



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

## MAESTRÍA EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

### Dispositivo de medición de almacenamiento de granos con comunicación LoRa

Autor:

**Mg. Ing. Franco A. Bucafusco**

Director:

Mg. Ing. Diego Brengi (INTI, UNLaM, FIUBA)

Jurados:

Mg. Ing. Nicolás Dassieu (FIUBA)  
Ing. Juan Manuel Cruz (FIUBA, UTN-FRBA)  
Mg. Ing. Ramiro Alonso (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires,  
entre abril de 2019 y abril de 2020.*



## *Resumen*

En la presente memoria se describe el desarrollo de una solución para la metrología de parámetros en almacenamiento en silos tradicionales o en barcazas de transporte realizado para la empresa WiAgro. El dispositivo cuenta con una red cableada de subnodos en los cuales se ubican los sensores de temperatura y humedad.

Se desarrolló un cliente para una red LoRaWAN que permite generar un mapa de temperatura y humedad del volumen total del almacenaje. Se diseñaron dos dispositivos, uno para el cliente LoRa, y otro para los subnodos de la red cableada. En ambos casos se consideraron lineamientos de normativas internacionales para equipamiento eléctrico en ambientes explosivos.

El firmware desarrollado correspondiente a cada dispositivo emplea el framework mBed, para el cliente LoRa, y técnicas baremetal para el de los subnodos. La biblioteca de código de red de subnodos fue implementada utilizando técnicas de desarrollo basadas en pruebas, utilizando pruebas unitarias y de sistema.



# Índice general

<b>Resumen</b>	<b>III</b>
<b>1. Introducción General</b>	<b>1</b>
1.1. Sistemas de almacenamiento de granos . . . . .	1
1.2. Descripción del trabajo realizado . . . . .	4
1.3. Objetivos y alcance . . . . .	5
<b>2. Introducción Específica</b>	<b>7</b>
2.1. Requerimientos . . . . .	7
2.2. Planificación . . . . .	9
2.2.1. Desglose de tareas . . . . .	9
2.3. Sensor de temperatura y sensor de humedad . . . . .	11
2.4. LoRa y LoRaWAN . . . . .	12
2.4.1. Modos de operación . . . . .	14
2.4.2. Capa de seguridad . . . . .	15
2.5. Normativas sobre atmósferas explosivas . . . . .	16
2.5.1. Marcado . . . . .	17
2.5.2. Consideraciones de diseño . . . . .	18
<b>3. Diseño e Implementación</b>	<b>25</b>
3.1. Bloques constitutivos del sistema . . . . .	25
3.2. Protocolo de red de subnodos . . . . .	26
3.3. Diseño de hardware . . . . .	27
3.3.1. Gabinetes . . . . .	27
3.3.2. Cliente LoRa . . . . .	29
3.3.3. Subnodos . . . . .	45
3.4. Diseño de firmware . . . . .	47
3.4.1. Bibliotecas comunes entre ambos firmware . . . . .	47
3.4.2. Firmware del cliente . . . . .	50
3.4.3. Diseño de protocolo montado sobre LoRA servidor-cliente .	52
3.4.4. Firmware de los subnodos . . . . .	58
<b>4. Ensayos y Resultados</b>	<b>61</b>
4.1. Hardware . . . . .	61
4.1.1. Mediciones . . . . .	64
4.2. Master Test Plan . . . . .	66
4.2.1. Pruebas unitarias . . . . .	67
4.2.2. Pruebas funcionales . . . . .	73
4.3. Pruebas de integración . . . . .	75
<b>5. Conclusiones</b>	<b>79</b>
5.1. Objetivos alcanzados . . . . .	79
5.2. Próximos pasos . . . . .	80

5.2.1. Asesoría en relación a normativa de atmósferas explosivas . . . . .	80
5.2.2. Diseño mecánico de la columna de sensores . . . . .	80
5.2.3. Miniaturización y disminución del costo de los subnodos . . . . .	80
5.2.4. Realizar evaluaciones con más nodos . . . . .	80
<b>Bibliografía</b>	<b>83</b>

# Índice de figuras

1.1. Índice TAS para la soja . . . . .	2
1.2. Sistema tradicional de medición en silos . . . . .	3
1.3. Diagrama general del arreglo de sensores . . . . .	4
2.1. Diagrama de <i>Activity On Node</i> . . . . .	11
2.2. Circuito integrado utilizado para medir humedad y temperatura . . . . .	11
2.3. Capas de abstracción del protocolo de red . . . . .	13
2.4. Arquitectura de una red LoRaWAN . . . . .	14
2.5. Descomposición de la trama del stack LoRaWAN . . . . .	15
2.6. Estándares y normativas internacionales sobre peligrosidad . . . . .	17
2.7. Esquema de una barrera zener básica . . . . .	20
2.8. Concepto de líneas de fuga y distancias en aire . . . . .	22
2.9. Técnicas para aumentar las líneas de fuga en un circuito impreso . . . . .	23
3.1. Diagrama en bloques del sistema. . . . .	25
3.2. Diagrama de interconexión de la red de subnodos. . . . .	27
3.3. Gabinete elegido para alojar al cliente LoRaWAN . . . . .	28
3.4. Gabinete elegido para alojar al circuito impreso de cada subnodo . . . . .	28
3.5. Render del cliente LoRaWAN . . . . .	29
3.6. Recursos externos disponibles del módulo xDot . . . . .	30
3.7. Diagrama en bloques de la arquitectura de segmentación de zonas	33
3.8. Circuito asociado al regulador linear . . . . .	35
3.9. Circuito asociado al regulador conmutado . . . . .	35
3.10. Circuito de barrera zener implementado . . . . .	37
3.11. Curvas asociadas al diodo zener y al fusible elegido . . . . .	38
3.12. Diseño físico de la barrera zener . . . . .	39
3.13. Modelo del cálculo para la distorsión introducida por la barrera zener . . . . .	40
3.14. Gráfico que determina la inductancia máxima del sistema . . . . .	41
3.15. Adaptador UFL a SMA . . . . .	41
3.16. Diseño electrónico de la interfaz aislada RS 485. . . . .	42
3.17. Perfil que muestra las distancias en un circuito impreso . . . . .	43
3.18. Detalle del diseño del circuito impreso de la interfaz RS485 . . . . .	44
3.19. Distancias en aire para el conjunto de conectores CN2 y CN3 . . . . .	44
3.20. Render del pcb ensamblado de un subnodo . . . . .	45
3.21. Diagrama en bloques de un subnodo . . . . .	46
3.22. Formato de paquete para los comandos utilizados en el protocolo . . . . .	48
3.23. Estructura de comando y respuesta “SenseAll” . . . . .	49
3.24. Diagrama de secuencia del protocolo de subnodos . . . . .	50
3.25. Diagrama de flujos de principal del cliente LoRaWAN . . . . .	52
3.26. Estructura del mensaje/comando genérico . . . . .	53
3.27. Estructura del mensaje de mediciones . . . . .	54
3.28. Estructura de los comandos para establecer parámetros . . . . .	55

3.29. Estructura de las tramas de “Aregar” datos y de “Finalizar” . . . . .	56
3.30. Diagrama de secuencia del algoritmo de firmware update . . . . .	57
3.31. Diagrama de flujos principal del firmware de los subnodos . . . . .	58
3.32. Capas de abstracción del firmware de los subnodos. . . . .	59
4.1. Comparación del prototipo fabricado del nodo maestro versus el render 3D diseñado . . . . .	62
4.2. Comparación del prototipo fabricado del subnodo versus el render 3D diseñado . . . . .	63
4.3. Circuito equivalente para medir la transmisión no ideal de la USART	64
4.4. Diagrama de ojo para señales recibidas en un subnodo . . . . .	65
4.5. Tiempos de crecimiento y de caída de los flancos de la USART . . . . .	65
4.6. Resultado de la pruebas unitarias del driver para el STH31 . . . . .	68
4.7. Resultado de la prueba de cobertura del driver para el STH31 . . . . .	68
4.8. Resultado de la pruebas unitarias de las funciones de interpretación de comandos de configuración . . . . .	71
4.9. Resultado de la prueba de cobertura del módulo de interpretación de comandos de la aplicación . . . . .	71
4.10. Resultado de la pruebas unitarias de las funciones de interpretación de comandos de actualización de firmware . . . . .	72
4.11. Resultado de la prueba de cobertura del módulo de interpretación de comandos de actualización de firmware. . . . .	73
4.12. Pruebas del módulo de red de subnodos . . . . .	74
4.13. Pruebas del módulo de red de subnodos . . . . .	74
4.14. Banco de prueba para realizar el trabajo . . . . .	76
4.15. Kits de desarrollo emulando un arreglo de dos subnodos . . . . .	76
4.16. Trafico de paquetes en el servidor de red TTN . . . . .	77
4.17. Trafico de paquetes en el servidor de aplicación TTN . . . . .	78

# Índice de Tablas

1.1.	Productos existentes para metrología en silos tradicionales . . . . .	3
2.1.	Requerimientos acotados . . . . .	9
2.2.	Desglose de tareas . . . . .	10
2.3.	Versiones de STH31 . . . . .	12
2.4.	Requisitos de diseño de los componentes que brindan protección .	21
3.1.	Ideas descartadas para red de subnodos . . . . .	27
3.2.	Patillaje disponible del módulo xDot . . . . .	31
3.3.	Consumos de corriente del módulo xDot para VDD=3,3 V . . . . .	32
3.4.	Cálculo de máxima corriente que puede entregar el regulador de conmutación en función de la tensión de las baterías . . . . .	36
3.5.	Resultados de los cálculos de componentes para barrera zener . . . .	38
3.6.	Requerimientos generales del protocolo de red de subnodos . . . .	48
3.7.	Comandos para el protocolo de aplicación . . . . .	53
3.8.	Subcomandos para comando de configuración de parámetros . .	54
4.1.	Pruebas unitarias realizadas sobre el driver del sensor STH31 . .	67
4.2.	Pruebas unitarias realizadas sobre el protocolo de red de subnodos	69
4.3.	Pruebas unitarias realizadas sobre el protocolo de aplicación sobre LoRaWAN . . . . .	70
4.4.	Pruebas unitarias realizadas sobre los interpretes de comandos de actualización de firmware . . . . .	72
4.5.	Pruebas funcionales realizadas sobre el protocolo de red de subnodos	73



# Capítulo 1

## Introducción General

Este capítulo introduce al lector en la problemática relacionada con el almacenamiento de granos. Se analiza el estado de arte de los sistemas de monitoreo de estos, mencionando dispositivos equivalentes en el mercado. Además, se presentan los objetivos y el alcance del presente trabajo.

### 1.1. Sistemas de almacenamiento de granos

El monitoreo remoto de parámetros en tiempo real en ambientes agroindustriales es una necesidad. El factor limitante más usual es la escasez de enlaces de datos para vincular los sitios de medición con los servidores de datos y los centros de monitoreo.

Según el Manual de buenas prácticas del INTA [1], uno de los desafíos más grande que enfrenta en Argentina es el de "minimizar las pérdidas de la calidad de los granos durante la etapa de poscosecha".

Se espera que durante los próximos treinta años la población mundial crezca en un tercio del valor actual. Esto, sumado a que existen otros usos industriales para los granos (por ejemplo, la fabricación de biocombustibles), requerirá aumentar la producción considerablemente. Las proyecciones de crecimiento de la demanda muestran que para alimentar a la población mundial en 2050 se requerirá aumentar la producción en un 70 % [2]. La demanda comercial de alimentos seguirá creciendo.

Para cubrir la demanda, se necesitará:

- Aumentar la producción a partir de siembras más extensas (lo que genera mayor deforestación).
- Mejorar la calidad del transporte del grano, evitando que el mismo se pierda en el proceso (por carrocería o contenedores en mal estado).
- Evitar la degradación del grano en la etapa de almacenamiento.
- Proteger al grano ante robos.

Durante el almacenamiento, la actividad biológica del granel (correspondiente a hongos del almacenamiento, insectos y los propios granos) va deteriorando la calidad del grano. Otro factor que mide la calidad es el porcentaje de granos dañados mecánicamente. Si el deterioro es importante, puede causar incluso la disminución del grado comercial.

En este sentido, se define como TAS (Tiempo de Almacenamiento Seguro) el período máximo que puede ser almacenado un grano a determinadas condiciones de humedad y temperatura, sin perder su condición de grado [1]. A mayor humedad se incrementa el crecimiento de hongos y a mayor temperatura se fomenta la aparición de insectos. Existen curvas en relación con estos dos parámetros, para cada tipo de grano, como la que se muestra en la figura 1.1, correspondiente a la soja. Por ejemplo, si se destina un cierto silo a conservar los granos por más de 90 días, pero la humedad relativa de los granos es del 17 % la temperatura no debería superar los 10 °C.

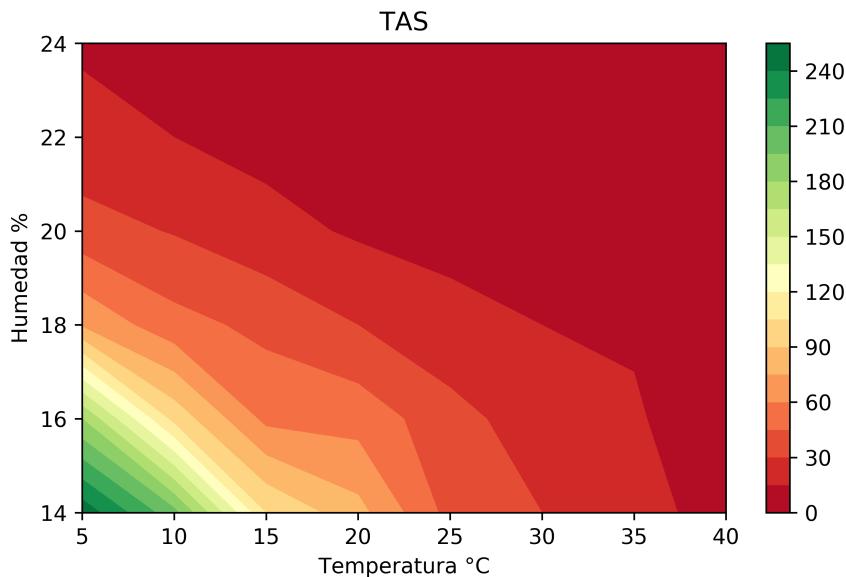


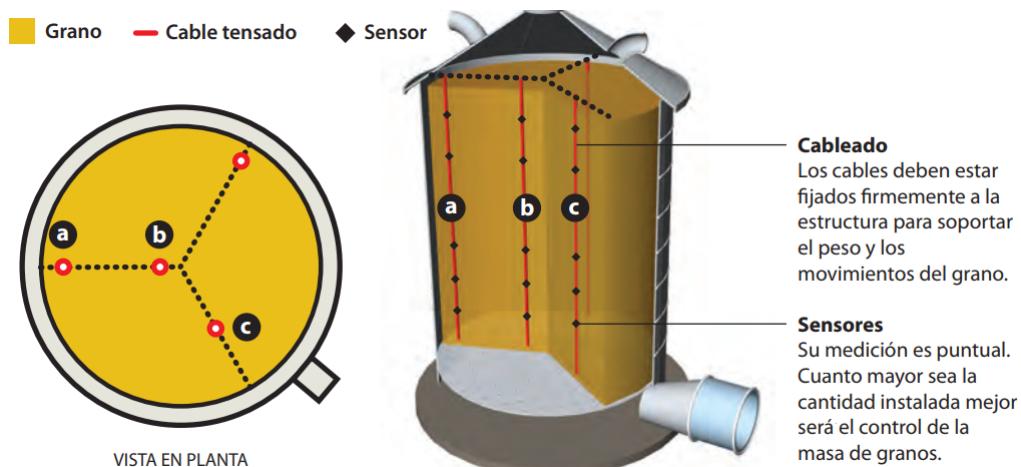
FIGURA 1.1: TAS para la soja. Cantidad de días en esas condiciones antes de perder el 0,5 % de la materia seca<sup>1</sup>.

En la cadena de valor, se encontró una necesidad no cubierta: la metrología de parámetros del grano durante el transporte fluvial, tanto para transporte ribereño como transoceánico. Durante el transporte, los granos se almacenan en grandes contenedores, parecidos a los silos tradicionales, y se cierran con una compuerta superior.

La medición de humedad permite conocer el contenido de agua de los granos y es crítica para tomar decisiones correctas respecto al tiempo de almacenamiento. Por otro lado, la medición de temperatura es fundamental para poder anticiparse a los problemas de deterioro del producto. El aumento de la temperatura del granel se relaciona directamente con el aumento de la actividad biológica. Estas mediciones deben hacerse en lugares estratégicos de los reservorios, tanto en superficie como en profundidad, particularmente en el centro y en la periferia.

Actualmente existen productos destinados a la metrología de parámetros en silos tradicionales, pensados para una instalación estática, donde se cuenta con una fuente de energía cercana. Estos productos poseen interfaces industriales cableadas, como muestra la figura 1.2. Cada columna de sensores posee un controlador maestro que a su vez se conecta por red a un concentrador central fuera del silo.

<sup>1</sup>Fuente: elaboración propia a partir de los datos provistos por [3]

FIGURA 1.2: Sistema tradicional de medición en silos<sup>2</sup>.

La principal falencia de los productos disponibles en el mercado es que no pueden ser utilizados en transportes fluviales o marítimos porque, en esas aplicaciones, no se posee una red cableada para la alimentación y la comunicación de datos.

Por otro lado, no todos los sistemas que se consiguen en el mercado poseen medición de humedad. En la tabla 1.1 se pueden observar productos similares que actualmente están disponibles para la instalación en silos tradicionales, comparándolos por tipo de medición que realizan, la región en donde se fabrican (AR para Argentina e INT para países extranjeros) y las certificaciones disponibles para cada producto.

TABLA 1.1: Productos existentes para metrología en silos tradicionales y las mediciones que realizan.

Marca y modelo	Temp.	Hum.	Certificación	Región	Ref
Proyecto académico	X	X		AR	[4]
TORNUM	X		ATEX zona 20/21/22	INT	[5]
TMS2000/2500					
Measure Instruments	X			AR	[6]
SGM-LEKTRA - SON-	X		ATEX IID Ex ta IIIC IP67 T92 C Da ATEX Zona 22	INT	[7]
DA TM					
FineTek EST110	X		IP67	INT	[8]
GrainWatch GWSL1100	X		Ex II 1D Ex ia IIIC T85C	INT	[9]

En resumen: habiendo detectado estas falencias, la motivación principal del trabajo fue el de diseñar un sistema que, además de poseer las características generales de los que se encuentran en el mercado, incluya un enlace de datos inalámbrico y que agregue otras mediciones, como la de humedad.

<sup>2</sup>Fuente: [1]

## 1.2. Descripción del trabajo realizado

Durante estos últimos años ha habido una evolución considerable en tecnologías de telecomunicaciones a corta, media y larga distancia que, combinadas de manera adecuada, permiten brindar cobertura en áreas cada vez más remotas del planeta. Ciertamente, la combinación de sistemas debe adaptarse a la disponibilidad geográfica y al presupuesto del cliente.

El trabajo realizado consistió en el desarrollo de un sistema para medir un arreglo de sensores instalado en las bodegas de los transportes fluviales. El mismo cuenta con un transceptor de radio, para enviar los datos de las mediciones a un concentrador que permita enviar los datos por satélite o almacenarlos localmente. Dado que, conceptualmente, es similar a la solución para silos tradicionales, se agregó también al hardware una interfaz de comunicación de datos cableada. El dispositivo implementado está diseñado para ser alimentado eléctricamente con baterías.

Se eligió a LoRa [10] como protocolo inalámbrico de transferencia de datos debido a que es un sistema cuya infraestructura es flexible. Por ejemplo, permite al cliente utilizar puertas de enlace propias o contratar puertas de enlaces a otro proveedores (similar al servicio de telefonía celular). El modelo elegido es el primero: los datos se transfieren a un *gateway*, ubicado en la torre de control de las naves. Por otro lado, su bajo consumo de energía lo hace atractivo para que las baterías no se drenen tan rápido, más que nada en viajes de varios días.

Como muestra la figura 1.3, cada arreglo de sensores consiste en un elemento maestro con el hardware necesario para la comunicación LoRa, y hasta veinticinco elementos esclavos (subnodos) separados en un metro. Cada subnodo posee sensores de temperatura y humedad, aunque el diseño electrónico se dejó preparado para poder medir otras variables<sup>3</sup>.

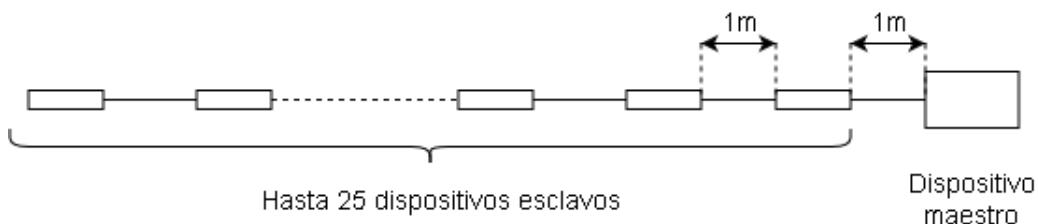


FIGURA 1.3: Diagrama general del arreglo de sensores.

En el capítulo 3 se darán los detalles de la implementación del hardware y del firmware.

---

<sup>3</sup>WiAgro se reserva el derecho de publicación de estos detalles.

### **1.3. Objetivos y alcance**

El presente trabajo tuvo como objetivo principal el diseño del hardware y el firmware del nodo maestro y de los subnodos. Para ello se desarrolló también el protocolo de red entre el maestro y los subnodos.

El alcance del proyecto incluyó:

- Desarrollo de circuito impreso del nodo maestro y circuito impreso de los subnodos.
- Desarrollo de firmware.
- Desarrollo de firmware de los subnodos.
- Diseño de protocolo de alto nivel para la configuración de dispositivo.

Se incluyeron también otros elementos en el alcance, como ser:

- Diseño e implementación del protocolo de actualización de firmware.
- Implementación de estrategias de bajo consumo.

El trabajo no incluyó:

- Desarrollo de gabinete(s)
- Desarrollo de software en servidor
- Consideraciones mecánicas de la columna de sensores.



## Capítulo 2

# Introducción Específica

Se abordan aquí los temas relacionados con la gestión y planificación del trabajo. Además, este capítulo introduce al lector a temáticas técnicas relacionadas con los sensores, con la conectividad y con las normativas de atmósferas explosivas.

### 2.1. Requerimientos

Con el avance del proyecto se encontraron algunas incoherencias en la especificación técnica y una carencia en los requerimientos originales [11] que generó que se modificaran los requerimientos ligeramente y que el avance del trabajo se retrasara.

El requerimiento más importante, que el cliente no contempló desde un principio, fue que el producto debía ser certificado para atmósferas explosivas. Esto ocasionó un retraso considerable, debido a que hubo que replantear el hardware en varios aspectos. En el capítulo 3 se explicarán los detalles de la implementación y de los elementos extra que se tuvieron en cuenta en el diseño.

Los requerimientos considerados en el trabajo fueron:

1. Requerimientos de documentación.
  - a) Se deberá redactar una memoria técnica relacionada con el desarrollo del dispositivo.
  - b) Se deberá redactar un documento de casos de prueba.
2. Requerimientos de hardware.
  - a) Generales:
    - 1) El dispositivo deberá seguir los lineamientos de la normativa IEC 60079 con respecto a atmósferas explosivas con polvo.
    - 2) La red de subnodos deberá ser de hasta veinticinco elementos.
  - b) Requerimientos de hardware del circuito impreso del dispositivo maestro:
    - 1) Se utilizará un módulo xDOT de Multitech, que integra un microcontrolador STM32L151CCU6 y un transceptor LoRa.
    - 2) Gestión de energía: se deberá contar con una entrada principal de alimentación para dos celdas de baterías de Li-FeS2.

- 3) Gestión de energía: se debería disponer de llaves de conmutación para la alimentación que permitan encender o apagar los sensores o la red de subnodos, bajo demanda.
  - 4) Memoria: se deberá contar con una memoria externa no volátil de 256 KB como mínimo.
  - 5) Deberá contar con un circuito para medición de nivel de batería
- c) Requerimientos de hardware del circuito impreso de los subnodos:
- 1) Deberá poseer un conector de entrada y otro de salida. Deberán ser del mismo tipo.
  - 2) Cada conector deberá tener las líneas de comunicación y de alimentación.
  - 3) Deberá tener un grupo de sensores de temperatura y humedad.
  - 4) Deberá contar con la electrónica necesaria extra a los sensores para lograr gestionar la comunicación por el bus de datos.
3. Requerimientos funcionales:
- a) Se deberán enviar reportes periódicos con las mediciones a través de la red LoRa.
  - b) Las mediciones deberán almacenarse en un sistema de archivos en eeprom interna.
  - c) Se deberá brindar un tiempo de muestreo configurable para cada sensor.
  - d) Se deberá brindar un tiempo de muestreo configurable para la medición de nivel de batería.
  - e) Se deberá dar la capacidad de filtrar las señales de cada sensor, de manera configurable (FIR / IIR)
  - f) Se deberá configurar un umbral para que el nivel de batería genere alarmas.
  - g) Durante los lapsos de reposo, el sistema deberá estar en el modo de menor consumo posible, apagando la mayor cantidad de periféricos.
  - h) La detección de un nivel de batería fuera del umbral deberá generar un reporte a través de la red.

Los requerimientos que se eliminaron o acotaron en alcance se resumen en la tabla 2.1.

TABLA 2.1: Requerimientos que sufrieron modificaciones durante el desarrollo.

Req.	Modificaciones
3b	No llegó a implementarse el sistema de archivos para almacenamiento de mediciones.
2c1	A partir del análisis que se hizo de la normativa, se concluyó que no se iban a utilizar conectores sino borneras.
3e	No se implementaron los filtrados dado que no tienen mayor sentido en una aplicación de tan baja frecuencia de muestreo y además el circuito integrado STH31 lo hace internamente.

## 2.2. Planificación

La planificación original presentada al iniciar el proyecto contemplaba un trabajo de 645 horas. Los atrasos relacionados con la problemática expuesta en la sección 2.1 obligaron a acotar algunas tareas, para llegar a tiempo con la entrega del presente trabajo. Se corrigió la planificación para entregar el informe de avance.

Al iniciar el proyecto, se pactó la entrega al cliente de los documentos listados a continuación:

1. Esquemáticos.
2. *Gerbers* para producción de los circuitos impresos.
3. Lista de materiales.
4. Documentación de protocolos.
5. Código fuente.
6. Informe final.

Estos estuvieron correlacionados con el desglose de tareas del proyecto, cuyos hitos más importantes fueron:

1. Definición de componentes de hardware.
2. Diseño de hardware.
3. Diseño del protocolo de red de subnodos.
4. Elección de *framework* para implementación de firmware en el dispositivo maestro.
5. Brindar de conectividad LoRaWAN al nodo maestro.
6. Elección de *framework* en la implementación de firmware para los subnodos.
7. Redacción de la documentación.

### 2.2.1. Desglose de tareas

El trabajo realizado al incluir hardware y firmware se organizó de tal forma que se tuviera tiempo para mandar a fabricar los prototipos y poder presentarlos al final de la maestría. Esto hizo que se trasladaran todas las actividades de diseño del hardware al principio.

El desglose de tareas mostrado en la tabla 2.2 está basado en el grupo de hitos antes mencionado y refleja el tiempo total de la planificación original.

TABLA 2.2: Desglose de tareas.

<b>WBS</b>	<b>Tarea</b>	<b>Tiempo [hs]</b>
1.	<b>Gestión</b>	
1.1	Planificación del proyecto	12
1.2	Redacción de informe de avance	5
1.3	Redacción de memoria	80
1.4	Redacción de presentación	15
2.	<b>Diseño del circuito impreso</b>	
2.1	Elección de sensor de temperatura	2
2.2	Elección de sensor de humedad	2
2.3	Estudio y selección de bus de comunicaciones para grupos de sensores	30
2.4	Estudio y selección de estrategia para extender el bus de sensores	20
2.5	Elección de memoria externa	2
2.6	Elección de llaves para conmutar la alimentación de energía de módulos	2
2.7	Evaluación y análisis de módulo xDot	20
2.8	Dibujo de esquemático	30
2.9	Ruteo de circuito impreso principal	60
2.10	Ruteo de circuito impreso secundario	20
3.	<b>Diseño de firmware</b>	
3.1	Definición de casos de prueba	40
3.2	Adaptación de la librería Mbed de la biblioteca del protocolo LoRaWAN	30
3.3	Implementación del protocolo propietario para la comunicación con el servidor	20
3.4	Implementación del driver para mediciones de señales analógicas	10
3.5	Implementación del driver para acceder al bus de subnodos	10
3.6	Implementación del driver para medición de temperatura	10
3.7	Implementación del driver para medición de humedad	10
3.8	Estudio de algoritmo de actualización de firmware	10
3.9	Implementación de algoritmo de actualización de firmware	30
3.10	Implementación de aplicación	20
3.11	Estudio de la arquitectura de bajo consumo	10
3.12	Implementación de las estrategias de bajo consumo	30
3.13	Test de protocolo propietario	20
3.14	Test de driver para mediciones analógicas	10
3.15	Test de driver para medición de temperatura	10
3.16	Test de driver para medición de humedad	10
3.17	Test de algoritmo de actualización de firmware	10
3.18	Test de sistema	45
3.19	Validación	50

Este desglose de tareas permitió construir el esquema gráfico denominado *Activiy*

*On Node* [12]. El AON es un diagrama de precedencia de los hitos más importantes de la planificación. Estos hitos se esquematizan en pequeñas cajas conectadas por flechas y resumen el progreso lógico del avance del proyecto. En la figura 2.1 se detalla el diagrama para el desarrollo realizado. En rosa se muestran las actividades de desarrollo de firmware, en celeste las de desarrollo de hardware y en naranjas las relativas a los trabajos de documentación.

