



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE INGENIERÍA  
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**“DISEÑO E IMPLEMENTACIÓN DE UNA RED INALÁMBRICA PARA  
MONITOREO DE PRODUCTOS AGRÍCOLAS BASADO EN EL  
INTERNET DE LAS COSAS”**

AUTOR  
PEDRO ALEJANDRO FICA GUTIÉRREZ

TRABAJO DE TÍTULO PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN AUTOMATIZACIÓN

CONCEPCIÓN – CHILE

2019



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE INGENIERÍA  
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UNA RED INALÁMBRICA PARA MONITOREO DE  
PRODUCTOS AGRÍCOLAS BASADO EN EL INTERNET DE LAS COSAS”

AUTOR

PEDRO ALEJANDRO FICA GUTIÉRREZ

DOCENTE PATROCINANTE: DR. CRISTIAN DURÁN FAÚNDEZ.

DOCENTE ADJUNTO O CORRECTOR: DR. PATRICIO GALEAS ECHEVERRÍA.

## Resumen.

---

### **RESUMEN**

Este trabajo hace parte del proyecto Fondef ID16I10206 “Sistema de apoyo a la toma de decisiones para la cosecha de frutas basado en Internet de las Cosas”, desarrollado en conjunto con la Universidad de la frontera y la Universidad de Talca.

Dentro del contexto del proyecto, este documento describe el diseño e implementación de una red de sensores inalámbricos para el monitoreo de productos agrícolas. La plataforma permite adquirir datos de sensores directamente empotrados en la caja de cosecha, almacenarlos localmente y transmitirlo a un servidor con el objetivo estar disponible para cualquier dispositivo en la red.

A lo largo del documento se describe la implementación del hardware del sistema de monitoreo, así como su estructura de red basado en un esquema tipo celular. Se propone un prototipo basado en módulos inAir9 que implementa la tecnología Lora, específicamente el chip SX1276. Para la interacción entre nodos se utiliza un protocolo de comunicación inalámbrica fundado en el protocolo PCF, Diseño que es parte de este desarrollo. Para la transmisión final de los datos se utilizaron los módulos Tp Link CPE510.

Para el diseño de la puerta de enlaces se utilizó una Raspberry Pi y para la implementación de código el software Qt Creator en el sistema operativo Raspbian. Para los nodos transmisores se utilizó la tarjeta Teensy 3.2 y todo el software elaborado para su funcionamiento se realizó sobre la IDE de Arduino.

Dedicatoria.

---

Dedicado a:

“A mi abuelo paterno por su incondicional apoyo, a mis padres y a todos aquellos que de una u otra forma contribuyeron en mi desarrollo profesional y académico”

Pedro Fica Gutiérrez. -

## Agradecimientos.

---

### AGRADECIMIENTOS

Terminar una tesis es la culminación de un largo tiempo de trabajo, estudio y esfuerzo. Fue un largo camino y hubiera sido muy difícil lograrlo si no es con el apoyo de muchas personas que de una u otra forma contribuyeron para su finalización.

Quiero agradecer al Dr. Cristián Durán Faúndez, quien en primera instancia me ofreció la oportunidad de participar en este proyecto y fue mi tutor para el desarrollo de esta tesis, por sus oportunos consejos y directrices que me fueron de gran utilidad. También las acertadas críticas y aportaciones de posibilidades de mejora, siempre desde una perspectiva constructiva, que contribuyeron sustancialmente en todos los aspectos de este trabajo.

Esta tesis fue realizada en el marco del proyecto Fondef ID16I10206, por lo que agradezco el apoyo financiero entregado. A los todos los investigadores del proyecto quienes, de diferentes formas, entregaron valiosa información para la construcción de esta tesis, siempre estaré agradecido.

Quiero hacer un agradecimiento al Profesor Vladimir Esparza, por su contante apoyo durante mi permanecía en la carrera. De igual forma le doy las gracias a todos los profesores el departamento de electricidad y electrónica de la universidad del Bío-Bío, por los conocimientos entregados y buena disposición para ayudar a resolver dudas.

Finalmente, y no por eso menos importante, pues está en mi corazón, Quiero agradecer a Dios por tener esta segunda oportunidad de vivir esta experiencia universitaria y la culminación exitosa de este proceso.

## ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	13
1.1 Introducción General .....	13
1.2 Objetivos .....	17
1.2.1 Objetivo General.....	17
1.2.2 Objetivos Específicos.....	17
1.3 Definición de alcances y exclusiones .....	17
1.4 Contenidos del trabajo de título.....	17
1.5 Metodología.....	18
1.6 Requisitos del proyecto.....	18
1.7 Condiciones de Frontera del proyecto .....	19
1.7.1 Frontera 1 .....	19
1.7.2 Frontera 2 .....	20
CAPÍTULO 2. TEORÍA DE PROTOCOLOS Y TECNOLOGÍAS DE COMUNICACIÓN..	21
2.1 Introducción.....	21
2.2 Dispositivos Finales .....	22
2.2.1 Lora.....	22
2.2.2 Comunicación SPI .....	26
2.3 Estación base .....	28
2.3.1 Comunicación serial .....	28
2.3.2 Protocolo Frontend/Backend.....	29
2.4 Servidores de red .....	30
2.4.1 Protocolo IEEE 802.11 n .....	30
2.4.2 WebSocket .....	30

Índice de contenidos.

---

2.4.3	Protocolo SSL.....	32
2.5	Protocolo de acceso al medio .....	32
2.5.1	Protocolo de transferencia de información en la red .....	33
2.6	Conclusión.....	36
CAPÍTULO 3. DESCRIPCIÓN DE LA PROPUESTA DE RED.....		37
3.1	Introducción.....	37
3.2	Descripción de los dispositivos utilizados .....	38
3.2.1	Nodos inalámbricos .....	38
3.2.2	Puerta de enlaces .....	45
3.3	Estructura de red.....	51
3.4	Conclusiones.....	51
CAPÍTULO 4. DISEÑO Y CONSTRUCCIÓN DEL SOFTWARE DE LA RED .....		52
4.1	Introducción.....	52
4.2	Descripción del software utilizado .....	53
4.2.1	Software red de sensores inalámbricos.....	54
4.2.2	Software puerta de enlace .....	64
4.2.3	Software de conexión a servicios de Internet.....	72
4.3	Conclusiones.....	73
CAPÍTULO 5. PRUEBAS DE CAMPO Y RESULTADOS .....		74
5.1	Introducción.....	74
5.1.1	Pruebas Aisladas.....	74
5.1.2	Pruebas de integración .....	75
5.1.3	Pruebas en las condiciones de sitio.....	76
5.2	Pruebas de campo .....	77
5.2.1	Dispositivos Lora.....	78
5.2.2	Transmisores Tp Link CPE510 .....	86

Índice de contenidos.

---

5.2.3 Pruebas de integración .....	88
5.3 Conclusión.....	90
CAPÍTULO 6. CONCLUSIONES.....	91
6.1 Alcances de diseño.....	91
6.2 Alcances económicos.....	92
6.3 Trabajo a futuro .....	92
Bibliografía.....	93

**ÍNDICE DE FIGURAS**

Figura 1.1: <i>Esquema de límites de responsabilidad.</i> .....	19
Figura 1.2: <i>Montaje de dispositivos en la base de caja cosechera.</i> .....	19
Figura 1.3: <i>Maqueta de visualización de estado (Peso, Temperatura y Aceleración) de nodos en tiempo real.</i> .....	20
Figura 2.1: <i>Diseño propuesto para establecer la comunicación.</i> .....	22
Figura 2.2: <i>Estructura de paquete de datos Lora.</i> .....	23
Figura 2.3: <i>Conexión entre maestro y esclavo.</i> .....	26
Figura 2.4: <i>Proceso de transmisión de información entre maestro y esclavo.</i> .....	27
Figura 2.5: <i>Esquema de comunicación SPI para los dispositivos Lora.</i> .....	28
Figura 2.6: <i>Estructura de una trama WebSocket.</i> .....	31
Figura 2.7: <i>Fase de descubrimiento de la red.</i> .....	33
Figura 2.8: <i>Formato de paquete de puerta de enlaces.</i> .....	34
Figura 2.9: <i>Fase de autenticación y asociación.</i> .....	34
Figura 2.10: <i>Fase de transmisión de datos útiles para el sistema de adquisición.</i> .....	35
Figura 3.1: <i>Topología de red tipo celular propuesta.</i> .....	37
Figura 3.2: <i>Disposición de dispositivos para la creación del nodo inalámbrico.</i> .....	39
Figura 3.3: <i>Placa Teensy 3.2.</i> .....	39
Figura 3.4: <i>Diagrama de Pines Teensy 3.2.</i> .....	40
Figura 3.5: <i>Soundon New Energy® Batería de Polímero de Litio.</i> .....	44
Figura 3.6: <i>Conexionado nodo sensor.</i> .....	44
Figura 3.7: <i>Raspberry Pi 3 B.</i> .....	45
Figura 3.8: <i>Captura de pantalla sistema operativo Raspbian PIXEL.</i> .....	47
Figura 3.9: <i>TP Link CPE510.</i> .....	48
Figura 3.10: <i>Topología estrella de transmisores finales.</i> .....	48
Figura 3.11: <i>Batería Ultracell Ciclo Profundo 7ah.</i> .....	49
Figura 3.12: <i>Diagrama unilineal de fuerza.</i> .....	49
Figura 3.13: <i>Conexionado de equipos que forman parte de la puerta de enlaces.</i> .....	50
Figura 3.14: <i>Estructura de red propuesta.</i> .....	51
Figura 4.1: <i>Representacion gráfica del proceso.</i> .....	53

Índice de figuras.

---

Figura 4.2: <i>Gráfico de muestras de aceleración eje x en un periodo de 8 horas.</i> .....	54
Figura 4.3: <i>Gráfico de muestras de giroscopio eje x en un periodo de 8 horas.</i> .....	55
Figura 4.4: <i>Gráfico de muestras de aceleración en un periodo de 8 horas.</i> .....	55
Figura 4.5: <i>Gráfico de muestras de peso en un periodo de 8 horas.</i> .....	56
Figura 4.6: <i>Gráfico de muestras de temperatura en un periodo de 8 horas.</i> .....	56
Figura 4.7: <i>Diagrama de flujo del software del nodo transmisor.</i> .....	57
Figura 4.8: <i>Diagrama de flujo de puerta enlaces etapa 1.</i> .....	65
Figura 4.9: <i>Diagrama de flujo de puerta enlaces etapa 2.</i> .....	68
Figura 5.1: <i>Esquema propuesto para la medición de RSSI.</i> .....	75
Figura 5.2: <i>Aplicación cliente /servidor.</i> .....	75
Figura 5.3: <i>Arboles o Arbustos frutales presentes en las condiciones de sitio.</i> .....	76
Figura 5.4: <i>Esquema de desplazamiento nodo transmisor.</i> .....	77
Figura 5.5: <i>Emplazamiento de Pruebas dentro del campus universitario.</i> .....	78
Figura 5.6: <i>Gráfico de datos obtenidos de la medición de RSSI cada 10 metros.</i> .....	79
Figura 5.7: <i>Gráfico de datos obtenidos de la medición de RSSI en distintos tiempos de muestreo.</i> .....	79
Figura 5.8: <i>Ubicación de empresa Agrícola Industrial El Boldo.</i> .....	80
Figura 5.9: <i>Panorámica de huerto n°1, fundo santa Isabel.</i> .....	81
Figura 5.10: <i>Instalación de estación base.</i> .....	82
Figura 5.11: <i>Instalación del nodo transmisor.</i> .....	82
Figura 5.12: <i>Gráfico de medición de RSSI.</i> .....	83
Figura 5.13: <i>Grafico de medición de SNR.</i> .....	83
Figura 5.14: <i>Gráfico de paquetes perdidos.</i> .....	84
Figura 5.15 : <i>Datos de RSSI, SNR en una zona de 350 metros.</i> .....	84
Figura 5.16: <i>Estación base ubicada en punto central.</i> .....	85
Figura 5.17: <i>Emplazamiento de las pruebas realizadas.</i> .....	86
Figura 5.18: <i>Montaje de CPE510 aplicación cliente.</i> .....	87
Figura 5.19: <i>Montaje de CPE510 aplicación servidor.</i> .....	87
Figura 5.20: <i>Diagnóstico de red utilizando la herramienta ping.</i> .....	88
Figura 5.21: <i>Montaje del sistema de diseñado.</i> .....	88
Figura 5.22: <i>Aplicacio que sirve de datos al sistema.</i> .....	89
Figura 5.23: <i>Tabla de datos tipo 1.</i> .....	89

## Índice de tablas.

**ÍNDICE DE TABLAS**

Tabla 2.1: <i>Estructura de paquete de datos Lora.</i> .....	24
Tabla 2.2: <i>Estructura de datos del payload.</i> .....	25
Tabla 2.3: <i>Parámetros comunicación serial.</i> .....	29
Tabla 2.4: <i>Paquete configuración de puerta de enlaces.</i> .....	36
Tabla 3.1: <i>Tabla de conexiónado Lora-Teensy 3.2.</i> .....	40
Tabla 3.2: <i>Características del dispositivo.</i> .....	42
Tabla 3.3: <i>Banda de frecuencia ISM compatibles con el módulo inaIR9 de radio propuestos.</i> ...	42
Tabla 3.4: <i>Características técnicas del módulo SX1276.</i> .....	43
Tabla 5.1: <i>Registro de medición de RSSI frente a variaciones de distancia.</i> .....	74
Tabla 5.2: <i>Registro de medición de RSSI frente a variaciones de tasa de muestreo.</i> .....	74
Tabla 5.3: <i>Registro de medición de RSSI frente a variaciones de distancia.</i> .....	76
Tabla 5.4: <i>Registro de condiciones climáticas.</i> .....	77
Tabla 5.5: <i>Tabla de configuración de nodo inalámbrico.</i> .....	78
Tabla 5.6: <i>Condiciones climáticas fundo santa Luisa.</i> .....	81
Tabla 5.7: <i>Configuraciones CPE510.</i> .....	86

Abreviaciones.

---

**ABREVIACIONES**

Mayúsculas

FONDEF : Fondo de Fomento al Desarrollo Científico y Tecnológico

SPI : Serial Peripheral Interface

TCP : Protocolo de control de transmisión

IP : Internet Protocol

IEEE : Instituto de Ingeniería Eléctrica y Electrónica

TLS : Transport Layer Security

JSON : JavaScript Object Notation

SSL : Secure Socket Layer

MAC : Media Access Control

IOT : Internet of Things

M2M : Machine to machine

ISM : Industrial, Scientific and Medical

RSSI : Received Signal Strength Indicator

SNR : Signal-to-noise ratio

SQL : Structured Query Language

## CAPÍTULO 1. INTRODUCCIÓN

### 1.1 Introducción General

Desde hace tiempo se sabe que la tecnología se ha posicionado como el principal pilar que da soporte al desarrollo y crecimiento económico de las naciones. Una de las tecnologías más importantes que dio a luz en el siglo XX es el Internet [1], que ha evolucionado todas las instancias creadas por el hombre, no sólo si observamos desde la perspectiva de la comunicación si no también desde como pensamos hasta como aprendemos. Pocas estructuras tecnológicas han cambiado la humanidad hasta el punto definir un antes y un después, avanza con nosotros y para nosotros, llegando a cada lugar que el hombre ha conquistado [2].

El Internet cada vez tiene una mayor penetración en el mundo cotidiano, ha modificado la forma de relacionarnos con todos los objetos tecnológicos que han hecho más sencilla y productiva la vida, con esto se han delegado responsabilidades a simples instrucciones con la capacidad de tener el control de la actividad de cada una de las aplicaciones que nos rodean y de cada artefacto creado para la satisfacción de las necesidades de la humanidad. El Internet, no sólo son miles de computadores conectados, ahora deben considerarse todos los objetos que participan en nuestras actividades cotidianas y que mediante la inclusión de sensores son capaces de entregar información. Esta tecnología ha tomado una nueva forma, más cercana, conocida como el Internet de las Cosas y es en la actualidad la próxima revolución y trae consigo profundas transformaciones.

Dicha revolución tecnológica se ha incorporado transversalmente a todas las actividades que son parte de nuestra civilización, siendo una de especial relevancia en la historia de la humanidad, aquella que impactó la forma de los asentamientos humanos y que sentó las bases para el desarrollo de múltiples naciones, la agricultura.

La actividad humana ha dependido en gran manera de la agricultura para su supervivencia [3], desde que hace su aparición en el neolítico se ha transformado y ha adquirido características que hacen posible la explotación eficiente de los recursos disponibles en la tierra.

La agricultura y sus ramas no son ajenas al avance del Internet, en ninguna manera. Contar con medios para disponer de información de cosechas ha permitido crear procedimientos que permiten un mayor control y conseguir resultados que responden a las expectativas de un mercado creciente.

## Capítulo 1: Introducción.

---

El auge de tecnologías y técnicas más sofisticadas ha traído una ola de cambios significativos en la manera en se plantean las metas productivas. Es decir, cada vez se hace más necesario la incorporación de metodologías innovadoras capaces de mitigar riesgos y aumentar ganancias

La tecnología aplicada a la agricultura se conoce también con el término de agrotecnología [4], y en el último tiempo el número de aplicaciones digitales ha crecido conforme las necesidades se tornan más explícitas y los requisitos de clientes ponen desafíos más difíciles de alcanzar.

A lo largo de la historia se han resuelto muchas problemáticas que han aquejado a los productores de entre ellas se puede mencionar el control de plagas.

Es sabido que los daños provocados por las plagas y las enfermedades han estado desde la fundación de la agricultura, por tanto, combatir una plaga para un agricultor significa que está en juego toda una inversión y no deja de ser un tema relevante si es puesto sobre la mesa. Dentro de ese marco la detección temprana de plagas ha jugado un papel importante para sembrar las bases del desarrollo de tecnología en esta área.

En el campo de la investigación el doctor, Christian Hansen de la Universidad de California ha desarrollado un sistema de cámaras capaz de detectar en los cultivos síntomas de estrés y con esto determinar una posible infección por plagas [5]. Esta detección se basa en el hecho que un cultivo saludable refleja una luz que es muy distinta a la que reflejan aquellos que están estresados debido a la presencia de insectos u otro atacante. Para esto, han desarrollado un sistema de visión artificial capaz de detectar mínimos cambios en la reflectancia foliar y, junto con esto reflejar, de manera muy precisa el surgimiento de las infestaciones de plagas.

Una contribución interesante se refleja en el trabajo de título “Diseño de una red inalámbrica de sensores para apoyar actividades de agricultura de precisión en el jardín botánico de Cali” [6], en este trabajo se desarrolló una red de sensores para monitorear variables ambientales en grandes extensiones y espacios abiertos, solución aplicable en ambientes agrícolas y medioambientales para contribuir y abordar problemáticas tales como la prevención y el control de incendios forestales.

La solución al problema se desarrolló en un prototipo de nodo que es capaz de registrar variables ambientales seleccionadas y transmitir las a través de la red, para su posterior almacenamiento en una base de datos y que permite realizar consultas a través de la web.

## Capítulo 1: Introducción.

---

Hay muchas más aplicaciones, Pero todas tienen en su esencia un cambio en forma en la información es adquirida y procesada, donde la toma de decisiones se basa principalmente en el análisis de grandes volúmenes de datos. Diversas plataformas son construidas por los ingenieros de software, capaces de interpretar la información y entregar los medios asequibles para que el agricultor fácilmente pueda tomar una decisión.

La industria frutícola no queda exenta de la penetración del internet de las cosas, que de por si es una tendencia generalizada, Gracias a estas tecnologías es posible aumentar el rendimiento de cada m<sup>2</sup> construido, optimizar cada uno de los procesos, todo esto con el fin de mejorar la calidad del producto final y por sobre todo un aumento considerable del margen de ganancias.

Dentro de la contextualización de la realidad de mercado que vive el país, en 2016 el PIB chileno totalizó 247.290 millones de dólares y el PIB per cápita, calculado en paridad de poder de compra por el FMI, 23.969 dólares [7]. Uno de los sectores que más creció en la economía nacional fue el sector silvoagropecuario con lo que aumentó su participación en el PIB nacional, es importante recalcar que la industria fruticultura aporta un tercio del valor agregado por el sector.

Desde esa perspectiva y tomando los antecedentes antes expuestos. Se presenta una problemática que ha amenazado los intereses de los inversores en Chile de la industria frutícola. Hoy en día exportar fruta, no es tan simple, es necesario asegurar altos estándares de calidad exigidos por los clientes y los diferentes países destinos de la fruta.

En la actualidad el proceso de cosecha de frutas, es realizado por personas que no requieren formación académica para su recolección, A pesar de que reciben una breve charla, sólo es necesario para la recolección de esta fruta requerir de cierta destreza debido a su alta susceptibilidad a sufrir daños en el proceso, Bajo ese contexto, la mantención de la calidad de la fruta queda sujeta principalmente a la manipulación que ejerce el personal a cargo, Ahora, si consideramos todas las etapas que son necesarias para que estas lleguen a su lugar de destino, el daño es acumulado y se exponen en un producto que no reúne las características necesarias para su venta.

En estudios realizados en frutillas, se estableció que alrededor del 51% del daño mecánico ocurre en la operación de cosecha, y es aquí donde se identifica una problemática a resolver [8]. Muchos estudios afirman que el daño mecánico es ocasionado principalmente por el impacto, la vibración

## Capítulo 1: Introducción.

---

y la presión que ejerce el peso de la fruta [9], [10]. Otro autor [11] agrega que la temperatura de almacenamiento y el daño mecánico afectan directamente la calidad del arándano.

Es relevante conocer qué sucede durante todo el proceso de manipulación de la fruta, con el objetivo de determinar si los procedimientos o personal son los idóneos para realizar la actividad. En ese sentido, monitorear en tiempo real condiciones de movimiento y temperatura en que se ve expuesta la fruta, abre un abanico de oportunidades para el análisis y estudio, todo esto con el fin de estimar daños y la vida útil que puedan tener.

En el marco del proyecto Fondef ID16I10206 se está desarrollando tecnología de sensorización de variables agrícolas, manifestada en un dispositivo con la capacidad de registrar peso, temperatura y cantidad de movimiento (vibraciones). Este diseño contempla la inclusión de la electrónica directamente empotrada en la caja de recolección, con el objetivo que la integración sea simple y no afecte la estructura del cajón cosechero. Las variables en estudio serán enviadas inalámbricamente a un nodo central, donde serán almacenadas localmente y enviadas hasta un servidor, en la que serán analizadas por un modelo matemático que determinara la calidad de la fruta.

El desafío del presente trabajo de título es construir la red inalámbrica que dará soporte a toda la interacción comunicativa entre sensores y elementos finales. Para esto se ha desarrollado la ingeniería en la que se consideró el diseño de la estructura y software necesario para construir un prototipo funcional que reúna las condiciones para funcionar en las condiciones de trabajo de la industria de la recolección de frutas.

## Capítulo 1: Introducción.

---

### 1.2 Objetivos

#### 1.2.1 Objetivo General

- Desarrollar e implementar tecnología de comunicación inalámbrica para adquisición, paquetización y transmisión de datos sensados de un sistema de recolección frutas

#### 1.2.2 Objetivos Específicos

- Diseñar el protocolo de comunicación inalámbrica entre los dispositivos de hardware del sistema
- Diseñar la estructura de la red
- Implementar un prototipo de comunicación capaz de operar en las condiciones de sitio

### 1.3 Definición de alcances y exclusiones

El trabajo de título considera el diseño de un prototipo de red inalámbrica, Los alcances para este desarrollo son los siguientes por disciplina.

Electrónica: Determinación de equipos para el desarrollo y todas configuraciones necesarias para la integración final

Informática: Definición de los protocolos de red a utilizar desde el nodo sensor hasta el servidor, se incluye la construcción de todo el software necesario.

Se excluye del estudio el diseño del software necesario para el análisis de los datos obtenidos.

### 1.4 Contenidos del trabajo de título

Este proyecto contempla la inclusión de diferentes tópicos que dan forma a este trabajo de título. En el capítulo introducción, se dan a conocer al lector desde una perspectiva general el trasfondo que sustenta la problemática. En este tópico se dan a conocer los objetivos generales y específicos. Se presentan los límites de batería y la metodología para concretar este proyecto de ingeniería.

En el capítulo 2, se presenta las bases teóricas de los protocolos de comunicación propuestos para establecer la interacción entre los equipos y entidades que dan forma a la red, dando a conocer el estándar adoptado para el desarrollo de la comunicación y cómo se implementó.

