



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO Y PROTOTIPADO DE PLATAFORMA DE MONITOREO DE VARIABLES
DE CALIDAD DE AGUA EN HUMEDALES URBANOS EN EL SUR DE CHILE

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

MATÍAS ANDRÉS MACAYA LEIVA

PROFESOR GUÍA:
CLAUDIO ESTÉVEZ MONTERO

MIEMBROS DE LA COMISIÓN:
SANDRA CÉSPEDES UMAÑA
MARÍA ORELLANA MONTECINO

Este trabajo ha sido parcialmente financiado por Fundación Legado Chile e Innova Corfo
Los Lagos

SANTIAGO DE CHILE
2018

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: MATÍAS ANDRÉS MACAYA LEIVA
FECHA: 2018
PROF. GUÍA: CLAUDIO ESTÉVEZ MONTERO

DISEÑO Y PROTOTIPADO DE PLATAFORMA DE MONITOREO DE VARIABLES DE CALIDAD DE AGUA EN HUMEDALES URBANOS EN EL SUR DE CHILE

La tecnología siempre ha ayudado al hombre a llegar más allá y superar sus capacidades. Actualmente, existen problemas en diferentes partes del globo donde la tecnología puede ser de gran ayuda pero no se ha incorporado como apoyo. Problemas como la escasez de agua, manejo de recursos, predicción y respuesta ante eventos como desastres naturales o fallas en una industria, son algunas de las situaciones donde nuevos avances en electrónica y telecomunicaciones pueden tener un gran impacto.

La combinación de bajos costos y tamaños de chips electrónicos y sensores, junto con el nuevo desempeño de tecnologías inalámbricas, permite la implementación de equipos que monitoreen diferentes sistemas llegando a lugares antes inaccesibles y generando información periódica y en tiempo real respecto del funcionamiento y estado de sistema. Esto permite saber que está ocurriendo en terreno, generar alertas preventivas, reaccionar ante eventos importantes y mediante un registro histórico de datos generados es posible anticipar que puede llegar a ocurrir.

Actualmente las soluciones disponibles para desplegar una red de monitoreo son pocas y de alto costo. Además, muchas de las opciones son privativas y no permiten un control propio sobre los datos. Esto no debe ser así, y se plantea la generación de equipos de monitoreo abiertos y de bajo costo. Estos equipos serán desplegados en una situación real de captura de datos en áreas de importancia natural en el sur de Chile.

En la ciudad de Llanquihue en la X Región de los Lagos al sur de Chile, existe un grupo de áreas naturales y humedales de gran importancia pero en muy mal estado. Ante esto, junto con la Fundación Legado Chile se implementará un plan de monitoreo de estos entornos que permitirá establecer una línea base del estado de salud de estos humedales, y que permitirá evaluar el impacto de futuras acciones de conservación y restauración de los espacios.

El sistema generará además una base de datos de información de alta importancia científica y no antes vista en Chile en estas cantidades, dado el gran número de sensores y la alta periodicidad de muestreo comparado con muestreos clásicos de humedales. Un efecto del sistema de monitoreo será el trabajo con la comunidad de Llanquihue, que interesada en el cuidado de sus humedales se ve motivada a conocer la información generada del monitoreo y que debe poder acceder a ella.

Se plantea entonces el prototipado de equipos que permitan desplegar un sistema de monitoreo de humedales en la ciudad de Llanquihue. Estos equipos deben ser de bajo costo dados límites de financiamiento y utilizarán tecnologías abiertas. Sus datos serán compartidos de manera pública en una plataforma web.

A mi madre, de cuyos cabellos grises yo soy una de las principales causas.

A mi hermana.

A mi padre, a toda mi familia y sus nuevos integrantes.

*A todas las nuevas amistades y los amigos que ya estaban. Les quiero, aunque no me sepa
sus cumpleaños y nadie lea esto.*

Al mar, las montañas, y otras cosas.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Hipótesis	2
1.3. Objetivo del Trabajo de Título	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
1.4. Alcances	2
1.5. Estructura de la Memoria	3
2. Revisión Bibliográfica	5
2.1. Marco Teórico	5
2.1.1. Wireless Sensor Network (WSN)	5
2.1.2. Low-Power Wide Area Network (LPWAN)	6
2.1.3. LoRa	7
2.1.3.1. LoRaWAN	8
2.2. Referentes	9
3. Humedales en Llanquihue	10
3.1. Humedales	10
3.1.1. Eutrofización	11
3.1.2. Monitoreo de Humedales	12
3.1.2.1. Variables a registrar	12
3.2. Humedales Urbanos en Llanquihue	13
4. Arquitectura del Sistema	16
4.1. Consideraciones de Diseño	16
4.2. Requerimientos	17
4.2.1. Requerimientos Generales	17
4.2.2. Requerimientos Específicos	17
4.3. Arquitectura	18
5. Diseño	21
5.1. Red Inalámbrica	21
5.2. Nodo Sensor	23
5.2.1. Empaque	23
5.2.2. Procesamiento	23
5.2.3. Sensores	24

5.2.4. Energía	25
5.3. Gateway	25
6. Implementación	27
6.1. Selección y Descripción de Componentes	27
6.1.1. Nodo	27
6.1.1.1. Carcasa	27
6.1.1.2. Procesamiento	28
6.1.1.3. Comunicación	29
6.1.1.4. Sensores	30
6.1.1.5. Energía	31
6.1.2. Gateway	33
6.2. Comentarios acerca de la Energía	34
6.3. Ensamble	35
6.3.1. Nodo	35
6.3.1.1. Conexiones	35
6.3.1.2. Problema 1 - Caída de Voltaje	35
6.3.1.3. Problema 2 - Consumo de aisladores	36
6.3.1.4. Prototipo	37
6.3.2. Gateway	37
6.4. Programación	39
6.4.1. Nodo	39
6.4.2. Gateway	39
7. Sistema para Estudio de Cobertura	42
7.1. Metodología	43
7.2. Diseño	44
7.2.1. Comunicación: LoRa	44
7.3. Implementación	44
7.3.1. Nodo LoRa	45
7.3.2. Aplicación Móvil (<i>Loreap</i>)	46
7.3.3. Gateway	48
7.3.4. Visualización	49
8. Base de Datos y Plataforma de Datos Abiertos	50
8.1. Base de Datos	50
8.2. Plataforma de Datos	51
8.2.1. Interfaz de Usuario	52
9. Pruebas, Resultados y Análisis	54
9.1. Pruebas de Cobertura	54
9.1.1. Primera Aproximación	54
9.1.2. Segunda Aproximación	55
9.1.3. Tercera Aproximación	57
9.2. Pruebas Prototipo del Sistema	59
9.3. Análisis Económico	61
Conclusiones	64

9.4. Sistema de monitoreo	66
9.5. Prototipos implementados	66
9.6. Pruebas realizadas	67
9.7. Trabajo Futuro	68
9.7.1. Trabajo Futuro Inmediato	68
9.7.2. Trabajo Futuro General	68
Bibliografía	68
A. Códigos	72
A.1. Código Gateway - Arduino	72
A.2. Código Nodo Monitoreo	78

Índice de Tablas

6.1. Comandos para Arduino más SX1276	39
9.1. Rendimiento de LoRa en lugares de interés en Llanquihue.	59

Índice de Ilustraciones

2.1. Tabla comparativa de tecnologías LPWAN. Junio 2017. Tomado de [1]	7
3.1. Humedales urbanos de Llanquihue (mapa ladeado en 90°)	14
3.2. Mapa satelital de Llanquihue, Enero de 2003. Misma escala que Figura 3.3 .	15
3.3. Mapa satelital de Llanquihue, Abril de 2017. Misma escala que figura 3.2 . .	15
4.1. Mapa de Llanquihue bajo grilla de 500 mt., zonas de conservación y posición tentativa de lugares de medición	19
4.2. Arquitectura del Sistema de Monitoreo	20
5.1. Diagrama de bloques nodo sensor	23
5.2. Diagrama de bloques gateway	26
6.1. Ejemplo de carcasas de PVC	28
6.2. Arduino Pro Mini	29
6.3. Módulo LoRa inAir9	29
6.4. Sensores de medición de variables internas del nodo	30
6.5. Elementos para medición de pH con sensor Atlas Scientific	31
6.6. Estimación inicial de consumo para un nodo de la red de monitoreo.	32
6.7. Autonomía energética del sistema vs Periodo de muestreo de variables	33
6.8. Raspberry Pi Zero	34
6.9. Lipo Rider Pro: Placa de administración de energía.	34
6.10. Conexiones componentes nodo	35
6.11. Caída de voltaje de alimentación efecto de los sensores externos	36
6.12. Prototipo de nodo sensor	38
6.13. Diagrama de conexiones gateway	38
6.14. Diagrama de flujo programación nodo	40
6.15. Diagrama de flujo programación Arduino gateway	41
7.1. Modos de uso del sistema para estudio de cobertura.	43
7.2. Modos de configuración LoRa. Tomado de [2]	45
7.3. Conexión Arduino Pro Mini con módulo HC-05	45
7.4. Módulo LoRa implementado para pruebas de cobertura	46
7.5. Aplicación Loreapp desarrollada para Sistema de Estudio de Cobertura LPWAN	47
7.6. Vistas del chat con Bot Telegram implementado en Gateway	49
8.1. Estimación de los datos generados por el sistema de monitoreo en 1 año. . .	51
8.2. Interfaz gráfica de página abierta de datos ambientales y de aguas.	53

9.1. Primera prueba de cobertura en Llanquihue. http://tabsoft.co/2De5gwA	55
9.2. Mapa de relieve de Llanquihue	56
9.3. Segunda prueba de cobertura en Llanquihue. http://tabsoft.co/2mFnNY5	57
9.4. Tercera prueba de cobertura en Llanquihue. http://tabsoft.co/2D3ZmtQ	58
9.5. Configuración para prueba de prototipo	60
9.6. Google Sheet con datos de medición remota	61
9.7. Datos de pH y Oxígeno Disuelto obtenidos de manera remota desde un recipiente de agua.	62
9.8. Datos de Temperatura y Humedad Ambiental obtenidos de manera remota .	62
9.9. Resumen de costo de nodos del sistema de monitoreo.	63
9.10. Resumen de costo de gateway	64
9.11. Comparación de costos entre equipo Libelium Devicelynk Smart Water Solution Kit y una oferta equivalente en base a los equipos desarrollados.	64
9.12. Libelium-Devicelynk Smart Water Solution Kit	65

Capítulo 1

Introducción

El presente informe documenta el trabajo realizado en el establecimiento de un prototipo para los elementos de una red de monitoreo de variables de calidad de agua en humedales urbanos en la ciudad de Llanquihue en la X Región de Los Lagos. Este trabajo se enmarca dentro del Plan de Acción Medioambiental llevado a cabo por la Fundación Legado Chile en la ciudad a orillas del Lago Llanquihue.

1.1. Motivación

Llanquihue es una pequeña ciudad entre Frutillar y Puerto Varas, de origen industrial y donde nace el Río Maullín, afluente por el que corren las aguas del lago hacia el océano Pacífico. Llanquihue posee una extensa área de espacios verdes de gran importancia ecológica, entre los cuales destacan cuatro humedales urbanos. Estos humedales albergan gran cantidad de flora y fauna nativa y son importantes lugares de avistamiento de aves en la zona.

Pero el mal uso y la falta de cuidado y consciencia de estos cuerpos acuáticos han generado la pérdida estimada de más del 60% de los humedales del mundo desde el siglo pasado. Es así que Fundación Legado Chile (en adelante FLC) toma acción y comienza a generar un plan para la conservación de estos entornos.

Este plan consta de 4 etapas a través de las cuales pretende ser un prototipo de innovación social para la conservación participativa del patrimonio natural en entornos urbanos. La Etapa 1 consiste en generar un diagnóstico participativo de la situación actual del ecosistema natural-urbano en la zona. Esta información da lugar a la Etapa 2, la proyección de estrategias de conservación a futuro. De esta manera se hace necesario poder evaluar y cuantificar el impacto ecológico de estas iniciativas.

Es así que nace el plan de monitoreo de humedales, el cual pretende medir variables de calidad de agua en humedales que junto con otros indicadores permitan generar un indicador del estado de salud, primero de los humedales, y luego de los demás espacios de importancia para la conservación en un entorno urbano, social y natural.

1.2. Hipótesis

A través de tecnologías abiertas disponibles en la actualidad, es posible generar una plataforma de monitoreo de entornos hídricos, cuyo desarrollo en términos de costos sea menor y su desempeño sea comparables con las soluciones comerciales actualmente disponibles en el mercado.

1.3. Objetivo del Trabajo de Título

1.3.1. Objetivo General

Desarrollar principios de diseño, arquitectura de la red y una primera versión de prototipos para la elaboración de un plan de monitoreo de humedales en Llanquihue.

1.3.2. Objetivos Específicos

1. Comprender el fenómeno de los humedales y su dinámica en la ciudad de Llanquihue.
2. Definir la arquitectura del sistema, requerimientos de los dispositivos e identificar las variables de diseño.
3. Implementar prototipos de los dispositivos que componen el sistema y hacerlos funcionar en conjunto.
4. Diseñar una plataforma abierta de datos que permita visualizarlos a la comunidad.

1.4. Alcances

El proyecto en el que se enmarca el desarrollo de este trabajo consiste en el diseño, implementación, instalación, puesta en marcha y operación de un sistema de monitoreo. Este sistema de monitoreo posee una arista de desarrollo tecnológico, y otra de trabajo en territorio.

El desarrollo tecnológico se encarga de producir los equipos y tecnología a instalar. El trabajo en territorio se encarga de coordinar, definir lugares de medición, solucionar aspectos legales e instalar infraestructura de los sensores.

En primer lugar los alcances de este trabajo son los de desarrollo tecnológico de la solución a instalar. No se definen aspectos como la forma de la instalación, los lugares específicos de medición, aspectos legales de instalación, trabajo con la comunidad, coordinación del proyecto, etc.

En segundo lugar, los alcances de este trabajo se ajustan a las fechas de avance de desarrollo del proyecto general. En base a esto se define que el objetivo de este trabajo es la generación de un prototipo. Se espera que el prototipo sea de características finales, pero en caso de no serlo se deben justificar las razones de implementación o diseño que llevaron al resultado desarrollado.

1.5. Estructura de la Memoria

El informe, incluyendo esta sección, y de aquí en adelante, se estructura de la siguiente manera:

Capítulo 1 - Introducción: Es el presente capítulo encargado de presentar el tema, generar la hipótesis, describir los objetivos y definir los alcances del trabajo.

Capítulo 2 - Revisión Bibliográfica: Este capítulo comienza con el marco teórico, aquí se repasan conceptos necesarios para comprender el informe. Temáticas como LoRa, IoT y LPWAN son definidos. Además se revisan referentes importantes a tener en cuenta.

Capítulo 3 - Humedales en Llanquihue: Se hace un repaso por el estudio del ecosistema de los humedales, definiendo importancia, efectos, influencia, comportamiento y variables más importantes que estructuran su funcionamiento. Esto es de gran importancia para entender la relevancia del trabajo a realizar y las variables medidas.

Capítulo 4 - Arquitectura del Sistema: En este capítulo se repasan los requerimientos que debe cumplir la red y la arquitectura de red propuesta que cumple con estas definiciones.

Capítulo 5 - Diseño: Describe la etapa de diseño, se repasan características que deberán tener los diferentes elementos. En base a los requerimientos se definen componentes, funcionalidades, decisiones de diseño y componentes a utilizar.

Capítulo 6 - Implementación: Una vez definidos los componentes, la arquitectura y las funcionalidades, el Capítulo 6 describe la implementación e integración de los diferentes sistemas.

Capítulo 7 - Sistema para Estudio de Cobertura: Se describe el desarrollo de una aplicación para la medición de cobertura en terreno utilizando LoRa. Esto es necesario ya que define la factibilidad del despliegue de una red en terreno.

Capítulo 8 - Base de Datos y Plataforma de Datos Abiertos: Además de la red de sensores, el sistema contempla un base de datos que almacene la información, y una plataforma abierta de datos ambientales y de agua.

Capítulo 9 - Pruebas, Resultados y Análisis: Se describen las pruebas realizadas con las diferentes tecnologías, se estudian resultados y se compara la solución desarrollada con otras opciones en el mercado.

Capítulo 2

Revisión Bibliográfica

2.1. Marco Teórico

A continuación se discuten conceptos importantes para la comprensión del trabajo a desarrollar.

2.1.1. Wireless Sensor Network (WSN)

Desde hace siglos, quizás milenios, que se realiza registro de datos y la tecnología siempre fue una ayuda esencial para esta tarea. Un primer ejemplo son instrumentos como los quipus que permitían registrar valores contabilizados mediante observación. Ya en tiempos modernos hacia mediados del siglo XIX, se distribuían instrumentos de medición del clima que permitían crear los primeros mapas climáticos de nuestra época [3]. En la década de los 80 del siglo XX los avances en electrónica propiciaron la aparición de los primeros dataloggers [4], estos instrumentos permiten registrar de manera automática variables externas y la almacenan en una memoria interna para su posterior recuperación.

Una vez que tecnologías inalámbricas de comunicación comenzaron a aparecer no tardó mucho en que estas se aplicarían a sistemas de medición de variables naciendo así las primeras redes inalámbricas de sensores (WSN por sus siglas en inglés).

Una WSN es entonces una red inalámbrica de nodos autoorganizados, conformada por dispositivos con capacidades sensitivas y dispuesta sobre un área de interés [5]. Existen variados tipos de nodos: nodos sensores, nodos repetidores, nodos de salida o gateways. Estas definiciones son conceptuales y en la práctica un nodo puede ejercer más de un rol a la vez.

Por lo general una red inalámbrica de sensores, para estudios ambientales, se conforma por dispositivos con limitaciones en sus capacidades: baja capacidad de procesamiento, poco ancho de banda, y tamaño reducido, son algunas de las características de un dispositivo de una WSN. Razones de esto son limitantes en los costos y la fuente de energía. Además puede

haber restricciones de tamaño o facilidad de transporte para instalación.

En muchos casos una WSN poseerá una cantidad considerable de nodos, por lo tanto los costos del dispositivo deben ser lo suficientemente bajos cómo para que la implementación de la red tenga sentido. Esto implica seleccionar los componentes de capacidades justamente necesarias para desempeñar su trabajo y no más que eso.

Los dispositivos o nodos de una WSN serán dispuestos en lugares en donde por lo general no existirá un suministro energético constante. La mayoría de las veces además son lugares de difícil acceso, o la cantidad de nodos es tal que realizar un cambio de baterías es una tarea muy extensiva. Por esto, el suministro de energía la segunda mayor limitante. Los nodos deberán ser lo menos consumidores posibles lo que implica los elementos con prestaciones más simples.

Las limitaciones y características recién descritas comienzan a armar un marco de especificaciones, definiciones, requerimientos y consideraciones de diseño que comienzan a caracterizar a las WSN como un áreas de gran desarrollo.

Las WSN se pueden clasificar en 3 grandes tipos [6]:

1. **Definidas por tiempo:** Adquiere y comunica mediciones periódicamente.
2. **Definidas por evento:** Notifica en base a ocurrencia de eventos de interés.
3. **Definidas por consulta:** Envía información sólo ante una petición previa.

2.1.2. Low-Power Wide Area Network (LPWAN)

Impulsado en parte por el desarrollo de redes inalámbricas de sensores y por el desarrollo del Internet de las Cosas (IoT, por sus siglas en inglés). El problema de comunicar dispositivos a grandes distancias y consumiendo poca batería se hizo cada vez más relevante. Este tipo de escenario se define como Redes de Larga Distancia y Bajo Consumo o LPWANs por sus siglas en inglés.

La generación y éxito en el desarrollo de tecnologías para LPWANs se ha convertido en uno de los mayores habilitantes para el desarrollo de proyectos de IoT y han encontrado un gran espacio de desarrollo en la industria especialmente.

LPWAN define redes de comunicación a grandes distancias, para dispositivos que consumen baja potencia, poseen bajas tasas de transmisión y poseen cortos ciclos de trabajo.

Principales características de una LPWAN se puede resumir de la siguiente manera:

- Bajo consumo de energía
- Extensa cobertura
- Dispositivos de bajo costo
- Escalabilidad
- Bajas tasas de transmisión

- Transmisión a intervalos prolongados
- Por lo general, baja movilidad
- Suelen soportar productos o servicios

Dado al amplio impacto que estas tecnologías han generado en la industria y otras áreas de importante desarrollo, existe una fuerte competencia y oferta de tecnologías para LPWAN. Se resumen en la Figura 2.1 las tecnologías en desarrollo a Junio de 2017.







	 LoRa / LoRaWAN	 Sigfox	NB-IoT	 LTE-M	 RPMA	 Weightless-P	 Symphony Link
Origin	France	France	USA (Global)	USA (Global)	USA	UK	USA
Proprietary or open	LoRa – proprietary LoRaWAN – open	Net – proprietary Devices – open	Open	Open	Proprietary	Open	Proprietary
Cellular	No	No	Yes	Yes	No	No	No
Spectrum	Unlicensed	Unlicensed	Licensed	Licensed	Unlicensed	Unlicensed	Unlicensed
Range, km	urban: 2-5 rural: 15	urban: 3-10 rural: 30-50	urban: 1-5 rural: 10-15	urban: 2-5	urban: 1-3 rural: 25-50	urban: 2	urban: 2-5 rural: 15
Speed, uplink / downlink	50 kbps / 50 kbps	300 bps / –	250 kbps / 250 kbps	1 Mbps / 1 Mbps	634 kbps / 156 kbps	100 kbps / 100 kbps	100 kbps / 100 kbps
Power consumption	●●●	●	●	●●●	●●	●	●●
Security	●●	●●	●●●	●●●	●●●	●●●	●●●
Availability of devices	●●	●●●	●●	●	●●	●	●●
Price*	●●	●	●●	●●●	●●●	●	●●
Areas of application	Precision farming, manufacturing automation, pipeline monitoring	Predictive maintenance, capacity planning, demand forecasting	Electric metering, manufacturing automation, retail PoS	tracking objects, wearables, energy management, utility metering, city infrastructure	Digital oilfield, connected cities, usage-based insurance, agriculture	Smart grid, healthcare, automotive, smart cities, asset tracking	Industrial control systems, lighting control, alarm systems
Supporting companies	IBM, Semtech, Cisco, HP, Orange, Kerlink, Actility	STMicroelectronic, Texas Instruments, Atmel, Silicon Labs	Huawei, Ericsson, Qualcomm, Vodafone	Verizon, AT&T, Nokia	Ingenu	Accenture, Sony Europe, uniik, ARM, Telensa	Link Labs

Figura 2.1: Tabla comparativa de tecnologías LPWAN. Junio 2017. Tomado de [1]

2.1.3. LoRa

LoRa es una especificación de LPWAN desarrollada por la empresa francesa Cycleo y patentada por primera vez en 2008. La empresa fue luego adquirida por Semtech, empresa estadounidense dedicada al desarrollo de semiconductores. Actualmente desarrolla y comercializa chips de comunicación y gateways que implementan el protocolo. Además de Semtech, otras empresas que pagan derechos de patente también desarrollan soluciones LoRa.

LoRa (Long Range) define especificaciones de capa física diseñadas especialmente para aplicaciones de bajo consumo y grandes extensiones geográficas [7]. Estas permiten enviar paquetes de datos a kilómetros de distancia. Se especializa en el tráfico up-link desde los endpoints al gateway.

Utiliza un esquema de modulación de ensanchamiento de banda [8], SS, por sus siglas en inglés. Esta modulación utiliza una secuencia de bits (chirp) cuya frecuencia es mayor que la señal original y lo aplica sobre esta. Esto genera que el espectro de ancho de banda de la señal original se esparza en un espectro mucho más amplio ocupando más ancho de banda pero a menor potencia. La nueva señal es más robusta en varios aspectos, primero a interferencia de fuentes con frecuencias puntuales dado que esta repartida en el espectro de frecuencia. También, a degradaciones de canal por multicamino, doppler o ISI (Inter Symbol Interference).

LoRa no posee especificaciones de capa MAC, es decir, por si misma no es capaz de generar redes de sensores, ni coordinar dispositivos, asignar prioridades, controlar el flujo, definir permisos de acceso, etc., todas las funciones de capa MAC y superiores no se encuentran disponibles con la especificación LoRa.

LoRa posee tres parámetros de configuración que permiten modificar la manera en que se modulan los datos, esto permite hacer un trade-off entre: distancia de comunicación, tasa de transmisión y consumo energético [2] [9]. Utilizando sus siglas en ingles, los parámetros son:

- **Bandwidth (BW):** Es el ancho de banda ocupado por la señal en el espectro. Mayor BW implica mayor tasa de transmisión. Existen 3 anchos de banda posible: 125 kHz, 250 kHz y 500 kHz.
- **Spreading Factor (SF):** Define el tamaño del chirp utilizado. Toma valores de 6 a 12. Mayor SF implica mayor capacidad de interpretar los símbolos del mensaje en el receptor. También un mayor SF aumenta el tiempo de envío del paquete, lo cuál aumenta el consumo.
- **Coding Rate (CR):** Se relaciona con la redundancia en el mensaje enviado. Mayor redundancia disminuye la probabilidad neta de error del canal, pero aumenta la información a enviar. Existen 4 CR posibles: 4/5, 4/6, 4/7 y 4/8. Menores valores (el menor valor es 4/8) de CR implican mayor redundancia del mensaje.

2.1.3.1. LoRaWAN

Sobre la capa física de LoRa se sitúa en la capa MAC el protocolo LoRaWAN. Esta permite generar estructuras de red con todas las funcionalidades que esto conlleva: control de flujo, control de acceso, orden de utilización, entrega confiable de paquetes, entre otros.

Identifica a cada uno de los dispositivos de la red mediante una dirección dinámica de 32 bits asignada al momento de unirse a la red. Implementa también dos capas de seguridad tanto en red cómo en aplicación utilizando el algoritmo de cifrado AES-128 [10].

LoRaWAN define la existencia de tres clases de dispositivos:

1. **Clase A:** Dispositivos que soportan comunicación bidireccional entre un dispositivo y un gateway. Es posible enviar paquetes al gateway (*Up-link*) en cualquier momento. Y es posible recibir paquetes (*Down-link*) desde el gateway al dispositivos solo en ventanas de espera después de enviar un paquete.

