

# UCUENCA

## Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Telecomunicaciones

### Transformación Digital: Modernización del Sistema de Registro Hidrometeorológico del PROMAS.

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

#### **Autores:**

Henry David Maldonado Tuza

Kevin Ismael Mogrovejo Nieves

#### **Director:**

Darwin Fabian Astudillo Salinas

ORCID:  0000-0001-7644-0270

#### **Co-Director:**

Juan Diego Belesaca Mendieta

ORCID:  0000-0001-8609-0358

Cuenca, Ecuador

2024-08-27

## Resumen

En este trabajo se diseña e implementa un dispositivo de registro de datos hidrometeorológicos que integra tecnología Long Range (LoRa) en un microcontrolador ESP32-S3. El dispositivo cuenta con dos módulos Printed Circuit Board (PCB): el primero incluye puertos de conexión para los sensores de la estación y de propósito general, un módulo Analog-Digital Converter (ADC), un contador, un módulo Real Time Clock (RTC) y un interruptor de alimentación; el segundo módulo integra el microcontrolador ESP32 y el chip LoRa. El software está diseñado para minimizar el consumo de energía mediante la reducción de la frecuencia de funcionamiento del CPU, el uso del modo *deep sleep* y la implementación del modelo de trama Variable Meteorological Transmission Protocol (VMTP). Las mediciones de temperatura, humedad, velocidad y dirección del viento, precipitación, radiación solar y UV, así como información sobre el funcionamiento de los sensores y la hora de las mediciones, se envían al servidor The Things Stack (TTS) utilizando el protocolo de red Long Range Wide Area Network (LoRaWAN), para posteriormente establecer una comunicación con Thingsboard y presentar la información en un *dashboard*. El dispositivo mostró un buen funcionamiento en las pruebas de campo, permitiendo la recolección y validación de datos pertenecientes al monitoreo ambiental.

*Palabras clave del autor:* datos hidrometeorológicos, LoRaWAN, monitoreo ambiental TTS, VMTP.



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

**Repositorio Institucional:** <https://dspace.ucuenca.edu.ec/>

## Abstract

In this work, a hydrometeorological data logging device is designed and implemented, integrating LoRa technology into an ESP32-S3 microcontroller. The device consists of two PCB modules: the first includes connection ports for station and general-purpose sensors, an ADC module, a counter, an RTC module, and a power switch; the second module integrates the ESP32 microcontroller and the LoRa chip. The software is designed to minimize power consumption by reducing the CPU operating frequency, using Deep Sleep mode, and implementing the VMTP frame model. Measurements of temperature, humidity, wind speed and direction, precipitation, solar radiation, and UV radiation, as well as information on sensor operation and the time of measurements, are sent to the TTS server using the LoRaWAN network protocol, to later establish communication with Thingsboard and present the information in a dashboard. The device demonstrated good performance in field tests, enabling the collection and validation of environmental monitoring data.

*Author keywords:* hydrometeorological data, LoRaWAN, environmental monitoring, TTS, VMTP.



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

**Institutional Repository:** <https://dspace.ucuenca.edu.ec/>

## Índice de contenido

<b>1. Introducción</b>	<b>17</b>
1.1. Identificación del problema . . . . .	17
1.2. Antecedentes . . . . .	19
1.3. Justificación . . . . .	20
1.4. Alcance . . . . .	22
1.5. Objetivos . . . . .	23
1.5.1. Objetivo general . . . . .	23
1.5.2. Objetivos específicos . . . . .	23
<b>2. Marco teórico y trabajos relacionados</b>	<b>24</b>
2.1. Variables hidrometeorológicas . . . . .	24
2.2. LoRaWAN . . . . .	24
2.2.1. Características de LoRaWAN . . . . .	26
2.2.2. Formato de trama de LoRaWAN . . . . .	27
2.3. Comunicación serial . . . . .	29
2.3.1. UART . . . . .	30
2.3.2. SPI . . . . .	30
2.3.3. I2C . . . . .	31
2.4. Módulo ESP32 S3 . . . . .	32
2.4.1. Modos de funcionamiento . . . . .	33
2.5. VMTP . . . . .	34
2.6. Docker . . . . .	34
2.7. Thingsboard . . . . .	35
2.8. Trabajos relacionados . . . . .	36
<b>3. Materiales y métodos</b>	<b>40</b>
3.1. Diseño de la PCB . . . . .	40
3.1.1. Placa de sensores . . . . .	41
3.1.1.1. Características de los componentes utilizados . . . . .	41
3.1.1.2. Esquema eléctrico . . . . .	42
3.1.2. Placa de comunicación . . . . .	43

3.1.2.1. Características de los componentes utilizados . . . . .	43
3.1.2.2. Esquema eléctrico . . . . .	44
3.2. Adquisición de datos . . . . .	45
3.2.1. Señales analógicas . . . . .	46
3.2.1.1. Radiación solar . . . . .	46
3.2.1.2. Radiación UV . . . . .	47
3.2.1.3. Dirección de viento . . . . .	48
3.2.2. Señales digitales . . . . .	49
3.2.2.1. Temperatura y humedad . . . . .	49
3.2.2.2. Velocidad de viento . . . . .	49
3.2.2.3. Precipitación . . . . .	50
3.2.2.4. Módulo RTC . . . . .	51
3.3. Aplicación del protocolo VMTP . . . . .	51
3.3.1. Bandera de variación . . . . .	52
3.3.2. Marca de tiempo . . . . .	52
3.3.3. Bandera de nulos . . . . .	52
3.3.4. Bandera de ceros . . . . .	52
3.3.5. Variables hidrometeorológicas . . . . .	53
3.3.6. Carga útil de la trama . . . . .	53
3.4. Transmisión y gestión de datos . . . . .	53
3.4.1. Aplicación del protocolo Long Range Wide Area Network (LoRa- WAN) . . . . .	54
3.4.2. Diagrama de adquisición de datos y comunicación con TTS . . . . .	55
3.4.3. Envío de datos a Thingsboard . . . . .	55
3.5. Metodología de análisis de datos . . . . .	59
<b>4. Resultados y discusión . . . . .</b>	<b>61</b>
4.1. Módulos PCB de comunicación y de sensores en la adquisición de datos . . . . .	61
4.2. Consumo de corriente . . . . .	62
4.3. Pruebas del protocolo VMTP . . . . .	63
4.4. Comunicación con TTS usando LoRaWAN . . . . .	65
4.5. Verificación del decodificador . . . . .	67
4.6. Evaluación de Métricas de Comunicación LoRaWAN . . . . .	71

- 4.7. Análisis comparativo de los datos capturados por la estación desarrollada y la estación hidrometeorológica de la Microrred . . . . . 73
  - 4.7.1. Temperatura . . . . . 74
  - 4.7.2. Humedad . . . . . 75
  - 4.7.3. Radiación solar . . . . . 76
  - 4.7.4. Radiación UV . . . . . 77
  - 4.7.5. Dirección y velocidad del viento . . . . . 78
  - 4.7.6. Precipitación . . . . . 81
  
- 5. Conclusiones, recomendaciones y trabajos futuros . . . . . 83**
  - 5.1. Conclusiones . . . . . 83
  - 5.2. Recomendaciones . . . . . 84
  - 5.3. Trabajos futuros . . . . . 84
  
- Referencias . . . . . 86**
  
- A. Anexos . . . . . 92**

## Índice de figuras

1.1. Evolución del proyecto de adquisición de variables hidrometeorológicas del Programa para el Manejo del Agua y del Suelo (PROMAS) a lo largo del tiempo . . . . .	18
1.2. Esquema completo del funcionamiento del sistema. . . . .	22
2.1. Arquitectura de LoRaWAN [1]. . . . .	26
2.2. Diagrama de bloques de UART [2] . . . . .	30
2.3. Comunicación con varios esclavos usando Serial Peripheral Interface (SPI) [3]. . . . .	31
2.4. Trama VMTP . . . . .	34
2.5. Arquitectura Docker [4]. . . . .	35
2.6. Diagrama de bloques de la estructura de Thingsboard [5]. . . . .	36
3.1. Fases del proyecto . . . . .	40
3.2. Capa superior del PCB de conexión con los sensores . . . . .	43
3.3. Capa inferior del PCB de conexión con los sensores . . . . .	43
3.4. Capa superior del módulo de comunicación . . . . .	45
3.5. Capa inferior del módulo de comunicación . . . . .	45
3.6. Rosa de los vientos de 16 puntas [6]. . . . .	48
3.7. Carga útil de VMTP . . . . .	53
3.8. Diagrama de flujo del procedimiento de adquisición de datos y comunicación con TTS . . . . .	56
3.9. Diagrama de flujo del reenvío de datos de TTS a Thingsboard implementado en un contenedor Docker. . . . .	57
3.10. Diagrama de flujo del decodificador implementado en Thingsboard. . . . .	58
4.1. Lado superior del módulo de comunicación . . . . .	61
4.2. Lado inferior del módulo de comunicación . . . . .	61
4.3. Módulos PCB de comunicación y de sensores . . . . .	62
4.4. Trama VMTP recibida . . . . .	66
4.5. Proceso de inicialización en el servidor LoRaWAN . . . . .	66
4.6. Metadatos del paquete <i>uplink</i> . . . . .	67

4.7. Decodificación de variables VMTP correspondientes al <i>timestamp</i> , temperatura, humedad, radiación solar y radiación UV . . . . .	68
4.8. Decodificación de variables VMTP correspondientes a la dirección de viento, velocidad de viento y precipitación . . . . .	69
4.9. <i>Dashboard</i> que presenta un gráfico de las mediciones de temperatura, humedad, radiación solar, radiación UV, precipitación y velocidad de viento, a lo largo del tiempo . . . . .	70
4.10. Mediciones y mapeo de dirección de viento, según el transcurso del día	70
4.11. Representación de la bandera de nulos de VMTP . . . . .	71
4.12. Tasa de pérdida de paquetes en función de la distancia. . . . .	72
4.13. Intensidad de la señal recibida (RSSI) en función de la distancia. . . . .	72
4.14. Relación Señal a Ruido (SNR) en función de la distancia. . . . .	73
4.15. Temperatura de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	74
4.16. Correlación de la temperatura de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	74
4.17. Humedad de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	75
4.18. Correlación de la humedad de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	75
4.19. Radiación solar de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	77
4.20. Correlación de la radiación solar de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	77
4.21. Radiación UV de la estación implementada vs la estación de la Microrred.	78
4.22. Velocidad de viento de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	79
4.23. Correlación de la velocidad de viento de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	79
4.24. Medidas de dirección de viento de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	80

4.25. Correlación de la dirección del viento de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	80
4.26. Ubicación de la estación de la Microrred. . . . .	81
4.27. Ubicación del anemómetro de la estación implementada. . . . .	81
4.28. Cantidad de lluvia de la estación hidrometeorológica implementada vs la estación de Microrred. . . . .	82
4.29. Correlación de la cantidad de lluvia de la estación hidrometeorológica implementada vs la estación de la Microrred. . . . .	82
A.1. Esquema del módulo de comunicación. . . . .	92
A.2. Esquema del módulo de conexión de sensores. . . . .	93
A.3. Configuración de la Aplicación . . . . .	94
A.4. Configuración de frecuencia de operación y versión de LoRaWAN . . . . .	95
A.5. Configuración del método de activación, la clase de LoRaWAN y cluster. . . . .	96
A.6. Configuración de aprovisionamiento. . . . .	97
A.7. Ventana y lista de cadena de reglas. . . . .	98
A.8. Cadena de reglas PROMAS . . . . .	99
A.9. Ventana de pruebas de <i>script</i> decodificador. En la parte superior izquierda se ingresa la entrada al <i>script</i> , en la parte izquierda inferior se implementa el decodificador y su salida se muestra en la parte inferior derecha. . . . .	100

**Índice de tablas**

- 2.1. Descripción de variables hidrológicas y meteorológicas . . . . . 25
- 2.2. Resumen del formato de trama de LoRaWAN [7] . . . . . 29
  
- 3.1. Conexiones de los sensores en el sistema. . . . . 42
- 3.2. Configuración de pines entre el ESP32 S3 y el SX1262. . . . . 45
- 3.3. Interpretación del coeficiente de correlación de Pearson según las sugerencias de Cohen. Extraído de [8] . . . . . 60
  
- 4.1. Consumo de corriente usando distintas frecuencias de CPU y en modo de funcionamiento *Activo* y *Deep Sleep* . . . . . 62
- 4.2. Desglose de trama VMTP . . . . . 64
- 4.3. Total de agua recolectada por el pluviómetro de la estación implementada vs la estación de la Microrred . . . . . 82

**Dedicatoria**

Dedico este trabajo de titulación a mi madre, Carmen Tuza, cuyo apoyo incondicional ha sido mi mayor fortaleza en cada desafío. A mi padre, Leonardo Maldonado, le agradezco por haberme brindado la determinación para superar obstáculos. A mis hermanos, Erika Maldonado y Patricio Maldonado, les doy las gracias por su constante inspiración y apoyo.

**Henry Maldonado**

## Dedicatoria

A mi mamá, Yolanda, por dedicar todo su tiempo a apoyarme en alcanzar esta meta y por enseñarme a enfrentar los problemas y dificultades con valentía y responsabilidad. A mi padre, José, por ser un pilar sólido en el transcurso de mi vida académica. A mis hermanos, Alex, Vivi y José Luis, por estar junto a mí, compartiendo tristezas y llenando mi vida de alegrías, y por hacerme saber que siempre contaré con su apoyo. A mi novia, Valeria, porque sin ella mi vida habría tomado otro rumbo. Finalmente, a mí mismo, por mi resiliencia ante las dificultades en este largo camino.

**Ismael**

## Agradecimientos

Agradecemos al Ing. Fabián Astudillo, PhD, y al Ing. Juan Diego Belesaca, Mgtr, por su guía y apoyo durante la realización de este proyecto.

Extendemos nuestro agradecimiento al PROMAS por facilitarnos los sensores necesarios para la prueba e implementación del prototipo, especialmente al Ing. Óscar Morales por su disposición para proporcionarnos los requerimientos del proyecto.

También reconocemos al Laboratorio de la Microrred por permitirnos realizar las pruebas de la estación en sus instalaciones y por facilitarnos los datos recopilados de su estación para la validación del proyecto.

Finalmente, agradecemos a nuestros familiares por su apoyo incondicional a lo largo de nuestra carrera.

**Henry David Maldonado Tuza**

**Kevin Ismael Mogrovejo Nieves**

**Abreviaciones y acrónimos**

- ADC** Analog-Digital Converter. 2, 3, 22, 41, 42, 44–48, 53
- AES** Advanced Encryption Standard. 27, 29
- AT** Attention. 37
- CoAP** Constrained Application Protocol. 36
- CPU** Central Processing Unit. 62, 63, 83
- CS** Chip Select. 31, 44, 45
- CSV** Comma Separated Values. 39
- GFSK** Gaussian Frequency Shift Keying. 44
- GPIO** General Purpose Input/Output. 32, 42–46, 49, 50
- GPRS** General Packet Radio Service. 37
- GPS** Global Positioning System. 27, 37, 44
- GSM** Global System for Mobile Communications. 37
- HTTP** *Hypertext Transfer Protocol*. 36, 98
- HTTPS** *Hypertext Transfer Protocol Secure*. 54
- I2C** Inter-Integrated Circuit. 22, 29, 31, 32, 41–44, 46, 49, 50, 85
- IoT** *Internet of Things*. 25, 26, 32, 35, 38, 55, 94
- ISM** Industrial, Scientific and Medical. 24
- LCD** Liquid Cristal Display. 32, 37
- LDR** Light Dependent Resistor. 38
- LE** Low Energy. 32
- LoRa** Long Range. 2, 3, 19–27, 38–40, 44, 61, 83

**LoRaWAN** Long Range Wide Area Network. 2–5, 7, 9, 10, 20–27, 29, 34, 38–40, 43, 53–55, 61, 65, 66, 83, 84, 95, 96

**LPWAN** Low Power Wide Area Network. 25

**LwM2M** Lightweight. 36

**MAC** Media Access Control. 28, 29, 37

**MISO** Master In Slave Out. 31, 44, 45

**MOSI** Master Out Slave In. 31, 44, 45

**MQTT** Message Queuing Telemetry Transport. 36, 54, 55, 57, 98

**NSS** Not Slave Select. 45

**NTP** Network Time Protocol. 51

**OTAA** Over the Air Activation. 54, 96

**PCB** Printed Circuit Board. 2, 3, 7, 22, 23, 40, 41, 43, 44, 61, 62, 83, 85

**PDR** Packet Delivery Ratio. 60, 71

**PROMAS** Programa para el Manejo del Agua y del Suelo. 7, 9, 13, 17–20, 22, 23, 59, 84, 98, 99

**PSRAM** Pseudo SRAM. 32

**RSSI** Received Signal Strength Indicator. 60, 66, 71, 72

**RTC** Real Time Clock. 2, 3, 22, 33, 41, 42, 49, 51, 61, 62, 67

**SCL** System Clock. 31, 32

**SCLK** Serial Clock. 31, 44, 45

**SD** Secure Digital. 84

**SDA** System Data. 31

**SMS** Short Message Service. 38

**SNR** Signal-to-Noise Ratio. 60, 66, 71, 72

**SPI** Serial Peripheral Interface. 7, 29–32, 44, 45

**SRAM** Static Random-Access Memory. 32, 33

**SS** Slave Select. 31

**TTS** The Things Stack. 2, 3, 21, 22, 39, 40, 53–55, 57, 61, 65, 67, 83–85, 94, 96, 97

**UART** Universal Asynchronous Receiver-Transmitter. 29, 30, 37, 44, 84

**UDP** *User datagram protocol*. 27

**ULP** Ultra Low Power. 32, 33, 50

**VMTP** Variable Meteorological Transmission Protocol. 2, 3, 5, 7, 8, 10, 18, 21–24, 34, 39, 40, 48, 51–53, 57, 58, 61, 63, 64, 66, 68, 69, 71, 83, 85

**WSN** Wireless Sensor Networks. 19

## Introducción

Este capítulo muestra tópicos de importancia para la comprensión de este trabajo. Incluye la identificación del problema (Sección 1.1), antecedentes (Sección 1.2) justificación (Sección 1.3), alcance (Sección 1.4) y los objetivos del proyecto (Sección 1.5)

### 1.1. Identificación del problema

Programa para el Manejo del Agua y del Suelo (PROMAS) es un centro de investigación e innovación tecnológica que opera como una unidad desconcentrada y adscrita al Departamento de Ingeniería Civil de la Facultad de Ingeniería de la Universidad de Cuenca, que tiene encomendada la investigación científica y técnica, así como la Extensión y Consultoría en todos aquellos temas relacionados con el agua y el suelo considerados como bienes naturales, y agentes geodinámicos. Tiene como misión el ser un centro de innovación científica y desarrollo tecnológico de la Universidad de Cuenca, para contribuir al manejo sostenible del agua y del suelo mediante la generación de conocimiento científico, la capacitación y la extensión incorporando nueva tecnología y adaptando metodologías que permita dar soluciones integrales en equilibrio ambiental y con responsabilidad social [9].

La red de estaciones del PROMAS abarca una extensa área geográfica que se extiende desde el sector del Cajas, en la provincia de Azuay, hasta la provincia de Cañar. Estas estaciones son esenciales en la recopilación de datos y la investigación hidrometeorológica. Además, sirven para consultoría, monitoreo y planificación de la conservación de los recursos hídricos y suelo.

Cada una de las estaciones del PROMAS cuenta con varios sensores que miden diversas variables hidrometeorológicas, como temperatura, humedad, velocidad y dirección del viento, entre otras. Una vez recopilada la información, las estaciones la almacenan en memorias internas. Para recolectar estos datos, PROMAS moviliza personal para que realicen la descarga manual y trasladen la información al centro de datos del PROMAS en el campus central de la Universidad de Cuenca. Debido a la escasez de personal, la recolección y traslado de las mediciones se realiza en

un periodo de 30 a 45 días, o incluso más. Por esta razón, se han iniciado acciones que representan soluciones para llevar a cabo la recolección de información de forma inalámbrica.

En el proyecto “Aplicación de tecnologías inalámbricas al monitoreo climatológico en la cuenca del Río Paute” se implementó una red prototipo en el sector Loma Larga [10]. Este proyecto fue implementado usando dispositivos microcontroladores LoPy de la empresa Pycom. Las pruebas en esta red se las realizaron hasta finales del 2023. Sin embargo, se ha identificado una limitación importante en el uso del microcontrolador, la empresa que comercializaba los dispositivos LoPy ya no los está fabricando. Por lo que, la disponibilidad de este microcontrolador está experimentando una reducción gradual, y su documentación y versatilidad para realizar ajustes están en declive, lo que indica una inminente obsolescencia.

En la Figura 1.1 se resume la evolución del proyecto del PROMAS para la recolección de datos hidrometeorológicos, junto con sus principales características. El uso del módulo LoPy comenzó en 2019. Aunque el protocolo Variable Meteorological Transmission Protocol (VMTP) fue creado ese año, no se utilizó en el prototipo de 2020, que solo adquiriría datos pluviométricos. Sin embargo, se añadió la descarga de datos mediante WiFi. Los inconvenientes encontrados durante este proyecto del PROMAS han llevado a la necesidad de actualizar el prototipo de registro actualmente en uso.

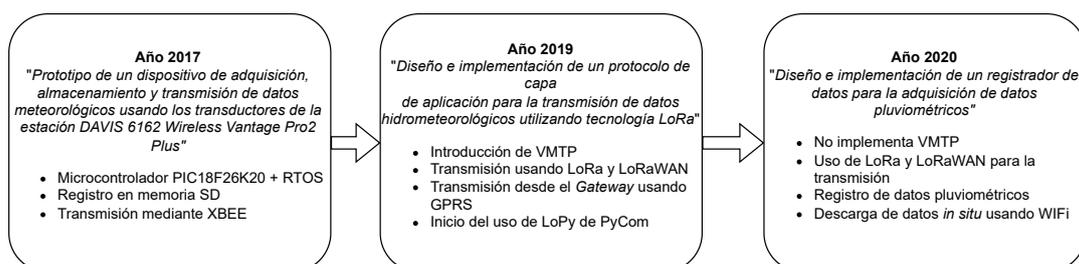


Figura 1.1: Evolución del proyecto de adquisición de variables hidrometeorológicas del PROMAS a lo largo del tiempo

Los desafíos específicos de la infraestructura de los nodos sensores del PROMAS, subrayan la necesidad de mejorar la gestión de datos para incrementar la flexibilidad en el procesamiento, la transferencia de información, y garantizar un funcionamiento

eficiente y fiable de los nodos sensores a largo plazo. Con el objetivo de respaldar la misión del PROMAS en la gestión sostenible de los recursos hídricos y manejo del suelo en la región.

## 1.2. Antecedentes

Las Wireless Sensor Networks (WSN) representan una revolución en la forma en que se recopila y usa datos en entornos distribuidos o de difícil acceso. Estas redes, compuestas por nodos interconectados, están diseñadas para capturar información de múltiples sensores distribuidos en un área extensa, permitiendo la monitorización en tiempo real de diferentes fenómenos [11]. La versatilidad y la capacidad de adaptación de las WSN las han convertido en una tecnología esencial en una amplia gama de aplicaciones, desde la agricultura, gestión ambiental, la seguridad, monitoreo industrial, entre otras.

En el ámbito de las estaciones meteorológicas, estas son fundamentales para recopilar datos atmosféricos y ambientales. Están equipadas con sensores para medir temperatura, humedad, presión, velocidad y dirección del viento, entre otros [12]. Al integrar estos sensores, proporcionan información crucial para sectores como la agricultura de precisión y la predicción de fenómenos climáticos extremos.

La Universidad de Cuenca, a través del PROMAS, está comprometida con la actualización y mejora constante de su infraestructura de monitoreo. Esta adaptación proactiva no solo optimiza la eficacia en la transmisión y la gestión de datos hidrometeorológicos, sino que también sienta las bases para futuras innovaciones en el monitoreo ambiental.

En un contexto más amplio, es esencial reconocer la necesidad de adaptación a los avances tecnológicos. La obsolescencia de hardware y tecnología limita significativamente la eficiencia y la relevancia de los sistemas de monitoreo. Por lo tanto, es necesario evolucionar y aprovechar nuevas soluciones tecnológicas para mantener la efectividad y precisión en la recolección de datos.

Por lo tanto, el presente trabajo de titulación, centrado en la implementación del microcontrolador ESP32 y la tecnología Long Range (LoRa), representa un paso importante

hacia una solución moderna y eficaz. Este avance mejorará la capacidad de recopilación y transmisión de datos. Además, garantizará el funcionamiento a largo plazo de las estaciones hidrometeorológicas del PROMAS. Esta iniciativa promueve la importancia de mantener al sistema de monitoreo del PROMAS al día con los avances tecnológicos para maximizar el impacto y la utilidad de las herramientas de monitoreo ambiental.

### 1.3. Justificación

En las primeras etapas, las estaciones meteorológicas utilizadas por el PROMAS requerían la recopilación manual de datos provenientes de diversos sensores. Esta operación implicaba que el personal se desplace periódicamente cada 30 o 45 días, lo que resultaba en una limitación logística significativa. Para superar este obstáculo, se implementó una primera solución que empleaba un microcontrolador PIC32 de Microchip con tecnología Zigbee para la recopilación y transmisión de datos de la estación hidrometeorológica [13].

Con el tiempo, se llevó a cabo una actualización sustancial de la infraestructura de las estaciones, adoptando un enfoque basado en el chip LoPy de Pycom. Este chip aprovecha el protocolo de comunicaciones LoRa de baja potencia y largo alcance [14]. Esta mejora significativa facilitó una transmisión de datos más eficiente y confiable, lo que a su vez fortaleció la operatividad y el alcance del PROMAS en su importante labor de monitoreo y conservación de los recursos naturales.

En este contexto, diversos proyectos han surgido con el objetivo de mejorar el sistema ya existente. Un ejemplo destacado es la investigación de K. S. Piedra [15], quien desarrolló un dispositivo capaz de recopilar datos limnigráficos mediante el uso de un microcontrolador ATMEL y la tecnología de red de área amplia de baja potencia Long Range Wide Area Network (LoRaWAN). Sin embargo, persiste la necesidad de abordar la recopilación y transmisión de datos provenientes de otros sensores que componen la estación hidrometeorológica.

Dada la obsolescencia y antigüedad del hardware en las estaciones hidrometeorológicas, surge la necesidad de migrar hacia una solución moderna. Nuestra propuesta

consiste en la implementación del microcontrolador ESP32 para la adquisición, almacenamiento y transmisión inalámbrica de datos a través de tecnología LoRa. Además, utilizar el modelo de trama del protocolo de capa de aplicación VMTP, el cual se ejecuta sobre LoRaWAN y es capaz de ajustar su carga útil en función de la información meteorológica que transporta, esto permite comprimir al máximo los datos enviados y reducir el consumo energético global del sistema. El uso de este protocolo en proyectos similares ha resultado de gran utilidad y ha mostrado ser muy confiable [16, 17].