## Capítulo 1: Introducción.

---

Seguidamente, en el capítulo 3 se da a conocer la solución propuesta, el diseño estructural de la red de sensores inalámbricos y la selección de todo el hardware que será parte del prototipo final.

En el cuarto capítulo, se expone la construcción de software necesario para el procesamiento y paquetización de datos sensados, haciendo referencia a todas la plataformas y tecnologías necesarias para el transporte, almacenamiento y posterior revisión de los datos de sistema de adquisición.

El capítulo 5 da a conocer las pruebas de funcionalidad de los dispositivos inalámbricos y los resultados obtenidos.

Finalmente se presentan las conclusiones obtenidas en el desarrollo de este trabajo.

### **1.5 Metodología.**

La metodología para el desarrollo del presente trabajo de título se divide en las siguientes etapas:

- Estudiar la problemática y definir una solución que resuelva íntegramente el problema,
- Realizar pruebas aisladas con los equipos seleccionados.
- Integrar la solución y realizar pruebas de campo.
- Documentar toda la información y software necesario para replicar la solución

### **1.6 Requisitos del proyecto**

Las instalaciones y el equipamiento eléctrico serán adecuados para operar en forma permanente 8 horas por día, durante los meses de cosecha.

El prototipo deberá ser funcional según las condiciones de frontera del proyecto Fondef ID16I10206 descritos en el siguiente documento.

La red nodos inalámbricos propuestos deberá cubrir un área superior a los 2.25 km<sup>2</sup>.

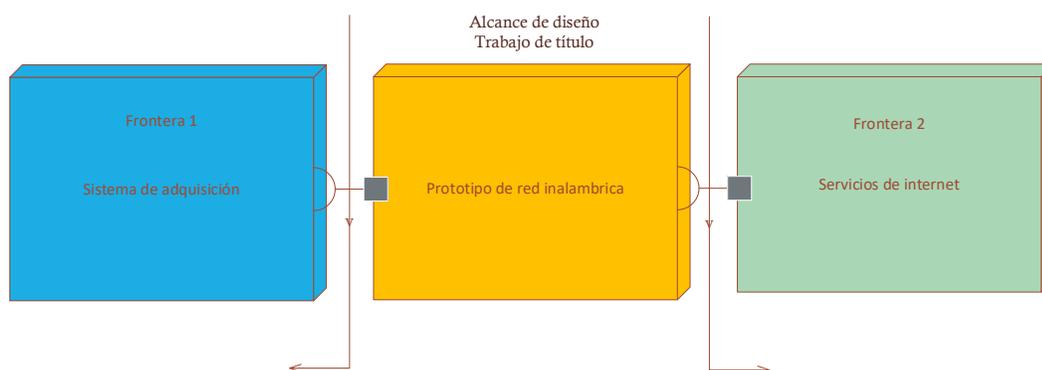
El sistema deberá constar de almacenamiento local y un acceso redundante a los datos adquiridos.

## Capítulo 1: Introducción.

### 1.7 Condiciones de Frontera del proyecto

Este trabajo de título es parte del proyecto Fondef ID16I10206 “Sistema de apoyo a la toma de decisiones para la cosecha de frutas basado en Internet de las Cosas”, Por tanto, es necesario establecer las condiciones y requisitos que permitirán la integración de este trabajo de título con las demás partes involucradas en el proyecto anfitrión.

La Figura 1.1 esquematiza la situación y determina los límites de responsabilidad.



**Figura 1.1:** Esquema de límites de responsabilidad.

#### 1.7.1 Frontera 1

La frontera 1 corresponden todos los trabajos relacionados al sistema de adquisición, específicamente el diseño de la caja de cosecha y toda la electrónica empotrada con el objetivo de obtener datos de los sensores instalados. En la Figura 1.2 se aprecia la base de la caja cosechera y toda la electrónica relacionada al transmisor, tarjeta de desarrollo y sensores.



**Figura 1.2:** Montaje de dispositivos en la base de caja cosechera.

**Fuente:** Reporte de Hito 1.4 “Prototipo Beta del módulo de monitoreo”.

## Capítulo 1: Introducción.

En este diseño se incluye el montaje de la tarjeta de desarrollo y el conexionado de todos los sensores involucrados.

### 1.7.2 Frontera 2

La frontera 2 corresponden todos los trabajos relacionados con los servicios de internet, específicamente el diseño del sistema de soporte de decisión (“Decision Support System” o “DSS” por sus siglas en inglés). Este software cubre todas las etapas de recolección de datos y presentación de información procesada por el sistema, con el fin de entregar los medios necesarios para que la organización que implemente un DSS pueda tomar decisiones con el mayor nivel de información [12].



**Figura 1.3:** Maqueta de visualización de estado (Peso, Temperatura y Aceleración) de nodos en tiempo real.

**Fuente:** Informe de Hito 2.4 “Prototipo Final del DSS”.

La conexión con esta plataforma condiciona entregar los datos desde la puerta de enlaces hacia el APP Server que contiene el bus de mensajes (ws\_server). en formato JSON y utilizando la tecnología WebSocket para abrir un canal bidireccional sobre un mismo socket TCP.

## CAPÍTULO 2. TEORÍA DE PROTOCOLOS Y TECNOLOGÍAS DE COMUNICACIÓN

### 2.1 Introducción

Las redes de comunicaciones, en la actualidad, se han convertido en la materia esencial de la vida digital del mundo moderno, siendo las tecnologías de la información el medio fundamental.

La transmisión de datos no es posible sin la existencia de protocolos de comunicación. En el contexto de redes de datos, un protocolo comprende a todas las reglas que permiten que dos o más entidades de un sistema de comunicación puedan transmitir o intercambiar información a través de la red.

Los sistemas de comunicación utilizan reglas y formatos bien definidos para establecer el intercambio de datos, nada queda al azar. Estas reglas definen la sintaxis, semántica, sincronización de la comunicación y fijan los procedimientos que permiten recuperar datos que pueden ser perdidos durante el intercambio [12].

Es importante establecer los protocolos en cada uno de los niveles de transmisión de información con el fin de determinar los equipos necesarios.

Para el desarrollo e implementación del prototipo de red se tomó el modelo referencial OSI, el cual está definido por 7 capas, estas son: física, enlace de datos, de red, de transporte, presentación y aplicación [13]. Se ha tomado este modelo, como guía general para que todos los equipos involucrados puedan comunicarse a través de la red y garantizando de esta forma conectividad de extremo a extremo.

Se han definido tres niveles para dar soporte al prototipo, desde los dispositivos finales hasta dispositivos como el servidor. Estos son;

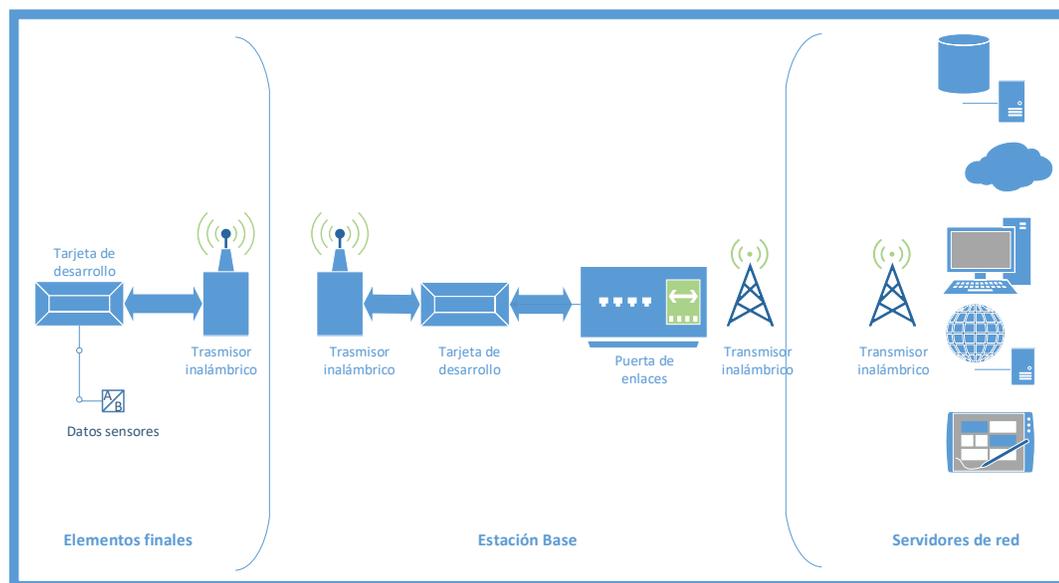
1. Dispositivos finales: Dentro de esta categoría, se encuentran los nodos, que reciben la data de los sensores empotrados en la caja de recolección de frutas.
2. Estacion Base: Punto central de los dispositivos inalámbricos, que recibe las transmisiones de múltiples nodos sensores, almacena la información en un base de datos local y la retransmite a la nube.
3. Servidores de red: Equipos que reciben la información y la procesan.

La red propuesta entre los nodos sensores y la puerta de enlaces obedece a una configuración de tipo celular, Cada nivel propuesto, necesita la implementación de un protocolo o tecnología para

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

poder permitir la comunicación. En ese sentido a continuación se presentan todos los recursos utilizados, para permitir la transferencia de datos desde la adquisición hasta el sistema de monitoreo en línea.

En la Figura 2.1 se aprecia el diseño del esquema propuesto.



**Figura 2.1:** Diseño propuesto para establecer la comunicación.

## 2.2 Dispositivos Finales

Dentro de los dispositivos finales tenemos a la red de nodos inalámbricos, se ha sugerido utilizar la tecnología inalámbrica Lora. Basado en el esquema anterior, se definen las tecnologías y protocolos implementados en este nivel.

### 2.2.1 Lora

Lora, abreviatura de Long Range, es una tecnología inalámbrica que utiliza un tipo de modulación en radiofrecuencia, basada en técnicas de espectro ensanchado y una variación de Chirp Spread Spectrum (CSS), que modula los datos sobre diferentes canales y velocidades, con corrección de errores Forward Error Correction (FEC) integrada [14].

Lora está diseñada para redes de baja potencia, de largo alcance y bajo consumo de energía. Funciona sobre la banda ISM, dentro de las consideraciones se configurado el dispositivo para utilizar la banda de frecuencia 868 MHz

Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

En referencia al modelo OSI, tenemos que la implementación de la capa física utiliza Spread Spectrum Modulation (SSM), Este tipo de modulación, codifica la señal de entrada en una secuencia de alta frecuencia, con un “pitido” o Chirp y la propaga a través de un mayor ancho de banda que aumenta o disminuye con el tiempo [15].

Dentro de los parámetros configurables que permite el dispositivo, se tiene el factor de propagación (SF7-SF12) , y los diferentes anchos de banda (125 kHz,250 kHz,500 kHz).

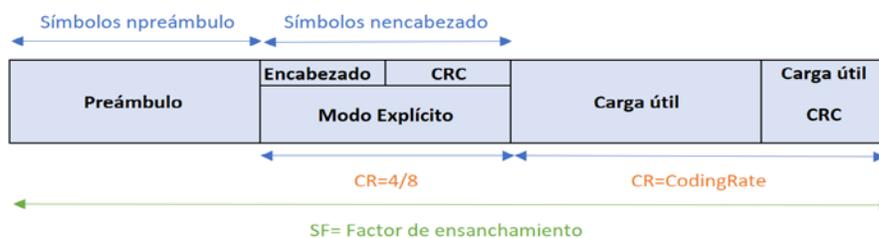
Se ha considerado utilizar un factor de propagación SF7 y un ancho de banda 125 kHz. Con esta definición es posible determinar la tasa de bits, que quedan definidos por la Ecuación 2.1. Con esta configuración la tasa de bit alcanza el valor 6836.

$$Bit\ rate = SF * \frac{BW}{2^{SF}} \tag{2.1}$$

Esta tecnología permite generar un conjunto de canales virtuales, considerando que los dispositivos pueden ser configurados a diferentes velocidades de datos permitiendo que los mensajes se propaguen ortogonalmente unos con otros, disminuyendo así las tasas de interferencia y generando un aumento en la capacidad de la puerta de enlace.

La capa de enlace de datos, es necesario programarla a través de la interfaz de Arduino, donde es necesario, asignar una dirección MAC a cada nodo. Esto supone una gran ventaja a lo hora de implementar estrategias en el control de la transmisión y recepción de mensajes.

En los que respecta a la estructura del mensaje, el módulo Lora emplea dos tipos de formato de paquete, explícito e implícito. El paquete explícito incluye un encabezado corto que contiene información sobre el número de bytes, la tasa de codificación y si se utiliza un CRC en el paquete. El formato del paquete se muestra en la siguiente Figura:



**Figura 2.2:** Estructura de paquete de datos Lora.

**Fuente:** Semtech SX1276 (2016).

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

---

De forma nativa, el paquete Lora debe contar con al menos 3 elementos

1. Un preámbulo
2. Un encabezado opcional
3. Un payload (datos)

El preámbulo se usa para sincronizar el receptor con el flujo de datos entrantes. Por defecto, el paquete está configurado con doce símbolos de larga secuencia. Esta es una variable programable, por lo que la longitud del preámbulo puede extenderse.

Dependiendo del modo de operación elegido, hay dos tipos de encabezado disponibles. El tipo de encabezado es seleccionado por el bit `ImplicitHeaderModeOn`, que se encuentra dentro del registro `RegModemConfig1`, para el desarrollo se ha seleccionado el modo de encabezado Explícito.

El encabezado se transmite con el código máximo de corrección de errores (8/4). También tiene su propio CRC para permitir que el receptor descartar encabezados inválidos.

En ciertos escenarios, donde la carga útil, la tasa de codificación y la presencia de CRC son fijas o conocidas de antemano, puede ser ventajoso para reducir el tiempo de transmisión al invocar el modo de encabezado implícito. En este modo, el encabezado se elimina del paquete. En este caso, la longitud del payload, la tasa de codificación de errores y la presencia de CRC deben configurarse manualmente [16].

La carga útil del paquete (payload) es un campo de longitud variable que contiene los datos del sistema de adquisición en byte. El máximo payload soportado corresponde a 256 byte.

En lo que respecta para el proyecto se ha considerado el paquete de datos Lora, en la tabla 2.1

**Tabla 2.1:** Estructura de paquete de datos Lora.

Paquete Lora	Tamaño [byte]
Dirección de destino	1
Dirección de Fuente	1
OCP	1
Largo Identificador	1
Identificador	4
Largo Payload	1
Payload	47

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

---

En el encabezado del paquete, se especifica la dirección de destino 0xFF, que tendrá la definición de broadcast para nuestro sistema. En lo que respecta a la estructura del payload se ha determinado el esquema de la Tabla 2.2

**Tabla 2.2:** Estructura de datos del payload.

Datos	Tamaño [Byte]	Descripción
byte control	1	Inicio paquete
Opt	2	Opcional
ID nodo	4	Identificador del nodo
Timestamp	4	Tiempo
batería	2	Indicador de batería
Cluster1	2	Acelerómetro
Cluster2	2	Acelerómetro
Cluster3	2	Acelerómetro
Cluster4	2	Acelerómetro
Cluster5	2	Acelerómetro
Peso	4	Indicador Peso
Tem1	2	Temperatura 1
Tem2	2	Temperatura 2
Tem3	2	Temperatura 3
LatitudGPS	4	Latitud GPS
LongitudGPS	4	Longitud GPS
GPSP2	4	GPSP2
Checksum	1	Suma de verificación
Bytefinal	1	Fin de paquete

En la Tabla 2.2 se especifica el orden en que serán almacenados los datos y cantidad de byte asignados en la memoria, que suman un valor de 47 byte.

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

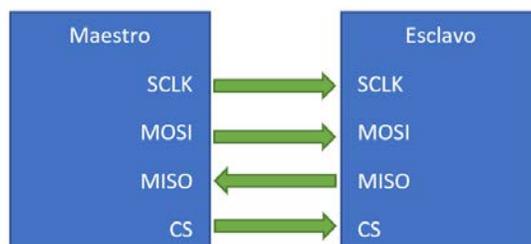
### 2.2.2 Comunicación SPI

Se ha propuesto utilizar una tarjeta de desarrollo para adquirir y procesar los datos de los sensores instalados en la caja de cosecha. Una vez que han sido tratados, por lo procedimientos descritos en el capítulo 4, estos deben ser transmitidos, para esta instancia se utiliza el protocolo SPI como interfaz entre el transmisor inalámbrico y la tarjeta de desarrollo

SPI es un protocolo de comunicación síncrono, cuyas siglas utilizadas son para referirse a Serial Peripheral Interface. Este Protocolo nace de la mano de Motorola en el año 1980 para permitir la comunicación entre diferentes equipos electrónicos [17].

Este estándar de comunicación trabaja en modo full dúplex para recibir y transmitir información, permitiendo que dispositivos puedan comunicarse en el mismo instante de tiempo utilizando líneas diferentes contenidas en el mismo cable. Utiliza un cable dedicado para asegurar que el proceso de transferencia de información este sincronizado.

Se define un maestro, quien es responsable de enviar la información a los esclavos, por otro lado están los esclavos, quienes reciben información y envían información al maestro. En este proceso se realiza utilizando dos líneas, una para la transferencia maestro esclavo (MOSI) y otra utilizada para el esclavo maestro (MISO). Sin embargo, la transmisión es serial, por tanto, la información es escrita de un byte a la vez en cada ciclo de reloj (SCLK) . Para comenzar o transmitir un mensaje, se incluye una línea encargada de seleccionar y a su vez, habilitar un esclavo (CS). El esquema mostrado a continuación representa lo anterior.



**Figura 2.3:** Conexión entre maestro y esclavo.

Existen 4 modalidades en las cuales es posible transmitir la información y cada una de estas está determinada por la configuración de la señal de reloj. Estas configuraciones obedecen a la polaridad

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

y la fase características propias de una señal. Por tanto, es posible obtener 4 modos distintos de realizar la transmisión y el envío de información.

Estos Modos son

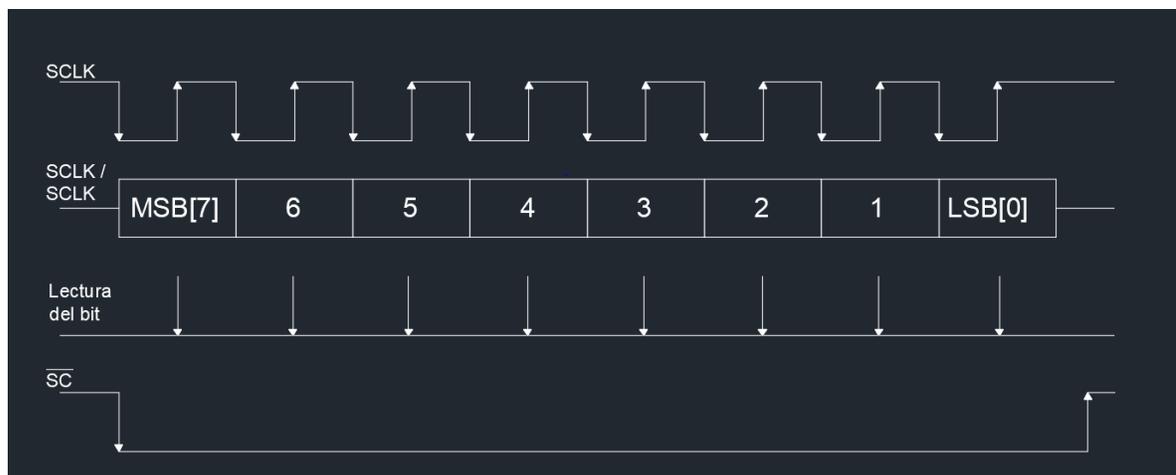
Modo 0: CPOL = 0 y CPHA = 0. Modo en el cual el estado del reloj permanece en estado lógico bajo y la información se envía en cada transición de bajo a alto, es decir alto activo.

Modo 1: CPOL = 0 y CPHA = 1. Modo en el cual el estado del reloj permanece en estado lógico bajo y la información se envía en cada transición de alto a bajo, es decir bajo activo.

Modo 2: CPOL = 1 y CPHA = 0. Modo en el cual el estado del reloj permanece en estado lógico alto y la información se envía en cada transición de bajo a alto, es decir alto activo.

Modo 3: CPOL = 1 y CPHA = 1. Modo en el cual el estado del reloj permanece en estado lógico alto y la información se envía en cada transición de alto a bajo, es decir bajo activo [18]

La Figura 2.4 representa el proceso de transmisión de información entre maestro y esclavo, utilizando la configuración CPHA=0 y CPOL=0



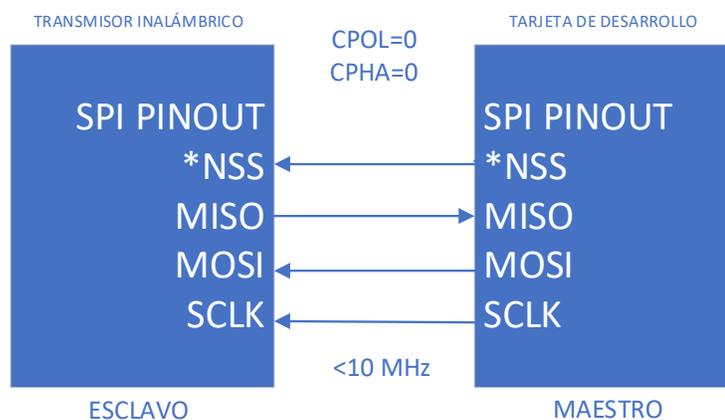
**Figura 2.4:** Proceso de transmisión de información entre maestro y esclavo.

Este protocolo ofrece una serie de ventajas, una de ellas a destacar es la velocidad de transmisión, debido a que trabaja en modo full dúplex, es muy superior a otros protocolos del tipo serial que trabajan en modo half dúplex.

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

El transmisor inalámbrico se comporta como un esclavo dentro del sistema y la tarjeta de desarrollo asumirá el comportamiento de maestro en la comunicación SPI. Se utiliza el Modo 0 SPI, es decir CPOL=0 y CPHA=0.

A continuación, se presenta el esquema propuesto de conexión entre la radio Lora y el microcontrolador.



**Figura 2.5:** Esquema de comunicación SPI para los dispositivos Lora.

**Fuente:** Semtech SX1276 (2016).

## 2.3 Estación base

### 2.3.1 Comunicación serial

La puerta de enlaces de los datos también considera un sistema embebido para la recepción de datos, sin embargo, la tarjeta de desarrollo utilizada en este punto de la red se integra con la finalidad de hacer de intermediario entre el dispositivo Lora y el concentrador de datos. La interfaz que hace el posible esta comunicación es el protocolo SPI y el protocolo serial.

El concepto de comunicación serial, se traduce en enviar y recibir byte de información, un bit a la vez. Este protocolo es asíncrono, por tanto, el puerto puede enviar datos por una línea y transmitir datos por la otra, que resulta bastante conveniente a la hora de poder enviar información desde la puerta de enlaces hacia los nodos transmisores.

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

---

Algunas de las características seriales importantes que fueron necesariamente configuradas se aprecian en la tabla 2.3.

**Tabla 2.3:** *Parámetros comunicación serial.*

Parámetros	valor
Tasa en baudio:	9600
Bit de datos	8
Bit de paro	no
Bit de parida	par

La interfaz serial, permite que el controlador pueda ser configurado por software con la finalidad de escribir una secuencia de bytes conocida y dar lectura de los datos en receptor en un orden predeterminado.

### 2.3.2 Protocolo Frontend/Backend

Uno de los requisitos necesarios para dar cumplimiento a las especificaciones impuestas por el proyecto es la incorporación de un sistema de almacenamiento local con la finalidad de dar soporte en el supuesto que exista pérdida de conexión entre la puerta de enlaces y el servidor.

Con el objetivo de dar una solución que sea de fácil implementación y que agregue al sistema características que den soporte a las exigencias, se ha introducido un sistema de gestión de base de datos del tipo relacional, publicado bajo la licencia PostgreSQL.

PostgreSQL utiliza un protocolo basado en mensajes para la comunicación entre frontend y backends. El frontend es la parte del software que interactúa con el usuario, por otro lado tenemos los componentes backends que incluyen la base de datos u otros medios de almacenamientos de información [19]. El protocolo está implementado sobre TCP/IP y también sobre Unix sockets [20].