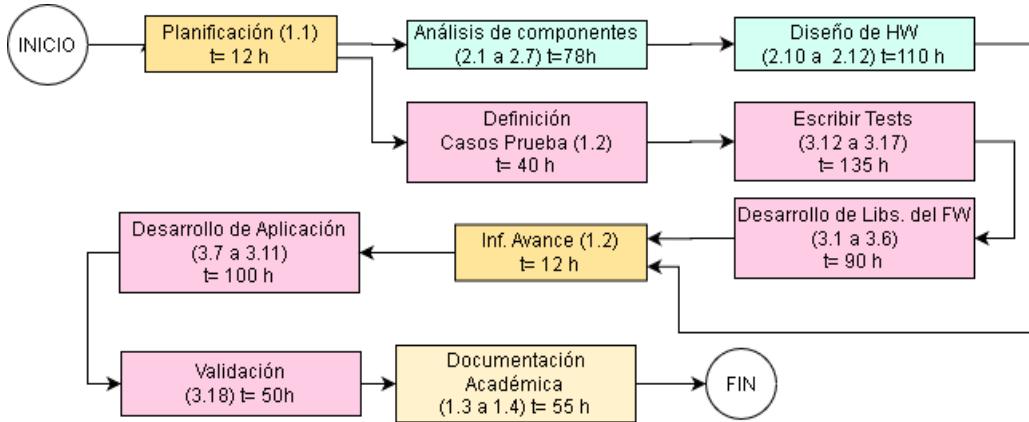


FIGURA 2.1: Diagrama de *Activity On Node*<sup>1</sup>.

### 2.3. Sensor de temperatura y sensor de humedad

En cuanto a la medición de temperatura y humedad que requiere el producto no hay exigencia en cuanto a su desempeño. Los sensores deben medir en un rango de medición de 0 a 100 °C para la temperatura y de 0 y 100 % para la humedad.

Existen muchos productos en el mercado que cumplen con esta característica. No obstante, se encontró que el sensor STH31 (de la empresa Sensirion)[13], en su versión con interfaz I<sup>2</sup>C, cumple los requerimientos de precio y desempeño requeridos para el producto.

El STH31 es un sensor de humedad y temperatura con muchas opciones en cuanto a interfaz, así como a precisión, siempre manteniendo el encapsulado DFN de 2,5 mm x 2,5 mm x 0,9 mm (figura 2.2). Además, el sensor se puede adquirir con una membrana especial [14] que le da grado IP67 (fundamental en la certificación para atmósferas explosivas).



FIGURA 2.2: Circuito integrado utilizado para medir humedad y temperatura [15]

Dentro de los modelos que poseen interfaz I<sup>2</sup>C, se encuentran disponibles los listados en la tabla 2.3. Se eligió el marcado en negrita, dado que posee la membrana mencionada, y sus mediciones poseerán un precisión adecuada.

TABLA 2.3: Versiones de STH31, con I<sup>2</sup>C.

Modelo	Precisión %RH	Precisión °C	Opciones
SHT30-DIS-B	2 @10-90 %RH	0.2 @0-65 C	
SHT31-DIS-B	2 @0-100 %RH	0.2 @0-90 C	
<b>SHT31-DIS-F</b>	<b>2 @0-100 %RH</b>	<b>0.2 @0-90 C</b>	<b>Con membrana</b>
STH31-DIS-P	2 @0-100 %RH	0.2 @0-90 C	Con protección <sup>2</sup>
SHT35-DIS-B	1.5 @0-80 %RH	0.1 @20-60 C	
SHT35-DIS-F	1.5 @0-80 %RH	0.1 @20-60 C	Con membrana

## 2.4. LoRa y LoRaWAN

LoRa es una tecnología inalámbrica que utiliza un tipo de modulación en radio-frecuencia denominada *Chirp Spread Spectrum*, o CSS[16]. Esta tecnología se emplea en comunicaciones militares y espaciales desde hace décadas, y fue incorporada al mundo de IoT por la empresa Semtech. Es un protocolo de capa física propietario de esta empresa. Su gran ventaja es que puede lograr comunicaciones de datos a largas distancias (típicamente kilómetros) y tiene gran solidez frente a las interferencias[17].

Al ser una capa física, LoRa por sí solo no permite implementar una red. Simplemente permitiría realizar un enlace punto a punto con otro transceptor LoRa. Por lo tanto se estandarizó el protocolo LoRaWAN, definido por la “LoRa Alliance”[18]. Este protocolo define las capas restantes del protocolo de red y de la arquitectura del sistema (ver la figura 2.5).

LoRaWAN es responsable de gestionar las frecuencias, la tasa de transmisión y la potencia de los dispositivos.

---

<sup>2</sup>Es una película de material adherido al sensor para protegerlo durante los procesos de ensamblado. Luego a este, se retira para que el sensor opere normalmente.

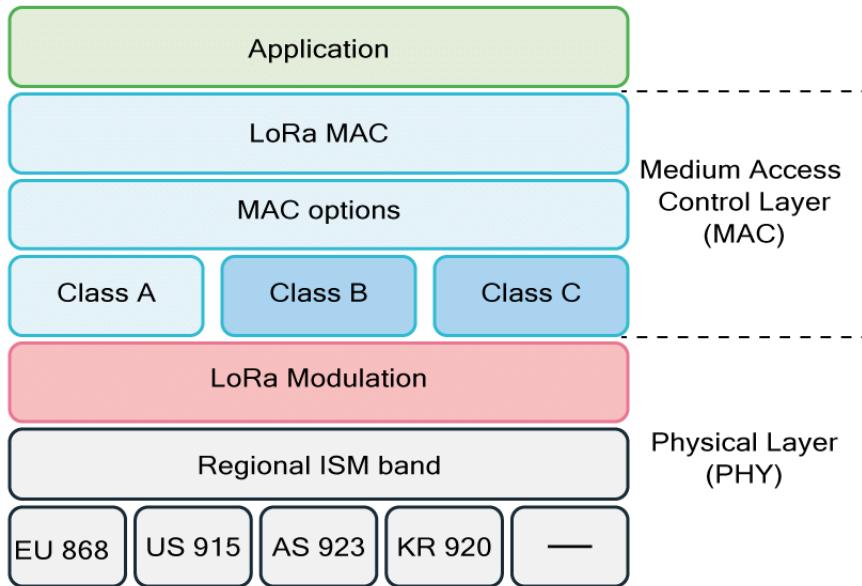


FIGURA 2.3: Capas de abstracción del protocolo de red<sup>3</sup>.

Sobre LoRA podrían montarse otros protocolos de red. A modo informativo se mencionan los siguientes:

1. DASH7 [20]: protocolo de red que se jacta de ser superior a LoRaWAN en términos de prestaciones [21]. Es promovido por la DASH7 Alliance.
2. 6LowPAN [22]: es un protocolo que permite la utilización de IPv6 sobre redes de bajo consumo. Está pensado para redes personales, pero podría utilizarse sobre una red de menor ancho de banda y mayor cobertura [23].

En el presente trabajo se eligió a LoRaWAN como protocolo de red, por las siguientes razones:

- WiAgro ya posee productos que utilizan esta tecnología.
- WiAgro ya posee *gateways* compatibles con LoRaWAN y no dispone de presupuesto para mantener en inventario el equipamiento para otras tecnologías.
- Es un protocolo que está “hermanado” con la capa física elegida.
- Existen muchas implementaciones de software [24][25][26].

En la figura 2.4 se muestra la arquitectura una red LoRaWAN. Muchas veces los *gateways* y los servidores de red se encuentran embebidos en un mismo equipamiento por ejemplo, el MultiConnect Conduit de MultiTech utilizado para el presente trabajo[27].

---

<sup>3</sup>Fuente [19]

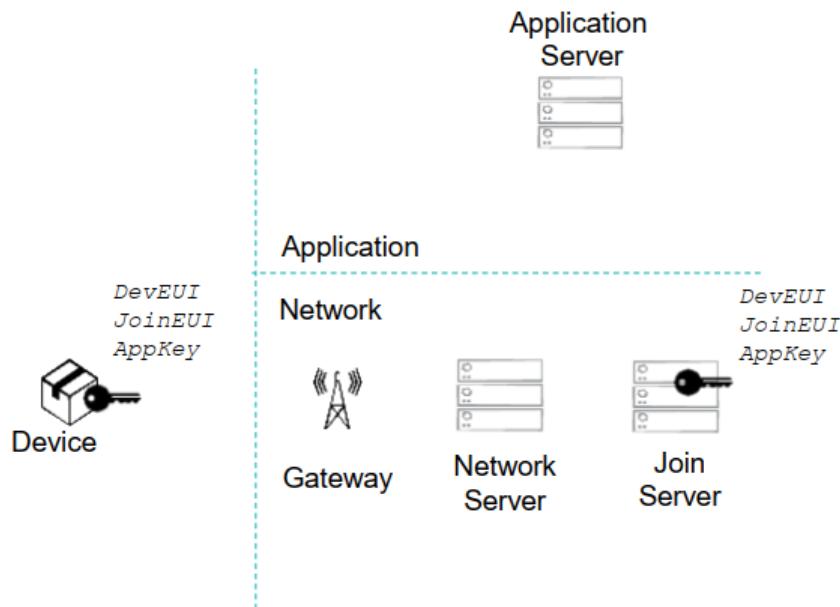


FIGURA 2.4: Arquitectura de una red LoRaWAN<sup>4</sup>.

#### 2.4.1. Modos de operación

Los modos de operación hacen referencia al perfil de transmisiones y recepciones entre un cierto nodo LoRaWAN y el *gateway*, y definen la clase del nodo. Elegir un modo para un cierto dispositivo impacta de manera directa en el consumo del mismo. Esto ocurre porque los perfiles más exigentes mantienen el transmisor encendido un tiempo mayor. Los nodos transmiten datos de manera asincrónica y los comunican de dos posibles maneras: de manera programada o de manera reactiva ante los eventos del nodo. Según la necesidad de la aplicación de cada nodo LoRaWAN el protocolo define:

- **Modo A:** se configura un nodo en este modo cuando el requerimiento fundamental es preservar la energía de la batería. Es el modo más eficiente porque posee un perfil de transmisión planificado cada cierto tiempo. El nodo permanece durante los tiempos muertos en estado dormido. En cada ciclo de transmisión, el nodo abre una ventana en donde permite recibir datos del *gateway* (a esto se le llama *downlink*). Durante el tiempo de reposo, el *gateway* debe almacenar todos los mensajes pendientes que llegan de la red. Los mensajes pendientes en ese período, pueden o no tener un ACK (Acknowledge: confirmación de recepción) del nodo. Si el *gateway* recibe el ACK, lo reenvía al servidor de aplicación. La latencia de la transmisión para que el *gateway* pueda hacer un *downlink* es muy alta.
- **Modo B:** en esta modalidad se le comienza a dar más importancia a la latencia de *downlink*. Sin embargo, el nodo sigue siendo eficiente con respecto al consumo de energía de la batería. El reloj interno del nodo se sincroniza con una baliza enviada por el *gateway* hasta que se considere que el reloj está “enganchado”. Una vez que llega a ese estado, el *gateway* y el nodo negocian un intervalo (*ping*), que definirá el tiempo en el cual el nodo encenderá el receptor para la operación de *downlink*.

<sup>4</sup>Extraído de [28]

- **Modo C:** se baja la latencia del *downlink* a cero, sacrificando el gasto energético de los nodos. Los nodos están escuchando continuamente.

### 2.4.2. Capa de seguridad

La seguridad de los enlaces de las comunicaciones es imprescindible. Hoy en día, es fundamental lograr:

- Que solo los nodos autorizados se conecten a la red.
- Que el tráfico de la red no sea alterado.

El protocolo de seguridad de LoRaWAN está pensado de manera que respete las características de los nodos: bajo consumo, implementación sencilla, bajo costo y alta escalabilidad.

El protocolo utiliza el algoritmo criptográfico AES, que ha sido analizado y aceptado por el NIST [29] como uno de los mejores algoritmos de encriptación.

Cada nodo de la red LoRa posee una “llave de aplicación” (AppKey). Es una llave que debe elegirse aleatoriamente y quedar guardada en los nodos. En general se utilizan métodos de almacenamiento seguros (por ejemplo, los circuitos integrados *Secure Element* de Gemalto[30]). En caso de no contar con un medio de almacenamiento seguro, la seguridad de los nodos dependerá solo de la seguridad física del mismo.

De la AppKey derivan otras dos denominadas llaves de sesión. La primera, llamada llave de sesión de Aplicación (AppSKey), es la encargada de encriptar el contenido crudo de los mensajes a transmitir por el usuario (para el presente trabajo, los valores de las mediciones y otros comandos de configuración y actualización de firmware). Además, la AppKey provee integridad de los datos.

El esquema de capas de abstracción del stack LoRaWAN se puede observar en la figura 2.5. En el mismo también se pueden observar los campos de cada trama que son encriptados o calculados con las distintas llaves de seguridad.

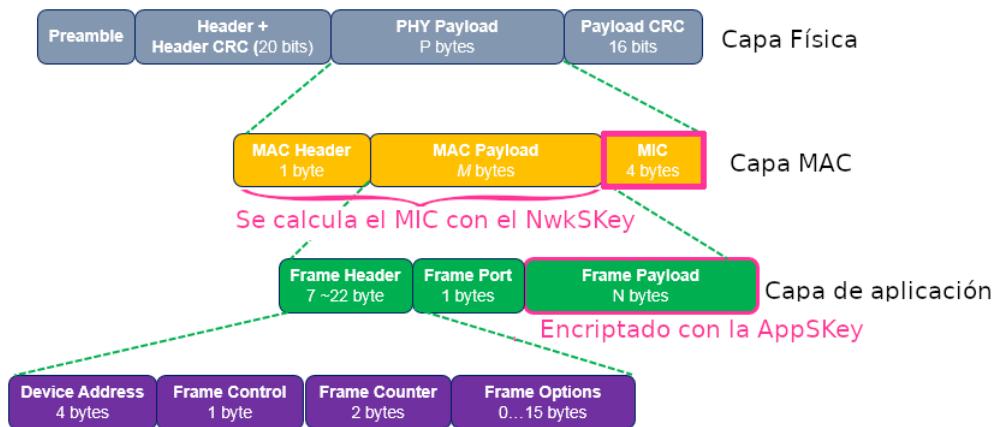


FIGURA 2.5: Descomposición de la trama del stack LoRaWAN<sup>5</sup>.

<sup>5</sup>Traducido y modificado del artículo [31]

## 2.5. Normativas sobre atmósferas explosivas

El manejo de granos a nivel agroindustrial (almacenamiento y transporte principalmente) provoca que el grano genere mucho polvo. Este puede ser combustible, e incendiarse, o crear nubes inflamables y es una fuente de peligro, ya que cuando se dispersa en el aire, puede formar una atmósfera potencialmente explosiva. Además, las capas depositadas de polvo combustible pueden inflamarse y ser fuentes de ignición de dicha atmósfera.

Dentro del marco regulatorio de Argentina, varias resoluciones y decretos nacionales y provinciales hacen referencia a ciertas prácticas obligatorias relacionadas con el almacenamiento de granos en silos. Se intentó encontrar una reglamentación explícita con respecto a la aplicabilidad del estándar, pero no se encontró. Por otro lado, el documento [1] insiste en aplicar buenas prácticas relacionadas con la limpieza y el control de fuentes de ignición. En otra parte del mismo documento se menciona de manera textual: “Utilizar instalaciones, equipos eléctricos y herramientas adecuadas para ambientes pulverulentos. Por ejemplo, utilizar palas de plástico o aluminio (no de acero, por el peligro de chispas) e instalación eléctrica antichispa”.

De los varios decretos y resoluciones provinciales relacionados con el almacenamiento de granos a los que se accedió, se pudo obtener un resumen de los puntos que sí se exigen:

- Limpieza de los establecimientos, para minimizar la sedimentación del polvo.
- Instalaciones con sistemas de control de temperatura y de aireación con el fin de minimizar el riesgo de explosión o incendio.

El único documento [32] que pudo encontrarse en el ámbito nacional (se descargó del sitio de prefectura naval, pero fue redactado por el IMO<sup>6</sup>), donde se menciona una normativa concreta. Este alude a la aplicabilidad de la IEC 60079 en productos como medición de nivel de agua en buques de transporte de granos.

Internacionalmente existen diversas reglamentaciones. Estas las imponen diversos organismos propios de una región o país, y, en su gran mayoría, hacen referencia al estándar internacional IEC60079 titulado “Atmósferas Explosivas” (la figura 2.6 muestra un mapa que resume los estándares y las agencias). En Argentina, IRAM adoptó el mismo estándar y lo bautizó “IRAM-IEC60079”.

La IEC60079 es una normativa técnica que abarca todos los detalles para el diseño, construcción, manufactura e instalación de dispositivos en zonas peligrosas. Es una normativa extensísima, que aplica a una gran variedad de productos. En el caso del presente trabajo se focalizó el análisis de las partes que poseían relevancia para zonas de clasificación relacionadas con las atmósferas de polvo. Estas fueron:

- IEC 60079-0 “Requisitos generales” [34][35].
- IEC 60079-10-2 “Zonas de clasificación. Ambientes de polvo explosivo” [36].
- IEC 60079-31 “Protección de materiales eléctricos contra la ignición de polvo por envoltura del material” “t” [37].

---

<sup>6</sup>International Maritime Organization.

<sup>7</sup>Disponible en [33]

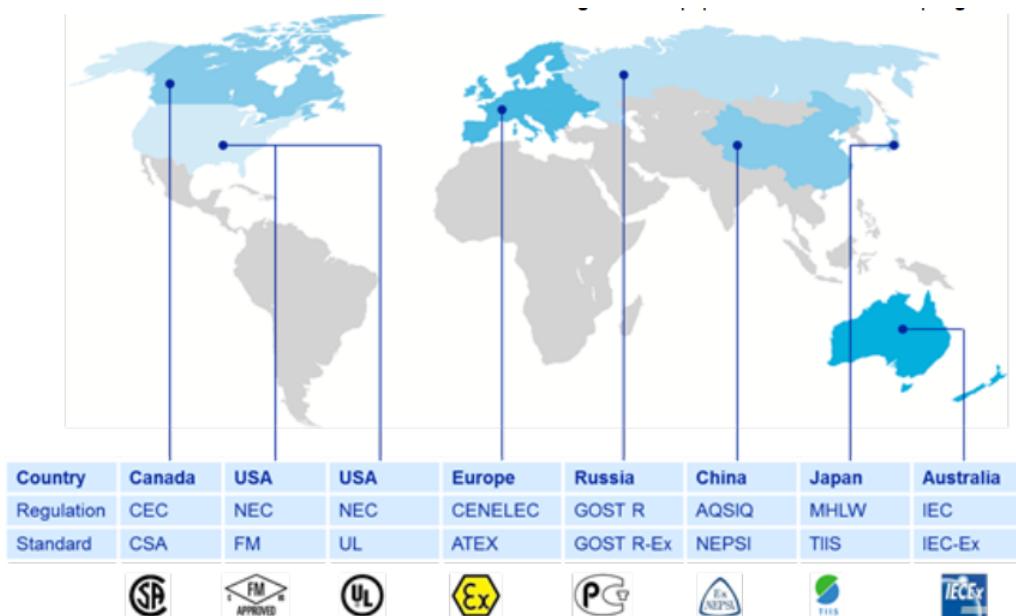


FIGURA 2.6: Estándares y normativas internacionales sobre peligrosidad<sup>7</sup>.

- IEC 60079-11 “Protección del equipo por seguridad intrínseca” “i” [38][39].

Durante la lectura de las normativas anteriormente citadas, hubo que consultar también otras, accesorias a estas, como ser las más relevantes:

- IEC 60664-1 “Coordinación de aislamiento de los equipos en los sistemas (redes) de baja tensión” [40].
- IEC 60529 “Grados de protección proporcionados por las envolventes” [41].

### 2.5.1. Marcado

El marcado de los productos es el conjunto de símbolos y etiquetas que acompañan un producto y que acredita que este cumple con cierto estándar. Ese marcado lo otorga la autoridad certificante y le permite a la empresa dueña del producto, entre otras cosas, distinguirse en el mercado.

En el caso del sistema desarrollado en este trabajo y en relación con el estándar IEC60079, no fue tarea fácil definir qué certificación buscar cuando el producto estuviera disponible. Por ello se consultó a INTI intentando obtener ayuda respecto a este tema (no se pidió una consultoría, por lo que a futuro habrá que hacerlo, con el posible riesgo de tener que modificar algún elemento del diseño actual). Desde INTI se logró obtener alguna información al respecto para definir el marcado en función del tipo de equipamiento a desarrollar.

El marcado “objetivo”<sup>8</sup> del equipo es:

- **Para el dispositivo maestro:** “Ex ta[ia] IIIB T90C Da”.
- **Para los subnodos:** “Ex ia IIIB T90C Da”.

<sup>8</sup>Se indica así porque el dispositivo no está certificado. Para ello, habrá que solicitar una consultoría en función de tener un marcado acertado.

A continuación, se desglosará cada parte de esos marcados, para luego explicar cada uno de los mismos:

- **Ex:** es un símbolo que indica que el producto está certificado para ser instalado en zonas con atmósferas explosivas.
- **ia:** el término “i” hace mención a que el dispositivo cumple con los requerimientos de la IEC 60079-11. El dispositivo en cuestión estará diseñado para que el modo de protección sea el de seguridad intrínseca. Este método de protección se basa en limitar la energía de los circuitos eléctricos para que sean incapaces por sí solos de generar una explosión. La letra “a” es una subclasiﬁcación que describe del nivel de protección.
- **ta[ia]:** el término “t” hace mención a que el dispositivo cumple con los requerimientos de la IEC 60079-31. El dispositivo en cuestión estará diseñado para que el método de protección contra atmósferas de polvo lo proporcione el gabinete. El modo de protección que se encuentra entre corchetes indica que el dispositivo en cuestión deberá conectarse a otros dispositivos externos y que, por otorgar energía a estos, deberán tener protecciones en esas salidas por seguridad intrínseca.
- **IIIB:** enmarca al producto para ser instalado en zonas cuyas atmósferas poseen polvo no conductivo (el polvo de grano entra en esa categoría).
- **T90 °C:** clasificación en temperatura. Indica que la máxima temperatura de superficie del dispositivo será de 90 °C.
- **Da:** hace referencia al tipo y al nivel de protección. La “D” indica que el producto fue certificado para atmósferas con polvo. La “a” alude al nivel más exigente.

En las siguientes subsecciones se explican requerimientos extra que se tuvieron en cuenta en el desarrollo basado en los requisitos del estándar y las clasificaciones del producto antes mencionadas.

### 2.5.2. Consideraciones de diseño

En esta sección se comentan partes del estándar, aplicadas al trabajo realizado, cruciales para realizar el diseño mecánico y electrónico.

**Gabinete:** para el caso del trabajo realizado el cliente optó por utilizar gabinetes plásticos. Para el modo de protección “t” se emplea el gabinete como garantía de que en su exterior no se provoquen explosiones. Los gabinetes plásticos poseen algunas particularidades que los de metal no: se deforman más fácilmente por temperatura y, por otro lado, pueden acumular carga estática. Para cubrir los requisitos de la normativa, estos dos parámetros son ensayados estrictamente.

Mecánicamente, el gabinete también está sujeto a requisitos estrictos como ser:

- Deberá tener grado de estanqueidad IP6x (en este caso, lo importante no es la humedad que entre a su interior, sino el polvo).
- Si posee tapas de acceso al interior, deberán tener un *o-ring* continuo. En este caso, la tapa solo se deberá poder abrir utilizando una herramienta.

**Temperatura:** una de las exigencias de esta normativa es que la temperatura de superficie del dispositivo no supere un cierto umbral. Con esto se pretende evitar

que temperatura externa no llegue a las temperaturas de ignición del polvo. Por ejemplo, la normativa IEC60079-14 establece que la temperatura de superficie no debe superar los dos tercios de la temperatura de autoignición de los granos o que la temperatura del dispositivo esté por debajo de la temperatura de ignición de una capa de 5 mm de polvo, menos 75 °C. De estas dos temperaturas deberá adoptar la mínima.

Analizando un caso práctico, se buscaron tablas ([42] [43]) con las que se pudo calcular un peor caso del cálculo mencionado. La temperatura de autoignición del maíz es de 400 °C (Otras especies superan ese valor). Este valor determina una máxima temperatura de carcasa de 266 °C. Si se realiza el cálculo según el criterio de las capas de polvo, las temperaturas de ignición de una capa de polvo de grano es de 250 °C, por lo tanto el segundo criterio arroja una temperatura máxima de 175 °C. A partir de ambos criterios, tomando la mínima temperatura, el valor que no podrá superarse es el de 175 °C.

**Baterías:** respecto de las baterías, la normativa posee tabuladas las composiciones químicas más conocidas, estableciendo requerimientos para cada una. En caso emplear una batería cuya composición química no se conozca o cuyo fabricante no haya realizado ensayos sobre las mismas, la normativa exige y propone un grupo de pruebas obligatorias. Este grupo de pruebas puede ignorarse si, en el producto final, las baterías son ensambladas con una resistencia de limitación de energía, y además si están empotradas en un compuesto sólido. En este caso, hay exigencias sobre las características del resistor elegido y sobre la composición del compuesto para empotrar (generalmente son resinas especiales que garantizan que, con el aumento de temperatura de la batería y de la resistencia limitadora de corriente, no se generen grietas y/u orificios que dejen aperturas al exterior del conjunto).

El dispositivo maestro se diseñó para utilizar baterías de litio-disulfuro de hierro, cuya composición química se encuentra aceptada en el estándar. Otra química aceptada por el mismo es la de litio-cloruro de tionilo.

La parte once del estándar, expande el concepto del uso de batería, porque justamente habla del modo de protección por limitación de la energía. Como la batería que se eligió para el trabajo ya está certificada bajo esta normativa, no requiere ensayos sobre estas. Lo mismo ocurre con algunos fabricantes de baterías de litio-cloruro de tionilo (por ejemplo, Saft[44]).

Las baterías diseñadas para que superen los ensayos de la normativa incluyen la siguientes consideraciones:

- Sistemas de venteo para liberar excesos de temperatura y presión interna.
- Fusible reseteable (PTC: Positive Temperature Coefficient) para evitar entregar energía a un circuito externo que esté en una situación de corto circuito.
- Carcasa reforzada para minimizar el daño de las celdas por abusos físicos de las mismas.
- Control de temperatura interna.

El utilizar baterías pre-certificadas acelera el desarrollo y minimiza también el esfuerzo en el diseño del mismo. Además, repercute en un diseño más sencillo, menos voluminoso y de manufactura más económica [45].

### Energía:

Un circuito eléctrico intrínsecamente seguro deberá satisfacer tres puntos:

- No deberán generarse chispas durante las pruebas y evaluaciones.
- La temperatura no deberá aumentar considerablemente, para evitar la ignición por superficies calientes.
- Deberá respetar las líneas de fuga y los distancias en aire.

Este punto aplica al diseño de la interfaz eléctrica entre el nodo maestro y los subnodos y al diseño de los subnodos en sí. El principio de la seguridad intrínseca consta en lograr limitar la energía de los circuitos para que no se produzcan chispa y para que no eleven la temperatura por arriba de los umbrales de auto-ignición del polvo.