La selección del módulo LoRa considera criterios como el bajo consumo de energía, alta ganancia, facilidad de integrar una antena externa, compatibilidad con estándares abiertos y costo. Tras una evaluación de las especificaciones técnicas y de rendimiento, se selecciona el módulo LoRa SX1262. La elección de este dispositivo se basa en su capacidad para sostener una potencia de transmisión de +22 dBm, con un bajo consumo energético, diseñados para cumplir con los requisitos de la capa física de la especificación LoRaWAN publicada por LoRa Alliance®. El objetivo final es cumplir con los requerimientos del proyecto, asegurando un rendimiento confiable y eficiente a lo largo del tiempo.

La adopción del módulo ESP32 tiene como objetivo principal reemplazar el sistema anterior de recolección de datos, proporcionando una capacidad de procesamiento, un bajo consumo energético y una comunicación similar al hardware previamente utilizado.

Para la recolección de datos, procesamiento de información y transmisión, se implementa el diagrama mostrado en la Figura 1.2, donde se lleva a cabo la captura de datos hidrometeorológicos como la temperatura, humedad, velocidad del viento, precipitación, nivel de luz solar y radiación UV utilizando el dispositivo ESP32. Este dispositivo emplea el protocolo LoRaWAN para transmitir las lecturas al *gateway* LoRa, el cual posteriormente envía los datos recibidos a un servidor local The Things Stack (TTS). En la capa de aplicación, el protocolo VMTP se encarga de un almacenamiento y transmisión eficientes, considerando tanto las variables actuales como las que se pueden medir en el futuro, y ofreciendo así la comunicación extremo a extremo deseada.

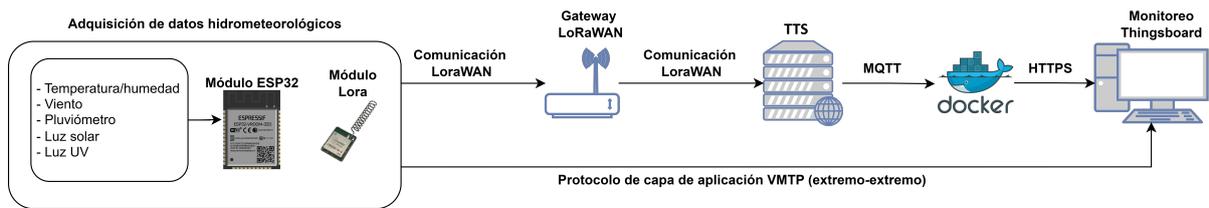


Figura 1.2: Esquema completo del funcionamiento del sistema.

#### 1.4. Alcance

Este proyecto propone el rediseño del dispositivo de registro de datos hidrometeorológicos del PROMAS. El rediseño implica la creación de una nueva placa de circuito impreso (Printed Circuit Board (PCB)) utilizando componentes de montaje superficial y orificio pasante.

Esta nueva placa PCB reemplaza el microcontrolador LoPy con un módulo ESP32, el cual se conecta al módulo LoRa SX1262 e implementa los protocolos LoRaWAN y VMTP para transmitir los datos recopilados hacia un *gateway*. El *gateway* posteriormente se encarga de enviar los datos al servidor TTS, que al comunicarse con la plataforma *Thingsboard* permite la visualización de los datos mediante un *dashboard*.

El rediseño incluye la modularización del dispositivo de registro hidrometeorológico. El primer módulo consta de una PCB que cuenta con los puertos y elementos necesarios para la comunicación Inter-Integrated Circuit (I2C) con el sensor de temperatura y humedad, el módulo Analog-Digital Converter (ADC), el módulo Real Time Clock (RTC) y el contador. El segundo módulo integra el dispositivo LoRa, el ESP32 y los pines necesarios para la comunicación entre módulos, con el objetivo de evitar el descarte total de las placas PCB en caso de requerir un cambio en el microprocesador o dispositivo LoRa.

Este trabajo de titulación contribuye a mejorar los sistemas de recolección de datos hidrometeorológicos de la Universidad de Cuenca, teniendo un impacto directo en el desarrollo de la investigación, la ciencia y la sociedad.

## 1.5. Objetivos

### 1.5.1. Objetivo general

Rediseñar e implementar un dispositivo de registro de datos hidrometeorológicos que integre la tecnología LoRa en un microcontrolador ESP32, con el objetivo de habilitar la transmisión eficiente de los datos recopilados hacia un centro de análisis central en el PROMAS.

### 1.5.2. Objetivos específicos

- Rediseñar y construir un PCB para un dispositivo registrador de datos usando componentes de montaje superficial para la adquisición de datos de una estación hidrometeorológica.
- Implementar el protocolo VMTP en el microcontrolador ESP32 para la transmisión de datos hidrometeorológicos usando LoRaWAN.
- Realizar pruebas de transmisión usando el dispositivo implementado.

## Marco teórico y trabajos relacionados

En este capítulo se presentan los conceptos fundamentales para el desarrollo del proyecto. Se comienza con la descripción de las variables hidrometeorológicas (Sección 2.1), la tecnología LoRa y su protocolo de largo alcance y bajo consumo, LoRaWAN (Sección 2.2) el cual es utilizado para la transmisión de los datos recolectados. A continuación, se detalla el módulo ESP32, que actúa como procesador central, implementando el protocolo VMTP. Finalmente, se revisan trabajos relacionados que aportan contexto y antecedentes a este proyecto.

### 2.1. Variables hidrometeorológicas

Las variables hidrometeorológicas surgen de la fusión de dos tipos de variables: hidrológicas y las meteorológicas, las cuales están directamente relacionadas con el ciclo del agua en la naturaleza [18]. Las variables hidrológicas se refieren a medidas relacionadas con el agua, como el nivel y caudal de los ríos, la cantidad de agua en embalses, la humedad del suelo, entre otros. Por otro lado, las variables meteorológicas se refieren a medidas relacionadas con el clima, como la temperatura del aire, la humedad relativa, la velocidad y dirección del viento, la precipitación, entre otras. Su alta variabilidad espacial y temporal [19] requiere un monitoreo constante para la correcta creación de pronósticos, balances hídricos, extrapolación de variables climáticas e investigación en general.

Basándonos en lo antes mencionado, es posible clasificar estas variables en dos grandes grupos [20, 21], dichos grupos, junto con sus elementos y conceptos, se describen en la Tabla 2.1.

### 2.2. LoRaWAN

LoRa es una tecnología de modulación de radio que permite la comunicación de largo alcance entre dispositivos a través de señales de radio de baja potencia y bajo *bit rate*. Utiliza bandas de frecuencia no licenciadas, como las frecuencias Industrial, Scientific and Medical (ISM) [22], dedicadas a fines científicos, industriales y experimentales.

Tabla 2.1: Descripción de variables hidrológicas y meteorológicas

<b>Categoría</b>	<b>Variable</b>	<b>Concepto</b>
Variables hidrológicas	Nivel del agua	La altura del agua en cuerpos de agua superficiales.
	Temperatura del agua	La medida de temperatura en el agua.
	Caudal	La cantidad de agua que fluye por unidad de tiempo en un río o arroyo.
Variables meteorológicas	Temperatura del aire	La medida de la temperatura en la atmósfera.
	Humedad relativa	La cantidad de humedad presente en el aire en relación con la cantidad máxima posible a esa temperatura.
	Velocidad y dirección del viento	La velocidad a la que se mueve el aire y la dirección desde la cual proviene.
	Precipitación	La cantidad de agua que cae sobre una superficie en forma de lluvia, nieve, granizo, etc.
	Presión atmosférica	La fuerza que ejerce la masa de aire sobre una unidad de superficie.
	Radiación solar	La cantidad de energía solar que llega a la superficie terrestre.

Esto permite una implementación sencilla y abarata costos. LoRa es conocida por su eficiencia energética, permitiendo que los dispositivos alimentados por batería, como sensores y dispositivos *Internet of Things* (IoT), operen durante largos períodos de tiempo sin necesidad de recarga o cambio de baterías.

Por su parte, LoRaWAN es un protocolo de red de área amplia y baja potencia Low Power Wide Area Network (LPWAN), que utiliza la tecnología de modulación LoRa para comunicar datos entre dispositivos y una red central. Es decir, corresponde a una capa de red y enlace de datos que indica a los dispositivos como usar la capa física LoRa [23].

La arquitectura de red de LoRaWAN se organiza en varias capas, como se muestra en la Figura 2.1, lo que permite la comunicación entre dispositivos IoT a larga distancia con un bajo consumo de energía. La arquitectura de LoRaWAN incluye una capa de aplicación, una capa de enlace, parámetros regionales y una capa física. En la capa física, se encuentran los dispositivos finales, como sensores y actuadores, que utili-

zan la tecnología LoRa para transmitir datos de manera inalámbrica. Estos datos son recibidos por los *gateways*, dispositivos ubicados estratégicamente que actúan como puntos de acceso a la red. Los *gateways* retransmiten los datos a la red de acceso LoRaWAN, donde se realizan tareas como la gestión de conexiones y el enrutamiento de mensajes. Finalmente, la red central de LoRaWAN supervisa y controla todo el sistema, incluyendo servidores de red y de aplicación que manejan la autenticación de dispositivos, el enrutamiento de mensajes y la gestión de datos. Esta arquitectura permite que los dispositivos IoT se comuniquen de forma efectiva y segura, facilitando la implementación de aplicaciones como la monitorización ambiental.

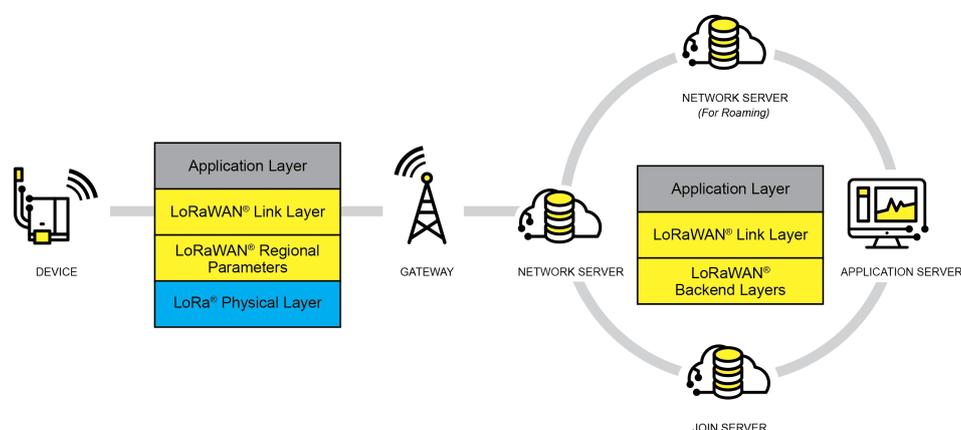


Figura 2.1: Arquitectura de LoRaWAN [1].

### 2.2.1. Características de LoRaWAN

Algunas de las características más importantes de LoRaWAN [23] son:

- **Consumo ultra bajo:** los dispositivos finales LoRaWAN están optimizados para funcionar en modo de bajo consumo y pueden durar hasta 10 años con una sola batería de tipo botón.
- **Largo alcance:** los *gateways* LoRaWAN pueden transmitir y recibir señales a una distancia de entre 10 y 20 kilómetros en áreas rurales y hasta 3 kilómetros en áreas urbanas densas [24].
- **Penetración interior profunda :** las redes LoRaWAN pueden proporcionar una cobertura interior profunda y cubrir fácilmente edificios de varios pisos.

- **Espectro sin licencia:** no tiene que pagar costosas tarifas de licencia de espectro de frecuencia para implementar una red LoRaWAN.
- **Geolocalización:** una red LoRaWAN puede determinar la ubicación de dispositivos finales mediante triangulación sin necesidad de Global Positioning System (GPS). Se puede localizar un dispositivo final LoRa si al menos tres *gateways* captan su señal [25].
- **Alta capacidad:** los servidores de red LoRaWAN manejan millones de mensajes desde miles de *gateways*.
- **Implementaciones públicas y privadas:** es fácil implementar redes LoRaWAN públicas y privadas utilizando el mismo hardware (*gateways*, dispositivos finales, antenas) y software (reenviadores de paquetes *User datagram protocol* (UDP), software de estación básica, pilas LoRaWAN para dispositivos finales).
- **Seguridad de extremo a extremo:** LoRaWAN garantiza una comunicación segura entre el dispositivo final y el servidor de aplicaciones mediante cifrado Advanced Encryption Standard (AES)-128 [1].
- **Actualizaciones de firmware por aire :** puede actualizar de forma remota el firmware (aplicaciones y la pila LoRaWAN) para un único dispositivo final o un grupo de dispositivos finales.
- **Roaming:** los dispositivos finales LoRaWAN pueden realizar traspasos fluidos de una red a otra.
- **Bajo costo:** infraestructura mínima, nodos finales de bajo costo y software de código abierto.

### 2.2.2. Formato de trama de LoRaWAN

La arquitectura de LoRaWAN presentados en la Figura 2.1 utiliza un formato de trama definido a través de tres capas. La capa física (PHY) es responsable de definir cómo se transmiten los datos, junto con las bandas de frecuencia, la potencia de transmisión, el ancho de banda, y otros parámetros técnicos específicos a cada región correspondientes a los parámetros regionales. Finalmente, la capa de enlace de

datos (Media Access Control (MAC)) y capa de aplicación [26].

La capa física tiene los siguientes segmentos:

■ **Capa física (PHY)**

- Preámbulo: Sincroniza al receptor con el transmisor e indica el esquema de modulación de la trama. La longitud típica del preámbulo es de 12.25 tiempos de símbolo,  $T_s$ .
- Encabezado (FHDR): Contiene información sobre la dirección de origen, la dirección de destino, el identificador de red, el control de acceso y el tamaño de la carga útil. El encabezado tiene una longitud de 13 bytes.
- CRC de encabezado (FHDR\_CRC): Es un código de redundancia cíclica (CRC) de 16 bits que se utiliza para detectar errores en el encabezado.
- Carga útil (PHY *Payload*): Contiene los datos que se envían. La carga útil puede tener una longitud máxima de 255 bytes.
- CRC de carga útil (FRMPayload\_CRC): Es un CRC de 16 bits que se utiliza para detectar errores en la carga útil (Opcional).

La trama MAC está encapsulada dentro de la carga útil (PHY *Payload*) y consta de los siguientes segmentos:

■ **Capa de enlace de datos (MAC)**

- Encabezado MAC (MHR): Contiene información sobre el tipo de mensaje, el identificador de dispositivo, el contador de tramas y los indicadores de control. El encabezado MAC tiene una longitud de 1 a 3 bytes.
- Carga útil MAC: Contiene los datos del mensaje MAC. La longitud de la carga útil MAC varía según el tipo de mensaje.
- MIC: Se crea al utilizar la clave de sesión de red en la información de encabezado y contenido, lo que añade seguridad para evitar que se envíen mensajes desde nodos no autorizados o la suplantación de identidad de otros.

La trama de aplicación no tiene un formato definido, debido a que la estructura de los

datos dependen de la aplicación específica. Sin embargo, existen algunos campos que son comunes en la mayoría de tramas:

#### ■ Capa de aplicación

- Encabezado Header: Sirve como identificador la red LoRaWAN del dispositivo, además de llevar un contador de mensajes y la secuencia, lo cual garantiza la correcta entrega de los mismos.
- Puerto Port: Se utiliza para identificar el canal de aplicación específico que se debe utilizar para enviar el mensaje. Depende del tipo de aplicación que utilicemos.
- Carga útil Payload: Contiene los datos reales de la aplicación que se quieren transmitir, la cual es cifrada usando la clave de sesión de aplicación (App\_Skey) usando el algoritmo AES-128.

Es importante destacar que, los campos Port y Payload son opcionales. Por lo tanto, una trama con un Header válido, pero sin Port y Payload, es posible y se considera una trama válida [27].

La Tabla 2.2 resume el formato de trama de LoRaWAN.

Tabla 2.2: Resumen del formato de trama de LoRaWAN [7]

Capa Física	Capa MAC	Capa de Aplicación
Preámbulo (12.25 símbolos)	MAC Header (1-3 bytes)	F. Header (7-22 bytes)
FHDR (13 bytes)	MAC Payload (t. variable)	F. Port
PHYPAYLOAD (max. 255 bytes)	MIC (4 bytes)	F. Payload
CRC (16 bits)		

### 2.3. Comunicación serial

La comunicación serial es un término general que se refiere a cualquier tipo de comunicación en la que los datos se envían de manera secuencial, a través de un canal de comunicación. Existen varios protocolos de comunicación como Universal Asynchronous Receiver-Transmitter (UART), Serial Peripheral Interface (SPI), I2C, RS-232, entre otros [28].

### 2.3.1. UART

El protocolo UART es un estándar de comunicación serial utilizado para la transmisión y recepción de datos entre dispositivos electrónicos. Este protocolo consta de tres bloques principales: transmisor, receptor y control de flujo, como se detalla en la Figura 2.2.

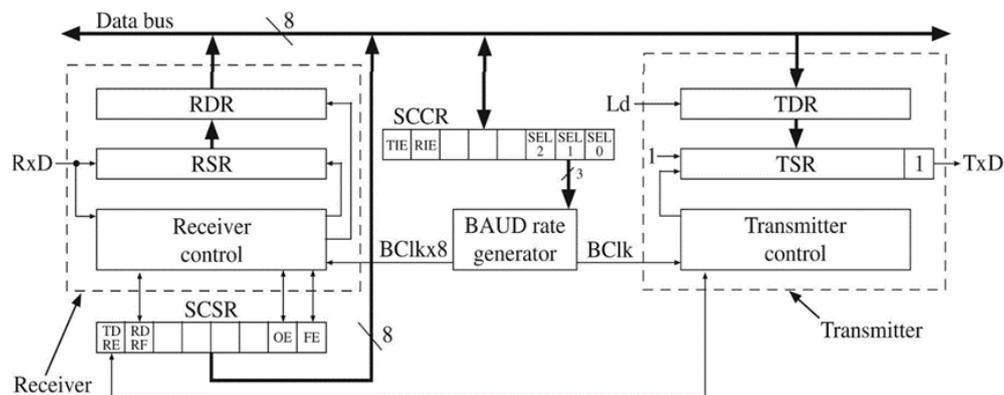


Figura 2.2: Diagrama de bloques de UART [2]

El transmisor envía datos desde el dispositivo que contiene el UART hacia otro dispositivo mediante un *buffer* de transmisión, un generador de bits de inicio, registros de desplazamiento y bits de parada. El receptor recibe datos desde otro dispositivo con UART, utilizando un *buffer* de recepción, un detector de bits de inicio, registros de desplazamiento, un detector de paridad y control de errores, y un temporizador de bits. El control de flujo gestiona el intercambio de datos entre el transmisor y el receptor para garantizar una comunicación fiable, incluso cuando operan a diferentes velocidades o protocolos. Estos bloques trabajan en conjunto para permitir una transmisión y recepción de datos eficiente y sincronizada. A pesar de la aparición de protocolos de comunicación más sofisticados, UART sigue siendo una parte importante de la electrónica moderna debido a su robustez y facilidad de implementación.

### 2.3.2. SPI

El protocolo SPI es un estándar de comunicación serial síncrona ampliamente utilizado en sistemas embebidos y aplicaciones de hardware para la transferencia de datos entre microcontroladores, sensores, periféricos y otros dispositivos digitales [29]. Este

estándar se compone de al menos un dispositivo maestro y uno o más dispositivos esclavos, donde el maestro controla la comunicación y selecciona los dispositivos esclavos con los que desea interactuar.

Como se ilustra en el diagrama de la Figura 2.3, la comunicación por medio del protocolo SPI se realiza a través de varias líneas de señal dedicadas, incluyendo Serial Clock (SCLK), Master Out Slave In (MOSI), Master In Slave Out (MISO) y Slave Select (SS)/Chip Select (CS). A través de estas líneas, el maestro envía datos a los esclavos y simultáneamente recibe datos de ellos. La velocidad de transferencia de datos y el modo de transmisión, que define la polaridad y la fase del reloj, pueden configurarse según las necesidades del sistema. La simplicidad y flexibilidad del protocolo SPI lo convierten en una opción popular para una amplia gama de aplicaciones que requieren una comunicación eficiente entre dispositivos digitales.

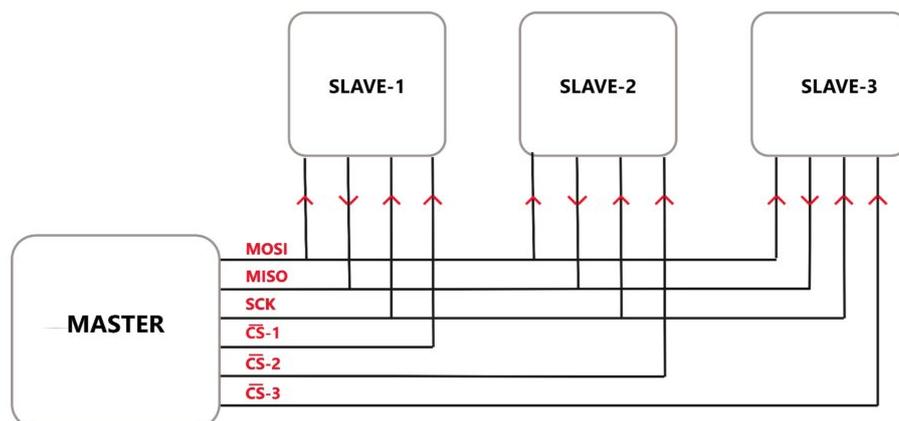


Figura 2.3: Comunicación con varios esclavos usando SPI [3].

### 2.3.3. I2C

El protocolo I2C es otro estándar de comunicación serial utilizado en sistemas embebidos y periféricos. A diferencia del protocolo SPI, I2C es una interfaz de comunicación serial síncrona de bus que permite la comunicación entre múltiples dispositivos utilizando dos líneas de señal: una línea de datos System Data (SDA) y una línea de reloj System Clock (SCL)[30].

En el protocolo I2C, los dispositivos están conectados en un bus compartido, donde

cada dispositivo tiene una dirección única que le permite ser identificado por otros dispositivos en el bus. La comunicación en el bus I2C se realiza en modo maestro/esclavo. El dispositivo maestro inicia y controla la comunicación, y uno o más dispositivos esclavos responden a las solicitudes del maestro [31].

Específicamente, el maestro inicia la comunicación enviando una señal de inicio seguida de la dirección del dispositivo esclavo con el que desea comunicarse y una señal indicando si está escribiendo o leyendo datos. Luego, los datos se transmiten secuencialmente entre el maestro y el esclavo, sincronizados por la línea de reloj SCL.

El protocolo I2C se utiliza comúnmente en una amplia gama de aplicaciones, como sensores, módulos de memoria, pantallas Liquid Cristal Display (LCD), entre otros. En general, en sistemas donde se necesita una comunicación eficiente entre componentes.

#### 2.4. Módulo ESP32 S3

El ESP32-S3 es una unidad de microcontrolador (Micro-Controller Unit (MCU)) basada en la arquitectura XTensa LX7 de doble núcleo, capaz de operar a frecuencias de hasta 240 MHz. Esta MCU destaca por sus avanzadas capacidades de procesamiento y eficiencia energética. Con 512 KB de Static Random-Access Memory (SRAM) interna, ofrece una conectividad inalámbrica integrada de 2.4 GHz, compatible con Wi-Fi 802.11 b/g/n y Bluetooth 5 Low Energy (LE), proporcionando un soporte ideal para aplicaciones de IoT. Además, incluye 45 pines (General Purpose Input/Output (GPIO)) programables y una amplia gama de periféricos. Su núcleo de ultra bajo consumo (Ultra Low Power (ULP)) permite múltiples modos de bajo consumo, prolongando significativamente los tiempos de funcionamiento en proyectos donde el ahorro energético es importante. El módulo ESP32-S3 también admite *flash* SPI octal de alta velocidad y Pseudo SRAM (PSRAM) con una mayor capacidad comparado con los dispositivos de la serie S2 y C3, configurables con caché de datos e instrucciones [32]. Debido a estas características y ventajas, se ha escogido el ESP32-S3 para nuestro proyecto sobre otras opciones disponibles.

### 2.4.1. Modos de funcionamiento

Los coprocesadores integrados de ULP permiten que el módulo funcione en modo de suspensión profunda, reduciendo así el consumo energético. Debido a que configurar la unidad de suministro de energía del módulo es un proceso complejo, se tienen predefinidos una serie de modos, que proporcionan distintas configuraciones para manejar los escenarios típicos de funcionamiento. A continuación se describen los modos predefinidos y sus características:

- **Active mode:** la CPU, los circuitos de RF y todos los periféricos están encendidos. El chip puede procesar datos, recibir, transmitir y escuchar
- **Modem-sleep mode:** la CPU está encendida, pero se puede reducir la frecuencia del reloj. Las conexiones inalámbricas se pueden configurar para permanecer activas, ya que los circuitos de RF se encienden periódicamente cuando sea necesario.
- **Light-sleep mode:** la CPU deja de funcionar y, opcionalmente, se puede encender. Los periféricos RTC, así como el coprocesador ULP, pueden activarse periódicamente mediante el temporizador. El chip se puede activar mediante todos los mecanismos de activación: MAC, temporizador RTC o interrupciones externas. Las conexiones inalámbricas pueden permanecer activas. Algunos grupos de periféricos digitales se pueden apagar opcionalmente.
- **Deep-sleep mode:** solo está encendido el RTC. Los datos de la conexión inalámbrica se almacenan en la memoria RTC.

La memoria RTC se refiere a la memoria que está asociada con el módulo del Reloj en Tiempo Real. Esta memoria es utilizada para almacenar datos que necesitan persistir cuando el dispositivo entra en modos de bajo consumo, y está mantenida por una fuente de energía separada que permite que los datos sean retenidos.

En el caso del ESP32-S3, la memoria RTC consiste en una memoria lenta y rápida tipo SRAM, cada una de 8 KB [33].

## 2.5. VMTP

VMTP es un protocolo de capa de aplicación diseñado para funcionar sobre LoRa-WAN. Este protocolo facilita la verificación del estado de los sensores y registra la hora y fecha de la toma de datos. Además, permite una reducción significativa en la cantidad de información que los sensores deben enviar desde la estación hidrometeorológica, lo que resulta en una disminución considerable del consumo energético del sistema, ideal para aplicaciones que tienen un limitado suministro de energía.

En la Figura 2.4 se detalla el modelo de trama VMTP. Esta se compone de un total de 104 bits, o 13 bytes, y contiene un total de 5 campos. El primer campo tiene el tamaño de 1 bit, corresponde a la bandera de variación. El segundo campo consta de 24 bits, y consiste en la marca de tiempo. Las banderas de nulos ocupan el tercer campo, con un total de 8 bits. El cuarto campo corresponde a las banderas de ceros, ocupando 4 bits. Lo cual deja un total de 63 bits para la carga útil, en donde se ubican las variables meteorológicas [16].

