

**UNIVERSIDAD PRIVADA ANTENOR ORREGO  
FACULTAD DE INGENIERÍA**

**PROGRAMA DE ESTUDIO DE INGENIERÍA ELECTRÓNICA**



**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

---

Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los  
cultivos del distrito de San José – Cartavio

---

Línea de Investigación: **Sistemas inteligentes**

Autores:

Valeriano Ulloa, Jorge Manuel

Vereau Campos, Lucas

Jurado Evaluador:

Presidente : Azabache Fernández, Filiberto

Secretario : Cerna Sánchez Eduardo, Elmer

Vocal : Llanos Leon Lenin Humberto

Asesor:

Alvarado Rodríguez, Luis Enrique

Código Orcid: <https://orcid.org/0000-0001-6444-2922>

Fecha de Sustentación: 2025/01/24

Trujillo – Perú  
2025

# Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los cultivos del distrito de San José – Cartavio.

por Jorge Manuel Valeriano Ulloa

Fecha de entrega: 21-ene-2025 11:58a.m. (UTC-0500)  
Identificador de la entrega: 2568389029  
Nombre del archivo: INFORME\_DE\_TESIS\_FINAL.docx (6.94M)  
Total de palabras: 21674  
Total de caracteres: 121074

## Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los cultivos del distrito de San José – Cartavio.

### INFORME DE ORIGINALIDAD

8% INDICE DE SIMILITUD    8% FUENTES DE INTERNET    % PUBLICACIONES    % TRABAJOS DEL ESTUDIANTE

### FUENTES PRIMARIAS

1	repositorio.unp.edu.pe Fuente de Internet	1%
2	hdl.handle.net Fuente de Internet	1%
3	repositorio.upao.edu.pe Fuente de Internet	1%
4	core.ac.uk Fuente de Internet	1%
5	kupdf.net Fuente de Internet	1%
6	learningdata.hubiberiaagrotech.eu Fuente de Internet	1%
7	ri.agro.uba.ar Fuente de Internet	1%
8	www.luisllamas.es Fuente de Internet	1%
9	repositorio.untels.edu.pe Fuente de Internet	1%

		1%
10	www.kusitest.pe Fuente de Internet	1%

Excluir citas    Apagado    Excluir coincidencias < 1%  
Excluir bibliografía    Activo

## Jurado de sustentación Oral



---

Azabache Fernandez Filiberto:  
CIP N° 97916  
Presidente



---

Cerna Sanchez Eduardo Elmer:  
CIP N° 80252  
Secretario



---

Llanos Leon Lenin Humberto  
CIP N° 139213  
Vocal

Entregado el:

Aprobado por:



---

Valeriano Ulloa Jorge Manuel:  
DNI: 70775934



---

Vereau Campos  
Lucas: DNI:  
45701065



---

Alvarado Rodríguez Luis Enrique:  
Asesor de Tesis

UNIVERSIDAD PRIVADA ANTENOR ORREGO  
FACULTAD DE INGENIERÍA

PROGRAMA DE ESTUDIO DE INGENIERÍA ELECTRÓNICA



TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO

---

Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los  
cultivos del distrito de San José – Cartavio

---

Línea de Investigación: **Sistemas inteligentes**

Autores:

Valeriano Ulloa, Jorge Manuel

Vereau Campos, Lucas

Jurado Evaluador:

Presidente : Azabache Fernández, Filiberto

Secretario : Cerna Sánchez Eduardo, Elmer

Vocal : Llanos Leon Lenin Humberto

Asesor:

Alvarado Rodríguez, Luis Enrique

Código Orcid: <https://orcid.org/0000-0001-6444-2922>

Fecha de Sustentación: 2025/01/24

Trujillo – Perú

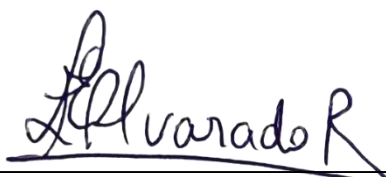
2025

## DECLARACIÓN DE ORIGINALIDAD

Yo, Luis Enrique Alvarado Rodríguez, docente del Programa de Estudio de Pregrado de la Universidad Privada Antenor Orrego, asesor de la tesis titulada “Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los cultivos del distrito de San José – Cartavio”, de los autores Jorge Manuel Valeriano Ulloa y Lucas Vereau Campos

- El mencionado documento tiene un índice de puntuación de similitud del 8%. Así lo consigna el reporte de similitud emitido por el software Turnitin el día 21 de enero del 2025.
- He revisado con detalle dicho reporte de la tesis “Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los cultivos del distrito de San José – Cartavio” y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las normas establecidas por la Universidad.

Ciudad y fecha: Trujillo 21/01/2025



Alvarado Rodríguez Luis Enrique:

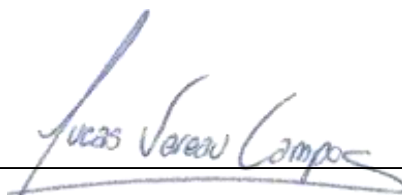
DNI: 43344790

ORCID: 0000-0001-6444-2922



Jorge Manuel Valeriano Ulloa:

DNI: 70775934



Lucas Vereau Campos:

DNI: 45701065

## **DEDICATORIA**

### **A mis padres, Ángel y Angélica**

Por ser mi guía y fortaleza en cada paso. Quienes, con su profundo amor, esfuerzo y fe en mi persona hicieron posible llegar a esta etapa de mi vida. Los amo mucho.

### **A mi hermanita, Angélica**

Por su alegría y ocurrencias me recuerdan el valor de la familia y me dan las fuerzas para seguir creciendo.

### **A mis tías y tíos**

Que siempre estuvieron presentes y ayudándome a mejorar cada día.

### **A mis abuelos: Jorge Ulloa, Juana García**

Por ser ejemplo de superación y lucha constante.

Jorge Manuel Valeriano Ulloa

### **A mi madre, Nelly Ruth Campos Kong**

Aunque ya no estés físicamente conmigo, tu presencia vive en mí todos los días. Gracias por mostrarme, con tu vida, el verdadero significado de la fortaleza y el amor incondicional. Esta tesis es para ti, en honor a todo lo que me diste y a la persona que me ayudaste a ser.

### **A mis tías: Carmen Campos Kong, Maritza Campos Kong y Gladys Vereau García**

Quiero dedicar este trabajo especialmente a ustedes, que han sido un pilar fundamental en mi vida. Carmen, Maritza y Gladys, con su amor, generosidad y apoyo incondicional, me han brindado el mismo amor y cuidado que mi madre siempre me dio. Les agradezco de corazón por estar siempre a mi lado.

Lucas Vereau Campos

## **AGRADECIMIENTOS**

En primer lugar, quiero agradecer a Dios por guiarme y llenarme de sabiduría. Agradecer a mi familia, en especial a mis padres Angélica Ulloa y Ángel Valeriano, por su amor incondicional, su apoyo moral y su fe en mi en los momentos más difíciles, siendo los pilares de este logro. A mi hermana Angélica Valeriano, quien me brindo su tiempo para escucharme y apoyarme, y a mis abuelos y tíos, quienes están conmigo apoyándome y brindándome su cariño.

A mis amigos Daniel Chaname y Lucas Vereau que estuvieron conmigo en los momentos de estrés y alegría durante todo este trayecto. Gracias por su compañía, su confianza y apoyo, fueron invaluableles.

Jorge Manuel Valeriano Ulloa

Quiero expresar mi más sincero agradecimiento a todas las personas que, de una u otra forma, han sido parte fundamental en este proceso y me han brindado su apoyo y cariño.

A mi familia, por su amor, su paciencia y por ser el pilar que me ha sostenido en cada paso de mi vida. A mis abuelos, que siempre con sus sabias enseñanzas y su ejemplo de vida, me han inspirado a seguir adelante con determinación.

A mis amigos, quienes han sido una fuente constante de motivación y alegría. En especial, quiero agradecer a Jorge Valeriano Ulloa, mi compañero de tesis, por su dedicación, su compañerismo y por haber compartido este camino lleno de desafíos y aprendizajes. También a Daniel Isaac Chanamé Díaz, quien, además de ser un gran amigo, siempre ha estado presente a lo largo de nuestra carrera, brindándonos su apoyo y compartiendo momentos inolvidables.

A todos ustedes, por estar siempre a mi lado, por su apoyo constante y por hacer de este proceso una experiencia más significativa. Este logro es tan suyo como mío.

Lucas Vereau Campos

## RESUMEN

El presente trabajo tiene como objetivo el diseño de un sistema IoT (Internet de las cosas) para el monitoreo de la salinidad del suelo en los cultivos del distrito de San José - Cartavio, con el fin de mejorar la gestión agrícola y optimizar el uso de los recursos en esta zona rural. La salinidad del suelo es un factor crítico para la salud de los cultivos, y el control eficiente de este parámetro es esencial para evitar la reducción en la productividad agrícola.

El sistema propuesto se basa en la medición de conductividad eléctrica del suelo utilizando el sensor CWT-Soil-THC-S, el cual es compatible con el módulo ESP32 gracias a la conexión intermediaria de un Max485 para la transmisión de datos. Los datos recolectados son enviados mediante MQTT a una Raspberry Pi, que actúa como intermediario para distribuir la información al servicio de Ubidots. Este último se utiliza como base de datos en la nube aprovechando su capacidad de almacenar datos y permitiendo su acceso remoto a través de un dashboards, diseñado para visualizar el historial de mediciones.

La implementación del sistema incluye el diseño del prototipo electrónico, la configuración de la red de comunicación y la validación del funcionamiento del sistema en condiciones reales de operación. Como resultado, se logró un sistema de monitoreo eficiente con lecturas 2 segundos y actualizaciones cada 30 minutos a través de la plataforma Ubidots.

Palabras clave: Internet de las Cosas (IoT), Monitoreo ambiental, Gestión agrícola, Optimización de recursos, Automatización agrícola, Sensores inteligentes, Agricultura de precisión, Smart Farming, Tecnología aplicada a la agricultura, Innovación tecnológica en el agro

## **ABSTRACT**

The objective of this work is the design of an IoT (Internet of Things) system for monitoring soil salinity in crops in the San José - Cartavio district, with the aim of improving agricultural management and optimizing the use of resources in this rural area. Soil salinity is a critical factor for crop health, and efficient control of this parameter is essential to prevent a reduction in agricultural productivity.

The proposed system is based on measuring the electrical conductivity of the soil using the CWT-Soil-THC-S sensor, which is compatible with the ESP32 module through the intermediary connection of a Max 485 for data transmission. The collected data is sent via MQTT to a Raspberry Pi, which acts as an intermediary to distribute the information to the Ubidots service. The latter is used as a cloud database, leveraging its data storage capabilities and enabling remote access through a dashboard designed to display the measurement history.

The system implementation includes the design of the electronic prototype, the configuration of the communication network, and the validation of the system's functionality under real operating conditions. As a result, an efficient monitoring system was achieved, with reading every 2 seconds and updates every 30 minutes through the Ubidots platform.

Keywords: Internet of Things (IoT), Environmental monitoring  
Agricultural  
management, Resource optimization, Agricultural automation  
Smart

sensors, Precision agriculture, Smart farming, Technology applied to  
agriculture  
Technological innovation in agriculture

## **PRESENTACIÓN**

Señores miembros del Jurado:

De conformidad con lo estipulado en el Reglamento de Grados y Títulos de la Universidad Privada Antenor Orrego, ponemos a su disposición el informe de tesis titulado: “Diseño de un sistema IoT para el monitoreo de salinidad del suelo en los cultivos del distrito de San José – Cartavio.” Para que sea revisado y evaluado y de ser aprobado pueda ser defendido oralmente para optar el título profesional de Ingeniero Electrónico.

De antemano, nos excusamos de los errores involuntarios en que se hubiera incurrido en el desarrollo y redacción del misma, esperando del honorable jurado un justo dictamen.

Valeriano Ulloa Jorge Manuel

Vereau Campos Lucas

## Tabla de contenidos

DEDICATORIA .....	VI
AGRADECIMIENTOS .....	VII
RESUMEN.....	VIII
ABSTRACT.....	IX
PRESENTACIÓN .....	X
TABLA DE CONTENIDOS .....	XI
INDICE DE TABLAS .....	XIII
INDICE DE GRAFICOS.....	XIV
<b>I. INTRODUCCIÓN .....</b>	<b>- 16 -</b>
1.1. PROBLEMA DE INVESTIGACIÓN.....	- 17 -
1.2. OBJETIVOS .....	- 17 -
1.2.1. <i>Objetivo General</i> .....	- 17 -
1.2.2. <i>Objetivos específicos</i> .....	- 17 -
1.3. JUSTIFICACIÓN DEL ESTUDIO .....	- 17 -
<b>II. MARCO DE REFERENCIA .....</b>	<b>- 18 -</b>
2.1. ANTECEDENTES .....	- 18 -
2.2. MARCO TEÓRICO.....	- 21 -
2.1.1. <i>Botánica</i> .....	- 21 -
2.1.2. <i>Conductividad Eléctrica del suelo</i> .....	- 23 -
2.1.3. <i>Tecnologías de control: Clasificación:</i> .....	- 27 -
2.1.4. <i>Internet of things (IoT)</i> .....	- 30 -
2.1.5. <i>Redes inalámbricas</i> .....	- 31 -
2.1.6. <i>Componentes y plataformas</i> .....	- 34 -
2.3. MARCO CONCEPTUAL.....	- 37 -
2.4. SISTEMA DE HIPÓTESIS.....	- 38 -
2.5. VARIABLES E INDICADORES .....	- 38 -
<b>III. METODOLOGÍA EMPLEADA.....</b>	<b>- 40 -</b>
3.1. TIPO Y NIVEL DE INVESTIGACIÓN .....	- 40 -
3.1.1. <i>De acuerdo con la orientación o finalidad</i> .....	- 40 -
3.1.2. <i>De acuerdo con su alcance temporal</i> .....	- 40 -
3.1.3. <i>De acuerdo con su carácter</i> .....	- 40 -
3.1.4. <i>De acuerdo con la técnica de contrastación</i> .....	- 40 -
3.2. POBLACIÓN Y MUESTRA DE ESTUDIO .....	- 40 -

3.2.1.	<i>Población</i> .....	- 40 -
3.2.2.	<i>Muestra</i> .....	- 40 -
3.3.	DISEÑO DE INVESTIGACIÓN.....	- 41 -
3.4.	TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN .....	- 41 -
3.5.	PROCESAMIENTO Y ANÁLISIS DE DATOS.....	- 41 -
3.5.1.	<i>Identificar los sistemas de medición de conductividad eléctrica del suelo existentes en Trujillo</i> .....	- 41 -
3.5.2.	<i>Diseñar un prototipo electrónico que lea la información de la conductividad eléctrica del suelo usando un sensor de CE que envíe la información a una base de datos en la nube</i> .....	- 46 -
3.5.3.	<i>Diseñar el dashboards para mostrar el registro histórico de los datos del sensor de suelo.</i> - 101 -	
3.5.4.	<i>Comparar las especificaciones del sistema con las del mercado actual</i> .....	104
<b>IV.</b>	<b>PRESENTACIÓN DE RESULTADOS</b> .....	<b>106</b>
4.1.	PROPUESTA DE INVESTIGACIÓN.....	106
4.2.	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	106
4.3.	DOCIMASIA DE HIPÓTESIS .....	116
<b>V.</b>	<b>DISCUSIÓN DE LOS RESULTADOS</b> .....	<b>117</b>
	<b>CONCLUSIONES</b> .....	<b>118</b>
	<b>RECOMENDACIONES</b> .....	<b>120</b>
	<b>REFERENCIAS</b> .....	<b>121</b>
	<b>ANEXO 1</b> .....	<b>125</b>
	<b>ANEXO 2</b> .....	<b>126</b>
	<b>ANEXO 3</b> .....	<b>127</b>
	<b>ANEXO 4</b> .....	<b>127</b>
	<b>ANEXO 5</b> .....	<b>128</b>
	<b>ANEXO 6</b> .....	<b>128</b>
	<b>ANEXO 7</b> .....	<b>129</b>
	<b>ANEXO 8</b> .....	<b>129</b>
	<b>ANEXO 9</b> .....	<b>131</b>

## INDICE DE TABLAS

TABLA 1: CLASIFICACIÓN DEL SUELO SEGÚN SU SALINIDAD.....	- 24 -
TABLA 2: CLASIFICACIÓN DEL CONTENIDO DE SAL EN EL SUELO POR LA FAO.....	- 24 -
TABLA 3: ESPECIFICACIONES TÉCNICAS DEL SENSOR DE EJEMPLO RK520-02 .....	- 26 -
TABLA 4: COMPARACIÓN DE TECNOLOGÍAS INALÁMBRICAS .....	- 33 -
TABLA 5: VARIABLE DEPENDIENTE .....	- 38 -
TABLA 6: VARIABLE INDEPENDIENTE.....	- 39 -
TABLA 7: ESPECIFICACIONES TÉCNICAS DEL SENSOR HACH SENSION 5 .....	- 42 -
TABLA 8: ESPECIFICACIONES TÉCNICAS DEL SENSOR HANNA, MODELO HI8424 .....	- 43 -
TABLA 9: ESPECIFICACIONES TÉCNICAS DEL SENSOR COND 3110 SET 1 .....	- 45 -
TABLA 10: CUADRO COMPARATIVO DE ESP32 BY ESPRESSIF SYSTEMS .....	- 49 -
TABLA 11: CUADRO COMPARATIVO DE RASPBERRY PI 3B+, 4B, 400 Y 5 .....	- 53 -
TABLA 12: CUADRO COMPARATIVO DE SENSORES DE CONDUCTIVIDAD ELÉCTRICA DEL SUELO... - 55 -	
TABLA 13: COMPARACIÓN DE LAS ESPECIFICACIONES TÉCNICAS DEL SISTEMA CON LAS DEL MERCADO ACTUAL .....	104
TABLA 14: REGISTRO DE LOS VALORES DE LA LECTURA DEL SENSOR THC-S .....	109
TABLA 15: ANÁLISIS DE ERROR PORCENTUAL.....	114
TABLA 16: VALORES AJUSTADOS Y ERROR PORCENTUAL CON OFFSET = 130.....	114

## INDICE DE GRAFICOS

ILUSTRACIÓN 1: ESTRUCTURA DE LA RAÍZ DE UNA PLANTA.....	- 22 -
ILUSTRACIÓN 2: ARQUITECTURA CENTRALIZADA .....	- 27 -
ILUSTRACIÓN 3: ARQUITECTURA DESCENTRALIZADA .....	- 28 -
ILUSTRACIÓN 4: ARQUITECTURA DISTRIBUIDA .....	- 28 -
ILUSTRACIÓN 5:ARQUITECTURA HIBRIDA/MIXTA.....	- 29 -
ILUSTRACIÓN 6: CLASIFICACIÓN DE LAS REDES INALÁMBRICAS .....	- 31 -
ILUSTRACIÓN 7: ARQUITECTURA DE PUBLICACIÓN/SUSCRIPCIÓN DE MQTT .....	- 35 -
ILUSTRACIÓN 8: DESCRIPCIÓN GENERAL DEL SISTEMA DE MONITOREO .....	- 46 -
ILUSTRACIÓN 9: DIAGRAMA DE FLUJO DEL SISTEMA IOT .....	- 47 -
ILUSTRACIÓN 10: PARÁMETROS DE CONFIGURACIÓN Y CALIBRACIÓN DEL SENSOR THC-S..	- 57 -
ILUSTRACIÓN 11: FRONTAL DEL CIRCUITO DE LECTURA Y ENVÍO DE DATOS .....	- 61 -
ILUSTRACIÓN 12: POSTERIOR DEL CIRCUITO DE LECTURA Y ENVÍO DE DATOS .....	- 61 -
ILUSTRACIÓN 13: DISEÑO DE LA BASE DE LA CARCASA DEL CIRCUITO DE LECTURA.....	- 87 -
ILUSTRACIÓN 14: DISEÑO DE LA TAPA DE LA CARCASA DEL CIRCUITO DE LECTURA .....	- 88 -
ILUSTRACIÓN 15: CONFIGURACIÓN DE LAS PROPIEDADES DEL NODO MQTT.....	- 90 -
ILUSTRACIÓN 16: CONFIGURACIÓN DEL NODO MQTT-BROKER.....	- 91 -
ILUSTRACIÓN 17: PROGRAMACIÓN DEL NODO FUNCTION .....	- 92 -
ILUSTRACIÓN 18: CONFIGURACIÓN DEL NODO EXEC .....	- 93 -
ILUSTRACIÓN 19: DIAGRAMA DE FLUJO DE DATOS EN NODE-RED .....	- 94 -
ILUSTRACIÓN 20: DISPOSITIVO CREADO EN UBIDOTS .....	- 102 -
ILUSTRACIÓN 21: VARIABLES DEL SENSOR EN UBIDOTS.....	- 102 -
ILUSTRACIÓN 22: DASHBOARD FINAL DEL SISTEMA DE MONITOREO.....	- 103 -
ILUSTRACIÓN 23: SISTEMA DE CAPTURA DE DATOS DE LA CONDUCTIVIDAD ELÉCTRICA DEL SUELO.....	106
ILUSTRACIÓN 24: PROTOTIPO FINAL DEL CIRCUITO DE LECTURA.....	107
ILUSTRACIÓN 25: PRUEBA DE LABORATORIO CON EL SENSOR COND 3110 .....	108
ILUSTRACIÓN 26: PRUEBA 2 EN LABORATORIO CON EL SENSOR COND 3110.....	108
ILUSTRACIÓN 27: DIAGRAMA DE FLUJO DE DATOS EN NODE-RED .....	110
ILUSTRACIÓN 28: PROGRAMACIÓN EN PYTHON PARA EL ENVÍO A UBIDOTS .....	111
ILUSTRACIÓN 29: DASHBOARD FINAL DEL SISTEMA DE MONITOREO .....	111
ILUSTRACIÓN 30: PROFUNDIDAD RECOMENDADA PARA INSTALACIÓN DEL SENSOR .....	112
ILUSTRACIÓN 31: INSTALACIÓN EN LA SUPERFICIE DEL SUELO .....	113
ILUSTRACIÓN 32: RELACIÓN TEMPORAL ENTRE LA CONDUCTIVIDAD ELÉCTRICA Y LA HUMEDAD DEL SUELO .....	115
ILUSTRACIÓN 33: SENSOR COND 3110 SET 1.....	125

ILUSTRACIÓN 34: PROTOTIPO COMPLETO DEL SISTEMA DE MONITOREO DE LA CONDUCTIVIDAD ELÉCTRICA DEL SUELO .....	126
ILUSTRACIÓN 35: PROTOTIPO FINAL DEL CIRCUITO DE LECTURA .....	126
ILUSTRACIÓN 36: TERRENO DE MUESTREO.....	127
ILUSTRACIÓN 37: PROFUNDIDAD DE INSTALACIÓN .....	127
ILUSTRACIÓN 38: INSTALACIÓN DEL SENSOR THC-S .....	128
ILUSTRACIÓN 39: INSTALACIÓN DEL PROTOTIPO .....	128
ILUSTRACIÓN 40: INSTALACIÓN DEL RASPBERRY PI Y WI-FI .....	129
ILUSTRACIÓN 41: PLANO DE LA BASE DE LA CARCASA DEL CIRCUITO DE LECTURA A ESCALA 1:2 .....	130
ILUSTRACIÓN 42: TAPA DE LA CARCASA DEL CIRCUITO DE LECTURA DEL SENSOR THC-S.....	131

## I. Introducción

En septiembre del 2015, en la Cumbre De Las Naciones Unidas, se acordó un programa denominado “Objetivos de Desarrollo Sostenible (ODS) -17”, donde el objetivo número 15 es el de “Proteger, restablecer y promover el uso sostenible de los ecosistemas terrestres, gestionar sosteniblemente los bosques, luchar contra la desertificación, detener e invertir la degradación de las tierras y detener la pérdida de biodiversidad en América Latina y el Caribe” (NACIONES UNIDAS, 2023).

En los cultivos los factores que imponen un déficit hídrico son la temperatura ambiental, la insolación y la salinidad. (Benítez Burraco, 2019). El término “salt-affected soil” hace referencia a los suelos donde las sales interfieren con el crecimiento normal de las plantas (suelos salinos, salino-sódico y sódicos) y dependiendo del tipo, la cantidad de sodio y la alcalinidad de cada suelo afectado tendrá características únicas, lo que determinara su forma en que se pueden gestionar. (FAO, 2023)

A nivel mundial el área total de suelos salinos era de 397 millones de hectáreas y el de suelos sódicos de 434 millones de hectáreas. Del entonces 230 millones de hectáreas de riego, 45 millones de hectáreas (19.5%) eran suelos afectados por la sal: y de los casi 1500 millones de hectáreas de agricultura de secano, 32 millones (2.1%) eran suelos afectados por la sal. (FAO, 2023)

En la región de Latinoamérica el área total de suelo cultivable y no cultivable es de 2038.6 millones de hectáreas, donde el suelo salino equivalen a unas 60.5 millones de hectáreas (3%) y el suelo sódico a 50.9 millones de hectáreas (2.5%). (FAO, 2023)

En el Perú se identificaron alrededor de 300 mil hectáreas de riego con problemas de salinización, 150 mil hectáreas con altos niveles de salinidad. En particular, la salinidad del suelo es un problema que afecta al 40% del área agrícola total de la costa peruana aproximadamente. (Raúl S. Lavado, 2021)

Dentro de las ciudades cerca de la costa peruana se encuentra Cartavio en donde se encuentra un pueblo llamado San José, pequeño pueblo agricultor de frutas y verduras el cual no cuenta con un monitoreo constante de la salinidad del suelo. A razón de ello, en esta investigación se plantea el diseño de un dispositivo electrónico para recopilar los parámetros de salinidad del suelo el cual permita a los agricultores tomar medidas de prevención, tomando en cuenta los criterios dados por la FAO.

## **1.1. Problema de investigación**

¿Cómo medir los niveles de salinidad del suelo en los sembradíos del distrito de San José – Cartavio?

