

Universidad Nacional de La Pampa, Facultad de Ingeniería

PROYECTO FINAL DE GRADO DE INGENIERÍA EN COMPUTACIÓN

“FertSens: construcción de un dispositivo para medir parámetros químicos en suelos agrícolas”

Autor: Leandro Gabriel Centelles.

Grado académico: Ingeniería en Computación.

Tutor: Ing. Ricardo Augusto Furch – Cátedras Redes y Comunicaciones II y Sistemas Embebidos.

Co-tutor: Ing. Diego Alberto Vicente – Cátedras Introducción a la Automatización Industrial, Maquinas Hidráulicas y Conocimiento de Materiales.

Lugar de presentación: General Pico

Año: 2024

Fecha de Aprobación: 5 de septiembre

Jurados: Olsina Santos, Luis Antonio - Becker, Pablo Javier - Bergés, Ernesto Daniel. Filiación: Facultad Ingeniería (UNLPam).

Resumen: en este Trabajo Final se presenta el desarrollo y construcción de un dispositivo para medir parámetros químicos esenciales en suelos agrícolas, específicamente nitrógeno, potasio y fósforo. La investigación incluyó un análisis de los nutrientes necesarios para el desarrollo de los cultivos, así como el número, forma y frecuencia con la que se requieren tomar muestras del suelo. Además, se abordó la arquitectura IoT, los protocolos e interfaces de comunicación involucrados y los componentes electrónicos necesarios para el diseño del dispositivo. El mismo fue ensamblado y sometido a pruebas de campo, cuyos resultados fueron comparados con los obtenidos mediante análisis químico en un laboratorio. Los resultados demuestran la eficacia y precisión del dispositivo, destacando su potencial como herramienta para agricultores y profesionales del área agronómica en la monitorización de la calidad del suelo y la optimización de la fertilización.

Palabras clave: Dispositivos IoT, Parámetros Químicos, Suelos Agrícolas, Monitoreo de Suelo, Optimización de la Fertilización.

Abstract: this Final Project presents the development and construction of a device for measuring essential chemical parameters in agricultural soils, specifically nitrogen, potassium, and phosphorus. The research included an analysis of the nutrients required for crop development, as well as the number, form, and frequency with which soil samples need to be taken. Additionally, the IoT architecture, the communication protocols and interfaces involved, and the electronic components necessary for the design of the device were addressed. The device was assembled and subjected to field tests, whose results were compared with those obtained through chemical analysis in a laboratory. The results demonstrate the device's effectiveness and precision, highlighting its potential as a tool for farmers and professionals in the agronomy field in monitoring soil quality and optimizing fertilization.

Key Words: IoT Devices, Chemical Parameters, Agricultural Soils, Soil Monitoring, Fertilization Optimization.



FACULTAD DE INGENIERÍA
Universidad Nacional de La Pampa

FertSens: construcción de un dispositivo para medir parámetros químicos en suelos agrícolas

PROYECTO FINAL DE CARRERA

Ingeniería en Computación

Autor

Centelles, Leandro Gabriel

Director

Ing. Furch, Ricardo Augusto

Codirector

Ing. Vicente, Diego Alberto

Jurado

Dr. Becker, Pablo Javier

Ing. Bergés, Ernesto Daniel

Dr. Olsina Santos, Luis Antonio

General Pico, La Pampa

Julio, 2024

Agradecimientos

Este trabajo final le pone fin a una etapa que tuvo muchas emociones y en la que crecí tanto a nivel académico como personal. Quiero reconocer a aquellas personas que estuvieron junto a mí en todo este tiempo.

En primer lugar, agradecer a la Facultad de Ingeniería de la Universidad Nacional de La Pampa por permitirme formar como profesional.

Gracias a mi director, ingeniero Ricardo Furch, y a mi co-director, ingeniero Diego Vicente, por sus consejos y supervisión en el desarrollo de esta tesis. Su experiencia y compromiso han sido una fuente de aprendizaje y motivación.

Por último, gracias a mi familia y amigos por su compañía y apoyo incondicional a lo largo de estos años. Sin ustedes, este logro no habría sido posible.

Resumen

En este Trabajo Final se presenta el desarrollo y construcción de un dispositivo para medir parámetros químicos esenciales en suelos agrícolas, específicamente nitrógeno, potasio y fósforo. La investigación incluyó un análisis de los nutrientes necesarios para el desarrollo de los cultivos, así como el número, forma y frecuencia con la que se requieren tomar muestras del suelo. Además, se abordó la arquitectura IoT, los protocolos e interfaces de comunicación involucrados y los componentes electrónicos necesarios para el diseño del dispositivo. El mismo fue ensamblado y sometido a pruebas de campo, cuyos resultados fueron comparados con los obtenidos mediante análisis químico en un laboratorio. Los resultados demuestran la eficacia y precisión del dispositivo, destacando su potencial como herramienta para agricultores y profesionales del área agronómica en la monitorización de la calidad del suelo y la optimización de la fertilización.

Índice

Índice de figuras	2
Índice de tablas	3
Índice de códigos	4
1 Introducción	5
1.1 Contextualización del problema	5
1.2 Objetivo	8
1.3 Estructura del Trabajo Final de Grado	8
2 Fundamentos teóricos	10
2.1 Nutrientes para el crecimiento del cultivo	10
2.1.1 Nitrógeno (N)	12
2.1.2 Fósforo (P)	13
2.1.3 Potasio (K)	14
2.1.4 Magnesio (Mg)	14
2.1.5 Azufre (S)	15
2.1.6 Calcio (Ca)	15
2.1.7 Micronutrientes	15
2.2 Potencial de Hidrógeno (pH)	15
2.3 Fertilizantes	17
2.3.1 Utilización de fertilizantes	17
2.4 Análisis del suelo	18
2.4.1 Número de muestras y submuestras	18
2.4.2 Forma de muestreo	19
2.4.3 Frecuencia y momento de muestreo	20
2.4.4 Profundidad de muestreo	21
2.5 Internet of Things (IoT)	21
2.5.1 Arquitectura IoT	22
2.5.2 IoT y la agricultura	24

2.6	Arquitectura de protocolos	24
2.7	Interfaces de comunicación	26
2.7.1	I2C	26
2.7.2	SPI	27
2.8	Protocolos de comunicación	28
2.8.1	RS-485	29
2.8.2	Modbus RTU	29
2.9	Reguladores de voltaje	32
2.10	Tecnologías de memoria	34
3	Desarrollo	36
3.1	Especificación y análisis del proyecto	36
3.1.1	Especificación de los requisitos del sistema	36
3.1.2	Estándar de codificación	37
3.2	Componentes	38
3.2.1	Placa de desarrollo	39
3.2.2	Batería recargable 18650	40
3.2.3	Sensor de suelo NPK	42
3.2.4	Módulo conversor RS-485 a TTL	44
3.2.5	Módulo step-up MT3608	45
3.2.6	Módulo TP4056	46
3.2.7	Pantalla OLED	48
3.2.8	Módulo RTC (Real Time Clock)	48
3.2.9	Módulo lector de memorias SD	50
3.2.10	Buzzer activo	50
3.3	Conexión física	51
3.4	Implementación	54
3.4.1	InfluxDB	58
3.4.2	Grafana	61
3.5	Interfaz gráfica	65
3.6	Firmware	66

3.7	Exportación de datos para análisis del usuario	68
4	Construcción de FertSens	69
4.1	Dispositivo final	69
5	Caso de estudio	73
5.1	Pruebas y resultados obtenidos	73
6	Conclusión	75
7	Referencias	77
8	Anexo	81
8.1	Pinout de la placa de desarrollo	81
8.2	Estructura interna del MAX485	81
8.3	Prototipos	82
8.4	Código	83
8.4.1	Función para subir los registros a InfluxDB	83
8.4.2	Script para el tratamiento de datos	86
8.5	Imágenes de FertSens	88

Índice de figuras

1.1	Impacto de la agricultura sobre el suelo. Extraído de [Andrade et al., 2017]	7
1.2	Logo del dispositivo.	8
2.1	Porcentaje de nutrientes en las plantas. Extraído de [FAO, 2002]	11
2.2	Rendimiento limitado por los nutrientes. Extraído de [FAO, 2002]	12
2.3	Formas de muestreo sistemático. Extraído de [Red BPA, 2020]	19
2.4	Formas de muestreo al azar. Extraído de [Red BPA, 2020]	20
2.5	Capas del modelo OSI. Extraído de [Cloudflare, 2024]	26
2.6	Inicio y parada - I2C. Extraído de [Afzal, S., 2016]	27
3.1	Vista superior de la placa de desarrollo. Extraído de [WiFi Kit 32 Datasheet]	39
3.2	Vista inferior de la placa de desarrollo. Extraído de [WiFi Kit 32 Datasheet]	39
3.3	Batería 18650. Extraído de [18650 Battery Datasheet]	41
3.4	Sensor NPK Pilipane. Extraído de Amazon.	42
3.5	Sensor NPK Rapitest. Extraído de Amazon.	42
3.6	Sensor NPKPHCTH-S. Extraído de Amazon.	43
3.7	Sensor NPK. Extraído de [Soil NPK sensor Datasheet]	43
3.8	Convertor RS-485 a TTL. Extraído de [MAX485 Datasheet]	44
3.9	Módulo step-up. Extraído de [MT3608 Datasheet]	46
3.10	Módulo TP4056. Extraído de [TP4056 Datasheet]	47
3.11	Pantalla OLED. Extraído de [SH1106 Datasheet]	48
3.12	Módulo RTC. Extraído de [DS1307 Datasheet]	49
3.13	Módulo lector de memorias SD. Extraído de [74LVC125A Datasheet]	50
3.14	Buzzer activo. Extraído de [Buzzer Datasheet]	51
3.15	Diagrama pictórico.	52
3.16	Diagrama esquemático.	53
3.17	Logo de InfluxDB. Extraído de [InfluxData, 2024]	59
3.18	Logo de Grafana. Extraído de [Grafana, 2024]	61
3.19	Panel de nitrógeno - Grafana.	62
3.20	Panel de fósforo - Grafana.	63
3.21	Panel de potasio - Grafana.	63

3.22	Panel de niveles NPK - Grafana.	64
3.23	Panel de promedio de NPK - Grafana.	64
3.24	Pantalla de inicio.	65
3.25	Pantalla de resultado.	65
3.26	Diagrama de flujo de la interfaz de usuario.	66
3.27	Diagrama de flujo del firmware.	67
3.28	Planilla de Excel con los registros.	68
4.1	Carcasa del dispositivo - vista frontal.	69
4.2	Carcasa del dispositivo - vista superior.	69
4.3	Tapa del dispositivo.	70
4.4	Diseño de la PCB.	70
4.5	PCB del dispositivo.	71
4.6	Dispositivo final - vista superior.	71
4.7	Dispositivo final - vista frontal.	72
4.8	Dispositivo final - tapa.	72
5.1	Muestras de tierra.	73
8.1	Pinout de la placa de desarrollo. Extraído de [ESP32 WiFi Kit Datasheet] . . .	81
8.2	Estructura interna MAX485. Extraído de [MAX485 Datasheet]	82
8.3	Pantalla de introducir sensor en suelo.	82
8.4	Pantalla de espera.	82
8.5	Pantalla de retire el sensor del suelo.	83
8.6	Pantalla de inicio - FertSens	88
8.7	Pantalla de introducir sensor en suelo - FertSens	88
8.8	Pantalla de espera - FertSens	89
8.9	Pantalla de retirar el sensor - FertSens	89
8.10	Pantalla de un registro - FertSens	89
8.11	Pantalla de falta de tarjeta SD - FertSens	90

Índice de tablas

2.1	Tipos de suelo según el valor de pH. Extraído de [Garrido, S., 1993]	16
2.2	Trama Modbus RTU.	30
2.3	Solicitud - trama Modbus RTU.	31
2.4	Respuesta - trama Modbus RTU.	31
2.5	Solicitud del nivel de nitrógeno - trama Modbus RTU.	31
2.6	Respuesta del nivel de nitrógeno - trama Modbus RTU.	31
2.7	Solicitud del nivel de fósforo - trama Modbus RTU.	32
2.8	Respuesta del nivel de fósforo - trama Modbus RTU.	32
2.9	Solicitud del nivel de potasio - trama Modbus RTU.	32
2.10	Respuesta del nivel de potasio - trama Modbus RTU.	32
3.1	Especificaciones técnicas de la placa de desarrollo.	40
3.2	Especificaciones técnicas de la batería 18650.	41
3.3	Especificaciones técnicas del sensor NPK.	44
3.4	Especificaciones técnicas del módulo RS-485 a TTL.	45
3.5	Especificaciones técnicas del módulo step-up MT3608.	46
3.6	Especificaciones técnicas del módulo TP4056.	47
3.7	Especificaciones técnicas de pantalla OLED.	48
3.8	Especificaciones técnicas del RTC.	49
3.9	Especificaciones técnicas del lector de memorias SD.	50
3.10	Especificaciones técnicas del buzzer activo.	51
3.11	Pines utilizados del microcontrolador.	54
5.1	Resultados de medición NPK realizadas con el dispositivo.	73
5.2	Resultados de medición NPK del laboratorio.	74
5.3	Tabla comparativa de resultados de medición NPK - Muestra 6898.	74
5.4	Tabla comparativa de resultados de medición NPK - Muestra 6899.	74
5.5	Tabla comparativa de resultados de medición NPK - Muestra 6900.	74

Índice de códigos

3.1	Creación de tarea para conexión a WiFi e InfluxDB.	55
3.2	Función para conexión a WiFi e InfluxDB.	56
3.3	Inicialización de pantalla OLED.	56
3.4	Muestra de medición por OLED.	56
3.5	Inicialización del lector SD.	57
3.6	Configuración de RTC.	57
3.7	Trama Modbus RTU de requerimiento.	57
3.8	Trama Modbus RTU de respuesta.	58
3.9	Obtención de datos para el registro.	58
3.10	Declaración de cliente InfluxDB.	60
3.11	Validación con el servidor InfluxDB.	61
3.12	Consulta Flux para obtener los niveles de N, P y K.	62
3.13	Consulta Flux para obtener los niveles de N, P y K por hora.	64
3.14	Consulta Flux para obtener el promedio de N, P y K.	65
8.1	Función para almacenar el último registro subido a la nube.	83
8.2	Función para leer el último registro subido a la nube.	84
8.3	Función para subir los registros a InfluxDB.	85
8.4	Script para generar la planilla Excel.	88

1. Introducción

El **suelo** es la capa superficial de la tierra que ha sido transformada lentamente por la descomposición a través de la acción meteorológica, de la vegetación y del ser humano. El material original del cual un suelo se forma puede ser la roca subyacente, los depósitos de los ríos, de los mares y del viento o suelos de cenizas volcánicas. El suelo da soporte a las plantas en forma de una capa permeable para las raíces y es una especie de depósito para los nutrientes y el agua. Dependiendo de su composición, los suelos difieren en su capacidad para proveer los diferentes nutrientes [FAO, 2002].

El estudio de la fertilidad de suelos comprende el análisis de cada uno de los nutrientes esenciales para que toda planta o cultivo se desarrolle normalmente. La fertilización ha evolucionado con el paso del tiempo y ha cobrado una gran importancia en las prácticas agrícolas pues está estrechamente ligada con el incremento del rendimiento de los cultivos.

La **fertilidad del suelo** es la capacidad que tiene el mismo para suministrar los macro y micro nutrientes necesarios para el normal desarrollo del cultivo [Miranda et al., 1994]. Los principales nutrientes que las plantas necesitan proceden del aire y del suelo. Si el suelo está abundantemente provisto de estos nutrientes, los cultivos crecerán adecuadamente y producirán alto rendimientos. Sin embargo, la escasez de uno solo de los nutrientes necesarios puede limitar el crecimiento. Para obtener altos rendimientos, es esencial el uso de **fertilizantes** que suministren a las plantas los nutrientes que les faltan. No obstante, es crucial aplicar la cantidad adecuada en el momento oportuno, ya que un exceso de fertilizante puede resultar en bajos rendimientos y costos adicionales.

1.1. Contextualización del problema

La agricultura tiene una importancia estratégica en la economía argentina. La creciente demanda de productos agropecuarios, constituye una gran oportunidad para el desarrollo del país a través de la producción primaria de granos y el desarrollo de la agroindustria. En este contexto, Argentina tiene una gran responsabilidad en la futura seguridad alimentaria global, por ser uno de los países con mayor potencial para la producción de alimentos [Andrade et al., 2017].

Desde el punto de vista ecológico, la agricultura debe preservar la fertilidad y estructura del suelo, utilizar el agua para satisfacer las necesidades hídricas de manera sostenible, fortalecer las rotaciones agrícola-ganaderas y conservar la biodiversidad local y regional. Los principales efectos de la agricultura argentina sobre el ambiente incluyen: la degradación de los suelos, la contaminación con agroquímicos, la deforestación, la pérdida de biodiversidad y las emisiones de gases de efecto invernadero.

En los últimos años, la agricultura en Argentina se ha intensificado, lo que ha incrementado la producción, pero también ha deteriorado la calidad del suelo. La degradación del suelo se asocia con la intensidad de las labranzas, la duración de los ciclos agrícolas, la tendencia al monocultivo y la falta de reposición de nutrientes. La Figura 1.1 presenta un diagrama del impacto que ejerce la agricultura sobre los suelos, los factores que intervienen y los diferentes tipos de procesos de degradación de suelos resultantes.

La agricultura afecta el estado y las propiedades de los suelos, causando impactos negativos que difieren en su duración, nivel de severidad y grado de reversibilidad. Estos impactos se conocen como la degradación de suelos [Scherr, 1999]. La velocidad de degradación es influenciada por variables externas, como los cambios de uso de la tierra y la variabilidad climática (temporales, sequías y cambios importantes de la temperatura).

El suelo es modificado en su estado y propiedades por la agricultura a través de cambios en el ciclado, contenido y distribución de los compuestos de materia orgánica y nutrientes vegetales, en las comunidades microbianas, organismos del suelo, y en los procesos de almacenamiento, redistribución y consumo de agua. Además, el uso de agroquímicos y el tránsito de maquinaria agrícola son factores externos que también generan cambios en el suelo.

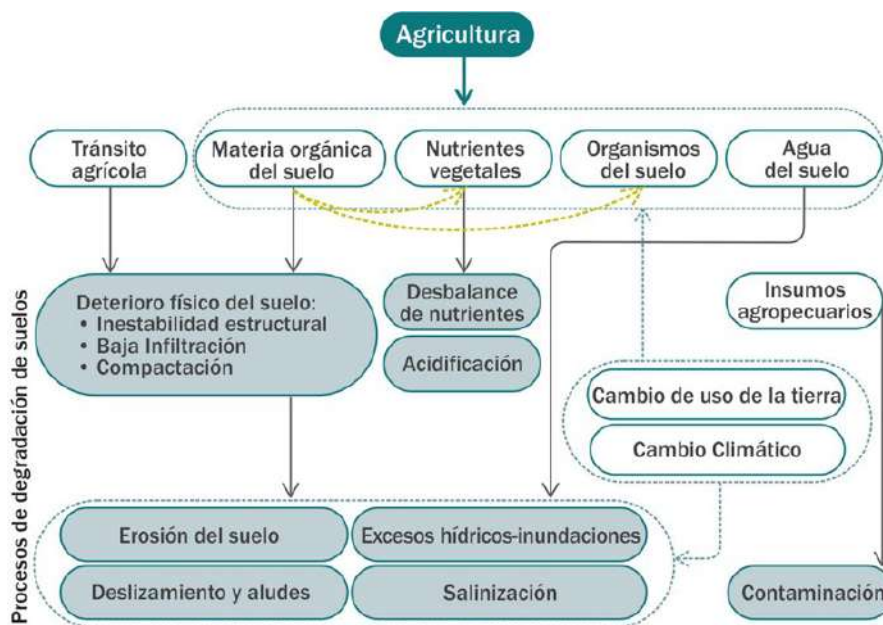


Figura 1.1: Impacto de la agricultura sobre el suelo. Extraído de [Andrade et al., 2017]

Según un informe de la [Bolsa de Comercio de Rosario, 2021], en el mundo se consumen 185 millones de toneladas de fertilizantes, siendo más requeridos los nitrogenados, fosforados y potásicos. En Argentina, el consumo es de 5,3 millones de toneladas, ubicándose en el puesto 24 a nivel mundial, con un promedio de aplicación de 40 kg/ha de fertilizante, cifra muy inferior al promedio mundial de 120 kg/ha. Esta escasa reposición de nutrientes se atribuye a varias causas, como las fluctuaciones en las políticas impositivas y de incentivos para el campo, la falta de información sobre métodos de diagnóstico de fertilidad, el limitado uso de análisis de suelos por parte de algunos productores y la creencia en la alta fertilidad natural de los suelos. Este escenario ha llevado a un grave desbalance de nutrientes en muchas regiones del país.