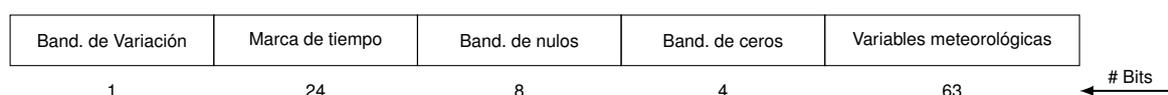


Figura 2.4: Trama VMTP

## 2.6. Docker

Docker es un entorno robusto de ejecución de contenedores. Los contenedores son una tecnología de virtualización que comparten el núcleo de un sistema operativo y contienen todas las dependencias necesarias para ejecutar una aplicación de software contenida. Es así que, Docker puede ser desplegado en cualquier sistema sin que sea necesario realizar configuraciones adicionales para que la aplicación funcione, ya que, las dependencias necesarias son empaquetadas con la aplicación en el contenedor [34].

En la Figura 2.5 se ilustra la arquitectura de Docker, la cual adopta un modelo cliente-servidor. En este modelo, el cliente Docker actúa como la interfaz principal con la que los usuarios interactúan. Mientras que, el servidor Docker se encarga de la gestión de

los objetos Docker, tales como contenedores, imágenes, volúmenes y redes.

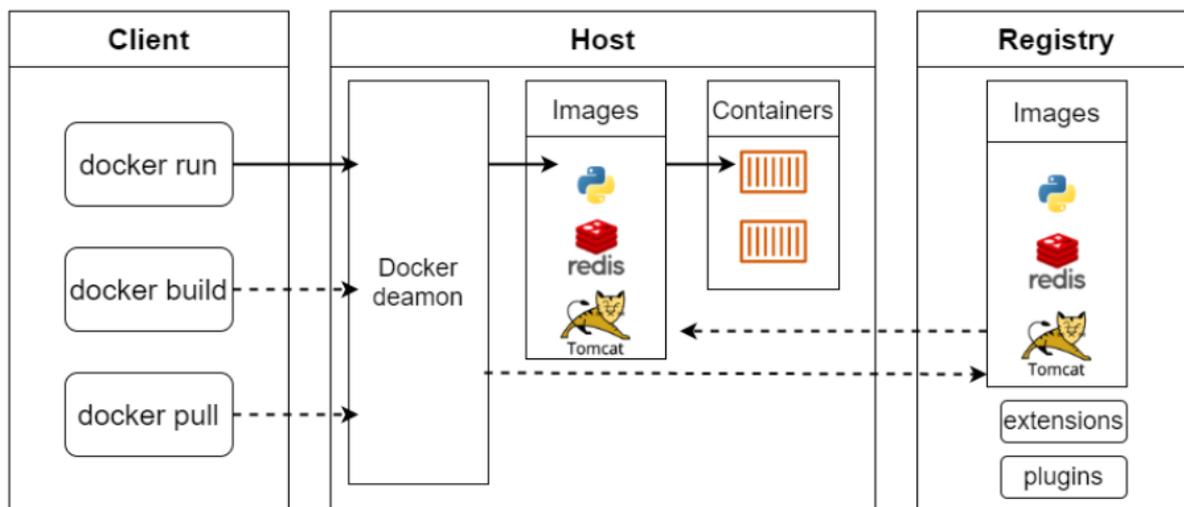


Figura 2.5: Arquitectura Docker [4].

Las imágenes Docker son plantillas de solo lectura que incluyen los códigos de la aplicación y todas las dependencias necesarias para su ejecución. Por otro lado, los contenedores, que son instancias en ejecución de una imagen Docker, están aislados, pero comparten un núcleo de ejecución común.

Los registros Docker, como Docker Hub, funcionan como repositorios de imágenes Docker. Docker Hub, en particular, es un registro público que alberga una amplia colección de imágenes disponibles para su descarga y uso.

Los volúmenes Docker se utilizan para gestionar la persistencia de los datos en los contenedores, asegurando que los datos persistan más allá del ciclo de vida de los contenedores. Finalmente, las redes Docker facilitan la conexión entre contenedores y el sistema *host* mediante puentes, redes superpuestas y redes *host*, proporcionando una infraestructura flexible para la comunicación entre contenedores.

## 2.7. Thingsboard

ThingsBoard es una plataforma de IoT de código abierto que permite un rápido desarrollo, gestión y escalado de proyectos de IoT [5]. Permite soluciones tanto en la nube como en instalaciones locales, que habilita la infraestructura para distintas aplicaciones IoT. Su arquitectura se presenta en la Figura 2.6 y sus características principales

son:

- Integración de varios microservicios que permiten conectarse con distintos servidores de transporte como son: *Hypertext Transfer Protocol* (HTTP), *Message Queuing Telemetry Transport* (MQTT), *Constrained Application Protocol* (CoAP), *Lightweight* (LwM2M).
- Aprovisionamiento de dispositivos, clientes y activos, además de recopilación y visualización de sus datos.
- Flujos de trabajo basados en un evento del ciclo de vida del dispositivo.
- Incluye microservicios de interfaz de usuario web.

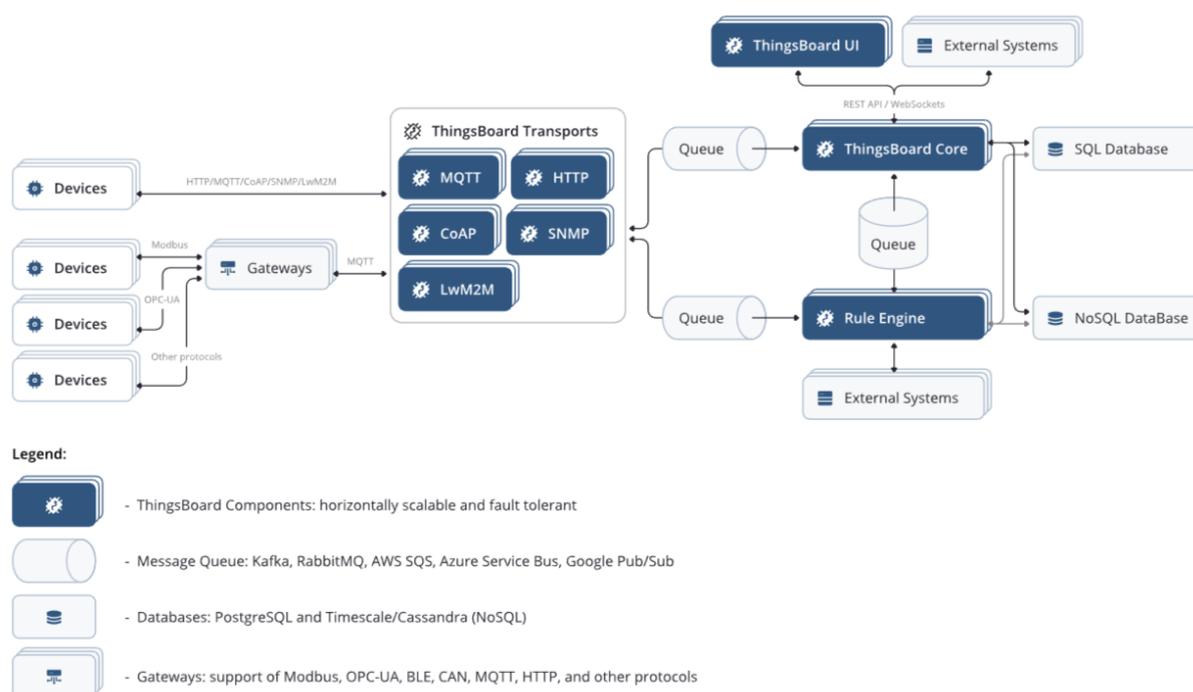


Figura 2.6: Diagrama de bloques de la estructura de Thingsboard [5].

## 2.8. Trabajos relacionados

En el contexto de investigaciones similares, se han llevado a cabo despliegues que hacen uso de una variedad de componentes y tecnologías para estaciones meteorológicas. Algunas de las alternativas más comunes abarcan el empleo de microcontroladores como Arduino, Raspberry Pi y ESP8266/ESP32, en conjunto con diversos

tipos de sensores para la medición de variables climáticas como temperatura, humedad, presión atmosférica, velocidad y dirección del viento, precipitación, entre muchos otros.

En este contexto, en el trabajo [35], los autores implementan una estación meteorológica en Bagdad-Irak, dotada de sensores para medir parámetros clave como temperatura, humedad, velocidad del viento, presión, precipitación y polvo. La comunicación entre el nodo central y los nodos secundarios se realiza de manera inalámbrica a través de la tecnología Zigbee, basada en el estándar MAC/PHY IEEE 802.15.4. Esto permite establecer una comunicación en un rango que abarca desde 120 metros hasta 64 kilómetros. Tanto el nodo central como nodos secundarios están equipados con microcontroladores Arduino Uno. La implementación mostró una precisión notable al medir parámetros climáticos. La comparación de datos con el sitio web *The Weather Channel* [36] reveló una discrepancia mínima, validando la exactitud de las mediciones.

El proyecto presentado en [37] se distingue por el uso de un microcontrolador PIC18F 45K22. En este proyecto se detalla el funcionamiento principal de una estación meteorológica inteligente que se destaca por el uso de la plataforma en la nube ThingSpeak para la gestión y análisis de los datos recopilados. El proyecto incluye una amplia gama de componentes y sensores como: sensor de temperatura y presión atmosférica (BMP180), sensor de humedad, pantalla LCD, módulo GPS, módulo General Packet Radio Service (GPRS), módulo Ethernet, batería recargable, panel solar y el módulo de telefonía celular SIM8001. Además, integran un circuito de alimentación y otro de carga de batería en el sistema. La comunicación se realiza de forma inalámbrica a través del módulo SIM8001. Por otra parte, el módulo Global System for Mobile Communications (GSM) y GPRS se encargan de enviar los datos a la plataforma ThingSpeak en la nube. Estos módulos también son utilizados para notificar al usuario mediante un mensaje de texto cuando se ha completado la medición. La comunicación entre este módulo y el microcontrolador se lleva a cabo a través del primer módulo UART del microcontrolador, utilizando comandos Attention (AT).

El módulo ESP también ha sido utilizado en estaciones meteorológicas, un ejemplo se presenta en [12]. En este trabajo usan específicamente el módulo NodeMCU (12E),

como núcleo central del dispositivo. Este módulo proporciona la plataforma para IoT y actúa como intermediario entre los diversos sensores y la nube. Cuatro tipos de sensores son utilizados; el DHT11 para medir temperatura y humedad, el BMP180 para la presión barométrica, el sensor Light Dependent Resistor (LDR) para la intensidad lumínica y un módulo de detección de lluvia. Estos sensores están conectados al NodeMCU, permitiendo la recopilación de datos y su posterior carga a la nube a través de Wi-Fi. Además, utilizan la plataforma IBM Bluemix que se encarga del análisis de los valores recopilados. Cuando se superan ciertos umbrales predefinidos, se envían alertas al propietario del dispositivo a través de correo electrónico, Short Message Service (SMS) y publicaciones en redes sociales.

En el Ecuador, se han implementado diversas aplicaciones de la tecnología LoRa en el ámbito de alertas hidrometeorológicas [13–15]. Dada la similitud en la recopilación de datos entre sensores, la mayoría de estas implementaciones se centran en la recopilación de datos de un solo sensor. Un ejemplo de ello es el proyecto presentado en [38]. Este proyecto tiene como objetivo prever posibles inundaciones en la parroquia Alangasí. Para lograrlo, se construyó un pluviómetro cuyos datos son procesados por un microprocesador y transmitidos a través de un *Gateway* LoRa implementado en una Raspberry Pi 3. Los datos recopilados son enviados por correo electrónico cuando se superan ciertos límites, con el propósito de mantener los sistemas de alerta temprana.

En el proyecto [39], se crea un dispositivo de adquisición, almacenamiento y transmisión de datos hidrometeorológicos para la Universidad de Cuenca. El procesamiento de señales se lo hace usando el microcontrolador PIC18F26K20. La transmisión inalámbrica se realiza con un módulo Xbee. Este desarrollo permite reducir costos en la recopilación de datos, transmitiendo automáticamente la información recopilada por las estaciones hidrometeorológicas.

Del mismo modo, en el proyecto [15], los autores presentan un diseño e implementación de registro para la adquisición de datos pluviométricos a través del uso del microcontrolador ATMELE. La tecnología LoRa se integra para posibilitar la transmisión inalámbrica de datos, permitiendo la obtención remota de información. El protocolo usado es LoRaWAN, que establece la comunicación entre nodos, sensores, *gateways*

y servidores, asegurando una transmisión de datos fiable. Adicionalmente, se desarrolla una aplicación móvil nativa en JAVA para dispositivos Android en el marco del proyecto. Esta aplicación habilita la búsqueda y visualización de datos almacenados, así como su descarga en formato CSV.

Con base en lo antes mencionado, se evidencian implementaciones de la tecnología LoRa para la recopilación de datos hidrometeorológicos a nivel mundial, incluyendo casos específicos en Ecuador y en la Universidad de Cuenca. Estas implementaciones han mostrado la confiabilidad de LoRa en diversos ámbitos de aplicación, allanando el camino para nuevas posibilidades en función de los beneficios deseados. En este sentido, se opta por centrar este trabajo de titulación en el uso de esta tecnología, principalmente por su capacidad para mantenerse actualizadas a lo largo del tiempo, evitando la obsolescencia en un corto plazo.

En este trabajo de titulación, se propone un sistema diseñado para la captura de datos hidrometeorológicos. El sistema recopila información detallada que incluye temperatura, humedad, velocidad y dirección del viento, precipitación, radiación solar y radiación UV. Los datos de los sensores son procesados por el módulo ESP32-S3, el cual utiliza los protocolos VMTP y LoRaWAN. Mediante el módulo LoRa SX1262, los datos son enviados al *gateway*. Posteriormente, son transmitidos al servidor TTS de la Universidad de Cuenca y, finalmente, a Thingsboard. La información recopilada se visualiza y gráfica en un *dashboard*, facilitando así la monitorización en tiempo real de las variables hidrometeorológicas.

## Materiales y métodos

En este capítulo se describe la implementación del proyecto, abarcando desde el diseño y la fabricación de las placas PCBs para la comunicación LoRa hasta la conexión con los sensores. Se detallan los componentes y sensores utilizados, la creación de la trama VMTP, la configuración del protocolo de comunicación LoRaWAN, el envío de datos y su conexión con el servidor TTS. Finalmente, se explica el procedimiento para permitir la visualización de los datos recibidos en un *dashboard*. El proceso descrito se detalla en el diagrama de la Figura 3.1.

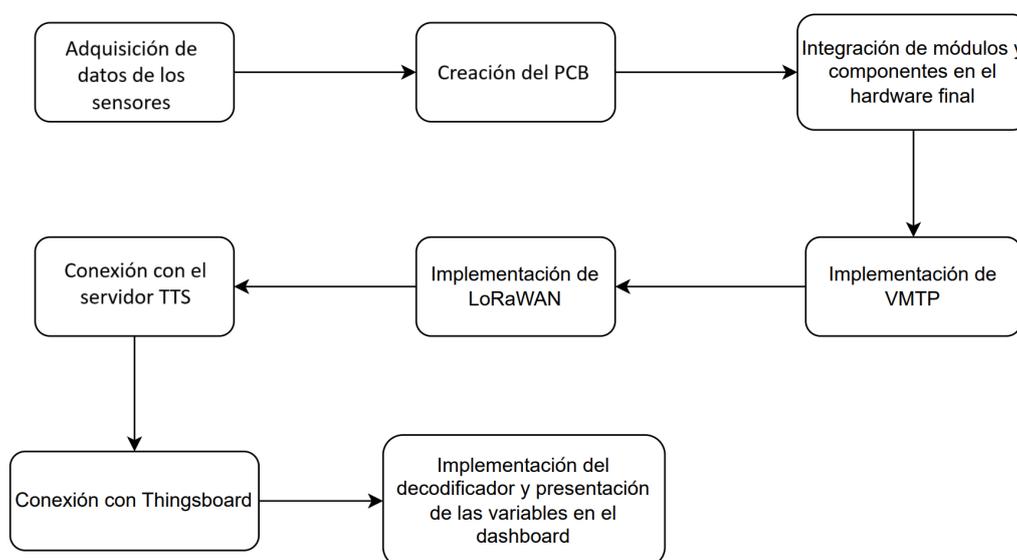


Figura 3.1: Fases del proyecto

### 3.1. Diseño de la PCB

La estación meteorológica diseñada consta de dos módulos, uno para la conectividad de los sensores y otro para la comunicación y procesamiento de lecturas. Para el diseño de ambas PCB se usa la herramienta EasyEDA, que es una herramienta gratuita, no requiere instalación y además está basada en la nube, por lo que se pueden crear directamente diseños de manera online [40]. En las siguientes secciones se detallan los componentes usados en el montaje de cada uno de los módulos y las características técnicas consideradas en el diseño de los PCB.

### 3.1.1. Placa de sensores

La placa de sensores está diseñada para la conectividad de diversos sensores, gestionando la lectura de los mismos. Esta sección describe los componentes utilizados, las características técnicas y la disposición de las capas de la placa de sensores.

#### 3.1.1.1. Características de los componentes utilizados

El PCB diseñado para la adquisición de datos incluye los siguientes componentes:

- ADC (ADS1115IDGSR): Este ADC es un dispositivo de precisión, bajo consumo y 16 bits, compatible con I2C. Realiza conversiones a tasas de datos de hasta 860 muestras por segundo y ofrece rangos de entrada desde  $\pm 256$  mV hasta  $\pm 6.144$  V [41].
- Contador (S-35770E01I-K8T2U): Es un contador permite contar señales de reloj externas con una precisión de 24 bits, capaz de contar desde 0 hasta 16,777,215. Las lecturas del contador puede ser leída a través de una interfaz serial I2C. Entre sus características destacan el bajo consumo de corriente de  $0.01 \mu A$  típico (a 3.0 V, 25 °C, sin comunicación), amplio rango de voltaje de operación (1.5 V a 5.5 V), y rango de temperatura de operación de -40 °C a +85 °C [42].
- RTC (DS3231S#T&R): El DS3231 es un reloj en tiempo real extremadamente preciso y de bajo costo. Mantiene información de segundos, minutos, horas, día, fecha, mes y año, ajustando automáticamente los meses con menos de 31 días y corrigiendo los años bisiestos. Opera en formato de 24 horas o en formato de 12 horas con indicador AM/PM. Incluye una referencia de voltaje compensada por temperatura, un circuito comparador para detectar fallos de alimentación y cambiar automáticamente a la batería de respaldo cuando sea necesario. Se comunica a través de un bus bidireccional I2C rápido (400 kHz) [43].
- Controlador de alimentación (MIC2545A-2YM-TR): El MIC2545A es un interruptor de alimentación, optimizado para aplicaciones de administración de energía. Entre sus características principales se incluyen: entrada de 2.7 V a 5.5 V, límite de corriente ajustable hasta 3 A, bloqueo de flujo de corriente inversa, corriente

de suministro en estado activo de  $90 \mu A$  típica, corriente de suministro en estado inactivo de  $1 \mu A$  típica, resistencia de encendido máxima de  $50 m\Omega$ , señal de fallo de drenaje abierto, apagado térmico y arranque suave de 2 ms (lento) y apagado rápido [44].

### 3.1.1.2. Esquema eléctrico

Previo a definir el esquema eléctrico, se establece las conexiones de cada componente de la placa de sensores. El ADC se encarga de obtener las lecturas de los sensores de radiación UV, sensor de radiación solar y dirección del viento. Mientras que el contador guarda la cantidad de pulsos generados por el pluviómetro de tipo balancín. Finalmente, el RTC, el sensor de temperatura y humedad se conectan a las líneas de comunicación I2C, lo que permite la lectura de datos de los sensores y el control de los distintos componentes. En la Tabla 3.1 se sintetiza las conexiones de los distintos sensores implementados.

Tabla 3.1: Conexiones de los sensores en el sistema.

Variable Meteorológica	Conexión	Descripción
Radiación UV	IN0 del ADC	Mide la radiación ultravioleta presente en el entorno. La señal analógica de salida del sensor ingresa al ADC.
Radiación Solar	IN1 del ADC	Mide la radiación solar incidente. La señal analógica de salida del sensor ingresa al ADC.
Dirección del Viento	IN3 del ADC	Determina la dirección del viento. La señal analógica de salida del sensor ingresa al ADC.
Precipitación (Pluviómetro)	Pin CLKin del Contador	Los pulsos generados por el pluviómetro tipo balancín son detectados y almacenados en el contador.
Temperatura y Humedad	Línea ADC	Mide la temperatura y la humedad del ambiente.
Velocidad del Viento	GPIO	Mide la velocidad del viento mediante pulsos generados por el anemómetro que ingresan directamente al ESP32 mediante un GPIO.

El PCB cuenta con conectores THT para sensores de radiación solar, UV, temperatura, precipitación, velocidad y dirección del viento. Además, se incluyó entradas para comunicación con el bus I2C y conexiones adicionales a los GPIO del ESP32 en el módulo de comunicación.

El diseño esquemático con las conexiones de los componentes de la placa de sensores se presenta a detalle en la Figura A.2. En general, las dimensiones del PCB es de 88.9 mm por 61.092 mm, el grosor de pistas para la alimentación corresponden a 0.5 mm (21 mil) y 0.254 mm (10 mil) para el resto de pistas.

En la Figura 3.2 y 3.3 se presenta la capa superior e inferior del PCB respectivamente, en vista 2D de la placa de conexión de los sensores.

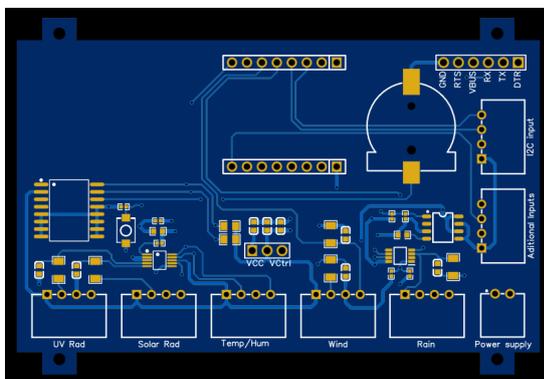


Figura 3.2: Capa superior del PCB de conexión con los sensores

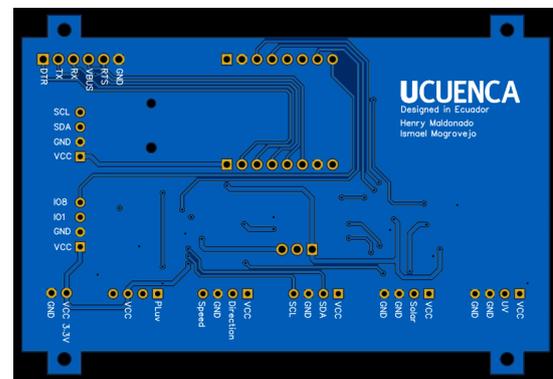


Figura 3.3: Capa inferior del PCB de conexión con los sensores

### 3.1.2. Placa de comunicación

La placa de comunicación está diseñada para gestionar la transmisión de datos mediante LoRaWAN y la comunicación con diversos sensores. Esta sección describe los componentes utilizados, las características técnicas y la disposición de las capas de la placa de comunicación.

#### 3.1.2.1. Características de los componentes utilizados

- ESP32-S3:** El ESP32-S3 ha sido seleccionado debido a su reciente lanzamiento en el año 2022, y la creciente comunidad de desarrollo, que garantiza soporte y actualizaciones continuas. Este microcontrolador ofrece una gran cantidad de

GPIOs, lo que permite su uso en diversas aplicaciones. Además, cuenta con varios puertos UART, SPI, I2C, entre otros, lo que facilita la conexión con módulos GPS y otros dispositivos. Una característica destacable del ESP32-S3 es que muchos de sus puertos SPI, UART e I2C pueden mapearse a diferentes pines [33]. Esto significa que no es necesario utilizar pines específicos para estas interfaces, lo que brinda una mayor flexibilidad en el diseño de hardware.

- **Lora Core SX1262:** Para la comunicación se ha seleccionado el módulo LoRa Core SX1262, el cual tiene una potencia de transmisión de 22 dBm. Proporciona un *data rate* en modulación LoRa de hasta 62.5 Kbps y en modo Gaussian Frequency Shift Keying (GFSK) de hasta 300 Kbps. Este módulo requiere una alimentación en el rango de 1.8 V a 3.7 V. La comunicación entre el ESP32-S3 y el módulo LoRa se realiza mediante el protocolo SPI. Por lo tanto, se realizó la conexión entre los pines MISO, MOSI, SCLK y CS correspondientes.

### 3.1.2.2. Esquema eléctrico

En la Figura A.1 se presenta el esquemático con las conexiones realizadas para la placa de comunicación. En este esquema, el ESP32-S3 y el módulo *LoRa* SX1262 están conectados mediante comunicación SPI. Además, se han considerado entradas para comunicación I2C, utilizadas para la conexión con el ADC que captura las lecturas de los sensores de radiación solar, índice de radiación UV y dirección del viento. También se han utilizado GPIO para la lectura del sensor de velocidad del viento, el control del contador del pluviómetro y la conexión del módulo de programación del ESP32-S3.

Las dimensiones del PCB desarrollado son de 48.768 mm × 20.828 mm. El ancho de las pistas para alimentación es de 3.3 V de 0.5 mm (21mil), mientras que, para el resto de conexiones es de 0.254 mm (10mil), de acuerdo a las recomendaciones de diseño de hardware proporcionadas por el fabricante [45].

Con respecto a la disposición de los componentes de la PCB. El módulo SX1262 se ubicó en la capa superior, mientras que el ESP32-S3 se ubicó en la capa inferior del PCB. Esta decisión se fundamenta en la optimización del espacio en la placa. Además, la ubicación del módulo LoRa en la parte superior facilita el acceso para la

conexión de la antena.

El ESP32-S3 y el módulo LoRa SX1262 están conectados mediante comunicación SPI y la configuración de pines utilizada se presenta en la Tabla 3.2.

Tabla 3.2: Configuración de pines entre el ESP32 S3 y el SX1262.

ESP32 S3	SX1262	Descripción
GPIO 12	SCLK	Señal de reloj
GPIO 11	MOSI	Transporta los datos enviados desde el ESP32 ( <i>Master</i> ) al SX1262 ( <i>Slave</i> ).
GPIO 13	MISO	Transporta los datos enviados desde el SX1262 ( <i>Slave</i> ) al ESP32 ( <i>Master</i> ).
GPIO 10	Not Slave Select (NSS) (CS)	Este pin es utilizado por el ESP32 para habilitar la comunicación con el SX1262. Se activa (baja) para iniciar la comunicación SPI.
GPIO 7	DIO1	Salida de interrupción en el SX1262 para notificar eventos importantes al ESP32, como la recepción de un paquete.
GPIO 14	<i>RESET</i>	Utilizado para reiniciar el SX1262.
GPIO 15	<i>BUSY</i>	Indica si el SX1262 está ocupado procesando una tarea.

