un día cuya temperatura promedio fue 30°C. Fuente: Autores

Gráfica 4

Humedad de Suelo: configuración Dragino en Clase C



Nota: Esta figura muestra el promedio de humedad de suelo obtenido cada hora en un día cuya temperatura promedio fue 30°C. Fuente: Autores

Como se observa en las Gráficas 3 y 4 la humedad del suelo bajó a menos del 20% en algunos puntos del día, lo que pudo favorecer el debilitamiento de las plantas, y por ende, la aparición de enfermedades en la huerta 2. Por el contrario, la huerta 1 mantuvo un intervalo de humedad mucho más estable, que solo se vio afectado en épocas de lluvia. Si se hace un promedio de humedad entre las 2 huertas se puede observar que la diferencia no es muy grande, sin embargo, el estrés que causa a las plantas tener valores de humedad de suelo bajo en un clima como el de la ciudad de Neiva es muy influyente en la salud y desarrollo de estas.

8.1.1. Suministro de Agua

A continuación, se presenta la cantidad de agua que se suministró al día con los sistemas de riego establecidos; la obtención de este parámetro en el riego por goteo (Huerta 1), se basó en la diferencia de nivel de agua visualizada de las marcas del tanque de 50L que se obtuvo durante el mes, mientras que en el riego manual con pulverizador (Huerta 2) se estableció un volumen de 3.5L agua para cada planta de acuerdo con la publicación titulada “Agricultura bajo techo reduce consumo de agua a la mitad “presentada por CATIE. Cabe resaltar que utilizando el segundo método de riego se tuvo en cuenta si el suelo se encontraba húmedo antes de suministrar más agua.

Tabla 7*Cantidad de agua suministrada a la Huerta 1*

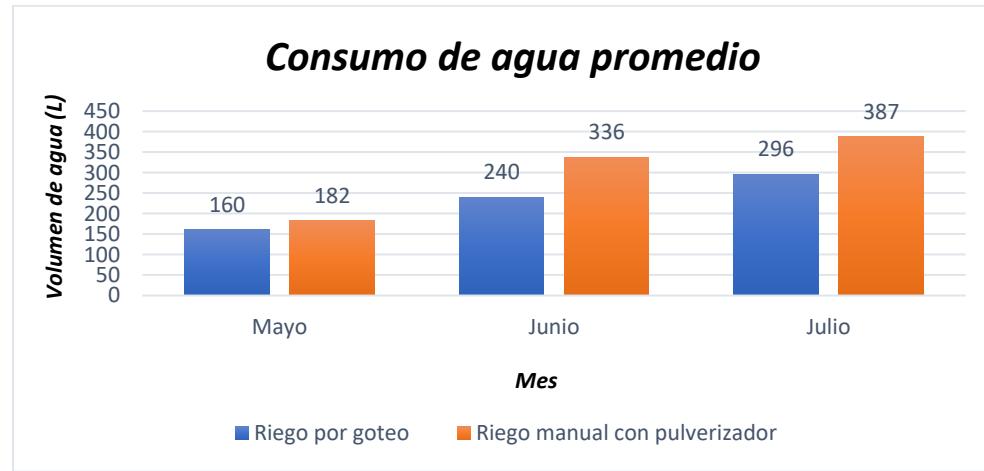
Huerta 1 (riego por goteo)	
Cantidad plantas	Litros de Agua/día por planta
4	2,5
Total	10

Fuente: Autores**Tabla 8***Cantidad de agua suministrada a la Huerta 2*

Huerta 2 (riego manual)	
Cantidad plantas	Litros de Agua/día por planta
4	3,5
Total	14

Fuente: Autores.

De igual forma se llevó un registro del promedio de agua usado mensualmente el cual se presenta a continuación:

Gráfica 5*Consumo de agua promedio*

Nota: Esta gráfica describe el consumo promedio de agua mensual (durante el desarrollo del proyecto)

que implicó el uso del sistema de riego y el riego manual con pulverizador. *Fuente:* Autores.

Al observar la Gráfica 5, es posible determinar que se presentó un aumento en los niveles de agua suministrados en los dos métodos de riego durante los meses de junio y julio, esto se debe a la variación del clima que se presentó durante el transcurso de estos dos meses; pues al contar con temperaturas superiores a 30 °C, aumentó la exigencia hídrica por parte de la planta y consigo las frecuencias de riego.

La información suministrada por el sensor de humedad implementado en la huerta 1, junto con el enlace LoRaWAN que se estableció entre los dispositivos y el registro de información en tiempo real en el servidor de aplicación en ChirpStack, favoreció al uso del sistema de riego por goteo, el cual proporcionó un ahorro de agua poco más del 20% (en los meses de junio y julio) a comparación del riego manual con pulverizador.

Esto se debe a que el suelo se humedeció gradualmente sin tener que alterar su estructura, por lo que el intercambio gaseoso no se vio interrumpido. La huerta 2 demandó mayor cantidad de agua debido a que al humedecer el sustrato hubo tendencia a producirse mayor perdida por evapotranspiración debido a las condiciones climáticas.

Por su parte, durante el mes de mayo se presentó poca diferencia entre los sistemas de riego debido a que hubo mayor presencia de lluvias durante este mes, por lo que se aprovechó la lluvia para el riego gradual de las dos huertas.

La implementación de sistemas de riego localizados como el riego por goteo son fáciles de instalar y de gran utilidad a la hora de reducir impacto ambiental, pues este sistema de riego presentó un ahorro de poco más del 20% representado en una reducción de 91L de agua frente al riego manual mediante pulverizador.

Y su característica de riego localizado permitió que las plantas de la huerta 1 contaran con niveles de humedad óptimos para su desarrollo, evitando el depósito de agua en hojas y las perdidas por evaporación.

8.2. Nodo Clase A y Clase C

El Dragino LT33222 transmitió mensajes Uplink cada 30 segundos, y la humedad de suelo fue leída cada 90 segundos por la TTGO. Por lo tanto, en el peor de los casos, el sistema de riego solo podía detenerse 123 segundos después de que la humedad de suelo hubiese superado el rango deseado. Si el nodo se encontraba en clase C, a este solo le tomaba 91segundos accionar el sistema de riego, debido a que, tan pronto era leída la humedad, este se encontraba disponible para recibir mensajes Downlink.

Tabla 9*Tiempos de activación del sistema de riego*

	Uplink TTGO	Uplink Dragino LT33222	RX1	RX2	Duración del paquete en el Aire	Total (s)
Clase A (1)	90	1 - 30	1	2	1	94 - 123
Clase A (2)	90	1 - 30	1	----	1	93 - 122
Clase A (3)	90	-----	1	2	1	93
Clase A (4)	90	-----	1	----	1	92
Clase C	90	-----	---	----	1	91

Nota: Esta tabla contiene todos los posibles escenarios para la activación del sistema de riego. *Fuente:* Autores.

La Clase A(2) de la Tabla 9 describe el caso en el que apenas el nodo de la TTGO lee un valor de humedad de suelo critico el nodo del Dragino LT33222 tiene que esperar entre 0 y 30 segundos para tener su ventana de oportunidad de recepción en RX1. En la Clase A(1) ocurrió lo mismo mencionado anteriormente con la diferencia que la ventana de oportunidad de recepción fue en RX2. En la Clase A(3) y A(4) ocurre que la ventana de oportunidad coincide con el momento en el que el nodo de la TTGO ha leído un valor critico de humedad de suelo. Sin embargo, en el primero el mensaje Downlink tiene lugar en RX1 y en el segundo en RX2. Finalmente, en la Clase C el Dragino LT33222 puede accionar los actuadores tan pronto la humedad de suelo critica es leída.

