

SISTEMA SUELO-PLANTA-ATMÓSFERA (SSPA) PROYECTO

Versión 2.0
28/04/2024

INFORMACIÓN DEL PROYECTO

Nombre completo del integrante 1	Capozzelli, Lucas Santiago
Legajo	42894
e-mail	lucascapozzelli@gmail.com
Nombre completo del integrante 2	Camargo, Julián Nicolás
Legajo	42741
e-mail	julicmrgo@gmail.com
Nombre completo del integrante 3	Tedesco, Facundo Tobares
Legajo	42742
e-mail	Facu.Tedesco96@gmail.com

Tutor	Ing. Néstor Manzur
Director	Ing. Carlos Taffernaberry
Jurado	Ing. Carlos Taffernaberry
Año Académico	2024
Responsable de la cátedra	Ing. Antonio Álvarez Abril

Empresa / Cliente / Laboratorio	Instituto Nacional de Tecnología Agropecuaria (INTA)
Patrocinador (Sponsor)	Bodega Experimental del INTA

PROYECTO FINAL

SISTEMA SUELO PLANTA
ATMÓSFERA - SSPA

Capozzelli Lucas
Tedesco Facundo
Camargo Julián

Universidad Tecnológica Nacional
Facultad Regional Mendoza
Ingeniería electrónica



1 RESUMEN DEL PROYECTO

1.1 RESUMEN

En la Bodega Experimental del Instituto Nacional de Tecnología Agropecuaria (INTA), dedicada al desarrollo sostenible del sector agropecuario, agroalimentario y agroindustrial, se llevan a cabo diversas actividades, abarcando desde la generación de información técnica hasta la promoción de productos regionales. En este contexto, el INTA busca dispositivos de medición y monitoreo para optimizar sus objetivos, dando origen a nuestro proyecto.

El proyecto se centra en desarrollar un sistema integral para medir la Evapotranspiración del cultivo (ETc), realizar un análisis de microclima y permitir un riego automático mediante sensores en el campo, que integran una estación meteorológica y un lisímetro de pesada. La auto sustentabilidad será lograda mediante la implementación de un sistema de energía solar.

La estación meteorológica contará con sensores de radiación solar, temperatura ambiente, humedad relativa, precipitación, presión atmosférica, humedad de hoja, dirección y velocidad del viento. En el lisímetro de pesada, se utilizará una celda de carga. La adquisición y transmisión de datos se llevará a cabo mediante un conversor analógico-digital y un módulo LoRa, transmitiendo hacia el módulo receptor en un edificio del INTA con conexión a internet. La información se almacenará en una base de datos y se visualizará remotamente a través de una interfaz gráfica.

La metodología de trabajo se organiza mediante herramientas como Notion y Trello, utilizando un tablero Kanban, un dashboard para la administración de documentación, un dashboard para la anotación de ideas (brainstorm) y otro para la programación de reuniones.

Los resultados esperados incluyen el acceso remoto para el usuario, la reducción de gastos y la gestión eficiente de los recursos hídricos en el sector agropecuario. El origen del proyecto radica en la necesidad de contar con un dispositivo de medición completo, confiable y de uso remoto capaz de relacionar variables del suelo, el cultivo y la atmósfera para análisis posteriores.

1.2 SUMMARY

At the Experimental Winery of the National Institute of Agricultural Technology (INTA), dedicated to the sustainable development of the agricultural, agri-food, and agro-industrial sectors, various activities are carried out, ranging from generating technical information to promoting regional products. In this context, INTA is seeking measurement and monitoring devices to optimize its objectives, giving rise to our project.

The project focuses on developing a comprehensive system to measure Crop Evapotranspiration (ETc), conduct a microclimate analysis, and enable automatic irrigation through field sensors, integrating a weather station and a weighing lysimeter. Sustainability will be achieved through the implementation of a solar energy system.

The weather station will feature sensors for solar radiation, ambient temperature, relative humidity, precipitation, atmospheric pressure, leaf humidity, wind direction, and speed. A load cell will be used in the weighing lysimeter. Data acquisition and transmission will be carried out through an analog-to-digital converter and a LoRa module, transmitting to



the receiving module in an INTA building with internet connection. Information will be stored in a database and remotely visualized through a graphical interface.

The work methodology is organized using tools such as Notion and Trello, with a Kanban board, a dashboard for document management, one for brainstorming ideas, and another for meeting scheduling.

Expected results include remote user access, reduction of expenses, and efficient management of water resources in the agricultural sector. The project originates from the need for a complete, reliable, and remotely operated measuring device capable of correlating soil, crop, and atmospheric variables for subsequent analysis.

2 PALABRAS CLAVES

Agricultura de precisión; Estación meteorológica; Lisímetro de pesada; Comunicación LoRa; Energía solar; ADCSPA; RXPSA; Gestión de datos agrícolas; Riego automático; Interfaz gráfica; Acceso remoto; Gestión eficiente de recursos.



3 ÍNDICE

1 RESUMEN DEL PROYECTO	3
1.1 RESUMEN	3
1.2 SUMMARY	3
2 PALABRAS CLAVES	4
3 ÍNDICE.....	5
4 INTRODUCCIÓN.....	8
4.1 IDEA Y DESCRIPCIÓN DEL PROYECTO	8
4.1.1 <i>Objetivo general</i>	9
4.1.2 <i>Objetivo particular</i>	9
4.2 JUSTIFICACIÓN DEL PROYECTO	10
4.2.1 <i>Antecedentes del proyecto</i>	10
4.2.2 <i>Estado actual</i>	10
4.2.3 <i>Necesidad del negocio y definición del problema</i>	11
4.2.4 <i>Beneficios del proyecto</i>	11
4.3 ALCANCE	12
4.3.1 <i>Alcance</i>	12
4.3.2 <i>Límites o fuera de alcance</i>	13
4.3.3 <i>Soluciones y entregables principales</i>	14
4.4 PLANIFICACIÓN DEL PROYECTO	15
4.4.1 <i>Cronograma</i>	16
4.4.2 <i>Hitos</i>	18
4.5 <i>RIESGO</i>	18
5 DESARROLLO DEL PROYECTO.....	21
5.1 DESARROLLO TÉCNICO	21
5.1.1 <i>Arquitectura del sistema</i>	21
5.1.1.1 Esquema general del Sistema SPA	21
5.1.1.2 Descripción general.....	21
5.1.1.3 Funcionamiento básico	22
5.1.2 <i>Equipo de energía solar</i>	23
5.1.2.1 Introducción.....	23
5.1.2.2 Componentes del equipo de energía solar del SSPA	23
5.1.2.3 Configuración e instalación	30
5.1.3 <i>Sensores y dispositivos de medición</i>	32
5.1.3.1 Estación meteorológica.....	32
5.1.3.1.1 Velocidad del viento.....	33
5.1.3.1.2 Dirección del viento	36
5.1.3.1.3 Pluviómetro	38
5.1.3.1.4 Humedad relativa	40
5.1.3.1.5 Temperatura y presión atmosférica.....	44
5.1.3.1.6 Humedad de hoja	46
5.1.3.1.7 Radiación solar	48
5.1.3.1.8 Importancia de cada medición para el análisis del microclima.....	50
5.1.3.2 Lisímetro de pesada	51
5.1.3.2.1 Desarrollo del lisímetro.....	51



5.1.4 Comunicación inalámbrica.....	62
5.1.4.1 Implementación de módulos LoRa RAK3172	62
5.1.4.1.1 Módulo RAK3172.....	63
5.1.4.1.2 Implementación de placa de desarrollo.....	66
5.1.4.2 Configuración del enlace	73
5.1.4.3 Transmisión, recepción y manejo de datos.....	78
5.1.5 Módulo ADCSPA.....	83
5.1.5.1 Hardware.....	83
5.1.5.1.1 Componentes principales	83
5.1.5.1.2 Diseño y desarrollo de la PCB	86
5.1.5.1.3 Diseño e impresión carcasa 3D	91
5.1.5.2 Firmware	96
5.1.5.2.1 Enfoque y lenguaje de programación	96
5.1.5.2.2 Justificación y análisis de códigos	97
5.1.6 Módulo RXSPA.....	100
5.1.6.1 Hardware.....	100
5.1.6.1.1 Diseño y desarrollo de la PCB	100
5.1.6.1.2 Diseño e impresión carcasa 3D	104
5.1.6.2 Firmware	108
5.1.6.2.1 Enfoque y objetivo.....	108
5.1.6.2.2 Justificación y análisis de códigos	109
5.1.7 Base de datos SPADB.....	115
5.1.7.1 Herramientas utilizadas	115
5.1.7.2 Desarrollo e implementación	117
5.1.7.3 Esquemas y tablas.....	121
5.1.8 BACKEND-SPA	123
5.1.8.1 Introducción.....	123
5.1.8.2 Arquitectura y tecnologías utilizadas.....	123
5.1.8.2.1 Uso de Node.js y Express.js	123
5.1.8.2.2 Integración con PostgreSQL (pg).....	124
5.1.8.3 Estructura del código fuente (index.js).....	125
5.1.8.3.1 Autenticación y seguridad.....	127
5.1.8.3.2 Manejo de datos.....	130
5.1.8.3.3 Servicios y Endpoints.....	132
5.1.8.3.4 Despliegue de la API	134
5.1.9 Interfaz gráfica.....	136
5.1.9.1 Introducción.....	136
5.1.9.2 Integración de Grafana con SPADB	137
5.1.9.3 Diseño de dashboards	140
5.1.9.3.1 Dashboard SSPA	141
5.1.9.3.2 Dashboard LOGS.....	148
5.1.10 Instalación.....	155
5.1.10.1 Equipo de energía solar.....	155
5.1.10.2 Sensores	156
5.1.10.3 Módulo ADCSPA	161
5.1.10.4 Módulo RXSPA	163
5.2 FACTIBILIDAD ECONÓMICA.....	165
5.2.1 Costos.....	165
5.2.2 Flujo de caja para VAN = 0	166



5.2.3 Flujo de caja para VAN > 0	168
5.2.3.1 Aproximación al valor actual neto	169
5.2.3.2 Tasa interna de retorno	169
5.2.3.3 Payback o plazo de recuperación	169
5.2.4 Productos y servicios de otros fabricantes.....	169
6 CONCLUSIONES Y ANEXOS	171
ANEXO A. Bases teóricas del Sistema SPA.	
ANEXO B. Metodología de trabajo utilizada en SSPA.	
ANEXO C. Firmware SSPA.	
ANEXO D. Software SSPA.	
7 BIBLIOGRAFÍAS Y REFERENCIAS BIBLIOGRÁFICAS.....	172



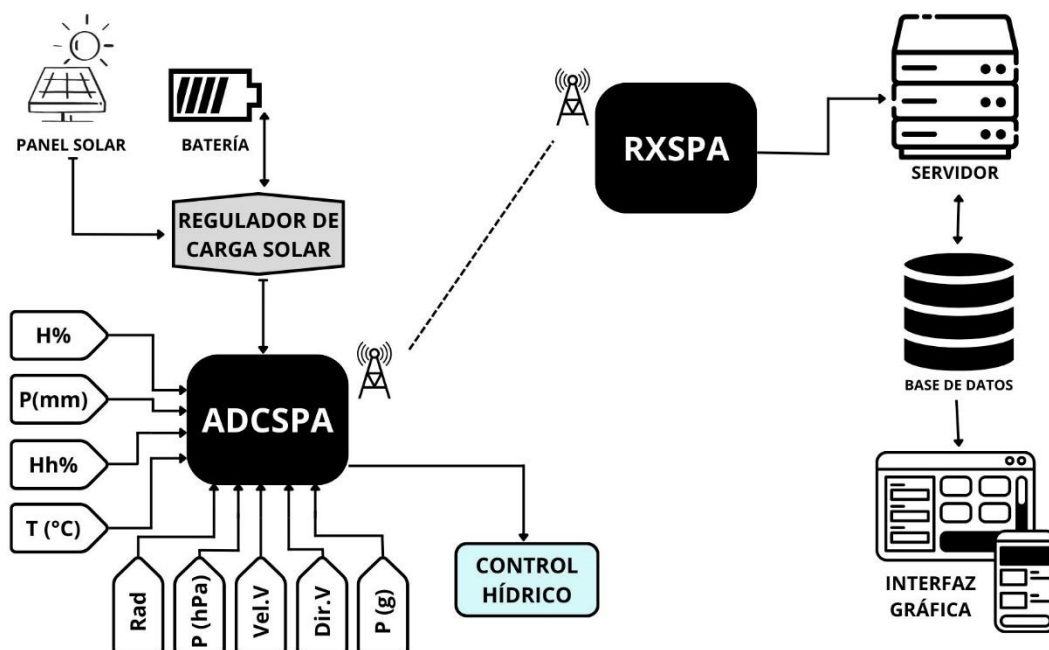
4 INTRODUCCIÓN

4.1 IDEA Y DESCRIPCIÓN DEL PROYECTO

Este proyecto, realizado por los estudiantes de la Universidad Tecnológica Nacional, Facultad Regional Mendoza, cuyos detalles se proporcionaron al principio del informe, se enfoca en la implementación del Sistema SPA (Suelo-Planta-Atmósfera). La ejecución abarcará desde fines de 2022 hasta fines de 2023 en la Bodega Experimental del INTA en Luján de Cuyo, Mendoza, Argentina.

El desafío abordado por este proyecto se centra en la necesidad de establecer un sistema que vincule de manera efectiva el estado hídrico de los cultivos con el microclima circundante. La meta es obtener datos precisos de forma remota, íntegra y confiable. Este enfoque responde a la visión de mejorar la productividad de los agricultores al proporcionarles información valiosa para la toma de decisiones. Al lograr una gestión más eficiente de los recursos hídricos, este proyecto tiene como objetivo evitar gastos innecesarios y contribuir al cuidado del agua en el proceso de producción agrícola.

El diagrama general del funcionamiento del proyecto se observa a continuación:



El **sistema SPA**, presentado de forma simplificada en el esquema, opera de manera integral para proporcionar información esencial relacionada con el entorno agrícola. El equipo de energía solar, instalado en el campo, consta de un panel solar, una batería y un regulador de carga solar. Este conjunto alimenta tanto al ADCSPA como al control hídrico y a los diversos sensores, incluyendo: humedad ambiente, pluviómetro, humedad de hoja, temperatura, radiación solar, velocidad del viento, dirección del viento y un lisímetro de pesada.

El **ADCSPA** se comunica con el **RXSPA** mediante un enlace de radio utilizando tecnología LoRa. Este enlace tiene el propósito de transmitir datos de los sensores al **RXSPA** cuando este último lo solicita. El **RXSPA**, ubicado en el edificio de geomática de la Estación Experimental Agropecuaria del INTA en Mendoza, está conectado a la red wifi del edificio. A su vez, el **RXSPA** establece una comunicación con el servidor, que tiene acceso a la base de datos del sistema SPA.



El RXSPA se comunica con el servidor, donde este último tiene la capacidad de postear la información recopilada y realizar consultas necesarias para el riego.

La **interfaz gráfica** representa la cara amigable del sistema SPA, permitiendo a los usuarios visualizar de manera intuitiva y accesible todos los datos recopilados. Desde esta interfaz, los usuarios pueden monitorear el sistema en su conjunto (datos y funcionamiento), tomando decisiones informadas para optimizar las prácticas agrícolas.

4.1.1 Objetivo general

Este proyecto tiene como objetivo general la implementación del Sistema SPA (Suelo-Planta-Atmósfera), centrándose en la adquisición precisa y eficiente de datos relacionados con el estado hídrico de los cultivos y el microclima circundante. Se busca la obtención de datos, permitiendo la monitorización remota, íntegra y confiable de estas variables clave.

Para lograr este propósito, se utilizarán herramientas macro como una estación meteorológica y un lisímetro de pesada, respaldados por tecnologías como sensores especializados, conversores analógico-digitales y módulos de transmisión LoRa. Además, se implementará un sistema de energía solar para garantizar autonomía y sostenibilidad.

La adquisición de datos se complementará con el establecimiento de una base de datos, respaldada por un servidor, y la creación de una interfaz gráfica intuitiva en Grafana. Estos componentes permitirán la organización, almacenamiento y visualización eficientes de la información recopilada. Estas son las herramientas principales para lograr el objetivo general, que es obtener un dispositivo confiable, de fácil utilización y que responda de forma adecuada a las situaciones y especificaciones propuestas.

4.1.2 Objetivo particular

El objetivo particular de este proyecto se centra en asegurar la ejecución eficiente de cada fase desarrollada, con el propósito de alcanzar el producto final que cumplirá con el objetivo general previamente establecido. Las metas específicas incluyen:

- **Estudio detallado de dispositivos:** Realizar un análisis exhaustivo de los dispositivos seleccionados, comprendiendo su funcionamiento y sus posibles limitaciones.
- **Pruebas y Validación:** Conducir pruebas exhaustivas en cada etapa del proyecto, sometiendo el dispositivo a diversas situaciones probables y adversas. Estas pruebas permitirán obtener un desarrollo completo y confiable. Los resultados de las pruebas determinarán si el dispositivo cumple con los requisitos mínimos establecidos.
- **Desarrollo de código:** Lograr un código claro, bien estructurado y robusto que aborde y resuelva posibles problemas derivados del uso del dispositivo.
- **Documentación:** Generar un documento detallado que funcione como manual de usuario. Este documento describirá como utilizar correctamente el sistema, ofrecerá recomendaciones y destacará precauciones a tener en cuenta. La documentación se diseñará para proporcionar una referencia completa y comprensible para los usuarios.



4.2 JUSTIFICACIÓN DEL PROYECTO

4.2.1 Antecedentes del proyecto

En la Bodega Experimental del Instituto Nacional de Tecnología Agropecuaria (INTA), se abordan temáticas relacionadas con el desarrollo sostenible del sector agropecuario, agroalimentario y agroindustrial mediante actividades de investigación y extensión. Las áreas de enfoque incluyen la generación de información técnica y socioeconómica, la promoción del asociativismo, el acompañamiento a productores, la experimentación adaptativa, la organización de capacitaciones y eventos, la promoción de productos regionales, la asistencia en la elaboración de proyectos asociativos, viajes de intercambio y la vinculación con actores del medio.

En este contexto, se identifica la necesidad del INTA de contar con dispositivos de medición y monitoreo que faciliten la realización de mediciones y análisis alineados con sus objetivos fundamentales. La génesis de nuestro proyecto surge en respuesta a esta demanda específica. Buscamos desarrollar un dispositivo de medición integral, confiable y de uso remoto, capaz de correlacionar variables del suelo, el cultivo y la atmósfera. Este dispositivo permitirá realizar análisis detallados para contribuir eficazmente a los objetivos de investigación y desarrollo sostenible del INTA.

4.2.2 Estado actual

En la actualidad, el Internet de las Cosas (IoT) juega un papel central en la evolución de la agricultura, introduciendo el concepto de "agricultura inteligente". Este término, emblemático de la tecnología de la información y la comunicación moderna, está revolucionando nuestras prácticas agrícolas. La monitorización online destaca como una aplicación sobresaliente del IoT, permitiendo a los agricultores supervisar parámetros cruciales como temperatura, humedad, precipitaciones, etc.

La sensorización dota de información a los agricultores, proporcionándoles datos instantáneos y precisos que se traducen en ahorros sustanciales y mejoras significativas en la productividad agrícola. La adopción de tecnologías IoT ha sido un catalizador clave para la eficiencia y sostenibilidad en el sector, facilitando una toma de decisiones más informada y una gestión más eficaz de los recursos agrícolas.

En este contexto, nuestro proyecto se posiciona estratégicamente como respuesta a estas tendencias. En el ámbito de la agricultura de precisión, donde la coexistencia de diversas tecnologías es evidente a nivel nacional e internacional, se observa una creciente demanda de soluciones que integren sistemas avanzados de medición y monitoreo para optimizar la gestión de cultivos. En Argentina y globalmente, la recopilación de datos agrícolas se ve obstaculizada por la falta de dispositivos especializados y la dependencia de métodos manuales, desafíos que nuestro proyecto aborda mediante la implementación de un sistema integral. Nuestra propuesta busca mejorar la integración y centralización de datos, facilitando la obtención eficiente de información precisa sobre el estado hídrico de los cultivos, el microclima y otras variables clave.



4.2.3 Necesidad del negocio y definición del problema

La necesidad del negocio surge principalmente en la preservación de los recursos hídricos y en el aumento de la eficiencia productiva en los cultivos. Para alcanzar estos objetivos, es imperativo optimizar el consumo de agua mediante estrategias de riego más eficientes. Asimismo, se identifica una falta de comunicación entre el lisímetro de pesada y la estación meteorológica. Esta ausencia de conexión implica que cualquier ajuste o recolección de datos debe llevarse a cabo en el sitio donde se encuentran instalados, generando costos logísticos y consumiendo tiempo.

Adicionalmente, los datos recopilados se encuentran en formato crudo y se hallan fragmentados según el dispositivo correspondiente. Este escenario demanda un proceso adicional para integrar y analizar de manera cohesiva los datos de cada dispositivo en un sistema unificado. La falta de integración dificulta la visualización remota de los datos de manera clara y precisa, impactando negativamente en la toma de decisiones dentro del ámbito de la producción agrícola. En respuesta a estos desafíos, nuestro proyecto se presenta como una solución integral, que no solo aborda estas limitaciones, sino que también mejora significativamente la eficiencia en la gestión de recursos hídricos y la toma de decisiones en el ámbito agrícola.

4.2.4 Beneficios del proyecto

Patrocinador: Bodega Experimental del INTA en Luján de Cuyo, Mendoza

- **Optimización de recursos hídricos:** La implementación del sistema SPA permitirá una gestión eficiente de los recursos hídricos en la Bodega Experimental. La monitorización precisa contribuirá a optimizar el consumo de agua, asegurando un riego más efectivo y sostenible para los cultivos.
- **Incremento de la productividad agrícola:** La mejora en la toma de decisiones, habilitada por la recopilación y análisis de datos, se traducirá en un aumento significativo de la productividad agrícola. La capacidad de correlacionar variables clave del suelo, el cultivo y la atmósfera permitirá ajustes precisos para maximizar los rendimientos.
- **Eficiencia en la operación:** La centralización y visualización remota de datos mejorarán la eficiencia operativa y facilitarán la toma de decisiones desde la distancia.
- **Fortalecimiento de la investigación y extensión:** Nuestro proyecto proporciona una plataforma para la investigación y extensión, permitiendo al INTA expandir su liderazgo en la generación de conocimiento técnico y socioeconómico. Este enfoque contribuirá a la transferencia efectiva de tecnologías a la comunidad agrícola.

Estos beneficios específicos destacan el valor estratégico del proyecto para la Bodega Experimental del INTA en Luján de Cuyo, Mendoza, brindando mejoras significativas en la gestión de recursos, la productividad agrícola y la eficiencia operativa.



4.3 ALCANCE

4.3.1 Alcance

El proyecto tiene como objetivo desarrollar e implementar un sistema integral en la Bodega Experimental del Instituto Nacional de Tecnología Agropecuaria (INTA) en Luján de Cuyo, provincia de Mendoza. El sistema, denominado Sistema SPA (Suelo-Planta-Atmósfera), se centrará en la medición y monitoreo de variables clave relacionadas con el estado hídrico de los cultivos, el microclima y otros factores relevantes para la agricultura de precisión.

El proyecto abarcará la implementación de una estación meteorológica, la cual estará equipada con sensores de radiación solar, temperatura ambiente, humedad relativa, precipitación, presión atmosférica, humedad de hoja, dirección y velocidad del viento. Además, se llevará a cabo la integración de un lisímetro de pesada que contará con una celda de carga para la medición de la evapotranspiración de los cultivos (ETc), y estará equipado con una válvula para el riego automático. Para la adquisición y transmisión de datos, se implementará un sistema de conversión analógico-digital y un módulo LoRa, posibilitando la comunicación desde la estación meteorológica y el lisímetro hacia un módulo receptor ubicado en un edificio del INTA con conexión a internet. Asimismo, se empleará una base de datos para el almacenamiento de los datos recopilados, y se desarrollará una interfaz gráfica que permitirá la visualización remota de la información recopilada.

Tareas principales y tiempo estimado de ellas:

- **Puesta en funcionamiento del sistema de energía solar y su configuración** (80 horas): Esta tarea implica la instalación y configuración del sistema de energía solar, que comprende el panel solar, el regulador de carga y la batería de 12V.
- **Relevamiento, estudio de funcionamiento, calibración y puesta en funcionamiento de sensores** (220 horas): Se llevará a cabo un proceso de relevamiento de cada sensor en la estación meteorológica. Esto incluirá un estudio detallado de su funcionamiento, calibración para asegurar la precisión de las mediciones y la puesta en funcionamiento efectiva de cada sensor.
- **Relevamiento, estudio de funcionamiento, calibración y puesta en funcionamiento del lisímetro de pesada con la válvula para el riego** (180 horas): Similar al proceso realizado para la estación meteorológica, se llevará a cabo un relevamiento, estudio de funcionamiento, calibración y puesta en funcionamiento del lisímetro de pesada. Este incluirá la integración y verificación del correcto funcionamiento de la válvula para el riego, asegurando que la medición de la cantidad de agua utilizada en el riego sea precisa y eficiente.
- **Implementación de módulos LoRa** (150 horas): Aunque se optó por utilizar un diseño de PCB preexistente recomendado por el fabricante, se asignarán horas para la construcción de estos módulos. Esta tarea implica realizar una investigación sobre los requisitos, funcionamiento y especificaciones necesarias para su construcción. Se llevarán a cabo pruebas para verificar la comunicación entre los módulos LoRa, garantizando su funcionalidad y preparándolos para la transmisión eficiente de datos en el proyecto.
- **Diseño y desarrollo del módulo ADCSPA** (150 horas): Se realizará el diseño y desarrollo de la PCB para el módulo ADCSPA, que integra un ESP32 y un módulo LoRa, permitiendo la conexión de todos los sensores de la estación meteorológica, el lisímetro de pesada y la válvula para el riego.



- **Desarrollo del firmware para el módulo ADCSPA (250 horas):** Se llevará a cabo la programación y desarrollo del firmware para el Módulo ADCSPA, que incluye la integración de los sensores de la estación meteorológica, el lisímetro de pesada y el sistema de riego. Este firmware permitirá la adquisición precisa de datos y la transmisión eficiente a través del módulo LoRa. Implementación de algoritmos de medición y desarrollo de riego automático.
- **Diseño y desarrollo del módulo RXSPA (130 horas):** Se desarrollará e implementará el Modulo RXSPA, que estará ubicado en el edificio del INTA y conectado a internet. Este módulo contendrá otro ESP32 y módulo LoRa correspondiente.
- **Desarrollo del firmware para el módulo RXSPA (250 horas):** Se realizará la programación y desarrollo del firmware destinado al Módulo RXSPA, ubicado en el edificio del INTA. Este firmware facilitará la recepción de datos transmitidos por el Módulo ADCSPA y la comunicación con el servidor, donde este último es el que tiene acceso a la base de datos.
- **Configuración y pruebas de transmisión de datos (100 horas):** Se llevarán a cabo pruebas para asegurar la eficiente transmisión de datos entre el Modulo RXSPA y el Modulo ADCSPA utilizando los módulos LoRa en modos transmisor y receptor.
- **Desarrollo de API para la base de datos (180 horas):** Se desarrollará una aplicación servidor que cumpla la función de API para facilitar la comunicación con la base de datos que estará en un servicio de alojamiento de bases de datos. Este componente será crucial para la recopilación, procesamiento y acceso a los datos almacenados.
- **Integración de la base de datos en la plataforma de alojamiento (100 horas):** Se realizarán las acciones necesarias para integrar y alojar la base de datos en un servicio de alojamiento adecuado, asegurando los datos recopilados.
- **Diseño y creación de la interfaz gráfica en Grafana (150 horas):** Se diseñará y desarrollará la interfaz gráfica en Grafana, asegurando una presentación clara y accesible de los datos recopilados.
- **Pruebas Integrales y optimización (100 horas):** Se llevarán a cabo pruebas integrales del sistema completo, identificando y corrigiendo cualquier inconveniente.
- **Documentación completa (40 horas):** Se diseñará y desarrollará el manual del Sistema SPA para su correcta utilización.

4.3.2 Límites o fuera de alcance

El proyecto establece ciertos límites para definir claramente el marco de ejecución:

- **Desarrollo del servidor:** La iniciativa no abarcará el desarrollo de infraestructura del servidor y la base de datos.
- **Latencia en la presentación de datos:** Se aclara que la presentación de datos no será en tiempo real, sino que se mostrarán con una latencia específica. Este enfoque se considera dentro de los parámetros establecidos para la entrega de información.
- **Reparación de componentes usados/dañados:** No se contempla la reparación de componentes ya instalados que puedan presentar fallas durante la ejecución del proyecto.



- Servicio técnico posterior al proyecto: El proyecto no incluirá la prestación de servicios técnicos de mantenimiento en las instalaciones una vez finalizado. Cualquier necesidad de servicio técnico posterior quedará excluida de los compromisos de este proyecto.

4.3.3 Soluciones y entregables principales

La siguiente tabla muestra un listado de los entregables del proyecto (productos o servicios)

Entregables principales	Descripción del entregable
Equipo de energía solar	Instalado en el campo, consta de un panel solar, una batería y un regulador de carga solar. Este conjunto alimenta tanto al ADCSPA como a los diversos sensores de la estación meteorológica y el lisímetro de pesada.
Estación meteorológica	Compuesta por los siguientes sensores: humedad ambiente, pluviómetro, humedad de hoja, temperatura ambiente, radiación solar, velocidad del viento, dirección del viento.
Lisímetro de pesada	Compuesto por una celda de carga para la medición de la evapotranspiración de los cultivos (ETc), y equipado con una válvula para el riego automático.
Módulo ADCSPA	Compuesto principalmente por un ESP32 y un módulo LoRa, permite la conexión de todos los sensores de la estación meteorológica, el lisímetro de pesada y la válvula para el riego. Transmite los datos recopilados hacia el RXSPA.
Módulo RXSPA	Compuesto principalmente por un ESP32 y un módulo LoRa, recibe los datos transmitidos por el módulo ADCSPA, establece comunicación con el servidor y este último accede a la base de datos.
Interfaz gráfica	Ofrece a los usuarios una experiencia visual intuitiva para la interpretación de los datos recopilados y también brinda información del funcionamiento del Sistema SPA.
Manual de usuario	Proporciona una guía detallada y accesible para la correcta utilización del sistema. Además, ofrece soluciones a posibles inconvenientes y brinda recomendaciones para optimizar la experiencia del usuario.



4.4 PLANIFICACIÓN DEL PROYECTO

La estrategia de gestión del proyecto que se seguirá corresponde a la metodología del Project Management Institute (PMI), la cual nos permitirá, mediante constante planificación y control de las etapas dentro del proyecto, alcanzar las metas y objetivos impuestos en cada una de ellas.

Para ello debemos:

- Definir el alcance de nuestro proyecto, indicando lo que nuestro producto hará y aquello que no, como también los procesos y acciones que se engloban dentro del proyecto y aquellas que quedan fuera del mismo.
- Recopilar requisitos: para ello se debe mantener una constante comunicación con los stakeholders para definir aquellos requisitos que se deben cumplir a toda costa, tanto para el producto elaborado como para la gestión y desarrollo del proyecto. Se deben tener en cuenta todos los requisitos, dándoles mayor importancia a aquellos que tengan un nivel de prioridad mayor y, en base a ellos, fijar las especificaciones que deberá tener nuestro producto.
- Elaborar un plan de alcance, mediante el cual se definirán las tareas, métodos, herramientas y los pasos a seguir para lograr los requerimientos especificados y los alcances previamente definidos.
- Crear la EDT. Luego de haber definido el alcance y lo que haremos, mediante la estructura de desglose del trabajo dividiremos nuestro proyecto en bloques manejables y les asignaremos recursos, tiempo y costos.
- Se deberán secuenciar aquellas actividades que deban realizarse en un determinado orden y estimar las duraciones de cada una de ellas.
- Elaborar un cronograma que nos permita ordenar y planificar los distintos bloques de trabajo y las actividades a lo largo del tiempo para cumplir con las fechas establecidas.
- También se deberán estimar costos en cada etapa y elaborar presupuestos.
- Se deberá especificar un plan de comunicación, para lograr que los interesados, tanto dentro como fuera del proyecto, se encuentren informados respecto del avance del proyecto.
- Por último, también se deberán identificar los posibles riesgos que pueda presentar nuestro proyecto y elaborar un plan de respuesta para evitarlos o mitigarlos tanto como se pueda.

Es importante aclarar que, aunque las distintas fases expuestas para la gestión del proyecto hayan sido presentadas en un determinado orden, no necesariamente se debe respetarlo, ya que algunas actividades pertenecientes a fases distintas pueden realizarse sin orden premeditado o al mismo tiempo.

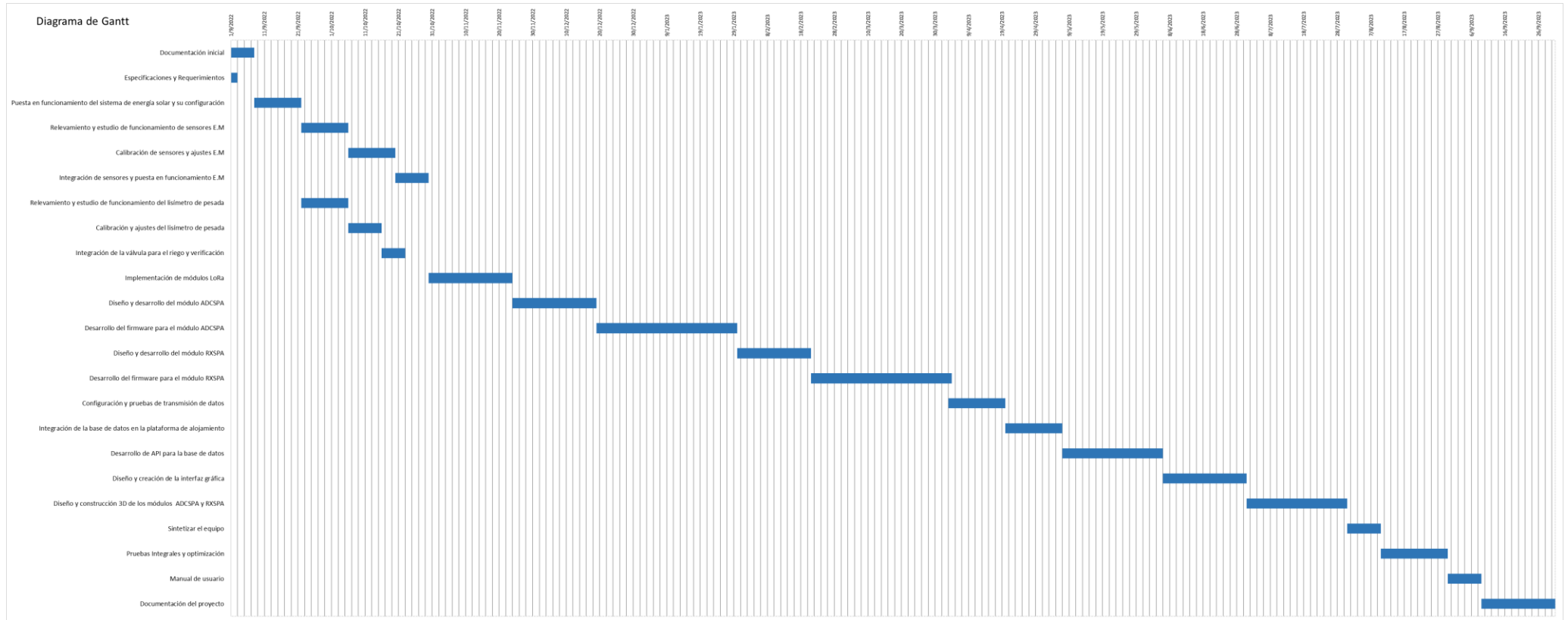
**4.4.1 Cronograma**

Tarea	Referencia	Predecesora	Fecha inicio	Duración (Horas H)	Fecha fin
Documentación inicial	A	Inicio	1/9/2022	28	8/9/2022
Especificaciones y Requerimientos	B	Inicio	1/9/2022	8	3/9/2022
Puesta en funcionamiento del sistema de energía solar y su configuración	C	A	8/9/2022	56	22/9/2022
Relevamiento y estudio de funcionamiento de sensores E.M	D	C	22/9/2022	56	06/10/2022
Calibración de sensores y ajustes E.M	E	D	06/10/2022	56	20/10/2022
Integración de sensores y puesta en funcionamiento E.M	F	E	20/10/2022	40	30/10/2022
Relevamiento y estudio de funcionamiento del lisímetro de pesada	G	C	22/9/2022	56	06/10/2022
Calibración y ajustes del lisímetro de pesada	H	G	06/10/2022	40	16/10/2022
Integración de la válvula para el riego y verificación	I	H	16/10/2022	28	23/10/2022
Implementación de módulos LoRa	J	F	30/10/2022	100	24/11/2022
Diseño y desarrollo del módulo ADCSPA	K	J	24/11/2022	100	19/12/2022
Desarrollo del firmware para el módulo ADCSPA	L	K	19/12/2022	168	30/1/2023
Diseño y desarrollo del módulo RXSPA	M	L	30/1/2023	88	21/02/2023
Desarrollo del firmware para el módulo RXSPA	N	M	21/02/2023	168	03/4/2023
Configuración y pruebas de transmisión de datos	Ñ	N	03/04/2023	68	20/04/2023
Integración de la base de datos en la plataforma de alojamiento	O	Ñ	20/04/2023	68	07/5/2023
Desarrollo de API para la base de datos	P	O	07/5/2023	120	06/6/2023
Diseño y creación de la interfaz gráfica	Q	P	06/6/2023	100	01/7/2023
Diseño y construcción 3D de los módulos ADCSPA y RXSPA	R	Q	01/7/2023	120	31/7/2023
Sintetizar el equipo	S	R	31/7/2023	40	10/8/2023
Pruebas Integrales y optimización	T	S	10/8/2023	80	30/8/2023
Manual de usuario	U	T	30/8/2023	40	09/9/2023
Documentación del proyecto	V	T, U	09/9/2023	88	01/10/2023

- Total, horas hombre: 1708 horas.



Diagrama de Gantt





4.4.2 Hitos





La tabla muestra un listado de hitos generales del proyecto y el cronograma estimado de finalización.

Hitos	Fecha de finalización
Inicio del desarrollo del proyecto	1/9/2022
Puesta en funcionamiento del sistema de energía solar y su configuración	22/9/2022
Relevamiento, estudio de funcionamiento, calibración y puesta en funcionamiento de sensores (E.M)	30/10/2022
Relevamiento, estudio de funcionamiento, calibración y puesta en funcionamiento del lisímetro de pesada con la válvula para el riego	23/10/2022
Implementación de módulos LoRa	24/11/2022
Diseño, desarrollo y programación del módulo ADCSPA	30/1/2023
Diseño, desarrollo y programación del módulo RXSPA	03/4/2023
Configuración y pruebas de transmisión de datos	20/04/2023
Desarrollo de API para la base de datos	06/6/2023
Diseño y creación de la interfaz gráfica	01/7/2023
Diseño y construcción 3D de los módulos ADCSPA y RXSPA	31/7/2023
Sintetizar el equipo	10/8/2023
Manual de usuario	09/9/2023
Finalización del proyecto	01/10/2023

4.5 RIESGO

LEYENDA							
			GRAVEDAD (IMPACTO)				
			MUY BAJO 1	BAJO 2	MEDIO 3	ALTO 4	MUY ALTO 5
APARICIÓN (probabilidad)	MUY ALTA 5	5	10	15	20	25	
	ALTA 4	4	8	12	16	20	
	MEDIA 3	3	6	9	12	15	
	BAJA 2	2	4	6	8	12	
	MUY BAJA 1	1	2	3	4	5	



	Riesgo muy grave. Requiere medidas preventivas urgentes. No se debe iniciar el proyecto sin la aplicación de medidas preventivas urgentes y sin acotar sólidamente el riesgo.
	Riesgo importante. Medidas preventivas obligatorias. Se deben controlar fuertemente las variables de riesgo durante el proyecto.
	Riesgo apreciable. Estudiar económicamente si es posible introducir medidas preventivas para reducir el nivel de riesgo. Si no fuera posible, mantener las variables controladas.
	Riesgo marginal. Se vigilará aunque no requiere medidas preventivas de partida.

MATRIZ DE RIESGOS

RIESGO

Aparición (probabilidad)	Gravedad (Impacto)	Valor del Riesgo	Nivel de Riesgo
-----------------------------	-----------------------	---------------------	--------------------

Inconvenientes con los sensores ya instalados	4	2	8	Apreciable
Requerir un componente electrónico de difícil obtención o de costos elevados	3	3	9	Importante
Demoras en proveedores	3	3	9	Importante
Condiciones climáticas desfavorables	1	2	2	Marginal
Inconvenientes administrativos	3	3	9	Importante
No cumplir con las fechas propuestas	3	2	8	Apreciable
Modificaciones en los requisitos	3	4	12	Importante



Riesgos	Mitigación
Inconvenientes con los sensores ya instalados	Se procede a reemplazar los mismos con sensores nuevos
Requerir un componente electrónico de difícil obtención o de costos elevados	Buscar un componente similar que cumpla la misma función que el faltante
Demoras en proveedores	Buscar un proveedor alternativo dentro de la provincia para ajustar los tiempos de envío de los posibles componentes necesarios.
Condiciones climáticas desfavorables	Se debe tener en cuenta el cambio de clima a la hora de instalar y probar los distintos componentes, como así también asegurar contra contingencias climáticas los circuitos de los mismos.
Inconvenientes administrativos	Mantener al tanto sobre los beneficios de realizar este proyecto, para evitar oposición al desarrollo del mismo.
No cumplir con las fechas propuestas	Para mitigar este efecto, las tareas deberán ser planeadas con un cierto margen de resguardo, con el cual se podrá cubrir cualquier eventualidad espontánea (o parte de ella) para evitar que el tiempo total del proyecto no se exceda del requerido
Modificaciones en los requisitos	Mantener una comunicación constante con los interesados para realizar los cambios que se soliciten y reportar los avances.

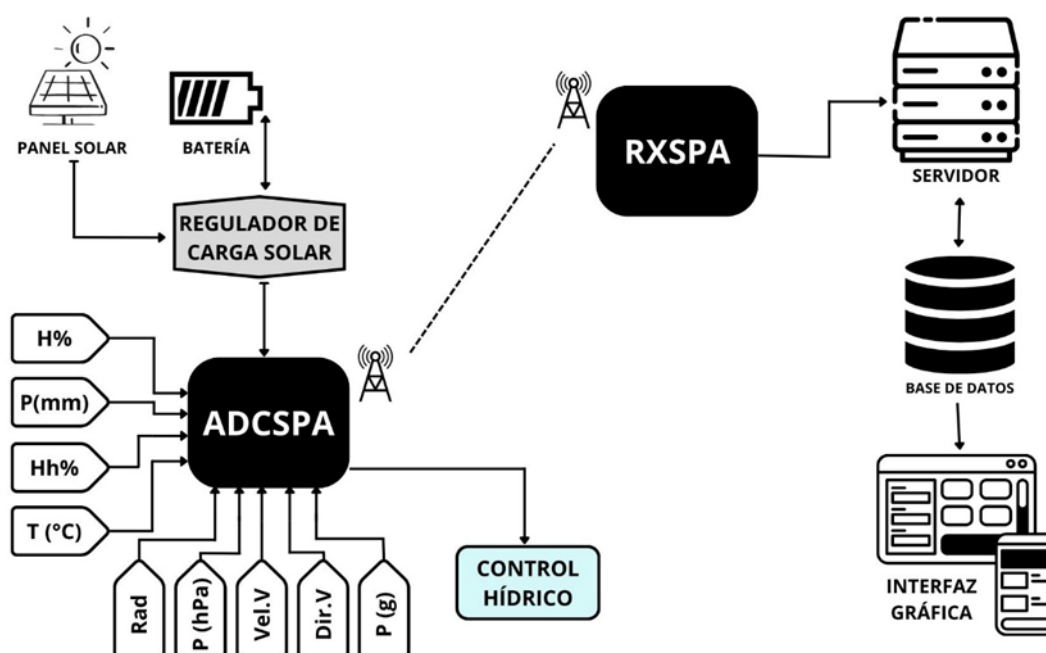


5 DESARROLLO DEL PROYECTO

5.1 DESARROLLO TÉCNICO

5.1.1 Arquitectura del sistema

5.1.1.1 Esquema general del Sistema SPA



5.1.1.2 Descripción general

La arquitectura del Sistema SPA (sistema suelo-planta-atmósfera) se concibe como un sistema integral que aborda el análisis de microclima, la medición de la evapotranspiración del cultivo (ETc), control hídrico y la visualización de esta información de manera remota. Esta sección proporcionará una descripción de la arquitectura, destacando sus componentes clave y su interconexión.

Energía Solar:

El sistema se sustenta mediante un sistema de energía solar, compuesto por un panel solar, regulador de carga y batería de 12V. Esta configuración garantiza la autonomía energética del sistema, vital para su operación en entornos agrícolas.

Estación Meteorológica:

La estación meteorológica recopila datos cruciales para el monitoreo del entorno. Equipada con sensores de radiación solar, temperatura ambiente, humedad relativa, precipitación, presión atmosférica, humedad de hoja, dirección y velocidad del viento, la estación proporciona información detallada sobre las condiciones climáticas.

Lisímetro de Pesada:

El lisímetro de pesada, equipado con una celda de carga y una válvula para el control hídrico, mide la evapotranspiración del cultivo (ETc) y proporciona información vital para un riego preciso. Esta información se integra con la estación meteorológica para evaluar la relación entre las condiciones climáticas y las necesidades hídricas del cultivo.



Módulo ADCSPA:

Este componente central realiza la adquisición de datos de los diversos sensores disponibles (estación meteorológica, lisímetro de pesada), el tratamiento de las señales y su conversión analógico-digital, asegurando la precisión de la información recopilada. Además, gestiona el proceso de transmisión de datos hacia el Módulo RXSPA, y posee la capacidad de supervisar el control hídrico.

Módulo RXSPA:

Ubicado en las instalaciones del INTA y conectado a internet, desempeña un papel clave en la operatividad del sistema. Este módulo recibe y gestiona los datos transmitidos por el módulo ADCSPA. Tiene la capacidad de comunicarse con el servidor para almacenar la información en la base de datos, asegurando la disponibilidad y seguridad de los datos para su posterior visualización remota.

Base de Datos y Servidor:

La información recopilada se almacena en una base de datos alojada en un servicio de alojamiento, garantizando la accesibilidad y seguridad de los datos. El servidor facilita la comunicación entre el módulo RXSPA y la base de datos, permitiendo la visualización remota a través de una interfaz gráfica.

Interfaz Gráfica:

La interfaz gráfica, desarrollada en Grafana, ofrece una representación visual clara de los datos recopilados. Esto permite el acceso remoto para el usuario final y facilita la interpretación de las condiciones del cultivo y el entorno.

5.1.1.3 Funcionamiento básico

El sistema SPA, presentado de forma simplificada en el esquema, opera de manera integral para proporcionar información esencial relacionada con el entorno agrícola. El equipo de energía solar, instalado en el campo, consta de un panel solar, una batería y un regulador de carga solar. Este conjunto alimenta tanto al ADCSPA como a los diversos sensores, incluyendo: humedad, pluviómetro, humedad de hoja, temperatura, radiación solar, velocidad del viento, dirección del viento y un lisímetro de pesada.

El ADCSPA se comunica con el RXSPA mediante un enlace de radio utilizando tecnología LoRa. Este enlace tiene el propósito de transmitir datos de los sensores al RXSPA cuando este último lo solicita. El RXSPA, ubicado en el edificio de geomática de la Estación Experimental Agropecuaria del INTA en Mendoza, está conectado a la red wifi del edificio. A su vez, el RXSPA establece una comunicación con el servidor, que tiene acceso a la base de datos del sistema SPA.

El RXSPA se comunica con el servidor, donde este último tiene la capacidad de postear la información recopilada y realizar consultas necesarias para el riego.

La interfaz gráfica representa la cara amigable del sistema SPA, permitiendo a los usuarios visualizar de manera intuitiva y accesible todos los datos recopilados. Desde esta interfaz, los usuarios pueden monitorear el sistema en su conjunto (datos y funcionamiento), tomando decisiones informadas para optimizar las prácticas agrícolas.



5.1.2 Equipo de energía solar

5.1.2.1 Introducción

En el contexto de proyectos agrícolas, especialmente aquellos situados en áreas remotas desprovistas de infraestructuras energéticas convencionales, la adopción de fuentes de energía renovable se convierte en una necesidad. El Sistema SPA (Sistema Suelo-Planta-Atmósfera), posicionado en ubicaciones rurales y distantes, enfrenta el desafío de garantizar una operatividad continua y eficiente. En este escenario, la energía solar emerge como la solución idónea para suplir la demanda energética, ofreciendo no solo sostenibilidad ambiental, sino también la autonomía necesaria para el funcionamiento ininterrumpido de los componentes electrónicos.

Este enfoque estratégico no solo asegura la operación constante del sistema en lugares donde la infraestructura eléctrica convencional no está disponible, sino que también posiciona al proyecto en la vereda de la eficiencia energética y la sostenibilidad. En las próximas secciones, se profundizará en los fundamentos teóricos de la generación de energía solar, destacando la relevancia de cada componente.

La elección de la energía solar fotovoltaica para alimentar el Sistema SPA se basa en criterios fundamentales que convergen hacia la eficiencia, sostenibilidad y autonomía operativa. Esta decisión estratégica se sustenta en varios factores claves:

1. **Disponibilidad en entornos remotos:** El Sistema SPA se encuentra ubicado en áreas rurales y remotas, donde la conexión a la red eléctrica convencional resulta impracticable. La energía solar ofrece una solución autosuficiente, eliminando la dependencia de infraestructuras energéticas externas.
2. **Sostenibilidad ambiental:** La generación de energía solar fotovoltaica es inherentemente limpia y renovable, contribuyendo a la reducción de emisiones de gases de efecto invernadero y minimizando la huella ambiental del proyecto. Esta elección se alinea con los principios de responsabilidad ambiental y sostenibilidad.
3. **Autonomía y fiabilidad:** La energía solar proporciona una fuente de alimentación continua y confiable, esencial para la operación constante del Sistema SPA. La capacidad de almacenamiento en baterías permite mitigar las variaciones en la radiación solar y garantiza un suministro energético durante períodos sin luz solar, como en horas nocturnas.
4. **Bajo mantenimiento:** Los sistemas fotovoltaicos requieren un mantenimiento mínimo en comparación con las fuentes de energía convencionales. La ausencia de partes móviles y la durabilidad de los componentes reducen los costos y las intervenciones periódicas, favoreciendo la eficiencia y la rentabilidad a largo plazo.

5.1.2.2 Componentes del equipo de energía solar del SSPA

Panel solar

Está constituido por células solares interconectadas. Cada célula solar es esencialmente una unidad capaz de convertir la radiación solar en electricidad. En términos más específicos, estas células emplean capas de semiconductor, mayormente silicio cristalino, que, al interactuar con la luz solar, generan corriente eléctrica.



La célula solar consta de dos capas semiconductoras, una dopada positivamente y otra dopada negativamente, formando una unión conocida como unión p-n. Cuando los fotones de la luz solar inciden sobre esta estructura, provocan la liberación de electrones, generando así una corriente eléctrica en el circuito conectado al panel solar.

El panel solar, en su conjunto, alberga varias de estas células dispuestas estratégicamente para aumentar la capacidad de generación de energía. La conexión de las células, ya sea en serie o en paralelo, se realiza para lograr la tensión y corriente deseadas. Además, el panel incorpora capas protectoras, como vidrio templado y materiales encapsulantes, para resguardar las células de condiciones ambientales adversas.

Dentro del marco del Sistema SPA, se ha adoptado el modelo de panel solar KS10T de la marca Solartec para satisfacer los requisitos energéticos en la ubicación remota.

El panel solar KS10T de la marca Solartec destaca por su calidad y eficiencia, respaldado por celdas fotovoltaicas de silicio policristalino de alta eficiencia producidas por Kyocera en Japón. Con una eficiencia de conversión superior al 14%, estas celdas aseguran un rendimiento óptimo en la generación de energía solar.

Para garantizar la durabilidad y resistencia, las celdas están encapsuladas con material plástico EVA (etil-vinil-acetato), que proporciona aislamiento eléctrico y estabilidad frente a la radiación ultravioleta. El vidrio templado de alta transparencia en la cara expuesta al sol, con 4 mm de espesor y bajo contenido de hierro, confiere resistencia al impacto y una mayor vida útil al panel.

La cara posterior, compuesta de TPE (elastómero termoplástico), aporta resistencia mecánica y eléctrica. El marco de aluminio anodizado no solo contribuye a la rigidez estructural, sino que también facilita la instalación. La caja de conexiones fijada a la cara posterior permite una interconexión eficiente con los demás componentes del sistema, integrándose perfectamente en el contexto del Sistema SPA.

Las características eléctricas principales del panel se presentan a continuación:

Potencia Nominal (PN)	10 Wp
Tensión a PN	17,4 V
Corriente a PN	0,58 A
Tensión de circuito abierto	21,7 V
Corriente de corto circuito	0,63 A

Regulador de Carga:

El regulador de carga desempeña diversas funciones para garantizar el rendimiento y la durabilidad del equipo de energía solar en el contexto del Sistema SPA. En primer lugar, su papel central radica en supervisar y controlar la carga de la batería, evitando situaciones de sobrecarga que podrían resultar perjudiciales para la integridad de la batería. Además, regula el flujo de corriente proveniente del panel solar, asegurando una carga óptima y eficiente.

Durante las horas diurnas, cuando la radiación solar es abundante, el regulador supervisa la generación de energía a partir del panel solar. Su función principal en este periodo es controlar la carga de la batería, asegurándose de que esta reciba una cantidad óptima de energía sin exceder los límites seguros.



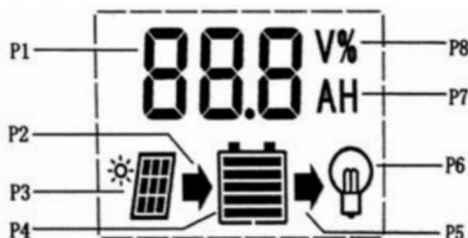
Durante este tiempo, el regulador también puede implementar estrategias para maximizar la eficiencia del sistema, como ajustar la tensión de carga según las condiciones ambientales. Asimismo, puede gestionar la temperatura de la batería para prevenir sobrecalentamientos y garantizar un rendimiento óptimo.

En las horas nocturnas, cuando no hay generación de energía solar, el regulador continúa desempeñando un papel importante. Supervisa la descarga de la batería, asegurándose de que se utilice de manera eficiente y evitando descargas profundas que podrían dañar la batería. Su capacidad para controlar este proceso contribuye a mantener la integridad de la batería y garantiza que el sistema esté listo para aprovechar nuevamente la energía solar en el próximo ciclo diurno.

Dentro del marco del Sistema SPA, se ha adoptado el siguiente regulador de carga (OPL1210), provisto por el Instituto Nacional de Tecnología Agropecuaria:



DESCRIPCIÓN DE LOS SÍMBOLOS GRÁFICOS DE LA PANTALLA:



P1: Parámetros digitales.

P2: Indicación de carga. Este símbolo indica que el panel solar está cargando la batería; sin este símbolo, significa que el panel solar no puede cargar la batería debido a un voltaje bajo. Si el símbolo parpadea, significa que la batería está completamente cargada y ha entrado en estado de carga flotante.

P3: Indicación de panel solar. Este símbolo indica que el controlador ha detectado la conexión del panel solar; sin este símbolo, significa que no se puede detectar la conexión del panel solar o que no hay luz solar en el panel solar.

P4: Indicación de energía de la batería con 5 barras.

P5: Indicación de descarga. Este símbolo indica que el controlador está en estado de salida; de lo contrario, no está en estado de salida. El parpadeo de este símbolo indica daños en los dispositivos de control internos.

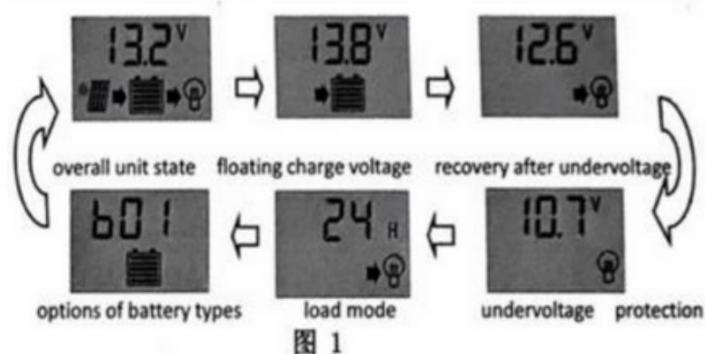
P6: Indicación de carga. Este símbolo indica que el controlador está en estado de salida; de lo contrario, no está en estado de salida. El parpadeo de este símbolo indica sobrecarga o daño en la carga.



P7: "A" representa la unidad de corriente; "H" representa la hora.

P8: "V" representa la unidad de voltaje.

DESCRIPCIÓN DE LA FUNCIÓN DE LOS BOTONES



MENU: Navegación cíclica mediante el botón de cambio de interfaz, permite explorar las distintas páginas en cada secuencia de cambio de ciclo, como se ilustra. En el estado de configuración de parámetros, este botón también opera como función de "suma".

SET: El botón SET, situado en la interfaz principal, controla el encendido o apagado de la carga. Durante el estado de configuración de parámetros, cumple la función de "resta".

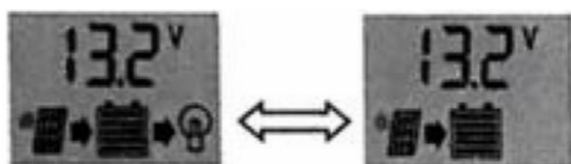
VISUALIZACIÓN Y CONFIGURACIÓN DE LOS PARÁMETROS

Al encenderse correctamente, el controlador se dirige automáticamente a la interfaz principal, que muestra la "tensión de la batería". Utilice el botón MENÚ para explorar sucesivamente las interfaces de parámetros. Si es posible realizar configuraciones en una interfaz específica, mantenga presionado el botón MENÚ (> 5 segundos, los números comenzarán a parpadear) para acceder a la interfaz de configuración de parámetros. Puede cancelar la interfaz de parámetros al mantener presionado nuevamente el botón MENÚ (los números dejarán de parpadear)

- Estado general de la unidad: Esta interfaz muestra el estado general de la unidad. Es la interfaz predeterminada después del encendido correcto, que muestra el estado de carga y descarga, una indicación de energía de la batería de 5 barras y el voltaje de la batería.



- Apertura y cierre de la carga: Puede usar el botón SET en la placa frontal para abrir o apagar la carga en la interfaz predeterminada. Nota: No existe tal función para este botón en otras interfaces.

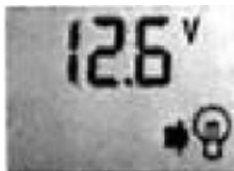




- Visualización y configuración de la tensión de flotación: Como se observa, se muestra el voltaje de carga flotante. Cuando la batería alcanza el voltaje flotante, el controlador mantendrá los valores de voltaje mediante el modo de carga PWM para evitar la sobrecarga. Mantenga presionado el botón MENÚ (> 5 segundos, los números comenzarán a parpadear) para finalizar la configuración de los valores de voltaje flotante y utilice los botones SET y MENÚ para ajustar el parámetro; cancele la interfaz de parámetros después de mantener presionado nuevamente el botón MENÚ (los números dejarán de parpadear). El valor de voltaje flotante será conservado por el controlador.



- Visualización y configuración de la recuperación después de la baja tensión: Como se observa, se muestra la tensión recuperada. Después de que el controlador realiza la función de protección contra la baja tensión, la salida de la carga se recuperará tan pronto como la tensión de la batería se recupere a un valor superior al de la baja tensión. Mantenga presionado el botón MENÚ (>5 segundos los números comenzarán a parpadear) para ingresar a la interfaz de configuración de recuperación después de la baja tensión; vuelva a presionar prolongadamente el botón MENÚ nuevamente (los números dejarán de parpadear) para cancelar la interfaz de parámetros después de completar la configuración. El valor configurado será conservado por el controlador.

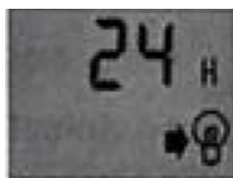


- Visualización y configuración de la protección contra la baja tensión: Como se muestra abajo, se muestra el valor de protección contra la baja tensión. El controlador cortará el circuito de carga cuando la tensión de la batería sea inferior a este valor, para evitar la sobre descarga de la batería. Mantenga presionado el botón MENÚ (> 5 segundos, los números comenzarán a parpadear) para ingresar a la interfaz de configuración de la protección contra la baja tensión y utilice los botones SET y MENÚ para ajustar el parámetro; mantenga presionado el botón MENÚ nuevamente (los números dejarán de parpadear) para cancelar la interfaz de parámetros después de completar la configuración. El valor de configuración será conservado por el controlador.





- Visualización y configuración del modo de funcionamiento de la carga: Como se observa en la figura, se muestra la interfaz del modo de carga, y los diferentes números representan diferentes modos de carga.
24h representa el Modo Normal; la carga siempre está encendida en circunstancias normales sin fallas.
1h-23h representa el Modo de Control de Luz con Control de Tiempo; en este modo, el controlador encenderá la carga después de la oscuridad y la apagará después de ciertas horas establecidas.
0h representa el Modo de Control de Luz; la carga estará encendida después de la oscuridad y se apagará después de un tiempo establecido.
Mantenga presionado el botón MENÚ (> 5 segundos, los números comenzarán a parpadear) en esta interfaz para activar la interfaz de configuración de modos de funcionamiento de la carga y utilice los botones SET y MENÚ para ajustar el parámetro; mantenga presionado el botón MENÚ nuevamente (los números dejarán de parpadear) para cancelar la interfaz de parámetros después de completar la configuración. El valor de configuración será conservado por el controlador.