Por otro lado, el exceso de nutrientes aplicados por fertilización suele ocurrir en cultivos de muy alto valor, como los intensivos (frutas, verduras o flores en invernadero) y los semi-intensivos (papa, zanahoria, etc.). Estos suelos son sometidos a laboreos agresivos y frecuentes que afectan sus propiedades físicas, además de ser fertilizados con dosis que superan los requerimientos nutrimentales, lo que genera un impacto negativo en sus propiedades químicas y biológicas

Determinar la proporción de nutrientes presentes en el suelo es crucial para identificar cuáles deben ser aportados y en qué cantidades, asegurando así una fertilización adecuada y

eficiente. Por lo tanto, es indispensable contar con herramientas de análisis de suelo que faciliten el muestreo y la medición en campo, optimizando la gestión de nutrientes y promoviendo una agricultura sostenible.

1.2. Objetivo

En la sección anterior quedó en evidencia la importancia del análisis del suelo para proveer al mismo de las proporciones adecuadas de fertilizantes. El objetivo de este trabajo es desarrollar un dispositivo portátil, denominado *FertSens* (Figura 1.2), que permita obtener los valores de nitrógeno, potasio y fósforo del suelo de manera eficiente y precisa.

El dispositivo contará con una unidad microcontrolada (MCU), un módulo de recarga y gestión de energía, una interfaz de usuario y un sensor para el análisis del suelo. Las mediciones obtenidas serán visualizadas en una pantalla pequeña, almacenadas en una tarjeta de memoria y, con conexión Wi-Fi, subidas a una base de datos (InfluxDB) para su visualización en un tablero de control (Grafana). Además, se permitirá la exportación de los datos a una hoja de cálculo Excel mediante un script desarrollado en Python, facilitando así la gestión y el análisis de la información para una mejor toma de decisiones en la fertilización del suelo.



Figura 1.2: Logo del dispositivo.

1.3. Estructura del Trabajo Final de Grado

Este trabajo se estructura en seis capítulos:

- El primer capítulo presenta la introducción a esta tesis y describe: 1) la contextualización de la problemática a abordar, 2) los objetivos que se estipularon alcanzar y 3) la estructura del trabajo.

- El segundo capítulo introduce al lector hacia los fundamentos teóricos esenciales para la comprensión del trabajo y que constituyeron el estudio que se tuvo que abordar para realizarlo.
- El tercer capítulo detalla el diseño e implementación del dispositivo propuesto, haciendo mención al desarrollo tecnológico realizado.
- El cuarto capítulo presenta la construcción del dispositivo, desde el diseño de la carcasa y la placa electrónica hasta el instrumento de medición final obtenido.
- El quinto capítulo presenta el caso de estudio realizado.
- Por último, el capítulo seis expone las conclusiones derivadas de la realización de este trabajo.

2. Fundamentos teóricos

El presente capítulo introduce el marco teórico necesario para el desarrollo de esta tesis, a saber:

- tipos de nutrientes para el crecimiento de cultivos,
- los fertilizantes y su utilización,
- formas de analizar el suelo, incluyendo número de muestras, tipos y frecuencia de muestreo,
- la arquitectura de la internet de las cosas y su relación con la agricultura,
- protocolos e interfaces de comunicación a utilizar por el dispositivo,
- tipos de reguladores de voltaje y,
- tecnologías de memoria.

2.1. Nutrientes para el crecimiento del cultivo

Los elementos esenciales para el crecimiento de una gran mayoría de plantas provienen del aire y del suelo circundante. En el suelo, el medio de transporte de los nutrientes es la *solución del suelo*¹. Los elementos que se encuentran son provenientes **del aire** (carbono (C)), **del agua** (hidrógeno (H) y oxígeno (O)), **del suelo, del fertilizante y del abono animal** (nitrógeno (N), fósforo (P), potasio (K), calcio (Ca), magnesio (Mg), azufre (S), hierro (Fe), manganeso (Mn), zinc (Zn), cobre (Cu), boro (B), molibdeno (Mo) y cloro (Cl)).

Estos nutrientes y sus porcentuales promedio en la sustancia seca de la planta son mostrados en la Figura 2.1.

¹Solución compuesta por agua y por todos los elementos minerales que son necesarios para el crecimiento de las plantas.

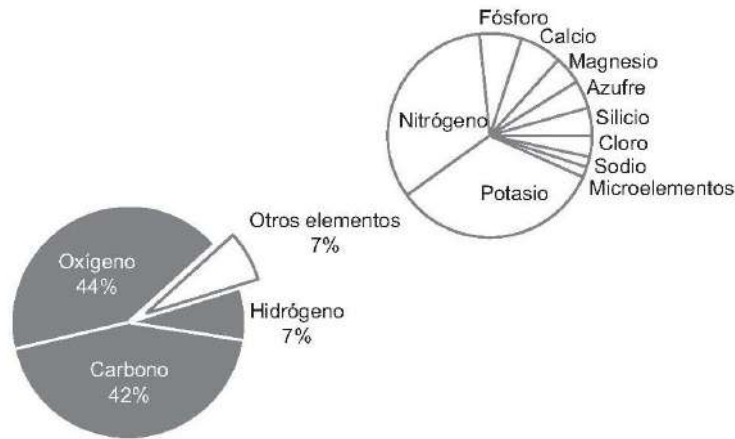


Figura 2.1: Porcentaje de nutrientes en las plantas. Extraído de [FAO, 2002]

La planta absorbe todos los nutrientes de la solución del suelo, los cuales se dividen en dos categorías:

- a) **macronutrientes**, divididos en nutrientes *primarios* y *secundarios*.
- b) **micronutrientes** o microelementos.

Los **macronutrientes** se necesitan en grandes cantidades si el suelo es deficiente en uno o más de ellos. Los suelos son naturalmente pobres en nutrientes debido a la extracción de los mismos por los cultivos a lo largo de los años. Dentro del grupo de los macronutrientes, entre los *nutrientes primarios* se encuentra el nitrógeno, fósforo y potasio mientras que los *nutrientes secundarios* son magnesio, azufre y calcio. En contraste a los macronutrientes, los **micronutrientes** son requeridos en cantidades ínfimas para el crecimiento correcto de las plantas y tienen que ser agregados en cantidades muy pequeñas cuando no son provistos por el suelo.

Un suministro suficiente de nutrientes es importante para un funcionamiento correcto de la *fotosíntesis*². Esto se debe al hecho de que si uno de los nutrientes del suelo no está presente, la fotosíntesis se retrasa. Si está presente, pero en cantidad insuficiente, la planta desarrolla síntomas de deficiencia [FAO, 2002]. El crecimiento de una planta depende de un suministro suficiente de cada nutriente y el rendimiento está limitado por los nutrientes que son restringidos (Figura 2.2). En las prácticas agrícolas este es el caso para el nitrógeno, fósforo, potasio,

²Proceso químico que consiste en la conversión de materia inorgánica a orgánica gracias a la energía solar.

magnesio y azufre, de allí que estos nutrientes tienen que ser aplicados en forma de fertilizantes para obtener rendimientos satisfactorios.



Figura 2.2: Rendimiento limitado por los nutrientes. Extraído de [FAO, 2002]

2.1.1. Nitrógeno (N)

El nitrógeno (N) es uno de los elementos más ampliamente distribuidos en la naturaleza, siendo el motor del crecimiento de la planta ya que supone del 1% al 4% del extracto seco de la misma [FAO, 2002]. El principal reservorio de N es la atmósfera y en el suelo se encuentra bajo tres formas:

- **Nitratos (NH_3^-):** es una forma de N asimilable o disponible por las raíces de las plantas.
- **Amoniacal (NH_4^+):** es una forma de N de transición y no abunda en el suelo.
- **Orgánicas:** se encuentra en la materia orgánica y es la única reserva de N en el suelo.

Por sus funciones en la fisiología de las plantas es un elemento esencial para el crecimiento y desarrollo de los cultivos. Los altos requerimientos de N por parte de los mismos lo convierten en un factor limitante en todos los suelos del mundo [Quiroga et al., 2012]. En la planta se combina con componentes producidos por el metabolismo de carbohidratos para formar aminoácidos y proteínas, siendo el constituyente esencial de estas últimas. Un buen suministro de N es importante ya que está involucrado en todos los procesos principales de desarrollo, en

el rendimiento y en la absorción de otros nutrientes.

Debido al avance de la agricultura y los procesos de degradación, se acentúa la deficiencia de N dado que los primeros centímetros son los más fértiles. El N orgánico y los nitratos son las formas más importantes a tener en cuenta en la productividad de los cultivos.

2.1.2. Fósforo (P)

El fósforo (P), después del nitrógeno, es el nutriente que más frecuentemente afecta la producción de los cultivos. El P forma parte de enzimas, ácidos nucleicos y proteínas, supone del 0,1 % al 0,4 % del extracto seco de la planta y está involucrado en prácticamente todos los procesos de transferencia de energía [FAO, 2002]. El contenido de P en el suelo está definido por el material madre y se observa un marcado efecto del clima, siendo las zonas húmedas las más deficientes en este nutriente [Tisdale et al., 1993].

El P es uno de los nutrientes considerados esenciales para el crecimiento y desarrollo ya que las plantas lo toman de la solución del suelo. Normalmente esta solución es demasiado pobre para sostener un cultivo y debe ser realimentada continuamente de las formas más insolubles de P. La mayor parte de los suelos agrícolas son demasiado pobres como para sostener este proceso y precisan de la fertilización.

El fósforo es necesario para la respiración, fotosíntesis, funcionamiento celular y transferencia y reproducción de genes. El P se absorbe principalmente por las raíces desde la solución del suelo como iones fosfato (en suelos ácidos se absorbe preferentemente como H_2PO_4^- y en suelos básicos como HPO_4^{2-}). Las plantas en crecimiento no almacenan dichos iones, sino que exigen una abundante provisión desde el suelo. Si no se obtiene de manera suficiente el P necesario, sufren importantes retardos en su crecimiento [Quiroga et al., 2012].

La cantidad de P total en los suelos es muy superior a lo que necesitan los cultivos, pero la baja solubilidad del mismo y los sitios de absorción que compiten por P con las plantas, generan una baja disponibilidad. Cuando los suelos son cultivados, esta baja disponibilidad se traduce en una deficiencia de P que tendrá que ser compensada con una fertilización.

2.1.3. Potasio (K)

Junto con el N y el P, el potasio es el tercer elemento de mayor importancia en las plantas. El mismo, que suple del 1 % al 4 % del extracto seco de la planta, es absorbido en su forma catiónica K^+ y juega un papel esencial al ser el principal activador de numerosas enzimas [FAO, 2002]. Además tiene un papel clave en la osmorregulación celular de la planta, interviniendo en la apertura y cierre de los *estomas*³. Por esta función, el potasio aumenta la resistencia a la sequía y al frío.

El K se encuentra en la fase sólida del suelo y en equilibrio dinámico con la fase líquida, es decir, a medida que las plantas absorben potasio del agua del suelo, más potasio es liberado desde las reservas en la fase sólida. La mayor parte del K absorbido por los cultivos retorna a la superficie del suelo, para estar disponible para los próximos cultivos [Cubilla et al., 2012]. El potasio incide sobre la correcta formación del fruto, maduración y coloración, mejora de la producción en fruta y tubérculos, crecimiento radicular y resistencia a la sequía y heladas.

El K es requerido en grandes cantidades (similar al N) pero no es absorbido fácilmente. Al igual que el N y el P, el K es un elemento muy móvil y los síntomas de su carencia se manifiestan en las hojas más viejas donde se caracteriza la aparición de *necrosis*⁴ en las puntas y bordes de las hojas. En general presentan un menor tamaño y dado su importante papel en la osmorregulación, pueden sufrir mayor susceptibilidad al estrés hídrico y al ataque por patógenos. Cabe mencionar que, en condiciones de exceso de K, su consumo se incrementa y puede interferir en la captación de calcio (Ca) y magnesio (Mg) provocando de forma indirecta una deficiencia de ambos nutrientes.

2.1.4. Magnesio (Mg)

El Magnesio (Mg) es el constituyente central de la clorofila, el pigmento verde de las hojas que funciona como un aceptador de la energía provista por el sol; por ello, del 15 % al 20 % del magnesio contenido en la planta se encuentra en las partes verdes. El Mg se incluye también en las reacciones enzimáticas relacionadas a la transferencia de energía de la planta [FAO, 2002].

³Poros o aperturas regulables que se encuentran situados en el tejido epidérmico de las plantas.

⁴Decoloración amarillado que indica la muerte del tejido. Es irreversible, no puede ser curada con la aplicación de nutrientes.

2.1.5. Azufre (S)

El Azufre (S) es un constituyente esencial de proteínas y está involucrado en la formación de la clorofila. En la mayoría de las plantas supone del 0,2 % al 0,3 % del extracto seco. Por dicho motivo, es tan importante en el crecimiento de la planta como el fósforo y el magnesio; pero su función es, generalmente, subestimada [FAO, 2002].

2.1.6. Calcio (Ca)

El Calcio (Ca) es esencial para el crecimiento de las raíces y constituye el tejido celular de las membranas. Aunque la mayoría de los suelos contienen suficiente disponibilidad de Ca para las plantas, la deficiencia puede darse en los suelos tropicales muy pobres en este nutriente. Sin embargo, el objetivo de la aplicación de Ca es usualmente el del encalado, es decir, reducir la acidez del suelo [FAO, 2002].

2.1.7. Micronutrientes

Los micronutrientes o microelementos son el hierro (Fe), el manganeso (Mn), el zinc (Zn), el cobre (Cu), el molibdeno (Mo), el cloro (Cl) y el boro (B). Ellos son parte de sustancias claves en el crecimiento de la planta, siendo comparables con las vitaminas en la nutrición humana. Son absorbidos en cantidades minúsculas, el rango de provisión óptima es muy pequeño y su disponibilidad en las plantas depende principalmente de la reacción del suelo.

Algunos microelementos pueden ser tóxicos para las plantas a niveles algo más elevados que lo normal. En la mayoría de los casos esto ocurre cuando el pH es de bajo a muy bajo. Es importante notar que todos los nutrientes, ya sean necesarios en pequeñas o grandes cantidades, cumplen una función específica en el crecimiento de la planta, en la producción alimentaria y que un nutriente no puede ser sustituido por otro [FAO, 2002].

2.2. Potencial de Hidrógeno (pH)

El potencial de hidrógeno (pH) es la medida del grado de acidez o alcalinidad de un suelo, es decir, la concentración de hidrogeniones (H^+) que existen en el suelo. En la escala de valor

máximo 14, el valor de un suelo neutro es 7, siendo ácidos todos aquellos que tengan valores inferiores a 7 y básicos todos aquellos que tienen valores superiores a éste. Las plantas cultivadas presentan su mejor desarrollo en valores cercanos a la neutralidad ya que en estas condiciones los elementos nutritivos están más fácilmente disponibles.

Si el suelo es excesivamente ácido impide que otros elementos necesarios tales como el calcio, magnesio, sodio o potasio permanezcan en él, pasando a ser solubles y siendo fácilmente eliminados con el agua de lluvia o de riego. Si el suelo tiene un valor de pH menor de 5,5 es conveniente, en general, elevarlo hasta un valor cercano a 6/6,5, para que los elementos nutritivos puedan estar más fácilmente disponibles para las plantas.

Si el suelo es básico, el exceso de calcio en el medio impide que otros elementos, tales como el hierro, puedan ser absorbidos por las plantas. El pH de un suelo puede disminuirse aplicando azufre (S) y, si se requiere hacerlo con rapidez, se puede añadir ácido sulfúrico o nítrico, dependiendo de las condiciones del suelo [Garrido, S., 1993].

La medida del pH se puede realizar mezclando la muestra con agua o con cloruro potásico a diferentes concentraciones. Los suelos en zonas agrícolas tienen normalmente valores de pH entre 4,5 y 9,5 (medidos en disolución en agua en proporción 1:2) como se muestra en la Tabla 2.1.

Tabla 2.1: Tipos de suelo según el valor de pH. Extraído de [Garrido, S., 1993]

pH (medido en agua, en disolución 1:2)	Tipo	Observaciones
Menor de 5,5	Muy ácido	Dificultad de desarrollo en la mayoría de los cultivos y retención de nutrientes.
5,5 - 6,5	Ácido	
6,5 - 7,5	Neutro o cercano a la neutralidad	Intervalo óptimo para los cultivos.
7,5 - 8,5	Básico	
Mayor de 8,5	Muy básico	Dificultad de desarrollo en la mayoría de los cultivos.

Los factores que influyen en el valor del pH son diversos, por ejemplo, la piedra que dio origen a dicho suelo, si la misma es de reacción ácida o alcalina, las precipitaciones que tienden a acidificar el suelo o el uso de fertilizantes, los cuales pueden dejar residuos ácidos o alcalinos.

2.3. Fertilizantes

Un **fertilizante** es cualquier material natural o industrializado que contenga al menos 5 % de uno o más de los tres nutrientes primarios (N, P_2O_5 , K_2O). Los términos P_2O_5 y K_2O son convencionalmente usados para expresar los nutrientes de los fertilizantes fósforo y potasio (son los óxidos de dichos elementos).

Los fertilizantes fabricados industrialmente son llamados fertilizantes minerales y la presentación de éstos es muy variada, dependiendo del proceso de fabricación, las partículas son de muy diferentes tamaños y formas: granos, pastillas, cristales, polvo de grano grueso/compactado o fino. La mayoría de los fertilizantes es provista en forma sólida [FAO, 2002].

En adelante se utilizará el acrónimo *NPK* para indicar la relación entre los elementos químicos nitrógeno, fósforo y potasio.

2.3.1. Utilización de fertilizantes

Para determinar la necesidad de aplicar fertilizantes a los cultivos y suelos se deben saber dos interrogantes:

- ¿Qué nutrientes son necesarios en el fertilizante?
- ¿Cuánto de cada nutriente se necesita para lograr el rendimiento óptimo?

Según [FAO, 2002] las técnicas para responder a dichas preguntas se enumeran a continuación:

1. Signos de deficiencia de nutrientes en cultivos en desarrollo.
2. Análisis de suelos para determinar los nutrientes necesarios para los fertilizantes y las cantidades que se necesitan.
3. Análisis de la planta y/o de su tejido.
4. Ensayos de validación de fertilizantes en el campo.

2.4. Análisis del suelo

Al momento de realizar un análisis de suelo, una parte esencial es la elección de muestras representativas para realizar un diagnóstico adecuado. En muchas ocasiones es difícil elegir los puntos de muestreo, por lo que esta fase requiere atención [Garrido, S., 1993]. Las actividades a realizar son las siguientes:

1. Observación del suelo en toda su extensión y profundidad.
2. Definición de los puntos de muestreo.
3. Muestreo y toma de datos en el campo.
4. Decisión sobre los análisis a realizar.
5. Preparación de las muestras para su envío al laboratorio.
6. Análisis de las muestras en el laboratorio.
7. Interpretación de los análisis.
8. Decisiones para el buen manejo del suelo.

Al tomar correctamente la muestra de suelo se evitan los errores asociados a la variabilidad del lote. La etapa del muestreo es una fuente de error importante en relación con los resultados obtenidos, por lo tanto, una correcta técnica de muestreo y un posterior análisis permite ajustar la fertilización a las condiciones específicas del lote, minimizando los riesgos de pérdida de nutrientes. La confiabilidad de los resultados obtenidos, sin perjuicio de la idoneidad del laboratorio o del dispositivo planteado en esta tesis para el análisis del suelo, depende en gran medida de que la muestra sea representativa.

2.4.1. Número de muestras y submuestras

Muestra simple: es la que se obtiene con una sola extracción de suelo y, por lo general, es usada cuando se trata de suelos homogéneos. También resultan de utilidad cuando se trabaja con sistemas de agricultura de precisión donde es necesario realizar un muestreo sistemático e intensivo.

Muestra compuesta: es la técnica habitual para evaluar la disponibilidad de nutrientes promedio de un lote. Se obtiene al extraer en cada área homogénea varias muestras simples

(submuestras), tomadas a la misma profundidad y con un volumen de suelo semejante. Posteriormente son llevadas a un recipiente para mezclarlas y extraer una muestra de 0,5 a 1 kg de suelo, que es la que se enviará al laboratorio para su análisis. Cuanto mayor sea la cantidad de submuestras, el muestreo es más representativo y, por extensión, los resultados obtenidos en el laboratorio son más próximos a la realidad [Red BPA, 2020].

Cultivos extensivos: la superficie a relevar por cada muestra compuesta abarca entre 20 y 50 ha (siempre que la homogeneidad del área lo permita), con una toma de submuestra cada 2-3 ha.

Cultivos intensivos (quintas, frutales, huertas, etc.): la superficie a relevar por cada muestra compuesta es de hasta 1 ha o 1 parcela, tomando entre 10 a 30 submuestras por ha o parcela.

2.4.2. Forma de muestreo

Muestreo sistemático espacial (en superficie): es el método más apropiado si el objetivo es producir mapas para la aplicación variable de fertilizantes. Con este método, las muestras simples son tomadas a intervalos regulares en todas las direcciones, analizándose por separado. En este caso es necesario definir previamente los puntos de muestreo, georreferenciarlos y luego identificar las muestras con precisión [Red BPA, 2020]. En la Figura 2.3 se muestran las diferentes formas de este tipo de muestreo.

