



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

Departamento de Electrónica

## Desarrollo de arquitectura de data logger de bajo consumo con comunicación LPWAN

Memoria de Título presentada por

**Daisy Alexandra Berríos Araya**

como requisito parcial para optar al título de la carrera de

**Ingeniería Civil Electrónica**

Profesor Guía  
Alejandro Weinstein.  
Profesor Correferente  
Agustín González Valenzuela

MARZO, 2026



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción):  Memoria o trabajo de título  Tesis de Postgrado

Título del trabajo: Desarrollo de arquitectura de data logger de bajo consumo con comunicación LPWAN

Nombre del candidato(a): Daisy Alexandra Berríos Araya

Carrera / Grado: Ingeniería civil electrónica

Campus: Casa Central Departamento: Electronica

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Alejandro Weinstein, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses  12 meses  2 años  3 años  5 años  10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

---

---

---

### 4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 20-4-2026

Firma: A. Weinstein D.

Estudiante o Candidato(a):

Fecha: 17-4-2026

Firma:



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

TÍTULO DE LA TESIS:

**Desarrollo de arquitectura de data logger de bajo consumo con comunicación LPWAN**

AUTOR:

**Daisy Alexandra Berríos Araya**

TRABAJO DE MEMORIA, presentado como requisito parcial para optar al título de la carrera de INGENIERÍA CIVIL ELECTRÓNICA de la Universidad Técnica Federico Santa María.

Ph.D. Alejandro Weinstein

Ph.D. Agustín González Valenzuela

---

---

# CONTENIDOS

Glosario	12
<b>1. Estado del arte y alternativas de solución</b>	<b>17</b>
1.1. Estado del Arte	17
1.1.1. Alternativas de solución	19
<b>2. Requerimientos de solución</b>	<b>20</b>
2.1. Antecedentes y alcance	20
2.2. Requerimientos generales del sistema	20
2.3. Variantes de solución	21
2.3.1. Variante A: LoRa + batería secundaria + aporte solar	21
2.3.2. Variante B: LoRa + batería primaria	21
2.3.3. Variante C: NB-IoT + batería secundaria + aporte solar	22
2.3.4. Variante D: NB-IoT + batería primaria	22
2.4. Almacenamiento y visualización para LoRaWAN en servidor local	22
2.5. Programa recomendador de configuración	22
<b>3. Arquitectura de hardware</b>	<b>24</b>
3.1. Visión general de la arquitectura	24
3.2. Herramienta de diseño electrónico	25
3.3. Variante A: LoRa + batería secundaria + aporte solar	26
3.3.1. Microcontrolador (MCU)	26
3.3.2. Interfaz de arranque y reinicio	27
3.3.3. Programación y depuración	28
3.3.4. Transceptor LoRa y antena	29
3.3.5. Reloj en tiempo real (RTC) externo	29
3.3.6. Alimentación: “ <i>harvesting</i> ”, conversión y conmutación	30
3.3.7. Cargador y Cosechadora	30
3.3.8. Convertidor principal.	31
3.3.9. Conmutación de cargas.	32
3.3.10. Almacenamiento local	33
3.3.11. Sensores ambientales	33
3.3.12. Conectividad de sensores y periféricos	34
3.3.13. Interfaz USB-C y protección	35
3.3.14. Protección ambiental y montaje	36
3.4. Variante B: LoRa + batería primaria	36
3.5. Variante C: NB-IoT + batería secundaria + aporte solar	37
3.6. Variante D: NB-IoT + batería primaria	39
3.7. PCB LoRa + solar	40
3.8. PCB LoRa	42



3.9. PCB NB-IoT . . . . .	44
3.10. PCB NB-IoT + solar . . . . .	48
3.10.1. Selección de baterías y justificación . . . . .	50
<b>4. Desarrollo de firmware</b>	<b>52</b>
4.1. Plataforma de desarrollo . . . . .	52
4.2. Arquitectura de software y gestión de energía . . . . .	52
4.2.1. Fuentes de reactivación (" <i>wake-up sources</i> ") . . . . .	53
4.3. Sensores y periféricos . . . . .	53
4.3.1. Pluviómetro (interrupción externa) . . . . .	53
4.3.2. Sensores ambientales (I <sup>2</sup> C) . . . . .	54
4.3.3. Reloj de tiempo real (RV-3028) . . . . .	54
4.4. Almacenamiento local (microSD) . . . . .	54
4.5. Comunicación LoRaWAN . . . . .	54
4.5.1. Lógica de transmisión . . . . .	55
4.6. Interfaz USB de configuración (Microshell) . . . . .	56
4.7. Implementación de <i>drivers</i> y utilidades . . . . .	56
4.7.1. Driver TMP117: lectura de registros, configuración y conversión a temperatura	56
4.7.2. Driver RV-3028: fecha/hora, configuración de respaldo y generación de eventos	57
4.7.3. Microshell USB-CDC: parseo de comandos y despacho . . . . .	59
4.7.4. Driver BME280: configuración <i>forced</i> y lectura compensada . . . . .	60
4.7.5. Registro en microSD y secuencia de reactivación de periféricos tras <i>Stop Mode 2</i>	61
<b>5. Recepción de datos</b>	<b>64</b>
5.1. Gateway y servidor LoRaWAN local . . . . .	64
5.2. Servicios de software utilizados . . . . .	64
5.3. Configuración regional AU915 . . . . .	65
5.4. Alta de gateway y dispositivo en ChirpStack . . . . .	65
5.5. Visualización . . . . .	66
<b>6. Herramienta de selección de arquitectura</b>	<b>67</b>
6.1. Arquitectura general de la aplicación . . . . .	67
6.2. Capa de datos y registro de variantes . . . . .	68
6.3. Modelo de producción solar . . . . .	68
6.4. Motor geoespacial para conectividad celular . . . . .	69
6.5. Lógica de factibilidad y cálculo de puntaje . . . . .	70
6.6. Interfaz de usuario y resultados . . . . .	71
<b>7. Fabricación y montaje</b>	<b>75</b>
7.1. Lista de materiales (BOM) . . . . .	75
7.2. Orden de fabricación . . . . .	75
7.3. Montaje del prototipo . . . . .	76
<b>8. Carcasa, prensaestopas y monturas mecánicas</b>	<b>79</b>
8.1. Salida de cables y conectividad hacia sensores externos . . . . .	80
8.2. Fabricación de arneses y terminaciones . . . . .	80
8.3. Stevenson Screen para sensores ambientales . . . . .	81
8.4. Montura interna para PCB . . . . .	82

8.5. Montura para panel solar . . . . .	82
<b>9. Pruebas experimentales</b>	<b>84</b>
9.1. Verificación de hardware y corrección de fallas . . . . .	84
9.1.1. Revisión del panel solar: sustitución y <i>clamping</i> de tensión . . . . .	87
9.2. Verificación de Firmware . . . . .	89
9.3. Pruebas de consumo . . . . .	91
9.3.1. Metodología de medición . . . . .	94
9.3.2. Consumo promedio y comentario sobre dimensionamiento: batería vs./ aporte solar . . . . .	95
9.3.3. Registro ambiental: temperatura, humedad y presión . . . . .	96
9.4. Conclusiones . . . . .	100
9.5. Trabajo futuro . . . . .	101
9.5.1. Validación de hardware en variantes restantes . . . . .	101
9.5.2. Maduración de software y soporte completo multi-variante . . . . .	102
9.5.3. Despliegue multi-nodo y validación en terreno . . . . .	102
<b>A. Guía de replicación del sistema de recepción</b>	<b>107</b>
A.1. Instalación de Raspberry Pi OS (64-bit) . . . . .	107
A.2. Confirmación de IP de red (host) . . . . .	107
A.3. Instalación de Docker y Docker Compose . . . . .	107
A.4. Descarga y ejecución del stack ChirpStack . . . . .	108
A.5. Ajustes de región (AU915) en ChirpStack . . . . .	108
A.6. Gateway Bridge (Semtech UDP) para AU915_0 . . . . .	108
A.7. Verificación de puerto UDP 1700 . . . . .	109
A.8. Configuración del Dragino LPS8N (AU915, Semtech UDP) . . . . .	109
A.9. Confirmación de llegada de paquetes al servidor . . . . .	109
A.10. Confirmación de publicación MQTT . . . . .	110
A.11. Configuración en la interfaz web de ChirpStack . . . . .	110
A.12. Respaldo de configuración . . . . .	110
<b>B. Guía de replicación: visualización local en Raspberry Pi con InfluxDB y Grafana</b>	<b>111</b>
B.1. Requisitos y supuestos . . . . .	111
B.2. Estrategia de integración . . . . .	111
B.3. Procedimiento paso a paso . . . . .	112
B.3.1. 1) Acceder al directorio del stack . . . . .	112
B.3.2. 2) Extender el <code>docker-compose.yml</code> . . . . .	112
B.3.3. 3) Levantar el stack actualizado . . . . .	113
B.3.4. 4) Verificación de estado . . . . .	113
B.3.5. 5) Acceso desde la red local . . . . .	113
B.3.6. 6) Inicialización y credenciales . . . . .	113
B.3.7. 7) Conexión de Grafana a InfluxDB . . . . .	114
B.3.8. 8) Arranque automático tras reinicio . . . . .	114
B.4. Notas prácticas de operación . . . . .	114
<b>C. Documentación de diseño consultada</b>	<b>115</b>

**D. BOM y estimación de costo de fabricación (lote de 10 unidades) 117**  
D.1. Alcance y criterios de costeo . . . . . 117  
D.2. Costos adicionales no incluidos en el BOM principal . . . . . 117  
D.3. Costo de manufactura de placas y batería . . . . . 118  
D.4. Resumen de costos y total para 10 unidades . . . . . 118  
D.5. Tabla de materiales (BOM) . . . . . 118



---

---

# FIGURAS

3.1.	Diagrama de bloques general de la arquitectura para las cuatro variantes. . . . .	25
3.2.	Esquemático alimentación microcontrolador en software KiCad. . . . .	27
3.3.	Esquemático pulsadores microcontrolador en software KiCad. . . . .	28
3.4.	Esquemático pines de programación microcontrolador en software KiCad. . . . .	29
3.5.	Esquemático modulo LoRa en software KiCad. . . . .	29
3.6.	Esquemático RTC en software KiCad. . . . .	30
3.7.	Esquemático BQ25570 en software KiCad. . . . .	31
3.8.	Esquemático TPS62840DLCR en software KiCad. . . . .	32
3.9.	Esquemático TPS22916CYFPT para sensores y transmisión en software KiCad. . . . .	32
3.10.	Esquemático socket microSD en software KiCad. . . . .	33
3.11.	Esquemático BME280 en software KiCad. . . . .	34
3.12.	Esquemático TMP117 en software KiCad. . . . .	34
3.13.	Esquemático de conectores externos en KiCad. . . . .	35
3.14.	Esquemático conector USB y proteccion en software KiCad. . . . .	36
3.15.	Esquemático lectura de estado de batería y alimentacion en software KiCad. . . . .	37
3.16.	Esquemático del bloque NB-IoT en KiCad: BC660KGLAA, conexión RF por U.FL, control de reset con 2N7002, red de alimentación y desacoplos del módem. . . . .	39
3.17.	Esquemático <i>SIM holder</i> DT1042-04SO-7 con acondicionamiento de líneas. . . . .	39
3.18.	PCB LoRa + Solar. . . . .	41
3.19.	Modelo 3D PCB LoRa + Solar. . . . .	42
3.20.	PCB LoRa. . . . .	43
3.21.	Modelo 3D de la PCB LoRa. . . . .	44
3.22.	PCB NB-IoT. . . . .	46
3.23.	Modelo 3D de la PCB NB-IoT. . . . .	47
3.24.	PCB NB-IoT + solar. . . . .	49
3.25.	Modelo 3D de la PCB NB-IoT + solar. . . . .	50
4.1.	Máquina de estados finitos de la arquitectura de firmware del <i>data logger</i> . . . . .	53
5.1.	Interfaz web de ChirpStack en la Raspberry Pi mostrando el estado del <i>gateway</i> . . . . .	66
6.1.	Diagrama de alto nivel de la herramienta de selección: entradas de usuario, capa de datos, motores de cálculo y salidas. . . . .	68
6.2.	Modelo de producción solar utilizado por la herramienta. Se calcula una estimación promedio anual y una estimación conservadora asociada a condiciones de invierno. . . . .	69
6.3.	Secuencia de evaluación geoespacial: filtrado rápido con índice espacial y verificación precisa de distancia sobre un subconjunto de torres candidatas. . . . .	70
6.4.	Flujo de evaluación: compatibilidad por comunicación, verificación de conectividad, balance energético y asignación de puntaje. . . . .	71
6.5.	Ejecución de la herramienta de selección: entradas y visualización inicial. . . . .	72
6.6.	Inicio de selección. . . . .	72

6.7.	Ejecución de la herramienta de selección: Resultados. . . . .	73
6.8.	Interfaz de gestión del registro de placas: formulario para añadir nuevas variantes. . .	74
7.1.	Orden de fabricación de PCB. . . . .	76
7.2.	Montaje de componentes SMD: PCB fijada en el sistema de <i>pick and place</i> con pasta aplicada y componentes posicionados. . . . .	77
7.3.	Resultados de montaje: (a) placa principal tras completar ensamble SMD, (b) subplaca <i>breakaway</i> de sensores tras reflow, y (c) placa principal finalizada con conectores montados mediante soldadura manual. . . . .	78
8.1.	Carcasa PNB-30417-KO con puntos de salida de cables mediante prensaestopas. . . .	79
8.2.	Carcasa PNB-30417-KO con puntos de salida de cables mediante prensaestopas y antena. . . . .	80
8.3.	Arnés ensamblado para interconexión entre la placa principal y periféricos externos. .	81
8.4.	Stevenson Screen impreso en 3D para el prototipo, con recubrimiento de protección UV. . . . .	81
8.5.	Montura interna impresa en 3D para fijación de la PCB dentro del gabinete (M2.5). . .	82
8.6.	Montura para panel solar y referencia geométrica del panel utilizado (P105, 6V–5.5W, ETFE). . . . .	83
9.1.	Corrección de esquemático: adición de resistencia <i>pull-down</i> en el pin de configuración/control del TPS62840DLCR para evitar estados flotantes. . . . .	84
9.2.	Corrección de esquemático: adición de resistencias <i>pull-down</i> en pines de control del TPS22916CYFPR para definir el estado durante arranque y transitorios. . . . .	85
9.3.	Corrección de esquemático: conector de lluvia y conectores externos, indicando la reasignación necesaria para evitar conexión errónea a tierra. . . . .	85
9.4.	Prototipo con correcciones aplicadas: conexiones externas de reparación y componentes añadidos para estabilizar el arranque y el subsistema de potencia. . . . .	87
9.5.	Protección en la entrada del panel: diodo Zener en paralelo para limitar $V_{IN}$ y proteger el BQ25570. . . . .	88
9.6.	Archivo <code>DATA.csv</code> almacenado en la microSD, abierto en computador para verificación de escritura. . . . .	90
9.7.	Medición de corriente en <i>Stop Mode 2</i> mediante Joulescope (estado de suspensión profunda con periféricos desenergizados). . . . .	92
9.8.	Captura de consumo asociada a la interrupción del pluviómetro: evento breve que actualiza el conteo sin ejecutar el ciclo completo de adquisición. . . . .	93
9.9.	Captura de consumo durante el <i>wake-up</i> periódico del RTC (60 s): energización de sensores, lectura, escritura en microSD y retorno a suspensión. . . . .	94
9.10.	Serie temporal de temperatura registrada por el prototipo (TMP117) entre el 26/01/2026 12:42 y el 28/01/2026 07:56. . . . .	97
9.11.	Serie temporal de humedad relativa registrada por el prototipo (BME280) en el periodo de prueba (26/01/2026–28/01/2026). . . . .	97
9.12.	Serie temporal de presión atmosférica registrada por el prototipo (BME280) en el periodo de prueba (26/01/2026–28/01/2026). . . . .	98
9.13.	Temperatura referencial viña del mar obtenida de redmeteo.cl - Quinta Vergara. . . .	98
9.14.	Presión referencial viña del mar obtenida de redmeteo.cl - Quinta Vergara. . . . .	99



---

---

# RESUMEN

Este trabajo presenta el diseño e implementación de una plataforma autónoma de adquisición de variables meteorológicas orientada a operación de largo plazo y bajo consumo. El objetivo principal fue desarrollar un registrador ambiental capaz de medir temperatura, humedad, presión y precipitación, almacenar localmente los datos y habilitar transmisión inalámbrica mediante LoRaWAN, priorizando la eficiencia energética mediante el uso de modos de bajo consumo del microcontrolador **STM32L476RG** y la desconexión selectiva de periféricos. Para abordar distintos escenarios de despliegue se definieron variantes de arquitectura (según subsistema energético y tecnología de comunicación), fabricándose y ensamblándose en esta etapa la variante **LoRa + batería secundaria + aporte solar** como plataforma de validación integral.

La metodología incluyó el diseño de PCB en KiCad, fabricación y montaje con proceso *stencil + reflow*, y el desarrollo de firmware en **STM32CubeIDE/CubeMX** utilizando HAL, FatFS para microSD y USB CDC para configuración. El software se organizó como una máquina de estados enfocada en maximizar el tiempo en *Stop Mode 2*, definiendo como fuentes de reactivación la alarma/temporizador del RTC externo **RV-3028**, interrupción por pluviómetro (EXTI) y detección de VBUS. Se desarrollaron *drivers* I<sup>2</sup>C para **TMP117** y **BME280**, y un driver del **RV-3028** con conversiones BCD, configuración de respaldo y rutinas de recuperación tras ciclos de apagado/encendido del riel de sensores. Para robustecer la operación en bajo consumo, se incorporó una secuencia explícita de apagado e inicio que incluye *de-init* y reinicialización de I<sup>2</sup>C/SPI, estacionamiento de pines en modo análogo y recuperación del bus I<sup>2</sup>C, mitigando pérdidas de estado y bloqueos del RTC al retornar desde suspensión.

Los resultados de verificación confirmaron el despertar periódico configurado cada **60 s**, la reactivación confiable desde *Stop Mode 2*, la lectura consistente de sensores y la escritura repetible del archivo **DATA.csv** en microSD. Además, durante la depuración se identificó y corrigió una condición de tensión residual en el riel de sensores asociada a *back-feeding* por líneas de comunicación, resuelta mediante el apagado lógico de periféricos, el estacionamiento de pines y la secuencia de reinicialización. Para la capa de red, se integró una adaptación de **LoRaMAC-node** sobre HAL para el módulo **LR62E**, planificando una política de transmisión por promedios mediante un contador de muestras (por defecto  $N = 15$ ) para reducir consumo. La infraestructura de backend se implementó con **ChirpStack** en una **SBC**, verificando la operación del *gateway* y el flujo de paquetes a nivel de red. Finalmente, la metodología de caracterización energética consideró observación inicial con fuente de laboratorio y medición de alta resolución con **Joulescope**, registrando perfiles de consumo en suspensión, eventos de interrupción y ciclos de muestreo/almacenamiento (y transmisión, cuando corresponde).

En conclusión, se logró una plataforma funcional para adquisición y registro local con operación periódica en bajo consumo, con una arquitectura de firmware robusta frente a ciclos de suspensión profunda y reactivación. La integración LoRaWAN quedó establecida a nivel de pila y backend, habilitando la extensión hacia pruebas sistemáticas de transmisión y evaluación comparativa entre variantes de hardware en trabajos posteriores.

**Keywords:** data logger, ultra-low-power, STM32L476, Stop Mode 2, LoRaWAN, LoRaMAC-node, ChirpStack, RV-3028, TMP117, BME280, microSD, FatFS, USB CDC, energy harvesting, load switch TPS22916.

---

---

# ABSTRACT

This work presents the design and implementation of an autonomous platform for meteorological variable acquisition aimed at long-term, low-power operation. The main objective was to develop an environmental data logger capable of measuring temperature, humidity, pressure, and precipitation, storing data locally, and enabling wireless transmission via LoRaWAN, while prioritizing energy efficiency through the use of low-power modes of the **STM32L476RG** microcontroller and selective disconnection of peripherals. To address different deployment scenarios, architecture variants were defined (according to the energy subsystem and communication technology), and in this stage the **LoRa + secondary battery + solar contribution** variant was fabricated and assembled as an integral validation platform.

The methodology included PCB design in KiCad, fabrication and assembly using a *stencil + reflow* process, and firmware development in **STM32CubeIDE/CubeMX** using HAL, FatFS for the microSD card, and USB CDC for configuration. The software was organized as a state machine focused on maximizing time in *Stop Mode 2*, defining as wake-up sources the alarm/timer of the external **RV-3028** RTC, a rain gauge interrupt (EXTI), and VBUS detection. I<sup>2</sup>C *drivers* were developed for the **TMP117** and **BME280**, and an **RV-3028** driver including BCD conversions, backup configuration, and recovery routines after sensor-rail power cycles. To improve robustness in low-power operation, an explicit power-down and start-up sequence was added, including I<sup>2</sup>C/SPI de-initialization and re-initialization, parking pins in analog mode, and I<sup>2</sup>C bus recovery, mitigating loss of state and RTC lockups when returning from deep sleep.

Verification results confirmed periodic wake-up configured every **60 s**, reliable reactivation from *Stop Mode 2*, consistent sensor readings, and repeatable writing of the **DATA.csv** file to the microSD card. In addition, during debugging a residual-voltage condition on the sensor rail associated with communication-line back-feeding was identified and corrected by logically disabling peripherals, parking pins, and applying the re-initialization sequence. For the network layer, an adaptation of **LoRaMAC-node** over HAL was integrated for the **LR62E** module, planning a batch transmission policy using a sample counter (default  $N = 15$ ) to reduce power consumption. The backend infrastructure was implemented with **ChirpStack** on an SBC, verifying gateway operation and packet flow at the network level. Finally, the energy characterization methodology included initial observation with a bench power supply and high-resolution measurement with a **Joulescope**, recording current profiles during sleep, interrupt events, and sampling/storage cycles (and transmission, when applicable).

In conclusion, a functional platform for acquisition and local logging with periodic low-power operation was achieved, with a firmware architecture robust to deep-sleep cycles and wake-up. LoRaWAN integration was established at both stack and backend levels, enabling extension toward systematic transmission testing and comparative evaluation among hardware variants in future work. **Keywords:** data logger, ultra-low-power, STM32L476, Stop Mode 2, LoRaWAN, LoRaMAC-node, ChirpStack, RV-3028, TMP117, BME280, microSD, FatFS, USB CDC, energy harvesting, TPS22916 load switch.

---

---

# GLOSARIO

- ADC** *Analog-to-Digital Converter*. Convertidor analógico–digital.
- AES-128** Cifrado simétrico de 128 bits utilizado en LoRaWAN para autenticación y confidencialidad.
- API** *Application Programming Interface*. Interfaz de programación de aplicaciones.
- AU915** Plan regional de frecuencias LoRaWAN en banda 915 MHz (configuración de canales usada en la memoria).
- BSM** *Backup Switchover Mode*. Modo de conmutación a respaldo en el RTC RV-3028.
- CDC** *Communications Device Class*. Clase USB que emula un puerto serial virtual.
- CSV** *Comma-Separated Values*. Formato de archivo de texto con campos separados por comas.
- DC–DC** Convertidor conmutado de corriente continua a corriente continua.
- Deep Sleep** Modo de bajo consumo en el que el microcontrolador detiene la mayoría de sus funciones para ahorrar energía.
- EEPROM** *Electrically Erasable Programmable Read-Only Memory*. Memoria no volátil reprogramable.
- eDRX** *extended Discontinuous Reception*. Modo de ahorro energético en NB-IoT/LTE-M que reduce la frecuencia de escucha en recepción.
- Energy Harvesting** Proceso de captura y almacenamiento de energía de fuentes externas (como la solar).
- ENIG** *Electroless Nickel Immersion Gold*. Terminación superficial níquel/oro por inmersión.
- EXTI** *External Interrupt*. Interrupción externa del microcontrolador.
- FatFS** Sistema de archivos ligero para medios FAT (microSD) utilizado como *middleware*.
- Firmware** Programa de bajo nivel que controla el funcionamiento de un sistema embebido y su hardware asociado.
- FSM** *Finite State Machine*. Máquina de estados finitos.
- FR-4** Sustrato estándar de PCB basado en fibra de vidrio y resina epóxica.
- Gateway** Dispositivo que actúa como puente de comunicación entre nodos finales y la red (p. ej., entre LoRaWAN y el *backend*).
- GHI** *Global Horizontal Irradiance*. Irradiancia solar global sobre un plano horizontal.

**GPIO** *General-Purpose Input/Output*. Pines de entrada/salida de propósito general.

**HAL** *Hardware Abstraction Layer*. Capa de abstracción de hardware de STM32.

**I<sup>2</sup>C** *Inter-Integrated Circuit*. Bus serial síncrono de dos hilos (SCL/SDA).

**IDE** *Integrated Development Environment*. Entorno de desarrollo integrado.

**IP65 / IP67 / IP68** Grados de protección de envoltantes frente a ingreso de polvo y agua (norma IEC 60529).

**IRQ** *Interrupt Request*. Señal o evento de interrupción.

**JTAG / SWD / ST-LINK** Interfaces y herramientas de depuración/programación para microcontroladores.

**KiCad** Suite de diseño electrónico utilizada para esquemáticos y PCB.

**LDO** *Low-Dropout Regulator*. Regulador lineal de baja caída.

**LoRa** *Long Range*. Modulación de espectro ensanchado basada en chirp.

**LoRaWAN** *LoRa Wide Area Network*. Protocolo de red sobre LoRa para comunicaciones LPWAN.

**LPWAN** *Low-Power Wide-Area Network*. Redes de amplio alcance y bajo consumo.

**MCU** *Microcontroller Unit*. Microcontrolador.

**MPPT** *Maximum Power Point Tracking*. Seguimiento del punto de máxima potencia para optimizar la extracción de energía desde un panel solar.

**MQTT** *Message Queuing Telemetry Transport*. Protocolo de mensajería ligero para comunicaciones M2M e IoT.

**MSB** *Most Significant Byte/Bit*. Byte/bit más significativo.

**NB-IoT** *Narrowband Internet of Things*. Tecnología celular LPWAN.

**NVM** *Non-Volatile Memory*. Memoria no volátil.

**OTAA** *Over-The-Air Activation*. Método de activación LoRaWAN mediante procedimiento de unión.

**Payload** Carga útil: datos transportados por una trama o paquete de comunicación (sin considerar cabeceras).

**PCB** *Printed Circuit Board*. Placa de circuito impreso.

**POR** *Power-On Reset*. Reinicio por encendido.

**PSM** *Power Saving Mode*. Modo de ahorro energético en NB-IoT que reduce la actividad de radio manteniendo el registro en red.

**RF** *Radio Frequency*. Radiofrecuencia.

**RTC** *Real-Time Clock*. Reloj de tiempo real.

**SBC** *Single Board Computer*. Computador de placa única (p. ej., Raspberry Pi) usado para aplicaciones embebidas o *gateways*.

**SIM** *Subscriber Identity Module*. Módulo de identidad de abonado para conectividad celular.

**SMD** *Surface-Mount Device*. Componente de montaje superficial.

**SPI** *Serial Peripheral Interface*. Bus serial síncrono (SCK/MISO/MOSI/CS).

**Stop Mode 2** Modo de bajo consumo del STM32 que detiene la CPU y maximiza ahorro energético manteniendo fuentes de *wake-up*.

**TVS** *Transient Voltage Suppressor*. Diodo supresor de transitorios para protección frente a ESD/-sobretensiones.

**UDP** *User Datagram Protocol*. Protocolo de transporte sin conexión (p. ej., usado en *gateway bridge*).

**UID** *Unique Identifier*. Identificador único (por dispositivo o componente).

**USB** *Universal Serial Bus*. Interfaz de comunicación y alimentación.

**VBUS** Línea de 5 V del bus USB, usada para detección de conexión.

**WFI** *Wait For Interrupt*. Instrucción de CPU para entrar en espera hasta una interrupción.

**Load switch** Interruptor de carga (p. ej., TPS22916) que conecta/desconecta rieles de alimentación a periféricos.

**Wake-up** Evento que reanuda la ejecución desde un modo de bajo consumo.

**Bus recovery** Secuencia para liberar un bus I<sup>2</sup>C bloqueado, típicamente generando pulsos en SCL y forzando STOP.

**One-shot** Modo de medición en el que el sensor realiza una única conversión bajo demanda y vuelve a reposo.

**Forced mode** Modo del BME280 en el que el sensor mide sólo cuando se gatilla una conversión.

**Backend LoRaWAN** Infraestructura de red (p. ej., ChirpStack) para gestión de *gateways*, dispositivos, claves y enrutamiento de tramas.

*V* Voltaje.

*I* Corriente.

*P* Potencia.

*E* Energía.

*t* Tiempo.

*N* Número de muestras acumuladas antes de transmisión (parámetro de agrupación).

**Hz** Hertz, unidad de frecuencia.

**s** Segundo, unidad de tiempo.

**mA** Miliampere, unidad de corriente.

**$\mu$ A** Microampere, unidad de corriente.

**nA** Nanoampere, unidad de corriente.

**mV** Milivolt, unidad de voltaje.

**°C** Grado Celsius, unidad de temperatura.

**Pa** Pascal, unidad de presión.

**hPa** Hectopascal, unidad de presión (1 hPa = 100 Pa).

**STM32L476RG** Microcontrolador principal utilizado en el prototipo.

**TMP117** Sensor de temperatura de alta precisión (I<sup>2</sup>C).

**BME280** Sensor de presión, humedad y temperatura (I<sup>2</sup>C).

**RV-3028** RTC externo de ultra bajo consumo con registros en BCD e interrupciones de alarma/-  
temporizador.

**LR62E** Módulo LoRa basado en SX1262 utilizado para comunicación LoRaWAN.

**TPS22916** Interruptor de carga utilizado para conmutar rieles de sensores y comunicaciones.

**microSD** Medio de almacenamiento removible utilizado para registro local.

**ChirpStack** Plataforma *open-source* para administración de redes LoRaWAN.

**SX1262 / SX126x** Familia de transceptores LoRa de Semtech utilizada en módulos de comunicación (p. ej., LR62E).



# ESTADO DEL ARTE Y ALTERNATIVAS DE SOLUCIÓN

El diseño de sistemas de monitoreo ambiental para despliegues prolongados requiere considerar tanto los avances tecnológicos disponibles como las limitaciones prácticas que imponen los escenarios de aplicación. En este sentido, resulta fundamental analizar el estado del arte en lo referido a plataformas de adquisición de datos, tecnologías de comunicación y estrategias de gestión energética, con el propósito de identificar fortalezas y debilidades en las soluciones existentes.

El estado del arte muestra que los *data loggers* convencionales disponibles en el mercado se caracterizan por su capacidad de registrar múltiples variables ambientales con alta precisión. No obstante, presentan limitaciones asociadas a los costos de implementación y la dependencia de infraestructura de comunicaciones establecida. Asimismo, su diseño cerrado restringe la incorporación de nuevas tecnologías de sensado o transmisión, lo que reduce su adaptabilidad frente a distintos contextos de uso.

En paralelo, la literatura científica y técnica evidencia un creciente interés en el desarrollo de arquitecturas modulares y abiertas, basadas en microcontroladores de bajo consumo, esquemas de recolección de energía y tecnologías de comunicación inalámbrica de largo alcance, como LoRa y NB-IoT. Estas propuestas buscan enfrentar el desafío de la autonomía energética y la operación en entornos remotos, aunque muchas de ellas permanecen en etapas experimentales o carecen de validaciones extensivas en condiciones reales de terreno.

## 1.1. Estado del Arte

El desarrollo de dispositivos Internet of Things (IoT) de bajo consumo ha sido ampliamente abordado en la literatura, particularmente en contextos donde la eficiencia energética resulta crítica para prolongar la autonomía de sistemas alimentados por batería. Entre los referentes más influyentes en el ámbito del diseño electrónico se encuentra la obra *The Art of Electronics* de Horowitz y Hill [1], que presenta fundamentos esenciales para la reducción del consumo en circuitos analógicos y digitales. Aunque algunas de las técnicas descritas han sido superadas por avances tecnológicos recientes, muchas de sus recomendaciones siguen siendo vigentes, como la utilización de fuentes conmutadas, la reducción de los tiempos activos de los componentes y la implementación de osciladores de baja frecuencia.

La investigación reciente ha explorado activamente la implementación de sistemas IoT de bajo consumo empleando microcontroladores y tecnologías de comunicación modernas. Paiva et al. [2] desarrollan un dispositivo basado en NB-IoT orientado a aplicaciones de monitoreo, comparando enfoques basados en FreeRTOS y programación bare-metal. Su análisis demuestra que el segundo permite un control más preciso sobre los ciclos de actividad y suspensión, logrando una eficiencia superior en términos de consumo energético. En la misma línea, Dordevic [3] presenta una plataforma de sensado NB-IoT de ultra bajo consumo, en la cual se incorporan técnicas de optimización tanto en

hardware como en software, junto con un uso intensivo de *duty cycling* para maximizar la autonomía del sistema. A diferencia de tales propuestas, en las que el diseño de las placas se externaliza, el presente trabajo contempla el desarrollo completo de la arquitectura electrónica, lo que asegura un mayor control sobre el consumo energético y la flexibilidad del sistema. Además, mientras Dordevic emplea un Power Profiler Kit II [4] para la medición, aquí se utiliza un Joulescope [5], instrumento que permite una caracterización detallada y continua del perfil de consumo, con mayor granularidad y precisión en ciclos de actividad y suspensión.

El microcontrolador STM32L476 ha demostrado su eficacia en aplicaciones específicas de sensado de bajo consumo. Fan et al [6], por ejemplo, diseñan un anemómetro ultrasónico basado en este dispositivo, evidenciando la capacidad de combinar tiempos cortos de muestreo con modos de suspensión profundos, lo que permite reducir el consumo en aplicaciones activas. En paralelo, Hidalgo et al [7], proponen un sistema de cómputo en el borde orientado al monitoreo estructural, que integra bajo costo, eficiencia energética y capacidad de procesamiento local. Su enfoque subraya la relevancia de la arquitectura del microcontrolador y de las políticas de administración energética en aplicaciones que requieren procesamiento distribuido, lo que se alinea con la posibilidad de incorporar algoritmos de preprocesamiento en el presente proyecto.

En el ámbito del monitoreo ambiental, García et al [8], desarrollan una plataforma multisensor basada en LoRa que captura, procesa y transmite datos de temperatura, humedad, luminosidad e imágenes, combinando distintos microprocesadores en función de las necesidades energéticas de cada sensor. Su propuesta destaca por la escalabilidad y versatilidad, aspectos clave para arquitecturas modulares y configurables, como la que se plantea en este trabajo. También se identifican iniciativas de *data loggers* de código abierto, como el proyecto Untammed Data Logger [9], desarrollado con un microcontrolador ESP32 y almacenamiento en tarjeta microSD. Aunque representa un ejemplo de robustez y simplicidad en despliegues instrumentales acotados en el tiempo, carece de modularidad en comunicaciones y optimización para ultra bajo consumo, condiciones centrales en este proyecto.

Durante la etapa preliminar de desarrollo se evaluaron además plataformas de hardware existentes como posibles bases para el prototipo. Entre ellas, la Bee Data Logger de Smart Bee Designs [10], basada en ESP32, permitió validar conceptos iniciales de registro de datos y transmisión mediante LoRaWAN. Sin embargo, su autonomía se vio limitada por el consumo energético del microcontrolador, especialmente en modos de reposo y durante el uso de WiFi. Otra opción considerada fue la H32 Basic de Joachim Baumann [11], diseñada como placa modular para aplicaciones LPWAN. Si bien su versatilidad inicial resultó atractiva, fue descartada debido a un consumo superior al requerido para aplicaciones de ultra bajo consumo y a la falta de mantenimiento activo de la plataforma, lo que representa un riesgo de sostenibilidad técnica a largo plazo.

Complementariamente, se ha destacado en la literatura el uso de herramientas prácticas para la evaluación energética, como el STM32CubeMX [12] Power Calculator y equipos de laboratorio tales como Joulescope. Estos permiten comparar estimaciones teóricas con mediciones reales, optimizando el diseño desde las etapas tempranas de desarrollo. Del mismo modo, experiencias reportadas por comunidades abiertas, como los Hackaday Environmental Sensors, muestran que con un manejo adecuado de ciclos de sueño y muestreo es posible alcanzar varios años de autonomía con baterías de capacidad limitada, reforzando la importancia de una integración armónica entre hardware y firmware.

Finalmente, en el plano de la usabilidad, soluciones como los Bluetooth Data Loggers de Dickson [13] y la plataforma EnviroDIY Mayfly [14] aportan antecedentes relevantes en cuanto a facilidad de operación. Estas propuestas permiten modificar parámetros de funcionamiento como intervalos de muestreo o identificadores de dispositivo mediante aplicaciones móviles, sin necesidad de conexión por cable, lo que facilita la operación en terreno sin comprometer la eficiencia energética.

### 1.1.1. Alternativas de solución

A partir de este análisis, se planteó la necesidad de explorar alternativas de solución que conciliaran eficiencia energética, flexibilidad arquitectónica y escalabilidad funcional. En la etapa de exploración se evaluaron, en primer lugar, plataformas de desarrollo comerciales como Arduino y ESP32, que si bien ofrecen facilidad de programación y amplia documentación, resultaron inadecuadas por su consumo energético elevado y su escasa robustez para aplicaciones de largo plazo. Posteriormente, se estudiaron microcontroladores de la familia ARM Cortex-M, que permiten un diseño de hardware a medida con un consumo reducido, a costa de una mayor complejidad en la implementación. Finalmente, se consideraron arquitecturas modulares que integran distintas opciones de sensado y comunicación, lo que permite escalar el sistema según las necesidades de cada aplicación, aunque requieren un diseño más detallado de integración entre hardware y firmware.

# REQUERIMIENTOS DE SOLUCIÓN

## 2.1. Antecedentes y alcance

Los requerimientos que se presentan en esta sección corresponden al conjunto preliminar con el que se inició el proyecto. Estos requisitos de partida, definidos antes de cerrar la arquitectura final, establecen las funciones esenciales del sistema: registro local de datos, temporización confiable mediante RTC, identificación única del dispositivo, operación autónoma de largo plazo sin conexión a la red eléctrica, interfaz USB para configuración y descarga, opciones de transmisión mediante tecnologías LPWAN (LoRaWAN o NB-IoT) y una carcasa apta para uso a la intemperie junto al pluviómetro. A partir de este marco inicial, y a medida que se profundizó en el análisis de consumo, cobertura y mantenibilidad, dichos requerimientos se adaptaron y refinaron para soportar variantes de hardware con diferentes combinaciones de comunicación y energía. El resultado es una arquitectura modular con cuatro placas base alternativas, todas ellas compartiendo una base de firmware común para minimizar la duplicación y facilitar la evolución del sistema.

## 2.2. Requerimientos generales del sistema

En esta sección se consolidan los criterios transversales que toda variante debe satisfacer, independientemente de la tecnología de enlace o esquema de alimentación, pues describen la experiencia funcional mínima, el desempeño esperado y las condiciones de diseño que garantizan fiabilidad y bajo consumo.

En términos funcionales, el sistema debe muestrear y registrar las variables ambientales pertinentes (pluviómetro de balancín, temperatura, humedad y presión ambiental) con una frecuencia configurable típicamente entre uno y sesenta minutos. Cada muestra se almacena junto a una marca de tiempo provista por un RTC y al identificador único del dispositivo, y puede exportarse a través de una consola serie por USB tanto para descarga de datos en formatos legibles (CSV) como para operaciones de configuración y diagnóstico. En las variantes con telemetría, la transmisión se realiza en diferentes tiempos al muestreo, los cuales deben ser configurables.

Respecto del desempeño y el consumo, se busca una autonomía efectiva de, al menos, doce meses en configuraciones a batería primaria con un perfil de muestreo de referencia y transmisión (por ejemplo, cada un minuto y quince minutos), y una operación sostenida en el tiempo para las variantes con aporte solar. En reposo profundo, con RTC activo y memoria no volátil disponible, la corriente objetivo se mantiene en el orden de decenas de microamperes, mientras que los picos de actividad (medición y transmisión) se acotan temporalmente mediante un *duty-cycle* estricto, de forma que el equipo vuelva a su modo de bajo consumo en unos pocos segundos.

El almacenamiento local se implementa sobre memoria no volátil, con retención de datos acorde a la vida útil del dispositivo y tolerancia a ciclos de escritura reiterados. La estructura de archivo es de crecimiento secuencial (*append-only*) y prioriza la integridad ante pérdidas de energía. La capacidad se dimensiona para cubrir el horizonte de operación durante la campaña incluso en ausencia de

telemetría.