- Visualización y configuración de los tipos de batería: Como se muestra, los números diferentes representan diferentes tipos de batería:
b00: Batería de litio (otros parámetros de control deben ajustarse para que coincidan con la batería).
b01: Batería sellada (tipo predeterminado para el controlador)
b02: Batería ventilada
b03: Batería de gel
Mantenga presionado el botón MENÚ (> 5 segundos, los números comenzarán a parpadear) en esta interfaz para activar el tipo de batería y utilice los botones SET y MENÚ para ajustar el parámetro; mantenga presionado el botón MENÚ nuevamente (los números dejarán de parpadear) para cancelar la interfaz de parámetros después de completar la configuración. El valor de configuración será conservado por el controlador.



FALLAS COMUNES Y MANEJO

- Protección y tratamiento contra la baja tensión: Si aparece y parpadea en la pantalla el símbolo de una batería descargada, significa que la tensión de la batería es inferior al voltaje de protección contra la baja tensión. El controlador ha entrado en el estado de protección contra la baja tensión y la salida se ha detenido. Solución: Utilice un panel solar o un cargador de batería para cargar la batería; cuando la tensión de la batería



alcance el valor de recuperación, la carga se encenderá nuevamente y entrará en estado de funcionamiento normal.

- Protección y tratamiento contra la sobrecarga: Si una bombilla aparece y parpadea en la pantalla, significa la ocurrencia de sobre corriente o cortocircuito. El controlador detendrá la salida y entrará en estado de protección por sobrecarga. Solución: Después de resolver el problema de cortocircuito en la salida y reducir la carga, presione el botón SET para restaurar la alimentación a la carga.
- Sobretensión de entrada y manejo: Si una batería llena aparece y parpadea en la pantalla, significa que la tensión de entrada de la batería en el controlador es más alta que la tensión de entrada nominal; el controlador detendrá la salida y entrará en estado de protección por sobretensión. Solución: 1. Elija una batería con el grado de voltaje apropiado para conectar con el controlador; 2. Retire cualquier otro cargador conectado a la batería.

Batería

La batería utilizada es la MG1270, diseñada para proporcionar un almacenamiento eficiente y confiable de la energía solar captada por los paneles solares. Este modelo específico pertenece a la categoría de baterías de plomo-ácido selladas (VRLA), conocidas por su rendimiento estable y su capacidad de funcionar en diversas condiciones ambientales.

La MG1270 tiene una capacidad nominal de 12 voltios y 7 amperios-hora, lo que la hace adecuada para aplicaciones que requieren un suministro de energía constante y confiable. Su diseño sellado garantiza la ausencia de mantenimiento, eliminando la necesidad de verificación de niveles de electrolito y ofreciendo una operación sin fugas ni emisiones.

Su construcción robusta y resistente a las vibraciones la hace idónea para aplicaciones en entornos agrícolas, donde puede estar sujeta a condiciones exigentes.

La MG1270, al integrarse con el regulador de carga del sistema, contribuye a una gestión eficiente de la energía almacenada, optimizando su rendimiento y prolongando su vida útil. Con estas características, la batería MG1270 se convierte en un componente fiable y duradero, fundamental para asegurar la continuidad del suministro energético en el Sistema SPA.



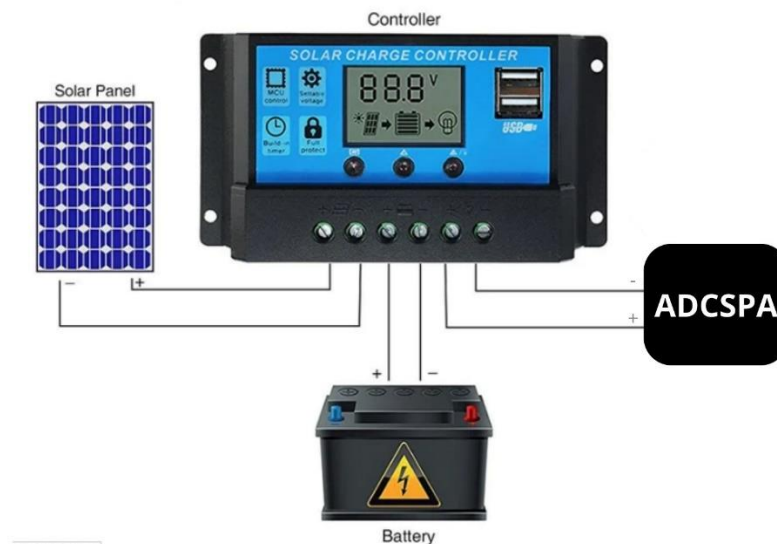


5.1.2.3 Configuración e instalación

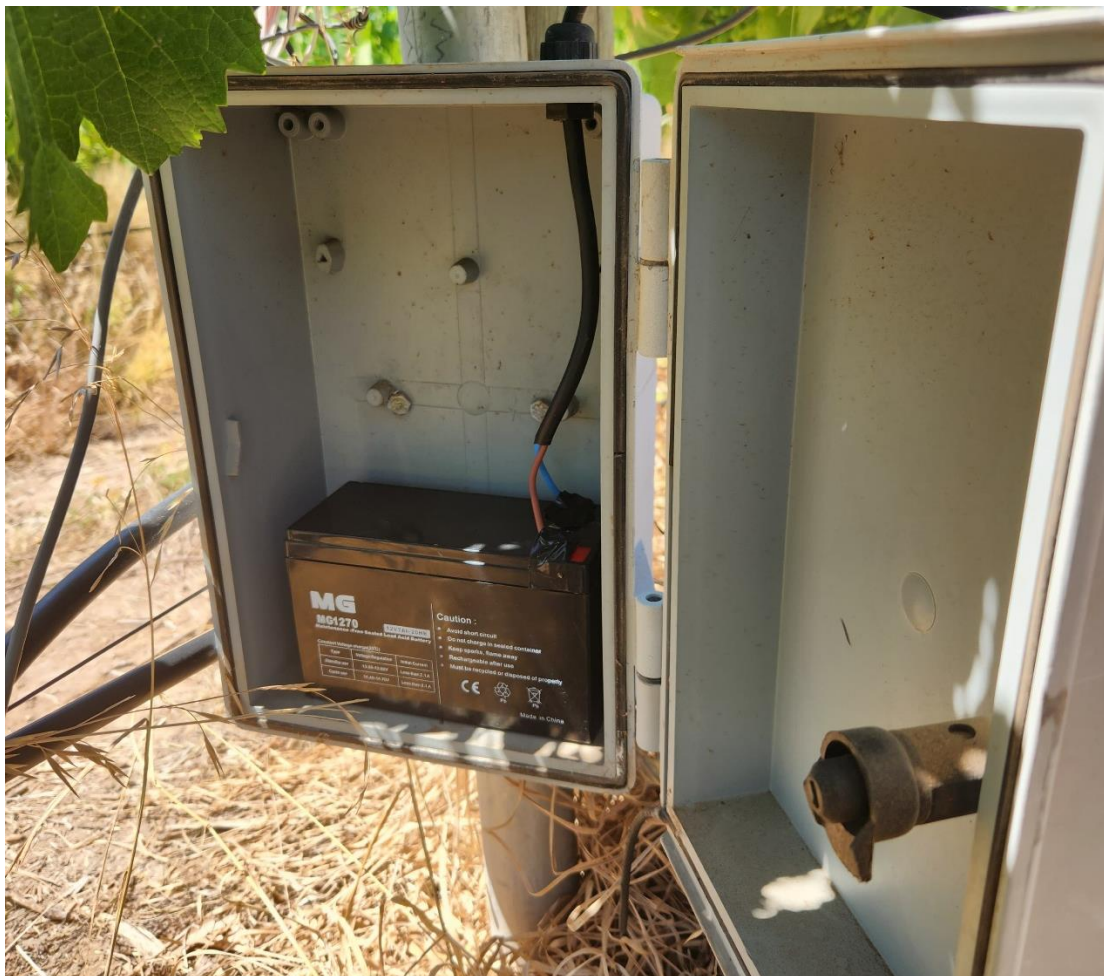
La configuración e instalación del sistema de energía solar requiere considerar las particularidades del medio ambiente y la optimización de la captación solar. A continuación, se detallan algunas consideraciones específicas:

- **Orientación hacia el norte:** Dada la ubicación en el hemisferio sur (Mendoza, Argentina), se recomienda orientar el panel solar hacia el norte para captar la máxima radiación solar disponible durante el día. Esto implica dirigir la superficie del panel hacia el norte geográfico para aprovechar al máximo la luz solar.
- **Inclinación ajustada a la latitud:** La inclinación del panel se ajusta a la latitud de la provincia para optimizar la captación solar a lo largo de las estaciones. En este caso, la inclinación sugerida sería alrededor de 32 grados con respecto a la horizontal. Este ángulo optimizado busca maximizar la exposición solar y garantizar una captura eficiente de la radiación solar a lo largo del año.
- **Altura de montaje estratégica:** La altura de montaje se determina considerando la vegetación circundante y otras posibles obstrucciones. Se busca una elevación que evite sombras no deseadas y asegure una exposición directa al sol.
- **Resistencia a condiciones ambientales:** La robustez del sistema se adapta a las condiciones ambientales locales, como vientos fuertes y variaciones climáticas. La resistencia a la corrosión y durabilidad estructural son esenciales para garantizar un rendimiento óptimo a lo largo del tiempo.

La conexión resulta de la siguiente manera:



A continuación, algunas imágenes del equipo de energía solar del Sistema SPA:





El panel solar tiene una eficiencia de conversión de más del 14%, asegurando una captura eficaz de la energía solar incidente. Con una corriente de carga de aproximadamente 0.59 A, el tiempo estimado para cargar completamente la batería MG1270 es de alrededor de 12 horas bajo condiciones ideales.

Considerando un consumo del sistema de alrededor de 200 mA, el tiempo de descarga de la batería se calcula dividiendo la capacidad de la batería por el consumo. Bajo estas condiciones, el tiempo de descarga es de aproximadamente 35 horas.

Estos cálculos proporcionan una estimación del rendimiento del sistema, sirviendo como referencia para la planificación y gestión de la energía en el contexto del Sistema SPA. Así se supone que el sistema funcionará ininterrumpidamente, pero es importante tener en cuenta que estos cálculos se basan en condiciones ideales y que se deberá corroborar a lo largo de las distintas estaciones del año si efectivamente son precisos.

5.1.3 Sensores y dispositivos de medición

Esta sección se adentra en la infraestructura que permite la recopilación de datos ambientales y agrícolas. Enfocándonos específicamente en la estación meteorológica y el lisímetro de pesada, componentes esenciales del sistema, exploraremos cómo estos dispositivos desempeñan un papel fundamental en la adquisición de información climática y en la medición precisa de la evapotranspiración del cultivo (ETc). A través de una selección de sensores, la estación meteorológica del Sistema SPA proporciona datos importantes, desde la radiación solar hasta la velocidad del viento. Asimismo, el lisímetro de pesada integra la medición de la evapotranspiración del cultivo y permite controlar la cantidad de agua utilizada en el riego, ofreciendo información valiosa para un manejo eficiente del recurso hídrico. En esta sección, exploraremos los componentes clave de ambos dispositivos y cómo contribuyen al monitoreo integral del entorno agrícola en el marco del Sistema SPA.

5.1.3.1 Estación meteorológica

La estación meteorológica del Sistema SPA representa la columna vertebral del monitoreo ambiental. Compuesta por una serie de sensores especializados, esta estación proporciona una evaluación detallada del microclima, permitiendo la toma de decisiones informadas en la gestión agrícola. Los componentes clave de la estación incluyen:



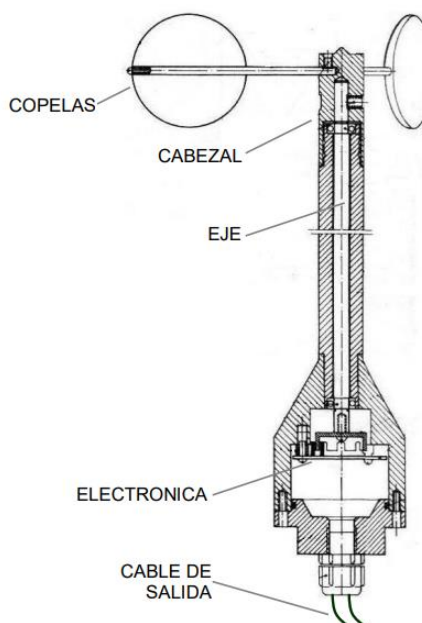
- Radiación solar: Mide la energía radiante proveniente del sol, proporcionando datos cruciales para entender la disponibilidad de luz solar.
- Temperatura: Evalúa la temperatura ambiente, esencial para comprender las condiciones térmicas del entorno agrícola.
- Humedad relativa: Mide la cantidad de vapor de agua en el aire, ofreciendo información sobre la humedad ambiental.
- Pluviómetro: Registra la cantidad de precipitación, crucial para evaluar el impacto de eventos climáticos en el cultivo.
- Presión atmosférica: Evalúa la presión atmosférica, proporcionando datos sobre las condiciones barométricas.
- Humedad de Hoja: Mide la humedad en la superficie de las hojas vegetales.
- Dirección del viento: Proporciona datos sobre la dirección del viento.
- Velocidad del viento: Proporciona datos sobre la velocidad del viento.

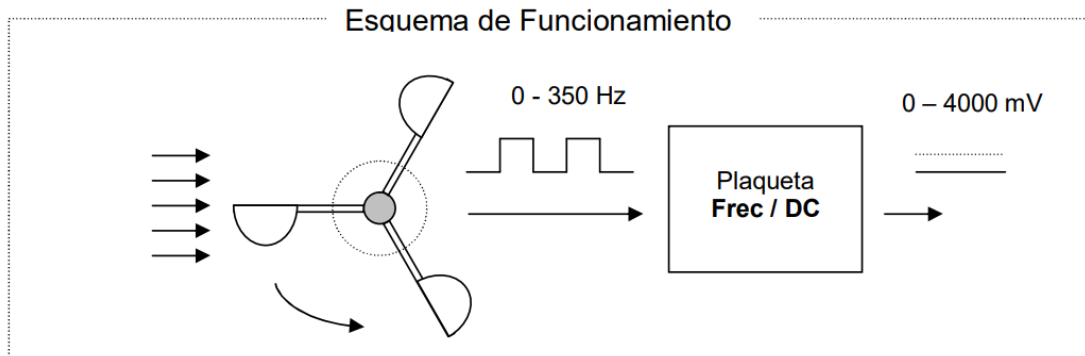
A continuación, se describen los principios de funcionamiento de cada uno de estos sensores, como así también los modelos utilizados e información importante:

5.1.3.1.1 Velocidad del viento

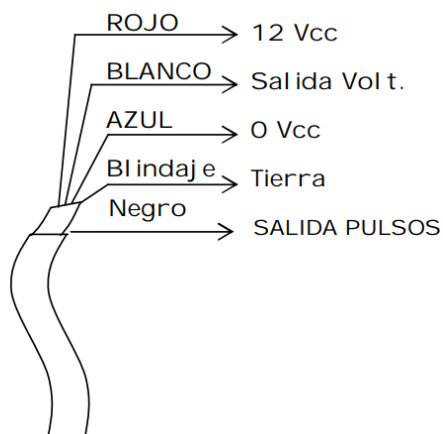
El sensor de velocidad del viento proporcionado por el Instituto Nacional de Tecnología Agropecuaria es el modelo TS231 de la marca TECMES. Su principio de medición radica en un cabezal rotatorio conteniendo 3 copelas, montado este sobre rodamientos sellados de acero inoxidable.

La fuerza que el viento ejerce sobre las copelas produce el giro del cabezal y del eje acoplado a este. En el extremo opuesto del eje un sistema de disco ranurado y acoplador óptico generan 6 pulsos eléctricos por revolución obteniéndose así una señal cuya frecuencia es proporcional a la velocidad de rotación del cabezal. Esta señal se pasa por un convertor de frecuencia a tensión DC y esta es la salida eléctrica que entrega el sensor. El cuerpo del sensor y el eje están fabricados en acero inoxidable de alta calidad.





La conexión eléctrica es la siguiente:



Nota: La salida que no se utilice (Volt. o Pulsos) deberá aislarse para evitar contacto con el chasis u otras partes metálicas.

Especificación Técnica:

Rango de Medición:	0 – 66.67 m/s \equiv 240 Km/h
Rango de Salida:	0 – 3833 mV
Transductor:	Optoacoplador de Ranura 6 Pulsos por Revolución
Electrónica	Rango de Frecuencia de Entrada: 0 – 350 Hz Rango de Tensión de Salida: 0 - 4000 mV
Exactitud:	0,3 m/s \equiv 1,08 Km/h
Alimentación:	9 - 16 Vcc (típico 12Vcc)
Consumo:	10 mA (en 12Vcc)
Conductores:	4 x 0,22 mm ² con blindaje general
Dimensiones:	Altura: 237 mm Base: 60 mm diámetro Radio de giro: 92 mm
Temp. Operación:	-20 a +60 °C



Algunas imágenes del sensor en el laboratorio y luego ya instalado en el campo:

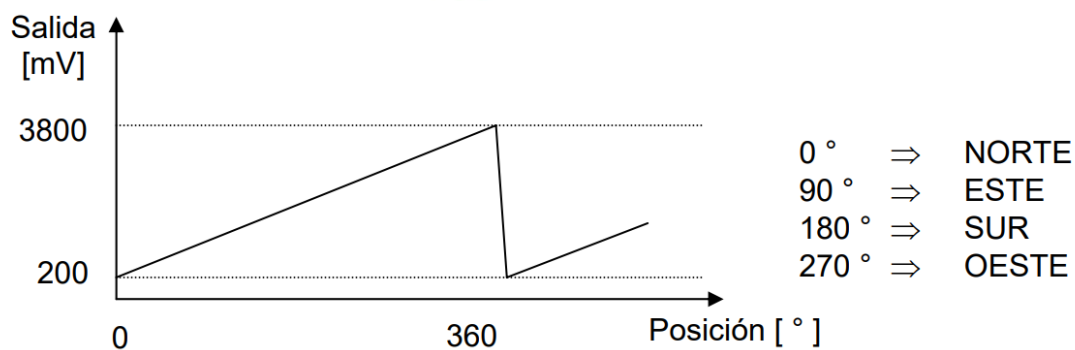
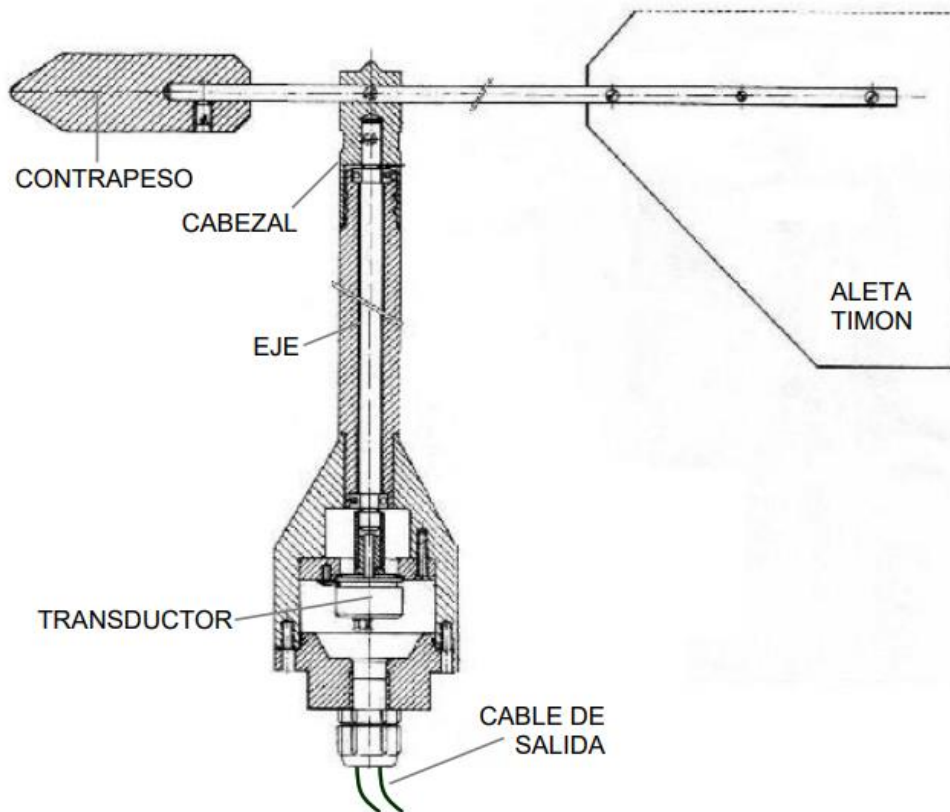




5.1.3.1.2 Dirección del viento

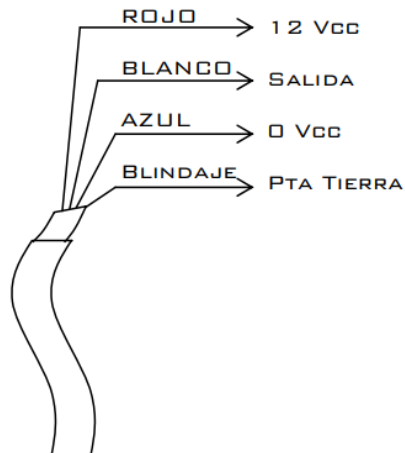
El sensor de dirección del viento proporcionado por el Instituto Nacional de Tecnología Agropecuaria es el modelo TS232 de la marca TECMES. Su principio de medición radica en un cabezal rotatorio conteniendo un eje radial sobre el cual se monta una aleta timón en un extremo y un contrapeso en el opuesto, soportado este conjunto sobre rodamientos sellados de acero inoxidable.

La fuerza que el viento ejerce sobre la aleta timón hace que el eje timón quede alineado con el viento siendo el contrapeso quien indica la dirección del mismo. En el extremo opuesto del eje del sistema rotatorio se acopla un sistema magnético a partir del cual y contando con cierta electrónica se genera una tensión proporcional a la posición angular del eje principal. Existe una posición angular en el cual la salida eléctrica pasa de su valor máximo al valor mínimo y se toma esta como posición de referencia la cual se hace coincidir con el Norte geográfico. El cuerpo del sensor y eje están fabricados en acero inoxidable lo cual lo hace inmune a los efectos del tiempo y evita la corrosión.





La conexión eléctrica es la siguiente:



Especificación técnica:

Rango de Medición:	0 - 360 °
Rango de Salida:	200 - 3800 mV
Sensibilidad de Salida:	10 mV / °
Exactitud:	± 3 °
Alimentación:	9 - 16 Vcc (12Vcc típico)
Consumo:	28 mA (en 12Vcc)
Conductores:	3 x 0,22 mm ² con blindaje general
Dimensiones:	Altura: 310 mm Cuerpo: 60 mm diámetro Radio de giro: 250 mm
Temp. Operación:	-20 a +60 °C
Peso:	1,650 kg.

A continuación, una imagen del sensor ya instalado en el campo:





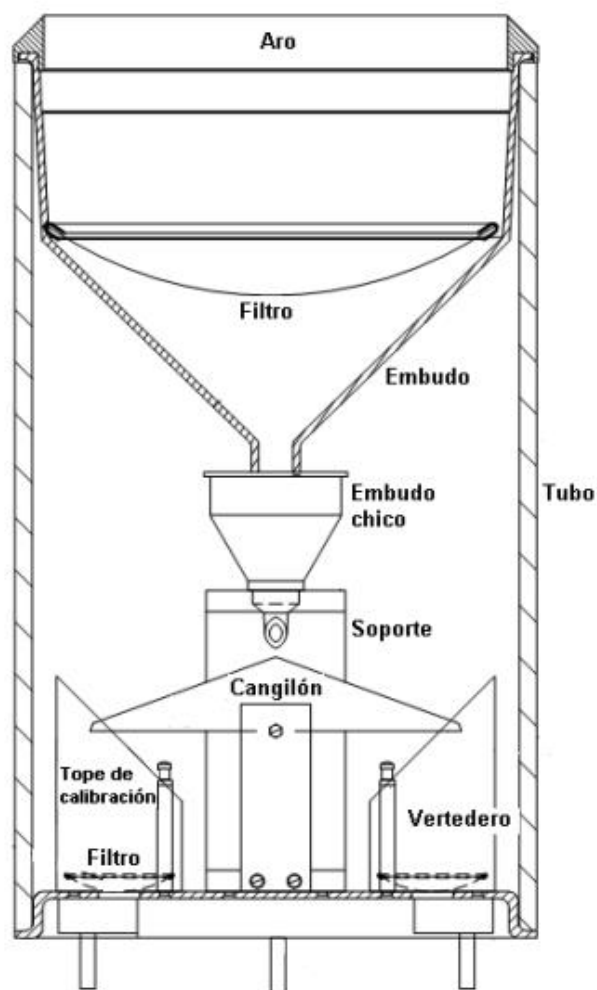
5.1.3.1.3 Pluviómetro

El sensor de precipitaciones proporcionado por el Instituto Nacional de Tecnología Agropecuaria es el modelo TS221 de la marca TECMES. El elemento fundamental que define su funcionamiento es el cangilón. A través del mismo se conmutan volúmenes de agua constante y cada conmutación es detectada por un sensor reed switch entregando un pulso eléctrico de salida.

El área de captación de precipitación queda determinada por el diámetro del embudo superior, de 200 mm. La relación entre el volumen de agua conmutado por cada cangilón y el área de captación del embudo se denomina paso del cangilón y representa la cantidad de lluvia que producirá un vuelco de cangilón y por ende un pulso de salida.

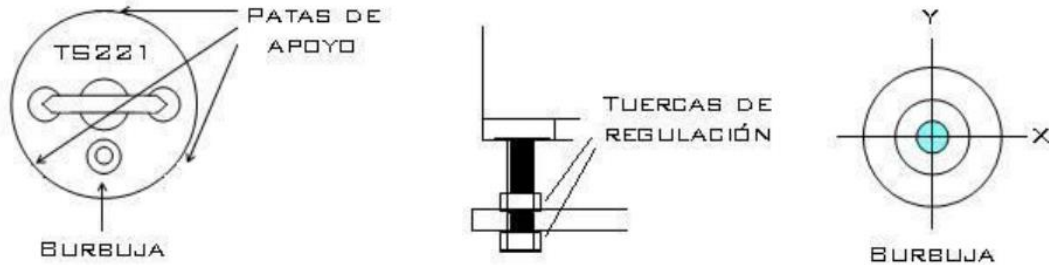
$$\text{Paso del Cangilón} = \text{Volumen Conmutado} / \text{Área de Captación}$$

Un valor habitual de paso es 0,25 mm lo cual significa que por cada 1 mm de precipitación se producirán 4 vuelcos de cangilón. El pluviómetro requiere de un contabilizador de pulsos a los fines de poder totalizar la lluvia caída durante un período de tiempo determinado, el mismo debe poseer la capacidad de reset o puesta a cero del contador.

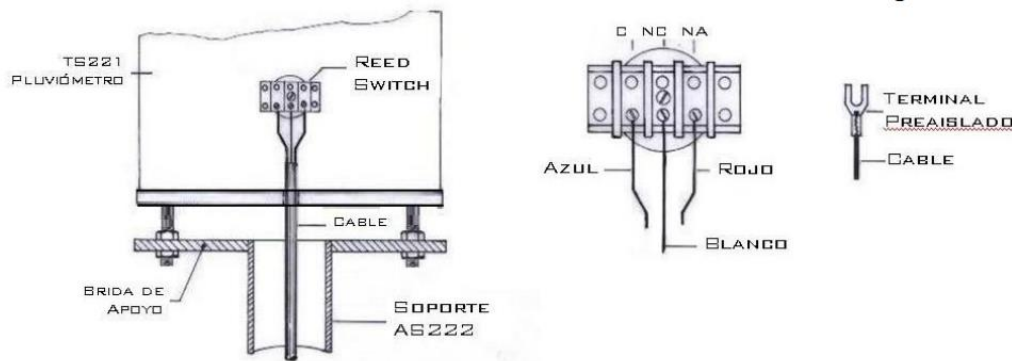




A los efectos de tener una buena medición el pluviómetro requiere estar nivelado horizontalmente, para lo cual cuenta en su base con una burbuja de nivel y el sistema de fijación mecánica al soporte permite la regulación de altura de cada uno de sus puntos de apoyos.



La conexión eléctrica es la siguiente:



Especificación técnica:

Paso del Cangilón:	0,1; 0,25; 0,5 mm. Otros pasos a pedido.
Superficie Colectora:	314,16 cm ²
Diámetro:	20 cm
Exactitud:	+/- 1% a 50mm/h ó a la intensidad que se especifique.
Contactos de Salida	COM, NA, NC (reed switch)
Conductores:	3 x 0,22 mm ² con blindaje general

Para realizar la calibración, colocar en un recipiente el volumen de agua correspondiente al ensayo:

Paso: 0.50mm	>>	1570.8 cm ³
Paso: 0.25mm	>>	785.4 cm ³
Paso: 0.10mm	>>	314.15 cm ³

Volcar el volumen a la intensidad requerida. Regular la intensidad midiendo el tiempo entre vuelcos. Al finalizar la descarga del volumen de agua colocado en el recipiente, se toma nota de los vuelcos. El valor ideal es 100 vuelcos. De ser necesario, ajustar la altura de los topes con el siguiente criterio:

- real > ideal >> enroscar el tope de calibración.
- real < ideal >> desenroscar el tope de calibración.



A continuación, algunas imágenes:



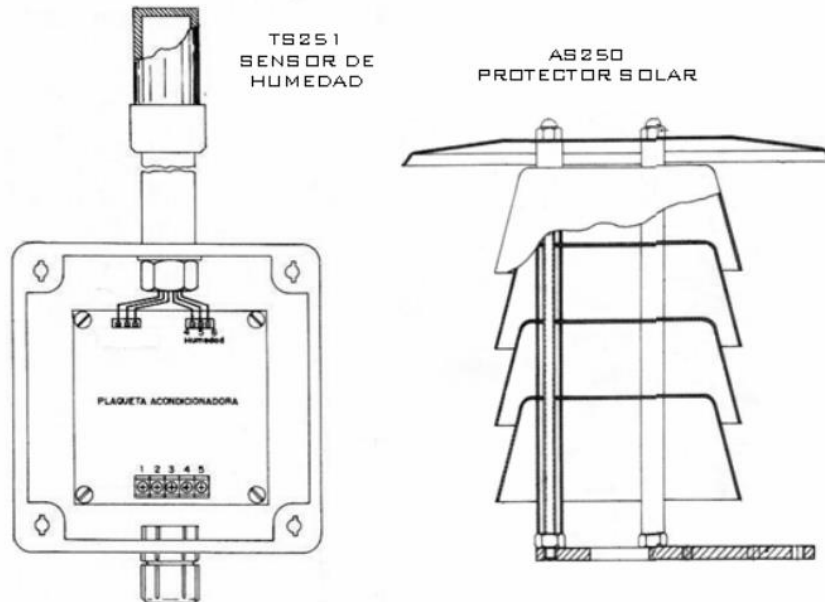
5.1.3.1.4 Humedad relativa

El sensor de humedad relativa proporcionado en primera instancia por el Instituto Nacional de Tecnología Agropecuaria es el modelo TS251 de la marca TECMES.

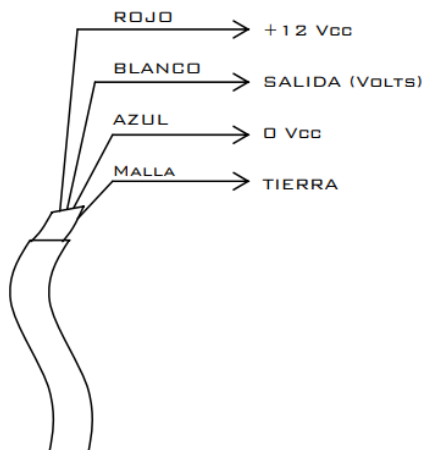
La medición se realiza a través de un circuito integrado en el cual un capacitor modifica su valor de capacidad en proporción a la existencia de humedad en aire. A esto se incorpora una electrónica de acondicionamiento de señal lo cual permite calibrar las mediciones con precisión en el rango deseado.



El elemento sensor se encuentra alojado dentro de un cabezal de aluminio ranurado y recubierto con un filtro, lo cual le da protección mecánica a los sensores, permite el contacto directo con el aire y evita el ingreso de polvo y partículas del aire. Adicionalmente se utiliza como accesorio un Protector Solar el cual impide el calentamiento por exposición directa al sol y tiene una construcción tal que favorece la auto ventilación del sistema.



La conexión eléctrica es la siguiente:



Especificación técnica:

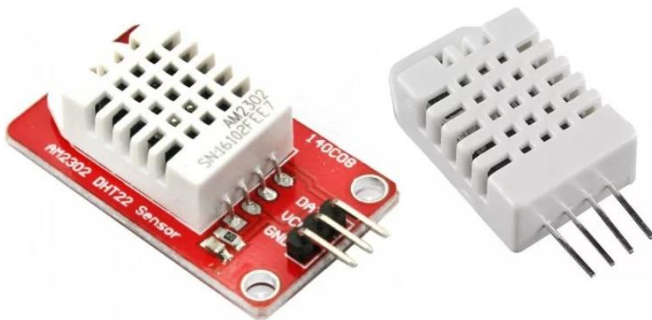
Rango de Medición:	0 a 100 %
Rango de Salida:	500 a 4000 mV
Sensibilidad de Salida:	35 mV / %
Exactitud:	± 3 %
Transductor:	de tipo Capacitivo
Alimentación:	9 a 16 Vcc (12Vcc típico)
Consumo:	5 mA



Al realizar el relevamiento de este sensor, se verificó que el funcionamiento del mismo no era correcto, se encontraba dañado. Luego de efectuar un análisis de alternativas, dentro de las cuales se encontraba la reparación del sensor, se tomó la decisión de reemplazarlo por un nuevo debido a que su reparación demandaba una alta cantidad de recursos.

El sensor de humedad elegido para su reemplazo es el DHT22 (Digital Humidity and Temperature) es un sensor de bajo costo que integra un sensor capacitivo para medir la humedad ambiente y un termistor para medir la temperatura. (En este caso solo utilizamos el dato de la humedad).

Habitualmente este sensor, está disponible en un encapsulado plástico blanco, con cuatro pines de conexión, aunque también se lo puede encontrar montado sobre una placa con sólo 3 pines.



Conexión eléctrica:

En el encapsulado de 4 pines, dos de ellos corresponden a la alimentación (5 Voltios y masa), un pin es para la lectura y escritura de comandos y datos y otro sin uso. El pin de datos necesita una resistencia de pull up de 10K para funcionar correctamente.



Pin	Función
1	Vcc
2	Datos
3	Sin conexión
4	GND

Las características principales del DHT22 son las siguientes:

- Alimentación de 3 a 5V
- Consumo máximo de 2.5mA durante la conversión
- Rango de humedad: 0-100% con un error entre 2 y 5%
- Rango de temperatura: -40 a 80°C con un error de $\pm 0.5^{\circ}\text{C}$
- Velocidad de lectura máxima: 0.5 Hz (una lectura cada 2 segundos)

Para la instalación de este sensor se utiliza solamente la protección del sensor original TS251 de TECMES (su circuito y sensor fue removido para una futura reparación en el INTA).

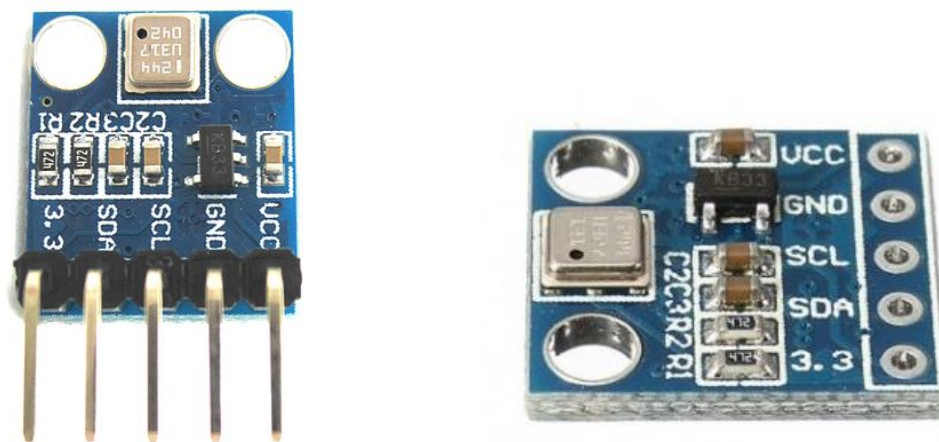
A continuación, una imagen del sensor DHT22 instalado en el campo con su protección:



5.1.3.1.5 Temperatura y presión atmosférica

En este contexto, se procedió a la adquisición de un sensor capaz de medir ambas métricas, dado que en el Instituto Nacional de Tecnología Agropecuaria (INTA) no se disponía de este tipo de dispositivos.

La elección recayó en el BMP180, un sensor altamente preciso de presión y temperatura desarrollado por Bosch Sensortec. Este dispositivo ofrece mediciones precisas de la presión atmosférica con una resolución de hasta 0.01 hPa, abarcando un rango de 300 a 1100 hPa, lo cual lo posiciona como una solución idónea para diversas altitudes. Asimismo, dispone de un termómetro de alta precisión que mide la temperatura ambiente con una resolución de 0.1°C, operando en un rango de -40°C a +85°C.



VCC: es el pin por donde se alimenta al módulo.

GND: pin de tierra. Se conecta directamente con GND.

SCL: señal de reloj del protocolo I2C.

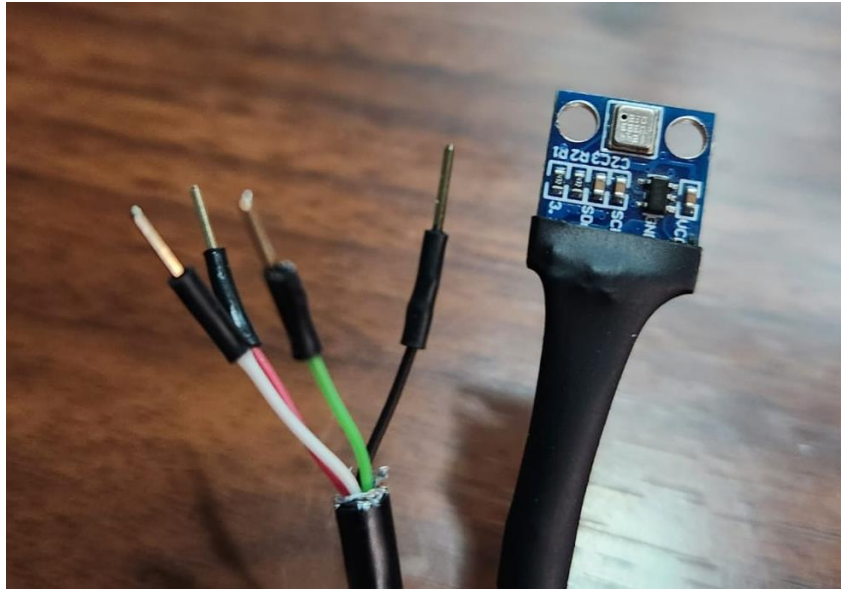
SDA: señal de datos del protocolo I2C.

Especificaciones técnicas:

- Rango de detección de presión barométrica: 300 hPa a 1100 hPa (-500 m a +9000 m en relación con el nivel del mar) con una precisión de hasta 0,02 hPa, en modo de resolución avanzada.
- Resolución: 0,03 hPa (0,25 m).
- Precisión del sensor de temperatura: +-3,6 Fahrenheit (+-2 Celsius).
- Voltaje de suministro: 1,8 V a 6 V; cuando utilice un voltaje de suministro de 3,3 V o menos, conecte el voltaje de suministro al pin "3.3" en lugar del pin "VCC".
- Interfaz I2C
- Bajo nivel de ruido, con 0,06 hPa (0,5 m) en modo de ultra bajo consumo y 0,02 hPa (0,17 m) en modo de resolución avanzada
- Rango de temperatura de funcionamiento: -40 a 185 Fahrenheit (-40 a 85 Celsius).
- Dimensiones (sin pines): 15 mm (0,6") de largo x 13,5 mm (0,05") de ancho x 2,6 mm (0,1") de alto.



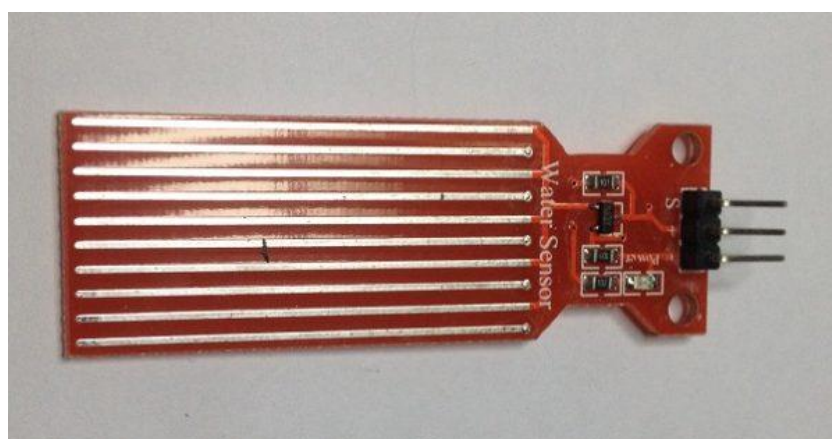
Para aprovechar la protección del sensor de humedad, este sensor se monta junto al mencionado, como se puede observar en las siguientes imágenes:





5.1.3.1.6 Humedad de hoja

Debido a que el sensor de humedad de hoja que se encontraba en el Instituto Nacional de Tecnología Agropecuaria era obsoleto, se avanzó en la compra de un sensor nuevo y más moderno, el mismo tiene como objetivo indicar si existe agua sobre las hojas de las plantas (vid u otras). El agua puede existir por lluvia o por rocío, a continuación, una imagen del sensor de humedad de hoja.





En situaciones donde la lluvia puede haber ocurrido en el microclima supervisado por la estación, factores como la magnitud de la lluvia, la temperatura y la hora del día pueden afectar la duración de la humedad en las hojas después de la precipitación. El dato de "hoja mojada", se convierte en un elemento clave para el análisis y modelado matemático de fito-enfermedades como la peronospora (hongos conocidos por producir esporas que se desarrollan en el tejido vegetal, generando síntomas como manchas, necrosis y decoloración en las hojas, causa comúnmente enfermedades foliares, y su impacto puede ser significativo en la agricultura, afectando el rendimiento y la calidad de los cultivos). Además, una vez que los investigadores han modelado la enfermedad, este dato específico se integra en el modelo, junto con otros, para prever la posible aparición de la enfermedad.

La filosofía subyacente del sensor es innovadora: en lugar de detectar directamente la presencia de agua en una hoja de planta (una tarea logísticamente compleja), el sensor opta por crear una hoja ficticia con material plástico u otro material, similar en tamaño a las hojas reales. Esta hoja ficticia actúa como un elemento captor, permitiendo que el rocío se forme aproximadamente al mismo tiempo en la hoja ficticia y en las hojas reales. Este enfoque ingenioso simplifica la medición de la humectación foliar, ofreciendo una solución práctica y eficaz para monitorear las condiciones.

La conexión eléctrica es la siguiente:

VCC: pin por donde se alimenta al módulo.

GND: pin de tierra.

S: señal de salida.

Especificaciones técnicas:

- Voltaje de funcionamiento: DC 3-5V
- Corriente de funcionamiento: < 20mA
- Tipo de sensor: analógico
- Área de detección: 40 mm x 16 mm
- Humedad: 10% -90% sin condensación

A continuación, una imagen del sensor de humedad de hoja instalado en el campo:





5.1.3.1.7 Radiación solar

El sensor de radiación solar que proporciono el Instituto Nacional de Tecnología Agropecuaria, es un fotodiodo con una resistencia, dentro de un pequeño cilindro plástico de color negro, con un difusor óptico en un extremo. A continuación, podemos ver una imagen del mismo:



El conjunto del fotodiodo y la resistencia está contenido dentro de un pequeño cilindro plástico de color negro. El color negro es conocido por su capacidad para absorber la radiación solar, minimizando las reflexiones y mejorando la precisión de la medición. En uno de los extremos del cilindro, hay un difusor óptico. Este componente tiene la función de dispersar la luz de manera uniforme sobre el fotodiodo, asegurando que la radiación solar incidente se distribuya de manera homogénea sobre la superficie fotosensible del dispositivo.

El sensor no requiere alimentación externa, ya que opera de manera pasiva. La salida del sensor es una señal de tensión que varía en el rango de 0 mV a 20 mV, proporcionando una medida proporcional a la intensidad de la radiación solar recibida. Cuanto mayor sea la radiación solar, mayor será la señal de voltaje generada por el sensor.

La problemática a abordar reside en la limitación de la tensión de salida del sensor, que se encuentra en un rango de 0 mV a un máximo de 20 mV. La necesidad es adecuar esta señal a un valor de tensión compatible con el convertidor analógico-digital (ADC), que opera en un rango de 0 a 3.3V. Para resolver esta situación, se implementará un amplificador operacional con el propósito de amplificar y ajustar la señal de salida del sensor, asegurando así una correspondencia adecuada con los requisitos del ADC.

El amplificador operacional seleccionado es el LM358P, a continuación, algunas de sus características:

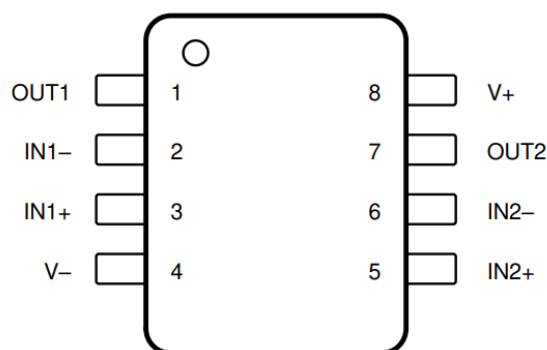


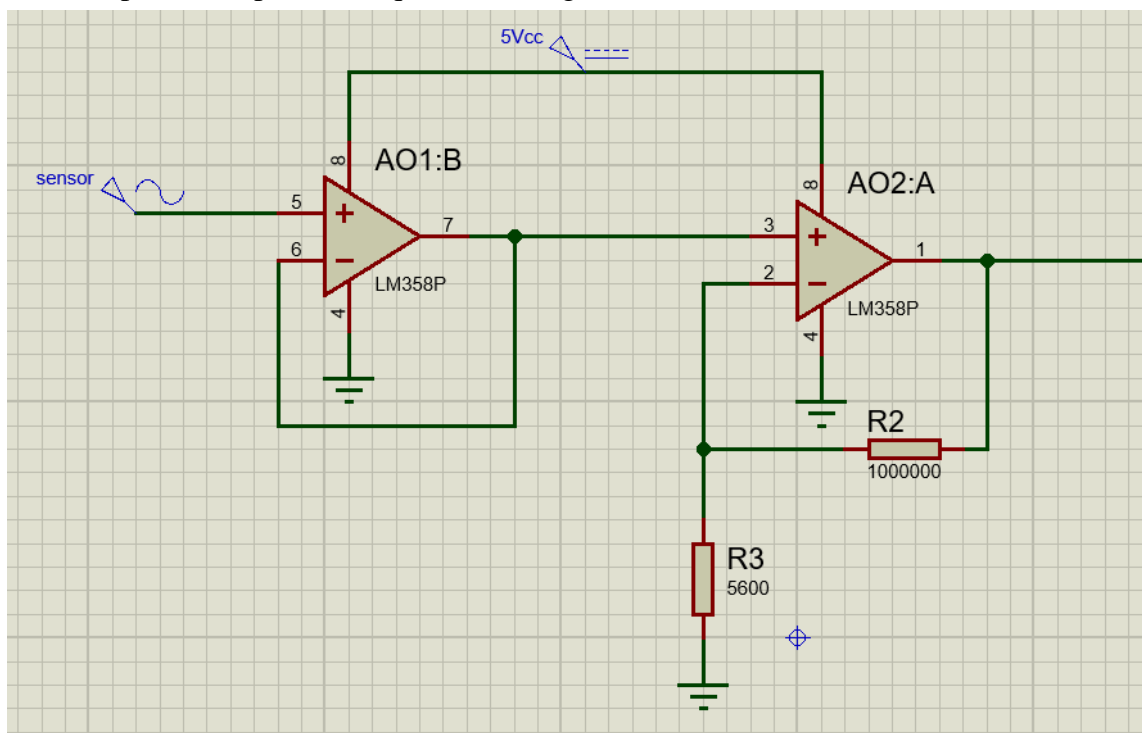


Table 5-1. Pin Functions

NAME	PIN		I/O	DESCRIPTION
	LCCC ⁽¹⁾	SOIC, SOT23-8, VSSOP, CDIP, PDIP, SO, TSSOP, CFP ⁽¹⁾		
IN1-	5	2	I	Negative input
IN1+	7	3	I	Positive input
IN2-	15	6	I	Negative input
IN2+	12	5	I	Positive input
OUT1	2	1	O	Output
OUT2	17	7	O	Output
V-	10	4	—	Negative (lowest) supply or ground (for single-supply operation)
NC	1, 3, 4, 6, 8, 9, 11, 13, 14, 16, 18, 19	—	—	No internal connection
V+	20	8	—	Positive (highest) supply

		MIN	MAX	UNIT	
V _S	Supply voltage, V _S = (V ₊) - (V ₋)	LM358B, LM358BA, LM2904B, LM2904BA	3	36	V
		LM158, LM258, LM358, LM158A, LM258A, LM358A, LM2904V	3	30	
		LM2904	3	26	
V _{CM}	Common-mode voltage	V-	V+ - 2	V	
T _A	Operating ambient temperature	LM358B, LM358BA	-40	85	°C
		LM2904B, LM2904BA, LM2904, LM2904V	-40	125	
		LM358, LM358A	0	70	
		LM258, LM258A	-20	85	
		LM158, LM158A	-55	125	

El circuito para el amplificador queda de la siguiente manera:



Con esta configuración, la señal que entra al ADC varía linealmente con la radiación solar entre 0 V y 3.3 V de tal manera de aprovechar al máximo el rango del convertidor. Luego de armar y probar el circuito la ganancia quedo igual a $AV \approx 165$.



Finalmente, podemos ver el sensor de radiación solar ya instalado en el campo:



5.1.3.1.8 Importancia de cada medición para el análisis del microclima

La Estación Meteorológica del Sistema SPA despliega una serie de sensores seleccionados para medir diversos parámetros atmosféricos. Cada una de estas mediciones desempeña un papel en el análisis del microclima. A continuación, se detalla la importancia de cada medición:

1. **Radiación Solar:** La medición de la radiación solar proporciona información sobre la cantidad de energía solar recibida en la ubicación específica. Este dato es esencial para comprender la disponibilidad de energía para el crecimiento de los cultivos.
2. **Temperatura Ambiente:** La temperatura ambiente influye directamente en el desarrollo de los cultivos y en la evaporación del agua del suelo. Un monitoreo preciso de la temperatura contribuye a evaluar el confort térmico de las plantas y a comprender las condiciones que pueden afectar su crecimiento.
3. **Humedad Relativa:** La humedad relativa del aire es un indicador clave para evaluar la cantidad de vapor de agua presente en la atmósfera. Esta medida es fundamental para entender la capacidad del aire para retener humedad y, por ende, su impacto en la evaporación y transpiración de las plantas.
4. **Precipitación:** El registro preciso de la precipitación permite cuantificar la cantidad de agua recibida por el suelo. Este dato sirve para evaluar el balance hídrico y determinar las necesidades de riego, contribuyendo así a un uso eficiente del agua.



5. **Presión Atmosférica:** La presión atmosférica proporciona información sobre la densidad del aire. Esta medida es relevante para comprender la estabilidad atmosférica y su influencia en los patrones climáticos locales.
6. **Humedad de Hoja:** Proporciona datos críticos sobre la presencia de agua en las hojas de las plantas. Su función principal contribuye al modelado y predicción de fito-enfermedades, especialmente relevante para comprender la variabilidad de la humectación foliar y tomar medidas preventivas en la gestión agrícola.
7. **Dirección y Velocidad del Viento:** Estos parámetros son esenciales para comprender la dinámica del viento, que puede afectar la dispersión de patógenos, la polinización y la distribución de la temperatura y la humedad en el cultivo.

5.1.3.2 Lisímetro de pesada

El lisímetro de pesada, dentro del Sistema SPA, es un instrumento importante para la medición precisa de la evapotranspiración del cultivo (ETc). Este dispositivo captura la variación de peso asociada al consumo de agua por las plantas y posibilita el riego autónomo al estar equipado con una válvula para dicho propósito. En esta sección, exploraremos los fundamentos teóricos de la medición, los principios de funcionamiento, así como la integración de componentes, relés y filtros. A través de este análisis, buscamos destacar la contribución del lisímetro de pesada a la comprensión de las necesidades hídricas del cultivo.

5.1.3.2.1 Desarrollo del lisímetro

Un lisímetro de pesada es un dispositivo utilizado en la investigación agrícola para medir con precisión la cantidad de agua que una planta o cultivo utiliza en un área específica. Su diseño consiste en un contenedor o cámara que contiene el suelo y la planta en estudio, montado sobre una celda/as de carga de alta precisión. La celda de carga registra cambios en el peso del lisímetro con el tiempo, lo que permite cuantificar la cantidad exacta de agua consumida.

El propósito principal del lisímetro de pesada en la investigación agrícola es evaluar el balance hídrico de un cultivo y comprender sus necesidades de agua. Al medir con precisión la entrada de agua (por riego o lluvia) y la salida de agua (por evaporación y transpiración), el lisímetro proporciona información valiosa sobre el uso eficiente del agua por parte de las plantas. Esto es esencial para optimizar las prácticas de riego, entender la respuesta hídrica de diferentes cultivos y variedades, y contribuir al desarrollo de estrategias de gestión del agua en la agricultura. Además, el lisímetro de pesada se utiliza para calibrar modelos de balance hídrico y mejorar la comprensión de los factores que afectan el rendimiento de los cultivos en relación con el suministro de agua.

El principio de funcionamiento detrás de un lisímetro de pesada se basa en la medición precisa de cambios en el peso del sistema suelo-planta. El lisímetro consta de una celda de carga de alta precisión sobre la cual se coloca un contenedor que contiene el suelo y la planta en estudio. Cuando el cultivo crece y utiliza agua a través de procesos como la transpiración y la absorción de raíces, se produce una pérdida de peso en el lisímetro. La celda de carga registra estos cambios y proporciona mediciones continuas y precisas del consumo de agua por parte del cultivo, lo que luego se utiliza para calcular la evapotranspiración.



El Lisímetro de pesada original que se tenía previsto utilizar para el proyecto era de grandes dimensiones, con un tamaño de 2 metros de largo, 2 metros de ancho y 2 metros de altura. Este equipo se encuentra instalado en el INTA y está compuesto por un tanque de fibra de vidrio y cuatro celdas de carga de 10.000 kg cada una. Lamentablemente, se presentó un inconveniente imprevisto. La estructura del equipo cedió y se hizo imposible la extracción de las celdas de carga, lo que imposibilitó su uso. Este problema representó un contratiempo en el desarrollo del proyecto, ya que se había planificado el uso de este equipo y su utilización para la medición de la ETC de los cultivos.

Para resolver esta situación, se trabajó en conjunto con el INTA y se obtuvo la autorización correspondiente de la cátedra de Proyecto Final para buscar una alternativa. Después de analizar diversas opciones, se decidió utilizar otro lisímetro de pesada de menor tamaño, pero que cumpliera con las mismas funciones y objetivos del equipo original. Este equipo fue adecuadamente acondicionado y calibrado para que midiera correctamente el peso del cultivo. Además, se incorporó la automatización del riego (que no estaba en el proyecto original) para facilitar el monitoreo del agua que se le agrega y así tener una mayor precisión en las mediciones de ETC.

Este lisímetro de pesada se compone de una estructura metálica robusta que alberga una celda de carga de 10 kg estratégicamente instalada para medir con precisión los cambios de peso del sistema. En la parte superior de esta estructura, se encuentra una plataforma de acrílico de 35 cm x 35 cm sobre la que se permite colocar el cultivo en estudio. En uno de los lados de la estructura, se ha montado una válvula, la cual se conecta a una manguera en la parte posterior para el suministro de agua y otra en la parte delantera para el riego. Esta válvula desempeña un papel importante al proporcionar un control preciso sobre el suministro de agua al cultivo.

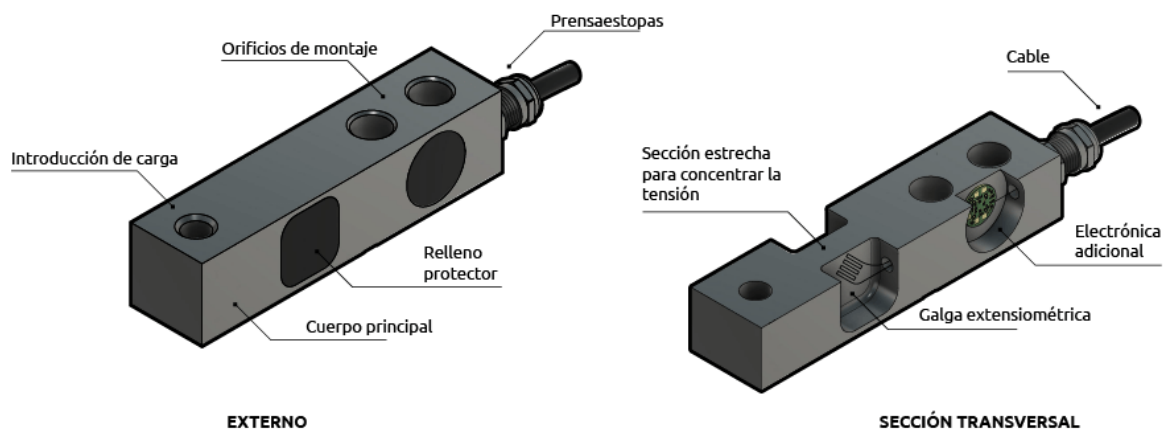
A continuación, se presenta una imagen del lisímetro en el laboratorio:



Celda de carga

Una celda de carga, también conocida como transductor de fuerza, es un dispositivo utilizado para medir la fuerza ejercida sobre ella. Está diseñada para convertir la fuerza aplicada en una señal eléctrica proporcional, que luego puede ser medida y registrada. Estas celdas son comúnmente utilizadas en aplicaciones industriales, científicas y de investigación para medir la fuerza o carga en diversas situaciones.

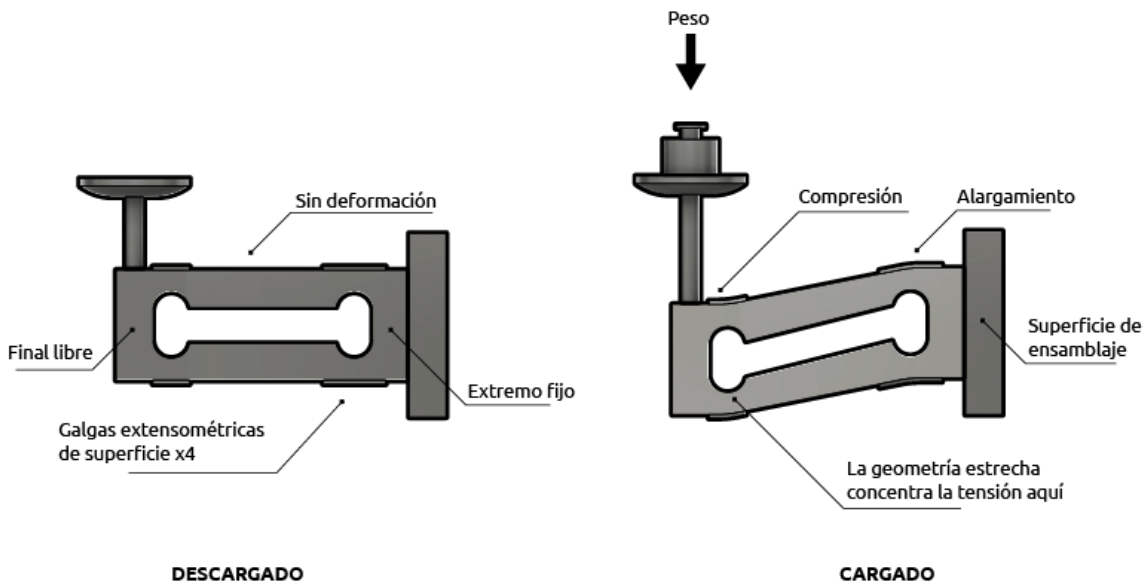
Las celdas de carga están compuestas por materiales elásticos, como aleaciones de acero o aluminio, que deforman su estructura cuando se les aplica una fuerza. La deformación resultante se convierte en una señal eléctrica, que puede ser medida y cuantificada para determinar la magnitud de la fuerza aplicada. Estas celdas son esenciales en muchas aplicaciones, como básculas industriales, sistemas de pesaje en laboratorios, maquinaria de producción, control de calidad, entre otros.



Cuando usamos celdas de carga, un extremo generalmente está asegurado a un marco o base, mientras que el otro extremo está libre para unir el peso o el elemento que soporta el peso. Cuando se aplica fuerza al cuerpo de la celda de carga, se flexiona ligeramente bajo la tensión. Esto es similar a lo que le sucede a una caña de pescar cuando un pescador engancha un pez. Cuando esta acción le sucede a un sensor de carga, la deformación es muy sutil y no es visible a simple vista.

Para medir la deformación, las galgas extensiométricas se unen firmemente al cuerpo de la celda de carga en puntos predeterminados, lo que hace que se deformen al unísono con el cuerpo. El movimiento resultante altera la resistencia eléctrica de las galgas extensiométricas en proporción a la cantidad de deformación causada por la carga aplicada.

