

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“ ARQUITECTURA BASADA EN LORA PARA
INTELIGENCIA AMBIENTAL EN RESERVAS NATURALES ”

SOFÍA GLORIA MAÑANA BAÑALES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Erika Rosas
Profesor Correferente: José Luis Martí

Agosto - 2023

DEDICATORIA

A mi madre

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi familia por haber estado en todo este proceso a mi lado apoyándome, en especial a Pato, por haber visto las clases conmigo, por todas las conversaciones de ingeniería y por haber estado igual de emocionado que yo con cada paso que daba, mil mil gracias y los quiero mucho a todos.

En segundo lugar, a mi profesora guía y amiga Erika, la confianza que fue depositada en mí no lo hace cualquiera y estaré eternamente agradecida. Realizar un trabajo así nunca es fácil, pero con tu guía y apoyo se hizo más ameno, muchas muchas gracias.

Tercero, al GRC, fue una experiencia inolvidable que me dejará marcada por el resto de mis días, lo aprendido, lo disfrutado y lo bailado no lo saca nadie. Muchísimas gracias por haberme aceptado y por haber hecho de este trabajo posible. Mención especial a Benjamín, por haberme dejado utilizar ALLoRa y haberse convertido en un gran amigo en el proceso, muchas gracias.

En cuarto lugar, a Alain, su apoyo incondicional durante estos cinco años ha sido fundamental para mí, desde las largas noches de estudio hasta las levantadas temprano a pruebas. Gracias desde el fondo de mi corazón, te amo.

Por último, a cada persona que influyó de alguna manera en llegar hasta este momento, mis amigos, compañeros, colegas y conocidos. Cada uno realizó su aporte y agradezco que hayan sido parte de mi experiencia universitaria. Mil gracias a todos.

RESUMEN

Resumen— Dentro del contexto de Inteligencia Ambiental, surgen distintos sistemas con el fin de monitorear a la naturaleza de manera remota. Se presentan distintas opciones en las que una mezcla de sensores y gateways colocados en zonas remotas, monitorean los aspectos principales y, mediante un análisis posterior de los datos, el ser humano es capaz de cuidar los ecosistema sin involucrarse. En este trabajo se presenta un arquitectura de Edge Computing basada en LoRa, la capa física de un protocolo de comunicación, utilizando dos redes LoRaWAN y AllLoRa. El sistema está diseñado para monitorear zonas remotas, por lo que el aspecto de cobertura y escalabilidad son claves a la hora de presentar el diseño. El sistema fue implementado usando un RAK7391, un gateway que permite hasta 16 canales de LoRaWAN y se le fue incluido una Heltec WiFi LoRa 32 para poder recibir la señal de AllLoRa. Se analizó el sistema en base a la cobertura dada, la posibilidad de escalar, la sostenibilidad y la posibilidad de integración con otras aplicaciones. Se llegó a la conclusión de que es un sistema que sirve cómo base para implementaciones futuras, la cobertura es lo suficiente cómo para cubrir una zona remota y la sostenibilidad es deseada, en cuanto a escalabilidad deja espacio para la mejora, por lo que se propone un análisis más profundo en trabajos futuros.

Palabras Clave— Inteligencia Ambiental, IoT, LoRaWAN, Edge Computing, AllLoRa

ABSTRACT

Abstract— Within the context of Ambient Intelligence, different systems emerge to remotely monitor nature. Various options are presented, involving a combination of sensors and gateways placed in remote areas, to monitor key aspects. Through subsequent data analysis, humans are capable of caring for ecosystems without direct involvement. This work presents an Edge Computing architecture based on LoRa, a physical layer of a communication protocol, utilizing two networks: LoRaWAN and AllLoRa. The system is designed for monitoring remote areas, where coverage and scalability are crucial aspects of the design. The implementation utilized a RAK7391 gateway, supporting up to 16 LoRaWAN channels, along with a Heltec WiFi LoRa 32 to receive the AllLoRa signal. The system was analyzed based on its coverage, scalability, sustainability, and potential integration with other applications. It was concluded that the system serves as a foundation for future implementations, providing sufficient coverage for remote areas and desirable sustainability. As for scalability, there is room for improvement, suggesting further in-depth analysis in future works.

Keywords— Ambient Intelligence, IoT, LoRaWAN, Edge Computing, AllLoRa

GLOSARIO

ADR: *Adaptative Data Rate.*

AWS: *Amazon Web Services.*

GPS: *Global Positioning System.*

IA: *Inteligencia Artificial.*

IoT: *Internet of Things.*

ISM: *Industrial, Scientific and Medical.*

LP-WAN: *Low Power Wide Area Network.*

LTE: *Long-Term Evolution.*

PDR: *Packet Drop Rate.*

SF: *Spreading Factor.*

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
CAPÍTULO 1: INTRODUCCIÓN	1
1.1 CONTEXTO	1
1.1.1 SITUACIÓN EN CHILE	2
1.1.2 ALTERNATIVAS DE SOLUCIÓN	3
1.2 OBJETIVOS	5
1.2.1 OBJETIVOS ESPECÍFICOS	5
1.3 IMPACTO DE SOLUCIONAR EL PROBLEMA	5
1.4 METODOLOGÍA DE INVESTIGACIÓN	6
1.5 ORGANIZACIÓN DEL DOCUMENTO	7
CAPÍTULO 2: MARCO CONCEPTUAL	8
2.1 EDGE COMPUTING	8
2.2 LoRa	9
2.3 LoRaWAN	10
2.3.1 COBERTURA	11
2.3.2 CREDENCIALES DE CONEXIÓN	12
2.3.3 SPREADING FACTOR (SF)	13
2.3.4 ADAPTIVE DATA RATE (ADR)	13
2.3.5 OVER-THE-AIR-ACTIVATION (OTAA) vs ACTIVATION BY PERSONALIZATION (ABP)	13
2.3.6 RECEIVED SIGNAL STRENGTH INDICADOR (RSSI)	14
2.3.7 CODING RATE (CR)	14
2.4 AllLoRa	15
2.5 CHIRPSTACK	16
2.6 THE THINGS STACK	17
2.7 DOCKER	18
2.8 TRABAJOS RELACIONADOS	19
CAPÍTULO 3: DISEÑO DEL SISTEMA	24
3.1 DESCRIPCIÓN	24
3.1.1 REQUISITOS FUNCIONALES	24
3.1.2 REQUISITOS NO FUNCIONALES	24

3.2	ARQUITECTURA	25
3.3	CONECTIVIDAD	28
3.3.1	NETWORK SERVER	29
3.4	SENSORES	30
3.4.1	ARDUINO PORTENTA H7 Y PORTENTA VISUAL SHIELD	31
3.4.2	ARDUINO NICLA VISION	32
3.4.3	PYCOM LOPY4	33
3.5	SERVIDOR EDGE	34
3.5.1	DLOS8N	35
3.5.2	RAK7249	36
3.5.3	RAK7391	37
CAPÍTULO 4: EVALUACIÓN DEL SISTEMA		40
4.1	IMPLEMENTACIÓN	40
4.1.1	HARDWARE	40
4.1.2	SOFTWARE	45
4.2	ANÁLISIS DE COSTOS	48
4.2.1	HARDWARE	48
4.2.2	MIGRACIÓN	49
4.2.3	COSTO DE MANTENCIÓN	50
4.2.4	RESUMEN	51
4.2.5	OTRAS OPCIONES	52
4.3	COBERTURA	53
4.4	ESCALABILIDAD	54
4.5	INTEGRACIONES	57
4.6	SOSTENIBILIDAD	58
CAPÍTULO 5: CONCLUSIONES		60
5.1	CONCLUSIONES GENERALES	60
5.1.1	RESULTADOS GENERALES	60
5.1.2	LIMITACIONES	61
5.1.3	PRINCIPALES DESAFÍOS	61
5.2	RESULTADOS PRINCIPALES	61
5.2.1	OBJETIVOS SECUNDARIOS	61
5.2.2	OBJETIVO PRINCIPAL	62
5.3	TRABAJOS FUTUROS	63
5.3.1	INTELIGENCIA ARTIFICIAL	63
5.3.2	AIoLoRa	63
5.3.3	ESCALABILIDAD	63
ANEXOS		64
REFERENCIAS BIBLIOGRÁFICAS		70

ÍNDICE DE FIGURAS

1	Señal de transmisión en el tiempo.	3
2	Ejemplo de red Edge.	9
3	Topología de LoRaWAN.	11
4	Capas de LoRAWAN.	11
5	Modulos de Protocolo de AllLoRa.	15
6	Aquitectura de ChirpStack.	17
7	Arquitectura de The Things Stack.	18
8	Ejemplo de Aplicaciones en Contenedor.	19
9	Arquitectura del sistema propuesto.	20
10	Arquitectura del sistema propuesto.	21
11	Arquitectura basada en red LP-WAN del sistema propuesto.	22
12	Ejemplo de arquitectura Edge para implementar en el sistema.	27
13	Distribución de Sensores de Arduino Portenta H7 y Portenta Visual Shield.	32
14	Distribución de Sensores de Arduino Nicla Vision.	33
15	Distribución de Sensores de LoPy4.	34
16	Gateway LoRaWAN DLOS8N.	36
17	Gateway RAK7249.	37
18	Estructura del Software.	38
19	Arquitectura interna de RAK7391.	39
20	Foto de RAK7391.	41
21	Foto de RAK7391 dentro del cierre exterior.	42
22	Foto de RAK5146 conectada a RAK7391 y foto de RAK5146.	42
23	Antena LoRaWAN.	43

24 Heltec WiFi Lora 32 (v2) y fotografía de la integración al gateway.	43
25 Gateway registrado en ChirpStack.	45
26 Mensaje de prueba recibido por el gateway.	46
27 Configuración de integración ChirpStack-InfluxDB.	47
28 Terminal indicando los contenedores iniciados al prender el sistema.	48
29 Mapa de distribución de RSSI en el Mar Menor, Murcia, España.	54
30 Puertos Seriales conectados en el gateway RAK7391.	65
31 Puertos Seriales conectados en Heltec WiFi LoRa 32.	66

ÍNDICE DE TABLAS

1 Sensores a utilizar.	44
2 Tabla resumen de precios.	49

CAPÍTULO 1

INTRODUCCIÓN

En el presente capítulo, se presenta el contexto en el cuál se realizará esta memoria, además de la situación en Chile, los objetivos a cumplir y el impacto de realizar un proyecto de este tipo.

1.1. CONTEXTO

El término *Inteligencia Ambiental* es usado para describir un ambiente adaptado a seres humanos, de manera que el ambiente sea vigilado mediante la extracción de datos [Stipanicev *et al.*, 2007]. En este sentido, el medio ambiente termina siendo monitoreado constantemente, lo que conlleva a una gran cantidad de datos que pueden ser analizados posteriormente. Con esto, el esfuerzo humano es enfocado en el análisis de estos y en cómo utilizarlos para prevenir posibles actos dañinos. Este concepto está siendo utilizado en la actualidad en varias organizaciones para ayudar a los distintos tipos de zonas naturales a mantener su estabilidad frente a un cambio climático inminente. Con las tecnologías emergentes de hoy en día, es posible realizar un monitoreo de zonas de difícil acceso, dónde no solía existir la posibilidad de alcanzar debido a la falta de cobertura de las redes disponibles. La evolución de las tecnologías de comunicación ha llegado a un punto en que la distancia o lo remoto del lugar ya no es un factor limitante cómo en tiempos anteriores.

Dentro de la Inteligencia Medioambiental, se presenta el desafío de cómo recolectar los datos, cómo se analizan y cómo realizar todo esto de la manera más eficiente posible, sin ser invasivos con el sector a estudiar. Es por esto que se introduce el concepto de IoT (*Internet of Things*). Es presentado cómo la tecnología clave a la hora de realizar el diseño de un sistema que permita aportar a la toma de decisiones en distintos niveles y facilita la interacción en tiempo real con el medio estudiado [Furstenau *et al.*, 2022]. IoT es un paradigma que permite la conexión entre distintos tipos de dispositivos o *smart devices*, es posible desplegar una red de dispositivos que midan y monitoreen una gran variedad de parámetros en un medio y conectar todos estos en un solo lugar [Kumar *et al.*, 2019]. Los dispositivos IoT, como sensores ambientales, cámaras, drones y otros dispositivos, pueden recopilar y transmitir datos sobre las condiciones ambientales, la flora y fauna, la calidad del aire y el agua, entre otros aspectos, permitiendo a los gestores de estas áreas tener un mayor conocimiento y control de los problemas ambientales. Este tipo de aplicaciones se han vuelto muy populares debido a la gran utilidad de los datos recolectados, se pueden realizar modelos de predicción o simulaciones de varias situaciones basados en datos reales, todo para lograr tomar decisiones en favor del medio estudiado. Al tener información en tiempo real, se logra saber el estado actual y no tomar las decisiones basados en datos desactualizados.

IoT trajo consigo nuevos desafíos y nuevos temas de estudio, sobre todo relacionado con

la comunicación entre los dispositivos y en la eficiencia del envío de información. Los problemas de estudio fueron apareciendo a medida que se implementaban en distintas áreas. Por ejemplo, en el caso de realizar una aplicación en el área de la salud, una de las principales preocupaciones es almacenamiento y visualización de datos [Alil y Alil, 2020]. En otras aplicaciones orientadas a analizar zonas naturales, la principal complicación es envío de los datos a zonas urbanas, en cómo se lograba mantener una conexión en lugares dónde no es posible tener una gran cantidad de servidores, manteniendo un margen de error aceptable. Sin embargo, la implementación de estos sistemas puede ser costosa y enfrentar desafíos como la falta de infraestructura y energía en áreas remotas, la necesidad de garantizar la privacidad y seguridad de los datos recopilados, la necesidad de involucrar a las comunidades locales en el proceso, entre otros. Esta memoria se enfocará en las redes de comunicación y sistemas relacionados para lograr resolver este problema.

Al tener una red desplegada, los sensores y dispositivos en el exterior de la red deben estar listos para medir los datos en cualquier momento, cómo consecuencia se tiene que los dispositivos se mantienen encendidos en todo momento, incluso cuando no es necesario, lo que levanta un nuevo desafío: la eficiencia de energía y cómo se reduce la contaminación teniendo una gran cantidad de dispositivos emitiendo señales día y noche. Se estima que para el año 2025 un total de 75 billones de dispositivos, por lo que se vuelve un tema importante de resolver [Bansal *et al.*, 2020]. Es por esto que surge una nueva línea de estudio, *Green IoT*. Busca realizar el mismo proceso de monitoreo y análisis que presenta IoT, pero teniendo en cuentas las emisiones de CO₂, la eficiencia energética y en cómo afecta a la comunidad.

La Unión Europea cuenta con distintas metas a cumplir dentro de los próximos años, especialmente en el área medioambiental. Una de estas medidas es el proyecto 2030, dónde se buscan reducir en un 55 % las emisiones de gases invernaderos para el año 2030 [Agency, 2023]. Es por esto que se están realizando incentivos a distintos proyectos de investigación que fomenten y aporten a la meta. Uno de estos es *CHAN-TWIN*, un proyecto realizado por la Universidad Politécnica de Valencia y la Universidad de Alicante que busca implementar un gemelo digital del parque natural "Lagunas de la Mata y Torrevieja", ubicado en Alicante, España. Un gemelo digital consiste en realizar réplicas digitales de las entidades presentes en áreas naturales, con esto se facilita el monitoreo de la zona y se logra comprender la dinámica de los elementos dentro del parque.

1.1.1. SITUACIÓN EN CHILE

Siendo un país largo y angosto, Chile cuenta con diversos sistemas ecológicos en la totalidad de su soberanía. Es reconocido por tener los climas más extremos, desde el extremo calor del desierto al extremo frío de la Antártica. Con el tiempo, los sistemas ecológicos han evolucionado, adaptándose al turismo y a las consecuencias del cambio climático. Esta memoria se enfocará en los 42 parques nacionales, 46 reservas nacionales y 18 monumentos naturales a lo largo de todo el país [Forestal, 2023]. Lugares dónde el turismo ha ido en aumento este último tiempo y las consecuencias de este han dejado su rastro. Desde incendios forestales,

contaminación y el alejamiento de las criaturas nativas de la zona.

Aparte de una gran cantidad de reservas naturales, Chile cuenta con gran territorio agrícola, siendo este una parte importante de la economía del país. En esta área, se han realizados diversos estudios de cómo implementar tecnologías IoT en dicha área y con esto lograr aumentar la eficacia de las cosechas. Dentro de los trabajos realizados destaca [Ahmed *et al.*, 2022], un trabajo que realizó una plataforma para el monitoreo de agricultura de manera remota utilizando LoRa como protocolo de comunicación. Se realizó un monitoreo de la temperatura y el estado de la tierra en cultivos de palta y arándanos.

1.1.2. ALTERNATIVAS DE SOLUCIÓN

Proyectos de este tipo necesitan tener varios aspectos en cuenta a la hora de presentar una solución. Se puede abordar el tema desde distintos puntos de vista y hay que tener en consideración todos los posibles aspectos que puedan afectar a la solución final, tales como la red que conectará los dispositivos, la arquitectura y la estructura del traspaso de datos. En primer lugar se analizará el tipo de red y los posibles problemas que traigan consigo. En el caso de proponer un sistema que implemente un monitoreo utilizando sensores conectados por WiFi, se podría lograr obtener los datos buscados, pero no se estaría tomando en cuenta la cobertura de este, además del gasto energético asociado a utilizar dicha conexión. En [Wang *et al.*, 2020] se analizó cómo se comportaba una red WiFi a medida que se agregaban dispositivos mandando datos a una misma frecuencia, en la figura 1 se muestra cómo va disminuyendo la intensidad de la señal a medida que se aumenta la cantidad de dispositivos conectados.

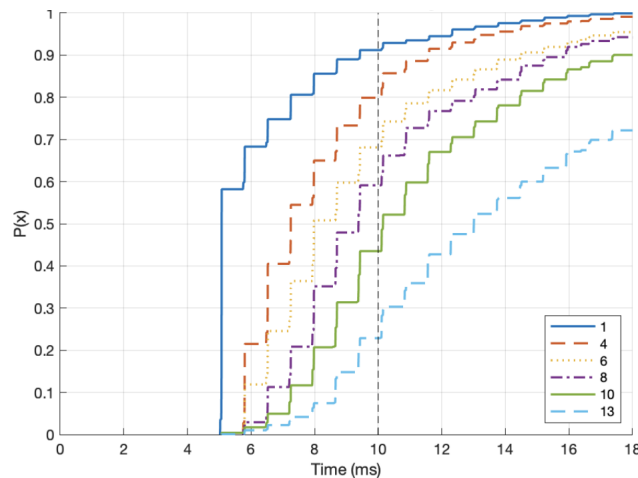


Figura 1: Señal de transmisión en el tiempo.

Fuente: [Wang *et al.*, 2020].

Por lo general, las reservas naturales cuentan con grandes áreas, por lo que si se quiere

realizar un sistema de monitoreo que cubra la mayor parte de dicha superficie, la cobertura de la red a utilizar es un tema importante de analizar a la hora de tomar decisiones de diseño. Las redes *Wide Area Network* aparecen cómo la opción a seguir al querer tratar de cubrir una mayor superficie. Cómo se mencionó anteriormente, al implementar un despliegue de dispositivos, se debe tener en cuenta las emisiones que estos producen y el gasto energético que conlleva tener una gran cantidad de dispositivos encendidos en un periodo largo de tiempo. Es por esto que las redes LPWAN (*Low Power Wide Area Network*) aparecen cómo una solución a estos problemas. Son redes de bajo consumo energético, amplia cobertura y de bajo costo comunicacional. Provee entre 10 y 40 km de cobertura en áreas rurales y entre 1 y 5 km en zonas urbanas [Mekki *et al.*, 2019].

En [Klimiashvili *et al.*, 2020] se compara a la red WiFi con LoRaWAN, una de las más conocidas redes LPWAN. Tomando en cuenta a distintos aspectos, tales como la intensidad de la señal, la capacidad de enviar paquetes de diversos tamaños y la cobertura de ambas señales en distintos tipos de áreas (urbanas y rurales). Se llegó a la conclusión que al enviar paquetes de gran tamaño, WiFi es la opción que tiene menos fallas, pero en distancias pequeñas. LoRaWAN demostró ser más eficiente en distancias largas pero mandando paquetes de menor tamaño. Para realizar un sistema de monitoreo de una reserva natural se requiere tener varios tipos de sensores, que midan temperatura, humedad, luminosidad, entre otros. Los datos recolectados no suelen ser de gran tamaño, sobre todo considerando que van a estar constantemente enviando pequeñas señales, por lo que se debería priorizar una gran cobertura de superficie sobre una señal que permita enviar un mayor tamaño de paquetes, además de priorizar, la amplia cobertura de las redes LPWAN.

Cómo posible alternativa, se encuentran las redes celulares, tales como 3G, 4G y 5G. Estas cuentan con la ventaja de tener una gran cobertura, pero no alcanzan a tener una eficiencia energética. Las baterías de los dispositivos no llegan a tener 10 años de batería continua cómo lo harían algunos dispositivos conectados a redes LPWAN [Raza *et al.*, 2017]. La red 5G ha logrado diversos avances comparados con la red 4G/LTE, debido al nuevo estilo de arquitectura desarrollado logra tener una velocidad de transmisión mayor en un sistema más denso (con más dispositivos conectados), esta puede llegar a ser de hasta 10 Gbps (*Gigabytes per second*) [Ogbodo *et al.*, 2022]. Aún así, el problema de la eficiencia energética no logra ser superado, la batería de los dispositivos que soportan esta red tiende a agotarse rápidamente, por lo que para aplicaciones de IoT en zonas remotas no podría ser una opción viable.

Otra posible alternativa de solución es un sistema basado IoT basado en Cloud. En el contexto de este trabajo, un sistema Cloud-IoT se refiere a una capa que contenga distintos tipos de sensores y dispositivos monitoreando algún aspecto del ambiente a estudiar, estos envían los datos recolectados a una nube que almacena estos datos, para que luego aplicaciones puedan utilizarlos para realizar las visualizaciones y análisis correspondientes. Un sistema de este tipo aporta grandes ventajas, tales como la accesibilidad a los datos, al estar en la nube, se pueden acceder desde cualquier lado, o también la flexibilidad de usar distintas herramientas para lograr el mismo sistema base. A pesar de esto, cuenta con varios problemas, sobre todo relacionados con la latencia del sistema [Saddeeq *et al.*, 2021]. Si todos los

dispositivos se encuentran conectados a solo una nube, se genera un único punto de falla que puede perjudicar al sistema completo en caso de algún error. Esto es conocido como sistemas centralizados y, a pesar de tener todos los datos en un mismo lado (lo cuál ayuda a las aplicaciones a recolectar los datos para su uso), esto termina siendo una vulnerabilidad para el sistema completo.

Por otro lado, se encuentra el problema de dónde realizar el procesamiento de los datos. Si se utiliza un sistema Cloud centralizado, este procesamiento debería realizarse en el mismo lugar, por lo que se aumentaría la latencia del sistema, además de continuar teniendo un único punto de falla. Es por esto que el concepto de *Edge Computing* aparece como una posible solución a este problema. Este paradigma lleva el procesamiento de los datos a los bordes de la red, a los dispositivos. Utilizando herramientas como TinyML, ayudan al procesamiento de los datos dentro de los mismos sensores y así, se envían los datos ya procesados en algún sentido y así se alivia la latencia del sistema.

1.2. OBJETIVOS

El objetivo general de esta memoria consiste en diseñar, implementar y analizar un sistema IoT sostenible para monitorear reservas naturales.

1.2.1. OBJETIVOS ESPECÍFICOS

Para lograr el objetivo general planteado, se deben cumplir los siguientes objetivos específicos:

1. Investigar el estado del arte de los distintos sistemas disponibles para el monitoreo de áreas verdes.
2. Diseñar una arquitectura Edge-LoRa junto con IoT para el monitoreo de distintos tipos de parámetros dentro de reservas naturales.
3. Implementar la arquitectura de manera modular para habilitar la comunicación y procesamiento de datos.
4. Analizar eficacia en el sistema usando dos o tres sensores conectados al gateway con LoRa.