En sincronización temporal, el sistema emplea un RTC, complementado cuando la variante lo permite por sincronización de red (por ejemplo, mensajes de *AppTime* en NTP sobre NB-IoT), manteniendo el error anual en márgenes compatibles con el análisis posterior de datos.

La identidad y la seguridad se abordan desde la fabricación, asociando a cada unidad un identificador único (por ejemplo, combinando el UID del microcontrolador con una etiqueta lógica de gestión). En transmisión, se utilizan los mecanismos de seguridad propios de cada tecnología (claves AES-128 en LoRaWAN; autenticación mediante SIM/eSIM en NB-IoT), y se restringe el acceso a la consola USB para prevenir modificaciones no autorizadas en terreno.

En la capa de energía, el diseño contempla, según la variante, baterías primarias o recargables y, cuando corresponde, recolección solar mediante un cargador de muy baja potencia de arranque. Se incorporan protecciones ante sobre/infra-tensión y telemetría de estado de carga para adaptar dinámicamente las políticas de muestreo o transmisión a la energía disponible.

En ambiente y mecánica, la carcasa alcanza un grado de protección mínimo IP65, con pasamuros adecuados para el pluviómetro, materiales resistentes a radiación UV. Se privilegian conectores sellados para garantizar continuidad de servicio en condiciones reales de terreno.

Finalmente, en interfaz de usuario y mantenimiento, se dispone de una consola por USB con comandos de configuración y diagnóstico, transmisión de bajo consumo, y rutinas de autodiagnóstico al arranque. El firmware se estructura sobre una base común con *drivers* intercambiables de sensores y comunicación, y contempla actualización local por USB.

## 2.3. Variantes de solución

A partir de los requerimientos generales se desarrollaron cuatro variantes de placa, que difieren en la tecnología de comunicación y en el esquema de alimentación. Todas comparten el mismo microcontrolador de ultra bajo consumo, la pila de almacenamiento local y el arnés de sensores, pero adoptan decisiones específicas de potencia y enlace de acuerdo con el escenario de despliegue.

### 2.3.1. Variante A: LoRa + batería secundaria + aporte solar

Esta variante se orienta a campañas prolongadas en zonas con cobertura LoRaWAN local, maximizando la autonomía mediante recolección fotovoltaica. El cargador *harvester* permite iniciar la carga en condiciones de irradiancia reducida y sostener el balance energético promedio, de modo que el equipo opere indefinidamente con perfiles de muestreo regulares. La comunicación se implementa en LoRaWAN. La política de transmisión se establece al momento de configurar el dispositivo pero ante baja energía disponible, el equipo reduce el ritmo de envíos o prioriza el almacenamiento local hasta recuperar condiciones favorables. Al cierre de esta variante, se espera una corriente media diaria inferior al centenar de microamperes a intervalos de muestreo del orden de un minuto, con disponibilidad continua del servicio y persistencia de datos en el servidor local.

### 2.3.2. Variante B: LoRa + batería primaria

Pensada para ubicaciones con gateway LoRaWAN cercano pero sin factibilidad de panel solar, esta opción privilegia la simplicidad y la previsibilidad del consumo con baterías primarias de alta densidad energética. Se mantiene la misma pila de comunicación y de almacenamiento que en la Variante A, pero se ajustan la periodicidad de muestreo y la ventana de transmisión para asegurar al menos un año de autonomía. Para proteger la campaña en sus etapas finales, ciertas operaciones

energéticamente demandantes como transmisiones de datos se podrían inhabilitar automáticamente cuando el sistema detecta niveles críticos de almacenamiento o de energía restante, priorizando la completitud del registro.

### 2.3.3. Variante C: NB-IoT + batería secundaria + aporte solar

Esta variante aborda escenarios con cobertura celular NB-IoT, favoreciendo la entrega directa de datos a un backend IP. Dado que el módem impone picos de corriente elevados en transmisión, el diseño eléctrico incorpora regulación de alta eficiencia y una ruta de potencia que soporte transitorios de hasta el amperio, manteniendo la estabilidad del resto de la electrónica. El subsistema energético incluye un panel fotovoltaico y un cargador adecuados para compensar el consumo de reposo y una fracción significativa de las ráfagas de transmisión. La autenticación se realiza mediante SIM/eSIM y, cuando corresponde, se complementa con mecanismos de seguridad a nivel de aplicación en la comunicación con el backend.

### 2.3.4. Variante D: NB-IoT + batería primaria

Dirigida a instalaciones donde el aporte solar no es viable, esta alternativa mantiene la conectividad NB-IoT pero con una política de transmisión más conservadora para proteger la autonomía. Se recomiendan intervalos de envío más espaciados, consolidación de mediciones en paquetes y una estrategia de *store-and-forward* que garantice la entrega cuando las condiciones de red lo permitan. Para mantenimiento, se contempla un puerto de servicio protegido (por ejemplo, USB interno) que facilita la actualización local y la inspección sin sacrificar la estanqueidad del gabinete.

## 2.4. Almacenamiento y visualización para LoRaWAN en servidor local

En las variantes con LoRa, la solución contempla una infraestructura local compuesta por un *gateway* y un *Network Server* auto-hospedado que recibe las tramas *uplink*, aplica las claves de sesión y gestiona parámetros como el *ADR* y, cuando se habilita, la sincronización temporal. Sobre esta capa se implementa una aplicación de backend que persiste los datos decodificados en una base de datos con retención suficiente para el análisis histórico de campañas, expone una API para consultas y orquesta la emisión de *downlinks* de configuración bajo reglas acotadas. La interfaz de visualización ofrece un tablero web con el estado de cada dispositivo (último contacto, tensión de batería), gráficos de series temporales y exportación de datos; además, permite definir alertas sencillas por umbrales o inactividad. En conjunto, este despliegue asegura la trazabilidad desde el sensor hasta el repositorio central y habilita la operación en terreno sin dependencia de servicios externos.

## 2.5. Programa recomendador de configuración

Como complemento de la arquitectura, se desarrolla una herramienta de recomendación que, a partir de coordenadas y metadatos de la ubicación (disponibilidad de NB-IoT, condiciones energéticas estimadas y restricciones de muestreo), sugiere la variante de hardware más adecuada junto a parámetros operacionales. El motor aplica un conjunto de reglas derivadas de los modelos de consumo y de la cobertura disponible para proponer, la tasa de datos o el *data rate* en LoRa y los parámetros **PSM/eDRX** en NB-IoT, así como la periodicidad de muestreo y de envío. La herramienta genera

un reporte con la configuración recomendada, el *BOM* preliminar correspondiente a la placa, los documentos de diseño y una estimación de autonomía bajo los supuestos de entrada. Para su uso en laboratorio y en terreno, se distribuye tanto como ejecutable autónomo (empaquetado) como en forma de script, manteniendo tiempos de cálculo breves y registro de versiones de reglas y fuentes utilizadas para asegurar trazabilidad.

# ARQUITECTURA DE HARDWARE

Esta sección describe las decisiones de diseño de hardware adoptadas para la plataforma de *data logger* de bajo consumo, incluyendo criterios de selección de componentes, integración eléctrica y consideraciones de bajo consumo. Se privilegió el uso de un microcontrolador de muy baja potencia, un subsistema de reloj robusto, almacenamiento local no volátil y, según la variante, módulos de comunicación LoRa o NB-IoT con un esquema de alimentación que combina batería (primaria o recargable) y, cuando es pertinente, aporte solar mediante “*harvesting*”.

## 3.1. Visión general de la arquitectura

La Figura 3.1 presenta un diagrama de bloques que resume la arquitectura común de hardware para las cuatro variantes desarrolladas. En el corazón del sistema se encuentra el microcontrolador, que coordina la adquisición de señales, el almacenamiento local y la comunicación de datos, mientras el subsistema de energía provee los rieles de alimentación y aplica políticas de bajo consumo acordes al perfil operativo de cada campaña. La estructura se concibe de manera modular: los bloques son desacoplables y se interconectan a través de interfaces digitales (I<sup>2</sup>C, SPI, UART, USB) y líneas de control dedicadas para habilitar o suspender dominios completos de carga cuando no se requieren.

El bloque de *gestión de energía* reúne las funciones de conversión, protección y conmutación. En las variantes con aporte solar incluye el integrado de “*energy harvesting*” y “*Maximum Power Point Tracking*” MPPT, además del convertidor reductor principal que establece el riel del sistema y los interruptores de potencia que permiten desenergizar sensores y radio cuando el dispositivo se encuentra en reposo. Este bloque recibe energía desde el *bloque batería*, donde se dispone la celda primaria o recargable según la variante y, cuando corresponde, la entrada fotovoltaica. La combinación de ambos define el presupuesto energético y habilita estrategias de ciclos de trabajos agresivos sin comprometer la integridad de datos.

El *bloque MCU* centraliza la lógica de control, el marcado de tiempo de las mediciones y la aplicación de políticas de ahorro. Se apoya en un *RTC externo* de muy baja deriva, que mantiene la marca de tiempo aun cuando el microcontrolador permanece en modos profundos de suspensión y, si es necesario, genera alarmas periódicas para reactivar la captura. La instrumentación ambiental se organiza en tres bloques funcionales: un sensor de temperatura de alta precisión, un sensor combinado de humedad y presión, y el pluviómetro de balancín para precipitación; los primeros dos se integran por I<sup>2</sup>C y el tercero por una entrada digital con acondicionamiento para conteo por interrupciones.

El *bloque de memoria* implementa el almacenamiento local no volátil, mediante tarjeta micro SD, garantizando la persistencia de los datos aun en ausencia de telemetría. Para transmisión en campo, el *bloque de transmisión* se materializa como un módulo LoRaWAN o NB-IoT según la variante, con su correspondiente interfaz digital hacia el MCU y la red RF y de desacoplos prevista por el fabricante. Las funciones de servicio se resuelven con dos bloques específicos: la *Comunicación USB* para configuración y descarga local, limitada a USB 2.0 de baja velocidad en modo dispositivo, y el *bloque de programación* para depuración y carga de firmware mediante el cabezal “*Serial Wire*”.

*Debug*” (SWD). En conjunto, estos bloques constituyen una plataforma coherente, reutilizable entre variantes.

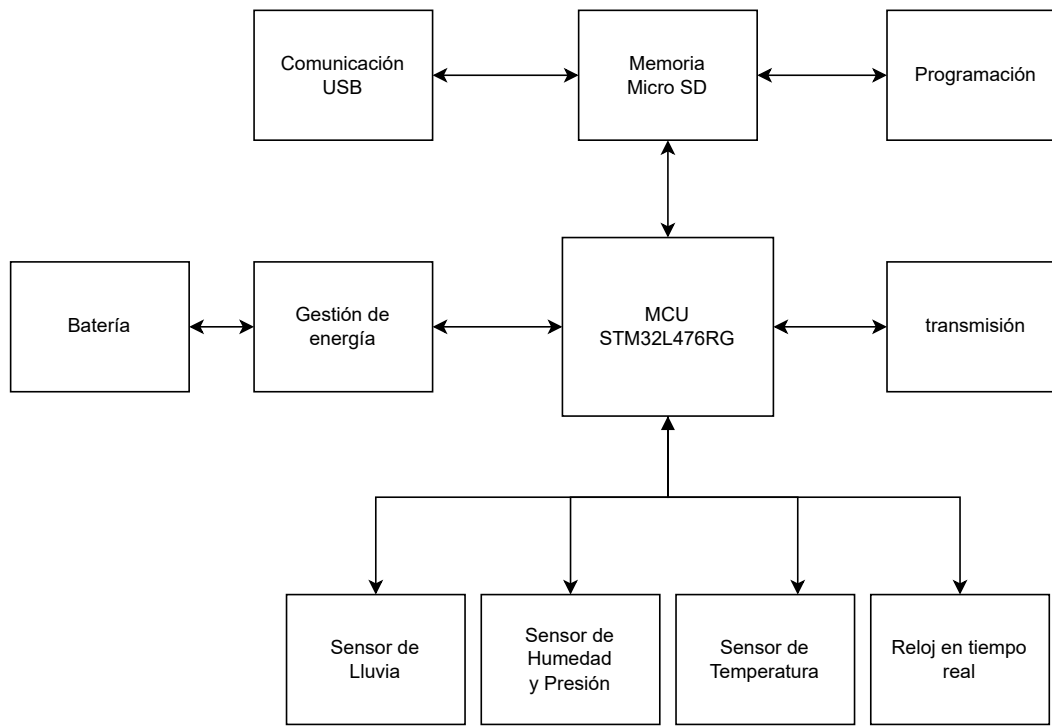


Figura 3.1: Diagrama de bloques general de la arquitectura para las cuatro variantes.

### Notas por variante

Aunque el diagrama es común, cada arquitectura concreta se diferencia por la presencia del “*harvester*” solar y por la tecnología de radio. Las variantes con LoRaWAN integran el módulo y antena de banda sub-GHz y se apoyan en un servidor local para el control de tasa de datos y sincronización opcional; las variantes con NB-IoT incorporan el módem celular y la SIM. En los modelos sin aporte solar, la política de transmisión se ajusta para priorizar la autonomía con la batería disponible, manteniendo idéntica base de sensado y almacenamiento local.

## 3.2. Herramienta de diseño electrónico

El diseño de las placas se realiza en **KiCad**, una plataforma de software libre y de código abierto que se eligió por su accesibilidad, su ecosistema de bibliotecas comunitarias y su compatibilidad con flujos de control de versiones centrados en texto. Esta elección es coherente con el objetivo general del proyecto de favorecer la reproducibilidad y la edición futura por parte de terceros sin incurrir en costos de licenciamiento, tanto para la electrónica como para el *selector* y el firmware, también desarrollados sobre herramientas abiertas.

KiCad permite mantener los esquemáticos y el “*layout*” bajo control de cambios, aplicar reglas de diseño específicas del fabricante y automatizar verificaciones eléctricas y geométricas antes de la fabricación. La disponibilidad de “*plugins*” para generación de “*build of materials*”, exportación de

archivos de fabricación y revisión de modelos 3D facilita la trazabilidad entre versiones de prototipo y el traspaso de información a fabricación y montaje. Además, su adopción extendida en la comunidad asegura la portabilidad de los proyectos y la posibilidad de incorporar contribuciones externas a lo largo de la vida del sistema.

### Librerías externas

Para la incorporación de huellas (*footprints*) y símbolos en el diseño se utilizaron, en primer lugar, las bibliotecas provistas por los propios fabricantes cuando estuvieron disponibles. En aquellos casos en que no existía una librería oficial, se recurrió a repositorios ampliamente utilizados como SnapEDA y Ultra Librarian. En todos los casos, antes de integrar un componente en el proyecto se verificó la consistencia entre la librería descargada y la documentación oficial, comparando dimensiones críticas, orientación, *land pattern* y numeración de pines con la hoja de datos correspondiente. Adicionalmente, cuando se dispuso de modelos 3D, estos se emplearon para validar la compatibilidad mecánica real (altura, interferencias y alineación), reduciendo el riesgo de errores de ensamblaje y asegurando que los componentes seleccionados fuesen físicamente montables en la PCB.

## 3.3. Variante A: LoRa + batería secundaria + aporte solar

### 3.3.1. Microcontrolador (MCU)

Se selecciona el microcontrolador **STM32L476RG** de STMicroelectronics a partir de las pruebas de prototipado y de su desempeño en modos de bajo consumo. Este dispositivo, basado en un núcleo ARM Cortex-M4F, ofrece una combinación favorable de consumo en “*Stop*”/“*Standby*”, periféricos de temporización, interfaz USB FS y ADCs de buena linealidad para tareas de sensado ocasional. La elección también considera la disponibilidad de bibliotecas y herramientas (STM32CubeIDE / HAL), lo que facilita la reutilización de firmware entre variantes y reduce el esfuerzo de mantenimiento.

En cuanto a desacoplos, como se muestra en la Fig. 3.2, se implementa la recomendación del fabricante: un condensador cerámico 100nF por cada pin VDD, localizado a escasos milímetros del encapsulado, complementado por 4,7  $\mu\text{F}$  de “*bulk*” por banco de alimentación. Para VDDA se dispone un filtro dedicado (cuenta con un “*ferrite bead*” entre VDD y VDDA) y los condensadores de 1 $\mu\text{F}$  y 10nF, con el fin de mitigar el acoplamiento de ruido al dominio analógico. Para la alimentación del bloque USB se ubican 100nF y 1 $\mu\text{F}$  cerca del pin de VDDUSB, conforme a la guía de hardware del fabricante.

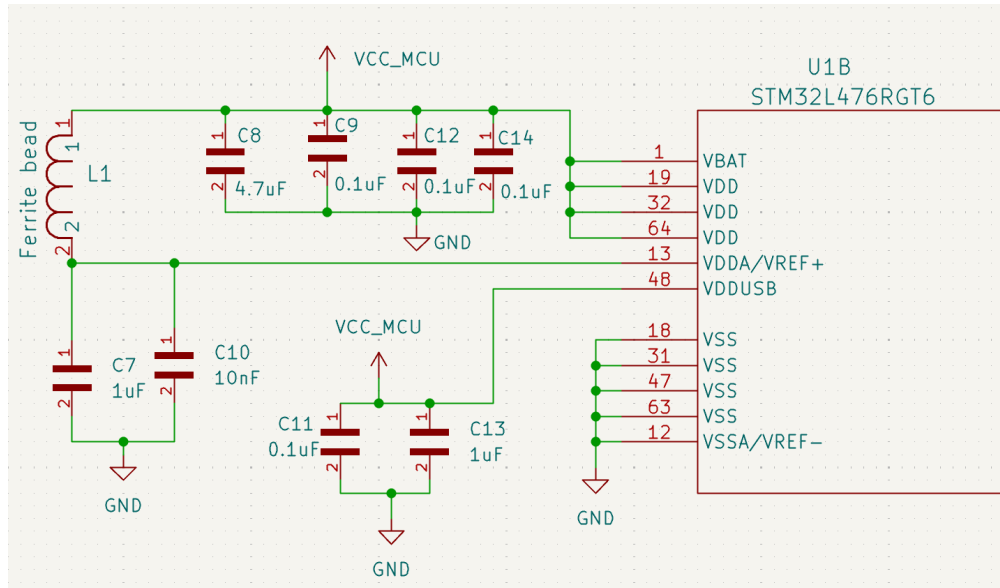


Figura 3.2: Esquemático alimentación microcontrolador en software KiCad.

### Red de relojes: cristales HSE y LSE

El diseño adopta un cristal **HSE de 8 MHz** (830106162701, Würth Elektronik) para la operación de alta velocidad y un cristal **LSE de 32,768 kHz** para la base de tiempos de bajo consumo. La selección de las capacidades de carga sigue la expresión clásica del oscilador Pierce:

$$C_L = \frac{C_1 C_2}{C_1 + C_2} + C_{\text{stray}}.$$

En esta expresión,  $C_L$  corresponde a la **capacitancia de carga especificada en la hoja de datos del cristal** (parámetro *Load capacitance*,  $C_L$ ),  $C_1$  y  $C_2$  son los condensadores externos conectados desde cada terminal del cristal a tierra, y  $C_{\text{stray}}$  representa la capacitancia parasitaria (huella, pistas y pines), estimada entre 2–6 pF. La implementación de ambas redes se ilustra en la Fig. 3.3. En la práctica se usa  $C_1 \approx C_2 = C$ , con lo cual  $C \approx 2(C_L - C_{\text{stray}})$  pF.

Para el HSE ( $C_L = 16$  pF), con  $C_{\text{stray}} \approx 5$  pF, el valor teórico es  $C \approx 2(16 - 5) = 22$  pF. Se emplean  $C_1 = C_2 = 20$  pF, lo que produce  $C_L \approx 10 + 5 = 15$  pF, es decir, un desvío de  $-1$  pF respecto del objetivo; este compromiso reduce la variedad de valores en la BOM sin afectar el arranque del oscilador. Para el LSE ( $C_L = 12,5$  pF) con la misma parasitaria, el cálculo entrega  $C \approx 2(12,5 - 5) = 15$  pF, por lo que se instalan  $C_1 = C_2 = 15$  pF, cumpliendo la carga nominal.

La topología física procura pistas simétricas, retorno inmediato y mínima longitud para reducir  $C_{\text{stray}}$  e inyección de ruido.

#### 3.3.2. Interfaz de arranque y reinicio

El reinicio del sistema se realiza mediante el pulsador táctil **EVP-BT1A4A000**, dispuesto con el desacoplo correspondiente para evitar rebotes a nivel de reinicio por acoplamientos espurios. La selección de modo de arranque se resuelve botón SPDT unipolar de doble tiro **TLSMDT3C020GLFS**, que permite cambiar estado de BOOT. La conexión se muestra en la Fig. 3.3.



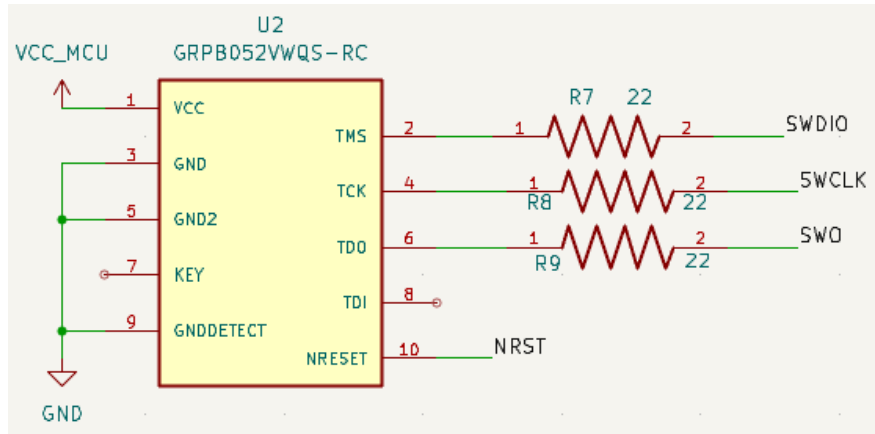


Figura 3.4: Esquemático pines de programación microcontrolador en software KiCad.

### 3.3.4. Transceptor LoRa y antena

En las variantes LoRa se utiliza el módulo certificado **LR62E** de Fanstel se muestra en la Fig. 3.5, que integra el transceptor y la red de adaptación de RF, reduciendo el esfuerzo de certificación. La antena seleccionada es la **ANT-20090EB56**, escogida por compatibilidad eléctrica y costo, y la interconexión a gabinete se realiza mediante el cable **CAB602RF-0200-A-1**.

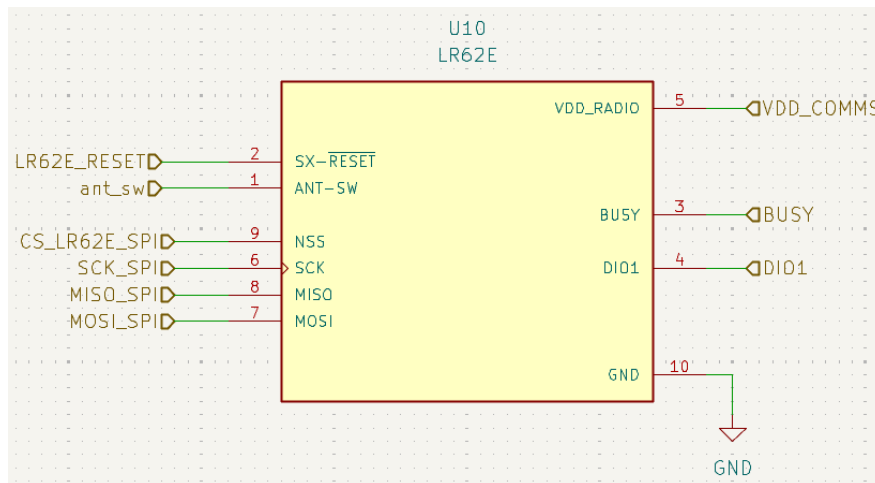


Figura 3.5: Esquemático modulo LoRa en software KiCad.

### 3.3.5. Reloj en tiempo real (RTC) externo

Para garantizar etiquetado temporal preciso y despertar del MCU desde el modo de muy bajo consumo se integra el **RV-3028-C7** (32,768kHz, 1ppm). El circuito se alimenta desde una batería (CR1220) de respaldo en portapilas **3000TR** como se observa en la Fig. 3.6, con desacoplos dedicados tanto en la rama de batería como en la rama desde la fuente principal de sensores. Se dispone además de un pulsador de servicio que permite limpiar la memoria del RTC cuando es necesario. Esta solución minimiza el error acumulado del reloj aun en campañas largas y reduce el tiempo activo del MCU para el sellado de marcas de tiempo.

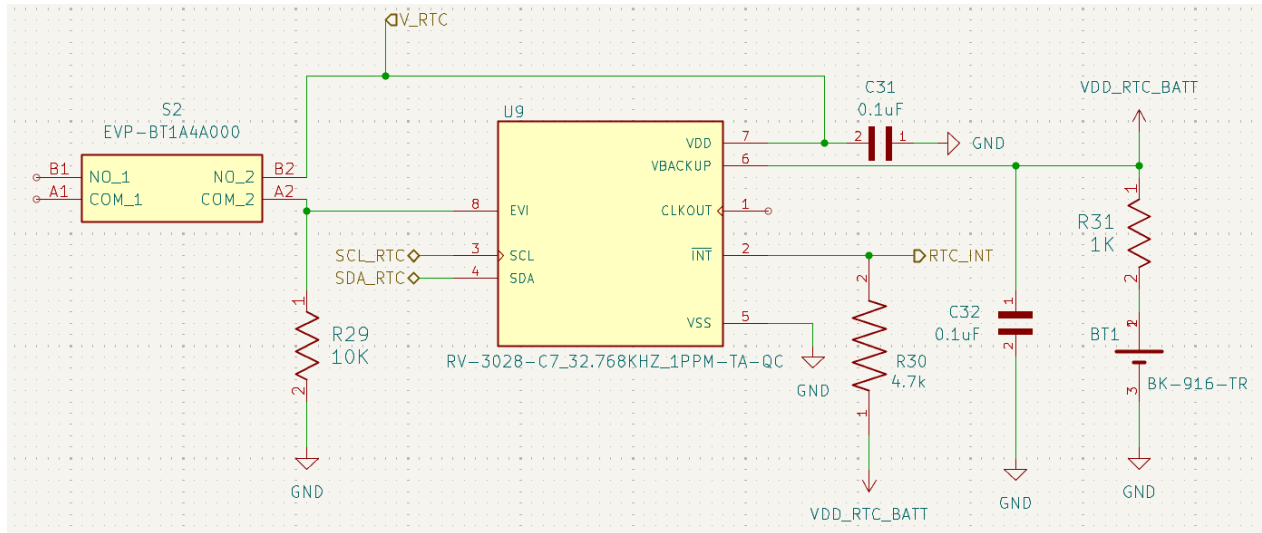


Figura 3.6: Esquemático RTC en software KiCad.

### 3.3.6. Alimentación: “*harvesting*”, conversión y conmutación

### 3.3.7. Cargador y Cosechadora

Se implementa el **BQ25570** de Texas Instruments por su capacidad de “*cold start*” y MPPT integrado a muy baja potencia. La selección de resistencias sigue las ecuaciones del fabricante en función de la curva I-V del panel y del perfil de batería Li-ion/Li-Po. La Tabla 3.1 resume los valores adoptados en el prototipo de referencia.

Tabla 3.1: Red de resistencias para BQ25570.

ROV1	5.62 M $\Omega$
ROV2	7.32 M $\Omega$
ROK1	4.99 M $\Omega$
ROK2	6.65 M $\Omega$
ROK3	1.24 M $\Omega$
ROUT1	2.10 M $\Omega$
ROUT2	0.90 M $\Omega$

Para el resto de los componentes pasivos, inductores y condensadores asociados a conversión y filtrado se adoptan los valores *referenciales* indicados en las hojas de datos. Esta decisión busca garantizar la estabilidad de lazo, el cumplimiento de *ripple* objetivo y el arranque confiable sin introducir iteraciones innecesarias en prototipado. Esta topología se aprecia en la Fig. 3.7.

La conexión física de la batería al sistema se realiza mediante el conector de dos posiciones **S2B-PH-K-S** (serie JST PH) montado en la PCB.

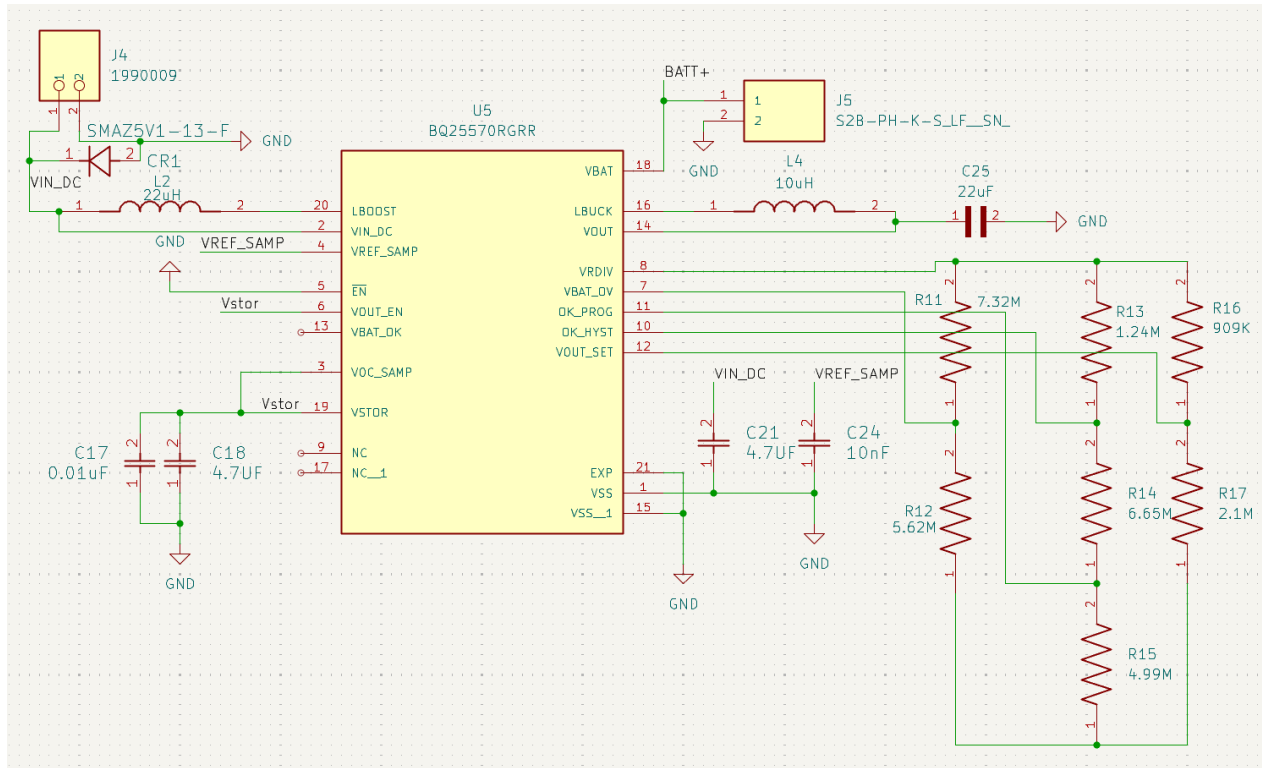


Figura 3.7: Esquemático BQ25570 en software KiCad.

### 3.3.8. Convertidor principal.

La alimentación del sistema desde batería se regula con el convertidor reductor **TPS62840-DLCR**, elegido por su configuración sencilla del voltaje de salida (una resistencia externa) como se muestra en la Fig. 3.8, su excelente rendimiento a cargas ligeras y una corriente de reposo típica del orden de 60nA, lo que contribuye de manera decisiva al presupuesto energético en *sleep*. El circuito requiere la resistencia de programación de salida, un condensador de entrada, el inductor correspondiente y el condensador de salida para almacenar y filtrar la energía con estabilidad.

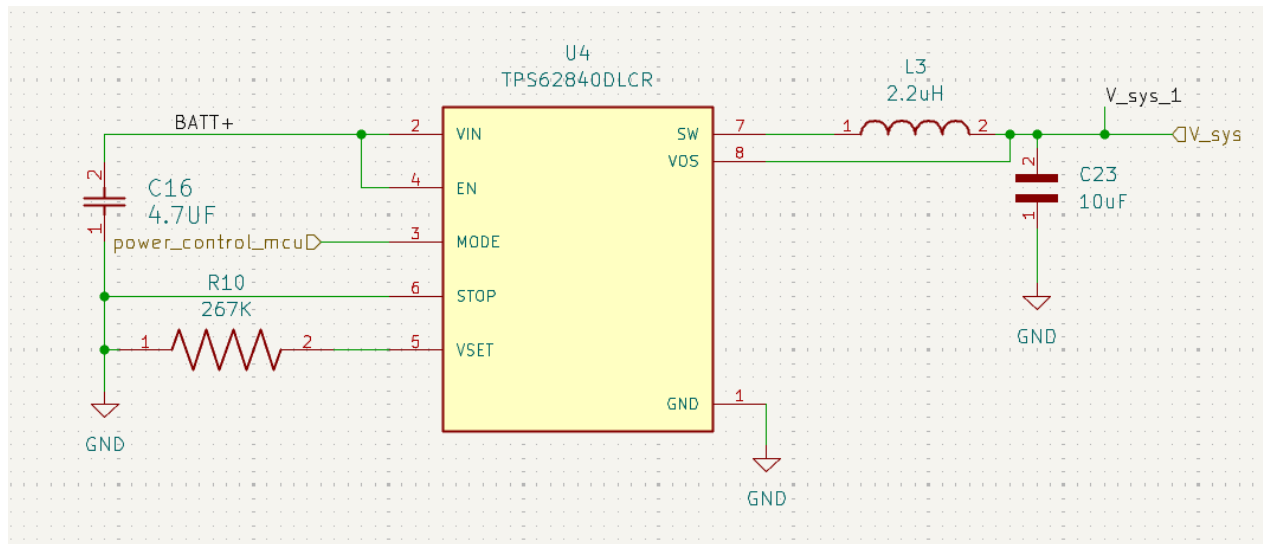


Figura 3.8: Esquemático TPS62840DLCR en software KiCad.

### 3.3.9. Conmutación de cargas.

Para desconectar selectivamente los dominios de sensores y de comunicación se emplean dos **TPS22916CYFPT**. Se optó por un *load switch* integrado en lugar de un MOSFET discreto por las siguientes razones, el TPS22916 presenta una corriente de fuga en estado apagado de 10nA típico, valor que minimiza el consumo durante los periodos de reposo del nodo; incorpora bloqueo de corriente inversa y control de corriente de irrupción (*inrush*) de forma integrada, prescindiendo de componentes de protección externos que serían necesarios con un MOSFET discreto; e incluye descarga rápida de salida (*quick output discharge*), asegurando la desconexión efectiva de las cargas. Además, su encapsulado DSBGA de 4 terminales reduce significativamente el área ocupada en la PCB respecto a una solución discreta equivalente. Esta estrategia permite apagar completamente cargas que no se utilizan entre mediciones, reduciendo los consumos residuales. De acuerdo con la hoja de datos, se instalan los condensadores de "load" y desacoplos a la salida, para el módulo de comunicación se colocan 2x22µF, mientras que para el dominio de sensores/almacenamiento se usa 22µF como "bulk", además de los 100nF locales en cada IC. La implementación de ambos ICs se ilustra en la Fig. 3.9.

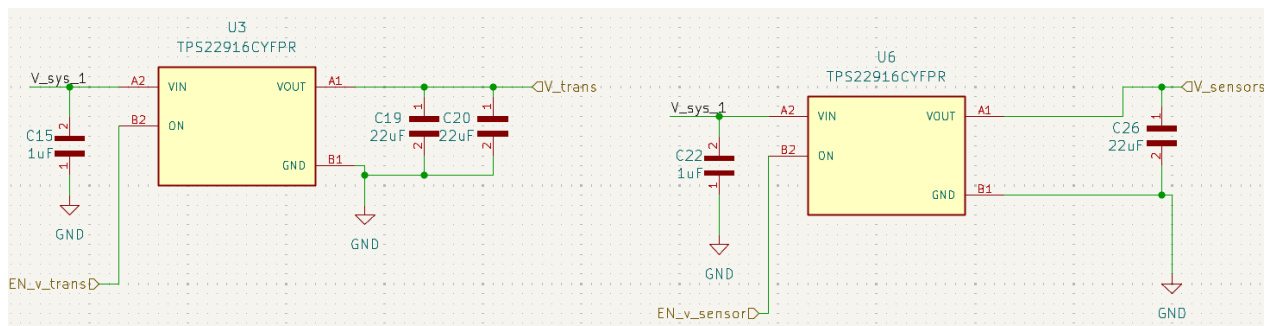


Figura 3.9: Esquemático TPS22916CYFPT para sensores y transmisión en software KiCad.

### 3.3.10. Almacenamiento local

El diseño contempla almacenamiento local no volátil para garantizar la continuidad del registro aun ante pérdida de enlace. Según la variante y el escenario de uso, se puede poblar un “*socket microSD*” **5033981892** (Molex) complementado por los elementos de protección y desacoplo requeridos como se observa en la Fig. 3.10. En el bus SPI y en los conectores de alimentación se incluyen resistencias de protección de corriente; además, se reserva huella para *pull-ups* opcionales, habilitados sólo si el “*layout*” o la longitud de cableado lo justifican.

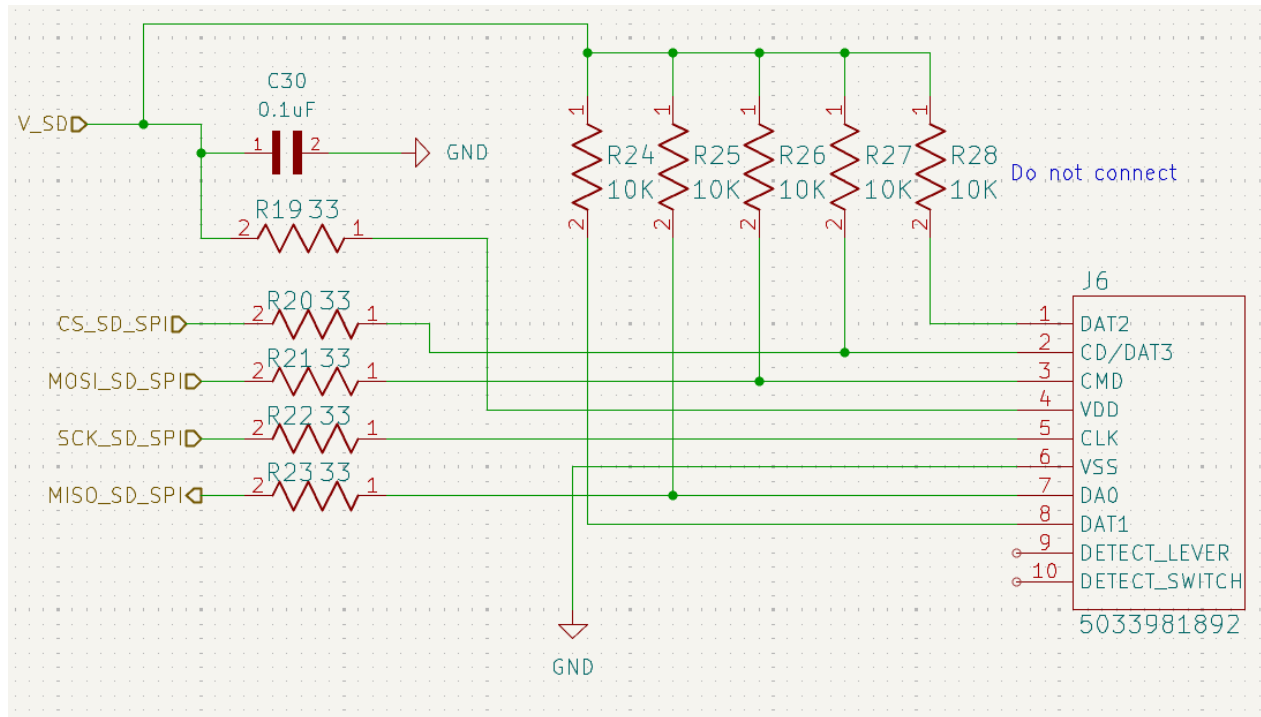


Figura 3.10: Esquemático socket microSD en software KiCad.

### 3.3.11. Sensores ambientales

Para humedad relativa y presión atmosférica se emplea el **BME280**, que se observa en la Fig. 3.11, un dispositivo ampliamente validado en proyectos previos y con buen equilibrio entre consumo y precisión. La comunicación se realiza por **I<sup>2</sup>C** y el circuito se complementa con los desacoplos de 100nF próximos al encapsulado. La temperatura del aire se mide con un **TMP117NAIYBGR**, como se observa en la Fig. 3.12, que reemplaza al TMP102 empleado en los prototipos de la etapa preliminar de desarrollo, aportando mayor precisión ( $\pm 0,1,^{\circ}\text{C}$  frente a  $\pm 0,5,^{\circ}\text{C}$ ). Este sensor también opera por **I<sup>2</sup>C** y requiere un condensador de desacoplo y un “*pull-up*” en la línea de alarma si se utiliza esa función. La precipitación se detecta mediante un **pluviómetro de balancín**, cuya salida por contacto se acondiciona para reducir consumo y se cuenta mediante entradas del MCU.

