

UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA
LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS
Y DE TELECOMUNICACIONES

**Transformación agrícola: implementación de un sistema
de Monitoreo IoT para la gestión inteligente de cultivos**

T E S I S

QUE PARA OPTAR POR EL TÍTULO DE
**LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS
Y DE TELECOMUNICACIONES**

P R E S E N T A

JOSÉ ANGEL QUIROZ VALDEZ

DIRECTOR: **DR. EDUARDO RAMOS DÍAZ DRA**

CODIRECTOR: **DR. JOSÉ AGUSTÍN ALMARAZ DAMIÁN**

Ciudad de México, octubre de 2025.

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Dedicatoria

Dedico este trabajo a mi madre, quien con su amor incondicional y fortaleza ha sido mi mayor inspiración. Gracias por apoyarme día tras día, por tus consejos, por tu paciencia infinita y por ser el pilar que sostiene mis sueños. Tus sacrificios y enseñanzas han sido el motor que me ha impulsado a seguir adelante, incluso en los momentos más difíciles. No hay palabras suficientes para expresar mi gratitud por todo lo que has hecho por mí.

A mis hermanos, Berenice, Fabián, Nayeli y Ezequiel, quienes siempre han estado a mi lado, brindándome su apoyo incondicional y su confianza. Cada gesto de ánimo, cada palabra de aliento y cada momento compartido han sido fundamentales en este camino. Gracias por ayudarme a mantenerme firme cuando las cosas parecían imposibles y por ser mi refugio en los días de incertidumbre.

Gracias a mi familia, por su amor y apoyo, que han sido la base de cada logro que hoy celebro. Este trabajo es un reflejo de todo lo que hemos compartido y un testimonio de mi eterno agradecimiento.

A ustedes, dedico este esfuerzo con gratitud infinita y todo mi corazón.

Agradecimientos

Agradezco a todas las personas que contribuyeron a la realización de este trabajo y a mi crecimiento personal y profesional. A mis maestros, Gabriel Bengochea, Lucía Ivonne, Óscar Epzi, Alfredo Palacios y Agustín Alamraz, por compartir su conocimiento y experiencia conmigo. Sus enseñanzas y orientación han sido fundamentales para completar este proyecto. Cada uno de ustedes ha dejado una huella en mi formación, y por ello, les estoy eternamente agradecido.

Y a todos aquellos amigos que, sin buscarlo, hicieron de este camino una experiencia única y valiosa. Gracias por estar ahí, en los momentos más significativos, con la fuerza y presencia de una dama de hierro. . . silenciosa pero constante.

A todos ustedes, Familia, maestros y amigos GRACIAS

Resumen

El objetivo de este proyecto de investigación es desarrollar un prototipo de monitoreo por radiofrecuencia para la agricultura, utilizando IoT y tecnologías de largo alcance con bajo consumo energético. Esta iniciativa surge de la necesidad del sector agrícola de contar con un sistema que permita optimizar el uso de recursos naturales y mejorar la tecnificación de los procesos de producción.

El prototipo está compuesto por cuatro partes principales: el nodo de sensores, el nodo gateway, el servidor y el nodo actuador. El nodo de sensores integra cuatro dispositivos encargados de medir parámetros físicos y químicos del suelo, como temperatura, humedad, exposición solar y pH. Los datos obtenidos son transmitidos de forma inalámbrica mediante LoRa al gateway, que posteriormente los envía a un servidor alojado en una Raspberry Pi 4, donde se almacenan en una base de datos MySQL. Esta información se visualiza a través de una interfaz web, lo que permite su consulta en tiempo real y facilita la toma de decisiones.

Gracias a este sistema de monitoreo, los agricultores pueden recibir datos precisos sobre la humedad del suelo, la temperatura ambiente, la alcalinidad y la exposición solar. Con esta información, es posible optimizar el riego, mejorar la protección de los cultivos y planificar mejor la siembra.

En la etapa del nodo actuador, los datos procesados en el servidor son enviados al ESP32, que se comunica con el gateway para activar acciones específicas según las condiciones detectadas. Además, se ha incorporado un sistema de alertas que notifica cualquier anomalía en los sensores a través de una aplicación, permitiendo una respuesta oportuna ante posibles problemas.

La implementación de este prototipo ha demostrado su eficacia en la gestión de recursos agrícolas, asegurando condiciones óptimas para el desarrollo de las plantas y facilitando la supervisión de los factores ambientales clave.

Palabras clave: IoT, sistema, datos, LoRa®.

Abstract

This work focuses on developing a Radio Frequency monitoring prototype for agronomy using IoT and long-range, low-power technologies. This initiative arises from the need within the agricultural sector to implement systems that modernize crop management and optimize the use of natural resources throughout the growth process.

The IoT radio frequency prototype comprises four main components: the sensor node, the gateway node, the server, and the actuator node. The sensor node is designed along with three sensors that measure the environmental and chemical parameters of the soil, such as temperature, humidity, solar exposure, and pH levels. These data are transmitted wirelessly via LoRa® to the gateway, where the information is hosted on a Raspberry Pi 4 and a MySQL database acting as a server.

The collected data is visualized through a web interface, allowing for near real-time monitoring—meaning that sensor readings are transmitted and processed almost instantly, with no significant delays. Enabling near real-time transmission of data such as soil moisture, ambient temperature, soil alkalinity, or solar radiation. Aiming to help farmers make informed decisions regarding irrigation, crop protection, or planting schedules.

In the actuator node phase, data sent from the ESP32 to the gateway is received and processed. Additionally, an alert system is employed to notify users through an application if any sensor within the crop exhibits significant deterioration. The implementation of this IoT-based radio frequency prototype has improved resource management by addressing the needs of the plants and enabling continuous monitoring of the soil's physical and chemical parameters.

Keywords: harvest, IoT, data, system, monitor.

Índice general

1. Introducción	10
1.1. Definición del problema	11
1.2. Objetivo general	12
2. Agricultura Inteligente y Desarrollo de Proyectos IoT	14
2.1. Impacto del Cambio Climático en la Producción Agrícola	14
2.1.1. Agricultura	14
2.1.2. Agricultura De Precisión	15
2.2. Soluciones Comerciales	15
2.2.1. Libelium	15
2.2.2. Atlas Scientific	16
2.3. Soluciones académicas en Agricultura Inteligente	17
2.3.1. El prototipo de telecontrol para agricultura inteligente	17
2.3.2. Red de Sensores Inalámbricos bajo Protocolo LoRa® y Gestión de Pro- cesos para la Analítica de Datos mediante Meta Sistema Operativo para Monitoreo Ambiental en Invernaderos	18
2.3.3. Desarrollo de un micro servidor web con arduino	18
2.3.4. Implementación de un sistema de monitoreo de PH aplicado a un inver- nadero de la zona alta sector Salache	18
2.3.5. Estudio sobre las características del protocolo de transmisión de LoRa	19
3. Tecnologías Inalámbricas	26
3.1. ISM	27
3.2. WiFi	27
3.3. Bluetooth	27
3.4. LoRa®	28
3.4.1. Modulación y transmisión LoRa	29
3.5. Protocolo de Comunicación UART	30
3.5.1. FIFO en UART	33
3.6. Código CRC	33
4. Desarrollo del Sistema SIGASI	38
4.1. Selección de Tarjetas de Desarrollo	39
4.2. Selección, diseño y programación de sensores	40

4.2.1. Humedad	40
4.2.2. Sensor FC-28	41
4.2.3. Sensor SEN-HS-CAP	42
4.2.4. Temperatura	44
4.2.5. LM35	45
4.2.6. Solar	45
4.3. Módulo LoRa E220	52
4.3.1. Trasmision de Datos	53
4.4. ESP-32	54
4.5. Receptor y Sistema de Procesamiento	55
5. Diseño y caracterización	59
5.1. Sensores	59
5.1.1. Pruebas con sensor de humedad capacitivo	59
5.1.2. Pruebas con sensor de temperatura	62
5.1.3. Pruebas con Sensor de pH	63
5.1.4. Calibración del sensor PH-4502C con soluciones comunes	65
5.2. Librerías de comunicación	66
5.2.1. Biblioteca EBYTE22	66
5.2.2. Biblioteca SoftwareSerial	67
5.3. Transmisión y Recepción	68
5.3.1. Transmisión (TX)	68
5.3.2. Recepción (RX)	68
5.4. Pruebas de transmisión y recepción con módulo LoRa	69
5.4.1. Envío de Datos	69
5.5. Procesamiento y almacenamiento de datos en el servidor	70
5.5.1. Procesamiento de datos.	70
5.5.2. Código PHP	71
5.5.3. Código de Interfaz usuario	71
5.5.4. Recopilación de datos en PHP	72
5.5.5. Control de tiempo de espera (timeout)	73
5.5.6. Llenado de la Base de Datos	73
5.5.7. Conexión a la Base de Datos	73
5.5.8. Inserción de Datos en la Base de Datos	74
5.5.9. Cierre de la Conexión	74
5.5.10. Rango de envío	77
5.5.11. Desarrollo de Aplicación Android	79
6. Discusión / Análisis de Resultados	85
6.1. Descripción de los análisis realizados para validar el sistema	85
6.1.1. Análisis de distintos valores de voltaje	86
6.2. Evaluación de desempeño	86
7. Conclusiones del trabajo	89

8. Recomendaciones y Trabajo Futuro	91
8.1. Recomendaciones	91
8.2. Trabajo Futuro	92
Apéndices	97
A.	97
B.	109
B.1. INSTALACION SERVIDOR	109
B.2. Instalación de MySQL	110
B.3. Configuración Inicial de MySQL	110
B.3.1. Creación de un Nuevo Usuario	110
B.4. Instalación de phpMyAdmin	111
B.5. Enlace de phpMyAdmin con MySQL	111
B.6. Configuraciones Adicionales en MySQL	111
B.7. Ajustes en la Configuración de PHP	111
B.8. CODIGO PHP	112
B.9. Pseudocódigo PH4502C	113
B.10. biblioteca necesaria para el módulo EBYTE22	114
B.11. biblioteca necesaria para el módulo EBYTE	116
B.12. pseudocódigo TX	118
B.13. PseudoCódigo RX	120
B.14. Pseudocódigo para la Instalación y Configuración de un Servidor Web	123

Índice de figuras

3.1. Espectro electromagnético	26
3.2. modulación 4-FSK	30
3.3. Sistema simple LoRa [14]	30
3.4. Datagrama del protocolo <i>UART</i> [13]	31
3.5. Transmisión de la letra W. [16]	32
4.1. Diagrama a bloques del sistema SIGACI.	38
4.2. Hidrometro Fc-28	42
4.3. Sensor de Humedad Suelo Capacitivo Anticorrosivo.	43
4.4. Sensor de temperatura LM35.	45
4.5. Circuito acondicionador para la adquisición de Luminosidad	46
4.6. circuito sumador elaborado con amplificadores operacionales.	47
4.7. Escala vectorial de PH	48
4.8. Sensor de PH Liquido con electrodo E201-BNC	51
4.9. Módulo de comunicación E220 [19]	53
4.10. Unidad de Procesamiento ESP-32 [27]	55
4.11. Raspberry Pi 4B [24]	57
5.1. Base de datos en funcionamiento	76
5.2. Entorno WEB del sistema SIGACI.	76
5.3. Modelo FreeSpace	79
5.4. Diseño de pantallas.	80
5.5. Diseño registro.	81
5.6. Configuración de la primer pantalla.	81
5.7. Inicio de Sesión	83
5.8. Registro de Usuario y contraseña	83
5.9. Base de Datos Versión Aplicación	83
6.1. Modelo de radiación Longley-Rice	87