1.3. IMPACTO DE SOLUCIONAR EL PROBLEMA

Las reservas naturales a lo largo del mundo continúan siendo un misterio para el ser humano, cómo interactúan las especies, cómo se autoregula y cómo afecta algún cambio en el largo

plazo. Un sistema de monitoreo IoT permitiría la recolección continua y en tiempo real de datos ambientales y biológicos, tales como la calidad del aire, la humedad, la temperatura, el nivel de agua y la presencia de especies animales y vegetales en la reserva. Datos como estos permiten a científicos y administradores de áreas verdes entender mejor el estado de la vegetación y el ecosistema. La gestión de distintos factores puede ser mejorada al tener más datos, especialmente datos en tiempo real ya que se sabe el estado actual de las reservas. Distintas decisiones pueden ser tomadas en caso de alguna urgencia y ayuda a prevenir catástrofes a largo plazo. Tener datos de cómo se está comportando un ambiente, además de generar un dataset donde se pueda ver la evolución en el tiempo de las áreas verdes, puede ayudar a identificar problemas como la degradación del suelo, la pérdida de biodiversidad o el estado de reservas de agua, antes de que se conviertan en problemas graves y sin vuelta atrás.

Además, otro impacto importante de mencionar, sería la promoción de tecnologías enfocadas en la sostenibilidad y conservación del medio ambiente. Proyectos de este tipo pueden ser parte de una nueva línea de investigación, donde se trata de realizar tecnologías más conscientes con el medio ambiente y buscar aportar al análisis y cuidado del planeta, tratando de frenar las consecuencias de las actividades del ser humano. Por otro lado, el monitoreo de áreas verdes podrían cuantificar el efecto de personas en dichas zonas, si bien es de conocimiento general que el ser humano deja su huella en el medio ambiente, se podría analizar las posibles consecuencias de continuar con las actividades regulares de turistas en zonas y tomar medidas más drásticas.

Un sistema IoT también podría ser utilizado con fines educativos, permitiendo a los visitantes de las reservas aprender sobre la importancia de la conservación del medio ambiente y cómo pueden contribuir a ella. Educar a las personas sobre las posibles consecuencias de la contaminación de estas áreas, puede ser un impacto paralelo al principal de esta memoria. Monitorear las zonas dice el estado actual de las reservas, pero si no se acompaña con una reeducación de la población, no se podría alcanzar el máximo potencial de este tipo de proyectos.

Por último, el desarrollo de un sistema sostenible IoT para el monitoreo de áreas verdes remotas, lograría implementar tecnologías actuales en lugares no usuales, lo que llevaría al descubrimiento de nuevos desafíos en el área de investigación. Aplicar distintas técnicas en ambientes distintos al que fueron desarrollado ayuda a generar tecnologías más íntegras, logra que sean aplicables en ambientes que no se ha tratado antes y así, progresar con el desarrollo tecnológico.

1.4. METODOLOGÍA DE INVESTIGACIÓN

Al tratarse de una propuesta de sistema, se necesita seguir una estructura de trabajo que permita tener el resultado esperado de esta memoria. Es por esto que se implementará la siguiente metodología:

- **Identificación de requisitos:** se identificarán los requisitos del sistema, esto implica analizar el ambiente y conocer las necesidades de este, además de que aspectos de estos es vital monitorear. Se recopilará información clave para un análisis posterior.
- **Análisis del sistema:** se realizará un análisis detallado del sistema a estudiar utilizando la información recopilada del punto anterior. Se analizará la relación entre las distintas entidades presentes en el ambiente, se determinará un orden de prioridad para que aspectos es de vital importancia analizar y por último, se realizará un estudio de trabajos previos en el área, considerando las distintas tecnologías y herramientas emergentes.
- **Diseño del sistema:** Con base en el análisis, se realizará un diseño del sistema de monitoreo, esto implicará crear un diseño arquitectónico utilizando las tecnologías más adecuadas para lograr el objetivo planteado. Además, se definirán los requisitos tanto de software como de hardware necesarios.
- **Implementación del sistema:** se realizará la implementación del sistema, configurando los distintos componentes para así lograr una comunicación eficiente y segura. Luego, se realizarán las pruebas necesarias para comprobar la eficacia del sistema.
- **Pruebas y validación:** se probará el sistema implementado con distintos tipos de sensores y se validará que el sistema cumple con los requisitos planteados. Si las pruebas realizadas revelan información que altere el diseño del sistema, se volverá a dicho punto y se tomarán nuevas decisiones de diseño.

1.5. ORGANIZACIÓN DEL DOCUMENTO

El presente documento desarrolla los temas pertinentes para conseguir los objetivos planteados en este capítulo. Se presentarán en distintos capítulos, cada uno enfocado a un tema principal. El Capítulo 2 presenta el marco conceptual en el que se desarrolla esta memoria, se presentan los conceptos base para la comprensión de este documento. Se incluyen los temas de mayor relevancia, la descripción de las tecnologías utilizadas en la propuesta de solución y trabajos relacionados. En el Capítulo 3, se propondrá un diseño de sistema considerando los requisitos necesarios y especificando el hardware utilizado para la implementación. En el Capítulo 4 se describirá la implementación y evaluación del sistema presentado, analizando la cobertura, escalabilidad y los costos asociados. Por último, en el Capítulo 5 se presentarán las conclusiones de esta memoria.

CAPÍTULO 2

MARCO CONCEPTUAL

En el presente capítulo, se desarrollarán los conceptos fundamentales requeridos para el desarrollo de este trabajo. Se definen los conceptos de *Edge Computing*, *LoRa* y *LoRaWAN*, además del framework utilizado para la implementación de la solución propuesta. Existen diversos textos que discuten la evolución de los distintos componentes mencionados anteriormente. Se realizará un análisis por cada uno de estos y al final, se mencionará al texto que une todos estos enfocados en la temática de la memoria.

2.1. EDGE COMPUTING

Tal como se menciona anteriormente, uno de los desafíos que genera la utilización de IoT es el diseño de la red detrás, el cómo se conectan entre sí los dispositivos y cómo se mandan los datos para iniciar el procesamiento de estos. En primer lugar, el *edge* de la red está compuesto por los dispositivos que se encuentran en el borde de la red, esto pueden ser celulares, computadores, tablets, etc [Kurose y Ross, 2016]. Para efectos de este documento, los sensores representarán el borde de la red propuesta.

Existen varias opciones a la hora de decidir cómo será el procesamiento de los datos recolectados por los sensores. Estos pueden ser procesados en un servidor ubicado en el centro de la red, pero esto significaría recolectar los datos y enviarlos constantemente, lo que aumentaría la latencia del sistema completo, además de sobrecargar el servidor en caso de que se tengan muchos dispositivos en el borde de la red. En otras palabras, se limitaría la capacidad de escalar el sistema. Es por esto que se decide por Edge Computing, un paradigma que utiliza los nodos en el borde de la red (*end devices*) no solo para la recolección de datos, si no que para realizar un pre-procesamiento cerca del lugar dónde se generan, así se logra alivianar la carga de todo el sistema, aprovechando al máximo las capacidades de los componentes [Taehong *et al.*, 2021]. Además, logra dar un nuevo ángulo de estudio para aplicaciones que son sensibles al tiempo y necesitan tener un análisis en tiempo real.

La idea detrás del Edge Computing es reducir la latencia y mejorar la eficiencia en la transferencia de datos, lo que es especialmente importante para aplicaciones que requieren una respuesta rápida y en tiempo real, como la realidad aumentada, la realidad virtual, el control de procesos industriales y la automatización de fábricas.

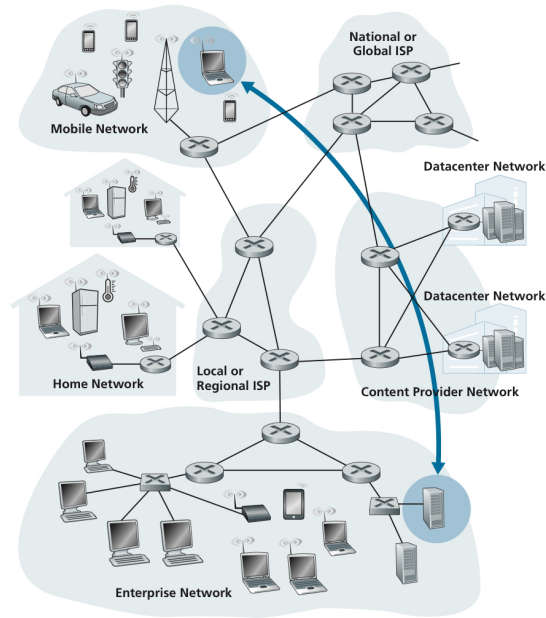


Figura 2: Ejemplo de red Edge.
Fuente: [Kurose y Ross, 2016].

2.2. LoRa

LoRa (*Long Range*) es una tecnología de comunicación inalámbrica de largo alcance, fue diseñada pensando en aplicaciones de IoT. Utiliza una modulación de espectro ampliado, basado en Chirp Spread Spectrum (CSS) en su capa física, esto permite que se logre transmitir datos en largas distancias con una baja tasa de error [Bor *et al.*, 2016].

Una de las principales ventajas de LoRa es que consume muy poca energía, lo que permite que los dispositivos funcionen durante largos períodos de tiempo con una sola batería. Esto la hace ideal para aplicaciones como medición de temperatura, monitoreo de humedad, seguimiento de sectores agrícolas y otros usos de IoT en los que los dispositivos necesitan estar en operación por mucho tiempo sin acceso a una fuente de energía externa.

LoRa utiliza rangos de frecuencias en el espectro de frecuencias no licenciadas, ISM (Industrial, Scientific and Medical). Estas frecuencias son de libre acceso, por lo que LoRa define un rango para así saltar a la que esté más libre al momento de realizar el traspaso de información. Cada uno de los rangos son definidos por región del mundo, por ejemplo Europa tiene definido el rango 863-870 MHz.

La capa física de LoRa utiliza Chirp Spread Spectrum (CSS), una técnica de modulación en la que se utiliza un conjunto de señales chirp para transmitir información. Si bien los detalles no han sido revelados por Semtech (la compañía que tiene la patente de LoRa en estos mo-

mentos), en [Vangelista, 2017] se ha descrito a la modulación de LoRa como Frequency Shift Chirp Modulation (FSCM), dónde existe un componente (Frequency Shift Key) que define el cambio de frecuencia, afectando a la forma de la onda del chirp.

2.3. LoRaWAN

Cómo los sistemas IoT necesitan una gran cantidad de operaciones de bajo costo, con las redes tradicionales esto no logra ser posible debido a la sobrecarga que generaría tener millones de dispositivos conectados a una misma red. Es por esto que una de las soluciones a este problema son las redes LPWAN (*Low Power Wide Area Network*). LoRaWAN es una de las tecnologías que ha surgido en este último tiempo y que ha sido aceptada por la comunidad como una de las que tiene más potencial para solucionar los problemas relacionados con la mantenibilidad de aplicaciones IoT [Haxhibeqiri *et al.*, 2018].

LoRaWAN es una especificación de red basada en LoRa. Utilizando las frecuencias máximas y mínimas de red, logran distribuir las señales en todo el ancho de banda definido según la región en que se encuentre, logrando así una comunicación eficiente [Thomas y Eldhose, 2020]. Es considerada cómo una red oportunista, debido a que va saltando dentro del espectro de frecuencia no licenciada. Esto contribuye a su eficiencia a la hora de manejar grandes cantidades de llamadas a la red.

En cuanto a su arquitectura, LoRaWAN es una red de tipo estrella, lo que implica que cuenta con una capa *edge* donde los dispositivos IoT representan los bordes de la red y se conectan a un gateway, el cual reenvía la información a un servidor central de red (*network server*). Los *gateways* de LoRaWAN, son servidores que procesan los datos recolectados y usualmente, son mostrados en algún servidor web cómo The Things Network o ChirpStack. Para efectos de esta memoria, se utilizará ChirpStack. Es importante mencionar que los *end devices* no pueden comunicarse directamente entre sí, por lo que se utilizan diferentes tipos de sensores para recopilar la información y enviarla al gateway mediante paquetes encapsulados en protocolos como UDP (*User Data Protocol*) o IP *Internet Protocol*) [Haxhibeqiri *et al.*, 2018]. En la figura 3 se puede observar la topología de la red con más detalle.

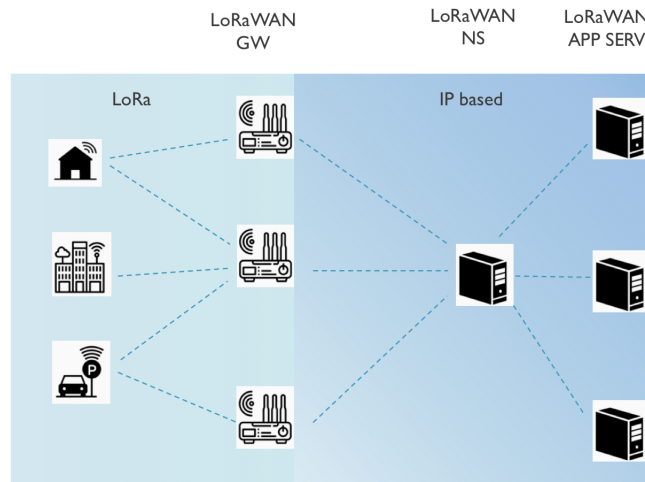


Figura 3: Topología de LoRaWAN.
Fuente: [Haxhibeqiri *et al.*, 2018].

Dentro de LoRaWAN se clasifican distintos tipos de dispositivos dependiendo de su *uplink* y *downlink*. Los dispositivos de Clase A cuentan con transmisión bi-direccional que son alocados de manera aleatoria. La comunicación es basada en ALOHA, un protocolo que permite el acceso múltiple a un mismo canal de broadcast [Kurose y Ross, 2016].

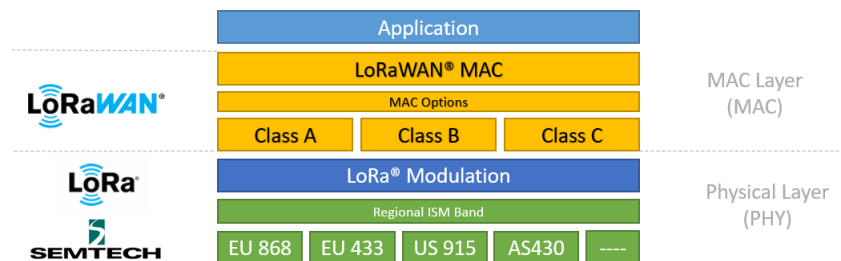


Figura 4: Capas de LoRAWAN.
Fuente: [Semtech, 2023]

2.3.1. COBERTURA

LoRaWAN es una red LPWAN, por lo que ofrece una amplia cobertura, entre 1 y 5 km en zonas urbanas y entre 10 y 40 km en áreas rurales [Mekki *et al.*, 2019]. Esto se debe principalmente a la técnica de modulación utilizada, FHSS (*Frequency Hopping Spread Spectrum*). Esta técnica utiliza el siguiente proceso, en primer lugar se define una lista de frecuencias (*spreading code*), esta lista va a ser el ancho de banda dividida en distintas partes, cada una de estas va a ser usada como posibles "saltos" que se pueden realizar. Luego, se define un

tiempo máximo en el que se estará en alguna de las frecuencias, si el mensaje no es enviado, se saltará a otra frecuencia. Este proceso se continúa hasta que se envía el mensaje por completo. Gracias a esto, la transmisión no es afectada por el ruido, las señales de estaciones de radio o otros factores ambientales. Además, la señal no es afectada en caso de conectar muchos dispositivos a la misma red, lo que hace de LoRaWAN un red ideal para sistemas IoT [Schwartz, 2001].

Otro de los factores que logran una gran cobertura, es la capacidad de los *end devices* de conectarse a múltiples gateways, luego cada uno de estos puede retransmitir los datos recolectados por los dispositivos, aumentando así la cobertura de la red. La topología de red continúa siendo estrella, pero de varias capas, así se aumenta el rango de transmisión y, gracias a FHSS, las interferencias no afectan a la transmisión en largas distancias.

En la figura 4 se pueden observar las bandas de frecuencias para cada una de las zonas del mundo. Por lo general son definidas para cada continente, pero al ser bandas de frecuencia no licenciadas, se define un rango para cada región en el que se puede ir "saltando" hasta finalizar la transmisión del mensaje a enviar. Esta frecuencia debe estar definida en el dispositivo y en el gateway, para así lograr una transmisión exitosa [Thomas y Eldhose, 2020].

2.3.2. CREDENCIALES DE CONEXIÓN

Para generar una conexión entre dispositivo y gateway, se necesitan una serie de credenciales que son enviadas en el inicio del mensaje a transmitir.

- **DevEUI (Device Unique Hardware ID):** es un identificador de 64 bits asociado a cada hardware. Todos los *end devices* traen asociado un identificador de fábrica.
- **AppEUI (Application Extended Unique Identifier):** es el identificador de 64 bits de la aplicación a la que se asociará el dispositivo. Las aplicaciones son registradas en los servidores web para así, mantener un orden en los mensajes recibidos por el gateway.
- **Gateway ID:** es el identificador del gateway utilizado para registrar este en el servidor web que procesará los datos recibidos.
- **DevAddr (Device Address):** es un identificador de 32 bits dado por el gateway que representa al dispositivo en la red. A diferencia de DevEUI (que es un identificador global del dispositivo), DevAddr es un identificador que se le asigna luego de que este se conecte a la red.

Cada uno de estos debe ser configurado manualmente en el código de transmisión y un error en los identificadores puede llevar a que el mensaje no se transmita correctamente [Thomas y Eldhose, 2020] [Industries, 2023].

2.3.3. SPREADING FACTOR (SF)

Se refiere a la cantidad de chips utilizados para transmitir un símbolo (señal) en el aire, en otras palabras, es el número de bits que se pueden encriptar a un símbolo en específico. Este número, puede intensificar o disminuir la señal de la antena, al usar menos chips para la transmisión, la señal es más débil, al usar más chips, la señal se intensifica. En LoRaWAN los valores de SF se mueven entre 7 y 12, un número mayor indica una señal más fuerte, lo cuál es usualmente para aumentar el alcance (cobertura) de la señal. Un SF bajo disminuye la velocidad de la transmisión (menos bits se transmiten) pero si se encuentra el dispositivo cerca del gateway, el mensaje debería transmitirse en la misma cantidad de tiempo. También, un SF bajo se utiliza para reducir la latencia del sistema pero a costa de una menor tolerancia al ruido y a la interferencia [Kufakunesu *et al.*, 2020].

2.3.4. ADAPTIVE DATA RATE (ADR)

Es una técnica que adapta la tasa de transmisión de los datos, optimizando la velocidad del envío de mensajes según la situación en la que se encuentre el *end device*. Con esto, se busca minimizar el uso de la batería de los dispositivos, se utilizan las configuraciones de transmisión óptimas en cada situación, con esto se evita el gasto energético innecesario y se aumenta el rendimiento del sistema. Por ejemplo, si un dispositivo se encuentra cerca del gateway, no es necesario utilizar un SF alto, ADR adaptaría esto y bajaría el SF para utilizar menos energía en transmitir el mensaje.

Las ventajas de utilizar este algoritmo son varias, principalmente enfocadas en el uso energético, pero también aumentan la eficiencia de la red. Al enviar paquetes en distintos SF, se optimiza el uso de los canales de comunicación y se aprovecha todo el ancho de banda. Un gateway de LoRa puede escuchar mensajes en distintos canales y en distintos SF, un *end device* puede mandar mensajes en solo un canal y SF, es por esto que si se encuentra habilitado ADR, el dispositivo es el responsable de ir adaptando el cómo transmite el paquete [Kufakunesu *et al.*, 2020].

2.3.5. OVER-THE-AIR-ACTIVATION (OTAA) vs ACTIVATION BY PERSONALIZATION (ABP)

Para que un dispositivo pueda comunicarse con el gateway de LoRaWAN, debe suceder un intercambio de mensajes en el que se solicita un permiso de conexión. Existen dos maneras de realizar esto, por Activation by Personalization (ABP) o por Over-The-Air-Activation (OTAA).

ABP es una manera de conectar el dispositivo pero necesita tener ciertos parámetros pre-configurados, estos parámetros se guardan en el hardware y se conecta directamente al gateway, lo que acorta el tiempo de conexión, pero se vuelve muy inseguro debido a que se

utilizan los mismos parámetros todas las veces que se conecte, por lo que se vuelve una inseguridad para el sistema [Industries, 2023].

Utilizando OTAA, el dispositivo debe mandar un mensaje de solicitud de conexión al gateway y este responde con las credenciales necesarias. Estas credenciales van cambiando con las conexiones, lo que vuelve más segura la conexión. Al realizar un intercambio de mensajes extras, este método de conexión tiende a tener un delay un poco mayor que ABP, pero una vez realizado el 'hand shake' el canal de conexión queda abierto, no es necesario realizar este intercambio de mensajes cada vez que se requiera transmitir algún dato [Na et al., 2017].

2.3.6. RECEIVED SIGNAL STRENGTH INDICADOR (RSSI)

Es un indicador de la intensidad de la señal recibida por el gateway, con esto intensidad recibida se puede calcular la distancia Euclidiana a la cuál se encuentra el dispositivo [Anjum et al., 2019]. Se mide en decibelios (dBm), mientras más grande el número medido, más cerca se encuentra el dispositivo. Este indicador sirve para ver la intensidad de la señal, pero no de cómo es la calidad de la señal ni la pérdida de paquetes asociada con el envío de los mensajes.

Es importante tener en cuenta que el RSSI no es una medida directa de la calidad de la señal, ya que puede estar influenciado por factores como la distancia, la interferencia y la obstrucción de la señal. Además, el RSSI no proporciona información sobre la tasa de error de bit (BER) de la transmisión, que es una medida más precisa de la calidad de la señal.

2.3.7. CODING RATE (CR)

Coding Rate (CR) es una técnica de codificación de canal utilizada para mejorar la resistencia a errores en las comunicaciones inalámbricas. Se expresa mediante una notación como CR4/5, donde el primer número representa la cantidad de bits de información codificados y el segundo indica la cantidad total de bits transmitidos. En la ecuación a continuación se muestra cómo es calculado, además de que n puede tomar los valores de 5,6,7 o 8 [Bouguera et al., 2018].

$$CR = \frac{4}{4 + n}, n$$

Este método es utilizado para prevenir errores en la transmisión de datos utilizando bits extras para mandar la información repetida en caso de pérdidas en el camino. Un CR4/5 ofrece una mayor resistencia a errores, pero esto implica un mayor tiempo para transmitir la misma cantidad de información. Mientras más pequeño el CR, más alto será el tiempo en el aire para transmitir los datos. Se deben considerar distintos factores relacionados al tipo

de sistema que se desea implementar, tales cómo las condiciones del canal, la distancia a la cuál se desea transmitir o el posible nivel de interferencia.

2.4. Allora

Allora es un protocolo de transferencia de mensajes capaz de realizar una malla de llamados entre dispositivos basados en LoRa [Arratia *et al.*, 2023a]. Es protocolo de half-duplex basado en un esquema stop-and-wait ARQ, significa que utiliza un método de comunicación en el que se transmite en una dirección a la vez, utilizando un mecanismo de confirmación para garantizar la entrega precisa de los datos antes de permitir la comunicación en la dirección opuesta. Uno o varios nodos de envío esperan por una solicitud de un nodo receptor o un gateway y responde con los datos pedidos. Ofrece *Mesh Mode*, lo cuál permite extender el alcance de la comunicación, utiliza múltiples nodos para reenviar paquetes hasta que llegan a su destino final.

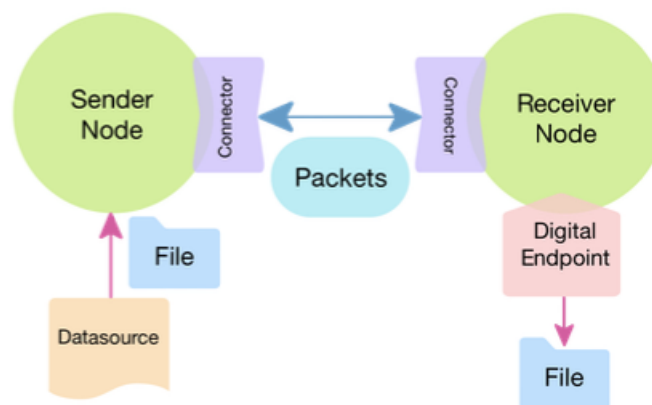


Figura 5: Módulos de Protocolo de Allora.
Fuente: [Arratia *et al.*, 2023a].