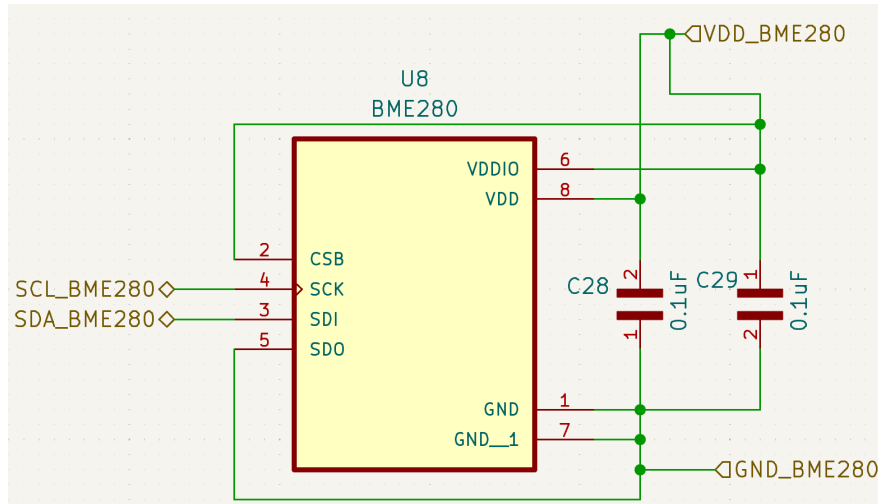


Figura 3.11: Esquemático BME280 en software KiCad.

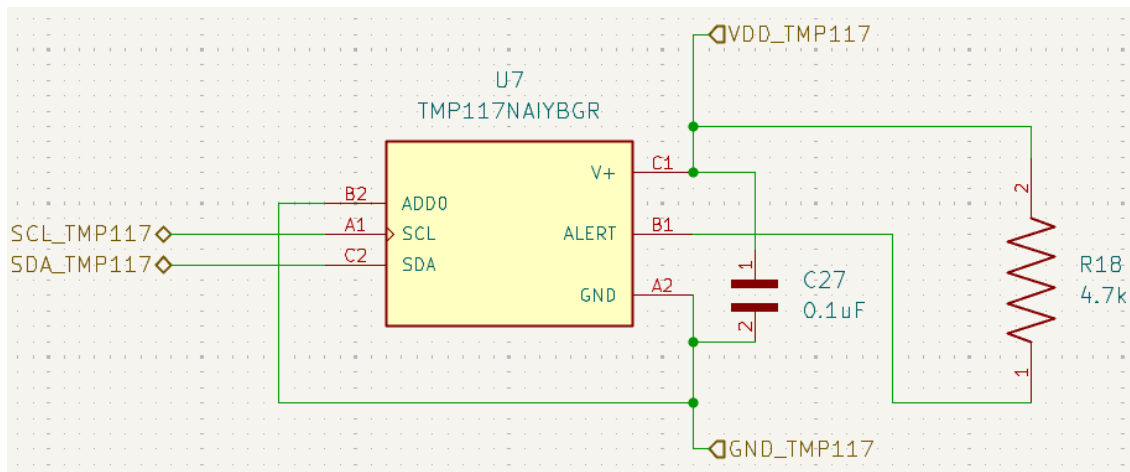


Figura 3.12: Esquemático TMP117 en software KiCad.

### 3.3.12. Conectividad de sensores y periféricos

La interconexión con la placa de sensores se implementa mediante conectores **B4B-XH-A**, mientras que el mazo de cables (*wire harness*) se ensambla con carcasas **XHP-4** y terminales **SXH-001T-P0.6**. Las interfaces externas del **panel solar** y del **pluviómetro** se resuelven con conectores **Phoenix Contact 1990009**. Adicionalmente, se exponen los pines no utilizados del microcontrolador a través del cabezal **10129381-924003BLF** (Amphenol) de 24 posiciones, junto con los rieles de tensión provenientes de los conmutadores de potencia, la salida del convertidor principal y un bus **I<sup>2</sup>C** para expansión y depuración. La asignación de señales y rieles puede verse en la Fig. 3.13.

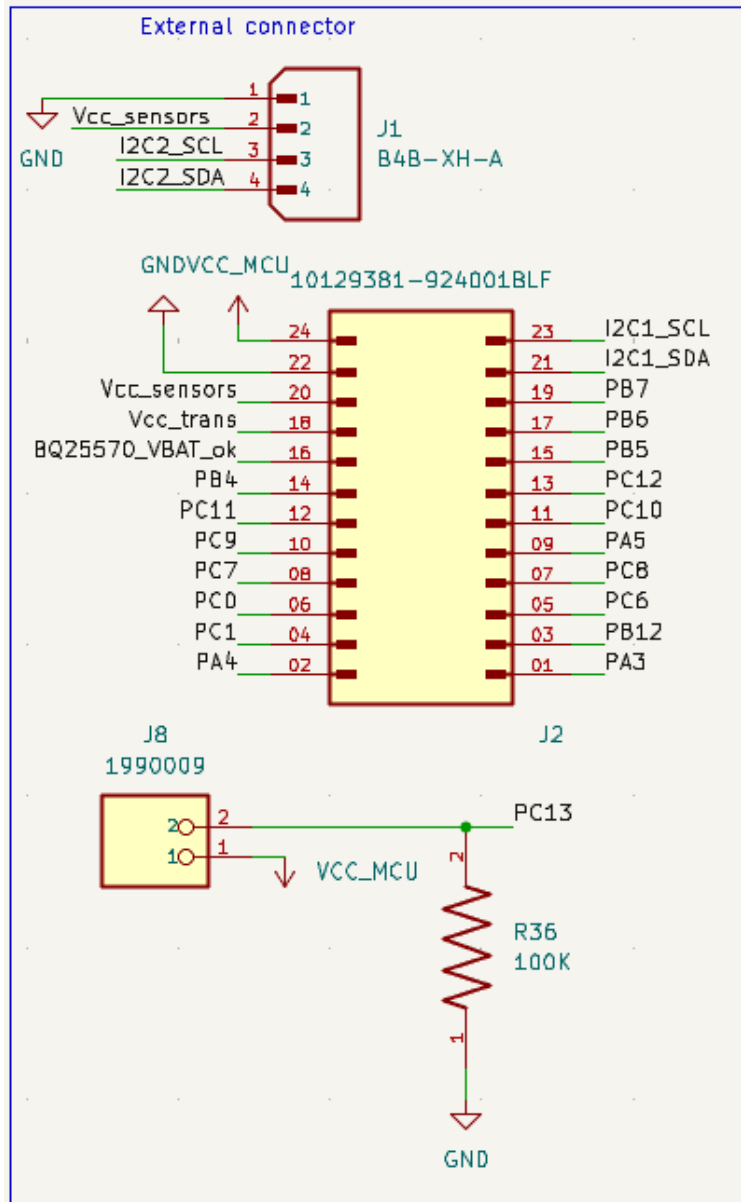


Figura 3.13: Esquemático de conectores externos en KiCad.

### 3.3.13. Interfaz USB-C y protección

Para la configuración y descarga de datos se incorpora un conector **USB tipo C GSB1C41110-SSHR**. Dado que el dispositivo opera como USB2.0, se implementan las resistencias de **5,1kΩ** en los pines CC para anunciarse correctamente al “host”. Las líneas de datos D+/D- se protegen con el supresor **USBLC6-2SC6** de baja capacitancia, mientras que la línea VBUS cuenta con TVS y filtrado mediante “ferrite bead” y condensadores de desacoplo cercanos. Además, se implementa un divisor resistivo a VBUS para que el MCU detecte de forma segura la presencia de alimentación por USB y pueda entrar en modo de servicio cuando corresponda. El circuito de adaptación y protección puede verse en la Fig. 3.14.

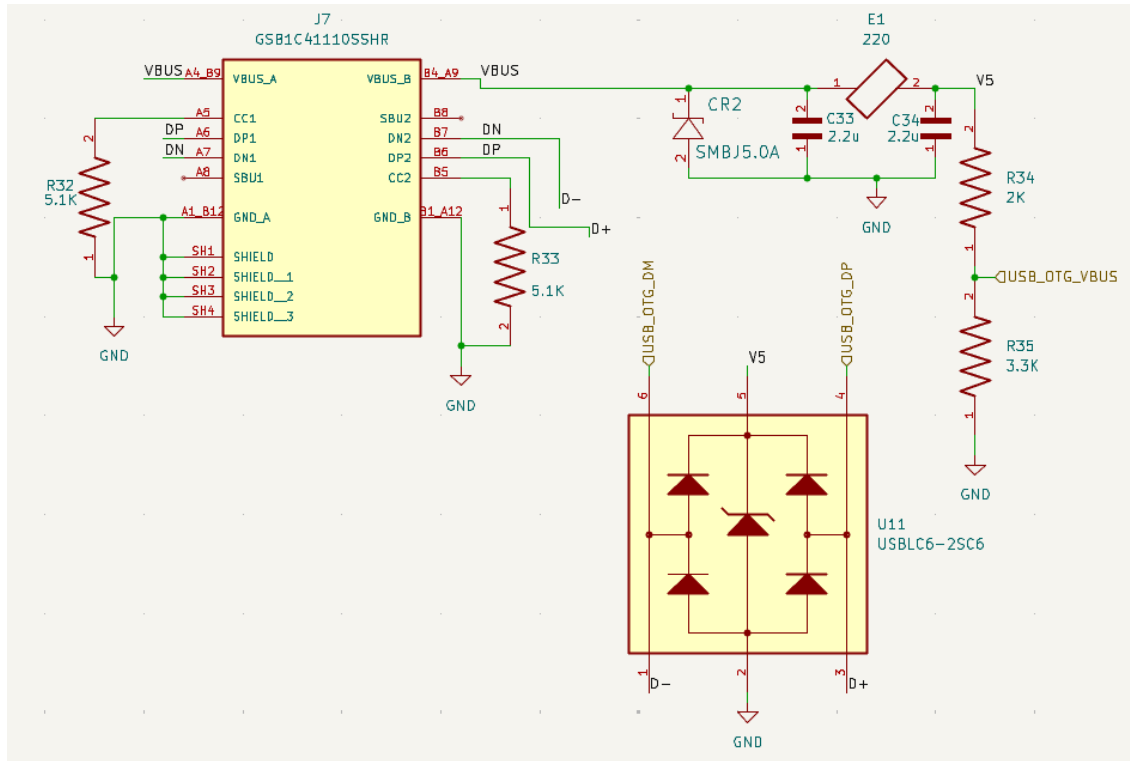


Figura 3.14: Esquemático conector USB y protección en software KiCad.

### 3.3.14. Protección ambiental y montaje

La electrónica se integra en una carcasa **PNB-30417-CKO** con grado de protección **IP67**, suficiente para exposición a la intemperie. Para el paso de cables hacia sensores y panel se instalan tres prensaestopas **1SNG601011R0000** con **IP68**. A fin de mejorar la representatividad de las mediciones de aire, el sensor de T/HR se monta dentro de un **escudo de radiación** impreso en resina y protegido con recubrimiento UV. Se diseña además una **montura** específica para el panel solar, impresa a medida, que asegura orientación y rigidez suficientes frente a viento y vibraciones.

## 3.4. Variante B: LoRa + batería primaria

En esta variante se conserva el convertidor reductor principal del riel del sistema y se refuerza la reserva de energía en la entrada con un condensador **220  $\mu$ F F930J227MBA**, cuyo *“bulk”* y ESR ayudan a amortiguar transitorios de corriente durante eventos de medición y transmisión. La batería de tipo **D LiSOCl<sub>2</sub>** se conecta al frente de potencia mediante el conector **Phoenix Contact 1990009**. La ruta de potencia y el desacoplo asociados a la entrada del convertidor se muestran en la Fig. 3.15.

Dado que en esta configuración se elimina el *energy harvester* y, con ello, la telemetría de estado de batería que éste proporcionaba, se incorpora un comparador **MAX9065EUK+T** como monitor de umbral. El comparador se alimenta a través de un **TPS22916CYFPT** gobernado por el MCU, de modo que el monitoreo sólo permanezca activo cuando se requiere, evitando consumo en reposo innecesario. La entrada IN del MAX9065 se conecta directamente al nodo de batería para evaluar su nivel bajo demanda, mientras que su salida se lee por una entrada digital del microcontrolador

para registrar eventos de bajo voltaje o tomar decisiones de reducción de carga. La integración del comparador, su control por “power switch”. El resto del circuito permanece inalterado respecto de la arquitectura base, incluyendo sensado, almacenamiento local y enlace LoRa.

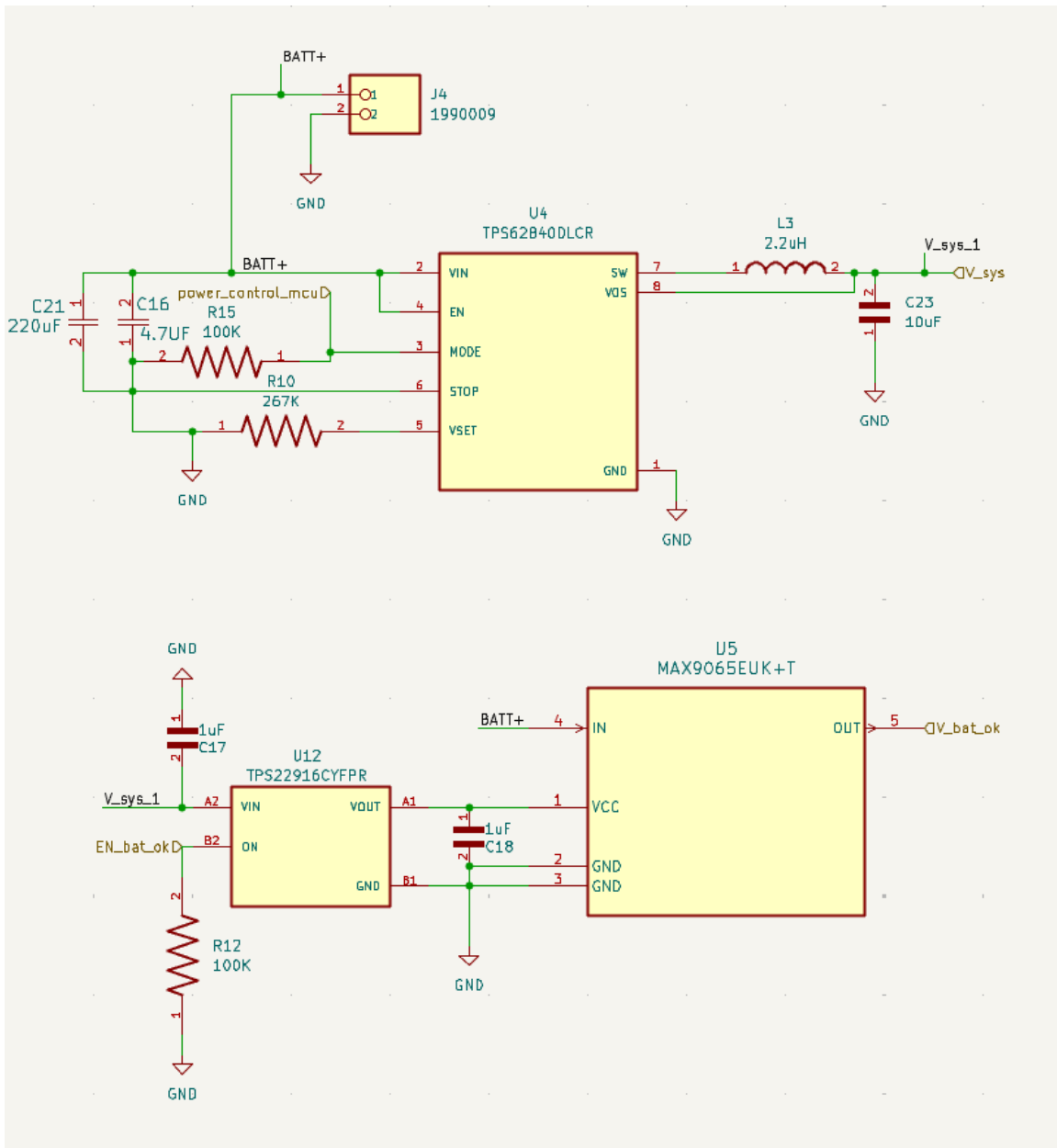


Figura 3.15: Esquemático lectura de estado de batería y alimentación en software KiCad.

### 3.5. Variante C: NB-IoT + batería secundaria + aporte solar

En esta variante se mantiene el subsistema de alimentación de la configuración LoRa con aporte solar, incluyendo el *harvester* y la filosofía de conmutación de dominios para minimizar el consumo en reposo. La principal modificación respecto de la arquitectura LoRa consiste en reemplazar el

bloque de comunicación por un módem **NB-IoT BC660KGLAA**, con el objetivo de habilitar telemetría sobre red celular en escenarios donde no se dispone de infraestructura LoRaWAN local. En consecuencia, el diseño integra además el portasim (*SIM holder*) y las protecciones asociadas para asegurar operación estable durante eventos de registro en red y ráfagas de transmisión.

El módem **BC660KGLAA** se implementa siguiendo la referencia de diseño recomendada por el fabricante para aplicaciones de bajo consumo. La interfaz de RF se expone mediante un conector **U.FL-R-SMT-1(80)** para antena externa, incorporando una resistencia en serie y dejando huellas para dos condensadores de ajuste, de modo de permitir adaptación fina de la antena si resulta necesaria durante la caracterización. La comunicación entre el microcontrolador y el módem se realiza por UART.

El control de *reset* del módem se implementa mediante un **2N7002** como etapa de conmutación, en conjunto con una red RC (condensador y resistencia) para asegurar niveles y tiempos compatibles con la secuencia de reinicio requerida.

Para la alimentación del BC660 se dispone una red de desacoplos distribuida, diseñada para absorber picos de corriente y reducir el ruido conducido hacia el riel del sistema. En particular, se implementa un banco de **100  $\mu\text{F}$**  (**TLJA107M006R0500**) complementado por desacoplos locales de alta frecuencia (**0,1  $\mu\text{F}$**  **CC0603JRX7R8BB104** y **100 pF** **0603ZA101FAT2A**), además de capacitores adicionales de menor valor para adecuar la impedancia de la red de suministro en el rango de interés del módem. Como protección ante transientes en la línea de alimentación se incorpora un supresor **PTVS4V8Z1UPCAZ**, mitigando sobretensiones breves asociadas a conmutaciones o acoplamientos externos. La integración del bloque de potencia y la red de desacoplos se representa en la Fig. 3.16.

El sistema incorpora el *SIM holder* **DT1042-04SO-7**, junto con las conexiones requeridas hacia el módem. Para mejorar la robustez frente a condiciones de conexión y descargas electrostáticas, se incluyen elementos de acondicionamiento en las señales del bus SIM, en particular limitadores de corriente para las líneas. Se agregan además condensadores de **33 pF** asociados a las líneas de datos según referencia de diseño, y un condensador de desacoplo de **0,1  $\mu\text{F}$**  próximo al conector para estabilizar el dominio local del SIM. En conjunto, estas medidas reducen la probabilidad de fallas intermitentes de lectura de SIM y mejoran la estabilidad durante *attach* y transmisión.

Finalmente, el conector **009162006301150** se incorpora como parte del soporte de la interfaz SIM según la solución seleccionada para el montaje y guiado mecánico. La Figura 3.17 resume el esquemático del bloque SIM.

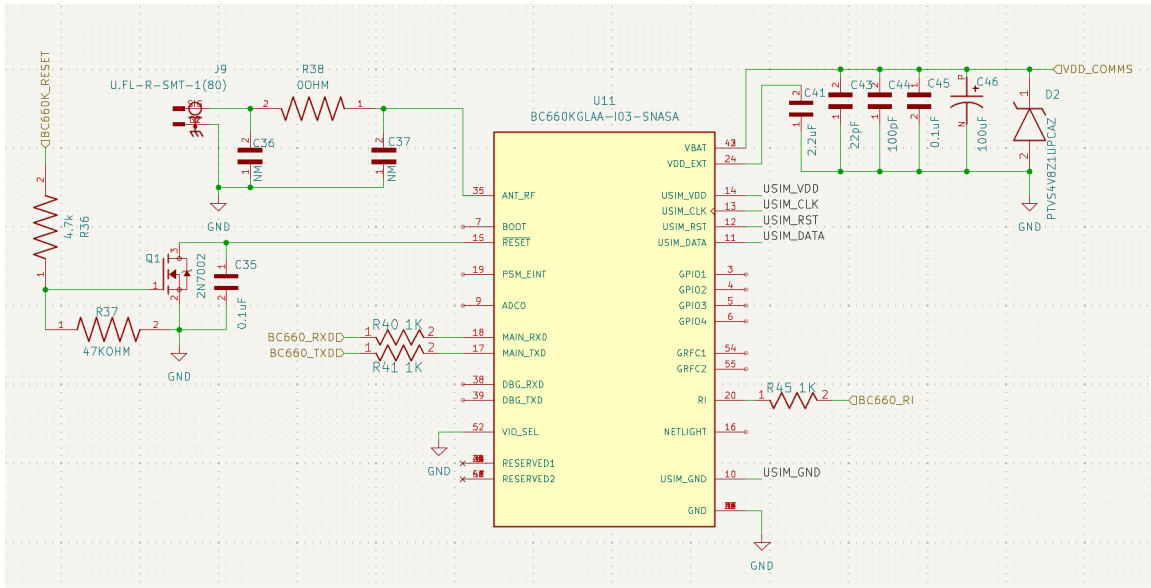


Figura 3.16: Esquemático del bloque NB-IoT en KiCad: BC660KGLAA, conexión RF por U.FL, control de reset con 2N7002, red de alimentación y desacoplos del módem.

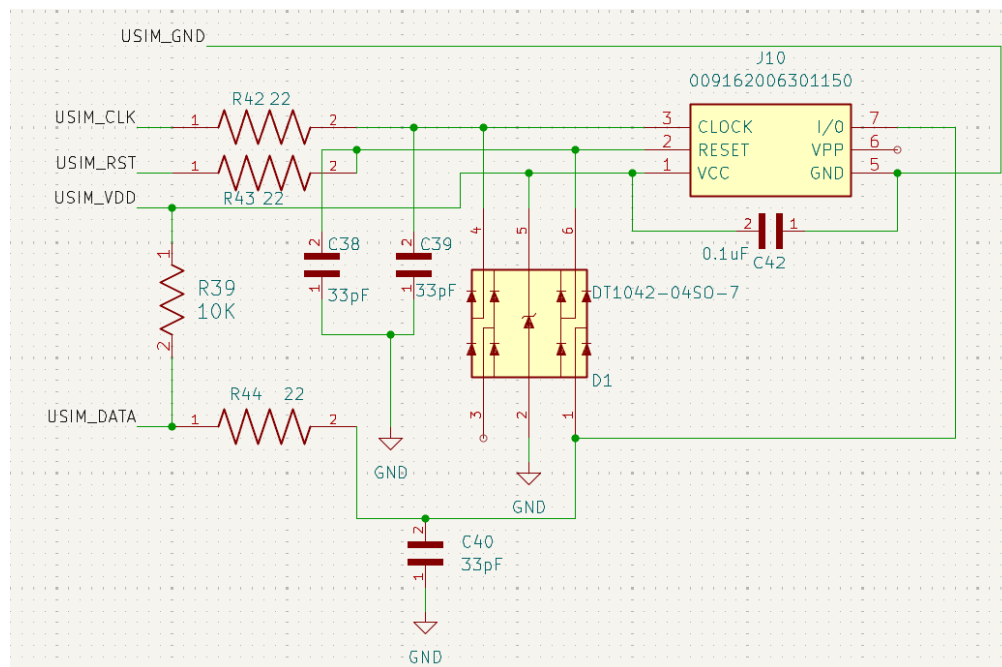


Figura 3.17: Esquemático *SIM holder* DT1042-04SO-7 con acondicionamiento de líneas.

### 3.6. Variante D: NB-IoT + batería primaria

Esta variante combina el subsistema de alimentación de la placa LoRa sin aporte solar con el bloque de comunicación celular NB-IoT. En particular, se adopta la ruta de potencia basada en batería como única fuente de energía, manteniendo el convertidor principal y la estrategia de

desacoplo utilizada en la configuración LoRa a batería, junto con las consideraciones de conexión y robustez asociadas a la batería seleccionada. Al eliminar el *harvester* solar, la arquitectura se simplifica y la autonomía queda determinada por la capacidad disponible de la batería y por el perfil de operación definido para muestreo y transmisión.

En el dominio de comunicaciones, esta variante incorpora el módem NB-IoT descrito en la sección correspondiente, conservando su implementación de bajo consumo, la interfaz UART con el microcontrolador y el subsistema de SIM. De este modo, el sistema mantiene la misma base de sensado, almacenamiento local y temporización, pero reemplaza la telemetría LoRaWAN por conectividad celular, lo que resulta especialmente útil en escenarios donde la infraestructura LoRa no está disponible o se requiere un enlace IP directo hacia el backend.

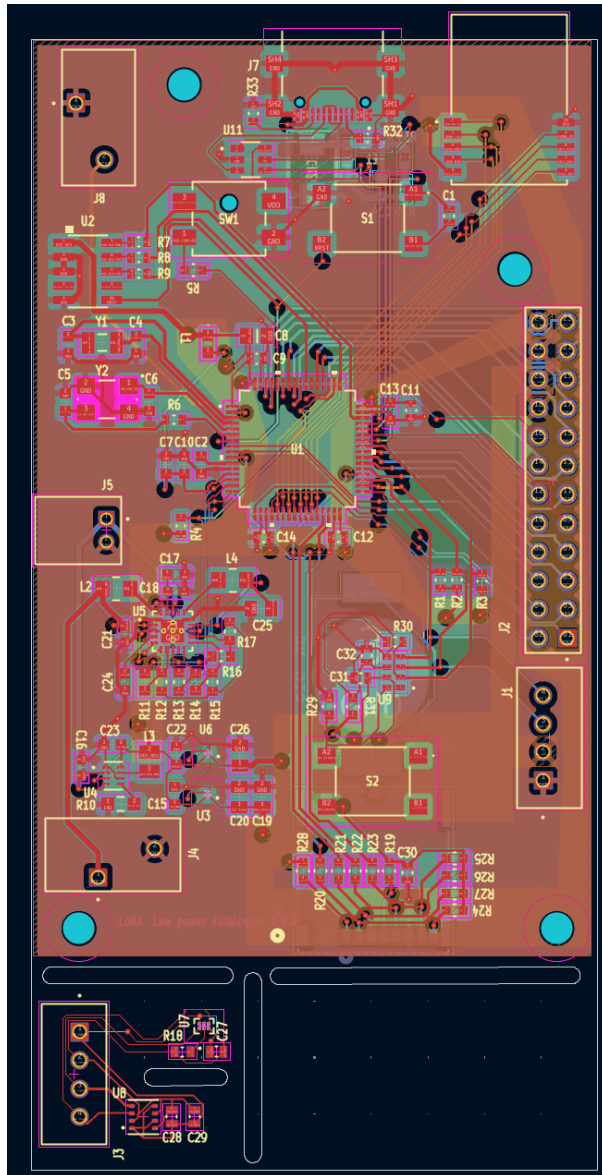
### 3.7. PCB LoRa + solar

El desarrollo de la placa LoRa con aporte solar se realizó a partir de las recomendaciones de diseño incluidas en hojas de datos y guías de aplicación de los integrados principales. En particular, se siguieron guías para el subsistema de recolección y carga (criterios de dimensionamiento y conexión del *harvester*) [15], y para el microcontrolador en lo relativo a desarrollo de hardware y operación de bajo consumo [16], [17], [18], incluyendo consideraciones específicas de diseño de osciladores [19]. Adicionalmente, se consideraron guías para el conexionado de interfaces críticas como almacenamiento en microSD vía SPI [20] y USB [21], así como recomendaciones de PCB para sensores en montaje superficial cuando aplica [22]. Estas referencias se listan en detalle en el Apéndice C

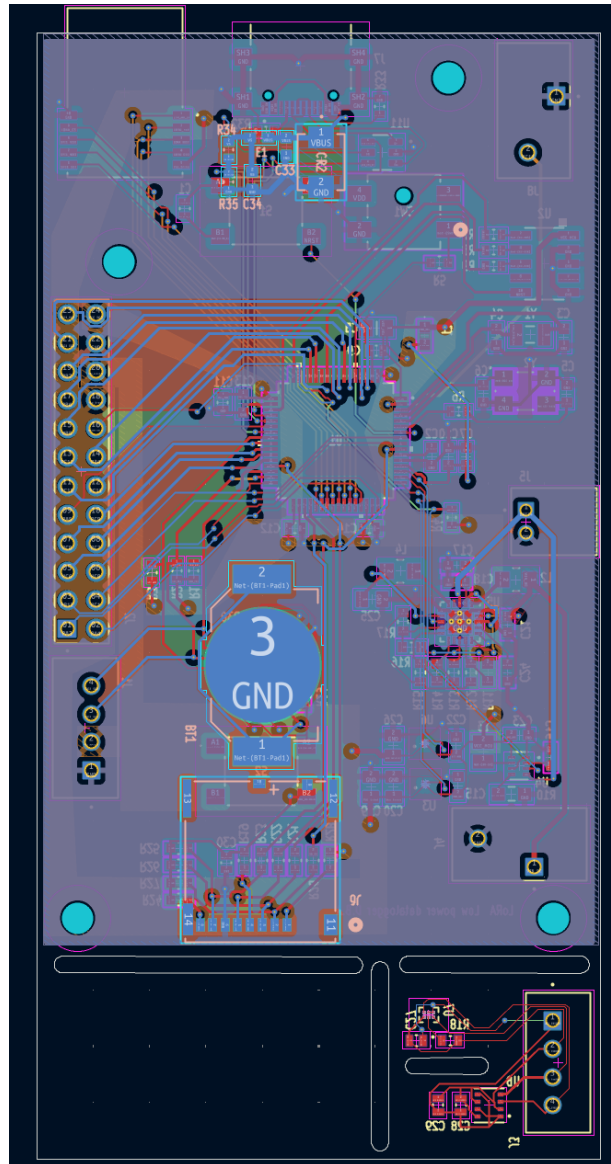
Las dimensiones de la PCB se definieron en función de la caja seleccionada **PNB-30417-KO**, con el propósito de evitar costos adicionales asociados a cajas con monturas o mecanizados especializados. Para asegurar una integración mecánica reproducible, se diseñó e imprimió una base de montaje que fija la tarjeta al gabinete mediante perforaciones **M2.5**; el detalle constructivo y la disposición de fijaciones se describen en la sección de integración mecánica correspondiente.

Junto con la placa principal se incorpora una sub-placa tipo *breakaway* destinada a los sensores que deben ubicarse fuera de la caja. Esta decisión permite separar físicamente elementos expuestos al ambiente, del volumen interno donde se concentra la electrónica de potencia y comunicaciones, manteniendo a la vez un arnés corto y controlado entre ambos dominios. La sub-placa se fabrica en el mismo panel de PCB y puede desprenderse de manera limpia una vez finalizado el ensamblaje.

Con el objetivo de reducir el área ocupada por la tarjeta sin comprometer la accesibilidad de los conectores externos, se reubicaron en la cara inferior de la PCB el *microSD holder*, el portapila del respaldo del RTC y la protección asociada a la interfaz USB. Esta redistribución libera superficie en la cara superior para el bloque de potencia, el módulo de comunicación y los conectores de campo, al mismo tiempo que mantiene trayectorias cortas y ordenadas para las señales críticas. Finalmente, se presentan los archivos del PCB y las vistas 3D del ensamble como parte del respaldo del diseño. La Fig. 3.18 presenta las vistas superior e inferior de la PCB, mientras que la Fig. 3.19 muestra el modelo 3D del ensamble para referencia mecánica.

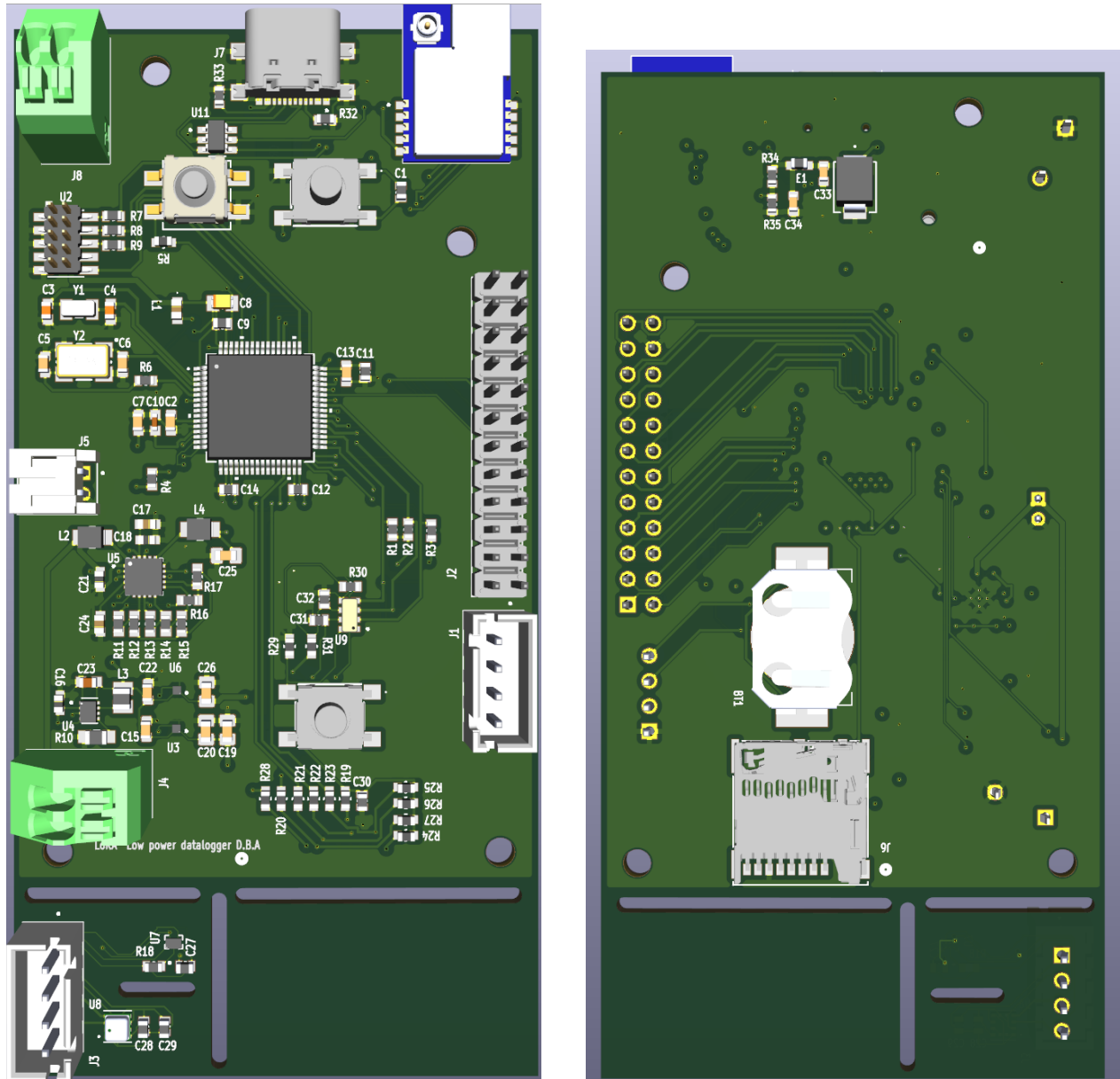


(a) Vista frontal.



(b) Vista trasera.

Figura 3.18: PCB LoRa + Solar.



(a) Vista frontal.

(b) Vista trasera.

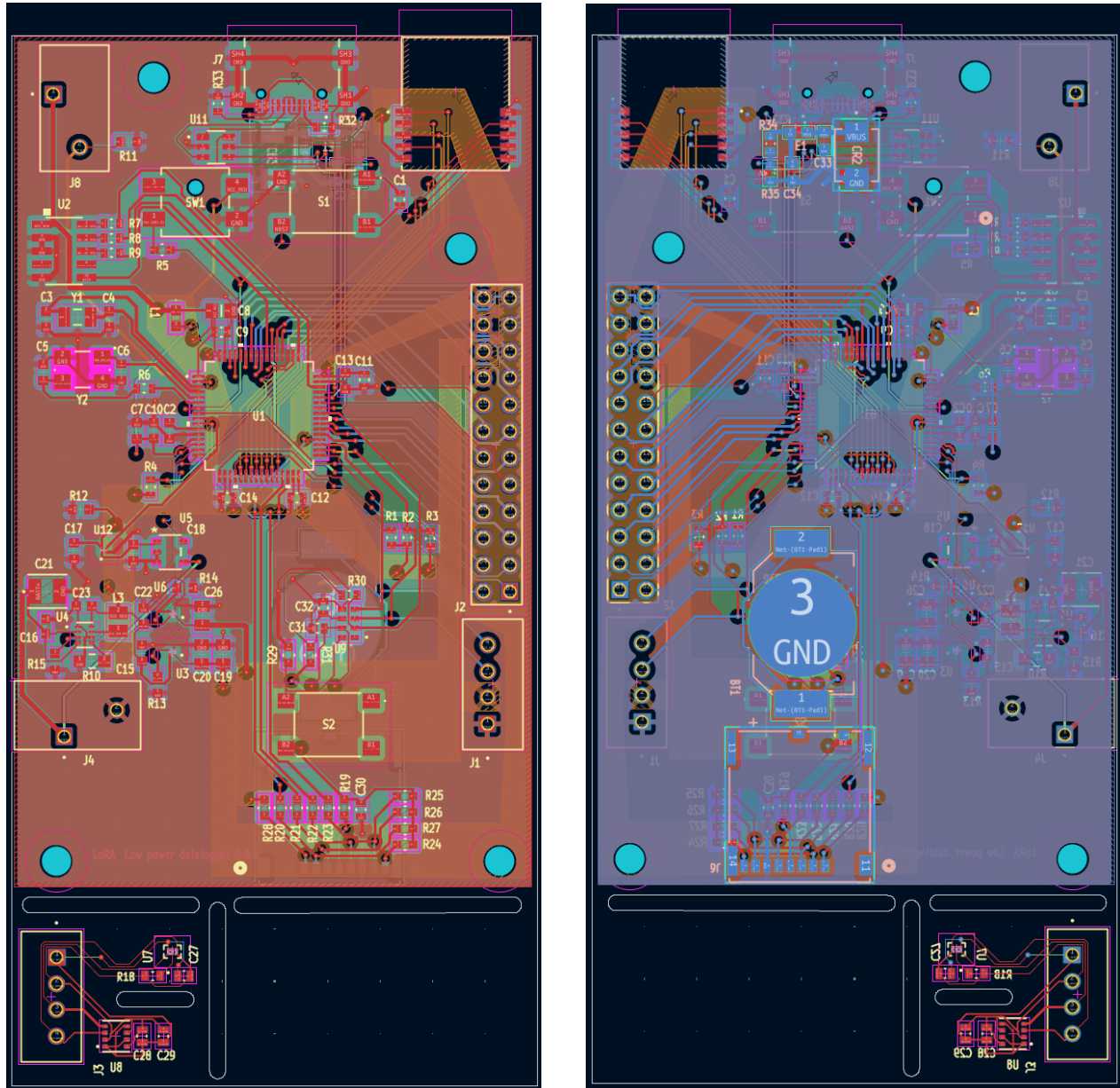
Figura 3.19: Modelo 3D PCB LoRa + Solar.

### 3.8. PCB LoRa

El diseño PCB de la variante LoRa sin aporte solar mantiene la misma definición mecánica adoptada en la placa LoRa + solar. En particular, se conservan las dimensiones y el patrón de fijación asociados al *enclosure*, así como la base de montaje impresa y la sub-placa *breakaway* destinada a los sensores ubicados fuera del gabinete. De este modo, la integración mecánica y el arnés de interconexión permanecen invariantes, facilitando el ensamblaje y la reutilización de accesorios entre variantes.