Con respecto a la interfaz entre el nodo maestro y la red cableada, se implementaron circuitos de limitación de energía. En el caso de la red cableada, las líneas a proteger serán dos: una para datos y una para alimentación, las cuales cubren un total de cuatro conductores. Los circuitos de limitación de energía determinan la máxima tensión y la máxima corriente que podrán entregar esas interfaces hacia afuera del gabinete del nodo maestro. Los circuitos propuestos por la normativa son los denominados “Barreras de seguridad de diodos”. Están compuestas básicamente por diodos zener y algunos resistores de limitación de corriente. Hay diversas topologías circuitales para implementar este tipo de barreras y dependen del sistema en donde se utilicen. En la figura 2.7 se puede observar la topología elegida.

La normativa detalla, según la clasificación del dispositivo a certificar, condiciones de diseño referidas a los elementos clave que brindarán protección.

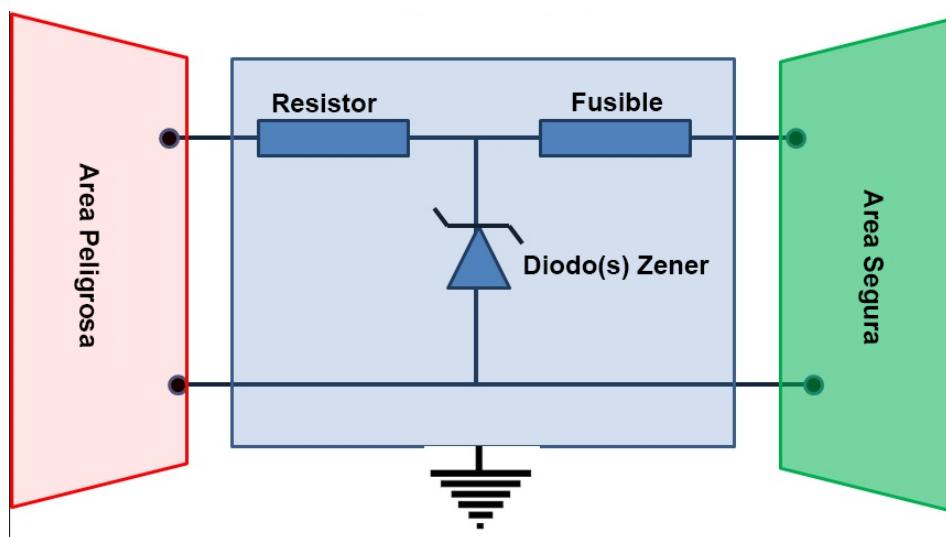


FIGURA 2.7: Esquema de una barrera zener básica. Dependiendo del tipo de certificación buscado, este esquema puede poseer más complejidad <sup>9</sup>.

<sup>9</sup>Fuente: versión traducida de[46].

De acuerdo a ISO60079, los componentes que brinden protección (en este caso, los de la barrera zener) cumplen con los requisitos de diseño marcados en la tabla 2.4

TABLA 2.4: Requisitos de diseño de los componentes que brindan protección.

<b>Requisito</b>
Soportar por lo menos 1,5 veces la tensión máxima del dispositivo.
Los diodos de protección requerirán una resistencia en serie para cumplir con la limitación de transitorios y a su vez soportar 1,5 veces la potencia de operación cuando operen en el modo zener.
Como el producto se certificará con modo de protección ia, la "a" indica que deberán ser provistos tres diodos en paralelo (por si fallan dos).
La corriente que deberán soportar los componentes deberá ser al menos de 1,5 veces la máxima de corriente continua para lo que fueron diseñados.
El valor de ruptura del fusible deberá elegirse para proteger a los diodos zener cuando la tensión de entrada supere la de los mismos.

Además de los requisitos mencionados, la normativa indica que el total de capacidad e inductancia (serie y paralelo) de los circuitos diseñados deberá estar limitada. Estos elementos (parásitos o no), al almacenar energía, pueden generar descargas que generen chispas.

#### Conectores y prensacables:

La utilización de conectores plantea una consideración importante en el diseño. Los conectores exponen, desde un circuito en área no segura, los contactos a zonas peligrosas. Esta exposición, hace que existan requerimientos para la elección de los mismos tales como:

- Ser conectores polarizados.
- No deberán ser iguales a otros conectores del sistema.
- Deberán poseer bloqueo mecánico, para que no se desconecten accidentalmente cuando el sistema esté energizado.
- Deberán mantener el grado IP del gabinete.
- Deberán respetar las exigencias con respecto a las líneas de fuga y las distancias en aire.

De no poder cumplir con los requisitos arriba expuestos, la opción es utilizar, internamente al equipo, borneras y la utilización de prensacables.

#### Radiaciones de RF:

El grupo III establece que la máxima potencia radiada puede ser de 6W. Como la potencia radiada por el nodo LoRa posee un máximo de 20dBm (0,1 W), no hubo que preocuparse. La normativa IEC60079-14, por otro lado, indica que para zonas peligrosas con clasificación "Da+" la densidad de energía deberá ser inferior a 0,1 mJ/mm<sup>2</sup> (promediando cada 5s). Si consideramos antenas isotópicas para el cálculo, esa limitación tampoco es un problema.

### Diseño de los circuitos impresos:

Para garantizar que los elementos de protección cumplan el objetivo, se diseña el aparato para que cumpla una serie de requerimientos:

- Separación de las partes conductivas entre circuitos intrínsecamente seguros y circuitos no intrínsecamente seguros.
- Separación de las partes conductivas entre circuitos intrínsecamente seguros y otros circuitos intrínsecamente seguros.

Esas separaciones hacen referencia a las líneas de fuga y a las distancias en aire.

Las líneas de fuga son las distancias más cortas entre diferentes partes de circuito medidas a través de la superficie.

Las distancias en aire son las distancias más cortas entre diferentes partes de circuito medidas a través del aire.

Se puede visualizar la diferencia en la figura 2.8 aplicado a un circuito impreso genérico. El mismo concepto puede aplicarse a otro tipo de elemento eléctrico, por ejemplo, un conector.

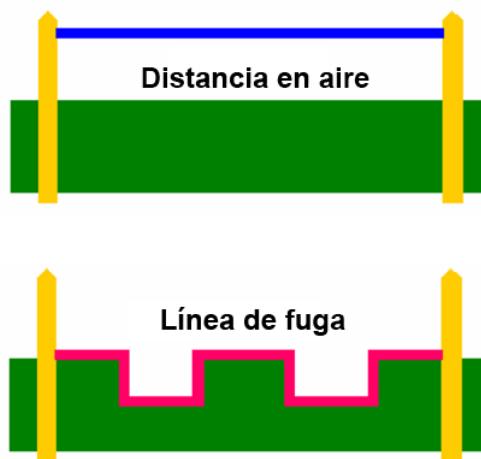


FIGURA 2.8: Concepto de líneas de fuga y distancias en aire, aplicados a un circuito impreso<sup>10</sup>.

Las exigencias respecto de las líneas de fuga y la distancia en aire, dependen de varios factores constructivos, por ej, la aislación que pueda llegar a haber entre los conductores. En caso de los circuitos impresos, por ejemplo, no es lo mismo que se aplique o no la mascara antisoldante. Muchas veces con aplicarla no alcanza, y se requiere del uso de coberturas conformes<sup>11</sup>. Existen otras técnicas para aumentar las líneas de fuga que consisten en practicar diferentes tipos de ranuras al sustrato del circuito impreso. El más utilizado es el método de ranura pasante, como se puede observar en vista superior de la figura 2.9

<sup>10</sup>Fuente: versión traducida de [47].

<sup>11</sup>Las coberturas conformes son películas poliméricas que se aplican al circuito impreso ensamblado por sobre los componentes, soldaduras y pistas que permiten, entre otras cosas, reducir las distancias entre conductores, al momento de evaluarlas para el cálculo de líneas de fuga y distancias en aire. Para obtener mayor información consultar [48] [49]

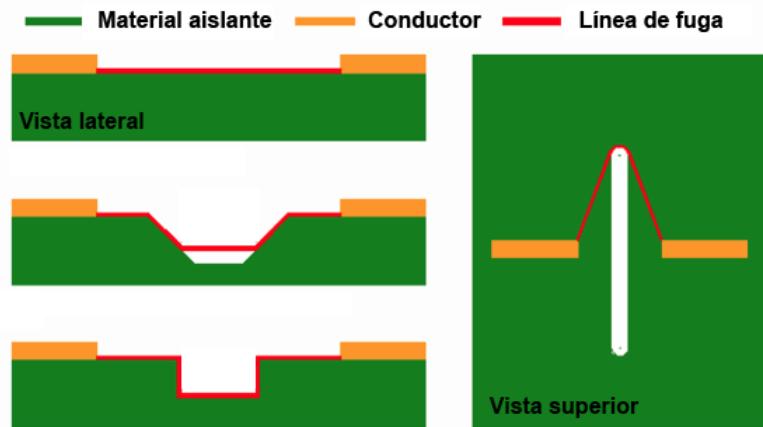


FIGURA 2.9: Técnicas para aumentar las líneas de fuga en un circuito impreso<sup>12</sup>. Las ranuras permiten aumentar las longitudes de las líneas de fuga.

#### Cableado:

El cableado también es considerado parte crítica en términos de la seguridad intrínseca porque, al poseer inductancias y capacidades parásitas, también son elementos que almacenan energía y se incluyen en la evaluación final del ente certificador.

#### Selección de componentes:

La elección de los componentes está reglamentada solamente para los que brindan seguridad. Estos son, más que nada, los que limitan la energía entre módulos.

En el caso de los resistores, se utilizan para limitar la corriente de los circuitos. Deberán ser de metal film. Para este tipo de componente el modo de falla considerado será el de circuito abierto. Deberán cumplir los requisitos técnicos detallados en la tabla 2.4.

En el caso de los fusibles, deberán soportar 1,5 veces la carga de cualquier transitorio que se presente en la carga. La capacidad de ruptura de los mismos deberá ser al menos de la corriente máxima de salida del equipo.

Para calcular las distancias en aire y líneas de fuga, deberá emplearse como tensión nominal (solo para este caso) la siguiente fórmula:

$$U_N = 1,7I_nR_L \quad (2.1)$$

donde  $I_n$  es la corriente máxima que circule por ellos, y  $R_L$  la resistencia limitadora en serie.

---

<sup>12</sup>Fuente: versión traducida de [47].



## Capítulo 3

# Diseño e Implementación

En este capítulo se detallan las decisiones de diseño tanto para el hardware del dispositivo como del firmware. Además, se presenta un detalle de los protocolos diseñados para la comunicación entre los subnodos y el cliente LoRaWAN.

### 3.1. Bloques constitutivos del sistema

El dispositivo diseñado está compuesto principalmente por:

- Un nodo maestro, que es el encargado de transmitir los datos a la red LoRaWAN, de gestionar la energía proporcionada a la red de subnodos y de iniciar la comunicación con el primer nodo de la cadena, para obtener las mediciones en cada punto del arreglo de sensores.
- Una red de veinticinco subnodos, encargados de realizar las mediciones de los sensores, en cada punto del reservorio de granos, y a su vez replicar las consultas de los nodos anteriores de la secuencia espacial.

El esquema del sistema puede observarse en la figura 3.1.

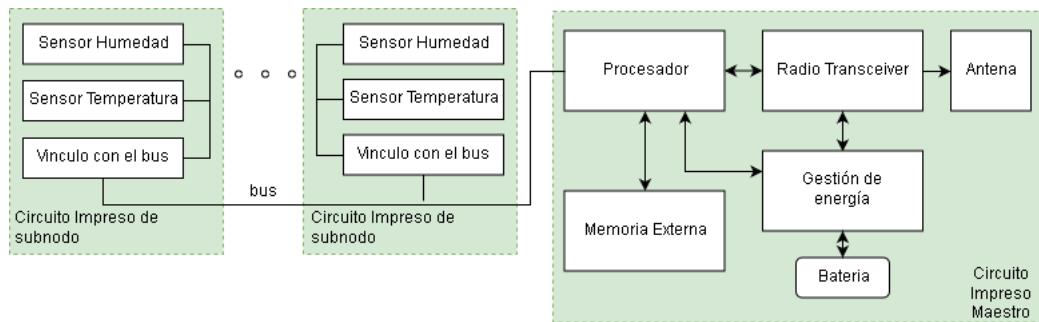


FIGURA 3.1: Diagrama en bloques del sistema.

Como puede observarse también en la figura 3.1, el nodo maestro cuenta con:

- Procesador: es el CPU más importante del sistema ya que se encargará de gestionar todos los procesos principales.
- Transceptor: es el circuito integrado en donde se hayan todos los módulos de capa física para implementar la comunicación inalámbrica.
- Gestión de energía: circuitos encargados de transformar y limitar la energía del sistema.

- Memoria: la memoria externa al procesador principal servirá para almacenar la imagen de nuevos firmware que se transmitan por el aire.

Además, los subnodos cuentan con:

- Procesador: el CPU de los subnodos se encargará de reunir todas las mediciones cuando se las soliciten y de gestionar el protocolo de comunicaciones de la red, con el maestro y el resto de los subnodos.
- Sensores: elementos de medición de los parámetros relevantes en el almacenamiento de granos.

### 3.2. Protocolo de red de subnodos

En esta sección se describe el camino recorrido para llegar la solución adoptada, con respecto a la implementación de la red entre los subnodos, tanto de capa física como nivel superior.

Durante la elección de los componentes, se eligió como sensor de temperatura y humedad al circuito integrado STH31 que posee ambos transductores en el mismo encapsulado, tal como fue descrito en la sección 2.3. El circuito integrado posee una interfaz I<sup>2</sup>C. Dado que la red de sensores requiere veinticinco subnodos, se evaluaron algunas topologías de red, como para permitir comunicar el dispositivo maestro con dichos elementos.

Para abaratar costos, originalmente se planteó utilizar directamente el bus I<sup>2</sup>C para alcanzar a los veinticinco subnodos (espaciados físicamente en 1 metro) empleando un repetidor que permitiera manejar al bus correctamente. Se encontró que el circuito integrado P82B715 (entre otros) hubiera podido servir. De todas maneras, la idea no era viable dado que cada circuito integrado de sensor de temperatura / humedad posee la misma dirección física. Se encontraron soluciones [50] pero todas invalidaban el diseño, más que nada por la cantidad de conductores que debía tener el bus o la cantidad de componentes extra que se hubieran necesitado.

Una alternativa posterior incluyó a un microcontrolador de bajo costo que contara con los recursos mínimos para entablar conversación con el sensor y que tuviera una UART para la comunicación asincrónica con el maestro a través de un transceptor (RS485, LIN u otro protocolo de capa física, con bus compartido). Esta solución también se descartó porque no permitía, de manera sencilla, identificar a cada nodo en la red. Este requerimiento fue muy importante, porque se buscó que el arreglo completo sea fácil de configurar y mantener. Otro factor que influyó en descartar esta idea fue el costo extra que cada transceptor le sumaba a cada uno de los nodos.

Las soluciones que fueron descartadas se encuentran resumidas en la tabla 3.1.

TABLA 3.1: Ideas descartadas para implementar la red de subnodos.

Alternativa	Motivo por el cual fue descartado
Expansores I <sup>2</sup> C	Direcciones físicas de los sensores.
Expansores I <sup>2</sup> C + líneas	Resultaría en un bus de más de veintiocho cables.
Microcontrolador + transceptor	Mayor costo y necesidad de configuración.

Finalmente se llegó a la siguiente solución: conectar cada subnodo al siguiente de la cadena con una UART punto a punto, según una topología *daisy chain*[51]. Esto requirió un microcontrolador un poco más grande que el de la segunda solución, porque cada subnodo iba a necesitar dos UARTs. Las UARTs entre un subnodo y el adyacente (incluso el del extremo, con el maestro) se conectaron con niveles de tensión TTL, debido a que la distancia entre ellos iba a ser de 1 metro. El microcontrolador elegido, además, posee el bus I<sup>2</sup>C local para el sensor de temperatura/humedad. Con esta topología, se permitió comunicar todos los nodos, con un protocolo de red sencillo, que permitiría, además, conocer la posición relativa al máster, para cada uno. La topología final puede observarse en la figura 3.2.

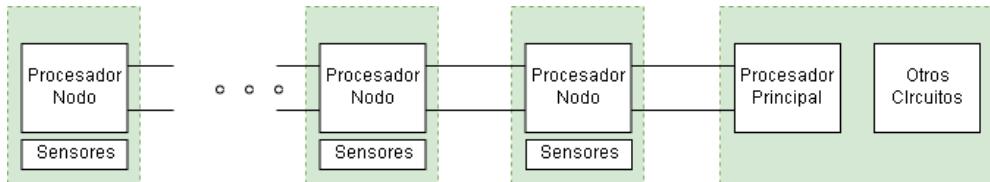


FIGURA 3.2: Diagrama de interconexión de la red de subnodos.

### 3.3. Diseño de hardware

En esta sección se aborda la descripción del hardware desarrollado y los criterios que han sido tenidos en cuenta a lo largo del proceso de diseño.

#### 3.3.1. Gabinetes

La planificación original del desarrollo no incluyó el diseño del gabinete. Posterior a la planificación, al relaviar la necesidad de que el sistema requería estar certificado por la IEC60079, se evidenció que el gabinete era un elemento importante del sistema, entonces, para no perder tiempo en el diseño del mismo, se eligió un gabinete “de catálogo” que cumpliera con los requisitos relevantes.

Los requerimientos del gabinete buscado para el nodo maestro buscado, se basaron en las exigencias presentadas en la sección 2.5, que son las siguientes:

- Deberá ser plástico.
- Deberá poseer un compartimento para dos baterías AA.
- Deberá ser estanco IP6x.

Se buscó en proveedores internacionales como DigiKey o Mouser, y se encontró un gabinete que, además de cumplir los requisitos, tenía espacio interno suficiente como para albergar la electrónica de cliente LoRaWAN. El gabinete es el

RH3001BK [52] cuyo fabricante es Hammond y se puede observar en la figura 3.3.

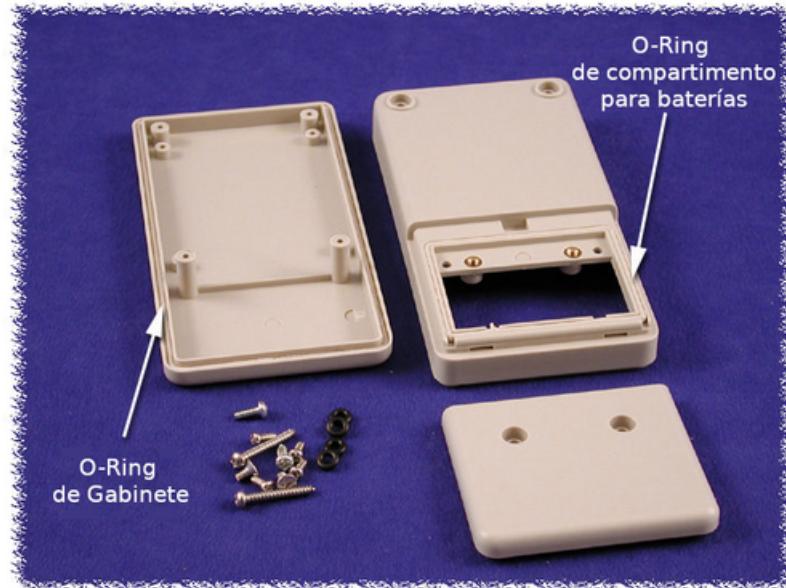


FIGURA 3.3: Gabinete elegido para alojar al cliente LoRaWAN<sup>1</sup>.

También se realizó una búsqueda para elegir el gabinete para los subnodos. En este caso, el requerimiento fue menor porque para el caso de los subnodos, el gabinete no es el elemento que brinde protección para los ambientes potencialmente explosivos. Solo brinda protección mecánica para el circuito impreso. Se diseño el circuito impreso para que utilice un gabinete de la empresa Chillemi, con modelo P07N [53], como se ve en la figura 3.4.



FIGURA 3.4: Gabinete elegido para alojar al circuito impreso de cada subnodo<sup>2</sup>.

<sup>1</sup>Fuente: versión traducida de [52].

<sup>2</sup>Fuente: [53]

### 3.3.2. Cliente LoRa

Para el diseño de los circuitos impresos se utilizó el software Altium Designer, particularmente por tener mas experiencia que con otros software y por contar con una biblioteca de componentes usada en varios trabajos anteriores.

El circuito impreso fue diseñado en doble faz, con montaje de componentes de un solo lado.

El diseño final del circuito impreso cliente LoRaWAN se muestra en la figura 3.5. En las siguientes subsecciones se detallará cada módulo.

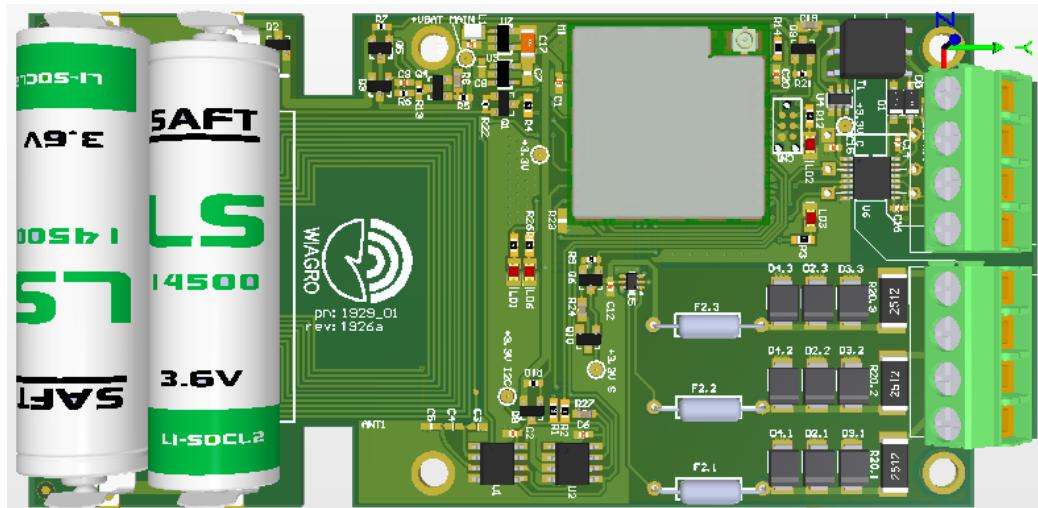


FIGURA 3.5: Render del cliente LoRaWAN

#### Plataforma

Para la implementación del Cliente LoRa se decidió utilizar el módulo xDot [54] de la empresa MultiTech.

El xDot es un módulo preparado para desarrollar productos que requieran de enlaces LoRa y salir rápidamente al mercado. Además permite lograr pruebas de concepto rápidamente. El mismo incluye (se agregan características relevantes para el diseño):

- Un procesador ARM Cortex M1 STM32L151 [55] cuyas características más importantes para el desarrollo son:
  - 256kB de Memoria Flash, 32kB de Mamoria RAM y 8kB de Mamoria eeprom.
  - 3 UARTS.
- Un trasceptor LoRa STX1272 [56].
- Un chip de seguridad contra *tampering*<sup>3</sup> del fabricante Gemalto (opcional, según el xDot que se utilice).
- Circuitos extra de soporte.

<sup>3</sup>La traducción literal es "manoseo". Es la práctica de alterar, dañar o usar erróneamente un dispositivo para, en este caso, extraer la información que posee en su interior.

Existen diferentes versiones según la región del mundo en donde se utilicen. En Argentina, se utiliza el modelo australiano que opera en la banda de frecuencias de 915 MHz.

El módulo trae un firmware precargado de fábrica que permite operar con la red LoRaWAN a través de comandos AT. De esta manera, el diseño requeriría un microcontrolador extra para enviarle dichos comandos. En el caso del trabajo realizado, no fue necesario conectarle un microcontrolador externo ya que se utilizó al mismo microcontrolador del módulo para realizar todas las operaciones del dispositivo. Lamentablemente no está disponible el esquemático del xDot de manera pública. No obstante, la hoja de datos provee una tabla en donde se muestran los recursos del microcontrolador que se pueden usarse de manera externa. Esto ayudó a mapear los recursos a las necesidades del diseño del nodo de la red. En la figura 3.6 se muestra la asignación de pines disponibles del módulo. Esta figura acompaña a la tabla 3.2. Esta indica para cuales funcionalidades se asignaron los recursos disponibles del módulo.

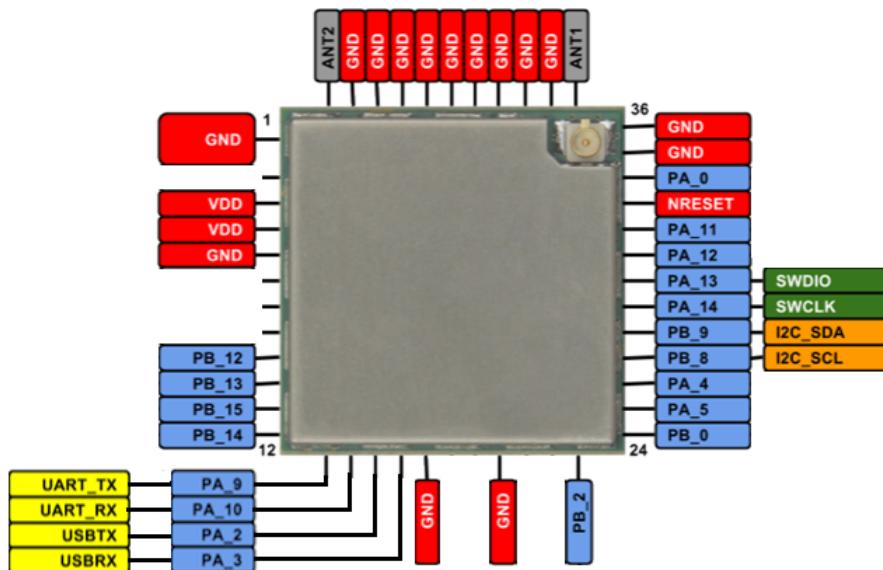


FIGURA 3.6: Recursos externos disponibles del módulo xDot<sup>4</sup>.

---

<sup>4</sup>Fuente: elaboración propia basada en diagrama extraído de [57]

TABLA 3.2: Patillaje disponible del módulo xDot

xDot	STM32	Función
9	PB12	Power Off: permite apagar al dispositivo
10	PB13	EN I <sup>2</sup> C: habilita de tensión de alimentación a circuitos asociados al bus I <sup>2</sup> C.
11	PB15	LED 6: controla un LED.
12	PB14	EN VBAT: habilita la tensión de batería a los circuitos asociados con la medición de la misma.
13	PA9	485 TX: pin de transmisión de datos a la interfaz 485.
14	PA10	485 RX: pin de recepción de datos a la interfaz 485
15	PA2	RED TX: pin de transmisión de datos a la red de sensores.
16	PA3	RED RX: pin de recepción de datos a la red de sensores
23	PB2	LED 3: controla un LED.
24	PB0	EN RED: habilita de tensión de alimentación a la red de sensores.
25	PA5	LED 1: controla un LED.
26	PA4	485 DIR: pin de control de dirección de los datos en el transceptor 485.
27	PB8	I <sup>2</sup> C SCL: línea de reloj para el bus I <sup>2</sup> C.
28	PB9	I <sup>2</sup> C SDA: línea de datos para el bus I <sup>2</sup> C.
29	PA14	SWCLK: línea de reloj para el programador y <i>debugger</i> .
30	PA13	SWCLK: línea de datos para el programador y <i>debugger</i> .
31	PA12	LED 2: controla un LED.
32	PA11	EN C: habilita de tensión de alimentación a circuitos asociados a la interfaz 485.
33	NRST	Circuito de reinicio del microcontrolador.
34	PA0	WAKE: Utilizado para despertar al microcontrolador de forma externa.

Los recursos del xDot se utilizaron principalmente para:

- Las interfaces de comunicaciones.
- Las líneas de control para habilitar/deshabilitar dominios de energía.
- Manejar los dispositivos I<sup>2</sup>C locales.

Además, se asignaron algunos pines libres a 4 LEDs. Esto se hizo porque, al ser un prototipo, no es importante el consumo. En el armado del producto final no se soldarán ya que el usuario final no podrá verlos (el gabinete no posee aberturas para esto). Eventualmente esos recursos que no era estrictamente un requerimiento, podrían utilizarse para otra función a futuro.

El módulo xDot posee tabulados los consumos energético para diferentes modos de operación. Éstos se incluyen en la tabla 3.3.

TABLA 3.3: Consumos de corriente del módulo xDot para VDD=3,3 V (en mA)

Condición	Min	Typ.	Max
Modo reposo		11,1	
Transmitiendo a 2 dBm		50	
Transmitiendo a 11 dBm		62,5	
Transmitiendo a 20 dBm	120		250
Recibiendo		26,8	
Modo dormido	0,0021		

### Baterías

En la sección 2.5.2 se mencionó que la química de la batería elegida fue litio-disulfuro de hierro.