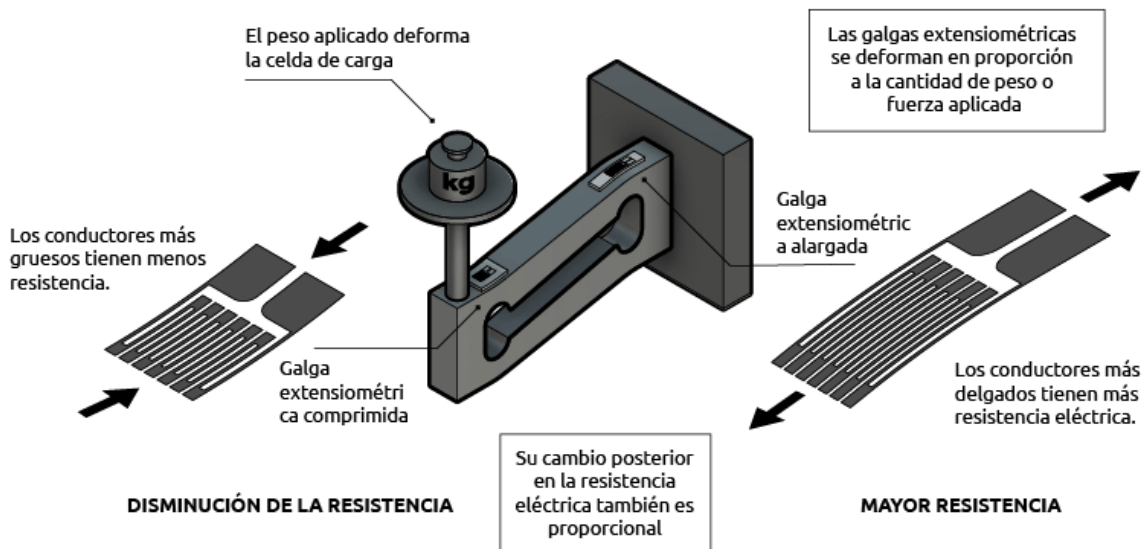
Mediante el uso de componentes electrónicos de acondicionamiento de señales, la resistencia eléctrica de las galgas extensiométricas se puede medir y la señal resultante se emite como una lectura de peso o fuerza.



Una celda de carga típica consta de dos partes: el cuerpo principal y un circuito eléctrico conectado. El cuerpo principal soporta el peso o la fuerza y representa la mayor parte de la célula de carga. Por lo general, se fabrica con acero o aluminio de alta calidad, lo que garantiza la fiabilidad mecánica y una distribución de tensiones predecible e uniforme.

El circuito eléctrico se encuentra dentro de la célula de carga, firmemente unido al cuerpo principal. El circuito incluye galgas extensiométricas que son componentes especiales del circuito, diseñados para detectar las deformaciones.

Las galgas extensiométricas consisten en un conductor eléctrico, en forma de alambre o fina lámina, dispuesto en un patrón de zig-zag. Este patrón las hace sensibles a la tracción y compresión a lo largo de su longitud, pero insensibles en todo su ancho. De tal manera pueden posicionarse con precisión para detectar fuerzas a lo largo de direcciones principales.





La celda de carga utilizada para el lisímetro es la siguiente:



Este sensor de carga tipo barra derecha (a veces llamado strain gauge) puede traducir hasta 10kg de presión (fuerza) en una señal eléctrica. Cada sensor de carga es capaz de medir la resistencia eléctrica que cambia en respuesta y proporcionalmente a la presión o fuerza (strain) aplicada a la barra. Estas celdas de carga tienen cuatro strain gauges que están interconectadas en una formación tipo wheatstone bridge. El código de color del cableado es el siguiente: rojo = E+, verde = O+, negro = E- y blanco = O-. Adicionalmente, estas barras tienen protección IP65 y poseen cuatro agujeros through hole M6 para ser montadas.

Las especificaciones se presentan en la siguiente tabla:

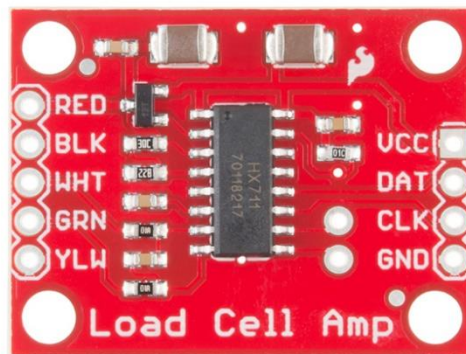
Specifications:		
capacity	kg	3,5,10,15,20,30,40,50,100,120
safe overload	%FS	150
ultimate overload	%FS	300
rated output	mV/V	2.0 ± 0.2
excitation voltage	Vdc	9~ 12
combined error	%FS	± 0.03
zero unbalance	%FS	± 2.0
non-linearity	%FS	± 0.02
hysteresis	%FS	± 0.02
repeatability	%FS	± 0.01
creep	%FS/30min	± 0.02
input resistance	Ω	405 ± 10
output resistance	Ω	350 ± 3
insulation resistance	M Ω	≥ 5000 @ 50 Vdc
operating temperature range	°C	-20 ~ +60
compensated temperature range	°C	-10 ~ +40
temperature coefficient of SPAN	%FS/10°C	± 0.02
temperature coefficient of ZERO	%FS/10°C	± 0.03
Electrical connection	cable	4 core shielded PVC cable, Ø4.5 × 450 mm



A medida que la fuerza actúa sobre la celda de carga, esta experimenta una deformación mecánica que se traduce en una variación en su resistencia eléctrica. Medir directamente esta variación puede ser complicado debido a las pequeñas señales generadas (en este caso la salida no supera los 25 mV), por lo que se utiliza un amplificador de instrumentación, como el HX711, para mejorar la precisión y la sensibilidad de la medición.

El HX711 es un convertidor analógico- digital de alta precisión con un amplificador de instrumentación integrado, diseñado específicamente para su uso con celdas de carga. Este dispositivo facilita la lectura precisa de señales analógicas generadas por las celdas de carga.

Su amplificador de instrumentación programable es esencial para mejorar la sensibilidad y la resolución al amplificar la señal analógica proveniente de la celda de carga, especialmente cuando se trata de señales débiles. Además, su capacidad para convertir señales analógicas en digitales facilita la integración fluida con microcontroladores y sistemas digitales. La inclusión de circuitos integrados para la compensación de offset y ganancia asegura una corrección precisa de las variaciones inherentes en las celdas de carga, como los cambios de temperatura y las características eléctricas, garantizando mediciones más exactas y consistentes. Asimismo, su diseño con baja resolución de ruido hace que el HX711 sea capaz de proporcionar mediciones estables y precisas incluso en entornos con interferencias eléctricas.



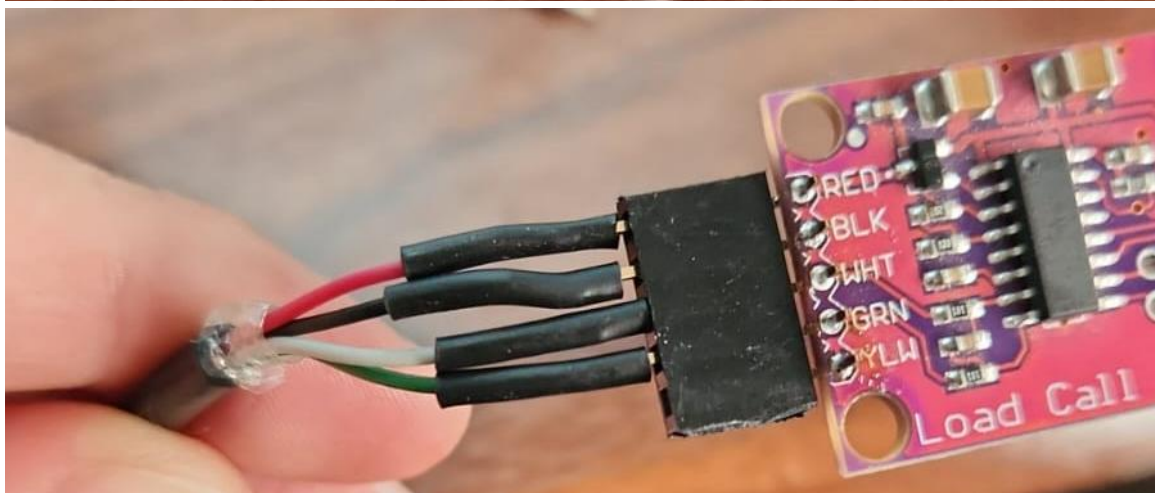
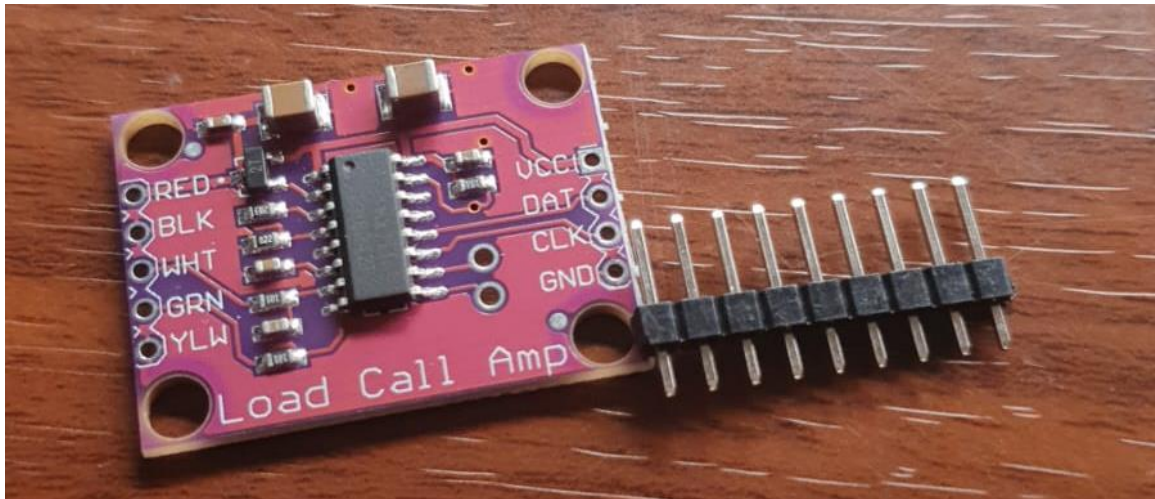
La distribución de los pines, los podemos ver en la siguiente imagen:

Regulator Power	VSUP	1	16	DVDD	Digital Power
Regulator Control Output	BASE	2	15	RATE	Output Data Rate Control Input
Analog Power	AVDD	3	14	XI	Crystal I/O and External Clock Input
Regulator Control Input	VFB	4	13	XO	Crystal I/O
Analog Ground	AGND	5	12	DOUT	Serial Data Output
Reference Bypass	VBG	6	11	PD_SCK	Power Down and Serial Clock Input
Ch. A Negative Input	INNA	7	10	INPB	Ch. B Positive Input
Ch. A Positive Input	INPA	8	9	INNB	Ch. B Negative Input

Para conectar la celda al módulo HX711 son necesarios 4 cables, los colores utilizados habitualmente son rojo, negro, blanco y verde. Cada color corresponde a una señal como se muestra a continuación:



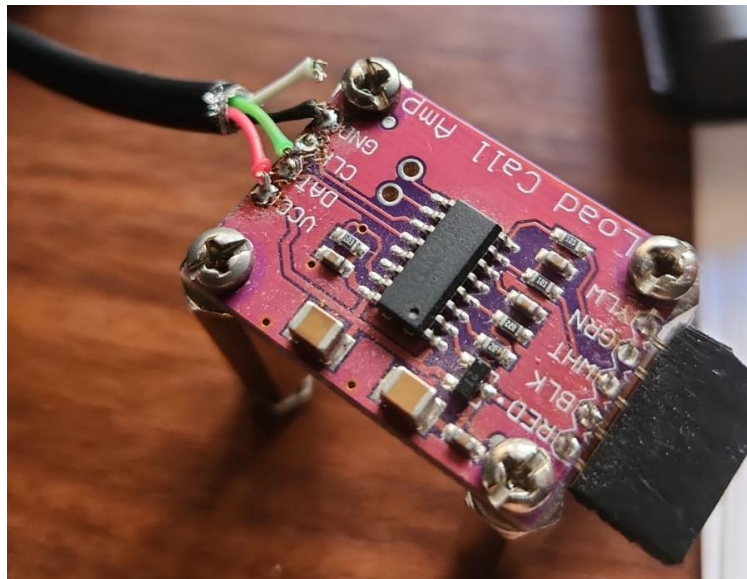
- Rojo: Voltaje de excitación +, E+, VCC
- Negro: Voltaje de excitación -, E-, GND
- Blanco: Amplificador +, Señal +, A+
- Verde: Amplificador -, Señal -, A-



Especificaciones técnicas:

- Voltaje de Operación: 5V DC
- Consumo de corriente: menor a 10mA
- Voltaje de entrada diferencial: $\pm 40\text{mV}$
- Resolución conversión A/D: 24 bit
- Frecuencia de lectura: 80 Hz
- Dimensiones: 33mm*20mm

Finalmente, algunas imágenes del HX711 durante el acondicionamiento y luego listo para conectarse al módulo ADCSPA:



Válvula para el riego

La implementación de una válvula para el riego en el lisímetro de pesada es un componente importante que añade un nivel de control preciso al sistema, permitiendo la regulación del suministro de agua de manera eficiente y adaptada a las necesidades específicas del cultivo.

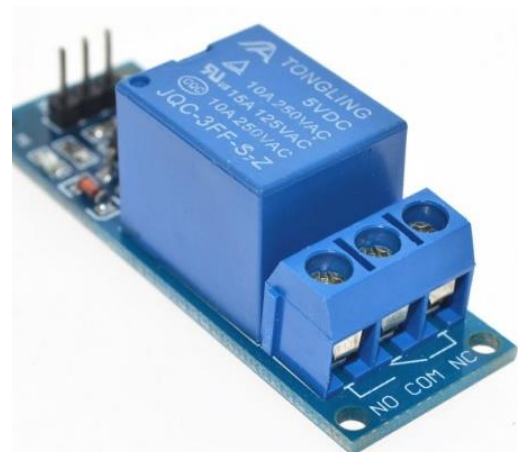
La válvula para el riego seleccionada es un dispositivo electromecánico que regula el flujo de agua en función de las señales recibidas. Su funcionamiento se basa en la apertura y cierre controlados, permitiendo una administración precisa del agua destinada al riego. La elección de una válvula específica para aplicaciones agrícolas asegura la durabilidad y resistencia requeridas para su operación continua en el entorno del campo.

El dispositivo seleccionado es una válvula de 12V 3/4 Normal Cerrada, a continuación, se presentan imágenes de la misma:



El sistema incorpora un relé que actúa como interruptor controlado electrónicamente para la válvula. Este relé es gestionado por el firmware del módulo ADCSPA, permitiendo la apertura y cierre automático de la válvula en respuesta a las necesidades detectadas por el lisímetro. La automatización aporta una capa adicional de eficiencia al proceso de riego, garantizando una respuesta rápida y precisa a las condiciones cambiantes del cultivo.

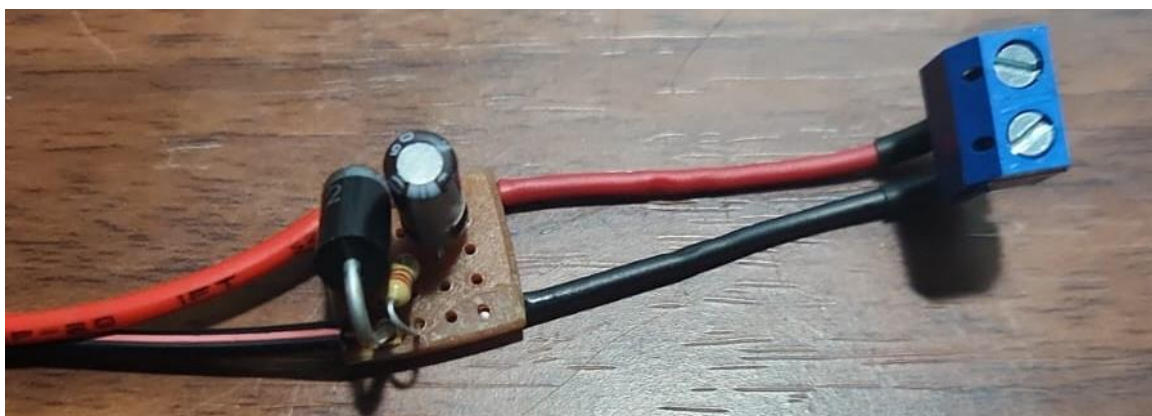
En este caso se optó por el Relé 12v 1ch Opto acoplado 220V 10A, que nos permite controlar la válvula sin inconvenientes y tratando de reducir las interferencias al máximo. A continuación, algunas imágenes del mismo:



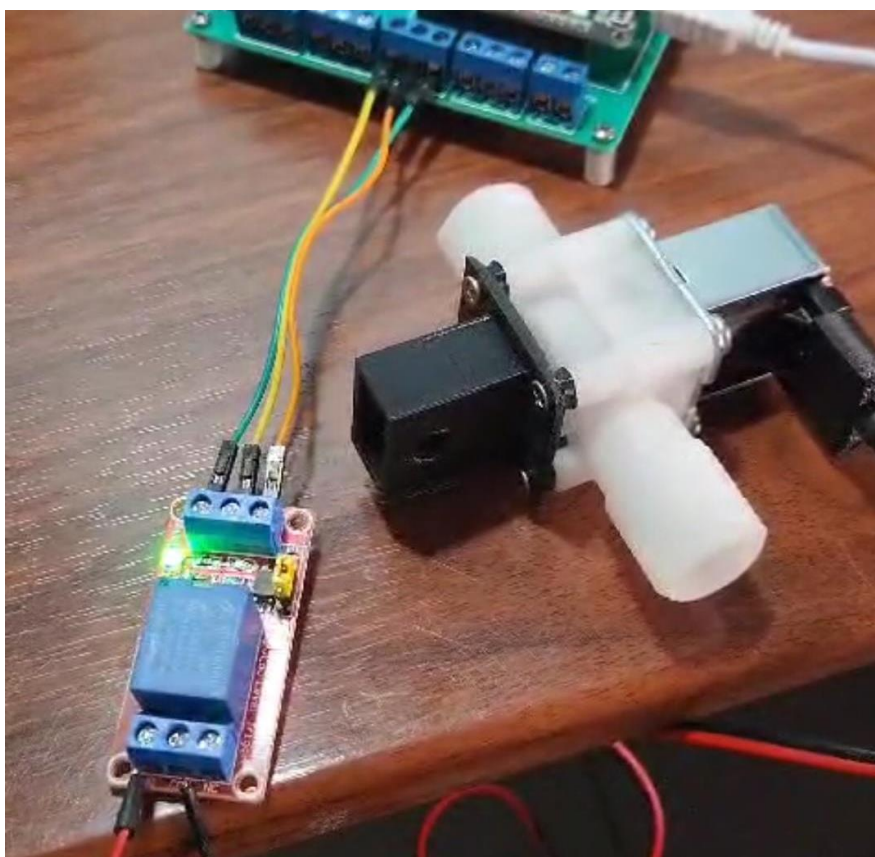
Un filtro compuesto por un diodo en paralelo con un RC serie, se ha incorporado estratégicamente y está conectado en paralelo a la válvula. Este filtro desempeña un papel importante al mitigar interferencias electromagnéticas que podrían afectar el rendimiento de componentes electrónicos sensibles. Al mantener la integridad de las señales eléctricas, asegura una operación confiable y estable del sistema. Los elementos utilizados son:

$R = 220 \text{ ohm}$, $C = 10 \text{ uF}$, Diodo = 1.5KE18CA

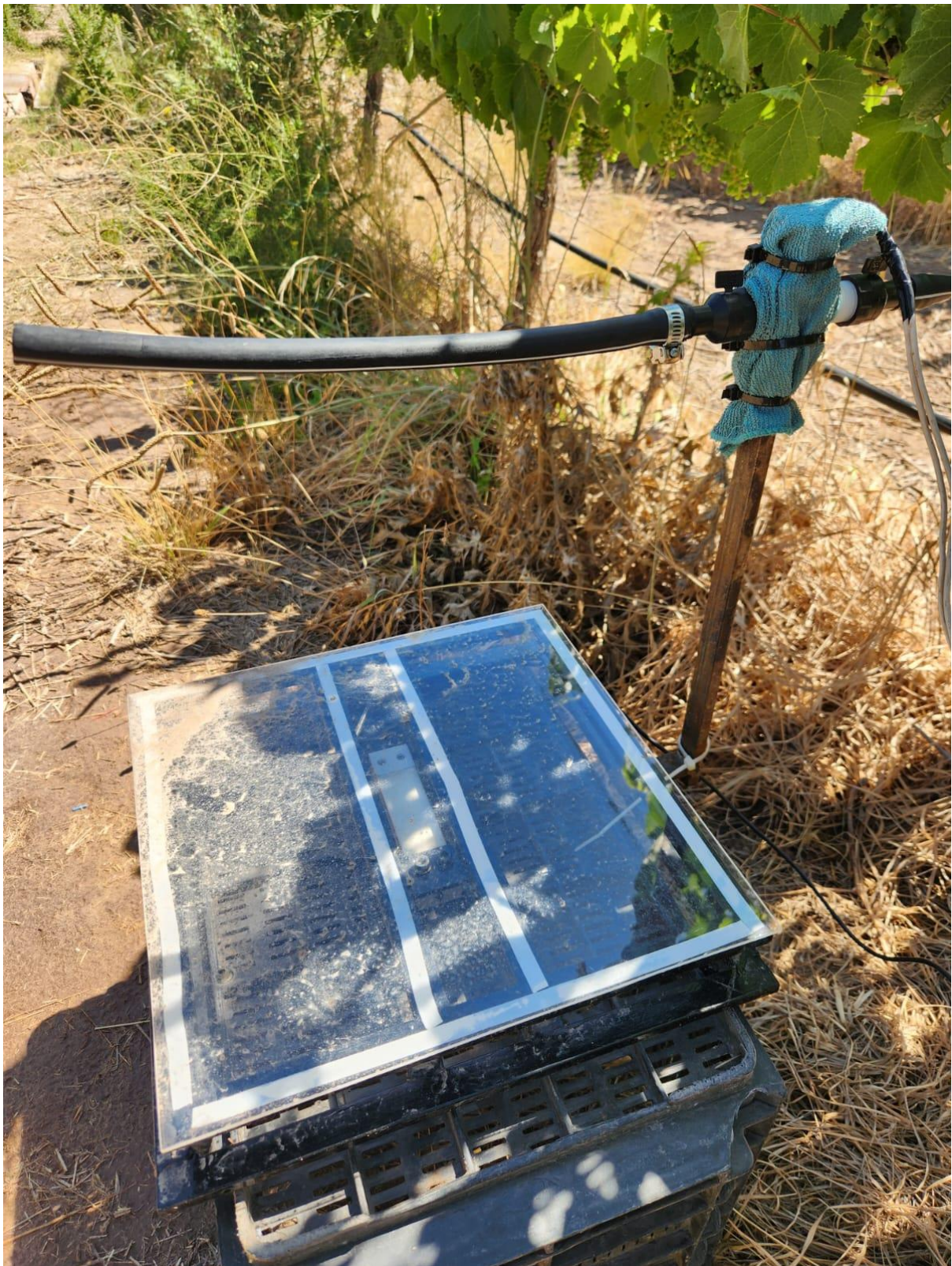
A continuación, una imagen del filtro armado:



Aquí una imagen testeando la válvula en conjunto con el relé:



Finalmente podemos observar en las siguientes imágenes, el lisímetro en el campo terminado:





5.1.4 Comunicación inalámbrica

5.1.4.1 Implementación de módulos LoRa RAK3172

La fase de implementación de los módulos LoRa representa un componente importante en el desarrollo exitoso del proyecto. Estos módulos son elementos fundamentales para establecer la comunicación inalámbrica, y el modelo RAK3172, en particular, ha sido seleccionado por sus características avanzadas y su capacidad para satisfacer los requisitos de nuestro proyecto.

En esta sección, detallaremos el proceso de implementación de los módulos RAK3172, destacando los pasos clave, las configuraciones específicas y los desafíos abordados durante esta etapa del proyecto. Su elección se basa en su capacidad para ofrecer un rendimiento robusto, eficiencia energética y la flexibilidad necesaria para adaptarse a las necesidades específicas del proyecto.

RAK Wireless es una empresa líder en el campo de soluciones de conectividad inalámbrica y dispositivos para el Internet de las Cosas (IoT). Con un enfoque centrado en facilitar la implementación de tecnologías inalámbricas en proyectos de IoT, RAK Wireless se ha destacado por ofrecer productos innovadores y confiables que satisfacen las crecientes demandas del mercado. La empresa se especializa en el desarrollo y la fabricación de una amplia variedad de dispositivos, entre los que se incluyen módulos LoRa, módulos Wi-Fi, gateways y sensores diseñados para abordar los desafíos específicos de la conectividad en el



ámbito del Internet de las Cosas. La gama de productos de RAK Wireless se distingue por su calidad, eficiencia energética y su capacidad para adaptarse a diversas aplicaciones.

5.1.4.1.1 Módulo RAK3172

RAK3172 es un módulo transceptor de largo alcance y bajo consumo basado en el chip STM32WLE5CC. Proporciona una solución fácil de usar, de tamaño pequeño y bajo consumo para aplicaciones inalámbricas de datos a larga distancia. Este módulo cumple con las especificaciones de Clase A, B y C de LoRaWAN 1.0.3. Puede conectarse fácilmente a diferentes plataformas de servidor LoRaWAN como TheThingsNetwork (TTN), Chirpstack, Actility, etc. También es compatible con el modo de comunicación LoRa punto a punto (P2P), lo que facilita la implementación de una propia red LoRa personalizada de largo alcance. Es posible configurar el modo y la operación del módulo utilizando comandos AT a través de una interfaz UART. RAK3172 también ofrece funciones de bajo consumo que son muy adecuadas para aplicaciones alimentadas por batería.



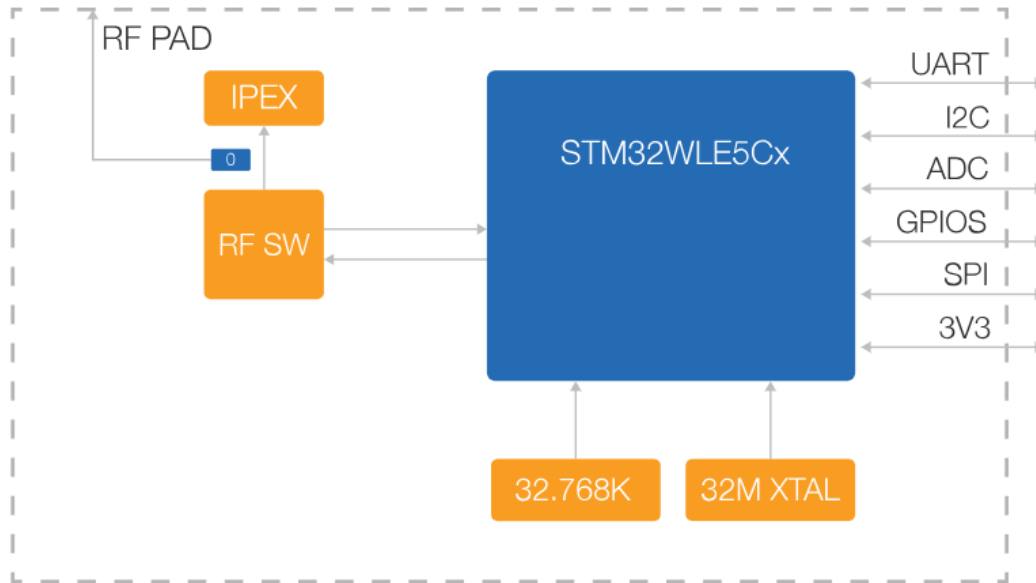
Presenta las siguientes características:

- Basado en STM32WLE5CCU6
- Cumple con la especificación LoRaWAN 1.0.3
- Bandas compatibles: EU433, CN470, IN865, EU868, AU915, US915, KR920, RU864 y AS923-1/2/3/4
- Activación LoRaWAN mediante OTAA/ABP
- Comunicación LoRa punto a punto (P2P)
- Firmware personalizado a través de la API RUI3
- Conjunto de comandos AT fácil de usar mediante interfaz UART
- Largo alcance, superior a 15 km con antena optimizada
- ARM Cortex-M4 de 32 bits
- 256 kbytes de memoria flash con ECC
- 64 kbytes de RAM



- Consumo ultra bajo de energía, 1.69 μ A en modo de suspensión
- Voltaje de suministro: 2.0 V ~ 3.6 V
- Rango de temperatura:
 - RAK3172: -20° C ~ 85° C
 - RAK3172-T: -40° C ~ 85° C

A continuación, podemos ver el diagrama en bloques del mismo:



Algunas características eléctricas:

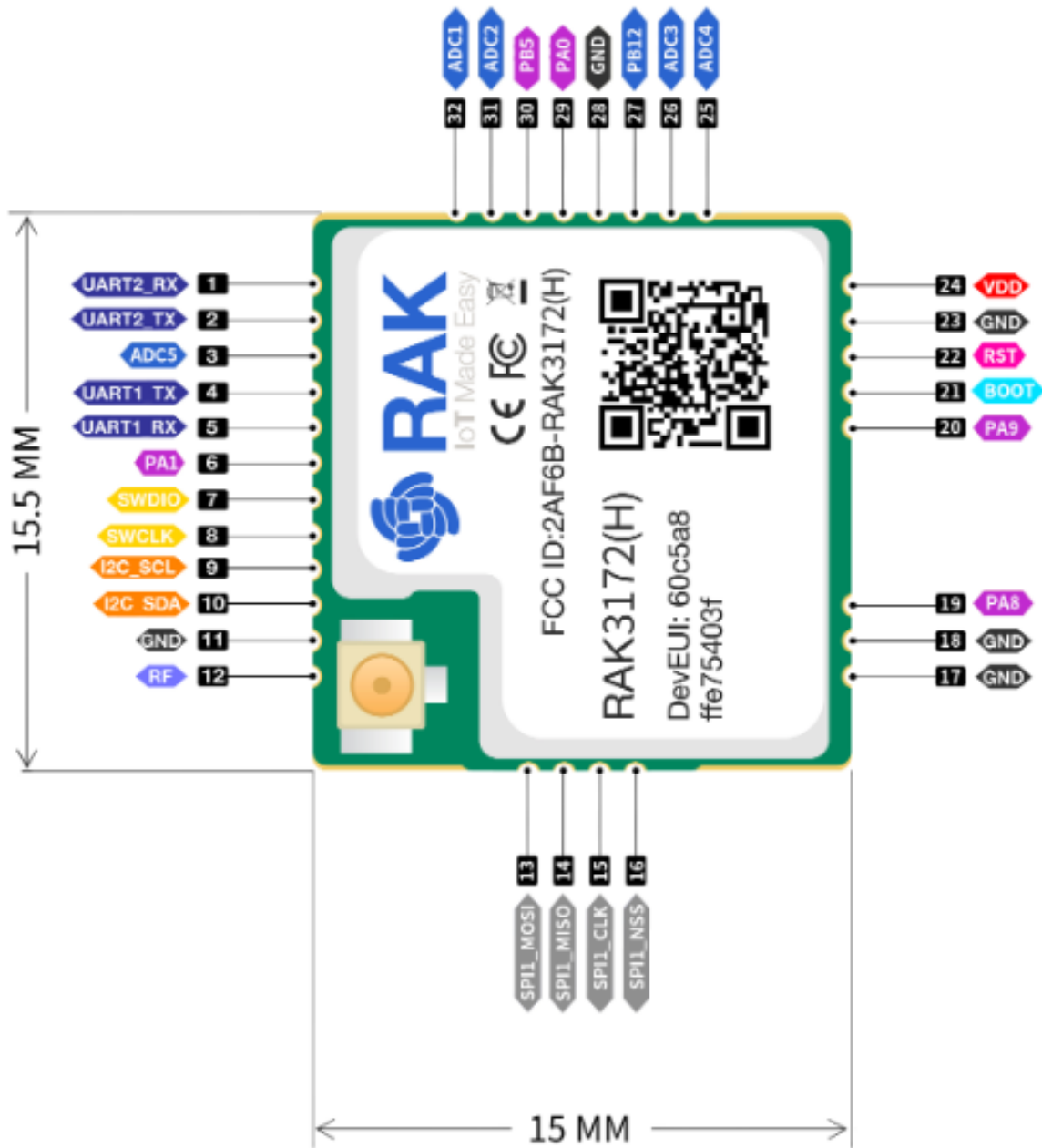
Operating Voltage

Feature	Minimum	Typical	Maximum	Unit
VCC	2.0	3.3	3.6	Volts (V)

Operating Current

Feature	Condition	Minimum	Typical	Maximum	Unit
Operating Current	TX Mode	87 (@ 20 dBm 868 Mhz)			mA
	RX Mode	5.22			mA

La distribución de pines se observa en la siguiente imagen:



Pin No.	Name	Type	Description
1	PA3/UART2_RX	I	Reserved - UART2/LPUART1 Interface (AT Commands and FW Update)
2	PA2/UART2_TX	O	Reserved - UART2/LPUART1 Interface (AT Commands and FW Update)
3	PA15/PIN_A4	I/O	GPIO and ADC
4	PB6/UART1_TX	O	UART1 Interface
5	PB7/UART1_RX	I	UART1 Interface
6	PA1	I/O	GPIO only
7	PA13/SWDIO		Reserved - SWD debug pin (SWDIO)



8	PA14/SWCLK		Reserved - SWD debug pin (SWCLK)
9	PA12/I2C_SCL	I/O	GPIO and I2C (SCL)
10	PA11/I2C_SDA	I/O	GPIO and I2C (SDA)
11	GND		Ground connections
12	RF		RF Port (only available on RAK3172 No-IPEX connector variant)
13	PA7/SPI1_MOSI	I/O	GPIO and SPI (MOSI)
14	PA6/SPI1_MISO	I/O	GPIO and SPI (MISO)
15	PA5/SPI1_CLK	I/O	GPIO and SPI (CLK)
16	PA4/SPI1_NSS	I/O	GPIO and SPI (NSS)
17	GND		Ground connections
18	GND		Ground connections
19	PA8	I/O	GPIO only
20	PA9	I/O	GPIO only
21	BOOT0		Boot0 mode enable pin - high active
22	RST		MCU Reset (NRST)
23	GND		Ground connections
24	VDD		VDD - Voltage Supply
25	PA10/PIN_A3	I/O	GPIO and ADC
26	PB2/PIN_A2	I/O	GPIO and ADC
27	PB12	I/O	10 kΩ internally pulled-up for high freq variant or pulled-down for low freq variant
28	GND		Ground connections
29	PA0	I/O	GPIO only
30	PB5	I/O	GPIO only
31	PB4/PIN_A1	I/O	GPIO and ADC
32	PB3/PIN_A0	I/O	GPIO and ADC

5.1.4.1.2 Implementación de placa de desarrollo

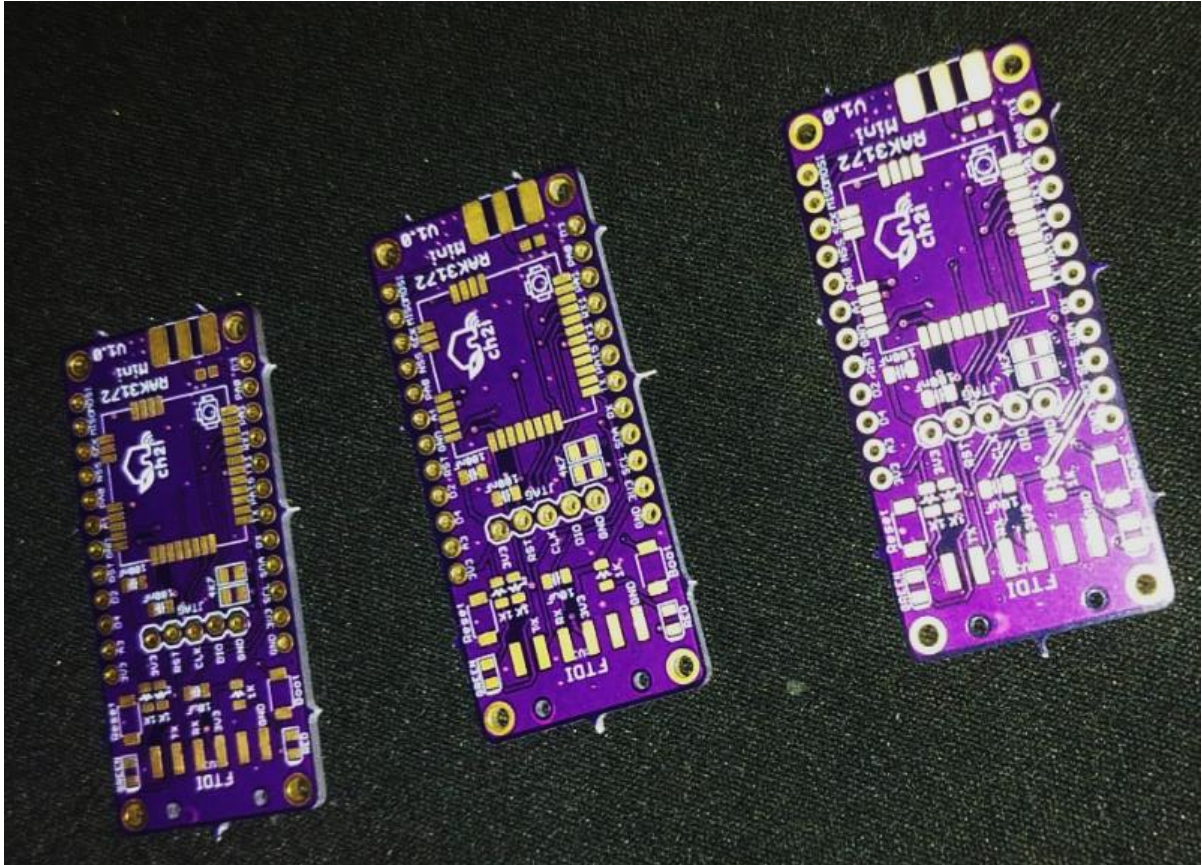
Como resulta evidente, el RAK3172 necesita una placa de desarrollo para su implementación y funcionamiento. Esta placa proporciona la interfaz necesaria para conectar y alimentar el procesador RAK3172, así como para facilitar la comunicación mediante la interfaz UART.

Para la materialización de la placa de desarrollo del RAK3172, se obtuvo el diseño a través de un repositorio en GitHub llamado [RAK3172-Breakout](#) que proporcionaba una implementación confiable y bien documentada.



Después de obtener el diseño, procedimos a enviar los archivos Gerber de la PCB a [Oshpark](#) para su fabricación. Oshpark es un servicio de fabricación de PCB en línea que se distingue por su calidad y eficiencia en la producción de prototipos. Enviando los archivos Gerber a Oshpark, aseguramos que la placa de desarrollo se fabrique con precisión y con los estándares de calidad requeridos.

Se encargaron 3 placas de desarrollo, las cuales podemos ver en la siguiente imagen:



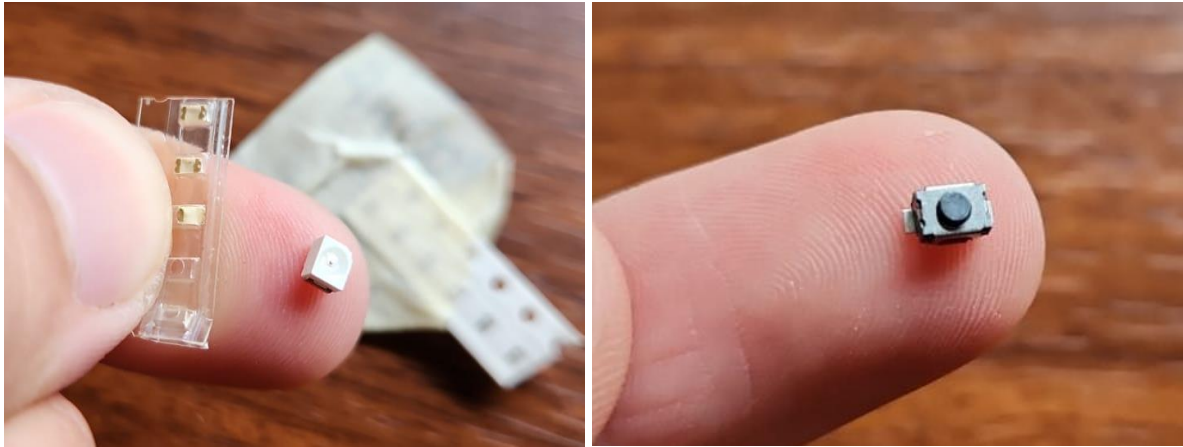
Como se puede observar las placas de desarrollo adquiridas llegaron sin los componentes montados, lo que proporcionó flexibilidad y control total sobre la selección y soldadura de los componentes SMD necesarios. Optamos por realizar este proceso de soldadura manualmente, ya que reducía los costos y disminuía los tiempos de entrega.

A continuación, el listado de componentes requeridos:

Designación	Cantidad	Descripción
C1	1	SMD CAP Ceramic 10uF-25V-10%-X5R;0805
C2, C3	2	SMD CAP Ceramic 100nF-50V-10%-X7R;0805
C4, C5, C6, C7	1	SMD CAP Ceramic 100uF-25V-10%-X5R;0805
RS	1	SMD RES 0R-1%-1/10W;0603
R4, R5	2	SMD RES 4.7K 0805
R1, R2, R3	2	SMD RES 1K 0805
D1	1	SMD LED Clear-Green
D2	1	LED-SMD Red Diffused
K1, K2	2	Tactile Switch 3x2.5mm SMD



Se presentan algunas imágenes de estos componentes para tomar dimensión de su tamaño:

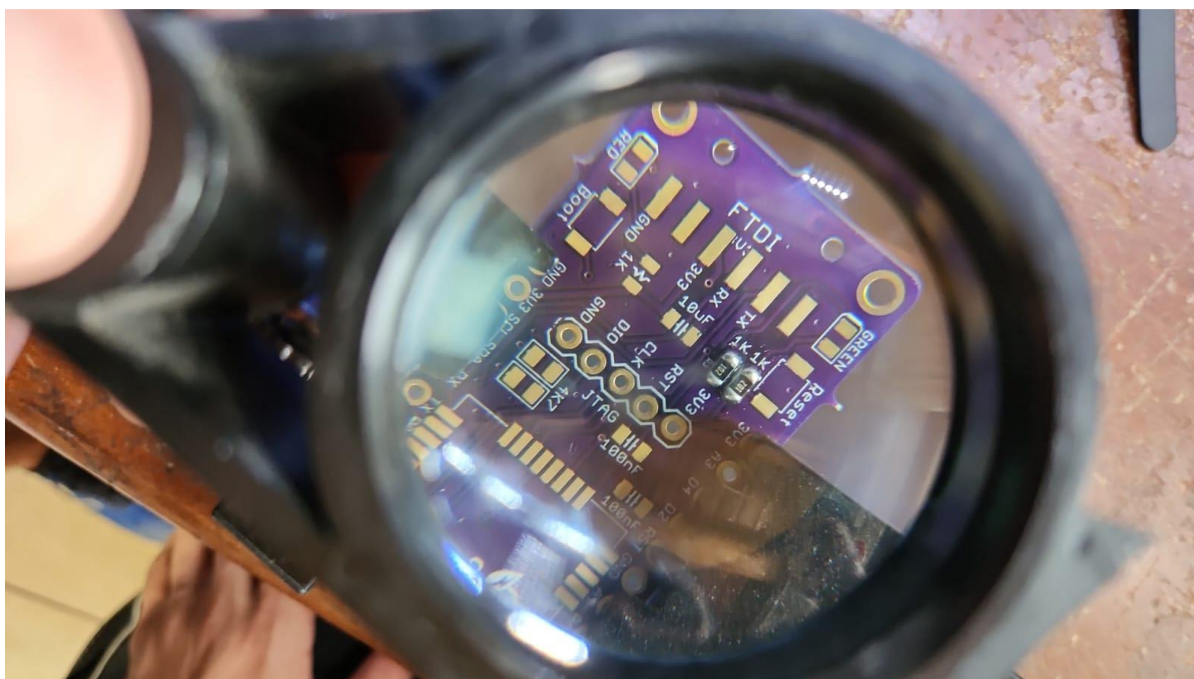


Además de estos componentes se realizó la compra de estaño en pasta, flux y cobre quita estaño.

La soldadura de componentes SMD es un proceso que implica la fijación de componentes electrónicos directamente sobre la superficie de la placa de circuito impreso (PCB). A continuación, se detalla el procedimiento seguido:

1. Se aplica una pequeña cantidad de estaño en pasta en cada uno de los puntos de soldadura correspondientes a los componentes SMD en la PCB.
2. El componente electrónico, ya sea una resistencia, capacitor, LED o el módulo RAK3172, se coloca sobre los puntos de soldadura. Antes de esto, se aplica un poco de flux sobre los contactos del componente para facilitar la soldadura.
3. Utilizando una estación de soldado equipada con una pistola de aire caliente, se aplica calor sobre el componente y los puntos de soldadura. El calor funde el estaño en pasta, creando una conexión eléctrica entre los contactos del componente y la PCB.
4. Gracias al flux y la distribución uniforme del estaño en pasta, el proceso de soldadura se realiza de manera eficiente. La pasta se funde, formando conexiones sólidas y confiables entre el componente y la PCB.
5. Durante la aplicación de calor, el componente tiende a alinearse automáticamente con los puntos de soldadura debido a la tensión superficial del estaño fundido, lo que facilita la correcta posición del componente.
6. Después de la soldadura, se permite que la soldadura se enfríe y solidifique, asegurando conexiones eléctricas estables y duraderas.

A continuación, algunas imágenes durante el proceso de soldado:



El RAK3172 incorpora una antena mediante un conector IPEX, diseñada específicamente para maximizar la eficiencia de la comunicación inalámbrica. Esta antena, fabricada con un tubo de cobre, presenta una longitud de 15,3 cm y es compatible con frecuencias que abarcan desde 433MHz hasta 915MHz. La elección de materiales de alta calidad y la configuración optimizada permiten al dispositivo operar con eficacia en diversas frecuencias, proporcionando una cobertura confiable y estable en entornos de baja potencia. La flexibilidad del conector



IPEX facilita la conexión y desconexión de la antena, ofreciendo así opciones de personalización y adaptación a las necesidades específicas del usuario.

Podemos ver una imagen de la misma a continuación:



Electrical Characteristics

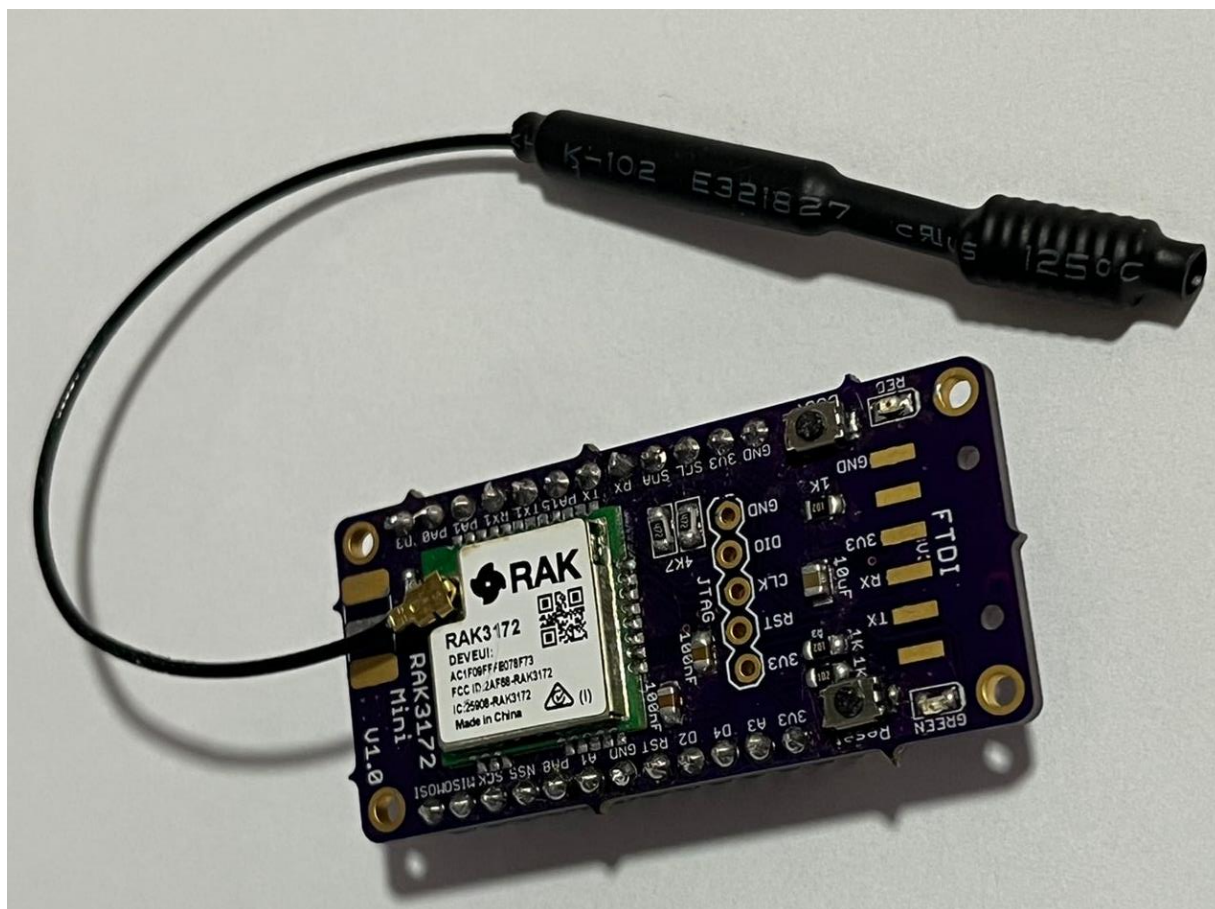
Frequency	433 / 470 / 868 / 915 MHz
V.S.W.R	1:1.5
Polarization	Linear
Impedance	50 Ohm
Antenna Type	Dipole

Material & Mechanical Characteristics

Material of Radiator	Cu
----------------------	----

Environmental

Operation Temperature	- 40 °C ~ + 75 °C
Storage Temperature	- 40 °C ~ + 85 °C



5.1.4.2 Configuración del enlace

Con una interfaz de comunicación UART accesible, el RAK3172 permite una configuración sencilla mediante comandos AT tanto para la comunicación LoRa P2P como para LoRaWAN. La comunicación serial UART se realiza a través del UART2, utilizando el TX y el RX. Los parámetros predeterminados para la comunicación UART se establecen en una velocidad de baudios de 115,200, 8 bits de datos, sin paridad y 1 bit de parada (8-N-1). Además, el módulo admite actualizaciones de firmware a través de este puerto.

Protocolo de comunicación UART

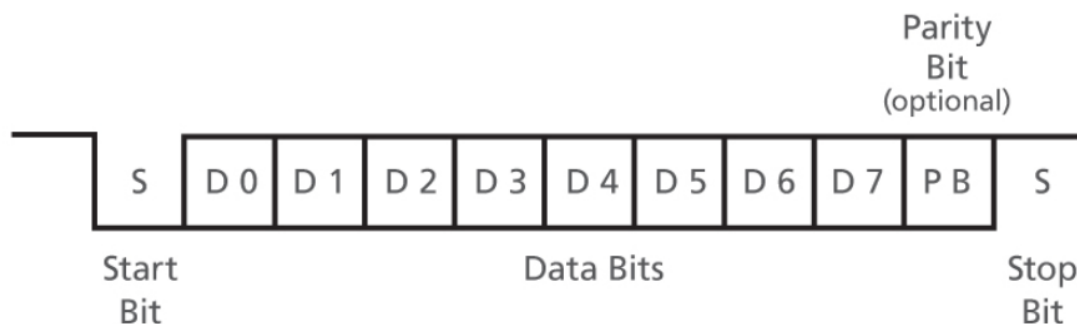
UART significa Universal Asynchronous Receiver/Transmitter y es simplemente una forma elegante de referirse a un puerto serie. UART solo requiere dos líneas: una línea de transmisión (TX) y una línea de recepción (RX).

- TX: Utilizado por el dispositivo para enviar datos.
- RX: Utilizado por el dispositivo para recibir datos.

Las transmisiones UART comienzan con un bit de inicio, donde la línea correspondiente (TX o RX) es llevada a un nivel bajo por la parte que envía. Luego, se envían de cinco a ocho bits de datos.



El siguiente diagrama muestra un escenario donde se envían ocho bits:



Después de los datos, se envía un bit de paridad opcional, seguido de uno o dos bits de parada, donde el módulo que envía lleva el pin a un nivel alto.

Para que este protocolo funcione, el emisor y el receptor deben ponerse de acuerdo en algunas cosas:

- La cantidad de bits de datos que se envían con cada paquete (cinco a ocho).
- La velocidad a la que se debe enviar los datos (la tasa de baudios).
- Si hay un bit de paridad después de los datos y si es alto o bajo.
- La cantidad de bits de parada que se enviarán al final de cada transmisión.

Comandos AT

Los Comandos AT son instrucciones específicas enviadas a un módulo o dispositivo a través de una interfaz, como UART (Universal Asynchronous Receiver/Transmitter). Estos comandos siguen una convención específica y permiten controlar y configurar diversas funciones del dispositivo.

El propósito principal de los Comandos AT es proporcionar una interfaz de usuario simple para interactuar con el dispositivo. Estos comandos permiten configurar parámetros, realizar pruebas, obtener información del dispositivo y controlar su funcionamiento.

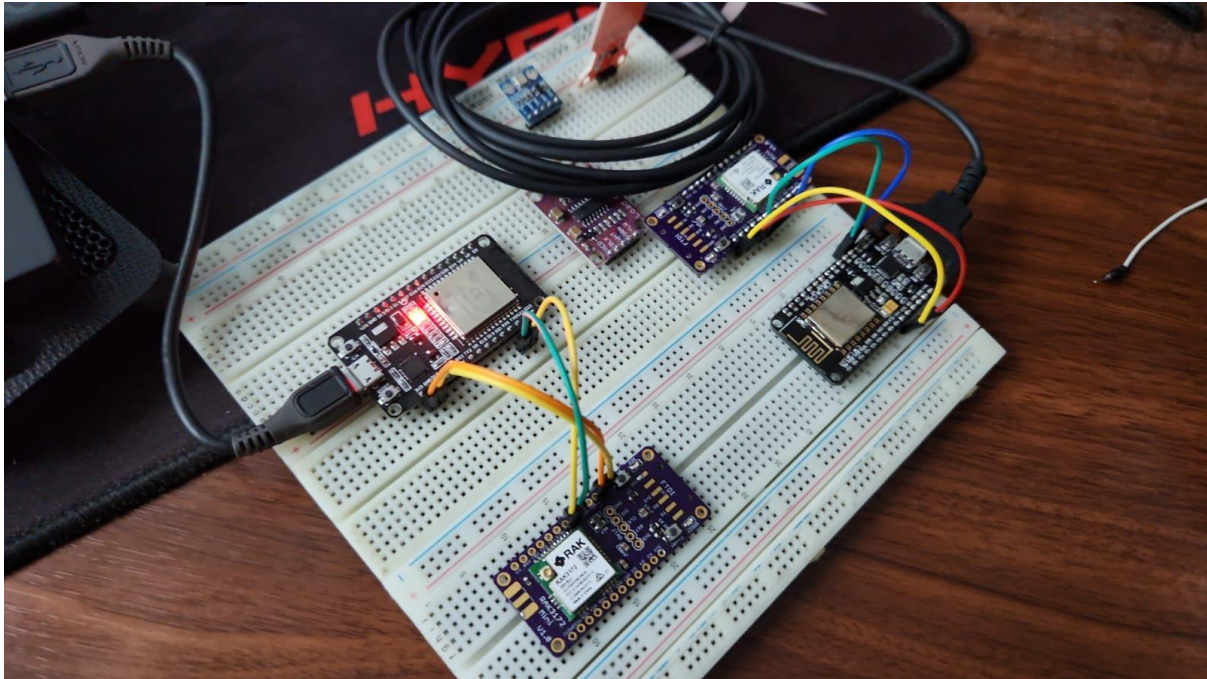
Los comandos AT generalmente siguen una estructura específica. Comienzan con las letras "AT" seguidas de un comando específico y, a veces, parámetros adicionales. Por ejemplo, "AT+BAUD=9600" podría ser un comando para establecer la velocidad de baudios a 9600.

Para enviar un comando AT a un dispositivo, se utiliza un terminal serie o software de comunicación serial. El usuario envía la cadena de texto correspondiente al comando AT, y el dispositivo responde con información o realiza la acción solicitada.

Para llevar a cabo las pruebas de comunicación punto a punto (P2P) entre los dos módulos RAK3172, se estableció una conexión directa entre los pines relevantes de los dos dispositivos. El pin TX del ESP32 se conectó al pin RX del módulo LoRa (RAK3172), mientras que el pin RX del ESP32 se conectó al pin TX del módulo LoRa. Además, se aseguró la conexión de los pines de tierra (GND) y alimentación (3.3V) del ESP32 a los pines correspondientes en el módulo LoRa RAK3172. Esta configuración permitió una comunicación bidireccional efectiva entre los dos dispositivos, facilitando las pruebas de funcionalidad en un entorno de comunicación punto a punto.



A continuación, podemos ver una fotografía durante las pruebas de comunicaciones entre ambos módulos:



Para configurar los dos módulos RAK3172 y que funcionen en modo LoRa P2P se realizó la conexión que se observa en la fotografía. Cada uno de los módulos se comunica con un ESP32 que será el encargado de introducir los comandos AT. La configuración de cada RAK3172 se puede realizar de forma individual, pero para probar el modo LoRa P2P será necesario tener ambas unidades conectadas simultáneamente.

Para comenzar, se prueba la comunicación serial y se verifica que la configuración actual esté funcionando enviando estos dos comandos AT:

```
AT
```

```
ATE
```

La activación del comando ATE permite que los comandos ingresados en el módulo se repitan, ofreciendo utilidad para rastrear las instrucciones y abordar posibles problemas.

Al ingresar los dos comandos, se recibe una confirmación mediante el mensaje "OK". Una vez configurado ATE, es posible visualizar todos los comandos ingresados, junto con sus correspondientes respuestas, permitiendo monitorear el flujo de acciones. Para verificar la correcta configuración, se realiza un nuevo intento con el comando AT y se observa la respuesta en el terminal seguido de "OK".

La configuración del RAK3172 para operar en modo LoRa P2P implica la modificación del comando de modo de red LoRa en ambos módulos RAK3172.



```
AT+NWM=0
```

El comando AT+NWM, con el parámetro "mode", puede ajustarse a 0 para habilitar el modo LoRa P2P o a 1 para activar el modo LoRaWAN.

Es necesario introducir la configuración P2P en ambos módulos RAK3172. Los parámetros deben ser exactamente iguales en los dos módulos.

```
AT+P2P=915000000:7:0:0:10:20
```

Para esta configuración P2P, los parámetros LoRa son los siguientes:

- Link frequency: **915000000 Hz**
- Spreading factor: **7**
- Bandwidth: 0 = **125 kHz**
- Coding Rate: 0 = **4/5**
- Preamble Length: **10**
- Power: **20 dBm**

Este comando (AT+P2P) proporciona la configuración de todos los parámetros para el modo P2P. A continuación vemos una tabla de utilización e información:

Command	Input Parameter	Return Value	Return Code
AT+P2P?	-	AT+P2P : get or set all P2P parameters	OK
AT+P2P=?	-	-	OK AT_BUSY_ERROR
AT+P2P= <param>	<Frequency>:<Spreading Factor>:<Bandwidth>:<Code Rate>:<Preamble Length>:<TX Power>	-	OK AT_PARAM_ERROR

AT_PARAM_ERROR se devuelve cuando se establece un valor incorrecto o malformado.

Frecuencia = {150000000-960000000}, SF = {6, 7, 8, 9, 10, 11, 12}, Ancho de banda {0=125, 1=250, 2=500, 3=7.8, 4=10.4, 5=15.63, 6=20.83, 7=31.25, 8=41.67, 9=62.5}, CR = {4/5=0, 4/6=1, 4/7=2, 4/8=3}, Longitud de preámbulo = {2-65535}, PT = {5-22}.

Para configurar un módulo como receptor (RX) en el contexto de LoRa P2P, es necesario establecer el valor correspondiente al comando de recepción P2P.

En el modo LoRa P2P en RX, la duración configurable varía de 1 a 65533 ms. En este ejemplo, el dispositivo estará atento a la recepción de paquetes LoRa P2P durante 30000 ms o 30 segundos. Después de este intervalo, desactivará automáticamente el modo RX y cambiará al modo TX. En caso de no recibir ningún paquete durante el período mencionado, se generará el callback +EVT: RXP2P RECEIVE TIMEOUT.



```
AT+PRECV=30000
```

Si el valor de AT+PRECV se establece en 65535, el dispositivo escuchará los paquetes LoRa P2P sin un tiempo de espera, pero dejará de escuchar una vez que se reciba un paquete LoRa P2P. Después de recibir los paquetes, desactivará el modo RX y cambiará automáticamente al modo TX nuevamente.

```
AT+PRECV=65535
```

Si el valor de AT+PRECV se establece en 65534, el dispositivo escuchará continuamente los paquetes LoRa P2P sin ningún tiempo de espera. Permanecerán continuamente en modo RX hasta que AT+PRECV se establezca en 0.

```
AT+PRECV=65534
```

Si el valor de AT+PRECV se establece en 0, el dispositivo dejará de escuchar los paquetes LoRa P2P. Desactivará el modo RX de LoRa P2P y cambiará al modo TX.

```
AT+PRECV=0
```

Con un módulo configurado como Transmisor (TX) y el otro dispositivo como Receptor (RX), es posible intentar enviar o transmitir datos de carga útil P2P.

```
AT+PSEND= <payload>
```

AT_PARAM_ERROR se generará al intentar establecer un valor incorrecto o malformado. AT_BUSY_ERROR ocurrirá si el dispositivo aún se encuentra en modo RX y se intenta enviar o reconfigurar el período de RX. En caso de que el comando AT+PRECV se haya establecido en 65534, es necesario ejecutar AT+PRECV=0 primero. Esto permitirá la reconfiguración del estado TX y RX, evitando así el error AT_BUSY_ERROR. Cuando se trata del campo <payload>, su longitud debe oscilar entre 2 y 500 dígitos, y debe ser un número par de dígitos. Además, solo se permiten caracteres 0-9, a-f, A-F. Este campo representa de 1 a 256 números hexadecimales. Por ejemplo, si la carga útil es como 0x03, 0xAA, 0x32, el comando AT correspondiente sería AT+PSEND=03AA32.