La diferencia principal respecto de la versión con aporte solar se concentra en el subsistema de

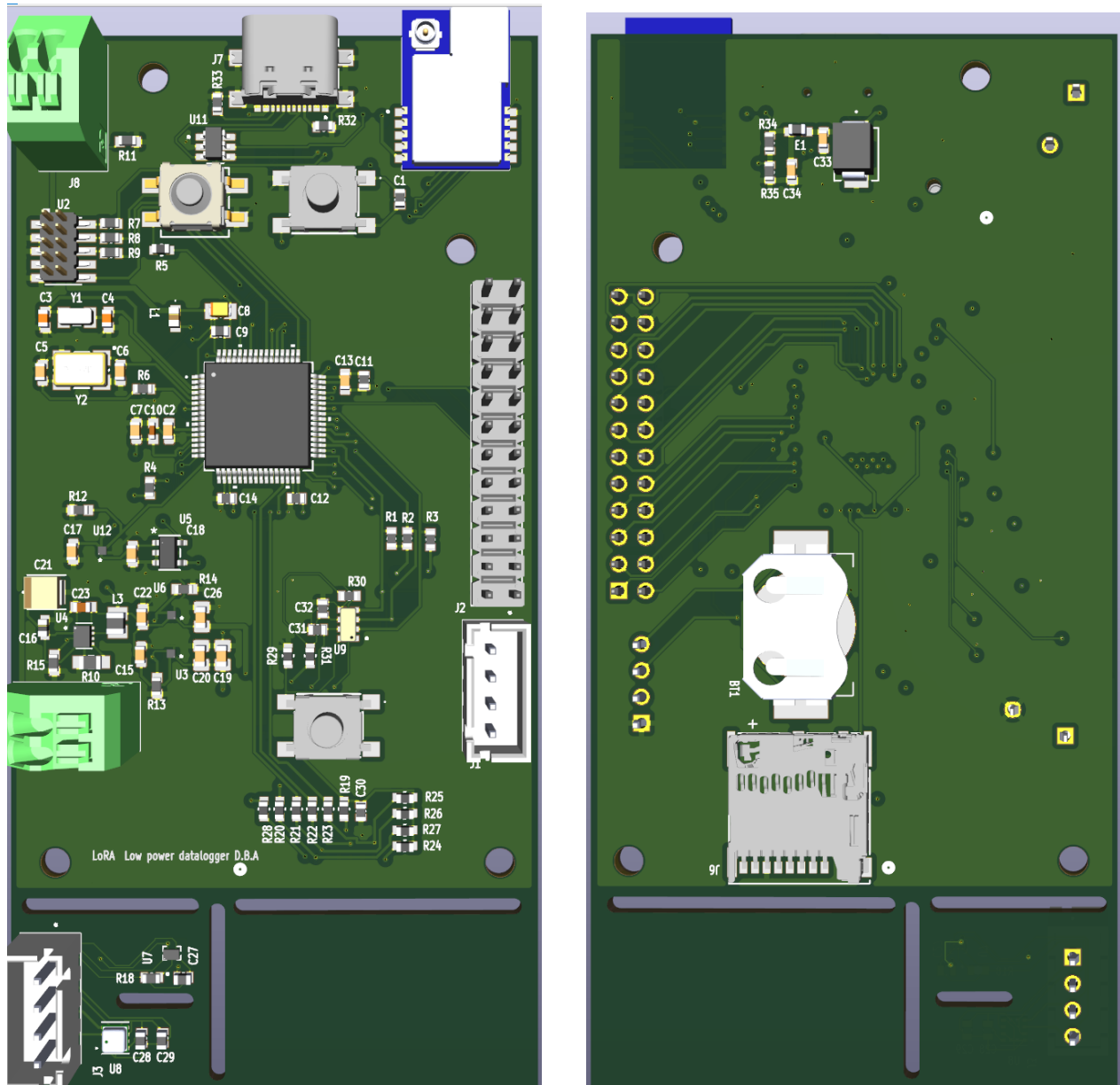
alimentación, donde se reemplazan los componentes específicos del *energy harvesting* por la configuración basada exclusivamente en batería. El resto del ruteo funcional y la ubicación relativa de bloques (microcontrolador, almacenamiento, RTC, comunicación y conectores externos) se mantiene sin cambios sustantivos, asegurando coherencia de diseño y compatibilidad con el firmware común. La Fig. 3.20 presenta las vistas superior e inferior de la PCB, mientras que la Fig. 3.21 muestra el modelo 3D del ensamble para referencia mecánica.



(a) Vista frontal.

(b) Vista trasera.

Figura 3.20: PCB LoRa.



(a) Vista frontal.

(b) Vista trasera.

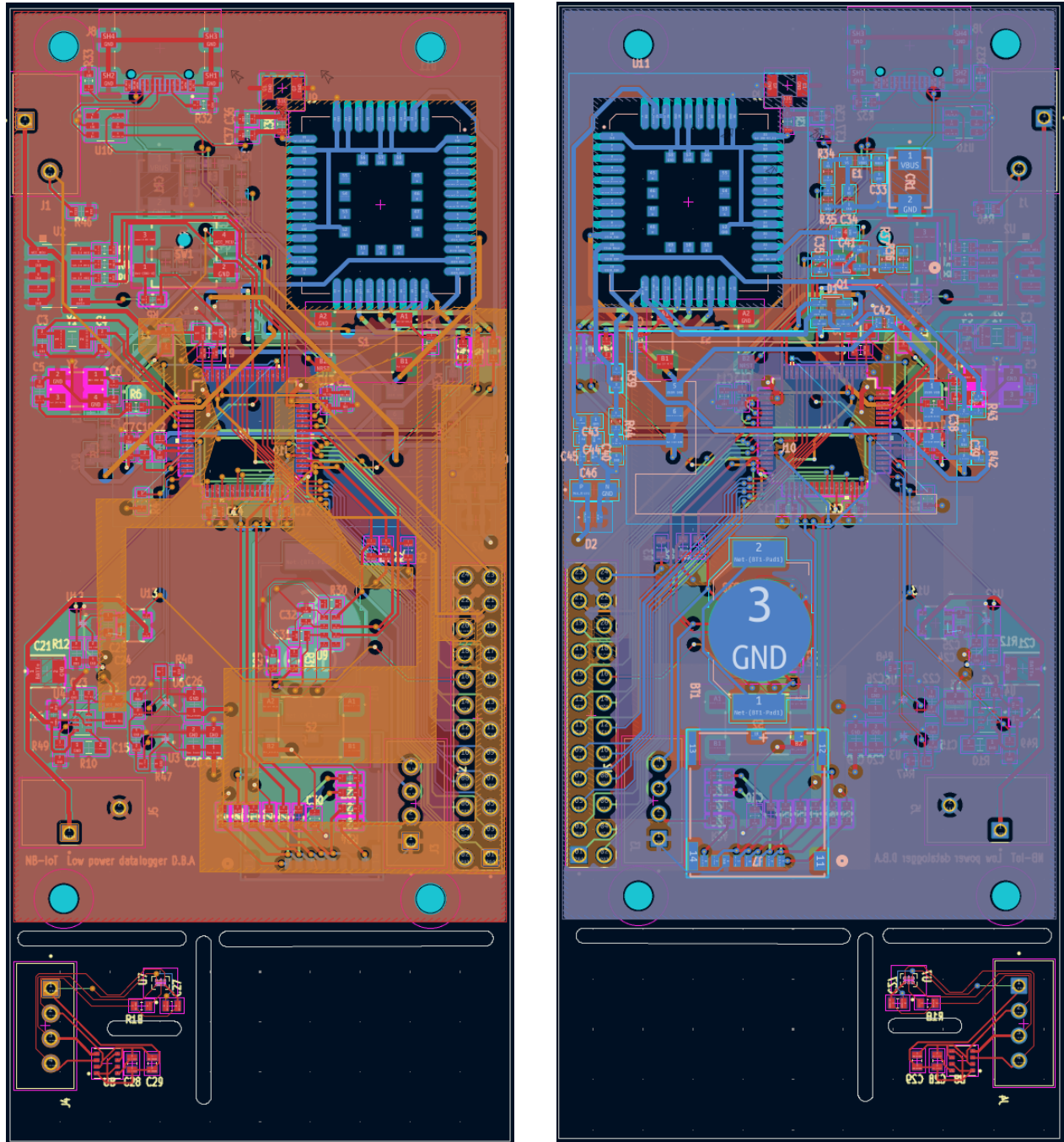
Figura 3.21: Modelo 3D de la PCB LoRa.

### 3.9. PCB NB-IoT

En la variante NB-IoT, las dimensiones y restricciones de ensamblaje del bloque de comunicación imponen diferencias relevantes respecto de las versiones LoRa. Debido al tamaño del módulo de comunicación y del *SIM holder*, ambos se ubican en la cara posterior de la PCB, junto con parte de sus componentes externos asociados, con el fin de optimizar el uso de área en la cara superior y mantener una distribución más limpia de los dominios de sensado y control.

La incorporación de este subsistema también desplaza la posición de los agujeros de montaje,

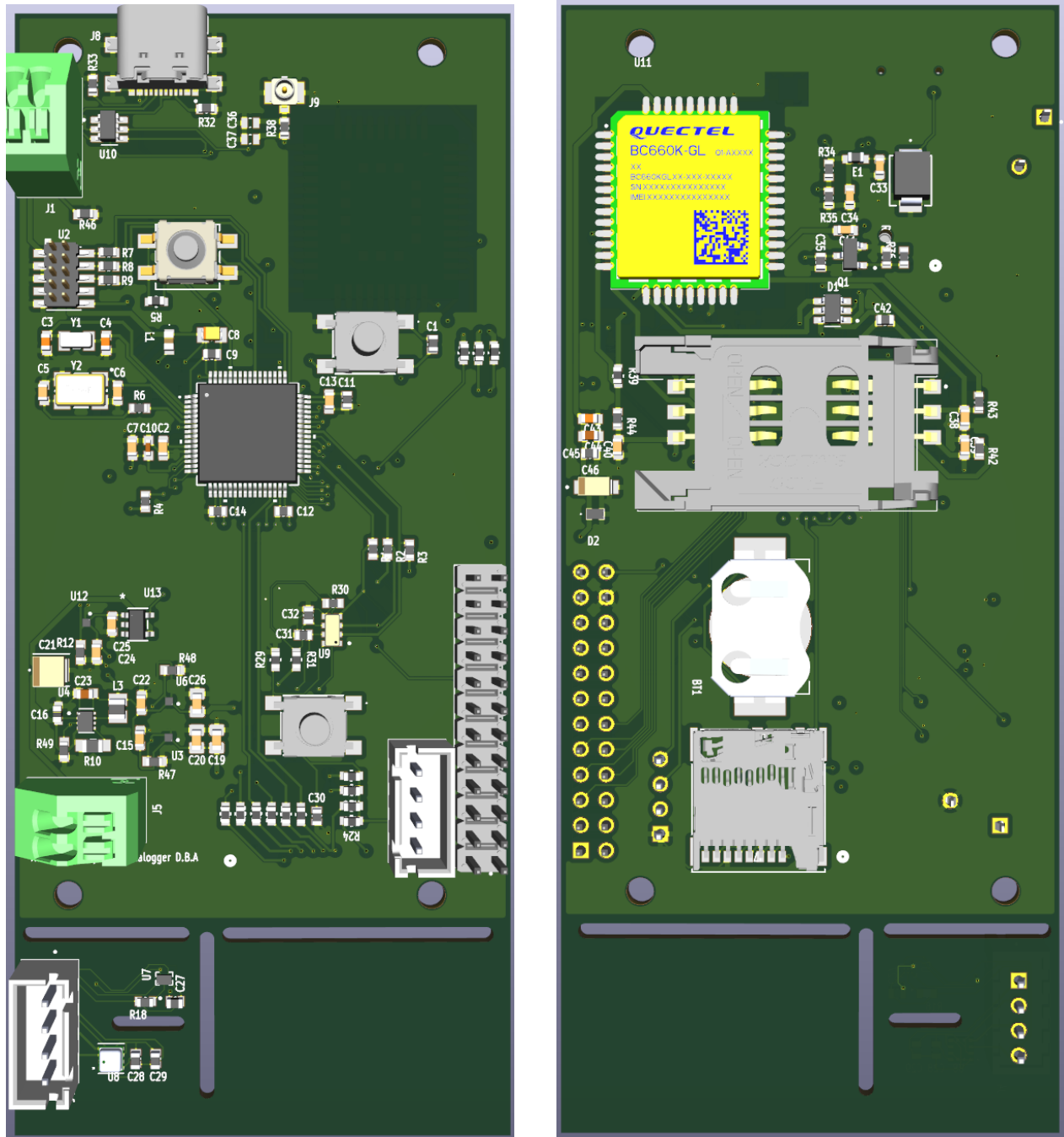
ya que se requiere respetar áreas de exclusión mecánica y eléctrica alrededor del módulo NB-IoT, su red de desacoplo y la ruta de RF. Como consecuencia, la PCB presenta un área total mayor que la de las versiones LoRa; sin embargo, se mantiene un tamaño inferior al de la caja, de modo que la integración final continúa siendo compatible con el gabinete seleccionado y permite conservar holguras para el cableado y los prensaestopas. La Fig. 3.22 muestra las vistas superior e inferior de la tarjeta, mientras que la Fig. 3.23 presenta el modelo 3D del ensamble para referencia mecánica.



(a) Vista frontal.

(b) Vista trasera.

Figura 3.22: PCB NB-IoT.



(a) Vista frontal.

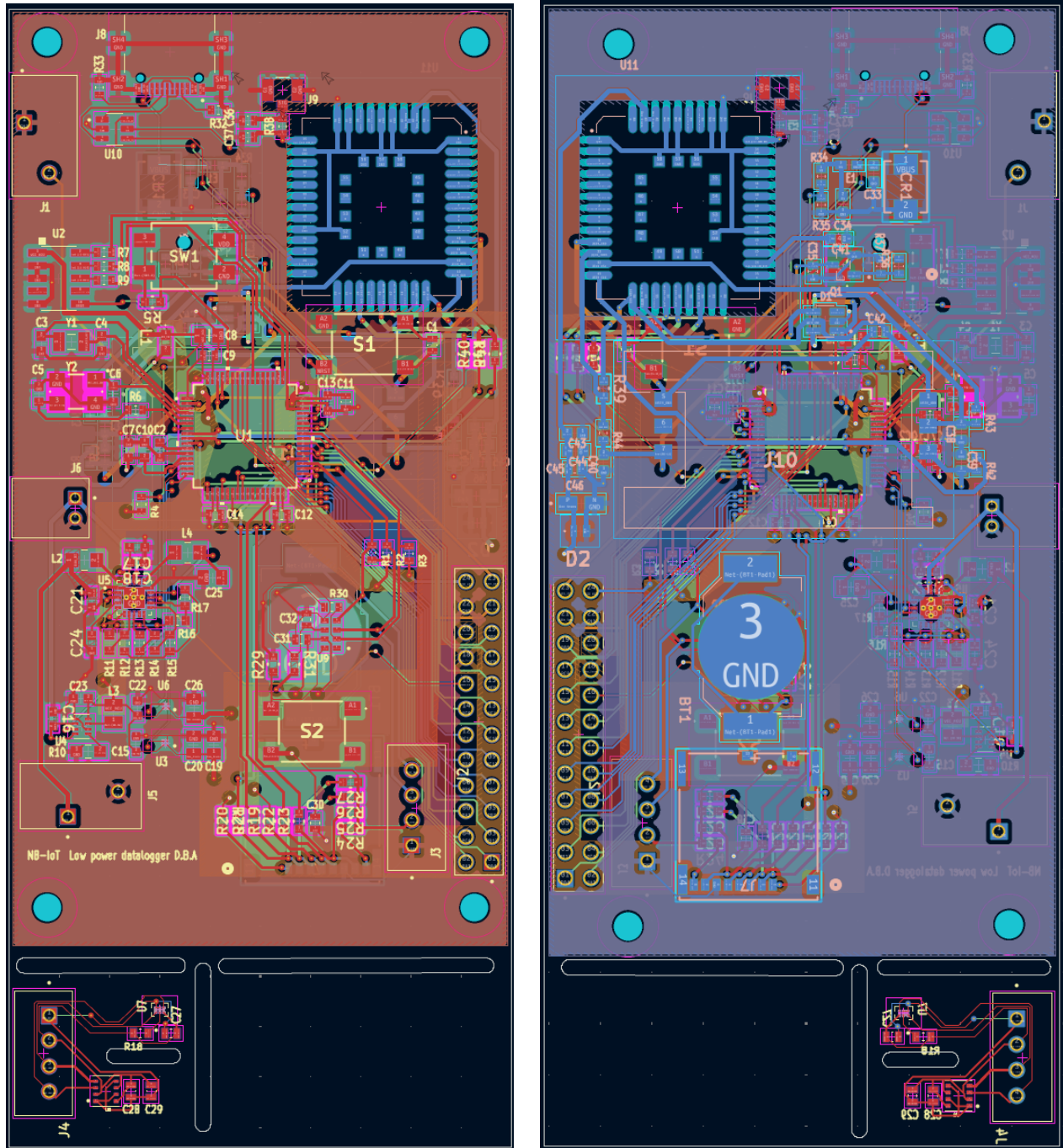
(b) Vista trasera.

Figura 3.23: Modelo 3D de la PCB NB-IoT.

### 3.10. PCB NB-IoT + solar

La PCB correspondiente a la variante NB-IoT con aporte solar mantiene la misma arquitectura general de la versión NB-IoT a batería, incorporando además el bloque de *energy harvesting* y los componentes asociados a la entrada fotovoltaica y carga. En consecuencia, la organización de bloques principales (microcontrolador, sensado, almacenamiento y comunicación NB-IoT) se conserva, mientras que el frente de potencia se complementa con el cargador y su red de pasivos recomendada.

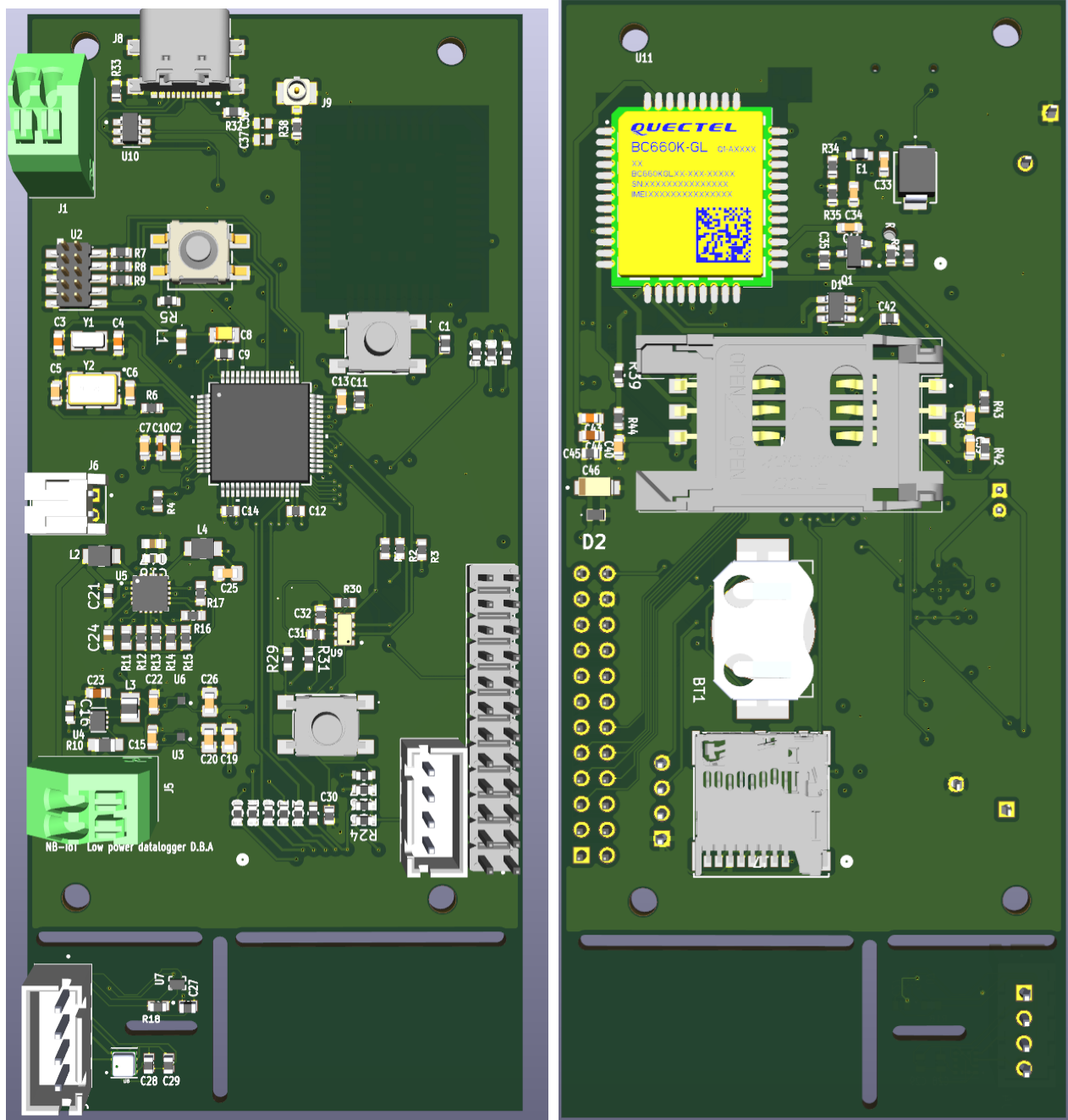
Debido al aumento de componentes en el subsistema de alimentación y a las áreas de exclusión necesarias para el *layout* del *harvester* y su ruteo, la posición de las monturas no coincide exactamente con la de la PCB NB-IoT sin solar. No obstante, el diseño mantiene compatibilidad con el gabinete seleccionado, conservando holguras para el cableado y permitiendo una integración mecánica equivalente a nivel de ensamblaje. La Fig. 3.24 muestra las vistas superior e inferior de la tarjeta, mientras que la Fig. 3.25 presenta el modelo 3D del ensamble para referencia mecánica.



(a) Vista frontal.

(b) Vista trasera.

Figura 3.24: PCB NB-IoT + solar.



(a) Vista frontal.

(b) Vista trasera.

Figura 3.25: Modelo 3D de la PCB NB-IoT + solar.

### 3.10.1. Selección de baterías y justificación

El sistema contempla dos configuraciones de almacenamiento de energía, definidas por el modo de operación, funcionamiento autónomo sin recarga y operación con recolección de energía solar. En ambos casos, la selección de la batería se realizó considerando compatibilidad eléctrica, integración mecánica y comportamiento ante cargas pulsantes asociadas a comunicaciones y escritura en

memoria.

### Opción no recargable: celda primaria Li-SOCl<sub>2</sub>

Para la variante autónoma sin capacidad de recarga se seleccionó una batería primaria de Litio–Cloruro de Tionilo (Li-SOCl<sub>2</sub>) **Saft LSH-20** en formato **D**, con tensión nominal de **3.6 V** y capacidad típica de **13 000 mAh**. La elección de la familia **LSH** (construcción tipo espiral) por sobre celdas Li-SOCl<sub>2</sub> de construcción tipo bobina responde a la naturaleza pulsante de la carga: el sistema presenta eventos de mayor consumo durante la transmisión inalámbrica y durante operaciones de escritura en tarjeta microSD. En baterías con alta resistencia interna, estos pulsos pueden generar caídas de tensión significativas (*voltage sag*), provocando reinicios del microcontrolador o fallas transitorias en periféricos [23].

La celda **LSH-20** está diseñada para entregar pulsos de corriente elevados (del orden de amperes), lo que reduce el riesgo de caídas abruptas de tensión durante eventos dinámicos y mejora la estabilidad operativa del sistema durante transmisiones y escritura de datos.

Considerando un consumo diario estimado de  $\sim 25.5$  mAh/d, la autonomía teórica se aproxima a:

$$\text{Autonomía} \approx \frac{13\,000 \text{ mAh}}{25.5 \text{ mAh/d}} \approx 510 \text{ días} \approx 1.4 \text{ años.} \quad (3.1)$$

lo que cumple con el objetivo de despliegue prolongado y bajo mantenimiento.

### Opción recargable: batería LiPo (formato plano)

Para la variante con recolección de energía solar se seleccionó una batería recargable de Polímero de Litio (**LiPo**) en formato *pouch* con tensión nominal de **3.7 V** y capacidad de **3500 mAh**, incorporando circuito de protección **PCM** (Protection Circuit Module). Esta selección se justifica principalmente por integración mecánica: el formato laminar permite un diseño de encapsulado de menor espesor y facilita la disposición interna del sistema en un volumen reducido, a diferencia de celdas cilíndricas rígidas.

Desde el punto de vista eléctrico, la tensión nominal de 3.7 V es compatible con la configuración de carga empleada, y su baja resistencia interna otorga una respuesta transitoria favorable ante pulsos de consumo, reduciendo la probabilidad de caídas de tensión que afecten la estabilidad del microcontrolador y del subsistema de comunicaciones.

La capacidad de **3500 mAh** se dimensionó con margen respecto al consumo diario del sistema ( $\sim 26$  mAh/día), de manera de mantener operación continua incluso ante ausencia prolongada de radiación solar. En condición extrema sin aporte del panel, la autonomía teórica sería<sup>1</sup>:

$$\text{Autonomía} \approx \frac{3500 \text{ mAh}}{26 \text{ mAh/día}} \approx 135 \text{ días.} \quad (3.2)$$

El uso de una celda con **PCM** integrado permite además proteger el sistema frente a sobrecarga, cortocircuitos y condiciones de operación fuera de rango, aumentando la seguridad y vida útil del conjunto.

---

<sup>1</sup>Los consumos y autonomías presentados corresponden a estimaciones aproximadas. Se calcularon a partir de corrientes típicas extraídas de hojas de datos de los componentes y de tiempos de escritura en microSD y de transmisión medidos experimentalmente en prototipos; por tanto, pueden variar según configuración de firmware, condiciones ambientales, estado de la batería y calidad de enlace.

# DESARROLLO DE FIRMWARE

Tras completar el diseño de hardware, se desarrolló el firmware que se ejecuta en el microcontrolador y coordina la interacción entre sensores, almacenamiento, comunicaciones y la interfaz de usuario. El propósito central del software es habilitar la adquisición periódica de variables meteorológicas, su registro local y, cuando corresponde, su transmisión inalámbrica. Dado que el sistema está orientado a operación autónoma de largo plazo, el firmware se diseñó priorizando el bajo consumo energético, explotando los modos de bajo consumo del **STM32L476RG** y controlando de forma explícita la energización de periféricos.

## 4.1. Plataforma de desarrollo

El firmware se implementó sobre el ecosistema de STMicroelectronics, utilizando **STM32CubeIDE v2.0.0** como entorno de desarrollo (basado en Eclipse), con el compilador **GNU Arm Embedded GCC** (`arm-none-eabi-gcc-13.3.1`) y herramientas de depuración integradas. La configuración de periféricos y la generación del código de inicialización (drivers *HAL* y *LL*) se realizó mediante **STM32CubeMX**, utilizando el paquete **STM32CubeFW\_L4 v1.18.2**, lo que permitió definir el árbol de relojes, el mapeo de pines GPIO y la inclusión de *middleware*. En particular, esta herramienta facilitó la integración de **FatFS** para la tarjeta microSD y la pila **USB Device CDC** para configuración y descarga de datos.

Durante la etapa de depuración se utilizó la configuración **Debug** con símbolos (`-g3`) y **optimización deshabilitada** (`-O0`), con el fin de facilitar el seguimiento de ejecución. Para pruebas de operación prolongada (múltiples días) y mediciones de consumo se utilizó una configuración **Release** optimizada para tamaño (`-Os`). En ambos casos se habilitó eliminación de secciones no utilizadas mediante `-ffunction-sections`, `-fdata-sections` y enlazado con `-gc-sections`, además del uso de `newlib-nano` (`-specs=nano.specs`).

La opción `-Os` se seleccionó para reducir el tamaño del binario y favorecer un comportamiento más representativo en consumo durante operación prolongada, mientras que `-O0` se reservó para depuración.

## 4.2. Arquitectura de software y gestión de energía

El flujo principal del programa se organiza como una máquina de estados finitos cuyo objetivo es maximizar el tiempo en suspensión profunda (*Stop Mode 2*). El microcontrolador controla la alimentación de periféricos mediante dos interruptores de carga **TPS22916CYFPT**, lo que permite energizar selectivamente dos dominios:

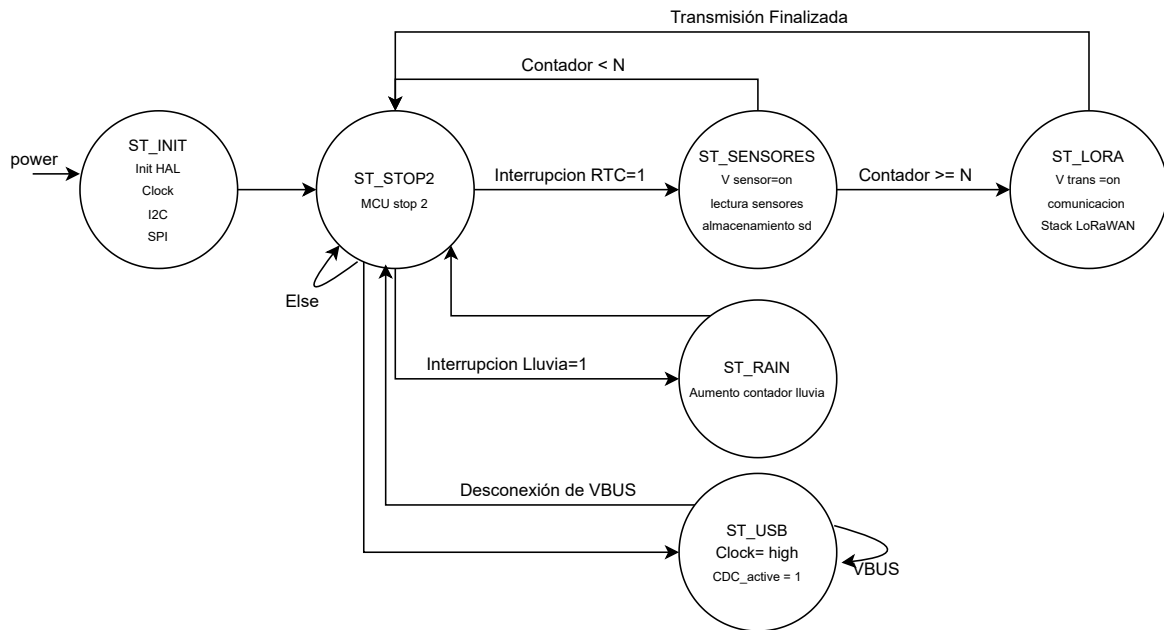


Figura 4.1: Máquina de estados finitos de la arquitectura de firmware del *data logger*.

1. **Sensores y almacenamiento:** TMP117, BME280, RTC y microSD.
2. **Comunicación:** módulo LoRa LR62E.

El ciclo de operación nominal consiste en permanecer en *Stop Mode 2* con reloj de baja velocidad activo y fuentes de interrupción habilitadas, despertando únicamente ante eventos definidos. La lógica de transición entre estados se implementa a partir de banderas de eventos (alarma RTC, pulso de pluviómetro, detección de USB) y contadores de muestreo/transmisión.

#### 4.2.1. Fuentes de reactivación ("*wake-up sources*")

El sistema sale del modo de suspensión mediante tres eventos principales:

- **Alarma del RTC (RV-3028):** interrupción periódica para iniciar el ciclo de muestreo.
- **Interrupción externa (pluviómetro):** generada por el cierre del interruptor magnético del balancín.
- **Detección de USB (VBUS):** generada al conectar el cable para configuración y descarga.

### 4.3. Sensores y periféricos

#### 4.3.1. Pluviómetro (interrupción externa)

El sensor de lluvia utiliza un mecanismo de balancín con interruptor magnético. En firmware, el pin asociado se configura como entrada con interrupción externa (*EXTI*). Cada pulso incrementa un contador acumulativo (por ejemplo, diario) almacenado en memoria no volátil o en el archivo de registro, de modo que el conteo no se pierda al retornar a suspensión.

### 4.3.2. Sensores ambientales (I<sup>2</sup>C)

La lectura de temperatura y presión/humedad se realiza sobre el bus I<sup>2</sup>C mediante:

- **TMP117**: lectura de temperatura.
- **BME280**: lectura de presión atmosférica y humedad relativa.

El flujo de lectura sigue una secuencia orientada a minimizar tiempo activo:

1. Activar el *load switch* del grupo de sensores.
2. Esperar el tiempo de estabilización (*settling time*).
3. Ejecutar transacciones I<sup>2</sup>C de configuración/lectura.
4. Procesar datos y apagar inmediatamente los periféricos.

### 4.3.3. Reloj de tiempo real (RV-3028)

El RTC externo **RV-3028** provee la referencia temporal del sistema y genera la interrupción de alarma que despierta al microcontrolador para el ciclo de muestreo. Al tratarse de un módulo independiente y de bajo consumo, permite mantener fecha/hora con deriva contenida incluso durante largos periodos de suspensión del MCU.

## 4.4. Almacenamiento local (microSD)

Para el almacenamiento local se utiliza una tarjeta microSD mediante SPI. La gestión del sistema de archivos se implementó con **FatFS** [24], integrado como *middleware* a través de **STM32CubeMX** (STM32CubeFW\_L4 v1.18.2) [25].

## 4.5. Comunicación LoRaWAN

La conectividad de largo alcance se gestiona mediante el módulo **LR62E** (radio **SX1262**). El *stack* de software utilizado corresponde a una adaptación de **LoRaMAC-node** [26] para operar sobre *HAL* en STM32, integrando la capa física (radio), la capa MAC de LoRaWAN y un *handler* de aplicación. Esta arquitectura permite separar responsabilidades: el driver de radio abstrae el acceso SPI y líneas *BUSY/DI01/RESET*, mientras que la capa *LoRaMAC* implementa procedimientos LoRaWAN como *join* (OTAA/ABP), control de sesiones y claves, configuración regional y planificación de ventanas de recepción.

En particular, la pila LoRaWAN se compone de los siguientes bloques:

- **Radio driver (SX126x)**: controla el transceiver (modulación, potencia, frecuencia, IRQs y buffers TX/RX) y entrega eventos a la capa superior a través de **RadioEvents** y la interrupción *DI01*.
- **LoRaMAC (capa MAC)**: implementa el protocolo LoRaWAN (formato de tramas, contadores de *frame*, *MIC*, cifrado, ADR, *duty-cycle*, *retransmissions* en confirmados y manejo de *Class A*). Esta capa define además las temporizaciones de **RX1/RX2** tras cada transmisión.

- **LmHandler (capa de aplicación)**: simplifica el uso del stack exponiendo funciones de alto nivel para inicializar, unir a la red, enviar tramas y recibir datos, encapsulando *callbacks* de eventos (p. ej., `OnJoinRequest`, `OnTxData`, `OnRxData`).

La configuración regional se ajusta a la banda empleada en la infraestructura de pruebas (gateway **Dragino LPS8**), definiendo el plan de canales correspondiente y parámetros como potencia máxima, *data rates* y temporizaciones. Debido a que el nodo opera en **bajo consumo** y despierta de forma periódica, se privilegia el envío **no confirmado** para minimizar el tiempo activo (se evita la sobrecarga de retransmisiones y se reduce el costo energético asociado a ventanas RX). Además, el dominio de alimentación del módulo LoRa se energiza únicamente cuando corresponde transmitir, manteniendo el transceiver apagado el resto del tiempo.

Con el fin de reducir consumo, el sistema no transmite en cada muestreo, se utiliza un contador de mediciones configurable (por defecto  $N = 15$ ) para agrupar información y enviar un resumen periódico. Este enfoque reduce la fracción de tiempo en estado activo, disminuye el número de transacciones SPI y eventos de radio, y evita el costo energético de múltiples aperturas de ventanas de recepción.

#### 4.5.1. Lógica de transmisión

El algoritmo de transmisión se resume en:

1. El sistema despierta mediante evento del RTC e inicia el ciclo de adquisición.
2. Se energiza el riel de sensores, se inicializan los periféricos necesarios y se adquieren las mediciones (temperatura, humedad, presión y lluvia acumulada).
3. Si el contador de muestras  $< N$ , se registra la muestra en microSD y el sistema retorna a *Stop Mode 2*.
4. Si el contador alcanza  $N$ , se ejecuta el ciclo de transmisión:
  - Se calcula el promedio (o estadísticos simples) de temperatura, presión y humedad desde el último envío.
  - Se obtiene el total de lluvia acumulada del periodo (por pulsos del pluviómetro).
  - Se energiza el dominio de comunicación (LR62E) y se inicializa el driver SX126x.
  - Se verifica el estado de red (p. ej., *joined*); si no existe sesión activa, se ejecuta el procedimiento de *join* mediante OTAA y se espera el resultado mediante *callbacks*.
  - Se construye el *payload* de aplicación con campos compactos (ID del nodo, marca de tiempo, promedios y lluvia), cuidando el tamaño para adecuarse al *data rate* y limitar el *airtime*.
  - Se transmite utilizando `LmHandlerSend()` en modo no confirmado. Posteriormente, el stack abre automáticamente las ventanas **RX1/RX2** para recepción (downlink) en caso de ser necesario.
  - Una vez completado el envío, se apaga el módulo LoRa, se reinician contadores de agregación y el sistema retorna a suspensión profunda.

## 4.6. Interfaz USB de configuración (Microshell)

Se implementó una clase USB CDC que emula un puerto serial virtual. Al detectar VBUS, el firmware abandona el flujo de bajo consumo y entra en un modo de configuración. Sobre esta interfaz se desarrolló una *microshell* que permite, desde un terminal serial, configurar la hora del RTC, ajustar intervalos y descargar registros desde la microSD. Al desconectar USB, el sistema retorna automáticamente a *Stop Mode 2*.

## 4.7. Implementación de *drivers* y utilidades

Esta sección presenta fragmentos representativos de los módulos desarrollados. Se muestran las funciones clave asociadas a comunicación I<sup>2</sup>C, inicialización, lectura de datos y la interfaz de comandos USB.

### 4.7.1. Driver TMP117: lectura de registros, configuración y conversión a temperatura

El driver del sensor TMP117 se implementó sobre transacciones I<sup>2</sup>C del tipo HAL\_I2C\_Mem\_Read / HAL\_I2C\_Mem\_Write, utilizando direcciones de registro de 8 bits y datos de 16 bits en formato *MSB-first*. A diferencia de una versión acoplada a una instancia fija del bus, la implementación actual recibe un puntero I2C\_HandleTypeDef\* en la API pública, permitiendo reutilizar el driver en distintos buses o configuraciones del sistema.

El Listado 4.7.1 resume las rutinas principales: (i) lectura/escritura de registros de 16 bits, (ii) verificación de presencia mediante Device\_ID, (iii) configuración del modo de conversión, promediado y tiempo de conversión en modo continuo, y (iv) conversión de la lectura cruda a grados Celsius. El TMP117 entrega la temperatura como un entero con signo (dos complementos) donde **1 LSB equivale a 0,0078125 °C**; por lo tanto, la conversión se realiza multiplicando el valor firmado por dicho factor. Adicionalmente, se incluye un flujo de medición *one-shot* que dispara una conversión, espera la bandera DATA\_READY y luego lee el registro de temperatura.

———— Listado 4.7.1: Rutinas principales del driver TMP117 (I<sup>2</sup>C, configuración y lectura). ————

```

1 static HAL_StatusTypeDef tmp117_read_reg16(I2C_HandleTypeDef *hi2c,
2                                           uint8_t reg, uint16_t *val)
3 {
4     uint8_t buf[2];
5     HAL_StatusTypeDef st;
6
7     if (val == NULL) return HAL_ERROR;
8
9     st = HAL_I2C_Mem_Read(hi2c, g_tmp117_i2c_addr, reg, 1, buf, 2, TIMEOUT_100MS);
10    if (st != HAL_OK) return st;
11
12    *val = (uint16_t)((buf[0] << 8) | buf[1]); // MSB first
13    return HAL_OK;
14 }
15
16 static HAL_StatusTypeDef tmp117_write_reg16(I2C_HandleTypeDef *hi2c,
17                                             uint8_t reg, uint16_t val)
18 {
19     uint8_t buf[2];
20     buf[0] = (uint8_t)(val >> 8);
21     buf[1] = (uint8_t)(val & 0xFF);

```

```

22
23     return HAL_I2C_Mem_Write(hi2c, g_tmp117_i2c_addr, reg, 1, buf, 2, TIMEOUT_100MS);
24 }
25
26 static float tmp117_raw_to_c(int16_t raw)
27 {
28     // 1 LSB = 0.0078125 °C
29     return ((float)raw) * TMP117_RESOLUTION_C;
30 }
31
32 bool tmp117_init(I2C_HandleTypeDef *hi2c)
33 {
34     uint16_t id = 0;
35
36     // Read Device ID (0x0F). Expected default = 0x0117.
37     if (tmp117_read_reg16(hi2c, TMP117_REG_DEVICE_ID, &id) != HAL_OK)
38         return false;
39
40     if (id != TMP117_ID_EXPECTED)
41         return false;
42
43     return true;
44 }
45
46 bool tmp117_read_temperature_c(I2C_HandleTypeDef *hi2c, float *temp_c)
47 {
48     uint16_t raw_u = 0;
49     int16_t raw_s;
50
51     if (temp_c == NULL) return false;
52
53     if (tmp117_read_reg16(hi2c, TMP117_REG_TEMP_RESULT, &raw_u) != HAL_OK)
54         return false;
55
56     raw_s = (int16_t)raw_u;
57     *temp_c = tmp117_raw_to_c(raw_s);
58     return true;
59 }
60
61 bool tmp117_one-shot_read_temperature_c(I2C_HandleTypeDef *hi2c,
62                                         float *temp_c, uint32_t timeout_ms)
63 {
64     // Trigger one-shot conversion, wait for Data_Ready, then read temperature.
65     if (!tmp117_set_mode(hi2c, TMP117_MODE_ONESHOT)) return false;
66     HAL_Delay(20);
67
68     if (!tmp117_wait_data_ready(hi2c, timeout_ms)) return false;
69
70     return tmp117_read_temperature_c(hi2c, temp_c);
71 }

```

#### 4.7.2. Driver RV-3028: fecha/hora, configuración de respaldo y generación de eventos

El RTC RV-3028-C7 almacena la información de calendario en registros codificados en BCD, por lo que el driver incorpora funciones de conversión BCD–binario y rutinas de acceso I<sup>2</sup>C en

modo Mem\_Read/Mem\_Write. En esta implementación, el periférico se abstrae mediante una estructura `rv3028_t` que encapsula el `I2C_HandleTypeDef*` y la dirección del dispositivo, facilitando la reutilización del driver en distintos buses o variantes de hardware.