2. **Clase B:** Dispositivos que extienden la funcionalidad de la clase A mediante la implementación de ventanas periódicas para comunicación *Down-link*.
3. **Clase C:** Extiende la clase B definiendo una ventana continua de recepción de mensajes *Down-link* solo exceptuando los momentos en que transmite un paquete.

2.2. Referentes

Se lista una serie de iniciativas que por sus características inspiran y sirven de ejemplo para este trabajo:

1. **Libelium** [11]: Empresa española que nació el 2012 con el objetivo de proveer dispositivos de medición para IoT. Han crecido de manera exponencial ocupando actualmente el liderazgo de provisión de este tipo de equipos a nivel mundial.
Comenzaron con un desarrollo abierto (open-source) pero fueron cerrándolo a medida que crecieron. Poseen una cartera de productos específicos para diferentes áreas listas para ser instaladas. Proveen además los servicios de mantención y servicio técnico.
2. **Riffle Design Philosophy** [12]: Es una iniciativa abierta de desarrollo de dataloggers para estudio de variables de agua. Poseen un gran desarrollo de pruebas, sensores y dispositivos propios además de una filosofía de diseño con marcos de desarrollo muy claros y pasos que se justifican con base en la investigación del medio acuático.
3. **Waziup** [13][14]: Iniciativa de uso de tecnologías de IoT para monitoreo de cultivos en África. Poseen un extenso desarrollo de dispositivos para medición de variables en terreno, todos con una gran documentación.
4. **Redes de sensores para monitoreo hidro-meteorológico en zonas remotas, Latina UC** [15][16]: En la Quebrada de Ramón hacia el sector oriente de Santiago, docentes y alumnos de la Universidad Católica han instalado una red de sensores que permite monitorear variables hidro-meteorológicas con base en las cuales se generan modelos y se levantan alertas ante peligros de aluviones o deslizamientos de tierra en la zona.
Poseen cerca de 18 estaciones midiendo continuamente estas variables y subiéndolas a Internet. A partir de los datos investigadores del área hidrológica generan modelos de predicción y alerta temprana.

Capítulo 3

Humedales en Llanquihue

3.1. Humedales

Son uno de los entornos más productivos de la naturaleza. Los procesos biológicos, químicos y geológicos son tan intensos en estos ecosistemas que ayudan a filtrar y mejorar la calidad del agua de ríos, esteros y napas con una eficiencia única. Existen humedales artificiales especialmente diseñados para imitar las condiciones de uno real los cuales son utilizados para los fines anteriormente descritos [17].

Los humedales poseen una diversidad biológica inmensa, más del 40 % de las especies del mundo y el 12 % de todas las especies de animales se hallan en humedales de agua dulce. Estos aportan con un 40 % de los servicios ecosistémicos renovables del planeta, ocupando solo un 1.5 % de la superficie total del planeta. Los servicios consisten en su trabajo como: filtros naturales de agua, reponedores de aguas subterráneas y fuentes inmensas de biodiversidad.

Dada su capacidad de absorción actúan como esponjas, en invierno absorben y acumulan agua de diversas fuentes, conteniendo así aludes e inundaciones, y luego en verano la liberan en periodos secos. En diferentes periodos del año son lugares de asentamiento de aves migratorias que viajan miles de kilómetros para llegar a descansar a estos lugares.

Los humedales son parte de la Convención de Ramsar, esta convención se organizó en la década de los 60' dada la creciente preocupación por la pérdida de los humedales a nivel mundial con el objetivo de generar acuerdos y un marco de protección de estos sitios. Según la convención Ramsar un humedal se define como “extensiones de marismas, pantanos y turberas, o superficies cubiertas de agua, sean estas de régimen natural o artificial, permanentes o temporales, estancadas o corrientes, dulces, salobres o saladas, incluidas las extensiones de agua marina, cuya profundidad en marea baja no exceda los 6 metros” [18].

Se entienden los humedales como cuerpos en los cuales el agua tiene un papel determinante en su estructura y comportamientos, también será su fuente de nutrientes. La existencia de ciclos en la abundancia de agua también es un factor importante en las dinámicas de estos.

Las transformaciones de los nutrientes en humedales se caracterizan por regímenes anaeróbicos de funcionamiento a causa de la falta de oxígeno en el suelo del humedal debido a que estos se encuentran cubiertos de agua. La causa de que estos procesos biogeoquímicos de los nutrientes ocurran a nivel del suelo del humedal tiene que ver con la baja altura de agua que caracteriza a estos cuerpos. La falta de oxígeno como oxidante natural, propicia la aparición de microorganismos anaeróbicos que utilizan otras fuente de oxidantes inorgánicos. Así este régimen de reducción de componentes inorgánicos en base a la cantidad de oxígeno disponible se mide en base al Potencial de Óxido Reducción [19].

3.1.1. Eutrofización

Uno de los problemas más importantes en humedales y otros cuerpos de agua tiene que ver con la eutrofización [20]. Esta consiste en el incremento repentino en la cantidad de nutrientes en un humedal lo que genera un crecimiento explosivo de algas y materia orgánica que se apropian del ecosistema.

Los nutrientes a los que se hace referencia son el fósforo y nitrógeno. Estos nutrientes son los más importantes habilitantes de la producción de un sistema, por esto son denominados nutrientes limitantes, ya que su concentración, abundancia o escasez, limita la actividad orgánica de los sistemas, la cuál se mantiene controlada siempre que no se alteren las concentraciones naturales de estos nutrientes en el ambiente.

A través de la actividad humana, especialmente en humedales urbanos, suceden eventos de incorporación de nutrientes a los cuerpos de agua a través de desagües, basura, desechos, jabón o fertilizantes, entre otros. Esto genera la incorporación de nutrientes, entre ellos nitrato y fósforo al sistema.

El aumento en estos nutrientes lleva a una proliferación de plantas y algas fotosintéticas que rápidamente en su crecimiento proceden a cubrir la zona fótica y superficial del humedal. Las algas poseen un color verdoso que bloquea el paso de la luz a zonas más profundas. Ante la falta de luz la vegetación bajo el nivel superficial comienza a morir y a depositarse en el fondo del humedal.

La proliferación de algas comienza a consumir los nutrientes disponibles y estas comienzan a morir y a depositarse en el fondo con las demás plantas y algas muertas. Toda esta materia orgánica depositada en el fondo es consumida y descompuesta por bacterias consumidoras de oxígeno, proceso a través del cual generan nuevos nutrientes. Las bacterias consumen el oxígeno disponible en el ambiente lo que acaba por producir un entorno anóxico, es decir, falta de oxígeno bajo el cual la fauna presente muere.

Así el sistema entra en un ciclo retroalimentado de algas que consumen nutrientes y bacterias que consumen material orgánico muerto denominado ciclo de floración de algas. En ciclos extremos de eutrofización la acumulación de material en el fondo del humedal puede ser tal que genera una gradual elevación del terreno llegando a hacer desaparecer al cuerpo de agua afectado por la eutrofización.

3.1.2. Monitoreo de Humedales

Existen variados métodos para llevar a cabo el monitoreo de humedales. El más común son expediciones a terrenos donde se miden variables químicas y físicas del agua de manera directa, se realizan estudios de población de invertebrados y se toman muestras de agua las cuáles son llevadas para estudio en el laboratorio.

Se han propuesto variados métodos de monitoreo y prospección remota de humedales muchos de los cuáles utilizan imágenes satelitales para definir la localización y extensión de humedales en zonas remotas. La gran variedad de distintos humedales también motiva a generar herramientas automáticas de clasificaciones de estos entornos [21]. A través de procesos de procesamiento de imágenes se ha logrado incluso estimar ciertas variables como turbidez del agua y contenido de clorofila de esta [22].

La utilización de redes de monitoreo inalámbricas permiten medir variables de tipo abióticas de manera remota. Estas son variables de características física o químicas como temperatura y pH. La medición de otro tipo de variables como agentes biológicos todavía no es posible dado que existe una limitación en el desarrollo tecnológico de sensores para estas medidas.

3.1.2.1. Variables a registrar

Las variables de interés a medir son las siguientes, se ordenan por nivel de importancia:

1. **Nitratos y Fosfatos:** Son los nutrientes limitantes de ecosistemas acuáticos y terrestres. Esto se traduce en que son uno de los factores más importantes encargados de controlar la actividad de microorganismos en un sistema. Un exceso de estos nutrientes puede desencadenar en un incremento de algas en el agua y derivar en un proceso de eutrofización.
2. **Oxígeno Disuelto:** Es consumido por organismos aeróbicos como peces, zooplancton y bacterias que descomponen materia orgánica. Valores menores a 3.5mg/L no permiten la vida en el entorno hidrológico por lo que lo recomendado es un valor mínimo de 5mg/L [23].
3. **Turbidez:** Indica la capacidad del agua para dispersar radiación solar por la presencia de partículas en suspensión. La vegetación situada bajo el agua necesita de luz para poder realizar fotosíntesis. Un agua de mayor turbidez dejará pasar menos luz al fondo acuático lo que disminuirá la actividad de plantas acuáticas y determina su posible muerte. Se mide en Unidades Nefelométricas de Turbidez (NTU).
4. **Temperatura:** Por un lado, tiene una influencia directa sobre organismos vivos. No existe un valor correcto, depende del ambiente que se estudie. De alejarse mucho del valor normal para el objeto de estudio, puede generar la muerte o desplazamiento de los organismos locales. En segundo lugar, posee una relación inversa con el Oxígeno Disuelto.
5. **pH:** Representa la concentración de iones de hidrógeno H^+ en el agua. Entre sus propiedades se encuentran que influye en la estructura y descomposición de sustancias

orgánicas, regula procesos biológicos mediados por enzimas como la fotosíntesis y la respiración, regula la disponibilidad de nutrientes esenciales como nitratos y fosfatos, y afecta o regula la estructura y función de macromoléculas y organelos como proteínas estructurales y sistemas de la membrana y pared celular [24]. Se mide en una escala de 0 a 14, y valores bajo 4 y sobre 8 son perjudiciales para los sistemas.

6. **Electroconductividad:** Representa el nivel de sales disueltas en el agua y se mide evaluando la capacidad del agua para conducir electricidad a través de los iones de las sales disueltas. Variaciones fuera de los valores normales generan descompensaciones en el equilibrio osmótico de los organismos con el medio, las cuales deben ser compensadas lo que genera un sobre esfuerzo y un desgaste. Un máximo legal común es de $2,000 \mu S/cm$ [23].

El objetivo es poder medirlas todas pero pueden existir restricciones en las tecnologías disponibles y costos.

3.2. Humedales Urbanos en Llanquihue

Llanquihue es una ciudad de 19.093 habitantes localizada a orillas del lago Llanquihue en la X Región de Los Lagos. Se puede observar en la Figura 4.1 su perfil alargado de norte a sur por la costa, el cuál es dividido por el nacimiento Río Maullín. Llanquihue posee una historia de origen industrial y ha crecido al margen del explosivo desarrollo turístico de ciudades vecinas como Frutillar y Puerto Varas. Esto se combina con desafíos y condiciones estratégicas desde el punto de vista ambiental. Es hogar de un grupo de humedales urbanos y una zona de áreas como objetivo de conservación que le otorgan el potencial de ser la ciudad con más áreas verdes por persona de Chile. Así mismo, es el nacimiento del Río Maullín y desembocadura del lago Llanquihue, esto se presenta como oportunidad y desafío.

Existe una creciente preocupación acerca de las condiciones sanitarias de las aguas del Lago Llanquihue, y Llanquihue como desembocadura del lago se encuentra en un punto clave para esta problemática. Por esto también el desarrollo de conocimiento y tecnologías que se relacionen en torno a la temática del agua resulta interesante en este contexto. Y en un contexto mundial, donde este tipo de problemas suscita un creciente interés.

En Llanquihue existen 4 humedales urbanos indicados en la Figura 3.1. Estos están en medio de la ciudad y son los principales objetos de conservación dentro de una red más grande de áreas verdes que rodea la zona.

El Humedal Baquedano al extremo norte de la ciudad es el más pequeño de todos. Se seca durante el verano y en invierno recibe suministro de aguas lluvias de la ciudad [25]. El Humedal El Loto es el más grande y conocido por la gente, tiene la forma de una laguna y posee agua de forma constante durante todo el año. El Humedal Los Helechos no posee un espejo de agua continuo y se compone de una densa vegetación lacustre. El Humedal Las Ranas en la orilla sur desemboca en el Río Maullín y posee características de laguna.

Muchos de los humedales no son reconocidos como tales por la gente de la zona y se ignora

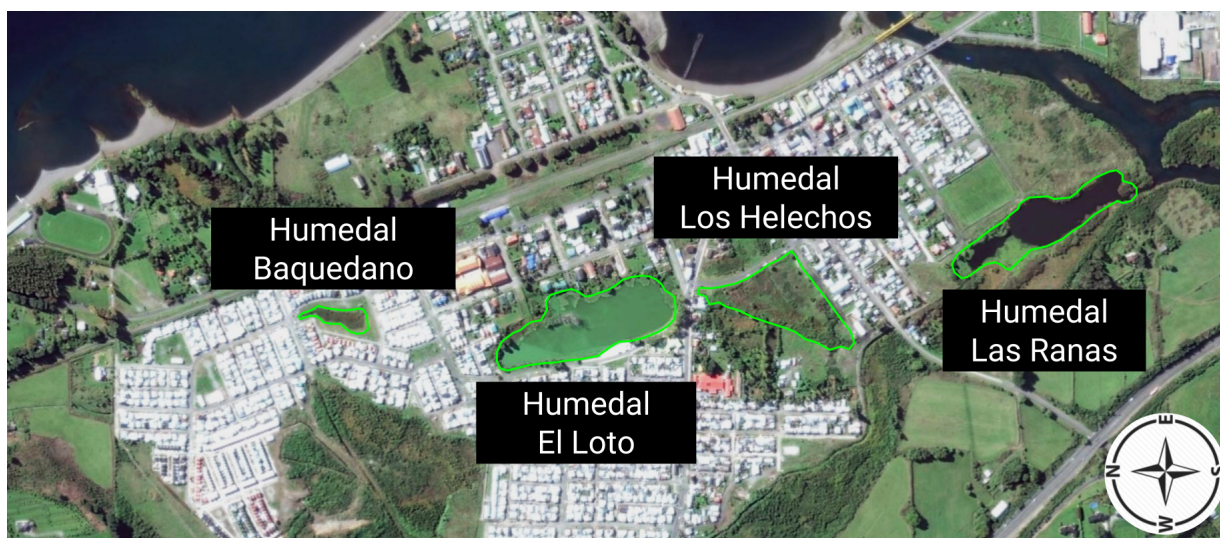


Figura 3.1: Humedales urbanos de Llanquihue (mapa ladeado en 90°)

la importancia y servicios que estos prestan a la comunidad. De hecho muchas veces se asocian a pantanos con los prejuicios negativos que esto conlleva, y son utilizados como basurero y depósito de escombros. Especialmente el Humedal Los Helechos y las zonas circundantes al Humedal Las Ranas.

Los humedales también soportan una amplia actividad de aves y son un creciente polo turístico en la zona para el avistamiento de aves. Cerca de 70 especies de aves son posibles avistar. Los humedales de la zona reciben a un 50 % de las aves migratorias que descansan en Chile.

Llanquihue si bien no se enfrenta a un explosivo crecimiento poblacional todavía, es una ciudad en expansión lo cual genera el uso de nuevos suelos. Esto es un llamado atención si se desea crecer de manera responsable y en una relación conjunta con el medio ambiente.

Se observa en las Figuras 3.3 y 3.2 (ambos mapas se encuentran en la misma escala), la expansión urbana sufrida por Llanquihue y la depredación y ocupación de algunos sitios de amplia importancia ecológica. Se observa en la parte norte de la ciudad hacia el año 2003 existía una zona de áreas verdes que comprendían un humedal comparable en tamaño al Humedal El Loto en ese momento. Al observar esta misma área en el año 2017, se da cuenta de la agresiva urbanización para con este cuerpo de agua, viéndose reducido en más de un 50 % de su tamaño original. Por esto es importante tomar consciencia al día de hoy cuáles son los sitios que queremos proteger y generar una estrategia de desarrollo que tome en cuenta estos factores.



Figura 3.2: Mapa satelital de Llanquihue, Enero de 2003. Misma escala que Figura 3.3



Figura 3.3: Mapa satelital de Llanquihue, Abril de 2017. Misma escala que figura 3.2

Capítulo 4

Arquitectura del Sistema

En este capítulo se desarrollan aspectos generales de diseño del sistema. Una red de sensores es un sistema complejo, y debe sortear variados desafíos como la comunicación, topología, manejo de la energía, fuentes energéticas y costos, entre otros.

Además, el desplegar una red de sensores en terreno conlleva a una serie de desafíos adicionales [26], condiciones adversas y variantes del clima, interacción con elementos del entorno como fauna y personas remarcan la necesidad de que el sistema sea capaz de autorevisarse y ejecutar rutinas de manejo de eventos y errores.

Existen dos funciones primordiales de la red a desplegar, estas definiciones comienzan a dar forma a variables de diseño dentro del sistema:

1. Registrar periódicamente variables del ambiente
2. Comunicar y almacenar datos de manera confiable

4.1. Consideraciones de Diseño

Algunas consideraciones importantes a tener en cuenta para el diseño del sistema son:

- Las variables a medir en los humedales poseen un comportamiento dinámico lento por lo que no es necesario una alta frecuencia de muestreo. Un periodo de muestreo de 30 minutos es una buena recomendación.
- Los datos serán utilizados con fines científicos por lo que las mediciones deben ser de buena calidad.
- Los humedales son sitios de fácil acceso. Esto implica que es posible un recambio periódico de baterías.
- Algunos humedales son sitios turísticos y todo ellos son de uso público. Además son sitios de alta y activa diversidad de vida animal. Esto define una política de menor intervención posible de los espacios por parte de la red de monitoreo desde los puntos

de vista ecológico, paisajístico y social.

- Llanquihue posee condiciones climáticas adversas de fuertes lluvias y vientos gran parte del año exceptuando periodo de verano.
- La información generada no es privada, a priori no es necesario cifrarla para su comunicación.

4.2. Requerimientos

Se desprenden de todo lo visto hasta ahora, requerimientos que el sistema debe cumplir para poder desempeñarse de manera efectiva y cumplir los objetivos. Existen requerimientos generales, estos le permiten al sistema ser lo suficientemente amplio para ser también apto de funcionar en otros entornos, lugares y condiciones. Y existen una serie de requerimientos específicos dados por la logística del proyecto, plazos, el diseño de la instalación y las funciones que se esperan de este.

4.2.1. Requerimientos Generales

- Nodos de bajo costo, con capacidad de auto chequeo de sus sistemas y reinicio ante errores.
- Nodo sensor de bajo consumo energético y con memoria suficiente para almacenar una cantidad de mediciones pasadas.
- Gateway de bajo costo con capacidad suficiente para soportar funcionalidades extras, como: almacenamiento de los datos, procesamiento de datos, manejo de interoperabilidad con otros protocolos e interfaces, etc.
- Gateway que garantice alta disponibilidad del servicio de comunicación.
- Configuración remota de los nodos sensores y actualización remota del firmware del gateway.
- Sistema que asegure la entrega y almacenamiento confiable de los datos de cada medición.

4.2.2. Requerimientos Específicos

Los requerimientos específicos hacen referencia a aquellos aspectos del sistema que tienen que ver con esta implementación en particular de un sistema de monitoreo. Además, el proyecto estipula una serie de metas mínimas que deben cumplirse lo que fue acordado con el fondo Corfo de Innovación Social que financia el sistema.

Como definiciones de diseño desarrolladas junto al equipo técnico para este plan de monitoreo se tienen:

- Nodo sensor debe ser de reducido tamaño. Espacio disponible para electrónica de máximo $8cm \times 8cm \times 20cm$.
- Detección automática de eventos externos en terreno cómo movimiento extraño de algún nodo es deseable.
- Vida útil de funcionamiento de nodos sensores de mínimo 3 meses.
- Medición de variables a más de 10 metros del nodo.

Como requerimientos dados por las bases del proyecto se tiene que:

- Se necesitan 9 puntos de monitoreo en terreno.
- La red de monitoreo debe funcionar continuamente por al menos 2 años.
- El costo total del sistema debe ser menor a CLP\$7.000.000.

4.3. Arquitectura

La red debe ser capaz de funcionar en el territorio y enviar datos capturados de manera confiable a un servidor a través del gateway. Esto cumpliendo con los requerimientos de una aplicación IoT de monitoreo ambiental de bajo costo, alta disponibilidad del servicio y bajo mantenimiento.

Para entregar la información de manera confiable al servidor, el sistema debe implementar protocolos de comunicación que permitan la correcta entrega. Además cada elemento por el que pasan los datos debe poseer capacidad de almacenamiento interna que le permita guardar últimas mediciones, cómo medida de seguridad antes caída de algún elemento del sistema.

Son necesarios 9 nodos emplazados en el territorio de Llanquihue. Los nodos se situarán en áreas de primera prioridad de conservación destacadas en rojo en la Figura 4.1.

A futuro también debe considerar áreas delineadas en morado y en ese sentido la red debe proveer cobertura a esas zonas igualmente. Se pueden observar en la figura las dimensiones del territorio mediante una grilla cuadrículada cada 500 metros . Tecnologías de conectividad LPWAN proveen alcances de kilómetros de distancia, a un bajo costo, bajo consumo energético y alta disponibilidad de servicio.

Cada terreno en el que se instala una red LPWAN es diferente y sus características tienen un fuerte impacto en el rango real de cobertura de la tecnología [27]. Su geografía y elementos, arboles y cuerpos de agua, y el área que se desea cubrir son claves a la hora de desplegar la red. Es importante evaluar la factibilidad de la red en terreno previo a su instalación, para esto será necesario contar con un sistema de medición de cobertura. Tener esto en cuenta.

La arquitectura del sistema se presenta en la figura 4.2. Esta entrega funcionalidades de captura de variables, conectividad en terreno, soporte de datos en un servidor externo y visualización de los datos mediante una plataforma abierta.

Los nodos sensores o *end-points* funcionarán como elementos independientes en terreno,

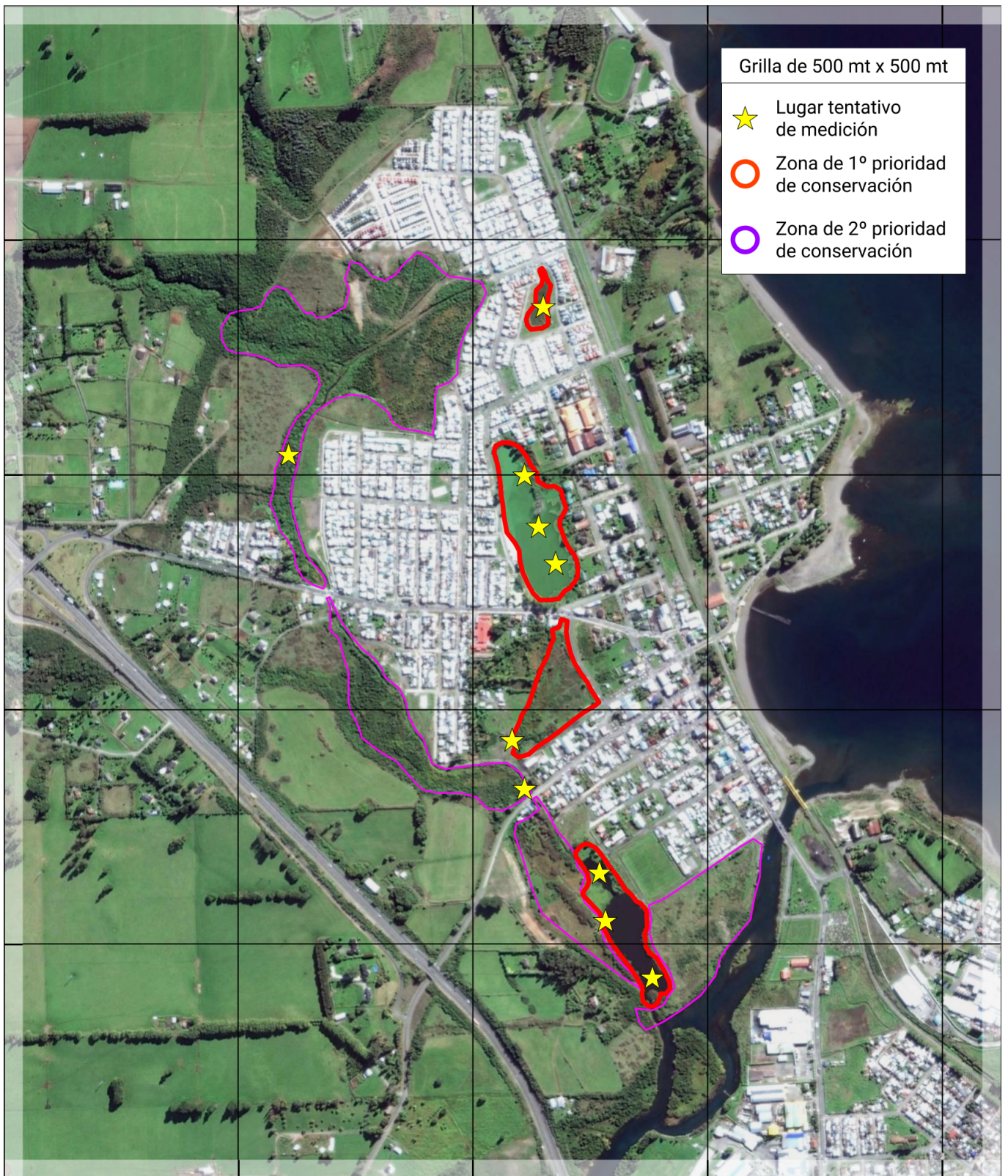


Figura 4.1: Mapa de Llanquihue bajo grilla de 500 mt., zonas de conservación y posición tentativa de lugares de medición

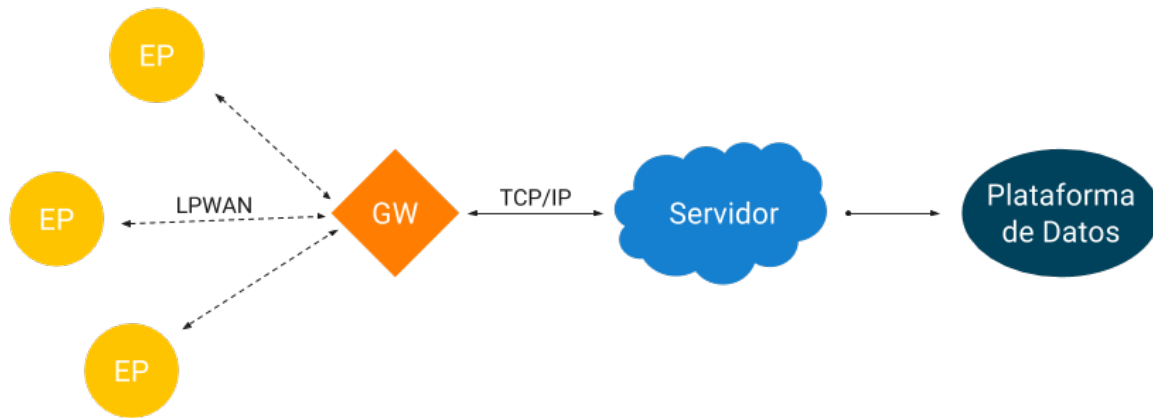


Figura 4.2: Arquitectura del Sistema de Monitoreo

con una fuente de energía limitada trabajarán en ciclos, la mayor parte de estos estarán en reposo, luego capturarán información y la enviarán, todo de manera periódica. Los nodos deben poder responder ante eventos como movimientos inesperados mediante la notificación de estos al gateway. Existirá un solo gateway utilizando una topología de tipo estrella. El gateway siempre atento a paquetes los recibirá, almacenará localmente y los subirá a una base de datos en Internet. El gateway tendrá un lugar fijo y no se encontrará limitado ni en su energía ni es su ancho de banda hacia Internet.