5.1.4.3 Transmisión, recepción y manejo de datos

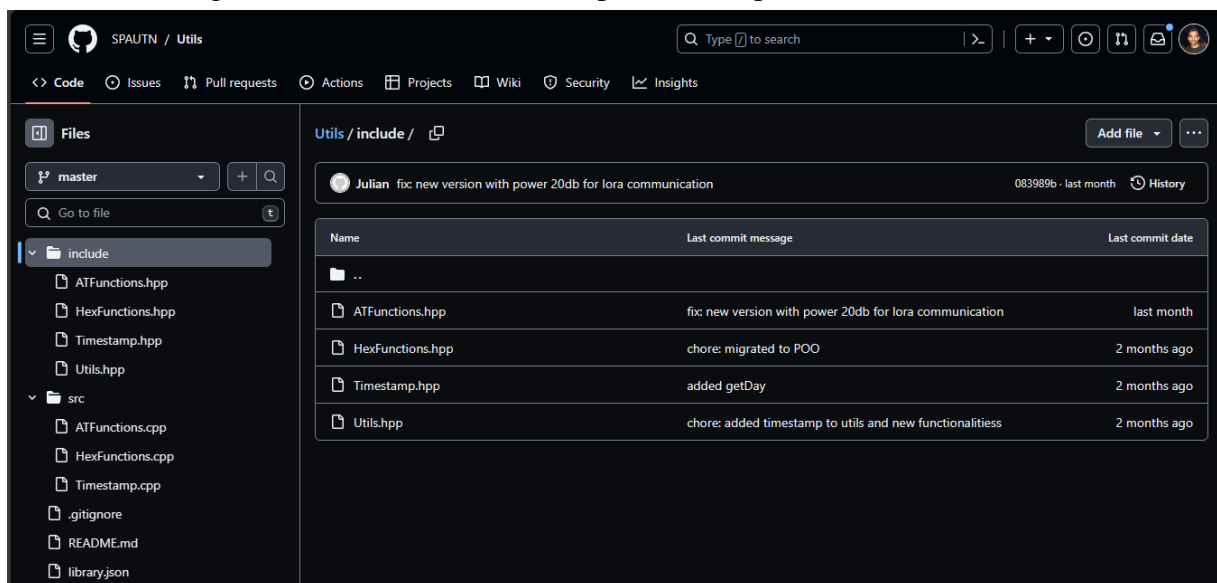
La efectiva comunicación entre los módulos RAK3172 se logra mediante la implementación de un código especializado desarrollado en C++ con un enfoque en programación orientada a objetos (POO) que utiliza comandos AT. Este código se encuentra alojado en el repositorio de GitHub denominado "Utils" en la organización "SPAUTN". Esta biblioteca de utilidades está diseñada específicamente para facilitar la comunicación serial entre un microcontrolador ESP32 y el módulo LoRa RAK3172 a través del puerto Serial2.

Incluye funciones destinadas a:

- Manipulación de datos hexadecimales - **HexFunctions**
- Envío y recepción de datos - **ATFunctions**
- Manipulación de tiempo - **Timestamp**

En el repositorio "Utils", cada programa escrito en C++ (.cpp) se acompaña de su correspondiente archivo de encabezado (.hpp). Esta práctica asegura una organización clara y modular del código, permitiendo entender y utilizar las funcionalidades de manera eficiente.

La estructura organizativa de archivos en el repositorio se puede observar a continuación:



Ahora abordaremos cada una de las funcionalidades desarrolladas que facilitan y permiten la transmisión y recepción de datos entre ambos módulos LoRa RAK3172.

HexFunctions

El objetivo de este programa es simple: Convertir una cadena de caracteres (ASCII) a valores hexadecimales y viceversa.

Esto es necesario debido a que el comando AT+PSEND está diseñado para recibir datos en formato hexadecimal. Al convertir la cadena de caracteres a hexadecimal, se asegura la compatibilidad directa con este comando. La representación hexadecimal es la forma estándar



de transmitir datos binarios a través de comandos AT en muchos dispositivos, incluidos los módulos LoRa.

Otras razones a tener en cuenta son las siguientes:

- La representación hexadecimal permite una transmisión más eficiente de datos, ya que cada byte de la cadena se convierte en dos caracteres hexadecimales, lo que reduce la cantidad de datos transmitidos.
- En comparación con la representación de caracteres ASCII, la representación hexadecimal utiliza menos ancho de banda y acorta la duración de la transmisión, lo que es crucial en entornos con restricciones de ancho de banda o en situaciones donde se busca optimizar el rendimiento.
- La conversión a hexadecimal ayuda a mantener la integridad de los datos durante la transmisión. Cada byte se representa de manera única, lo que facilita la identificación y reconstrucción precisa de la información en el extremo receptor.

A continuación, el código de HexFunctions.cpp y su explicación:

```
#include "../include/HexFunctions.hpp"

String HexFunctions::hexToASCII(String hex) {
    String ascii = "";
    for(int i=0; i<hex.length(); i+=2) {
        String ch = hex.substring(i, i+2);
        char c = (char) (int)strtol(ch.c_str(), NULL, HEX);
        ascii += (char) c;
    }
    return ascii;
}

String HexFunctions::asciiToHex(String ascii) {
    String hex = "";
    for(int i=0; i<ascii.length(); i++) {
        char c = ascii.charAt(i);
        int code = c;
        String h = String(code, HEX);
        hex += h;
    }
    return hex;
}
```

- **Método hexToASCII:** Convierte una cadena hexadecimal a su representación ASCII. Itera sobre cada par de caracteres en la cadena hexadecimal, convierte el par a su valor decimal y luego a un carácter ASCII.
- **Método asciiToHex:** Convierte una cadena ASCII a su representación hexadecimal. Itera sobre cada carácter en la cadena ASCII, obtiene el valor decimal del carácter y lo convierte a su representación hexadecimal.



ATFunctions

Este programa proporciona funciones esenciales para la interacción con los módulos LoRa mediante comandos AT. Está diseñado para facilitar la lectura de respuestas del módulo, enviar comandos AT y, específicamente, para enviar paquetes en modo punto a punto (P2P).

En este caso, primero se expone el archivo ATFunctions.hpp debido a que contiene macros importantes que configuran a los módulos RAK3172.

```
#include <Arduino.h>
#include "HexFunctions.hpp"

// ----- Macros for the AT commands -----
#define AT_RESET "ATZ"

// Baud rate configuration
#define AT_BAUD_38400_CONFIG_SET "AT+BAUD=38400"
#define AT_BAUD_115200_CONFIG_SET "AT+BAUD=115200"

// P2P commands
#define AT_P2P_CONFIG_SET "AT+P2P=915000000:7:0:0:10:20"
#define AT_P2P_CONFIG_GET "AT+P2P=?"
#define AT_P2P_CONFIG_TX_SET "AT+PRECV=0"
#define AT_P2P_PSEND_HEADER "AT+PSEND="
#define AT_CONTINUOUS_PRECV_CONFIG_SET "AT+PRECV=65534"
#define AT_SEMICONTINUOUS_PRECV_CONFIG_SET "AT+PRECV=65535"
#define AT_SET_P2P_MODE "AT+NWM=0"

// Instructions
#define POLL_COMMAND "POLL"
#define IRR_COMMAND "IRR"

#ifndef AT_FUNCTIONS_HPP
#define AT_FUNCTIONS_HPP

class ATFunctions {
public:
    /**
     * @brief Read the serial port 2 and return the response
     *
     * @param serialAT Stream with the serial port
     */
    String readSerial(Stream &serialAT);

    /**
     * @brief Send AT command to the module and return the response
     *
     * @param serialAT Stream with the serial port
     * @param command String with the AT command
     * @return String
     */
    String sendATCommand(Stream &serialAT, String command);

    /**
     * @brief Send P2P packet to the module and return the response
     * receive the packet data in ascii format
     *
     * @param serialAT Stream with the serial port
     * @param packet String with the packet data
     */

```



```
    * @return String with the response from the module
    */
    String sendP2PPacket(Stream &serialAT, String packet);

private:
    HexFunctions hexFunctions;
};
#endif
```

Ahora vemos el programa principal ATFunctions.cpp y su explicación:

```
#include "../include/ATFunctions.hpp"

String ATFunctions::readSerial(Stream &serialAT) {
    String readed = "";
    while(serialAT.available()>0) {
        char c = serialAT.read();
        readed += c;
        delay(5);
    }
    readed.trim();
    return readed;
}

String ATFunctions::sendATCommand(Stream &serialAT, String command) {
    String response = "";
    bool configCommand = command.indexOf('?') == -1;
    if(configCommand) {
        delay(1000);
    }
    serialAT.flush();
    serialAT.println();
    serialAT.println(command);
    delay(500);
    serialAT.flush();
    response = readSerial(serialAT);
    if( !configCommand ) {
        response = response.substring(0, response.indexOf('\r'));
    }
    response.trim();
    return response;
}

String ATFunctions::sendP2PPacket(Stream &serialAT, String packet) {
    String response = sendATCommand(serialAT, AT_P2P_PSEND_HEADER + this-
>hexFunctions.asciiToHex(packet));
    return response == "" ? "OK" : response;
}
```

- **Método readSerial:** Lee y devuelve los datos disponibles en el puerto serial.
- **Método sendATCommand:** Envía un comando AT al módulo LoRa y obtiene la respuesta.
- **Método sendP2PPacket:** Envía un paquete en modo punto a punto (P2P) al módulo LoRa y obtiene la respuesta.



Timestamp

Este programa contiene métodos para obtener información detallada sobre la marca de tiempo local en formato ISO 8601. Estos métodos incluyen la obtención de la marca de tiempo completa, así como funciones para extraer el día, la hora, los minutos y los segundos de la marca de tiempo.

El programa principal Timestamp.cpp es el siguiente:

```
#include "Timestamp.hpp"

String Timestamp::getLocalTimeStamp() {
    time_t now;
    struct tm timeInfo;
    char strftime_buf[64];

    time(&now);
    localtime_r(&now, &timeInfo);

    char timeString[30];
    snprintf(timeString, sizeof(timeString), "%04d-%02d%02dT%02d:%02d:%02d",
        timeInfo.tm_year + 1900, timeInfo.tm_mon + 1, timeInfo.tm_mday,
        timeInfo.tm_hour, timeInfo.tm_min, timeInfo.tm_sec);

    return timeString;
}

String Timestamp::getDay() {
    return this -> getLocalTimeStamp().substring(8,10);
}

String Timestamp::getHours() {
    return this -> getLocalTimeStamp().substring(11,13);
}

String Timestamp::getMinutes() {
    return this -> getLocalTimeStamp().substring(14,16);
}

String Timestamp::getSeconds() {
    return this -> getLocalTimeStamp().substring(18,19);
}
```

- **Método getLocalTimeStamp:** Devuelve la marca de tiempo local en formato ISO 8601.
- **Método getDay:** Devuelve el día de la marca de tiempo local. Utiliza getLocalTimeStamp y extrae la porción de la cadena correspondiente al día.
- **Método getHours:** Devuelve la hora de la marca de tiempo local. Utiliza getLocalTimeStamp y extrae la porción de la cadena correspondiente a las horas.
- **Método getMinutes:** Devuelve los minutos de la marca de tiempo local. Utiliza getLocalTimeStamp y extrae la porción de la cadena correspondiente a los minutos.
- **Método getSeconds:** Devuelve los segundos de la marca de tiempo local. Utiliza getLocalTimeStamp y extrae la porción de la cadena correspondiente a los segundos.



5.1.5 Módulo ADCSPA

Este módulo es un componente esencial del Sistema SPA, representa la pieza encargada de recopilar, procesar y transmitir la información clave proveniente de los diversos sensores distribuidos en la estación meteorológica y el lisímetro de pesada, como así también de ejecutar el control hídrico. El objetivo fundamental radica en la adquisición precisa de datos hídricos y ambientales. Este dispositivo se destaca como el núcleo tecnológico encargado de capturar las variaciones provenientes de todos los sensores en el campo. En las siguientes secciones, exploraremos su diseño, desarrollo de firmware y su interacción clave con otros componentes del sistema.

5.1.5.1 Hardware

5.1.5.1.1 Componentes principales

Aquí, se presentan los dos componentes principales que desempeñan roles importantes:

Microcontrolador ESP32

El corazón del módulo ADCSPA es el microcontrolador ESP32 producido por Espressif Systems. Equipado con una amplia gama de periféricos, este microcontrolador no solo coordina la ejecución del firmware, sino que también facilita la gestión de datos provenientes de los sensores. La placa de desarrollo DevKit ESP32 es una herramienta potente que integra en una placa el SoM ESP-WROOM-32 que tiene como base al SoC ESP32, el convertor USB-serial CP2102 necesario para programar por USB el ESP32, reguladores de voltaje y leds indicadores. La plataforma ESP32 es la evolución del ESP8266 mejorando sus capacidades de comunicación y procesamiento computacional. A nivel de conectividad permite utilizar diversos protocolos de comunicación inalámbrica como: WiFi, Bluetooth y BLE. En cuanto a procesamiento posee un CPU 32-bit de dos núcleos de hasta 240Mhz. Además, incluye internamente una gran cantidad de periféricos para la conexión con: sensores táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

Especificaciones técnicas:

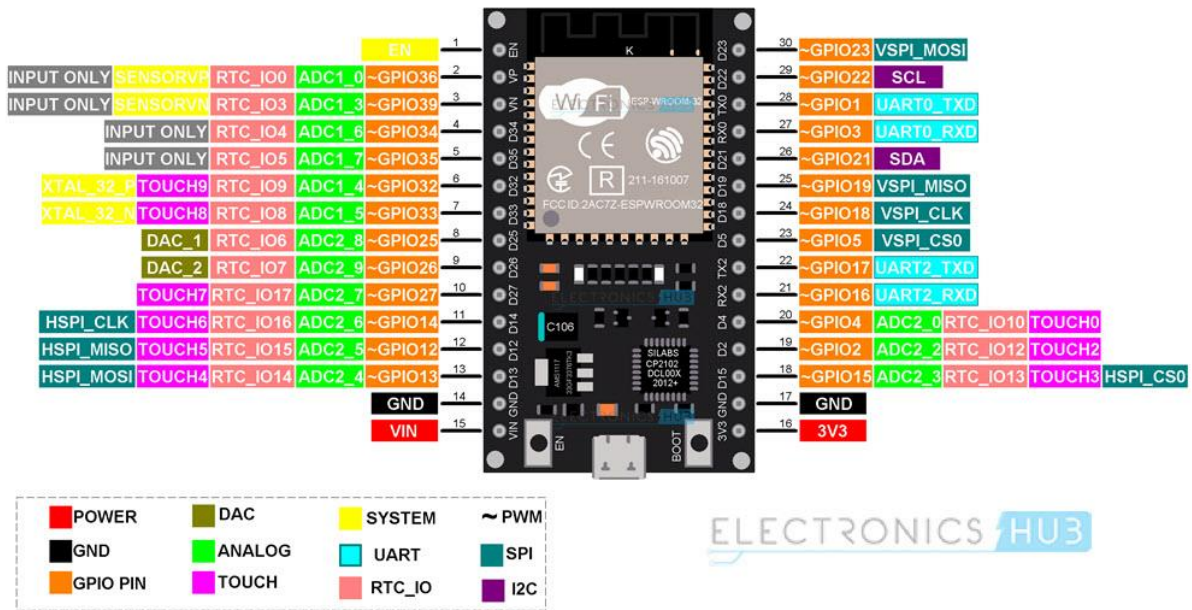
- Voltaje de Alimentación (USB): 5V DC
- Placa: ESP32 DEVKIT V1 (Espressif)
- SoM: ESP-WROOM-32 (Espressif)
- SoC: ESP32 (ESP32-D0WDQ6)
- CPU: Dual-Core Tensilica Xtensa LX6 (32 bit)
- Frecuencia de Reloj: hasta 240Mhz
- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- Memoria:
 - 448 Kbyte ROM
 - 520 Kbyte SRAM



16 Kbyte SRAM in RTC
QSPI Flash/SRAM, 4 MBytes

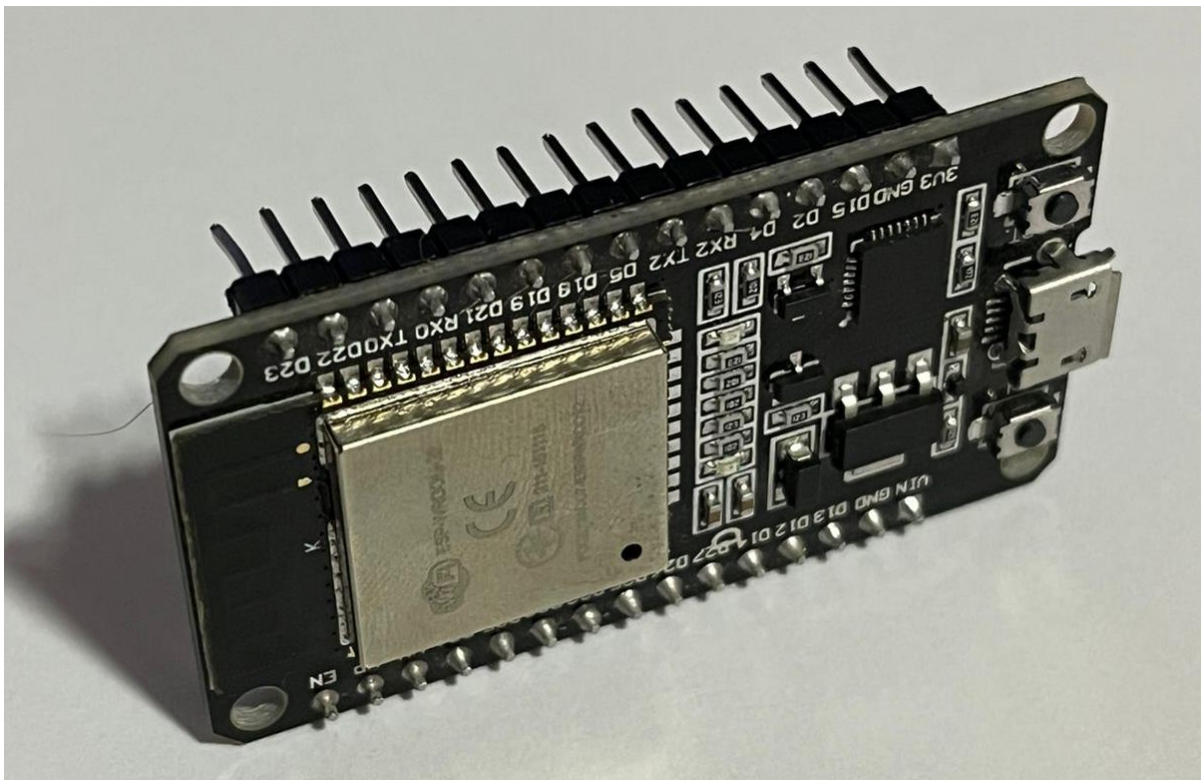
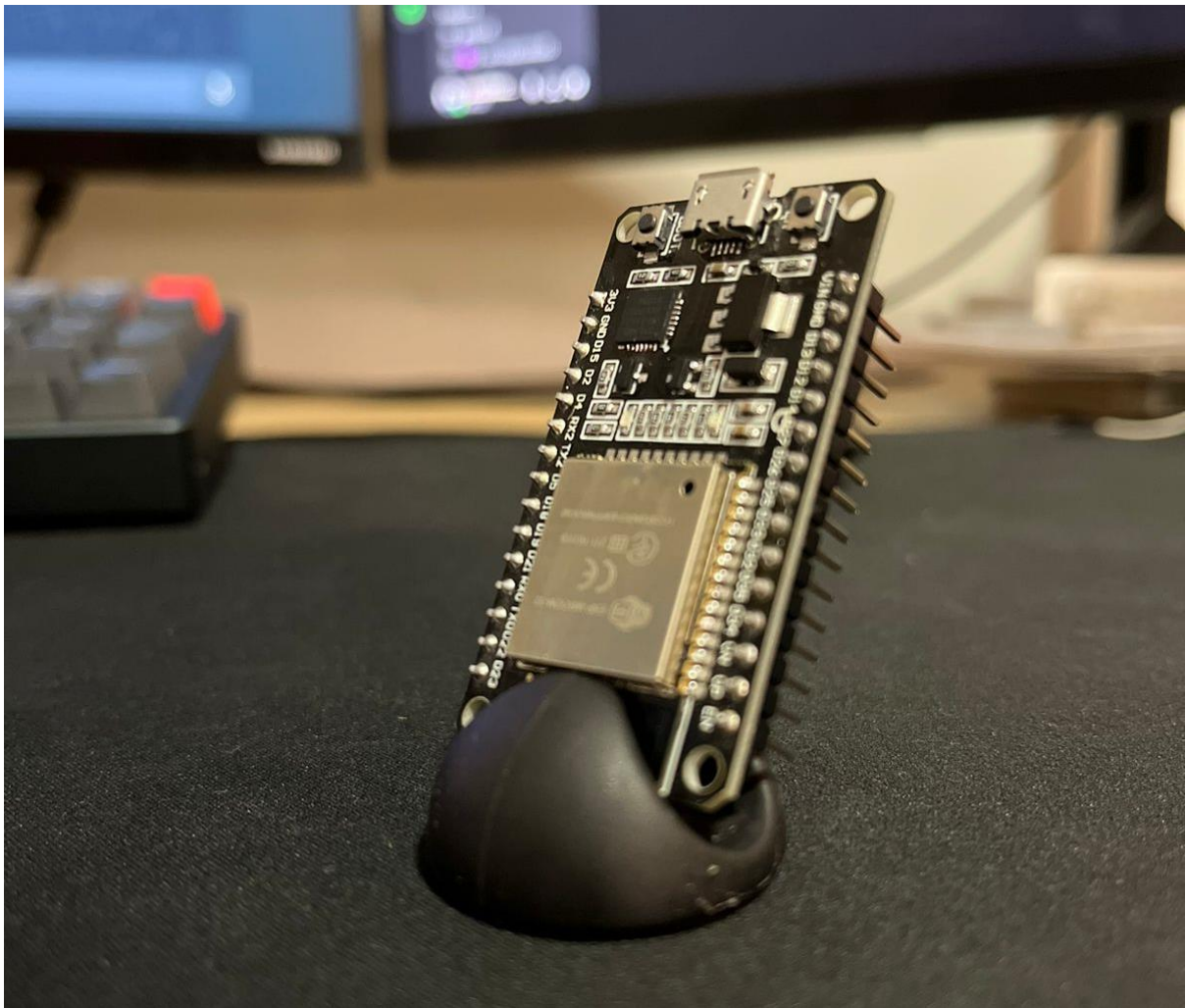
- Pines: 30
- Pines Analógicos ADC: 18 (3.3V, 12bit: 4095)
- Conversor Digital a Analógico DAC: 2 (8bit)
- UART: 2
- Chip USB-Serial: CP2102
- Dimensiones: 55*28 mm

La distribución de pines es la siguiente:



ELECTRONICS HUB

A continuación, algunas fotografías:



LoRa RAK3172

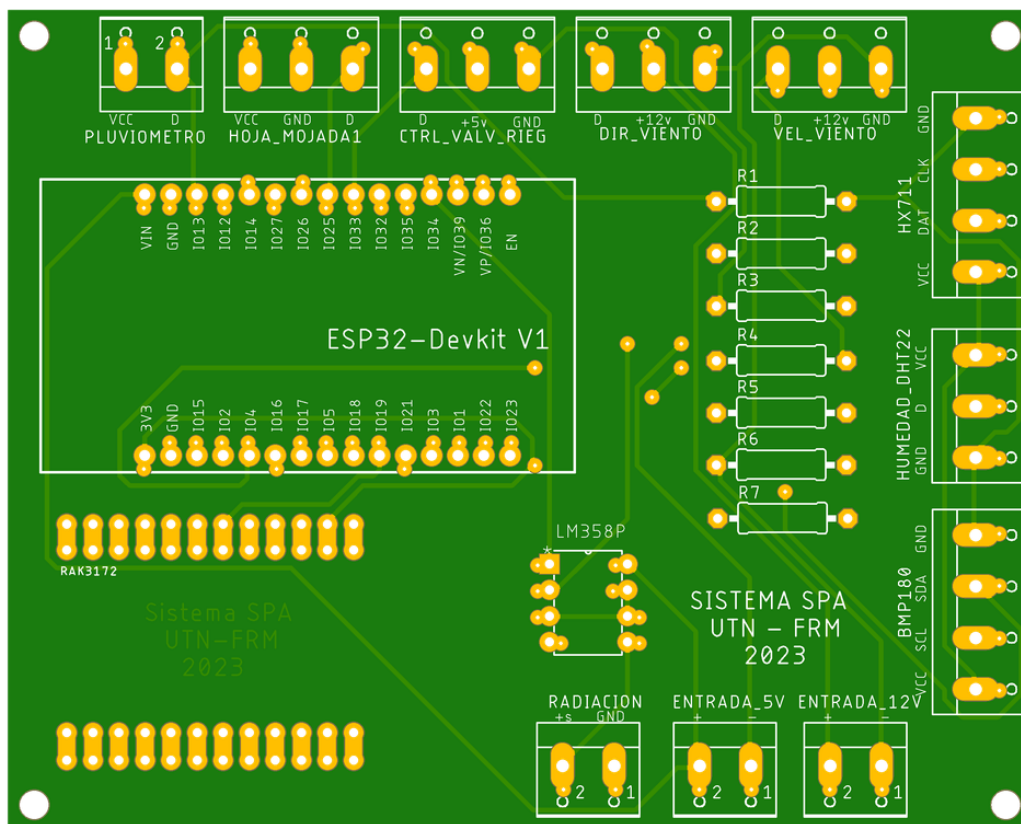
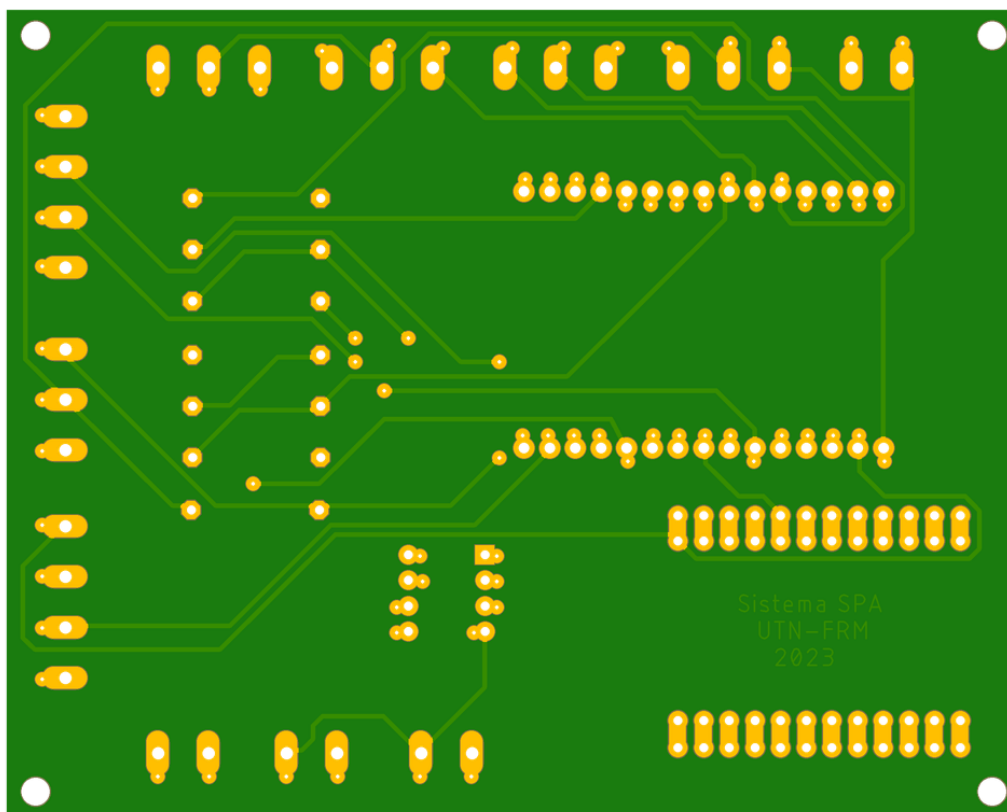
Este componente ya fue desarrollado en la sección **5.2.4.3.1 Módulo RAK3172**, por lo que no volverá a describirse. Se presenta una imagen para tener de referencia:



5.1.5.1.2 Diseño y desarrollo de la PCB

El diseño y desarrollo de la PCB del Módulo ADCSPA se llevó a cabo con la ayuda de Eagle, una herramienta de diseño electrónico ampliamente reconocida por su eficiencia y versatilidad. Eagle, desarrollado por Autodesk, proporciona una interfaz intuitiva y potentes capacidades de diseño que facilitan la creación de circuitos electrónicos. Eagle ofrece una interfaz de usuario amigable que facilita la creación de esquemáticos y la disposición de componentes en la PCB. Esto agiliza el proceso de diseño y permite una visualización clara de la estructura del circuito. Además, la amplia librería de Eagle incluye una variedad de componentes electrónicos estándar, lo que simplifica la selección y colocación de elementos esenciales, como el microcontrolador ESP32 y el módulo LoRa RAK3172.

La función de enrutamiento automático y manual en Eagle permite trazar las conexiones de manera eficiente, minimizando cruces y asegurando una distribución ordenada de las pistas en la PCB. También permite la incorporación de comentarios directamente en el diseño,



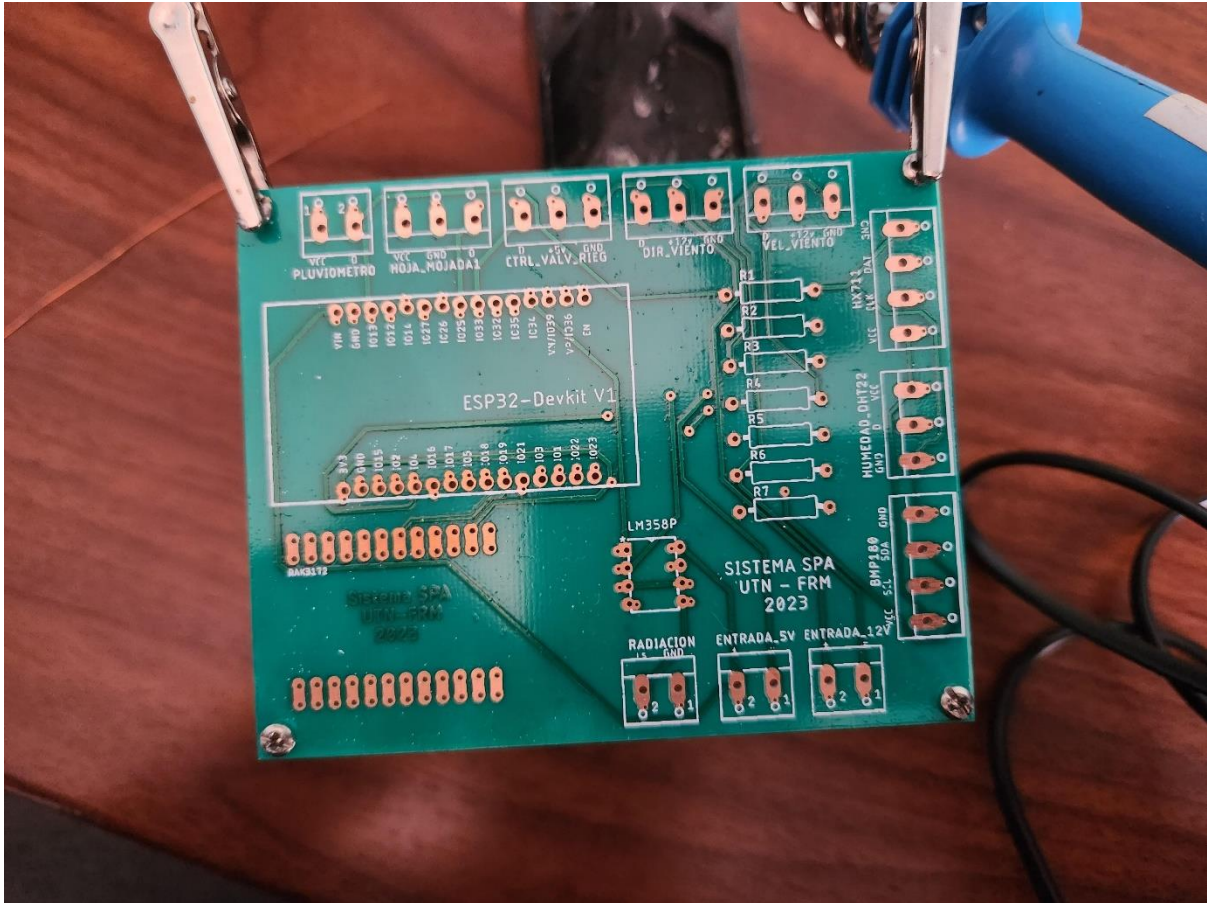
Como se puede observar en el centro hacia la izquierda se encuentra el microcontrolador ESP32 y debajo de él se encuentra el modulo LoRa RAK3172. En la esquina superior izquierda comienzan a posicionarse las distintas borneras encargadas de posibilitar la conexión de todos los sensores del Sistema SPA, podemos notar las siguientes: pluviómetro, humedad de hoja,



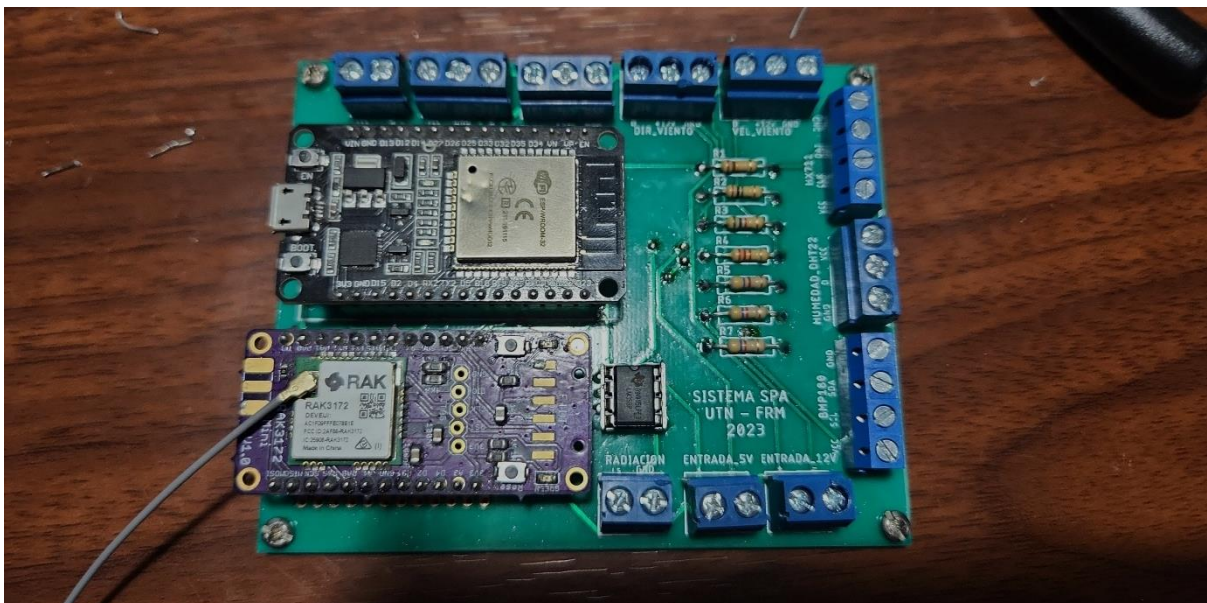
control de la válvula de riego, dirección del viento, velocidad del viento, HX711 (lisímetro de pesada), humedad ambiente, temperatura y presión, las entradas de 5V y 12V, y finalmente el sensor de radiación solar.

Las dimensiones de la PCB resultaron de 80mm x 100mm, doble faz con máscara antisoldante y serigrafía incluida. La empresa que materializó el diseño es complex PCB.

A continuación, una imagen de la PCB antes de colocar los componentes:



Finalmente, una fotografía de la PCB completa con los componentes montados:





5.1.5.1.3 Diseño e impresión carcasa 3D

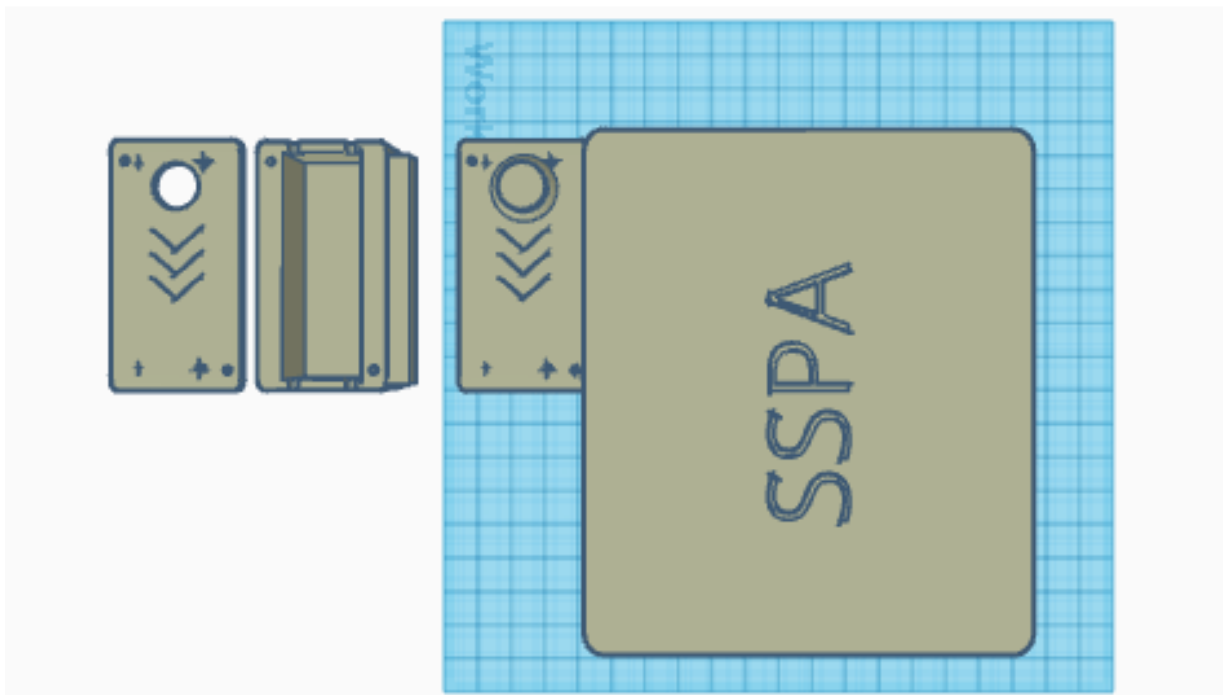
La integridad y protección de la PCB del módulo ADCSPA son importantes para su desempeño en el entorno agrícola. En este contexto, la carcasa 3D se presenta como un componente diseñado para albergar y resguardar la PCB. Este proceso abarca desde la conceptualización y diseño en Tinkercad hasta la materialización física mediante una impresora 3D.

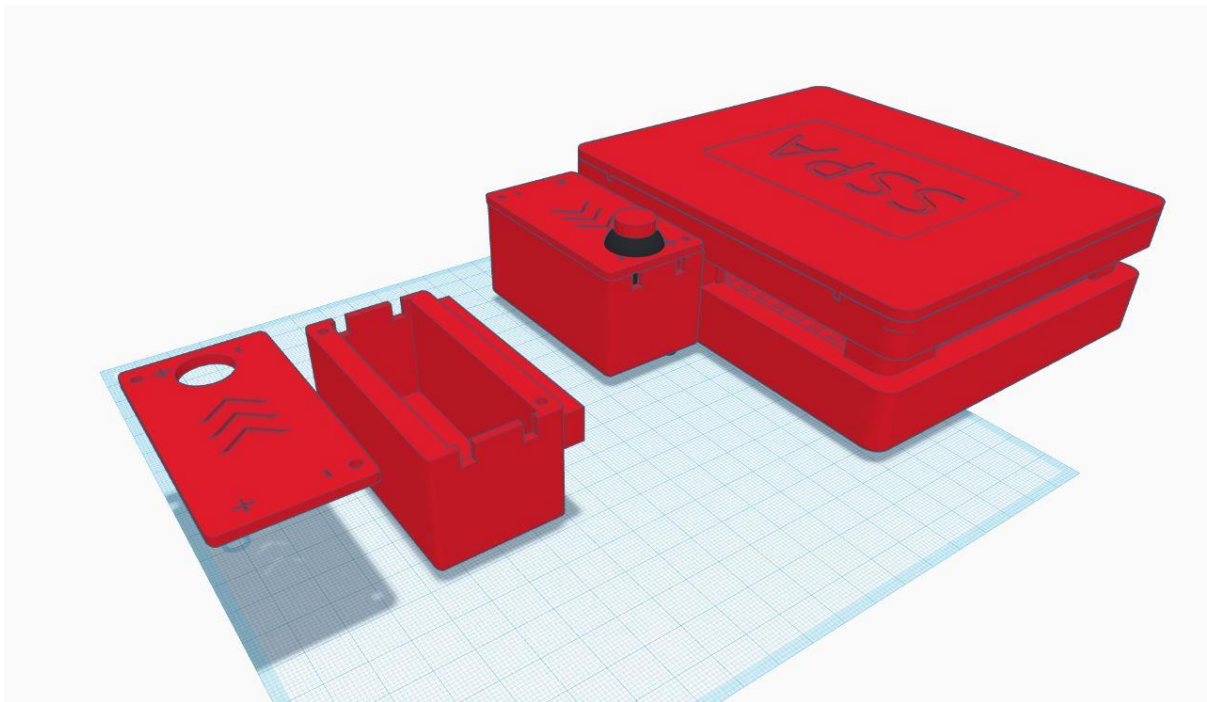
La creación del diseño de la carcasa se llevó a cabo en Tinkercad, una plataforma de modelado 3D accesible y versátil. Este proceso implica la conceptualización tridimensional de la carcasa, considerando factores como el tamaño de la PCB, disposición de componentes y accesibilidad para futuras modificaciones.

Consideraciones de diseño:

- **Espacio:** La carcasa fue diseñada para albergar la PCB, garantizando un buen ajuste y permitiendo la conexión de los componentes periféricos.
- **Accesibilidad:** Se incorporaron aberturas estratégicas para permitir el acceso a puertos de comunicación, conexiones de sensores y otros elementos esenciales sin comprometer la integridad de la carcasa.
- **Ventilación:** Se consideraron aberturas de ventilación para evitar el sobrecalentamiento de la PCB, contribuyendo así a un funcionamiento eficiente y prolongado.

A continuación, se presentan algunas imágenes del diseño en Tinkercad:

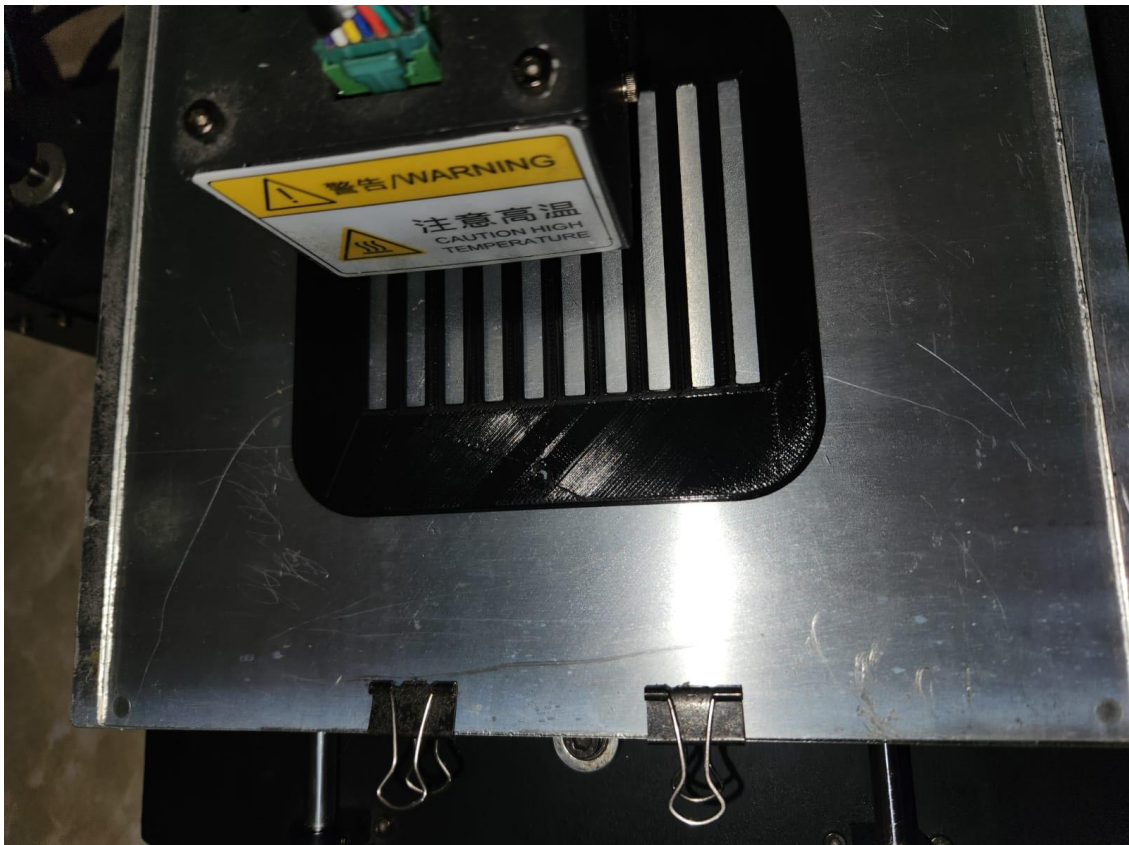


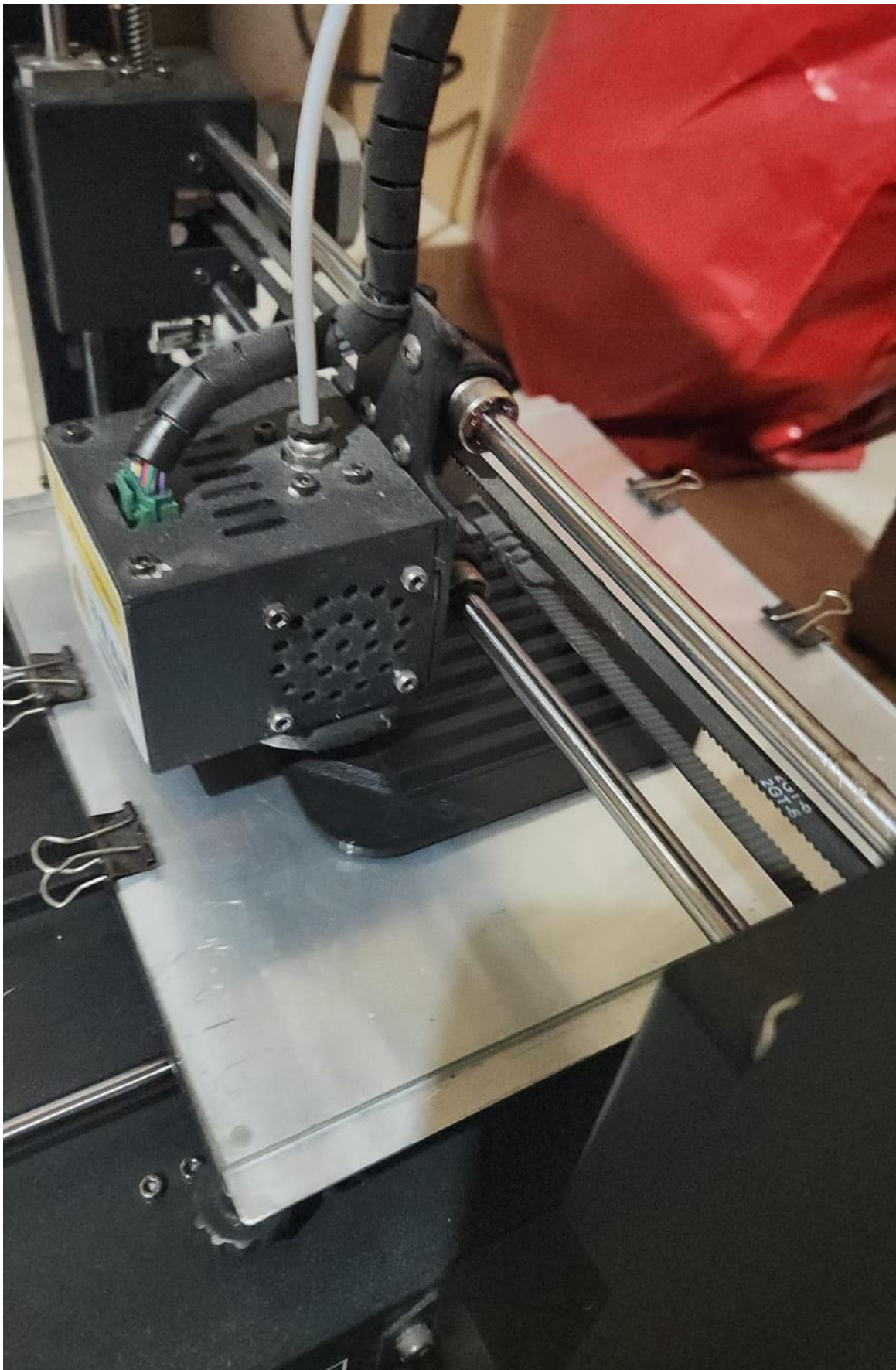


Una vez finalizado el diseño en Tinkercad, se procedió a la impresión de la carcasa utilizando una impresora 3D. Este paso implica la transformación del diseño digital en una estructura física, brindando la solidez y protección necesarias para la PCB.

Se destaca que durante el diseño se incorpora un botón de encendido/apagado del módulo ADCSPA y se utiliza este espacio para contener un regulador de tensión con salida de 5V.

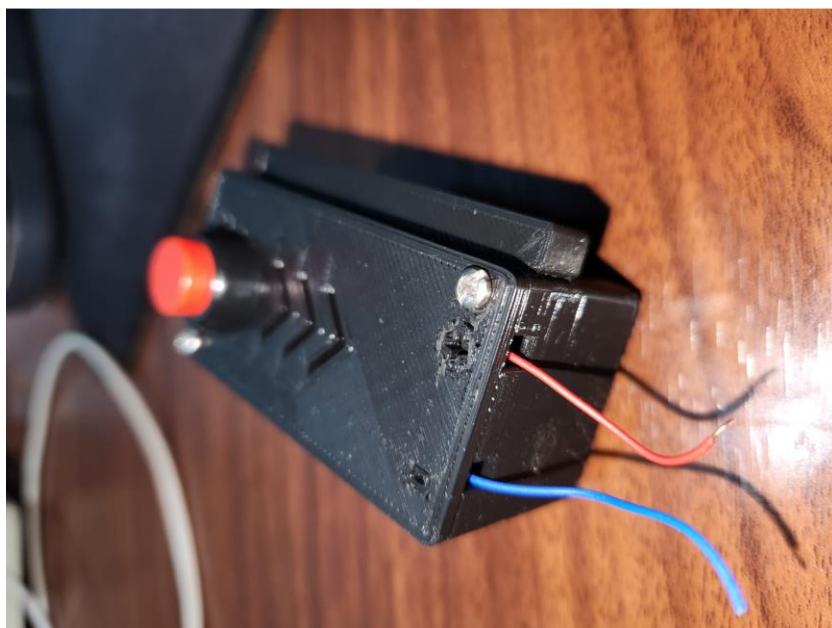
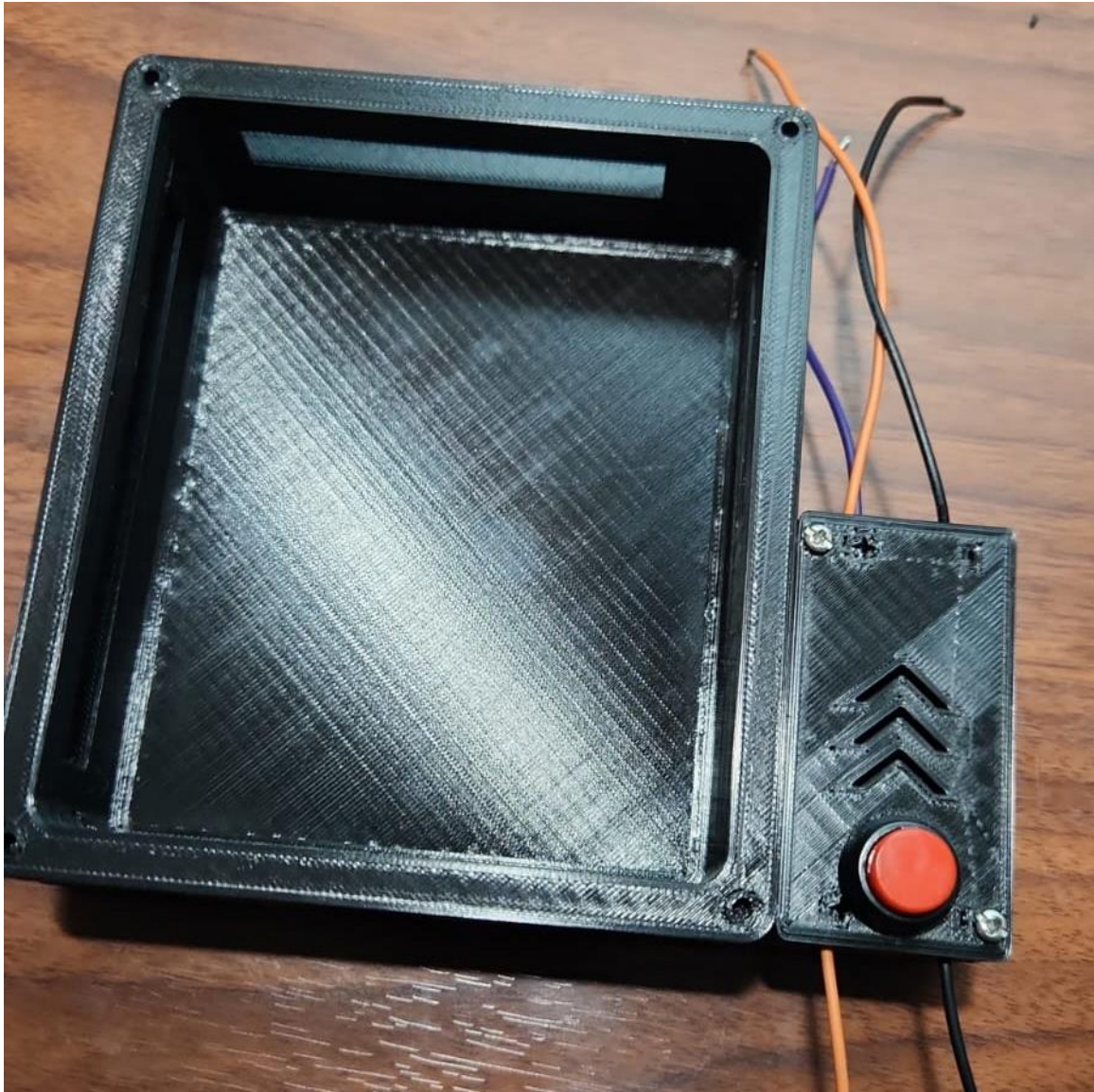
En las siguientes imágenes podemos ver el proceso de impresión:







Ahora, algunas imágenes de la carcasa finalizada:





Para finalizar, una fotografía de la PCB del módulo ADCSPA montada dentro de su carcasa:





5.1.5.2 Firmware

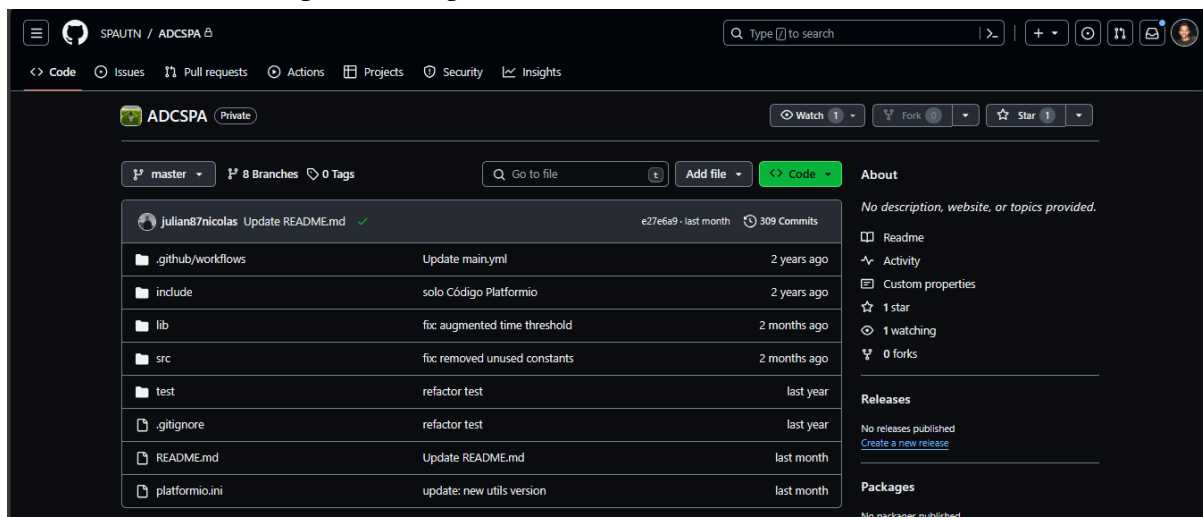
5.1.5.2.1 Enfoque y lenguaje de programación

El firmware del módulo ADCSPA, crucial para el funcionamiento del Sistema SPA, se gestó mediante un enfoque orientado a objetos en el lenguaje de programación C++. La colaboración y el control de versiones se llevaron a cabo a través de un repositorio en GitHub, bajo el nombre "ADCSPA", dentro de la organización "SPAUTN" como se desarrolló en la sección **5.1.4 Desarrollo de firmware y software**. Este enfoque estructurado facilitó el desarrollo, la colaboración y la mantenibilidad del código.

Se seleccionó Visual Studio como el entorno de desarrollo principal, aprovechando su robustez y amplias capacidades para proyectos en C++. La interfaz amigable y las herramientas de depuración avanzadas fueron fundamentales en el desarrollo y la optimización del código.

El firmware se diseñó siguiendo los principios de la programación orientada a objetos (POO), asegurando una estructura modular y una fácil expansión. Se crearon clases como "WeatherStation" para representar entidades lógicas, promoviendo la reutilización.

A continuación, una captura del repositorio ADCSPA:



El objetivo de este código es el siguiente: Debe recolectar la información de todos los sensores que componen la estación meteorológica y también el lisímetro de pesada, cuando se lo requiera. Convierte las señales analógicas en digitales, realiza un acondicionamiento según las variables involucradas y genera un JSON con toda la información. Luego este paquete de datos los transmite mediante el modulo LoRa al módulo receptor que se encuentra dentro de un edificio en el INTA. Además, se encarga del control hídrico cuando le llega una instrucción de riego.

En las siguientes páginas se describen los archivos de código más importantes para el funcionamiento del módulo ADCSPA.



5.1.5.2.2 Justificación y análisis de códigos

- WeatherStation.hpp

Este archivo define la interfaz de la clase *WeatherStation*, que representa la estación meteorológica capaz de medir y recopilar datos climáticos. Esta interfaz incluye la declaración de los métodos y miembros de la clase, así como la especificación de sus funciones principales.

La clase *WeatherStation* contiene métodos para la inicialización de los sensores. Además, se declaran métodos para configurar y obtener los valores medidos por los sensores, como la velocidad del viento, la dirección del viento, la humedad, la radiación solar, la temperatura, la presión y la humedad de la hoja.

Entre las declaraciones, se especifican métodos como *irrigateAndGetETc*, que calcula y ejecuta el riego en base a parámetros específicos, y *getPayload*, que genera un string estructurado con la información recopilada por la estación meteorológica. Además, se incluyen métodos *getter* para obtener los valores medidos y otras variables internas de la estación meteorológica.

Este archivo sirve como la interfaz que guía la implementación en el archivo **WeatherStation.cpp**. Juntos, estos archivos constituyen la implementación de la lógica necesaria para adquirir, procesar y generar datos climáticos a través de la clase *WeatherStation*, siguiendo el paradigma de la programación orientada a objetos.

El código fuente se puede analizar en el Anexo C: Firmware SSPA.

- WeatherStation.cpp

Este archivo implementa la funcionalidad definida en la clase *WeatherStation* del archivo de encabezado correspondiente **WeatherStation.hpp**. La clase representa la estación meteorológica que utiliza varios sensores para medir y recopilar datos climáticos.

El código comienza con la implementación de los constructores de la clase *WeatherStation*, inicializando variables como el tiempo de inicio y el contador del pluviómetro. Luego, se proporcionan métodos para la inicialización de los sensores, tales como el sensor BMP180 y el sensor DHT22. Además, se incluyen métodos para configurar los valores medidos por los sensores, como la velocidad del viento, la dirección del viento, la humedad, la radiación solar, la temperatura, la presión y la humedad de la hoja.