En la Figura 5 se pueden ver los distintos componentes que forman parte del protocolo, a continuación se explican en detalle cada uno de ellos.

- **Nodo:** es el encargado de manejar la comunicación con el Protocolo de Transferencia de Contenido, puede ser un Sender, Receiver o un Gateway, espera por solicitudes para enviar pedazos (chunks) de mensajes hasta que este termina. El Receiver pide información al Sender y espera a la respuesta. El Gateway actúa cómo similar a un Reciver pero es capaz de manejar múltiples Endpoints.
- **Conector:** es el componente que da acceso a los canales de LoRa a los Nodos. Se encarga de disponibilizar Allora a la mayor cantidad de dispositivos posibles, sobretodo

a dispositivos que no tienen LoRa integrado. Actúa como en puente entre dispositivos, logrando integrar LoRa a dispositivos que no lo incluyen de fábrica.

- **Digital Endpoint:** es utilizado por el Receiver o el Gateway para comunicarse con el Sender, el objeto contiene la dirección MAC del endpoint a comunicar y así lograr el estado de transferencia.
- **Data Source:** es el componente en el cuál se originan los datos a enviar, los provee para que el nodo Sender los envíe al nodo Receiver.
- **Archivo Allora:** maneja el contenido a enviar o recibir por los nodos, puede ser inicializado como un conjunto de bytes para luego ser dividido y enviado en pedazos hacia el nodo Receiver para luego ser vueltos a unificar y así tener todo el contenido.

2.5. CHIRPSTACK

ChirpStack es una plataforma de código abierto diseñada para permitir la implementación y administración de redes LoRaWAN de manera fácil y eficiente. La plataforma está compuesta por varios componentes que trabajan juntos para proporcionar una solución completa para la implementación y gestión de redes LoRaWAN [ChirpStack, 2023].

ChirpStack cuenta con un Network Server, es el responsable de recibir y procesar los mensajes que se envían desde los dispositivos LoRaWAN. El Network Server también gestiona la configuración y autenticación de los dispositivos, la gestión de la sesión y el control de acceso a la red. Además, es el encargado de la asignación de direcciones IP a distintos dispositivos y de la configuración de los canales de comunicación en los componentes de ChirpStack. Básicamente, es el encargado de realizar todas las gestiones de red relacionadas con el intercambio de mensajes dentro de la plataforma.

Luego, se encuentra el Application Server, que es responsable de procesar los mensajes de los dispositivos y entregarlos a las aplicaciones correspondientes. El Application Server también gestiona la integración con servicios en la nube y otros sistemas de backend. ChirpStack da la posibilidad de integrarse con distintos tipos de servicios, tales como AWS, Azure Service, RabbitMQ, InfluxDB, entre otros. En general, el Application Server es responsable de hacer que los datos de los dispositivos LoRaWAN sean útiles para las aplicaciones.

Por último, cuenta con un Gateway Bridge, que es responsable de recibir los paquetes de radiofrecuencia (RF) que son enviados por los dispositivos LoRaWAN y reenviarlos al Network Server. El Gateway Bridge también se encarga de traducir los diferentes formatos de datos que se utilizan en las diferentes marcas de gateways.

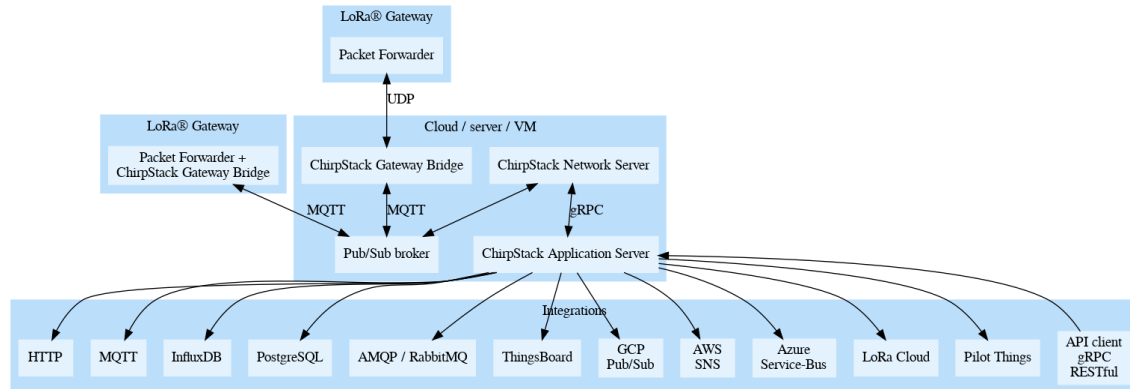


Figura 6: Aquitectura de ChirpStack.

Fuente: [ChirpStack, 2023].

2.6. THE THINGS STACK

The Things Stack, anteriormente conocido como The Things Network, es un Network Server de LoRaWAN de código abierto y altamente escalable. Permite implementar y gestionar redes LoRaWAN, permitiendo que los dispositivos se conecten y se comuniquen de manera segura entre sí y con los servidores de aplicaciones. Proporciona una infraestructura completa para construir y operar redes LoRaWAN. Incluye componentes como gateways, servidores de red, servidores de unión y servidores de aplicaciones. Estos componentes trabajan en conjunto para facilitar el registro de dispositivos, el enrutamiento de mensajes, el cifrado de datos y la gestión de dispositivos dentro del ecosistema LoRaWAN [Stack, 2020].

La plataforma ofrece diversas características y herramientas para desarrolladores, como una interfaz de programación de aplicaciones (API) para integrarse con otros sistemas, una consola Web para gestionar dispositivos y aplicaciones, y soporte para integraciones con otras aplicaciones, tales como bases de datos o herramientas para graficar resultados.

Al utilizar The Things Stack, individuos y organizaciones pueden configurar sus propias redes LoRaWAN privadas o conectarse a redes públicas, lo que permite la implementación de soluciones IoT en diversos ámbitos como ciudades inteligentes, agricultura, seguimiento de activos y monitoreo ambiental, entre otros.

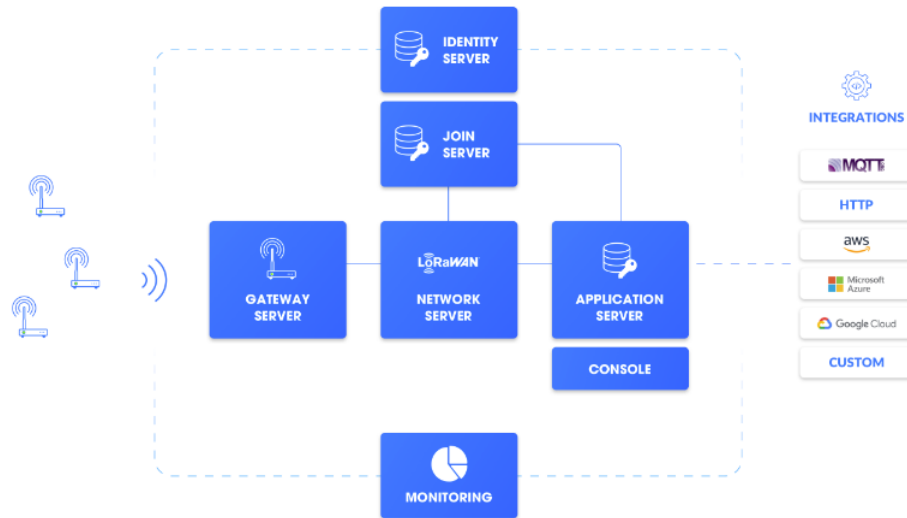


Figura 7: Arquitectura de The Things Stack.

Fuente: [Industries, 2022].

En la Figura 7 se pueden ver los distintos componentes del sistema y cómo se relacionan entre ellos. El Gateway Server es responsable de la comunicación entre los gateways físicos y el Network Server, gestionando el tráfico de datos entre ellos, también se encarga de realizar operaciones de seguridad, como la autenticación y el cifrado de los mensajes entre el gateway y el servidor de red. Esto asegura que la comunicación sea segura y protegida. Ofrece funcionalidades como el descubrimiento automático de gateways y la configuración remota de parámetros. El Network Server es responsable de gestionar y controlar el tráfico de datos dentro de una red LoRaWAN. Soporta distintos métodos de activación, como ABP (Activations By Personalization) y OOTA (Over-The-Air-Activation) y además permite agrupar los dispositivos en un sistema de aplicaciones, en caso de que se tenga un sistema que requiera distintos dispositivos similares. Este se conecta con el Application Server, que es el responsable de levantar una interfaz gráfica para facilitar el registro y gestión de los sistema a utilizar, contiene una consola en la que se pueden ver los componentes registrados y el flujo de datos entre los dispositivos. Además, permite realizar integraciones con otras aplicaciones, tales como MQTT, HTTP, AWS, Microsoft Azure y Google Cloud. Pero no se limita solo a estas, al ser un sistema de código abierto, se pueden realizar las integraciones que se deseen.

2.7. DOCKER

Docker es una plataforma de código abierto diseñada para facilitar la ejecución de aplicaciones en entornos aislados. Cada uno de estos entornos es llamado contenedor y contiene todo lo necesario para que la aplicación funcione correctamente, incluyendo las dependen-

cias, bibliotecas, frameworks y archivos de configuración.

Al utilizar una aplicación en un contenedor, se agrega una capa de abstracción (Figura 8), esta actúa por encima del sistema operativo, por lo que es posible acceder a las dependencias instaladas localmente, pero además se pueden instalar dependencias aparte para el funcionamiento de la aplicación.

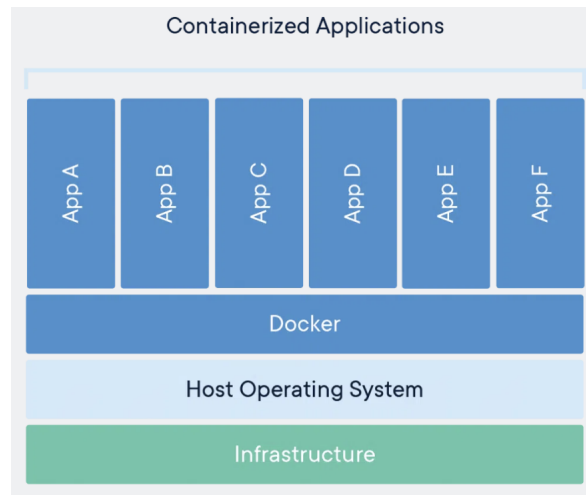


Figura 8: Ejemplo de Aplicaciones en Contenedor.
Fuente: [Docker, 2013].

2.8. TRABAJOS RELACIONADOS

Dentro de los trabajos relacionados, se buscaron distintas plataformas que implementen sistemas IoT en zonas de difícil acceso, así comparar los resultados ya sean positivos o negativos sobre la implementación de distintos sistemas en zonas remotas. Además, se buscaron otros sistemas de monitoreo, ya sean de monitoreo náutico, de animales o de UAVs, para así analizar los distintos enfoques utilizados y el cómo se utilizan las redes de larga distancia.

En primer lugar, se analizarán los distintos sistemas agrícolas basados en LoRaWAN. Al ser zonas de extensas superficies, LoRaWAN es una solución ideal para el monitoreo de los cultivos. En [Ahmed *et al.*, 2022] se propone un sistema de monitoreo agrícola utilizando dicha red y distintos tipos de sensores. Dentro de estos se encuentran sensores de temperatura, humedad, clima, luz, imágenes y vídeo. Con esto se busca mantener un registro de las plantaciones a estudiar, además de realizar un seguimiento de las condiciones en las cuáles crecen. Para realizar esto se propuso el sistema ejemplificado en la Figura 9. Cuenta con cuatro capas, una capa de percepción de la agricultura, dónde están los cultivos, luego una capa de sensores y gateway, dónde se recolectan y envían los datos, una capa de comunicación dónde está implementada la manera de interactuar entre servidores y sensores, y por último, una capa de aplicación dónde está un centro de control y monitoreo de datos.

En términos de hardware, se utilizó un gateway LoRa Dragino DLOS8 [Ahmed *et al.*, 2022] que cuenta con una interfaz LoRa SX1301. Se posicionó el gateway luego utilizando cuatro dispositivos ESP32 TTGO GPS NEO6 se midieron las distancias a la cuál la señal era recibida por el gateway, con esto se pudo cuantificar el alcance del gateway en distintos escenarios. El ambiente elegido para realizar el experimento fue una cancha de fútbol a campo abierto, así no estaba la interferencia de obstáculos cómo edificios o árboles, con esto además se controlaba el experimento en un ambiente parecido a un centro agrícola.

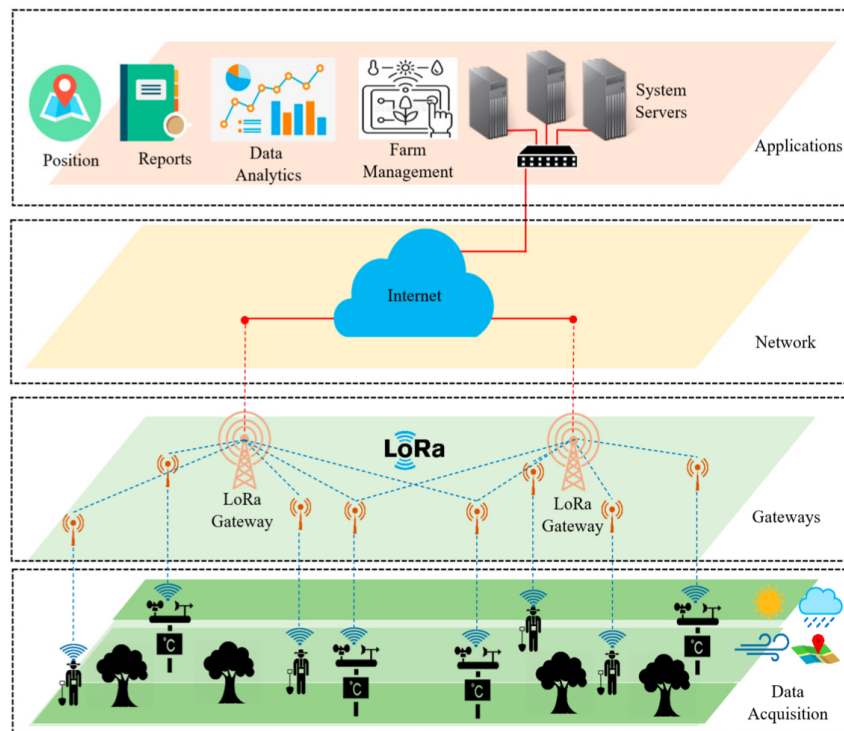


Figura 9: Arquitectura del sistema propuesto.
Fuente: [Ahmed *et al.*, 2022].

Otro sistema agrícola es el implementado entre el año 2016 y 2018 descrito en [Davcev *et al.*, 2018]. Dentro de este trabajo se realiza un análisis de las distintas redes que ofrece el mercado y son comparadas según el alcance y cobertura, el consumo de batería y la privacidad de la red. Se decide utilizar LoRaWAN debido a el largo rango de transmisión y la oportunidad que entrega para cubrir la mayor parte de los campos agrícolas. Además, se destaca el bajo consumo de batería de los sensores que trabajan con esta red, puede actuar en base a baterías y con esto, tienen una duración de años, eliminando así el problema de la carga externa de los dispositivos y flexibiliza la arquitectura del sistema. Otra de las ventajas que se mencionan es la capacidad de utilizar el espectro no licenciado de frecuencia, se compara con otras redes que no utilizan este espectro y se llega a la conclusión que, de usar una red que actúe en el espectro licenciado, se tendría que contratar un servicio para manejar la red, por lo que se pierde el control y la privacidad del sistema. Dentro de los puntos bajos de

LoRaWAN, se tiene que este cuenta con muchos problemas para soluciones de tipo *Indoor*, pero cómo el sistema a diseñar será para monitorear áreas al aire libre, esto no es tomado en cuenta para tomar la decisión de red.

En la Figura 10 se puede observar la arquitectura del sistema. Cuenta con varias diferencias con el sistema anteriormente mencionado, la arquitectura anterior esta definida en capas dónde se definen los roles y cómo interactúan entre ellas. En esta arquitectura el sistema es horizontal, si bien también cuenta con sus divisiones, los componentes interactúan entre ellos utilizando REST Endpoints. Cómo Network Server utiliza The Things Network IoT Platform, dentro administra el flujo de datos recolectados por los sensores y enviados por el gateway localizado en la zona. Para la visualización de datos se generó una página web dónde el usuario tiene control remoto de la base de datos y así pueden acceder a los datos desde cualquier parte.

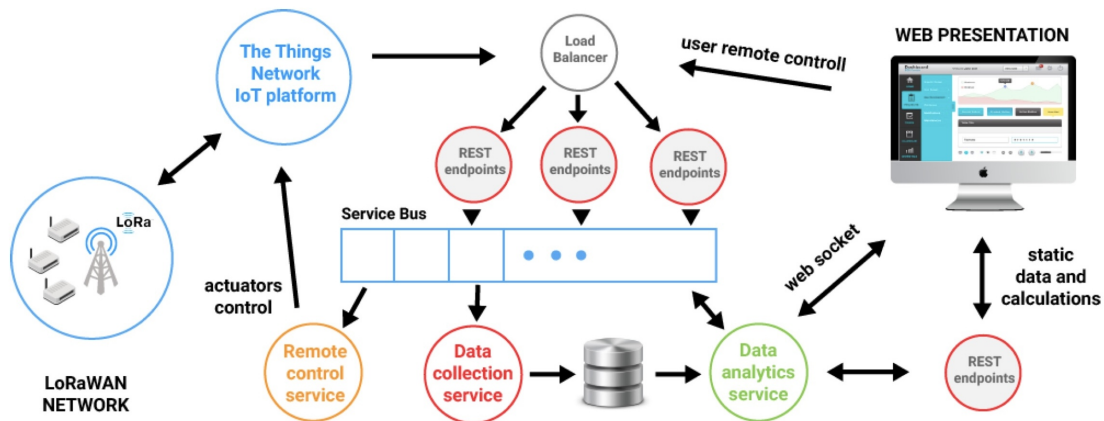


Figura 10: Arquitectura del sistema propuesto.
Fuente: [Davcev *et al.*, 2018].

Para probar el sistema, se realizó el monitoreo de una viña, dónde se midió la temperatura, la humedad del ambiente, la humedad de las hojas y la humedad de la tierra en un periodo de un año y 6 meses, entre enero 2016 y julio 2017. Se encontró que un 7.21% de los datos obtenidos se encuentran en un rango dañino para el ambiente, sobretodo la temperatura. Se espera que en trabajos futuros se pueda extender el análisis utilizando modelos de predicción y técnicas de Data Mining.

Dentro de los trabajos de monitoreo, se encuentran trabajos que abordan el tema de monitoreo de animales, en India se encuentra una población considerable de elefantes, estos tienden a cruzar carreteras provocando accidentes, tanto para el animal cómo para el ser humano. Es por esto que en [Mohandass *et al.*, 2021] se presenta un estudio realizado con dos enfoques, monitorear la salud de elefantes en entornos forestales, midiendo su temperatura basal y el ritmo cardiaco de estos, y además, se presenta un sistema para monitorear animales utilizando GPS (*Global Positioning System*), con esto se puede conocer la posición de los animales de manera remota. Si se identifica alguna condición crítica con respecto a

su salud, se activa un sistema de alerta a oficiales o veterinarios para proveer la asistencia necesaria. Si se detecta que un animal está por cruzar una carretera, se activa una alarma para alentarlos en la dirección contraria. Como sensores se utilizó sensores de movimiento y sensores de infrarrojo pasivo, estos son conectados a un gateway de LoRaWAN que transmite los datos a The Things Network. El sistema fue probado en una simulación y demostró ser efectivo en la detección temprana de intrusiones y en el monitoreo de la salud de los animales en un entorno forestal. Se deja cómo trabajo futuro el incluir podómetros en el sistema para detectar el ejercicio realizado por lo elefantes y así continuar monitoreando la salud de ellos.

Luego, se buscaron trabajos relacionados con el monitoreo marítimo. En el vasto escenario de la industria marítima, el tracking marítimo emerge como una piedra angular fundamental para el éxito y eficiencia de las operaciones náuticas. La capacidad de rastrear y monitorear en tiempo real la ubicación y el movimiento de embarcaciones y cargamentos se ha convertido en una herramienta esencial para garantizar la seguridad, la optimización logística y la toma de decisiones informadas. En este contexto, [Sanchez-Iborra *et al.*, 2019] presenta un sistema de seguimiento y monitoreo de embarcaciones basado en la tecnología LoRa. Se ofrece una descripción completa de esta solución de comunicación y se discuten sus ventajas cuando se aplica en escenarios marítimos. En la Figura 11 se muestra la arquitectura elegida, una arquitectura de capas donde se muestra cómo los botes se comunican utilizando LoRaWAN con el gateway, este redirige los datos vía Internet hacia un Data Server y en la última capa se presentan los datos en un dashboard hacia los usuarios remotos.

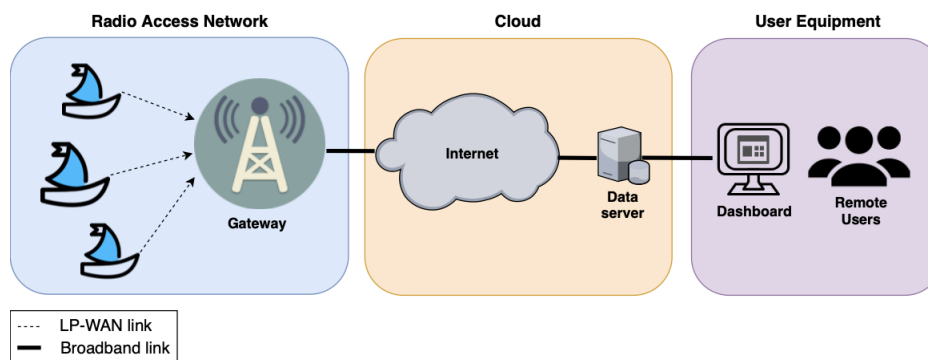


Figura 11: Arquitectura basada en red LP-WAN del sistema propuesto.

Fuente: [Sanchez-Iborra *et al.*, 2019].

Se presentan resultados extraídos de un caso de estudio, donde se monitorearon sesiones de entrenamiento de veleros de la Clase Optimist en Vigo, España, utilizando la arquitectura propuesta. Se realizó un estudio de cobertura en el área del puerto, donde se observaron zonas sin cobertura debido a la presencia de edificios altos que bloqueaban las transmisiones desde las embarcaciones. Se compararon dos configuraciones de LoRa y se encontró que usando SF7 se obtenía una mayor frecuencia de transmisión de datos en comparación con SF12. En cuanto a la robustez de los enlaces de transmisión, se obtuvieron niveles de tasa de

entrega de paquetes (PDR) superiores al 97 % en condiciones de buena transmisión.

En las conclusiones, se destaca que las comunicaciones marítimas pueden beneficiarse de las tecnologías de comunicación de IoT. Se considera adecuado el uso de soluciones LP-WAN para barcos con restricciones energéticas y de carga, como veleros pequeños o dispositivos marinos de bajo costo. Se presenta un estudio del rango de transmisión que muestra una distancia máxima de 4 km, considerada suficiente para monitorear embarcaciones ligeras. Además, se evalúa el consumo de energía de los dispositivos finales y se concluye que es posible alcanzar una autonomía de un día con baterías económicas. Como trabajo futuro, se planea evaluar el rendimiento del sistema en un escenario diferente, con embarcaciones más grandes y utilizando otra tecnología de transmisión.

