



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE INGENIERÍA ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO IOT PARA LA MONITORIZACIÓN DE
PARÁMETROS AMBIENTALES Y DE NUTRIENTES DEL SUELO APLICADOS A
CULTIVOS DE CACAO**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica

AUTORES:

ÁNGELO DAVID BRIONES RODRÍGUEZ

JOSELINE VANESSA SALMERÓN FLORES

TUTOR:

ING. DIEGO FREIRE QUIROGA MSc.

Guayaquil – Ecuador

2025

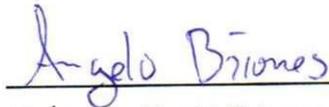
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL
TRABAJO DE TITULACIÓN**

Nosotros, Ángelo David Briones Rodríguez con documento de identificación N° 0925014086 y Joseline Vanessa Salmerón Flores con documento de identificación N° 0950762971 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

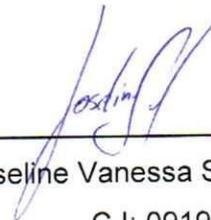
Guayaquil, 5 de marzo de 2025

Atentamente,



(f) Ángelo David Briones Rodríguez

C.I.: 0925014086



(f) Joseline Vanessa Salmerón Flores

C.I.: 0919349720

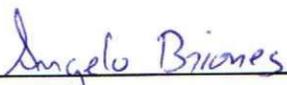
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN
A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Ángelo David Briones Rodríguez con documento de identificación No. 0925014086 y Joseline Vanessa Salmerón Flores con documento de identificación No. 0950762971, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del Proyecto técnico: "Diseño E Implementación De Prototipo IoT Para La Monitorización De Parámetros Ambientales Y De Nutrientes Del Suelo Aplicados A Cultivos De Cacao", el cual ha sido desarrollado para optar el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer y usar los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Guayaquil, 5 de marzo de 2025

Atentamente,



(f) Ángelo David Briones Rodríguez
C.I: 0925014086



(f) Joseline Vanessa Salmerón Flores
C.I: 0919349720

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, MSc. Diego Roberto Freire Quiroga con documento de identificación No. 0917208084, docente de la Universidad Politécnica Salesiana , declaro que bajo mi tutoría fue desarrollado el trabajo de titulación "Diseño E Implementación De Prototipo IoT Para La Monitorización De Parámetros Ambientales Y De Nutrientes Del Suelo Aplicados A Cultivos De Cacao", realizado por Ángel David Briones Rodríguez con documento de identificación N° 0925014086 y por Joseline Vanessa Salmerón Flores con documento de identificación N° 0950762971, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 5 de marzo de 2025

Atentamente,



MSc. Diego Roberto Freire Quiroga

C.I: 0917208084

DEDICATORIA

Este uno de los logros más importante en mi vida se lo dedico a Dios y a mi madre por ser los impulsores en cumplir mi meta y por enseñarme siempre que con paciencia, esfuerzo y responsabilidad puedo cumplir mis metas, una de estas es ser un profesional de bien y siempre enfrentar los obstáculos que se me han presentado con la cabeza en alto.

Ángelo David Briones Rodríguez

DEDICATORIA

A mis padres, por su amor incondicional y su apoyo constante, por sus sabios consejos profesional que me permitieron seguir avanzando en mis sueños, esforzándome cada día para cumplir mis objetivos. A mis compañeros, que fueron el apoyo en las aulas de clases compartiendo sus conocimientos. Esta tesis es el resultado de nuestras aspiraciones y esfuerzos compartidos.

Joseline Vanessa Salmerón Flores

AGRADECIMIENTO

Mis agradecimientos son a mi madre, mi familia, mis compañeros, mis profesores y a mis amigos quienes fueron un pilar fundamental a lo largo de mis estudios siempre apoyándome a seguir adelante y no rendirme. A la Universidad Politécnica Salesiana que siempre tiene sus puertas abiertas para inculcar una buena educación. A todos los profesores que me supieron transmitir sus conocimientos. A mis amigos y compañeros que hice en el transcurso de mi vida universitaria que siempre me apoyaron cuando tenía dudas.

Ángelo David Briones Rodríguez

AGRADECIMIENTO

Expreso mi más profundo agradecimiento a todas las personas que han sido fundamentales en la realización de esta tesis. A mis padres, Angelica Flores Lima y Miguel Salmerón, quienes con su amor, apoyo incondicional y valiosos consejos profesionales han sido mi mayor fuente de inspiración y motivación. A mi tutor de tesis, MSc. Diego Freire, por su orientación experta, paciencia y las valiosas críticas constructivas que me permitieron mejorar y avanzar en este proyecto. A mis familiares, especialmente a mi tía Yolanda Flores, por su apoyo en las pruebas de campo y su dedicación en ayudarme en todo lo que fue necesario, para que tuviera éxito en esta investigación. También agradezco a mis profesores, compañeros de trabajo y colegas por el intercambio de ideas y su apoyo constante, que enriquecieron tanto mi experiencia académica como profesional.

Finalmente, a mi compañero de tesis, Ángelo Briones, por su colaboración invaluable, que contribuyó al éxito de esta investigación.

“Sin el apoyo de todos ustedes, este proyecto no habría sido posible”.

Joseline Vanessa Salmerón Flores

RESUMEN DEL PROYECTO

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2025	ÁNGELO DAVID BRIONES RODRÍGUEZ JOSELINE VANESSA SALMERÓN FLORES	MSC.DIEGO FREIRE	" DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO IOT PARA LA MONITORIZACIÓN DE PARÁMETROS AMBIENTALES Y DE NUTRIENTES DEL SUELO APLICADOS A CULTIVOS DE CACAO “

El proyecto de titulación presenta el estudio de un prototipo de IOT para la monitorización ambiental y nutrientes del suelo del cacao, la cual consiste en la evaluación sistemática y periódica de varios elementos biológicos o componentes ambientales, se utilizará un macetero grande y alargado que albergará varias plantas de cacao jóvenes, de aproximadamente 3 a 4 meses de edad, cultivadas en tierra de abono traída directamente de una finca cacaotera. La implementación de este macetero busca recrear un entorno lo más similar posible al de una parcela real, permitiendo así la obtención de datos específicos sobre las condiciones del suelo y factores ambientales que influyen en el desarrollo del cultivo. Utilizando los dispositivos Lark Weather Station y el sensor NPK, que permitirá obtener los datos que serán clave para entender mejor el entorno ideal del cacao y optimizar el cultivo en diversas condiciones climáticas y de suelo.

Los datos obtenidos a través de los sensores instalados en la Lark Weather Station permite monitorear la temperatura, la humedad relativa, la presión atmosférica, así como la velocidad y dirección del viento. Adicionalmente, se utilizará un sensor NPK (Nitrógeno, Potasio y Fósforo) que medirá los nutrientes del suelo. Toda la información recopilada se integrará en el Dashboard en Grafana para una visualización centralizada.

Este prototipo de sistema IoT se presenta como un dispositivo físico autónomo y portátil, diseñado para operar de manera eficiente en entornos agrícolas. Su alimentación proviene de un panel solar que carga una batería recargable, garantizando su funcionamiento continuo durante el día y la noche. Esta independencia energética no solo permite la recopilación

ininterrumpida de datos, sino que también respalda la sostenibilidad del sistema.

Mediante la configuración de la Raspberry Pi, los datos se pueden transmitir en tiempo real a Grafana Cloud a través de una conexión inalámbrica Wi-Fi. Esta integración tecnológica facilita la visualización inmediata de la información obtenida de los sensores, que abarca tanto los parámetros ambientales como los niveles de nutrientes en el suelo. Durante un periodo de 7 a 15 días, al absorber los minerales y medición de factores ambientales de la solución se obtendrá un mejor cultivo

Palabras Clave: Sensor NPK, Lark Weather Station, IOT, Grafana, Cacao, Sensores

ABSTRACT

YEAR	STUDENTS	PRJ. DIRECTOR	SUBJECT
2025	ANGELO DAVID BRIONES RODRIGUEZ JOSELINE VANESSA SALMERON FLORES	MSC DIEGO FREIRE	"DESIGN AND IMPLEMENTATION OF IOT PROTOTYPE FOR THE MONITORING OF ENVIRONMENTAL PARAMETERS AND SOIL NUTRIENTS APPLIED TO COCOA CROPS"

The graduation project presents the study of an IoT prototype for environmental monitoring and soil nutrients of cocoa. This involves the systematic and periodic evaluation of various biological elements or environmental components. A large, elongated planter will be used to house several young cocoa plants, 3 to 4 months old, grown in fertilized soil sourced directly from a cocoa plantation. The implementation of this planter aims to recreate an environment as similar as possible to a real plot, thus allowing for the collection of specific data regarding soil conditions and environmental factors that influence crop development. By utilizing the Lark Weather Station devices and the NPK sensor, we will gather crucial data to better understand the ideal environment for cocoa and optimize cultivation under various climatic and soil conditions.

The data obtained through the sensors installed in the Lark Weather Station will enable us to monitor temperature, relative humidity, atmospheric pressure, as well as wind speed and direction. Additionally, an NPK (Nitrogen, Phosphorus, and Potassium) sensor will be used to measure soil nutrients. All collected information will be integrated into our dashboard in Grafana for centralized visualization.

This IoT system prototype is designed as a portable and autonomous physical device, built to operate efficiently in agricultural environments. It is powered by a solar panel that charges a rechargeable battery, ensuring continuous operation both day and night. This energy independence not only allows for uninterrupted data collection but also supports the

sustainability of the system.

By configuring the Raspberry Pi, data can be transmitted in real-time to Grafana Cloud via a wireless Wi-Fi connection. This technological integration facilitates the immediate visualization of the information obtained from the sensors, which encompasses both environmental parameters and nutrient levels in the soil. Over a period of 7 to 15 days, the absorption of minerals and measurement of environmental factors from the solution will result in improved crop yield.

Keywords: NPK Sensor, Lark Weather Station, IOT, Grafana, Cacao, Sensors

ÍNDICE GENERAL

INTRODUCCIÓN	20
1 EL PROBLEMA	2
1.1 Descripción del problema.....	2
1.2. Antecedentes	2
1.3. Importancia y alcances.....	3
1.4. Delimitación del problema.....	4
1.4.1. Temporal.....	4
1.4.2. Geográfica	4
1.4.3. Académica	5
1.5. Objetivos.....	5
1.5.1. Objetivo General	5
1.5.2. Objetivos Específicos	5
2 FUNDAMENTOS TEÓRICOS	6
2.1 Cacao	6
2.1.1 Composición y valor nutricional de la semilla del Cacao	7
2.1.2 Clasificación de plantas de cacao	7
2.2 Requerimientos del suelo para el cultivo del cacao	8
2.3 Nutrición del suelo del cacao.....	10
2.3.1 Nutrientes principales y su funcionamiento	11
2.3.2 Tipos de fertilizaciones.....	11
2.3.3 Análisis foliar en cultivos de cacao	12
2.3.4 Procesos análisis foliar en cultivos	12
2.3.5 Deficiencias nutricionales del cacao.....	13
2.3.6 Nutrientes Esenciales en el Cacao y Causas de su Deficiencia Nutricional.....	14
2.4 Internet de las cosas (IoT)	16
2.4.1 Tipos de conexiones IoT	18
2.4.2 En la domótica	18
2.4.3 Ambientes Inteligentes	19
2.4.4 Transporte y Logística.....	19
2.4.5 Agricultura Inteligente.....	19
2.4.6 Tecnología wifi	20
2.4.7 Aplicaciones IoT en tiempo real para Optimización de Procesos	20
2.5 Grafana	21

2.5.1	Características Principales de Grafana:	21
2.5.2	Facilidad de Instalación y Configuración:	22
2.5.3	Casos de Uso Comunes:	22
2.5.4	Ventajas de Usar Grafana:	23
2.5.5	Cómo Usarlo con InfluxDB:	24
2.6	Influxdb	25
2.6.1	Características Principales de InfluxDB:.....	25
2.6.2	Casos de Uso Comunes:	27
2.6.3	Estructura de Datos en InfluxDB:	28
2.7	Raspberry Pi	29
2.7.1	Características Principales de Raspberry Pi 4:.....	29
2.7.2	Ventajas de Raspberry Pi 4:.....	32
2.7.3	Aplicaciones Comunes de Raspberry Pi 4:.....	33
2.8	Sensor NPK	35
2.9	Lark Weather Station	37
2.10	Módulo USB a RS-485	40
2.10.1	Características del Módulo USB a RS-485:.....	41
2.10.2	Teoría del Protocolo RS-485:.....	43
2.11	Controlador de carga solar PWM	45
2.12	Módulo Step Down XL4015	46
2.12.1	Características del Módulo Step-Down XL4015:	47
3.1.1	Fases del Funcionamiento del XL4015:	48
3.1.2	Ventajas del Convertidor Buck:	49
3.1.3	Aplicaciones Comunes del Módulo Step-Down XL4015:	50
3.2	Panel Solar	51
3.2.1	Fotovoltaica (Efecto Fotovoltaico):	51
3.2.2	Componentes de un Panel Solar:.....	51
3.2.3	Tipos de Paneles Solares:	52
3.2.4	Generación de Electricidad (DC):.....	53
3.2.5	Eficiencia de un Panel Solar:.....	53
3.2.6	Instalación y Orientación:	54
3.2.7	Potencia y Tamaño de los Paneles Solares:	54
3.2.8	Sistema Fotovoltaico:	55
3.2.9	Ventajas de la Energía Solar:.....	55
3.3	Baterías	56

3.3.1	Baterías abiertas	56
3.3.2	Baterías selladas.....	57
4	MARCO METODOLÓGICO	58
4.1	Descripción del sistema IoT	58
4.1.1	Sensores y Adquisición de Datos	¡Error! Marcador no definido.
4.1.2	Procesamiento de Datos en Raspberry Pi 4	60
4.1.3	Almacenamiento en InfluxDB Cloud	60
4.1.4	Visualización y Alertas en Grafana Cloud.....	60
4.2	Ensamblaje físico del proyecto IoT.....	62
4.2.1	Integración de Sensores con la Raspberry Pi 4.....	65
4.2.2	Verificación del Funcionamiento del Sistema	67
4.3	Configuración de la Raspberry Pi 4	70
4.3.1	Actualización del Sistema Operativo	70
4.3.2	Configuración de Repositorios	71
4.3.3	Configuración de Puertos USB.....	71
4.3.4	Instalación de InfluxDB.....	72
4.3.5	Instalación de Grafana	73
4.3.6	Creación y Configuración del Entorno Virtual	74
4.3.7	Instalación del Sensor DFRobot_LarkWeatherStation.....	75
4.4	Configuración del sensor NPK.....	76
4.4.1	Configuración del Sensor NPK en Raspberry Pi.....	76
4.4.2	Conexión del Sensor NPK.....	76
4.4.3	Instalación de Dependencias	77
4.4.4	Configuración del Sensor NPK en Python.....	77
4.4.5	Explicación del Código	79
4.4.6	Verificación de la Comunicación	79
4.5	Programación de Raspberry Pi – Almacenamiento Local.....	79
4.5.1	Importación de Bibliotecas	79
4.5.2	Configuración de la Base de Datos InfluxDB.....	80
4.5.3	Inicialización de Sensores	80
4.5.4	Lectura de Sensores	81
4.5.5	Enviar Datos a InfluxDB	82
4.5.6	Ejecución del Programa	83
4.6	Configuración de código para enviar los datos a InfluxDB Cloud	84
4.6.1	Importación de Bibliotecas	84

4.6.2	Configuración de InfluxDB Cloud	85
4.6.3	Configuración de Sensores	85
4.6.4	Lectura de Sensores	85
4.6.5	Almacenamiento en InfluxDB Cloud	87
4.6.6	Configuración de InfluxDB Cloud	88
6	PRESUPUESTO	94
7	RESULTADOS	94
7.1	Resultados de la Configuración del Sistema IoT	94
7.1.1	Funcionamiento del Sistema Energético	95
7.1.2	Integración de Sensores	95
7.1.3	Transmisión y Almacenamiento de Datos	95
7.1.4	Análisis de Datos en Grafana Cloud.....	96
7.1.5	Variabilidad de Parámetros Meteorológicos	96
7.1.6	Análisis de la Calidad del Suelo	96
7.1.7	Configuración de Alertas	96
7.1.8	Impacto del Sistema en la Toma de Decisiones	97
7.2	Evidencias de los resultados de obtenidos del prototipo IoT	97
8	CONCLUSIONES	109
9	RECOMENDACIONES.....	110
10	BIBLIOGRAFÍA.....	111
11	ANEXOS	114
11.1	Código Python en Raspberry PI envío de datos a InfluxDB Cloud.....	114
11.2	Código Python en Raspberry PI envío de datos a InfluxdDB Local	119
11.3	Script Batch de encendido automático en Raspberry Pi.....	124

ÍNDICE DE FIGURAS

Figura 1. <i>Deficiencias nutricionales de las hojas Zonas de producción en el Ecuador</i>	16
Figura 2 <i>Plataforma Grafana</i>	25
Figura 3 <i>Plataforma InfluxDB</i>	29
Figura 4 <i>Raspberry pi 4</i>	35
Figura 5 <i>Sensor NPK</i>	35
Figura 6 <i>Lark Weather Station</i>	37
Figura 7 <i>Conexión LWS con microcontroladores</i>	38
Figura 8 <i>Convertidor bidireccional USB a RS485</i>	44
Figura 9 <i>Convertidor bidireccional USB a RS485</i>	45
Figura 10 <i>Controlador de carga solar PWM</i>	46
Figura 11 <i>Módulo Step Down</i>	50
Figura 12. <i>Panel solar</i>	55
Figura 13 <i>Batería 12V</i>	57
Figura 14 <i>Diagrama de flujo del Sistema Metrológico lot</i>	61
Figura 15 <i>Panel solar</i>	63
Figura 16 <i>Batería</i>	63
Figura 17 <i>Controlador de carga PWM</i>	64
Figura 18 <i>Conexión del PWM a los reguladores de voltaje a la Raspberry Pi 4</i>	65
Figura 19 <i>Evaluación de prueba del prototipo</i>	65
Figura 20 <i>Sensor NPK</i>	66
Figura 21 <i>Modulo convertidor modbus USB a RS485</i>	67
Figura 22. <i>Validaciones del prototipo</i>	68
Figura 23 <i>Elementos del prototipo IoT</i>	69
Figura 24 <i>InfluxDB Cloud</i>	88
Figura 25 <i>Data Explorer InfluxDB</i>	89
Figura 26 <i>Configuración del Bucket</i>	89
Figura 27 <i>Configuración datos en InfluxDB, script InfluxDB</i>	91
Figura 28 <i>Configuración datos en InfluxDB</i>	92
Figura 29 <i>InfluxDB Local dashboard de sensores</i>	98
Figura 30 <i>Datos de cantidad de fósforo del suelo</i>	98
Figura 31 <i>Datos de temperatura ambiente</i>	99
Figura 32 <i>InfluxDB script flux</i>	99

Figura 33 Script para visualizar la presión atmosférica	100
Figura 34 <i>Pantalla principal de Grafana Cloud</i>	101
Figura 35 Usuarios de Grafana Cloud	101
Figura 36 <i>Inicio de Grafana</i>	102
Figura 37 Panel de control Dashboard Grafana Cloud	103
Figura 38 Dashboard de sensores de dirección de viento, temperatura y humedad	104
Figura 39 Dashboard de sensores de presión y velocidad.....	104
Figura 40 Datos de sensores NPK.....	105
Figura 41 Data source	105
Figura 42 Configuración del dashboard	106
Figura 43 <i>Data Query</i>	106
Figura 44 Configuración de alertas mail	107
Figura 45 Alerta de altura mínima.....	108
Figura 46 Alerta de altura máxima	108

ÍNDICE DE TABLAS

Tabla 1 <i>Información Nutricional del Cacao por semilla</i>	7
Tabla 2 <i>Requerimientos agroecológicos para el cultivo del Cacao Fuente</i>	9
Tabla 3 <i>Cantidad nutricional (Kg/HA)</i>	13
Tabla 4. Especificaciones técnicas del sensor NPK	36
Tabla 5 Compatibilidad LWS con microcontroladores	39
Tabla 6 Especificaciones técnicas Lark Weather Station	39
Tabla 7 Cronograma de Actividades	93
Tabla 8 Presupuesto del proyecto.....	94

INTRODUCCIÓN

Este trabajo estudia una propuesta que buscó la implementación de un nuevo sistema IOT (Internet de las cosas), diseñando específicamente para la obtención de datos atmosféricos y nutrientes del suelo, que sea autosustentable, portátil y eficiente en la recolección de información específica, de tal manera que por medio de este método permitirá optimizar un buen cultivo de cacao en el Ecuador.

La agricultura moderna enfrenta desafíos significativos debido a las variaciones climáticas y las condiciones del suelo, lo que hace necesario contar con herramientas que faciliten un monitoreo constante y efectivo. A través de esta metodología, se busca no solo mejorar la producción de cacao, sino también promover prácticas agrícolas sostenibles que beneficien a los agricultores y al medio ambiente. Al integrar tecnologías avanzadas en la gestión agrícola, este sistema tiene el potencial de revolucionar la manera en que se cultiva y se gestiona el cacao, garantizando un futuro más productivo y sostenible

El cacao es uno de los productos más importantes en el Ecuador, este producto es deseado a nivel nacional e internacional debido a su gran sabor y calidad, siendo un ingrediente esencial para la elaboración de chocolate, destacándose por su amplia producción en la zona litoral del país, conocida como la Costa Ecuatoriana, donde se centra la mayor cantidad de cultivo.

Este prototipo del sistema meteorológico que se implementó tiene las funciones que permitirán monitorear y analizar los parámetros atmosféricos y nutrientes del suelo, para así obtener un mejor control en los cultivos de cacao.

1 EL PROBLEMA

1.1 Descripción del problema

El poco control y tecnificación de los cultivos del cacao en el Ecuador específicamente en pequeñas y medianas fincas han provocado la proliferación de problemas fitosanitarios como la moniliasis, mazorca negra y la escoba de bruja, produciendo pérdidas de hasta el 60% en la producción de una hectárea. Según estudios realizados por el Instituto Nacional Autónomo de Investigaciones Agropecuarias del Ecuador (INIAP), los problemas prioritarios que afectan a la producción del cacao son las plagas, enfermedades, falta de nutrientes en los suelos, así como también la falta de conocimientos técnicos en las granjas y fincas agrícolas (Bula, 2020).

La poca cantidad de nutrientes del suelo para cultivos del cacao ha causado que muchas plantaciones de cacao no se desarrollen de la mejor manera, y de como resultado una mazorca de menor tamaño o contextura (Santiago et al., 2023).

1.2. Antecedentes

Se han desarrollado varios temas de investigación sobre el diseño, monitoreo y de los datos atmosféricos y nutrientes en el suelo de los cultivos de cacao. En los cuales se destacan los siguientes proyectos:

En las Naves en el año 2017 Espinoza desarrollo un proyecto con la finalidad de obtener información cuantitativa sobre los nutrientes en el suelo de los cultivos de Cacao, el proyecto lo realizó explorando y estudiando la nutrición, desarrollo y crecimiento del Cacao a cortas distancias para obtener la variación nutricional en el suelo y en como esto afecta el comportamiento nutricional, útil y reproductivo de la plantación del Cacao en respuesta a las variaciones nutricional del suelo.

En el año 2021 Balladares Feijo y López Vargas desarrollaron un prototipo fijo autosustentable en el que incluyeron el uso de varios sensores para medir los datos atmosfericos como la velocidad del viento, la humedad, la temperatura, entre otros. El objetivo principal era crear una estación autosustentable que mida los valores atmosfericos para tecnificar el crecimiento

y mantener las plantaciones agrícolas de Cacao en el Ecuador.

Este último proyecto realizado en el año 2023 por Lojano Angamarca y Méndez Vásquez, formularon una red de monitoreo que tiene como fin el desarrollo de un sistema IoT para la búsqueda, observación y seguimiento de los productos agrestes en el estudio del Cacao utilizan redes de celular y geolocalización en tiempo real.

1.3. Importancia y alcances

La agricultura es una industria compleja que depende considerablemente de las condiciones ambientales para asegurar el desarrollo exitoso en los cultivos. Los labradores siempre están en la búsqueda de métodos para optimizar sus rendimientos y minimizar las pérdidas de sus cosechas. Una tecnología que está ganando aceptación en el ámbito agrícola es el uso de sensores ambientales en la producción de cultivos.

Estos sensores ofrecen datos en tiempo real sobre diversas condiciones ambientales, facilitando a los labradores a la toma de decisiones comunicando y aumentando la productividad general de sus cultivos. En este artículo, se examinarán las ventajas de instalar un sensor ambiental en entornos agrícolas. Al incorporar un sensor ambiental en sus cultivos, los agricultores pueden supervisar con precisión diversas condiciones ambientales que influyen directamente en el crecimiento de estos. Estos sensores evalúan parámetros como la temperatura, la humedad, la intensidad de la luz, la humedad del suelo y los niveles de pH (PrismaB, 2024).

Es importante subrayar que este sistema no solo incorporó sensores para monitorear las condiciones atmosféricas, sino que también cuenta con un sensor adicional que analizó los nutrientes presentes en el suelo. Esto aporta información clave para asegurar un desarrollo óptimo del cultivo de cacao. De esta manera, el monitoreo será más completo, permitiendo controlar tanto los factores ambientales como las condiciones nutricionales del suelo, lo que contribuirá a mejorar la salud y productividad de las plantaciones.

Lo que se busca en este proyecto es mejorar es la precisión en:

- La rentabilidad
- Las cualidades del fruto y el cultivo
- La productividad del Cacao
- La sanidad alimenticia

Y busca reducir:

- Los movimientos terrestres
- Evitar la adquisición de enfermedades
- Prevenir los efectos adversos climáticos
- El uso continuo de insecticidas
- El uso constante de abono o consumibles

La cualidad de medir de los sensores desarrolla el método para predecir los pasos a seguir para la toma de decisiones y direccionamiento en el uso frutífero de los elementos agrícolas. (PrismaB, 2024).

Este tema tiene como finalidad promover la participación en el innovador método de cultivo de cacao, fortalecer el diseño de controles de temperatura y riego, y fomentar su implementación en áreas de mayor producción.

1.4. Delimitación del problema

1.4.1. Temporal

El proyecto se desarrolló entre los meses de octubre 2024 a febrero 2025.

1.4.2. Geográfica

El proyecto es un prototipo de un IoT para la monitorización de parámetros ambientales y nutrientes del suelo.

Para este proyecto de investigación se utilizó una maqueta de cultivos de cacao que simuló una pequeña parcela de una finca como tal, la maqueta consta de un macetero grande alargado con varias plantas de cacao de 3 a 4 meses de edad y tierra de abono traída de una finca cacaotera, con la finalidad de emular un espacio que permita la obtención de los datos del suelo, y medioambientales. La maqueta y las pruebas, junto con demostraciones técnicas, se realizaron en el laboratorio de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, sede Guayaquil, luego de la afinación y programación de los sensores se realizó la presentación y entrega del prototipo, con el fin de ser utilizado para prácticas y futuras mejoras del prototipo.

1.4.3. Académica

El proyecto estudia el diseño e implementación de un prototipo IoT para la monitorización de parámetros ambientales y nutrientes del suelo aplicados a cultivos de cacao y se lo realizó cumpliendo con las normas impuestas por Universidad Politécnica Salesiana basado en su grado investigativo y modelo de presentación para proyectos de titulación, donde se implementaron todos los conocimientos adquiridos de materias tales como: automatización, microprocesadores, energía y medio ambiente.

1.5. Objetivos

1.5.1. Objetivo General

Diseñar e implementar un prototipo IoT para la monitorización de parámetros ambientales y de nutrientes del suelo aplicados a cultivos de cacao utilizando Lark Weather Station, Sensor NPK, Raspberry PI y Grafana Cloud.

1.5.2. Objetivos Específicos

- Diseñar prototipo de monitorización de parámetros ambientales con LARK Weather Station e integrarlo a la Raspberry PI para el procesamiento de datos de los

sensores de temperatura ambiente, humedad relativa, presión atmosférica, velocidad del viento y dirección del viento.