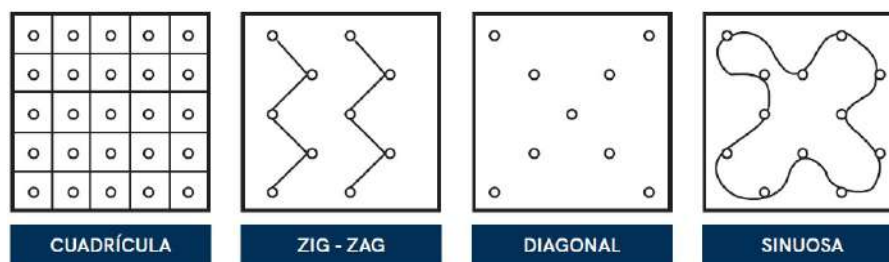


Figura 2.3: Formas de muestreo sistemático. Extraído de [Red BPA, 2020]

Muestreo no sistemático o al azar espacial (en superficie): consiste en recorrer un lote recolectando submuestras al azar, como se ilustra en la Figura 2.4, que luego son mezcladas para formar la muestra compuesta.

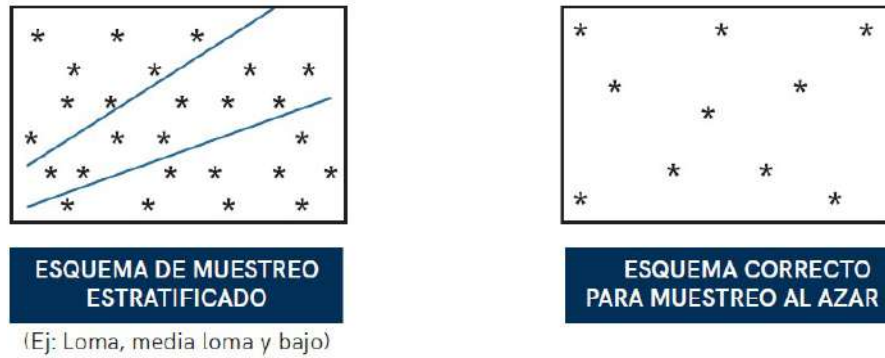


Figura 2.4: Formas de muestreo al azar. Extraído de [Red BPA, 2020]

Si lo que se busca es evaluar la salud actual de un suelo en producción o su estado de conservación a través de sus parámetros de fertilidad química, será necesario recurrir a muestras de referencia, contra las que se realizará la comparación. Para ello se obtienen muestras de suelo con el menor grado de disturbio posible, tales como sectores no trabajados cercanos a los lotes, sitios dentro del monte o debajo de los alambrados, que se utilizarán para compararlos con los suelos en producción.

Muestreo georreferenciado: si se quiere saber cómo evoluciona de un año al otro algún parámetro de suelo o el contenido de un nutriente, es recomendable sacar una muestra compuesta, siempre en el mismo lugar. Este muestreo se utiliza cuando el objetivo es evaluar los efectos de distintas rotaciones de cultivos o distintas estrategias de manejo sobre los parámetros de calidad del suelo.

2.4.3. Frecuencia y momento de muestreo

Al evaluar la evolución de las propiedades del suelo se considera la variabilidad temporal de los distintos parámetros. Por ejemplo, para determinaciones de materia orgánica del suelo, pH, calcio, magnesio, potasio o sodio, la frecuencia de control recomendable es cada 3 a 5 años, independientemente de la época del año y del cultivo presente al momento del muestreo.

Otros parámetros con mayor variabilidad temporal, en cambio, requieren de un control más frecuente. Así, dada la movilidad del nitrógeno, el muestreo para nitratos es conveniente realizarlo todos los años o antes de fertilizar un cultivo anual. Cuando se trabaja con cultivos

intensivos (horticultura, jardinería, invernáculos, etc.) la frecuencia del análisis debe ser más intensa y, por ello, es aconsejable realizar el muestreo anualmente.

Respecto al momento de muestreo para la mayoría de las determinaciones (pH, materia orgánica, fósforo extractable, sulfatos, calcio, magnesio, potasio o sodio) el muestreo puede realizarse en cualquier época del año o con una anticipación de 30 días a la siembra, de modo de permitir una adecuada planificación del uso de enmiendas o del fertilizante de base. Sin embargo, para facilitar la comparación con información anterior, es conveniente tomar las muestras en la misma época del año en que se tomaron durante los controles previos.

Para evaluar el contenido de nitrógeno en forma de nitratos, la muestra es tomada poco antes de la siembra o cercana al momento de mayor demanda de nitrógeno por parte del cultivo, permitiendo ajustar la aplicación del fertilizante en el momento adecuado. En cualquiera de los casos, si fuera necesario muestrear después de lluvia o riego intenso (cuando la lámina de agua supera los 20 mm), es recomendable dejar pasar 48hs [Red BPA, 2020].

2.4.4. Profundidad de muestreo

La profundidad de muestreo depende del objetivo del análisis, del parámetro que se desea evaluar, del tipo de labranza y de la profundidad de exploración que alcanzan las raíces. Para la mayoría de las determinaciones de parámetros de menor variabilidad y movilidad en el tiempo y/o en el espacio, tales como materia orgánica, fosforo extractable o pH, la profundidad recomendada es de 0-20 cm.

En el caso de diagnósticos de fertilización nitrogenada, dado la movilidad de los nitratos en el suelo, se deben tomar muestras hasta los 60 cm de profundidad, extrayendo tres estratos de 20 cm cada uno (0-20, 20-40 y 40-60 cm) o la profundidad considerada acorde al método de diagnóstico utilizado. Para estimar el dato de agua disponible, el muestreo se hace cada 30 cm, hasta llegar a 1,5 m de profundidad.

2.5. Internet of Things (IoT)

La IoT (Internet of Things - internet de las cosas) describe la red de objetos físicos (*cosas*) que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e inter-

cambiar datos con otros dispositivos y sistemas a través de Internet. Estos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas.

La idea de IoT existe desde hace mucho tiempo, sin embargo su implementación práctica solo fue posible luego de una serie de avances:

- *Acceso a la tecnología de sensores de bajo costo y potencia:* los sensores asequibles y fiables hacen posible la tecnología IoT para más fabricantes.
- *Nuevas tecnologías de bajo consumo:* las tecnologías de bajo consumo, como Bluetooth Low Energy (BLE) y Zigbee, han permitido que los dispositivos IoT funcionen con una mínima demanda de energía.
- *Integración de chips todo en uno:* la miniaturización e integración de componentes han permitido la creación de chips que contienen una variedad de funcionalidades, incluidas las capacidades de comunicación como Wi-Fi y Bluetooth. Estos chips simplifican el diseño y la fabricación de dispositivos IoT, reduciendo los costos y el tiempo de desarrollo.
- *Desarrollo de nuevas tecnologías de comunicación:* como LoRa (Long Range) que es una tecnología de comunicación inalámbrica de largo alcance que permite la transmisión de datos a largas distancias con un consumo de energía bajo [LoRa, 2024], NB-IoT (Narrow-band IoT) que es una tecnología de comunicación celular de baja potencia con una amplia cobertura de red [NB-IoT, 2024] o Sigfox que es una red de área amplia de baja potencia (LPWAN) que proporciona conectividad a largas distancias [Sigfox, 2024].
- *Conectividad:* una gran cantidad de protocolos de red facilita la conexión de los dispositivos a la nube para lograr una transferencia de datos eficiente.
- *Plataformas de informática en la nube:* el aumento en la disponibilidad de plataformas en la nube permite acceder a la infraestructura que se necesita para escalar sin tener que administrarlo todo.

2.5.1. Arquitectura IoT

La arquitectura IoT es un proceso en el que la información viaja a través de la red, en forma de datos digitalizados, desde los sensores a un centro de datos o a la nube, donde serán procesados y almacenados [Unir IoT, 2023]. Esta arquitectura está compuesta por cuatro fases:

- 1. Sensores y actuadores:** los *sensores* son responsables de capturar datos del mundo real, como temperatura, humedad, composición química, etc. Por otro lado, los *actuadores* transforman energía en una acción determinada como encendido de motores o accionamiento de relés e interruptores.
- 2. Sistema de adquisición de datos y puntos de acceso:** los sistemas de adquisición de datos son los encargados de recopilar la información analógica obtenida de los sensores y transformarla en datos digitales que se envían a Internet. El objetivo es establecer una conexión entre los periféricos (dispositivos u objetos conectados) y la nube, pero también permitir conectividad entre ellos. Esta conexión tiene que ser segura, robusta y tolerante a fallos. A través de los gateway o interfaces de comunicación, los dispositivos estarán conectados entre ellos y con la nube. Existen multitud de protocolos y tecnologías de comunicación, por ejemplo, ZigBee, WiFi, MQTT o Bluetooth.
- 3. Procesamiento de datos:** *el eje central de IoT son los datos*. Un sistema IoT debe ser capaz de recolectar información de los sensores, almacenarlos y analizarlos para obtener datos útiles en conjuntos de datos masivos para enviar a la nube.
- 4. Análisis en el centro de datos o en la nube:** la gestión y el almacenamiento de los datos es la última fase de la arquitectura y exige equipos que permitan alojar la información recabada por los diferentes sensores y ponerla a disposición de los interesados para poder visualizarla y, en ocasiones, modificar parámetros para que los sistemas se comporten de una manera determinada.

La arquitectura IoT tiene múltiples aplicaciones, en cualquier sector, independientemente de cuál sea su actividad:

- *Industria:* monitorización de la producción en tiempo real y robótica colaborativa.
- *Transporte y logística:* almacenes inteligentes con gestión y monitorización de flota en tiempo real.
- *Construcción:* sensores para monitorizar obras, detectar anomalías y predecir la vida útil de la misma.
- *Diseño de interiores:* instalaciones domóticas y mobiliario inteligente.

- *Agricultura y ganadería*: sistemas de riego inteligentes, drones, dispositivos para la monitorización de cultivos y *wearables*⁵ geoposicionados para realizar un seguimiento del ganado.
- *Sanidad*: dispositivos para la monitorización remota de signos vitales y geolocalización para seguimiento de pacientes con dependencia.

2.5.2. IoT y la agricultura

Los sistemas de monitoreo IoT en la agricultura permiten el mejoramiento de los cultivos aplicando técnicas para beneficiar su rendimiento, la optimización del uso de recursos, minimización del impacto ambiental y toma de decisiones adecuadas.

En este trabajo se propone el desarrollo de un sistema IoT para la monitorización de parámetros químicos (nitrógeno, fósforo y potasio), el cual está enmarcado dentro de la arquitectura de cuatro fases de IoT. En la *fase 1 (sensores y actuadores)* mediante el uso del sensor NPK, descrito en 3.2.3, se obtienen los parámetros del suelo. Es el mismo sensor el que participa de la *fase 2 (sistema de adquisición de datos y puntos de acceso)* ya que se encarga de transformar la información analógica en digital. Luego el microcontrolador es el encargado de la *fase 3 (procesamiento de datos)*, donde se almacena en memoria persistente dicha información para enviarla, cuando sea posible, a través de WiFi a la nube. De esta forma se finaliza con la *fase 4 (análisis en el centro de datos o en la nube)* donde es InfluxDB (base de datos de series temporales) y Grafana (visualizador gráfico de datos), los cuales serán explicados en secciones posteriores, encargados de almacenar los datos y presentar una visualización gráfica de los mismos.

2.6. Arquitectura de protocolos

Una *arquitectura de protocolos* es una estructura en capas de elementos hardware y software que facilita el intercambio de datos entre sistemas, posibilitando aplicaciones como el comercio electrónico, la telemedicina, el trabajo colaborativo en línea, las videoconferencias, las redes sociales, el Internet de las Cosas (IoT) y los servicios de transmisión en tiempo real. En resumen,

⁵Dispositivos electrónicos inteligentes incorporados corporalmente.

esta estructura permite el intercambio de cualquier tipo de datos, tanto en tiempo real como diferido, entre dos o más terminales.

En los sistemas de comunicación, en cada una de las capas de esta arquitectura, se implementa uno o más protocolos comunes donde cada uno de ellos proporciona un conjunto de reglas para el intercambio de datos entre sistemas. La arquitectura de protocolos más utilizada es TCP/IP, la cual está constituida por cinco capas (física, acceso a la red, internet, transporte y aplicación).

Otra arquitectura de protocolos importante es el modelo de referencia para la interconexión de sistemas abiertos OSI (Open Systems Interconnection), ilustrado en la Figura 2.5, el cual es una arquitectura normalizada de siete capas que se utiliza para describir las funciones de un sistema de comunicación. Cada capa realiza un subconjunto de tareas necesarias y relacionadas entre sí para llegar a comunicarse con otros sistemas. Por otra parte, cada capa se sustenta de la capa inmediatamente inferior, la cual realizará funciones más primitivas, ocultando los detalles a las capas superiores [Stallings, 2004]. A continuación se explican las funciones que realiza cada capa del modelo OSI:

1. **Física:** se encarga de la transmisión de cadenas de bits no estructuradas sobre el medio físico.
2. **Enlace de datos:** proporciona un servicio de transferencia de datos fiable a través del enlace físico y envía bloques de datos (tramas) llevando a cabo la sincronización, el control de tareas y el flujo.
3. **Red:** proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y transmisión utilizadas para conectar los sistemas, además es responsable del establecimiento, mantenimiento y cierre de las conexiones.
4. **Transporte:** proporciona una transferencia transparente y fiable de datos entre los puntos finales, además brinda procedimientos de recuperación de errores y control de flujo origen-destino.
5. **Sesión:** proporciona el control de la comunicación entre las aplicaciones, establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadoras.
6. **Presentación:** proporciona independencia a los procesos de aplicación respecto a las diferencias en la representación de datos (sintaxis).

7. Aplicación: proporciona el acceso al entorno OSI tanto a los usuarios como a los servicios de información distribuida.

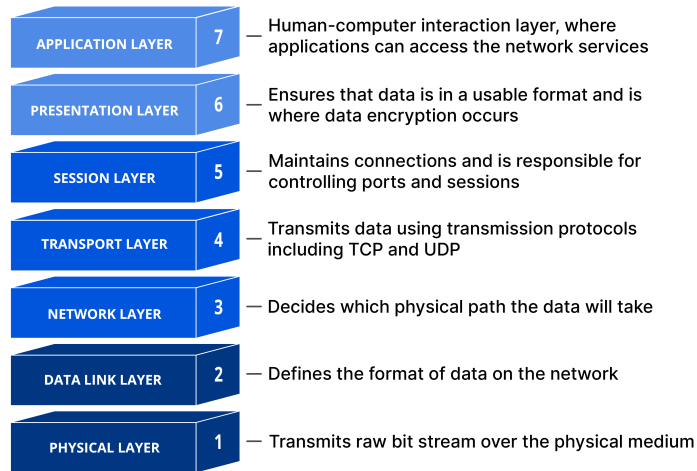


Figura 2.5: Capas del modelo OSI. Extraído de [Cloudflare, 2024]

2.7. Interfaces de comunicación

Una interfaz es un conjunto de reglas que permiten la comunicación entre diferentes dispositivos. En esta clasificación se incluye aquellas interfaces que solo especifican los medios físicos y eléctricos que los dispositivos deben usar para comunicarse ya que se utilizan para transmitir pequeños paquetes de datos (8, 16 o 32 bits). Se utilizan para interconectar un microcontrolador con dispositivos como memorias, conversores analógico-digital (AD) y digital-analógico (DA), displays, etc., por lo que solo requieren la especificación de la capa física.

2.7.1. I2C

La interfaz I2C (circuito inter-integrado) utiliza un enfoque bidireccional de dos hilos para intercambiar datos entre dispositivos ubicados en el mismo bus y para controlar otros dispositivos [Buyya et al., 2016]:

- *SCL*: Serial Clock o línea de reloj en serie.
- *SDA*: Serial Data o puerto de aceptación de línea de datos.

El maestro envía a cada esclavo la dirección del esclavo con el que desea iniciar la comunicación y un bit de lectura/escritura. El esclavo compara la dirección con la suya y si coincide, la línea SDA pasa del estado alto a bajo, mientras que SCL está en alto (Start Condition en Figura 2.6).

Si la dirección no coincide con su dirección, el esclavo deja la línea SDA alta. A continuación, el maestro enviará o recibirá la trama de datos. Al acabar la transferencia de datos, el maestro envía una señal de parada a los dispositivos esclavos indicándoles que deben volver a un estado inactivo, donde la línea SDA pasa del estado bajo al alto, mientras que SCL es alto (Stop Condition en Figura 2.6).

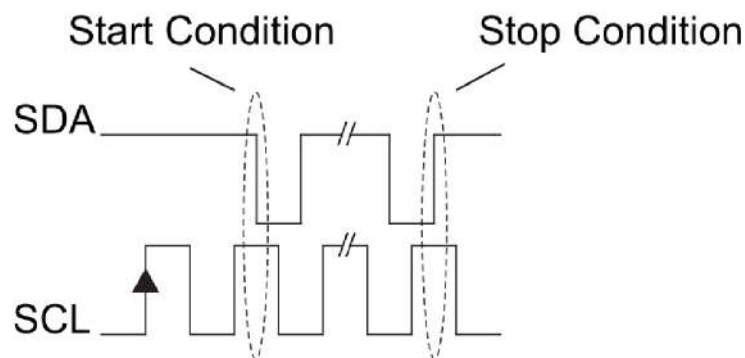


Figura 2.6: Inicio y parada - I2C. Extraído de [Afzal, S., 2016]

2.7.2. SPI

SPI (interfaz periférica en serie) se utiliza para implementar la transmisión de datos a alta velocidad y puede enviar y recibir un flujo continuo de datos sin interrupciones [Buyya et al., 2016]. SPI utiliza cuatro líneas:

- *MOSI* (Master Output Slave In - salida del maestro entrada del esclavo) es la línea a través de la cual el maestro envía datos a sus esclavos.
- *MISO* (Master In Slave Out - entrada del maestro salida del esclavo) es la línea a través de la cual los esclavos responden al maestro.
- *SCK* (Serial Clock) es el reloj proporcionado por el dispositivo maestro.
- *CS* (Chip Select) es la línea utilizada para seleccionar el esclavo con el que el maestro desea comunicarse.

Por lo general es una interfaz más rápida que I2C debido a su simplicidad y existen dos modos de comunicarse a través de SPI. En un *primer modo*, si se utiliza un único dispositivo controlador para activar un único dispositivo de bajada, se trata simplemente de una topología punto a punto. Se selecciona cada dispositivo con una línea CS por lo que se necesita una línea de selección de chip separada para cada dispositivo. La activación de varios dispositivos depende del número de salidas de selección de chip que proporcione el controlador.

Se llama daisy chain o cadena margarita a un esquema de cableado tal que un dispositivo A es conectado a un dispositivo B, el mismo dispositivo B a un dispositivo C, este dispositivo C a un dispositivo D, y así sucesivamente. El *segundo modo* utiliza este tipo de esquema, en la que una única salida de selección de dispositivo activa sucesivamente cada uno de los dispositivos de la propia cadena.

No hay límite para la cantidad de dispositivos SPI que se pueden conectar. Sin embargo, existen límites prácticos debido al número de líneas de selección de hardware disponibles en el dispositivo principal (el máster).

2.8. Protocolos de comunicación

Los protocolos de comunicación son estándares formales constituidos por procedimientos y restricciones que permiten el intercambio de información entre dispositivos. La *comunicación paralela* es un método para transmitir múltiples dígitos binarios (bits) de manera simultánea mientras que la *comunicación serie* permite la transmisión y recepción bit a bit de un byte completo. La principal ventaja de este tipo de comunicación es que necesita un número más pequeño de líneas de transmisión que una comunicación paralela para transmitir la misma información.

Cuando se emplea un protocolo serie *asíncrono*, el emisor debe proporcionar una señal de sincronización antes de transferir cualquier dato, esto debe hacerse antes de cada mensaje transmitido. Las características de un protocolo asincrónico son:

- No se requiere señal de reloj entre el remitente y el destinatario del mensaje.
- Se admite la transmisión de datos a mayor distancia.
- La confiabilidad es mayor que la transmisión de datos sincrónica.

Cuando se utiliza el protocolo serie *síncrono*, tanto el emisor como el receptor utilizan la misma señal de reloj durante la transferencia de datos. Las características de la transmisión síncrona, obtenidas de [Weis, O., 2020], son:

- Admiten velocidades de transferencia de datos más altas que la asíncrona.
- Debe ser comunicada la señal de reloj entre el emisor y el receptor.
- Se requiere una configuración maestro/esclavo para implementar la transferencia de datos.

2.8.1. RS-485

RS-485 es un protocolo de comunicación de la capa física perteneciente al primer nivel del modelo OSI. La conexión para este protocolo es realizada utilizando un cable de dos o tres hilos: un hilo de datos, otro con datos invertidos y, frecuentemente, un hilo de tierra (0 V). Su principal función es transportar una señal a través de dos cables, uno de ellos transmite la señal original y el otro transporta su copia inversa [Weis, O., 2021]. Este método de transmisión ofrece una gran resistencia a las interferencias.