Por último, se mencionará un trabajo relacionado con el monitoreo de UAV (vehículos aéreos no tripulados), en los últimos años, el mundo ha experimentado un crecimiento vertiginoso, transformando la forma en que el ser humano interactúa con el mundo. Desde aplicaciones militares y de seguridad, hasta misiones de rescate, agricultura de precisión y producción audiovisual, los UAV se han convertido en una poderosa herramienta que nos permite alcanzar nuevas alturas y descubrir horizontes inexplorados. En [Mason *et al.*, 2020] se presenta un sistema de seguimiento remoto de UAVs que combina el estándar de comunicación LoRaWAN con un modelo de movimiento tridimensional llamado 3D-CTRA. El sistema está diseñado para estimar y rastrear con precisión las trayectorias de los UAVs en un entorno tridimensional. Se utiliza el algoritmo Unscented Kalman Filter (UKF) para realizar el seguimiento y predecir las trayectorias de los UAVs. Los resultados de simulaciones muestran que el sistema propuesto mejora significativamente el rendimiento de seguimiento en comparación con los enfoques de seguimiento convencionales. El artículo destaca la viabilidad del uso de la tecnología LoRaWAN para transmitir datos de estado de UAVs a larga distancia, lo que permite la supervisión remota de drones en diferentes escenarios.

CAPÍTULO 3

DISEÑO DEL SISTEMA

En esta sección, se proporcionará una descripción exhaustiva del sistema propuesto para lograr los objetivos establecidos anteriormente. Se presentarán los requisitos que debe cumplir, así como las especificaciones que deben tener los componentes a elegir en etapas posteriores. Se describirá la arquitectura del sistema, la conectividad entre gateway y sensores, además de describir el hardware de sensores y servidor Edge a utilizar.

3.1. DESCRIPCIÓN

Para cumplir los objetivos propuestos, se deben analizar los requisitos que debe cumplir el sistema a proponer, tanto los requisitos funcionales como los no funcionales. Los requisitos funcionales son los que identifican las acciones del sistema, que "debe hacer". Los requisitos no funcionales son los que indican el comportamiento deseable del sistema. En las siguientes secciones se definirán los requisitos definidos para el sistema a proponer, luego se ahondará más en detalle cada uno de ellos.

3.1.1. REQUISITOS FUNCIONALES

- **Monitoreo de sensores:** el sistema debe ser capaz de recopilar y monitorear los datos medidos por los sensores instalados.
- **Análisis de datos en tiempo real:** el sistema debe ser capaz de procesar y analizar los datos ambientales recolectados en tiempo real, así proporcionar información útil a los gestores de la reserva.
- **Control remoto:** el sistema debe permitir el control remoto de dispositivos de la reserva natural, así poder realizar un mantenimiento más fácil.
- **Comunicación y conectividad:** el sistema debe ser capaz de establecer y mantener una conexión confiable con los dispositivos de la reserva y con los usuarios del sistema, utilizando tecnologías inalámbricas y de red adecuadas.

3.1.2. REQUISITOS NO FUNCIONALES

- **Escalabilidad:** el sistema debe ser escalable para manejar el intercambio de mensajes y el flujo de datos de múltiples sensores, y así poder crecer en capacidad y funcionalidad con el tiempo.

- **Disponibilidad:** el sistema debe estar disponible y en funcionamiento durante la mayor cantidad de tiempo posible, de modo que pueda proporcionar una supervisión constante de la reserva natural.
- **Interoperabilidad:** el sistema debe ser interoperable con otros sistemas y dispositivos para poder integrarse sin problemas con los sistemas existentes de la reserva natural.
- **Mantenibilidad:** el sistema debe ser fácil de mantener y actualizar, permitiendo el reemplazo o actualización de dispositivos y componentes individuales del sistema sin interrumpir su funcionamiento.
- **Usabilidad:** el sistema debe ser fácil de usar y comprender por los usuarios, incluyendo a los gestores de la reserva natural y otros usuarios que puedan interactuar con el sistema.
- **Eficiencia energética:** el sistema debe ser eficiente en el uso de la energía, utilizando tecnologías de bajo consumo para reducir la huella de carbono y minimizar el impacto ambiental del sistema.
- **Bajo Costo:** el sistema debería ser del menor bajo costo posible, para así lograr que sea lo más accesible posible a la comunidad.

Luego de describir los requisitos necesarios para cumplir con el objetivo de este trabajo, se describirán los componentes de cada una de las capas a implementar y cómo cada uno de esos cumplirá con alguno de los requisitos previamente mencionados.

3.2. ARQUITECTURA

La arquitectura del sistema se refiere a la estructura en que se organizarán los componentes y la relación entre ellos. La arquitectura debe ser una que cumpla con la mayor cantidad posible de los requisitos planteados en la sección anterior. La estructura más simple de un sistema se basa en un servidor, ya sea físico o en la nube, que recibe mensajes y envía una respuesta en caso de ser necesario, si se desean almacenar los mensajes (datos) enviados por los nodos, el servidor se conecta a una base de datos que puede estar dentro del mismo servidor o en la nube. Un servidor es el encargado de recibir y procesar los datos recibidos, por lo general, los datos que se reciben no están listos para almacenar, es por esto que se debe realizar un procesamiento de estos antes de ser enviados y este proceso es realizado en dicho servidor. En caso de tener una pequeña cantidad de nodos, el servidor logra aguantar la cantidad de solicitudes que llegan de los nodos, manteniendo una latencia ¹ aceptable.

El sistema descrito anteriormente entra en la categoría de las arquitecturas de sistema centralizadas, esto debido a que se cuenta con un centro de servidores dónde se almacena y se

¹La latencia de un sistema es el tiempo que tarda un mensaje en ir de un nodo al servidor y que la respuesta se devuelva.

procesa toda la información en un mismo lugar. La principal desventaja de estos sistemas es que se genera un único punto de falla, si falla el servidor, el sistema queda sin disponibilidad y no existiría ningún soporte para este. Otra desventaja es la poca escalabilidad de estos sistemas, si se requiere unir una gran cantidad de nodos al sistema, de a poco la latencia va a ir aumentando, bajando así la eficiencia de este. Es por esto que en este trabajo se optará por una arquitectura descentralizada.

Dentro de los sistemas descentralizados, se utilizará un sistema *Edge Computing*. Esta arquitectura trata de tener varios servidores más cercanos a los bordes de la red, así los mensajes enviados por los nodos son al servidor más cercano y no a un único servidor. Con esto se logra disminuir la distancia que recorre el mensaje, lo que disminuye la latencia del sistema en general. Además, se puede introducir el concepto de *Fog Computing*, que propone a los nodos cómo los primeros en realizar un procesamiento de los nodos, así se alivia la carga del sistema, los servidores se concentrarían más en responder a los mensajes y no tanto en la limpieza de los datos que llegan. Por ejemplo, si se desea medir la temperatura del ambiente, el sensor puede enviar los datos que mide al servidor y el servidor ser el que emita una alerta cuando se supere cierto límite definido. En cambio, se podrían medir las temperaturas y analizar estas en el mismo sensor, así cuando se supere el límite, el sensor emita la alerta y se comuniquen con el servidor. Con esto se libera espacio en el ancho de banda y se puede utilizar para el monitoreo de otros parámetros. Este es un ejemplo trivial para representar la idea general del concepto, hoy en día están apareciendo distintos tipos de técnicas para realizar el análisis de datos en los nodos, además también se está potenciando el *hardware* de los sensores, así realizar procesamientos más intensos. Una de las técnicas recientemente desarrolladas es *TinyML*, que propone correr programas de Machine Learning en dispositivos pequeños. Esto es especialmente útil en el caso de realizar modelos de clasificación de datos, dónde el modelo se puede cargar en el mismo sensor o en un servidor Edge y que el mensaje a enviar sea la respuesta del modelo, no lo medido por los sensores.

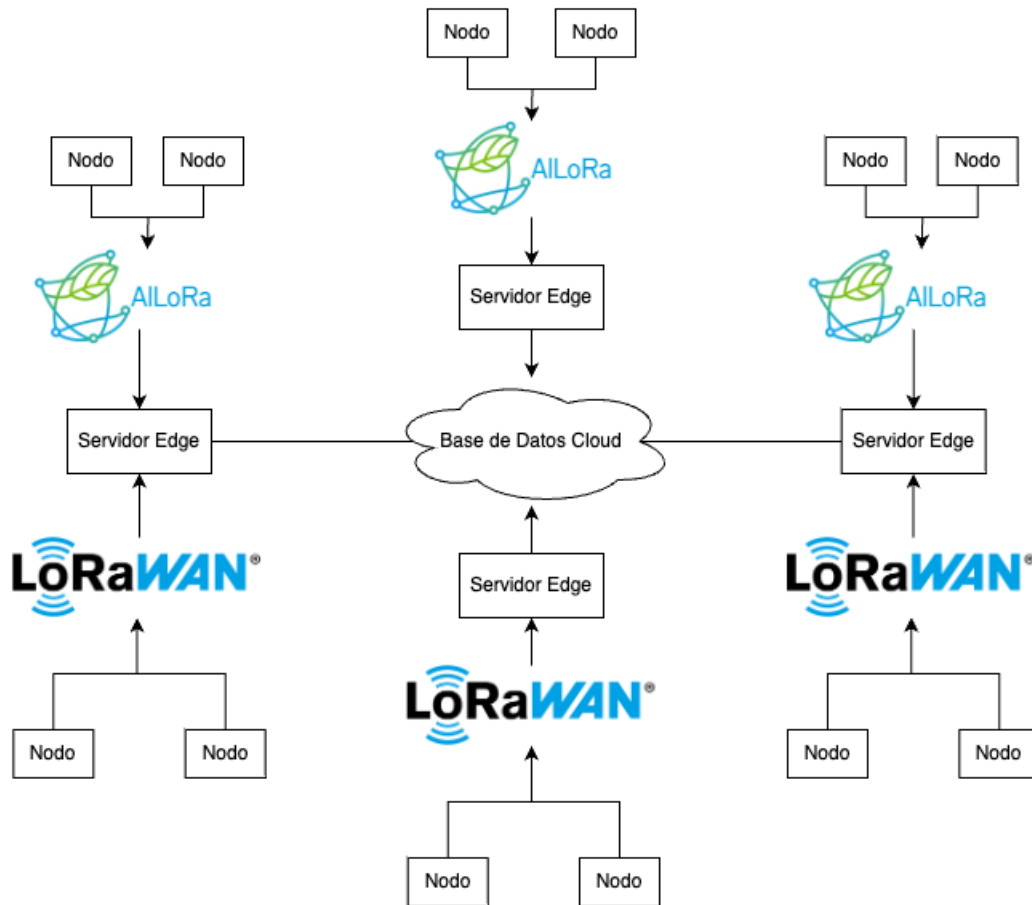


Figura 12: Ejemplo de arquitectura Edge para implementar en el sistema.
Fuente: Creación propia.

En la Figura 12 se puede ver la arquitectura a la que se busca llegar. Los nodos en la imagen representan a los dispositivos que se conectarán a los Servidores Edge, esto incluye a sensores, cámaras o dispositivos que actúen como puente para dar acceso a LoRa. Los Servidores Edge van a actuar como los administradores de los datos recibidos por los nodos, reenviando estos a una base de datos en la nube. Considerando que el sistema propuesto va a estar ubicado en zonas verdes remotas, guardar los datos de manera local no sería lo más cómodo para un posterior análisis, es por esto que se decide de utilizar la nube como un componente de la estructura, con esto es posible acceder a los datos en cualquier momento sin necesidad de realizar una conexión física con el sistema.

Dentro del sistema se muestran tres tipos de Servidores Edge, en primer lugar los servidores que realicen la conexión con los nodos mediante LoRaWAN, en segundo lugar los que realicen la conexión con los nodos mediante ALoRa, y por último los que realicen la conexión utilizando ambos métodos. Este último va a ser el implementado dentro de esta memoria.

3.3. CONECTIVIDAD

En términos de conectividad, se describirá de que manera se conectarán los distintos componentes del sistema entre sí y mediante que red. Como ya se ha ilustrado anteriormente, se utilizará como método de comunicación a LoRaWAN, una red que utiliza la modulación de espectro ensanchado (CSS) para transmitir datos de forma eficiente a larga distancia con baja potencia. Esta tecnología permite que los dispositivos IoT se comuniquen con los gateways a distancias de varios kilómetros en entornos rurales o urbanos, lo que es especialmente útil en zonas remotas donde la infraestructura de comunicación tradicional (como WiFi o redes LTE) puede ser limitada [Thomas y Eldhose, 2020].

Esta red se caracteriza por su eficiencia energética, permite una vida útil prolongada, tanto en gateways como en sensores. En [Bouguera *et al.*, 2018] se estudia el consumo energético de dispositivos LoRaWAN, se probaron distintos tipos de configuraciones de la red, alterando el *Spreading Factor*, el *Coding Rate* y las distancias a las que se transmiten los datos. Con esto se implementaron tres escenarios y se calculó la vida media de los dispositivos implementando LoRaWAN, se concluyó que una sola batería es capaz de durar hasta 5 años sin recargar. Sistemas implementados utilizando con LoRaWAN, no solo son capaces de transmitir datos a largas distancias, si no que también son capaces de generar una baja huella de carbono.

LoRaWAN implementa una arquitectura de estrella de estrellas, lo que significa que los dispositivos se comunican con los gateways pero no es posible la comunicación entre sensores, los gateways son los encargados de conectarse con el Network Server. En términos de escalabilidad, si bien es una red de *low power wide area network* (LPWAN), que logra admitir un gran número de dispositivos y transmitir los datos en grandes distancias, esta está limitada por la arquitectura que tiene. El gateway actúa como un servidor central en la zona local, lo que representa un único punto de falla que puede afectar al rendimiento del sistema y resultar en la pérdida de datos. Los gateway se pueden comunicar entre sí y enrutar los datos a una distancia lejana de los sensores, hasta llegar al Network Server implementado, pero si llega a pasar que alguno de los gateway conectados a los sensores falla, los sensores intentarán conectarse a otro gateway para mandar los datos, pero puede pasar que la distancia sea muy grande y se terminen perdiendo los datos.

Es por esto, que se decide implementar otro protocolo de comunicación, también basado en LoRa, para así ampliar la robustez del sistema frente a problemas. Allora es un protocolo de comunicación que permite realizar una malla de comunicación, los dispositivos pueden actuar como nodos receptores o como nodos de envío, en caso de no tener a ningún gateway cercano, este reenviará los datos a otro dispositivos (sensor) cercano para que este continúe con los saltos hasta llegar a un gateway con la información. Este tipo de arquitectura permite una mayor cobertura y una mayor tolerancia a fallos. Además, con este protocolo se logra enviar datos de mayor tamaño que desde LoRaWAN, ideal para monitorear aspectos del medio ambiente que antes no era posible. Por ejemplo, es posible enviar imágenes del sistema que pueden ser analizadas posteriormente.

El modo de malla (*mesh mode*) es una de las principales características que hacen que AllLoRa resulte ideal en el contexto de monitorear áreas remotas y de difícil acceso, siendo capaz de enviar datos en una distancia de 20 km aproximadamente [Arratia *et al.*, 2023b].

En esta memoria se llevará a cabo la implementación de un gateway que combina ambos tipos de conexiones en un único dispositivo. Esto es posible gracias a la capacidad del código de AllLoRa de utilizar adaptadores que brindan acceso a LoRa a dispositivos que originalmente no lo tienen. Por ejemplo, es posible establecer una conexión mediante WiFi o conexión serial entre una Raspberry Pi y una placa que incluye LoRa integrado, como la Heltec Wifi Lora 32. La placa se configura como adaptador, mientras que la Raspberry Pi se configura como un gateway. De esta manera, es posible permitir que dispositivos que no cuentan con LoRa integrado accedan al protocolo y reciban información desde los sensores. Esta característica del protocolo amplía las posibilidades para diversos tipos de sensores que anteriormente no tenían integrado LoRa, lo que significa que la búsqueda de gateways ideales para el sistema ya no se limita únicamente a aquellos con LoRa integrado.

3.3.1. NETWORK SERVER

El Network Server es el encargado de gestionar y controlar las comunicaciones en los gateways y los dispositivos en el borde de la red. Es un componente clave en las redes LoRaWAN, por lo que al elegir uno de las aplicaciones disponibles es vital para el funcionamiento. Dentro de las funcionalidades principales del Network Server se encuentra el control de acceso, verifica que los mensajes recibidos de los nodos cumplan con las normas de seguridad y valida la integridad de los datos, gestiona y coordina el tráfico entre gateways y dispositivos, enruta los mensajes hacia las aplicaciones correspondientes y viceversa, además es capaz de realizar ajustes dinámicos para garantizar que se están usando las configuraciones que permitan una transmisión óptima.

A continuación se compararán las dos opciones más populares del mercado, ChirpStack y The Things Stack (TTN). Ambas son implementaciones de Network Servers pero se diferencian en la arquitectura que tienen por detrás. Ambas cuentan con un Application Server, que es el que implementa la interfaz gráfica del sistema, conectando este al Network Server por detrás. Ambos Application Servers son capaces de realizar integraciones con otras aplicaciones como son bases de datos, aplicaciones para graficar datos o sistemas de solicitudes como HTTP. Esto presenta una gran utilidad a la hora de realizar un análisis de los datos colectados por el sistema. Además, considerando que el sistema va a estar en zonas remotas, guardar los datos de manera local representa una complicación para el análisis de estos.

La principal diferencia de ambos sistemas, es que TTN cuenta con una versión en la nube basada en los servicios de AWS, lo que significa que es posible enrutar los datos a un servidor en la nube y acceder a las gestiones del sistema desde cualquier lugar. Los dispositivos enviarían los datos al gateway más cercano y este los enviaría a la nube. Como consideración, el gateway debe tener acceso a la red LoRaWAN y además debe contar con acceso a Internet

por algún método, lo que aumentan los requerimientos de hardware del gateway y además alza nuevas problemáticas relacionadas a cómo acceder a Internet desde zonas remotas.

ChirpStack corre de manera local, es un software de código abierto, por lo que no es necesario pagar por una suscripción para utilizarlo. Funciona coordinando sus componentes dentro de contenedores de Docker, al Application Server, Network Server y al Gateway Bridge. Los datos pueden ser enviados por el gateway a aplicaciones en la nube o ser guardados de manera local. Esto implica las mismas complicaciones anteriormente mencionadas, cómo acceder a Internet en zonas remotas. Aún así es posible guardar los datos de manera local, se tendría que tener en consideración que el gateway debe tener espacio suficiente para lograr guardar los datos por un tiempo razonable.

3.4. SENSORES

Dentro de los sensores a utilizar, se requiere elegir los que se adapten más a lo que se desea medir por los administradores de las áreas verdes, los parámetros que puedan generar un aporte a los análisis futuros. Existen distintos tipos de sensores, cada uno con características que los hacen deseables, pero para efectos de este trabajo de tratará de medir los parámetros más generales que comparten las áreas verdes, especialmente la temperatura y las condiciones del aire. En el contexto de Chile, una de las principales amenazas que sufren las áreas verdes, son los incendios forestales y la tardía detección por parte de las autoridades. Con sensores que detecten la temperatura y las condiciones de viento, se podría realizar alertas para cuando las condiciones sean ideales para la propagación del fuego. Dentro de este trabajo de limitará al análisis de estos parámetros pero para trabajos futuros se puede continuar expandiendo el sistema a otros tipos de sensores.

Además de los sensores que contenga la placa a elegir, esta debe ser capaz de cumplir con el requisito de eficiencia energética. Al estar trabajando sobre zonas remotas, estos deben ser capaces de mantenerse de manera autónoma sin necesidad de estar revisando en cortos periodos de tiempo la batería de los sensores. Es por esto que las placas Arduino resultan ideales para el cometido de este trabajo.

Por último, las placas a elegir deben ser capaces de conectarse a LoRaWAN, para así tener un sistema unificado que logre conectarse con la antena elegida. Esta red va a ser la que conecte todo y además, como fue mencionado anteriormente, conectar el sistema de esta manera ayuda a mantener la eficiencia energética, debido a que la red LoRaWAN no presenta un gasto de batería tan significativo como las redes LTE o WiFi.

Las placas que tienen sensores integrados y además conexión a LoRaWAN, son escasas y por lo general, no cuentan con los sensores específicos que se necesitan. Por lo que, en vez de adaptar las placas a los problemas, se puede realizar una solución utilizando una placa que cumpla el rol de comunicación y otra que cumpla el rol de monitoreo. Por ejemplo, las placas LoRa 32 [Automation[®], 2020], pueden realizar la conexión con LoRaWAN y, mediante

conexión serial, se puede conectar otra placa que cuente con los sensores necesarios, con esto se pueden utilizar los sensores que se deseen sin tener que estar preocupándose por la conexión a LoRaWAN. Por ejemplo, la placa Arduino Nano 33 BLE cuenta con sensores de temperatura, humedad, movimiento, giroscopio, entre otros [Arduino®, 2019], pero no cuenta con conexión a LoRaWAN, realizando una conexión se puede adaptar esto al sistema propuesto.

Se utilizará las placas Arduino Portenta H7 junto con Portenta Visual Shield para comprobar la conexión LoRaWAN con el gateway. Cumplen con las especificaciones mencionadas anteriormente, además de que mantienen los requisitos a satisfacer. Las características de cada una de estos son presentados a continuación. Para probar la conexión vía AllLoRa, se utilizará la placa Arduino Nicla Vision con la placa Pycom LoPy4.

3.4.1. ARDUINO PORTENTA H7 Y PORTENTA VISUAL SHIELD

Es una placa Arduino que permite correr código de alto nivel al mismo tiempo que realiza tareas en tiempo real. Cuenta con un procesador dual core STM32H747 junto con un procesador Cortex® M7 corriendo a 480 MHz y un procesador Cortex® M4 corriendo a 240 MHz. Ambos se comunican vía Remote Procedure Call, esto permite que el llamado a funciones de un procesador a otro pase desapercibido por el usuario. Este tipo de procesador es necesario debido a que puede correr programas de Machine Learning dentro del mismo dispositivo. Al realizar una red distribuida, esta característica logra aliviar el sistema completo, bajando la latencia de las llamadas al servidor principal. Para efectos de este trabajo no se ahondará en el tema de TinyML, pero se deja cómo una característica que en el futuro puede ser muy importante.

Además, esta placa cuenta con sensor de temperatura que, contando con los procesadores internos, se podría realizar un pre-procesamiento de los datos internamente antes de ser enviados a los servidores. Esta es la principal característica que permitió tomar la decisión de utilizar esta placa.

Portenta Visual Shield es una placa que se adhiere al Arduino Portenta H7 y aporta con nuevas funcionalidades, de las más destacables es su capacidad de conexión a LoRaWAN. La solución propuesta utiliza este medio de comunicación, por lo que utilizar este shield es ideal para el desarrollo de la solución. Al tener conexión con LoRaWAN, se puede lograr la conexión entre dispositivo y gateway todo con un mismo dispositivo. Cuenta con una cámara y un micrófono que podrían resultar útiles a la hora de realizar distintos análisis de los parámetros presentes en las reservas naturales.

En la Figura 13 se muestra la distribución de los sensores en ambas placas.