- Configurar el sensor NPK para la toma de datos de nutrientes del suelo como nitrógeno, fósforo y potasio e integrarlo a la Raspberry PI para el procesamiento y almacenamiento de datos.
- Configurar la Raspberry PI para el envío de datos en tiempo real a Grafana Cloud mediante un acceso inalámbrico WIFI, para observar en un Dashboard la información de los sensores ambientales y de nutrientes del suelo.
- Realizar pruebas en laboratorio y en campo de los parámetros ambientales y nutrientes del suelo de cultivos de cacao.

2 FUNDAMENTOS TEÓRICOS

En esta sección se presentan los conceptos esenciales que servirán como base para entender el tema que se plantea, así como los aspectos fundamentales requeridos para garantizar el adecuado desarrollo de una plantación de cacao. Además, se detallan tecnologías utilizadas en el proyecto, así como, las tecnologías de comunicación a internet y el IoT.

2.1 Cacao

El cacao es un fruto que proviene del árbol cacaotero es de origen amazónico, el fruto de este árbol presenta una tonalidad que varía entre amarillo y morado, tiene una forma ovalada y alcanza aproximadamente 30 cm de longitud, dentro de este fruto tiene alrededor de 30 a 40 semillas, las cuales son ricas en nutrientes. El éxito en su cosecha depende de contar con el clima adecuado, ya que un ambiente óptimo no solo mejora la calidad de los granos, sino que también favorece el rendimiento de las plantaciones ((FAO), 2012).

En el procedimiento para sembrar, cultivar y vender del fruto, es esencial garantizar que en el terreno donde se está cultivando el Cacao se estén aplicando las medidas y técnicas físicas que logren que el suelo tenga el nivel deseado y óptimo de nutrientes y minerales. (Frutícola, 2024).

2.1.1 Composición y valor nutricional de la semilla del Cacao

En la **Tabla 1** información nutricional del Cacao por semilla presenta los valores nutricionales de una semilla o grano de cacao, destacando su alto contenido de nutrientes. Una semilla de cacao puro proporciona aproximadamente 20 g de proteínas, 1500 mg de potasio, 13.9 mg hierro, 33 g de fibra 734 mg de fosforo, 499 mg magnesio y 228 kcal de energía (cadenacacaoca.info, 2025).

Tabla 1

Información Nutricional del Cacao por semilla

Porción nutricional del cacao	Cantidad
Proteína	20g
Potasio	1500mg
Hierro	13.9mg
Fósforo	734mg
Fibra	33g
Magnesio	499mg
Energía	228kcal

Nota. (Productor, 2019)

2.1.2 Clasificación de plantas de cacao

La planta de cacao crece en regiones tropicales donde el clima cálido y las lluvias frecuentes son clave para su buen desarrollo. Existen tres tipos principales de plantas: forastero, criollo y trinitario.

2.1.2.1 El forastero

El Forastero también conocido como amazónico, posee una gran mutabilidad en la vida silvestre tanto en la Amazonia alta, así como en la baja. Su principal cualidad visible es presentar un estambre estéril de color púrpura o morada, la panícula suele ser amarillas cuando llegan a su estado de maduración, su forma ovoide posee una muesca hundida en la superficie, la cubierta exterior es lisa, con una capa delgada o recio.

El forastero es caracterizado para elaborar chocolate, debido a su gran capacidad para repeler y resistir enfermedades, así como su ofrece una comodidad en el momento de su cuidado. (MOCCA, 2023).

2.1.2.2 El criollo

Conocido como el rey del cacao, promete un fruto con gusto odorífico agradable, pese a esto el criollo tiene un método de cultivo más estricto debido a que tiene la debilidad de adquirir o verse afectada por un sinnúmero de enfermedades. Sus características más destacadas son que posee tallos más delgados obteniendo una productividad más baja y hacerlos más propensos a la infección de plagas, sus panículas obtienen unos tonos rojos o amarillos cuando llegan al estado de madurez y sus flores presentan una coloración rosada pálida. El criollo tiene 10 cortes profundos, con una textura áspera y con pigmentación moteada, sus frutos poseen paredes delgas con semillas que tienen un tamaño considerablemente grande de color purpura pálido o blancas(MOCCA, 2021).

2.1.2.3 El trinitario

Es el resultado de la mezcla de las dos anteriores, combinando características de ambas. El trinitario posee características que la diferencia del resto ya que posee un desarrollo fuerte y estable, con granos circulares de un color violeta claro. Sus mazorcas tienen variación de colores en su estaba de maduración que van desde el rojo, purpura y agradable a la vista, mostrando así sus mutaciones mestizas. (Góngora-Duarte et al., 2023).

2.2 Requerimientos del suelo para el cultivo del cacao

El *Theobroma cacao* L o cacao ecuatoriano, es un árbol originario de América del Sur que se cultiva principalmente en regiones tropicales. Según (Zarrillo et al., 2018), su origen se encuentra en la zona fronteriza entre Ecuador y Perú.

El cacao, como cualquier otra planta, requiere condiciones específicas para lograr una alta productividad, incluyendo factores relacionados con el clima, la topografía y el suelo ((FAO), 2012).

Para garantizar una buena productividad y cosecha, el cultivo de cacao exige condiciones específicas relacionadas con el clima, el suelo, el viento, la temperatura y la precipitación. Por ello, es fundamental que los agricultores consideren ciertos aspectos clave al iniciar el cultivo de las plantas de cacao, tales como:

- La temperatura ideal debe encontrarse entre los 23 °C y 25 °C
- Cuando la temperatura es igual o menor a 21 °C afecta en el desarrollo floreal, afectando el crecimiento, ocasionando poco rebote y es casi inexistente la floración de las plantas.
- La maduración de la fruta se ve ralentizada cuando la temperatura es baja
- La precipitación pluvial ideal es de 1200 a 1500 mm anualmente y mensual debería ser 100mm como mínimo
- En zonas muy secas se recomienda realizar riegos constantes para mantener el cultivo
- En zonas con mucha precipitación pluvial aumenta el riesgo de contraer bacterias, enfermedades fungosas y invasión de insectos.
- Se establece en suelos con textura suelta, franco-arenoso, franco-arcilloso
- Los suelos para las plantas de cacao deben tener un 3.5% de material orgánico
- Los suelos de la planta cacaotera deben ser ricos en nitrógeno, potasio, fósforo y calcio
- Las plantas de cacao deben ser plantada a 1.5 metros de profundidad
- El pH ideal debe ser entre 5.5 a 7
- La pedregosidad en los cultivos de cacao, no deben de presentar rocas grandes
- Las pendientes del suelo del cultivo deben ser onduladas y planas no deben de sobrepasar el 30%
- El nivel freático debe ser de 1.5 metros
- Los vientos no pueden ser mayores a 14 km/h, cuando los vientos sobrepasan esos valores es ideal colocar rompevientos.
- En la tabla 2 se observa los valores adecuados para el cultivo de cacao según el MAGAP (Ecuador, 2025).

Tabla 2

Requerimientos agroecológicos para el cultivo del Cacao Fuente

FACTOR	VARIABLE	APTITUD AGROECOLOGICA			
		OPTIMA	MODERADA	MARGINAL	NO APTA
SUELO	Pendiente	0-25%	25-50%	50-70%	>70%
	Textura		Limoso, Franco arcilloso		
		Franco, Franco arcilloso (> a 35%), Franco limoso	(<35% arcilla) franco arcillo arenoso, Franco arcillo limoso, Franco arenoso (fino a grueso), Arcillo arenoso, Arcillo limoso	Arcilloso, Arcilloso (>60%)	Arenoso (fina, media, gruesa) Arenoso Francoso
	Profundidad	Profundo	Moderadamente profundo	Poco profundo	Superficial
	Pedregosidad	Sin	Pocas	Frecuentes	Abundantes, Pedregoso a rocosa
	Drenaje	Bueno	Moderado	(-)	Mal Drenado, Excesivo
	Nivel Freático	Profundo	Mediamente profundo	Poco profundo	Superficial
	pH	Ligeramente acido, neutro	Acido, moderadamente alcalino	(-)	Muy acido, alcalino
	Toxicidad	sin o nula	Ligera	Media	Alta
	M.O.	Muy alta, alta	Media	Baja	Muy baja
	Salinidad	Sin ligera	Media	Alta	Muy baja
	Fertilidad	Alta	Media	Baja	Muy baja
	Precipitación	1200-1300	1000-1200	800-1200	<800>4000
	Clima	Temperatura (°C)	16-18	15-16	<15
	Altitud	0 a 500	(-)	(-)	>500

Nota. (Ecuador, 2025)

2.3 Nutrición del suelo del cacao

En Ecuador, la nutrición adecuada del suelo es fundamental para producir cacao de alta calidad, tanto para satisfacer demandas económicas como sociales, a nivel nacional e internacional. Los agricultores, productores y responsables de los cultivos deben monitorear constantemente los niveles de nutrientes y minerales en el suelo para asegurar que se mantengan en condiciones óptimas.

Para lograr esto, es esencial implementar un programa de fertilización adecuado, ya sea durante el crecimiento del cacao o al iniciar el cultivo desde sus primeras etapas. Una correcta fertilización en los campos cacaoteros no solo garantiza un producto de calidad superior, sino que también protege contra enfermedades y reduce el estrés hídrico. Este último no solo se asocia con la falta de agua, sino también con problemas en la absorción y transpiración del agua por parte de las plantas.

Es importante destacar que estos nutrientes esenciales provienen de los fertilizantes, los cuales contribuyen a una cosecha más abundante y de mejor calidad, beneficiando directamente a los agricultores (cadenacacaoca.info, 2025).

2.3.1 Nutrientes principales y su funcionamiento

- **Nitrógeno:** crecimiento floreal y producción de clorofila
- **Fosforo:** creación de raíces fuertes y crecimiento de flores
- **Potasio:** ayuda en la absorción de nutrientes y evita la privación del agua
- **Calcio:** conserva la estructura de las células
- **Magnesio:** Ayuda en el proceso de la fotosíntesis
- **Zinc:** Ayuda en la producción de la clorofila y de raíces fuertes
- **Hierro:** Parte importante en la producción de la clorofila
- **Manganeso:** realiza el proceso de creación de carbohidratos (Relief, 2011).

2.3.2 Tipos de fertilizaciones

La fertilización del suelo consiste en administrar los nutrientes indispensables para mejorar tanto el crecimiento floral de las plantas como su fertilidad. En Ecuador, se emplean diversos métodos de fertilización, entre los cuales se destacan:

- **Fertilización edáfica:** Este método implica la aplicación directa de nutrientes al suelo con el objetivo de optimizar su fertilidad y calidad. Puede llevarse a cabo en cualquier etapa del cultivo, incluyendo la siembra, el desarrollo floral, el crecimiento de hojas, raíces y frutos.
- **Fertilización foliar:** Este enfoque se basa en aplicar nutrientes en forma de líquido o aerosol directamente sobre las hojas. Es una práctica eficaz para corregir la falta específica de nutrientes en las plantas, ya que facilita una absorción rápida a través de las hojas.
- **Fertirriego:** Este método combina precisión y flexibilidad, ya que permite la aplicación de nutrientes disueltos en agua en cualquier momento del ciclo de cultivo. Es adecuado tanto para cultivos de gran escala como para pequeños, optimizando la distribución de nutrientes junto con el riego (Productor, 2019).

2.3.3 Análisis foliar en cultivos de cacao

El análisis foliar es una técnica para garantizar el crecimiento y mejorar el rendimiento de los cultivos de cacao utilizada en la agricultura para analizar el estado nutricional de las plantas a través del estudio de las hojas. Permite determinar si la cosecha cuenta con una buena nutrición, presenta alguna enfermedad o algún déficit nutricional, logrando medir concentraciones de nutrientes necesarios (como nitrógeno, fósforo, potasio, calcio, magnesio, etc.) para analizar los niveles químicos en las plantas y garantizar su óptimo desarrollo (Bula, 2020).

2.3.4 Procesos análisis foliar en cultivos

- **Toma de muestras:** En los cultivos de cacao, se recolectan hojas jóvenes completamente desarrolladas de ramas no sombreadas, preferiblemente en el estado fisiológico óptimo. Estas hojas representan mejor el estado nutricional del árbol y proporcionan datos confiables para el análisis.
- **Análisis en laboratorio:** Las muestras recolectadas se somete a un proceso fr limpieza y secado antes de ser analizado químicamente. Se determina la

concentración de nutrientes esenciales como nitrógeno, fósforo, potasio, calcio, entre otros. Este paso permite conocer el balance nutricional de la planta, los cuales son cruciales para el desarrollo del cacao.

- **Interpretación de resultados:** Los datos obtenidos se comparan con valores de referencia específicos para el cacao, permitiendo identificar deficiencias o excesos de nutrientes. Esta información es clave para ajustar las prácticas de fertilización, asegurando que las plantas reciban los nutrientes necesarios en las cantidades adecuadas. ((FAO), 2012) Ver **tabla 3**.

Tabla 3

Cantidad nutricional (Kg/HA)

REQUERIMIENTO NUTRICIONAL (KG/HA)								
ESTADOS	EDAD (meses)	N	P	K	Ca	Mg	Mn	Zn
VIVERO	5—12	2,4	0,6	2,4	2,3	1,1	0,004	0,001
DESARROLLO	13-28	136	14	156	113	473,9	0,5	
INICIO DE LA PRODUCCION	29-49	212	23	321	140	717,1	0,9	
PLENA PRODUCCION	50-87	438	48	633	373	1296,1	1,5	

Nota.((FAO), 2012)

2.3.5 Deficiencias nutricionales del cacao

Las deficiencias nutricionales del cacao se pueden observar comúnmente en las hojas de la planta cacaotera, aunque también se pueden presentar en el tallo, flores y frutos. Estas deficiencias se dan debido a la falta de minerales necesarios en los cultivos del cacao ocasionando muerte en la cosecha, crecimiento lento, pérdida de hojas, frutos y flores. Estas deficiencias se deben de corregir inmediatamente mediante la fertilización de los suelos usando cualquiera de los métodos de fertilización, ya sea fertirriego, fertilización foliar o fertilización edáfica, una vez realizado los análisis requeridos para identificar cual/es son los nutrientes faltantes se debe tomar en cuenta la movilidad de nutrientes, esto indica que

nutrientes pueden ingresar al suelo y luego a las plantas, hay que tener en cuenta que en los cultivos cacaoteros la movilidad de nutrientes se divide en tres categorías:

- Los nutrientes móviles como Magnesio, Potasio, Fosforo y Nitrógeno su deficiencia se ve en las hojas viejas
- Los nutrientes de movilidad media como el Azufre se observan en el tercio medio de la planta

Los nutrientes de movilidad baja como Manganeso, Hierro, Cobre, Calcio, Boro y Zinc su deficiencia se ve en las Hojas jóvenes (Frutícola, 2024).

2.3.6 Nutrientes Esenciales en el Cacao y Causas de su Deficiencia Nutricional

- **Nitrógeno (N):** En las hojas se observa un color amarillo, delgadas en las hojas, crecimiento de flores anormal y crecimiento lento de la planta.
- **Fosforo (P):** El desarrollo de la planta se ralentiza, provocando hojas pequeñas, raíces delgadas, cortas y con poca profundidad en la tierra, en el desarrollo temprano de las hojas se puede visualizar las venas descoloridas y en las hojas maduras se logra visualizar un tono descolorido en los bordes y puntas
- **Potasio (K):** se logra observar en las puntas de las hojas un amarillento que se va extendiendo hasta dentro de las hojas, los frutos presentan menor contenido y su falta de sabor en el mismo.
- **Calcio (Ca):** LA deficiencia del calcio se puede observar en la necrosis de las hojas jóvenes y nuevas, que ocasionan pequeñas islas en las hojas que se extienden hasta llegar a la vena central de las hojas. En los frutos se puede apreciar putrefacción en la zona apical, así como deformaciones frutales, además de estos síntomas se observa un desarrollo lento en las raíces provocando muerte de la planta y caída de hojas.

- **Magnesio (Mg):** los síntomas se presentan en las hojas viejas provocando clorosis, que es cuando una planta no produce suficiente clorofila, ocasionando muerte en los tejidos entre las nervaduras de las hojas, generando un nivel bajo de hidratos de carbono debido al bajo nivel de fotosíntesis en la planta, esto suele ocurrir en viveros o suelos ácidos.
- **Azufre (S):** En las hojas nuevas se notan con poco desarrollo y de color amarillento pálido, esto se debe a la baja integración de proteínas por la escasez de aminoácidos provocando una reducción en la producción de la clorofila.
- **Boro (B):** la deficiencia de Boro en el cacao se presenta en las hojas jóvenes de tamaño pequeño y curvadas en consecuencia del pequeño tamaño de la vena central, el limbo floreal adquiere una consistencia frágil y tiene baja viabilidad del polen, los frutos y las flores presentan malformaciones, esto se obtiene en suelos alcalinos, arenosos y lixiviados.
- **Hierro (Fe):** se observa en las hojas jóvenes una clara clorosis presentando un color amarillento y nervaduras verdes, cuando es avanzado la clorosis presenta un blanqueamiento en las hojas, también se observe poco desarrollo en el tronco dando un aspecto delgado y corto. Esto es común en tierras con poca materia orgánica, alcalinos y mal drenados.
- **Manganeso (Mn):** la falta de manganeso se presenta como una clorosis intervenal en las hojas jóvenes, con nervaduras de color verde. Es normal encontrar este tipo de efectos en suelos alcalinos.
- **Zinc (Zn):** la falta de zinc provoca que las hojas en desarrollo tengan mal deformaciones y estiramiento terminando en una punta con forma de hoz. Estos síntomas se presentan en suelos arenosos o alcalinos (elproductor.com, 2019).

En la **figura 1** se observa la deficiencia nutricional de las hojas en los cultivos en el Ecuador.

Figura 1.

Deficiencias nutricionales de las hojas Zonas de producción en el Ecuador



Nota. (cadenacacaoca.info, 2025)

2.4 Internet de las cosas (IoT)

El término IoT, o Internet de las Cosas, se refiere a la interconexión de dispositivos a través de la red, lo que les permite compartir datos y coordinar acciones. En la agricultura, esta tecnología posibilita la conexión de diversos dispositivos, de manera que trabajen juntos de forma sincronizada, funcionando como un organismo integrado para optimizar los procesos agrícolas.

Por medio de la red de seguimiento agrícola monitorizando los datos atmosféricos y topográficos, combinados con la información recolectada y almacenada de la IoT agrícola, se ha demostrado una gran mejora en la cosecha agrícola reduciendo tiempo y gastos en recursos agrícolas. El equipo utilizado como drones, máquinas agrícolas, sensores y redes satelitales están conectados en tiempo real para poder rastrear y detectar cualquier anomalía en los cultivos.

Con estos datos obtenidos, es seguro poder programar las máquinas para el riego de nutrientes y las máquinas de fertilizantes, para que puedan aplicar los productos necesarios

en tiempos adecuados en los determinados sectores, donde se detecte la falta de algún nutriente específico, mejorando así el desarrollo del cultivo agrícola logrando una mayor optimización de recursos.

Desde el inicio de la plantación hasta el análisis topográfico del terreno, la agronomía inteligente es apto para la obtención de información y actuar en medida para actuar en todas las etapas del cultivo. Al adaptar esta información con el IoT, los agricultores son más aptos para programar y automatizar mejor los procesos de riego y aplicación de nutrientes, obteniendo un mejor control en la siembra y que la maquinaria agrícola sea más eficiente. (Burbano Choez, 2022).

En la aplicación de los nuevos avances tecnológicos automatizados en el área agrícola ecuatoriana, se observa fallas y deficiencias debido a la carencia de información y conocimiento de los nuevos métodos agronomicos. El uso correcto de la IoT en la agricultura contribuye en una mejor productividad, ventajas económicas reduciendo costos y en el desarrollo del cultivo, todo esto integrando tecnología moderna especializada para un mismo propósito.

El uso del internet de las cosas aporta al agricultor acceso a datos en tiempo real sobre sus cultivos. Al instalar sensores en el campo, se recopilan datos atmosféricos, la topografía del suelo, el estado de las plantas y otros factores. Esta información se envía de manera inalámbrica a un dispositivo central, donde se puede analizar. De este modo, los agricultores pueden gestionar sus cultivos de manera eficiente, tomando decisiones informadas para mejorar el rendimiento y optimizar los recursos.

Para este proyecto se propone un sistema basado en el Internet de las Cosas (IoT) para el monitoreo ambiental y el análisis de nutrientes del suelo. Este sistema tiene como objetivo recopilar datos sobre las condiciones ambientales y del suelo, como temperatura, humedad, lluvia y viento, además de analizar la calidad del suelo en términos de nutrientes esenciales (nitrógeno, fósforo, potasio) y agua. Esto se realiza mediante un sistema que combina la DFRobot Lark Weather Station y un sensor NPK, que se instala bajo tierra. La información obtenida se visualiza de forma gráfica en la plataforma Grafana Cloud. Esto asegura que el cultivo reciba la cantidad exacta de agua y nutrientes necesarios, sin intervención manual ni desperdicio, optimizando el uso eficiente de los recursos hídricos y nutrientes. Asimismo,

facilita la optimización del riego, la fertilización y la gestión del cultivo en tiempo real, promoviendo prácticas agrícolas más eficientes y sostenibles (wiki.dfrobot.com, 2025).

2.4.1 Tipos de conexiones IoT

Existen varios tipos de conexiones utilizadas en IoT, cada una adecuada para diferentes necesidades, como el alcance, el consumo de energía y la capacidad de transmisión de datos. Las conexiones de corto alcance son ideales para dispositivos cercanos y de bajo consumo energético, como Wi-Fi, Bluetooth y Zigbee. Estas tecnologías permiten la transmisión de datos a distancias limitadas, lo que las hace perfectas para entornos domésticos, oficinas y dispositivos portátiles que no requieren grandes volúmenes de datos. Las conexiones de largo alcance (LPWAN), como LoRaWAN, Sigfox y NB-IoT, son especialmente útiles en situaciones donde se necesita conectar dispositivos a distancias más grandes sin consumir mucha energía. Estas redes son perfectas para áreas rurales, agrícolas y ciudades inteligentes, donde los dispositivos pueden estar dispersos y el consumo de energía debe ser mínimo. Las redes celulares, que incluyen 2G, 3G, 4G y 5G, proporcionan una mayor velocidad y capacidad de transmisión de datos, lo que las hace adecuadas para aplicaciones que requieren un mayor ancho de banda, como el monitoreo en tiempo real, la transmisión de videos y la comunicación constante con grandes volúmenes de datos. La conexión satelital es otra opción útil, especialmente en áreas remotas o sin infraestructura terrestre, ya que permite la comunicación mediante satélites. Esta tecnología es ideal para aplicaciones en zonas rurales, marítimas o incluso en la exploración espacial. Finalmente, las conexiones Ethernet son muy utilizadas en entornos industriales y de oficina que requieren conexiones estables, rápidas y de gran capacidad para el manejo de grandes cantidades de datos de manera constante. Estas redes proporcionan un rendimiento óptimo y son perfectas para sistemas que necesitan alta fiabilidad y baja latencia, como en fábricas inteligentes o centros de datos (Bernal & Enrique, 2020).

2.4.2 En la domótica

Esta categoría abarca el uso de dispositivos para el control remoto, permitiendo encender y apagar electrodomésticos a distancia, lo cual contribuye a prevenir accidentes y optimizar el consumo energético. También incluye la supervisión del uso de recursos como energía y agua, brindando información útil para reducir costos y fomentar un consumo responsable.

Además, se enfoca en la protección de obras de arte y propiedades mediante sistemas que monitorean las condiciones internas de museos, asegurando el estado de conservación de las piezas. Por último, incorpora sistemas de detección de intrusos en almacenes y edificios, los cuales identifican aperturas o movimientos en puertas y ventanas para prevenir accesos no autorizados (Burbano Choez, 2022).

2.4.3 Ambientes Inteligentes

Esta categoría engloba diversos aspectos fundamentales relacionados con el monitoreo y la prevención ambiental. Entre ellos, se encuentra la alerta temprana de terremotos, que utiliza sistemas distribuidos en puntos estratégicos para detectar actividad sísmica. También incluye la prevención de deslizamientos y avalanchas mediante el control de factores como la humedad del suelo, las vibraciones y la densidad del terreno, con el objetivo de anticiparse a situaciones de riesgo. El seguimiento del nivel de nieve permite evaluar en tiempo real la calidad de las pistas de esquí y garantizar la seguridad frente a posibles avalanchas. Asimismo, se aborda la detección de incendios forestales mediante el análisis de gases de combustión y las condiciones del fuego para delimitar áreas de peligro. Por último, se monitorea la contaminación del aire, controlando las emisiones de CO₂ de fábricas, la polución vehicular y otros gases tóxicos (Burbano Choez, 2022).

2.4.4 Transporte y Logística

Esta categoría comprende la detección de eventos relacionados con el almacenamiento, como el monitoreo de emisiones en contenedores que almacenan materiales inflamables y están ubicados junto a otros que contienen sustancias explosivas. También incluye el seguimiento de flotas, permitiendo rastrear rutas de transporte de cargas sensibles, como joyas, productos farmacéuticos o materiales peligrosos. Además, abarca la localización de artículos específicos en grandes espacios, como almacenes o puertos. Asimismo, se evalúa la calidad de las condiciones de envío, considerando aspectos relevantes para seguros, como la detección de vibraciones, impactos, aperturas de contenedores o su mantenimiento (Burbano Choez, 2022).

2.4.5 Agricultura Inteligente

Esta categoría abarca diferentes aplicaciones, como el monitoreo del abono, donde se controla la humedad y temperatura del heno, la paja y otros materiales para evitar la proliferación de hongos y microorganismos. También incluye el uso de estaciones meteorológicas para analizar las condiciones climáticas en los campos, lo que permite anticipar fenómenos como lluvias, sequías, nieve o formación de hielo. En los campos de golf, se realiza un riego selectivo en áreas áridas para optimizar el consumo de agua. Por otro lado, en los invernaderos, se regulan las condiciones del microclima con el objetivo de mejorar la calidad y el rendimiento de frutas y hortalizas. Finalmente, en sistemas hidropónicos, se supervisa el crecimiento de las plantas en soluciones acuosas para obtener cosechas más eficientes (Burbano Choez, 2022).

2.4.6 Tecnología wifi

El Wi-Fi es una tecnología inalámbrica que utiliza los estándares IEEE 802.11 para la transmisión de datos sin cables, cumpliendo con las especificaciones del IEEE. Su principal ventaja frente a otras opciones de menor consumo radica en su compatibilidad nativa con redes IP, lo que la hace esencial para las redes IoT. Además, su amplia adopción en redes LAN proporciona acceso a herramientas avanzadas y simplifica su integración (Barreiro & Martínez, 2024).

2.4.7 Aplicaciones IoT en tiempo real para Optimización de Procesos

Las aplicaciones en tiempo real basadas en la IoT se han consolidado como herramientas adecuadas para la adaptación de procesos en diversas industrias. En estos sistemas, la plataforma IoT desempeña un papel crucial al facilitar la conexión, gestión y análisis inmediato de la información generada por los dispositivos y sensores, lo que permite una toma de decisiones ágil y precisa. Gracias a su capacidad para procesar y transmitir datos en tiempo real, estas plataformas permiten el monitoreo y control dinámico de factores críticos, como temperatura, humedad, niveles de nutrientes y otros parámetros relevantes, lo que a su vez optimiza el uso de recursos y mejora la productividad. De este modo, un sistema centralizado permite gestionar y supervisar de manera eficiente todos los procesos, garantizando una respuesta rápida y precisa ante cualquier cambio o necesidad.

Existen diversas aplicaciones de interfaz IoT en tiempo real que permiten la visualización,

monitoreo y control eficiente de los sistemas IoT en tiempo real. Aquí te menciono algunas de las más populares:

2.5 Grafana

Grafana es una plataforma de código abierto para la representación y estudio de datos en tiempo real. Se utiliza comúnmente para monitorear aplicaciones, infraestructuras y sistemas IoT, permitiendo crear dashboards de control interactivos e individualizado para visualizar métricas, logs y otros datos de manera eficiente. A continuación, te proporciono más detalles sobre las características y el uso de Grafana (grafana.com, 2025):

2.5.1 Características Principales de Grafana:

2.5.1.1 Visualización de Datos en Tiempo Real:

- Grafana posibilita la creación de gráficos interactivos y paneles de control en tiempo real, mostrando métricas actualizadas constantemente.
- Soporta una amplia gama de visualizaciones, como gráficos lineales, métricas, tablas, alertas, y entre otros.