El Listado 4.7.2 resume el núcleo del acceso a fecha/hora: conversión BCD, lectura en bloque de 7 bytes desde el registro de segundos y escritura de la estructura `rv3028_datetime_t`. La lectura considera las máscaras de validez definidas por el fabricante (por ejemplo, bits de control en segundos/minutos/horas) y reconstruye el año como un *offset* respecto al 2000.

————— Listado 4.7.2: Núcleo del driver RV-3028: conversiones BCD e I<sup>2</sup>C para fecha/hora. —————

```

1  static uint8_t bcd_to_bin(uint8_t bcd)
2  {
3      return (uint8_t)(((bcd >> 4) * 10U) + (bcd & 0x0FU));
4  }
5
6  static uint8_t bin_to_bcd(uint8_t bin)
7  {
8      return (uint8_t)(((bin / 10U) << 4) | (bin % 10U));
9  }
10
11 static HAL_StatusTypeDef rv3028_read(rv3028_t *dev, uint8_t reg,
12                                     uint8_t *data, uint16_t len)
13 {
14     return HAL_I2C_Mem_Read(dev->i2c_handle, dev->dev_addr,
15                             reg, 1, data, len, TIMEOUT_100MS);
16 }
17
18 static HAL_StatusTypeDef rv3028_write(rv3028_t *dev, uint8_t reg,
19                                       const uint8_t *data, uint16_t len)
20 {
21     return HAL_I2C_Mem_Write(dev->i2c_handle, dev->dev_addr,
22                              reg, 1, (uint8_t*)data, len, TIMEOUT_100MS);
23 }
24
25 bool rv3028_get_datetime(rv3028_t *dev, rv3028_datetime_t *dt)
26 {
27     uint8_t buf[7];
28     if (!dev || !dt) return false;
29
30     if (rv3028_read(dev, RV3028_REG_SECONDS, buf, sizeof(buf)) != HAL_OK)
31         return false;
32
33     dt->seconds = bcd_to_bin(buf[0] & 0x7F);
34     dt->minutes = bcd_to_bin(buf[1] & 0x7F);
35     dt->hours   = bcd_to_bin(buf[2] & 0x3F);
36     dt->weekday = (uint8_t)(buf[3] & 0x07);
37     dt->date    = bcd_to_bin(buf[4] & 0x3F);
38     dt->month   = bcd_to_bin(buf[5] & 0x1F);
39     dt->year    = (uint16_t)(2000U + bcd_to_bin(buf[6]));
40
41     return true;
42 }
43
44 bool rv3028_set_datetime(rv3028_t *dev, const rv3028_datetime_t *dt)
45 {
46     uint8_t buf[7];
47     if (!dev || !dt) return false;
48
49     buf[0] = bin_to_bcd(dt->seconds);

```

```

50  buf[1] = bin_to_bcd(dt->minutes);
51  buf[2] = bin_to_bcd(dt->hours);
52  buf[3] = (uint8_t)(dt->weekday & 0x07);
53  buf[4] = bin_to_bcd(dt->date);
54  buf[5] = bin_to_bcd(dt->month);
55  buf[6] = bin_to_bcd((uint8_t)(dt->year - 2000U));
56
57  return (rv3028_write(dev, RV3028_REG_SECONDS, buf, sizeof(buf)) == HAL_OK);
58 }

```

Además del manejo de calendario, el driver incorpora rutinas orientadas a operación en bajo consumo y robustez tras ciclos de apagado/encendido del riel de sensores. En `rv3028_init()` se realiza una verificación de comunicación leyendo el identificador del dispositivo, incluyendo reintentos para mitigar fallas transitorias por ruido en el bus. Luego, se configura el bloque de respaldo para operación segura en escenarios con alimentación principal intermitente: se desactiva temporalmente la actualización automática desde EEPROM (bit `EERD`) para evitar condiciones indeterminadas durante la escritura, se ajusta el modo de conmutación de respaldo (BSM) y detección de flanco rápido (`FEDE`), y se ejecuta la secuencia de *commit* de RAM a EEPROM mediante el comando de actualización. Finalmente, se fuerza el modo horario de 24 h y se limpian banderas de arranque (`PORF`) para dejar el RTC en un estado conocido.

Para la salida desde modos de bajo consumo, `rv3028_wakeup_init()` implementa una rutina de re-sincronización posterior al retorno de alimentación: limpia condiciones de error del periférico I<sup>2</sup>C, inserta un retardo breve de estabilización y confirma la comunicación leyendo el registro de estado. En caso de existir banderas de evento (alarma o temporizador), estas se limpian explícitamente escribiendo ceros sobre los bits correspondientes, evitando re-disparos o lecturas ambiguas al retomar la ejecución.

Finalmente, el driver expone la configuración de eventos temporales para *wake-up* y planificación del muestreo. Se implementa una alarma diaria (`rv3028_alarm_set_daily`) que programa horas/minutos y habilita la interrupción asociada, así como un temporizador de cuenta regresiva (`rv3028_timer_set`) con selección de frecuencia y opción de repetición. La gestión de interrupciones se simplifica con `rv3028_handle_irq()`, que lee el registro de estado y limpia las banderas AF y TF cuando corresponda, permitiendo integrar el RTC con una línea de interrupción externa (EXTI) hacia el microcontrolador.

### 4.7.3. Microshell USB-CDC: parseo de comandos y despacho

La microshell implementa un búfer de línea, tokenización por espacios y un despachador de comandos (`help`, `rtc`, `wake`, `txint`, `data`). El Listado 4.7.3 muestra el flujo central de ejecución de una línea y el despacho de comandos.

\_\_\_\_\_ Listado 4.7.3: Tokenización y despacho de comandos en Microshell. \_\_\_\_\_

```

1  static int tokenize(char *line, char **argv, int max_tokens)
2  {
3      int argc = 0;
4      char *p = line;
5
6      while (*p && argc < max_tokens)
7      {
8          while (*p == ' ' || *p == '\t') p++;
9          if (!*p) break;
10

```

```

11     argv[argc++] = p;
12
13     while (*p && *p != ' ' && *p != '\t') p++;
14     if (*p) { *p = '\0'; p++; }
15 }
16 return argc;
17 }
18
19 static void exec_line(char *line)
20 {
21     char *argv[MICROSHELL_MAX_TOKENS] = {0};
22     int argc = tokenize(line, argv, MICROSHELL_MAX_TOKENS);
23
24     if (argc == 0) return;
25
26     if (strcmp(argv[0], "help") == 0 || strcmp(argv[0], "?") == 0) { cmd_help(); return; }
27     if (strcmp(argv[0], "rtc") == 0) { cmd_rtc(argc, argv); return; }
28     if (strcmp(argv[0], "wake") == 0) { cmd_interval("wake", argc, argv,
29                                                    s_hooks.wakeup_get_s, s_hooks.wakeup_set_s); return; }
30     if (strcmp(argv[0], "txint") == 0) { cmd_interval("txint", argc, argv,
31                                                    s_hooks.txint_get_s, s_hooks.txint_set_s); return; }
32     if (strcmp(argv[0], "data") == 0) { cmd_data(argc, argv); return; }
33
34     ms_err("unknown cmd (try help)");
35 }

```

---

#### 4.7.4. Driver BME280: configuración *forced* y lectura compensada

El BME280 se opera en modo *forced* para minimizar consumo: el sensor permanece en reposo y sólo mide cuando el firmware gatilla una conversión. El Listado 4.7.4 incluye la configuración base y el disparo de medición.

----- Listado 4.7.4: Configuración y disparo de medición en BME280 (modo *forced*). -----

```

1 void bme280_config_weather_forced(void)
2 {
3     uint8_t config = 0;
4     config |= (BME280_FILTER_OFF << 2);
5     config |= (BME280_TSB_1000MS << 5);
6     bme280_write_u8(BME280_REG_CONFIG, config);
7
8     bme280_write_u8(BME280_REG_CTRL_HUM, BME280_OSRS_X1);
9
10    uint8_t ctrl_meas = 0;
11    ctrl_meas |= (BME280_OSRS_X1 << 5);
12    ctrl_meas |= (BME280_OSRS_X1 << 2);
13    ctrl_meas |= BME280_MODE_SLEEP;
14    bme280_write_u8(BME280_REG_CTRL_MEAS, ctrl_meas);
15 }
16
17 void bme280_trigger_forced(void)
18 {
19     uint8_t ctrl_meas = bme280_read_u8(BME280_REG_CTRL_MEAS);
20     ctrl_meas = (uint8_t)((ctrl_meas & ~0x03) | BME280_MODE_FORCED);
21     bme280_write_u8(BME280_REG_CTRL_MEAS, ctrl_meas);
22 }

```

---

#### 4.7.5. Registro en microSD y secuencia de reactivación de periféricos tras *Stop Mode 2*

El registro local se realiza en una tarjeta microSD mediante SPI y el sistema de archivos **FatFS**. Dado que el firmware prioriza el bajo consumo, durante suspensión profunda (*Stop Mode 2*) se desenergiza el dominio de sensores/almacenamiento (**V\_SENSOR**) y se desinicializan periféricos (I<sup>2</sup>C/SPI), estacionando pines en modo análogo (*pin parking*) para reducir fugas. Al reactivarse por interrupción del RTC, es necesario ejecutar una secuencia explícita de encendido y reinicialización antes de acceder al bus y a la microSD.

En la práctica, la transición entre un estado inicial orientado a depuración/configuración (con reloj principal habilitado y periféricos inicializados para USB) y el estado de bajo consumo implica que, tras volver de *Stop Mode 2*, el contenido de registros de configuración de I<sup>2</sup>C/SPI puede no ser válido o quedar en un estado inconsistente. Adicionalmente, si algún esclavo I<sup>2</sup>C permanece energizado parcialmente o retoma más lento que el MCU, el bus puede quedar bloqueado (por ejemplo, con SDA en bajo). Por ello, previo a la lectura de sensores y escritura en SD se ejecutan, encendido del riel **V\_SENSOR** y retardo de estabilización, forzado de **CS** de la SD a nivel alto para evitar selección espuria, recuperación del bus I<sup>2</sup>C mediante pulsos de reloj y condición **STOP**, *reset* forzado de los bloques periféricos (**RCC**) y (v) reinicialización de **I2C2/SPI2**. La escritura en SD se realiza como *append* sobre un archivo CSV, agregando encabezado si el archivo no existía.

El Listado 4.7.5 resume la lógica utilizada para montar FatFS, agregar una línea al CSV y la secuencia de reactivación en el evento de *wake-up* antes de llamar a la rutina de registro.

—— Listado 4.7.5: Secuencia de reactivación y registro en microSD (FatFS) tras *Stop Mode 2*. ——

```

1 // --- SD/FatFS: montaje y escritura en modo append ---
2 static bool sd_ready = false;
3 static const char *CSV_HEADER = "rain,temp,hum,press,date,time\r\n";
4
5 bool sd_init_mount(void)
6 {
7     if (sd_ready) return true;
8
9     DSTATUS st = disk_initialize(0);
10    if (st & STA_NOINIT) return false;
11
12    if (f_mount(&USERFatFS, USERPath, 1) != FR_OK) return false;
13
14    sd_ready = true;
15    return true;
16 }
17
18 bool sd_append_line(const char *line)
19 {
20     FIL fil; UINT bw; FRESULT fr;
21
22     if (!sd_init_mount()) return false;
23
24     fr = f_open(&fil, "0:/DATA.csv", FA_OPEN_ALWAYS | FA_WRITE);
25     if (fr != FR_OK) return false;
26
27     if (f_size(&fil) == 0)
28         (void)f_write(&fil, CSV_HEADER, (UINT)strlen(CSV_HEADER), &bw);
29
30     (void)f_lseek(&fil, f_size(&fil));
31     (void)f_write(&fil, line, (UINT)strlen(line), &bw);
32     (void)f_sync(&fil);

```

```
33     f_close(&fil);
34
35     return (fr == FR_OK);
36 }
37
38 // --- Wake-up: encendido y re-init para asegurar I2C/SPI operables ---
39 if (g_rtc_irq)
40 {
41     g_rtc_irq = 0;
42     sd_ready = false;
43
44     // 1) Energizar sensores/SD y esperar settling
45     V_SENSOR_On();
46     HAL_Delay(100);
47
48     // 2) Asegurar CS de SD en HIGH (evita selección y consumo espurio)
49     __HAL_RCC_GPIOB_CLK_ENABLE();
50     GPIO_InitTypeDef cs_init = {0};
51     cs_init.Pin = SD_CS_Pin;
52     cs_init.Mode = GPIO_MODE_OUTPUT_PP;
53     cs_init.Pull = GPIO_NOPULL;
54     cs_init.Speed = GPIO_SPEED_FREQ_LOW;
55     HAL_GPIO_Init(SD_CS_GPIO_Port, &cs_init);
56     HAL_GPIO_WritePin(SD_CS_GPIO_Port, SD_CS_Pin, GPIO_PIN_SET);
57
58     // 3) Reinicio controlado de periféricos y recuperación del bus I2C
59     HAL_I2C_DeInit(&hi2c2);
60     HAL_SPI_DeInit(&hspi2);
61     I2C2_BusRecover();
62
63     __HAL_RCC_I2C2_FORCE_RESET(); HAL_Delay(1);
64     __HAL_RCC_I2C2_RELEASE_RESET();
65
66     __HAL_RCC_SPI2_FORCE_RESET(); HAL_Delay(1);
67     __HAL_RCC_SPI2_RELEASE_RESET();
68
69     // 4) Re-inicialización (CubeMX)
70     MX_I2C2_Init();
71     MX_SPI2_Init();
72
73     // 5) Validación de RTC post-wakeup y registro de muestra
74     if (rv3028_wakeup_init(&rtc_dev))
75     {
76         bme280_init(&hi2c2);
77         tmp117_init(&hi2c2);
78         log_one_sample(); // lee RTC + sensores y escribe en SD (CSV)
79     }
80
81     // 6) Limpiar flags RTC y volver a preparar suspensión profunda
82     rv3028_clear_flags(&rtc_dev, RV3028_FLAG_TF);
83
84     HAL_SPI_DeInit(&hspi2);
85     HAL_I2C_DeInit(&hi2c2);
86     Pins_Park();
87     V_SENSOR_Off();
88
89     Enter_Stop_Mode();
90 }
```

---

En resumen, la secuencia de apagado/encendido y reinicialización no es un detalle accesorio, es un requisito práctico para asegurar que, tras períodos prolongados de suspensión y desenergización del dominio de sensores/almacenamiento, las transacciones I<sup>2</sup>C/SPI retomen en un estado determinístico y se eviten fallas por *bus lock* o pérdida de configuración en registros del periférico.

# RECEPCIÓN DE DATOS

Una vez que el nodo LoRaWAN transmite una *uplink*, los datos son recibidos por la puerta de enlace (*gateway*) y encaminados hacia un servidor local. En esta implementación se utilizó una **Raspberry Pi 3 B v1.2** como plataforma de cómputo y una puerta de enlace **Dragino LPS8N** configurada en modo *Semtech UDP packet forwarder*. Sobre la Raspberry Pi se desplegó un *stack* local basado en **ChirpStack** [27] (funcionalmente equivalente a una plataforma tipo TTN, pero alojada localmente), permitiendo registrar *gateways*, *devices* y aplicaciones, además de persistir eventos y mediciones en una base de datos para su posterior explotación.

## 5.1. Gateway y servidor LoRaWAN local

La Figuración general del flujo de datos es la siguiente:

1. El **Dragino LPS8N** recibe tramas LoRa en la banda **AU915** y las encapsula como datagramas UDP.
2. La **Raspberry Pi** recibe estos datagramas en el puerto **1700/UDP** y los procesa mediante el servicio **ChirpStack Gateway Bridge** (backend Semtech UDP).
3. El **Gateway Bridge** publica los eventos (tramas, estadísticas, *join*, etc.) en un *broker* MQTT (**Mosquitto**) utilizando un prefijo de tópicos asociado a la región (**au915\_0**).
4. **ChirpStack** consume estos mensajes MQTT, aplica la lógica LoRaWAN (por ejemplo, *join* OTAA, seguridad y *routing*) y registra la información operacional y de aplicaciones en la base de datos.

En términos prácticos, el despliegue se realizó íntegramente mediante contenedores (**Docker + Docker Compose**), lo que permite reproducibilidad y mantenimiento simple. Un punto relevante observado durante la puesta en marcha fue la conectividad en red local: la resolución por `.local` (mDNS) no resultó confiable en el entorno utilizado, por lo que la configuración del gateway se realizó apuntando explícitamente a la **IP LAN** de la Raspberry Pi (obtenida con `hostname -I`).

## 5.2. Servicios de software utilizados

El sistema se desplegó mediante el repositorio **chirpstack-docker**, el cual orquesta múltiples servicios. A continuación se describe el rol de los principales componentes utilizados:

- **Docker / Docker Compose**: encapsula los servicios del *stack* (MQTT, servidor LoRaWAN, base de datos, etc.) y permite iniciar/detener el sistema con comandos simples, manteniendo configuraciones y volúmenes persistentes.

- **ChirpStack**: servidor LoRaWAN local que administra *tenants*, *gateways*, *device profiles*, *applications* y dispositivos, además de procesar *joins* y *uplinks*.
- **ChirpStack Gateway Bridge**: puente entre el *packet forwarder* UDP del gateway (Semtech) y la capa de mensajería MQTT. En esta implementación se configuró el prefijo de tópicos como `au915_0` para compatibilidad con AU915 subbanda 0.
- **Mosquitto (MQTT)**: *message broker* utilizado para transportar eventos del *gateway* hacia ChirpStack, y para habilitar integraciones posteriores.
- **PostgreSQL**: base de datos relacional empleada para persistir la configuración (entidades, llaves, perfiles) y los eventos/telemetría gestionados por el *stack*.
- **Redis** (si está presente en el *stack*): utilizado como almacenamiento en memoria para colas, *caching* y coordinación interna, mejorando desempeño del procesamiento de mensajes.

### 5.3. Configuración regional AU915

Dado que el sistema se operó en banda **AU915**, fue necesario alinear la configuración del *stack* con la subbanda utilizada por el gateway. Para ello se definió `enabled_regions = ["au915_0"]` en la configuración del servidor, y se ajustó el **topic prefix** MQTT del *Gateway Bridge* a `au915_0`. Adicionalmente, se realizó una limpieza de archivos de configuración de regiones no utilizadas (por ejemplo EU868/US915) para reducir ambigüedad y riesgo de errores de configuración en iteraciones futuras.

### 5.4. Alta de gateway y dispositivo en ChirpStack

Con el *stack* en ejecución, el procedimiento de operación consistió en:

1. Registrar el **gateway** en la interfaz web de ChirpStack, asegurando que el **Gateway ID** coincidiera con el observado en los tópicos MQTT (`au915_0/gateway/<gatewayid>/...`).
2. Crear un **device profile** para AU915, consistente con la versión LoRaWAN implementada en el nodo (parámetros MAC/región).
3. Crear una **aplicación** y registrar el **dispositivo** (OTAA), ingresando los identificadores y llaves (DevEUI/JoinEUI/AppKey; y NwkKey si aplica).

Una vez completados estos pasos, la verificación funcional se realizó revisando eventos de *join* y *uplink* en la interfaz de ChirpStack, y corroborando el tráfico UDP/MQTT con herramientas de diagnóstico (`tcpdump`, `mosquitto_sub`).

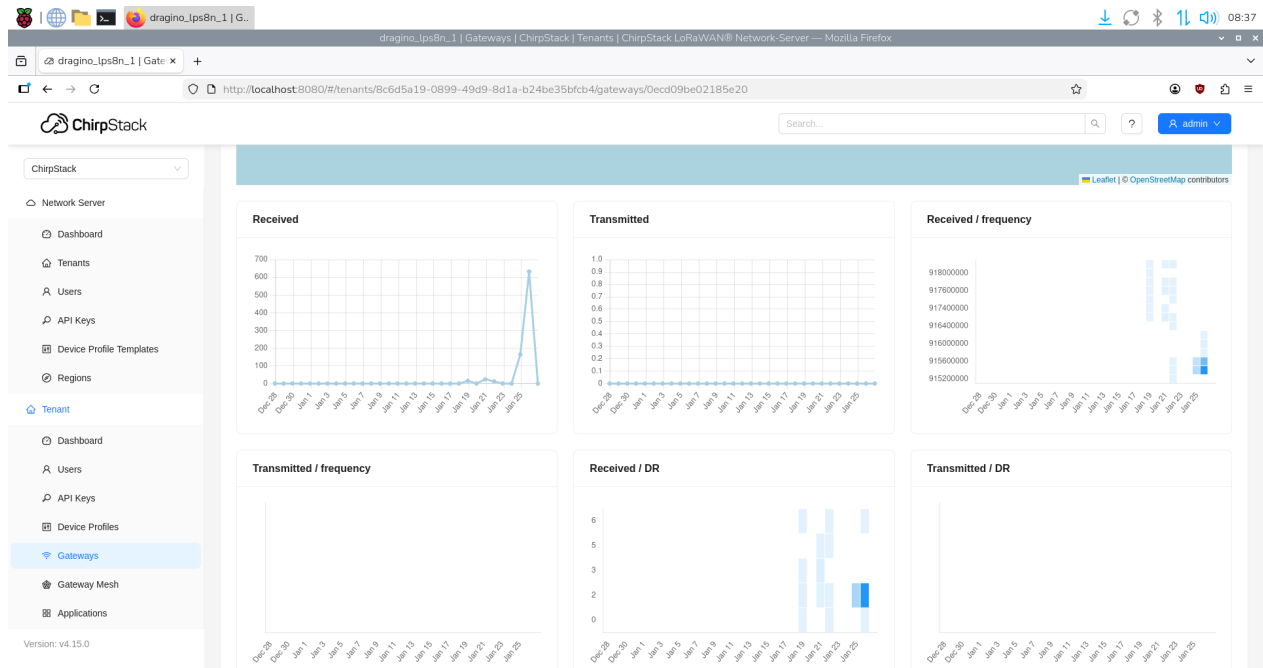


Figura 5.1: Interfaz web de ChirpStack en la Raspberry Pi mostrando el estado del *gateway*.

## 5.5. Visualización

Con el objetivo de disponer de una visualización local de los datos recibidos por el servidor, se incorporó una capa adicional de almacenamiento temporal y *dashboarding* sobre la misma **Raspberry Pi 3 B v1.2**. Para ello se integraron dos servicios en contenedores Docker **InfluxDB** [28] como base de datos de series de tiempo y **Grafana** [29] como interfaz de exploración y visualización. La solución se implementó como una extensión del *stack* existente de ChirpStack en `~/chirpstack-docker`, manteniendo una administración unificada mediante **Docker Compose**.

En términos de operación, InfluxDB expone una API HTTP en el puerto **8086** para la ingesta y consulta de series temporales, mientras que Grafana publica su interfaz web en el puerto **3000**. Al ejecutarse ambos servicios dentro de la misma red virtual de Docker, la comunicación entre Grafana e InfluxDB se realiza utilizando el nombre del servicio (`influxdb`) como *hostname*, evitando dependencias de `localhost` o de direcciones IP cambiantes. Una vez inicializados los contenedores, se accede desde un navegador web a ambos servicios mediante la IP de la Raspberry Pi (misma red LAN/Wi-Fi), lo que permite configurar credenciales, crear la fuente de datos y, posteriormente, construir paneles de visualización a partir del *bucket* de mediciones definido.

La configuración inicial de InfluxDB se automatizó mediante variables de entorno (`DOCKER_INFLUXDB_INIT_*`), que crean un usuario administrador, una organización y un *bucket* por defecto. De manera análoga, Grafana se inicializa con credenciales por defecto y solicita el cambio de contraseña en el primer acceso. Finalmente, la integración entre ambos se completa agregando InfluxDB como *Data Source* en Grafana, seleccionando *Flux* como lenguaje de consulta y empleando el *token* de acceso definido en la inicialización. De este modo, el sistema queda preparado para generar *dashboards* locales y persistentes, manteniendo la reproducibilidad del despliegue al estar todo descrito en el archivo `docker-compose.yml`.

# HERRAMIENTA DE SELECCIÓN DE ARQUITECTURA

La implementación del sistema contempla cuatro variantes de hardware, cada una con compromisos distintos en consumo, capacidad de comunicación y requerimientos de instalación (por ejemplo, disponibilidad de radiación solar o factibilidad de conectividad celular). Para apoyar la selección de la arquitectura más adecuada según el sitio de despliegue y las restricciones de la aplicación, se desarrolló una herramienta interactiva de recomendación basada en **Streamlit**. La aplicación permite ingresar coordenadas geográficas (latitud/longitud), condiciones de sombreado, requisitos de comunicación y la tasa de muestreo deseada; a partir de estas entradas evalúa la factibilidad energética y de conectividad, y entrega un ranking de alternativas junto con métricas de validación y visualización geoespacial.

En este capítulo se describe el funcionamiento de la herramienta mediante diagramas de flujo y secuencia. El objetivo es explicitar la separación entre capa de datos, motores de cálculo y lógica de decisión.

## 6.1. Arquitectura general de la aplicación

La Fig. 6.1 resume el flujo de información de la herramienta. El usuario entrega entradas desde la barra lateral (ubicación, restricciones y parámetros de la aplicación). En paralelo, la herramienta utiliza una capa de datos local que incluye: (i) *rasters* de recurso solar, (ii) una base de estaciones/antenas (Subtel) para aproximar cobertura celular, y (iii) un registro de placas (*board registry*) con sus capacidades, restricciones y perfiles energéticos. Estas fuentes alimentan tres motores principales: un motor geoespacial, un motor físico-energético y un motor de decisión/puntaje. Finalmente, la salida corresponde a un conjunto de resultados que incluye un ranking, métricas de aprobación/rechazo y una visualización en mapa.

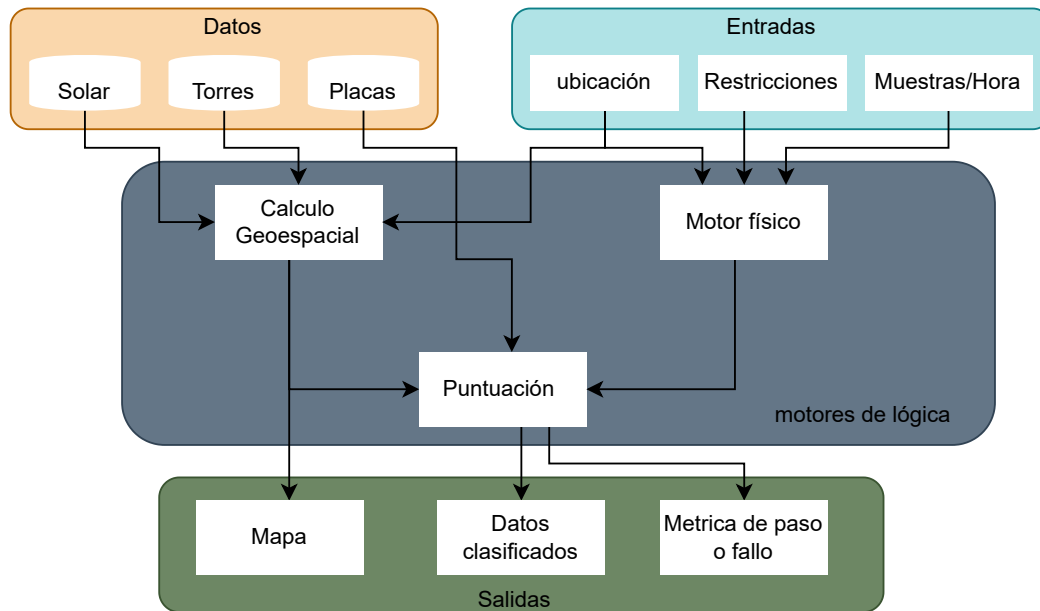


Figura 6.1: Diagrama de alto nivel de la herramienta de selección: entradas de usuario, capa de datos, motores de cálculo y salidas.

## 6.2. Capa de datos y registro de variantes

La herramienta se apoya en tres conjuntos de datos complementarios. En primer lugar, se utilizan *rasters* en formato `.asc` para obtener estimaciones del recurso solar (por ejemplo, GHI promedio) y, cuando se dispone, una referencia de temperatura ambiente. En segundo lugar, se emplea una base georreferenciada de infraestructura (archivos `.parquet/.geojson`) para aproximar la cercanía a torres celulares y, con ello, una indicación de factibilidad de enlaces móviles según un radio de búsqueda. En tercer lugar, se mantiene un **registro de placas** en formato JSON que define para cada variante: sensores integrados, opciones de comunicación soportadas, compatibilidad con modo solar o solo batería, y un perfil de potencia utilizado para estimar el consumo diario. Esta separación permite extender el sistema incorporando nuevas variantes sin modificar la lógica central: basta con añadir un nuevo registro con sus parámetros.

## 6.3. Modelo de producción solar

Para las variantes con panel solar, la herramienta estima la energía diaria disponible a partir del recurso solar del sitio (irradiación global horizontal promedio, **GHI**) y la potencia nominal del panel (**W<sub>p</sub>**). El modelo traduce estos insumos en una producción energética diaria mediante una cadena de factores multiplicativos que representan pérdidas típicas del sistema (*derates*), incluyendo sombreado, inclinación/orientación y pérdidas eléctricas (cableado y electrónica de acondicionamiento). Con ello se obtiene una estimación de *producción promedio* asociada al recurso anual medio del lugar.

Además, con el fin de evaluar robustez frente a la condición más exigente para la autonomía, se

calcula un *escenario conservador de invierno*. En este caso se aplica un factor geométrico asociado a la menor altura solar en el solsticio (variación estacional del ángulo de incidencia), junto con un factor adicional de penalización climática, de modo de aproximar una *disminución de flujo utilizable* respecto del promedio anual. Este enfoque es consistente con prácticas de dimensionamiento de sistemas fotovoltaicos autónomos, donde se verifica explícitamente el desempeño en el periodo crítico (invierno) para reducir la probabilidad de pérdida de carga. La Fig. 6.2 resume los bloques principales del cálculo y su objetivo de aproximar un “peor caso” operativo para la evaluación de autonomía [30], [31], [32].

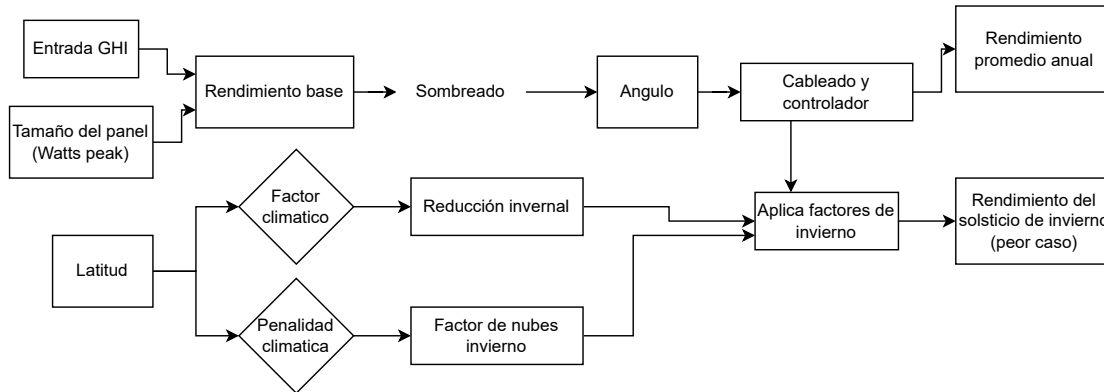


Figura 6.2: Modelo de producción solar utilizado por la herramienta. Se calcula una estimación promedio anual y una estimación conservadora asociada a condiciones de invierno.

## 6.4. Motor geoespacial para conectividad celular

Cuando el usuario exige conectividad celular (por ejemplo, NB-IoT o LTE-M), la herramienta evalúa si existe infraestructura cercana dentro de un radio de búsqueda. Para evitar tiempos de respuesta altos en zonas con alta densidad de puntos, se emplea un enfoque en dos etapas: primero se filtran candidatos mediante un índice espacial (*bounding box* e intersección rápida), y luego se verifica la distancia exacta solo sobre el subconjunto de candidatos. Este procedimiento reduce significativamente el costo computacional sin perder precisión en la verificación final. La Fig. 6.3 resume la secuencia de búsqueda y filtrado aplicada.

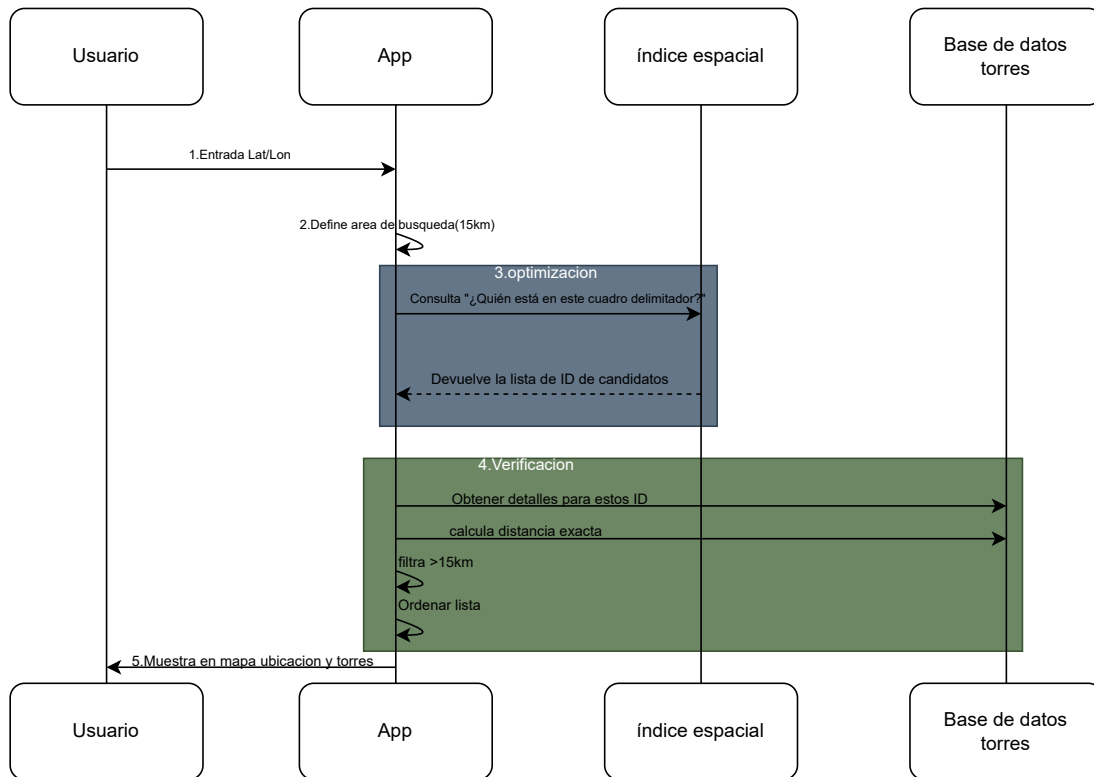


Figura 6.3: Secuencia de evaluación geoespacial: filtrado rápido con índice espacial y verificación precisa de distancia sobre un subconjunto de torres candidatas.

## 6.5. Lógica de factibilidad y cálculo de puntaje

La recomendación final se construye como una combinación de chequeos de compatibilidad y un puntaje relativo. Primero, la herramienta descarta variantes incompatibles con los requisitos de comunicación definidos por el usuario. Si se solicita conectividad celular, se valida además que exista infraestructura dentro del radio establecido. Superada la compatibilidad, se calcula el consumo diario esperado a partir del perfil energético de la variante y la tasa de muestreo seleccionada. Para modo "solo batería" se compara el consumo con un umbral de energía diaria; para modo solar se exige que la producción estimada supere la carga con márgenes de seguridad tanto en promedio anual como en el caso conservador de invierno. La Fig. 6.4 muestra el flujo de decisión y la forma general del puntaje: las alternativas factibles obtienen un puntaje base más un bono asociado al margen energético, mientras que las alternativas no factibles reciben un puntaje reducido o cero según el motivo de fallo.

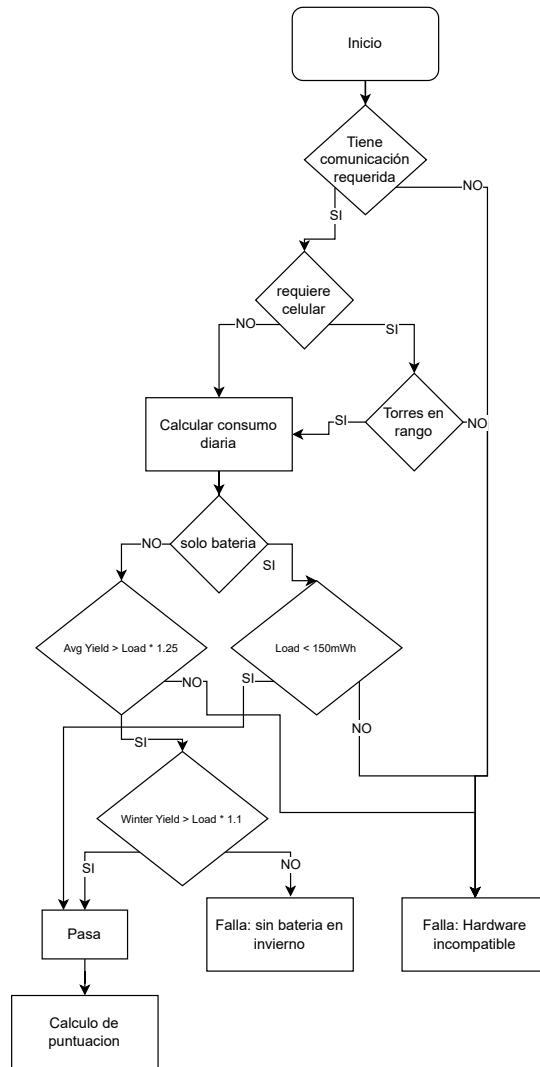


Figura 6.4: Flujo de evaluación: compatibilidad por comunicación, verificación de conectividad, balance energético y asignación de puntaje.

## 6.6. Interfaz de usuario y resultados

La herramienta entrega tres salidas principales, una clasificación de variantes (con métricas de consumo, producción estimada, modo de evaluación y causa de rechazo cuando corresponda), una visualización geográfica del sitio y su entorno de conectividad, y indicadores de aprobación/rechazo para facilitar la interpretación del resultado. En la versión utilizada para este trabajo, la recomendación se acompaña de un mapa interactivo con auto-zoom y de un resumen de la alternativa mejor clasificada. Las capturas de ejecución y pantallas de resultados se incluyen al final del capítulo en las Fig. 6.5–6.6-6.7.

**Site & Preferences**

Latitude (-56... -17, Chile) Longitude (-76...-66, Chile)

-33.4500 -70.6700

Shading

Partial shade

Samples per hour

2

**Communications**

Required links

LoRa x

Cellular

LTE-M

NB-IoT

WiFi

BLE

**Site & Preferences**

Latitude (-56... -17, Chile) Longitude (-76...-66, Chile)

-33.4500 -70.6700

Shading

Partial shade

Samples per hour

2

**Communications**

Required links

LoRa x

Battery-only (no solar)

Preferred operator (optional)

Entel / Movistar / Claro / WOM

(a) Entradas del sitio. (b) Mapa interactivo.

Figura 6.5: Ejecución de la herramienta de selección: entradas y visualización inicial.

