



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



## **Universitat Politècnica de València**

**Departamento de Informática de Sistemas y Computadores**

# **Sistema de Mensajería Instantánea Basado en LoRa**

TRABAJO DE FIN DE MÁSTER

Máster Universitario en Ingeniería de Computadores y Redes

*Autor*

Miguel Kiyoshy Nakamura Pinto

*Director*

Dr. Pietro Manzoni

A mis padres Miguel y Betty porque a pesar de mis tropiezos  
nunca han dejado de apoyarme y alentarme para seguir adelante.  
Gracias a ustedes soy lo que soy.

A mis hermanos Haru y Yoshio quienes siempre  
me han respaldado en cada reto que inicio.

A mi familia y amigos quienes han estado  
animándome sin importar la distancia.

# Agradecimientos

A mi director de tesis y coordinador del Grupo de Redes de Computadores (GRC) Dr. Pietro Manzoni, por la confianza y paciencia depositada durante el desarrollo del proyecto, así como su valiosa guía durante la redacción del presente trabajo de fin de máster.

A todos los docentes del Máster Universitario en Ingeniería de Computadores y Redes (MUICR), por compartir con nosotros sus conocimientos y experiencias a lo largo de su trayectoria profesional.

A mis compañeros de clase, con quienes gracias a los retos y al buen trabajo en equipo hemos terminado siendo amigos. En especial a mi amiga Angélica por todo el apoyo durante el desarrollo del proyecto.

A mis compañeros y amigos de Chiapas en México quienes de una u otra forma fueron de gran ayuda durante la convocatoria para la obtención de la beca.

Al Gobierno de la República Mexicana a través del Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Gobierno de Chiapas mediante el Consejo de Ciencia y Tecnología del Estado (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.

## Resumen

A pesar de los grandes avances tecnológicos que se han desarrollado en los últimos años, aún hay más de mil millones de personas que no se encuentran bajo la cobertura de la red GSM debido a que viven en países en vías de desarrollo o en comunidades aisladas, donde la infraestructura de comunicaciones es inexistente o requiere de mucha inversión económica lo cual representaría un gasto muy grande tanto para los gobiernos como para las empresas privadas.

Basándonos en la problemática ya mencionada hemos desarrollado un sistema que entra tanto en la categoría de redes tolerantes a retardos como en la categoría de redes comunitarias, donde la característica principal es que los nuevos participantes pueden unirse a estas redes, hacer uso de ellas libremente, además pueden contribuir con la expansión de este tipo de redes aportando infraestructura propia. El sistema desarrollado utiliza dispositivos de bajo costo, que además son de bajo consumo de energía basados en el protocolo LoRa, mediante los cuales se puede otorgar un sistema de mensajería donde no existiría ningún costo como en los sistemas tradicionales.

Las redes basadas en LoRa tienen la capacidad de lograr interconectar enlaces inalámbricos a largas distancias, por medio de estos se pueden comunicar pueblos o comunidades aisladas con las ciudades ayudando a romper la brecha de comunicaciones existente. Otros de los usos que se le pueden dar a las redes LoRa son los de recolectar información de una red de sensores para proporcionarla a cierta comunidad y en el peor de los casos poder transmitir alertas de desastres para poder alertar a la población.

**Palabras claves:** Redes Comunitarias (CN), Redes Tolerantes a Retardos (DTN), LoRa, LoPy.

## **Abstract**

Despite the great technological advances that have been developed in recent years, there are still more than one billion people who are not covered by the GSM network because they live in developing countries or in isolated communities, where the communications infrastructure is non-existent or requires a lot of economic investment, which would represent a huge expense for both governments and private companies.

Based on the aforementioned problems, we have developed a system that falls into the category of delay-tolerant networks as well as in the category of community networks, where the main characteristic is that new entrants can join these networks, make use of them freely, They can also contribute to the expansion of this type of networks by providing their own infrastructure. The developed system uses low-cost devices, which are also low power consumption based on the LoRa protocol, through which a messaging system can be granted where there would be no cost as in traditional systems.

LoRa-based networks have the ability to interconnect long-distance wireless links, through which they can communicate isolated towns or communities with cities, helping to break the existing communication gap. Other uses that can be given to networks LoRa are to collect information from a network of sensors to provide it to a certain community and in the worst case to transmit disaster alerts to be able to alert the population.

**Keywords:** Community Networks (CN), Delayed Tolerant Networks (DTN), LoRa, LoPy.

# Índice general

<b>Índice de figuras</b>	<b>iii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	2
1.2 Objetivos . . . . .	2
1.3 Metodología . . . . .	2
1.4 Estructura . . . . .	3
<b>2 Antecedentes</b>	<b>4</b>
2.1 Redes Comunitarias . . . . .	4
2.1.1 DakNet . . . . .	7
2.2 Redes Tolerantes al Retardo . . . . .	9
2.3 LORA . . . . .	11
2.3.1 Capa Física LoRa . . . . .	14
2.3.2 LoRa MAC . . . . .	16
<b>3 Análisis</b>	<b>18</b>
3.1 Protocolo . . . . .	19
3.2 Estructura del paquete, flujo y almacenamiento de datos . . . . .	20
<b>4 Diseño y Metodología de Desarrollo</b>	<b>22</b>
4.1 Dispositivos . . . . .	22
4.1.1 Lopy . . . . .	23
4.1.2 <i>Expansion Board 2.0</i> . . . . .	23
4.1.3 Antena LoRa & Sigfox . . . . .	24
4.2 Servidor Web . . . . .	25
4.3 Funciones principales . . . . .	28
<b>5 Resultados</b>	<b>37</b>
5.1 Nodos . . . . .	37
5.2 Escenarios . . . . .	39

5.3 Pruebas . . . . .	40
<b>6 Conclusiones</b>	<b>43</b>
6.1 Conclusiones . . . . .	43
6.2 Trabajos futuros . . . . .	44
<b>Lista de Acrónimos</b>	<b>45</b>
<b>Bibliografía</b>	<b>46</b>

# Índice de figuras

1	Curva de crecimiento guifi.net [9] . . . . .	6
2	Incremento anual guifi.net [9] . . . . .	7
3	Esquema DakNet [11] . . . . .	8
4	Ejemplo de una red DTN [18] . . . . .	10
5	Arquitectura de un sistema LoRa [26] . . . . .	13
6	Arquitectura del Protocolo LoRa [26] . . . . .	14
7	Variación de frecuencia a través del tiempo de una señal LoRa . . . . .	15
8	Clases LoRaWAN [32] . . . . .	17
9	Estructura general de la plataforma de mensajería . . . . .	19
10	Esquema de protocolo <i>Stop and Wait</i> . . . . .	19
11	Proceso de descubrimiento . . . . .	20
12	Estructura del paquete . . . . .	21
13	Estructura de la cabecera del paquete . . . . .	21
14	Microcontrolador LoPy . . . . .	23
15	Dispositivo Pycom . . . . .	24
16	Antena LoRa & Sigfox . . . . .	24
17	Pantalla de registro de usuario . . . . .	25
18	Pantalla de advertencia . . . . .	25
19	Pantalla de envío y recepción de mensajes . . . . .	26
20	Ejemplo de envío de mensaje a un solo usuario . . . . .	27
21	Pantalla de transmisión exitosa . . . . .	27
22	Pantalla de mensajes recibidos . . . . .	28
23	Proceso de cambio de pantallas y almacenamiento de datos . . . . .	29
24	Diagrama de proceso de transmisión . . . . .	30
25	Ejemplo de transmisión visto desde la consola . . . . .	31
26	Diagrama de proceso de recepción . . . . .	33
27	Proceso de reconocimiento visto desde la consola . . . . .	34
28	Proceso de recepción y guardado de mensaje visto desde la consola . . . . .	35
29	Función para el control de <i>sockets</i> . . . . .	36



30	Nodo . . . . .	37
31	LoPy, panel solar y batería . . . . .	38
32	Nombres de red de los dispositivos . . . . .	38
33	Escenarios de pruebas . . . . .	39
34	Proceso de descubrimiento de usuario . . . . .	40
35	Comportamiento del STT al variar la distancia entre dos nodos . .	41
36	Comportamiento del STT al variar el tamaño del mensaje . . . . .	41
37	Evaluación de escalabilidad del sistema al aumentar la cantidad de usuarios que envían simultáneamente . . . . .	42

# Capítulo 1

## Introducción

Poder transmitir información a grandes distancias y en el menor tiempo posible siempre ha traído grandes beneficios para los individuos y sus comunidades, tanto para el desarrollo socio-económico como en aspectos de prevención y seguridad. A lo largo de los años los desarrollos tecnológicos en la rama de las comunicaciones han buscado por todos los medios mantener conectadas al mayor número de personas, aunque para lograr esto se han tenido que realizar fuertes inversiones económicas para desplegar la infraestructura necesaria.

Según las estadísticas del 2017 de la *International Telecommunication Union (ITU)*, se reporta que en los últimos cinco años ha habido un incremento de más del 20 % en las suscripciones de banda ancha móvil, logrando así cubrir al 56,4 % de la población mundial a finales del 2017 [1]. Cabe resaltar que en los países menos desarrollados los precios de banda ancha móvil representan más del 5 % del INB per cápita, dejando así a gran parte de la población fuera del alcance de este tipo de servicios. Es totalmente claro que muchas ciudades alrededor del mundo gozan de los servicios de conectividad 3G y 4G, pero también es bien sabido que en muchos países aun en vías de desarrollo siguen existiendo zonas rurales donde los medios de comunicación entre estas zonas y las ciudades son muy limitados y en algunos casos son los sistemas de mensajería más simples mediante los cuales se intenta brindar un servicio de comunicación. Dejando así a más de 1,2 mil millones de personas según [2], sin ningún tipo de cobertura de conectividad. El principal obstáculo para poder realizar un despliegue de infraestructura de comunicaciones en áreas rurales es un desafío económico donde el costo de dicha infraestructura puede ser el doble de caro, mientras que los ingresos pueden ser hasta diez veces más bajos.

## 1.1 Motivación

El gran aumento de proyectos para realizar despliegues de infraestructura de comunicaciones para lograr interconectar zonas rurales con las ciudades fue una de las causas principales por las que decidimos desarrollar esta solución, utilizando dispositivos que fueran económicos, de bajo consumo energético y que tuvieran el mayor alcance posible. Los dispositivos deben ser económicos para no generar una inversión muy significativa para las poblaciones; deben ser de bajo consumo energético para que puedan mantenerse activos con pequeños bancos de batería o en su caso un panel solar pequeño; además deberán tener la capacidad de enviar información a largas distancias. La solución que proponemos tiene como propósito principal ofrecer servicios de mensajería a áreas aisladas, que no tengan ninguna cobertura de comunicación. El poder proveer de un servicio de mensajería barato y de poco consumo energético en zonas rurales, donde aún no se dispone de acceso a las redes de comunicaciones existentes será una aplicación que brindará muchos beneficios a las poblaciones, además en este sistema utilizamos de manera conveniente herramientas *Internet of Things (IoT)*. Cabe resaltar que en el desafortunado caso de existir algún desastre y la infraestructura celular sufriera interrupciones, la red *Long Range (LoRa)* es capaz de intercambiar mensajes de emergencia a grandes distancias lo que sería de gran ayuda en esos momentos críticos.

## 1.2 Objetivos

Este Trabajo de Fin de Máster tiene como objetivos:

- Desarrollar un Sistema de Mensajería Instantánea basado en LoRa.
- Realizar pruebas de envío de mensajes de diferentes tamaños con el sistema desarrollado a diferentes distancias.
- Analizar los resultados obtenidos en las diferentes pruebas realizando modificaciones en las características del envío y del mensaje.
- Definir trabajos futuros.

## 1.3 Metodología

Para el desarrollo de este sistema de mensajería se utilizaron dispositivos de bajo costo y bajo consumo energético, que a su vez sean capaces de transmitir información a distancias muy largas.

Realizamos diversos experimentos con la solución que se propone haciendo variaciones de configuración durante la transmisión de información entre los nodos a diferentes distancias en un entorno controlado, obteniendo así una serie de datos que sirvieron para la obtención de resultados.

Una vez analizando y evaluando los resultados se han definido algunas conclusiones y se plantean trabajos futuros.

## **1.4 Estructura**

El presente Trabajo de Fin de Máster consta de seis capítulos. El capítulo 1 en donde nos encontramos ofrece una visión previa de la problemática que pretendemos abordar, junto con los objetivos generales del trabajo. En el capítulo 2 encontramos algunas de las soluciones que ya se han propuesto para combatir la brecha de comunicaciones, así como algunos detalles de *Community Networks (CN)*, *Delay Tolerant Network (DTN)* y LoRa. El capítulo 3 nos brinda una descripción general del sistema de mensajería que se ha desarrollado. Una descripción más específica y con mayor grado de detalle a cerca del funcionamiento del sistema y las tecnologías usadas podremos encontrarlas en el capítulo 4. En el capítulo 5 hemos de presentar y analizar los resultados obtenidos en los experimentos. Por último en el capítulo 6 detallaremos las conclusiones obtenidas y haremos referencia a trabajos futuros.

# Capítulo 2

## Antecedentes

Aun en pleno siglo XXI existen países en vías de desarrollo o comunidades rurales donde los sistemas de mensajería más simples recientemente se están haciendo presentes o en su defecto son inexistentes lo cual complica la tarea para erradicar la brecha de comunicaciones que mantienen a los pobladores de estos lugares aislados.

El poder mantener comunicación entre comunidades rurales y a su vez con las ciudades principales es uno de los servicios que brindaría mayores beneficios socio-económicos a los pobladores. Esta clase de servicios pueden contribuir, por ejemplo, cuando una persona se tiene que trasladar a la ciudad para agendar una cita médica, regresar a su comunidad y después tener que volver a la ciudad en la fecha asignada anteriormente. Esta clase de situaciones se evitarían al contar con servicios de comunicación previniendo de esta forma tener que realizar dos largos y agotadores viajes a la ciudad, además de ahorrar el doble gasto para transportarse. Un análisis de las necesidades de comunicación en la atención primaria de salud rural en países en desarrollo se detalla en [3].

### 2.1 Redes Comunitarias

Las *Community Networks (CN)* son redes de construcción y gestión colectiva, las cuales tienen como características principales ser neutrales, abiertas y libres. Para la creación de estas redes los participantes (ciudadanos u organizaciones) realizan aportaciones de infraestructura y se gestionan como un recurso común. Contrarrestar la brecha digital y ampliar la cobertura en zonas desatendidas son algunas de las motivaciones principales en la creación de estas redes. La infraestructura con la que se construyen las CN varía dependiendo del tipo de participante que haga la aportación, pueden ser desde *routers WiFi* de bajo coste, hasta equipos de

fibra óptica.

