



DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y
COMPUTADORES

Diseño de un sistema de mensajería instantánea para redes comunitarias

Por

Miguel Kiyoshy Nakamura Pinto

Directores:

Dr. Pietro Manzoni

Dr. Marco Zennaro



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Valencia, España

Septiembre, 2022

A mi familia.

Son lo más importante en mi vida.

La mente que se abre a una nueva idea,
jamás volverá a su tamaño original.

Albert Einstein.

Agradecimientos

EN primer lugar, a mi director de tesis y coordinador del Grupo de Redes de Computadores (GRC), **Pietro Manzoni**, quien desde el primer contacto vía email me abrió las puertas de este increíble grupo, especialmente agradecerle por guiarme en el proceso de investigación durante estos años, por el apoyo, y pese a que algunas veces puse a prueba su paciencia, nunca dejo de alentarme y retarme a alcanzar los objetivos. Sobre todo, gracias por la confianza y la amistad.

A mi co-director e investigador en el *International Centre for Theoretical Physics* (ICTP), Marco Zennaro, por el apoyo y guía durante la realización de esta tesis doctoral.

A los profesores Carlos Calafate, Juan Carlos Cano, Enrique Hernández y José María Cecilia, por todos los consejos para realizar un mejor trabajo, por compartir su experiencia y las palabras de aliento durante las reuniones mensuales.

A todos mis compañeros y miembros del GRC por el apoyo en los momentos difíciles, por siempre estar dispuestos a compartir conocimiento y colaborar, por ser un increíble grupo de estudiantes que me hicieron disfrutar todo el proceso, sobre todo por las tardes de cervezas dentro y fuera del laboratorio.

Especialmente a mis padres y hermanos, Miguel, Betty, Haru y Yoshio, quienes siempre estuvieron motivándome y alentándome sin importar la distancia. A mi familia y amigos que siempre estuvieron atentos y en contacto a pesar de la diferencia horaria.

A mis familiares y amigos que trascendieron y no pude estar para despedirme, siempre estarán en mi memoria.

Gracias también a las personas que me apoyaron y acompañaron durante el proceso de postulación a la beca.

Finalmente, al Gobierno de los Estados Unidos Mexicanos a través del Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Gobierno de Chiapas mediante el Consejo de Ciencia y Tecnología del Estado de Chiapas (COCYTECH) por brindarme la oportunidad de mejoramiento profesional y académico por medio del programa de becas “CONACYT - Gobierno del Estado de Chiapas” en la convocatoria del 2017.

Miguel Kiyoshy Nakamura Pinto
Valencia, 3 de octubre de 2022

Resumen

CERCA de 3.000 millones de personas en todo el mundo no pueden aprovechar ni siquiera los servicios de conectividad más básicos, ya que la mayoría de ellas viven en zonas rurales o países en vías de desarrollo. Incluso los servicios de mensajería más simples serían de gran ayuda, por ejemplo, para los agricultores que desean conocer el precio de las mercancías que les interesa vender o comprar antes de decidir si se emprende un viaje posiblemente largo, caro y agotador.

La tecnología LoRa permite realizar enlaces de larga distancia con un consumo reducido de energía a bajo coste, siendo su principal limitación el escaso ancho de banda que ofrece. Con LoRa, los lugares remotos, como las zonas rurales, pueden beneficiarse de servicios basados en la conectividad que, de otro modo, serían imposibles. Nuestra propuesta entra en la categoría de redes comunitarias, en las que los usuarios construyen su propia red cuando no hay infraestructura comercial disponible. Además de la simple aplicación de mensajería, LoRa puede utilizarse para distribuir información de sensores a las comunidades o para proporcionar alertas de desastres o datos meteorológicos.

Presentamos un protocolo flexible basado en la tecnología LoRa que permite la transferencia de “contenido” denominado LoRaCTP, el cual proporciona los mecanismos necesarios para que LoRa sea fiable, introduciendo una configuración de conexión ligera. Hemos diseñado este protocolo también como soporte de comunicación para las soluciones IoT basadas en *edge computing*, dada su estabilidad, el bajo consumo de energía y la posibilidad de cubrir largas distancias.

Integramos una arquitectura que permite la recolección de datos de fuentes externas genéricas. Especialmente las fuentes de audio, apuntando a dos servicios básicos: un sistema de mensajería de voz que permite a los usuarios que no saben leer o escribir, realizar notas de voz, y un servicio de compresión de audio para extraer las principales características de la entrada de audio y utilizarla para desarrollar análisis de audio inteligente basado en *Machine Learning*.

Combinamos IoT y *Edge computing* en un enfoque de innovación frugal, para proponer una solución “ingeniosa”, en términos de utilización mínima de recursos y sostenibilidad, para construir un soporte básico para un sistema IoT en áreas rurales. Incluimos un proxy *MQTT* para integrar dispositivos de bajo coste y bajo consumo en un sistema de mensajería basado en LoRa. A través de una interfaz *REST*, mostramos cómo se puede usar nuestra plataforma para distribuir información de sensores de las comunidades rurales. Además, *MQTT* permite que estos datos se proporcionen a “lagos de datos” externos para que puedan usarse para tareas tales como informes, visualización, análisis avanzado y aprendizaje automático.

Describimos una arquitectura genérica de *edge/fog*, utilizando microservicios, un sistema basado en *MQTT* que puede recopilar datos de ingreso, manejar su persistencia y coordinar la integración de datos con la nube utilizando un servicio específico llamado agregador. Las estaciones *edge* tienen un canal dedicado con el agregador que se basa en LoRa para permitir transmisiones de largo alcance con bajo consumo de energía.

Abstract

N Early 3 billion people around the world are unable to take advantage of even the most basic connectivity services, as most of them live in rural areas or developing countries. Even the simplest messaging services would be of great help, for example, to farmers who want to know the price of goods they are interested in selling or buying before deciding whether to embark on a possibly long, expensive and exhausting journey.

LoRa technology enables long-distance links with reduced power consumption at low cost, its main limitation being the low bandwidth it offers. With LoRa, remote locations, such as rural areas, can benefit from connectivity-based services that would otherwise be impossible. Our proposal falls into the category of community networks, where users build their own network when commercial infrastructure is not available. In addition to a simple messaging application, LoRa can be used to distribute sensor information to communities or to provide disaster alerts or weather data.

We present a flexible protocol based on LoRa technology that enables the transfer of “content” called LoRaCTP, which provides the necessary mechanisms for LoRa to be reliable, introducing a lightweight connection setup. We have designed this protocol also as a communication support for IoT solutions based on edge computing, given its stability, low power consumption and the possibility of covering long distances.

Likewise, we integrated an architecture that allows data collection from generic external sources. Especially audio sources, targeting two basic services: a voice messaging system that allows users who cannot read or write to make voice notes, and an audio compression service to extract the main features of the audio input and use it to develop intelligent audio analytics based on Machine Learning.

We combine IoT and Edge computing in a frugal innovation approach, to propose an “ingenious” solution, in terms of minimum resource utilization and sustainability, to build a basic support for an IoT system in rural areas. We include an MQTT proxy to integrate low-cost and low-power devices into a LoRa-based messaging system. Through a REST interface, we show how our platform can be used to distribute sensor information from rural communities. In addition, MQTT allows this data to be provided to external “data lakes” so that it can be used for tasks such as reporting, visualization, advanced analytics, and machine learning.

We describe a generic edge/fog architecture, using microservices, an MQTT-based system that can collect ingress data, manage its persistence, and coordinate data integration with the cloud using a specific service called an aggregator. The edge stations have a dedicated channel with the aggregator that is based on LoRa to enable long-range transmissions with low power consumption.

Resum

Prop de 3.000 milions de persones a tot el món no poden aprofitar ni tan sols els serveis de connectivitat més bàsics, ja que la majoria d'elles viuen en zones rurals o països en vies de desenvolupament. Fins i tot els serveis de missatgeria més simples serien de gran ajuda, per exemple, per als agricultors que desitgen conèixer el preu de les mercaderies que els interessa vendre o comprar abans de decidir si s'emprèn un viatge possiblement llarg, car i esgotador.

La tecnologia LoRa permet realitzar enllaços de llarga distància amb un consum reduït d'energia a baix cost, sent la seua principal limitació l'escassa amplada de banda que ofereix. Amb LoRa, els llocs remots, com les zones rurals, poden beneficiar-se de serveis basats en la connectivitat que, d'una altra manera, serien impossibles. La nostra proposta entra en la categoria de xarxes comunitàries, en les quals els usuaris construeixen la seua pròpia xarxa quan no hi ha infraestructura comercial disponible. A més de la simple aplicació de missatgeria, LoRa pot utilitzar-se per a distribuir informació de sensors a les comunitats o per a proporcionar alertes de desastres o dades meteorològiques.

Presentem un protocol flexible basat en la tecnologia LoRa que permet la transferència de “contingut” denominat LoRaCTP, el qual proporciona els mecanismes necessaris perquè LoRa siga fiable, introduint una configuració de connexió lleugera. Hem dissenyat aquest protocol també com a suport de comunicació per a les solucions IoT basades en edge computing, donada la seua estabilitat, el baix consum d'energia i la possibilitat de cobrir llargues distàncies.

Integrem una arquitectura que permet la recollida de dades de fonts externes genèriques. Especialment, les fonts d'àudio, apuntant a dos serveis bàsics: un sistema de missatgeria de veu que permet als usuaris que no saben llegir o escriure, realitzar notes de veu, i un servei de compressió d'àudio per a extraure les principals característiques de l'entrada d'àudio i utilitzar-la per a desenvolupar anàlisi d'àudio intel·ligent basat en Machine Learning.

Combinem IoT i Edge computing en un enfocament d'innovació frugal, per a proposar una solució “enginyosa”, en termes d'utilització mínima de recursos i sostenibilitat, per a construir un suport bàsic per a un sistema IoT en àrees rurals. Incloem un proxy MQTT per a integrar dispositius de baix cost i baix consum en un sistema de missatgeria basat en LoRa. A través d'una interfície REST, vam mostrar com es pot usar la nostra plataforma per a distribuir informació de sensors de les comunitats rurals. A més, MQTT permet que aquestes dades es proporcionen a “llacs de dades” externes perquè puguin usar-se per a tasques com ara informes, visualització, anàlisi avançada i aprenentatge automàtic.

Descrivim una arquitectura genèrica de edge/fog, utilitzant microserveis, un sistema basat en MQTT que pot recopilar dades d'ingrés, manejar la seua persistència i coordinar la integració de dades amb el núvol utilitzant un servei específic anomenat agregador. Les estacions edge tenen un canal dedicat amb el agregador que es basa en LoRa per a permetre transmissions de llarg abast amb baix consum d'energia.

Índice general

Agradecimientos	vii
Resumen	ix
Índice de figuras	xvii
Índice de tablas	xxi
Índice de listas	xxiii
1 Introducción	1
1.1 Motivación	3
1.2 Objetivos	4
1.3 Estructura de la Tesis	5
2 Redes Comunitarias, una visión general	7
2.1 Redes Comunitarias: Definición	7
2.2 Tecnologías LPWAN	8
2.3 WiFi	13
2.4 Resumen	19
3 Análisis del Problema	21
3.1 Educación e infraestructura	23
4 Tecnologías relacionadas	25
4.1 IoT	25
4.2 Tecnología LoRa y LoRaWAN	29
4.3 MQTT	36

4.4	Edge Computing	39
5	Solución propuesta	43
5.1	Protocolo de comunicaciones	43
5.2	Recolección de datos de fuentes externas	48
5.3	Nodos edge	49
6	Resultados	55
6.1	Sistema de mensajería	55
6.2	Integración con Telegram	62
6.3	Mensajes de voz	64
6.4	Recolección de datos de fuentes externas y proxy MQTT	71
6.5	Nodos FUDGE	81
7	Conclusiones, Publicaciones y Trabajos Futuros	87
	Acrónimos	92
	Bibliografía	97

Índice de figuras

1.1	Evolución de usuarios de Internet	2
1.2	Porcentaje de personas que utilizan Internet por ubicación	2
2.1	Cobertura geográfica del acceso inalámbrico	11
2.2	Enlace inalámbrico desde Trieste hasta Piancavallo, Italia	14
2.3	Curva de crecimiento anual de nodos operativos Guifi.net	16
2.4	Incremento anual de nodos operativos Guifi.net	16
2.5	Vehículos habilitados para DakNet	17
2.6	Arquitectura general DakNet	18
3.1	Número estimado de personas que utilizan/no utilizan internet en todo el mundo	21
3.2	Evolución del porcentaje de personas que nunca había accedido a Internet en España de 2010 a 2020	22
3.3	Porcentaje de analfabetismo a nivel global	23
4.1	Número de dispositivos IoT conectados a Internet en el mundo de 2015 a 2025	27
4.2	Estructura de un sistema IoT	27
4.3	<i>Stack</i> de la tecnología LoRaWAN	30
4.4	Comparación de Spreading Factors de LoRa: SF 7 a SF 12	32
4.5	Arquitectura de red LoRaWAN	33
4.6	Clasificación de los dispositivos finales LoRaWAN	35
4.7	Arquitectura de MQTT.	36
4.8	Secuencia de interacciones de publicación/suscripción.	37
4.9	Estructura de los MQTT topics.	37
4.10	Mensaje MQTT	38
4.11	Arquitectura tradicional IoT	40

4.12	Paradigma de computación en la nube	40
4.13	Paradigma de <i>Edge Computing</i>	41
4.14	Clasificación de plataformas <i>edge computing</i>	42
5.1	Esquema de protocolo <i>Stop-and-Wait</i>	44
5.2	Proceso de descubrimiento.	45
5.3	Estructura del paquete.	45
5.4	Diagrama del proceso de envío de mensajes.	46
5.5	Diagrama del proceso de recepción de mensajes.	48
5.6	Arquitectura general de la solución.	49
5.7	Diseño de arquitectura frugal.	50
5.8	Estructura FUDGE (frugal edge-node).	51
6.1	Arquitectura general del sistema de mensajería.	56
6.2	SSID AP.	57
6.3	Interfaz web de la aplicación de mensajería.	58
6.4	Páginas de envío de mensaje.	59
6.5	Páginas de mensajes recibidos y reenvío.	59
6.6	Prototipos de hub LoRa diseñados.	60
6.7	Proceso de descubrimiento del usuario; conjunto ordenado de resultados.	61
6.8	Arquitectura con integración de Telegram.	62
6.9	Prototipo hub gateway.	63
6.10	Interacción entre hub LoRa y hub gateway.	64
6.11	Arquitectura con mensajes de voz.	65
6.12	Interfaz gráfica para mensajes de voz.	65
6.13	Identificando usuarios.	66
6.14	Proceso de grabación del mensaje de voz.	66
6.15	Botón para reproducir el clip de voz (izquierda) y botón para enviar mensaje (derecha).	67
6.16	Tiempo de compresión (valores medios).	68
6.17	Esquema del auto-codificador de Hinton y Salakhutdinov utilizado.	69
6.18	Arquitectura del compresor de sonido autoencoder.	70
6.19	Forma de onda normalizada del original y de la predicción.	71
6.20	Configuración necesaria para los sensores y proxy MQTT.	72
6.21	Flujo de datos entre un sensor y un hub.	73
6.22	Proxy MQTT obteniendo datos del hub.	74
6.23	Evaluación de la escalabilidad del sistema aumentando el número de usuarios que envían simultáneamente.	74
6.24	Ubicación en Valencia para las pruebas de 750 m.	75
6.25	Ubicación en Chiapas, México para las pruebas de 6 km.	75
6.26	Escenario de pruebas en Chiapas, México.	76
6.27	Comportamiento del STT con un tamaño de mensaje variable (valores medios) utilizando SF 7.	77

6.28 Comportamiento del STT con un tamaño de mensaje variable (valores medios) utilizando SF 12.	77
6.29 STT frente a la distancia entre dos nodos utilizando SF 7.	78
6.30 STT frente a la distancia entre dos nodos utilizando SF 12.	78
6.31 Representación de la energía consumida por el dispositivo cuando no se realizan operaciones.	79
6.32 Representación de la energía consumida por el dispositivo cuando se envía un único paquete de mensajes.	80
6.33 Representación de la energía consumida por el dispositivo cuando se envían cinco paquetes.	80
6.34 Arquitectura FUDGE.	81
6.35 Resultados de los experimentos de evaluación del rendimiento.	83
6.36 Ejemplos de Grafana para una aplicación de monitorización.	84

Índice de tablas

2.1	Aplicaciones de LPWAN	9
2.2	Relevancia de las características para las aplicaciones LPWAN	11
4.1	Tecnologías de red utilizadas en IoT	28
4.2	LoRa Spreading Factors	31
4.3	Especificaciones técnicas de la tecnología LoRa	32
6.1	Tamaño de los archivos WAV y MP3	68
6.2	Segmentación de mensajes de diferentes tamaños y STT.	76

Índice de listas

5.3.1 Estructura JSON del contenido.	53
6.3.1 Estructura de los mensajes intercambiados.	69
6.4.1 Estructura JSON del mensaje enviado.	72
6.5.1 Fragmento del código del gestor de persistencia para manejar los mensajes entrantes.	82
6.5.2 Fragmento del código de una búsqueda del servicio de mensajería.	84
6.5.3 Datos proporcionados por un dispositivo Ruuvi.	85
6.5.4 Fragmento del código de los proxys de contenido para el dispositivo de Ruuvi.	86

Capítulo 1

Introducción

En gran parte de los países en desarrollo, la banda ancha móvil (3G o superior) o la comunicación por satélites son la forma más común, y algunas veces la única, de tener acceso a comunicaciones digitales. Es una condición necesaria, pero no suficiente, ya que los usuarios pueden enfrentarse a otros obstáculos a la conectividad. Mientras que prácticamente todas las zonas urbanas del mundo están cubiertas por una red de banda ancha móvil, en las zonas rurales persisten muchas carencias.

Hace tiempo que las comunicaciones digitales son una fuente de innumerables oportunidades de realización personal, desarrollo profesional y creación de valor. Con la pandemia de COVID-19, se han convertido en una necesidad vital para trabajar, aprender, acceder a servicios básicos y mantenerse en contacto, la pandemia ha puesto de manifiesto el coste paralizante de la exclusión digital.

El reporte de estadísticas de la *International Telecommunication Union (ITU)* de 2021 [1] informan que, 4.900 millones de personas utilizaron Internet en 2021, esto significa que el 63% de la población mundial está ahora en línea y se estima que casi 800 millones de personas se han conectado desde 2019, lo que representa un aumento del 17% (ver Figura 1.1).

Las estadísticas revelan una “gran brecha” de conectividad que separa a los habilitados digitalmente de los excluidos digitalmente, ya que el 96% de los 2.900 millones de personas que siguen sin conexión viven en países en desarrollo. A nivel mundial, los habitantes de las zonas urbanas tenían el doble de probabilidades de utilizar Internet que los de las zonas rurales en 2020 (ver Figura 1.2).

1. INTRODUCCIÓN

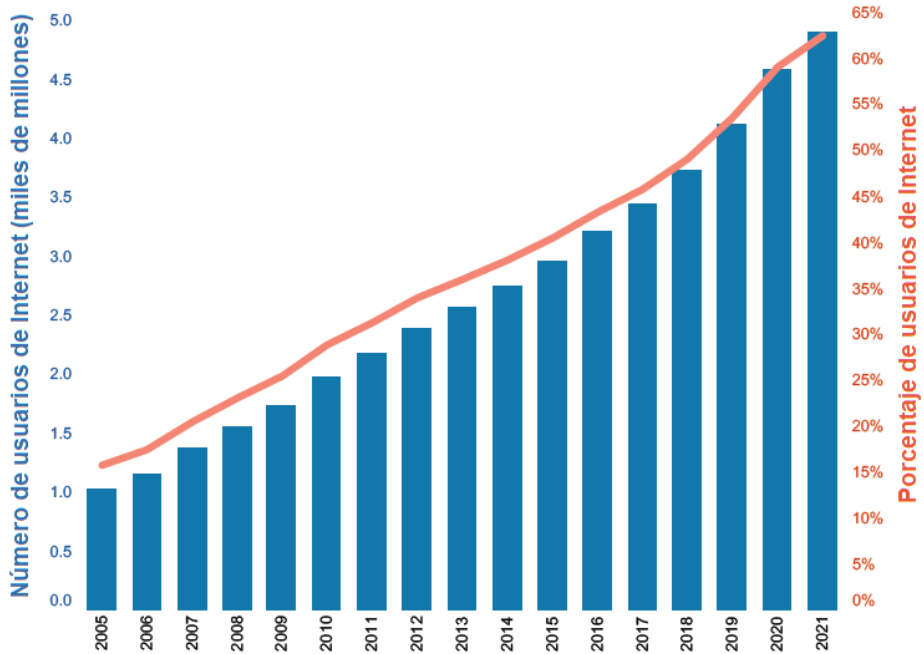


Figura 1.1: Evolución de usuarios de Internet [1].

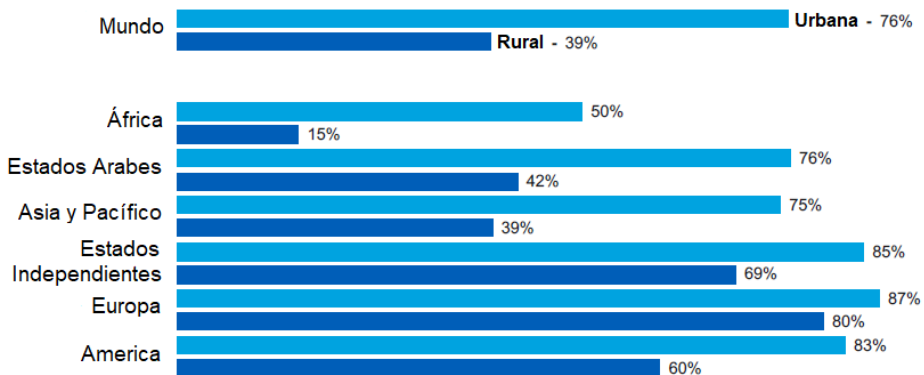


Figura 1.2: Porcentaje de personas que utilizan Internet por ubicación [1].

El despliegue de infraestructura en zonas rurales puede ser el doble de caro, mientras que los posibles ingresos pueden ser hasta diez veces menores, lo que representa un problema económico fundamental para los operadores, según Cruz et al. [2], una combinación que afecta profundamente el modelo de negocio. En otras palabras, el alto coste del despliegue de infraestructura, combinados con una baja demanda de servicios de Internet móvil, genera una proporción de oferta y demanda que resulta en niveles de baja cobertura de la población. Cuando el *Global System for Mobile communications (GSM)* está presente, el *Short Message Service (SMS)* ha sido la aplicación estrella para los servicios móviles en todo el mundo, debido a su capacidad para mantener comunicación entre individuos y también interconectar miembros de una comunidad.

La *Global System for Mobile communications Association (GSMA)* en su reporte del 2021 [3], menciona cinco barreras para la adopción y uso del Internet móvil:

- **Conocimientos y habilidades:** Los habitantes no conocen ni comprenden el Internet móvil y sus ventajas, o tienen un bajo nivel de alfabetización y habilidades digitales.
- **Asequibilidad:** La gente no puede pagar los dispositivos, planes de datos u otras tarifas de servicio.
- **Relevancia:** Contenidos, servicios y productos relevantes que satisfacen las necesidades de los usuarios y capacidades no están disponibles.
- **Seguridad y protección:** La gente está preocupada por los aspectos negativos y riesgos de Internet, como los contenidos nocivos, el acoso, el fraude y la seguridad en línea.
- **Acceso:** La gente no tienen acceso a redes y facilitadores, como la electricidad o los dispositivos y servicios no son suficientemente accesibles.

A su vez, destacan que para los usuarios de móviles que conocen Internet móvil, pero no lo utilizan, las principales barreras son la alfabetización, las habilidades digitales y la asequibilidad, en especial de los teléfonos habilitados para Internet.

1.1 Motivación

El acceso a Internet de banda ancha se ha promovido como una prioridad central de las agendas políticas de alto nivel en todo el mundo. Esto se refleja en iniciativas como el Objetivo de Desarrollo Sostenible (ODS) de las Naciones Unidas y el tratamiento del acceso a la banda ancha como una cuestión clave en la ITU, entre otros. La *European Commission (EC)* ha establecido ambiciosos objetivos para los próximos años, resumidos en las agendas de banda ancha EC 2021 [4] y 2025 [5].

Las Redes Comunitarias (RC) se caracterizan por ser redes de gestión y construcción colectiva, dentro de sus particularidades destacan ser abiertas, neutrales y libres. Este tipo de redes suelen desplegarse con aportaciones de infraestructura de los participantes y se gestionan como un recurso común. La infraestructura con la que suelen construirse varía dependiendo del tipo de participante que haga la aportación, se permiten desde *routers WiFi* de bajo coste, hasta equipos de fibra óptica. Incrementar la cobertura y contrarrestar la brecha digital en zonas desatendidas suelen ser algunas de las principales motivaciones en la creación de este tipo de redes.

Existe una amplia colección de trabajos de investigación donde se proponen este tipo de soluciones para proveer de un medio de comunicación digital a zonas con baja o nula cobertura de telefonía móvil. De algunos de esos trabajos de investigación han surgido propuestas de prototipos de RC, los cuales han sido desarrollados y desplegados, obteniendo ciertos resultados satisfactorios.

Proporcionar un medio de comunicación digital con incluso los servicios de mensajería más básicos serían de gran ayuda para los habitantes de las zonas rurales. Ya que la posibilidad de comunicarse entre comunidades o entre comunidades y ciudades principales es un servicio muy demandado, ya sea para uso puramente personal o con fines comerciales. Además, una solución de este tipo también puede utilizarse para recolectar información de sensores entre las comunidades, para proporcionar alertas de desastres, datos meteorológicos o distribuir un boletín de anuncios y noticias relevantes para los habitantes.

Además, se podría potenciar el alcance de una solución como la que proponemos, si combinamos aplicaciones estándar de mensajería, incluyendo dispositivos que tengan acceso a Internet y sean capaces de procesar los datos o en su defecto poderlos compartir para ser procesados.

1.2 Objetivos

El objetivo principal de esta tesis es diseñar una solución basada en redes comunitarias, para proveer de un medio de comunicación digital a zonas rurales.

Los siguientes son los objetivos específicos que han sido propuestos para alcanzar el objetivo principal:

- Diseñar una arquitectura de red de largo alcance.
- Desarrollar un protocolo de transmisión de datos basado en *Low-Power Wide Area Networks (LPWAN)*.
- Integrar con aplicaciones estándar de Internet.
- Desarrollar una interfaz para integrar mensajes de voz
- Integrar dispositivos *Internet of Things (IoT)* en la red con sistemas *pub/sub*.

1.3 Estructura de la Tesis

Esta tesis está organizada en siete capítulos. A continuación, describimos brevemente el contenido de cada uno:

- **Capítulo 1. Introducción:** Presentamos una breve introducción, la motivación y objetivos que nos han llevado a la realización de esta tesis doctoral.
- **Capítulo 2. Redes Comunitarias, una visión general:** Describimos aspectos generales y conceptos relacionados con las Redes Comunitarias, que sirven como referencia, así como algunos casos de uso.
- **Capítulo 3. Análisis del problema:** Analizamos la situación actual de la problemática sobre la que propondremos nuestra solución.
- **Capítulo 4. Tecnologías relacionadas:** Se ofrece una revisión de los aspectos generales relacionados con las diferentes tecnologías que hemos utilizado en nuestra propuesta.
- **Capítulo 5. Solución propuesta:** Realizamos una descripción de nuestro desarrollo tecnológico y algunas funcionalidades.
- **Capítulo 6. Resultados:** Se presenta de forma detallada la solución, así como su implementación y algunos resultados experimentales.
- **Capítulo 7. Conclusiones:** Finalmente, en este capítulo describimos las conclusiones, contribuciones y trabajos futuros.

Capítulo 2

Redes Comunitarias, una visión general

En este capítulo presentamos una revisión general sobre las Redes Comunitarias (RC), sus elementos y funcionamiento. A su vez, se mencionan algunas soluciones propuestas por la comunidad investigadora que han servido como base para el desarrollo de esta tesis.

2.1 Redes Comunitarias: Definición

Las RC [6, 7, 8] son redes inspiradas, construidas y gestionadas por ciudadanos y organizaciones, sin ánimos de lucro. Se trata de iniciativas en las que las personas combinan sus esfuerzos y recursos para crear colectivamente infraestructuras de redes de comunicación, pero también defendiendo el derecho humano básico de acceso a Internet. Por lo general, son iniciadas por pequeños grupos de personas, que la mayoría de las veces son impulsadas por motivos culturales y políticos, entre los que destaca la lucha contra la brecha digital a través de la prestación de servicios de telecomunicaciones en zonas desatendidas.

Este tipo de redes se caracterizan por ser abiertas, libres y neutrales. Se consideran abiertas, ya que todo el mundo tiene derecho a saber cómo se construyen. Son libres porque el acceso a la red se rige por el principio de no discriminación, por lo que son universales. Y son neutrales porque se puede emplear cualquier solución técnica para ampliar la red, además, la red puede ser utilizada para transmitir datos de cualquier tipo por los participantes, con el propósito que sea.

Las iniciativas de base, como las RC, se reconocen explícitamente como un posible enfoque para amortizar los costes de despliegue de la infraestructura de red y una de las cuatro formas de implicar a las autoridades públicas en la realización de la visión de banda ancha [9].

Estas redes pueden apoyar a disminuir un sin número de problemáticas en las comunidades rurales, por ejemplo, Martínez et al. ofrece [10] un análisis de las necesidades de comunicación en la atención primaria de salud rural en los países en desarrollo. Descubrieron que una aplicación muy sencilla implementada originalmente sobre la comunicación de radio de *High Frequency (HF)* resultó ser bastante exitosa, al permitir programar citas médicas para los pacientes. Se comprobó que los pacientes en áreas aisladas tenían que gastar mucho tiempo y recursos para llegar al hospital más cercano, y muchas veces no podían ser tratados de inmediato, sino que se les agendaba una cita en un horario que implicaba un segundo viaje desde su casa.