El sistema debe ser robusto y se debe garantizar el correcto almacenamiento definitivo de cada medición. Así los subsistemas deben prever y reaccionar ante casos de falla tanto internos como por agentes externos. Además, cada componente debe ser lo suficientemente independiente como para seguir funcionando en caso de que alguno de los demás falle, y una vez que el sistema vuelva a la normalidad reajustarse nuevamente.

Según [6] este tipo de arquitectura se clasifica dentro de dos tipos de WSN: definidas por tiempo y definidas por evento. Este tipo de clasificación organiza las WSN's en base a su dinámica de funcionamiento y se espera que una red diseñada para un tipo de WSN, funcione para todas las demás aplicaciones dentro de ese tipo de WSN.

Capítulo 5

Diseño

5.1. Red Inalámbrica

Desde hace bastante tiempo existen redes inalámbricas, así como también microcontroladores y sensores ambientales. Estos se han utilizado extensamente en áreas como meteorología y geofísica, entre otras. Pero no fue hasta los últimos años que los avances en tecnologías de comunicación lograron alcanzar al mismo tiempo prestaciones de largo alcance y bajo consumo energético.

Para poder proyectar requerimientos y definir el tipo de tecnología a utilizar se debe estudiar las condiciones del terreno y área de estudio. Llanquihue Norte, la zona donde se encuentran los humedales a estudiar, posee una longitud máxima de extremo a extremo de 2.5 km. No existen edificaciones de más de 3 pisos, pero posee abundante vegetación en algunos sectores lo que puede obstruir la señal. Hacia las afueras de la ciudad existen lugares en altura y algunas estructuras de comunicación que ofrecen la posibilidad de un gateway en altura. Existe abundante precipitación durante gran parte del año y la temperatura es generalmente baja.

En base a esta información y las características del sistema de monitoreo se definen los requerimientos necesarios para una red inalámbrica en Llanquihue:

1. Bajo costo
2. Bajo consumo energético
3. Larga distancia (4 km)
4. Tranceptores de pequeño tamaño
5. Rapidez en disponibilidad de componentes e implementación

Entonces, ¿Qué tipo de comunicación se debe utilizar?. Las opciones clásicas que hasta hace poco eran ampliamente utilizadas como Zigbee han comenzado a ser desplazadas por las nuevas especificaciones de LPWAN. Existen así variadas tecnologías que cumplen las condiciones necesarias.

Las opciones más relevantes al momento son LoRa, Sigfox, LTE-M, NB-IoT, Symphony Link y Wheightless. De éstas las más relevantes al momento son LoRa y Sigfox, esto en base a disponibilidad de componentes y un desarrollo que ya ha alcanzado suficiente uso en la comunidad cómo para tener un respaldo. Se seleccionó LoRa de manera definitiva dado que Sigfox impone barreras para la implementación de una red y no es posible establecer un gateway. Está ideado como un servicio donde Sigfox provee la red de soporte y el cliente se acopla a ella a través de módulos terminales.

LoRa entonces provee toda la flexibilidad necesaria para establecer una red propia. Existe además una gama lo suficientemente amplia de productos y soluciones a precios bastante accesibles. La comunidad que se ha generado en torno a LoRa es también amplia y es una tecnología que ha comenzado a revolucionar el desarrollo de IoT en el mundo. Esta misma etapa inicial de desarrollo es bastante interesante ya que permite generar herramientas y avances que otorguen nuevas maneras de aprovechar la tecnología, brindando el espacio para probar y desarrollar cosas.

El funcionamiento de LoRa es en configuración estrella, es decir, un gateway y nodos conectados a este gateway directamente, no hay comunicación entre nodos. De esta manera el plan inicial es situar un gateway que otorgue cobertura a todo Llanquihue Norte. De no ser posible se deberá iterar la forma de la red.

Respecto a temas de seguridad, la información transmitida será de uso público por lo que no es sensible, la intervención de la transmisión y extracción de los datos no se considera un problema. Lo que sí puede ocurrir es la generación de paquetes falsos enviados al gateway creando información errónea acerca del estado de los humedales. En primera instancia se considera esto un evento poco probable, pero de todas maneras debe tenerse en cuenta. No se abordará la solución de este problema en esta etapa pero algunas soluciones son la implementación de LoRaWAN que permita identificar nodos y generar un enlace seguro, no necesariamente encriptado.

El otro tipo de vulnerabilidad es la posibilidad de *jamming* de la señal, que en potencias bajas según la regulación no entraría en el marco de ilegal dado que LoRa funciona en bandas ISM libres, y generaría incapacidad de los nodos de comunicar la información obtenida. Nuevamente este escenario es poco probable, y aunque siempre los escenarios de seguridad son poco probables, específicamente para este proyecto no existe información relevante o un proceso que pueda significar problemas por lo que no se espera que ocurra. De todas maneras, los nodos deben considerar el almacenamiento interno de las variables medidas durante un periodo de tiempo desde que la variable fue adquirida. Esto además de generar redundancia de la información en las diferentes etapas, permite descartar el envío de información durante un ciclo y esperar el siguiente para probar si hay éxito en el envío. Esto puede continuar de manera prolongada, hasta que la memoria del procesador se acabe y comience a eliminar los datos más antiguos.

5.2. Nodo Sensor

Un nodo sensor se compone de las partes indicadas por la Figura 5.1.

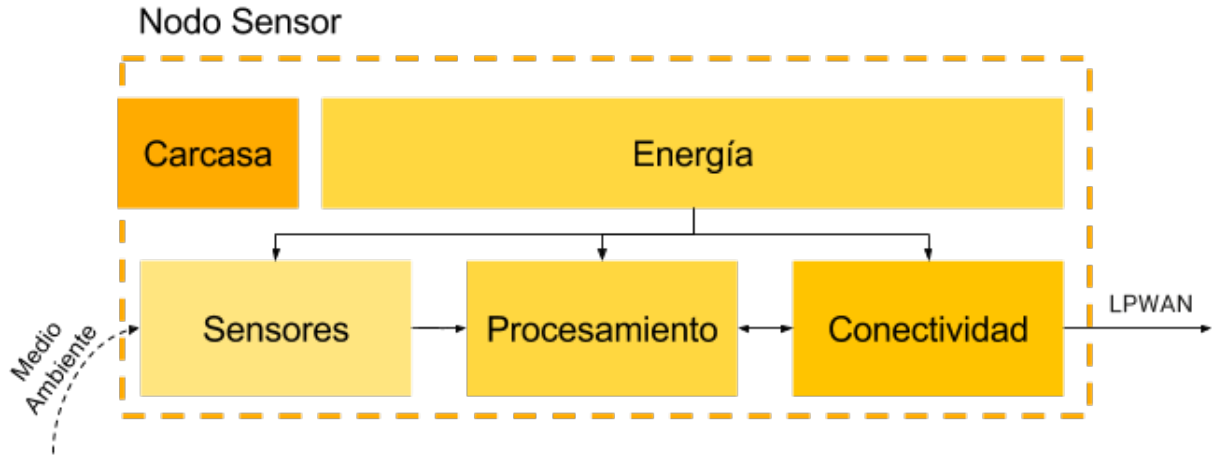


Figura 5.1: Diagrama de bloques nodo sensor

5.2.1. Empaque

Se utilizarán materiales simples, de fácil acceso y de bajo costo. La utilización de tubos de PVC aparece como una gran alternativa. Estos son diseñados para trabajar con agua y contener espacios de ella. Además, son fácilmente accesibles en mercados locales de gran parte del mundo.

5.2.2. Procesamiento

Es el cerebro del nodo sensor. Todos los demás elementos son comandados y controlados por la unidad de procesamiento. Coordina, decide, almacena, y ejecuta los procesos necesarios para que el nodo capture datos del ambiente y los envíe al gateway de manera confiable y asegurando su funcionamiento prolongado en el tiempo.

Para esto deberá implementar un ciclo de trabajo donde además de poseer una rutina de funcionamiento normal donde captura y envía datos, debe ejercer rutinas de verificación y auto chequeo de los sistemas para confirmar que todo está funcionando correctamente. Por último, mecanismos de reinicio del sistema que se gatillen ante fallas que detengan el funcionamiento del sistema también son necesarios.

Se definen requerimientos generales y técnicos base para la definición y diseño de la unidad de procesamiento:

Requerimientos Generales:

1. Bajo costo
2. Bajo consumo energético
3. Tamaño reducido
4. De fácil acceso
5. Open Source
6. Comunidad y facilidad de uso
7. Programación de bajo nivel

Requerimientos Técnicos:

1. Interfaz I2C
2. Interfaz Serial
3. Interfaz SPI
4. Mínimo 3 puertos ADC
5. Watchdog Timer
6. Alimentación 3.3V
7. Oscilador a 8MHz o menos

En este tipo de aplicaciones la unidad de procesamiento es un microcontrolador, no es necesario un procesador el cuál queda sobre capacitado para la tarea además de consumir más energía.

5.2.3. Sensores

Se definen requerimientos y aspectos importantes a fijarse para definir y diseñar los sensores:

1. Bajo costo
2. Bajo consumo energético
3. Sensores de alta calidad
4. Medición a 30 mt de distancia
5. Periodos largos de recalibración
6. Protocolos no privativos
7. Buena documentación y soporte

Los sensores definirán la calidad de las variables obtenidas. Dependiendo del objetivo se pueden utilizar diferentes tipos y calidades de sensores. En este caso dado que existe un uso científico deben utilizar sensores de alta calidad acordes a este requerimiento.

En este tipo de monitoreo los sensores se encuentran permanentemente sumergidos en el ambiente acuático de medición, esto implica una gran estrés por uso y por esta razón los sensores deben ser de alta calidad de manera de soportar las condiciones continuas de uso.

Es lógico y necesario también que los sensores estén calibrados para que la medición sea válida. La permanente exposición de los sensores al ambiente puede acelerar el proceso natural de descalibración además de agregar suciedad estos mismos. En este sentido es necesario que el período de re-calibración de los sensores sea lo más extenso posible. De la misma manera el sistema debe ser capaz de entrar en periodos de recalibración una vez en operación.

Otro aspecto importante es la disposición de los sensores y el endpoint. No siempre el lugar de medición será el mismo que el de instalación del nodo. Puede que por razones de anclaje el nodo se encuentre a un par o decenas de metros del lugar de medición y los sensores deberán llegar hasta este lugar. Esto puede influir en la transmisión de información del transductor

de los sensores a el Conversor Análogo Digital (ADC, por sus siglas en inglés) del sistema. Aspecto a tener en consideración.

5.2.4. Energía

Existen diferentes maneras de alimentar un nodo. Las más comunes son baterías o paneles solares. Se podrían desarrollar cosas como que aprovechando que es un medio acuático se utilicen corrientes de agua para generar energía para el nodo, pero el incluir partes mecánicas en el diseño complejiza todo el sistema y este tipo de desarrollos quedan propuestos.

Por el momento se descarta el uso de paneles solares, debido a que en el contexto actual el nodo debe ser pequeño y pasar lo más desapercibido posible, esto por varias razones. Primero, ya que el lugar de estudio es un lugar turístico y de uso público y la conservación del paisaje es primordial por lo que se desea intervenir lo menos posible. Y segundo, y esto abre toda una nueva área de discusión pero lo importante a rescatar es que, se desea que el conocimiento específico de los lugares de instalación de los nodos no sea de uso público. Y eso es lo que se necesita saber para la definición de esta sección.

Bajo el escenario de utilización de una batería la selección de esta se realiza con base en 3 variables: capacidad, dimensiones y costo.

La capacidad de la batería definirá la autonomía del nodo, se espera que este sea capaz de funcionar durante 3 meses continuados sin la necesidad de recargar/cambiar la batería. Se debe hallar un termino medio entre las tres variables.

Se proyectará para ser capaces de proyectar bien estas propiedades es necesario un estudio posterior del diseño e implementación del sistema. El tamaño de la batería dependerá del tamaño del empaque y los demás componentes, sensores procesadores, etc. Asimismo los costos, la cantidad de dinero disponible a invertir en baterías vendrá dada por los costos de los demás elementos. Y la capacidad de la batería estará sujeta también a los consumos de los elementos seleccionados e integrados en el sistema, esto definirá un mínimo ideal de capacidad que deberá entonces ser balanceado con las variables ya vistas.

5.3. Gateway

El gateway al igual que el nodo también se puede estructurar en una división de partes indicada en la Figura 5.2.

La comunicación viene dada por la red inalámbrica ya diseñada. El procesador a utilizar en esta ocasión será uno de mayores capacidades, este debe ser capaz de manejar múltiples paquetes entrantes, almacenar información de manera local, poseer conectividad a Internet para subir información a un servidor y ser capaz de tomar decisiones.

La Raspberry Pi se alimentará directamente de una instalación eléctrica. De todas mane-

Capítulo 6

Implementación

Se describe la implementación de los elementos que componen el prototipo del sistema de monitoreo.

El desarrollo presentado de la carcasa no será implementado en esta etapa y alcanza un nivel de análisis y propuesta de los materiales a utilizar.

6.1. Selección y Descripción de Componentes

6.1.1. Nodo

6.1.1.1. Carcasa

Una de las consideraciones de diseño más importantes para el nodo sensor es la limitación en tamaño que estos tienen. Un tamaño reducido facilita la instalación y desempeño en terreno de los nodos, genera una interacción de menor impacto con el medio ambiente e interviene menos el mismo. En este caso se definen dimensiones máximas de ancho, largo y alto de $8\text{ cm} \times 8\text{ cm} \times 15\text{ cm}$, determinando un capacidad de 0.00096 m^3 , aproximadamente 1L de volumen.

El tamaño del nodo también impone una limitante al tamaño de la batería, y mientras menor tamaño de la batería menos energía disponible. Existe así un tradeoff que hay que hacer entre tamaño del nodo y batería que debe ser tomado en cuenta y en el que los demás componentes del nodo influyen de manera importante, tanto por su tamaño cómo por su consumo.

Como carcasa se propone utilizar un tubo de PVC para canalización sanitaria de 75mm con tapas por ambos lados como en la Figura 6.1a. El uso de PVC provee múltiples ventajas: es ampliamente utilizado en sistemas de protección de agua, es barato, es liviano, es fácil de trabajar, su uso se extiende a todo el mundo por lo que es muy fácil de encontrar.

Las condiciones en un sistema de monitoreo del medio ambiente pueden ser muy duras, especialmente en Llanquihue, estos deben soportar ambientes de lluvia extrema y fuerte viento. Si bien este equipo no se encontrará en directo contacto con estas condiciones y no estará pensado para su versión final en esta etapa, el uso de la estructura propuesta es capaz de ser diseñada para soportar estas situaciones. Existen otros ejemplos de uso de este método [28] con resultados exitosos. Se puede observar en la Figura 6.1b, este proyecto de dataloggers acuáticos ha desarrollado un sistema de protección para su equipo con base en el uso de canalizaciones de PVC con resultados que les permite realizar mediciones durante semanas de manera sumergida.



(a) Ejemplo de tubo PVC para prototipo actual (b) Ejemplo de proyecto de carcasas para dataloggers acuáticos

Figura 6.1: Ejemplo de carcasas de PVC

6.1.1.2. Procesamiento

Para el procesamiento se utilizará un Arduino Pro Mini 3.3V/8MHz. Este emplea un microcontrolador ATmega328 que cuenta con 32KB de memoria de programa, 2KB para variables y 1KB de memoria no volátil. Además posee 20 puertos externos, 6 con ADC de 10 bit y dos con interrupciones externas, un puerto UART de comunicación serial, puertos SPI e I2C. Todo esto en una placa como la de la Figura 6.2 de 1.8x3.3cm.

Este entorno es muy popular en todo el mundo lo que lo hace de fácil acceso en casi cualquier lugar. El amplio conocimiento de la plataforma ayuda a un desarrollo más fluido y rápido del sistema, además que otorgar continuidad de manera más fácil. Esto mismo lo hace escalable y replicable en diferentes lugares disminuyendo los tiempo de aprendizaje del sistema.

Si bien por defecto un Arduino no es un elemento de bajo consumo, ni en hardware ni en software, es posible reducir su consumo a través de modificaciones a menos de un décimo del original alcanzando realmente buenos desempeños. El utilizar un oscilador de 8MHz ayuda a disminuir aún más su consumo. Además, el chip ATmega328 se puede programar a muy bajo nivel permitiendo aumentar su eficiencia y posee modos de espera de bajo consumo.

6.1.1.3. Comunicación

El módulo de comunicación utilizado fue el módulo LoRa inAir9 que utiliza el transceptor SX1276 fabricado y diseñado por Semtech. Este módulo ha presentado muy buenos desempeños en trabajos similares [29]. Se integran los componentes en una PCB de 4 capas implementada con electrónica de alta calidad y componentes de precisión, lo que garantiza una alta sensibilidad y consumo eficiente. Parte de este diseño de alta calidad incluye el uso de protección metálica observable en la Figura 6.3 para aislación de ruido electromagnético externo.

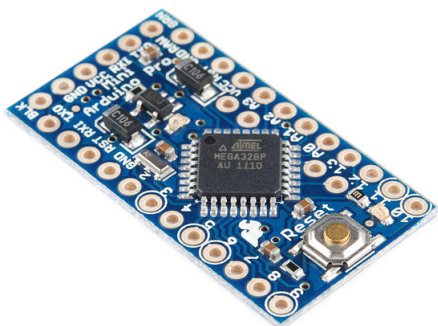


Figura 6.2: Arduino Pro Mini

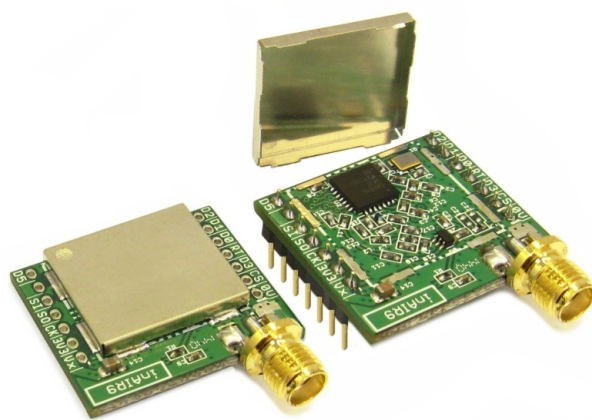


Figura 6.3: Módulo LoRa inAir9

Al elegir módulos LoRa es importante asegurarse que la banda de comunicación utilizada coincide con la regulación legal local. Esta versión de la línea inAirMod de Modtronix es configurable para transmitir en la banda de los 868 y 915MHz, la última es la que se ajusta a la regulación chilena. Por otro lado posee una potencia de transmisión máxima de +14dBm (25mW), la cual de no ser suficiente, existe una versión de mayor potencia, el inAir9B, que alcanza los +20dBm (100mW), pero a su vez consume más corriente.

Alimentándose de entre 1.8V a 3.7 V, el consumo del inAir9 alcanza en los 29mA en transmisión a +13dBm de potencia, y los 87mA a +17dBm. En recepción, alcanza los 10mA de consumo y posee modo de espera para bajo consumo. Entre otras cosas, esta radio LoRa implementa diferentes tipos de modulación: FSK, GFSK, MSK, GMSK y OOK. Además posee un link budget máximo de 168dB y una sensibilidad de hasta -148dBm.

Existen otras opciones populares como el RFM6x y el RFM95 pero el inAir9 es más económico, posee un menor consumo y una mejor calidad de construcción lo que se traduce en un enlace mejor y más confiable. Su fabricante se encuentra en Australia pero existen múltiples proveedores en países como Estados Unidos y de Europa, muy probablemente llegue su distribución a Chile dentro del siguiente año 2018.

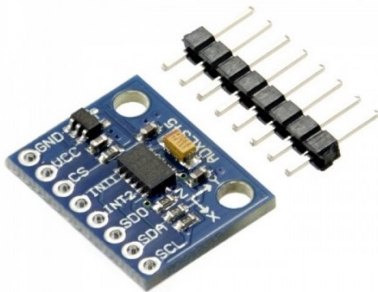
6.1.1.4. Sensores

Para los sensores de las condiciones internas del nodo de aceleración (movimiento), temperatura y humedad, se utilizarán el acelerómetro ADXL345 y el sensor de temperatura y humedad DHT22 expuestos en la Figura 6.4. Ambos son sensores digitales, lo que simplifica el trabajo del microcontrolador, de bajo costo, y poseen un bajo consumo de corriente.

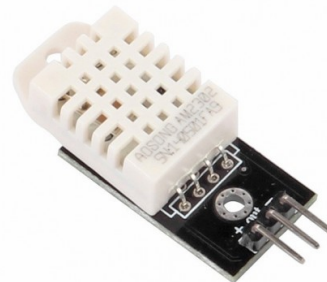
El ADXL345 es un acelerómetro digital MEMS de 3 ejes siendo capaz detectar movimiento en todas las direcciones. Se comunica mediante protocolo SPI e I2C, a elección, y posee una resolución de 13 bit en un rango configurable de máximo $\pm 16g$. Su consumo en operación normal es de entre $30\mu A$ y $140\mu A$ dependiendo de la tasa de muestreo, y de $0,1\mu A$ en modo de espera.

Es capaz de detectar movimientos y notificar ante eventos mediante un pin de interrupción. Esta es la característica más relevante. Además, permite implementar un módulo de aviso ante movimiento sin necesidad de estar leyendo los valores de aceleración constantemente desde el microcontrolador.

El DHT22 es un sensor simple sin grandes prestaciones, es lento, pero cumple su objetivo. Mediante protocolo I2C comunica sus valores dentro de rangos de medición de humedad de 0 - 100 % con una precisión de 2 - 5 % y un rango de temperaturas de $-40 - 80^{\circ}C$ con una precisión de $\pm 0,5^{\circ}C$.



(a) ADXL345: Acelerómetro



(b) DHT22: Sensor de temperatura y humedad

Figura 6.4: Sensores de medición de variables internas del nodo

Para la selección de los sensores de calidad del agua hubo requerimientos claves que definieron la elección:

- Plazo de 6 meses de implementación.
- Capacidad de medir a más de 10 metros de distancia del nodo.
- Tiempos de recalibración mayor a 3 meses.
- Tiempos de recambio de sensores de mínimo 2 años.
- Sumergimiento completo por tiempo indefinido.

Un plazo de 6 meses para la implementación de todo el sistema no concede mucho tiempo para desarrollo de sensores y tampoco para la adaptación de alguno existente que no cumpliera algún requerimiento para que sí lo haga, tomando que cuenta entonces que además se debiesen realizar pruebas para asegurar que se comportarán bien durante tiempo prolongado. Considerando esto, los sensores deben llegar listos para ajustarse a los demás requerimientos y realizar pruebas.

Se seleccionaron los sensores de Atlas Scientific para las variables abióticas de pH, Temperatura, Oxígeno Disuelto y Electroconductividad. Sensores de iones nitratos y fosfatos resultaron muy costosos y fueron descartados para esta etapa del proyecto.

Los sensores Atlas Scientific poseen una documentación extensa, detallada y un servicio de soporte siempre activo. Estos son sensores de alta calidad, solo uno requiere mantenimiento y poseen tiempos de recambio y calibración del orden de años. Utilizan interfaces Serial e I2C para comunicarse. Si bien durante la medición consumen bastante energía después entran a un modo de espera de bajo consumo.

Se componen de una probeta, un circuito lector, y un aislador de voltaje para el caso de utilización simultanea con sensor de Electroconductividad.



Figura 6.5: Elementos para medición de pH con sensor Atlas Scientific

6.1.1.5. Energía

Se realiza una primera estimación del consumo de los elementos la cual da indicios iniciales de las características de la batería a utilizar. Además se comienza a estudiar el ciclo de trabajo que poseerá el nodo y su impacto en el consumo.

Ya habiendo seleccionado los componentes se procede a armar la tabla expuesta en la Figura 6.6.

Arduino Pro Mini	ON (mA)	sleep (mA)		Ciclos (seg) ->	570	5	5	5	5	10	Total ON (seg)
inAir9	3.21	0.054	->	on/off [1/0]	0	1	1	1	1	1	30
sensor O.D.	22	0.015	->	on/off	0	0	0	0	0	1	10
sensor pH	12.1	0.3	->	on/off	0	1	0	0	0	0	5
sensor E.C	13.9	1	->	on/off	0	0	1	0	0	0	5
sensor T°	35	0.4	->	on/off	0	0	0	1	0	0	5
accel.	14.3	0.09	->	on/off	0	0	0	0	1	0	5
dht22	0.14	0.14	->	on/off	1	1	1	1	1	1	600
	2.5	0	->	on/off	0	0	0	0	1	0	5
											600
					v	v	v	v	v	v	
Consumo					2.00	16.96	18.06	39.76	21.87	27.14	
% tiempo					0.95	0.01	0.01	0.01	0.01	0.02	

Batería (mAh)	Consumo prom.	Duración (hrs)	(días)	(meses)
6000	3.157	1901	79.19830	2.6

Figura 6.6: Estimación inicial de consumo para un nodo de la red de monitoreo.

