



UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Mecatrónica

Red IoT comunitaria con tecnología LoRa para cultivos urbanos inteligentes: Aplicación

Trabajo de grado previo a la obtención del título de Ingeniero en Mecatrónica

Autor:

Ismael Alexander Loza Reinoso

Director:

Dr. Carlos Xavier Rosero Chandi.

Ibarra - Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100406923-1		
APELLIDOS Y NOMBRES:	Loza Reinoso Ismael Alexander		
DIRECCIÓN:	Pilanquí, psj E, Manzana 22, casa 2-68, Ibarra		
EMAIL:	ialozar@utn.edu.ec		
TELÉFONO FIJO:	062603899	TELÉFONO MÓVIL:	0986651947

DATOS DE LA OBRA	
TÍTULO:	Red IoT comunitaria con tecnología LoRa para cultivos urbanos inteligentes: Aplicación
AUTOR (ES):	Loza Reinoso Ismael Alexander
FECHA DE APROBACIÓN: DD/MM/AAAA	19/07/2024
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
ASESOR /DIRECTOR:	Ing. Xavier Rosero Chandi , Dr. / Ing.Milton Gavilánez Villalobos, MSc

2. CONSTANCIAS

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

Ibarra, a los 19 días del mes de julio de 2024

EL AUTOR:

(Firma).....

Nombre: Loza Reinoso Ismael Alexander



Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN
CURRICULAR

En mi calidad de director del trabajo de grado “RED IOT COMUNITARIA CON TECNOLOGÍA LORA PARA CULTIVOS URBANOS INTELIGENTES: APLICACIÓN”, presentado por el egresado Ismael Alexander Loza Reinoso, que opta por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 19 de julio de 2024




Dr. Carlos Xavier Rosero Chandi

Director de Tesis




Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación: Red IoT comunitaria con tecnología LoRa para cultivos urbanos inteligentes: Aplicación. Elaborado por Ismael Alexander Loza Reinoso, previo a la obtención del título de Ingeniero en Mecatrónica, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:


Dr. Carlos Xavier Rosero Chandi.

Director de Tesis


MSc. Alejandro Milton Gavilánez Villalobos
Asesor de Tesis

Dedicatorias

Dedico este trabajo a mi familia, quienes son el corazón de mi vida. A mis padres, cuyo amor y sacrificio me han dado la fuerza para perseguir mis sueños. A mis hermanos, por su apoyo inquebrantable y por ser siempre fuente de motivación y alegría.

A Dios, por iluminar mi camino y brindarme las oportunidades que han moldeado mi ser.

Y con profundo amor, dedico especialmente este logro a mi novia, Estefi, quien ha sido mi compañera incansable, mi confidente y mi mayor aliento en este viaje. Tu presencia ha sido mi calma en la tormenta y mi luz en la oscuridad.

Agradecimientos

Ante todo, deseo expresar mi más profunda gratitud a Dios, por darme la fortaleza y la inspiración necesaria para completar esta jornada y por guiarme en cada paso de este camino.

Mi reconocimiento eterno a mis queridos padres, Manolo y Erminia, cuyo amor, apoyo incondicional y sacrificios han sido el pilar fundamental de mi vida y estudios. Gracias por creer en mí y por enseñarme el valor del esfuerzo y la perseverancia.

A mis hermanos, Sebastian y Adrián, les agradezco por su constante aliento y por estar siempre presentes cuando más los necesité. Su compañía y apoyo han sido indispensables para mí.

No puedo dejar de agradecer a mis docentes, quienes no solo me transmitieron su conocimiento sino que también inspiraron y motivaron mi crecimiento académico y personal. En especial, quiero mencionar a [nombres de docentes específicos si lo deseas], cuya guía fue crucial en el desarrollo de este trabajo.

Finalmente, un agradecimiento especial a mi novia, Estefi, por su amor, paciencia y comprensión. Gracias por estar a mi lado en los momentos de estrés y por celebrar cada pequeño éxito conmigo. Tu apoyo ha sido un regalo invaluable en mi vida.

Índice general

Cesión de derechos de autor a favor de la Universidad Técnica del Norte	II
Declaración	III
Certificación del director del trabajo de grado	IV
Dedicatorias	V
Agradecimientos	VI
Índice general	VII
Índice de figuras	XI
Índice de tablas	XIII
Resumen	XIV
Abstract	XVI
I Introducción	1
1.1 Planteamiento del problema	1

1.2	Objetivos	2
1.2.1	Objetivo general	2
1.2.2	Objetivos específicos	2
1.3	Justificación	3
1.4	Alcance	4
II	Revisión Literaria	6
2.1	Estado del Arte	6
2.2	Tecnologías	8
2.2.1	LoRa	8
2.2.2	The Things Network	15
2.2.3	Dragino	16
2.2.4	Sistema Operativo GNU Linux	17
2.2.5	Google Cloud	17
2.2.6	Apache	18
2.2.7	MySQL	18
2.2.8	Firebase	19
2.2.9	Python	20
2.3	Conclusión del Estado del Arte	21
III	Desarrollo	23
3.1	Arquitectura	24
3.1.1	Nodo Final	24

3.1.2	Gateway	26
3.1.3	Conexión con el Servidor Propietario	29
3.1.4	Servidor Puente	31
3.1.5	Base de Datos en Tiempo Real	40
3.1.6	Interfaz Gráfica	44
IV	Pruebas de Funcionamiento	46
4.1	Descripción de las Pruebas	47
4.2	Pruebas de Funcionamiento	47
4.3	Pruebas de Rendimiento	53
4.3.1	Uso del CPU	53
4.3.2	Uso de Memoria	54
4.3.3	Tráfico de Red	56
4.3.4	Uso de Recursos Externos	57
4.3.5	IOPS del Disco	58
4.3.6	Capacidad de Procesamiento del Disco	59
V	Conclusiones y Trabajo Futuro	61
5.1	Conclusiones	61
5.2	Recomendaciones	62
5.3	Trabajo a Futuro	63
	Anexos	72

A	Configuración de la Base de Datos MySQL	73
B	Script PHP para Insertar Datos del Sensor DHT	74
C	Script Python para la Interconexión de Datos	76
D	Script Python para Graficar Temperatura	78
E	Script Python para Graficar el Canal ADC-CH0V	80
F	Script Python para Graficar el Canal ADCCH1V	82
G	Script MyQSL para Crear la Tabla	84
H	Script MySQL Para Crear el Usuario	85

Índice de figuras

2.1	Arquitectura LoRa [1].	11
2.2	Aplicaciones LoRa [2].	13
3.1	Arquitectura de la red.	24
3.2	Ventana principal del Gateway.	27
3.3	Configuración del Gateway.	28
3.4	Configuración LoRaWAN.	28
3.5	Identificaciones del Gateway.	29
3.6	Salida de la consola SSH.	34
3.7	Estado del servicio Apache2.	35
3.8	Estado del servicio PHP 8.0-FPM	36
3.9	Diagrama de flujo de la configuración para la conexión a la base de datos MySQL	37
3.10	Diagrama de flujo del proceso de recepción y almacenamiento de datos en MySQL	38
3.11	Política de Firewall	39
3.12	Estructura de las variables en Firebase	40
3.13	Reglas para lectura y escritura de variables en Firebase	41
3.14	Flujo de datos desde MySQL a Firebase mediante un script de Python	42

3.15	Diagrama de flujo del proceso de sincronización de datos con MySQL y Firebase	43
3.16	Diagrama de flujo del monitoreo y actualización de datos.	45
4.1	Verificación de llegada de datos.	48
4.2	Salida SSH del archivo Python en ejecución.	49
4.3	Gráfica en tiempo real de la Temperatura.	50
4.4	Gráfica en tiempo real de la humedad del aire.	50
4.5	Gráfica en tiempo real de la humedad del suelo.	51
4.6	Latencia de la red	52
4.7	Uso del CPU de la máquina virtual	53
4.8	Uso de memoria de la máquina virtual	55
4.9	Tráfico de red de la máquina virtual	57
4.10	Uso de recursos externos	58
4.11	IOPS del disco	58
4.12	Capacidad de procesamiento del disco.	59

Índice de tablas

3.1	Configuraciones iniciales para el nodo LSN50	25
3.2	Configuraciones iniciales para el nodo LSN50-1	25
3.3	Uso de los sensores en la red IoT	25
3.4	Información general del dispositivo	30
3.5	Información general del segundo dispositivo	30
3.6	Configuraciones iniciales del nodo LoRaWAN	30
3.7	Configuración del Webhook	31
3.8	Información básica de la instancia loratest-1.	32
3.9	Configuración detallada de la máquina.	32
3.10	Configuración de firewalls para tráfico web.	33
3.11	Configuración del disco de arranque	33
3.12	Configuraciones de seguridad de contraseña en MySQL	39

Resumen

En respuesta a los desafíos de la agricultura urbana, esta investigación desarrolla una red IoT basada en la tecnología LoRa, orientada a optimizar el monitoreo y gestión de cultivos en zonas urbanas. La motivación detrás de este estudio radica en la necesidad urgente de mejorar la eficiencia agrícola mediante tecnologías avanzadas que permitan un manejo preciso de los recursos y una mayor productividad.

El objetivo principal se centra en la implementación de una puerta de enlace para la red IoT que facilite la recolección y análisis de datos en tiempo real. La metodología adoptada incluye una revisión literaria sobre IoT y LoRa, seguida del diseño de la red, y una serie de pruebas de funcionamiento para validar su eficacia.

El desarrollo de la red se detalla a través de la configuración de los nodos, la integración del gateway y la interacción con servidores que procesan y almacenan datos. Se realizaron simulaciones para evaluar la respuesta de la red bajo diferentes condiciones operativas, demostrando su capacidad para adaptarse a los entornos urbanos y mejorar la gestión de los cultivos.

La investigación concluye que la red no solo facilita el monitoreo preciso y en tiempo real, sino que también evidencia su potencial para ser aplicada en otras áreas urbanas, escalando la solución a un nivel más amplio. Se sugiere la exploración futura de nuevas aplicaciones

y mejoras tecnológicas para expandir la funcionalidad y el alcance de la red, destacando su relevancia en el marco de las ciudades inteligentes y la agricultura sostenible.

Palabras clave : LoRa en IoT, Agricultura Urbana, Monitoreo de Cultivos, Redes Inalámbricas.

Abstract

In response to the challenges of urban agriculture, this research develops an IoT network based on LoRa technology, aimed at optimizing the monitoring and management of crops in urban areas. The motivation behind this study stems from the urgent need to enhance agricultural efficiency through advanced technologies that allow for precise resource management and increased productivity.

The primary goal focuses on implementing a gateway for the IoT network that facilitates real-time data collection and analysis. The methodology adopted includes a literature review on IoT and LoRa, followed by the network design, and a series of operational tests to validate its effectiveness.

The network development is detailed through the configuration of nodes, the integration of the gateway, and interaction with servers that process and store data. Simulations were conducted to assess the network's response under different operational conditions, demonstrating its ability to adapt to urban environments and improve crop management.

The research concludes that the network not only facilitates precise and real-time monitoring but also demonstrates its potential to be applied in other urban areas, scaling the solution to a broader level. Future exploration of new applications and technological improvements to expand

the functionality and reach of the network is suggested, highlighting its relevance in the context of smart cities and sustainable agriculture.

Keywords: LoRa in IoT, Urban Agriculture, Crop Monitoring, Wireless Network.

Capítulo I

Introducción

1.1. Planteamiento del problema

La escasez de alimentos debido al crecimiento elevado de la población se presenta como el espacio propicio para desarrollar nuevas tecnologías. La agricultura urbana surge como una estrategia efectiva para la familia y la comunidad al momento de acceder a alimentos frescos, nutritivos y saludables, mediante el cultivo en pequeños espacios de tierra [3]. Cuba es el ejemplo más significativo en América Latina ya que el gobierno fomenta la creación de huertos urbanos para cubrir la demanda de alimentos [4].

En el contexto local se ha incentivado la creación de huertos urbanos para el cultivo de hortalizas de consumo familiar, como una alternativa de alimentación sustentable y saludable [5]. Por otro lado, existen trabajos de grado donde se implementan sistemas de monitoreo basados en IoT para cultivos, sin embargo, no existe una red de acceso público que permita monitorear cultivos urbanos para mejorar la producción y la calidad de los productos [6].

Las diversas aplicaciones de IoT (Internet de las Cosas) y comunicación LoRa (Long Range) ayudan a resolver problemas relacionados con la agricultura originados por el uso poco frecuente de tecnología y la gran demanda de productos[7]. Gracias a la aceptación de esta tecnología de comunicación es posible implementar una red IoT que mejore el proceso de suministro de nutrientes y crecimiento de cultivos, mediante la manipulación y análisis de datos [8].

A nivel local, se promueve la implementación de huertos urbanos como una alternativa sustentable y saludable para la producción de hortalizas destinadas al consumo familiar. Sin embargo, se señala la ausencia de una red de acceso público que permita el monitoreo de los cultivos urbanos con el fin de mejorar tanto la productividad como la calidad de los productos. En este contexto, se destaca el potencial de las tecnologías de IoT y la comunicación LoRa para abordar los desafíos agrícolas, facilitando la implementación de una red IoT que optimice el suministro de nutrientes y el crecimiento de los cultivos a través del análisis y la manipulación de datos.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar la puerta de enlace para una red IoT con tecnología LoRa para el monitoreo de cultivos en áreas urbanas.

1.2.2. Objetivos específicos

Proponer un modelo de puerta de enlace LoRa.

Desarrollar el sistema de gestión para la conexión de la red LoRa a Internet.

Someter la red a condiciones reales de trabajo

1.3. Justificación

La industria 4.0 actualmente está presente a la mayor parte de países desarrollados y cada vez llega a más lugares, el internet de las cosas es parte fundamental de este tipo de industria [9]. En la ciudad de Ibarra son pocos los lugares que aplican este tipo de tecnología, a pesar de los beneficios que esta brinda, en el contexto educativo y universitario es importante implementarlo primero como una herramienta educativa, y de forma progresiva aplicarla en otras áreas, como la agricultura, para realizar el monitoreo en tiempo real de los cultivos, mediante la adquisición de datos actualizados.

Los beneficios de la tecnología LoRa han despertado un gran interés en esta tecnología para la monitorización en tiempo real de variables ambientales, especialmente en el ámbito de la agricultura. Esto permitiría tener un control sobre las condiciones climáticas de las áreas desde cualquier lugar, facilitando la regulación de las condiciones en beneficio de los cultivos y mejorando la calidad de los productos finales. Además, a diferencia de otras tecnologías emergentes basadas en IoT, LoRa nos brinda la posibilidad de crear completamente los diferentes elementos de la red y personalizarla según nuestras necesidades [10].

La importancia de implementar esta red en la agricultura es demostrar que se puede usar tecnología inalámbrica en cultivos de áreas urbanas y monitorear espacios estratégicos del suelo. Para que esto sea posible es necesario utilizar tecnología LoRa debido a las grandes ventajas que

ofrece. La gran ventaja que LoRa ofrece es que con un solo Gateway y diferentes protocolos de comunicación se puede transmitir los datos a un servidor en la nube [11]. Si este tipo de aplicaciones se diversifican y extrapolan a otros campos de la vida cotidiana se necesita de redes confiables de no pago para enviar la información a través de IoT con el fin de usarla en la creación de modelos inteligentes de control automático.

La importancia de implementar esta red para los estudiantes de ingeniería mecatrónica radica en enseñar la aplicación de las técnicas y conocimientos adquiridos en la universidad a lo largo de los años. Además, brinda a los estudiantes la oportunidad de desarrollar actitudes y aptitudes necesarias para aplicarlas en el área de interés que se desempeñarán como futuros profesionales.

1.4. Alcance

El trabajo de grado busca desarrollar una red IoT con tecnología LoRa con conectividad de libre acceso que se pruebe en un área de muestra para su posterior implementación en un área metropolitana.

Se desarrolla una puerta de enlace para conectar dispositivos finales preexistentes que miden variables de interés de un cultivo, con una base de datos en la nube mediante el protocolo de comunicación LoRa.

La información obtenida y los datos recolectados se almacenan en una base de datos con la ayuda de plataformas digitales las cuales son útiles para enlazar aplicaciones web y móviles, las cuales permiten al usuario acceder a una interfaz gráfica con un informe detallado del estado de

los sensores.

Mediante pruebas de red a condiciones de trabajo permite determinar los límites del sistema a implementar.

Capítulo II

Revisión Literaria

Este capítulo explora el contexto teórico y práctico en el que se enmarca el proyecto. Se abordan estudios anteriores que no solo evalúan la viabilidad de la propuesta, sino también justifican su aplicación en entornos urbanos. Se analizan las soluciones más efectivas identificadas por la comunidad científica en el ámbito de Internet de las Cosas (IoT), con un enfoque particular en la agricultura urbana. Además, se discuten las tecnologías y herramientas predominantes que facilitan el monitoreo de variables críticas en la agricultura, destacando cómo contribuyen a la mejora de la precisión y eficiencia en la gestión de cultivos urbanos.

2.1. Estado del Arte

Un estudio de Samuel Aguilar, en [12], demuestra la eficacia de una red IoT que utiliza tecnología LoRa. En este estudio se desarrolla un sistema de control de riego en un cultivo en Perú, operado por nodos finales que monitorean constantemente las condiciones. Los datos

obtenidos dirigen el riego adaptado a las necesidades del cultivo, lo que optimiza el uso del agua y promueve la salud de las plantas. Verificaciones a través de simulaciones confirman la operatividad del sistema antes de su implementación en campo. Aguilar afirma que el uso de IoT con LoRaWAN mejora notablemente el monitoreo y control de cultivos agrícolas, ofreciendo una herramienta valiosa para la agricultura inteligente.