Por mencionar, algunas otras iniciativas de Redes Comunitarias (RC) que han sido implementadas y obtuvieron resultados satisfactorios encontramos a: *guifi.net* [11] en España, *DakNet* [12] en India y Camboya, *RemIX* [13] y *Broadband for the Rural North (B4RN)* [14] en Reino Unido, *Funkfeuer free net* [15] en Austria y la *Athens Wireless Metropolitan Network (AWMN)* [16] en Grecia.

2.2 Tecnologías LPWAN

Como consecuencia de los diversos obstáculos que se presentan al usar *Wireless Fidelity (WiFi)* para enlaces de larga distancia, surgió otro paradigma para satisfacer las necesidades de comunicación, denominada *Low-Power Wide Area Networks (LPWAN)* [17, 18]. A la que se le considera una solución prometedora para las aplicaciones de comunicación de largo alcance y baja potencia de *Internet of Things (IoT)* y *Machine to Machine (M2M)*.





LPWAN es una clase de estándares y soluciones de comunicación inalámbrica con características tales como grandes áreas de cobertura, bajas velocidades de transmisión de datos con pequeños tamaños de paquetes y funcionamiento con baterías de larga duración. Se caracterizan principalmente por ser redes con recursos limitados, el bajo coste de los dispositivos y de implementación. Las tecnologías LPWAN se están desplegando y han mostrado un enorme potencial para una amplia gama de aplicaciones en IoT y M2M, especialmente en entornos restringidos [19].

Las expectativas claves [20] son los bajos costos de los dispositivos, los tamaños de los paquetes de 10 a 1000 *Bytes* a velocidades de enlaces ascendente de hasta 200 *kbps*, duración de la batería de más de 10 años, soporte para más de 100 mil dispositivos y una cobertura de 2 a 1000 km.




2.2.1 Aplicaciones inteligentes y sus características

La creciente popularidad de los casos de uso del IoT en dominios que dependen de la conectividad que abarca grandes áreas y de la capacidad de manejar un gran número de conexiones está impulsando la demanda de tecnologías de acceso LPWAN. Con el desarrollo en los campos de la electrónica miniaturizada, la detección, la comunicación, la actuación, la informática y las tecnologías de baterías, facilitan el diseño de soluciones de red de bajo consumo energético y decenas de kilómetros de cobertura. Estas propuestas deben ser compatibles con Internet para que la gestión de los datos, los dispositivos y la red puedan realizarse a través de plataformas basadas en la nube. Las tecnologías LPWAN pueden desplegarse para una amplia gama de aplicaciones inteligentes, como se indica en la Tabla 2.1.

Tabla 2.1: Aplicaciones de LPWAN [20].

Casos de uso	Aplicaciones principales
 <p>Medio ambiente</p>	Calidad del agua, reducción de la contaminación atmosférica, reducción del aumento de la temperatura del clima, incendios forestales, deslizamientos de tierra, seguimiento de animales, control del nivel de nieve y detección temprana de terremotos.
 <p>Agua inteligente</p>	Calidad del agua, fugas de agua, control de las crecidas de los ríos, gestión de las piscinas y fugas de productos químicos.
 <p>Medidores inteligentes</p>	Contadores eléctricos inteligentes, contadores de gas, contadores de flujo de agua, control de gasoductos y control de almacenes.
 <p>Agricultura y ganadería</p>	Medición de la temperatura, la humedad y la alcalinidad, mejora de la calidad del vino, invernaderos inteligentes, automatización y robótica agrícola, red de estaciones meteorológicas, compost, hidroponía, cuidado de las crías, control y seguimiento del ganado y niveles de gases tóxicos.

2. REDES COMUNITARIAS, UNA VISIÓN GENERAL

Casos de uso	Aplicaciones principales
 <p>Red inteligente y energía</p>	Control de la red, equilibrio de la carga, supervisión y medición a distancia, supervisión del estado de los transformadores y supervisión de los molinos de viento y las instalaciones de energía solar.
 <p>Casas inteligentes y bienes inmuebles</p>	Uso de la energía y el agua, temperatura, humedad, detección de incendios y humos, control remoto de electrodomésticos, sistemas de detección de intrusos, arte, conservación de bienes y espacio como servicio.
 <p>Ciencias de la salud</p>	Salud y parámetros de los pacientes, entornos médicos conectados, <i>wearables</i> sanitarios, vigilancia de los pacientes, telemedicina, detección de caídas, vida asistida, neveras médicas, atención a los deportistas, seguimiento de enfermedades crónicas y seguimiento de la población.

Los principales impulsores de las aplicaciones LPWAN son la cobertura, la capacidad, el coste y el funcionamiento de bajo consumo. Además, algunas aplicaciones requieren otras capacidades, por ejemplo, el interfuncionamiento con otras tecnologías y el soporte de voz. Por lo tanto, una solución LPWAN específica puede estar adaptada a un conjunto reducido de aplicaciones, mientras que otra solución puede estar diseñada para cubrir una amplia gama de aplicaciones y atributos.

La Tabla 2.2 muestra la relevancia de las principales características para las aplicaciones mencionadas anteriormente [20, 21]. Las escalas relativas de aplicabilidad de las características a la aplicación son alta (A), media (M) y baja (B). Para una solución LPWAN orientada a una aplicación determinada.

Las redes de acceso inalámbrico varían dependiendo de su aplicación y existen distintas tecnologías que abarcan diferentes rangos geográficos (ver Figura 2.1) que pueden ser utilizadas dependiendo del tipo de solución que se pretenda ofrecer.

Las LPWAN son las tecnologías de comunicación inalámbrica diseñadas para permitir las comunicaciones de largo alcance con un bajo consumo de energía, una interfaz de bajo coste y una tasa de *bits* relativamente baja para aplicaciones IoT y M2M. En general, las tecnologías LPWAN se clasifican en dos categorías, propietarias o estándar.

Tabla 2.2: Relevancia de las características para las aplicaciones LPWAN [20].

Aplicaciones	Cobertura	Capacidad	Costo	Bajo consumo
Medio ambiente	M	A	A	A
Agua inteligente	A	M	M	M
Medidores inteligentes	A	A	A	M
Agricultura y ganadería	A	A	M	A
Red inteligente y energía	A	A	M	M
Casas inteligentes y bienes inmuebles	A	M	B	B
Ciencias de la salud, eHealth y Wearables	A	A	M	A

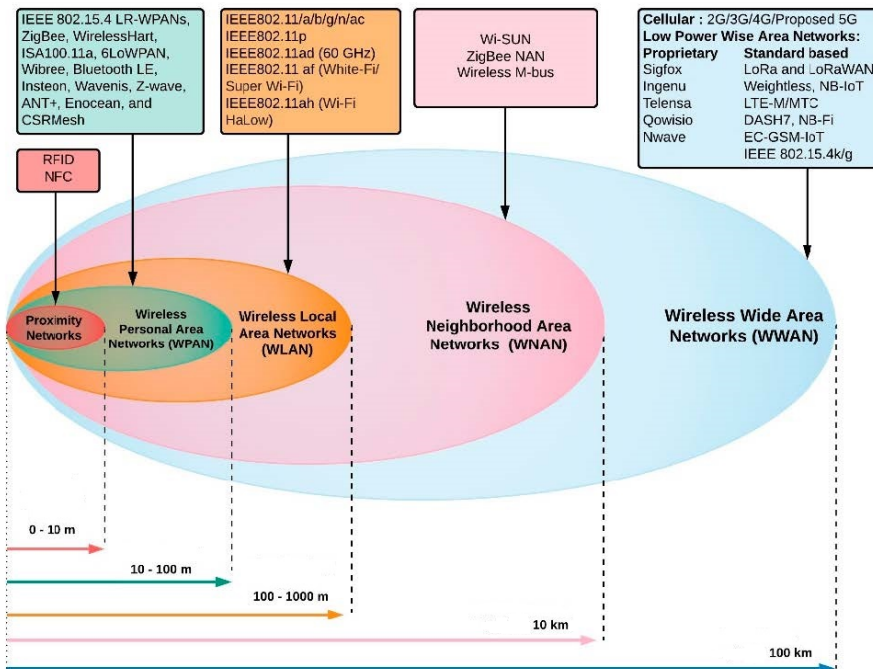


Figura 2.1: Cobertura geográfica del acceso inalámbrico [20].

Tecnologías propietarias:

- Sigfox: Es una tecnología propia de LPWAN [22] de banda ultra estrecha que utiliza una tasa de modulación lenta para lograr un mayor alcance.
- Ingenu: Se basa en la técnica patentada de *Random Phase Multiple Access (RPMA)* [23] con una regulación del espectro más flexible, lo que permite un mayor rendimiento y capacidad
- Telensa: es una tecnología LPWAN de banda ultra estrecha. Funciona en las bandas *Industrial, Scientific and Medical (ISM)* sin licencia de 868 MHz y 915 MHz. Tiene capacidad de comunicación bidireccional, por lo que puede utilizarse tanto para la supervisión como para el control [24].
- Qowisio: Es una tecnología de banda ultra estrecha y modo dual para aplicaciones LPWAN. Es compatible con LoRa y ofrece opciones tecnológicas y flexibilidad a los usuarios finales [25].
- Nwave: es principalmente una solución desarrollada por Nwave [26] para sistemas de aparcamiento inteligentes. Esta tecnología de banda ultra estrecha también se basa en el funcionamiento en la banda ISM sin licencia de menos de 1 GHz.

Tecnologías estándar:

- LoRa y LoRaWAN: LoRa es una tecnología de capa física que funciona en la banda *ISM* sin licencia y se basa en la técnica de *Chirped Spread Spectrum (CSS)* [27]. LoRaWAN incluye funcionalidades de red y de capa superior. LoRaWAN [28] ofrece tres clases de dispositivos finales para responder a los diferentes requisitos de una amplia gama de aplicaciones IoT.
- Weightless: Es un estándar abierto que ofrece conectividad LPWAN. Hay tres versiones de Weightless [29]: Weightless-W, Weightless-N y Weightless-Weightless-P.
- Narrowband IoT: es una tecnología LPWAN de la versión 13 del *3rd Generation Partnership Project (3GPP)* [30] que ofrece flexibilidad de despliegue al permitir el uso de una pequeña porción del espectro disponible en las bandas de *Long Term Evolution (LTE)*.
- LTE-M: es también una tecnología basada en el estándar 3GPP y opera en el espectro LTE con licencia. Es compatible con las redes LTE y proporciona la conexión para el tráfico M2M [31].
- DASH7: Fue desarrollado para las comunicaciones de sensores y actuadores inalámbricos y tiene su origen en la norma ISO 18000-7. DASH7 [32] es

una extensión de la tecnología Radio Frequency Identification (RFID) y es una tecnología de banda ISM de baja potencia y largo alcance que opera principalmente a 433 MHz.

- NB-Fi: Está diseñada para aplicaciones de comunicación M2M bidireccionales de banda estrecha, baja potencia y largo alcance [33]. Esta solución ha sido diseñada por WavIoT, un proveedor de LPWAN de *Infrastructure as a Solution (IaaS)*.
- EC-GSM-IoT: *Enhanced Coverage-Global System for Mobile Internet of Things (EC-GSM-IoT)*, desarrollado por el 3GPP, es uno de los candidatos prometedores para el IoT celular de bajo consumo y largo alcance (cIoT) por ofrecer una cobertura y una duración de la batería similares a las de NB-IoT [34]. Se basa en el *General Packet Radio Service (GPRS)* mejorado y está diseñado para aplicaciones LPWAN escalables y de baja complejidad.

2.3 WiFi

Aunque inicialmente se desarrolló para redes de área local, WiFi ha demostrado ser rentable para aplicaciones de larga distancia [35, 36] y el enorme éxito de esta tecnología inalámbrica será difícil de superar por tecnologías alternativas como WiMax [37].

El uso de WiFi a larga distancia tiene dos principales limitaciones, que son el requisito de la línea de visión entre los puntos finales y la vulnerabilidad a las interferencias en la banda sin licencia [38]. La primera limitación puede superarse a menudo aprovechando las elevaciones del terreno, o utilizando torres para librar los obstáculos y para proporcionar una zona de Fresnel. La segunda limitación es menos pronunciada en zonas rurales, ya que usando la banda de 5 GHz puede aliviarse, debido a que es menos saturada.

También existen dos obstáculos que superar al aplicar WiFi a larga distancia: Las limitaciones de presupuesto de potencia y las de temporización. El primer obstáculo puede resolverse fácilmente usando antenas direccionales de alta ganancia; el obstáculo de la temporización se soluciona al modificar el mecanismo de acceso al medio, como ha hecho el grupo TIER de la Universidad de Berkeley [39].

Otra cuestión relevante es el uso de equipos de bajo coste para el apuntamiento de las antenas, lo que hace que sean asequibles y más fáciles de utilizar por personas con poca formación. En [40] se mencionan pruebas de comunicación a 70, 101, 279 y 382 km usando *Wireless Local Area Network (WLAN)* de larga distancia empleando equipos WiFi de bajo coste.

Flickenger et al. [40] detalla los resultados que obtuvieron al realizar pruebas de comunicación usando WLAN a larga distancia, empleando equipos de WiFi de bajo coste a 101, 279 y 382 km. A su vez, han establecido un escenario de pruebas permanentes de 101 km desde Trieste hasta Piancavallo en Italia (ver Figura 2.2)

2. REDES COMUNITARIAS, UNA VISIÓN GENERAL

para examinar el rendimiento de diferentes equipos inalámbricos, correlacionando el rendimiento con las condiciones meteorológicas y otros factores ambientales.



Figura 2.2: Enlace inalámbrico desde Trieste hasta Piancavallo, Italia [40].

2.3.1 Guifi.net

Esta iniciativa nace en el año 2004 en la comarca de Osona (Cataluña) como un proyecto tecnológico de telecomunicaciones para resolver las dificultades de acceso a Internet de banda ancha en áreas rurales, con la intención de combatir la falta de interés de proporcionar servicios de los operadores tradicionales. En 2008, los miembros de la comunidad crearon la Fundación Guifi.net, una organización sin ánimo de lucro responsable de la coordinación de la comunidad y del apoyo al empleo de los usuarios. La fundación fomenta el empleo de la red, pero no la gestiona, por lo que las decisiones sobre el crecimiento y el mantenimiento de la misma las toman los usuarios.

Guifi.net [41, 42] se define como una Red Abierta, Libre y Neutral (RALN) construida por sus miembros, que son ciudadanos y organizaciones que ponen en común sus recursos y coordinan sus esfuerzos para construir y operar una infraestructura de red local [43, 44]. El principio subyacente de guifi.net es el modelo de recursos procomunes como método óptimo para gestionar una red. La mayoría de los enlaces de la red son inalámbricos [45], pero cada vez se utilizan más enlaces de fibra óptica.

Las bases fundamentales [46] de esta RALN son:

- Eres libre de utilizar la red para cualquier propósito mientras no perjudiques el funcionamiento de la propia red, la libertad de otros usuarios, y respetes las condiciones de los contenidos y servicios que circulan libremente.
- Eres libre de conocer como es la red, sus componentes y su funcionamiento, también puedes difundir su espíritu y funcionamiento libremente.
- Eres libre para incorporar servicios y contenidos a la red con las condiciones que quieras.
- Eres libre de incorporarte a la red y ayudar a extender estas libertades y condiciones.

La red se basa en un acuerdo de interconexión en el que se concede a cada nuevo participante el derecho a utilizar la red para cualquier fin, sin que ello afecte el funcionamiento de la red o la libertad de los demás usuarios, el derecho a conocer todos los detalles de la red, sus componentes, y la libertad de unirse para ampliarla siguiendo las mismas condiciones. La única limitación es que esta red no puede utilizarse de forma que afecte a su funcionamiento o la libertad de otros usuarios. Además, genera un modelo de actividad económica sostenible, colaborativa y de proximidad.

Para el año 2010 guifi.net alcanzó 10.000 nodos operativos, interconectados mediante radio enlaces WiFi. Los miembros de la comunidad y voluntarios empezaron a desplegar enlaces de fibra óptica ese mismo año para dotarse de conectividad mediante redes de telecomunicaciones de nueva generación. En diciembre de 2016 se contaba con 32.500 nodos activos, la mayoría en Cataluña, pero también muchos otros en la Comunidad Valenciana, Islas Baleares, Madrid, Andalucía, Asturias y el País Vasco. En el primer trimestre del 2022 se estiman 37.300 nodos operativos y para formar una red de cerca de 67.800 nodos, no solo en la Unión Europea, sino también alcanzando otros países. En la Figura 2.3 podemos observar la curva del aumento de los nodos operativos por año y en la Figura 2.4 se aprecia la cantidad de nodos operativos desde la creación de la red hasta la actualidad.

2. REDES COMUNITARIAS, UNA VISIÓN GENERAL

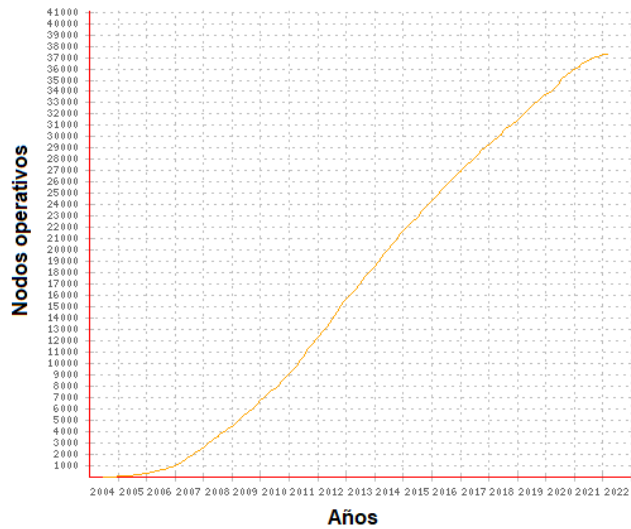


Figura 2.3: Curva de crecimiento anual de nodos operativos Guifi.net [47].

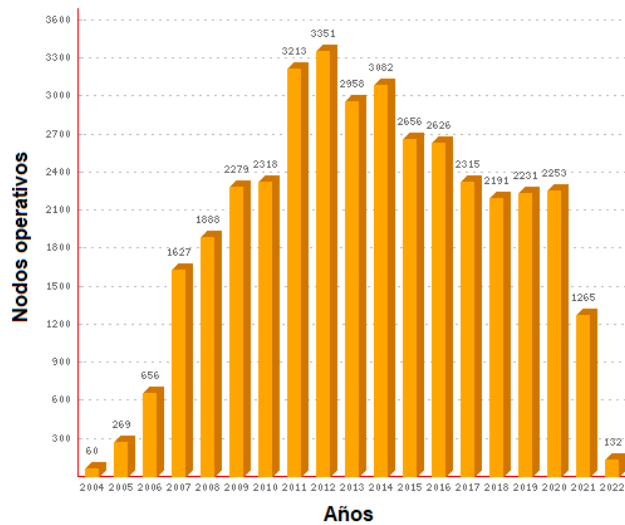


Figura 2.4: Incremento anual de nodos operativos Guifi.net [47].

2.3.2 DakNet

DakNet [48, 49] aprovecha la infraestructura de comunicaciones y transporte existente para distribuir la conectividad digital a los pueblos que carecen de infraestructura. Fue desarrollada por investigadores del *MIT Media Lab*. DakNet, cuyo nombre deriva de la palabra hindi “Dak”, que significa “correo”, combina un medio de transporte físico con la transferencia de datos inalámbrica y de esta forma ampliar la conectividad a Internet.

La topología de la red, en general, se considera dinámica, ya que con el tiempo la conectividad entre los nodos puede variar debido a la salida de nodos, la llegada de nuevos nodos y los posibles nodos móviles. Los datos se transmiten a través de enlaces cortos punto a punto entre kioscos y dispositivos de almacenamiento portátiles, llamados *Mobile Access Points (MAPs)*.

Los dispositivos MAP se montan y alimentan en vehículos (ver Figura 2.5) con un pequeño generador y a su paso por las zonas rurales, al estar dentro del alcance de comunicación con algún kiosco, transmiten los datos que podrán ser consultados por los habitantes posteriormente mediante dispositivos móviles (*smartphones, tablets*) WiFi. Los MAP transportan físicamente los datos entre los kioscos y los dispositivos de comunicación privados y entre los kioscos y los *hubs* (para el acceso a Internet).



(a) Autobús equipado con unidad MAP.

(b) Motocicleta equipada con unidad MAP.

Figura 2.5: Vehículos habilitados para DakNet [12].

Los componentes principales de la arquitectura DakNet son:

- MAP.
- HUB.
- Kiosco WiFi.

2. REDES COMUNITARIAS, UNA VISIÓN GENERAL

La Figura 2.6, nos brinda una visión general del funcionamiento de DakNet. Esta solución se ha desplegado con éxito en pueblos al norte de Camboya y la India.

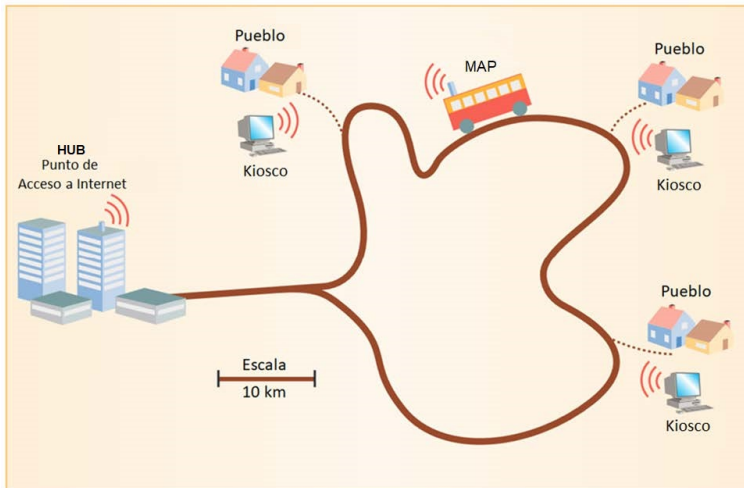


Figura 2.6: Arquitectura general DakNet [12].

El funcionamiento de DakNet tiene, por tanto, dos pasos:

- Cuando el vehículo equipado con MAP entra en alcance de un kiosco con WiFi en el pueblo, automáticamente detecta la conexión inalámbrica, carga y descarga los datos almacenados.
- Cuando el vehículo equipado con MAP entra en alcance de un punto de acceso de Internet (HUB), sincroniza automáticamente los datos de todos los kioscos rurales, utilizando Internet.

Los pasos se repiten para cada vehículo que lleve una unidad MAP, creando así una red inalámbrica de bajo coste e infraestructura limitada.

2.4 Resumen

El análisis general que hemos realizado nos brinda un panorama más amplio respecto a las Redes Comunitarias, así como el enfoque que ha tomado la comunidad científica para dotar de algún medio de comunicación digital a las áreas con poca o nula cobertura.

Los ejemplos anteriores son solo algunos de los que representan el creciente ámbito de las *Information and Communications Technologies for Development (ICT4D)*. Este ámbito pretende ofrecer una solución sostenible para la conectividad ubicua. Los miembros del grupo de *Internet Research Task Force (IRTF)*, *Global Access to the Internet for All Research Group (GAIA)* han elaborado recientemente un documento de referencia fundamental [50]. Las actividades de GAIA están orientadas principalmente a crear una mayor visibilidad e interés entre la comunidad en general, sobre los retos y oportunidades para permitir el acceso global a Internet, tanto en términos de tecnología como de impulsores sociales y económicos para su adopción.

En este capítulo hemos descrito las RC y sus principales características. Iniciamos con la definición de estas redes, así como de sus componentes y la forma en que los usuarios pueden ampliarlas. Después detallamos algunas tecnologías con las que se pueden implementar estas soluciones, mencionamos algunas aplicaciones y el alcance geográfico que tienen. Por último, describimos un par de ejemplos que se han puesto en marcha y han obtenido resultados satisfactorios.

Capítulo 3

Análisis del Problema

A pesar de los esfuerzos por parte de gobiernos y organizaciones por alcanzar la supresión de la brecha digital, se estima que algo más de, 2900 millones de personas, o el 37% de la población mundial (ver Figura 3.1), nunca han usado Internet, lo que las ha mantenido aisladas de una fuente vital de comunicación, información y educación, según datos de la *International Telecommunication Union (ITU)*, el 96% de la población mundial sin conexión, frecuentemente viven en países en desarrollo.

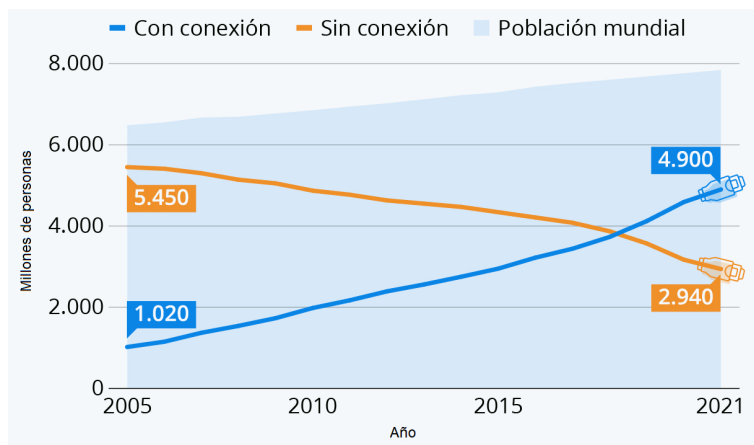


Figura 3.1: Número estimado de personas que utilizan/no utilizan internet en todo el mundo [1].

3. ANÁLISIS DEL PROBLEMA

De acuerdo con el reporte de la agenda digital del 2021 [51], en regiones como España destaca un notable decremento en el porcentaje de la población que no había tenido ningún contacto con o accedido a Internet, dicho porcentaje se situó por debajo del 7% en los últimos 10 años (ver Figura 3.2).

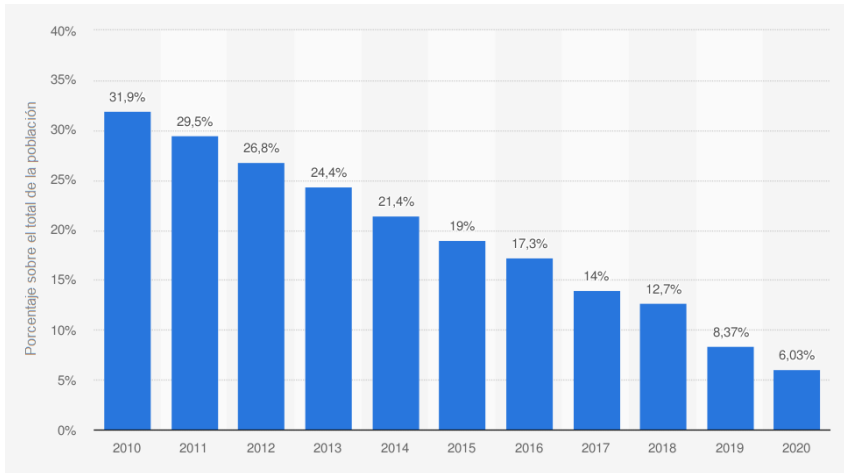


Figura 3.2: Evolución del porcentaje de personas que nunca había accedido a Internet en España de 2010 a 2020 [51].

Partiendo de la definición general de las Redes Comunitarias (RC), consideramos que una propuesta basada en este tipo de redes, sería una solución factible para brindar al menos un medio de comunicación digital a zonas rurales. Una solución de este tipo podría ser una medida contra los obstáculos económicos y de despliegue de infraestructuras que han sido los mayores impedimentos para alcanzar estos objetivos.

La solución debe apoyarse en tecnologías inalámbricas que permitan conexiones de largo alcance, que a su vez requieran de poco despliegue de infraestructura para conseguir dichas conexiones. Además, se debe tener en cuenta que en estas regiones también existen limitaciones de suministro eléctrico, por lo cual, los dispositivos a utilizarse, deben ser de bajo costo y de bajo consumo energético.

Proponemos desarrollar una arquitectura para un sistema de mensajería, para que los habitantes de zonas rurales puedan disponer de un medio de comunicación con otras comunidades o con ciudades, ya sea para uso puramente personal o con fines comerciales. Este sistema de mensajería permitirá que los usuarios puedan enviar y recibir mensajes de otros usuarios, así como mensajes genéricos que podrá recibir toda la comunidad, con información que sea relevante para ellos, por ejemplo: precios de algún producto que deseen comprar o vender, información sobre atención sanitaria o previsión meteorológica, etc.

Además de utilizarse como un medio de comunicación, el sistema de mensajería puede ser usado para recolectar datos de sensores que se encuentren cerca de la comunidad y reenviarlos a las partes interesadas, estos datos a su vez podrán ser procesados por dispositivos más potentes y que cuenten con acceso a Internet.

3.1 Educación e infraestructura

Otra de las grandes carencias en las zonas rurales es el acceso a la educación, esto se ve reflejado en los altos índices de analfabetismo que existen principalmente en estas regiones. Según datos del Instituto de Estadística de la *United Nations Educational, Scientific and Cultural Organization (UNESCO)*, hay cerca de 781 millones de personas en edad adulta analfabetas, 16% de la población global. De ellas, unos 500 millones son mujeres, el 64%, este porcentaje no ha mejorado desde 1990, según la UNESCO. Por otro lado, del total de adultos analfabetos, 126 millones son jóvenes de entre 15 y 24 años, de los cuales, 77 millones (el 16%) son chicas (ver Figura 3.3).

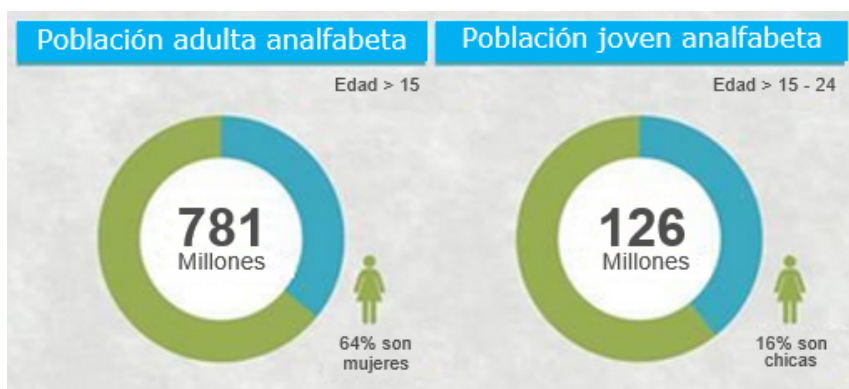


Figura 3.3: Porcentaje de analfabetismo a nivel global [52].