Se dice que este tipo de redes son libres ya que su principal impulso es la no discriminación, por lo tanto, son universales. Se consideran abiertas porque la información acerca de su construcción está disponible para todo el mundo. Son neutrales porque los participantes pueden hacer uso de la red y enviar datos de cualquier tipo, así como para usos comerciales; a su vez se puede hacer uso de cualquier infraestructura para ampliar la cobertura de la red.

*La expresión red de comunicación gestionada de forma colectiva se refiere a una infraestructura de red integrada por ciudadanos y organizaciones que ponen en común sus recursos y coordinan sus esfuerzos para crear estas redes [4].*

En los países en vías de desarrollo o zonas rurales donde las instituciones públicas o proveedores de servicio han dejado a la población sin cobertura de comunicación, las CN [5] han sido una solución con amplias expectativas, brindando soluciones de conectividad y de bajo costo para las personas que habitan en esa zona. A su vez el fácil acceso a dispositivos inalámbricos se ha hecho visible con la popularidad de la amplia gama de *laptops* y dispositivos móviles (*smartphones* y *tablets*) con capacidad de conexión *WiFi* a un bajo costo. Originando en los últimos años un sin número de iniciativas sin fines de lucro para apoyar en la construcción de CN utilizando una infraestructura inalámbrica y de esta forma poder brindar servicios de comunicaciones. Por mencionar algunas de estas iniciativas con éxito tenemos Funkfeuer [6], Guifi.net [7] y la Red Comunitaria Metropolitana Inalámbrica de Atenas [8].

Este tipo de soluciones tienen una característica muy peculiar, donde su principal objetivo es interconectar cada vez más nodos de nuevos participantes sin tener en cuenta una topología planificada ni ninguna otra consideración. Estas características son complicadas de controlar debido a que los nuevos participantes tienen la libertad de instalar sus nodos de interconexión mediante dispositivos hardware colocados en los techos de sus viviendas y de esta forma extender la red.

Con más de 35,000 nodos operativos y con una cobertura mayor a 63,000 km mediante enlaces inalámbricos [9], Guifi.net [7, 10] es una CN tipo malla *WiFi* que además es neutral, sin ánimos de lucro, de acceso abierto e independiente, la cual quizá sea la más grande en el mundo (número de nodos y participantes) y aún con crecimiento exponencial.

Guifi.net tiene ciertas regulaciones y/o derechos dentro de las que destacan que los nuevos participantes pueden hacer uso libremente de la red siempre y

cuando no afecte a otros usuarios o en su defecto el funcionamiento de la red, pueden conocer ampliamente la infraestructura de la red si así lo desean y de ser capaces unirse o ampliar la red siguiendo estas mismas regulaciones y derechos. Estos son algunos de los acuerdos que tienen que aceptar los nuevos participantes y que a su vez se encuentran coordinadas por la fundación Guifi.net. La red inicio sus operaciones en 2004 y en 2018 tiene más de 35,000 nodos operativos y poco más de 18,000 nodos proyectados.

La figura (1) ilustra claramente la gran popularidad que ha venido ganando año con año Guifi.net llegando a ser considerada la CN más grande del mundo en relación a nodos operativos y participantes. En la figura (2) podemos observar la cantidad de nodos que se han ido añadiendo año con año y que se encuentran en funcionamiento.

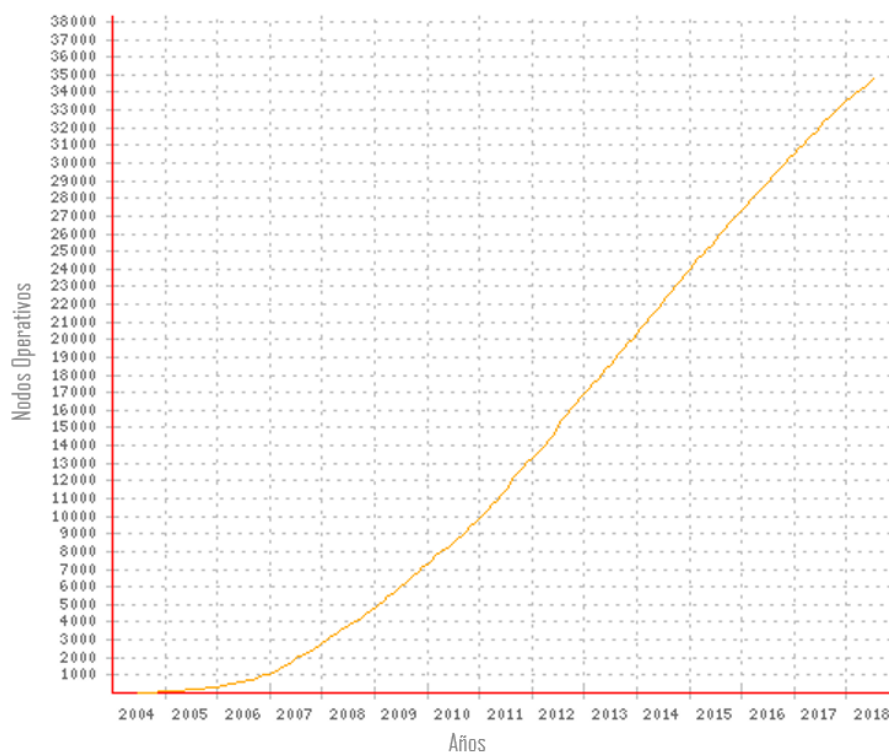


Figura 1: Curva de crecimiento guifi.net [9]

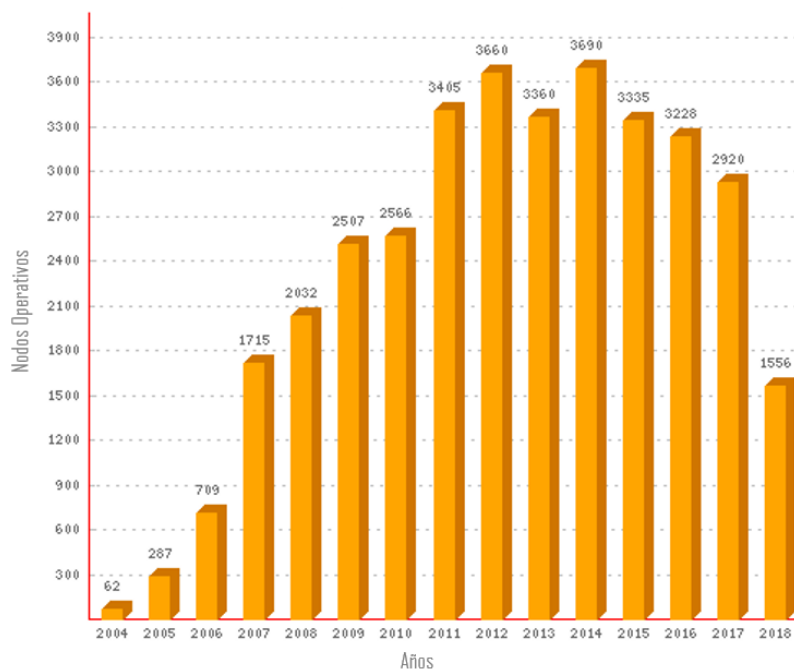


Figura 2: Incremento anual guifi.net [9]

### 2.1.1 DakNet

Una de las múltiples soluciones que se han propuesto hasta el momento para romper la brecha de conectividad en los países en vías de desarrollo, donde las áreas rurales carecen de los beneficios de conectividad 3G o 4G es DakNet [11] un sistema de mensajería empleado para proporcionar servicios de comunicación más allá de la cobertura de las redes GSM.

Este sistema se beneficia de la infraestructura existente en la comunidad tanto de transporte como de comunicaciones para brindar conectividad digital, de tal forma que hace uso de un medio físico de transporte con transferencia de datos inalámbrica y de esta forma potenciar la conectividad a Internet. Dentro de las utilidades que puede proporcionar DakNet a las comunidades son el correo electrónico, sistemas de boletín electrónico y correo de voz apoyándose en una infraestructura tolerante al retardo que a su vez es mucho más barato de lo que sería un servicio tradicional.

La forma de comunicación de este sistema es a través de enlaces cortos punto a punto que se encuentran en kioscos de la comunidad y los vehículos de transporte público que brindan servicio dentro del área los cuales han sido equipados



previamente con un transceptor *WiFi* que se encuentra conectado a un servidor, de tal forma que cuando el vehículo se encuentra dentro del alcance de comunicación con alguno de los kioscos los datos se sincronizan entre ambas partes. Los kioscos mantendrán la información que ha sido actualizada y los usuarios finales podrán consultarla mediante su dispositivo móvil o tableta a través de *WiFi*.

La figura (3) nos muestra como DakNet aprovecha la infraestructura existente para transmitir los datos, lo cual se logra mediante enlaces cortos punto a punto entre kioscos y dispositivos de almacenamiento portátiles, llamados *Mobile Access Point (MAP)*. Los MAP pueden ser transportados y además alimentados por una bicicleta con un pequeño generador, una motocicleta o un autobús, estos dispositivos son capaces de almacenar los datos y entregarlos entre kioscos y un concentrador (para Acceso a Internet en tiempo no real) o entre kioscos públicos y dispositivos de comunicaciones privados (como una *intranet*). Cada vez que se logra una conexión punto a punto los datos almacenados en el MAP se envían mediante los transceptores de radio *WiFi* a un ancho de banda alto y de esta manera aprovechar el tiempo de conexión.

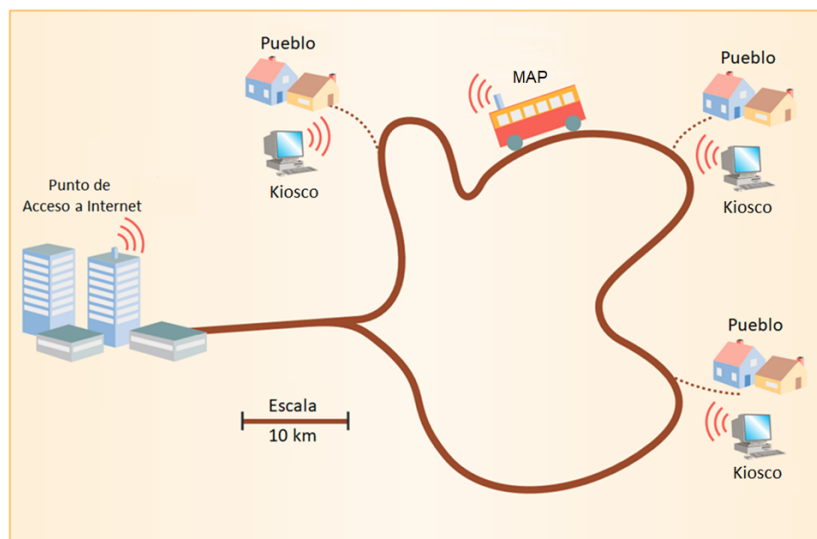


Figura 3: Esquema DakNet [11]

DakNet tiene dos formas de operación:

- Una vez que el vehículo dotado con el MAP se encuentre dentro del rango *WiFi* del kiosco que corresponda a la comunicación de un pueblo, detectará la conexión inalámbrica automáticamente mediante la cual podrá descargar y cargar decenas de megabytes de datos los cuales podrán ser consultados por los usuarios finales.

- Cuando un vehículo equipado con el MAP se encuentra dentro del rango *WiFi* de un punto de acceso que tiene conexión a Internet, realiza un proceso de sincronización de los datos en todos los kioscos rurales de manera automática a través de Internet.

Estos pasos pueden repetirse en todos los vehículos que estén equipados con un MAP y de esta manera crean una red de comunicaciones sin interrupciones además de ser una infraestructura de red inalámbrica de bajo costo.

Con al menos un vehículo que circule al día por la ruta donde se encuentren los kioscos en los pueblos que se están comunicando será suficiente para poder proveer de servicios de información diarios. A pesar de que DakNet no es un sistema que realice envíos de datos en tiempo real posee una calidad de conexión alta y es capaz de transmitir hasta 20 *MB* en ambas direcciones. En comparación con tecnologías de bajo ancho de banda DakNet nos brinda un mayor rendimiento de datos.

Actualmente pueblos al norte de Camboya y la India se están beneficiando de los servicios de DakNet con buenos resultados.

## **2.2 Redes Tolerantes al Retardo**

Podemos considerar a Internet como el caso de éxito más sobresaliente en la interconexión de dispositivos de comunicación a lo largo y ancho de nuestro planeta. Este gran logro ha sido a base de la utilización de un grupo homogéneo de protocolos, llamados protocolos TCP/IP [12].

Cualquier dispositivo conectado a los cientos de miles de redes con acceso a Internet que desee transmitir información de un lugar a otro deben usar estos protocolos.

Cabe señalar que las necesidades de comunicación han venido cambiando a lo largo de los años y existen casos donde el uso de estos protocolos no logra cumplir con su objetivo debido a alguna intermitencia de comunicación entre el origen y el destino, no exista una ruta extremo a extremo ya que el destino del mensaje se encuentre en una zona que no tenga cobertura de comunicación o simplemente la comunicación no sea confiable, lo cual genera que una red basada en TCP/IP deje de funcionar correctamente y en el peor de los casos detenga el funcionamiento de toda la red.

La DTN es una arquitectura diseñada para proporcionar comunicaciones interoperables entre amplias gamas de redes con diferentes estándares de rendimiento, entornos y confiabilidad en caso de falla de hardware (red) y/o software (protocolo) [13].

Las DTN [14, 15], fueron diseñadas originalmente para soportar *Interplanetary Internet (IPN)* [16, 17], aunque con el nacimiento de nuevas necesidades de comunicación como conectividad intermitente, bajas velocidades de datos y retardos de propagación se tuvo que diseñar una nueva arquitectura de red. Las DTN han sido diseñadas para poder trabajar en situaciones donde los protocolos TCP/IP llegan a tener algunos problemas. Estas redes utilizan protocolos de enrutamiento para DTN [15] que son capaces de superar los retardos en la comunicación, las interrupciones y que también soportan la conectividad intermitente.

En el ejemplo clásico de comunicación entre una ciudad y una zona rural como nos ilustra la figura (4), podemos observar que se presentan tres métodos de interconexión que podrían ser empleados para establecer comunicación entre ellas; pueden utilizarse enlaces satelitales, módems tradicionales y un conmutador móvil instalado en un vehículo de transporte público, siendo este último método quien mejor representa la manera en que se comunican las DTN.