En un contexto distinto, Paúl Salazar y su equipo, en [13], utilizan la tecnología LoRa en un cultivo de lechugas para gestionar el estrés abiótico. Monitorean condiciones ambientales como la humedad y la temperatura, permitiendo ajustes precisos en las prácticas de riego. Este estudio resalta la eficacia de LoRa para aplicaciones de larga duración con bajo consumo energético.

Además, Hakim et al., en [14], extienden el uso de la tecnología LoRa fuera del ámbito agrícola, aplicándola en la optimización de redes de sensores en zonas forestales complejas. Desarrollan un modelo basado en el sistema Fuzzy ANFIS que mejora la precisión en la predicción de pérdidas de señal en áreas densamente arboladas, demostrando la capacidad de LoRa para superar obstáculos físicos y ambientales en la comunicación inalámbrica.

Estos estudios ilustran la diversidad de aplicaciones de la tecnología LoRa y su impacto potencial en varios campos críticos, desde la agricultura hasta la gestión ambiental, mostrando su relevancia creciente en el mundo de IoT y las ciudades inteligentes.

2.2. Tecnologías

Para el desarrollo de aplicaciones LoRa, existen varias herramientas y plataformas que proporcionan un conjunto completo de características para diseñar, implementar y gestionar redes IoT. A continuación, se listan, describen y analizan las principales herramientas y plataformas disponibles.

2.2.1. LoRa

La tecnología LoRa (Long Range) en el ámbito de Internet de las Cosas (IoT) se ha consolidado como una de las soluciones más prometedoras por su capacidad para ofrecer un bajo consumo de energía y largo alcance de comunicación lo que la hace ideal para una amplia variedad de aplicaciones, desde las ciudades inteligentes hasta la agricultura inteligente [15]. Este análisis literario proporciona una visión completa del estado actual de la tecnología LoRa donde se examina sus fundamentos, aplicaciones y futuro potencial.

LoRa es una tecnología de modulación de espectro ensanchado derivada de la modulación CSS (Chip Spread Spectrum). En comparación con otras tecnologías IoT, LoRa permite la transmisión de datos a largas distancias contando con una eficiencia energética notablemente alta [16]. En un estudio comparativo de tecnologías LPWAN, incluyendo LoRa, Sigfox y NB-IoT, Mekki et al. (2019) identificaron que aunque LoRa sobresale en términos de alcance y consumo de energía, presenta desafíos en cuanto a la latencia de transmisión y la capacidad de red en comparación con otras tecnologías como Zigbee y NB-IoT. Estas características hacen a LoRa ideal para aplicaciones donde el alcance y la eficiencia energética son cruciales, pero menos

favorable para aplicaciones que requieren alta capacidad de transmisión y baja latencia [17].

Según The Things Network, LoRa ha demostrado ser eficaz en aplicaciones diversas de IoT; en la agricultura inteligente, donde permite la monitorización remota de condiciones del clima y el suelo, optimizando así el uso de recursos y mejorando la productividad agrícola [18]. En las ciudades inteligentes, es usada para gestionar servicios públicos como el alumbrado y la recolección de residuos, ayudando a una gestión más eficiente y sostenible de los recursos urbanos [19].

2.2.1.1. LoRaWAN

LoRa ha estandarizado el protocolo MAC (Media Access Control) se lo denomina como LoRaWAN (Long Range Wide Area Network) mismo que se encarga de definir el protocolo de comunicación así como la arquitectura del sistema de red [20].

LoRaWAN garantiza velocidades de datos de 0,3 kbps hasta 50 kbps, los mismos que se consideran aceptables para el envío de datos de sensores usualmente usados en tiempo real en IoT [21]. Por otro lado si se requiere de una transmisión de datos con un gran ancho de banda como pueden ser en envío de imágenes y vídeos en tiempo real, LoRa puede no ser adecuado, el bajo consumo de energía en los dispositivos finales se debe a las bajas velocidades con las que trabaja LoRa permitiendo así extender el uso de este tipo de dispositivos que duraría años de uso continuo [22].

El crecimiento de LoRa en el IoT es innegable, impulsado por sus ventajas de alcance y eficiencia energética. Aunque los desafíos técnicos requieren soluciones innovadoras. A medida que la tecnología evoluciona se espera su adopción en nuevas aplicaciones de sectores en

crecimiento, consolidando así su papel en el ecosistema IoT [23, 24].

2.2.1.2. Arquitectura de LoRaWAN

La arquitectura de LoRaWAN se organiza en una estructura jerárquica que incluye tres componentes principales: servidores de red, gateways y dispositivos finales. Los dispositivos finales, que son nodos equipados con módulos de comunicación LoRa, recopilan y transmiten datos desde su entorno. Estos dispositivos finales no se comunican directamente con los servidores de red; en su lugar, envían sus datos a través de gateways, que actúan como intermediarios. Los gateways reciben las señales de radio LoRa de los dispositivos finales y luego retransmiten esos datos al servidor de red mediante una conexión de alta capacidad, como Ethernet o celular. El servidor de red gestiona la red LoRaWAN, coordinando la comunicación, asegurando la autenticación de los dispositivos y dirigiendo los datos a las aplicaciones finales. Esta estructura permite una comunicación eficiente y de largo alcance, adecuada para diversas aplicaciones de IoT, desde la monitorización ambiental hasta la gestión de ciudades inteligentes [25]. En la figura 2.1 se muestra la arquitectura LoRa.

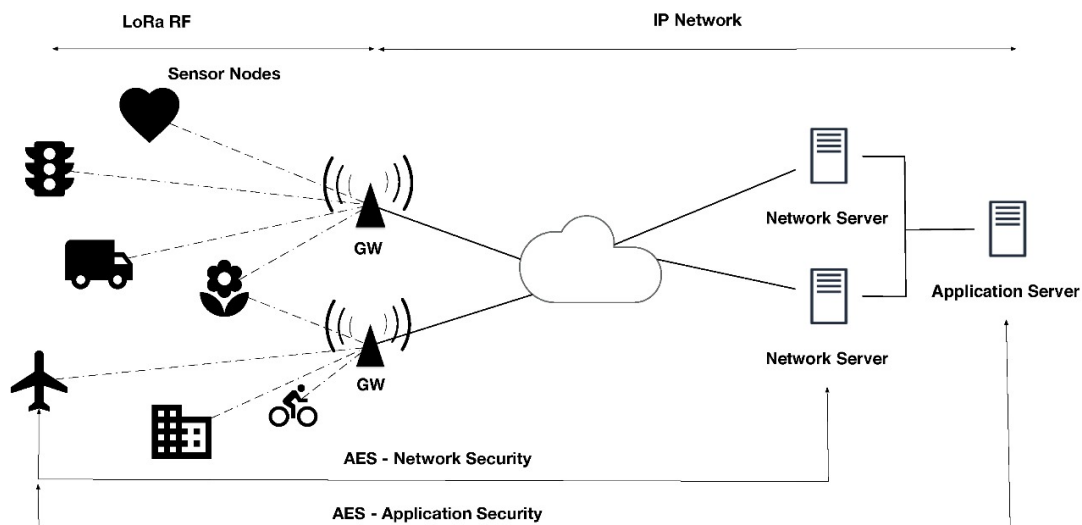


Figura 2.1: Arquitectura LoRa [1].

2.2.1.3. Desafíos y Limitaciones de LoRa y LoRaWAN

Por lo general la tecnología LoRa ofrece numerosas ventajas para aplicaciones IoT, sin embargo, presenta varios desafíos y limitaciones que deben ser tomados en cuenta para maximizar su efectividad y eficiencia en los diferentes entornos que se busca implementar.

- Interferencia y capacidad de Red:** Uno de los principales desafíos de LoRa es la interferencia, en especial en entornos urbanos densamente poblados. Pues LoRa opera en bandas de frecuencias no licenciadas como es el caso de la ISM (Industrial, Scientific, and Medical), otras tecnologías que utilizan las mismas bandas de frecuencia como Wi-Fi y Bluetooth pueden presentar interferencias con LoRa [26]. De esta manera la calidad de comunicación y la fiabilidad se puede reducir en la transmisión de datos.

Además, cuando se conecta un gran número de dispositivos la capacidad de la red LoRaWAN puede ser limitada. Debido a que la tecnología utiliza un mecanismo de acce-

so aleatorio denominado ALOHA, puede provocar colisiones en los paquetes cuando se transmiten datos de varios dispositivos simultáneamente [27]. Las aplicaciones de alta densidad de dispositivos se ven comprometidas.

- **Latencia y Tasa de Datos:** La tecnología esta diseñada para priorizar la eficiencia energética y el alcance de los datos por sobre la velocidad de transmisión. Como resultado la tasa de datos es relativamente baja, y la latencia puede ser alta, por lo cual se limita en la transmisión de grandes volúmenes de datos como por ejemplo el streaming de video [28].
- **Seguridad:** Los dispositivos IoT a menudo manejan datos sensibles, es por ello que en LoRaWAN la seguridad es crítica. A pesar de implementar varias medidas de seguridad tal como la autenticación de dispositivos y la encriptación de los datos, aun existen potenciales vulnerabilidades. Una de las amenazas mas comunes son los ataques de repetición donde un atacante intercepta los datos los cuales son retransmitidos [29]. La gestión de claves es compleja y si no se maneja adecuadamente puede ocasionar vulnerabilidades.
- **Limitaciones Energéticas:** LoRa es conocido por su bajo consumo energético, pero las limitaciones energéticas aún así representan un desafío, puesto que existen aplicaciones donde se requiere una transmisión de datos frecuente en las cuales se debe equilibrar la frecuencia de transmisión para garantizar así la vida útil del dispositivo final [30].
- **Regulaciones y Estándares:** Según la región en la que opere LoRa varían las bandas de frecuencia y las regulaciones pueden imponer restricciones adicionales tal es el caso de las frecuencias específicas a las que se pueden usar. Los dispositivos de esta manera necesitarían diferentes configuraciones de hardware y software para cumplir con las re-

gulaciones específicas de acuerdo a la región en la que se trabaje [31].

2.2.1.4. Aplicaciones de LoRa en IoT

LoRaWan se ha posicionado en el mercado en términos de aplicaciones y sus respectivas áreas de aplicación por sus propiedades únicas en el adecuado manejo de escenarios que requieren acceso restringido o limitado de la electricidad, lugares remotos o donde se requiera un acceso complejo, permite conexiones de dispositivos finales significativamente mayores, y no requiere de mensajes continuos pero si a largo plazo [32].

Las variables mas importantes en la agricultura inteligente como lo son el riego adecuado, la humedad del suelo, la temperatura pueden ser medidas y usadas por LoRa para la toma de decisiones en función de los datos a tiempo real para aumentar la eficacia y la eficiencia del cultivo [32].

A continuación se muestra un diagrama de los usos mas significativos de LoRa en IoT 2.2:



Figura 2.2: Aplicaciones LoRa [2].

Avances Recientes de LoRa

Mejoras en la Modulación y Codificación

Recientemente, se han desarrollado técnicas avanzadas de modulación y codificación para mejorar la eficiencia espectral y la resistencia a la interferencia de LoRa. Estas técnicas permiten una mayor robustez de la señal y una mejor capacidad para manejar ambientes de alta interferencia, lo que amplía la aplicabilidad de LoRa en entornos urbanos densos [33].

Integración con Redes 5G

La integración de LoRa con redes 5G ha sido un área de investigación activa. Esta integración permite combinar las capacidades de largo alcance y bajo consumo de energía de LoRa con la alta velocidad y baja latencia de 5G. Este enfoque híbrido puede facilitar aplicaciones IoT más complejas, que requieren tanto alta densidad de dispositivos como comunicación en tiempo real [19].

Nuevos Algoritmos de Gestión de Energía

Se han desarrollado nuevos algoritmos de gestión de energía para dispositivos LoRa, optimizando su consumo energético en función de la carga de trabajo y las condiciones ambientales. Estos algoritmos permiten extender la vida útil de los dispositivos alimentados por batería, lo que es crucial para aplicaciones IoT en áreas remotas o de difícil acceso [34].

Implementación de Inteligencia Artificial y Aprendizaje Automático

La integración de inteligencia artificial (IA) y aprendizaje automático en redes LoRa ha abierto nuevas posibilidades para la gestión y el análisis de datos en tiempo real. Estas tecnologías permiten predecir fallas, optimizar el uso de recursos y mejorar la toma de decisiones basada en datos históricos y en tiempo real [35].

Seguridad Mejorada

Las investigaciones recientes han enfocado sus esfuerzos en mejorar la seguridad de las redes LoRaWAN. Se han propuesto nuevos protocolos de seguridad y mecanismos de cifrado avanzados para proteger los datos transmitidos y garantizar la autenticación segura de los dispositivos [36]. Estos avances son esenciales para aplicaciones críticas, como la atención médica y la infraestructura crítica.

Expansión del Ecosistema de Dispositivos y Aplicaciones

El ecosistema de dispositivos y aplicaciones que utilizan LoRa se ha expandido significativamente. Nuevos sensores, actuadores y plataformas de desarrollo han sido lanzados al mercado, facilitando la creación de soluciones personalizadas para diversos sectores, como la agricultura de precisión, la gestión de recursos hídricos y las ciudades inteligentes [37].

2.2.2. The Things Network

The Things Network es una red pública destinada principalmente a simplificar el uso de tecnologías como LoRaWAN, la cual tiene una gran cobertura que dependiendo del dispositivo seleccionado alcanza hasta 15 km en zonas urbanas lo que permite la correcta implementación de soluciones IoT [38].

El uso de The Things Network (TTN) en la agricultura urbana inteligente es un avance significativo, ya que permite la conexión de varios sensores y dispositivos para recopilar datos para monitorear y analizar las variables climatológicas en la agricultura urbana. La integración de tecnologías de la IoT como la TTN permite la captura, el almacenamiento, el análisis y la visualización eficientes de los datos relacionados con las condiciones ambientales en las

zonas urbanas, lo que contribuye a los procesos de toma de decisiones para la gestión de los cultivos [39]. Además, la aplicación de la IoT en la agricultura, particularmente en entornos urbanos, contribuye a la optimización de los recursos, la mejora de la eficiencia y las prácticas sostenibles, alineándose con los objetivos de las ciudades inteligentes para mejorar la seguridad alimentaria y la sostenibilidad [40]. Los agricultores urbanos pueden mejorar la productividad, minimizar los riesgos y tomar decisiones informadas utilizando soluciones de IoT como TTN, lo que promueve los sistemas agrícolas urbanos inteligentes.

Webhooks

Las integraciones con otras plataformas IoT son importantes en los proyectos por que permiten acceder a las herramientas necesarias para el procesamiento de los datos. The Things Network ha implementado una solución para este hecho el cual lo denominó Webhook que es una herramienta que permite realizar un puente seguro para el paso de los datos hacia otros servidores u otras plataformas IoT [41].

2.2.3. Dragino

Dragino se especializa en el desarrollo y manufactura de soluciones IoT, ofreciendo tecnología ideal para proyectos que requieren integración avanzada de hardware y software, así como protocolos de comunicación. Este enfoque facilita la implementación eficaz de soluciones IoT adaptadas a necesidades específicas, la tecnología Dragino es fundamental en el desarrollo de cultivos inteligentes, ya que proporciona herramientas clave para la implementación eficiente de soluciones IoT en la agricultura. Estos dispositivos facilitan la recopilación y análisis de datos cruciales del entorno agrícola, como humedad del suelo, temperatura y niveles de nutrien-

tes, permitiendo a los agricultores optimizar el riego y la fertilización. Además, la capacidad de Dragino para integrarse con diversos protocolos de comunicación asegura una coordinación efectiva entre los diferentes dispositivos IoT en el campo, lo que mejora significativamente la gestión y supervisión de los cultivos en tiempo real [42].

2.2.4. Sistema Operativo GNU Linux

Una de las características más destacadas de Linux es su capacidad de adaptación. A diferencia de otros sistemas operativos comerciales, Linux permite una personalización extensa para cumplir con los requisitos específicos de cualquier dispositivo o aplicación. Esto es crucial para la tecnología IoT, donde las configuraciones a medida son frecuentes. Gracias a su estructura modular, los desarrolladores pueden elegir e integrar únicamente los componentes esenciales, simplificando y aligerando el sistema operativo. Además, el diseño eficiente de Linux facilita su operación en dispositivos de baja potencia, comúnmente empleados en el ámbito IoT. Otro punto fuerte de Linux en el contexto de IoT es su robustez en seguridad. Dado que los dispositivos IoT están expuestos a amenazas cibernéticas por su conexión constante a Internet, la seguridad es primordial. Linux ofrece características de protección probadas y, al ser de código abierto, las vulnerabilidades se detectan y remedian con rapidez, reforzando la defensa contra ataques [43].

2.2.5. Google Cloud

Para el estudio realizado por Kundu en [44] donde utiliza Google Cloud como herramienta para el almacenamiento y posterior procesamiento de los datos donde comprueba que este

sistema demuestra la potencialidad de Google Cloud en la agricultura inteligente, ofreciendo soluciones que mejoran la precisión en la gestión del cultivo y aumentan la eficiencia operativa. La infraestructura de Google Cloud proporciona una plataforma robusta para el almacenamiento y análisis de grandes volúmenes de datos agrícolas, facilitando una agricultura más informada y sostenible. Esta integración de tecnologías IoT con la capacidad de computación en la nube de Google permite a los agricultores supervisar y ajustar prácticas agrícolas con gran precisión, promoviendo un uso más eficiente de los recursos y mejorando la productividad agrícola.