Basándonos en los datos de la UNESCO, consideramos conveniente que el sistema de mensajería no solo puedan enviar/recibir mensajes de texto, sino también mensajes de voz, para ofrecer una alternativa de comunicación a las personas analfabetas, como hacen aplicaciones como WhatsApp¹ y Telegram².

Las zonas rurales o en desarrollo pueden ser algunas de las últimas en obtener los beneficios de las recientes innovaciones. La falta de infraestructura de energía o red móvil puede limitar su utilidad en estos entornos y reducir potencialmente su adopción.

¹<https://www.whatsapp.com/>

²<https://telegram.org/>

3. ANÁLISIS DEL PROBLEMA

Sin embargo, algunos estudios han demostrado que la penetración de dispositivos como *smartphones* no ha sido completamente nula en estas regiones [53, 54, 55]. La telefonía móvil ha mejorado las condiciones sociales y económicas en algunas zonas rurales, ya que reduce eficazmente la “distancia” que existe entre individuos e instituciones, facilitando el acceso a la educación, información comercial y sobre mercados, la provisión de servicios de orientación y asistencia técnica, notificaciones y alertas tempranas.

Además, el uso de paneles solares es una de las grandes soluciones para las familias que viven en zonas rurales, por ser energía sostenible y de bajo coste. La instalación de paneles solares es una solución en sectores sin electricidad, ya que estos pueden generar energía en los sitios más remotos [56, 57]. Estas propuestas suelen tener mucho éxito debido también a que no requieren la instalación de mucha infraestructura como con los proveedores de servicio eléctrico tradicional.

La presencia de dispositivos como *smartphones* o paneles solares en estas zonas, puede representa una pequeña ventaja al intentar proveer de un medio de comunicación digital, ya que se podría aprovechar la infraestructura con la que se cuenta y de esta forma disminuir los gastos adicionales para los habitantes.

Capítulo 4

Tecnologías relacionadas

En este capítulo realizaremos una descripción detallada de las tecnologías con las que hemos desarrollado la arquitectura para el sistema de mensajería, así como los dispositivos que se usaron para realizar las pruebas con las que se obtuvieron algunos resultados preliminares.

4.1 IoT

Internet of Things (IoT) es la red de objetos físicos (dispositivos, instrumentos, vehículos, edificios y otros elementos), dotados de electrónica, circuitos, software, sensores y conectividad de red, que permiten a estos objetos recolectar e intercambiar datos. IoT permite detectarlos y controlarlos a distancia a través de la infraestructura de red existente, creando oportunidades para una integración más directa al mundo físico en los sistemas informáticos, y dando lugar a una mayor eficiencia y precisión. IoT tiene la capacidad de interactuar sin intervención humana.

El concepto de una red de dispositivos inteligentes se discutió ya en 1982, cuando una máquina de Coca-Cola modificada en la Universidad Carnegie Mellon se convirtió en el primer objeto conectado a Internet [58], era capaz de informar de su inventario y la temperatura de las bebidas.

Kevin Ashton (nacido en 1968) es un pionero tecnológico británico conocido por haber inventado el término “*Internet of Things*” para describir un sistema en el que Internet está conectado al mundo físico a través de sensores ubicuos. Ashton propuso por primera vez el concepto de IoT en 1999, y se refirió a este, como

4. TECNOLOGÍAS RELACIONADAS

objetos conectados identificables de forma única con tecnología *Radio Frequency Identification (RFID)*.

En general, IoT se definió como “una infraestructura de red global dinámica con capacidad de autoconfiguración basada en estándares y protocolos de comunicación”.

Si observamos la evolución de Internet, podemos clasificarla en cinco épocas:

1. Internet de los documentos: Bibliotecas electrónicas, páginas web basadas en documentos.
2. Internet del comercio: Comercio electrónico, banca electrónica y páginas web de bolsa.
3. Internet de las aplicaciones: Web 2.0
4. Internet de las personas: Redes sociales.
5. Internet de las cosas: IoT, Dispositivos y máquinas conectadas.

Actualmente, el uso de Internet no se limita a los seres humanos que acceden a la red a través de los dispositivos clásicos, como los ordenadores personales y los teléfonos móviles, sino que se extiende cada vez más a objetos cotidianos y del entorno inmediato (por ejemplo, bombillas, frigoríficos o autos) que están conectados a Internet. Se estima que el número de dispositivos conectados Internet (teléfonos inteligentes, *wearables*, dispositivos domésticos inteligentes) supere los 30.000 millones en 2020 y los 75.000 millones para 2025 (ver Figura 4.1).

IoT mezcla el mundo virtual y el mundo físico al reunir diferentes conceptos y componentes técnicos: redes omnipresentes, miniaturización de los dispositivos, comunicación móvil y nuevo ecosistema. En este paradigma, las aplicaciones, los servicios, los componentes *middleware*, las redes y los nodos finales se organizan estructuralmente y se utilizan de formas totalmente nuevas.

Los elementos claves de cualquier estructura IoT (ver Figura 4.2) generalmente son los mismos:

- Cosas inteligentes: Son los objetos cotidianos que se encargaran de recolectar los datos.
- Redes y *gateways*: Permiten a los dispositivos de baja potencia (comunes en IoT) acceder a Internet.
- *Middleware* o plataformas IoT: Proporcionan espacios de almacenamiento de datos y motores de computación avanzados junto con capacidades analíticas.
- Aplicaciones: Permiten a los usuarios finales beneficiarse de IoT y manipular objetos físicos.

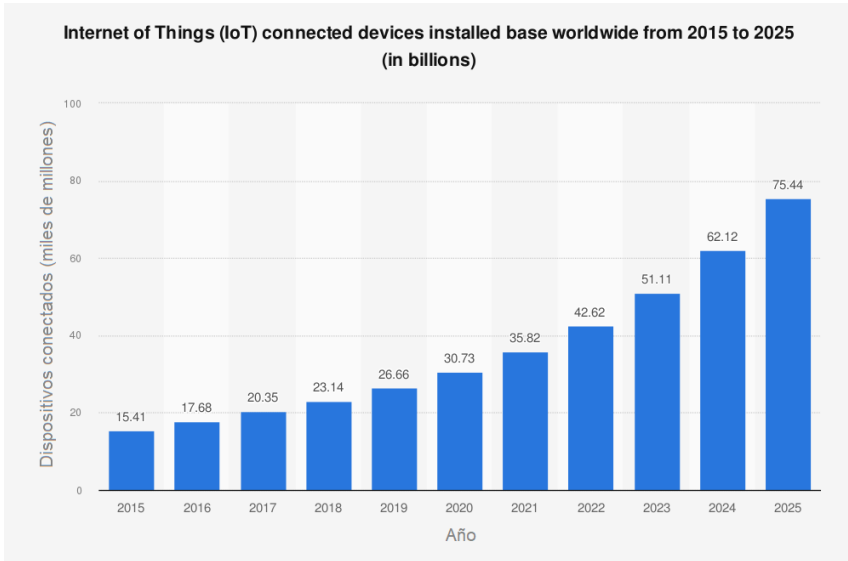


Figura 4.1: Número de dispositivos IoT conectados a Internet en el mundo de 2015 a 2025 [59].

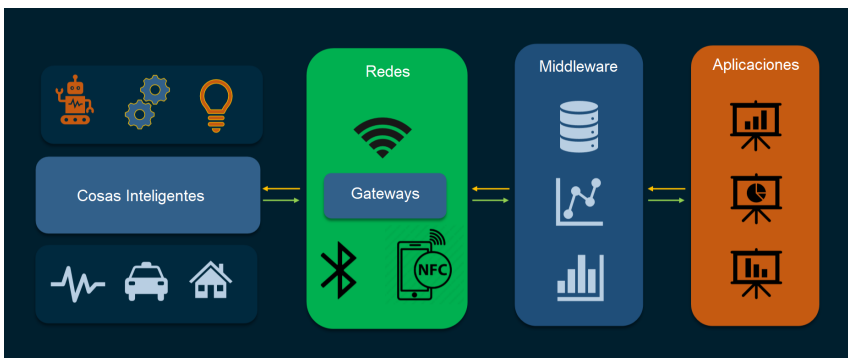


Figura 4.2: Estructura de un sistema IoT [60].

4. TECNOLOGÍAS RELACIONADAS

Tabla 4.1: Tecnologías de red utilizadas en IoT [61].

Red	Conectividad	Casos de uso
Ethernet	Cableado, corto alcance	Estaciones IoT: cámaras de vídeo, videoconsolas, equipos fijos
WiFi	Inalámbrico, corto alcance	Hogar inteligente, dispositivos que se recargan fácilmente
NFC	Inalámbrico, muy corto alcance	Sistemas de pago, hogar inteligente
Bluetooth	Inalámbrico, corto alcance	Dispositivos domésticos y wearables
LPWAN	Inalámbrico, largo alcance	Hogar inteligente, ciudad, inteligente, agricultura inteligente
ZigBee	Inalámbrico, corto alcance	Domótica, sanidad y centros industriales
Red celular	Inalámbrico, largo alcance	Drones que envían vídeo e imágenes

Las comunicaciones entre los dispositivos y los servicios o gateways implican diferentes tecnologías de red, como podemos observar en la Tabla 4.1.

IoT tiene el potencial de transformar radicalmente muchos ámbitos de la vida de las personas, algunos de los principales campos de aplicación son:

- **Industria:** Control del proceso de producción, supervisión del medio ambiente industrial, seguimiento de la cadena de suministro de la fabricación, supervisión del ciclo de vida del producto (CVP), seguridad en la fabricación, ahorro de energía y control de la contaminación.
- **Agricultura:** Utilización de los recursos agrícolas, gestión cuantitativa en el proceso de producción agrícola, producción y cultivo de la vigilancia del medio ambiente, gestión de la calidad, seguridad y trazabilidad de los productos agrícolas.
- **Logística:** Control de inventarios, gestión de la distribución, trazabilidad y otros sistemas logísticos modernos, plataforma de servicios logísticos públicos que abarca diferentes zonas y dominios, con comercio electrónico y logística inteligente.
- **Transporte:** Percepción y notificación del estado del tráfico, orientación del tráfico y control inteligente, posicionamiento y programación de vehículos, seguimiento y servicio de vehículos a distancia, coordinación de vehículos y carreteras, plataforma integrada de transporte inteligente.

- Red eléctrica: Monitorización de instalaciones de energía, subestación inteligente, despacho automático de energía, energía inteligente, programación inteligente, lectura remota de contadores.
- Protección del medio ambiente: Control de las fuentes de contaminación, control de la calidad del agua y aire, red de recogida de información medioambiental y su plataforma de información.
- Seguridad: Vigilancia de la seguridad social, del transporte de mercancías peligrosas y químicas, vigilancia de la seguridad alimentaria, alerta temprana y respuesta de emergencia para infraestructuras como puentes importantes o edificios, tránsito ferroviario, suministro público de agua/desagüe.
- Medicina: Control inteligente de medicamentos, gestión hospitalaria, recogida/análisis de parámetros de fisiología y medicina humana, servicio médico a distancia para la familia y la comunidad.
- Hogar: Red de área doméstica, seguridad en el hogar, control inteligente de electrodomésticos, medición inteligente, ahorro de energía, bajas emisiones de carbono, y aprendizaje a distancia.

4.2 Tecnología LoRa y LoRaWAN

Long-Range (LoRa) es una tecnología *Low-Power Wide Area Networks (LPWAN)*. LoRa implica dos capas diferentes: (a) una capa física que utiliza la técnica de modulación *Chirped Spread Spectrum (CSS)* [62], y (b) un protocolo de capa *Media Access Control (MAC)* también conocido como LoRaWAN (ver Figura 4.3). LoRa es desarrollada y patentada por *SEMTECH*¹, define la capa física del sistema. Mientras que LoRaWAN, es desarrollada por *LoRa Alliance*², y define el protocolo.

Sigfox, es una red privada, que desarrollo la primera tecnología LPWAN que se lanzó en 2009 y proporcionó servicios de red para aplicaciones de IoT. Después, Ingenu, una red privada y pública, proporcionó servicios para aplicaciones IoT con tecnología de acceso múltiple en fase aleatoria (*random phase multiple access*). Posteriormente, en 2012, la tecnología LoRa llegó como una LPWAN privada o pública para proporcionar servicios de IoT.

4.2.1 LoRa

El nombre, LoRa, es una referencia a los enlaces de datos de muy largo alcance que permite esta tecnología, hasta 5 kilómetros en zonas urbanas, y hasta 15 kilómetros o más en zonas rurales (con línea de visión). Una característica clave de

¹<https://www.semtech.com/>

²<https://lora-alliance.org/>

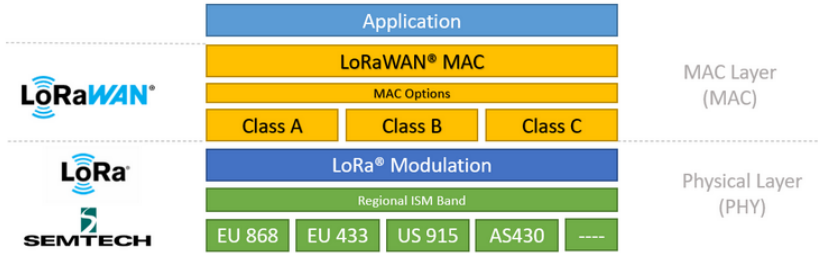


Figura 4.3: *Stack* de la tecnología LoRaWAN [63].

las soluciones basadas en LoRa es que requieren muy poca energía, lo que brinda la oportunidad de crear dispositivos con batería que pueden durar por mucho tiempo.

La señal LoRa posee características que la hacen apta para transmisiones de largo alcance y baja potencia: la duración de la señal LoRa es más larga que las ráfagas típicas de interferencia generadas por los sistemas *Frequency Hopping Spread Spectrum (FHSS)*, los errores que se presentan por dicha interferencia se pueden corregir de una manera sencilla mediante *Forward Error-correction Codes (FEC)*. El rechazo en el mismo canal y la selectividad típica fuera del canal de los receptores LoRa son 20 dB y 90 dB [64, 65] respectivamente, lo cual, sobrepasa los esquemas de modulación tradicionales como *Frequency-Shift Keying (FSK)*.

Esta tecnología tiene cinco parámetros importantes [66], que son: *Data Rate (DR)*, *Range*, *Spreading Factor (SF)*, *Bandwidth (BW)* y *Bitrate (BR)*. LoRa define de una forma no convencional a SF, como el logaritmo en base 2, del número de *chirps* por unidad de señal (ver Ecuación 4.1).

$$SF = \log_2 \left(\frac{R_c}{R_s} \right) \quad (4.1)$$

donde R_c y R_s son *Chirps Rate* y *Symbol Rate*, respectivamente [67]. SF indica el número de chirps por símbolo y oscila entre 7 y 12. En el caso de LoRa, el DR se determina usando la Ecuación 4.2:

$$Data\ Rate = SF \times \frac{BW}{2^{SF}} \times CR \quad (4.2)$$

donde CR es la relación entre los datos no redundantes y todos los datos dentro de las tramas de transmisión y recepción. BW puede tomar uno de los tres valores: 125, 250 o 500 kHz.

Para transmitir se utilizan las bandas sin licencia por debajo de 1 *Gigahertz (GHz)* y dependen de la región, Europa se puede transmitir en 868 y 433 MHz, en América en 915 MHz y Asia en 430 MHz (ver Figura 4.3). Además, las interferencias se pueden combatir aprovechando la modulación LoRa [68], en la que la velocidad de

transmisión se intercambia por alcance, en una aplicación clásica de la Ecuación 4.3 de capacidad de canal de Shannon:

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (4.3)$$

donde C es la capacidad o rendimiento en b/s, B es el ancho de banda en Hz, S es la potencia de la señal en W y N es la potencia del ruido en W.

La ganancia de procesamiento de LoRa se introduce en el canal de *Radio Frequency (RF)* multiplicando la señal de datos con un *Spreading Code (SC)* o *Chirps Sequence (CS)*. Al incrementar la tasa de chips, incrementamos los componentes de frecuencia del espectro total de la señal. Es decir, la energía de la señal total se distribuye en una gama más amplia de frecuencia, lo que permite al receptor discernir una señal con una relación señal-ruido más baja. En términos de LoRa, la cantidad de código de propagación aplicada a la señal de datos origina se denomina *Spreading Factor (SF)*. La modulación LoRa tiene seis SF (SF7 a SF12). Cuanto mayor sea el SF utilizado, más lejos podrá viajar la señal y seguir siendo recibida sin errores.

La Tabla 4.2 muestra los diferentes SF que pueden utilizarse para el envío de mensajes en un canal de 125 kHz. Muestra el BR equivalente, así como el alcance estimado (este depende del terreno; se alcanzarán distancias más largas en un entorno rural que en un entorno urbano). También muestra los valores de tiempo en el aire de la señal.

Tabla 4.2: LoRa Spreading Factors [69].

Spreading Factor	Bitrate	Range	Tiempo en el aire
SF7	5470 bps	2 km	56 ms
SF8	3125 bps	4 km	100 ms
SF9	1760 bps	6 km	200 ms
SF10	980 bps	8 km	370 ms
SF11	440 bps	11 km	740 ms
SF12	290 bps	14 km	1400 ms

Los SF más bajos significan *chirps* más rápidos y, por tanto, una mayor velocidad de transmisión de datos. Por cada aumento del SF, la velocidad de barrido de los *chirps* se reduce a la mitad, al igual que la velocidad de transmisión de los datos (ver Figura 4.4).

Finalmente, en la Tabla 4.3 podemos observar algunas otras especificaciones de LoRa.

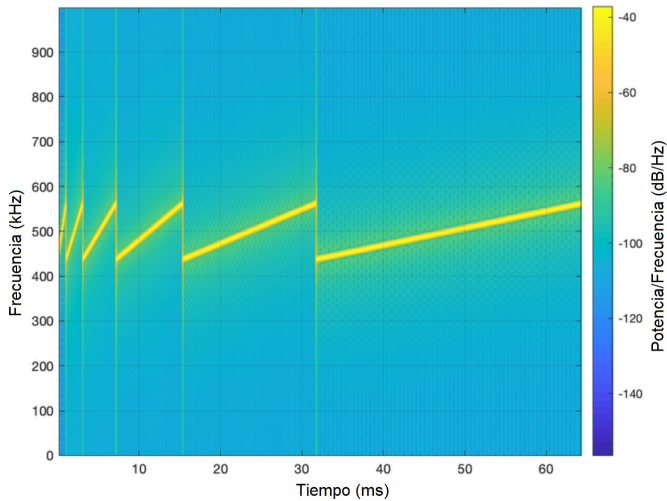


Figura 4.4: Comparación de Spreading Factors de LoRa: SF 7 a SF 12 [70].

Tabla 4.3: Especificaciones técnicas de la tecnología LoRa [69].

Especificaciones	LoRa
Modulación	CSS
Banda	Sub-GHz ISM-EU (433, 868 MHz), US (915 MHz), Asia (430 MHz).
Data rate	0.3 - 37.5 kbps (LoRa) y 50 kbps (FSK)
Rango	5 km (entorno urbano) 15 km (entorno rural)
MAC	Unslotted ALOHA
Payload	Hasta 250 B (dependiendo del SF y la región)
Autenticación y encriptación	AES 128 b
Eficiencia energética	Muy alta
Escalabilidad	Si
Movilidad	Si

4.2.2 LoRaWAN

Long-Range Wide Area Network (LoRaWAN) define el protocolo de comunicaciones y la arquitectura de la red, mientras que, LoRa define la capa física que garantiza los enlaces a largas distancias. El protocolo determina la seguridad, la capacidad de la red y la calidad del servicio [71]. La capa MAC de LoRaWAN proporciona el mecanismo de control de acceso al medio que permite la comunicación entre múltiples dispositivos y los *gateways* de la red.

La arquitectura de una red LoRaWAN tiene una topología en estrella, donde los dispositivos finales solo pueden comunicarse con los *gateway* LoRaWAN y no directamente entre sí. Varios *gateways* pueden estar conectados a un servidor de red central. Los *gateway* LoRaWAN únicamente se encargan de reenviar los paquetes de datos en “bruto” de los dispositivos finales hacia el servidor de red, encapsulándolos en paquetes TCP/IP. Por último, la comunicación termina en los servidores de aplicaciones que pueden ser propiedad de terceros. La arquitectura de red LoRaWAN resultante se puede observar en la Figura 4.5.

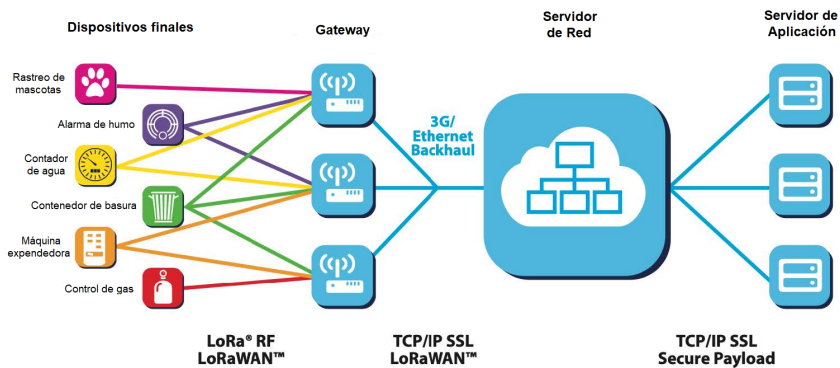


Figura 4.5: Arquitectura de red LoRaWAN [72].

Los componentes de una red LoRaWAN [63]:

- Un dispositivo final habilitado para LoRaWAN es un sensor o un actuador que se conecta de forma inalámbrica a una red LoRaWAN a través de un *gateway* utilizando la modulación de RF LoRa. En la mayoría de las aplicaciones, un dispositivo final es un sensor autónomo alimentado por baterías, que digitaliza las condiciones físicas y los eventos ambientales. Cuando se fabrican, se les asignan varios identificadores únicos. Estos identificadores se utilizan para activar y administrar de forma segura el dispositivo.
- Un *gateway* LoRaWAN recibe mensajes de RF modulados por LoRa desde cualquier dispositivo final a distancia de audición y reenvía estos mensajes de

datos al servidor de red, que está conectado a través de una red troncal IP. No hay una asociación fija entre un dispositivo final y un *gateway* específico. En su lugar, el mismo sensor puede ser atendido por múltiples *gateway* en la zona. Con LoRaWAN, cada paquete enviado por el dispositivo final será recibido por todos los *gateway* que estén a su alcance. Esto reduce significativamente la tasa de error de los paquetes (ya que las posibilidades de que al menos un *gateway* reciba el mensaje son muy altas).

- El servidor de red gestiona toda la red, controla dinámicamente los parámetros de la red para adaptar el sistema a las condiciones siempre cambiantes, y establece conexiones seguras AES de 128 bits para el transporte tanto de los datos de extremo a extremo como para el control de tráfico que fluye desde el dispositivo final hasta el servidor de red (y viceversa). También garantiza la autenticidad de cada sensor en la red y la integridad de cada mensaje.
- Los servidores de aplicación se encargan de manejar, gestionar e interpretar de forma segura los datos de las aplicaciones de los sensores. También generan todas las cargas útiles de la capa de aplicación que se envían a los dispositivos finales conectados.

IEEE 802.15.4 es el protocolo de seguridad de LoRaWAN [73]. Cada dispositivo final tiene su propia *App Key* [74]. Los dispositivos finales pueden operar en uno de los tres modos, dependiendo de su clase de dispositivos. Todos estos dispositivos deben soportar el funcionamiento en clase A. Los dispositivos de clase B deben soportar los modos de clase A y B, y los de clase C deben soportar los tres modos de funcionamiento. Estos modos de funcionamiento tienen que ver con la forma en que los dispositivos se comunican con la red.

- Clase A: El dispositivo final pasa la mayor parte del tiempo en un estado de inactividad, es decir, en modo *sleep*. Cuando se produce un cambio en el entorno relacionado con lo que el dispositivo está programado para supervisar, se “despierta” e inicia un enlace ascendente (*uplink*), transmitiendo los datos sobre el cambio de estado a la red (Tx). Posteriormente, el dispositivo espera una respuesta de la red, durante un segundo (esta duración es configurable). De no recibir un enlace descendente (*downlink*) durante esta ventana de recepción (Rx1), vuelve a modo *sleep* brevemente, “despertando” un momento después, escuchando de nuevo una respuesta (Rx2). De no recibir ninguna respuesta durante esta segunda ventana de Rx, el dispositivo vuelve a modo *sleep* hasta la próxima vez que tenga datos que comunicar. Los dispositivos finales son quienes inician la comunicación con la red.
- Clase B: Se considera una mejora sobre la clase A, ya que ofrece oportunidades programadas con regularidad y tiempo fijo para que los dispositivos finales reciban enlaces descendentes (*downlink*) de la red, lo cual genera

que los dispositivos finales de clase B sean adecuados tanto para supervisar sensores como actuadores. Estos dispositivos ofrecen ventanas de recepción programadas regularmente, además de las que se abren cada vez que se envía un enlace ascendente (*uplink*) de clase A al servidor.

- Clase C: Los dispositivos finales de esta clase están siempre “despiertos”, es decir, no dependen de la energía de la batería, por ejemplo, farolas, contadores eléctricos, etc. Estos dispositivos siempre están a la escucha de los mensajes de enlace descendente (*downlink*), a menos que estén transmitiendo un enlace ascendente (*uplink*). Como resultado, ofrecen la menor latencia para la comunicación desde el servidor a un dispositivo final.

LoRaWAN no admite comunicación de dispositivo a dispositivo. Si dos dispositivos finales desean transmitir un paquete, entonces al principio el paquete será transferido desde el dispositivo transmisor al servidor de la red y luego desde el servidor al dispositivo receptor. La única manera de conectar dos dispositivos finales es a través de los *gateway*. En Figura 4.6 podemos observar un resumen de las clases de dispositivos finales LoRaWAN.

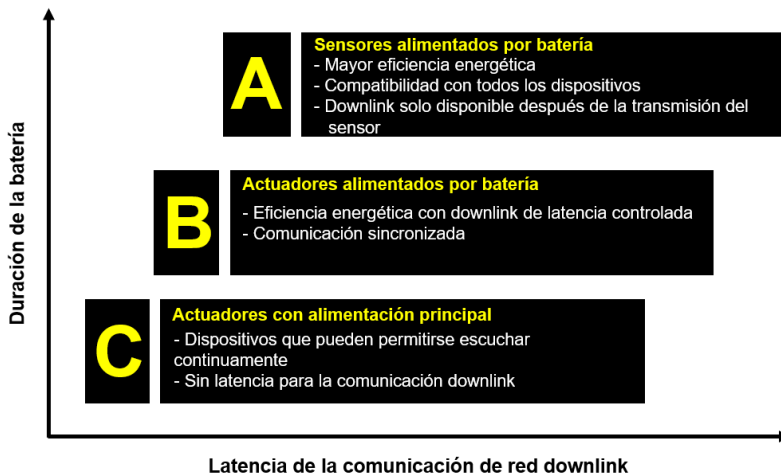


Figura 4.6: Clasificación de los dispositivos finales LoRaWAN [69].

LoRaWAN se utiliza para distintas aplicaciones, que van desde la monitorización de la salud y bienestar [75, 76], monitorización de la agricultura [77, 78, 79, 80], redes de sensores inalámbricos [81, 82], monitorización del tráfico [83] y aplicaciones de ciudades inteligentes. Se utilizan principalmente para aplicaciones no sensibles a la latencia y para aplicaciones en las que se necesitan despliegues a gran escala.

4.3 MQTT

*Message Queuing Telemetry Transport (MQTT)*³ es un protocolo de mensajería estándar OASIS⁴ para IoT y M2M. Está diseñado como un transporte de mensajería de publicación/suscripción extremadamente ligero que es ideal para conectar dispositivos remotos con una pequeña huella de código y un ancho de banda de red mínimo [84].

Permitir que los dispositivos de monitoreo empleados en la industria del gas y petróleo enviaran sus datos a servidores remotos, fue el propósito principal de su creación en 1999. MQTT se estandarizó como código abierto en 2013 por *Organization for the Advancement of Structured Information Standards (OASIS)*, y esta misma organización aún gestiona el estándar [85].

El principio del modelo de comunicación *publish/subscribe (pub/sub)* [86], es que los componentes que están interesados en consumir cierta información registran su interés. A este proceso de registro de interés se le denomina suscripción, por lo que a la parte interesada se le denomina “suscriptor”. Los componentes que quieren producir cierta información lo hacen publicándola, por tanto, se les denomina “publicadores” [87]. Por lo tanto, los publicadores generan información de algún tema (*topic*) y los suscriptores la consumen si es de su interés. A la entidad que se encarga de que los datos lleguen de los publicadores a los suscriptores es conocida como *broker* y se encarga de coordinar las suscripciones (ver Figura 4.7).

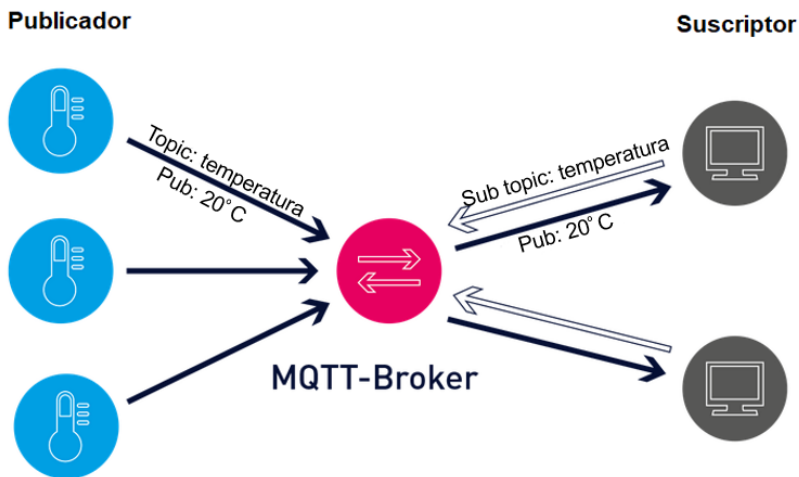


Figura 4.7: Arquitectura de MQTT.