Originalmente el cliente había propuesto utilizar las de litio-cloruro de tionilo ( $Li/SOCl_2$ ). Este tipo de química, si bien posee una capacidad de carga muy grande, posee también una resistencia interna alta. Esto hace que, para los pulsos de transmisión del transceptor LoRa, el nivel de tensión baje y aumente el riesgo de que se reinicie el procesador principal u otra lógica asociada.

Por otro lado, una ventaja de las baterías de litio-disulfuro de hierro, por cierto, es que tienen una tensión nominal de 3,6 V, cuyo valor permite utilizar un LDO (Low Dropout Regulators<sup>5</sup>) de ultra bajo  $Iq$ <sup>6</sup>.

Por lo tanto, se prefirió cambiar la química de las baterías por las de litio-disulfuro de hierro ( $Li/FeS_2$ ). Se encontró que la empresa Energizer provee unas baterías de dicha química, certificadas (Cuyo modelo es L91 [58]). Es una química que permite altas descargas pulsantes y la capacidad es de 3500 mAh. Lo malo de este tipo de baterías es que su tensión nominal es de 1,5 V, por lo tanto se tuvo que colocar dos baterías en serie. A su vez, para regular 3,3 V (que es la tensión de alimentación elegida para el sistema) a partir de 3 V se utilizó un conversor elevador de tensión para regular a la salida los 3,3 V requeridos.

No obstante, y debido a que se consideró el diseño del circuito impreso un prototipo, se desarrolló un circuito genérico que permitiera, utilizando distintas lista de materiales, alimentar el sistema con cualquiera de las baterías mencionadas.

En relación a la peligrosidad en relación a las atmósferas explosivas, algunas baterías de fabricantes reconocidos pueden brindar modelos con certificación para la IEC60079-11. Otros modelos de esos mismos fabricantes, no están precertificados, pero brindan toda la documentación técnica para presentar al ente certificador. Existen también, marcas secundarias (o sin marca, como algunas procedentes de china) que en general no poseen la documentación completa. En ese caso, el

<sup>5</sup>Reguladores de baja caída. Son reguladores lineales de tensión cuya caída entre la entada y la salida se encuentra en el rango de los mV

<sup>6</sup>Corriente de reposo: es la corriente que consume el circuito integrado cuando en la salida no se conecta nada, o sea no hay corriente de salida. Lo ideal en cualquier elemento de conversión de tensión o de potencia, es que  $Iq$  sea cero. El valor de  $Iq$  contribuye a reducir la eficiencia de conversión.

estándar IEC 60079 indica que hay que someterlas a pruebas específicas para que se permita usarlas en ambientes con potencial peligrosidad<sup>7</sup>.

Para el caso de las baterías elegidas, la certificación que posee es: Ex ia IIC Ga. Sin embargo, esta certificación no es válida para más de dos baterías en serie. En el caso del trabajo realizado, no es un problema porque se resolvió utilizando dos en serie.

#### Power switches:

Unas de las características más importantes del diseño es preservar la utilización de la energía proveniente de las baterías. Para ello, se planificó una arquitectura de llaves que definen zonas en las cuales puede o no haber tensión aplicada, para que, cuando un módulo no se necesite, no esté encendido. La idea de esta funcionalidad fue disminuir al máximo la corriente de reposo del sistema<sup>8</sup>.

La figura 3.7 muestra la arquitectura respecto de la segmentación las zonas. Se identificaron las llaves con letras, para poder explicar el propósito de cada una.

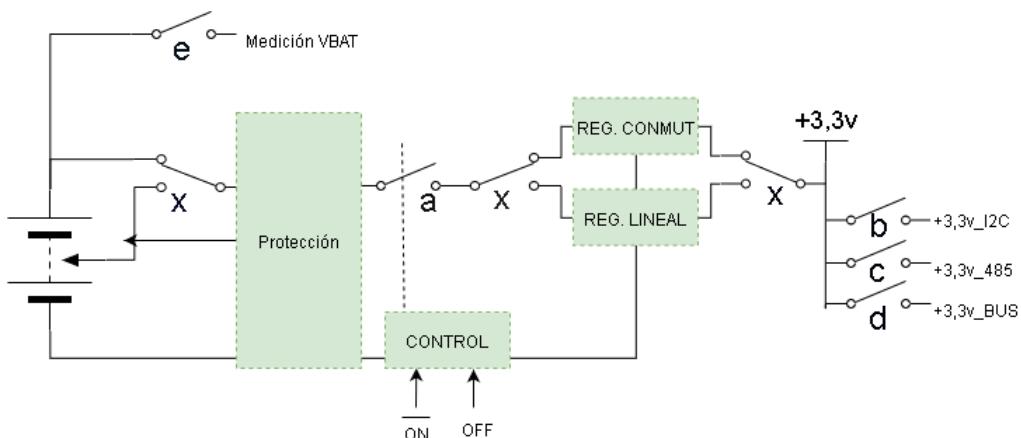


FIGURA 3.7: Diagrama en bloques de la arquitectura de segmentación de zonas.

En primer lugar, el dispositivo principal del sistema fue diseñado para poseer ambas opciones para energizar al mismo: usar una batería de cloruro de tionilo-litio o dos baterías de disulfuro de hierro. Esta opción, implementada en el circuito impreso es excluyente. Dependiendo de que componentes se suelden se puede tener una versión u otra. Esto se hizo así, más que nada porque es un prototipo y para tener las dos opciones respecto de la batería principal ( $2 \times 1,5\text{ V}$  o  $1 \times 3,6\text{ V}$ ). Eso permitirá a futuro evaluar cada opción, y además tener más grados de libertad para cuando se necesite certificar el producto.

<sup>7</sup>Como ejemplo, en un desarrollo realizado por el autor de este documento en 2006, se eligieron baterías de ( $\text{Li}/\text{SOCl}_2$ ) provenientes de china, sin especificación alguna. Durante la certificación de ese producto hubo un hallazgo que llevó a un rediseño del pack de baterías, en donde hubo que agregar limitación de corriente, y además empotrar en resina todo el conjunto

<sup>8</sup>Al utilizar el módulo xDot, no se tiene control sobre los circuitos que éste posee. Un trabajo futuro será quitar este módulo y discretizar el diseño del mismo eligiendo un microcontrolador y un transceptor LoRa, y el resto de componentes necesarios, pero incluyendo en esta modificación una mayor cantidad de llaves para sectorizar aún más al diseño. Por ejemplo, el transceptor LoRa, si se utiliza en Clase A, puede estar apagado durante el reposo y encenderse solo cuando se necesite transmitir.

Por ese motivo, la figura 3.7 muestra las llaves “x” que, verdaderamente, no son llaves físicas. Su “conexión” la define el nombre de la versión del ensamblado.

La llave “a” es la llave principal del sistema. Se cierra cuando al bloque de control le llega un pulso de otra parte del circuito<sup>9</sup>. El mecanismo esta provisto para que la batería no se drene cuando no esta en servicio (cuando el producto se empaca, y se deja en depósito, por ejemplo). Una vez cerrada la llave, no se abrirá hasta que el microcontrolador lo determine.

La llave “b” es la encargada de dar energía a todos los circuitos relacionados con los esclavos I<sup>2</sup>C del nodo maestro, en los que se encuentra la memoria externa. Estos circuitos se utilizan bajo demanda. En el caso de la memoria externa, esto ocurre cuando esté recibiendo datos para realizar una actualización de firmware. Eventualmente el *bootloader*, necesitará energizar esos circuitos para efectivizar la actualización.

La llave “c” permite dar alimentación al circuito de comunicación cableada que utiliza el protocolo RS485. Para este desarrollo, esa funcionalidad se utilizó como elemento de depuración del firmware (básicamente para enviar mensajes a un PC).

La llave “d” es una de las más importantes. Debido a que la red cableada posee veinticinco nodos, aun utilizando los microcontroladores de ellos en modo de ultra bajo consumo, el consumo del conjunto podría ser considerable. Para ello se instaló en el diseño la llave de tensión para alimentar a toda la red cableada de sensores, al momento de necesitarse. La carga de esta llave incluye también el circuito de protección para limitar la energía que “sale” del dispositivo hacia el exterior del gabinete.

La llave “e” es la encarga de poner tensión de batería en los circuitos analógicos que permiten medir la misma.

Todas las llaves se manejan con entradas y salidas de propósito general del módulo xDot.

#### **Regulación de voltaje:**

La figura 3.7 muestra que el producto posee, según el ensamblado que se realice, la opción de regular la tensión con un regulador lineal o con una fuente comunitada. Esto es para una fuente de energía de más de 3,3 V o para una de menos de ese valor, respectivamente.

Para el caso en donde la regulación de tensión es lineal, se eligió el MCP1812 [59] del fabricante Microchip. El diseño de este circuito puede observarse en la figura 3.8.

---

<sup>9</sup>WiAgro se reserva el derecho de publicar el detalle de esta implementación

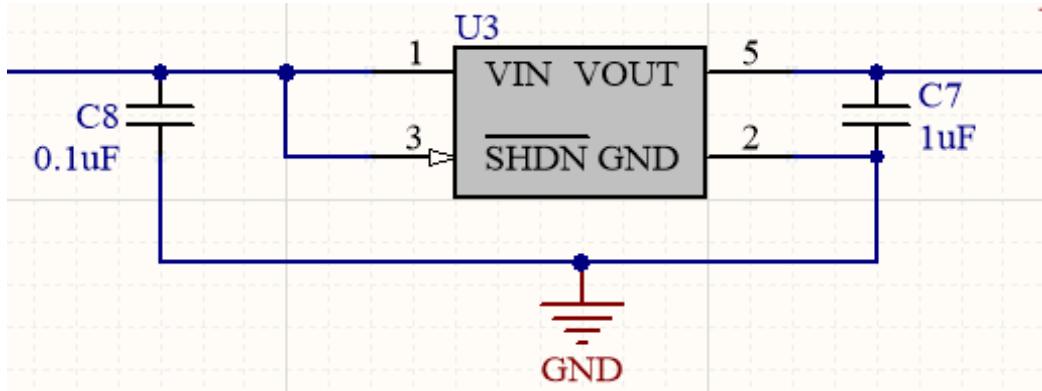


FIGURA 3.8: Circuito asociado al regulador lineal

Para la opción en donde la fuente principal de energía no supere los 3.3v, se eligió como fuente comutada al circuito integrado TPS613221A [60] del fabricante Texas Instruments. El diseño de la fuente se basó en un diagrama recomendado en la hoja de datos (el circuito integrado solamente necesita tres componentes externos). El circuito esquemático de este circuito puede observarse en la figura 3.9.

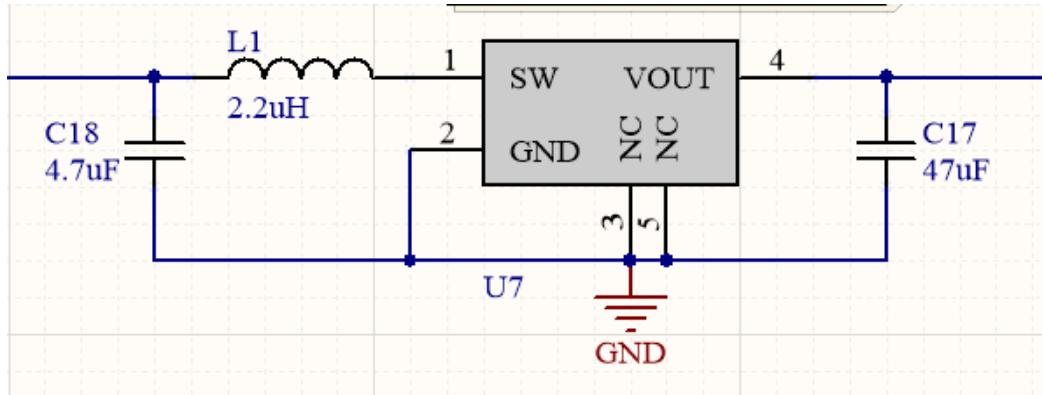


FIGURA 3.9: Circuito asociado al regulador comutado

Es una fuente que regula 3,3 V de un rango de tensión de entrada de 0,9 V a 5,5 V y permite regular 3,3 V, con una limitación de corriente en la llave de conmutación de 1,2 A. Al ser una fuente elevadora de tensión que permite a la entrada una o dos baterías de 1,5 V regulando a la salida 3,3 V. No obstante, la corriente máxima que puede obtenerse a la salida esta limitada por la siguiente ecuación:

$$I_O = V_I \cdot (I_{LIM} - I_{LH}/2) \cdot eff/V_O \quad (3.1)$$

donde:

- $I_O$  = máxima corriente de salida
- $V_I$  = tensión de entrada mínima
- $V_O$  = tensión de salida
- $I_{LIM}$  = límite de corriente en el inductor
- $I_{LH}$  = corriente de ripple en el inductor
- $eff$  = eficiencia de conversión

Se puede observar que depende de la máxima corriente que requiera el módulo xDot. Según la tabla 3.3, la corriente que requiere el xDot, para transmitir, es de 250 mA.

La tabla 3.4 muestra el cálculo de la máxima corriente que puede entregar la fuente comutada, para diferentes valores de tensión de entrada, tomando valores de las curvas de eficiencia. Se puede observar que la misma puede entregar tranquilamente la corriente requerida para transmitir.

TABLA 3.4: Cálculo de máxima corriente que puede entregar el regulador de commutación en función de la tensión de las baterías<sup>10</sup>.

$V_I$	EFF	$I_{OUX_{MAX}}$
1 V	0,6	720 mA
1,5 V	0,75	970 mA
2,5 V	0,86	1,35 A
3 V	0,92	1,55 A

Otro punto importante en haber elegido esta fuente se basó en su parámetro  $I_q$ . Una vez encendido el sistema, al cerrarse la llave “a”, esa corriente de reposo drenará la batería aun cuando no exista consumo (por ejemplo, aun cuando el microcontrolador del xDot haya apagado todas las zonas, y se haya puesto en modo de ultra bajo consumo). Un último parámetro para elegir esta fuente fue la poca cantidad de componentes, como se puede observar en la figura 3.9.

Si se optara por la versión de ensamblaje para una tensión de batería superior a 3,3 V, entonces la fuente comutada no tendría ventajas sobre la fuente lineal, ya que ésta posee mucha menos  $I_q$  que la mencionada primeramente. La  $I_q$  del TPS61321 es de 6,5  $\mu$ A (10  $\mu$ A en peor caso) y la  $I_q$  del MCP1812AT-033 de 250nA (500 nA en peor caso). Por otro lado, al no tener circuitos comutados, las características de ruido son mucho mejores.

#### Barrera zener:

El cliente LoRaWAN posee tres interfaces eléctricas expuestas al exterior del gabinete: tensión de alimentación, línea de recepción y línea de transmisión de datos. A esas tres interfaces se le agregó una barrera zener para limitar la energía provista por dispositivo, al exterior. Para el diseño de las barreras zener, se consideraron siguientes los parámetros de diseño:

- Tensión de operación: 3,3 V
- Tensión máxima: 4,3 V
- Corriente máxima: 100 mA

El diagrama de la barrera implementada se puede observar en la figura 3.10. Se implementó con tres diodos zener (en vez de uno) dado que se requiere un nivel de protección “ia” lo que implica una redundancia de dos elementos.

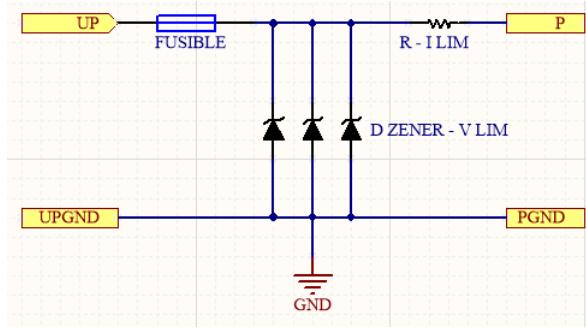


FIGURA 3.10: Circuito de barrera zener implementado

Considerando el diagrama esquemático y los parámetros de diseño, se realizó un cálculo de las características necesarias para los componentes que componen la barrera utilizando como referencia el método descrito en el documento [61].

En primer lugar, el valor de tensión zener  $V_Z$  se eligió por arriba del valor de tensión de operación. Esos circuitos trabajan con tensiones de 3,3 V, por lo tanto el valor zener elegido fue 4,3 V (existen otros valores entre 3,3 V y 4,3 V, pero al aumentar el valor se buscó reducir la corriente de perdida de esos elementos de protección). Esta tensión, en la jerga del estándar se denomina  $U_N$ .

Con ese valor se puede calcular el valor del resistor de limitación de corriente como sigue:

$$R_L = V_Z / I_O = 4,3V / 100mA \quad (3.2)$$

$R_L$  = resistencia limitadora de corriente

$V_Z$  = tensión nominal del diodo zener

$I_O$  = máxima corriente de salida de la interfaz

A partir de éste primer calculo, se puede escalar potencia máxima que deberá soportar el resistor. Esto se realiza a partir de un corto circuito a la salida.

$$P_{RL} = I_O \cdot U_N \cdot 1,05 \cdot 1,5 \quad (3.3)$$

$P_{RL}$  = potencia máxima de la resistor  $R_L$

$U_N$  = tensión máxima de salida de la interfaz

1,05 = tolerancia en el valor de  $V_Z$

1,5 = factor de seguridad para el nivel de protección

Además, se calculó la potencia máxima que podía disipar el zener, según la ecuación 3.4. Este cálculo se realiza para el caso de que el diodo zener se encuentre limitando, conduciendo una corriente 1,7 veces mayor que la de funcionamiento del fusible. Para el diseño elegimos un fusible con corriente nominal de 125 mA.

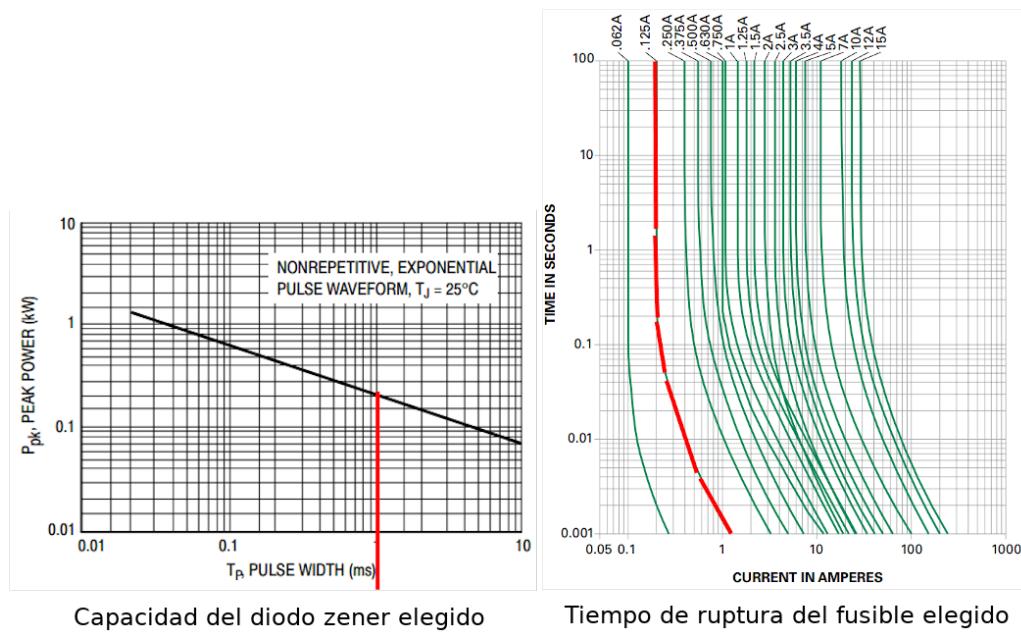
$$P_Z = U_N \cdot 1,05 \cdot 0,125 \cdot 1,7 \cdot 1,5 \quad (3.4)$$

TABLA 3.5: Resultados de los cálculos de componentes para barrera zener

Parámetro	Valor	Unidad	Ecuación
$R_L$	43	$\Omega$	3.2
$P_{RL}$	680	mW	3.3
$P_Z$	1,4	W	3.3

Además de los valores estáticos calculados arriba, se requiere realizar cálculos relacionados con los transitorios temporales. Esto es, cuando del lado “peligroso” hay una tensión que supera a la de la tensión de zener. En ese caso, el único elemento que limitará la corriente en el grupo de diodos, será la resistencia interna del fusible. En el cálculo hay que validar, para los componentes elegidos, que el fusible se abra antes de que se quemen los diodos. Si no fuera así, los diodos dejarían de limitar la tensión, y llegaría a la zona segura la tensión de falla. Verdaderamente este paso no es un cálculo sino una validación de los componentes elegidos. Se utilizan las curvas presentadas en las hojas de datos de los mismos y asegura que durante el tiempo que tarda el fusible en abrirse, el zener no se degradará.

Cuando la barrera, sin carga conectada a su salida, tiene aplicada una tensión a la entrada que supera la del zener, se activará el zener y comenzará circular una corriente limitada solamente por la resistencia del fusible (para el caso del fusible elegido es de  $1,7 \Omega$ ). La corriente crece muy rápidamente, así también como la potencia que debe disipar el diodo zener. En esa situación el fusible deberá actuar a tiempo, abriendose y eliminando el vínculo eléctrico entre la zona de “alta tensión” y la zona que debe resguardarse. La figura 3.11 muestra que para un tiempo de pulso de un segundo, en donde el zener todavía soporta la potencia disipada, el fusible se abrirá en menos de un segundo, cuando circule por él una corriente de 200 mA.



Capacidad del diodo zener elegido para soportar picos de potencia.

Tiempo de ruptura del fusible elegido

FIGURA 3.11: Curvas asociadas al diodo zener y al fusible elegido

En cuanto a la construcción física de las barreras, se cumplieron las líneas de fuga establecidas por la normativa. La figura 3.12 muestra cotas que se tuvieron en cuenta en diseño del circuito impreso.

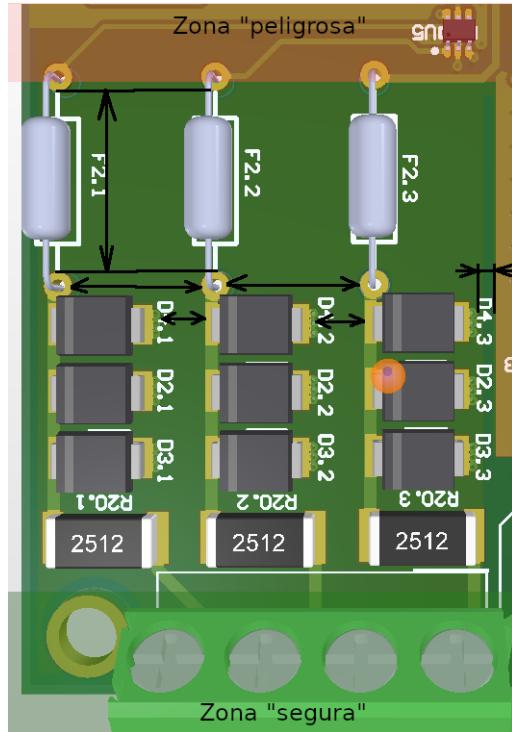


FIGURA 3.12: Diseño físico de la barrera zener. Todas las distancias entre zonas de cobre respetan las establecidas por el estándar.

Un aspecto importante a tener en cuenta, al insertar un circuito en las líneas de comunicación de la UART con niveles TTL, es la integridad de la señal. La inserción de las barreras zener, tanto en la línea de recepción como en la línea de transmisión y además el agregado de un cable de al menos un metro de longitud, hacen que ambas señales no sean ideales.

La barrera zener se comporta para cada una de las señales, como un filtro pasa bajos del tipo "RC". Al ser un circuito asimétrico en relación a cada una de las señales. El peor de los casos, es la señal de UART que emite el pin de transmisión del nodo esclavo, hacia el pin de recepción del maestro. Esto es porque en dicho sentido, el circuito RC con menor frecuencia de corte se forma entre la resistencia de limitación de corriente y la capacidad parásita de los diodos zener (ver figura 3.13).

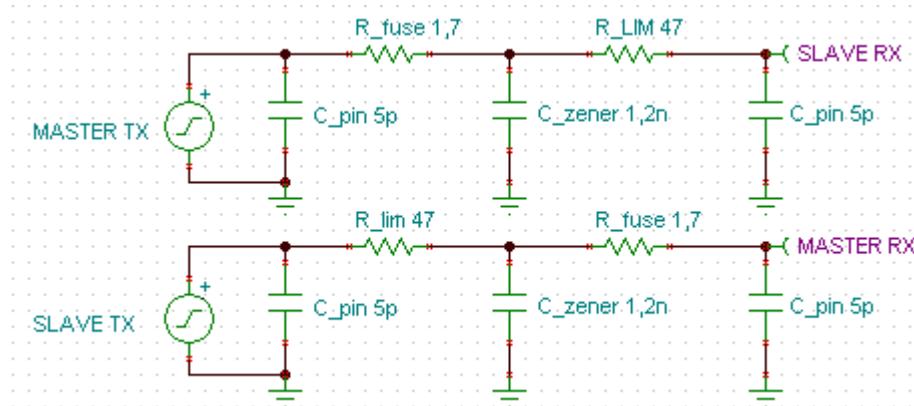


FIGURA 3.13: Modelo del cálculo para la distorsión introducida por la barrera zener.

El cálculo de la frecuencia de corte, se realiza según al ecuación 3.5.

$$f_{corte} = 1/(2 \cdot \pi \cdot R \cdot C) Hz \quad (3.5)$$

Si se aplican los valores de los componentes a dicha formula la frecuencia de corte es mayor que 2,5 MHz con lo cual no impondrá limitación en la tasa de bits máxima en el puerto serie.

#### **Limitación de capacidades e inductancias del sistema.**

Las tablas del anexo A de la ISO60079-11 limitan estos valores de capacidad máxima e inductancia máxima del sistema. Para el caso del dispositivo a desarrollar (grupo IIIC) se deberán adoptar las curvas usadas para el grupo IIB.

En cuanto a la capacidad máxima, para las tensiones que se manejan en el dispositivo (menores a 5 V), no habría restricciones (tanto las tablas como las curvas presentadas en el estándar se van fuera de escala). Para garantizar la seguridad, sin embargo, habrá que garantizar una inductancia equivalente serie de menos de  $30 \mu H$  en toda la cadena de sensores (en gráfico A-4, mostrado abajo)

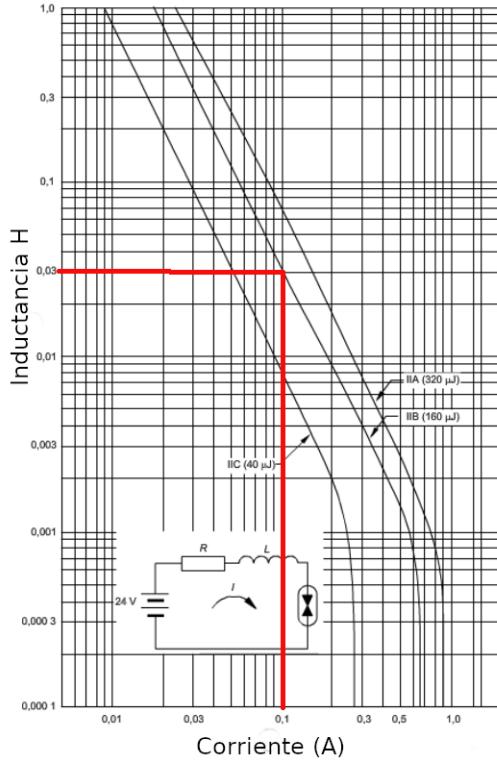


FIGURA 3.14: Gráfico que determina la inductancia máxima del sistema (A-3 del estándar)

#### Antena:

Para implementar la interfaz de radiofrecuencia, se optó por utilizar el conector UFL del módulo xDot y montar sobre el gabinete el conector SMA para panel, con la línea de transmisión adecuada que llegue al conector mencionado (figura 3.15). Este adaptador, requiere que se practique un agujero al gabinete. Dado que el gabinete deberá mantener el grado IP, deberá asegurarse de elegir un conector SMA que posea un *oring*, o algún otro medio para mantener la estanqueidad.



FIGURA 3.15: Adaptador UFL a SMA<sup>11</sup>.

### Conectores:

Los requisitos expuestos en la sección 2.5.2 implican utilizar conectores estancos entre los subnodos del sistema. Considerando que los conectores que cumplen con esos requisitos son de costo elevado y a su vez, que el equipo a diseñar posee cincuenta y uno (uno para el nodo maestro y dos por cada subnodo), está la opción es inviable. En ese caso se utilizará prensacables. Estos elementos no fueron elegidos para el presente trabajo.

### Soporte mecánico

La realización del trabajo no se incluyó el diseño del soporte mecánico para amarrar cada arreglo de nodos, en los diferentes sitios de los silos o de los contenedores de los transportes fluviales.