En las Figuras 3.4 y 3.5 se muestra la capa superior e inferior en 2D del módulo de comunicación.

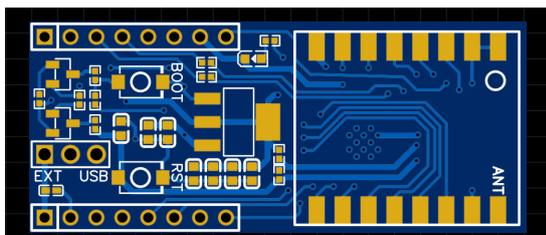


Figura 3.4: Capa superior del módulo de comunicación

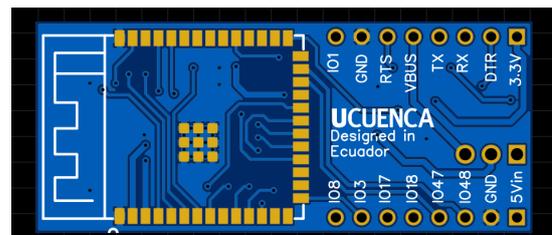


Figura 3.5: Capa inferior del módulo de comunicación

### 3.2. Adquisición de datos

Las señales adquiridas por la estación hidrometeorológica implementada puede categorizarse en dos tipos: señales analógicas y señales digitales. Los sensores que proporcionan a su salida señales analógicas, requieren de una conversión mediante un ADC. Mientras que, las señales digitales se manejan mediante puertos GPIO. La

lectura de datos de estos sensores se realiza, en su mayoría, mediante comunicación I2C, como variables de temperatura y humedad, además los sensores conectados al ADC y al contador de pulsos S-35770. No obstante, la medición del sensor de la velocidad de viento se realiza mediante un puerto GPIO.

Con el fin de eliminar ruidos e interferencias debido al funcionamiento de los sensores, se realiza un acondicionamiento de las señales de cada uno de los sensores, para ello, se usa el trabajo presentado en [13].

### 3.2.1. Señales analógicas

La estación se compone de distintos sensores analógicos, entre los que se encuentra el sensor de radiación solar, radiación UV y dirección de viento. La salida de dichos sensores ingresan al módulo ADC, el cual se encarga de integrar los datos de todos los sensores y enviar la información al microprocesador ESP32-S3 usando el protocolo I2C.

#### 3.2.1.1. Radiación solar

Para la medición de la radiación solar se usa el piranómetro 6450 de *Davis Instruments*. Basado en la información presentada en su hoja de datos [46], es posible transformar su salida analógica, que se encuentra en un rango de 0 a 3 V a la unidad correspondiente,  $W/m^2$ . La transformación se efectúa usando la relación lineal entre la salida del sensor y el valor de radiación solar. Es decir, la salida incrementa 1.67 mV por cada  $W/m^2$ , con un rango de medición de 0 a  $1796.4 W/m^2$ .

Dado que las variaciones en las mediciones del sensor de radiación solar son bajas, el módulo ADC debe ser capaz de percibir estas mínimas variaciones. Para determinar la resolución del módulo ( $Res$ ) se usó el siguiente procedimiento:

$$\ln(Res) = \ln(V_{ref}) - n_{bits} * \ln(2)$$

$$Res = e^{\ln(V_{ref}) - n_{bits} * \ln(2)}$$

$$Res = e^{\ln(3.3) - 16 * \ln(2)}$$

$$Res = e^{-9.896} = 0.05 \text{ mV}$$

Donde,  $V_{ref}$  es el voltaje de referencia o alimentación (3.3 V) y  $n_{bits}$  es el número de bits para representación, que para este caso es 16. Con estos valores se obtuvo que con una resolución de 0.05 mV es posible representar una lectura de 1.65 mV, un valor muy cercano al proporcionado por el sensor.

Para obtener el valor de radiación solar ( $R.Sol$ ) se utilizó la siguiente transformación [46]:

$$R.Sol = \frac{ADC1}{0.00167} \quad (3.1)$$

Donde:

- R.Sol es la radiación solar en  $W/m^2$ .
- ADC1 es la salida en voltios proporcionada por el ADC, resultado de la lectura de la entrada correspondiente del sensor.

### 3.2.1.2. Radiación UV

Para lectura de índice de radiación UV se usa el sensor UV 6490 de *Davis Instruments*. El sensor proporciona una salida en el rango de 0 a 2.5 V con una resolución de 0.1 de índice de radiación [47]. La relación de transformación es que por cada unidad de índice UV, la salida del sensor incrementa 150 mV, lo que otorga un rango de medición de 0 a 16.

Entonces, la transformación del valor analógico a índice UV se la realiza de la siguiente expresión [47]:

$$I.UV = \frac{ADC0}{0.150} \quad (3.2)$$

Donde:

- I.UV es el índice UV resultante.
- ADC0 es la salida en voltios proporcionada por el ADC, resultado de la lectura de la entrada correspondiente del sensor, en este caso, la primera.

### 3.2.1.3. Dirección de viento

El sensor de dirección de viento 6410 de *Davis Instruments* brinda una salida de 0 V a 3.3 V según la dirección en la que apunte la veleta [48]. Realizando la conversión, obtenemos que por cada grado de variación, el voltaje a su salida varía en 9.16 mV. Para la conversión de voltaje a dirección en grados se utiliza la siguiente expresión:

$$D^{\circ} = \frac{ADC2}{360^{\circ}} \quad (3.3)$$

Donde:

- $D^{\circ}$  es la dirección en grados.
- $ADC2$  es la salida en voltios proporcionada por el ADC, resultado de la lectura de la entrada correspondiente del sensor, en este caso, la tercera.

Es importante destacar que, la trama VMTP tiene un campo para dirección de viento, el cual ocupa 4 bits, lo que significa que la dirección de viento solo puede ser representada por 16 valores. Los 16 valores corresponden a las direcciones de la rosa de vientos mostrada en la Figura 3.6.

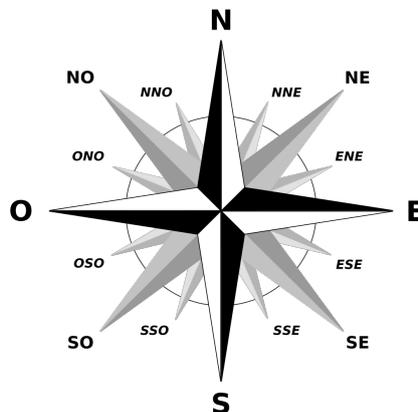


Figura 3.6: Rosa de los vientos de 16 puntas [6].

Por lo que, el valor de la dirección en grados de  $0^{\circ}$  a  $360^{\circ}$  se escala a un valor de 0 a 16, respectivamente. Entonces, cada dirección de la rosa de vientos representa una variación de  $22.5^{\circ}$ .

### 3.2.2. Señales digitales

La estación utiliza sensores que proporcionan una señal digital a su salida como; las señales de temperatura y humedad provienen del sensor AHT20 e ingresan al ESP32-S3 mediante el protocolo I2C, similar a la comunicación establecida con el contador de precipitación y con el módulo RTC. Por su parte, el sensor de velocidad del viento proporciona pulsos a su salida, que ingresan a un pin GPIO del ESP32-S3.

#### 3.2.2.1. Temperatura y humedad

La lectura de temperatura y humedad se realiza mediante el módulo AHT25. Este requiere un voltaje de alimentación de entre 2.2 a 5.5 V. La salida tiene una resolución de 0.01 °C en temperatura y de 0.024% en humedad relativa, en rangos de -40 a 80 °C y 0 a 100%, respectivamente [49]. Además, el módulo utiliza el protocolo I2C para la comunicación.

#### 3.2.2.2. Velocidad de viento

Para medir la velocidad del viento se usa el anemómetro 6410 de *Davis Instruments*. El cual, implementa un interruptor *reed-switch* normalmente abierto que cierra por cada paso de un imán, generando pulsos. Tiene una precisión de  $\pm 2$  mph o  $\pm 5\%$ , y un rango de medición de 1 a 200 mph [48].

Para medir la velocidad del viento se configuró un pin en el microcontrolador para ejecutar una interrupción que detecta los pulsos del anemómetro. Luego, se mide la frecuencia de estos pulsos durante un período de muestreo (por defecto 10 segundos). Para traducir el número de pulsos a velocidad, se utiliza la relación de conversión proporcionada por el fabricante [48]. Esta expresión es la siguiente:

$$V = P \left( \frac{2.25}{T} \right) \quad (3.4)$$

Donde:

- $V$  corresponde a la velocidad en millas por hora, que requiere una conversión adicional para obtener el valor en km/h.
- $P$  representa la cantidad de pulsos ocurridos en un período de muestreo.
- $T$  se define como el período de muestreo en segundos.

Para realizar la conversión de millas por hora a kilómetros por hora, considerando además un período de muestreo de 10 segundos, la fórmula resultante para obtener la velocidad del viento en km/h es:

$$V_{km/h} = P \times 0.225 \times 1.609 \quad (3.5)$$

La velocidad de viento más alta registrada en la Tierra es de 408 km/h [50]. Los pulsos que generaría dicha velocidad son 1,127 en 10 segundos, o 112.7 pulsos por segundo. Esto significa que, para que el microcontrolador ESP32-S3 sea capaz de registrarla, debe leer una señal de aproximadamente 112 Hz. Esto es posible, ya que la frecuencia del ULP usado de 20 MHz.

### 3.2.2.3. Precipitación

Para medir la precipitación, se usa el pluviómetro 7857 de *Davis Instruments*. Este es de tipo balancín, mismo que, mediante un colector de agua, el agua de lluvia entra en un depósito que al llenarse deja caer el agua y se vacía. Este mecanismo implementa un imán y en la base se localiza un interruptor. Con cada cambio de posición del balancín debido al llenado, el imán pasa sobre el interruptor y provoca que este se cierre y abra, generando pulsos que representan la precipitación [51].

Para capturar la cantidad de lluvia en un intervalo de tiempo, se debe contar la cantidad de pulsos generados durante ese intervalo. Debido a que el microprocesador ESP32-S3 pierde su capacidad de capturar datos en sus puertos GPIO cuando entra en modo *deep sleep*, se ha seleccionado el contador S-35770 para almacenar la cantidad de pulsos.

El módulo S-35770 es un contador de 24 bits con interfaz I2C de bajo consumo, capaz de contar de 0 a 16,777,215 [42]. Los pulsos enviados por el pluviómetro se registran

en este contador, que almacena el valor hasta que el módulo ESP32 sale del estado de *deep sleep* y puede recibir la información recopilada durante su inactividad.

El valor del contador requiere una conversión para representarlo en milímetros de lluvia. El fabricante especifica que cada pulso corresponde a 0.01 pulgadas, lo que equivale a 0.254 milímetros [52]. Por lo tanto, el valor en milímetros de lluvia se calcula multiplicando el número de pulsos registrados por 0.254 mm/pulso.

#### **3.2.2.4. Módulo RTC**

En la trama VMTP existe un campo dedicado a las marcas de tiempo (*timestamp*) que usan el formato Unix. Por esta razón, es fundamental contar con un módulo RTC que proporcione la hora y la fecha precisas.

El módulo RTC tiene su propio suministro de energía (una batería de 3.3 V externa e independiente). Este módulo se sincroniza con un servidor Network Time Protocol (NTP) [53], en condiciones normales, no es necesario volver a sincronizar el RTC, sin embargo, si se requiere alta precisión o hay cambios significativos en la temperatura, se recomienda volver a sincronizar el DS3231 periódicamente para garantizar la precisión de la hora. Posteriormente, se ejecuta el software general de la estación hidrometeorológica, que lee la hora y fecha para cada medición y coloca este valor en el campo correspondiente de la trama.

### **3.3. Aplicación del protocolo VMTP**

El protocolo VMTP es una herramienta de gran utilidad para enviar información minimizando el consumo de energía. La disposición de variables en la carga útil, junto con las banderas del protocolo, permite analizar no solo las lecturas de los sensores, sino también verificar el correcto funcionamiento de los mismos, además de proporcionar una lectura precisa del tiempo de la medición. Las secciones siguientes describen cada uno de los campos de la trama, junto con su funcionamiento.

### 3.3.1. Bandera de variación

La bandera de variación se ha usado para indicar variación en los datos con respecto a las mediciones anteriores. Esta ha sido modificada cuando hay una variación entre los datos actuales y los anteriores. En este contexto, un valor de 0 indica que no hubo variación y un valor de 1 indica que sí la hubo.

### 3.3.2. Marca de tiempo

Para la creación de la marca de tiempo se ha utilizado el tiempo Unix, lo que permite representar la hora y la fecha en un formato de 32 bits. En el transmisor se hace un ajuste para representarlo con los 24 bits que exige VMTP. Dicho cambio es revertido en la aplicación receptora, restaurando así la representación de 32 bits.

### 3.3.3. Bandera de nulos

El software ha sido diseñado de tal forma que permita conocer el estado de funcionamiento de sensores y módulos usados para las lecturas. Su correcto funcionamiento se ha representado con un valor de 1, mientras que, en caso de error o fallo en la comunicación con estos, el valor se representa con 0. Finalmente, dichos valores se colocan en la posición correspondiente del campo de bandera de nulos de VMTP.

### 3.3.4. Bandera de ceros

La bandera de ceros ha sido usada para indicar cuando una variable ha tomado el valor de cero. Estos bits representan a las variables de velocidad de viento, lluvia, radiación solar e índice UV. En caso de que la lectura de las variables tome un valor distinto de cero, entonces el bit correspondiente a esa variable se ubica en el campo VMTP tomando el valor de 1.

### 3.3.5. Variables hidrometeorológicas

Este campo se ha completado tomando las lecturas de cada sensor de la estación, y haciendo las modificaciones necesarias para que dichas variables sean colocadas en VMTP. Para las variables leídas por el módulo ADC, dicha modificación corresponde a la transformación de voltaje (proveniente del ADC) a la unidad fundamental de esa variable de clima ( $W/m^2$  para radiación solar, valor de 0 a 16 para el índice UV y valor de  $0^\circ$  a  $360^\circ$  para dirección de viento). El sensor AHT20 cuenta con su propia librería que facilita su lectura, por lo que no es necesario realizar cambios en las variables de temperatura y humedad. Finalmente, los valores del contador de lluvia y del contador para la velocidad de viento son usados para convertirlos en cantidad de lluvia y velocidad en km/h, respectivamente.

Cada variable de clima se representa con la cantidad de bits que le corresponde, en caso de que el sensor no se encuentre disponible en la estación, el campo correspondiente toma el valor de cero. Este es el caso del sensor de presión atmosférica.

### 3.3.6. Carga útil de la trama

La carga útil corresponde a la información que deseamos transmitir, excluyendo información de control y verificación de errores de los sensores. Específicamente, se compone de las variables de los sensores y el *timestamp*. El contenido y longitud en bits del *payload* se especifica en la Figura 3.7.

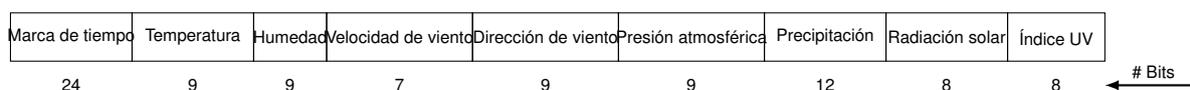


Figura 3.7: Carga útil de VMTP

## 3.4. Transmisión y gestión de datos

Los datos capturados de los sensores son enviados utilizando el protocolo VMTP y comunicación LoRaWAN a un servidor TTS de la Universidad de Cuenca. Para esto, es necesario conectarse a un *gateway* LoRaWAN que se encuentra ubicado en la

terrazza del Laboratorio de Microrred. Obtenidos los datos en el servidor TTS, se captura los paquetes *uplink* mediante un manejador de datos *uplink* implementado en un contenedor *Docker*. Este manejador de datos captura por MQTT los paquetes *uplink* del servidor TTS y los reenvía por *Hypertext Transfer Protocol Secure* (HTTPS) a un servidor de Thingsboard de la Universidad de Cuenca. Aquí, los paquetes son decodificados para finalmente mostrarlos en el *dashboard*. En la Figura 1.2 se presenta el esquema completo del funcionamiento del sistema.

A continuación se presenta a detalle la aplicación del protocolo LoRaWAN, la conexión con el servidor TTS, el envío y tratamiento de datos en Thingsboard.

### 3.4.1. Aplicación del protocolo Long Range Wide Area Network (LoRaWAN)

La estación hidrometeorológica está compuesta por el módulo de sensores, el módulo de comunicación. Esta estación opera como un nodo final dentro de la arquitectura de la red LoRaWAN. La integración de LoRaWAN en el nodo final se realiza mediante la librería RadioLIB [54], que es compatible con LoRaWAN versión 1.1.0 y 1.0.4. Se utilizó la versión 1.1.0 y el método de activación Over the Air Activation (OTAA) que requiere los siguientes identificadores y claves para registrar el dispositivo en el servidor TTS:

- *JOIN EUI*: Identificador único utilizado para la solicitud de unión a la red.
- *DEV EUI*: Identificador único del dispositivo.
- *APP KEY*: Clave de aplicación utilizada para la autenticación y generación de claves de sesión.
- *NWK KEY*: Clave de red utilizada para asegurar la comunicación con el servidor de red.

Para establecer el envío de datos desde el nodo final al servidor TTS se requiere que el nodo final se configure con los mismos parámetros regionales que el servidor. Además, necesita estar en la misma banda de operación que el *gateway* LoRaWAN. Es decir, en la región US915 y en el plan de frecuencias de 902-928 MHz. Adicionalmente, la librería de RadioLIB permite modificar parámetros como el tiempo de envío,

entre otros.

Junto a la librería RadioLIB se implementa un mecanismo de persistencia. Este no permite al nodo final perder los datos de la sesión establecida con el servidor luego de salir del modo *deep sleep*. Para ello, se usó el ejemplo de persistencia que se presenta en el repositorio de *GitHub* [55] compatible con la versión 6.5 de *RadioLIB*.

Finalmente, para enviar los datos al servidor TTS se requiere que el dispositivo esté registrado en el servidor TTS y con credenciales válidas. En el Apéndice A se presenta el procedimiento para agregar un nuevo dispositivo al servidor TTS y obtener los datos de aprovisionamiento para el nodo final.

### 3.4.2. Diagrama de adquisición de datos y comunicación con TTS

El procedimiento descrito en secciones previas puede ser representado usando un diagrama de flujo, dicho diagrama ayuda a simplificar la comprensión del software implementado. La Figura 3.8 presenta el diagrama de flujo del software implementado en el microcontrolador ESP32-S3, en el cual se detallan los procedimientos necesarios para la recolección de datos y su envío al servidor TTS mediante comunicación LoRaWAN.

### 3.4.3. Envío de datos a Thingsboard

Luego de recibir los datos en TTS es necesario presentarlos en una interfaz de visualización que facilite la monitorización del estado de los sensores. Para este propósito, se utiliza Thingsboard. Esta es una plataforma de IoT de código abierto para la recopilación, el procesamiento, la visualización y la gestión de dispositivos [56].

Para lograr este objetivo, se utiliza el servidor MQTT que integra TTS, que permite al cliente MQTT suscribirse a enlaces *uplink*. El suscriptor MQTT se implementó en un contenedor *Docker*.

El cliente MQTT en el contenedor *Docker* se pone en marcha mediante un programa en Python. Este programa recibe el tráfico en el enlace *uplink* y envía los datos del enlace ascendente, mediante *HTTP* al servidor de *Thingsboard*. Este servidor decodi-

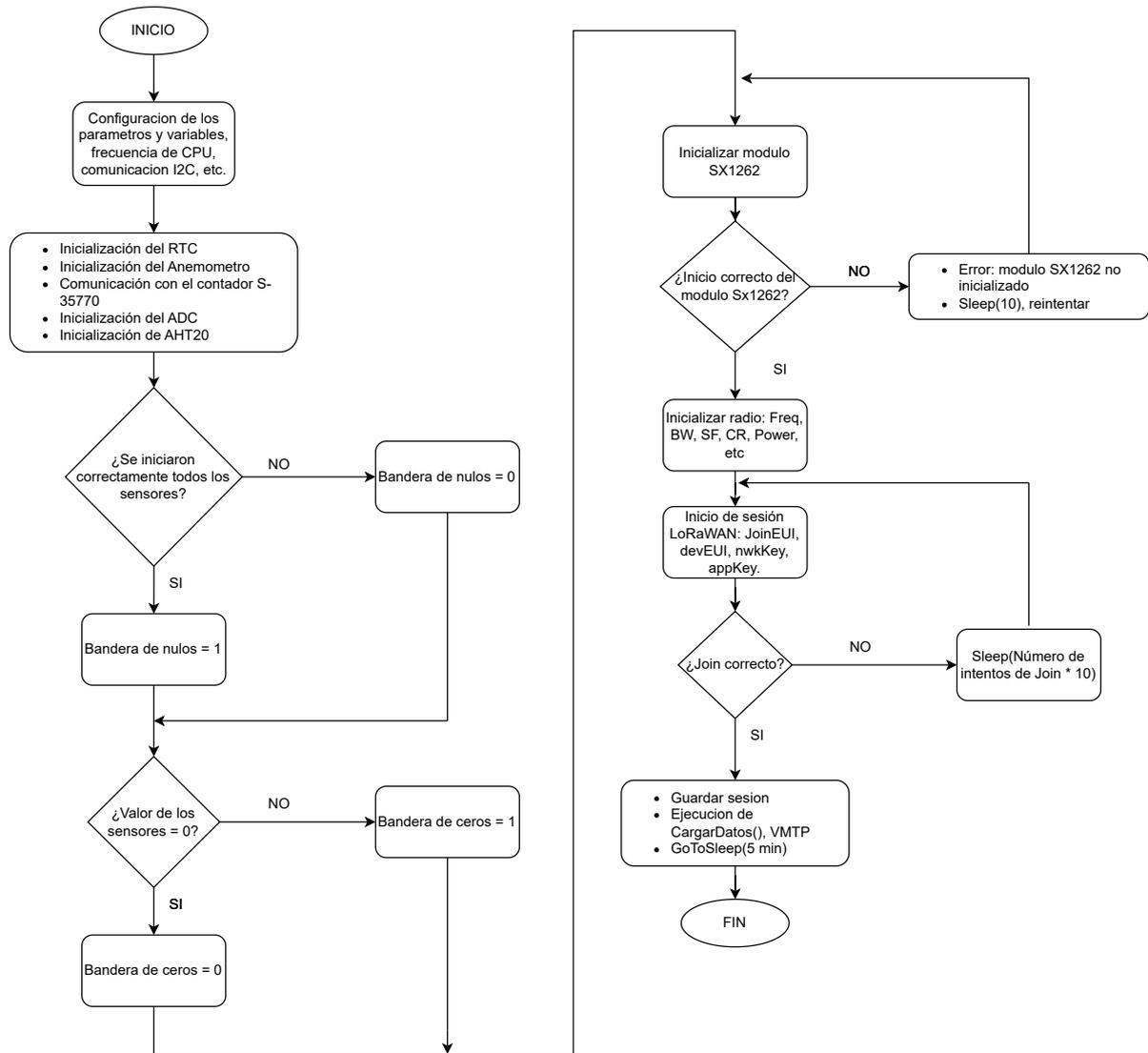


Figura 3.8: Diagrama de flujo del procedimiento de adquisición de datos y comunicación con TTS

fica el *Payload* con la trama VMTP para obtener los datos de cada una de las variables meteorológicas y los muestra en el *dashboard*.

El diagrama de flujo del funcionamiento del manejador de datos alojado en el contenedor Docker se presenta en la Figura 3.9, en este se detallan el procedimiento para el reenvío de datos de TTS a Thingsboard. Para lo cual, se configuran las credenciales de MQTT y se establece la conexión, posterior a lo cual se realiza la suscripción al *broker* MQTT de TTS y se envía el mensaje MQTT recibido a Thingsboard mediante HTTPS.

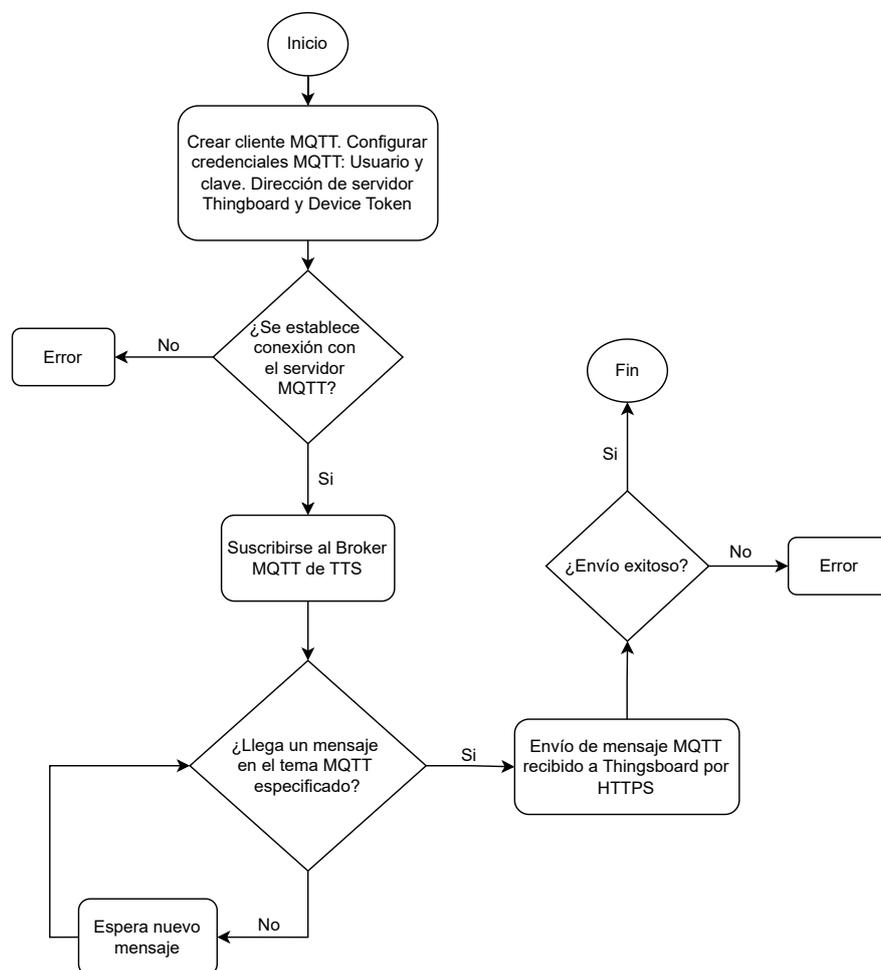


Figura 3.9: Diagrama de flujo del reenvío de datos de TTS a Thingsboard implementado en un contenedor Docker.