2.8.2. Modbus RTU

Modbus es un protocolo de comunicación industrial perteneciente a la capa de aplicación del modelo OSI. Para la transferencia de datos utiliza comunicación RS-485, RS-422 y RS-232, así como redes Ethernet TCP/IP. Existen varios tipos de implementación de Modbus, siendo los más populares:

- Modbus RTU
- Modbus ASCII
- Modbus TCP

Se hará énfasis en el tipo *Modbus RTU (Remote Terminal Unit)* ya que es el protocolo utilizado por el sensor de suelo que se detalla en la sección 3.2.3. El protocolo es de tipo maestro-esclavo donde solo el dispositivo maestro (en la presente tesis, un microcontrolador) puede iniciar la comunicación. El resto de dispositivos de la red son los esclavos (en este caso, el sensor a utilizar para tomar las mediciones del suelo) y responden a solicitudes por parte del maestro.

A través del bus Modbus se intercambian mensajes entre el maestro y el esclavo en forma de tramas de datos, que pueden ser de solicitud o respuesta. Una *solicitud* es un mensaje enviado por el maestro a uno de los esclavos mientras que una *respuesta* es un mensaje enviado por el esclavo al maestro. La solicitud también incluye una suma de verificación (*checksum*) que se utiliza para garantizar que el mensaje no esté dañado en el camino hacia el esclavo.

Un mensaje Modbus RTU típico contiene la dirección del esclavo, el código de función, los datos basados en el código de función y el CRC de la suma de verificación (Tabla 2.2). Es de importancia notar que cada parte de la trama está representada en forma hexadecimal.

Tabla 2.2: Trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
1 byte	1 byte	n bytes	2 bytes

El código de función especifica la acción que se debe realizar en el dispositivo esclavo, es decir, qué tipo de operación se solicita: lectura de datos, escritura de datos, diagnóstico, etc. Se detallan algunos ejemplos comunes de códigos de función Modbus RTU:

- 1. Lectura de registros (Read Holding Registers):** el código de función para la lectura de registros es 0x03. Se utiliza para solicitar al dispositivo esclavo que envíe los valores de uno o más registros de datos específicos.
- 2. Escritura de un solo registro (Write Single Register):** el código de función para escribir un solo registro es 0x06. Se usa para enviar al dispositivo esclavo una solicitud para modificar el valor de un solo registro.
- 3. Escritura de varios registros (Write Multiple Registers):** el código de función para escribir varios registros es 0x10. Se utiliza para enviar al dispositivo esclavo una solicitud para modificar los valores de múltiples registros consecutivos.
- 4. Diagnóstico (Diagnostics):** existen códigos de función específicos para diagnóstico, como 0x08, que se utilizan para solicitar al dispositivo esclavo que realice pruebas de diagnóstico.

Por ejemplo, la trama Modbus RTU de la Tabla 2.3 le indica al esclavo número 1 que devuelva el valor de un único registro que comienza en la dirección 2.

Tabla 2.3: Solicitud - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	00 02 00 01	25 CA

Notar que el código de función es 0x03, indicando lectura de registros. Los primeros dos bytes de datos (00 02) corresponde a la posición de comienzo del primer registro a leer y, los restantes dos bytes, al número de registros a leer. Todos los esclavos excepto el que posee el ID 1 ignoran este mensaje. El esclavo envía un mensaje de respuesta similar al de la Tabla 2.4.

Tabla 2.4: Respuesta - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	02 07 FF	FA 34

Se repite el ID y el código de función requerido por el maestro. El primer byte de datos (0x02) indica el número de bytes de información que devuelve, los bytes restantes es el valor del registro (0x07FF).

A continuación se detallan las solicitudes y respuestas Modbus RTU para leer los valores de nitrógeno, fósforo y potasio del sensor:

- La Tabla 2.5 muestra la *solicitud Modbus RTU* para la **lectura del nivel de nitrógeno**.

Tabla 2.5: Solicitud del nivel de nitrógeno - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	00 1E 00 01	E4 0C

La *respuesta* será similar a lo que muestra la Tabla 2.6.

Tabla 2.6: Respuesta del nivel de nitrógeno - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	02 00 20	B9 9C

Desde la respuesta se puede obtener el nivel de nitrógeno. En este caso, como la respuesta contiene el valor 0x0020 en el campo de datos, el nivel de nitrógeno se puede calcular por simple conversión: 0x0020 en hexadecimal es 32 en decimal. Por lo tanto, el nivel de nitrógeno es 32 mg/kg.

- La Tabla 2.7 muestra la *solicitud Modbus RTU* para la **lectura del nivel de fósforo**.

Tabla 2.7: Solicitud del nivel de fósforo - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	00 1F 00 01	B5 CC

La *respuesta* será similar a lo que muestra la Tabla 2.8.

Tabla 2.8: Respuesta del nivel de fósforo - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	02 00 25	79 9F

Desde la respuesta se puede obtener el nivel de fósforo. En este caso, como la respuesta contiene el valor 0x0025 en el campo de datos, que es 37 en decimal, el nivel de fósforo es 37 mg/kg.

- La Tabla 2.9 muestra la *solicitud Modbus RTU* para la **lectura del nivel de potasio**.

Tabla 2.9: Solicitud del nivel de potasio - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	00 20 00 01	85 C0

La *respuesta* será similar a lo que muestra la Tabla 2.10.

Tabla 2.10: Respuesta del nivel de potasio - trama Modbus RTU.

ID de esclavo	Código de función	Datos	Checksum
01	03	02 00 30	B8 50

Desde la respuesta se puede obtener el nivel de potasio. En este caso, como la respuesta contiene el valor 0x0030 en el campo de datos, que es 48 en decimal, el nivel de potasio es 48 mg/kg.

2.9. Reguladores de voltaje

La regulación del voltaje es un aspecto fundamental en los sistemas y dispositivos electrónicos. El propósito principal del mismo es mantener un nivel de voltaje constante, independientemente de las variaciones en la carga o en la fuente de alimentación. Existen dos tipos de

reguladores de voltaje: lineales y conmutados. Aunque ambos cumplen la misma función, difieren significativamente en su funcionamiento, eficiencia y aplicaciones ideales.

Un **regulador de voltaje lineal** opera mediante la variación continua del voltaje de entrada para producir un voltaje de salida constante. Este tipo de regulador utiliza un dispositivo de paso, como un transistor o un amplificador operacional, para ajustar el voltaje de salida en función de la carga y las variaciones en la entrada. Cuando la carga o la entrada cambian, el dispositivo de paso se ajusta para mantener el voltaje de salida constante.

Los reguladores lineales son simples de implementar y generan poco *ruido*⁶, lo que los hace adecuados para aplicaciones de precisión y sensibles al ruido. Sin embargo, su eficiencia energética es generalmente baja, especialmente cuando la diferencia entre el voltaje de entrada y de salida es grande. Esto se debe a que el exceso de energía se disipa como calor, lo que puede requerir sistemas de enfriamiento adicionales y resultar en una vida útil reducida del dispositivo.

Un **regulador de voltaje conmutado** funciona de manera diferente. En lugar de variar continuamente el voltaje de entrada, un regulador conmutado cambia rápidamente el voltaje de entrada, encendiendo y apagando (conmutando) un interruptor de potencia. La salida es el promedio del voltaje de entrada durante los períodos en que el interruptor está encendido y apagado. Básicamente, un regulador conmutado deja pasar voltaje en dirección a la carga en pulsos, los que luego se filtran para tener un voltaje uniforme [Boylestad et al., 2009].

Los reguladores conmutados son más eficientes que los lineales porque no disipan tanta energía en forma de calor. Sin embargo, generan más ruido debido a su funcionamiento de conmutación rápida, lo que puede interferir con otros componentes electrónicos. Además, son más complejos de diseñar e implementar debido a los requerimientos de componentes adicionales, como inductores y condensadores.

El uso de reguladores lineales o conmutados depende en gran medida de la aplicación específica y las necesidades del sistema. En general, los reguladores lineales son ideales para aplicaciones que requieren baja generación de ruido, como los circuitos de audio y las aplicaciones de control preciso. Sin embargo, su baja eficiencia y la generación de calor pueden ser

⁶El ruido eléctrico son señales de interferencias no deseadas que producen efectos más o menos perjudiciales.

problemáticas en aplicaciones con alto consumo de energía o en sistemas con restricciones de espacio y refrigeración.

Los reguladores conmutados son la mejor opción cuando la eficiencia energética es una preocupación primordial, como en dispositivos alimentados por baterías o en sistemas con altas diferencias de voltaje entre la entrada y la salida. Su generación de ruido puede mitigarse con técnicas de filtrado y diseño adecuado. A pesar de su mayor complejidad de diseño e implementación, su alta eficiencia y versatilidad los hacen atractivos para una amplia gama de aplicaciones.

2.10. Tecnologías de memoria

Los mecanismos de persistencia se refieren a las técnicas utilizadas en programación para asegurar que los datos generados por un programa se mantengan disponibles incluso después de que el mismo haya finalizado su ejecución. Estos mecanismos (como las memorias) son esenciales para garantizar la integridad y disponibilidad de los datos a largo plazo.

Los microcontroladores exhiben en general tres tipos de tecnologías de memoria: RAM o SRAM, tipo FLASH y tipo EEPROM. Un punto de vital importancia es la durabilidad de los datos almacenados en cada tipo de memoria (persistencia y retención de datos). Adicionalmente es posible agregar distintos tipos y tamaños de memoria externa.

La **SRAM (Static Random Access Memory)**, perteneciente al tipo de memoria volátil, se emplea para denominar a un tipo de tecnología de memoria RAM basada en semiconductores capaz de mantener los datos mientras siga alimentada, sin necesidad de *circuito de refresco*⁷. Este concepto surge en oposición a la memoria DRAM (Dynamic Random Access Memory), con la que se denomina al tipo de tecnología RAM basada en capacitores, que sí necesita del circuito de refresco.

La **EEPROM (Electrically Erasable Programmable Read-Only Memory)** es un tipo de memoria ROM, no volátil, que puede ser programada, borrada y reprogramada eléctricamente.

⁷Proceso de lectura periódica de información en un área de memoria e inmediata reescritura de la información leída, en la misma área, sin modificaciones.

camente. Aunque una EEPROM puede ser leída un número ilimitado de veces, solo puede ser borrada y reprogramada un número finito de veces.

La **memoria FLASH** es una derivación de las EEPROM. La principal diferencia es que las EEPROM son borrables a nivel de byte mientras que las FLASH lo son a nivel de bloques de diferente tamaño.

3. Desarrollo

En este capítulo se presenta la descripción del desarrollo del dispositivo FertSens. Las secciones que lo componen tratan sobre:

- la especificación y análisis del proyecto,
- los componentes electrónicos a utilizar en el dispositivo,
- el conexionado físico,
- la descripción de la implementación e interfaz gráfica; y finalmente,
- un script para la visualización de las mediciones por parte del usuario.

3.1. Especificación y análisis del proyecto

El propósito de este dispositivo es ser una herramienta para la asistencia de los usuarios, ya sean profesionales del agro, dueños de tierras o todas aquellas personas que requieran analizar el suelo en pos de controlar los niveles de nitrógeno, fósforo y potasio. Además, se persigue la optimización del tiempo de los usuarios, permitiéndoles realizar controles en cualquier momento, sin necesidad de recurrir repetidamente a laboratorios que brinden dicha información. Con este sistema se permite al usuario realizar las mediciones y obtener los resultados de forma inmediata, llegando a una conclusión acerca de los pasos a seguir.

3.1.1. Especificación de los requisitos del sistema

Una vez realizado el relevamiento de información acerca del proceso de análisis del suelo y los aspectos a considerar en relación a la medición de nitrógeno, fósforo y potasio, se definieron los requerimientos funcionales y no funcionales. Los requerimientos funcionales están ligados al comportamiento que debe tener el sistema al momento en que un usuario interactúe con él. Los requerimientos no funcionales describen las características de calidad del sistema.

Los **requerimientos funcionales** son los siguientes:

1. El usuario debe poder realizar la medición de nitrógeno, fósforo y potasio.
2. El usuario debe poder visualizar el historial de medición.

3. El usuario debe poder descargar las mediciones almacenadas en el dispositivo.
4. El usuario debe poder interactuar con las funcionalidades del dispositivo y navegar por las diferentes pantallas a través de una interfaz de usuario intuitiva.

Mientras que los **requerimientos no funcionales** son:

1. *Usabilidad*: la interfaz de usuario debe ser intuitiva y fácil de entender, incluso para usuarios con poca experiencia técnica. Para lograrlo, se mostrará por pantalla solo la información necesaria para tomar mediciones y visualizar el historial.
2. *Integridad*: las mediciones deben ser consistentes, confiables, precisas y no estar sujetas a corrupción o manipulación no autorizada. Para esto, las mediciones registradas del sensor de suelo serán almacenadas en una memoria externa, sin posibilidad de que el usuario pueda manipularlas desde el dispositivo.
3. *Rendimiento*: el dispositivo debe ser capaz de realizar las mediciones de manera rápida y eficiente, sin retrasos significativos entre las solicitudes del usuario y la visualización de los resultados en la pantalla. Para ello se utilizará código no bloqueante, interrupciones y los dos núcleos del microcontrolador.
4. *Portabilidad*: el dispositivo debe ser compacto y fácil de transportar, permitiendo su uso en diferentes ubicaciones y entornos. Para ello, los componentes electrónicos serán soldados en una placa de circuito impreso (PCB, por sus siglas en inglés) y encapsulados en una carcasa diseñada a medida.

3.1.2. Estándar de codificación

El objetivo de los estándares de codificación de software es llevar a cabo prácticas de programación comprobadas que conduzcan a un código seguro, confiable, reutilizable y mantenible. Por lo general, esto significa evitar prácticas no seguras o código que pueda causar un comportamiento impredecible. Esto se vuelve crítico en lenguajes como C y C++ debido al alto riesgo de escribir código inseguro [Seacord, 2013].

Los *estándares de codificación* son un conjunto de recomendaciones que describen los procedimientos ideales para crear código. Dentro de dichos estándares, que reducen los defectos y aumentan la calidad general, se utilizarán los siguientes en el desarrollo:

1. **Sangría y formato consistentes:** para que el código sea legible y posea un formato coherente se debe mantener la sangría, lo que facilita la identificación de fragmentos de código y fortalece la organización general del mismo. De esta manera, se comprende rápidamente el software y su flujo lógico.
2. **Convenciones de nombres descriptivos y significativos:** para comprender el código es crucial que las variables, funciones y otras entidades tengan nombres descriptivos y significativos. Esto facilita la comprensión de la función y el propósito de cada parte mediante etiquetas claras y breves.
3. **Uso de nomenclatura:** las nomenclaturas son formas de llamar a elementos, sentencias o acciones más específicas para tener uniformidad en el código y un estándar de trabajo. Para codificar en el microcontrolador se utilizará la nomenclatura *camelCase*, que es una práctica de escritura que consiste en la unión de dos o más palabras sin espacios entre ellas donde las diferencian una letra mayúscula inicial a partir de la segunda palabra.
4. **Modularización y reutilización:** se mejora la reutilización y la mantenibilidad dividiendo el código complejo en componentes modulares más pequeños. Este método no solo acelera el desarrollo sino que también asegura la consistencia y reduce la posibilidad de cometer errores.
5. **Comentarios:** el código bien escrito incluye comentarios que explican partes específicas y/o algoritmos complejos, actuando como un recurso útil para futuras modificaciones o adiciones.
6. **Manejo de errores y gestión de excepciones:** para que los proyectos de desarrollo sean resistentes y confiables, el manejo adecuado de errores y la gestión de excepciones son cruciales. Se evitan fallas brindando a los usuarios mensajes de error útiles al anticipar y administrar problemas potenciales y así mejorar la experiencia del mismo.

3.2. Componentes

En las subsecciones siguientes se presentan todos los módulos utilizados para el desarrollo del dispositivo de medición. Esto incluye una placa de desarrollo microcontrolada, un sensor NPK, una batería recargable con su módulo de gestión de recarga, una pantalla OLED para la interfaz de usuario y un transceptor para las señales de comunicación, entre otros. En cada

caso, se ofrece una breve descripción de las características y la funcionalidad del módulo, así como las razones por las cuales fue seleccionado para esta tesis.

3.2.1. Placa de desarrollo

Para la presente tesis se utiliza una placa de desarrollo diseñada y producida por Heltec Automation (Figura 3.1 y Figura 3.2), la cual es un producto altamente integrado. Esta placa fue elegida por varias razones:

- Posee un chip ESP32 de bajo costo con dos núcleos de procesamiento (dual-core), lo que permite la ejecución simultánea de diferentes tareas.
- Incluye una pantalla OLED de 0,96 pulgadas para la interfaz gráfica.
- Tiene comunicación Wi-Fi, lo que permite subir las mediciones a la base de datos en la nube.
- Cuenta con un sistema de administración de batería, evitando la necesidad de incorporar componentes adicionales para proveer alimentación regulada a la placa.



Figura 3.1: Vista superior de la placa de desarrollo. Extraído de [WiFi Kit 32 Datasheet]



Figura 3.2: Vista inferior de la placa de desarrollo. Extraído de [WiFi Kit 32 Datasheet]

En la Tabla 3.1 se detallan las especificaciones generales obtenidas de [ESP32 WiFi Kit Datasheet] y en el Anexo 8.1 se encuentra la asignación de pines (pinout) de la placa.

Tabla 3.1: Especificaciones técnicas de la placa de desarrollo.

Características	Descripción
Microcontrolador	ESP32 (240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS)
Interfaz USB a Serial	CP2102
WiFi	802.11 b/g/n (802.11n hasta 150 Mbps)
Bluetooth	Bluetooth V4.2 BR/EDR y Bluetooth baja energía
Tamaño de pantalla	OLED de 0,96 pulgadas
Recursos de hardware	UART x 3; SPI x 2; I2C x 2; I2S x 1; 12-bits ADC input x 18; 8-bits DAC output x 2; GPIO x 22; GPI x 6
Memoria	4Mb (64Mb) SPI FLASH; 520KB internal SRAM
Interfaces	Micro USB x 1; 18 x 2.54 pin x 2
Soporte de batería	3.7V litio (SH1.25 x 2 socket)
Temperatura operativa	-20 °C a 70 °C

3.2.2. Batería recargable 18650

Las baterías de iones de litio se utilizan en una variedad de dispositivos electrónicos, son livianas y tienen una alta densidad de energía, lo que significa que pueden almacenar más energía que otros tipos de baterías tradicionales.

Una batería 18650 (Figura 3.3) es una celda recargable de iones de litio que lleva el nombre de sus dimensiones: 18 mm de diámetro y 65 mm de largo. Puede soportar altas temperaturas y tiene tiempos de funcionamiento prolongados en comparación con las pilas alcalinas, como las AA desechables. A continuación se detallan algunas de las ventajas de usar baterías 18650:

- 1. Alta capacidad:** la *capacidad* es la cantidad de energía contenida en la batería y se mide en mAh (miliamperios-hora). Existen baterías 18650 que tienen capacidad de hasta 6,000

mAh, que es mucho más que la capacidad de una batería AA primaria (entre 500 mAh y 2700 mAh).

- 2. Baja tasa de autodescarga:** esto hace que duren mucho más que otro tipo de baterías recargables. La *autodescarga* es un evento natural que ocurre cuando no se usa una batería y se descarga lentamente con el tiempo.
- 3. Gran número de ciclos de carga:** esto significa que se pueden cargar y descargar cientos de veces, proporcionando una larga duración y evitando la necesidad de ser reemplazada con frecuencia.

La energía acumulada en una batería se mide en watts hora (Wh) y representa la capacidad de realizar un trabajo en un determinado tiempo. La potencia en watts (W) de una batería es igual al producto de su voltaje por su capacidad medida en miliamperios hora (mAh).

Las baterías pueden estar *protegidas* o *sin protección*. Las baterías protegidas tienen un circuito integrado (IC, por sus siglas en inglés) en sus carcasas que la protege contra descargas excesivas, sobrecarga y cortocircuitos. Aunque las baterías sin protección son más baratas y poseen mayor capacidad, la confiabilidad es mucho menor.

Debido a todas las ventajas mencionadas anteriormente, incluida la disponibilidad para su compra en el mercado, se eligió este tipo de batería para alimentar el dispositivo.



Figura 3.3: Batería 18650. Extraído de [18650 Battery Datasheet]

Las características de la batería 18650 a utilizar en la presente tesis, obtenidas de [18650 Battery Datasheet], se detallan en la Tabla 3.2.

Tabla 3.2: Especificaciones técnicas de la batería 18650.

Voltaje nominal	3,7 V
Capacidad	1500 mAh
Energía acumulada	5,55 Wh
Con IC de protección	Si

3.2.3. Sensor de suelo NPK

Existen diversos sensores de suelo que miden NPK en el mercado. Los sensores de la Figura 3.4 permiten obtener las mediciones y mostrarlas por un display. El sensor de la Figura 3.5 es un analizador rápido que muestra las mediciones de forma analógica, mientras que el sensor de la Figura 3.6 posee salida RS-485 y además mide PH, conductividad eléctrica, temperatura y humedad.