Índice de Tablas

3.1. El cuadro compara Wi-Fi, Bluetooth y LoRa: Wi-Fi ofrece gran velocidad pero alto consumo, Bluetooth es de corto alcance y bajo consumo, y LoRa destaca por su largo alcance y bajo consumo, ideal para IoT.	31
4.1. Datos Básicos de Microcontroladores	39
4.2. Desventajas de Microcontroladores	40
4.3. Valores de pH de diferentes sustancias[21]	48
5.1. Relación entre temperatura y voltaje de salida del LM35.	63
5.2. Valores de pH esperados y medidos para soluciones comunes.	65
5.3. Relación Ideal Entre Voltaje y pH	66
5.4. Comparación de datos entre el transmisor (TX) y el receptor (RX). A la izquierda se muestra el transmisor, que envía datos desde el microcontrolador ESP32, y a la derecha se observa el receptor ESP-12E, que recibe la información a través del módulo EBYTE22.	69
6.1. Datos de medición durante 12 horas.	85
6.2. Estadística descriptiva de las mediciones con valores propuestos.	86
6.3. Consumo energético estimado del sistema SIGACI.	87

Capítulo 1

Introducción

La agricultura enfrenta una serie de desafíos que pueden obstaculizar la producción de alimentos de manera eficiente y sostenible. Entre estos desafíos se encuentran las condiciones climáticas adversas, las plagas y enfermedades que afectan a los cultivos, los problemas relacionados con la calidad del suelo, la escasez de agua y la toxicidad por minerales, lo que repercute en características como el color y el sabor de los frutos.

Estos factores generan miles de pérdidas monetarias para agricultores y comerciantes. Sin embargo, gracias a la tecnología existente, es posible enfrentar estas adversidades con el objetivo de mejorar la calidad de los productos y, al mismo tiempo, maximizar la productividad a bajo costo. Estas tecnologías pueden ser implementadas en sectores donde tradicionalmente no ha llegado la innovación tecnológica. Al introducir tecnologías modernas en zonas rurales o remotas, se brinda una solución a la limitada o inexistente infraestructura tecnológica.

En muchas partes del mundo, especialmente en regiones rurales o en países en desarrollo, la falta de acceso a la tecnología representa un obstáculo significativo. La implementación de sistemas de monitoreo agrícola basados en tecnologías inalámbricas permitiría a los agricultores recibir información casi en tiempo real sobre aspectos cruciales como las condiciones climáticas, el estado de los cultivos y la calidad del suelo. Esta información les permitirá tomar decisiones más informadas y, en última instancia, mejorar su producción agrícola.

Existen numerosos sectores dentro de la agricultura que pueden beneficiarse del uso de tecnologías inalámbricas como Wi-Fi, Bluetooth, LoRa®, RFID (Identificación por Radiofrecuencia), redes celulares y satelitales, entre otras. Si a esto se le suman dispositivos de monitoreo, se puede mejorar considerablemente la eficiencia en la productividad agrícola [17].

La agricultura es uno de los pilares fundamentales de la civilización humana y, actualmente, enfrenta desafíos cruciales que van más allá de la simple producción de alimentos. Entre ellos destacan los impactos ambientales derivados de las prácticas agrícolas modernas, como la escasez y contaminación del agua, la degradación del suelo, el cambio climático y la pérdida de biodiversidad. Estos problemas hacen que la gestión eficiente de los recursos y la mitigación de los riesgos ambientales sean cada vez más prioritarias. En este contexto, la adopción de tecno-

logías disruptivas como el Internet de las Cosas (IoT) ofrece nuevas oportunidades para mejorar la eficiencia y la productividad en el campo.

La implementación de un sistema de monitoreo agrícola basado en IoT, que recopile y analice datos casi en tiempo real, puede revolucionar la forma en que se gestionan los cultivos y los insumos agrícolas. Este enfoque innovador representa un paso fundamental hacia una agricultura más sostenible y respetuosa con el medio ambiente, al permitir un cambio significativo en la eficiencia de los procesos agrícolas mediante el análisis de datos. Actualmente, la agricultura moderna se enfrenta a retos importantes debido al aumento de la demanda global de alimentos y la necesidad de adoptar prácticas sostenibles.

Entre estos desafíos, nuevamente, se encuentran la escasez y contaminación del agua, la degradación del suelo, el cambio climático y la pérdida de biodiversidad. Estos problemas requieren nuevas estrategias para gestionar los recursos agrícolas de forma eficiente y reducir el impacto ambiental. En este escenario, el IoT se presenta como una tecnología innovadora capaz de transformar la gestión agrícola.

La implementación de sistemas de monitoreo basados en IoT permite recopilar y analizar datos casi instantáneamente sobre las condiciones del cultivo, facilitando decisiones informadas y oportunas. Este estudio aborda el desarrollo de un sistema de monitoreo agrícola utilizando el microcontrolador ESP32 y una serie de sensores analógicos, con transmisión de datos a través de un módulo LoRa® (Long Range). Los datos recolectados se almacenan en una base de datos y se visualizan en una página web, proporcionando una visión continua y detallada del estado de los cultivos, incluyendo niveles de riego y posibles afectaciones.

El propósito de este sistema es optimizar el uso de los recursos, mejorar la productividad agrícola y avanzar hacia una agricultura más sostenible y respetuosa con el medio ambiente. La integración de tecnologías avanzadas y el análisis de datos en tiempo real representan un cambio fundamental en la forma en que se gestionan los cultivos, permitiendo una respuesta rápida ante las necesidades y condiciones cambiantes del entorno agrícola.

1.1. Definición del problema

Para afrontar los desafíos ambientales en la agricultura contemporánea y establecer un sistema de monitoreo agrícola basado en el IoT que nos permita determinar si un cultivo está en condiciones óptimas, es crucial comprender los distintos problemas que pueden surgir durante su desarrollo. Este análisis, nos capacita para diseñar, desarrollar e implementar un prototipo capaz de gestionar y controlar eficientemente cada una de estas problemáticas. La implementación del Sistema de Gestión Agrícola con Control Inteligente (SIGACI).

Este trabajo es relevante porque responde a la necesidad urgente de optimizar la producción agrícola en un contexto global marcado por el cambio climático, la escasez de recursos y la

creciente demanda de alimentos. La agricultura, siendo uno de los sectores más importantes para la economía y la seguridad alimentaria mundial, enfrenta desafíos cada vez mayores relacionados con la disponibilidad de agua, la calidad del suelo y las condiciones ambientales adversas. Tradicionalmente, los agricultores han tomado decisiones basadas en la observación manual de los cultivos, lo que puede resultar en decisiones tardías o inexactas que comprometen la productividad y calidad de las cosechas.

La implementación de un sistema de monitoreo agrícola basado en tecnologías de (*IoT*) y *LoRa*® ofrece una solución innovadora a estos problemas, proporcionando a los agricultores la capacidad de obtener datos casi en tiempo real sobre las condiciones del suelo, tales como humedad, temperatura, pH y exposición solar. Estos datos, accesibles a través de una plataforma web, permiten una gestión más eficiente y precisa de los recursos naturales, particularmente el agua y los fertilizantes, optimizando así el uso de estos insumos críticos.

Este sistema también permite una automatización parcial en la toma de decisiones, al alertar a los agricultores sobre condiciones críticas en los cultivos, como sequía o variaciones en el pH que podrían afectar el desarrollo de las plantas. La automatización y el monitoreo continuo no solo mejoran la eficiencia del manejo de los cultivos, sino que también reducen el riesgo de pérdidas económicas debido a condiciones climáticas adversas o a una gestión inadecuada de los recursos.

Por otro lado, este proyecto es especialmente relevante para regiones agrícolas rurales y remotas, donde la infraestructura tecnológica suele ser limitada o inexistente. La tecnología *LoRa*®, al permitir la transmisión de datos a largas distancias con un bajo consumo energético, es ideal para estas zonas, ya que no requiere de una infraestructura compleja para su implementación. Esto permite a los pequeños y medianos agricultores acceder a tecnologías avanzadas que antes solo estaban disponibles para grandes productores.

1.2. Objetivo general

Desarrollar, implementar y probar un sistema de monitoreo agrícola basado en tecnologías IoT y LoRa®, que permita gestionar los cultivos de manera inteligente mediante la supervisión de parámetros físicos y químicos del suelo, con el fin de optimizar el uso eficiente de los recursos naturales y aumentar la productividad agrícola.

Objetivos específicos

1. Diseñar e implementar un conjunto de sensores capaces de medir los parámetros clave del suelo, como la humedad, temperatura, pH y exposición a la luz solar.
2. Desarrollar un sistema de comunicación basado en tecnología LoRa® para asegurar la transmisión de datos a largas distancias de manera eficiente y con bajo consumo de energía.

3. Crear una plataforma web que permita visualizar los datos recolectados en casi tiempo real, brindando a los usuarios la posibilidad de monitorear las condiciones del cultivo desde cualquier lugar y en cualquier momento.
4. Implementar un sistema de notificaciones que alerte a los agricultores cuando los parámetros de los cultivos no estén dentro de los rangos óptimos o cuando se detecten anomalías, para facilitar una intervención oportuna y eficiente.

Capítulo 2

Agricultura Inteligente y Desarrollo de Proyectos IoT

2.1. Impacto del Cambio Climático en la Producción Agrícola

Las inquietudes sobre la variación climática presagian un efecto notable en la producción alimentaria debido a las modificaciones en factores climáticos fundamentales para el desarrollo de las cosechas, como la pluviosidad y las temperaturas. Las repercusiones del cambio climático en la producción agrícola diferirán según la región; no obstante, se prevé que las naciones situadas en áreas tropicales sean especialmente propensas.

Estos países generalmente enfrentan patrones de pluviosidad que fluctúan entre semiáridos y húmedos, lo que las hace más susceptibles a las alteraciones en las condiciones climáticas. Además, en las regiones semiáridas, se espera un incremento en la frecuencia y severidad de sequías y olas de calor, lo que podría restringir considerablemente el crecimiento y rendimiento de las cosechas en estas áreas. [6]

2.1.1. Agricultura

La agricultura es una actividad primordial que ha sido crucial para el desarrollo de las sociedades humanas a lo largo de la historia. Se define como el conjunto de técnicas y procedimientos destinados al cultivo de vegetales y la crianza de animales con el propósito de producir alimentos, fibras y otros productos esenciales para la vida cotidiana. No obstante, en la actualidad, la agricultura enfrenta una serie de retos significativos, entre los que sobresalen la variación climática, la escasez de recursos hídricos, el aumento poblacional y la necesidad de adoptar prácticas más sostenibles.