2.5.1.2 Integración con Múltiples Fuentes de Datos:

- Grafana puede conectarse a diversas fuentes de datos como InfluxDB, Prometheus, MySQL, PostgreSQL, Elasticsearch, CloudWatch, Google Cloud Monitoring, Azure Monitor, y más.
- Esto permite una visualización centralizada de datos provenientes de diferentes sistemas y bases de datos (grafana.com, 2025).

2.5.1.3 Alertas:

- Puedes configurar alertas basadas en reglas, de modo que si los valores superan un umbral predefinido (por ejemplo, si la concentración de CO2 supera un nivel crítico), Grafana puede enviar notificaciones por correo electrónico,

Slack, OpsGenie, o a través de webhooks.

- Estas alertas son necesarias para el seguimiento en tiempo real de sistemas y para evitar eventos inesperados.

2.5.1.4 Interactividad y Personalización:

- Los paneles de control creados en Grafana son interactivos, permitiendo que los usuarios filtren datos por rango de tiempo, etiquetas o parámetros.
- Además, Grafana permite la personalización de paneles para satisfacer necesidades específicas, como cambiar colores, agregar títulos o elegir diferentes tipos de gráficos (grafana.com, 2025).

2.5.2 Facilidad de Instalación y Configuración:

- Grafana es fácil de instalar y configurar. Está disponible en múltiples plataformas como Linux, macOS, y Windows, y puede instalarse en servidores locales o en contenedores Docker.
- Además, Grafana tiene una interfaz web intuitiva que permite crear y gestionar paneles sin necesidad de escribir código.

2.5.2.1 Paneles de Control y Dashboards:

- Los paneles de control pueden ser diseñados para mostrar información en tiempo real de manera clara y comprensible.
- Puedes agregar diferentes "paneles" (widgets) a un único dashboard, cada uno mostrando diferentes visualizaciones de datos.

2.5.2.2 Seguridad:

- Grafana ofrece opciones de seguridad como verificación de usuario, roles y permisos, para controlar el acceso a los datos y paneles de control.

2.5.3 Casos de Uso Comunes:

2.5.3.1 Monitoreo de Infraestructura:

- Grafana es ampliamente utilizado para monitorear servidores, redes y aplicaciones. Los usuarios pueden visualizar el uso de CPU, memoria, discos y otros parámetros clave de la infraestructura.

2.5.3.2 Análisis de IoT:

- En proyectos de IoT (como en el análisis de nutrientes del suelo), Grafana puede integrarse con bases de información de series temporales como InfluxDB para crear dashboards que muestren las lecturas de los sensores (temperatura, humedad, partículas en el aire, etc.) en tiempo real (grafana.com, 2025).

2.5.3.3 Análisis de Logs:

- Grafana puede integrarse con plataformas de logs como Elasticsearch estudiar y observar logs en tiempo real, ayudando en la detección de errores o problemas en los sistemas.

2.5.3.4 Visualización de Datos de Negocio:

- Muchas empresas utilizan Grafana para analizar métricas de negocio en tiempo real, como ventas, tráfico web, conversiones, etc.

2.5.4 Ventajas de Usar Grafana:

- Código abierto y gratuito: Grafana es completamente gratuito si se utiliza con la versión de código abierto. También ofrece opciones de pago para características avanzadas y soporte empresarial.
- Extensible y modular: La comunidad de Grafana ha creado numerosos complementos para ampliar la funcionalidad, como paneles de visualización, fuentes de datos y alertas.
- Fácil de integrar con otras herramientas: Puedes integrarlo fácilmente con otras herramientas de monitoreo o sistemas IoT como Prometheus, InfluxDB, y bases de

datos SQL/NoSQL (grafana.com, 2025).

2.5.5 Cómo Usarlo con InfluxDB:

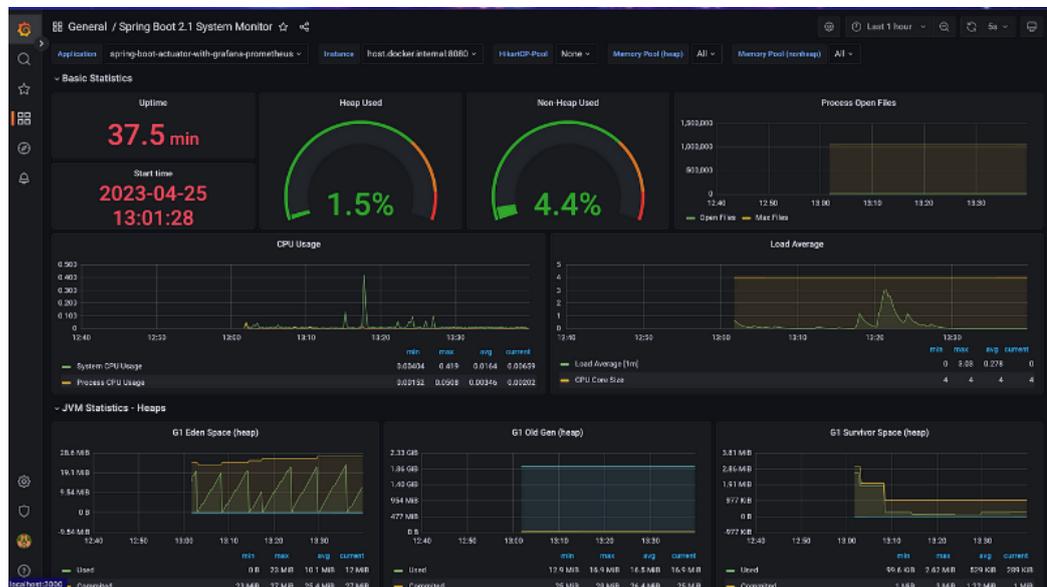
Grafana se integra perfectamente con InfluxDB, lo que es muy útil para proyectos de IoT donde se manejan grandes cantidades de datos de series temporales (por ejemplo, mediciones de calidad del aire). Para configurar Grafana con InfluxDB, puedes seguir estos pasos básicos:

1. Instalar Grafana en tu máquina o servidor.
2. Instalar InfluxDB y configurar la base de información para guardar los datos de los sensores.
3. Conectar Grafana a InfluxDB:
 - En la interfaz de Grafana, agrega InfluxDB como fuente de datos.
 - Configura la URL de tu servidor InfluxDB y la base de datos que contiene los datos de los sensores.
4. Crear un Dashboard:
 - Crea un nuevo panel en Grafana, selecciona la fuente de datos (InfluxDB), y luego construye las consultas para obtener los datos de los sensores.
5. Visualización:
 - Agrega paneles de visualización, como gráficos de líneas o tablas, para mostrar los datos de forma clara y comprensible.
6. Configurar Alertas:
 - Si es necesario, configura alertas en Grafana para que notifique a los usuarios si los datos de los sensores superan un umbral crítico (grafana.com, 2025).

En la figura 2 se observa la pantalla principal con toda la información obtenida de los sensores ya configurados.

Figura 2

Plataforma Grafana



Nota. (grafana.com, 2025)

2.6 Influxdb

InfluxDB es una base de información de series temporales de código abierto, diseñada para manejar grandes volúmenes de información que están relacionados con el tiempo. Es usualmente utilizada en aplicaciones de monitoreo, análisis de métricas y sistemas IoT debido a su capacidad para almacenar, consultar y analizar datos con marcas de tiempo de manera eficiente (influxdata.com, 2025).

2.6.1 Características Principales de InfluxDB:

1. Optimizada para Series Temporales:

- InfluxDB está especialmente diseñada para manejar series temporales, es decir, datos que se registran en intervalos regulares de tiempo, como las mediciones de sensores de IoT, métricas de servidores, o datos de monitoreo de sistemas.
- Es ideal para almacenar y consultar datos como temperatura, humedad,

presión, etc., de forma eficiente.

2. Alto Rendimiento:

- InfluxDB puede manejar un gran volumen de datos de series temporales a alta velocidad de inserción, lo que la convierte en una opción ideal para sistemas que generan grandes cantidades de datos en tiempo real, como proyectos IoT.
- Soporta inserciones de millones de puntos de datos por segundo.

3. Consulta de Datos con el Lenguaje InfluxQL:

- InfluxDB utiliza un lenguaje de consulta similar a SQL llamado **InfluxQL**. Esto permite realizar consultas complejas y personalizadas sobre los datos almacenados, como la selección de un rango de tiempo, agregaciones, filtros, y cálculos matemáticos.
- Además, se está introduciendo un nuevo lenguaje de consulta llamado **Flux**, que es más potente y flexible para tareas avanzadas de análisis de datos.

4. Facilidad de Integración:

- InfluxDB se integra fácilmente con diversas plataformas de monitoreo y visualización como **Grafana**, **Chronograf** y **Kapacitor**.
- Esto la convierte en una opción popular para la creación de sistemas de monitoreo en tiempo real, dashboards y alertas.

5. Almacenamiento de Datos Eficiente:

- InfluxDB utiliza una estructura de almacenamiento optimizada para series temporales, lo que la hace más eficiente que las bases de datos tradicionales cuando se trata de almacenar datos basados en tiempo.
- Los datos se almacenan en "buckets" (cubos), que son particiones lógicas de tiempo, para mejorar la productividad en el análisis de las consultas.

6. Retención de Datos y Compresión:

- InfluxDB permite establecer políticas de retención de datos para controlar cuánto tiempo se almacenan los datos. Esto es útil en aplicaciones que no necesitan conservar todos los datos indefinidamente.
- También incluye mecanismos de compresión para minimizar el volumen de almacenamiento de los datos (influxdata.com, 2025).

7. Alertas y Acciones:

- InfluxDB, a través de **Kapacitor** (un componente de la suite TICK), puede configurarse para generar alertas cuando los datos superan ciertos umbrales, permitiendo la ejecución de acciones automáticas como el envío de notificaciones por correo electrónico o la integración con otros sistemas.

2.6.2 Casos de Uso Comunes:

1. Monitoreo de Infraestructura:

- InfluxDB es ampliamente utilizada para almacenar y analizar métricas de servidores, aplicaciones, redes y bases de datos. Puede usarse para medir el uso de CPU, la memoria, el almacenamiento y otros recursos del sistema.

2. Monitoreo de IoT:

- Es ideal para sistemas IoT que generan grandes cantidades de datos de sensores. En tu caso, para un proyecto de calidad del aire en un consultorio, InfluxDB almacenaría las lecturas de sensores como PM2.5, CO2, temperatura y humedad a lo largo del tiempo.

3. Análisis de Datos en Tiempo Real:

- En aplicaciones de análisis en tiempo real, como la monitorización de procesos

industriales o el seguimiento de eventos deportivos, InfluxDB permite obtener información instantánea y realizar análisis detallados con consultas de series temporales.

4. Visualización de Datos con Grafana:

- InfluxDB se utiliza comúnmente en conjunto con **Grafana**, una herramienta de visualización, para crear paneles de control interactivos que muestren los datos de manera clara y comprensible (influxdata.com, 2025).

2.6.3 Estructura de Datos en InfluxDB:

1. Measurement (Medición):

- El término "Measurement" en InfluxDB es similar a una tabla en una base de datos SQL. Representa una colección de puntos de datos de una métrica específica (por ejemplo, "temperatura", "humedad", "calidad del aire").

2. Tags:

- Los **tags** son pares de clave-valor indexados que pueden usarse para filtrar y agrupar los datos. Por ejemplo, se puede tener un tag para "ubicación" (consultorio 1, consultorio 2) o "sensor" (sensor de CO2, sensor de temperatura).

3. Fields:

- Los **fields** son los datos medidos, como la velocidad del viento, el pH o el valor de partículas PM2.5. Los fields no están indexados, por lo que se pueden realizar operaciones matemáticas sobre ellos (sumas, promedios, etc.).

4. Time:

- InfluxDB asocia cada punto de datos con una marca de tiempo. Este es uno de los aspectos más importantes, ya que permite almacenar y consultar datos con relación al tiempo (influxdata.com, 2025).

Figura 3

Plataforma InfluxDB



Nota. (influxdata.com, 2025)

2.7 Raspberry Pi

La Raspberry Pi 4 es una de las versiones más potentes y avanzadas de la popular familia de microcontrolador de placa única (SBC, por sus siglas en inglés) Raspberry Pi. Es una plataforma versátil y de bajo costo, ideal para proyectos de aprendizaje, desarrollo de prototipos, en el Internet de las Cosas, y el uso de computación personal o profesional.

A continuación, detallamos su principales cualidades, características, especificaciones, usos y ventajas principales de la Raspberry Pi 4:

2.7.1 Características Principales de Raspberry Pi 4:

1. Procesador (CPU):

- **Modelo:** Broadcom BCM2711, Quad-core Cortex-A72 (ARMv8) de 64 bits.
- **Velocidad:** 1.5 GHz.

- **Arquitectura:** ARM Cortex-A72 de 4 núcleos, lo que mejora significativamente su rendimiento.
- Es capaz de manejar múltiples tareas simultáneamente de manera más eficiente gracias a su procesador más potente.

2. Memoria RAM:

- Existen variantes con diferentes configuraciones de RAM:
 - 2 GB
 - 4 GB
 - 8 GB (lanzada más recientemente, para proyectos más exigentes).
- La memoria LPDDR4-3200 ofrece un rendimiento notablemente mejorado, ideal para tareas multitarea y proyectos que requieren más memoria.

3. Gráficos (GPU):

- **Modelo:** VideoCore VI.
- La Raspberry Pi 4 incluye soporte para gráficos de alta calidad, 4K a 60 Hz, lo que la posiciona en una buena opción sólida para proyectos multimedia y aplicaciones de video.

4. Puertos USB:

- **2 puertos USB 3.0:** Permite una transferencia de información más rápida con dispositivos como discos duros externos, teclados, ratones, etc.
- **2 puertos USB 2.0:** Para dispositivos de menor velocidad de transferencia, como teclados y ratones básicos.
- Compatibilidad con una variedad de dispositivos USB.

5. Conexión de Pantalla:

- **2 puertos micro-HDMI** (hasta 4K a 60 Hz).
 - Estos puertos permiten conectar hasta dos pantallas, lo que es útil para aplicaciones de monitoreo, computación o proyectos de visualización avanzada.

6. Red:

- **Ethernet Gigabit:** Ofrece una conexión de red rápida y estable para proyectos que requieren alta velocidad de transmisión de datos.
- **Wi-Fi 802.11ac (Dual-band):** Conectividad inalámbrica rápida y confiable (2.4 GHz y 5 GHz).
- **Bluetooth 5.0:** Permite la conexión con piezas de hardware adicionales como sensores, pantallas, mouses y entre otros.

7. Almacenamiento:

- **MicroSD Card Slot:** El sistema operativo y los datos se almacenan en una tarjeta microSD, que debe tener al menos 8 GB para instalar un sistema operativo como Raspberry Pi OS.
- **Soporte para almacenamiento USB:** También se puede conectar un disco duro o unidad flash por medio de los puertos USB 3.0 que le permite mayor habilidad de almacenamiento.

8. Puerto GPIO (General Purpose Input/Output):

- **40 pines GPIO:** Le da la capacidad de conectar y controlar dispositivos electrónicos como pantallas, variadores de voltaje y otros componentes electrónicos en proyectos de IoT, automatización y robótica.

- Ofrece una interfaz directa para trabajar con proyectos hardware sin necesidad de una placa adicional.

9. **Audio:**

- **Salida de audio estéreo a través de Jack de 3.5 mm.**
- **Soporte para audio HDMI** para conectividad con pantallas que soportan audio a través del puerto HDMI.

10. **Alimentación:**

- La Raspberry Pi 4 requiere un **adaptador de corriente USB-C** de 5V/3A para alimentar el dispositivo.

11. **Sistema Operativo:**

- **Raspberry Pi OS (anteriormente Raspbian):** Tiene un sistema operativo oficial basado en Linux, perfecto para proyectos formativo, constructivo y aplicaciones generales.
- También se puede usar otras distribuciones de Linux como Ubuntu, o incluso sistemas operativos como **Windows IoT Core**.

2.7.2 **Ventajas de Raspberry Pi 4:**

1. **Rendimiento Mejorado:**

- Con su CPU de 4 núcleos y memoria RAM más rápida, la Raspberry Pi 4 puede manejar tareas más complejas y aplicaciones que anteriormente solo podrían haberse ejecutado en computadoras de escritorio.

2. **Conectividad Avanzada:**

- Su capacidad de conectividad, incluyendo puertos USB 3.0, HDMI dual,

Ethernet gigabit y Wi-Fi, lo hace ideal para aplicaciones de red, proyectos multimedia y trabajo en equipos distribuidos.

3. **Flexibilidad:**

- Raspberry Pi 4 puede usarse en una variedad de aplicaciones, desde un servidor de medios, una estación de trabajo de desarrollo, hasta un nodo en una red IoT.
- Se puede usar para crear prototipos de hardware o conectar múltiples sensores y dispositivos para aplicaciones en tiempo real.

4. **Bajo Costo:**

- A pesar de sus capacidades avanzadas, la Raspberry Pi 4 sigue siendo una opción asequible en comparación con computadoras tradicionales y sistemas embebidos de alto rendimiento.

5. **Amplia Comunidad:**

- La comunidad de Raspberry Pi es extensa y activa. Tiene una enorme cantidad de recursos, cursos, proyectos y soporte para resolver dudas o problemas.
- Hay una gran cantidad de accesorios disponibles, como pantallas táctiles, carcasas, cámaras, y más.

6. **Ideal para Proyectos IoT:**

- Con el soporte de sensores a través de GPIO, la capacidad de conectarse a Internet (Wi-Fi, Ethernet), y la capacidad de procesamiento, la Raspberry Pi 4 es ideal para ser el centro de proyectos IoT (Pi, 2024).

2.7.3 **Aplicaciones Comunes de Raspberry Pi 4:**

1. **Proyectos IoT (Internet de las Cosas):**

- Comprobación y monitorización de sensores y actuadores.
- Creación de servidores locales o gateways para enviar datos a la nube.
- Implementación de sistemas de monitorización en tiempo real para variables como pH, velocidad de viento, nivel nutricional, etc.

2. **Estaciones de Trabajo de Desarrollo:**

- Usada como una estación de desarrollo ligera para programadores, estudiantes o proyectos de aprendizaje.
- Ejecución de entornos de desarrollo como Python, JavaScript, Node.js y otros lenguajes.

3. **Servidor de Medios (Media Center):**

- Usada como servidor de medios en casa con software como **Plex**, **Kodi** o **OpenMediaVault**.

4. **Educación y Aprendizaje:**

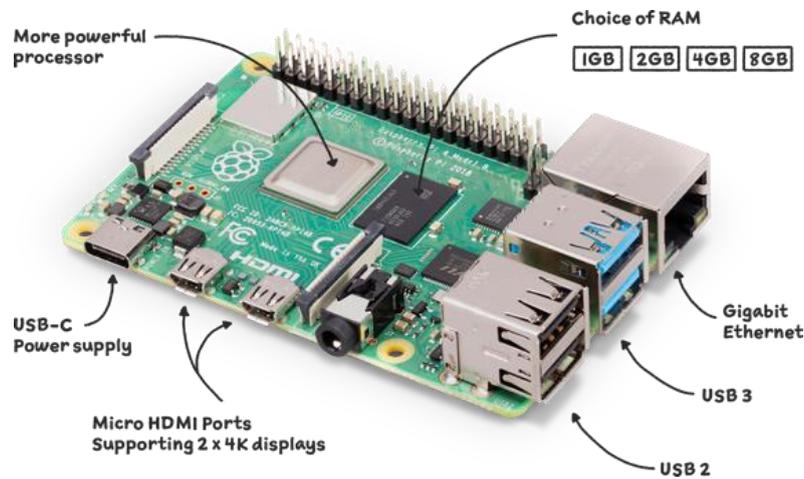
- Ideal para enseñanza de programación, robótica y electrónica a estudiantes de todas las edades.
- Usada en aulas como una computadora asequible y flexible para aprendizaje.

5. **Computación de Escritorio:**

- Puedes usarla como una computadora de escritorio básica para tareas como navegación web, procesamiento de texto, y más (Pi, 2024).

Figura 4

Raspberry pi 4



Nota. (Raspberry, 2024)

2.8 Sensor NPK

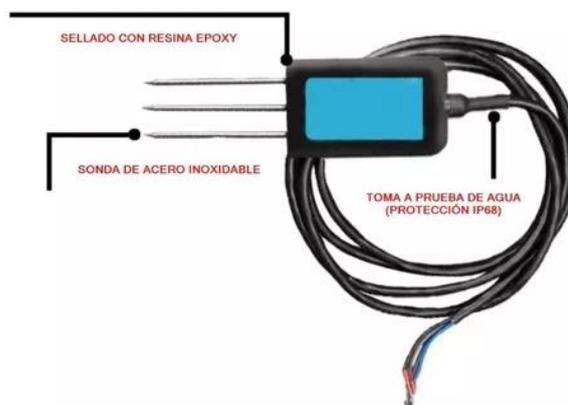
El sensor NPK es un dispositivo electrónico que calcula los valores de Potasio, Nitrógeno y Fosforo en la tierra, para determinar si el suelo es apto para la agricultura.

Uno de los beneficios por lo cual se destacan los sensores NPK es que se pueden colocar en distintos sectores del campo agrícola a evaluar, detectando la constante dieléctrica del suelo, calculan los niveles de NPK, la humedad del suelo y envían los datos en tiempo real, esto ayuda al agricultor a ajustar el nivel de fertilizante a usar en los cultivos.

En la **figura 5** se observa un modelo de sensor NPK y en **tabla 4** se observa sus especificaciones técnicas (ozonomajes.com, 2025)

Figura 5

Sensor NPK



Nota. (ozonomajes.com, 2025)

En la **tabla 4** se observan las especificaciones técnicas del sensor NPK para su óptimo funcionamiento.

Tabla 4.

Especificaciones técnicas del sensor NPK

Especificaciones técnicas del sensor NPK		
Alimentación DC (Predeterminada)	DC 5-30V	
Consumo máximo de energía	≤0.15W (@12V DC, 25°C)	
Temperatura de funcionamiento	0°C~55°C	
Parámetros NPK	Rango	1-1999 mg/kg(mg/L)
	Resolución	1 mg/kg(mg/L)
	Precisión	±2%FS
Tiempo de respuesta	<1S	
Clase de protección	IP68	
Material de sonda	Acero inoxidable	
Material de sellado	Black flame-retardant epoxy resin	

Longitud de cable predeterminado	Dos metros, la longitud del cable se puede personalizar
Dimensiones	45*15*123mm
Señal de salida	RS485(Modbus protocol)

Nota. Especificaciones técnicas del sensor NPK (ozonomajes.com, 2025)

2.9 Lark Weather Station

El Lark Weather Station es una solución integral para la obtención exacta de datos meteorológicos, implicando el proceso de registro y medición de datos atmosféricos claves, como la dirección del viento, la velocidad de viento, la humedad, la temperatura y la presión atmosférica. Entre su característica principal es su fácil compatibilidad con los microcontroladores más populares como la Raspberry Pi, la ESP32, la micro: bit y con las placas en desarrollo de Uniliker, y su espacio de almacenamiento compacto integrado de 16M idóneo para el almacenamiento de datos por 160 días a una velocidad y registro de una vez por minuto, la obtención de estos datos es a través de una interconexión tipo C (community.dfrobot.com, 2025).

El sistema portátil Lark Weather Station resuelve mucho de los problemas de la estación meteorológicas actuales, eso se debe a que el LWS es un sistema móvil, de pequeño tamaño, peso y no requiere de mucho cableado para su montaje, eso ayuda a los profesores y estudiantes que están interesados en el estudio meteorológico se realice de manera más practica y sencilla.

En la **figura 6** se observa como el LWS es un equipo 5 en 1 capaz de recibir información precisa y en tiempo real, de una forma más rápida y precisa.

Figura 6

Lark Weather Station



Nota. Sensores de temperatura, humedad, presión atmosférica, dirección del viento y velocidad del viento (wiki.dfrobot.com, 2025)

Como se nombró anteriormente la estación Lark puede conectarse con los microcontroladores más populares en la actualidad como Raspberry Pi, ESP32, Unihiker, Arduino y micro: bit.

Esto ayuda a que el microcontrolador tenga mayor flexibilidad en la toma de decisiones, que proporcionara un uso ilimitado para sus distintos modos de uso, como predecir el clima, activación de sistemas de riego y crear estación de clima digital en línea.

Figura 7

Conexión LWS con microcontroladores



Nota. Conexión del Lark Weather Station con distintos tipos de controladores (wiki.dfrobot.com, 2025)

En la **tabla 5** se observa la compatibilidad que tiene los microcontroladores y que lenguaje de programación se puede utilizar para el buen funcionamiento del Lark Weather Station.

Tabla 5

Compatibilidad LWS con microcontroladores

COMPATIBILIDAD					
	C++	Python	Arduino IDE	MindPLUS	Makecode
Arduino	X		X	X	
micro: bit V1/V2				X	X
ESP32	X		X		
Raspberry Pi		X			
UNIHKER		X		X	

Nota. Compatibilidad del LWS con diferentes tipos de microcontroladores y su lenguaje de programación (community.dfrobot.com, 2025)

Las especificaciones técnicas del Lark Weather Station se observa en la **tabla 6**:

Tabla 6

Especificaciones técnicas Lark Weather Station

Especificaciones Técnicas Lark Weather Station

Voltaje De Alimentación	3,3 -5,5 V Dc
Corriente De Trabajo	40 Ma
Corriente De Suelo	2 Ma
Modo De Comunicación	I2c UART
Velocidad De Viento	0,5 - 12 M/S
Dirección De Viento	8 direcciones
Temperatura	-20 - 60°C ± 0,2°C
Humedad	0 - 99% Hr ± 2% Hr
Presión De Aire	300 - 1100 HPa ± Pa
Disco USB	16m
Tamaño Del Cuerpo Principal	160 x 55 Mm
Peso Del Cuerpo Principal	270 g

Nota. Especificaciones técnicas del LWS y rango de medición de sensores involucrados (wiki.dfrobot.com, 2025)

2.10 Módulo USB a RS-485

El módulo USB a RS-485 es un dispositivo electrónico que permite la comunicación entre un computador o una plataforma embebida (como una Raspberry Pi) y un dispositivo o red que utiliza el protocolo de comunicación RS-485. Este módulo convierte las señales USB en señales RS-485, permitiendo la conexión y comunicación en aplicaciones industriales, automatización, y otros sistemas que requieren transmisión de datos a larga distancia y en entornos ruidosos.

RS-485 es un estándar de comunicación diferencial utilizado ampliamente en aplicaciones industriales y de automatización, que permite transmitir datos de forma robusta a través de cables de dos hilos, proporcionando ventajas como mayor distancia de comunicación, inmunidad al ruido y capacidad de interconectar múltiples dispositivos en una misma red.

2.10.1 Características del Módulo USB a RS-485:

El módulo USB a RS-485 de Waveshare tiene varias características que lo hacen adecuado para aplicaciones que requieren comunicación serial robusta y de alta calidad. A continuación, se describen su funcionamiento como sus partes instaladas en este módulo:

1. Interfaz de Conexión USB:

- El módulo se conecta al computador o dispositivo mediante un puerto USB estándar, lo que lo hace fácil de integrar en sistemas existentes sin necesidad de interfaces adicionales.
- El USB proporciona alimentación al módulo, lo que facilita su uso en dispositivos con puertos USB, eliminando la necesidad de fuentes de alimentación adicionales.

2. Conversión de Señales RS-485:

- El módulo convierte las señales USB en señales RS-485, permitiendo la comunicación con dispositivos que utilizan este protocolo estándar en entornos industriales o de automatización.
- RS-485 es un protocolo diferencial, lo que significa que las señales se transmiten a través de un par de cables, lo que reduce la interferencia y permite la transmisión de datos a largas distancias (hasta 1200 metros) con una alta tasa de datos.

3. Comunicación de Larga Distancia:

- RS-485 es conocido por su capacidad para transmitir datos a distancias largas (hasta 1200 metros), lo que lo convierte en una opción ideal para aplicaciones industriales que requieren interconexión de dispositivos distribuidos en grandes áreas.