Se define un ciclo de total 10 minutos, cada 9 mins. y 30 segundos, el nodo se despierta y mide cada una de las variables, posteriormente las almacena, luego las envía al gateway. Esto tomará un tiempo de 30 segundos luego de lo cual vuelve a entrar en modo de espera de bajo consumo.

Cada elemento posee consumos en estado de funcionamiento normal y estado de bajo consumo o *sleep*. Se obtienen los consumos estimados en base a datasheets, en caso de no existir se acude a reportes de uso desarrollados por usuarios en internet.

Con base en esta información se calcula el consumo promedio del sistema por ciclo. Luego en base a este consumo promedio se calcula el tiempo total de duración del nodo con este consumo medio. La estimación de consumo a través de esta metodología es válida para tiempo de estimación mucho mayores al tiempo de un ciclo lo cuál será válido para cualquier aplicación que aspire a funcionar durante más de 10 horas seguidas.

Al simular que el sistema funciona sobre una batería de 6000 mAh se obtiene un tiempo de duración del sistema de 2.6 meses.

Para esta estimación se asumió una periodicidad de toma de muestras de 10 minutos, esta variable también influye en el tiempo final de vida energética del sistema. Se calculan los tiempos de vida del sistema en base a diferentes periodos de muestreo para ver cómo influye el ciclo de trabajo en la autonomía de un nodo.

En la Figura 6.7 se observa un comportamiento asintótico. Para sistemas con alta frecuencia de muestreo el realizar cada 5 minutos más la medición aumenta considerablemente el tiempo de vida del sistema. A medida que se aumenta el periodo entre mediciones el aporte es cada vez menor pero aún considerable hasta los 40 minutos de este gráfico. En este punto los aportes de aumentar el periodo de muestreo aumentan cada vez menos, lo cual define en la práctica una cota máxima de autonomía del sistema posible de alcanzar en base a modificaciones al ciclo de trabajo.

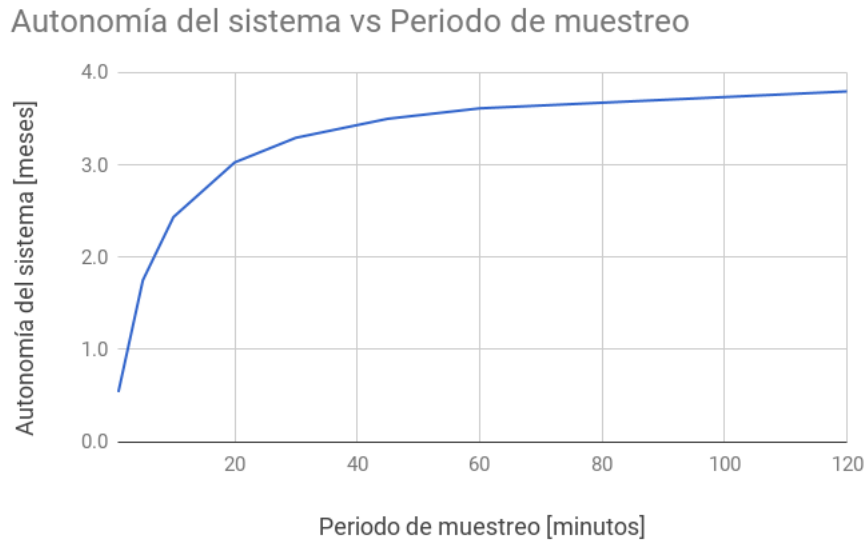


Figura 6.7: Autonomía energética del sistema vs Periodo de muestreo de variables

6.1.2. Gateway

Se utilizará como módulo central de procesamiento una Raspberry Pi. Este pequeño computador posee un procesador Quad Cortex de 1.2GHz, 1GB de memoria RAM, capacidad de hasta 32GB mediante tarjeta Micro SD externa, puerto Ethernet 10/100, Bluetooth y WiFi incorporado. Funciona mediante Raspbian, un sistema operativo Linux especial para la placa y se alimenta mediante un puerto micro usb de 5V y 2.5A.

Existe una versión más pequeña de la placa llamada Raspberry Pi Zero, más pequeña, con casi las mismas prestaciones y con la ventaja de consumir menor potencia. Por estas razones se recomienda la utilización la Raspberry Pi Zero, mas ambas son compatibles entre si por lo que la utilización de una o la otra dependerá exclusivamente de su disponibilidad.

La administración la alimentación de los equipos y la utilización de una batería en caso de corte de energía la realizará la placa Lipo Rider Pro. Capaz de administrar alimentación externa, una batería y un panel solar para brindar una fuente de energía confiable a dispositivos electrónicos.

El módulo de comunicación utilizado es el mismo inAir9 descrito para el nodo en la sección anterior. La batería será una Ion Litio de 6000mAh similar a la del nodo sensor. El consumo de una Raspberry Pi Zero junto con el módulo LoRa consumirán un aproximado de 150mA, esto da una autonomía de 40 horas, que otorgan casi 2 días de funcionamiento continuado en modo normal de operación al sistema en caso de emergencia.

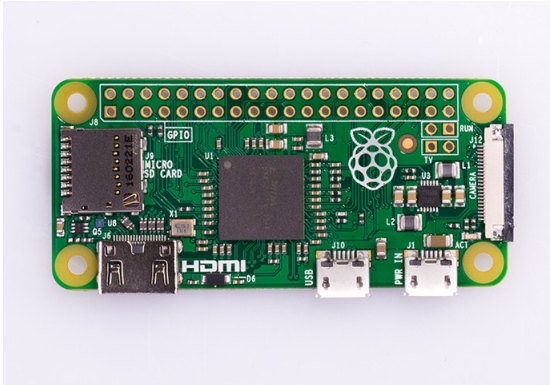


Figura 6.8: Raspberry Pi Zero



Figura 6.9: Lipo Rider Pro: Placa de administración de energía.

6.2. Comentarios acerca de la Energía

La energía puede ser una de las limitantes más importantes a tener en cuenta. Esta definirá el tiempo de operación del sistema sin necesidad de intervención lo cuál en muchos casos es necesario que sea lo más grande posible dada la imposibilidad de recargar baterías debido al difícil acceso y/o a la cantidad de nodos en el sistema.

En este caso cómo se reviso en la Figura 6.6, en base a consumos normales de los componentes se obtiene un tiempo de vida aceptable para este proyecto por lo que se enfoca el trabajo en otras áreas. De todas maneras existen unas serie de comentarios a realizar.

La selección de componentes puede tener un impacto grande en el consumo energético. Se debe pensar en *hardware* de bajo consumo. Elementos cómo un Arduino Pro Mini no puede ser usados de manera directa en una implementación final, pero es posible *hackear* el hardware del Arduino para disminuir su consumo de manera considerable. Además existen versiones abiertas de bajo consumo de componente.

La programación también es un aspecto que puede influir fuertemente en el consumo energético del sistema. Se debe pensar en *software* de bajo consumo. Un producto final debe ser programado en lenguaje de bajo nivel que utilice los recursos del microcontrolador y e interactúe con los demás componentes de manera eficiente.

Estas consideraciones en energía son esenciales para el desarrollo de un sistema de este tipo, y se debe evaluar su importancia en base al sistema que se desea implementar y sus requerimientos.

6.3. Ensamble

6.3.1. Nodo

6.3.1.1. Conexiones

La conexiones del nodo se pueden observar en la Figura 6.10. Notar que en la figura se despliegan también un botón y un sensor de luz, estos elementos no son utilizados en esta etapa sino que en la de medición de cobertura descrita en el Capítulo 7 y en la siguiente subsección de programación del nodo de monitoreo.

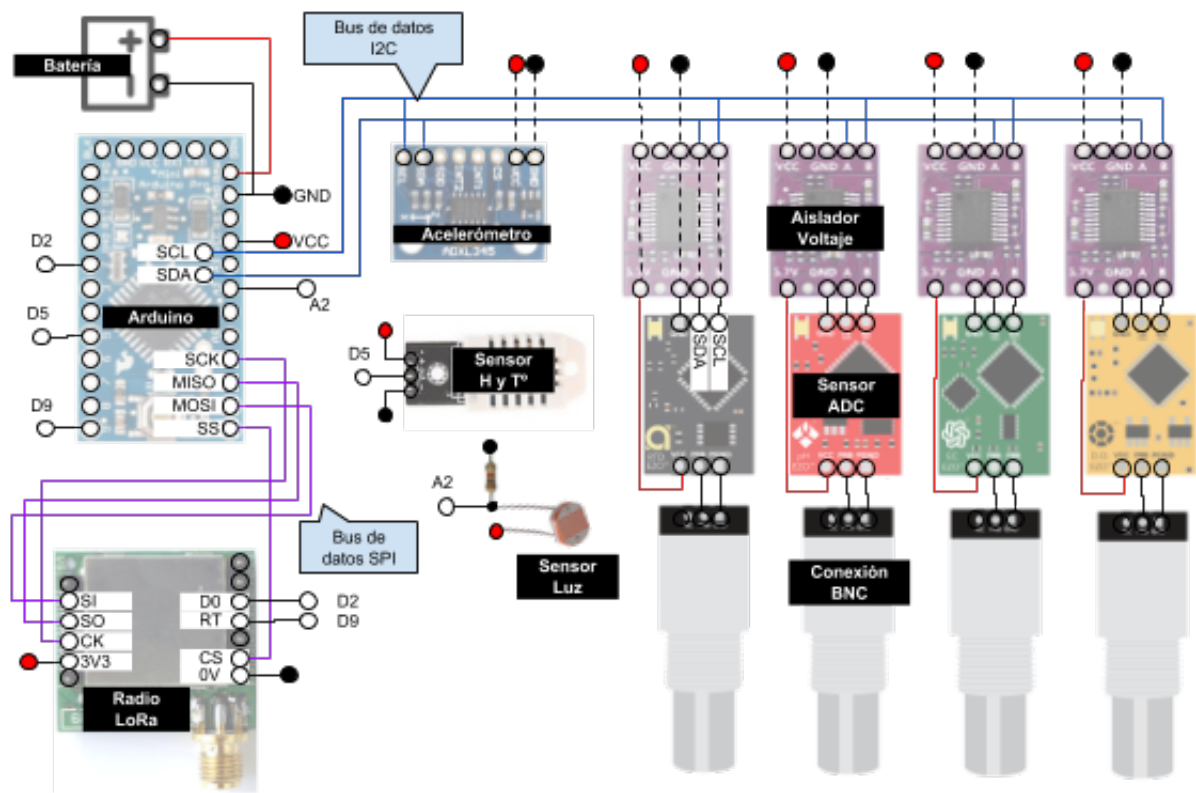


Figura 6.10: Conexiones componentes nodo

De estas es importante destacar las conexiones a realizar con los sensores y sus placas. Estas deben ser realizadas con cables de protoboard cortados a la medidas de las conexiones, sin utilizar cables jumper típicos de arduino. Esto debido a que los sensores son muy sensibles en su alimentación especialmente cuando son alimentados con 3.3V.

6.3.1.2. Problema 1 - Caída de Voltaje

En la teoría y con base en todos los datasheets y documentación, las conexiones de la Figura 6.10 debería funcionar, pero esta configuración entrega problemas con los sensores de

Atlas Scientific.

El problema consiste en que al utilizar los aisladores de voltaje, necesarios para mediciones sin ruido, cada uno genera una pequeña caída de voltaje en la entrada, de manera que al acumular 4 aisladores hace que los 3.3V de alimentación disminuyan a 2.7V lo que es muy bajo y hace que los sistemas dejen de funcionar.

Este problema no se encontraba documentado de manera oficial y no se encontró mención de este en foros técnicos, su comportamiento queda descrito en la Figura 6.11. Al conectar los sensores directamente también ocurre este efecto pero con menor intensidad como se observa también en la figura por lo que no interviene de manera negativa el funcionamiento del sistema.

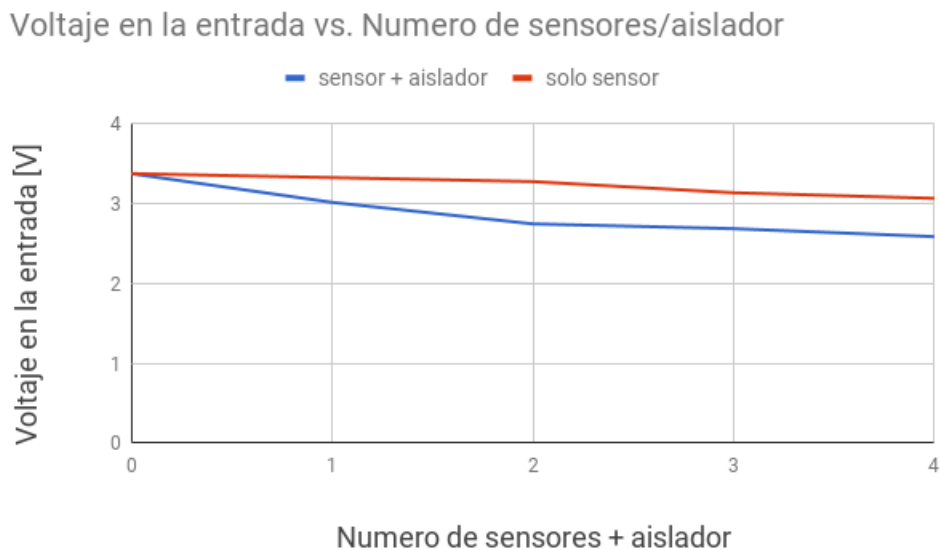


Figura 6.11: Caída de voltaje de alimentación efecto de los sensores externos

Cabe destacar que al utilizar un Arduino Uno u algún otro microcontrolador de 5V el problema parece desaparecer, en realidad los efectos no se manifiestan de manera tan agresiva, y es posible hacer funcionar el sistema. Esto abre la posibilidad de utilizar un step-up de voltaje de 3.3V a 5V pero se descarta dado el aumento en el consumo que esto implica.

Una propuesta de solución al problema se desarrolla en la sección 6.3.1.3.

6.3.1.3. Problema 2 - Consumo de aisladores

Otro problema no documentado es el del consumo de los aisladores. Estos poseen un consumo en operación de aproximadamente $20mA$ el cuál no disminuye al poner los sensores en modo *sleep* de baja energía. Esto se presenta como un gran problema ya que añade una cantidad de consumo fijo muy grande, disminuyendo el tiempo de operación de un nodo de monitoreo a solo unos cuantos días.

La utilización del aislador es necesaria dado que el sensor de electroconductividad al utilizar la generación de una pequeña corriente para la medición, introduce ruido en las demás medidas si estas no están debidamente aisladas. En vista de resolver los problemas expuestos se proponen tres opciones.

1. Utilizar un solo aislador de voltaje para todos los sensores, se debe investigar la factibilidad de esta solución. Un problema de esta solución es que si bien disminuye el consumo neto de los aisladores, persiste un consumo constante de $20mA$.
2. Utilizar transistores mosfets de bajo consumo para activación y desactivación de los sensores en el momento en que se utilizan. Se consultó con Atlas Scientific y no es necesario que los sensores estén constantemente encendidos para entregar una buena medición. Un inconveniente sería que se utilizan muchos componentes lo que hace el sistema más complejo y grande en tamaño.
3. No utilizar aisladores de voltaje. Dado que se tomarán medidas de manera consecutiva y cada una separada de la otra, los efectos del sensor de electroconductividad no debería influir en las demás tomas de datos. Esto se implementa con transistores mosfets de bajo consumo. Inconvenientes, puede no ser posible no utilizar los aisladores de voltaje para obtener una buena medición.

El desarrollo de estas soluciones escapa los alcances de este trabajo.

6.3.1.4. Prototipo

Para proseguir el trabajo, se procede a realizar un prototipo sin aisladores y sin sensor de E.C. para este prototipo, el cual se ilustra en la Figura 6.12. Se utiliza temporalmente una batería lipo de 2000mAh. En la imagen faltan las probetas de los sensores las cuales se acoplan a los conectores BNC de color blanco en la parte inferior de la figura.

6.3.2. Gateway

Se diagraman las conexiones necesarias para el gateway en la Figura 6.13. Luego de probar múltiples opciones para conectar el inAir9 a la Raspberry Pi, todas fueron descartadas y se procedió a utilizar un arduino como interfaz entre la Raspberry Pi y el módulo LoRa. De esta manera el modulo LoRa se comunica con el Arduino a través de la interfaz SPI, y el arduino cada vez que recibe un paquete envía el mensaje por comunicación serial a la Raspberry Pi.

Si bien no fue diseñado de esta manera, este tipo de arquitectura facilita el prototipado y las pruebas con la tecnologías LoRa ya que se ha generado un nuevo módulo LoRa con interfaz serial e instrucciones mediante las cuales se accede a configuración de los parámetros de canal, la potencia y el envío de mensajes. Esto posibilita el tomar este nuevo módulo y conectarlo directamente a la computadora para realizar pruebas con la conexión LoRa desde el escritorio, luego desconectar y volver a conectar este módulo a la Raspberry para realizar pruebas en terreno. El desarrollo de estas pruebas se describe en la siguiente sección.

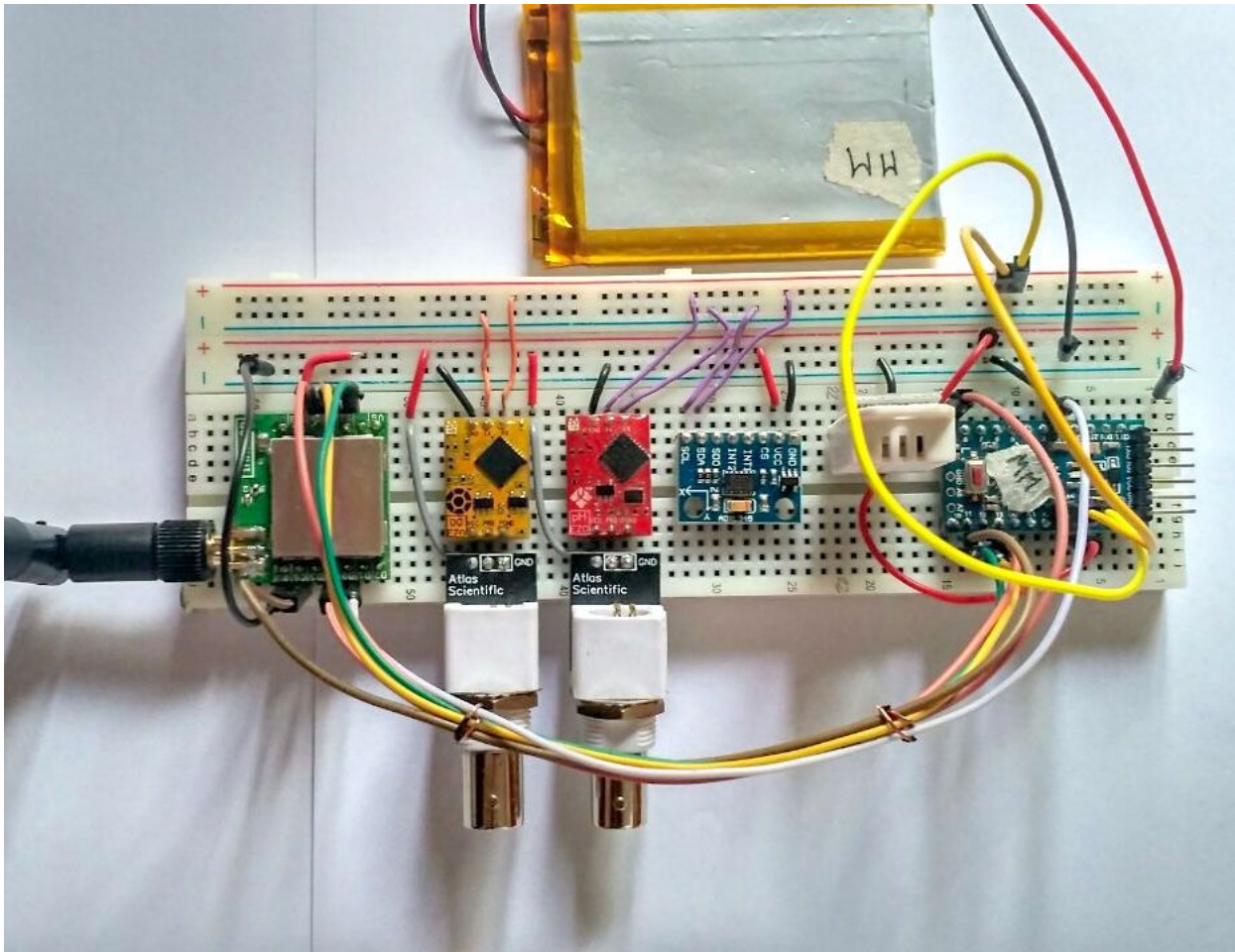


Figura 6.12: Prototipo de nodo sensor

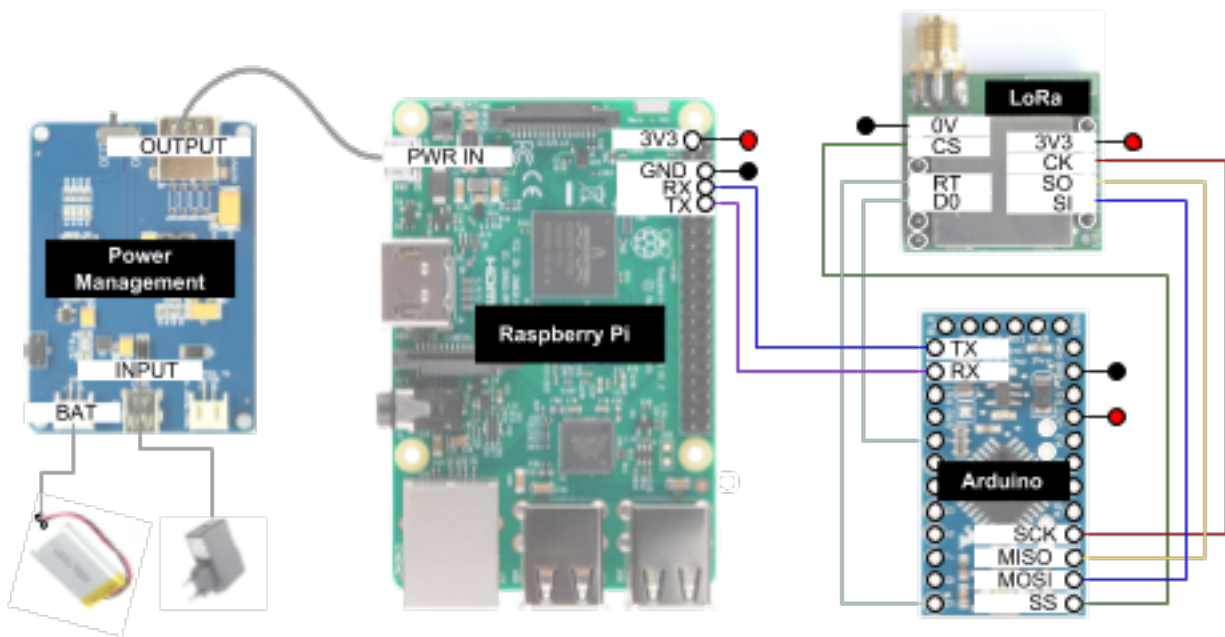


Figura 6.13: Diagrama de conexiones gateway

6.4. Programación

6.4.1. Nodo

La programación del nodo se realiza en lenguaje C utilizando el IDE y la plataforma Atom y PlatformIO, ambas se integran en un entorno de desarrollo para sistemas embebidos de IoT. Así, este IDE permite programar microcontroladores Atmega328 (Arduino Pro Mini) en lenguaje C y en un entorno de mayor nivel y más funcionalidades que el IDE de Arduino.

La función del programa es: adquirir datos, enviarlos, entrar en estado *sleep* de bajo consumo durante un tiempo X, y volver a iniciar el ciclo en base a la rutina que se diagrama en la Figura 6.14. Dado que también se utilizará el nodo cómo módulo de prueba de cobertura (más detalles en el Capítulo 7), se pueden enviar instrucciones de inicio de comenzar transmisión y detener transmisión. Es posible encontrar el código en la sección de Anexos A.2

Se incorporan en la programación elementos utilizados en las pruebas de cobertura posteriores a realizar detallada en el Capítulo 7. Una de estas funciones extra es la posibilidad de simular interferencia en la transmisión mediante la presión de un botón. Esto sigue ejecutando todas las rutinas de adquisición de datos y armado del paquete solo que no envía el paquete.

6.4.2. Gateway

Con base en la configuración del gateway siendo compuesto de una Raspberry Pi y un Arduino. El Arduino en conjunto con la radio LoRa generan un nuevo dispositivo LoRa de más alto nivel capaz de recibir un set de instrucciones que permiten enviar datos bajo diferentes configuraciones de parámetros y otorgar capacidad LoRa a cualquier dispositivo con interfaz serial.

Los comandos implementados se detallan en la Tabla 6.1 a continuación.