2.2.6. Apache

Apache y Nginx son dos de los servidores de código abierto más utilizados, conocidos por su facilidad de manejo y robustez, lo que los hace ideales para operaciones complejas. Su popularidad asegura beneficios como mantenimiento constante y configuración sencilla de sitios. Además, la seguridad que ofrecen los convierte en una opción preferente para una amplia variedad de aplicaciones. Una de sus características más valoradas es la flexibilidad, ya que soportan los lenguajes de programación más populares, incluido PHP, lo que amplía su aplicabilidad en diferentes contextos[45]. Lo convierte en el candidato ideal para el desarrollo en su entorno del enlace integral entre The Things Network y la base de datos

2.2.7. MySQL

En el contexto de IoT, MySQL ofrece una serie de beneficios significativos para la gestión de datos de sensores. Su capacidad para manejar estructuras de datos estables y su eficacia en la gestión de relaciones entre datos hacen de MySQL una opción ideal para aplicaciones que

requieren integridad referencial. Además, su escalabilidad y robustez en la seguridad, junto con capacidades avanzadas de replicación y distribución, aseguran la disponibilidad y consistencia de los datos bajo demanda elevada o en caso de fallos. El soporte para SQL estándar de MySQL también facilita la integración con diversas herramientas y plataformas, permitiendo a los desarrolladores utilizar recursos y habilidades existentes para optimizar el desarrollo y mantenimiento de soluciones IoT [46].

2.2.8. Firebase

Firestore se destaca como una solución ideal para aplicaciones IoT, ofreciendo escalabilidad automática que se adapta sin intervención manual, es esencial para manejar fluctuaciones en el volumen de datos. Su Realtime Database permite la sincronización instantánea de datos a todos los clientes, facilitando el monitoreo en tiempo real. La integración de Firebase Authentication proporciona un proceso seguro y eficiente para la autenticación y autorización de usuarios y dispositivos. Además, ofrece resistencia offline, manteniendo los datos accesibles incluso sin conexión y sincronizándolos automáticamente al restablecerse Internet. La plataforma asegura la protección de datos mediante reglas de seguridad configurables que controlan el acceso y las operaciones de datos, y se integra perfectamente con herramientas de Google como Google Analytics para optimizar el rendimiento basado en análisis detallados del comportamiento del usuario. Estas características hacen de Firestore una opción robusta y escalable para desarrolladores que buscan implementar soluciones avanzadas de IoT [47].

2.2.9. Python

Python es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones de IoT debido a su simplicidad, versatilidad y robustez. En este proyecto, Python ha sido elegido por sus capacidades avanzadas para la manipulación y análisis de datos, así como por su amplia gama de bibliotecas y herramientas que facilitan la implementación de soluciones IoT .

Una de las principales ventajas de Python es su compatibilidad con diversas bibliotecas que simplifican la comunicación entre diferentes componentes del sistema. Por ejemplo, se ha utilizado `firebase-admin` para la interacción con Firebase, lo que permite una integración fluida y eficiente de la base de datos en tiempo real con la infraestructura de la red IoT . Además, bibliotecas como `matplotlib` y `numpy` han sido fundamentales para la visualización y el procesamiento de datos, respectivamente, lo que permite una gestión más efectiva de la información recolectada por los sensores .

La elección de Python también se debe a su facilidad para realizar pruebas y depuración de código, lo cual es esencial en el desarrollo de sistemas complejos como las redes IoT. La capacidad de Python para ejecutar scripts y realizar simulaciones en entornos controlados permite identificar y solucionar problemas de manera rápida y efectiva .

Además, la comunidad activa de Python y la amplia disponibilidad de recursos y documentación proporcionan un soporte significativo durante el desarrollo del proyecto . Esto asegura que cualquier problema o desafío técnico pueda ser abordado con rapidez, aprovechando las contribuciones de una vasta red de desarrolladores y expertos en el campo .

2.3. Conclusión del Estado del Arte

La revisión del estado del arte ha demostrado que la tecnología LoRa es una solución robusta y adaptable para IoT, particularmente efectiva en el campo de la agricultura inteligente. Se ha destacado su capacidad para proporcionar comunicación de largo alcance y bajo consumo energético, lo cual es esencial para el monitoreo eficiente de cultivos urbanos. A pesar de sus numerosas ventajas, se identificaron limitaciones técnicas como la capacidad limitada de datos y la baja velocidad de transmisión, que podrían restringir su uso en aplicaciones que requieren transmisión de grandes volúmenes de datos en tiempo real. Sin embargo, globalmente, la tecnología LoRa se está adaptando rápidamente para diversas aplicaciones IoT. Por otro lado, en Ibarra, Ecuador, la adopción ha sido limitada, posiblemente debido a la falta de infraestructura tecnológica adecuada y a barreras de conocimiento y económicas. Este trabajo se propone superar las barreras identificadas mediante el diseño e implementación de una red IoT comunitaria que utilice la tecnología LoRa, adaptada específicamente para las condiciones y necesidades de Ibarra.

El desarrollo de la red IoT se basará en la integración de LoRaWAN y The Things Network, proporcionando una solución de bajo costo y fácil implementación. La metodología incluirá una fase de diseño detallado del sistema, seleccionando componentes que optimicen el consumo energético y la cobertura de la red. La arquitectura de la red consistirá en nodos sensores LoRa distribuidos estratégicamente en áreas agrícolas, un gateway LoRaWAN centralizado que recopile los datos de los sensores, y una integración con Google Cloud para el almacenamiento y análisis de los datos recogidos. Se implementará un protocolo de pruebas para verificar la

funcionalidad, el alcance y la confiabilidad de la comunicación LoRa en el entorno específico de Ibarra. Esto incluirá pruebas de campo para evaluar la eficacia del sistema en diferentes condiciones meteorológicas y topográficas. Tras la validación, la red será implementada a escala completa. Se establecerá un sistema de monitoreo continuo para evaluar el rendimiento de la red, identificar y resolver problemas operativos, y optimizar la configuración del sistema según sea necesario.

Capítulo III

Desarrollo

Este capítulo se enfoca en desarrollar una puerta de enlace avanzada, diseñada para visualizar de manera precisa los datos de nodos finales en cultivos urbanos, esencial para un monitoreo eficaz del entorno agrícola. Se seleccionarán y analizarán las variables críticas de cultivo, fundamentales para el seguimiento y la gestión efectiva. También se detallará el proceso de interconexión entre el hardware y el software de la puerta de enlace, explicando cómo estas conexiones permiten una integración fluida y eficiente, crucial para el sistema de monitoreo. Las configuraciones óptimas para cada componente se describirán para asegurar un funcionamiento eficiente. Este análisis profundo busca implementar una solución tecnológica que no solo sea efectiva sino que también optimice el rendimiento del sistema para mejorar la sustentabilidad y productividad de los cultivos urbanos.

3.1. Arquitectura

En la Figura 3.1 se muestra la arquitectura de la red propuesta a partir del estado del arte con las herramientas mas idóneas para la aplicación que se busca obtener.

La siguiente sección detalla las características de los dispositivos y las instancias requeridas para la implementación de la red. Además, se describen los elementos configurados para garantizar un funcionamiento eficiente y coherente del sistema.

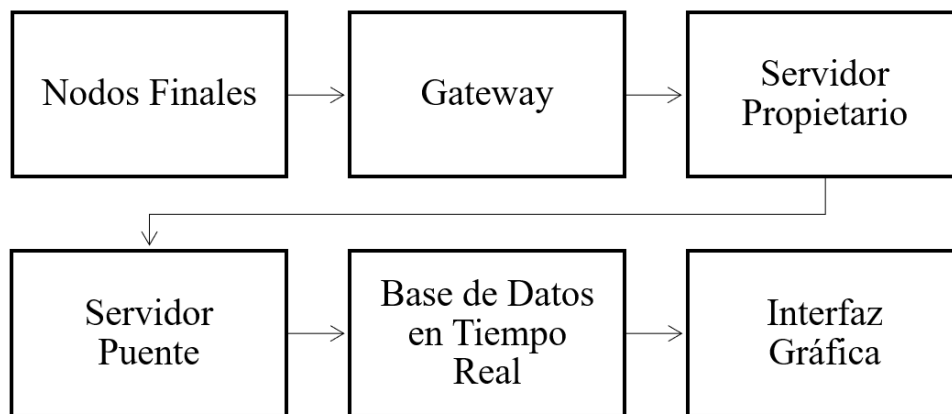


Figura 3.1: Arquitectura de la red.

3.1.1. Nodo Final

El sensor LNS-50 de Dragino incorpora la tecnología LoRa como protocolo de comunicación, diseñado específicamente para facilitar el uso de nodos finales de sensores. Este dispositivo ofrece diversas entradas y salidas que permiten la conexión de múltiples tipos de sensores, adaptándose a una amplia gama de aplicaciones. Especialmente relevante para la agricultura inteligente, el LNS-50 está optimizado para integrar sensores específicos que mejoran la gestión y el monitoreo de cultivos [48]. Para este trabajo se implementan 2 Nodos Finales con configuraciones similares.

3.1.1.1. Configuración

Los dispositivos finales se configuran con los aspectos básicos que se presentan en la Tabla 3.1 para el primer dispositivo y en la Tabla 3.2 para el segundo.

Campo	Configuración
ID del dispositivo	eui-a84041239358f65b
Plan de frecuencia	EU868 / US915 / AU915
Versión LoRaWAN	1.0.3
Parámetros regionales	Revisión A
Tiempo de transmisión de los datos	90 segundos
Modo de operación	Modo 8

Tabla 3.1: Configuraciones iniciales para el nodo LSN50

Campo	Configuración
ID del dispositivo	eui-a84041521187bcaf
Plan de frecuencia	EU868 / US915 / AU915
Versión LoRaWAN	1.0.3
Parámetros regionales	Revisión A
Tiempo de transmisión de los datos	90 segundos
Modo de operación	Modo 8

Tabla 3.2: Configuraciones iniciales para el nodo LSN50-1

Junto con los nodos finales se implementan los sensores encargados de medir los datos de las variables del cultivo, tales son la temperatura del suelo, la humedad del suelo y la humedad del aire mostrados con más detalle en la Tabla 3.3.

Sensor	Uso
DS18B20	Medición de temperatura del suelo [49].
DHT11	Medición de temperatura y humedad del aire [50].
FC-28	Medición de la humedad del suelo [51].

Tabla 3.3: Uso de los sensores en la red IoT

3.1.1.2. Modo de Operación

El dispositivo nodo final ofrece diversos modos de operación que se adaptan a variadas aplicaciones, dependiendo de la configuración seleccionada [48]. Específicamente para el monitoreo de cultivos, el modo 3 ADC+1DS18B20 resulta ser el más completo. Según el manual de usuario, este modo facilita tres entradas analógico-digitales que permiten conectar diferentes tipos de sensores. Estas entradas, a través de transformaciones lineales, permiten la captura y el análisis de una amplia gama de variables medidas.

Por consiguiente, los sensores están conectados a los canales del conversor análogo-digital, facilitando así la transferencia de los datos. Dichos datos se transmiten en forma de niveles de voltaje generados por los sensores.

3.1.2. Gateway

La configuración del Gateway es esencial para su operación adecuada. El acceso inicial a la configuración del dispositivo se realiza a través de una dirección IP específica para cada unidad. Este acceso requiere credenciales que varían según el modelo específico del dispositivo Dragino. La Figura 3.2 muestra la ventana principal de configuración.



Figura 3.2: Ventana principal del Gateway.

3.1.2.1. Configuración

Los parámetros para configurar el Gateway se seleccionan según la zona geográfica y los dispositivos finales disponibles, asegurando la correcta transmisión de datos mediante un ancho de banda adecuado para la región donde opera la red. La Figura 3.3 ilustra los valores recomendados para un funcionamiento óptimo. En este contexto, el ancho de banda del Gateway se ajusta de acuerdo con las especificaciones del dispositivo; el manual del Dragino LG308N indica que las frecuencias empleadas son apropiadas para nuestra zona [52].

Figura 3.3: Configuración del Gateway.

Para la conexión con el servidor externo, se debe utilizar The Things Network. Se recomienda usar los puertos predeterminados para minimizar la necesidad de configuraciones adicionales como lo muestra la Figura 3.4.

Figura 3.4: Configuración LoRaWAN.

El dispositivo debe configurarse utilizando las identificaciones específicas provistas por Dragino, las cuales se encuentran directamente en el mismo dispositivo. Para el caso específico del

dispositivo adquirido son las credenciales que se muestran en la Figura 3.5

LoRaWAN -- Basic Station

General Settings

Email

Gateway ID Restore ?

Primary LoRaWAN Server

Service Provider

LNS URI

TLS CA Certificate tc.trust Ningún archivo seleccionado

TLS Certificate tc.crt Ningún archivo seleccionado

TLS Key tc.key Ningún archivo seleccionado

Current Mode: **LoRaWAN Semtech UDP** Click Save & Apply will change to mode :**LoRaWAN Basic Station**

Figura 3.5: Identificaciones del Gateway.

3.1.3. Conexión con el Servidor Propietario

Igualmente, el análisis literario reconoce a The Things Network como un servidor propietario esencial para transmitir las variables desde el Gateway hasta un servidor puente posterior. A continuación se muestran las pestañas en las que se agregan los dispositivos para su autenticación y manejo.

3.1.3.1. Aplicaciones

En la sección de aplicaciones del servidor propietario, se integran los Nodos Finales previamente configurados. Se detalla la información de autenticación y la ubicación específica del Gateway en la Tabla 3.4, asegurando así la recepción de las variables en el servidor.

Campo	Configuración
End device ID	eui-a84041239358f65b
Frequency plan	United States 902-928 MHz, FSB 2 (used by TTN)
LoRaWAN version	LoRaWAN Specification 1.0.3
Regional Parameters version	RP001 Regional Parameters 1.0.3 revision A
Created at	Jun 21, 2024 20:58:07

Tabla 3.4: Información general del dispositivo

Para el segundo nodo final, se requieren configuraciones similares, variando únicamente los detalles de autenticación del dispositivo, como se detalla en la Tabla 3.5.

Campo	Configuración
End device ID	eui-a84041521187bcaf
Frequency plan	United States 902-928 MHz, FSB 2 (used by TTN)
LoRaWAN version	LoRaWAN Specification 1.0.3
Regional Parameters version	RP001 Regional Parameters 1.0.3 revision A
Created at	Jun 21, 2024 20:58:07

Tabla 3.5: Información general del segundo dispositivo

3.1.3.2. Gateway

Las principales configuraciones del Gateway se representan en la Tabla 3.6.

Campo	Configuración
LoRaWAN Public	True
Clock Source	1
Antenna Gain	0 dB
Radio 0 Frequency	904.3 MHz
Radio 1 Frequency	905 MHz
RSSI Offset	-166 dBm
TX Enable	True for Radio 0
TX Min Frequency	923 MHz
TX Max Frequency	928 MHz

Tabla 3.6: Configuraciones iniciales del nodo LoRaWAN

3.1.3.3. Webhook

En este apartado se especifica la ruta hacia donde se redireccionan los datos. En este caso es hacia la base de datos MySQL la cual se aloja en un servidor público. Tal como lo muestra la Tabla 3.7.

Configuración	Valor
Webhook ID	Ins50
Webhook format	JSON
Base URL	<code>http://34.172.196.246/insert_dht.php</code>
Downlink API key	-
Request authentication	Autenticación básica
Additional headers	-
Filter event data	-
Enabled event types	-

Tabla 3.7: Configuración del Webhook

3.1.4. Servidor Puente

Mediante la API Compute Engine de Google Cloud, es posible configurar una máquina virtual adaptada a las necesidades específicas del proyecto. Para el monitoreo actual, se recomiendan las configuraciones mostradas en la Tabla 3.8 para la instancia virtual creada denominada *loratest-1*.

Información Básica	
Nombre	loratest-1
ID de instancia	5621140461498169338
Descripción	Ninguna
Tipo	Instancia
Estado	Activa
Hora de creación	jun 30, 2024, 4:07:34 p.m. UTC-05:00
Zona	us-central1-b
Plantilla de instancia	Ninguna
En uso por	Ninguno
Reservas	Elegir automáticamente
Etiquetas	Ninguna
Protección contra la eliminación	Inhabilitado
Servicio Confidential VM	Inhabilitado
Tamaño de estado preservado	0 GB

Tabla 3.8: Información básica de la instancia loratest-1.

La elección de la imagen de máquina es crucial, pues define el entorno del sistema operativo que facilitará las tareas específicas de recepción y transmisión de datos. En el análisis literario realizado, se destaca la selección de Ubuntu, un sistema operativo basado en GNU/Linux, por su adecuación a las necesidades del proyecto, las especificaciones se muestran en la Tabla 3.9.

Configuración de la máquina	
Tipo de máquina	e2-medium
Plataforma de CPU	Intel Broadwell
Plataforma de CPU mínima	Ninguna
Arquitectura	x86-64
CPU virtuales para proporción de núcleos	—
Núcleos visibles personalizados	—
Dispositivo de visualización	Inhabilitado
GPU	Ninguna
Políticas de recursos	—

Tabla 3.9: Configuración detallada de la máquina.

Las reglas de firewall son importantes para el desarrollo del proyecto pues mediante ellas

se acceden a los puertos necesarios para el manejo de tráfico web, es decir, habilitar el tráfico HTTP y HTTPS es esencial para el funcionamiento del proyecto. Su configuración se muestra en la Tabla 3.10.

Firewalls	
Tráfico HTTP	Activo
Tráfico HTTPS	Activo
Permitir las verificaciones de estado del balanceador de cargas	Inactivo

Tabla 3.10: Configuración de firewalls para tráfico web.

3.1.4.1. Configuración de Ubuntu 20.04

El sistema operativo elegido es Ubuntu 20.04 por ser un sistema de uso libre además del poco uso de recursos innecesarios para la aplicación de IoT que se requiere implementar. Las configuraciones del sistema se muestran en la Tabla 3.11.