³<https://mqtt.org/>

⁴<https://www.oasis-open.org/>

El *broker* es quien se encarga de controlar las suscripciones y el flujo de los datos, en la Figura 4.8 podemos observar la interacción de un sistema MQTT.

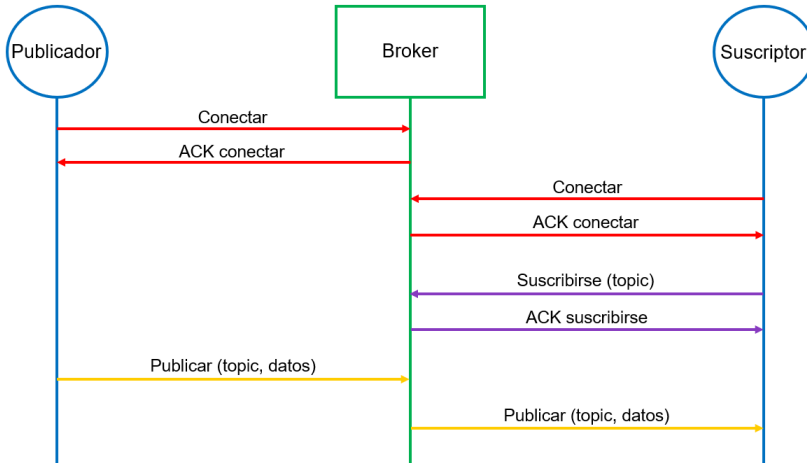


Figura 4.8: Secuencia de interacciones de publicación/suscripción.

Los *topic* en MQTT se estructuran en una jerarquía similar a la de las carpetas y archivos en un sistema de archivos, utilizando la barra diagonal (/) como delimitador. Lo cual permite crear una estructura de nomenclatura fácil de usar y autodescriptiva. Por lo tanto, en el ejemplo de la Figura 4.9, de estar suscritos, recibiríamos los datos de la temperatura, de la habitación principal de la casa.



Figura 4.9: Estructura de los MQTT topics.

MQTT es un protocolo ligero de publicación/suscripción que puede funcionar en dispositivos integrados y plataformas móviles. Se ejecuta sobre TCP/IP. Es una arquitectura cliente-servidor, donde los clientes (publicador, suscriptor) siempre establecen la conexión de red con el servidor (*broker*) [88].

Los clientes pueden:

- Publicar mensajes para los usuarios interesados.
- Suscribirse a un *topic* de su interés para recibir mensajes.
- Abandonar un *topic* para procesar los datos.

4. TECNOLOGÍAS RELACIONADAS

- Darse de baja del *broker*.

El servidor puede:

- Aceptar las peticiones de los clientes.
- Recibir los mensajes publicados por los usuarios.
- Procesar diferentes peticiones como suscripción y dar de baja a usuarios.
- Después de recibir los mensajes del publicador, enviarlos a los suscriptores interesados.

MQTT tiene un tamaño de mensaje pequeño y bien definido, con esto logra minimizar sus transmisiones. Los mensajes tienen una cabecera fija de 2 bytes, es posible utilizar una cabecera opcional que contenga el nombre de usuario y contraseña que facilita la autenticación, pero incrementa el tamaño del mensaje. El *payload* del mensaje está limitado a 256 MB (ver Figura 4.10) . Por lo tanto, los diseñadores de redes deben elegir entre minimizar la transmisión de datos y maximizar la fiabilidad, empleando tres niveles diferentes de calidad de servicio (*QoS*).

- **QoS 0:** Ofrece la cantidad mínima de transmisión de datos. Cada mensaje es entregado a un suscriptor una vez, sin confirmación, por lo tanto, no hay forma de saber si los suscriptores recibieron el mensaje. Este nivel asume que la entrega fue exitosa, por lo que, los mensajes no se almacenan para ser entregados posteriormente a los clientes desconectados.
- **QoS 1:** El *broker* intenta entregar el mensaje y espera una respuesta de confirmación del suscriptor. Si no hay mensaje de confirmación dentro de un período de tiempo especificado, el mensaje se envía de nuevo.
- **QoS 2:** El cliente y el *broker* emplean un protocolo de enlace de cuatro pasos que garantiza no solo que el mensaje sea recibido, sino que lo haga una única vez.

Bit	7	6	5	4	3	2	1	0	
Byte 1	Tipo de mensaje				DUP Flag		Nivel QoS		Ret
Byte 2	Longitud restante								
Cabecera opcional									
Payload									

Figura 4.10: Mensaje MQTT [89].

Existe un gran número de *brokers* MQTT, por mencionar algunos:

- Mosquitto [90] es un *broker* de mensaje de código abierto con licencia EPL/EDL, desarrollado por la Fundación Eclipse.
- HiveMQ [91] es una plataforma de mensajería escalable y con licencia comercial basada en MQTT, desarrollada por HiveMQ GmbH.
- EMQ X [92] es un broker de código abierto, altamente escalable y soporta *clustering*. Fue desarrollado por EMQ *Inc.*
- *IBM Watson IoT Platform Message Gateway* [93] es un servicio de mensajería escalable con licencia comercial basado en MQTT y ofrece bibliotecas de clientes de MQTT.
- RabbitMQ [94] es un broker de mensajes basado en erlang que soporta la versión 3.1.1 de MQTT, está desarrollado por Pivotal Software y utiliza una licencia MPL 1.1.

4.4 Edge Computing

Hay un incremento significativo de los datos generados por los dispositivos móviles y IoT. Estos dispositivos (incluidos *smartphones*, *wearables* y los sensores) tienen como característica común sus limitados recursos computacionales y energéticos (ver Figura 4.11). Estas limitaciones actualmente se resuelven descargando el procesamiento y almacenamiento de los dispositivos con recursos limitados a la nube (*cloud computing*) [95]. Esta es una solución ideal para la descarga de cómputo debido a su naturaleza escalable y bajo demanda (ver Figura 4.12). Sin embargo, una de las principales desventajas es que los recursos de estas soluciones se alojan en grandes centros de datos construidos en lugares alejados de los usuarios finales. Lo cual representa una elevada latencia de comunicación entre la nube y los usuarios. Con el crecimiento en la cantidad de datos que se intercambian, añade una tensión considerable a los enlaces de red.

Con la intención de ayudar a resolver estos problemas, se han propuesto los conceptos de *edge computing* [98, 99] y *fog computing* [100, 101]. *edge computing* (computación en la frontera), sugiere que, los recursos informáticos se coloquen cerca de los dispositivos finales (o incluso junto a ellos), lo cual genera la reducción de la latencia de la comunicación. Además, los datos intensivos de la red pueden ser procesados y analizados a un “salto” de distancia de los dispositivos finales y de esta forma reducir la demanda de ancho de banda en los enlaces de red de datos distantes. El poder procesar y almacenar datos cerca de los dispositivos que los generan permite nuevos servicios [102, 103, 104]. Las plataformas de *edge computing* admiten la movilidad de los dispositivos y las aplicaciones distribuidas geográficamente [100].

4. TECNOLOGÍAS RELACIONADAS

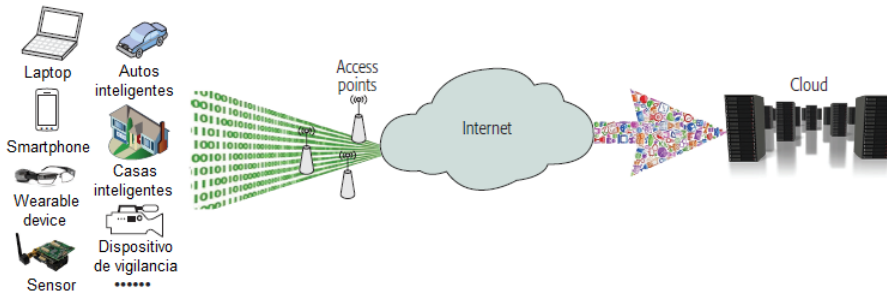


Figura 4.11: Arquitectura tradicional IoT [96].

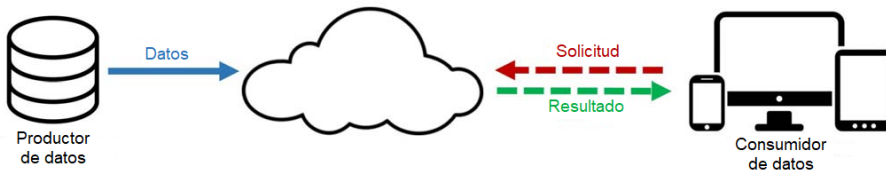


Figura 4.12: Paradigma de computación en la nube [97].

Siendo la movilidad y la distribución geográfica las características claves de los despliegues de IoT que pueden beneficiarse del *edge computing* (ver Figura 4.13).

Shi et al. [97, 105, 106] introdujeron la aparición del concepto de *edge computing*: “*Edge computing* es un nuevo modo de computación de ejecución en la frontera de la red”.

Zha et al. [107] propusieron la definición: “*edge computing* es un nuevo modelo de computación que une los recursos que están cerca del usuario en la distancia geográfica o de red para proporcionar servicios de computación, almacenamiento y red para el servicio de las aplicaciones”.

En otras palabras, *edge computing* consiste en migrar *cloud computing*, la computación, las capacidades de almacenamiento y los recursos de red, y proporcionar servicios inteligentes en *edge* para satisfacer las necesidades críticas de la industria de TI en enlace ágil, negocio en tiempo real, optimización de datos, inteligencia de aplicaciones, seguridad, privacidad, y además cumple con los requisitos de baja latencia y gran ancho de banda en la red.

Surgieron diferentes propuestas de arquitecturas para la creación de plataformas de *edge computing*. Un análisis de las arquitecturas revela que el *edge* de la red no está claramente definido, y los nodos que se espera que participen en el *edge* pueden variar. La terminología empleada para describir el *edge* difiere mucho, ya que el mismo término se utiliza para definir diferentes arquitecturas y funcionalidades

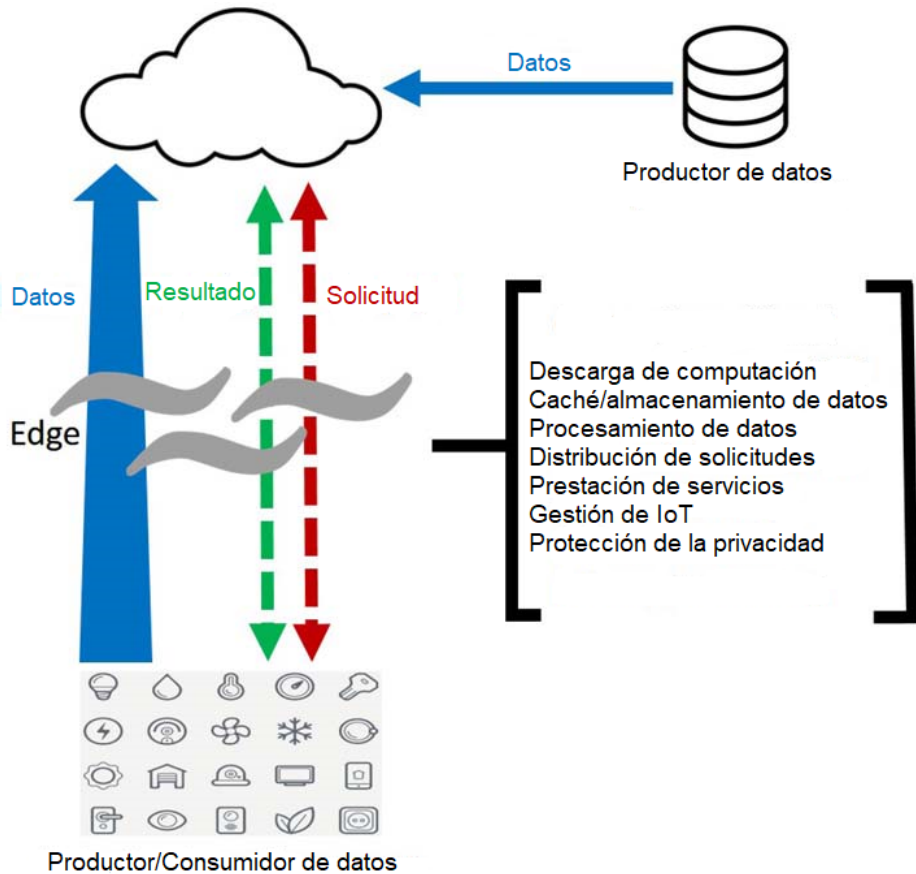


Figura 4.13: Paradigma de *Edge Computing* [96].

[108, 109, 110]. La categorización se basa en las características comunes de las implantaciones. Aunque, en la práctica, las características de una categoría pueden utilizarse en combinación con otras. Las tres clasificaciones son:

- Servidores ricos en recursos desplegados cerca de los dispositivos finales (ver Figura 4.14a).
- Nodos *edge* heterogéneos, incluyendo los dispositivos finales (ver Figura 4.14b).
- Federación de recursos en el *edge* y centros de datos centralizados (ver Figura 4.14c).

4. TECNOLOGÍAS RELACIONADAS

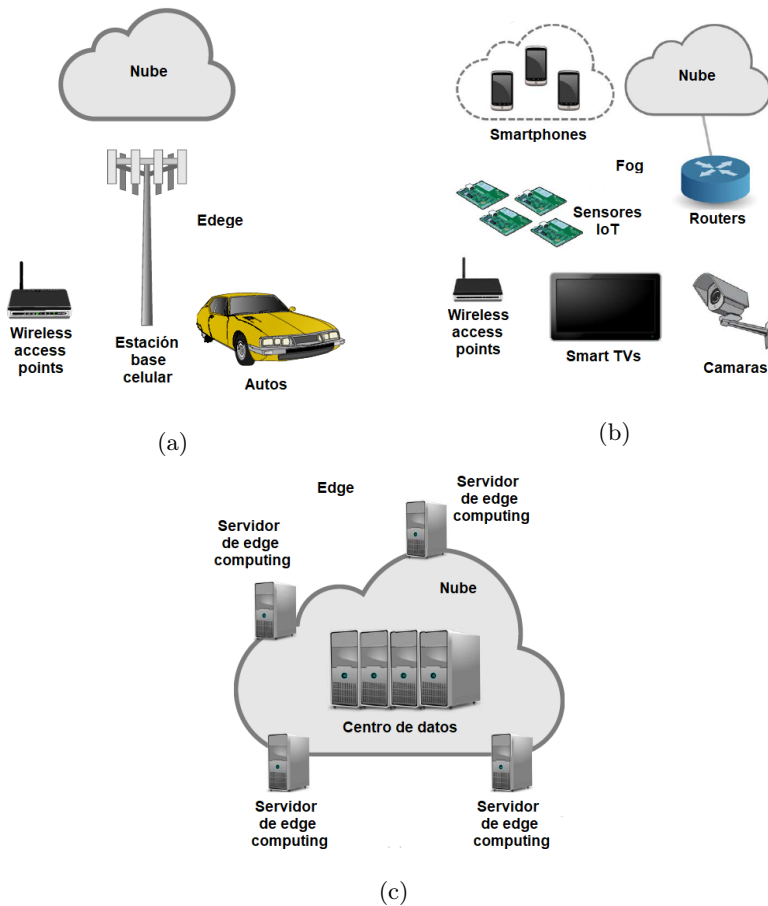


Figura 4.14: Clasificación de plataformas *edge computing* [111].

Capítulo 5

Solución propuesta

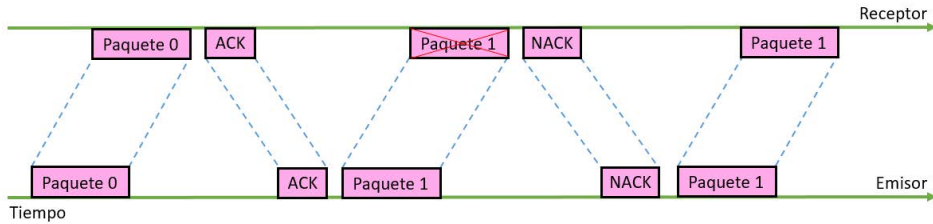
Nuestra propuesta se centra en una arquitectura basada en Redes Comunitarias (RC), donde la prioridad es proveer de un medio de comunicación digital a las zonas rurales.

Como se ha mencionado en capítulos anteriores, para los proveedores de servicio representa un reto económico el despliegue de infraestructura en las comunidades aisladas. Por esta razón, planteamos una solución basada en dispositivos de bajo consumo energético y de bajo coste.

Estos dispositivos centrales o “*Hubs*” deben ser capaces de establecer enlaces de comunicación de larga distancia, entre comunidades o entre comunidades y ciudades. Los “*hubs*” generarán un punto de conectividad a donde los dispositivos finales cercanos pueden enlazarse y mediante una interfaz *REpresentational State Transfer (REST)* enviar mensajes, para que puedan ser transmitidos. Los “*hubs*” también se pueden emplear para recolectar datos de sensores u otras fuentes externas y transmitirlos mediante la misma interfaz *REST*.

5.1 Protocolo de comunicaciones

Para lograr la comunicación entre los “*hubs*” y manejar el envío de mensajes, proponemos un protocolo *unicast* fiable basado en un enfoque *Automatic Repeat-reQuest (ARQ)* clásico basado en *stop-and-wait* (ver Figura 5.1). El protocolo garantiza que no se pierda información debido a pérdida de paquetes y que se reciban en el orden correcto. Nombramos a este protocolo *LoRa based Content Transfer Protocol (LoRaCTP)*.

Figura 5.1: Esquema de protocolo *Stop-and-Wait*.

5.1.1 Envío de mensajes

Antes de transmitir cualquier mensaje, el protocolo agrega todos los datos en un *JavaScript Object Notation (JSON)*, los codifica y los segmenta en paquetes de tamaño específico, cuando el mensaje ha sido recibido por completo, el protocolo realiza un proceso de decodificación y reestructura el *JSON*.

Cuando un usuario envía un mensaje, el *hub* “aprende” que este usuario está conectado a través de él y crea una entrada en su registro de usuarios.

Para el envío de mensajes, el sistema responde a dos supuestos, si es un mensaje *broadcast* o un mensaje a un usuario específico. Si se envía un mensaje para todos los usuarios de la red, el paquete a transmitir se estructura con nombre de usuario *broadcast*, se agrega el mensaje e inicia el proceso de transmisión. En el caso de un envío de mensaje para un solo usuario, como primer paso, el protocolo debe descubrir dónde se encuentra el usuario destino. Para ello, el *hub* origen envía un mensaje *broadcast* con dirección de destino “FFFFFFF” que será recibido por todos los dispositivos de la red, “preguntando” en cuál de ellos se encuentra registrado el usuario a quien va dirigido el mensaje. El mensaje de búsqueda de usuario será enviado tres veces en un lapso de veinte segundos, si ninguno de los dispositivos de la red responde a este mensaje se muestra una alerta de destino no encontrado. Los dispositivos que reciben el mensaje, realizan una búsqueda en su lista de usuarios. El *hub* que tiene registrado al usuario destino responde al dispositivo solicitante, con un mensaje que incluye su dirección *Media Access Control (MAC)*, que durante la transmisión del mensaje identifica al dispositivo destino (ver Figura 5.2). A este proceso se le llama de descubrimiento y se realiza cada vez que se envía un mensaje. La estructura del paquete de descubrimiento que utiliza el protocolo es muy sencilla, con dos bits para datos de control y un campo de 32 bytes para almacenar la dirección destino y el nombre del usuario que se busca.

Una vez que el nodo emisor “conoce” al nodo receptor, el mensaje se segmenta en pequeños paquetes para transmitirse usando el protocolo LoRaCTP. Los paquetes están formados por una cabecera y *payload* que en total serán 230 bytes (ver Figura 5.3).

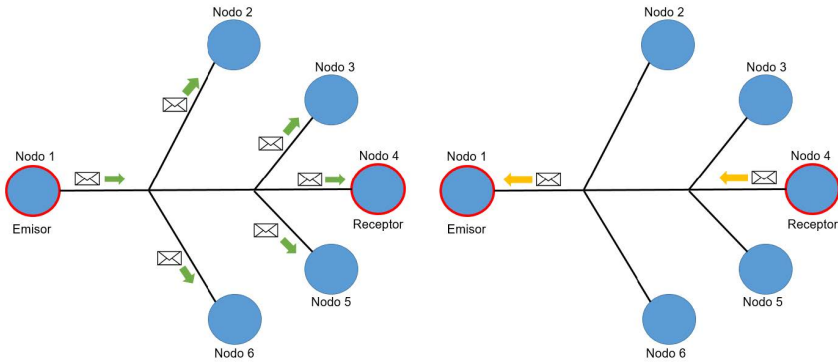


Figura 5.2: Proceso de descubrimiento.

La cabecera del mensaje está compuesta por 20 de los 230 bytes, de tal manera que estará compuesta por 8 bytes de la dirección MAC del nodo emisor, 8 bytes para la dirección del nodo receptor, 1 byte como identificador de último paquete (*flags*) y el *checksum* será de 3 bytes, el resto del paquete contendrá datos del mensaje, como el nombre del usuario origen, nombre del usuario destino y el mensaje.

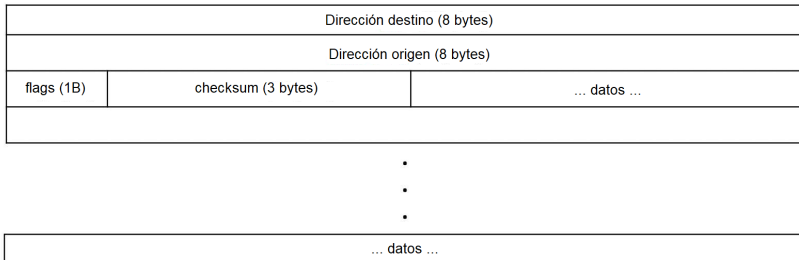


Figura 5.3: Estructura del paquete.

En la Figura 5.4 podemos observar de forma gráfica los pasos que sigue el sistema para poder enviar mensajes.

5.1.2 Recepción de mensajes

El proceso de recepción de mensajes es un poco más extenso, ya que en este proceso se pueden ejecutar dos tipos de funciones, puede ser una función LoRa o una función *HyperText Transfer Protocol (HTTP)* y dependiendo de ellas se realizan procesos diferentes. El dispositivo está configurado para que las acciones del usuario le indiquen que función debe realizar. Si se recibe una función LoRa,

5. SOLUCIÓN PROPUESTA

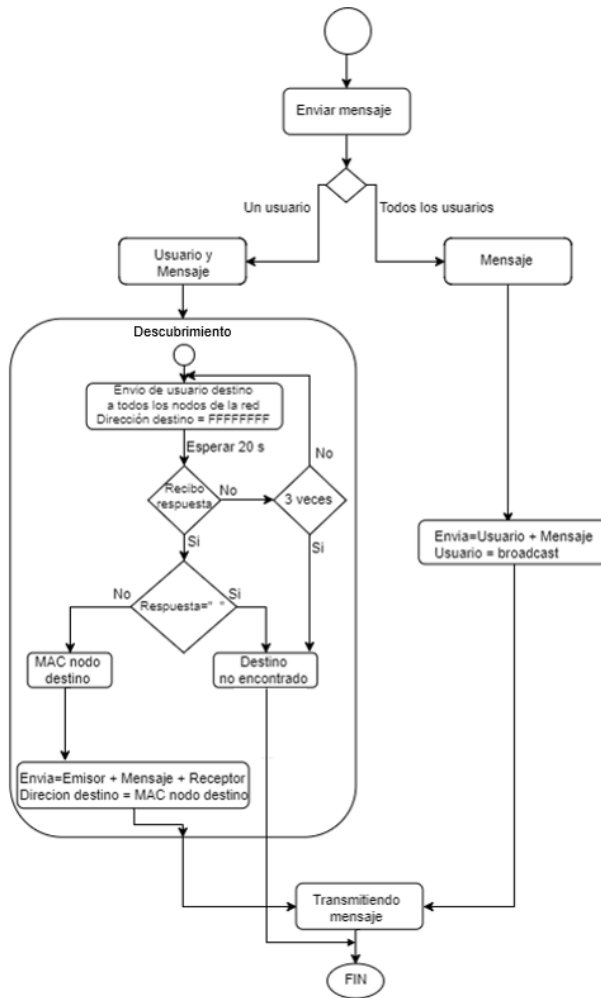


Figura 5.4: Diagrama del proceso de envío de mensajes.

el sistema verifica si se ha recibido una dirección *broadcast* (FFFFFFF), lo que indica que se está buscando a un usuario en la red (proceso de descubrimiento), el dispositivo comprobará su lista de usuarios registrados en busca de ese usuario, en caso de no encontrarlo simplemente se desecha la petición; en caso contrario, el dispositivo responde al mensaje enviando su dirección MAC para que el dispositivo origen “conozca” quien será el dispositivo destino, con el cual realizará el proceso de transmisión hasta que el mensaje completo sea recibido. Si se recibe una función HTTP, el sistema verifica que el paquete recibido contenga una estructura HTTP que reconoce, de no ser así la petición es desechada; en caso contrario, se procede a descomponer los paquetes y verifica que tipo de petición se está solicitando. De ser una petición tipo “*GET*” se procede a mostrar la página de registro, para que el usuario se identifique (ver Figura 6.3a); de ser una petición “*POST*”, le indica al sistema que el usuario ha presionado alguno de los botones contenidos en la página, que puede realizar alguna de las siguientes acciones:

- Registro: Este proceso es obligatorio para el usuario, ya que proporciona el ID con el cual se le conocerá en la red, además será útil tanto para enviar como para recibir mensaje. Si es la primera vez que ese usuario se registra en el dispositivo, se crea una entrada en la lista de usuarios local, para almacenar su ID.
- Enviar mensaje: Se debe indicar el ID del usuario a quien va dirigido, así como el mensaje a transmitir. Se realiza el proceso de descubrimiento y una vez que se “conocen” el dispositivo origen y destino, se envía el mensaje. El dispositivo destino almacena el nombre de usuario destino y el mensaje.
- Consultar mensajes: Se realiza una búsqueda en la lista de usuarios, para obtener el índice de ese ID, después se obtienen los mensajes almacenados en ese índice de la lista de mensajes, finalmente, se consulta la lista de mensajes *broadcast*.
- Mensaje *broadcast*: A este tipo de mensajes no es necesario agregar el nombre de usuario destino, el sistema se encarga de incluir el ID de usuario *broadcast*, junto con una bandera que idéntica el tipo de mensaje. Si el valor de la bandera es “1” significa que es un mensaje *broadcast* y si el valor es “0” será para un usuario específico.

En caso de que el sistema reciba un paquete “basura”, no se reconoce como ninguna de las funciones para ser procesada y se desecha la petición. En la Figura 5.5 podemos observar de forma gráfica los pasos que seguirá el sistema para realizar el proceso de recepción de mensajes.

Planteamos agregar dispositivos con acceso a Internet que funcionarán como *gateway*, que permitirán potenciar el alcance del sistema de mensajería mediante aplicaciones estándar para enviar los mensajes vía Internet.

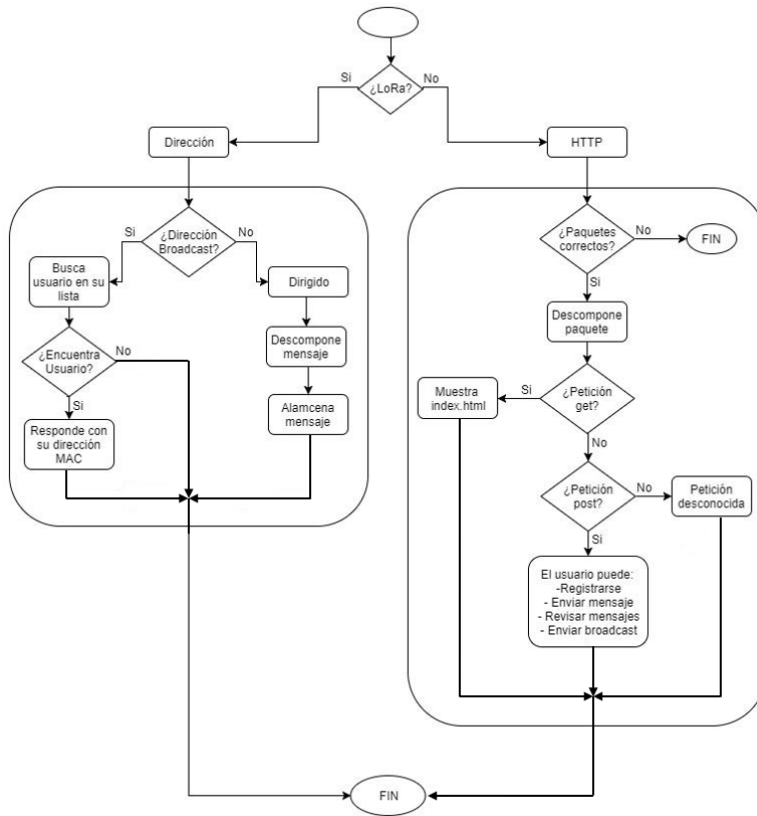


Figura 5.5: Diagrama del proceso de recepción de mensajes.

5.2 Recolección de datos de fuentes externas

Esta solución debe ser capaz de manejar datos de diferentes fuentes externas, por ejemplo, debe poder generar y procesar clips de audio, ya que en las zonas rurales existe un alto índice de analfabetismo, de esta forma los habitantes que no sepan leer y escribir podrán utilizar nuestra solución mediante mensajes de voz para acceder a información y servicios remotos.

También integramos el soporte para aplicaciones de detección ordinarias basadas en cualquier tipo de sensores, como por ejemplo, sensores ambientales para pronóstico del tiempo, sensores químicos, de nivel, de imágenes, etc.

La ubicación de estos dispositivos, cerca de los “hubs” de nuestra arquitectura, les permitirá enviar datos a largas distancias e integrarse en sistemas de publicación-suscripción (ver Figura 5.6).