Entre los métodos implementados, se destaca la función *irrigateAndGetETc*, que calcula y ejecuta el riego en base a parámetros específicos, y la función *getPayload*, que genera un string estructurado con la información recopilada por la estación meteorológica para su posterior transmisión.

Adicionalmente, se proporcionan métodos *getter* para obtener los valores medidos y otras variables internas de la estación meteorológica.

En resumen, el archivo contiene la implementación de la lógica necesaria para adquirir, procesar y generar datos climáticos a través de la clase *WeatherStation*. Los bloques de código encapsulados en los métodos operan en los datos de la instancia a la que pertenecen, siguiendo así el paradigma de la programación orientada a objetos.

Los diagramas en bloque de algunos métodos pertenecientes a la clase *WeatherStation*, y el código fuente se pueden observar en el Anexo C: Firmware SSPA. }



- main.cpp

Este archivo constituye el punto de entrada principal del firmware. En este código, se implementa la lógica central que coordina la comunicación, la adquisición de datos climáticos y la interacción con todos los sensores representados por la clase *WeatherStation*. A continuación, se describen los elementos clave del archivo:

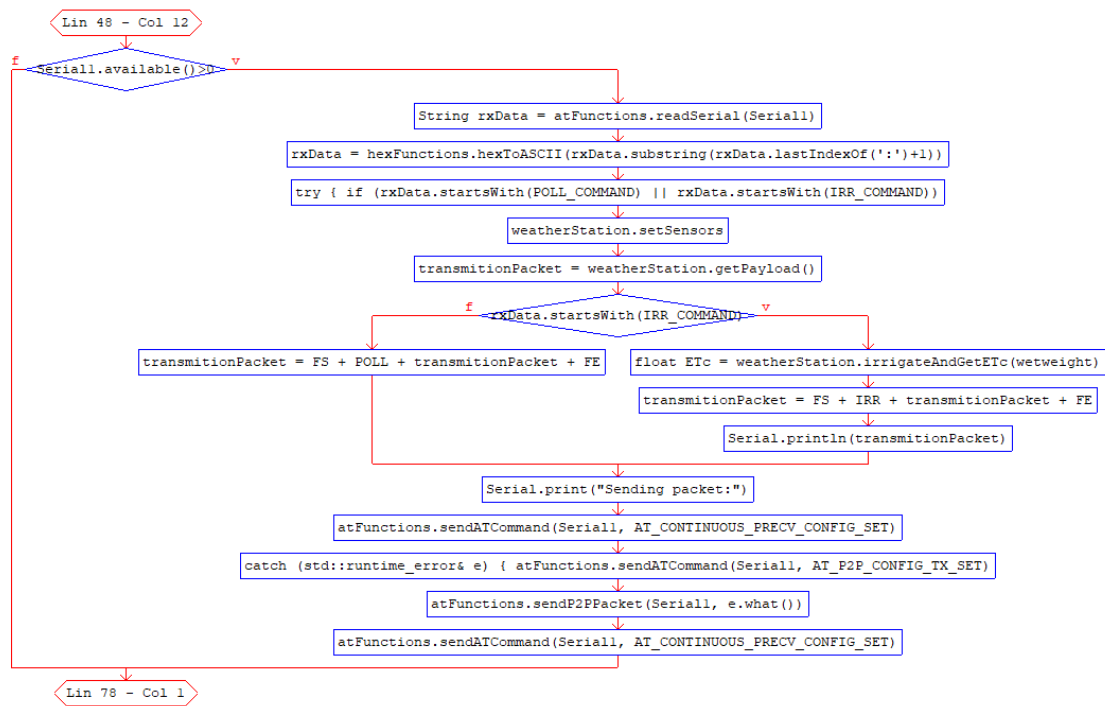
El código inicia declarando e incluyendo las bibliotecas necesarias, así como las instancias de las clases *WeatherStation*, *ATFunctions*, y *HexFunctions*.

En la función `setup()`, se realiza la configuración inicial del entorno, que incluye la inicialización de la comunicación serial, la configuración de pines, la conexión a los sensores y la configuración del módulo LoRa RAK3172 mediante comandos AT. También se establecen interrupciones ligadas al registro de precipitaciones debido al funcionamiento del pluviómetro.

La función `loop()` encapsula la lógica de control central del sistema, coordinando la adquisición de datos meteorológicos, la gestión de comandos, y la transmisión de información a través del módulo LoRa RAK3172. La función inicia comprobando si hay datos disponibles en el puerto serial. En caso afirmativo, lee el comando recibido por el módulo LoRa. Si el comando recibido comienza con los identificadores `POLL_COMMAND` o `IRR_COMMAND`, se realiza una serie de acciones para actualizar los valores del lisímetro de pesada y la estación meteorológica representada por la clase *WeatherStation*. Se obtienen lecturas de sensores, como la velocidad y dirección del viento, la humedad, la radiación solar, la temperatura, la presión, la humedad de la hoja y las precipitaciones, y se almacenan en la instancia de *WeatherStation*. Se genera un paquete de transmisión con la información recopilada utilizando el método `getPayload()` de la instancia de *WeatherStation*.

Si el comando es de riego (`IRR_COMMAND`), se procesa la información adicional contenida en el comando para determinar la cantidad de riego necesario. Se realiza el riego mediante el método `irrigateAndGetETc()`, y se ajusta el paquete de transmisión con información adicional relacionada con el riego. Se prepara el paquete de transmisión con marcas de inicio y fin (`FRAME_START` y `FRAME_END`). Se muestra el paquete antes de enviarlo, y se ejecutan comandos AT para configurar la transmisión con el módulo LoRa. Se envía el paquete mediante comandos AT y se imprime la respuesta recibida.

A continuación, se presenta un diagrama en bloque simplificado del código:



A continuación, se presentan las jerarquías de todas las clases involucradas en el firmware del modulo ADCSPA, para referencia:

WeatherStation
<ul style="list-style-type: none"> - long initTime - long int windSpeed -int windDirection - float humidity - unsigned long int radiation - long int temperature - long int pressure - int leafMoisture - long pluviometerCounter - HX711 lysimeter - SFE_BMP180 bmp180Sensor - DHT *dht
<ul style="list-style-type: none"> + String getPayload() + float getLysimeterWeight() + float getPluviometerCounter() + int getLeafMoisture() + long int getPressure() + long int getTemperature() + unsigned long int getRadiation() + float getHumidity() + int getWindDirection() + long int getWindSpeed() + float irrigateAndGetETc(float) + void setPulseCounter (long int) + void setLeafMoisture(int) + void setPresion() + void setTemperature() + void setRadiation(long int) + void setHumidity() + void setwindDirection(int) + void setWindSpeed(long int) + void init() + WeatherStation()

ATFunctions
- HexFunctions hexFunctions
+ String sendP2PPacket (Stream &,String)
+ String sendATCommand (Stream &,String)
+ String readSerial (Stream &)

HexFunctions
+ String asciiToHex (String)
+ String hexToASCII (String)

El código fuente, se puede analizar en el Anexo C: Firmware SSPA.



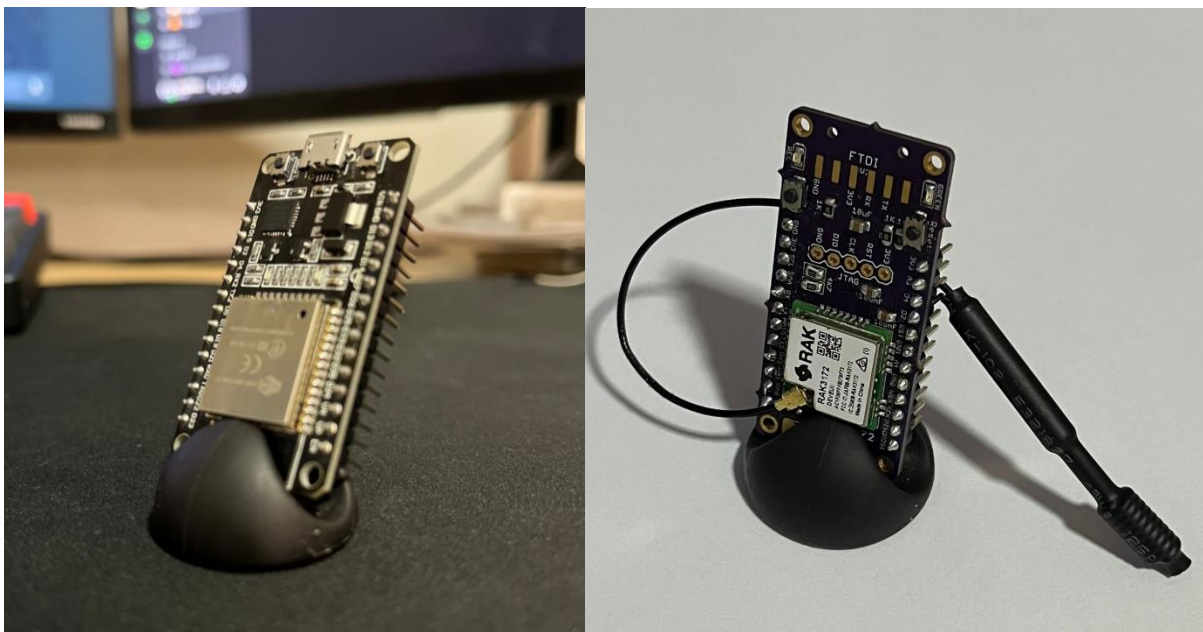
5.1.6 Módulo RXSPA

Su objetivo principal es actuar como el receptor central de datos transmitidos por el módulo ADCSPA ubicado en el campo, y establecer una comunicación con el servidor para almacenar esta información en la base de datos del Sistema SPA. En las siguientes secciones, exploraremos su diseño, desarrollo de firmware y su interacción clave con otros componentes del sistema.

5.1.6.1 Hardware

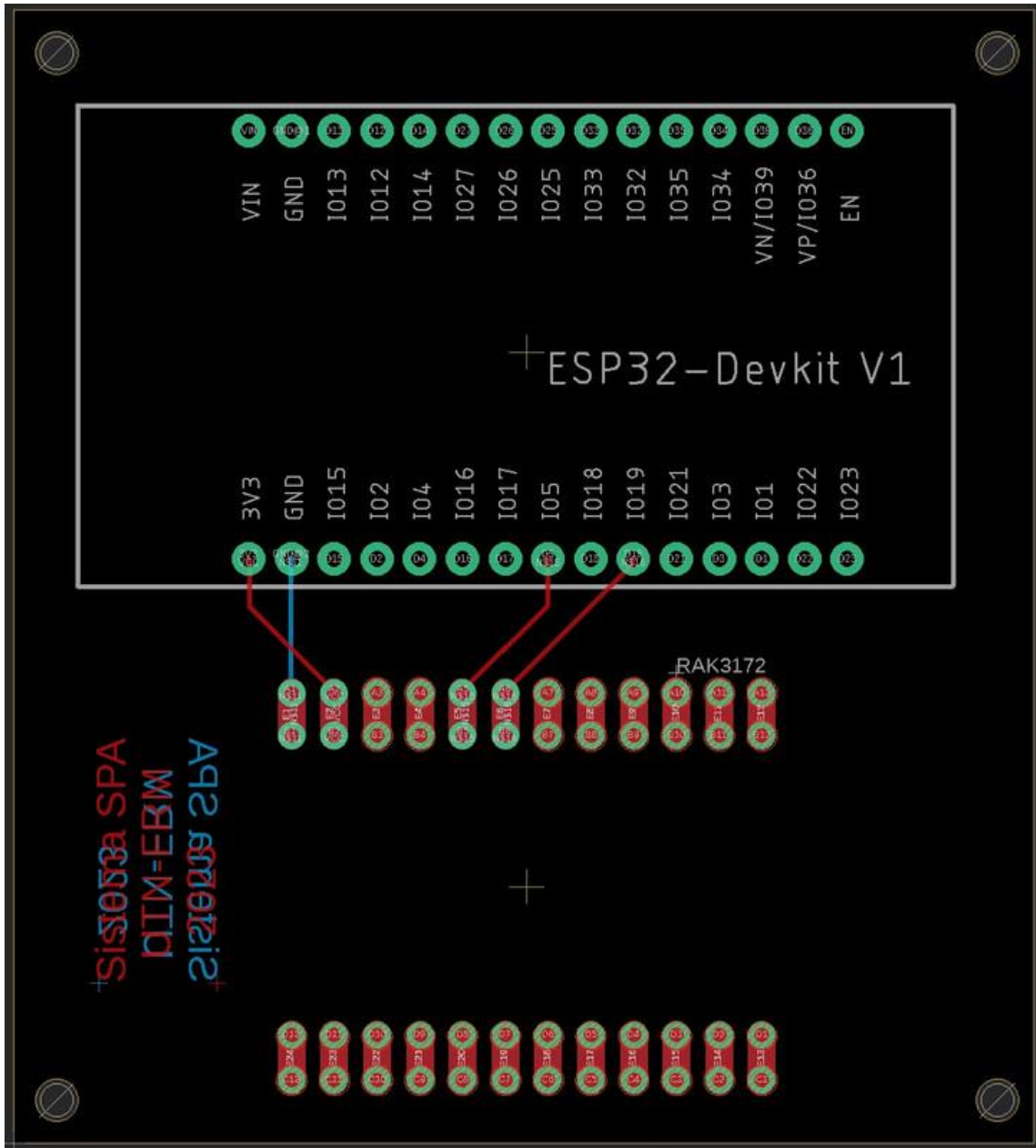
5.1.6.1.1 Diseño y desarrollo de la PCB

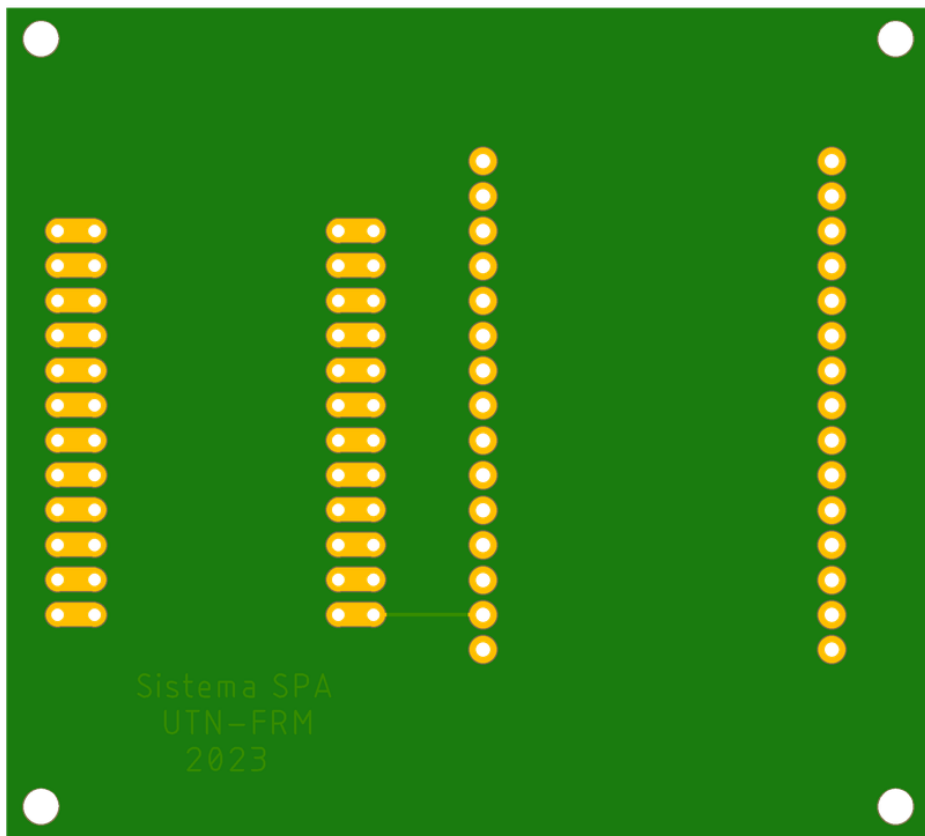
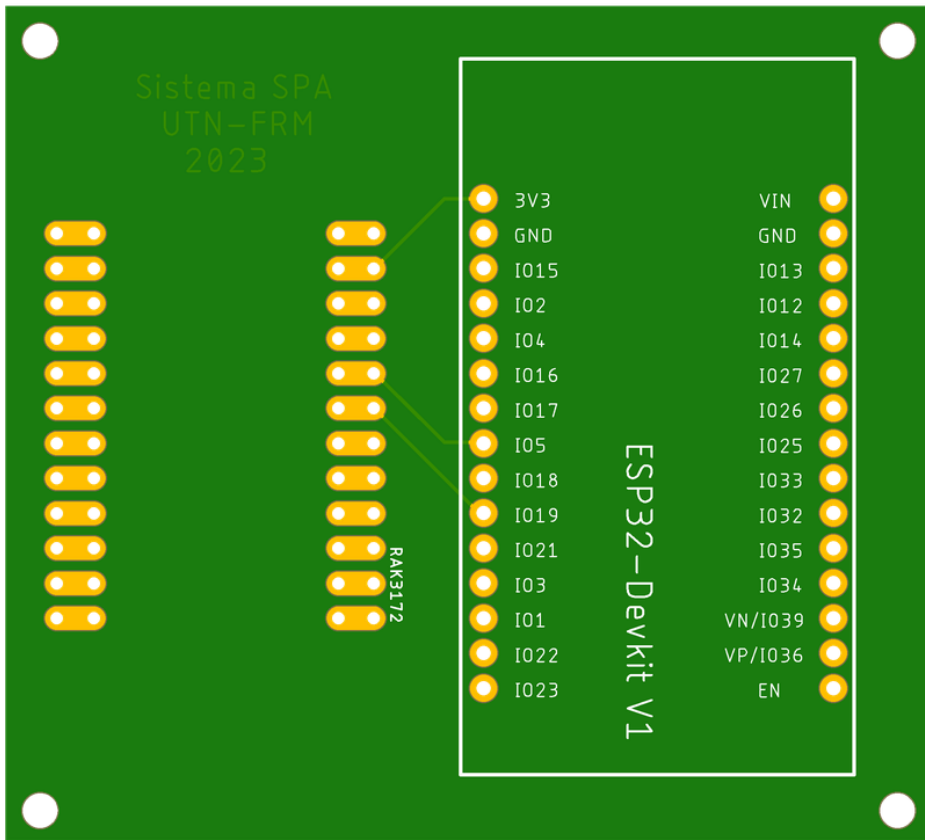
Este módulo está compuesto por un microcontrolador ESP32 y un módulo LoRa RAK3172, ambos componentes ya fueron desarrollados en secciones anteriores de este informe, por lo que no se los volverá a describir. Sin embargo, se coloca una imagen de cada uno de ellos para su referencia:



El diseño y desarrollo de la PCB del módulo RXSPA se llevó a cabo con la ayuda de Eagle, una herramienta de diseño electrónico ampliamente reconocida por su eficiencia y versatilidad. Eagle, desarrollado por Autodesk, proporciona una interfaz intuitiva y potentes capacidades de diseño que facilitan la creación de circuitos electrónicos. Eagle ofrece una interfaz de usuario amigable que facilita la creación de esquemáticos y la disposición de componentes en la PCB. Esto agiliza el proceso de diseño y permite una visualización clara de la estructura del circuito. El objetivo de la PCB para este caso es sencillo, debe permitir la conexión y comunicación entre el microcontrolador ESP32 y el módulo LoRa RAK3172. La alimentación necesaria para el funcionamiento del módulo se tomará directamente de la red eléctrica del edificio donde se encuentre, haciendo uso de una fuente de 5V y conexión micro USB permitida por el ESP32.

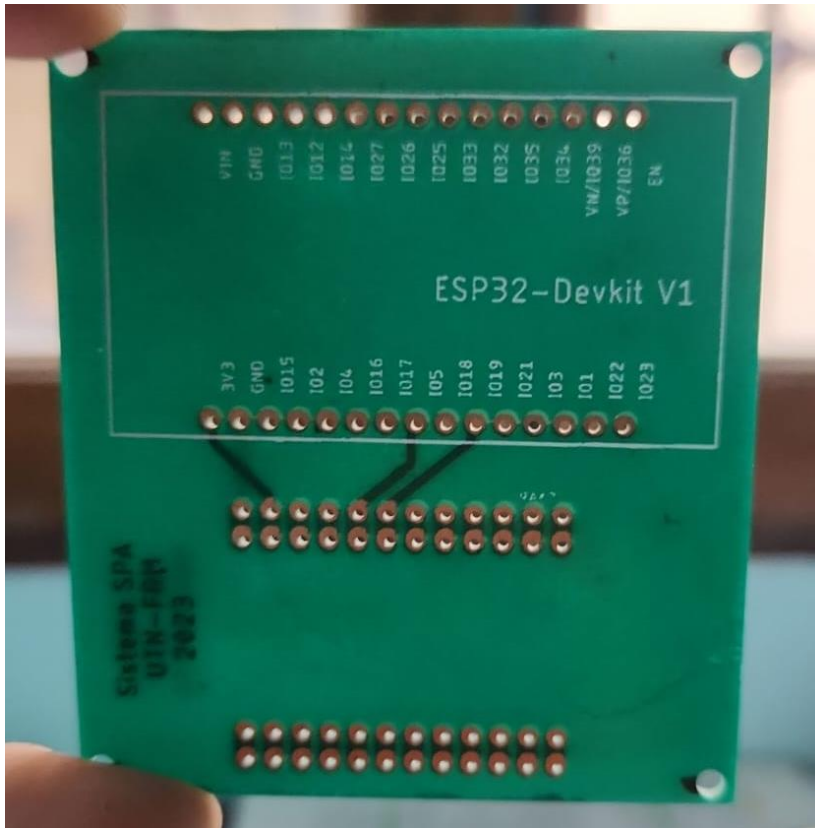
A continuación, se presentan la PCB diseñada en Eagle:



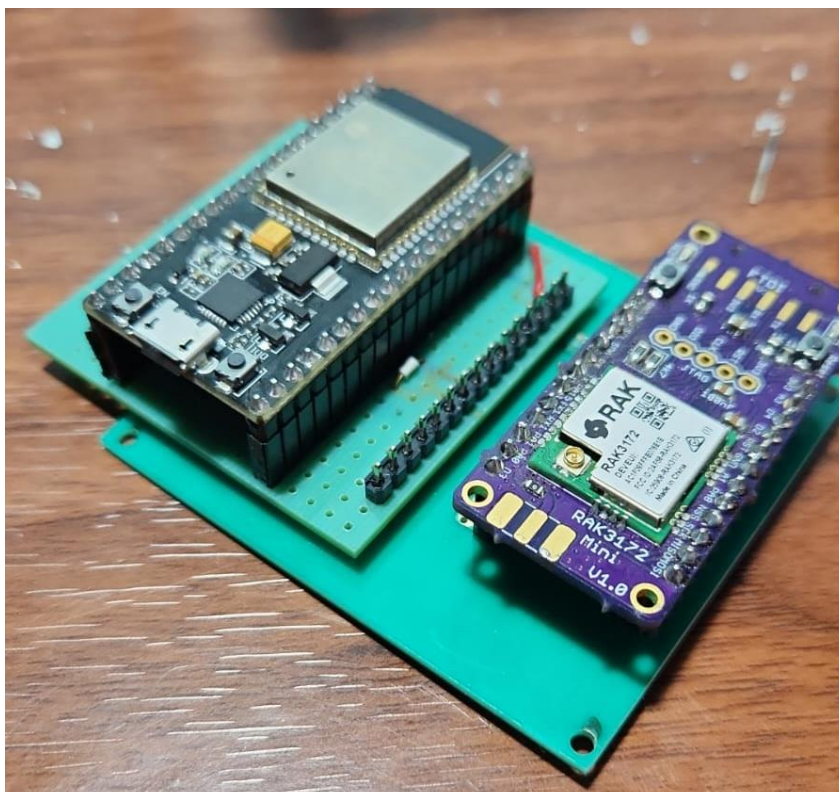




A continuación, una fotografía de la PCB recibida:



Sobre este diseño se decidió realizar unos pequeños ajustes debido a incomodidades respecto al espacio, y también para brindar la posibilidad en un futuro de retirar los componentes principales para su reutilización y/o reparación. Teniendo en cuenta lo anterior, la PCB final con los componentes montados se puede observar en la siguiente imagen:





5.1.6.1.2 Diseño e impresión carcasa 3D

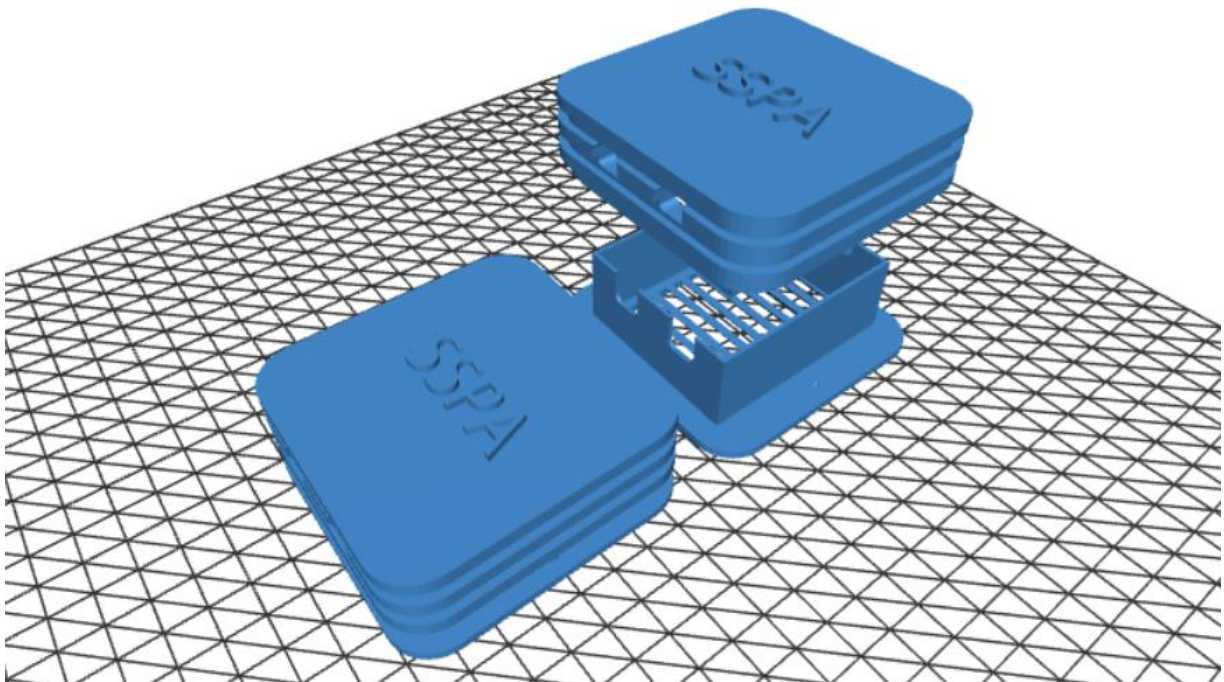
Al igual que en la sección 5.2.5.1.3, la carcasa se presenta como un componente diseñado para albergar y resguardar la PCB. Este proceso abarca desde la conceptualización y diseño hasta la materialización física mediante una impresora 3D.

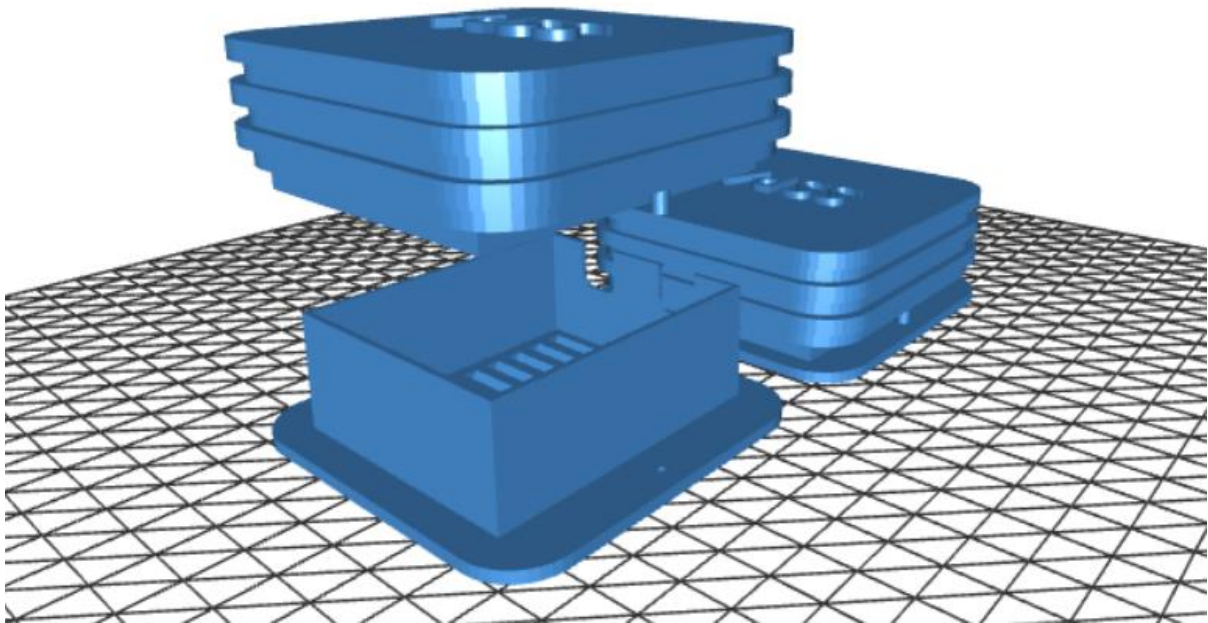
La creación del diseño de la carcasa se llevó a cabo en Tinkercad, una plataforma de modelado 3D accesible y versátil. Este proceso implica la conceptualización tridimensional de la carcasa, considerando factores como el tamaño de la PCB, disposición de componentes y accesibilidad para futuras modificaciones.

Consideraciones de diseño:

- **Espacio:** La carcasa fue diseñada para albergar la PCB, garantizando un buen ajuste.
- **Accesibilidad:** Se incorporaron aberturas estratégicas para permitir el acceso a puertos de comunicación, alimentación y salida de antena LoRa.
- **Ventilación:** Se consideraron aberturas de ventilación para evitar el sobrecalentamiento de la PCB, contribuyendo así a un funcionamiento eficiente.

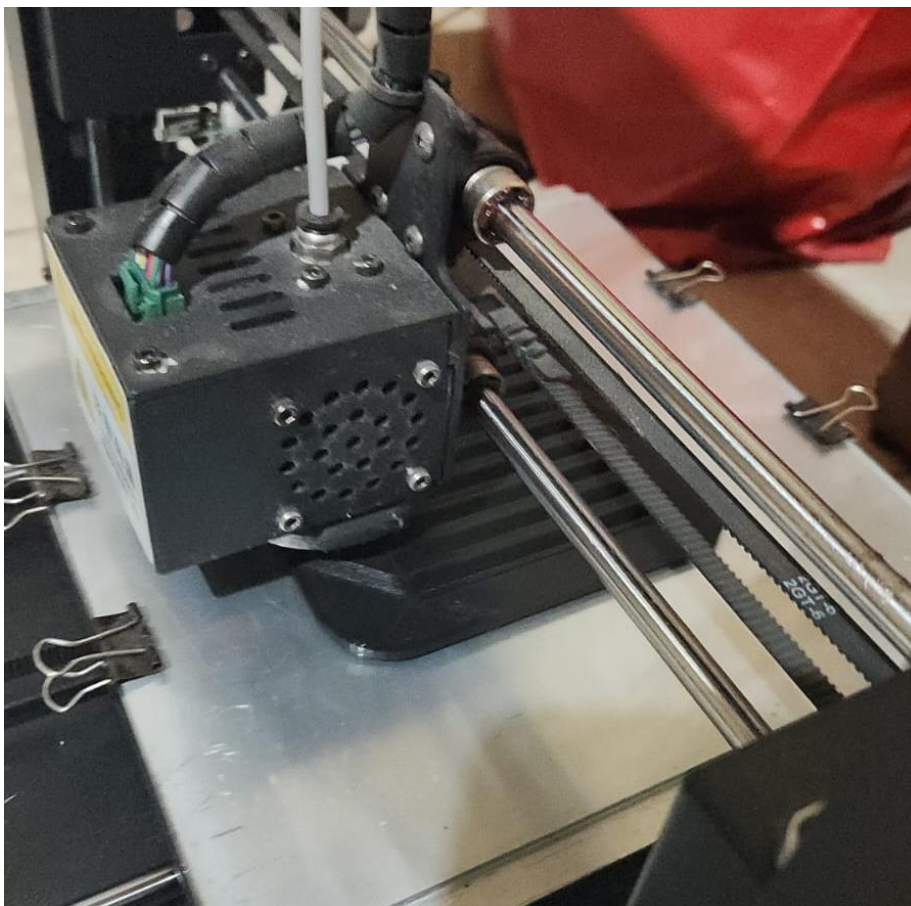
A continuación, se presentan algunas imágenes del diseño en Tinkercad:

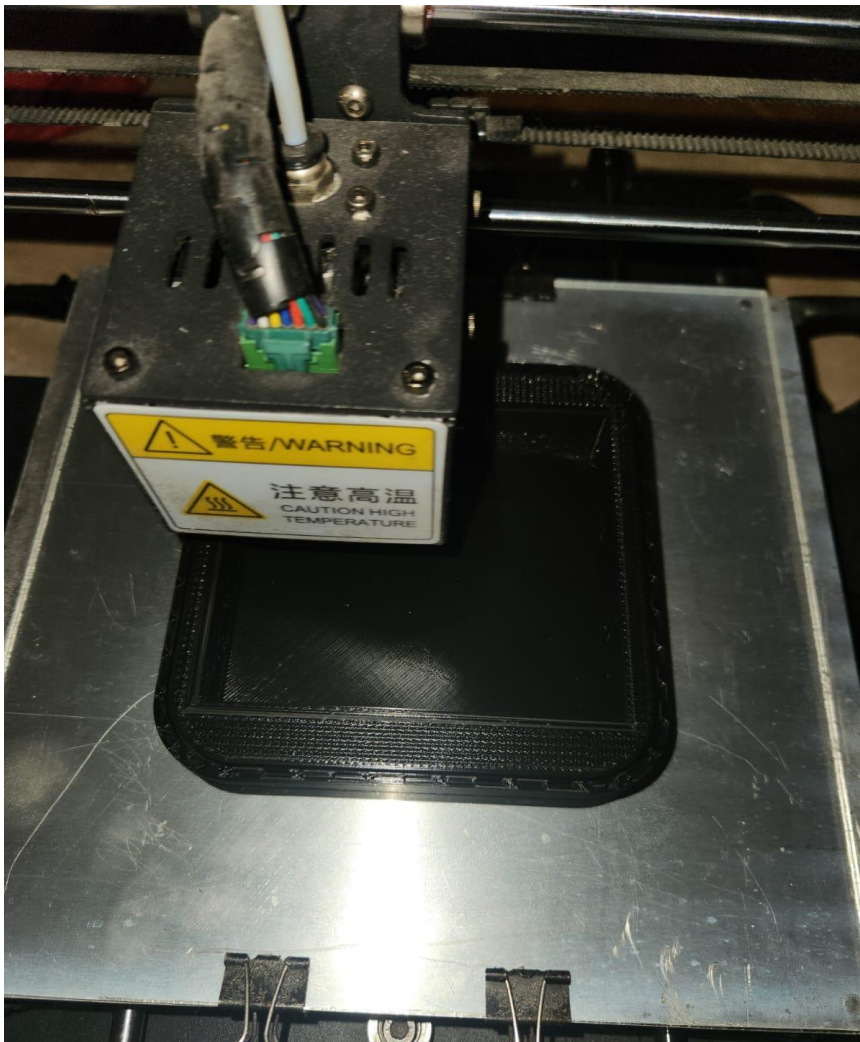




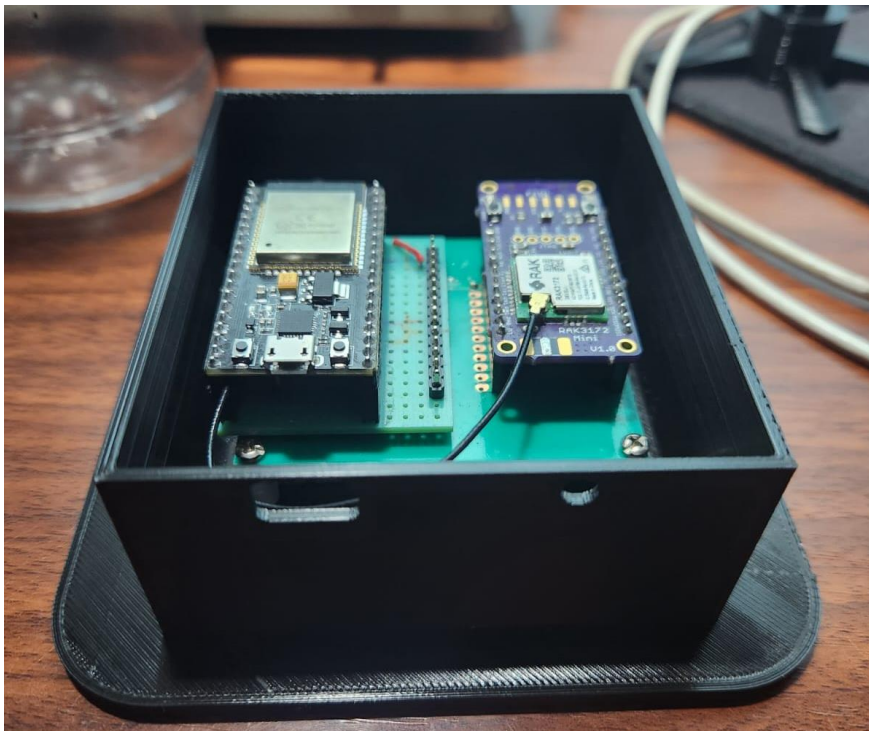
Una vez finalizado el diseño en Tinkercad, se procedió a la impresión de la carcasa utilizando una impresora 3D. Este paso implica la transformación del diseño digital en una estructura física, brindando la solidez y protección necesarias para la PCB.

En las siguientes imágenes podemos ver el proceso de impresión:





Ahora, algunas imágenes de la carcasa finalizada y con la PCB instalada:







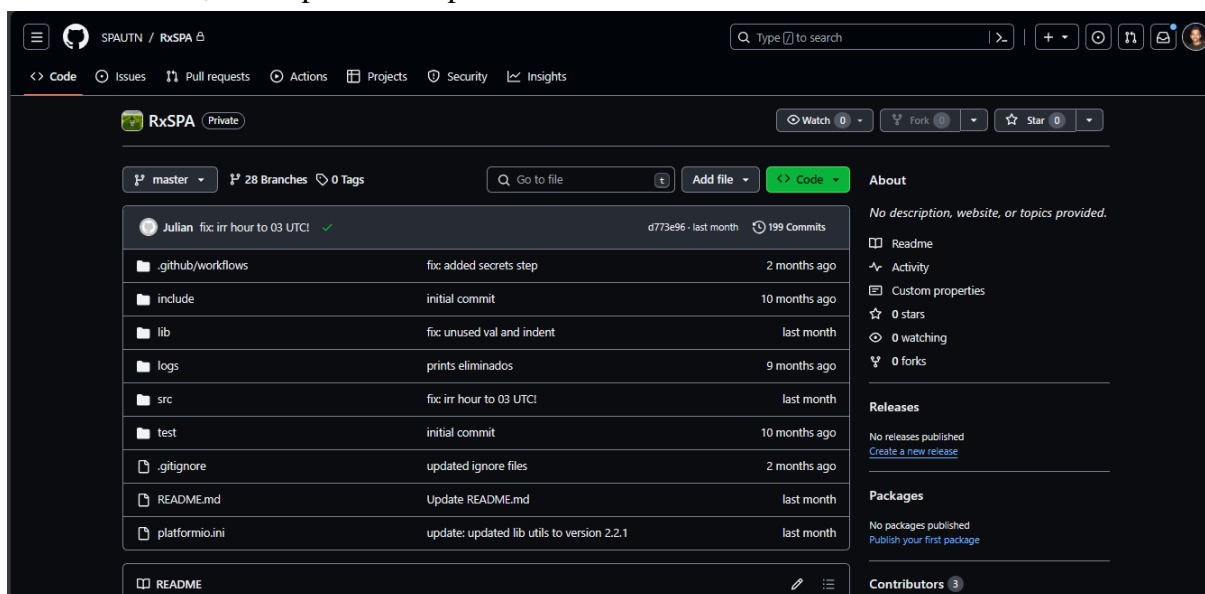
5.1.6.2 Firmware

5.1.6.2.1 Enfoque y objetivo

El firmware del módulo RXSPA, muy importante para el funcionamiento del Sistema SPA, se desarrolló mediante un enfoque orientado a objetos en el lenguaje de programación C++. La colaboración y el control de versiones se llevaron a cabo a través de un repositorio en GitHub, bajo el nombre "RxSPA", dentro de la organización "SPAUTN" como se desarrolló en la sección **5.1.4 Desarrollo de firmware y software**. Este enfoque estructurado facilitó el desarrollo, la colaboración y la mantenibilidad del código.

Se seleccionó Visual Studio como el entorno de desarrollo principal, aprovechando su robustez y amplias capacidades para proyectos en C++. La interfaz amigable y las herramientas de depuración avanzadas fueron fundamentales en el desarrollo y la optimización del código. El firmware se diseñó siguiendo los principios de la programación orientada a objetos (POO), asegurando una estructura modular y una fácil expansión.

A continuación, una captura del repositorio RxSPA:



El objetivo principal de este firmware es recibir los datos provenientes del módulo ADCSPA cuando se lo requiera y establecer una comunicación con el servidor para almacenar esta información en la base de datos. Para lograrlo el módulo RXSPA le envía comandos al módulo ADCSPA cada un cierto intervalo de tiempo, estos comandos pueden ser "POLL_COMAND" o "IRR_COMAND". Una vez enviados los comandos, el firmware aguarda la respuesta del ADCSPA, gestionando la recepción de datos a través de la comunicación serial. La información recibida se procesa y adecua para su posterior almacenamiento en la base de datos, estableciendo una comunicación con el servidor. Para garantizar la confiabilidad de la transmisión, se implementan mecanismos de reintento en caso de no recibir respuesta del ADCSPA en un tiempo determinado.

Adicionalmente, el firmware está diseñado para registrar eventos y estados significativos mediante la funcionalidad de registro proporcionada por la clase *Logger*. Esto permite tener un historial detallado de las operaciones realizadas, facilitando la depuración y el seguimiento de eventos críticos. En las siguientes páginas se describen los archivos de código más importantes para el funcionamiento del módulo RXSPA.



5.1.6.2.2 Justificación y análisis de códigos

- Logger.hpp

Este archivo define la implementación de una clase llamada *Logger* que se utiliza para realizar el registro (logging) de mensajes en el servidor a través de HTTP (es el archivo de encabezado de la clase). La clase tiene un método llamado “config” que se utiliza para configurar el logger. También tiene otro método llamado “log” que se utiliza para enviar mensajes de registro al servidor. Además del método “log”, se proporcionan métodos específicos para distintos niveles de registro (error, info, y debug). Cada método de nivel de registro toma los mismos parámetros que el método log, pero el nivel correspondiente ya está predefinido.

La clase contiene miembros privados para almacenar la información de conexión como el host, el nombre de usuario, la contraseña, y un objeto *HTTPClient* para realizar las solicitudes HTTP.

El código fuente se puede analizar en el Anexo C: Firmware SSPA.

- Logger.cpp

En este archivo, se proporciona la implementación (definición) detallada de cada método de la clase *Logger* que fue declarada en *Logger.hpp*. Incluye el archivo de encabezado “*Logger.hpp*” para que el compilador tenga acceso a la declaración de la clase y sus miembros.

El método “config” asigna los valores de los parámetros proporcionados a los miembros correspondientes de la clase (logHost, user, pass, y attempts).

El método “log” construye un mensaje de registro en formato JSON que incluye información como el código HTTP, el mensaje, el nivel y la fuente del mensaje. Utiliza un bucle para intentar enviar el mensaje al servidor. Dentro del bucle, se configuran los encabezados HTTP, la autorización con el nombre de usuario y la contraseña, y se realiza la solicitud POST al servidor. Si la respuesta del servidor no es exitosa (código HTTP diferente de 201), se muestra un mensaje de error y se intenta nuevamente, hasta que se alcance el número máximo de intentos o se obtenga una respuesta exitosa. El código de respuesta HTTP final se devuelve.

Los métodos de niveles de registro (error, info, debug), simplemente llaman al método “log” con el nivel correspondiente predefinido (ERROR_LEVEL, INFO_LEVEL, DEBUG_LEVEL).

En resumen, el archivo *Logger.cpp* implementa las funciones necesarias para realizar el registro de mensajes en el servidor utilizando HTTP. Configura la conexión HTTP con la información proporcionada en el método “config” y proporciona funciones específicas para diferentes niveles de registro (error, info, debug). El método “log” se encarga del envío real del mensaje de registro y maneja los intentos en caso de fallos.

El diagrama en bloque del método “log” de la clase *Logger*, y el código fuente se pueden observar en el Anexo C: Firmware SSPA.



- [RestCall.hpp](#)

Este archivo define la estructura de la clase *RestCall*, que se encarga de facilitar las operaciones de llamadas REST hacia el servidor. La clase proporciona métodos para configurar la conexión, enviar comandos específicos al servidor, y obtener respuestas. Para su funcionamiento, la clase utiliza la biblioteca *HTTPClient.h* para realizar solicitudes HTTP y *ArduinoJson.h* para manipulación de datos en formato JSON. Además, se incluye la clase *Logger* para gestionar los registros y niveles de depuración.

REST (Representational State Transfer) es un enfoque para diseñar servicios web de manera sencilla y eficiente. Se basa en principios como el uso de recursos identificables por URLs y la realización de operaciones estándar (GET, POST, PUT, DELETE) sobre estos recursos.

En la clase *RestCall*, el uso de REST implica enviar solicitudes HTTP para realizar acciones específicas, como obtener información (*getWeight*), enviar datos (*sendFrameData*), o verificar la disponibilidad del servidor (*ping*). Esto se hace de manera estructurada y sigue principios que promueven la simplicidad y la independencia entre el cliente y el servidor.

El método “*config*” se encarga de configurar la instancia de la clase con la URL del host, el nombre de usuario, y la contraseña necesarios para autenticarse en el servidor. Esto establece la base para las operaciones subsiguientes.

El método “*getWeight*” realiza una solicitud al servidor con un comando específico y devuelve la respuesta correspondiente. Esta función está diseñada para obtener valores de peso desde el backend.

El método “*sendFrameData*” permite enviar datos de un marco al servidor, especificando la tabla de destino y el número de intentos en caso de falla en el envío.

El método “*ping*” envía un ping al servidor para verificar su disponibilidad. El número de intentos permite controlar la tolerancia a fallos en la conexión.

Adicionalmente, la clase proporciona métodos para configurar y obtener el nivel de depuración, así como establecer y recuperar el código de respuesta de la última solicitud HTTP realizada.

El código fuente se puede analizar en el Anexo C: Firmware SSPA.

- [RestCall.cpp](#)

Este archivo contiene la implementación real de las funciones declaradas en *RestCall.hpp*. Aquí es donde se detallan las acciones específicas de cada método, se maneja la lógica interna, y se utilizan las bibliotecas y herramientas necesarias para realizar las operaciones deseadas.

La clase *RestCall* encapsula la lógica para interactuar con el servidor mediante REST, ofreciendo una interfaz simple para realizar operaciones como obtención de datos, envío de información y verificación del estado del servidor. La integración con la clase *Logger* permite mantener un registro estructurado de las operaciones y facilita el seguimiento de posibles problemas en la comunicación con el servidor.

El método *config* asigna las variables proporcionadas a los miembros correspondientes de la clase, estableciendo así la información necesaria para las solicitudes HTTP.

Los métodos *setDebugLevel* y *getDebugLevel* permiten configurar y obtener el nivel de depuración de la clase, respectivamente.

Los métodos *setResponseCode* y *getResponseCode* se utilizan para establecer y obtener el código de respuesta de la última solicitud HTTP realizada.



El método `sendFrameData` envía un marco de datos al servidor, permitiendo reintentos en caso de respuestas no exitosas. Gestiona el registro de eventos y establece el código de respuesta y nivel de depuración correspondientes.

El método `getWeight` realiza una solicitud al servidor para obtener valores de peso, con reintentos en caso de respuestas no exitosas. Analiza la respuesta JSON del servidor para extraer el valor correspondiente.

El método `ping` envía un ping al servidor para verificar su disponibilidad, con reintentos en caso de respuestas no exitosas.

Los diagramas en bloque de algunos métodos de la clase *RestCall*, y el código fuente correspondiente se puede analizar en el Anexo C: Firmware SSPA.

- [main.cpp](#)

Este archivo es el punto de entrada principal del programa y contiene la lógica principal de la aplicación. Orquesta la ejecución del programa, gestionando la configuración inicial, las llamadas a funciones específicas en momentos predeterminados y las acciones periódicas para mantener la comunicación con el servidor y realizar operaciones programadas.

Inicia la ejecución definiendo la configuración inicial del programa, que incluye la inicialización de la comunicación serial y la conexión WiFi a través de `WiFiManager`. Además, establece instancias, como las clases *Logger* y *RestCall*, para gestionar registros y realizar llamadas a la API mediante REST. Dentro de su estructura, se definen constantes que representan URLs, comandos específicos y otras configuraciones relevantes. Se inician instancias de clases útiles, como *Timestamp*, que proporciona información sobre la fecha y hora actuales. Organiza la ejecución del programa mediante la configuración de alarmas. Se definen dos alarmas, `pollAlarm` e `irrAlarm`, que invocan la función `sendPollCommand` con comandos específicos.

La función `sendPollCommand` gestiona la comunicación entre el módulo RXSPA y el módulo ADCSPA, como así también la comunicación con el servidor del Sistema SPA. La función comienza enviando un comando al módulo ADCSPA mediante el uso del método `sendP2PPacket(Serial2, pollCommand)` de la clase `atFunctions`. Este comando específico se espera que genere una respuesta por parte del módulo ADCSPA, y la función configura el módulo LoRa para recibir dicha respuesta. Posteriormente, la función entra en un bucle de espera, donde monitorea la llegada de datos al puerto serie. Si se recibe una respuesta, se interpreta y se almacena en la variable `frame`. Además, se registra este marco de datos mediante la instancia de la clase *Logger* para llevar un registro detallado de la operación. La función maneja la lógica de reintentos en caso de no recibir una respuesta exitosa dentro del tiempo especificado. Si no se recibe un marco de datos o si se supera el número máximo de intentos, la función realiza una serie de acciones, como registrar un error, reconfigurar la comunicación serial y volver a enviar el comando.

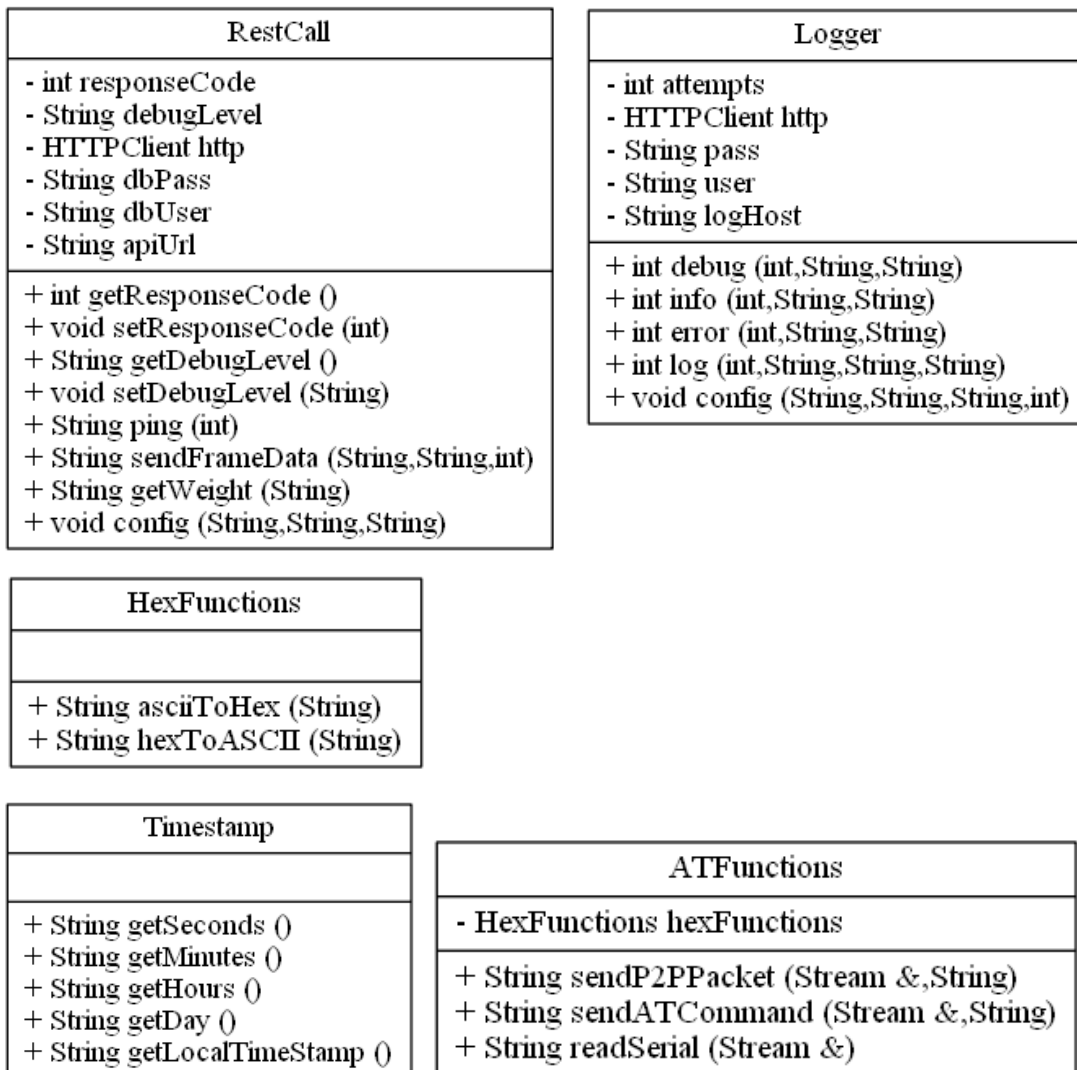
Si se recibe un marco de datos sin errores, la función utiliza la instancia de la clase *RestCall* para enviar este marco al servidor mediante `restCall.sendFrameData(frame, STATION_TABLE, 3)`. La respuesta de esta operación se registra nuevamente utilizando la instancia de la clase *Logger*. En el caso específico de comandos relacionados con la irrigación (`IRR_COMMAND`), la función realiza una operación adicional. Divide el marco de datos en tres partes correspondientes a diferentes tablas en la base de datos (`STATION_TABLE`, `ETC_TABLE`, y `WET_WEIGHT_TABLE`). Cada parte se envía por separado al servidor



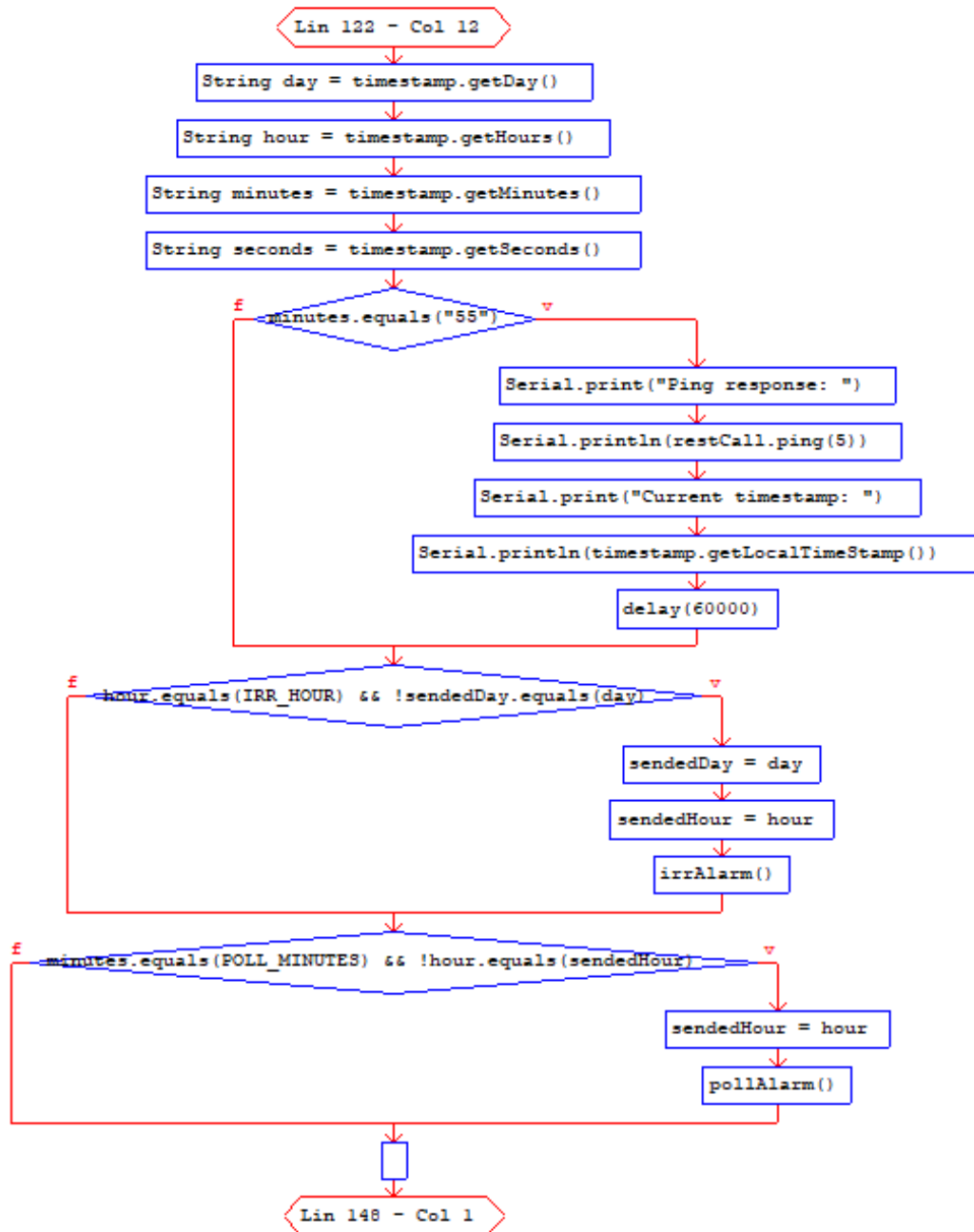
mediante llamadas a `restCall.sendFrameData`, y las respuestas se registran con la instancia de `Logger`.

En el bucle principal, se gestiona la lógica continua del programa. Se obtiene la fecha y hora actuales, y se verifica si es necesario ejecutar las funciones asociadas a las alarmas. Además, se realiza un ping al servidor y se imprime la marca de tiempo local cuando los minutos son igual a 55, proporcionando una comprobación periódica del estado del sistema.

A continuación, se presentan las jerarquías de todas las clases involucradas en el firmware del modulo RXPSA, para referencia:



A continuación, el diagrama de flujo simplificado del bucle principal:

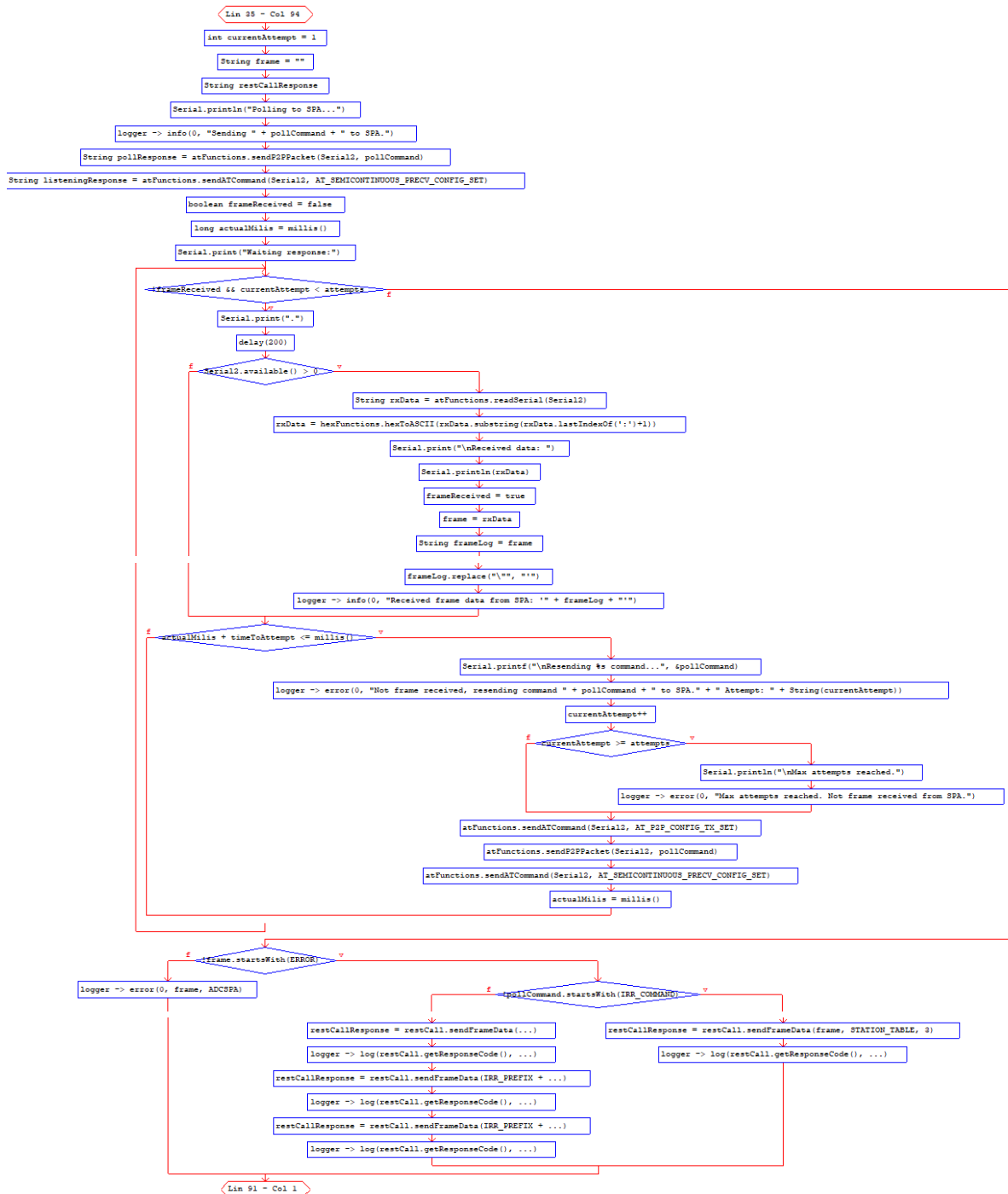


Tanto la función pollAlarm() como la función irrAlarm(), utilizan la función sendPollCommand, por lo que no tiene sentido hacer sus diagramas en bloque.