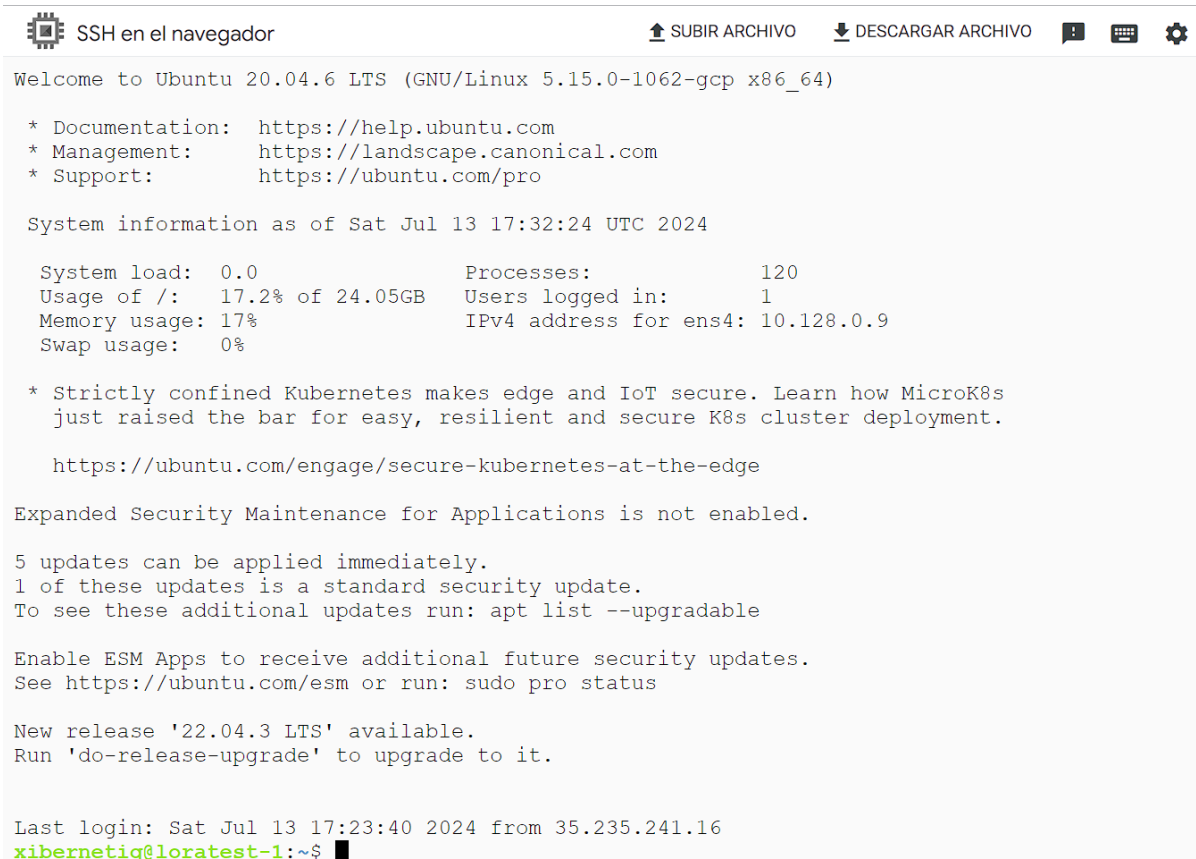
Campo	Configuración
Nombre	loratest-1
Imagen	ubuntu-20.04
Tipo de interfaz	SCSI
Tamaño (GB)	25
Nombre del dispositivo	loratest-1
Tipo	Disco SSD persistente
Arquitectura	x86-64

Tabla 3.11: Configuración del disco de arranque

La instalación del sistema puede verificarse accediendo a la máquina virtual mediante SSH, que viene preconfigurado por defecto en Google Cloud, facilitando el acceso sin necesidad de configuraciones adicionales. Durante la configuración inicial de la máquina virtual, se recomienda mantener activo el acceso a través de este puerto. Sin embargo, para garantizar la

seguridad de la red, se aconseja desactivar el puerto tras completar las configuraciones, a fin de prevenir posibles ataques externos.

La pantalla que se debe mostrar en la consola SSH se evidencia en la Figura 3.6.



```
SSH en el navegador SUBIR ARCHIVO DESCARGAR ARCHIVO
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1062-gcp x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Sat Jul 13 17:32:24 UTC 2024

System load:  0.0                Processes:    120
Usage of /:   17.2% of 24.05GB   Users logged in:  1
Memory usage: 17%                IPv4 address for ens4: 10.128.0.9
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Jul 13 17:23:40 2024 from 35.235.241.16
xibernetiq@loratest-1:~$
```

Figura 3.6: Salida de la consola SSH.

3.1.4.2. Instalación y Configuración de Apache

A través de la misma consola se instala en el sistema operativo el servidor Apache y se verifica su instalación dando como salida la Figura 3.7.

```
SSH en el navegador ↑ SUBIR ARCHIVO ↓ DESCARGAR ARCHIVO ! 📄 ⚙️  
xibernetiq@loratest-1:~$ sudo systemctl status apache2  
● apache2.service - The Apache HTTP Server  
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
  Active: active (running) since Sat 2024-07-13 06:21:41 UTC; 14h ago  
    Docs: https://httpd.apache.org/docs/2.4/  
  Main PID: 130255 (apache2)  
    Tasks: 9 (limit: 4680)  
  Memory: 15.6M  
  CGroup: /system.slice/apache2.service  
          └─130255 /usr/sbin/apache2 -k start  
            └─130257 /usr/sbin/apache2 -k start  
              └─130259 /usr/sbin/apache2 -k start  
                └─130260 /usr/sbin/apache2 -k start  
                  └─130261 /usr/sbin/apache2 -k start  
                    └─130262 /usr/sbin/apache2 -k start  
                      └─130263 /usr/sbin/apache2 -k start  
                        └─131572 /usr/sbin/apache2 -k start  
                          └─133477 /usr/sbin/apache2 -k start  
  
Jul 13 06:21:41 loratest-1 systemd[1]: apache2.service: Succeeded.  
Jul 13 06:21:41 loratest-1 systemd[1]: Stopped The Apache HTTP Server.  
Jul 13 06:21:41 loratest-1 systemd[1]: Starting The Apache HTTP Server...  
Jul 13 06:21:41 loratest-1 systemd[1]: Started The Apache HTTP Server.
```

Figura 3.7: Estado del servicio Apache2.

3.1.4.3. Instalación y Configuración de PHP 8.0


Las versiones de PHP desde la 8.0 en adelante necesitan un repositorio específico para su instalación, por lo que se deberá agregar el repositorio *Ondrej* al cual debe tener acceso Ubuntu para la instalación correcta de la versión de PHP [53]. Se utiliza las versiones más actualizadas de PHP para evitar así conflictos con las librerías también actualizadas de las bases de datos las cuales requieren de requisitos específicos.

A pesar de que los módulos más usuales para la mayor parte de aplicaciones se instalan por defecto, además, se deben incluir en la instalación los módulos adicionales para la aplicación que se desarrolla:

- php8.0-mysql
- php8.0-json

Para asegurar una comunicación efectiva entre los servicios de Apache y PHP, es esencial configurarlos con PHP-FPM, un gestor de procesos FastCGI que facilita la interacción entre aplicaciones y el servidor web. Esta configuración permite que Apache maneje aplicaciones escritas en PHP de manera eficiente [53].

De igual manera se verifica la correcta instalación del servicio y su salida se muestra en la Figura 3.8.



```
SSH en el navegador SUBIR ARCHIVO DESCARGAR ARCHIVO ! 🗄 ⚙️
xibernetiq@loratest-1:~$ sudo systemctl status php8.0-fpm
● php8.0-fpm.service - The PHP 8.0 FastCGI Process Manager
   Loaded: loaded (/lib/systemd/system/php8.0-fpm.service; enabled; vendor preset: enable)
   Active: active (running) since Sat 2024-07-06 05:25:24 UTC; 1 weeks 0 days ago
     Docs: man:php-fpm8.0(8)
  Main PID: 585 (php-fpm8.0)
   Status: "Processes active: 0, idle: 3, Requests: 1920, slow: 0, Traffic: 0req/sec"
    Tasks: 4 (limit: 4680)
   Memory: 20.7M
   CGroup: /system.slice/php8.0-fpm.service
           └─585 php-fpm: master process (/etc/php/8.0/fpm/php-fpm.conf)
             └─855 php-fpm: pool www
               └─856 php-fpm: pool www
                 └─857 php-fpm: pool www

Jul 06 05:25:16 loratest-1 systemd[1]: Starting The PHP 8.0 FastCGI Process Manager...
Jul 06 05:25:24 loratest-1 systemd[1]: Started The PHP 8.0 FastCGI Process Manager.
lines 1-16/16 (END)
```

Figura 3.8: Estado del servicio PHP 8.0-FPM

Una vez configurado el servidor Apache para que admita el lenguaje de programación PHP se escribe el documento encargado de autenticar el inicio de sesión en la base de datos MySQL el mismo se muestra en la Figura 3.9 será alojado en la dirección `/var/www/html/` la cual posee los permisos necesarios para el servicio de hosting. Se detalla el código fuente del archivo en el A.

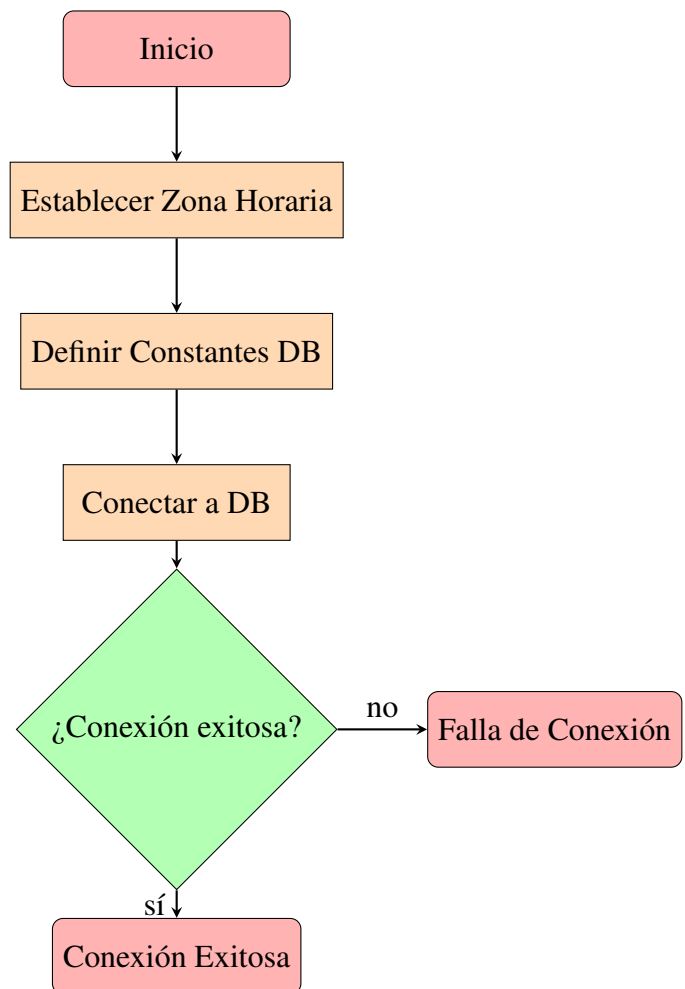


Figura 3.9: Diagrama de flujo de la configuración para la conexión a la base de datos MySQL

The Things Network envía datos utilizando el formato JSON, un método estándar para la transmisión de información estructurada a través de la Internet. Este formato es manejado por un script dedicado, denominado *insert dht*, cuya función es decodificar estos datos y facilitar su integración en la base de datos designada para este propósito. Este proceso incluye la interpretación de los datos JSON para extraer la información relevante y luego proceder a su almacenamiento adecuado y seguro en la base de datos. El flujo detallado de las operaciones llevadas a cabo por el script PHP se detalla en el diagrama de flujo presentado en la Figura 3.10,

el cual proporciona una representación visual de cada paso involucrado en este procedimiento.

De igual forma el código fuente que logra este flujo es el Anexo B.

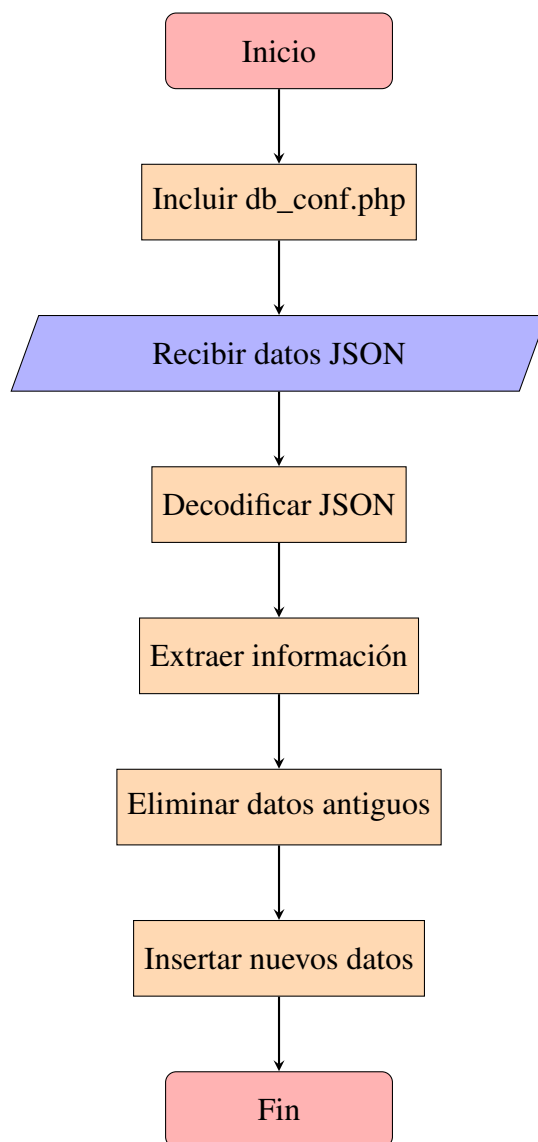


Figura 3.10: Diagrama de flujo del proceso de recepción y almacenamiento de datos en MySQL

3.1.4.4. Instalación y Configuración de MySQL

MySQL, ampliamente utilizado en entornos PHP debido a sus variadas funcionalidades, requiere para su instalación un entorno preconfigurado con un servidor activo y políticas de

firewall adecuadas, incluyendo el acceso HTTP y HTTPS. Este acceso se facilita mediante la activación de UFW (Uncomplicated Firewall) en Ubuntu, que por defecto está deshabilitado. Por tanto, es necesario habilitar UFW antes de proceder con la instalación de MySQL [54]. La Figura 3.11 muestra la política a implementar.

```

SSH en el navegador
SUBIR ARCHIVO  DESCARGAR ARCHIVO
xibernetiq@loratest-1:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
80 ALLOW Anywhere
443 ALLOW Anywhere
22 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
22 (v6) ALLOW Anywhere (v6)

```

Figura 3.11: Política de Firewall

Una vez cumplidos los requisitos previamente mencionados, se instala la base de datos en el sistema Ubuntu utilizando la línea de comandos. Además, se realiza una configuración inicial ajustada a las necesidades del proyecto, según las especificaciones detalladas en la Tabla 3.12. Durante el proceso de configuración, se recomienda seleccionar opciones afirmativas en las preguntas adicionales; esto facilita la eliminación de usuarios anónimos y la base de datos de prueba, asegurando un entorno más seguro y adecuado para el proyecto [54].

Campo	Configuración
Validate password plugin	YES
Password validation policy	STRONG
Estimated strength of the password: 100	YES

Tabla 3.12: Configuraciones de seguridad de contraseña en MySQL

Posteriormente, se crea una nueva base de datos y una tabla con todos los campos necesarios

donde se almacenan los datos enviados desde los nodos finales LoRa, la base de datos tiene como nombre *lms50*. La creación de la tabla se detalla en el código del Anexo G.

Luego, se crea un nuevo usuario que tenga todos los privilegios sobre la base de datos y sobre la tabla antes creada el cual se usa en la autenticación que realiza el documento PHP tal como muestra el Anexo H.

3.1.5. Base de Datos en Tiempo Real

Es imprescindible transferir los datos a una base de datos en tiempo real como Firebase para este proyecto, lo que facilitará la visualización de los mismos, un componente crucial para el monitoreo eficaz. Para el procesamiento y publicación de estos datos en Firebase, se emplearán scripts puente desarrollados en Python. Este lenguaje de programación es ampliamente reconocido y utilizado en la comunidad científica por su eficacia y flexibilidad.

La configuración de la base de datos en tiempo real de Firebase se realiza mediante la creación de un nodo específico destinado a almacenar las variables clave para el monitoreo. Este nodo se actualiza automáticamente con cada nueva transmisión de datos por parte de los dispositivos finales. La figura 3.12 ilustra esta configuración detalladamente.

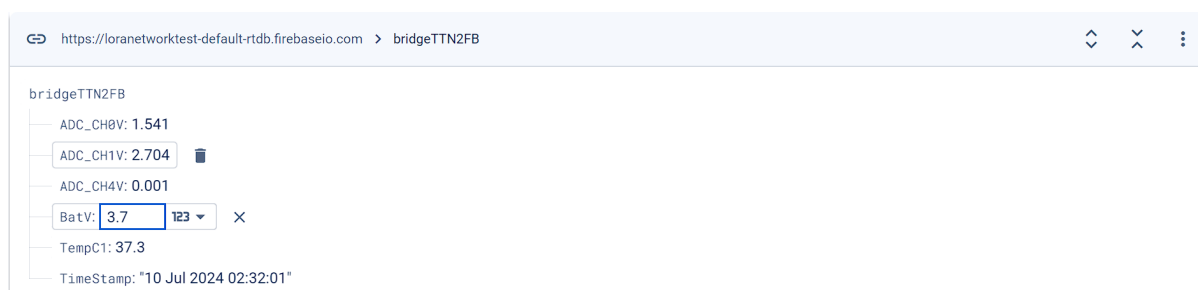


Figura 3.12: Estructura de las variables en Firebase

Por motivos de seguridad, el acceso a los datos almacenados en Firebase desde fuera está restringido. Para permitir actualizaciones seguras y eficientes, es necesario configurar permisos específicos que faciliten la lectura y escritura de las variables. Estos permisos se establecen mediante reglas detalladas en la configuración de la base de datos en tiempo real, como se demuestra en la Figura 3.13.