### Comunicación vía RS485:

Este agregado se realizó, fuera de especificación y alcance, para tener una opción más en el sistema y permitir una primera iteración experimental para expandir la funcionalidad del dispositivo a futuro.

Para ello que se eligió como transceptor de RS485 al ISO1500DBQ [63]. En este tipo de diseño, al tener dos regiones galvánicamente aisladas, posee dos zonas físicamente separadas y se requiere que en ambas se apliquen tensiones de alimentación de manera independiente. Las dos zonas son las “zona lógica” en donde se controla al dispositivo con niveles de tensión TTL y la otra es la “zona de bus” en donde se ubican las tensiones relacionadas con la capa física asociada con el protocolo RS485.

Se decidió que la alimentación de la zona del bus la provea el mismo circuito impreso del nodo maestro del sistema. Para ello, se tuvo que implementar una fuente conmutada para que desde la zona lógica se provea energía a la otra zona, mediante un transformador. Se utilizó una topología del tipo “convertidor push-pull”[64].

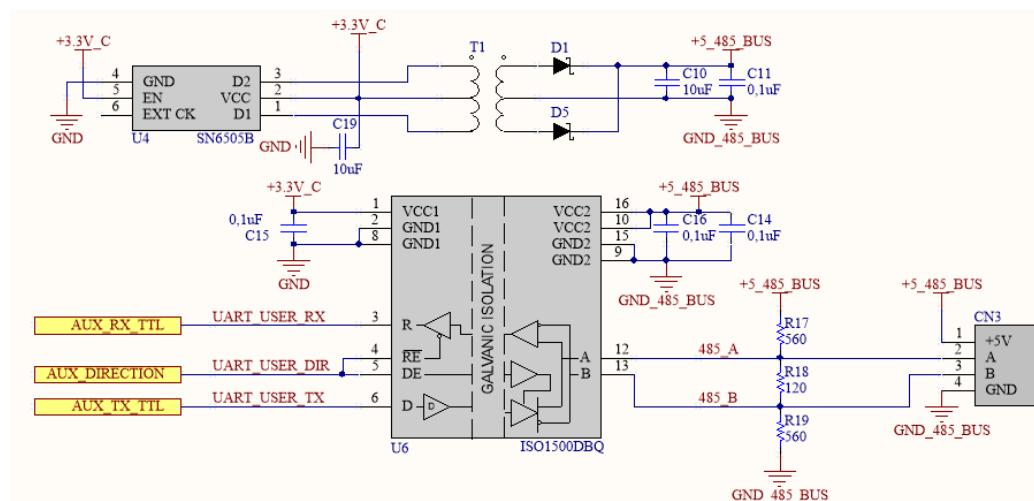


FIGURA 3.16: Diseño electrónico de la interfaz aislada RS 485.

<sup>11</sup>Fuente: [62]

Como se observa en la figura 3.16, el transformador, que es excitado con pulsos provenientes de un oscilador, genera a la salida pulsada, que es necesario rectificar. Es por ello que se coloca además un rectificador de onda completa y un filtro para generar el nivel de tensión continua requerido. Para evitar colocar mayor cantidad de componentes, se eligió una configuración tabulada de la hoja de datos del oscilador en donde no es necesario colocar un regulador lineal en la zona del bus<sup>12</sup>. Con esa configuración se lograría una salida (no regulada) de 5 V a partir de los 3,3 V del zona lógica. La tensión de alimentación la provee la llave "c" de la figura 3.7.

A nivel físico, para realizar el diseño del circuito impreso se tomaron las recomendaciones de la hoja de datos del ISO1500DBQ. Además, se consideraron los requisitos enmarcados en el estándar IEC60079-11 respecto de las correctas líneas de fuga y distancias en aire. La hoja de datos detalla que, para mantener la aislación, se deberá respetar la separación de 3,8 mm entre ambas zonas del circuito<sup>13</sup>. La figura 3.17 muestra una vista de como es el criterio de separación. Básicamente que existan distancias físicas mínimas entre los conductores de una zona y de otra. En la figura 3.18 se puede mostrar el trabajo realizado en cuanto a este aspecto. Por el lado de la IEC60079-11, se establece que para tensiones de menos de 10 V tanto la distancias en aire como las líneas de fuga deben ser de por lo menos 1,5 mm. Este valor es más chico que la otra restricción mencionada, por lo que se usó como cota inferior la separación recomendada por la hoja de datos del circuito integrado.

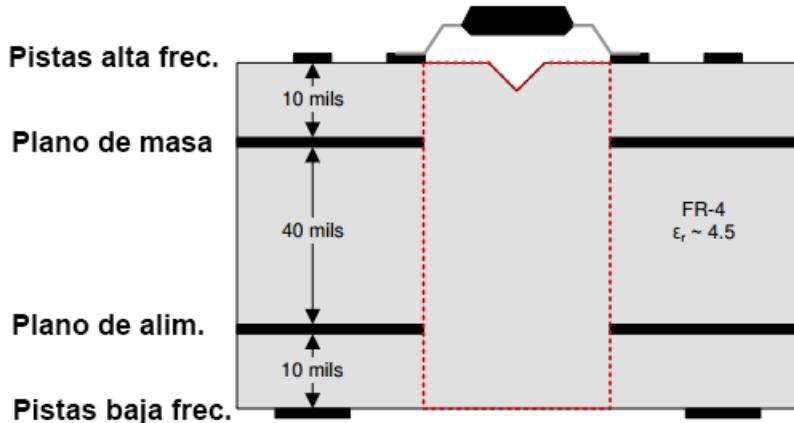


FIGURA 3.17: Perfil que muestra las distancias en un circuito impreso<sup>14</sup>.

<sup>12</sup>Tabla 3 de [65]

<sup>13</sup>Esta dimensión se calculó a partir del criterio detallado en la figura 3.17 y el *footprint* recomendado en la hoja de datos[63] para el circuito integrado

<sup>14</sup>Fuente: versión traducida de [63].

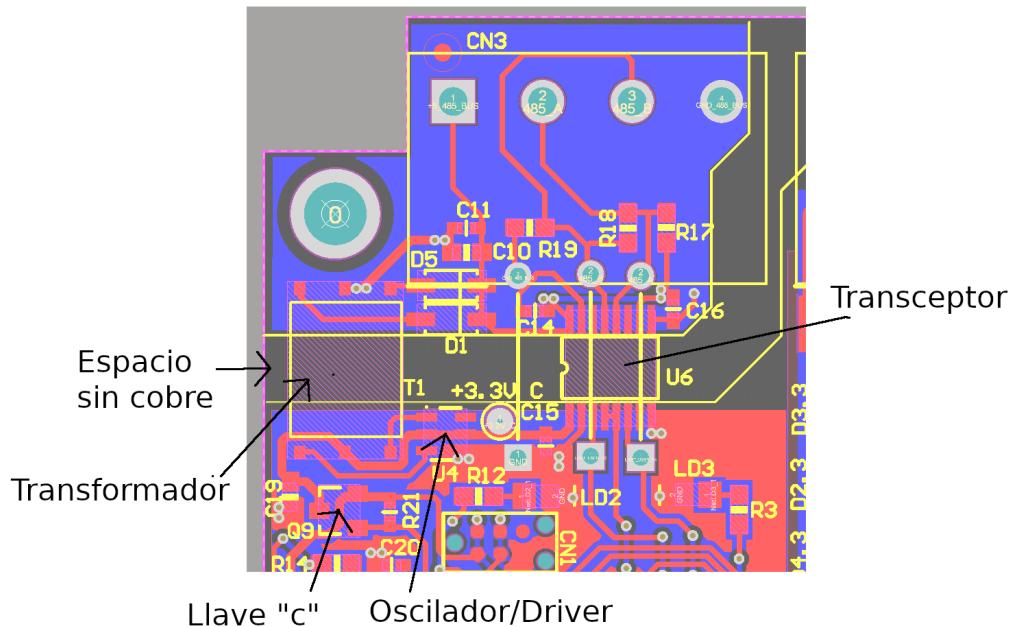


FIGURA 3.18: Detalle del diseño del circuito impreso de la interfaz RS485.

Cabe mencionar que las restricciones en las distancias de aire entre una zona y otra se contemplaron además en los espacios aéreos (no solo el cobre superficial del circuito impreso). Esta restricción obligó a separar los conectores CN3 y CN2 para que la distancia en aire, entre los tornillos de ajuste de los cables, superen 1,5 mm como se mencionó anteriormente. La figura 3.19 muestra la distancia entre las partes metálicas entre los conectores mencionados. Esta distancia no se puede apreciar en la misma, pero supera los 1,5 mm.

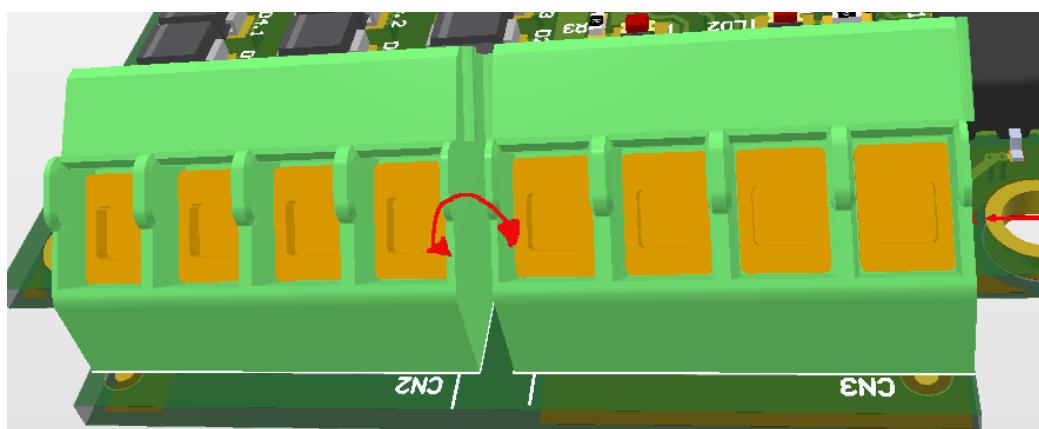


FIGURA 3.19: Ejemplo de distancia en aire para el conjunto de conectores CN2 y CN3.

### 3.3.3. Subnodos

Para el caso del circuito impreso para los subnodos, el circuito impreso fue diseñado en doble faz, con montaje de componentes de un solo lado.

Originalmente, se pensó al microcontrolador para los subnodos un microcontrolador de pocos recursos y muy económico. Si bien va a ser un criterio de decisión para la versión final del hardware, para los prototipos se utilizó un microcontrolador un poco excedido en recursos, principalmente en memoria flash. Esto se hizo así para evitar quedarse sin memoria durante el desarrollo<sup>15</sup>. El microcontrolador elegido además, por añadidura, posee muchos más recursos de los necesarios.

El microcontrolador elegido es el STM32L010R8T [66] y posee la mínima cantidad de periféricos requerida que es:

- 2 UARTS
- 1 I<sup>2</sup>C

Además de lo mencionado anteriormente respecto del sobredimensionamiento del microcontrolador, para este prototipo, el circuito impreso también se hizo mas grande de lo que realmente fue necesario, tal como muestra el *render* de la figura 3.20. Se diseñó así, para albergar los sensores referidos en la sección 2.3 y otros sensores (que no están colocados en el *render* 3D)<sup>16</sup> y además agregar un conector de 2 x 20 contactos, asignados a todos los pines libres que no se utilizaron del microcontrolador (para realizar interfaces con otros futuros circuitos a futuro).



FIGURA 3.20: Render del pcb ensamblado de un subnodo

En el caso del diseño de los subnodos, también se definieron zonas de tensión para habilitar y deshabilitar. Si bien la energía de todos los subnodos se conmuta

<sup>15</sup>El firmware del subnodo se basará en principio en el *framework* provisto por ST (STM32CUBE). Por experiencia personal del autor, esta biblioteca no está optimizada y es bastante pesada, para microcontroladores de bajos recursos de memoria.

<sup>16</sup>Los cuales no se detallan en la presente memoria, por pedido del cliente

según la voluntad del nodo maestro, se agregó esta característica para el caso en que el nodo maestro no pudiera apagar toda la red junta, un control local al nodo.

El diagrama en bloques de un subnodo de la red es el mostrado en la figura 3.21.

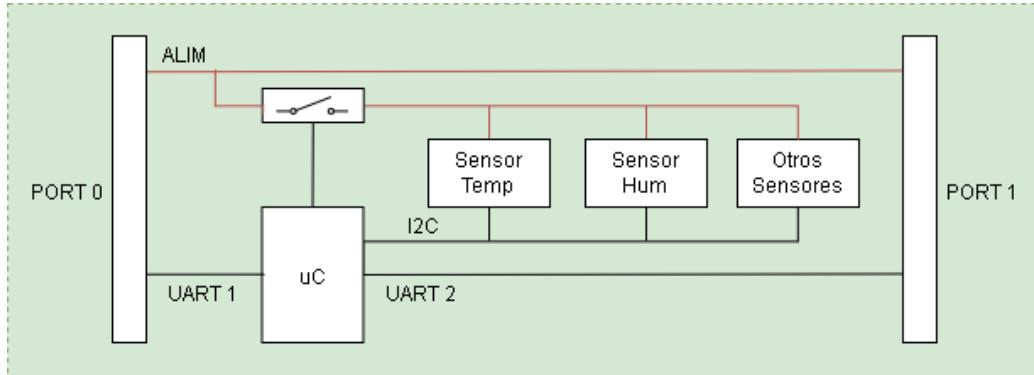


FIGURA 3.21: Diagrama en bloques de un subnodo

## 3.4. Diseño de firmware

Esta sección dará detalles del diseño del firmware de ambos subsistemas (cliente LoRaWAN y subnodos).

En un principio se intentó utilizar para el firmware de ambos proyectos el IDE STM32CubeIDE [67]. Este posee integrado todas las herramientas para trabajar con STM32. Dado que el microcontrolador de los subnodos es STM32 y el del módulo xDot también, en principio pareció una idea razonable. No obstante, como para el cliente LoRaWAN se decidió utilizar el *framework* Mbed, se prefirió utilizar para el firmware el IDE uVision de Keil (básicamente para llegar a término con el trabajo).

Con respecto a las arquitecturas de ambos firmware se optó por un diseño totalmente modularizado. En los siguientes secciones se comentarán las capas de abstracción utilizadas en este trabajo y se dará un detalle de la implementación de cada firmware en particular.

### 3.4.1. Bibliotecas comunes entre ambos firmware

En esta sección se comentarán aspectos generales de las bibliotecas de código desarrolladas para ambos firmware.

#### Biblioteca “Packet Framer”

Es una biblioteca de código que permite insertarse entre la capa de drivers de UART y una capa de procesamiento, por ejemplo. Abstacta la recepción de paquetes según las reglas del protocolo que se necesiten implementar. Se pueden definir diferentes maneras de separar paquetes recibidos y brindar al programador menos trabajo en este aspecto. La biblioteca permite entonces, una vez configurada, señalizar a la aplicación cuando llegan paquetes de datos a través de la uart, según los criterios elegidos.

La biblioteca de código posee muchas opciones de configuración, y permite adaptarse a cualquier arquitectura de firmware, y a cualquier plataforma<sup>17</sup>.

#### Biblioteca de temporizadores: “stim”

Es una pequeña biblioteca de código que permite gestionar timeouts, basando su lógica en un tick periódico. Le da soporte a la biblioteca de “Packet Framer” y a la biblioteca relacionada con el protocolo de red de subnodos.

Se implementó principalmente para el firmware baremetal de los subnodos, pero también se utilizó en el firmware del cliente LoRa, aun cuando el *framework* Mbed posee herramientas similares.

#### Biblioteca para protocolo de red de subnodos

Para la implementación del protocolo de red de subnodos se redactó un grupo de requerimientos generales que luego se plasmaron en definiciones concretas.

---

<sup>17</sup> Esta biblioteca la escribió el autor del presente trabajo, con anterioridad. Fue utilizada en varios proyectos con éxito, con diferentes arquitecturas y en distintas plataformas. No está publicada aún para que la utilice la comunidad pero se va a hacer pública bajo licencia BSD [68]

TABLA 3.6: Requerimientos generales del protocolo de red de subnodos

<b>Generales</b>	
0	La implementación deberá ser portable a nivel CPU como a nivel <i>framework</i> .
1	La implementación deberá permitir la compilación en C y/o C++.
2	El nodo maestro poseerá un puerto (denominado puerto 0) como puerta de salida y entrante de paquetes.
3	Los subnodos poseerán dos puertos. El puerto 0 para redireccionar comandos del maestro y recibir las respuesta de otros nodos y el puerto 1 para recibir comandos del maestro y enviar las respuestas al mismo, además que respuestas de otros nodos.
4	La capa física para el envío de mensajes entre los nodos sera una UART con niveles de tensión TTL.
5	El protocolo deberá permitir a los subnodos auto asignarse la dirección.
6	Los paquetes de datos se delimitarán por un tiempo de 10ms.
7	Los paquetes de datos poseerán un campo de comprobación que utilice el algoritmo LRC <sup>18</sup> .

Luego, para avanzar con la definición del protocolo, se definieron reglas sencillas con respecto al manejo de paquetes en los subnodos.

- Todo paquete que llegue al puerto 1 de un cierto nodo que sea broadcast, se procesará, se enviará la respuesta a través del puerto 1, y se reenviará al puerto 0 con la dirección incrementada en uno.
- Todo Paquete que llegue al puerto 1 de un cierto nodo, que posea la dirección de dicho nodo, se procesará y se enviará la respuesta a través del puerto 1.
- Todo paquete que llegue al puerto 1 de un cierto nodo, que posea una dirección que no es igual a la de dicho nodo, se reenviará a través del puerto 0.
- Todo paquete que llegue a través del puerto 0, se reenviará intacto al puerto 1.

Estas reglas permitieron definir el formato de las tramas del protocolo el cual se muestra en la figura 3.22.

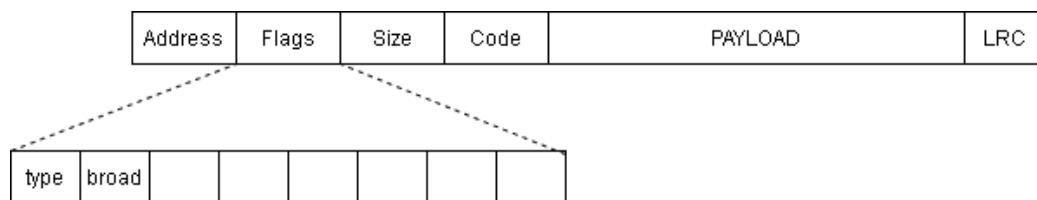


FIGURA 3.22: Formato de paquete para los comandos utilizados en el protocolo

<sup>18</sup>Irc son las siglas de Longitudinal Redundancy Check. A futuro se podría realizar una comprobación un poco más elaborada como un CRC16 o un CRC32, dado que los microcontroladores actuales poseen hardware que realizan esos cálculos, facilitando la implementación.

Solamente se definió el comando “Sense All”. Este comando, solicita a los subnodos que tomen mediciones en un cierto grupos de sensores. El comando identifica a los sensores solicitados mediante un campo de flags.

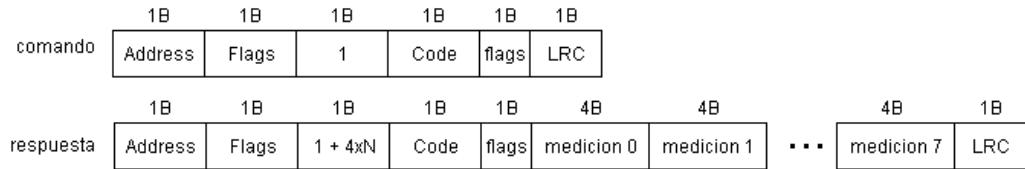


FIGURA 3.23: Estructura de comando y respuesta “SenseAll”

Los campos elegidos permite definir distintos tipos de comportamientos en los subnodos. Al inicializarse los subnodos no posee dirección. En ese estado, la única manera de asignarse una es cuando el maestro inicia las consultas a través de un comando broadcast. Al mandar un broadcast, la dirección asignada será 0 y el primer nodo se asignará la dirección 1. Al ser un comando broadcast, ese nodo deberá retransmitirlo, pero para ello modificará la dirección del paquete para que el siguiente se auto asigne también la dirección de manera correcta.

Con respecto a la implementación del maestro, se definieron las siguientes reglas:

- Al enviar un comando a un nodo con una dirección concreta, el maestro arrancará un temporizador y esperará por la respuesta durante ese tiempo.
- Al enviar un comando broadcast, el maestro arrancará un temporizador y esperará por una respuesta durante ese tiempo. AL recibir una respuesta, lo reiniciará dando la posibilidad de recepción de otra respuesta. Cuando finalice el tiempo, el maestro habrá considerado que la transacción finalizó, y procesara los datos de los nodos que contestaron al comando.
- Utilizará la dirección de las respuestas para conocer la ubicación física de cada subnodo.

Para completar lo antedicho y dejar claras las dinámicas del protocolo, se estableció el diagrama de secuencia mostrado en la figura 3.24, que muestra el comportamiento normal del protocolo, para un master y dos nodos.

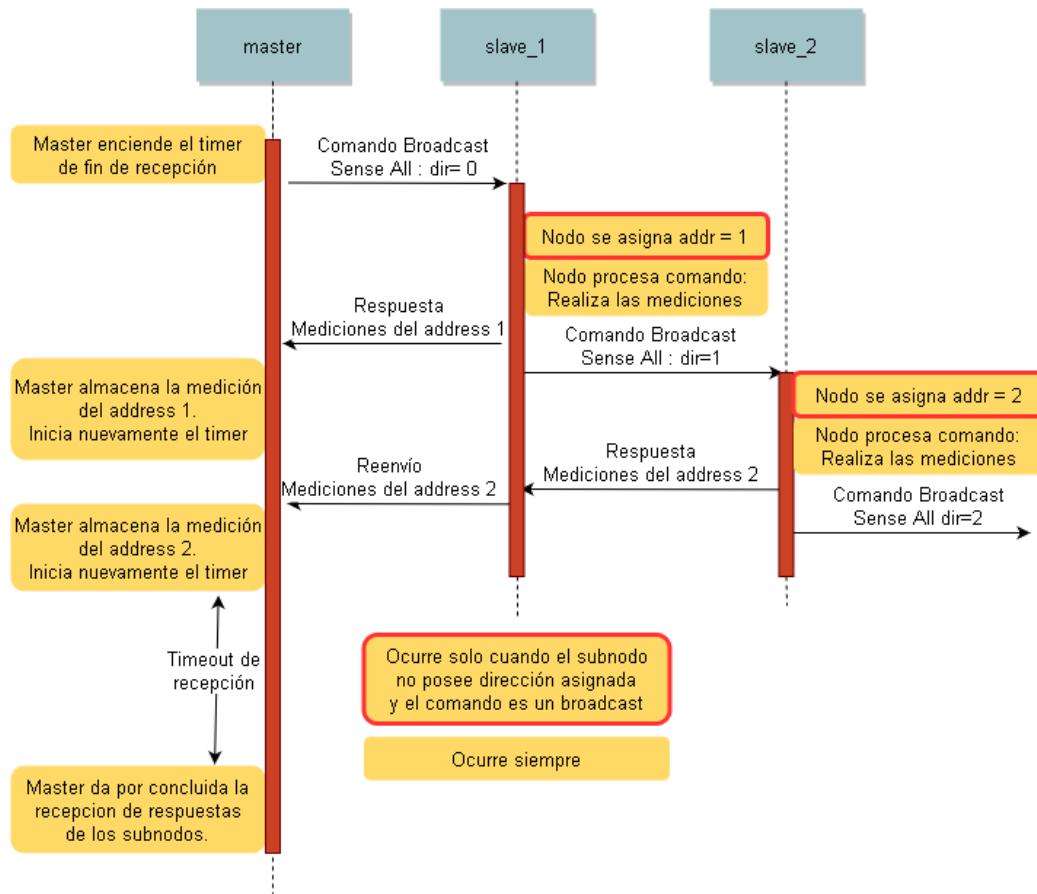


FIGURA 3.24: Diagrama de secuencia del protocolo de subnodos

### 3.4.2. Firmware del cliente

El firmware del cliente LoRaWAN se montó sobre el *framework* Mbed. Mbed es un gran *framework* que permite desarrollar firmware en diferentes entornos. Una manera es usar un software llamado MbedCli que automatiza la construcción de makefiles y la compilación. El problema en el uso del MbedCli es que posee una curva de aprendizaje bastante abrupta, y por otro lado, a criterio del autor de este trabajo, esta bastante mal documentado para realizar proyectos grandes. Si tiene una ventaja significativa cuando se quiere ejecutar un código de ejemplo para aprender a utilizar el *framework*, al sistema operativo o a las bibliotecas de código que trae. Por eso, se prefirió no utilizarlo.

Otra opción, que es la que se prefirió, es la de exportar los proyectos de ejemplo utilizando el IDE online que Mbed provee, a distintos entornos de desarrollo. La exportación a un entorno de desarrollo conocido, hizo que se pudiera comenzar a desarrollar de manera más ágil. En este caso, los exportadores que estuvieron disponibles para los ejemplos que utilizaban las librerías LoRaWAN no incluían STM32CubeIDE. Por otro lado, se intentó exportar a otros entornos basados en eclipse sin buenos resultados. Es por eso que se exportaron los ejemplos a uVision y se arrancó utilizando esa herramienta<sup>19</sup>.

<sup>19</sup>Se dedicaron muchas horas de trabajo para unificar herramientas, pero no se llegó a buen puerto, por lo que se prefirió avanzar rápido de ésta manera, dejando para un trabajo posterior la unificación.

El desarrollo del firmware para el cliente LoRaWAN se basó en un ejemplo para un nodo LoRaWAN clase A. La arquitectura del ejemplo utilizado queató perfectamente en las necesidades de este proyecto. Es un firmware basado en una sola tarea del sistema operativo la cual actúa como despachador de eventos. Las librerías de red de LoRaWAN están diseñadas para funcionar con dicho despachador por lo tanto la tarea principal del sistema consiste simplemente en configurarlas y llamar al despachador para que vaya ejecutando todos los eventos que ésta proporciona.

La funcionalidad propia del presente trabajo se montó sobre misma arquitectura y se aprovechó el despachador de eventos (módulo EventHandler) para realizar todas las tareas de consulta a la red de subnodos y de recepción de respuestas. Para ello se utilizaron algunas herramientas que provee el *framework* (por ejemplo el módulo Ticker, para la biblioteca de temporizadores). Dado que es una aplicación que requiere ejecutarse con un ciclo de trabajo muy bajo, la tarea principal posee un bucle infinito en donde la última acción es irse a dormir por un período de tiempo determinado. Al despertarse posterior al período, las acciones se repiten.

La figura 3.25 muestra el diagrama de flujos de la ejecución de dicha tarea. En la misma, se puede observar que en todas las acciones relacionadas con el despachado de eventos (que básicamente lo que hace es gestionar las librerías de LoRaWAN y las transacciones con la red de subnodos) se evalúa si se tiene que finalizar el ciclo. Esto no es una bifurcación lógica del programa en si. Es simplemente para explicitar que el despachador de eventos eventualmente finaliza sus acciones, y da lugar a la pausa de muy bajo consumo.