Las inquietudes relacionadas con la variación climática prevén un impacto notable en la producción alimentaria, afectando factores climáticos esenciales para el desarrollo de los cultivos, como la pluviosidad y las temperaturas.

En particular, los países situados en zonas tropicales son especialmente vulnerables a estos cambios. Estas naciones, que con frecuencia enfrentan regímenes de pluviosidad variables, desde condiciones semiáridas hasta húmedas, experimentan un aumento en la frecuencia e intensidad de sequías y olas de calor. Estos fenómenos extremos limitan significativamente el crecimiento y el rendimiento de los cultivos, lo que representa una seria amenaza para la seguridad alimentaria a nivel global.

2.1.2. Agricultura De Precisión

Para enfrentar estos retos, la agricultura inteligente ha emergido como una solución innovadora que integra tecnologías novedosas con prácticas agrícolas tradicionales. La agricultura inteligente emplea herramientas y sistemas basados en datos para optimizar la producción, mejorar la eficiencia en el uso de recursos y minimizar el impacto ambiental. Esto incluye la implementación de tecnologías como sensores, sistemas de monitoreo en tiempo real, análisis de datos y la conectividad del (IoT).

La investigación preliminar sobre la agricultura inteligente revela un creciente interés en la adopción de tecnologías que faciliten la gestión agrícola sostenible. Al examinar diversos documentos sobre el tema, se identificaron enfoques que subrayan el uso de tecnologías inalámbricas para resolver problemas en el ámbito agrícola. Estas tecnologías permiten un monitoreo más preciso de las condiciones del suelo y del clima, así como una gestión más eficaz del riego y la fertilización.

En respuesta a estos desafíos, tecnologías emergentes como el (IoT) están desempeñando un papel fundamental en la monitorización y administración de los recursos agrícolas. Compañías como Libelium y Atlas Scientific desarrollan soluciones tecnológicas avanzadas que permiten medir y analizar parámetros críticos en tiempo real. Estas soluciones no solo mejoran la eficacia en la gestión de cultivos, sino que también contribuyen a mitigar los efectos adversos de la variación climática en la agricultura. A continuación, se presentan detalles sobre las soluciones ofrecidas por estas dos compañías.

2.2. Soluciones Comerciales

2.2.1. Libelium

Libelium es una empresa española que desarrolla hardware y soluciones para el (IoT), destacándose en sectores como agricultura, medio ambiente, ciudades inteligentes e industria. Sus productos permiten la monitorización y automatización de procesos a través de la recolección y transmisión de datos en tiempo real, facilitando la toma de decisiones y optimización de recursos. [22]

Dentro de sus principales productos de Libelium se encuentra:

- **Waspnote:** Plataforma modular que permite conectar distintos sensores y módulos de comunicación.
- **Plug & Sense:** Solución preconfigurada lista para desplegar con sensores y conectividad inalámbrica.
- **Meshlium:** Gateway que recolecta y transmite datos a plataformas en la nube.
- **Smart Cities:** Kits para monitoreo de calidad del aire, gestión de residuos y control del tráfico.
- **Agricultura:** Kits que monitorean humedad del suelo, temperatura y radiación solar.
- **Software de Gestión:** Herramientas para la visualización y análisis de datos.

2.2.2. Atlas Scientific

Atlas Scientific es una empresa especializada en el desarrollo y fabricación de sensores de alta calidad para el análisis de agua y otros líquidos. Sus productos están diseñados para medir diversos parámetros químicos y físicos, como pH, ORP (potencial de reducción-oxidación), conductividad, oxígeno disuelto y temperatura. Para garantizar una integración eficiente y versátil en diversas aplicaciones, Atlas Scientific emplea tecnologías de comunicación inalámbrica, como *Wi-Fi*, *Bluetooth* y *LoRa®*. Estas tecnologías permiten la transmisión de datos en tiempo real, facilitando el monitoreo y la gestión de la calidad del agua en entornos industriales, ambientales y de investigación. Gracias a su enfoque en la precisión y la conectividad, *Atlas Scientific* contribuye a la implementación de soluciones innovadoras en el ámbito de la calidad del agua y la sostenibilidad. [35]

Sensores más utilizados:

- **Sensores de pH:** Miden la acidez o alcalinidad de un líquido.
- **Sensores ORP:** Detectan el potencial de reducción-oxidación en líquidos.
- **Sensores de Conductividad:** Miden la capacidad de los líquidos para conducir electricidad.
- **Sensores de Oxígeno Disuelto:** Cuantifican la cantidad de oxígeno en el agua.
- **Sensores de Temperatura:** Incluyen sondas RTD y termopares de alta precisión.
- **Sensores de Ion Selectivo y Turbidez:** Miden la concentración de iones específicos y la claridad del agua, respectivamente.
- **Sensores de CO₂:** Detectan la concentración de dióxido de carbono en líquidos.

Tecnologías de Comunicación Inalámbrica

- **Wi-Fi y Bluetooth:** Tecnologías comunes para conectar sensores.
- **LoRa:** Incluye LoRa E220 y módulos como RFM95/RFM96 y SX1276/SX1278 para transmisión de largo alcance.
- **Heltec WiFi LoRa 32 y Adafruit Feather M0:** Placas que combinan microcontroladores con módulos LoRa® para aplicaciones IoT.
- **Zigbee y Redes Celulares (4G/LTE):** Tecnologías adicionales para la conectividad de sensores.

2.3. Soluciones académicas en Agricultura Inteligente

La agricultura inteligente ha surgido como un enfoque novedoso para optimizar la producción agrícola y afrontar los retos contemporáneos en el sector. Con el propósito de comprender mejor este fenómeno, se llevó a cabo una investigación preliminar que incluyó la revisión de cinco documentos pertinentes. Esta revisión permitió identificar las tendencias actuales, las tecnologías emergentes y las mejores prácticas en la implementación de sistemas de agricultura inteligente, estableciendo las bases para un análisis más profundo de sus aplicaciones y ventajas.

Dichos documentos abordaban los problemas agrícolas y las soluciones tecnológicas utilizadas para abordarlos.

2.3.1. El prototipo de telecontrol para agricultura inteligente

Vera Polanco se centra en el desarrollo de un prototipo de telecontrol para agricultura inteligente, utilizando tecnologías emergentes como el (*IoT*) y la tecnología inalámbrica. Este proyecto responde a la necesidad de modernizar el sector agrícola mediante la implementación de soluciones tecnológicas que optimicen el uso de recursos y mejoren la eficiencia operativa, su objetivo principal es diseñar, desarrollar y evaluar un sistema de telecontrol que permita la gestión inteligente de los procesos agrícolas.

Este sistema se basa en la integración de sensores, actuadores y plataformas de comunicación inalámbrica como *Wi-Fi*, *Bluetooth* y *Zigbee* que facilitan el monitoreo y control en casi tiempo real de diversos parámetros agrícolas. Los resultados obtenidos demostraron que el prototipo de telecontrol desarrollado es una solución viable y efectiva para la agricultura inteligente. El sistema permitió una gestión más precisa y eficiente de los recursos agrícolas, mejorando la productividad y reduciendo el desperdicio de agua y otros insumos. [40]

2.3.2. Red de Sensores Inalámbricos bajo Protocolo LoRa® y Gestión de Procesos para la Analítica de Datos mediante Meta Sistema Operativo para Monitoreo Ambiental en Invernaderos

Por otro lado, el trabajo de Gordillo [18] se centra en el desarrollo de una red de sensores inalámbricos basada en el protocolo LPWAN y un meta sistema operativo, integrando algoritmos de aprendizaje automático para el análisis de variables ambientales como la radiación ultravioleta, el dióxido de carbono (CO), la temperatura y la humedad. Esta solución tiene como objetivo mejorar la precisión del riego y optimizar otras prácticas agrícolas mediante el procesamiento avanzado de los datos.

En cuanto a los resultados, la tesis muestra que el uso de la tecnología *LoRa*® es eficiente, permitiendo establecer una comunicación estable y de larga distancia, con una tasa de entrega de paquetes superior al 95 %. Además, se comprobó que el bajo consumo de energía de los nodos sensores, *LoRa*® permitía un funcionamiento prolongado sin necesidad de recargar o reemplazar las baterías, lo cual es fundamental para el monitoreo continuo en invernaderos.

2.3.3. Desarrollo de un micro servidor web con arduino

El desarrollo de Cristian David [5] consiste en un micro servidor web basado en la plataforma Arduino, diseñado para monitorear y analizar la calidad de los suelos en las parroquias rurales del cantón Latacunga. El objetivo principal del proyecto es implementar un sistema tecnológico que permita a los agricultores de estas zonas recopilar datos precisos y actualizados sobre las características de sus suelos, evitando así inspecciones manuales que suelen ser costosas y complejas.

Para alcanzar este objetivo, el micro servidor web está equipado con una variedad de sensores avanzados capaces de medir en casi tiempo real parámetros como pH, temperatura, humedad, entre otros.

En términos de desempeño, el proyecto ha logrado desarrollar y probar exitosamente un prototipo funcional del micro servidor web. Las pruebas de campo han demostrado que el sistema es capaz de recopilar y transmitir datos de manera confiable, incluso en las condiciones climáticas y geográficas desafiantes de las parroquias rurales.

2.3.4. Implementación de un sistema de monitoreo de PH aplicado a un invernadero de la zona alta sector Salache

Jhonny Javier y Ángel Orlando [9] proponen un sistema de monitoreo de pH para un invernadero ubicado en la zona alta del sector Salache, diseñando e implementando un sistema automatizado que incluye el uso de sensores de pH, un microcontrolador y un sistema de comunicación inalámbrica. Los sensores de pH recogen de manera continua los datos del suelo,

mientras que el microcontrolador procesa y transmite esta información de forma inalámbrica a un sistema de monitoreo central. De este modo, el personal del invernadero podrá supervisar y controlar remotamente las condiciones de pH del suelo, lo que les permitirá ajustar las condiciones del cultivo a tiempo y mantener un pH óptimo. Se espera que este sistema mejore la eficiencia del cultivo, reduzca los problemas relacionados con el pH y, como consecuencia, aumente la productividad y calidad de los productos del invernadero, destacando la importancia del monitoreo constante del pH.

2.3.5. Estudio sobre las características del protocolo de transmisión de *LoRa*

La investigación de Julián David Olivaresse [28] se enfoca en un análisis detallado del protocolo de transmisión *LoRa*®, destacando su relevancia y potencial en el desarrollo de soluciones para el (*IoT*). A través de un estudio profundo de la arquitectura, los principios operativos y el rendimiento de *LoRa*®, la tesis proporciona valiosa información sobre cómo esta tecnología puede adaptarse y aplicarse en el ámbito del *IoT*.

El autor subraya, entre otros aspectos, la excelente cobertura geográfica y capacidad de penetración de *LoRa*®, características esenciales para el despliegue de redes *IoT* a gran escala en áreas urbanas y rurales. Además, se aborda la eficiencia energética del protocolo, un factor crucial para permitir que los dispositivos *IoT* alimentados por baterías operen durante largos períodos sin necesidad de recarga.