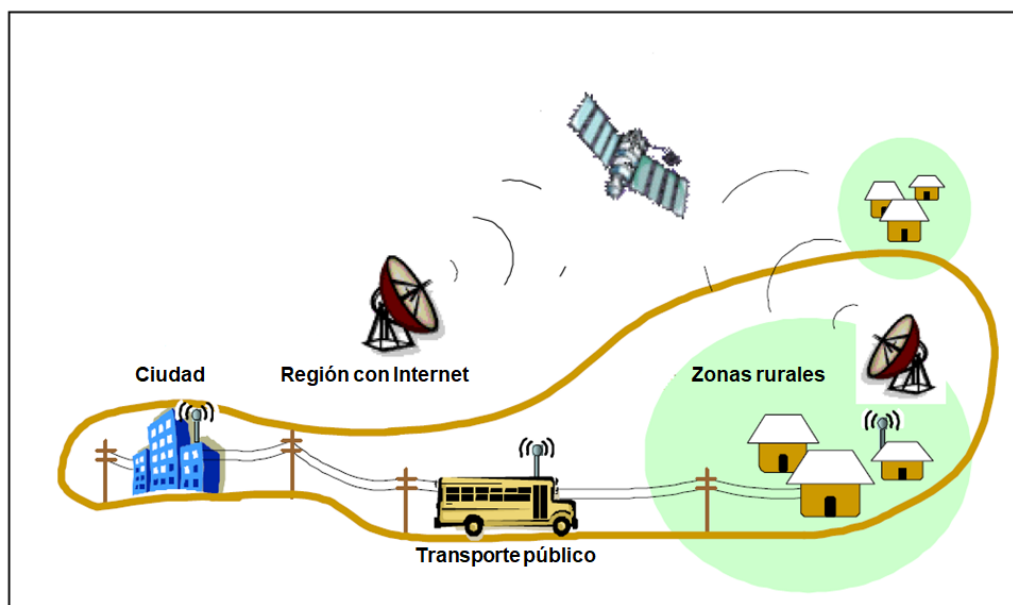


Figura 4: Ejemplo de una red DTN [18]

Algunas aplicaciones [15] que se le han dado a las DTN son:

- Comunicación digital en áreas rurales.
- Comunicaciones en entornos silvestres.
- Operaciones militares.
- Rescate en desastres.
- Comunicaciones de monitoreo del medio ambiente

## 2.3 LORA

Se han diseñado algunas soluciones basadas en *WiFi* para largas distancias [19, 20] las cuales han obtenido muy buenos resultados interconectando comunidades rurales y a pesar de ser una tecnología que se generó para redes de área local ha demostrado un excelente funcionamiento implementándose en soluciones de larga distancia.

Sin embargo, también se ha detectado que usar esta tecnología en largas distancias tiene dos limitaciones principales que son la necesidad de que exista una línea de vista entre las antenas de transmisión y la vulnerabilidad a la interferencia en la banda sin licencia. La primera de estas limitaciones puede ser superada usando torres o las elevaciones del terreno y de esta forma poder mantener la línea de vista entre los dispositivos transmisores. La segunda limitación llega a reflejarse menos en áreas rurales, pero en caso de ser necesario podría migrarse a una banda menos concurrida como lo es la de 5 GHz.

También deben ser considerados dos obstáculos al aplicar este tipo de soluciones: las limitaciones de tiempo y limitaciones energéticas. Una opción para solucionar el primer problema puede ser modificando el mecanismo de acceso a los medios, como lo realizó el grupo TIER en la Universidad de Berkeley [21], mientras que el problema energético puede ser solucionado utilizando antenas direccionales de alta ganancia.

Debido a las diferentes complicaciones que se han presentado al usar *WiFi* para largas distancias, en los últimos años ha surgido otra gama de tecnologías y protocolos para satisfacer los requisitos de comunicación *Low-Power Wide Area Networks (LPWAN)*. Estas tecnologías ofrecen cobertura de radio en un área (muy) grande a través de estaciones base, modulación, ciclos de trabajo, adaptación de

velocidades de transmisión, etc. Y de esta forma se ocasiona un notable bajo consumo de energía por parte de los dispositivos. Donde LoRa [22,23] es una de estas tecnologías LPWAN.

LoRa es un sistema de telecomunicaciones inalámbricas de largo alcance, baja potencia y baja tasa de bits, siendo una de las últimas tecnologías nacidas con altas expectativas, patentada en sus inicios por Semtech, que ahora se encuentra en continuo desarrollo por *Lora Alliance* [24] la cual está siendo promocionada como una solución de infraestructura para IoT.

Las nuevas generaciones de tecnologías inalámbricas son empujadas por una tendencia general hacia bandas de frecuencia cada vez mayores, a diferencia de las soluciones LPWAN que trabajan sobre la banda de 863 MHz a 870 MHz en Europa (también conocida como SDR860), que se mantienen reguladas por [25], y sobre la banda ISM de 902 MHz a 928 MHz en los Estados Unidos. Estas tecnologías pueden clasificarse en dos categorías según la modulación sobre la que operen:

- *Ultra Narrow Band (UNB)*: Opera sobre el ancho de banda de 25 kHz y utiliza canales de banda estrecha.
- Banda ancha: Emplea técnicas de acceso múltiple de amplio espectro, de esta forma comunica múltiples usuarios en un canal y opera sobre un ancho de banda del orden de 125 kHz o 250 kHz.

La arquitectura de un sistema LoRa está compuesta por tres elementos:

- Dispositivos finales LoRa: Son los sensores/actuadores conectados a través de la interfaz de radio LoRa con uno o más *Gateway* LoRa.
- *Gateway* LoRa: Es el concentrador o estación base que realiza la tarea de comunicar los dispositivos finales con el Servidor de Red LoRa, quien es el núcleo de la arquitectura de red.
- Servidor de Red LoRa: Es el dispositivo más importante ya que se encarga de controlar toda la red (control de admisión, administración de recursos de radio, seguridad, etc.).

La figura (5) nos ilustra los componentes principales de una arquitectura básica de un sistema LoRa, así como los medios de comunicación entre ellos y que además típicamente es una topología estrella.

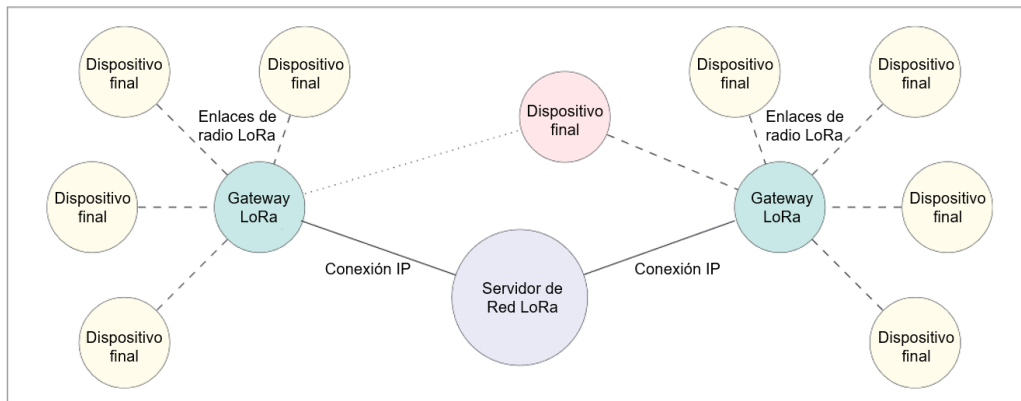


Figura 5: Arquitectura de un sistema LoRa [26]

Esta tecnología tiene como objetivo poder ser implementada en dispositivos donde el consumo energético es de suma importancia. LoRa frecuentemente puede hacer referencia a dos diferentes capas:

- La capa física, que fue desarrollada por Semtech emplea la técnica de modulación de radio *Chirp Spread Spectrum (CSS)* [27,28], soporta comunicaciones de baja potencia, largo alcance y bajo procesamiento. La velocidad de transmisión de datos puede alcanzar hasta  $50\text{ Kbps}$  empleando la agregación de canales y la carga útil de cada transmisión puede variar entre los 2-225 octetos.
- La capa *Medium Access Control (MAC)* del protocolo (*LoRaWAN*), que brinda un mecanismo de control de acceso al medio que admite que varios dispositivos finales establezcan comunicación con un *Gateway* empleando la modulación LoRa.

Como se menciona anteriormente Semtech posee la patente de la modulación LoRa, *LoRaWAN* es un estándar abierto que se continúa desarrollando por *Lora Alliance*.

La forma en que los *Gateway* retransmiten los mensajes entre los dispositivos finales y el Servidor de Red LoRa, de acuerdo con la arquitectura de protocolo tal y como se ve en la figura (6).

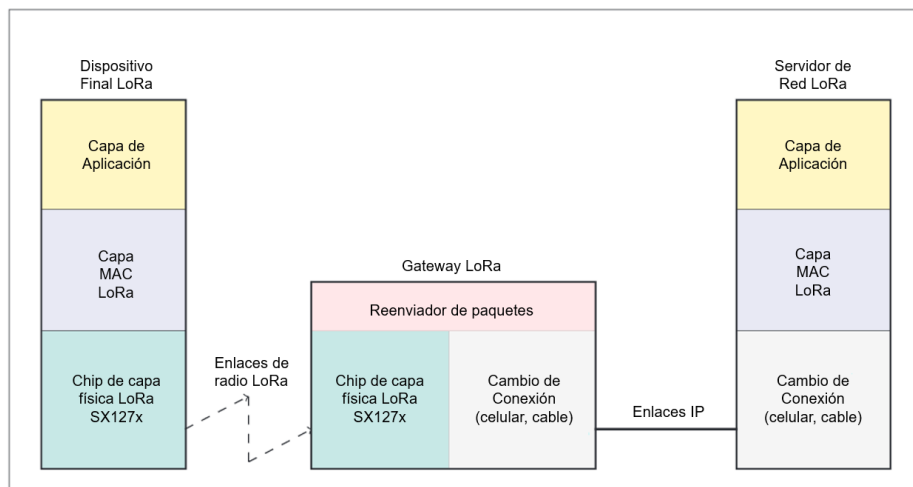


Figura 6: Arquitectura del Protocolo LoRa [26]

### 2.3.1 Capa Física LoRa

LoRa es una modulación CSS [27, 28], que utiliza *chirps* con una variación lineal de frecuencia en el tiempo para codificar información. Debido a que los pulsos *chirp* son emitidos de forma lineal, los desplazamientos de frecuencia entre el transmisor y el receptor son semejantes a los desplazamientos de temporización, el decodificador es capaz de eliminarlos fácilmente. El desplazamiento de frecuencia entre el receptor y el transmisor puede alcanzar el 20 % del ancho de banda sin afectar el rendimiento de la decodificación [29]. Lo cual se considera un beneficio ya que los cristales incrustados en los transmisores no requieren ser fabricados con extrema precisión lo que hace que su precio se reduzca de forma considerable. Los receptores LoRa pueden sincronizarse con los pitidos de frecuencia recibidos, ofreciendo una sensibilidad del orden de  $-130 \text{ dBm}$  [29, 30].

Dentro de las características que hacen a la señal LoRa apta para transmisiones de largo alcance y baja potencia podemos encontrar que la duración de la señal LoRa es más larga que las ráfagas típicas de interferencia generados por los sistemas *Frequency Hopping Spread Spectrum (FHSS)*, además que 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  $20 \text{ dB}$  y  $90 \text{ dB}$  [29, 30] respectivamente lo cual sobrepasa los esquemas de modulación tradicionales, como *Frequency-Shift Keying (FSK)*.

La modulación LoRa dispone de varios parámetros para personalizarse: *Spreading Factor (SF)*, *Code Rate (CR)* y *Bandwidth (BW)*. LoRa define de una forma no convencional al SF, como el logaritmo, en base 2, del número de *chirps* por unidad de señal. La modificación de los parámetros influye en la resistencia al ruido de interferencia, la facilidad de decodificación y en la tasa de bis efectiva de la modulación.

Para la modulación LoRa el parámetro más importante es BW. Una unidad de señal LoRa está compuesta por  $2^{SF}$  *chirps*, los cuales cubren toda la banda de frecuencia. La transmisión inicia con una serie de *chirps* ascendentes, que al llegar a la frecuencia máxima de la banda, esta se ajusta y el aumento de la frecuencia reinicia partiendo desde la frecuencia mínima. La figura (7) donde los datos fueron tomados de [31], nos muestra un ejemplo de una transmisión LoRa en la variación de frecuencia a través del tiempo; donde  $f_c$  es la frecuencia central del canal y  $BW$  es el ancho de banda. La posición de esta discontinuidad en la frecuencia es lo que codifica la información. Como hay  $2^{SF}$  *chirps* en una unidad de señal, una unidad de señal puede codificar SF bits de información efectivamente.

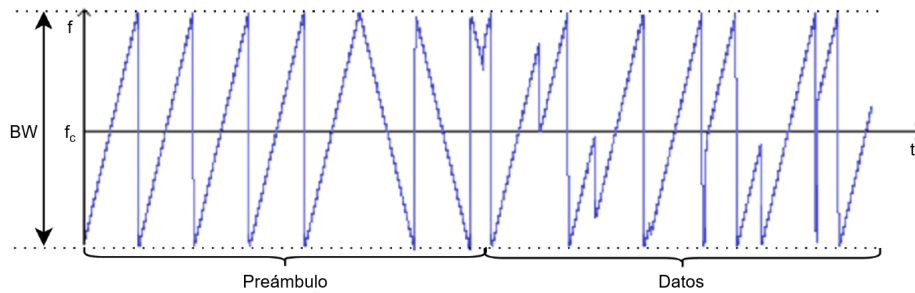


Figura 7: Variación de frecuencia a través del tiempo de una señal LoRa

En LoRa, la dependencia que existe entre la tasa de *chirp* y el ancho de banda: la frecuencia de *chirp* es igual al ancho de banda (un *chirp* por segundo por hercios de ancho de banda). Da como resultado algunas consecuencias sobre la modulación, por ejemplo, divide la velocidad de *bits* por dos, ya que se transmitirá un *bit* más en cada unidad de señal. También, la velocidad de unidad de señal y la velocidad de *bits* en un SF dado son proporcionales al ancho de banda de frecuencia, por lo cual, duplicar el ancho de banda también duplicaría la velocidad de transmisión. Esto se refleja en la ecuación (2.1), que vincula la duración de una unidad de señal ( $T_s$ ) con el BW y el SF.

$$T_s = \frac{2^{SF}}{BW} \quad (2.1)$$



LoRa también tiene incorporado un código de corrección de errores. El CR es igual a  $4/(4 + n)$ , con  $n \in \{1, 2, 3, 4\}$ . Tomando esto como referencia, además del hecho de que los *bits* de SF de información se transmiten por unidad de señal, utilizando la ecuación (2.2) podemos calcular la tasa de *bits* útil ( $R_b$ ).