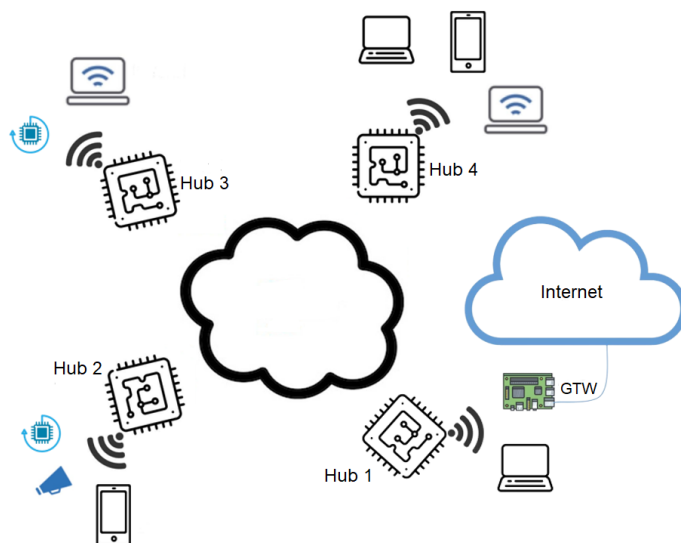


Figura 5.6: Arquitectura general de la solución.

5.3 Nodos edge

Además, proponemos el diseño de una arquitectura frugal, compuesta por un nodo central y nodos basados en microservicios, para proporcionar servicios informáticos en escenarios con recursos limitados y poca conexión. Los nodos se encargarán de administrar y procesar los datos provenientes de fuentes externas. La estructura básica de esta arquitectura se basa en un nodo central denominado “agregador” y nodos que tendrán un canal dedicado con el agregador para permitir transmisiones de largo alcance con bajo consumo de energía (ver Figura 5.7).

La solución de nodos basados en microservicios, permite que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Y en ellos pueden coexistir varios microservicios que intercambian datos con otros microservicios.

El agregador coordina el flujo de datos entre los servicios en los nodos *edge* con los de la nube. Utiliza una técnica de sondeo para: (1) obtener una mayor flexibilidad en el uso del canal, (2) permitir el uso del modo de suspensión en el nodo *edge* y (3) manejar los periodos de desconexión.

Los nodos *edge* en esta arquitectura se denominan “FUDGE” (nodo FrUgal eDGE). Un FUDGE se basa en microservicios, un enfoque arquitectónico en el que el *software* se compone de pequeños servicios independientes que se comunican a través de *Application Programming Interfaces (APIs)* bien definidas. Esta arquitectura hace que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar.

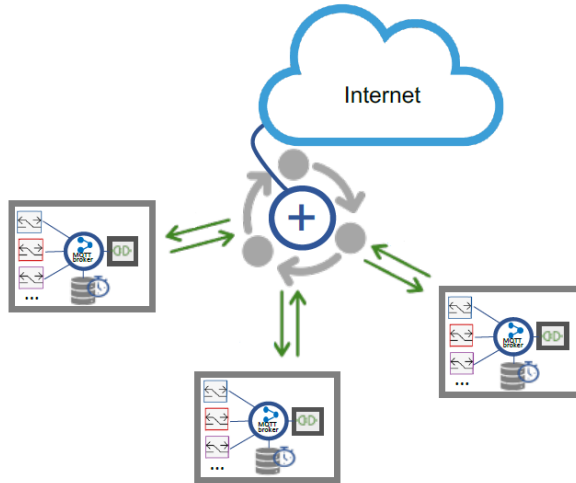


Figura 5.7: Diseño de arquitectura frugal.

En un FUDGE, varios microservicios coexisten e intercambian datos con otros microservicios utilizando una API basada en el sistema publicación/suscripción MQTT. Los datos, indicados como contenido, ingresan localmente al FUDGE a través de los denominados “proxys de contenido”. Se pueden encontrar tres componentes principales en un FUDGE: un broker MQTT, un administrador de persistencia y un reenviador de contenido. El *broker* MQTT maneja el flujo del contenido dentro del nodo. El elemento del administrador de persistencia se encarga de almacenar el contenido etiquetado como persistente en una base de datos de series de tiempo.

Esto permite mantener la evolución temporal del sistema, conservando todos los datos necesarios para manejar el funcionamiento asíncrono o posibles desconexiones. Además, las herramientas de análisis de datos se pueden usar para extraer métricas que tengan sentido a partir de los datos recopilados y las aplicaciones de monitoreo a través de paneles personalizables. Finalmente, el reenviador de contenido se encarga de hablar con el agregador mediante el protocolo LoRaCTP.

El contenido enviado se cifra mediante el certificado del agregador. El certificado debe preinstalarse manualmente en los nodos *edge*. Este enfoque garantiza tanto la privacidad en la comunicación como que solo el agregador autorizado puede manejar los datos entrantes.

El agregador coordina el flujo de datos con los nodos *edge* mediante un enfoque de sondeo. Se eligió el sondeo debido a la dificultad en LoRa para detectar y manejar adecuadamente las colisiones, y por la necesidad, dada la baja banda

ancha que ofrece LoRa para garantizar la confiabilidad de la transmisión a máxima velocidad. Además, el sondeo no requiere *hardware/software* más complejo para administrar el establecimiento de enlaces con los dispositivos *edge*, en general, este enfoque permite: (1) una mayor flexibilidad en el uso del canal, (2) el uso del modo de suspensión en el nodo *edge* y (3) facilitar el manejo de los períodos de desconexión.

Durante el sondeo, una vez que se selecciona un nodo *edge*, sigue una secuencia de *push/pull* donde primero el nodo *edge* realiza un *push* de los datos al agregador y una segunda fase donde los nodos de *edge* realizan un *pull* de los datos almacenados en el agregador. Básicamente, en la fase de inserción, el reenviador de contenido envía todo el contenido etiquetado como *Global* que el administrador de coherencia mantiene almacenado. En la fase de extracción, el agregador envía el contenido que mantuvo almacenado desde la fase de sondeo anterior. El agregador devuelve contenido cuyos temas se registraron desde cualquier servicio en el nodo *edge*. La idea detrás de esto es dar la posibilidad a los procesos locales de recibir tanto (1) respuestas a solicitudes enviadas localmente como (2) datos de otros servicios en la nube.

Los nodos *edge* o FUDGE, están compuestos por tres elementos básicos: un *broker* MQTT, un administrador de persistencia y un reenviador de contenido (ver Figura 5.8). Además de estos tres elementos principales, hay tantos “proxys de contenido” como las diversas fuentes de contenido que maneja este FUDGE.

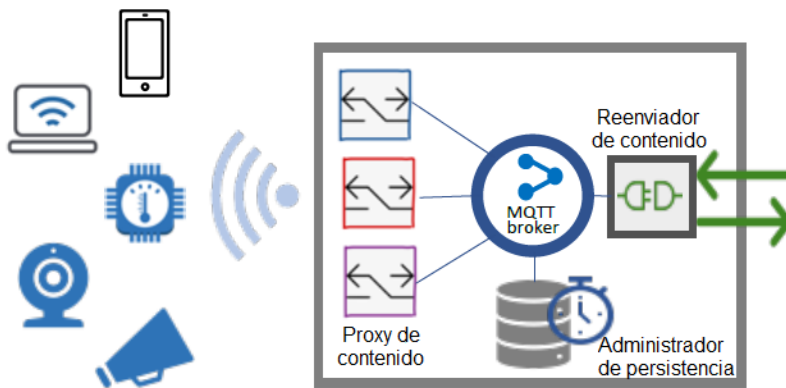


Figura 5.8: Estructura FUDGE (frugal edge-node).

El *broker* MQTT es el núcleo de FUDGE y maneja el flujo del contenido dentro del nodo. El elemento del administrador de persistencia se encarga de almacenar el contenido etiquetado como persistente en una base de datos de series

de tiempo. Esto permite mantener la evolución temporal del sistema, conservando todos los datos necesarios para manejar el funcionamiento asíncrono o posibles desconexiones. Además, las herramientas de análisis de datos se pueden utilizar para visualizar y extraer métricas de los datos recopilados o implementar paneles de control personalizados. Finalmente, el reenviador de contenido se encarga de hablar con el agregador usando LoRaCTP para intercambiar todo el contenido requerido, como se ha descrito anteriormente.

Dado que MQTT es el núcleo del sistema, definimos un formato estándar para los *topics*. La estructura utilizada en el sistema es la siguiente:

```
<device id>/<service id>/scope/persistence/...others ...
```

Donde:

- **<device id>**: identifica el dispositivo del nodo FUDGE específico.
- **<service id>**: >: identifica el servicio que proporciona los contenidos. Los proveedores de contenido pueden ser cualquier cosa, desde simples sensores de temperatura hasta cámaras, sistema de mensajería, etc.
- **scope**: indica si el contenido debe ser utilizado solo por otros servicios localmente dentro del nodo FUDGE (“L”) o si debe reenviarse al agregador (“G”).
- **persistence**: se usa para indicar si el contenido debe ser manejado o no por el administrador de persistencia. Una (“P”) hará que el contenido se vuelva persistente y una (“N”) indica que no se requiere ninguna acción. Un tercer valor, (“X”), se usa para realizar búsquedas en el repositorio de persistencia para recuperar datos, y usa los campos **others** para indicar si se trata de una solicitud o una respuesta para una búsqueda, u otros detalles.
- **others**: Se pueden agregar más etiquetas si el servicio específico lo requiere.

El contenido en sí debe estructurarse utilizando el formato de intercambio de datos JSON de acuerdo con el formato de la Lista 5.3.1.

<measurement_id> indica el conjunto específico de valores. Puede haber varias fuentes que generen datos sobre el mismo conjunto, por ejemplo, varias estaciones meteorológicas en un área cercana al nodo edge que proporciona los valores climáticos. La diferencia entre **<tag_value>** y **<field_value>** viene dada por la base de datos de series de tiempo subyacente. Básicamente, una medida que cambia con el tiempo debería ser un campo y los metadatos sobre la medida deberían ser una etiqueta, por ejemplo, los valores de presión y temperatura son campos y el nombre de la estación meteorológica una etiqueta.

```
1 payload = {
2     "measurement": <measurement_id>,
3     "tags": {
4         "tag1": <tag1_value>,
5         ...,
6         "tagn": <tagn_value>
7     },
8     "fields": {
9         "field1": <field1_value>,
10        ...,
11        "fieldn": <fieldn_value>
12    }
13 }
```

Lista 5.3.1: Estructura JSON del contenido.

Capítulo 6

Resultados

Diseñamos una arquitectura de redes comunitarias que nos permite realizar enlaces a largas distancias utilizando LoRa, mediante un protocolo de comunicaciones y una interfaz *REST*, que brinda la posibilidad de transmitir mensajes de texto y audio, así como recolectar datos de sensores. Ampliamos el alcance de nuestra propuesta, permitiendo el envío de mensajes a través de aplicaciones de mensajería estándar con acceso a Internet. También hemos integrado dispositivos capaces de recibir y procesar los datos de sensores para después compartir esta información con otros usuarios mediante tecnologías de publicación/suscripción. Diseñamos una arquitectura genérica *edge/fog* frugal para proporcionar servicios informáticos en escenarios con recursos limitados y con poca conexión. Por último, presentamos algunos resultados experimentales con los que medimos el rendimiento y las prestaciones de nuestra solución.

6.1 Sistema de mensajería

Una de las partes más importantes de nuestra propuesta radica en el sistema de mensajería, la arquitectura general del sistema está formada por dispositivos llamados *hub*, a los cuales se conectan los dispositivos finales para enviar/recibir mensajes, los *hub* transmiten estos mensajes utilizando LoRa a los *hub* vecinos (ver Figura 6.1).

El núcleo de la arquitectura propuesta son los dispositivos dedicados *hub*, que crean un punto de conectividad dentro de una zona, para los dispositivos finales. Los *hub* tienen tanto un transceptor WiFi (IEEE 802.11b/g/n) como un transceptor LoRa.

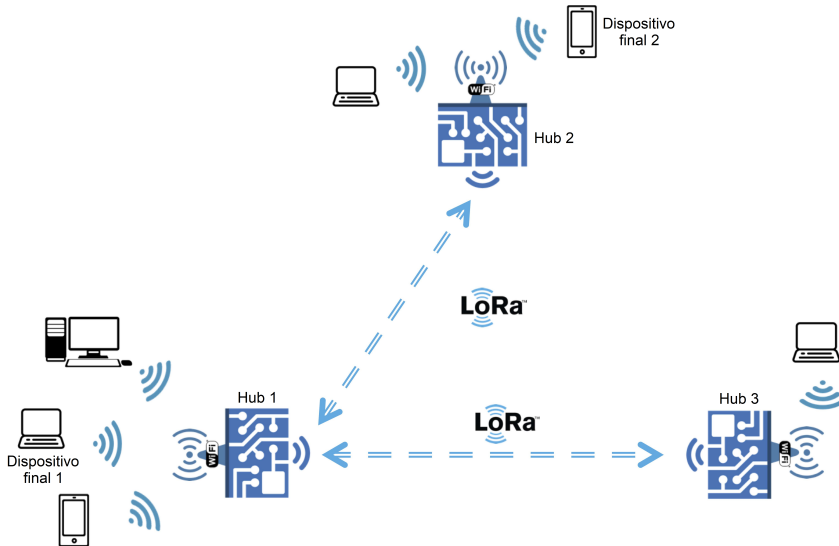


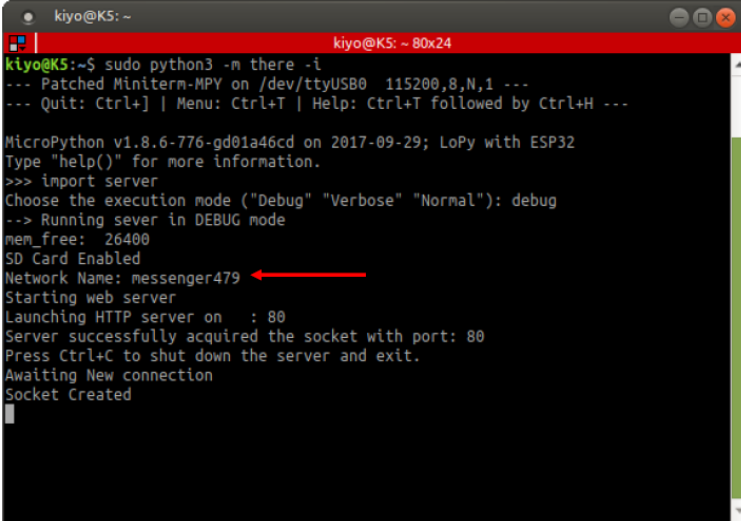
Figura 6.1: Arquitectura general del sistema de mensajería.

Para nuestra solución, los dispositivos *hub* que hemos empleado son los PyCom¹ LoPy4, ya que son dispositivos *IoT* de bajo coste, bajo consumo energético y además cuentan con diferentes tecnologías inalámbricas (WiFi, LoRa, Sigfox, Bluetooth). Estos dispositivos tienen un microprocesador Xtensa LX6 de doble núcleo de 32 bits, un *chipset* Espressif ESP32, 520 *Kilobyte (KB)*, de *Random Access Memory (RAM)*, una memoria flash de 8 *Megabyte (MB)*, además de ser programables con MicroPython.

Los *hub* funcionan como un *Access Point (AP)* WiFi estándar, para proporcionar conectividad a los dispositivos cercanos al difundir un *Service Set Identifier (SSID)* de red (ver Figura 6.2), a su vez como servidor web que permite a los usuarios acceder a una interfaz y mediante el protocolo *LoRa based Content Transfer Protocol (LoRaCTP)* se encarga de la transmisión de mensajes. Los *hub* reciben órdenes POST de los dispositivos conectados para enviar un mensaje o devolver los recibidos y almacenados localmente.

Desarrollamos el protocolo LoRaCTP usando MicroPython para que pueda ser ejecutado por los LoPy4, este se encarga de procesar las funciones *HTTP* controlando la interfaz web con la que el usuario interactúa, almacena/gestiona los usuarios y mensajes, realiza el proceso de descubrimiento, estructura los mensajes para poder ser transmitidos, segmenta los mensajes en paquetes de ser necesario, administra el envío/recepción de los paquetes que se transmiten mediante LoRa y reestructura los paquetes para poder mostrar dicho mensaje.

¹<https://pycom.io/>



```
kiyo@KS: ~  
kiyo@KS: ~ 80x24  
kiyo@KS:~$ sudo python3 -m there -i  
--- Patched Miniterm-MPY on /dev/ttyUSB0 115200,8,N,1 ---  
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---  
  
MicroPython v1.8.6-776-gd01a46cd on 2017-09-29; LoPy with ESP32  
Type "help()" for more information.  
>>> import server  
Choose the execution mode ("Debug" "Verbose" "Normal"): debug  
--> Running sever in DEBUG mode  
mem_free: 26400  
SD Card Enabled  
Network Name: messenger479  
Starting web server  
Launching HTTP server on : 80  
Server successfully acquired the socket with port: 80  
Press Ctrl+C to shut down the server and exit.  
Awaiting New connection  
Socket Created
```

(a) SSID vista LoPy4.

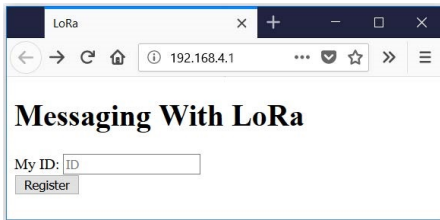


(b) SSID vista dispositivo final.

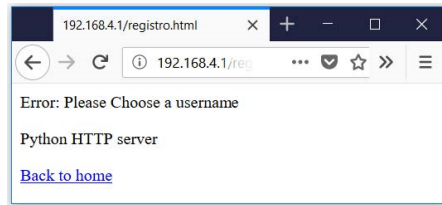
Figura 6.2: SSID AP.

6. RESULTADOS

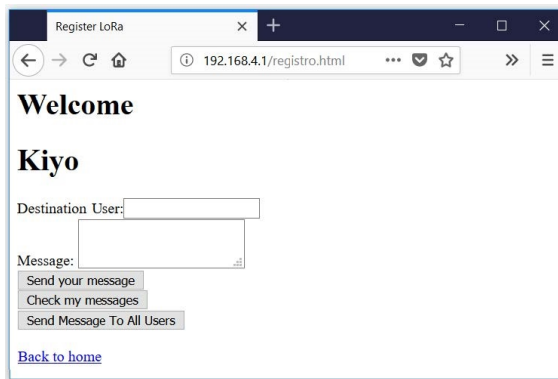
Una vez que los dispositivos finales se enlazan al *hub* a través de una conexión WiFi, los usuarios pueden acceder a una página web simple, que hemos diseñado como interfaz de la aplicación de mensajería (ver Figura 6.3a). Todos los usuarios tienen que “registrarse” antes de intercambiar cualquier mensaje. En caso de no registrarse, se muestra una página de error (ver Figura 6.3b). El registro es necesario para que el sistema pueda localizar los puntos finales, que pueden desplazarse a diferentes lugares en el tiempo. Una vez el usuario se registra, podrá ver la página de bienvenida (ver Figura 6.3c).



(a) Página de registro



(b) Página de error



(c) Página de bienvenida

Figura 6.3: Interfaz web de la aplicación de mensajería.

La página web tiene una interfaz bastante sencilla, para poder aprovechar las prestaciones de *hardware* de la LoPy4 y utilizarlas en el proceso de transmisión de mensajes.

Los usuarios pueden enviar mensajes a un usuario específico o enviar mensajes “*broadcast*” que pueden ser leídos por todos los usuarios que estén registrados en los *hub* vecinos. Si desea enviar un mensaje a otro usuario, tiene que escribir el nombre del usuario destino, el mensaje y después presionar el botón “*Send your message*” (ver Figura 6.4). En caso de que quiera enviar un mensaje *broadcast*, solo deberá escribir el mensaje y presionar el botón “*Send Message To All Users*” y el sistema se encarga de estructurar el mensaje con un identificador de usuario *broadcast*.

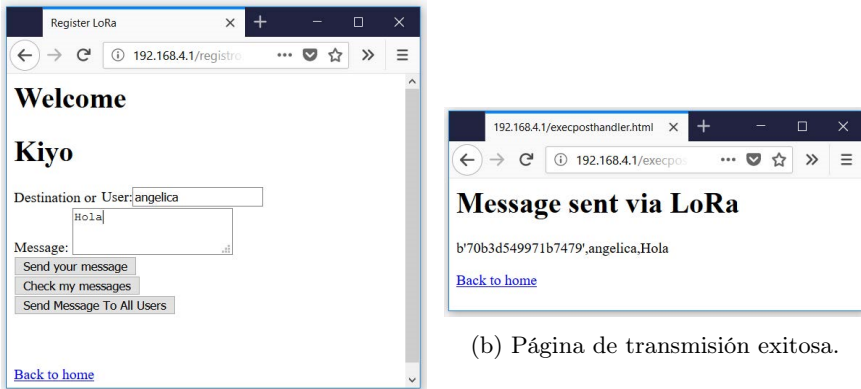


Figura 6.4: Páginas de envío de mensaje.

En caso de que el usuario quiera consultar si ha recibido algún mensaje, debe presionar el botón “*Check my messages*”, a continuación podrá observar una pantalla con dos secciones, en la primera sección puede observar los mensajes que son únicamente para este usuario y en la segunda sección se muestran los mensajes *broadcast* que pueden ser leídos por todos los usuarios que se registren en la red (ver Figura 6.5a).

Si el mensaje no puede ser entregado, por pérdida de paquetes o desconexión, el sistema mostrará una página de aviso y permite al usuario intentar enviar el mensaje nuevamente presionando el botón “*Resend*” (ver Figura 6.5b).

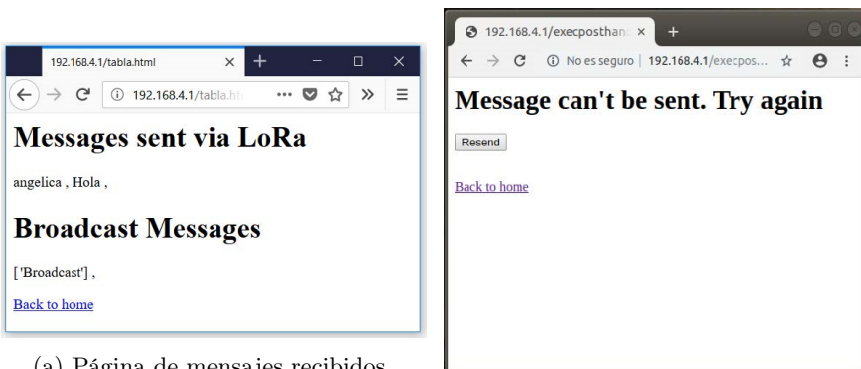


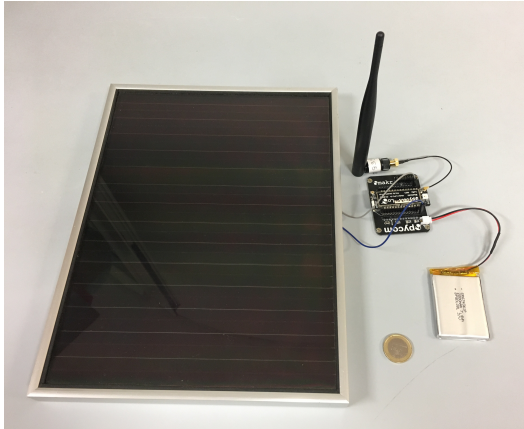
Figura 6.5: Páginas de mensajes recibidos y reenvío.

6. RESULTADOS

Como hemos mencionado, LoRaCTP también se encarga de administrar los usuarios y mensajes mediante una combinación de listas de diccionarios. Los usuarios se almacenan en una lista llamada “BaseU” donde se agregan, si es la primera vez que se registran en el *hub* o se consultan si se han registrado anteriormente, los mensajes se guardan en una lista llamada “BaseM” donde por cada usuario destino, se agrega una entrada con el usuario emisor, el usuario destino y el mensaje.

Debido a las limitaciones de hardware, los dispositivos no cuentan con la capacidad suficiente para el manejo de bases de datos. Los usuarios y los mensajes son almacenados en la LoPy4 en una tarjeta MicroSD, lo que permite que los usuarios puedan recibir mensajes sin saturar la memoria del dispositivo. Una vez que el usuario consulta sus mensajes en el dispositivo destino, los mensajes son eliminados para liberar espacio en la tarjeta.

Diseñamos algunos prototipos de los *hub* LoRa, con los cuales hemos realizado algunos experimentos, donde conseguimos resultados satisfactorios que presentaremos más adelante (ver Figura 6.6). En la Figura 6.6a podemos ver el *hub* con una antena omnidireccional LoRa, una batería externa, un panel solar de 5 W que se utiliza para cargar la batería y extender el tiempo de uso del sistema. La Figura 6.6b nos muestra un *hub* portable con su antena Lora, alimentado por una batería externa. Hemos realizado diferentes experimentos con estos prototipos donde pusimos a prueba la capacidad de transmisión variando la distancia, el tamaño de los mensajes y el *SF*.



(a) Prototipo LoPy4, batería y panel solar.



(b) Prototipo LoPy4 y batería externa.

Figura 6.6: Prototipos de hub LoRa diseñados.

Después de realizar el proceso de descubrimiento, LoRaCTP en el *hub* emisor se encarga de estructurar un objeto *JSON* que contendrá el nombre del usuario origen, el usuario destino y el mensaje. LoRaCTP es capaz de transmitir mensajes de hasta 100 KB. Dependiendo del tamaño del *JSON*, el protocolo segmenta el mensaje en paquetes con la estructura que hemos descrito anteriormente e inicia el proceso de transmisión hasta enviar todos los paquetes necesarios. En el *hub* receptor, el protocolo recibe los paquetes, devuelve un mensaje ACK para confirmar que este ha llegado, este proceso se repite hasta recibir el paquete con identificador de último paquete.

La Figura 6.7 muestra el conjunto ordenado de resultados del proceso de descubrimiento de usuario. El valor medio de todo el proceso es de 2,077 ms. Una vez conocida la ubicación del usuario, este proceso no se repite. Determinamos un parámetro llamado *Successful Transfer Time (STT)*, que mide el tiempo de transferencia de un mensaje desde el punto de vista del emisor. Se calcula desde el momento en que se envía el primer fragmento del mensaje hasta el momento en que se recibe el último ACK del último fragmento del mensaje.

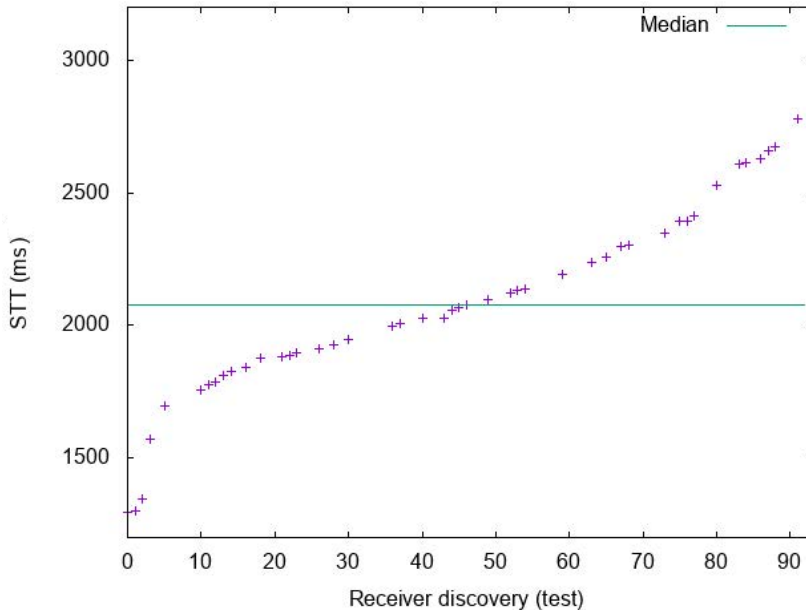


Figura 6.7: Proceso de descubrimiento del usuario; conjunto ordenado de resultados.

6.2 Integración con Telegram

Para integrar mejor nuestra arquitectura con las aplicaciones estándar de Internet, diseñamos un *hub gateway* para enlazarla con Telegram², una aplicación muy utilizada (ver Figura 6.8). Elegimos Telegram porque ofrece los llamados *Bots*, es decir, aplicaciones de terceros que se ejecutan dentro de Telegram. Los usuarios pueden interactuar con los *Bots* enviándoles mensajes, comandos y peticiones en línea.

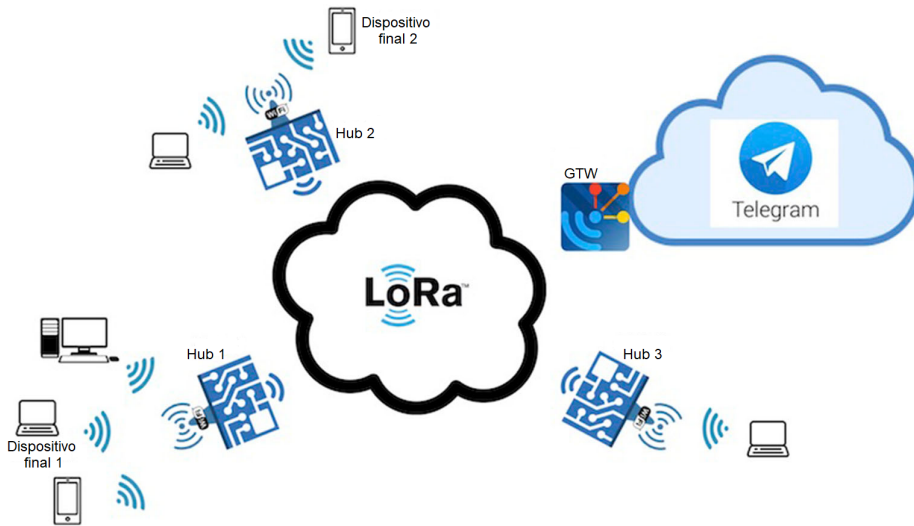


Figura 6.8: Arquitectura con integración de Telegram.

La creación del *Bot*, se hizo empleando la herramienta llamada BotFather, un complemento de Telegram que permite generar *Bots* respondiendo un conjunto de preguntas sencillas y que proporciona un token único que identifica al *Bot*.

El usuario destino debe registrarse previamente en el *Bot* para poder recibir mensajes. El usuario origen debe proporcionar el nombre del usuario destino y el mensaje que desea enviar, como si fuera a enviar un mensaje a otro *hub* LoRa, en la interfaz web hemos agregado el botón “*Send Message Via Telegram*” que al ser presionado, el sistema realiza el proceso de descubrimiento, si el usuario no se ha registrado previamente al *Bot* se muestra el mensaje destino no encontrado, en caso contrario, se inicia con el proceso de transmisión para entregarlo al *hub gateway*.