4. Conectividad Multidrop:

- El módulo soporta una topología multidrop, lo que significa que puede conectar varios dispositivos a un solo bus RS-485. Esta característica es útil en redes industriales donde es necesario conectar múltiples dispositivos a un solo controlador o unidad central.

5. Compatibilidad con Varios Sistemas Operativos:

- El módulo USB a RS-485 es compatible con diversos sistemas operativos, como Windows, Linux y Mac OS X, lo que lo hace fácil de usar en diferentes plataformas sin necesidad de controladores o configuraciones complejas.

6. Baud Rate y Configuración Flexible:

- El módulo soporta diferentes tasas de transferencia de datos (baud rates), lo que lo hace adaptable a diversas aplicaciones y sistemas que operan a diferentes velocidades.
- Es configurable mediante software para adaptarse a las necesidades de comunicación de cada proyecto.

7. Construcción Robusta:

- El diseño del módulo es compacto y resistente, lo que le permite operar en condiciones industriales difíciles, donde los dispositivos pueden estar sujetos a vibraciones, polvo o interferencias eléctricas.

8. Indicación LED:

- El módulo está equipado con indicadores LED que muestran el estado de la transmisión y recepción de datos, lo que facilita la monitorización y diagnóstico durante la operación.

2.10.2 Teoría del Protocolo RS-485:

El RS-485 es un estándar de comunicación serial diferencial utilizado para transmitir datos a larga distancia en entornos ruidosos, lo que lo hace ideal para sistemas industriales y de automatización. A continuación, describiremos los elementos clave para el funcionamiento de este protocolo:

1. Comunicación Diferencial:

- RS-485 utiliza un par de cables para transmitir las señales, lo que se denomina comunicación diferencial. Esto significa que las señales se transmiten en dos líneas (A y B), con una diferencia de voltaje entre ellas. Este tipo de comunicación reduce el impacto de las interferencias electromagnéticas, lo que permite transmitir datos a largas distancias sin degradación significativa de la señal.

2. Multipunto:

- RS-485 soporta la conexión de varios dispositivos en una misma red, lo que permite la configuración de topologías multipunto. Esto significa que varios dispositivos pueden compartir el mismo par de cables para comunicarse con un controlador central, lo que reduce el cableado necesario en grandes instalaciones.

3. Distancia y Velocidad de Transmisión:

- RS-485 permite una comunicación de larga distancia (hasta 1200 metros) a tasas de transferencia de hasta 10 Mbps, lo que lo convierte en una opción ideal para redes industriales y sistemas que requieren cobertura amplia.
- A medida que aumenta la distancia, la velocidad de transmisión puede reducirse para mantener la integridad de la señal (waveshare.com, 2025).

4. Protocolos y Aplicaciones:

- RS-485 es el protocolo de comunicación subyacente utilizado en varios protocolos industriales, como Modbus, Profibus y DNP3, que son ampliamente utilizados en automatización industrial, control de procesos y sistemas SCADA (Supervisory Control and Data Acquisition).
- Este protocolo es especialmente útil en sistemas de control de maquinaria, monitoreo de condiciones ambientales, control de sistemas HVAC, y otras aplicaciones industriales donde la fiabilidad y la capacidad de transmitir datos a largas distancias son esenciales.

5. Ventajas de RS-485:

- **Resistencia al ruido:** Debido a su naturaleza diferencial, RS-485 es muy resistente al ruido electromagnético, lo que lo hace ideal para entornos industriales con interferencias.
- **Escalabilidad:** La capacidad de conectar múltiples dispositivos en una sola red facilita la expansión y configuración de sistemas complejos sin necesidad de cables adicionales.
- **Bajo costo:** El cableado de RS-485 es más económico y flexible en comparación con otras opciones de comunicación de larga distancia, como Ethernet o fibra óptica.

Figura 8

Convertidor bidireccional USB a RS485



Nota. Convertidor bidireccional USB a RS485 (externo) (waveshare.com, 2025)

En la figura 9 se observa la estructura interna del convertidor bidireccional USB a RS485, así como los componentes con los que está construido.

Figura 9

Convertidor bidireccional USB a RS485



Nota. Convertidor bidireccional USB a RS485 (interno) (waveshare.com, 2025)

2.11 Controlador de carga solar PWM

El PWM (Pulse With Modulation o Modulación de Ancho de Pulso) es un método utilizado en electrónica para modular el nivel de energía suministrada a dispositivos eléctricos por medio de la variación del ancho de los pulsos de una señal de onda cuadrada. Esta modulación ajusta los tiempos de encendido (T_{on}) y apagado (T_{off}), modificando así el ciclo de trabajo (Duty Cycle), que es la relación entre el tiempo en alto y el tiempo total de la señal.

Al cambiar el Duty Cycle, se varía la tensión media de la señal, lo que permite controlar la intensidad de dispositivos como LEDs (ajustando su brillo), motores de corriente continua (regulando su velocidad) o altavoces y zumbadores (generando sonidos). Un Duty Cycle del 100% mantiene el dispositivo encendido a máxima potencia, mientras que uno del 50% lo hace funcionar a la mitad de su capacidad.

El PWM se destaca por su simplicidad, bajo costo y eficiencia, aunque tiene limitaciones, como la necesidad de coincidir el voltaje nominal con la fuente de energía y una capacidad limitada en corrientes superiores a 60A. En sistemas solares, los reguladores de carga PWM controlan la energía que llega a las baterías, prolongando su vida útil y optimizando el rendimiento del sistema (Megatronica.cc, 2025).

Figura 10

Controlador de carga solar PWM



Nota. Controlador de carga solar PWM (Megatronica.cc, 2025)

2.12 Módulo Step Down XL4015

El módulo step-down XL4015 es un convertidor de voltaje de DC-DC de tipo buck (reductor), diseñado para reducir un voltaje de entrada mayor a un voltaje de salida más bajo con alta eficiencia. Es una solución popular para proyectos de electrónica y sistemas embebidos que requieren convertir una fuente de energía de alto voltaje (como baterías o fuentes de alimentación de 12V, 24V, etc.) a un voltaje más bajo (como 5V, 9V, 12V, etc.) para alimentar componentes como microcontroladores, sensores, actuadores, o circuitos de comunicación.

A continuación, se presentan las principales características y la teoría del funcionamiento del módulo XL4015 (uelectronics.com, 2025).

2.12.1 Características del Módulo Step-Down XL4015:

1. Chip Regulador:

- El módulo Step-Down XL4015 es un controlador de c de voltaje step-down, lo que significa que reduce un voltaje más alto a uno más bajo.

2. Rango de Entrada:

- El módulo Step-Down XL4015 soporta un rango de entrada de 4 a 36V.
- Lo cual lo hace ideal para una gran numero de aplicaciones que utilizan fuentes de energía como paneles solares, baterías y entre otros que utilizan fuentes de corriente continua

3. Rango de Salida:

- El rango de salida del módulo Step-Down XL4015 se puede ajustar dependiendo de las necesidades de los equipos a utilizar tiene un rango de 1.25V a 36V.
- Su corriente máxima de salida suele ser de 5A

4. Eficiencia:

- Una de las características principales del módulo Step-Down XL4015 es su alta eficiencia, debido a que gran parte de la energía que ingresa al dispositivo se convierte en energía útil, lo que evita las perdidas en forma de calor y mejora enormemente la duración de las baterías si se usa en proyectos o aplicaciones portátiles.

5. Protecciones Integradas:

El módulo Step-Down XL4015 cuenta con 3 protecciones, que se activan si sobrepasan un valor predeterminado o ocurre una falla en el sistema, estas 3 protecciones son:

- Protección contra sobrecarga de corriente
- Protección contra el sobrecalentamiento
- Protección contra cortocircuitos

Si una de estas 3 protecciones se llega a activar el modulo XL4015 suele tener la configuración predeterminada de apagarse para proteger el dispositivo electrónico conectado.

6. Control de Potencia:

- El módulo Step-Down XL4015 permite una regulación precisa del voltaje de salida mediante un potenciómetro.

7. Tamaño y Diseño:

- El módulo es compacto, lo que lo hace adecuado para una variedad de aplicaciones de electrónica portátil, automatización o proyectos de energía renovable como sistemas solares, donde se necesita un convertidor eficiente pero pequeño.

3 Teoría de Funcionamiento del Módulo Step-Down XL4015:

El módulo Step-Down XL4015 funciona según el principio de un convertidor buck (o reductor de voltaje). Los convertidores buck son conocidos por su eficiencia y se basan en un ciclo de conmutación rápido para transformar la energía.

3.1.1 Fases del Funcionamiento del XL4015:

1. Conmutación del Transistor:

- El XL4015 utiliza un transistor de conmutación (generalmente un MOSFET) para controlar el flujo de corriente. El transistor se enciende y apaga rápidamente, lo que permite que la energía fluya a través del circuito de manera controlada.

2. Inductor y Almacenamiento de Energía:

- El transistor conectado al inductor almacena energía en forma de campo magnético cuando el transistor está encendido.

3. Condensador de Salida:

- El condensador de salida filtra las fluctuaciones de voltaje que se generan durante la conmutación, produciendo una salida de voltaje más estable para alimentar los dispositivos.

4. Control de la Corriente y Voltaje:

- La cantidad de energía que se transfiere a la carga depende del ciclo de trabajo del transistor. El ciclo de trabajo (duty cycle) es el tiempo durante el cual el transistor permanece encendido en cada ciclo de conmutación.
- Un ciclo de trabajo más largo permite más energía transferida a la carga, lo que aumenta el voltaje de salida. Un ciclo de trabajo más corto reduce el voltaje de salida.

3.1.2 Ventajas del Convertidor Buck:

1. Alta Eficiencia:

- A diferencia de los reguladores lineales, los convertidores buck no convierten la energía en calor de manera significativa. En su lugar, ajustan la energía mediante conmutación, lo que resulta en menos pérdidas y mayor eficiencia.

2. Reducción de Calor:

- El módulo Step-Down XL4015 genera menos calor en comparación con los reguladores de voltaje lineales, lo que lo hace más adecuado para aplicaciones donde el calor es un problema (como en dispositivos alimentados por batería).

3. Capacidad de Manejar Cargas Grandes:

- El módulo Step-Down XL4015 puede manejar cargas de hasta 5A, lo que lo hace adecuado para alimentar dispositivos que requieren una corriente considerable, como motores pequeños, sistemas de iluminación LED de alta potencia, sistemas de comunicación y entre otros.

3.1.3 Aplicaciones Comunes del Módulo Step-Down XL4015:

1. Alimentación de Dispositivos Electrónicos:

- Puede dar alimentación a microcontroladores, sensores, módulos Wi-Fi, pantallas LCD, y otros dispositivos que requieren voltajes más bajos (por ejemplo, 5V, 12V) a partir de una fuente de alimentación más alta.

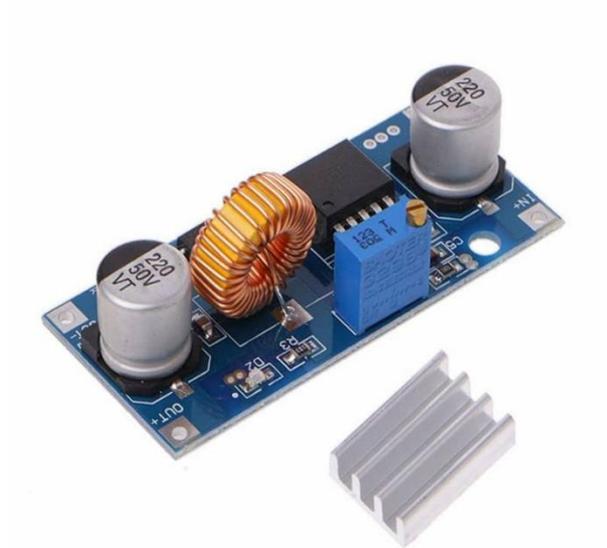
2. Sistemas de Energía Renovable:

- En sistemas solares o eólicos, el módulo Step-Down XL4015 se usa para ajustar el voltaje de las baterías o paneles solares a los niveles requeridos por los dispositivos, como cargadores o controladores de energía.

3. Sistemas de Iluminación LED:

- Usado en aplicaciones de iluminación donde es necesario reducir el voltaje para alimentar tiras o luces LED que requieren voltajes más bajos.

Módulo Step Down



Nota. Modulo Step-Down LX4015 (uelectronics.com, 2025)

3.2 Panel Solar

Un panel solar es un dispositivo que convierte la energía de la luz solar en energía eléctrica utilizando el principio de la fotovoltaica. Los paneles solares son la base de los sistemas de energía solar, que permiten generar electricidad a partir de una fuente de energía renovable y limpia: el sol.

A continuación, se detallan los conceptos clave relacionados con los paneles solares:

3.2.1 Fotovoltaica (Efecto Fotovoltaico):

El efecto fotovoltaico es el principio físico que permite a un panel solar convertir la luz en electricidad. Cuando la luz solar influye sobre ciertos materiales semiconductores, los fotones de la luz estimulan los electrones en el material, generando una corriente eléctrica.

3.2.2 Componentes de un Panel Solar:

- **Celdas Solares:**

- Las celdas solares son los componentes fundamentales de un panel solar. Cada celda solar está formada por materiales semiconductores, que absorben la luz y producen electricidad.
- Las celdas solares suelen estar conectadas en serie y en paralelo para formar un panel solar. Esto permite ajustar el voltaje y la corriente para satisfacer las necesidades del sistema eléctrico.
- **Material Semiconductor (Silicio):**
 - Para la construcción de los paneles solares se utiliza el silicio como material semiconductor, que puede ser de diferentes tipos:
 - **Silicio monocristalino:** Alta eficiencia y coste más elevado.
 - **Silicio policristalino:** Menor costo, pero también menor eficiencia.
 - **Silicio amorfo:** Usado en aplicaciones de baja eficiencia, pero más barato y flexible.
- **Baldosas y Marco:**
 - El panel solar está compuesto por una serie de celdas solares dispuestas en un marco de aluminio o acero, lo que permite que el panel sea resistente a condiciones meteorológicas y fácil de montar.
- **Vidrio:**
 - Un panel solar generalmente tiene una capa de **vidrio templado** en su superficie para proteger las celdas solares de los impactos, la suciedad y las condiciones climáticas, permitiendo al mismo tiempo el paso de la luz hacia las celdas.

3.2.3 Tipos de Paneles Solares:

Los paneles solares se dividen principalmente en tres tipos, según el material y la eficiencia de las celdas solares que utilizan:

- **Paneles Solares Monocristalinos:**

- Están hechos de silicio de una sola cristalización, lo que les da una mayor eficiencia energética (entre el 15% y el 20%).
- Son los más eficientes y duran más tiempo, pero también son los más caros.

- **Paneles Solares Policristalinos:**

- Hechos de silicio con varios cristales fusionados, lo que resulta en menor eficiencia que los monocristalinos (alrededor del 13% al 16%).
- Son más baratos y su rendimiento disminuye más rápidamente con el tiempo.

- **Paneles Solares de Película Delgada:**

- Están hechos de materiales como el telururo de cadmio o silicio amorfo, que permiten un diseño más flexible.
- Tienen menor eficiencia (alrededor del 10% al 12%), pero son más baratos y adecuados para aplicaciones donde el espacio no es un problema.

3.2.4 Generación de Electricidad (DC):

Los paneles solares generan corriente continua (DC), lo que significa que la electricidad fluye en una única dirección. La mayoría de los electrodomésticos y dispositivos que se usan en casa funcionan con corriente alterna (AC), por lo que es necesario usar un inversor para convertir la corriente continua (DC) producida por el panel solar en corriente alterna (AC) utilizable.

3.2.5 Eficiencia de un Panel Solar:

La eficiencia de un panel solar es la capacidad de este para convertir la luz solar en

electricidad útil. Cuanto mayor sea la eficiencia, más energía se puede generar con el mismo espacio. Los paneles más eficientes suelen ser más caros, pero ofrecen más energía a largo plazo.

- Los paneles solares monocristalinos tienen una mayor eficiencia que los policristalinos.
- La eficiencia también depende de otros factores como la temperatura y la radiación solar disponible.

3.2.6 Instalación y Orientación:

Para maximizar la generación de electricidad, los paneles solares deben instalarse en una ubicación con buena exposición al sol. Algunos factores que afectan la eficiencia incluyen:

- **Ángulo de inclinación:** El ángulo de los paneles solares respecto al suelo debe optimizarse según la latitud del lugar y la estación del año.
- **Orientación:** La orientación de los paneles deben siempre estar orientados hacia el sur (en el hemisferio norte) para recibir la mayor cantidad de luz solar durante el día.
- **Sombra:** Es importante evitar que los paneles solares estén sombreados, ya que incluso una pequeña sombra puede reducir significativamente su eficiencia.

-

3.2.7 Potencia y Tamaño de los Paneles Solares:

Los paneles solares se miden en términos de potencia de salida o potencia nominal, que se expresa en vatios (W). Por ejemplo, un panel solar típico puede generar entre 250W y 400W de potencia, dependiendo de su tamaño, eficiencia y condiciones de exposición al sol.

- La cantidad de energía generada por un panel solar depende de la radiación solar disponible en la ubicación geográfica y la orientación del panel.
- Los paneles más grandes pueden generar más electricidad, pero también ocupan más espacio, lo que debe tenerse en cuenta en instalaciones residenciales o comerciales.

3.2.8 Sistema Fotovoltaico:

Un sistema fotovoltaico completo incluye varios elementos además de los paneles solares, como:

- **Inversores:** Convierte la corriente continua (DC) generada por los paneles solares en corriente alterna (AC).
- **Baterías:** En algunos sistemas, las baterías almacenan la energía generada durante el día para su uso durante la noche o en momentos de baja radiación solar.
- **Reguladores de Carga:** Controlan el flujo de energía a las baterías y evitan que se sobrecarguen.
- **Cables y Conectores:** Conectan todos los componentes del sistema.

3.2.9 Ventajas de la Energía Solar:

- **Energía renovable y limpia:** La energía solar es una fuente de energía renovable que no emite gases de efecto invernadero, lo que ayuda a reducir la huella de carbono.
- **Bajos costos operativos:** Una vez instalada, la energía solar es gratuita y requiere poco mantenimiento.
- **Independencia energética:** Los sistemas solares permiten a las viviendas y empresas generar su propia electricidad, reduciendo la dependencia de las fuentes de energía convencionales (Administration, 2020). Ver figura 12.

Figura 12.

Panel solar



Nota. Panel solar fotovoltaico de 15W con herramienta para cargar batería (Administration, 2020)

3.3 Baterías

Son dispositivos electroquímicos que almacenan energía eléctrica, mediante una combinación de plomo y ácido sulfúrico. Su estructura se compone de celdas conectadas entre sí, donde ocurre una reacción química durante los ciclos de carga y descarga, lo que permite el almacenamiento y la liberación de energía.

Estas baterías son útiles por su capacidad para almacenar y suministrar energía. En la industria automotriz, juegan un papel fundamental en el arranque de los vehículos y en el suministro de energía a sus sistemas eléctricos y en el sector de las energías renovables son vitales para almacenar y distribuir energía en paneles solares o sistemas eólicos. Son comúnmente usadas en sistemas de respaldo de energía, garantizando una fuente confiable en caso de cortes de eléctricos. Hay dos tipos de baterías: las baterías abiertas y las baterías selladas.

3.3.1 Baterías abiertas

Está compuesta por celdas que contine dos placas de plomo y actúan como electrodos, sumergidas en un electrolito liquido de ácido sulfúrico diluido, durante los procesos de carga y descarga, estos gases son liberados en el ambiente lo que es peligroso ya que liberan gases extremadamente explosivos.

Además, el ácido sulfúrico, que es corrosivo y tóxico, podría derramarse si la batería se rompe. Son las más económicas y fáciles de fabricar, pero requieren estar en posición horizontal y necesitan un suministro regular de agua para mantener su funcionamiento. No deben ser usadas en lugares cerrados debido a la posibilidad de expulsar gases. A pesar de estas desventajas, ofrecen una alta capacidad de descarga y, si se mantienen adecuadamente, tienen una vida útil prolongada. Son comúnmente utilizadas para el arranque de vehículos (Reddy, 2015).

3.3.2 Baterías selladas

También conocidas como VRLA (baterías de plomo-ácido reguladas por válvula), estas baterías no requieren mantenimiento. Se diferencian de las baterías abiertas en que el electrolito está inmovilizado, utilizando una malla de fibra de vidrio (baterías AGM) o sílice en polvo (baterías de gel) para hacer que el electrolito sea más viscoso. Debido a este diseño, no expulsan gases y pueden ser usadas en espacios cerrados. Además, al no tener electrolito líquido, pueden ser colocadas en cualquier posición y son menos sensibles a golpes o vibraciones (Reddy, 2015)

Figura 13

Batería 12V



Nota. Batería Cs3 de 12V 7A (Reddy, 2015)

4 MARCO METODOLÓGICO

En el presente proyecto de investigación se utilizó un sistema de monitoreo para el cultivo de cacao, basado en el dispositivo Lark Weather Station para el monitoreo ambiental y el sensor NPK para evaluar la calidad del suelo en términos de nutrientes esenciales (nitrógeno, fósforo y potasio). El proceso de desarrollo del cacao se llevó a cabo mediante la recopilación y análisis de datos sobre el estado del ambiente, como temperatura, humedad y condiciones del suelo. A través de estos sensores, se monitorean factores clave para asegurar un crecimiento óptimo de las plantas.

Se desarrollo un prototipo IoT para la monitorización de parámetros ambientales y nutrientes del suelo aplicados a cultivo de cacao, como la temperatura, humedad, nivel de agua y nutrientes, que son captadas mediante la plataforma Grafana Cloud e Influxdb Cloud, que recopila los datos a través de Internet, permitiendo un monitoreo constante del cultivo de cacao. Este prototipo esta diseñado para aprovechar al máximo el espacio y recursos con una perspectiva en el uso del agua y nutrientes esenciales del cultivo.

A continuación, se visualiza el proceso de desarrollo del prototipo lot para un buen cultivo de cacao:

4.1 Descripción del sistema IoT

Este sistema está diseñado para describir el flujo de datos de los sensores meteorológicos y calidad del suelo (NPK) hasta su almacenamiento y visualización en la nube (InfluxDB Cloud y Grafana Cloud) mediante en una Raspberry Pi 4, que se encarga del procesamiento y envío de la información capturada en tiempo real, garantizando la correcta adquisición, y análisis de datos ambientales.

Se visualiza este prototipo IoT mediante un diagrama de flujo **figura 14** que representa la secuencia de operaciones del sistema, desde la recolección de datos almacenados por los sensores hasta su visualización final en un dashboard de Grafana Cloud.

4.1.1 Recolección de datos y funcionamiento de los sensores

4.1.1.1 Sensor NPK (RS-485)

Permite medir o evaluar la calidad de manera precisa sus niveles de suelo a través de la concentración de tres nutrientes importante:

- Nitrógeno (N): Determina la fertilidad del suelo y su capacidad para el crecimiento de cultivos.
- Fósforo (P): Favorece el crecimiento y desarrollo de raíces fuertes y saludables, permitiendo la conversión de energía solar en energía química.
- Potasio (K): Equilibra el balance hídrico de la planta, controlando la transpiración y el intercambio gaseoso, mejorando la resistencia a la sequias o temperaturas extremas de la tierra.

El sensor NPK envía datos a través del protocolo de comunicación serial Modbus RTU (RS-485), permitiendo transmitir a largas distancia, estableciendo comunicación con la Raspberry Pi.

4.1.1.2 Sensores Meteorológicos (I2C)

Los sensores meteorológicos capturan variables ambientales clave:

- Temperatura (°C): Indica la variación térmica del ambiente.
- Humedad Relativa (%): Determina la cantidad de agua en el ambiente

- Velocidad del Viento (m/s): Determina la intensidad del viento.
- Dirección del Viento: Indica el origen del flujo de aire.
- Presión Atmosférica (hPa): Influye en los cambios climáticos y condiciones meteorológicas.
- Altitud (m): Indica la altura sobre el nivel del mar.

Estos sensores utilizan el protocolo I2C para enviar datos directamente a la Raspberry Pi 4.

4.1.2 Procesamiento de Datos en Raspberry Pi 4

La Raspberry Pi 4 actúa como el centro de procesamiento del sistema:

1. Recibe los datos de los sensores NPK a través de un convertidor Modbus USB-RS485.
2. Obtiene los valores meteorológicos mediante I2C.
3. Formatea los datos y los organiza en estructuras adecuadas.
4. Envía los datos a InfluxDB Cloud para su almacenamiento.

4.1.3 Almacenamiento en InfluxDB Cloud

InfluxDB Cloud es una base de datos optimizada para manejar series temporales. En este proyecto:

- Los datos son almacenados en el bucket denominado tesis_bucket.
- La información se organiza por mediciones (weather_measurement_v2 y soil_npk_measurement_v2).
- Se permite la consulta eficiente para la visualización en Grafana Cloud.

4.1.4 Visualización y Alertas en Grafana Cloud

4.1.4.1 Creación de Dashboards en Grafana

Grafana Cloud se conecta a InfluxDB Cloud para obtener los datos y generar gráficos en tiempo real. En este sistema:

- Se configuran paneles interactivos con gráficas de tendencias de NPK y condiciones meteorológicas.
- Se pueden aplicar filtros por fecha y hora para análisis más detallados.

4.1.4.2 Generación de Alertas y Notificaciones

Grafana permite la configuración de alertas basadas en umbrales críticos:

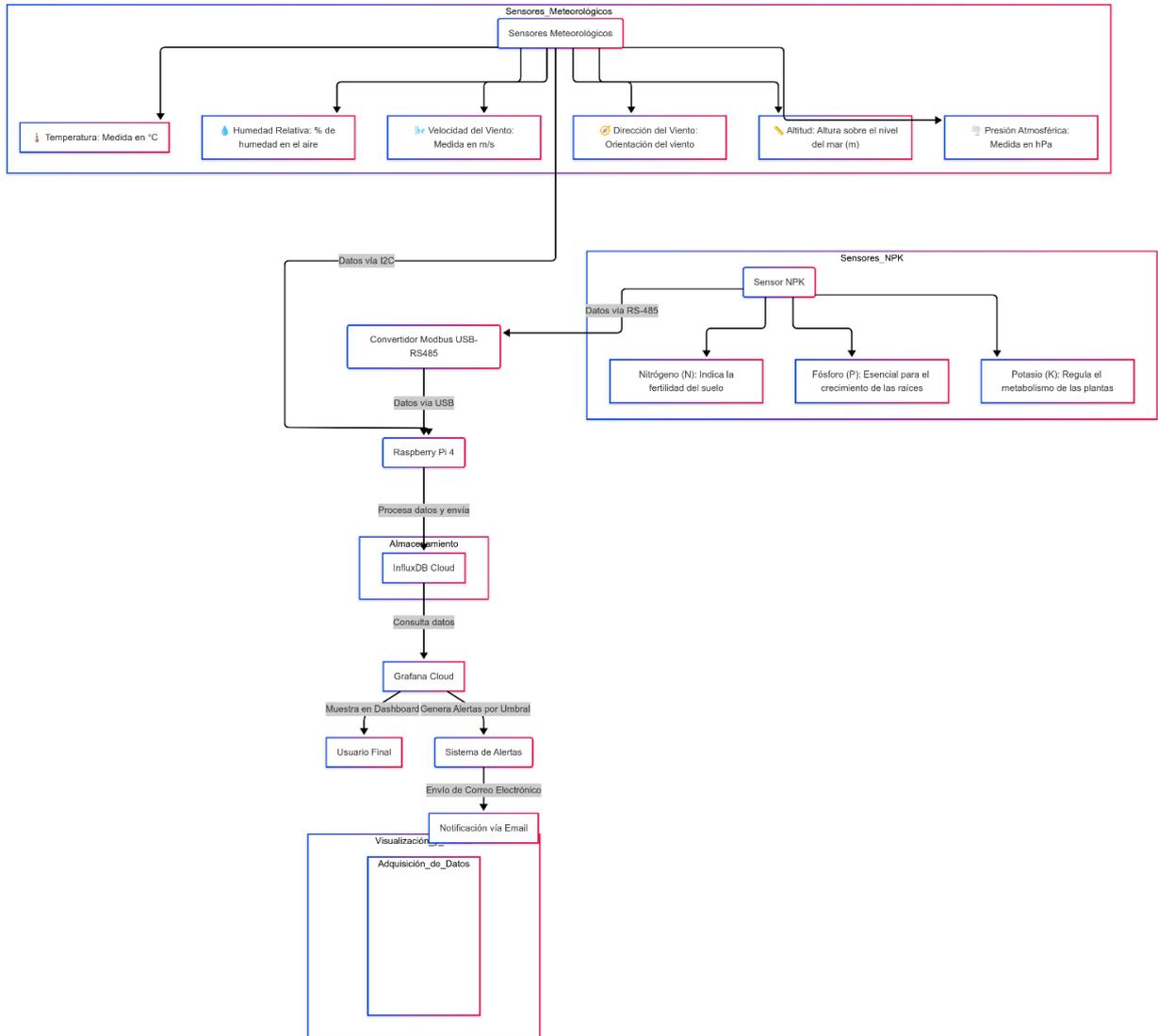
- Si la concentración de NPK supera un valor definido, se activa una alerta.
- Si la temperatura, humedad o presión atmosférica varía drásticamente, se notifica al usuario.
- Se envían correos electrónicos automáticos cuando ocurre una anomalía.