Command	Function	Params	Syntax (always end's with '+')
at	check if ok	-	at+
begin	start listening	-	begin+
stop	stop receiving	-	stop+
send	send lora packet	Message	send+decir adios es crecer+
config-mod	configure lora mod	Mod	config-mod+5+
config-params	configure lora params	BW, CR, SF	config-params+125+4/5+7+
sleep	enter sleep mode	-	sleep+
config-tx-power	configure tx power	Tx Power	config-tx-power+13+

Tabla 6.1: Comandos para Arduino más SX1276

La función principal del módulo LoRa-Arduino es de escuchar paquetes LoRa entrantes,

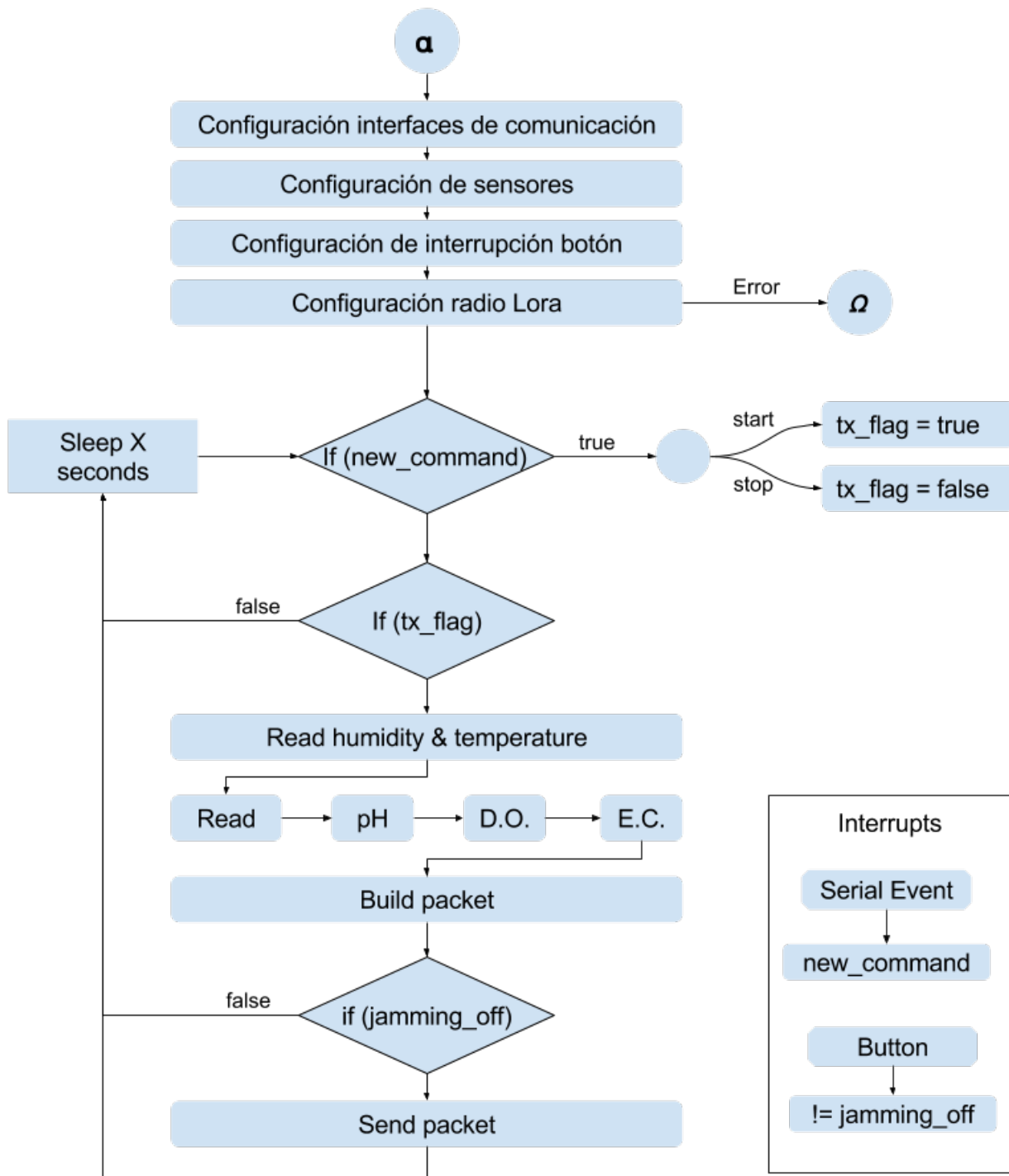


Figura 6.14: Diagrama de flujo programación nodo

tomar el RSSI del paquete recibido y enviarlo al gateway. Se diagrama el flujo en la Figura 6.15. El código se puede encontrar en la sección de Anexos A.1

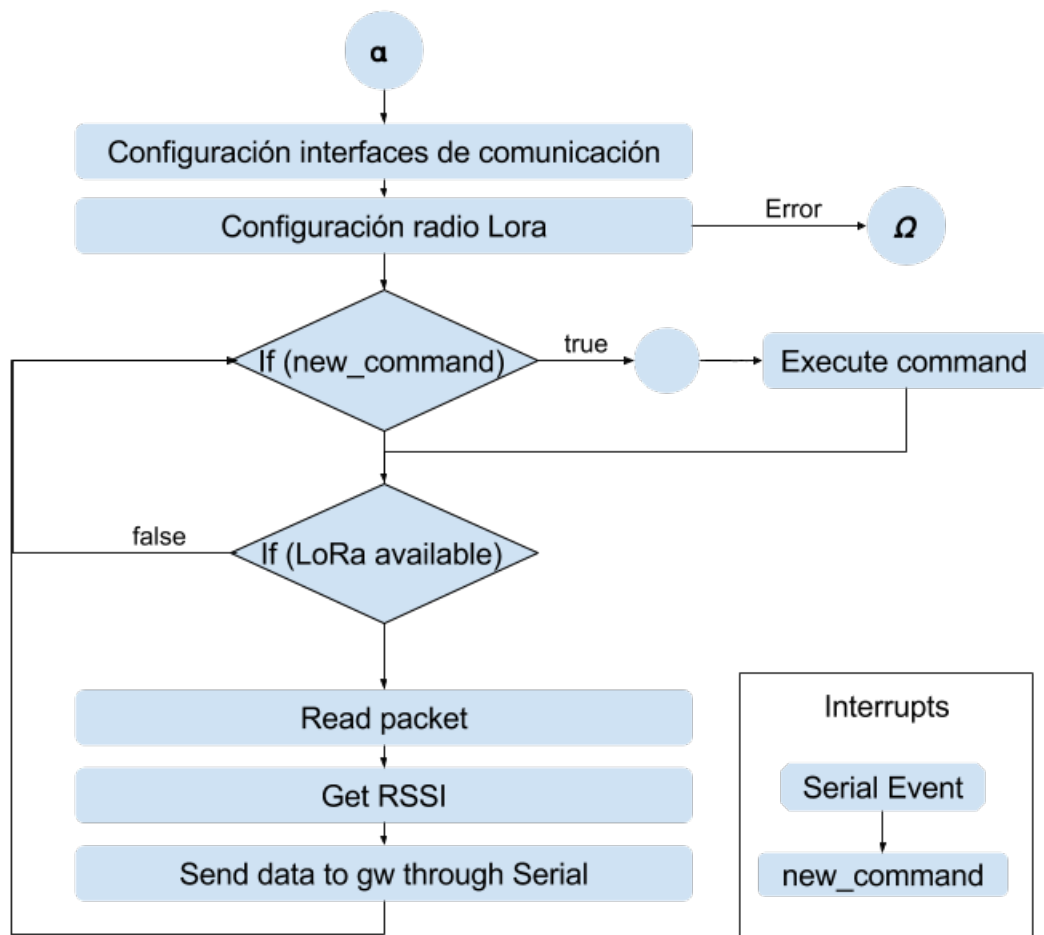


Figura 6.15: Diagrama de flujo programación Arduino gateway

Capítulo 7

Sistema para Estudio de Cobertura

El poner en marcha un sistema de monitoreo remoto de variables en algún sistema, implica que se deberá generar una red de comunicación sobre la cuál los nodos puedan transmitir los datos. Las características y viabilidad de esta red pueden verse fuertemente afectadas por la extensión y el tipo de lugar donde esta se levante. Es por esto que un estudio previo debe hacerse sobre el terreno para (1) saber si es posible levantar un sistema de comunicación que entregue conectividad a todos los nodos, y (2) definir la arquitectura y configuración que deberá tener la red para poder funcionar de manera lo suficientemente confiable.

El estudio de factibilidad de una red se puede realizar a través de mediciones de cobertura en terreno previas a la instalación. A través de pruebas que midan la pérdida de paquetes e intensidad de la señal en diferentes puntos, se puede mapear el terreno y generar un visualización del desempeño de la tecnología de red en el terreno. Esto podrá guiar la forma de la red y la disposición de los nodos buscando que queden en lugares que garanticen buena conexión.

Es necesario entonces contar con un sistema para la medición de cobertura en terreno. Existen actualmente soluciones en el mercado para esta tarea, Modtronix ofrece un Kit de Desarrollo Semtech/Lora basado en el mismo transceptor utilizado en este trabajo el SX1276. Este kit permite realizar medición de cobertura en terreno pero posee bastantes limitantes cómo almacenamiento de los datos, conocer la localización geográfica, la pantalla es bastante pequeña y el menú puede ser lento de utilizar.

Por estas razones se plantea desarrollar un sistema de equipos para la medición de cobertura LPWAN en terreno que compense las falencias de las soluciones actuales. Por lo demás la estructura de los dispositivos de este sistema será bastante similar a la de un nodo sensor de la red de monitoreo, sin contar con los sensores externos.

El sistema deberá contar con características de fácil portabilidad, bajo costo, rápido despliegue, rápido almacenamiento, fácil utilización, utilización bajo condiciones adversas y almacenamiento automático de variables. En definitiva la idea es poder encender los equipos y recorrer el área de interés sin tener que estar preocupado de anotar todo de manera manual y ser un experto en LoRa o telecomunicaciones.

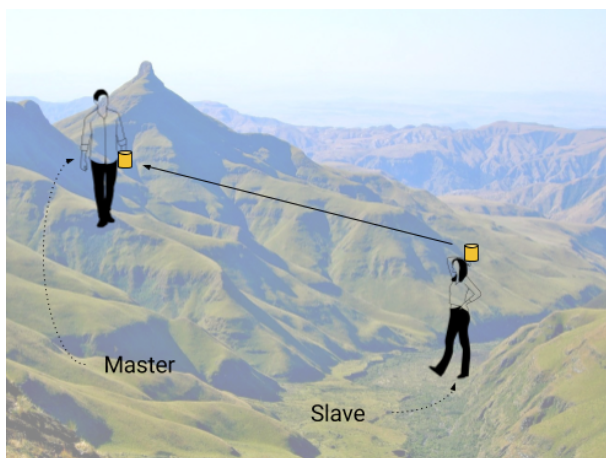
7.1. Metodología

Debe existir previo a la exploración del territorio o ambiente, un estudio de las condiciones geográficas de la zona, un mapa de relieve, cosas como zonas de altura o vegetación voluminosa. Si es una ciudad, tomar en cuenta obstáculos, edificios, fuentes de interferencia, etc. Todos estos aspectos influyen en el enlace y permiten así generar un plan previo de puntos de interés e hipótesis sobre las cuales basar el recorrido de medición de cobertura.

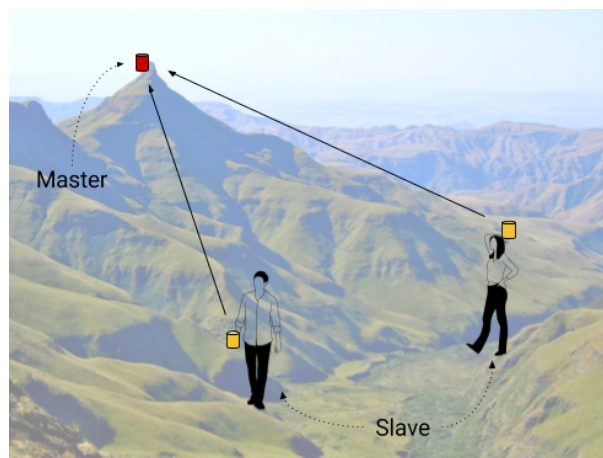
Se recomienda siempre favorecer lugares en altura, a pesar que exista un lugar más cercano a un mismo nivel, esta misma distancia en altura genera un mayor impacto en el enlace. Esto con base en pruebas reales de cobertura que se detallan más adelante.

Existirán dos modos de uso del sistema como se indica en la Figura 7.1. El primero consiste en un despliegue más rápido, dos personas cada una con su equipo de distribuye en la zona de interés de manera ágil y rápida. Debe existir una coordinación de ambos durante la toma de muestras. Este tipo de exploración puede ser utilizada en zonas remotas y de difícil acceso. También como primeras mediciones y familiarización con el terreno. Si no se posee mucho tiempo y no existieron visitas previas también es una buena posibilidad.

Para el segundo modo de uso del sistema de exploración de cobertura, debe existir un gateway previamente instalado en un lugar que se considere potencial punto fijo de instalación de un gateway. No es necesario, pero es de mucha ayuda si el gateway además posee acceso a Internet durante las pruebas. Esto porque como se verá más adelante, el gateway actualiza constantemente a la persona del estado actual de la prueba permitiendo a esta tomar decisiones en tiempo real acerca del enlace y el curso a tomar. Esta pruebas pueden ser realizadas de manera individual. Y si las mismas dos personas trabajan mediante este modo de uso, realizarán el doble de trabajo en el mismo tiempo, siempre que hayan planificado correctamente su trabajo.



(a) Primer modo de uso



(b) Segundo modo de uso

Figura 7.1: Modos de uso del sistema para estudio de cobertura.

7.2. Diseño

El sistema se compondrá de dos tipos de dispositivo: un maestro y un esclavo. En la práctica puede haber más de un esclavo pero, por el momento, siempre habrá un solo maestro. Por conveniencia cada equipo debe poder tomar roles tanto de maestro como de esclavo siendo perfiles configurables en el momento de iniciar la prueba. Los equipos deberán poder:

- Enviar y recibir paquetes
- Configurar los parámetros de canal (ancho de banda, coding rate y spreading factor)
- Conocer su localización geográfica
- Guardar información de paquetes enviados y recibidos
- Desplegar una interfaz gráfica que permita (1) ver que paquetes son enviados o recibidos y (2) realizar configuraciones en los parámetros de comunicación.

Se decide combinar el nodo de transmisión LoRa con la capacidades de un celular móvil. Ya prácticamente todos los celulares poseen GPS e internet móvil. A través de una aplicación móvil se puede acceder a estas capacidades pudiendo además utilizar el almacenamiento del celular entre otras cosas. Será necesario poder comunicar el nodo con la aplicación.

7.2.1. Comunicación: LoRa

El enlace de comunicación LoRa puede ser configurado para desempeñarse de distinta manera mediante tres parámetros. Así puede elegir llegar más lejos pero en un mayor tiempo de transmisión, o gastar menos energía pero llegar más cerca, etc.

Estos parámetros son 3: Ancho de Banda (BW), Spreading Factor (SF) y Coding Rate (CR). Entre ellos se generan 72 posibles configuraciones de canal, y es impráctico el tratar de probarlas todas para encontrar la mejor.

De esta manera, se sigue el ejemplo de lo realizado por Libelium [2]. Ellos definieron 10 modos de configuración, generando combinación de BW, CR y SF de manera que el modo 1 fuera el de mayor distancia y menor velocidad, y el modo 10 el de menor distancia, mayor velocidad y menos consumo de batería.

De aquí en adelante se utiliza esta definición de parámetros para referirse a la configuración LoRa.

7.3. Implementación

Gran parte de los componentes a utilizar ya fueron definidos en el Capítulo 6. Dado que los equipos utilizados son casi los mismo exceptuando pequeñas diferencias, no se detallarán los componentes a utilizar en esta sección. Se desarrollarán elementos adicionales que permiten

Mode	BW	CR	SF	Sensitivity (dB)	Transmission time (ms) for a 100-byte packet sent	Transmission time (ms) for a 100-byte packet sent and ACK received	Comments
1	125	4/5	12	-134	4245	5781	max range, slow data rate
2	250	4/5	12	-131	2193	3287	-
3	125	4/5	10	-129	1208	2120	-
4	500	4/5	12	-128	1167	2040	-
5	250	4/5	10	-126	674	1457	-
6	500	4/5	11	-125,5	715	1499	-
7	250	4/5	9	-123	428	1145	-
8	500	4/5	9	-120	284	970	-
9	500	4/5	8	-117	220	890	-
10	500	4/5	7	-114	186	848	min range, fast data rate, minimum battery impact

Figura 7.2: Modos de configuración LoRa. Tomado de [2]

el funcionamiento como sistema de medición de cobertura.

7.3.1. Nodo LoRa

El nodo se comunicará con la aplicación mediante bluetooth serial, para esto utiliza el módulo HC-05. Este posee un puerto serial por el cual se pueden enviar o recibir datos a través de bluetooth, y los datos que le llegan al módulo a través de bluetooth son transmitidos y recibidos por el puerto respectivamente.

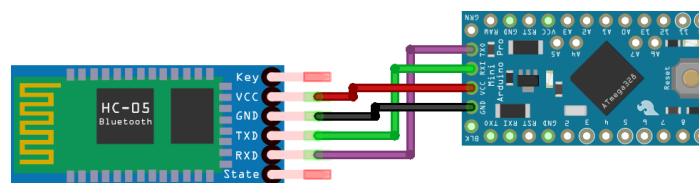


Figura 7.3: Conexión Arduino Pro Mini con módulo HC-05

La implementación del nodo es la misma que la indicada en la Figura 6.10 para el nodo sensor a excepción de los elementos de los sensores de agua y el acelerómetro, se mantuvo el sensor de humedad y temperatura. Como carcasa se utilizó una caja de instalación eléctrica de $11 \times 11 \times 6\text{cm}$. Para este modelo se usó una batería de 2000 mAh con la cual según estimación neta de los consumos de los componentes el nodo debería tener una autonomía de 3 días completos de prueba. Todo se organizó dentro de la caja como se puede ver en la Figura 7.4.

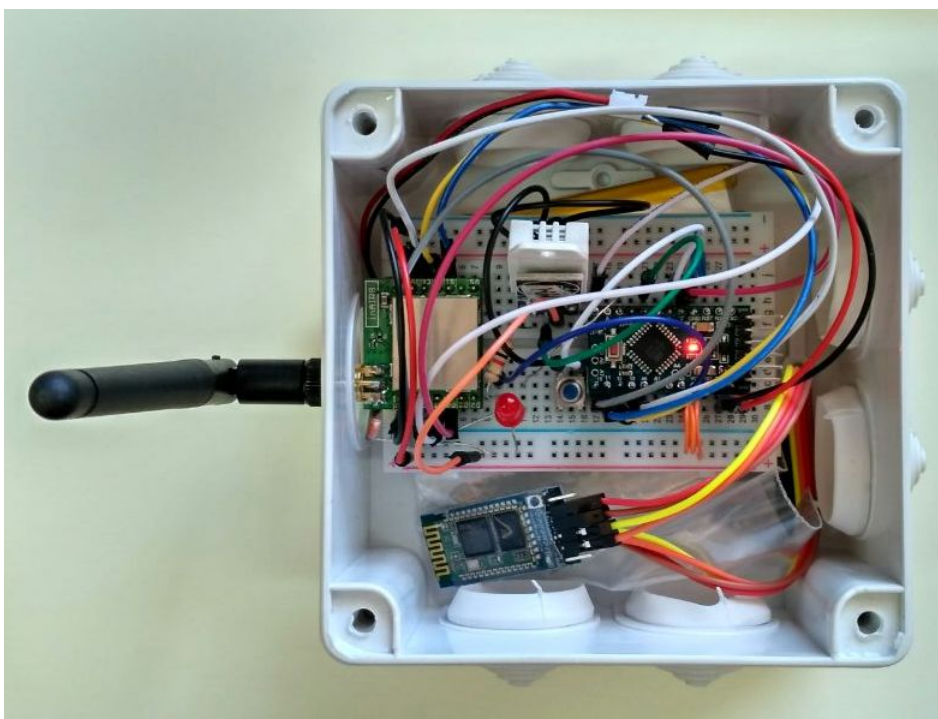


Figura 7.4: Módulo LoRa implementado para pruebas de cobertura

7.3.2. Aplicación Móvil (*Loreap*)

La aplicación se desarrolla para celulares con sistema operativo Android y funciona en conjunto con el nodo LoRa. Sus funciones principales son tres.

1. Hacer de interfaz de control y monitor del nodo, permitiendo: (1) configurar perfil de maestro o esclavo, (2) iniciar o detener la transmisión, (3) ajustar la potencia y los parámetros LoRa de la comunicación y (4) mediante un monitor mostrar los paquetes enviados o recibidos por el nodo.
2. Hacer un seguimiento de la localización de los dispositivos a lo largo de la prueba de cobertura, asignando a cada paquete enviado o recibido una posición geográfica según latitud y longitud. Esto permitirá posteriormente visualizar el perfil de cobertura del terreno en un mapa.
3. Almacenar los datos de la prueba en la memoria del celular en formato excel para luego ser exportado o enviado por alguna aplicación.

Se pueden observar en la Figura 7.5 las tres vistas de la aplicación. La principal es mediante la cuál se realiza la prueba de cobertura. Aquí se controla y observa lo que está ocurriendo. Esta se divide en tres áreas de uso: monitoreo en la parte superior, área de control en la parte media y conexión e inicio de pruebas en la parte inferior.

En la parte superior a través de un monitor se despliegan todos los mensajes enviados por el arduino vía bluetooth, estos son los paquetes que llegan o son enviados en los casos de maestro y esclavo respectivamente. Además se muestra información de localización actual, tiempo de prueba y paquetes enviados.

El área media de control permite iniciar o detener la transmisión, configurar la potencia de transmisión para el nodo esclavo y el modo LoRa de transmisión (o recepción). Además permita agregar información acerca del lugar actual y la altura a la que se encuentra el nodo. Por último en la parte inferior se encuentra el botón que enlaza la aplicación con el nodo y comienza la prueba.

La segunda vista de la aplicación consiste en una lista que permite revisar los archivos de prueba generados hasta el momento. Cada vez que se termina una conexión con el nodo, se guarda un archivo con los datos generados en esa conexión. La tercera vista permite configurar el perfil de funcionamiento y el dispositivo bluetooth a conectarse.

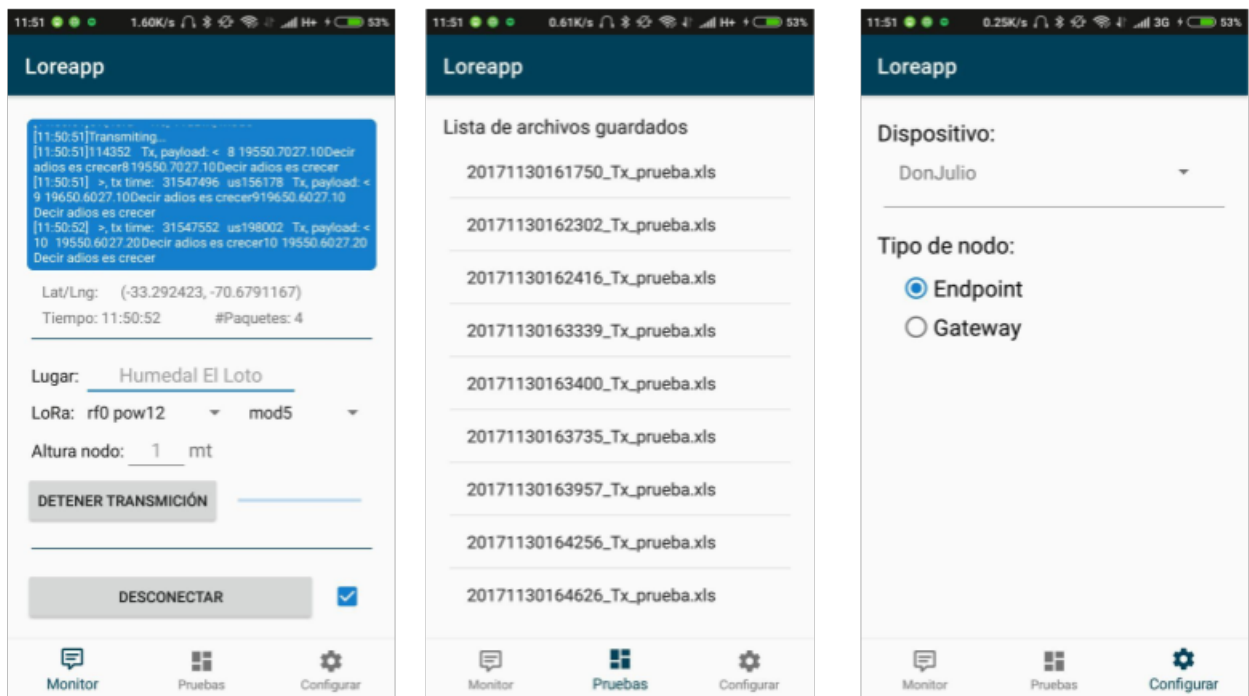


Figura 7.5: Aplicación Loreapp desarrollada para Sistema de Estudio de Cobertura LPWAN

El almacenamiento de los datos de prueba se hace en archivos excel formato *.xls*. Estos se generan de manera automática y una vez terminada la prueba solo es necesario enviarlos del celular al computador para su análisis. Los datos almacenados son los siguientes:

- Rol (maestro o esclavo)
- Número de paquete
- Paquete
- LoRa PA_BOOST (true or false)
- LoRa Power
- LoRa Mode
- Hora
- Fecha
- Latitud
- Longitud
- Altura
- Lugar
- RSSI (Sólo nodo maestro)

7.3.3. Gateway

La arquitectura del gateway utilizado para la medición de cobertura es la misma usada en la red final de sensores. El uso del gateway en la medición de cobertura aumenta mucho la eficiencia de esta.

El gateway guarda todo los datos de cobertura y paquetes en un archivo de datos en su memoria. También existe la posibilidad de subirlos a Google Sheets. Se debe antes crear una archivo con formato deseado para después referenciarlo.