## **1.2. Objetivos**

### **1.1.1. Objetivo General**

Diseñar un sistema electrónico IoT de monitoreo de los parámetros de salinidad del suelo en el distrito San José – Cartavio.

### **1.1.2. Objetivos específicos**

- Identificar los sistemas de medición de conductividad eléctrica del suelo existentes en Trujillo.
- Diseñar un prototipo electrónico que lea la información de la conductividad eléctrica del suelo usando un sensor de conductividad eléctrica y envíe la información a una base de datos en la nube.
- Diseñar el Dashboard para mostrar el registro histórico de los datos del sensor.
- Comparar las especificaciones del sistema con las del mercado actual.

## **1.3. Justificación del estudio**

El estudio de la salinidad del suelo es un factor ambiental de suma importancia que afecta la calidad y productividad de los suelos agrícolas. En el ámbito de la ingeniería electrónica estudiar la salinidad del suelo cobra relevancia debido a la necesidad de desarrollar tecnologías y nuevos

sistemas de monitoreo que nos permitan así mitigar los efectos negativos que trae el exceso de salinidad en los sembradíos.

Los avances tecnológicos en sensores actualmente nos ofrecen una gran oportunidad para el desarrollo de soluciones innovadoras que permitirán al ser humano detectar, monitorear y también gestionar la salinidad del suelo de manera eficiente y casi precisa.

Por lo tanto, el estudio de la salinidad del suelo no solo resulta relevante para la sostenibilidad ambiental y la seguridad alimentaria, sino también representa un área de investigación interdisciplinaria que beneficia tanto al sector agrícola como al desarrollo de tecnologías aplicada a la gestión y control de los recursos naturales.

## **II. Marco de Referencia**

### **2.1. Antecedentes**

A continuación, para validar nuestra investigación se presentan los antecedentes de diversos estudios con referencia al tema elegido para este proyecto.

Isabel Ortega (2022), en su investigación titulada “IoT y agricultura 4.0: Control de cultivos en función de la conductividad eléctrica del suelo”, presenta el diseño e implementación de un escenario empleando el protocolo MQTT, junto a la herramienta Mosquito, en el emulador de redes Mininet-wifi mediante el lenguaje de programación Python para la lectura del sensor de conductividad y así generar alarma cuando supere los 1.2 dS/m. Como resultado consiguen crear un programa que pone en alerta al agricultor para que pueda actuar sobre una zona del cultivo afectada por un exceso de salinidad del suelo y así evitar costes innecesarios. El aporte a nuestro trabajo se basa en ver la funcionalidad de un sistema de control de cultivos en función de la conductividad eléctrica del suelo y su implementación, trabajando sensores, emuladores y Python, además, la IoT facilita las labores diarias a aquellos que trabajan en el sector de la agricultura.

Iseas Mariano Santiago (2023), en su investigación titulada “Evaluación de propiedades físicas y químicas en el suelo regados y relaciones con la conductividad eléctrica aparente en un establecimiento agrícola del noreste bonaerense” presenta el análisis de la eficiencia de sensores geofísicos proximales para la detección de la variabilidad espacial en las propiedades intrínsecas del suelo y evaluar los cambios producidos por el riego complementario en algunas propiedades edáficas. Como resultados demostraron que las zonas cubiertas por riego presentan mayores valores de CEap, disminuyendo abruptamente en zonas de secano; la exploración de CEap resulta apropiada para evaluar cambios en la salinidad y sodicidad del suelo debido al riego complementario; la CEap tiene mayor correlación con la salinidad del suelo y no con la humedad; existe una baja relación entre la porosidad y la CEap en la correlación “general”; y la aplicación de sensores geofísicos en el estudio del suelo y en particular el sensor electromagnético EMI, permite relevar la CEap del suelo en forma sencilla, rápida y eficaz. El aporte a nuestra investigación es las relaciones encontradas entre el CEap y el pH, Ces y PSI y que en menor medida lo es para el contenido hídrico volumétrico y la densidad aparente.

Ruben Herrera (2021) en su investigación titulada “Implementación de un sistema automático de riego agrícola basado en Raspberry pi y plataforma IoT artik cloud” presenta la implementación de un sistema autónomo de riego utilizando el sensor DT11 para la medición de la temperatura y humedad ambiental y el sensor FC-28 para la medición de la humedad del suelo y junto a 2 relés para la activación de una válvula para el paso del agua de riego de los campos agrícolas. El sistema se basó en la raspberry pi y la plataforma UBIDOTS, los sensores conectados cada uno a una raspberry pi cuyos datos leídos de humedad ambiental, temperatura y humedad del suelo son enviados a la plataforma UBIDOTS vía WIFI donde son almacenados y visualizados mediante graficas donde también mediante la plataforma se permite la toma de decisión de apertura y cierre de la válvula. El campo de acción está dividido en 3 zonas. El trabajo concluyo en el logro del diseño y simulación de un sistema automático de riego basado en la raspberry pi y UBIDOTS, el diseño de un sistema IoT, el diseño

de la arquitectura de red del dispositivo, el diseño de un dispositivo con raspberry pi 3 por zonas, la medición con el sensor DHT11, el control de la humedad del suelo usando el sensor FC28 y un potenciómetro, se logró el control manual de apertura de la válvula de agua desde la plataforma UBIDOTS, la implementación de la comunicación para conectar con la plataforma web UBIDOTS, la implementación del panel de visualización en la plataforma UBIDOTS para visualizar el comportamiento de las variables medidas. El aporte a la investigación es el uso de la miniconsola RASPBERRY PI 3 y la plataforma UBIDOTS para el control y visualizador de los datos obtenidos por los sensores.

Paolo Tovar, Jesús Hernán, et al (2022) en su investigación titulada “Propuesta de un sistema de medición de conductividad eléctrica aparente y humedad del suelo para pequeños agricultores” presenta una puesta de sistema de medición de suelo a partir del diseño de dos sensores de bajo costo: sensor de conductividad aparente y sensor de humedad, los cuales tienen la particularidad que sus características se pueden asociar a las propiedades físicas del suelo para la toma de decisiones. Los sensores fueron desarrollados mediante 2 técnicas: el método de Wenner, para la CEa y la medición de la impedancia eléctrica, para la identificar la permitividad dieléctrica asociada con el contenido de agua del suelo. Se hizo uso de una raspberry pi 3 para el control del sistema a través de un software desarrollado que permitió correlacionar los datos con su geolocalización. Como resultado el diseño implicó la elección entre diferentes alternativas para la medición de CEa y humedad del suelo, implementando aquellas con mayor relación costo-efectividad, demostrando que el sistema tiene bajo costo y es una alternativa para que los pequeños agricultores accedan a estas tecnologías y logren un mejor rendimiento en sus cultivos, estableciendo así la importancia que tienen las ICT para el estudio de suelo. Además, el dispositivo presenta la capacidad de realizar gráfico en tiempo real de las variables medidas en suelo y almacena los datos históricos en archivos Excel. El aporte a nuestra investigación es el estudio en la creación de los sensores de conductividad eléctrica y humedad del suelo implementando técnicas de bajo costo,

logrando brindar un diseño funcional para el diagnóstico del suelo, junto a la raspberry pi 3 y al módulo ADC ADS1115.

## **2.2. Marco teórico**

### **2.2.1. Botánica**

Entendida como el campo de la biología que estudia los vegetales, es una disciplina que combina los conocimientos adquiridos durante su larga tradición de avances que se producen a diario, consiguiendo ser una ciencia de plena vigencia, socialmente necesaria y con múltiples campos de actuación y desarrollo. (UAM, s.f.)

Las plantas son organismos autótrofos, que construyen su cuerpo con la energía del sol, el CO<sub>2</sub> de la atmósfera y el agua y las sales minerales del suelo. Y mediante la fotosíntesis ellas producen glucosa (C<sub>6</sub>H<sub>12</sub>O<sub>6</sub>) y oxígeno(O<sub>2</sub>). Las plantas además de producir el alimento de los animales herbívoros, ellas constituyen el primer eslabón de las cadenas tróficas por las que fluyen la biomasa y la energía en los ecosistemas de la Tierra. (Tíscar Oliver, 2021).

#### **A. Nutrición de las plantas**

Tiene cuatro fases: absorción, circulación, fotosíntesis y alimentación.

La etapa de absorción es donde la planta asimila el agua y minerales del entorno mediante las raíces, después los nutrientes absorbidos circulan a través del vegetal hasta las hojas en donde se suele realizar la fotosíntesis. (Bona, 2023)

El objetivo del proyecto se basa solamente en las 2 primeras fases que son absorción y circulación.

La absorción en la raíz, los pelos absorbentes están rodeados por el agua del suelo y las sales disueltas en ella; esta solución es menos concentrada que el jugo celular de los pelos absorbentes, por lo cual merceda la osmosis, pasa a través de la membrana de las células siguientes. El protoplasma de estas células es menos concentrado por consiguiente se repite el fenómeno, por la misma causa, en todas las células siguientes

hasta que el agua con las sales disueltas llegue a los vasos leñosos de la raíz, donde comienza a circular hacia el tallo, dándose el nombre de sabia bruta. (Cornejo García, Verónica Rosales, Gauna Flores, Esteban Rubio, & Campo Gutiérrez, 2006). La cual se extiende por toda la planta a través del tallo y ramificaciones hasta llegar a las hojas.

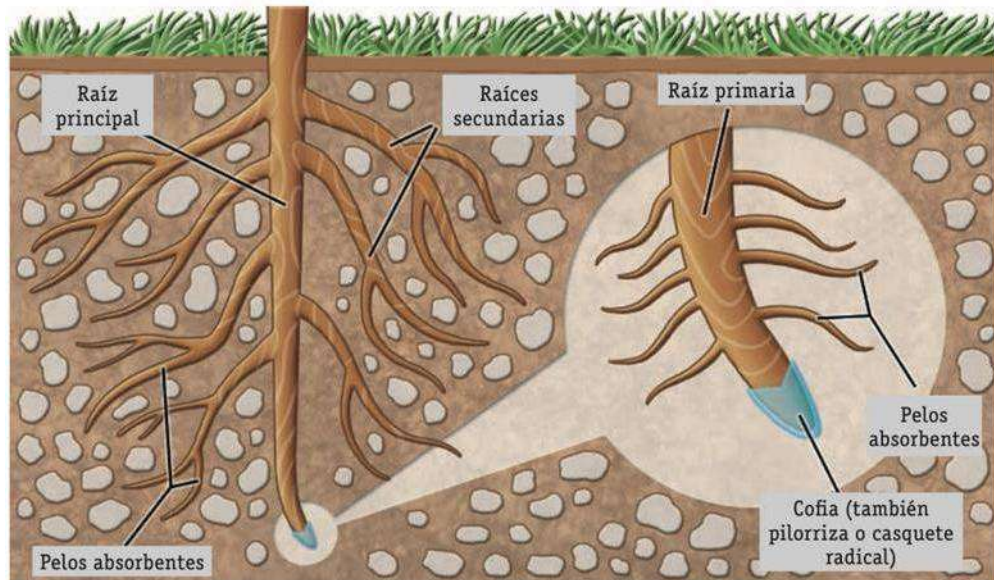


Ilustración 1: Estructura de la raíz de una planta

Fuente: <https://www.ikostech.es/cue/blossom-end-rot-causas-y-como-remediarlo/>

## B. Tipo de suelo

Los suelos naturales dependen del nivel de pH presentes en ellos, el pH suele oscilar entre 4,5 y 8 (FAO, 2023)

- Suelo neutro: son suelos con pH de 6,5 a 7,5. Ricos en materia orgánica y residuos vegetales.
- Suelo ácido: son suelos con pH inferior a 5,5. Asociado con un número de toxicidades (aluminio) y deficiencias (molibdeno) y otras condiciones restringentes para las plantas
- Suelos alcalinos: son aquellos suelos con pH superior a 7,5. Son aquellos afectados por el carbonato de calcio y yeso afectando el crecimiento de las plantas.

Ahora en base a la CEe y a la respuesta de cultivo, los suelos se clasifican en (Interreg, 2023):

- No salino (CEe)
- Ligeramente salinos (entre 2 y 4 dS/m)
- Moderadamente salinos (entre 4 y 8 dS/m)
- Muy salinos (CEe>8 dS/m)

### **2.2.2. Conductividad Eléctrica del suelo**

Los estudios para aumentar la productividad de los cultivos se han basado hasta ahora en el control de temperatura, humedad, fertilizantes, oxigenación e intensidad de luz.

La conductividad eléctrica del suelo es una medida indirecta de la concentración de sales. El suelo natural tiene disueltas sales, por lo que la CE puede ser baja pero nunca nula, las sales son buenas para los organismos que las consumen, sin embargo, el exceso puede afectar tanto al crecimiento de las plantas como a la actividad de los microorganismos del suelo. (Cremona & Enriquez, 2020)

La conductividad eléctrica varía en un amplio rango dependiendo de los factores como son el material de origen o los factores deformadores de suelo. La topografía también condiciona este parámetro, encontrando acumulación de sales en zonas bajas del paisaje. Existen numerosas escalas que clasifican a los suelos según su salinidad, la escala presentada a continuación es una orientación para evaluar la salinidad en función de la CE de una suspensión de suelo:agua en una relación 1:2,5. (Cremona & Enriquez, 2020)

Tabla 1: Clasificación del suelo según su salinidad

Suspensión – relación suelo/agua 1:2,5	Conductividad eléctrica (dS/m)
No salina	0-1
Levemente salina	1-2
Moderadamente salina	2-4
Muy salina	4-8

### A. Salinidad en la agricultura

En la agricultura las sales disueltas tienen cargas (generando iones positivos y negativos) que retienen a las moléculas de agua con una fuerza que compite con la que tiene que hacer las plantas para absorberlas desde el suelo. Como consecuencia, la salinidad reduce la disponibilidad de agua para las plantas, mediante el efecto desecante que provoca en el suelo. Así una elevada CE será entonces perjudicial para el crecimiento de las plantas, lo que se verá afectado en función de la tolerancia de cada especie. (Cremona & Enriquez, 2020)

### B. Estándares de calidad de salinidad del suelo

Tabla 2: Clasificación del contenido de sal en el suelo por la FAO

		EC <sub>SE</sub> =dS/m(25 °C)
N	(cerca) No salino	< 0,75
SL	Ligeramente salino	0,75 – 2
MO	Moderadamente salino	2 – 4
ST	Fuertemente salino	4 – 8
VST	Muy fuertemente salino	8 – 15
EX	Extremadamente salino	> 15

Fuente: (Organización de las Naciones Unidas para la Agricultura y la Alimentación, 2009)

### C. Tipos de sensores de conductividad eléctrica aparente del suelo

Estos se pueden clasificar según su principio de funcionamiento (Interreg, 2023):

- Resistividad eléctrica (Re): Exigen un buen contacto entre el sensor y el suelo (económicos).
- Capacitancia (FDR y TDR): algunos sensores dentro de este grupo no requieren de un contacto tan perfecto como los de Re. (amplio rango de prestaciones y precios).
- Inducción electromagnética (EMI): no necesitan contacto entre el instrumento y el suelo. Son rápidos, pero de precio elevado.
- De inyección de corriente: también llamados de electrodos de contacto. Se basa en la creación de un campo eléctrico directamente en el suelo mediante dos electrodos de contacto que inyectan una corriente entre otros electrodos y miden las diferencias de voltaje que circula.

#### **D. Funcionamiento del sensor de conductividad eléctrica del suelo.**

Los sensores para cultivo se utilizan para brindar a los agricultores lecturas rápidas, objetivas, precisas y cuantitativas que serían difíciles o demandante de tiempo para conseguir mediante otras formas.

Los sensores de suelo están conformados por sensor de humedad, temperatura y conductividad eléctrica, parámetros más importantes y demandante del suelo por los profesionales de la agricultura.

El sensor de humedad, temperatura y conductividad eléctrica del suelo (RK520-02) cuenta con un grado de protección IP68, una sonda de material 316SS y una señal de salida RS485 a 0-2V.

En la siguiente tabla se muestra las especificaciones técnicas del sensor de ejemplo, el RK520-02:

Tabla 3: Especificaciones técnicas del sensor de ejemplo RK520-02

Parámetro	Especificación técnica		
	Humedad	Temperatura	Conductividad eléctrica
Rango	0-100% (m <sup>3</sup> /m <sup>3</sup> )	-30°C~ +70°C	0~10mS/cm (0 a 2V) 0~20mS/cm (RS485)
Precisión	±3%	±0,3	±2% FS
Señal de salida	RS485, 0 a 2V		
Tiempo de respuesta	< 1s		
Alimentación	5VDC, 12~24VDC		
Área de medición efectiva	70mm de diámetro por 70mm de altura representación cilíndrica de medida.		
Material de carcasa	ABS		
Dimensiones	66*45*16mm <sup>3</sup> (puntas de prueba: 3Φ*70mm <sup>3</sup> )		
Temperatura de operación	-40°C ~80°C		
Protección IP	IP68		
Condiciones de almacenamiento	10°C ~ 60°C a 20% ~ 90%RH		
Material de sonda	Acero inoxidable 316L		

Fuente: Kusitest (2023) <https://www.kusitest.pe/sensor-de-humedad-temperatura-y-conductividad-electrica-en-suelo-rk520-02>

### 2.2.3. Tecnologías de control: Clasificación:

- **Sistema centralizado:** se pueden catalogar dentro de este grupo aquellos sistemas en los que haya un solo microcontrolador, encargado de la gestión de la instalación, que será el encargado de ejecutar el programa que rige el funcionamiento de esta. (Guzmán Navarro & Merino Córdoba, 2015).

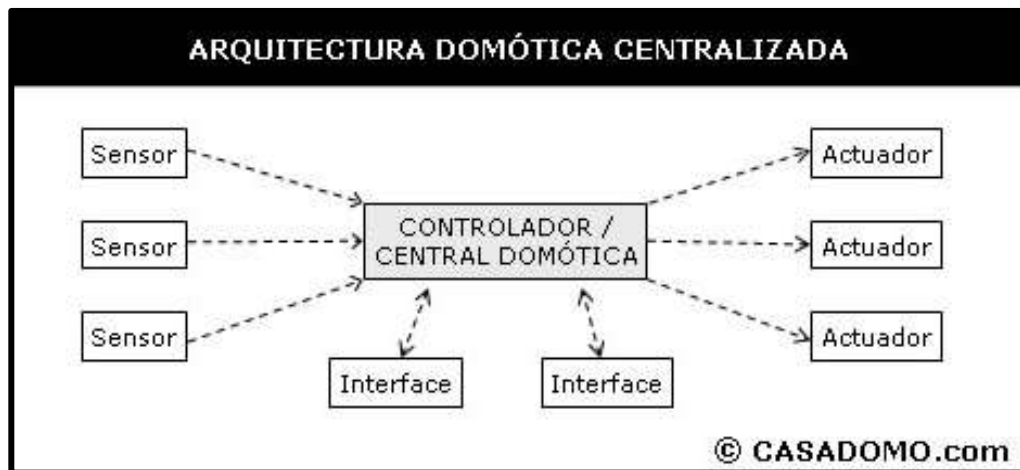


Ilustración 2: Arquitectura centralizada

Fuente: <https://www.casadomo.com>.

Que un sistema se pueda monitorizar desde un ordenador no quiere decir que sea un control centralizado, solo entran en esta categoría aquellos en los que se puede variar el estado de las entradas o salidas desde un control central. Si esta llegara a fallar y quede fuera de servicio, el sistema global queda inutilizable por completo.

- **Sistemas descentralizados:** Se encuentran en esta categoría los sistemas en los que el control de la instalación se realiza de forma parcial e independiente mediante varios microcontroladores repartidos por ella. Su principal ventaja es que en caso de haber un fallo de algún controlador solo se ve afectado la parte de la instalación que esta gobierna. (Guzmán Navarro & Merino Córdoba, 2015)

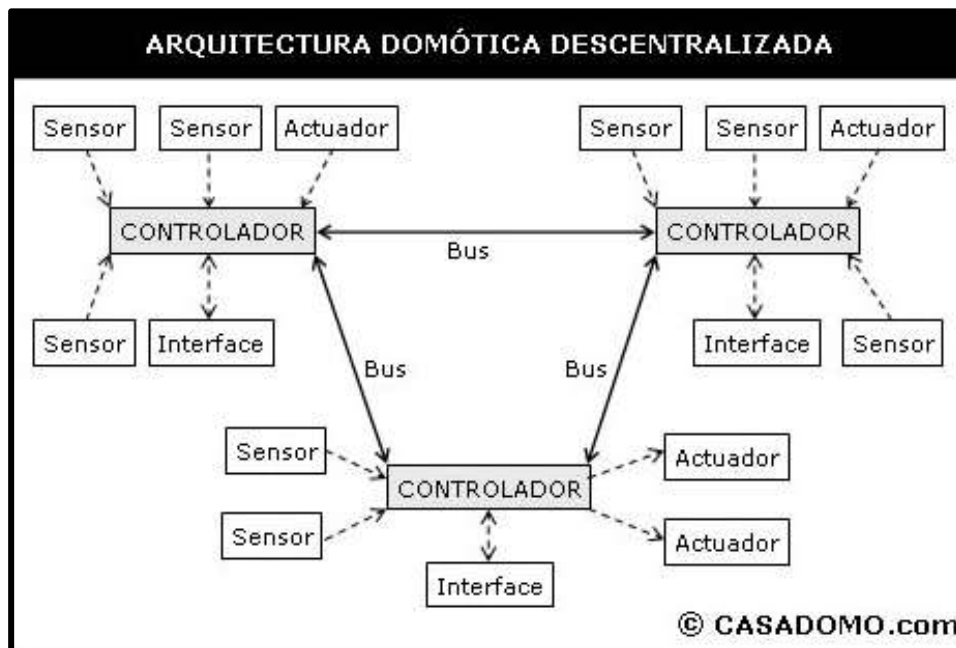


Ilustración 3: Arquitectura descentralizada

Fuente: <https://www.casadomo.com>.

- **Sistema distribuido:** Sistema donde toda la inteligencia del sistema está distribuida por todos los módulos sean sensores o actuadores. Suele ser típico de los sistemas de cableado en bus, o redes inalámbricas. (Elías Castells & Bordas Alsina, 2012)

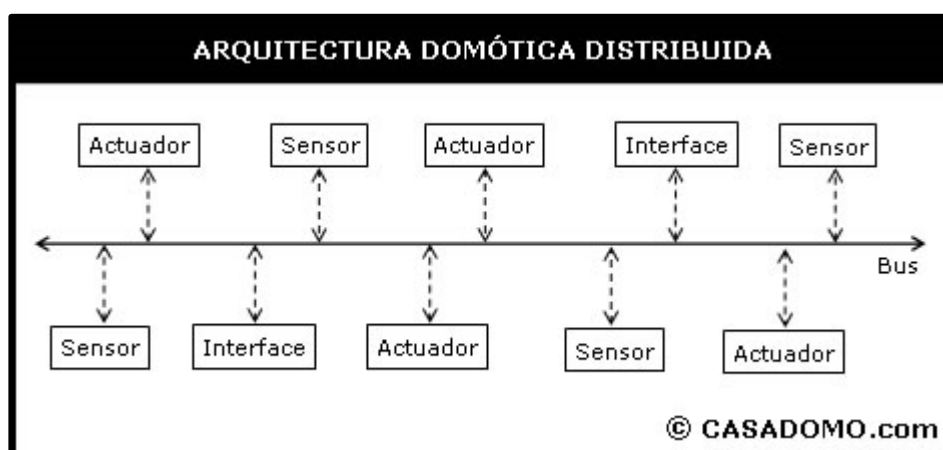


Ilustración 4: Arquitectura distribuida

Fuente: <https://www.casadomo.com>.

- **Sistema híbrido/mixta:** Sistema con arquitectura descentralizada en cuanto a que disponen de varios pequeños dispositivos capaces de adquirir y procesar la información de múltiples sensores y transmitirlos al resto de dispositivos distribuidos por la vivienda, por ejemplo, sistemas basados en ZigBee y totalmente inalámbricos. (Elías Castells & Bordas Alsina, 2012)

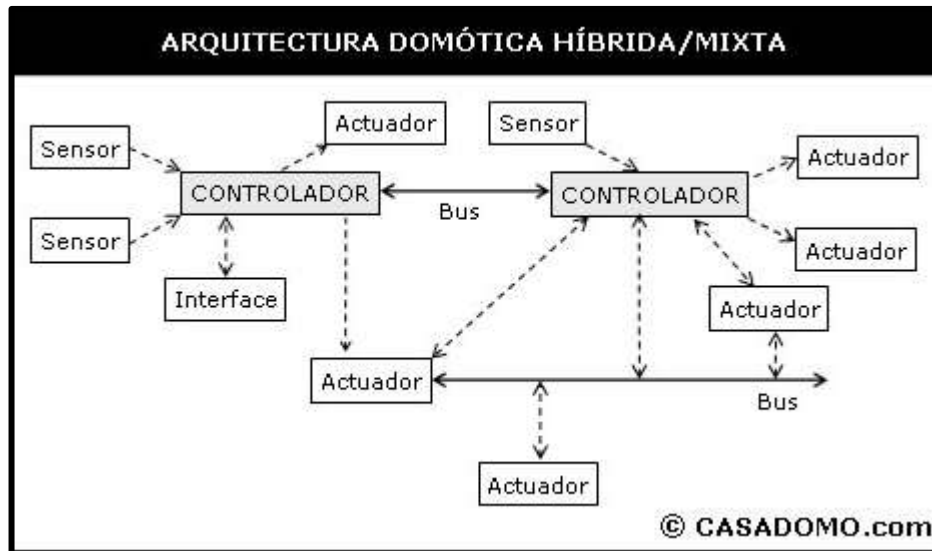


Ilustración 5:Arquitectura Híbrida/Mixta

Fuente: <https://www.casadomo.com>.

#### A. Medios de interconexión

Los medios de interconexión de un sistema domótico son múltiples:

- Alámbricos: Xdsl, fibra óptica.
- Cable: coaxial y UTP.
- Inalámbrico: WIFI, GPRS, Bluetooth, radiofrecuencia, infrarrojo, ZigBee, etc.