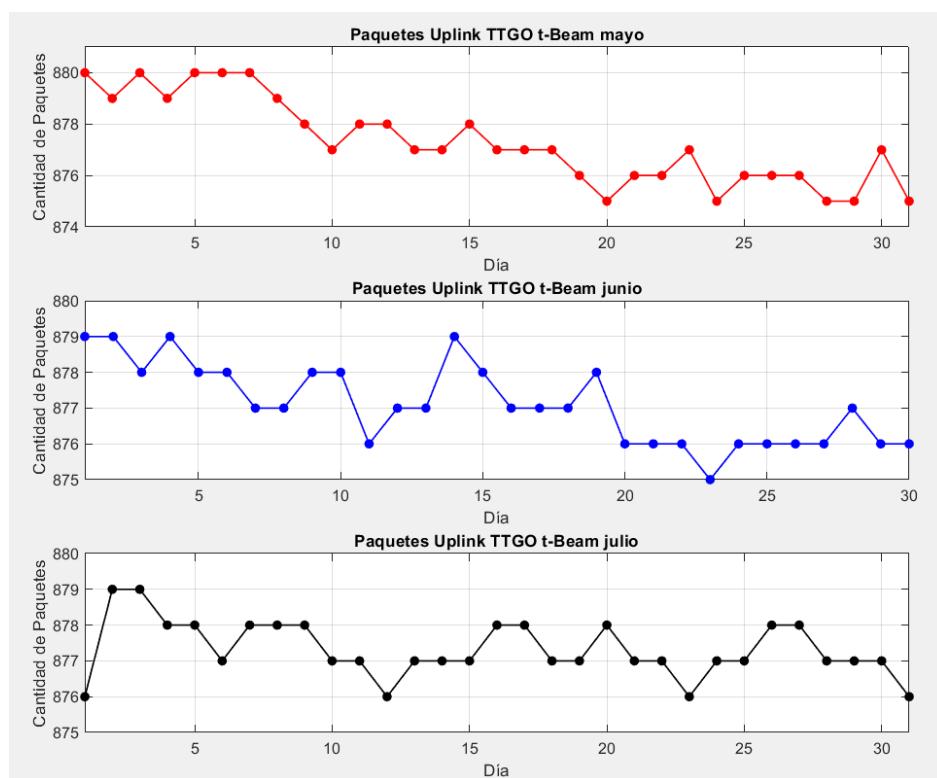
8.3. Comportamiento de los dispositivos LoRa

Mediante el servidor de aplicación creado en ChirpStack se obtuvo información sobre el funcionamiento de los dispositivos LoRa implementados como nodos finales (TTGO-t beam y Dragino). La plataforma ChirpStack ofreció una visualización de datos en tiempo real mediante gráficas, pero también permite exportar dichos datos en formato Excel (.xlsx) para efectos de estudio del usuario como, por ejemplo: registros, comparaciones entre otros. A continuación, se presenta la cantidad de paquetes recibidos en el enlace LoRaWAN TTGO – Gateway:

8.3.1. Paquetes recibidos TTGO t-Beam

Gráfica 6

Paquetes recibidos por la TTGO t-Beam



Nota: Los puntos corresponden al promedio diario de paquete Uplink recibidos por la TTGO durante los meses de trabajo. *Fuente:* Autores.

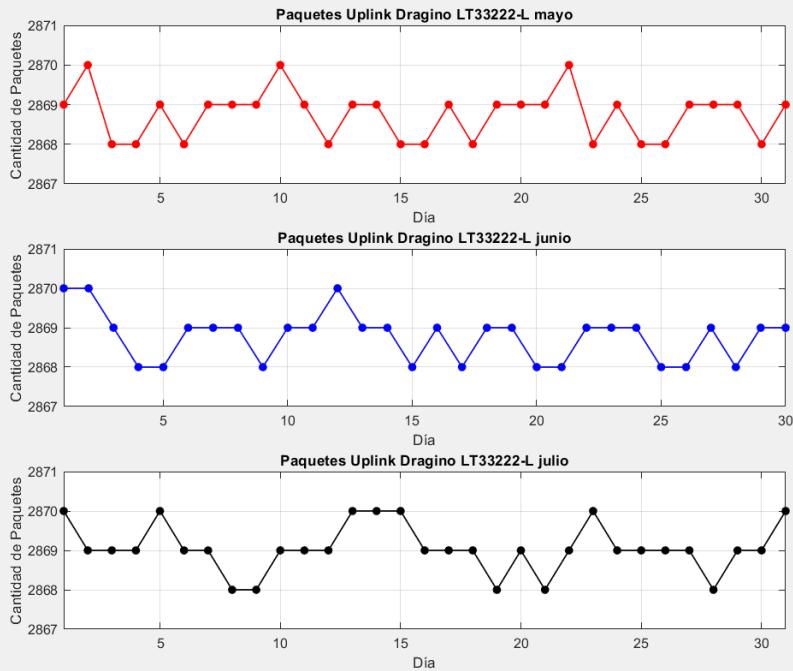
La variación en la cantidad de paquetes observados en la Gráfica 6 durante los meses de trabajo se debe a que la TTGO t-Beam estuvo conectada al Gateway aproximadamente 21 horas al día, por lo que hubo un lapso de 3 horas (en la madrugada) que estuvo desconectada del Gateway (apagada) debido a la descarga de la batería de litio, por lo que el enlace LoRaWAN entre el Gateway y la TTGO se detuvo durante este periodo de tiempo.

8.3.2. Paquetes recibidos Dragino LT33222-L

A continuación, se presenta la cantidad de paquetes recibidos por el nodo LoRa encargado del control de actuadores para el uso del sistema de riego por goteo:

Gráfica 7

Paquetes recibidos Dragino



Nota: Los puntos corresponden al promedio diario de paquetes Uplink recibidos por el Dragino durante los meses de trabajo. *Fuente:* Autores.

Para el caso del Dragino LT33222-L (Gráfica 7), la pequeña perdida de paquetes puede deberse a que el radio enlace entre el Gateway y el Dragino no contó con línea de vista pues a diferencia de la TTGO, este dispositivo estuvo conectado todo el día de forma permanente durante dichos meses.

8.3.3. Indicador de Intensidad de Señal Recibida (RSSI)

De acuerdo con (SpeedCheck, s.f.) el indicador de intensidad de señal recibida conocido como RSSI indica la potencia de la señal recibida en milivatios y cuya medición es el dBm, generalmente se utiliza como una medida de qué tan bien puede “escuchar” el receptor la señal remitente permitiendo que este detecte y reciba señales de cualquier punto de acceso o de un router específico.

Como ya se mencionó, el RSSI se mide en dBm y corresponde a un valor negativo, tiene un rango de medición que va desde 0 (cero) hasta -120 por lo que, cuanto más cerca esté de 0 mejor será la señal (SonicWall, 2019).