Figura 3.4: Sensor NPK Pilipane. Extraído de Amazon.



Figura 3.5: Sensor NPK Rapitest. Extraído de Amazon.



Figura 3.6: Sensor NPKPHCTH-S. Extraído de Amazon.

El sensor elegido para utilizar en el dispositivo es el que se muestra en la Figura 3.7, el cual es apto para todo tipo de suelo, solo mide NPK, es encapsulado al vacío, impermeable, puede ser enterrado a largo plazo y es resistente a la electrólisis y a la corrosión. Además, su encapsulado posee protección de grado *IP68*, lo que ofrece hermeticidad contra el polvo y protección contra la inmersión permanente, es decir, son adecuados para su uso bajo el agua.



Figura 3.7: Sensor NPK. Extraído de [Soil NPK sensor Datasheet]

A continuación se detallan los parámetros de medición del sensor:

- Rango de medición: 1-1999 mg/kg
- Precisión: $\pm 2\%$ FS
- Resolución: 1 mg/kg
- Tiempo de respuesta: ≤ 10 s

Se detallan en la Tabla 3.3 las especificaciones técnicas del sensor, obtenidas de [Soil NPK sensor Datasheet].

Tabla 3.3: Especificaciones técnicas del sensor NPK.

Alimentación	12 V - 24 V
Clase de protección	IP68
Temperatura operativa	5 °C a 45 °C
Puerto de comunicación	RS-485

El puerto de comunicación del sensor es *RS-485* y debido a que el microcontrolador no permite este tipo de comunicación, es necesario utilizar un conversor de señales. Para ello se utiliza el conversor RS-485 a TTL.

3.2.4. Módulo conversor RS-485 a TTL

El módulo que se muestra en la Figura 3.8 se basa en el IC MAX485 y se utiliza para convertir señales diferenciales, como las correspondientes al estándar RS-485, en señales seriales TTL y viceversa. Se permite comunicación del tipo *half-dúplex*⁸.

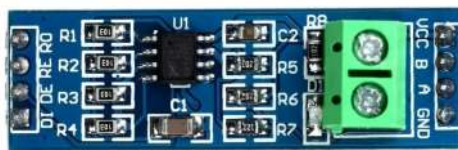


Figura 3.8: Conversor RS-485 a TTL. Extraído de [MAX485 Datasheet]

A continuación se describe la funcionalidad de los pines del módulo:

- **VCC:** es el pin de alimentación y su voltaje debe estar entre 4.75 V y 5.25 V.
- **GND:** es la referencia de voltaje 0.
- **A y B:** son los pines del bus diferencial.
- **RE (Receiver Output Enable):** permite habilitar o deshabilitar el modo de recepción.

Si se requiere recibir datos desde otro dispositivo, es necesario poner este pin en estado LOW o bajo.

⁸También denominado semidúplex, es un modo de envío de información que es bidireccional pero no simultáneo.

- **RO (Receiver Output):** es el pin por el cual se recibe la información enviada por otro módulo, usualmente es conectado al pin RX del microcontrolador. Para que la información sea recibida, el pin RE debe estar en estado bajo.
- **DE (Driver Output Enable):** permite habilitar o deshabilitar el modo de transmisión. Para eso debe ser puesto en estado HIGH o alto.
- **DI (Driver Input):** este pin es utilizado para transmitir información a otro dispositivo, usualmente es conectado al pin TX del microcontrolador. Es necesario que el pin DE esté configurado con un estado alto para poder transmitir.

Las especificaciones técnicas del módulo, obtenidas de [MAX485 Datasheet], se detallan en la Tabla 3.4.

Tabla 3.4: Especificaciones técnicas del módulo RS-485 a TTL.

Alimentación	5 V
Consumo de corriente	500 μ A
Temperatura operativa	-10 °C a 85 °C
Velocidad de comunicación	10 Mbit/s hasta 12m

El módulo funciona en modo *emisor* o *receptor* dependiendo de la conexión que se realice. Para que funcione en modo emisor, se conecta el pin DI al TX del microcontrolador y los pines RE y DE a VCC. Para que funcione en modo receptor, se conecta el pin RO al RX del microcontrolador y los pines RE y DE a GND. Para realizar una comunicación half-dúplex, se conectan los pines DE y RE a dos pines GPIO del microcontrolador y desde el código se establece al módulo como transmisor o receptor.

Se debe notar que, cuando se hace mención a conectar un pin a VCC o a GND, implica poner dicho pin en un estado alto o bajo, respectivamente. En el Anexo 8.2 se encuentran los detalles de la estructura interna del MAX485.

3.2.5. Módulo step-up MT3608

El módulo, que consta de un circuito integrado MT3608, es un elevador de voltaje conmutado diseñado para aplicaciones de baja potencia que tiene la capacidad de regular el voltaje de

salida hasta 28 V (gracias a un potenciómetro de alta precisión) y entregar una corriente máxima de 2 A. Posee aproximadamente un 93 % de eficiencia, bloqueo por bajo voltaje, limitación de corriente y protección contra sobrecarga térmica.

Debido a que el sensor NPK tiene un rango de alimentación de 12 V a 24 V y la batería a utilizar posee un voltaje nominal de 3,7 V, se necesita incrementar la tensión. Para ello se utiliza el módulo que se describe, el cual se ilustra en la Figura 3.9.



Figura 3.9: Módulo step-up. Extraído de [MT3608 Datasheet]

Las especificaciones técnicas propias del módulo, obtenidas de [MT3608 Datasheet], se detallan en la Tabla 3.5.

Tabla 3.5: Especificaciones técnicas del módulo step-up MT3608.

Voltaje de entrada	2 V a 24 V
Voltaje de salida	5 V a 28 V
Corriente máxima de salida	2 A

3.2.6. Módulo TP4056

Al utilizar una batería 18650 es necesario contar con un módulo capaz de recargarla. El TP4056 es un cargador lineal de batería de iones de litio de una sola celda que opera con una combinación de corriente constante y voltaje constante (CC/CV) que se utiliza para realizar una carga de forma rápida, segura y eficiente de las baterías Li-Ion. Primero suministra corriente regulada y finaliza la carga mediante voltaje [Gongora et al., 2013].

El módulo es ideal para cargar celdas 18650 y otras baterías de 3,7 V, protege la batería de sobrecargas, sobredescargas y *no posee protección de polaridad inversa*. El módulo se conecta

a la carga (la placa de desarrollo y el dispositivo de medición NPK) que será alimentada por la batería (OUT+ y OUT- en Figura 3.10).



Figura 3.10: Módulo TP4056. Extraído de [TP4056 Datasheet]

El módulo TP4056 se alimenta mediante un suministro de 5 V desde un cable USB tipo C o desde los terminales + y -. Para que se cargue correctamente una batería conectada en los terminales de salida B+ y B- es necesaria una corriente de 1 A y es de vital importancia desconectar la carga del módulo si se está cargando la batería.

Se debe notar que si bien el módulo se llama comercialmente TP4056, en realidad está compuesto por el circuito integrado TP4056, el chip de protección DW01 y el MOSFET 8205. El DW01 se encarga de proteger la batería de litio y garantizar una carga y descarga segura [DW01 Datasheet] mientras que el MOSFET 8205 es un transistor de efecto de campo de metal-óxido-semiconductor diseñado para aplicaciones de conmutación de baja potencia, como en este caso, la gestión de carga de baterías [MOSFET 8205 Datasheet].

Este componente fue elegido debido al pequeño tamaño del chip TP4056 y su bajo número de componentes externos, que lo hacen ideal para aplicaciones portátiles. Las especificaciones técnicas, obtenidas de [TP4056 Datasheet], se presentan en la Tabla 3.6.

Tabla 3.6: Especificaciones técnicas del módulo TP4056.

Voltaje de entrada	4,5 V a 5,5 V
Corriente de carga máxima	1 A
Corte de carga	4,2 V \pm 1 %
Protección de sobredescarga	2,5 V
Protección de corriente	3 A
Interfaz de entrada	USB tipo C

3.2.7. Pantalla OLED

Las pantallas OLED se destacan por su gran contraste, mínimo consumo de energía y buena calidad de imagen. La pantalla a utilizar en la presente tesis (Figura 3.11) tiene un tamaño de 1,3 pulgadas, emplea un driver SH1106 y posee una resolución de 128*64 píxeles. Se puede controlar cada píxel individualmente y mostrar tanto texto como gráficos, además, no necesita retroiluminación como las pantallas LCD, lo que implica que el consumo de energía sea menor.



Figura 3.11: Pantalla OLED. Extraído de [SH1106 Datasheet]

El motivo de su utilización se debe a que cuando se soldaron los pines del dispositivo, el soldador de estaño cayó sobre la pantalla integrada de la placa de desarrollo, provocando que se dañe. Se presenta en la Tabla 3.7 las especificaciones técnicas de la pantalla, obtenidas de [SH1106 Datasheet].

Tabla 3.7: Especificaciones técnicas de pantalla OLED.

Voltaje de operación	3 V a 5 V
Driver	SH1106
Resolución	128*64 píxeles
Temperatura de trabajo	-30 °C a 70 °C
Interfaz de comunicación	I2C

3.2.8. Módulo RTC (Real Time Clock)

El módulo RTC (Figura 3.12) contiene el circuito de soporte para el DS1307, el cual es un chip de bajo consumo de energía capaz de almacenar y llevar la cuenta de la fecha y hora.

Se provee información de segundos, minutos, horas, día, mes y año en formato *BCD*⁹ (Binary Coded Decimal).

La fecha es ajustada automáticamente a final de mes para meses con menos de 31 días, incluyendo las correcciones para año bisiesto. Funciona en formato de 12 o 24 horas y tiene incorporado un sensor de tensión que detecta la ausencia de energía para cambiar al suministro de la batería de respaldo CR2032 (circuito de conmutación).

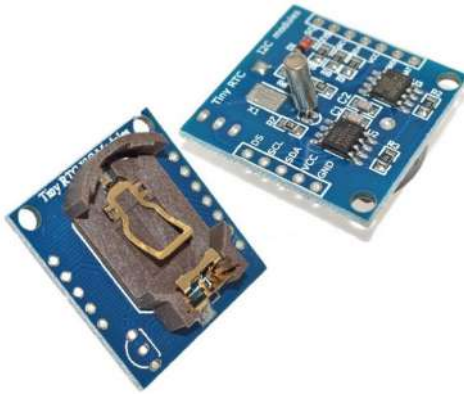


Figura 3.12: Módulo RTC. Extraído de [DS1307 Datasheet]

El motivo de su elección para implementar en el dispositivo se debe a la necesidad de contar con la fecha y hora exacta de medición de NPK. Se podría pensar en llevar a cabo un reloj interno en el ESP32, lo cual es una buena opción si nunca se desenergiza, pero al momento de cargar la batería del dispositivo se debe desconectar la carga (debido a una limitación del TP4056), lo que provocaría un desfase del reloj.

Las especificaciones técnicas del módulo, obtenidas de [DS1307 Datasheet], se detallan en la Tabla 3.8.

Tabla 3.8: Especificaciones técnicas del RTC.

Voltaje de alimentación	5 V
Consumo de corriente	500 nA
Interfaz de comunicación	I2C

⁹Es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits.

3.2.9. Módulo lector de memorias SD

Es necesario almacenar la información de las mediciones tomadas y, debido al poco espacio de almacenamiento en la memoria interna del microcontrolador, se debe utilizar una tarjeta de memoria externa. Por este motivo se utiliza el módulo ilustrado en Figura 3.13, el cual funciona con un circuito integrado 74LVC125A.



Figura 3.13: Módulo lector de memorias SD. Extraído de [74LVC125A Datasheet]

En la Tabla 3.9 se detallan las especificaciones técnicas del módulo, obtenidas de [74LVC125A Datasheet].

Tabla 3.9: Especificaciones técnicas del lector de memorias SD.

Voltaje de funcionamiento	3,3 V o 5 V
Interfaz de comunicación	SPI
Soporte	Tarjetas SD, SDHC y MicroSD con adaptador a SD.

3.2.10. Buzzer activo

Este módulo contiene un zumbador o buzzer activo, el cual posee su propia frecuencia de oscilación que permite generar un tono o pitido constante cuando se pone en bajo el pin I/O y no genera sonido alguno cuando se pone en alto (*low level trigger*). Se utiliza en el dispositivo para indicarle al usuario, mediante un sonido de corta duración, que la medición de NPK finalizó y que puede retirar el sensor del suelo.

El buzzer se ilustra en Figura 3.14 mientras que las especificaciones técnicas, obtenidas de [Buzzer Datasheet], están detalladas en la Tabla 3.10.



Figura 3.14: Buzzer activo. Extraído de [Buzzer Datasheet]

Tabla 3.10: Especificaciones técnicas del buzzer activo.

Voltaje de funcionamiento	3,3 V a 5 V
Corriente de operación	25 mA
Salida de sonido a 10 cm	85 dB
Frecuencia emitida	2300 ± 500 Hz
Temperatura de funcionamiento	-30 °C a 70 °C

3.3. Conexión física

Una vez obtenidos los componentes, se procede a realizar la conexión física. En la Figura 3.15 se presenta el diagrama pictórico mientras que en la Figura 3.16 se muestra el diagrama esquemático, ambos realizados en Fritzing, el cual es un programa libre de diseño electrónico. Ambos diagramas proporcionan la misma información, aunque de manera distinta. Uno de ellos ofrece una representación más gráfica mientras que el otro presenta una vista más técnica y esquemática de las mismas conexiones y componentes. Con estos diagramas como insumo, se diseña una placa de circuito impreso (PCB), como se verá en una sección posterior.

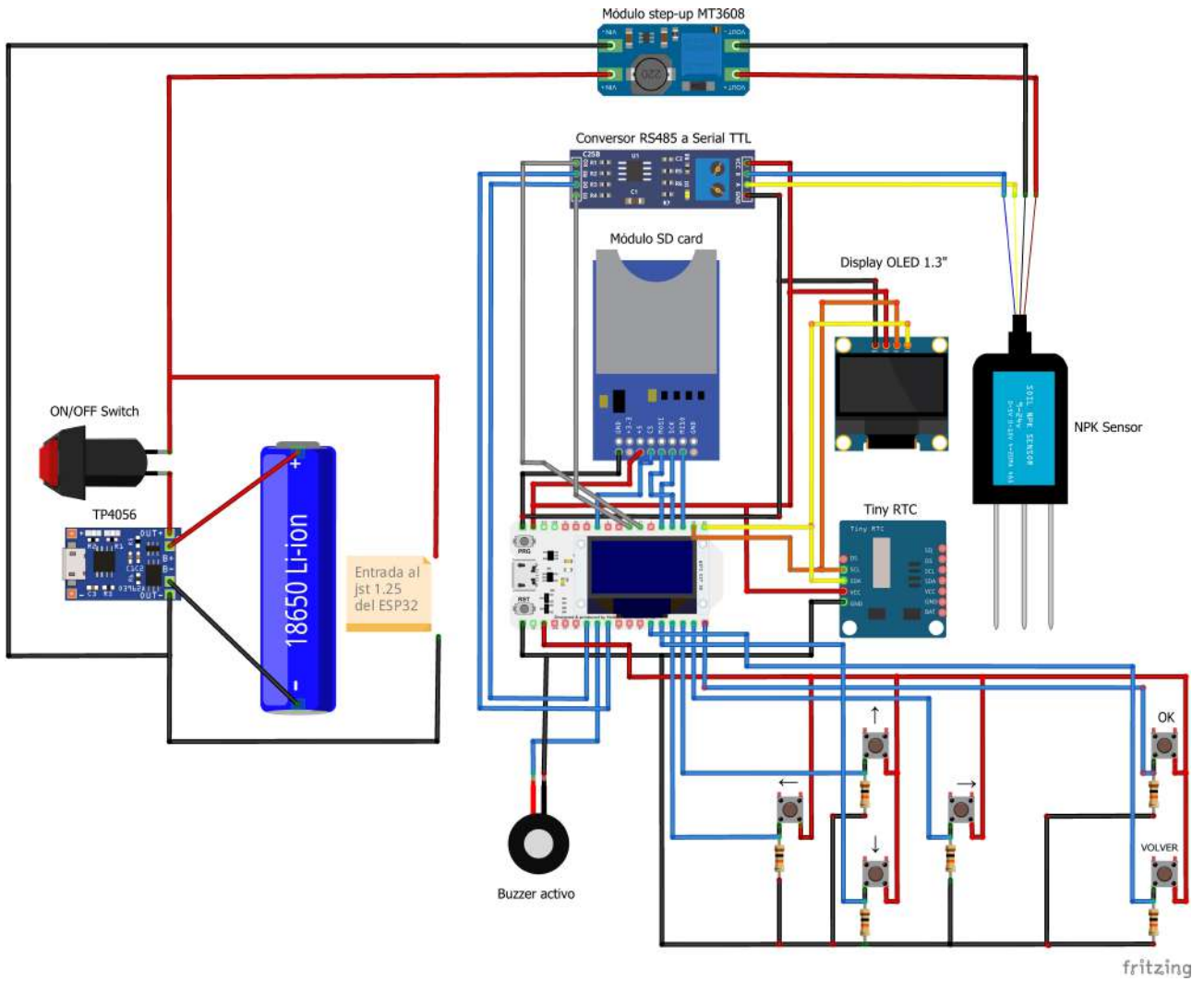


Figura 3.15: Diagrama pictórico.

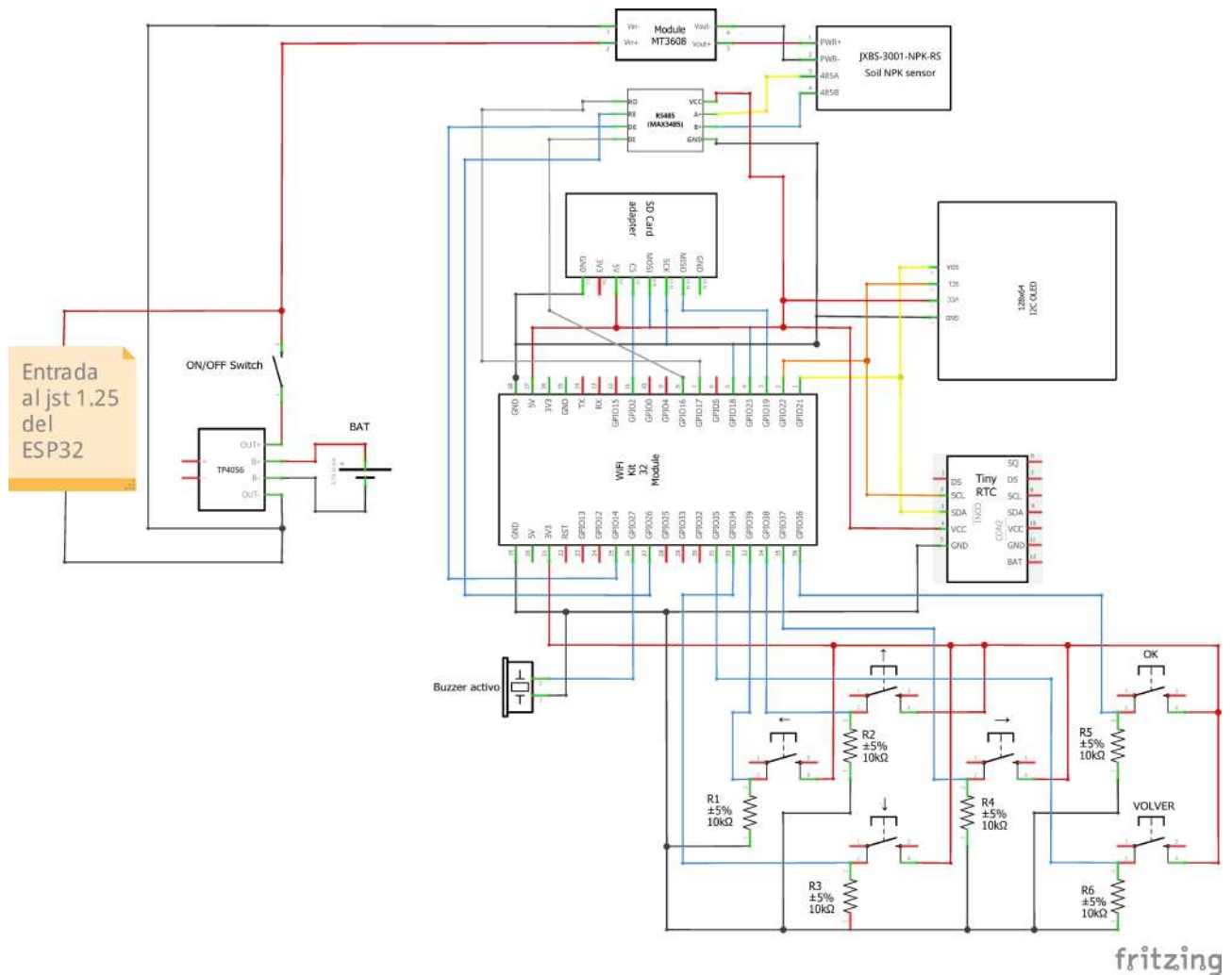


Figura 3.16: Diagrama esquemático.