#### B. Protocolo de comunicación

- KNX/EIB: Es el bus de instalación europeo con más de veinte años y más de 100 fabricantes de productos compatibles entre sí.
- X10: Protocolo de comunicaciones para el control remoto de dispositivos eléctricos, hace uso de los enchufes eléctricos, sin

necesidades de nuevo cableado. Tiene poca fiabilidad frente a ruidos eléctricos, si los hay.

- Zigbee: protocolo estándar, recogido en el IEEE 802.15.4, de comunicaciones inalámbricas. Tecnología de transmisión inalámbrica por radiofrecuencia de rango de alcance entre 10 y 75 metros.
- WIFI: protocolo de comunicación inalámbrica digital binaria que transmite los datos a través del aire con la velocidad suficiente para aplicaciones domóticas. Cada aplicación tiene necesidad de diferentes anchos de banda.
- MQTT: protocolo de comunicación M2M de tipo cola de mensajes o message queue.
- Entre otras tenemos a la DALI, Z-Wave, LonWorks, Modbus, BACnet, INSTEON.

#### **2.2.4. Internet of things (IoT)**

El Internet de las Cosas, es un componente tecnológico fundamental sobre el que sienta sus bases el paradigma de la industria 4.0.

La IoT es una red abierta y completa de objetos inteligentes que tienen la capacidad de autoorganizarse, compartir información, datos y recursos, reaccionar y actuar ante situaciones y cambios en el entorno. (Villa Crespo & Morales Alonso, 2022)

##### **A. Arquitectura del 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.

Se compone de 4 fases (UNIR: La Universidad en Internet, 2023):

- Fase 1: Sensores y actuadores
- Fase 2: Sistemas de adquisición de datos
- Fase 3: Análisis en la frontera
- Fase 4: Análisis en el centro de datos o en la nube

## B. Agricultura 4.0

La aplicación del IoT a la industria ha permitido la recolección y análisis de grandes cantidades de datos, que pueden usarse para mejorar el rendimiento de los sistemas industriales y reducir los costes de operación.

Requiere mayor nivel de seguridad física, lógica y comunicaciones confiables. Siendo el concepto común como el uso de tecnologías y aplicaciones avanzadas optimizadas especialmente para soportar casos de uso de procesos industriales. Sus beneficios con el optimizar procesos y el mantenimiento predictivo. (Villa Crespo & Morales Alonso, 2022)

### 2.2.5. Redes inalámbricas

Las redes inalámbricas son redes que utilizan ondas de radio para conectar los dispositivos, sin la necesidad de utilizar cables de ningún tipo. Los dispositivos que usan estas redes son los laptops, pc de escritorio, netbooks, asistentes digitales personales (PDA), Celulares, tabletas y dispositivos de geolocalización. (Salazar, 2017)

Existen diferentes tecnologías que difieren en la frecuencia de transmisión utilizada, en la velocidad y en el alcance de sus transmisiones.

Según Salazar (2017), las redes inalámbricas se pueden clasificar en cuatro grupos específicos según el área de aplicación y el alcance de la señal. Los grupos se visualizan en la imagen siguiente:

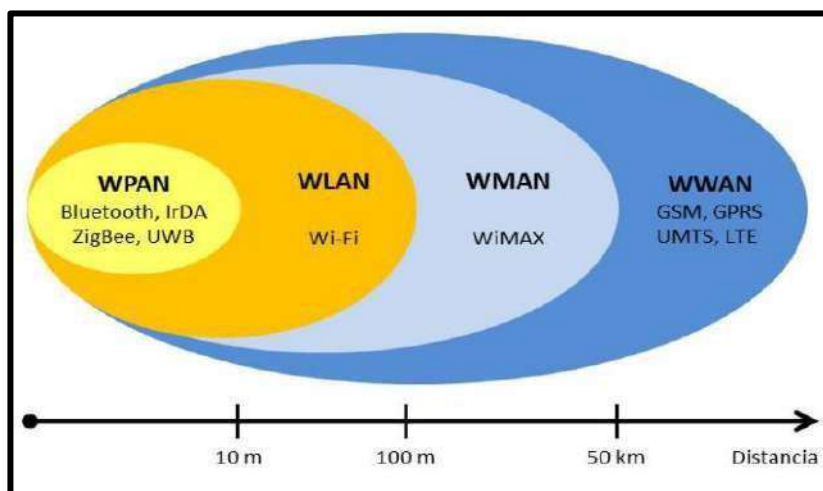


Ilustración 6: Clasificación de las redes inalámbricas

Fuente: Redes inalámbricas, 2017

Por otro lado, las redes inalámbricas se pueden dividir en dos segmentos: de corto y largo alcance.

Las de corto alcance hace referencia a las redes confinadas en un área limitada, redes de área local (LAN), redes de área personal (PAN) y la red que suele operar sobre un espectro sin licencia y reservado para uso industrial, científica y médica (banda ISM). Su banda de frecuencia más comunes son la de 2,4 GHz y la de 5GHz. (Salazar, 2017)

La de largo alcance su conectividad es típicamente proporcionada por las empresas que comercializan la conectividad inalámbrica como un servicio. Estas redes cubren grandes áreas, como un área metropolitana (WMAN), un estado, provincia o hasta un país entero. El objetivo de estas redes es proporcionar cobertura inalámbrica a nivel mundial, la más común es la red amplia (WWAN), para una cobertura global también están disponible las redes por satélite. (Salazar, 2017).

## A. Comparación de tecnologías inalámbricas de corto alcance más usadas

Tabla 4: Comparación de Tecnologías Inalámbricas

	Bluetooth	ZigBee	WI-FI
Estándar IEEE	802.15.1	802.15.4	802.11a/b/g
Topología de red	Ad-hoc, red pequeña	Ad-hoc, punto a punto, estrella, malla	Point to hub
Celda básica	Piconet	Star	BSS
Aplicación	Reemplazo de cableado	Automatización, control	Web, e-mail, video
Banda de Frecuencias	2,4 GHz	868/915 MHz 2,4 GHz	2,4 GHz 5 GHz 6 GHz
Tasa de transferencia	1 Mb/s	250 kb/s	54 Mb/s
Canal de banda ancha	1 MHz	0.3/0.6 MHz; 2MHz	22 MHz
Potencia de transmisión nominal	0-10 dBm	(-25) – 0 dBm	15 – 20 dBm
Número de Canales RF	79	1/10; 16	14 (2,4GHz)
Tipo de modulación	GFSK	BPSK(+ASK), O-QPSK	BPSK, QPSK COFDM, M-QAM
Tipo de datos	Digital, audio	Digital (texto)	Digital
Tamaño de red	7	65535	32
Alimentación	1-7	>1000	1-5
Rango nominal	10 metros	10 - 100 metros	100 metros
Ventajas	Costo Comodidad	Confiabilidad Rendimiento Costo	Velocidad de datos Flexibilidad
Consumo de Potencia	Medio	Muy bajo	Alto
Precio	Bajo	Bajo	Alto
Complejidad	Alta	Baja	Alta

Fuente: <https://www.researchgate.net/publication/316103356>

## **2.2.6. Componentes y plataformas**

### **A. ESP32**

El ESP32 es una serie de microcontroladores con sistema en un chip (SoC) de bajo costo y bajo consumo energético, el cual integra WI-FI y Bluetooth de doble modo con interfaces SPI / SDIO o I2C / UART. Es desarrollado por la empresa Espressif Systems. En su núcleo emplea un microprocesador Tensilica Xtensa LX6 de simple o doble núcleo con una velocidad de reloj de 160 a 240 MHz. Además, integra interruptores de antena, balun de radio frecuencia, amplificadores de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía. Diseñado para dispositivos móviles, electrónica portátil y aplicaciones de IoT, logrando un consumo de energía ultra bajo con características de última generación, como el control de reloj precisa, varios modos de energía y escalado de energía dinámica. (Espressif Systems, 2023)

### **B. Raspberry Pi**

La Raspberry Pi es una computadora de bajo costo y compacta, puede conectarse a un monitor o una tv y usarse con mouse y teclado. Usa el sistema operativo Linux, es capaz de hacer de realizar las tareas típicas de un computador de escritorio como navegar en internet, reproducir videos de alta resolución, manipular documentos de ofimática y hasta reproducir juegos. (Raspberry Pi, s.f.)

### **C. Ubidots**

Ubidots es una plataforma de IoT que ayuda a miles de emprendedores, integradores de sistemas y empresas de IoT a lanzar y escalar proyectos de IoT a producción.

La plataforma de Ubidots envía datos a la nube desde cualquier dispositivo conectado a internet, permitiendo visualizar y procesar datos en tiempo real. Para luego configurar acciones y alertas basadas en el análisis de los datos.

Ubidots nos ofrece un API REST el cual nos permite leer y escribir datos en los recursos disponibles: fuentes de datos, variables, valores, eventos y

perspectivas. La API admite HTTP y HTTPS y una clave de API es obligatoria.

Ubidots nos ofrece además protección de datos con dos replicaciones, almacenamiento cifrado y soporte de datos TLS/SSL opcional. Además, puede personalizar los grupos de permisos para cada módulo de la plataforma, así asegura de que se muestre la información correcta al usuario correcto. (Ubidots, s.f.)

#### D. Protocolo MQTT

MQTT es un protocolo de mensajería estándar de OASIS para el IoT, diseñada como un transporte de mensajería de publicador/suscriptor extremadamente ligero que es ideal para conectar dispositivos remotos con una huella de código pequeña y un ancho de banda de red mínimo.

MQTT permite enviar mensajes entre el dispositivo y la nube y viceversa. La confiabilidad de la entrega de mensajes es importante para muchos casos de uso de IoT. Por esta razón MQTT ofrece 3 niveles de calidad de servicios definidos: 0 – como máximo una vez, 1 – al menos una vez, 2 – exactamente una vez.

MQTT facilita el cifrado de mensajes mediante TLS y la autenticación moderna como OAuth.

Hoy en día, MQTT es suada en una amplia variedad de industrias, como la automotriz, la manufacturera, las telecomunicaciones, el petróleo y el gas, etc. (MQTT, 2022)

#### Arquitectura de publicación/suscripción de MQTT

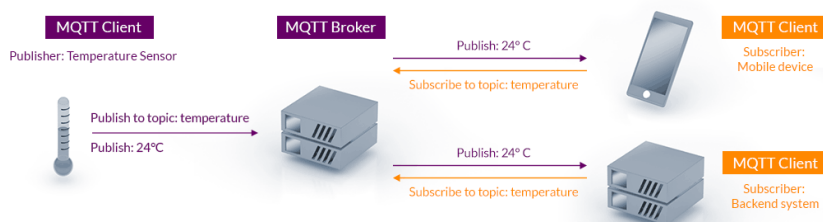


Ilustración 7: Arquitectura de publicación/suscripción de MQTT

Fuente: (MQTT, 2022)

- **MQTT Client:** se refiere a publicador y suscriptor. Un publicador es un cliente que envía el mensaje, mientras que un suscriptor es un cliente que recibe el mensaje. Sin embargo, un cliente MQTT también puede ser un publicador y suscriptor. Entonces un cliente MQTT es cualquier dispositivo, desde un pequeño microcontrolador hasta un servidor gigantesco, que ejecuta una biblioteca MQTT y se conecta a un bróker MQTT a través de la red.
- **MQTT Broker:** es un eje central en el sistema de mensajería de publicador/suscriptor el cual recibe mensajes de los publicadores y los distribuye a los suscriptores. Desempeña un papel fundamental en la gestión del flujo de comunicación entre los clientes MQT y en la garantía de una entrega fiable de mensajes. Dentro de sus funcionalidades incluye:
  - El manejo de un gran número de conexiones simultaneas.
  - El filtrado y direccionamiento de mensajes.
  - La gestión de sesiones.
  - La autenticación y autorización.
  - La escalabilidad, integración y monitoreo.
  - Y algunos admiten la agrupación en clústeres, lo que permite que varias instancias de bróker trabajen en simultaneo para gestionar un gran número de clientes y mensajes.
- **MQTT Connection:** la conexión MQTT siempre son entre cliente y bróker, y los clientes nunca se conectan directamente con otros clientes. Para iniciar conexión, el cliente envía un mensaje "CONNECT" al intermediario, el cual responde con un mensaje "CONNACK" y un código de estado. Una vez realizada la conexión, el bróker la mantiene abierta hasta que el cliente envía un comando de desconexión o la conexión se interrumpa. El cliente y bróker deben tener una pila TCP/IP para poder comunicarse.

### 2.3. Marco conceptual

- Arduino IDE: o entorno de Desarrollo Integrado, es un conjunto de herramientas de software que nos permiten el desarrollo y grabado del código necesario para hacer que nuestro esp32 funcione como queramos.
- Python: Lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learnig. (AMAZON, 2023)
- Nube: es una metáfora para referirse a una red mundial de servidores remotos que funcionan como un único ecosistema, normalmente asociada a internet. La nube nos permite almacenar datos y archivos en una ubicación externa a la que se accede a través de internet pública o privada, un portal web, un sitio web o una aplicación móvil. La nube es una alternativa rentable y escalable, además, cuenta con servidores de respaldo o servidores de redundancia. Se dividen en nubes públicas, privadas e híbridas. Y según el tipo de almacenamiento se dividen en: archivo, bloque y objeto. (IBM, 2023)
- Dashboard: es una interfaz de usuario donde se organiza y presenta la información más importante en un solo lugar, la cual es necesaria para lograr uno o más objetivos, consolidada y dispuesta en una sola pantalla para que los usuarios puedan monitorear y analizar el desempeño de una empresa, departamento o proyecto en tiempo real, lo que permitirá tomar decisiones más informadas y eficaces. Los dashboards contienen gráficos, tablas, indicadores y demás representaciones visuales de datos que se actualizan constantemente y que proporcionan una vista general de la información más relevante para una organización. Dentro de los contextos a usar se incluyen: el monitoreo del desempeño de la empresa, el seguimiento de un proyecto, análisis de datos y el seguimiento de KPI (Indicadores de clave de rendimiento). (L, 2023)
- IoT: El termino IoT, o también llamado el Internet de las Cosas, hace referencia a una red colectiva de dispositivos conectados y a la tecnología que facilita la comunicación entre los dispositivos y la nube, así como entre

los propios dispositivos. Funciona mediante la recopilación y el intercambio de datos en tiempo real, tiene tres componentes: dispositivos inteligentes, aplicación IoT e interfaz de usuario gráfico. (Amazon, 2023)

#### 2.4. Sistema de hipótesis

Mediante el diseño de un sistema IoT electrónico de monitoreo basado en el sensor por medición de conductividad eléctrica del suelo (CE) nos permite medir los parámetros de concentración de sales en el suelo, obteniendo un registro histórico de datos en la nube.

#### 2.5. Variables e indicadores

##### A. Variables Dependientes:

Salinidad del suelo en los sembradíos del pueblo de San José

##### B. Variables Independientes:

Sistema electrónico de monitoreo basado en la CE.

##### C. Operacionalización de las variables

Tabla 5: Variable dependiente

Variable Dependiente	Dimensión	Indicadores	Unidad de medida	Instrumento
Salinidad del suelo	CE	Estado de la vegetación / prueba de laboratorio de la salinidad del suelo	dS/m	Ficha técnica

Tabla 6: Variable independiente

Variable Independiente	Dimensión	Indicadores	Unidad de medida	Instrumento
Sistema electrónico de monitoreo basado en la CE, la raspberry, ESP32 y la conexión a la nube	Conductividad eléctrica	Variación de la CE del suelo	dS/m	Sensor de Conductividad eléctrica del suelo

### **III. Metodología Empleada**

#### **3.1. Tipo y nivel de investigación**

##### **3.1.1. De acuerdo con la orientación o finalidad**

Aplicada

##### **3.1.2. De acuerdo con su alcance temporal**

Transversal

##### **3.1.3. De acuerdo con su carácter**

Cualitativa y cuantitativa

##### **3.1.4. De acuerdo con la técnica de contrastación**

Descriptiva

#### **3.2. Población y muestra de estudio**

##### **3.2.1. Población**

Bernal (2006), nos menciona que de acuerdo con Fracica, la población es “El conjunto de todos los elementos a los cuales se refiere la investigación. Se puede definir también como el conjunto de todas las unidades de muestreo” (Pág.164).

Se pueden definir a la población, a las hectáreas de terreno agrícola en San José – Cartavio.

##### **3.2.2. Muestra**

Malhotra (2004), nos menciona que: “El tamaño de la muestra se refiere al número de elementos que se incluirán en el estudio. Determinar el tamaño de la muestra es complejo e incluye diversas consideraciones cualitativas y cuantitativas.” (Pág. 318).

Una parcela de sembradío del Pueblo de San José – Cartavio.

### **3.3. Diseño de investigación**

Diseño temporal: longitudinal, realizando una medición de varias horas del terreno de cultivo.

### **3.4. Técnicas e instrumentos de investigación**

- Técnicas de muestreo: Simple.
- Técnicas de recolección de datos: de gabinete. El estudio se realizará usando la información de internet con respecto a IoT y la agricultura y mediante encuestas a los agricultores.
- Instrumentos de recolección de datos: Revisión de información estadística de portales gubernamentales: FAO, INEI, MINEDU, entre otros, de interés IoT y agricultura.
- Confiabilidad y validez de los instrumentos: la información dada por entes gubernamentales y es válida y confiable, por lo tanto, no se requiere validación de algún especialista externo.

### **3.5. Procesamiento y análisis de datos**

#### **3.5.1. Identificar los sistemas de medición de conductividad eléctrica del suelo existentes en Trujillo.**

En esta sección se menciona los dispositivos utilizados en los estudios para la medición de conductividad eléctrica del suelo por investigadores trujillanos y en algunas universidades.

- Luis Campoverde (2012) en su investigación titulada “Evaluación De Áreas Agrícolas Con Problemas De Salinidad Para Uso Potencial En Acuicultura En El Valle Bajo Del Rio Santa, Ancash-Perú, 2008”, trabajo presentado en Trujillo el 2012, se realiza la medición de la conductividad eléctrica del suelo (CE) indicadora de la concentración salina de un suelo, haciendo uso de un conductímetro HACH Sension 5.

Tabla 7: Especificaciones técnicas del sensor HACH Sension 5

Parámetro	Especificación técnica			
	Salinidad	TDS	Temperatura	Conductividad eléctrica
Rango	0 a 42 g/l (ppt)	0 a 50.000 mg/l con NaCl	-10.0 a 110.0°C	0 a 199.9µS/cm 200 a 1999µS/cm 2 a 19.99mS/cm 20 a 199.9mS/cm
Precisión	± 0.1 ppt (-2 - 35°C)	± 0.5% de la escala plena	± 0.3°C a 0 – 70°C; ± 1.0°C a 70– 110°C	± 0.5% en primeros tres rangos; 1% en rangos más altos
Señal de salida	Unidireccional RS-232 vía estación de comunicación			
Memoria de datos	Memoria de 99 puntos			
Alimentación	4 baterías AA o 110 a 220 VAC.			
Entradas	Un conector de 5 Pin con temperatura			
Pantalla	LCD			
Dimensiones	21.2 x 8.7 x 4.2 cm			
Protección IP	IP67			
Duración de batería	6 meses con uso normal (500h aprox.)			
Material de sonda	Acero inoxidable 316L			
Fuente: <a href="https://www.yumpu.com/es/document/view/11906037/sension-5-medicion-de-conductividad-tds-y-salinidad-en-campo">https://www.yumpu.com/es/document/view/11906037/sension-5-medicion-de-conductividad-tds-y-salinidad-en-campo</a>				

- Denis Castañeda et al., (2022) en su investigación titulada “Recuperación De Suelos Salinos, Identificando El Método De Desalinización Más Eficiente En El Distrito De Salaverry-Trujillo-Perú.” Realizaron la medición del pH, conductividad, temperatura, nivel de salinidad y porcentaje de materia orgánica por un periodo de 5 días, haciendo uso del dispositivo HANNA, modelo HI8424-Chile.

Tabla 8: Especificaciones técnicas del sensor HANNA, modelo HI8424

Especificaciones	
Rango pH	-2.00 a 16.00 pH
Rango mV	± 699.9 mV; ± 1999 mV
Rango de Temperatura	-20.0 a 120.0°C / -4.0 a 248.0°F
Resolución pH	0.01
Resolución mV	0.1 mV; 1 mV
Resolución Temperatura	0.1°C/0.1°F
Precisión (20°C) pH	± 0.01
Precisión (20°C) mV	± 0.2 mV; ± 1 mV
Precisión (20°C) Temperatura	± 0.4°C / ± 0.8°F
Calibración pH	Automática, uno o dos puntos con tres valores buffer memorizables (Ph 4.01, 7.01, 10.01)
Calibración de ajuste	± 1pH
Calibración pendiente	75 a 110%
Impedancia de entrada	10 <sup>12</sup> ohm
Compensación de temperatura	Automática, -20 a 120°C o manual sin sonda de temperatura

Electrodo de pH	Electrodo de cuerpo PEI HI 1230B con conector
Sonda de temperatura	Sonda de temperatura HI
Tipo de batería / Vida	9V / aprox. 150 horas de uso continuo; auto - apagado después de 20 min de no uso o deshabilitado (seleccionable por el usuario).
Ambiente	0 a 50°C (32 a 122°F); HR máx. 100%
Dimensiones	164 x 76 x 45 mm (6.5 x 3.0 x 1.8")
Peso	180g (6.3 oz)
Fuente: <a href="https://www.hannacolombia.com/products/product/344/pdf">https://www.hannacolombia.com/products/product/344/pdf</a>	

- La Universidad Privada Antenor Orrego en su clase denominada Química Analítica, realiza prácticas de medición de la conductividad eléctrica de diferentes muestras de suelo en ( $\mu\text{S}/\text{cm}$ ), determinan el contenido de sales disueltas totales en partes por millón (ppm) y diferencian los tipos de salinidad en suelos agrícolas por concentración de sales. En la práctica de aula utilizan el conductímetro de bolsillo Cond 3110 Set 1.

Tabla 9: Especificaciones técnicas del sensor Cond 3110 Set 1

Especificaciones: Rangos de medición / resolución / precisión / tiempo de respuesta	
Conductividad	0.0 ... 1000 mS/cm $\pm$ 0.5% del valor medido, >10seg
Temperatura	-5.0°C ... + 105.0°C $\pm$ 0.1°C, >15seg
Salinity	0.0 ... 70.0 (según IOT)
TDS	-
Resistencia específica	Seleccionable 20°C o 25°C
Temperatura de referencia	Seleccionable 20°C o 25°C
El constante celular puede ser calibrada permanentemente	0.475 cm <sup>-1</sup> 0.450 ... 0.500cm <sup>-1</sup> , 0.800...0.880 cm <sup>-1</sup>
Compensación de temperatura	Automáticamente
Coefficiente de temperatura	Función no lineal del agua natural (nLF) según EN 27 888
Memoria de datos/registrator	-
Pantalla	LCD de 7 segmentos personalizado
Funcionamiento continuo	Hasta las 1000 horas
Protección IP	IP 67
Fuente: <a href="https://www.geotechenv.com/Manuals/WTW_Manuals/WTW_Cond_3110.pdf">https://www.geotechenv.com/Manuals/WTW_Manuals/WTW_Cond_3110.pdf</a>	

Se identificaron tres sistemas de medición en la ciudad de Trujillo: el conductímetro HACH Sension 5, el conductímetro HANNA modelo HI8424, y el conductímetro Cond 3110 set 1. Cada uno ofrece características únicas en rangos de medición precisión y funcionalidad.

Aunque todos estos dispositivos cumplen con su propósito, presentan limitaciones en integración IoT, almacenamiento de datos en la nube y accesibilidad de datos en tiempo real, lo cual evidencia una brecha tecnológica para aplicaciones avanzadas de monitoreo en tiempo real.

### 3.5.2. Diseñar un prototipo electrónico que lea la información de la conductividad eléctrica del suelo usando un sensor de CE que envíe la información a una base de datos en la nube.

#### D. Descripción general del sistema de monitoreo de conductividad eléctrica.



Ilustración 8: Descripción general del sistema de monitoreo

En la ilustración 8 se presenta el diagrama general del diseño de nuestro sistema para el monitoreo de la salinidad por medio de la conductividad eléctrica para los terrenos agrícolas, creado con una arquitectura mixta. En este sistema integramos 3 componentes principales y un servicio en la nube/dashboard: el sensor de suelo CWT-Soil-THC-S, una ESP32, una Raspberry Pi, y la plataforma Ubidots.

El sensor de suelo CWT-Soil-THC-S, diseñado para monitorear las condiciones del suelo agrícola. Capaz de medir la humedad, temperatura, conductividad, además, nos ofrece los datos de salinidad y TDS (Total Dissolved Solids). Nos ofrece datos medidos con alta precisión y confiabilidad usando el protocolo de comunicación modbus RTU (RS485), en otras palabras, se establece una conexión serial eficiente entre la ESP32 y el sensor.

El Esp32, encargado de capturar los datos provenientes del sensor de suelo CWT-Soil-THC-S. Los datos capturados son transmitidos de manera inalámbrica hacia la raspberry Pi usando el protocolo MQTT, quien actúa como un intermediario o interconector de dispositivos, gestionando la conexión con el sensor y asegurando la transmisión eficiente, cada 30min, hacia la plataforma de servicios Ubidots.

Finalmente, la plataforma de servicios Ubidots es utilizada como base de datos en la nube y centro de visualización (Dashboards), permitiendo almacenar y monitorear los datos recolectados.