La creación de un dispositivo y el proceso de cómo implementar un bloque de decodificación en Thingsboard se presentan en el Apéndice A. Adicionalmente, en la

Figura 3.10 se presenta el diagrama de flujo del funcionamiento del programa realizado en JavaScript para decodificar los mensajes de llegada en Thingsboard, en donde constan los procedimientos para la decodificación de la trama VMTP, incluyendo la verificación de la existencia y tamaño del *payload*, su extracción y la generación del archivo JSON. La salida de este decodificador actualiza las métricas de las variables hidrometeorológicas en el dispositivo correspondiente de Thingsboard.

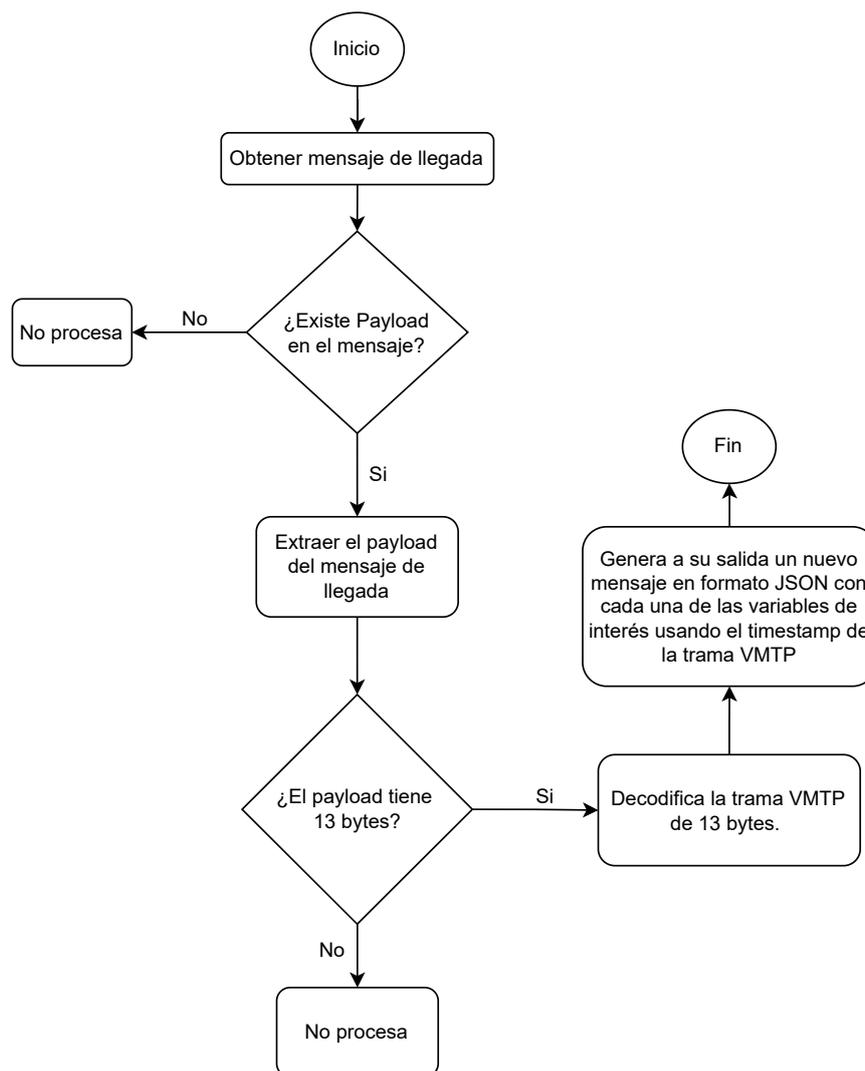


Figura 3.10: Diagrama de flujo del decodificador implementado en Thingsboard.

### 3.5. Metodología de análisis de datos

La captura de datos se realiza en el Laboratorio de Microrred, específicamente en el techo del laboratorio. La ubicación ha sido seleccionada debido a la ubicación de la estación que es usada como referencia para comparar los datos capturados. La estación del Laboratorio de Microrred compuesta por sensores de Nippo Electric Instruments INC., mide variables de humedad relativa, temperatura ambiente, presión atmosférica, precipitación, radiación solar, velocidad y dirección del viento. Esta estación almacena internamente valores promedio de las mediciones en intervalos de 5 minutos.

En la estación hidrometeorológica implementada, el almacenamiento y medición de precipitación se hace cada que el balancín de agua marca una muestra. Mientras que, las variables de radiación solar, radiación UV, temperatura, humedad y dirección de viento, se miden cada 3 minutos, por efecto de pruebas. Los requerimientos del PROMAS indican un periodo de 5 minutos entre muestras, por lo que una vez que la estación se instale en campo, el valor del periodo de muestreo se restablece en 5 minutos. El resto de tiempo la estación está en el estado *deep sleep*.

Un aspecto importante a destacar es que la estación implementada tiene una mayor cantidad de información hidrometeorológica en comparación con esta estación de la Microrred, debido principalmente a que la frecuencia de muestreo es mayor. Por esta razón, los datos de la estación implementada fueron interpolados, lo cual permite obtener la misma cantidad de muestras y comparar los resultados de ambas estaciones. Para ello, se usa el coeficiente de correlación de Pearson.

El coeficiente de correlación de Pearson mide la relación lineal entre dos variables [8]. Un coeficiente de  $\pm 1$  indica una relación lineal perfecta, positiva o negativa, dependiendo del signo. Sin embargo, un coeficiente 0 indica que no hay una relación. En la Tabla 3.3 se presenta la interpretación cualitativa de la magnitud del coeficiente de correlación de Pearson entre dos variables  $X$  y  $Y$ .

Por lo tanto, este coeficiente se utiliza como métrica de comparación de las mediciones entre la estación propuesta y la estación de la Microrred. El análisis de correlación de cada una de las variables hidrometeorológicas se detalla en la Sección 4.7.

Rango de valores de $r_{XY}$	Interpretación
$0.00 \leq  r_{XY}  < 0.10$	Correlación nula
$0.10 \leq  r_{XY}  < 0.30$	Correlación débil
$0.30 \leq  r_{XY}  < 0.50$	Correlación moderada
$0.50 \leq  r_{XY}  \leq 1.00$	Correlación fuerte

Tabla 3.3: Interpretación del coeficiente de correlación de Pearson según las sugerencias de Cohen. Extraído de [8]

En el contexto de la evaluación del desempeño de la comunicación mediante la tecnología LoRaWAN, se lleva a cabo mediciones para analizar los siguientes parámetros: Packet Delivery Ratio (PDR), Received Signal Strength Indicator (RSSI) y Signal-to-Noise Ratio (SNR). Para este análisis, se establecen cuatro puntos de prueba situados a diferentes distancias entre el *gateway* LoRaWAN y el nodo implementado, con un total de 320 paquetes enviados en cada punto. Los datos de RSSI y SNR se obtienen del paquete *uplink* recibido en el servidor, mientras que, el cálculo de la PDR se realiza mediante la siguiente ecuación:

$$\text{PDR} = \left( \frac{\text{Número de paquetes recibidos}}{\text{Número de paquetes enviados}} \right) \times 100 \quad (3.6)$$

## Resultados y discusión

En este capítulo, se detallan los resultados obtenidos en las distintas fases del proyecto. Primero, se analiza los resultados del funcionamiento de los módulos que componen el hardware del proyecto, incluyendo el consumo de corriente. Luego, la adquisición de datos de cada sensor, la sincronización del RTC y la composición de la trama VMTP, seguido de la comunicación con el servidor TTS utilizando el protocolo LoRaWAN. El decodificador se ha configurado en la plataforma Thingsboard, donde se analizan los resultados presentados en el *dashboard* y se comparan con la información recopilada por una estación hidrometeorológica ubicada en las cercanías del laboratorio de la Microrred.

### 4.1. Módulos PCB de comunicación y de sensores en la adquisición de datos

El módulo de comunicación resultante cuenta con dos capas. En la capa superior está colocado el chip LoRa SX1262, reguladores de voltaje, capacitores y resistencias, como se muestra en la Figura 4.1. En la capa inferior solamente se encuentra colocado el microcontrolador ESP32-S3, dicho resultado se muestra en la Figura 4.2. El dispositivo completo se muestra en la Figura 4.3. En la Figura se observa el PCB de comunicación montado en el PCB de sensores.

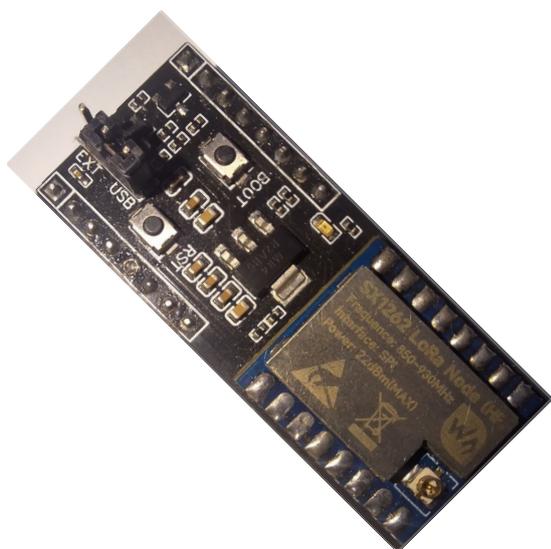


Figura 4.1: Lado superior del módulo de comunicación

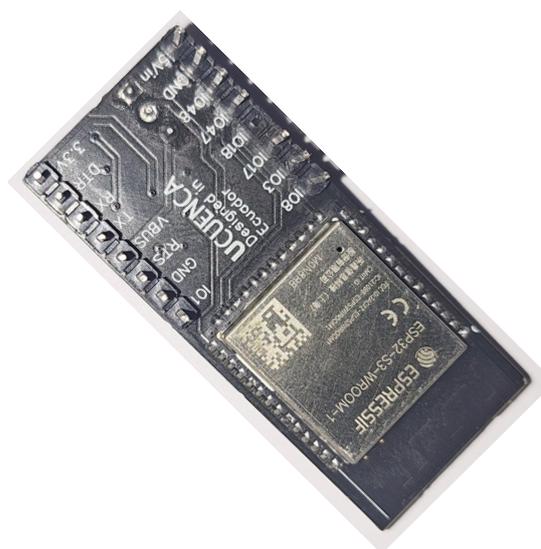


Figura 4.2: Lado inferior del módulo de comunicación

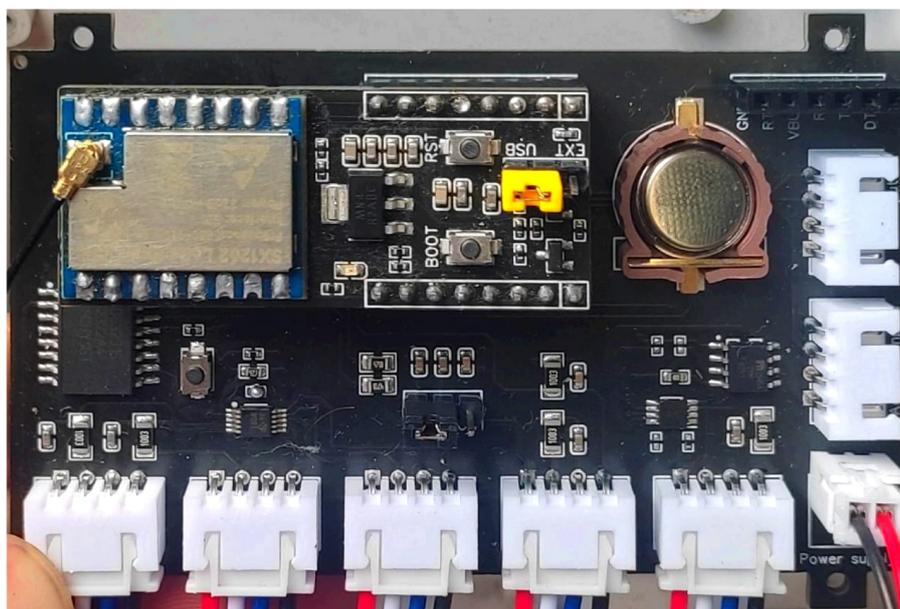


Figura 4.3: Módulos PCB de comunicación y de sensores

## 4.2. Consumo de corriente

Se analizó el consumo de corriente del PCB implementado, identificando las diferencias de consumo al reducir la frecuencia de funcionamiento del Central Processing Unit (CPU) y funcionar en modo *Deep Sleep*. La Tabla 4.1 muestra el consumo de corriente usando frecuencias de 20, 40 y 240 MHz para el CPU, en modo de funcionamiento normal o Activo y en *Deep Sleep*.

Tabla 4.1: Consumo de corriente usando distintas frecuencias de CPU y en modo de funcionamiento Activo y *Deep Sleep*

Frecuencia de CPU	Consumo de corriente	
	Modo Activo	Modo Deep Sleep
20 MHz	17.6 mA	2.1 mA
40 MHz	22 mA	2.2 mA
240 MHz	50 mA	2.2 mA

El cambio de la frecuencia de funcionamiento no tiene un impacto en el consumo energético en modo *Deep Sleep*, debido a que la principal característica de este modo es que apaga todos los periféricos, incluyendo CPU y únicamente mantiene energizada la memoria RTC, la cual no depende de la frecuencia de funcionamiento del CPU. Sin embargo, sí se observa una reducción de consumo al trabajar con menor frecuencia

de CPU en Modo Activo, además que, de forma general, las operaciones y cálculos se realizan en mayor tiempo.

El tiempo que el módulo ESP32-S3 permanece en modo Activo es de aproximadamente 30 segundos, mientras que, en modo *Deep Sleep* se mantiene 3 minutos. Es decir, en un periodo de una hora, se tiene un total de aproximadamente 17 ciclos (modo Activo - modo *Deep Sleep*). Con lo que, usando una frecuencia de funcionamiento de 20 MHz se tiene un consumo de corriente por ciclo de:

$$I_{\text{Modo Activo}} = 17.6 \text{ mA} \times 0.5 \text{ min} \times \frac{1\text{h}}{60\text{min}} = 0.1466 \text{ mA/h}$$

$$I_{\text{Deep Sleep}} = 2.1 \text{ mA} \times 3 \text{ min} \times \frac{1\text{h}}{60\text{min}} = 0.105 \text{ mA/h}$$

$$I_{\text{ciclo}} = I_{\text{Modo Activo}} + I_{\text{Deep Sleep}} = 0.1466 + 0.105 = 0.2516 \text{ mA/h}$$

En contraste con el consumo energético al usar una frecuencia de funcionamiento de 240 MHz:

$$I_{\text{Modo Activo}} = 50 \text{ mA} \times 0.5 \text{ min} \times \frac{1\text{h}}{60\text{min}} = 0.4166 \text{ mA/h}$$

$$I_{\text{Deep Sleep}} = 2.2 \text{ mA} \times 3 \text{ min} \times \frac{1\text{h}}{60\text{min}} = 0.11 \text{ mA/h}$$

$$I_{\text{ciclo}} = I_{\text{Modo Activo}} + I_{\text{Deep Sleep}} = 0.4166 + 0.11 = 0.5266 \text{ mA/h}$$

Los resultados de consumo de corriente indican que, reducir la frecuencia de funcionamiento a 20 MHz reduce el consumo de corriente a menos del 50%.

### 4.3. Pruebas del protocolo VMTP

VMTP proporciona un modelo de trama que permite enviar la información de los sensores eficientemente. Una vez definidos los datos de todos los sensores, el resultado

es una trama con un tamaño de 13 bytes. En un escenario de prueba, se fijaron las variables en un valor constante, tal como se muestran a continuación:

- Bandera de variación: 1
- Marca de tiempo: 257
- Bandera de nulos:17
- Bandera de ceros: 7
- Temperatura: 245
- Humedad: 33
- Velocidad de viento: 25
- Dirección de viento: 7
- Presión atmosférica: 55
- Precipitación: 15
- Radiación solar: 1,047
- Radiación UV: 7

La trama codificada resultante es la que se muestra en la Figura 4.4. Esta trama es enviada al receptor y este se encarga de decodificar la trama recibida usando los tamaños definidos de cada campo. El resultado es el que se muestra en la Tabla 4.2.

Tabla 4.2: Desglose de trama VMTP

<b>Campo</b>	<b>Binario</b>	<b>Decimal</b>
Bandera de variación (1 bit)	1	1
Marca de tiempo (24 bits)	0000000 00000000 10000000 1	257
Bandera de nulos (8 bits)	0001000 1	17

Tabla 4.2: (Continuación)

<b>Campo</b>	<b>Binario</b>	<b>Decimal</b>
Bandera de ceros (4 bits)	0111	7
Temperatura (9 bits)	011 110101	245
Humedad (7 bits)	01 00001	33
Velocidad de viento (9 bits)	000 011001	25
Dirección de viento (4 bits)	01 11	5
Presión atmosférica (9 bits)	000110 111	55
Precipitación (9 bits)	00000 1111	15
Radiación solar (12 bits)	0100 00010111	1,047
Radiación UV (8 bits)	00000111	7

Al revisar la información, se observó un correcto envío y recepción de las variables climáticas. Además, se envió únicamente la información necesaria, sin bits de relleno, lo cual minimiza el consumo energético y favorece la eficiencia del sistema.

#### 4.4. Comunicación con TTS usando LoRaWAN

En la comunicación mediante LoRaWAN el nodo final inicia el proceso de unión (*JOIN*) al servidor TTS mediante el *gateway* LoRaWAN. El servidor recibe la solicitud y verifica que las credenciales correspondan a las alojadas en el servidor.

En la Figura 4.5 se muestra que el proceso de *JOIN* ha sido completado, también se muestra que el primer paquete *uplink* ha sido recibido. Una vez establecida la sesión, el nodo final no vuelve a hacer el proceso de *JOIN*, a menos que se desconecte o reinicie.

Dentro de la información del paquete *uplink* (Figura 4.6) se puede verificar los pa-

10000000	00000000	10000000
Byte 1	Byte 2	Byte 3
10001000	10111011	11010101
Byte 4	Byte 5	Byte 6
00001000	01100101	11000110
Byte 7	Byte 8	Byte 9
11100000	11110100	00010111
Byte 10	Byte 11	Byte 12
00000111		
Byte 13		

Figura 4.4: Trama VMTP recibida



Figura 4.5: Proceso de inicialización en el servidor LoRaWAN

rámetros de operación de la comunicación LoRaWAN, como son: ancho de banda, *spreading factor*, *coding rate* y frecuencia de funcionamiento. Además, parámetros como: SNR, RSSI y *timestamp* del paquete recibido. Finalmente, se encuentran algunos de los datos de aprovisionamiento y el *payload* codificado mediante VMTP.

```
56     "rx_metadata": [  
57         {  
58             "gateway_ids": {  
59                 "gateway_id": "eui-b827ebfffe2f0b48",  
60                 "eui": "B827EBFFFE2F0B48"  
61             },  
62             "timestamp": 2089417740,  
63             "rssi": -58,  
64             "channel_rssi": -58,  
65             "snr": 10.5,  
66             "uplink_token": "CiIKIAoUZXVpLWI4MjdlYmZmZmUyZjBiNDgSCLgn6//+LwtIEIz4p+QHGGwIiOrSswYQz9PC9AEg4J2C20f9FA==",  
67             "channel_index": 6,  
68             "received_at": "2024-06-20T23:02:32.512797135Z"  
69         }  
70     ],  
71     "settings": {  
72         "data_rate": {  
73             "lora": {  
74                 "bandwidth": 125000,  
75                 "spreading_factor": 7,  
76                 "coding_rate": "4/5"  
77             }  
78         },  
79         "frequency": "905100000",  
80         "timestamp": 2089417740  
81     },  
82     "received_at": "2024-06-20T23:02:32.514312437Z",  
83     "consumed_airtime": "0.061696s",  
84     "network_ids": {  
85         "net_id": "000000"  
86     }
```

Figura 4.6: Metadatos del paquete *uplink*

#### 4.5. Verificación del decodificador

Los datos se reciben en el servidor TTS y se envían a la plataforma Thingsboard para el procedimiento de decodificación. Su estructura se presenta en la Figura 4.6. La decodificación se implementó mediante un *script* en JavaScript, que actualiza las telemetrías de cada una de las variables hidrometeorológicas capturadas.

El correcto funcionamiento del decodificador implica que en Thingsboard reciba telemetrías que concuerden con los valores capturados por los sensores del nodo final. En la Figura 4.7 se muestra las variables decodificadas, y se actualizan al arribo de un nuevo dato. Las variables decodificadas representan las lecturas de cada uno de los sensores, el campo *timestamp* corresponde al tiempo obtenido del RTC al momento de la medición.

Actividad Sensores

🕒 Tiempo-real - último(s) 7 días

Timestamp ↓	Temperatura [°C]	Humedad [%]	Radiación Solar [W/m²]	Índice de radiación UV
2024-06-20 17:28:53	16	80	35	0
2024-06-20 17:25:30	16.1	79	35	0
2024-06-20 17:22:07	16	77	35	0
2024-06-20 17:18:45	16	78	35	0
2024-06-20 17:15:22	16.1	80	38	0
2024-06-20 17:11:59	16.2	79	38	0
2024-06-20 17:08:37	16.3	79	35	0
2024-06-20 17:05:14	16.2	77	38	0
2024-06-20 17:01:51	16.4	79	59	0
2024-06-20 16:58:28	16.5	78	68	0

Figura 4.7: Decodificación de variables VMTP correspondientes al *timestamp*, temperatura, humedad, radiación solar y radiación UV

Es importante mencionar que, con respecto a la variable de dirección del viento, es necesario mapearla a la dirección cardinal correspondiente. El resultado de decodificación y mapeo, junto con la velocidad del viento y las mediciones de precipitación, se ilustran en la Figura 4.8.

Dirección del Viento	Velocidad del Viento [Km/h]	Precipitación [mm]
SE	0	0
SSE	0	0
SSO	14.48406	0
SE	0	0
SSE	0	0
SSE	0	0.51
S	3.21868	0
SSE	12.87472	0
S	11.26538	0
SSE	12.87472	0

Figura 4.8: Decodificación de variables VMTP correspondientes a la dirección de viento, velocidad de viento y precipitación

Adicionalmente, los datos decodificados son mostrados en tiempo real mediante gráficas temporales, usando el *dashboard* como se muestra en la Figura 4.9. En esta es posible modificar la ventana de tiempo que deseamos observar, junto con la cantidad de muestras, permitiendo mayor detalle en el análisis de información.

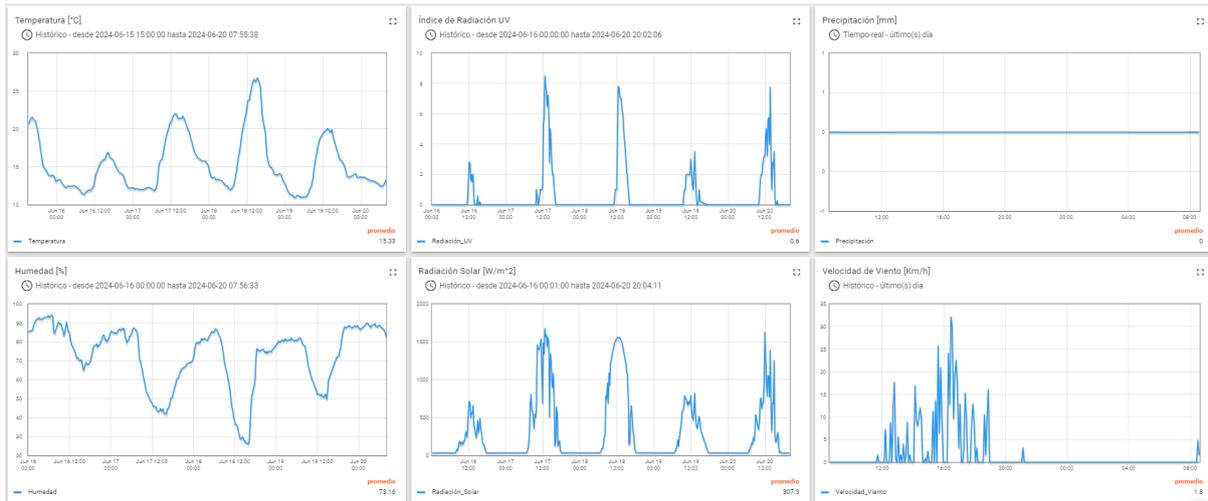


Figura 4.9: *Dashboard* que presenta un gráfico de las mediciones de temperatura, humedad, radiación solar, radiación UV, precipitación y velocidad de viento, a lo largo del tiempo

En cuanto a la dirección del viento, se presenta los datos en forma de tabla, como se observa en el *widget* de la Figura 4.10.

Dirección de viento	
Historico - último(s) día	
Timestamp ↓	Dirección del Viento
2024-06-20 21:52:28	SSE
2024-06-20 21:49:06	SSE
2024-06-20 21:45:43	SSE
2024-06-20 21:42:20	S
2024-06-20 21:38:58	S

1 - 10 of 424

Figura 4.10: Mediciones y mapeo de dirección de viento, según el transcurso del día

Adicionalmente, mediante *widget* de tipo *Led Indicator* se visualizan de las banderas de nulos, ceros y bit de variación. En la Figura 4.11 se presentan las banderas de nulos, ceros y variación. Se puede observar que el estado de la bandera de nulo del sensor de presión atmosférica es inactivo, debido a que este sensor no ha sido imple-

mentado. Mientras que el resto de variables de las banderas de nulos se encuentran activas, lo que indica el correcto funcionamiento. De manera similar, las banderas de ceros, representan qué variables son cero o no, que el led se encuentre activo indica que el valor de la variable es diferente de cero. Así también, el led correspondiente al bit de variación indica que existe una variación con respecto a los resultados de la muestra anterior.



Figura 4.11: Representación de la bandera de nulos de VMTP

#### 4.6. Evaluación de Métricas de Comunicación LoRaWAN

En esta sección, se lleva a cabo el análisis de la calidad de la comunicación a través de LoRaWAN, centrada en tres métricas en función de la distancia entre el *gateway* y el nodo implementado: PDR, RSSI y SNR.

Para ello, se estableció cuatro puntos de medición con línea de vista directa ubicados a: 82, 168, 227 y 436 metros. En cada punto se realizó el envío de un total de 320 paquetes.

En la Figura 4.12 se presenta la PDR expresada en porcentaje en función de la distancia. Los resultados indican el decremento de paquetes recibidos a medida que aumenta la distancia entre el *gateway* y el nodo, evidenciando la influencia de la distancia en la eficacia de la comunicación. En general, los resultados muestran una pérdida de paquetes menor al 2% a distancias menores 450 metros.