Otro aspecto analizado es la estabilidad de *LoRa*®, que permite soportar un gran número de nodos conectados, lo que resulta fundamental en aplicaciones *IoT* con miles o millones de dispositivos. La revisión de la literatura científica realizada por el autor también le permitió conocer la implementación de *LoRa*® en aplicaciones concretas de *IoT*, como redes de sensores urbanos, monitoreo ambiental y agricultura de precisión. Esta información ofrece valiosas lecciones sobre los desafíos, las mejores prácticas y las experiencias aprendidas en el despliegue de soluciones *IoT* basadas en *LoRa*®.

Finalmente, la tesis señala áreas que requieren mayor atención, como la seguridad de las comunicaciones y la optimización del uso del espectro radioeléctrico, aspectos clave para garantizar el desarrollo futuro de *LoRa*® como una tecnología robusta y confiable en el *IoT*.

Conclusiones en la parte comercial

Libelium ofrece soluciones *IoT* altamente flexibles y modulares, ideales para proyectos de gran escala en sectores como la agricultura, las ciudades inteligentes y la gestión de recursos naturales. Su plataforma Waspote permite crear sistemas personalizados según las necesidades específicas de cada proyecto, con una amplia gama de sensores y opciones de conectividad, como *LoRa*®, Sigfox, Wi-Fi y Bluetooth. Además, la integración con plataformas de gestión de

datos facilita la toma de decisiones informadas en tiempo real, reduciendo la complejidad en la implementación y mantenimiento de los proyectos.

No obstante, su dependencia de redes externas como LoRa® o Sigfox puede afectar el rendimiento en situaciones que requieren mayor ancho de banda o cobertura global constante. Además, el alto costo inicial y la complejidad técnica asociada a su uso pueden representar barreras para empresas pequeñas o proyectos con presupuestos limitados. A pesar de la alta calidad de sus soluciones, la precisión de algunos sensores en aplicaciones muy específicas, como la detección de gases peligrosos o la agricultura de precisión, podría no ser comparable con productos más especializados.

Igualmente, la interoperabilidad con otras plataformas y tecnologías no siempre es perfecta, lo que podría generar problemas en el futuro al incorporar nuevas soluciones. Si bien Libelium es una opción destacada para proyectos grandes y complejos que requieren personalización, escalabilidad y flexibilidad. Sin embargo, su costo, la complejidad técnica y la dependencia de infraestructuras externas pueden representar un desafío para quienes cuentan con presupuestos limitados o necesidades menos especializadas.

Atlas Scientific se destaca por ofrecer sensores de alta precisión, ideales para aplicaciones científicas, industriales y de monitoreo ambiental. Sus productos son conocidos por su facilidad de integración con plataformas como Arduino y Raspberry Pi, lo que permite crear sistemas personalizados sin necesidad de contar con un alto nivel de experiencia técnica. La amplia gama de sensores disponibles, que incluyen pH, oxígeno disuelto y conductividad, hace que Atlas sea una opción preferida para proyectos de monitoreo ambiental, acuicultura y biotecnología. Además, la durabilidad de sus dispositivos asegura un rendimiento constante en entornos exigentes, lo que minimiza el mantenimiento.

A pesar de sus múltiples ventajas, existen algunas desventajas que deben ser consideradas. El costo de los productos de Atlas puede resultar elevado, especialmente para proyectos con presupuestos limitados. Aunque la precisión de los sensores justifica este precio, algunos usuarios podrían optar por alternativas más económicas, aunque con menor precisión. Además, los sensores de Atlas requieren reemplazo o recalibración periódica, lo que genera costos adicionales a largo plazo. Este mantenimiento frecuente puede ser un inconveniente para proyectos que buscan una solución con bajos costos operativos.

La implementación y calibración de los sensores también pueden ser complejas, especialmente para quienes no tienen experiencia en electrónica. Aunque la integración con plataformas como Arduino y Raspberry Pi es sencilla, una calibración incorrecta puede afectar la precisión de las mediciones, lo que comprometería la fiabilidad de los resultados. Además, algunos de los sensores pueden ser sensibles a condiciones ambientales extremas, lo que podría afectar su rendimiento y requerir protección adicional en entornos difíciles.

Por último, aunque Atlas ofrece una amplia variedad de sensores, algunos proyectos especializados podrían necesitar sensores personalizados, los cuales no siempre están disponibles en

su catálogo. Además, ciertos accesorios, como kits de calibración o cables específicos, no están incluidos en el paquete inicial y pueden aumentar el costo total del proyecto.

En conclusión, Atlas Scientific ofrece soluciones de alta calidad, precisas y duraderas, ideales para proyectos que requieren control exacto de variables críticas. Sin embargo, el costo, la necesidad de mantenimiento frecuente y la complejidad en la implementación son factores que deben tenerse en cuenta al elegir sus productos para proyectos con presupuesto limitado o requerimientos específicos.

Soluciones en la parte académica

En los estudios recientes, se destacan avances significativos en la agricultura moderna gracias a las tecnologías emergentes. En particular, el prototipo de telecontrol para agricultura inteligente demuestra la eficacia del uso del (IoT) y tecnologías inalámbricas en la optimización de recursos, así como en la mejora de la eficiencia operativa.

En cuanto a las ventajas, en primer lugar, este sistema permite a los agricultores monitorear en tiempo real parámetros críticos como temperatura, humedad y luminosidad. De esta manera, se facilita una toma de decisiones más informada y precisa. Además, la tecnología contribuye a la eficiencia y sostenibilidad al optimizar el uso de recursos, lo que reduce tanto los costos como el tiempo en la recolección de datos. Finalmente, fomenta la integración de tecnología, impulsando la innovación y mejorando la competitividad en el sector agrícola a través del uso de IoT y tecnología inalámbrica.

No obstante, el uso de estas tecnologías también presenta ciertas desventajas. Por ejemplo, la implementación de estas soluciones puede generar una dependencia tecnológica, lo cual representa un desafío en áreas con infraestructura limitada. Asimismo, la adopción del prototipo requiere una inversión considerable en tecnología y capacitación, lo que podría ser una barrera para algunos agricultores debido a los costos iniciales. Finalmente, la integración de dispositivos y sistemas puede ser compleja, demandando tanto tiempo como recursos adicionales para asegurar una correcta implementación.

Para enfrentar estos desafíos, se han propuesto diversas mejoras. En primer lugar, es fundamental ofrecer capacitación continua a los agricultores sobre el uso y mantenimiento del sistema, asegurando así su correcta aplicación y maximización de beneficios. Además, se hace necesario desarrollar soluciones para la optimización de energía, de modo que se gestione de manera más eficiente el consumo energético de los dispositivos, aumentando su sostenibilidad. Finalmente, sería ideal implementar un sistema de retroalimentación constante que permita recoger opiniones de los usuarios, mejorando el prototipo en función de sus necesidades y experiencias.

El segundo trabajo, enfocado en la implementación de redes de sensores inalámbricos bajo el protocolo LoRa®, junto con un meta sistema operativo, propone una plataforma robusta para el monitoreo ambiental. Los resultados obtenidos evidencian una alta tasa de entrega de datos y un bajo consumo energético, aspectos fundamentales para la sostenibilidad de los sistemas agrícolas.

Respecto a las ventajas, en primer lugar, este sistema contribuye a una mejora en la producción agrícola, generando beneficios económicos significativos para el dueño del invernadero. Además, su diseño como sistema de monitoreo de bajo costo facilita la toma de decisiones al proveer información clave de manera eficiente. Por último, la incorporación de una interfaz gráfica permite una fácil interpretación de los datos recolectados, haciendo el sistema accesible y amigable para el usuario.

Sin embargo, existen también algunas desventajas a considerar. Por ejemplo, la distancia entre invernaderos y herramientas puede dificultar el acceso y la gestión de los datos recolectados. Asimismo, existe una dependencia tecnológica inherente a este sistema, lo cual puede presentar problemas si ocurre algún fallo en la infraestructura. Finalmente, el sistema requiere capacitación específica, ya que los usuarios necesitan entrenamiento en el manejo adecuado de la plataforma para maximizar su efectividad.

Para abordar estos desafíos, se han propuesto varias mejoras. En primer lugar, sería beneficioso implementar un sistema de transporte eficiente para facilitar el traslado y acceso a las herramientas en áreas de difícil acceso. Además, es importante ofrecer capacitación y soporte técnico continuo a los usuarios, asegurando que comprendan y puedan manejar el sistema de manera óptima. Por último, la optimización de la red de sensores es esencial para mejorar la precisión y eficiencia en la recolección de datos, garantizando una supervisión ambiental aún más efectiva.

Además, el **Desarrollo de un micro servidor web basado en Arduino para el monitoreo** de la calidad del suelo permite a los agricultores rurales acceder a información crucial sin necesidad de realizar inspecciones manuales, lo cual mejora significativamente la toma de decisiones. La implementación de un sistema automatizado de monitoreo de pH en invernaderos destaca la importancia del monitoreo continuo y su impacto positivo en la productividad agrícola.

Respecto a las ventajas del sistema, en primer lugar, se fomenta una colaboración constante entre el cliente y el equipo de desarrollo, lo que asegura que el sistema final cumpla con las necesidades del usuario. Además, el uso de metodologías ágiles, como la programación extrema (XP), facilita el proceso de programación y mejora la calidad del código mediante revisiones constantes y programación en pareja. Sin embargo, el sistema también presenta ciertas desventajas. Por ejemplo, la falta de una metodología estrictamente definida puede dificultar la colaboración y coordinación en el equipo de trabajo. Asimismo, metodologías ágiles como XP resultan complicadas si no se cumplen todas las fases del proceso de manera rigurosa, lo cual podría afectar la eficiencia del desarrollo.

Para abordar estos desafíos, se han propuesto diversas mejoras. En primer lugar, es recomendable implementar una estructura de trabajo definida que favorezca la colaboración y la comunicación efectiva dentro del equipo. Además, resulta importante aumentar la estabilidad y seguridad del sistema operativo utilizado y mejorar la compatibilidad en versiones anteriores de PHP, para asegurar un entorno de trabajo confiable y seguro.

La implementación de un sistema de monitoreo de pH en un invernadero tiene como objetivo medir y controlar el pH del suelo y del agua para optimizar el desarrollo de los cultivos. Este sistema utiliza sensores de pH que envían datos a un programa en LabVIEW, el cual permite visualizar los resultados en un sistema SCADA. De este modo, facilita el monitoreo constante por parte de estudiantes de agronomía, mejorando la productividad y reduciendo las pérdidas en los cultivos.

En cuanto a las ventajas, en primer lugar, el sistema permite un control preciso del pH del suelo y del agua, asegurando condiciones óptimas para el crecimiento de los cultivos, lo cual impacta positivamente en la productividad. Además, la automatización mediante SCADA y LabVIEW permite el monitoreo en tiempo real sin necesidad de intervención constante, lo que reduce el esfuerzo manual y minimiza el error humano. También se observa una mejora en la eficiencia del uso de recursos, ya que el ajuste adecuado del pH optimiza el uso de fertilizantes y agua, contribuyendo a la reducción de costos.