Pero la función sendPollCommand si requiere de un diagrama en bloque, por lo que se lo representa a continuación su diagrama simplificado:



void sendPollCommand(String pollCommand, Logger *logger, int timeToAttempt, int attempts = 5)



El código fuente del archivo main.cpp se puede observar en el Anexo C: Firmware SSPA.



5.1.7 Base de datos SPADB

5.1.7.1 Herramientas utilizadas

En esta sección se abordan las herramientas fundamentales utilizadas para alojar, organizar y permitir el acceso a la información recopilada por el sistema. En este contexto, se destacan dos herramientas clave que desempeñan roles complementarios en la administración de la base de datos: FLO y DataGrip.

- FLO

FLO es una plataforma de ingeniería de backend que se destaca por su capacidad para permitir el despliegue rápido y eficiente de aplicaciones de backend y bases de datos. Esta plataforma innovadora está diseñada para simplificar la infraestructura en la nube, eliminando la necesidad de entender Kubernetes o de configurar recursos en la nube.



Una de las características más notables de FLO es su capacidad de escalado automático. Esta característica permite que la plataforma se ajuste automáticamente con el tráfico del sitio web, lo que garantiza un rendimiento óptimo incluso durante los períodos de mayor demanda.

FLO también se distingue por su alta fiabilidad, ofreciendo un tiempo de actividad del 99.9%. Esta confiabilidad asegura que las aplicaciones y bases de datos desplegadas en FLO estén siempre disponibles cuando los usuarios las necesiten.

La integración con GitHub es otra característica clave de FLO. Esta integración permite desplegar automáticamente desde tu repositorio de GitHub, lo que facilita la implementación y actualización de aplicaciones y bases de datos.

FLO ofrece una notable flexibilidad en términos de gestión de bases de datos, lo que la convierte en una opción atractiva para los proyectos de ingeniería que requieren un manejo eficiente y efectivo de los datos.

La plataforma FLO permite a los usuarios elegir entre bases de datos compartidas o dedicadas. Las bases de datos compartidas son una opción rentable para los proyectos que no requieren una gran cantidad de recursos de base de datos. Por otro lado, las bases de datos dedicadas ofrecen un mayor rendimiento y son ideales para los proyectos que manejan grandes volúmenes de datos o requieren un alto grado de aislamiento de datos.

Además, FLO ofrece la opción de acceso privado o público a las bases de datos. El acceso privado garantiza que sólo las aplicaciones desplegadas en la misma cuenta de FLO pueden acceder a la base de datos, lo que proporciona un alto nivel de seguridad. El acceso público, por otro lado, permite el acceso desde cualquier lugar, lo que puede ser útil para las pruebas o para las aplicaciones que requieren un acceso más amplio.

FLO también proporciona métricas en tiempo real, lo que permite a los usuarios monitorizar su uso de la base de datos con registros detallados y alertas oportunas. Esta característica es



especialmente útil para mantener un control estricto sobre el rendimiento de la base de datos y el uso de los recursos.

- DataGrip

DataGrip, desarrollado por JetBrains, es un entorno de desarrollo integrado (IDE) multiplataforma para bases de datos y SQL. Este IDE está diseñado para satisfacer las necesidades específicas de los desarrolladores.



DataGrip es un entorno de bases de datos multi-motor. Si el sistema de gestión de bases de datos (DBMS) tiene un controlador JDBC, es posible conectarse a él a través de DataGrip. Proporciona introspección de bases de datos y varios instrumentos para crear y modificar objetos para los motores soportados. Las bases de datos son todo acerca de los datos. DataGrip permite manejar todas las manipulaciones de datos: editar, buscar, importar y exportar. El potente editor de datos permite añadir, eliminar, editar y clonar filas de datos. Es posible navegar a través de los datos por claves foráneas y utilizar la búsqueda de texto para encontrar cualquier cosa en los datos mostrados en el editor de datos.

Al igual que cualquier IDE, DataGrip proporciona finalización inteligente de código, inspecciones de código, resaltado de errores sobre la marcha, soluciones rápidas y capacidades de refactorización. Además, permite ejecutar consultas en diferentes modos y proporciona un historial local que mantiene un registro de toda la actividad.

Es posible saltar a cualquier tabla, vista o procedimiento por su nombre a través de la acción correspondiente, o directamente desde sus usos en el código SQL. Otorga una visión extendida de cómo funcionan las consultas y del comportamiento del motor de la base de datos. DataGrip proporciona soporte unificado para todos los principales sistemas de control de versiones: Git, SVN, Mercurial y muchos otros.

FL0 al proveer una máquina virtual en la nube, establece un entorno robusto para SPADB. Esta infraestructura escalable se integra con DataGrip, permitiendo una conexión directa para el diseño, implementación y manipulación de la base de datos. La interacción entre estas dos herramientas posibilita un flujo de trabajo continuo, desde la concepción de la estructura de la base de datos hasta su administración y consulta. La conexión segura entre DataGrip y FL0, respaldada por las políticas de seguridad de la plataforma, garantiza la protección de datos sensibles y la integridad de la base de datos.

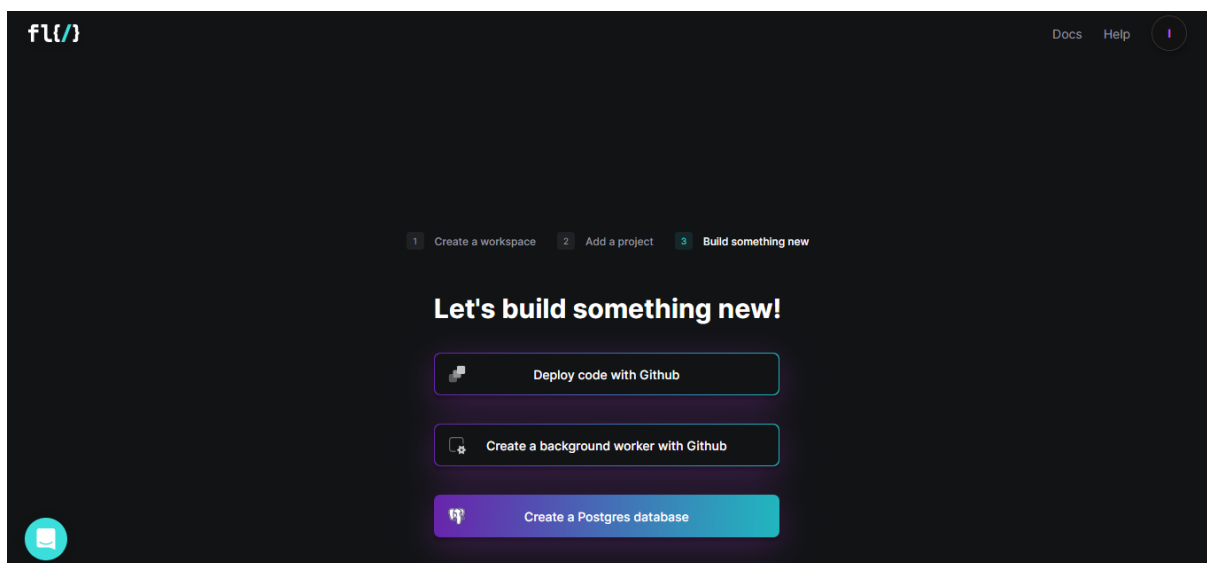
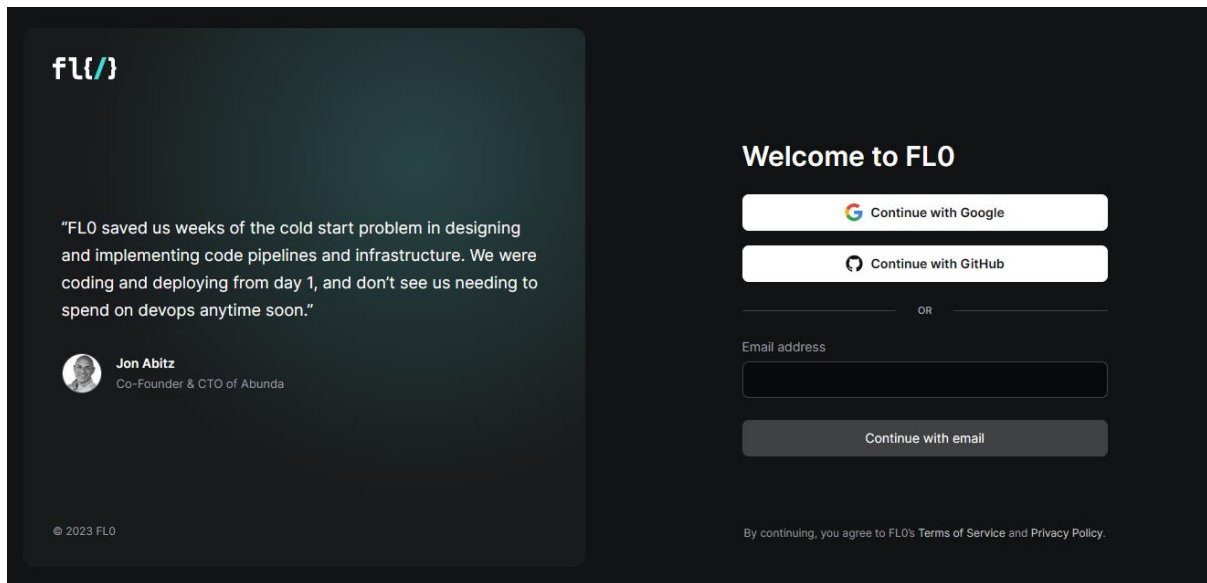
DataGrip facilita la creación de esquemas y tablas mediante su interfaz intuitiva, mientras que FL0 agiliza el proceso de aprovisionamiento de recursos en la nube. La gestión de usuarios, asignación de permisos y ejecución de consultas SQL se realizan a través de DataGrip, aprovechando la conectividad directa con FL0.

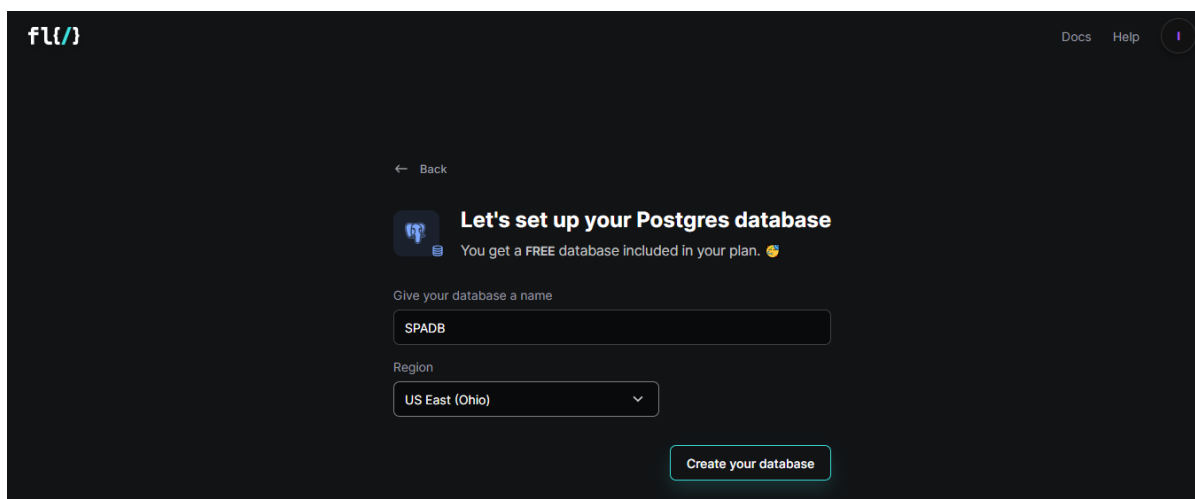


5.1.7.2 Desarrollo e implementación

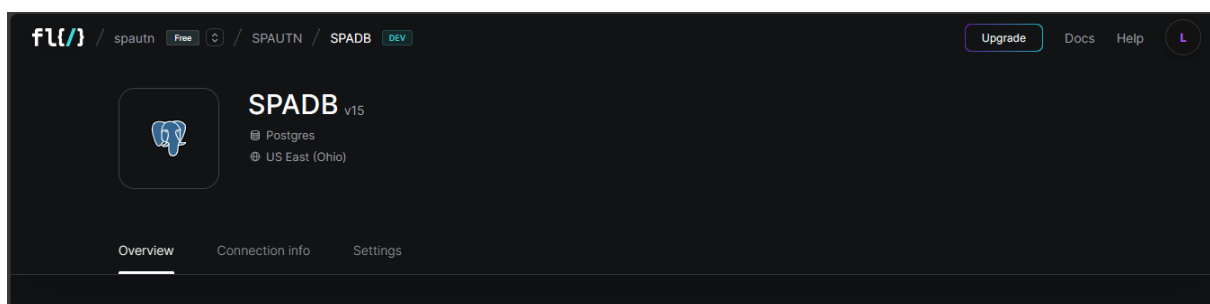
Esta sección detalla el proceso de desarrollo e implementación de la base de datos SPADB, destacando la utilización de FLO como plataforma de alojamiento en la nube y DataGrip como la interfaz de desarrollo de base de datos. A continuación, se guiará a través de los pasos clave, desde la creación de la cuenta hasta la configuración de esquemas y tablas.

El punto de partida fue la creación de una cuenta en la plataforma [FLO](#). Este proceso inicial proporcionó acceso a un conjunto de servicios, entre los cuales se optó por la creación de la base de datos. La elección de este servicio específico permitió no solo el alojamiento de datos, sino también el aprovechamiento de funcionalidades adicionales para una gestión integral.

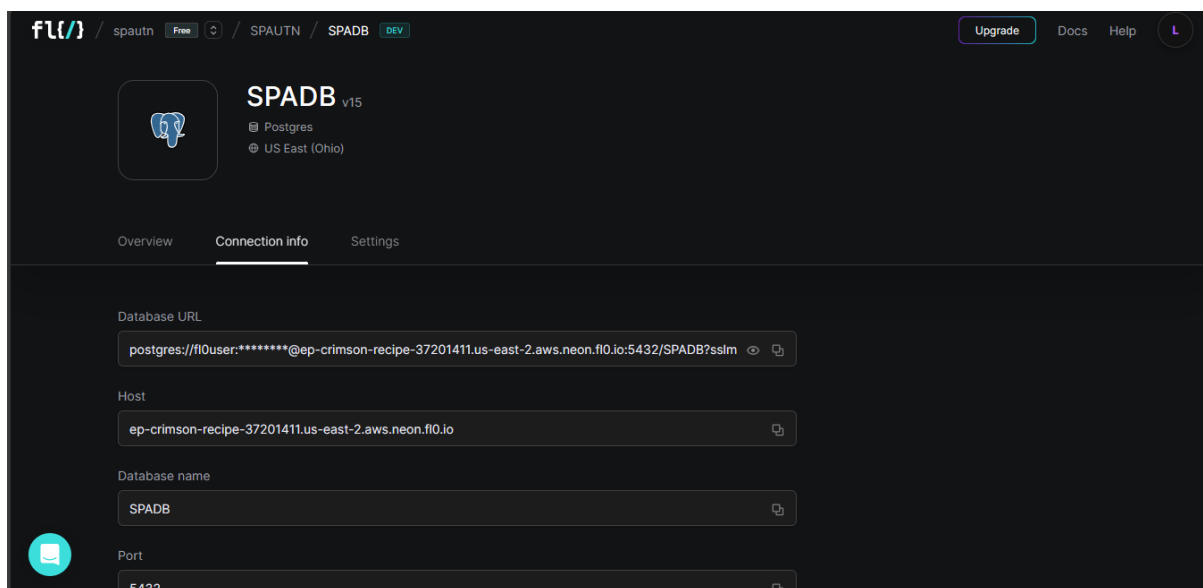




Tras seleccionar la opción de base de datos, FL0 facilitó la creación y aprovisionamiento de una máquina virtual en la nube. Este entorno dedicado se configuró para albergar la base de datos SPADB, proporcionando un escenario flexible y escalable para el desarrollo y almacenamiento de datos.

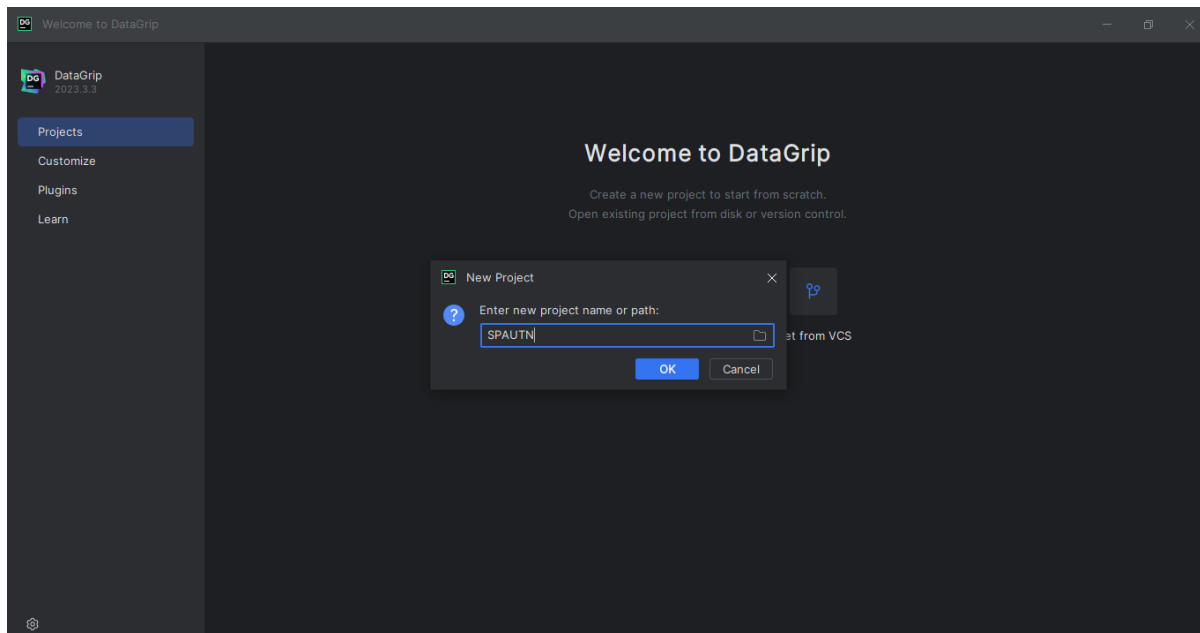


La conexión a la base de datos alojada en FL0 se estableció mediante DataGrip, una IDE de base de datos. Para ello fueron necesarios los detalles de host y credenciales proporcionadas por FL0, que se pueden consultar en la pestaña “conexion info”:

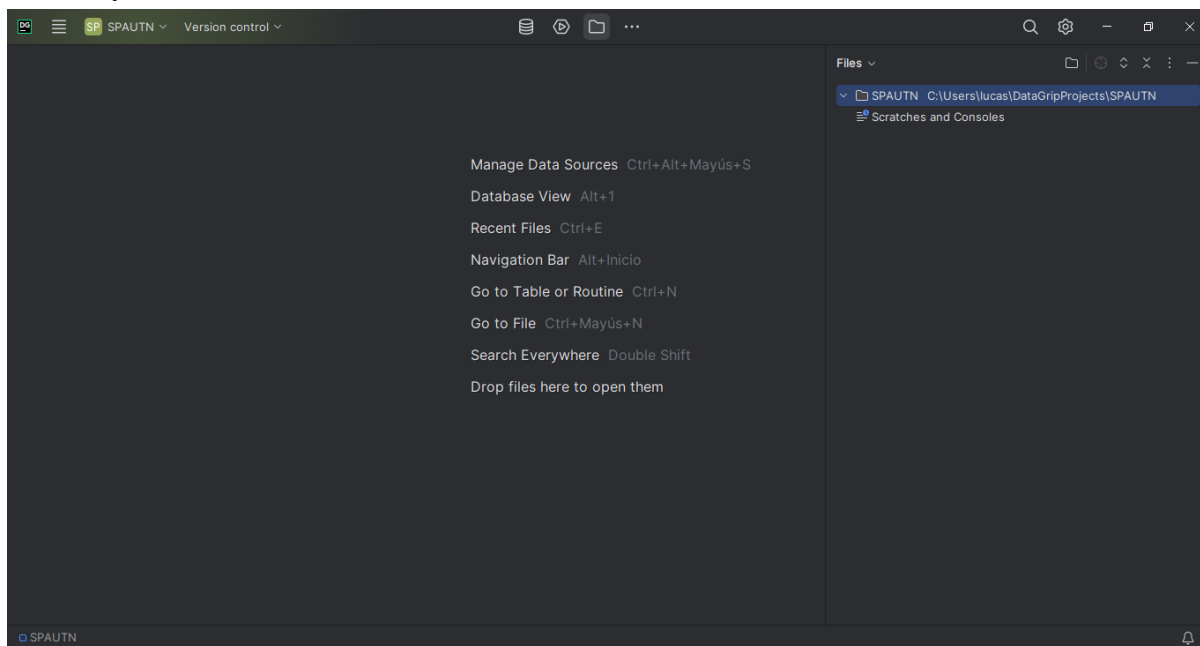




DataGrip muestra la pantalla de bienvenida cuando no se abre ningún proyecto. Por ejemplo, cuando se ejecuta DataGrip por primera vez o cuando se cierra el único proyecto abierto. Desde esta pantalla, se puede crear un nuevo proyecto, abrir un proyecto existente o clonar archivos desde un sistema de control de versiones (VCS). En este caso, se creó el proyecto SPAUTN como se observa en la imagen debajo:



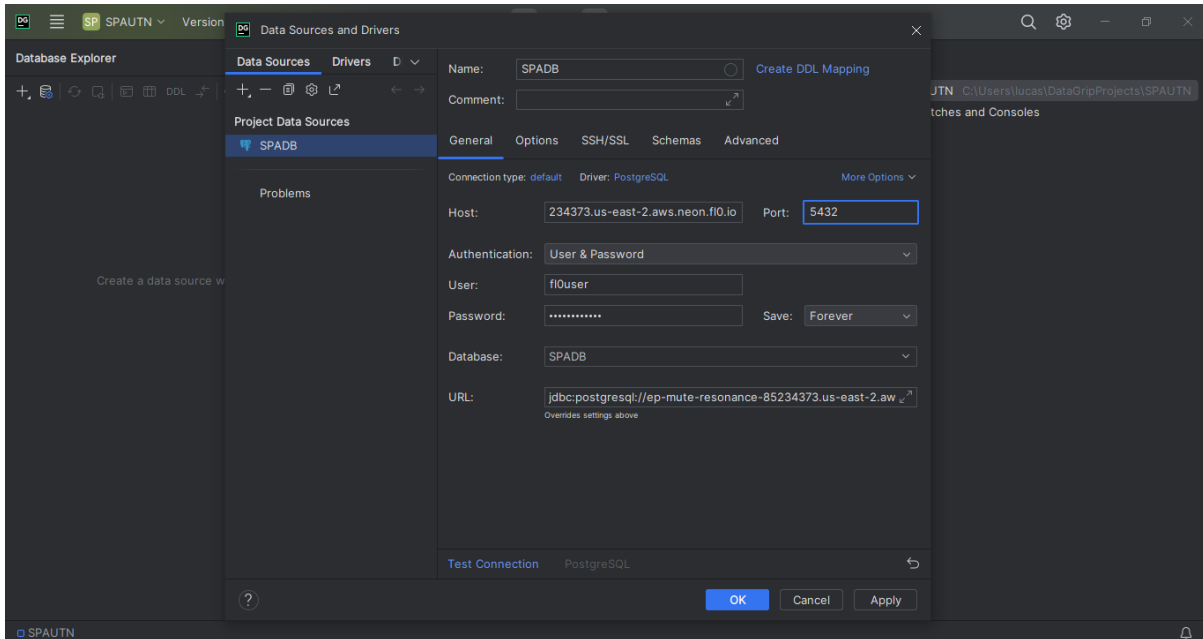
Una vez creado y abierto el proyecto, aparece la interfaz IDE con todas las ventanas de herramientas ocultas. Las ventanas de herramientas son pestañas que se adjuntan a la parte inferior y a los lados de la ventana IDE.



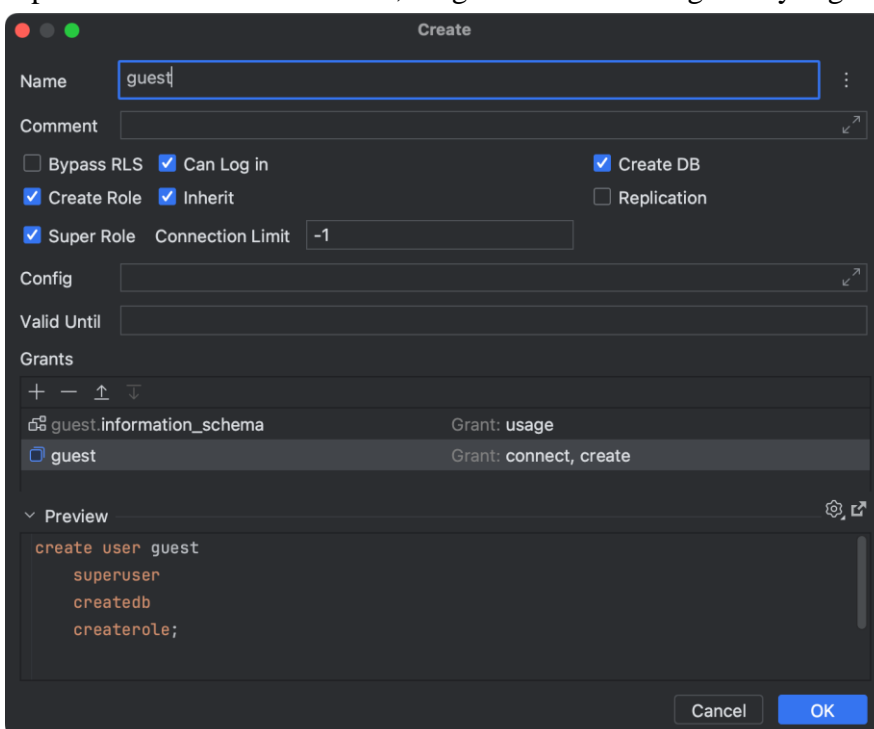


Para conectarse a una base de datos, DataGrip requiere detalles de conexión. Para cada base de datos, los detalles de conexión se almacenan en una configuración de conexión dedicada: origen de datos.

El origen de datos es una configuración de conexión. Almacena una lista de detalles de conexión que se utilizan para establecer la conexión a una base de datos. Por ejemplo, el host, el puerto, el nombre de la base de datos, el controlador, las opciones de configuración SSH y SSL, etc. A continuación, se muestra la configuración de la fuente de datos en el cuadro de diálogo Fuentes de datos y controladores:



Utilizando la funcionalidad de gestión de seguridad, se crearon perfiles de usuario específicos para el equipo. Los permisos fueron asignados para garantizar acceso a los componentes específicos de la base de datos, asegurando así la integridad y seguridad de los datos.

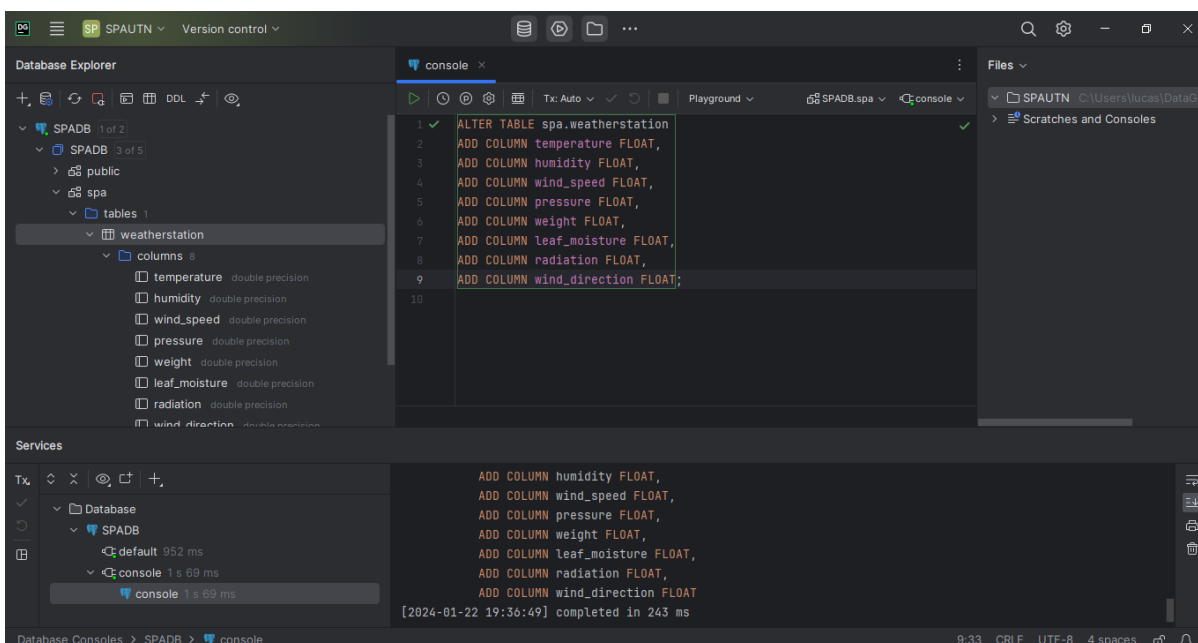
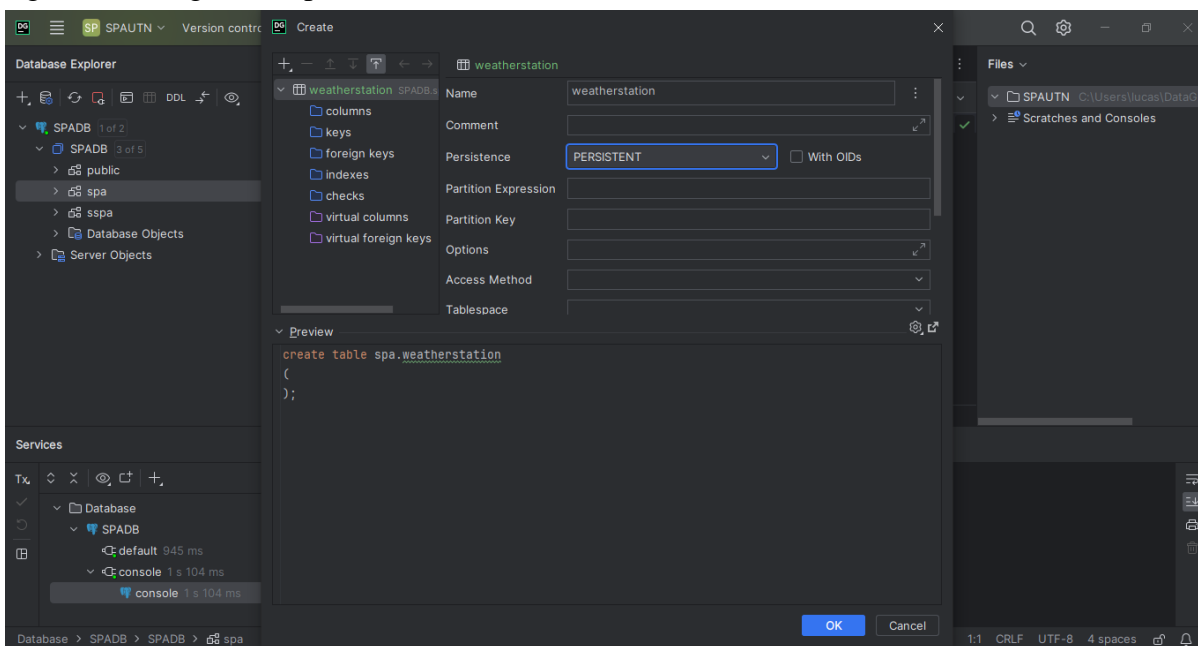




5.1.7.3 Esquemas y tablas

La infraestructura de la base de datos SPADB se organiza en torno a un esquema denominado "spa", el cual alberga cuatro tablas fundamentales: weatherstation, etc, wetweight y logs. En específico, las tres primeras, weatherstation, etc y wetweight, se dedican a almacenar datos generados por el módulo ADCSPA. Por otro lado, la tabla denominada logs tiene como propósito centralizar información relevante relacionada con el funcionamiento del sistema SPA, destinada principalmente a procesos de depuración y seguimiento.

Para la creación de esquemas, tablas y columnas, DataGrip ofrece la posibilidad de hacerlo utilizando su interfaz grafica o directamente por consola utilizando consultas SQL. En las siguientes imágenes se puede observar lo comentado:





A continuación, se presentan las tablas creadas dentro del esquema denominado spa:

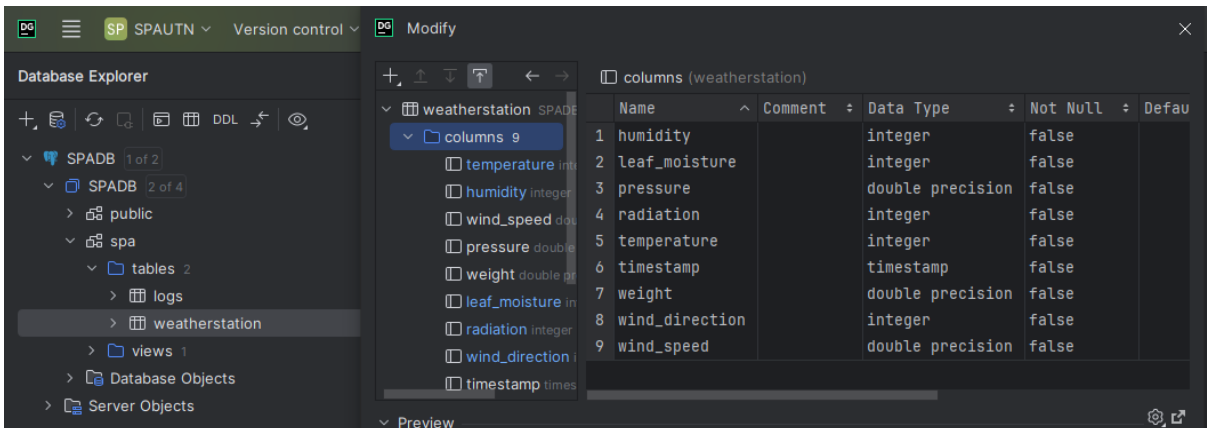
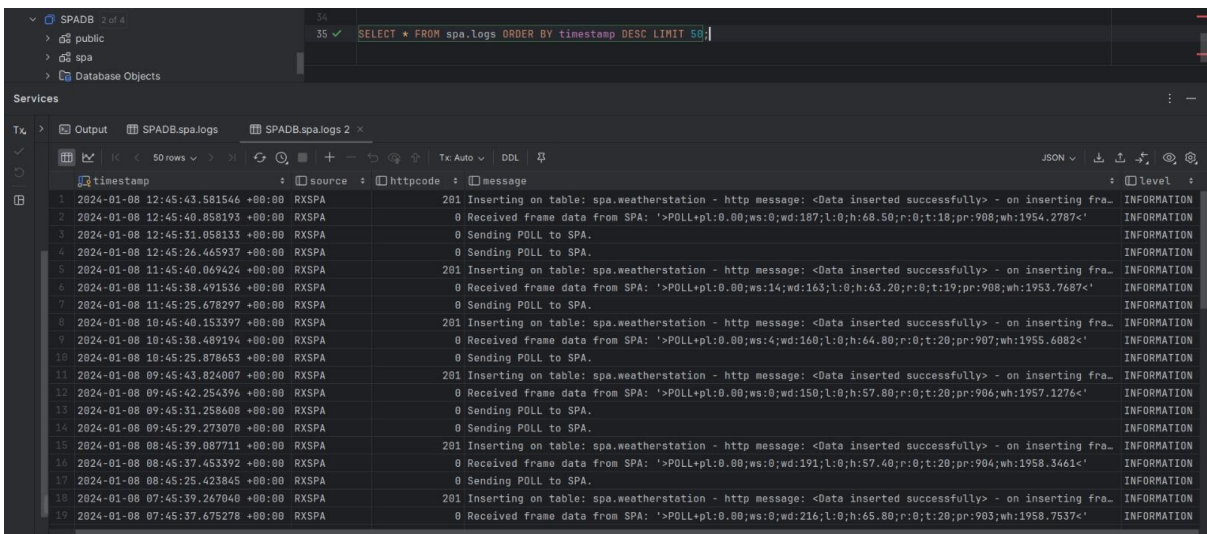
spa.weatherstation									
timestamp	wind_speed	wind_direction	radiation	humidity	temperature	pressure	leaf moisture	pluviometer	weight

spa.etc	
timestamp	etc

spa.wetweights	
timestamp	wetweight

spa.logs				
timestamp	source	httpcode	message	level

En las siguientes imágenes, se pueden observar algunas de las tablas desde DataGrip:





5.1.8 BACKEND-SPA

5.1.8.1 Introducción

El "backend-spa" se presenta como un servidor backend diseñado para facilitar la inserción de datos en la base de datos SPADB. Su función principal radica en recibir datos desde el módulo RXSPA, procesarlos y almacenarlos de manera eficiente en la base de datos PostgreSQL. Este servidor se destaca por su capacidad para gestionar diferentes tipos de datos y proporcionar puntos finales (endpoints) específicos para la inserción de información variada. La autenticación básica se implementa para garantizar la seguridad en el acceso a los puntos finales, asegurando que solo usuarios autorizados tengan la capacidad de enviar datos al servidor.

Objetivos y funcionalidades principales:

- **Inserción de datos:** El servidor es capaz de recibir datos, interpretar la estructura de los datos y almacenarlos en la base de datos correspondiente.
- **Autenticación básica:** Se implementa un mecanismo de autenticación básica para garantizar que solo usuarios autorizados puedan enviar datos al servidor.
- **Flexibilidad en la base de datos:** El sistema se adapta a la estructura de tablas específica de la base de datos, permitiendo la inserción de datos en tablas con columnas variables.
- **Consulta de datos específicos:** Se proporciona un endpoint específico para la consulta de datos particulares, como el "wetweight" más reciente almacenado en la base de datos.
- **Manejo de logs:** Además de la inserción de datos, el servidor puede recibir logs específicos, registrándolos en una tabla dedicada para el seguimiento y análisis.

Para implementar el servidor en un entorno de producción, se optó por la plataforma [Render](#) en lugar de FLO, donde esta alojada la base de datos correspondiente. Esta decisión se tomó debido a que FLO pasó a ser un servicio de pago, y Render proporciona una alternativa eficiente, escalable y gratuita para el despliegue de aplicaciones web y servidores backend.

5.1.8.2 Arquitectura y tecnologías utilizadas

5.1.8.2.1 Uso de Node.js y Express.js

Node.js es un entorno de ejecución de JavaScript basado en el motor V8 de Google Chrome. Permite ejecutar JavaScript en el lado del servidor, lo que facilita la construcción de aplicaciones web escalables y eficientes. Algunas de las características clave de Node.js incluyen su naturaleza asíncrona, que permite la ejecución de operaciones no bloqueantes, y el uso del sistema de módulos que facilita la organización y reutilización del código.

En nuestro caso, Node.js se utiliza como el entorno de ejecución principal para el servidor backend. Permite la ejecución de JavaScript del lado del servidor, proporcionando un entorno eficiente y de alto rendimiento para la gestión de solicitudes HTTP, operaciones de bases de datos y otras tareas del servidor.



Express.js es un marco web para Node.js que simplifica el desarrollo de aplicaciones web y APIs. Proporciona una capa adicional sobre Node.js que facilita la creación de rutas, el manejo de solicitudes y respuestas, y la gestión de middlewares. Express.js es conocido por su simplicidad y flexibilidad, permitiendo crear aplicaciones web de manera rápida y eficiente.

En nuestro caso, Express.js se utiliza como el marco principal para construir el servidor backend. Algunas de las funcionalidades clave de Express.js que se emplean en el código fuente (index.js) incluyen:

- Routing (Enrutamiento): Se define una serie de rutas que gestionan diferentes endpoints del servidor. Por ejemplo, se definen rutas para la inserción de datos, consulta de datos, y otros servicios.
- Middlewares: Se utilizan middlewares de Express para realizar acciones antes o después de que se manejen las solicitudes. En el código, se usa el middleware `express.json()` para analizar datos JSON en las solicitudes y `express.static("public")` para servir archivos estáticos desde la carpeta 'public'.
- Manejo de Errores: Se define una función genérica para el manejo de errores que se integra con Express.js para responder adecuadamente a las excepciones.
- Endpoint Definitions: Express.js facilita la definición de endpoints y la asignación de lógica específica para manejar las solicitudes en esos puntos finales.

La combinación de Node.js y Express.js proporciona un entorno robusto y flexible para la implementación del servidor backend, permitiendo un buen manejo de las solicitudes HTTP y una estructura modular y organizada en el código fuente.

5.1.8.2.2 Integración con PostgreSQL (pg)

PostgreSQL es un sistema de gestión de bases de datos relacional y de código abierto conocido por su robustez, flexibilidad y capacidad para manejar grandes volúmenes de datos. En este caso, PostgreSQL se utiliza como el sistema de gestión de bases de datos para almacenar y recuperar información. En el código fuente (index.js), se emplea el módulo **pg** para Node.js, que proporciona una interfaz para interactuar con PostgreSQL desde una aplicación Node.js. A continuación, se detallan las partes clave relacionadas con la integración de PostgreSQL:

- Importación del módulo PostgreSQL: Se importa la clase Pool del módulo pg. El Pool se utiliza para gestionar conexiones a la base de datos de manera eficiente.

```
const { Pool } = require('pg');
```

- Configuración y creación del Pool de conexiones: Se crea un nuevo pool de conexiones PostgreSQL con la información de conexión proporcionada a través de variables de entorno.

```
const pool = new Pool({
  user: process.env.PG_USER,
  host: process.env.PG_HOST,
  database: process.env.PG_DB,
  password: process.env.PG_PASS,
  port: process.env.PG_PORT,
  ssl: require
});
```



Ejecución de consultas SQL: Se utiliza el pool de conexiones para ejecutar consultas SQL definidas y obtener el resultado.

```
const wetweight_result = await pool.query(WETWEIGHT_QUERY);
```

Construcción y ejecución de consultas dinámicas: Se construye y ejecuta consultas SQL dinámicamente, lo que permite la inserción de datos en tablas específicas con estructuras variables.

```
const query = `INSERT INTO ${tb} (${columns.join(', ')} VALUES  
(${values.map((_, index) => `$$${index + 1}`)}.join(', '))`;`;  
await pool.query(query, values);
```

La integración con PostgreSQL se realiza aprovechando las funcionalidades proporcionadas por el módulo **pg**. La configuración del pool de conexiones se basa en las variables de entorno para garantizar la seguridad y flexibilidad en la gestión de conexiones a la base de datos. La capacidad de construir consultas SQL de manera dinámica facilita la adaptabilidad del servidor a diferentes estructuras de tablas y tipos de datos.

5.1.8.3 Estructura del código fuente (index.js)

El archivo index.js sirve como el punto de entrada principal para el servidor backend. En este archivo, se configuran y definen los elementos esenciales para el funcionamiento del servidor. Define cómo se manejan las solicitudes, qué funcionalidades se aplican a través de middlewares y cómo se gestionan los puntos finales de la API.

A continuación, se proporciona una descripción más detallada de su funcionalidad:

Al principio del archivo, se crea una instancia de la aplicación Express utilizando `express()`. Esta instancia actúa como el objeto central para la definición de rutas, middlewares y la configuración del servidor. Se establece un puerto en el que el servidor Express estará escuchando las solicitudes entrantes. El valor del puerto se obtiene de la variable de entorno `PORT` si está definida; de lo contrario, se utiliza el puerto predeterminado `8080`. Se importan los módulos `path` y `fs` para trabajar con rutas de archivos y el sistema de archivos, respectivamente. Estos módulos son esenciales para la lectura de archivos estáticos y la gestión de rutas en el servidor.

```
const express = require('express');  
const { Pool } = require('pg');  
const app = express();  
const port = process.env.PORT ?? 8080;  
const path = require('path');  
const fs = require('fs');
```

Se define una clase de excepción personalizada para manejar errores de autenticación no autorizada y una función genérica para manejar errores, clasificando adecuadamente las excepciones y respondiendo con mensajes de error apropiados. También una función para autenticar las solicitudes utilizando un token básico proporcionado en las cabeceras HTTP.



Se define un objeto que actúa como mapeo de nombres cortos a nombres completos de columnas en la base de datos. Facilita la interpretación de datos recibidos.

```
class UnauthorizedException extends Error {
  constructor(message) {
    super(message);
    this.name = this.constructor.name;
    this.code = 401;
  }
}
function errorHandler(error, res) {
  // Manejo de errores, respondiendo con el código de error apropiado y un
  // mensaje JSON.
}
function authenticate(basic_token) {
  // Verificación de la autenticación básica mediante comparación del token
  // proporcionado.
}
const columnName = {
  // Mapeo de nombres cortos a nombres completos de columnas en la base de
  // datos.
}
```

Se configuran middlewares utilizando el método `app.use()`. En particular, se utiliza `express.json()` para analizar datos JSON en las solicitudes entrantes y `express.static("public")` para servir archivos estáticos desde la carpeta 'public'. Los middlewares se ejecutan antes de manejar las solicitudes y permiten realizar acciones adicionales, como el análisis de datos o la gestión de archivos estáticos.

```
app.use(express.json());
app.use(express.static("public"));
```

Se definen los puntos finales de la API del servidor Express. Cada punto final está asociado con un tipo de solicitud HTTP (GET, POST, etc.) y tiene una función de manejo específica. Por ejemplo, el punto final `/insert` maneja las solicitudes POST para la inserción de datos en la base de datos.

```
app.post('/insert', async (req, res) => {
  // Lógica para el endpoint de inserción de datos.
});
app.post('/log', async (req, res) => {
  // Lógica para el endpoint de inserción de logs.
});
app.get('/etc', async (req, res) => {
  // Lógica para el endpoint de consulta de datos específicos.
});
app.get('/', (req, res) => {
  // Lógica para el endpoint raíz, sirviendo la página de inicio.
});
app.get('/ping', async (req, res) => {
  // Lógica para el endpoint de recepción de alarmas de ping.
});
```



Se utiliza el método `app.listen()` para iniciar el servidor Express en el puerto configurado. Una vez que el servidor está en funcionamiento, se imprime un mensaje en la consola indicando el puerto en el que está escuchando. Además de gestionar los puntos finales, el servidor también sirve la página de inicio desde la carpeta 'public'. El código está diseñado para capturar cualquier excepción que pueda ocurrir durante la ejecución y gestionarla de manera adecuada mediante la función `errorHandler`.

El código fuente del archivo `index.js`, se puede analizar en el Anexo D: Software SSPA.

5.1.8.3.1 Autenticación y seguridad

- Implementación de autenticación básica:

En la sección de autenticación básica, se establece un mecanismo para autenticar las solicitudes utilizando un método de autenticación básica. Este método implica la inclusión de un token básico en las cabeceras de la solicitud. Aquí se describe la implementación de la autenticación básica en el archivo `index.js`:

Función de autenticación (`authenticate`): Se define una función denominada `authenticate` que toma como parámetro el token básico incluido en las cabeceras de la solicitud (`req.headers.authorization`). Esta función compara el token proporcionado con un valor almacenado en las variables de entorno (`process.env.BASIC_AUTH`). Si los tokens coinciden, la autenticación se considera exitosa; de lo contrario, se lanza una excepción de tipo `UnauthorizedException`.

Uso de la función de autenticación: La función `authenticate` se invoca en puntos específicos del código donde se requiere autenticación. Por ejemplo, se utiliza en los puntos finales `/insert` y `/log` para verificar la autenticidad de las solicitudes.

Excepción de autenticación (`UnauthorizedException`): Se define una clase `UnauthorizedException` que extiende la clase `Error`. Esta excepción se utiliza para representar errores de autenticación no autorizada. La clase incluye un código de estado HTTP 401 para indicar que la solicitud no está autorizada.

```
class UnauthorizedException extends Error {
  constructor(message) {
    super(message);
    this.name = this.constructor.name;
    this.code = 401;
  }
}
```

La implementación de la autenticación básica proporciona una capa de seguridad al verificar la autenticidad de las solicitudes antes de permitir el acceso a puntos finales sensibles. La utilización de una excepción personalizada (`UnauthorizedException`) facilita el manejo específico de errores relacionados con la autenticación en el código. Este enfoque contribuye a la seguridad general del servidor al controlar y restringir el acceso no autorizado a funcionalidades críticas.



- Manejo de excepciones de autenticación:

En la sección de manejo de excepciones de autenticación, se aborda cómo el servidor maneja los errores relacionados con la autenticación. Este enfoque es crucial para proporcionar respuestas claras y seguras cuando una solicitud no cumple con los requisitos de autenticación. Aquí se describe cómo se gestiona el manejo de excepciones de autenticación en el archivo `index.js`:

Función de manejo de errores (errorHandler): Se define una función llamada `errorHandler` que toma como parámetros un error y la respuesta (`res`). Esta función se encarga de manejar diferentes tipos de errores, incluidos los relacionados con la autenticación. Si el error es una instancia de `UnauthorizedException`, se modifica el mensaje de error para indicar que se produjo un error durante la autenticación. Luego, se envía una respuesta JSON con el código de estado 401 y el mensaje de error.

```
function errorHandler(error, res) {
  if (error instanceof UnauthorizedException) {
    error.message += " at authentication";
    console.error(error.message);
    res.status(error.code).json({ error: error.message });
  } else {
    console.error(error.message);
    res.status(500).json({ error: 'Internal Server Error' });
  }
}
```

Manejo de errores en puntos finales: En los puntos finales que requieren autenticación, se utiliza un bloque `try-catch` para capturar posibles excepciones de autenticación. Si se lanza una excepción de `UnauthorizedException`, esta es capturada y manejada por la función `errorHandler`.

```
app.post('/insert', async (req, res) => {
  try {
    authenticate(req.headers.authorization);
    // Lógica para el endpoint de inserción de datos.
  } catch (error) {
    errorHandler(error, res);
  }
});

app.post('/log', async (req, res) => {
  try {
    authenticate(req.headers.authorization);
    // Lógica para el endpoint de inserción de logs.
  } catch (error) {
    errorHandler(error, res);
  }
});
```

El manejo de excepciones de autenticación garantiza que, en caso de fallos durante la autenticación, el servidor responda de manera coherente y segura. La función `errorHandler` desempeña un papel fundamental al clasificar y responder adecuadamente a los errores, proporcionando mensajes claros y códigos de estado HTTP correspondientes. Este enfoque



contribuye a la seguridad y la robustez del servidor al gestionar situaciones no autorizadas de manera controlada.

- Seguridad en las rutas de inserción y log:

En esta sección, se implementa seguridad específica en las rutas de inserción (`/insert`) y registro de logs (`/log`) para garantizar que solo usuarios autenticados y autorizados puedan acceder y realizar operaciones. La seguridad se logra mediante la autenticación básica y la validación del token en estas rutas sensibles. A continuación, se describe cómo se aseguran estas rutas en el archivo `index.js`:

Autenticación en ruta de inserción (`/insert`): La ruta de inserción de datos requiere autenticación básica para garantizar que solo usuarios con credenciales válidas puedan realizar inserciones en la base de datos. Se utiliza la función `authenticate` en el punto final correspondiente para verificar la autenticidad de la solicitud.

```
app.post('/insert', async (req, res) => {
  try {
    authenticate(req.headers.authorization);
    // Lógica para el endpoint de inserción de datos.
  } catch (error) {
    errorHandler(error, res);
  }
});
```

Autenticación en ruta de log (`/log`): De manera similar a la ruta de inserción, la ruta de registro de logs también requiere autenticación básica. Se utiliza la función `authenticate` en el punto final correspondiente para verificar que el usuario esté autenticado antes de permitir el registro de logs.

```
app.post('/log', async (req, res) => {
  try {
    authenticate(req.headers.authorization);
    // Lógica para el endpoint de inserción de logs.
  } catch (error) {
    errorHandler(error, res);
  }
});
```

La implementación de la autenticación básica en estas rutas sensibles mejora la seguridad del servidor, asegurando que solo usuarios autorizados puedan realizar operaciones críticas, como la inserción de datos en la base de datos y el registro de logs. Esto ayuda a prevenir accesos no autorizados y contribuye a la integridad y confidencialidad de los datos manipulados por el servidor.



5.1.8.3.2 Manejo de datos

- Recepción y procesamiento de datos:

En esta sección se detalla cómo el servidor maneja la recepción y procesamiento de datos provenientes de las solicitudes entrantes. El proceso de recepción y procesamiento se centra en los puntos finales `/insert` y `/log`, donde se espera la llegada de datos y se ejecuta la lógica correspondiente. A continuación, se describe cómo se lleva a cabo este proceso en el archivo `index.js`:

Endpoint /insert: Este punto final está diseñado para recibir datos de sensores y realizar la inserción correspondiente en la base de datos. La recepción de datos se realiza a través de una solicitud POST que incluye información en el cuerpo de la solicitud. La lógica de procesamiento implica autenticar la solicitud, extraer la información relevante del cuerpo de la solicitud y construir dinámicamente una consulta SQL para insertar los datos en la base de datos.

Endpoint /log: Similar al punto final `/insert`, la ruta `/log` se encarga de recibir y procesar datos de logs provenientes de las solicitudes POST. La autenticación se verifica, y luego se extraen y procesan los campos relevantes del cuerpo de la solicitud para su posterior inserción en la base de datos.

Ambos puntos finales involucran procesamiento adicional después de la autenticación. En el punto final `/insert`, se extraen y mapean los datos de sensores a las columnas correspondientes de la tabla en la base de datos. En el punto final `/log`, se extraen campos específicos de los logs para su posterior inserción.

```
/ Extract and map sensor data for insertion into the database.
const { tb, fr } = req.body;
const frame = fr.substring(tb.indexOf(">") + 1, fr.indexOf("<"));
var [command, finalFrame] = frame.split("+");
const sensors = finalFrame.split(";");
const columns = [];
const values = [];

sensors.forEach((sensor) => {
  const [name, value] = sensor.split(":");
  columns.push(columnName[name]);
  values.push(value);
});
```

Este enfoque modular y escalonado en el manejo de datos permite al servidor procesar solicitudes entrantes de manera efectiva. La autenticación, la extracción de datos y la construcción de consultas SQL dinámicas son pasos críticos para garantizar la integridad y seguridad de la información manipulada por el servidor.



- Estructura dinámica de las tablas:

El servidor implementa un enfoque que permite la flexibilidad en la manipulación de datos, adaptándose a las diversas fuentes y tipos de información que pueden ser recibidos. A continuación, se describe cómo se logra esta estructura dinámica en el archivo index.js:

Mapeo de nombres de sensores a columnas: Cuando se reciben datos de sensores, el servidor realiza un mapeo dinámico de los nombres de los sensores a las columnas correspondientes en la tabla de la base de datos. Esto permite que los datos de diversas fuentes se inserten correctamente, adaptándose a la estructura de la tabla.

Dinamismo en la construcción de la consulta SQL: Al construir la consulta SQL para la inserción de datos, el servidor utiliza la estructura dinámica para adaptarse a la cantidad y variedad de columnas presentes en los datos. Las columnas y los valores se determinan dinámicamente, permitiendo una inserción versátil y eficiente en la base de datos.

```
const query = `INSERT INTO ${tb} (${columns.join(', ')}) VALUES
(${values.map((_, index) => `$$${index + 1}`)}.join(', '))`;
await pool.query(query, values);
```

Este enfoque permite una mayor flexibilidad al tiempo que mantiene la consistencia en el manejo de la información. La capacidad de mapear dinámicamente los nombres de sensores a las columnas de la tabla mejora la escalabilidad y la mantenibilidad del sistema.

- Inserción de datos en la base de datos:

El servidor implementa lógica específica para recibir los datos procesados y realizar la inserción correspondiente en la base de datos PostgreSQL. A continuación, se describe cómo se lleva a cabo este proceso en el archivo index.js:

Creación de una conexión con la base de datos: Antes de realizar la inserción, el servidor crea una conexión con la base de datos utilizando el módulo pg y la información de conexión proveniente de las variables de entorno.

```
const pool = new Pool({
  user: process.env.PG_USER,
  host: process.env.PG_HOST,
  database: process.env.PG_DB,
  password: process.env.PG_PASS,
  port: process.env.PG_PORT,
  ssl: require
});
```

Construcción de la consulta SQL dinámica: La consulta SQL se construye dinámicamente en función de los datos recibidos y la estructura dinámica de las tablas. Se utilizan las columnas mapeadas y los valores extraídos para formar la consulta.

```
const query = `INSERT INTO ${tb} (${columns.join(', ')}) VALUES
(${values.map((_, index) => `$$${index + 1}`)}.join(', '))`;
```



Ejecución de la consulta en la base de datos: La consulta SQL se ejecuta en la base de datos utilizando la conexión del pool de PostgreSQL. Los valores extraídos se insertan en las columnas correspondientes de la tabla.

```
await pool.query(query, values);
```

La inserción de datos en la base de datos es una parte crítica del manejo de datos, y la implementación en el servidor garantiza que los datos procesados se almacenen de manera segura. La construcción dinámica de la consulta SQL mejora la flexibilidad del sistema para manejar diferentes tipos de datos sin comprometer la seguridad y consistencia en la base de datos.

5.1.8.3.3 Servicios y Endpoints

- Endpoint `"/insert"` para inserción de datos:

En esta sección, se describe el servicio proporcionado por el endpoint `"/insert"` que se utiliza para la inserción de datos en la base de datos. Este endpoint está diseñado para manejar solicitudes POST que contienen información de sensores. A continuación, se detalla su implementación.

Se define el endpoint `"/insert"` utilizando el método `app.post()` de Express. Este endpoint se encarga de recibir datos de sensores y realizar la inserción correspondiente en la base de datos. Antes de procesar la solicitud de inserción, se realiza la autenticación básica utilizando la función `authenticate`. Esto garantiza que solo usuarios autorizados puedan realizar inserciones en la base de datos. Los datos de la solicitud se procesan para extraer la información relevante, como el nombre de la tabla (`tb`) y el frame de datos (`fr`). Se realiza un mapeo dinámico de los nombres de los sensores a las columnas de la base de datos. Se construye dinámicamente una consulta SQL utilizando las columnas y valores extraídos, y luego se ejecuta en la base de datos para insertar los datos. En caso de que la inserción sea exitosa, se envía una respuesta JSON con un código de estado 201 indicando que los datos se han insertado correctamente.

- Endpoint `"/log"` para inserción de logs:

Este endpoint maneja solicitudes POST que contienen información de logs y garantiza la autenticación y la seguridad en la manipulación de estos registros. A continuación, se detalla su implementación.

Se define el endpoint `"/log"` utilizando el método `app.post()` de Express. Este endpoint se encarga de recibir datos de logs y realizar la inserción correspondiente en la base de datos. Al igual que en el endpoint `"/insert"`, se realiza la autenticación básica utilizando la función `authenticate`. Esto asegura que solo usuarios autorizados puedan realizar inserciones de logs. Los datos de la solicitud se procesan para extraer la información relevante, en este caso, el frame de logs (`fr`). Se realiza un análisis específico para obtener los campos necesarios. Se construye dinámicamente una consulta SQL utilizando los campos extraídos y se ejecuta en la base de datos para insertar los logs. Si la inserción de logs es exitosa, se envía una respuesta JSON con un código de estado 201 indicando que los logs se han insertado correctamente.



- Endpoint "/etc" para consulta de datos específicos:

Este endpoint maneja solicitudes GET y garantiza la autenticación antes de ejecutar consultas en la base de datos. A continuación, se detalla su implementación.

Se define el endpoint "/etc" utilizando el método `app.get()` de Express. Este endpoint maneja solicitudes GET y se utiliza para la consulta de datos específicos. Antes de procesar la solicitud de consulta, se realiza la autenticación básica utilizando la función `authenticate`. Esto asegura que solo usuarios autorizados puedan realizar consultas de datos específicos. Se realiza la conexión a la base de datos y se ejecuta una consulta específica para obtener los datos necesarios. En este caso, se realiza una consulta para obtener el valor más reciente del campo (`wetweight`) de la tabla `spa.wetweights`.

```
const wetweight_result = await pool.query(WETWEIGHT_QUERY);
```

Si la consulta de datos específicos es exitosa, se construye una respuesta JSON con los resultados y se envía al cliente.

```
const finalResponse = {  
  wetweight: wetweight_result.rows[0].wetweight  
};  
res.json(finalResponse);
```

La autenticación garantiza que solo usuarios autorizados puedan acceder a esta información. La respuesta JSON contiene los resultados de la consulta para su consumo por parte del cliente.