En ambos diagramas se observa como se utilizan todos los componentes descritos en las secciones anteriores, como el módulo step-up que tiene como entrada el voltaje de la batería 18650 y como salida el voltaje para el funcionamiento del sensor NPK, o el conversor RS485 a serial TTL para comunicar el microcontrolador con el sensor de suelo. Como el ESP32 posee un regulador de voltaje interno, se conecta a la batería desde la entrada JST 1,25 mm.

Notar la presencia de 6 pulsadores, 4 de ellos serán utilizados para el movimiento entre las diferentes pantallas y opciones que posee el sistema, mientras que los 2 restantes son utilizados para la toma de decisiones (OK o ACEPTAR y VOLVER o CANCELAR). En la Tabla 3.11 se detalla la utilización de los pines del microcontrolador.

Tabla 3.11: Pines utilizados del microcontrolador.

Pin	Uso
17	RO del conversor RS485 a TTL
16	DI del conversor RS485 a TTL
2	CS del módulo lector de memorias SD
18	SCK del módulo lector de memorias SD
23	MOSI del módulo lector de memorias SD
19	MISO del módulo lector de memorias SD
22	SCL del RTC
21	SDA del RTC
22	SCL de pantalla OLED
21	SDA de pantalla OLED
26	RE del conversor RS485 a TTL
14	DE del conversor RS485 a TTL
27	I/O del buzzer
35	pulsador VOLVER
34	pulsador ↓
39	pulsador ←
38	pulsador ↑
37	pulsador →
36	pulsador OK

3.4. Implementación

Para la implementación se utilizó como IDE (Integrated Development Environment o entorno de desarrollo integrado) a Arduino IDE en la versión 2.3.2 descargado de su página oficial <https://www.arduino.cc/en/software>. El ESP32 no viene incluido en el mismo por defecto, motivo por el cual se debe utilizar el *gestor de placas* para añadirlo. En primer lugar, se abre el IDE, se ingresa a *Archivo* y luego a *Preferencias*. En el campo *URLs adicionales de gestor de placas* se añade la URL: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json.

En *Herramientas* → *Placa* → *Gestor de placas* se instala **esp32** de Espressif Systems y **Heltec ESP32 Series Dev-boards** de Heltec Automation (TM). Una vez completada la instalación, ya se podrán elegir los distintos modelos de ESP32 para programar. En este caso, se selecciona *Placa* → *Heltec ESP32 Series Dev-boards* → *WiFi Kit 32*.

Una vez configurado el entorno de desarrollo, se sigue con la programación del dispositivo. Debido a que los encabezados permiten la modularización, reutilización y el mantenimiento eficiente del código, se organiza el proyecto en los siguientes archivos:

- **librerias.h**: se incluyen tanto las librerías específicas de ESP32 como las librerías externas a utilizar por los módulos.
- **config.h**: se incluyen las constantes, estructuras de datos y configuraciones globales.
- **pines.h**: contiene las definiciones de los pines.
- **FertSens.ino**: se incluyen los encabezados mencionados anteriormente y es donde reside la lógica principal.

Cuando el dispositivo inicia verifica si hay conexión WiFi, en ese caso, se conecta a InfluxDB y corrobora si existen registros que no se han subido para proceder a hacerlo. Este proceso puede llevar un tiempo considerable, por lo que si se realiza en un solo núcleo, el usuario deberá esperar antes de poder interactuar con el dispositivo. Para evitar esto y cumplir con los requerimientos no funcionales en cuanto a rendimiento, se utilizan ambos núcleos del ESP32. El Código 3.1 muestra la función *xTaskCreatePinnedToCore* propia del ESP32 que se utiliza para crear y asignar una tarea a un núcleo específico.

```
1 xTaskCreatePinnedToCore(tareaWifiInflux, "tareaWifiInflux", 8192, NULL, 1, NULL, 0);
```

Código 3.1: Creación de tarea para conexión a WiFi e InfluxDB.

El primer parámetro es un puntero a la función que se ejecutará como tarea, en este caso *tareaWifiInflux*, el cual se muestra en Código 3.2. El segundo parámetro es un nombre descriptivo de la tarea. El tercer parámetro es el tamaño de la pila (stack) asignado a la tarea en bytes. El cuarto parámetro es un puntero a los parámetros que se pasan a la tarea, en este caso al no pasar ninguno, se pone *NULL*. El quinto parámetro es la prioridad de la tarea. El sexto parámetro es un puntero a la variable que almacenará el identificador de la tarea (task handle) pero como no se utiliza, se pone *NULL*.

El séptimo y último parámetro es el número del núcleo al que se fija la tarea, como el *setup()* y el *loop()* se ejecutan en el núcleo 1, se deja el núcleo 0 para ejecutar la función *tareaWifiInflux*. Dentro de la misma se utiliza *vTaskDelete(NULL)* para eliminar la tarea actual y así liberar los recursos, el parámetro *NULL* indica que la tarea a eliminar es la que está llamando a

vTaskDelete. El código para subir los registros a la nube y su explicación se muestra en el Anexo 8.4.1.

```
1 void tareaWifiInflux(void *pvParameters) {
2   conectarWifi();
3   if (WiFi.status() == WL_CONNECTED){
4     conectarInflux();
5     subirRegistros();
6   }
7
8   vTaskDelete(NULL);
9 }
```

Código 3.2: Función para conexión a WiFi e InfluxDB.

Para mostrar información por pantalla se necesita inicializar el módulo OLED (Código 3.3) e indicar los pines SDA y SCL para la comunicación I2C. Se observa que el constructor tiene los parámetros (-1, -1), esto se debe a que se utilizan los pines SDA y SCL por defecto de la placa. Luego el Código 3.4 muestra una función la cual recibe como parámetros los valores de N, P y K que ubicará en la pantalla para luego mostrar.

```
1 Adafruit_SH1106 oled(-1,-1);
2 oled.begin(SH1106_SWITCHCAPVCC, 0x3C);
```

Código 3.3: Inicialización de pantalla OLED.

```
1 void resultadoMedicion(uint16_t N, uint8_t P, uint8_t K){
2   oled.clearDisplay();
3   oled.setTextSize(1);
4   oled.setTextColor(WHITE);
5   oledDisplayCenter("RESULTADO", -25);
6   oled.setCursor(0, 25); oled.println("N: " + String(N) + " mg/kg");
7   oled.setCursor(0, 35); oled.println("P: " + String(P) + " mg/kg");
8   oled.setCursor(0, 45); oled.println("K: " + String(K) + " mg/kg");
9   oled.display();
10 }
```

Código 3.4: Muestra de medición por OLED.

Para almacenar los registros en la tarjeta de memoria externa se utiliza el módulo lector de memorias SD (sección 3.2.9). El Código 3.5 muestra la inicialización del mismo, en el cual se presenta una bifurcación donde en caso de que no esté la tarjeta SD insertada, se notifica al usuario por pantalla de la situación, el cual puede realizar mediciones sin que sean almacenadas.

```

1 void inicializarSD(void){
2     if (!SD.begin(CS_PIN)) {
3         pantallaActual = sinSD;
4         existSD = false;
5         pantallaSinSD();
6     }
7     else {
8         pantallaActual = opcionesMedicion;
9         pantallaInicio();
10    }
11 }

```

Código 3.5: Inicialización del lector SD.

El Código 3.9 ilustra la forma en la que se obtienen los datos que conforman el registro final. En lo que respecta al funcionamiento del RTC, se debe verificar que esté configurado y, en caso de no ser así, se utiliza como referencia la fecha y hora de compilación, como lo muestra el Código 3.6.

```

1 if (!rtc.isrunning()) rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

```

Código 3.6: Configuración de RTC.

Al momento de almacenar un registro se debe obtener la fecha y hora, para ello se utiliza una variable de tipo `DateTime` llamada `nowRTC`. La variable `unixtime` (de tipo `uint32_t`) almacena la fecha y hora actual en formato de tiempo Unix (el número de segundos transcurridos desde el 1 de enero de 1970). Luego se le envía al sensor de suelo la trama Modbus RTU (Código 3.7) que le indica al esclavo 0x01 (ID por defecto) que lleve a cabo la acción 0x03 (lectura) a partir de la posición 0x00 0x1e, que el número de registros para llevar a cabo dicha acción es 0x00 0x03 y que los últimos dos bytes es la verificación de errores.

```

1 const byte npkFrame[] = {0x01, 0x03, 0x00, 0x1e, 0x00, 0x03, 0x65, 0xCD};

```

Código 3.7: Trama Modbus RTU de requerimiento.

De esta manera, como se muestra en el Código 3.8, la respuesta será de 11 bytes donde el primero es el ID del dispositivo, el segundo byte es el código de función, el tercero es el número de bytes que se devuelven, el cuarto y quinto representa el contenido de nitrógeno, el sexto y séptimo representa el contenido de fósforo, el octavo y noveno representa el contenido de potasio y los dos últimos representan la verificación de errores.


```

1 byte npkMedicion[] = {0x01, 0x03, 0x06, 0x00, 0x20, 0x00, 0x25, 0x00, 0x30, 0x5A,
  ↪ 0x3D};

```

Código 3.8: Trama Modbus RTU de respuesta.

En las variables n , p y k del tipo *uint16_t* se almacena el contenido de cada nutriente gracias a operaciones a nivel de bits (desplazamiento hacia la izquierda y operación OR). Por último, en el caso de estar insertada la tarjeta SD se guardará el registro.

```

1 void obtenerMediciones(void){
2   nowRTC = rtc.now();
3
4   unixtime = nowRTC.unixtime();
5   fecha = String(nowRTC.day()) + "/" + String(nowRTC.month()) + "/" +
  ↪ String(nowRTC.year() % 100);
6   hora = String( (nowRTC.hour() < 10 ? "0" : "") + String(nowRTC.hour()) ) + ":" +
  ↪ String( (nowRTC.minute() < 10 ? "0" : "") + String(nowRTC.minute()) );
7
8   digitalWrite(DE, HIGH);
9   digitalWrite(RE, HIGH);
10  delay(1);
11  for (uint8_t i = 0; i < sizeof(npkFrame); i++ ) SerialNPK.write( npkFrame[i] );
12  SerialNPK.flush();
13  digitalWrite(DE, LOW);
14  digitalWrite(RE, LOW);
15  delay(2000);
16  for (byte b = 0; b < TAM_FRAME_RESP; b++) npkMedicion[b] = SerialNPK.read();
17  n = (npkMedicion[3] << 8) | npkMedicion[4];
18  p = (npkMedicion[5] << 8) | npkMedicion[6];
19  k = (npkMedicion[7] << 8) | npkMedicion[8];
20
21  if(existSD) agregarRegistro(unixtime, fecha, hora, n, p, k);
22 }

```

Código 3.9: Obtención de datos para el registro.

3.4.1. InfluxDB

InfluxDB (Figura 3.17) es una plataforma de series temporales que posee una base de datos diseñada específicamente para manejar grandes volúmenes de datos con marca de tiempo producidos por sensores, aplicaciones o dispositivos. Está diseñada para el almacenamiento rápido y de alta disponibilidad, es compatible para la utilización en IoT y soluciones de automatización, es de código abierto y fue desarrollada por la empresa InfluxData [InfluxData, 2024].



Figura 3.17: Logo de InfluxDB. Extraído de [InfluxData, 2024]

InfluxDB admite la recopilación, almacenamiento, monitoreo, visualización y alerta de datos de series temporales. Cada dato que se le envíe tendrá un sello con la fecha y hora en que se recibió, haciéndolo realmente útil para cualquier información que cambie con el tiempo.

El modelo de datos de InfluxDB no utiliza el concepto de tablas, en su lugar, organiza los datos en buckets (depósitos), los cuales pueden contener varias medidas y estos a su vez contienen múltiples etiquetas y campos. Las *medidas* son como tablas SQL, las *etiquetas* son valores indexados y los *campos* son los datos reales.

Esta plataforma cuenta con visualizaciones y paneles básicos pero para sacar el máximo potencial debe implementarse junto con soluciones dedicadas de visualización y tablero, como Grafana o Chronograf. La base de datos puede existir en la nube, en los servidores de InfluxDB o puede instalarse en un servidor local. En este caso, se utilizarán los servidores de la plataforma para almacenar los datos provenientes del sensor de suelo. La versión a utilizar es gratuita, teniendo la limitación de un período de almacenamiento máximo de 30 días. Los campos que son subidos desde el dispositivo a InfluxDB son:

- *unixtime*: de tipo entero sin signo de 32 bits que almacena la estampa de tiempo en formato Unix del registro.
- *fecha*: de tipo string que almacena la fecha del registro.
- *hora*: de tipo string que almacena la hora del registro.
- *n*: de tipo entero sin signo de 16 bits que almacena el valor del nitrógeno leído desde el sensor de suelo.
- *p*: de tipo entero sin signo de 16 bits que almacena el valor del fósforo leído desde el sensor de suelo.
- *k*: de tipo entero sin signo de 16 bits que almacena el valor del potasio leído desde el sensor de suelo.

Notar que se utiliza redundancia, ya que el campo `unixtime` posee la misma información que el campo `fecha y hora`. Sin embargo, se hizo de esta manera porque el valor `unixtime` es utilizado por Grafana, mientras que el campo `fecha y hora` es usado por el dispositivo para mostrar el historial, sin necesidad de una función adicional para realizar la conversión de tiempo.

Para conectar el dispositivo a la base de datos en la nube, primero se declara un cliente InfluxDB y luego se valida la conexión con el servidor, tal como se muestra en el Código 3.10 y 3.11 respectivamente. El tipo de dato `Point` (perteneciente a la librería `InfluxDbClient.h`) permite especificar varios aspectos de una medición, tales como la medida, las etiquetas, los campos y la marca de tiempo.

Notar que el argumento de `npkRegistro` es “`npkRegistro`” (no necesariamente tiene que ser el nombre de la variable igual al argumento) e indica la medida del bucket. En este caso, para almacenar los parámetros químicos en InfluxDB, el bucket se llama `NPK` y la medida `npkRegistro`. El código donde se observa su utilización y se agregan los campos `N`, `P` y `K` a `npkRegistro`, se presenta en Anexo 8.4.1.

```
1 InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET,  
2 ↪ INFLUXDB_TOKEN, InfluxDbCloud2CACert);  
Point npkRegistro("npkRegistro");
```

Código 3.10: Declaración de cliente InfluxDB.

Donde `INFLUXDB_URL` es la URL del servidor de InfluxDB, `INFLUXDB_ORG` es el nombre de la organización, `INFLUXDB_BUCKET` es el nombre del bucket donde se escribirán los datos, `INFLUXDB_TOKEN` es el token de autorización e `InfluxDbCloud2CACert` es el certificado de confianza del servidor InfluxDB. A continuación se detalla cada uno:

- `INFLUXDB_URL`: <https://us-east-1-1.aws.cloud2.influxdata.com>
- `INFLUXDB_TOKEN`: `JEb4ov5NcP_Veo6Mm3yP-vQyItTWL6sfO1IFx0obOaT465ieBP3t`
- `INFLUXDB_ORG`: `d5a8e77f51f05353`
- `INFLUXDB_BUCKET`: `NPK`

```

1 void conectarInflux(void){
2   if (client.validateConnection()) {
3     Serial.print("Conectado a InfluxDB: ");
4     Serial.println(client.getServerUrl());
5   } else {
6     Serial.print("Conexión fallida a InfluxDB: ");
7     Serial.println(client.getLastErrorMessage());
8   }
9 }

```

Código 3.11: Validación con el servidor InfluxDB.

3.4.2. Grafana

Grafana (Figura 3.18) es una herramienta de interfaz de usuario centralizada en la obtención de datos a partir de consultas, como también del almacenamiento de los mismos y su visualización. Es una solución de código abierto y sirve para ejecutar análisis de datos, extraer métricas y monitorear aplicaciones y recursos de hardware con la ayuda de paneles de control personalizables [Grafana, 2024]. Se conecta con fuentes de datos tales como Graphite, Prometheus, InfluxDB, MySQL, PostgreSQL, entre otros.



Figura 3.18: Logo de Grafana. Extraído de [Grafana, 2024]

Para la presente tesis Grafana se conecta a la fuente InfluxDB, por lo que toda medición subida desde el dispositivo se refleja automáticamente en el panel de control. Para ello, dentro de Grafana se debe configurar la conexión: *Connections*, *Add new connection* se elige la opción *InfluxDB* de la lista de fuente de datos y se clickea en *Add new data source*. En la página de configuración de la fuente de datos se ingresa el nombre que se le desee asignar a la misma y en el apartado *Query Language* se elige *Flux*.

En *URL* se ingresa la URL de InfluxDB, que en este caso es `https://us-east-1-1.aws.cloud2.influxdata.com/`. Para finalizar se ingresa el API token de InfluxDB (dentro de InfluxDB, se debe ingresar a *Load Data, API tokens* y se genera) y se elige el bucket por defecto para la lectura de datos. De esa manera ya se podrán crear paneles de control con sus diferentes visualizaciones.

Flux es el lenguaje de programación de datos funcional de InfluxData diseñado para consultar, analizar y actuar sobre datos. Flux será utilizado para realizar las consultas dentro de cada panel de control y así visualizar gráficamente la información.

La Figura 3.19, Figura 3.20 y Figura 3.21 muestran los paneles que indican los niveles de nitrógeno, fósforo y potasio, respectivamente. El Código 3.12 presenta la consulta Flux para estos tres paneles, donde luego de forma manual se configura en Grafana el nutriente a mostrar.

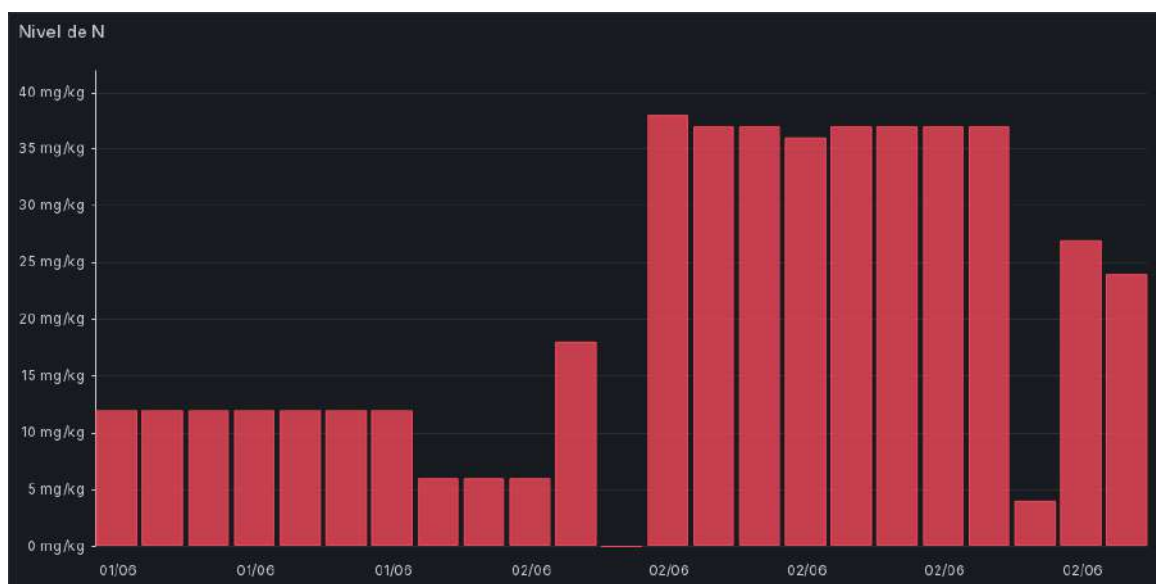


Figura 3.19: Panel de nitrógeno - Grafana.

La línea 2 del Código 3.12 se utiliza para filtrar los datos según un intervalo de tiempo específico que el usuario selecciona en la interfaz de Grafana. Esta porción de código se utiliza en las demás consultas.

```
1 from(bucket: "NPK")
2 |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3 |> filter(fn: (r) => r["_measurement"] == "npkRegistro")
```

Código 3.12: Consulta Flux para obtener los niveles de N, P y K.



Figura 3.22: Panel de niveles NPK - Grafana.

```

1 from(bucket: "NPK")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "npkRegistro")
4   |> filter(fn: (r) => r["_field"] != "unixTimeStamp")

```

Código 3.13: Consulta Flux para obtener los niveles de N, P y K por hora.

La Figura 3.23 y su consulta Flux asociada (Código 3.14) muestran el promedio de los tres nutrientes en el rango de tiempo elegido por el usuario en Grafana.