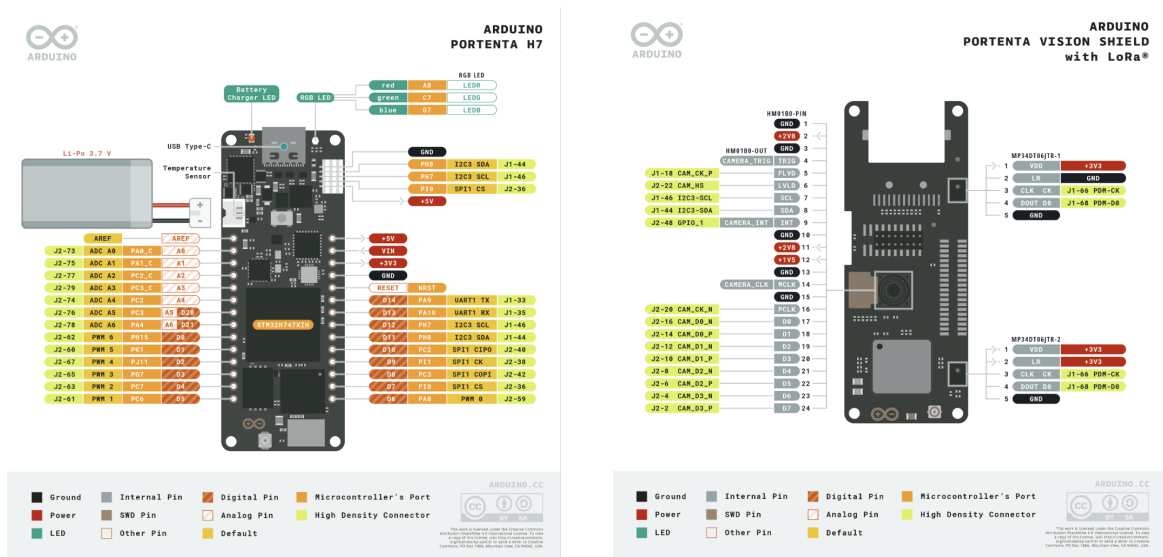


Figura 13: Distribución de Sensores de Arduino Portenta H7 y Portenta Visual Shield.
Fuente: [Arduino®, 2022]

3.4.2. ARDUINO NICLA VISION

Arduino Nicla Vision es una placa que combina la captura de imágenes con su procesamiento. Con su potente procesador STM32H747AI16 Dual ARM® Cortex®, permite cargar programas de Machine Learning directamente en la placa, promoviendo así el uso de Edge Computing y aliviando la carga del sistema. Además, es compatible con MicroPython y las bibliotecas de TinyML, lo que brinda flexibilidad para diversos programas según los objetivos deseados.

La placa incluye una cámara a color de 2Mp, así como sensores de movimiento y un micrófono, lo que permite medir y analizar datos en un solo lugar. En términos de conectividad, cuenta con WiFi y Bluetooth de baja energía para facilitar la comunicación con otros dispositivos [Arduino®, 2023].

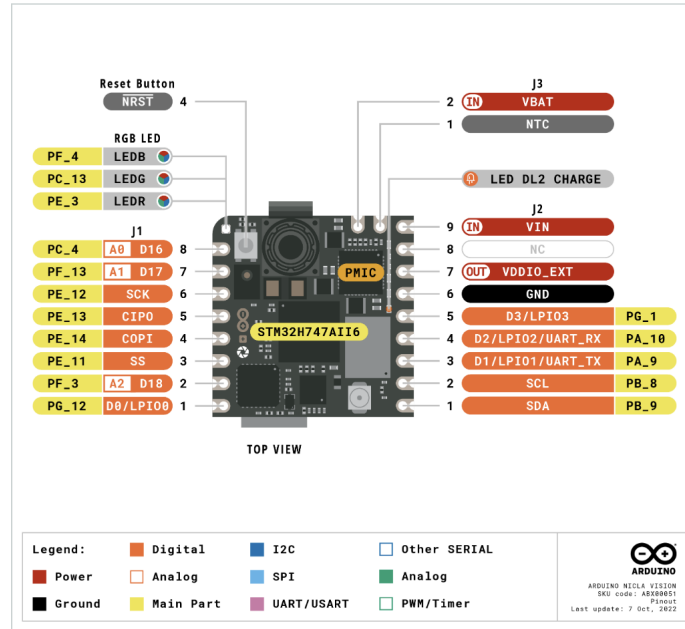


Figura 14: Distribución de Sensores de Arduino Nicla Vision.
Fuente: [Arduino®, 2023].

3.4.3. PYCOM LOPY4

LoPy 4 es una placa de desarrollo pequeña y versátil diseñada para aplicaciones IoT. Es parte de la línea de placas de Pycom enfocadas en el largo alcance de comunicación y el bajo consumo energético. Cuenta con un microcontrolador ESP32 que puede ser programada en Python y MicroPython, esto permite el procesamiento de programas dentro de la placa. Además, cuenta con WiFi, Bluetooth, Sigfox y LoRaWAN, una antena integrada y puerto GPIO dónde se pueden conectar sensores adicionales.

Esta placa es capaz de realizar las soluciones IoT completas, puede recopilar los datos de los sensores, procesarlos y enviarlos de manera inalámbrica y a larga distancia, considerando que cuenta con LoRaWAN. Por último, es una placa de bajo consumo, por lo que puede ser capaz de durar un largo tiempo, ayudando así al medio ambiente [Pycom, 2018].

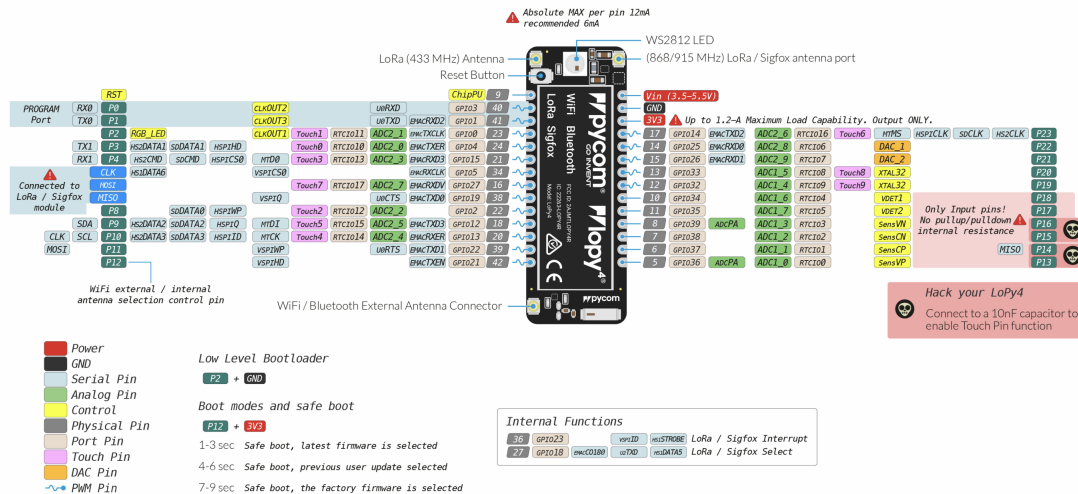


Figura 15: Distribución de Sensores de LoPy4.
Fuente: [Pycom, 2018].

3.5. SERVIDOR EDGE

Al estar en zonas remotas, es necesario instalar un gateway que provea la señal para los dispositivos. Si el sistema fuera implementado en zonas urbanas, se podría utilizar los gateway desplegados en las ciudades y utilizar estos como intermediario entre los sensores y los servidores que procesen los datos. Como este no es el caso, es necesario discutir que tipo de gateway se implementará como parte del sistema propuesto.

En primer lugar, un gateway es el intermediario entre los sensores y el servidor, permite que los datos se envíen mediante la red elegida de manera bidireccional. En el contexto de este trabajo, la red a utilizar será LoRaWAN, esto nos lleva a elegir gateways que soporten esta red y que además, tengan conexión de algún otro tipo con los servidores, ya sea mediante WiFi o conexiones seriales directas. Como se estará implementando el sistema en zonas remotas, lo ideal sería tener un sistema fácil de llevar y que conlleve la menor cantidad de configuraciones iniciales. En primer lugar, se pensó en implementar un gateway de LoRaWAN y luego levantar un servidor en algún dispositivo aparte, por ejemplo, una Raspberry Pi. Las Raspberry Pi son muy populares a la hora de implementar sistemas IoT debido a su pequeño tamaño y a la gran capacidad que puede llegar a tener, además de su económico precio. La última versión de este dispositivo cuenta con puertos tres USB, un puerto Ethernet, dos puertos Micro HDMI que soportan resolución de hasta 4k y, en términos de capacidad, se pueden elegir distintas versiones de RAM, desde 1GB hasta 8 GB [Pi®, 2019]. En el mercado hoy en día, se pueden encontrar desde \$35 USD, lo cual es un precio considerablemente bajo para un dispositivo de esta capacidad.

Una primera aproximación al problema puede ser conectar una Raspberry Pi con un gateway de LoRaWAN. Los gateway de LoRaWAN varían en precio y calidad, pueden ir desde \$20 a

\$150 USD, las principales diferencias entre estos son la calidad de la señal y la cobertura que ofrecen. Cómo el sistema se quiere implementar en zonas remotas, es necesario que la cobertura de la señal sea lo más amplia posible pero manteniendo un costo razonable para el sistema. Cómo segunda aproximación, se propone un sistema que utilice la misma Raspberry Pi cómo gateway, conectando un dispositivo que entregue señal de LoRaWAN, ya sea un concentrador de LoRaWAN o un dispositivo que tenga antena, cómo las placas Heltec WiFi LoRa 32. Estas son placas diseñadas para proyectos de IoT, cuenta con WiFi, BLE (*Bluetooth Low Energy*), LoRa y una OLED display [Automation®, 2020]. Al conectar una de estas placas a una Raspberry Pi por conector serial, se puede lograr montar un gateway de LoRaWAN y además continuar con las funciones de servidor de la Raspberry Pi. Hoy en día, se pueden utilizar varios programas en paralelo dentro de un mismo computador y considerando la capacidad de memoria RAM actual con la que vienen los dispositivos, se puede aprovechar al máximo y lograr así un mismo resultado con menos dispositivos.

Por último, se tiene que tener en cuenta que cualquier solución que se implemente, debe tener una capa que proteja al sistema de los posibles cambios climáticos que pueden suceder a lo largo del año en una reserva natural. El sistema debería ser capaz de resistir lluvias, calores extremos o bajas temperaturas, todo sin que necesite ir a realizar configuraciones o ajustes adicionales. Es importante anticipar todas las posibles dificultades que puedan surgir al instalar el sistema en áreas de difícil acceso. Para garantizar una protección adecuada de las placas, existen diversas opciones como cajas de metal herméticas que pueden protegerlas de los elementos externos. Además, para evitar el sobrecalentamiento de las placas, es posible incorporar dispersores de calor en su interior, permitiendo que el sistema funcione de manera eficiente y protegida durante un período prolongado.

A continuación, se mencionaran distintos dispositivos que cumple con los requisitos planteados anteriormente, esto considerando los costos y la disponibilidad, además de que sean aptos para el exterior.

3.5.1. DL08N

Este dispositivo es fabricado por Dragino y está diseñado específicamente para su uso en entornos exteriores en aplicaciones LoRaWAN. Es un gateway de largo alcance que combina la tecnología LoRaWAN con conectividad 4G para garantizar una conectividad confiable, esto resulta especialmente útil al diseñar un sistema en zonas remotas. Dentro de las características más importantes, se encuentra su carcasa resistente a exteriores, está diseñada para soportar condiciones climáticas adversas y así proteger los componentes internos del gateway. Puede operar en entre -20 y 85 °C, siendo así muy versátil a la hora de ser posicionado en el exterior [Dragino, 2023b].

En términos de conexión, este gateway actúa cómo un puente entre LoRaWAN e Internet, vía WiFi, Ethernet o redes celulares. Cuenta con un concentrador SX1302 de LoRaWAN, uno de los concentradores de última generación, mejora la temperatura del sistema, el consumo de

batería y además, tiene 8 diferentes canales de LoRa, además de demoduladores de SF5-SF12 [Semtech, 2020]. La capacidad de contar con el concentrador de LoRaWAN más la posibilidad de conectarse a redes celulares como 4G, hace que este gateway sea una opción viable para el sistema a diseñar. Se tiene que tener en consideración que este gateway reenviará los datos una larga distancia a otro servidor pero no será capaz de realizar un pre-procesamiento de los datos en él, por lo que para implementar este gateway como parte de la solución, sería necesario considerar un servidor adicional que se conecte vía Internet con el gateway.

Es compatible con los Network Server mencionados anteriormente, The Things Network y ChirpStack. Se puede conectar a estos servicios en su versión Cloud y los datos pueden ser extraídos mediante las distintas integraciones que estos servicios cuentan.

En términos de costos, el precio varía entre 300 y 350 USD en el mercado, esto puede variar dependiendo de la tienda en la que se compre y además de que tipo de concentrador se agregue. Antes de realizar la compra se debe elegir la frecuencia en la que será usado, por lo que esa decisión puede afectar al costo.



Figura 16: Gateway LoRaWAN DLOS8N.
Fuente: [Dragino, 2023a].

3.5.2. RAK7249

Es un gateway de LoRaWAN de la marca RAKWireless, diseñado para aplicaciones de comunicación de largo alcance manteniendo el bajo consumo energético. Cuenta con una carcasa de exterior que permite mantener los componentes internos protegidos del clima. Tiene una autonomía de hasta 5 horas sin conexión, en caso de que se acabe la batería, utiliza para alimentarse utiliza conexión PoE (*Power Over Ethernet*). Dentro de las sugerencias del proveedor, se encuentra un kit para alimentar la antena de mediante energía solar. Si bien, el precio es bastante elevado (739 USD), este elemento puede aumentar la independencia del gateway y así lograr un sistema más flexible a la hora del posicionamiento del gateway. La integración de la energía solar al sistema es una ventaja de este gateway sobre los otros ejemplos mencionados.

Tiene integrado un concentrador de LoRaWAN dual, que soporta hasta 16 canales distintos de frecuencias, al momento de realizar la compra se debe especificar cuantos canales va a soportar, 8 canales o 16 canales. Además, puede ser conectada a Internet a través de WiFi, redes celulares y Ethernet. Puede ser configurado cómo una Basic Station de LoRaWAN y es compatible con los Network Server tradicionales, ChirpStack, The Things Network, AWS IoT Core para LoRaWAN, etc. Además, cuenta con GPS, el cuál resulta muy útil a la hora de realizar la sincronización de mensajes con otros servidores. El precio del gateway varía entre 372 y 525 USD, dependiendo de la cantidad de canales que se quieran, si se desea tener red celular y la frecuencia en la que se desea operar [Wireless, 2023a].

Aparte, se debe realizar la compra de la antena a utilizar para LoRaWAN. En la página oficial son sugeridas dos tipos de antenas de fibra de cristal con distintas ganancias, de 5.8 dBi y de 8 dBi. Si se tiene el gateway con 8 canales, es necesario comprar solo una antena, para el gateway de 16 canales, dos antenas. Sus precios varían entre 40 y 60 USD.



Figura 17: Gateway RAK7249.
Fuente: [Wireless, 2023a].

3.5.3. RAK7391

Es un gateway basado en Raspberry Pi CM4, de la marca RAKWireless. Fue diseñado para su uso en aplicaciones de comunicación inalámbrica de largo alcance, utilizando LoRaWAN o Sigfox. Cuenta con varias interfaces de conexión para adaptarse a las distintas necesidades de los usuarios. La característica más atractiva para efectos de este trabajo es la mezcla entre Raspberry Pi y antena de LoRaWAN, con esto es posible recibir los datos en el gateway, además de tener la posibilidad de instalar de manera local el Network Server a utilizar, en este caso, ChirpStack. Es una antena compacta de 17,5 cm de longitud y 1,5 cm de diámetro, por lo que no produce demasiadas molestias en caso de ser posicionada en algún parque turístico. Cuenta con diversas interfaces que se adaptan a las distintas necesidades de sus usuarios, cuenta con HDMI, Ethernet, USB, mPCIe, CSI, DSI, M.2, WisBlock, PoE, Raspberry Pi HAT [RAKWireless®, 2022b]. Puede ser utilizada como un gateway de LoRaWAN con hasta 16

sub-canales dentro del mismo dispositivo, lo que significa que es capaz de recibir mensajes en 16 distintas frecuencias a la vez, aspectos muy importante a la hora de diseñar un sistema escalable.

Cuenta con una ganancia de 5dBi, lo que permite una mayor cobertura y alcance de la señal, esto es especialmente útil en áreas donde la señal puede ser débil o intermitente. Esta es la principal característica que la hace ideal para el sistema propuesto, al tener un tamaño compacto y gran cobertura, se necesitan menos cantidad de gateways para cubrir grandes cantidades de terreno, disminuyendo así la contaminación visual que puede generar tener dispositivos tecnológicos en áreas verdes, sin sacrificar la cobertura espacial del sistema.

Para lograr la señal de LoRaWAN, se utilizar el concentrador RAK5146, cuenta con un conector mini-PCIe, lo cuál lo hace compatible con los puertos presentes en el gateway. En términos de señal de LoRa, está basado en el chip de Semtech, SX1303, la última generación de chips en el mercado, que cuenta con control de temperatura y con un sistema que reduce el consumo de batería a largo plazo. Además, tiene un software de Listen Before Talk, lo cuál hace que sea de fácil comunicación con el servidor instalado, en este caso, la Raspberry Pi. Por último, cuenta con un chip ZOE-M8Q de GPS integrado para mejorar el sincronización temporal de los datos enviados [RAKWireless®, 2022a].

Las especificaciones de software del sistema presentan diversas cualidades que lo hacen único. En la Figura 18 se puede observar la estructura interna del gateway. Cuenta con un sistema operativo llamado RAK Pi OS, fue basado en el sistema operativo base de Raspberry Pi, pero cuenta con alguna diferencias, dentro de estas el uso de Docker para contener distintas aplicaciones tales cómo LoRaWAN Packet Forwarder, LoRaWAN Network Server, los componentes de ChirpStack, The Things Stack, entre otras. Este sistema operativo cuenta con el kernel de Linux, por lo que la terminal de comandos utiliza la misma base.

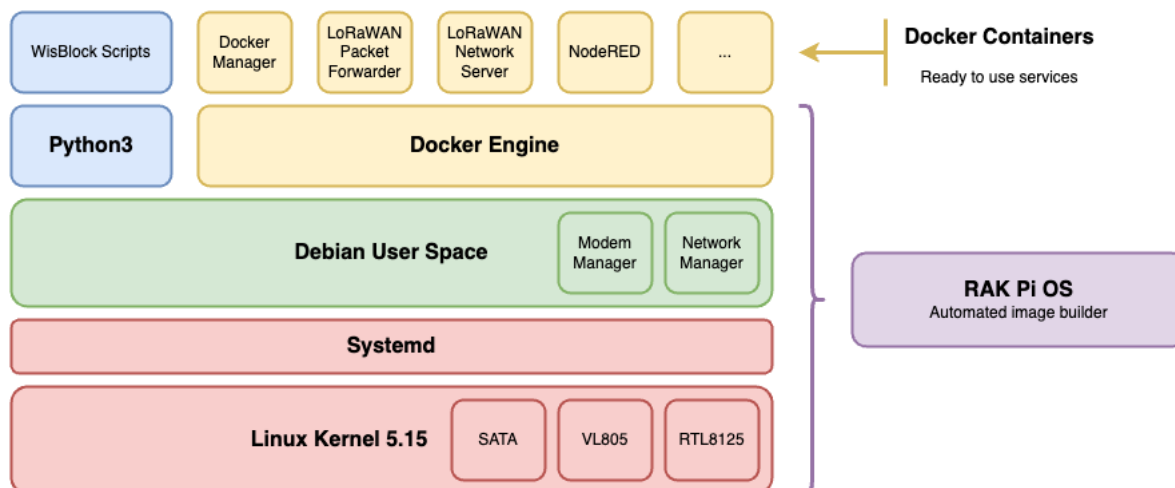


Figura 18: Estructura del Software.
Fuente: [RAKWireless®, 2022b].

Cómo fue mencionado anteriormente, para efectos de este trabajo se utilizará ChirpStack cómo Network Server del sistema de manera local, los tres componentes principales de este están montados en contenedores de Docker, el Network Server, App Server y Gateway Bridge. Cada uno de estos cumple un rol fundamental para lograr que los datos lleguen al servidor. En el puerto 1700 se encuentra el Gateway Bridge, el encargado de unir los mensajes TCP recibidos y enviarlos hacia el Network Server, montado en el puerto 8000. Por último, el encargado de la visualización de los datos, además de la configuración de los sensores que se agreguen a la red, es el App Server, que corre en el puerto 8080. Todos estos son contenedores de Docker, dónde se corre en local al iniciar el gateway. En la Figura 18 se puede apreciar la arquitectura interna de RAK7391.

La arquitectura del sistema propuesto queda ilustrado en la Figura 19. Se asocian los componentes de hardware con los componentes de software correspondientes, ejemplificando los roles específicos de cada uno de ellos. ChirpStack será la base del sistema, dónde se podrá ver el estado interno y dónde se podrá configurar el estado de los dispositivos conectados al sistema.

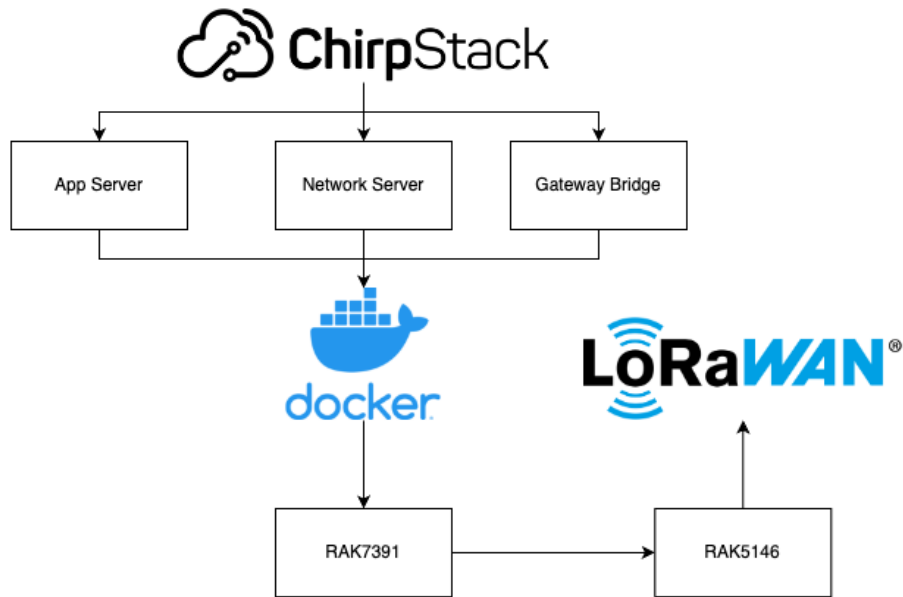


Figura 19: Arquitectura interna de RAK7391.

Fuente: Creación propia.

CAPÍTULO 4

EVALUACIÓN DEL SISTEMA

En la presente sección se realizará la descripción de la evaluación del sistema. Se describirá la implementación del sistema, tanto el hardware utilizado como el software. Luego se realizará un análisis de los costos relacionados, la escalabilidad del sistema, la cobertura y las posibles integraciones.

4.1. IMPLEMENTACIÓN

En esta sección, se detallarán los componentes implementados para llevar a cabo la experimentación del sistema. Se proporcionará información sobre el hardware y el software utilizados en cada parte del sistema, así como los aspectos que serán evaluados durante el proceso experimental.

Para garantizar una configuración adecuada de LoRaWAN, es fundamental tener en cuenta la frecuencia en la que operan los dispositivos. Cada región del mundo tiene asignada una frecuencia específica para el envío de mensajes. Es por ello que resulta crucial asegurarse de que las frecuencias utilizadas estén en consonancia con la región en la que se encuentran los dispositivos, de lo contrario, los mensajes no lograrán llegar a su destino.

Con el fin de llevar a cabo la experimentación de manera efectiva, se ha configurado los dispositivos en las frecuencias correspondientes a la región de Europa (868MHz). Esta experimentación fue realizada en la Universidad Politécnica de Valencia, España, y se tomaron en consideración las frecuencias establecidas para esta región.

En el proceso de experimentación, se describirán detalladamente los componentes implementados en cada etapa del sistema, se analizará el hardware y el software involucrado, y se evaluará cuidadosamente el rendimiento y la efectividad del sistema en relación con los objetivos establecidos.

4.1.1. HARDWARE

En términos de hardware, se utilizaron los siguientes dispositivos para el gateway:

- **RAK7391:** Este es el principal componente de la implementación del sistema. Es el encargado de conectar el concentrador de LoRaWAN a la Raspberry Pi integrada en la placa. Además, a él se conecta el dispositivo Heltec WiFi LoRa 32 que actúa como

puente entre la antena de LoRa y el RAK7391. En la Figura 20 se puede ver a la placa vista desde arriba sin el cierre exterior.



Figura 20: Foto de RAK7391.
Fuente: Creación propia.