Adicionalmente, el sistema tiene un beneficio educativo significativo, pues permite a los estudiantes de agronomía aprender de forma práctica sobre el manejo automatizado de invernaderos y la importancia de controlar variables críticas como el pH. Por último, el control adecuado del pH aumenta la producción, dado que mejora la salud de las plantas y disminuye las pérdidas económicas.

No obstante, el sistema presenta algunas desventajas. Por ejemplo, aunque se considera una alternativa económica, el costo inicial de implementación de sensores, el sistema SCADA y la infraestructura tecnológica puede representar una inversión considerable, especialmente para pequeños agricultores. Además, el mantenimiento de los sensores de pH, que están en contacto constante con el suelo y el agua, es fundamental para garantizar lecturas precisas, lo cual implica calibraciones y revisiones constantes.

La dependencia tecnológica también es una desventaja, ya que cualquier fallo en los sensores o en el software podría afectar el monitoreo y el control del invernadero. Finalmente, se requiere capacitación para utilizar y mantener el sistema, ya que los usuarios necesitan conocimientos en tecnologías como SCADA y LabVIEW, lo que podría implicar capacitaciones adicionales.

Para mejorar el sistema, se proponen diversas mejoras. Una opción sería la optimización energética mediante la inclusión de fuentes de energía renovable, como paneles solares, lo cual haría el sistema más sostenible y reduciría los costos operativos a largo plazo. Asimismo, la integración de más variables permitiría un control más integral del ambiente en el invernadero, añadiendo parámetros como temperatura, humedad y nivel de luz. El uso de tecnologías IoT también contribuiría, al permitir el acceso al sistema desde cualquier lugar mediante dispositivos móviles y el envío de alertas cuando las condiciones estén fuera de los parámetros óptimos.

Otra mejora sería la automatización avanzada, añadiendo un sistema de riego que ajuste la cantidad de agua en función de las mediciones de pH y otros parámetros para maximizar la eficiencia. Finalmente, la implementación de sensores más robustos y resistentes al desgaste

reduciría los costos de mantenimiento a largo plazo.

La implementación del protocolo de transmisión LoRa®, junto con sus características técnicas, lo posiciona como una tecnología relevante para aplicaciones de (IoT) en redes de área amplia de baja potencia (LPWAN). LoRa® y LoRaWAN® destacan por su capacidad de transmitir datos a largas distancias manteniendo un bajo consumo de energía, lo que las hace ideales para el monitoreo remoto, especialmente en entornos rurales o zonas con dispositivos dispersos. Su diseño está orientado a aplicaciones que requieren una cobertura amplia y confiable con recursos energéticos limitados.

Entre las ventajas generales de LoRa® se encuentran varias características que la convierten en una tecnología accesible y eficiente para el sector agrícola y otras industrias que dependen del monitoreo a larga distancia. Primero, LoRa® ofrece una cobertura extensa sin necesidad de muchas estaciones base, lo que es ideal para aplicaciones en áreas rurales o de difícil acceso. Además, es una tecnología energéticamente eficiente; los dispositivos pueden operar durante largos periodos con baterías pequeñas, lo cual es crucial para reducir los costos de mantenimiento y prolongar la vida útil de los sensores.

El uso de frecuencias en la banda ISM, que no requieren licencia, también hace que la implementación de LoRa® sea más accesible en términos económicos. Asimismo, LoRa® es resistente a las interferencias gracias a su modulación de espectro ensanchado, lo que asegura una transmisión de datos más confiable en entornos con ruido de señal.

Pese a sus ventajas, LoRa® presenta algunas limitaciones que pueden restringir su aplicabilidad en ciertos escenarios. La velocidad de transmisión de datos es baja, lo cual dificulta su uso en aplicaciones que necesitan enviar grandes volúmenes de datos o comunicaciones en tiempo real. Además, el rendimiento de la red puede verse afectado a medida que aumenta el número de dispositivos conectados, ya que una alta densidad de dispositivos puede provocar colisiones y reducir la calidad del servicio.

En áreas donde existen múltiples redes o tecnologías que comparten la banda ISM, pueden surgir problemas de interferencia, lo cual es especialmente relevante en entornos congestionados. Finalmente, la dependencia de la infraestructura de red y la tecnología puede limitar su efectividad en aplicaciones que demanden alta disponibilidad o fiabilidad en la transmisión de datos.

Para mejorar el rendimiento de LoRa®, existen varias áreas de oportunidad. Optimizar la tasa de datos adaptable (ADR) puede mejorar la eficiencia de la red al ajustar la velocidad de transmisión en función de la distancia y las condiciones de señal, lo cual reduce el riesgo de colisiones en redes densas. También es posible mejorar la escalabilidad mediante la implementación de algoritmos avanzados de gestión de tráfico y priorización, lo que ayudaría a reducir las colisiones en redes con un gran número de dispositivos conectados.

Además, ampliar el uso de bandas de frecuencia o habilitar el uso de otras bandas podría

mitigar las interferencias en entornos congestionados. Incorporar herramientas de monitoreo y ajuste de rendimiento en tiempo real permitiría una optimización continua de la red, adaptándola según las condiciones cambiantes del entorno. Finalmente, fortalecer los aspectos de seguridad, especialmente en aplicaciones críticas, aseguraría una protección adecuada de los datos transmitidos en la red.

En conclusión, LoRa® y LoRaWAN® ofrecen una solución de bajo costo y bajo consumo energético para el monitoreo en redes de IoT de larga distancia. Aunque presentan limitaciones en términos de velocidad de transmisión y escalabilidad, las ventajas de cobertura y eficiencia energética hacen que esta tecnología sea viable para aplicaciones específicas que no requieren grandes volúmenes de datos ni comunicación en tiempo real. Con ajustes en la gestión de tráfico, el uso de bandas de frecuencia adicionales y herramientas de monitoreo, LoRa® podría ser optimizada para cumplir con las demandas de redes IoT en escenarios más exigentes.

Capítulo 3

Tecnologías Inalámbricas

Las tecnologías de comunicación inalámbrica han transformado la manera en que interactuamos con el mundo, permitiendo conexiones rápidas, seguras y eficientes sin necesidad de cables. Estas tecnologías aprovechan el espectro electromagnético, un recurso natural compuesto por ondas de distintas frecuencias, para transmitir información a través del aire.

Entre las más destacadas se encuentra el Wi-Fi, ampliamente utilizado en hogares y oficinas para conectar dispositivos a internet; el Bluetooth, que permite la transferencia de datos a corta distancia entre dispositivos como teléfonos, auriculares y computadoras; y las bandas de frecuencia no licenciadas, conocidas como ISM (Industrial, Scientific, and Medical), que sirven como base para diversos sistemas de comunicación y dispositivos IoT.

El espectro electromagnético, como fundamento de estas tecnologías, incluye frecuencias que van desde las ondas de radio hasta los microondas ver figura 3.1 y otras más avanzadas. Su uso eficiente y regulado es clave para garantizar el funcionamiento simultáneo de múltiples sistemas sin interferencias.

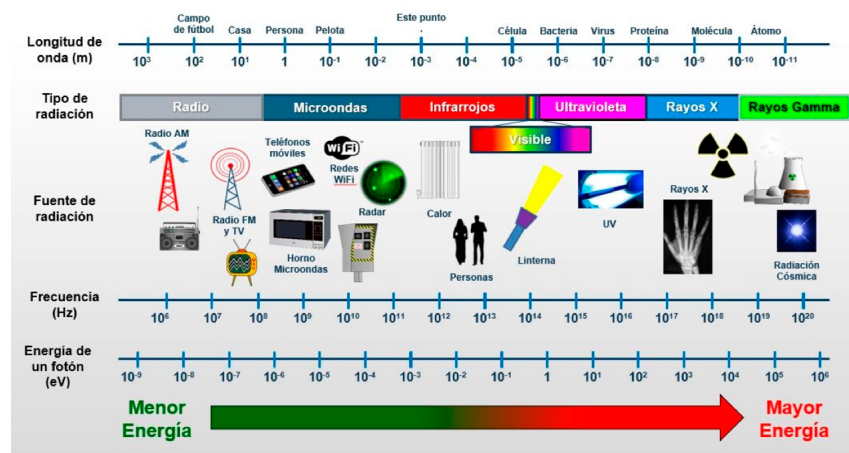


Figura 3.1: Espectro electromagnético, algunas fuentes de cada tipo de OEM y la relación de la longitud de onda con diferentes objetos o entidades. [32]

Estas tecnologías no solo han facilitado la conectividad en la vida cotidiana, sino que también son pilares fundamentales en la industria, la medicina, la investigación científica y la automatización de procesos. Su estudio y desarrollo constante continúan impulsando innovaciones que transforman nuestra sociedad.

3.1. ISM

Es importante señalar que las bandas de frecuencias clasificadas como “espectro libre” se utilizan comúnmente en diversas aplicaciones, tales como redes locales y personales inalámbricas (Wi-Fi y Bluetooth), sistemas de radioenlaces punto a punto y punto a multipunto, radios de dos vías, dispositivos de radiocomunicación de corto alcance, sensores de perturbación de campo, entre otras aplicaciones de control inalámbrico a distancia.

De acuerdo con el IFT, el “espectro libre” se refiere a aquellas bandas de frecuencia que son de acceso libre, permitiendo su uso por el público en general bajo los lineamientos o especificaciones que el instituto determine, sin necesidad de concesión o autorización adicional.

En este sentido, para el desarrollo de este trabajo se han seleccionado módulos LoRa® que operan en frecuencias libres. Concretamente, el módulo E220 funciona en el rango de frecuencias de 410.125 a 493.125 MHz, las cuales, según las regulaciones del IFT, están clasificadas como bandas para aficionados. [11]

3.2. WiFi

El WiFi es una tecnología de red inalámbrica que permite la conexión a Internet y la comunicación entre dispositivos dentro de una área determinada, generalmente en un entorno doméstico o de oficina. Opera en las bandas de frecuencia de 2.4 GHz y 5 GHz, lo que le permite ofrecer altas velocidades de transmisión de datos, generalmente desde varios megabits por segundo hasta gigabits. Wi-Fi es ideal para aplicaciones que requieren un alto ancho de banda, como la transmisión de video en alta definición y juegos en línea, aunque su alcance es limitado, típicamente alrededor de 30 a 100 metros, dependiendo de las condiciones. [10]

3.3. Bluetooth

Bluetooth es una tecnología de comunicación inalámbrica diseñada para la transferencia de datos a corta distancia entre dispositivos. Utiliza la banda de 2.4 GHz y está optimizada para el consumo de energía, lo que la convierte en una opción popular para dispositivos portátiles y aplicaciones que requieren conexión a corta distancia, como auriculares, relojes inteligentes y altavoces. Bluetooth tiene un rango de operación de aproximadamente 10 a 100 metros y es más adecuado para la transmisión de datos de menor volumen, como audio o archivos pequeños. [8]

3.4. *LoRa*®

LoRa® (acrónimo de *Long Range*) es una tecnología de comunicación diseñada específicamente para lograr **largo alcance** y **bajo consumo energético** en aplicaciones de IoT. Opera en frecuencias **sub-GHz** (como 433 MHz, 868 MHz y 915 MHz), permitiendo la transmisión de datos a varios kilómetros, incluso en entornos urbanos. Es especialmente útil en dispositivos que envían pequeñas cantidades de datos a intervalos regulares, como sensores ambientales, contadores inteligentes y dispositivos de monitoreo agrícola. En estos casos, la eficiencia energética y el alcance son más críticos que la velocidad de transmisión.