Este diagrama de flujo muestra la metodología aplicada en el diseño del sistema de monitoreo ambiental basado en Raspberry Pi 4. La combinación de sensores NPK y meteorológicos, junto con el uso de InfluxDB Cloud y Grafana Cloud, permite almacenar y visualizar los datos en tiempo real. Gracias a este enfoque, se facilita la toma de decisiones informadas en aplicaciones agrícolas y medioambientales.

Facilita la precisión de captura de datos para garantizar su análisis, optimización de almacenamiento y accesibilidad de la información de los datos obtenidos.

Figura 14

Diagrama de flujo del Sistema Metrológico lot



Nota. Diagrama de flujo demuestra como los sensores toman los datos, los almacenas y los envía a la plataforma de Grafana para su visualización

4.2 Montaje físico del sistema IoT

Proceso de montaje del sistema lot que es alimentado por un panel de 15W, un controlador de carga 12V-24VDC, una batería de 12VDC y dos reguladores de voltaje step-down a través de un Raspberry Pi 4, que permite el monitoreo continuo de variables ambientales y calidad del suelo.

El objetivo principal del sistema es proporcionar un suministro de energía constante realizando la conexión del panel solar de 15W (FIGURA 15)

Figura 15
Panel solar



Nota. Panel Solar de 15W

Con la batería de plomo-acido de 12VDC(figura 16), verificando qua batería este precargada posteriormente.

Figura 16
Batería



Nota. Batería de moto de 12V

Como último paso conectamos la batería y el panel solar al controlador de carga solar PWM (FIGURA 17), permitiendo que este controlador ajuste y administre la carga de la batería, suministrando la tensión adecuada a los distintos dispositivos del sistema, evitando sobrecargas.

Figura 17

Controlador de carga PWM



Nota. Administra y regula el flujo de energía a través de una batería y del panel solar

El controlador de carga PWM, realizara la función de administrar la energía para que pueda alimentar a los otros dispositivos, suministrando una conexión de carga de salida de 12V hacia los dos reguladores de voltaje que se encuentra conectado paralelamente.

- Raspberry Pi 4, que opera a 5VDC.
- Sensor NPK, que requiere 12VDC para su funcionamiento óptimo.

Este procedimiento es esencial para garantizar que cada componente reciba el voltaje adecuado, evitando fluctuaciones que puedan afectar el rendimiento del sistema. Ver figura 18.

Figura 18

Conexión del PWM a los reguladores de voltaje a la Raspberry Pi 4



Nota. Conexión del PWM a los reguladores de voltaje a la Raspberry Pi 4

4.2.1 Integración de Sensores con la Raspberry Pi 4

El siguiente paso en el ensamblaje consiste en conectar los reguladores de voltaje a la Raspberry Pi 4, asegurando una fuente de alimentación estable. Además, se integran los sensores encargados de recopilar datos ambientales y del suelo:

- Lark Weather Station (figura 19), que proporciona información sobre temperatura, humedad, presión atmosférica, velocidad y dirección del viento, y altitud.

Figura 19

Evaluación de prueba del prototipo



Nota. Conexión de la Raspberry pi4 a la estación meteorológica LARK

- Sensor NPK (FIGURA 20): Calcula los niveles de nutrientes del suelo como Nitrógeno, fosforo y potasio, enviando datos a través de un módulo de puente de comunicación RS232 a USB hacia la Raspberry Pi4, este módulo ajusta el voltaje y transmite los datos en serie asíncrono, recibiendo la señal sin dañarse.

Figura 20

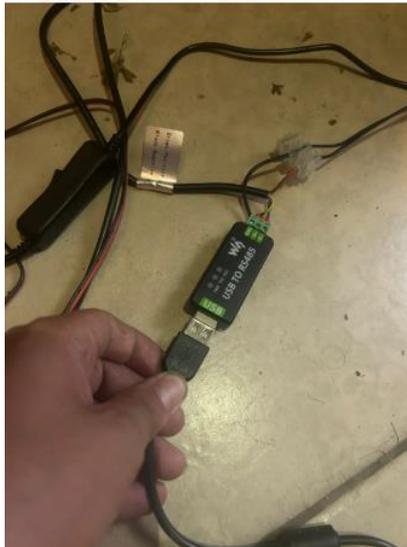
Sensor NPK



Nota. Regulador de voltaje es conectado al *Sensor NPK* para su funcionamiento
En la figura 21, visualización del módulo convertidor modbus USB a RS485.

Figura 21

Modulo convertidor modbus USB a RS485



Nota. Puerto convertidor USB a RS485 conectado a la *Raspberry pi4*

Este proceso asegura que los datos recopilados sean precisos permitiendo ser transmitidos de manera eficiente para su procesamiento y almacenamiento.

4.2.2 Comprobación del Sistema IoT en laboratorio

Para verificar el sistema se debe llevar las siguientes acciones comprobando si cada uno de ellos cumple:

- Con la cantidad recibida de cada componente exacta de energía necesaria para su funcionamiento.
- Pueda transmitir datos directo de los sensores a la Raspberry Pi 4.
- Capturen la transmisión de datos enviados a la base de datos para su análisis.

La implementación física del prototipo IoT, se centra en el desarrollo del código de programación y la configuración de herramientas como GRAFANA CLOUD, que permite la visualización y análisis de información recogida de los sensores.

El sistema del prototipo IoT con energía solar, muestra una eficiencia autónoma para la monitorización ambiental y calidad del suelo. La integración de los sensores con las Raspberry permite una mejor captura de transmisión de datos en tiempo real, mientras que se configura de manera correcta los componentes del sistema de energía solar, esto establece un mejor suministro estable. La implementación de herramientas de análisis de datos como INFLUXDB CLOUD Y GRAFANA CLOUD, permite garantizar la fiabilidad de los sistemas como un enfoque metodológico, y garantizar la fiabilidad del sistema antes de finalizar el desarrollo de software.

En la figura 22, se visualiza las pruebas del sistema de prototipo IoT en campo.

Figura 22.

Validaciones del prototipo



Nota. Equipo ensamblado y pruebas en campo

En la figura 23, se observa el sistema lot ya ensamblado con los sensores y la fuente de alimentación del sistema

Figura 23

Elementos del prototipo IoT



Nota. Elementos ensamblados y utilizados en este proyecto

4.3 Configuración de la Raspberry Pi 4

Esta sección describe la instalación y configuración de Raspberry Pi 4 para el uso de InfluxDB y Grafana para el procesamiento de los datos obtenidos por los sensores, también incluye la actualización del sistema, configuración de repositorios, ajustes en los puertos USB y la instalación inicial de la base de datos y herramientas de visualización.

4.3.1 Actualización del Sistema Operativo

Antes de instalar cualquier software, es recomendable actualizar la Raspberry Pi para garantizar compatibilidad y seguridad.

```
sudo apt update && sudo apt upgrade -y
```

- **sudo apt update:** Actualiza la lista de paquetes disponibles.
- **sudo apt upgrade -y:** Instala las versiones más recientes de los paquetes ya instalados sin requerir confirmación.

Para aplicar los cambios, se recomienda reiniciar el sistema:

```
sudo reboot
```

4.3.2 Configuración de Repositorios

Para asegurar que la Raspberry Pi tenga acceso a los paquetes necesarios, se verifica la configuración de los repositorios.

```
sudo nano /etc/apt/sources.list
```

Dentro del archivo, se deben incluir las siguientes líneas:

```
deb http://raspbian.raspberrypi.org/raspbian/ bookworm main contrib non-free rpi
deb http://archive.raspberrypi.org/debian/ bookworm main
```

Después de editar, guardar los cambios con CTRL + X, Y, y Enter.
Actualizar la lista de paquetes nuevamente:

```
sudo apt update
```

4.3.3 Configuración de Puertos USB

Para identificar los dispositivos conectados a los puertos USB:

```
lsusb
```

Muestra la lista de dispositivos USB detectados.

Para verificar los puertos serie disponibles:

```
ls /dev/ttyUSB*
```

Esto mostrará dispositivos como **/dev/ttyUSB0**, **/dev/ttyUSB1**, etc.

Si se utiliza un adaptador **RS-485**, es posible que se necesiten permisos adicionales:

```
sudo usermod -a -G dialout $USER
```

Esto agrega el usuario al grupo **dialout**, que tiene permisos sobre dispositivos seriales. Para aplicar los cambios, se recomienda reiniciar:

```
sudo reboot
```

4.3.4 Instalación de InfluxDB

4.3.4.1 Agregar el repositorio oficial de InfluxDB

Para instalar la última versión de **InfluxDB 2.x**, primero se agrega el repositorio oficial:

```
wget -qO- https://repos.influxdata.com/influxdata-archive_compat.key | sudo apt-key add -
```

Esto descarga y agrega la clave GPG del repositorio.

```
echo "deb https://repos.influxdata.com/debian stable main" | sudo tee  
/etc/apt/sources.list.d/influxdb.list
```

Añadir el repositorio de InfluxDB a la lista de fuentes de apt.

Actualizar la lista de paquetes:

```
sudo apt update
```

4.3.4.2 Instalación de InfluxDB

```
sudo apt install influxdb2 -y
```

Este comando instala la última versión estable de InfluxDB 2.x.

4.3.4.3 Habilitar e Iniciar el Servicio

```
sudo systemctl enable influxdb
```

Configurar InfluxDB para iniciarse automáticamente al encender la Raspberry Pi.

```
sudo systemctl start influxdb
```

Iniciar el servicio manualmente en la sesión actual.

Para verificar que el servicio se esté ejecutando correctamente:

```
sudo systemctl status influxdb
```

Debe mostrar el estado "**active (running)**".

4.3.4.4 Configuración Inicial de InfluxDB

Se ejecuta el asistente de configuración:

```
influx setup
```

Se solicitarán los siguientes datos:

- **Nombre de usuario:** admin
- **Contraseña:** Definir una segura
- **Nombre de la organización:** tesis_org
- **Nombre del bucket:** tesis_bucket_local
- **Token de autenticación:** Este se utilizará para conectar los scripts con **InfluxDB**.

Para consultar el token generado:

```
influx auth list
```

4.3.5 Instalación de Grafana

```
sudo apt install -y Grafana
```

Este comando instala **Grafana**, una herramienta de visualización de datos.

4.3.5.1 Habilitar e Iniciar el Servicio

```
sudo systemctl enable grafana-server  
sudo systemctl start grafana-server
```

Para verificar que el servicio se esté ejecutando correctamente:

```
sudo systemctl status grafana-server
```

Debe mostrar el estado "**active (running)**".

4.3.5.2 Acceso a la Interfaz de Grafana

Para acceder a Grafana, abrir un navegador e ingresar:

```
http://<IP_RASPBERRY>:3000
```

Las credenciales por defecto son:

- **Usuario:** admin
- **Contraseña:** admin

Se recomienda cambiar la contraseña después del primer inicio de sesión.

4.3.6 Creación y Configuración del Entorno Virtual

4.3.6.1 Creación del Entorno Virtual

```
python3 -m venv ~/myenv
```

Crea un entorno virtual para aislar las dependencias del proyecto.

4.3.6.2 Activación del Entorno Virtual

```
source ~/myenv/bin/actíivate
```

Activa el entorno virtual.

4.3.6.3 Instalación de Dependencias

```
pip install influxdb-client 75ython-dotenv pymodbus
```

Si hay restricciones en la instalación, se puede usar:

```
pip install --break-system-packages influxdb-client python-dotenv pymodbus
```

4.3.6.4 Verificación de la Instalación

```
pip show influxdb-client
```

```
pip show pymodbus
```

Hay que confirmar que las bibliotecas están correctamente instaladas en el entorno virtual.

4.3.7 Instalación del Sensor DFRobot_LarkWeatherStation

4.3.7.1 Instalación del driver

Desde el directorio donde se encuentra el código fuente:

```
pip install . --break-system-packages
```

4.3.7.2 Verificación de la instalación

```
pip show DFRobot-LarkWeatherStation
```

4.3.7.3 Prueba de importación

```
python -c "from DFRobot_LarkWeatherStation import DFRobot_LarkWeatherStation_I2C;
print('Módulo importado correctamente')"
```

4.3.7.4 Ejecución del Script Final

Para ejecutar el script de monitoreo:

```
python proyecto_final_local.py
```

Para ejecutarlo en segundo plano:

```
nohup python proyecto_final_local.py &
```

Para monitorear su ejecución:

```
tail -f nohup.out
```

Para detener la ejecución en segundo plano:

```
pkill -f proyecto_final_local.py
```

4.4 Configuración del sensor NPK

4.4.1 Configuración del Sensor NPK en Raspberry Pi

El sensor NPK mide los niveles de Nitrógeno (N), Fósforo (P) y Potasio (K) en el suelo y se comunica a través del protocolo Modbus RTU (RS-485). Para configurarlo en la Raspberry Pi, sigue los siguientes pasos:

4.4.2 Conexión del Sensor NPK

El sensor se conecta mediante un adaptador USB-RS485 en la Raspberry Pi. Verificar que el dispositivo está conectado con:

```
ls /dev/ttyUSB*
```

Si el sensor está correctamente detectado, mostrará algo como:

```
/dev/ttyUSB0
```

Si hay más de un dispositivo, puede aparecer */dev/ttyUSB1*, */dev/ttyUSB2*, etc.

4.4.3 Instalación de Dependencias

Si no se tiene pymodbus instalado, se debe ejecutar dentro del entorno virtual:

```
pip install pymodbus
```

Si hay restricciones en la instalación, usar:

```
pip install pymodbus --break-system-packages
```

4.4.4 Configuración del Sensor NPK en Python

El siguiente código configura el sensor y lee los valores de Nitrógeno (N), Fósforo (P) y Potasio (K) usando Modbus RTU:

```
from pymodbus.client import ModbusSerialClient as ModbusClient

# Configuración del puerto serie
NPK_PORT = "/dev/ttyUSB0" # Ajustar según el puerto detectado
client_npk = ModbusClient(port=NPK_PORT, baudrate=9600, parity='N', stopbits=1,
bytesize=8, timeout=1)

def setup_npk():
    """Intenta conectar con el sensor NPK."""
    if client_npk.connect():
        print("Conexión exitosa con el sensor NPK.")
    else:
```

```

    print("No se pudo conectar al sensor NPK.")
    exit(1)

def leer_npk():
    """Lee los valores del sensor NPK."""
    try:
        result = client_npk.read_holding_registers(30, count=3, slave=1) # Dirección Modbus 1
        if result.isError():
            print("Error en la lectura del sensor NPK.")
            return None
    except Exception as e:
        print(f"Error al leer el sensor NPK: {e}")
        return None

    data = {
        "Nitrogeno": result.registers[0], # mg/kg
        "Fosforo": result.registers[1], # mg/kg
        "Potasio": result.registers[2] # mg/kg
    }

    return data

# Inicializar el sensor y leer datos
setup_npk()
datos_npk = leer_npk()
if datos_npk:
    print("\n--- Datos del Sensor NPK ---")
    print(f"Nitrógeno: {datos_npk['Nitrogeno']} mg/kg")
    print(f"Fósforo: {datos_npk['Fosforo']} mg/kg")
    print(f"Potasio: {datos_npk['Potasio']} mg/kg")

# Cerrar la conexión con el sensor
client_npk.close()

```

4.4.5 Explicación del Código

- **ModbusClient:** Se usa para establecer la conexión con el sensor NPK a través de Modbus RTU.
- **setup_npk():** Verifica si el sensor responde y se conecta correctamente.
- **Leer_npk():** Envía una solicitud Modbus para leer los registros donde están almacenados los datos de NPK.
- Los valores devueltos están en mg/kg, lo que representa la concentración de cada nutriente en el suelo.
- Se imprime la información en pantalla y se cierra la conexión al finalizar.

4.4.6 Verificación de la Comunicación

Si el sensor no responde, se debe cambiar la dirección Modbus (por defecto 1) con:

```
result = client_npk.read_holding_registers(30, count=3, slave=2) # Prueba con otra dirección
```

También se puede probar un escaneo de direcciones Modbus con:

```
for addr in range(1, 247):
    result = client_npk.read_holding_registers(30, count=3, slave=addr)
    if not result.isError():
        print(f"Sensor NPK detectado en la dirección {addr}")
        break
```

4.5 Programación de Raspberry Pi – Almacenamiento Local

El siguiente código de programación en Python permite la lectura y almacenamiento de datos provenientes de sensores meteorológicos y de calidad del suelo (NPK). Utiliza InfluxDB para registrar las mediciones y permite la conexión con sensores a través de I2C y Modbus RS-485.

4.5.1 Importación de Bibliotecas

El código inicia importando las bibliotecas necesarias:

- time, os, sys: Para operaciones del sistema y control del tiempo.
- influxdb_client: Manejo de la base de datos InfluxDB.
- DFRobot_LarkWeatherStation: Para la lectura del sensor meteorológico.
- pymodbus.client: Para la comunicación con el sensor NPK a través de Modbus.

4.5.2 Configuración de la Base de Datos InfluxDB

Se establecen los parámetros de conexión a InfluxDB:

```
INFLUXDB_URL = "http://localhost:8086"
```

```
INFLUXDB_TOKEN = "TOKEN"
```

```
INFLUXDB_ORG = "tesis_org"
```

```
INFLUXDB_BUCKET = "tesis_bucket_local"
```

Se crea un cliente para interactuar con la base de datos:

```
client_influx = InfluxDBClient(url=INFLUXDB_URL, token=INFLUXDB_TOKEN,  
org=INFLUXDB_ORG)
```

```
write_api = client_influx.write_api()
```

4.5.3 Inicialización de Sensores

4.5.3.1 Sensor Meteorológico (I2C)

Se inicializa el sensor utilizando la dirección I2C:

```
ADDRESS = 0x42
```

```
EDU0157 = DFRobot_LarkWeatherStation_I2C(ADDRESS)
```

Función para inicializarlo correctamente:

```
def setup_meteorologico():  
    while EDU0157.begin() != 0:  
        print("Error al inicializar el sensor meteorológico.")  
        time.sleep(1)  
    print("Sensor meteorológico inicializado correctamente.")
```

4.5.3.2 Sensor NPK (Modbus RS-485)

Se establece la conexión con el sensor NPK:

```
NPK_PORT = '/dev/ttyUSB0'
```

```
client_npk = ModbusClient(port=NPK_PORT, baudrate=9600, parity='N', stopbits=1,  
bytesize=8, timeout=1)
```

Función de configuración:

```
def setup_npk():  
    if client_npk.connect():  
        print("Conexión exitosa con el sensor NPK.")  
    else:  
        print("No se pudo conectar al sensor NPK.")  
        sys.exit(1)
```

4.5.4 Lectura de Sensores

4.5.4.1 Sensor Meteorológico

Se obtienen valores como velocidad y dirección del viento, temperatura, humedad, presión y altitud:

```
def leer_meteorologico():  
    try:  
        data = {  
            "speed": (safe_float(EDU0157.get_value("Speed")),  
EDU0157.get_unit("Speed")),  
            "direction": (EDU0157.get_value("Dir"), ""),  
            "temperature": (safe_float(EDU0157.get_value("Temp")),  
EDU0157.get_unit("Temp")),  
            "humidity": (safe_float(EDU0157.get_value("Humi")),  
EDU0157.get_unit("Humi")),  
            "pressure": (safe_float(EDU0157.get_value("Pressure")),  
EDU0157.get_unit("Pressure")),  
            "altitude": (safe_float(EDU0157.get_value("Altitude")),  
EDU0157.get_unit("Altitude")),
```

```

    }
except Exception as e:
    print(f"Error al leer el sensor meteorológico: {e}")
    return None
return data

```

4.5.4.2 Sensor NPK

Se obtienen las concentraciones de **Nitrógeno (N)**, **Fósforo (P)** y **Potasio (K)**:

```

def leer_npk():

    try:
        result = client_npk.read_holding_registers(30, count=3, slave=1)
        if result.isError():
            print("Error en la lectura del sensor NPK.")
            return None
    except Exception as e:
        print(f"Error al leer el sensor NPK: {e}")
        return None

    data = {
        "Nitrogeno": safe_float(result.registers[0]),
        "Fosforo": safe_float(result.registers[1]),
        "Potasio": safe_float(result.registers[2]),
    }
    return data

```

4.5.5 Enviar Datos a InfluxDB

Se formatean y envían los datos a la base de datos:

```

def enviar_datos_influxdb(datos_meteorologicos, datos_npk):
    if datos_meteorologicos:
        point_weather = (

```

```

        Point("weather_measurement")
        .field("speed", datos_meteorologicos["speed"][0])
        .field("temperature", datos_meteorologicos["temperature"][0])
        .field("humidity", datos_meteorologicos["humidity"][0])
        .field("pressure", datos_meteorologicos["pressure"][0])
    )
    write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,
record=point_weather)
    if datos_npk:
        point_npk = (
            Point("soil_npk_measurement")
            .field("nitrogen", datos_npk["Nitrogeno"])
            .field("phosphorus", datos_npk["Fosforo"])
            .field("potassium", datos_npk["Potasio"])
        )
        write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,
record=point_npk)

```

4.5.6 Ejecución del Programa

El programa principal se ejecuta en un bucle infinito, leyendo datos cada 20 segundos y enviándolos a InfluxDB cada 5 minutos:

```

def main():
    setup_meteorologico()
    setup_npk()
    next_send_time = time.time() + 300

    while True:
        datos_meteorologicos = leer_meteorologico()
        datos_npk = leer_npk()

        if time.time() >= next_send_time:
            enviar_datos_influxdb(datos_meteorologicos, datos_npk)
            next_send_time = time.time() + 300

```

```
time.sleep(20)
```

Este sistema permite monitorear variables meteorológicas y del suelo en tiempo real, almacenando los datos en una base de datos InfluxDB para su posterior análisis.

4.6 Configuración de código para enviar los datos a InfluxDB Cloud

Esta sección explica detalladamente el funcionamiento del código diseñado para el monitoreo de sensores meteorológicos y de calidad del suelo (NPK), almacenando los datos en InfluxDB Cloud. La implementación se realiza en una Raspberry Pi 4, utilizando InfluxDB como base de datos para el almacenamiento de datos en series temporales y Grafana para su visualización.

4.6.1 Importación de Bibliotecas

El código comienza importando las bibliotecas necesarias:

```
import time
import os
import sys
from influxdb_client import InfluxDBClient, Point, WritePrecision
from dotenv import load_dotenv
from DFRobot_LarkWeatherStation import DFRobot_LarkWeatherStation_I2C
from pymodbus.client import ModbusSerialClient as ModbusClient
```

- **time**: Permite la gestión de intervalos de tiempo en la ejecución del código.
- **os**: Se usa para acceder a variables de entorno.
- **sys**: Permite la manipulación del entorno de ejecución del script.
- **influxdb_client**: Cliente para conectar con la base de datos InfluxDB en la nube.
- **dotenv**: Se usa para cargar variables de entorno desde un archivo .env.
- **DFRobot_LarkWeatherStation**: Biblioteca para interactuar con el sensor meteorológico.
- **pymodbus.client**: Biblioteca que permite la comunicación con el sensor NPK a través del protocolo Modbus RTU.

4.6.2 Configuración de InfluxDB Cloud

El código configura la conexión con **InfluxDB Cloud** a través de variables de entorno:

```
load_dotenv()
INFLUXDB_URL = os.getenv("INFLUXDB_URL", "https://us-east-1-1.aws.cloud2.influxdata.com")
INFLUXDB_TOKEN = os.getenv("INFLUXDB_TOKEN", "TOKEN")
INFLUXDB_ORG = os.getenv("INFLUXDB_ORG", "UPS")
INFLUXDB_BUCKET = os.getenv("INFLUXDB_BUCKET", "tesis_bucket")
Posteriormente, se inicializa el cliente de InfluxDB:
client_influx = InfluxDBClient(url=INFLUXDB_URL, token=INFLUXDB_TOKEN,
org=INFLUXDB_ORG)
write_api = client_influx.write_api()
```

4.6.3 Configuración de Sensores

4.6.3.1 Sensor Meteorológico (I2C)

```
ADDRESS = 0x42
EDU0157 = DFRobot_LarkWeatherStation_I2C(ADDRESS)
El sensor meteorológico utiliza el protocolo I2C con dirección 0x42.
```

4.6.3.2 Sensor NPK (Modbus RS-485)

```
NPK_PORT = '/dev/ttyUSB0'
client_npk = ModbusClient(port=NPK_PORT, baudrate=9600, parity='N', stopbits=1,
bytesize=8, timeout=1)
```

Este sensor utiliza **Modbus RTU** y está conectado mediante un adaptador USB-RS485.

4.6.4 Lectura de Sensores

4.6.4.1 Sensor Meteorológico

```
def leer_meteorologico():
```

```

try:
    data = {
        "speed": (safe_float(EDU0157.get_value("Speed")), EDU0157.get_unit("Speed")),
        "direction": (EDU0157.get_value("Dir"), ""),
        "temperature":(safe_float(EDU0157.get_value("Temp")), EDU0157.get_unit("Temp")),
        "humidity": (safe_float(EDU0157.get_value("Humi")), EDU0157.get_unit("Humi")),
        "pressure":(safe_float(EDU0157.get_value("Pressure")),
EDU0157.get_unit("Pressure")),
        "altitude": (safe_float(EDU0157.get_value("Altitude")), EDU0157.get_unit("Altitude")),
    }
except Exception as e:
    print(f"Error al leer el sensor meteorológico: {e}")
    return None
return data

```

Esta función obtiene valores de velocidad del viento, temperatura, humedad, presión y altitud.

4.6.4.2 Sensor NPK

```

def leer_npk():
    try:
        result = client_npk.read_holding_registers(30, count=3, slave=1)
        if result.isError():
            print("Error en la lectura del sensor NPK.")
            return None
    except Exception as e:
        print(f"Error al leer el sensor NPK: {e}")
        return None

    data = {
        "Nitrogeno": safe_float(result.registers[0]),
        "Fosforo": safe_float(result.registers[1]),
        "Potasio": safe_float(result.registers[2]),
    }

```

```
return data
```

Esta función obtiene las concentraciones de **Nitrógeno (N)**, **Fósforo (P)** y **Potasio (K)** en el suelo.

4.6.5 Almacenamiento en InfluxDB Cloud

```
def enviar_datos_influxdb(datos_meteorologicos, datos_npk):  
    if datos_meteorologicos:  
        point_weather = (  
            Point("weather_measurement_v2")  
            .field("speed", datos_meteorologicos["speed"][0])  
            .field("temperature", datos_meteorologicos["temperature"][0])  
            .field("humidity", datos_meteorologicos["humidity"][0])  
            .field("pressure", datos_meteorologicos["pressure"][0])  
        )  
        write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,  
record=point_weather)  
  
    if datos_npk:  
        point_npk = (  
            Point("soil_npk_measurement_v2")  
            .field("nitrogen", datos_npk["Nitrogeno"])  
            .field("phosphorus", datos_npk["Fosforo"])  
            .field("potassium", datos_npk["Potasio"])  
        )  
        write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG, record=point_npk)
```

Esta función envía los datos de los sensores a **InfluxDB Cloud**.