El gateway se alimenta utilizando el transformador suministrado con la compra, el cual se conecta mediante un cable Ethernet. Para ello, se conecta un extremo del cable Ethernet al puerto *Power over Ethernet (PoE)* del transformador y el otro extremo se conecta a uno de los dos puertos Ethernet del gateway. Para conectarse a Internet, puede realizarse de dos maneras, a través de un cable Ethernet conectado directamente a una salida de Internet, o a través de WiFi, el cuál se puede configurar directamente en la terminal de la Raspberry Pi. Para lograr ver la terminal y realizar las configuraciones necesarias, se debe conectar un monitor con un cable HDMI y también es necesario un teclado, para así lograr controlar la terminal.



Figura 21: Foto de RAK7391 dentro del cierre exterior.
Fuente: Creación propia.

- **LoRaWAN RAK5146:** Este va a ser el dispositivo que proveerá de LoRaWAN al gateway. Está conectado en el conector MiniPCIe #1 del gateway y a él se conecta la antena de LoRaWAN y el GPS. En la Figura 21 se puede ver el conector de la antena por el exterior y en la Figura 22 se ve cómo de arriba a abajo están conectados la antena LoRaWAN y el GPS respectivamente. Por temas de disponibilidad, se utilizó una antena provisional (Figura 23).

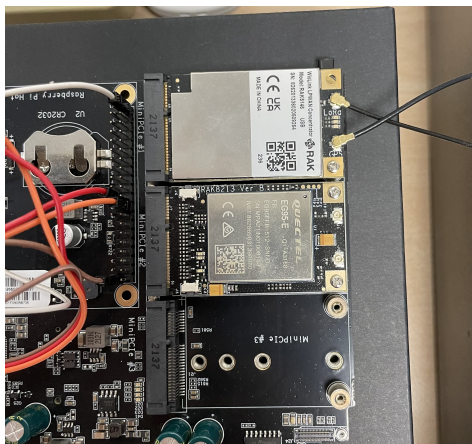


Figura 22: Foto de RAK5146 conectada a RAK7391 y foto de RAK5146.
Fuente: Creación propia.



Figura 23: Antena LoRaWAN.
Fuente: Creación propia.

- **Heltec Wifi LoRa 32:** Este dispositivo actúa como un segundo concentrador de LoRa para el gateway, así es posible recibir mensajes usando LoRa por dos distintas antenas. Con este dispositivo es que se logrará recibir mensajes mediante ALoRa. Al tener el concentrador RAK5146 actuando como antena para LoRaWAN y este dispositivo actuando como antena para ALoRa, se logrará recibir datos desde dos redes paralelas. Para el gateway, va a ser como si se hubiera agregado un canal extra por el cual recibir datos. Para la integración, se conectó el dispositivo al gateway mediante una conexión serial, se conectaron los pines 16, 17, GND y 5V del dispositivo con los pines 2 (5V), 21, 24 y GND del gateway. En la Figura 24 se puede ver cómo la conexión quedó integrada al gateway.

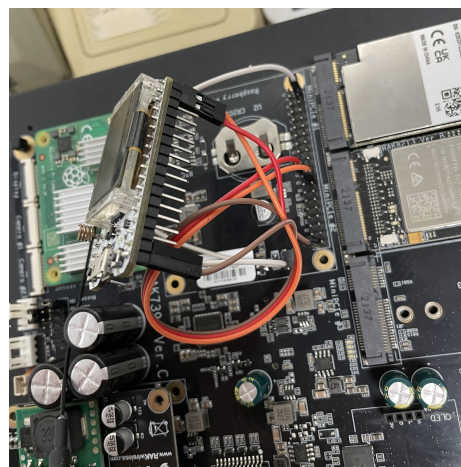


Figura 24: Heltec WiFi Lora 32 (v2) y fotografía de la integración al gateway.
Fuente: Creación propia.

En Tabla 1 se muestran los sensores a utilizar en la experimentación. Portenta H7 Pro y Portenta Visual Shield LoRa serán conectados y utilizados como uno solo, se utilizará el procesador del Portenta H7 Pro y la comunicación de LoRa del Visual Shield, este se comunicará con el gateway utilizando LoRaWAN. Con este sensor, se medirá la temperatura del ambiente, se espera probar la conexión con LoRaWAN y el correcto funcionamiento de la codificación de mensajes por parte del gateway. Probar que esto funciona sería el primer paso a demostrar el funcionamiento del sistema, por lo que se considerará exitoso que se envíe la temperatura desde el dispositivo al gateway y que este resultado se vea reflejado en la base de datos.

En cuanto a la Lopy4, esta será el puente entre la Nicla Vision y el gateway, se utilizará esto para probar la efectividad de AllLoRa como protocolo de comunicación. Lopy4 recibirá el dato desde la Nicla Vision y este lo enviará hacia el gateway. Al enviar un mensaje mediante AllLoRa, este se guarda en una carpeta, llamada como la dirección Mac del dispositivo. Esta carpeta se crea en el directorio del protocolo, por lo que el mensaje es recibido cuando dicha carpeta es creada.

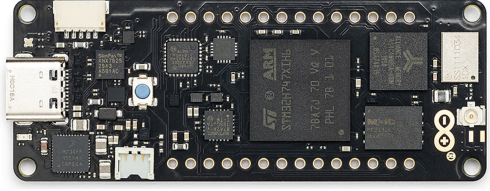
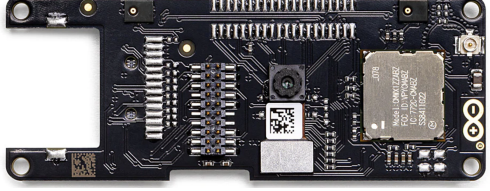
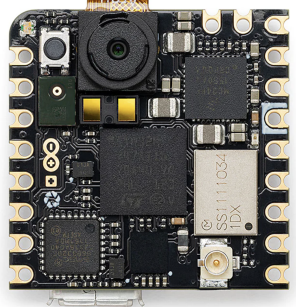

N°	Nombre	Imagen
1	Portenta H7 Pro	
2	Portenta Vision Shield LoRa	
3	Nicla Vision	
4	Pycom LoPy 4	

Tabla 1: Sensores a utilizar.
Fuente: Creación propia.

4.1.2. SOFTWARE

A continuación, se describirá el software a utilizar en la experimentación, esto incluye el Network Server a utilizar y la manera en que será instalado en el gateway, además de los códigos utilizados para la codificación y decodificación de mensajes. Por último, se describirá la base de datos en la que serán almacenados y visualizados los mensajes finales.

- Network Server:** Se utilizará ChirpStack instalado en el gateway utilizando Docker. Cada uno de los componentes de ChirpStack (Network Server, App Server y Gateway Bridge) será instalado en diferentes contenedores de Docker y se realizará la comunicación entre estos para que corra de manera local. El contenedor que contiene al Gateway Bridge estará en el puerto :1700, el App Server en el puerto :8080 y el Network Server en el puerto :8000. Para que funcione de la manera correspondiente, los contenedores deben correr al mismo tiempo, por lo que se tomó la decisión de correr los contenedores al iniciar el sistema y en segundo plano, por lo que al iniciar el gateway se prenden todos los contenedores y el sistema queda listo para utilizar. El gateway se debe registrar en el App Server, para esto se debe realizar la configuración inicial descrita en la página oficial de ChirpStack. En primer lugar, se debe conectar el gateway y un computador a la misma red de internet, al estar con la misma dirección IP, es posible conectarse al App Server desde el computador y así comenzar las configuraciones.

The screenshot displays the ChirpStack web interface for a gateway named 'rak-gateway'. The interface includes a navigation menu with tabs for 'GATEWAY DETAILS', 'GATEWAY CONFIGURATION', 'CERTIFICATE', 'GATEWAY DISCOVERY', and 'LIVE LORAWAN FRAMES'. A 'DELETE' button is visible in the top right corner. The 'GATEWAY DETAILS' tab is active, showing the following information:

Gateway details	
Gateway ID	e45f01fffe51b74b
Altitude	16 meters
GPS coordinates	39.48263, -0.34657
Last seen at	May 25, 2023 2:57 PM

To the right of the details table is a map showing the location of the gateway. The map includes a blue location pin and labels for various streets and landmarks, such as 'Cami de Vera', 'Ronda Nova', 'Avinguda dels Tarongers', and 'Campus dels Tarongers'.

Figura 25: Gateway registrado en ChirpStack.
Fuente: Creación propia.

- InfluxDB:** Como base de datos, se utilizará InfluxDB, una base de datos de series temporales que cuenta con versión Cloud y versión local. Dentro de las integraciones de

ChirpStack, se encuentra InfluxDB, dónde se puede configurar para dirigir los mensajes que llegan al gateway directamente a la base de datos, estos son organizados en dos tablas, una de ellas guarda los datos relacionados al *uplink*, los datos cómo el RSSI, los datos del dispositivo que envió los datos y la aplicación a la que está asociado. La otra tabla guarda los mensajes enviados hacia el gateway (ya decodificados), estos los clasifica por aplicaciones (registradas previamente en ChirpStack) y por dispositivos. Para efectos de la experimentación, se utilizará la versión Cloud, se enviarán los mensajes vía la API disponible y se comprobará que lleguen de manera correcta. Los datos que se guardarán en esta base de datos serán los enviados por el Arduino Portenta H7 Pro, este dispositivo los enviará mediante LoRaWAN hacia el gateway y el Network Server los redirigirá a InfluxDB.


En Anexo 1 se puede observar el código utilizado para medir la temperatura del ambiente y enviarla al gateway. Dentro del App Server de ChirpStack, es posible incluir un código para decodificar el mensaje que llega. Para esta prueba se envió un mensaje de texto (*string*) que contiene la temperatura en el momento. En primer lugar, se realizó un mensaje de prueba para comprobar que la conexión efectivamente funciona. En la Figura 26 se puede observar el mensaje ("HeLoRa world!") y los datos del dispositivo, además de la hora y día del envío. Dentro de ChirpStack es necesario realizar perfiles para los dispositivos que manden datos al gateway, esto para organizar a los dispositivos que manden el mismo tipo de datos sean organizados juntos, es por esto que el último dato del mensaje es el perfil al que está registrado el dispositivo.

```
▼ objectJSON: {} 1 key
  message: "HeLoRA world!"
  tags: {} 0 keys
  confirmedUplink: true
  devAddr: "01541451"
  publishedAt: "2023-04-05T11:23:20.098219997Z"
  deviceProfileID: "7d6537a7-b492-4906-ba2c-1330a80fcee8"
  deviceProfileName: "portenta-profile"
```

Figura 26: Mensaje de prueba recibido por el gateway.
Fuente: Creación propia.

En la Figura 27 se puede observar las configuraciones de la integración ChirpStack-InfluxDB. Cómo fue mencionado anteriormente, la prueba de conexión fue realizada en la Universidad Politécnica de Valencia (UPV), en España, por lo que el API endpoint fue configurado con Europa. Para que los datos lleguen correctamente, se debe registrar una organización y a esa organización asociarle un bucket, que va a ser el conjunto de tablas dónde se guardarán los datos. Para asegurar la seguridad de los datos, se utilizó un API Token, este es creado dentro de InfluxDB y se puede visualizar por el usuario

solo una vez, luego es copiado en el App Server de ChirpStack, con esto se logra que ninguna aplicación que no tenga esta serie de caracteres pueda escribir en la base de datos.



The image shows a web form titled "Update InfluxDB integration". It contains several input fields: "InfluxDB version *" with a dropdown menu showing "InfluxDB 2.x"; "API endpoint (write) *" with the text "https://eu-central-1-1.aws.cloud2.influxdata.com/api/v2/write"; "Organization *" with the text "UPV"; "Bucket *" with the text "rakupv"; and "Token *" with a masked field of dots. A blue button labeled "UPDATE INTEGRATION" is located at the bottom right of the form.

Figura 27: Configuración de integración ChirpStack-InfluxDB.
Fuente: Creación propia.

Una vez realizadas las configuraciones necesarias, se probó que funcionara utilizando el código (ver Anexo 1) en el Arduino Portenta H7. Una vez enviado el mensaje, este se puede ver en InfluxDB, cómo se ejemplifica en la figura.

- **AlLoRa:** Dentro de las configuraciones realizadas para el funcionamiento de AlLoRa en el gateway, se incluye la creación de un contenedor de Docker con el programa y la configuración del código necesario dentro de la Heltec WiFi LoRa 32. Se utilizó la Raspberry Pi dentro del gateway para la configuración de AlLoRa y este se configuró utilizando el código de Serial Gateway dentro de los ejemplos del repositorio oficial ². Se decidió por esta solución debido a que el RAK5146 (el encargado de entregar LoRa al gateway) está siendo ocupado por LoRaWAN, además, cómo se desea tener dos señales basadas en LoRa al mismo tiempo, es necesario implementar otra antena dentro del gateway y ese es el rol que cumplirá la Heltec WiFi LoRa 32. Una vez configurado el dispositivo cómo un adaptador utilizando el código antes mencionado, se conectó vía conexión serial al gateway uniendo los puertos seriales correspondientes para una conexión exitosa (ver Anexo 3 y 4). Fue necesario agregar líneas de código en el archivo de configuración inicial de la RaspberryPi, esta para habilitar los puertos UART del gateway y así habilitar el traspaso de mensajes entre la Heltec y el gateway.

Para instalar AlLoRa dentro del gateway, se decidió crear un contenedor de Docker con el programa a correr dentro, así al iniciar el gateway, este correría en segundo plano. Esto se decidió debido a la necesidad de mantener la conexión disponible para la recepción de mensajes, si el contenedor no se inicia por sí solo, sería necesario

²<https://github.com/SMARTLAGOON/AlLoRa/tree/main>

la intervención de alguien para que el sistema funcione y cómo la idea es tener este sistema instalado en zonas remotas, se optó por la opción que requiera menos intervención humana. Para que el contenedor se inicie en segundo plano, se utilizó Docker Compose, una herramienta que permite la gestión de las redes, volúmenes y distintos contenedores que conforman una aplicación. Se construyó la imagen del contenedor indicando las dependencias necesarias para el funcionamiento, cómo *pyserial* y el programa de ALoRa, que recientemente fue agregado a las dependencias de Python por lo que es posible instalarlo vía *pip3*. Luego, se generó un archivo *docker_compose.yml* con las configuraciones del sistema necesarias, esto incluye la imagen construida y los dispositivos a utilizar, en este caso, el puerto en el que se conecta la Heltec. Una vez realizado esto, se gestionó el iniciado en segundo plano con los comando respectivos de Docker Compose (ver Anexo 6). Una vez terminado, se comprobó que el sistema estuviera correctamente funcionando, cómo se ve en la Figura 28.

```

rak@rak-gateway:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
d4bb1a701830   allora:latest "python3 main.py --h..." 3 weeks ago   Up 2 days    0.0.0.0:3001->3001/tcp, :::3001->3001/tcp
e2bc28f730b4   postgres:14-alpine "docker-entrypoint.s..." 2 months ago  Up 2 days    5432/tcp
a9389241e17f   redis:7-alpine  "docker-entrypoint.s..." 2 months ago  Up 2 days    6379/tcp
rak@rak-gateway:~$

```

Figura 28: Terminal indicando los contenedores iniciados al prender el sistema.

Fuente: Creación propia.

Una vez todo estuvo listo, se realizaron pruebas para enviar mensajes desde la Pycom LoPy hacia el gateway, y se confirmó la recepción exitosa. El programa envía archivos de prueba de tamaño incremental en kilobytes, los cuales se guardan en una carpeta dentro del contenedor con el nombre de la dirección MAC del dispositivo emisor.

4.2. ANÁLISIS DE COSTOS

En la siguiente sección se describirán los costos asociados con el sistema presentado, esto incluye el costo inicial del hardware y la instalación, los servicios prestados por integraciones externas y el costo de mantención en el tiempo.

4.2.1. HARDWARE

Los siguientes precios serán obtenidos de las páginas de distribución oficiales de los productos, por lo que los precios serán indicados en dólares estadounidenses.

- **RAK7391:** La compra de este dispositivo incluye el Compute Module 4 de 4GB RAM, 32GB eMMC y Wifi. Además, el concentrador de LoRaWAN con 16 canales y apto para

las frecuencias US915/AS923/AU915/KR920. Por último, incluye la carcasa de exterior, apta para poder proteger al gateway de los diversos climas. Es necesario comprar la antena de fibra de vidrio, y así obtener el máximo rango de cobertura posible por LoRaWAN, y también el adaptador PoE Injector, para darle energía a la antena [Wireless, 2023b].

- **Arduino Portenta H7 Pro y Portenta Visual Shield:** Esta compra incluye ambas partes necesarias para lograr la comunicación entre gateway y sensor, además de incluir la antena de LoRaWAN. No incluye un cable USB-C para la comunicación con un computador [Store, 2023b].
- **Arduino Nicla Vision:** Al igual que el item anterior, esta compra incluye el dispositivo pero no el cable USB-C para conectarlo a un computador [Store, 2023a]
- **Pycom LoPy 4:** La compra de este dispositivo incluye solo el sensor, pero no la antena de LoRaWAN, por lo que se debe considerar esto a la hora de contemplar los costos [Electronics, 2023].

Nº	Dispositivo	Precio
1	RAK7391	\$531
2	Portenta H7 Pro	\$113
3	Portenta Vision Shield LoRa	\$51
4	Nicla Vision	\$115
5	Pycom LoPy 4	\$40
6	Antena de Fiberglass	\$40
7	1000M PoE Injector	\$21
	Total	\$911

Tabla 2: Tabla resumen de precios.

Fuente: Creación propia.

4.2.2. MIGRACIÓN

El sistema propuesto utiliza cómo base de datos a InfluxDB, una base de datos que puede ser utilizada en su versión local o en su versión Cloud. Anteriormente se describió la versión Cloud, cómo se realiza la integración con ChirpStack y cómo se configuraron las tablas base para guardar los datos en la nube, en esta sección se discutirán los costos asociados a utilizar la versión Cloud.

InfluxDB utiliza un modelo de precios flexible basado en el uso que el usuario haga de la plataforma, lo que implica que el costo se ajusta en función de la intensidad de la utilización. El sistema de tarifas se divide en cuatro categorías principales, cada una de ellas relacionada con aspectos específicos de la aplicación. Primero, tenemos el cobro por la cantidad de datos

que ingresan a la base de datos (Data In), con un precio de \$0.002 USD por megabyte ingresado. Seguidamente, se aplica una tarifa a la cantidad de consultas realizadas (Query Count), con un valor de \$0.01 USD cada 100 consultas. Además, existe un cobro asociado al volumen de datos almacenados (Storage), con un costo de \$0.002 USD por gigabyte-hora. Por último, se considera el flujo de datos transferidos a otras aplicaciones (Data Transfer Out), con una tarifa de \$0.09 USD por gigabyte [Data, 2023]. De esta manera, el usuario tiene la posibilidad de controlar y ajustar los costos en función de sus necesidades específicas, incentivando una gestión eficiente y adaptada a cada caso de uso.

Es esencial tener en cuenta estos costos a largo plazo, especialmente el costo de almacenamiento, ya que se utiliza un sistema de cobro por hora. Suponiendo un límite de 10 GB de datos para almacenar, el costo aproximado por un año de uso sería de \$195 USD. Esto incluye tanto el precio de ingreso de datos (Data In) como el costo de almacenamiento necesario para mantener los datos en el sistema. Es fundamental considerar este aspecto para tener una visión clara de la inversión requerida y tomar decisiones informadas sobre la implementación del sistema.

4.2.3. COSTO DE MANTENCIÓN

Para calcular los costos de mantención asociados al sistema presentado se realizará un análisis según los distintos aspectos a considerar, cómo lo es el hardware, los costos en el tiempo del lugar en el que se instalará y los costos asociados con el consumo energético y la conectividad.

- **Hardware:** Anteriormente se han descrito los costos de la compra inicial de los dispositivos a utilizar, por lo que ahora se discutirá la vida útil que tienen los dispositivos. Es difícil calcular exactamente la vida útil de los dispositivos debido a que esta depende de varios factores, cómo las condiciones de uso, la frecuencia de uso, la obsolescencia tecnológica y el mantenimiento realizado. Los dispositivos deben ser limpiados de manera periódica, ya sea el dispositivo físico y la memoria interna. Al ser pequeños, no cuentan con una memoria muy extensa por lo que es necesario realizar un mantenimiento de estos cada cierto tiempo. No existe mucha información de cuánto es la vida útil promedio de los Arduinos, debido a que la mayoría indica que depende de las condiciones mencionadas, pero en lo que están de acuerdo es que se debe realizar una mantención entre 4 y 6 meses después de la instalación.
- **Conectividad:** En términos de conectividad, los servicios utilizados para LoRaWAN (ChirpStack y AllLoRa) son de software libre, por lo que no cuentan con un costo asociado. Si se utiliza la versión Cloud de InfluxDB, se debe considerar el costo de Internet que debe utilizar el gateway para enviar los datos hacia la base de datos. Por último, los costos asociados con InfluxDB fueron especificados anteriormente y deben ser considerados a la hora de realizar el cálculo final de los costos del sistema.

- **Personal:** Para la mantención del sistema es necesario contratar a por lo menos una persona que se encargue de mantener el sistema en un estado de ejecución ideal. Se buscó dentro de los sueldos promedio de Chile y un Técnico en Informática gana un sueldo de alrededor de \$500.000 pesos chilenos al mes [Talend, 2023].
- **Arriendo:** El gateway central debería estar instalado en algún lugar que pueda proveer de la energía necesaria para que se mantenga encendido. Es posible desplegar el sistema de manera independiente utilizando baterías, pero esto aumenta el riesgo de falla y al ser un sistema de gateway central (una arquitectura de estrella de estrellas) es necesario asegurar que se mantenga conectado la mayor cantidad de tiempo que se pueda. Es por esto, que se debe considerar utilizar alguna cabaña o casa dentro del rango de conectividad en la zona remota a elegir.

4.2.4. RESUMEN

El análisis de costos para el sistema propuesto ha considerado tres aspectos fundamentales. En primer lugar, se evaluaron los costos de hardware, que abarcan la adquisición de sensores, dispositivos de comunicación y equipos necesarios para el funcionamiento del sistema en el entorno remoto. Estos costos fueron obtenidos de los vendedores oficiales y se considerará que los dispositivos mencionados se mantendrán con el cuidado necesario para que logren tener una vida media de más de un año. En segundo lugar, se tomaron en cuenta los costos de las licencias de software, cómo el Network Server es gratuito, solo se consideró la licencia para la base de datos en su versión Cloud. Por último, se analizaron los costos de mantenimiento, contemplando el gasto en la renovación del hardware, personal de mantención y las condiciones necesarias para instalar el gateway.

Tras un análisis minucioso de los costos involucrados, se estima que la inversión inicial para el primer año de uso se situaría alrededor de \$7000 USD. Es importante tener en cuenta que este monto no abarca otros gastos adicionales como los costos de electricidad, arriendo de infraestructura y posibles fallos que puedan surgir, especialmente al implementar un sistema nuevo.

Es vital recalcar que este cálculo se basa en un escenario con un gateway y una cantidad limitada de sensores. Sin embargo, es necesario tener una perspectiva más amplia, ya que el alcance del proyecto es mayor. Esta estimación proporciona un punto de partida para la versión inicial del sistema de monitoreo IoT, pero el costo total puede variar significativamente según las necesidades y la complejidad del proyecto en su totalidad. Es fundamental considerar todos estos factores para tomar decisiones informadas y efectivas durante la planificación y ejecución del proyecto.

4.2.5. OTRAS OPCIONES

El costo de implementar un sistema de monitoreo basado en internet puede variar significativamente dependiendo de diversos factores. En un enfoque de bajo costo, se pueden considerar sensores y dispositivos económicos, así como utilizar conectividad a internet a través de redes LTE o WiFi disponibles. Sin embargo, al optar por esta opción, surgen varias consideraciones importantes.

En primer lugar, el uso de redes LTE o WiFi implicaría el uso de dispositivos de comunicación que consumen más energía. Esto resultaría en una necesidad más frecuente de recargar baterías, lo que aumentaría los gastos de mantenimiento y requeriría intervención humana periódica. Además, el aumento en el consumo energético podría tener un impacto negativo en el medio ambiente, ya que aumentaría la huella de carbono del sistema.