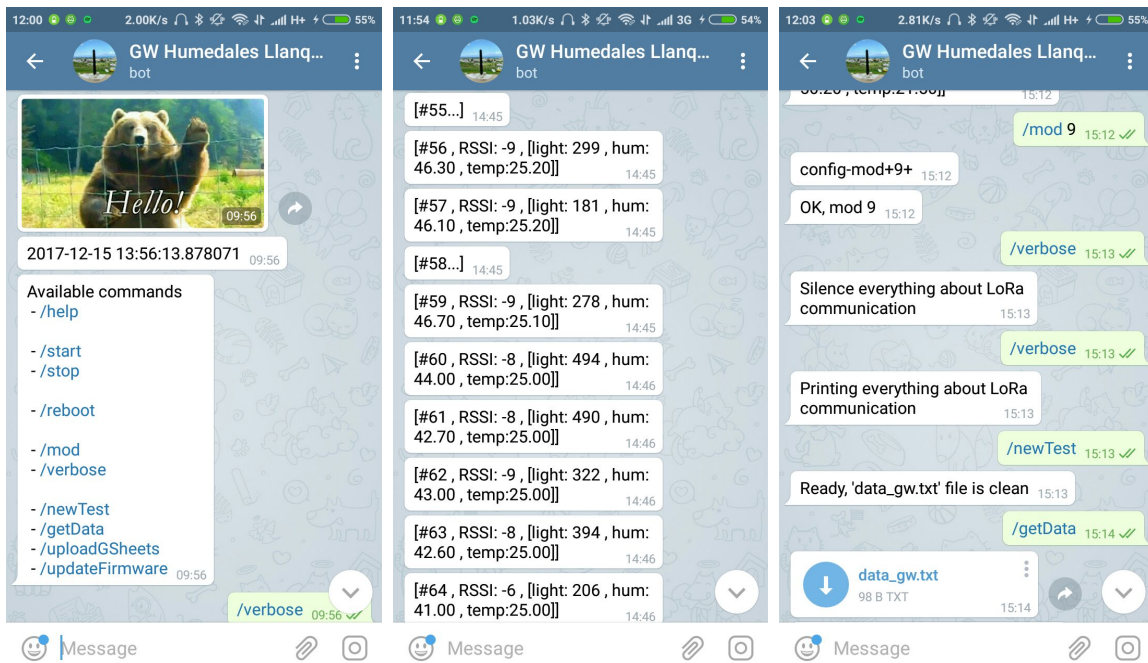
Para una comunicación persona-gateway que permita a la primera cambiar las configuraciones del canal, el gateway implementa un Bot de Telegram. Este funciona generando un usuario de telegram al gateway con el cual se puede chatear. A través del chat se envían comandos para que el gateway cambie su configuración de modos LoRa.

El tener un chat con el gateway permite desarrollar un conjunto de funcionalidades interesantes. No sólo es posible escribirle al gateway, sino que el gateway también puede escribir al usuario. De esta manera cada vez que se recibe un paquete LoRa en el gateway, éste puede enviar un mensaje con el RSSI recibido.

Se implementan 6 comandos de instrucciones al gateway:

- *mod*: Permite configurar el modo LoRa de funcionamiento del gateway.
- *reboot*: Reinicia el gateway.
- *verbose*: Se habilita el envío de un resumen de cada paquete que llega al gateway.
- *newTest*: Borra los datos de prueba guardados en el gateway hasta ese momento reiniciando el archivo de datos.
- *getData*: Envía el archivo de datos de pruebas al usuario.
- *updateFirmware*: Descarga y auto-instala última versión del firmware del gateway. Se reinicia al final del proceso.

Se observa un ejemplo de interacción con el gateway en la Figura 7.6. La Figura 7.6a es la rutina de inicio del gateway, después de un saludo procede a desplegar una lista de los comandos disponibles. La Figura 7.6b presenta al gateway en modo de operación durante una prueba de cobertura. Se puede observar cómo información en tiempo real respecto de los paquetes recibidos está disponible ahora para el usuario. La Figura 7.6c presenta los demás comandos importantes en funcionamiento. Una función interesante es mediante la cuál el gateway hace envío del archivo de datos de prueba de la sesión.



(a) Inicio del gateway (b) Prueba de cobertura (c) Demás funcionalidades

Figura 7.6: Vistas del chat con Bot Telegram implementado en Gateway

Todas estas funciones otorgan una gran flexibilidad al sistema. Funcionalidades como poder actualizar el firmware de manera remota permiten dejarlo instalado en un localización y poder seguir desarrollando el sistema sin tener que acudir a actualizarlo. Esta característica además aumenta la capacidad de respuesta del sistema permitiendo que ante cualquier tipo de error o *bug*, éste sea actualizado de manera remota.

Al mismo tiempo, la seguridad es un aspecto importante a tener en cuenta en la implementación de este Bot, que otorga acceso directo a la funciones del gateway. Como primer paso el definir un conjunto de usuarios administradores ayuda en esta labor, los cuales serán filtrados por sus id's de usuarios únicas de telegram. Posteriormente se debe evaluar la implementación de más mecanismos.

7.3.4. Visualización

Para visualizar los datos de cobertura se utilizó el software de visualización de datos *Tableau*. Este software posee una versión pública gratuita y se encuentra disponible para sistemas Mac y Windows.

Los datos georeferenciados permiten ser dispuestos sobre un mapa. Cada paquete en el mapa se representa por un círculo donde su tamaño y color variaran con base en el RSSI percibido en esa zona.

Capítulo 8

Base de Datos y Plataforma de Datos Abiertos

8.1. Base de Datos

La base de datos consiste de 5 tablas. La primera, guarda los detalles de instalación de cada nodo: localización, id. único de nodo y fecha de instalación. La segunda tabla guarda detalles de operación del sistema: fecha de último cambio de batería para cada nodo, fecha de última calibración para cada sensor y última fecha de cambio de cada sensor.

La tercera tabla es la tabla de datos, esta guardará toda la información de monitoreo a lo largo del proyecto. Una variable relevante de esta tabla es el id. único asignado por medición. La lista completa de variables:

- | | | |
|----------------|------------------------|-----------------------|
| – Id. medición | – Electroconductividad | – Salinidad |
| – Hora | – Oxígeno disuelto | – Humedad interna |
| – Fecha | – Temperatura | – Temperatura interna |
| – Id. nodo | – Turbidez | – Nivel batería |
| – pH | – Sólidos disueltos | |

La cuarta tabla almacena información de los eventos en el sistema: id. evento, hora y fecha. La quinta y última tabla define los tipos de evento y un número identificador.

Se realiza una estimación de los datos que generará la plataforma de monitoreo a lo largo de un año de manera de proyectar la cantidad de memoria necesaria para soportar la plataforma. Por cada medición se generan 78 Bytes de datos que son enviados cada 5 minutos. Al cabo de un año, el sistema completo de 10 nodos acumula aproximadamente 81MB de datos, se puede observar el cálculo en la Figura 8.1. Se asume que todos los nodos son idénticos y envían la misma cantidad de datos a la misma frecuencia.

Nodo		
x paquete	78	Bytes
frecuencia	5	minutos
Datos p/ día	22.464	KB
Datos p/ mes	685.152	KB
Datos 1 año	8.221824	MB
Total sistema		
# nodos	10	
Datos totales p/ día	224.64	KB
Datos totales p/ mes	6.7392	MB
Datos totales 1 año	80.8704	MB

Figura 8.1: Estimación de los datos generados por el sistema de monitoreo en 1 año.

La base de datos se almacenará en Google Drive. Esto permite una implementación rápida, segura y sin costo. Pero no será la única copia de la base de datos, esto otorga redundancia de la información en el sistema. El gateway almacena todos los datos de forma local antes de subirlos, y los respalda periódicamente en un servicio en la nube.

El formato de tablas tipo excel es de común entendimiento lo que permite a cualquier persona de la organización revisar los datos y generar filtros y gráficas de manera rápida. Los permisos de acceso a esta data son primordiales. Sólo deben poder administrarla un grupo determinado de usuarios definidos.

8.2. Plataforma de Datos

El sistema de monitoreo a implementar cumple un rol científico y de regulación, pero además los datos generados significan algo, más aún en un entorno social que quiere darle sentido y relevancia a lo que ocurra en su ciudad y busca medios para generar consciencia de su entorno.

Por esto fundamental para el proyecto que los datos sean abiertos y accesibles. Esto permite generar una base de datos pública de los estados de salud del humedal. Además sirve como base de conocimiento para otros proyectos.

Hoy no existen buenas plataformas abiertas de datos ambientales y de aguas en Chile y el extranjero. Hay algunas con información georeferenciada pero son incómodas, lentas y difíciles de interpretar. Existe entonces un amplio espacio para desarrollar una buena solución en este tipo de áreas. Se implementa entonces el *mock-up* de una página abierta de datos ambientales y de aguas, de momento, en Llanquihue.

8.2.1. Interfaz de Usuario

Se hace enfoque en un diseño atractivo a través de pocos elementos. No se incluirán cosas como administración del sistema ni muchas opciones las cuales desconcentrarán al usuario, alejándolo del objetivo final de la página que es contarle al usuario que está ocurriendo con los humedales y el medio ambiente de Llanquihue a través de la información generada en un sistema en terreno.

En una sola página a la izquierda se despliega un mapa de Llanquihue y sus humedales. El mapa también minimalista permitirá tener una idea del territorio y asociar los datos a un lugar y tiempo específico. La inclusión del mapa también pretende generar pertenencia de la información desplegada y creada. En el área derecha se despliegan los demás elementos de la página.

Se presenta el diseño de la página en la Figura 8.2, un marco negro la separa del fondo del documento. Se delimita el área visible al ingresar desde un navegador oscureciendo levemente lo que queda fuera del campo de visión. Al navegar hacia la parte inferior de la página el mapa se mueve junto con la vista y es siempre visible en la parte izquierda.

La maqueta de la página se encuentra disponible en <https://matimacayal.github.io/cyber-data-llanquihue/>.



Figura 8.2: Interfaz gráfica de página abierta de datos ambientales y de aguas.

Capítulo 9

Pruebas, Resultados y Análisis

9.1. Pruebas de Cobertura

Fueron realizadas tres pruebas de cobertura en terreno. Estas permitieron validar la tecnología y definir el lugar de instalación del gateway. Las primeras dos pruebas fueron realizadas en Octubre del año 2017 y fue el primer acercamiento al terreno y a la tecnología LoRa en espacios abiertos. La segunda serie de pruebas fue realizada en Noviembre de 2017 para confirmar la posición del gateway.

Se han desarrollado mediciones de cobertura previas en zonas urbanas de Santiago [30]. Esta es la primera en base a lo investigado en una zona rural chilena y puede ser interesante realizar un contraste con la zona urbana. Tener en cuenta que Llanquihue no cuenta con edificios de más de tres pisos y su geografía en la parte central no posee grandes relieves.

9.1.1. Primera Aproximación

Estas fueron las primeras pruebas realizadas con LoRa en Llanquihue y conformaron una primera exploración del terreno. Permitieron definir una posición tentativa del gateway y entender como se comporta la tecnología en el terreno.

La primera prueba se realizó con ambos nodos a nivel del suelo y el estudio se enfocó en la eficacia en la entrega de paquetes. La prueba fue realizada mediante dos personas. El nodo receptor se situó en el techo de una casa de dos pisos vigilado por su encargado. El nodo transmisor es cargado a nivel del pecho por la segunda persona. No existe una línea de visión directa entre transmisor y receptor y cabe destacar que para esta prueba la zona más edificada del pueblo se encuentra justo en la mitad del enlace de comunicación.

El clima del día fue nublado y hubo 9mm de leves precipitaciones, la temperatura estuvo entre 11° y 5°C. Se utilizó para toda la prueba el modo 1 de configuración LoRa a una potencia de +14dBm. Este modo provee un largo alcance a menor velocidad de transmisión.

En la Figura 9.1 se presenta el camino recorrido. El nodo maestro o receptor se indica con un hexágono amarillo junto con una referencia de distancia. Cada punto azul es un paquete enviado, cada círculo celeste es un paquete recibido.

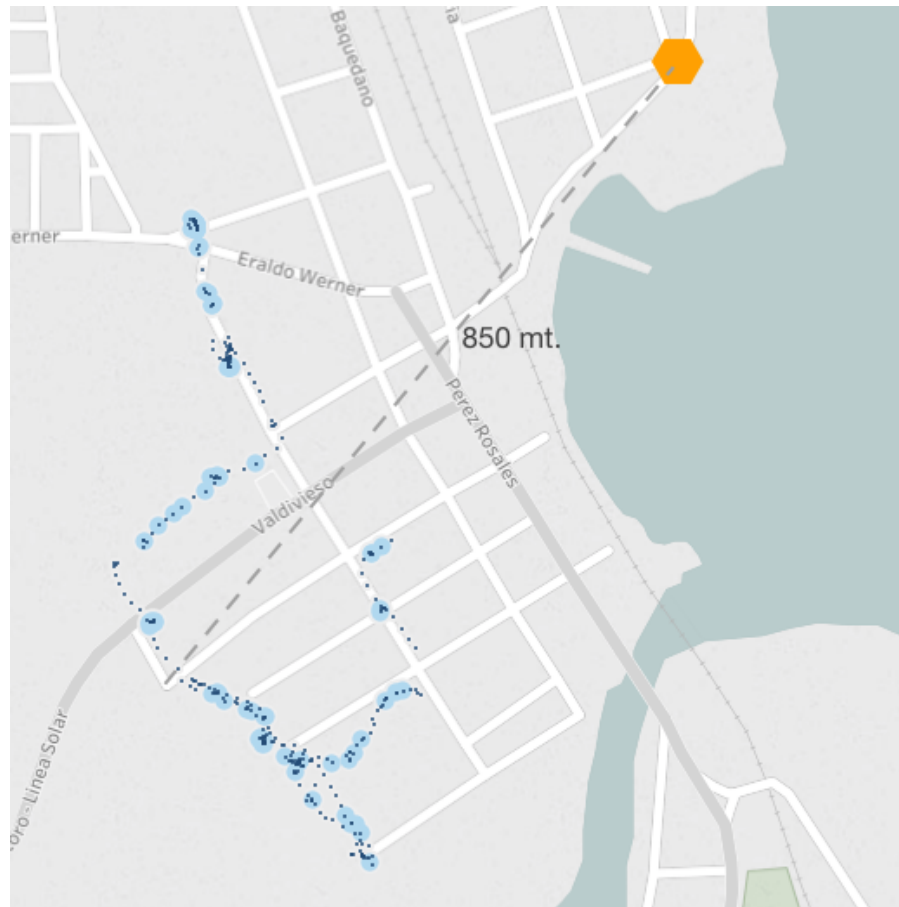


Figura 9.1: Primera prueba de cobertura en Llanquihue.
<http://tabsoft.co/2De5gwA>.

Se puede observar que existe recepción de paquetes pero esta no es muy buena. Hay zonas en las que definitivamente no hay cobertura. No parece haber a este nivel una relación directa con la distancia de transmisión. Se obtiene un PER de 0.88 lo cuál es muy elevado para el tipo de enlace requerido.

El desempeño del enlace en la primera prueba no fue bueno, teniendo además en cuenta que se utilizó el modo 1 de canal LoRa. Cómo principal causa se atribuye a la baja altura del nodo receptor.

9.1.2. Segunda Aproximación

Para la segunda prueba de cobertura se situó el gateway a una cota de 100 metros, que otorga una diferencia de altura al nodo receptor de aproximadamente 80 metros. Se observa en la Figura 9.2 un mapa de relieve de Llanquihue, junto con la nueva posición del nodo receptor en morado.

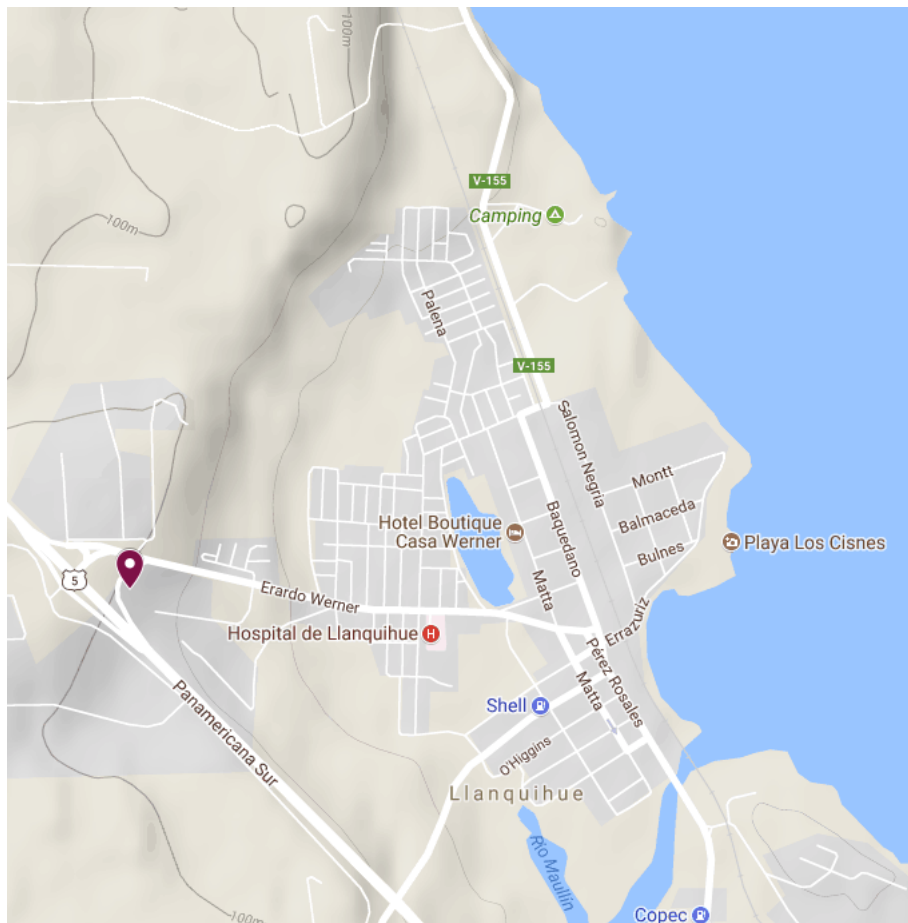


Figura 9.2: Mapa de relieve de Llanquihue

Nuevamente la prueba fue realizada con dos personas, una fija en el nodo receptor y otra en terreno realizando el recorrido indicado en la Figura 9.3. Ahora una línea de visión directa es posible entre el transmisor y receptor pero no se mantiene siempre, depende de la posición en el recorrido. Se utilizó en todo momento el modo 1 de transmisión a una potencia de +14dBm. El clima del día fue de 10°/5°C de temperatura con 9mm de precipitaciones parciales. El gateway se indica en la figura cómo un hexágono verde. Además de estudiar la eficacia en la entrega de paquetes se analiza la intensidad de la señal mediante el RSSI recibido.

Nuevamente cada punto azul es un paquete enviado y un círculo naranja es un paquete recibido. A primera vista existe un mejor desempeño que para la prueba anterior. Vuelven a existir zonas de no cobertura pero son pequeñas y puntuales.

Se puede observar cómo varía la intensidad de la señal con base en el color de los círculos. Un color más oscuro es indicador de una señal más fuerte. La mejor señal se obtiene en la parte central de la ruta hacia la izquierda. En la parte superior de la ruta al igual que en la prueba pasada se pierde calidad de la señal. En zonas más lejanas al gateway la intensidad de la señal decrece lo cual es un correcto comportamiento. Existen zonas puntuales en que estando más lejos se tienen niveles de señal muy buenos, esto ocurre porque se generan momentos de visión en línea directa lo que aumenta inmediatamente la calidad de la señal.

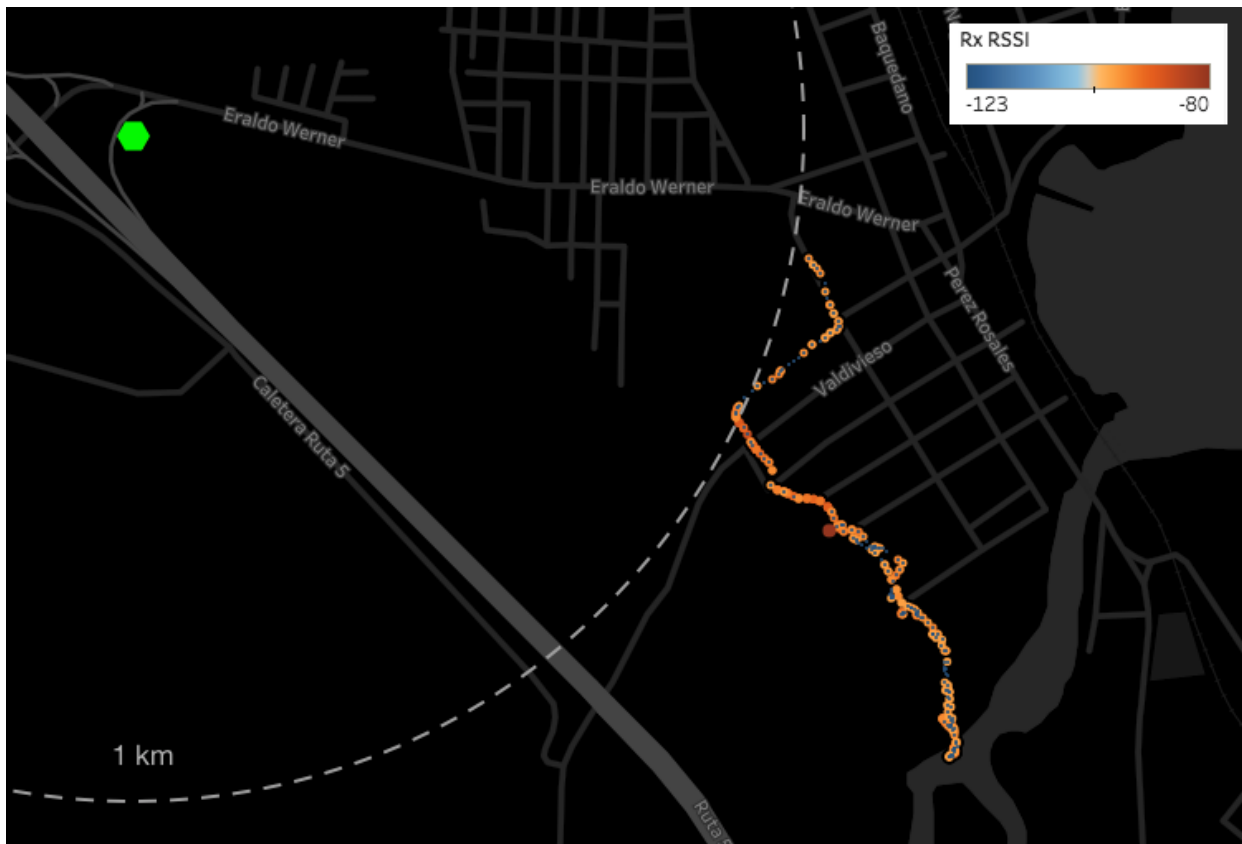


Figura 9.3: Segunda prueba de cobertura en Llanquihue.
<http://tabsoft.co/2mFnNY5>

El PER total de esta prueba fue de 0.57, es decir, un poco más de la mitad de todos los paquetes se perdieron. Esto no es un buen resultado a primera vista, pero es mejorable y lo suficientemente bueno como para confirmar la utilización de LoRa en el sistema de monitoreo.

El recorrido es casi igual al de la prueba pasada por lo que se determina que ambas pruebas son comparables entre sí. De esta manera la mejora en el desempeño de PER 0.88 a 0.57 en esta prueba se atribuye al único cambio realizado que fue el lugar del GW.

9.1.3. Tercera Aproximación

Ya teniendo una idea más clara de cómo es el territorio y el comportamiento de LoRa en este se procede a evaluar un desempeño general pero también más detallado en lugares puntuales y específicos de interés dado por la posibilidad de instalación de nodos de monitoreo en estos lugares. Esto permitirá el tener antecedentes suficientes que justifiquen una instalación definitiva del gateway en el lugar propuesto en la prueba 2.

En esta ocasión la prueba fue realizada por una sola persona en base a la metodología de medición descrita en el Capítulo 7 mediante la utilización del gateway, el nodo LoRa, la aplicación Loreapp y la aplicación Telegram. La utilización de este sistema permitió la posibilidad de poder cambiar los parámetros LoRa en terreno en base a los resultados obtenidos

en tiempo real. De esta manera se utilizó el modo 7 para comenzar la pruebas, este modo permite una comunicación más rápida y de menor consumo. Este modo se siguió utilizando durante la mayoría del recorrido excepto en lugares puntuales de estudio donde existía baja cobertura. El clima fue despejado sin lluvia y con una temperatura de entre 17° y 7°C.



Figura 9.4: Tercera prueba de cobertura en Llanquihue.
<http://tabsoft.co/2D3ZmtQ>

La ruta seguida esta ocasión fue más extensa y tomo en cuenta detenciones en los lugares de interés. Estos se marcan con globos en la Figura 9.4. Se disponen circunferencias que delimitan la distancia al gateway de 500mt, 1km y 1.5km. Nuevamente la intensidad de la señal disminuye consistentemente con la distancia. Existe buena cobertura general de la zona aunque se siguen perdiendo paquetes. El PER de esta prueba completa fue de 0.23.

En la Tabla 9.1 se resume el rendimiento de la cobertura en los lugares de interés.

Estas medidas son útiles debido al tipo de instalación que se efectuará. La antena estará a baja altura por razones de diseño del proyecto, por esto se realizaron pruebas de medición

Lugar	Identificador	RSSI medio	PER
Detrás de cerro y grupo de arboles	1	-81dBm	0
Orilla Humedal El Loto	2	-92dBm	0
Plaza orilla Este de El Loto	3	-83dBm	0.28
Orilla Este Humedal El Loto	4	-85dBm	0
Piso inicio Hume dal Helechos	5	-94dBm	0.48
Estero detrás Humedal Helechos	6	-97dBm	0.08
Estero antes de Humedal Las Ranas	7	-99dBm	0
Suelo tótem en Las Ranas	8	-99dBm	0.49
Orilla Este del Humedal Las Ranas	9	-98dBm	0
Suelo Orilla Este Humedal Las Ranas	10	-99dBm	0.39
Suelo Orilla del Río Maullín, Las Ranas	11	-99dBm	0.28

Tabla 9.1: Rendimiento de LoRa en lugares de interés en Llanquihue.

con el transmisor a nivel de suelo. Se observa que aquellas mediciones con antena a un metro de altura poseen bajos valor de PER cercano a 0. Pero al llevar el transmisor al suelo, si bien el RSSI se mantiene, el PER aumenta a casi 0.5 para todas las ocasiones.

El único lugar que se desempeña relativamente bien bajo condiciones de transmisión a baja altura es curiosamente el punto 11, el lugar más lejano del nodo. Esto se explica ya que si bien es el lugar más lejano, al recorrer la zona es aquel con menos arboles grandes alrededor, y no posee fuentes de atenuación cercanas. Para las demás mediciones a nivel de piso, existía frondosa vegetación, grandes arboles cercanos y eran espacios más cerrados por los elementos que las circundaban. Existen documentación de estos mismo efectos en otros estudios [9].