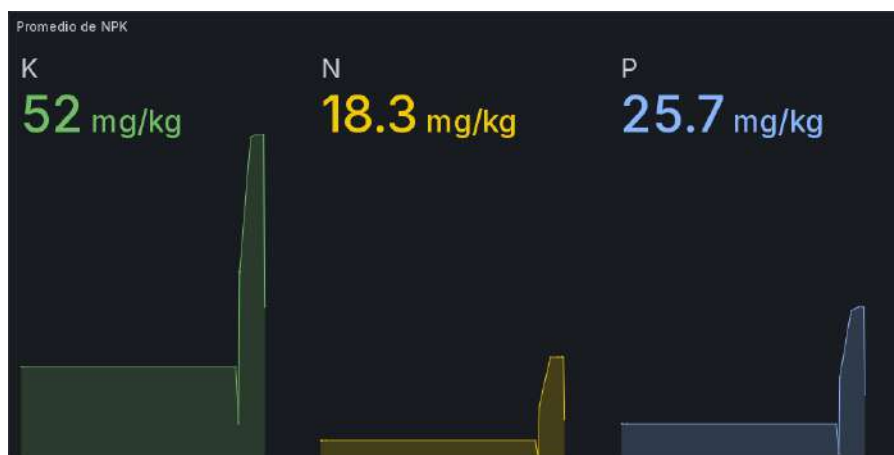


Figura 3.23: Panel de promedio de NPK - Grafana.

```

1 from(bucket: "NPK")
2 |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3 |> filter(fn: (r) => r["_measurement"] == "npkRegistro")
4 |> filter(fn: (r) => r["_field"] == "N" or r["_field"] == "P" or r["_field"] ==
   ↪ "K")
5 |> aggregateWindow(every: 5m, fn: mean, createEmpty: false)

```

Código 3.14: Consulta Flux para obtener el promedio de N, P y K.

3.5. Interfaz gráfica

Se utilizará el display OLED (sección 3.2.7) para la interfaz gráfica que interactuará con el usuario. Para obtener una aproximación de las funcionalidades del dispositivo, se diseñaron prototipos de las diversas pantallas posibles. Se logró un diseño simple que prioriza las necesidades del usuario.

La Figura 3.24 ilustra un prototipo de la pantalla que se muestra cuando se enciende el dispositivo mientras que en la Figura 3.25 se expone el ejemplo de una medición. El resto de prototipos diseñados se pueden ver en el Anexo 8.3.

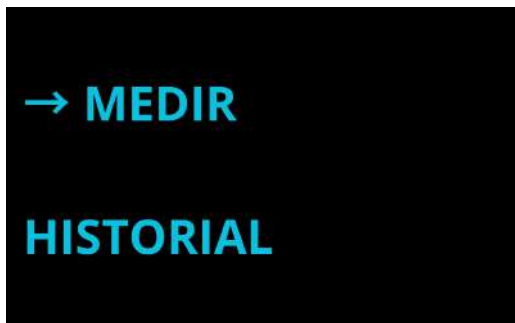


Figura 3.24: Pantalla de inicio.

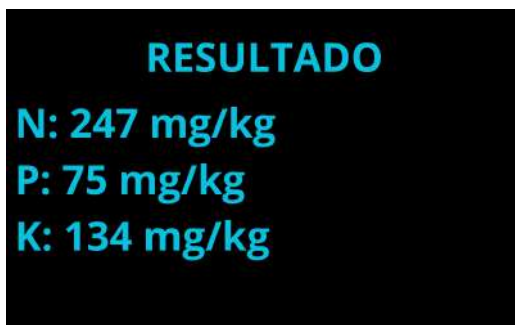


Figura 3.25: Pantalla de resultado.

En la Figura 3.26 se ilustra el diagrama de flujo de la interfaz de usuario. El mismo podrá tomar las mediciones que requiera, cancelar en caso de equivocación y visualizar el historial.

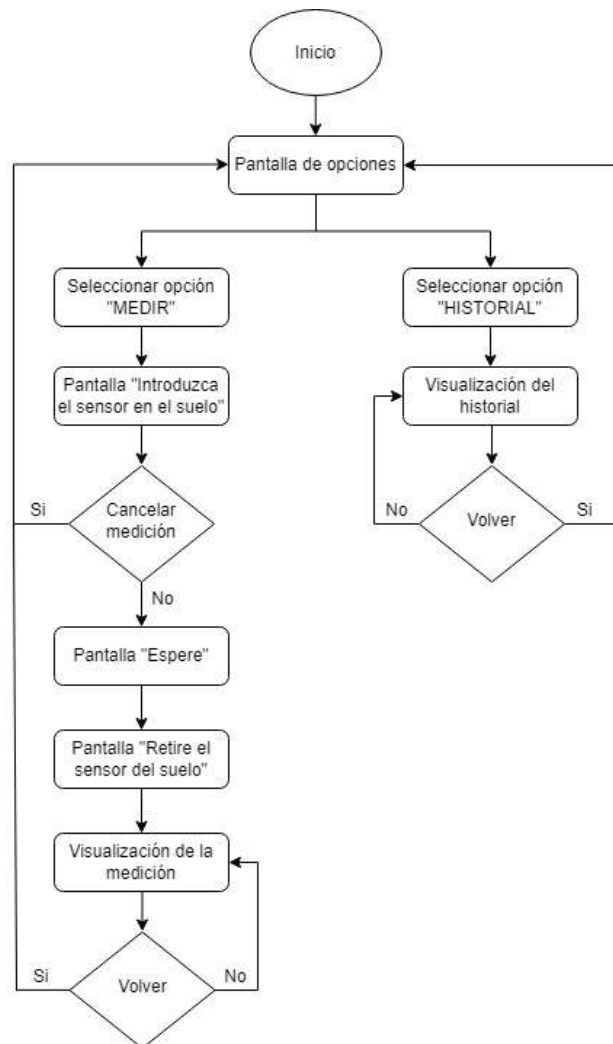


Figura 3.26: Diagrama de flujo de la interfaz de usuario.

3.6. Firmware

El firmware se define como un tipo de software embebido en la memoria de lectura de un dispositivo. Se encarga de proporcionar las instrucciones sobre el comportamiento del mismo y activar las funciones básicas [INCIBE, 2023].

El sensor NPK (sección 3.2.3) se demora 10 segundos, aproximadamente, en obtener la composición química del suelo mediante electrodos. La señal de salida de dicho sensor (RS-485) es convertida a serial TTL a través del módulo conversor (sección 3.2.4) para ser interpretado por

el microcontrolador, comunicación que se realiza a través del protocolo Modbus RTU (sección 2.8.2).

Una vez obtenida la medición, se almacena en la tarjeta de memoria externa y cuando el dispositivo posea conexión WiFi se envía a InfluxDB. En la pantalla OLED se visualizan las opciones posibles que tiene el usuario para llevar a cabo, como la toma de mediciones o la visualización del historial. Lo explicado anteriormente se muestra en el diagrama de flujo de la Figura 3.27.

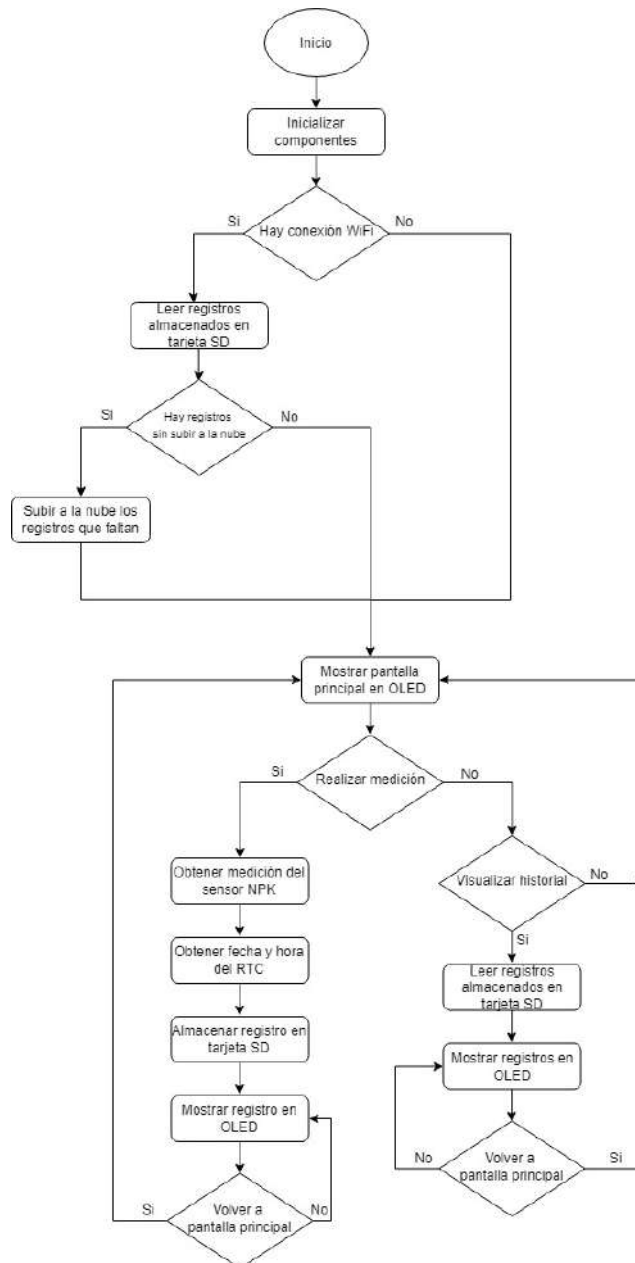


Figura 3.27: Diagrama de flujo del firmware.

3.7. Exportación de datos para análisis del usuario

Los registros (estampa de tiempo y medición de fertilidad) son almacenados dentro de un archivo JSON en la tarjeta de memoria externa. Para que el usuario pueda visualizarlos de una manera amigable, se realizó un script en Python que transforma el archivo JSON en una planilla Excel, como se observa en la Figura 3.28.



Fecha	Hora	N	P	K
1/6/24	19:51	12	17	34
1/6/24	20:49	12	17	34
1/6/24	21:14	12	17	34
2/6/24	15:18	6	8	17
2/6/24	15:22	18	25	50
2/6/24	15:23	38	53	107
2/6/24	15:24	37	52	104
2/6/24	17:35	37	52	104
2/6/24	17:42	37	52	104
2/6/24	17:47	4	5	11
2/6/24	17:48	24	34	68

Figura 3.28: Planilla de Excel con los registros.

Para generar el reporte, el usuario debe copiar el archivo *registros.json* proveniente de la tarjeta de memoria externa, pegarlo en la ubicación del script y ejecutarlo. De esta manera obtendrá un archivo Excel con el nombre "*Historial - DD-MM-AA, hh mm.xlsx*" siendo DD-MM-AA y hh mm la fecha y hora en el que el script fue ejecutado. El código se muestra en el Anexo 8.4.2.

4. Construcción de FertSens

En este capítulo se muestran los diseños de la carcasa y la placa de circuito impreso donde se ensamblan los componentes electrónicos. Al final se ilustran imágenes del dispositivo final ensamblado.

4.1. Dispositivo final

La carcasa del dispositivo fue diseñada con SolidWorks, el cual es un software de diseño asistido por computadora para modelar piezas y ensamblajes en 3D y planos en 2D. Las Figuras 4.1, 4.2 y 4.3 ilustran las representaciones del modelo 3D.

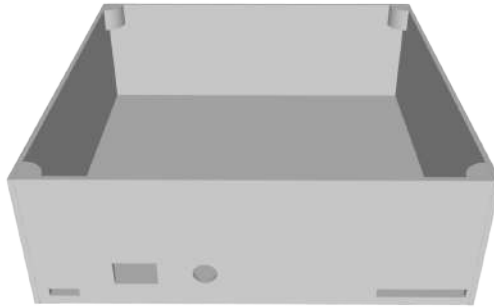


Figura 4.1: Carcasa del dispositivo - vista frontal.

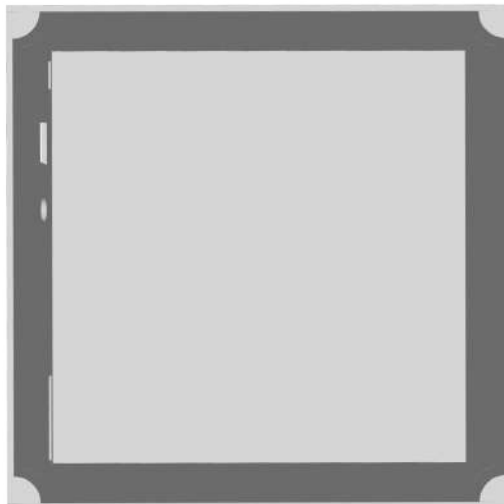


Figura 4.2: Carcasa del dispositivo - vista superior.

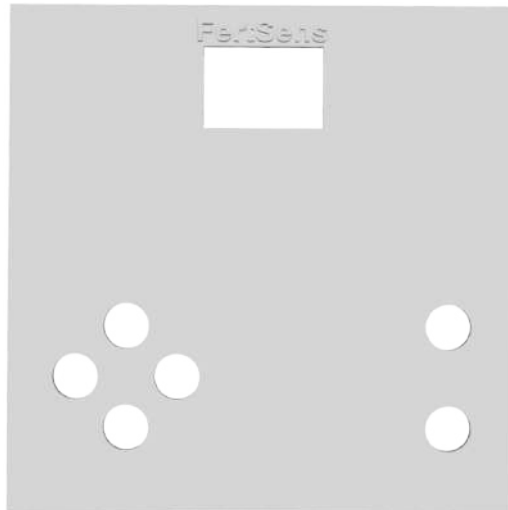


Figura 4.3: Tapa del dispositivo.

La Figura 4.4 ilustra el diseño de la PCB diseñada en KiCad, un software libre y de código abierto para la automatización del diseño electrónico. La PCB física, realizada en el Laboratorio de Electrónica de la Facultad de Ingeniería de la UNLPam, se muestra en Figura 4.5 y se utiliza para realizar el montaje y la integración de los distintos módulos y componentes electrónicos.

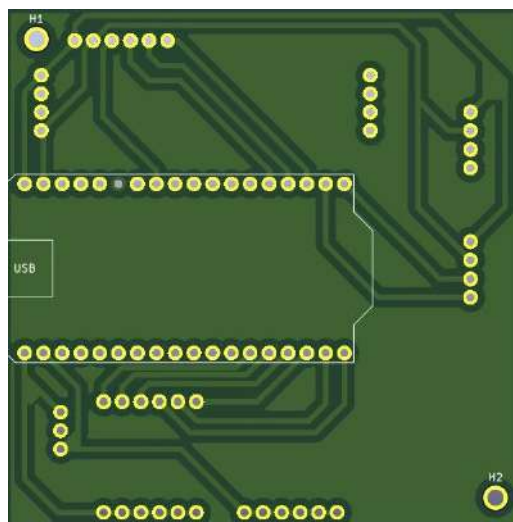


Figura 4.4: Diseño de la PCB.

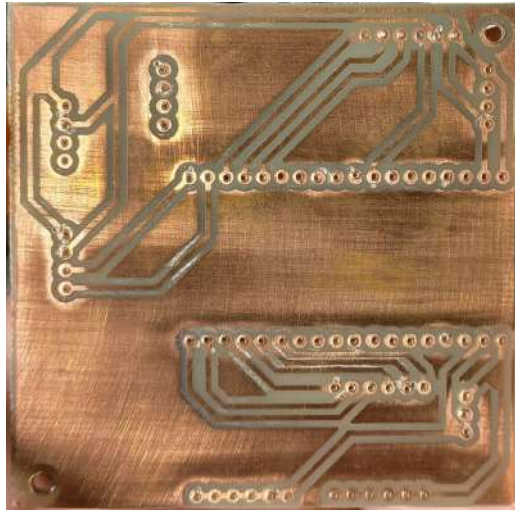


Figura 4.5: PCB del dispositivo.

Después de soldar todos los componentes a la placa, el siguiente paso fue ensamblar la misma en la carcasa. Este proceso permitió completar la construcción del dispositivo, que se muestra en Figura 4.6, Figura 4.7 y Figura 4.8. Las demás imágenes del dispositivo se encuentran en el Anexo 8.5.

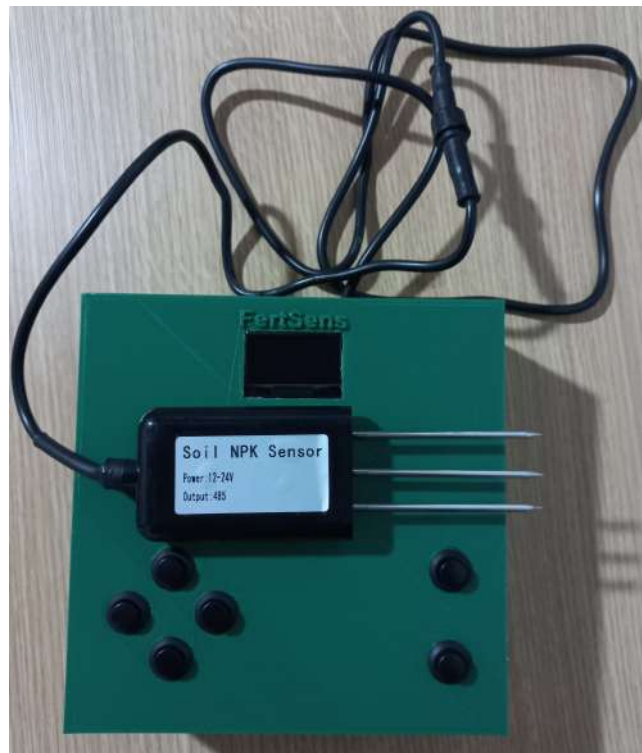


Figura 4.6: Dispositivo final - vista superior.



Figura 4.7: Dispositivo final - vista frontal.



Figura 4.8: Dispositivo final - tapa.

5. Caso de estudio

En este capítulo se describe el caso de estudio que sirvió para poner en práctica el dispositivo desarrollado y comprobar la precisión y confiabilidad de la solución planteada.

5.1. Pruebas y resultados obtenidos

Se mantuvo una conversación con productores y propietarios de tierras que necesitaban realizar análisis de sus suelos. Debido a que estos análisis se llevarían a cabo en los meses de octubre y noviembre, se decidió consultar con un laboratorio que disponía de muestras almacenadas. Se efectuaron mediciones de NPK con el dispositivo en tres muestras brindadas por el laboratorio de análisis agronómico (Figura 5.1).

En Tabla 5.1 se muestran los resultados de FertSens, mientras que los resultados del laboratorio se presentan en Tabla 5.2. Se mantuvo la numeración de muestras propias del laboratorio (6898, 6899 y 6900).



Figura 5.1: Muestras de tierra.

Tabla 5.1: Resultados de medición NPK realizadas con el dispositivo.

N° muestra	N (mg/kg)	P (mg/kg)	K (mg/kg)
6898	31	21	87
6899	32	27	90
6900	28	41	81

Tabla 5.2: Resultados de medición NPK del laboratorio.

N° muestra	N (mg/kg)	P (mg/kg)	K (mg/kg)
6898	28	19	81
6899	28	24	85
6900	26	37	78

En Tabla 5.3, Tabla 5.4 y Tabla 5.5 se muestran las tablas comparativas entre los resultados obtenidos con el dispositivo FertSens y el laboratorio. Además se indica el error de medición (error relativo) según la fórmula: $\left(\frac{|M_{\text{dispositivo}} - M_{\text{laboratorio}}|}{|M_{\text{laboratorio}}|}\right) \times 100$.

Tabla 5.3: Tabla comparativa de resultados de medición NPK - Muestra 6898.

Muestra 6898	Result. del dispositivo	Result. del laboratorio	Error de medición
Nitrógeno (N)	31 mg/kg	28 mg/kg	10,71 %
Fósforo (P)	21 mg/kg	19 mg/kg	10,52 %
Potasio (K)	87 mg/kg	81 mg/kg	7,41 %

Tabla 5.4: Tabla comparativa de resultados de medición NPK - Muestra 6899.

Muestra 6899	Result. del dispositivo	Result. del laboratorio	Error de medición
Nitrógeno (N)	32 mg/kg	28 mg/kg	14,28 %
Fósforo (P)	27 mg/kg	24 mg/kg	12,5 %
Potasio (K)	90 mg/kg	85 mg/kg	5,88 %

Tabla 5.5: Tabla comparativa de resultados de medición NPK - Muestra 6900.

Muestra 6900	Result. del dispositivo	Result. del laboratorio	Error de medición
Nitrógeno (N)	28 mg/kg	26 mg/kg	7,7 %
Fósforo (P)	41 mg/kg	37 mg/kg	10,81 %
Potasio (K)	81 mg/kg	78 mg/kg	3,85 %

6. Conclusión

Este trabajo presentó la construcción de un dispositivo para medir nitrógeno, fósforo y potasio del suelo. Ésta es una herramienta útil para los profesionales del área agronómica o propietarios de tierras, los cuales podrán monitorear el estado nutrimental y así abordar conclusiones que permitan una gestión más eficiente y sostenible de la fertilización.

El error relativo entre las mediciones del laboratorio y las del dispositivo, que varía entre el 4% y 14%, sugiere que el dispositivo presenta una discrepancia moderada en comparación con los resultados de referencia del laboratorio. Si el dispositivo se utiliza en aplicaciones donde una precisión del 4-14% es aceptable, entonces se considera adecuado para su uso.