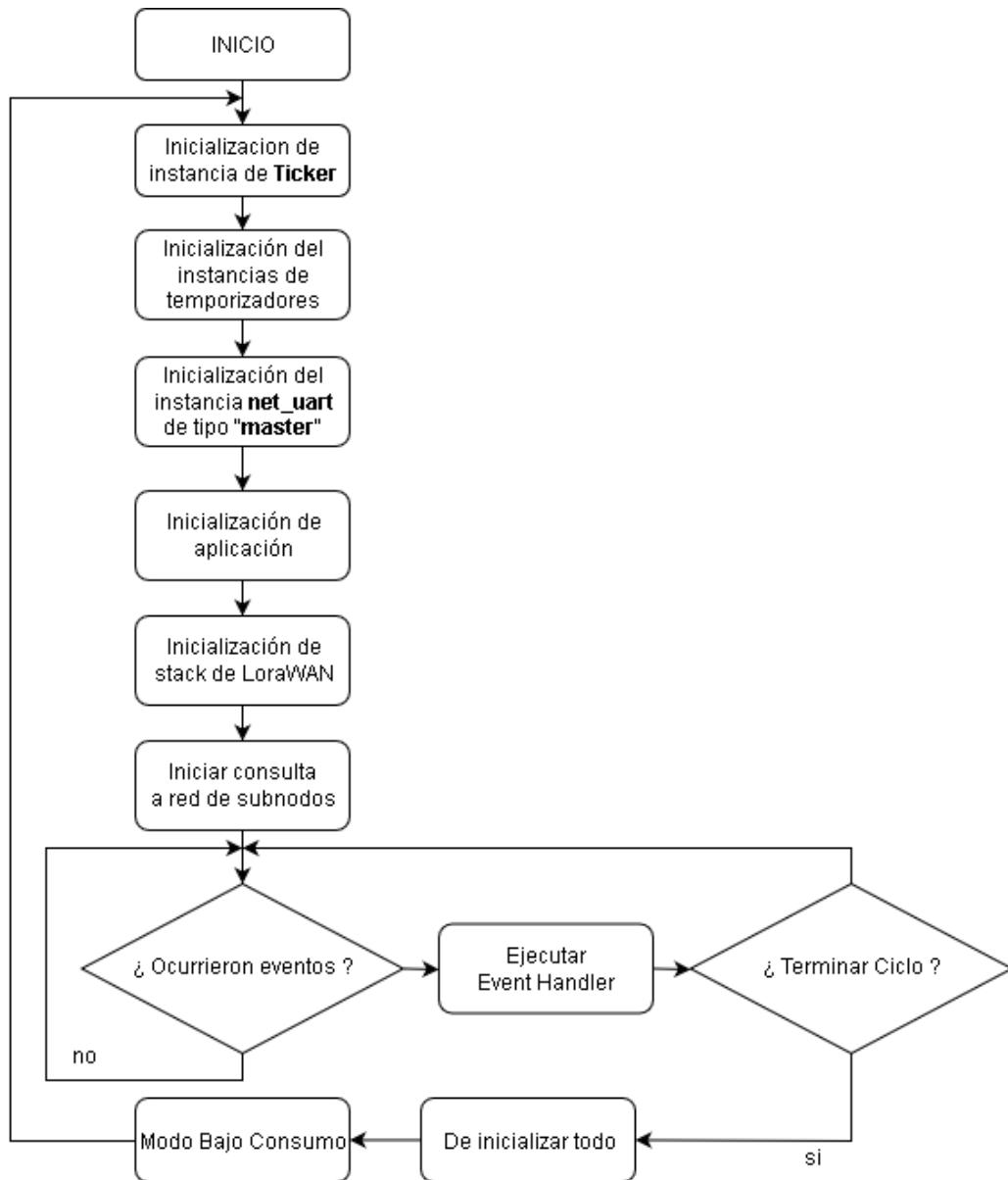


FIGURA 3.25: Diagrama de flujos de principal del cliente LoRaWAN

### 3.4.3. Diseño de protocolo montado sobre LoRA servidor-cliente

Sobre LoRaWAN se definieron:

- Mensajes de mediciones: se definió el formato del paquete que llevará las mediciones de todos los subnodos.
- Comandos de configuración: se definieron algunos comandos de configuración que permita modificar algunos parámetros de funcionamiento de cada cliente.
- Protocolo de actualización de firmware: comandos y mensajes relacionados con el protocolo de actualización de firmware remoto.

### Generales

Todos los comandos y mensajes de este protocolo poseerán un encabezado pero no poseerán campos de comprobación, dado que el protocolo LoRaWAN ya lo incluye en sus capas de abstracción.

El encabezado es muy sencillo y se muestra en la figura 3.26.

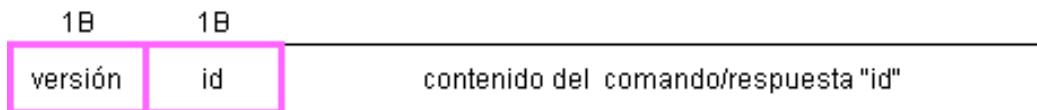


FIGURA 3.26: Estructura del mensaje/comando genérico

- Versión: campo que indica la versión del protocolo.
- Comando: este campo indicará el comando que se está enviando. La tabla 3.7 muestra los comandos implementados. Si es una respuesta o un mensaje, identificará al mismo.

TABLA 3.7: Comandos para el protocolo de aplicación

Comando	Id	Descripción
SEND_MEASUREMENT_CODE	0x01	Mensaje que envía el nodo maestro con las mediciones realizadas.
SET_PARAMETERS_CODE	0x02	Comando que recibe un Cliente LoRa con diversas configuraciones.
CMD_FU_ADD	0x03	Comando de actualización de firmware que transporta una porción de la nueva imagen.
CMD_FU_CANCEL	0x04	Comando de actualización de firmware que cancela cualquier porción de imagen recibida hasta ese momento.
CMD_FU_END	0x05	Comando de actualización de firmware que indica que no hay más porciones y que el dispositivo debe actualizar su firmware con la imagen recibida.

#### Mensajes de mediciones (SEND\_MEASUREMENT\_CODE)

El mensaje de mediciones es un mensaje que contendrá un grupo de mediciones de los subnodos. Estará formado por un encabezado y un arreglo de subpaquetes, en donde estarán las mediciones en sí. Cada subpaquete (relacionado con un subnodo) poseerá hasta ocho mediciones, y cada una estará identificada por un flag binario. Esto hace que un subpaquete pueda tener entre una y ocho mediciones. El subpaquete, posee un campo que indica qué mediciones posee. Este mensaje es de tamaño variable. Todo el paquete de mediciones de un subnodo, por lo tanto, es de tamaño variable y puede alcanzar en un peor caso una cantidad de bytes que superan al máximo permitido por el protocolo LoRaWAN. Por ello, el formato de los paquetes se diseñó para que las mediciones pudieran enviarse en varios paquetes más pequeños.

La figura 3.27 muestra el formato de una mensaje genérico.

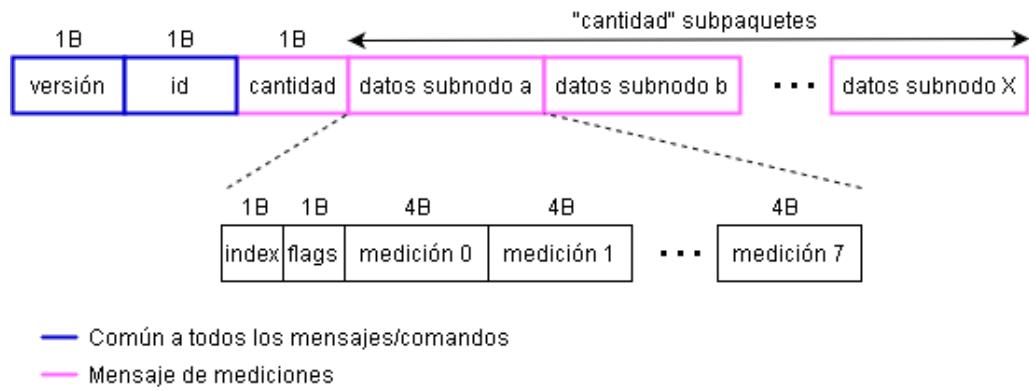


FIGURA 3.27: Estructura del mensaje de mediciones

### Comandos de configuración (SET\_PARAMETERS\_CODE)

Se implementó solo un comando de configuración que permita modificar varios parámetros a la vez. La idea detrás de esto es que el cliente, al estar configurado en “LoRaWAN clase A” en funcionamiento normal, utilice la ventana de recepción (posterior a una transmisión) para recibir el comando.

Se definieron comandos de configuración de parámetros según:

TABLA 3.8: Subcomandos para comando de configuración de parámetros

Parámetro	Descripción del <i>payload</i>	SubId	Tipo dato
Application Port	Puerto de la aplicación	1	entero de 16 bits
Application Key	Llave de seguridad de la aplicación	2	arreglo de 16 bytes
Application EUI	Identificador único del servidor	3	arreglo de 8 bytes
Device EUI	Identificador único del nodo	4	arreglo de 8 bytes
Período de reporte	Tiempo entre muestreos de los valores analógicos. Esto determinará el tiempo que el dispositivo este dormido consumiendo la menor energía posible.	5	entero de 32 bits

La estructura de la trama de un comando de configuración se puede observar en la figura 3.28.

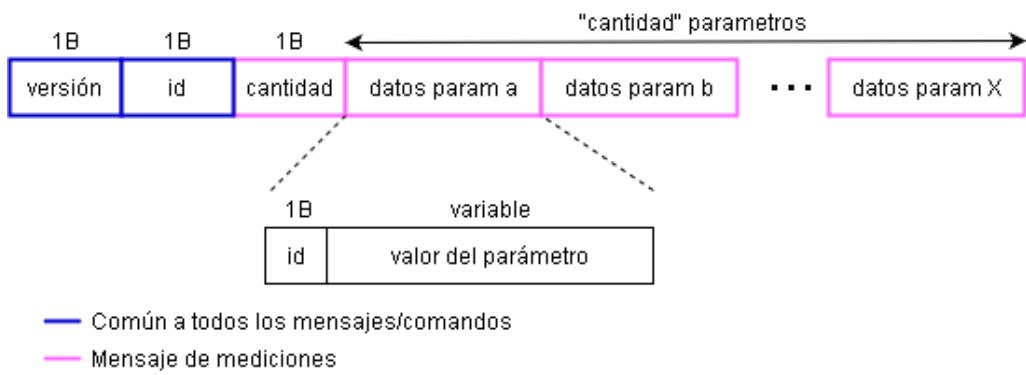


FIGURA 3.28: Estructura del comandos para establecer parámetros

#### Protocolo de actualización de firmware (CMD\_FU\_ADD, CMD\_FU\_CANCEL Y CMD\_FU\_END)

El protocolo de actualización de firmware es el protocolo de intercambio de paquetes con la finalidad de que se puede actualizar el firmware del Cliente LoRaWAN. La actualización de firmware solo aplica al nodo maestro del arreglo de subnodos.

El diseño de este protocolo se basó en la decisión de colocar una memoria *eeprom* externa para almacenar la imagen mientras el sistema sigue funcionando. Se eligió una memoria *eeprom* de 128kB. Como el microcontrolador posee 256kB de memoria flash, se decidió que el protocolo no actualizará toda la imagen, sino que permitirá actualizar una parte de la misma (Aplicación y librerías de red de subnodos, quedando fuera la actualización del sistema operativo, sus herramientas y las librerías de LoRaWAN).

El protocolo tiene premisas simples:

- El servidor, envía fragmentos de la imagen a reemplazar de una cierta cantidad de bytes.
- Cada vez que llega un fragmento, se guardará en una posición relativa al total de la imagen, y se marcará en un mapa un flag, que indicará que ese fragmento es válido.
- El cliente, al recibir un fragmento, enviará al servidor una confirmación de recepción.
- El servidor, al finalizar de enviar todos los fragmentos, enviará un paquete con un valor de comprobación (CRC16 por ejemplo) para que el cliente corrobore que todo el contenido de la *eeprom* es válido. De ser así, el cliente reiniciará y el *bootloader* se encargará de copiar la imagen de la *eeprom* externa a la flash del procesador principal.

Para realizar esto, se definieron los siguientes comandos y respuestas:

- **Agregar datos:** este comando enviará un fragmento de los datos de la imagen del nuevo firmware.
- **Finalizar:** indicará al algoritmo que se terminaron de enviar todos los fragmentos.
- **Cancelar:** indicará al algoritmo que el envío se cancelará.

La estructura de las tramas para cada comando se puede observar en la figura 3.29. No se muestra la estructura del comando Cancelar, porque simplemente es una trama de la aplicación sin *payload* (es decir, tendrá solo campo versión y id).

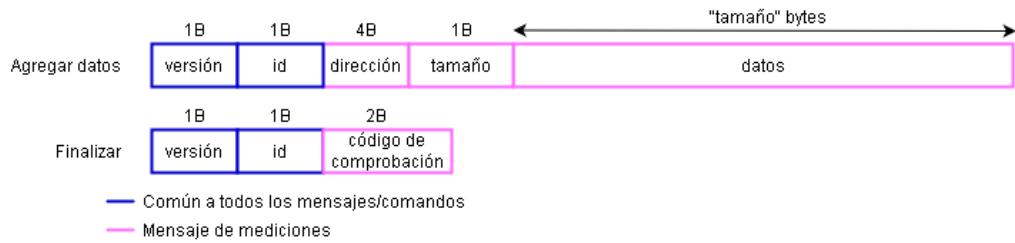


FIGURA 3.29: Estructura de las tramas de “Aregar” datos y de “Finalizar”

El algoritmo de actualización de firmware esta compuesto por dos etapas. La primera consta de recibir todas las tramas que componen la imagen. Para finalizar el servidor deberá enviar la trama de finalización, que contiene el código de comprobación de toda la imagen recibida. En ese momento, si la imagen es correcta, se procederá a frenar el sistema<sup>20</sup>, pisar la zona de flash a actualizar, y reiniciar el sistema.

La figura 3.30 muestra el diagrama de secuencia del procedimiento.

<sup>20</sup>Como todavía esta parte no se integró en el microcontrolador, la acción de “frenar el sistema” podrá ser simplemente finalizar la tarea principal, a través de alguna tarea de servicio, pisar la imagen y reiniciar la tarea con comportamiento actualizado, o simplemente reiniciar el sistema y dejar la responsabilidad de la actualización de la flash al *bootloader* y a pisar la zona de flash del microcontrolador, con la imagen en la *eeprom*.

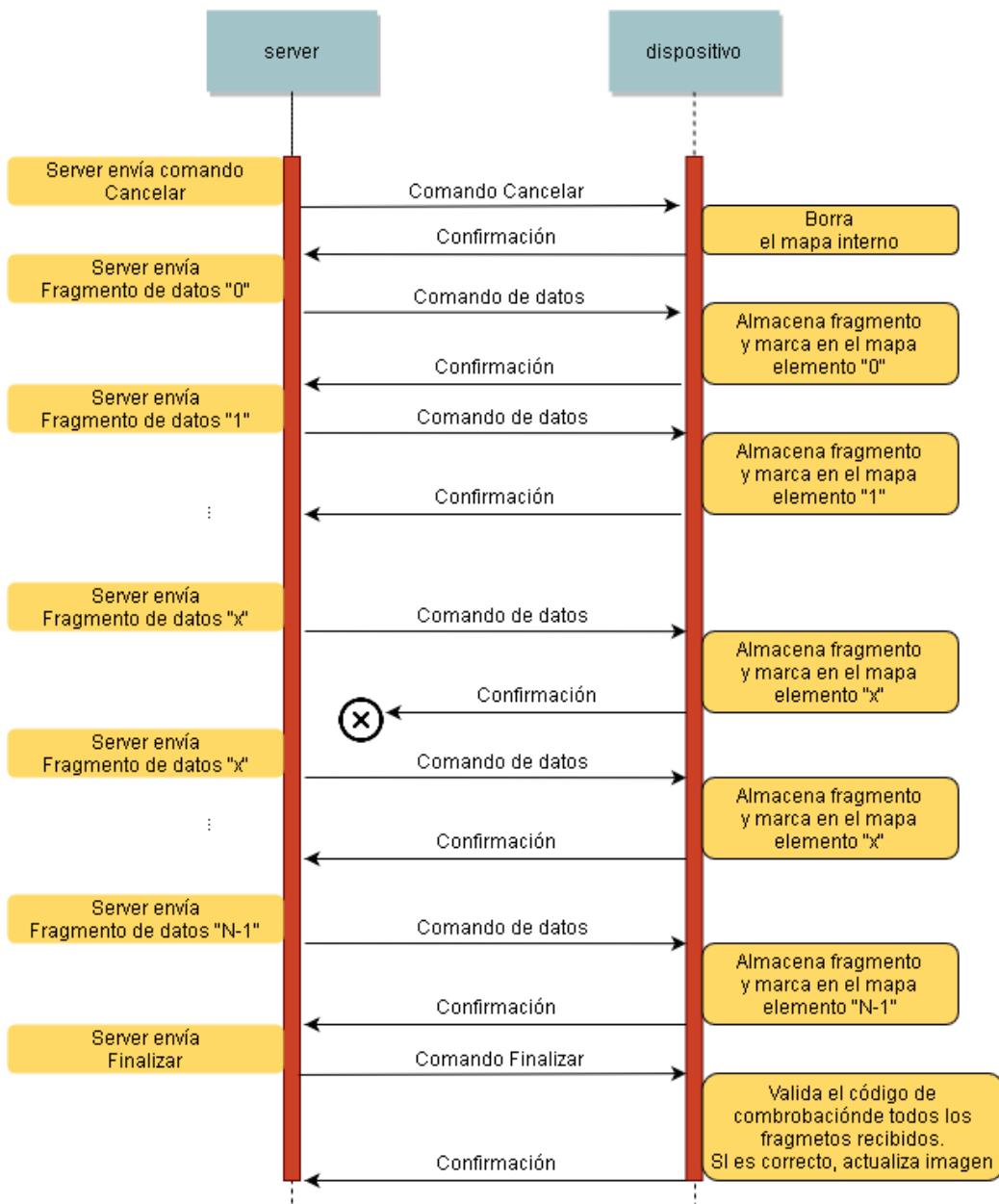


FIGURA 3.30: Diagrama de secuencia del algoritmo de firmware update

### 3.4.4. Firmware de los subnodos

El firmware de los subodos, se implementó empleando programación baremetal. No obstante se agregaron elementos de procesamiento asincrónico para la implementación del protocolo de subnodos (por ejemplo, empleando el la biblioteca Packet Framer). Es por eso que el diagrama de flujo principal, además de inicializar todos los módulos, lo único que realiza es evaluar eventos. Esto se logra delegando todo el procesamiento a las rutinas de interrupción a través de los drivers, y otras bibliotecas utilizadas, incluyendo las librerías para la red de subnodos. No se implementó un despachador de eventos. En reemplazo se utilizó simplemente señalización mediante banderas. La figura 3.31 muestra el diagrama de flujo principal del firwmare.

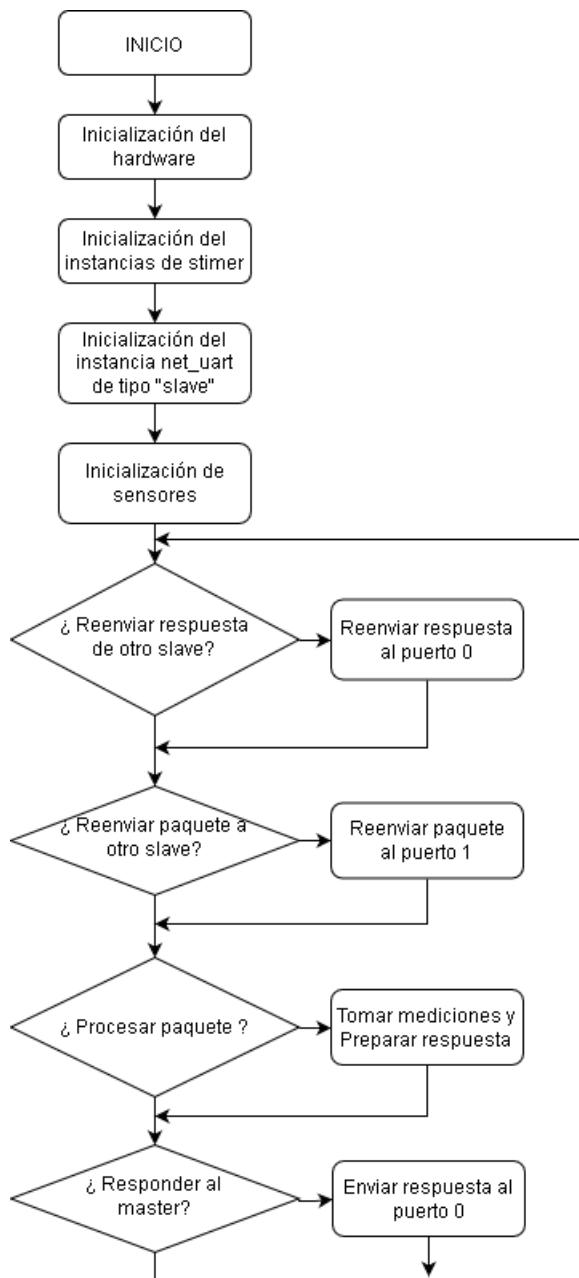


FIGURA 3.31: Diagrama de flujo principal del firmware de los subnodos

### Drivers

Para lograr una implementación ágil del firmware, se utilizaron los drivers provistos por ST, llamados STM32Cube [69].

En la figura 3.32 se muestra un esquema de capas de abstracción en el que se incluyen los siguientes módulos:

- Driver de STH31: se portó el driver de STH32 [70] provisto por AdaFruit[71] desarrollado para Arduino, utilizando los drivers del STM32Cube.
- Packet Framer: se configuró la librería para usar el driver de UART que provee el paquete de drivers mencionado.
- stim: se configuró la librería para utilizar el tick periódico provisto por el STM32Cube.

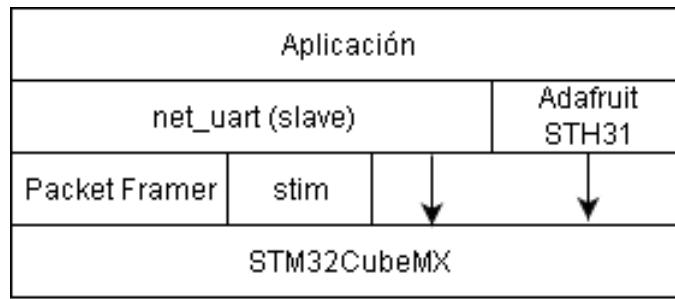


FIGURA 3.32: Capas de abstracción del firmware de los subnodos.



## Capítulo 4

# Ensayos y Resultados

En este capítulo se muestra el resultado de la fabricación del prototipo de los circuitos impresos.

Por otro lado, se describen los resultados de las pruebas sobre las bibliotecas de software desarrolladas.

Además, Se detallan los ensayos realizados para comprobar el correcto funcionamiento del hardware y del firmware de los módulos que componen el sistema.

### 4.1. Hardware

Los prototipos de los circuitos impresos no llegaron a tiempo para ser utilizados para la realización de este trabajo, pero aun así el fabricante envió fotografías de los circuitos impresos ensamblados. De haberlos tenido disponibles, además de desarrollar el firmware en base al diseño prototipado se hubieran realizado mediciones sobre los mismos para exponer en esta sección de esta memoria.

La figura 4.1 y la figura 4.2 muestran las diferencias entre las versiones renderizadas y las fotografías obtenidas de los prototipos.

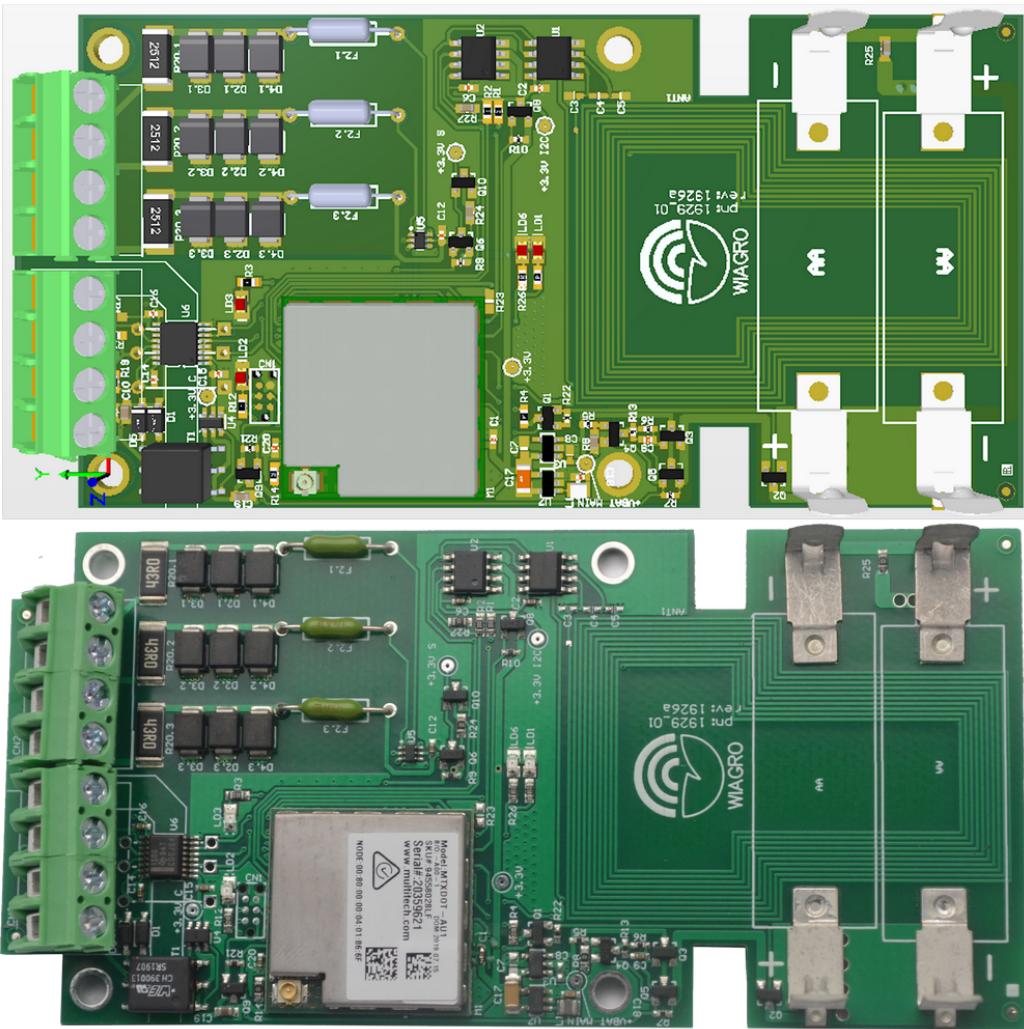


FIGURA 4.1: Comparación del prototipo fabricado del nodo maestro versus el render 3D diseñado.

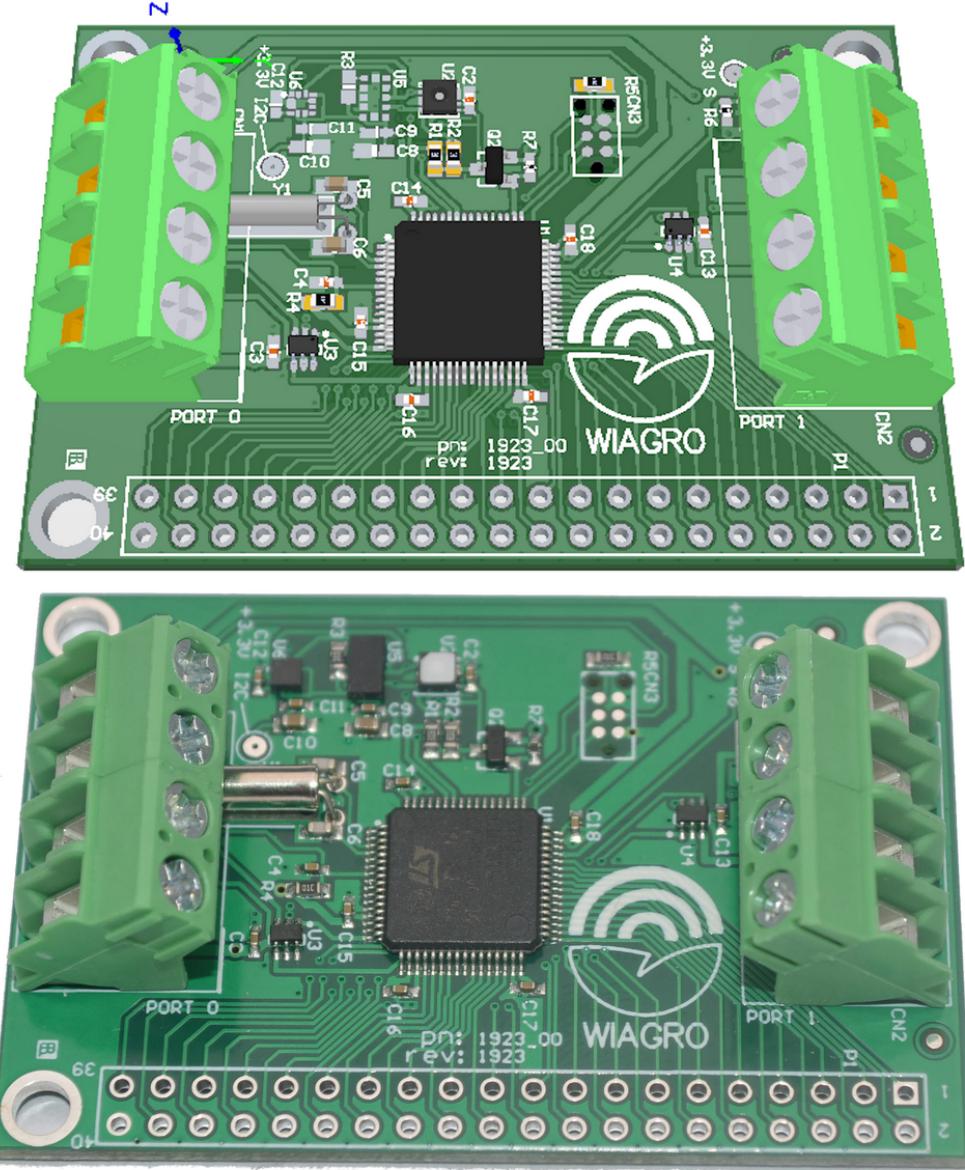


FIGURA 4.2: Comparación del prototipo fabricado del subnodo versus el render 3D diseñado.