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \quad (2.2)$$

### 2.3.2 LoRa MAC

Mientras que Semtech tiene la propiedad de la capa física de LoRa, el resto de la pila de protocolos se mantiene abierta y en continuo desarrollo. La capa MAC establecida por *LoRa Alliance* también es conocida como *LoRaWAN*, es un protocolo *ALOHA* según las especificaciones en [32] que principalmente es controlado por el Servidor de Red LoRa.

Normalmente las redes *LoRaWAN* se presentan mediante una topología donde los *Gateway* se encargan de retransmitir los mensajes entre el Servidor de Red y los dispositivos finales. Los paquetes que vienen desde cada dispositivo de la red son enrutados al servidor de aplicaciones asociado por el Servidor de Red.

El poder tener una tasa de datos adaptable es una de las características que distingue a LoRa MAC, ya que puede ser configurado desde el Servidor de Red y adaptar la velocidad de transmisión de un dispositivo final cambiando el índice SF, logrando de esta forma encontrar un equilibrio durante la transmisión entre la potencia del enlace y la eficiencia energética.

Otra característica que resalta son los fuertes mecanismos de seguridad los cuales son configurados mediante un procedimiento de activación inalámbrica, además de un procedimiento de activación por personalización (durante el tiempo de producción se establecen los parámetros en el dispositivo), estos mecanismos de seguridad incluyen una clave de aplicación y una clave de red.

El SF [33] es la relación que existe entre la velocidad de los *chips* y la tasa de unidades de señal. Un SF más alto incrementa la relación de *Signal to Noise Ratio* (SNR), afectando directamente el alcance y la sensibilidad, aunque se ve beneficiado el tiempo de aire del paquete. Para calcular el número de *chips* por unidades de señal se calcula como  $2^{SF}$ . Por ejemplo, al emplear un SF de 12 (SF12) se usan 4096 *chips*/unidades de señal. Con cada incremento de SF la velocidad de transmisión se ve reducida a la mitad, dando como resultado que se duplique la duración de la transmisión y, en última instancia, el consumo energético del

dispositivo. Los valores que puede tomar el SF van desde 6 hasta 12. SF6, con la transmisión de frecuencia más alta, es un caso especial y requiere operaciones especiales.

Todos los dispositivos *LoRaWAN* deben implementar al menos la funcionalidad de Clase A. Además, pueden implementar opciones denominadas Clase B o Clase C. En todos los casos, deben seguir siendo compatibles con la Clase A. Tal como nos muestra la figura (8).

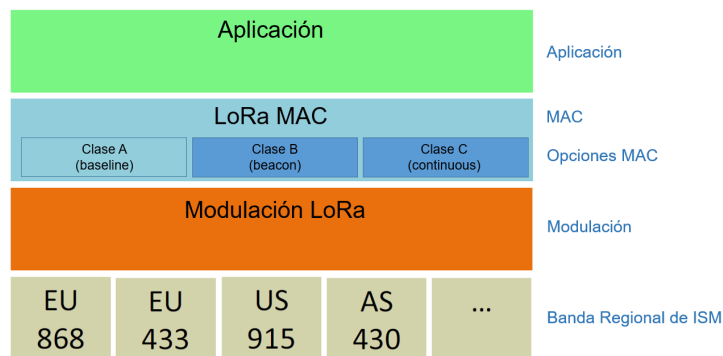


Figura 8: Clases LoRaWAN [32]

- Clase A, bidireccional: Los dispositivos de esta clase son capaces de programar un enlace de transmisión dependiendo de sus propias necesidades, con una pequeña variación aleatoria antes de la transmisión. Como estos dispositivos permiten comunicación bidireccional por cada transmisión le siguen dos ventanas cortas de enlace de recepción. La comunicación del enlace de recepción proveniente del servidor deberá esperar hasta que la siguiente transmisión ocurra. Estos dispositivos son los que tienen el consumo energético más bajo, aunque también son los que brindan menor flexibilidad con los enlaces de recepción.
- Clase B, bidireccional con ranuras de recepción programadas: Estos dispositivos son capaces de programar en horas establecidas ventanas adicionales de recepción. Por lo cual es necesaria una señalización sincronizada por parte del *Gateway* y de esta forma el servidor de red pueda saber en qué momento el dispositivo final se encuentra escuchando.
- Clase C, bidireccional con ranuras de recepción máximas: Los dispositivos finales de esta clase se caracterizan por tener ventanas de recepción casi continuas, lo que genera que sean los dispositivos que tienen el máximo consumo energético.

# Capítulo 3

## Análisis

El sistema que se ha desarrollado tiene como función esencial el poder comunicar mediante mensajería tipo SMS (uno a uno), como ya se ha mencionado la falta de medios de comunicación entre las zonas rurales y las ciudades es una de las razones por la que los pobladores se ven más afectados a nivel socio-económico. Además, la solución que se presenta también es capaz de enviar mensajes *broadcast* o también llamado *Bulletin Board System (BBS)*, el cual puede ser leído por los usuarios que se encuentren registrados (uno a muchos), los mensajes BBS pueden ser de gran utilidad ya que pueden portar datos sobre el estado del tiempo, sobre precios de mercado de cultivos e información sobre atención médica.

Para desarrollar este sistema de mensajería se hizo uso de dispositivos llamados *hubs*, los cuales son el núcleo de la estructura que se propone y que se puede observar en la figura (9). Estos dispositivos cuentan tanto con tecnología *WiFi* como con un transceptor que tiene la capacidad de emitir señal a largas distancias, los *hubs* crearán un punto de conectividad dentro de su rango de alcance al cual los usuarios finales podrán conectarse de manera inalámbrica para hacer uso de la herramienta. Una vez que los usuarios estén conectados tendrán que identificarse en la plataforma donde podrán enviar y recibir mensajes de otros usuarios.

Las aplicaciones que se le pueden dar a este sistema de mensajería son muy variadas ya que los mismos *hubs* pueden ser empleados para recopilar datos de otros dispositivos basados en LoRa, como, por ejemplo, sensores ambientales los cuales podrían estar separados por grandes distancias, los datos de estos sensores ser enviados al *hub* y este a su vez reenviarlos a los administradores de dichos sensores. Otra de las aplicaciones de gran valor y ayuda puede ser en la mitigación de desastres donde en ocasiones el servicio celular se ve interrumpido es ahí donde las comunicaciones de los dispositivos LoRa marcarían la diferencia.

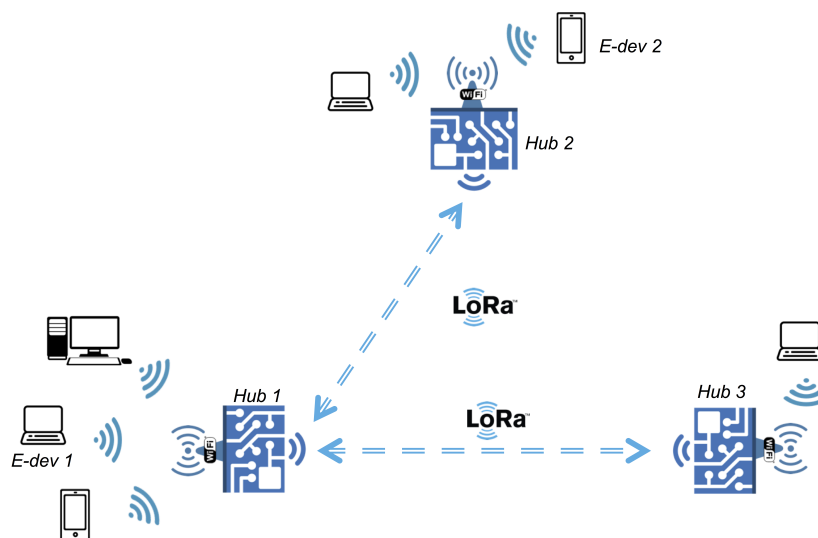


Figura 9: Estructura general de la plataforma de mensajería

### 3.1 Protocolo

Para el desarrollo de este sistema de mensajería se ha implementado un protocolo de unidifusión que se basa en un enfoque clásico *Automatic Repeat-reQuest (ARQ)* de parada y espera (*stop and wait*), tal como nos muestra la figura (10), con un valor dinámico y adaptativo para el retraso de la retransmisión, este protocolo garantiza que la información no se pierde debido al descarte de paquetes y que estos serán recibidos en el orden correcto.

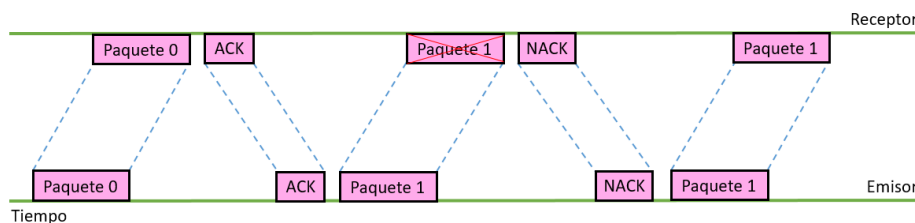


Figura 10: Esquema de protocolo *Stop and Wait*

A su vez también se encarga de gestionar la transmisión de mensajes largos mediante su partición en paquetes más pequeños, los cuales pueden ser almacenados en el *hub* destino.

### 3.2 Estructura del paquete, flujo y almacenamiento de datos

Los dispositivos centrales (*hubs*) mencionados en la estructura general crean un punto de conectividad dentro del área al cual se conectan los usuarios para enviar o recibir mensajes. Una vez que el usuario conecta de forma inalámbrica su dispositivo móvil podrá acceder a la plataforma donde debe autenticarse.

Debido a las limitadas características de hardware, los dispositivos no cuentan con la capacidad suficiente para el manejo de bases de datos, por tal motivo se ha optado por realizar el almacenamiento de los usuarios y mensajes mediante listas de diccionarios. De tal forma que el usuario se registra, ingresa el destinatario y captura el mensaje, una vez ocurre esto, el sistema tendrá que completar ciertos pasos para realizar una transmisión exitosa; el primer paso será almacenar al usuario emisor en el dispositivo local, el segundo paso será descubrir en que dispositivo de la red se encuentra el usuario receptor para lo cual deberá enviar un mensaje a todos los nodos que integren la red (mensaje *broadcast*) “preguntando” en cuál de ellos se encuentra registrado el usuario a quien va dirigido el mensaje, tal como se muestra en la figura (11), una vez que los demás nodos reciben el mensaje *broadcast* realizarán una búsqueda en la “Lista de usuarios” que tengan almacenadas y el nodo donde se encuentre el usuario receptor será el único que responda. A este proceso se le llama de descubrimiento y se realiza cada vez que se desea enviar un mensaje.

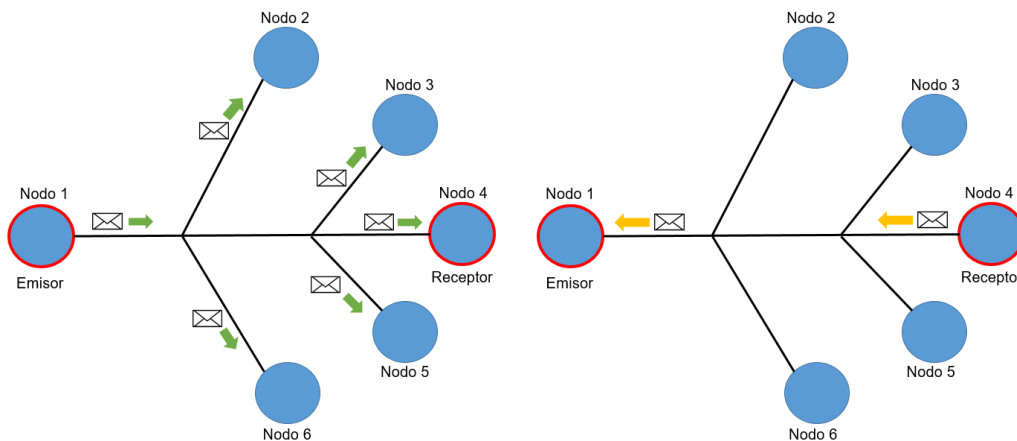


Figura 11: Proceso de descubrimiento

Una vez que el nodo emisor “conoce” al nodo destino el mensaje se segmenta en paquetes pequeños para después transmitirlos usando el protocolo descrito con anterioridad. El proceso de segmentación de paquetes se realiza debido a que el tamaño de los paquetes ya ha sido definido (230 *bytes*) además de que será necesario agregar una cabecera a cada paquete del mensaje para poder saber que nodo lo ha enviado, cuál será su nodo destino, que número de paquete es, si es o no el paquete final del mensaje, entre otras cosas. Al tener también la cabecera un tamaño específico el paquete estará compuesto por la cabecera y el resto del espacio del paquete serán para los segmentos del mensaje (*payload*), tal como lo podemos observar en la figura (12).

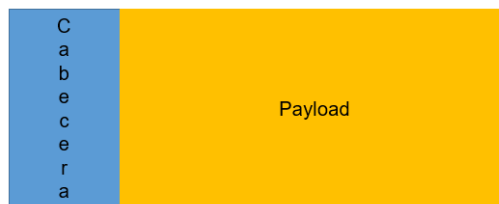


Figura 12: Estructura del paquete

Entrando un poco más a detalle en la cabecera del paquete cabe resaltar que su tamaño total será de 24 *bytes* del total de 230 *bytes* que puede contener el paquete, al ser los dispositivos empleados del mismo fabricante las direcciones MAC son muy similares entre ellas, por tal motivo se ha decidido emplear únicamente los últimos 8 *bytes* de la dirección MAC de cada origen y destino, de tal manera que serán 8 *bytes* para la dirección MAC del dispositivo fuente, 8 *bytes* para la dirección MAC del dispositivo destino, 2 *bytes* serán asignados para identificar el número de secuencia del paquete (*Seqnum*), 2 más para el número de ACK (*Acknum*), 1 *byte* será quien identifique si es o no el último paquete (*flags*) y al final el *checksum* será de 3 *bytes*. Podemos observar la estructura de la cabecera en la figura (13).

Dirección origen (8 bytes)			
Dirección destino (8 bytes)			
Seqnum (2B)	Acknum (2B)	Flags	Checksum (3B)

Figura 13: Estructura de la cabecera del paquete

# Capítulo 4

## Diseño y Metodología de Desarrollo

La solución que se ha desarrollado es capaz de enviar y recibir información a grandes distancias. Para ello se desarrolló en MicroPython un programa que se encarga de almacenar usuarios, mensajes, tratar los datos a ser enviados y un protocolo que se encarga de transferirlos entre la red de dispositivos LoPy, los cuales son dispositivos IoT de bajo costo y bajo consumo energético. Que a su vez cuenta con tres transceptores (LoRa, WiFi, Bluetooth).