LoRa® constituye la **capa física** de las conexiones en redes **LoRaWAN**®, las cuales han sido diseñadas para ofrecer comunicaciones de larga distancia y bajo consumo energético. Al operar en la banda ISM, emplea una modulación basada en **CSS (Chirp Spread Spectrum)**, similar en algunos aspectos a la modulación FSK.

Una de las principales ventajas de *LoRa*® es que una sola estación base puede cubrir áreas extensas, como una ciudad completa, aunque su alcance puede verse limitado por obstáculos presentes en el entorno. Sin embargo, su costo de enlace es superior al de tecnologías de comunicación más estandarizadas, lo cual puede influir en su implementación dependiendo del contexto.

- **Canal de comunicación:** Es el medio a través del cual se transmite la señal, determinado por la frecuencia central del canal. Este parámetro varía según la regulación de cada país y el número de canales disponibles.
- **Ancho de banda:** Representa el rango de frecuencias utilizado durante la transmisión. Un mayor ancho de banda aumenta la velocidad de transmisión, pero reduce la sensibilidad debido a la mayor susceptibilidad al ruido.
- **Factor de ensanchamiento (Spreading Factor, SF):** Este parámetro define la cantidad de bits utilizados para codificar un símbolo. Un aumento en el SF mejora la protección contra interferencias y amplía el alcance, pero disminuye la velocidad de transmisión.
- **Potencia de transmisión:** Corresponde a la potencia de la señal transmitida. Está regulada por las normativas locales y limitada por las capacidades del hardware.
- **Tasa de codificación (Coding Rate, CR):** La *tasa de codificación* implementa técnicas de corrección de errores (*Forward Error Correction, FEC*), que mejoran la robustez de la señal frente a interferencias. Sin embargo, un CR más alto implica una menor velocidad de transmisión. [7]

La tecnología *Lora*® se desarrolló en Francia en 2012, usa un tipo de modulación de amplio espectro, ideal para tolerar el ruido y para que una señal realice caminos múltiples. Presenta un ancho de banda reducido, pero adaptado a las necesidades de los dispositivos. Uno de los propósitos fundamentales del es darle conectividad a dispositivos remotos para que puedan enviar y recibir información, y puedan permanecer prolongados períodos de tiempo funcionando

sin necesidad de mantenimiento

Las redes LPWAN (Low Power Wide Area Network) están tomando una relevancia cada vez mayor, debido a que pueden conectar dispositivos a grandes distancias con un consumo de energía realmente bajo. Dentro de las LPWAN podemos encontrar varias tecnologías disponibles. LoRa® es una de ellas y cuenta con muy buenas características para proyectos de IoT.

El nombre LoRa® es un acrónimo de Long Range y consiste en una técnica de modulación de radiofrecuencia patentada por Semtech. Es la capa física que provee a los dispositivos de comunicación inalámbrica de largo alcance y bajo consumo. [1]

3.4.1. Modulación y transmisión LoRa

La tecnología LoRa® utiliza una modulación de espectro ensanchado (CSS) para transmitir datos a través del aire. Los datos se transmiten en forma de paquetes, utilizando una de las varias técnicas de modulación disponibles en LoRa, como LoRa modulation o FSK (Frequency Shift Keying) [30].

La señal del módulo LoRa e220 puede alcanzar distancias de 5 Km en entornos rurales, con una potencia de 22 dBm, según las condiciones del terreno y otros factores ambientales. [14]

$$F_{FSK}(t) = \begin{cases} A \cos(\omega_1 t) & \text{si } f(t) = 1V \text{ (1 lógico)} \\ A \cos(\omega_0 t) & \text{si } f(t) = 0V \text{ (0 lógico)} \end{cases}$$

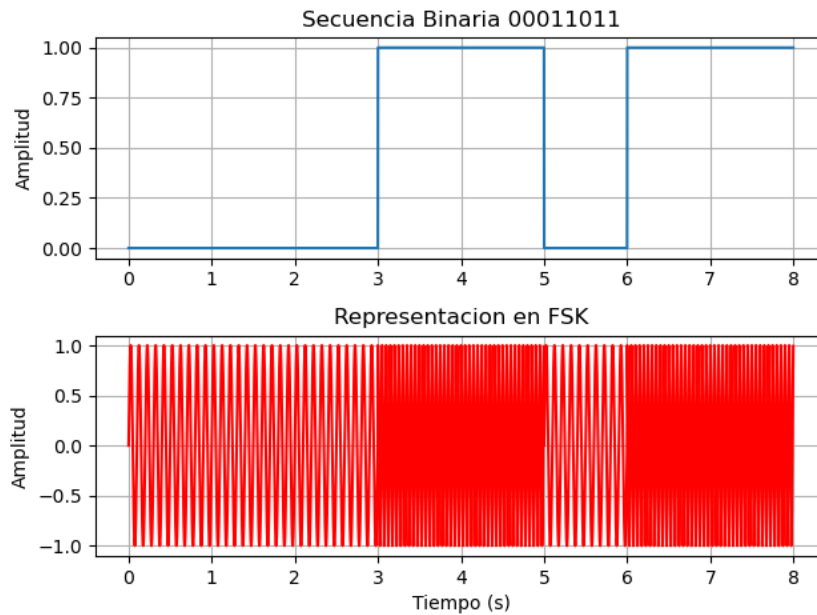


Figura 3.2: En la siguiente imagen se puede observar una modulación 4-FSK, lo que implica que es un FSK con cuatro símbolos distintos ('00', '01', '10' y '11'), permitiendo la transmisión de dos bits por símbolo.

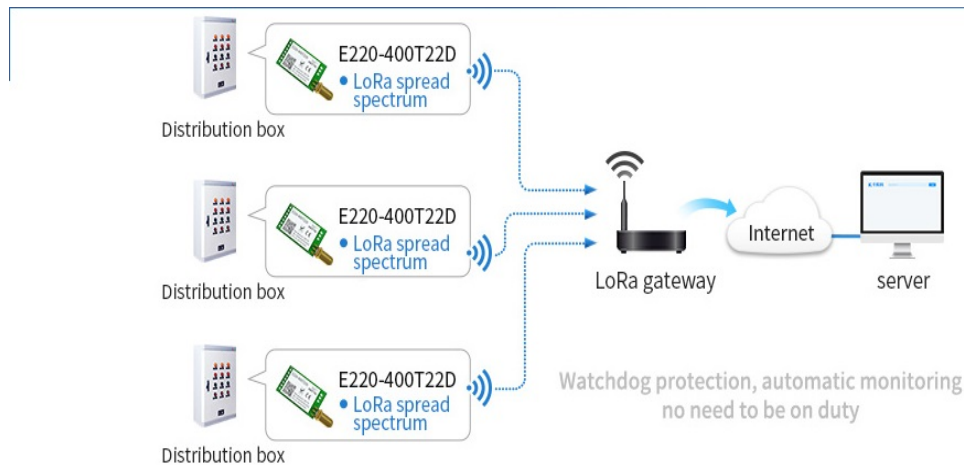


Figura 3.3: Sistema simple LoRa [14]

3.5. Protocolo de Comunicación UART

UART (Universal Asynchronous Receiver/Transmitter) es un protocolo de comunicación serial utilizado en la electrónica para transferir datos entre dispositivos. La UART se encarga de la transmisión y recepción de datos de manera secuencial, uno tras otro, a través de dos pines: uno

para transmitir (TX) y otro para recibir (RX). Aunque se llama asincrónico , la UART puede operar de manera síncrona o asincrónica. [31]

Comparación de las tecnologías inalámbricas: Wi-Fi, Bluetooth y LoRa

Característica	Wi-Fi	Bluetooth	LoRa
Rango de cobertura	50 - 100 m	10 - 100 m	Hasta 10 km
Velocidad de transmisión	Hasta 1 Gbps	Hasta 3 Mbps	0.3 - 50 Kbps
Consumo de energía	Alto	Bajo	Muy bajo
Costo de implementación	Medio a alto	Bajo	Bajo
Frecuencia	2.4 GHz, 5 GHz	2.4 GHz	433/868/915 MHz
Aplicaciones típicas	Internet, video, multimedia	Dispositivos personales, IoT	IoT, redes de sensores
Uso en agricultura	Monitoreo de alta demanda de datos	Comunicación a corta distancia	Monitoreo a gran escala

Cuadro 3.1: El cuadro compara Wi-Fi, Bluetooth y LoRa: Wi-Fi ofrece gran velocidad pero alto consumo, Bluetooth es de corto alcance y bajo consumo, y LoRa destaca por su largo alcance y bajo consumo, ideal para IoT.

Características de la Comunicación UART

- **Velocidad de bits:** Antes de comenzar la comunicación, tanto el dispositivo transmisor como el receptor deben acordar una velocidad de bits, que especifica la cantidad de bits que se enviarán o recibirán por segundo.
- **Bits de datos:** Se debe acordar la cantidad de bits de datos que se transmitirán en cada trama. Los valores comunes son 8, 7 o 9 bits, siendo 8 bits el más común.
- **Bit de inicio:** Cada trama de datos comienza con un bit de inicio que indica el inicio de la transmisión.
- **Bits de paridad (opcional):** A veces, se incluye un bit de paridad para la detección de errores. Puede ser paridad par, impar o ninguno.

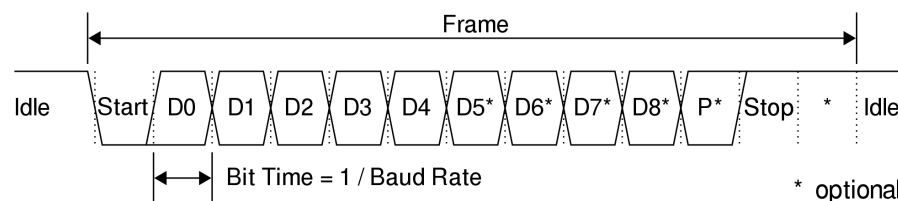


Figura 3.4: Datagrama del protocolo *UART* [13]

Paridad

- **Paridad Par:** En este método, el bit de paridad se elige para que el número total de unos en el conjunto de bits, incluido el bit de paridad, sea par. Por ejemplo, si tienes un conjunto de datos como 1010110, contarías el número de unos (cuatro en este caso) y agregarías un bit adicional para hacer que el total de unos sea par. Entonces, el conjunto de bits con el bit de paridad sería 10101100.
- **Paridad Impar:** En este método, el bit de paridad se elige para que el número total de unos en el conjunto de bits, incluido el bit de paridad, sea impar. Usando el mismo ejemplo anterior, el conjunto de datos 1010110 tendría un total de cuatro unos. Para mantener la paridad impar, se agregaría un bit adicional para que el total de unos sea impar. Por lo tanto, el conjunto de bits con el bit de paridad sería 10101101.