Para el caso del circuito impreso del nodo maestro, se puede observar que se montaron todos los componentes. No obstante, la versión final solo tendría que tener montados los componentes relacionados con la fuente por conmutación (y no la fuente lineal). Se hizo de esta manera, para poder probar ambas opciones (durante las pruebas de hardware, se quitaran los componentes que no deban ir de los prototipos).

Por otro lado, se montaron también los LEDs que, como se explicó anteriormente, no irán en la versión final.

Una funcionalidad del circuito impreso no comentada en la sección de diseño, es la relacionada con el inductor impreso en cobre que puede observarse en la figura 4.2. No se realizó porque es una funcionalidad que WiAgro prefiere mantener en secreto.

Para el caso del circuito impreso de un subnodo, se pueden observar diferencias tales como:

- Algunos componentes se montaron en el prototipo pero en la versión renderizada no están. Esto es porque el prototipo incluye todos los sensores (además del de temperatura y humedad) solicitados por WiAgro.
- El sensor STH31 que fué montado en el prototipo, posee la membrana protectora, y en el render 3d no está.

#### 4.1.1. Mediciones

Una medición que es importante que se realizó fue la de evaluar la integridad de la señal de TX y RX de las USARTs para el caso de la red de subnodos, en donde se utilizan niveles de tensión TTL directamente provenientes del microcontrolador. Estas señales, poseen una línea de transmisión de un metro entre transmisor y receptor y además posee una barrera zener “en el camino”. La barrera zener equivale a un filtro en donde las resistencias involucradas (de los cables y las resistencias físicas de las barreras). En la figura 4.3 se puede observar el camino de señal mencionado.

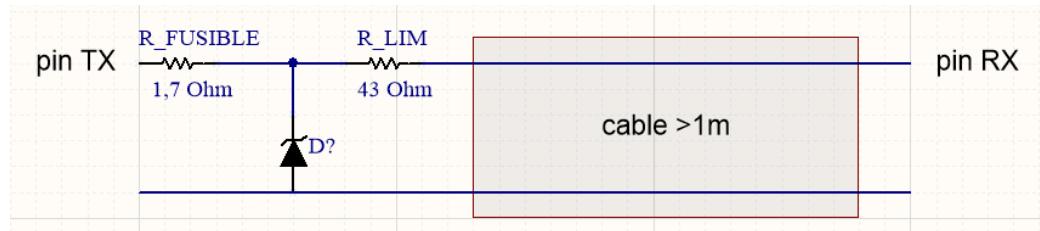


FIGURA 4.3: Circuito equivalente para medir la transmisión no ideal de la USART.

Se realizaron mediciones utilizando los kits de desarrollo, modificando el firmware del nodo maestro para que envié bytes aleatorios. Esto permitió evaluar el diagrama de ojo de la señal (figura 4.4). Además, en la figura 4.5 se capturó un detalle de los flancos de las señales, tanto en la fuente (pin TX de los microcontroladores) como en el lado de la recepción (pin RX del slave).

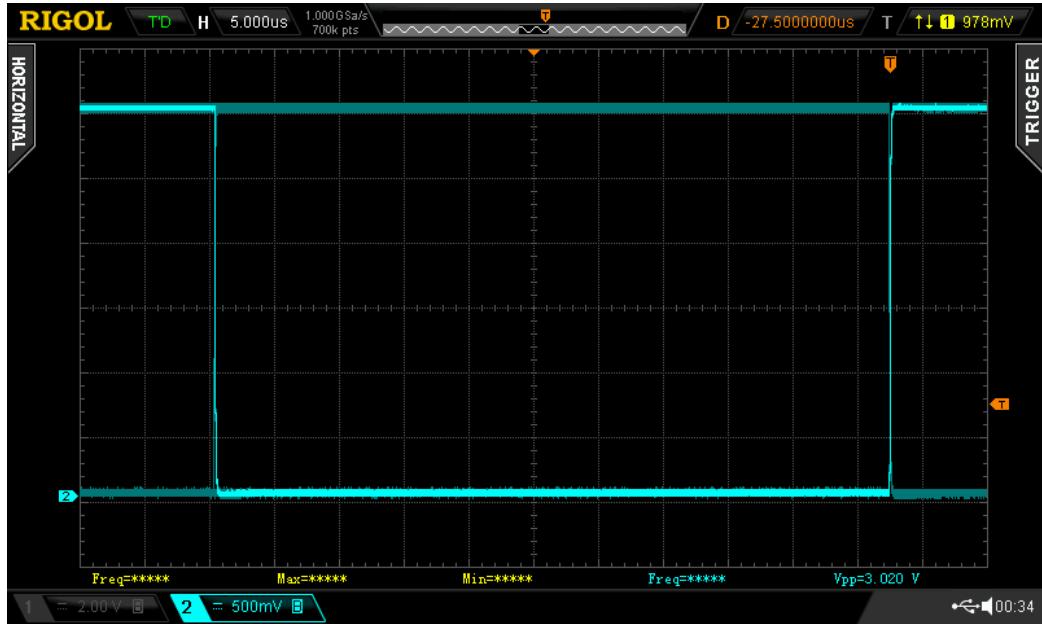


FIGURA 4.4: Diagrama de ojo para señales recibidas en un subnodo. Señal en amarillo: TX. Señal en celeste: RX.

La medición se realizó utilizando una tasa de transmisión de 19200 bps. Esto equivale a un tiempo de bit de  $52 \mu\text{s}$ . Se puede observar en la figura 4.5 que el *rise time* y el *fall time* no van a influir en la transmisión de los datos.

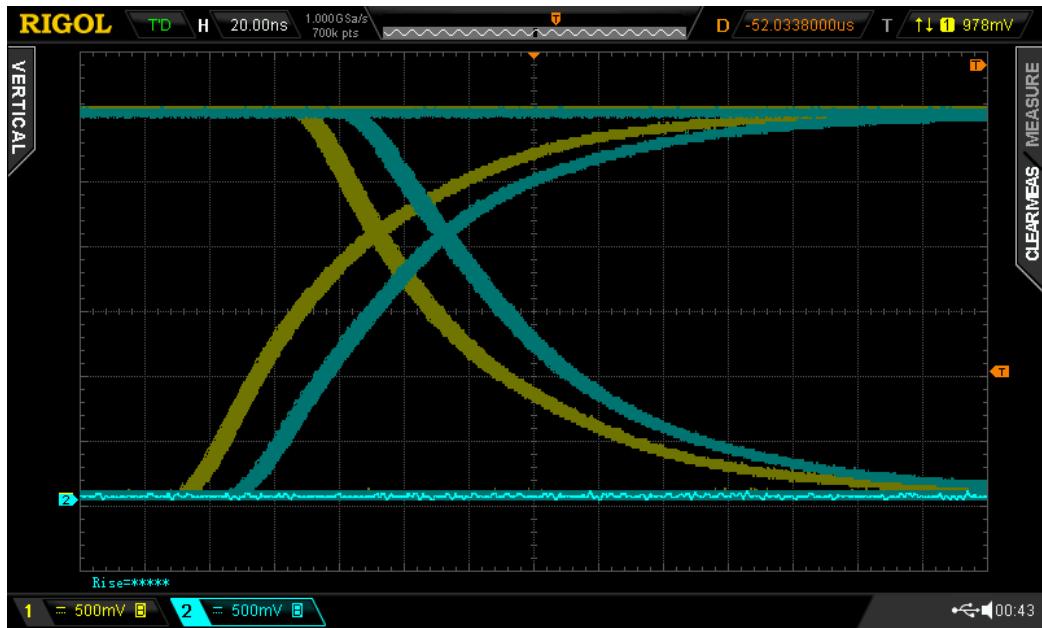


FIGURA 4.5: Tiempos de crecimiento y de caída de los flancos de la USART. Señal en amarillo: TX. Señal en celeste: RX.

Se concluye que la elección de transmitir utilizando niveles TTL en líneas de un metro de largo, mas el agregado de la barrera zener fue adecuada. Por otro lado, estas mediciones confirman los cálculos realizados en las sección 3.3.2 a través de la ecuación 3.5.

## 4.2. Master Test Plan

Se realizaron pruebas unitarias y funcionales de algunos de los módulos del firmware, tanto del cliente LoRaWAN y como el del los subnodos. Para esto, se utilizó el paradigma de programación orientada a pruebas.

Las pruebas unitarias se centraron en funcionalidades individuales de cada función de los módulos mientras que las pruebas funcionales, se diseñaron para evaluar la lógica del funcionamiento de los algoritmos.

Cabe aclarar que el módulo “packet framer”, si bien se desarrolló con anterioridad y se reutilizó en este trabajo, no contaba originalmente con pruebas, por lo tanto se le realizaron un conjunto de pruebas unitarias y funcionales también a dicho módulo.

Tanto para las pruebas unitarias como para las pruebas funcionales, se utilizó el *framework* ceedling[72]. Ceedling es un sistema de construcción para proyectos en C basado en el sistema de construcción rake [73]. El mismo, integra tres submódulos llamados CMock, Unity y CException.

### 4.2.1. Pruebas unitarias

#### Driver de mediciones

El driver de mediciones utilizado fue portado y modificado para que se adaptara al esquema del firmware de los subnodos. Por ello, se realizaron pruebas para evaluar el funcionamiento del mismo. Durante la generación de las pruebas, se encontraron algunos bugs que se remediaron (por ejemplo, simulando respuestas del sensor que eran invalidas).

En la tabla 4.1 se recopilaron los casos de prueba para este módulo de software.

Dado que es un grupo de pruebas que se ejecutan sobre un periférico, los test que se hicieron utilizaron *mocks*, para simular las respuestas del bus I<sup>2</sup>C y del sensor en si. De esa manera, las pruebas terminaron validando la parte lógica del driver.

TABLA 4.1: Pruebas unitarias realizadas sobre el driver del sensor STH31.

<b>Id</b>	<b>Tipos</b>	<b>Nombre</b>	<b>Descripción</b>
STH31_001	positivo	Inicialización	Verifica la correcta inicialización de las variables.
STH31_002	positivo	Valores de medición de temperatura	Verifica la correcta decodificación de los valores mínimos, medios y máximos del sensor de temperatura.
STH31_003	positivo	Valores de medición de humedad	Verifica la correcta decodificación de los valores mínimos, medios y máximos del sensor de humedad.
STH31_004	negativo	Error en comunicación I <sup>2</sup> C	Verifica que el driver detecte correctamente los errores de crc de las respuestas del sensor.

Todas las pruebas unitarias de esta biblioteca fueron superadas, tal como muestra la figura 4.6 y su prueba de cobertura puede observarse en la figura 4.7.

```
Test 'test_sth.c'
-----
Compiling Adafruit_SHT31.c...
Compiling mocks.c...
Compiling cmock.c...
Linking test_sth.out...
Running test_sth.out...

-----
TEST OUTPUT
-----
[test_sth.c]
- "running test_resetted"
- "running test_meas_hum_mid_point"
- "running test_meas_hum_max_point"
- "running test_meas_hum_min_point"
- "running test_meas_temp_mid_point"
- "running test_meas_temp_max_point"
- "running test_meas_temp_min_point"
- "running test_meas_corrupt"

-----
OVERALL TEST SUMMARY
-----
TESTED: 8
PASSED: 8
FAILED: 0
IGNORED: 0
```

FIGURA 4.6: Resultado de la pruebas unitarias del driver para el STH31.

File	Lines	Branches
<a href="#">Adafruit_SHT31.c</a>	100.0 %	56 / 56 100.0 %

FIGURA 4.7: Resultado de la prueba de cobertura del driver para el STH31.

### Protocolo de red de subnodos

En este caso se realizaron pruebas sobre las funciones de generación de respuestas y de interpretación de comandos.

En la tabla 4.2 se recopilaron los casos de prueba para este módulo de software.

TABLA 4.2: Pruebas unitarias realizadas sobre el protocolo de red de subnodos

<b>Id</b>	<b>Tipos</b>	<b>Nombre</b>	<b>Descripción</b>
RSN_001	positivo	Tipo de trama	Verifica el comportamiento de la función que devuelve el tipo de trama (comando/respuesta)
RSN_002	positivo y negativo	Integridad de la trama	Verifica el comportamiento de la función que valida el LRC de la trama.
RSN_003	negativo	Integridad de la trama por error de tamaño	Verifica el comportamiento de la función que valida la trama según el campo de tamaño
RSN_004	positivo	Generación de la trama	Verifica el comportamiento de la función que genera el comando "Sense All"
RSN_005	positivo	Cálculo de LRC	Verifica el comportamiento de la función que calcula el LRC
RSN_006	positivo	Inicialización del nodo maestro	Verifica el comportamiento de la función que inicializa una instancia de nodo maestro
RSN_007	positivo	Inicialización del subnodo	Verifica el comportamiento de la función que inicializa una instancia de un subnodo
RSN_008	positivo	Procesamiento de comando "Sense All"	Se genera un comando SenseAll con ciertas mediciones conocidas, y se prueba el comportamiento de la función que lo interpreta

Todas las pruebas unitarias de esta biblioteca fueron superadas, tal como muestra la figura 4.12. En la figura 4.13<sup>1</sup> se muestra el resultado de las mismas.

---

<sup>1</sup>Las pruebas unitarias y funcionales se corren todas juntas, por ello los resultados de las pruebas y los resultados de cobertura se encuentran más adelante en este documento.

### Protocolo de aplicación montado sobre LoRAWAN

Dado que el protocolo es un grupo de funciones que genera respuestas y interpretar comandos, las pruebas realizadas fueron unitarias. La tabla 4.3 muestra una lista de pruebas realizadas.

TABLA 4.3: Pruebas unitarias realizadas sobre el protocolo de aplicación sobre LoRaWAN

<b>Id</b>	<b>Tipos</b>	<b>Nombre</b>	<b>Descripción</b>
ALW_001	positivo y negativo	Trama válida	Verifica el comportamiento de la función que valida de trama.
ALW_002	positivo	Funciones de conversión	Verifica el comportamiento de funciones internas de la biblioteca.
ALW_003	positivo	Interpretación de comandos	Verifica el funcionamiento básico de la interpretación de comandos.
ALW_004	positivo	Herramientas para creación de sub payloads	Verifica el funcionamiento de funciones de soporte para darle formato comandos más elaborados.
ALW_005	positivo	Mensaje de mediciones	Genera un comando de mediciones usando las funciones para ello, y valida la creación haciendo uso de un comando de interpretación del mismo.
ALW_006	positivo	Comandos de configuración	Se valida la interpretación de comandos de configuración, para varios casos de uso.
ALW_007	positivo	Comandos relacionados con la actualización de firmware	Se valida la correcta interpretación de los comandos relacionados con la actualización de firmware.

Los resultados arrojados se pueden observar en la figura 4.8. Además de las pruebas realizadas, que se superaron con éxito, se realizaron pruebas de cobertura alcanzando el 100 % como muestra la figura 4.9.

```
Test 'test_app_lora_parse.c'
-----
Running test_app_lora_parse.out...

Test 'test_app_lora_parse_set_parameters.c'
-----
Running test_app_lora_parse_set_parameters.out...

-----
OVERALL TEST SUMMARY
-----
TESTED: 14
PASSED: 14
FAILED: 0
IGNORED: 0
```

FIGURA 4.8: Resultado de la pruebas unitarias de las funciones de interpretación de comandos de configuración.



FIGURA 4.9: Resultado de la prueba de cobertura del módulo de interpretación de comandos de la aplicación.

Con los comandos de actualización de firmware, se hizo lo mismo, pero además se agregaron pruebas funcionales. Las pruebas unitarias realizadas sobre este módulo se resumen en la tabla 4.4.

TABLA 4.4: Pruebas unitarias realizadas sobre los interpretes de comandos de actualización de firmware.

<b>Id</b>	<b>Tipos</b>	<b>Nombre</b>	<b>Descripción</b>
ALW_008	negativo	Trama ADD invalida 1	Verifica que la interpretación del comando de actualización de firmware “agregar” falle cuando el código de comando es inválido.
ALW_009	negativo	Trama ADD invalida 2	Verifica que la interpretación del comando de actualización de firmware “agregar” falle cuando la dirección de flash es invalida.
ALW_010	negativo	Trama ADD invalida 3	Verifica que la interpretación del comando de actualización de firmware “agregar” falle cuando la porción de imagen supera la flash disponible.
ALW_011	positivo	Trama ADD valida	Verifica que la interpretación del comando de actualización de firmware “agregar” sea válida.
ALW_012	negativo	Trama END invalida 1	Verifica que la interpretación de un comando de actualización de firmware “finalizar” falle cuando el código de comando es inválido.
ALW_013	negativo	Trama END invalida 2	Verifica que la interpretación de un comando de actualización de firmware “finalizar” falle cuando el código de comprobación.
ALW_014	negativo	Trama CANCEL invalida	Verifica que la interpretación de un comando de actualización de firmware “cancelar” falle cuando el código de comando es inválido.

Los resultados obtenidos de estas pruebas pueden observarse en las figuras 4.10 y 4.11.

```

Test 'test_app_lora_fu_functional_1.c'
-----
Running test_app_lora_fu_functional_1.out...

Test 'test_app_lora_fu_functional_2.c'
-----
Running test_app_lora_fu_functional_2.out...

Test 'test_app_lora_fu_parsers.c'
-----
Running test_app_lora_fu_parsers.out...

-----
OVERALL TEST SUMMARY
-----
TESTED: 12
PASSED: 12
FAILED: 0
IGNORED: 0

```

FIGURA 4.10: Resultado de la pruebas unitarias de las funciones de interpretación de comandos de actualización de firmware.

File		Lines	Branches
<code>protocols/firmware_update/src/app_lora_abstraction_fu.c</code>		100.0 %	58 / 58
<code>protocols/firmware_update/src/app_lora_parse_fu.c</code>		100.0 %	39 / 39

FIGURA 4.11: Resultado de la prueba de cobertura del módulo de interpretación de comandos de actualización de firmware.

#### 4.2.2. Pruebas funcionales

##### Protocolo de red de subnodos

Las pruebas funcionales relacionadas con éste protocolo se centraron en definir varios escenarios posibles con relación a distintas secuencias (en este documento solo se mostró el diagrama más completo que el que muestra la figura 3.24).

La tabla 4.5 muestra los casos de prueba ejecutados.

TABLA 4.5: Pruebas funcionales realizadas sobre el protocolo de red de subnodos

Id	Tipos	Nombre	Descripción
RSN_009	positivo	Transacción entre maestro y un subnodo	Simula y valida la secuencia de intercambio de comandos entre un maestro y un nodo, similar al mostrado en la figura 3.24.
RSN_010	positivo	Transacción entre maestro y un subnodo 2	Simula y valida el envío de un paquete dirigido a un nodo “lejano”.
RSN_011	positivo	Transacción entre maestro y un subnodo 2	Simula y valida el envío de un paquete que llega al nodo con datos inválidos.
RSN_012	positivo	Transacción entre maestro y dos nodos subnodos	Simula y valida la secuencia de intercambio de comandos entre un maestro y dos nodos tal cual el de la figura 3.24.

El resultado de las pruebas se puede observar en la figura 4.12 y los resultados de cobertura en la figura 4.13.

```

Test 'test_net.c'
-----
Running test_net.out...

Test 'test_net_parse.c'
-----
Running test_net_parse.out...

Test 'test_process.c'
-----
Running test_process.out...

Test 'test_sueltos.c'
-----
Running test_sueltos.out...

Test 'test_transaction.c'
-----
Running test_transaction.out...

Test 'test_transaction_2.c'
-----
Running test_transaction_2.out...

-----
OVERALL TEST SUMMARY
-----
TESTED: 19
PASSED: 19
FAILED: 0
IGNORED: 0

```

FIGURA 4.12: Resultado de la prueba de unitarias y funcionales del módulo de red de subnodos

File	Lines		Branches	
<a href="#">protocols/net/src/net_parse.c</a>		100.0 %	70 / 70	100.0 %
<a href="#">redundancy_check/src/lrc.c</a>		100.0 %	6 / 6	100.0 %
<a href="#">protocols/net/src/net_process.c</a>		100.0 %	114 / 114	91.1 %
<a href="#">protocols/net/src/net_uart.c</a>		100.0 %	47 / 47	80.0 %

FIGURA 4.13: Resultado de la prueba de cobertura del módulo de red de subnodos

En la figura 4.13, se puede observar que el módulo net\_uart.c posee algunas ramas no probadas. Esta ramas están vinculadas con condiciones relacionadas con la manera de compilar a la biblioteca. Para realizar las pruebas, la biblioteca se debe compilar con los dos comportamientos posibles, maestro y esclavo. Para que funcione correctamente, y permitir la reutilización de código fuente, hubieron que utilizar condicionales en tiempo de ejecución. Es por eso, que algunas ramas no fueron cubiertas.

### Protocolo de aplicación montado sobre LoRAWAN

Las pruebas funcionales realizadas para esta biblioteca, fueron hechas para la validación de los comandos relacionados con el protocolo de actualización de firmware.

El alcance del presente trabajo no incluía realizar software para el servidor que reciba/envíe los comandos de LoRaWAN. Por lo tanto el protocolo de actualización de firmware se implementó según 3.4.3 y se realizaron pruebas sobre los mismos para garantizar su lógica. No se incluyeron aun en los proyectos del firmware del cliente LoRaWAN.

Las pruebas realizadas sobre los algoritmos de actualización de firmware, consistieron en simular el arribo de comandos del servidor y la validación de los contenidos de la memoria resultante de haber aplicado dichos comandos.

Las pruebas funcionales se dividieron en dos partes. La primera consistió en probar los comandos individualmente y validar las porciones de memoria flash actualizadas que corresponde a los comandos enviados. La segunda parte de las pruebas consistió en simular una actualización completa en donde de manera aleatoria se genera, para cada prueba, el contenido de la imagen, y posterior a eso, los comandos asociados se envían en un orden no secuencial (excluyendo al comando “finalizar” que siempre se envió al final).

La figura 4.10 y la figura 4.11 además de mostrar los resultados de las pruebas unitarias muestran el resultado de las pruebas funcionales aquí descritos.

## 4.3. Pruebas de integración

En esta sección se muestran los resultados obtenidos relacionados con la prueba del sistema.

Las pruebas fueron realizadas empleando kits de desarrollo, tanto para el nodo maestro como para los subnodos. Se utilizaron el “MultiTech mDot Developer Kit”<sup>[74]</sup> para emular el nodo maestro y dos “NUCLEO-L010RB”<sup>[75]</sup> para los subnodos. Para el sensor de temperatura y humedad, en cada uno de los “NUCLEO-L010RB”, se utilizó el módulo de AdaFruit<sup>[76]</sup>.

Por otro lado, como “gateway” se utilizó el Multitech Conduit<sup>[27]</sup> para que cumpla la función de gateway. Este dispositivo se puede configurar como “Network server” (y gestionar al protocolo LoRaWAN por si solo) o como “packet forwarder” (en cuyo caso, solo hace de puente entre el sistema LoRa y un “Network server” remoto).

La elección para este banco de prueba fue utilizar al dispositivo de Multitech, como “packet forwarder”. Se utilizó entonces, un “network server” remoto llamado “The Things Network” (TTN)<sup>[77]</sup>. Se eligió esta topología para poder evaluar con mayor confianza a la transmisión de datos. Además, el sistema “TTN” permitió acceder a los datos recibidos a través de un *dashboard*.

La figura 4.14 esquematiza el banco de prueba del sistema. La figura 4.15 muestra imágenes de los dispositivos utilizados para realizar estas pruebas y con las cuales se desarrolló todo el firmware.

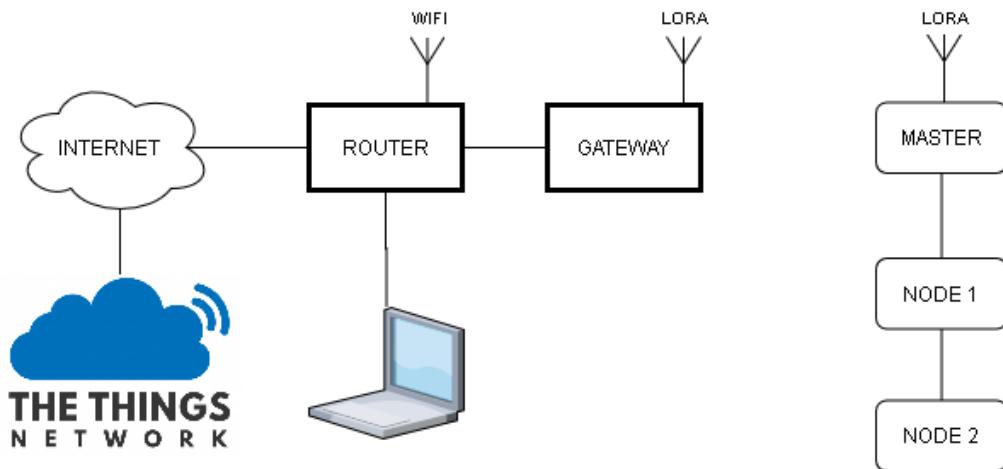


FIGURA 4.14: Banco de prueba para realizar el trabajo.

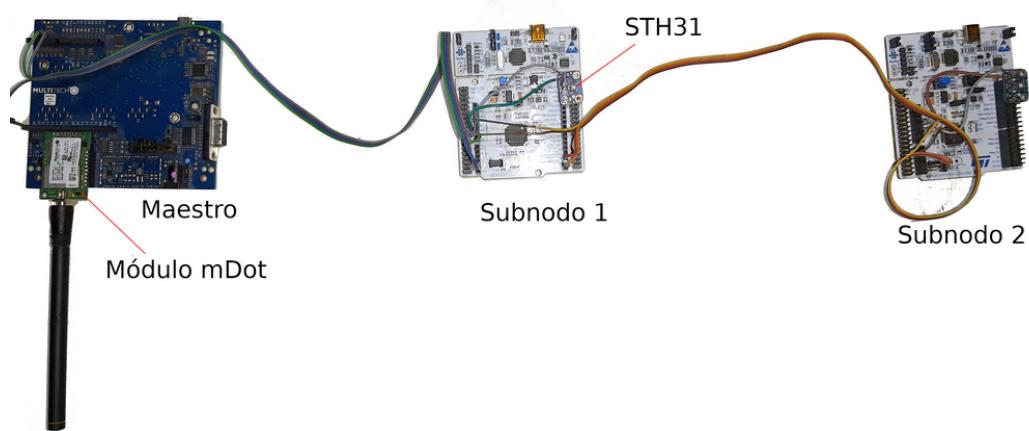


FIGURA 4.15: Kits de desarrollo emulando un arreglo de dos subnodos.

Las pruebas que se realizaron consistieron en:

- Configurar todos los parámetros de networking.
- Configurar todos los parámetros de la red LoRAWAN.
- Descargar las imágenes de los firmware a cada kit de desarrollo.
- Encender el kit de desarrollo.
- Validar la llegada periódica de datos al servidor de TTN.

Como puede observarse en la figura 4.16 el *dashboard* del network server muestran los paquetes entrantes proveniente desde el cliente LoRaWAN y habiendo atravesado al “gateway” Multitech Conduit.