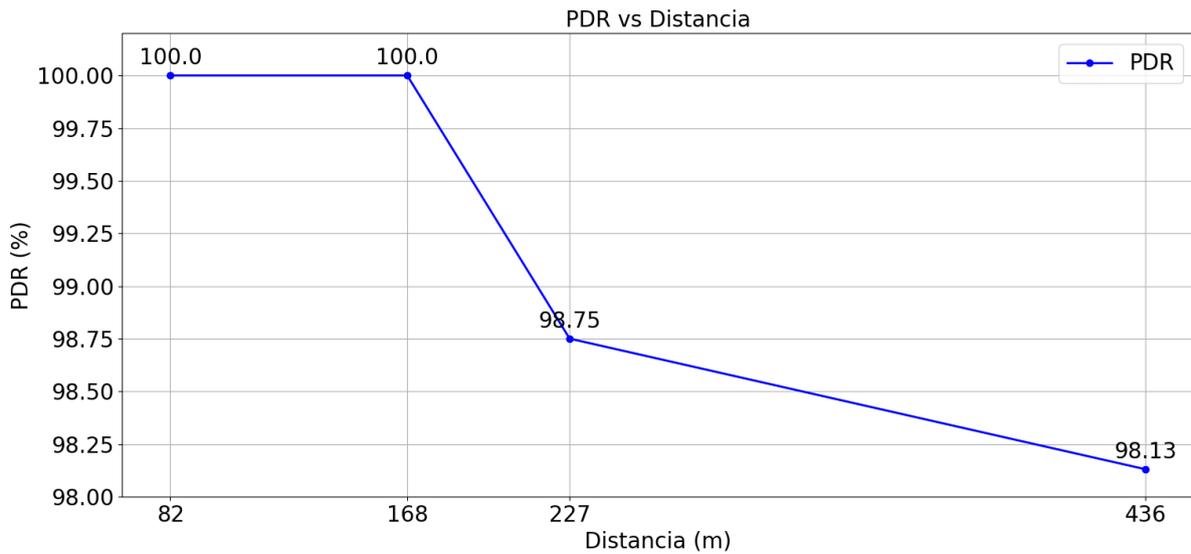


Figura 4.12: Tasa de pérdida de paquetes en función de la distancia.

Las Figuras 4.13 y 4.14 ilustran la Intensidad de la Señal Recibida (RSSI) y la Relación Señal a Ruido (SNR) respectivamente, en función de la distancia. Estas gráficas revelan cómo tanto la intensidad de la señal, como la relación señal a ruido, disminuyen a medida que incrementa la distancia.

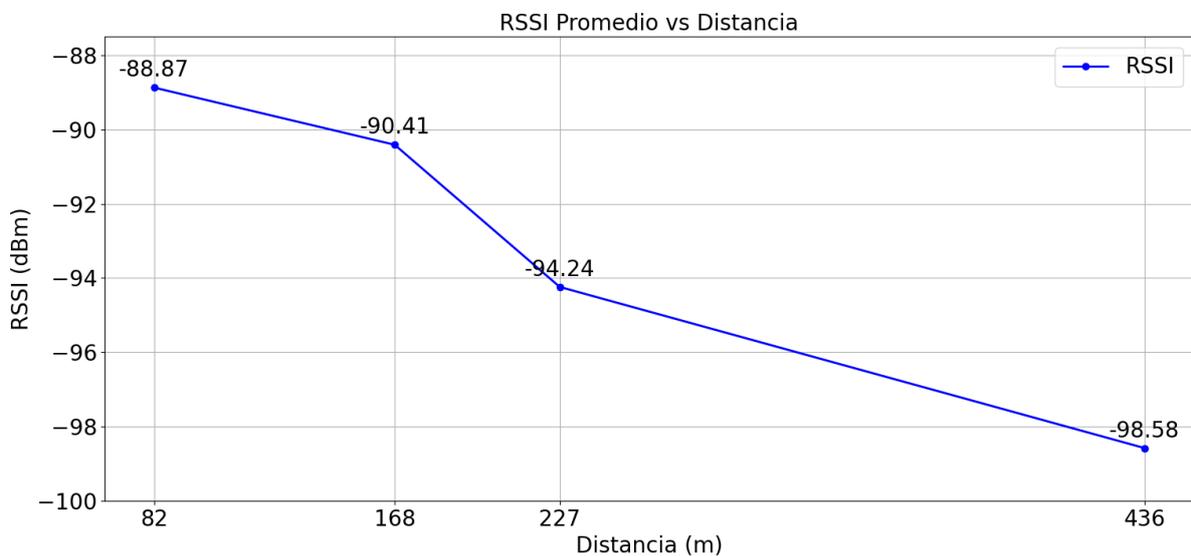


Figura 4.13: Intensidad de la señal recibida (RSSI) en función de la distancia.

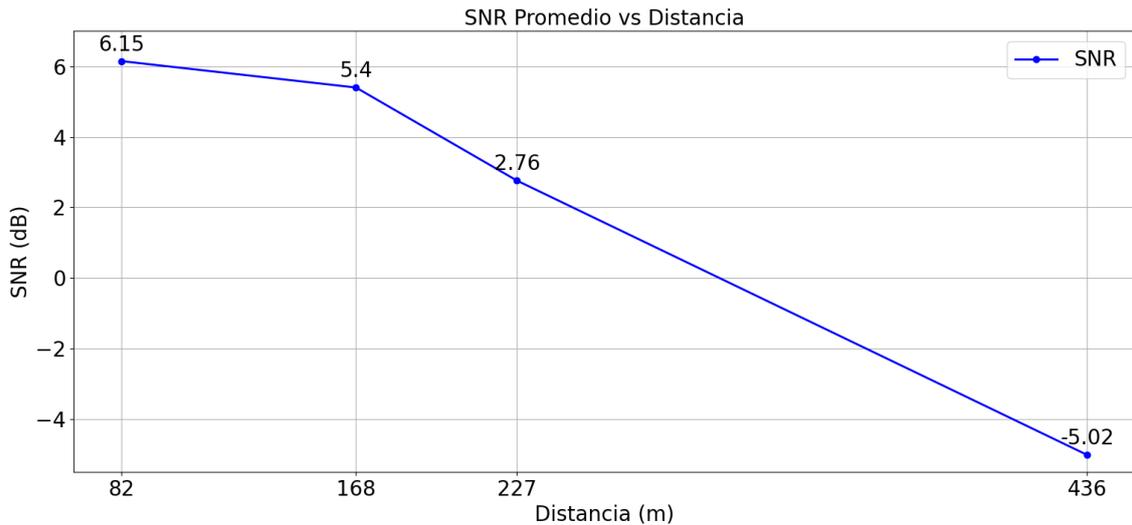


Figura 4.14: Relación Señal a Ruido (SNR) en función de la distancia.

Además, se realizó un experimento a una distancia de 436 metros en condiciones sin línea de vista directa entre el nodo y el *gateway*. Los resultados revelaron una pérdida de paquetes del 25.23%, un promedio de RSSI de -99.72 dBm y un promedio de SNR de -12.69 dB.

Estos resultados sugieren que la calidad de la comunicación se degrada con la distancia, especialmente en situaciones donde no hay línea de vista directa y el entorno circundante contiene numerosos obstáculos. Estos datos están en concordancia con los resultados reportados en [57], que demuestran una tendencia clara en la pérdida de paquetes tanto en condiciones de línea de vista directa como en situaciones con obstáculos y ausencia de línea de vista.

#### 4.7. Análisis comparativo de los datos capturados por la estación desarrollada y la estación hidrometeorológica de la Microrred

Esta sección está dedicada a comparar las mediciones durante un intervalo de 72 horas de la estación desarrollada con las mediciones de la estación hidrometeorológica perteneciente al Laboratorio de Microrred, ambas ubicadas en la azotea del edificio del mismo nombre. La ubicación de la estación de la Microrred se observa en la Figura 4.26, a su vez, la estación implementada se ha colocado a pocos metros, en la ubicación mostrada en la Figura 4.27.

### 4.7.1. Temperatura

En la Figura 4.15 se presenta de color amarillo, la temperatura registrada por la estación hidrometeorológica de la Microrred. Mientras que, de color azul, se presenta la temperatura recopilación por la estación implementada. Los resultados muestran similitud entre las mediciones de las dos estaciones. Para ser específicos, el coeficiente de correlación de Pearson es de 0.99 (ver Figura 4.16, lo que indica una correlación fuerte según la Tabla 3.3. Esto nos revela que las mediciones de temperatura de ambas estaciones están altamente alineadas, validando así la lectura de temperatura del sensor AHT25 utilizado en la propuesta.

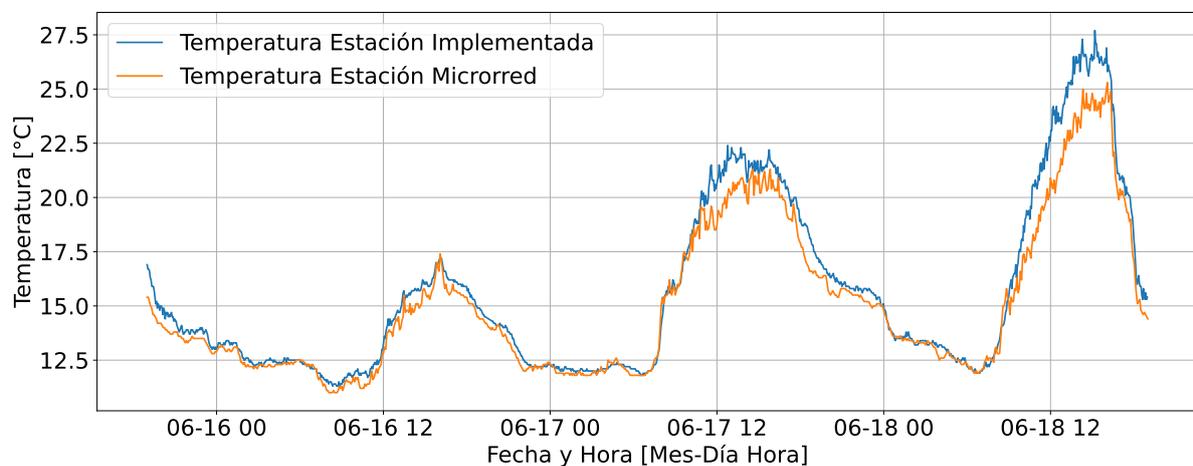


Figura 4.15: Temperatura de la estación hidrometeorológica implementada vs la estación de la Microrred.

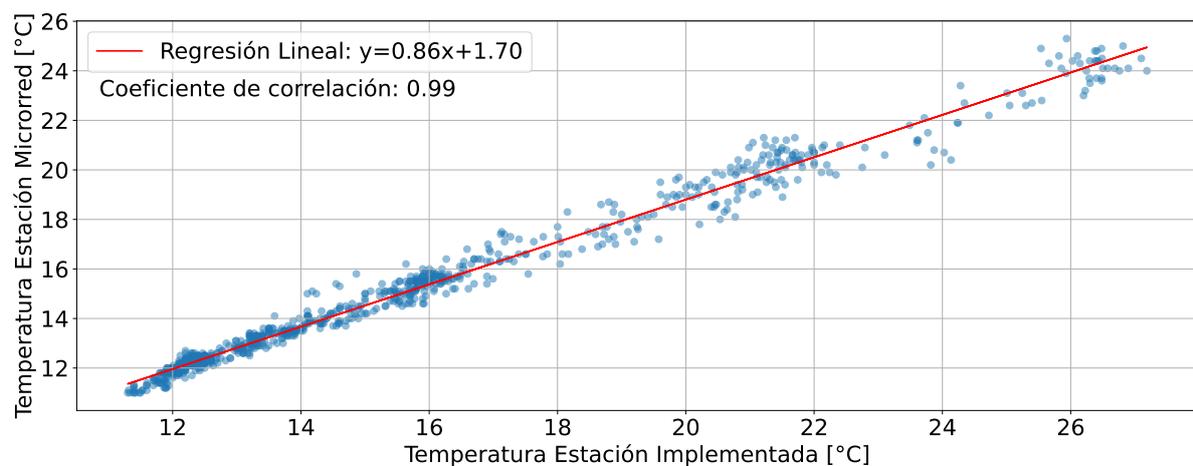


Figura 4.16: Correlación de la temperatura de la estación hidrometeorológica implementada vs la estación de la Microrred.

#### 4.7.2. Humedad

La Figura 4.17 muestra en color amarillo el registro de humedad de la estación de la Microrred y en color azul el registro de humedad de la estación implementada. En cuanto a la variable de humedad, al observar las mediciones de humedad presentadas en la Figura 4.17, se evidencia una diferencia de aproximadamente el 8% entre las mediciones de la estación implementada y la estación de la Microrred. Presumiblemente debido a los tipos y calidad de los sensores. Esto es sustentado por el coeficiente de Pearson de 0.99, es decir, ambos resultados están fuertemente correlacionados. La correlación se detalla en la Figura 4.18.

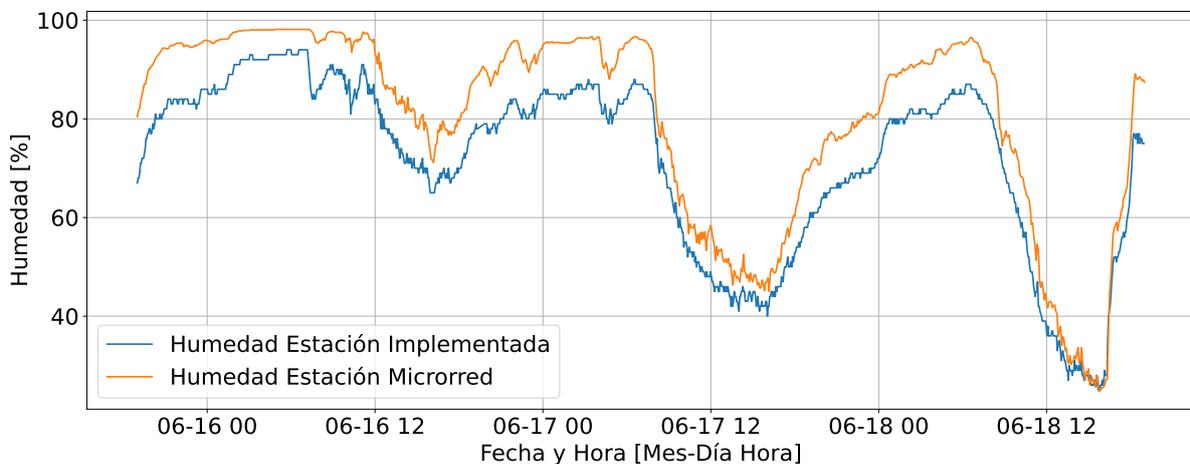


Figura 4.17: Humedad de la estación hidrometeorológica implementada vs la estación de la Microrred.

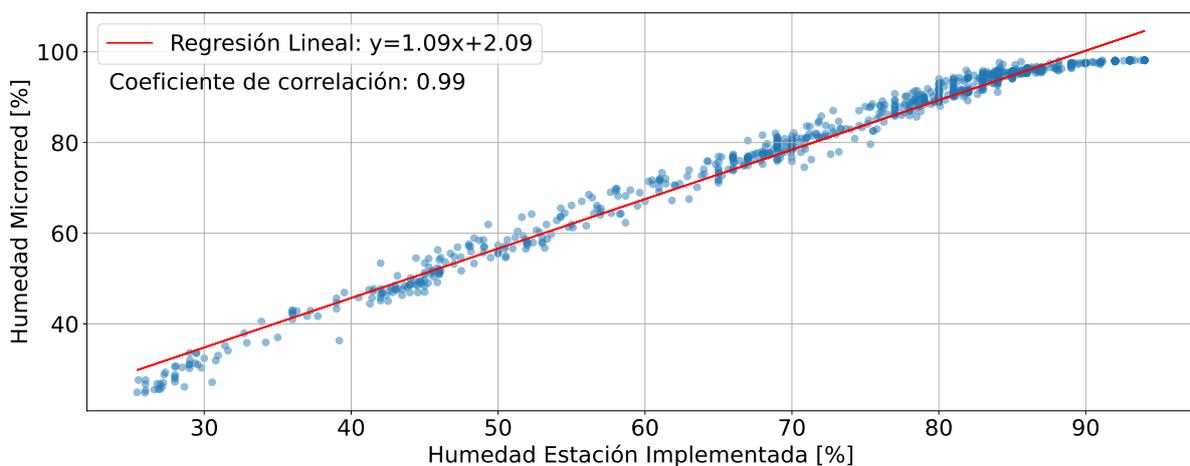


Figura 4.18: Correlación de la humedad de la estación hidrometeorológica implementada vs la estación de la Microrred.

Al analizar los resultados de temperatura y humedad (Figura 4.15 y 4.17), se observa una relación inversa entre las dos variables. Esto se debe a que el aumento de la temperatura del aire incrementa su capacidad para retener vapor de agua, lo que reduce la humedad relativa, en el caso de no haber un incremento proporcional en la cantidad de vapor de agua. Por esta razón, cuando la temperatura del aire aumenta, la humedad relativa tiende a disminuir. Este efecto es más notable al analizar las horas de mayor incremento en la temperatura, que corresponden al mediodía.

### 4.7.3. Radiación solar

La Figura 4.19 muestra las mediciones de radiación solar tanto de la estación de la Microrred (color amarillo) como de la estación implementada (color azul). En esta se observa que los momentos de mayor radiación corresponden a las horas cercanas al mediodía, con picos de radiación de gran amplitud en estas horas. Ambos sensores presentan perfiles de radiación similar en rangos por debajo de  $600 \text{ W/m}^2$ . Sin embargo, existe una diferencia de aproximadamente el 10% cuando se excede este valor. Dicha diferencia se atribuye al desgaste natural del sensor de la estación implementada (el sensor fue recuperado de una estación antigua).

La Figura 4.20 ilustra el nivel de correlación entre las mediciones de la estación implementada y la estación de la Microrred. El coeficiente de correlación obtenido es de 0.96, lo cual indica una correlación fuerte entre ambas mediciones. Es importante destacar que existen varias mediciones que difieren entre las dos estaciones, principalmente sobre los  $600 \text{ W/m}^2$  como ya se explicó.

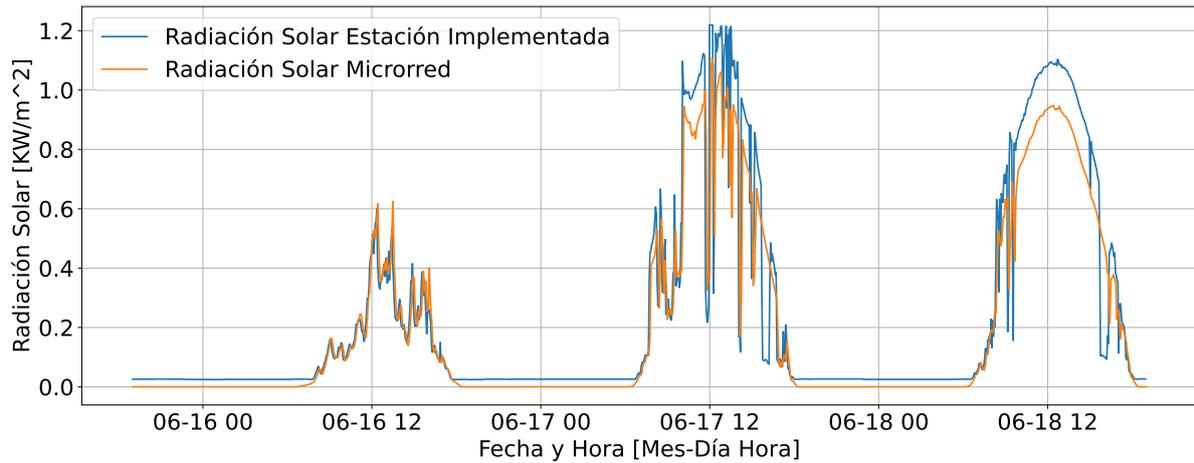


Figura 4.19: Radiación solar de la estación hidrometeorológica implementada vs la estación de la Microrred.

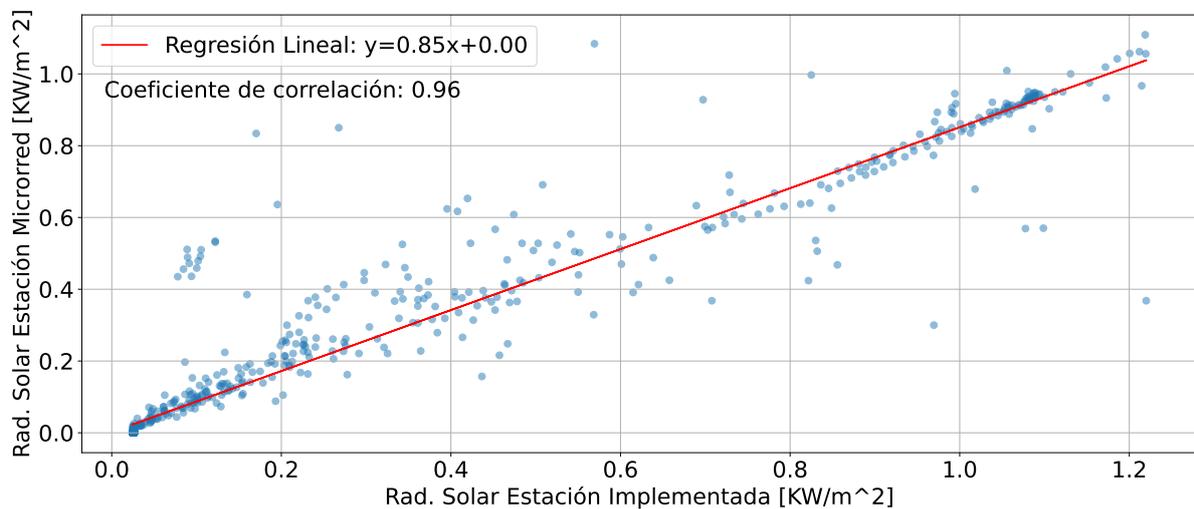


Figura 4.20: Correlación de la radiación solar de la estación hidrometeorológica implementada vs la estación de la Microrred.

#### 4.7.4. Radiación UV

La estación de la Microrred no cuenta con un sensor específico para radiación UV, y no existe información de sensores con características similares en la zona. Por tanto, no es posible realizar una comparación de validación de las mediciones. Sin embargo, es posible comparar con las mediciones de radiación solar, ya que la radiación UV medida por el sensor comienza en una longitud de onda de 400 nanómetros, y el espectro medido por el sensor de radiación solar alcanza a medir una longitud de onda de hasta 360 nanómetros. En la Figura 4.21 se presenta las mediciones de

radiación solar en color amarillo y eje vertical derecho de referencia, mientras que las mediciones de radiación UV se presentan en color azul con el eje vertical izquierdo de referencia. Entonces, es posible comparar las horas pico de radiación solar con las mediciones de radiación UV, donde es evidente la cercana relación en la respuesta de ambos sensores.

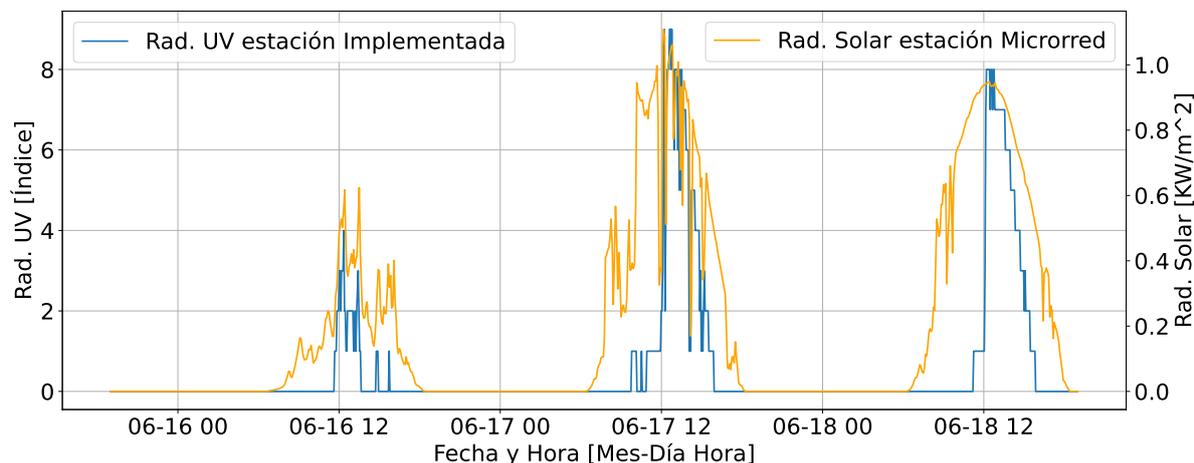


Figura 4.21: Radiación UV de la estación implementada vs la estación de la Microrred.

#### 4.7.5. Dirección y velocidad del viento

En cada ciclo de captura de datos, la estación implementada dedica 10 segundos para capturar la información correspondiente a la velocidad de viento. Las mediciones capturadas por la estación implementada (color azul) y la estación de la Microrred (color amarillo) se presentan en la Figura 4.22. En esta se evidencia una clara discrepancia entre las dos estaciones. La estación implementada captura mayor velocidad de viento en comparación con la estación del Laboratorio de Microrred. Esto se ratifica con el coeficiente de correlación de 0.6, mostrado a detalle en la Figura 4.23. Según la Tabla 3.3 esto indica una fuerte correlación.

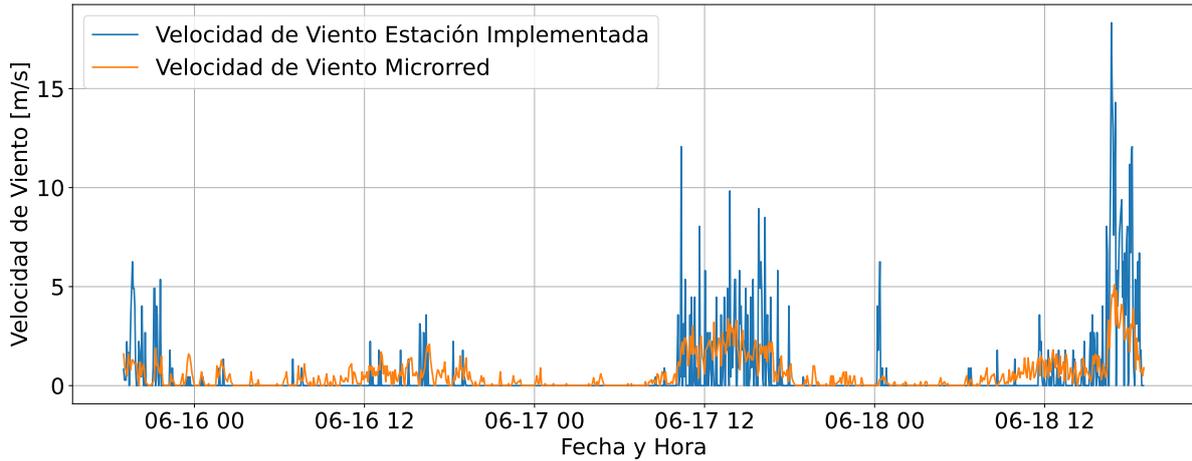


Figura 4.22: Velocidad de viento de la estación hidrometeorológica implementada vs la estación de la Microrred.