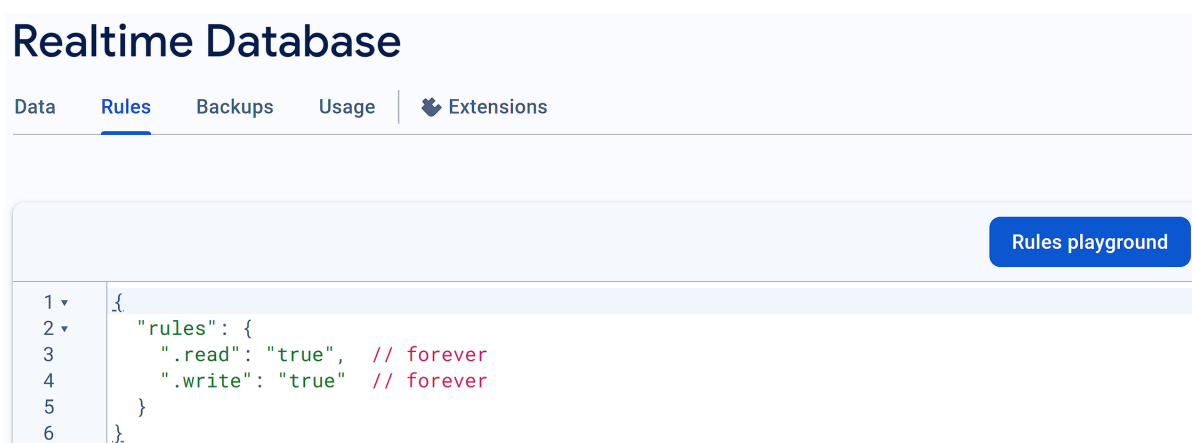


Figura 3.13: Reglas para lectura y escritura de variables en Firebase

Estas configuraciones facilitan el acceso a los archivos de procesamiento de datos, los cuales se ejecutan en tiempo real. El archivo más crítico se aloja en la máquina virtual. Como se mencionó anteriormente, el uso de Python, junto con sus bibliotecas, permite acceder a la base de datos especificando el DNS proporcionado por Firebase.

Este documento desempeña el rol de puente entre las bases de datos empleadas. Accede a MySQL para extraer datos actualizados provenientes de los sensores y elimina los registros superfluos, asegurando que solo permanezcan los datos más recientes. Posteriormente, almacena estos registros y los transfiere a Firebase, donde los datos se vuelven más accesibles para su posterior procesamiento tal como lo muestra la Figura 3.14.

Los canales del conversor análogo-digital integrados en los dispositivos finales de Dragino



Figura 3.14: Flujo de datos desde MySQL a Firebase mediante un script de Python

tienen la capacidad de transmitir voltajes a través de los canales de posteo de datos, presentando las variables en este formato específico. Para realizar un análisis adecuado y tomar decisiones informadas en el monitoreo de los cultivos, es esencial procesar estos datos correctamente. Los detalles del código utilizado para este procesamiento se encuentran detallados en el Anexo C.

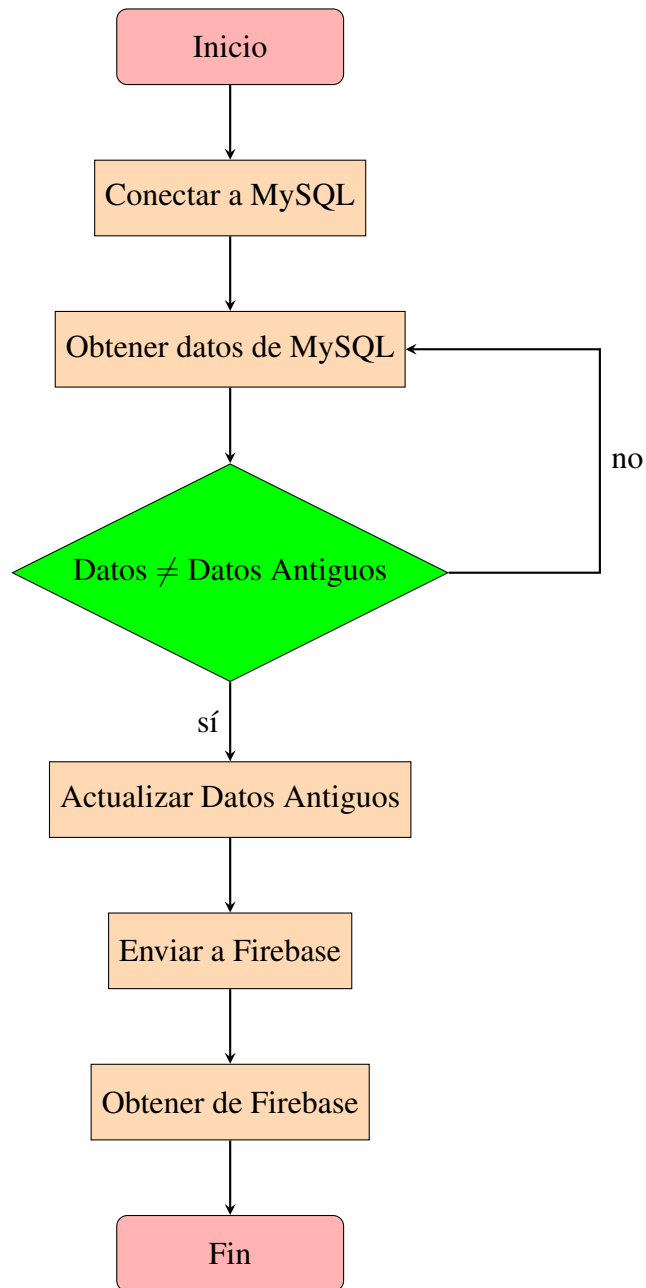


Figura 3.15: Diagrama de flujo del proceso de sincronización de datos con MySQL y Firebase

3.1.6. Interfaz Gráfica

Python, un lenguaje de programación versátil, facilita la visualización de datos en tiempo real mediante la utilización de librerías especializadas. Para este propósito, un script en Python accede y lee variables desde Firebase, permitiendo su representación gráfica instantánea. EL Script permite la lectura de los datos desde Firebase que se ha conectado mediante autenticaciones de llaves para evitar así violaciones de seguridad. Comprueba si hay cambios con respecto a los datos anteriores, si encuentra un cambio actualiza la variable correspondiente y actualiza la gráfica, si supera los cien datos elimina el registro más antiguo permitiendo así solo observar la información importante del sistema y actualizada el diagrama de flujo que sigue el documento se puede observar en la Figura 3.16. Los Scripts se pueden ver con más detalle en los Anexos D, E y el F para las variables correspondientes.

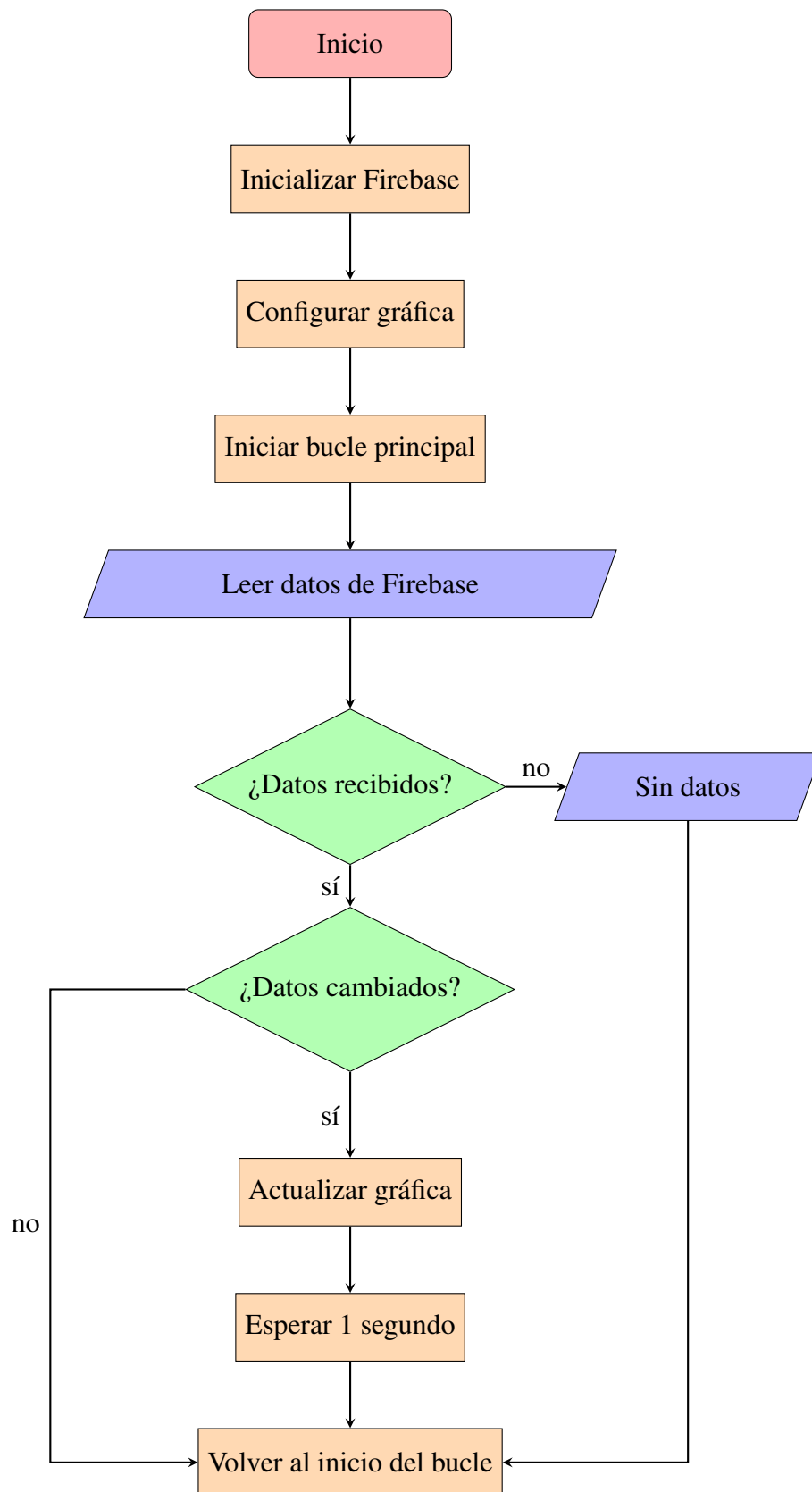


Figura 3.16: Diagrama de flujo del monitoreo y actualización de datos.

Capítulo IV

Pruebas de Funcionamiento

Este capítulo describe y analiza las pruebas de funcionamiento de la red IoT bajo condiciones operativas reales. Debido a la importancia de garantizar un rendimiento fiable y eficiente, se llevan a cabo evaluaciones exhaustivas para determinar la respuesta del sistema ante diversos escenarios y posibles complicaciones.

La evaluación se enfoca en la robustez del sistema, la latencia de la red, y su capacidad para manejar altas cargas y situaciones de estrés. Este análisis es crucial para confirmar la funcionalidad del sistema en el contexto para el que fue diseñado, asegurando que cumple con los requisitos y expectativas.

Además, se simulan condiciones adversas para observar la respuesta de la red ante posibles interferencias o fallos, lo que facilita la identificación de áreas de mejora y el fortalecimiento de la infraestructura. El objetivo de estas pruebas es validar de manera concreta la operatividad de la red, confirmando su capacidad para desempeñar eficaz y eficientemente su función en el ámbito de la agricultura inteligente.

4.1. Descripción de las Pruebas

Cada prueba realizada está diseñada para evaluar un aspecto específico de la red. En las secciones siguientes, detallaremos el proceso de cada prueba y los resultados obtenidos. Este enfoque nos permitirá comprender mejor cada componente de la red y su contribución al funcionamiento global.

4.2. Pruebas de Funcionamiento

Para asegurar que cada componente del sistema funcione según las especificaciones y que todos los componentes trabajen juntos de manera efectiva, es crucial verificar el correcto funcionamiento de cada uno de ellos. Este proceso comienza con la transferencia adecuada de datos desde los sensores hacia el gateway, seguido por su correcta transmisión hacia The Things Network. Para evaluar esto de manera efectiva, se llevarán a cabo simulaciones de envío de datos. Estos datos simulados permitirán verificar su integridad y precisión a través de la herramienta "Live Data", una funcionalidad clave de la plataforma de The Things Network. Este enfoque no solo asegura que cada segmento de la red funcione conforme a las expectativas, sino que también permite identificar y corregir cualquier discrepancia en tiempo real.

Además, se realizan pruebas unitarias para verificar cada sensor individualmente, seguidas de pruebas de integración que aseguran que los componentes interactúan correctamente entre sí y con la central de procesamiento. Este método riguroso de evaluación garantiza la fiabilidad y eficacia de la red en su conjunto.

Para evaluar inicialmente el correcto funcionamiento de los nodos de datos a través de la

plataforma The Things Network, la Figura 4.1 presenta la salida de la consola Live Data mencionada anteriormente. En ella se observa claramente que los datos de los nodos finales llegan cada 90 segundos, conforme a la configuración establecida.

Time	Entity ID	Type	Data preview
↑ 21:37:51	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.559, ADC_CH1V: 2.754, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:36:21	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.585, ADC_CH1V: 2.754, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:34:51	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.582, ADC_CH1V: 2.795, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:33:21	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.584, ADC_CH1V: 2.724, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:31:51	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.569, ADC_CH1V: 2.763, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:30:21	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.567, ADC_CH1V: 2.744, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:28:51	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.566, ADC_CH1V: 2.827, ADC_CH4V: 0.001, BatV: 3.667, Digital_IStatus: ...
↑ 21:27:21	eui-a84041239358f65b	Forward uplink data message	DevAddr: 26 0C D1 72 <> Payload: { ADC_CH0V: 1.586, ADC_CH1V: 2.765, ADC_CH4V: 0.001, BatV: 3.67, Digital_IStatus: ...

Figura 4.1: Verificación de llegada de datos.

Posteriormente, verificamos el correcto funcionamiento de los servicios esenciales en la máquina virtual. Utilizando la consola SSH, examinamos el estado de los servicios Apache2, PHP-FPM y MySQL, asegurando así la operatividad eficiente del sistema.

Además, confirmamos el correcto funcionamiento del documento puente entre MySQL y Firebase. Para ello, se utilizan mensajes que explican claramente la transmisión exitosa de datos. La salida SSH de la máquina virtual, mostrada en la Figura 4.2, refleja esta información mientras se ejecuta el archivo Python.

```
SSH en el navegador SUBIR ARCHIVO DESCARGAR ARCHIVO

File "/usr/lib/python3.8/ssl.py", line 1202, in send
    return self._sslobj.write(data)
KeyboardInterrupt

xibernetiq@loratest-1:~$ ls
add-google-cloud-ops-agent-repo.sh bridge.py bridgetest.py graph.py
xibernetiq@loratest-1:~$ sudo nano bridgetest.py
xibernetiq@loratest-1:~$ python3 ./bridgetest.py
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:35:32', 'ADC_CH4V': 0.001, 'ADC_CH0V': 0.242, 'ADC_CH1V': 0.232, 'TempC1': 23.0}
<<< { ADC_CH0V: 0.242 ADC_CH1V: 0.232 }
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:46:40', 'ADC_CH4V': 0.001, 'ADC_CH0V': 2.002, 'ADC_CH1V': 2.12, 'TempC1': 25.1}
<<< { ADC_CH0V: 2.002 ADC_CH1V: 2.12 }
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:48:10', 'ADC_CH4V': 0.001, 'ADC_CH0V': 2.059, 'ADC_CH1V': 2.117, 'TempC1': 25.1}
<<< { ADC_CH0V: 2.059 ADC_CH1V: 2.117 }
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:49:39', 'ADC_CH4V': 0.001, 'ADC_CH0V': 2.067, 'ADC_CH1V': 2.124, 'TempC1': 25.1}
<<< { ADC_CH0V: 2.067 ADC_CH1V: 2.124 }
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:51:09', 'ADC_CH4V': 0.001, 'ADC_CH0V': 2.067, 'ADC_CH1V': 2.126, 'TempC1': 25.1}
<<< { ADC_CH0V: 2.067 ADC_CH1V: 2.126 }
Base de datos " https://loranetworktest-default-rtdb.firebaseio.com/ "
>>> {'BatV': '16 Jul 2024 19:52:39', 'ADC_CH4V': 0.001, 'ADC_CH0V': 2.062, 'ADC_CH1V': 2.122, 'TempC1': 25.1}
<<< { ADC_CH0V: 2.062 ADC_CH1V: 2.122 }
```

Figura 4.2: Salida SSH del archivo Python en ejecución.

El objetivo de este estudio es evaluar la eficiencia del sistema, considerando la latencia de la red, la capacidad de procesamiento del servidor y la respuesta ante cargas de datos significativas. Se analiza la latencia de los datos transmitidos a través del intervalo temporal desde el momento de su envío desde el nodo final hasta su procesamiento en los servidores. Este análisis también incluye el tiempo que los datos procesados tardan en ser transmitidos para su visualización final y almacenamiento en la nube. Este enfoque ofrece una medida precisa de la eficiencia del sistema en términos de tiempo de respuesta, esencial para asegurar la fiabilidad y efectividad de la red en operaciones en tiempo real. Las Figuras 4.3, 4.4, y 4.5 ilustran las gráficas en tiempo real de las variables procesadas que reflejan el estado actual de los cultivos.