A continuación, se muestra el diagrama de flujo del sistema de monitoreo:

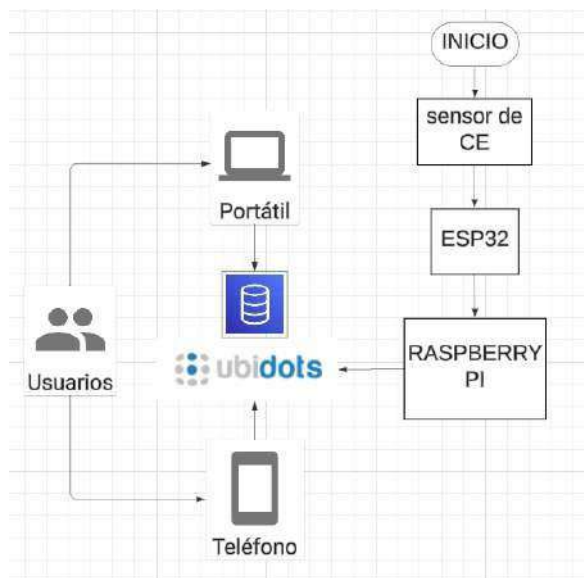


Ilustración 9: Diagrama de flujo del sistema IoT

## E. Hardware y software.

- **Modulo ESP32:** Es un microcontrolador que ha sido desarrollado por Espressif Systems. Se caracteriza por su arquitectura de doble núcleo. (dual-core), tiene conectividad wi-fi y bluetooth integrada y una amplia variedad de periféricos de entrada y salida.

El ESP32 es programado usando el Arduino IDE configurando el gestor de tarjetas con la URL de Espressif eligiendo la placa a utilizar y cargando el código a través de un cable USB.

En el siguiente cuadro se detallan, en comparativa, diferentes modelos ESP32 desarrollados por Espressif Systems:

Tabla 10: Cuadro comparativo de ESP32 by ESPRESSIF SYSTEMS

Características	ESP32 Series	ESP32-S2 Series	ESP32-C3 Series	ESP32-S3 Series
Lanzamiento	2016	2020	2020	2020
Núcleo	Xtensa® dual- /single core 32-bit LX6	Xtensa® single-core 32-bit LX7	32-bit single-core RISC-V	Xtensa® dual-core 32- bit LX7
Protocolos Wi-Fi	802.11 b/g/n, 2.4 GHz			
Bluetooth	Bluetooth v4.2 y BLE		Bluetooth 5.0	Bluetooth 5.0
Frecuencia	240 MHz (160 MHz for ESP32- S0WD)	240 MHz	160 MHz	240 MHz
SRAM	520 KB	320 KB	400 KB	512 KB
ROM	448 KB	128 KB	384 KB	384 KB
Flash externo	Hasta 16 MB	Hasta 1 GB	Hasta 16 MB	Hasta 1 GB
RAM externo	Hasta 8 MB	Hasta 1 GB		Hasta 1 GB
ADC	2x 12-bit	2x 12-bit	2x 12-bit	2x 12-bit
DAC	2x 8-bit	2x 8-bit		
Temporizador	4x 64-bit, 3x watchdog	4x 64-bit, 3x watchdog	2x 54-bit, 3x watchdog	4x 54-bit, 3x watchdog
Sensor de temperatura		1	1	1
Sensor Touch	10	14	x	14
Sensor Hall	1			
GPIO	34	43	22	45
SPI	4	4	3	4
Interfaz LCD	1	1		1
UART	3	2	2	3
I2C	2	2	1	2
I2S	2	1	1	2
Interfaz de cámara	1	1		1
Contador de pulsos	8	4		4
Led PWM	16	8	6	8
SDIO slave controller	1			
Ethernet MAC	1			
ULP	ULP FSM	PicoRV32 8 KB SRAM, ULP FSM		PicoRV32 8 KB SRAM, ULP FSM
Deep-sleep	100 µA	22 µA		TBD
Fuente: <a href="https://products.espressif.com/#/product-comparison">https://products.espressif.com/#/product-comparison</a>				

De acuerdo con las características detallada en la tabla 10, se ha elegido la opción de la ESP32 SERIES, el cual es una placa de desarrollo ESP32 con una serie de SOC sucesora del ESP8266. La elección se justifica por las siguientes razones técnicas:

- Presenta conectividad múltiple, útil cuando se hace uso de conectividad WI-FI para conectarse a un bróker MQTT y también ofrece poder usar funcionalidades como mDNS para la detección del dispositivo en la red. Estas tareas mencionadas requieren un módulo WI-FI eficiente y confiable.
- Nos ofrece la capacidad de uso de diversos protocolos, como el protocolo Modbus RTU usado en nuestro proyecto para la comunicación con el sensor de suelo y el protocolo MQTT usada para la transferencia de datos. Dichas capacidades se benefician del procesador de doble núcleo que permiten el manejo de tareas múltiples de manera simultánea.
- También tenemos soporte de periféricos dándonos pines GPIO suficientes para la conexión de hasta 2 LCD y la capacidad de manejar estos periféricos sin afectar otros procesos en el programa.
- Y por último al incorporar la funcionalidad de Bluetooth permite futuras expansiones del proyecto, como la adición de dispositivos móviles o el uso de sensores adicionales sin la necesidad de hardware extra. Siendo la opción con gran flexibilidad y escalabilidad.

Esta combinación de características hace que la ESP32 Series sea una elección óptima, teniendo las cualidades necesarias para cumplir los requisitos de este trabajo.

Dentro de la ESP32 SERIES, tenemos a la ESP32-DevKitC V4, que es una placa de tamaño compacto (18\*25.5\*3.1 mm) de la marca Espressif System, conocido por su versatilidad y capacidad de conectividad inalámbrica. Se elige por su capacidad de conectividad Wi-Fi el cual permite la transmisión de datos en tiempo real desde el sensor de conductividad

eléctrica del suelo a la raspberry pi y su módulo ESP32-WROOM-32D quien en su núcleo se encuentra un chip ESP32-D0WD que pertenece a la serie de chips ESP32 SERIES. El chip está diseñado para ser escalable y adaptable, contiene 2 núcleos de CPU que se pueden controlar individualmente y la frecuencia de reloj de la CPU se puede ajustar de 80 MHz a 240 MHz. Contiene además un coprocesador de bajo consumo que se puede usar en lugar de la CPU para ahorrar energía al realizar tareas que no requieren mucha potencia informática, como la supervisión de periféricos. Emplea un amplio rango de conjuntos de periféricos que van desde sensores táctiles capacitivos, interfaz de tarjeta de memoria SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C y la antena viene integrada en la placa.

También se tuvo en cuenta otras versiones de la ESP32 de otras compañías como:

- Adafruit ESP32 Feather: (compañía Adafruit)

Una ventaja de esta placa es que dispone de un conector para el uso de baterías LiPo (Litio y Polímero), lo que le permite utilizarlo para proyectos portables, ya que se le puede adjuntar una batería portátil.

Este modelo fue descartado por que la autonomía de su batería supondría una limitación uso.

- Sparkfun ESP32 Thing: (compañía Sparfun)

Lleva las mismas ventajas de que el ESP32 de Adafruit, pero de diferente compañía, puede alimentarse con baterías LiPo o un cargador que ya viene incluido.

Este modelo fue descartado porque tiene un sistema de alimentación exclusiva para la ESP32. Nuestro proyecto usa una sola fuente de alimentación para todo el sistema de lectura.

- ESP32 con soporte de batería:

Esta placa tiene características similares a las placas Sparkfun y Adafruit, solo que a esta puede acoplarse una batería LiPo tipo 18650

Las baterías 18650 tiene forma de pila AA, solo que esta es una recargable Li-ion. La diferencia es que la pila AA alcalina ofrece voltajes de 1,5/1,2V, pero la batería 18650 tiene una salida de 3,7V y una capacidad de 1600 a 3600mAh.

Esta placa con soporte de batería permite cargar también la batería 18650 y dispone de un indicador de carga fijado con LEDs

Este modelo fue descartado, por las mismas razones que las anteriores mencionadas.

- ESP32 con OLED:

Esta placa tiene una pantalla OLED de 0,96 pulgadas acoplada, no tiene un soporte de batería.

El motivo por el que este modelo fue descartado es el encapsulado, ya que el ambiente en el que se expone nuestro proyecto hay tierra, humedad y otros factores que pueden estropearlo.

- **Raspberry Pi:** es una computadora de bajo costo y con un tamaño compacto, del tamaño aproximado de una tarjeta de crédito el cual puede ser conectado a un monitor o una tv, y usarse con un teclado y mouse estándar. Utiliza un sistema operativo Linux capaz de permitir correr lenguajes de programación como Scratch y Python.

A lo largo del tiempo han salido una amplia variedad de modelos, siendo las más ultimas tomadas desde el año 2018 las siguientes:

Tabla 11: Cuadro comparativo de Raspberry Pi 3B+, 4B, 400 y 5

	Raspberry Pi 3B+	Raspberry Pi 4B	Raspberry Pi 400	Raspberry Pi 5
Año	2018	2019	2020	2023
Procesador	64 bits ARM Cortex-A53	64 bits ARM Cortex-A72	Quad-core 64-bits Cortex A-72	64 bits ARM cortex-A76
Frecuencia de reloj	1.4GHz	1.8 GHz	1.8 GHz	2.4GHz
Memoria RAM	1Gb LPDDR2 SDRAM	1/2/4/8 Gb	4 Gb	4/8 Gb
Conectividad	2.4GHz / 5GHz, IEEE 802.11.b/g/n/a c, Bluetooth 4.2, BLE	2.4 GHz y 5.0 GHz (IEEE 802.11b/g/n/a c), Bluetooth 5.0, BLE	WiFi Bluetooth y puerto Ethernet	Wi-Fi de doble banda 802.11ac y Bluetooth 5.0 / BLE
Puertos	1 HDMI, 1 conector Ethernet, 4 USB 2.0, 1 conector de audio de 3.5mm	2 USB 2.0, 2 USB 3.0, 2 micro HDMI, 1 Ethernet, 1 conector de audio de 3.5mm.	1 USB 2.0  2 USB 3.0	2 USB 2.0, 2 USB 3.0, 2 MIPI de 4 líneas de transmisión, PCLE 2.0 x 1 para conexión de periférico de alta velocidad a través de tarjetas adaptadoras como HATs u otros.
Almacenamie nto	MicroSD	MicroSD	MicroSD	MicroSD SDR104

GPIO	40 pines	40 pines	40 pines	40 pines
Otros	Puerto DSI para cámaras, pantalla	Puerto DSI para cámaras, pantalla	2 salidas de video en resolución 4K, incluye teclado	reloj incorporado a tiempo real (RTC) alimentado por batería externa, conector dedicado para ventilador incorporado, conector dedicado para monitor serial, botón de encendido.
Alimentación	5V/2.5 A (3.5 W) vía micro-USB	5V por puerto USB-C (mínimo requerido 3A)	5V y 3A con conector USB-C	27W-5V/5A (recomendados) via puerto USB-C, con soporte para Power Delivery
Precio	S/178.00	S/374.50	S/378.50	S/399.00
Fuente: <a href="https://thepibox.pe">https://thepibox.pe</a>				

De acuerdo con las características detalladas en la tabla 11 sobre los diferentes modelos de Raspberry Pi en el mercado desde el 2018, se llegó a la conclusión de usar la Raspberry Pi 5 por su gran capacidad de procesamiento y flexibilidad para integrar sistemas IoT en entornos de campo. El procesador ARM Cortex-A76 a 2.4GHz y la memoria RAM LPDDR4X de hasta 8gb permiten un manejo de múltiples flujos de datos de manera simultánea, siendo fundamental para un sistema de monitoreo.

En comparación con los modelos anteriores como la Raspberry Pi 3 y 4, el Raspberry Pi 5 nos ofrece un salto considerable en conectividad y rendimiento, con el Wi-Fi 802.11ac y además el Bluetooth 5.0/BLE,

asegurando una transmisión de datos estable y de manera rápida en entornos remotos. Por otro lado, el soporte para tarjetas microSD de alta velocidad SDR104 nos asegura que los datos del sensor de conductividad eléctrica de los suelos puedan ser almacenados y recuperados rápidamente si se desea realizar una integración de una base de datos dentro del mismo Raspberry Pi.

Otro punto clave es la expansión a través del puerto PCIe, que nos permite conectarnos con dispositivos adicionales para futuras ampliaciones, como el uso de nuevos sensores, módulos HATs para análisis de datos.

Para finalizar, el reloj en tiempo real o por sus siglas RTC y el eficiente consumo energético de 27 W garantizan que el raspberry Pi 5 sea viable para proyectos de largo plazo en entornos con recursos limitados.

- **Sensor de conductividad eléctrica del suelo**

Los sensores de conductividad eléctrica a usar se fundamentan en la medida de la resistencia que el suelo opone al paso de la corriente eléctrica. Estas exigen un buen contacto entre el suelo y el sensor. A continuación, se presentan 3 modelos a comparación que se pueden usar:

Tabla 12: Cuadro comparativo de sensores de conductividad eléctrica del suelo

Marca Especificaciones	Halisense	ComWinTop	Rika Electronic Technology
Modelo	TH-EC	THC-S	RK520-02
Rango de medición	0 – 20 000 uS/cm	0 – 20000 uS/cm	0 – 10 mS/cm
Precisión	±3% (0-10000uS/cm); ±5% (10000-20000uS/cm)	±3% (0-10000 uS/cm); ±5% (10000-20000 uS/cm)	±3% FS (Full Scale)
Estabilidad a largo plazo	≤ 1% uS/cm	≤ 0.1% dS/m	-

Tiempo de respuesta	≤ 1s	≤ 1s	< 1s
Output	RS485	RS485	RS485
Alimentación	4.5 – 30VDC	4.5 – 30VDC	5VDC, 12 – 24VDC
Consumo de energía	0.5W o 24V CC	0.5W o 24V CC	-
Clase de protección	IP68	IP68	IP68
Longitud de cable	2m	2m	-
Temperatura de operación	-40°C a 80°C	-40°C a 80°C	-40°C a 80°C
Dimensiones	45*15*123mm	45*15*123mm	45*15*145mm
Material de la sonda	Acero inoxidable 316L	Acero inoxidable 316L	Acero inoxidable 316L
Precio	S/67.55	S/86.20	S/425
Fuente	aliexpress.com	aliexpress.com	aliexpress.com

El sensor elegido de acuerdo con la tabla 12 es el modelo THC-S de la marca ComWinTop por su equilibrio entre prestaciones técnicas, costo y compatibilidad con el protocolo Modbus RTU. Es ideal para aplicaciones en entornos agrícolas exigentes.

Además, el sensor tiene la capacidad de ser calibrado, permitiendo el ajuste de sus mediciones a condiciones reales, asegurando lecturas precisas y confiables de sus parámetros: la conductividad eléctrica, la humedad y la temperatura del suelo.

En la ilustración 10 se muestran los parámetros por defecto que usa el sensor para comunicarse y sus direcciones de registros para realizar la lectura y escritura.

**RS485 communication**

Default parameters: 4800,n,8,1

Default device address is 1

Modbus RTU protocol

Read status registers, read function code: 0x30					
Register address (Hex)	PLC Address (decimal)	meaning	Number of bytes	content	remark
0000	40001	Humidity	2	0.1%RH	read
0001	40002	Temperature	2	0.1℃	read
0002	40003	Conductivity	2	1	read
0003	40004	Salinity	2	1	read
0004	40005	TDS	2	1	read
0022	40035	Conductivity factor	2	0-100 correspond to 0.0%-10.0% Default 0.0%	read / write
0023	40036	Salinity factor	2	0-100 correspond to 0.00-1.00 Default 55 (0.55)	read / write
0024	40037	TDS factor	2	0-100 correspond to 0.00-1.00 Default 50 (0.5)	read / write
0050	40081	Temperature offset	2	0.1	read / write
0051	40082	Humidity offset	2	0.1	read / write
0052	40083	Conductivity offset	2	1	read / write
Parameters registers, read function code: 0x30, write function code: 0x60					
07D0	42001	Slave ID	2		1-254
07D1	42002	baud rate	2		0: 2400 1: 4800 2: 9600 Default 4800

Ilustración 10: Parámetros de configuración y calibración del sensor THC-S

## F. Diseño del circuito de conexión del sensor CWT-Soil-THC-S

En el desarrollo del trabajo, se diseñó el circuito de lectura para integrar el sensor de suelo modelo THC-S de la marca ComWinTop con el módulo Max485 y la ESP32, alimentados con un convertidor de voltaje DC-DC Step-Down 2A MP1584EN.

Para garantizar el funcionamiento correcto del sistema propuesto, el diseño de la placa se realizó utilizando el software de simulación Proteus, herramienta ampliamente reconocida en el ámbito de la ingeniería electrónica por su capacidad para modelar y simular circuitos eléctricos de manera precisa antes de su implementación física. Previo se probó haciendo uso de un protoboard.

El circuito integra los siguientes componentes y configuraciones:

- **Esp32 devkitc v4:** es la unidad central de procesamiento y comunicación utilizando los siguientes pines:
  - GPIO 4: pin seleccionado para el control del modo transmisión/recepción del módulo Max485.
  - GPIO 18: pin seleccionado para la entrada RX, encargado de recibir los datos del sensor.
  - GPIO 19: pin seleccionado para el envío de comandos, para poder modificar si se desea el Slave ID, el baud rate, o puedes realizar escritura en los registros para cambiar factores de la conductividad, salinidad y TDS, útil para la calibración del sensor.
- **Módulo Max485:** este módulo es el puente entre el sensor y el ESP32, encargado de convertir la señal con el protocolo modbusRTU del sensor al nivel lógico compatible del ESP32 el TTL, asegurando así una comunicación fiable.

- **Sensor THC-S:** como pieza primaria del proyecto, es la encargada de suministrar los datos de conductividad eléctrica del suelo del terreno agrícola.
- **Convertidor DC-DC Step-Down Mp1584EN:** pieza utilizada para la regulación y estabilización de la alimentación de 5v requerida para el módulo Max485 y el ESP32, con un voltaje de alimentación de entrada de 24V 2A.
- **Cálculo de la corriente:** para realizar el cálculo de la corriente es necesario considerar las especificaciones de alimentación de los componentes anteriores.
  - La ESP32 devkitc V4: con un consumo de voltaje y corriente de 4V-12V (5V a 9V recomendado) y 80mA-500mA máx. respectivamente.
  - El módulo Max485: con un consumo de voltaje y corriente de 5V y 500 uA máx. respectivamente.
  - Sensor THC-S: con un consumo de voltaje y potencia de 24V y 0.5W respectivamente.
  - Convertidor DC-DC MP1584EN: acepta un voltaje de entrada entre 4.5V a 28V con una salida de voltaje entre 0.8V a 20V y una corriente de salida de 1A recomendado, 3A máximo. Tiene una eficiencia de conversión del 96%.

Ahora teniendo en cuenta la eficiencia del conversor y un voltaje de entrada de 24v DC, se calcula la corriente requerida para el ESP32 y el módulo max485 con la formula siguiente:

$$I_{entrada} = \frac{I_{salida} \times V_{salida}}{\eta \times V_{entrada}} \left\{ \begin{array}{l} \text{Ecuación 1} \end{array} \right.$$

Donde:

- $\eta$  = eficiencia del convertidor de voltaje.
- $V_{entrada}$  = voltaje suministrado al conversor de voltaje.
- $V_{salida}$  = voltaje de salida que se requiere configurar.
- $I_{salida}$  = corriente de consumida por los componentes alimentados por el convertidor MP1584EN.

Siendo ambos componentes (ESP32 devkitc V4 y módulo Max485) alimentados con un voltaje de 5V DC y una corriente de 500mA y 500uA respectivamente, tenemos una entrada de:

$$I_{entrada} = \frac{500.5mA \times 5V DC}{0.96 \times 24V DC}$$

$$I_{entrada} = 108.615 mA$$

Obtenemos que la corriente de entrada requerida para alimentar al ESP32 devkitc V4 y el módulo Max485 es de 108.615 miliamperios.

Con este dato procedemos a calcular la corriente final que requiere el circuito de lectura junto con el consumo del sensor de suelo.

$$I_{final} = I_{entrada} + I_{sensor}$$

$$I_{final} = 108.615 mA + \frac{0.5W}{24V}$$

$$I_{final} = 108.636 mA$$

Ahora consideramos un margen de seguridad del 40% a la corriente final obteniendo como resultado:

$$I_{total} = 108.636 mA \times 1.4$$

$$I_{total} = 152.09 mA$$

En conclusión, una fuente de 24V debe suministrar una corriente de al menos 160mA. Por lo tanto, para mayor estabilidad, se usa una fuente de 24v y 0.5A a 1A.

- **Placa del circuito final diseñado en el programa proteus.**

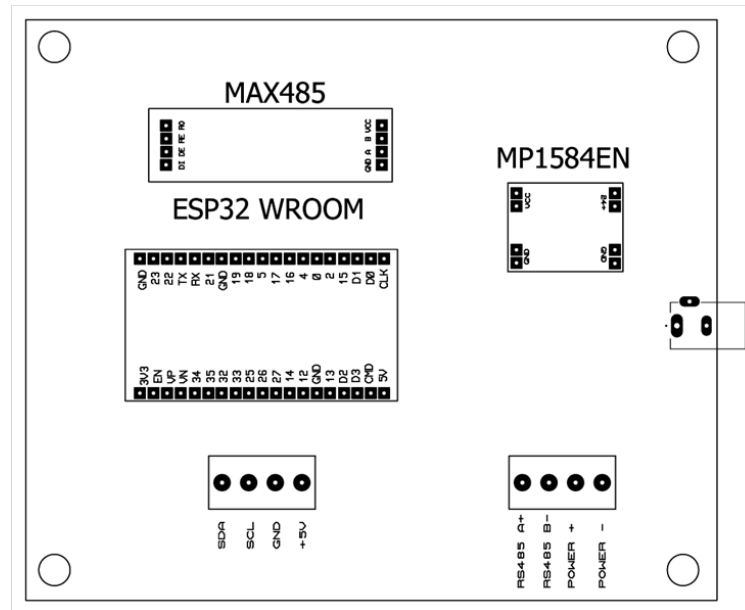


Ilustración 11: Frontal del circuito de lectura y envío de datos

Fuente: Propia

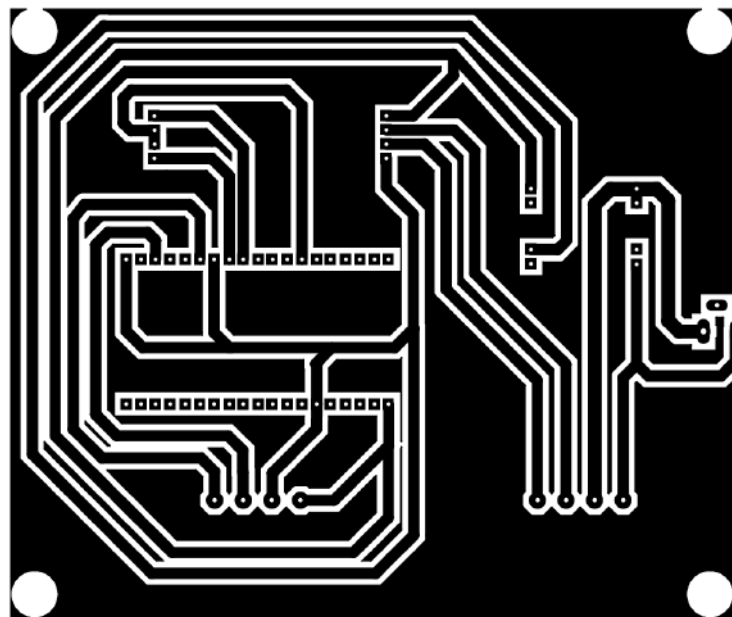


Ilustración 12: Posterior del circuito de lectura y envío de datos

Fuente: Propia

## G. Programación del módulo ESP32

El código se ha realizado usando el programa Arduino IDE y los parámetros para la configuración y calibración del sensor se localizan en la ilustración 10.

El programa controla el ESP32 que se conecta a una red wifi, lee datos del sensor de suelo CWT-SOIL-THC-S, muestra la información en una pantalla LCD y la envía a un servidor MQTT. Si se pierde la conexión wifi o MQTT, el programa se reconecta automáticamente.

Se ha tenido que usar librerías, que son como cajas de herramientas listas para usar. Las que hemos usado son:

### LIBRERÍAS PARA LA CONFIGURACIÓN INICIAL

```
#include <ESPmDNS.h>
```

```
#include <PubSubClient.h>
```

```
#include <WiFi.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <ModbusMaster.h>
```

- `#include <ESPmDNS.h>` incluye la librería que permite que el ESP32 se identifique en la red con el nombre designado por el usuario.
- `#include <PubSubClient.h>` agrega una librería que se usa para conectar el ESP32 con un servidor MQTT, sirve para enviar datos a otros dispositivos o sistemas.
- `#include <WiFi.h>` esta librería permite que nuestro ESP32 se conecte a internet.
- `#include <LiquidCrystal_I2C.h>` con esta librería se controla una pantalla LCD donde mostramos información.
- `#include <ModbusMaster.h>` Permite la conexión con el sensor de suelo usando un protocolo llamado modbusRTU.

## CONFIGURACIÓN DEL TRANSEPTOR MAX 485

```
#define dere_max485 4
```

```
#define ro_max485 18
```

```
#define di_max485 19
```

```
#define Serial_max485 4800
```

- `#define dere_max485 4` hace uso del pin N°4 del ESP32 que indica si el transceptor MAX 485 debe enviar o recibir datos.
- `#define ro_max485 18` hace uso del pin N°18, donde el ESP32 recibe datos del sensor a través del MAX485.
- `#define di_max485 19` hace uso de los pines N°19, donde el ESP32 envía datos al sensor a través del MAX485.
- `#define Serial_max485 4800` esto indica que la velocidad de comunicación con el sensor será de 4800 bits por segundo.

## CONFIGURACIÓN DE LA CONEXIÓN MQTT

```
#define broker_mqtt "raspberrypi5.local"
```

```
#define puerto_mqtt 1883
```

- `#define broker_mqtt "raspberrypi5.local"` se define como `broker_mqtt` a la dirección "raspberrypi5.local" que es la dirección del servidor MQTT, en este caso una raspberry pi 5.
- `#define puerto_mqtt 1883` es el puerto estándar utilizado por el protocolo MQTT sin cifrado.

## CONFIGURACIÓN DE LA PANTALLA LCD

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

- `LiquidCrystal_I2C lcd(0x27, 16, 2);` configura la pantalla LCD conectada al ESP32 usando el protocolo I2C. La dirección del LCD es 0x27 y tiene 16 columnas y 2 filas.

```
byte wifiSignal[8] = { ... };
```

```
byte mqttCloud[8] = { ... };
```

```
byte noConnection[8] = { ... };
```

- `byte wifiSignal[8] = { ... };` para la señal Wi-Fi
- `byte mqttCloud[8] = { ... };` para la conexión MQTT
- `byte noConnection[8] = { ... };` para la desconexión

Aquí es donde se define los gráficos personalizados que se usarán en la pantalla LCD para mostrar el estado de wifi, MQTT o errores.