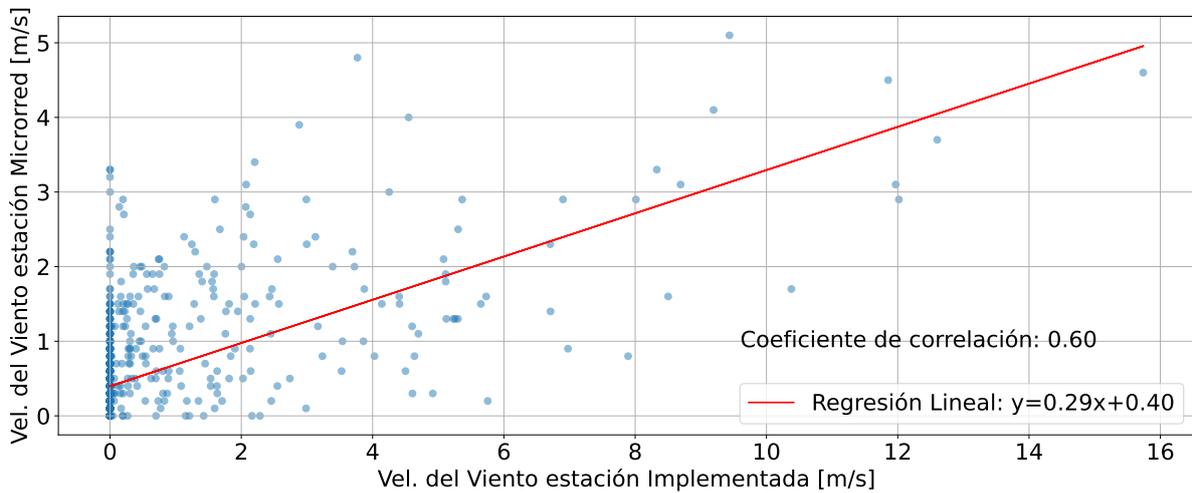


Figura 4.23: Correlación de la velocidad de viento de la estación hidrometeorológica implementada vs la estación de la Microrred.

Sucede algo similar con los resultados obtenidos en las mediciones de dirección de viento (Figura 4.24). Estos resultados muestran un coeficiente de correlación de 0.32 (Figura 4.25), lo que resulta en una correlación moderada. Esta diferencia se atribuye a la leve distancia de ubicación entre los anemómetros de cada una de las estaciones. La estación de la Microrred se encuentra instalada en la azotea del edificio. En una zona en donde no existe mucha incidencia de las estructuras cercanas en las mediciones, la ubicación se muestra en la Figura 4.26. Sin embargo, por motivos de la infraestructura disponible en el techo de la Microrred, el anemómetro de la estación implementada se ubicó en una parte que tiene influencia por parte de estructuras cercanas, además de encontrarse a menor altitud (Figura 4.27). Por tanto, las medi-

ciones de dirección y velocidad de viento se ven claramente afectadas y no tiene un perfil parecido a la estación de la Microrred.

Adicionalmente, es importante mencionar que el anemómetro usado se mantuvo en funcionamiento desde hace 7 años, por lo que fue notable el desgaste. Previamente, se realizó el respectivo mantenimiento; sin embargo, no fue posible recuperar su condición original de funcionamiento.

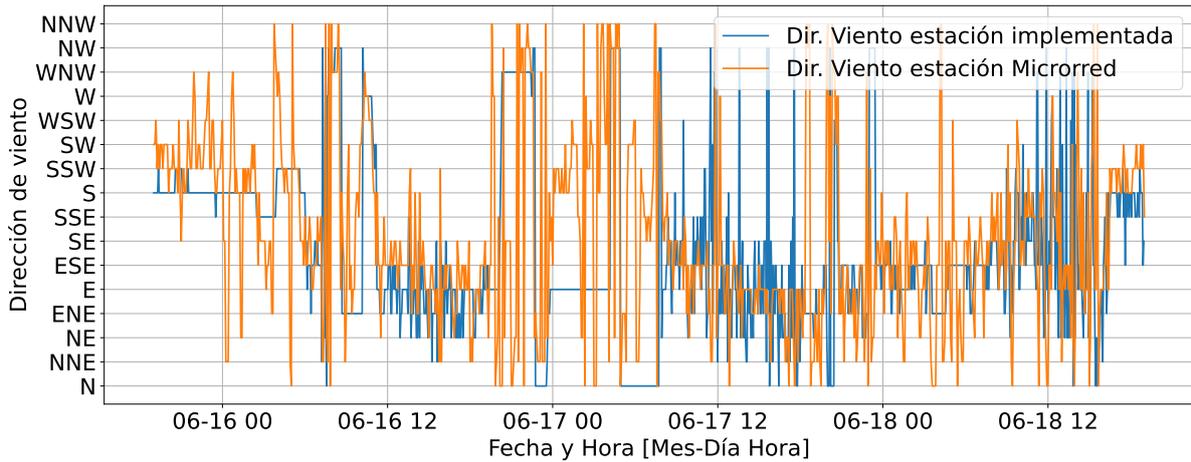


Figura 4.24: Medidas de dirección de viento de la estación hidrometeorológica implementada vs la estación de la Microrred.

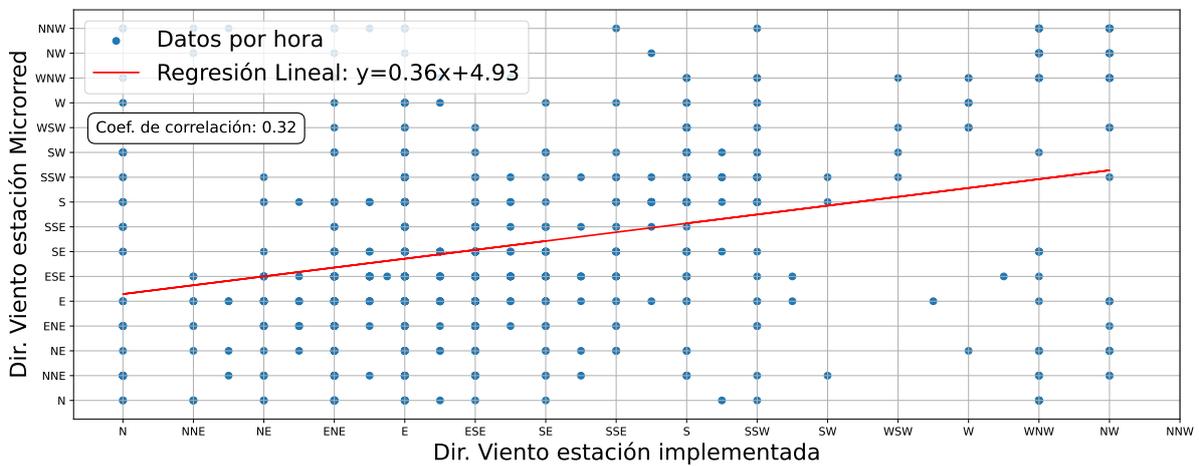


Figura 4.25: Correlación de la dirección del viento de la estación hidrometeorológica implementada vs la estación de la Microrred.



Figura 4.26: Ubicación de la estación de la Microrred.



Figura 4.27: Ubicación del anemómetro de la estación implementada.

#### 4.7.6. Precipitación

Para la medición de lluvia se utiliza un sensor que utiliza un balancín que marca un pulso cada vez que se llena. Es importante mencionar que el tamaño de los balancines es diferente en cada estación. Debido a la diferencia, el pulso de la estación implementada representa una cantidad de lluvia de 0.25 mm. Mientras que, la cantidad de lluvia para generar un pulso en la estación de la Microrred es de 0.5 mm. La Figura 4.28 muestra la comparación de la cantidad de agua recolectada por hora por el pluviómetro de la estación implementada (color azul) y la estación de la Microrred (color amarillo). Adicionalmente, se realizó el cálculo de la correlación usando intervalos de una hora de muestras. Con esto se obtuvo una correlación de 0.94 y el gráfico presentado en la Figura 4.29.

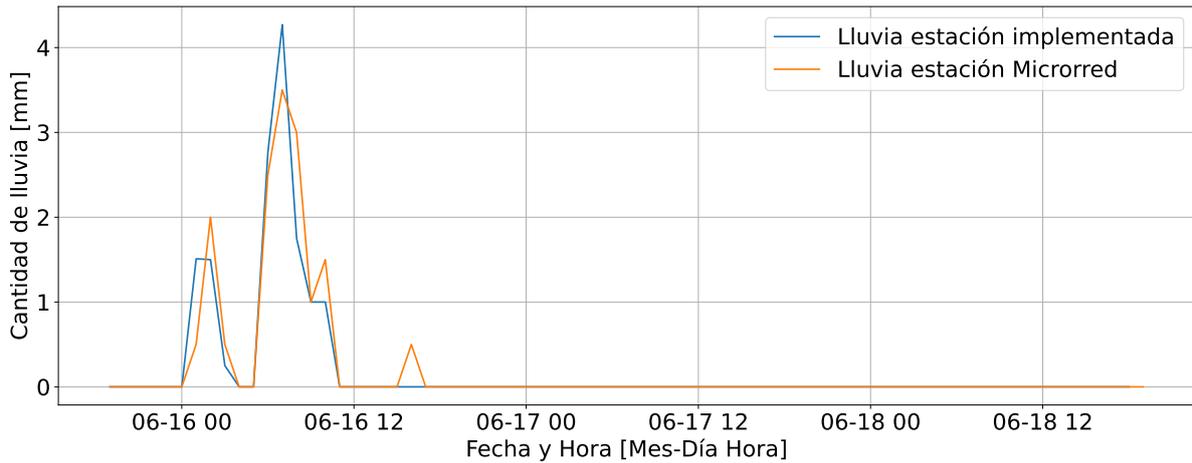


Figura 4.28: Cantidad de lluvia de la estación hidrometeorológica implementada vs la estación de Microrred.

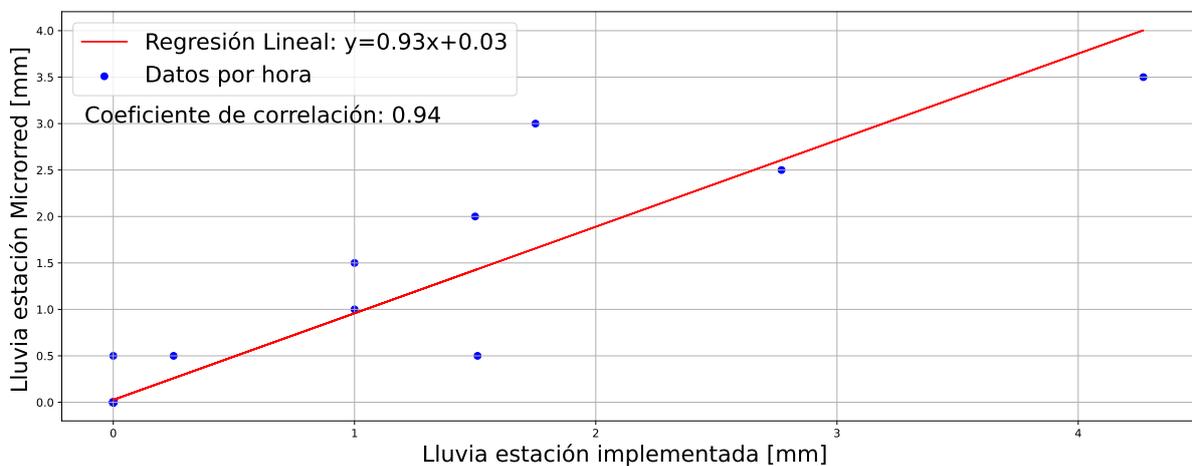


Figura 4.29: Correlación de la cantidad de lluvia de la estación hidrometeorológica implementada vs la estación de la Microrred.

Del mismo modo se presenta resultados acumulados por horas de precipitación de la Figura 4.28, tomando el periodo lluvioso del día 16 de junio. El resultado se presenta en la Tabla 4.3. Estos resultados evidencian que existe una diferencia de 0.45 mm, es decir, del 3.1%. Esto es un valor tolerable, debido a que se encuentra dentro del rango de exactitud de 5% presentado en la hoja de datos del pluviómetro usado [52].

Tabla 4.3: Total de agua recolectada por el pluviómetro de la estación implementada vs la estación de la Microrred

Estación	Rango: 00h00 a 04h00	Rango: 06h00 a 09h00	Rango: 09h15 a 11h00	Total
Implementada	3.26 mm	8.79 mm	2.0 mm	14.05 mm
Microrred	3.0 mm	9 mm	2.5 mm	14.5 mm

## Conclusiones, recomendaciones y trabajos futuros

### 5.1. Conclusiones

El microcontrolador ESP32-S3 manejó un programa que integra múltiples tecnologías y protocolos en un solo sistema. Su capacidad para operar en modo de bajo consumo energético permitió una transmisión eficiente de los datos recopilados. La implementación del modo *deep sleep* redujo el tiempo de funcionamiento activo del microcontrolador al 11%. Además, el microcontrolador ejecutó el programa utilizando menos del 10% de la frecuencia disponible de CPU, lo que redujo el consumo energético a menos de la mitad, comparado con el consumo al usar la frecuencia de 240 MHz.

El protocolo LoRaWAN posibilitó la comunicación de los datos de las variables hidrometeorológicas de extremo a extremo utilizando VMTP. VMTP redujo la cantidad de información transmitida gracias a su disposición de datos en la trama. Además, permitió transportar variables para control y verificación de errores. El uso de campos como la bandera de nulos facilitó la verificación del funcionamiento de los sensores en el transcurso del proyecto.

Los módulos PCB simplificaron la conexión con los sensores de la estación hidrometeorológica, permitiendo una correcta recolección de datos. Todos los componentes, excepto los *headers* y puertos, fueron montados utilizando tecnología de montaje superficial. Se construyeron dos módulos PCB de doble capa: el primero se encarga de la comunicación y el procesamiento del sistema, mientras que el segundo permite la conexión con periféricos y sensores para la recolección de datos. Esta estructura garantiza la resistencia y utilidad de la solución implementada.

Las pruebas de transmisión realizadas con el dispositivo demostraron fidelidad en la recopilación de datos hidrometeorológicos y un alto grado de correlación con los datos recopilados por la estación hidrometeorológica de la Microrred. La conexión con el servidor TTS y el *dashboard* permitió el monitoreo en tiempo real de la información recopilada, verificando así la correcta transmisión de datos utilizando los protocolos LoRaWAN y VMTP.

En conclusión, en este trabajo de titulación se implementó un dispositivo de datos hidrometeorológicos que integra tecnología LoRa en un microcontrolador ESP32, y

diversos sensores para la medición de variables hidrometeorológicas, destinado a modernizar el sistema de registro hidrometeorológico del PROMAS.

## 5.2. Recomendaciones

Los sensores que utilizan elementos mecánicos para las mediciones son propensos a presentar variaciones del valor real en su medición. Por lo que, se recomienda realizar un acondicionamiento de los sensores por cada año de uso, en el que se incluye la correcta lubricación del anemómetro y limpieza del balancín en el caso del pluviómetro.

En el momento de instalación de la estación hidrometeorológica, se recomienda colocarla alejada de estructuras que pudieran interferir en la medición de los sensores, principalmente el anemómetro. Así también, de contar con línea de vista directa hacia el *gateway* siempre que sea posible. Esto minimiza la pérdida de paquetes y asegura una mejor recepción de señal.

En cuanto a la frecuencia funcionamiento, el microcontrolador ESP32-S3 permite manejar una frecuencia de funcionamiento de 10 MHz. Pese a que dicha frecuencia pudiese resultar beneficiosa en términos de ahorro energético, se han evidenciado dificultades en la conexión a TTS usando LoRaWAN, por lo que se recomienda trabajar con frecuencias de 20 MHz o superiores.

Debido a la relativamente reciente aparición del módulo ESP32-S3 en el mercado a la fecha, se recomienda verificar la compatibilidad de las librerías con el módulo. Además, no modificar el sistema mediante UART mientras la estación se encuentra energizada con su fuente de alimentación, ya que esta comunicación posee su propia línea de voltaje.

## 5.3. Trabajos futuros

Durante el desarrollo del trabajo de titulación surgieron varias opciones en las que se puede trabajar para mejorar el desempeño del sistema.

- Almacenar los datos en una memoria Secure Digital (SD), ya que el sistema

implementado no cuenta con un respaldo de información en caso de fallo de la comunicación con el servidor TTS. La PCB cuenta con puertos adicionales I2C que permiten la comunicación con la nueva memoria sin requerir cambios en el hardware.

- Implementar un contador S-35770 o similar para la medición de velocidad del viento. Actualmente, se dedica tiempo de procesamiento al conteo de la velocidad del viento utilizando interrupciones. De ser necesario, se podría emplear un esquema similar al de la recolección de datos del pluviómetro, eliminando así la carga computacional del conteo en el ESP32.
- Implementar el protocolo VMTP versión 2. La nueva versión del protocolo VMTP incluye mejoras que aumentarían la eficiencia en la transmisión de información.
- Con el objetivo de alargar el tiempo de funcionamiento del sistema, se recomienda implementar un sistema de carga de la batería que alimenta al sistema. Los paneles solares representan una valiosa herramienta para suplir esta necesidad.

## Referencias

- [1] L. Alliance, “What is lorawan® specification,” 2024. [En línea]. Disponible: <https://lora-alliance.org/about-lorawan/>
- [2] U. Nanda y S. K. Pattnaik, “Universal asynchronous receiver and transmitter (uart),” in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 01, 2016, pp. 1–5.
- [3] H. Tailor, “Master-slave spi communication and arduino spi read example,” 02 2022. [En línea]. Disponible: <https://www.makerguides.com/es/master-slave-spi-communication-arduino/>
- [4] J. Carlos, “Introducción. en el mundo del desarrollo y despliegue de software, la “contenerización” ha surgido como una tecnología transformadora.” Oct 2023.
- [5] thingsboard, “What is thingsboard?” 2017. [En línea]. Disponible: <https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>
- [6] S. S. Experience™, “La rosa de los vientos,” <https://www.sicilysailingexperience.com/la-rosa-de-los-vientos/>, 2019.
- [7] A. Hector, “Lorawan – parte 3 – arquitectura y tramas,” 2019. [En línea]. Disponible: <https://elbosquedesilicio.es/lorawan3/>
- [8] J. D. H. Lalinde, F. E. Castro, J. E. Rodríguez, J. G. C. Rangel, C. A. T. Sierra, M. K. A. Torrado, S. M. C. Sierra, y V. J. B. Pirela, “Sobre el uso adecuado del coeficiente de correlación de pearson: definición, propiedades y suposiciones,” *Archivos venezolanos de Farmacología y Terapéutica*, vol. 37, num. 5, pp. 587–595, 2018.
- [9] U. de Cuenca, “Promas,” <https://www2.ucuenca.edu.ec/ingenieria/departamentos/departamento-ingenieria-civil/promas>, 2022, accessed: 2024-06-05.
- [10] A. Vázquez-Rodas, D. Astudillo, y I. Minchala, “Aplicación de tecnologías inalámbricas al monitoreo climatológico en la cuenca del río paute,” *Revista de Investigación en Tecnologías de la Información*, vol. 9, pp. 89–96, 01 2021.

- [11] J. S. Rueda y J. M. T. Portocarrero, “Similitudes y diferencias entre redes de sensores inalámbricas e internet de las cosas: Hacia una postura clarificadora,” *Revista colombiana de computación*, vol. 18, num. 2, pp. 58–74, 2017.
- [12] R. K. Kodali y S. Mandal, “lot based weather station,” in *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kumaracoil, India, 2016, pp. 680–683.
- [13] M. P. Gutama Morocho y J. J. Vázquez Patiño, “Prototipo de un dispositivo de adquisición, almacenamiento y transmisión de datos meteorológicos usando los transductores de la estación davis 6162 wireless vantage pro2 plus,” Master’s thesis, Universidad de Cuenca, Cuenca, 2017. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/28600>
- [14] J. Guaman, F. Astudillo-Salinas, A. Vazquez-Rodas, L. I. Minchala, y S. Placencia, “Water level monitoring system based on lopy4 microcontroller with lora technology,” in *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Lima, Peru, 2018, pp. 1–4.
- [15] K. S. Piedra Orellana, “Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos,” Master’s thesis, Universidad de Cuenca, Cuenca, 10 2020. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/34858>
- [16] J. P. Narvárez Bernal y J. B. Tenesaca Quilambaqui, “Diseño e implementación de un protocolo de capa de aplicación para la transmisión de datos hidrometeorológicos utilizando tecnología lora,” Master’s thesis, Universidad de Cuenca, Cuenca, 04 2019. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/32407>
- [17] P. H. U. Salamea y J. L. V. Siguenca, “Diseño e implementación de un protocolo de capa de aplicación para la transmisión de datos hidrometeorológicos utilizando lorawan: caso de estudio promas,” Tesis de Pregrado, Universidad de Cuenca, Cuenca, Ecuador, jun 2023, código: TET;136, Grado Académico: Ingeniero en Electrónica y Telecomunicaciones.

- [18] PYLCOMAYO, “Hidrometeorología,” 2017. [En línea]. Disponible: <https://www.picomayo.net/hidrometeorologia>
- [19] J. Aragón-Hernández, G. Aguilar-Martínez, U. Velázquez-Ríos, M. Jiménez-Magaña, y A. Maya-Franco, “Distribución espacial de variables hidrológicas. implementación y evaluación de métodos de interpolación,” *Ingeniería Investigación y Tecnología (México)*, vol. 02, pp. 1–15, 04 2019.
- [20] Organización Meteorológica Mundial, *International Glossary of Hydrology*, 2012, vol. 469.
- [21] Meteoblue, “Variables meteorológicas,” 2024. [En línea]. Disponible: <https://content.meteoblue.com/es/investigacion-educacion/especificaciones/variables-meteorologicas>
- [22] A.-Z. Jesus, “Sistemas de comunicacion en la banda ism,” 2017.
- [23] T. T. Network, “What are lora and lorawan?” 2024. [En línea]. Disponible: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>
- [24] CatSensors, “Tecnología lora y lorawan,” 2024. [En línea]. Disponible: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan#:~:text=Largo%20alcance%2010%20a%2020%20km>
- [25] R. Gustavo, “Geolocalización en lorawan,” *Tecnología Humanizada*, vol. 2, 2019.
- [26] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, y A. F. Skarmeta, “Performance evaluation of lora considering scenario conditions,” *Sensors*, vol. 18, num. 3, 2018. [En línea]. Disponible: <https://www.mdpi.com/1424-8220/18/3/772>
- [27] LoRa Alliance, “TS009-1.2.0 Certification Protocol,” Tech. Rep., 2024, accessed: 2024-06-13. [En línea]. Disponible: <https://resources.lora-alliance.org/technical-specifications/ts009-1-2-0-certification-protocol>
- [28] Rohde-Schwarz, “Entendiendo los protocolos serie,” Master’s thesis, Ciudad de Mexico, C.P. 01210, Mexico, 04 2014. [En línea]. Disponible: <https://www.rohde-schwarz.com/lat/productos/prueba-y-medicion/essentials-test-equ>

ipment/digital-oscilloscopes/entendiendo-los-protocolos-en-serie\_254522.html#:~:text=Los%20tres%20principales%20protocolos%20serie,como%20CAN%2C%20Lly%20FlexRay.

- [29] Z. Peterson, "I2c, spi y uart: Cómo hacer el layout de estos buses comunes," *Altium Designer*, 2021. [En línea]. Disponible: <https://resources.altium.com/es/p/i2c-vs-spi-vs-uart-how-layout-these-common-buses>
- [30] N. Semiconductors, "I2c-bus specification and user manual," October 2021. [En línea]. Disponible: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [31] J. M. M. García, "Desarrollo de una plataforma de monitorización y control de un minirobot móvil basada en redes de sensores inalámbricos," Master's thesis, Spain, 2020. [En línea]. Disponible: [https://biblus.us.es/bibing/proyectos/abreproy/5009/fichero/3\\_DescripciÃnProtocolos.pdf](https://biblus.us.es/bibing/proyectos/abreproy/5009/fichero/3_DescripciÃnProtocolos.pdf)
- [32] E. Systems, "Esp32-s3 designed for aiot applications," 2024. [En línea]. Disponible: <https://www.espressif.com/en/products/socs/esp32-s3>
- [33] Espressif Systems, "Esp32-s3 series comparison," 2024, último acceso: 26 de mayo de 2024. [En línea]. Disponible: <https://docs.espressif.com/projects/esp-idf/en/v5.0/esp32s3/hw-reference/chip-series-comparison.html>
- [34] G. L. De la Bastida Sornoza y A. P. Zhinin Gómez, "Análisis comparativo del uso de máquinas virtuales y docker para el despliegue de aplicaciones de software: caso de estudio aplicación de un gestor documental," B.S. thesis, Ecuador: Universidad Técnica De Cotopaxi Facultad De Ciencias De La . . . , 2022.
- [35] Z. K. Hussein, H. J. Hadi, M. R. Abdul-Mutaleb, y Y. S. Mezaal, "Low cost smart weather station using arduino and zigbee," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, num. 1, p. 282, 2020.
- [36] T. W. Channel. (2024) The weather channel. [En línea]. Disponible: <https://weather.com>
- [37] M. Djordjevic y D. Dankovic, "A smart weather station based on sensor technology," *Facta Universitatis, Series: Electronics and Energetics*, vol. 32, num. 2, pp. 195–210, 2019.