Recommender Board Manager



## Weather Station Recommender — Chile

Recommend

Set your site and click [Recommend](#).

Figura 6.6: Inicio de selección.

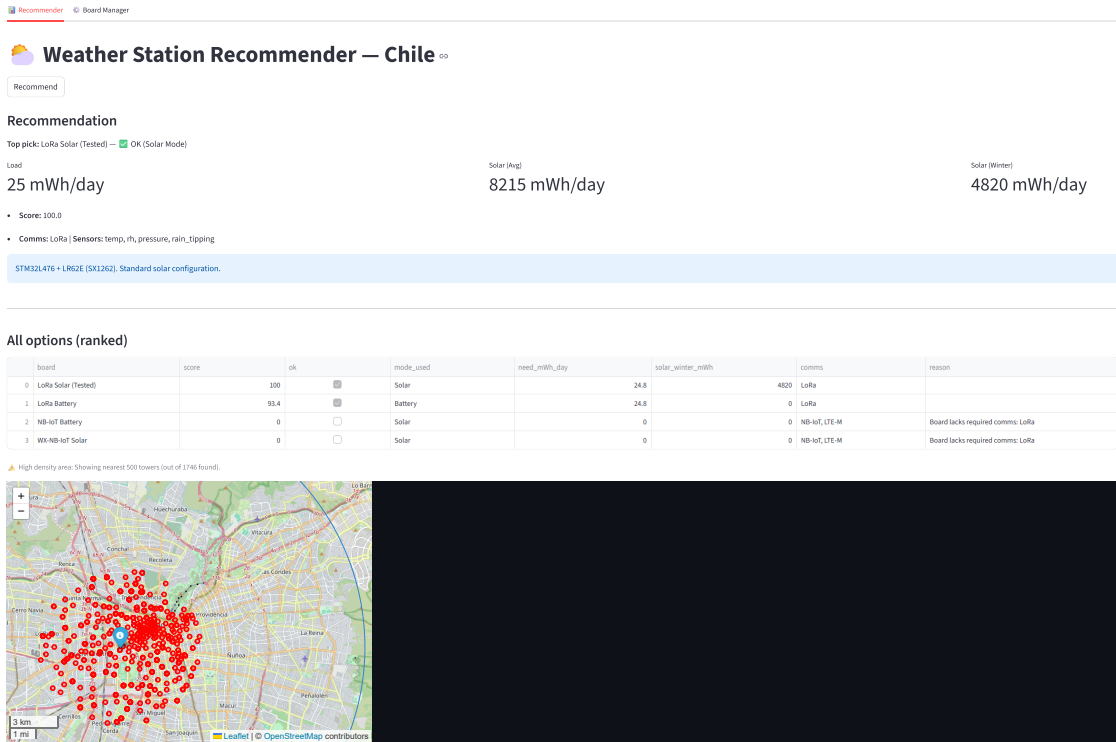


Figura 6.7: Ejecución de la herramienta de selección: Resultados.

La herramienta incorpora un modo de administración del *board registry* que permite extender el sistema sin modificar la lógica principal. Como se observa en la Fig. 6.8, el usuario puede ingresar una nueva variante mediante un formulario que define sus capacidades (sensores y enlaces de comunicación), compatibilidad con operación solar o solo batería y parámetros relevantes del perfil energético. Al guardar la información, la nueva placa queda disponible inmediatamente para el motor de evaluación y ranking, facilitando la mantención del catálogo de diseños y la incorporación incremental de futuras iteraciones de hardware.

Recommender Board Manager

### Board Manager

▼ Add new board design

Board name

Unique key ID

GitHub / docs link

Sensors (comma separated)  
temp,h,pressure

Camera  
Choose options

Solar-compatible

Battery-only OK

Max panel Wp  
3.00

Notes

#### Power profile

Sleep uA	2.00	Sample mA	2.00	Sample ms	60
TX mA	45.00	TX s	1.20	TX interval min	15
Overhead mW	0.80				

➤ Add board

Figura 6.8: Interfaz de gestión del registro de placas: formulario para añadir nuevas variantes.

# FABRICACIÓN Y MONTAJE

Este capítulo describe el proceso seguido para materializar el dispositivo físico correspondiente a la variante **LoRa + batería + aporte solar**. Esta configuración fue seleccionada para validación experimental debido a que satisface de mejor forma los requerimientos de autonomía y operación en terreno, y además se ajusta al tiempo disponible para la etapa de prototipado e integración. Se detalla la preparación de la lista de materiales (*BOM*), la orden de fabricación de la PCB y estencil, y el procedimiento de montaje y soldadura de componentes.

## 7.1. Lista de materiales (BOM)

A partir de los archivos de diseño se generó una lista de materiales en formato **CSV** que incluye los componentes, sus fabricantes y cantidades. Los elementos que no se encuentran directamente asociados al esquemático o al *layout* fueron incorporados manualmente, considerando su necesidad para completar el sistema físico e integrar la electrónica en gabinete. Entre estos se incluyen la caja seleccionada, los prensaestopas (*cable glands*), los conectores complementarios a los presentes en la PCB (por ejemplo, la contraparte hembra/macho según corresponda), y la antena del sistema de comunicaciones.

Con esta información se utilizó la plataforma Octopart y su generador de BOM para consolidar datos de abastecimiento y, en lo posible, reducir la cantidad de proveedores involucrados. Dado que, para esta primera iteración, se decidió adquirir componentes desde un proveedor diferente al contemplado inicialmente en el diseño (migrando desde Digi-Key hacia Mouser), fue necesario reemplazar ciertos componentes por equivalentes compatibles. En particular, se sustituyeron resistencias y condensadores por alternativas con especificaciones eléctricas equivalentes y encapsulado idéntico, y se actualizaron algunos conectores (incluyendo un conector de 20 pines y el portapila) para mantener compatibilidad mecánica y de huella (*footprint*) sin modificar la PCB.

El resultado fue una lista final con información de compra, incluyendo *Matched Part Number*, *Matched Manufacturer*, descripción, cantidad, oferta seleccionada, distribuidor, SKU, stock y precios unitarios y totales. Al tratarse de la primera versión, se adquirieron componentes mayoritariamente en cantidad suficiente para ensamblar **dos placas**, con el propósito de disponer de repuestos ante fallas de montaje, daño por manipulación o necesidad de pruebas adicionales durante la validación.

## 7.2. Orden de fabricación

**Alcance de fabricación de prototipos.** Si bien en este trabajo se definen *variantes* de arquitectura, estas corresponden a **diseños de PCB distintos** debido a diferencias en el subsistema energético (batería primaria/secundaria, presencia de *harvester* solar) y en el módulo de comunicaciones (LoRa vs. NB-IoT). No obstante, debido al tiempo disponible para desarrollo y validación, en esta etapa se **fabricó y montó únicamente** la variante **LoRa + batería secundaria + aporte solar** (Variante A), seleccionada por permitir caracterizar el comportamiento de consumo y

el flujo completo de adquisición/almacenamiento/transmisión. Los resultados obtenidos se utilizan como **referencia aproximada** para la versión LoRa sin aporte solar, bajo condiciones de operación comparables.

La fabricación de la PCB se encargó a partir de los archivos de manufactura (Gerbers) exportados desde KiCad. Para la producción de la placa se utilizó el servicio de **PCBWay**, mientras que la fabricación del estencil se solicitó a **JLCPCB** utilizando el mismo conjunto de archivos de diseño destinado a la apertura del estencil. Esta separación se adoptó por conveniencia de costo y disponibilidad. La placa fue diseñada considerando las capacidades estándar de ambos proveedores y evitando requerimientos de manufactura fuera de catálogo.

Las especificaciones solicitadas para la PCB fueron tamaño **100 mm × 50 mm**, **4 capas**, espesor **1,6 mm**, cobre final **1 oz**, terminación superficial **ENIG** (oro por inmersión), máscara de soldadura **verde** y serigrafía **blanca**, sobre sustrato **FR-4**. Adicionalmente, se configuraron vías cubiertas conforme a la opción estándar del proveedor. En la Fig. 7.1 se muestra un respaldo de la orden realizada.



Figura 7.1: Orden de fabricación de PCB.

**Observaciones del proveedor y ajustes previos a fabricación.** Durante la revisión de archivos por parte del proveedor se recibieron observaciones menores, las cuales fueron resueltas sin modificar la funcionalidad del diseño. En primer lugar, en la huella del **TPS22916CYFPR** se había incluido un elemento circular superpuesto sobre el pad correspondiente y conectado a la misma red, siguiendo la recomendación del fabricante. Sin embargo, el proveedor lo interpretó como una superposición redundante dentro del *land pattern* y solicitó su corrección; para evitar ambigüedades en la fabricación, se unificaron dichas geometrías en un único pad continuo.

Adicionalmente, el texto de serigrafía con el nombre de la placa fue ubicado inicialmente en una capa incorrecta. Para asegurar su impresión y legibilidad en el ensamble final, se reubicó al *Front Silkscreen* antes de liberar los archivos definitivos a fabricación. Estas correcciones se consideran ajustes de preparación para manufactura y no alteran las decisiones eléctricas ni la arquitectura del sistema.

### 7.3. Montaje del prototipo

Una vez recibidos los componentes y la PCB, se realizó el montaje superficial siguiendo un proceso estándar de *stencil + reflow* para el lado superior y dosificación puntual para el lado inferior. Para el ensamblaje se emplearon las siguientes herramientas: estencil, pinzas de precisión, pasta de soldar, flux, estación de *pick and place* con dispensador de pasta y jeringa de vacío, horno de reflow, microscopio electrónico para inspección, y cinta Kapton para protección térmica. La Fig. 7.2 muestra una etapa del proceso de montaje durante la aplicación de pasta y posicionamiento de componentes.

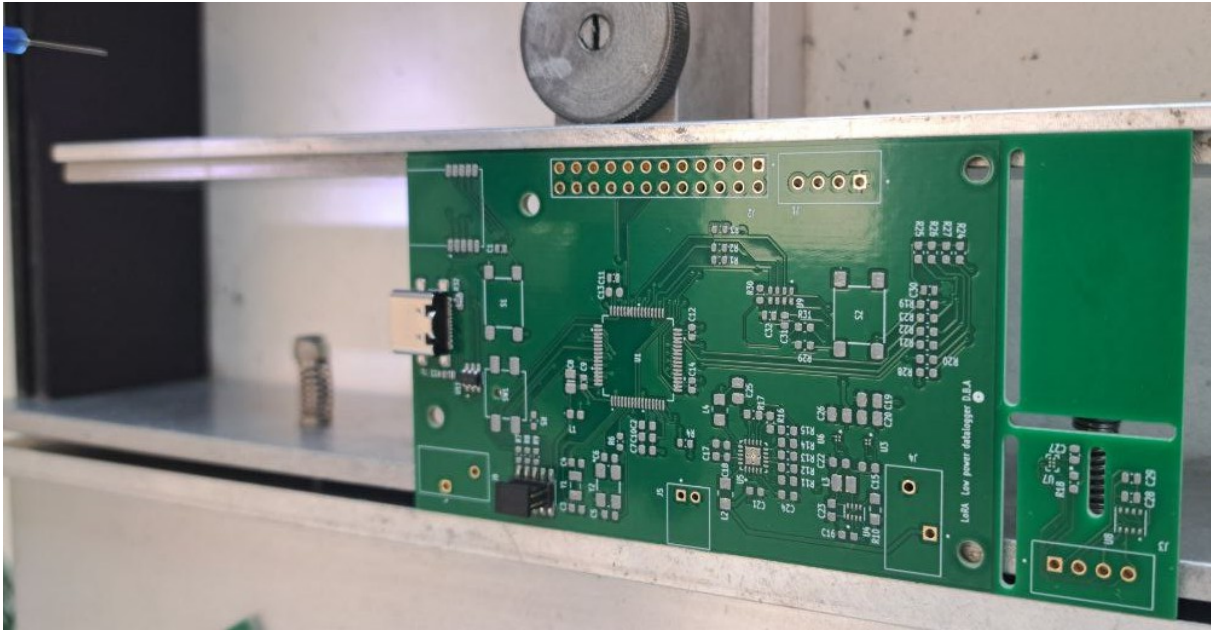
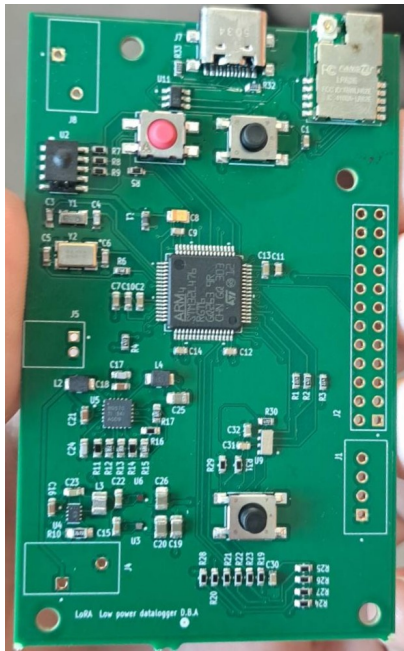


Figura 7.2: Montaje de componentes SMD: PCB fijada en el sistema de *pick and place* con pasta aplicada y componentes posicionados.

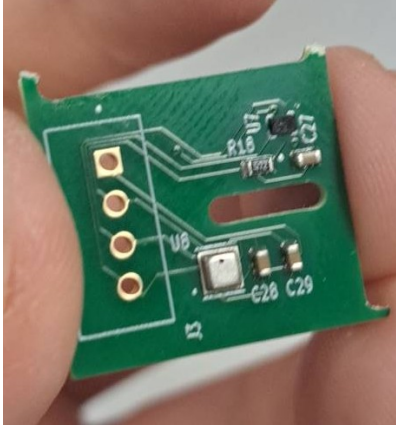
En primer lugar, se aplicó pasta de soldar sobre la cara superior utilizando el esténcil, complementando con flux para mejorar la humectación y reducir defectos en la etapa de reflow. Posteriormente, mediante la estación de *pick and place* se posicionaron los componentes SMD del lado superior. Dado que los encapsulados utilizados presentan temperaturas de soldadura compatibles, se empleó una curva de reflow estándar del horno, acorde al tipo de pasta seleccionada.

Tras el reflow, se realizó una inspección visual detallada con microscopio para verificar la calidad de las uniones visibles, identificar puentes de soldadura, componentes desalineados o insuficiencia de pasta y, cuando fue necesario, corregir manualmente defectos menores. Para soldar los componentes de la cara inferior se protegió la cara superior con cinta Kapton, reduciendo el riesgo de que el calentamiento refluya nuevamente los componentes ya montados y provoque desprendimientos.

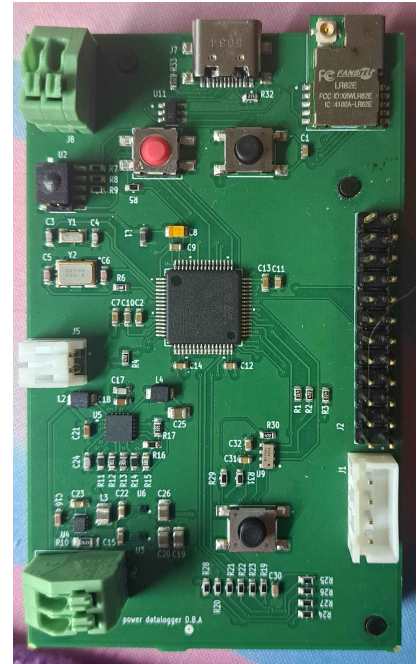
Debido a que no se utilizó el esténcil para la cara inferior en esta iteración, la pasta de soldar se aplicó mediante el dispensador de la estación de montaje y se posicionaron los componentes restantes de forma manual/asistida. Una vez completadas ambas caras y soldado el conjunto SMD, se procedió al montaje de componentes *through-hole* o conectores plásticos (conectores de sensores y batería) utilizando estación de soldadura, asegurando la correcta humectación y resistencia mecánica de las uniones en conectores sometidos a esfuerzos de inserción y extracción. El resultado del ensamble se resume en la Fig. 7.3.



(a) Placa principal (SMD).



(b) Sub-placa sensores (SMD).



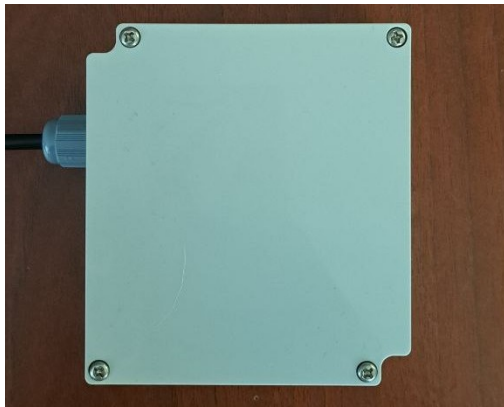
(c) Placa principal (final).

Figura 7.3: Resultados de montaje: (a) placa principal tras completar ensamble SMD, (b) sub-placa *breakaway* de sensores tras reflow, y (c) placa principal finalizada con conectores montados mediante soldadura manual.

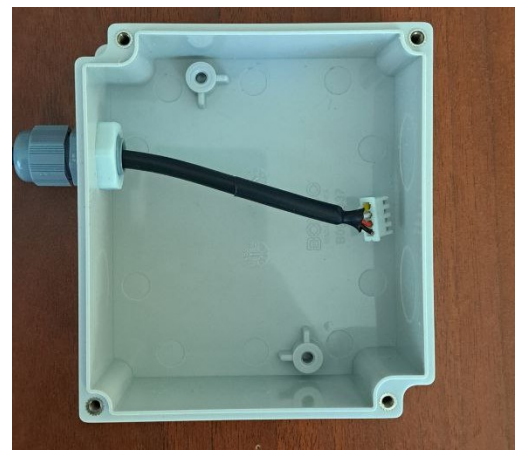
**Nota sobre costos de fabricación y BOM.** Para el detalle del *Bill of Materials* (BOM), los supuestos de costeo y la estimación del costo total para un lote de **10 unidades** (incluyendo fabricación de PCBs, batería y elementos externos como pluviómetro/cableado), véase el Anexo D.

# CARCASA, PRENSAESTOPAS Y MONTURAS MECÁNICAS

Para permitir la operación del sistema a la intemperie se requirió una solución mecánica que protegiera la electrónica frente a lluvia y polvo, y que además fuese de bajo costo y fácil de adquirir. En este trabajo se seleccionó una carcasa comercial **PNB-30417-KO**, con grado de protección **IP67**, lo que permite el uso en exterior siempre que se asegure un correcto sellado de sus puntos de entrada de cables. Sin embargo, debido a que la carcasa está fabricada en **ABS8.1**, material sensible a degradación por radiación UV, se recomienda evitar la exposición directa y prolongada a luz solar o, alternativamente, aplicar un recubrimiento protector para mejorar su durabilidad.



(a) caja cerrada.



(b) Caja abierta.

Figura 8.1: Carcasa PNB-30417-KO con puntos de salida de cables mediante prensaestopas.

## 8.1. Salida de cables y conectividad hacia sensores externos

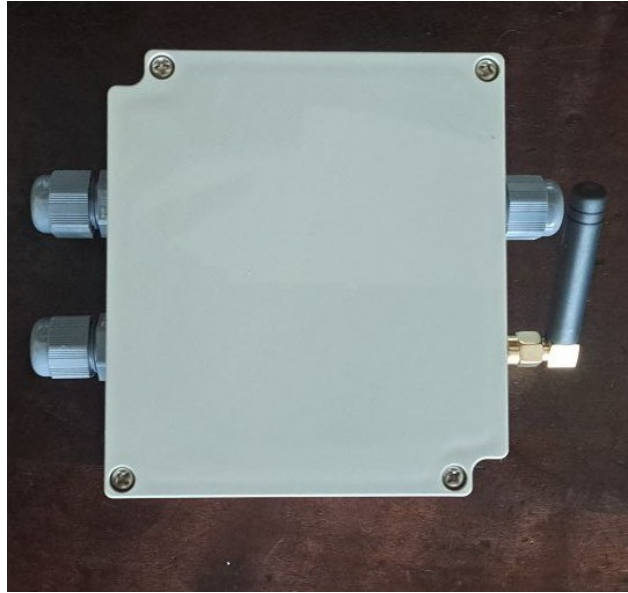


Figura 8.2: Carcasa PNB-30417-KO con puntos de salida de cables mediante prensaestopas y antena.

Para la extracción de cables hacia sensores externos se utilizaron prensaestopas **1SNG601011R0000**, que permiten mantener el sellado del gabinete y entregar alivio de tensión al arnés. Se emplearon conductores con aislamiento resistente a intemperie y radiación, utilizando cable con protección UV de calibre **AWG22**. Esta selección facilita el montaje en terreno, reduce el riesgo de ingreso de humedad y mejora la robustez frente a tracción o vibraciones.

La salida de RF se realiza a través del cable **CAB602RF-0200-A-1**, cuyo conector SMA *bulkhead* con junta tórica mantiene el grado IP67 en la pared del gabinete. No obstante, la antena **ANT-20090EB56** conectada en el exterior no posee clasificación IP, por lo que la unión SMA expuesta constituye un punto susceptible al ingreso de humedad. Para preservar la protección del conjunto se plantean dos alternativas, reemplazar la antena por un modelo con sellado IP67, como la **AANE-WH-0033-1** de Abracon, o sellar la unión mediante cinta autovulcanizante (*self-amalgamating tape*) tipo **TSSA-1.5-.76-15-1**. La segunda opción resulta más económica, en el prototipo actual esta protección no fue implementada, identificándose como una mejora necesaria para despliegues en condiciones reales de intemperie.

## 8.2. Fabricación de arneses y terminaciones

La fabricación de los arneses se realizó utilizando una herramienta de crimpado compatible con los terminales JST empleados en el diseño, con el propósito de asegurar una terminación eléctrica y mecánica consistente. El procedimiento de armado incluyó corte a medida, pelado controlado, crimpado de terminales y verificación de retención y continuidad antes de la instalación en gabinete. Un ejemplo de arnés ensamblado se muestra en la Fig. 8.3.

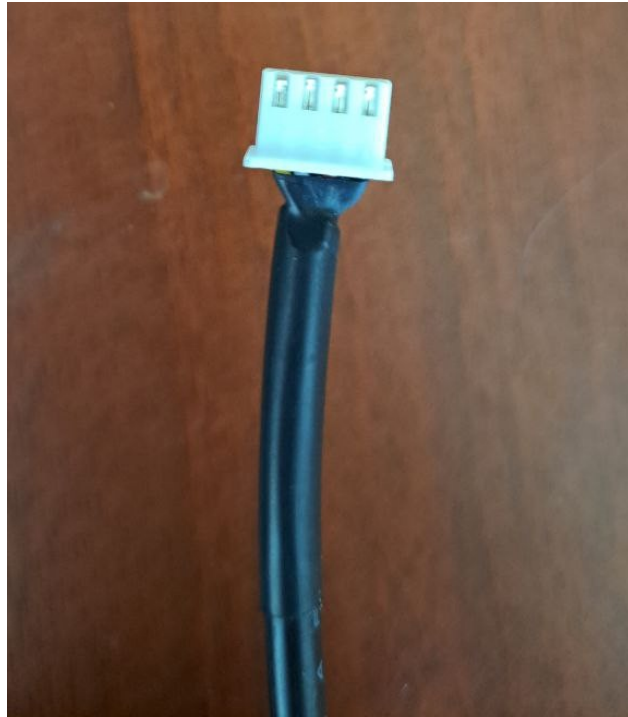


Figura 8.3: Arnés ensamblado para interconexión entre la placa principal y periféricos externos.

### 8.3. Stevenson Screen para sensores ambientales

Para mejorar la calidad de medición de temperatura y humedad, y reducir la incidencia directa de radiación solar y lluvia sobre el sensor, se incorporó un *radiation shield* tipo **Stevenson Screen**. En el prototipo se optó por una solución impresa en 3D basada en un modelo abierto disponible en Thingiverse [33], distribuido bajo licencia *Creative Commons Attribution–NonCommercial–ShareAlike*. Este enfoque permitió acelerar la integración mecánica y ajustar dimensiones al volumen del sensor y el arnés.

En la literatura se recomienda utilizar materiales con buena resistencia UV para escudos expuestos a exterior como ASA [34]. No obstante, debido a disponibilidad y por tratarse de un prototipo, se fabricó el escudo en resina y se aplicó un recubrimiento con pintura de protección UV, buscando mejorar su estabilidad superficial frente a exposición solar. La Fig. 8.4 muestra el escudo impreso y tratado.



Figura 8.4: Stevenson Screen impreso en 3D para el prototipo, con recubrimiento de protección UV.

## 8.4. Montura interna para PCB

Para asegurar una integración mecánica reproducible dentro del gabinete, se diseñó una montura interna impresa en 3D. Para ello se utilizó el modelo 3D del *enclosure* y la geometría de la PCB, generando una estructura que aprovecha los soportes internos del gabinete para posicionarse correctamente. La fijación de la PCB se realiza mediante espaciadores que terminan dimension **M2.5**, permitiendo asegurar la tarjeta sin modificar el gabinete y manteniendo tolerancias consistentes entre ensamblajes. La Fig. 8.5 presenta la montura desarrollada.

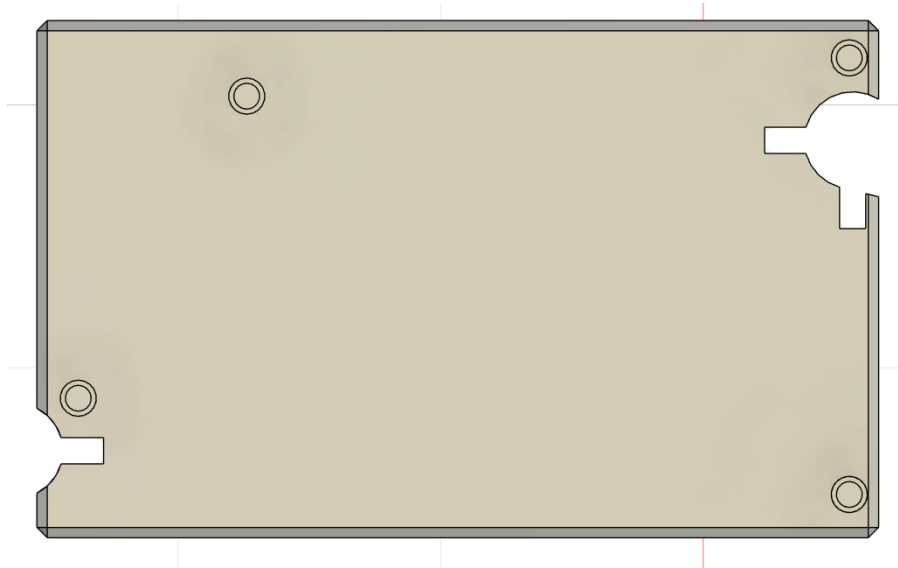
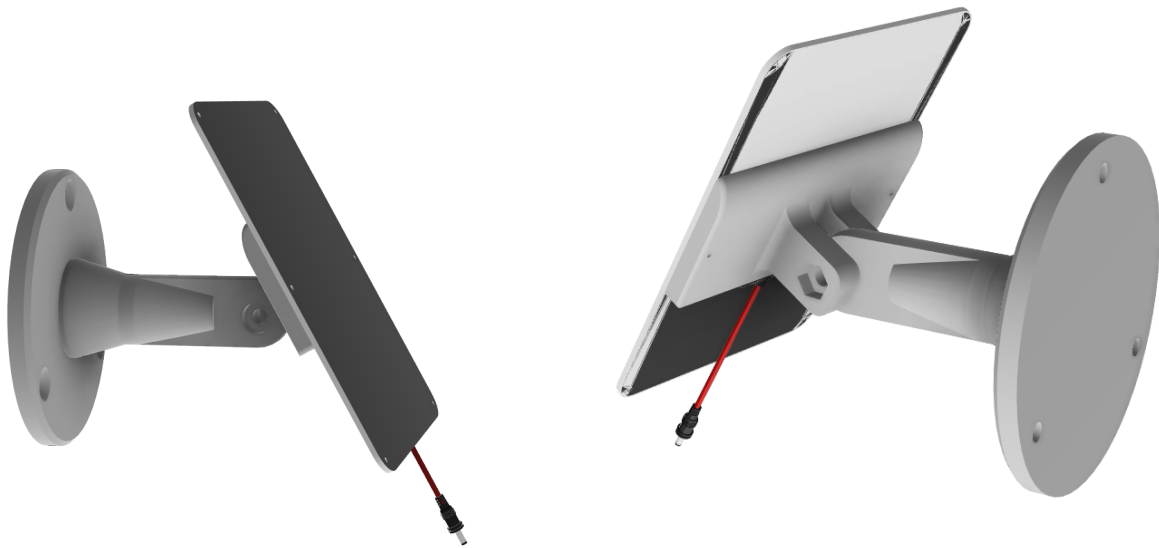


Figura 8.5: Montura interna impresa en 3D para fijación de la PCB dentro del gabinete (M2.5).

## 8.5. Montura para panel solar

Para el panel solar se diseñó una base de montaje simple utilizando como punto de partida el diseño abierto “*Solar panel holder*” de *xpari* [35], y el modelo 3D provisto por el fabricante del panel. La geometría final se ajustó para aprovechar los puntos de sujeción del panel **P105 Voltaic Systems 5.5W 6V Solar Panel (ETFE)**, de modo de asegurar una fijación mecánica reproducible sin intervenir el panel ni requerir herrajes especiales. Esta montura permite instalar el panel sobre una estructura o superficie plana, entregando rigidez suficiente para uso en terreno y manteniendo la posibilidad de desmontaje y reemplazo. La Fig. 8.6 presenta el modelo 3D del soporte desarrollado junto con el modelo 3D del panel.



(a) Modelo 3D de la montura propuesta frente.

(b) Modelo 3D de la montura propuesta parte posterior.

Figura 8.6: Montura para panel solar y referencia geométrica del panel utilizado (P105, 6V-5.5W, ETFE).

# PRUEBAS EXPERIMENTALES

Este capítulo presenta la validación experimental del sistema, abarcando pruebas de **hardware**, **software** y **comunicación**. En primer lugar se verifican los subsistemas eléctricos y de montaje mediante ensayos de encendido, medición de rieles y detección de fallas, incluyendo las correcciones implementadas sobre el prototipo. Posteriormente se evalúa el firmware base, considerando programación, inicialización de periféricos y funciones de adquisición y registro. Finalmente, se comprueba el funcionamiento del enlace de comunicación correspondiente a la variante bajo prueba, analizando el comportamiento de transmisión, estabilidad del enlace y consistencia de los datos recibidos respecto de lo almacenado localmente.

## 9.1. Verificación de hardware y corrección de fallas

Con el fin de iniciar la validación del sistema, la placa se conectó inicialmente de forma directa a una batería. Sin embargo, durante este primer encendido se detectó un comportamiento anómalo, por lo que la prueba se suspendió para evitar daños adicionales y se procedió a un análisis sistemático del diseño y del montaje. A partir de la revisión de los esquemáticos se identificaron errores de diseño que podían explicar un funcionamiento errático del subsistema de potencia. En particular, los integrados **TPS22916CYFPR** y **TPS62840DLCR** no contaban con una resistencia *pull-down* externa en sus pines de configuración y control, dependiendo únicamente del estado interno del microcontrolador. En estas condiciones, dichos pines podían quedar en estado flotante durante el arranque o en transitorios, provocando cambios irregulares de modo de operación. Este hallazgo motivó la incorporación de *pull-downs* de acuerdo con las recomendaciones de hoja de datos y buenas prácticas de robustez en encendido.

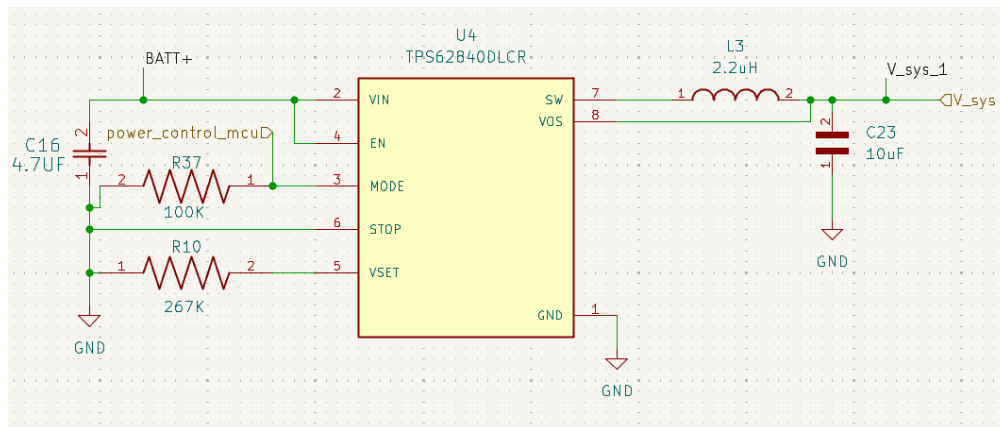


Figura 9.1: Corrección de esquemático: adición de resistencia *pull-down* en el pin de configuración/-control del TPS62840DLCR para evitar estados flotantes.

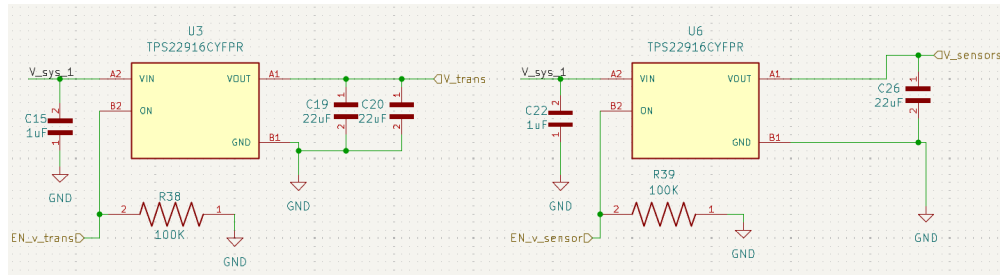


Figura 9.2: Corrección de esquemático: adición de resistencias *pull-down* en pines de control del TPS22916CYFPR para definir el estado durante arranque y transitorios.

En paralelo, se detectó un error en el conector del sensor de lluvia: el pin destinado a alimentación había sido conectado incorrectamente a tierra junto con un pin del microcontrolador. Para corregirlo sin rediseñar la PCB en esta iteración, se implementó una modificación externa de cableado que permitió entregar la alimentación al sensor desde el arnés y, además, se añadió un *pull-down* posterior al sensor en el cableado hacia el pin correspondiente del MCU, asegurando un estado definido cuando el contacto se encuentra abierto. El punto de interconexión y el mapeo de señales considerados para esta corrección se muestran en la Fig. 9.3.

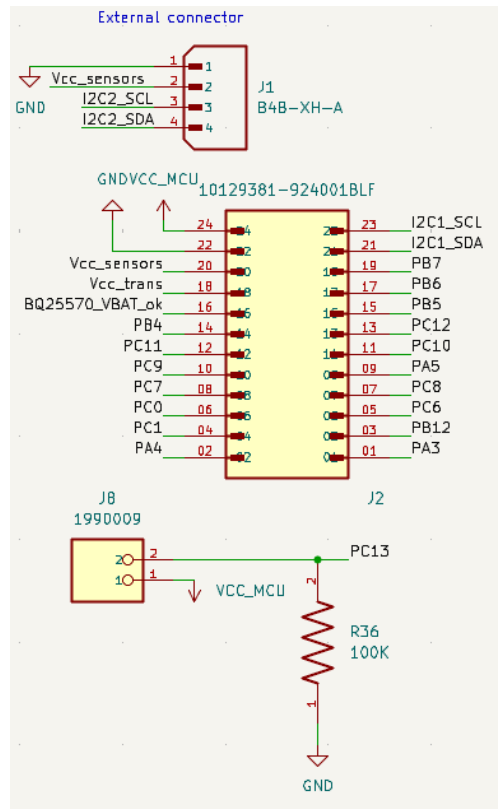


Figura 9.3: Corrección de esquemático: conector de lluvia y conectores externos, indicando la reasignación necesaria para evitar conexión errónea a tierra.

Durante la inspección física del ensamble se observó además un error crítico de montaje: el microcontrolador había sido colocado rotado en 180 grados debido a una interpretación incorrecta

de las marcas de orientación del encapsulado. Para abordar el problema se desoldó el MCU y, antes de reinstalarlo, se decidió verificar el funcionamiento del convertidor **TPS62840DLCR** de manera aislada. Esta verificación se realizó utilizando una fuente de laboratorio con corriente limitada y protección, lo que permitió caracterizar la respuesta sin exponer el resto del sistema a sobrecorrientes.

Al medir un voltaje de salida anormalmente bajo para una configuración de salida fija se consideró la posibilidad de un cortocircuito en algún punto del PCB. Para localizarlo se utilizó una cámara térmica FLIR, buscando puntos de calentamiento anómalos bajo excitación con corriente limitada. Como resultado, se determinó que uno de los **TPS22916CYFPR** presentaba un cortocircuito entre pines, el cual no era evidente visualmente ni fácilmente medible debido al tamaño del encapsulado y la naturaleza del puente de soldadura. Se procedió a desoldar el componente y re-soldarlo para descartar defectos de montaje y evaluar nuevamente el estado del riel.

Tras esta intervención se confirmó que ya no existía un cortocircuito dominante en la placa; sin embargo, el convertidor continuó sin operar como se esperaba. Por ello se realizó un análisis detallado del esquemático y de la hoja de datos del **TPS62840DLCR** y se decidió ensamblar el circuito de conversión de manera aislada sobre una placa vacía, replicando los pasivos y conexiones mínimas recomendadas. Este ensayo permitió identificar un problema de orientación del integrado: las marcas externas del encapsulado resultaban visualmente simétricas y el dispositivo había sido montado en 180 grados, diferencia que se corroboró al verificar la disposición real de uno de los pads de referencia. Una vez corregida la orientación, y al aplicar carga controlada al riel, se observó el voltaje objetivo de **3,3 V**.

Adicionalmente, se corrigió un error de conexión en el *switch* de BOOT, que había quedado desconectado de la alimentación. En esta versión del prototipo se solucionó mediante un cable externo para asegurar la referencia correcta. En la misma etapa se agregó el *pull-down* faltante asociado al **TPS62840DLCR**, con el propósito de asegurar un estado definido de los pines de configuración durante el arranque.

Finalmente, se reinstaló el microcontrolador en la orientación correcta y se verificó la comunicación con la placa mediante un **ST-Link V2**. Con ello fue posible programar y ejecutar un firmware de prueba, confirmando que el camino de alimentación principal y la interfaz de depuración se encontraban operativos tras las correcciones implementadas. El estado del prototipo luego de las modificaciones de cableado y la incorporación de componentes adicionales se presenta en la Fig. 9.4.

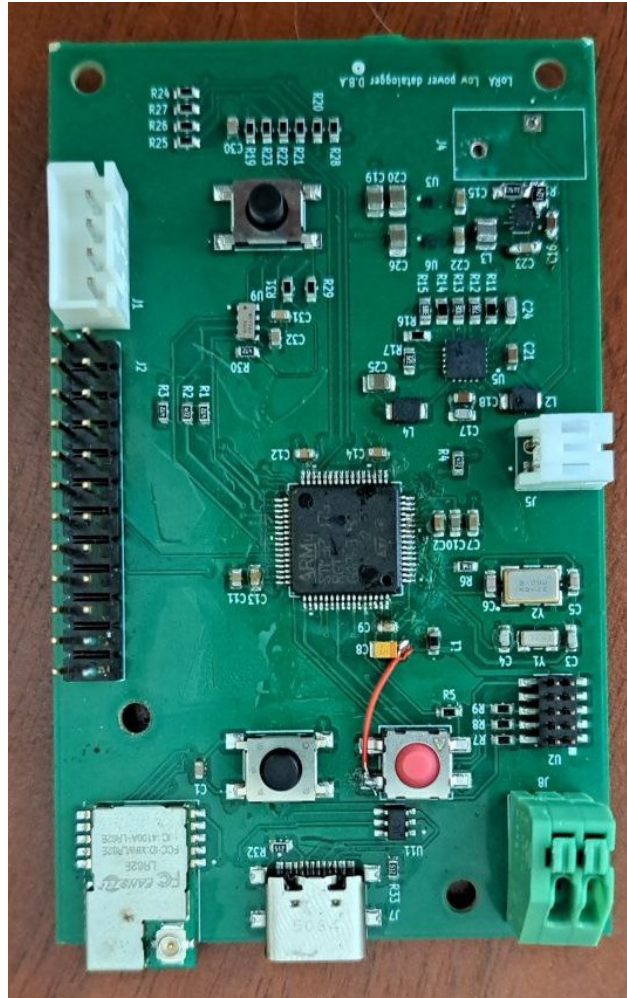


Figura 9.4: Prototipo con correcciones aplicadas: conexiones externas de reparación y componentes añadidos para estabilizar el arranque y el subsistema de potencia.