Bits de Alto

Cada trama de datos termina con uno o dos bits de alto, que indican el final de la transmisión. [12]

Ejemplo de Transmisión de Datos:

Al transmitir datos a través de una conexión UART, es esencial comprender cómo se maneja el orden de los bits en el proceso. Por ejemplo, al enviar el número 87 (correspondiente al carácter 'W' en la tabla ASCII) desde el transmisor al receptor, el byte se carga en el FIFO (First In, First Out) de transmisión.

La UART se configura para operar a 9600 baudios, 8 bits de datos, sin bits de paridad y 1 bit de parada. Transmite el carácter 'W' (01010111 en binario): Bit de inicio: 0 Bits de datos: 1010111 Bit de parada: 1.

Receptor: De igual manera se configuro UART de manera idéntica a la del transmisor, los datos entrantes como el bit de inicio, se reciben y se decodifican los bits de datos. Final mente se verifica el bit de parada al final de la trama. [16]

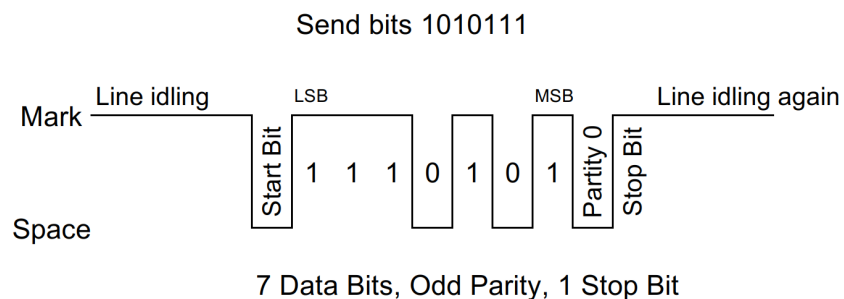


Figura 3.5: Transmisión de la letra W. [16]

3.5.1. FIFO en UART

FIFO (First In, First Out) es un tipo de buffer o cola que se utiliza en muchos sistemas de comunicación, incluyendo la UART. En el contexto de UART, el FIFO es una memoria intermedia que almacena temporalmente los datos que se van a transmitir o recibir.

En un FIFO, los datos se almacenan en el orden en que se ingresan, lo que significa que el bit más significativo (MSB) del byte se coloca primero en el FIFO. Por lo tanto, el bit '0' del byte 87 se coloca en la parte superior del FIFO, seguido por los bits restantes en orden secuencial. Cuando se inicia la transmisión, el FIFO vacía sus datos en la línea de transmisión UART. Dado que UART transmite el bit más significativo primero, el bit '0' se envía primero, seguido por los bits restantes hasta que se transmite el bit menos significativo del byte. [21]

Función del FIFO en UART

- **Transmisión:** Cuando envías datos a través de UART, el FIFO de transmisión (TX FIFO) guarda los bytes que serán transmitidos, permitiendo que la CPU siga trabajando en otras tareas mientras los datos se transmiten en segundo plano.
- **Recepción:** De manera similar, el FIFO de recepción (RX FIFO) almacena los bytes recibidos hasta que la CPU esté lista para procesarlos, ayudando a evitar la pérdida de datos si llegan más bytes de los que la CPU puede procesar inmediatamente.

Beneficios del FIFO en UART

- **Reducción de carga en la CPU:** Al permitir que los datos se acumulen en el FIFO, la CPU no tiene que atender cada byte en el instante en que se recibe o se transmite, mejorando la eficiencia general del sistema.
- **Mayor tolerancia a diferencias de velocidad:** El FIFO permite manejar ráfagas de datos a mayor velocidad, ya que los datos pueden esperar en el buffer sin que se pierdan.

3.6. Código CRC

Cuando se interrumpe la comunicación entre el transmisor (TX) y el receptor (RX), el sistema emplea un protocolo que se basa en la verificación continua del estado del módulo y la gestión de tiempos de espera (timeouts). El funcionamiento del sistema en estas circunstancias está diseñado en torno a dos mecanismos clave. Este protocolo se apoya en la tecnología de modulación *LoRa Spread Spectrum*, que permite la transmisión de datos en largas distancias con bajo consumo de energía y resistencia a interferencias.

El primer mecanismo es la verificación del estado de ocupación del módulo, que se lleva a cabo mediante la monitorización del cambio de estado del Pin AUX. Este pin actúa como un indicador del estado interno del módulo *LoRa®*, permitiendo detectar si el canal está ocupado o si el módulo está disponible para transmitir o recibir datos. En el caso de que el canal esté

ocupado, el sistema entra en un modo de espera y ajusta los tiempos de transmisión para evitar colisiones de señal y asegurar que los datos lleguen al receptor de manera eficiente una vez que el canal esté libre.

El segundo mecanismo está relacionado con el manejo de tiempos de espera, donde el sistema monitorea la duración de la pérdida de comunicación. Si se excede un tiempo determinado sin restablecer la conexión, el sistema puede desencadenar acciones correctivas como reintentos de envío o alertas para indicar que el enlace se ha perdido. Ambos mecanismos trabajan en conjunto para garantizar la estabilidad y confiabilidad del sistema en aplicaciones de monitoreo a largo plazo y de gran alcance.

Este procedimiento permite al sistema identificar si el transmisor está ocupado o si hay algún retraso al enviar un paquete de comprobación. Si el Pin AUX indica un estado bajo, lo que señala que el módulo aún está procesando, el sistema entra en un modo de espera, permitiendo que el módulo finalice sus tareas antes de intentar una nueva transmisión.

En caso de que, pasado el lapso de 5 segundos, el transmisor continúe ocupado, no se ejecutan acciones adicionales en ese ciclo. La transmisión de datos se detiene temporalmente, pero el sistema no tiene un método específico para manejar la pérdida completa de comunicación. En su lugar, cada ciclo revisa si el transmisor sigue ocupado, y de ser así, se reanuda el intento de comunicación en el siguiente ciclo, permitiendo que los intentos continúen de forma periódica sin detener la operación por completo.

El enfoque del sistema es permitir que el módulo concluya sus actividades pendientes a través de un ciclo de espera. Si después de 5 segundos la comunicación no se recupera, el sistema sigue intentando en los próximos ciclos, aunque no cuenta con un mecanismo dedicado para manejar una desconexión permanente, manteniendo el intento constante de reconexión en cada iteración.

Por otro lado, la tecnología *LoRa*® utiliza un algoritmo llamado **CRC-16** (Comprobación de Redundancia Cíclica) de 16 bits. Este *CRC* se calcula y se añade al final de cada paquete de datos antes de enviarlo. Una vez recibido el paquete, el receptor vuelve a calcular el *CRC* y lo compara con el valor recibido, asegurando que no haya errores en la transmisión. El polinomio generador del *CRC-16* está representado por la ecuación:

$$x^{16} + x^{12} + x^5 + 1. \quad (3.2)$$

El uso del *CRC* en la tecnología *LoRa* es opcional, y su activación o desactivación puede depender tanto del fabricante del módulo *LoRa* como de la configuración específica del protocolo en el sistema. Algunos fabricantes permiten ajustar el uso del *CRC* en función de las necesidades del proyecto, ya que aunque el *CRC* agrega un nivel de seguridad al verificar la integridad de los datos transmitidos, también puede aumentar ligeramente la latencia y el consumo de energía.

En proyectos donde la confiabilidad de la transmisión es crítica, activar el *CRC* es una opción recomendada, mientras que en aplicaciones donde la eficiencia energética o la velocidad es prioritaria, podría desactivarse para optimizar el rendimiento.

Es importante tener en cuenta la seguridad durante todo el proceso de transmisión de datos. Esto puede implicar el cifrado de los datos transmitidos mediante LoRa y el uso de protocolos de comunicación seguros en el servidor central para proteger los datos recibidos.

Conclusiones del capítulo

El análisis presentado en este capítulo destaca la creciente relevancia de la agricultura inteligente como respuesta a los desafíos contemporáneos que enfrenta el sector agrícola, tales como el impacto del cambio climático, la escasez de recursos naturales y la necesidad de prácticas más sostenibles. La implementación de tecnologías avanzadas como el (IoT) y sistemas de monitoreo en tiempo real ofrece enfoques innovadores para optimizar la gestión de recursos, en particular el manejo del agua, el control del riego y la fertilización de los cultivos.

La revisión de la literatura y estudios de caso, como los desarrollos de empresas como Libelium y Atlas Scientific, evidencia que el uso de sensores especializados para medir variables clave —como humedad del suelo, temperatura, pH y exposición solar— permite una gestión más precisa y eficiente de los cultivos. Asimismo, las tecnologías de comunicación inalámbrica como LoRa® destacan como una solución efectiva para áreas rurales, gracias a su bajo consumo energético y su capacidad de transmitir datos a largas distancias.

En este contexto, el módulo LoRa E220 se posiciona como una solución ideal para aplicaciones que requieren comunicación inalámbrica a larga distancia, eficiencia energética y alta fiabilidad. Su capacidad para operar en bandas de frecuencia libres de licencia, su resistencia a interferencias y su sistema de corrección de errores lo convierten en una tecnología robusta y adaptable a diversos entornos. Además, su costo competitivo lo hace especialmente atractivo para proyectos con recursos limitados, consolidándose como una opción superior frente a otras tecnologías inalámbricas como Bluetooth o Wi-Fi.

Finalmente, la agricultura de precisión, impulsada por la adopción de tecnologías emergentes como el IoT, no solo tiene el potencial de incrementar la productividad agrícola, sino que también representa un avance significativo hacia la sostenibilidad ambiental. Estas tecnologías permiten mitigar los efectos adversos del cambio climático, optimizar el uso de recursos naturales y mejorar la capacidad de adaptación ante las condiciones cambiantes del entorno agrícola.

Ahora en cuanto a la tecnología de comunicación inalámbrica es un aspecto crucial en el diseño de sistemas IoT para la agricultura inteligente. Entre las opciones disponibles, el módulo LoRa E220 destaca por sus numerosas ventajas técnicas y operativas, lo que lo convierte en una solución preferida para este tipo de aplicaciones. Una de sus principales fortalezas es su capacidad para transmitir datos a largas distancias, alcanzando hasta 5 km en áreas rurales abiertas, una cobertura que supera ampliamente tecnologías como Bluetooth y Wi-Fi. Además, el LoRa E220 está diseñado para ser altamente eficiente en el consumo de energía, una característica clave en entornos donde la disponibilidad de energía es limitada.

Otra ventaja significativa del LoRa E220 es su robustez frente a interferencias físicas y electromagnéticas, gracias a su modulación de espectro ensanchado por chirp (CSS), lo que garantiza una transmisión confiable incluso en entornos con obstáculos. Este módulo también ofrece flexibilidad y adaptabilidad, operando en bandas de frecuencia libres de licencia, como 433 MHz o 915 MHz, lo que reduce los costos operativos y facilita su implementación en diversas regiones

del mundo.