Para hacer uso de este sistema de mensajería el usuario origen podrá establecer comunicación de manera inalámbrica al punto de conexión generado por la LoPy más cercana a él una vez conectado podrá tener acceso a una página web donde deberá registrarse, para después poder introducir el nombre del usuario destino, así como el mensaje que pretende enviar, la LoPy tendrá que realizar el procedimiento de descubrimiento de usuarios enviando un mensaje broadcast a toda la red de dispositivos, el cual será respondido únicamente por la LoPy que tenga en su lista de usuarios registrados al usuario a quien va dirigido el mensaje, el paso siguiente será segmentar el mensaje en caso de que no pueda ser enviado en un solo paquete, agregará las cabeceras y transmitirá las veces que sean necesarias hasta enviar el mensaje completo. El procedimiento debe ser repetido para cada mensaje que se desee enviar.

### 4.1 Dispositivos

Los diferentes dispositivos empleados para el desarrollo del sistema de mensajería son parte de la variedad de soluciones de bajo coste, bajo consumo energético y que además tienen integradas varias tecnologías inalámbricas que provee la empresa Pycom© [34], los cuales se detallarán en las subsecciones siguientes.

### 4.1.1 Lopy

Es un microcontrolador que se encuentra habilitado para ser programado bajo MicroPython, basado en el último conjunto de chips ESP32 de Espressif. Está equipado con un procesador dual y tres transceptores (LoRa, WiFi, Bluetooth) además de dos conectores de antena: uno para la banda de 868 MHz utilizada por LoRa y otro para la banda de 2,4 GHz utilizada por *WiFi* y Bluetooth. Lo que nos da la oportunidad de poder instalar la antena más adecuada para la aplicación. Por ejemplo, las antenas direccionales de alta ganancia se pueden utilizar para conectar un pueblo rural con una ciudad que podría estar a una distancia muy larga, utilizando muy poca potencia aprovechando las características de espectro ensanchado de la modulación LoRa. Tiene una memoria *RAM* de 512KB y permite un flash externo de 4MB. Tiene aceleración de punto flotante de hardware y se puede programar en *multithreading* de Python, más especificaciones técnicas pueden ser consultadas en [35]. Podemos observar el dispositivo en la figura (14).

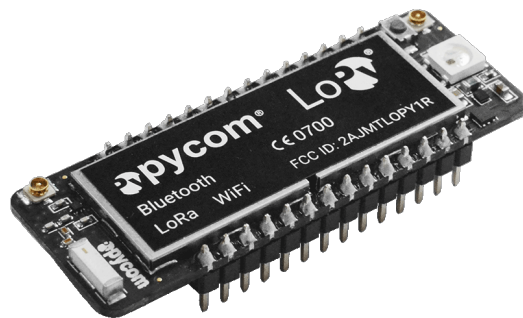
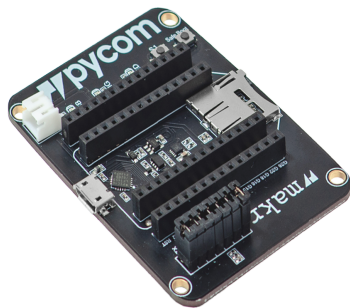


Figura 14: Microcontrolador LoPy

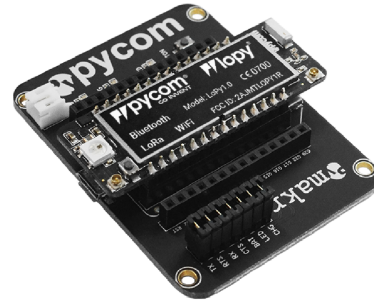
### 4.1.2 *Expansion Board 2.0*

Dentro del catálogo de tarjetas que maneja la marca Pycom©, podemos encontrar la tarjeta Pysense y la *Expansion Board 2.0* con las cuales se inició el desarrollo de este proyecto, desafortunadamente la tarjeta Pysense llegó a dar problemas de reconocimiento entre ella y los equipos de cómputo ya que había que conectar y desconectar del puerto *usb* hasta que era reconocida, aun se busca la manera de poder integrarla ya que posee una amplia variedad de sensores (acelerómetro, humedad, presión, temperatura, luz) a diferencia de la tarjeta *Expansion Board 2.0* con la cual se desarrolló este sistema de mensajería y es la que aparece en la figura (15). En la figura (15a) podemos observar únicamente la tarjeta *Expansion Board 2.0* y en la figura (15b) aparece el microcontrolador LoPy ya montado en la tarjeta.





(a) Expansion Board 2.0



(b) Expansion Board 2.0 & LoPy

Figura 15: Dispositivo Pycom

Algunas de las características de esta tarjeta son, que puede ser alimentada mediante una batería LiPo o vía usb, posee un convertidor usb a serie FT234XD, cuenta con una ranura para tarjeta MicroSD, también es capaz de conectarse fácilmente con los dispositivos WiPy, SiPy, LoPy y FiPy, posee LED de alimentación y de estado de carga, además de que puede ser reiniciada de manera local o remota vía Telnet, más especificaciones técnicas pueden consultarse en [36].

### 4.1.3 Antena LoRa & Sigfox

Esta antena también es un producto distribuido por la empresa Pycom©, la cual puede trabajar en la banda de 868 MHz y en 915 MHz, es una antena capaz de trabajar con LoRa y con Sigfox además de ser compatible con la mayoría de los dispositivos LoPy, SiPy, LoPy4 y FiPy. Más detalles de esta antena pueden ser consultados en [37] y en la figura (16) podemos observarla.

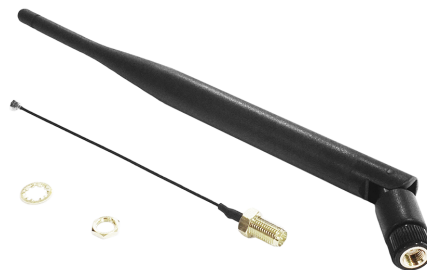


Figura 16: Antena LoRa & Sigfox

## 4.2 Servidor Web

El dispositivo contendrá un servidor web, el cual brindará una página muy sencilla debido a las características de estos dispositivos, la cual funcionara como plataforma gráfica donde los usuarios deberán proporcionar un ID como podemos observar en la figura (17), este paso se considera obligatorio para el sistema ya que no permite el envío de mensajes sin conocer el usuario origen, por lo tanto, en caso de no introducir el ID del usuario origen se mostrará una pantalla de advertencia como se muestra en la figura (18), una vez que el proceso de registro se haya completado, el ID del usuario será guardado en la lista de usuarios dentro de la LoPy para después dar paso a la pantalla de envío y recepción que se muestra en la figura (19), donde podrá seleccionarse si se desea enviar mensaje a un usuario en específico o a todos los usuarios de la red. Esta pantalla también le da la oportunidad al usuario que ha ingresado de verificar si existen mensajes dirigidos a él y con solo presionar el respectivo botón podrá verlos.

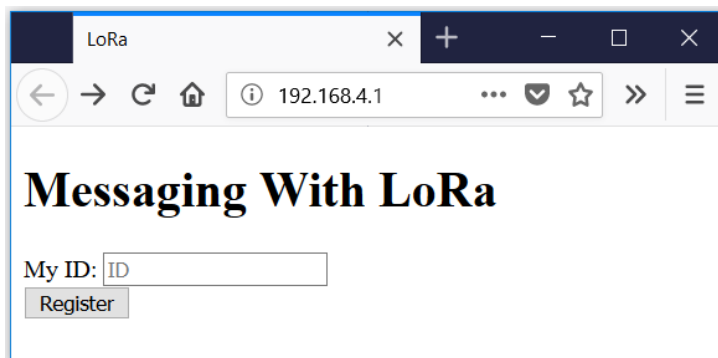


Figura 17: Pantalla de registro de usuario

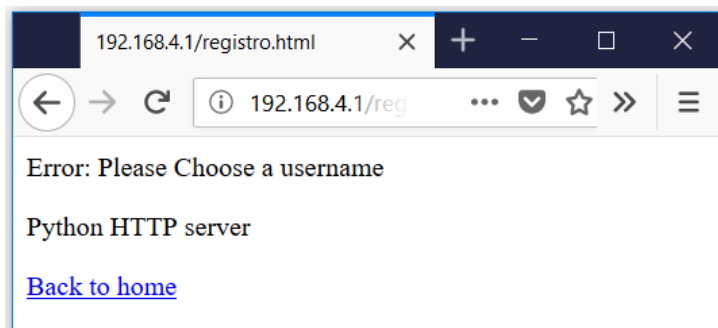


Figura 18: Pantalla de advertencia

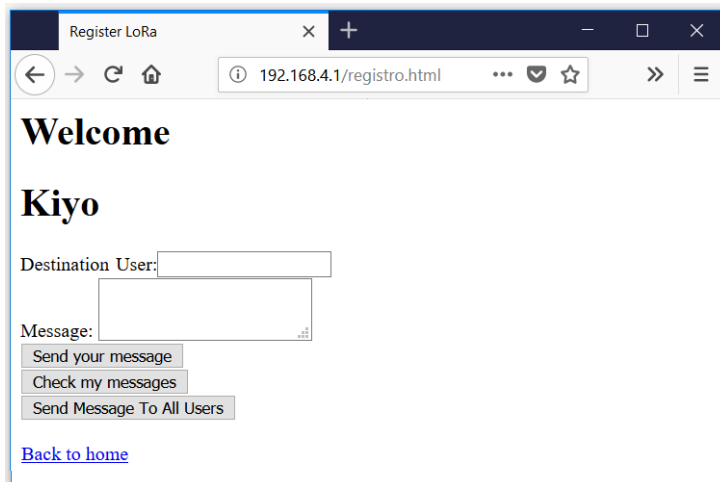


Figura 19: Pantalla de envío y recepción de mensajes

La página web que se utiliza para el envío de mensajes es bastante sencilla y fácil de usar, tal y como se describe anteriormente el usuario podrá decidir entre enviar un mensaje a un solo usuario o enviar un mensaje a todos los usuarios de la red, en caso de elegir el envío a un solo usuario deberá introducir el ID del usuario destino y escribir el mensaje a enviar, como se puede observar en la figura (20), en caso de que elija enviar un mensaje a todos los usuarios únicamente deberá escribir el mensaje ya que el sistema se encargara de estructurar el paquete a transmitir. Una vez que el botón de envío sea presionado el programa realizara el proceso de descubrimiento de usuario, empaquetado y envío del mensaje, una vez concluido el proceso de transmisión el usuario podrá ver la pantalla de transmisión exitosa que se muestra en la figura (21).

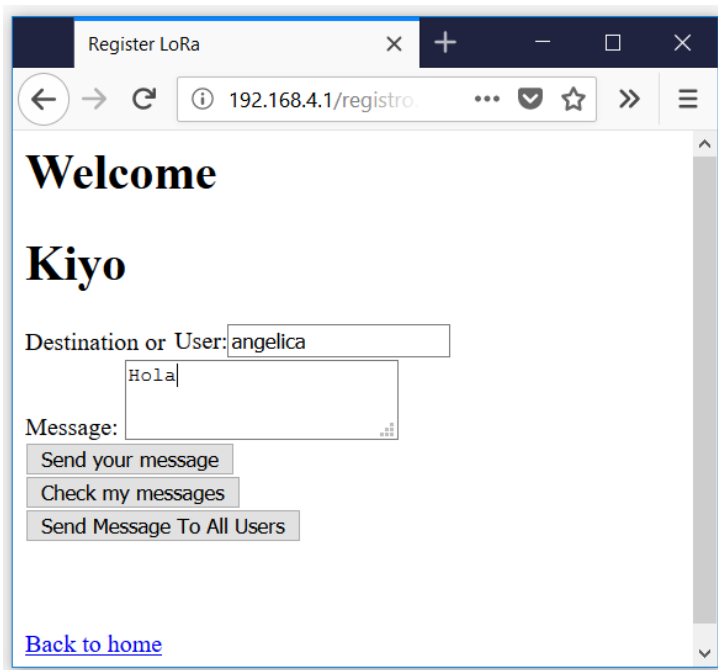


Figura 20: Ejemplo de envío de mensaje a un solo usuario

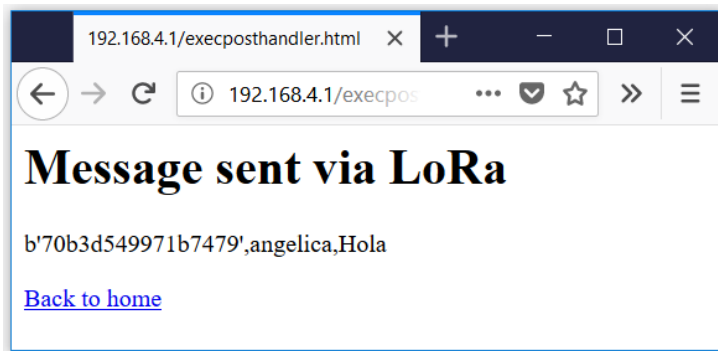


Figura 21: Pantalla de transmisión exitosa

En caso de que el usuario que se ha registrado deseara verificar si tiene mensajes se debe presionar el botón “*Check my messages*”, el cual desplegara la pantalla que se muestra en la figura (22) donde se podrán observar dos secciones; la primera mostrará los mensajes que son únicamente para este usuario, podrá ver que usuario se lo ha enviado, así como el mensaje. En la otra sección podrá observar los mensajes broadcast que se hayan enviado a toda la red.



Figura 22: Pantalla de mensajes recibidos

Cabe destacar que la página web funciona de forma correcta en la mayoría de los exploradores (Mozilla, Chrome, Opera), aunque con el explorador *Safari* presenta algunos problemas, debido a las características del programa al parecer no es compatible con este explorador, por lo que se recomienda usar alguno de los mencionados anteriormente.

### 4.3 Funciones principales

El sistema de mensajería hace uso de varias funciones para que la transmisión de datos se haga de forma correcta, en esta sección se detallan las funciones más sobresalientes, por ejemplo, la forma en que el usuario interactúa con el sistema, el proceso de transmisión y recepción de datos, como son tratadas las instrucciones *post* y *get* por el servidor *web*, como son creados y almacenados los usuarios y mensajes, etc.

La figura (23) muestra de forma más detallada como el usuario va introduciendo los datos al sistema cuando desea enviar o recibir mensajes y como se le van mostrando las diferentes pantallas, así como el proceso que existe entre las pantallas y el almacenamiento de esos datos.