## DEFINICIÓN DEL ARREGLO DE REGISTROS MODBUS

```
uint16_t data_register[5] = { 0x0000,0x0001, 0x0002, 0x0003, 0x0004 };
```

- `uint16_t data_register` define los registros modbus que el sensor de suelo usa para reportar:
  - 0x0000: humedad
  - 0x0001: temperatura
  - 0x0002: conductividad eléctrica (EC)
  - 0x0003: salinidad
  - 0x0004: TDS (Sólidos Totales Disueltos)

## CREACIÓN DEL OBJETO DE COMUNICACIÓN MODBUS

```
ModbusMaster modbus;
```

- `ModbusMaster modbus;` representa el cliente de modbus que interactuará con el sensor.

## VARIABLES PARA TIEMPO DE MUESTREO

```
unsigned long last_read_time = 0;
```

```
const long sampling_time = 2000;
```

```
unsigned long wifi_retry_interval = 10000;
```

```
unsigned long last_wifi_attempt = 0;
```

- `last_read_time` registra el último tiempo de muestreo.
- `sampling_time` es el tiempo entre lecturas, que en este caso es de 2 segundos.
- `wifi_retry_interval` es el tiempo entre intentos de reconexión Wi-fi, que en este caso son de 10 segundos.
- `last_wifi_attempt` es el último intento de conexión.

## ESTADO DE CONEXIÓN WI-FI

```
bool wifi_connected = false;
```

- `wifi_connected` es el estado de conexión.

## **FUNCIONES AUXILIARES PARA MODBUS**

```
void modbusPreTransmission() {  
    digitalWrite(dere_max485, HIGH); }
```

```
void modbusPostTransmission() {  
    digitalWrite(dere_max485, LOW); }
```

- Estas funciones controlan el pin `dere_max485`, para cambiar entre modo de transmisión y recepción
  - HIGH pone el módulo MAX485 en modo transmisión
  - LOW lo pone en modo de recepción

## **CONFIGURACIÓN DE REDES WI-FI**

```
const char* ssid[] = { "USUARIO-0", "USUARIO-1", "USUARIO-2" };
```

```
const char* password[] = { "CONTRASEÑA", "admin-1", "admin-2" };
```

Se crea un arreglo de punteros a cadena de caracteres donde estas son la lista de nombres de redes wifi y sus contraseñas, en el cual fue usado USUARIO-0 como usuario y CONTRASEÑA como la contraseña.

## **CONFIGURACIÓN INICIAL DEL SETUP**

```
void setup () {  
    Serial.begin(4800);  
    lcd.init();  
    lcd.backlight();
```

- `Serial.begin(4800);` inicia la comunicación serial.
- `lcd.init();` Inicializa el LCD
- `lcd.backlight();` enciende la luz de fondo

## CREACIÓN DE CARACTERES PERSONALIZADOS PARA EL LCD

```
lcd.createChar(0, wifiSignal);
```

```
lcd.createChar(1, mqttCloud);
```

```
lcd.createChar(2, noConnection);
```

- `lcd.createChar(0, wifiSignal);` "0" es el número que identificará este carácter especial en la memoria de la pantalla. `WifiSignal` es un arreglo de datos que define cómo se verá el dibujo, en este caso, es un símbolo que representa la señal de Wi-fi
- `lcd.createChar(1, mqttCloud);` "1" es el número de carácter especial. `MqttCloud` es otro arreglo que define un dibujo, en este caso una nube que representará el estado del servidor MQTT.
- `lcd.createChar(2, noConnection);` "2" es el carácter especial. `noConnection` es el símbolo que indica que no hay conexión ya sea de Wi-Fi y MQTT.

## MENSAJE DE INICIACIÓN:

```
lcd.setCursor(0, 0);
```

```
lcd.print("Iniciando...");
```

```
delay(2000);
```

```
lcd.clear();
```

- `lcd.setCursor(0, 0);` indica dónde empezar a escribir en la pantalla LCD. 0,0 significa "empezar desde la columna 0, fila 0" que es la esquina superior izquierda de la pantalla.
- `lcd.print("Iniciando...");` escribe el texto "iniciando..." en la pantalla LCD, empezando desde la posición establecida con `lcd.setCursor()`.
- `delay(2000);` Pausa el programa durante 2 segundos. Esto permite que el texto "Iniciando..." quede visible en la pantalla por ese tiempo antes de borrarlo.

- `lcd.clear();` Borra todo el contenido de la pantalla LCD, deja la pantalla vacía para que se pueda mostrar nueva información más adelante.

## CONFIGURACIÓN DEL PIN DE CONTROL DEL TRANCEPTOR RS485

```
pinMode(dere_max485, OUTPUT);
```

```
digitalWrite(dere_max485, LOW);
```

- `pinMode(dere_max485, OUTPUT);` configura el pin específico del microcontrolador, en este caso de la ESP32 como entrada o salida. `dere_max485` es el nombre de la variable asociada al pin, que en este caso es el pin número 4. `OUTPUT` indica que el pin está configurado como salida, es decir, que enviará una señal.
- `digitalWrite(dere_max485, LOW);` escribe un valor alto o bajo en el pin especificado. `Dere_max485` es el pin que se ha configurado como salida y `LOW` es el valor bajo (0v) del pin, es decir que envía una señal baja a ese pin.

## INICIALIZACIÓN DE LA COMUNICACIÓN SERIE PARA RS485

```
Serial2.begin(Serial_max485, SERIAL_8N1, ro_max485, di_max485);
```

```
Serial2.setTimeout(1000);
```

- `Serial2.begin(Serial_max485, SERIAL_8N1, ro_max485, di_max485);` Inicializa la comunicación con el sensor a través del puerto serial2
- `Serial2.setTimeout(1000);` establece el tiempo de 1 segundo para esperar la respuesta del sensor. Si el sensor no responde en ese tiempo, el programa continuará ejecutándose sin esperar de forma indefinida.

## INICIALIZACIÓN Y CONFIGURACIÓN DEL PROTOCOLO MODBUS

```
modbus.begin(1, Serial2);
```

```
modbus.preTransmission(modbusPreTransmission);
```

```
modbus.postTransmission(modbusPostTransmission);
```

- `modbus.begin(1, Serial2);` este comando inicia la comunicación Modbus. “1” es la dirección de esclavo del dispositivo Modbus, en este caso, el ESP32 está configurado como el esclavo número 1 en la red Modbus. Serial2 es el puerto serial que se va a utilizar para la comunicación Modbus, que es el Serial2 que se configuró antes para comunicarse con el sensor. Con esta línea se indica que ESP32 va a actuar como un esclavo de Modbus utilizando el puerto Serial2 para comunicarse con el dispositivo principal
- `modbus.preTransmission(modbusPreTransmission);` Este es un callback o devolución de llamada, que generalmente se utiliza para preparar el transceptor RS485 antes de enviar datos.
- `modbus.postTransmission(modbusPostTransmission);` define una función postransmisión como callback. Esta función se ejecutará inmediatamente después de que el dispositivo haya terminado de enviar datos.

## CONFIGURACIÓN DE CONEXIÓN WI-FI Y CLIENTE MQTT

```
setup_wifi();
```

```
client.setServer(broker_mqtt, puerto_mqtt);
```

- `setup_wifi();` es una función que configura la conexión Wi-Fi del dispositivo, permitiendo que se conecte a una red inalámbrica.
- `client.setServer(broker_mqtt, puerto_mqtt);` configura el cliente MQTT para que se conecte a un broker MQTT. Este broker actúa como

intermediario en la comunicación entre dispositivos dentro de una red IoT.

### **CALIBRACIÓN DEL SENSOR:**

Creamos las siguientes variables en el entorno global del Arduino IDE

```
uint16_t factorCE = 0x0022;  
uint16_t offsetCE = 0x0052;  
//uint16_t SALfactor = 0x0023;  
uint16_t TDSfactor = 0x0024;  
uint16_t factorvalor=70;  
uint16_t offsetvalor=250;  
//uint16_t SALvalor = 55;  
uint16_t TDSvalor = 64;
```

y dentro del setup() integramos la escritura al sensor de los parámetros configurados.

```
uint8_t configurar = modbus.writeSingleRegister(factorCE, factorvalor);  
configurar = modbus.writeSingleRegister(offsetCE, offsetvalor);  
//configurar = modbus.writeSingleRegister(SALfactor, SALvalor);  
configurar = modbus.writeSingleRegister(TDSfactor, TDSvalor);
```

Usamos la función de la biblioteca modbusRTU:

- `configurar = modbus.writeSingleRegister(factorCE, factorvalor);` esta función escribe un valor en un registro específico, en este caso el registro 0x0022 que corresponde al factor de la conductividad. Su sintaxis es la siguiente: `uint8_t name = writeSingleRegister(uint16_t, uint16_t);`

## CONFIGURACIÓN DE LA CONEXIÓN WI-FI

```
void setup_wifi() {  
  
int num_redes = sizeof(ssid) / sizeof(ssid[0]);
```

- `void setup_wifi() {` se utiliza para gestionar la conexión del ESP32 a una red Wi-fi. Su propósito es iterar por una lista de redes disponibles y tratar de conectarse a alguna de ellas.
- `int num_redes = sizeof(ssid) / sizeof(ssid[0]);` la parte de `sizeof(ssid)` calcula el tamaño total del array `ssid` en bytes. La parte de `sizeof(ssid[0])` representa el tamaño de un único elemento del array. Dividiendo ambas se obtiene el total de elementos en el array, es decir, el número de redes disponibles.

## BUCLE PARA INTENTAR LA CONEXIÓN A REDES WI-FI DISPONIBLES

```
for (int r = 0; r < num_redes; r++) {
```

- `for (int r = 0; r < num_redes; r++) {` es un bucle que itera a través del índice de las redes disponibles almacenadas en el arreglo `ssid`. El comando `num_redes` calcula el número total de redes usando un código previo `int num_redes = sizeof(ssid) / sizeof(ssid[0]);`

## INTENTO DE CONEXIÓN WI-FI CON TEMPORIZADOR

```
WiFi.begin(ssid[r], password[r]);  
  
unsigned long redtime = millis();  
  
while (WiFi.status() != WL_CONNECTED && millis() - redtime < 2000) {  
  
delay(500);
```

- `WiFi.begin(ssid[r], password[r]);` es una función de la biblioteca `WiFi.h` que inicia el intento de conexión del ESP32 a una red Wi-Fi. `ssid[r]` es el nombre de la red Wi-Fi que se intenta conectar, `password[r]` es la contraseña correspondiente a esa red.

- `unsigned long redtime = millis();` devuelve el tiempo actual en milisegundos desde que el ESP32 comenzó a ejecutarse, almacena en `redtime` el momento exacto en el que comienza el intento de conexión, para establecer un límite de tiempo para esta acción.
- `while (WiFi.status() != WL_CONNECTED && millis() - redtime < 2000) {` el bucle `while` mantiene activo el intento de conexión Wi-Fi durante un máximo de 2 segundos. `Wifi.status() !=WL_CONNECTED`, verifica si el estado de Wi-Fi no es `WL_connected`, lo que significa que aún no se ha conectado a la red.
- `delay(500);` Pausa la ejecución del programa durante 500 milisegundos, esto evita que el bucle `while` consuma excesivos recursos al verificar continuamente el estado de conexión.

## VERIFICACIÓN DE CONEXIÓN WIFI EXITOSA

```
if (WiFi.status() == WL_CONNECTED) {
```

- `if (WiFi.status() == WL_CONNECTED) WiFi.status ()` devuelve el estado actual de la conexión Wi-Fi, `WL_CONNECTED` es una constante que indica que el dispositivo está conectado correctamente a una red Wi-Fi y ha recibido una dirección IP válida. El propósito de esta línea de código es verificar si el ESP32 ha logrado establecer la conexión Wi-Fi.

## INFORMACIÓN DE CONEXIÓN WI-FI MOSTRADA EN PANTALLA LCD

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Conectado a: ");
```

```
lcd.setCursor(0, 1);
```

```
lcd.print(ssid[r]);
```

```
delay(3000);
```

```
lcd.clear();
```

- `lcd.clear();` borra el contenido actual de la pantalla LCD, para mostrar los nuevos datos.
- `lcd.setCursor(0, 0);` esta función posiciona las coordenadas del cursor en la fila y columna 0
- `lcd.print("Conectado a: ");` Imprime el texto "Conectado a:" en la pantalla LCD, mostrando así un mensaje de conexión exitosa.
- `lcd.setCursor(0, 1);` mueve el cursor a la primera columna de la segunda fila de la pantalla
- `lcd.print(ssid[r]);` Imprime el nombre de la red Wi-Fi (SSID) a la que el ESP32 se ha conectado. `ssid[r]` contiene las redes Wi-Fi disponibles.
- `delay(3000);` detiene la ejecución del programa durante 3 segundos
- `lcd.clear();` Una vez que ha pasado el tiempo de 3 segundos, se limpia la pantalla LCD nuevamente.

## INDICACIÓN DE QUE LA CONEXIÓN WI-FI HA SIDO UN ÉXITO

```
wifi_connected = true;
```

- `wifi_connected = true;` indica que el ESP32 está conectado con éxito a una red Wi-Fi, evitando intentos de reconexión y asegurando un flujo eficiente en la comunicación.

## MENSAJE DE ERROR SE MUESTRA EN PANTALLA SI LA CONEXIÓN WI-FI FALLA

```
} else {  
  
    lcd.clear();  
  
    lcd.setCursor(0, 0);  
  
    lcd.print("Fallo Conexión");  
  
    delay(1000);
```

```
lcd.clear();
```

- `} else {` este bloque else se ejecuta si la condición previa: `if (WiFi.status() == WL_CONNECTED)` no se cumple, es decir, el ESP no logra conectarse a la red Wi-Fi después de un intento.
- `lcd.clear();` limpia la pantalla LCD
- `lcd.setCursor(0, 0);` coloca el cursor en la columna y fila 0
- `lcd.print("Fallo Conexión");` muestra el mensaje "Fallo Conexión" en la pantalla LCD, indicando visualmente al usuario que el intento de conexión a la red actual no fue exitoso.
- `delay(1000);` detiene la ejecución del programa durante 1 segundo, esto asegura que el usuario tenga tiempo suficiente para leer el mensaje en la pantalla antes de que la información sea borrada o cambiada.
- `lcd.clear();` limpia la pantalla LCD nuevamente después de mostrar el mensaje de error.

## MENSAJE DE SIN CONEXIÓN SI NO HAY WI-FI

```
if (!wifi_connected) {  
    lcd.setCursor(0, 0);  
    lcd.print("Sin conexión");  
    delay(1000);  
    lcd.clear();  
}
```

- `if (!wifi_connected) {` indica que el estado de la variable `wifi_connected` logró establecer conexión Wi-Fi. `!wifi_connected` evalúa a `true` si la variable `wifi_connected` es `false`, lo que significa que no se estableció conexión con ninguna red Wi-Fi.
- `lcd.setCursor(0, 0);` coloca el cursor en la columna 0 y fila 0 de la pantalla LCD.
- `lcd.print("Sin conexión");` muestra el mensaje “Sin conexión” en la pantalla LCD, para indicar que el dispositivo no está conectado.
- `delay(1000);` Detiene la ejecución del programa durante 1 segundo para dar tiempo a al usuario a que lea el mensaje.
- `lcd.clear();` limpia la pantalla LCD.

## INICIALIZACIÓN DE mDNS PARA EL ESP32

```
if (!MDNS.begin("ESP32-SOILSENSOR")) {  
    Serial.println("Error al iniciar mDNS");  
    return;  
}
```

- `if (!MDNS.begin("ESP32-SOILSENSOR")) {` inicia el servicio mDNS en el ESP32, esto permite que el dispositivo sea identificado en la red local, en este caso `SOILSENSOR.local`
- `Serial.println("Error al iniciar mDNS");` imprime en el monitor el mensaje de “Error al iniciar mDNS”

- `return;` finaliza la ejecución en el que se encuentra y evita que el programa continúe con configuraciones que dependan del servicio mDNS, ya que no está operativo.

## CONFIRMACIÓN DE INICIO EXITOSO DE mDNS

```
Serial.println("mDNS iniciado como ESP32-SOILSENSOR.local")
```

- `Serial.println("mDNS iniciado como ESP32-SOILSENSOR.local")` imprime en la pantalla LCD el mensaje “mDNS iniciado como ESP32-SOILSENSOR.local”

## RECONEXIÓN AL BROKER MQTT

```
void reconnect() {
    int retryCount = 0;
    const int maxRetryCount = 10;
    if (!client.connected()) {
        Serial.print("Conectando al broker MQTT...");
        if (client.connect("ESP32Client")) {
            Serial.println("Conectado");
        } else {
            Serial.print("Fallo en la conexión. Código de error: ");
            Serial.println(client.state());
            retryCount++;
            long backoffTime = pow(2, retryCount) * 1000;
            Serial.print("Esperando ");
            Serial.print(backoffTime / 1000);
```

```
Serial.println(" segundos antes de intentar reconectar al broker.");  
delay(backoffTime);
```

- `void reconnect() {`
- `int retryCount = 0;` lleva el conteo del número de intentos de reconexión consecutivos al broker MQTT
- `const int maxRetryCount = 10;` establece un límite después del cual se aplicará un mayor tiempo de espera antes de reintentar.
- `if (!client.connected())` comprueba si el cliente MQTT está desconectado. Si está desconectado intenta establecer una nueva conexión con el broker.
- `Serial.print("Conectando al broker MQTT...");`
- `if (client.connect("ESP32Client"))` intenta conectar al broker MQTT usando el identificador "ESPClient", retorna true si la conexión es exitosa.
- `Serial.println("Conectado");` imprime en la pantalla que la conexión fue exitosa.
- `} else {` alternativa para indicar si la conexión falla
- `Serial.print("Fallo en la conexión. Código de error: ");` muestra en pantalla "Fallo en la conexión" junto con el estado de error proporcionado por el `client.state()`
- `Serial.println(client.state());`
- `retryCount++;` Incrementa el contador de reintentos
- `long backoffTime = pow(2, retryCount) * 1000;` calcula el tiempo de espera en milisegundos utilizando una fórmula de backoff exponencial, evitando así sobrecarga de intentos de reconexión simultáneos.
- `Serial.print("Esperando ");` muestra en el monitor el mensaje "Esperando" lo que indica que el dispositivo está esperando antes de intentar reconectar al broker MQTT

- `Serial.print(backoffTime / 1000);` muestra en la pantalla el tiempo de espera calculado en segundos.
- `Serial.println("segundos antes de intentar reconectar al broker.");` muestra en el monitor el mensaje “segundos antes de intentar reconectar al broker.”
- `delay(backoffTime);` es una función de Arduino que detiene la ejecución del programa durante un tiempo determinado por `backoffTime`.

## CONTROL DE INTENTOS FALLIDOS Y ESPERA PROLONGADA

```
if (retryCount > maxRetryCount) {
    Serial.println("Demasiados intentos fallidos, esperando más tiempo.");
    retryCount = 0;
    delay(60000);
}
```

- `if (retryCount > maxRetryCount) {` verifica si el número de intentos fallidos de conexión `retryCount` ha superado un límite predefinido, que se define en la variable `maxRetryCount`, que es una constante que define el número máximo de intentos fallidos permitidos antes de aplicar una pausa más larga, en este caso, el valor de `maxRetryCount` está definido como 10
- `Serial.println("Demasiados intentos fallidos, esperando más tiempo.");` si la condición `if` es verdadera, se muestra un mensaje en la pantalla “Demasiados intentos fallidos, esperando más tiempo.”
- `retryCount = 0;` restablece el contador de intentos fallidos a cero, lo que significa que después de haber alcanzado el número máximo de intentos fallidos, el contador se reinicia y empieza a contar nuevamente desde cero cuando se reanuda el ciclo de intentos.
- `delay(60000);` detiene la ejecución del programa durante el número de milisegundos especificado, en este caso, el valor es de 1 minuto.

## REINICIO DEL CONTADOR DE INTENTOS DE RECONEXIÓN

```
retryCount = 0;
```

- `retryCount = 0;` restablece el contador a 0 si logra conectarse.

## MANEJO DE LA CONEXIÓN WI-FI Y RECONEXIÓN AUTOMÁTICA

```
void loop() {
```

```
  client.loop();
```

```
  if (!wifi_connected && millis() - last_wifi_attempt >= wifi_retry_interval) {
```

```
    last_wifi_attempt = millis();
```

```
    setup_wifi();
```

- `client.loop();` mantiene la conexión MQTT activa y gestiona la recepción y envío de mensajes.
- `if (!wifi_connected && millis() - last_wifi_attempt >= wifi_retry_interval) {` verifica si el dispositivo no está conectado a Wi-Fi y si ha pasado el tiempo suficiente desde el último intento de conexión para permitir otro intento.
- `last_wifi_attempt = millis();` actualiza el tiempo del último intento de conexión.
- `setup_wifi();` es la función encargada de gestionar la reconexión Wi-Fi si es necesario.

## ESTADO DE CONEXIÓN WI-FI SE MUESTRA EN LCD

```
if (WiFi.status() == WL_CONNECTED) {  
    lcd.setCursor(7, 0);  
    lcd.write(byte(0));  
} else {  
    lcd.setCursor(7, 0);  
    lcd.write(byte(2));  
    wifi_connected = false;
```

- `if (WiFi.status() == WL_CONNECTED) {` devuelve el estado actual de la conexión Wi-Fi del ESP32. `WL_connected` indica que el dispositivo está conectado a una red Wi-Fi.
- `lcd.setCursor(7, 0);` coloca el cursor en la posición de la columna 7 y la fila 0 del LCD
- `lcd.write(byte(0));` escribe el carácter personalizado almacenado en el índice 0 de la memoria del LCD, Según el código inicial `byte(0)` representa la barra de señal Wi-Fi creada previamente con `lcd.createChar(0,wifiSignal);`. Este símbolo simula una barra de señal para indicar que la conexión Wi-Fi está activa.
- `} else {` caso contrario
- `lcd.setCursor(7, 0);` Mueve el cursor en la posición de la columna 7 y la fila 0 del LCD
- `lcd.write(byte(2));` escribe el carácter personalizado almacenado en el índice 2, que corresponde a un símbolo en forma de "X", este símbolo indica que no hay conexión Wi-Fi activa.
- `wifi_connected = false;` Cambia la variable `wifi_connected` a `false`, lo que indica que el dispositivo ya no está conectado a Wi-Fi.

## MANEJO DE LECTURAS PERIÓDICAS DE SENSORES

```
unsigned long now = millis();

if (now - last_read_time >= sampling_time) {

    last_read_time = now;

    uint8_t result;

    uint16_t sensor;

    float reading;

    String main_topic = "SENSOR/SOIL/";

    String topic;

    String msg;
```

- `unsigned long now = millis();` esta función devuelve el número de milisegundos que ha transcurrido desde que el dispositivo ESP32 comenzó a ejecutar el programa. Guarda el valor actual del temporizador en la variable `now`.
- `if (now - last_read_time >= sampling_time) {` Verifica si ha pasado el tiempo suficiente para realizar un nuevo muestreo.
- `last_read_time = now;` Una vez que el tiempo de muestreo ha llegado, se actualiza la variable `last_read_time` con el valor actual de `millis()`.
- `uint8_t result;` declara una variable `result` de tipo `uint8_t` (entero sin signo de 8 bits). Esta variable se utilizará para almacenar el resultado de la lectura de los registros del sensor.
- `uint16_t sensor;` declara una variable `sensor` de tipo `uint16_t` (entero sin signo de 16 bits). Esta variable almacenará el valor leído del sensor.
- `float reading;` declara la variable `reading` de tipo `float`, que almacenará el valor convertido del sensor.
- `String main_topic = "SENSOR/SOIL/";` declara una variable `main_topic` de tipo `String` que contiene el prefijo para los temas MQTT, en este caso

“SENSOR/SOIL/” es el tema base para los mensajes de datos del sensor relacionados con el suelo.

- `String topic`; declara una variable topic de tipo String
- `String msg`; declara una variable msg de tipo String que almacena el mensaje que se envía a través de MQTT, que contiene los valores del sensor.

## LECTURA DE REGISTROS MODBUS

```
for (int i = 0; i <= 4; i++) {  
  
    result = modbus.readHoldingRegisters(data_register[i], 1);  
  
    if (result == modbus.ku8MBSuccess) {  
  
        sensor = modbus.getResponseBuffer(0x00);
```

- `for (int i = 0; i <= 4; i++) {` lee secuencialmente los valores de diferentes registros del sensor, procesa los datos leídos y publica los resultados a través de MQTT para cada tipo de dato del sensor. `int i = 0` declara e inicializa la variable “i” con un valor inicial de 0. Esta variable se usa como contador para controlar las iteraciones del bucle. `i <=4` evalúa si la condición es verdadera antes de ejecutar cada iteración, en este caso el bucle continuará mientras i sea menos o igual a 4. `i++` Incrementa el valor de i en 1 al final de cada iteración.
- `result = modbus.readHoldingRegisters(data_register[i], 1);` se hace una lectura de registros Modbus. La función `readHolding Registers()` de la biblioteca `ModbusMaster` envía una solicitud para leer un registro específico del dispositivo Modbus. `Data_register[i]` es el registro que se está leyendo, según el valor de “i”, en este caso el arreglo `data_register` que tiene los registros `0x0000`, `0x0001`, `0x0002`, `0x0003`, `0x0004`. El segundo parámetro “1” indica que se leerá un solo registro(16bits)

- `if (result == modbus.ku8MBSuccess) {` verifica si la lectura fue exitosa. `modbus.ku8MBSuccess` es una constante que indica que la operación Modbus se completó correctamente.
- `sensor = modbus.getResponseBuffer(0x00);` obtiene el valor del primer registro, este valor se almacena en la variable `sensor`

## PROCESAMIENTO DE LECTURAS DE SENSORES MODBUS