Lo que ocurre es que al llevar el transmisor al suelo, los efectos de atenuación e interferencia del entorno se incrementan y su efecto se vuelve mucho más dañino. Este comportamiento es indicio de dos cosas, por un lado, la fuerte influencia de la vegetación y grandes obstáculos en la transmisión de una señal LoRa. Y por otro lado, el fuerte impacto que el variar la altura genera en la interacción con los elementos circundantes y la pérdida neta de calidad de transmisión.

9.2. Pruebas Prototipo del Sistema

Se puso a prueba el funcionamiento del prototipo del nodo de monitoreo más el gateway. Esta prueba consiste en la medición de pH y Oxígeno Disuelto de manera remota desde un recipiente de agua, los cuales son recibidos por el gateway que los sube a Internet.

Esta prueba realiza un testeo integrado de todos los sistemas. No conlleva el uso de una carcasa y como se describió en el Capítulo 6, los equipos de monitoreo presentaron una serie de problemas asociados lo que llevo a la simplificación del diseño utilizada en este momento.

La utilización de una batería proporciona autonomía al nodo de monitoreo y como se

puede observar en la Figura 9.5 se utiliza una extensión SMA para la antena. Los sensores utilizados son de pH y Oxígeno Disuelto y fueron puestos en un recipiente de agua sumergidos durante 12 horas. Se realizaron mediciones cada 2 segundos.

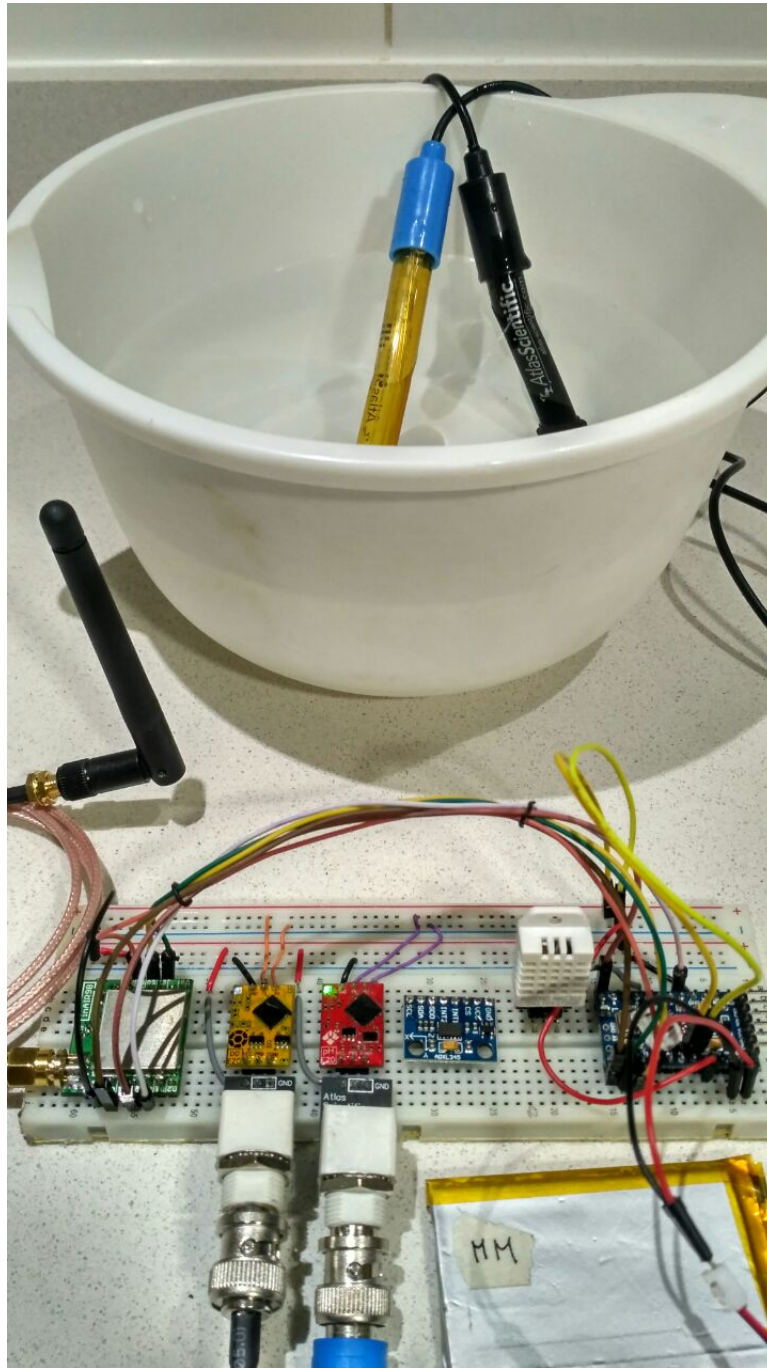


Figura 9.5: Configuración para prueba de prototipo

El dispositivo realiza de manera autónoma las tareas de adquirir los datos periódicamente, almacenarlos de manera temporal y enviarlos al gateway para su almacenamiento local y respaldo en la base de datos en Internet.

El gateway se encuentra conectado a Internet mediante Wi-Fi y con alimentación eléc-

trica fija. Este se encuentra constantemente esperando nuevos paquetes y al recibir uno los almacena en memoria y los sube a un Spreadsheet en Google Drive como se observa en la Figura 9.6.

The screenshot shows a Google Sheet interface with the title "Datos monitoreo remoto de pH y Oxígeno Disuelto". The spreadsheet contains a table with 8 columns: Time, uC Clock [ms], Packet Number, RSSI [dBm], Temperatura, Humedad, pH, and Oxígeno Disuelto. The data is recorded from 21:53:29 to 21:58:58. The pH values are consistently around 7.5, while the dissolved oxygen values fluctuate between 6.27 and 10.55.

	A	B	C	D	E	F	G	H
1	Time	uC Clock [ms]	Packet Number	RSSI [dBm]	Temperatura	Humedad	pH	Oxígeno Disuelto
2								
3	21:53:29	1047662	46	-34	25.6	43.1	7.52	9.93
4	21:53:37	1057997	47	-36	25.6	43.2	7.48	9.92
5	21:53:48	1068335	48	-37	25.6	43.2	7.48	9.9
6	21:53:58	1078671	49	-37	25.6	43.3	7.5	9.91
7	21:54:08	1089006	50	-41	25.6	43.3	7.52	9.91
8	21:54:19	1099341	51	-32	25.6	43.4	7.55	10.55
9	21:54:29	1109678	52	-40	25.6	43.4	8.17	9.25
10	21:54:39	1120013	53	-44	25.6	43.5	7.91	8.45
11	21:54:50	1130348	54	-41	25.6	43.5	7.83	7.6
12	21:55:00	1140683	55	-38	25.6	43.5	7.78	7.13
13	21:58:07	1327212	56	-39	25.6	43.5	7.74	6.86
14	21:58:17	1337547	57	-50	25.6	43.6	7.7	6.68
15	21:58:27	1347884	58	-48	25.6	43.7	7.67	6.55
16	21:58:38	1358219	59	-49	25.6	43.7	7.64	6.43
17	21:58:48	1368555	60	-50	25.6	43.7	7.61	6.34
18	21:58:58	1378891	61	-45	25.6	43.7	7.59	6.27

Figura 9.6: Google Sheet con datos de medición remota

Los datos adquiridos se presentan en las Figuras 9.7 y 9.8. Se observa un comportamiento casi constante del pH. Por otro lado el oxígeno disuelto es más dinámico y posee fluctuaciones pronunciadas en una tendencia a decaer la cual se verá invertida durante la segunda parte de la prueba. En un momento de la prueba, se abre la ventana del laboratorio situada sobre el prototipo, esto ocurre en la medición 1033. El oxígeno disuelto se eleva abruptamente para luego volver a decaer. La humedad aumenta y la temperatura comienza a disminuir en este punto. El pH se mantiene constante ante esta perturbación.

9.3. Análisis Económico

Un desarrollo de bajo costo era uno de los principales objetivos de este trabajo, y no solo bajo costo sino menor al de las soluciones actuales de mercado. Además, se contaba con un límite de presupuesto de 7 millones de pesos para la tecnología de todo el sistema de monitoreo.

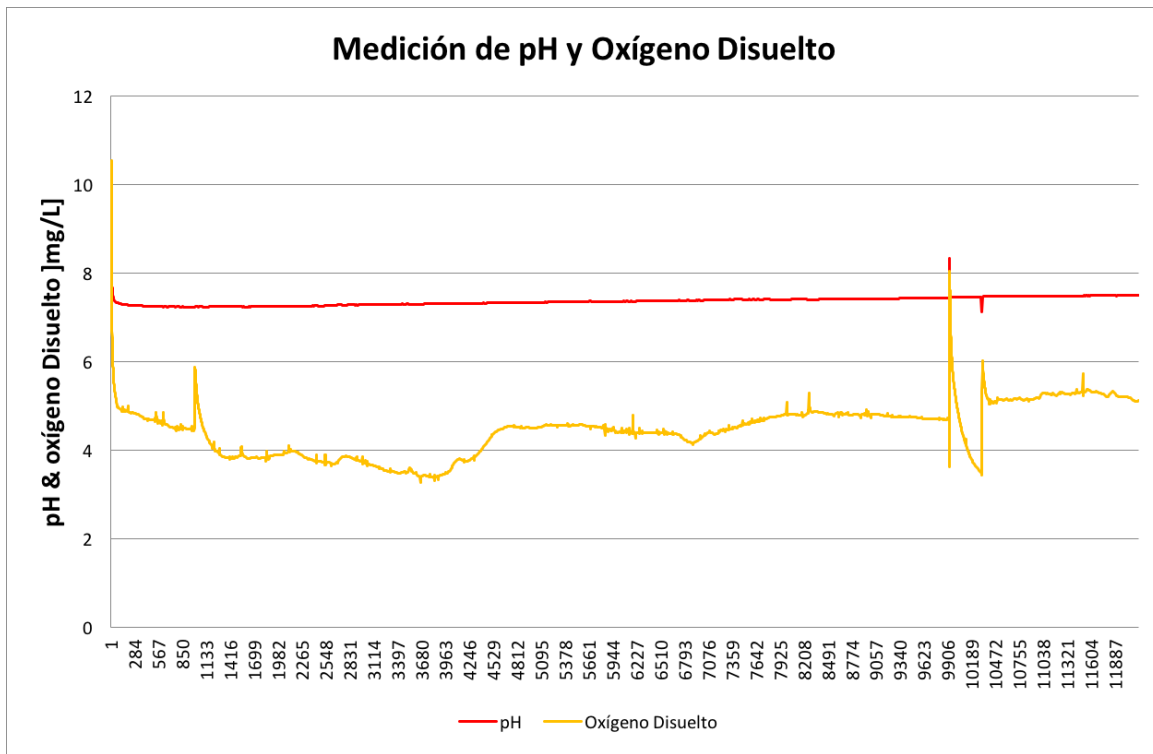


Figura 9.7: Datos de pH y Oxígeno Disuelto obtenidos de manera remota desde un recipiente de agua.

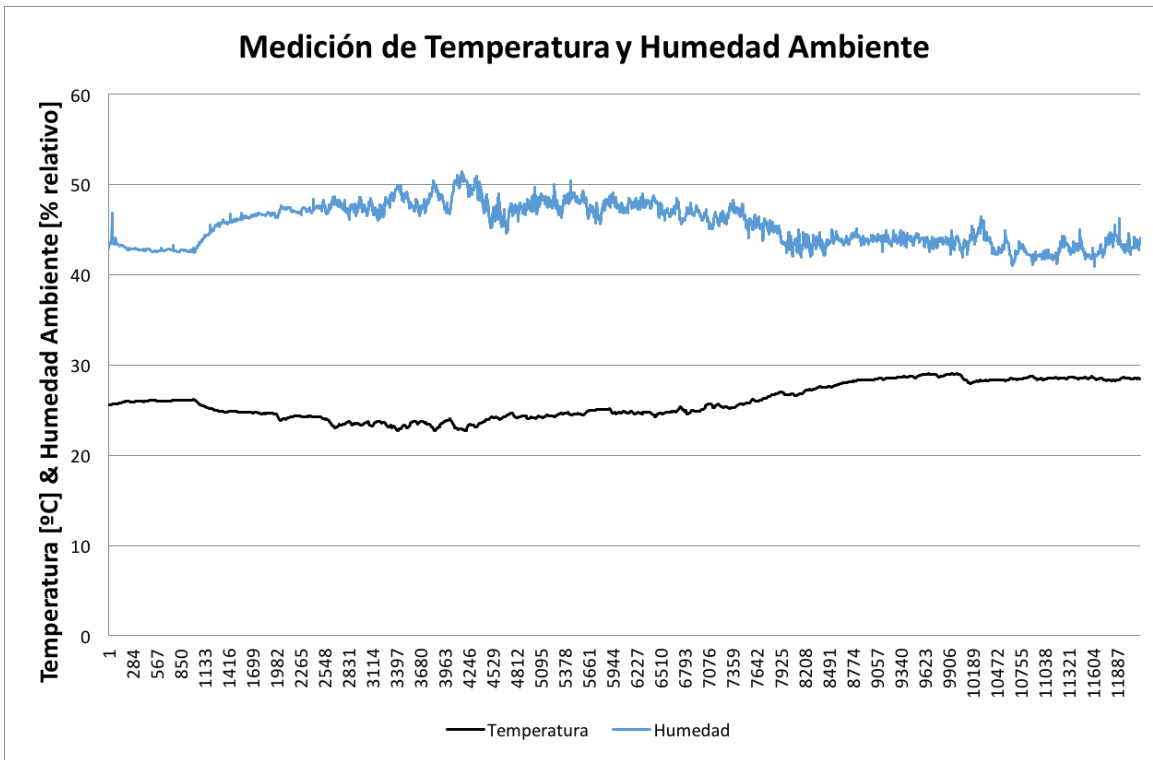


Figura 9.8: Datos de Temperatura y Humedad Ambiental obtenidos de manera remota

En la tablas de la Figura 9.9 se detallan todos los componentes necesarios para construir los nodos y un gateway. Además, en el contexto del proyecto, se consideraron componentes extras de repuesto para reparación y o cambio de componentes durante la operación del sistema. En base a estos costos se estiman costos individuales por equipo y del sistema completo para instalación de 10 estaciones de monitoreo y un gateway.

Elemento	Cant. p/nodo	Cantidad total	Precio CLP\$	Precio USD\$	Subtotal p/nodo	Subtotal CLP\$
Prototipo 1 Nodo						
Arduino Pro Mini	1	12	\$3,890		\$3,890	\$46,680
Protoboard Grande	2	6	\$3,290		\$6,580	\$19,740
Batería 6000 mAh	1	15	\$26,990		\$26,990	\$404,850
Comunicación						
inAir9B	1	15		\$15.95	\$9,730	\$145,943
Antena	1	15		\$4.95	\$3,020	\$45,293
Extensión SMA	1	15	\$7,990		\$7,990	\$119,850
Sensores						
DHT22	1	15	\$5,390		\$5,390	\$80,850
ADXL345	1	15	\$3,790		\$3,790	\$56,850
Turbiedad	1	15	\$8,500		\$8,500	\$127,500
pH Probe	1	12		\$66.60	\$40,626	\$487,512
pH EZO Circuit	1	12		\$35.15	\$21,442	\$257,298
D.O. Probe	1	12		\$183.15	\$111,722	\$1,340,658
D.O. EZO Circuit	1	12		\$40.70	\$24,827	\$297,924
E.C. Probe (K=1)	1	12		\$181.03	\$110,428	\$1,325,140
E.C. EZO Circuit	1	12		\$53.65	\$32,727	\$392,718
T° Probe	1	15		\$16.65	\$10,157	\$152,348
T° EZO Circuit	1	15		\$24.05	\$14,671	\$220,058
O.R.P. Probe	0	0		\$108.00	\$0	\$0
O.R.P. Circuit	0	0		\$38.00	\$0	\$0
Female BNC	4	50		\$11.70	\$28,548	\$356,850
Voltage Isolator	4	50		\$22.20	\$54,168	\$677,100
Carcasa						
PVC 75mm diam.	1	15	\$800		\$800	\$12,000
Tapa PVC 75mm	2	30	\$800		\$1,600	\$24,000
Extras aislación	1	15	\$5,000		\$5,000	\$75,000
Etc ...	1	12	\$15,000		\$15,000	\$180,000
Subtotal p/nodo					\$527,593	
Envío				\$60.00		\$36,600
Total						\$6,882,760

Figura 9.9: Resumen de costo de nodos del sistema de monitoreo.

Se obtiene un costo individual por cada nodo de monitoreo de CLP\$527.593. De estos, un monto CLP\$449.314 (85 %) es utilizado en la compra de los sensores de calidad del agua. Todo el resto de los componentes suman un total CLP\$78.279. Esto es lo que costaría un nodo de medición de cobertura. El costo total de equipos para el monitoreo asciende a CLP\$6.882.760. Los requerimientos de costos son alcanzados satisfactoriamente.

Expuesto lo anterior, se comparan los costos de un producto real ofrecido en el mercado con una equivalente oferta desarrollada con estos dispositivos. El productor es el *Libelium-*

Prototipo GW						
Raspberry Pi zero	1	1	\$25,890		\$25,890	\$25,890
inAir9B	1	1		\$15.95	\$9,730	\$9,730
Extensión SMA	1	1	\$7,990		\$7,990	\$7,990
Arduino Pro Mini	1	1	\$3,890		\$3,890	\$3,890
Energía						
Enchufe macho	1	1	\$690		\$690	\$690
Cable 5 mt	1	1	\$3,190		\$3,190	\$3,190
Enchufe hembra	1	1	\$1,100		\$1,100	\$1,100
Transformador 5V 2.5A	1	1	\$8,650		\$8,650	\$8,650
Shield batería	1	1	\$14,990		\$14,990	\$15,990
Batería 6000 mAh	1	1	\$26,990		\$26,990	\$26,990
Carcasa						
Cinta doble contacto	1	1	\$4,190		\$4,190	\$4,190
Caja	1	1	\$5,000		\$5,000	\$5,000
Extras aislación	1	1	\$5,000		\$5,000	\$5,000
Etc ...	1	1	\$5,000		\$5,000	\$5,000
Total						\$123,300

Figura 9.10: Resumen de costo de gateway

Devicelynk Smart Water Solution Kit ilustrado en la Figura 9.12 y a la venta en The IoT Marketplace. Este viene equipado con múltiples sensores para monitoreo de calidad de agua: pH, Oxígeno Disuelto, Conductividad y Potencial de Oxido Reducción. Incluye 2 equipos de monitoreo, soluciones de calibración, paneles solares para cada equipo y un gateway Ethernet/4G.

Se iguala su oferta para que la comparación sea válida, incluyendo a nuestros equipos sensor de Potencial Óxido Reducción, paneles solares, soluciones de calibración y un gateway más costoso. Esto se detalla en la Figura 9.11.

Costo Lorea	Elemento	Cant	Precio	Total
	Nodos	2	\$620,000	\$1,240,000
	Gateway	1	\$150,000	\$150,000
	Soluciones cal.	1	\$30,000	\$30,000
	Paneles solares	2	\$40,000	\$80,000
	Total			\$1,500,000
Costo Libelium	Elemento	Cant	Precio Euros	CLP
	Libelium-Devicelynk Smart Water Solution Kit	1	€5,250.00	\$3,833,288
	Ahorro del			60.87%

Figura 9.11: Comparación de costos entre equipo Libelium Devicelynk Smart Water Solution Kit y una oferta equivalente en base a los equipos desarrollados.



Figura 9.12: Libelium-Devicelynk Smart Water Solution Kit

Conclusiones

Se hace un repaso de las conclusiones y aspectos más importantes desarrollados durante el informe.

9.4. Sistema de monitoreo

El prototipo de sistema implementado permite demostrar que es factible, mediante el uso de tecnologías abiertas, generar equipos y plataformas que registren variables de calidad de agua en terreno en una base de datos en Internet.

Se evidenció la falta de cuidado en la ciudad de Llanquihue respecto de sus humedales. Y, habiendo entendido la dinámica de estos, un sistema de monitoreo de los humedales que permita saber en que estado se encuentran sus aguas, se plantea como una buena herramienta de ayuda en la conservación de estos.

Se entendió también que para tener un real impacto es necesario que esta herramienta sea abierta y se deben generar instancias para compartir la información generada. Así la maqueta de plataforma de datos diseñada permite realizar esta labor de manera directa e intuitiva. Se agrega el hecho de que páginas de visualización de datos ambientales no existen muchas en Chile y la calidad de las existentes no son buenas, son difíciles de utilizar, lentas, y no poseen un atractivo que llame a visitarlas. Todo esto agrega valor a la propuesta de página web presentada.

9.5. Prototipos implementados

En la oferta actual de sensores predominan productos privativos tanto en hardware, como en software. Existen soluciones abiertas como la seleccionada pero son de alto costo y tienen problemas al funcionar en sistemas de bajo consumo. Esto abre un espacio al desarrollo de este tipo de equipos. En este sentido también es importante tener en cuenta la finalidad de los datos a generar ya que pueden no ser necesarios sensores de altas prestaciones.

En cuanto los componentes del nodo sensor, los sensores son actualmente el área que más investigación requiere para casos de bajo consumo y buena calidad de los datos. Ya que

presentar varios problemas en cuando a su consumo de corriente y operaciones de funcionamiento. Se propusieron soluciones a estos problemas.

Los sensores son el componente más caro de la solución implementada. La utilización de los sensores seleccionados no solo permitió desarrollar el prototipo sino que el mismo prototipo pueden ser utilizada para generar un equipo de mayores prestaciones al cambiar los sensores por unos mejores, o uno de menores prestaciones pero más económico al utilizar sensores más económicos.

La utilización de sensores de movimiento, y humedad y temperatura internos se presentan cómo buena solución ante la necesidad de que el nodo reaccione si es intervenido sin permiso, y para detectar filtraciones que puedan ocurrir debido a condiciones adversas del clima.

Las consideraciones de energía son uno de los aspectos más importantes a tomar en cuenta. Es importante que al programar y seleccionar y diseñar el *hardware* se piense en eficiencia energética. El realizar una estimación previa del consumo del sistema en base a una lista inicial de componentes permite estudiar el desempeño, y realizar modificaciones antes de comprar. Los requerimientos de energía deben ser bien definidos ya que influyen en otras variables cómo el tamaño del dispositivo.

El periodo de muestreo tiene un impacto directo sobre el consumo energético, esto es lógico pero no muchas veces se tiene en cuenta a la hora de proyectar los requerimientos energéticos del sistema. Existe una zona óptima de operación y en aplicaciones donde este equilibrio sea clave se recomienda graficar esta relación.

Debido al bajo costo de los equipos desarrollados en comparación con otros equipos, si bien falta realizar una versión final, son una conveniente solución a tomar en cuenta a la hora de desarrollar sistemas de monitoreo de aguas, u otros. Además las posibles aplicaciones son muchas y el hecho de centrarse en torno al agua es estratégico dadas las necesidades actuales y futuras. El desarrollo será publicado para libre acceso y se le otorgará una licencia Open-Source.

9.6. Pruebas realizadas

La medición de cobertura de la tecnología LoRa a través del nuevo sistema mediante la utilización del nodos y la aplicación fue exitoso. Este sistema además de facilitar la tarea de medición de cobertura permitió a una sola persona realizar el doble de trabajo en el mismo tiempo.

La utilización de solo 10 posibilidades de configuraciones de canal disponibles facilitó mucho las mediciones de cobertura y se recomienda su utilización. La presencia de vegetación abundante afectó negativamente la calidad del enlace. Y en el momento de decidir la posición del gateway se recomienda priorizar altura versus cercanía, se puso confirmar en la práctica que para grandes distancias aumentar la altura mejora más la calidad de la comunicación que la cercanía.

9.7. Trabajo Futuro

9.7.1. Trabajo Futuro Inmediato

Existen una serie de tareas inmediatas a realizar de manera que los prototipos desarrollados pasen a ser versiones finales listos a ser instalados:

1. Incorporar la medición de todos los sensores.
2. Implementar LoRaWAN en la red, o desarrollar protocolos de confirmación de recepción de paquetes. En vista de una entrega confiable de los paquetes y evitar colisiones.
3. Desarrollo de una carcasa adecuada.
4. Diseñar una PCB para el nodo de monitoreo que acople todos los sistemas en un menor tamaño y de manera más segura. Esta además permitirá diseñar el hardware especializado de bajo consumo tanto para el procesador cómo para utilizar los sensores en un esquema de bajo consumo.
5. Programar el nodo pensando en bajo consumo energético.

9.7.2. Trabajo Futuro General

Una vez que el sistema se encuentre instalado y funcionando, se pueden incorporar múltiples funciones y mecanismos que otorguen mayor desempeño a la red, tanto para estudio, como para pruebas o actualización de tecnología.

Se propone una lista con algunas opciones:

1. Realizar conexión full IPv6 de los dispositivos de monitoreo.
2. Implementar protocolos de configuración adaptativa del canal.
3. Abrir la plataforma de datos ambientales para que otras organizaciones o individuos puedan subir sus datos ambientales. Esto requerirá una implementación del *back-end* de la plataforma.
4. Desarrollar nueva tecnología de sensores.
5. Probar nuevas tecnologías LPWAN y comparar desempeños.
6. Generar modelos de comportamiento y alerta temprana en los humedales.