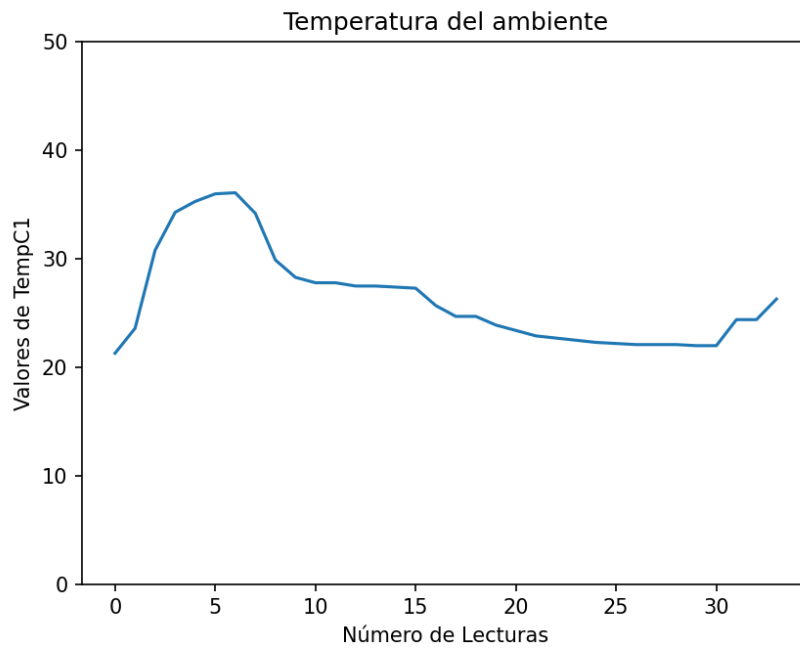


Figura 4.3: Gráfica en tiempo real de la Temperatura.

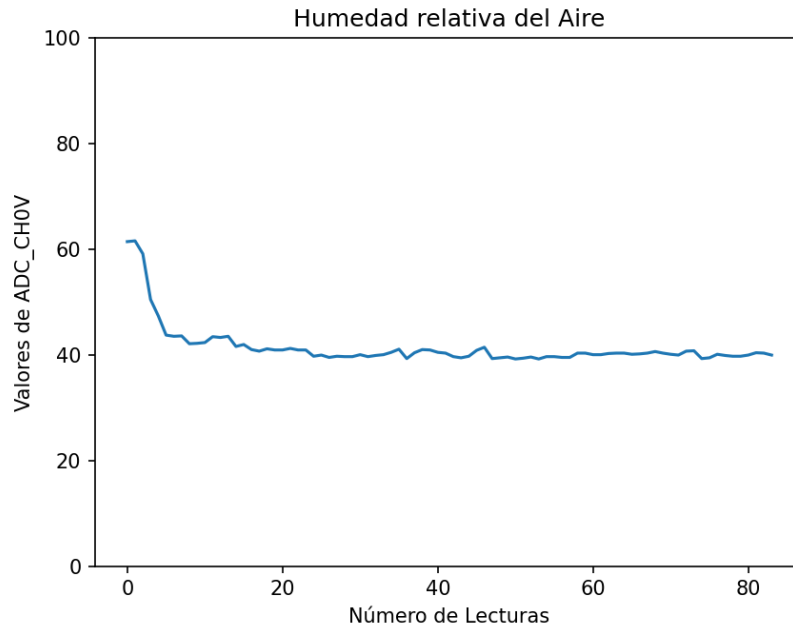


Figura 4.4: Gráfica en tiempo real de la humedad del aire.

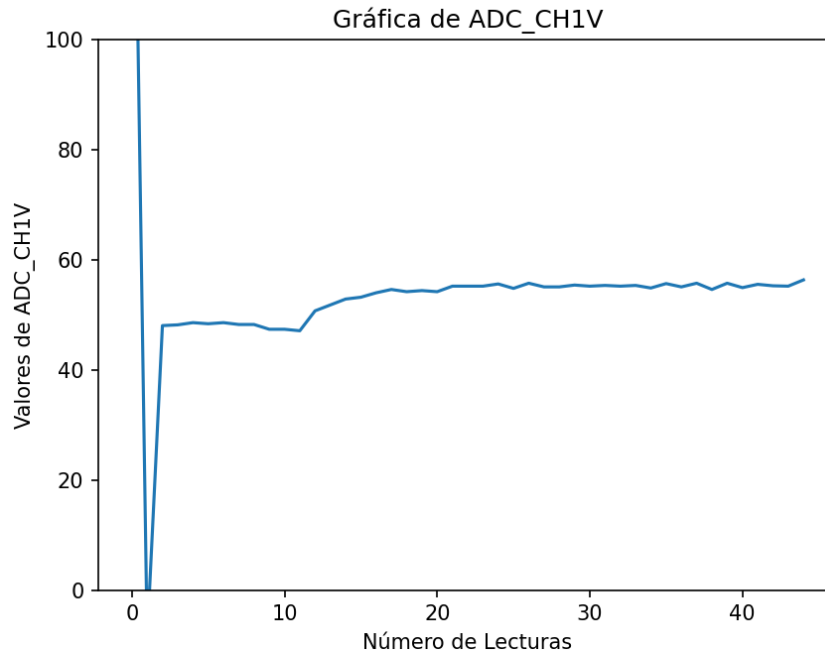


Figura 4.5: Gráfica en tiempo real de la humedad del suelo.

En el análisis del desempeño de redes IoT, la latencia es un indicador clave de eficiencia. Este tiempo de respuesta es crítico en aplicaciones que dependen de la transmisión en tiempo real. Para evaluar adecuadamente la red, es crucial medir la latencia desde el punto inicial en The Things Network hasta el destino final en Firebase.

Se define la latencia como el intervalo que requieren los datos para trasladarse a través de la red, desde su origen hasta su destino. Esta métrica es fundamental para evaluar la capacidad de la red de manejar datos en tiempo real y para identificar posibles cuellos de botella o ineficiencias. El análisis se fundamenta en datos obtenidos de The Things Network y Firebase, lo cual proporciona una base sólida para las conclusiones derivadas.

La Figura 4.6 ilustra 20 puntos críticos recopilados durante un pico de tráfico de datos en la red. Este análisis permite evaluar el comportamiento de la red bajo condiciones de alta demanda

y confirmar su eficacia en situaciones similares.

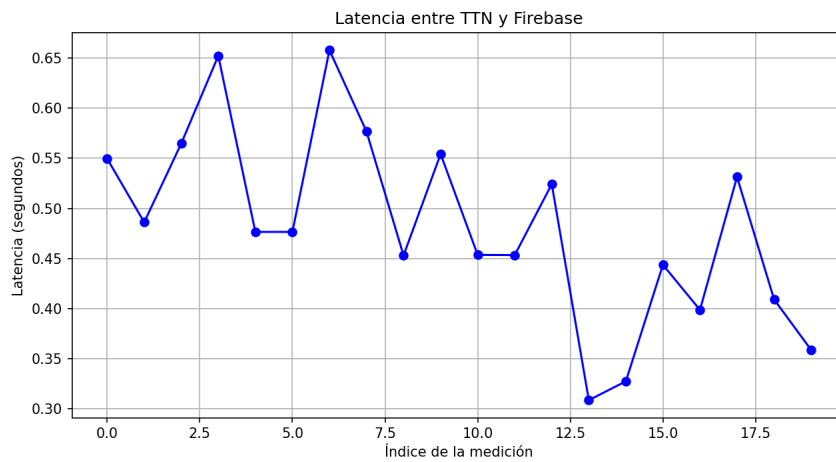


Figura 4.6: Latencia de la red

La latencia varía entre aproximadamente 0.3 y 0.7 segundos. Esta variabilidad puede ser representativa de las fluctuaciones típicas que se esperarían en una red real debido a factores como la congestión de la red, la capacidad de procesamiento del servidor y la eficiencia de las rutas de datos.

La latencia no sigue un patrón claramente definido de aumento o disminución a lo largo del tiempo, lo cual es característico de sistemas donde las condiciones de red y tráfico pueden cambiar rápidamente y ser impredecibles.

A pesar de la variabilidad, la latencia se mantiene dentro de un rango relativamente estrecho, lo que sugiere que el sistema es capaz de mantener un nivel de rendimiento consistente bajo las condiciones simuladas.

Una latencia de menos de un segundo en la mayoría de los casos es generalmente aceptable para muchas aplicaciones IoT, especialmente en contextos que no son de misión crítica. Sin embargo, para aplicaciones que requieren tiempos de respuesta casi instantáneos, sería crucial

explorar maneras de reducir aún más esta latencia.

4.3. Pruebas de Rendimiento

Para estas pruebas, el análisis se centra principalmente en el rendimiento de la máquina virtual, que es responsable de la transmisión de datos. Se examinan las métricas generadas por la misma, las cuales son cruciales para evaluar la respuesta de la red ante el tráfico real de datos.

4.3.1. Uso del CPU

La Figura 4.7 presenta un gráfico del uso de la CPU de una instancia de máquina virtual en Google Cloud, específicamente en el proyecto identificado como loranetworktest en la zona us-central1-b. La métrica observada es la "VM Instance - CPU utilization", la cual muestra variaciones en el uso del CPU a lo largo del tiempo.

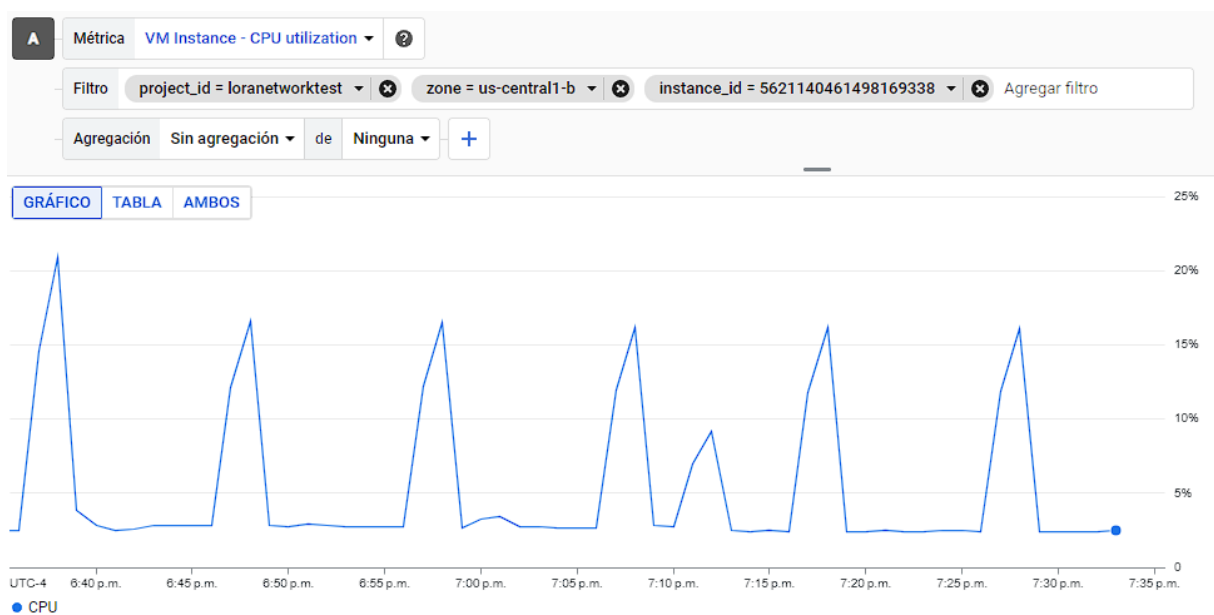


Figura 4.7: Uso del CPU de la máquina virtual

Durante el periodo observado, se pueden notar picos recurrentes en la utilización del CPU que alcanzan aproximadamente el veinte por ciento, seguidos de caídas abruptas a niveles cercanos al cero por ciento. Estos picos ocurren en intervalos regulares de aproximadamente noventa segundos, lo que coincide con el intervalo de transmisión de datos del sistema. Esta regularidad sugiere que los picos están directamente relacionados con la actividad de transmisión de datos desde los nodos hacia la máquina virtual.

La baja utilización promedio del CPU, con niveles que rara vez superan el veinte por ciento, indica que la máquina virtual tiene capacidad suficiente para manejar su carga de trabajo actual. Sin embargo, los picos recurrentes señalan momentos de alta demanda que, aunque manejables, deberían ser monitoreados para asegurar que no se conviertan en un cuello de botella si la carga de trabajo aumenta.

4.3.2. Uso de Memoria

La Figura 4.8 ilustra la utilización de memoria de una instancia de máquina virtual específicamente en el proyecto loranetworktest y en la zona us-central1-b. La métrica mostrada es la "VM Instance - Memory utilization", que refleja cómo varía el uso de la memoria a lo largo del tiempo.

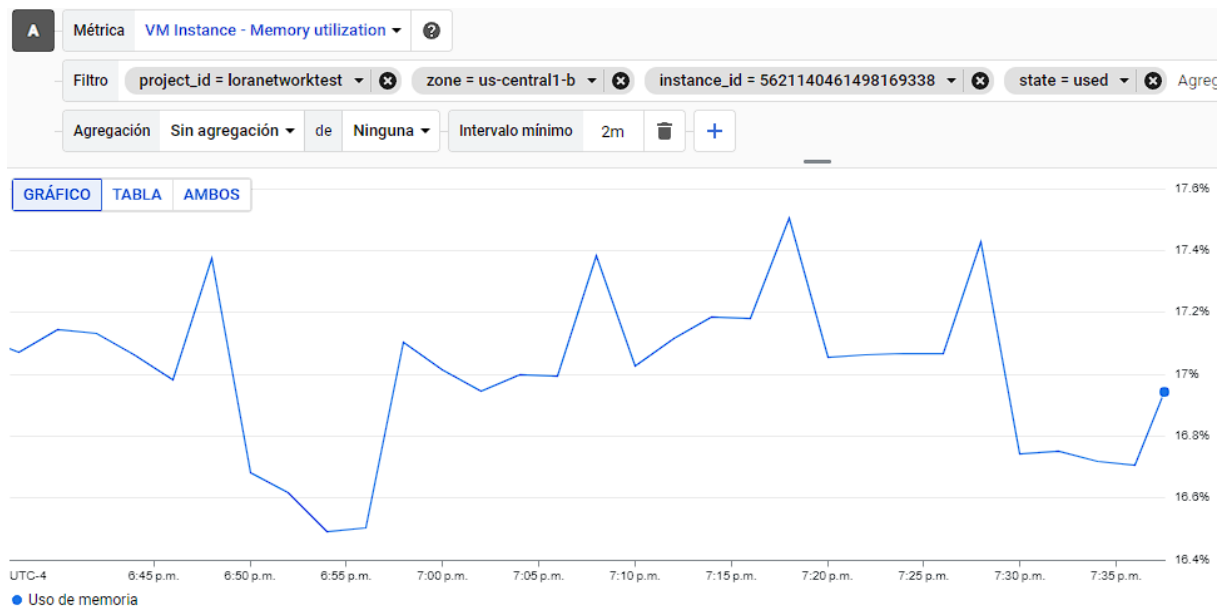


Figura 4.8: Uso de memoria de la máquina virtual

El gráfico exhibe fluctuaciones en el uso de la memoria, con un patrón que parece menos regular que el del uso de la CPU. La utilización de memoria oscila entre aproximadamente el dieciséis punto ocho cinco por ciento y el diecisiete punto seis por ciento, lo que sugiere que la carga de memoria es moderadamente baja y estable. Este rango indica que la máquina virtual está manejando bien su carga de memoria actual sin signos evidentes de estrés o de agotamiento de recursos.

Las variaciones en el uso de memoria podrían atribuirse a las operaciones específicas que se están ejecutando en la máquina virtual, posiblemente incluyendo la manipulación de datos y operaciones de base de datos que fluctúan con las demandas de procesamiento. Aunque no se observan picos significativos que sugieran fugas de memoria o consumo excesivo, es importante mantener un monitoreo continuo, especialmente durante periodos de carga intensa o cuando se implementen nuevas aplicaciones.

4.3.3. Tráfico de Red

La Figura 4.9 ilustra el tráfico de red, tanto de bytes recibidos como enviados, de una instancia de máquina virtual en el proyecto loranetworktest, ubicado en la zona us-central1-b. La gráfica muestra un patrón recurrente y relativamente estable en el tráfico de red a lo largo del tiempo observado.

El gráfico exhibe picos regulares en el tráfico de red, tanto para los datos enviados como los recibidos. Estos picos indican momentos de actividad intensa, posiblemente relacionados con la transmisión de datos programada o respuestas a solicitudes externas. Los picos de tráfico enviados alcanzan hasta casi sesenta y cinco kilobits por segundo, mientras que los picos de tráfico recibido son ligeramente más bajos pero siguen un patrón similar.

La regularidad de estos picos sugiere que la máquina virtual está gestionando una carga de trabajo consistente y que las transmisiones de datos están siendo manejadas de manera eficiente. No se observan caídas significativas ni inactividad prolongada que podría indicar problemas de conectividad o fallos en el sistema.