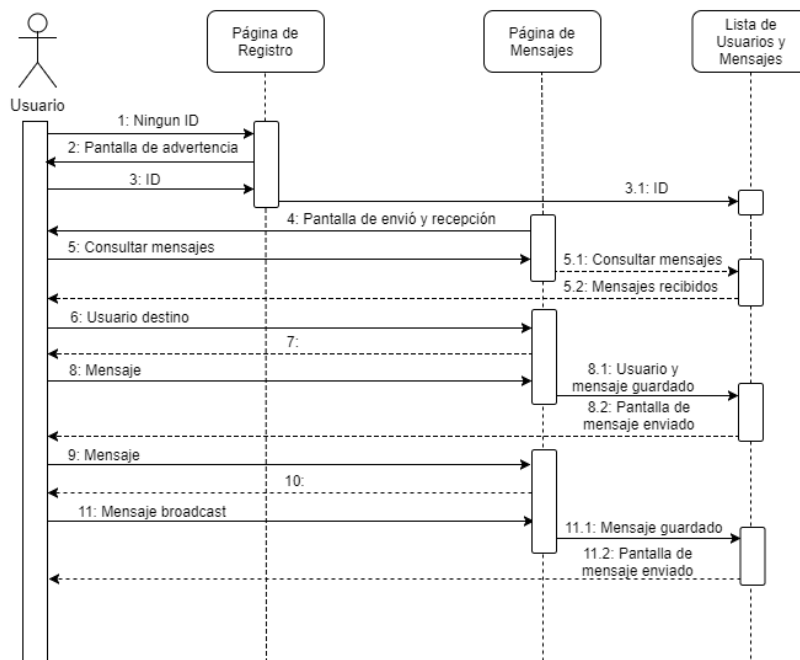


Figura 23: Proceso de cambio de pantallas y almacenamiento de datos

El sistema de mensajería funciona en base a dos supuestos, la transmisión y la recepción de mensajes. El diagrama de la figura (24) representa el proceso por el cual deberá pasar un mensaje antes de ser enviado, en este caso existen dos posibilidades de transmisión las cuales son enviar mensaje a un solo usuario o a todos los usuarios de la red, si el usuario decide enviar un mensaje a todos los usuarios de la red el paquete a transmitir se estructura por el nombre de usuario destino el cual para este caso será *broadcast* y se agrega el mensaje. En caso contrario si el usuario decide enviar un mensaje a un solo usuario, el sistema realiza el proceso de reconocimiento donde se envía un mensaje en la red con dirección destino "FFFFFFF" lo cual significa que será recibido por todos los dispositivos de la red "*preguntando*" si en alguno de ellos se encuentra registrado el usuario destino, el mensaje de búsqueda del 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; en caso de que la respuesta contenga una dirección MAC, esta será del dispositivo donde se encuentra registrado el usuario destino, una vez teniendo la dirección MAC de la LoPy se estructura el paquete a ser enviado donde se especifica el emisor, el mensaje, el receptor y la dirección del dispositivo destino. Una vez que se conoce la dirección MAC del dispositivo destino únicamente se realiza el envío del mensaje. Durante la transmisión de mensajes por LoRa el dispositivo ilumina un led en color azul.

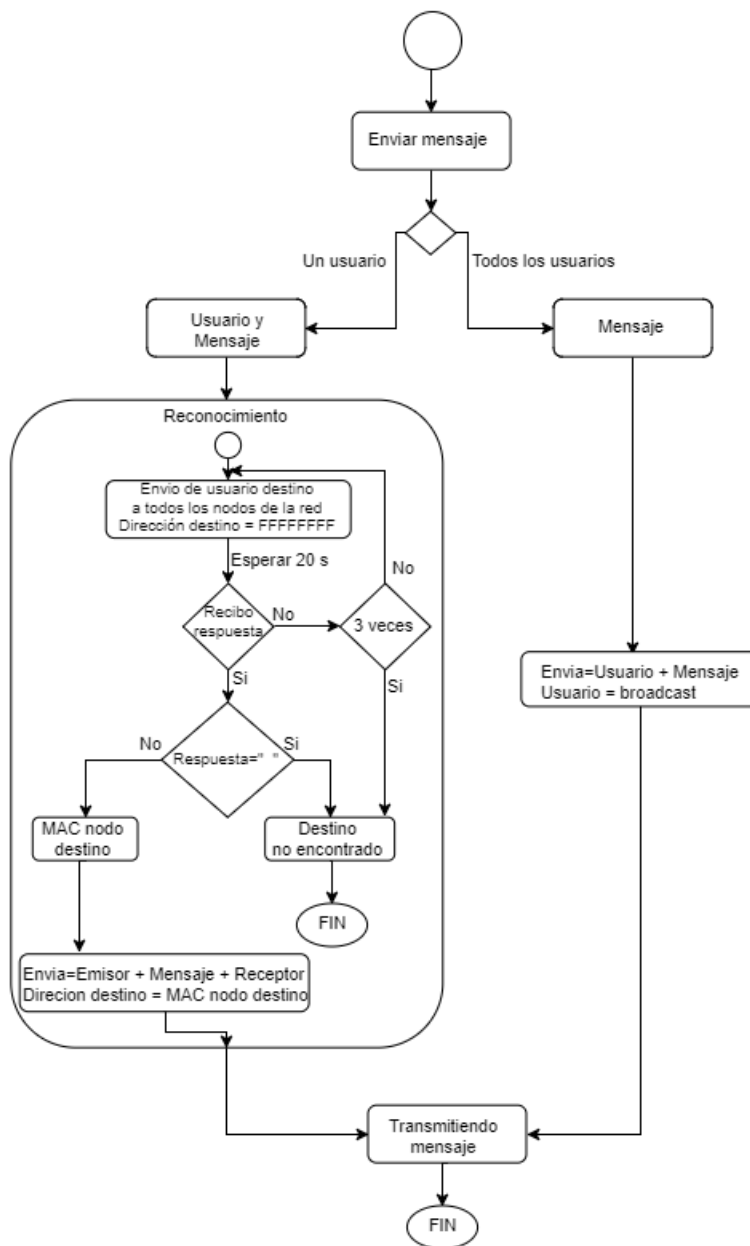


Figura 24: Diagrama de proceso de transmisión

En la figura (25) se observa a nivel consola el proceso de envío de mensaje *broadcast* que sirve para el descubrimiento de usuario, así como la respuesta con la dirección MAC del dispositivo donde se encuentra registrado el usuario destino, una vez que se conoce la dirección MAC se hace la transmisión de mensajes.

```

DEBUG: Searching: angelica → Usuario destino
DEBUG: Searching: 0
RCV_ADDR b'FFFFFFFFFFFFFFFF'
New RCV_ADDR b'FFFFFFF' → Dirección destino
Checksum
b'b7c'
sending 1st: s_p: b'971b7479', d_p: b'FFFFFFF', seqn: 0, ackn: 0, ack: False, f
in: True, check: b'b7c'
SND_ADDR b'971b7479'
received ack: s_p: b'96fad316', d_p: b'971b7479', seqn: 0, ackn: 0, ack: True, f
in: True, check: b'855'
b'96fad316'
RETURNING tsend
DEBUG: From Swlp My Address: b'971b7479' → Respuesta al mensaje
From Swlp receiving source Address: b'96fad316'
received 1st packet: s_p: b'96fad316', d_p: b'971b7479', seqn: 0, ackn: 0, ack:
False, fin: True, check: b'76f', cont: b'70b3d54996fad316'
Checksum
b'76f' → Respuesta con la dirección MAC del dispositivo
True
Packet Valid
Checksum
b'855'
sent 1st ACK: s_p: b'971b7479', d_p: b'96fad316', seqn: 0, ackn: 0, ack: True, f
in: True, check: b'855'
DEBUG: Message: b'70b3d54996fad316'
DEBUG: Retransmissions 0
DEBUG: dest lora address: b'b3d54996fad316' → Dirección destino
DEBUG: Search Destination time: 1334.9999189376 mseconds.
DEBUG: Payload to be sent: Kiyó,Saludos,angelica → Emisor + Mensaje + Receptor

```

Figura 25: Ejemplo de transmisión visto desde la consola

Para el supuesto de recepción el proceso es un poco más amplio, debido a que existen dos funciones que pueden ser ejecutadas, pueden ser una función LoRa o una función *HTTP* y dependiendo de ellas se realizan procesos diferentes. El dispositivo se mantendrá a la espera de que el usuario realice alguna acción que dispare alguna de estas dos funciones, en caso de ser una función LoRa se iluminará un led en color amarillo, el programa verifica si se ha recibido una dirección *broadcast* (FFFFFFFF), lo que le indica que se está buscando a un usuario en la red, por lo cual procederá a verificar en su lista de usuarios registrados si existe dicho 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 “aprenda” quien será el dispositivo destino con el cual mantendrá comunicación hasta que el mensaje completo sea recibido. Si la función que se activa es *HTTP*, el led del dispositivo se ilumina en color verde, el programa verifica que sean paquetes *HTTP* con estructura que el reconoce, de no ser así se desecha la petición; en caso de que reconozca la estructura de los paquetes se procede a descomponerlo y a verificar que tipo de petición se está solicitando. Si es una petición tipo “*GET*” se procede a mostrar la página web contenida en el archivo *index.html* que se observa en la figura (17); si la petición es de tipo “*POST*” significa que el usuario ha presionado alguno de los botones contenidos en la página donde podrá realizar alguna de las siguientes acciones:



- Registro: Como se puede observar en la figura (17) la primera pantalla con la que tiene interacción el usuario lo obliga a proporcionar un ID, tanto para poder enviar como para poder recibir mensajes el cual será obligatorio en cualquiera de los casos. También como ya se ha mencionado los dispositivos carecen de las características suficientes para poder trabajar con bases de datos, de tal forma que la solución que se propone es usando listas de diccionarios y el proceso de registro de usuario utiliza una de estas, a la que hemos llamado “Lista de usuarios”. Una vez que el usuario introduce su ID y presiona el botón “*Register*”, el programa accede a la lista y agrega el nombre del usuario en el índice que corresponda.
- Enviar mensaje: Una vez que se haya introducido el ID del usuario destino y el mensaje a enviarle, el usuario debe presionar el botón “*Send your messages*”, el cual realiza el procedimiento de descubrimiento de usuario que ya hemos explicado anteriormente, una vez que “sabe” cuál será el dispositivo destino, este recibe el paquete, buscará cual es el índice del usuario destino en su “Lista de usuarios” y guardará en la “Lista de mensajes” en el mismo índice el mensaje y el emisor del mismo.
- Revisar mensajes: Este proceso es el que consulta todas las listas del sistema, ya que después de que el usuario se haya registrado y presione el botón “*Check my messages*”, se realiza una búsqueda en la “Lista de usuarios” para obtener el índice donde se encuentra almacenado este ID, después se obtienen de la “Lista de mensajes” lo que está almacenado en ese índice, como paso final se consulta la “Lista de *broadcast*” y se obtienen todos los mensajes que existan.
- Mensaje *broadcast*: Cuando el usuario presiona el botón “*Send message to all users*” el proceso para enviar a todos los usuarios hace uso de la misma función que se usa para el envío a un solo usuario haciendo algunas pequeñas modificaciones, ya que este tipo de mensajes no llevará un usuario destino por lo cual se agrega un ID con valor “*broadcast*”, junto con una bandera que identifica el tipo de mensaje, cuando esta bandera lleva el valor de “1” significa que es un mensaje broadcast y cuando lleva el valor de “0” será para un usuario en específico.

En caso de que la petición recibida no sea de ninguno de los dos tipos anteriormente descrito se desecha la petición. La figura (26) nos muestra de manera gráfica el proceso de recepción.

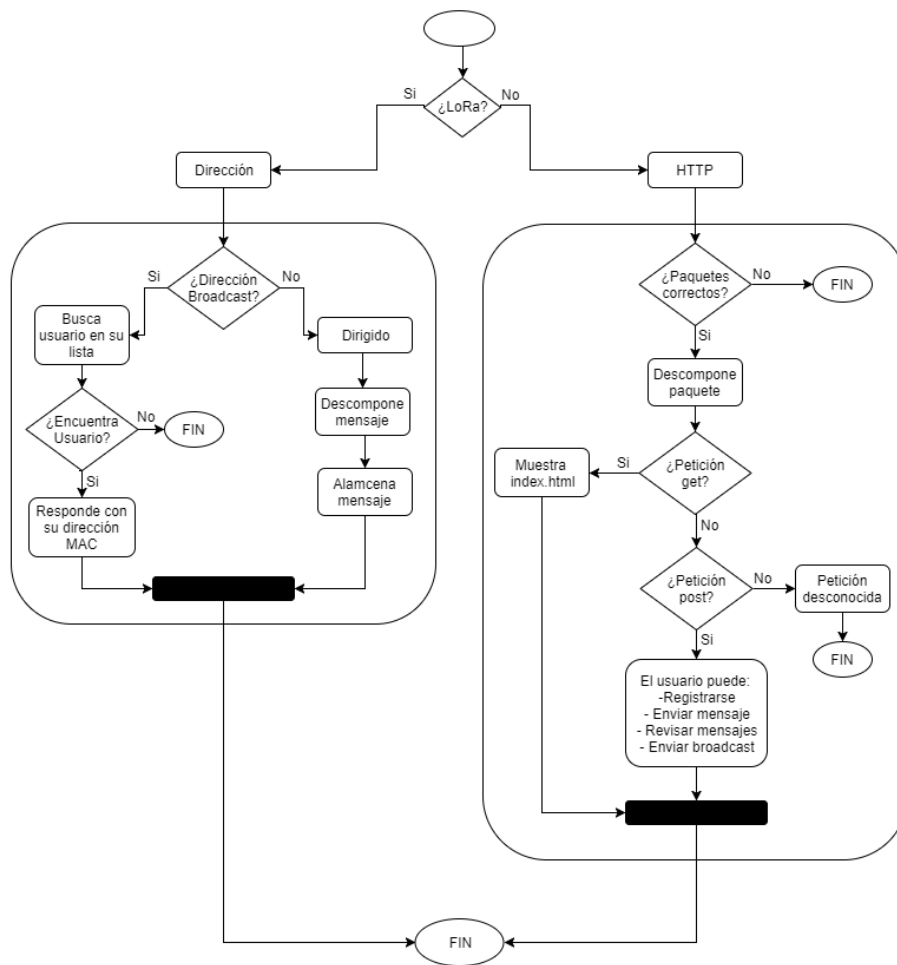


Figura 26: Diagrama de proceso de recepción

En la figura (27), podemos observar el proceso de reconocimiento del dispositivo que tenga al usuario destino visto desde la consola, donde el dispositivo origen envía la dirección “FFFFFFFF” para que los demás dispositivos busquen en su “Lista de usuarios”, en caso de que ese usuario se encuentre registrado, se devolverá un mensaje con la dirección MAC del dispositivo que lo contenga.

```

received 1st packet: s_p: b'96fad316', d_p: b'FFFFFFF', seqn: 0, ackn: 0, ack:
False, fin: True, check: b'008', cont: b"b'70b3d54996fad316',Kiy0"
checksum de la funcion checksum
b'008'
checksum completo
b'1c6204b2a5a2df9b37d8e4518b64f46345a3013a6236c4d2a1a64cff6698a008'
True
Packet Valid
checksum de la funcion checksum
b'855'
checksum completo
b'e3b0c44298fclc149afb4c8996fb92427ae41e4649b934ca495991b7852b855'
sent 1st ACK: s_p: b'971b7479', d_p: b'96fad316', seqn: 0, ackn: 0, ack: True,
fin: True, check: b'855'
DEBUG: Content in reception LoRa b"b'70b3d54996fad316',Kiy0"
DEBUG: Source Address in LoRaRec b'FFFFFFF'
P Lora: b"b'70b3d54996fad316'
DEBUG: User Kiy0 → Usuario destino → Dirección broadcast
DEBUG: User Database: ['Kiy0']
consulta → Lista de usuarios
DEBUG: Flag 1
DEBUG: Lora Address 70b3d54996fad316'
RCV_ADDR 70b3d54996fad316' → Dirección MAC del dispositivo
new RCV_ADDR 96fad316' → Solo se usan 8 bits
checksum de la funcion checksum
b'487'
checksum completo
b'bafc899077f764c4ae41d5970dc3e43e6e4cab91083fbf4a3df308c9298d5487'
sending 1st: s_p: b'971b7479', d_p: b'96fad316', seqn: 0, ackn: 0, ack: False,
fin: True, check: b'487'
SEND_ADDR b'971b7479'
received ack: s_p: b'96fad316', d_p: b'971b7479', seqn: 0, ackn: 0, ack: True,
fin: True, check: b'855'
b'96fad316'
RETURNING tsend
DEBUG: done reading data from the LORA channel using swlpv3: b"b'70b3d54996fad3
16',Kiy0"