Dado que este sistema está diseñado para ser implementado en reservas naturales, es fundamental considerar el impacto humano adicional causado por la intervención frecuente requerida para reemplazar baterías y dispositivos. En estas áreas, donde la actividad humana tiende a ser limitada, un sistema que demande mantenimiento constante podría afectar negativamente la zona de estudio y la vida silvestre presente.

Por otro lado, es importante tener en cuenta que la infraestructura requerida para utilizar redes LTE o WiFi es considerablemente mayor en comparación con la implementación de redes LoRa. Mientras que las antenas para LTE y WiFi están ampliamente desplegadas en zonas urbanas debido a su popularidad entre la población, en áreas rurales o remotas, el acceso a estas redes puede ser limitado o inexistente. Esto significa que para implementar un sistema de monitoreo basado en internet en estas áreas, sería necesario desplegar nuevas antenas, lo que aumentaría significativamente los costos del proyecto.

Además del aspecto económico, el despliegue de nuevas antenas tendría un impacto en el medio ambiente y en la zona de estudio. La construcción de nuevas estructuras y la instalación de antenas podrían afectar negativamente los hábitats naturales y la vida silvestre presente en la reserva. La expansión de la infraestructura de redes en áreas naturales puede alterar el equilibrio ecológico y desencadenar otros efectos no deseados.

Finalmente, es importante resaltar que Chile cuenta con una amplia diversidad de paisajes, que van desde desiertos hasta montañas y glaciares. En este contexto, considerar la implementación de una extensa red de internet que abarque todos estos diferentes climas y ecosistemas puede resultar altamente laborioso y, en muchos casos, innecesario. La mayoría de las infraestructuras de despliegue no son transportables, lo que implicaría la necesidad de instalar nuevas antenas en cada ubicación distinta.

Este proceso de despliegue adicional no solo implicaría un gasto considerable, sino que también conduciría a una subutilización de la infraestructura, desperdiciando recursos valiosos en más de un sentido. El impacto ambiental de este despliegue también debe ser considerado, ya que la instalación de nuevas antenas en áreas naturales podría perturbar los hábitats

y afectar negativamente la vida silvestre.

4.3. COBERTURA

En esta sección se discutirá el alcance de cobertura del sistema propuesto y cómo este es afectado por los distintos escenarios naturales que existen en Chile.

El sistema se basa en la tecnología LoRaWAN, lo que implica que comparte la misma cobertura que esta red, obteniendo tanto sus ventajas como desventajas. En áreas urbanas, los mensajes pueden ser enviados en un rango de 2 a 5 km, mientras que en áreas rurales, el alcance se extiende entre 12 a 15 km [Adelantado *et al.*, 2017]. Esta cobertura considera la comunicación entre un gateway y un dispositivo en el borde de la red. La arquitectura del sistema es una estrella de estrellas, permitiendo la implementación de varios gateways y múltiples dispositivos finales.

Sin embargo, es importante tener en cuenta que la escalabilidad de LoRaWAN está restringida por el ciclo de trabajo, un parámetro regulado por los gobiernos en todo el mundo [Finnegan *et al.*, 2018]. Este ciclo de trabajo define la cantidad máxima de tiempo que un gateway o dispositivo puede utilizar el ancho de banda de la red, generalmente limitado a tan solo el 1 % del tiempo (equivalente a 36 segundos por cada hora) [Benkahla *et al.*, 2018]. En otras palabras, durante el tiempo restante, el dispositivo debe permanecer inactivo o en modo de escucha. A pesar de la posibilidad de agregar más gateways, la cantidad de tiempo que efectivamente se puede utilizar para enviar mensajes estará restringida por el ancho de banda máximo regulado en las frecuencias ISM (frecuencias del espectro no licenciado) mencionadas anteriormente.

En cuanto a la cobertura en distintos paisajes, existen varios estudios relacionados a comprobar si afectan o no a la cobertura de la señal de LoRaWAN. En [Avila-Campos *et al.*, 2019] se encontró que el alcance de la transmisión de LoRaWAN en bosques ribereños (bosques a la ribera de algún cuerpo de agua como un río o lago), puede ser atenuada fuertemente por la vegetación, y que la cobertura puede variar significativamente entre entornos urbanos y rurales. En [Ansah *et al.*, 2020] se evaluó la propagación de LoRaWAN en un entorno de vegetación tropical y se encontró que los modelos de propagación de vegetación tradicionales se ajustan bien a los datos, excepto en el área del dosel arbóreo, donde la pérdida fue de aproximadamente 56 dB. En [Kunz y Karpinska, 2021] se probó la visibilidad y la intensidad de la señal de la red en un área urbanizada y se encontró que la cobertura de la señal y la visibilidad pueden variar según la densidad de los edificios. Finalmente, en [Mendes *et al.*, 2022] se realizó una prueba de cobertura y alcance antes de la implementación de una red de sensores LoRaWAN en un huerto de cítricos y se encontró que el despliegue de la red tuvo un efecto más perjudicial en la calidad de las comunicaciones de LoRaWAN que lo reportado previamente para entornos forestales. Es por esto que, a pesar de tener un rango estándar de envío de mensajes, la cobertura de la red estará definida por el tipo de zona en que se despliegue.

Además de LoRaWAN, se tiene AllLoRa. Si bien comparten a LoRa cómo capa física, se diferencia en la manera de enviar mensajes. Dentro de la publicación del protocolo, se realizó una prueba de rango en el Mar Menor, un lago ubicado cerca de Murcia, España. Se colocó el Sender Node en un bote y se posicionó en el mismo lugar en que en un futuro se colocaría una boya fija. El gateway fue posicionado en el techo del Instituto Español de Oceanografía a una altura aproximada de 25 metros. Luego de instalar todo, se logró enviar un mensaje a 11.69 km [Arratia *et al.*, 2023a]. Esto indica que la cobertura de AllLoRa puede lograr un alto rango de distancia. Se debe considerar que en esta prueba, no se contaba con obstáculos de por medio, cómo edificios o una masa densa de árboles, es por esto que se deben continuar realizando pruebas para poder definir correctamente el rango de cobertura de AllLoRa. En la Figura 29 se puede observar el mapa de RSSI que deja un mensaje al ser transmitido desde el Sender Node al gateway.

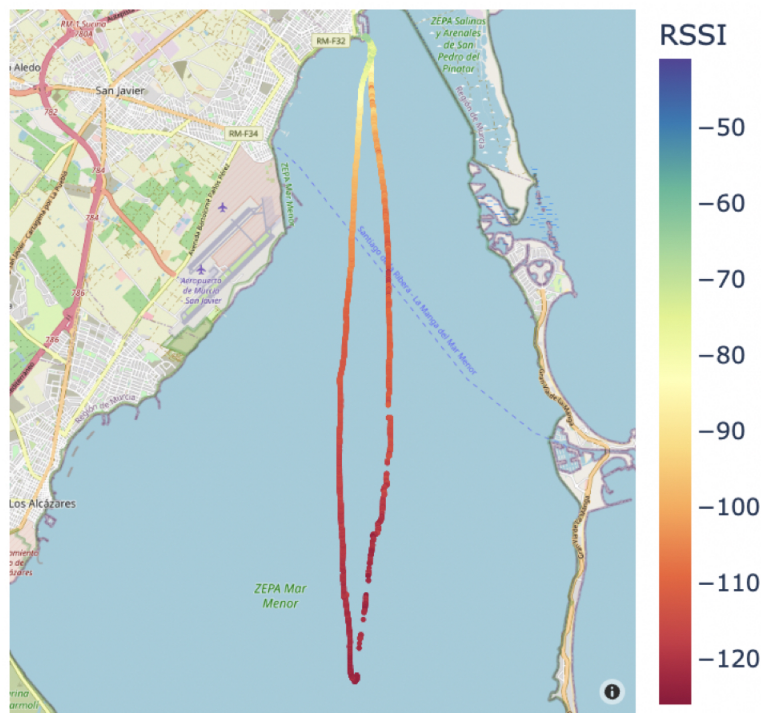


Figura 29: Mapa de distribución de RSSI en el Mar Menor, Murcia, España.
Fuente: [Arratia *et al.*, 2023a]

4.4. ESCALABILIDAD

En la presente sección se realizará un análisis de la escalabilidad del sistema. Se analizará LoRaWAN y AllLoRa, se verá cómo afecta el límite de envío de mensajes de la red y además el efecto de utilizar una base de datos basada en Cloud versus implementada de manera local.

Al utilizar LoRaWAN, se obtienen las mismas características de la red y, al igual que con la cobertura de la red, la escalabilidad también es limitada por el ciclo de trabajo mencionado. Si bien la red podría tener muchos gateways adicionales, estos se mantendrían inactivos debido al tiempo límite dado por las regulaciones.

Esta limitación del ciclo de trabajo se implementa con el objetivo de evitar una sobrecarga del ancho de banda por parte de un único usuario o dispositivo, y así asegurar un uso equitativo y justo de la red para todos los usuarios. Es una medida para incentivar un uso responsable y eficiente del espectro de comunicación, garantizando que el recurso esté disponible para el mayor número de personas.

La necesidad de planificar el envío de mensajes de manera eficiente surge debido a consideraciones clave. Por ejemplo, al medir la temperatura, no sería óptimo tomar una medición cada minuto, ya que las variaciones serían mínimas. En lugar de ello, se puede tomar una medición cada 30 minutos para evitar sobrecargar el ancho de banda con datos similares y que no aportan significativamente al análisis deseado. La cantidad de datos enviados dependerá de la información necesaria y del análisis ambiental requerido. Sin embargo, esta decisión debe ser tomada por el responsable de la implementación del sistema.

Otra de las razones de porqué LoRaWAN no es un red del todo escalable es la manera en la que procesa los mensajes. Utiliza el protocolo ALOHA, un protocolo que se basa en un esquema de acceso aleatorio y sin coordinación centralizada. En lugar de esperar su turno para transmitir, los dispositivos en la red pueden enviar datos en cualquier momento, lo que da lugar a una transmisión no determinista y potencialmente colisiones. Las colisiones de mensajes llevan a pérdida de paquetes, por lo que mientras más mensajes se envíen, más pérdida de paquetes se generará. En [Thomas y Eldhose, 2020] se realizó un estudio de la escalabilidad de LoRaWAN y se llegó a la conclusión de que utilizar el mismo canal y el mismo SF (*Spreading Factor*) conlleva a colisiones y pérdidas de paquetes, además de que por la naturaleza de ALOHA, se sigue enviando el mensaje hasta que se declara recibido por lo que la pérdida de paquetes puede tener como consecuencia una carga extra en la red. Si se tienen muchos dispositivos del mismo tipo, enviando mensajes con el mismo SF, podría terminar en perjudicar el sistema más que ayudarlo.

En el contexto de un sistema para inteligencia ambiental, LoRaWAN resulta ser una opción ideal. Al monitorear reservas naturales, es fundamental medir aspectos clave del entorno, pero esto no implica necesariamente la utilización de millones de dispositivos. Esta elección puede ser justificada por dos razones fundamentales. En primer lugar, la presencia de una gran cantidad de dispositivos en la zona de la reserva conllevaría un impacto ambiental significativo. El propósito de este sistema es monitorear la reserva de forma efectiva sin comprometer su estado natural y controlando la actividad humana en el área. Por lo tanto, mantener un equilibrio entre el monitoreo y la conservación del entorno es primordial. En segundo lugar, no se requiere una redundancia excesiva en la medición de ciertos datos. Por ejemplo, en el caso de la temperatura, no es necesario medirla en cada metro cuadrado, ya que los patrones climáticos suelen ser más uniformes en áreas extensas. Por tanto, la prioridad recae en contar con una cobertura amplia y eficiente en lugar de buscar una escalabilidad

infinita.

Por otro lado, ALLoRa se encuentra en fase de desarrollo para mejorar la escalabilidad de la plataforma. Por el momento, el protocolo es compatible con microcontroladores basados en ESP32, como la Heltec Wi-Fi LoRa 32 utilizada en esta implementación. Sin embargo, debido a que se utiliza un único canal de comunicación, las posibilidades de escalabilidad están limitadas. Se está trabajando en una versión multi-canal que permitirá una red más densa, pero esto implica consideraciones importantes al implementar las nuevas versiones [Arratia *et al.*, 2023b].

A pesar de la limitación de un solo canal, el sistema actual ofrece ventajas significativas. Su amplio rango de cobertura y la capacidad de enviar una gran cantidad de datos a largas distancias mejoran considerablemente la flexibilidad del sistema. Esto lo diferencia de otras soluciones basadas únicamente en LoRaWAN. Aunque el camino hacia la escalabilidad total está en desarrollo, la actual flexibilidad y rendimiento de ALLoRa en distancias extendidas hacen que sea una opción valiosa para aplicaciones que requieran amplia cobertura y transferencia de datos eficiente.

Para realizar el análisis de base de datos, se considerará el despliegue de varios gateways en vez de solo uno. Si se utiliza una base de datos de manera local, los datos estarían distribuidos en los distintos gateways y para realizar la extracción de estos, se debería ir físicamente a cada uno de los gateways (en caso de que estos no cuenten con Internet). En el caso de que se utilice una base de datos basada en Cloud, los datos serían redirigidos por el gateway de manera inmediata, por lo que el análisis y estudio de los datos se podría realizar en tiempo real.

Cada una de las opciones cuenta con ventajas y desventajas, la opción local complica la extracción de datos pero da la opción de desplegar los gateways de manera independiente, sin la necesidad de un lugar que provea energía eléctrica e Internet. Si se utiliza Cloud, es necesario instalar los gateways en zonas ya habitadas y cómo el objetivo es monitorear zonas remotas, no colabora mucho con el tema. Escalar el sistema va a resultar en realizar un análisis más profundo en cómo y dónde desplegar los gateways.

En el caso de ALLoRa, esta red es capaz de utilizar los dispositivos en el borde de la red para reenviar los mensajes hasta llegar al gateway más cercano. La arquitectura es distinta a LoRaWAN, por lo que se utiliza esta red como una ventaja para el sistema. Es necesario mencionar que esta red se encuentra en investigación, por lo que los alcances reales y probados no están terminados.

Existen más opciones de como aumentar la escalabilidad del sistema, sobre todo distintos tipos de algoritmos utilizados para la coordinación de mensajes. En [Thomas y Eldhose, 2020] se menciona la posibilidad de cambiar ALOHA por *Slotted Aloha*, en esta variante, el tiempo se divide en ranuras, los dispositivos solo pueden transmitir datos al comienzo de una ranura. Esto ayuda a sincronizar las transmisiones y reduce la probabilidad de colisiones. Aun así, si ocurre una colisión, el dispositivo debe esperar hasta la siguiente ranura antes de volver a

intentar la transmisión. Implementar cambios en el protocolo de LoRaWAN puede resultar en menos colisiones pero se estarían utilizando más recursos en los gateways, por lo que el consumo de energía aumentaría.

4.5. INTEGRACIONES

En la actualidad, el mundo está repleto de diversas aplicaciones, cada una con sus beneficios únicos que pueden llevar a proyectos como el que se presenta en este trabajo a un nivel superior. Es altamente deseable lograr un sistema que sea fácilmente actualizable y adaptable a diferentes situaciones, especialmente considerando que se implementará en zonas de difícil acceso. ChirpStack, seleccionado como el Network Server en este proyecto, ofrece importantes ventajas al integrarse con aplicaciones externas, lo que resulta en un sistema completo con la capacidad de ajustarse a las condiciones que se presenten. Esta integración proporciona un enfoque integral y robusto, asegurando que el sistema pueda enfrentar diversos desafíos y maximizar su potencial en la monitorización de zonas remotas. Algunas de las principales integraciones con otras aplicaciones son:

- **Aplicaciones de visualización y análisis de datos:** ChirpStack puede integrarse con aplicaciones de visualización y análisis de datos, como dashboards personalizados, herramientas de análisis de datos en tiempo real o plataformas de Business Intelligence (BI). Esto permite a los usuarios ver y comprender de manera más efectiva los datos recopilados por los dispositivos IoT, lo que facilita la toma de decisiones informadas. Un ejemplo de esta aplicación es Grafana, una herramienta de visualización de datos y análisis que permite crear dashboards personalizados con gráficos, tablas y paneles interactivos. La integración con ChirpStack permite mostrar de forma clara y concisa los datos recopilados por los dispositivos IoT, lo que facilita el seguimiento del rendimiento y el estado de la red LoRaWAN.
- **Sistemas de notificación y alertas:** También puede integrarse con sistemas de notificación y alertas, como correos electrónicos, mensajes SMS o aplicaciones de mensajería instantánea. De esta manera, los usuarios pueden recibir notificaciones en tiempo real sobre eventos importantes, como fallas críticas, cambios de estados o umbrales predefinidos alcanzados por los dispositivos IoT. Uno de las integraciones disponibles es HTTP, cuando se configura la integración de HTTP en ChirpStack, se pueden definir ciertos eventos o condiciones que, cuando ocurren, activan una solicitud HTTP a la URL especificada. Si ChirpStack detecta una condición específica, como una falla en el dispositivo o la superación de un umbral predeterminado, puede enviar una alerta o notificación a través de una solicitud HTTP a una URL de destino y este a su vez, alertar a los usuarios.
- **Plataformas de automatización y control:** ChirpStack puede integrarse con sistemas de automatización y control, permitiendo a los dispositivos IoT interactuar con otros

dispositivos o aplicaciones para realizar acciones específicas en función de los datos recopilados. Esto habilita la creación de soluciones más complejas y potentes basadas en la automatización. Un ejemplo de aplicación para esta categoría es Node-RED, una herramienta de código abierto para la creación de flujos de trabajo y automatización. Al integrar ChirpStack con Node-RED, los datos capturados por los dispositivos IoT pueden utilizarse para activar acciones específicas, como enviar notificaciones, actualizar bases de datos o controlar otros dispositivos conectados.

ChirpStack ofrece una amplia gama de posibilidades de integración con diversas aplicaciones, y los ejemplos mencionados son solo una muestra de su versatilidad. Esta plataforma demuestra su capacidad para adaptarse a diferentes casos de uso, especialmente en la monitorización de reservas naturales de diversas índoles. La posibilidad de conectarse con varias herramientas y servicios permite que ChirpStack se ajuste perfectamente a las necesidades específicas de cada proyecto, facilitando la gestión y análisis de datos de dispositivos IoT en entornos naturales protegidos. Con este Network Server, es posible crear soluciones personalizadas y efectivas para monitorear y proteger reservas naturales, aprovechando al máximo la información proporcionada por los dispositivos de la red LoRaWAN.

4.6. SOSTENIBILIDAD

En el contexto de la creciente preocupación por el cuidado del medio ambiente y la conservación de la biodiversidad, las tecnologías sostenibles se han convertido en una pieza fundamental para el manejo responsable de los recursos naturales. Las arquitecturas basadas en LoRa han ganado relevancia en el ámbito de la inteligencia ambiental debido a sus numerosas ventajas, desde su eficiente consumo de energía hasta su capacidad de transmisión de datos a larga distancia. Estas características permiten el despliegue de una red de sensores ambientales distribuidos en áreas extensas, sin la necesidad de una infraestructura compleja y costosa. Al reducir la huella de infraestructura, se minimiza el impacto humano directo en la reserva y se evitan perturbaciones en la vida silvestre.

La sostenibilidad es una parte esencial de esta arquitectura, ya que sus dispositivos de baja potencia de transmisión y su prolongada vida útil de batería o uso de energía solar contribuyen a la conservación de los recursos naturales y a la reducción de residuos electrónicos. Además, el monitoreo remoto posibilitado por LoRaWAN y AllLoRa evita el desplazamiento constante de personal en el terreno, garantizando una gestión más responsable y eficiente de la reserva.

Una arquitectura basada en redes LoRa es sostenible debido a que utiliza dispositivos con bajo consumo de energía, lo que permite larga duración de las baterías o uso de energía solar. Esto reduce la necesidad de reemplazar frecuentemente las baterías y disminuye el impacto ambiental asociado con la eliminación de baterías usadas. Además, la comunicación a larga distancia reduce la necesidad de repetidores y disminuye la infraestructura requerida.

Esto facilita el monitoreo remoto, evitando perturbaciones en la vida silvestre y reduciendo el impacto humano en la reserva. La arquitectura presentada es aplicable en áreas rurales y remotas, donde hay poca infraestructura disponible. Con un monitoreo continuo y preciso de datos ambientales, es posible tomar decisiones informadas para una gestión más efectiva y sostenible de la reserva. Asimismo, la baja potencia de transmisión de las redes LoRa reduce la interferencia con la vida silvestre y hábitats. Su flexibilidad dan con una solución duradera y respetuosa con el medio ambiente a largo plazo.

CAPÍTULO 5

CONCLUSIONES

5.1. CONCLUSIONES GENERALES

A lo largo de esta investigación, se ha explorado en profundidad la importancia y relevancia de implementar tecnologías avanzadas en la protección de los ecosistemas de Chile. El enfoque ha sido desarrollar una solución integral que aproveche la potencia de IoT y las redes de largo alcance para brindar un sistema que logre ayudar a cuidar las reservas naturales, aportando al desarrollo de la Inteligencia Medioambiental.

5.1.1. RESULTADOS GENERALES

En primer lugar, se procedió al diseño del sistema, teniendo en cuenta las particularidades y necesidades específicas de las reservas naturales. Se identificaron las variables ambientales críticas que requerían monitoreo continuo, así como los dispositivos y sensores más adecuados para capturar estos datos de manera precisa y confiable. Luego, el sistema fue implementado en la Universidad Politécnica de Valencia, donde se logró la conexión buscada entre gateway y los sensores elegidos. Por último, el sistema fue analizado en términos de cobertura, escalabilidad, sostenibilidad, posibilidad de integración con otras aplicaciones y se realizó un análisis de costo en un primer año de uso.

Dentro de las conclusiones más importantes obtenidas al momento de realizar la evaluación de las diversas categorías, se pueden destacar las siguientes:

- La cobertura dada por el sistema es aceptable para el sistema propuesto (un gateway y una pequeña cantidad de sensores), pero se debería mejorar en trabajos futuros y para tener resultados más exactos se debería probar el sistema en un ambiente real, midiendo la diferencia del alcance de la red en diferentes ecosistemas.
- La escalabilidad del sistema se ve limitada por el ciclo de trabajo de la red, pero la arquitectura de LoRaWAN permite que esta no afecte a la cobertura del sistema.
- Incluir AllLoRa junto con LoRaWAN es una innovación que logra aportar a la cobertura, dándole un valor agregado al sistema. Con AllLoRa es posible llegar a largas distancias y transferir datos de gran peso, algo que se dificultaba utilizando solo LoRaWAN.
- LoRaWAN es una herramienta muy popular hoy en día y, basado en lo investigado en este trabajo, ha demostrado cumplir con las expectativas iniciales de esta memoria.

5.1.2. LIMITACIONES

Cómo el área de inteligencia ambiental está en desarrollo, todavía existe espacio para la mejora en varios aspectos, sobretodo en la escalabilidad de LoRaWAN. Es uno de los limitantes del sistema saber que no es posible agregar todos los gateways y sensores que se requieran, es por esto que es necesario continuar con la investigación y lograr avances significativos en esta área. Dentro de las limitaciones de la evaluación, no fue posible realizar pruebas en un espacio real, esto resultó en que la evaluación del sistema fuera más teórica que práctica.

5.1.3. PRINCIPALES DESAFÍOS

Dentro de los principales desafíos enfrentados durante el desarrollo de la memoria se pueden destacar tres:

- El primer desafío fue el llegar a un conocimiento base de lo que ya existe para así poder innovar. Existe una gran cantidad de trabajos que involucran LoRaWAN, por lo que encontrar una manera distinta de realizar un sistema de monitoreo fue uno de los desafíos encontrados al iniciar este trabajo.
- El segundo desafío corresponde a la implementación física del sistema. Si bien los dispositivos fueron dados por la Universidad Politécnica de Valencia, se tuvo un tiempo limitado para realizar la implementación, por lo que era necesario actuar con rapidez y efectividad para lograr aprovechar el tiempo dado.
- Por último, realizar una evaluación teórica fue un desafío debido a que se requirió una extensiva búsqueda de información y de lectura de varios trabajos para lograr una evaluación extensiva, por lo que fue un desafío que no se esperaba al iniciar este trabajo.