El *hub gateway* recibe a través de LoRa los mensajes, y lo reenvía al teléfono del usuario por medio del enlace de Internet. El *hub gateway* de nuestro prototipo se

²<https://telegram.org/>

basa en una Raspberry Pi 1 B+ v1.2 con una CPU Broadcom BCM2835, ARMv6, 32 bits, 700 MHz, tiene 512 MB de RAM, provista de un adaptador LoRa y una conexión estable a Internet (ver Figura 6.9). Los procesos requeridos por el *Bot* se ejecutan en la Raspberry Pi.

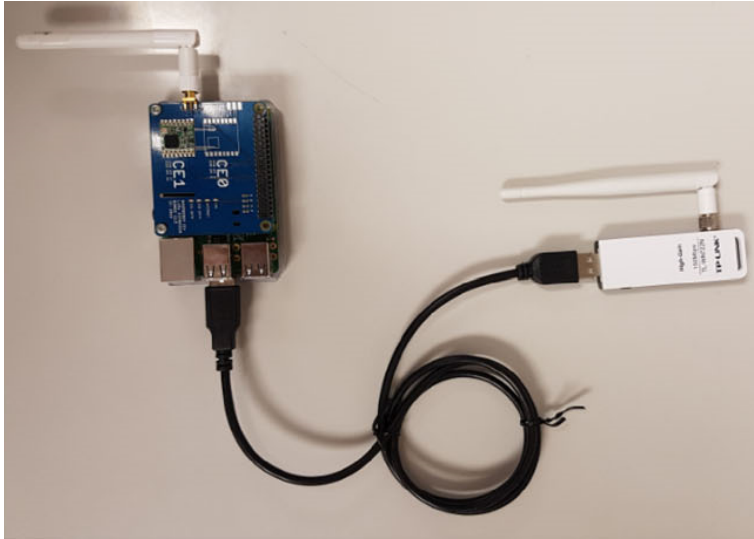


Figura 6.9: Prototipo hub gateway.

El modo de interacción entre el *hub gateway* y el *hub LoRa* es muy parecida a la forma en que se comunican los *hub LoRa* usando LoRaCTP, primero se realiza un intercambio de mensajes para confirmar si el usuario (en este caso de Telegram) está registrado, y luego se envía el mensaje (ver Figura 6.10).

Dado que el dispositivo utilizado para el prototipo *hub gateway* tiene mayores prestaciones de *hardware*, empleamos herramientas de *software* adicionales, como la inclusión de una pequeña base de datos en *MySQL* para registrar los nombres de usuario junto con el ID de Telegram de cada dispositivo, para permitir consultas más eficientes.

Por último, para facilitar la gestión, se añadieron comandos adicionales al *Bot* para permitir una mejor interacción:

- */start*: Inicia la interacción con el *Bot*.
- */register*: Permite el registro o la actualización de los nombres de usuario o números de teléfono.
- */help*: Proporciona ayuda sobre los comandos disponibles y su uso.

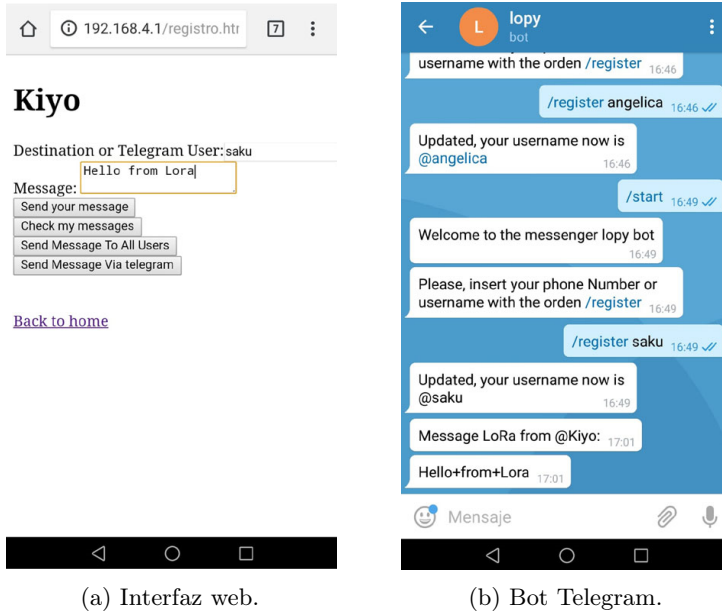


Figura 6.10: Interacción entre hub LoRa y hub gateway.

6.3 Mensajes de voz

Teniendo en cuenta los datos de las diferentes organizaciones respecto a los altos índices de analfabetismo en el mundo y especialmente en las zonas rurales, desarrollamos un prototipo que es capaz de realizar capturas de mensajes de voz y de esta forma nuestra solución permita a las personas que no saben leer ni escribir comunicarse a través de nuestro sistema de mensajería.

Los mensajes de voz se usan también en sistemas de mensajería como WhatsApp³ y Telegram, esta propuesta ha sido una forma de incluir a las personas analfabetas.

Agregamos a nuestra arquitectura original, el dispositivo que se encargará de grabar los mensajes de voz y procesarlos, para después transferirlos al *hub LoRa*, donde empleando LoRaCTP encontrará al usuario destino y envía el mensaje, cuando el mensaje sea recibido por completo se transfiere nuevamente a un dispositivo capaz de reproducir el mensaje de voz (ver Figura 6.11).

Diseñamos una interfaz fácil de usar y eficaz, para el entorno donde se va a utilizar. El prototipo se basa en una Raspberry Pi 3 B+ que cuenta con una CPU Broadcom BCM2837B0, Cortex-A53 (ARMLv8) SoC de 64 bits a 1.4 GHz, tiene 1 GB SDRAM e IEEE 802.11.b/g/n/ac LAN inalámbrico, Bluetooth 4.2, conectividad BLE y una pantalla táctil LCD de 3,5 pulgadas.

³<https://www.whatsapp.com/>



Figura 6.11: Arquitectura con mensajes de voz.

Este prototipo cuenta con una interfaz gráfica bastante intuitiva que muestra una serie de iconos que se activarán cuando sea necesario, para guiar al usuario paso a paso (ver Figura 6.12). Mediante esta interfaz, el usuario se identifica y genera el mensaje de voz que desea enviar. El dispositivo se encarga de procesar el mensaje para que pueda ser enviado por el protocolo.



Figura 6.12: Interfaz gráfica para mensajes de voz.

6. RESULTADOS

Como primer paso, el usuario origen debe identificarse utilizando los botones “arriba” y “abajo”, deberá elegir su fotografía de una serie de fotos de usuarios registrados previamente y presionar sobre la que lo identifique (ver Figura 6.13a), esto habilitará el área para identificar al usuario destino (ver Figura 6.13b), donde debe repetir los pasos anteriores hasta encontrar la foto del usuario a quien desea enviar el mensaje de voz. La interfaz también cuenta con un botón “Reset” que está activo durante todo el proceso de grabación del mensaje, el usuario podrá presionarlo si desea reiniciar el proceso y borrar todos los datos que ha proporcionado hasta ese momento.

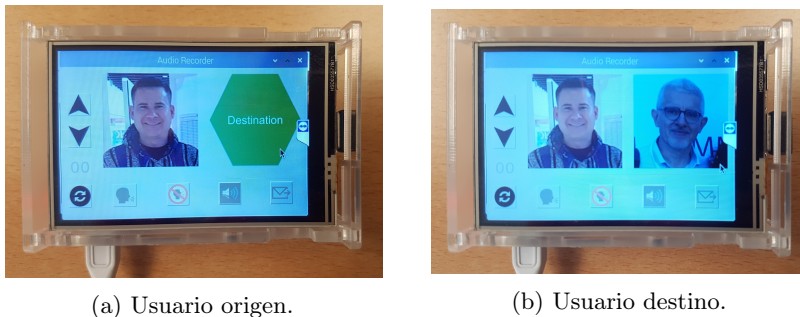


Figura 6.13: Identificando usuarios.

Después de identificar a los usuarios los botones “arriba” y “abajo” se deshabilitan y se activa el botón para iniciar la grabación del mensaje de voz (ver Figura 6.14a), estos mensajes pueden tener una duración máxima de 50 segundos, este valor se ha fijado en función del equilibrio entre la información que se puede enviar y el tiempo total necesario para hacerlo, si el usuario alcanza el límite de la duración el sistema detiene el proceso de grabación automáticamente, el usuario también puede detener este proceso en el momento que él así lo desee (ver Figura 6.14b), cuando inicia la grabación del mensaje también se muestra la duración de este.



Figura 6.14: Proceso de grabación del mensaje de voz.

Cuando el proceso de grabación ha concluido, se activará un botón que permite escuchar el clip de voz antes de enviarlo (ver Figura 6.15a), también se activa el botón para enviar el mensaje (ver Figura 6.15b).

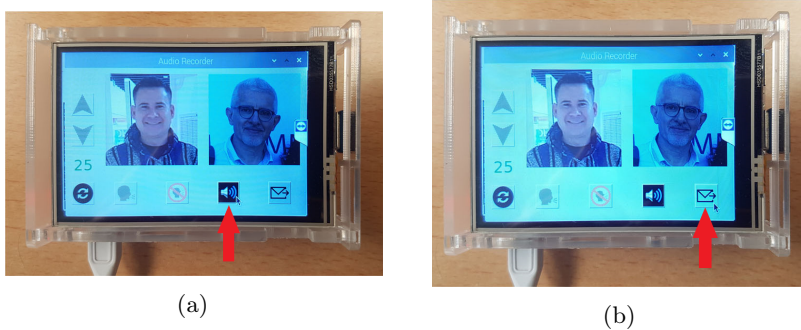


Figura 6.15: Botón para reproducir el clip de voz (izquierda) y botón para enviar mensaje (derecha).

En este punto, el usuario puede decidir enviar el mensaje de voz o reiniciar el proceso en caso de querer enviar un mensaje distinto o enviarlo a otro usuario.

El clip de voz se almacena en la memoria microSD de la Raspberry como un archivo *Waveform Audio File Format (WAV)*; un mensaje de 50 segundos suele tener un tamaño de 1 MB. Como se ha mencionado, el protocolo LoRaCTP puede transmitir mensajes de hasta 100 KB, por este motivo el archivo debe pasar por un proceso de compresión. Para comprimir el archivo WAV, utilizamos el *framework FFmpeg*⁴ para generar un archivo *Moving Picture Experts Group Audio Layer 3 (MP3)*. La ventaja de los archivos MP3 es que no pierden calidad de audio y su tamaño se reduce hasta en un factor de diez. De esta forma, un archivo WAV con 50 segundos de audio que tiene un tamaño de 1 MB, se convierte en un archivo MP3 con un tamaño de 100 KB y así puede transmitirse utilizando LoRaCTP. En la Tabla 6.1 podemos ver el tamaño de los archivos WAV generados y el tamaño de los archivos MP3 una vez comprimidos.

La Figura 6.16 permite ver la evolución del tiempo de compresión en función del tiempo de grabación del clip de audio. Como podemos ver, el tiempo de compresión crece claramente a medida que aumenta el tiempo de grabación. Las pruebas se han realizado con clips de audio de 10, 20, 30, 40 y 50 segundos. Cabe señalar que el tiempo de compresión, incluso en los clips de audio de 50 segundos, no es significativo con respecto al tiempo de transmisión de los mensajes mediante LoRaCTP.

⁴<https://ffmpeg.org/>

6. RESULTADOS

Tabla 6.1: Tamaño de los archivos WAV y MP3

Tiempo de grabación (s)	Tamaño del archivo WAV	Tamaño del archivo MP3
10	220 KB	24 KB
20	436 KB	40 KB
30	652 KB	60 KB
40	868 KB	80 KB
50	1 MB	100 KB

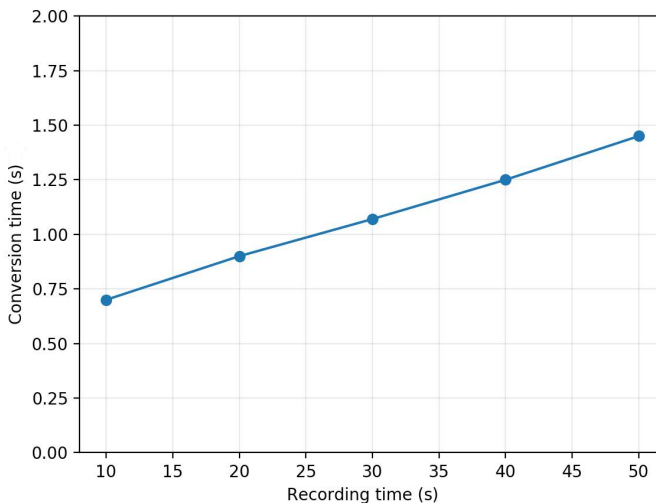


Figura 6.16: Tiempo de compresión (valores medios).

Los datos transferidos se estructuran como un objeto *JSON* con la forma indicada en la Lista 6.3.1.

La compresión del audio para el análisis de datos es necesaria para enviarlo a través de una conexión de bajo ancho de banda como LoRa. El objetivo es extraer las principales características del audio original, para utilizarlas en el desarrollo de aplicaciones inteligentes basadas en métodos de aprendizaje automático. Para ello, es necesario entrenar un modelo capaz de caracterizar, comprimir y reconstruir una grabación de audio. Con el método descrito anteriormente, el archivo de audio sufre una pérdida de rendimiento en los modelos de aprendizaje automático [112].

Por lo tanto, desarrollamos un auto-codificador ligero de *deep learning* que comprime el flujo de sonido a partir del audio WAV original para obtener una “pérdida de datos” limitada.

```

1 {
2   'DEV_ID': 'voice_recorder',
3   'QOS': 0,
4   'TOPIC': 'rasp1095/voicemail',
5   'VALUE': {'sender': 'pietro',
6             'receiver': 'kiyo'
7             'message': <mp3 file>}
8 }

```

Lista 6.3.1: Estructura de los mensajes intercambiados.

Un auto-codificador [113] es un tipo de red neuronal multicapa que se caracteriza por una determinada estructura formada por capas que disminuyen su dimensionalidad desde la capa de entrada hasta la capa central, y luego aumenta simétricamente el número de neuronas por capa de entrada hasta terminar en una capa de salida del mismo tamaño que la inicial (ver Figura 6.17).

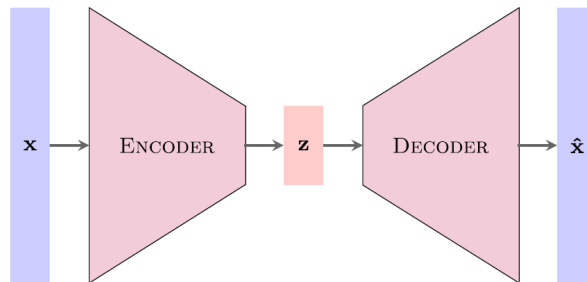


Figura 6.17: Esquema del auto-codificador de Hinton y Salakhutdinov utilizado.

La etapa de entrenamiento de la red neuronal se ha realizado en un servidor *High Performance Computing (HPC)* con dos *Graphics Processing Units (GPUs)* GeForce RTX 2080 utilizando la librería PyTorch⁵. Los parámetros principales del procedimiento de entrenamiento fueron 1280 épocas, el optimizador Adam y una tasa de aprendizaje de $1e^4$. El modelo tiene un tamaño total de 256076000 parámetros entrenados. Los datos utilizados para el entrenamiento se han tomado del conjunto de datos CSS10 [114], que es una colección de conjuntos de datos de un solo hablante en 10 idiomas. Cada uno de ellos consiste en archivos de audio grabados por un solo voluntario y su texto alineado procedente de LibriVox⁶. Todas las 6515 muestras de audio extraídas para el entrenamiento están compuestas

⁵<https://pytorch.org/>

⁶<https://librivox.org/>

6. RESULTADOS

por audios de 10 segundos, que se han dividido en submuestras que se ajustan al tamaño de la capa de entrada de la red.

Cada uno de los audios de entrada ha sido pre-procesado utilizando la transformada discreta de Fourier de una secuencia real, para convertir los datos de muestra en tiempo a frecuencia. La transformación en el dominio de la frecuencia muestra cómo varía la amplitud con la frecuencia. Esto suele denominarse espectro de frecuencias de la señal. Una onda sinusoidal de una sola frecuencia en el dominio del tiempo dará una sola línea a esa frecuencia en el dominio de la frecuencia. Un pico en el dominio del tiempo dará una dispersión de frecuencia en el dominio de la frecuencia. Y el proceso inverso se ha realizado para la reconstrucción del archivo de audio después de que haya sido procesado por el codificador.

Una vez que el modelo ha sido entrenado, se compila en las plataformas basadas en Raspberry para que funcione en estas arquitecturas *Acorn RISC Machines (ARM)*. Después de la etapa de entrenamiento, el modelo es serializado y desplegado simultáneamente en el emisor y receptor (ver Figura 6.18). De este modo, el emisor realizará la compresión desde la capa de entrada hasta la capa más pequeña para comprimir la información. El receptor ejecutará la parte del modelo desde la capa de normalización de la capa intermedia hasta la salida de la red.

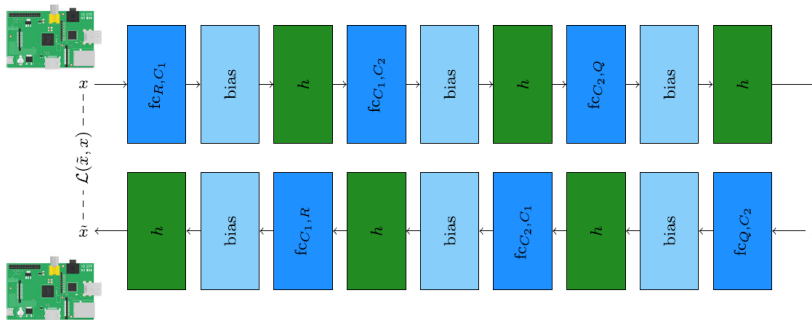


Figura 6.18: Arquitectura del compresor de sonido autoencoder.

La red neuronal artificial diseñada era de 980 MB en total. Está dividida en dos redes neuronales diferentes que se desplegaron en el emisor y receptor respectivamente. La primera red neuronal codifica el mensaje y, por tanto, se despliega en el emisor. La segunda decodifica el mensaje y se despliega en el receptor. Esto significa que cada plataforma requiere menos de 500 MB para ejecutar el modelo de compresión que es lo suficientemente pequeño como para ser ejecutado en una Raspberry. Nuestro método proporciona hasta el 40% de tasa de compresión del archivo WAV original y esta tasa es constante a medida que aumenta el tamaño del archivo de entrada.

La Figura 6.19 muestra la forma de onda normalizada del registro de audio original y la obtenida tras la descompresión con nuestro enfoque. Es importante señalar que el audio regenerado coincide casi al 95% con el registro original. Además, este procedimiento permite el aprendizaje automático, por lo que se podrían desarrollar otros procedimientos adicionales de minería de datos.

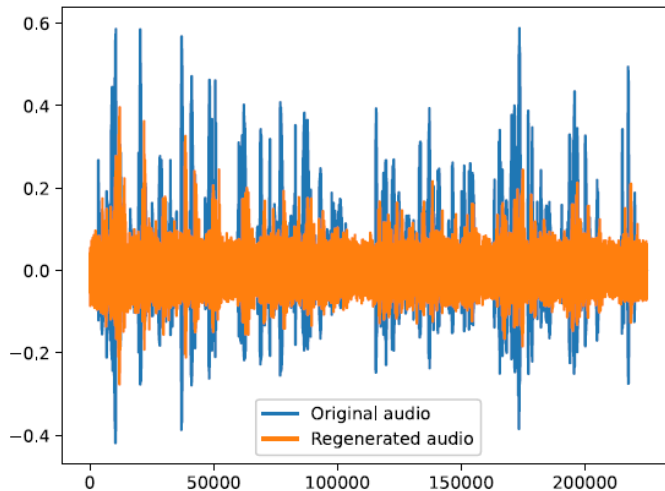


Figura 6.19: Forma de onda normalizada del original y de la predicción.

Por último, los procedimientos de compresión y descompresión con nuestro enfoque tienen un tiempo de ejecución similar, y se escalan linealmente junto con el tamaño del registro de audio. Por ejemplo, para un audio de 29 segundos, con una frecuencia de muestreo de 11025 y un muestreo final de 320828 valores, la fase del codificador tarda 31,21 segundos y el decodificador necesita 40 segundos.

6.4 Recolección de datos de fuentes externas y proxy MQTT

Para ampliar la funcionalidad del sistema de mensajería, integramos un proceso que nos permite recopilar datos de sensores remotos, transmitir estos datos a largas distancias utilizando LoRa e integrarlos automáticamente en un sistema de publicación/suscripción basado en *Message Queuing Telemetry Transport (MQTT)*.

La recopilación de datos de sensores está integrada dentro de nuestra plataforma a través de un servicio dedicado que se adjunta al sistema como un cliente específico.

6. RESULTADOS

La idea general es que los sensores recopilarán los datos, los empaquetarán como una pieza estructurada de información y los enviarán a este servicio como un mensaje. El dispositivo receptor, donde se está ejecutando el servicio, (1) descomprimirá el mensaje, (2) creará un mensaje MQTT adecuado y (3) lo publicará en el *broker* que se esté utilizando. La configuración requerida es la que se muestra en la Figura 6.20. Los sensores se conectan al *hub*, mediante el enlace WiFi, como cualquier otro cliente. El dispositivo que ejecutará el servicio *proxy* MQTT también debe estar conectado a un *hub* mediante WiFi y debe tener una conexión clara con el *broker* utilizado, ya sea a través de Internet o mediante un enlace directo *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

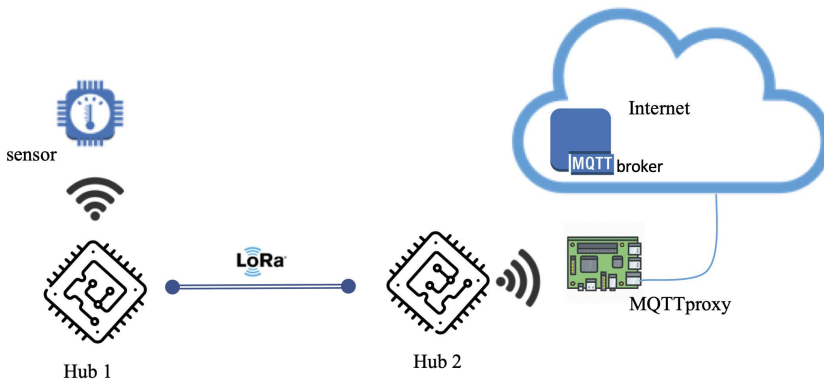


Figura 6.20: Configuración necesaria para los sensores y proxy MQTT.

Los sensores están integrados en esta plataforma mediante la interfaz *REST*. La secuencia es básicamente la misma que utilizan los clientes habituales: hay una primera fase de registro seguida de una fase de “*Push*”. La Figura 6.21 describe gráficamente esta operación.

Los datos transferidos se almacenan en un objeto *JSON* con la estructura indicada en la Lista 6.4.1.

```
1 {  
2     'DEV_ID': 'lopy_374534',  
3     'QOS': 0,  
4     'TOPIC': 'sensor1/luminosity',  
5     'VALUE': 25}  
6 }
```

Lista 6.4.1: Estructura JSON del mensaje enviado.

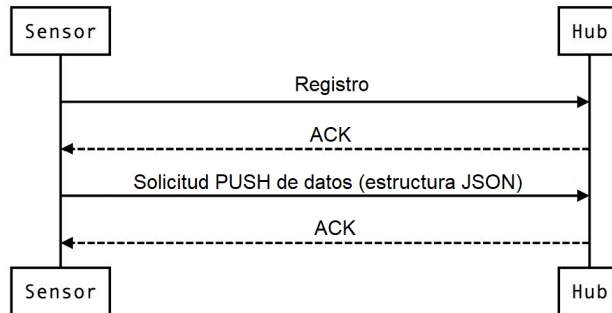


Figura 6.21: Flujo de datos entre un sensor y un hub.

Los datos contenidos en el objeto *JSON* pueden tener un tamaño variable limitado solo por la velocidad de datos que podemos obtener del canal LoRa.

La provisión del *proxy* MQTT se basa en un enfoque “*anycasting*”. Esto significa que puede haber varios *proxys* MQTT disponibles en el área cubierta por cualquier *hub*. En cuanto a los clientes habituales, el *hub* que recibió el mensaje *JSON* iniciará la búsqueda de un *proxy* MQTT como si fueran usuarios finales habituales; si se reciben varias repeticiones, se selecciona la primera. La FIGURA describe gráficamente esta operación. Se podrían adoptar otras estrategias basadas, por ejemplo, en la carga detectada de un determinado dispositivo *proxy* MQTT. El *hub* empaquetará el mensaje *JSON* que se le envíe al *proxy* MQTT seleccionado, utilizando el procedimiento estándar empleado por el sistema de mensajería. El *proxy* MQTT, utilizando la interfaz *REST*, interrogará periódicamente al *hub* con el que está conectado para obtener los datos. La Figura 6.22 describe gráficamente esta operación.

Una vez obtenido el mensaje, *proxy* MQTT extraerá el *JSON* y creará un mensaje de publicación adecuado para el *broker* conectado.

Para evaluar el desempeño de nuestra solución, realizamos algunos experimentos en escenarios reales utilizando nuestros prototipos, donde analizamos el tiempo de transferencia de los mensajes con diferentes parámetros, como por ejemplo, la distancia, el *Spreading Factor* (*SF*), el consumo energético.

Como mencionamos anteriormente, medimos el rendimiento del sistema mediante una métrica denominada *Successful Transfer Time* (*STT*). Mide el tiempo de transferencia de un mensaje desde el punto de vista del remitente, y se calcula desde el momento en que se envía el primer fragmento del mensaje hasta el momento en que se recibe el último Acknowledgement (ACK) del último fragmento del mensaje.

La Figura 6.23 muestra el comportamiento del *STT* al variar la frecuencia de los mensajes generados, para ser más precisos, al aumentar el número de mensajes simultáneos generados. Los resultados se obtienen utilizando un *SF* 7 y un tamaño

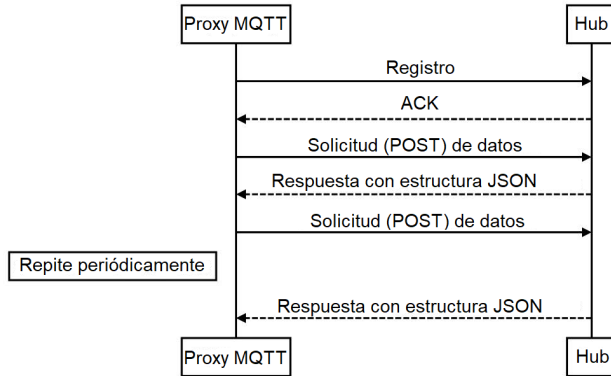


Figura 6.22: Proxy MQTT obteniendo datos del hub.

de mensaje de 256 bytes. Como puede observarse, el STT crece linealmente a medida que aumenta el número de mensajes simultáneos, dándose el caso extremo en el que 14 usuarios simultáneos intentaban enviar un mensaje; en algunos casos se obtuvo un retraso máximo de 80 segundos. El valor medio, para los 14 usuarios simultáneos, fue de 35,892 segundos. Con más de 14 usuarios a la vez, el sistema mostró un comportamiento muy poco fiable y lo consideramos básicamente inutilizable.

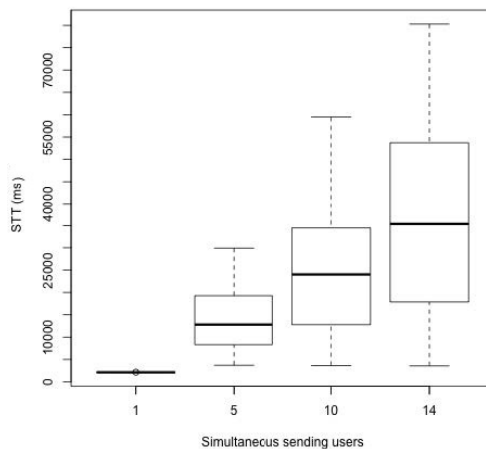


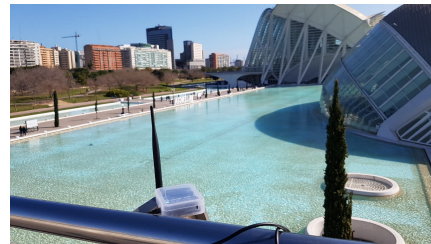
Figura 6.23: Evaluación de la escalabilidad del sistema aumentando el número de usuarios que envían simultáneamente.

6.4. Recolección de datos de fuentes externas y proxy MQTT

Las pruebas de comunicación se realizaron en distancias de 1, 100, 750 y 6000 metros. Los experimentos de 1 y 100 metros se hicieron en las instalaciones de la Universitat Politècnica de València, con la finalidad de obtener datos de comparación. Las pruebas de 750 metros se realizaron en Valencia, en la zona de la Ciutat de les Arts i les Ciències, véase la Figura 6.24, mientras que las de 6 km se realizaron entre dos miradores en Chiapas, México (ver Figura 6.25). Estos últimos lugares se encuentran en zonas lo suficientemente altas como para no tener obstáculos en medio; la Figura 6.26a muestra el escenario donde se realizaron las pruebas; hay una clara línea de visión entre los dos puntos que están a 6 km de distancia, como se muestra en la Figura 6.26b.

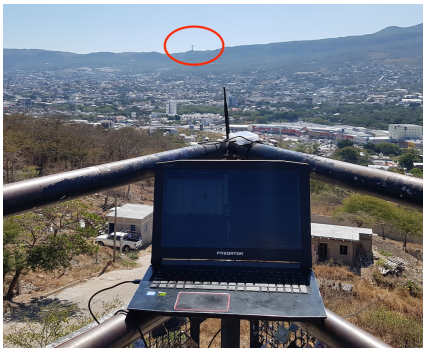


(a) Vista desde el hub origen.