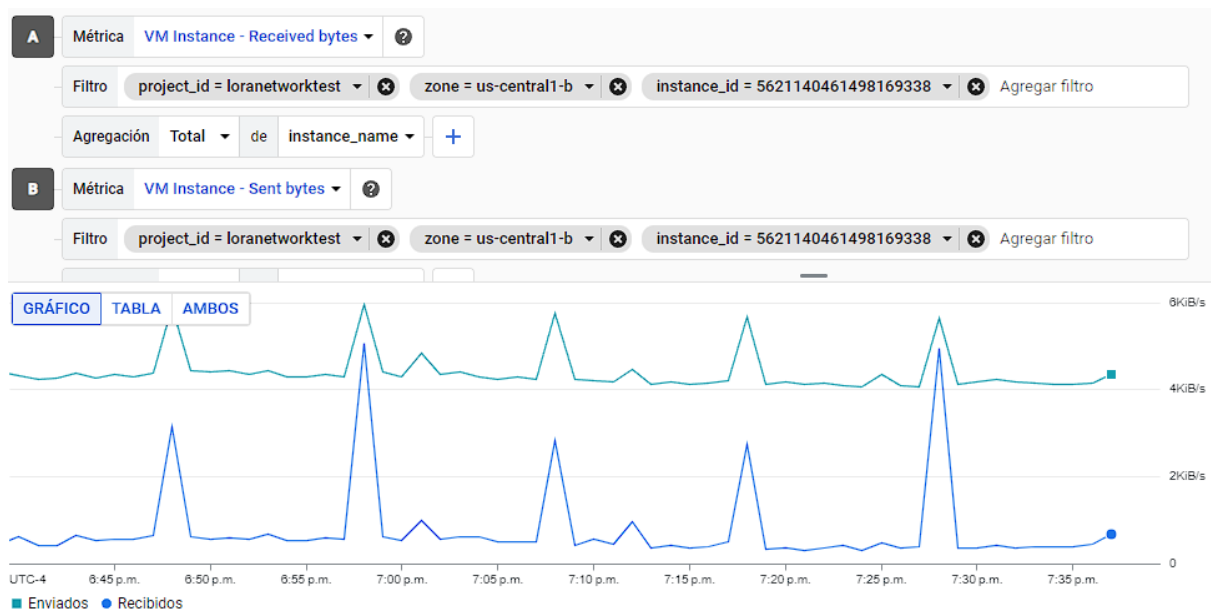


Figura 4.9: Tráfico de red de la máquina virtual

4.3.4. Uso de Recursos Externos

La Figura 4.10 muestra dos líneas distintas que representan diferentes tipos de uso de recursos: una para los servicios de Google y otra para un recurso externo vinculado a un proyecto diferente. A lo largo del tiempo observado, el uso de servicios de Google se mantiene relativamente bajo y estable, mientras que el uso del recurso externo muestra variaciones significativas.

El área sombreada en azul claro representa el uso del recurso externo, que muestra picos y caídas regulares, indicando fluctuaciones en la demanda o en la actividad de la máquina virtual. Estas variaciones pueden deberse a operaciones programadas o a respuestas a solicitudes específicas que requieren la interacción con este recurso externo.

Por otro lado, la línea verde, que representa el uso de los servicios de Google, se mantiene cerca del eje horizontal, sugiriendo que estos servicios no representan una carga significativa para la máquina virtual. Este bajo nivel de uso indica que los servicios de Google, aunque

constantemente activos, no imponen una demanda intensiva de recursos.

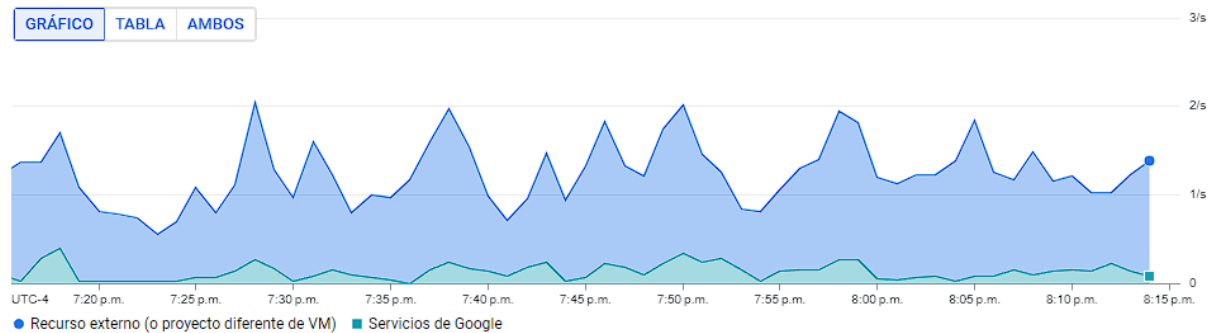


Figura 4.10: Uso de recursos externos

4.3.5. IOPS del Disco

La Figura 4.11 muestra el gráfico de las operaciones de entrada/salida por segundo (IOPS) del disco en una máquina virtual implementada en Google Cloud. Este gráfico destaca las actividades de lectura y escritura sobre el disco durante un periodo específico.

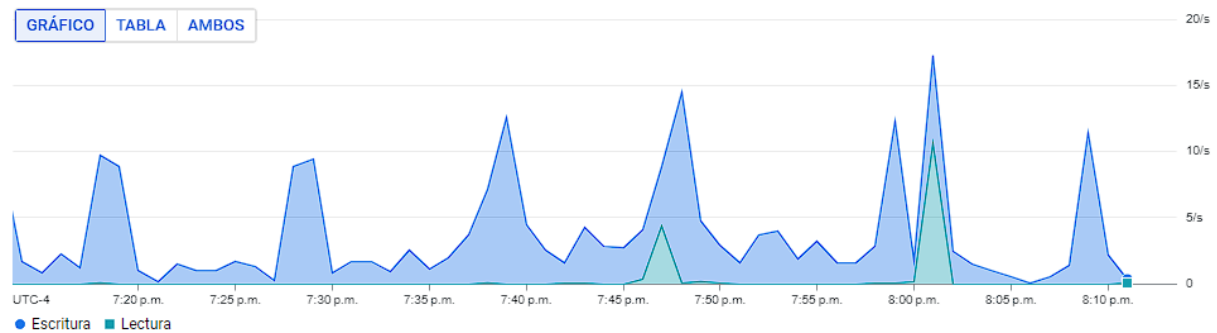


Figura 4.11: IOPS del disco

El gráfico ilustra claramente los picos de actividad tanto en las operaciones de escritura como de lectura, mostrando un patrón de variabilidad significativa en el uso del disco. Las operaciones de escritura (representadas en azul) exhiben picos más altos y más frecuentes en comparación con las operaciones de lectura (en verde), indicando una mayor demanda de escri-

tura durante el intervalo observado.

El pico más prominente en las operaciones de escritura alcanza cerca del quince por ciento, mientras que las operaciones de lectura alcanzan un máximo mucho más bajo. Este patrón sugiere que la máquina virtual está procesando y almacenando grandes volúmenes de datos, posiblemente relacionados con la transmisión de datos o actualizaciones de la base de datos que requieren una escritura intensiva en el disco.

4.3.6. Capacidad de Procesamiento del Disco

La Figura 4.12 muestra la capacidad de procesamiento del disco de una máquina virtual en Google Cloud, destacando las operaciones de lectura y escritura a lo largo de un intervalo de tiempo específico. Este gráfico ofrece una visión detallada del tráfico de datos en términos de megabytes por segundo.

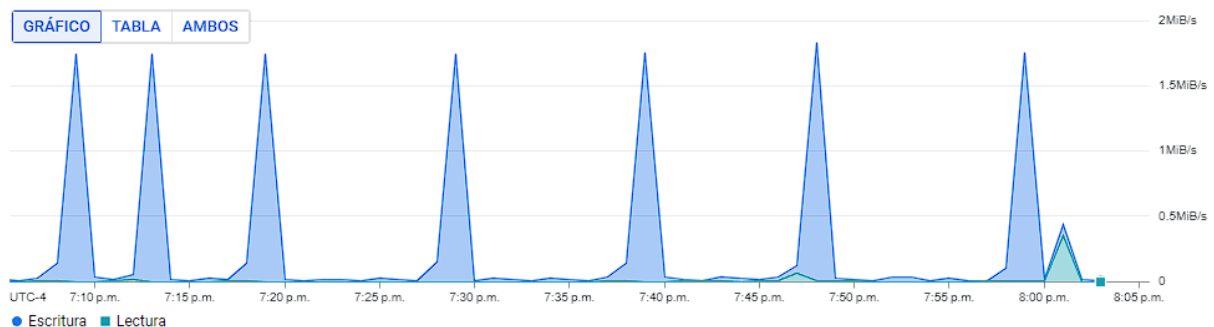


Figura 4.12: Capacidad de procesamiento del disco.

El gráfico exhibe picos agudos en las operaciones de escritura (en azul), con momentos en los que la capacidad de escritura alcanza casi los dos megabytes por segundo. Las operaciones de lectura (en verde) son significativamente más bajas durante el periodo observado, lo que indica que las tareas de escritura dominan la actividad del disco en estos momentos.

Los picos en la capacidad de procesamiento del disco muestran una periodicidad clara, con incrementos notables que se suceden de forma regular. Estos picos podrían estar relacionados con tareas programadas que ejecutan operaciones de escritura intensiva, tales como actualizaciones de bases de datos o procesamiento de grandes volúmenes de datos generados por la red IoT.

Capítulo V

Conclusiones y Trabajo Futuro

5.1. Conclusiones

- Este trabajo desarrolló una puerta de enlace utilizando tecnología LoRa, diseñada para facilitar la transmisión de datos críticos desde sensores conectados a dispositivos finales, específicamente para el monitoreo de cultivos. La implementación se llevó a cabo mediante una red propietaria, integrada a un servidor privado alojado en una máquina virtual en Google Cloud. Esta configuración no solo aseguró el almacenamiento adecuado de las variables sino también su eficiente transferencia hacia Firebase, una base de datos en tiempo real. Este enfoque permitió un procesamiento, visualización y análisis efectivo de los datos recopilados, demostrando la viabilidad y eficacia de la solución propuesta para mejorar la gestión de cultivos urbanos.
- La puerta de enlace diseñada para este proyecto incorpora tecnologías ampliamente utilizadas en el ámbito de IoT, seleccionadas específicamente por su eficacia en investiga-

ciones anteriores. La elección de la tecnología LoRa como protocolo principal de comunicación fue motivada por su adecuación a los requisitos de carga de datos y ancho de banda del proyecto. Además, se optó por utilizar software libre y plataformas que ofrecen recursos gratuitos, con el fin de asegurar que la red sea accesible para la comunidad y sostenible en términos de costos. Esta estrategia garantiza no solo la funcionalidad del sistema sino también su accesibilidad y adaptabilidad en un contexto comunitario.

- El manejo eficaz de la información de conexión de red es crucial para identificar y resolver problemas que puedan surgir, especialmente bajo condiciones de aumento de la carga de datos o en situaciones que requieran escalabilidad futura. Es fundamental determinar el tiempo necesario para adaptarse a estos cambios. El sistema de gestión de red desempeña un papel esencial al proporcionar datos relevantes sobre la necesidad de escalabilidad, especialmente cuando el contexto del proyecto lo exige.
- Se comprobó que la red es completamente viable para el monitoreo de cultivos. La transmisión eficiente y precisa de datos ha demostrado ser clave, permitiendo un monitoreo efectivo y continuo. Estos resultados validan la robustez de la red y su potencial para aplicaciones en tiempo real en ambientes agrícolas, asegurando que puede manejar adecuadamente tanto las demandas actuales como las futuras expansiones previstas.

5.2. Recomendaciones

- La red demostró ser capaz de transmitir datos de manera eficiente desde los nodos. Para futuras mejoras, se recomienda explorar la posibilidad de escalar la red, incorporando

más dispositivos finales y aumentando la carga de datos. Esto es factible siguiendo las configuraciones detalladas en este documento y ajustando los componentes clave para asegurar una transmisión de datos óptima.

- La seguridad es un aspecto crítico de la red; una configuración inadecuada puede exponerla a vulnerabilidades, incluido el potencial robo de datos por entidades no autorizadas. Por esta razón, se recomienda encarecidamente explorar y adoptar herramientas de seguridad avanzadas. Además, es crucial implementar configuraciones adicionales para reforzar la fiabilidad de la red y garantizar la confianza de los usuarios.
- La tecnología LoRa es ampliamente compatible con diversas plataformas IoT, un hecho respaldado por numerosos estudios. Dado este contexto, se recomienda explorar nuevas plataformas para la transmisión de datos, lo que podría permitir la integración de herramientas específicas de cada plataforma y, consecuentemente, mejorar la usabilidad de la red.

5.3. Trabajo a Futuro

- La implementación de un almacenamiento eficiente de datos en la nube facilita su análisis y procesamiento subsiguiente. Se recomienda incorporar herramientas avanzadas de análisis de datos, incluyendo algoritmos de aprendizaje automático y aplicaciones de inteligencia artificial. Estas tecnologías podrían mejorar significativamente la precisión de los datos en tiempo real, proporcionando insights más certeros que influyen directamente en la toma de decisiones.

- Para la gestión automatizada de cultivos urbanos, se propone la integración de actuadores dentro de la red IoT. Es crucial evaluar la compatibilidad de la tecnología LoRa con estos nuevos sistemas, los cuales facilitarán el control remoto y la automatización del cultivo, mejorando así la eficiencia y la respuesta a las necesidades específicas del mismo.
- A futuro, la red IoT podría expandirse a otras áreas de aplicación que requieran monitoreo continuo. Al estar ya establecida una infraestructura IoT local, se abren numerosas posibilidades para integrar diversos dispositivos, ampliando así el alcance y las funcionalidades de la red. Esta expansión permitiría adaptar la red a diferentes contextos y necesidades, incrementando su utilidad y eficacia.

Bibliografía

- [1] M. A. Ertürk, M. Aydin, T. Büyükakkaşlar, and H. Evirgen, “A survey on lorawan architecture, protocol and technologies,” *Future Internet*, vol. 11, p. 216, 10 2019.
- [2] “Lorawan - one of the most interesting iot technologies on the market,” 2022.
- [3] A. Siegner, J. Sowerwine, and C. Acey, “Does urban agriculture improve food security? examining the nexus of food access and distribution of urban produced foods in the united states: A systematic review,” *Sustainability*, vol. 10, no. 9, p. 2988, 2018. Special Issue: Local Food Initiatives in the World’s Cities.
- [4] Food and Agriculture Organization, “The urban, peri-urban and family agriculture program in cuba and its impacts.” Online, 2015. Accessed: 2024-07-07.
- [5] A. Adamsone-Fiskovica and M. Grivins, “Understanding the potential of sustainability turn in farming: review of sociotechnical adoption factors of agri-environmental cropping practices,” *Cambridge University Press*, May 2024. Accessed: 2024-06-07.
- [6] P. G. Khedekar, N. R. Deshpande, and A. D. Shaligram, “Iot-enabled smart crop monitoring systems for sustainable agriculture,” *IJERT*, vol. 12, p. Article ID

IJERTV12IS060042, Jun 2023. Accessed: Your Access Date Here.

- [7] ANTT Robotics Ltd, “Lora and iot technology revolutionizing agriculture: A farmer’s guide to modern farming.” Online, Jan 2024. Prepare For The Future Of Work.
- [8] M. K. Senapaty, A. Ray, and N. Padhy, “Iot-enabled soil nutrient analysis and crop recommendation model for precision agriculture,” *Computers*, vol. 12, p. 61, Mar 2023. Available online 12 March 2023.
- [9] C. Ynzunza, J. Izar, J. Bocarando, F. Pereyra, and M. Osorio, “El entorno de la industria 4.0: Implicaciones y perspectivas futuras.” Instituto Tecnológico de Aguas Calientes, 2017. Trabajo no publicado.
- [10] N. Kouvelas, V. Rao, V. Prasad, G. Tawde, and K. Langendoen, “Politely scaling lorawan,” *Embedded and Networked Systems*, pp. 25–36, 2020.
- [11] M. A. Ahmed, J. L. Gallardo, M. D. Zuniga, M. A. Pedraza, G. Carvajal, N. Jara, and R. Carvajal, “Lora based iot platform for remote monitoring of large-scale agriculture farms in chile,” *Sensors*, vol. 22, p. 2824, Apr 2022. Published online 7 April 2022.
- [12] S. Aguilar, “Implementación de un sistema iot basado en lora para el control de riego en cultivos en Perú,” Master’s thesis, Universidad de Lima, Peru, 2020. Tesis de maestría.
- [13] P. Salazar, M. Vásquez, S. Montoya, J. García, and D. Robalino, “Implementación de tecnología lora en el monitoreo de cultivos de lechugas para evitar estrés abiótico,” tesis de maestría, Universidad de Quito, Ecuador, 2020.

- [14] G. P. N. Hakim, M. H. Habaebi, S. F. Toha, M. R. Islam, S. H. B. Yusoff, E. Y. T. Adesta, and R. Anzum, "Near ground pathloss propagation model using adaptive neuro fuzzy inference system for wireless sensor network communication in forest, jungle and open dirt road environments," *Sensors*, vol. 22, no. 9, p. 3267, 2022.
- [15] D. F. P. Bernardo Henrique Leso, Daisy Valle Enrique, "O papel do ecossistema de inovação para desenvolver uma agricultura inteligente," *Exacta ingenieria de Produccion*, 2021.
- [16] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.
- [17] Semtech, "Lora technology overview." <https://www.semtech.com/lora>, 2023. Accessed: Sept. 10, 2023.
- [18] The Things Network, "What is lorawan?," 2024. Accedido: Mar. 12, 2024.
- [19] Semtech Corporation, "Creating sustainable public lighting with lorawan," 2021. Accessed: Mar. 12, 2024.
- [20] L. A. T. M. Workgroup, "A technical overview of lora and lorawan," tech. rep., LoRa Alliance, November 2015. Accessed: Mar. 12, 2024.
- [21] E. MDPI, "Advancing performance in lorawan: Implications for iot," *Electronics MDPI*, 2021. Accessed: Sept. 20, 2023.