### 9.1.1. Revisión del panel solar: sustitución y *clamping* de tensión

Durante la revisión del subsistema de alimentación asociado a las variantes con recolección de energía, se identificó una incompatibilidad crítica entre el panel solar originalmente seleccionado (Voltaic Systems **P105**) y el circuito integrado de gestión de energía **BQ25570**. El panel P105 presenta un voltaje de circuito abierto  $V_{OC}$  nominal del orden de 7.7V, mientras que la hoja de datos del BQ25570 establece un voltaje máximo absoluto de 5.5V en el pin de entrada  $V_{IN\_DC}$ . En consecuencia, una conexión directa del P105 al BQ25570 expone al dispositivo a una sobretensión fuera de especificación, con alta probabilidad de falla permanente del PMIC durante condiciones de alta irradiancia o bajo carga ligera.

**Acción correctiva: sustitución del panel solar.** Para asegurar la integridad del hardware y mantener la autonomía energética del sistema, se reemplazó el panel P105 por un panel **Power-Film MPT3.6-150**, cuyo voltaje nominal es de 3.6V y cuyo  $V_{OC}$  se encuentra en torno al límite superior admisible del BQ25570 (5.5V). Si bien este panel tiene menor tamaño y potencia nominal

(aprox. 360mW), su capacidad resulta suficiente para cubrir el consumo diario estimado del dispositivo ( $\sim 100\text{mWh}/\text{día}$ ), manteniendo margen incluso bajo escenarios desfavorables (cielo nublado o sombreado parcial), donde la energía disponible aún se proyecta superior a la demanda del sistema.

**Acción correctiva: protección por *clamping* en la entrada.** Adicionalmente, dado que el  $V_{OC}$  del panel MPT3.6-150 se sitúa cercano al máximo absoluto del BQ25570, se consideró el efecto de temperatura: en paneles fotovoltaicos el voltaje de circuito abierto tiende a incrementarse a bajas temperaturas. Con el fin de robustecer el diseño frente a variaciones climáticas y transitorios, se incorporó un limitador de tensión (*voltage clamp*) mediante un diodo Zener **1SMA4733A** (5.1V, 1W) conectado en paralelo a la entrada del panel, en polarización inversa entre  $V_{IN}$  y GND. Este componente deriva corriente hacia tierra cuando la tensión excede el umbral, evitando que el pin  $V_{IN\_DC}$  del BQ25570 supere niveles críticos y mejorando la tolerancia del prototipo ante condiciones de operación extremas.

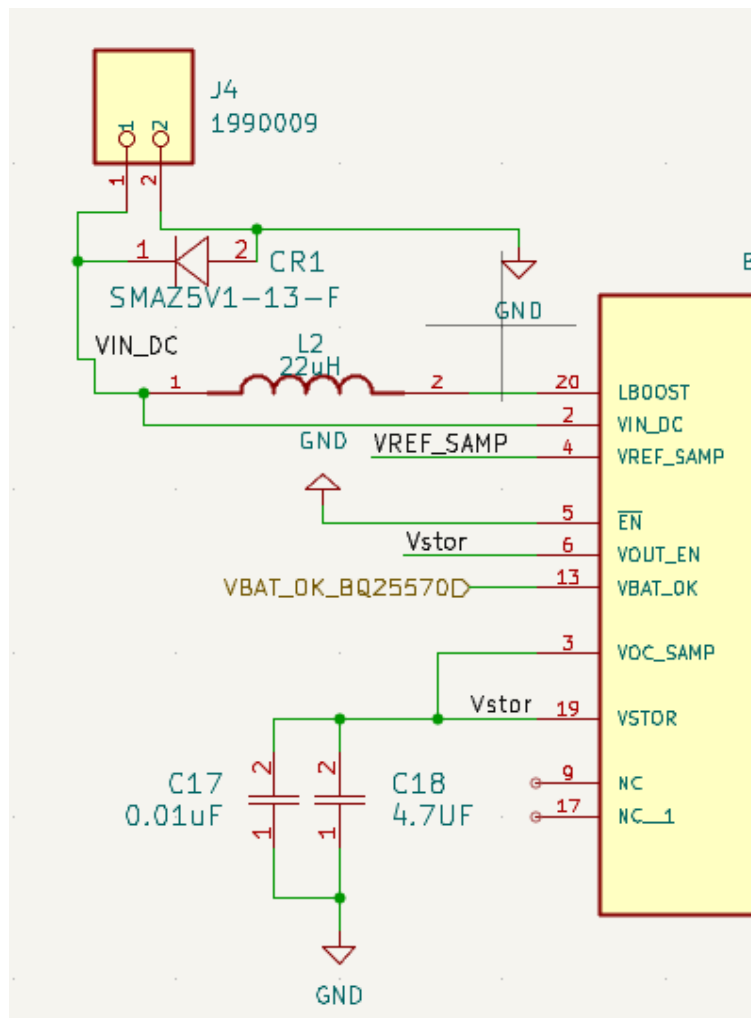


Figura 9.5: Protección en la entrada del panel: diodo Zener en paralelo para limitar  $V_{IN}$  y proteger el BQ25570.

Como resultado, se generó **versiones corregidas** de los diseños. Los archivos actualizados (es-

quemáticos y PCB) para las **cuatro variantes** se encuentran disponibles en el repositorio del proyecto: <https://github.com/DABerrios/data-logger-de-bajo-consumo-con-comunicaci-n-LPWAN.git>.

## 9.2. Verificación de Firmware

La verificación del firmware se inició evaluando el comportamiento del subsistema de alimentación con el *load switch* **TPS22916CYFPR** en el dominio `V_SENSOR`. Inicialmente, aun con la señal de habilitación en estado bajo, se observó un voltaje residual en la salida del riel (superior a 0 V esperado, pero inferior a 3,3 V), asociado a *back-feeding* a través de líneas digitales (principalmente SPI e I<sup>2</sup>C). Este efecto se eliminó desactivando explícitamente los periféricos **SPI** e **I<sup>2</sup>C**, y aplicando una estrategia de *pin parking* (configuración de pines en modo **analógico**) durante suspensión, reduciendo fugas y evitando alimentación parasitaria. Con esta medida fue posible medir **0 V** en la salida `V_SENSOR` cuando el riel se encontraba apagado.

Posteriormente, se verificó la secuencia completa de apagado y reactivación mediante *Stop Mode 2* y fuentes de interrupción, comprobando que el sistema puede despertar, re-inicializar periféricos, leer sensores y almacenar datos en la tarjeta microSD. Tal como se describe en la Sección 4.2, al retornar desde suspensión profunda se incorporó una secuencia de **reinicio y re-inicialización** (reset de bloques RCC, recuperación del bus y reconfiguración) para asegurar que las comunicaciones SPI/I<sup>2</sup>C y la operación de escritura en FatFS retomen en un estado determinístico, evitando fallas por pérdida de configuración o *bus lock*.

Durante estas pruebas también se observó que el RTC **RV-3028** puede quedar en un estado no operativo tras pérdidas transitorias de comunicación; para resolverlo, se implementó una rutina de recuperación del bus I<sup>2</sup>C que genera pulsos de reloj y fuerza una condición *STOP*, permitiendo que el dispositivo libere la línea y se restablezca la comunicación.

Como resultado, se confirmó el funcionamiento correcto de la interrupción periódica del RTC configurada cada **60 s**, la reactivación desde *Stop Mode 2*, y la escritura consistente de registros en la tarjeta microSD. La Fig. 9.6 muestra un ejemplo del archivo `DATA.csv` generado y abierto en un computador para su inspección.

```

0,24.19,46.29,1014.28,11:00:06,23/01/2026
0,24.20,45.98,1014.40,11:01:08,23/01/2026
0,24.25,46.80,1014.05,11:02:09,23/01/2026
0,24.27,46.14,1013.95,11:03:06,23/01/2026
0,24.20,46.04,1014.03,11:04:07,23/01/2026
0,24.18,46.33,1013.86,11:05:09,23/01/2026
0,24.15,45.86,1013.77,11:06:07,23/01/2026
0,24.16,45.67,1013.77,11:07:08,23/01/2026
0,24.25,45.77,1013.67,11:08:09,23/01/2026
0,24.22,45.48,1013.70,11:09:05,23/01/2026
0,24.23,45.89,1013.66,11:10:10,23/01/2026
0,24.27,45.75,1013.66,11:11:09,23/01/2026
0,24.18,45.19,1013.68,11:12:08,23/01/2026
0,24.12,45.56,1013.61,11:13:07,23/01/2026
0,24.09,45.56,1013.49,11:14:09,23/01/2026
0,24.19,46.85,1013.42,11:15:07,23/01/2026
0,24.24,45.90,1013.25,11:16:09,23/01/2026
0,24.11,45.91,1013.18,11:17:06,23/01/2026
0,24.16,45.67,1013.02,11:18:08,23/01/2026
0,24.20,45.61,1013.00,11:19:07,23/01/2026
0,24.48,45.74,1012.63,11:20:08,23/01/2026

```

Figura 9.6: Archivo `DATA.csv` almacenado en la microSD, abierto en computador para verificación de escritura.

Con el subsistema de adquisición y registro validado, se procedió a verificar el funcionamiento del módulo de comunicaciones tras integrar la pila LoRaMAC. Para ello se intentó transmitir utilizando el *gateway* Dragino LPS8N configurado en **AU915\_0**. En una primera etapa se observaron dificultades para completar la comunicación de extremo a extremo, por lo que se decidió aislar la causa realizando una prueba independiente del transceptor, se cargó un firmware mínimo basado únicamente en funciones de radio (sin LoRaWAN), con el objetivo de confirmar la operación física del LR62E (SPI, señales de control y RF). Esta prueba permitió verificar la transmisión correcta del módulo, confirmando que el hardware y la interfaz de radio eran funcionales.

Con este respaldo, la depuración posterior se enfocó en la configuración del stack LoRaWAN. Para establecer la comunicación, fue necesario adaptar el código para seguir el comportamiento específico ejemplificado por el fabricante Fanstel, el cual presenta diferencias significativas respecto a la operación de otros módulos LoRa estándar. Esta modificación permitió que el dispositivo iniciara la conexión a la red mediante el método OTAA (Over-The-Air Activation).

Sin embargo, a pesar de múltiples investigaciones y pruebas, el nodo no logró recepcionar el paquete de bajada (downlink) correspondiente al mensaje Join Accept, aun cuando se verificó que el gateway realizaba el envío correctamente. Debido a esta limitación en la recepción de datos, se optó por cambiar el método de activación a ABP (Activation By Personalization), eliminando así la dependencia de los mensajes de bajada para operar en la red.

Considerando los inconvenientes derivados del funcionamiento particular del hardware de Fanstel, se recomienda para el desarrollo de versiones posteriores reemplazar el transeptor actual por un módulo LoRa de arquitectura más genérica o estandarizada, reduciendo los problemas de compatibilidad e integración.

### 9.3. Pruebas de consumo

Una vez completadas las pruebas funcionales del firmware (despertar periódico, lectura de sensores y escritura en microSD), se caracterizó el consumo del sistema en los distintos estados de operación. Previamente a las mediciones definitivas, durante la etapa de verificación se identificaron condiciones de hardware y de configuración que afectaban tanto el consumo como la seguridad de operación, por lo que fue necesario corregirlas para obtener resultados representativos.

En primer lugar, se desconectó la señal **VBAT\_OK** del **BQ25570**, ya que, a diferencia de otros módulos, esta salida no requiere *pull-up* externo y puede entregar un nivel lógico asociado al umbral configurado de batería (del orden de 4.17V en la configuración utilizada), lo que podría exceder el rango permitido en un GPIO del microcontrolador. Adicionalmente, en el **RV-3028** se detectó que la línea de interrupción contaba con un *pull-up* duplicado: se había añadido erróneamente un *pull-up* adicional en la placa, pese a existir ya uno asociado al dominio de batería/respaldo, lo que podía modificar niveles en escenarios de conmutación de alimentación y aumentar fugas.

Durante las pruebas también se observaron discrepancias de consumo debidas a la configuración de depuración, al no utilizar **STM32CubeProgrammer** [36] para la carga de firmware, el sistema quedaba con el modo de *debug* habilitado, manteniendo activos recursos internos del MCU incluso sin el *ST-LINK* conectado. Esta condición elevaba artificialmente el consumo en reposo. Al programar y ejecutar el binario sin depuración, se obtuvieron mediciones consistentes con el objetivo de bajo consumo.

En una primera aproximación se monitoreó el consumo directamente desde la fuente de alimentación de laboratorio durante la ejecución del ciclo nominal. Este análisis inicial permitió detectar una condición anómala en el dominio de comunicaciones: el interruptor de carga asociado a la alimentación del **LR62E** presentaba un puente no deseado entre entrada y salida, impidiendo el corte efectivo del riel y elevando el consumo en reposo. Tras remover el puente, se repitieron las mediciones y se confirmó una reducción significativa del consumo durante suspensión profunda, alcanzando valores por debajo del umbral de lectura confiable de la fuente. Debido a esta limitación instrumental, se utilizó un **Joulescope** para registrar corriente y potencia con mayor resolución temporal, permitiendo observar transitorios de activación y cuantificar el gasto energético por evento.

La Fig. 9.7 muestra una medición representativa en *Stop Mode 2*, donde el microcontrolador permanece con periféricos desenergizados y pines *parked* para minimizar fugas. En estas condiciones se observó un consumo promedio del orden de **1.6 $\mu$ A**. La Fig. 9.8 presenta el consumo asociado a una interrupción externa del pluviómetro, correspondiente a un evento breve que incrementa el contador sin activar el ciclo completo, consumiendo en promedio **65 $\mu$ A** por **140ms**. Finalmente, la Fig. 9.9 muestra el evento periódico del RTC (cada **60 s**), en el cual el sistema energiza el riel de sensores, re-inicializa buses, adquiere mediciones y escribe en la microSD antes de retornar a suspensión. Para este ciclo se registró un consumo promedio aproximado de **10mA** y una duración de **313.51ms** entre el inicio del *wake-up* y el retorno a *Stop Mode 2*.

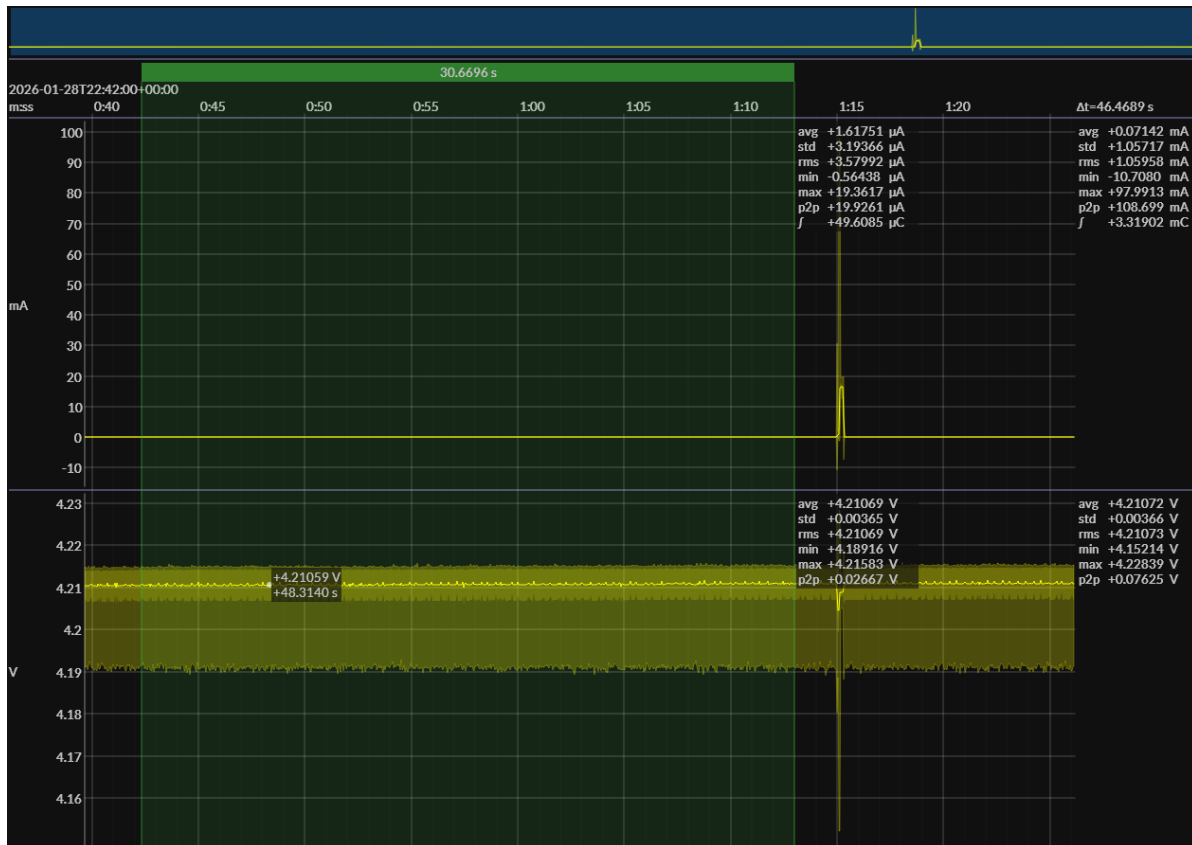


Figura 9.7: Medición de corriente en *Stop Mode 2* mediante Joulescope (estado de suspensión profunda con periféricos desenergizados).

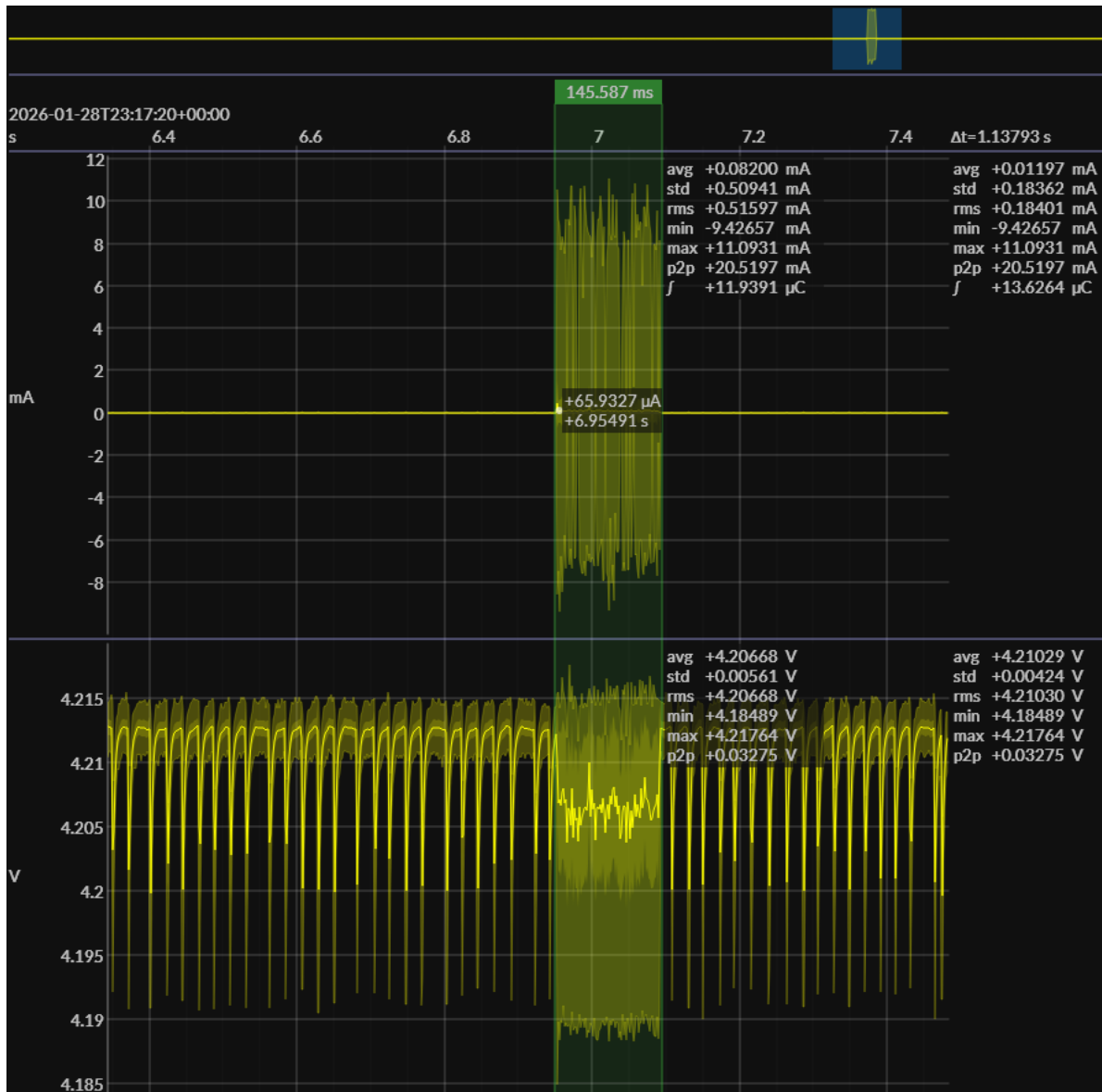


Figura 9.8: Captura de consumo asociada a la interrupción del pluviómetro: evento breve que actualiza el conteo sin ejecutar el ciclo completo de adquisición.



carga (V\_SENSOR y V\_TRANS) y se confirmó el estado de pines *parked* durante suspensión.

La evaluación se organizó por **estados de operación**:

1. **Suspensión profunda** (*Stop Mode 2*): medición de la corriente de reposo con periféricos desenergizados y buses desinicializados, para cuantificar el consumo base del sistema.
2. **Evento pluviómetro (EXTI)**: medición del transitorio asociado a la interrupción externa, donde se incrementa el contador de pulsos sin activar el ciclo completo de sensores.
3. **Ciclo periódico RTC (60 s)**: medición del evento de *wake-up* por temporizador/alarm del RTC, incluyendo energización del riel de sensores, re-inicialización de buses, lectura de sensores y escritura en microSD, seguido del retorno a suspensión.
4. **Transmisión LoRa**: una vez corregido el dominio de comunicaciones, medición del incremento de consumo durante la ventana de transmisión, energizando el LR62E sólo durante el envío y desenergizándolo inmediatamente después.

Para cada estado se registró: el **valor base** en reposo previo al evento, el **perfil temporal** de corriente durante el evento (incluyendo picos y duración), y el retorno al estado de baja potencia.

Finalmente, los resultados se documentaron como capturas representativas del Joulescope para cada caso, mostrando tanto la corriente instantánea como el comportamiento transitorio, lo que permite comparar el impacto energético relativo de cada evento y justificar las decisiones de gestión de energía implementadas en el firmware.

### 9.3.2. Consumo promedio y comentario sobre dimensionamiento: batería vs./aporte solar

A partir de las capturas de Joulescope se estimó el consumo promedio considerando un *periodo base* de **15 min** ( $T = 900$  s), ya que en dicho intervalo se repite el patrón completo de operación incluyendo el evento más demandante (transmisión). Para la configuración evaluada se utilizaron los siguientes parámetros medidos:  $I_{\text{sleep}} = 1,6 \mu\text{A}$ , evento de adquisición+SD con  $I_{\text{sd}} = 10$  mA durante  $t_{\text{sd}} = 0,315$  s cada 60 s (15 veces por periodo base), y transmisión LoRa con  $I_{\text{tx}} = 45$  mA durante  $t_{\text{tx}} = 15$  s (1 vez por periodo base). La carga consumida en un periodo se calculó como:

$$Q_{\text{tx}} = I_{\text{tx}} t_{\text{tx}} = 45 \text{ mA} \cdot 15 \text{ s} = 675 \text{ mAs} \quad (9.1)$$

$$Q_{\text{sd}} = I_{\text{sd}} t_{\text{sd}} \cdot 15 = 10 \text{ mA} \cdot 0,315 \text{ s} \cdot 15 = 47,25 \text{ mAs} \quad (9.2)$$

$$t_{\text{sleep}} = T - (t_{\text{tx}} + 15 t_{\text{sd}}) = 900 - 19,725 = 880,275 \text{ s} \quad (9.3)$$

$$Q_{\text{sleep}} = I_{\text{sleep}} t_{\text{sleep}} = 0,0016 \text{ mA} \cdot 880,275 \text{ s} \approx 1,41 \text{ mAs} \quad (9.4)$$

Por lo tanto,  $Q_{\text{total}} = 723,66$  mAs y el consumo promedio equivalente resulta:

$$I_{\text{prom}} = \frac{Q_{\text{total}}}{T} \approx \frac{723,66}{900} = 0,804 \text{ mA} \quad (\approx 804 \mu\text{A}) \quad (9.5)$$

Este valor implica un consumo diario aproximado de:

$$C_{\text{día}} = I_{\text{prom}} \cdot 24 \text{ h} \approx 19,3 \text{ mAh/día} \quad (9.6)$$

**Caso batería (sin aporte solar).** La autonomía se aproxima como días  $\approx \eta_{\text{util}} C_{\text{bat}}/C_{\text{día}}$ , donde  $\eta_{\text{util}}$  representa capacidad utilizable (p. ej., 0,8 por márgenes de voltaje, temperatura y envejecimiento). Para referencia:

- $C_{\text{bat}} = 2200 \text{ mAh} \Rightarrow \sim 114 \text{ días ideal } (\sim 91 \text{ días con } \eta_{\text{util}} = 0,8)$ .
- $C_{\text{bat}} = 5000 \text{ mAh} \Rightarrow \sim 259 \text{ días ideal } (\sim 207 \text{ días con } \eta_{\text{util}} = 0,8)$ .
- Para **1 año** de operación continua:  $C_{\text{bat}} \approx 19,3 \cdot 365 \approx 7040 \text{ mAh (ideal)}$ , o  $\sim 8800 \text{ mAh}$  considerando  $\eta_{\text{util}} = 0,8$ .

**Caso con aporte solar.** El aporte solar permite compensar el consumo diario ( $\sim 19,3 \text{ mAh/día}$ ) y reducir el tamaño de batería requerido, pero en operación real debe considerarse la eficiencia del camino energético (harvester, carga, reguladores) y variabilidad por sombras/estación. Como guía, si se asume una eficiencia global  $\eta_{\text{sist}}$  (típicamente 0,4-0,7) y  $H_{\text{PSH}}$  horas solares pico por día, la energía requerida desde el panel escala como  $E_{\text{panel}} \approx E_{\text{carga}}/(\eta_{\text{sist}} H_{\text{PSH}})$ . En la práctica, esto favorece una arquitectura híbrida (solar + almacenamiento) para asegurar operación nocturna y durante días de baja irradiancia.

### 9.3.3. Registro ambiental: temperatura, humedad y presión

Con el firmware validado (despertar periódico, lectura de sensores y almacenamiento), se realizó un registro continuo de variables ambientales desde el **26/01/2026 12:42** hasta el **28/01/2026 07:56**. Durante este intervalo se almacenaron muestras periódicas de **temperatura** (TMP117), **humedad relativa y presión** (BME280) en el archivo `DATA.csv`.

La Fig. 9.10 muestra la evolución temporal de la temperatura, donde se aprecia el patrón diurno/nocturno característico. La Fig. 9.11 presenta la humedad relativa, que varía de forma inversa a la temperatura en parte del periodo, consistente con cambios de enfriamiento nocturno y condiciones locales. La Fig. 9.12 muestra la presión atmosférica, con rangos comparables a registros meteorológicos típicos para la zona, aunque con diferencias atribuibles a la ubicación exacta y a la altitud relativa.

Adicionalmente, al comparar temperatura y presión del prototipo con un registro referencial Fig.??, se observó que la **temperatura sigue el mismo patrón general**, pero con **límites superiores e inferiores desplazados** debido a las condiciones de prueba: el sensor se ubicó en **ambiente exterior pero cercano a una edificación y murallas**, lo que modifica el balance térmico por **sombra parcial, radiación reflejada y menor ventilación**, alterando la amplitud térmica observada. En el caso de la **presión**, los rangos resultan similares, pero existe una **diferencia en valores absolutos** consistente con la **distinta ubicación** del registro referencial (cambios de altitud/localización y microvariaciones espaciales).

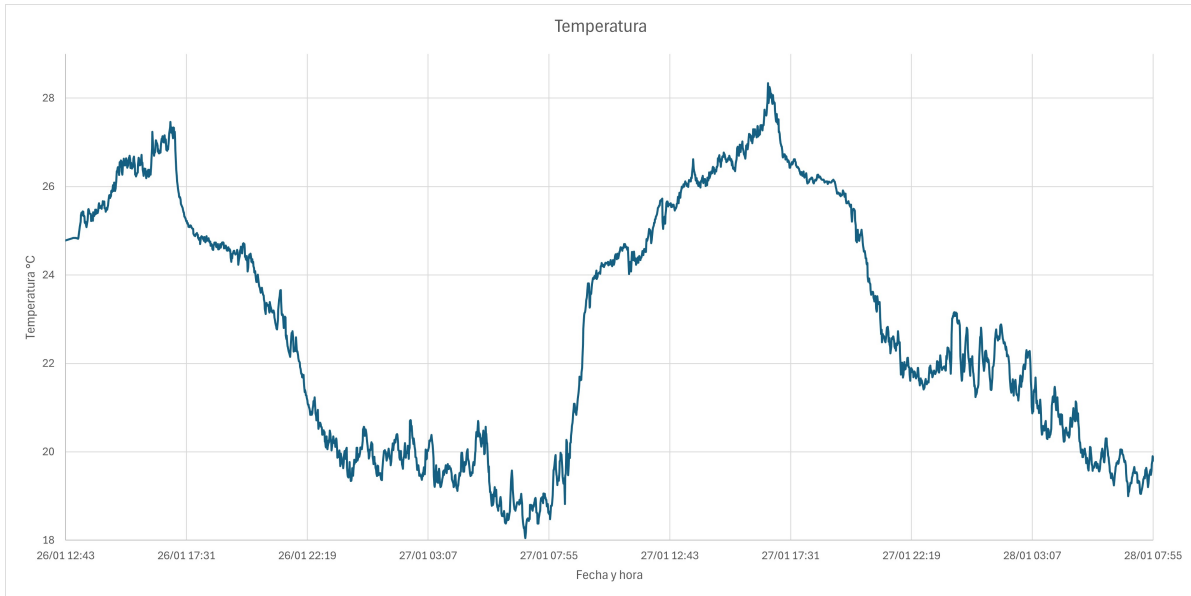


Figura 9.10: Serie temporal de temperatura registrada por el prototipo (TMP117) entre el 26/01/2026 12:42 y el 28/01/2026 07:56.

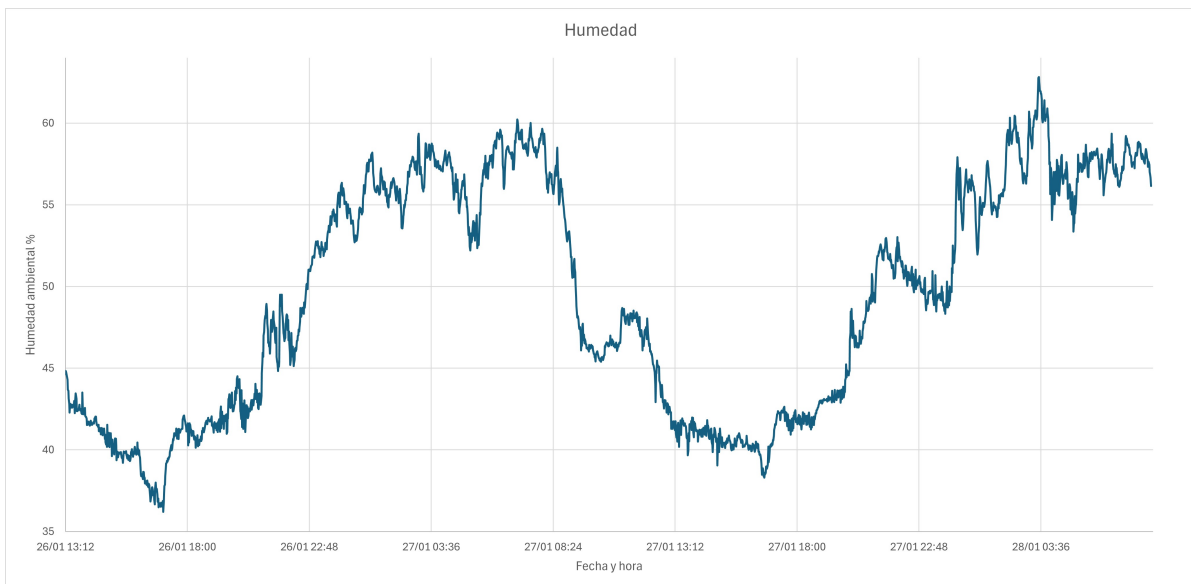


Figura 9.11: Serie temporal de humedad relativa registrada por el prototipo (BME280) en el periodo de prueba (26/01/2026–28/01/2026).

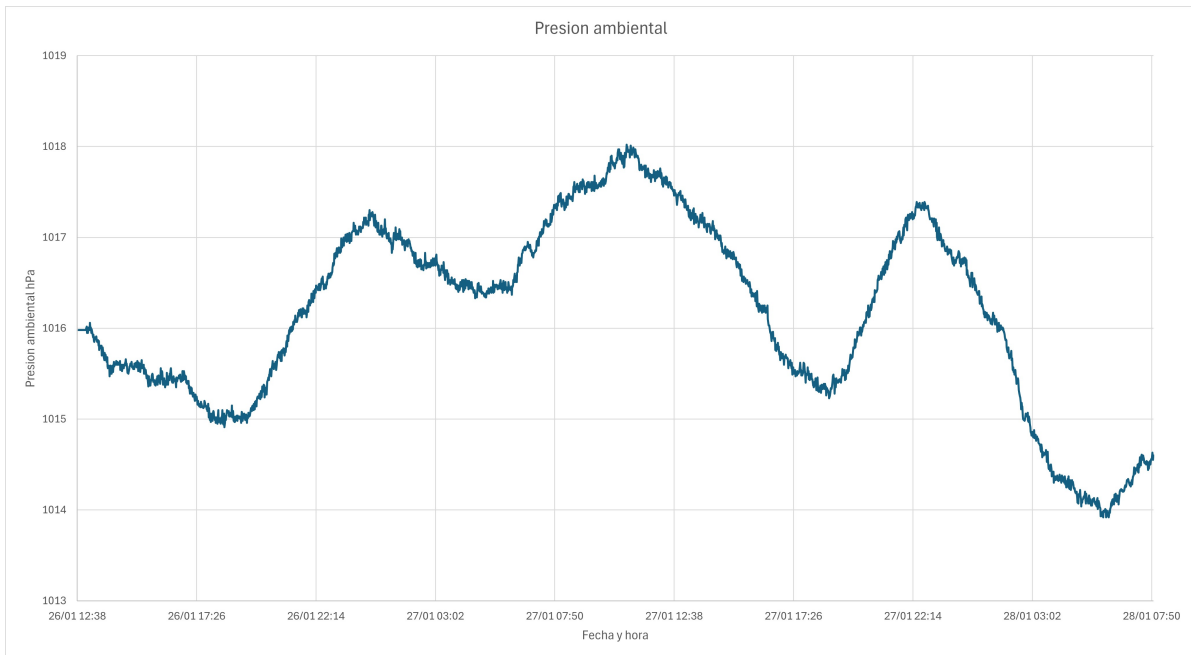


Figura 9.12: Serie temporal de presión atmosférica registrada por el prototipo (BME280) en el periodo de prueba (26/01/2026–28/01/2026).

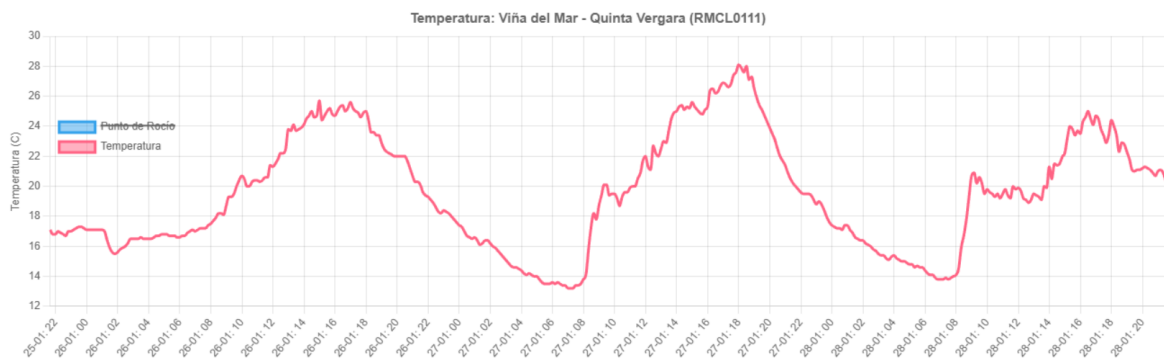


Figura 9.13: Temperatura referencial viña del mar obtenida de redmeteo.cl - Quinta Vergara.

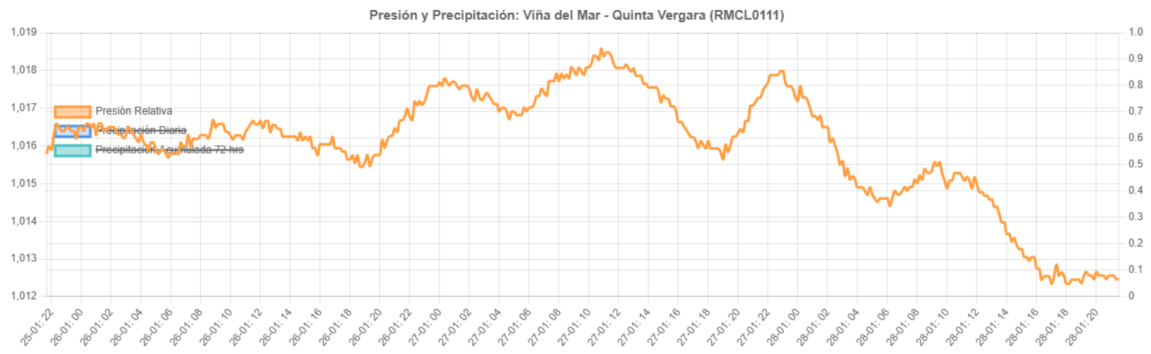


Figura 9.14: Presión referencial viña del mar obtenida de redmeteo.cl - Quinta Vergara.

---

---

# CONCLUSIONES Y TRABAJO FUTURO

## 9.4. Conclusiones

En este proyecto de título se desarrolló una arquitectura modular de *data logger* ambiental de ultra bajo consumo, orientada a operación autónoma de largo plazo, capaz de medir variables meteorológicas (temperatura, presión, humedad y precipitación), registrar localmente y habilitar conectividad LPWAN. Con el objetivo de cubrir escenarios contrastantes de despliegue y disponibilidad energética, se definieron y diseñaron **cuatro variantes de hardware** (LoRa / NB-IoT, con batería primaria o secundaria, y con o sin aporte solar). Esta estrategia permitió desacoplar el subsistema energético y el módulo de comunicaciones, manteniendo una base común de sensores, almacenamiento y arquitectura de firmware. En esta etapa se fabricó y validó experimentalmente la variante **LoRa + batería secundaria + aporte solar**, seleccionada por ofrecer un compromiso adecuado entre continuidad de operación, caracterización de consumo y validación de extremo a extremo (adquisición–registro–transmisión).

Como complemento al diseño físico, se implementó una **herramienta de selección de variante** que formaliza la toma de decisión según condiciones del sitio (irradiancia, cobertura y restricciones operacionales). Esta herramienta integra información geoespacial y parámetros del sistema (perfil de consumo, disponibilidad energética y requerimientos de conectividad), entregando una recomendación reproducible de arquitectura (LoRa vs. NB-IoT y solar vs. batería) para coordenadas dadas. Con lo anterior, el proyecto integra el diseño del nodo y sistematiza un conjunto de herramientas y procedimientos que habilitan su adopción y la planificación de despliegues en terreno.

A nivel de firmware, un resultado central fue la consolidación de una arquitectura robusta de bajo consumo basada en *Stop Mode 2*. Se verificó que, en un sistema donde los periféricos se desenergizan por dominios mediante *load switches*, es **necesaria una secuencia explícita de apagado/encendido y reinicialización** de buses **I<sup>2</sup>C/SPI**, acompañada del **parking de pines** para minimizar fugas y evitar retroalimentación (*back-feeding*). En particular, se estableció que la reanudación determinística tras ciclos repetidos de suspensión requiere, desinicialización de periféricos antes de cortar rieles, reconfiguración temporal de GPIO para recuperación del bus I<sup>2</sup>C, y re-sincronización del RTC tras reconexión de alimentación, ya que el **RV-3028** puede quedar bloqueado ante pérdida de comunicación si no se fuerza una recuperación del bus y limpieza de banderas.