```
switch (data_register[i]) {
    case 0x0000:
        {
            lcd.setCursor(10, 0);
            reading = sensor * 0.1;
            lcd.print(reading);
        }
}
```

- `switch (data_register[i]) {` el switch evalúa el valor de `data_register[i]`. Dependiendo de este valor, se ejecutará uno de los bloques de código dentro de los case.
- `case 0x0000:` ejecuta solo si `data:register[i]` es igual a `0x0000`, lo que en este caso indica que estás leyendo el registro que corresponde a la humedad del sensor.
- `lcd.setCursor(10, 0);` establece el cursor en la pantalla para escribir en la columna 10 y en la fila 0
- `reading = sensor * 0.1;` el valor obtenido del registro sensor se multiplica por 0.1
- `lcd.print(reading);` imprime el valor calculado `reading` en la pantalla LCD.

## MENSAJE MQTT PARA LA HUMEDAD

```
topic = main_topic + "HUMEDAD";  
  
msg = "HUMEDAD: " + String(reading, 2);  
  
break;
```

- `topic = main_topic + "HUMEDAD";` en esta parte `main_topic` es una variable que contiene la raíz del topic MQTT que va a usar. Los topics en MQTT son canales o rutas que se usan para organizar mensajes.
- `msg = "HUMEDAD: " + String(reading, 2);` “reading” es la variable que contiene el valor de la lectura de humedad obtenida del sensor. “String(reading, 2)” contiene el valor numérico de reading en una cadena de texto con solo 2 decimales.
- `break;` se usa para terminar el bloque del switch case donde este código está ejecutándose.

Los siguientes valores como case 0x0001, 0x0002, 0x0003 y 0x0004, que corresponden a la temperatura, conductividad, salinidad y TDS respectivamente. Se usan líneas de programación muy similares.

## ENVÍOS DE DATOS MQTT SI ESTÁ CONECTADO AL BROKER

```
if (client.connected()) {  
  
    client.publish(topic.c_str(), msg.c_str());
```

- `if (client.connected())` verifica si el cliente MQTT está conectado al broker.
- `client.publish(topic.c_str(), msg.c_str());` cuando el usuario está conectado, publica el mensaje en el topic especificado.

## MENSAJE DE ERRORES AL LEER SENSORES MODBUS

```
else {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Error:");  
    lcd.setCursor(0, 1);  
    lcd.print(result, HEX);  
    delay(2000);  
    lcd.clear();
```

- `else {` a partir de aquí, este bloque se ejecutará solo cuando la condición anterior no se cumple
- `lcd.clear();` limpia la pantalla LCD
- `lcd.setCursor(0, 0);` esta función establece el cursor en la columna 0 y la fila 0 de la pantalla LCD
- `lcd.print("Error:");` muestra el texto "Error:" en la pantalla LCD
- `lcd.setCursor(0, 1);` coloca el cursor en la columna 0 y en la fila 0 de la pantalla LCD
- `lcd.print(result, HEX);` imprime el valor del código de error "result" en la segunda línea del LCD y lo convierte en hexadecimal.
- `delay(2000);` esta función introduce una espera de 2 segundos
- `lcd.clear();` limpia la pantalla LCD

## INDICADOR DE CONEXIÓN MQTT Y REINTENTOS

```
if (client.connected()) {  
    lcd.setCursor(7, 1);  
    lcd.write(byte(1));  
} else {  
    lcd.setCursor(7, 1);  
    lcd.write(byte(2));  
    reconnect();
```

- `if (client.connected())` { verifica si el cliente MQTT está conectado
- `lcd.setCursor(7, 1);` coloca el cursor en la columna 7 y la fila 1 del LCD.
- `lcd.write(byte(1));` envía un byte a la pantalla LCD
- `} else {` esta cláusula se ejecutará si la condición `if (client.connected())` es falsa.
- `lcd.setCursor(7, 1);` coloca el cursor en la columna 7 y en la fila 1
- `lcd.write(byte(2));` muestra el byte 2
- `reconnect();` si el cliente no está conectado, la función `reconnect()` será llamado, lo que intentará reconectar el cliente MQTT al servidor.

## H. Diseño de la carcasa.

Se ha diseñado la carcasa de almacenamiento del trabajo para la parte del circuito de lectura en el programa SolidWorks. Cuenta con un diseño rectangular y orificios de ventilación con espacio para un filtro de polvo debido al lugar de destino. Los planos a detalle se encuentran en el anexo 8

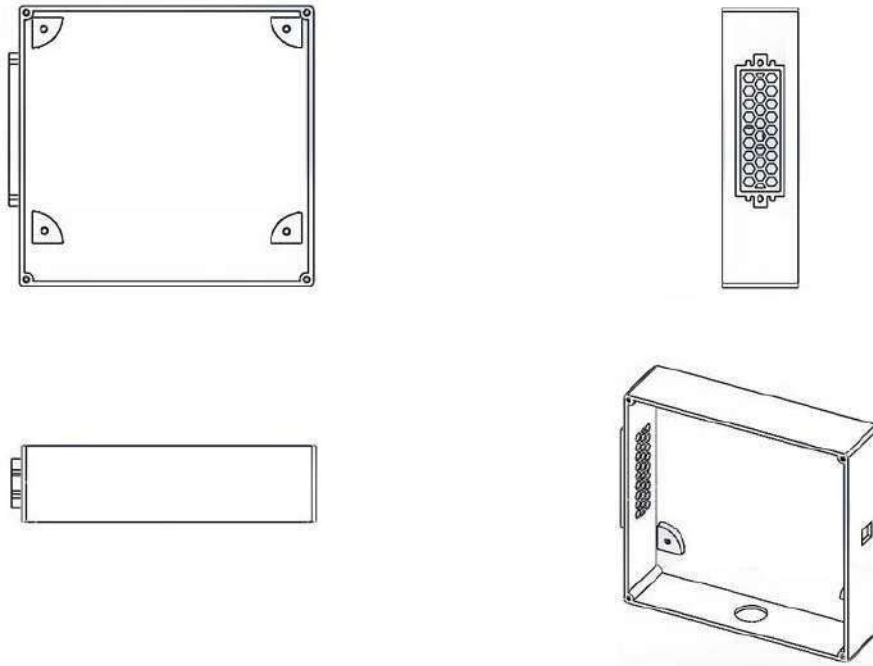


Ilustración 13: Diseño de la base de la carcasa del circuito de lectura

Fuente: Propia

En la ilustración 13 encontramos la base de la carcasa con unas dimensiones de 145 x 138 x 37.6 mm, diseñada para albergar el circuito de lectura del sensor THC-S.

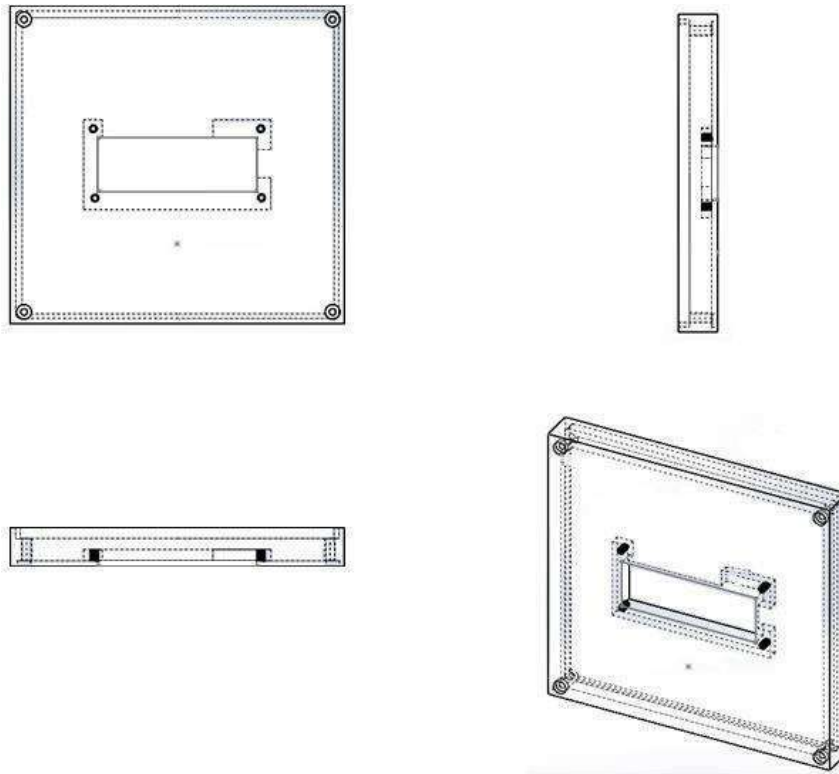


Ilustración 14: Diseño de la tapa de la carcasa del circuito de lectura

Fuente: Propia

En la ilustración 14 tenemos la tapa de la carcasa con dimensiones de 150.20 x 143.20 x 17.5, diseñada para instalar el lcd donde se van a visualizar los datos. Los planos a detalle se encuentran en el anexo 9.

## I. Diagrama de bloques y programación en NODE-RED para la Raspberry Pi 5.

En esta sección configuramos la Raspberry Pi 5 para el funcionamiento de la NODE-RED y la instalación del programa Python encargado del gestionar el envío de los datos al Ubidots donde se almacenarán y posteriormente mostrarán los datos en un dashboard.

### a. Instalación de Node-RED.

Para poder instalar Node-RED en una raspberry pi se requiere un prerequisite, usar el Raspberry Pi OS como sistema operativo.

Con la Raspberry Pi 5 preparada con su respectivo sistema operativo, se procede a:

- Actualizar la lista de los paquetes disponibles en el sistema con el comando: `sudo apt-get update`
- E instalar las versiones más recientes de los paquetes disponibles con el comando: `sudo apt-get upgrade`. Todo desde el símbolo del sistema (CMD) del Raspberry Pi.

Luego usamos el script proporcionado por la página principal de Node-RED para la instalación del Node.js, npm y Node-RED en la raspberry Pi. Además, el script también se puede utilizar para actualizar una instalación existente cuando hay una nueva versión disponible. (Node-RED, s.f.)

Al ejecutar el siguiente comando desde el símbolo del sistema (CMD) del Raspberry Pi se descargará y ejecutará el script.:

```
Script: bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Ahora con el Node-RED instalado configuramos su inicio automático al encender el Raspberry Pi.

- Para habilitar el servicio de inicio automático, se ejecuta el comando:  
`sudo systemctl enable nodered.service`
- Para deshabilitar el servicio de inicio automático, se ejecuta el comando:  
`sudo systemctl disable nodered.service`

#### **b. Diagrama de bloque funcionales o nodos.**

Luego de haber realizado la instalación necesaria, procedemos a definir las paletas y nodos que usamos y su respectiva configuración.

La paleta usada para este trabajo fue la paleta por defecto instalada llamada "node-red" versión 3.1.5 el cual cuenta con 50 nodos de trabajo.

Los nodos usados son los siguientes:

- En la sección de “red” (listado de nodos en el lado izquierdo) encontramos al nodo “mqtt in” encargada de conectar a un gestor MQTT y suscribirse a mensajes del tema especificado. El servidor se configura haciendo un clic en el icono del lápiz. Varios nodos MQTT (dentro o fuera) pueden compartir la misma conexión de gestor si es necesario.

La configuración del nodo MQTT y el bróker se muestra en las siguientes imágenes:

**Editar nodo mqtt in**

Eliminar Cancelar Hecho

**Propiedades**

Servidor SOIL SENSOR

Acción Suscríbete a un solo tema

Tema SENSOR/SOIL/CONDUCTIVIDAD

CdS 2

Salida un Texto

Nombre Nombre

Ilustración 15: Configuración de las propiedades del nodo MQTT

Fuente: Propia

Editar nodo mqtt in > **Editar nodo mqtt-broker**

Eliminar Cancelar **Actualizar**

**Propiedades** ⚙️ 📄

📁 Nombre

**Conexión** Seguridad Mensajes

🌐 Servidor  Puerto

Conectar automáticamente

Utilizar TLS

⚙️ Protocolo

📁 ID Cliente

💓 Mantener activo

📄 Sesión  Usar sesión limpia

Ilustración 16: Configuración del nodo MQTT-BROKER

Fuente: Propia

- En la sección “función” encontramos el nodo llamado function. Es una función de JavaScript que se ejecuta en los mensajes que recibe el nodo. Los mensajes se pasan como un objeto JavaScript llamado “msg”. Por convención, tiene la propiedad “msg.payload” que contiene el cuerpo del mensaje. La función devuelve un objeto de mensaje (o varios objetos de mensaje). Este nodo fue usado para convertir los datos entrantes por el nodo MQTT y convertirlos en variables de flujo para su mejor manejo. La programación del nodo función se muestra a continuación:



Ilustración 17: Programación del nodo function

Fuente: Propia

- Ahora en la misma sección del nodo function, encontraremos al nodo "exec". Es un nodo capaz de ejecutar un comando del sistema y devolvemos una salida. Este nodo se puede configurar para esperar hasta que se complete el comando o para enviar su salida a medida que el comando lo va generando. El comando que se ejecuta puede configurarse en el nodo o proporcionarse mediante el mensaje recibido.

A continuación, la configuración del nodo exec:

**Editar nodo exec**

Eliminar Cancelar Hecho

**Propiedades**

Comando

Adjuntar  msg. payload

Salida

Tiempo Espera  segundos

Esconder terminal

Nombre

Ilustración 18: Configuración del nodo exec

Fuente: Propia

- De la explicación anterior sobre los nodos y configuraciones utilizadas, se procede a crear el diagrama final de conexión con los datos recibidos de humedad, temperatura, conductividad eléctrica, salinidad y TDS desde el ESP32, medidas necesarias para conocer la salinidad de un terreno agrícola.

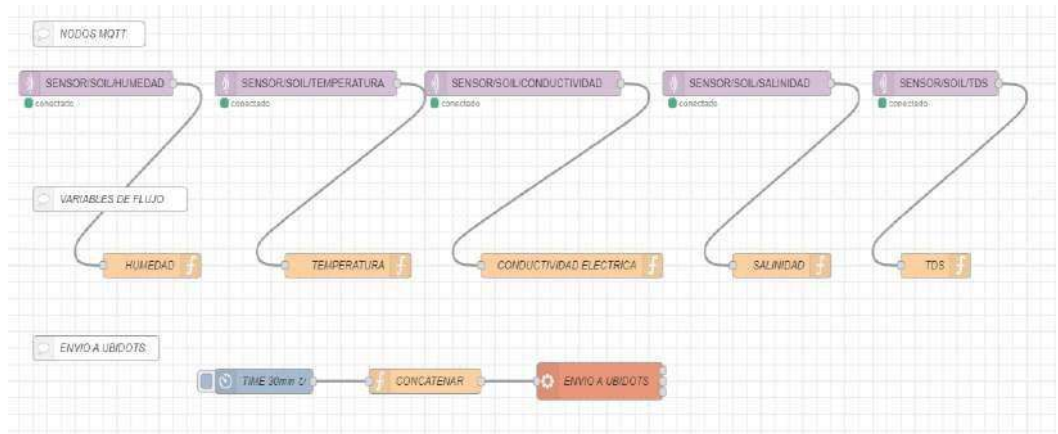


Ilustración 19: Diagrama de flujo de datos en Node-RED

Fuente: Propia

### c. Programación en Python usando el nodo exec para la comunicación con Ubidots.

El código en lenguaje Python fue creado para conectarse a la API de Ubidots y enviar los datos del sensor: Humedad, Temperatura, Conductividad eléctrica, Salinidad y TDS, desde Node-RED en la Raspberry Pi 5 al dispositivo “raspberrypi5-soilsensor-jl” creado en la plataforma Ubidots.

Python y Raspberry Pi funcionan muy bien juntos, y viene preinstalado en ella. La versión de Python que se usó es 3.11.2.

Se creó una carpeta en la Raspberry Pi 5 llamada “SistemaIoT” donde almacenamos el archivo de Python a ejecutar. El archivo con extensión Python se llama “envíos-ubidots.py”. A continuación, se detalla la programación.

```
import sys
```

```
import requests
```

```
try:
```

```
    print("Argumentos recibidos:", sys.argv)
```

```
    sensor_humedad = float(sys.argv[2])
```

```
    sensor_temperatura = float(sys.argv[3])
```

```

        sensor_conductividad = float(sys.argv[4])

        sensor_salinidad = float(sys.argv[5])

        sensor_tds = float(sys.argv[6])

except IndexError:

    print("Error: No se obtuvo el argumento")

    sys.exit(1)

except ValueError:

    print("Error: No se ha recibido un valor valido")

    sys.exit(1)

token_ubidots = "BBUS-7g9LnHpaf3UvYQ87g1Dsptj2Qr6U4"

device = "raspberrypi5-soilsensor-ji"

variable1 = "HUMEDAD"

variable2 = "TEMPERATURA"

variable3 = "CONDUCTIVIDAD ELECTRICA"

variable4 = "SALINIDAD"

variable5 = "TDS"

URL="https://industrial.api.ubidots.com/api/v1.6/devices/{}/".format(device)

sensor = {

    variable1: sensor_humedad,

    variable2: sensor_temperatura,

    variable3: sensor_conductividad,

    variable4: sensor_salinidad,

    variable5: sensor_tds

}

```

```

headers = {
    "X-Auth-Token": token_ubidots,
    "Content-Type": "application/json"
}

try:
    response = requests.post(URL, json=sensor, headers=headers)
    response.raise_for_status()
    print("Datos enviados correctamente: {}".format(response.json()))
except requests.exceptions.RequestException as e:
    print(f"Error al enviar datos a Ubidots: {e}")
    sys.exit(1)

```

Procedemos a explicar el código.

`import sys`

`import requests`

- `import sys`: importa el módulo `sys`, que permite acceder a argumentos pasados desde la línea de comandos y otras funciones del sistema.
- `import requests`: importa el módulo `request`, una biblioteca de Python para hacer solicitudes HTTP, necesaria para enviar datos a la API de Ubidots.

`try:`

```

print("Argumentos recibidos:", sys.argv)

sensor_humedad = float(sys.argv[2])

sensor_temperatura = float(sys.argv[3])

sensor_conductividad = float(sys.argv[4])

sensor_salinidad = float(sys.argv[5])

```

```
sensor_tds = float(sys.argv[6])
```

- `try`: inicia un bloque de código para capturar y manejar errores. Ejecutamos `try` para intentar obtener y convertir los argumentos de la línea de comando.
- `sensor_humedad = float(sys.argv[2])`: convierte el segundo argumento de la línea de comandos `sys.argv[2]` en un número decimal (`float`). Este argumento es el dato de humedad del sensor.
- `sensor_temperatura = float(sys.argv[3])`: convierte el tercer argumento `sys.argv[3]` en un número decimal, correspondiente a la temperatura.
- `sensor_conductividad = float(sys.argv[4])`: convierte el cuarto argumento `sys.argv[4]` en un número decimal, correspondiente a la corriente eléctrica.
- `sensor_salinidad = float(sys.argv[5])`: convierte el quinto argumento `sys.argv[5]` en un número decimal, que representa la salinidad.
- `sensor_tds = float(sys.argv[6])`: convierte el quinto argumento `sys.argv[6]` en un número decimal, que representa los TDS (Total de Sólidos Disueltos).

`except IndexError`:

```
print("Error: No se obtuvo el argumento")  
  
sys.exit(1)
```

- `except IndexError`: captura un error si no se pasan suficientes argumentos desde la línea de comandos.
- `print("Error: No se obtuvo el argumento")`: Muestra un mensaje de error indicando que faltan argumentos
- `sys.exit(1)`: finaliza el programa con un código de error 1, indicando que algo salió mal.

`except ValueError`:

```
print("Error: Valor recibido invalido")
```

```
sys.exit(1)
```

- except ValueError: captura errores si los argumentos pasados no pueden convertirse a float, por ejemplo, si hay un texto en lugar de un número.
- print("Error: No se ha recibido un valor valido"): imprime un mensaje indicando que hubo un error de valor.
- sys.exit(1): finaliza el programa con un código de error 1.

```
token_ubidots = "BBUS-7g9LnHpaf3UvYQ87g1Dsptj2Qr6U4"
```

```
device = "raspberrypi5-soilsensor-j1"
```

```
variable1 = "HUMEDAD"
```

```
variable2 = "TEMPERATURA"
```

```
variable3 = "CONDUCTIVIDAD ELECTRICA"
```

```
variable4 = "SALINIDAD"
```

```
variable5 = "TDS"
```

```
URL="https://industrial.api.ubidots.com/api/v1.6/devices/{}".format(device)
```

- token\_ubidots = "BBUS-7g9LnHpaf3UvYQ87g1Dsptj2Qr6U4" define el token de autenticación para acceder a la API de Ubidots.
- device = "raspberrypi5-soilsensor-j1" define el nombre del dispositivo creado en Ubidots al que se enviarán los datos:
  - variable1 = "HUMEDAD"
  - variable2 = "TEMPERATURA"
  - variable3 = "CONDUCTIVIDAD ELECTRICA"
  - variable4 = "SALINIDAD"
  - variable5 = "TDS"
- URL="https://industrial.api.ubidots.com/api/v1.6/devices/{}".format(device) Crea la URL de destino en Ubidots usando el nombre del dispositivo.

```

sensor = {
    variable1: sensor_humedad,
    variable2: sensor_temperatura,
    variable3: sensor_conductividad,
    variable4: sensor_salinidad,
    variable5: sensor_tds
}

```

- sensor= {variable1: sensor\_humedad, variable2: sensor\_temperatura, variable3: sensor\_conductividad, variable4: sensor\_salinidad, variable5: sensor\_tds}: crea un diccionario llamado sensor donde cada clave es una variable, por ejemplo "humedad" y cada valor es el dato correspondiente, por ejemplo, sensor\_humedad. Este diccionario enviará a Ubidots en formato JSON

```

headers = {
    "X-Auth-Token": token_ubidots,
    "Content-Type": "application/json"
}

```

- headers={"X-Auth-Token": token\_ubidots, "Content-Type": "application/json"}: define los encabezados de la solicitud HTTP. Incluye el token de autenticación "X-Auth-Token" y el tipo de contenido "Content-Type", que son JSON

try:

```

response = requests.post(URL, json=sensor, headers=headers)
response.raise_for_status()

print("Datos enviados correctamente: {}".format(response.json()))

```

- `response = requests.post(URL, json=sensor, headers=headers)` realiza una solicitud post a la URL de Ubidots, enviando el diccionario sensor como JSON y utilizando los encabezados definidos.
- `response.raise_for_status()` Verifica si hubo algún error en la solicitud. Si ocurre un error lanza una excepción.
- `print("Datos enviados correctamente: {}".format(response.json()))`: si la solicitud fue exitosa, imprime el mensaje de confirmación junto con la respuesta en JSON de Ubidots.

`except requests.exceptions.RequestException as e:`

```
    print(f"Error al enviar datos a Ubidots: {e}")
    sys.exit(1)
```

- `except requests.exceptions.RequestException as e:` captura cualquier error en la solicitud HTTP, como problemas de red o errores del servidor.
- `print(f"Error al enviar datos a Ubidots: {e}")` imprime un mensaje de error con detalles del error capturado.
- `sys.exit(1)` finaliza el programa con un código de error 1 en caso de fallo en la solicitud.

El código está pensado para obtener los datos de sensores de humedad, conductividad eléctrica, salinidad y TDS, como argumentos desde la línea de comandos, los convierte en formato decimal, y luego los envía a la API de Ubidots para su almacenamiento y visualización. Si hay errores en el proceso, se muestra un mensaje de error y el programa termina.

### **3.5.3. Diseñar el dashboards para mostrar el registro histórico de los datos del sensor de suelo.**

#### **A. Diseño e implementación del dashboards.**

El diseño se planteó con el objetivo de proporcionar una interfaz gráfica intuitiva la cual permita la visualización de los datos históricos de las variables dadas por el sensor de suelo modelo THC-S: Humedad, Temperatura, Conductividad eléctrica, Salinidad y TDS (Total de solidos disueltos).

En un Sistema destinado al monitoreo, el dashboards es una herramienta clave para facilitar la interpretación de los datos y apoyar a la toma de decisiones relacionadas con los cultivos agrarios.

El diseño fue esbozado considerando las características siguientes:

- Organización: agrupamos las variables relacionadas en secciones específicas para su mejor visualización.
- Claridad visual: Usamos gráficos de línea para la conductividad eléctrica y la humedad y métrico para la salinidad y TDS. Un termómetro para la temperatura
- Accesible: diseñado en una plataforma web para estar disponible desde cualquier dispositivo con acceso a internet.
- Interactividad: permitirnos seleccionar rangos de tiempo para el análisis de los datos históricos de la conductividad eléctrica y la humedad.

Para la implementación se hizo uso de la plataforma Ubidots, aprovechando su capacidad para recibir y almacenar los datos desde la Raspberry Pi 5 y mostrar visualmente con gráficos personalizados.

- Primero creamos un dispositivo en blanco llamado RASPBERRYPI5/SOILSENSOR/JL:

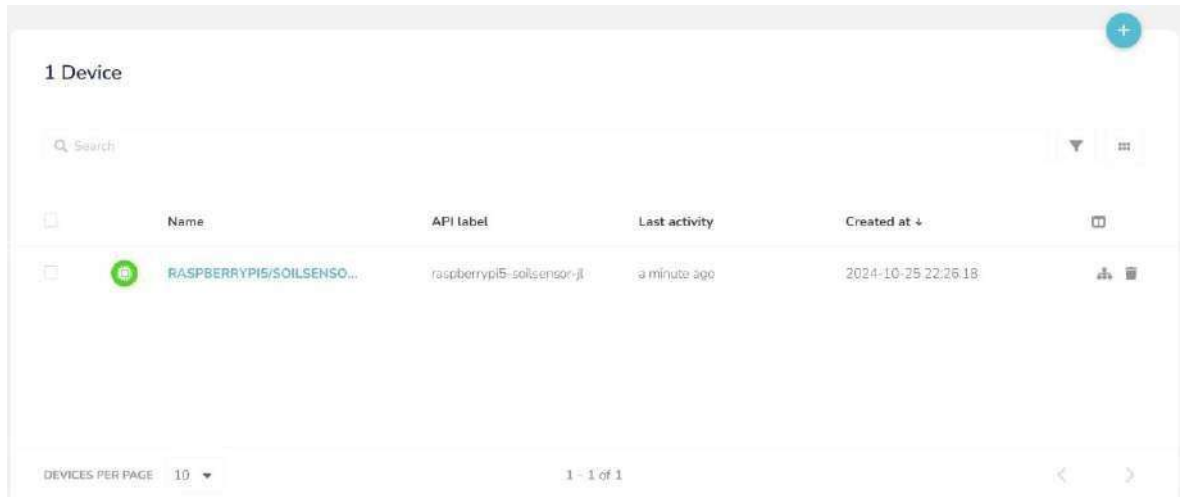


Ilustración 20: Dispositivo creado en Ubidots

Fuente: Propia

- Creamos y configuramos las 5 variables en formato “Raw variable” donde se reciben los datos del sensor desde la Raspberry Pi 5 con los mismos nombres a los programados en Python.