A continuación, se presenta una tabla con valores típicos de LoRa RSSI:

Tabla 10

Valor típico de RSSI en enlaces LoRaWAN

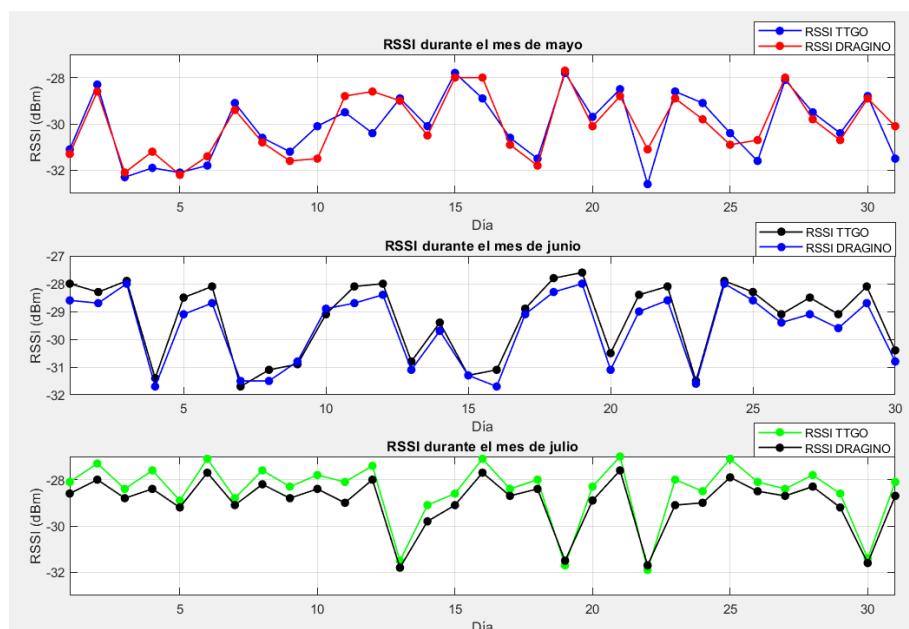
Intensidad de la señal (dBm)	Indicador
≥ -30	Señal Fuerte
≥ -60	Señal Buena
≥ -70	Señal Aceptable
≥ -115	Señal Regular
≤ -120	Señal Débil

Nota: Esta tabla representa la calidad de la señal en función al RSSI medido en dBm. Adaptado de Conceptos de actualidad: LoRa y LoRaWan (<https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/>).

El indicador de RSSI del enlace LoRaWAN y se obtuvo gracias a la información proporcionada por ChirpStack, se determinó y graficó el promedio mensual de RSSI obtenido durante el desarrollo del proyecto:

Gráfica 8

Promedio diario de RSSI durante los meses de trabajo



Nota: Los puntos corresponden al promedio diario del RSSI para cada uno de los nodos implementados durante cada mes de trabajo. *Fuente:* Autores.

Conforme a lo mencionado al inicio del apartado 8.3.3 y al observar la Gráfica 8, se puede afirmar que el enlace LoRaWAN establecido entre la TTGO, el Dragino y el Gateway presentó buena calidad en la señal puesto que el RSSI se mantuvo en un rango mayor o igual a 30 dBm donde se cataloga la señal como fuerte, lo que generó estabilidad en la transmisión de mensajes entre este nodo y el Gateway durante los meses de trabajo.

Al observar la Gráfica 6 y Gráfica 7 del apartado 8.3.1 y 8.3.2 referentes a los paquetes recibidos entre los nodos y el Gateway, se apreció que el mes de julio, mes donde hubo mejor respuesta en paquetes Uplink, registró el mejor valor de RSSI en comparación a los meses de mayo y junio, esto quiere decir que el RSSI se relaciona de forma directa con la cantidad de paquetes recibidos.

La pérdida de paquetes y disminución leve en el valor RSSI, pudo ser afectada por los obstáculos físicos como paredes, así como también por las transmisiones simultáneas en frecuencias de radio adyacentes debido a la instalación torres de telecomunicación u otras estructuras de tecnologías que manejan radiofrecuencias localizadas alrededor de la ubicación del proyecto (Figura 86).

Figura 86

Antenas de telecomunicación situadas alrededor de la localización del proyecto



Nota: En esta figura se puede observar las antenas de telecomunicación situadas a 15 metros cada una, respecto a la localización del proyecto y sus dispositivos de transmisión. *Fuente:*

Autores

8.3.4. Relación Señal a Ruido (SNR)

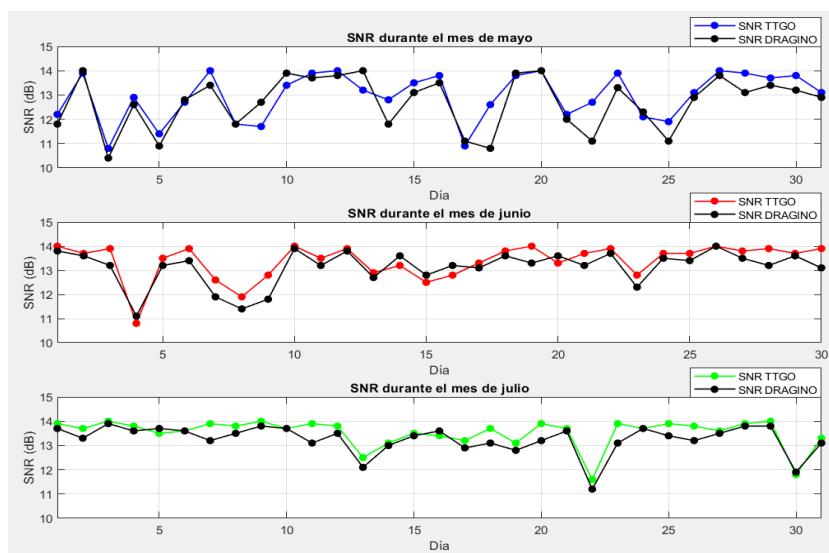
De acuerdo con (2CI Group, 2021) el SNR hace referencia a la relación mínima que existe entre la potencia de la señal recibida y el nivel de potencia de piso de ruido. El piso de ruido indica el área de todas aquellas fuentes de señales

que causan interferencias no deseadas y que pueden corromper señal transmitida (información) causando retransmisiones; en pocas palabras el SNR es un indicador de la calidad de la señal.

Para evaluar el SNR en el enlace LoRaWAN establecido con velocidad de datos adaptativa (ADR), se exportó la información proporcionada por ChirpStack y se sacó el promedio mensual de SNR obtenido durante el desarrollo del proyecto:

Gráfica 9

Promedio diario SNR durante los meses de trabajo



Nota: Los puntos corresponden al promedio diario de SNR para cada uno de los nodos implementos durante cada mes de trabajo. *Fuente:* Autores.

De acuerdo con la información suministrada en (LoRa Developer Portal, 2017), para cada factor de dispersión (SF) existe un límite de SNR por lo que, si se alcanza dicho valor, el receptor no será capaz de demodular la señal. En la Tabla 11 se presenta el límite SNR en función del SF.