Para esto fue necesario implementar una aplicación frontend capaz de realizar una autenticación y un backend que respondería al intercambio de paquetes de consulta y resultados.

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

---

### 2.4 Servidores de red

La finalidad de la adquisición de datos es poder tener acceso a ellos remotamente, y en se sentido es necesario incluir una interfaz que permita esta comunicación entre la estación base y cualquier dispositivo.

#### 2.4.1 Protocolo IEEE 802.11 n

El diseño del prototipo considera dar soporte a la transmisión de grandes volúmenes de datos a altas velocidades, esta conexión involucra la puerta de enlaces y cualquier otro medio de red.

La disposición física del sistema adquisición en sectores rurales obliga a tomar en cuenta la distancia entre los puntos de red, ese sentido, se propuesto garantizar un mínimo de 2 kilómetros en línea recta. Con el objetivo de cumplir con las especificaciones, se ha implementado el estándar de transmisión inalámbrica IEEE 802.11 n.

Este estándar puede trabajar en la banda de 2.4 GHz y 5 GHz, y garantiza una velocidad mínima 54 Mbps y como máximo 600 Mbps de datos sin procesar, la capa física de los equipos seleccionado soporta una velocidad de 300 Mbps y pueden establecer un enlace máximo de 15 km [21].

#### 2.4.2 WebSocket

WebSocket es relativamente una nueva tecnología que comenzó siendo parte de la especificación del HTML5. Está diseñado para proporcionar un canal de comunicación bidireccional, persistente y full-dúplex sobre un único socket TCP. Es implementado en navegadores y servidores web, pero puede ser utilizado para cualquier aplicación cliente / servidor.

Esta tecnología ha sido incluida obedeciendo a las características necesarias que han sido expuestas en las condiciones de frontera del proyecto y que hacen posible la conexión con los servicios de internet.

Las metas conseguidas con esta tecnología son:

1. Permitir a cada lado, cliente o servidor, transmitir información en cualquier momento.
2. Provee canales de comunicación full-duplex sobre una única conexión TCP.

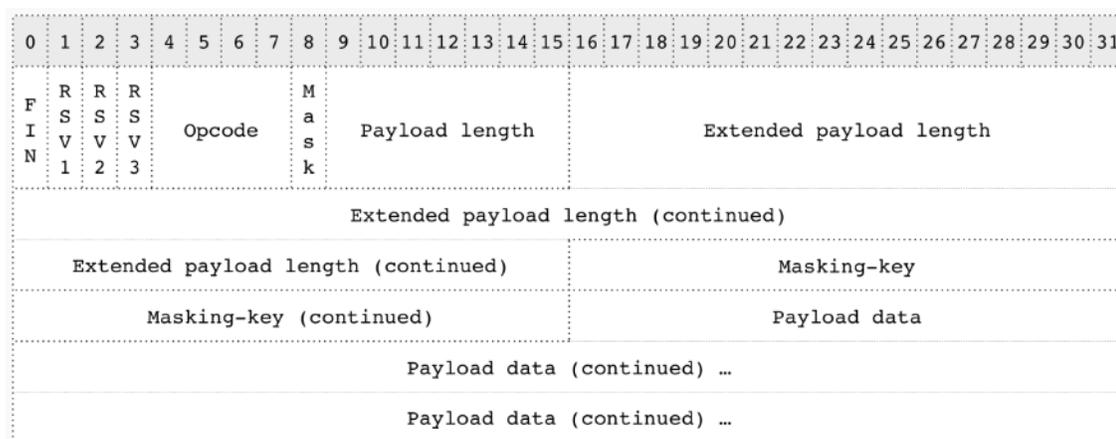
Las comunicaciones se realizan a través de los mismos puertos que utiliza HTTP con el fin de ofrecer compatibilidad con el software HTTP del lado del servidor ya existente. Es decir, cuando

Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

el protocolo trabaja directamente sobre TCP utiliza el puerto 80 y cuando lo hace sobre TLS utiliza el 443. No obstante, WebSocket es un protocolo independiente [22].

El protocolo se divide en dos partes: la negociación y la transferencia de datos. Para establecer una conexión WebSocket, el cliente manda una petición de negociación WebSocket, y el servidor manda una respuesta de negociación WebSocket. Una vez establecida, las tramas WebSocket de datos pueden empezar a enviarse en ambos sentidos entre el cliente y el servidor en modo full-dúplex.

La Figura 2.6. muestra una visión general de alto nivel de la estructura de las tramas.



**Figura 2.6:** Estructura de una trama WebSocket.

**Fuente:** <https://victordiaz.me/websocket>.

WebSocket se puede utilizar en prácticamente cualquier plataforma y consta de implementación en otros ámbitos fuera del navegador web. Dentro de la aplicación propuesta el protocolo es utilizado para enviar los datos a los usuarios finales, quienes tomarán los datos para procesarlos y así poder realizar acciones frente a los indicadores que resulten del análisis.

En lo que respecta a la transmisión solo es necesario considerar las funciones y librerías de los compiladores que permiten generar la aplicación, para establecer la conexión y el formato en el que debe estructurar los datos para ser transmitidos.

En lo que respecta al formato se ha utilizado la estructura JSON, que es un formato liviano para el intercambio de datos. Es fácil de leer, tanto para humanos como para computadoras. Está basado en un subconjunto de JavaScript, JSON se ha establecido para el intercambio de datos en la puerta de enlaces y la Web, por su simplicidad y facilidad de lectura.

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

---

La aplicación consta de un servidor alojado en la web, el cual servirá de datos a las APIs que realizarán el procesamiento necesario para el sistema de toma de decisiones, La puerta de enlaces mencionada en la Figura 2.1, tomará el rol de cliente en la comunicación.

### **2.4.3 Protocolo SSL**

Son muchas las aplicaciones que potencialmente pueden acceder a los datos, situación que trae ventajas a hora de hacer disponible la información, pero que abre la brecha a intrusos que eventualmente pueden hacer uso no autorizado de los datos, En ese sentido, se hace necesario establecer comunicaciones seguras a través de la red.

Un protocolo diseñado para que las aplicaciones puedan transmitir información segura es SSL, que significa Secure Sockets Layer.

Utilizando un certificado SSL, los datos son codificados por códigos de cifrado que recurren a una llave de seguridad de 128 bits de longitud (clave pública / clave privada), y que únicamente conocen el servidor que proporciona la conexión y el dispositivo del usuario, garantizando la protección de la información [21].

.

### **2.5 Protocolo de acceso al medio**

El protocolo de comunicación MAC (Media Access Control) es una pieza fundamental que sirve de base para que otros protocolos de capas superiores puedan comunicar datos en una red de sensores inalámbricos.

Una de las funciones principales de este tipo de protocolos es gestionar el acceso al medio compartido que en nuestra red de sensores será la banda de radio 868 mHz, con el fin de evitar las interferencias entre transmisores, mitigando posibles encolamientos en el nodo central.

Dentro de las consideraciones que se han tomado para el diseño del protocolo tenemos las siguientes

1. Eficiencia: El protocolo debe ser eficiente frente exigencias de trabajo en tiempo real, fiables frente alteraciones de comportamiento de los nodos y robusto frente a interferencias del medio.

Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

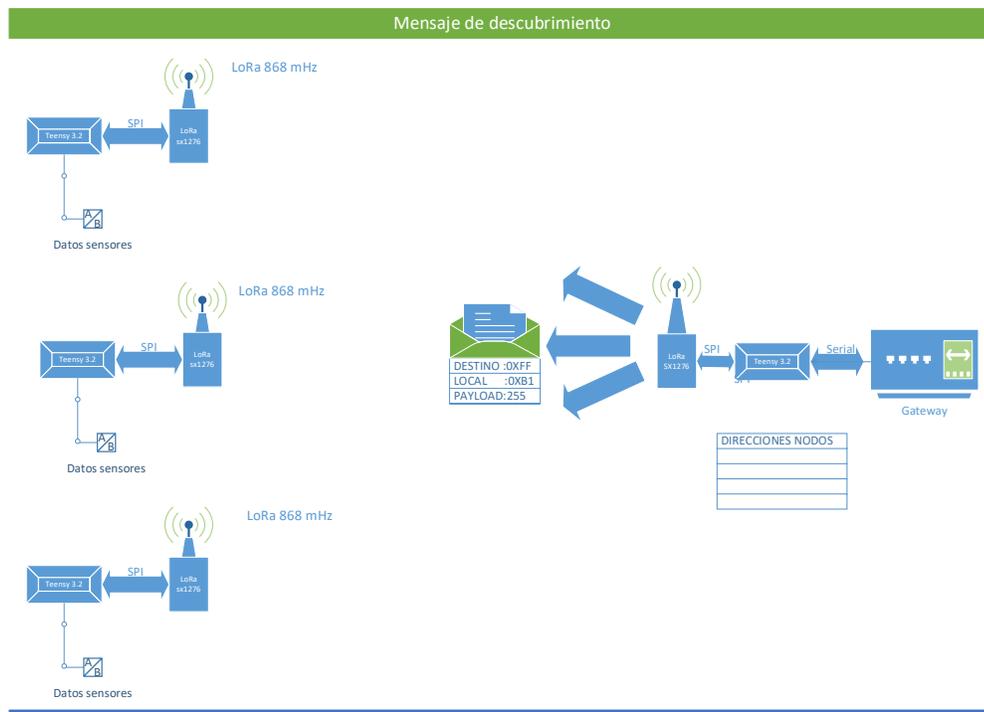
2. Flexibilidad: El sistema debe tolerar la movilidad física de los nodos presentes, es decir, cambios en el número de nodos y soportar posibles incrementos en el tráfico de red.

En base a lo anterior se ha desarrollado un protocolo de comunicación asociado a la transmisión libre de contienda y que utiliza una técnica de acceso determinista basado en PCF(Point Coordination Función ).

**2.5.1 Protocolo de transferencia de información en la red**

El procedimiento consiste en la realización de un sondeo, por parte del gestor de la comunicación y que se encuentra ubicado en la puerta de enlaces del sistema.

El punto coordinador enviará un mensaje de descubrimiento con la finalidad de obtener una respuesta de los nodos presentes, quienes responderán con una trama especifica según la secuencia mostrada en la Figura 2.7.



**Figura 2.7:** Fase de descubrimiento de la red.

En principio una vez que se ha encendido una estación, los nodos se encuentran a la espera del mensaje de descubrimiento del coordinador, recibido el mensaje este envía su respuesta y su dirección es almacenada por el nodo central.

Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

El paquete datos que es enviado por la puerta de enlaces es una versión reducida del formato de los nodos transmisores y solo es utilizado para entregar directrices de funcionamiento de acuerdo con el contenido del payload.

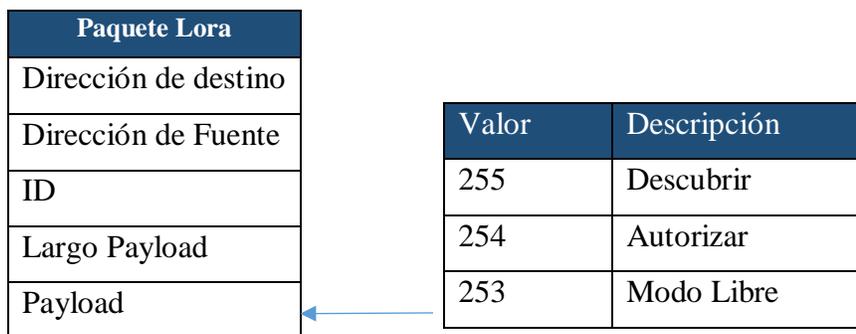


Figura 2.8: Formato de paquete de puerta de enlaces.

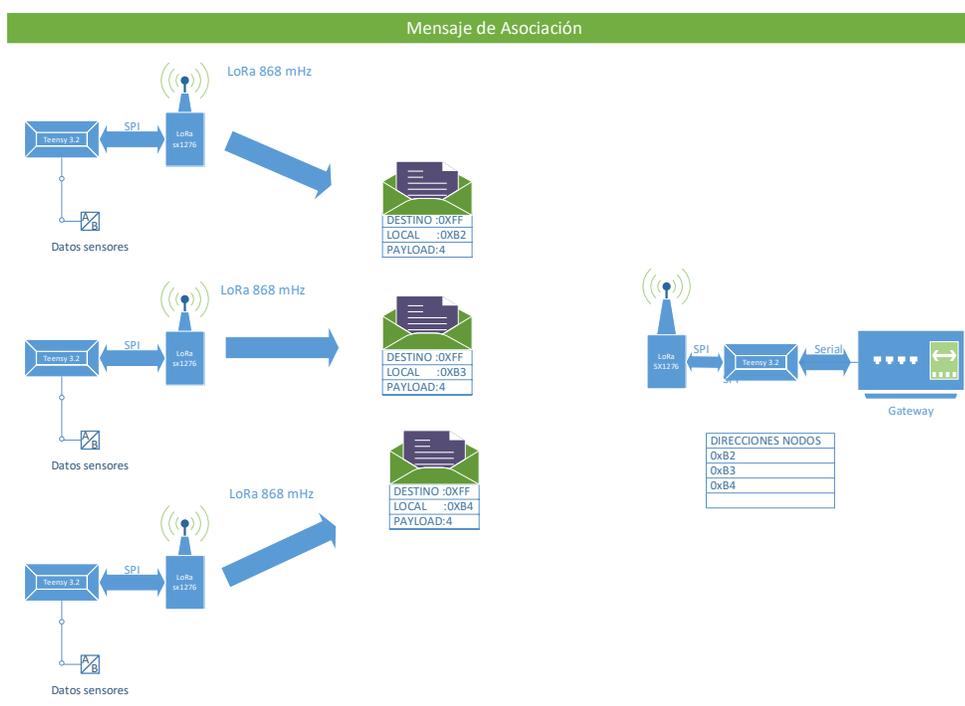


Figura 2.9: Fase de autenticación y asociación.

La estación central sólo almacena direcciones diferentes y que contienen el identificador de respuesta asociado al mensaje de respuesta.

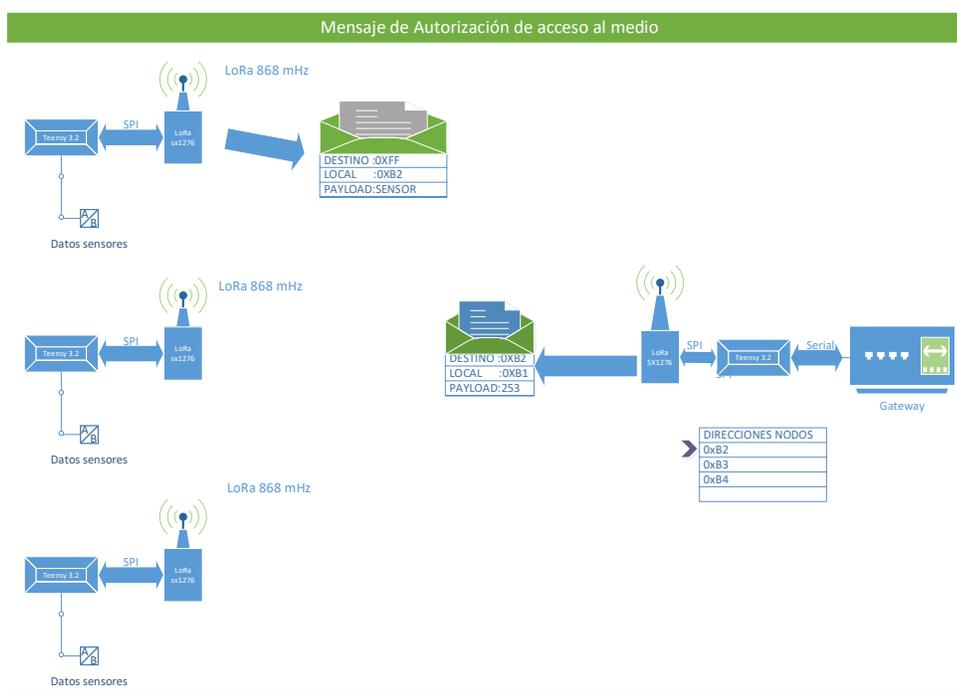
Se enviarán 3 mensajes de descubrimiento, si el número de nodos encontrados es igual a 0, se continuará a la búsqueda hasta que al menos uno sea registrado.

Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

Con los nodos registrados, se iniciará una secuencia de transmisión de mensajes a destinos que han sido previamente almacenados, con la instrucción de enviar datos de sensores. Es necesario precisar que existe un tiempo de activación ( $T_a$ ), que determina el tiempo que existe entre la transmisión de un mensaje de autorización y otro.

El nodo sensor al recibir la orden de enviar datos, enviará el paquete de datos de sensores. La Figura 2.9 muestra el proceso de autorización para acceder al medio.

El ciclo se realiza un número determinado de veces, hasta que el coordinador decida desasociar los nodos registrados y reiniciar el proceso de búsqueda de nodos activos.



**Figura 2.10:** Fase de transmisión de datos útiles para el sistema de adquisición.

Este proceso se realiza con el fin de actualizar la tabla de direcciones con el objetivo de agregar nuevos nodos o eliminar aquellos presumiblemente inactivos.

Cabe señalar que existen 3 tipos de mensajes que pueden enviar los nodos transmisores, existiendo la posibilidad de agregar otros, de acuerdo con las necesidades del cliente. Este tipo de mensajes está determinado de acuerdo al OCP (Opciones de configuración de paquete), Estas opciones determinan si el paquete representa un contenido comprimido (OCP=2), sin comprimir (OCP=3) o simplemente una respuesta al mensaje de descubrimiento (OCP=1).

## Capítulo 2: Teoría de protocolos y tecnologías de comunicación.

Existe la posibilidad de cambiar el protocolo de comunicación. Para esta situación el servidor debe enviar un paquete con la configuración adecuada, (ver tabla 2.5) y que permite configurar los nodos transmisores para que envíen la información en broadcast sin ningún tipo de autorización.

**Tabla 2.4:** Paquete configuración de puerta de enlaces.

Paquete Configuración	Significado
OGG	1 Cambio en la configuración Gateway 0 No existen cambios
Tiempo Activación	Tiempo entre una autorización y otra Configurable de 0-10
Tasa de refresco	Tasa de refresco, Define el número de mensajes enviados en antes de actualizar las tablas de registro de nodos (Multiplicador de 0 a 100)
Modo de operación	1 Modo Libre para transmitir 0 Modo Protocolo de transferencia de información (defecto)
Opcional	Byte de reserva para otro tipo de configuraciones

### 2.6 Conclusión

Un desafío importante es definir dentro de las tecnologías existentes la más adecuada para establecer la comunicación bajo los requisitos que ha impuesto el proyecto Fondef y que están estrechamente ligados a las condiciones de sitio, almacenamiento y seguridad en la transmisión de los datos.

Cada protocolo y tecnología utilizada cumple con los requisitos descritos en el capítulo introducción, garantizando de extremo a extremo la comunicación, redundancia en el almacenamiento de los datos y sencillez en la implementación, logrando que el prototipo sea viable en su construcción al existir equipos en el mercado que implementan los protocolos descritos.

El protocolo de acceso propuesto fue diseñado para coordinar una red de nodos transmisores móviles, dinamismo que es determinado por las condiciones de trabajo del medio en que deben estar desempeñando funciones, situación que genera que el nodo cambie de ubicación en el tiempo y que eventualmente pudiese variar la cantidad de elementos de la red de sensores inalámbricos.

## CAPÍTULO 3. DESCRIPCIÓN DE LA PROPUESTA DE RED

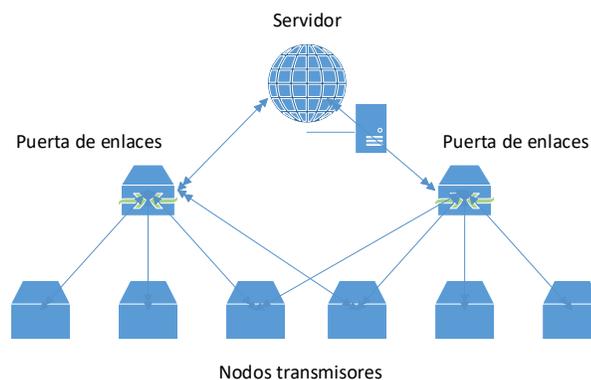
### 3.1 Introducción

Una red de comunicaciones proporciona las vías necesarias para establecer interconexiones cuando estas sean necesarias con el fin de transportar la información, para esto es necesario definir la organización que tendrán los recursos necesarios para realizarla, siendo estos los sistemas de transmisión y métodos de acceso, así como las informaciones relativas al establecimiento de los enlaces.

Las estructuras o topologías de red son las diferentes formas en las que podemos organizar la red, en síntesis, con esto nos referimos al emplazamiento geométrico en el cual estarán distribuidos los nodos y los medios físicos que harán posible la conexión. El objetivo de diseño es buscar la forma más económica y eficaz de conexión, pero, al mismo tiempo, aumentar la fiabilidad del sistema, evitar los tiempos de espera en la transmisión, permitir un mejor control de la red y lograr de forma eficiente el aumento del número de nodos transmisores.

La topología de red está asociada a una topología física y una topología lógica. La primera guarda relación en la disposición que tendrán los elementos de red. La topología lógica es un conjunto de reglas normalmente asociado a una topología física, que define el modo en el que se gestiona la transmisión de los datos en la red. La utilización de una topología influye en el flujo de información (velocidad de transmisión, tiempos de llegada, etc.), en el control de la red, y en la forma en la que ésta se puede expandir y actualizar.

El sistema propuesto considera una topología del tipo red celular como la que se aprecia en la Figura 3.1.



**Figura 3.1:** Topología de red tipo celular propuesta.

### Capítulo 3: Descripción de la propuesta de red.

---

En este tipo de red, se pretende que todos los nodos inalámbricos envíen la información a un nodo central y este concentrador, transmita la información a los servicios alojados en la web. Esta topología permite que eventualmente un nodo pueda registrar información en dos puertas de enlaces distintas.

El sistema de adquisición de datos involucra directamente al concepto del Internet de las Cosas, sin embargo, para dar soporte a esta idea es necesario definir todos los equipos y tecnologías que deben estar presente.

Es claro que para dar solución basado en IoT se debe constar con tres estructuras que son básicas para hacer un desarrollo con estas características, siendo estos sensores, puerta de enlaces y un bus que permita la integración del todo.

Una vez resuelto el problema de las comunicaciones, definidas las etapas, protocolo y tecnologías que tendrá el sistema, es necesario aterrizar la solución a los equipos que son compatibles con el esquema propuesto. Se sugiere utilizar los siguientes equipos que serán descritos a continuación.

#### **3.2 Descripción de los dispositivos utilizados**

La red propuesta para resolver el sistema de transmisión de datos consta de nodos sensores, y una puerta de enlaces, que realizara la concentración, almacenamiento y transmisión final de los datos, para su procesamiento.

La puerta de enlaces será responsable de coordinar el acceso al medio, ejecutando el protocolo de comunicación expuesto en el Capítulo 2.

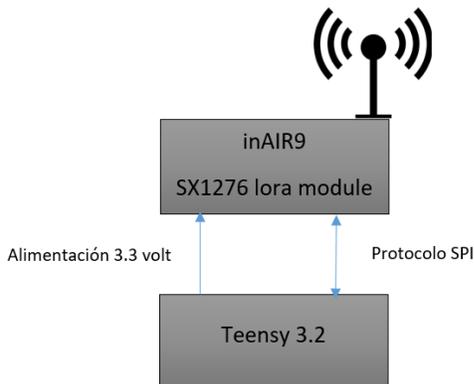
##### **3.2.1 Nodos inalámbricos**

Para los nodos inalámbricos se propone el uso del dispositivo inAir9 con la tecnología Lora, propiedad de la corporación Semtech [15], que utiliza el chip SX1276. Este dispositivo requiere de la lógica de programación para ser configurado. Por tanto, debe incorporar una tarjeta de desarrollo con funciones específicas para controlar el módulo Lora, Se propone el uso de la tarjeta Teensy 3.2.

La Figura 3.2 presenta la disposición de los dispositivos involucrados tanto en el sistema embebido destinado transmisión de los datos desde el sistema de adquisición y la del sistema de recepción en la puerta de enlace destinada para concentrar los datos de los distintos nodos. Con la configuración

## Capítulo 3: Descripción de la propuesta de red.

señalada es posible dar funcionalidad a los dispositivos Lora.