Este sistema permite monitorear variables meteorológicas y de calidad del suelo en tiempo real, almacenando los datos en InfluxDB Cloud para su posterior análisis y visualización en Grafana. Si se requiere una mayor personalización en la configuración o integración con otros sistemas, se pueden realizar ajustes en la conexión con la base de datos o en la frecuencia

de muestreo de los sensores.

4.6.6 Configuración de InfluxDB Cloud

En esta sección se explica la configuración de la cuenta en InfluxDB Cloud y la creación del bucket.

1) Acceder a la página web de InfluxDB Cloud:

- Visitar [InfluxDB Cloud](#).
- Click en "Get InfluxDB" para registrarse.
- Iniciar sesión con cuenta de mail.

2) Seleccionar la Región del Servidor

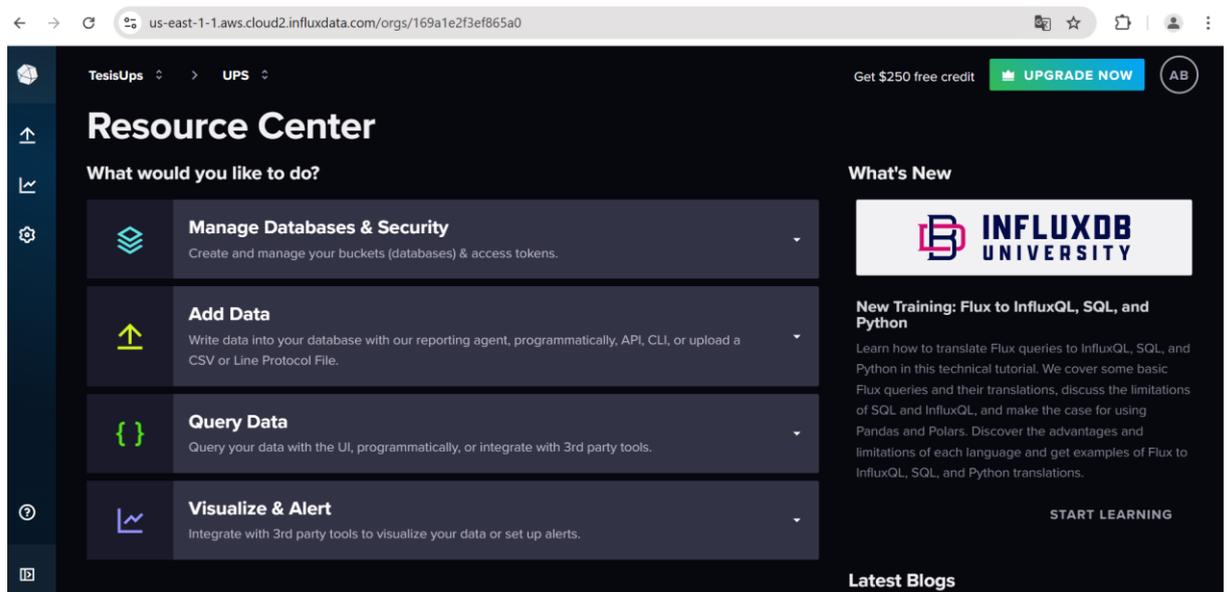
- Para este proyecto, la URL usada es:

<https://us-east-1-1.aws.cloud2.influxdata.com>

- Seleccionar una región que tenga menor latencia para la ubicación.}

Figura 24

InfluxDB Cloud



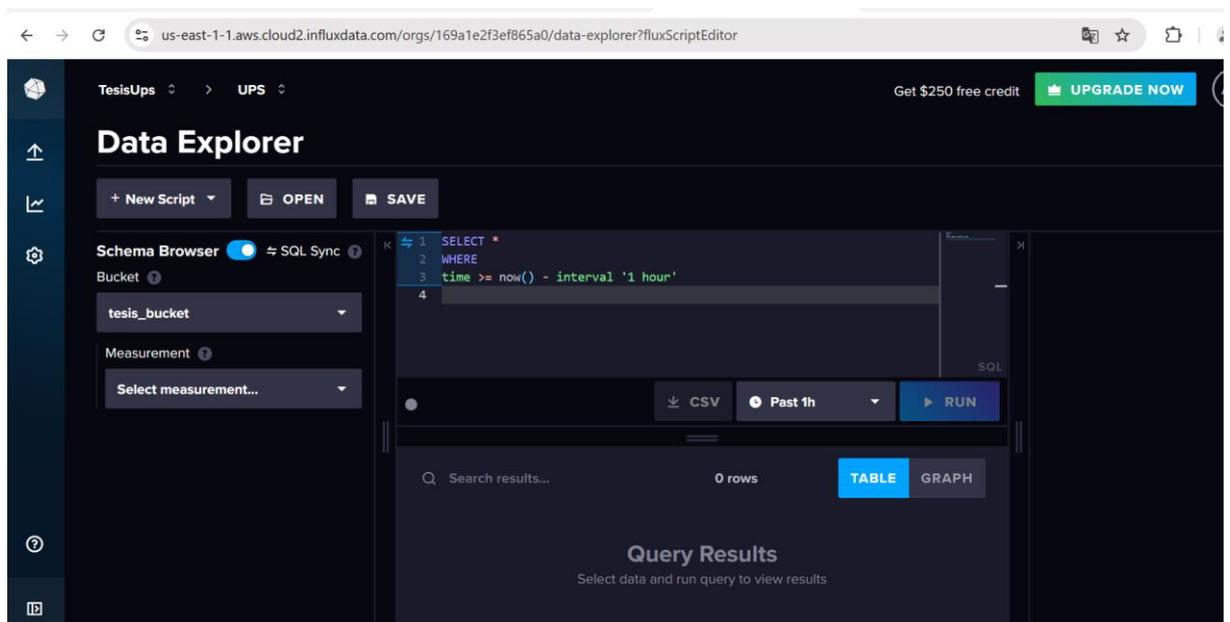
Nota. Pantalla principal de InfluxDB

4.6.6.1 Configurar una Organización y Bucket

- Al crear la cuenta, pedirá un nombre para la organización.
- En este caso, se usó: *UPS*
- Luego, crear un **Bucket** (base de datos en InfluxDB Cloud).
- En este proyecto, el **Bucket** configurado es: *tesis_bucket*

Figura 25

Data Explorer InfluxDB

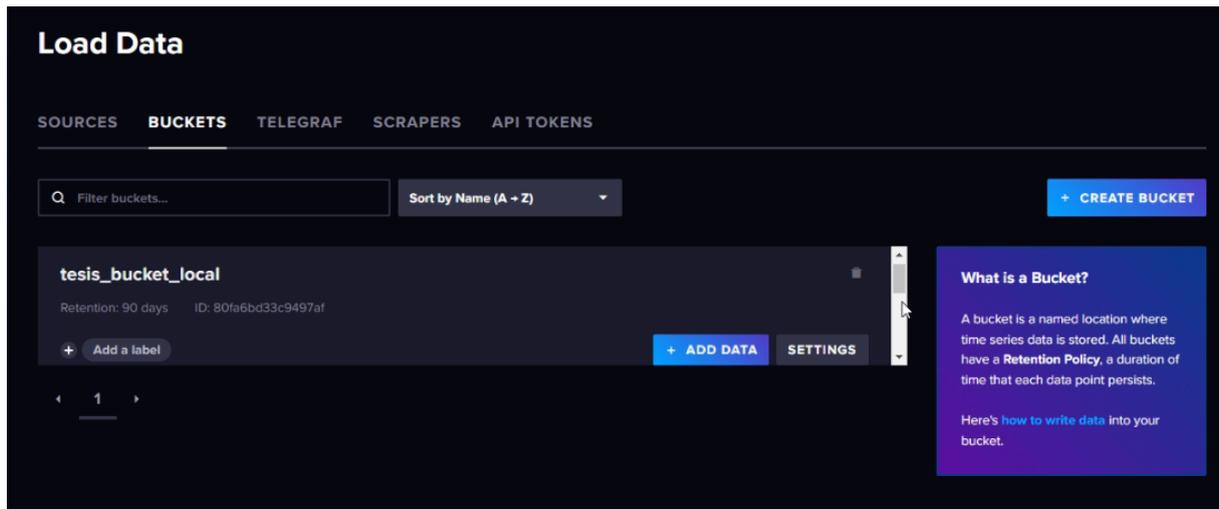


Nota. Creación del Bucket para el almacenamiento de datos

En la figura 26 se visualiza la pantalla principal, donde se encuentran los buckets que almacenan los datos de los sensores NPK y Lark Weather Station

Figura 26

Configuración del Bucket



Nota. Configuración de Bucket para el almacenamiento de datos

4.6.6.2 Creación del API Token para la Conexión

Para que la Raspberry Pi pueda enviar datos a InfluxDB Cloud, se necesita un API Token.

- 1) Acceder a la Configuración de Tokens
 - En el panel de InfluxDB, ir a "Load Data" → "API Tokens".
 - Hacer clic en "Generate API Token".
 - Seleccionar "All Access Token" para tener permisos de lectura y escritura.

- 2) Guardar el API Token
 - Copiar el Token generado.
 - En este proyecto, el token utilizado es:

`0SGPuio3LhrpgdXDo0nKEX2tnB7I23zaxrsjHZsy6USoZLNo_nMsGB8JX17VM5RLP7qZVek
OhCTnzUqx5UB40==`

- 3) Configurar el Token en el Código
 - En el código Python, se almacena en una variable de entorno:

```
INFLUXDB_TOKEN = os.getenv("INFLUXDB_TOKEN",
```

```
"0SGPuio3LhrpgdXDo0nKEX2tnB7I23zaxrsjHZsy6USoZLNo_nMsGB8JX17VM5RLP7qZVek  
OhCTnzUqx5UB40=="
```

4.6.6.3 Creación de un Dashboard en InfluxDB Cloud

Para visualizar los datos, se puede crear un dashboard en InfluxDB Cloud.

Ir a la pestaña "Dashboards"

- Dentro de InfluxDB, accede a "Dashboards" y haz clic en "Create Dashboard".

Añadir una Consulta de Datos

- Seleccionar "Add Cell" y usa la consulta para visualizar los datos de los sensores:

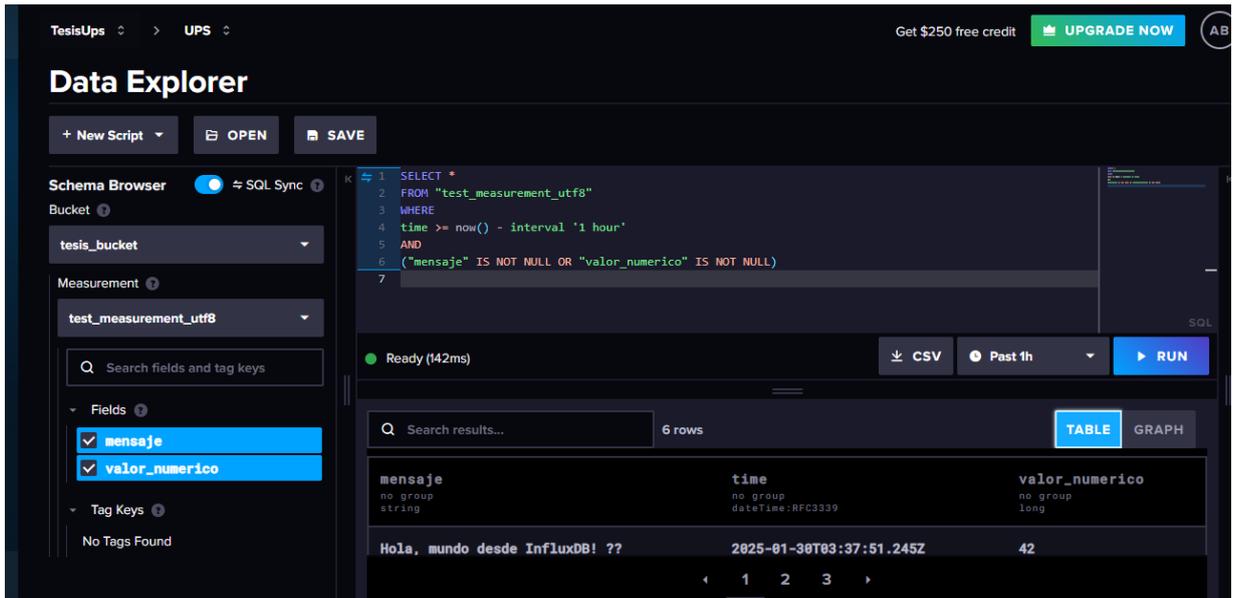
```
from(bucket: "tesis_bucket")  
  |> range(start: -1h)  
  |> filter(fn: (r) => r._measurement == "weather_measurement_v2")
```

- Esto mostrará datos meteorológicos de la última hora

En las figuras del 27 se observa la configuración de InfluxDB Cloud que se uso para la el desarrollo de este sistema IoT.

Figura 27

Configuración datos en InfluxDB, script InfluxDB

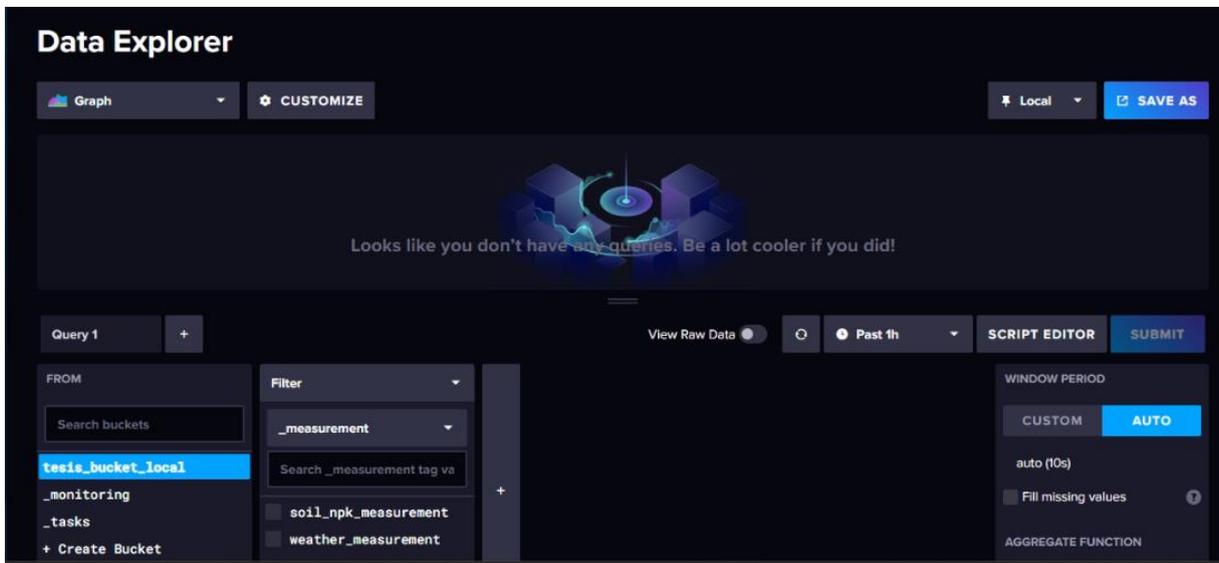


Nota. Configuración de datos almacenados en InfluxDB, script InfluxDB

En la figura 28 la creación de los buckets NPK y weather donde se puede visualizar los datos mediante consultas InfluxDB.

Figura 28

Configuración datos en InfluxDB



Nota. Configuración de los datos almacenados en InfluxDB

5. CRONOGRAMA

En la siguiente tabla 7 se muestra las actividades realizadas en este proyecto

Tabla 7

Cronograma de Actividades

Actividad	Semanas											
	1	2	3	4	5	6	7	8	9	10	11	12
Comprar materiales												
Diseño del sistema portátil												
Descarga y actualización de librerías para el funcionamiento de la raspberry pi												
Realizar programación para el monitoreo de los datos ambientales y nutrientes												
Realizar ensamblado de sensores, LARK, NPK y raspberry pi4												
Análisis de datos recopilados y almacenados en Grafana cloud												
Realizar pruebas preliminares en campo												

Nota. Cronograma de actividades

6 PRESUPUESTO

En la tabla 8 se presenta los materiales a usar y su costo

Tabla 8

Presupuesto del proyecto

PRESUPUESTO DEL PROYECTO			
DESCRIPCIÓN	CANT.	VALOR POR UNID.	VALOR TOTAL
Raspberry Pi 4 (case, Fuente, microSD 32GB)	1	\$80.00	\$80.00
Lark Weather Station	1	\$150.00	\$150.00
Sensor NPK	1	\$30.00	\$30.00
Módulo TTL a RS-485	1	\$5.00	\$5.00
Controlador de carga solar PWM	1	\$11.00	\$11.00
Módulo Step Down	2	\$5.00	\$10.00
Batería de moto	1	\$35.00	\$35.00
Panel Solar y soporte	1	\$30.00	\$30.00
TOTAL			\$ 352

Nota. Presupuesto para el desarrollo de este proyecto

7 RESULTADOS

Este capítulo presenta los resultados obtenidos a partir de la implementación del sistema IoT para la monitorización de variables ambientales y calidad del suelo. Se analizarán los datos recolectados por los sensores meteorológicos y NPK, la eficiencia en el almacenamiento de datos en InfluxDB Cloud, y la visualización en Grafana Cloud. Además, se evaluará el impacto de los datos obtenidos en la toma de decisiones para el monitoreo agrícola.

7.1 Resultados de la Configuración del Sistema IoT

El sistema IoT se configuró para garantizar la correcta adquisición, almacenamiento y

visualización de datos en tiempo real. A continuación, se detallan los resultados obtenidos tras la implementación de cada componente del sistema:

7.1.1 Funcionamiento del Sistema Energético

- **Suministro autónomo de energía:** Se comprobó que el panel solar de 15W, junto con el controlador de carga PWM de 12V-24VDC y la batería de 12VDC, proporcionaron energía estable a los sensores y a la Raspberry Pi 4.
- **Optimización del controlador PWM:** La carga de la batería se completó en un tiempo de 4 a 6 horas de exposición de sol, permitiendo que la carga completa sea útil para el sistema durante la noche.

7.1.2 Integración de Sensores

- **Sensores meteorológicos (Lark Weather Station):** Se verificó la transmisión de datos en tiempo real, logrando registrar mediciones de temperatura, humedad relativa, presión atmosférica, velocidad y dirección del viento, y altitud con una tasa de actualización de cada 10 segundos.
- **Sensor NPK (RS-485):** Se logró establecer comunicación con el sensor a través del convertidor USB a RS485, obteniendo mediciones de nitrógeno, fósforo y potasio (NPK) en el suelo, con una frecuencia de muestreo de cada 5 minutos.

7.1.3 Transmisión y Almacenamiento de Datos

- InfluxDB Cloud permitió el almacenamiento estructurado de los datos en un bucket denominado "tesis_bucket".
- Se logró enviar datos desde la Raspberry Pi 4 utilizando el protocolo HTTP, con una latencia promedio de 40-70 ms.
- Se verificó que los datos se almacenaban correctamente en la base de datos y podían

ser recuperados con consultas en Flux, como:

```
from(bucket: "tesis_bucket")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "weather_measurement_v2")
```

7.1.4 Análisis de Datos en Grafana Cloud

La integración de Grafana Cloud con InfluxDB Cloud permitió la representación gráfica de las mediciones, facilitando la interpretación de los datos en tiempo real. Se obtuvieron los siguientes resultados:

7.1.5 Variabilidad de Parámetros Meteorológicos

Se observaron tendencias estacionales en las mediciones meteorológicas:

- **Temperatura:** Osciló entre 22°C y 35°C, con picos máximos al mediodía y mínimos en la madrugada.
- **Humedad relativa:** Registró valores entre 55% y 90%, mostrando una correlación inversa con la temperatura.
- **Velocidad del viento:** En días con mayor temperatura, la velocidad del viento aumentó, alcanzando valores de hasta 8 m/s.

7.1.6 Análisis de la Calidad del Suelo

- **Niveles de nitrógeno:** Variaron entre 25 mg/kg y 60 mg/kg, siendo más altos en suelos con mayor humedad.
- **Niveles de fósforo y potasio:** Se registraron valores promedio de 20 mg/kg y 40 mg/kg, respectivamente.
- **Efecto de la humedad en la fertilidad:** Se identificó que en días de alta humedad (>80%), la disponibilidad de fósforo disminuyó, sugiriendo un efecto de lixiviación en el suelo.

7.1.7 Configuración de Alertas

Se establecieron alertas en Grafana Cloud para notificar variaciones críticas en los parámetros monitoreados:

- **Alertas de humedad:** Se configuró una alerta para valores inferiores al **50%**, indicando riesgo de sequía.
- **Alertas de NPK:** Se estableció un umbral de **30 mg/kg** para nitrógeno, alertando cuando los valores caían por debajo de este límite.
- **Notificaciones por correo:** Cuando los valores críticos eran alcanzados, el sistema enviaba notificaciones a través del servicio de alertas de **Grafana Cloud**.

7.1.8 Impacto del Sistema en la Toma de Decisiones

El análisis de datos en tiempo real permitió a los usuarios tomar decisiones informadas para mejorar la fertilización del suelo y optimizar las condiciones ambientales:

- **Optimización del riego:** Se ajustaron los tiempos de riego en función de la humedad del suelo, reduciendo el desperdicio de agua.
- **Uso eficiente de fertilizantes:** Se aplicaron fertilizantes de nitrógeno y fósforo solo cuando los valores caían por debajo de los umbrales recomendados.
- **Predicción de condiciones climáticas:** Se identificaron patrones en la velocidad del viento y la humedad, permitiendo prever cambios climáticos adversos.

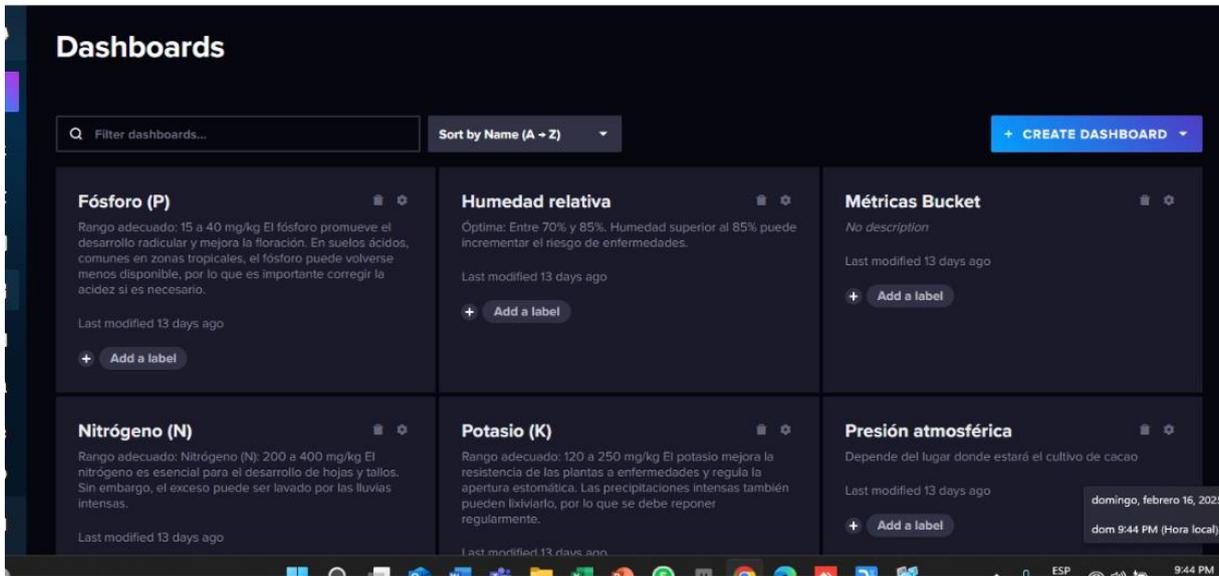
7.2 Evidencias de los resultados de obtenidos del prototipo IoT

En las siguientes figuras se observan las evidencias de las configuraciones realizadas y los resultados obtenidos para la visualización de la información de cada uno de los parámetros de los sensores.

En la figura 29 se observa el dashboard configurado en InfluxDB local (Raspberry Pi 4), las métricas son obtenidas del bucket tesis_local

Figura 29

InfluxDB Local dashboard de sensores

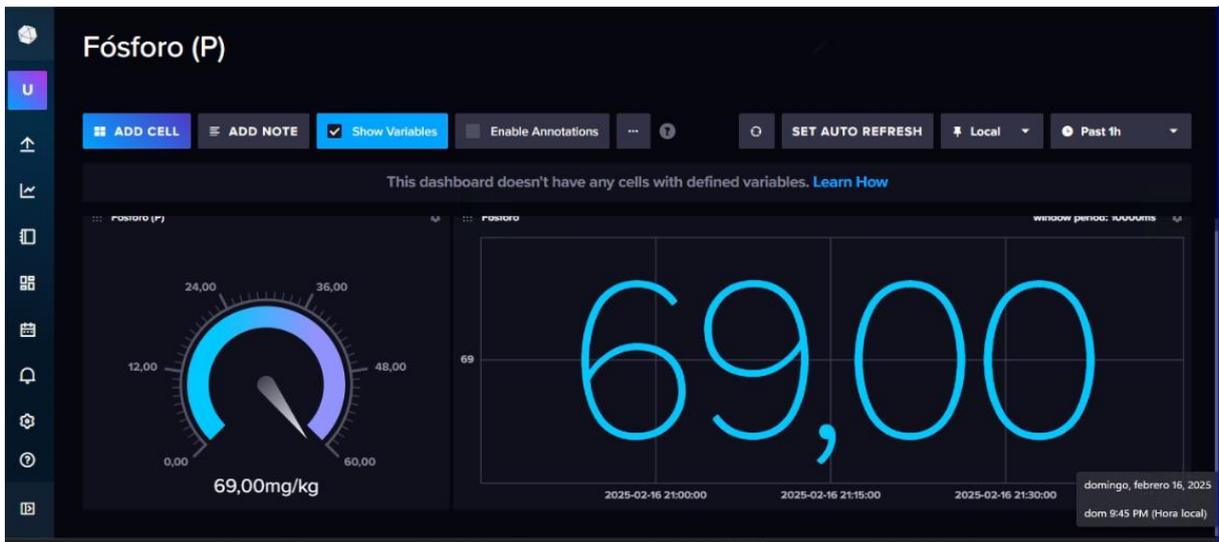


Nota. InfluxDB local dashboards con información de los sensores NPK y LARK

En la figura 30 se observa el detalle la dashboard con la información obtenida del fósforo.