Tabla 11

SNR en función del SF en enlaces LoRaWAN

SF	SF7	SF8	SF9	SF10	SF11	SF12
Límite SNR (dB)	-7.5	-10	-12.5	-15	-17.5	-20

Nota: Esta tabla indica el valor límite de relación señal a ruido para los diferentes factores de dispersión (SF) medido en decibeles (dB). Adaptado

de Predicting LoRaWAN capacity, 2017, (<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/predicting-lorawan-capacity/>)

Para los enlaces LoRa se establecen valores típicos de SNR que varían entre -20dB y +10dB por lo que, SNR cercanos a +10dB indica que la señal recibida está menos dañada, niveles SNR >10dB describe que existe buena conexión entre los dispositivos. Cabe resaltar que LoRa puede demodular señales que cuentan con un SNR -7,5 dB y -20 dB, es decir señales que operan por debajo del piso de ruido.

Ahora, de acuerdo con la información anterior, el enlace LoRaWAN presentó buena conexión puesto que, en mayor medida, el promedio de niveles de SNR diario – mensual durante el proyecto se mantuvo por encima de los +12dB, esto debido a que el enlace contó con un buen nivel de potencia de señal (RSSI) y que la distancia entre los dispositivos no fue mayor a 20 metros.

Sin embargo, durante los meses de mayo y junio hubo una leve caída en el nivel de dBs (provocando pequeñas perdidas de paquetes) debido a la configuración del enlace, puesto que se implementó velocidad de datos adaptativa y de acuerdo con LoRa Developer Portal dichas configuraciones de ADR pueden tener diferentes objetivos de redundancia, lo que produce una proporción más baja de recepción en el Gateway,

En general, el enlace LoRAWAN establecido presentó buena calidad puesto que se contó con valores RSSI > -67 dBm y SNR > -7dB.

8.4. Efecto de la Distancia en los Parámetros de Comunicación

Al interior de la casa donde se localizó el proyecto se obtuvo los siguientes valores referentes a RSSI, SNR, de acuerdo con el SF:

Para SF7

- RSSI: -53
- SNR: 13.5

Para SF10

- RSSI: -50
- SNR: 14.1

Este resultado indica que el radioenlace entre el Gateway y los nodos finales al interior de la vivienda presentó excelente calidad ya que LoRa está pensado para abarcar grandes kilómetros y durante el desarrollo del proyecto la distancia entre el Gateway y los nodos no fue más de 20 metros.

Concluido los meses de trabajo, se realizó una prueba de distancia entre el nodo final TTGO T-Beam y el Gateway con el objetivo de observar la calidad del enlace LoRaWAN utilizando SF7 y SF10. Esta prueba consistió en situar la TTGO en cuatro puntos específicos lejos del Gateway, a distancias considerables. La ubicación de estos puntos se basó en características como alta densidad urbana, y áreas abiertas.

Figura 87

Ubicación de los puntos de prueba en el municipio de Neiva, Huíja. Colombia (Google Maps)



Nota: Esta figura corresponde a la vista satelital (por medio de Google Maps) de la ubicación de los puntos de medición: La nomenclatura P1, P2, P3 y P4 hace referencia a los puntos donde se ubicó la TTGO, GT se refiere a la posición fija del Gateway. *Fuente:* Autores.

Durante la realización de la prueba, la señal del radio enlace estuvo expuesta a diferentes tipos de obstáculos, como árboles, casas, densidad de tráfico y personas variable, en áreas abiertas y las antenas de los dispositivos no contaron con línea de vista.

A continuación, se describe las coordenadas de localización (latitud y longitud) de cada uno de los puntos de prueba:

- GT: 2.9295798, -75.2737636
- P1: 2.9272695, -75.2735473
- P2: 2.9266329, -75.2760857
- P3: 2.9268656, -75.2789614
- P4: 2.9264575, -75.2817317

Para efectos prácticos, se consideró RSSI y SNR para evaluar la calidad del enlace LoRaWAN. A continuación, se presenta el resultado de la prueba realizada:

Tabla 12*Resultados obtenidos por el enlace LoRaWAN utilizando SF7*

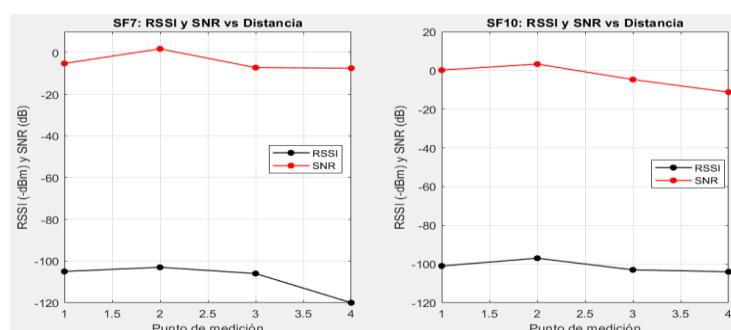
Distancia entre el Gateway y la TTGO t-Beam con SF7			
Punto	Distancia (Km)	RSSI (dBm)	SNR (dB)
P1	0.45	-105	-5.2
P2	0.5	-103	1.8
P3	0.9	-106	-7.2
P4	1.1	-----	-----

Nota: Esta tabla indica los parámetros de RSSI y SNR obtenidos con SF7 al ir aumentando la distancia entre el Gateway y la TTGO. *Fuente:* Autores.

Tabla 13*Resultados obtenidos por el enlace LoRaWAN utilizando SF10*

Distancia entre el Gateway y la TTGO t-Beam con SF10			
Punto	Distancia (Km)	RSSI (dBm)	SNR (dB)
P1	0.45	-101	0.2
P2	0.5	-97	3.3
P3	0.9	-103	-4.7
P4	1.1	-104	-11.2

Nota: Esta tabla indica los parámetros de RSSI y SNR obtenidos con SF10 al ir aumentando la distancia entre el Gateway y la TTGO. *Fuente:* Autores.

Gráfica 10*Parámetro RSSI y SNR vs Distancia*

Nota: Esta gráfica refleja el comportamiento de los parámetros RSSI y SNR respecto a los puntos de medición P1 (1), P2 (2), P3 (3), P4 (4), cuando se implementa el factor de dispersión (SF) 7 y 10.

Fuente: Autores.

La Grafica 10 contiene los resultados presentados en las Tablas 9 y 10, además representa la relación directa que existe entre la distancia y la reducción en la calidad del enlace de radio, pues tanto para SF7 como SF10, a medida que se alejó la TTGO del Gateway, el RSSI y SNR fueron afectados negativamente, por lo que los mensajes van a tardar un poco más en buscar sobre el aire la señal del Gateway.

A pesar de que la ubicación de la TTGO en los puntos P1 y P2 estaba a una distancia “cercana” del Gateway, la señal estuvo expuesta a diferentes tipos de obstáculos, como árboles, casas, densidad de tráfico y personas variable, lo cual generó la disminución de estos parámetros. Como se observa en las Tablas 11 y 12 el punto P1 tuvo peores valores de RSSI y SNR, pese a estar más cerca, debido a que la ubicación de este no tiene línea de vista con el Gateway

Los puntos P3 y P4 fueron áreas abiertas con mayor posibilidad para que la señal se transmitiera, pero al estar más lejos del Gateway la señal se vio afectada.

Sin embargo, a pesar de los resultados, para un SF7 el enlace LoRaWAN entre el Gateway y la TTGO en los cuatro puntos se mantuvo con rangos de $\text{RSSI} > -115\text{dBm}$ y $\text{SNR} > -7\text{dB}$, y de acuerdo con los rangos establecidos en la Tabla 7 y Tabla 8, para el caso en práctica (monitoreo y control de actuadores) se consideró el enlace como bueno, ya que la señal no contiene demasiada información y los intervalos de transmisión varían entre 30 segundos y 1 minuto.