La integración de tecnologías IoT junto con la utilización de bases de datos en la nube, paneles de visualización gráfica y la generación de reportes en Excel con el historial de las mediciones, mejora la accesibilidad y el manejo de los datos recolectados. Esto facilita tanto el acceso como la interpretación de la información.

Este trabajo ofrece varios puntos para mejorar. Entre ellos se incluye el desarrollo de una aplicación móvil, permitiendo tener los registros en el celular y/o tablet sin necesidad de una computadora. Otro punto de mejora es incluir otro tipo de mediciones, como temperatura, humedad, conductividad eléctrica y pH, así como la posibilidad de georreferenciar las mediciones mediante la incorporación de un GPS.

Un punto de mejora primordial para el dispositivo es la incorporación de una pantalla táctil. Ésto permitirá a los usuarios introducir las credenciales WiFi de manera directa, posibilitando el almacenamiento y la gestión de múltiples redes WiFi en lugar de estar limitado a una sola. Actualmente, la limitación radica en que el dispositivo solo puede conectarse a una red WiFi preconfigurada. La implementación de una pantalla táctil no solo mejorará la usabilidad y adaptabilidad del dispositivo, sino que también facilitará el proceso de configuración y conexión, haciéndolo más versátil y eficiente para los usuarios.

Como conclusión personal destaco que la realización de esta tesis me permitió adquirir nuevos conocimientos y reforzar otros. Por un lado, diseñé la carcasa del dispositivo en un software CAD para modelado 3D, diseñé la placa de circuito impreso y apliqué el protocolo de comunicación RS-485 en un entorno real. Además, consolidé mis conocimientos sobre tecno-

logías IoT, protocolos e interfaces de comunicación y construí mi primer dispositivo electrónico desde cero y de manera individual, lo que me obligó a reforzar los conceptos estudiados durante la carrera y ponerlos en práctica en un desarrollo interdisciplinario. Asimismo, este trabajo me permitió adquirir conocimientos sobre un tema relacionado a la agricultura del cual tenía escaso conocimiento.

Para finalizar es importante destacar que aunque el dispositivo es una herramienta valiosa, no reemplaza la importancia de los análisis químicos tradicionales realizados por expertos en el área. Sin embargo, sirve como complemento que ayuda a los usuarios a tener conocimiento de la fertilidad de la tierra.

7. Referencias

- Quiroga, A. y Bono, A., (2012). Manual de fertilidad y evaluación de suelos. Ediciones INTA. https://repositorio.inta.gob.ar/Manual_de_Fertilidad_Suelos.pdf
- Tisdale, S., Nelson, W. and Beaton, J., (1993). Soil fertility and fertilizers (5th ed.). Macmillan.
- Garrido, S., (1993). Interpretación de análisis de suelos. Rivadeneyra.
- Scherr, S., (1999). Soil Degradation. A Threat to Developing-Country Food Security by 2020? International Food Policy Research Institute.
- Cubilla, A., Wendling, A., Eltz, F., Amado, T. y Mielniczuk, J., (2012). Recomendaciones de fertilización bajo el sistema de siembra directa. Capeco.
- Stallings, W., (2004). Comunicaciones y Redes de Computadores (7ma ed.). Pearson.
- Boylestad, R., Nashelsky, L., (2009). Electrónica: teoría de circuitos y dispositivos electrónicos (10ma ed.). Pearson.
- Buyya, R., Dastjerdi, A., (2016). Internet of Things Principles and Paradigms. Elsevier.
- Seacord, R., (2013). Secure coding in C and C++ (2nd ed.). Pearson.
- Miranda, F. y García, L., (1994). Fertilidad y fertilización de suelos. Universidad Nacional Agraria.
- Red BPA, (2020). Recomendaciones para muestreo de suelos. <https://redbpa.org.ar/wp-content/uploads/2021/04/EP-RedBPA-MuestreoDe-Suelos.pdf>
- Asociación Internacional de la Industria de los Fertilizantes, (2002). Los fertilizantes y su uso. <https://www.fao.org/3/x4781s/x4781s.pdf>
- Andrade, F., Posse, G., Taboada, M. y Garra, D., (2017). Los desafíos de la agricultura argentina. Satisfacer las futuras demandas y reducir el impacto ambiental. Ediciones INTA. <https://>

repositorio.inta.gob.ar/bitstream/handle/20.500.12123/2149/INTA_CRBsAsSur_EEABalcarce_Andrade_FH_Desafios_agricultura_argentina.pdf

LoRa (2024). Documentación. <https://lora-alliance.org/about-lorawan/>

Sigfox (2024). Documentación. <https://www.sigfox.com/what-is-sigfox/>

NB-IoT (2024). Documentación. <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-nb-iot-y-como-funciona/>

Cloudflare (2024). ¿Qué es el modelo OSI?. <https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>

Afzal, S. (2016). I2C Primer: What is I2C? (Part 1). <https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html>

Weis, O., (2020). Protocolos de comunicación. Principales tipos de transferencia serie. <https://www.serial-port-monitor.org/types-of-serial-protocols/>

Weis O., (2021). Guía de la comunicación RS485. <https://www.eltima.com/es/article/rs485-communication-guide/>

INCIBE, (2023). Estudio de análisis de firmware en dispositivos industriales. https://www.incibe.es/INCIBE-CERT_GUIA_ANALISIS_DE_FIRMWARE_SCI_2023_v1.1.pdf

Secretaría de Tecnologías de la Información y las Comunicaciones. (s.f.). Internet de las Cosas. Ministerio de Modernización. <https://www.argentina.gob.ar/sites/default/files/paperbenchmarkinternacional-iot.pdf>

Gongora, K. y Benito, J., (2013). Diseño e implementación de sistema de carga para baterías LI-ION a dispositivo portátil de monitoreo y comunicación satelital para embarcaciones de flota menor en el sureste de México. Tecnológico Nacional de México. <https://rinacional.tecnm.mx/jspui/handle/TecNM/2771>

Elcontrol Energy (2006). Modbus RS485 serial protocol. Recuperado de <https://www.elcomponent.co.uk/wp-content/uploads/2018/05/VIP396-Modbus-Protocol.pdf>

Heltec Automation. (2022). WiFi Kit 32 Datasheet [Hoja de datos]. Recuperado de https://resource.heltec.cn/download/WiFi_Kit_32/WiFi%20Kit32.pdf

Vapex. (2016). Rechargeable Lithium Ion Battery 18650 Datasheet [Hoja de datos]. Recuperado de http://kinstarbattery.com/Uploads/Celldatasheet/20210218111024_LG_ICR18650HB4_Datasheet.pdf

Xinwoer. (s.f). Soil NPK sensor Datasheet [Hoja de datos]. Recuperado de <https://www.imiconsystem.com/Soil-NPK-sensor>

Maxim Integrated. (s.f). MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487 Datasheet [Hoja de datos]. Recuperado de <https://www.analog.com/MAX-Datasheet>

Aerosemi Technology. (s.f). MT3608 Datasheet [Hoja de datos]. Recuperado de <https://pdf1.alldatasheet.es/datasheet-pdf/download/1131968/ETC1/MT3608.html>

NanJing Top PowerASIC Corp. (s.f). TP4056 Datasheet [Hoja de datos]. Recuperado de <https://pdf1.alldatasheet.es/datasheet-pdf/1132405/ASIC/TP4056.html>

Fortune Semiconductor Corp. (2006). DW01-P One Cell Lithium-ion/Polymer Battery Protection IC Datasheet [Hoja de datos]. Recuperado de https://cdn.sparkfun.com/assets/learn_tutorials/2/5/1/DW01-P_DataSheet_V10.pdf

RZC Microelectronics Co., Ltd. (2012). 8205A Dual N-Channel MOSFET Datasheet [Hoja de datos]. Recuperado de <https://datasheetpdf.com/pdf-down/8/2/0/8205A>

Sino Wealth. (2013). SH1106 132X64 Dot Matrix OLED/PLED Segment/Common Driver with Controller Datasheet [Hoja de datos]. Recuperado de <https://www.pololu.com/file/0J1813/SH1106.pdf>

Maxim Integrated. (s.f). DS1307 64x8, Serial, I2C Real-Time Clock Datasheet [Hoja de datos]. Recuperado de <https://www.analog.com/media/en/technical-documentation/data-sheets/ds1307.pdf>

Diodes Incorporated. (2012). 74LVC125A quadruple 3-state buffers Datasheet [Hoja de datos]. Recuperado de <https://docs.rs-online.com/e4fb/0900766b815009ee.pdf>

Pro-Signal. (2016). Buzzer Datasheet [Hoja de datos]. Recuperado de <https://www.farnell.com/datasheets/2171929.pdf>

Unir (2023). La arquitectura IoT y sus usos en distintos sectores. <https://www.unir.net/ingenieria/revista/arquitectura-iot/>

BCR (2021). Demandantes y exportadores de fertilizantes en el mundo. <https://www.bcr.com.ar/es/mercados/investigacion-y-desarrollo/>

I2C Bus (2023). I2C – What’s That. <https://www.i2c-bus.org/>

Altium (2023). SPI vs I2C: Cómo elegir el mejor protocolo para tus chips de memoria. <https://resources.altium.com/es/p/spi-versus-i2c-how-choose>

InfluxData (2024). Documentation. <https://docs.influxdata.com/>

Grafana (2024). Technical Documentation. <https://grafana.com/docs/>

8. Anexo

8.1. Pinout de la placa de desarrollo

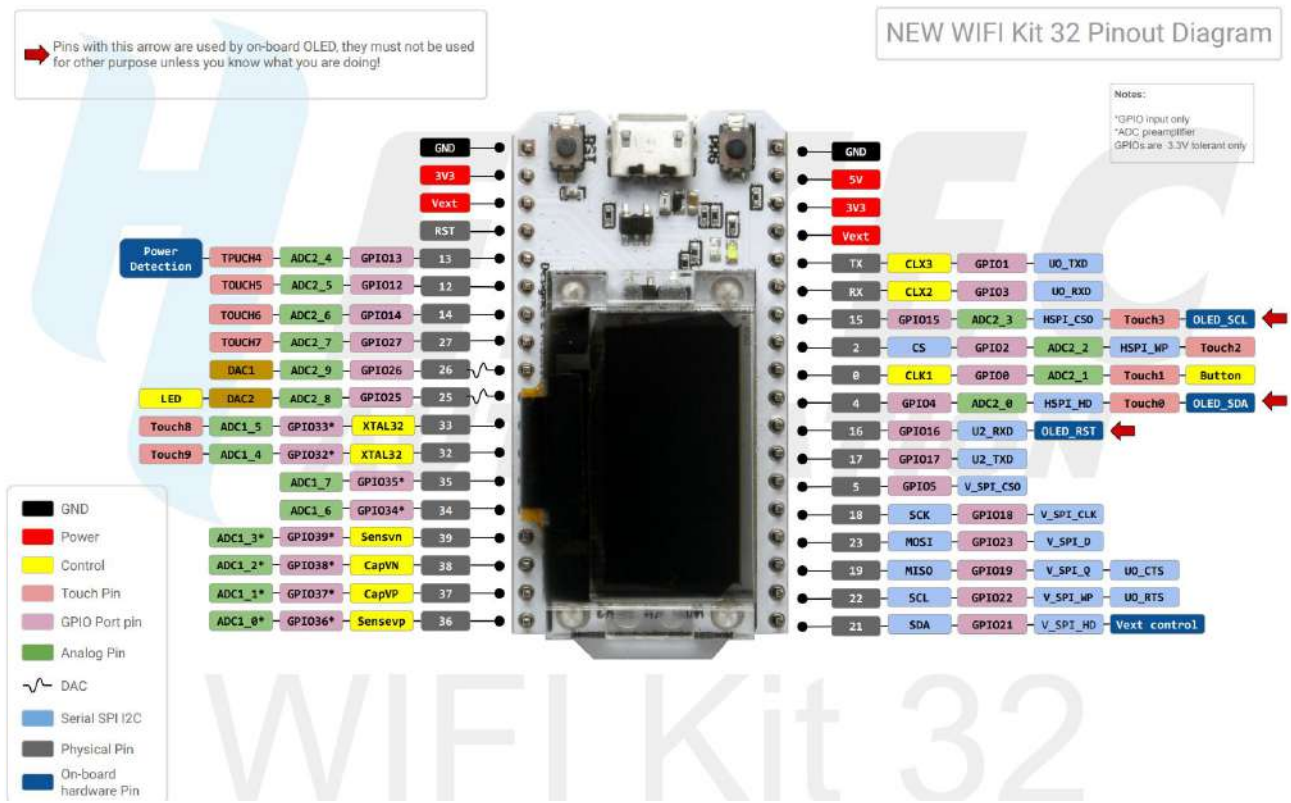


Figura 8.1: Pinout de la placa de desarrollo. Extraído de [ESP32 WiFi Kit Datasheet]

8.2. Estructura interna del MAX485

En la Figura 8.2 se denota con la letra D al conversor serie a RS-485. su entrada está conectada al pin DI y su salida a los pines A y B, por lo tanto, toda señal que se transmita al pin DI es transformada y transferida a los pines A y B.

Se denota con la letra R al conversor RS-485 a serie. Los pines A y B están conectados a la entrada del conversor y su salida al pin RO. Esto implica que la señal RS-485 que llegue por los pines A y B son transmitidas por el pin RO para que el microcontrolador las pueda recibir.

Como la salida de D y la entrada de R coinciden, solo es posible utilizar uno a la vez, o salida o entrada. Aquí entran en juego los pines DE y RE para establecer el modo emisor o receptor.

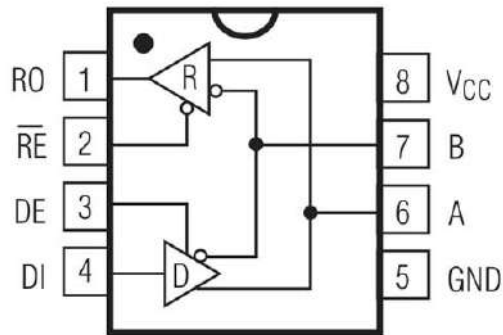


Figura 8.2: Estructura interna MAX485. Extraído de [MAX485 Datasheet]

8.3. Prototipos



Figura 8.3: Pantalla de introducir sensor en suelo.



Figura 8.4: Pantalla de espera.



Figura 8.5: Pantalla de retire el sensor del suelo.

8.4. Código

8.4.1. Función para subir los registros a InfluxDB

El algoritmo, que se muestra en Código 8.3, compara el campo *unixtime* de cada registro almacenado en la tarjeta de memoria externa con el valor del último registro que fue subido a la nube (*ultRegSubMemoria*). En el caso de que *unixtime* sea mayor, quiere decir que no fue subido y, por lo tanto, se envía a la nube.

Cuando se sube un registro a la nube correctamente, se almacena en la memoria interna del microcontrolador el valor *unixtime* de dicho registro. Para esto se utiliza la función que se muestra en Código 8.1.

```
1 void guardarUltReg(uint32_t valor){
2   EEPROM.write(0, (valor >> 0) & 0xFF);
3   EEPROM.write(1, (valor >> 8) & 0xFF);
4   EEPROM.write(2, (valor >> 16) & 0xFF);
5   EEPROM.write(3, (valor >> 24) & 0xFF);
6
7   EEPROM.commit();
8 }
```

Código 8.1: Función para almacenar el último registro subido a la nube.

La variable *ultRegSubMemoria* es de tipo `uint32_t` y obtiene el último registro que fue subido a la nube mediante la lectura desde la memoria interna del microcontrolador, como lo muestra el Código 8.2.

```

1  uint32_t leerUltReg(void){
2  uint32_t valor = 0;
3  valor |= (EEPROM.read(0) << 0);
4  valor |= (EEPROM.read(1) << 8);
5  valor |= (EEPROM.read(2) << 16);
6  valor |= (EEPROM.read(3) << 24);
7
8  return valor;
9  }

```

Código 8.2: Función para leer el último registro subido a la nube.

```

1  void subirRegistros(void) {
2  int startInd[100], endInd[100];
3  uint32_t start = 0, end = 0;
4
5  File file = SD.open(filename, FILE_READ);
6  if (!file) return;
7
8  String fileContent = file.readString();
9  file.close();
10
11  ultRegSubMemoria = leerUltReg();
12
13  StaticJsonDocument<100> doc;
14
15  while ((start = fileContent.indexOf('[', end)) != -1) {
16      end = fileContent.indexOf(']', start);
17      if (end == -1) break; // Si no se encuentra cierre de corchete, detener
18
19      String jsonSegment = fileContent.substring(start, end + 1); // Extrae el
20      ↪ segmento JSON entre corchetes
21
22      DeserializationError error = deserializeJson(doc, jsonSegment);
23      if (error) return;
24
25      JsonArray jsonArray = doc.as<JsonArray>();

```

```

25
26  for (JsonObject registro : jsonArray){
27      uint32_t UT = registro["unixtime"];
28      String fecha = registro["fecha"];
29      String hora = registro["hora"];
30      uint16_t n = registro["n"];
31      uint8_t p = registro["p"];
32      uint8_t k = registro["k"];
33
34      if (UT > ultRegSubMemoria) {
35          npkRegistro.clearFields();
36
37          npkRegistro.addField("N", n);
38          npkRegistro.addField("P", p);
39          npkRegistro.addField("K", k);
40          npkRegistro.addField("unixTimeStamp", UT);
41
42          if (!client.writePoint(npkRegistro)) {
43              Serial.print("InfluxDB escritura fallida: ");
44              Serial.println(client.getLastErrorMessage());
45          }
46          else {
47              guardarUltReg(UT);
48              ultRegSubMemoria = UT;
49          }
50      }
51  }
52  end++;
53 }
54 }

```

Código 8.3: Función para subir los registros a InfluxDB.

8.4.2. Script para el tratamiento de datos

En el Código 8.4 se muestra el código realizado en Python que recibe como entrada el archivo JSON de los registros y devuelve una planilla en Excel.

```
1 import json
2 import pandas as pd
3 from datetime import datetime
4 from openpyxl import load_workbook
5 from openpyxl.styles import Alignment, Border, Side, Font
6 from openpyxl.drawing.image import Image
7
8 archivoJson = 'registros.json'
9
10 fechaHoraActual = datetime.now().strftime('%d-%m-%y, %Hh %Mm')
11 nombreArchivoExcel = f'Historial - {fechaHoraActual}.xlsx'
12
13 data = []
14
15 with open(archivoJson, 'r') as file:
16     contenido = file.read()
17     registrosJson = contenido.split('][')
18
19     if registrosJson:
20         registrosJson[0] = registrosJson[0].rstrip('[')
21         registrosJson[-1] = registrosJson[-1].rstrip(']')
22
23     for registro in registrosJson:
24         try:
25             data.append(json.loads(registro))
26         except json.JSONDecodeError as e:
27             print(f"Error al decodificar JSON: {registro}. Error: {e}")
28
29 df = pd.DataFrame(data)
```

```

30
31 columnas = ['fecha', 'hora', 'n', 'p', 'k']
32 df = df[columnas]
33 df.columns = ['Fecha', 'Hora', 'N', 'P', 'K']
34 df.to_excel(nombreArchivoExcel, index=False, startrow=4, startcol=1)
35
36 wb = load_workbook(nombreArchivoExcel)
37 hoja = wb.active
38
39 imagenRuta = 'FSensLogo.png'
40 imagen = Image(imagenRuta)
41 imagen.anchor = 'C1'
42 hoja.add_image(imagen, imagen.anchor)
43
44 fuenteNegrita = Font(name='Abadi', bold=True)
45 bordeLado = Side(style='thin')
46 borde = Border(left=bordeLado, right=bordeLado, top=bordeLado, bottom=bordeLado)
47
48 filaEncabezado = 5
49 columnaInicial = 2
50 columnaFinal = len(df.columns) + columnaInicial
51
52 for columna in range(columnaInicial, columnaFinal):
53     celda = hoja.cell(row=filaEncabezado, column=columna)
54     celda.font = fuenteNegrita
55
56 fuente = Font(name='Abadi')
57 for fila in range(filaEncabezado, len(df) + filaEncabezado + 1):
58     for columna in range(columnaInicial, columnaFinal):
59         celda = hoja.cell(row=fila, column=columna)
60         celda.alignment = Alignment(horizontal='center', vertical='center')
61         celda.font = fuente
62         celda.border = borde

```

```

63
64 for columna in range(columnaInicial, columnaFinal):
65     celda = hoja.cell(row=filaEncabezado, column=columna)
66     celda.font = fuenteNegrita
67
68 wb.save(nombreArchivoExcel)
69
70 print(f"El archivo Excel se ha creado exitosamente: {nombreArchivoExcel}")

```

Código 8.4: Script para generar la planilla Excel.

8.5. Imágenes de FertSens



Figura 8.6: Pantalla de inicio - FertSens



Figura 8.7: Pantalla de introducir sensor en suelo - FertSens



Figura 8.8: Pantalla de espera - FertSens



Figura 8.9: Pantalla de retirar el sensor - FertSens



Figura 8.10: Pantalla de un registro - FertSens



Figura 8.11: Pantalla de falta de tarjeta SD - FertSens