Valor	Nombre	Última actualización
23.7	TEMPERATURA	17 minutos ago
723	SALINIDAD	17 minutos ago
841	TDS	17 minutos ago
33.2	HUMEDAD	17 minutos ago
1.31	CONDUCTIVIDAD ...	17 minutos ago

Ilustración 21: Variables del sensor en Ubidots.

Fuente: Propia

- Creamos el dashboards siguiendo las 4 características anteriores y obteniendo el siguiente resultado:

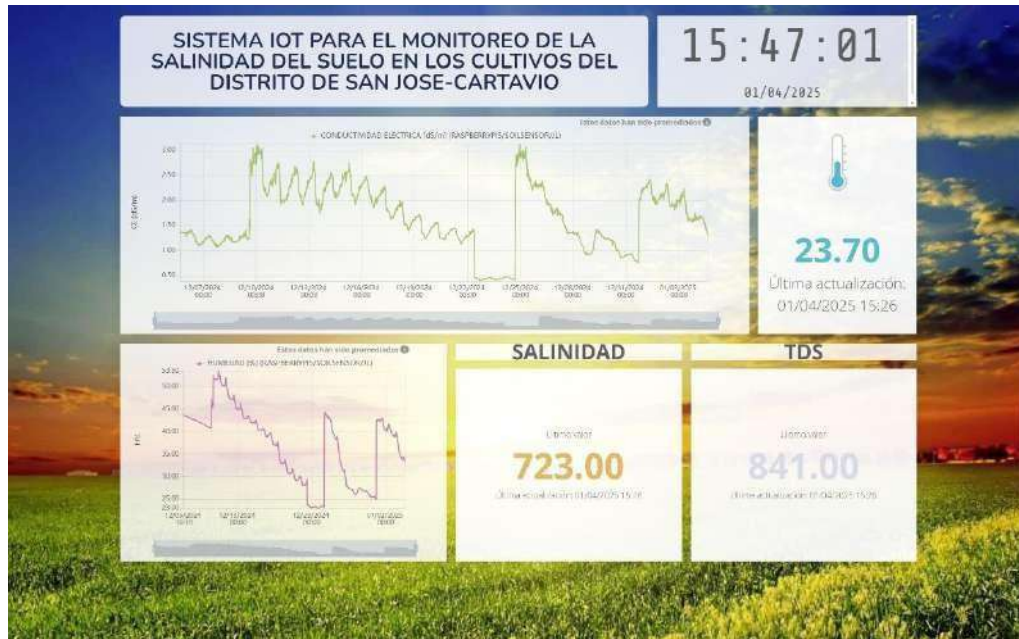


Ilustración 22: Dashboard final del sistema de monitoreo

Fuente: Propia

Se finalizó con un diseño de disposición estructurada que jerarquiza la información según su relevancia; destacando su claridad visual con el uso de gráficos un poco difuminados para mezclarse con el fondo sin perder su propósito, representar los datos de manera comprensible y precisa. Con colores estratégicos para dar una diferenciación de elementos clave, manteniendo un esquema limpio y agradable a la vista.

### 3.5.4. Comparar las especificaciones del sistema con las del mercado actual.

Tabla 13: Comparación de las especificaciones técnicas del sistema con las del mercado actual.

	Propio	Elements by Entel Ocean	Agricolus	AGQ Labs (MIR)	INIA by SINIA
Parámetros monitoreados con respecto al suelo agrícola.	<ul style="list-style-type: none"> <li>- Humedad</li> <li>- Conductividad eléctrica</li> <li>- Salinidad</li> <li>- TDS</li> </ul>	<ul style="list-style-type: none"> <li>- Contenido volumétrico de agua.</li> <li>- Potencial mátrico.</li> <li>- Contenido de sales.</li> <li>- Temperatura</li> </ul>	<ul style="list-style-type: none"> <li>- Humedad</li> </ul>	<ul style="list-style-type: none"> <li>- Humedad</li> <li>- Temperatura</li> </ul>	<ul style="list-style-type: none"> <li>- Humedad</li> <li>- Riego</li> <li>- Salinidad</li> <li>- Temperatura</li> <li>- Nutrientes</li> <li>- Acidez</li> <li>- Alcalinidad</li> </ul>
Precisión	<ul style="list-style-type: none"> <li>- 3% (0-50%), 5% (50-100%)</li> <li>- <math>\pm 3\%</math> (0-10 dS/m), <math>\pm 5\%</math> (10-20 dS/m)</li> </ul>	Alta	Alta	Alta	experimental
Conectividad	IoT	IoT, satelital	IoT	IoT	Sin especificar
Plataforma	Web (Ubidots)	Web independiente	Web (GIS)	Web y Móvil	Sin especificar
Fuente de alimentación	Cargador	Panel solar, entre otros.	Panel solar	Panel solar	Panel solar
Temperatura de operación	-40 °C a 80 °C	Robusto	Robusto	Robusto	Sin especificar
Nivel de protección	IP68	Robusto	Robusto	Robusto	Sin especificar
Fuente:	Propio	mundomagro.cl	agricolus.com	agqlabs.pe	inia.gob.pe

Como se puede observar en la tabla 13, nuestro sistema IoT para el monitoreo de la salinidad del suelo propuesto constituye una solución innovadora, diseñada para responder a las necesidades específicas del distrito de San José-Cartavio, donde los posibles problemas asociados a la salinidad del suelo podrían estar afectando su productividad agrícola por falta de un monitoreo a sus cultivos.

En comparación con otros sistemas disponibles en el mercado, nuestro sistema sobresale por su capacidad para no solo medir parámetros críticos como la humedad y salinidad, sino que, además obtenemos los datos de TDS y conductividad eléctrica (datos relacionados con la salinidad), lo que lo diferencia de opciones como agricolus, Elements o AGQ labs, que priorizan la humedad y temperatura del suelo.

Otro punto fuerte es la precisión del sensor utilizado, con un margen de error del  $\pm 3\%$  en la medición de conductividad eléctrica y humedad. Característica fundamental para posicionarse como un dispositivo confiable para la toma de decisiones en la agricultura. A sí mismo, nuestro sistema emplea tecnologías de conectividad IoT, lo que nos permite una transmisión en tiempo real y una accesible visualización de los datos a través de la plataforma web Ubidots, facilitándole su uso a de los agricultores de San José.

El sistema se diseñó para ser robusto y con un sensor que opera en un rango de temperatura de  $-40\text{ }^{\circ}\text{C}$  a  $80\text{ }^{\circ}\text{C}$  y cuenta con certificación de protección IP68, característica que garantiza su funcionamiento bajo condiciones adversas.

Por otro lado, el sistema queda abierto para futuras mejores y ampliaciones de sensores, como pH, etc. Pudiendo incrementar su alcance y efectividad.

En conclusión, nuestro sistema complementa y adiciona opciones a las tecnologías existentes en el mercado peruano, ofreciendo una solución precisa y accesible. Enfocado en la salinidad del suelo mediante la lectura de la conductividad eléctrica, convirtiéndolo en una herramienta esencial para promover la productividad agrícola en los cultivos, promoviendo el desarrollo de prácticas agrícolas más innovadoras y eficientes.

## IV. Presentación de Resultados

### 4.1. Propuesta de investigación

La propuesta plantea el diseño de un sistema IoT para el monitoreo de la salinidad del suelo en los cultivos del distrito de San José-Cartavio. Este sistema servirá para recopilar datos mediante el sensor de suelo CWT-Soil-THC-S y los transmitirá a un Raspberry Pi 5 mediante el protocolo MQTT, cuya información es enviada a la plataforma Ubidots que hace la función de nube dado que permite almacenar los datos recibidos y visualizarlos mediante un dashboards.

### 4.2. Análisis e interpretación de resultados

Se usa el sensor de suelo CWT-Soil-THC-S para la adquisición de los datos de conductividad eléctrica, humedad y temperatura. Este sensor permite capturar los datos de la conductividad eléctrica, humedad y temperatura y enviarlos por medio del protocolo modbusRTU, usado para transmitir datos de forma balanceada a largas distancias (1200 metros) y en entornos ruidosos., que pasa por un módulo MAX485 quien lo convierte a una señal TTL (transistor-transistor logic), señal que puede recibir el ESP32 DEVKITC V4.

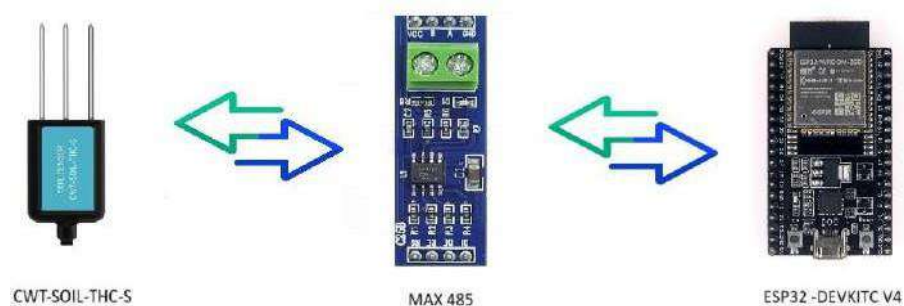


Ilustración 23: Sistema de captura de datos de la conductividad eléctrica del suelo

Fuente: Propia

Para la lectura de los registros se utilizó el programa Arduino IDE, donde leímos y enviamos por medio de MQTT al Node-RED en la Raspberry Pi 5, los registros de:

0x0000 = Humedad

0x0001 = Temperatura

0x0002 = Conductividad eléctrica

0x0003 = Salinidad

0x0004 = TDS

En la ilustración 24 se muestra el resultado del sistema de lectura y envío MQTT en físico.

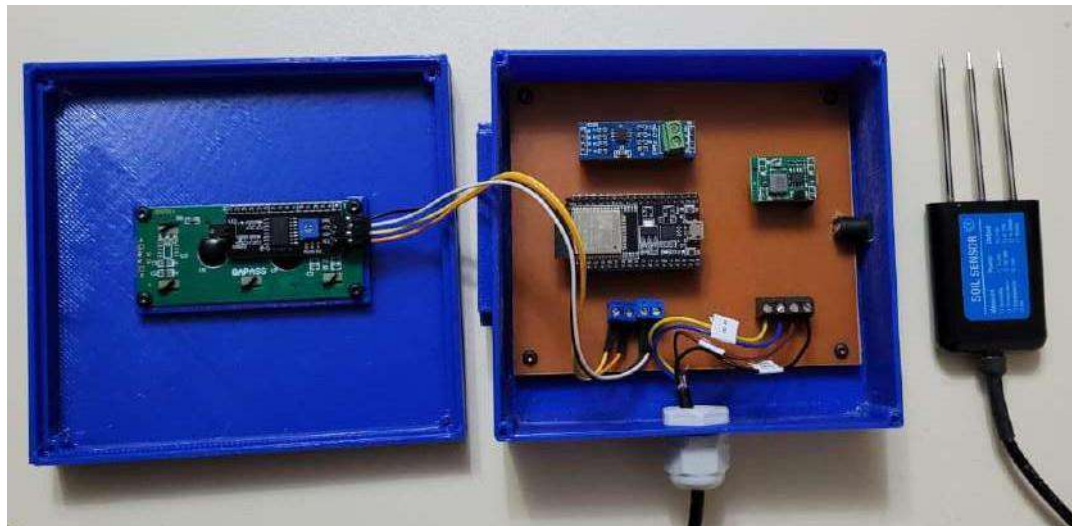


Ilustración 24: Prototipo final del circuito de lectura

Fuente: Propia

Para tener una medición más precisa, se ha calibrado nuestro sensor modelo THC-S utilizando como referencia el sensor Cond 3110 SET1 de la marca WTW propiedad de la UNIVESIDAD PRIVADA ANTENOR ORREGO.

Para obtener el valor de referencia, se ha usado un muestreo aleatorio sistemático en una parcela de terreno de cultivo del mismo nivel y se analizó 2 muestras en laboratorio obteniendo el siguiente resultado:



Ilustración 25: Prueba de laboratorio con el sensor Cond 3110

Fuente: Propia



Ilustración 26: Prueba 2 en laboratorio con el sensor Cond 3110

Los valores obtenidos en las ilustraciones 25 y 26 son de 2.28mS/cm y 2.83mS/cm, respectivamente (recordando que  $1\text{dS/m} = 1\text{mS/cm} = 1000\mu\text{S/cm}$ ). Estos valores se emplearon como referencias piloto para

determinar el factor de calibración y el offset de la conductividad eléctrica, los cuales resultaron ser 70 y 250 respectivamente. Cabe mencionar que esta calibración se realizó de manera aproximada, debido a la ausencia de un estudio detallado de las características del suelo de la zona de estudio.

Para la calibración de los factores de la salinidad y el TDS se han usado los valores teóricos que son 0.55 y 0.64 respectivamente (S. Sonon, Saha, & E. Kissel, 2022) y (Pérez Azahuanche, Pérez Rodríguez, & León Aponte, 2024)

Tenemos como resultado en las pruebas del sistema los siguientes valores máximos de conductividad eléctrica, humedad y temperatura:

Tabla 14: Registro de los valores de la lectura del sensor THC-S

Fecha	Riegos	Método	Profundidad	Máx. de Conductividad eléctrica	Máx. de Humedad	Máx. de Temperatura
29-Nov-2024	Riego 1	Campo	30cm	2.25 dS/m	45.8 %	21 °C
10-Dic-2024	Riego 2	Campo	Superficie	3.12 dS/m	53.5 %	22 °C
24-Dic-2024	Riego 3	Campo	Superficie	3.13 dS/m	44 %	22.6 °C
1-Ene-2025	Riego 4	Campo	Superficie	2.43 dS/m	43.1 %	23.7 °C
Fuente: Propia						

Ahora, el desarrollo para él envió de los datos obtenidos a Ubidots se realizó en Node-RED, quien es una herramienta de desarrollo basada en flujo de programación visual, diseñada para conectar dispositivos de hardware, API y servicios en línea como parte del internet de las cosas.

Se obtuvo el siguiente flujo:

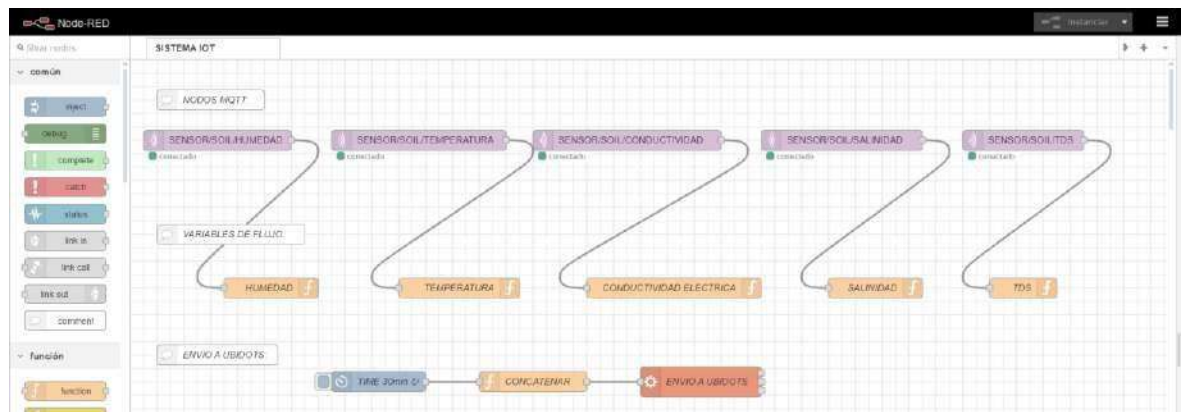


Ilustración 27: Diagrama de flujo de datos en Node-RED

Fuente: Propia

El flujo se dividió en 3 partes: los nodos MQTT encargados de recibir los datos enviados por el ESP32, las variables de flujo creadas para una mejor organización y tratado de los datos y por último el envío a Ubidots con un timer de activación de 30min y un concatenado de las variables para su envío en el nodo EXEC.

La programación del nodo EXEC denominado “envio-ubidots.py” se muestra en la ilustración 28.

```

COCO@raspberrypi: ~/SistemaloT
GNU nano 3.2          envio-ubidots.py
import sys
import requests

try:
    print("Argumentos recibidos:", sys.argv)
    sensor_humedad = float(sys.argv[2])
    sensor_temperatura = float(sys.argv[3])
    sensor_conductividad = float(sys.argv[4])
    sensor_salinidad = float(sys.argv[5])
    sensor_tds = float(sys.argv[6])
    print("Humedad: {}, Temp: {}, CE: {}, Salinidad: {}, TDS: {}".format(sensor_humedad, sensor_temperatura, sensor_conductividad, sensor_salinidad, sensor_tds))

except IndexError:
    print("Error: No se obtuvo el argumento")
    sys.exit(1)

except ValueError:
    print("Error: No se a recibido un valor válido")
    sys.exit(1)

token_ubidots = "8B0S-7g5Lni0paf3uVYQ87gi0eptzrQr0S"
device = "raspberrypi5-soilsensor-11"
variable1 = "HUMEDAD"
variable2 = "TEMPERATURA"
variables = "CONDUCTIVIDAD ELECTRICA"
variable3 = "SALINIDAD"
variable4 = "TDS"
URL = "https://industrial.api.ubidots.com/api/v1.6/devices/{}/".format(device)

sensor = (variable1: sensor_humedad, variable2: sensor_temperatura, variable3: sensor_conductividad, variable4: sensor_salinidad, variable5: sensor_tds)
headers = {"X-Auth-Token": token_ubidots, "Content_Type": "application/json"}

try:
    response = requests.post(URL, json=sensor, headers = headers)
    response.raise_for_status()
    print("Datos enviados correctamente:{}".format(response.json()))
    print(sensor)

except requests.exceptions.RequestException as e:
    print("Error al enviar datos al Ubidots: {}".format(e))
    sys.exit(1)

```

Ilustración 28: Programación en Python para el envío a Ubidots

Fuente: Propia

Se creo el archivo en una carpeta llamada SistemaloT, con el nombre de envio-ubidots.py.

Y el dashboard se ha realizado en Ubidots, una herramienta muy utilizada para visualizar y almacenar las variables en tiempo real además permitiendo modificar las variables del proceso desde cualquier parte del mundo, con acceso a internet.

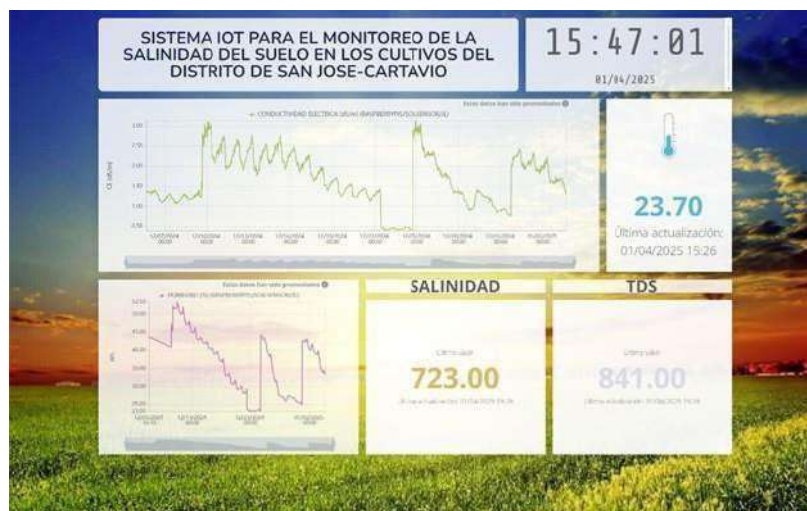


Ilustración 29: Dashboard final del sistema de monitoreo

Para la validación de resultados se ha realizado un análisis comparativo entre las mediciones obtenidas en laboratorio y los valores obtenidos en campo con el sensor THC-S. las muestras y mediciones se realizaron en un suelo franco, el cual es un tipo de suelo que tiene una mezcla equilibrada de arena, limo y arcilla. (GESMONTES, n.d.)

Se consideraron 2 profundidades para la toma de datos:

- Instalamos el sensor a la distancia recomendada de 30cm de profundidad aproximadamente (ver anexo 4 y 5) (PRISMAB, 2024) como se observa en la ilustración 30 e,
- Instalamos el sensor en la superficie del terreno, como se observa en la ilustración 31.

para después realizar la toma de datos en el terreno a muestrear para visualizar el comportamiento del sensor.

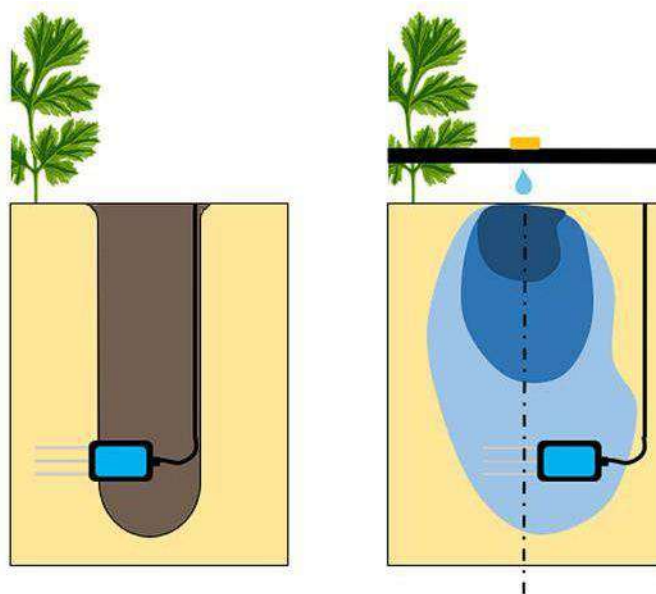


Ilustración 30: Profundidad recomendada para instalación del sensor

Fuente: <https://prismab.com/blog/como-instalar-sensores-de-humedad-del-suelo/>

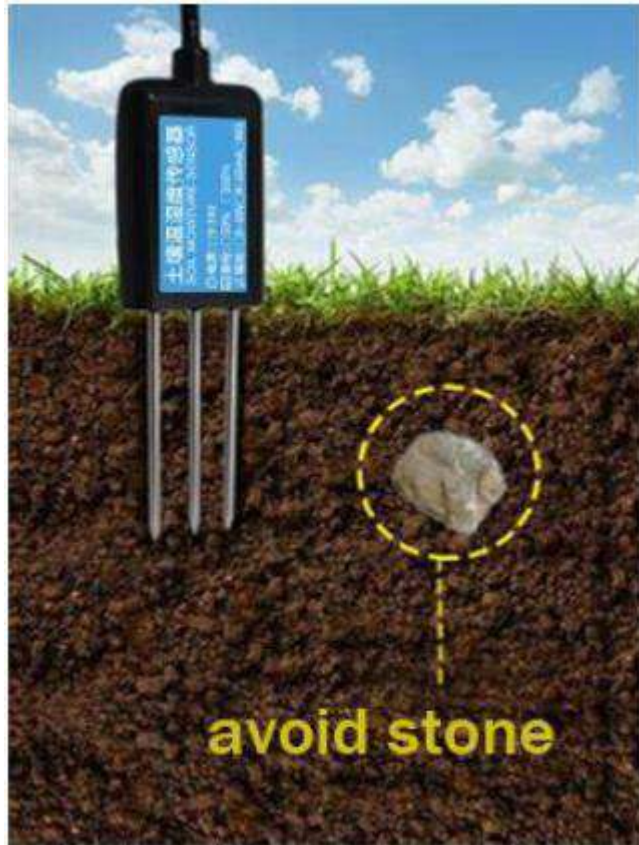


Ilustración 31: Instalación en la superficie del suelo

Fuente: Manual del sensor CWT-SOIL-THC-S

Como se ha mencionado anteriormente, para garantizar la precisión y confiabilidad de los datos obtenidos por el sensor THC-S, se ha evaluado el error absoluto en los parámetros medidos de la conductividad eléctrica. Este análisis permite determinar la desviación de las lecturas del sensor respecto al rango de valores de referencia piloto obtenidos en laboratorio (2.28 - 2.83 dS/m), validando su comportamiento y aplicación en condiciones de campo. Es importante destacar que estos valores son aproximados y no absolutos, debido a que la conductividad eléctrica del suelo está sujeta a variaciones significativas causadas por múltiples factores, como el tipo y frecuencia del riego, el uso de fertilizantes, las características intrínsecas del suelo y las condiciones climáticas predominantes.