5.2. RESULTADOS PRINCIPALES

5.2.1. OBJETIVOS SECUNDARIOS

1. Investigar el estado del arte de los distintos sistemas disponibles para el monitoreo de áreas verdes.

El estudio del estado del arte de los distintos sistemas disponibles fue esencial para el diseño del sistema propuesto. Este análisis sirvió para plantear las bases del sistema y aportar con las ventajas y desventajas de lo elegido. Este análisis se puede ver reflejado en la amplia bibliografía requerida para la redacción de esta memoria. Es por esto que se considera este objetivo como cumplido.

2. Diseñar una arquitectura Edge-LoRa junto con IoT para el monitoreo de distintos tipos de parámetros dentro de reservas naturales.

Este objetivo se encuentra directamente relacionado con el objetivo general de esta memoria. Se diseñó un sistema Edge, dónde se utilizó como conectividad dos redes basadas en LoRa, LoRaWAN y AllLoRa. Fue presentada la arquitectura del sistema propuesto además de distintas opciones para la implementación. Es por esto que este objetivo se considera cumplido.

3. Implementar la arquitectura de manera modular para habilitar la comunicación y procesamiento de datos.

El sistema fue implementado usando un gateway RAK7391 y tres tipos distintos de sensores, un Arduino Nicla Vision con la capacidad de grabar videos y enviarlos mediante LoRaWAN, un Arduino Portenta H7 junto con un Visual Shield LoRa que es capaz de monitorear aspectos del medioambiente como la temperatura o la humedad y enviarlo también utilizando LoRaWAN, y por último, un LoPy 4 de Pycom capaz de medir aspectos del medioambiente y enviarlos al gateway mediante AllLoRa. Es por esto que se considera este objetivo como cumplido.

4. Analizar eficacia en el sistema usando dos o tres sensores conectados al gateway con LoRa.

El sistema fue evaluado en base a la cobertura, la escalabilidad, la sostenibilidad, la capacidad de integrarse con otras aplicaciones y el costo. Se demostró que la capacidad de integración le da flexibilidad y la capacidad de adaptarse a distintas situaciones. Combinar LoRaWAN con AllLoRa logra que el sistema tenga una cobertura deseable para un sistema de monitoreo de reservas naturales y la robustez del sistema se ve reflejada en la manera en que ambos tipos de conectividad se potencian entre sí, dando con un sistema eficaz en la tarea a cumplir. Es por esto que este objetivo se considera como cumplido.

5.2.2. OBJETIVO PRINCIPAL

- Diseñar, implementar y analizar un sistema IoT sostenible para monitorear reservas naturales.

Considerando todo lo mencionado anteriormente, es que se considera este objetivo como cumplido. Fue posible, diseñar una arquitectura basada en redes LoRa, implementar esta en su forma más básica para poder demostrar que es posible la integración de dos redes LoRa en una misma arquitectura y fue analizado el sistema propuesto en los aspectos claves, considerando las ventajas y desventajas de lo planteado.

5.3. TRABAJOS FUTUROS

Esta memoria fue implementada bajo el contexto del proyecto Chan-Twin, un proyecto que busca generar un gemelo digital de las Lagunas de la Mata-Torrevieja. La implementación en un ecosistema real será llevada a cabo en septiembre 2023, por lo que pronto será posible conocer los alcances reales del sistema. Se tendrá que realizar una nueva evaluación del sistema considerando las distintas variables a las que fue creado, además de contar con nuevas funcionalidades que se agregarán al sistema por los encargados del proyecto.

De todas maneras, se mencionan los distintos aspectos en los que podría ser interesante desarrollar en un futuro bajo el contexto de este trabajo:

5.3.1. INTELIGENCIA ARTIFICIAL

Cómo uno de los principales ángulos que no fueron abordados en este trabajo, está la Inteligencia Artificial (IA). Se podrían implementar técnicas de TinyML dentro de los sensores descritos para así llevar el análisis de los datos obtenidos a los bordes de la red. Con esto se lograría un sistema que presente no solo la capa de monitoreo, si no que también la capa de análisis de datos dentro del mismo dispositivo.

5.3.2. AlloRa

Dentro de este trabajo se discute el uso de AlloRa, pero cómo este protocolo sigue en desarrollo, en un futuro podría tener mejoras considerables que lleven a otro nivel la manera de comunicar los dispositivos. Podría resultar interesante plantear un sistema que solo utilice AlloRa cómo protocolo de comunicación.

5.3.3. ESCALABILIDAD

Una de las principales limitantes de este sistema es la posibilidad de escalar, por lo que se podría investigar sobre nuevos alcances o técnicas para mejorar este aspecto. Sería interesante que la escalabilidad no fuera una limitante del sistema, así lograr monitorear un ecosistema de manera más extenuante.

ANEXOS

1. CÓDIGO ARDUINO PORTENTA H7

```
import machine
import time
import ubinascii
from lora import *
from cayennelpp import LppFrame
from machine import ADC

# Configuración del sensor de temperatura
adc = ADC(machine.Pin("A0"))

# Configuración de LoRaWAN
lora = Lora(band=BAND_EU868, poll_ms=60000, debug=False)
appEui = "0000000000000000"
appKey = "36acb008a4813d6a7f2a1d0fad42ce2a"
devEui = lora.get_device_eui()

try:
    lora.join_OTAA(appEui, appKey, timeout=10000)

except LoraErrorTimeout as e:
    print("ErrorTimeout:", e)
except LoraErrorParam as e:
    print("ErrorParam:", e)

# Medir la temperatura

raw_val = adc.read_u16()
voltage = raw_val * 3.3 / 4095 # Convertir valor crudo a voltaje
temperature = (voltage - 0.5) * 100

# Enviar paquete de datos a LoRaWAN

lora.send_data('Datos enviados: temperatura = {}°C'.format(temperature))
print('Datos enviados: temperatura = {}°C'.format(temperature))
```

2. CÓDIGO DE DECODIFICACIÓN DE MENSAJES EN EL GATEWAY

```
function hex_to_ascii(str1) {
  var hex = str1.toString();
  var str = '';
  for (var n = 0; n < hex.length; n += 2) {
    str += String.fromCharCode(parseInt(hex.substr(n, 2), 16));
  }
  return str;
}

function toHexString(bytes) {
  return bytes.map(function(byte) {
    return ("00" + (byte & 0xFF).toString(16)).slice(-2)
  }).join('')
}

function Decode(fPort, bytes) {

  var tohex = toHexString(bytes);
  var toascii = hex_to_ascii(tohex);
  return {
    "message": toascii
  };
}
```

3. PUERTOS CONECTADOS EN GATEWAY

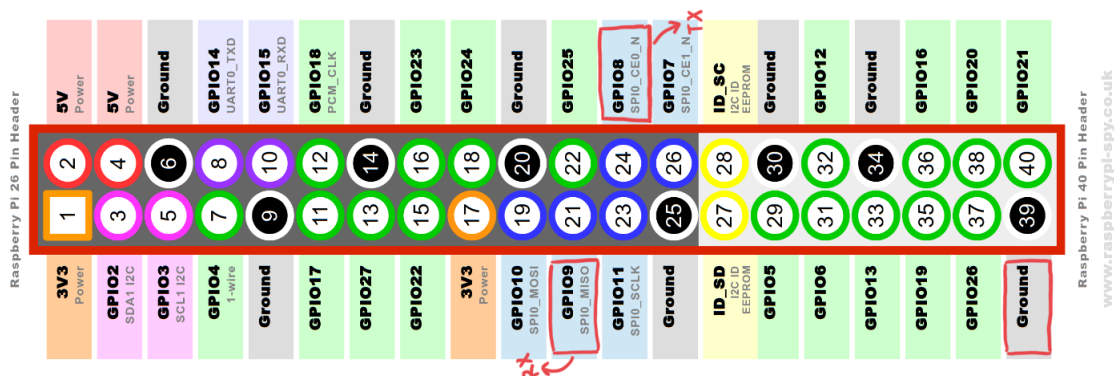


Figura 30: Puertos Seriales conectados en el gateway RAK7391.
Fuente: Creación propia.

4. PUERTOS CONECTADOS EN HELTEC WIFI LORA 32

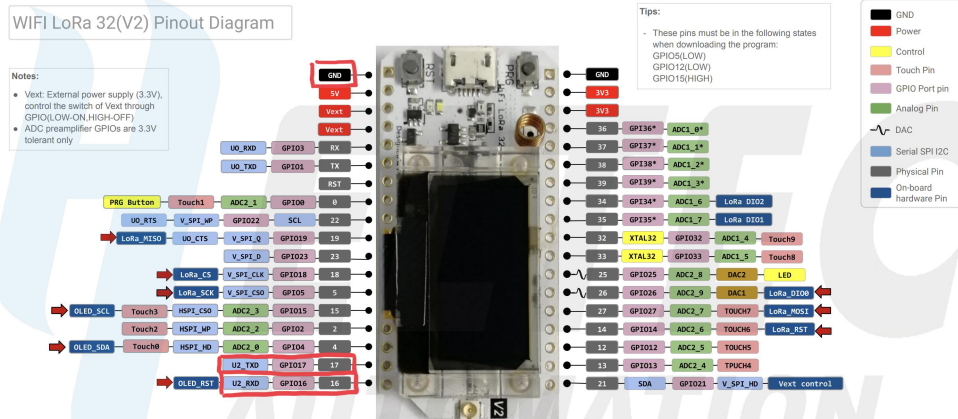


Figura 31: Puertos Seriales conectados en Heltec WiFi LoRa 32.
Fuente: Creación propia.

5. CÓDIGO A AGREGAR EN EL ARCHIVO CONFIG.TXT DEL GATEWAY

```
enable_uart=1
dtoverlay=uart0
dtoverlay=uart1
dtoverlay=uart2
dtoverlay=uart3
dtoverlay=uart4
```

6. PREPARACIÓN DE CONTENEDOR DE DOCKER DE ALLORA

requirements.txt

```
AllLoRa==1.1.0
pyserial==3.5
```

dockerfile

```
FROM python:3.9
WORKDIR /allora
COPY . /allora
```

```
RUN pip install --no-cache-dir -r requirements.txt
```

```
CMD [ "python", "main.py" ]
```

Comando para construir la imagen del contenedor:

```
docker build -t nombre_imagen .
```

docker_compose.yml

```
version: "3.9"
services:
  allora:
    image: allora
    restart: always
    working_dir: /app/
    volumes:
      - shared_volume:/app/
    devices:
      - /dev/ttyACM0:/dev/ttyACM0
```

Comando para iniciar el contenedor en segundo plano:

```
docker-compose up -d
```

7. CÓDIGO DE PYCOM LOPY

```
import pycom

from AllLoRa.Nodes.Sender import Sender
from AllLoRa.Connectors.LoPy4_connector import LoPy4_connector
from AllLoRa.File import CTP_File
from time import sleep

# For testing
sizes = [1, 2, 4, 8, 16, 32, 64, 128, 256, 512] #, 1024
file_counter = 0

def clean_timing_file():
```

```
test_log = open('log.txt', "wb")
test_log.write("")
test_log.close()

if __name__ == "__main__":

# First, we set the connector
connector = LoPy4_connector()

# Then, we set up our Sender Node
lora_node = Sender(connector, config_file = "LoRa.json")

# We turn on a led for a second to know that we are doing ok...
pycom.rgbled(0x1aa7ec) # Picton Blue
sleep(1)
pycom.rgbled(0) # off

chunk_size = lora_node.get_chunk_size()
    # We use it to create the files to be sent...

try:
clean_timing_file()
backup = lora_node.establish_connection()
print("Connected!")

# This is how to handle a backup file if needed
if backup:
print("Asking backup")
#file = Datasource.get_backup()
#lora_node.restore_file(file)

# With an established connection, we start sending data periodically
while True:
if not lora_node.got_file():
n = file_counter % len(sizes)
file_counter += 1
size = sizes[n]
print("Setting file")
pycom.rgbled(0xd74894) # Kirby Pink.

file = CTP_File(name = '{}.json'.format(size),
content = bytearray('{}'.format(n%10)*(1024 * size)),
chunk_size=chunk_size)
```

```
lora_node.set_file(file)

print("New file set, ", file.get_name())
pycom.rgbled(0) # LED off

lora_node.send_file()

except KeyboardInterrupt as e:
print("THREAD_EXIT")
```

REFERENCIAS BIBLIOGRÁFICAS

- [Adelantado *et al.*, 2017] Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., y Watteyne, T. (2017). Understanding the limits of lorawan. *IEEE Communications Magazine*, 55(9):34–40.
- [Agency, 2023] Agency, E. E. (2023). 2030 climate target plan.
- [Ahmed *et al.*, 2022] Ahmed, M. A., Gallardo, J. L., Zuniga, M. D., Pedraza, M. A., Carvajal, G., Jara, N., y Carvajal, R. (2022). Lora based iot platform for remote monitoring of large-scale agriculture farms in chile. *Sensors*.
- [Alil y Alil, 2020] Alil, Z. y Alil, H. (2020). Towards sustainable smart iot applications architectural elements and design: opportunities, challenges, and open directions. *The Journal of Supercomputing*.
- [Anjum *et al.*, 2019] Anjum, M., Khan, M. A., Hassan, S. A., Mahmood, A., y Gidlund, M. (2019). Analysis of rssi fingerprinting in lora networks. En *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1178–1183, Tangier, Morocco.
- [Ansah *et al.*, 2020] Ansah, M. R., Sowah, R. A., Melià-Seguí, J., Katsriku, F. A., Vilajosana, X., y Banahene, W. O. (2020). Characterising foliage influence on lorawan pathloss in a tropical vegetative environment. *IET Wireless Sensor Systems*, 10(5):285–290.
- [Arduino®, 2019] Arduino® (2019). Arduino® nano 33 ble datasheet.
- [Arduino®, 2022] Arduino® (2022). Arduino® portenta h7 collective datasheet.
- [Arduino®, 2023] Arduino® (2023). Arduino® nicla vision datasheet.
- [Arratia *et al.*, 2023a] Arratia, B., García-Guillamón, P., Calafate, C. T., Cano, J.-C., Cecilia, J. M., y Manzoni, P. (2023a). A modular and mesh-capable lora based content transfer protocol for environmental sensing. *IEEE 20th Consumer Communications & Networking Conference*.
- [Arratia *et al.*, 2023b] Arratia, B., Rosas, E., Calafate, C. T., Cano, J.-C., Cecilia, J. M., y Manzoni, P. (2023b). Allora: Empowering environmental intelligence through an advanced lora-based iot solution. *IEEE Communications Magazine*.
- [Automation®, 2020] Automation®, H. (2020). Wifi lora 32 (v3).
- [Avila-Campos *et al.*, 2019] Avila-Campos, P., Astudillo-Salinas, F., Vazquez-Rodas, A., y Araujo, A. (November 2019). The impact of landscape on the coverage of lorawan. En *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '19)*, pp. 199–206.

- [Bansal *et al.*, 2020] Bansal, M., Kumar, A., y Virmani, A. (2020). Green iot: Current scenario & future prospects. *Journal of Trends in Computer Science and Smart Technology*.
- [Benkahla *et al.*, 2018] Benkahla, N., Tounsi, H., Song, Y.-Q., y Frikha, M. (2018). Enhanced dynamic duty cycle in LoRaWAN network. En *Ad-hoc, Mobile, and Wireless Networks*, volumen 11104, pp. 147–162.
- [Bor *et al.*, 2016] Bor, M., Vidler, J., y Roedig, U. (2016). Lora for the internet of things. *Lancaster University*.
- [Bouguera *et al.*, 2018] Bouguera, T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., y Andrieux, G. (2018). Energy consumption model for sensor nodes based on lora and lorawan. *Sensors*, 18(7).
- [ChirpStack, 2023] ChirpStack (2023). Chirpstack architecture.
- [Data, 2023] Data, I. (2023). Influx data pricing. <https://www.influxdata.com/influxdb-pricing/>. Fecha de acceso: 20 Julio 2023.
- [Davcev *et al.*, 2018] Davcev, D., Mitreski, K., Trajkovic, S., Nikolovski, V., y Koteli, N. (2018). lot agriculture system based on lorawan. En *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–4.
- [Docker, 2013] Docker (2013). What is a container?
- [Dragino, 2023a] Dragino (2023a). Dlos8n outdoor lorawan gateway.
- [Dragino, 2023b] Dragino (2023b). Dlos8n outdoor lorawan gateway datasheet.
- [Electronics, 2023] Electronics, M. (2023). Lopy 4.0. Disponible en: https://www.mouser.cl/ProductDetail/Pycom/LoPy-4.0?q_s=MLItCLRbWsz2Ys%252BwjFa7JA%3D%3D&_gl=1*1odhum5*_ga*dW5kZWZpbmVk*_ga_15W4STQT4T*dW5kZWZpbmVk*_ga_1KQLCYKRX3*dW5kZWZpbmVk. Fecha de acceso: 18 Julio 2023.
- [Finnegan *et al.*, 2018] Finnegan, J., Brown, S., y Farrell, R. (2018). Evaluating the scalability of lorawan gateways for class b communication in ns-3. En *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6.
- [Forestal, 2023] Forestal, C. N. (2023). Arduino® portenta vision shield lora® datasheet.
- [Furstenau *et al.*, 2022] Furstenau, L. B., Rodrigues, Y. P. R., Sott, M. K., Leivas, P., Dohan, M. S., López-Robles, J. R., Cobo, M. J., Bragazzi, N. L., y Choo, K.-K. R. (2022). Internet of things: Conceptual network structure, main challenges and future directions. *Digital Communications and Networks*.
- [Haxhibeqiri *et al.*, 2018] Haxhibeqiri, J., De Poorter, E., Moerman, I., y Hoebeke, J. (2018). A survey of lorawan for iot: From technology to application. *Sensors*.

- [Industries, 2022] Industries, T. T. (2022). Start benefiting from the enterprise features of the things stack.
- [Industries, 2023] Industries, T. T. (2023). Abp vs otaa.
- [Klimiashvili et al., 2020] Klimiashvili, G., Tapparello, C., y Heinzelman, W. (2020). Lora vs. wifi ad hoc: A performance analysis and comparison. *International Conference on Computing, Networking and Communications*.
- [Kufakunesu et al., 2020] Kufakunesu, R., Hancke, G. P., y Abu-Mahfouz, A. M. (2020). A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges. *Sensors*, 20(18):5044.
- [Kumar et al., 2019] Kumar, S., Tiwari, P., y Zymbler, M. (2019). Internet of things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*.
- [Kunz y Karpinska, 2021] Kunz, M. y Karpinska, D. (2021). The analysis of the visibility and signal strength of the lorawan network in an urbanized area: A case study of the bielany campus at the nicolaus copernicus university in toruń. *Bulletin of Geography. Socio-economic Series*, pp. 137–149. [Accessed 24 July 2023].
- [Kurose y Ross, 2016] Kurose, J. F. y Ross, K. W. (2016). *Computer Networking: A Top-Down Approach*. Pearson, Boston, MA, 8 edición.
- [Mason et al., 2020] Mason, F., Chiariotti, F., Capuzzo, M., Magrin, D., Zanella, A., y Zorzi, M. (2020). Combining lorawan and a new 3d motion model for remote uav tracking. En *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 412–417.
- [Mekki et al., 2019] Mekki, K., Bajic, E., Chaxel, F., y Meyer, F. (2019). A comparative study of lpwan technologies for large-scale iot deployment. *CT Express*.
- [Mendes et al., 2022] Mendes, B., Passos, D., y Correia, N. (2022). Coverage characterization of lorawan sensor networks for citrus orchard monitoring. En *2022 International Young Engineers Forum (YEF-ECE)*, pp. 86–91.
- [Mohandass et al., 2021] Mohandass, S., Sridevi, S., y Sathyabama, R. (2021). Animal health monitoring and intrusion detection system based on lorawan. *Turkish Journal of Computer and Mathematics Education*, 12(2):2397–2403.
- [Na et al., 2017] Na, S., Hwang, D., Shin, W., y Kim, K.-H. (2017). Scenario and countermeasure for replay attack using join request messages in lorawan. En *2017 International Conference on Information Networking (ICOIN)*, pp. 718–720.
- [Ogbodo et al., 2022] Ogbodo, E. U., Abu-Mahfouz, A. M., y Kurien, A. M. (2022). A survey on 5g and lpwan-iot for improved smart cities and remote area applications: From the aspect of architecture and security. *Sensors*, 22(16).

- [Pi[®], 2019] Pi[®], R. (2019). Raspberry pi 4.
- [Pycom, 2018] Pycom (2018). Lopy4 datasheet.
- [RAKWireless[®], 2022a] RAKWireless[®] (2022a). Rak5146 wislink lpwan concentrator datasheet.
- [RAKWireless[®], 2022b] RAKWireless[®] (2022b). Rak7391 wisgate connect datasheet.
- [Raza *et al.*, 2017] Raza, U., Kulkarni, P., y Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873.
- [Sadeeq *et al.*, 2021] Sadeeq, M., Abdulkareem, N. M., Zeebaree, S. R. M., Ahmed, D. M., Sami, A. S., y Zebari, R. R. (2021). Iot and cloud computing issues, challenges and opportunities: A review. *Qubahan Academic Journal*, 1(2):1–7.
- [Sanchez-Iborra *et al.*, 2019] Sanchez-Iborra, R., Liaño, I. G., Simoes, C., Couñago, E., y Skarmeta, A. F. (2019). Tracking and monitoring system based on lora technology for light-weight boats. *Electronics*, 8(15).
- [Schwartz, 2001] Schwartz, S. M. (2001). Frequency hopping spread spectrum (fhss) vs. direct sequence spread spectrum (dsss) in broadband wireless access (bwa) and wireless lan (wlan). *Director*.
- [Semtech, 2020] Semtech (2020). Sx1302 lora gateway baseband processor datasheet.
- [Semtech, 2023] Semtech (2023). Lora layer system.
- [Stack, 2020] Stack, T. T. (2020). The things stack documentation.
- [Stipanicev *et al.*, 2007] Stipanicev, D., Bodrozic, L., y Stula, M. (2007). Environmental intelligence based on advanced sensor networks. *IEEE*.
- [Store, 2023a] Store, A. (2023a). Nicla vision. Disponible en: <https://store-usa.arduino.cc/products/nicla-vision?selectedStore=us>. Fecha de acceso: 18 Julio 2023.
- [Store, 2023b] Store, A. (2023b). Portenta h7. Disponible en: <https://store-usa.arduino.cc/products/portenta-h7?selectedStore=us>. Fecha de acceso: 18 Julio 2023.
- [Taehong *et al.*, 2021] Taehong, K., Seong-eun, Y., y Youngsoo, K. (2021). Edge/fog computing technologies for iot infrastructure. *Sensors*.
- [Talend, 2023] Talend (2023). Salario medio para técnico informático en chile 2023. <https://cl.talent.com/salary?job=t%C3%A9cnico+inform%C3%A1tico#:~:text=El%20salario%20t%C3%A9cnico%20inform%C3%A1tico%20promedio,m%C3%A1s%20experimentados%20perciben%20hasta%20%248.400>. Fecha de acceso: 24 Julio 2023.

- [Thomas y Eldhose, 2020] Thomas, A. y Eldhose, N. V. (2020). Lorawan scalability analysis - co spreading factor interference. *International Journal of Computer Networks & Communications*, 12(1):65-81.
- [Vangelista, 2017] Vangelista, L. (2017). Frequency shift chirp modulation: The lora modulation. *IEEE SIGNAL PROCESSING LETTERS*.
- [Wang et al., 2020] Wang, J., Meneses, E., y Wanderley, M. M. (2020). The scalability of wifi for mobile embedded sensor interfaces. *Royal Birmingham Conservatoire*.
- [Wireless, 2023a] Wireless, R. (2023a). Rak7249 wisgate edge max documentation center.
- [Wireless, 2023b] Wireless, R. (2023b). Rak7391 compute module 4 (cm4) carrier board. Disponible en: <https://store.rakwireless.com/products/wisgate-connect-base-kit-rak7391?variant=42715873542342>. Fecha de acceso: 18 Julio 2023.