**Figura 3.2:** Disposición de dispositivos para la creación del nodo inalámbrico.

### 3.2.1.1 Teensy 3.2

Teensy 3.2 es una tarjeta de desarrollo nacida del crowdfunding mediante Kickstarter [22]. Fue seleccionada por concentrar una gran potencia de cálculo en un espacio acotado, característica importante a la hora de contar con espacio reducido para su integración, debido a que el sistema de adquisición se encuentra empotrado en el área interna, específicamente en la base de la de caja cosechera.

Esta tarjeta de desarrollo está basada en un procesador ARM Cortex M4 de 32 bits. Está implementada en forma de placa para su montaje en protoboards, sin embargo, se consideró soldar los circuitos directamente a la placa, con el objetivo de disminuir considerablemente la profundidad de la instalación.

Se encuentra pregrabada con un bootloader que da la posibilidad de ser programada directamente con el IDE de Arduino, característica que hace posible la compatibilidad con la mayoría de las librerías que utiliza la plataforma.

La Figura 3.3 presenta una fotografía de la placa Teensy 3.2.



**Figura 3.3:** Placa Teensy 3.2.

**Fuente:** [https://www.pjrc.com/store/teensy32\\_pins.html](https://www.pjrc.com/store/teensy32_pins.html).

Capítulo 3: Descripción de la propuesta de red.

En lo que respecta a la integración al transmisor Lora, cuenta con el hardware necesario para implementar el protocolo SPI , en la Figura número 3.4 se presenta el diagrama de pines la tarjeta Teensy.

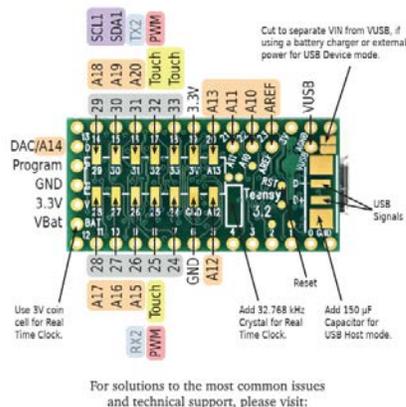


Figura 3.4: Diagrama de Pines Teensy 3.2.

Fuente: [https://www.pjrc.com/store/teensy32\\_pins.html](https://www.pjrc.com/store/teensy32_pins.html).

En la Tabla 3.1 se describe la tabla de correspondencia para realizar el conexionado.

Tabla 3.1: Tabla de conexionado Lora-Teensy 3.2.

Semtech SX1276	Teensy 3.2	Descripción
VCC	3.3 v	Voltaje de alimentación
GND	GND	Referencia de tierra
SCK	13	Pulso que marca la sincronización
MISO	11	Salida de datos del Esclavo y entrada al Master
MOSI	12	Salida de datos del Master y entrada de datos al Esclavo
NSS	10	Utilizado para la activación del esclavo
NRESET	90	Usado para funciones de interrupción
DIO0	2	Necesario para recibir el modo de devolución de llamada

### Capítulo 3: Descripción de la propuesta de red.

---

Dentro de las características principales de hardware de la tarjeta Teensy 3.2 tenemos las siguientes.

1. Procesador: 32 bit ARM Cortex-M4 72 MHz CPU (MK20DX256VLH7)
2. Memoria: 256K, 64K RAM, 2K EEPROM
3. 14 pines analógicos de alta resolución (13 bits usables y 16 bit en hardware)
4. 34 pines digitales I/O (10 son compartidos con los analógicos)
5. 3 puertos UARTs (serie)
6. Protocolos soportados: SPI, I2C, I2S, modulador IR
7. Soporta RTC añadiendo un cristal de 32.768 KHz y batería externa
8. Dimensiones: 35x1

La tarjeta Teensy dará soporte para el procesamiento inicial de los datos. Dentro de las funciones que desempeñara se distinguen las siguientes.

#### A. Nodo transmisor

1. Adquisición de variables analógicas
2. Orden y estructuración de datos
3. Compresión de datos
4. Construcción del paquete Lora

#### B. Nodo receptor

1. Orden y estructuración del paquete lora recibido
2. Transmisión serial de los datos a la puerta de enlaces
3. Ejecución de protocolo de comunicación

#### **3.2.1.2 inAir9**

El inAir9 es un módulo inalámbrico que usa el chip Semtech SX1276 [24], Este módulo forma parte del sistema de transmisión y recepción. Estos módulos utilizan la tecnología Lora, tienen la capacidad de comunicarse con la topología M2M (Máquina a Máquina), Pueden ser utilizados en

Capítulo 3: Descripción de la propuesta de red.

complejas redes de comunicación, de largo alcance, con alta tolerancia a las interferencias y un bajo consumo energético.

El chip SX1276 permite la comunicación inalámbrica a distancias de hasta 15 km, utilizando la tecnología Lora (Long Range). Gracias a que incorpora el módem de espectro ensanchado, en modo Lora es capaz de proveer una distancia de comunicación significativamente mayor que los sistemas basados en modulación FSK y OOK [14].

En la Tabla 3.2 se señalan las características del dispositivo.

**Tabla 3.2:** Características del dispositivo.

Fuente: Semtech SX1276 (2016).

Rango de frecuencia	Factor de ensanchamiento	Ancho de Banda	Velocidad transmisión	Sensibilidad estimada
137-1020 MHZ	6-12	7.8 -500 KHz	0.018-37.5 kbps	-111 to -148 dBm

Existen frecuencias de libre uso conocidas como ISM (Industrial, Scientific and Medical) que son frecuencias intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico. La Tabla 3.3 presenta las principales frecuencias ISM compatibles con los dispositivos Lora propuestos.

**Tabla 3.3:** Banda de frecuencia ISM compatibles con el módulo inAIR9 de radio propuestos.

Fuente: Semtech SX1276 (2016).

Banda ISM (Mhz)	Frecuencia Mínima (MHz)	Frecuencia Máxima (MHz)	Ancho de Banda	Uso por continente
433	433.05	434.79	1.84	Europa, África y parte de del norte de Asia
869	868	870	2	Europa, África, Asia y Oceanía
900	902	928	26	Continente americano

De acuerdo con la Tabla 3.3, la frecuencia seleccionada para configurar el inAir9 es 868 MHz

### Capítulo 3: Descripción de la propuesta de red.

Las principales características técnicas se muestran en la Tabla 3.4.

**Tabla 3.4:** Características técnicas del módulo SX1276.

**Fuente:** Semtech SX1276 (2016).

Característica	Máximos absolutos		Rango de operación		Unidad
	Min	Max	Min	Max	
Voltaje de alimentación	-0.5	3.9	1.8	3.7	V
Temperatura	-55	+115	-40	+85	°C
Nivel entrada RF	-	+10	-	+10	dBm
Temperatura Juntura	-	+125	-	-	°C
Carga capacitiva puertos digitales	-	-		25	pF

Con respecto a la operación del módulo en el modo Lora, se tienen las siguientes características

1. Voltaje de alimentación = 3.3V
2. Temperatura = 25°C
3. Frecuencia OSC = 32MHz
4. Ancho de banda (BW) = 125 kHz
5. Factor de ensanchamiento (SF) = 7-12
6. Código de corrección de errores (EC)=4/6
7. Porcentaje de error en paquetes (PER) = 1%
8. Verificación de CRC en el área de datos habilitada
9. Potencia de salida = 13dBm en transmisión (máximo configurable)
10. Tamaño de área de datos = 255 bytes (máximo)
11. Tamaño del preámbulo = 12 símbolos (registro programado PreambleLength=8)

Capítulo 3: Descripción de la propuesta de red.

**3.2.1.3 Consideraciones eléctricas nodos transmisores**

Se ha sugerido para los nodos sensores, utilizar una batería que se puede apreciar en la Figura 3.5.



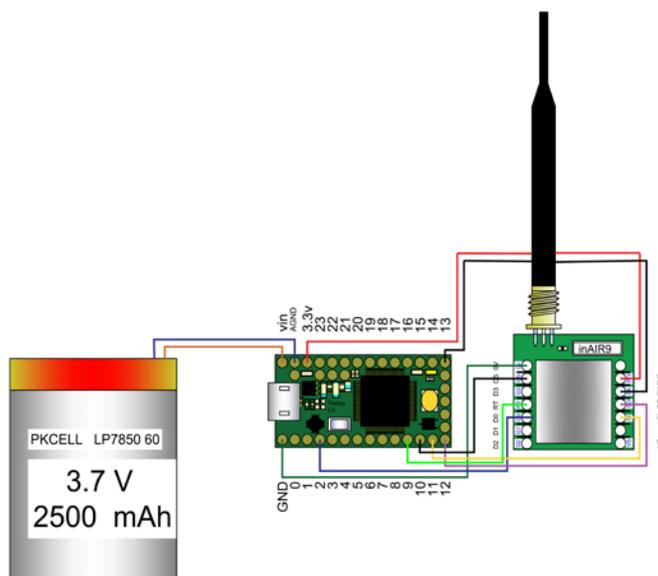
**Figura 3.5:** Soundon New Energy® Batería de Polímero de Litio.

**Fuente:** Pc Factory, accesorios electrónica.

Esta batería tiene una capacidad de 2500mAh con lo que se tiene un total de 10Wh. Este tipo de batería recargable permitiría una autonomía de funcionamiento a los nodos sensores de 200 horas, considerando que en régimen permanente el consumo de corriente es de 12.5 mA según las mediciones realizadas en laboratorio.

**3.2.1.4 Integración**

Con los antecedentes descritos anteriormente, el nodo sensor y su conexionado se presenta en la Figura 3.8. Esta aplicación es utilizada para la captura de datos irrá inmersa en la caja de recolección de frutas.



**Figura 3.6:** Conexionado nodo sensor.

## Capítulo 3: Descripción de la propuesta de red.

---

El sistema de transmisión de datos será instalado en una caja de cosecha diseñada con características que permiten la instalación del nodo y que son parte de las condiciones de fronteras definidas en este documento.

### 3.2.2 Puerta de enlaces

La puerta de enlaces es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino.

La puerta de enlace considerada en la red propuesta, junto con desempeñar funciones asociadas a la interconexión de redes, será responsable de ejecutar el protocolo de comunicación y dar soporte al almacenamiento de datos del sistema.

A continuación, se describen todos equipos que son parte de esta estructura.

#### 3.2.2.1 Raspberry Pi 3

Raspberry Pi, un es un ordenador de tamaño reducido, que soporta varios periféricos que utilizan los ordenadores comunes, tales como mouse, teclado, salida de audio etc.

En todas sus versiones incluye un procesador, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara y una tarjeta MicroSD. En la Figura 3.7 se puede apreciar la Raspberry Pi en su versión 3 B.



**Figura 3.7:** *Raspberry Pi 3 B.*

### Capítulo 3: Descripción de la propuesta de red.

---

Este procesador contendrá varias aplicaciones que realizarán el tratamiento necesario para poder disponer de los datos y que estos sean de fácil acceso para cualquier usuario autorizado.

Dentro de funciones de las aplicaciones que están soportadas se tiene:

1. Adquisición de la data de los nodos inalámbricos
2. Descompresión de datos
3. Respaldo de datos en base de datos del tipo relacional postgres
4. Transmisión de datos al servidor

Dentro de las características técnicas que tiene la Raspberry Pi se pueden mencionar las siguientes

- Procesador: Broadcom BCM2837B0, Cortex-A53 64 bits SoC @ 1.4GHz
- Memoria: 1GB LPDDR2 SDRAM
- Conectividad: LAN inalámbrica de 2,4 GHz y 5 GHz IEEE 802.11.b / g / n / ac, Bluetooth 4.2, BLE. Gigabit Ethernet sobre USB 2.0 (rendimiento máximo de 300 Mbps). 4 puertos USB 2.0
- Soporte de tarjeta SD: formato Micro SD para cargar el sistema operativo y el almacenamiento de datos
- Potencia de entrada: 5V / 2.5A DC a través del conector micro USB. 5V DC a través del encabezado GPIO. Potencia a través de Ethernet (PoE) habilitada (requiere un HAT de PoE independiente)
- Temperatura de funcionamiento, 0-50 ° C

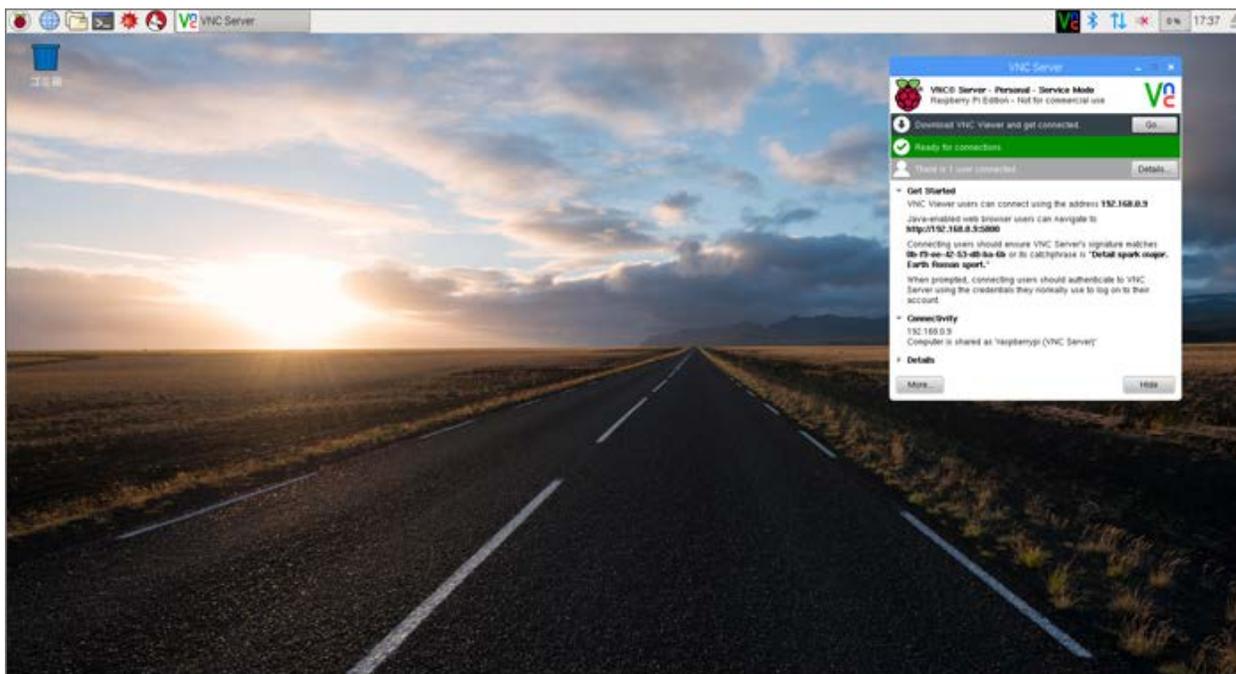
El sistema operativo utilizado es una versión de Linux basada en Debian, denominada Raspbian. Sobre este sistema es desarrollado el código necesario para la gestión de datos recibidos sobre la plataforma del software Qt Creator y se genera la base de datos destinada al almacenamiento local utilizando el software PostgreSQL.

La versión instalada es PIXEL (Pi Improved Xwindows Environment Lightweight) que es el nuevo entorno gráfico de Raspbian, incluye software educativo, para programación y para uso general, como un cliente de email, un navegador web o incluso la suite de ofimática LibreOffice. Es posible

## Capítulo 3: Descripción de la propuesta de red.

utilizarla como ordenador de sobremesa agregando periféricos, de esta forma un usuario que no tenga mayor conocimiento en Linux puede cambiar de ser necesario alguna configuración o agregar software.

La Figura 3.8 muestra una captura del sistema operativo instalado.



**Figura 3.8:** Captura de pantalla sistema operativo Raspbian PIXEL.

### 3.2.2.2 TP link CPE510

La disposición geográfica que tendrán los equipos en la red supone que los datos deben ser transmitidos a largas distancias, por tanto, resulta plausible concentrar los datos en un solo punto de transmisión y generar un enlace único capaz de transportar la información.

Dentro del esquema presentado, se incluye un punto de acceso para exteriores CPE 510 de la línea Tp Link Pharos. Este equipo será responsable de transmitir la información desde la puerta de enlaces al servidor principal. Dentro de las características principales tenemos las siguientes.

- Procesador Qualcomm AR9350B con una velocidad de reloj de 560MHz,
- Memoria RAM DDR de 64MB de capacidad, y una
- Memoria FLASH con 8MB de capacidad.
- un puerto Fast-Ethernet blindado con PoE pasivo de entrada
- Fast-Ethernet blindado con PoE Passthrough.

Capítulo 3: Descripción de la propuesta de red.

En la Figura 3.11 se aprecia el CPE510.



Figura 3.9: TP Link CPE510.

Este equipo tiene un alcance de 15 km para las transmisiones e incorpora un software de gestión centralizada, con el que se es capaz de monitorear todos los dispositivos desde un solo ordenador. El desarrollo contempla una configuración punto a multipunto, según lo descrito en la Figura 3.12.

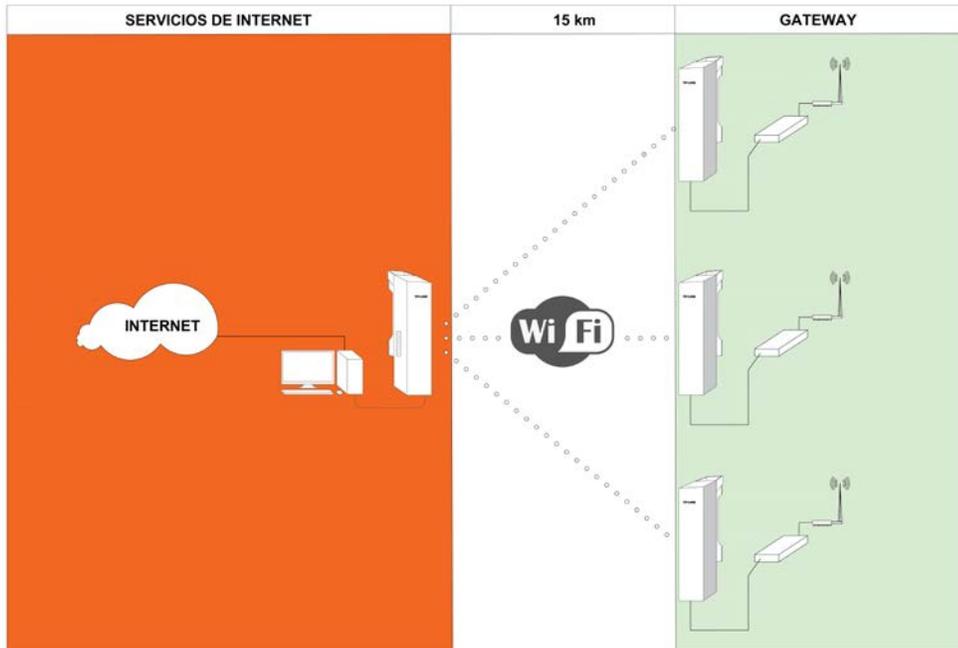


Figura 3.10: Topología estrella de transmisores finales.

Capítulo 3: Descripción de la propuesta de red.

3.2.2.3 Consideraciones eléctricas

Para la alimentación eléctrica del nodo central se ha considerado una fuente de alimentación basada en la inclusión de batería Ultracell Ciclo Profundo 7 ah, adicionalmente se consideró un inversor de tensión de 12 volt CC a 220 volt CA para la alimentación de la Raspberry Pi y el transmisor CPE510.



Figura 3.11: Batería Ultracell Ciclo Profundo 7ah.

En la Figura 3.12, se presenta el diagrama unilineal de fuerza del sistema.

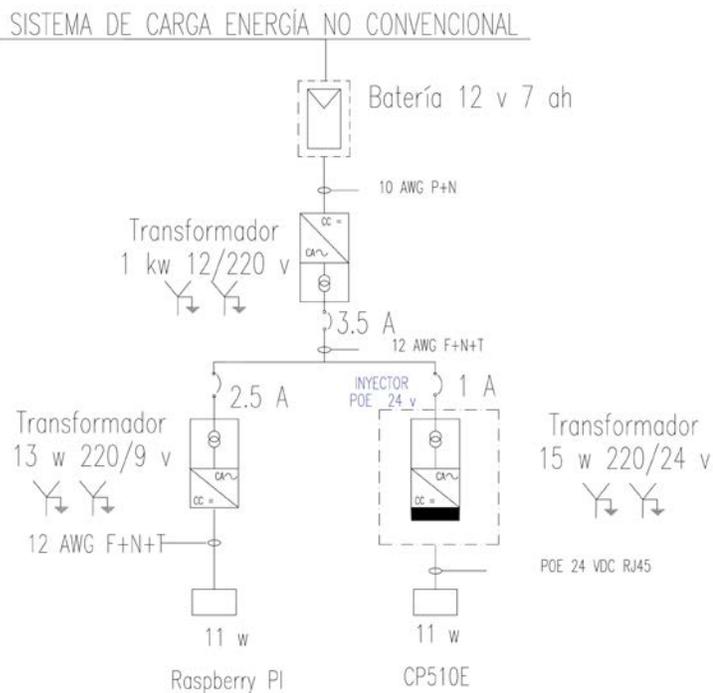
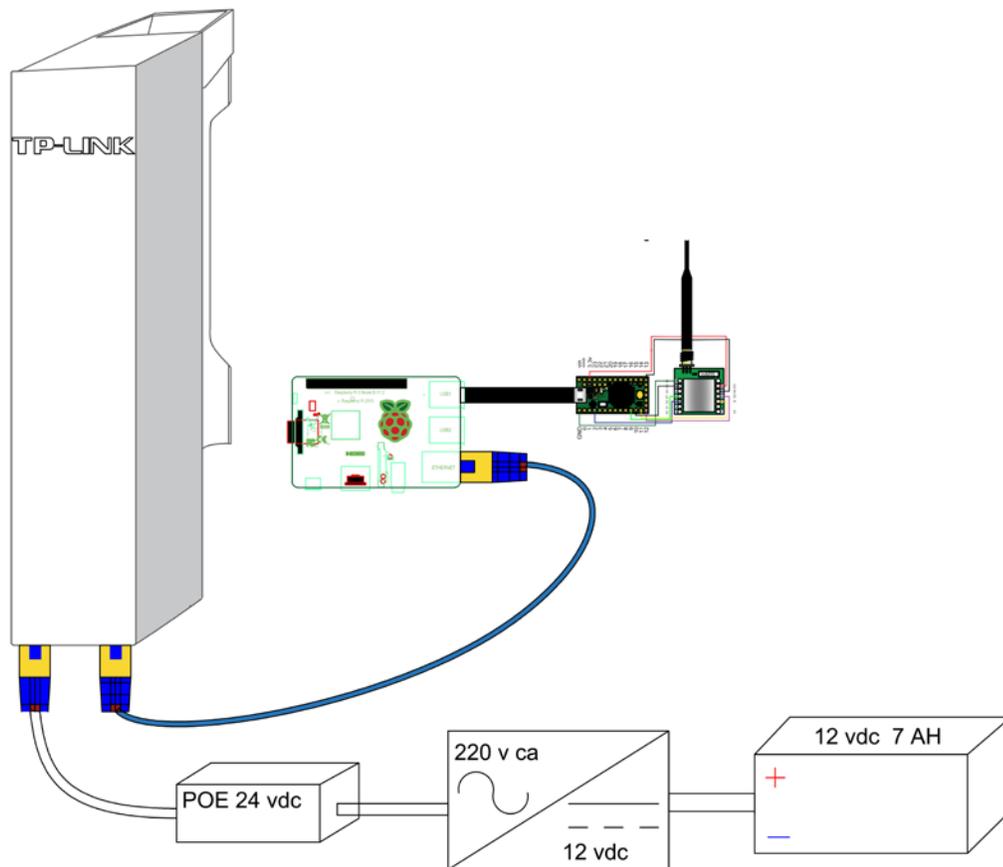


Figura 3.12: Diagrama unilineal de fuerza.

### Capítulo 3: Descripción de la propuesta de red.

#### 3.2.2.4 Integración nodo central

Con los antecedentes descritos anteriormente, la puerta de enlaces y su conexionado se presenta en la Figura 3.13.