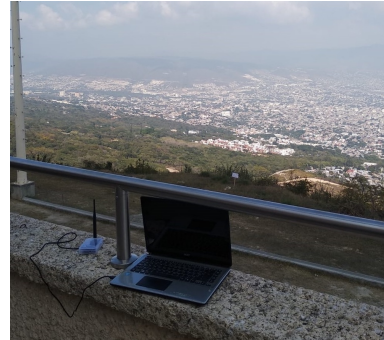


(b) Vista desde el hub destino.

Figura 6.24: Ubicación en Valencia para las pruebas de 750 m.



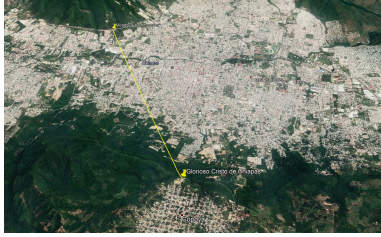
(a) Ubicación "Mirador los Amorosos".



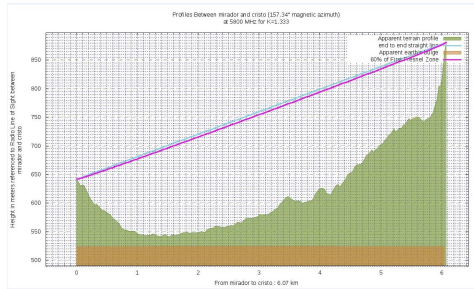
(b) Ubicación "Mirador del Glorioso Cristo de Chiapas".

Figura 6.25: Ubicación en Chiapas, México para las pruebas de 6 km.

6. RESULTADOS



(a) Vista del mapa de la trayectoria del enlace. (Imagen obtenida con Google Earth, ©2020 DigitalGlobe)



(b) Perfil del terreno entre las dos localizaciones.

Figura 6.26: Escenario de pruebas en Chiapas, México.

Para determinar la estabilidad del sistema, se enviaron ráfagas de 10 mensajes. El rendimiento del sistema fue estable y casi idéntico al de las pruebas a distancias más cortas. Las retransmisiones fueron escasas en los experimentos de largo alcance y tuvieron un impacto insignificante en el STT. Todas las pruebas se realizaron utilizando un SF 7 y SF 12 con mensajes de 1, 10, 50 y 100 KB.

La Tabla 6.2 nos muestra la cantidad de paquetes que se tienen que transmitir para enviar mensajes de diferentes tamaños usando SF 7 y SF 12, así como el tiempo de transmisión del mensaje.

Tabla 6.2: Segmentación de mensajes de diferentes tamaños y STT.

Tamaño del mensaje	Paquetes	STT SF 7	STT SF 12
Hola	1	1,7 s	6 s
1 KB	5	2,8 s	52,04 s
10 KB	48	23,5 s	8,25 min
50 KB	239	2,08 min	41,07 min
100 KB	486	4,29 min	82,43 min

La Figura 6.27 ofrece una clara ilustración de la evolución del STT como función directa del tamaño de los mensajes dado el SF 7 (es decir, el STT crece con el tamaño de los mensajes), mientras que la distancia entre los nodos tuvo un efecto insignificante. Como era de esperar, el rendimiento global ofrecido por LoRa es bastante bajo, del orden de 250 bps (ver la Figura 6.28).

Las Figuras 6.29 y 6.30 muestran los valores medios de STT de mensajes de distintos tamaños enviados en SF 7 y SF 12, respectivamente, medidos en función de las distancias entre dos nodos. En los resultados se observa un comportamiento casi constante, aunque el STT crece claramente a medida que aumenta el tamaño del mensaje. El sistema es bastante estable a una distancia creciente, y se requirieron muy pocas retransmisiones durante los experimentos.

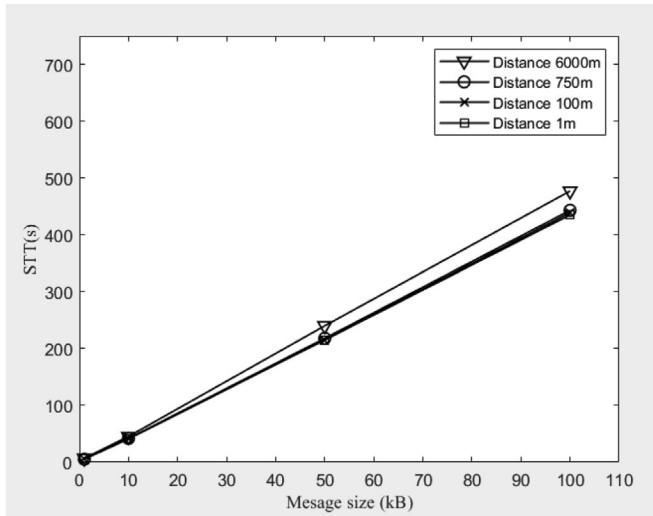


Figura 6.27: Comportamiento del STT con un tamaño de mensaje variable (valores medios) utilizando SF 7.

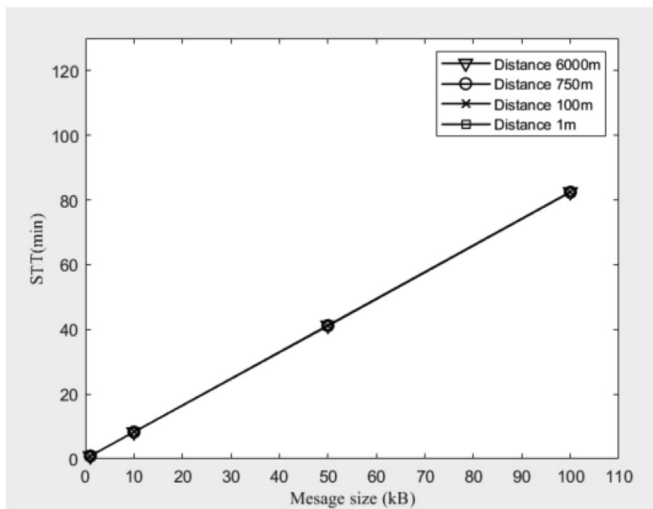


Figura 6.28: Comportamiento del STT con un tamaño de mensaje variable (valores medios) utilizando SF 12.

6. RESULTADOS

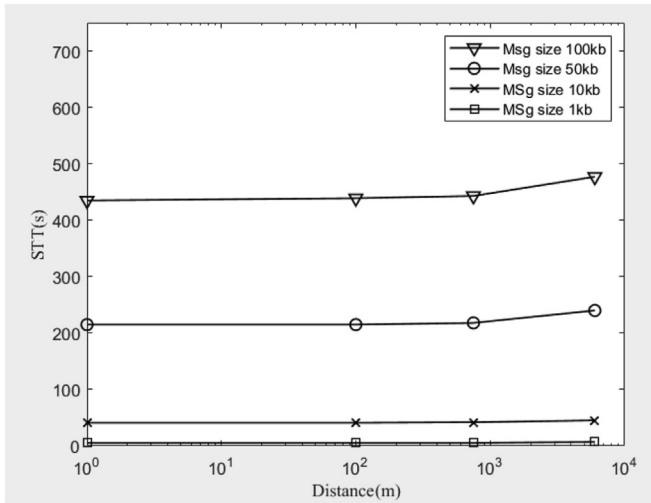


Figura 6.29: STT frente a la distancia entre dos nodos utilizando SF 7.

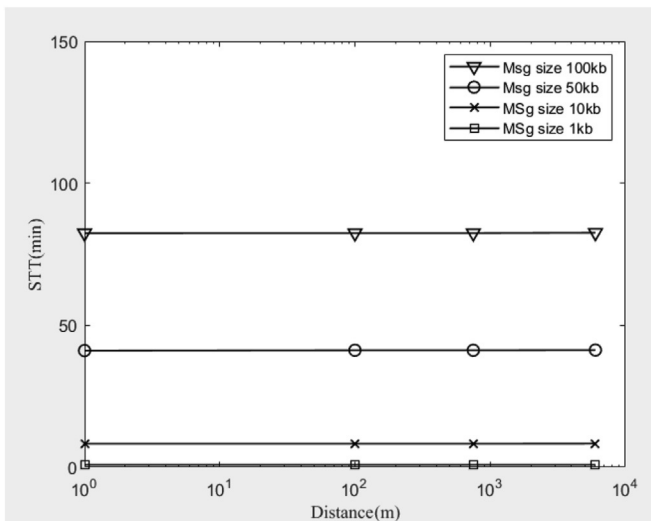


Figura 6.30: STT frente a la distancia entre dos nodos utilizando SF 12.

6.4. Recolección de datos de fuentes externas y proxy MQTT

A partir de estos resultados, podemos concluir que nuestra solución es una solución eficaz y estable para integrar datos desde largas distancias utilizando LoRa. Por ejemplo, los mensajes de 100 KB transmitidos bajo SF 7 obtuvieron un retraso máximo de 457,56 segundos y un mínimo de 451,49 segundos (es decir, unos 7,5 minutos). Bajo SF 12, los retrasos máximo y mínimo aumentaron a 82,545 y 82,513 minutos, respectivamente. Aunque está claro que el peor aspecto del uso de LoRa es el bajo rendimiento, ese aspecto se compensa con la distancia recorrida y el bajo coste energético que requieren estos dispositivos, proporcionando una solución frugal a un problema notable.

También hemos realizado evaluaciones preliminares para determinar el coste energético de la transferencia de datos. La Figura 6.31 muestra la cantidad de energía requerida por un dispositivo que no estaba realizando ninguna operación. Los picos se debieron al parpadeo de los LED's, y el uso medio de energía fue de aproximadamente $176 \mu\text{Wh}$.

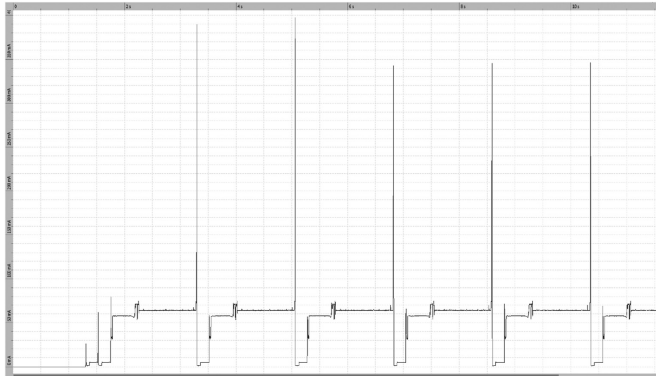


Figura 6.31: Representación de la energía consumida por el dispositivo cuando no se realizan operaciones.

La Figura 6.32 muestra la energía necesaria para enviar un solo paquete de mensajes, con el segmento entre las dos flechas rojas representando todas las fases involucradas en el establecimiento de la sesión, el envío del paquete y la recepción del ACK. El bloque requirió $287 \mu\text{Wh}$, lo que indica un gasto de $111 \mu\text{Wh}$ para enviar el paquete.

La Figura 6.33 muestra la energía que necesita el dispositivo para enviar cinco paquetes y el proceso correspondiente. Esta figura indica claramente el envío de cada paquete, el intervalo de espera para que llegue el ACK y la llegada del mensaje ACK (por ejemplo, la secuencia indicada entre las dos flechas rojas). Cada una de estas secuencias de bloques tiene un coste energético de $60 \mu\text{Wh}$.

6. RESULTADOS

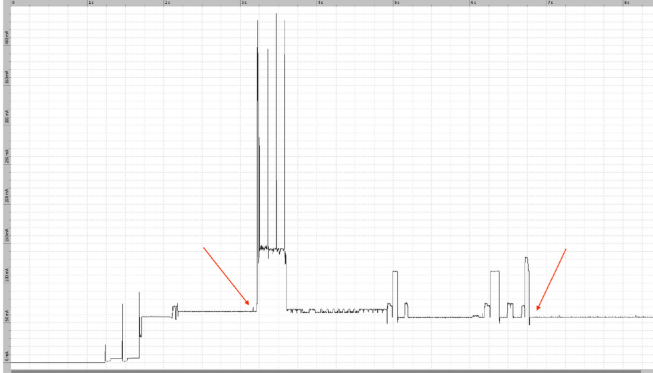


Figura 6.32: Representación de la energía consumida por el dispositivo cuando se envía un único paquete de mensajes.

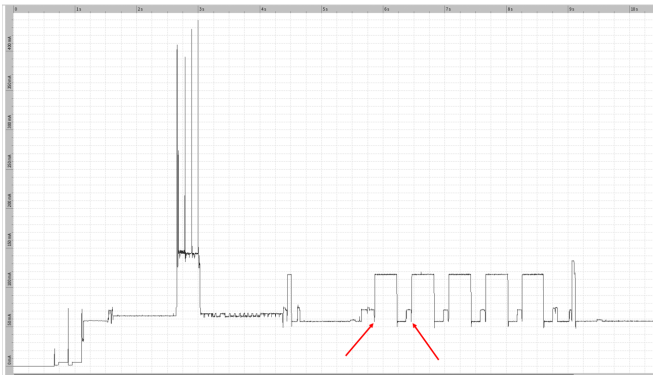


Figura 6.33: Representación de la energía consumida por el dispositivo cuando se envían cinco paquetes.

Por lo tanto, podemos idear una función para calcular el coste energético e_c de enviar un paquete. Con SF 7, un paquete de tamaño p en bytes da:

$$e_c[\mu Wh] = 287 + ([p/222] - 1) * 60$$

Como la LoPy4 se alimenta con 3,3V, el coste energético en amperios es:

$$e_c[\mu Ah] = e_c[\mu Wh]/3,3$$

6.5 Nodos FUDGE

La proliferación del IoT inteligente en asociación con la *Artificial Intelligence (AI)* plantea muchos desafíos que requieren enfoques novedosos y repensar toda la arquitectura, la comunicación y el procesamiento para cumplir con los requisitos de latencia, confiabilidad y uso de recursos [115]. A su vez, está dando lugar al desarrollo de nuevas aplicaciones que no solo requieren recursos informáticos adicionales, sino que agregan nuevas restricciones de privacidad y baja latencia. Esta tendencia ha motivado la reciente adopción en IoT del paradigma de *edge/fog computing*, que introduce un nivel intermedio de recursos entre la nube y los dispositivos de IoT [116, 117].

Edge computing también puede resultar beneficiosa para brindar soluciones avanzadas de IoT en áreas donde la conectividad es escasa y donde en general los recursos son limitados, como por ejemplo en áreas rurales y en países en desarrollo.

Proponemos una arquitectura frugal básica compuesta por un nodo central denominado “agregador” y varias estaciones edge (ver Figura 6.34).

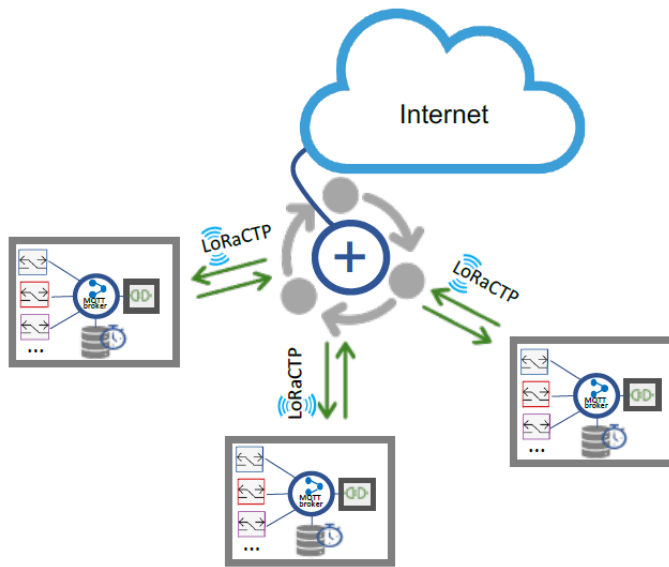


Figura 6.34: Arquitectura FUDGE.

6. RESULTADOS

Para probar la viabilidad de la arquitectura FUDGE, desarrollamos un prototipo que tiene como plataforma básica una Raspberry Pi 3 Modelo B+.

El *broker* MQTT utilizado fue *Eclipse Mosquitto*⁷, un *broker* de mensajes de código abierto con licencia *Eclipse Public License (EPL)/Eclipse Distribution License (EDL)*, que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. *Mosquitto* es liviano y adecuado para su uso en computadoras de baja potencia. El administrador de persistencia utilizó para el almacenamiento de datos el soporte de *InfluxDB*⁸, una base de datos de series de tiempo de código abierto.

Mosquitto e *InfluxDB* se ejecutan al inicio como contenedores Docker⁹. Luego, el administrador de persistencia comienza conectándose al servidor *InfluxDB* y suscribiéndose al tema genérico “rpi/red/#” (en este caso, el <device id> es rpi/red). Al recibir un mensaje del *broker*, el código ejecutado fue el que se muestra en la Lista 6.5.1 (eliminamos los detalles de bajo nivel).

```
1 def on_message(mqttc, userdata, msg):
2     top = msg.topic.split('/')
3     if (top[3]=='P'): # Checking if data must be persistent
4         jrecord = create_json_data(msg.topic, msg.payload)
5         clientIX.write_points(jrecord, database=IXDB,
6                               protocol='json')
7     elif (top[3]=='X') and (top[4]=='request'):
8         read_from_db_messapp(msg.topic, msg.payload)
9     else:
10        pass # nothing to do
```

Lista 6.5.1: Fragmento del código del gestor de persistencia para manejar los mensajes entrantes.

Básicamente, el *topic* del mensaje entrante se analiza en busca de la solicitud para que su contenido sea persistente. Si este es el caso, se crea un registro de base de datos adecuado a partir del *payload* del mensaje (línea 4) y, finalmente, se almacena en la base de datos (línea 5).

La Figura 6.35 muestra los primeros resultados de los experimentos de evaluación del desempeño que hicimos. La figura muestra el comportamiento del retraso, cuando un *proxy* de contenido genera un contenido aleatorio con una frecuencia predefinida. En verde se muestra el caso de un contenido de 100 bytes generado cada 5 segundos, mientras que en rojo es el caso de un contenido de 100 KB generado cada 100 ms. El tiempo se mide desde el momento en que se empaqueta el contenido para crear el mensaje que se enviará al *broker*, hasta que el administrador

⁷<https://mosquitto.org/>

⁸<https://www.influxdata.com/>

⁹<https://www.docker.com/>

de persistencia completó el almacenamiento del mensaje en la base de datos. El primer caso se puede considerar como una fuente de carga muy ligera y se utilizó como referencia; el valor medio fue de 29 ms con un percentil 75 de 30 ms. El segundo caso intentó estresar un poco el sistema, pero el rendimiento se mantuvo estable; el valor medio fue de 84 ms con un percentil 75 de 63 ms.

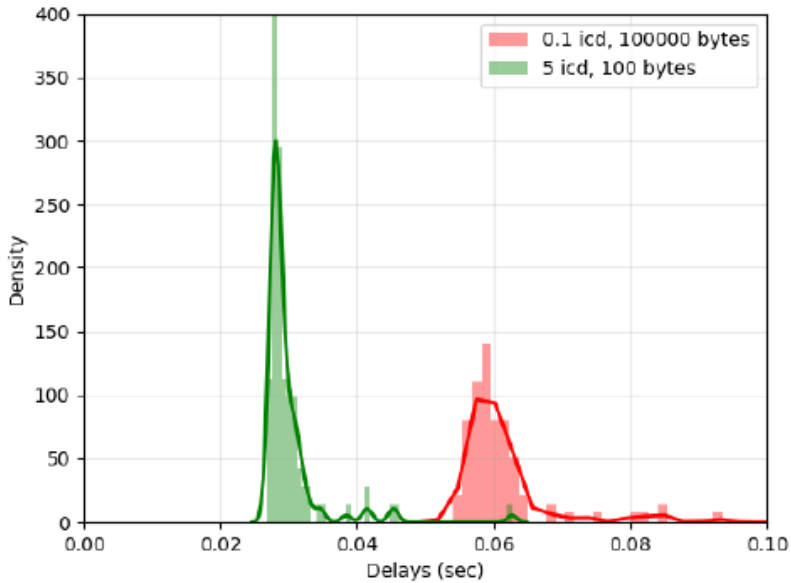


Figura 6.35: Resultados de los experimentos de evaluación del rendimiento.

En caso de que sea una solicitud de búsqueda en el repositorio de persistencia (“X”), se inicia una búsqueda en la base de datos. en la Lista 6.5.2 muestra, el caso del servicio de mensajería, ejecutar la función de llamada en la línea 8 de la Lista 6.5.1. Primero, la solicitud se transforma en una respuesta. Usando el *payload*, se prepara el *SELECT* adecuado para la base de datos y se envía al servidor de base de datos. Los valores se devuelven como una lista de valores, o si no se encontró ningún valor, se devuelve la etiqueta especial “NO DATA”

Finalmente, como se indicó anteriormente, las herramientas de análisis de datos se pueden utilizar para extraer métricas que tengan sentido a partir de los datos recopilados y las aplicaciones de monitoreo a través de paneles personalizables. La Figura 6.36 muestra un par de ejemplos de como Grafana¹⁰, ejecutado como

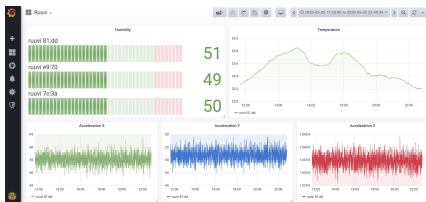
¹⁰<https://grafana.com/>

6. RESULTADOS

```
1 r = cIX.query("SELECT * FROM "+
2             IXDB+".autogen."+pload["measurement"]+
3             " where destination='"+
4             pload["tags"]["destination"]+"'")
5 if len(r) == 0:
6     mqttc.publish(topic, "NO DATA")
7 else:
8     points = r.get_points()
9     for point in points:
10        payload = {
11            "measurement": "messapp",
12            "tags": {"sender": point['sender'],
13                  "time": point['time']},
14            "fields": {"message": point['message']}
15        }
16        jpayload = json.dumps(payload)
17        mqttc.publish(topic, jpayload)
```

Lista 6.5.2: Fragmento del código de una búsqueda del servicio de mensajería.

contenedor Docker, se usa para monitorear datos de los dispositivos Ruuvi y del dispositivo que aloja el FUDGE.



(a) Datos de monitorización de los dispositivos Ruuvi.



(b) Monitorización de datos del sistema desde los dispositivos FUDGE.

Figura 6.36: Ejemplos de Grafana para una aplicación de monitorización.

Los *proxys* de contenido son los microservicios que se encargan de obtener los datos de los distintos sensores y publicarlos en el intermediario para su manejo. Básicamente, los datos se obtienen de una fuente física específica y se crea una estructura de datos *JSON* que incluye la medición específica, las etiquetas (por ejemplo, la dirección MAC del dispositivo), y como campos los valores del dispositivo.

En esta fase de desarrollo y prueba utilizamos, como fuentes de contenido: (1) algunos sensores ambientales Ruuvi¹¹, (2) los datos del sistema interno de la Raspberry Pi y (3) un sistema de mensajería.

Los proxys de contenido son los microservicios que se encargan de obtener los datos de los distintos sensores y publicarlos en el intermediario para su manejo. Hemos detallado la estructura del *topic* anteriormente. El código de la Lista 6.5.4 se extrae del de Ruuvi y muestra la estructura básica de estos servicios. Básicamente, los datos se obtienen de la fuente física específica (línea 4) y se crea una estructura de datos JSON (líneas 7-12) que incluye la medición específica (por ejemplo, ruuviz"), las etiquetas (por ejemplo, la dirección MAC del dispositivo Ruuvi específico), y como campos los valores del dispositivo específico, que se muestran en la Lista 6.5.3.

```
1  {
2      'data_format': 5,
3      'humidity': 48.17,
4      'temperature': 22.84,
5      'pressure': 1014.65,
6      'acceleration': 997.3645271414058,
7          'acceleration_x': -44,
8          'acceleration_y': 28,
9          'acceleration_z': 996,
10     'tx_power': 4, 'battery': 2907,
11     'movement_counter': 234,
12     'measurement_sequence_number': 14039,
13     'mac': 'f4af92d97c3a'
14 }
```

Lista 6.5.3: Datos proporcionados por un dispositivo Ruuvi.

¹¹<https://ruuvi.com/>

6. RESULTADOS

```
1 while True:
2     for i in range(len(sensors)):
3         # update state from the device
4         state = sensors[i].update()
5
6         devid = RTAGS[i]
7         payload = {
8             "measurement": "ruuvis",
9             "tags": {"devid": devid
10                },
11             "fields": state
12        }
13        jpayload = json.dumps(payload)
14        mqttc.publish("rpiored/ruuvis/L/P",
15                    payload=jpayload,
16                    qos=0, retain=False)
```

Lista 6.5.4: Fragmento del código de los proxys de contenido para el dispositivo de Ruuvi.

Capítulo 7

Conclusiones, Publicaciones y Trabajos Futuros

El principal objetivo de esta tesis ha sido diseñar una solución basada en Redes Comunitarias (RC), para proveer de un medio de comunicación digital a las comunidades rurales, mediante mensajes de texto y voz. Además de aprovechar la infraestructura para recolectar datos de sensores y transmitirla a largas distancias, para después compartirla con otros usuarios a través de tecnologías pub/sub MQTT.

Las aportaciones claves de este trabajo son las siguientes:

- Hemos presentado una solución de bajo coste y bajo consumo para proporcionar mensajería a comunidades aisladas. Creemos que este tipo de soluciones son necesarias, ya que todavía hay un porcentaje importante de la población mundial fuera del alcance de los proveedores de servicios de telecomunicaciones comerciales, especialmente en las zonas rurales de los países en desarrollo. Una vez instalado, el *hub* LoRa también puede utilizarse para transmitir datos de sensores y otros tipos de datos de interés, así como un sistema de alerta temprana para la mitigación de desastres. Nuestra plataforma integra un *gateway* Telegram, una aplicación de mensajería ampliamente utilizada. Esto permite ampliar el alcance del sistema de mensajería a Internet.
- Desarrollamos un protocolo flexible basado en la tecnología LoRa, que permite la transferencia de “contenido” a grandes distancias con muy baja energía. LoRaCTP proporciona todos los mecanismos necesarios para que LoRa sea fiable, introduciendo una conexión ligera, lo que permite el envío de un mensaje de datos idealmente tan largo como sea necesario. Diseñamos este

7. CONCLUSIONES, PUBLICACIONES Y TRABAJOS FUTUROS

protocolo como soporte de comunicación para soluciones IoT basadas en *edge* de datos pequeños, dada su estabilidad, bajo consumo de energía y capacidad de distancia.

- Además, brinda la posibilidad de incluir clips de audio que permite a los usuarios que no saben leer ni escribir, enviar mensajes basados en la voz. La inclusión de personas analfabetas, en este tipo de sistemas, sigue siendo un objetivo crítico, y la simplificación de las interacciones con el usuario es muy demandada. Se integró también un servicio de compresión de audio para extraer las principales características del audio original, para ser utilizado para el desarrollo de análisis de audio inteligente basado en *Machine Learning (ML)*. Este tipo de soluciones están siendo usadas frecuentemente, permitiendo el diseño de aplicaciones para analizar la contaminación acústica o la previsión de niveles de sonido.
- También presentamos una integración que permite recolectar datos de sensores remotos, transmitir estos datos usando LoRa para cubrir largas distancias e integrarlos automáticamente en un sistema *pub/sub* basado en MQTT. El objetivo es brindar soporte para sistemas IoT en áreas rurales conectando cualquier tipo de sensor, como sensores ambientales para pronóstico del tiempo, sensor químico, sensores de nivel, sensores de imágenes, etc. Este sistema está integrado en un sistema de mensajería basado en LoRa para conformar una nueva plataforma *Frugal IoT*.
- Detallamos el diseño de una arquitectura genérica frugal *edge/fog* para proporcionar servicios informáticos en escenarios con recursos limitados y con poca conexión. Esbozamos el concepto de “agregador”, el proceso que coordina el flujo de datos entre los servicios en los nodos *edge* con los de la nube. Detallamos la estructura de los nodos *edge*, denominados “FUDGE” (nodo FrUgal eDGE).
- Se describieron algunos detalles de los aspectos de implementación junto con algunos resultados preliminares. Las pruebas iniciales de la arquitectura indicaron que es lo suficientemente flexible y robusta, como para proporcionar una buena plataforma para el despliegue de servicios *edge* en contextos remotos o rurales.

Publicaciones

Revistas Internacionales

- A LoRa-based protocol for connecting IoT edge computing nodes to provide small-data-based services. Kiyoshy Nakamura, Pietro Manzoni, Alessandro Redondi, Edoardo Longo, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate. En *Digital Communications and Networks*. **I.F. 2020: 6.797; JCR: Q1 Category.**
- LADEA: A Software Infrastructure for Audio Delivery and Analytics. Kiyoshy Nakamura, Daniel Hernández, José M. Cecilia, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate. En *Mobile Networks and Applications*, Vol. 26, 2021. (pp. 2048-2054). **I.F. 2020: 3.426; JCR: Q2 Category.**
- A Low-cost and Low-Power Messaging System Based on the LoRa Wireless Technology. Angelica Monero Cardenas, Miguel Kiyoshy Nakamura Pinto, Ermanno Pietrosemoli, Marco Zennaro, Marco Rainone, Pietro Manzoni. En *Mobile Networks and Applications*, Vol. 25, 2020. (pp. 961-968). **I.F. 2020: 3.426; JCR: Q2 Category.**

Congresos Internacionales

- LoRaCTP: a LoRa based Content Transfer Protocol for sustainable edge computing. Kiyoshy Nakamura, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate. Presentado en diciembre del 2020. En *16th International Conference on Mobility, Sensing and Networking (MSN '20)*. (pp. 539-545). **CORE B.**
- Integrating an MQTT Proxy in a LoRa-Based Messaging System for Generic Sensor Data Collection. Kiyoshy Nakamura, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate. Presentado en octubre del 2020. En *19th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2020)*. (pp. 282-294) **CORE B.**
- FUDGE: a frugal edge node for advanced IoT solutions in contexts with limited resources. Kiyoshy Nakamura, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate, José M. Cecilia. Presentado en septiembre del 2020. En *1st Workshop on Experiences with the Design and Implementation of Frugal Smart Objects (FRUGALTHINGS '20)*. (pp. 30-35).