- [22] M. Saari, A. M. Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, "Lora and lorawan testbeds: A review," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1544–1549, May 2018.
- [23] I. Staff, "Lorawan: Powering iot forward," 2023. Accessed: Sept. 20, 2023.
- [24] M. S. Staff, "Future industrial applications: Exploring lpwan-driven iot protocols," *Sensors*, vol. 24, no. 8, p. 2509, 2023. Accessed: Sept. 20, 2023.
- [25] M. A. Ertürk, M. A. Aydın, M. T. Büyükakkaşlar, and H. Evirgen, "A survey on lorawan architecture, protocol and technologies," *Future Internet*, vol. 11, p. 216, Oct 2019.
- [26] A. B. Johnson and C. D. Smith, "Impact of interference on lora networks in urban environments," *IEEE Communications Magazine*, vol. 56, pp. 120–127, Jul 2018.
- [27] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melià-Seguí, and T. Watteyne, "Understanding the limits of lorawan," *arXiv*, 2017. Accessed: Sept. 20, 2023.
- [28] R. Piyare and otros, "On-demand lora: Asynchronous tdma for energy efficient and low latency communication in iot," *Sensors*, vol. 18, no. 11, p. 3718, 2018.
- [29] D.-H. K. Hazard Farahd and J.-J. Pyun, "Resource allocation to massive internet of things in lorawans," *Sensors*, vol. 20, no. 9, p. 2645, 2020.
- [30] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-demand lora: Asynchronous tdma for energy efficient and low latency communication in iot," *Sensors*, vol. 18, no. 11, p. 3718, 2018.

- [31] The Things Network, “Frequency plans by country,” 2023. Accessed: Oct. 1, 2023.
- [32] IoT For All, “Lorawan - most common applications and use cases,” 2023. Accessed: Oct. 1, 2023.
- [33] L. Vangelista *et al.*, “On the lora modulation for iot: Waveform properties and spectral analysis,” *IEEE Access*, vol. 8, pp. 131857–131868, 2020.
- [34] F. Giuliano, A. Pagano, D. Croce, G. Vitale, and I. Tinnirello, “Adaptive algorithms for batteryless lora-based sensors,” *Sensors*, vol. 23, no. 14, p. 6568, 2023.
- [35] S. Hosseinzadeh, M. Ashawa, N. Owoh, H. Larijani, and K. Curtis, “Explainable machine learning for lorawan link budget analysis and modeling,” *Sensors*, vol. 24, no. 3, p. 860, 2024. Published: January 29, 2024.
- [36] R. Sanchez-Iborra, J. Sánchez-Gómez, S. Pérez, P. J. Fernández, J. Santa, J. L. Hernández-Ramos, and A. F. Skarmeta, “Enhancing lorawan security through a lightweight and authenticated key management approach,” *Sensors*, vol. 18, no. 6, p. 1833, 2018. Published: June 5, 2018.
- [37] TEKTELIC, “Iot in real estate,” 2023. Accessed: Oct. 1, 2023.
- [38] The Things Network Barranquilla, “The things network barranquilla community update.” Online, 2024. Accessed: July 10, 2024.
- [39] T. Almaaitah, D. Joksimovic, and T. Sajin, “Real-time iot-enabled water management for rooftop urban agriculture using commercial off-the-shelf products,” *Chemical Processes*,

- vol. 10, no. 1, p. 34, 2022. Presented at the 1st International Online Conference on Agriculture—Advances in Agricultural Science and Technology, February 2022.
- [40] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, “Smart farming: Internet of things (iot)-based sustainable agriculture,” *Agriculture*, vol. 12, no. 10, p. 1745, 2022. Accessed: 7 June 2024.
- [41] The Things Industries, “Webhooks integration documentation.” Online, 2024. Accessed: July 10, 2024.
- [42] Dragino, “Lorawan end node: Sn50v3-lb.” Dragino Official Website, Jul 2023. Technical specifications and applications for the SN50V3-LB LoRaWAN device.
- [43] TuxCare, “Linux for iot: Key benefits and considerations.” TuxCare Blog, 2023. Accedido el 7 de julio de 2024.
- [44] S. Kundu and otros, “Cloud monitoring system for agriculture using internet of things,” in *Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, (Vellore, India), pp. 1–6, IEEE, 2020.
- [45] DigitalOcean, “Apache vs nginx: Practical considerations.” DigitalOcean Community Tutorials, 2023. Accessed on 7 July 2024.
- [46] M. Eyada *et al.*, “Performance evaluation of iot data management using mongodb versus mysql databases in different cloud environments,” *Journal of Database Management*, vol. 12, no. 3, pp. 45–59, 2020. Accessed on 7 July 2024.

- [47] A. T. Gaikwad *et al.*, “Firebase - overview and usage,” *Journal of Engineering and Technology Management*, vol. 12, p. 2824, 2022. Accessed on 7 July 2024.
- [48] Dragino, “Lsn50 lorawan end node.” Dragino Official Website, 2024. Accessed on 7 July 2024.
- [49] Analog Devices, “Ds18b20 programmable resolution 1-wire digital thermometer,” 2021.
- [50] Naylamp Mechatronics, “Sensor de temperatura y humedad relativa dht11.” Naylamp Mechatronics, 2024. Último acceso: julio 2024.
- [51] Naylamp Mechatronics, “Sensor de humedad de suelo fc-28.” Naylamp Mechatronics, 2024. Último acceso: julio 2024.
- [52] Dragino, “Lg308n - lorawan gateway user manual.” Dragino Wiki, 2024. Accessed on 7 January 2024.
- [53] Linuxize, “How to install php 8 on ubuntu 20.04.” Linux Tips, Tricks and Tutorials | Linuxize, 2024. Accessed on 4 July 2024.
- [54] DigitalOcean, “How to install linux, apache, mysql, php (lamp) stack on ubuntu 20.04.” DigitalOcean Community Tutorials, 2024. Accessed on 7 July 2024.

Anexos

Anexo A

Configuración de la Base de Datos MySQL

Este anexo incluye el código fuente de la configuración de la base de datos utilizada en el proyecto, almacenada en el archivo "dbconf.php". El código configura la conexión con la base de datos MySQL, especificando los parámetros de conexión como el servidor, el nombre de la base de datos, el usuario y la contraseña.

```
<?php
define('TIMEZONE', 'America/Guayaquil');
date_default_timezone_set(TIMEZONE);

define("DB_HOST", "localhost");
define("DB_USER", "xibernetiq");
define("DB_PASSWORD", "*1C79B79639BA5AE3667BE52D00346929715A84CF");
define("DB_DATABASE", "lms50");

$conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_DATABASE);

if ($conn->connect_error) {
    echo "Failure";
    die("Connection failed: " . $conn->connect_error);
} else {
    echo "Connected successfully";
}
?>
```

Anexo B

Script PHP para Insertar Datos del Sensor DHT

Este anexo contiene el código fuente del script 'insertdht.php', que se utiliza para insertar datos capturados por los sensores DHT en la base de datos. Este script maneja la conexión con la base de datos y la inserción de registros de temperatura y humedad.

```
<?php
require_once("db_conf.php");

$data = file_get_contents("php://input");

$json = json_decode($data, true);

$received_at = date('Y-m-d H:i:s');

$end_device_ids = $json['end_device_ids'];
$device_id = $end_device_ids['device_id'];
$application_id = $end_device_ids['application_ids']['application_id'];

$uplink_message = $json['uplink_message'];
$ADC_CH0V = $uplink_message['decoded_payload']['ADC_CH0V'];
$ADC_CH1V = $uplink_message['decoded_payload']['ADC_CH1V'];
$ADC_CH4V = $uplink_message['decoded_payload']['ADC_CH4V'];
$TempC1 = $uplink_message['decoded_payload']['TempC1'];
$BatV = $uplink_message['decoded_payload']['BatV'];

$sqlCommand = "INSERT INTO lns50 (device_id, application_id, received_at,
ADC_CH0V, ADC_CH1V, ADC_CH4V, TempC1, BatV)
VALUES ('$device_id', '$application_id', '$received_at',
'$ADC_CH0V', '$ADC_CH1V', '$ADC_CH4V', '$TempC1', '$BatV')";

if (mysqli_query($conn, $sqlCommand)) {
    echo "1";
} else {
    echo "Error: " . $sqlCommand . "<br>" . mysqli_error($conn);
}
```

|?>

Anexo C

Script Python para la Interconexión de Datos

Este anexo incluye el código fuente del script Python 'bridgetest.py'. Este script es fundamental para la interconexión entre la base de datos y Firebase, facilitando la transferencia y sincronización de datos en tiempo real entre diferentes plataformas utilizadas en el proyecto.

```
import time
import mysql.connector
from mysql.connector import Error
from firebase import firebase
from datetime import datetime

bufferSize = 1024
urlDB = 'https://loranetworktest-default-rtdb.firebaseio.com/'
nombreDispositivo = 'bridgeTTN2FB'

def sendData(usuario, listaNombres, listaValores):
    datos = {}
    for key, value in zip(listaNombres, listaValores):
        datos[key] = value
    fb = firebase.FirebaseApplication(urlDB, None)
    resultado = fb.patch('// + usuario + '//', datos)
    return resultado

def getData(usuario):
    fb = firebase.FirebaseApplication(urlDB, None)
    result = fb.get('// + usuario + '//', None)
    ch0 = float(result.get('ADC_CH0V'))
    ch1 = float(result.get('ADC_CH1V'))
    return ch0, ch1

def connectMySQL():
    try:
        connection = mysql.connector.connect(host='localhost',
```

```

        database='lns50',
        user='xibernetiq',
        password='Automatic@123')

if connection.is_connected():
    dbInfo = connection.get_server_info()
    cursor = connection.cursor()
    cursor.execute("select database();")
    record = cursor.fetchone()

    cursor.execute("SELECT ADC_CH0V, ADC_CH1V, ADC_CH4V, TempC1,
BatV FROM lns50 ORDER BY ID DESC LIMIT 1")
    dato = cursor.fetchone()
    return dato
except Error as e:
    print("Error al intentar conectarse a MySQL", e)
    return None
finally:
    if connection.is_connected():
        cursor.close()
        connection.close()

def readTime():
    now = datetime.now()
    timestamp = datetime.timestamp(now)
    return datetime.fromtimestamp(timestamp).strftime("%d %b %Y %H:%M:%S")

if __name__ == '__main__':
    dataOld = None

    while True:
        data = connectMySQL()
        if data and data != dataOld:
            dataOld = data
            dataList = list(data)
            timeStamp = readTime()
            dataList[-1] = timeStamp
            timestamp

            result = sendData(nombreDispositivo, ['ADC_CH0V', 'ADC_CH1V',
'ADC_CH4V', 'TempC1', 'BatV', 'TimeStamp'], dataList)
            print('Base de datos "', urlDB, '"')
            print('>>> ', result)

            ch0, ch1 = getData(nombreDispositivo)
            print('<<< { ADC_CH0V:', ch0, ' ADC_CH1V:', ch1, '}')
            time.sleep(10)

```

Anexo D

Script Python para Graficar Temperatura

Este anexo contiene el código fuente del script Python 'graphTempC1.py'. Este script es utilizado para visualizar en tiempo real la temperatura capturada por los sensores, lo que permite un análisis y monitoreo efectivo del ambiente de los cultivos.

```
import firebase_admin
from firebase_admin import credentials, db
import matplotlib.pyplot as plt
import time

cred = credentials.Certificate(r'C:\Users\lozai\Documents\Tesis\
puttycopy\loranetworktest-firebase-adminsdk-73a9a-7dac061796.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://loranetworktest-default-rtdb.firebaseio.com/'
})

print("Firebase App initialized")

plt.ion()
fig, ax = plt.subplots()
y_data = []
line, = ax.plot(y_data)
ax.set_ylim(0, 50)
ax.set_title('Temperatura del ambiente')
ax.set_xlabel('N mero de Lecturas')
ax.set_ylabel('Valores de TempC1')

def update_graph(new_data):
    y_data.append(new_data)
    if len(y_data) > 100:
        y_data.pop(0)
    line.set_ydata(y_data)
    line.set_xdata(range(len(y_data)))
    ax.relim()
    ax.autoscale_view()
    fig.canvas.draw()
    fig.canvas.flush_events()
```

```
def get_data(path):
    ref = db.reference(path)
    return ref.get()

if __name__ == '__main__':
    last_adc_ch0v = None
    last_tempC1 = None
    while True:
        adc_ch0v = get_data('/bridgeTTN2FB/ADC_CH0V')
        tempC1 = get_data('/bridgeTTN2FB/TempC1')
        if adc_ch0v is not None and tempC1 is not None:
            if adc_ch0v != last_adc_ch0v or tempC1 != last_tempC1:
                print(f'tempC1: {tempC1}')
                update_graph(float(tempC1))
                last_adc_ch0v = adc_ch0v
                last_tempC1 = tempC1
            else:
                print('No data found for ADC_CH0V or TempC1.')
        time.sleep(1)
```


Anexo E

Script Python para Graficar el Canal ADC-CH0V

Este anexo contiene el código fuente del script Python 'graphADCCH0V.py'. Este script se utiliza para visualizar en tiempo real los datos del canal ADCCH0V, lo que es crucial para el monitoreo preciso de las variables físicas recogidas por los sensores.

```
import firebase_admin
from firebase_admin import credentials, db
import matplotlib.pyplot as plt
import time

cred = credentials.Certificate(r'C:\Users\lozai\Documents\Tesis\puttycopy\loranetworktest-firebase-adminsdk-73a9a-7dac061796.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://loranetworktest-default-rtdb.firebaseio.com/'
})

print("Firebase App initialized")

plt.ion()
fig, ax = plt.subplots()
y_data = []
line, = ax.plot(y_data)
ax.set_ylim(0, 100)
ax.set_title('Humedad relativa del Aire')
ax.set_xlabel('N mero de Lecturas')
ax.set_ylabel('Valores de ADC_CH0V')

def update_graph(new_data):
    y_data.append(new_data)
    if len(y_data) > 100:
        y_data.pop(0)
    line.set_ydata(y_data)
    line.set_xdata(range(len(y_data)))
    ax.relim()
    ax.autoscale_view()
```

```

fig.canvas.draw()
fig.canvas.flush_events()

def get_data(path):
    ref = db.reference(path)
    return ref.get()

if __name__ == '__main__':
    last_adc_ch0v = None
    last_tempC1 = None
    hum = 0.0

    while True:
        adc_ch0v = get_data('/bridgeTTN2FB/ADC_CH0V')
        tempC1 = get_data('/bridgeTTN2FB/TempC1')
        if adc_ch0v is not None and tempC1 is not None:
            if adc_ch0v != last_adc_ch0v or tempC1 != last_tempC1:
                print(f'ADC_CH0V: {adc_ch0v}')
                hum = 74.074074*adc_ch0v - 73.8518
                print(f'Humedad Aire: {hum}')
                update_graph(float(hum))
                last_adc_ch0v = adc_ch0v
                last_tempC1 = tempC1
            else:
                print('No data found for ADC_CH0V or TempC1.')
        time.sleep(1)

```

Anexo F

Script Python para Graficar el Canal ADCCH1V

Este anexo incluye el código fuente del script Python 'graphADCCH1V.py'. Este script se utiliza para la visualización en tiempo real de los datos recogidos del canal ADCCH1V, facilitando el seguimiento y análisis continuo de variables específicas importantes para el monitoreo del cultivo.

```
import firebase_admin
from firebase_admin import credentials, db
import matplotlib.pyplot as plt
import time

cred = credentials.Certificate(r'C:\Users\lozai\Documents\Tesis\puttycopy\loranetworktest-firebase-adminsdk-73a9a-7dac061796.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://loranetworktest-default-rtdb.firebaseio.com/'
})

print("Firebase App initialized")

plt.ion()
fig, ax = plt.subplots()
y_data = []
line, = ax.plot(y_data)
ax.set_ylim(0, 100)
ax.set_title('Gráfica de ADC_CH1V')
ax.set_xlabel('Número de Lecturas')
ax.set_ylabel('Valores de ADC_CH1V')

def update_graph(new_data):
    y_data.append(new_data)
    if len(y_data) > 100:
        y_data.pop(0)
    line.set_ydata(y_data)
    line.set_xdata(range(len(y_data)))
```

```

ax.relim()
ax.autoscale_view()
fig.canvas.draw()
fig.canvas.flush_events()

def get_data(path):
    ref = db.reference(path)
    return ref.get()

if __name__ == '__main__':
    last_adc_ch1v = None
    last_tempC1 = None
    hum = 0.0
    while True:
        adc_ch1v = get_data('/bridgeTTN2FB/ADC_CH1V')
        tempC1 = get_data('/bridgeTTN2FB/TempC1')
        if adc_ch1v is not None and tempC1 is not None:
            if adc_ch1v != last_adc_ch1v or tempC1 != last_tempC1:
                print(f'ADC_CH0V: {adc_ch1v}')
                hum = 66.9811*adc_ch1v - 133.2902
                print(f'Humedad del Suelo: {hum}')
                update_graph(float(hum))
                last_adc_ch1v = adc_ch1v
                last_tempC1 = tempC1
            else:
                print('No data found for ADC_CH0V or TempC1.')
        time.sleep(1)

```

Anexo G

Script MySQL para Crear la Tabla

Este anexo presenta el script SQL utilizado para crear la tabla en la base de datos MySQL, que se utiliza para almacenar los datos de humedad y temperatura recopilados por los dispositivos IoT. Este script se ejecuta dentro del sistema de gestión de bases de datos MySQL para preparar la base de datos para la recepción de datos en tiempo real.

```
CREATE TABLE Ins50 (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  device_id VARCHAR(255) NOT NULL,  
  application_id VARCHAR(255) NOT NULL,  
  received_at DATETIME NOT NULL,  
  humidity FLOAT NOT NULL,  
  temperature FLOAT NOT NULL  
);
```

Anexo H

Script MySQL Para Crear el Usuario

Este anexo incluye el código SQL utilizado para la creación y configuración de un usuario en la base de datos MySQL, necesario para gestionar las operaciones de la base de datos dentro del proyecto.

```
CREATE USER 'xibernetiq'@'localhost' IDENTIFIED BY 'Automatic@';  
UPDATE USER SET PLUGIN='mysql_native_password' WHERE USER='xibernetiq';  
FLUSH PRIVILEGES;
```