**Figura 3.13:** *Conexionado de equipos que forman parte de la puerta de enlaces.*

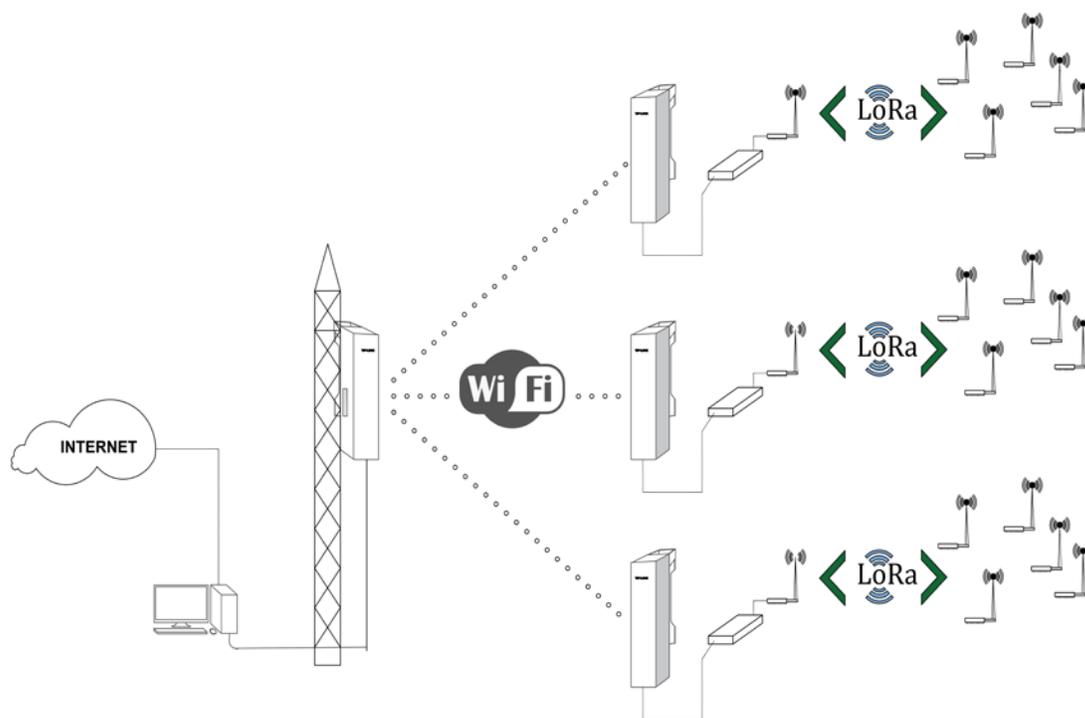
En la Figura 3.13, se encuentra el detalle de conexionado de la puerta de enlaces, aplicación necesaria para gestionar el tráfico de la información y su posterior almacenamiento en la base de datos.

La información es transmitida al servidor central, donde fue considerado un tp link CPE510, capaz de lograr transmisiones de hasta 15 km y diseñado para aplicaciones Outdoor .

## Capítulo 3: Descripción de la propuesta de red.

### 3.3 Estructura de red

Una vez descritos todos los elementos que forman parte de la estructura propuesta es posible presentar el diseño en la Figura 3.14.



**Figura 3.14:** Estructura de red propuesta.

### 3.4 Conclusiones

Definida la estructura de la red fue posible adquirir los equipos necesarios para construir el esquema propuesto. Las tarjetas de desarrollo Raspberry Pi y Teensy 3.2 son dispositivos que ofrecen un espectro amplio de posibilidades para el desarrollo de aplicaciones de todo tipo, debido a su gran poder de cómputo, su versatilidad, interfaces intuitivas y que por lo demás se puede adquirir a precios bastante reducidos. Existen diversos parámetros que se deben ajustar para poder lograr una gran distancia de transmisión de los dispositivos Lora, tales como una antena de alta ganancia y la elección de la potencia de transmisión. Los CPE510 representan una gran alternativa y de bajo costo para transmitir datos a varios kilómetros de distancia, sin embargo, es necesario considerar un tipo de alimentación eléctrica no convencional por la falta de suministro eléctrico en lugar de la instalación (Sector rural).

## CAPÍTULO 4. DISEÑO Y CONSTRUCCIÓN DEL SOFTWARE DE LA RED

### 4.1 Introducción

La existencia de la estructura de red propuesta no tendría sentido sin los programas que hacen posible la comunicación entre los elementos de la red y los usuarios e incluso la interacción entre ambos.

El software ha trascendido a ser un ente ubicuo en la actual era digital, característica que potencia aún más el desarrollo de plataformas directamente involucradas en redes IoT.

Una plataforma integrada al Internet of Things (IoT) es el software que conecta todo el hardware de una red destinada a la adquisición de datos, sean esto puntos de acceso, nodos sensores y que finalmente a través de una aplicación el usuario puede acceder y hacer uso de los datos adquiridos.

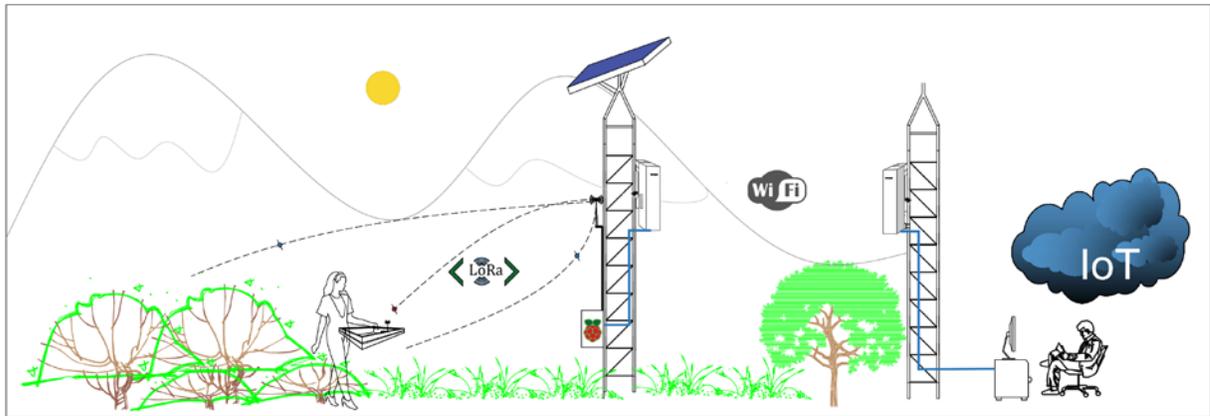
Una plataforma IoT es la base para que dispositivos estén interconectados y permitir el flujo ordenado de datos desde la caja de cosecha hasta la aplicación que procesara la información.

Para la construcción del software necesario para genera la plataforma IoT se consideraron los siguientes ítems.

1. **Conectividad y normalización:** Se debe asegurar que diferentes protocolos y diferentes formatos de datos en una interfaz de “software” se garantice la precisa transmisión de datos y la interacción con todos los dispositivos.
2. **La gestión de dispositivos:** Se debe asegurar que todos los nodos sensores conectados estén funcionando correctamente.
3. **Base de datos:** almacenamiento escalable de datos del dispositivo basados en la nube a un nuevo nivel en términos de volumen de datos, variedad, velocidad y veracidad.
4. **Procesamiento y gestión de la acción:** Aporta datos basados en reglas de acción de evento-disparadores que permitan la ejecución de las acciones “inteligentes” basados en datos específicos del sensor.
5. **Analítica:** llevar a cabo una serie de análisis complejo de la agrupación de datos básicos y de aprendizaje automático.
6. **Visualización:** permite a los seres humanos observar las tendencias mediante de visualización de datos.

## Capítulo 4: Diseño y construcción de software de la red.

El software diseñado responde a una necesidad, donde están involucrados seres humanos y procesos que deben llevarse a cabo para el normal funcionamiento de una empresa con activos destinados a la producción agrícola. En la Figura 4.1 se representa el proceso en estudio, donde se aprecia el lugar donde se adquieren los datos, la transmisión al servidor y las personas que requieren estos para la toma de decisiones.



**Figura 4.1:** Representación gráfica del proceso.

### 4.2 Descripción del software utilizado

El software es actor principal dentro de este desarrollo. Los sistemas operativos utilizados para el desarrollo de las pruebas de los dispositivos Lora, fueron Microsoft Windows 10 y PIXEL. Sobre Windows, se utilizó el IDE (Entorno de Desarrollo integrado) de Arduino, sobre PIXEL se utilizaron las versiones libres de Qt Creator y PostgreSQL, para crear la aplicación puerta de enlaces y la base de datos respectivamente. Desarrollos que forman parte de la red de transmisores que han sido descritos en el capítulo anterior.

El diseño del software está dividido en 3 zonas determinadas por la estructura de red, estas son

- Software red de sensores inalámbricos.
- Software puerta de enlaces.
- Software asociado a la conexión con servicios de internet.

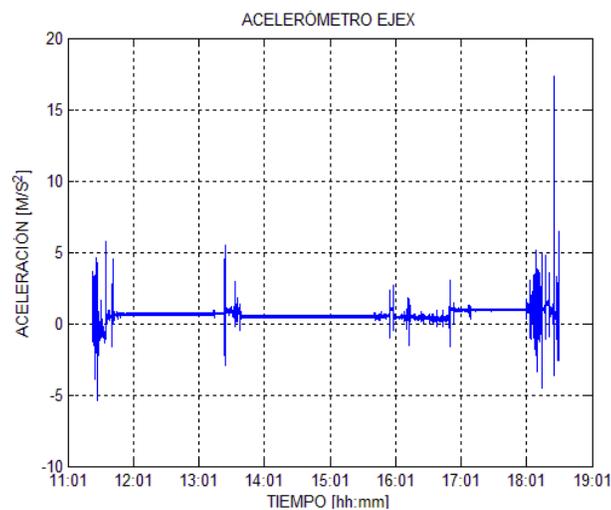
## Capítulo 4: Diseño y construcción de software de la red.

### 4.2.1 Software red de sensores inalámbricos

El diseño del programa que utiliza los nodos transmisores se realizó sobre IDE de Arduino que es un entorno de desarrollo y en él se realiza la programación de cada una de las placas de Teensy de la red propuesta. Es capaz de crear, editar, depurar y transferir código a las diferentes tarjetas que se encuentran soportadas. Tiene como base el entorno de Processing al igual que un lenguaje de programación fundamentado en Wiring [23].

Los nodos transmisores son los responsables de registrar datos provenientes de los sensores que se distribuyen en la caja cosechera, estos sensores registran variables de diferente naturaleza, siendo estas: temperatura, giro, aceleración, humedad, peso y carga batería.

En las gráficas siguientes se pueden apreciar gráficos de datos obtenidos de campañas del proyecto Fondef ID16I10206.

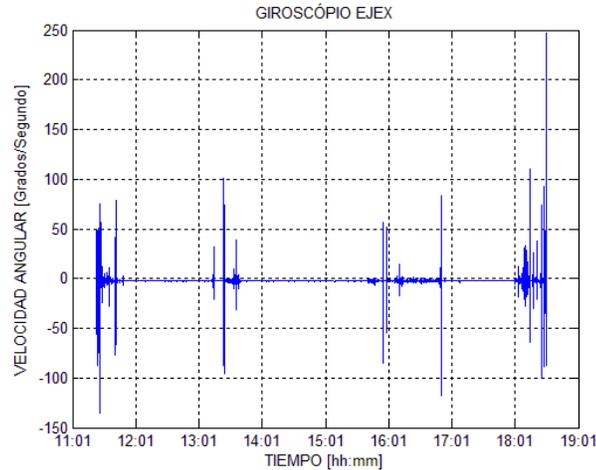


**Figura 4.2:** Gráfico de muestras de aceleración eje x en un periodo de 8 horas.

**Fuente:** Proyecto Fondef ID16I10206, campaña n°2.

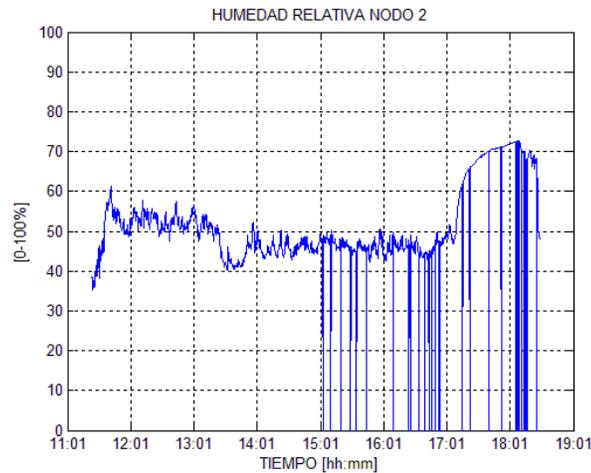
Las actividades realizadas en la campaña de terreno, fueron ejecutadas con el objetivo de evaluar la funcionalidad y características del equipo, “caja recolectora”, durante la recolección de arándanos, en estas pruebas se consiguió evaluar el registro de datos de peso, temperaturas, vibración y posicionamiento de la caja durante cosecha [24].

## Capítulo 4: Diseño y construcción de software de la red.



**Figura 4.3:** Gráfico de muestras de giroscopio eje x en un periodo de 8 horas.

**Fuente:** Proyecto Fondef ID16I10206, campaña n°2.

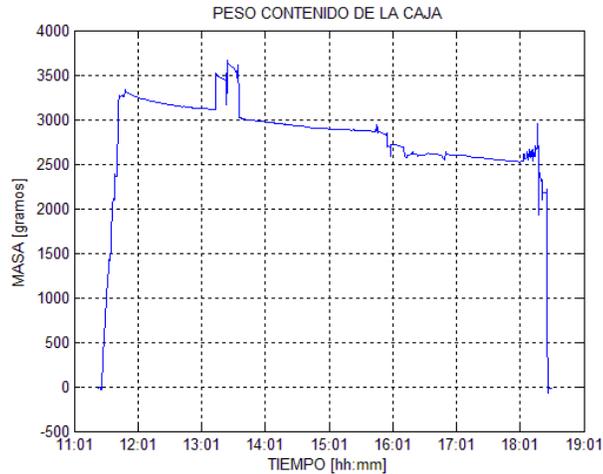


**Figura 4.4:** Gráfico de muestras de aceleración en un periodo de 8 horas.

**Fuente:** Proyecto Fondef ID16I10206, campaña n°2.

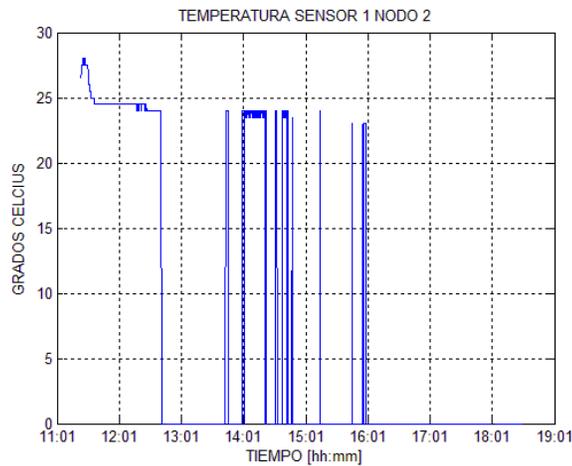
Los datos registrados corresponden a la campaña n°2 ejecutada por el grupo de trabajo encargado de diseñar el dispositivo de adquisición descrito en la frontera 1 y fue realizada en el fundo Santa Luisa ubicado en la localidad de campanario. Estas variables fueron registradas por 3 cajas cosechadoras y almacenados durante periodos de 1 minuto en una microSD durante el transcurso de 8 horas.

Capítulo 4: Diseño y construcción de software de la red.



**Figura 4.5:** Gráfico de muestras de peso en un periodo de 8 horas.

**Fuente:** Proyecto FONDEF ID16110206, campaña n°2.



**Figura 4.6:** Gráfico de muestras de temperatura en un periodo de 8 horas.

**Fuente:** Proyecto Fondef ID16110206, campaña n°2.

Para simular la adquisición de datos se utilizó una aplicación se lee los datos de ficheros que contienen la data de los sensores y los vuelve a escribir en la puerta serial donde son leídos por la aplicación que fue descargad en la tarjeta Teensy.

Capítulo 4: Diseño y construcción de software de la red.

En general el software creado obedece al siguiente diagrama de flujo.

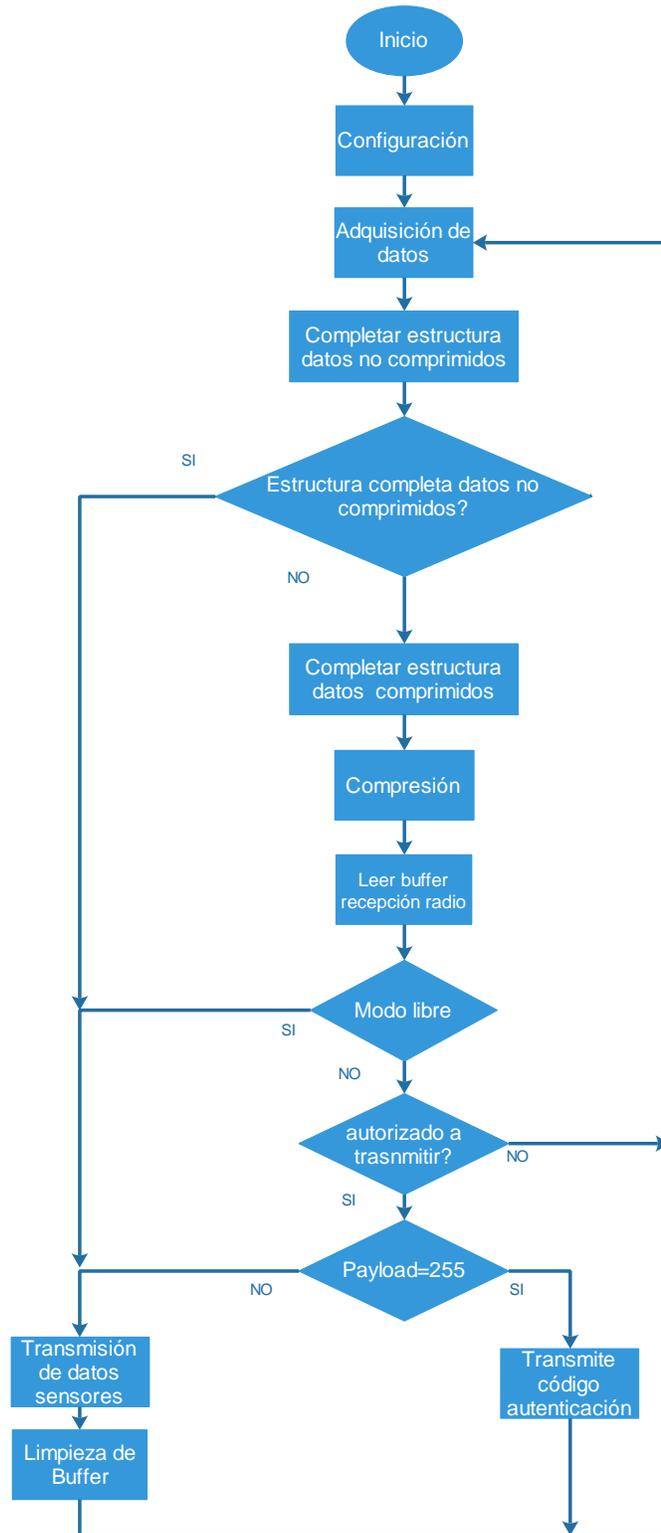


Figura 4.7:Diagrama de flujo del software del nodo transmisor.

## Capítulo 4: Diseño y construcción de software de la red.

---

### 4.2.1.1 Configuración

Para el inicio del programa fue preciso incluir 3 librerías necesarias para el envío y recepción de paquetes Lora, estas son:

1. SPI: Permite la comunicación entre inAir9 y Teensy 3.2.
2. Lora: Permite utilizar las funciones necesarias para recibir y transmitir datos.
3. Compresor Lec: Permite la utilización del compresor LEC.

Iniciadas todas las configuraciones iniciales, el programa entra el ciclo principal, donde inicia la captura de datos provenientes de la puerta serial y que son servidos por la aplicación de datos.

Los dispositivos inAir9 cuentan con una serie de configuraciones que deben especificadas en el setup del programa para el correcto funcionamiento de los dispositivos. El código 1 muestra la configuración del nodo transmisor.

```
void setup()
{
  Serial.begin(9600);    //seteo de baudios para comunicación serial.
  Serial1.begin(9600);

  LoRa.setPins(csPin, resetPin, irqPin);           //

  while (!LoRa.begin(915E6));                       //bucle para inicializar LoRa.

  LoRa.setTxPower(14,PA_OUTPUT_RFO_PIN);           //potencia.

  LoRa.setFrequency(866E6);                         //frecuencia de transmisión.

  LoRa.setSpreadingFactor(7);                       //Spread factor.

  LoRa.setSignalBandwidth(115E3);                  //ancho de banda.

  lec.initialization(MAX_SAMPLES_COMPRESSED_AMOUNT2, MAX_PAYLOAD_SIZE); //inicialización de compresor LEC.

  cleanReadBuffer();                               //limpieza de buffer de lectura.
}
```

**Código 1:** Configuración del nodo transmisor.

## Capítulo 4: Diseño y construcción de software de la red.

---

En el código se aprecia la configuración de los parámetros de comunicación de la Radio, comunicación serial, la inicialización de compresor LEC y sus variables.

### 4.2.1.2 Adquisición de datos

El ciclo comienza con la lectura del puerto serial, donde los datos de los sensores son identificados por el byte de cabecera que contiene el paquete de muestras y que determina el tipo de sensor,

En el código 2 se puede apreciar la bufferización de los datos capturados por la aplicación

```
//----- Loop -----
void loop()
{
if (Serial.available() > 0) //consulta por cantidad de datos recibidos.
{
data_type = Serial.read(); //lectura del tipo de datos enviado.
if(data_type == 1) //si es el tipo 1, entonces.
{
cleanReadBuffer();//limpieza de buffer de lectura.
for (int i1 = 0; i1 < 9; ++i1) //bucle para leer datos.
{
read_buffer[0] = Serial.read();           //captura del dato recibido.
read_buffer[1] = Serial.read();           //captura del dato recibido.
data_set1[i1][samples_counter1] = *(int *)& read_buffer;           //reconstrucción.
}
Serial.write(1);                           //acuso de recibo de datos.

++samples_counter1; //incremento del contador de muestras.
```

**Código 2:** Lectura serial de datos de sensores.

### 4.2.1.3 Compresión

Una de las dificultades técnicas que se pueden encontrar en el diseño de una red de sensores inalámbricos es el drenaje de energía generado principalmente por la comunicación de datos. La energía de los nodos es limitada y eso supone una problemática, porque es necesario que la red funcione por periodos prolongados, En ese sentido, para reducir la comunicación de datos a través de la red se ha determinado comprimir la información localmente antes de ser transmitida.

Una vez que los datos han sido ordenado y almacenados en buffer provisorios es necesarios comprimir las muestras, El compresor utilizado para realizar esta acción es el código LEC , que es un esquema de compresión simple que utiliza una sola tabla de codificación fija de tamaño pequeño para comprimir datos sin pérdida. El esquema divide el alfabeto de los símbolos en grupos cuyo tamaño aumenta exponencialmente.

LEC explota una versión modificada del código Exponential – Golomb de orden 0 , que es un tipo de código universal. Por lo tanto, una palabra de código en LEC es un híbrido de códigos de entropía y códigos binarios. Los grupos están codificados en entropía, mientras que las posiciones de índice de cada residuo en su grupo están codificadas en binario [25]. De esta manera, los códigos de entropía pueden volver a calcularse para adaptarse mejor a la distribución de probabilidad particular de las fuentes que se están comprimiendo.

LEC es un código que comprime los datos sobre la marcha, es decir, en la medida que se adquieren los datos de los distintos sensores, se concatenan los bytes en el buffer de compresión , hasta completar el número máximo soportado por el transmisor( 255 bytes) y no sufre el aumento del tamaño del diccionario [26]. Aunque el esquema LEC es simple, su rendimiento de compresión es bueno y mucho mejor que los obtenidos por otros códigos.

Las relaciones de compresión obtenidas por este código son alrededor del 80% tomando en consideración que los buffers de datos rápidos y lentos son diferentes de esta forma se explota la correlación natural que existe en los datos recogidos.

La descompresión de los datos se hará en la puerta de enlaces, específicamente en la Raspberry Pi, debido a la gran cantidad de paquetes de datos generados por todos los nodos sensores de la red de transmisores.