7. CONCLUSIONES, PUBLICACIONES Y TRABAJOS FUTUROS

- Adding voice messages to a low-cost long-range data messaging system. Kiyoshy Nakamura, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, Carlos T. Calafate. Presentado en septiembre del 2020. En 6th EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '20). (pp. 42-47).
- A LoRa enabled sustainable messaging system for isolated communities. Angelica Monero Cardenas, Miguel Kiyoshy Nakamura Pinto, Ermanno Pietrosemoli, Marco Zennaro, Marco Rainone, Pietro Manzoni. Presentado en noviembre del 2018. En 4th EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '18). (pp. 118-123).

Congresos Nacionales (España)

- Añadiendo mensajes de voz a un sistema de mensajería de datos de largo alcance y bajo coste. Kiyoshy Nakamura, Pietro Manzoni, Carlos T. Calafate, Juan-Carlos Cano, Enrique Hernandez-Orallo. Presentado en septiembre del 2021. En Sociedad de Arquitectura y Tecnología de Computadores (SARTECO 20/21), en las XXXI Jornadas de Paralelismo (JP20/21).
- Un sistema de mensajería viable para comunidades aisladas basado en LoRa. Kiyoshy Nakamura, Pietro Manzoni, Carlos T. Calafate, Juan-Carlos Cano, Enrique Hernandez-Orallo. Presentado en septiembre del 2019. En Sociedad de Arquitectura y Tecnología de Computadores (SARTECO 2019), en las XXX Jornadas de Paralelismo (JP2019).

Trabajos Futuros

En esta tesis proponemos el diseño de una arquitectura para redes comunitarias basada en la tecnología LoRa, el desarrollo del protocolo LoRaCTP, la integración al sistema de tecnologías pub/sub MQTT y nodos edge frugales. Con relación a nuestra solución, proponemos las siguientes tareas como trabajos futuros:

- Dotar de algún método de seguridad a los usuarios en el sistema de mensajería.
- Añadir un algoritmo inteligente a LoRaCTP, que permita cambiar entre SF 7 y SF 12 según escenarios específicos, tamaños de mensajes y otros aspectos.
- Proporcionar una estructura de malla entre los *hub* LoRa, para ampliar el alcance de la red y así abarcar distancias aún mayores.
- Proporcionar alguna forma de programación de envíos para evitar posibles interferencias con otras fuentes LoRa.
- Extender la topología en estrella que representamos en la arquitectura FUDGE a una malla para distribuir la carga entre varios agregadores y reducir la latencia.

Acrónimos

A

ACK	Acknowledgement
AI	Artificial Intelligence
AP	Access Point
API	Application Programming Interfaces
ARM	Acorn RISC Machines
ARQ	Automatic Repeat-reQuest
AWMN	Athens Wireless Metropolitan Network

B

BR	Bitrate
BW	Bandwidth
B4RN	Broadband for the Rural North

C

CN	Community Networks
CS	Chirps Sequence
CSS	Chirped Spread Spectrum

D

DR	Data Rate
----	-----------

E

EDL	Eclipse Distribution License
EPL	Eclipse Public License
EC	European Commission
EC-GSM-IoT	Enhanced Coverage-Global System for Mobile Internet of Things

F

FEC	Forward Error-correction Codes
FHSS	Frequency Hopping Spread Spectrum
FSK	Frequency-Shift Keying
FPS	Frames Per Second

G

GAIA	Global Access to the Internet for All Research Group
GHz	Gigahertz
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service
GPU	Graphics Processing Unit
GSM	Global System for Mobile communications
GSMA	Global System for Mobile communications Association

H

HF	High Frequency
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HPC	High Performance Computing

I

IaaS	Infrastructure as a Solution
ICT4D	Information and Communications Technologies for Development
IoT	Internet of Things
IRTF	Internet Research Task Force
ISM	Industrial, Scientific and Medical
ITU	International Telecommunication Union

J

JSON JavaScript Object Notation

K

KB Kilobyte

L

LCD Liquid Cristal Display
LoRa Long-Range
LoRaCTP LoRa based Content Transfer Protocol
LoRaWAN Long-Range Wide Area Network
LPWAN Low-Power Wide Area Networks
LTE Long Term Evolution

M

MAC Media Access Control
MAP Mobile Access Point
MB Megabyte
ML Machine Learning
MP3 Moving Picture Experts Group Audio Layer 3
MQTT Message Queuing Telemetry Transport
M2M Machine to Machine

N

NFC Near Field Communication

O

ODS Objetivo de Desarrollo Sostenible
OASIS Organization for the Advancement of Structured Information Standards

Q

QoS Quality of Service

R

RALN	Red Abierta, Libre y Neutral
RAM	Random Access Memory
RC	Redes Comunitarias
REST	REpresentational State Transfer
RF	Radio Frequency
RFID	Radio Frequency Identification
RPMA	Random Phase Multiple Access

S

SC	Spreading Code
SF	Spreading Factor
SMS	Short Message Service
SSID	Service Set Identifier
STA	Sistema de Tablón de Anuncios
STT	Successful Transfer Time

T

TCP/IP	Transmission Control Protocol/Internet Protocol
--------	---

U

UNESCO	United Nations Educational, Scientific and Cultural Organization
--------	--

W

WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WNAN	Wireless Neighborhood Area Network
WPAN	Wireless Personal Area Networks
WAV	Waveform Audio File Format
WWAN	Wireless Wide Area Networks

#

3G	3rd Generation of Wireless Mobile Telecommunications Technology
3GPP	3rd Generation Partnership Project

Bibliografía

- [1] ITU. *Measuring digital development: Facts and figures 2021*. Inf. téc. ITU, dic. de 2021 (vid. págs. 1, 2, 21).
- [2] G. Cruz, G. Touchard, M. Buckwell y F. Liberatore. *Enabling rural coverage: regulatory and policy recommendations to foster mobile broadband coverage in developing countries*. 2018 (vid. pág. 3).
- [3] G. Association y col. *The State of Mobile Internet Connectivity 2021*. URL: <https://www.gsma.com/r/wp-content/uploads/2021/09/The-State-of-Mobile-Internet-Connectivity-Report-2021.pdf>. (accessed: 10/03/2022) (vid. pág. 3).
- [4] E. Commission. *Shaping Europe's digital future*. URL: <https://ec.europa.eu/digital-single-market/en/broadband-strategy-policy/>. (accessed: 11/03/2022) (vid. pág. 3).
- [5] E. Commission. *Support for Broadband rollout*. URL: <https://digital-strategy.ec.europa.eu/en/policies/broadband-support/>. (accessed: 11/03/2022) (vid. pág. 3).
- [6] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. Navarro, R. B. Vias, D. Boucas, M. Michalis y P. Antoniadis. "Community networks and sustainability: a survey of perceptions, practices, and proposed solutions". En: *IEEE Communications Surveys & Tutorials* 20.4 (2018), págs. 3581-3606 (vid. pág. 7).

- [7] L. Navarro, R. Baig, F. Freitag y col. “Report on the Existing CNs and their Organization (v2)”. En: *netCommons Deliverable D 1* (2016) (vid. pág. 7).
- [8] B. Braem, C. Blondia, C. Barz y col. *A case for research with and on community networks*. 2013 (vid. pág. 7).
- [9] M Forzati, C Mattson, M Corbet y D Cullen. “Guide to High-Speed Broadband Investment”. En: *Monday, December 7, 2015*. European Commission. 2014 (vid. pág. 8).
- [10] A. Martinez, V. Villarroel, J. Seoane y F. del Pozo. “Analysis of information and communication needs in rural primary health care in developing countries”. En: *IEEE transactions on Information Technology in Biomedicine* 9.1 (2005), págs. 66-72 (vid. pág. 8).
- [11] R. Baig, R. Roca, F. Freitag y L. Navarro. “Guifi. net, a crowdsourced network infrastructure held in common”. En: *Computer Networks* 90 (2015), págs. 150-165 (vid. pág. 8).
- [12] A. Pentland, R. Fletcher y A. Hasson. “Daknet: Rethinking connectivity in developing nations”. En: *Computer* 37.1 (2004), págs. 78-83 (vid. págs. 8, 17, 18).
- [13] W. Waites, J. Sweet, R. Baig, P. Buneman, M. Fayed, G. Hughes, M. Fourman y R. Simmons. “Remix: A distributed internet exchange for remote and rural networks”. En: *Proceedings of the 2016 Workshop on Global Access to the Internet for All*. 2016, págs. 25-30 (vid. pág. 8).
- [14] B4RN. *Broadband for the Rural North Ltd*. URL: <https://b4rn.org.uk/>. (accessed: 14/03/2022) (vid. pág. 8).
- [15] Funkfeuer. *Funkfeuer free net*. URL: <https://www.funkfeuer.at/>. (accessed: 14/03/2022) (vid. pág. 8).
- [16] AWMN. *Athens Wireless Metropolitan Network*. URL: <https://www.www.awmn.net/>. (accessed: 16/03/2022) (vid. pág. 8).
- [17] S. Farrell, L. Toutain, A. Yegin y col. “Low-Power Wide Area Network (LPWAN) overview”. En: (2018) (vid. pág. 8).

-
- [18] K. Mekki, E. Bajic, F. Chaxel y F. Meyer. “A comparative study of LPWAN technologies for large-scale IoT deployment”. En: *ICT express* 5.1 (2019), págs. 1-7 (vid. pág. 8).
- [19] B. S. Chaudhari y M. Zennaro. “Introduction to low-power wide-area networks”. En: *LPWAN Technologies for IoT and M2M Applications*. Elsevier, 2020, págs. 1-13 (vid. pág. 8).
- [20] B. S. Chaudhari y M. Zennaro. *LPWAN Technologies for IoT and M2M Applications*. Academic Press, 2020 (vid. págs. 8-11).
- [21] Frost y Sullivan. *Growing Convergence of LPWAN and IoT Technologies*. URL: https://rfdesignuk.com/uploads/9/4/6/0/94609530/murata_lpwan_study.pdf. (accessed: 21/03/2022) (vid. pág. 10).
- [22] SIGFOX. *Sigfox, the 0G network*. URL: <https://www.sigfox.com/>. (accessed: 22/03/2022) (vid. pág. 12).
- [23] I. Inc. *RPMA Technology*. URL: <https://www.ingenu.com/technology/rpma/>. (accessed: 22/03/2022) (vid. pág. 12).
- [24] J. Finnegan y S. Brown. “A comparative survey of LPWA networking”. En: *arXiv preprint arXiv:1802.04222* (2018) (vid. pág. 12).
- [25] Qowisio. *Next Generation IoT Solutions*. URL: <https://www.qowisio.com/>. (accessed: 23/03/2022) (vid. pág. 12).
- [26] Nwave. *Revolutionary Solution for Smart Parking Management Systems*. URL: <https://www.nwave.io/>. (accessed: 23/03/2022) (vid. pág. 12).
- [27] SEMTECH. *LoRa Modulation Basics. AN1200.22*. URL: <http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf>. (accessed: 25/03/2022) (vid. pág. 12).
- [28] L. Alliance. *LoRa Alliance*. URL: <https://loro-alliance.org/>. (accessed: 25/03/2022) (vid. pág. 12).
- [29] Weightless. *Open Weightless*. URL: <https://www.openweightless.org/>. (accessed: 26/03/2022) (vid. pág. 12).
- [30] A. Høglund, X. Lin, O. Liberg y col. “Overview of 3GPP release 14 enhanced NB-IoT”. En: *IEEE network* 31.6 (2017), págs. 16-22 (vid. pág. 12).

- [31] S. Krishna Sharma y X. Wang. “Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions”. En: *arXiv e-prints* (2018), arXiv-1808 (vid. pág. 12).
- [32] M. Weyn, G. Ergeerts, L. Wante, C. Vercauteren y P. Hellinckx. “Survey of the DASH7 alliance protocol for 433 MHz wireless sensor communication”. En: *International Journal of Distributed Sensor Networks* 9.12 (2013), pág. 870430 (vid. pág. 12).
- [33] M. Bembe, A. Abu-Mahfouz, M. Masonta y T. Ngqondi. “A survey on low-power wide area networks for IoT applications”. En: *Telecommunication Systems* 71.2 (2019), págs. 249-274 (vid. pág. 13).
- [34] S. Lippuner, B. Weber, M. Salomon, M. Korb y Q. Huang. “EC-GSM-IoT network synchronization with support for large frequency offsets”. En: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2018, págs. 1-6 (vid. pág. 13).
- [35] B. Raman y K. Chebrolu. “Experiences in using WiFi for rural internet in India”. En: *IEEE Communications Magazine* 45.1 (2007), págs. 104-110 (vid. pág. 13).
- [36] A. Sheth, S. Nedeveschi, R. Patra, S. Surana, E. Brewer y L. Subramanian. “Packet loss characterization in WiFi-based long distance networks”. En: *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE. 2007, págs. 312-320 (vid. pág. 13).
- [37] S. Ahmadi. “An overview of next-generation mobile WiMAX technology”. En: *IEEE communications Magazine* 47.6 (2009), págs. 84-98 (vid. pág. 13).
- [38] K. Chebrolu, B. Raman y S. Sen. “Long-distance 802.11 b links: performance measurements and experience”. En: *Proceedings of the 12th annual international conference on Mobile computing and networking*. 2006, págs. 74-85 (vid. pág. 13).
- [39] R. K. Patra, S. Nedeveschi, S. Surana, A. Sheth, L. Subramanian y E. A. Brewer. “WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks.” En: *NSDI*. Vol. 1. 1. 2007, pág. 1 (vid. pág. 13).

-
- [40] R. Flickenger, S. Okay, E. Pietrosemoli, M. Zennaro y C. Fonda. “Very long distance wi-fi networks”. En: *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*. 2008, págs. 1-6 (vid. págs. 13, 14).
- [41] Guifi.net. *Red de Telecomunicaciones del Procomún, Abierta, Libre y Neutral*. URL: <https://guifi.net/es/>. (accessed: 17/03/2022) (vid. pág. 14).
- [42] M. Oliver, J. Zuidweg y M. Batikas. “Wireless commons against the digital divide”. En: *2010 IEEE International Symposium on Technology and Society*. IEEE. 2010, págs. 457-465 (vid. pág. 14).
- [43] R. Baig, R. Roca, L. Navarro y F. Freitag. “guifi. net: A network infrastructure commons”. En: *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*. 2015, págs. 1-4 (vid. pág. 14).
- [44] R. Baig, R. Roca, F. Freitag y L. Navarro. “Guifi. net, a crowdsourced network infrastructure held in common”. En: *Computer Networks* 90 (2015), págs. 150-165 (vid. pág. 14).
- [45] D. Vega, L. Cerda-Alabern, L. Navarro y R. Meseguer. “Topology patterns of a community network: Guifi. net”. En: *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2012, págs. 612-619 (vid. pág. 14).
- [46] Guifi.net. *Documentación de Guifi.net*. URL: <https://guifi.net/es/ESDocs/>. (accessed: 19/03/2022) (vid. pág. 15).
- [47] Guifi.net. *Estadísticas de nodos*. URL: <https://guifi.net/guifi/menu/stats/nodes/>. (accessed: 17/03/2022) (vid. pág. 16).
- [48] A. Sountharraj, S. Ashwin y M. A. Gokul. “An Overview: DakNet Technology-Broadband Ad-hoc Connectivity”. En: *Journal for Research/ Volume 4.01* (2018) (vid. pág. 17).
- [49] A. A. Hasson, R. Fletcher y A. Pentland. “DakNet: A road to universal broadband connectivity”. En: *Wireless Internet UN ICT Conference Case Study*. 2003, págs. 1-9 (vid. pág. 17).

- [50] J. Saldana, A. Arcia-Moret, E. Pietrosevoli, M. Zennaro, B. Braem y A. Sathiaselan. *Alternative Network Deployments: Taxonomy, Characterization, Technologies, and Architectures*. Inf. téc. 2016 (vid. pág. 19).
- [51] Statista. *Internet, demografía y uso*. URL: <https://es.statista.com/estadisticas/647829/porcentaje-de-poblacion-que-nunca-habia-usado-internet-espana/>. (accessed: 06/04/2022) (vid. pág. 22).
- [52] UNESCO. *Education and Literacy*. URL: <http://uis.unesco.org/en/visualisations/education-and-literacy>. (accessed: 07/04/2022) (vid. pág. 23).
- [53] J. Watkins, K. R. Kitner y D. Mehta. “Mobile and smartphone use in urban and rural India”. En: *Continuum* 26.5 (2012), págs. 685-697 (vid. pág. 24).
- [54] K. Heimerl, A. Menon, S. Hasan, K. Ali, E. Brewer y T. Parikh. “Analysis of smartphone adoption and usage in a rural community cellular network”. En: *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*. 2015, págs. 1-4 (vid. pág. 24).
- [55] W. Ma, R. Q. Grafton y A. Renwick. “Smartphone use and income growth in rural China: Empirical results and policy implications”. En: *Electronic Commerce Research* 20.4 (2020), págs. 713-736 (vid. pág. 24).
- [56] M. G. Smith y J. Urpelainen. “Early Adopters of Solar Panels in Developing Countries: Evidence from Tanzania”. En: *Review of Policy Research* 31.1 (2014), págs. 17-37 (vid. pág. 24).
- [57] A Al-Mamun, K Sundaraj, N Ahmed, N. Ahamed, S. Rahman, R. Ahmad y M. H. Kabir. “Design and development of a low cost solar energy system for the rural area”. En: *2013 IEEE Conference on Systems, Process & Control (ICSPC)*. IEEE. 2013, págs. 31-35 (vid. pág. 24).
- [58] S. Ornes. “Core Concept: The Internet of Things and the explosion of interconnectivity”. En: *Proceedings of the National Academy of Sciences* 113.40 (2016), págs. 11059-11060 (vid. pág. 25).
- [59] Statista. *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025*. URL: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. (accessed: 09/04/2022) (vid. pág. 27).

-
- [60] A. Sunyaev. “The internet of things”. En: *Internet Computing*. Springer, 2020, págs. 301-337 (vid. pág. 27).
- [61] S. Madakam, V. Lake, V. Lake, V. Lake y col. “Internet of Things (IoT): A literature review”. En: *Journal of Computer and Communications* 3.05 (2015), pág. 164 (vid. pág. 28).
- [62] A. Berni y W. Gregg. “On the utility of chirp modulation for digital signaling”. En: *IEEE Transactions on Communications* 21.6 (1973), págs. 748-751 (vid. pág. 29).
- [63] SEMTECH. *LoRa Developer Portal*. URL: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>. (accessed: 09/04/2022) (vid. págs. 30, 33).
- [64] SEMTECH. *Semtech SX1272*. URL: <https://www.semtech.com/products/wireless-rf/lora-core/sx1272/>. (accessed: 09/04/2022) (vid. pág. 30).
- [65] SEMTECH. *Semtech S1276*. URL: <https://www.semtech.com/products/wireless-rf/lora-core/sx1276/>. (accessed: 09/04/2022) (vid. pág. 30).
- [66] A. Zourmand, A. L. K. Hing, C. W. Hung y M. AbdulRehman. “Internet of things (IoT) using LoRa technology”. En: *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE, 2019, págs. 324-330 (vid. pág. 30).
- [67] SEMTECH. *LoRa Modulation Basics*. URL: <http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf/>. (accessed: 10/04/2022) (vid. pág. 30).
- [68] L. Vangelista, A. Zanella y M. Zorzi. “Long-range IoT technologies: The dawn of LoRa™”. En: *Future access enablers of ubiquitous and intelligent infrastructures*. Springer, 2015, págs. 51-58 (vid. pág. 30).
- [69] S. M. Thampi, S. Mitra, J. Mukhopadhyay, K.-C. Li, A. P. James y S. Berretti. *Intelligent Systems Technologies and Applications*. Springer, 2020 (vid. págs. 31, 32, 35).
- [70] D.-H. Kim, E.-K. Lee y J. Kim. “Experiencing LoRa network establishment on a smart energy campus testbed”. En: *Sustainability* 11.7 (2019), pág. 1917 (vid. pág. 32).

- [71] B. Robbe y W. Danny. “A qos-aware adaptive mobility handling approach for lora-based iot systems”. En: *2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE. 2018, págs. 130-139 (vid. pág. 33).
- [72] L. Alliance. *LoRaWAN Specification*. URL: <https://loro-alliance.org/about-lorawan/>. (accessed: 12/04/2022) (vid. pág. 33).
- [73] M. Liya y M Aswathy. “LoRa technology for Internet of Things (IoT): a brief survey”. En: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE. 2020, págs. 8-13 (vid. pág. 34).
- [74] K.-L. Tsai, F.-Y. Leu, I. You, S.-W. Chang, S.-J. Hu y H. Park. “Low-power AES data encryption architecture for a LoRaWAN”. En: *IEEE Access* 7 (2019), págs. 146348-146357 (vid. pág. 34).
- [75] P. A. Catherwood, D. Steele, M. Little, S. McComb y J. McLaughlin. “A community-based IoT personalized wireless healthcare solution trial”. En: *IEEE journal of translational engineering in health and medicine* 6 (2018), págs. 1-13 (vid. pág. 35).
- [76] J. Petäjäljärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen y J. Iinatti. “Evaluation of LoRa LPWAN technology for indoor remote health and wellbeing monitoring”. En: *International Journal of Wireless Information Networks* 24.2 (2017), págs. 153-165 (vid. pág. 35).
- [77] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad y M. Ismail. “Energy-efficient wireless sensor networks for precision agriculture: A review”. En: *Sensors* 17.8 (2017), pág. 1781 (vid. pág. 35).
- [78] S. Benaissa, D. Plets, E. Tanghe y col. “Internet of animals: characterisation of LoRa sub-GHz off-body wireless channel in dairy barns”. En: *Electronics Letters* 53.18 (2017), págs. 1281-1283 (vid. pág. 35).
- [79] D. Sartori y D. Brunelli. “A smart sensor for precision agriculture powered by microbial fuel cells”. En: *2016 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2016, págs. 1-6 (vid. pág. 35).
- [80] D. Ilie-Ablachim, G. C. Pătru, I.-M. Florea y D. Rosner. “Monitoring device for culture substrate growth parameters for precision agriculture: Acronym:

- MoniSen”. En: *2016 15th RoEduNet Conference: Networking in Education and Research*. IEEE. 2016, págs. 1-7 (vid. pág. 35).
- [81] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadiania y N. Strachan. “Evaluation of LoRa and LoRaWAN for wireless sensor networks”. En: *2016 IEEE SENSORS*. IEEE. 2016, págs. 1-3 (vid. pág. 35).
- [82] H.-C. Lee y K.-H. Ke. “Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation”. En: *IEEE Transactions on Instrumentation and Measurement* 67.9 (2018), págs. 2177-2187 (vid. pág. 35).
- [83] V. Sharma, I. You, G. Pau, M. Collotta, J. D. Lim y J. N. Kim. “LoRaWAN-based energy-efficient surveillance by drones for intelligent transportation systems”. En: *Energies* 11.3 (2018), pág. 573 (vid. pág. 35).
- [84] MQTT. *MQTT Specifications*. URL: <https://mqtt.org/mqtt-specification/>. (accessed: 17/03/2022) (vid. pág. 36).
- [85] B. Mishra y A. Kertesz. “The use of MQTT in M2M and IoT systems: A survey”. En: *IEEE Access* 8 (2020), págs. 201071-201086 (vid. pág. 36).
- [86] V. Lampkin, W. T. Leong, L. Olivera y col. *Building smarter planet solutions with mqtt and ibm websphere mq telemetry*. IBM Redbooks, 2012 (vid. pág. 36).
- [87] U. Hunkeler, H. L. Truong y A. Stanford-Clark. “MQTT-S—A publish/-subscribe protocol for Wireless Sensor Networks”. En: *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. IEEE. 2008, págs. 791-798 (vid. pág. 36).
- [88] D. Soni y A. Makwana. “A survey on mqtt: a protocol of internet of things (iot)”. En: *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*. Vol. 20. 2017, págs. 173-177 (vid. pág. 37).
- [89] M. Singh, M. Rajan, V. Shivraj y P. Balamuralidhar. “Secure mqtt for internet of things (iot)”. En: *2015 fifth international conference on communication systems and network technologies*. IEEE. 2015, págs. 746-751 (vid. pág. 38).

- [90] E. Foundation. *Eclipse Mosquitto, an open source MQTT broker*. URL: <http://mosquitto.org/>. (accessed: 19/03/2022) (vid. pág. 39).
- [91] HiveMQ. *HiveMQ MQTT Broker*. URL: <https://www.hivemq.com/>. (accessed: 19/03/2022) (vid. pág. 39).
- [92] E. Inc. *EMQ X*. URL: <https://www.emqx.io/>. (accessed: 19/03/2022) (vid. pág. 39).
- [93] IBM. *IBM Watson IoT Platform - Message Gateway*. URL: <https://www.ibm.com/docs/en/wip-mg/5.0.0.1/>. (accessed: 19/03/2022) (vid. pág. 39).
- [94] P. Software. *RabbitMQ Broker*. URL: <https://www.rabbitmq.com/>. (accessed: 19/03/2022) (vid. pág. 39).
- [95] H. T. Dinh, C. Lee, D. Niyato y P. Wang. “A survey of mobile cloud computing: architecture, applications, and approaches”. En: *Wireless communications and mobile computing* 13.18 (2013), págs. 1587-1611 (vid. pág. 39).
- [96] X. Sun y N. Ansari. “EdgeIoT: Mobile edge computing for the Internet of Things”. En: *IEEE Communications Magazine* 54.12 (2016), págs. 22-29 (vid. págs. 40, 41).
- [97] W. Shi, J. Cao, Q. Zhang, Y. Li y L. Xu. “Edge computing: Vision and challenges”. En: *IEEE internet of things journal* 3.5 (2016), págs. 637-646 (vid. pág. 40).
- [98] M. Satyanarayanan, P. Bahl, R. Caceres y N. Davies. “The case for vm-based cloudlets in mobile computing”. En: *IEEE pervasive Computing* 8.4 (2009), págs. 14-23 (vid. pág. 39).
- [99] K. Cao, Y. Liu, G. Meng y Q. Sun. “An overview on edge computing research”. En: *IEEE access* 8 (2020), págs. 85714-85728 (vid. pág. 39).
- [100] F. Bonomi, R. Milito, P. Natarajan y J. Zhu. “Fog computing: A platform for internet of things and analytics”. En: *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, págs. 169-186 (vid. pág. 39).
- [101] M. Mobasheri, Y. Kim y W. Kim. “Toward developing fog decision making on the transmission rate of various IoT devices based on reinforcement

-
- learning”. En: *IEEE Internet of Things Magazine* 3.1 (2020), págs. 38-42 (vid. pág. 39).
- [102] R. Atat, L. Liu, H. Chen, J. Wu, H. Li e Y. Yi. “Enabling cyber-physical communication in 5G cellular networks: Challenges, spatial spectrum sensing, and cyber-security”. En: *IET Cyber-Physical Systems: Theory & Applications* 2.1 (2017), págs. 49-54 (vid. pág. 39).
- [103] K. Wang, Y. Wang, Y. Sun, S. Guo y J. Wu. “Green industrial Internet of Things architecture: An energy-efficient perspective”. En: *IEEE Communications Magazine* 54.12 (2016), págs. 48-54 (vid. pág. 39).
- [104] J. Wu, S. Guo, J. Li y D. Zeng. “Big data meet green challenges: Big data toward green applications”. En: *IEEE Systems Journal* 10.3 (2016), págs. 888-900 (vid. pág. 39).
- [105] S. Weisong, Z. Xingzhou, W. Yifan y Z. Qingyang. “Edge computing: state-of-the-art and future directions”. En: *Journal of Computer Research and Development* 56.1 (2019), pág. 69 (vid. pág. 40).
- [106] W. Shi, H. Sun, J. Cao, Q. Zhang y W. Liu. “Edge computing—an emerging computing model for the internet of everything era”. En: *Journal of computer research and development* 54.5 (2017), pág. 907 (vid. pág. 40).
- [107] Z. Ziming, L. Fang, C. Zhiping y X. Nong. “Edge computing: platforms, applications and challenges”. En: *Journal of computer research and development* 55.2 (2018), pág. 327 (vid. pág. 40).
- [108] P. Mach y Z. Becvar. “Mobile edge computing: A survey on architecture and computation offloading”. En: *IEEE Communications Surveys & Tutorials* 19.3 (2017), págs. 1628-1656 (vid. pág. 41).
- [109] Y. Mao, C. You, J. Zhang, K. Huang y K. B. Letaief. “A survey on mobile edge computing: The communication perspective”. En: *IEEE communications surveys & tutorials* 19.4 (2017), págs. 2322-2358 (vid. pág. 41).
- [110] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta y D. Sabella. “On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration”. En: *IEEE Communications Surveys & Tutorials* 19.3 (2017), págs. 1657-1681 (vid. pág. 41).

- [111] G. Prensankar, M. Di Francesco y T. Taleb. “Edge computing for the Internet of Things: A case study”. En: *IEEE Internet of Things Journal* 5.2 (2018), págs. 1275-1284 (vid. pág. 42).
- [112] R. Mukherjee y S. Das. “Impact of Dithering in Compressed and Spectrally Distorted Speech for Emotion Recognition”. En: () (vid. pág. 68).
- [113] G. E. Hinton y R. R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. En: *science* 313.5786 (2006), págs. 504-507 (vid. pág. 69).
- [114] K. Park y T. Mulc. “Cssl0: A collection of single speaker speech datasets for 10 languages”. En: *arXiv preprint arXiv:1903.11269* (2019) (vid. pág. 69).
- [115] M. A. Guillén, A. Llanes, B. Imbernón, R. Martínez-España, A. Bueno-Crespo, J.-C. Cano y J. M. Cecilia. “Performance evaluation of edge-computing platforms for the prediction of low temperatures in agriculture using deep learning”. En: *The Journal of Supercomputing* 77.1 (2021), págs. 818-840 (vid. pág. 81).
- [116] H. Gedawy, K. A. Harras, K. Habak y M. Hamdi. “Femtoclouds beyond the edge: The overlooked data centers”. En: *IEEE Internet of Things Magazine* 3.1 (2020), págs. 44-49 (vid. pág. 81).
- [117] F. Bonomi, R. Milito, J. Zhu y S. Addepalli. “Fog computing and its role in the internet of things”. En: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, págs. 13-16 (vid. pág. 81).