El error porcentual se calcula con la siguiente formula:

$$\text{Error porcentual} = \left( \frac{|\text{Valor medido} - \text{Valor de referencia}|}{\text{Valor de referencia}} \right) \times 100$$

Tabla 15: Análisis de Error Porcentual

<b>Fecha</b>	<b>Riego</b>	<b>Profundidad</b>	<b>CE (THC-S)</b>	<b>CE (Cond 3110)</b>	<b>Error porcentual</b>
29-Nov-2024	Riego 1	30cm	2.25 dS/m	2.28 dS/m	1,32%
10-Dic-2024	Riego 2	Superficie	3.12 dS/m	2.83 dS/m	10,25%
24-Dic-2024	Riego 3	Superficie	3.13 dS/m	2.83 dS/m	10,60%
1-Ene-2025	Riego 4	Superficie	2.43 dS/m	2.28 dS/m	6.58%
Fuente: Propia					

De acuerdo con el error porcentual obtenido, se realiza un ajuste en el valor del offset (valor de compensación) de 250 a 130, obteniendo siguientes resultados:

Tabla 16: Valores ajustados y Error Porcentual con offset = 130

<b>Riego</b>	<b>Profundidad</b>	<b>CE (THC-S)</b>	<b>CE (Cond 3110)</b>	<b>Error porcentual</b>
Riego 1	30cm	2.13 dS/m	2.28 dS/m	6.58%

Riego 2	Superficie	3.00 dS/m	2.83 dS/m	6 %
Riego 3	Superficie	3.01 dS/m	2.83 dS/m	6.36%
Riego 4	Superficie	2.31 dS/m	2.28 dS/m	1.32%
Fuente: Propia				

Los valores ajustados demuestran que, aunque el sensor presenta un error porcentual del 6%, este margen se considera aceptable para un monitoreo general del suelo. En este contexto, el terreno se clasifica como moderadamente salino, dentro del rango de 2-4 dS/m.

No obstante, para mejorar la precisión de los datos y minimizar el margen de error en futuras implementaciones, se recomienda realizar una investigación más detallada de las características del suelo en el distrito de San José – Cartavio. Esto va a permitir una calibración más precisa del sensor, aumentando su confiabilidad y utilidad de las mediciones en condiciones reales.



Ilustración 32: Relación temporal entre la Conductividad Eléctrica y la Humedad del Suelo

Fuente: Propia

En la ilustración 32 se muestran los gráficos de monitoreo temporal de la Conductividad Eléctrica (CE) del suelo, expresada en dS/m, y la humedad relativa del suelo (H%).

En el primer gráfico se observa el comportamiento de la Conductividad Eléctrica, que varía de manera cíclica con picos y caídas relacionados con la humedad del suelo. Estas fluctuaciones reflejan la dinámica de las sales disueltas en el agua del suelo, las cuales incrementan la conductividad eléctrica cuando el suelo está con mayor humedad y disminuye cuando el agua es absorbida por las plantas o se evapora al medio ambiente.

En el segundo gráfico, se observa la humedad relativa del suelo, que presenta variaciones igualmente cíclicas que corresponden al riego y la capacidad del suelo de retener agua. Ambos gráficos presentan una correlación directa, a mayor humedad los valores de la Conductividad Eléctrica se incrementan debido a mayor cantidad de sales disueltas en ella.

Por lo tanto, ambos gráficos proporcionan información clave sobre las relaciones entre la salinidad y el contenido de agua de un terreno. El sistema de monitoreo no solo permite identificar patrones asociados con los eventos de riego, la evaporación y la absorción de agua por los sembradíos, sino también sienta las bases para implementar estrategias de manejo eficiente del suelo agrícola. Como determinar el estado de salinidad del suelo para seleccionar las plantaciones óptimas.

#### **4.3. Docimasia de hipótesis**

Dado que no se cuenta con datos previos o estudios específicos en el campo estudiado que permitan comprobar la hipótesis planteada, esta tesis se considera como una propuesta inicial, respaldada por el marco teórico y la revisión de literatura disponible. En este contexto, la hipótesis se considera válida como punto de partida y sirve como base para futuras investigaciones.

## V. Discusión de los Resultados

El sistema IoT desarrollado para la medición de la salinidad del suelo en los sembradíos del distrito de San José – Cartavio demostró ser una solución efectiva al problema planteado. Este sistema integra tecnologías accesibles y prácticas para monitorear parámetros como la humedad, temperatura, conductividad eléctrica y, mediante cálculos derivados, la salinidad y los sólidos disueltos totales (TDS).

- Después de una identificación con diferentes sistemas de medición de conductividad eléctrica del suelo existentes en Trujillo, se identificó que, aunque todos estos dispositivos cumplen con su propósito, presentan limitaciones en integración IoT, almacenamiento de datos en la nube y accesibilidad de datos en tiempo real, lo cual evidencia una brecha tecnológica para aplicaciones avanzadas de monitoreo en tiempo real.
- El prototipo diseñado cumplió con los requisitos técnicos y funcionales, integrando el sensor THC-S, el ESP32 y la Raspberry Pi 5 para recolectar y transmitir los datos de la conductividad eléctrica del suelo junto con los valores de humedad y temperatura. El uso de los protocolos de comunicación Modbus RTU y MQTT aseguraron una comunicación eficiente, mientras que el convertidor DC-DC garantizó una alimentación estable y eficiente. El código en Python implementado en la Raspberry Pi permitió enviar los datos a Ubidots de manera confiable, validando y gestionando posibles errores. El sistema demostró precisión en las mediciones y robustez en su operación, siendo modular y escalable para futuras ampliaciones. Su carcasa robusta y diseño energético eficiente lo hacen viable para aplicaciones agrícolas a largo plazo.
- El dashboard desarrollado cumplió su función de visualizar los datos de humedad, temperatura, conductividad eléctrica, salinidad y TDS de forma clara e interactiva mediante la plataforma de servicios Ubidots. Se organizó de forma intuitiva, facilitando el análisis de datos históricos y la toma de decisiones. La interactividad y adaptabilidad del diseño lo hacen flexible para futuras expansiones. En general, el dashboard resultó una herramienta funcional y clave para el monitoreo agrícola en San José – Cartavio.
- El sistema IoT desarrollado ofrece una solución innovadora al incorporar mediciones de TDS, conductividad eléctrica y humedad, compitiendo con

alternativas comerciales que se enfocan en parámetros más limitados. Su precisión, con un margen de error de  $\pm 3\%$ , y su diseño robusto garantizan confiabilidad y resistencia en condiciones adversas. Además, la conectividad IoT y la integración con Ubidots permiten una transmisión en tiempo real y acceso sencillo a los datos, facilitando su uso por los agricultores. El sistema también es escalable, permitiendo futuras mejoras como la incorporación de sensores de pH, lo que amplía su alcance. Esta herramienta precisa y accesible promueve prácticas agrícolas más eficientes y sostenibles, contribuyendo al desarrollo agrícola de la región.

## Conclusiones

- El análisis de los sistemas de medición de conductividad eléctrica del suelo existentes en Trujillo reveló que, aunque son funcionales y precisos, presentan limitaciones significativas, los sistemas de medición carecen de integración con plataformas IoT y soluciones de almacenamiento en la nube. Esto valida la necesidad de desarrollar un sistema personalizado que no solo mantenga la precisión de los dispositivos actuales, sino que también amplíe su funcionalidad mediante tecnologías IoT.
- El diseño del prototipo electrónico para leer la información de la conductividad eléctrica del suelo y enviarla a una base de datos en la nube cumple con los requisitos técnicos y funcionales planteados, adaptada a las necesidades del monitoreo agrícola en el distrito de San José – Cartavio. La arquitectura mixta del sistema integro el sensor de conductividad eléctrica modelo THC-S, el módulo ESP32, la raspberry Pi 5 y la plataforma de servicio Ubidots, logrando una comunicación fluida y precisa. El Esp32, configurado mediante Arduino IDE, permitió la lectura de datos a través del protocolo Modbus RTU y su transmisión inalámbrica hacia la Raspberry Pi 5 mediante el protocolo MQTT. Además, el circuito fue diseñado en Proteus e incluye el conversor DC-DC STEP – DOWN 2A MP1584EN para garantiza una alimentación estable y un funcionamiento correcto de los componentes. El prototipo ofrece alta precisión en las mediciones gracias a las características del sensor de suelo elegido y la capacidad del ESP32 para gestionar multitareas, como la lectura del sensor, la conexión Wi-Fi y el manejo de periféricos. Además, el diseño modular y escalable del sistema

permite futuras ampliaciones, como la incorporación de nuevos sensores o dispositivos Bluetooth según se requiera.

El prototipo está soportado por un diseño de carcasa robusto y adecuado para condiciones agrícolas al aire libre, el cual fue diseñado usando SolidWorks. Esto, sumado al cálculo eficiente del consumo energético, asegura la viabilidad del sistema en entornos reales con recursos limitados. La implementación del código en Python en la Raspberry Pi 5 fue un aspecto clave del sistema. El código recibió los datos del ESP32 y los envió a Ubidots mediante su API, empleando las bibliotecas sys y requests. El programa incluyó técnicas para la validación de datos, gestión de errores durante el envío, y la autenticación mediante el token de acceso, lo que aseguró la fiabilidad y continuidad del flujo de información hacia Ubidots. Esto permitió que las variables de humedad, temperatura, conductividad eléctrica, salinidad y TDS fueran almacenadas de forma ordenada y correcta y estuvieran disponibles para su visualización.

Por lo tanto, el prototipo no solo satisface las necesidades de monitoreo de la salinidad del suelo agrícola mediante la conductividad eléctrica, sino que también representa una solución tecnológica integral, escalable y confiable para los cultivos en el distrito de San José – Cartavio.

- El desarrollo del diseño del Dashboard permitió visualizar de manera clara y organizada los datos históricos del sensor de suelo modelo THC-S, incluyendo los datos de humedad, temperatura, conductividad eléctrica, salinidad y TDS. Utilizando la plataforma de servicios Ubidots, se desarrolló la interfaz accesible desde cualquier dispositivo con conexión a internet, integrando gráficos y widgets interactivos para facilitar el análisis y la toma de decisiones en el monitoreo de cultivos. El Dashboard cumple con su propósito como herramienta clave dentro del sistema IoT, optimizando la gestión de la información y contribuyendo a poder crear a una agricultura más eficiente y sostenible en el distrito de San José – Cartavio.
- El sistema diseñado muestra ventajas significativas en el mercado actual porque integra mediciones más completas (humedad, temperatura, conductividad eléctrica, salinidad, TDS) y ser altamente competitivo en términos de precisión, conectividad y robustez. Además, el sistema no solo satisface las necesidades locales de monitoreo de suelo, sino que también

ofrece una solución avanzada como la integración del Internet de las Cosas, frente a opciones comerciales actuales.

### **Recomendaciones**

- Realizar una ampliación de los parámetros monitoreados, considerando la integración de sensores adicionales para medir acidez (pH), nutrientes, con el fin de proporcionar una caracterización más completa del terreno agrícola, permitiendo una mayor adaptación del sistema a diversos cultivos y condiciones ambientales.
- Realizar un estudio más detallado, que incluya análisis de laboratorio y calibración específica del sensor en un ambiente controlado para reducir significativamente el margen de error.
- Ubidots para el plan gratuito ofrece solo 1 mes de almacenamiento de datos, si se requiere más tiempo de almacenamiento se puede contratar planes que ofrecen desde 6 meses hasta 1 año. Además, permite la descarga de los datos en formato Excel directo al correo electrónico.
- Realizar la incorporación de fuentes de energía autónomas para poder operar en ubicaciones remotas y mejorar la sostenibilidad del proyecto.
- Realizar funciones avanzadas en el Dashboard, como agregar alertas personalizadas, predicciones basadas en datos históricos y recomendaciones de riego y fertilización.

## Referencias

- Amazon. (2023). Obtenido de <https://aws.amazon.com/es/what-is/iot/#:~:text=El%20término%20IoT%2C%20o%20Internet,como%20entre%20los%20propios%20dispositivos.>
- AMAZON. (2023). Obtenido de AWS: <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes.>
- Benítez Burraco, A. (2019). *Avances Recientes en Biotecnología Vegeal e Ingeniería Genética de Plantas*. España: Reverte. Obtenido de [https://www.google.com.pe/books/edition/Avances\\_recientes\\_en\\_biotecnología\\_vege/T4fyDwAAQBAJ?hl=es-419&gbpv=0](https://www.google.com.pe/books/edition/Avances_recientes_en_biotecnología_vege/T4fyDwAAQBAJ?hl=es-419&gbpv=0)
- Bona, C. (2023). *Educación Sostenible*. España: Penguin Random House Grupo Editorial. Obtenido de [https://www.google.com.pe/books/edition/Educación\\_sostenible/OmaoEAAQBAJ?hl=es-419&gbpv=0](https://www.google.com.pe/books/edition/Educación_sostenible/OmaoEAAQBAJ?hl=es-419&gbpv=0)
- Cornejo García, J., Verónica Rosales, P., Gauna Flores, P., Esteban Rubio, C., & Campo Gutiérrez, A. (2006). *Biología 2*. México: Umbral Editorial, S.A.
- Cremona, M. V., & Enriquez, A. S. (2020). Algunas propiedades del suelo que condicionan su comportamiento: El pH y la conductividad eléctrica. *PRESENCIA*, 4. Obtenido de <http://hdl.handle.net/20.500.12123/7709>
- Elías Castells, X., & Bordas Alsina, S. (2012). *Energía, Agua, Medioambiente, Territorialidad y Sostenibilidad*. Madrid: Ediciones Díaz de Santos, S.A.
- Espressif Systems. (2023). *ESPRESSIF*. Obtenido de <https://www.espressif.com/en/products/socs/esp32>
- FAO. (2023). *ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA ALIMENTACIÓN Y LA AGRICULTURA*. Recuperado el 11 de 2023, de <https://www.fao.org/soils-portal/soil-management/manejo-de-suelos-problematicos/suelos-afectados-por-salinidad/more-information-on-salt-affected-soils/es/>

GESMONTES. (s.f.). *GESMONTES: DONDE LAS FINCAS SON RENTABLES*.

Recuperado el 2024, de <https://gesmontes.es/tipos-de-suelos/>

Guzmán Navarro, F., & Merino Córdoba, S. (2015). *Domótica: Gestión de la energía y gestión técnica de edificios*. ESPAÑA: RA-MA. Obtenido de [https://www.google.com.pe/books/edition/Domótica\\_Gestión\\_de\\_la\\_energía\\_y\\_gsti/3l2fDwAAQBAJ?hl=es-419&gbpv=1](https://www.google.com.pe/books/edition/Domótica_Gestión_de_la_energía_y_gsti/3l2fDwAAQBAJ?hl=es-419&gbpv=1)

IBM. (2023). *IBM Cloud*. Obtenido de <https://www.ibm.com/mx-es/topics/cloud-storage>

Inga Herrera, R. (2021). *Implementación de un sistema automático de riego agrícola basado en Raspberry Pi y plataforma IoT Artik Cloud*. Piura - Perú.

Interreg . (2023). *Interreg España - Portugal*. Obtenido de <https://learningdata.hubiberiaagrotech.eu/sensores-de-conductividad-electrica-del-suelo/>

L, N. (21 de 04 de 2023). *Linkedin: El Desarrollador Inmobiliario*. Obtenido de <https://www.linkedin.com/pulse/qué-es-un-dashboard-y-para-que-se-usa-nicolas-lamadrid/?originalSubdomain=es>

MQTT. (2022). *MQTT: El estándar para la mensajería de IoT*. Obtenido de <https://mqtt.org/>

NACIONES UNIDAS. (2023). *Informe de los Objetivos de Desarrollo Sostenible 2023: Edición Especial*. NACIONES UNIDAS, Secretaria General de las Naciones Unidas. EEUU: Copyright © 2023 United Nations. Obtenido de [https://unstats.un.org/sdgs/report/2023/The-Sustainable-Development-Goals-Report-2023\\_Spanish.pdf?\\_gl=1\\*1eoqglc\\*\\_ga\\*MTA4NDcxNjE3NS4xNzAwNzU5MDc0\\*\\_ga\\_TK9BQL5X7Z\\*MTcwMDc2MjY1NC4yLjEuMTcwMDc2MzAyNC4wLjAuMA..](https://unstats.un.org/sdgs/report/2023/The-Sustainable-Development-Goals-Report-2023_Spanish.pdf?_gl=1*1eoqglc*_ga*MTA4NDcxNjE3NS4xNzAwNzU5MDc0*_ga_TK9BQL5X7Z*MTcwMDc2MjY1NC4yLjEuMTcwMDc2MzAyNC4wLjAuMA..)

Node-RED. (s.f.). *Node-RED*. Recuperado el 2024, de <https://nodered.org/docs/getting-started/raspberrypi>

- Organización de las Naciones Unidas para la Agricultura y la Alimentación. (2009). *GUIA PARA LA DESCRIPCIÓN DE SUELOS* (Cuarta Edición ed.). Roma. Obtenido de <https://www.fao.org/3/a0541s/a0541s.pdf>
- Pérez Azahuanche, F., Pérez Rodríguez, C., & León Aponte, G. (2024). *Ingeniería Agrónoma: Manual de laboratorio de Química Analítica*. Trujillo.
- PRISMAB. (18 de Septiembre de 2024). *PRISMAB*. Obtenido de <https://prismab.com/blog/como-instalar-sensores-de-humedad-del-suelo/>
- Raspberry Pi*. (s.f.). Recuperado el 21 de 12 de 2023
- Raúl S. Lavado. (10 de 2021). *FAO*. Obtenido de ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA ALIMENTACIÓN Y LA AGRICULTURA: [https://www.fao.org/fileadmin/user\\_upload/GSP/GSAS21/day1/017\\_Lavado.pdf](https://www.fao.org/fileadmin/user_upload/GSP/GSAS21/day1/017_Lavado.pdf)
- S. Sonon, L., Saha, U., & E. Kissel, D. (2022). *UNIVERSITY OF GEORGIA*. Obtenido de <https://extension.uga.edu/publications/detail.html?number=C1019>
- Salazar, J. (2017). *Redes Inalámbricas* (Primera edición ed.). Praga: Erasmus+. Obtenido de <https://techpedia.eu/topic/9>
- Tíscar Oliver, P. A. (2021). *Botánica Agronómica* (Primera ed.). España: Ediciones Paraninfo S.A. Obtenido de [https://www.google.com.pe/books/edition/Botánica\\_agronómica/v6MIEAAAQBAJ?hl=es-419&gbpv=1](https://www.google.com.pe/books/edition/Botánica_agronómica/v6MIEAAAQBAJ?hl=es-419&gbpv=1)
- Tovar Soto, J., Camacho Tomayo, J., Bermeo Clavijo, L., & García Navarrete, O. (2022). Propuesta de un sistema de medición de conductividad eléctrica aparente y humedad del suelo para pequeños agricultores. *Revista U.D.C.A Actualidad & Divulgación Científica, under Creative Commons License CC BY-NC 4.0*, 10. doi:<http://doi.org/10.31910/rudca.v25.n2.2022.1592>
- UAM. (s.f.). *Universidad Autónoma de Madrid*. Recuperado el 2023, de <https://www.uam.es/Ciencias/Botánica->

hoy/1446766816513.htm?language=es#:~:text=La%20Botánica%2C%20en  
tendida%20como%20el,múltiples%20campos%20de%20actuación%20y

Ubidots. (s.f.). *Sencillo pero potente IoT industrial*. Recuperado el 21 de 12 de  
2023, de <https://es.ubidots.com>

UNIR: *La Universidad en Internet*. (17 de 05 de 2023). Obtenido de  
[https://www.unir.net/ingenieria/revista/arquitectura-  
iot/#:~:text=La%20arquitectura%20IoT%20es%20un, donde%20serán%20p  
rocesados%20y%20almacenados](https://www.unir.net/ingenieria/revista/arquitectura-<br/>iot/#:~:text=La%20arquitectura%20IoT%20es%20un, donde%20serán%20p<br/>rocesados%20y%20almacenados).

Villa Crespo, E., & Morales Alonso, I. (2022). *Ciberseguridad IoT y su aplicación  
en Ciudades Inteligentes*. Madrid: RA-MA S.A. Editorial y Publicaciones.

## ANEXO 1



Ilustración 33: Sensor Cond 3110 SET 1

## ANEXO 2



Ilustración 34: Prototipo completo del Sistema de monitoreo de la conductividad eléctrica del suelo

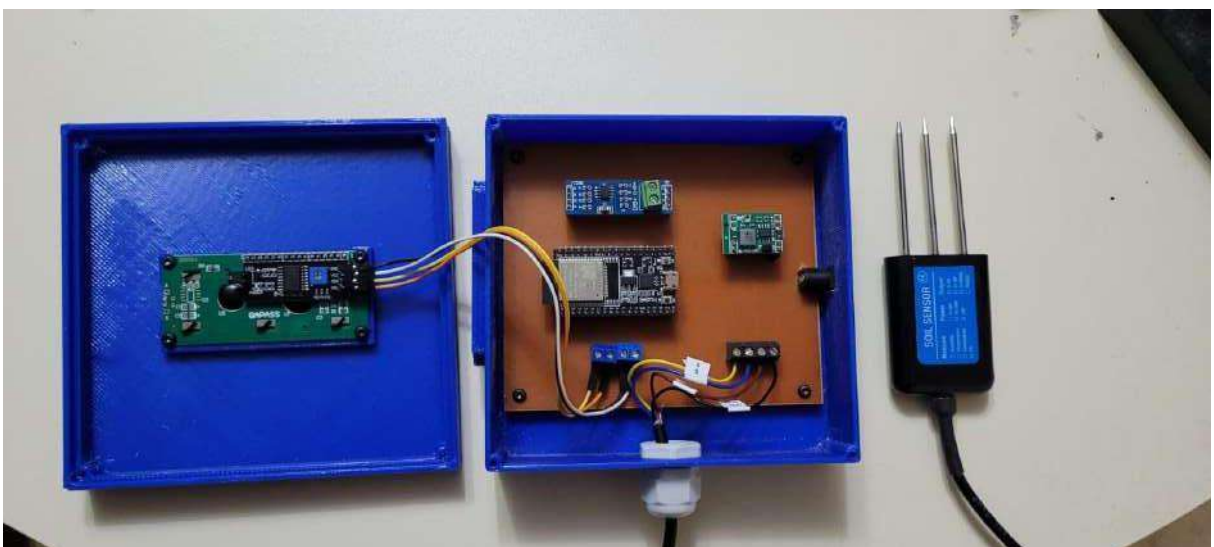


Ilustración 35: Prototipo final del circuito de lectura

### ANEXO 3



Ilustración 36: Terreno de muestreo

### ANEXO 4



Ilustración 37: Profundidad de instalación

## ANEXO 5



Ilustración 38: Instalación del sensor THC-S

## ANEXO 6



Ilustración 39: Instalación del prototipo

## ANEXO 7



Ilustración 40: Instalación del Raspberry Pi y Wi-Fi

## ANEXO 8



# ANEXO 9

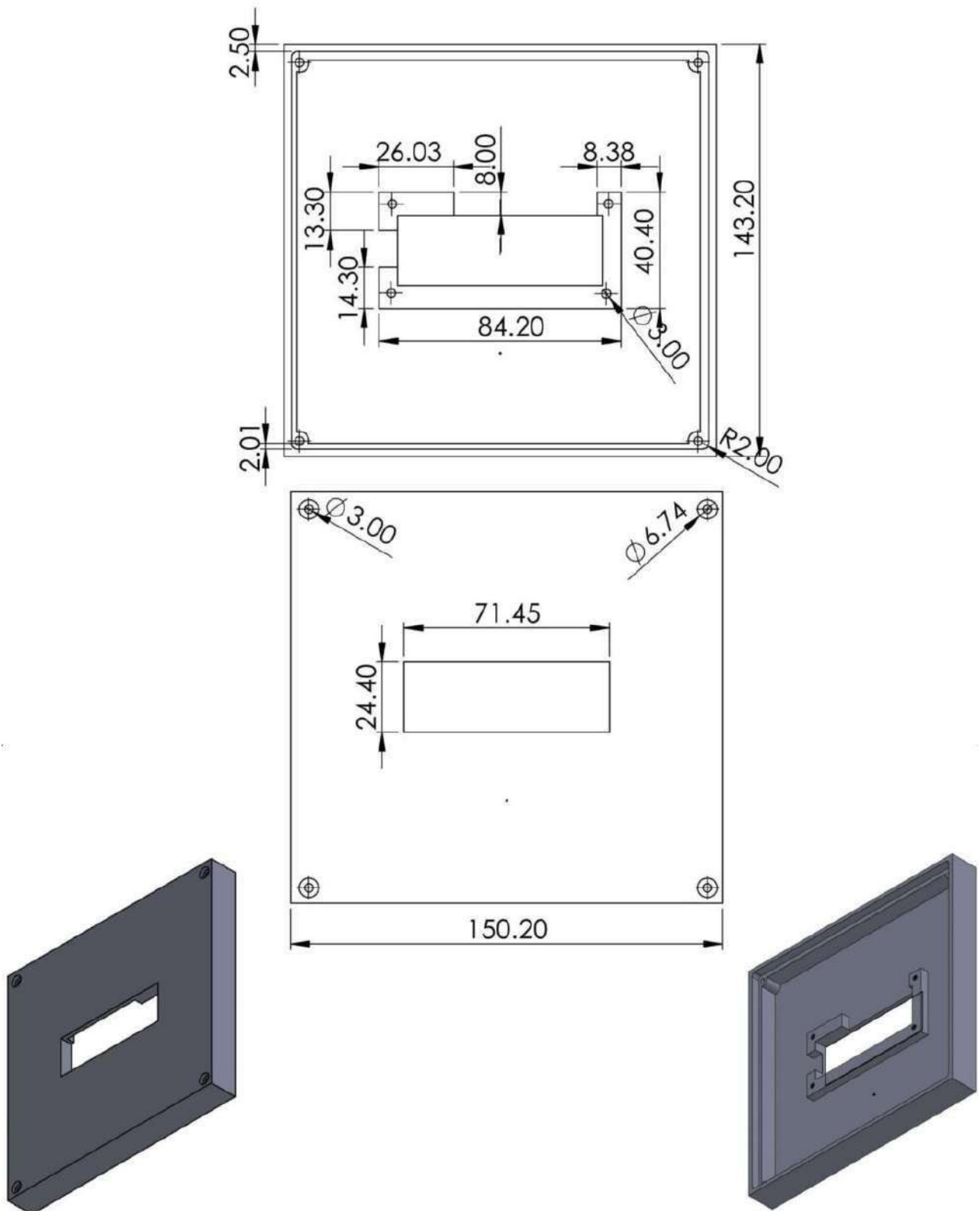


Ilustración 42: Tapa de la carcasa del circuito de lectura del sensor THC-S

Fuente: Propia