## Capítulo 4: Diseño y construcción de software de la red.

En el código 3 se aprecia que cada buffer `data_set1[i]` contiene 6 muestras de cada sensor, se define la resolución de los datos comprimidos y el número de bit disponibles después de cada compresión, nótese que independiente la naturaleza de los datos los datos se comprimen en el mismo buffer (`stream_com_data`).

```

if(samples_counter1 == samples_compressed_amount1)
{
    //reset del contador de muestras.

    samples_counter1 = 0;

    //compresión de las muestras.

    lec.compressorLEC(data_set1[0], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[1], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[2], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[3], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[4], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[5], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[6], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    lec.compressorLEC(data_set1[7], samples_compressed_amount1, stream_com_data, stream_index_available, stream_available_bits);

    //última compresión y obtención de la cantidad de bytes resultantes.

    compress_size = lec.compressorLEC(data_set1[8], samples_compressed_amount1, stream_com_data, stream_index_available,
stream_available_bits);

    //transmisión de los datos comprimidos.

    sendCompressData(stream_com_data, compress_size, FAST_SENSORS, samples_compressed_amount1, 0xFF);

    //limpieza de buffer y variables de compresión.

```

**Código 3:** *Etapa de compresión de muestras.*

Realizada la compresión del ultimo buffer de sensores, se obtiene que el número de bytes resultantes de la compresión teniendo como tope 255 byte, que es el máximo payload que puede transmitir el dispositivo trasmisor.

Utilizando la función `sendCompressData`, los datos son enviados para transmitidos por el dispositivo transmisor.

## Capítulo 4: Diseño y construcción de software de la red.

### 4.2.1.4 Transmisión

La etapa de transmisión se crea un paquete de datos donde se especifican direcciones, identificadores y otras características propias del paquete a enviar. Dentro de la función `sendCompressData`, se considera una cabecera en el diseño del payload, donde se especifica el tipo de dato y el número de muestras a comprimir. El código 4 muestra los descrito anteriormente.

```
//función para enviar paquetes de datos.
void sendCompressData(byte _data[], byte _data_length, byte _data_type, byte _samples_compressed_amount, byte _destination_address)
{
    _data[0] = _data_type; //tipo de datos comprimidos (sensores rápidos: 1, sensores lentos: 2).
    _data[1] = _samples_compressed_amount; //cantidad de muestras comprimidas.
    /*_data[3]= nuevo_datos;*/
    if(gt==254 || gt=253) { Byte de autorización de envío de datos comprimidos
    OCP=2 ; //Carga de sensores comprimidos
    leakinfo.msg.Secuencial=data_id;
    LoRa.beginPacket(); //inicio del paquete.
    LoRa.write(_destination_address); //dirección de destino.
    LoRa.write(localAddress); //dirección de fuente.
    LoRa.write(OCP); // El contenido de este paquete son datos comprimidos
    LoRa.write(PACKET_SIZE); // Largo de Id del paquete
    LoRa.write(leakinfo.I2Cpacket,PACKET_SIZE); //id del paquete.
    LoRa.write(_data_length); //largo del payload.
    LoRa.write(_data, _data_length); //payload.
    LoRa.endPacket(); //finalización del paquete.
    //incremento del id del paquete.
    ++data_id;
    }
}
```

**Código 4:** *Transmision de datos comprimidos de sensores.*

El paquete de datos dentro de su encabezado cuenta con las direcciones de origen y destino, específicamente para esta aplicación se ha considerado la dirección de destino la `0xFF`, es decir los mensajes son enviados en broadcast de tal forma que todos los nodos centrales que estén dentro del rango del nodo emisor puedan recibir el mensaje y retransmitirlo. por tanto, se tiene considerado

## Capítulo 4: Diseño y construcción de software de la red.

---

incluir dentro de las rutinas de código de los receptores finales, una aplicación que elimine posibles mensajes repetidos.

### 4.2.1.5 Autorización de transmisión

La transmisión de los datos está condicionada a la autorización que entrega la Puerta de enlaces para poder enviar el mensaje. La radio constantemente chequeará los paquetes de datos recibidos y de acuerdo a los encabezados y el valor del OCP enviará un mensaje de Autenticar (1), mensaje de datos comprimidos (2) o no comprimidos (3).

```
void onReceive(int packetSize) {
  if (packetSize == 0) return;
  // read packet header bytes:
  recipient = LoRa.read();    // recipient address
  byte sender = LoRa.read();  // sender address
  byte incomingMsgId = LoRa.read(); // incoming msg ID
  byte incomingLength = LoRa.read(); // incoming msg length

  while (LoRa.available()) { // can't use readString() in callback, so
    incoming =LoRa.read();   // add bytes one by one
  }
}
```

**Código 5:** *Recepción de mensajes del nodo trasmisor.*

El código 5 muestra la función de recepción, donde se leen las direcciones de destino y del nodo receptor, junto con el contenido del mensaje. El byte de incoming determinará la respuesta que entrega el nodo inalámbrico. Si byte incoming contiene un 255, se enviará un mensaje de hola como se puede apreciar en el código 6. Si el byte contiene un 254 y la dirección de destino leída es la del nodo trasmisor que es autorizado a enviar datos de sensores, según lo indicado en el código 4.

El diseño considera la posibilidad de modificar el comportamiento del nodo trasmisor, es decir, el nodo puede prescindir de la autorización de la puerta de enlaces para enviar mensajes, si este recibe un mensaje con el byte incoming 253 y del mismo modo volver al protocolo por defecto si el byte tiene un valor distinto.

## Capítulo 4: Diseño y construcción de software de la red.

---

En el código 6 se aprecia lectura de las instrucciones enviadas por la puerta de enlaces para modificar el comportamiento del nodo transmisor.

```

gt=incoming;
incoming=0;

// if the recipient isn't this device or broadcast,
if (recipient != localAddress && recipient != 0xFF) {
    return;          // skip rest of function
}

if(gt==253){Mo=1;}//Modo de transmisión broadcast
if(gt!=253){Mo=0;}//Modo de transmisión PCF
if(gt==255){
    OCP=1;//Mensaje hola
    LoRa.beginPacket();          // Inicio transmisión
    LoRa.write(0xFF);           // Dirección destino
    LoRa.write(0xB3);           // Dirección local
    LoRa.write(OCP);            // Tipo de mensaje
    LoRa.write(1);              // Largo del mensaje
    LoRa.write(4);              // Mensaje hola de autenticación para ser registrado en la Puerta de enlaces
    LoRa.endPacket();           // Fin de la transmisión
}

```

**Código 6:** *Recepción de mensajes del nodo transmisor.*

### 4.2.2 Software puerta de enlace

La puerta de enlaces cuenta un nodo receptor, que implementa un código capaz de recibir datos desde varios nodos sensores y escribir la información útil en la puerta serial de la Raspberry Pi, de tal forma que puedan ser leídos y procesados. Adicionalmente, se le ha entregado el control del tráfico en la red.

El tráfico de la red es controlado mediante un protocolo basado en la gestión de los nodos descubiertos por un mensaje que es transmitido por la Puerta de enlaces. En este escenario, ella

Capítulo 4: Diseño y construcción de software de la red.

decide en qué momento trasmite cada nodo y es capaz de verificar periódicamente la existencia de los nodos cercanos.

En el diagrama de flujo siguiente se aprecia un esquema simplificado del funcionamiento de la etapa 1 de la puerta de enlaces, código ejecutado por la Teensy.

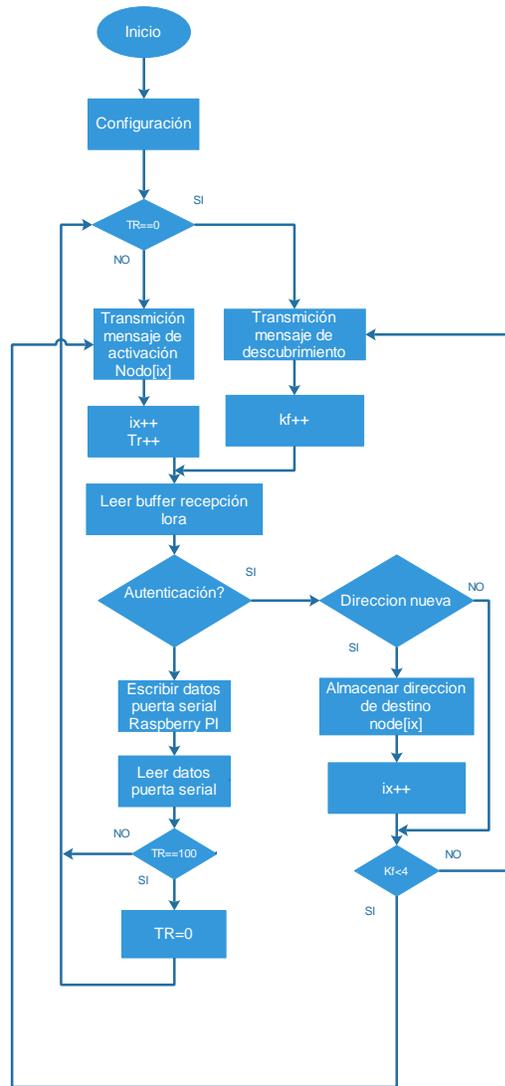


Figura 4.8: Diagrama de flujo de puerta enlaces etapa 1.

La aplicación envía un mensaje en broadcast con el objetivo de obtener una respuesta de los nodos sensores, esta es enviada 3 veces, Acto seguido registra en un buffer las direcciones de los transmisores que respondieron al llamado y envía un mensaje con la dirección del nodo que va a comunicar. El mensaje contiene un byte de activación y solo puede ser leído por una dirección específica.

## Capítulo 4: Diseño y construcción de software de la red.

---

Con el objetivo de almacenar direcciones nuevas y borrar aquellas de nodos que se han alejado del radio de alcance del nodo central, de forma periódica se vacían los buffers de almacenamiento de direcciones y se comienza el proceso de búsqueda nuevamente.

El código 7 describe la secuencia de envío de mensajes de la puerta de enlaces.

```

void sendMessage() {
    if(Mo==1){ msg=253;}
    if(kf<3&& Mo==0){ msg=255;}
    if(kf>3 && kx>0 && Mo==0){ msg=254;
        destination=nodes[ix];
        ix++;
        if(ix==kx){ix=0;}
    }
    LoRa.beginPacket(); // start packet
    LoRa.write(destination); // add destination address
    LoRa.write(localAddress); // add sender address
    LoRa.write(msgCount); // add message ID
    LoRa.write(1); // add payload length
    LoRa.write(msg); // add payload
    LoRa.endPacket(); // finish packet and send it
    kf++;
    if(kf==30){
        kf=0;
        kx=0;
        for(int i=0;i<5;i++){
            nodes[i]=0;

```

**Código 7:** Ciclo de envío de mensajes para la gestión de la transmisión de los nodos.

La función onreceive constantemente chequea si existen datos disponibles, De existir, se lee byte a byte, en el orden que fueron enviados y de esta forma, determinar el tipo de mensaje, Si este paquete corresponde a una respuesta al mensaje de descubrimiento se almacena la dirección.

Si corresponde a una estructura de datos útiles para el sistema de adquisición estos son escritos en la puerta serial con la función serial.write, en la estructura necesaria para poder reconstruir el paquete e identificar el tipo de dato, sean estos comprimidos o no comprimidos.

## Capítulo 4: Diseño y construcción de software de la red.

El código 8 contiene la función de recepción de datos y la escritura de ellos en la puerta serial de la Raspberry Pi una vez que han sido analizados para ser procesados.

```

void onReceive(int packetSize) {
  if (packetSize == 0) return;    // if there's no packet, return
  // read packet header bytes:
  int recipient = LoRa.read();    // recipient address
  byte sender = LoRa.read();     // sender address
  byte OPC = LoRa.read();        // msg ID contenido (OPC=1,Hola;OPC=2, Mensaje comprimido, OPC=3, Datos sin comprimir
  byte incomingLength = LoRa.read(); // incoming msg length
  /.....
  Serial.write(recipient);        //Escritura Byte receptot
  Serial.write(sender);          // Escritura Byte Emisor
  Serial.write(size_ide);        // Escritura Byte tamaño IDE
  for(int i1=0;i1< ide_size;i1++){
    Serial.write(incomingMsgId[i1]);
  }
  Serial.write(incomingLength);   // Escritura tamaño payload
  payload_size = int(incomingLength);
  for (int i1 = 0; i1 < payload_size; i1++) {
    (stream_com_data[i1]) = LoRa.read(); //Lectura de payload
    Serial.write(stream_com_data[i1]); //Escritura de payload
  }
  for (int i1 = 0; i1 < payload_size; i1++) {
    (stream_com_data[i1]) = LoRa.read(); //Lectura de payload
    Serial.write(stream_com_data[i1]); //Escritura de payload
  }
  byte Ta1=Serial.read();
  byte Tr1=Serial.read();
  byte Mo1=Serial.read();
  byte Ma1=Serial.read();
  if(Ta1*Tr1>0)
  {
  Ta=int(Ta1)*1000;//Tiempo de activación
  Tr=int(Tr1);//Tasa de refresco
  Mo=int(Mo1);//Modo de Operación nodo ( Protocolo de comunicación o Broadcast)

```

**Código 8:** Lectura de datos de nodos transmisores.

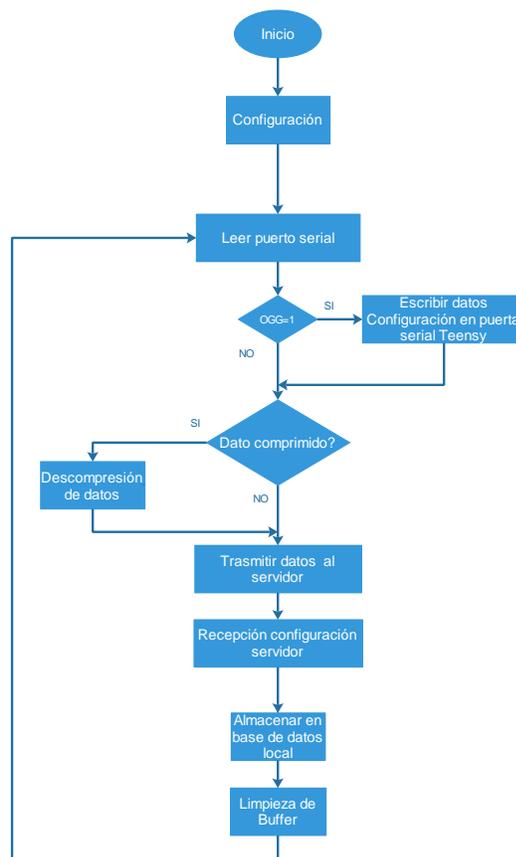
## Capítulo 4: Diseño y construcción de software de la red.

### 4.2.2.1 Procesamiento de los datos

Qt Creator es un IDE multi plataforma programado en C++, JavaScript y QML creado por Trolltech el cual es parte de SDK para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario (GUI por sus siglas en inglés) con las bibliotecas Qt [27]. Este software ha sido considerado para programar la puerta de enlace del proyecto.

1. Lectura del Puerto serial de datos.
2. Descompresión de datos.
3. Conexión y construcción de tabla en servidor local PostgreSQL.
4. Establecer comunicación cliente-servidor mediante WebSocket.

En el diagrama de flujo siguiente se aprecia un esquema simplificado del funcionamiento de la etapa 1 de la puerta de enlaces, código ejecutado por la Teensy.



**Figura 4.9:** Diagrama de flujo de puerta enlaces etapa 2.

## Capítulo 4: Diseño y construcción de software de la red.

---

En la primera etapa del código, se incluyen las bibliotecas que forman parte de la estructura de la aplicación puerta de enlaces. Para esto fue necesario la inclusión de las siguientes librerías

1. QSerialPort
2. compressor\_lec.h
3. QtSql/QtSql
4. QtWebSockets/QWebSocket
5. QJsonObject

El programa principal de la puerta de enlaces en su inicio conecta la red de transmisores Lora y las aplicaciones que residen en Internet, con el objetivo de traducir la información del protocolo utilizado en la red inicial, al protocolo usado en la red de destino.

```
int main(int argc, char *argv[])
{
    QCoreApplication a(argc,argv);

    Mythreadserial readserial; // Crear el hilo de ejecución de lectura serial
    readserial.start();// Comienzo del Hilo
    SslEchoClient client(QUrl(QStringLiteral("wss://192.168.0.88:1234"))); //Dirección del
    Host Donde esta almacenado el servidor

    Q_UNUSED(client);

    return a.exec();
}
```

**Código 9:** Programa principal de la Puerta de enlaces.

El primer hilo de ejecución inicializa la comunicación serial entre la Raspberry Pi y la Teensy. En segundo lugar, se realiza la conexión con el servidor para iniciar el intercambio de mensajes.

La dirección 192.168.0.88:1234 utilizada en el código 9, corresponde a la dirección de red asignada a la aplicación servidor que será utilizado para conectar la aplicación cliente desarrollado ente este trabajo de título.

## Capítulo 4: Diseño y construcción de software de la red.

Los datos que son leídos por la aplicación que reside en la Raspberry Pi son analizados byte a byte de forma de determinar con exactitud los encabezados y la información que contienen.

```
for(;;){
    // se accede a leer Si existen datos disponibles
    if(serialPort.bytesAvailable() || serialPort.waitForReadyRead(10)){

        //Se leen los primero 4 bytes de la trama correspondientes
        //(destino, Origen , largo iden mensaje)
        Buffer= serialPort.read(4);
        int destino= (int)Buffer [0]; // Destino
        int Origen= (int)Buffer [1]; // Origen
        int OCP= (int)Buffer [3]; // Origen
        int msgid_lenght= (int)Buffer [2]; // Largo del identificador secuencial
        // Lectura de bytes necesarios para reconstruir el identificador
        identimsg=serialPort.read(msgid_lenght);//
        for (int i=0;i<SECUE_SIZE ;i++)
        {
            leakinfo.I2CPacket[i]=identimsg[i];
        }
        If(OCP==2){ // Datos comprimidos
            // Se leen los siguientes 3 bytes ( Largo dato, tipo de dato , numero de muestras)
            Buffer1= serialPort.read(3);

            identimsg.clear();

            int msgid=leakinfo.msg.Secuencial;

            int data_length= (int)Buffer1 [0]; // Se almacena el largo del payload
            int _data_type =(int)Buffer1 [1]; // Se almacena el tipo de dato
            //Se almacena la cantida de muestras
            int samples_compressed_amount_aux =(int)Buffer1 [2];
        }
    }
}
```

**Código 10:** Lectura de encabezados para determinar parámetros del paquete de datos.

Identificado el largo del payload se procede a leer todos bytes que lo contienen. Sin embargo estos datos están comprimidos y es necesario convertir su representación al tipo de dato unsigned char, formato soportado por el descompresor LEC para reconstruir los datos originales .

```
// Se leen los datos del payload contenidos en el mensaje
Buffer_data=serialPort.read(data_length-2);
// Se realiza la conversión de QByteArray a unsigned char

const std::size_t count =Buffer_data.size();
unsigned char*datosCOM=new unsigned char[count];
memcpy(datosCOM,Buffer_data.constData(),count);
Buffer_data.clear();// Limpieza de buffer de datos
```

**Código 11:** Lectura variables de entrada para el descompresor.

El descompresor de datos LEC necesita configuraciones específicas entrada para funcionar correctamente. Se debe considerar que los datos deben ser ingresados según una estructura determinada

## Capítulo 4: Diseño y construcción de software de la red.

En primera instancia se configuran las muestras tomadas, de tal forma que el buffer de salida, la resolución de los datos , tengan las dimensiones adecuadas según el tipo de dato y el número de muestras .

```
//seteo de cantidad de muestras comprimidas.
setSamplesCompressedAmount(_data_type, samples_compressed_amount_aux);
if (_data_type == 1) { // Identifica si los datos son del tipo 1 o tipo 2
//descompresión.
lec.decompressorLEC(datosCOM, 9, samples_compressed_amount, bits_resolution_data1, stream_decom_data);
```

**Código 12:** Función para dimensionar buffer de datos de salida.

Donde los datos de entrada del descompresor se detallan a continuación

1. **\_stream\_com\_data:** Buffer de datos del tipo Array byte
2. **\_data\_types\_amount:** Tipo de muestra ,tipo 1 o Tipo 2
3. **\_samples\_amount :**Numero de muestras , Dato tipo byte
4. **\_bits\_resolution\_data:** Resolución de datos (16 bit), dato tipo int
5. **\_stream\_decom\_data:** Buffer de salida de datos, tipo array int

```
// se Almacenan datos en servidor local
if(db.open()){
//qDebug() << "Conexion servidor OK " << "\n";
if(db.tables().contains("sensores_tipo_1")){

QSqlQuery query;
query.prepare("INSERT INTO sensores_tipo_1 (ide,
acc_x,acc_y,acc_z,giro_x,giro_y,giro_z,magn_x,magn_y,magn_z) "
"VALUES (:ide, :acc_x,:acc_y,:acc_z,:giro_x,:giro_y,:giro_z,:magn_x,:magn_y,:magn_z)");

query.bindValue(":ide", msgid);
query.bindValue(":acc_x", stream_decom_data[0][0]);
query.bindValue(":acc_y", stream_decom_data[1][0]);
query.bindValue(":acc_z", stream_decom_data[2][0]);
query.bindValue(":giro_x", stream_decom_data[3][0]);
query.bindValue(":giro_y", stream_decom_data[4][0]);
query.bindValue(":giro_z", stream_decom_data[5][0]);
query.bindValue(":magn_x", stream_decom_data[6][0]);
query.bindValue(":magn_y", stream_decom_data[7][0]);
query.bindValue(":magn_z", stream_decom_data[8][0]);
query.exec();

}
}
}
```

**Código 13:** Almacenamiento de datos en la base de datos PostgreSQL.

Los datos son almacenados en una base de datos local del tipo relacional SQL, utilizando el sistema de gestión PostgreSQL.

## Capítulo 4: Diseño y construcción de software de la red.

Los datos pueden ser accedidos mediante un cliente SQL, La herramienta Query Tool que viene incorporada con pgAdmin que permite ejecutar comandos SQL y da la opción de analizar la base de datos de forma gráfica.

### 4.2.3 Software de conexión a servicios de Internet

La finalidad de sistema de adquisición es poder tener acceso a los datos desde un lugar remoto, usando aplicaciones u otro medio de procesamiento de datos, con el objetivo de obtener indicadores de producción basados en las mediciones realizadas por los sensores que se encuentran en las cajas de la cosecha.

Para realizar el intercambio de datos se creó una aplicación basada en la tecnología WebSocket que hace posible establecer una conexión continua full-dúplex, entre un cliente y servidor.

```
//Conexion de la señal con SLOT señal conectada
connect(&m_webSocket, &QWebSocket::connected, this, &SslEchoClient::onConnected);
//Conexion señal con SLOT error
connect(&m_webSocket, QOverload<const QList<QSSL::SslError>&>::of(&QWebSocket::sslErrors),
        this, &SslEchoClient::onSslErrors);

m_webSocket.open(QUrl(url));// abre la URL especificada
```

**Código 14:** *Conexión de señales destinada a establecer conexión cliente-servidor.*

El intercambio de datos también considera instrucciones que provienen del servidor con el objetivo de cambiar variables de funcionamiento del protocolo de comunicación, del mismo modo generar la posibilidad de modificar el comportamiento de los nodos de la red. El código 15 muestra la función que recibe estas instrucciones.

```
// Recibe Mensajes desde el servidor con configuraciones especiales de la Puerta de enlaces
//! [processBinaryMessage]
void SslEchoClient::processBinaryMessage(QByteArray message)
{
    C_puerta de enlaces[0]= (int)message[0];//Mensaje de servidor
    C_puerta de enlaces[1]= (int)message[1];//Tiempo de Activación
    C_puerta de enlaces[2]= (int)message[2];//Tasa de refresco
    C_puerta de enlaces[3]= (int)message[3];//Modo de Funcionamiento Nodos
    C_puerta de enlaces[4]= (int)message[4];//Opcional

    way.conf_i_puerta de enlaces( C_puerta de enlaces);

    qDebug() << "Configuración recibida:";
}
```

**Código 15:** *Recepción de instrucciones del servidor*

En lo que respecta a la data adquirida, la información es convertida en una trama JSON donde finalmente es enviada al servidor.