```

Figura 27: Proceso de reconocimiento visto desde la consola

Una vez que el dispositivo origen “conoce” en que dispositivo se encuentra el usuario destino, inicia el envío de paquetes. El dispositivo destino recibe el paquete y procede a descomponerlo, para de esta forma obtener el mensaje donde vendrá el emisor, el mensaje y el receptor; una vez que se tiene esta información inicia el proceso de guardado, se busca en la “Lista de usuarios” el índice en el que se encuentra almacenado el usuario destino, después de obtenerlo se guarda el emisor y el mensaje en ese mismo índice. La figura (28) nos muestra este procedimiento visto desde la consola.

```

DEBUG: From Swlp My Address: b'971b7479'
Content received b'96fad316971b7479\x00\x00\x00\x001531angelica,Saludos,Kiyo'
check rcv checksum                                     → Paquete recibido, listo para descomponerse
b'531'
From Swlp receiving source Address: b'96fad316'        → Dirección origen
received 1st packet: s_p: b'96fad316', d_p: b'971b7479', seqn: 0, ackn: 0, ack:
False, fin: True, check: b'531', cont: b'angelica,Saludos,Kiyo'
Checksum de la funcion checksum                         → Contenido del paquete
b'531'
checksum completo
b'a19668d81d409433f56c51f5f63d852f79e730342275a18fc087d13827530531'
True
Packet Valid
Checksum de la funcion checksum
b'855'
checksum completo
b'e3b0c44298fclc149afb4c8996fb92427ae41e4649b934ca495991b7852b855'
sent 1st ACK: s_p: b'971b7479', d_p: b'96fad316', seqn: 0, ackn: 0, ack: True,
fin: True, check: b'855'
DEBUG: Content in reception LoRa b'angelica,Saludos,Kiyo'
DEBUG: Source Address in LoRaRec b'971b7479'
DEBUG: message in server b'angelica,Saludos,Kiyo'      → Emisor + Mensaje + Receptor
Sender: 'angelica'
Message: Saludos
User: Kiyo'
Saving Message
DEBUG: Message Database: [{}]                          → Lista de mensajes antes almacenar
DEBUG: User Database: ['Kiyo']                        → Lista de usuarios, receptor existe en la lista
DEBUG: Position: 0
DEBUG: New Users Database: ['Kiyo']
DEBUG: New Message Database: [{'0Emisor ': 'angelica', '0Mensaje ': 'Saludos'}]
]
DEBUG: Number of Message: 1
DEBUG: done reading data from the LORA channel using swlpv3: b'angelica,Saludos
,Kiyo'
The End

```

Figura 28: Proceso de recepción y guardado de mensaje visto desde la consola

El servidor se mantiene a la escucha de las peticiones que pueda hacer el usuario, la figura (29) nos presenta la fracción de código encargada del control de los *sockets*, de esta manera evitamos tener que usar instrucciones más complejas como “hilos”.

```

def checking_connection(self,s_left,addr):
    data=""
    print("Got connection from:", addr)
    data = s_left.recv(1024) #receive data from client
    if DEBUG_MODE: print("DEBUG: Data received:",data)
    if(data==b""): —————> Función para descartar peticiones HTTP nulas
        print("Null Method, Discarding")
    else:
        treq = bytes.decode(data)
        self._wait_for_connections(s_left,addr,treq)

def conexion(self): #Function in charge of the coordination of sockets
    ANY_ADDR = b'FFFFFFFFFFFFFFFF'
    while True:
        s_read, _, _ = select.select([self.socket, self.s_right], [], [])
        for a in s_read:
            if a == self.socket: —————> Función para peticiones HTTP
                # reading data from the HTTP channel
                print("DEBUG: in connections handler: reading data from the HTTP channel")
                s_left, addr = self.socket.accept()
                self.checking_connection(s_left,addr)
            elif a == self.s_right: —————> Función para peticiones LoRa
                # reading data from the LORA channel using swlpv3
                print("DEBUG: reading data from the LORA channel using swlpv3")
                ufun.flash_led_to(YELLOW) —————> Al ser una petición LoRa, se ilumina el led en color amarillo
                data, sender = swlp.trecvcontrol(self.s_right, my_lora_address, ANY_ADDR)
                LoRaRec(data,self.s_right,sender)
                if DEBUG_MODE: print("DEBUG: done reading data from the LORA channel using swlpv3:",data)
                if DEBUG_MODE: print("The End")
                ufun.flash_led_to(OFF)

```

Figura 29: Función para el control de *sockets*

# Capítulo 5

## Resultados

Se realizaron una serie de pruebas variando el tamaño del paquete a transmitir, la distancia entre nodos y el número de mensajes que se transmiten a la vez, de donde pudimos observar el comportamiento que tienen los dispositivos al realizar transmisión y recepción de mensajes, así como el tiempo que se tarda en “descubrir” en que nodo se encuentra un usuario, entre otras cosas que se describen a continuación.

### 5.1 Nodos

Durante la realización de las pruebas el nodo de transmisión se compuso por un dispositivo LoPy con una antena omnidireccional conectado vía *usb* a un ordenador portátil, mientras que el nodo receptor estaba compuesto por el dispositivo LoPy con su respectiva antena conectados a una batería externa como podemos ver en la figura (30), esto para mayor portabilidad.

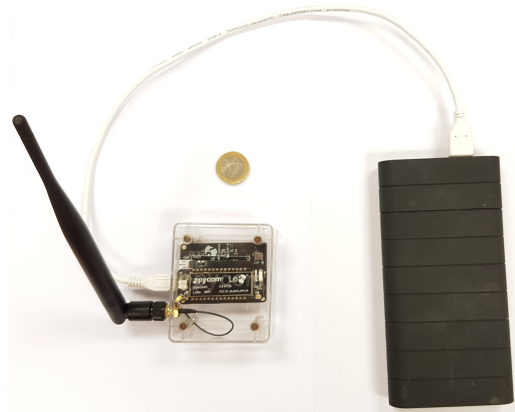


Figura 30: Nodo

También se han hecho pruebas de consumo energético del sistema general, donde se pudo observar que se tiene un consumo de energía máximo de 0.587 W, la alimentación mediante fuentes renovables como la solar puede ser de gran ayuda para el despliegue de estos dispositivos en áreas rurales. Una batería de 3.7 V, 2000 mAh es capaz de mantener durante 17 horas el *hub* y un panel solar de 5 W es capaz de generar la energía suficiente para recargar esta batería en la mayoría de los climas. La figura (31) nos muestra la LoPy, el panel solar y la batería.

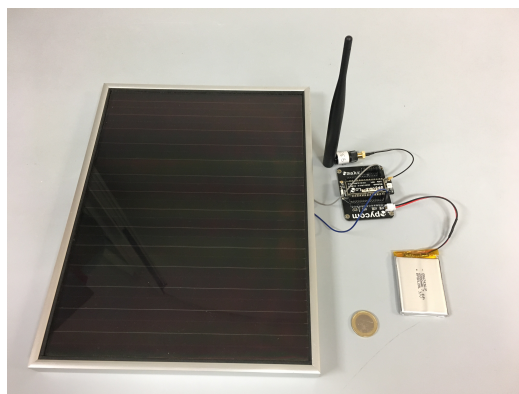


Figura 31: LoPy, panel solar y batería

Los nodos despliegan una señal inalámbrica donde los usuarios pueden conectarse para enviar y recibir mensajes mediante sus dispositivos móviles o equipos portátiles, se ha elegido utilizar el nombre de red “*messengerXXX*” donde las X representan dígitos del cero al nueve, cada vez que se enciende el dispositivo se genera un nombre aleatorio, en la figura (32) se pueden observar los nombres de red de dos dispositivos.

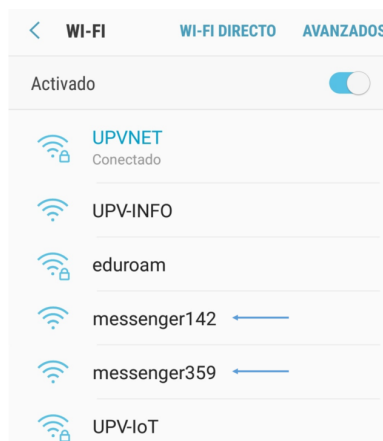


Figura 32: Nombres de red de los dispositivos

## 5.2 Escenarios

Las pruebas se realizaron en un entorno controlado, se fijó un punto de transmisión y tres puntos de recepción, el primer punto de recepción se encuentra a 50 m, el segundo a 100 m y el tercero a 200 m, para cada uno de los puntos de recepción se fijaron diferentes combinaciones de parámetros como son SF y el tamaño de mensajes. En la figura (33) podemos observar con una marca de color verde el punto de transmisión, la marca de color naranja identifica el punto de recepción a 50 m, la de color azul el punto de recepción a 100 m y la de color rojo el punto de recepción a 200 m, cabe destacar que existe línea de vista muy clara entre cada uno de los puntos.



Figura 33: Escenarios de pruebas



## 5.3 Pruebas

En la figura (34) podemos observar el conjunto ordenado de resultados en el proceso de descubrimiento del usuario como una función del parámetro 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.

El valor mediano para todo el proceso es 2.077 msec. Una vez que ya se conoce el dispositivo donde se encuentra el usuario, este proceso no se repite.

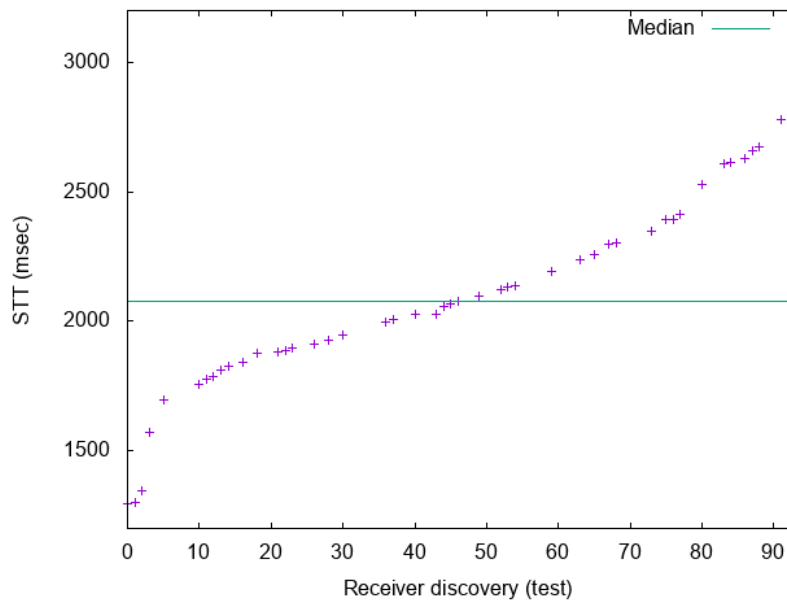


Figura 34: Proceso de descubrimiento de usuario

Al variar la distancia se puede observar en la figura (35) el comportamiento de los valores medianos del STT. La gráfica se obtiene usando un SF de 7 y un tamaño de mensaje de 1, 256, 512 *bytes*. Como se puede ver, hay un comportamiento estable en los resultados y es muy claro que el STT crece con forme se incrementa el tamaño del mensaje.

Los valores absolutos del STT resaltan que el tiempo de configuración requerido por los dos dispositivos para sincronizar el transmisor tiene la sobrecarga más alta, seguido del envío real de datos.

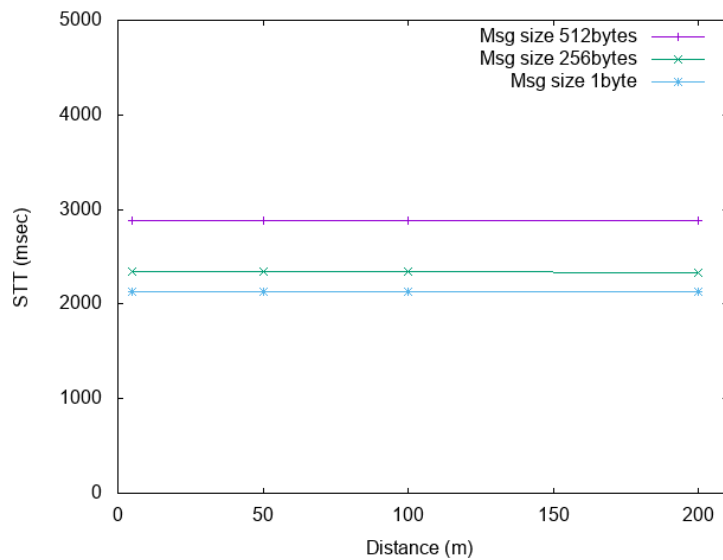


Figura 35: Comportamiento del STT al variar la distancia entre dos nodos

La figura (36) permite una mejor visualización de la evolución del STT en función del tamaño del mensaje que se transmite. Podemos observar que el STT crece a medida que se incrementa el tamaño del mensaje, pero resalta notablemente que el impacto de la distancia es insignificante para nuestro punto de recepción a 50 m y para 200 m.