Durante la verificación experimental se detectaron y corrigieron condiciones relevantes para consumo y seguridad eléctrica. Entre ellas, se identificó el efecto de mantener el entorno de depuración activo (programación sin STM32CubeProgrammer), lo que elevaba el consumo en reposo, y se corrigieron aspectos de integración de hardware como, la desconexión efectiva del dominio de comunicaciones, la revisión de señales que podían aplicar niveles de tensión no deseados a GPIO (p. ej. el uso de VBAT\_OK del BQ25570 sin requerir *pull-ups* externos y su potencial impacto de nivel), y la duplicidad de *pull-up* en la línea de interrupción del RTC. Estas correcciones permitieron establecer una metodología de caracterización basada en **Joulescope**, indispensable para cuantificar consumos del orden de microamperes y transitorios breves.

En cuanto a resultados medidos, se confirmó el funcionamiento correcto del sistema bajo operación periódica (interrupción del RTC cada **60 s**, despertar desde *Stop Mode 2*, lectura de sensores y escritura consistente en microSD). Tras las correcciones descritas, se obtuvo un consumo promedio en suspensión profunda de aproximadamente **1,6  $\mu\text{A}$**  con periféricos desenergizados. El ciclo de adquisición y almacenamiento (lectura de sensores y escritura en SD) presentó un consumo promedio de aproximadamente **10 mA** con una duración típica de **313,520 ms** entre el inicio del *wake-up* y el retorno a *Stop Mode 2*. Al reintegrar el dominio de comunicaciones y medir una transmisión representativa, se observó un consumo del orden de **45 mA** durante la etapa de RF, concentrando una fracción relevante del presupuesto energético del ciclo. Considerando un esquema de envío periódico (por ejemplo, una transmisión cada 15 minutos), el consumo promedio del sistema se estimó del orden de **0,804 mA** en dicho periodo base, valor que entrega una referencia directa para dimensionamiento energético.

Finalmente, se implementó y validó una infraestructura de recepción LoRaWAN local basada en **gateway Dragino** y un **SBC** ejecutando **ChirpStack** con servicios asociados (incluyendo MQTT) en contenedores, dejando un flujo reproducible desde la recepción de tramas hasta la visualización y almacenamiento de eventos. Si bien el trabajo se centró en validar el nodo y su operación de bajo consumo con registro local, la plataforma queda preparada para campañas de terreno y para ensayos sistemáticos de comunicación y autonomía. En particular, los resultados de consumo permiten discutir el compromiso entre operación con **batería** y **aporte solar**, dada la contribución dominante de la transmisión y de los ciclos activos, el uso de *energy harvesting* reduce la exigencia de capacidad de batería para autonomía extendida, mientras que un esquema exclusivamente a batería requiere dimensionamiento cuidadoso del intervalo de transmisión y del periodo de muestreo. Como proyección, la validación completa de las variantes restantes (NB-IoT y configuraciones sin aporte solar), junto con pruebas prolongadas en condiciones reales de irradiancia y cobertura, permitirán cerrar el ciclo de diseño orientado a despliegues de largo plazo.

## 9.5. Trabajo futuro

El trabajo desarrollado deja una plataforma funcional y una base metodológica para extender la validación hacia todas las variantes propuestas y consolidar la operación en terreno. Como líneas principales de trabajo futuro se proponen:

### 9.5.1. Validación de hardware en variantes restantes

- **Fabricación y validación de variantes pendientes.** Fabricar y ensamblar las versiones LoRa con batería primaria y NB-IoT (con y sin aporte solar), verificando consumo por estado (suspensión, adquisición/SD, transmisión), estabilidad de rieles, y comportamiento ante transitorios. En particular, para NB-IoT se debe caracterizar la demanda dinámica del módem y la robustez del subsistema de potencia frente a picos de corriente.
- **Comparación experimental entre arquitecturas.** Ejecutar una evaluación cruzada entre variantes contrastando autonomía *estimada vs. medida*, márgenes de diseño del subsistema energético, sensibilidad a temperatura y degradación de batería/panel. Esto permitiría refinar supuestos del modelo energético y ajustar parámetros de operación (intervalos de muestreo, agrupación y envío).
- **Pruebas ambientales y robustez.** Incorporar ensayos de operación en rangos térmicos amplios y condiciones de humedad/polvo, así como validación de conectores, protección ESD/TVS

y comportamiento frente a desconexiones intermitentes (batería/panel/sensores).

### 9.5.2. Maduración de software y soporte completo multi-variante

- **Abstracción formal por variante.** Consolidar una base de firmware común con una capa de configuración por variante (energía y radio), de modo que el mismo código compile y opere cambiando únicamente parámetros de *board support* (pines, rieles, presencia de periféricos y tipo de radio).
- **Políticas de transmisión y tolerancia a fallas.** Completar estrategias de envío para operación prolongada: agrupación por ventana temporal, reintentos con *backoff*, manejo de pérdidas de red, persistencia local de colas de mensajes, y telemetría de estado (batería, eventos de reinicio, razones de *wake-up*, contadores de error I<sup>2</sup>C/SPI).
- **NB-IoT y modos de ahorro.** Integrar la lógica de conexión y registro a red, y validar estrategias de ahorro **PSM/eDRX** con políticas de envío compatibles con la autonomía objetivo. Esto incluye medición de consumo por estado del módem (attach, transmisión, idle, PSM) y definición de umbrales para adaptar periodicidad según energía disponible.

### 9.5.3. Despliegue multi-nodo y validación en terreno

- **Campaña con múltiples nodos simultáneos.** Realizar un despliegue controlado con varios nodos en paralelo para evaluar confiabilidad (pérdida de paquetes, latencia, estabilidad del RTC), operación sostenida y consistencia del registro local en microSD.
- **Contraste con referencias y calibración de modelos.** Comparar mediciones con estaciones cercanas o sensores de referencia para validar calidad de datos (temperatura, presión, humedad y lluvia) y usar los resultados para ajustar parámetros de muestreo/transmisión y mejorar el modelo de factibilidad (energía/cobertura) incorporado en la herramienta de selección.
- **Explotación del backend local.** Extender la infraestructura local (gateway + ChirpStack + MQTT + persistencia/visualización) para soportar monitoreo de flotas, alarmas y paneles comparativos por nodo, habilitando análisis de desempeño y trazabilidad de fallas durante periodos de operación prolongados.

---

---

# BIBLIOGRAFÍA

- [1] P. Horowitz y W. Hill, *The Art of Electronics*, 2nd. Cambridge University Press, 1989.
- [2] S. Paiva, S. Branco y J. Cabral, «Design and Power Consumption Analysis of a NB-IoT End Device for Monitoring Applications,» en *IECON 2020 - The 46th Annual Conference of the IEEE Industrial Electronics Society*, Available at: <https://ieeexplore.ieee.org/document/9254374>, 2020.
- [3] M. Dordevic, «NB-IoT Ultra-low power sensing platform,» Available at: <https://webthesis.biblio.polito.it/34149/>, Master's thesis, Politecnico di Torino, 2024.
- [4] P. Aggarwal, *Top 8 Power Profilers For Low Power Embedded System Design*, Accedido el 14 de mayo de 2025, 2021. dirección: <https://pallavaggarwal.in/dc-current-analyzer-for-embedded-iot-product-development/>.
- [5] Jetperch LLC, *Joulescope User's Guide*, Accedido el 14 de mayo de 2025, 2023. dirección: <https://www.joulescope.com/downloads>.
- [6] H. B. Fan, J. L. Liu y G. Z. Sun, «Design of low-power ultrasonic anemometer based on STM32L476,» *IOP Conference Series: Materials Science and Engineering*, 2018, Available at: <https://iopscience.iop.org/article/10.1088/1757-899X/408/1/012036>.
- [7] E. Hidalgo-Fort, P. Blanco-Carmona, F. Muñoz-Chavero, A. Torralba y R. Castro-Triguero, «Low-cost, low-power edge computing system for structural health monitoring in an IoT framework,» *Sensors*, 2024, Available at: <https://www.mdpi.com/1424-8220/24/15/5078>.
- [8] S. García, D. F. Larios, J. Barbancho, E. Personal, J. M. Mora-Merchán y C. León, «Heterogeneous LoRa-Based Wireless Multimedia Sensor Network Multiprocessor Platform for Environmental Monitoring,» *Sensors*, 2019, Available at: <https://www.mdpi.com/1424-8220/19/16/3446>.
- [9] hardwarex, *Untamed Data Logger*, <https://github.com/hardware-x/UntamedDataLogger>, Accedido el 14 de mayo de 2025, 2025.
- [10] S. B. Designs, *Bee Data Logger*, Accedido el 14 de mayo de 2025, 2024. dirección: <https://github.com/strid3r21/Bee-Data-Logger>.
- [11] J. Baumann, *H32 Basic Board*, <https://github.com/joachim-baumann/H32>, Accedido el 14 de mayo de 2025, 2022.
- [12] STMicroelectronics, *STM32CubeMX for STM32 configuration and initialization C code generation - Power Consumption Calculator (Section 5.3)*, Accedido el 14 de mayo de 2025, 2024. dirección: [https://www.st.com/resource/en/user\\_manual/um1718-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um1718-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf).
- [13] Dickson Data, *Bluetooth Data Loggers*, Available at: <https://dicksondata.com/product/cobalt-x-data-logger>, 2024.
- [14] EnviroDIY, *Mayfly Data Logger*, Available at: <https://www.envirodiy.org>, 2024.

- [15] Texas Instruments, *SLUC484: bq25505/bq25570 Design Help V1.3*, <https://www.ti.com/tool/SLUC484>, Calculation Tool (Spreadsheet). Accedido el 22 de enero de 2026, 2013.
- [16] STMicroelectronics, *AN4555: Getting started with STM32L4 Series and STM32L4+ Series hardware development*, [https://www.st.com/resource/en/application\\_note/an4555-getting-started-with-stm32l4-series-and-stm32l4-series-hardware-development-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4555-getting-started-with-stm32l4-series-and-stm32l4-series-hardware-development-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2022.
- [17] STMicroelectronics, *AN4621: STM32L4 and STM32L4+ ultra-low-power features overview*, [https://www.st.com/resource/en/application\\_note/an4621-stm32l4-and-stm32l4-ultralowpower-features-overview-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4621-stm32l4-and-stm32l4-ultralowpower-features-overview-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2019.
- [18] STMicroelectronics, *AN4899: STM32 microcontroller GPIO hardware settings and low-power consumption*, [https://www.st.com/resource/en/application\\_note/an4899-stm32-microcontroller-gpio-hardware-settings-and-lowpower-consumption-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4899-stm32-microcontroller-gpio-hardware-settings-and-lowpower-consumption-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2022.
- [19] STMicroelectronics, *AN2867: Guidelines for oscillator design on STM8AF/AL/S and STM32 MCUs/MPUs*, [https://www.st.com/resource/en/application\\_note/an2867-guidelines-for-oscillator-design-on-stm8afals-and-stm32-mcusmpus-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an2867-guidelines-for-oscillator-design-on-stm8afals-and-stm32-mcusmpus-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2025.
- [20] STMicroelectronics, *AN5595: SPC58xEx/SPC58xGx multimedia card via SPI interface*, [https://www.st.com/resource/en/application\\_note/an5595-spc58xexspc58xgx-multimedia-card-via-spi-interface-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an5595-spc58xexspc58xgx-multimedia-card-via-spi-interface-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2021.
- [21] STMicroelectronics, *AN4879: Introduction to USB hardware and PCB guidelines using STM32 MCUs*, [https://www.st.com/resource/en/application\\_note/an4879-introduction-to-usb-hardware-and-pcb-guidelines-using-stm32-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an4879-introduction-to-usb-hardware-and-pcb-guidelines-using-stm32-mcus-stmicroelectronics.pdf), Application Note. Accedido el 22 de enero de 2026, 2025.
- [22] Texas Instruments, *SNOA967A: Temperature sensors: PCB guidelines for surface mount devices*, <https://www.ti.com/lit/an/snoa967a/snoa967a.pdf>, Application Report. Revisado en enero 2019. Accedido el 22 de enero de 2026, 2019.
- [23] Saft, *How we choose primary batteries to achieve the required lifespan*, Artículo técnico del fabricante sobre selección de baterías primarias para IoT, 2024. visitado 5 de ene. de 2026. dirección: <https://saft.com/en/energizing-iot/how-we-choose-primary-batteries-achieve-required-lifespan>.
- [24] ChaN, *FatFs – Generic FAT File System Module*, [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html), Project page and documentation. Accedido el 22 de enero de 2026, 2025.
- [25] STMicroelectronics, *STM32CubeL4 Firmware Package (STM32Cube\_FW\_L4) v1.18.2*, <https://www.st.com/en/embedded-software/stm32cube14.html>, Firmware package including middleware integration (e.g., FatFs). Accedido el 22 de enero de 2026, 2022.
- [26] Semtech, *LoRaMac-node: Reference implementation of a LoRa network stack*, ver. v4.7.0, Accedido el 27 de enero de 2026, 2026. dirección: <https://github.com/Lora-net/LoRaMac-node>.
- [27] ChirpStack, *ChirpStack*, <https://www.chirpstack.io/>, Sitio oficial del proyecto. Accedido el 22 de enero de 2026.

- [28] InfluxData, *InfluxDB: The Open Source Time Series Database*, <https://www.influxdata.com/>, Accedido el 24 de enero de 2026, 2026.
- [29] Grafana Labs, *Grafana: The open observability platform*, <https://grafana.com/>, Accedido el 24 de enero de 2026, 2026.
- [30] IEEE, *IEEE Recommended Practice for Sizing Stand-Alone Photovoltaic (PV) Systems*, 1562-2021, IEEE Std 1562-2021, IEEE, 2021.
- [31] J. A. Duffie y W. A. Beckman, *Solar Engineering of Thermal Processes*, 4.<sup>a</sup> ed. John Wiley & Sons, 2013.
- [32] E. Quiles, C. Roldán-Blay, G. Escrivá-Escrivá y C. Roldán-Porta, «Accurate Sizing of Residential Stand-Alone Photovoltaic Systems Considering System Reliability,» *Sustainability*, vol. 12, n.º 3, pág. 1274, 2020. DOI: 10.3390/su12031274.
- [33] T. user, *Stevenson Screen / Radiation Shield (3D model)*, Creative Commons BY-NC-SA, 2018.
- [34] J. S. Botero-Valencia, M. Mejia-Herrera y J. M. Pearce, «Design and implementation of 3-D printed radiation shields for environmental sensors,» *HardwareX*, 2022.
- [35] xpari, *Solar panel holder*, Thingiverse, Licensed under Creative Commons Attribution–NonCommercial, 2018. dirección: <https://www.thingiverse.com/thing:2864663>.
- [36] STMicroelectronics, *STM32CubeProg – STM32CubeProgrammer software for all STM32*, <https://www.st.com/en/development-tools/stm32cubeprog.html>, Accedido: 28-01-2026, 2026.
- [37] STMicroelectronics, *TN1433: Reference device marking schematics for STM32 microcontrollers and microprocessors*, [https://www.st.com/resource/en/technical\\_note/tn1433-reference-device-marking-schematics-for-stm32-microcontrollers-and-microprocessors-stmicroelectronics.pdf](https://www.st.com/resource/en/technical_note/tn1433-reference-device-marking-schematics-for-stm32-microcontrollers-and-microprocessors-stmicroelectronics.pdf), Technical Note. Accedido el 22 de enero de 2026, 2025.
- [38] STMicroelectronics, *STM32L476xx: Ultra-low-power Arm Cortex-M4 32-bit MCU+FPU*, Datasheet. Accedido el 22 de enero de 2026, 2024. dirección: <https://www.st.com/resource/en/datasheet/stm32l476je.pdf>.
- [39] *UM1075: ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*, User Manual. Rev 9, STMicroelectronics, 2021. dirección: [https://www.st.com/resource/en/user\\_manual/um1075-stlinkv2-incircuit-debuggerprogrammer-for-stm8-and-stm32-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um1075-stlinkv2-incircuit-debuggerprogrammer-for-stm8-and-stm32-stmicroelectronics.pdf).
- [40] *ARM-JTAG 20-10 ADAPTER*, Rev. A. User Manual, Olimex Ltd., 2011. dirección: <https://www.olimex.com/Products/ARM/JTAG/ARM-JTAG-20-10/resources/ARM-JTAG-20-10.pdf>.
- [41] Texas Instruments, *bq25570: Nano Power Boost Charger and Buck Converter for Energy Harvester Powered Applications*, Datasheet SLUSBH2G. Revised March 2019, 2019. dirección: <https://www.ti.com/lit/ds/symlink/bq25570.pdf>.
- [42] Texas Instruments, *TPS62840: 1.8-V to 6.5-V, 750-mA, 60-nA IQ Step-Down Converter*, Datasheet SLVSEC6D. Revised March 2020, 2020. dirección: <https://www.ti.com/lit/ds/symlink/tps62840.pdf>.
- [43] Texas Instruments, *TPS22916: 1-V to 5.5-V, 2-A, 60-mΩ Ultra-Low Leakage Load Switch*, Datasheet SLVSDO5F. Revised December 2021, 2021. dirección: <https://www.ti.com/lit/ds/symlink/tps22916.pdf>.

- 
- [44] Texas Instruments, *TMP117: High-Accuracy, Low-Power, Digital Temperature Sensor with SMBus- and I2C-Compatible Interface*, Datasheet SNOSD82D. Revised September 2022, 2022. dirección: <https://www.ti.com/lit/ds/symlink/tmp117.pdf>.
- [45] Bosch Sensortec, *BME280: Digital Humidity, Pressure and Temperature Sensor*, Datasheet Rev 1.24. Doc BST-BME280-DS001-24, 2024. dirección: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf>.
- [46] *RV-3028-C7: Extreme Low Power Real-Time Clock Module with I2C-Bus Interface*, Application Manual Rev. 1.4, Micro Crystal AG, 2021. dirección: [https://www.microcrystal.com/fileadmin/Media/Products/RTC/App\\_Manual/RV-3028-C7\\_App-Manual.pdf](https://www.microcrystal.com/fileadmin/Media/Products/RTC/App_Manual/RV-3028-C7_App-Manual.pdf).
- [47] Fanstel Corp., *LR62E, LR62XE LoRa and LoRa-BLE Combo Modules*, Datasheet Ver 1.04, 2025. dirección: <https://www.fanstel.com/s/LR62E-datasheet.pdf>.
- [48] Würth Elektronik, *Quartz Crystal Specification: Part No. 830106162701*, Datasheet generated 16 Dec 2024, 2024. dirección: <https://www.we-online.com/catalog/en/WE-XTAL-32-830106162701>.
- [49] Abracon, *32.768kHz SMD Crystal*, Datasheet Revised 08-10-22, 2022. dirección: <https://abracon.com/Resonators/ABS07.pdf>.

# GUÍA DE REPLICACIÓN DEL SISTEMA DE RECEPCIÓN

Esta guía resume los pasos seguidos para reproducir el *stack* local de recepción y administración LoRaWAN utilizando una Raspberry Pi 3 B v1.2 y una puerta de enlace Dragino LPS8N.

## A.1. Instalación de Raspberry Pi OS (64-bit)

1. Grabar la microSD con *Raspberry Pi Imager* usando **Raspberry Pi OS 64-bit**.
2. Iniciar la Raspberry Pi y completar la configuración inicial.
3. Actualizar el sistema:

---

```
1 sudo apt update
2 sudo apt full-upgrade -y
3 sudo reboot
```

---

## A.2. Confirmación de IP de red (host)

Obtener la IP LAN real (no la de Docker):

---

```
1 hostname -I
```

---

Anotar una IP del tipo 10.x.x.x o 192.168.x.x para configurar el gateway.

## A.3. Instalación de Docker y Docker Compose

---

```
1 curl -fsSL https://get.docker.com | sh
2 sudo usermod -aG docker $USER
3 sudo reboot
```

---

Verificación:

```
1 docker --version
2 docker run hello-world
3 docker compose version
```

---

Si el plugin de Compose no estuviese disponible:

```
1 sudo apt install -y docker-compose-plugin
```

---

## A.4. Descarga y ejecución del stack ChirpStack

---

```
1 git clone https://github.com/chirpstack/chirpstack-docker.git
2 cd chirpstack-docker
3 docker compose up -d
4 docker compose ps
```

---

## A.5. Ajustes de región (AU915) en ChirpStack

Editar:

```
1 nano configuration/chirpstack/chirpstack.toml
```

---

Asegurar:

```
1 enabled_regions = ["au915_0"]
```

---

Reiniciar:

```
1 docker compose restart chirpstack
```

---

(Opcional) eliminar archivos de región no utilizados en `configuration/chirpstack/` y reiniciar el servicio.

## A.6. Gateway Bridge (Semtech UDP) para AU915\_0

Crear un archivo de configuración Semtech UDP específico:

```
1 cat > configuration/chirpstack-gateway-bridge/semtech-udp-au915_0.toml << "EOF"
2 [backend.semtech_udp]
3 udp_bind = "0.0.0.0:1700"
4
5 [integration.mqtt]
6 server = "tcp://mosquitto:1883"
7 topic_prefix = "au915_0"
8 EOF
```

---

Editar `docker-compose.yml` para:

- Publicar **1700/UDP**.
- Montar el directorio de configuración.
- Cargar ambos archivos de configuración con `command:`.

Ejemplo (referencial):

---

```

1 chirstack-gateway-bridge:
2   ports:
3     - "1700:1700/udp"
4   volumes:
5     - ./configuration/chirstack-gateway-bridge:/etc/chirstack-gateway-bridge
6   command:
7     ["-c", "/etc/chirstack-gateway-bridge/chirstack-gateway-bridge.toml",
8     "-c", "/etc/chirstack-gateway-bridge/semtech-udp-au915_0.toml"]

```

---

Validar y reiniciar:

---

```

1 docker compose config >/dev/null && echo "compose file OK"
2 docker compose up -d
3 docker compose restart chirstack-gateway-bridge

```

---

## A.7. Verificación de puerto UDP 1700

---

```

1 sudo ss -ltnp | grep 1700

```

---

## A.8. Configuración del Dragino LPS8N (AU915, Semtech UDP)

En la interfaz web del Dragino:

- Modo: **Semtech UDP packet forwarder**
- Región: **AU915**
- Subbanda: **0** (canales 0–7), consistente con `au915_0`
- Servidor: **IP LAN de la Raspberry Pi**
- Puerto: **1700**

Guardar y reiniciar el gateway.

## A.9. Confirmación de llegada de paquetes al servidor

Instalar `tcpdump` si es necesario:

---

```

1 sudo apt update
2 sudo apt install -y tcpdump

```

---

Monitorear:

---

```

1 sudo tcpdump -n udp port 1700

```

---

## A.10. Confirmación de publicación MQTT

---

```
1 docker compose exec mosquitto mosquitto_sub -v -t '#'
```

---

Se esperan tópicos del tipo: au915\_0/gateway/<gatewayid>/event/...

## A.11. Configuración en la interfaz web de ChirpStack

Acceso desde navegador:

- `http://<IP_RPI>:8080` (o `http://localhost:8080` si se navega en la misma Raspberry Pi)

Credenciales por defecto: `admin/admin`.

Luego:

1. Crear/usar **Tenant**.
2. Registrar **Gateway** (Gateway ID debe coincidir con MQTT).
3. Crear **Device profile** (AU915 y versión MAC coherente con el nodo).
4. Crear **Application** y añadir **Device** (OTAA), ingresando DevEUI/JoinEUI/AppKey (y Nwk-Key si aplica).

## A.12. Respaldo de configuración

---

```
1 cp docker-compose.yml docker-compose.yml.backup
2 tar -czf chirpstack-config-backup.tar.gz configuration/
```

---

# GUÍA DE REPLICACIÓN: VISUALIZACIÓN LOCAL EN RASPBERRY PI CON INFLUXDB Y GRAFANA

Este apéndice documenta el procedimiento utilizado para habilitar una instancia de **visualización local** en una **Raspberry Pi 3 B v1.2**, integrando **InfluxDB** (base de datos de series de tiempo) y **Grafana** (paneles de visualización) dentro del mismo *stack* Docker Compose donde se ejecuta ChirpStack. El objetivo es disponer de una plataforma autónoma, accesible desde la red local, para consultar métricas y series temporales sin depender de servicios externos.

## B.1. Requisitos y supuestos

Antes de comenzar se asume:

- Raspberry Pi con **Raspberry Pi OS 64-bit** configurado y con conectividad de red.
- **Docker** y el complemento **Docker Compose** instalados y operativos.
- El *stack* de ChirpStack ya está clonado y funcionando en `~/chirpstack-docker`.

## B.2. Estrategia de integración

En lugar de desplegar Grafana e InfluxDB como servicios independientes, se incorporan como **nuevos contenedores** dentro del mismo archivo `docker-compose.yml`. Esto permite:

- Compartir la red interna de Docker, usando nombres de servicio (p. ej., `influxdb`) para interconexión.
- Persistir datos mediante **volúmenes** administrados por Docker.
- Simplificar el arranque del sistema completo con un único comando (`docker compose up -d`).

## B.3. Procedimiento paso a paso

### B.3.1. 1) Acceder al directorio del stack

En la Raspberry Pi, ingresar al directorio donde se encuentra el *compose* de ChirpStack:

---

```
1 cd ~/chirpstack-docker
```

---

### B.3.2. 2) Extender el docker-compose.yml

Se edita el archivo `docker-compose.yml` para añadir dos servicios: `influxdb` y `grafana`, además de sus volúmenes persistentes.

---

```
1 nano docker-compose.yml
```

---

#### 2.1 Servicio InfluxDB

InfluxDB se expone en el puerto 8086 hacia la red local, y se inicializa automáticamente con un usuario, organización, *bucket* y token de administración. La persistencia se asegura con el volumen `influxdb_data`.

---

```
1 influxdb:
2   image: influxdb:2.7
3   ports:
4     - "8086:8086"
5   volumes:
6     - influxdb_data:/var/lib/influxdb2
7   environment:
8     - DOCKER_INFLUXDB_INIT_MODE=setup
9     - DOCKER_INFLUXDB_INIT_USERNAME=admin
10    - DOCKER_INFLUXDB_INIT_PASSWORD=changeme123
11    - DOCKER_INFLUXDB_INIT_ORG=myorg
12    - DOCKER_INFLUXDB_INIT_BUCKET=lorawan
13    - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=my-super-token
```

---

#### 2.2 Servicio Grafana

Grafana se expone en el puerto 3000. Se configura con persistencia en `grafana_data` y se declara dependencia de InfluxDB para ordenar el arranque.

---

```
1 grafana:
2   image: grafana/grafana:latest
3   ports:
4     - "3000:3000"
5   volumes:
6     - grafana_data:/var/lib/grafana
7   depends_on:
8     - influxdb
```

---

## 2.3 Volúmenes persistentes

Al final del archivo, se declaran los volúmenes utilizados por ambos servicios (si la sección `volumes`: ya existe, solo se agregan las entradas faltantes):

---

```
1 volumes:
2   influxdb_data:
3   grafana_data:
```

---

Guardar y salir del editor.

### B.3.3. 3) Levantar el stack actualizado

Una vez modificada la definición, se recrean los contenedores necesarios:

---

```
1 docker compose up -d
```

---

### B.3.4. 4) Verificación de estado

Para confirmar que los servicios quedaron activos:

---

```
1 docker compose ps
```

---

El resultado debe mostrar `Up` (o equivalente) en `influxdb` y `grafana`, además de los servicios del stack de ChirpStack.

### B.3.5. 5) Acceso desde la red local

Para acceder desde un PC en la misma red, se obtiene la IP de la Raspberry Pi:

---

```
1 hostname -I
```

---

Luego, desde el navegador:

- InfluxDB: `http://<IP_PI>:8086`
- Grafana: `http://<IP_PI>:3000`

### B.3.6. 6) Inicialización y credenciales

InfluxDB queda preconfigurado por variables de entorno. En el primer acceso se utiliza:

- Usuario: `admin`
- Clave: `changeme123`

Grafana utiliza credenciales por defecto:

- Usuario: `admin`
- Clave: `admin`

y solicitará cambiar la contraseña al primer inicio.

### B.3.7. 7) Conexión de Grafana a InfluxDB

Para conectar Grafana con la base de datos:

1. En Grafana, ir a **Connections** → **Data sources**.
2. Seleccionar **Add data source** y elegir **InfluxDB**.
3. Configurar el modo **Flux**.

#### 7.1 URL del servicio

Como ambos contenedores comparten red de Docker, la URL debe apuntar al nombre de servicio:

---

```
1 http://influxdb:8086
```

---

#### 7.2 Parámetros Flux

Completar los campos:

- **Organization:** myorg
- **Token:** my-super-token
- **Default bucket:** lorawan

Finalmente, ejecutar **Save & Test** para validar conectividad.

### B.3.8. 8) Arranque automático tras reinicio

Para asegurar que Docker se ejecute al arrancar el sistema operativo:

---

```
1 sudo systemctl enable docker
```

---

Si tras un reinicio fuera necesario levantar manualmente el stack:

---

```
1 cd ~/chirpstack-docker
2 docker compose up -d
```

---

## B.4. Notas prácticas de operación

- El uso de **volúmenes** evita pérdida de datos al recrear contenedores.
- Los puertos 8086 y 3000 deben estar accesibles en la red local; en redes restringidas puede requerirse ajustar reglas de firewall.
- En una Raspberry Pi 3 B, el desempeño de Grafana e InfluxDB depende fuertemente del volumen de datos y la frecuencia de escritura; para despliegues prolongados se recomienda limitar retención y/o tasa de muestreo.

# DOCUMENTACIÓN DE DISEÑO CONSULTADA

Para el diseño de la placa y su ruteo se consultaron hojas de datos, manuales de usuario y las siguientes guías técnicas proporcionadas por los fabricantes:

### Guías de Aplicación y Notas Técnicas

- Interfaz de almacenamiento (microSD vía SPI): [20].
- Subsistema de recolección/carga (*harvester*): [15].
- Lineamientos de hardware para STM32L4/L4+: [16].
- Funciones de ultra bajo consumo STM32L4/L4+: [17].
- Configuración de GPIO y consumo en bajo consumo: [18].
- Diseño de osciladores: [19].
- USB (hardware y guías de PCB): [21].
- Sensores SMD (recomendaciones de PCB): [22].
- Identificación y marcado de dispositivos (Ref. técnica): [37].

### Hojas de Datos y Manuales de Componentes

- **Microcontrolador y Herramientas:**
  - MCU STM32L476xx: [38].
  - Depurador ST-LINK/V2 (Manual de usuario): [39].
  - Adaptador JTAG (Olimex): [40].
- **Gestión de Energía:**
  - Cargador/Harvester (bq25570): [41].
  - Convertidor Buck (TPS62840): [42].
  - Switch de carga (TPS22916): [43].

**■ Sensores, Radio y Periféricos:**

- Sensor de temperatura de precisión (TMP117): [44].
- Sensor ambiental (BME280): [45].
- Módulo RTC de bajo consumo (RV-3028-C7): [46].
- Módulo LoRa/BLE (Fanstel LR62E): [47].
- Cristales de cuarzo (32.768 kHz): [48], [49].

# BOM Y ESTIMACIÓN DE COSTO DE FABRICACIÓN (LOTE DE 10 UNIDADES)

Este anexo resume el costo de materiales (*Bill of Materials*, BOM) y una estimación del costo total de fabricación para un lote de **10 placas**. El objetivo es transparentar los componentes considerados, los ítems excluidos del BOM principal y el impacto de añadir costos de manufactura y elementos externos (batería, pluviómetro y cableado).

### D.1. Alcance y criterios de costeo

El costo del BOM reportado (**USD 817.93**) corresponde al costo total de componentes para **10 unidades**, **excluyendo** explícitamente:

- fabricación de PCBs,
- batería principal del sistema.

El BOM considerado **sí incluye** el panel solar seleccionado, caja, prensaestopas y materiales para construcción de cables (con la excepción indicada en la Sección D.2). Los valores corresponden a precios aproximados de compra (no incluyen impuestos, costos de importación, ni variaciones por disponibilidad o cotización).

### D.2. Costos adicionales no incluidos en el BOM principal

Además del BOM base, se identificaron dos elementos relevantes que no estaban incorporados en el total anterior:

- **Cable 4 polos 22 AWG con protección UV**: para una longitud total de **20 m** (suficiente para el conjunto de unidades consideradas), con un costo aproximado de **USD 10** en total.
- **Pluviómetro** (*rain gauge*) **WH-SP-RG**: costo aproximado de **USD 12** por unidad. Para **10 unidades**, el costo total asociado es **USD 120**.

### D.3. Costo de manufactura de placas y batería

Para completar la estimación de costo por unidad, se agregan los siguientes ítems:

- **Fabricación de 10 PCBs: USD 82** (lote completo).
- **Batería LiPo 3500 mAh: USD 5** por unidad. Para 10 unidades: **USD 50**.

### D.4. Resumen de costos y total para 10 unidades

La Tabla D.1 presenta el resumen consolidado del costo estimado para un lote de 10 unidades.

Tabla D.1: Resumen de costos estimados para 10 unidades.

Ítem	Costo (USD)
BOM (10 unidades), sin PCB ni batería	817.93
Fabricación de PCBs (10 unidades)	82.00
Batería LiPo 3500 mAh (10 unidades)	50.00
Cable 4 polos 22 AWG UV (20 m total)	10.00
Pluviómetro WH-SP-RG (10 unidades)	120.00
<b>Total estimado (10 unidades)</b>	<b>1079.93</b>

A partir del total anterior, el costo promedio por unidad se estima como:

$$C_{\text{unidad}} \approx \frac{1079.93}{10} \approx 107.99 \text{ USD/unidad.} \quad (\text{D.1})$$

### D.5. Tabla de materiales (BOM)

La Tabla D.5 presenta el listado de componentes utilizado como base para el cálculo del BOM. En el documento principal se puede incluir una versión resumida (por subsistema), dejando el detalle completo en este anexo para facilitar trazabilidad y actualización.

Tabla D.2: Lista de materiales (BOM) detallada del *Data Logger*.

SKU Distribuidor	Fabricante	Descripción	Unit \$	Cant.	Total \$
534-3000TR	Keystone	Battery holder	0.808	10	8.08
497-15877-ND	STMicroelectronics	Ultra-low-power MCU	7.08	10	70.81
277-1794-ND	Phoenix Contact	Barrier Strip Terminal Block	0.395	20	7.9
710-830106162701	Wurth Elektronik	WE-XTAL Quartz Crystal	0.778	10	7.78
377-PNB-30417-KO-ND	Bud Industries	Enclosure, IP67	6.75	10	67.5
296-39934-1-ND	Texas Instruments	Switching Battery Charger	6.12	10	61.17
640-CAB602RF200A1	Global Conn.	Cable Assembly	5.88	10	58.8
523-GSB1C41110SSHR	Amphenol	USB Connector	0.528	10	5.28
815-ABS07-32.768KHZT	Abracon	Crystal 0.032768MHz	0.5	10	5
769-EVP-BT1A4A000	Panasonic	Tactile Switch	0.247	20	4.94
828-1063-1-ND	Bosch Tools	Sensor (P, T, H)	3.45	10	34.51
2057-ANT-20090EB56-ND	Adam Tech	Rf Antenna, Lora	3.06	10	30.61
649-1012938192403BLF	Amphenol	Header Dr Vt Th 2X12	0.374	10	3.74
P4.7KDBCT-ND	Panasonic	Res Thin Film 0603 4.7K	0.0536	60	3.22
455-4220-100-ND	JST	Connector Accessory	0.0386	100	3.86
TMP117NAIYBGR	Texas Instruments	TEMP SENSOR	2.95	10	29.54

Continúa en la siguiente página...

Continuación de la Tabla D.2					
SKU Distribuidor	Fabricante	Descripción	Unit \$	Cant.	Total \$
WM11190CT-ND	Molex	microSD Connector	2.22	10	22.25
581-0603X7R104JT1AT	KYOCERA AVX	Ceramic Cap, 0.1uF	0.184	110	20.24
581-TAJR475K010R	KYOCERA AVX	Cap Tant Solid 4.7uF	0.268	10	2.68
187-CL21A226KQCLRNC	Samsung	Ceramic Cap, 22uF	0.052	40	2.08
306-B4BXHALFSN	JST	Header for PCB	0.12	20	2.4
81-DFE201612E-2R2MP2	Murata	Inductor 2.2uH	0.2	10	2
10-ERJ-3RBD1002VCT-ND	Panasonic	Res Thick Film 10K	0.0426	70	2.98
511-USBL6-2SC6	STMicroelectronics	Surge Protection	0.243	10	2.43
571-1SNG609010R0000	TE Connectivity	Washer Gasket	0.081	30	2.43
428-203603-MG01	Micro Crystal	Real Time Clock	1.96	10	19.6
108-TLSMDT3C020GLFSC	C&K Components	Switch Tactile	1.83	10	18.26
595-TPS22916CYFPT	Texas Instruments	Load switch	0.776	20	15.52
296-TPS62840DLCRCT-ND	Texas Instruments	REG BUCK ADJ 750MA	1.54	10	15.38
952-3090-1-ND	Harwin	Conn Unshrouded	1.33	10	13.34
730-MPT3.6-150	PowerFilm	Solar Cell 360MW	12.27	10	122.7
308-LR62E	Fanstel	SX1262 Lora Module	11.7	10	117
571-1SNG601011R0000	TE Connectivity	CABLE GLAND	0.339	30	10.17
77-VJ0603A150FXQPBC	Vishay	Ceramic Cap, 15 pF	0.085	20	1.7
81-GRM188R61E475KE1D	Murata	4.7uF Ceramic Cap	0.036	30	1.08
81-GRT1885C1E103JA2J	Murata	Cap Ceramic 0.01uF	0.061	20	1.22
187-CL10A225KA5LNNC	Samsung	Ceramic Cap, 2.2uF	0.085	20	1.7
576-SMBJ5.0A	Littelfuse	TVS Diode	0.147	10	1.47
810-NLV25T-220J-EF	TDK	Ind 22uH	0.143	10	1.43
810-NLV25T-100J-EF	TDK	Inductor 10uH	0.137	10	1.37
667-ERJ-S03F9093V	Panasonic	Res Thick Film 909K	0.111	10	1.11
667-ERA-3AEB512V	Panasonic	Res Thin Film 5.1K	0.054	20	1.08
306-XHP-4	JST	Connector Housing	0.058	20	1.16
621-SMAZ5V1-F	Diodes Inc.	Zener Diode	0.144	10	1.44
187-CL10B105KP8NNNC	Samsung	1uF Ceramic Cap	0.005	30	0.15
187-CL10C200JB8NNNC	Samsung	20pF Ceramic Cap	0.024	20	0.48
81-GRM1857U1A103JA4D	Murata	Ceramic Cap 0.01 uF	0.059	10	0.59
187-CL10A105KQ8NNNC	Samsung	Ceramic Cap, 1uF	0.018	20	0.36
81-GRM188R60G106ME47	Murata	10uF Ceramic Cap	0.046	20	0.92
875-LI0603G221R-10	Laird	Ferrite Beads 220Ohm	0.076	10	0.76
306-S2BPHKSLFSNPP	JST	Shrouded Header	0.082	10	0.82
667-ERJ-3GEYJ391V	Panasonic	Res Thick Film 390	0.017	10	0.17
667-ERJ-3EKF22R0V	Panasonic	Res Thick Film 22	0.021	30	0.63
667-ERJ-6ENF2673V	Panasonic	Res Thick Film 267K	0.033	10	0.33
71-CRCW06037M32FKEA	Vishay	Res Thick Film 7.32M	0.019	10	0.19
603-RC0603FR-075M62L	Yageo	Res Thick Film 5.62M	0.007	10	0.07
603-RC0603FR-071M24L	Yageo	Res Thick Film 1.24M	0.007	10	0.07
541-6.65MHCT-ND	Vishay	Res Thick Film 6.65M	0.03	10	0.3
603-AC0603FR-074M99L	Yageo	Res Thick Film 4.99M	0.008	10	0.08
603-RC0603FR-072M1L	Yageo	Res Thick Film 2.1M	0.007	10	0.07
667-ERJ-3EKF33R0V	Panasonic	Res Thick Film 33	0.021	40	0.84
667-ERJ-3RBD1001V	Panasonic	Res Thick Film 1K	0.043	10	0.43
667-ERJ-3EKF2001V	Panasonic	Res Thick Film 2K	0.021	10	0.21
667-ERJ-3EKF3301V	Panasonic	Res Thick Film 3.3K	0.026	10	0.26
541-3950-1-ND	Vishay	100kΩ Chip Resistor	0.0216	40	0.864
4713-FB160808T-601Y	Cal-Chip	Ferrite Bead 600 Ohm	0.038	10	0.38

*Nota:* Los costos presentados son aproximados y dependen de proveedor, disponibilidad, cantidad, condiciones de envío e impuestos. No se incluyen costos de ensamblaje, pruebas, ni horas de trabajo.