## Capítulo 4: Diseño y construcción de software de la red.

```

void SslEchoClient::enviotrama(){ // Se convierten los datos en una trama JSON

    QJsonObject recordObject;
    recordObject.insert("IDE", QJsonValue::fromVariant(dato[10]));
    recordObject.insert("ACX", QJsonValue::fromVariant(dato[0]));
    recordObject.insert("ACY", QJsonValue::fromVariant(dato[1]));
    recordObject.insert("ACZ", QJsonValue::fromVariant(dato[2]));
    recordObject.insert("GIRO X", QJsonValue::fromVariant(dato[3]));
    recordObject.insert("GIRO Y", QJsonValue::fromVariant(dato[4]));
    recordObject.insert("GIRO Z", QJsonValue::fromVariant(dato[5]));
    recordObject.insert("MAGN X", QJsonValue::fromVariant(dato[6]));
    recordObject.insert("MAGN Y", QJsonValue::fromVariant(dato[7]));
    recordObject.insert("MAGN Z", QJsonValue::fromVariant(dato[8]));

    QJsonArray recordsArray;
    recordsArray.push_back(recordObject);
    QJsonDocument doc(recordsArray);

    if(dato[10]==0){ // Si no hay datos disponibles no envía mensajes
        qDebug() <<"No hay datos";
    }
    else { // si Hay datos disponibles se envía el mensaje
        m_webSocket.sendTextMessage(QString::fromUtf8(doc.toJson(QJsonDocument::Compact)));
    }
}

```

**Código 16:** Función para transmisión de datos tipo 1 al servidor.

### 4.3 Conclusiones

La construcción del software significó una carga importante para el desarrollo del proyecto. El 80% de las HH invertidas corresponden a este ítem, donde se incluyen trabajos asociados a programación y pruebas.

El respaldo de la información adquirida utilizando el motor de bases de datos PostgreSQL, entrega al sistema el soporte necesario para almacenar la información y poder acceder a ella mediante la intranet propia del sistema. Esta tecnología abre la posibilidad de que cualquier dispositivo con la capacidad de poder desplegar una interfaz gráfica y que tenga instalado un cliente SQL autorizado pueda consultar la data de cualquier nodo.

A la hora de implantar cualquier solución IoT, se tienen que garantizar al cliente o al usuario un servicio de red que garantice la conectividad y el adecuado manejo de la información, incluir la tecnología WebSocket permitió, asegurar un canal bidireccional para el tránsito de la información y por otro lado tener control desde el servidor de ciertas variables del sistema de transmisión.

## CAPÍTULO 5. PRUEBAS DE CAMPO Y RESULTADOS

### 5.1 Introducción

En este capítulo se presentan los experimentos realizados con los dispositivos inalámbricos seleccionados. Involucra las actividades relacionadas a la realización de pruebas de funcionamiento y la comunicación entre los sistemas que forman parte del diseño. Para esta etapa se consideraron 3 tipos de pruebas.

#### 5.1.1 Pruebas Aisladas

En esta prueba se considera probar la funcionalidad del equipo y detectar configuraciones adecuadas de funcionamiento. Para esto se ha definido registrar RSSI que es un indicador de fuerza de la señal recibida. La primera experiencia contempla registrar el RSSI en intervalos de 10 metros, cubriendo una distancia total de 120 metros. El registro de los resultados se hará en la Tabla 5.1

**Tabla 5.1:** Registro de medición de RSSI frente a variaciones de distancia.

Distancia (m)	RSSI (dBm)
10	
20	
....	
120	

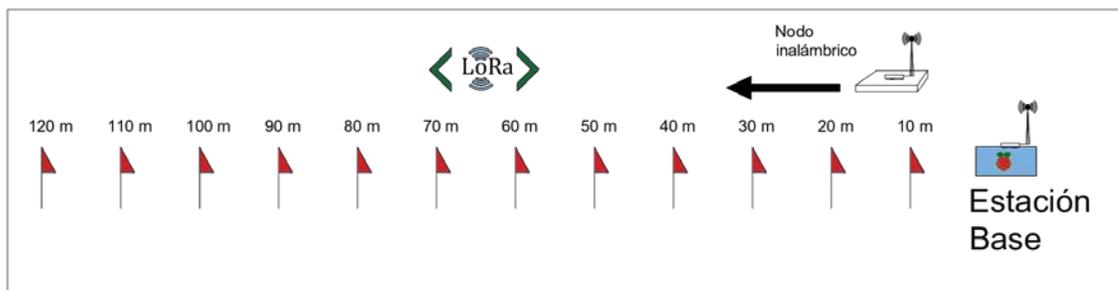
La segunda experiencia contempla mantener fija la distancia entre los nodos que participan de la prueba y variar la tasa de muestreo. La Tabla 5.2 debe ser utilizada para el registro de los datos obtenidos.

**Tabla 5.2:** Registro de medición de RSSI frente a variaciones de tasa de muestreo.

Tiempo de muestro (s)	RSSI (dBm)
1	
0.5	
....	
0.0625	

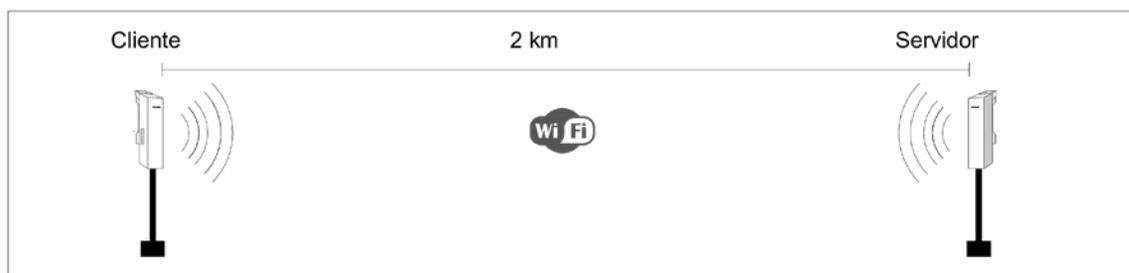
Capítulo 5: Pruebas de campo y resultados.

La Figura 5.1 muestra el esquema del experimento planteado anteriormente.



**Figura 5.1:** Esquema propuesto para la medición de RSSI.

Para los puntos de acceso (trasmisores CPE510) es necesario verificar conectividad del enlace inalámbrico considerando una distancia 2 kilómetros en línea recta. En esta prueba es necesario montar una aplicación cliente/servidor básico según lo indica en el diagrama de la Figura 5.2 y realizar las configuraciones indicadas en el manual de usuario del dispositivo.



**Figura 5.2:** Aplicación cliente /servidor.

**5.1.2 Pruebas de integración**

La prueba de integración consiste en probar la funcionalidad del hardware propuesto utilizando el software correspondiente a cada una de las zonas definidas en el Capítulo 4.

Se debe verificar el conexionado de todos los equipos de la red y asegurar el correcto funcionamiento durante un periodo de 24 horas.

Se debe constatar la integridad de los datos transmitidos, es decir, Asegurar que los datos adquiridos por el nodo trasmisor deben ser los mismos que han sido almacenados en la base de datos local y que son transmitidos al servidor.

Capítulo 5: Pruebas de campo y resultados.

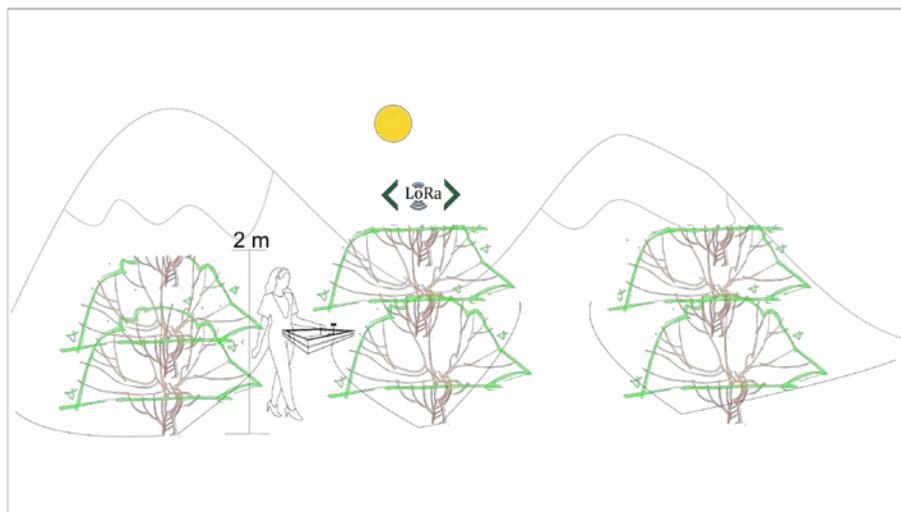
**5.1.3 Pruebas en las condiciones de sitio**

Las pruebas realizadas en las condiciones de sitio contemplan evaluar la funcionalidad del prototipo en sectores de cosecha de fruta. Bajo esta modalidad se necesita conocer el comportamiento de los equipos inAir9, específicamente medir las variables RSSI, SNR y paquetes perdidos.

**Tabla 5.3:** Registro de medición de RSSI frente a variaciones de distancia.

Distancia (m)	RSSI (dBm)	SNR (dB)	PX (Unidad)
10			
20			
....			
Ancho de predio			

Las pruebas anteriores no consideran la interferencia que pueden provocar los árboles o arbustos frutales presentes en el lugar de sitio. La Figura 5.3 representa esta situación.

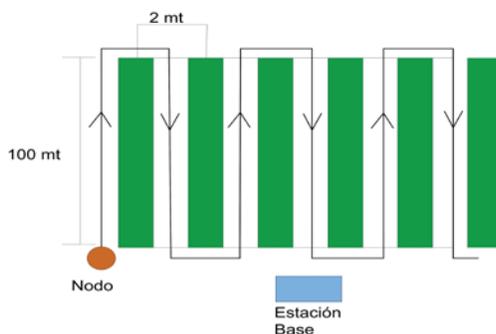


**Figura 5.3:** Árboles o Arbustos frutales presentes en las condiciones de sitio.

Frente a esta situación es necesario cambiar el esquema de pruebas utilizado en la Figura 5.1 y considerar una trayectoria que permita evaluar la funcionalidad frente a los obstáculos que pueden interferir en los procesos de comunicación entre un nodo inalámbrico y la estación base.

Capítulo 5: Pruebas de campo y resultados.

La Figura 5.4 se puede apreciar el esquema de traslación del nodo sensor y la disposición de la estación base.



**Figura 5.4:** Esquema de desplazamiento nodo transmisor.

Para todas las pruebas es preciso registrar todas condiciones climáticas existentes según la Tabla 5.4.

**Tabla 5.4:** Registro de condiciones climáticas.

Hora	
Precipitaciones	
Humedad	
Temperatura	
Viento	
Ráfagas	
Punto de rocío	

**5.2 Pruebas de campo**

Las pruebas de alcance de los diferentes dispositivos se desarrollaron en plaza de la Universidad del Bío-Bío y en un sector aledaño a la comuna de Cabrero. Las pruebas de funcionalidad en las condiciones de sitio fueron realizadas en agrícola industrial El Boldo S.A. ubicada a 10 kilómetros al noroeste de la localidad de campanario.

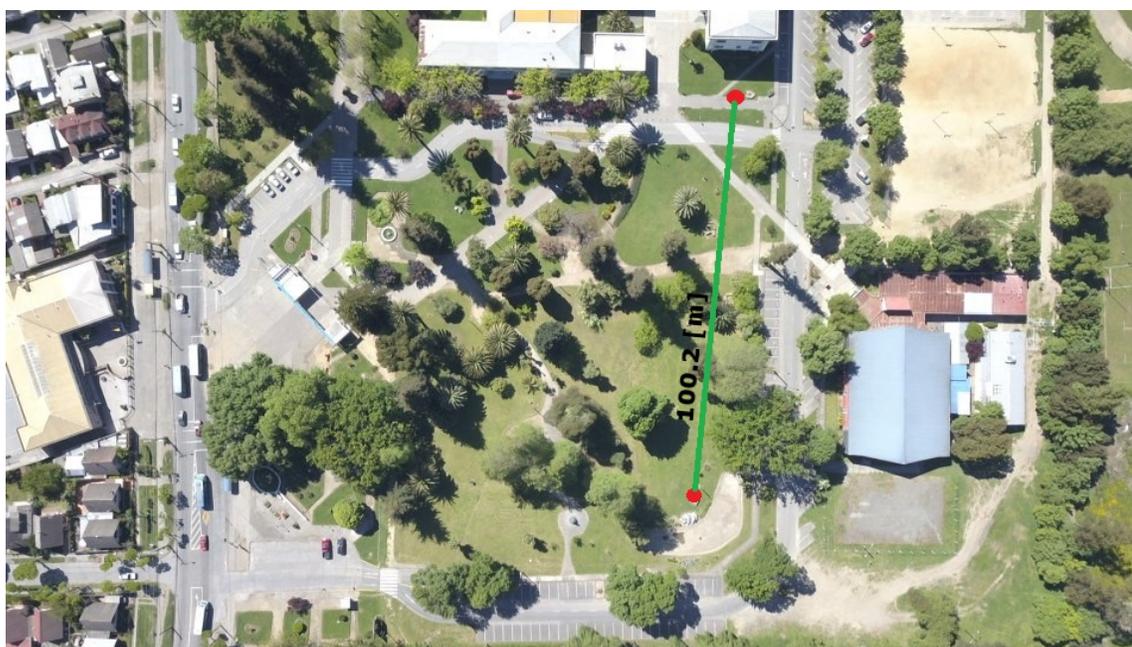
Tanto los dispositivos transmisores y todos sus accesorios para la ejecución de las pruebas fueron suministrados por los fondos del Proyecto Fondef ID16I10206.

Capítulo 5: Pruebas de campo y resultados.

Se desarrollaron pruebas de máximo alcance para espacios Indoor y Outdoor. Se busco en las pruebas encontrar la mejor configuración de los dispositivos Lora y CPE510, para asegurar las distancias mínimas comprometidas y su funcionamiento en condiciones de sitio.

**5.2.1 Dispositivos Lora**

Las pruebas realizadas dentro de las instalaciones de la Universidad del Bio fueron ejecutadas con el objetivo de verificar la funcionalidad de los equipos y el software (librerías) que acompaña a los dispositivos Lora.



**Figura 5.5:** *Emplazamiento de Pruebas dentro del campus universitario.*

Los nodos fueron configurados con las características que se mencionan en la Tabla 5.5.

**Tabla 5.5:** *Tabla de configuración de nodo inalámbrico.*

setTxPower	14 PA_OUTPUT_RFO_PIN
setFrequency	866E6 Mhz
setSpreadingFactor	7
setSignalBandwidth	125E3 Mhz

Capítulo 5: Pruebas de campo y resultados.

Se transmitieron paquetes de datos con un tamaño de 100 bytes. Las pruebas se realizaron variando la distancia de transmisión para obtener el gráfico de decaimiento del RSSI y se utilizaron diferentes tasas de muestreo con el objetivo de verificar la mínima variabilidad de este parámetro.

En las Figuras 5.6 se aprecian los resultados obtenidos de las mediciones de RSSI realizadas en intervalos de 10 metros, cubriendo un área total de 100 m. En la Figura 5.7 se mantuvo la distancia constante (100 m) y se varió el intervalo de tiempo de espera que utiliza el trasmisor antes de enviar un mensaje.

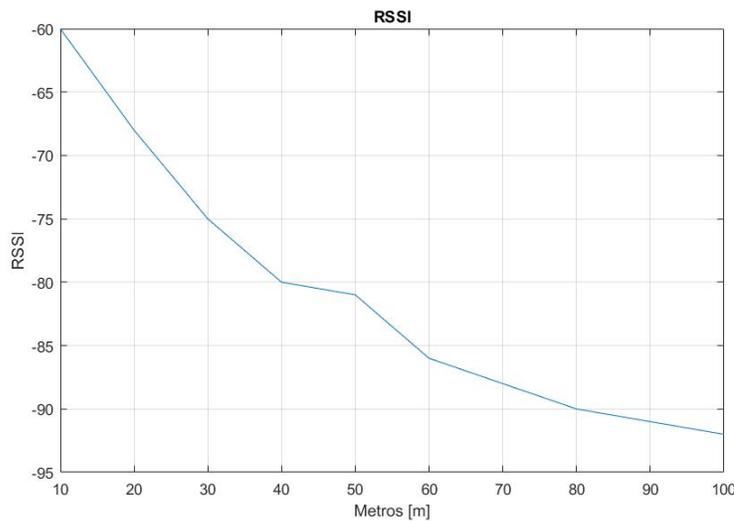


Figura 5.6: Gráfico de datos obtenidos de la medición de RSSI cada 10 metros.

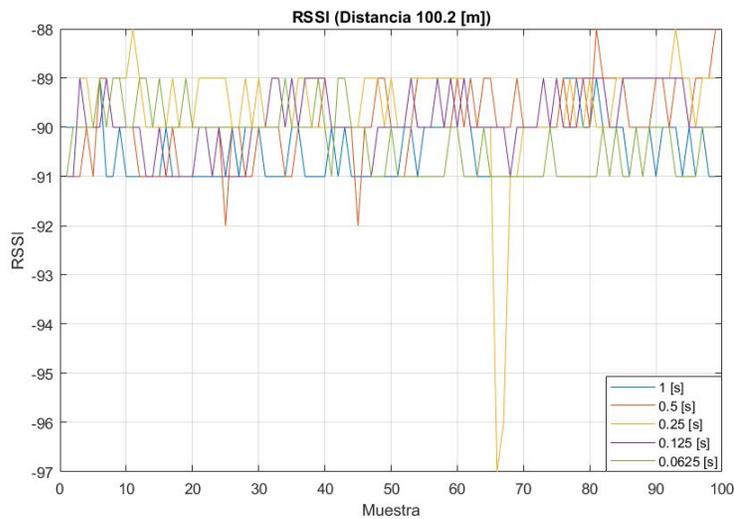


Figura 5.7: Gráfico de datos obtenidos de la medición de RSSI en distintos tiempos de muestreo.

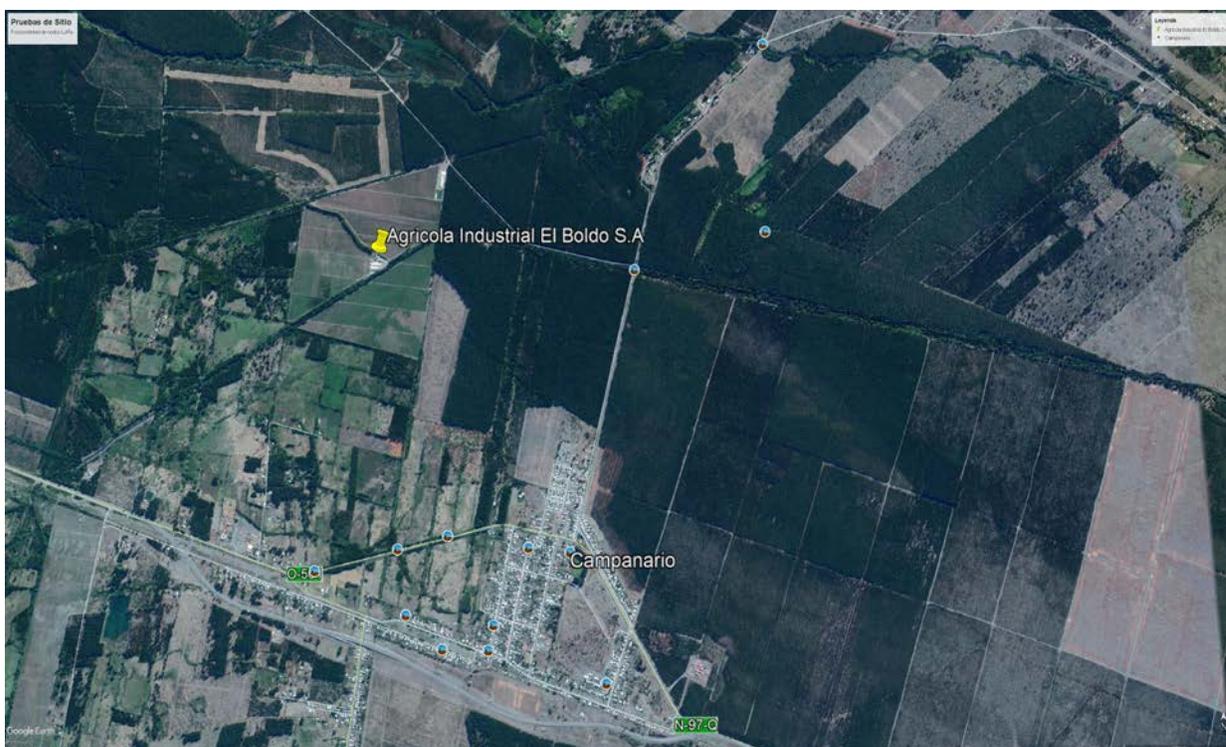
## Capítulo 5: Pruebas de campo y resultados.

De las pruebas se aprecia que existe un decaimiento rápido hacia el valor -94 dB (ver Figura 5.6). Sin embargo, es un valor esperado para este tipo de aplicaciones. Se prueba observando la Figura 5.7 la mínima variabilidad del RSSI al disminuir el periodo de muestro.

### 5.2.1.1 Pruebas en lugar de Sitio.

Agrícola industrial El Boldo una empresa dedicada al cultivo, cosecha y exportación agrícola en Chile. El producto principal que exporta está constituido principalmente por el arándano en 5 variedades de presentación (Duke, Brightwell, O'neal, Tifblue, Brigitta).

Su infraestructura se encuentra ubicada en la comuna de Yungay, ubicada en la provincia de Ñuble en la VIII región del Bio, En la Figura 5.6 se aprecia el lugar donde fueron ejecutadas las pruebas.



**Figura 5.8:** Ubicación de empresa Agrícola Industrial El Boldo.

Capítulo 5: Pruebas de campo y resultados.

Las condiciones ambientales en el momento de las pruebas fueron las siguientes.

**Tabla 5.6:** *Condiciones climáticas fundo santa Luisa.*

**Fuente:** *https://weather.com.*

Hora	9:00 Am
Precipitaciones	0%
Humedad	72%
Temperatura	15°c
Viento	7 km/h
Ráfagas	9 km/h
Punto de rocío	10°

Los nodos Lora fueron configurados según la Tabla 5.5 Para la estación base, se consideró que el tiempo de activación, es de 2 segundos y fue instalada a 1 metro sobre el nivel del suelo, ver Figura 5.10. El nodo transmisor enviará mensajes desde una altura de 15 cm sobre el nivel del suelo y se moverá a una velocidad de 0,5 metros por segundos. En la Figura 5.9 se aprecia una panorámica del lugar donde se instalaron los equipos.



**Figura 5.9:** *Panorámica de huerto n°1, fundo santa Isabel.*

## Capítulo 5: Pruebas de campo y resultados.

---



*Figura 5.10: Instalación de estación base.*



**Figura 5.11: Instalación del nodo transmisor.**

En las pruebas realizadas se utilizó el protocolo de comunicación diseñado y descrito en capítulo 2 y se obtuvieron datos de RSSI, SNR y cantidad de paquetes perdidos (PX) de un nodo que se

Capítulo 5: Pruebas de campo y resultados.

desplazó en intervalos de 10 y 50 metros, cubriendo una distancia total de 350 metros. En las Figuras 5.12, 5.13 y 5.14. se pueden apreciar los resultados de estas pruebas.



Figura 5.12: Gráfico de medición de RSSI.

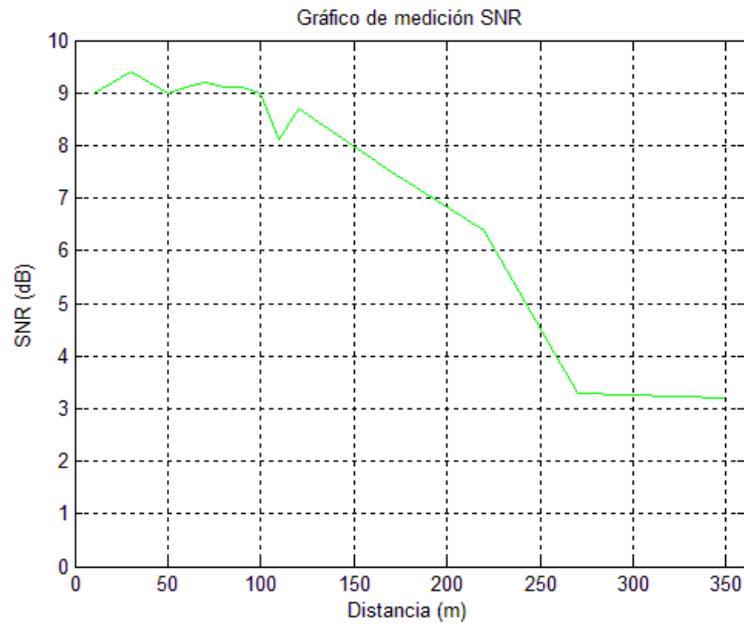


Figura 5.13: Gráfico de medición de SNR.