Bibliografía

- [1] Flespi, “Top 7 technologies for iot connectivity 2017.” [Online; posted 2017].
- [2] Libelium Comunicaciones Distribuidas S.L., *Waspote-LoRa-868MHz_{915MHz} – SX1272, NetworkingGuide*, 2017.
- [3] U. National Weather Service, “History of the national weather service.” [Online; posted 2017].
- [4] T. in Education Magazine Online, “Datalogging – a history.” [Online; posted 2017].
- [5] F. L. Lewis *et al.*, “Wireless sensor networks,” *Smart environments: technologies, protocols, and applications*, vol. 11, p. 46, 2004.
- [6] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, “Sensorscope: Out-of-the-box environmental monitoring,” in *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 332–343, IEEE Computer Society, 2008.
- [7] Semtech, “What is lora®?.” href<https://www.semtech.com/technology/lora/what-is-lora>[Online; posted 2018].
- [8] RevSpace, “Decodinglora.” [Online; posted 2018].
- [9] O. Iova, A. L. Murphy, G. Pietro Picco, L. Ghio, D. Molteni, F. Ossi, and F. Cagnacci, “Lora from the city to the mountains: Exploration of hardware and environmental factors,” in *EWSN MadCom Workshop, At Uppsala, Sweden*, feb 2017.
- [10] T. T. Network, “Lorawan.” [Online; posted 2017].
- [11] L. C. D. S.L., “Waspote plug sense technical overview.” [Online; posted 2017].
- [12] P. Lab, “Riffle: an open source water monitoring approach.” [Online; posted 2017].
- [13] W. EU, “A diy low-cost lora gateway.” [Online; posted 2017].
- [14] W. EU, “Building your lora devices.” [Online; posted 2017].
- [15] C. C. L. M. L. C. S. G. J. A. E. C. R. Contreras Vargas, M. T.; Oberli, “A multidisciplinary

nary approach for monitoring flood and landslide hazards: Application to the quebrada de ramón watershed in central chile.” Unpublished, dic 2016.

- [16] Emol.com, “Nuevo proyecto determinaría el riesgo de crecidas y aluviones en quebrada de ramón.” [Online; posted 2015].
- [17] IAGua, “Los humedales: grandes descontaminantes de los metales pesados de la minería.” [Online; posted 2017].
- [18] S. de la Convención de Ramsar, ed., *Manual de la Convención de Ramsar*, vol. 4 of *Guía a la Convención sobre los Humedales*. Ramsar, Irán: Ramsar, 1971.
- [19] S. Sánchez-Carrillo, D. Angeler, M. Álvarez-Cobelas, and R. Sánchez-Andrés, “Freshwater wetland eutrophication,” in *Eutrophication: causes, consequences and control*, pp. 195–210, Springer Netherlands, 2010.
- [20] Tripleenlace, “Eutrofización: causas y efectos.” [Online; posted 2012].
- [21] A. Davranche, G. Lefebvre, and B. Poulin, “Wetland monitoring using classification trees and SPOT-5 seasonal time series,” *Remote Sensing of Environment*, vol. 114, pp. 552–562, mar 2010.
- [22] E. R. D. Roeck, N. E. Verhoest, M. H. Miya, H. Lievens, O. Batelaan, A. Thomas, and L. Brendonck, “Remote sensing and wetland ecology: a south african case study,” *Sensors*, vol. 8, pp. 3542–3556, may 2008.
- [23] C. de Medio Ambiente y Ordenación del Territorio España, “Calidad de las aguas del río guadaíra.” [Online (link); posted 2011].
- [24] U. de Puerto Rico Recinto de Mayaguez, ed., *Parámetros Físico-Químicos: pH*.
- [25] S. Mandiola, “Humedales de la ciudad de llanquihue, parte del pam.” Unpublished, jun 2017.
- [26] M. T. Lazarescu, “Design of a WSN platform for long-term environmental monitoring for IoT applications,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, pp. 45–54, mar 2013.
- [27] J. Petajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “On the coverage of LPWANS: range evaluation and channel attenuation model for LoRa technology,” in *2015 14th International Conference on ITS Telecommunications (ITST)*, IEEE, dec 2015.
- [28] X. Yang, K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and C. A. Grimes, “Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment,” *Sensors*, vol. 2, pp. 455–472, nov 2002.
- [29] M. Aref and A. Sikora, “Free space range measurements with semtech lora technology,” in *2014 2nd International Symposium on Wireless Systems within the Con-*

ferences on Intelligent Data Acquisition and Advanced Computing Systems, IEEE, sep 2014.

- [30] D. A. J. Ballart, “Estudio de cobertura de la modulación lora en la banda de 915[mhz] en ambientes urbanos.” Unpublished, dic 2017.

Apéndice A

Códigos

A.1. Código Gateway - Arduino

```
1 /*****
2 *
3 * Matias Macaya L.
4 * keny92@gmail.com
5 * github.com/matimacayal
6 *
7 * Arduino sketch for the lora receiver of the GW for llanquihue's
8 * smart-wetland monitoring platform.
9 *
10 * The gateway its composed of (1) a Rasperry Pi 3 saving local data
11 * and uploading it to the cloud, plus other services. And (2) an
12 * arduino serving as lora receiver and passing communication data to
13 * the raspi.
14 *
15 * Conexiones -----
16 *
17 *
18 *          |
19 *          v
20 *
21 *          (Arduino) Teensy      inAir4 inAir9
22 *          GND-----0V      (ground in)
23 *          3V3-----3.3V    (3.3V in)
24 *          interrupt 0 pin D2-----D0      (interrupt request out)
25 *          SS pin D10-----CS      (CS chip select in)
26 *          SCK pin D13-----CK      (SPI clock in)
27 *          MOSI pin D11-----SI      (SPI Data in)
28 *          MISO pin D12-----SO      (SPI Data out)
29 *          reset pin D9-----RST      (RST)
30 *
31 * Comandos "at" disponibles:
32 *
33 * Command      Function      Params      Syntax (
34 * always end's with '+' )
35 *
36 * at          check if ok      -          at+
37 * begin      start listening    -          begin+
38 * stop      stop receiving      -          stop+
39 * send      send lora packet    Message    send+decir
40 * adios es crecer+
```

```

34 *   config-mod           configure lora mod           Mod           config-mod
   +5+
35 *   config-params       configure lora params       BW, CR, SF       config-
   params+125+4/5+7+
36 *   sleep               enter sleep mode           -           sleep+
37 *   config-tx-power     configure tx power           Tx Power       config-tx-
   power+13+
38 *
39 *****/
40 // For PlatformIO
41 #include <Arduino.h>
42
43 // SX1276 libraries
44 #include <SPI.h>
45 #include <RH_RF95.h>
46
47 #define RFM95_CS 10
48 #define RFM95_RST 9
49 #define RFM95_INT 2
50
51 #define RF95_FREQ 915.0
52
53 RH_RF95 rf95(RFM95_CS, RFM95_INT);
54
55 // Second serial library
56 #include <DHT.h>
57
58 // Various vars
59 bool newCmd = false;
60 String cmd;
61
62 bool listening = true;
63
64 int packetNumber;
65 int lastRSSI = 0;
66 int lastPacketNumber = 0;
67 int lastPacketSentDuration = 0;
68
69 const String LORA_TAG = "lora\t";
70 const String INFO_TAG = "info\t";
71
72 void setup() {
73   Serial.begin(9600); delay(100);
74
75   pinMode(RFM95_RST, OUTPUT);
76   digitalWrite(RFM95_RST, HIGH); delay(100);
77
78   digitalWrite(RFM95_RST, LOW); delay(10);
79   digitalWrite(RFM95_RST, HIGH); delay(10);
80
81   String error = "ok";
82
83   if (error == "ok" && !rf95.init())           error = "LoRa
   radio initialization failed";
84   if (error == "ok" && !rf95.setFrequency(RF95_FREQ)) error = "Frequency
   configuration failed";

```

```

85  if (error == "ok" && !rf95.setModemConfig(RH_RF95::mod9)) error = "LoRa mod
    configuration error";
86
87  if (error != "ok") {
88      Serial.print(INFO_TAG);
89      Serial.println("Error: " + error);
90      while(1);
91  }
92
93  Serial.print(INFO_TAG);
94  Serial.println("LoRa is ready!");
95  Serial.print(INFO_TAG);
96  Serial.println("Freq: " + String(RF95_FREQ) + "MHz");
97  }
98
99  void serialEvent() {
100     cmd = Serial.readStringUntil('+');
101     newCmd = true;
102  }
103
104  void configureLoraMod(int modmod) {
105     switch(modmod) {
106         case 0: { rf95.setModemConfig(RH_RF95::mod0); break;} // <BW,      CR,  SF>
107         case 1: { rf95.setModemConfig(RH_RF95::mod1); break;} // 125kHz, 4/5, 12
108         case 2: { rf95.setModemConfig(RH_RF95::mod2); break;} // 250kHz, 4/5, 12
109         case 3: { rf95.setModemConfig(RH_RF95::mod3); break;} // 125kHz, 4/5, 10
110         case 4: { rf95.setModemConfig(RH_RF95::mod4); break;} // 500kHz, 4/5, 12
111         case 5: { rf95.setModemConfig(RH_RF95::mod5); break;} // 250kHz, 4/5, 10
112         case 6: { rf95.setModemConfig(RH_RF95::mod6); break;} // 500kHz, 4/5, 11
113         case 7: { rf95.setModemConfig(RH_RF95::mod7); break;} // 250kHz, 4/5, 9
114         case 8: { rf95.setModemConfig(RH_RF95::mod8); break;} // 500kHz, 4/5, 9
115         case 9: { rf95.setModemConfig(RH_RF95::mod9); break;} // 500kHz, 4/5, 8
116         case 10: { rf95.setModemConfig(RH_RF95::mod10); break;} //500kHz, 4/5, 7
117     }
118 }
119
120 void configureLoraParams(String bandWidth, String codingRate, String
    spreadingFactor) {
121     uint8_t bw, cr, sf;
122     bool invalid = false;
123
124     float n_bw = bandWidth.toFloat();
125     switch((int) n_bw*100) {
126         case 780: {bw = 0b00000000; break;} // 7.8 kHz
127         case 1040: {bw = 0b00010000; break;} // 10.4 kHz
128         case 1560: {bw = 0b00100000; break;} // 15.6 kHz
129         case 2080: {bw = 0b00110000; break;} // 20.8 kHz
130         case 3125: {bw = 0b01000000; break;} // 31.25 kHz
131         case 4170: {bw = 0b01010000; break;} // 41.7 kHz
132         case 6250: {bw = 0b01100000; break;} // 62.5 kHz
133         case 12500: {bw = 0b01110000; break;} // 125 kHz
134         case 25000: {bw = 0b10000000; break;} // 250 kHz
135         case 50000: {bw = 0b10010000; break;} // 500 kHz
136         default: {invalid = true; break;}
137     }
138 }

```

```

139 char n_cr = codingRate.charAt(2);
140 switch(n_cr) { // CR
141     case '5': {cr = 0b00000010; break;} // 4/5
142     case '6': {cr = 0b00000100; break;} // 4/6
143     case '7': {cr = 0b00000110; break;} // 4/7
144     case '8': {cr = 0b00001000; break;} // 4/8
145     default: {invalid = true; break;}
146 }
147
148 int n_sf = spreadingFactor.toInt();
149 switch(n_sf) { // SF
150     case 6: {sf = 0b01100100; break;} // 6
151     case 7: {sf = 0b01110100; break;} // 7
152     case 8: {sf = 0b10000100; break;} // 8
153     case 9: {sf = 0b10010100; break;} // 9
154     case 10: {sf = 0b10100100; break;} // 10
155     case 11: {sf = 0b10110100; break;} // 11
156     case 12: {sf = 0b11000100; break;} // 12
157     default: {invalid = true; break;}
158 }
159
160 if (!invalid) {
161     uint8_t register_1D = bw | cr;
162     uint8_t register_1E = sf;
163
164     rf95.configureLoraRegs(register_1D, register_1E);
165 } else {
166     Serial.print(INFO_TAG);
167     Serial.println("Error: Invalid lora params");
168     return;
169 }
170 }
171
172 void sendLoraPacket(String msg) {
173     uint8_t len = msg.length() + 1;
174     char payload[len];
175     msg.toCharArray(payload, len);
176
177     unsigned long time1 = micros();
178
179     rf95.send((uint8_t *)payload, len);
180     delay(10);
181     rf95.waitPacketSent();
182
183     unsigned long time2 = micros();
184     lastPacketSentDuration = time2 - time1;
185 }
186
187 void executeCommand(String command) {
188     int n_command = 0;
189     if (command == "at") {n_command = 1;}
190     else if (command == "begin") {n_command = 2;}
191     else if (command == "stop") {n_command = 3;}
192     else if (command == "send") {n_command = 4;}
193     else if (command == "config-mod") {n_command = 5;}
194     else if (command == "config-params") {n_command = 6;}

```

```

195 else if (command == "sleep") {n_command = 7;}
196 else if (command == "config-tx-pow") {n_command = 8;}
197
198 switch(n_command) {
199     case 1: // at
200     {
201         Serial.print(INFO_TAG);
202         Serial.println("OK");
203         break;
204     }
205     case 2: // begin
206     {
207         rf95.setModeRx();
208         listening = true;
209         Serial.print(INFO_TAG);
210         Serial.println("OK, Rx mode");
211         break;
212     }
213     case 3: // stop
214     {
215         rf95.setModeIdle(); // stdby mode, msgs should not be received
216         listening = false;
217         Serial.print(INFO_TAG);
218         Serial.println("OK, Stdby mode");
219         break;
220     }
221     case 4: // send
222     {
223         String message = Serial.readStringUntil('+');
224         sendLoraPacket(message);
225         Serial.print(INFO_TAG);
226         Serial.println("OK, '" + message + "' sent");
227         break;
228     }
229     case 5: // config-mod
230     {
231         int mod = Serial.readStringUntil('+').toInt();
232         configureLoraMod(mod);
233         Serial.print(INFO_TAG);
234         Serial.println("OK, mod " + String(mod));
235         break;
236     }
237     case 6: // config-params
238     {
239         String bw = Serial.readStringUntil('+');
240         String cr = Serial.readStringUntil('+');
241         String sf = Serial.readStringUntil('+');
242         configureLoraParams(bw, cr, sf);
243         Serial.print(INFO_TAG);
244         Serial.println("OK, <" + bw + ", " + cr + ", " + sf + ">");
245         break;
246     }
247     case 7: // sleep
248     {
249         // TODO: Averiguar m s sobre el modo 'sleep' del sx1276, ya que parece
           existe

```



```

250 // un tiempo máximo luego del cual se despierta automáticamente para
    escuchar.
251 rf95.sleep();
252 Serial.print(INFO_TAG);
253 Serial.println("OK, sleep mode");
254 break;
255 }
256 case 8: // config-tx-pow
257 {
258 // Se asume que ocupamos boost mode
259 String power = Serial.readStringUntil('+');
260 rf95.setTxPower(power.toInt(), false);
261 Serial.print(INFO_TAG);
262 Serial.println("OK, tx power " + power + "dBm");
263 break;
264 }
265 default:
266 {
267 Serial.print(INFO_TAG);
268 Serial.println("ERROR, invalid cmd");
269 break;
270 }
271 }
272
273 // flush serial buffer
274 while(Serial.available()) {
275 Serial.read();
276 }
277
278 newCmd = false;
279 cmd = "";
280 }
281
282 void printPreviousPacketsNotReceived(uint32_t n) {
283 // Print previous packets not received
284 if (lastPacketNumber != 0) {
285 for (uint32_t i=lastPacketNumber+1; i<n; i++) {
286 Serial.print(LORA_TAG);
287 Serial.print(""); Serial.print("\tRx\t"); Serial.println(i);
288 }
289 }
290 }
291
292 void loop() {
293 if (newCmd) {
294 executeCommand(cmd);
295 }
296
297 if (rf95.available() && listening) {
298 uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
299 uint8_t len = sizeof(buf);
300
301 if (rf95.recv(buf, &len)) {
302 // Check & Print packets not received
303 uint8_t c;
304 for (uint32_t i=0; i<sizeof(buf); i++) {

```

```

305     if ((c = buf[i]) == '\t')
306         break;
307     packetNumber = packetNumber * 10 + (c-48);
308 }
309
310 if (packetNumber != lastPacketNumber + 1) {
311     printPreviousPacketsNotReceived(packetNumber);
312 }
313 lastPacketNumber = packetNumber;
314 packetNumber = 0;
315
316 lastRSSI = rf95.lastRssi();
317 Serial.print(LORA_TAG);
318 Serial.print(
319     String(millis()) + "\t" +
320     "Rx" + "\t" +
321     String(lastPacketNumber) + "\t" +
322     "payload" + "\t" +
323     "<" + "\t" +
324     String((char*) buf) + "\t" +
325     ">" + "\t" +
326     "rssi" + "\t" +
327     String(lastRSSI) + "\n"
328 );
329
330 // Send a reply
331 // uint8_t data[] = "And hello back to you";
332 // rf95.send(data, sizeof(data));
333 // rf95.waitPacketSent();
334 // Serial.println("Sent a reply");
335 // digitalWrite(LED, LOW);
336 } else {
337     Serial.println("\tfailed");
338 }
339 }
340 }

```

A.2. Código Nodo Monitoreo

```

1 // Código de Transmision Arduino + inAir9 (SX1276)
2 // Tx
3 //
4 // Conexiones -----
5 //
6 //
7 //           Arduino           inAir4 inAir9
8 //           GND-----0V    (ground in)
9 //           3V3-----3.3V (3.3V in)
10 // interrupt 0 pin D2-----D0    (interrupt request out)
11 //           SS pin D10-----CS    (CS chip select in)
12 //           SCK pin D13-----CK    (SPI clock in)
13 //           MOSI pin D11-----SI    (SPI Data in)
14 //           MISO pin D12-----SO    (SPI Data out)
15 //           reset pin D9-----RST (RST)

```

```

16 //
17 //
18 //           DHT22
19 //           GND—————NEG (ground in)
20 //           3V3—————POS (3.3V in)
21 //           pin D5—————OUT (data out)
22 //
23 //           photo-resistor
24 //           GND—————resistor (ground in)
25 //           3V3—————photo-resistor (3.3V in)
26 //           pin A2—————resistor + photo-resistor (data out)
27 //
28 const bool VERBOSE = false;
29
30 void verbosePrint(String msgOut) {
31     if (VERBOSE)
32         Serial.print(msgOut);
33 }
34
35 void verbosePrintln(String msgOut) {
36     if (VERBOSE)
37         Serial.println(msgOut);
38 }
39
40 /* sx1276*/
41 #include <SPI.h>
42 #include <RH_RF95.h>
43
44 #define RFM95_INT 2
45 #define RFM95_CS 10
46 #define RFM95_RST 9
47 #define RF95_FREQ 915.0
48
49 RH_RF95 rf95(RFM95_CS, RFM95_INT);
50
51 //char payload[23]="Decir adios es crecer ";
52 String msg = "Decir adios es crecer";
53 String aux;
54 unsigned long n = 0;
55
56 /* sensors */
57 #define BUTTON_PIN 3
58 #define LED_PIN 7
59 unsigned long lastDebounceTime = 0;
60 unsigned long debounceDelay = 100;
61
62 bool virtual_channel_jamming_off = true;
63
64 #include "DHT.h"
65 #define DHT_PIN 5 // what digital pin we're connected to
66 #define DHT_TYPE DHT22 // DHT 22 (AM2302), AM2321
67 DHT dht(DHT_PIN, DHT_TYPE);
68
69 #define LIGHT_PIN A2
70
71 float hum, temp;

```

```

72 int light;
73
74 #include <Wire.h>
75 #define PH_ADDRESS 99
76 #define EC_ADDRESS 100
77 #define DO_ADDRESS 97
78
79 uint8_t trial = 0;
80 char response_code = 0;
81 char sensor_data[20];
82 char in_char = 0;
83 uint8_t i = 0;
84 uint16_t acquisition_time = 900;
85
86 float ph_float;
87 float do_float;
88 float ec_float;
89
90 float read_value(uint8_t sensor);
91
92 unsigned long time1;
93 unsigned long time2;
94 String tab = String("\t");
95
96 // Various vars
97 bool newCmd = false;
98 String cmd;
99
100 bool transmit = true;
101
102 void setup() {
103     Serial.begin(9600);
104     Wire.begin();
105
106     pinMode(LED_PIN, OUTPUT);
107     digitalWrite(LED_PIN, virtual_channel_jamming_off);
108     pinMode(BUTTON_PIN, INPUT_PULLUP);
109     //attachInterrupt(1, onClick, FALLING);
110
111     pinMode(LIGHT_PIN, INPUT);
112     pinMode(DHT_PIN, INPUT);
113     dht.begin();
114
115     pinMode(RFM95_RST, OUTPUT);
116     digitalWrite(RFM95_RST, HIGH);
117
118     // inAir9 manual reset
119     digitalWrite(RFM95_RST, LOW); delay(10);
120     digitalWrite(RFM95_RST, HIGH); delay(10);
121
122     String error = "ok";
123
124     // Configuring 915 MHz band
125     // defaults config <434.0MHz, modulation GFSK_Rb250Fd250, +13dbm>
126     // Configuring Tx power
127     // rf95.setTxPower(10)      -> use PA_BOOST transmitter pin. +5 to +23 (

```

```

128   for modules that use PA_BOOST)          <- INAIR9B.
//   rf95.setTxPower(10, true) -> use PA_RFO transmitter pin. -1 to +14 (
//   for modules that use RFO transmitter pin) <- INAIR9.
129
130   if (error == "ok" && !rf95.init())          error = "LoRa
radio initialization failed";
131   if (error == "ok" && !rf95.setFrequency(RF95_FREQ))          error = "Frequency
configuration failed";
132   if (error == "ok" && !rf95.setModemConfig(RH_RF95::mod10)) error = "LoRa mod
configuration error";
133   rf95.setTxPower(1, true);
134
135   if (error != "ok") {
136     verbosePrintln("Error: " + error);
137     while(1);
138   }
139
140   verbosePrintln("LoRa is ready!");
141   verbosePrintln("Freq: " + String(RF95_FREQ) + "MHz");
142   verbosePrintln("Tx Pow: " + String(10) + "dBm");
143
144   verbosePrintln("Begining Tx...\n");
145 }
146
147 void serialEvent() {
148   cmd = Serial.readStringUntil('+');
149   newCmd = true;
150 }
151
152 void onClick() {
153   if ((millis() - lastDebounceTime) > debounceDelay) {
154     virtual_channel_jamming_off = !virtual_channel_jamming_off;
155     digitalWrite(LED_PIN, virtual_channel_jamming_off);
156   }
157
158   lastDebounceTime = millis();
159 }
160
161 void executeCommand(String command) {
162   int n_command = 0;
163   if (command == "start") {n_command = 1;}
164   else if (command == "stop") {n_command = 2;}
165
166   switch(n_command) {
167     case 1: // start
168     {
169       String boost = Serial.readStringUntil('+');
170       String power = Serial.readStringUntil('+');
171       String mod = Serial.readStringUntil('+');
172
173       if (boost.toInt() == 1) rf95.setTxPower(power.toInt());
174       else rf95.setTxPower(power.toInt(), true); // 2nd arg
'useRFO'
175
176       switch(mod.toInt()){
177         case 0: { rf95.setModemConfig(RH_RF95::mod0); break;}

```

```

178     case 1: { rf95.setModemConfig(RH_RF95::mod1); break;}
179     case 2: { rf95.setModemConfig(RH_RF95::mod2); break;}
180     case 3: { rf95.setModemConfig(RH_RF95::mod3); break;}
181     case 4: { rf95.setModemConfig(RH_RF95::mod4); break;}
182     case 5: { rf95.setModemConfig(RH_RF95::mod5); break;}
183     case 6: { rf95.setModemConfig(RH_RF95::mod6); break;}
184     case 7: { rf95.setModemConfig(RH_RF95::mod7); break;}
185     case 8: { rf95.setModemConfig(RH_RF95::mod8); break;}
186     case 9: { rf95.setModemConfig(RH_RF95::mod9); break;}
187     case 10: { rf95.setModemConfig(RH_RF95::mod10); break;}
188 }
189
190 String bost;
191 if (boost.toInt() == 1) bost = "boost";
192 else bost = "rf0";
193
194 verbosePrintln("OK, lora -> " + bost + ", " + power + "dBm, mod" + mod);
195
196 transmit = true;
197 verbosePrintln("Transmitting...");
198
199 break;
200 }
201 case 2: // stop
202 {
203     transmit=false;
204     verbosePrintln("stopped transmitting.");
205     break;
206 }
207 default:
208 {
209     verbosePrintln("ERROR, invalid cmd");
210     break;
211 }
212 }
213
214 // flush serial buffer
215 while(Serial.available()) {
216     Serial.read();
217 }
218
219 newCmd = false;
220 cmd = "";
221 }
222
223 void loop() {
224     if (newCmd) {
225         executeCommand(cmd);
226     }
227
228     if (transmit) {
229         //light = analogRead(LIGHT_PIN);
230         hum = dht.readHumidity();
231         temp = dht.readTemperature();trn
232
233         do_float = read_value(DO_ADDRESS);

```

```

234   ph_float = read_value(PH_ADDRESS);
235   //ec_float = read_value(EC_ADDRESS);
236
237   //msg.concat(String(n));
238   aux = String(n) + tab +
239         String(hum) + tab +
240         String(temp) + tab +
241         String(ph_float) + tab +
242         String(do_float) + tab +
243         msg;
244   uint8_t len = aux.length() + 1;
245   char payload[len];
246   aux.toCharArray(payload, len);
247   n++;
248
249   verbosePrint(String(millis()) + "\tTx, payload: <\t" + String(payload));
250   time1 = micros();
251
252   if (virtual_channel_jamming_off) {
253     rf95.send((uint8_t *)payload, len);
254     delay(10);
255     rf95.waitPacketSent();
256
257     Serial.print(payload);
258     Serial.write(10);
259   }
260
261   time2 = micros();
262   verbosePrint("\t>, tx time:\t" + String(time2 - time1) + "\tus");
263   delay(5000);
264 }
265 }
266
267 float read_value(uint8_t sensor) {
268   for (int i=0; i<20; i++) {
269     sensor_data[i] = 0;
270   }
271
272   do {
273     Wire.beginTransaction(sensor);
274     Wire.write("R");
275     Wire.endTransmission();
276
277     delay(acquisition_time);
278
279     Wire.requestFrom(sensor, 20, 1);
280     response_code = Wire.read();
281
282     while (Wire.available()) {
283       in_char = Wire.read();
284       sensor_data[i] = in_char;
285       i++;
286       if (in_char == 0) {
287         i = 0;
288         Wire.endTransmission();
289         break;

```

```
290     }
291 }
292
293     trial++;          // If it fails to read a value
294     if (trial == 5) { // it tries five more times
295         trial = 0;
296         break;
297     }
298
299 } while (response_code != 1);
300
301 return atof(sensor_data);
302 }
```