Para el caso del SF10, el enlace LoRaWAN entre el Gateway y la TTGO en los puntos P1 y P2 presentó una calidad buena pues también se mantuvo dentro del rango de $\text{RSSI} > -115\text{dBm}$ y $\text{SNR} > -7\text{dB}$, pero a medida que la TTGO se alejó más del Gateway, es decir en los puntos P3 y P4, estos parámetros se encontraban en un rango límite, por lo que se consideró el enlace regular, pues la señal comenzará a verse afectada de forma leve por el ruido. De todos modos, este deterioro no fue perceptible de ninguna forma, ya que, el ancho de banda del Payload fue muy bajo como para verse afectado.

9. CONCLUSIONES

- Este proyecto contribuye a la identificación de parámetros, implementación y costos a considerar para llevar a cabo la aplicación de tecnología LoRa en el desarrollo de proyectos basados en IoT en Colombia. Es de resaltar que esta tecnología posee un enorme potencial en sistemas de monitoreo inteligente, además las características técnicas del equipo implementado, la calidad del enlace LoRa obtenido y la instalación rápida y sencilla, junto con las plataformas IoT actuales, lo convierte en una opción ideal a bajo costo para emplearse en el campo de la agricultura, aumentando la eficiencia de registro y control de procesos.
- La aplicación de electrónica en la industria agrícola representa una gran ventaja frente procesos tradicionales, y esto se evidenció en los resultados obtenidos en cada huerta frente al uso de agua para el riego de plantas. Gracias al monitoreo de humedad y temperatura de la huerta 1, se llevó un control de riego localizado mediante el método de riego por goteo ahorrando poco más del 20% de agua frente a la huerta 2 regada manualmente mediante pulverización. Lo que permitió observar claras ventajas en la salud general de las plantas. Ya que, las plantas del huerto 2 fueron mas susceptibles a contraer enfermedades, como consecuencia de la baja humedad en algunos puntos del día donde el calor fue extremo. Dicho factor fue determinante para un pequeño deterioro de las plantas del huerto 2, ya que, al encontrarse debilitadas fueron un foco para plagas oportunistas. Además, enfermedades por exceso de humedad también se presentaron de manera más marcada en la huerta 2. La principal causa para esto es que, algunos días se regó la huerta 2 por la mañana, y posteriormente comenzó a llover durante todo el día. Derivando así, en un exceso de humedad que causó la proliferación de hongos.
- De acuerdo con la prueba de distancia, se observó que con un factor de dispersión bajo (SF7), la señal no tuvo problemas para superar obstáculos (árboles, casas y pendientes asfaltadas) en un radio de medio kilómetro. Un factor de dispersión más bajo implica más “chirridos” por segundo, que a su vez conlleva a una mayor codificación de datos por unidad de tiempo. Sin embargo, vale la pena resaltar que el ancho de banda (10 bytes) utilizado por la carga útil (payload) fue sumamente bajo, por lo que, la disminución del ancho de banda que conlleva aumentar el factor de dispersión (SF10) no fue ningún problema. Ya que, esto le permitió a la red LPWAN alcanzar

distancias de comunicación fiables a más de 1 kilómetro dentro de un entorno urbano.

- Es muy importante la selección correcta de la clase de los nodos en una red basada en LoRaWAN. Debido a que esto limita la ventana de oportunidad en la que el nodo está disponible a recibir mensajes Downlink provenientes del Gateway. En el caso de la clase A se tiene que el nodo solo va a “escuchar” al Gateway durante un pequeño intervalo RX después que la transmisión haya tenido lugar. Por el contrario, en la clase C esta ventana de oportunidad está siempre disponible mientras no tenga lugar ningún evento de transmisión. En la práctica, no hubo mucha diferencia entre clases para el límite inferior (45%), ya que, la variable de humedad de suelo cambia muy lentamente, y 26 segundos no son determinantes para que un gran cambio tenga lugar. Por el contrario, en el límite superior si tuvo gran importancia, debido a que dichos 26 segundos representaron en promedio, un aumento de casi el 10% sobre el valor de humedad de suelo deseado. Otro factor importante a tener en cuenta para la correcta implementación de una red basada en LoRa, es conocer la compatibilidad entre las especificaciones existentes, ya que, existen nodos, gateways y firmare que no son compatibles entre sí.
- ChirpStack brinda una excelente infraestructura para el despliegue y desarrollo de proyectos a pequeña y mediana escala en entornos rurales, ya que, no necesita de conexión a internet para su ejecución, lo único que se necesita, es tener disponibilidad de energía eléctrica en la estación base, donde se encuentra el Gateway y los actuadores del proceso agrícola a automatizar. Toda la información de dicho proceso se puede almacenar en una base de datos local, que posteriormente puede ser almacenada en un servidor web. Por parte de los nodos, dependiendo del tipo de variable que se desea monitorear o controlar, pueden tener una autonomía de más de 2 años con una única carga, si se diseña un circuito específico para el bajo consumo energético. En el caso del presente proyecto, la autonomía del nodo que adquiría los datos fue de aproximadamente 21 hora, debido a que este es una tarjeta de desarrollo completa que cuenta con funcionalidades que no son necesarias para este caso en particular.

BIBLIOGRAFÍA

- 2CI Group. (22 de 03 de 2021). *Conceptos de actualidad: LoRa y LoRaWan*. Recuperado el 16 de 08 de 2021, de 2CI Group: <https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/>
- Agro Activo. (2021). *Aceite Mineral Citroemulsión*. Recuperado el 03 de 03 de 2021, de Agro Activo: <https://agroactivocol.com/producto/sanidad-vegetal-alimentos-saludables/aceite-mineral-citroemulsion/>
- Alliance, L. (2021). *About LoRaWAN*. Recuperado el 12 de 09 de 2020, de LoRa Alliance: <https://lora-alliance.org/about-lorawan/>
- B, E. (2018). *LoRa netwroks*. Recuperado el 03 de 12 de 2020, de LoRa: <https://lora.readthedocs.io/en/latest/>
- ChirpStack. (s.f.). *ChirpStack, open-source LoRaWAN Network Server stack*. Recuperado el 02 de 04 de 2021, de ChirpStack: <https://www.chirpstack.io/>
- Dow AgroSciences de Colombia S.A. (s.f.). *LORSBAN 4 EC*. Recuperado el 03 de 03 de 2021, de Dow AgroSciences: http://www.ghcia.com.co/plm/source/productos/3099_72_154.htm
- Dragino. (15 de 06 de 2020). *LT-33222-L LoRa I/O Controller*. Recuperado el 02 de 09 de 2020, de Dragino: <https://www.dragino.com/products/lora-lorawan-end-node/item/138-lt-33222-l.html>
- Guzmán L, A., Corradini S, F., Martínez C, J., Salinas P, L., Allende C, M., Olivares P, N., . . . Felmer E, S. (2017). *Manual de cultivo del tomate al aire libre*. Santiago, Chile.
- Liotta , M., Carrión , R., Ciancaglini, N., & Olguín, A. (2015). *Instituto Nacional de Tecnología Agropecuaria*. Recuperado el 02 de 11 de 2020, de Instituto Nacional de Tecnología Agropecuaria: https://inta.gob.ar/sites/default/files/inta_manual_riego_por_goteo.pdf
- LoRa Developer Portal. (2017). *Documentation*. Recuperado el 16 de 08 de 2021, de LoRa Developer Portal: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/predicting-lorawan-capacity/>
- Ma, Y.-W., & Chen, J.-L. (2018). *IEEEExplore*. Recuperado el 15 de 09 de 2020, de IEEEExplore: <https://ieeexplore.ieee.org/document/8394568>