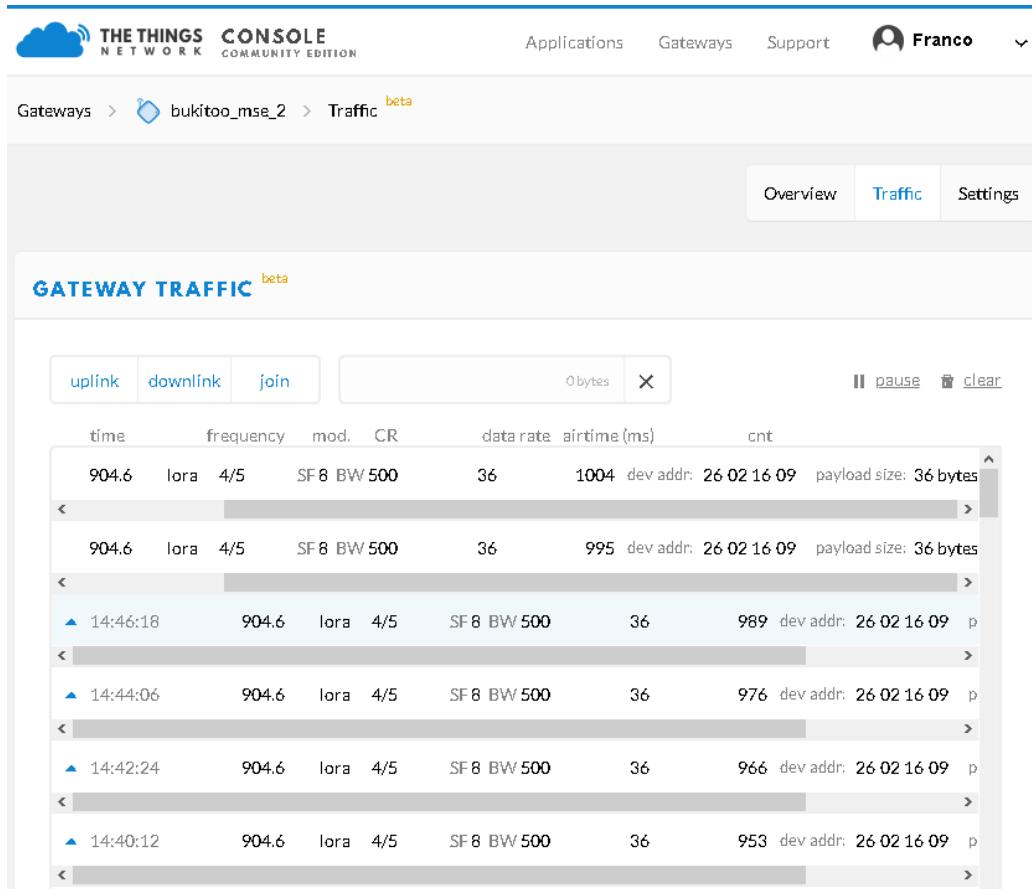


FIGURA 4.16: Trafico de paquetes en el servidor de red.

Luego, el sistema TTN permitió también implementar un servidor de aplicación en el cual se analizan los datos entrantes de los paquetes. En la figura 4.17 se muestra el *dashboard* del servidor de aplicación, en donde están almacenados los valores de las mediciones de temperatura y humedad de cada nodo del sistema. En la misma, se puede observar un paquete recibido en donde se muestra el detalle de como es interpretado mediante el protocolo de aplicación que se explicó en 3.4.3. Además, en otros paquetes, se puede observar el resultado del interprete programado en TTN, extrayendo de los mismos los valores de temperatura y humedad para cada subnodo (en el caso del banco de prueba, solo dos).

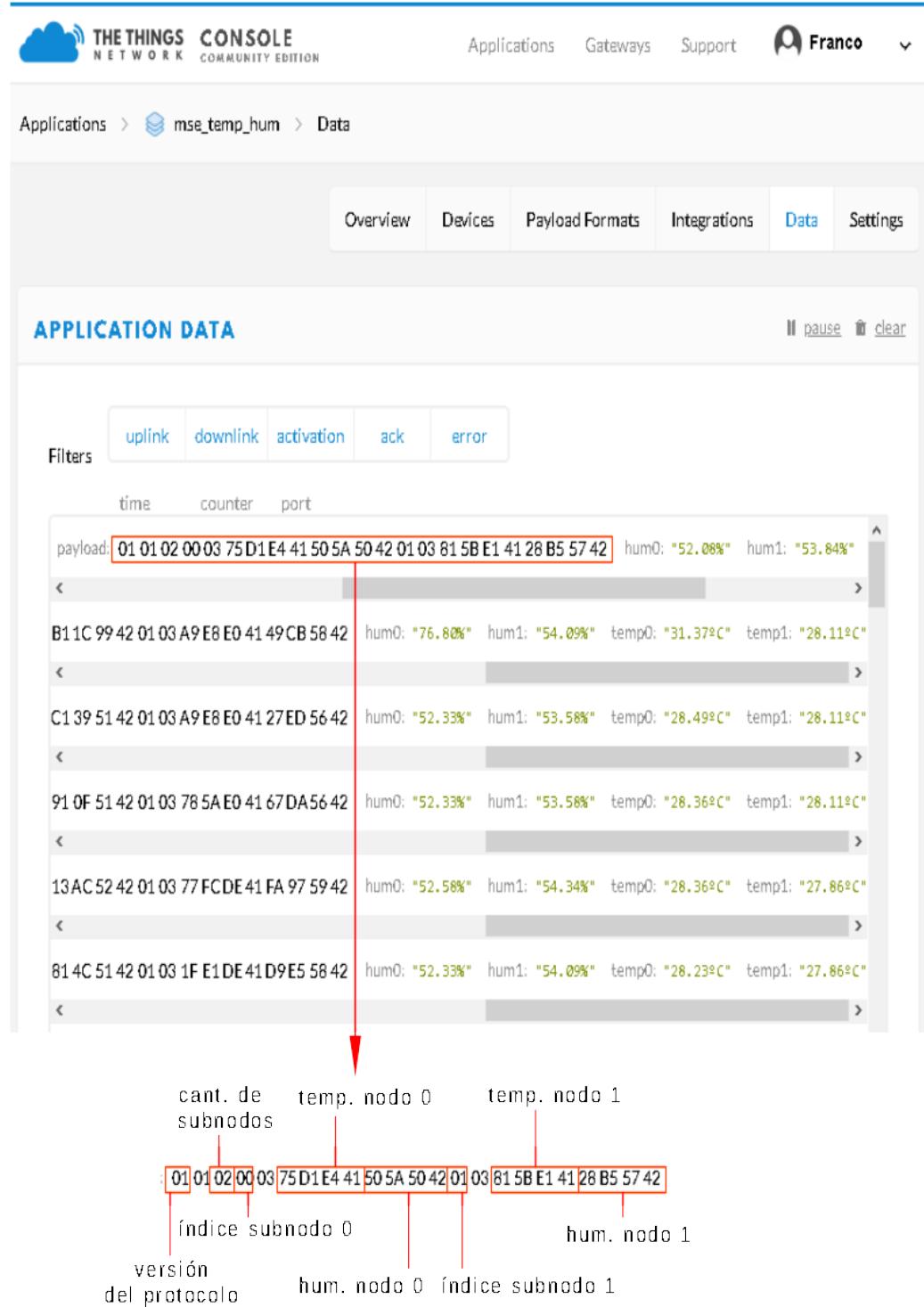


FIGURA 4.17: Trafico de paquetes en el servidor de aplicación.

De esta forma se validó el funcionamiento correcto de todas las etapas del sistema.

## Capítulo 5

# Conclusiones

En este capítulo se destacan los principales aportes del trabajo realizado y se muestran los logros obtenidos. Además se plantean los trabajos futuros a realizar.

### 5.1. Objetivos alcanzados

En el trabajo realizado, se logró implementar de forma exitosa el sistema de una subred de sensores cableada para utilizar en el transporte de granos fluvial, con conectividad LoRaWAN. Se pudo validar y demostrar su funcionamiento, con la ayuda del servidor de aplicación The Things Network. Para ello, se diseñaron y fabricaron con éxito los prototipos de los circuitos impresos del nodo maestro y de los subnodos.

Se pudo desarrollar el firmware de ambos subsistemas, de manera modular, empleando técnicas de desarrollo basado en pruebas, en algunos de los módulos. Se utilizó el sistema de control de versiones GIT utilizando un servidor privado.

En el desarrollo se utilizaron conocimientos adquiridos durante las materias de la Maestría en Sistemas Embebidos tales como:

- Gestión de la tecnología y la innovación: asignatura en la cual se dan herramientas para que produzca innovación tecnológica. En particular, durante esta asignatura se redacta el plan del trabajo final.
- Certificación de sistemas electrónicos: sirvió de puntapié para evaluar las normativas que abarcaban al dispositivo desarrollado, y se desarrolló un análisis de la norma IEC60079 aplicada al mismo.
- Testing de Software en Sistemas Embebidos: materia fundamental para la correcta confección y documentación de las pruebas automáticas para los módulos del firmware del dispositivo.
- Implementación de Sistemas Operativos: se utilizaron herramientas de la misma para adaptar los módulos portables del firmware tanto para una arquitectura baremetal (firmware de los subnodos) como para una arquitectura basada en un RTOS (firmware del cliente LoRaWAN).
- Taller de trabajo final: asignatura en donde se dieron técnicas de redacción de documentación académica.

Se completa de esta manera una primera iteración del dispositivo que permitirá a la empresa WiAgro contar con un mínimo producto viable para evaluar el impacto del mismo. Sin embargo, WiAgro tuvo inconvenientes para fabricar los prototipos

a tiempo, lo que ocasionó que algunos objetivos no pudieran alcanzarse. Entre ellos la migración del firmware a la plataforma final.

## 5.2. Próximos pasos

### 5.2.1. Asesoría en relación a normativa de atmósferas explosivas

Muchas de las prácticas y criterios adoptados para el diseño del hardware, en relación a las normativa para atmósferas explosivas, estuvieron basadas en interpretaciones del autor. Diseñar un dispositivo a certificar para atmósferas explosivas requiere de un asesoramiento profesional de personas que sean estudiantes de la normativa y que se capaciten constantemente al respecto. Por ello, a futuro será indispensable contratar una consultoría para evaluar el trabajo realizado y realizar los cambios pertinentes, aun cuando haya que diseñar un gabinete propio y rediseñar los circuitos impresos.

### 5.2.2. Diseño mecánico de la columna de sensores

En este trabajo se ignoró por completo el diseño de los elementos mecánicos para la sujeción de la columna de sensores en el sitio de instalación. Originalmente se pensó que el soporte mecánico podría ser algún tipo de gancho en el dispositivo maestro y posterior a eso que la red de sensores cuelgue simplemente de los cables. Para validar cualquiera de las ideas habrá que realizar pruebas, pero con la columna de sensores inmersa en varias toneladas de granos, para ver si soporta el estrés.

### 5.2.3. Miniaturización y disminución del costo de los subnodos

Es importante recalcar nuevamente que el diseño de los subnodos se realizó con un microcontrolador sobredimensionado. Esto se realizó por la incertidumbre en cuanto a ocupación de memoria flash por parte de algunos algoritmos y bibliotecas utilizados.

El firmware de los subnodos, en el estado actual en que se encuentra, posee una huella de aproximadamente 17 kB de FLASH y 2 kB de RAM. Esto hace que el microcontrolador elegido pueda ser uno de menos memoria. La elección de un micro más pequeño permitirá reducir el tamaño de los subnodos considerablemente, y así bajar el costo. Deberá también elegirse un gabinete apropiado para el nuevo tamaño, considerando las pautas impuestas por la certificación de atmósferas explosivas.

### 5.2.4. Realizar evaluaciones con más nodos

En el presente trabajo se pudo probar solamente la red de subnodos con dos nodos. La prueba definitiva se realizará cuando:

- Se redefinan algunas cuestiones relacionadas con la certificación para atmósferas explosivas, lo que podría ocasionar un cambio en el hardware.
- Se definan temas constructivos relacionados con el cableado (que dependen del punto anterior).
- El cliente fabrique una cantidad sustancial de nodos, para realizar las pruebas.

En ese nuevo contexto se podrá probar la latencia total de la red para el comando *Sense All*. Además se permitirá medir toda la parte relacionada con la gestión de energía del sistema.



# Bibliografía

- [1] Bernadette Abadía y Ricardo Bartosik. *Manual de Buenas Prácticas en Poscosecha de Granos*. Ene. de 2013, pág. 1-194. ISBN: 9789876792646.
- [2] *La agricultura mundial en la perspectiva del año 2050*. Foro de Expertos de Alto Nivel. 2009. URL: [http://www.fao.org/fileadmin/templates/wsfs/docs/Issues\\_papers/Issues\\_papers\\_SP/La\\_agricultura\\_mundial.pdf](http://www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/Issues_papers_SP/La_agricultura_mundial.pdf).
- [3] *Consideraciones previas al almacenamiento que se deben tener en cuenta*. Disponible: 2020-3-29. URL: <http://www.cosechaypostcosecha.org/data/articulos/postcosecha/AlmacenamientoSoja.asp>.
- [4] *SISTEMA DE MEDICIÓN Y CONTROL DE PLANTA DE SILOS PARA ALMACENAMIENTO DE CEREALES*. URL: [http://www.frp.utn.edu.ar/info2/?page\\_id=8588](http://www.frp.utn.edu.ar/info2/?page_id=8588).
- [5] *TEMPERATURE MONITORING*. URL: [www.tornum.com/product/temperature-monitoring](http://www.tornum.com/product/temperature-monitoring).
- [6] *SENSORES DE TEMPERATURA PARA SILOS - TERMOMETRÍA*. URL: [www.measureinstruments.com.ar/es/temperatura-silos.html](http://www.measureinstruments.com.ar/es/temperatura-silos.html).
- [7] *SONDA TM*. URL: [www.smartfactorysac.com.pe/categorias/instrumentacion-industrial/196/80/termometr%C3%ADa-de-silos/sonda-tm-detail.html](http://www.smartfactorysac.com.pe/categorias/instrumentacion-industrial/196/80/termometr%C3%ADa-de-silos/sonda-tm-detail.html).
- [8] *EST110 Multi-point temperature sensor*. URL: [www.fine-tek.com/main/pinfo\\_in.aspx?mnuid=1284&modid=4&pcid=14&pscld=18&pid=130](http://www.fine-tek.com/main/pinfo_in.aspx?mnuid=1284&modid=4&pcid=14&pscld=18&pid=130).
- [9] *GWSL 1100 – Temperature Cable Standard*. URL: <http://liros.se/products/gw-temperature-cable-gwsl1100/?lang=en>.
- [10] *LoRa — Wikipedia, The Free Encyclopedia*. [Consultado: 2020-03-20]. 2020. URL: [en.wikipedia.org/w/index.php?title=LoRa&oldid=947931748](http://en.wikipedia.org/w/index.php?title=LoRa&oldid=947931748).
- [11] Franco Bucafusco. *Plan de Proyecto del Trabajo Final de la Maestría de Sistemas Embebidos*. 2019. URL: [docs.google.com/document/d/1eRFhj7oJZ1OKOlMszFaNRGN\\_wjV8cpp81M5wUuSs-g/edit?usp=sharing](https://docs.google.com/document/d/1eRFhj7oJZ1OKOlMszFaNRGN_wjV8cpp81M5wUuSs-g/edit?usp=sharing).
- [12] Tom Alby. *Activity-on-Node*. Disponible: 2019-11-16. URL: [project-management-knowledge.com/definitions/a/activity-on-node](http://project-management-knowledge.com/definitions/a/activity-on-node).
- [13] *Digital Humidity Sensor SHT3x (RH/T)*. Disponible: 2020-03-29. URL: [www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications/](http://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications/).
- [14] *Datasheet Membrane Option for SHT3x Humidity and Temperature Sensors*. Disponible: 2019-11-9. URL: [www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/0\\_Datasheets/Humidity/Sensirion\\_Humidity\\_Sensors\\_SHT3x\\_Datasheet\\_Filter\\_Membrane.pdf](http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/0_Datasheets/Humidity/Sensirion_Humidity_Sensors_SHT3x_Datasheet_Filter_Membrane.pdf).
- [15] *Digital Humidity Sensor SHT3x (RH/T)*. Disponible: 2019-11-16. URL: [www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications](http://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications).
- [16] *Chirp spread spectrum — Wikipedia, The Free Encyclopedia*. [Consultado: 2020-03-20]. 2020. URL: [en.wikipedia.org/w/index.php?title=Chirp\\_spread\\_spectrum&oldid=934101195](http://en.wikipedia.org/w/index.php?title=Chirp_spread_spectrum&oldid=934101195).

- [17] *LoRa Modulation Basics | CSS Modulation | Advantages, Properties.* Disponible: 2020-03-29. URL: [www.rfwireless-world.com/Terminology/LoRa-modulation-vs-CSS-modulation.html](http://www.rfwireless-world.com/Terminology/LoRa-modulation-vs-CSS-modulation.html).
- [18] *LoRa Alliance.* Disponible: 2020-3-29. URL: [lora-alliance.org/](http://lora-alliance.org/).
- [19] Dong-Hoon Kim, Eun-Kyu Lee y Jibum Kim. «Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed». En: *Sustainability* 11 (mar. de 2019), pág. 1917. DOI: [10.3390/su11071917](https://doi.org/10.3390/su11071917).
- [20] *DASH7 — Wikipedia, The Free Encyclopedia.* [Consultado: 2020-03-20]. 2020. URL: [en.wikipedia.org/w/index.php?title=DASH7&oldid=938956438](http://en.wikipedia.org/w/index.php?title=DASH7&oldid=938956438).
- [21] *LoRaWAN vs Haystack.* Disponible: 2019-11-05. URL: [www.slideshare.net/haystacktech/lorawan-vs-haystack](http://www.slideshare.net/haystacktech/lorawan-vs-haystack).
- [22] *6LoWPAN — Wikipedia, The Free Encyclopedia.* [Consultado: 2020-03-20]. 2020. URL: [en.wikipedia.org/w/index.php?title=6LoWPAN&oldid=946752525](http://en.wikipedia.org/w/index.php?title=6LoWPAN&oldid=946752525).
- [23] Xavier Vilajosana. *Transmission of IPv6 Packets over LoRaWAN.* Disponible: 2019-11-05. URL: [tools.ietf.org/html/draft-vilajosana-6lpwa-lora-hc-01](http://tools.ietf.org/html/draft-vilajosana-6lpwa-lora-hc-01).
- [24] «mBed - LoRaWAN overview». En: (). URL: [os.mbed.com/docs/mbed-os/v5.12/apis/lorawan.html](http://os.mbed.com/docs/mbed-os/v5.12/apis/lorawan.html) (visitado 29-03-2020).
- [25] *Reference implementation and documentation of a LoRa network node.* URL: [github.com/Lora-net/LoRaMac-node](http://github.com/Lora-net/LoRaMac-node) (visitado 29-03-2020).
- [26] *The Things Stack, an Open Source LoRaWAN Network Server.* URL: [github.com/TheThingsNetwork/lorawan-stack](http://github.com/TheThingsNetwork/lorawan-stack) (visitado 29-03-2020).
- [27] *MultiTech Conduit Programmable Gateway for the Internet of Things.* Disponible: 2020-3-20. URL: [www.multitech.com/brands/multiconnect-conduit](http://www.multitech.com/brands/multiconnect-conduit).
- [28] «LoRaWAN security». En: (). Disponible: 2019-11-12. URL: [www.slideshare.net/Actility/lorawan-security-webinar](http://www.slideshare.net/Actility/lorawan-security-webinar).
- [29] *Cryptographic Standards and Guidelines.* Disponible: 2020-3-29. URL: <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development>.
- [30] *Thales eSE secure end-to-end solutions.* Disponible: 2020-3-29. URL: [www.gemalto.com/mobile/secure-elements/embedded-secure-element](http://www.gemalto.com/mobile/secure-elements/embedded-secure-element).
- [31] «LoRa- (Long Range) Network and Protocol Architecture with Its Frame Structure». En: (). Disponible: 2019-11-12. URL: <http://www.techplayon.com/lora-long-range-network-architecture-protocol-architecture-and-frame-formats/>.
- [32] *UNIFIED INTERPRETATIONS OF SOLAS CHAPTERS II-1 AND XII, OF THE TECHNICAL PROVISIONS FOR MEANS OF ACCESS FOR INSPECTIONS (RESOLUTION MSC.158(78)) AND OF THE PERFORMANCE STANDARDS FOR WATER LEVEL DETECTORS ON BULK CARRIERS AND SINGLE HOLD CARGO SHIPS OTHER THAN BULK CARRIERS (RESOLUTION MSC.188(79)).* Disponible: 2019-11-9. URL: [www.mardep.gov.hk/en/msnote/pdf/msin1329anx1.pdf](http://www.mardep.gov.hk/en/msnote/pdf/msin1329anx1.pdf).
- [33] «Estándares y Normativas de zonas peligrosas». En: (2014).
- [34] *IEC 60079-0/2007: Atmósferas explosivas. Parte 0. Equipo. Requisitos generales.* 2007.
- [35] *GOST 31610.0/2014: Atmósferas explosivas. Parte 0. Equipo. Requisitos generales.* 2014.
- [36] *GOST 31610.10-2/2017: Atmósferas explosivas. Parte 10-2. Zonas de clasificación. Ambientes de polvo explosivo.* 2017.

- [37] GOST 31610.31/2010 60079-0/2007: *Atmósferas explosivas. Parte 31. Equipo con protección contra explosiones por ignición de polvo “t”*. 2010.
- [38] IEC, ed. IEC 60079-11/1999: *Atmósferas explosivas. Parte 11. Seguridad Intrínseca “i”*. 1999.
- [39] GOST 31610.11-2014: *Atmósferas explosivas. Parte 11. Equipos con tipo de protección “circuito eléctrico intrínsecamente seguro” “i”*. 2014.
- [40] GOST 60664.1/2012: *Coordination of insulation for equipment in low-voltage systems. Part 1. Principles, requirements and tests*. 2012.
- [41] IEC 60529/2001: *Degrees of protection provided by enclosures (IP Code)*. 2001.
- [42] Layer ignition temperature (LIT): 5mm layer ignition temperature test. Disponible: 2019-11-9. URL: [www.explosiontesting.co.uk/layer\\_ignition\\_8.html](http://www.explosiontesting.co.uk/layer_ignition_8.html).
- [43] Dust Explosions - Critical Temperatures and Concentrations. Disponible: 2019-11-9. URL: [www.engineeringtoolbox.com/dust-explosions-temperature-d\\_394.html](http://www.engineeringtoolbox.com/dust-explosions-temperature-d_394.html).
- [44] LS, LSH, LSP. Disponible: 2020-3-29. URL: [www.saftbatteries.com/products-solutions/products/ls-lsh](http://www.saftbatteries.com/products-solutions/products/ls-lsh).
- [45] Nicolás Paquin. «Considerations for intrinsically safe batteries». En: (2009). Disponible: 2019-11-5. URL: [www.electronicproducts.com/Power\\_Products/Batteries\\_and\\_Fuel\\_Cells/Considerations\\_for\\_intrinsically\\_safe\\_batteries.aspx](http://www.electronicproducts.com/Power_Products/Batteries_and_Fuel_Cells/Considerations_for_intrinsically_safe_batteries.aspx).
- [46] Ex pert Electrical Technologies. *Intrinsic Safety (IS)*. Disponible: 2020-3-29. URL: [expeltec.com/2015/03/23/intrinsic-safety-is](http://expeltec.com/2015/03/23/intrinsic-safety-is).
- [47] Optimum Design Associates Blog. *Clearance and Creepage Rules for PCB Assembly*. Disponible: 2020-3-29. URL: <http://blog.optimumdesign.com/clearance-and-creepage-rules-for-pcb-assembly>.
- [48] *IPC-HDBK-830A: Guidelines for Design, Selection, and Application of Conformal Coatings*. URL: <https://shop.ipc.org/IPC-HDBK-830A-English-D>.
- [49] *Conformal coating — Wikipedia, The Free Encyclopedia*. [Consultado: 2020-03-20]. 2020. URL: [https://en.wikipedia.org/w/index.php?title=Conformal\\_coating&oldid=943928801](https://en.wikipedia.org/w/index.php?title=Conformal_coating&oldid=943928801).
- [50] *Multiple I2C sensors with the same address*. Disponible: 2019-11-06. URL: [arduino.stackexchange.com/questions/33004/multiple-i2c-sensors-with-the-same-address/33018](https://arduino.stackexchange.com/questions/33004/multiple-i2c-sensors-with-the-same-address/33018).
- [51] *Daisy chain (electrical engineering) — Wikipedia, The Free Encyclopedia*. [Consultado: 2020-03-20]. 2020. URL: [https://en.wikipedia.org/w/index.php?title=Daisy\\_chain\\_\(electrical\\_engineering\)&oldid=948728345](https://en.wikipedia.org/w/index.php?title=Daisy_chain_(electrical_engineering)&oldid=948728345).
- [52] Hammond Manufacturing. *ABS Plastic Hand Held Enclosures - RH Series*. Disponible: 2020-3-29. URL: [www.hammfg.com/electronics/small-case/plastic/rh-bd](http://www.hammfg.com/electronics/small-case/plastic/rh-bd).
- [53] Chillemi Hnos. *Gabinetes plásticos*. Disponible: 2020-3-29. URL: <http://www.chillemihnos.com.ar/espanol/productos.htm#P07N>.
- [54] *MultiTech xDot - Long Range LoRa® Modules (MTXDOT Series)*. URL: [www.multitech.com/brands/multiconnect-xdot](http://www.multitech.com/brands/multiconnect-xdot).
- [55] STM32L151/152. URL: [www.st.com/en/microcontrollers-microprocessors/stm32l151-152.html](http://www.st.com/en/microcontrollers-microprocessors/stm32l151-152.html).
- [56] *Semtech SX1272 - Long Range, Low Power RF Transceiver 860-1000 MHz with LoRa® Technology*. URL: [www.semtech.com/products/wireless-rf/lora-transceivers/sx1272](http://www.semtech.com/products/wireless-rf/lora-transceivers/sx1272).

- [57] *MultiConnect xDot Developer Guide*. Ver. 3. MultiTech.
- [58] *ENERGIZER L91*. URL: [data.energizer.com/PDFs/l91.pdf](http://data.energizer.com/PDFs/l91.pdf).
- [59] *MCP1812 - 300 mA Ultra-Low I<sub>q</sub> LDO*. URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/MCP1811A-11B-12A-12B-Data-Sheet-20006088B.pdf>.
- [60] *6-*u*A Quiescent Current, 1.8-A Switch Current Boost Converter*. URL: [www.ti.com/product/TPS61322](http://www.ti.com/product/TPS61322).
- [61] Dr Jim Allen. «Intrinsic Safety Design Considerations». En: (). Disponible: 2020-01-10. URL: <http://www.datalink-electronics.co.uk/assets/IS-Design-Considerations-Issued.pdf>.
- [62] *RP-SMA Female To U.FL IPX 1.13 Pigtail Cable 20cm*. Disponible: 2020-3-29. URL: [alexnl.com/product/rp-sma-female-to-u-fl-ipx-1-13-pigtail-cable-20cm](http://alexnl.com/product/rp-sma-female-to-u-fl-ipx-1-13-pigtail-cable-20cm).
- [63] *ISO1500 3-kVRMS Basic Isolated RS-485/RS-422 Transceiver in Ultra-Small Package datasheet*. Disponible: 2019-11-19. URL: <http://www.ti.com/lit/gpn/iso1500>.
- [64] Rubén Sánchez. *Convertidor push-pull*. 2016. URL: <http://rubensm.com/convertidor-push-pull/>.
- [65] *SN6505x Low-Noise 1-A Transformer Drivers for Isolated Power Supplies*. Disponible: 2019-11-19. URL: [www.ti.com/product/SN6505A](http://www.ti.com/product/SN6505A).
- [66] *STM32L0x0 Value Line*. URL: [www.st.com/en/microcontrollers-microprocessors/stm32l0x0-value-line.html](http://www.st.com/en/microcontrollers-microprocessors/stm32l0x0-value-line.html).
- [67] *STM32CubeIDE*. URL: [www.st.com/en/development-tools/stm32cubeide.html](http://www.st.com/en/development-tools/stm32cubeide.html).
- [68] *Packet Framer*. URL: [gitlab.com/fbucafusco/packet\\_framer](https://gitlab.com/fbucafusco/packet_framer).
- [69] *STM32Cube*. URL: <https://www.st.com/en/ecosystems/stm32cube.html>.
- [70] *Arduino Library for SHT31 Digital sensors*. URL: [github.com/adafruit/Adafruit\\_SHT31](https://github.com/adafruit/Adafruit_SHT31).
- [71] *Adafruit Industries*. URL: [www.adafruit.com](http://www.adafruit.com).
- [72] *WHAT IS Ceedling?* Disponible: 2020-1-20. URL: [www.throwtheswitch.org/ceedling](http://www.throwtheswitch.org/ceedling).
- [73] *Rake (software)*. Disponible: 2020-1-20. URL: [en.wikipedia.org/wiki/Rake\\_\(software\)](https://en.wikipedia.org/wiki/Rake_(software)).
- [74] *MultiTech mDotTM Developer Kit*. Disponible: 2020-3-20. URL: [www.multitech.com/brands/mdot-devkit](http://www.multitech.com/brands/mdot-devkit).
- [75] *STM32 Nucleo-64 development board with STM32L010RB MCU*. Disponible: 2020-3-20. URL: [www.st.com/en/evaluation-tools/nucleo-l010rb.html](http://www.st.com/en/evaluation-tools/nucleo-l010rb.html).
- [76] *Adafruit Sensirion SHT31-D - Temperature Humidity Sensor*. Disponible: 2020-3-20. URL: [www.adafruit.com/product/2857](http://www.adafruit.com/product/2857).
- [77] *The Things Network*. URL: [www.thethingsnetwork.org](http://www.thethingsnetwork.org).