- Endpoint "/ping" para recepción de alarmas de ping:

Este endpoint maneja solicitudes POST y responde con un mensaje indicando que la alarma de ping ha sido recibida. A continuación, se detalla su implementación.

Se define el endpoint "/ping" utilizando el método `app.post()` de Express. Este endpoint maneja solicitudes POST y se utiliza para recibir alarmas de ping.

```
app.post('/ping', async (req, res) => {  
  try {  
    authenticate(req.headers.authorization);  
    // Lógica para el endpoint de recepción de alarmas de ping.  
    res.status(200).json({ message: 'Ping alarm received' });  
  } catch (error) {  
    errorHandler(error, res);  
  }  
});
```

Antes de procesar la solicitud de alarma de ping, se realiza la autenticación básica utilizando la función `authenticate`. Esto asegura que solo usuarios autorizados puedan recibir estas alarmas. Si la autenticación es exitosa, se envía una respuesta JSON con un código de estado 200 indicando que la alarma de ping ha sido recibida correctamente.



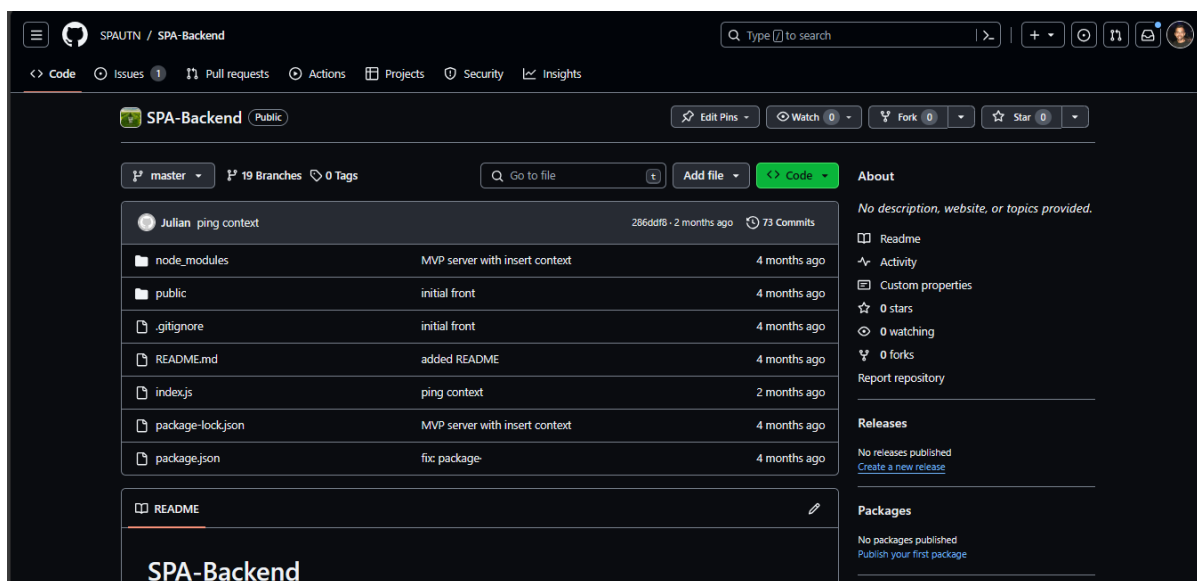
5.1.8.3.4 Despliegue de la API

El despliegue y operación de la API desarrollada en Node.js se simplifica mediante la plataforma [Render](#), que ofrece una solución integral para la gestión de servicios web.



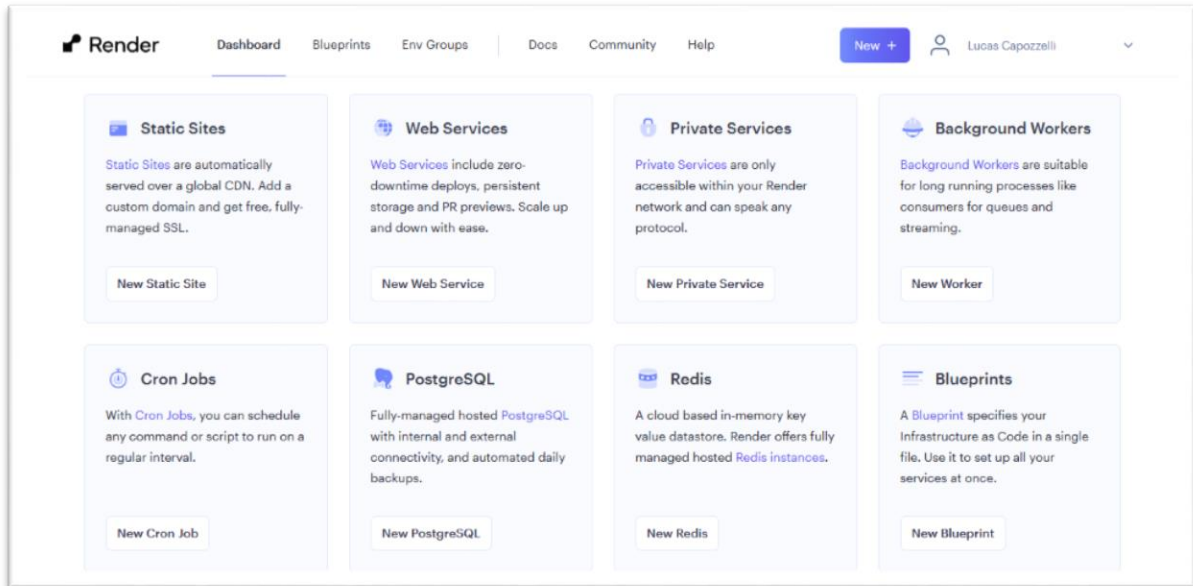
Render es una plataforma de alojamiento y gestión de servicios en la nube que se destaca por su enfoque sencillo para el despliegue de aplicaciones y servicios web. Su interfaz y sus características robustas hacen que sea una opción popular para equipos que buscan una solución integral para el hosting y la administración de sus aplicaciones. Simplifica el proceso de despliegue de aplicaciones, permitiendo a los usuarios conectar sus repositorios de código directamente desde plataformas como GitHub. La plataforma es compatible con una amplia variedad de tecnologías, incluyendo Node.js, Python, Ruby, Docker y más. Esto brinda flexibilidad para utilizar tecnologías preferidas. También ofrece una forma sencilla de gestionar variables de entorno, permitiendo a los usuarios configurar información sensible, como credenciales de base de datos, de manera segura y fácil. La plataforma proporciona herramientas de monitoreo para evaluar el rendimiento de las aplicaciones. Además, permite escalar recursos automáticamente según las necesidades de la aplicación. Render facilita la integración con bases de datos y servicios externos. Los usuarios pueden conectar sus aplicaciones a bases de datos gestionadas y otros servicios para mejorar la funcionalidad de sus aplicaciones.

Una vez creada la API el código fuente se gestiona mediante un repositorio en GitHub denominado "SPA-Backend" dentro de la organización SPAUTN, como se puede ver en la siguiente imagen:

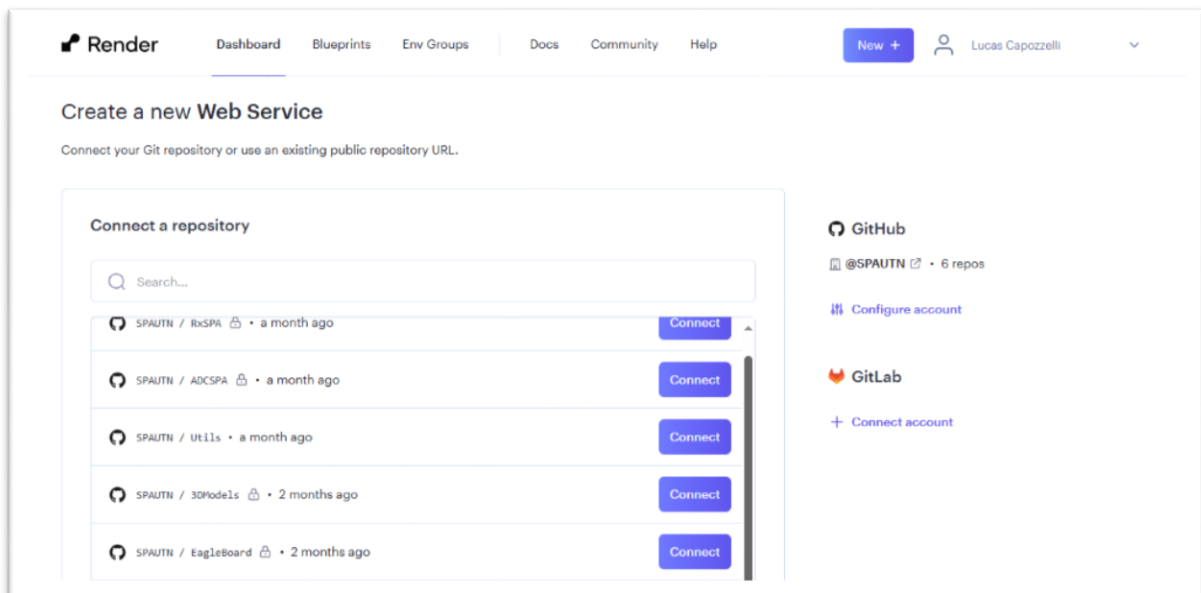


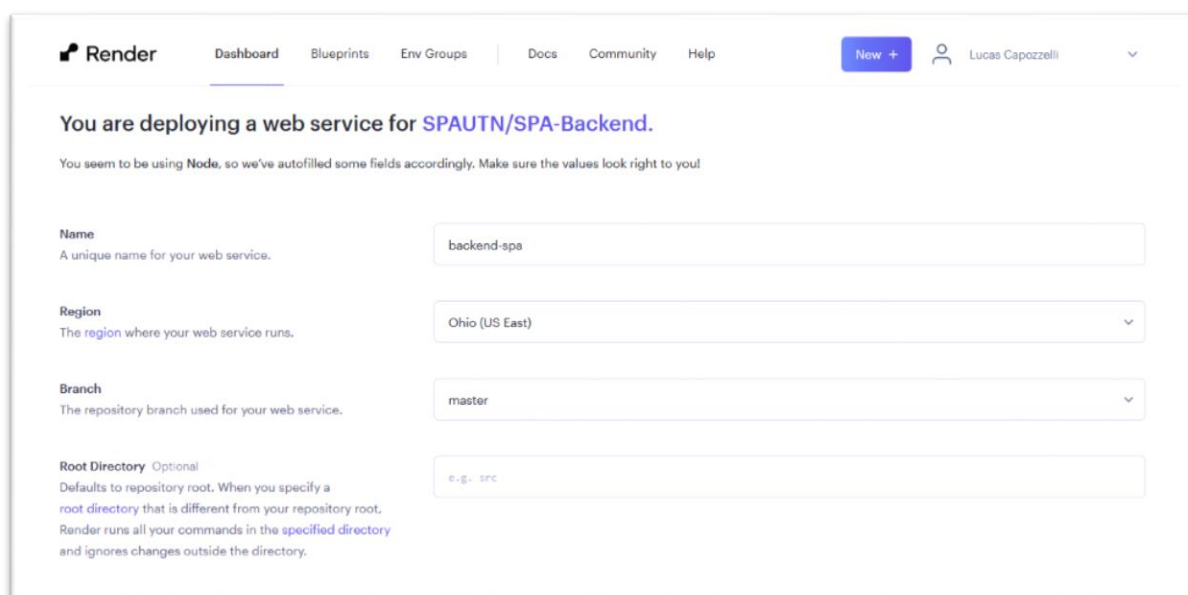


Luego, se accede al panel de control de Render y se selecciona "Crear Nuevo Web Service".



En el proceso de configuración, se elige la tecnología "Node" como entorno de ejecución y se especifican las variables de entorno necesarias, incluyendo la información de conexión a la base de datos (host, usuario, contraseña, entre otros). Además, se indica la URL del repositorio de GitHub y la rama desde la cual se realizará el despliegue. Con estos datos, se inicia el proceso de deploy, que consiste en obtener el código desde el repositorio y configurar el entorno en Render.





Una vez completado el despliegue, Render proporciona una URL única que apunta al servidor backend recién desplegado. Esta URL se utiliza para acceder y consumir la API desde cualquier aplicación cliente. Para cargar datos en la base de datos a través de la API, se hacen solicitudes a los distintos endpoints definidos, como `"/insert"` para la inserción de datos y `"/log"` para la inserción de logs. Durante estas solicitudes, es esencial proporcionar la información requerida y autenticarse utilizando las credenciales adecuadas.

Para obtener datos específicos desde la base de datos, se utiliza el endpoint `"/etc"` y se realizan solicitudes GET a la URL de la API, siempre autenticándose según las medidas de seguridad implementadas. Las herramientas proporcionadas por Render permiten monitorear el rendimiento del servidor y realizar tareas de mantenimiento y actualización del código cuando sea necesario. Este procedimiento garantiza un despliegue seguro de la API, facilitando su utilización y mantenimiento a lo largo del tiempo.

5.1.9 Interfaz gráfica

5.1.9.1 Introducción

La principal misión de la interfaz gráfica es facilitar la comprensión y supervisión de los datos recopilados por el sistema. A través de la presentación visual de métricas clave, la interfaz permite a los usuarios realizar un seguimiento de las condiciones ambientales, identificar patrones y anomalías, y tomar decisiones informadas basadas en la información proporcionada.

La elección de Grafana como herramienta central para la interfaz gráfica del Sistema SPA se fundamenta en su destacada capacidad para la visualización de datos y su versatilidad en la creación de paneles interactivos. Grafana es una plataforma de código abierto que ha ganado reconocimiento por su eficacia en la presentación de datos de manera comprensible y atractiva.

Grafana permite la creación de paneles altamente personalizables, adaptados a las necesidades específicas del monitoreo ambiental. Su flexibilidad gráfica posibilita representar datos en gráficos, tablas y otros elementos visuales de manera eficiente. La capacidad de Grafana para conectarse a diversas fuentes de datos, incluyendo bases de datos como SPADB, facilita la integración sin inconvenientes con los sistemas existentes. Esto garantiza una representación



completa de la información recopilada. Grafana ofrece herramientas interactivas que permiten a los usuarios explorar y profundizar en los datos. Funcionalidades como zoom, filtrado y desplazamiento temporal mejoran la experiencia de análisis. La capacidad de configurar alertas y notificaciones proporciona un mecanismo eficaz para mantener a los usuarios informados sobre eventos críticos o cambios significativos.

- Grafana

Grafana es una plataforma de código abierto diseñada para la visualización y análisis avanzado de datos. Su función principal es proporcionar una interfaz intuitiva y potente para representar información compleja de manera gráfica. Como herramienta de visualización, Grafana permite a los usuarios crear paneles interactivos y cuadros de mando personalizados que facilitan la interpretación y el monitoreo continuo de conjuntos de datos diversos.

Grafana se destaca por su capacidad para integrarse con una variedad de fuentes de datos, incluyendo bases de datos, sistemas de monitorización, y servicios en la nube. Esta versatilidad le confiere la capacidad de aglutinar información desde múltiples fuentes y presentarla de manera coherente y comprensible en un solo lugar.



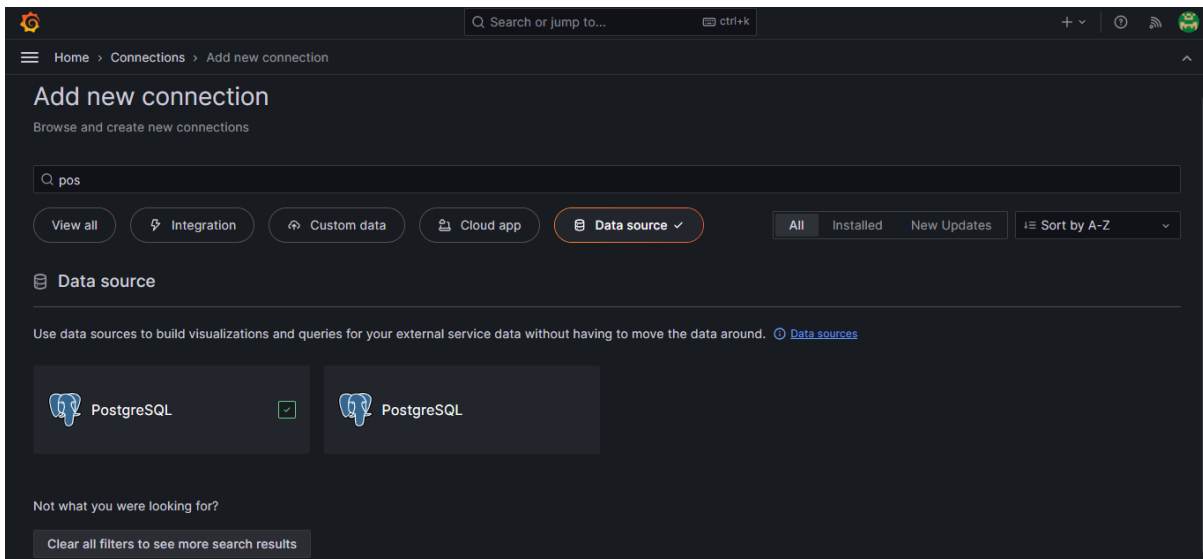
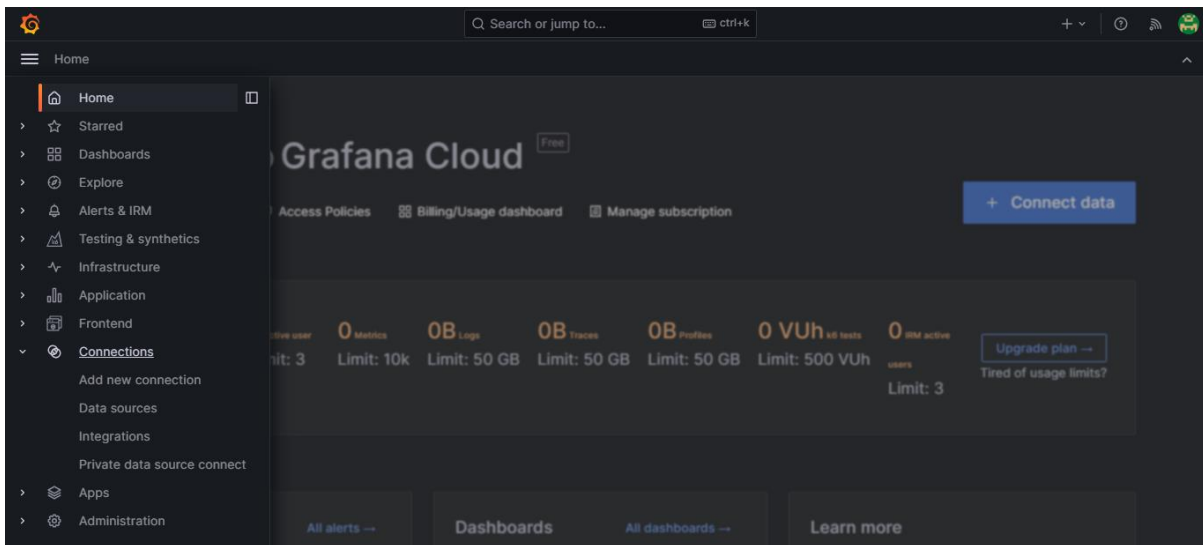
En el contexto de la interfaz gráfica del SSPA, Grafana cumple un rol fundamental al actuar como el componente central para la representación visual de los datos recopilados. Su enfoque en la flexibilidad y capacidad de adaptación hace que sea la elección ideal para traducir los datos del SSPA en representaciones visuales accesibles y significativas para los usuarios.

Grafana se ha convertido en una herramienta ampliamente adoptada en entornos de monitoreo y análisis de datos debido a sus características distintivas que potencian la experiencia del usuario y la efectividad en la representación de información. Algunas de las características clave que contribuyen a la popularidad de Grafana incluyen: flexibilidad gráfica, soporte multi-fuente de datos, interactividad avanzada, configuración de alertas y notificaciones, desarrollo continuo.

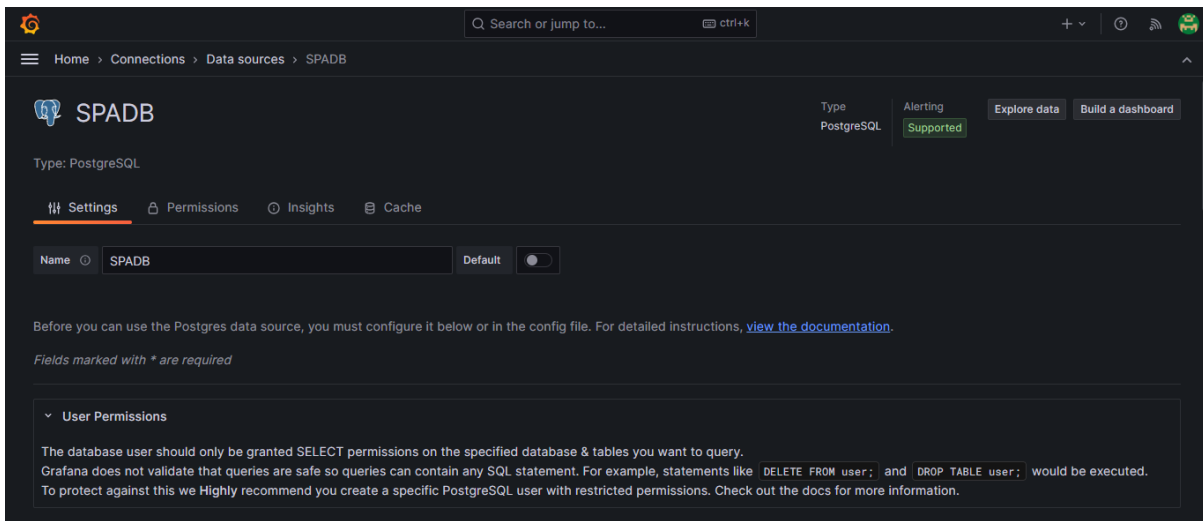
5.1.9.2 Integración de Grafana con SPADB

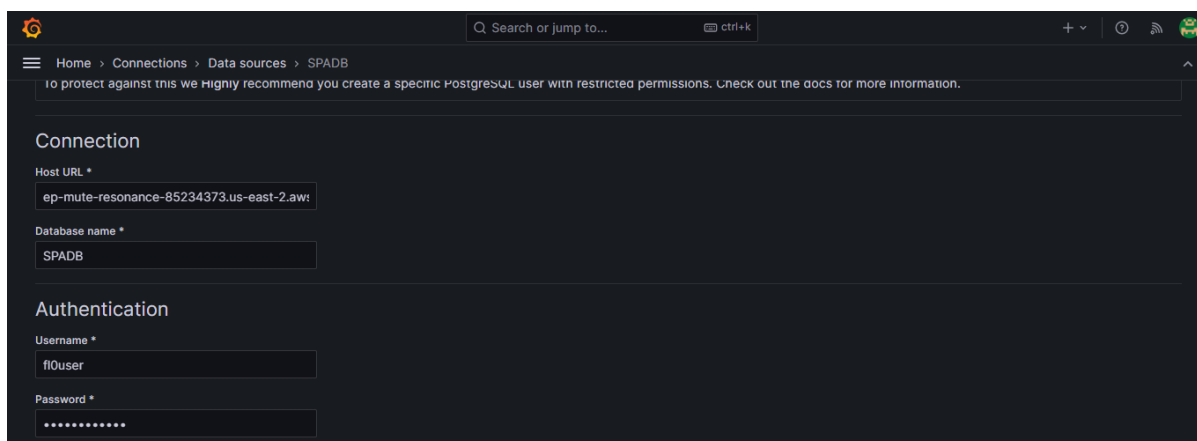
La integración de Grafana con la base de datos SPADB es un paso fundamental para garantizar la visualización de los datos recopilados por el Sistema SPA. El proceso de conexión se lleva a cabo siguiendo estos pasos:

Dentro del panel de administración de Grafana, se accede a la sección de "Configuración". Se selecciona la opción para agregar un nuevo origen de datos y se elige la fuente de datos correspondiente a SPADB (PostgreSQL).

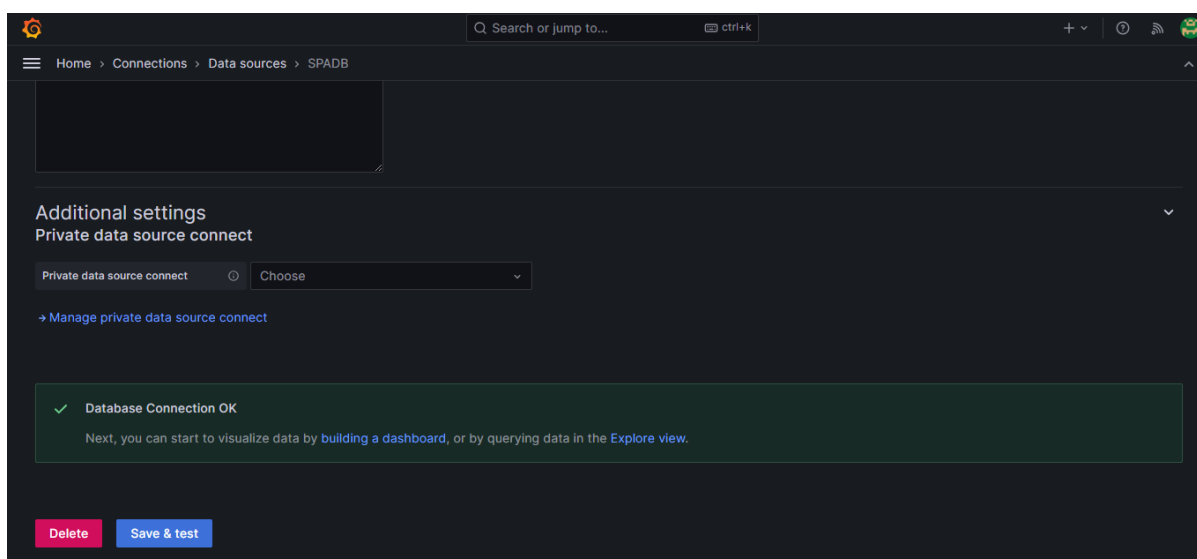


Se completan los parámetros necesarios para establecer la conexión con SPADB. Esto incluye la información de conexión, como la dirección del host, el nombre de la base de datos, el usuario y la contraseña.





Grafana proporciona la opción de realizar una prueba de conexión para verificar que los parámetros ingresados son correctos y que la conexión con SPADB se establece de manera exitosa. Dependiendo de las necesidades, se pueden configurar opciones avanzadas, como el tiempo de consulta, el formato de fecha, y otros aspectos relacionados con la recuperación y presentación de datos. Una vez completada la configuración, se guarda la conexión con SPADB. Grafana ahora tiene la capacidad de acceder y consultar datos de la base de datos.



Una vez establecida la conexión de Grafana con la base de datos SPADB, el siguiente paso crítico es la configuración que permite a Grafana acceder y consultar los datos de los sensores del SSPA. Este proceso se lleva a cabo mediante los siguientes pasos:

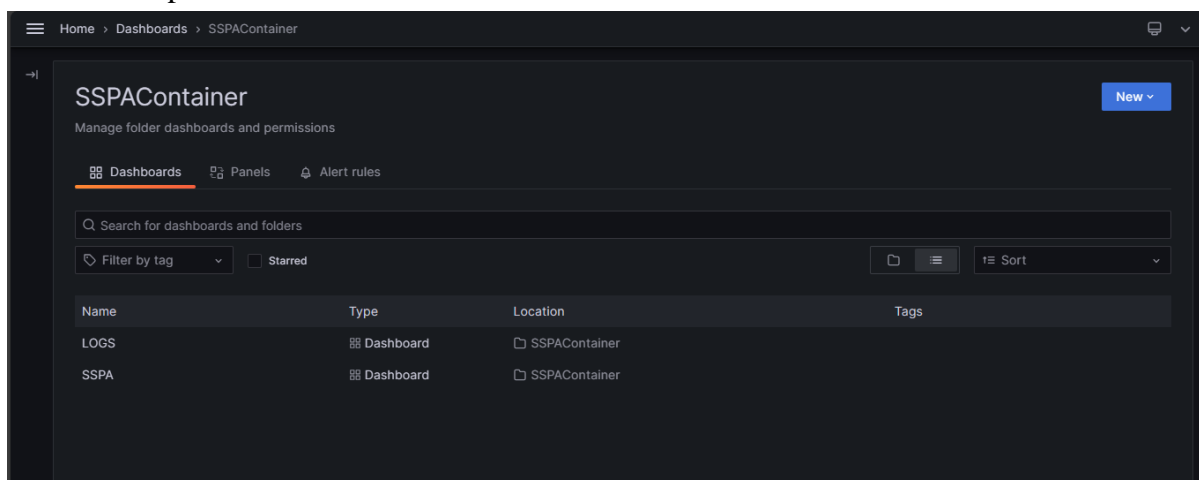
En el entorno de Grafana, se accede al panel de creación y configuración de paneles. Estos paneles serán las representaciones visuales de los datos provenientes de los sensores. Para cada panel creado, se elige el origen de datos configurado previamente que apunta a la base de datos SPADB. Esto establece la conexión específica para la recuperación de datos. Grafana permite la configuración de intervalos temporales para la recuperación de datos, lo que posibilita la visualización de datos históricos y actuales. Se aplican configuraciones específicas de visualización para cada panel, adaptando el formato, colores y escalas según los requisitos de presentación de datos. Una vez configurados los paneles de manera adecuada, se guardan las configuraciones.



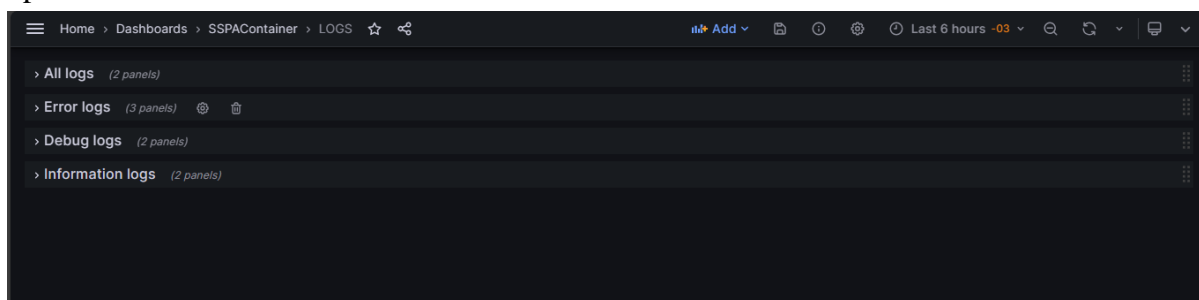
5.1.9.3 Diseño de dashboards

En esta sección, nos enfocaremos en el diseño de los dashboards que integran la interfaz gráfica del Sistema SPA. Estos dashboards estarán alojados dentro de la carpeta denominada SSPAContainer, ofreciendo una representación visual integral del sistema.

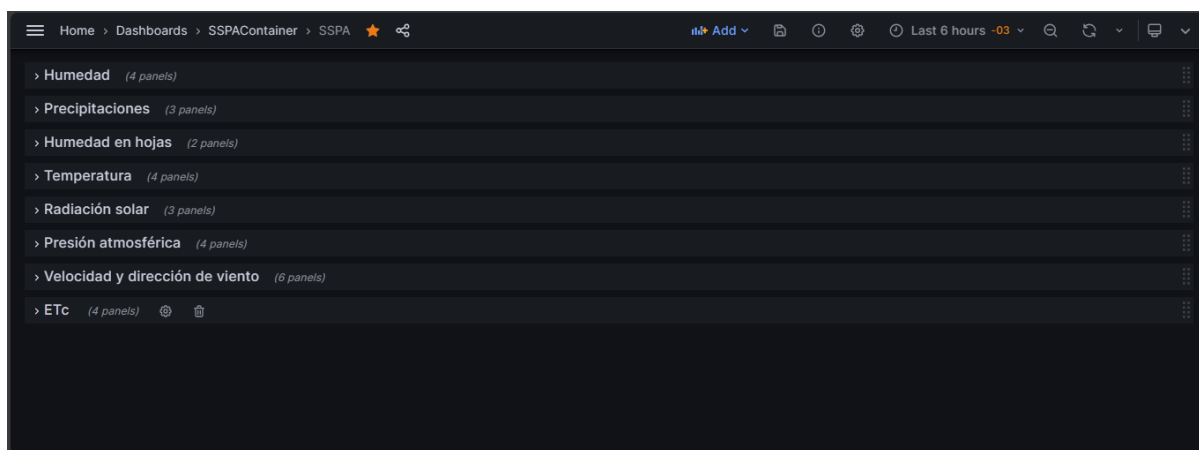
Los dashboards desempeñan un papel importante en el SSPA, proporcionando ventanas visuales específicas que permiten el monitoreo y análisis del mismo. Dentro de SSPAContainer, se destacan dos tableros principales: "LOGS" y "SSPA". Cada uno de estos tableros ha sido diseñado cuidadosamente para ofrecer visualizaciones específicas que abordan diferentes aspectos del sistema.



LOGS: Este dashboard se dedica a la exposición detallada de registros y eventos del sistema. Estructurado en secciones específicas, como "Error Logs", "Debug Logs" e "Information Logs", proporciona una visión de la salud y el rendimiento del sistema. Este enfoque permite una rápida identificación y resolución de problemas, asegurando la integridad y la eficiencia operativa del SSPA.



SSPA: Diseñado para visualizar todas las métricas ambientales recopiladas por el módulo ADCSPA, este dashboard organiza visualmente ocho filas, cada una dedicada a una métrica específica, como humedad, precipitaciones, temperatura, entre otras. Este enfoque por categorías facilita el análisis de las condiciones del entorno y el comportamiento de los sensores, proporcionando información para la toma de decisiones informadas.



5.1.9.3.1 Dashboard SSPA

El Dashboard SSPA se organiza en ocho filas, cada una dedicada a una métrica específica. Esta estructura permite una visualización clara y organizada de los datos, facilitando la interpretación y el análisis. Las filas se distribuyen de la siguiente manera:

- Humedad
- Precipitaciones
- Humedad en hojas
- Temperatura
- Radiación solar
- Presión atmosférica
- Velocidad y dirección del viento
- ETc

A continuación, veremos en detalle cada una de ellas:

Humedad

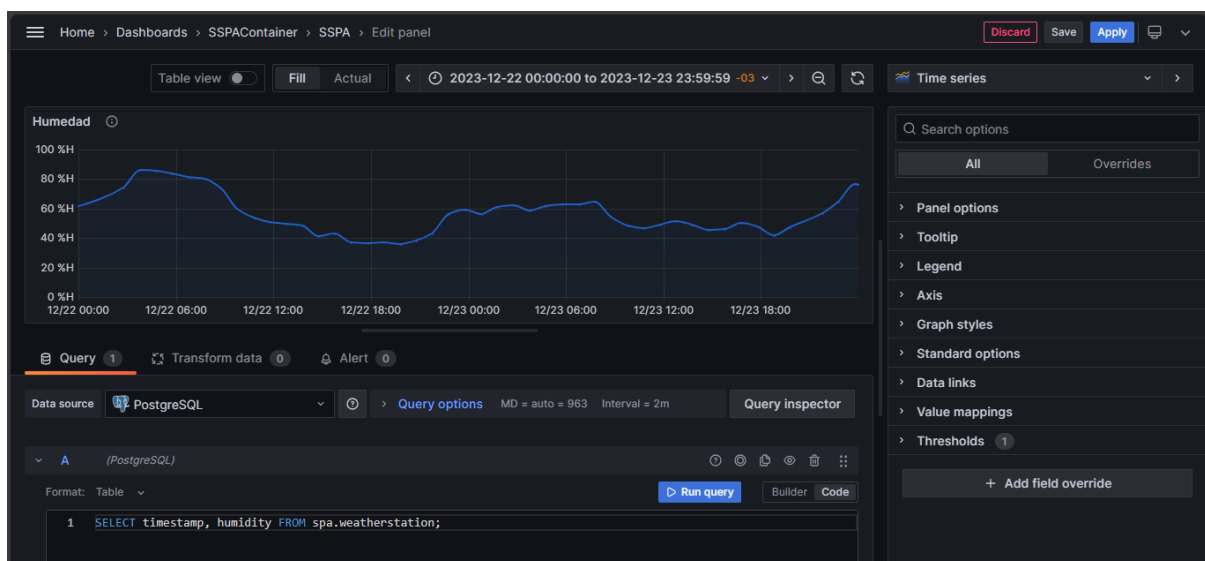




Para el sensor de humedad, se han desarrollado cuatro paneles, cuya disposición se detalla en la imagen proporcionada. En una primera instancia, el primer panel exhibe la variación de la humedad ambiental a lo largo del tiempo, ofreciendo una representación gráfica que permite analizar las fluctuaciones y tendencias a lo largo de un período determinado.

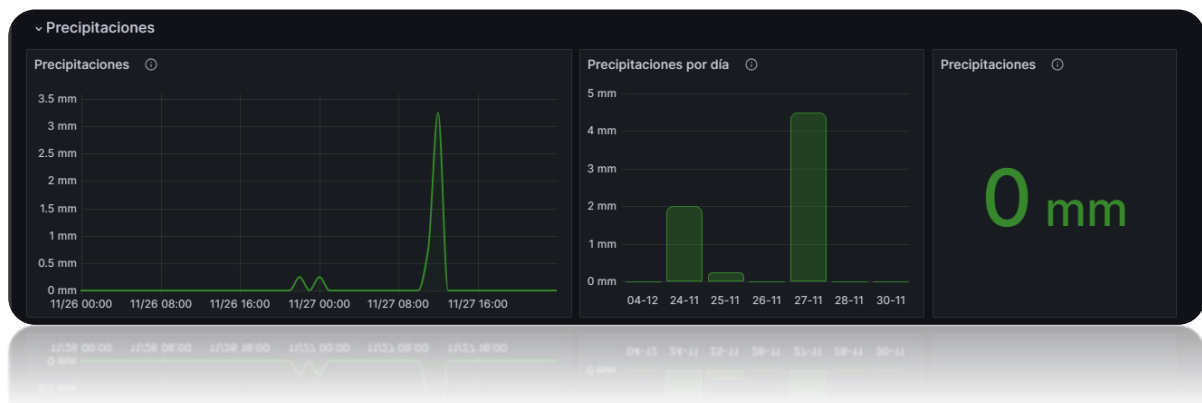
En la sección adyacente, ubicada a la derecha, se presenta la humedad ambiente instantánea, es decir, la medida más reciente registrada por el sensor. Esta visualización rápida proporciona información sobre las condiciones ambientales actuales. Además, se incluyen los valores máximo y mínimo de humedad observados a lo largo del día, lo que permite al usuario obtener una perspectiva completa de la variabilidad de la humedad en un intervalo de 24 horas.

A continuación, se presenta una imagen durante el diseño de uno de los paneles del sensor de humedad:



Como se puede observar en la parte superior tenemos una previsualización de como quedará el panel si confirmamos el diseño. En la parte de abajo, es donde se hace la conexión a la base de datos correspondiente y se escriben las consultas SQL para traer los datos de interés, necesarios para construir la grafica. Por ultimo, en la sección de la derecha, se permite seleccionar el tipo de representación y editar todas sus características.

Precipitaciones





Para el pluviómetro, se ha diseñado un conjunto de tres paneles, cuyas funciones se observan en la imagen proporcionada. En primer lugar, el panel principal presenta la variación de las precipitaciones a lo largo del tiempo, proporcionando una representación gráfica y detallada que facilita la observación de patrones y tendencias en la acumulación de precipitaciones.

En la sección adyacente, situada a la derecha, se encuentra un panel dedicado a la visualización de las precipitaciones acumuladas por día. Esta representación permite una evaluación rápida de la cantidad total de precipitación registrada durante la última semana, brindando una perspectiva de las condiciones pluviales recientes. Finalmente, el último panel muestra las precipitaciones registradas en la última hora, proporcionando una medición inmediata de la intensidad de la lluvia.

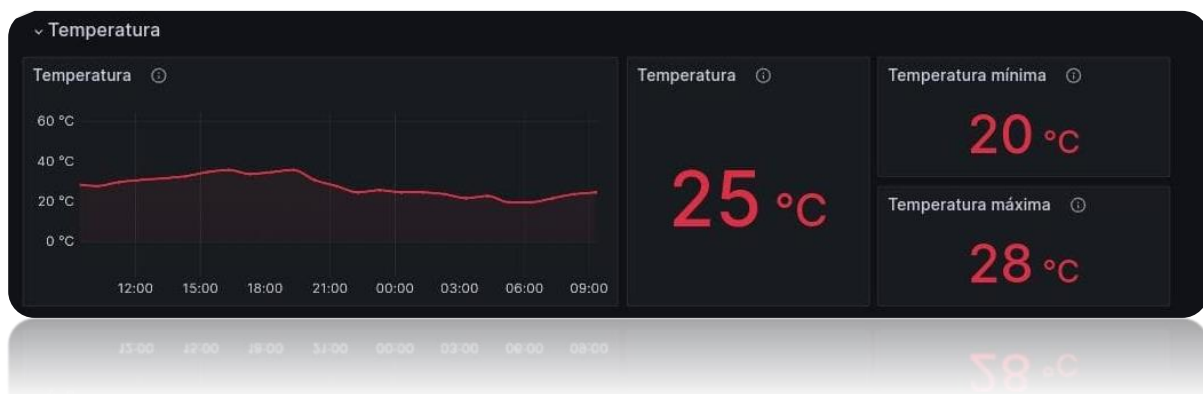
Humedad en hojas



Para el sensor de humedad de hojas, se han diseñado dos paneles, cada uno con funciones específicas, como se observa en la imagen de arriba. En el primer panel, se presenta la variación temporal de la humedad de las hojas, ofreciendo una representación gráfica que permite observar las fluctuaciones y patrones a lo largo de un periodo determinado.

En el segundo panel, se focaliza en la humedad de hoja actual, mostrando la última medición registrada por el sensor. Esta visualización proporciona información inmediata sobre el estado actual de la humedad de las hojas, permitiendo una evaluación rápida de las condiciones ambientales en el entorno de cultivo o área monitoreada.

Temperatura





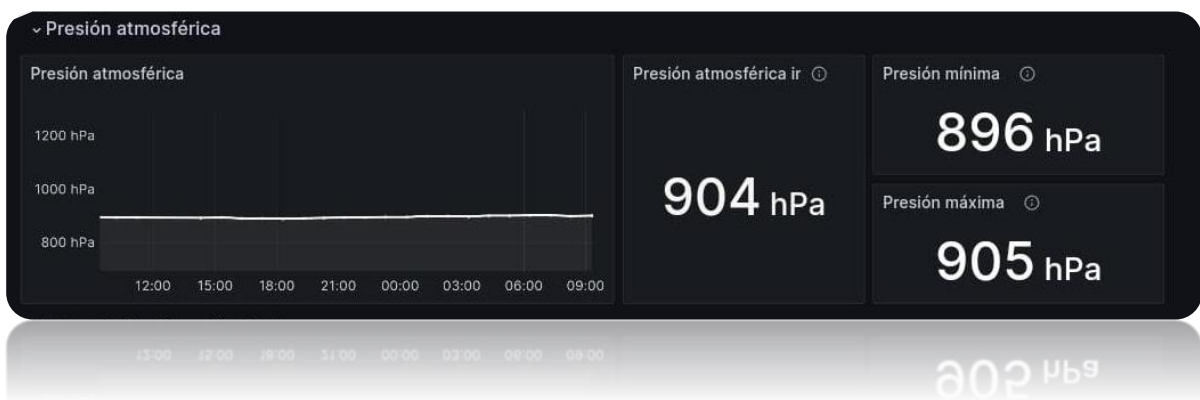
Para el sensor de temperatura, se crearon cuatro paneles, cada uno diseñado para proporcionar información específica, como se detalla en la imagen. En primer lugar, el panel principal presenta la variación temporal de la temperatura, ofreciendo una representación gráfica detallada que permite observar las fluctuaciones y tendencias a lo largo del tiempo. Luego se exhibe la temperatura actual, proporcionando una medición para evaluar las condiciones térmicas actuales. Los dos paneles restantes se centran en las condiciones extremas del día. Uno de estos paneles muestra la temperatura mínima registrada en el día, mientras que el otro panel presenta la temperatura máxima registrada en el mismo período. Estos paneles ofrecen una visión completa de las variaciones térmicas diarias, permitiendo al usuario obtener una comprensión integral de las condiciones térmicas experimentadas a lo largo del día.

Radiación solar



Para el sensor de radiación solar, se han implementado tres paneles, cada uno destinado a ofrecer información específica. El primer panel destaca la variación temporal de la radiación solar, proporcionando una perspectiva de las fluctuaciones y patrones a lo largo de un periodo determinado. En el segundo panel, se presenta la acumulación de radiación solar diaria, ofreciendo una visión general de la cantidad total de radiación solar recibida en los últimos días. Esta representación facilita el análisis de las tendencias a lo largo del tiempo y permite una comprensión más completa de las condiciones de radiación solar. El tercer panel se enfoca en la radiación solar actual. Esta representación permite una evaluación precisa de las condiciones actuales de radiación solar.

Presión atmosférica





Para el sensor de presión atmosférica, se han desarrollado cuatro paneles, cada uno diseñado para proporcionar información, según se puede observar en la representación gráfica. Inicialmente, el panel principal destaca la variación temporal de la presión atmosférica, ofreciendo una representación gráfica que permite observar las fluctuaciones a lo largo del tiempo. En el segundo panel, se presenta la presión atmosférica actual, proporcionando una medición para evaluar las condiciones atmosféricas actuales. Los dos paneles restantes se centran en las condiciones extremas del día. Un panel muestra la presión atmosférica mínima registrada en el día, mientras que el otro panel presenta la presión atmosférica máxima registrada en el mismo periodo.

Velocidad y dirección del viento



Para el sensor de velocidad del viento, el primer panel resalta la variación temporal de la velocidad del viento, brindando una representación gráfica detallada de las fluctuaciones y patrones a lo largo del tiempo. Luego en otro panel, se presenta la velocidad mínima registrada en el día, ofreciendo una perspectiva de las condiciones menos intensas. Además, otro panel muestra la velocidad máxima registrada en el día, proporcionando información sobre las condiciones más intensas experimentadas. Luego, el panel de la velocidad actual del viento, ofrece una medición para evaluar las condiciones actuales de viento.

Para el sensor de dirección del viento, un panel presenta la dirección del viento actual y el otro muestra la distribución de la dirección del viento en las últimas 24 horas, ofreciendo una visión de los patrones y cambios en la dirección del viento a lo largo del día. La longitud de las barras que se extienden desde el centro hacia fuera indica la frecuencia o la cantidad de veces que el



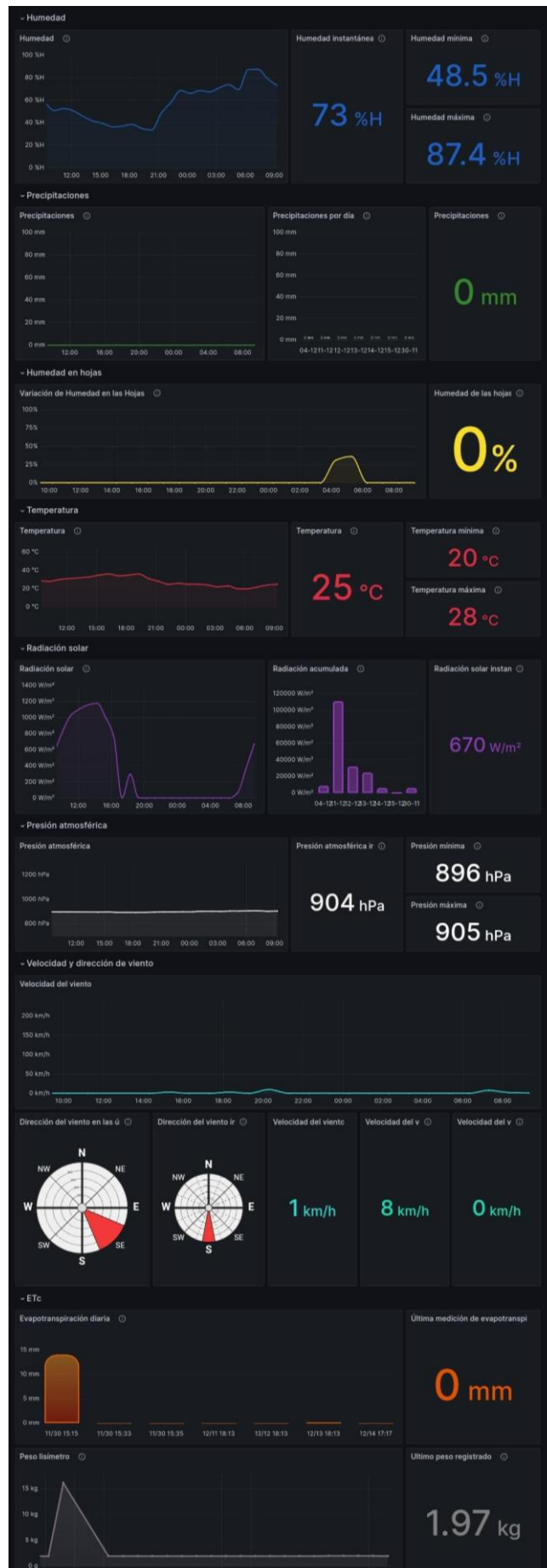
viento sopla desde esa dirección en el período de tiempo especificado. Cuanto más larga es la barra, más común es el viento desde esa dirección.

ETc



Para el lisímetro de pesada, se han diseñado dos conjuntos de información clave, detallados en la representación gráfica adjunta. En el primer conjunto, se destaca un gráfico que ilustra la Evapotranspiración del cultivo (ETc) diaria, permitiendo la observación de los últimos 7 días. Junto a este gráfico, se presenta la última ETc calculada, proporcionando una visión inmediata de las demandas hídricas recientes del cultivo. En el segundo conjunto, se encuentra un gráfico temporal que muestra el peso registrado por el lisímetro de pesada. Este gráfico proporciona una representación detallada de las variaciones en el peso del lisímetro a lo largo del tiempo. A la derecha de este gráfico, se presenta el último peso registrado, ofreciendo una medida actualizada del contenido hídrico del suelo.

En la siguiente imagen se presenta en forma conjunta como es la visualización de todos los paneles cuando se ingresa al dashboard SSPA:





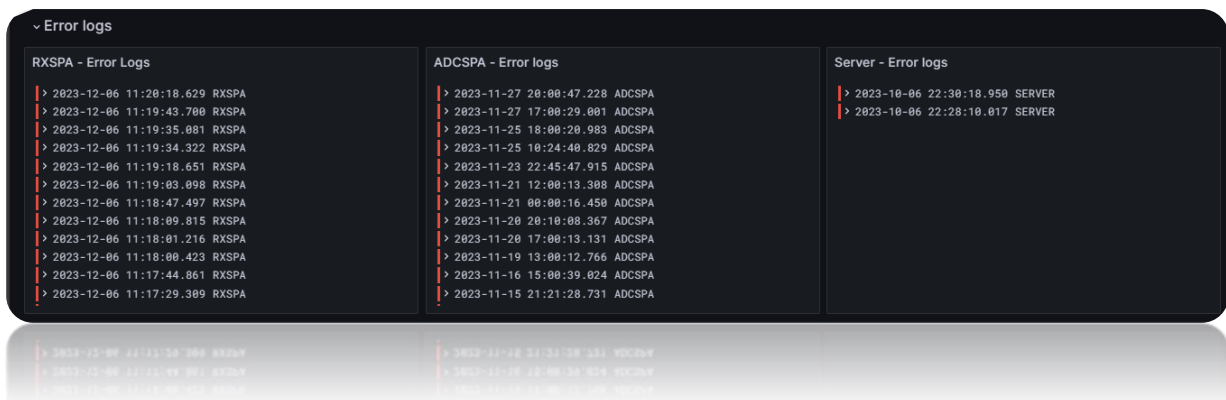
5.1.9.3.2 Dashboard LOGS

El dashboard LOGS constituye una herramienta valiosa dentro del Sistema SPA, enfocándose en la visualización de registros y eventos significativos. Este dashboard se ha estructurado de manera estratégica para proporcionar una comprensión clara y precisa del funcionamiento del sistema, facilitando la detección temprana de irregularidades y errores. A continuación, se describen los elementos clave presentes en este tablero.

El dashboard se organiza en tres secciones principales: "Error logs", "Debug logs" e "Information logs". Estas secciones agrupan y presentan registros específicos en función de su importancia y naturaleza. Los paneles brindan una visión detallada del funcionamiento del sistema. Los usuarios pueden seguir el rendimiento general y detectar posibles problemas mediante una presentación estructurada de registros.

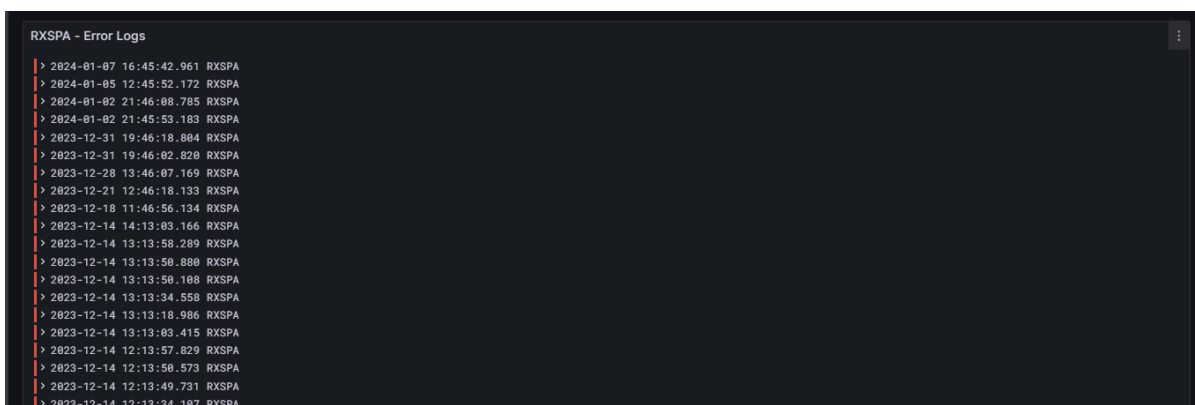
Error logs

La sección "Error Logs" presenta tres paneles distintos, cada uno centrado en la visualización y análisis de registros de errores específicos provenientes de distintas fuentes del sistema SPA.



- RXSPA - Error Logs

Este panel se enfoca en la recopilación y exhibición de errores relacionados con el módulo RXSPA del sistema SPA. Los registros aquí presentes proporcionan información crítica sobre cualquier anomalía o problema específico en el componente RXSPA.

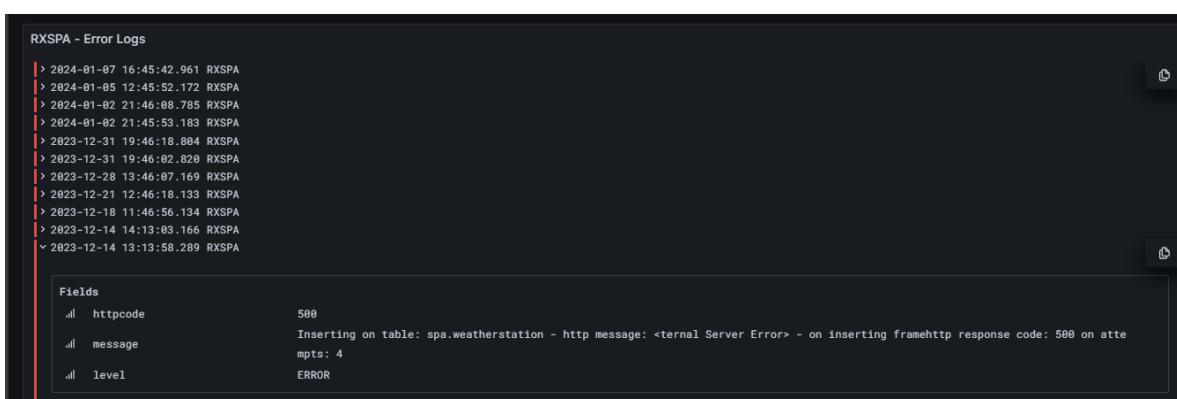




A continuación, se pueden observar algunos de los errores detectados en el modulo RXSPA:



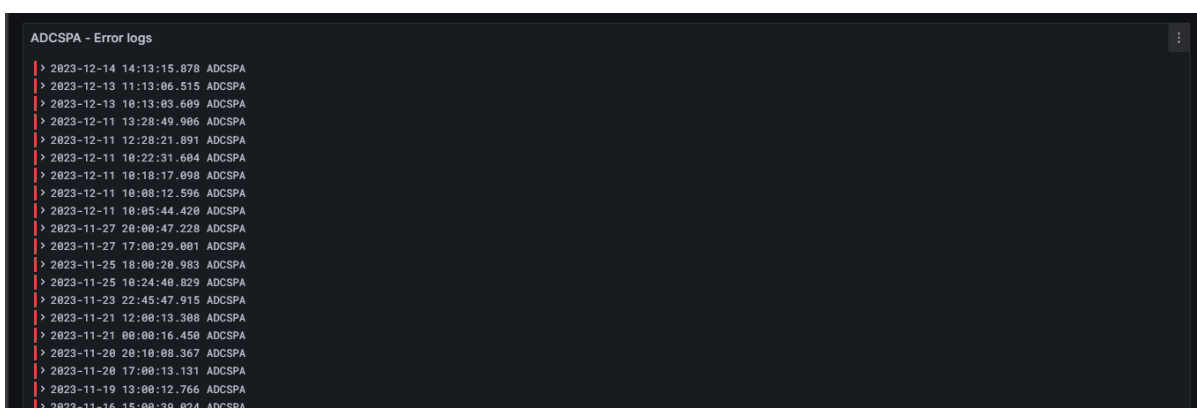
En este caso “*Not frame received, resending command POLL to SPA. Attempt: 1*”, indica que el módulo RXSPA no ha recibido la información que le solicitó al módulo ADCSPA, por lo que se la solicitará nuevamente a través del comando POLL.



En este otro caso “*Inserting on table: spa.weatherstation – http message: <internal server error> - on inserting frame http response code: 500 on attempts: 4*”, indica que hubo un problema al intentar realizar una inserción de datos en la tabla spa.weatherstation. Puede deberse a diversos motivos, como un error en la estructura de la consulta SQL, problemas de conexión con la base de datos, o situaciones similares que deben ser investigadas y corregidas en el lado del servidor.

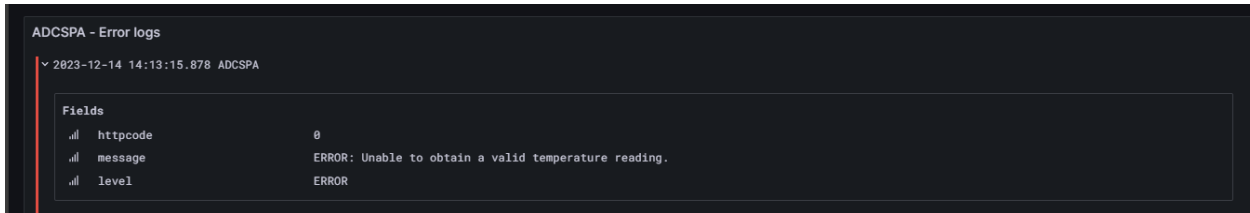
- ADCSPA - Error Logs

Este panel se dedica a la presentación de errores provenientes del módulo ADCSPA. Analizando los registros de este panel, los usuarios pueden identificar problemas específicos relacionados con la adquisición de datos en el sistema SSPA.





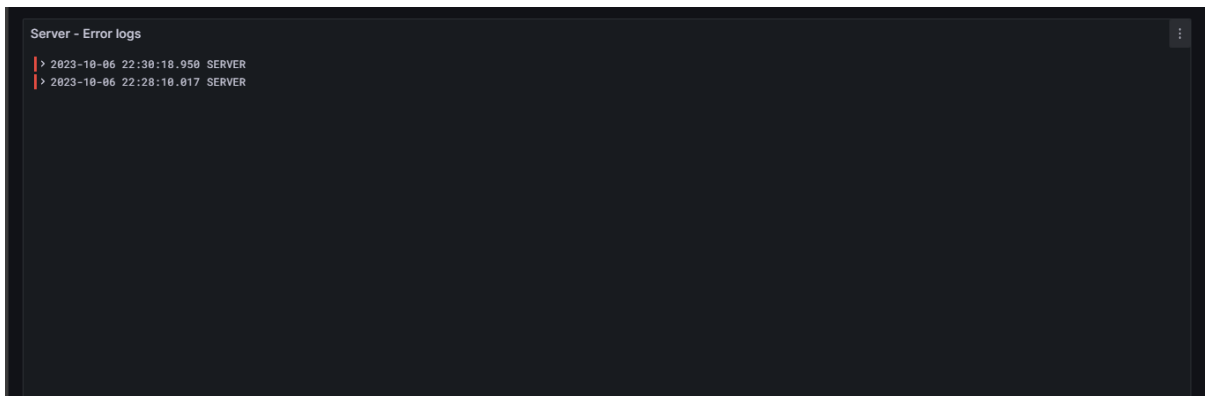
A continuación, se puede observar un error detectado en el modulo ADCSPA:



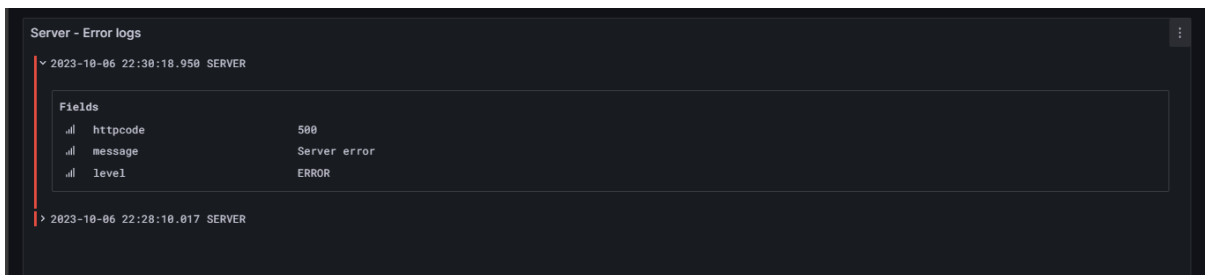
En este caso “Unable to obtain a valid temperature reading ”, indica que el sensor de temperatura no ha sido capaz de obtener el valor correspondiente.