- Martin, R. S. (15 de 10 de 2018). *IoT (Internet-of-Things) – El Internet de las “Cosas”*. Recuperado el 19 de 11 de 2020, de MICM máster en ingeniería de comunicaciones móviles: <http://micm.es/noticias/iot-el-internet-de-las-cosas/>
- Orson W, I., & Vaughn E, H. (03 de 2003). *Google Books*. Recuperado el 2 de 11 de 2020, de Google Books: <https://books.google.com.pe/books?id=vRtP-aU0QU8C&printsec=frontcover&hl=es#v=onepage&q&f=false>
- RAK. (2021). *RAK2287 Quick Start Guide*. Recuperado el 20 de 03 de 2021, de RAK Documentation Center: <https://docs.rakwireless.com/Product-Categories/WisLink/RAK2287/Quickstart/#prerequisites>
- Rivera Puentes, W. (2020). *Implementación de WPLAN para el envío de información remota en aplicaciones IoT*. Neiva, Huila, Colombia. Recuperado el 11 de 09 de 2020
- SonicWall. (20 de 12 de 2019). *Wireless: SNR, RSSI And Noise Basics Of Wireless Troubleshooting*. Recuperado el 15 de 08 de 2021, de SonicWall: <https://www.sonicwall.com/support/knowledge-base/wireless-snr-rssi-and-noise-basics-of-wireless-troubleshooting/180314090744170/>
- SpeedCheck. (s.f.). *RSSI*. Recuperado el 15 de 08 de 2021, de SpeedCheck: <https://www.speedcheck.org/wiki/rss/>
- Syngenta S. A. (s.f.). *AMISTAR 50 WG*. Recuperado el 03 de 03 de 2021, de Syngenta: http://www.ghcia.com.co/plm/source/productos/2461_131_169.htm
- Technical Marketing Workgroup 1.0. (11 de 2015). *A technical overview of LoRa and LoRaWAN*. Recuperado el 11 de 09 de 2020, de LoRa Alliance: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>
- The Things Network. (03 de 08 de 2021). *What are LoRa and LoRaWAN?* Recuperado el 03 de 08 de 2021, de The Things Network: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>
- Triana Useche, J., & Rodriguez Leguizamo, R. (2018). *Universidad Distrital Francisco José de Caldas*. Recuperado el 11 de 09 de 2020, de Universidad Distrital Francisco José de Caldas: <https://repository.udistrital.edu.co/handle/11349/13388>
- Underwater Arduino Data Loggers. (2021). *Hacking a Capacitive Soil Moisture Sensor (v1.2) for Frequency Output*. Recuperado el 15 de 02 de 2021, de Underwater Arduino Data Loggers: <https://thecavepearlproject.org/2020/10/27/hacking-a-capacitive-soil-moisture-sensor-for-frequency-output/>

wikipedia.org. (28 de 08 de 2021). *Espectro ensanchado*. Recuperado el 12 de 07 de 2021, de Wikipedia: https://es.wikipedia.org/wiki/Espectro_ensanchado

ANEXOS

Anexo A Código para realizar prueba del módulo GPS de la TTGO t-Beam

```
GPS
#define SERIAL1_RX 34 // GPS_TX -> 34
#define SERIAL1_TX 12 // 12 -> GPS_RX
void setup () {
    Serial.begin ( 115200 );
    Serial.println ( "PRUEBA GPS TTGO" );
    delay ( 2000 );
    Serial1.begin ( 9600 , SERIAL_8N1, SERIAL1_RX, SERIAL1_TX);
}
void loop () {
    if (Serial1.available ()) {
        Serial.write (Serial1.read ());
        Serial1.println ();
    }
}
```

Anexo B Código ABP para la TTGO t-Beam

```
*****
* Copyright (c) 2015 Thomas Telkamp and Matthijs Kooijman
* Copyright (c) 2018 Terry Moore, MCCI
*****/
#include <Arduino.h>
#include <SHT1x-ESP.h>
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>

#define dataPin 13
#define clockPin 14

// LoRaWAN NwkSKey usando el msb
static const PROGMEM u1_t NWKSKEY[16] = { 0x96, 0x21, 0xD0, 0xA2, 0x12, 0x4B, 0x1F, 0xCD, 0xD1,
0x8F, 0x80, 0xF7, 0xD0, 0x28, 0xE3, 0xEE };

// LoRaWAN AppSKey usando el msb
static const u1_t PROGMEM APPSKEY[16] = { 0x4F, 0x6E, 0xAD, 0xFF, 0x60, 0x33, 0x5E, 0x48, 0xC3,
0xC3, 0x08, 0xBD, 0x0F, 0x96, 0x73, 0x15 };

// Dirección del dispositivo final LoRaWAN (DevAddr) en msb
static const u4_t DEVADDR = 0x018fc83d ;

void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static uint8_t payload[10];
static osjob_t sendjob;

const int ValorAire1 = 2564;
const int ValorAire2 = 2587;
const int ValorAgua1 = 2136;
const int ValorAgua2 = 2025;
int HumedadSuelo1 = 0;
int HumedadSuelo2 = 0;
int porcentajeHumedad1 = 0;
```

```

int porcentajeHumedad2 =0;

// Intervalo de tiempo en segundos en el cual se van a leer las variables
const unsigned TX_INTERVAL = 90;

// Mapeo de pines
const lmic_pinmap lmic_pins = {
    .nss = 18,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 23,
    .dio = {/*dio0*/ 26, /*dio1*/ 33, /*dio2*/ 32}
};
SHT1x sht1x(dataPin, clockPin, SHT1x::Voltage::DC_3_3v);
void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));
            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
        case EV_JOINED:
            Serial.println(F("EV_JOINED"));
            break;
        case EV_JOIN_FAILED:
            Serial.println(F("EV_JOIN_FAILED"));
            break;
        case EV_REJOIN_FAILED:
            Serial.println(F("EV_REJOIN_FAILED"));
            break;
        case EV_TXCOMPLETE:
            Serial.println(F("---Transmisión Completada---"));
            if (LMIC.txrxFlags & TXRX_ACK)
                Serial.println(F("ACK Recibido"));
            if (LMIC.dataLen) {
                Serial.println(F("Recibido "));
                Serial.println(LMIC.dataLen);
                Serial.println(F(" bytes de payload"));
            }
            // Programar la próxima transmisión
            os_setTimedCallback(&sendjob, os_getTime() + sec2osticks(TX_INTERVAL), do_send);
            break;
        case EV_LOST_TSYNC:
            Serial.println(F("EV_LOST_TSYNC"));
            break;
        case EV_RESET:
            Serial.println(F("EV_RESET"));
            break;
        case EV_RXCOMPLETE:
            // data received in ping slot
    }
}