Figura 30

Datos de cantidad de fósforo del suelo



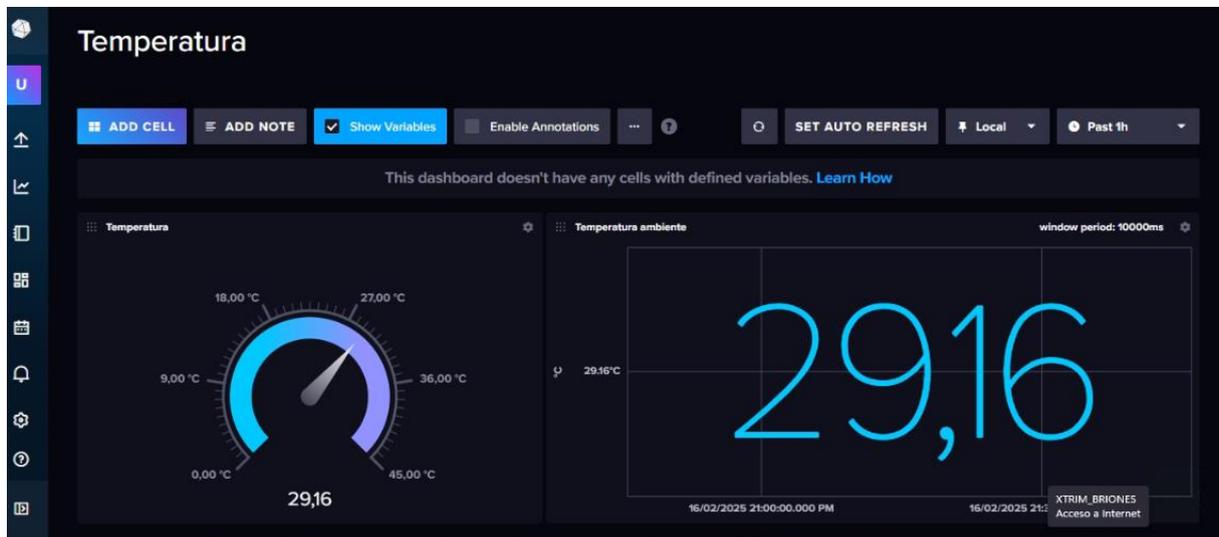
Nota. Dashboard de fosforo, indica la cantidad de fosforo obtenida en las pruebas de campo

Y en la figura 31 observamos la dashboard de la temperatura que son transmitidos en tiempo

real por el sensor del Lark Weather Station

Figura 31

Datos de temperatura ambiente



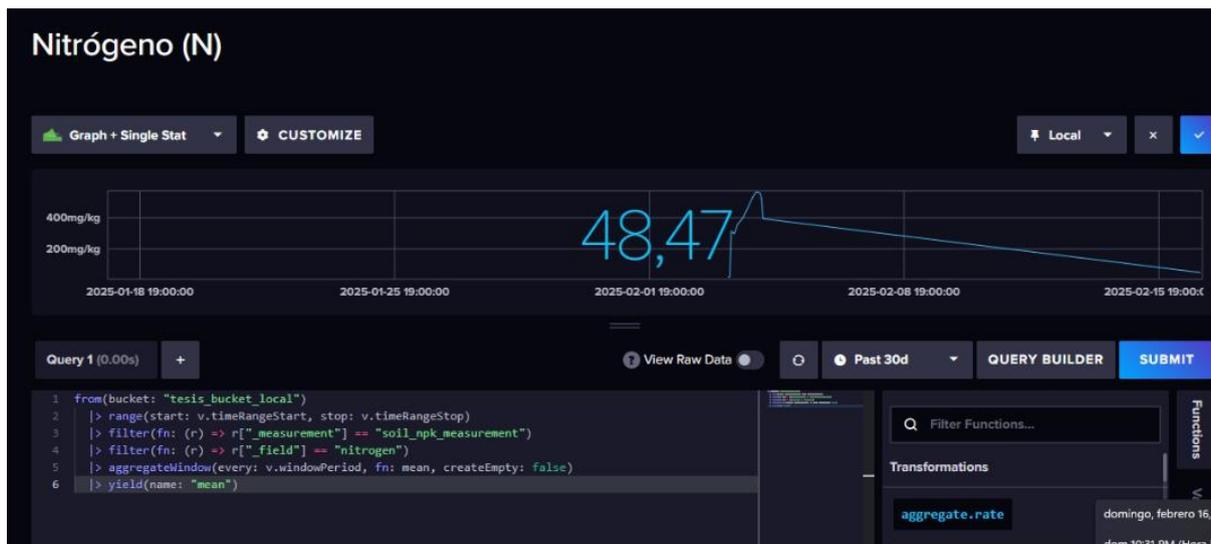
Nota. Dashboard de temperatura, datos obtenidos en la prueba recolectada por el LARK. En la figura 32 se observa la configuración realizada con el script flux para obtener la data del bucket y graficarla.

El código flux es el siguiente:

```
from(bucket: "tesis_bucket_local")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "soil_npk_measurement")
  |> filter(fn: (r) => r["_field"] == "phosphorus")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

Figura 32

InfluxDB script flux

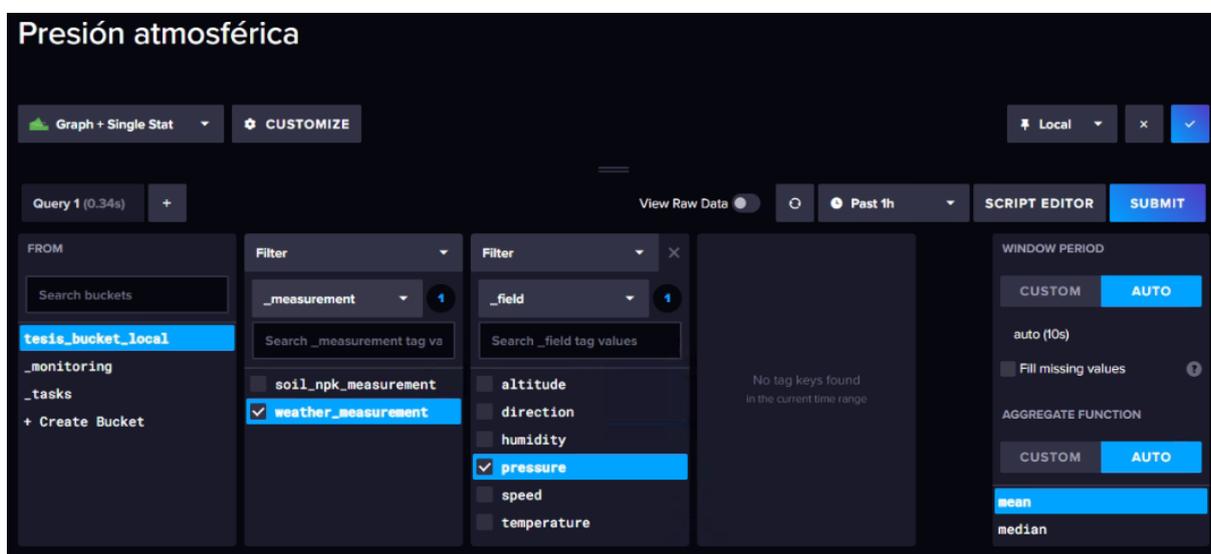


Nota. Configuración en la dashboard para la obtención de datos del sensor NPK

En la figura 33 se observa cómo se arma el script para obtener los datos obtenidos por los sensores en el NPK y en el Lark Weather Station

Figura 33

Script para visualizar la presión atmosférica



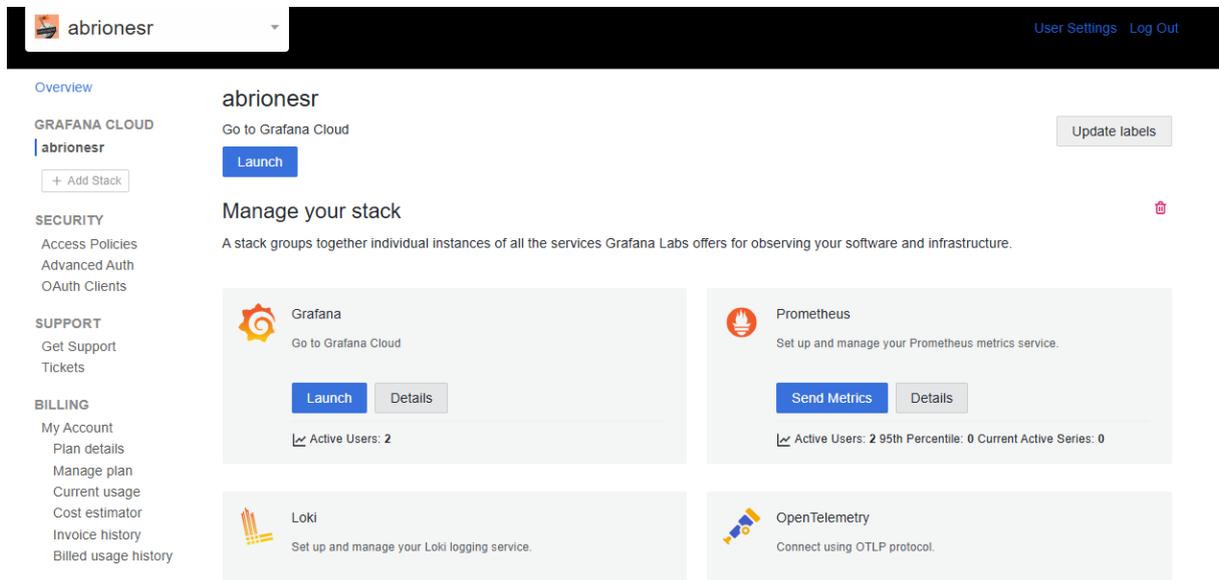
Nota. InfluxDB script para configurar dashboard para la obtención de la presión atmosférica

A continuación, en la figura 34, se visualiza el dashboard de la plataforma Grafana Cloud

donde se almacena la información obtenida por los sensores en tiempo real.

Figura 34

Pantalla principal de Grafana Cloud



Nota. Pantalla principal de Grafana Cloud

En la plataforma Grafana Cloud se puede administrar que usuarios pueden visualizar la información que se presenta en el sistema.

Figura 35

Usuarios de Grafana Cloud

grafana.com/orgs/abrionesr/members

Overview

GRAFANA CLOUD
abrionesr
+ Add Stack

SECURITY
Access Policies
Advanced Auth
OAuth Clients

SUPPORT
Get Support
Tickets

BILLING
My Account
Plan details
Manage plan
Current usage
Cost estimator
Invoice history
Billed usage history

ORG SETTINGS
My Plugins (0)

Members

Your organization has **2 members** and **0 invites**.

+ Invite New Member

Members (2)

Search...

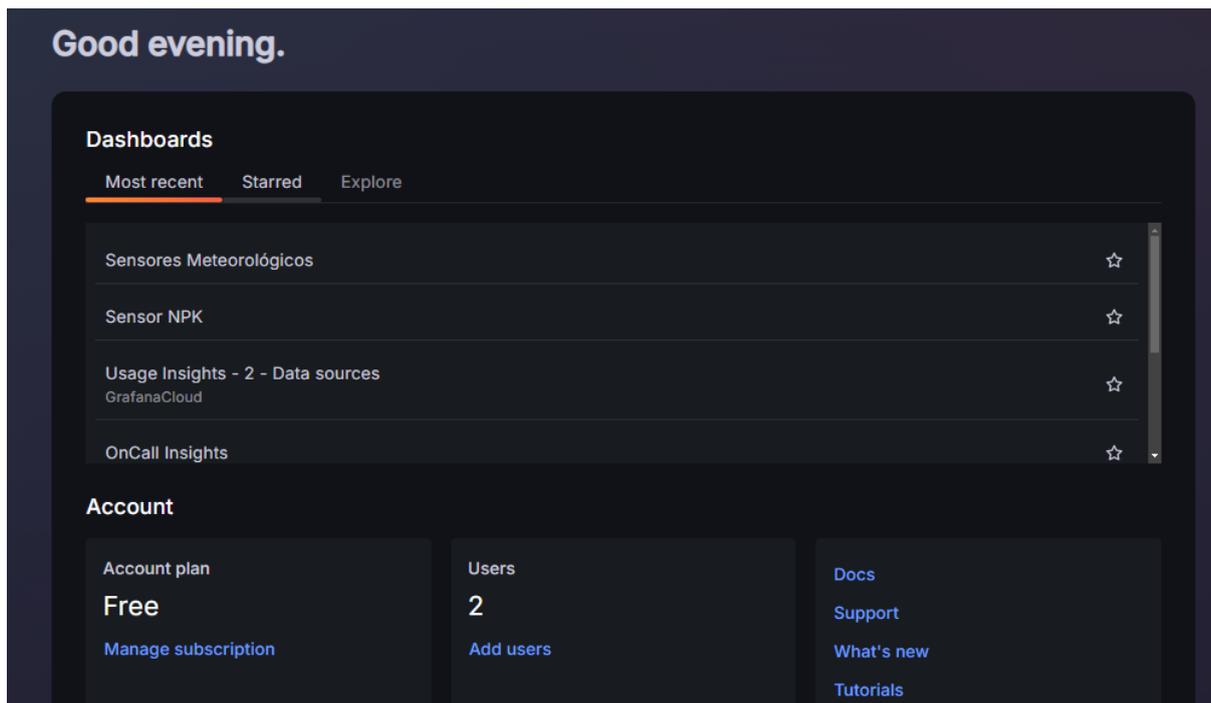
	abrionesr abrionesr@est.ups.edu.ec Angelo Briones Admin added on Jan 19, 2025 Receive billing email User Profile Update
	jsalmeronf jsalmeronf@est.ups.edu.ec Admin added on Jan 30, 2025 User Profile Update

Nota. Pantalla de Usuarios de Grafana Cloud, se visualiza creadores del proyecto

En la figura 36 se visualiza el dashboard principal que presenta el resumen del proyecto como los usuarios y los sensores.

Figura 36

Inicio de Grafana

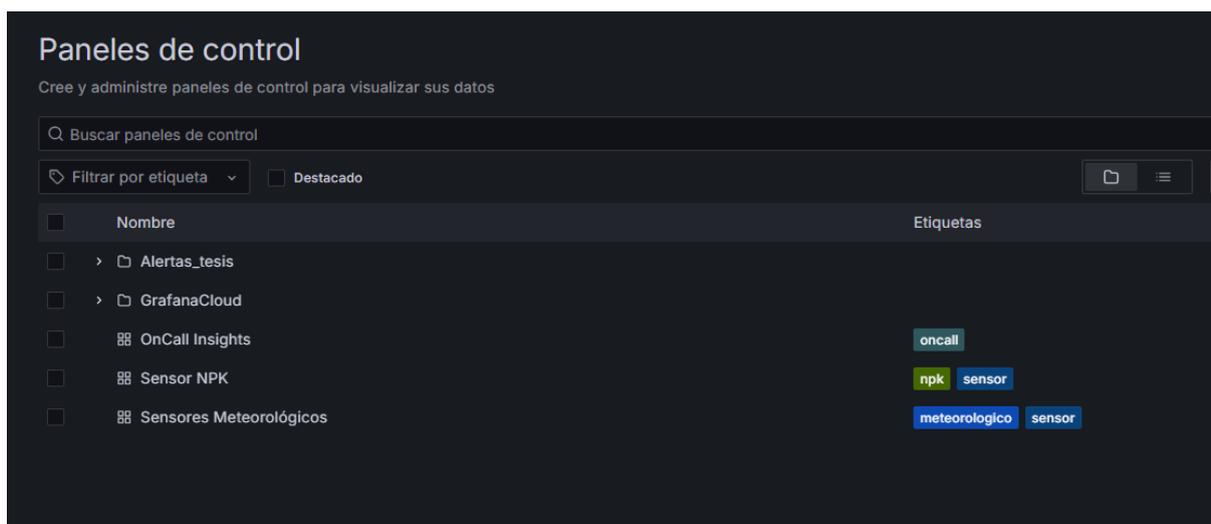


Nota. Inicio de Grafana Cloud muestra sensores usados para medir los parámetros ambientales, los nutrientes del suelo y miembros que pueden modificar el proyecto

En la figura 37 se observan los paneles creados por el usuario administrador y invitado.

Figura 37

Panel de control Dashboard Grafana Cloud

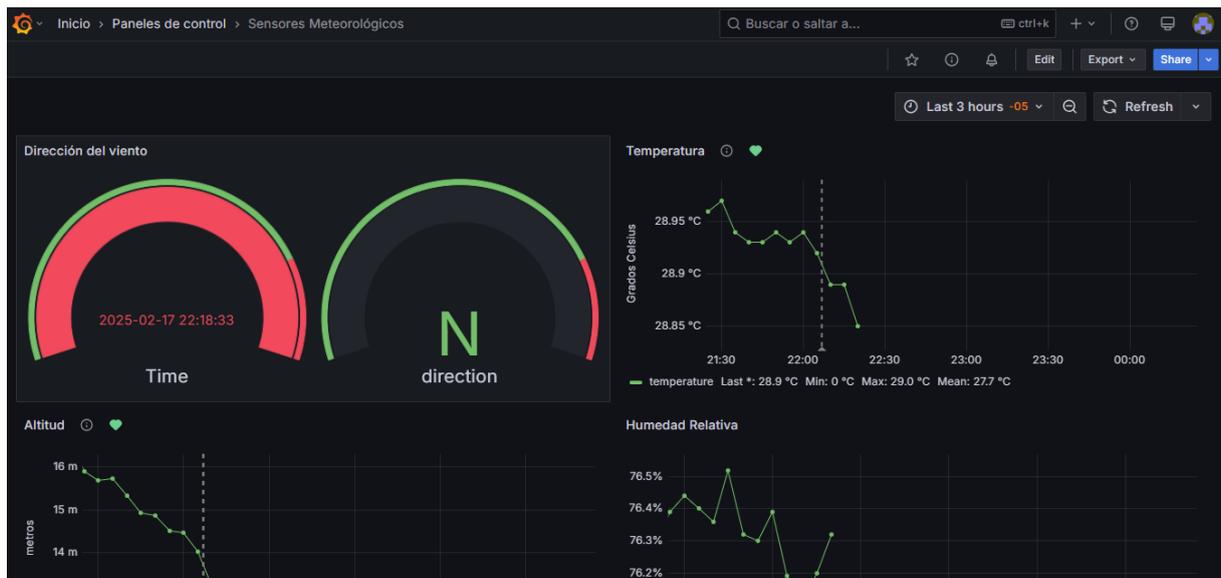


Nota. Panel de control Dashboard Grafana Cloud

En las figuras 38 y 39 se observa los datos obtenidos y almacenados por los sensores instalados y programados por la Lark Weather Station.

Figura 38

Dashboard de sensores de dirección de viento, temperatura y humedad



Nota. Dashboard muestra datos en tiempo real de sensores de dirección del viento, temperatura, altitud y humedad relativa

Figura 39

Dashboard de sensores de presión y velocidad

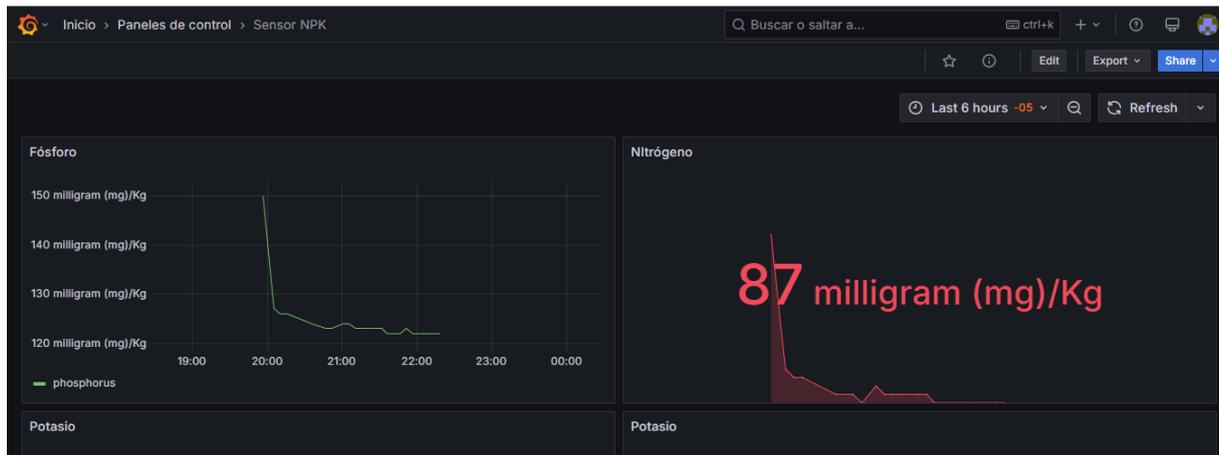


Nota. Dashboard muestra datos en tiempo real de sensores de presión atmosférica y velocidad el viento

A continuación, en la figura 40 se presenta los datos obtenidos y almacenados de los sensores NPK.

Figura 40

Datos de sensores NPK

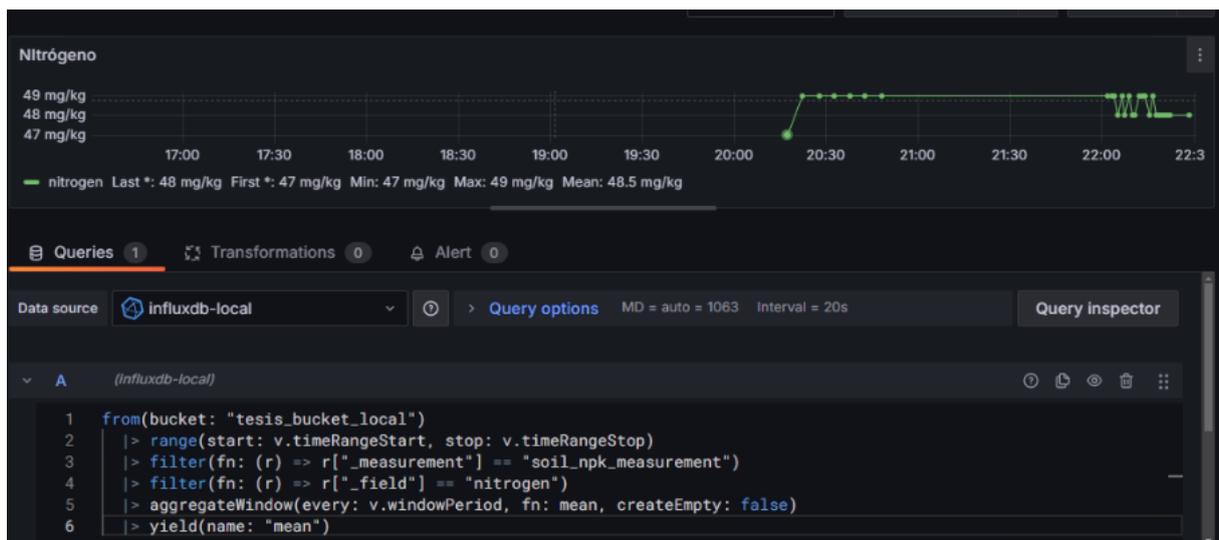


Nota. Dashboard muestra datos en tiempo real de sensores de Nitrógeno, potasio y fosforo

En la figura 41 se visualiza un código abierto de InfluxDB-local donde se filtra los datos obtenidos en los buckets configurados anteriormente.

Figura 41

Data source

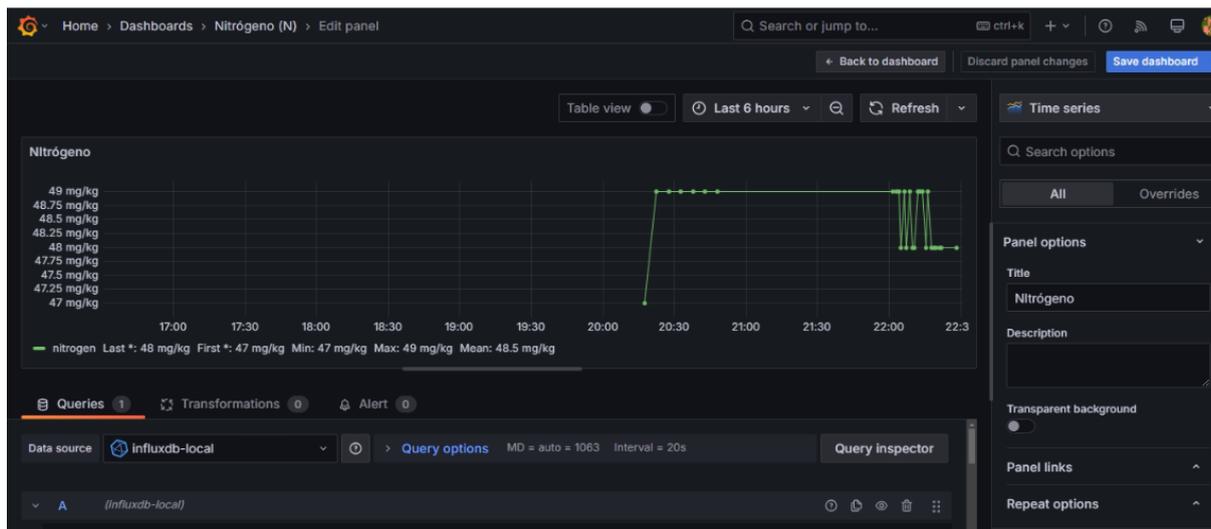


Nota. Grafana Cloud código abierto para filtrar datos de la bucket

En la figura 42 se presenta las diferentes configuraciones que se puede agregar en la dashboard con algunos parámetros disponibles.

Figura 42

Configuración del dashboard

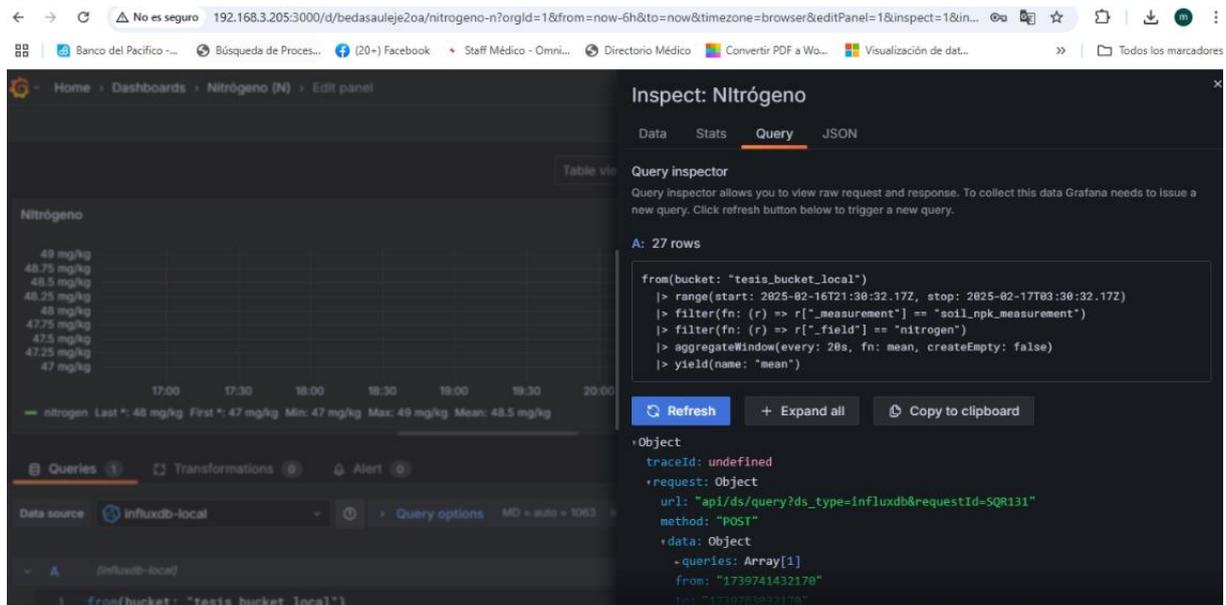


Nota. Configuración de la dashboard de nitrógeno nombre, forma, diseño, color y entre otros

En la figura 43, se visualiza un código libre representando los parámetros con un límite de programación de manera local.

Figura 43

Data Query

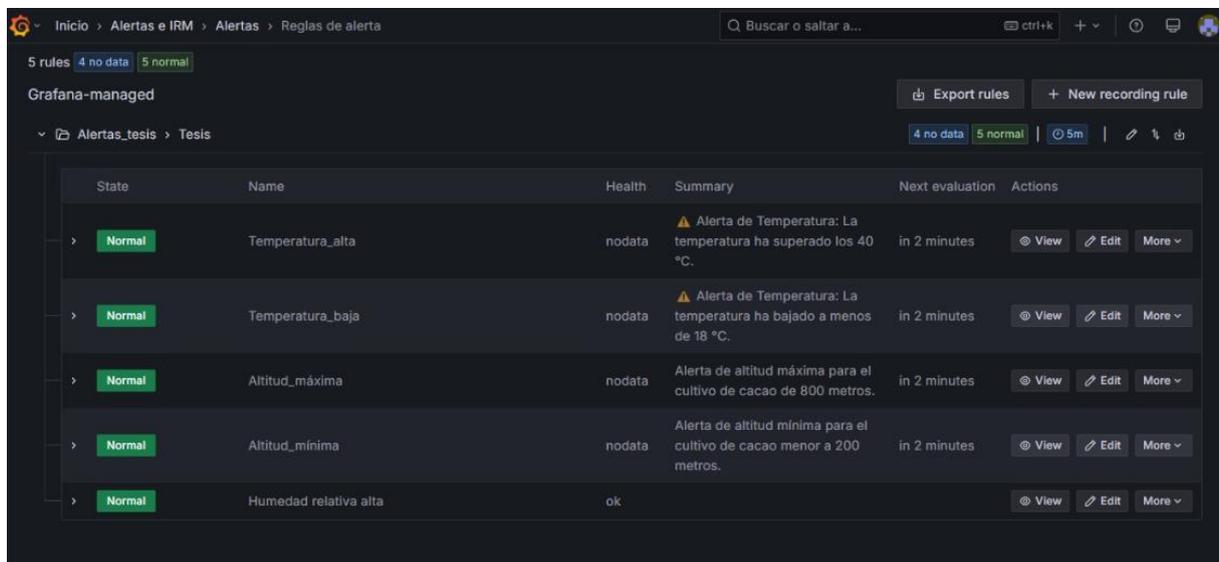


Nota. La data Query nos muestra cuantos datos de nitrógeno hemos obtenidos en el parámetro de tiempo establecido

En la figura 44 se observa el resultado de las configuraciones de las alertas por mail

Figura 44

Configuración de alertas mail



Nota. La configuración de alarmas en Grafana envía mensajes de alertas al mail cuando los sensores captan una data que sobrepasa o estén debajo de los parámetros requeridos

A continuación, en la figura 45, se observa la alerta cuando los sensores configurados

detectan una anomalía en el sistema, que incumplan con los parámetros establecidos

Figura 45

Alerta de altura mínima



Nota. Mensaje de alerta al Hotmail sobre altitud máxima superada

En la figura 46, se observa cuando otro parámetro preestablecido se incumple, en este caso particular demostrando como el sensor de altitud detecta que se ha superado la altitud máxima configurada.

Figura 46

Alerta de altura máxima

[FIRING:1] Altitud_máxima Alertas_tesis



Recibidos x



Grafana <noreply@grafana.net>
para abrionesr, jsalmeronf, mí

lun, 3 feb, 0:57 ☆ 😊 ↶ ⋮

Traducir al español x



Alertas_tesis › Altitud_máxima

1 firing instances

Firing	Altitud_máxima	View alert
--------	----------------	------------

Nota. Mail donde indica que el sensor de altitud alcanzo la altitud máxima establecida

8 CONCLUSIONES

En función del cumplimiento de los objetivos planteados en la investigación, se ha logrado desarrollar un prototipo IoT funcional basado en Raspberry Pi 4 para la monitorización de parámetros ambientales y de nutrientes del suelo en cultivos de cacao. La integración de sensores como Lark Weather Station y NPK permitió la recolección y procesamiento eficiente de datos, asegurando un monitoreo más preciso y continuo, facilitando el seguimiento de variables esenciales para el crecimiento del cultivo, optimizando su gestión y mejorando la toma de decisiones.

Se logró medir con precisión variables como temperatura, humedad, presión atmosférica, velocidad y dirección del viento, así como los niveles de nitrógeno, fósforo y potasio en el suelo. Gracias a la comunicación mediante el protocolo RS-485 y el almacenamiento en InfluxDB Cloud, el sistema pudo transmitir y visualizar estos datos de manera confiable en una plataforma de Grafana Cloud, permitiendo un análisis más detallado.

Las pruebas realizadas tanto en laboratorio como en campo demostraron que el sistema puede operar de forma autónoma y adaptarse a diferentes condiciones agrícolas,

garantizando datos precisos y confiables. Su diseño modular y escalable lo convierte en una herramienta útil para el monitoreo de cultivos, con la posibilidad de integrar mejoras futuras como modelos de predicción climática o automatización del riego.

En general, esta solución representa un aporte significativo a la gestión agrícola, brindando información en tiempo real que ayuda a optimizar los procesos de cultivo y fomentar prácticas más eficientes y sostenibles.

9 RECOMENDACIONES

Se recomienda optimizar el almacenamiento de datos en InfluxDB Cloud para un mejor rendimiento y funcionalidad del sistema IoT, se sugiere aplicar técnicas de algoritmo de inteligencia artificial para que analice los datos y prediga las condiciones climáticas y nutrientes del suelo.

Mejorar la capacidad en paneles solares garantizando un mayor rendimiento en la captación de radiación solar, y el almacenamiento eficiente de energía en las baterías para el prototipo que garantice el funcionamiento incluso en días con poca radiación solar.

Agregar sensores inteligentes que permita medir la calidad del carbono orgánico, conductividad eléctrica y el pH del suelo, para así obtener más información de un buen cultivo. Desarrollar una aplicación móvil que facilita el monitoreo en tiempo real, lo cual permitirá recibir alertas de cualquier anomalía fuera del margen de error.

Es importante realizar pruebas en distintos entornos agrícolas para evaluar cómo se adapta el sistema a diferentes condiciones climáticas y garantizar su estabilidad en distintos escenarios.

Por último, optimizar el código en Python de la Raspberry Pi 4 ayudará a reducir el consumo de energía y mejorar la velocidad de respuesta del sistema, asegurando un funcionamiento más eficiente.

10 BIBLIOGRAFÍA

- Administration, E. I. (2020). *Solar Photovoltaic Technology Basics*. U.S. Department of Energy. <https://www.energy.gov/eere/solar/solar-photovoltaic-technology-basics>
- Balladares Feijoo, C. L., & López Vargas, D. F. (2021). *Diseño e implementación de un prototipo de estación meteorológica agrícola autosustentable para el monitoreo de parámetros ambientales en cultivos de cacao mediante Raspberry Pi*. <https://dspace.ups.edu.ec/handle/123456789/20505>
- Barreiro, & Martínez. (2024). Diseño e implementación de un módulo interactivo IOT usando dispositivos LOW-COST y servicios Web. *DSpace UPS*. <http://dspace.ups.edu.ec/handle/123456789/27935>
- Bernal, C., & Enrique, Y. (2020). *Internet of Things (IoT) DISEÑO DE UNA RED DE IoT PARA EL HOGAR*. <https://repository.ucc.edu.co/entities/publication/59fecde5-c9ea-4497-96a4-2ecc0e668ee2>
- Bula, A. (2020). IMPORTANCIA DE LA AGRICULTURA EN EL DESARROLLO SOCIOECONÓMICO. In *Observatorio Económico Social UNR*. <https://observatorio.unr.edu.ar/wp-content/uploads/2020/08/Importancia-de-la-agricultura-en-el-desarrollo-socio-económico.pdf>
- Burbano Choez, E. X. (2022). *Análisis de mecanismos de seguridades en redes inalámbricas mediante IoT*. <http://repositorio.ucsg.edu.ec/handle/3317/20160>
- cadenaacacaoca.info. (2025). *Nutrición mineral del cacao*. https://cadenaacacaoca.info/CDOC-Deployment/documentos/NUTRICION_MINERAL_DEL_CACAO.pdf
- community.dfrobot.com. (2025). *Lark Weather Station: Monitoring the Five Weather Data - DFRobot Maker Community*. <https://community.dfrobot.com/makelog-313986.html>
- Megatronica.cc. (2025). *Controlador De Carga Solar 20A - MEGATRONICA*. <https://megatronica.cc/producto/controlador-de-carga-solar-20a/?srsltid=AfmBOoqLelo8Aiw8s55wsNMuSSZWNJhzUPITs2v-PMEwdoob53WE4ZyC>
- Ecuador, M. de A. y G. del. (2025). *MAGAP impulsa proyecto de reactivación del Cacao Fino y de Aroma*. <https://www.agricultura.gob.ec/magap-impulsa-proyecto-de-reactivacion-del-cacao-fino-y-de-aroma/>
- elproductor.com. (2019). Deficiencias nutricionales y fertilización del cacao. In *El Productor*. <https://elproductor.com/2019/04/deficiencias-nutricionales-y-fertilizacion-del-cacao/>
- Espinoza, J. (2017). *Explorando la nutrición foliar, crecimiento y rendimiento del cacao en relación*

- con cambios en la calidad del suelo, zona de Las Naves.
<https://repositorio.uteq.edu.ec/bitstream/43000/3285/1/T-UTEQ-0119.pdf>
- (FAO), O. de las N. U. para la A. y la A. (2012). *CACAO: Operaciones Postcosecha*. FAO.
<https://openknowledge.fao.org/server/api/core/bitstreams/053297ee-0c2b-4d9c-a871-1e71602dcfcd/content>
- Francisco-Santiago, Sandra Patricia, Palma-López, David Jesús, Sánchez-Hernández, Rufo, Obrador-Olán, José Jesús, & García-Alamilla, P. (2023). Fertilidad edáfica y nutrición en el cultivo de cacao (*Theobroma cacao* L.) en la Chontalpa, Tabasco, México. *Revista Mexicana de Ciencias Agrícolas*, [Volumen,([Número, si aplica]), [Páginas de inicio]-[Páginas de fin].
https://www.scielo.org.mx/scielo.php?pid=S0187-57792023000100113&script=sci_arttext
- Frutícola, P. (2024). *Requerimientos edafoclimáticos para el cultivo del cacao*.
<https://www.portalfruticola.com/noticias/2024/04/19/requerimientos-edafoclimaticos-para-el-cultivo-del-cacao/>
- Góngora-Duarte, A. F., Morales-Espitia, F. J., Trujillo-González, J. M., & Torres-Mora, M. A. (2023). Caracterización de los procesos en el beneficio del cacao (*Theobroma cacao* L.) en producciones a pequeña escala en el municipio de Guamal del Piedemonte llanero colombiano. *Tecnológicas*, 26(57), e2633.
<https://dialnet.unirioja.es/descarga/articulo/9518099.pdf>
- grafana.com. (2025). *Grafana: The open and composable observability platform | Grafana Labs*.
<https://grafana.com/>
- waveshare.com. (2025). *Industrial USB TO RS485 Bidirectional Converter, Onboard original CH343G, Multi-Protection Circuits | USB TO RS485 (B)*. <https://www.waveshare.com/usb-to-rs485-b.htm>
- influxdata.com. (2025). *InfluxDB Cloud | InfluxData*. <https://www.influxdata.com/products/influxdb-cloud/>
- wiki.dfrobot.com. (2025). *Lark Weather Station*.
https://wiki.dfrobot.com/SKU_EDU0157_EN_Lark_Weather_Station
- Lojano Angamarca, M. D., & Méndez Vásquez, R. B. (2023). *Desarrollo de un sistema de monitoreo y rastreo de productos agrícolas utilizando IoT, caso de estudio cacao*.
<https://dspace.ups.edu.ec/handle/123456789/24683>
- MOCCA. (2021). *Guía: Clima, altitud y suelo para sembrar cacao*. <https://mocca.org/wp-content/uploads/2021/08/guia-clima-altitud-y-suelo-para-sembrar-cacao.pdf>
- MOCCA. (2023). *Clima, altura y suelo recomendados para sembrar cacao*.

<https://www.youtube.com/watch?v=kpAidcpwI0o>

Pi, R. (2024). *Raspberry Pi 4*. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

PrismaB. (2024). *Ventajas de instalar un sensor ambiental en cultivos*.
<https://prismab.com/blog/ventajas-de-instalar-un-sensor-ambiental-en-cultivos/>

Productor, E. (2019). *El cultivo del cacao: clima y suelo*. <https://elproductor.com/2019/04/el-cultivo-del-cacao-clima-y-suelo/>

Reddy, T. S. (2015). *Electric Vehicle Battery Systems*. Elsevier.
<https://www.elsevier.com/books/electric-vehicle-battery-systems/reddy/978-0-12-801759-5>

Relief, L. W. (2011). *CACAO ORGÁNICO: Manual de Cultivo*. Lutheran World Relief.
https://indepth.lwr.org/sites/default/files/2018-09/MANUAL_DE_CACAO_ORGANICO.pdf

ozonomajes.com. (2025). Sensor NPK - Ozono Majes. <https://Ozonomajes.Com/>.
<https://ozonomajes.com/sensor-npk/>

uelectronics.com. (2025). *XL4015 Modulo Regulador Step Down 50W 5A - UNIT Electronics*.
https://uelectronics.com/producto/modulo-regulador-step-up-1-25-35v-xl4015-5-40v-5a/?srsltid=AfmBOoqM1YE-a12V8BFqYXdU0xpQxuDHlhDIE_7ILb8EYcuO-r88M_MG

Zarrillo, S., Gaikwad, N., Lanaud, C., Powis, T., & Viot, C. (2018). The use and domestication of *Theobroma cacao* during the mid-Holocene in the upper Amazon. *Nature Ecology & Evolution*, 2(12), 1879–1888. <https://doi.org/10.1038/s41559-018-0697-x>

11 ANEXOS

11.1 Código Python en Raspberry PI envío de datos a InfluxDB Cloud