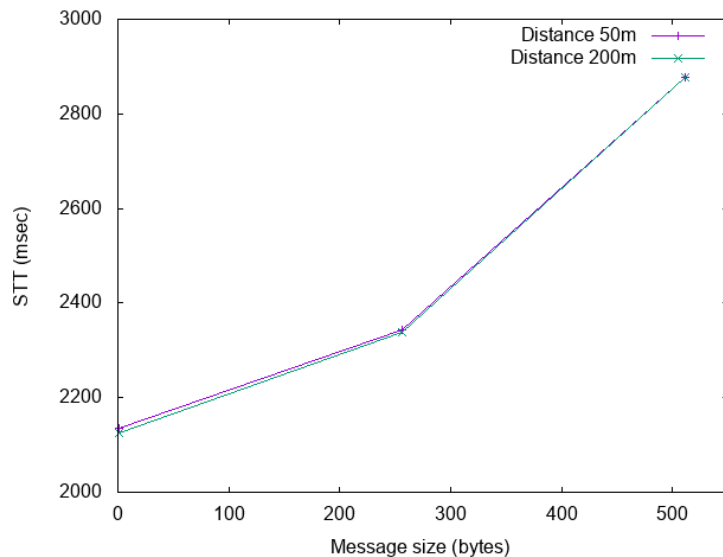


Figura 36: Comportamiento del STT al variar el tamaño del mensaje

Finalmente, la figura (37) permite observar el comportamiento del STT al variar la frecuencia de los mensajes generados, específicamente, cuando se incrementa el número de mensajes generados simultáneamente. La gráfica se obtiene utilizando un SF de 7 y un tamaño de mensaje de 256 bytes. Como puede verse, el STT crece de forma lineal a medida que aumenta el número de mensajes generados simultáneamente, dando el caso extremo en el que 14 usuarios simultáneos intentaron enviar un mensaje donde se obtuvo un retraso máximo de 80 segundos en algunos casos. El valor mediano, para los 14 usuarios simultáneos fue de 35.892 segundos. Con más de 14 usuarios simultáneos, el sistema mostró un comportamiento poco confiable y lo consideramos básicamente inutilizable.

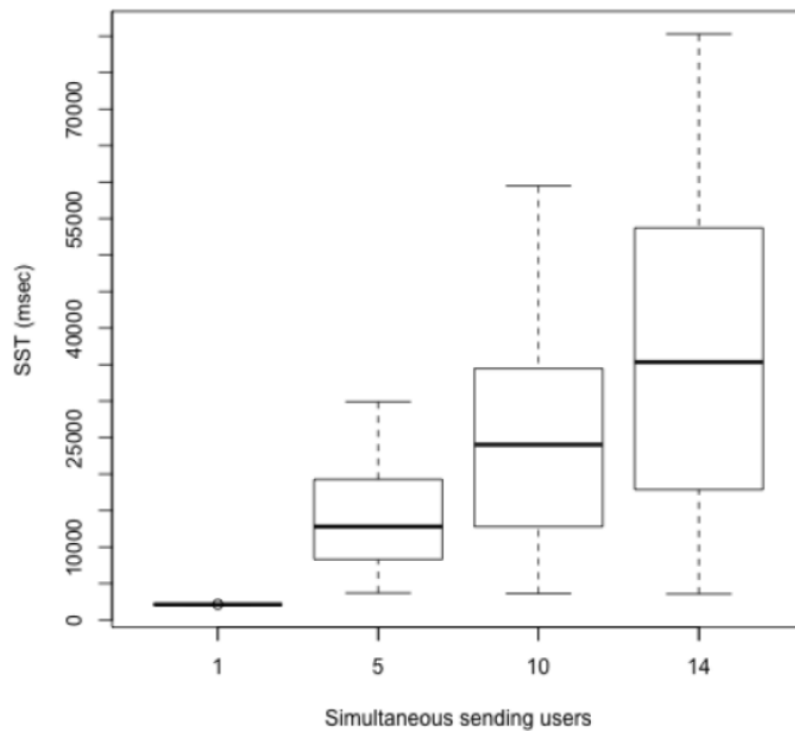


Figura 37: Evaluación de escalabilidad del sistema al aumentar la cantidad de usuarios que envían simultáneamente

Al parecer el sistema es bastante estable al aumento de la distancia y muy pocas retransmisiones fueron necesarias durante las pruebas. En general, se obtuvo una tasa de retransmisiones del 9 %, pero cabe señalar que la distribución de las retransmisiones era prácticamente independiente de la distancia.

# Capítulo 6

## Conclusiones

### 6.1 Conclusiones

Se presentó el diseño de una solución de bajo costo y bajo consumo energético que será capaz de proporcionar comunicación a comunidades aisladas. Las arquitecturas de red tradicionales se basan en el supuesto de que existe una infraestructura fija, y que los usuarios finales hacen uso de esta infraestructura mediante el pago de una tarifa. Es verdad que este modelo funciona bien en los países industrializados (donde las empresas o el gobierno se encargan de la inversión inicial), constituye una gran barrera en muchos países en desarrollo, en especial en zonas rurales y remotas.

La solución que se propuso requiere una inversión inicial limitada que podrán compartir los miembros de la comunidad y no requiere ningún costo recurrente. La arquitectura de red que se planteó se basa en dispositivos LoRa con un costo de alrededor de 30 euros. El costo total del dispositivo, incluido el panel solar, la batería y la carcasa, es de aproximadamente 70 euros.

Una vez el dispositivo se encuentre instalado y funcionando, el hub LoRa también se puede usar para sensores y otros datos de interés. Por ejemplo, la instalación de sensores meteorológicos en las laderas de las montañas, volcanes, ríos y otros lugares sensibles, además de útiles para el pronóstico del tiempo, puede ser particularmente valiosa para las advertencias tempranas sobre derrumbes, inundaciones, erupciones y otros desastres.

En esta primera etapa el sistema de mensajería fue capaz de transmitir paquetes de hasta 512 bytes, con un SF de 7 a 200 m, así como soportar el envío simultáneo de 14 usuarios, siendo bastante confiable y dejando la oportunidad

para realizar muchas mejoras ya que muestra amplias posibilidades de implementación.

Derivado de los trabajos realizados durante el desarrollo de este proyecto fue aceptado un póster del primer avance en ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS 2018) y se ha enviado un artículo con resultados más completos al 4th EAI International Conference on Smart Objects and Technologies for Social Good (GOODTECHS 2018), para dar a conocer los avances que se han obtenido.

## 6.2 Trabajos futuros

Dentro de los trabajos futuros para esta versión de la plataforma se encuentra realizar pruebas en escenarios en los cuales existan obstáculos en la línea de vista ya sean edificios o árboles, ambientes con o sin ruido.

Otra de las mejoras que se está considerando para la plataforma es agregar mensajes de voz para facilitar el uso del sistema a usuarios que no se sienten cómodos con la lectura y escritura.

Una opción para incrementar el alcance del sistema puede ser incluir dispositivos alternativos que ofrezcan mayores prestaciones tipo *Raspberry*, que actúen como *Gateway* LoRa de esta manera se puede considerar una conexión a aplicaciones de mensajería conectadas a Internet como por ejemplo *Telegram*. De esta manera las comunidades tendrían un mayor rango de comunicaciones.

## **Lista de Acrónimos**

- ARQ** Automatic Repeat-reQuest
- BBS** Bulletin Board System
- BW** Bandwidth
- CN** Community Networks
- CR** Code Rate
- CSS** Chirp Spread Spectrum
- DTN** Delay Tolerant Network
- FEC** Forward Error-correction Codes
- FHSS** Frequency Hopping Spread Spectrum
- FSK** Frequency-Shift Keying
- HTTP** Hypertext Transfer Protocol
- ID** Identificador
- IoT** Internet of Things
- IPN** Interplanetary Internet
- ITU** International Telecommunication Union
- LoRa** Long Range
- LPWAN** Low-Power Wide Area Networks
- MAC** Medium Access Control
- MAP** Mobile Access Point
- SF** Spreading Factor
- SNR** Signal to Noise Ratio
- STT** Successful Transfer Time
- UNB** Ultra Narrow Band

# Bibliografía

- [1] ITU, “Global ICT developments, 2001-2017,” ITU, Tech. Rep., 07 2018.
- [2] G. Cruz and G. Touchard, “Enabling rural coverage: regulatory and policy recommendations to foster mobile broadband coverage in developing countries,” GSMA, Tech. Rep., 07 2018.
- [3] A. Martinez, V. Villarroel, J. Seoane, and F. del Pozo, “Analysis of information and communication needs in rural primary health care in developing countries,” *IEEE transactions on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 66–72, 2005.
- [4] R. Baig, R. Roca, F. Freitag, and L. Navarro, “guifi. net, una infraestructura de red procomún gestionada de forma colectiva,” *Recuperado de: <http://people.ac.upc.edu/leandro/pubs/crowds-guifi-es.pdf>*, 2015.
- [5] D. Vega, L. Cerda-Alabern, L. Navarro, and R. Meseguer, “Topology patterns of a community network: Guifi. net,” in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 612–619.
- [6] Funkfeuer. (2018) Funkfeuer free net. Last visited: 2018-07-10. [Online]. Available: <https://www.funkfeuer.at/>
- [7] Guifi.net. (2018) Open, libre and neutral telecommunications network. Last visited: 2018-07-10. [Online]. Available: <https://guifi.net/en>
- [8] AWMN. (2010) Athens wireless metropolitan network. Last visited: 2018-07-10. [Online]. Available: <http://www.awmn.net>
- [9] Guifi.net. (2018) Guifi.net hoy. Last visited: 2018-07-12. [Online]. Available: <https://guifi.net/guifi/menu/stats/nodes>
- [10] M. Oliver, J. Zuidweg, and M. Batikas, “Wireless commons against the digital divide,” in *Technology and Society (ISTAS), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 457–465.

- [11] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: Rethinking connectivity in developing nations,” *Computer*, vol. 37, no. 1, pp. 78–83, 2004.
- [12] B. A. Forouzan and S. C. Fegan, *TCP/IP protocol suite*. McGraw-Hill Higher Education, 2002.
- [13] M. L. Doddamani and V. Shanwad, “Delay tolerant network,” *The International Journal of Science and Technoledge*, vol. 2, no. 5, p. 153, 2014.
- [14] V. Goncharov, “Delay-tolerant networks,” *Lehrstuhl für Rechnernetze und Telematik, Institut für Informatik, Albert Ludwigs Universität Freiburg*, 2010.
- [15] L. Gao, S. Yu, T. H. Luan, and W. Zhou, *Delay tolerant networks*. Springer, 2015.
- [16] F. Warthman *et al.*, “Delay tolerant networks (dtms); a tutorial. version 1.1,” 2003). <http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf>.
- [17] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, “Delay-tolerant networking: an approach to interplanetary internet,” *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.
- [18] M. H. Jain and R. Patra, “Implementing delay tolerant networking,” *University of California, Berkeley, Computer Science Division, Berkeley, CA 94720*, December 2003.
- [19] R. Flickenger, S. Okay, E. Pietrosemoli, M. Zennaro, and C. Fonda, “Very long distance wi-fi networks,” in *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*. ACM, 2008, pp. 1–6.
- [20] B. Raman and K. Chebrolu, “Experiences in using wifi for rural internet in india,” *IEEE Communications Magazine*, vol. 45, no. 1, pp. 104–110, 2007.
- [21] R. K. Patra, S. Nedeveschi, S. Surana, A. Sheth, L. Subramanian, and E. A. Brewer, “Wildnet: Design and implementation of high performance wifi based long distance networks.” in *NSDI*, vol. 1, no. 1, 2007, p. 1.
- [22] L. Vangelista, A. Zanella, and M. Zorzi, “Long-range iot technologies: The dawn of lora,” in *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*. Springer, 2015, pp. 51–58.



- [23] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of lora: Long range & low power networks for the internet of things,” *Sensors*, vol. 16, no. 9, p. 1466, 2016.
- [24] L. Alliance. (2018) Lora alliance. Last visited: 2018-07-24. [Online]. Available: <https://www.lora-alliance.org/>
- [25] E. ETSI, “300 220-1 (v2. 4.1),” *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods*, 2012.
- [26] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.
- [27] A. Berni and W. Gregg, “On the utility of chirp modulation for digital signaling,” *IEEE Transactions on Communications*, vol. 21, no. 6, pp. 748–751, 1973.
- [28] A. Springer, W. Gugler, M. Huemer, L. Reindl, C. Ruppel, and R. Weigel, “Spread spectrum communications using chirp signals,” in *EURO-COMM 2000. Information Systems for Enhanced Public Safety and Security. IEEE/AFCEA*. IEEE, 2000, pp. 166–170.
- [29] S. Corporation. (2016) Sx1276/77/78/79 datasheet. Last visited: 2018-08-01. [Online]. Available: [https://www.semtech.com/uploads/documents/DS\\_SX1276-7-8-9\\_W\\_APP\\_V5.pdf](https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf)
- [30] ———. (2017) Sx1272/73 datasheet. Last visited: 2018-08-01. [Online]. Available: <https://www.semtech.com/uploads/documents/sx1272.pdf>
- [31] B. Sikken. (2018) Project decodinglora. Last visited: 2018-08-01. [Online]. Available: <https://revspace.nl/DecodingLora>
- [32] L. A. T. Committee, *LoRaWAN™ 1.1 Specification*, final release ed., LoRa Alliance, Inc., 3855 SW 153rd Drive, Beaverton, OR 97003, October 2017.
- [33] M. Bor, J. E. Vidler, and U. Roedig, “Lora for the internet of things,” *International Conference on Embedded Wireless Systems and Networks*, 2016.
- [34] Pycom. (2018) Pycom go invent. Last visited: 2018-08-10. [Online]. Available: <https://pycom.io/>

- [35] P. Ltd., *LoPy 1.0 Manual*, final release ed., Pycom Ltd., High Point, 9 Sydenham Road, Guildford, Surrey GU1 3RX, UK, August 2017, last visited: 2018-08-16. [Online]. Available: <https://pycom.io/wp-content/uploads/2017/08/lopySpecsheetAugust2017.pdf>
- [36] —, *Expansion Board 2.0 Manual*, final release ed., Pycom Ltd., High Point, 9 Sydenham Road, Guildford, Surrey GU1 3RX, UK, June 2017, last visited: 2018-08-16. [Online]. Available: <https://pycom.io/wp-content/uploads/2017/06/expansionBoard2SpecsheetJune.pdf>
- [37] —, *Antena Lora & Sigfox*, Pycom Ltd., High Point, 9 Sydenham Road, Guildford, Surrey GU1 3RX, UK, 2018, last visited: 2018-08-16. [Online]. Available: <https://pycom.io/product/lora-antenna-kit/>