Capítulo 5: Pruebas de campo y resultados.

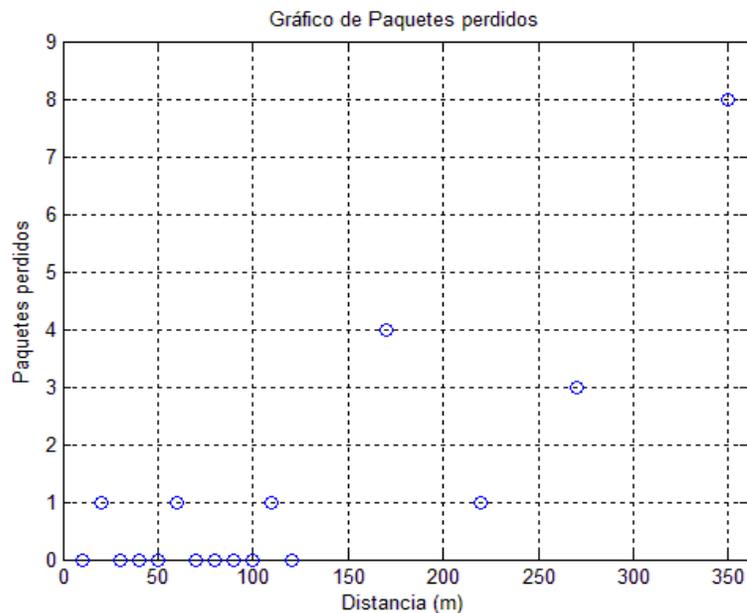


Figura 5.14: Gráfico de paquetes perdidos.

En este escenario las mediciones se realizaron en línea directa, es decir, la estación base se ubicó en directamente frente al nodo. y se desplazó hacia el límite del huerto señalado en la Figura 5.15, los resultados corresponden al promedio de un total de 99 paquetes recibidos.

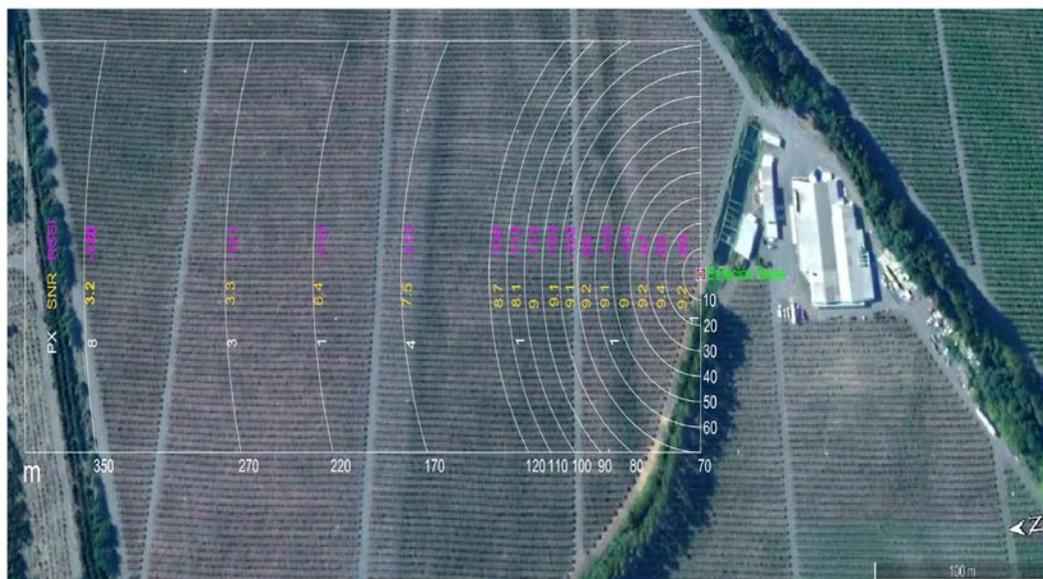


Figura 5.15 : Datos de RSSI, SNR en una zona de 350 metros.

El paquete de datos es transmitido una vez que la estación base envía la autorización para el envío de datos, El payload considerado para esta prueba fue de 47 bytes.

## Capítulo 5: Pruebas de campo y resultados.

---

Las pruebas entregan datos valiosos como el indicador de fuerza de la señal recibida (RSSI) y que su valor más bajo alcanza  $-121$  dBm en el punto más alejado del huerto. Otro indicador que guarda relación con la calidad de la señal es el SNR que es una relación entre la señal y el ruido, que en nuestro caso alcanzo en el punto más extremo  $3.2$  dB.

Se verificó la funcionalidad del sistema de transmisión, instalando la estación base en un punto central. Bajo esta modalidad el nodo trasmisor se trasladó entre arbustos a una velocidad de  $0.5$  m/s. En la Figura 5.16 se aprecia el montaje de la estación base en un punto central del huerto.



**Figura 5.16:** Estación base ubicada en punto central.

El nodo trasmisor se desplazó alternadamente entre diferentes puntos de cosecha, siguiendo un patrón de expansión en el eje  $+x$ , si consideramos la estación base como punto central del plano cartesiano  $(X,Y,Z)$ . Los datos extraídos de estas pruebas corresponden al promedio de 200 paquetes recibidos. Se tienen los resultados de  $RSSI=-88$ dBm,  $SNR=9$ dB y 1 paquete perdido.

Capítulo 5: Pruebas de campo y resultados.

**5.2.2 Transmisores Tp Link CPE510**

Las pruebas realizadas con los transmisores CPE510 fueron ejecutadas en la localidad de cabrero, provincia de Bio-Bio, El lugar fue seleccionado por ofrecer una línea directa y despejada para crear el enlace entre el trasmisor que realizara la función de punto de acceso y un cliente. En la Figura 5.17 se puede apreciar el lugar de las pruebas.



**Figura 5.17:** Emplazamiento de las pruebas realizadas.

La distancia considerada para realizar la medición fue 2 kilómetros y se consideró una prueba de ping para comprobar la conectividad de los transmisores. Los datos que fueron configurados son los descritos en la Tabla 5.7.

**Tabla 5.7:** Configuraciones CPE510

MAXtream		Apagado
Canal/frecuencia		157/5785 MHZ
Ancho del canal		20/40 MHZ
Modo IEE802.11		A/N Mixto
Tasa Max de TX		300.0 Mbps
Potencia de transmisión		7 dBm
Distancia		5.0 km
Dirección MAC		C4-E9-84-F5-45-4C

Capítulo 5: Pruebas de campo y resultados.

---

El montaje de los equipos se puede apreciar en las siguientes fotografías. En la Figura 3.31 se aprecia el montaje del cliente y en la Figura 3.32, el transmisor que servirá de servidor en la aplicación.



**Figura 5.18:** Montaje de CPE510 aplicación cliente.

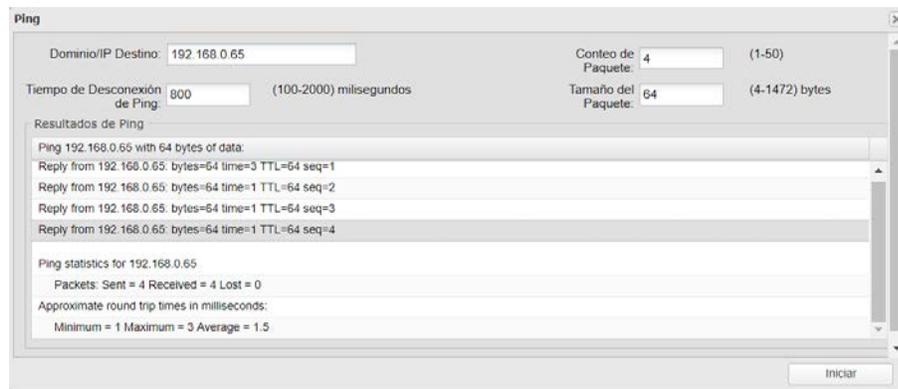


**Figura 5.19:** Montaje de CPE510 aplicación servidor.

## Capítulo 5: Pruebas de campo y resultados.

Se verifico la conexión mediante una utilidad de red conocida como ping, integrada por defecto en el software propio de los equipos transmisores. El objetivo del ping es determinar si el host de destino (servidor), identificado con una IP determinada, es accesible desde otro host (Cliente).

En la prueba el cliente, envía al servidor un paquete de información de 64 bytes utilizando el protocolo ICMP y espera una respuesta que debe contener los mismos datos enviados.

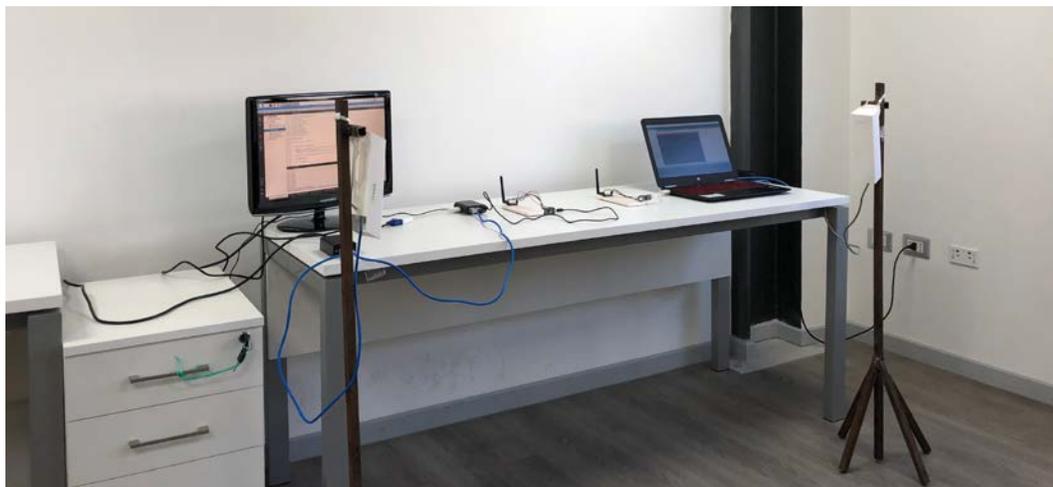


**Figura 5.20:** Diagnóstico de red utilizando la herramienta ping.

Si la repuesta corresponde a lo esperado, el ping ha sido satisfactorio, En la Figura 5.20, se enviaron 4 mensajes y se recibieron 4, con 0 mensajes perdidos.

### 5.2.3 Pruebas de integración

En la Figura 5.21 se aprecia la integración total de todos los dispositivos y software descritos en el capítulo 3 y 4. A la Izquierda se encuentra la estacion base y en el sector izquierdo el nodo transmisor junto a la aplicación que sirve de datos al sistema.



**Figura 5.21:** Montaje del sistema de diseñado.

Capítulo 5: Pruebas de campo y resultados.

Para las pruebas se consideró utilizar una aplicación que sirve de datos al nodo sensor que se encuentra Figura 5.22.

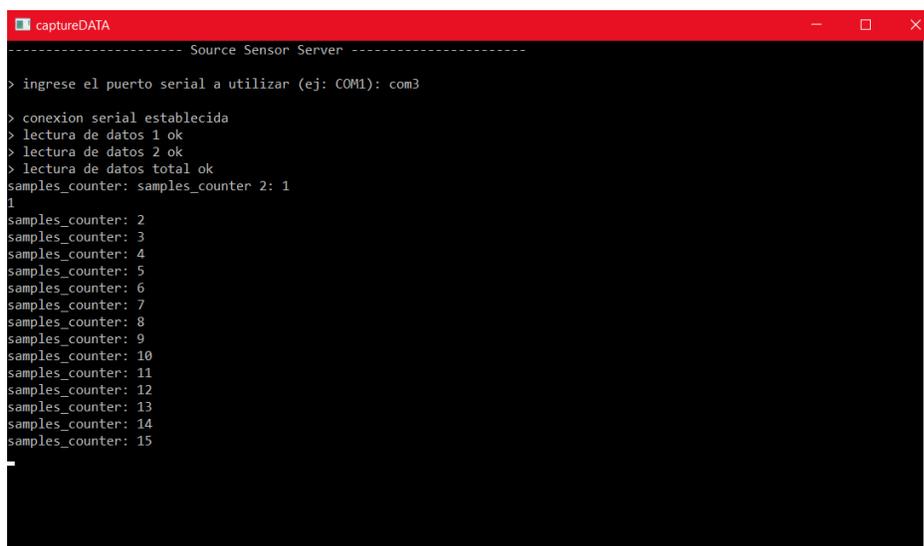


Figura 5.22: Aplicación que sirve de datos al sistema.

La aplicación sirve de datos de 14 sensores, de los cuales 9 son del tipo rápidos; Aceleración (x,y,z), giro (x,y,z) y mag ( x,y,z) que en adelante serán referidos como sensores tipo 1 y 5 sensores del tipo lentos , Temperatura(1,2 y 3) , Humedad y Peso , serán referidos como sensores del tipo 2.

En la Figura 5.23 se puede apreciar una consulta SQL a la base de dato local y se verifica la integridad de los datos que han pasado por todos los procesos descritos en el Capítulo 4.

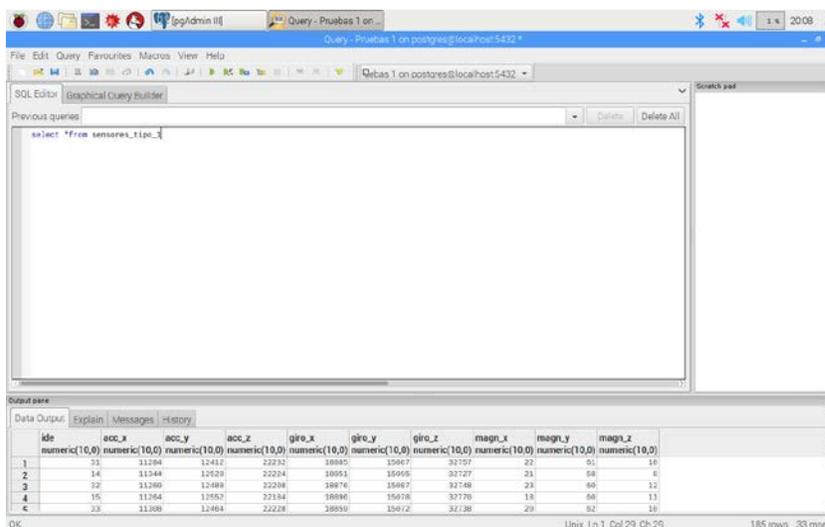


Figura 5.23:Tabla de datos tipo 1.

## Capítulo 5: Pruebas de campo y resultados.

---

### 5.3 Conclusión

Los transmisores inAir9 cumplen con las distancias establecidas en los criterios de diseño y los resultados obtenidos de las mediciones se encuentran dentro de la normalidad esperada. Si bien es cierto, los valores de RSSI son bajos en los extremos del huerto y que además son tipificados como “Bajísima cobertura” por la literatura existente, se tiene que la pérdida de paquetes de datos sólo alcanza el 8%. un valor aceptable en ese rango de distancias.

Las pruebas realizadas siguiendo el patrón de desplazamiento de la Figura 5.4. comprueban la funcionalidad del sistema frente al obstáculo que representan los arbustos de arándanos.

Los transmisores CPE510 pueden ser configurados para establecer un enlace inalámbrico de hasta 15 kilómetros, sin embargo, es necesario tener línea libre de obstáculos entre el emisor y el receptor de la señal.

Para este tipo aplicaciones Outdoor es de especial interés contar con los niveles de tensión adecuados para el normal funcionamiento de los equipos que forman parte de la estación base. La alimentación del Equipo CPE510 corresponde a una tensión de 24 v dc a través de Ethernet, esto significa que los datos y la energía eléctrica ocupan un sólo cable (distintos filamentos) y supone una ventaja para el montaje y conexionado.

## CAPÍTULO 6. CONCLUSIONES

El desarrollo del presente proyecto de título deja como resultado un prototipo funcional que implementa tecnología de comunicación inalámbrica para un sistema de adquisición de datos.

### 6.1 Alcances de diseño

El prototipo es adecuado para las condiciones de sitio para las cuales fue diseñado; huertos de arándanos u otro tipo de berries, donde es posible cubrir extensas áreas de monitoreo, distancias mayores a 350 metros, sin embargo, es necesario ajustar la potencia transmisión a un valor adecuado, considerando adecuada para la frecuencia de diseño del dispositivo Lora.

Las pruebas realizadas en distintos puntos de cosecha han generados resultados prometedores y satisfactorios para probar en primera instancia la funcionalidad de los transmisores lora frente a los obstáculos que pudiesen impedir la transmisión (arbustos de arándano) y por otro lado validar la estructura de red propuesta, determinado por los excelentes valores de RSSI, SNR y la nula pérdida de paquetes, al instalar la estación base en un punto central.

Si bien es cierto las pruebas entregaron buenos resultados, hay que señalar que el dispositivo no se encontraba instalado en la caja cosechera cuando se realizaron las pruebas, por tanto es necesario analizar el comportamiento de los transmisores Lora instalados en las condiciones de operación, es decir empotrado en el interior de la caja.

El protocolo de comunicación desarrollado es compatible con el dinamismo de cosecha que presentan las temporeras que trabajan el lugar, debido a la capacidad de adaptarse a los cambios provocados por el traslado del personal al interior de los huertos de arándanos.

La implementación del prototipo requiere una caja protección para los dispositivos que constituyen la puerta de enlaces a los menos IP55, por la gran cantidad de polvo, humedad y la alta posibilidad de precipitaciones que podrían dañar los circuitos electrónicos.

## Capítulo 6: Conclusiones

---

### **6.2 Alcances económicos**

La solución propuesta es viable de implementar en términos económicos debido al bajo costo que supondría la adquisición de miles de módulos inalámbricos. Tomando en consideración el área cubierta por transmisores descritos en la propuesta se tiene el que costo del área cubierta por estos equipos alcanza 45.8 USD/km<sup>2</sup> /nodo, indicador económico que frente a todos los beneficios de tener un control absoluto de todas las variables de la cosecha resulta despreciable.

### **6.3 Trabajo a futuro**

Se recomienda construir una versión del prototipo utilizando la tecnología LoRaWAN, que permitiría al servidor comunicarse directamente con el nodo transmisor y de este modo desarrollar nuevas estrategias para el control de acceso al medio.

Utilizar un dron como estacion base y agregar tecnología de posicionamiento con el objetivo de asegurar su ubicación en el centro geométrico de la red de sensores.

Es necesario también agregar un código de encriptación robusto, para evitar en primera instancia el robo de datos y la introducción de información errónea por parte de terceros.

## Bibliografía

- [1] T. V., Historia y evolución del Internet, vol. 33, 2004, p. 10.
- [2] P. J.R, Las ciencias sociales en Internet, J. D. EXTREMADURA, Ed., Mérida: JAVIER FELIPE S.L. (Producciones & Diseño), 2001.
- [3] ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA ALIMENTACIÓN Y LA AGRICULTURA, El estado mundial de la agricultura y la alimentación, Roma, 2015, p. 165.
- [4] P. B., «<http://www.sofoscorp.com>,» SofOS Organización de Conocimiento , 8 12 2017. [En línea]. Available: <http://www.sofoscorp.com/impacto-tecnologia-aplicada-agricultura>. [Último acceso: 21 03 2019].
- [5] C. Nansen, «"Using Remote Sensing to Optimize IPM in Greenhouses",» *Greenhousegrowe*, p. 1, 12 Febrero 2017.
- [6] Fabián, G. William, M. German U, «"Diseño de una red inalámbrica de sensores para apoyar actividades de agricultura de precisión en el jardín botánico de Cali",» Colombia, 2013.
- [7] Ministerio de Agricultura, «Panorama de la Agricultura chilena,» Odepa, Santiago, 2017.
- [8] FONDEF ID16I10206, «Diseño electrónico del módulo de monitoreo,» Temuco, 2018.
- [9] P. Vergano, R. Testing y W. Newall, Peach bruising: susceptibility to impact, vibration, and compression abuse, 1991, p. 2110–2116.
- [10] X. Lin y G. H. Brusewitz, «Peach bruise thresholds using the instrumented sphere,» vol. 10, n° 4, p. 509–513, 1994.
- [11] K. A. Sanford, P. D. Lidster, K. B. McRae, E. D. Jackson, R. A. Lawrence, R. Stark y R. K. Prange, «Lowbush blueberry quality changes in response to mechanical damage and storage temperature,» *J. Am. Soc. Hortic. Sci.*, vol. 116, n° 1, pp. 47-51, 1991.
- [12] Proyecto FONDEF ID16I10206, «Prototipo Final del DSS,» Temuco, 2018.
- [13] culturacion, «<http://culturacion.com/>,» [En línea]. Available: <http://culturacion.com/que-es-un-protocolo-de-red/>. [Último acceso: 20 3 2019].
- [14] S. W., Comunicaciones y redes de computadores 6ª, vol. 6, Madrid: PRENTICE HALL, 2001, pp. 31-35.
- [15] Semtech, «Sx1276 Datasheet,» 200 Flynn Road, Camarillo, CA 93012, 2016.
- [16] Digi-Key Corporation, «ArticleLibrary,» *Desarrollar con LoRa para aplicaciones IoT de baja tasa y largo alcance*, p. 1, 29 06 2017.
- [17] S. Corporation, «SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver,» *WIRELESS, SENSING & TIMING*, vol. 1, p. 16, 2015.
- [18] A. G. Santiago, «Diseño de un sistema de comunicación maestro-esclavo rs-485 con aplicación en labview,» Mexico, 2013.
- [19] K. Navarro, «<http://panamahitek.com>,» 15 10 2014. [En línea]. Available: <http://panamahitek.com/como-funciona-el-protocolo-spi/>. [Último acceso: 12 12 2018].

## Bibliografía

- [20] S. S., «<https://techlandia.com>,» [En línea]. Available: [https://techlandia.com/diferencia-frontend-backend-info\\_373239/](https://techlandia.com/diferencia-frontend-backend-info_373239/). [Último acceso: 2019 3 21].
- [21] The PostgreSQL Global Development Group, «[www.postgresql.org](http://www.postgresql.org),» 8 11 2018. [En línea]. Available: <https://www.postgresql.org/docs/9.5/protocol.html>. [Último acceso: 2019 1 3].
- [22] TP-Link , «<https://www.tp-link.com>,» [En línea]. Available: [https://www.tp-link.com/es/products/details/cat-37\\_CPE510.html](https://www.tp-link.com/es/products/details/cat-37_CPE510.html). [Último acceso: 21 03 2019].
- [23] V. Días, «<https://victordiaz.me>,» 15 11 2015. [En línea]. Available: <https://victordiaz.me/websocket>. [Último acceso: 15 1 2019].
- [24] E. Velázquez, «[pymesyautonomos](http://pymesyautonomos.com),» 17 1 2009. [En línea]. Available: <https://www.pymesyautonomos.com/tecnologia/para-que-sirven-los-certificados-ssl>. [Último acceso: 16 1 19].
- [25] E.-P. S. d. Internet, «[BricoGeek.com](http://BricoGeek.com),» [En línea]. Available: <https://tienda.bricogeek.com/teensy/549-teensy-32.html>. [Último acceso: 21 07 2018].
- [26] Modtronix Engineering, «<http://modtronix.com>,» [En línea]. Available: <http://modtronix.com/inair9.html>. [Último acceso: 21 3 2019].
- [27] MCI electronics , «<https://arduino.cl/que-es-arduino/>,» [En línea]. Available: <https://arduino.cl/que-es-arduino/>. [Último acceso: 2019 02 25].
- [28] E. R. Campos, «Informe de salida a terreno "Arandanos",» Temuco, 2017.
- [29] F. M. y. M. Vecchio, «An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks,» *The Computer Journal*, vol. 52, n° 8, p. 4, 30 Abril 2009.
- [30] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim y L.-M. Ang, «Fast and efficient lossless adaptive compression scheme,» *ELSEVIER*, pp. 7-8, 2014.
- [31] T. Q. Company, «[wiki.qt.io](http://wiki.qt.io),» [En línea]. Available: [https://wiki.qt.io/About\\_Qt/es](https://wiki.qt.io/About_Qt/es). [Último acceso: 2019 02 25].