```
# -*- coding: utf-8 -*-
import time
import os
import sys
from influxdb_client import InfluxDBClient, Point, WritePrecision
from dotenv import load_dotenv
from DFRobot_LarkWeatherStation import DFRobot_LarkWeatherStation_I2C
from pymodbus.client import ModbusSerialClient as ModbusClient

# Cargar las variables de entorno desde el archivo .env
load_dotenv()

# Configuración de InfluxDB Cloud
INFLUXDB_URL=os.getenv("INFLUXDB_URL","https://us-east-1-1.aws.cloud2.influxdata.com")
INFLUXDB_TOKEN=os.getenv("INFLUXDB_TOKEN","0SGPuio3LhrpgdXDo0nKEX2tnB7I23zaxrsjHZsy6USoZLNo_nMsGB8JX17VM5RLP7qZVekOhCTnzUqx5UB40g==")
INFLUXDB_ORG = os.getenv("INFLUXDB_ORG", "UPS")
INFLUXDB_BUCKET = os.getenv("INFLUXDB_BUCKET", "tesis_bucket")

# Cliente de InfluxDB
client_influx=InfluxDBClient(url=INFLUXDB_URL,token=INFLUXDB_TOKEN,org=INFLUXDB_ORG)
write_api = client_influx.write_api()

# Configuración del sensor meteorológico
ADDRESS = 0x42
EDU0157 = DFRobot_LarkWeatherStation_I2C(ADDRESS)

# Configuración del sensor NPK (RS-485)
```

```

NPK_PORT = '/dev/ttyUSB0'
client_npk = ModbusClient(port=NPK_PORT, baudrate=9600, parity='N', stopbits=1,
bytesize=8, timeout=1)

def setup_meteorologico():
    """Inicializa el sensor meteorológico."""
    while EDU0157.begin() != 0:
        print("Error al inicializar el sensor meteorológico.")
        time.sleep(1)
    print("Sensor meteorológico inicializado correctamente.")

def setup_npk():
    """Intenta conectar con el sensor NPK."""
    if client_npk.connect():
        print("Conexión exitosa con el sensor NPK.")
    else:
        print("No se pudo conectar al sensor NPK.")
        sys.exit(1)

def safe_float(value, default=0.0):
    """Convierte un valor a float de manera segura, devolviendo un valor por defecto si falla."""
    try:
        return float(value)
    except (ValueError, TypeError):
        return default

def leer_meteorologico():
    try:
        data = {
            "speed": (safe_float(EDU0157.get_value("Speed")), EDU0157.get_unit("Speed")),
            "direction": (EDU0157.get_value("Dir"), ""),
            "temperature":(safe_float(EDU0157.get_value("Temp")), EDU0157.get_unit("Temp")),
            "humidity": (safe_float(EDU0157.get_value("Humi")), EDU0157.get_unit("Humi")),

```

```

        "pressure":(safe_float(EDU0157.get_value("Pressure")),
EDU0157.get_unit("Pressure")),
        "altitude": (safe_float(EDU0157.get_value("Altitude")), EDU0157.get_unit("Altitude")),
    }
except Exception as e:
    print(f"Error al leer el sensor meteorologico: {e}")
    return None # Retorna None si hay un error

# Imprimir datos
print("\n--- Datos MeteorolÃ³gicos ---")
print(f"Velocidad del viento: {data['speed'][0]} {data['speed'][1]}")
print(f"DirecciÃ³n del viento: {data['direction'][0]}")
print(f"Temperatura: {data['temperature'][0]} {data['temperature'][1]}")
print(f"Humedad: {data['humidity'][0]} {data['humidity'][1]}")
print(f"PresiÃ³n: {data['pressure'][0]} {data['pressure'][1]}")
print(f"Altitud: {data['altitude'][0]} {data['altitude'][1]}")

return data

def leer_npk():
    """Lee datos del sensor NPK y los imprime."""
    try:
        result = client_npk.read_holding_registers(30, count=3, slave=1)
        if result.isError():
            print("Error en la lectura del sensor NPK.")
            return None
    except Exception as e:
        print(f"Error al leer el sensor NPK: {e}")
        return None

data = {
    "Nitrogeno": safe_float(result.registers[0]),
    "Fosforo": safe_float(result.registers[1]),

```

```

        "Potasio": safe_float(result.registers[2]),
    }

    # Imprimir datos
    print("\n--- Datos del Sensor NPK ---")
    for key, value in data.items():
        print(f"{key}: {value} mg/kg")

    return data

def enviar_datos_influxdb(datos_meteorologicos, datos_npk):
    """Crea y envÃa los puntos a InfluxDB."""
    if datos_meteorologicos:
        point_weather = (
            Point("weather_measurement_v2")
            .field("speed", datos_meteorologicos["speed"][0])
            .field("direction", datos_meteorologicos["direction"][0])
            .field("temperature", datos_meteorologicos["temperature"][0])
            .field("humidity", datos_meteorologicos["humidity"][0])
            .field("pressure", datos_meteorologicos["pressure"][0])
            .field("altitude", datos_meteorologicos["altitude"][0])
        )
        write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,
record=point_weather)
        print("Datos meteorolÃgicos enviados a InfluxDB.")

    if datos_npk:
        point_npk = (
            Point("soil_npk_measurement_v2")
            .field("nitrogen", datos_npk["Nitrogeno"])
            .field("phosphorus", datos_npk["Fosforo"])
            .field("potassium", datos_npk["Potasio"])
        )

```

```
write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG, record=point_npk)
print("Datos del sensor NPK enviados a InfluxDB.")
```

```
def main():
```

```
    """Funcion principal."""
```

```
    setup_meteorologico()
```

```
    setup_npk()
```

```
    next_send_time = time.time() + 300 # 5 minutos
```

```
    while True:
```

```
        try:
```

```
            print("\n---- Lectura de Sensores ----")
```

```
            datos_meteorologicos = leer_meteorologico()
```

```
            datos_npk = leer_npk()
```

```
            if datos_meteorologicos is None or datos_npk is None:
```

```
                print("Error critico en la lectura de sensores. Reintentando en 10 segundos...")
```

```
                time.sleep(10)
```

```
                continue
```

```
            # Enviar datos a InfluxDB cada 5 minutos
```

```
            if time.time() >= next_send_time:
```

```
                enviar_datos_influxdb(datos_meteorologicos, datos_npk)
```

```
                next_send_time = time.time() + 300
```

```
            time.sleep(10)
```

```
        except KeyboardInterrupt:
```

```
            print("\nLectura detenida por el usuario.")
```

```
            break
```

```
        except Exception as e:
```

```
            print(f"Error inesperado: {e}")
```

```
            print("Reintentando en 10 segundos...")
```

```

        time.sleep(10)

    client_npk.close()
    client_influx.close()
    print("Conexion con sensores cerrada correctamente.")

if __name__ == "__main__":
    main()

```

11.2 Código Python en Raspberry PI envío de datos a InfluxdDB Local

```

# -*- coding: utf-8 -*-
import time
import os
import sys
sys.path.append("/home/tesis/myenv/DFRobot_LarkWeatherStation/python/raspberry")
from DFRobot_LarkWeatherStation import DFRobot_LarkWeatherStation_I2C
from influxdb_client import InfluxDBClient, Point, WritePrecision
from dotenv import load_dotenv
from pymodbus.client import ModbusSerialClient as ModbusClient

# Cargar las variables de entorno desde el archivo .env
load_dotenv()

# Configuración de InfluxDB Local
INFLUXDB_URL = os.getenv("INFLUXDB_URL", "http://localhost:8086")
INFLUXDB_TOKEN = os.getenv("INFLUXDB_TOKEN",
"7XvEAsUzM_O1xa0Ft8uR1ps1wOKg7rSg4dqCUwtJUaT7rPJIGDBUTb3u4RCoen67SMgW
HfNOsgpdPprxAy-oZQ==")
INFLUXDB_ORG = os.getenv("INFLUXDB_ORG", "UPS_local")
INFLUXDB_BUCKET = os.getenv("INFLUXDB_BUCKET", "tesis_bucket_local")

# Cliente de InfluxDB
client_influx = InfluxDBClient(url=INFLUXDB_URL, token=INFLUXDB_TOKEN,
org=INFLUXDB_ORG)

```

```

write_api = client_influx.write_api()

# Configuraci3n del sensor meteorol3gico
ADDRESS = 0x42
EDU0157 = DFRobot_LarkWeatherStation_I2C(ADDRESS)

# Configuraci3n del sensor NPK (RS-485)
NPK_PORT = '/dev/ttyUSB0'
client_npk = ModbusClient(port=NPK_PORT, baudrate=9600, parity='N', stopbits=1,
bytesize=8, timeout=1)

# Constante para el intervalo de env3o (en segundos)
ENVIO_INTERVALO = 300 # 5 minutos

def setup_meteorologico():
    """Inicializa el sensor meteorol3gico."""
    while EDU0157.begin() != 0:
        print("Error al inicializar el sensor meteorol3gico. Reintentando...")
        time.sleep(1)
    print("Sensor meteorol3gico inicializado correctamente.")

def setup_npk():
    """Intenta conectar con el sensor NPK."""
    if client_npk.connect():
        print("Conexi3n exitosa con el sensor NPK.")
    else:
        print("No se pudo conectar al sensor NPK.")
        sys.exit(1)

def safe_float(value, default=0.0):

```

```
"""Convierte un valor a float de manera segura, devolviendo un valor por defecto si falla."""
```

```
try:
```

```
    return float(value)
```

```
except (ValueError, TypeError):
```

```
    return default
```

```
def leer_meteorologico():
```

```
    """Lee datos del sensor meteorológico y los imprime con sus unidades."""
```

```
    try:
```

```
        data = {
```

```
            "speed": (safe_float(EDU0157.get_value("Speed")), EDU0157.get_unit("Speed")),
```

```
            "direction": (EDU0157.get_value("Dir"), ""),
```

```
            "temperature": (safe_float(EDU0157.get_value("Temp")),
```

```
EDU0157.get_unit("Temp")),
```

```
            "humidity": (safe_float(EDU0157.get_value("Humi")), EDU0157.get_unit("Humi")),
```

```
            "pressure": (safe_float(EDU0157.get_value("Pressure")),
```

```
EDU0157.get_unit("Pressure")),
```

```
            "altitude": (safe_float(EDU0157.get_value("Altitude")), EDU0157.get_unit("Altitude")),
```

```
        }
```

```
    except Exception as e:
```

```
        print(f"Error al leer el sensor meteorológico: {e}")
```

```
        return None
```

```
# Imprimir datos
```

```
print("\n--- Datos Meteorológicos ---")
```

```
print(f"Velocidad del viento: {data['speed'][0]} {data['speed'][1]}")
```

```
print(f"Dirección del viento: {data['direction'][0]}")
```

```
print(f"Temperatura: {data['temperature'][0]} {data['temperature'][1]}")
```

```
print(f"Humedad: {data['humidity'][0]} {data['humidity'][1]}")
```

```
print(f"Presión: {data['pressure'][0]} {data['pressure'][1]}")
```

```
print(f"Altitud: {data['altitude'][0]} {data['altitude'][1]}")
```

```
return data
```

```
def leer_npk():
```

```
    """Lee datos del sensor NPK y los imprime."""
```

```
    try:
```

```
        result = client_npk.read_holding_registers(30, count=3, slave=1)
```

```
        if result.isError():
```

```
            print("Error en la lectura del sensor NPK.")
```

```
            return None
```

```
    except Exception as e:
```

```
        print(f"Error al leer el sensor NPK: {e}")
```

```
        return None
```

```
    data = {
```

```
        "Nitrogeno": safe_float(result.registers[0]),
```

```
        "Fosforo": safe_float(result.registers[1]),
```

```
        "Potasio": safe_float(result.registers[2]),
```

```
    }
```

```
    # Imprimir datos
```

```
    print("\n--- Datos del Sensor NPK ---")
```

```
    for key, value in data.items():
```

```
        print(f"{key}: {value} mg/kg")
```

```
    return data
```

```
def enviar_datos_influxdb(datos_meteorologicos, datos_npk):
```

```
    """Crea y envÃ-a los puntos a InfluxDB."""
```

```
    try:
```

```
        if datos_meteorologicos:
```

```
            point_weather = (
```

```

        Point("weather_measurement")
        .field("speed", datos_meteorologicos["speed"][0])
        .field("direction", datos_meteorologicos["direction"][0])
        .field("temperature", datos_meteorologicos["temperature"][0])
        .field("humidity", datos_meteorologicos["humidity"][0])
        .field("pressure", datos_meteorologicos["pressure"][0])
        .field("altitude", datos_meteorologicos["altitude"][0])
    )
    write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,
record=point_weather)
    print("Datos meteorológicos enviados a InfluxDB.")

if datos_npk:
    point_npk = (
        Point("soil_npk_measurement")
        .field("nitrogen", datos_npk["Nitrogeno"])
        .field("phosphorus", datos_npk["Fosforo"])
        .field("potassium", datos_npk["Potasio"])
    )
    write_api.write(bucket=INFLUXDB_BUCKET, org=INFLUXDB_ORG,
record=point_npk)
    print("Datos del sensor NPK enviados a InfluxDB.")
except Exception as e:
    print(f"Error al enviar datos a InfluxDB: {e}")

def main():
    """Función principal."""
    setup_meteorologico()
    setup_npk()

    next_send_time = time.time() + ENVIO_INTERVALO

```

```

while True:
    try:
        print("\n---- Lectura de Sensores ----")
        datos_meteorologicos = leer_meteorologico()
        datos_npk = leer_npk()

        if datos_meteorologicos is None or datos_npk is None:
            print("Error en la lectura de sensores. Reintentando en 10 segundos...")
            time.sleep(10)
            continue

        if time.time() >= next_send_time:
            enviar_datos_influxdb(datos_meteorologicos, datos_npk)
            next_send_time = time.time() + ENVIO_INTERVALO

        time.sleep(10)
    except KeyboardInterrupt:
        print("\nLectura detenida por el usuario.")
        break
    except Exception as e:
        print(f"Error inesperado: {e}")
        print("Reintentando en 10 segundos...")
        time.sleep(10)

client_npk.close()
client_influx.close()
print("Conexión con sensores cerrada correctamente.")

if __name__ == "__main__":
    main()

```

11.3 Script Batch de encendido automático en Raspberry Pi

```

#!/bin/bash

# Archivo para bloqueo
LOCKFILE="/tmp/check_internet_and_run.lock"

# Intenta adquirir el bloqueo exclusivo
exec 200>"$LOCKFILE"
flock -n 200 || exit 1 # Salir si ya hay otra instancia en ejecución

# Esperar 60 segundos para permitir que el sistema se estabilice
sleep 10

# Comprobar conexión a Internet con reintentos (máximo 5 intentos)
MAX_RETRIES=5
RETRY_DELAY=10
ATTEMPT=0
INTERNET_AVAILABLE=0

echo "Comprobando conexión a Internet con reintentos..."

# Reintentar la conexión a Internet
while [ "$ATTEMPT" -lt "$MAX_RETRIES" ]; do
    if ping -c 1 8.8.8.8 &> /dev/null; then
        INTERNET_AVAILABLE=1
        break
    fi
    ATTEMPT=$((ATTEMPT + 1))
    echo "$(date): Intento $ATTEMPT de $MAX_RETRIES fallido. Reintentando en
$RETRY_DELAY segundos..."
    sleep "$RETRY_DELAY"
done

# Ejecutar el script correspondiente

```

```
if [ "$INTERNET_AVAILABLE" -eq 1 ]; then
    echo "$(date): Internet disponible. Ejecutando Proyecto final..."

    # Activar el ambiente virtual y ejecutar el script Python
    source /home/tesis/nuevo_env/bin/activate
    python /home/tesis/nuevo_env/DFRobot_LarkWeatherStation/proyecto_final2.py
    deactivate
else
    echo "$(date): No hay conexión a Internet. Ejecutando run_weather_station.sh..."

    # Activar el ambiente virtual y ejecutar el script alternativo
    source /home/tesis/myenv/bin/activate
    python /home/tesis/myenv/DFRobot_LarkWeatherStation/proyecto_final_local.py
    deactivate
fi

echo "$(date): Script finalizado."
```