```

```

        Serial.println(F("EV_RXCOMPLETE"));
        break;
    case EV_LINK_DEAD:
        Serial.println(F("EV_LINK_DEAD"));
        break;
    case EV_LINK_ALIVE:
        Serial.println(F("EV_LINK_ALIVE"));
        break;

    case EV_TXSTART:
        Serial.println(F("Transmisión Iniciada..."));
        break;
    case EV_TXCANCELED:
        Serial.println(F("EV_TXCANCELED"));
        break;
    case EV_RXSTART:
        /* no imprimir nada */
        break;
    case EV_JOIN_TXCOMPLETE:
        Serial.println(F("EV_JOIN_TXCOMPLETE: no se aceptó la union"));
        break;
    default:
        Serial.print(F("Evento desconocido: "));
        Serial.println((unsigned) ev);
        break;
    }
}

float humedad1(){
    HumedadSuelo1 = analogRead(25); //Lee un valor analógico entre 0 y 4095 que corresponde a la
    //humedad de suelo en la huerta 1
    //Mapea los valores obtenidos anteriormente en un rango de 0 a 100
    porcentajeHumedad1 = map(HumedadSuelo1, ValorAire1, ValorAgua1, 0, 100);
    if(porcentajeHumedad1 >= 100){
        porcentajeHumedad1 =100;
    }
    else if(porcentajeHumedad1 <=0){
        porcentajeHumedad1 =0;
    }
    else if(porcentajeHumedad1 >0 && porcentajeHumedad1 < 100){
    }
    return porcentajeHumedad1;
}

float humedad2(){
    HumedadSuelo2 = analogRead(15); //Lee un valor analógico entre 0 y 4095 que corresponde a la
    //humedad de suelo en la huerta 2
    porcentajeHumedad2 = map(HumedadSuelo2, ValorAire2, ValorAgua2, 0, 100);
    if(porcentajeHumedad2 >= 100){
        porcentajeHumedad2 =100;
    }
    else if(porcentajeHumedad2 <=0){
        porcentajeHumedad2 =0;
    }
    else if(porcentajeHumedad2 >0 && porcentajeHumedad2 < 100){
    }
    return porcentajeHumedad2;
}

void do_send(osjob_t* j){
    // Comprobar si no hay un trabajo TX / RX actual en ejecución
    if (LMIC.opmode & OP_TXRXPEND) {

```

```

        Serial.println(F("OP_TXRXPEND, No enviado"));
    }
else {
    // Lee la temperatura del SHT10
    float temperatura = sht1x.readTemperatureC();
    Serial.print("Temperatura: "); Serial.print(temperatura);
    Serial.println("°C");
    // Ajuste para el rango f2sflt16 (-1 a 1)
    temperatura = temperatura / 100;

    // Lee la humedad del SHT10
    float HumedadRelativa = sht1x.readHumidity();
    Serial.print("Humedad Relativa: "); Serial.print(HumedadRelativa);
    Serial.println("%");
    // Ajuste para el rango f2sflt16 (-1 a 1)
    HumedadRelativa = HumedadRelativa / 100;

    float humSuelo1 = humedad1();
    Serial.print("Humedad de Suelo 1: ");Serial.print(humSuelo1);
    Serial.println("%");
    humSuelo1 = humSuelo1 / 100;

    float humSuelo2 = humedad2();
    Serial.print("Humedad de Suelo 2: ");Serial.print(humSuelo2);
    Serial.println("%");
    humSuelo2 = humSuelo2 / 100;

    // float -> int
    // Esto usa el datum sflt16 (https://github.com/mcci-catena/arduino-lmic#sflt16)
    uint16_t payloadTemp = LMIC_f2sflt16(temperatura);
    // int -> bytes
    byte tempBaja = lowByte(payloadTemp);
    byte tempAlta = highByte(payloadTemp);
    // colocar los bytes en la carga útil
    payload[0] = tempBaja;
    payload[1] = tempAlta;

    // float -> int
    uint16_t payloadHum = LMIC_f2sflt16(HumedadRelativa);
    // int -> bytes
    byte humBaja = lowByte(payloadHum);
    byte humAlta = highByte(payloadHum);
    payload[2] = humBaja;
    payload[3] = humAlta;

    uint16_t payloadSuelo1 = LMIC_f2sflt16(humSuelo1);
    // int -> bytes
    byte sueloBajo1 = lowByte(payloadSuelo1);
    byte sueloAlto1 = highByte(payloadSuelo1);
    // colocar los bytes en la carga útil
    payload[4] = sueloBajo1;
    payload[5] = sueloAlto1;

    uint16_t payloadSuelo2 = LMIC_f2sflt16(humSuelo2);
    // int -> bytes
    byte sueloBajo2 = lowByte(payloadSuelo2);
    byte sueloAlto2 = highByte(payloadSuelo2);
    // colocar los bytes en la carga útil
    payload[6] = sueloBajo2;
    payload[7] = sueloAlto2;
}

```

```

    //Preparar la transmisión de datos en sentido ascendente en el próximo momento posible.
    // Transmite en el puerto 1 (el primer parámetro); se puede utilizar cualquier valor entre 1 y 223 (los
demás
están reservados).
    LMIC_setTxData2(1, payload, sizeof(payload)-1, 0);
}
// La próxima transmisión está programada después del evento TX_COMPLETE.
}

void setup() {
// pinMode(13, OUTPUT);
while (!Serial); // Espera a que se inicialice el Serial
Serial.begin(115200);
analogReadResolution(12);
delay(100);
Serial.println(F("COMENZANDO TRANSMISIÓN"));

#ifndef VCC_ENABLE
pinMode(VCC_ENABLE, OUTPUT);
digitalWrite(VCC_ENABLE, HIGH);
delay(1000);
#endif

// LMIC init
os_init();
// Restablecer el estado de MAC. Se descartarán las transferencias de datos de sesión y pendientes.
LMIC_reset();

// Establece parámetros de sesión estáticos. En lugar de establecer una sesión de forma dinámica
// al unirse a la red, se proporcionan parámetros de sesión precalculados.
#ifndef PROGMEM
// En AVR, estos valores se almacenan en flash y solo se copian en la RAM una vez.
uint8_t appskey[sizeof(APPSKEY)];
uint8_t nwkskey[sizeof(NWKSKEY)];
memcpy_P(appskey, APPSKEY, sizeof(APPSKEY));
memcpy_P(nwkskey, NWKSKEY, sizeof(NWKSKEY));
LMIC_setSession (0x13, DEVADDR, nwkskey, appskey);
#else
// Si no está ejecutando un AVR con PROGMEM, simplemente utilizar las matrices directamente
LMIC_setSession (0x13, DEVADDR, NWKSKEY, APPSKEY);
#endif

#if defined(CFG_eu868)
// Configurar los canales utilizados por ChirStack o Things Network, que corresponda
// a los valores predeterminados de la mayoría de las puertas de enlace.
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI); // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK, DR_FSK), BAND_MILLI); // g2-band
#endif || defined(CFG_us915) || defined(CFG_au915)
LMIC_selectSubBand(1);
#endif || defined(CFG_as923)
#endif || defined(CFG_kr920)
#endif || defined(CFG_in866)
#else
#error Region not Soportado
#endif