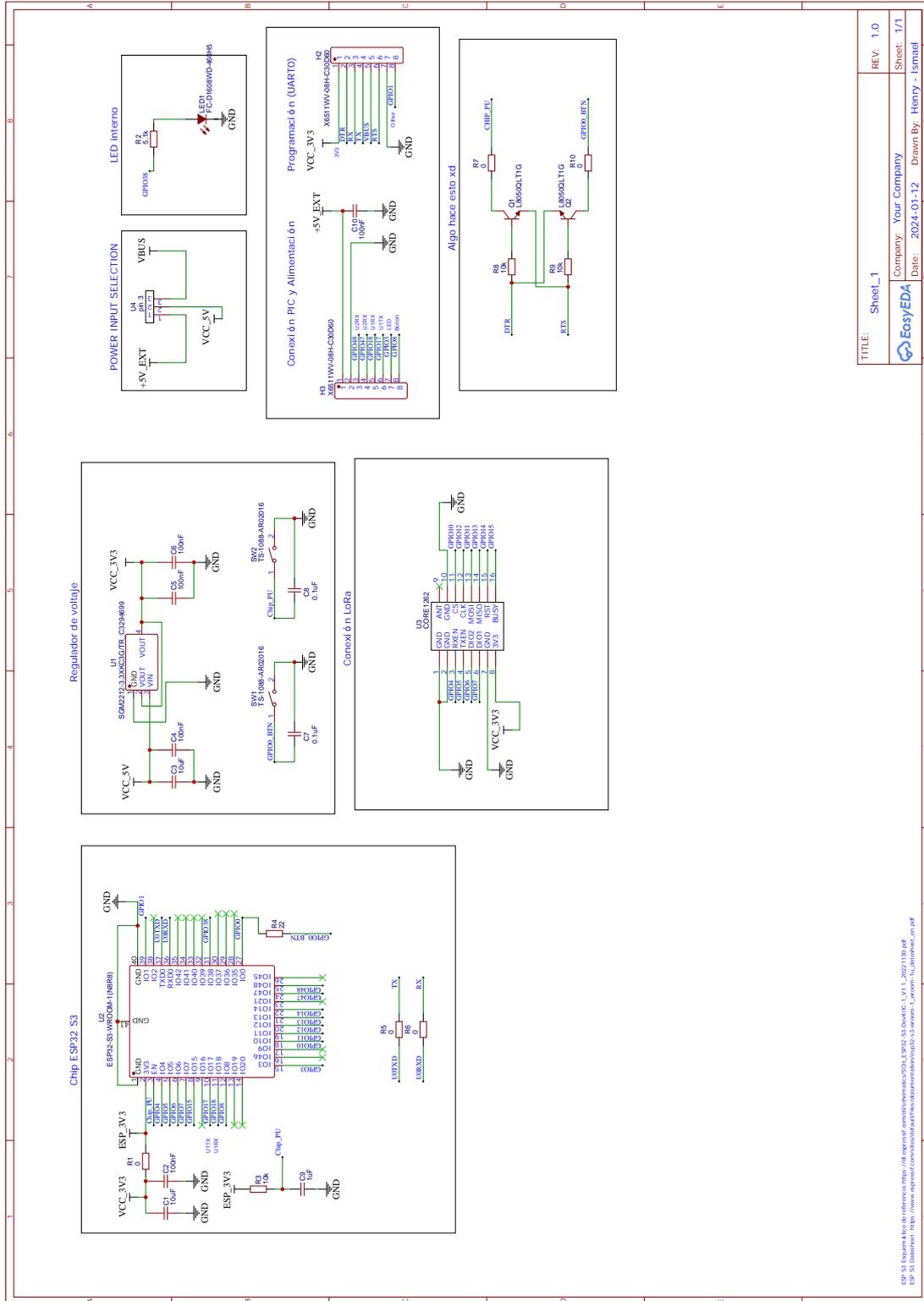
- [38] J. G. Puente, J. Andres, y M. L. L. Hernán, “Desarrollo de un pluviómetro de bajo costo para monitorización y alerta de precipitaciones utilizando tecnología lora en la parroquia de alangasí,” 2022. [En línea]. Disponible: <http://dspace.uazuay.edu.ec/handle/datos/9998>
- [39] M. Paúl y J. José, “Prototipo de un dispositivo de adquisición, almacenamiento y transmisión de datos meteorológicos usando los transductores de la estación davis 6162 wireless vantage pro2 plu,” *Ucuenca.edu.ec*, 2017. [En línea]. Disponible: <http://dspace.ucuenca.edu.ec/handle/123456789/28600>
- [40] EasyEDA, “Easyeda,” 2024. [En línea]. Disponible: <https://sites.google.com/view/caracterizaciondesimuladores/home/easyeda>
- [41] *ADS111x Ultra-Small, Low-Power, I2C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator*, Texas Instruments, 1 2018, rev. D.
- [42] A. Inc., *S-35770 COUNTER IC WITH 2-WIRE (I 2C-bus) INTERFACE*, datasheet. [En línea]. Disponible: [https://www.ablic.com/en/doc/datasheet/counter\\_ic/S35770\\_I\\_E.pdf](https://www.ablic.com/en/doc/datasheet/counter_ic/S35770_I_E.pdf)
- [43] *Extremely Accurate I2C-Integrated RTC/TCXO/Crystal*, Analog Devices, 12 2015, rev. 10.
- [44] *MIC545A/2549A Programmable Current-Limit High-Side Switch*, Microchip Technology Inc., 6 2011, dS20005722A.
- [45] E. Systems, “Esp32 hardware design guidelines: Pcb layout design,” <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/pcb-layout-design.html>, 2023, accessed: 2024-06-05.
- [46] D. Instruments, *Vantage Pro2™ Accessories: Solar Radiation Sensor 6450*, datasheet. [En línea]. Disponible: [https://cdn.shopify.com/s/files/1/0515/5992/3873/files/6450\\_SS.pdf](https://cdn.shopify.com/s/files/1/0515/5992/3873/files/6450_SS.pdf)
- [47] —, *Vantage Pro2™ Accessories: UV Sensor 6490*, 2024, datasheet. [En línea]. Disponible: [https://cdn.shopify.com/s/files/1/0515/5992/3873/files/6490\\_SS.pdf](https://cdn.shopify.com/s/files/1/0515/5992/3873/files/6490_SS.pdf)

- [48] —, *Anemometer 6410 Datasheet*, Davis Instruments, 2024, accedido el 26 de mayo de 2024. [En línea]. Disponible: [https://www.davisnet.dk/pub/media/pdf/6410\\_SS.pdf](https://www.davisnet.dk/pub/media/pdf/6410_SS.pdf)
- [49] *Data Sheet AHT25: Humidity and Temperature Module*, Asair, 2024, datasheet. [último acceso: 26 de mayo de 2024]. [En línea]. Disponible: <http://aosong.com/userfiles/files/media/Data%20Sheet%20AHT25%20A2.pdf>
- [50] B. E. Ella Davies. (2015, 10) ¿dónde está el lugar más ventoso de la tierra? [En línea]. Disponible: [https://www.bbc.com/mundo/noticias/2015/10/151015\\_vert\\_earth\\_finde\\_lugar\\_mas\\_ventoso\\_yv#:~:text=El%2010%20de%20abril%20de,se%20haya%20registrado%20alguna%20vez.](https://www.bbc.com/mundo/noticias/2015/10/151015_vert_earth_finde_lugar_mas_ventoso_yv#:~:text=El%2010%20de%20abril%20de,se%20haya%20registrado%20alguna%20vez.)
- [51] K. S. P. Orellana, “Diseño e implementación de un registrador de datos para la adquisición de datos pluviométricos,” *Universidad de Cuenca*, p. 109, 2020.
- [52] D. Instruments, *Stand Alone Rain Collectors*, datasheet. [En línea]. Disponible: [https://cdn.shopify.com/s/files/1/0515/5992/3873/files/DS6464\\_6466\\_Rev\\_B\\_web.pdf?v=1668040716](https://cdn.shopify.com/s/files/1/0515/5992/3873/files/DS6464_6466_Rev_B_web.pdf?v=1668040716)
- [53] N. Pool, “Ntp pool project,” <https://www.ntppool.org/es/>, 2024.
- [54] J. Gromes, “Radiolib: Universal wireless communication library for embedded devices,” <https://github.com/jgromes/RadioLib>, 2024, accessed: 2024-06-17.
- [55] StevenCellist, “Persistence support for radiolib,” 2024, accessed: 2024-06-16. [En línea]. Disponible: <https://github.com/radiolib-org/radiolib-persistence>
- [56] “Thingsboard - open-source iot platform,” accessed: 2024-06-17. [En línea]. Disponible: <https://thingsboard.io/>
- [57] C. A. A. Aguilar, A. M. V. Rodas, y A. F. A. Pacheco, “Evaluacion del rango de transmision de lora para redes de sensores inalámbricos en un ambiente rural de gran extension y baja densidad de árboles.”

Anexos

Anexo A: Esquemáticos de módulos

ESP 32 S3 VROOM 1 N8R8



ESP 32 Esquema Eico de referencia: <https://es.aliexpress.com/item/1005004565046464/68011120211300.pdf>  
 ESP 32 Distribución: <https://www.aliexpress.com/item/1005004565046464/68011120211300.pdf>

TITLE: Sheet_1	REV: 1.0
Company: Your Company	Sheet: 1/1
Date: 2024-01-12	Drawn By: Henry - Ismael

Figura A.1: Esquema del módulo de comunicación.

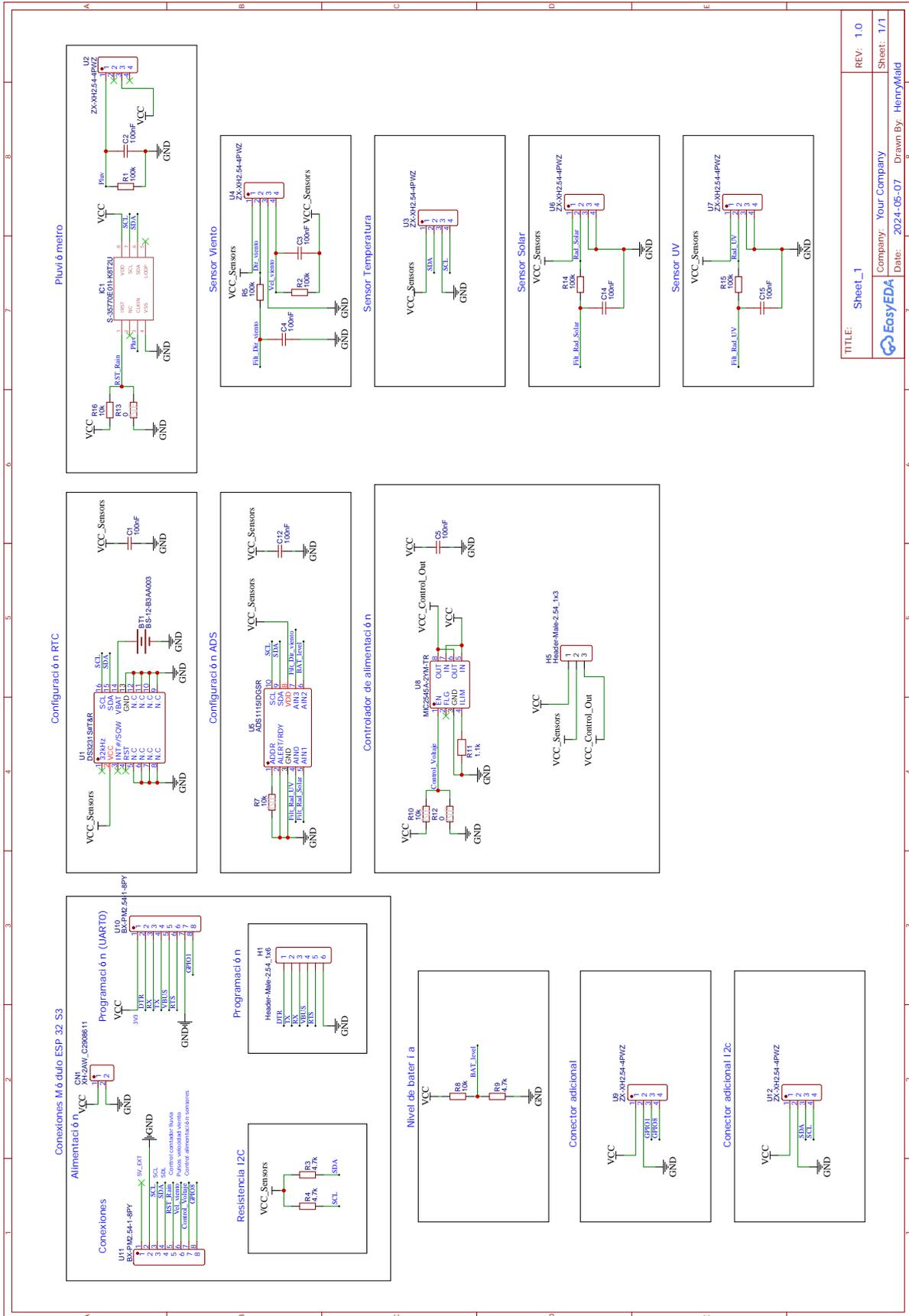


Figura A.2: Esquema del módulo de conexión de sensores.

## Anexo B: Conexión con el servidor TTS

En esta sección se presenta el procedimiento que se debe seguir para agregar un dispositivo y obtener los datos de aprovisionamiento para que el nodo final pueda subir los datos meteorológicos a la aplicación en el servidor TTS.

A continuación se muestran los pasos para registrar un dispositivo en TTS de la Universidad de Cuenca:

1. Acceder al servidor TTS de la Universidad de Cuenca.
2. Crear una aplicación de la siguiente manera, esta permite agrupar y gestionar uno o más dispositivos.
  - Ir a la *Consola* de TTS
  - Ir a la ventana de *Applications* y seleccionar *Create Application*
  - Completar los campos que se muestran en la Figura A.3:
    - *Application ID*: Es un identificador único para tu aplicación dentro del servidor TTS. Este ID también se utilizará para referenciar tu aplicación cuando te conectes a otros servidores, como plataformas de integración IoT.
    - *Application Name*: Nombre descriptivo de la aplicación.
    - *Description*: Descripción de la aplicación.

**Create application**

Within applications, you can register and manage end devices and their network data. After setting up your device fleet, use one of our many integration options to pass relevant data to your external services.  
Learn more in our guide on [Adding Applications](#).

**Application ID \***

**Application name**

**Description**

Optional application description; can also be used to save notes about the application

Figura A.3: Configuración de la Aplicación

3. Una vez creada la aplicación se puede agregar un nuevo dispositivo, para ello hay que ingresar a la aplicación creada y seleccionar *Register end device* y configurar.

- Seleccionar el método de entrada *Enter end device specifics manually*
- Seleccionar el plan de frecuencias *United States 902-928 MHz, FSB 2 (Used by TTN)*
- Seleccionar la versión de LoRaWAN, en este caso se utiliza la especificación 1.1.0 la cual es compatible con la librería *RadioLIB*, es decir, las configuraciones deben estar como se muestra en la Figura A.4.

**End device type**

Input method ⓘ

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan ⓘ \*

United States 902-928 MHz, FSB 2 (used by TTN) | v

LoRaWAN version ⓘ \*

LoRaWAN Specification 1.1.0 | v

Regional Parameters version ⓘ \*

RP001 Regional Parameters 1.1 revision A | v

Figura A.4: Configuración de frecuencia de operación y versión de LoRaWAN

- Seleccionar la versión de los parámetros regionales, la cual también es indicada en las instrucciones del uso de la librería *RadioLIB*, donde se indica que se debe seleccionar *RP001 Regional Parameters 1.1 revision A*.
- La configuración del método de activación, clase de LoRaWAN y la configuración del *cluster* se dejan por defecto, como se muestra en la Figura A.5. El método de activación seleccionado debe coincidir con el método utilizado en el nodo final.

Show advanced activation, LoRaWAN class and cluster settings ^

**Activation mode** ⓘ

Over the air activation (OTAA)

Activation by personalization (ABP)

Define multicast group (ABP & Multicast)

**Additional LoRaWAN class capabilities** ⓘ

None (class A only) | v

**Network defaults** ⓘ

Use network's default MAC settings

**Cluster settings** ⓘ

Skip registration on Join Server

Figura A.5: Configuración del método de activación, la clase de LoRaWAN y cluster.

- Se crea la información de aprovisionamiento, para lo cual es necesario ingresar los siguientes parámetros:
  - *JoinEUI*: Identificador único usado por el Servidor Join durante la activación del dispositivo. Según [54] este parámetro puede ser completado con ceros, sin embargo, para esta implementación se ha usado un generador de claves aleatorias para completar este parámetro.
  - *DevEUI*: Se trata de un identificador único para un nodo final. Puede ser generado aleatoriamente desde TTS seleccionando el botón *Generate* o usando un generador de claves aleatorias.
  - *AppKey*: Es una clave de cifrado específica del dispositivo final utilizada durante OTAA para la aplicación. Esta se genera aleatoriamente desde TTS.
  - *NwkKey*: Una clave de cifrado específica del dispositivo final utilizada para derivar FNwkSintKey, SNwkSintKey, NwkSEncKey en LoRaWAN 1.1. Cuando un dispositivo compatible con LoRaWAN 1.1 se conecta a un servidor de red LoRaWAN 1.0 que no admite claves raíces duales, el valor de NskKey se utiliza como valor de AppKey. Esta clave es

generada automáticamente en TTS al dar *clic* en el botón *Generate*.

- *End device ID*: Un identificador único y legible por humanos para su dispositivo final.
- Finalmente, luego de las configuraciones deben estar como se muestra en la Figura A.6. Y se debe seleccionar *Register end device*.

Provisioning information

JoinEUI ⓘ \*

92 FD 8B 78 CC 91 2E 7B

This end device can be registered on the network

DevEUI ⓘ \*

70 B3 D5 7E D0 06 86 D4  10/50 used

AppKey ⓘ \*

3A 1E 51 42 76 1F 62 C2 67 D2 FF E1 16 EF B5 EB

NwkKey ⓘ \*

5A 56 33 16 20 91 B1 00 F7 68 64 54 C2 3B 65 A8

End device ID ⓘ \*

nodo-final

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

Figura A.6: Configuración de aprovisionamiento.

## Anexo C: Creación de un dispositivo en Thingsboard

Para obtener los datos en Thingsboard se crea un nuevo dispositivo. A continuación se describe el proceso:

1. Acceder al panel principal de Thingsboard.
2. Seleccionar la opción Dispositivos, esta sección gestiona todos los dispositivos.
3. Seleccionar el icono + y Agregar un nuevo dispositivo.
4. En el formulario disponible se completa con el nombre que se desea asignar al dispositivo, en este trabajo se denominó Estación Hidrometeorológica.
5. Una vez creado el dispositivo es necesario obtener el *Access Token* y la di-

rección del servidor, en este trabajo se utiliza el servidor del Laboratorio de Microrred de la Universidad de Cuenca, estas credenciales son usadas en el contenedor *Docker* que recibe la información mediante MQTT y reenvía hacia Thingsboard mediante HTTP.

## Anexo D: Decodificación de datos en *Thingsboard*

Los datos obtenidos en Thingsboard aún se encuentran codificados, por lo que, es necesario decodificarlos, con ese propósito, se sigue los siguientes pasos:

1. Ingresar al servidor de Thingsboard, en este trabajo se usa el servidor del Laboratorio Microrred de la Universidad de Cuenca.
2. Seleccionar la Cadena de Reglas, para este trabajo se ha realizado una copia de la *Root Rule Chain* a la cual se ha nombrado PROMAS (Figura A.7).

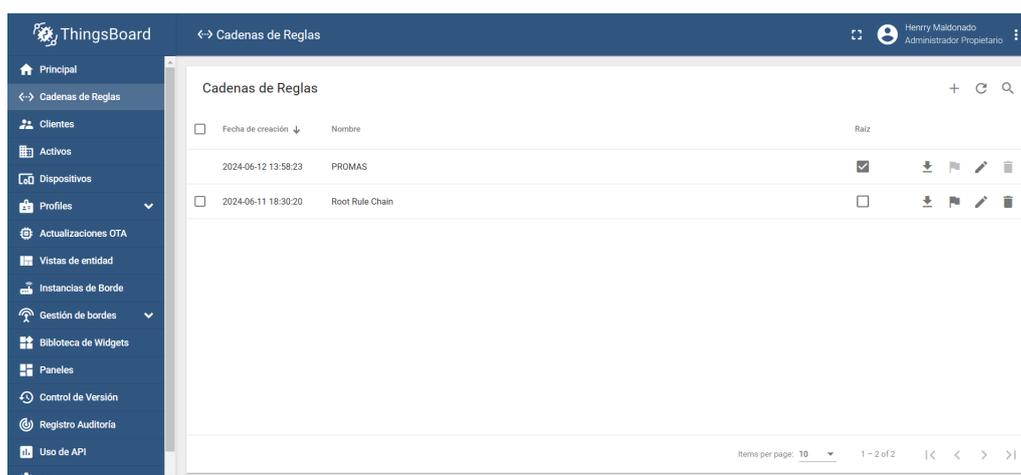


Figura A.7: Ventana y lista de cadena de reglas.

3. En esta cadena de reglas PROMAS se ha implementado el decodificador en el nodo *Script*. En la Figura A.8 Se muestra la configuración de la cadena de reglas PROMAS.
4. Para agregar el decodificador, se debe buscar el nodo de transformación *script* en la biblioteca de nodos y arrastrar el nodo a la cadena de reglas. Posteriormente, dando clic en el nodo *script* se agregó el nombre *DECODER* y se implementa el programa para decodificar las variables de interés mediante *Java Script*, según la trama VMTP definida. La entrada y salida del nodo *script* deben estar

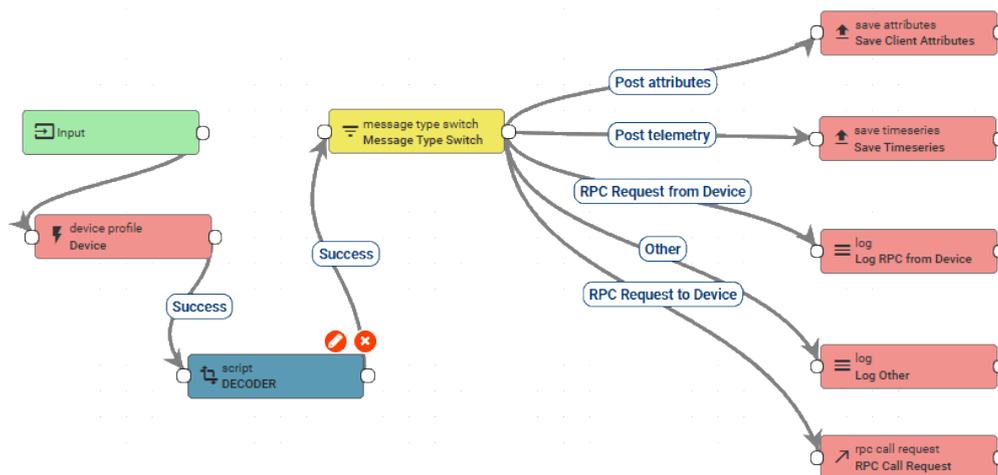


Figura A.8: Cadena de reglas PROMAS

conectadas con el nodo *Device* y el nodo *Message Type Switch* respectivamente.

5. El nodo *script* permite probar el funcionamiento del código realiza mediante una ventana de pruebas (ver Figura A.9). En esta ventana se ingresan los datos codificados como los mostrados en la Figura 4.6. El resultado muestra un mensaje de telemetría con las lecturas de los sensores, así como, las banderas de ceros, nulos, variación y *timestamp* de las muestras.
6. Una vez decodificada la información en el nodo *script* los valores de las variables meteorológicas se publican en el dispositivo.
7. Para visualizar los datos, el Thingsboard permite crear un *Dashboard*. Para ello, se debe enlazar cada una de las claves de interés que llegan al dispositivo a los *widgets* que se desea mostrar. Las variables monitoreadas en esta implementación son: temperatura, humedad, velocidad de viento, dirección de viento, radiación solar, radiación UV, precipitación, bits de nulos, bits de ceros, bit variación y marca de tiempo.

The screenshot shows a web-based interface for testing a JavaScript script. The title is "Probar Script de función (Java Script)".

**Input Section (Top Left):** Labeled "Mensaje", it contains a JSON object representing a message:

```
16 "f_cnt": 1014,  
17 "frm_payload": "hhI6+6ICiAFAAAKAA==",  
18 "decoded_payload": {  
19   "bytes": [134, 18, 58, 251, 162, 2, 136, 1, 64, 0, 0, 36, 0]  
20 },  
21 "rx_metadata": [{  
22   "gateway_ids": {  
23     "gateway_id": "eui-b827ebfffe2f0b48",  
24     "mac": "9a37e6ffff3c00a0"25   }  
26 }]
```

**Script Section (Bottom Left):** Labeled "function Transform(msg, metadata, msgType) {", it contains JavaScript code for decoding the message:

```
126 // Convertir DIR_VIENTO a texto  
127 var DIRECCION_VIENTO_TEXTUAL = ROSA_VIENTOS[DIR_VIENTO - 1]; // Ajustar el índice  
128  
129 // var5 (9 bits): presión atmosférica  
130 var PRESION = ((ocho & 0x3F) << 3) |  
131   ((nueve >> 5) & 0x07);  
132  
133 // var6 (9 bits): Precipitación  
134 var LLUVIA = ((nueve & 0x1F) << 4) |  
135   ((diez >> 4) & 0x0F);  
136 LLUVIA = LLUVIA / 100;  
137  
138 // var7 (12 bits): Rad. Solar  
139 var SOLAR = ((diez & 0x0F) << 8) |  
140   (once & 0xFF);  
141  
142 // var8 (8 bits): Rad. UV  
143 var UV = doce;  
144  
145 // Crear un nuevo mensaje con los valores decodificados y los bits de nulos y ceros  
146 newMsg = {  
147   ts: tiempoUnixMilisegundosDesde1970, // Marca de tiempo en milisegundos desde 1970  
148   values: {  
149     bandera: BANDERA,  
150     "nulos": "11110111",  
151     "Nulo0": 1,  
152     "Nulo1": 1,  
153     "Nulo2": 1,  
154     "Nulo3": 0,  
155     "Nulo4": 1,  
156     "Nulo5": 1,  
157     "Nulo6": 1,  
158     "Nulo7": 1,  
159     "ceros": "0100",  
160     "Cero0": 0,  
161     "Cero1": 0,  
162     "Cero2": 1,  
163     "Cero3": 0,  
164     "Temperatura": 12.8,  
165     "Humedad": 81,  
166     "Velocidad_Viento": 0,  
167     "Dirección_Viento": "E",  
168     "Presión_atm": 0,  
169     "Precipitación": 0,  
170     "Rad_Solar": 0,  
171     "Rad_UV": 12  
172   }  
173 }  
174 }  
175 }
```

**Output Section (Bottom Right):** Labeled "Salida", it shows the resulting JSON object:

```
1- {  
2-   "msg": {  
3-     "ts": 1719041145000,  
4-     "values": {  
5-       "bandera": 1,  
6-       "nulos": "11110111",  
7-       "Nulo0": 1,  
8-       "Nulo1": 1,  
9-       "Nulo2": 1,  
10-      "Nulo3": 0,  
11-      "Nulo4": 1,  
12-      "Nulo5": 1,  
13-      "Nulo6": 1,  
14-      "Nulo7": 1,  
15-      "ceros": "0100",  
16-      "Cero0": 0,  
17-      "Cero1": 0,  
18-      "Cero2": 1,  
19-      "Cero3": 0,  
20-      "Temperatura": 12.8,  
21-      "Humedad": 81,  
22-      "Velocidad_Viento": 0,  
23-      "Dirección_Viento": "E",  
24-      "Presión_atm": 0,  
25-      "Precipitación": 0,  
26-      "Rad_Solar": 0,  
27-      "Rad_UV": 12  
28-    }  
29-   }  
30- }
```

Buttons for "Mensaje", "Agregar", "Transformer", "Tidy", and "Test" are visible throughout the interface.

Figura A.9: Ventana de pruebas de *script* decodificador. En la parte superior izquierda se ingresa la entrada al *script*, en la parte izquierda inferior se implementa el decodificador y su salida se muestra en la parte inferior derecha.