- Server – Error logs

Enfocado en los errores generados por el servidor, este panel proporciona una visión de problemas que podrían afectar la conectividad o el procesamiento general del sistema SSPA.

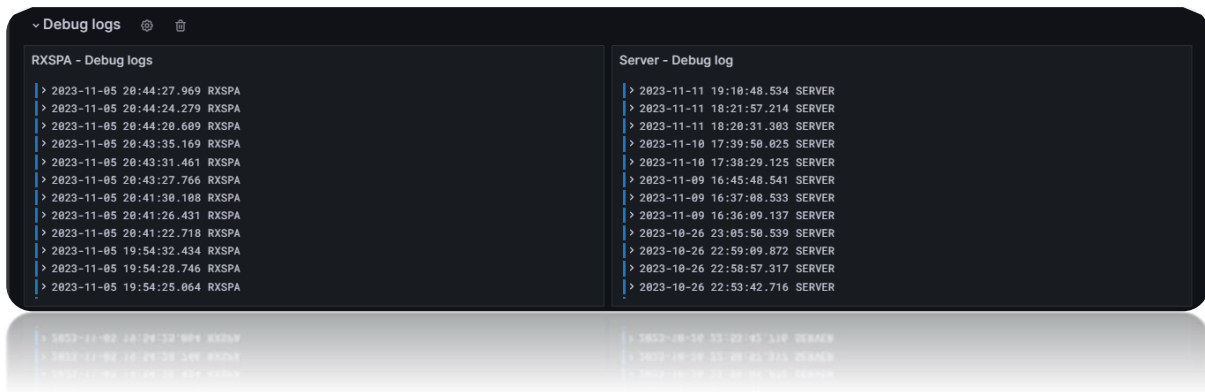


A continuación, se puede observar un error detectado en el servidor:



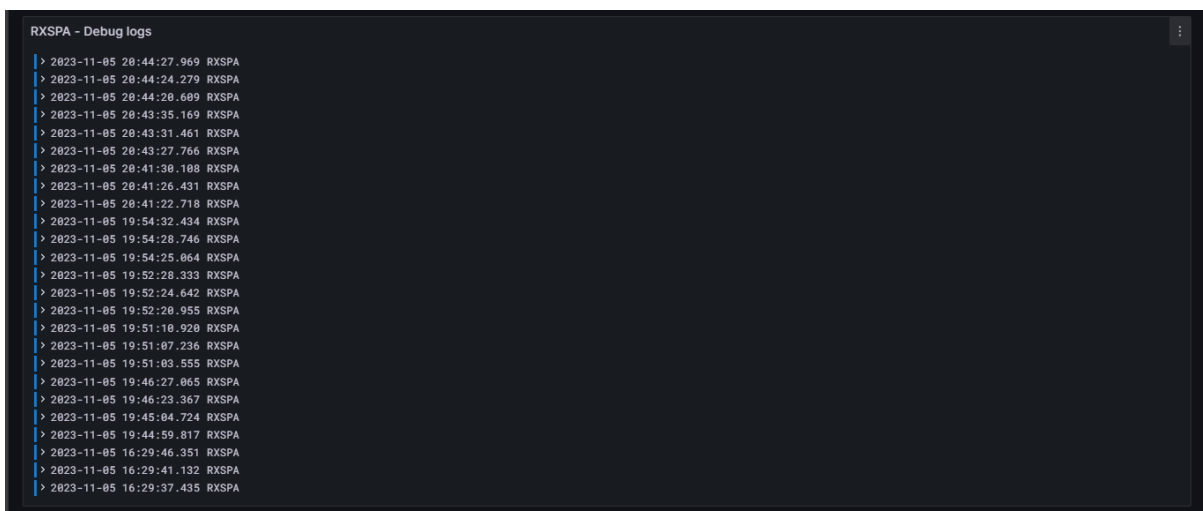
Debug logs

Esta sección presenta dos paneles, cada uno diseñado para ofrecer una visión del flujo de ejecución y permitir un análisis del sistema SPA.

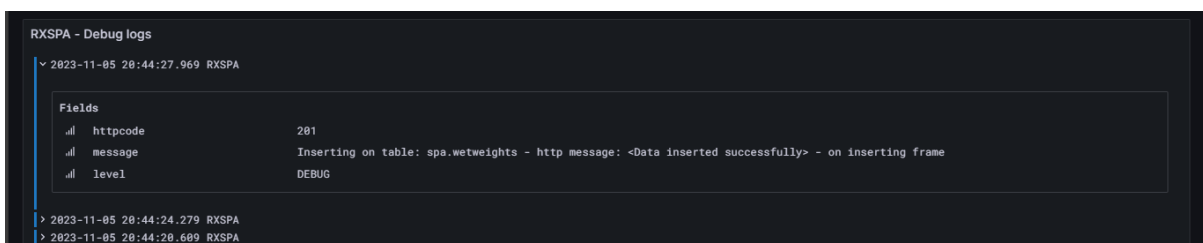


- RXSPA - Debug logs

Este panel se centra en proporcionar registros de depuración específicos del módulo RXSPA. Los registros aquí presentes son esenciales para entender el comportamiento interno y la ejecución de tareas en el RXSPA.



A continuación, se pueden observar algunos de los registros de depuración detectados:



En este caso “*Inserting on table: spa.wetweights - http message: <Data inserted successfully> - on inserting frame*” indica que la operación de inserción de datos en la tabla "spa.wetweights" se realizó con éxito.



```
RXSPA - Debug logs
> 2023-11-05 20:44:27.969 RXSPA
> 2023-11-05 20:44:24.279 RXSPA
> 2023-11-05 20:44:20.609 RXSPA
> 2023-11-05 20:43:35.169 RXSPA
> 2023-11-05 20:43:31.461 RXSPA
> 2023-11-05 20:43:27.766 RXSPA
> 2023-11-05 20:41:30.108 RXSPA
v 2023-11-05 20:41:26.431 RXSPA

Fields
  httpcode      201
  message       Inserting on table: spa.etc - http message: <Data inserted successfully> - on inserting frame
  level         DEBUG
```

En este caso “*Inserting on table: spa.etc - http message: <Data inserted successfully> - on inserting frame*” indica que la operación de inserción de datos en la tabla "spa.etc" se realizó con éxito.

- Server - Debug logs

Enfocado en la depuración del servidor, este panel ofrece una visión detallada de eventos que permiten realizar un análisis del código y la lógica de ejecución en el servidor del SSPA.

```
Server - Debug log
> 2023-11-11 19:10:48.534 SERVER
> 2023-11-11 18:21:57.214 SERVER
> 2023-11-11 18:20:31.303 SERVER
> 2023-11-10 17:39:50.025 SERVER
> 2023-11-10 17:38:29.125 SERVER
> 2023-11-09 16:45:48.541 SERVER
> 2023-11-09 16:37:08.533 SERVER
> 2023-11-09 16:36:09.137 SERVER
> 2023-10-26 23:05:50.539 SERVER
> 2023-10-26 22:59:09.072 SERVER
> 2023-10-26 22:58:57.317 SERVER
> 2023-10-26 22:53:42.716 SERVER
> 2023-10-26 22:50:19.408 SERVER
> 2023-10-26 22:47:15.733 SERVER
> 2023-10-26 22:45:01.646 SERVER
> 2023-10-09 19:08:10.225 SERVER
> 2023-10-09 19:05:49.629 SERVER
> 2023-10-09 18:21:05.438 SERVER
> 2023-10-06 22:29:45.624 SERVER
```

A continuación, se pueden observar algunos de los registros de depuración detectados:

```
Server - Debug log
v 2023-11-11 19:10:48.534 SERVER

Fields
  httpcode      404
  message       test new frame
  level         DEBUG

> 2023-11-11 18:21:57.214 SERVER
> 2023-11-11 18:20:31.303 SERVER
> 2023-11-10 17:39:50.025 SERVER
> 2023-11-10 17:38:29.125 SERVER
> 2023-11-09 16:45:48.541 SERVER
```

En este caso “test new frame” indica la generación de un registro de depuración en el cual se está probando o verificando un nuevo marco de datos o trama.



```
Server - Debug log
> 2023-11-11 19:10:48.534 SERVER
> 2023-11-11 18:21:57.214 SERVER
> 2023-11-11 18:20:31.303 SERVER
> 2023-11-10 17:39:50.025 SERVER
> 2023-11-10 17:38:29.125 SERVER
v 2023-11-09 16:45:48.541 SERVER

Fields
||| httpcode          201
||| message          Testing insert on logs
||| level            DEBUG
```

En este caso “Testing insert on logs” indica que se está llevando a cabo una prueba de inserción en registros (logs) en el servidor.

Information logs

Esta sección presenta dos paneles, cada uno destinado a mostrar registros informativos provenientes de distintas fuentes del sistema SPA.

```
Information logs
RXSPA - Information log
> 2024-01-08 09:45:43.581 RXSPA
> 2024-01-08 09:45:40.858 RXSPA
> 2024-01-08 09:45:31.058 RXSPA
> 2024-01-08 09:45:26.465 RXSPA
> 2024-01-08 08:45:40.069 RXSPA
> 2024-01-08 08:45:38.491 RXSPA
> 2024-01-08 08:45:25.678 RXSPA
> 2024-01-08 07:45:40.153 RXSPA
> 2024-01-08 07:45:38.489 RXSPA
> 2024-01-08 07:45:25.878 RXSPA
> 2024-01-08 06:45:43.824 RXSPA
> 2024-01-08 06:45:42.254 RXSPA

Server - Information logs
> 2023-10-06 22:38:17.775 SERVER
> 2023-10-06 22:25:57.625 SERVER
```

- RXSPA - Information logs

Este panel se dedica a la presentación de registros informativos específicos del módulo RXSPA. Los registros aquí presentes brindan insights sobre eventos normales y operaciones exitosas del RXSPA.

```
RXSPA - Information log
> 2024-01-08 09:45:43.581 RXSPA
> 2024-01-08 09:45:40.858 RXSPA
> 2024-01-08 09:45:31.058 RXSPA
> 2024-01-08 09:45:26.465 RXSPA
> 2024-01-08 08:45:40.069 RXSPA
> 2024-01-08 08:45:38.491 RXSPA
> 2024-01-08 08:45:25.678 RXSPA
> 2024-01-08 07:45:40.153 RXSPA
> 2024-01-08 07:45:38.489 RXSPA
> 2024-01-08 07:45:25.878 RXSPA
> 2024-01-08 06:45:43.824 RXSPA
> 2024-01-08 06:45:42.254 RXSPA
> 2024-01-08 06:45:31.258 RXSPA
> 2024-01-08 06:45:29.273 RXSPA
> 2024-01-08 05:45:39.087 RXSPA
> 2024-01-08 05:45:37.453 RXSPA
> 2024-01-08 05:45:25.423 RXSPA
> 2024-01-08 04:45:39.267 RXSPA
> 2024-01-08 04:45:37.675 RXSPA
> 2024-01-08 04:45:25.481 RXSPA
> 2024-01-08 03:45:39.535 RXSPA
> 2024-01-08 03:45:37.916 RXSPA
> 2024-01-08 03:45:25.720 RXSPA
> 2024-01-08 02:45:39.883 RXSPA
> 2024-01-08 02:45:37.973 RXSPA
```




A continuación, podemos ver algunos registros de información destacados:

```
RXSPA - Information log
> 2024-01-08 09:45:43.581 RXSPA
> 2024-01-08 09:45:40.858 RXSPA
> 2024-01-08 09:45:31.058 RXSPA
> 2024-01-08 09:45:26.465 RXSPA
> 2024-01-08 08:45:40.069 RXSPA
> 2024-01-08 08:45:38.491 RXSPA
> 2024-01-08 08:45:25.678 RXSPA
> 2024-01-08 07:45:40.153 RXSPA
> 2024-01-08 07:45:38.489 RXSPA
v 2024-01-08 07:45:25.878 RXSPA

Fields
  httpcode      0
  message       Sending POLL to SPA.
  level         INFORMATION
```

En este caso “Sending POLL to SPA” indica que el módulo RXSPA le esta enviando una solicitud de datos al módulo ADCSPA.

```
RXSPA - Information log
> 2024-01-08 09:45:43.581 RXSPA
> 2024-01-08 09:45:40.858 RXSPA
> 2024-01-08 09:45:31.058 RXSPA
> 2024-01-08 09:45:26.465 RXSPA
> 2024-01-08 08:45:40.069 RXSPA
> 2024-01-08 08:45:38.491 RXSPA
> 2024-01-08 08:45:25.678 RXSPA
> 2024-01-08 07:45:40.153 RXSPA
> 2024-01-08 07:45:38.489 RXSPA
v 2024-01-08 07:45:38.489 RXSPA

Fields
  httpcode      0
  message       Received frame data from SPA: '>POLL+pl:0.00;ws:4;wd:160;l:0;h:64.80;r:0;t:20;pr:907;wh:1955.6882<'
  level         INFORMATION
```

En este caso “Received frame data from SPA: ...” indica que el módulo ADCSPA ha respondido con la información solicitada anteriormente por el RXSPA.

```
RXSPA - Information log
v 2024-01-08 09:45:43.581 RXSPA

Fields
  httpcode      201
  message       Inserting on table: spa.weatherstation - http message: <Data inserted successfully> - on inserting frame http response code: 201
  on attempts: 1
  level         INFORMATION
```

En este caso “Inserting on table: spa.wetweights - http message: <Data inserted successfully> - on inserting frame http responde code: 201 on attempts: 1” indica que la operación de inserción de datos en la tabla "spa.wetweights" se realizó con éxito.”

- Server - Information logs

Enfocado en la información generada por el servidor, este panel ofrece registros que proporcionan detalles sobre el funcionamiento general del servidor.

```
Server - Information logs
> 2023-10-06 22:30:17.775 SERVER
> 2023-10-06 22:25:57.625 SERVER
```



5.1.10 Instalación

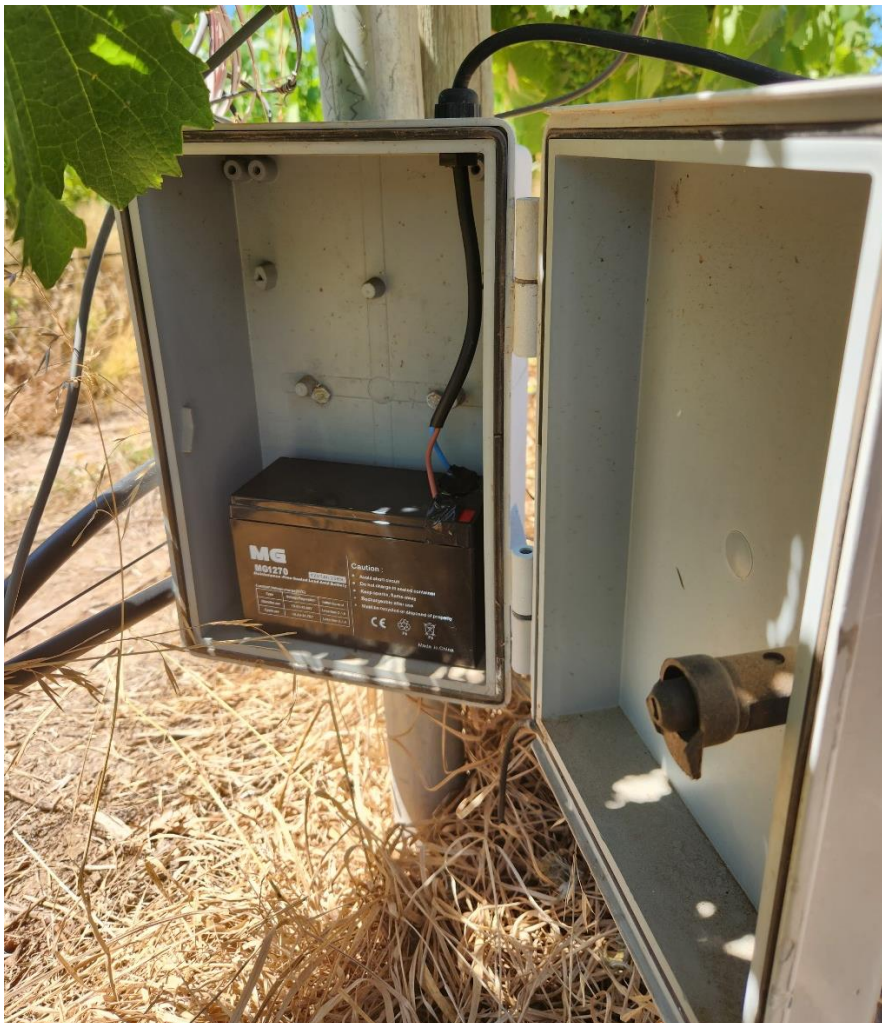
En esta sección se pretende dar a conocer como fue el proceso de instalación del sistema SPA en el Instituto Nacional de Tecnología Agropecuaria ubicado en Luján de Cuyo, Mendoza. Para ello se colocarán breves descripciones acompañadas de fotografías representativas.

5.1.10.1 Equipo de energía solar

Este componente es esencial para garantizar la autonomía energética del sistema. En este caso, se llevaron a cabo las siguientes etapas durante la instalación:

El panel solar quedó posicionado estratégicamente en una ubicación que maximiza la exposición a la radiación solar, considerando la latitud y orientación adecuadas en el hemisferio sur. El regulador de carga se integró cuidadosamente dentro del gabinete principal colocado en el campo para controlar la carga de las baterías y evitar sobrecargas, garantizando la eficiencia y vida útil de la batería. La batería MG1270 fue instalada como el componente principal de almacenamiento de energía. Su ubicación y conexión se realizaron en un gabinete propio por razones de seguridad y espacio. Se realizaron ajustes y configuraciones necesarias para adaptar el sistema de energía solar a las condiciones específicas del entorno.

A continuación, se presentan algunas imágenes representativas del proceso de instalación del equipo de energía solar en el INTA:

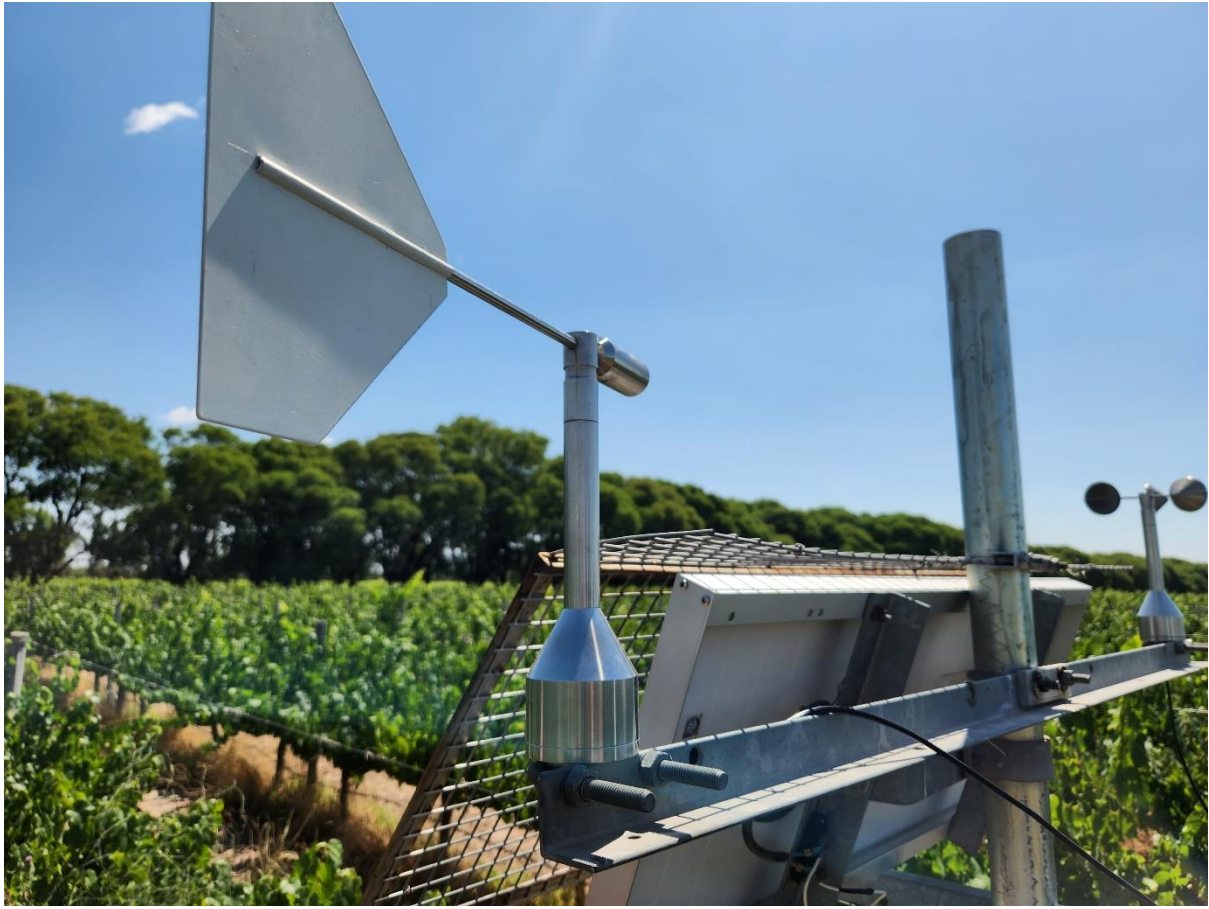




5.1.10.2 Sensores

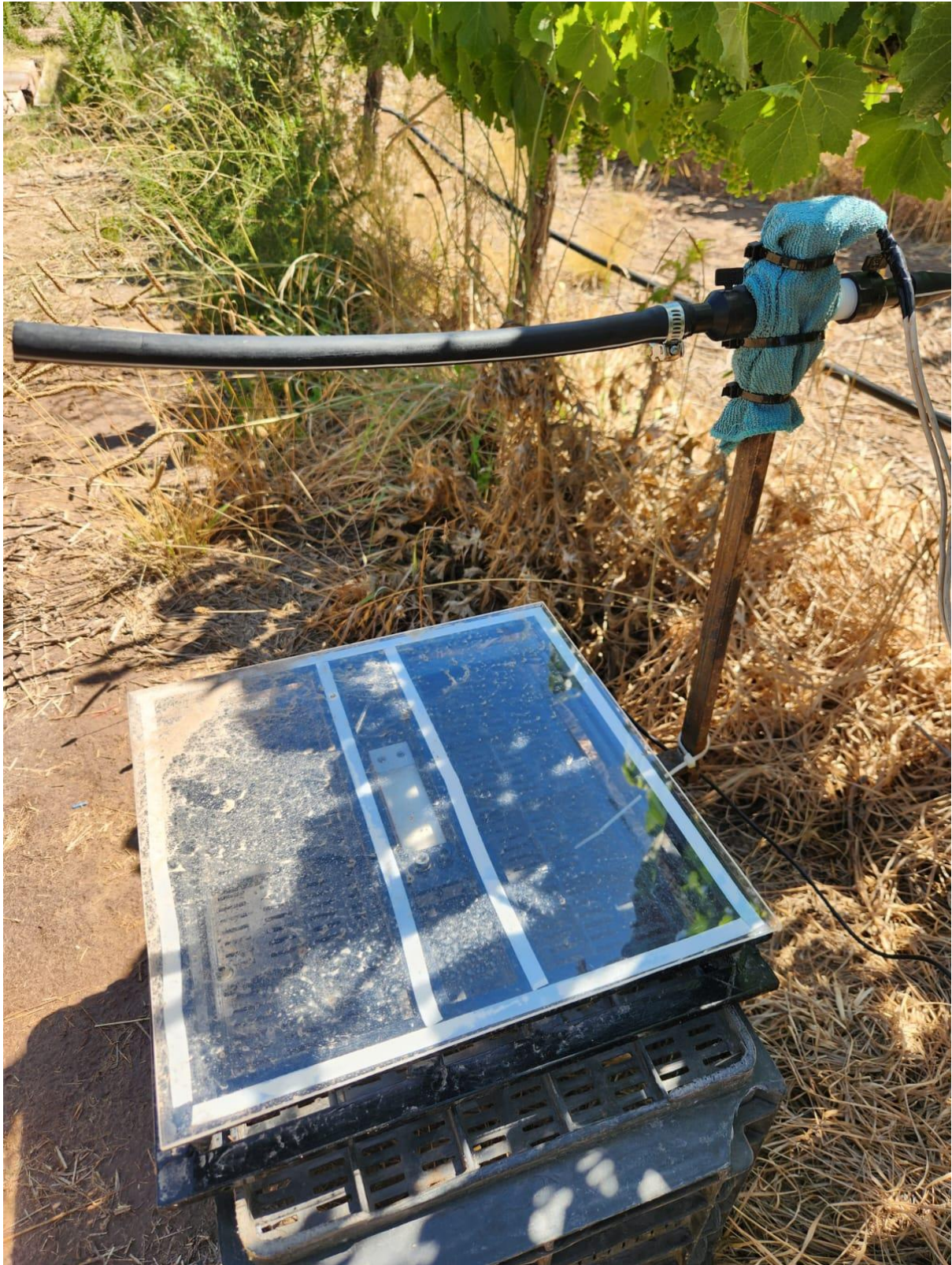
Cada sensor fue ubicado estratégicamente para maximizar su efectividad y precisión. Los sensores de radiación solar, temperatura ambiente, humedad relativa, precipitación, presión atmosférica, humedad de hoja, dirección y velocidad del viento fueron instalados y orientados según las mejores prácticas. Todos los sensores se montaron sobre una misma estructura metálica a excepción del pluviómetro que cuenta con una estructura propia. Por su parte, el lisímetro se ubicó en proximidad al área de cultivo de interés.

A continuación, se presentan algunas imágenes representativas del proceso de instalación de los sensores en el INTA.







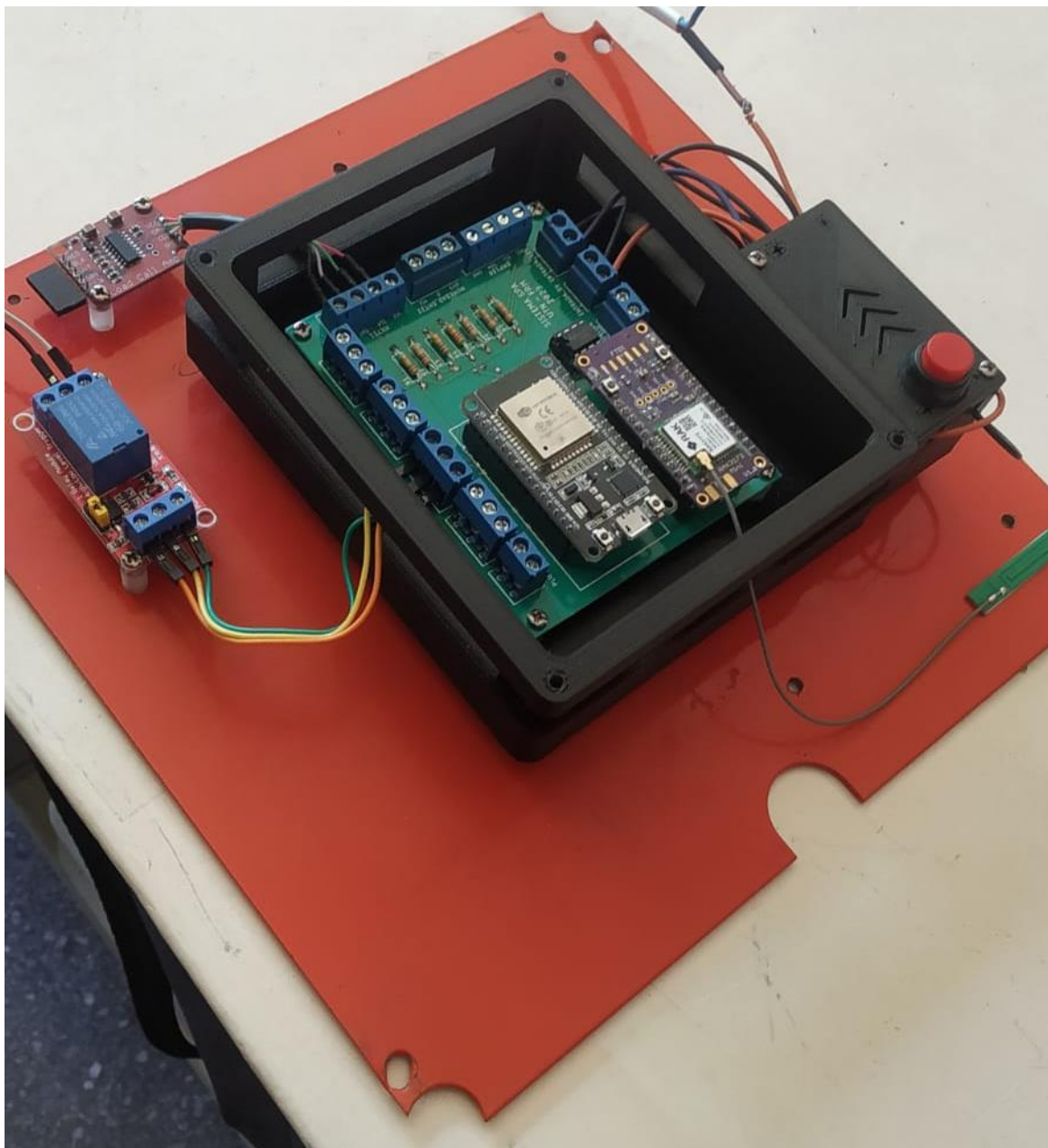


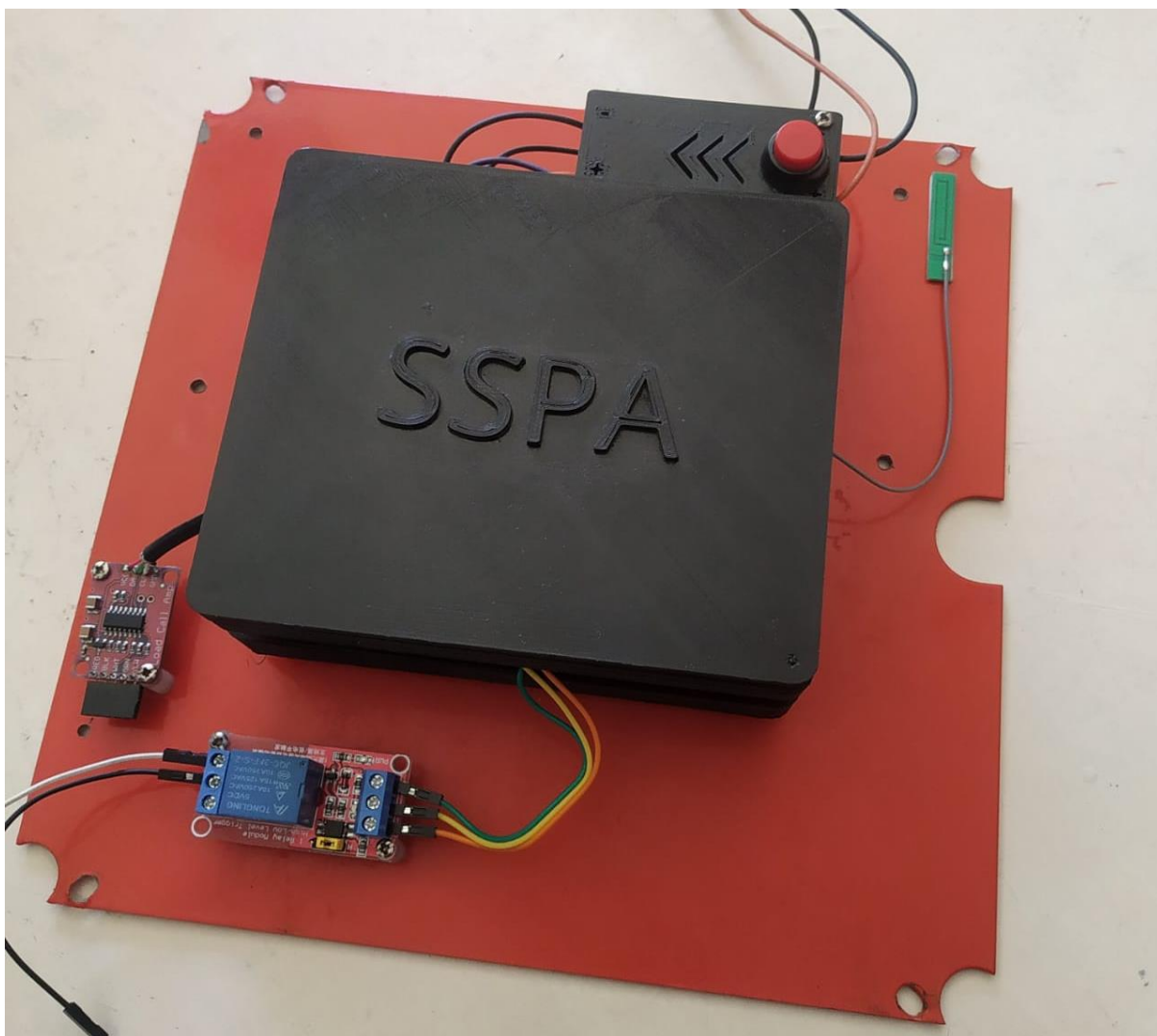


5.1.10.3 Módulo ADCSPA

El módulo ADCSPA se instaló dentro del gabinete donde también se encuentra el regulador de carga del equipo de energía solar. Para su colocación se utilizó una placa metálica, la cual fue debidamente acondicionada para poder atornillar el módulo propiamente dicho. A la hora de la instalación se tuvieron en cuenta aspectos como el espacio, el acceso a conexión/desconexión de sensores y de energía.

A continuación, se presentan algunas imágenes representativas de lo comentado:



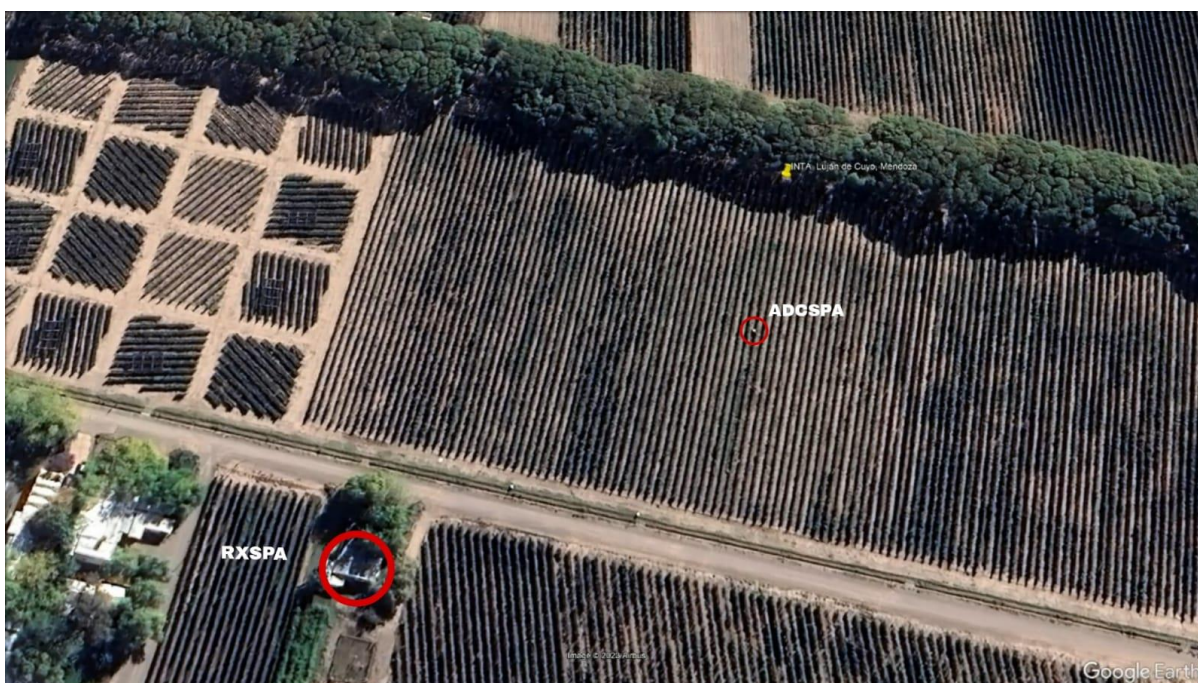


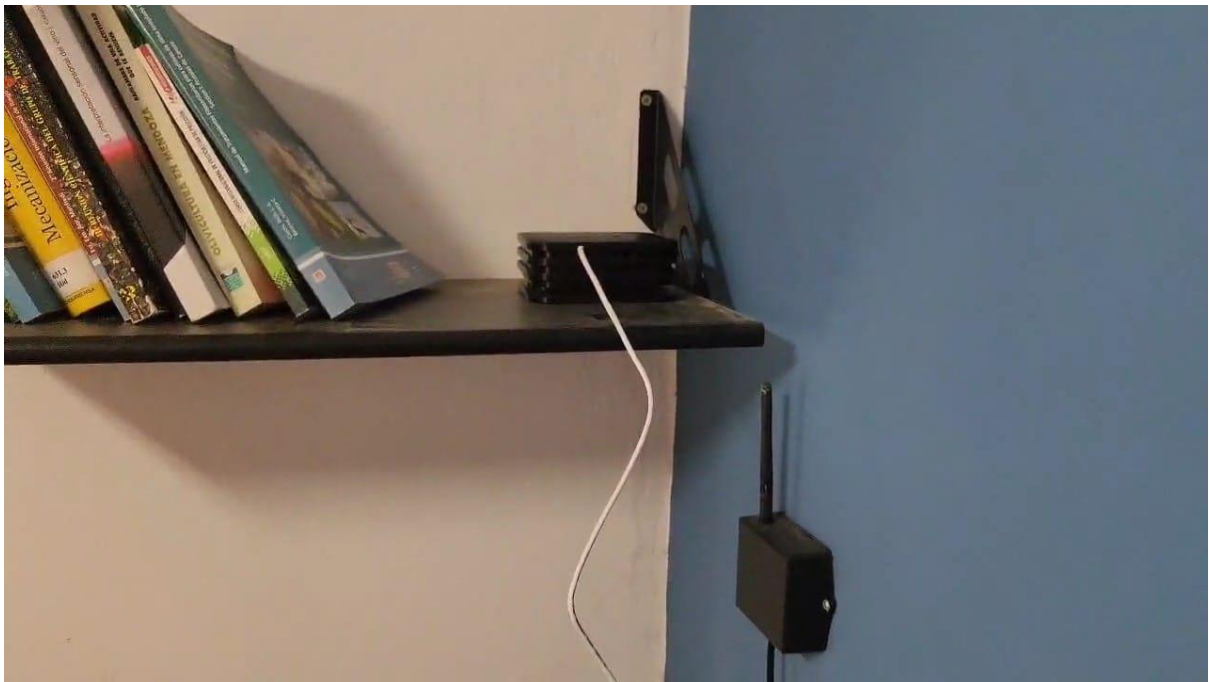


5.1.10.4 Módulo RXSPA

El modulo RXSPA opera desde el edificio de geomatica del Instituto Nacional de Tecnología Agropecuaria en Lujan de Cuyo, Mendoza. Se encuentra en una de las oficinas de dicho edificio, conectado directamente al suministro eléctrico de la red. Su ubicación se decidió en base a las pruebas de transmisión realizadas, disponibilidad energética y disponibilidad de conexión a internet.

A continuación, algunas imágenes que ilustran lo comentado anteriormente:







5.2 FACTIBILIDAD ECONÓMICA

Para el análisis de factibilidad económica del Sistema SPA, se considera la venta del producto para evaluar su impacto en el mercado y determinar las ganancias esperadas, así como los tiempos de recuperación de la inversión. Se han calculado de manera estimada los costos asociados a la realización del proyecto.

5.2.1 Costos

En esta sección se presentan los costos asociados a la implementación del Sistema SPA, teniendo en cuenta mano de obra, sensores, microprocesadores, etc. En la siguiente tabla se pueden observar dichos costos:

PRODUCTO	CANTIDAD	COSTO	TOTAL
EQUIPO DE ENERGÍA SOLAR			
Panel solar	1	\$40,00	\$ 40
Regulador de carga	1	\$23,00	\$ 23
Batería 12V (MG1270)	1	\$25,00	\$ 25
Horas de trabajo	2	\$10,00	\$ 20
ADCSPA			
Sensor radiación solar	1	\$4,00	\$ 4
Sensor temperatura	1	\$3,00	\$ 3
Sensor humedad relativa	1	\$11,00	\$ 11
Sensor presión atmosférica	1	\$3,00	\$ 3
Pluviómetro	1	\$500,00	\$ 500
Sensor humedad de hoja	1	\$2,00	\$ 2
Sensor dirección del viento	1	\$1.200,00	\$ 1.200
Sensor velocidad del viento	1	\$1.600,00	\$ 1.600
Lisímetro de pesada	1	\$55,00	\$ 55
ESP32	1	\$28,00	\$ 28
Módulo LoRaWAN RAK3172	1	\$14,00	\$ 14
PCB ADCSPA	1	\$30,00	\$ 30
Carcasa 3D	1	\$14,00	\$ 14
Otros	1	\$4,00	\$ 4
Horas de trabajo	8	\$10,00	\$ 80
RXSPA			
ESP32	1	\$28,00	\$ 28
Módulo LoRaWAN RAK3172	1	\$14,00	\$ 14
PCB RXSPA	1	\$23,00	\$ 23
Carcasa 3D	1	\$10,00	\$ 10
Fuente alimentación 5V	1	\$6,00	\$ 6
Horas de trabajo	5	\$10,00	\$ 50
DB E INTERFAZ GRÁFICA			
Horas de trabajo	19	\$10,00	\$ 190
TOTAL			\$ 3.977

Valores expresados en dólares.



El análisis de factibilidad económica se ha segmentado en dos partes. La primera parte consiste en calcular el Valor Actual Neto (VAN) igual a cero durante un período de 5 años de producción. El objetivo de este análisis inicial es determinar la cantidad de sistemas SPA necesarios para recuperar la inversión inicial al finalizar el quinto año. La segunda parte del análisis abarca un periodo de 5 años, durante el cual la producción se incrementa año tras año según las proyecciones de crecimiento del mercado y demanda potencial.

5.2.2 Flujo de caja para VAN = 0

Para la fijación del precio de venta del sistema se partió del costo total de fabricación y se tomó un margen de ganancia, teniendo presente que el precio final sea competitivo en el mercado. Para el cálculo del **capital de trabajo** se tomó como referencia la cantidad de ventas estimadas del primer año para un flujo de caja de VAN = 0. Mediante un proceso iterativo se llegó a los valores de ventas que se observan en la tabla.

INGRESO MENSUAL AÑO UNO	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Unidades vendidas	0	0	1	0	0	0	0	0	0	0	0	0
Total de ventas	\$0,00	\$0,00	\$5.377,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00
Cobro al momento de la compra	60%	\$0,00	\$0,00	\$3.226,20	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00
Cobro 30 días después de la compra	40%	\$0,00	\$0,00	\$0,00	\$2.150,80	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00
PRODUCCIÓN												
Stock	0	0	0	0	0	0	0	0	0	0	0	0
Ventas	0	0	1	0	0	0	0	0	0	0	0	0
Producción Mensual	0	0	1	0	0	0	0	0	0	0	0	0
EGRESOS												
Costos Fijos	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00
Costos de Fabricación	\$0,00	\$0,00	\$3.977,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00	\$0,00
Egreso total	\$380,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00
CAPITAL DE TRABAJO												
Ingresos	0	0	3226,2	2150,8	0	0	0	0	0	0	0	0
Egresos	\$380,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00
Acumulado	-\$380,00	-\$760,00	-\$1.890,80	-\$120,00	-\$500,00	-\$880,00	-\$1.260,00	-\$1.640,00	-\$2.020,00	-\$2.400,00	-\$2.780,00	-\$3.160,00
Máximo Acumulado	-\$3.160,00											
VERIFICACIÓN DEL CAPITAL DE TRABAJO												
Ingresos	0	\$3.160,00	0	3226,2	2150,8	0	0	0	0	0	0	0
Egresos	0	380	380	4357	380	380	380	380	380	380	380	380
Acumulado	0	\$2.780,00	\$2.400,00	\$1.269,20	\$3.040,00	\$2.660,00	\$2.280,00	\$1.900,00	\$1.520,00	\$1.140,00	\$760,00	\$380,00

CAPITAL DE TRABAJO SIN MARGEN		\$3.160,00
CAPITAL DE TRABAJO CON MARGEN	1,2	\$3.792,00

El flujo de caja se presenta en la siguiente tabla. También se realizaron los cálculos de depreciaciones y amortizaciones correspondientes.



SSPA

Flujo de caja	0	1	2	3	4	5
Unidades vendidas		1	1	1	2	2
Ingreso		\$ 5.377	\$ 5.377	\$ 5.377	\$ 10.754	\$ 10.754
Precio unitario		\$ 5.377	\$ 5.377	\$ 5.377	\$ 5.377	\$ 5.377
Costos fijos						
Total costos fijos		\$ 380	\$ 380	\$ 380	\$ 380	\$ 380
Costos variables						
Costo de venta del producto		\$ 3.977	\$ 3.977	\$ 3.977	\$ 7.954	\$ 7.954
Herramientas e instrumentos	\$ 350	\$ -	\$ -	\$ -	\$ -	\$ -
Equipos electrónicos	\$ 1.500	\$ -	\$ -	\$ 1.500	\$ -	\$ -
Costos variables totales		\$ 3.977	\$ 3.977	\$ 5.477	\$ 7.954	\$ 7.954
Depreciación herramientas		\$ 570	\$ 570	\$ 570	\$ 570	\$ 570
Utilidad antes de impuestos		\$ 450	\$ 450	\$ -1.050	\$ 1.850	\$ 1.850
Impuesto a las ganancias (35%)		\$ 158	\$ 158	\$ -368	\$ 648	\$ 648
Utilidad después de impuestos		\$ 293	\$ 293	\$ -683	\$ 1.203	\$ 1.203
Depreciación herramientas		\$ 570	\$ 570	\$ 570	\$ 570	\$ 570
Inversión inicial						
Herramientas y equipos electrónicos	\$ -1.850					
Capital de trabajo	\$ -3.792					
Recuperación capital de trabajo						\$ 3.792
Valor de desecho						\$ -
Flujo del proyecto	\$ -5.642	\$ 863	\$ 863	\$ -113	\$ 1.773	\$ 5.565
Tasa de interés		12%				
VAN	\$	19				
TIR		12%				

TABLA DE DEPRECIACIONES						
DEPRECIACIÓN HERRAMIENTAS						
Vida útil	5					
ITEM	0	1	2	3	4	5
Valor de adquisición	\$ 350					
Gasto depreciación		\$ 70	\$ 70	\$ 70	\$ 70	\$ 70
Depreciación acumulada		\$ 70	\$ 140	\$ 210	\$ 280	\$ 350
Valor en libros		\$ 280	\$ 210	\$ 140	\$ 70	\$ -
EQUIPOS ELECTRÓNICOS						
Vida útil	3					
ITEM	0	1	2	3	4	5
Valor de adquisición	\$ 1.500					
Gasto depreciación		\$ 500	\$ 500	\$ 500		
Depreciación acumulada		\$ 500	\$ 1.000	\$ 1.500		
Valor en libros		\$ 1.000	\$ 500	\$ -		
EQUIPOS ELECTRÓNICOS						
Vida útil	3					
ITEM	0	1	2	3	4	5
Valor de adquisición				\$ 1.500		
Gasto depreciación					\$ 500	\$ 500
Depreciación acumulada					\$ 500	\$ 1.000
Valor en libros					\$ 1.000	\$ 500
TOTAL DE DEPRECIACIONES		\$ 570	\$ 570	\$ 570	\$ 570	\$ 570



5.2.3 Flujo de caja para VAN > 0

En este segundo análisis, en lugar de determinar la cantidad de dispositivos fabricados en cada año para lograr un VAN determinado, estas cantidades fueron determinadas en base a las posibilidades de mercado y producción. Para el cálculo realizado en VAN > 0, se han supuesto las siguientes ventas en los próximos 5 años: año 1: 3 sistemas, año 2: 3 sistemas, año 3: 3 sistemas, año 4: 4 sistemas, año 5: 5 sistemas.

Estas ventas se basan en proyecciones de crecimiento del mercado y demanda potencial del sistema SPA, considerando el aumento del conocimiento y la conciencia sobre la eficiencia en el uso del agua y la necesidad de monitorear y optimizar los procesos agrícolas. Además, se espera que la expansión de la agricultura de precisión y la adopción de tecnologías sostenibles impulsen la aceptación y adquisición de este tipo de sistemas.

Flujo de caja	0	1	2	3	4	5
Unidades vendidas		3	3	3	4	5
Ingreso		\$ 16.131	\$ 16.131	\$ 16.131	\$ 21.508	\$ 26.885
Precio unitario		\$ 5.377	\$ 5.377	\$ 5.377	\$ 5.377	\$ 5.377
Costos fijos						
Total costos fijos		\$ 380	\$ 380	\$ 380	\$ 380	\$ 380
Costos variables						
Costo de venta del producto		\$ 11.931	\$ 11.931	\$ 11.931	\$ 15.908	\$ 19.885
Herramientas e instrumentos	\$ 350	\$ -	\$ -	\$ -	\$ -	\$ -
Equipos electrónicos	\$ 1.500	\$ -	\$ -	\$ 1.500	\$ -	\$ -
Costos variables totales		\$ 11.931	\$ 11.931	\$ 13.431	\$ 15.908	\$ 19.885
Depreciación herramientas		\$ 570	\$ 570	\$ 570	\$ 570	\$ 570
Utilidad antes de impuestos		\$ 3.250	\$ 3.250	\$ 1.750	\$ 4.650	\$ 6.050
Impuesto a las ganancias (35%)		\$ 1.138	\$ 1.138	\$ 613	\$ 1.628	\$ 2.118
Utilidad después de impuestos		\$ 2.113	\$ 2.113	\$ 1.138	\$ 3.023	\$ 3.933
Depreciación herramientas		\$ 570	\$ 570	\$ 570	\$ 570	\$ 570
Inversión inicial						
Herramientas y equipos electrónicos	\$ -1.850					
Capital de trabajo	\$ -1.813					
Recuperación capital de trabajo						\$ 1.813
Valor de desecho						\$ -
Flujo del proyecto	\$ -3.663	\$ 2.683	\$ 2.683	\$ 1.708	\$ 3.593	\$ 6.315

El capital de trabajo en este caso resulta de la siguiente manera:

INGRESO MENSUAL AÑO UNO	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Unidades vendidas	0	1	0	1	0	0	0	1	0	0	0	0
Total de ventas	\$0,00	\$5.377,00	\$0,00	\$5.377,00	\$0,00	\$0,00	\$0,00	\$5.377,00	\$0,00	\$0,00	\$0,00	\$0,00
Cobro al momento de la compra	60%	\$0,00	\$3.226,20	\$0,00	\$3.226,20	\$0,00	\$0,00	\$3.226,20	\$0,00	\$0,00	\$0,00	\$0,00
Cobro 30 días después de la compra	40%	\$0,00	\$0,00	\$2.150,80	\$0,00	\$2.150,80	\$0,00	\$0,00	\$0,00	\$2.150,80	\$0,00	\$0,00
Ingreso total	0	3226,2	2150,8	3226,2	2150,8	0	0	3226,2	2150,8	0	0	0
PRODUCCIÓN												
Stock	0	0	0	0	0	0	0	0	0	0	0	0
Ventas	0	1	0	1	0	0	0	1	0	0	0	0
Producción Mensual	0	1	0	1	0	0	0	1	0	0	0	0
EGRESOS												
Costos Fijos	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00	\$380,00
Costos de Fabricación	\$0,00	\$3.977,00	\$0,00	\$3.977,00	\$0,00	\$0,00	\$0,00	\$3.977,00	\$0,00	\$0,00	\$0,00	\$0,00
Egreso total	\$380,00	\$4.357,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$380,00
CAPITAL DE TRABAJO												
Ingresos	0	3226,2	2150,8	3226,2	2150,8	0	0	3226,2	2150,8	0	0	0
Egresos	\$380,00	\$4.357,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$4.357,00	\$380,00	\$380,00	\$380,00	\$380,00
Acumulado	-\$380,00	-\$1.510,80	\$260,00	-\$870,80	\$900,00	\$520,00	\$140,00	-\$990,80	\$780,00	\$400,00	\$20,00	-\$360,00
Máximo Acumulado	-\$1.510,80											
VERIFICACIÓN DEL CAPITAL DE TRABAJO												
Ingresos	0	\$1.510,80	3226,2	2150,8	3226,2	2150,8	0	0	3226,2	2150,8	0	0
Egresos	0	380	4357	380	4357	380	380	380	4357	380	380	380
Acumulado	0	\$1.130,80	\$0,00	\$1.770,80	\$640,00	\$2.410,80	\$2.030,80	\$1.650,80	\$520,00	\$2.290,80	\$1.910,80	\$1.530,80



CAPITAL DE TRABAJO SIN MARGEN		\$1.510,80
CAPITAL DE TRABAJO CON MARGEN	1,2	\$1.812,96

5.2.3.1 Aproximación al valor actual neto

Tasa de interés	12%
VAN	\$ 7.953

5.2.3.2 Tasa interna de retorno

TIR	71%
------------	------------

5.2.3.3 Payback o plazo de recuperación

PRI			
PERIODO	FLUJO	ACUMULADO	
0	\$ -3.663		
1	\$ 2.683	\$ 2.683	\$ -980
2	\$ 2.683	\$ 5.365	\$ 1.702
3	\$ 1.708	\$ 7.073	\$ 3.410
4	\$ 3.593	\$ 10.665	\$ 7.002
5	\$ 6.315	\$ 16.980	\$ 13.318

PRI	1,37 años
------------	------------------

5.2.4 Productos y servicios de otros fabricantes

Esta sección se centra en la comparación del sistema SPA con otros productos disponibles en el mercado que pueden ofrecer funcionalidades similares o relacionadas. Aunque no existe un producto exactamente igual al sistema SPA, se pueden realizar comparaciones con productos como la Estación Meteorológica EM-101-PAR y la Estación Meteorológica Inalámbrica Portátil Pegasus EP 201.

Una de las diferencias clave es la ausencia de un lisímetro de pesada en los productos alternativos. El lisímetro de pesada es una característica distintiva del sistema SPA que permite una medición precisa del balance de agua en el suelo, lo que facilita una gestión hídrica más efectiva y precisa. Otra diferencia importante radica en la capacidad de riego automático del sistema SPA, una función que no está presente en los productos comparativos. Además, los productos alternativos ofrecen sensores de menor calidad y en menor cantidad comparados con los del Sistema SPA, los cuales le proporcionan a los agricultores una visión más completa y



detallada de las condiciones ambientales y del suelo, lo que les permite tomar decisiones más informadas y precisas en la gestión de sus cultivos.

- **Estacion Meteorologica EM-101-PAR**

[Estacion Meteorologica EM-101-PAR - Don Agro](#)

Don Agro COSAS PARA EL CAMPO

Buscar productos y categorías...

(0291) 485 0410 / (02926) 421527
(2926) 546489
Lun a Vie: 08 a 16hs - Sáb: 08 a 12hs

OFERTAS CONTACTO

Enviar a Seleccione Ubicacion CARRITO 0 INGRESAR

Estacion Meteorologica EM-101-PAR

Precio en 1 pago
\$ 5.310.340,98 IVA Inc.

1 cuotas de \$ 5.310.340,98

Mas formas de pago >

Envio a todo el pais.
¡Despachos en el dia!

Seleccione su Localidad >

Agregar

1

AGREGAR AL CARRITO

A un tipo de cambio oficial de 853,50 pesos por dólar, el precio es de 6222 dólares estadounidenses. En comparación con el sistema SPA, este carece de un lisímetro de pesada, no cuenta con funcionalidades de riego automático, emplea sensores de calidad inferior y es mas caro que el Sistema SPA. (US\$ 5377)

- **Estacion Meteorologica Inalambrica Portatil Pegasus Ep 201**

También puede interesarte: estacion meteorologica

Volver al listado | Herramientas > Testers y Equipos de Medicion > Medidores Ambientales > Estaciones Meteorológicas

Vender uno igual | Compartir

Nuevo

Estacion Meteorologica Inalambrica Portatil Pegasus Ep 201

\$ 5.064.538³³
en 1 cuotas de \$ 5.064.538³³

Ver los medios de pago

Llega gratis entre el martes y el viernes
23/feb

Más formas de entrega

¡Última disponible!

Comprar ahora

Agregar al carrito

Vendido por **TECMES INSTRUMENTOS-ESPECIALES**



A un tipo de cambio oficial de 853,50 pesos por dólar, el precio del producto es de 5934 dólares estadounidenses. Este producto, al igual que el anteriormente mencionado, carece de un lisímetro de pesada, no ofrece funcionalidades de riego automático y cuenta con una cantidad reducida de sensores en comparación con el sistema SPA.

6 CONCLUSIONES Y ANEXOS

El desarrollo y análisis del Sistema SPA ha revelado una serie de hallazgos que destacan su potencial para mejorar la eficiencia y la productividad en la agricultura. A continuación, se presentan las principales conclusiones derivadas de este proyecto:

El Sistema SPA ofrece una solución integral para monitorear las condiciones del suelo, la planta y la atmósfera. Con la capacidad de medir la Evapotranspiración del cultivo (ETc) a través del lisímetro de pesada, el sistema permite una gestión precisa del riego, adaptándolo a las necesidades específicas de los cultivos y reduciendo el desperdicio de agua.

Al proporcionar datos detallados sobre las condiciones ambientales y el estado de los cultivos, el SSPA permite tomar decisiones informadas y oportunas para maximizar la productividad agrícola. La automatización del riego basada en la ETc optimiza el uso del agua y mejora el rendimiento de los cultivos.

Una de las ventajas clave es su capacidad de otorgar acceso a los datos de manera remota. Esto brinda a los agricultores la libertad de supervisar sus cultivos desde cualquier dispositivo inteligente con conexión a internet. Esta flexibilidad mejora la eficiencia operativa y permite una respuesta rápida a cualquier cambio en las condiciones agrícolas.

A través del análisis económico realizado, se ha demostrado que el modelo de negocio de venta del sistema es viable. Si bien el análisis se centró en el modelo de negocio de venta del sistema, el SSPA presenta oportunidades adicionales de expansión y diversificación. Con su capacidad para recopilar datos sobre el suelo y las condiciones ambientales, el sistema podría ser utilizado para ofrecer servicios de consultoría agrícola y análisis de datos, lo que ampliaría su alcance y su valor en el mercado.

- **Anexo A. Bases teóricas del sistema SPA.**
- **Anexo B. Metodología de trabajo utilizada en SSPA.**
- **Anexo C. Firmware SSPA**
- **Anexo D. Software SSPA.**



7 BIBLIOGRAFÍAS Y REFERENCIAS BIBLIOGRÁFICAS

- *Notion*. (s.f.). Recuperado de <https://www.notion.so>
- *GitHub*. (s.f.). Recuperado de <https://github.com>
- *Solartec*. (s.f.). Panel solar KS10T. Recuperado de [SOLARTEC KS10T v11.xls](#)
- *Smartek Solutions*. (s.f.). Regulador de carga solar. Recuperado de [PWM Solar Charger Controller User Manual \(electrosistemas.com.ar\)](#)
- *TECMES*. (s.f.). Sensor de velocidad del viento TS231. Recuperado de [Sensor de velocidad de viento TS231 - TECMES](#)
- *TECMES*. (s.f.). Sensor de dirección del viento TS232. Recuperado de [Sensor de dirección del viento TS232 - TECMES](#)
- *TECMES*. (s.f.). Pluviómetro TS221. Recuperado de [Sensor de precipitación TS221 - TECMES](#)
- *TECMES*. (s.f.). Sensor de humedad relativa TS251. Recuperado de <https://www.tecmes.com/wp-content/uploads/2019/10/MANUAL-TS251H.pdf>
- *GY-68 BMP180* Sensor de presión barométrica y temperatura. (s.f.). Recuperado de [Gy-68 Bmp180 Sensor De Presion Barometrica Y Temperatura Itytag - IT&T Argentina S.A.](#)
- *FAO*. (2006). Evapotranspiración del cultivo: Guías para la determinación de los requerimientos de agua de los cultivos (Estudio FAO Riego y Drenaje No. 56). Recuperado de [01-Contents \(fao.org\)](#)
- *Flintec*. (s.f.). ¿Qué es una celda de carga y cómo funciona? Recuperado de [Celdas de carga: Guía de funcionamiento \(flintec.com\)](#)
- *SparkFun Electronics*. (s.f.). HX711. Recuperado de [Microsoft Word - hx71_english.doc \(sparkfun.com\)](#)
- *Semtech Corporation*. (s.f.). Academy for LoRaWAN®. Recuperado de [Academy for LoRaWAN: All courses \(semtech.com\)](#)
- *RAK Wireless*. (s.f.). RAK3172. Recuperado de [RAK3272S Quick Start Guide | RAKwireless Documentation Center](#)
- *FL0*. (s.f.). Documentación FL0. Recuperado de [FL0 Documentation](#)
- *JetBrains*. (s.f.). DataGrip. Recuperado de [Getting started | DataGrip Documentation \(jetbrains.com\)](#)
- *Render*. (s.f.). Documentación de la plataforma. Recuperado de [Docs + Quickstarts | Render Docs](#)
- *Grafana Labs*. (s.f.). Grafana. Recuperado de [Grafana: The open observability platform | Grafana Labs](#)