```

```

LMIC_setLinkCheckMode(0);

LMIC.dn2Dr = DR_SF9;

// Establecer la velocidad de datos y la potencia de transmisión para el enlace ascendente
LMIC_setDrTxpow(DR_SF10,15);

// Iniciar el proceso
do_send(&sendjob);
}

void loop() {
    unsigned long now;
    now = millis();
    if ((now & 512) != 0) {
        digitalWrite(13, HIGH);
    }
    else {
        digitalWrite(13, LOW);
    }

    os_runloop_once();
}

```

Anexo C Código Upnlink Dragino LT33222-L implementado en Chirpstack

```

*****
* Copyright ©2010-2021
* Shenzhen Dragino technology development co.LTD
* Dragino Technology Co., LTD.
*****/

```

```

// Formatos de carga útil de los dispositivos LT33222 o LT22222
var hardware= (bytes[10] & 0xC0)>>6;
var mode0= bytes[10] & 0xff;
var mode= bytes[10] & 0x3f;
var decode = {};

if(hardware=='0')
{
    decode.Hardware_mode="LT33222";
    decode.DO3_status=(bytes[8] &0x04)? "L":"H";
    if(mode0=='1')
    {
        decode.DI3_status= (bytes[8] &0x20)? "H":"L";
    }
}
else if(hardware=='1')
{
    decode.Hardware_mode= "LT22222";
}

if(mode!=6)
{
    decode.DO1_status= (bytes[8] &0x01)? "L":"H";
    decode.DO2_status= (bytes[8] &0x02)? "L":"H";
    decode.R01_status= (bytes[8] &0x80)? "ON":"OFF";
    decode.R02_status= (bytes[8] &0x40)? "ON":"OFF";
    if(mode!=1)
    {
        if(mode!=5)
        {

```

```

        decode.Count1_times= (bytes[0]<<24 | bytes[1]<<16 | bytes[2]<<8 | bytes[3]);
    }
    decode.First_status= (bytes[8] &0x20)? "Yes":"No";
}
}

if(mode=='1')
{
    decode.Work_mode= "2ACI+2AVI";
    decode.AVI1_V= parseFloat(((bytes[0]<<24>>16 | bytes[1])/1000).toFixed(3));
    decode.AVI2_V= parseFloat(((bytes[2]<<24>>16 | bytes[3])/1000).toFixed(3));
    decode.AC1_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));
    decode.AC2_mA= parseFloat(((bytes[6]<<24>>16 | bytes[7])/1000).toFixed(3));
    decode.DI1_status= (bytes[8] &0x08)? "H":"L";
    decode.DI2_status= (bytes[8] &0x10)? "H":"L"
}
else if(mode=='2')
{
    decode.Work_mode= "Count mode 1";
    decode.Count2_times= (bytes[4]<<24 | bytes[5]<<16 | bytes[6]<<8 | bytes[7]);
}
else if(mode=='3')
{
    decode.Work_mode= "2ACI+1Count";
    decode.AC1_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));
    decode.AC2_mA= parseFloat(((bytes[6]<<24>>16 | bytes[7])/1000).toFixed(3));
}
else if(mode=='4')
{
    decode.Work_mode= "Count mode 2";
    decode.Acount_times= (bytes[4]<<24 | bytes[5]<<16 | bytes[6]<<8 | bytes[7]);
}
else if(mode=='5')
{
    decode.Work_mode= " 1ACI+2AVI+1Count";
    decode.AVI1_V= parseFloat(((bytes[0]<<24>>16 | bytes[1])/1000).toFixed(3));
    decode.AVI2_V= parseFloat(((bytes[2]<<24>>16 | bytes[3])/1000).toFixed(3));
    decode.AC1_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));
    decode.Count1_times= bytes[6]<<8 | bytes[7];
}
else if(mode=='6')
{
    decode.Work_mode= "Exit mode";
    decode.Mode_status= bytes[9] ? "True":"False";
    decode.AV1L_flag= (bytes[0] &0x80)? "True":"False";
    decode.AV1H_flag= (bytes[0] &0x40)? "True":"False";
    decode.AV2L_flag= (bytes[0] &0x20)? "True":"False";
    decode.AV2H_flag= (bytes[0] &0x10)? "True":"False";
    decode.AC1L_flag= (bytes[0] &0x08)? "True":"False";
    decode.AC1H_flag= (bytes[0] &0x04)? "True":"False";
    decode.AC2L_flag= (bytes[0] &0x02)? "True":"False";
    decode.AC2H_flag= (bytes[0] &0x01)? "True":"False";
    decode.AV1L_status= (bytes[1] &0x80)? "True":"False";
    decode.AV1H_status= (bytes[1] &0x40)? "True":"False";
    decode.AV2L_status= (bytes[1] &0x20)? "True":"False";
    decode.AV2H_status= (bytes[1] &0x10)? "True":"False";
    decode.AC1L_status= (bytes[1] &0x08)? "True":"False";
    decode.AC1H_status= (bytes[1] &0x04)? "True":"False";
    decode.AC2L_status= (bytes[1] &0x02)? "True":"False";
    decode.AC2H_status= (bytes[1] &0x01)? "True":"False";
    decode.DI2_status= (bytes[2] &0x08)? "True":"False";
    decode.DI2_flag= (bytes[2] &0x04)? "True":"False";
}

```

```

decode.DI1_status= (bytes[2] &0x02)? "True":"False";
decode.DI1_flag= (bytes[2] &0x01)? "True":"False";
}

if(bytes.length==11)
{
    return decode;
}
}

```

Anexo D Código Uplink TTGO t-Beam implementado en Chirpstack

```

/****************************************************************************
 * Copyright ©2012
 * Adafruit Industries
 ****/
// Decodificador para ChirpStack
function Decode(fport,bytes) {
    // Decodifica un mensaje de enlace ascendente desde un búfer
    var decoded = {};

    // Temperatura
    rawTemp = bytes[0] + bytes[1] * 256;
    decoded.Temperatura = sflt162f(rawTemp) * 100;

    // Humedad de Aire
    rawHumed = bytes[2] + bytes[3] * 256;
    decoded.Humedad = parseInt(sflt162f(rawHumed) * 100);

    // Humedad de Suelo 1
    rawSuelo1 = bytes[4] + bytes[5] * 256;
    decoded.HumedadSuelo1 = parseInt(sflt162f(rawSuelo1) * 100);

    // Humedad de Suelo 2
    rawSuelo2 = bytes[6] + bytes[7] * 256;
    decoded.HumedadSuelo2 = parseInt(sflt162f(rawSuelo2) * 100);

    return decoded;
}

function sflt162f(rawSflt16)
{
    // Desecha los bits altos para mejorar la repetibilidad
    rawSflt16 &= 0xFFFF;

    // Caso especial menos cero
    if (rawSflt16 == 0x8000)
        return -0.0;

    // Extrae el signo
    var sSign = ((rawSflt16 & 0x8000) !== 0) ? -1 : 1;

    // Extrae el exponente
    var exp1 = (rawSflt16 >> 11) & 0xF;

    // Extrae la parte fraccionaria
    var mant1 = (rawSflt16 & 0x7FF) / 2048.0;

    // Convierte de nuevo a un número de coma flotante.
    var f_unscaled = sSign * mant1 * Math.pow(2, exp1 - 15);

    return f_unscaled;
}

```