La fiabilidad en la transmisión de datos es otra de sus fortalezas, gracias a un sistema de corrección de errores mediante CRC-16, que asegura la integridad de los datos transmitidos. Finalmente, el costo competitivo del LoRa E220, combinado con sus avanzadas capacidades técnicas, lo convierte en una opción ideal para proyectos con presupuestos limitados.

Estas características hacen que el LoRa E220 no solo sea una herramienta esencial para la agricultura inteligente, sino también una tecnología versátil y rentable para cualquier aplicación IoT que requiera comunicación inalámbrica confiable y eficiente.

En palabassimpres, el **LoRa E220** sobresale como una tecnología ideal para aplicaciones IoT en entornos rurales y remotos. Su alcance superior, bajo consumo energético, resiliencia ante interferencias y fiabilidad en la transmisión de datos lo posicionan como una alternativa altamente competitiva frente a otras tecnologías inalámbricas como Bluetooth o Wi-Fi.

Capítulo 4

Desarrollo del Sistema SIGASI

En el marco de la implementación de sistemas tecnológicos avanzados, este proyecto se enfocó en diseñar y desarrollar una solución integral que combina hardware y software para satisfacer necesidades específicas. El objetivo principal fue garantizar la precisión en la recolección de datos, optimizar la transmisión de información y ofrecer herramientas prácticas para la gestión de los mismos.

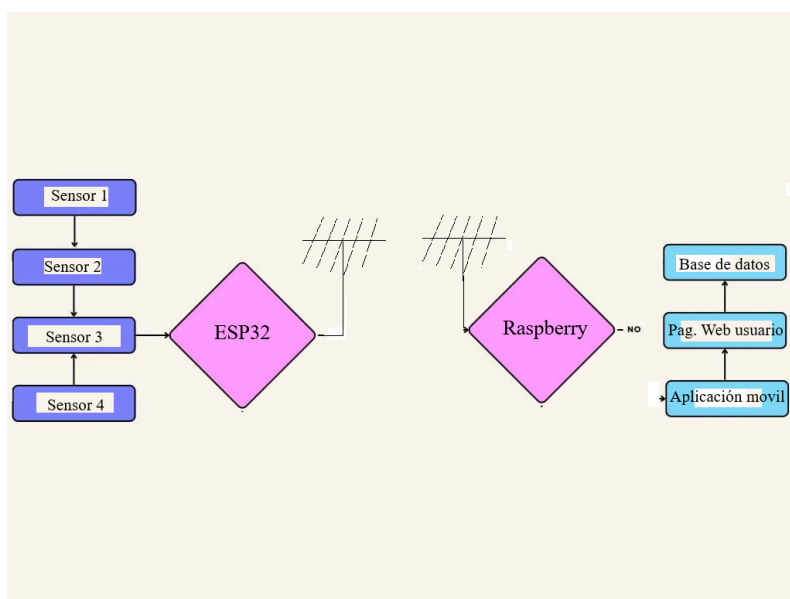


Figura 4.1: Diagrama a bloques del sistema SIGASI.

A continuación, se describe el proceso llevado a cabo, desde la selección de componentes fundamentales, como sensores y microcontroladores, hasta el desarrollo de interfaces y aplicaciones que facilitan la interacción con el sistema, llevando a cabo en diversas fases, las cuales se describirán a continuación:

1. Identificación y elección de los sensores adecuados para las aplicaciones específicas del

proyecto.

2. Evaluación y selección de las placas de desarrollo más adecuadas para la implementación del sistema.
3. Creación del diseño del sistema y desarrollo del software necesario para su funcionamiento.
4. Realización de pruebas para asegurar la precisión de los sensores, así como la correcta transmisión y recepción de datos.
5. Diseño y desarrollo de una base de datos para el almacenamiento y gestión de la información recolectada.
6. Creación de una interfaz web para la visualización y gestión de los datos recopilados.
7. Desarrollo de una aplicación móvil o de escritorio que permita la interacción con el sistema.

4.1. Selección de Tarjetas de Desarrollo

Durante la investigación, se buscaron varios microcontroladores que pudieran servir en este proyecto, entre los cuales se utilizaron: Arduino Uno, Arduino Nano, Arduino Mega con ESP8266 integrado, ESP8266, ESP32 y Raspberry Pi Pico. Cada una de estas tarjetas de desarrollo se probó con cada sensor individualmente, y posteriormente con el módulo LoRa® para evaluar su funcionamiento y desempeño al recibir y enviar datos.

Datos de Comparación

Cuadro 4.1: Datos Básicos de Microcontroladores

Microcontrolador	Pines Analógicos	Soporte para LoRa
Arduino Uno	6	Requiere shield adicional
Arduino Nano	8	Requiere shield adicional
Arduino Mega con ESP8266	16	Directo
ESP8266	1	Directo
ESP32	18	Directo
Raspberry Pi Pico	3	Directo

Cuadro 4.2: Desventajas de Microcontroladores

Microcontrolador	Desventajas
Arduino Uno	Memoria limitada, menos potente que otras opciones, sin soporte nativo para WiFi.
Arduino Nano	Limitaciones similares al Uno en potencia y memoria, sin WiFi.
Arduino Mega con ESP8266	Consumo de energía más alto que otras opciones más eficientes.
ESP8266	Menos pines analógicos y menos potente que el ESP32.
ESP32	Consumo de energía relativamente alto frente a opciones más simples como el Uno o Nano.
Raspberry Pi Pico	Aunque tiene capacidad ADC, necesita expansiones para WiFi y otros periféricos.

Después de analizar y realizar pruebas con cada tarjeta de desarrollo, el ESP32 y la Raspberry Pi Pico sobresalieron en el uso de pines analógicos, siendo mejores opciones debido a su resolución y precisión. Ambos microcontroladores ofrecen una resolución de conversión ADC de 12 bits, mientras que el Arduino tiene 10 bits. Esto significa que, al ser de 12 bits, pueden transformar una señal analógica en 4096 niveles distintos, proporcionando una mayor exactitud en la lectura de sensores.

En cuanto a versatilidad y capacidad de procesamiento, tanto el Arduino Mega como el ESP32 y la Raspberry Pi Pico son más potentes en términos de capacidad de procesamiento y memoria, en comparación con opciones más básicas como el Arduino Uno, Nano o el ESP8266. Esto los convierte en adecuados para proyectos más complejos que requieren manejar múltiples entradas analógicas y ejecutar tareas más exigentes.

4.2. Selección, diseño y programación de sensores

Parte de la investigación que se hizo fue saber qué tipos de sensores se implementan y cómo se utilizan en la agricultura. Si bien existen varios sensores en el mundo, se optó por usar los que fueran mejores y fácil de implementar.

4.2.1. Humedad

La humedad juega un papel crucial en la salud y el desarrollo de las plantas, tanto en exceso como en déficit. Una cantidad excesiva de humedad puede llevar a condiciones de encharcamiento en el suelo, limitando el suministro de oxígeno a las raíces y provocando su pudrición. Además, fomenta el crecimiento de hongos patógenos que dañan las raíces y partes aéreas de la planta. Con el tiempo, esto puede resultar en el marchitamiento y eventual muerte de la planta debido al estrés hídrico.

Por otro lado, la falta de humedad también causa problemas significativos. Las plantas pueden experimentar estrés hídrico, lo que afecta su capacidad para absorber nutrientes y realizar la fotosíntesis. La escasez de humedad puede marchitar las hojas y detener el crecimiento de la planta. En casos extremos, puede llevar a la muerte de la planta por desecación.

A pesar de los problemas que puede causar, la humedad tiene beneficios importantes para las plantas. Facilita el transporte de nutrientes, es esencial para la fotosíntesis, proporciona soporte estructural, ayuda a regular la temperatura y mejora la resistencia al estrés hídrico. Sin embargo, también presenta efectos negativos como propiciar enfermedades fúngicas, causar asfixia radicular, conducir a la podredumbre de la raíz y generar un desarrollo deficiente de la misma. Por lo tanto, es fundamental mantener un equilibrio adecuado de humedad para asegurar el crecimiento y salud óptimos de las plantas, estos datos se tomaron en cuenta para la selección de sensores. [25]

En el amplio ámbito de la intersección entre la agricultura y la electrónica, emergen dispositivos que actúan como interfaces entre los procesos naturales y las aplicaciones tecnológicas. Algunos de estos dispositivos son: FC-28 y SEN-HS-CA. Estos sensores son una innovación compacta pero altamente funcional que facilita la exploración de los fundamentos de la fertilidad del suelo, ya que la humedad representa un parámetro crítico para el desarrollo y el bienestar de las plantas, cuyos requerimientos pueden variar significativamente según la especie, el entorno y otras variables específicas. La capacidad de monitorear la humedad del suelo proporcionada por los sensores FC-28 y SEN-HS-CA abre la puerta a un entendimiento más profundo de los efectos de la humedad en las plantas.

4.2.2. Sensor FC-28

El sensor FC-28 consta de un par de electrodos en forma de “U” chapado en níquel y una tarjeta comparadora basada en un LM393, este sensor sirve para conocer la humedad en el ambiente, como por ejemplo; en la tierra, jardines, macetas y/o invernaderos pues al detectar humedad en el suelo se creará un puente entre una punta y otra, que será detectado por un circuito de control con un amplificador operacional que será el encargado de transformar la conductividad registrada a un valor analógico que podrá ser leído por un microcontrolador [3]

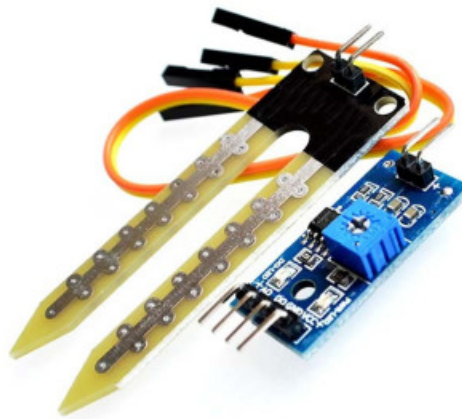


Figura 4.2: Hidrometro Fc-28

El sensor FC-28 utiliza un principio básico de la física eléctrica para determinar la humedad del suelo: la relación entre la resistencia eléctrica y la humedad del medio. Cuando el suelo está seco, su resistencia eléctrica es mayor, y cuando está húmedo, la resistencia disminuye. Esta relación se basa en la conductividad eléctrica del suelo, que aumenta con el contenido de humedad.

Para interpretar esta relación y convertirla en una medida de humedad del suelo, el sensor FC-28 utiliza un módulo de control que generalmente contiene un circuito integrado amplificador y un convertidor analógico a digital (ADC). La señal eléctrica proveniente del sensor, que varía con la humedad del suelo, se amplifica y luego se convierte en una señal digital que puede ser interpretada por un microcontrolador o una computadora. [15]

4.2.3. Sensor SEN-HS-CAP

El Sensor de humedad de suelo capacitivo permite medir la humedad en el suelo utilizando el principio de capacitancia entre electrodos en lugar de resistencia, lo que aumenta considerablemente la vida útil del sensor. El electrodo posee una capa de protección anticorrosión para una mayor duración (no se oxida como el sensor resistivo). El funcionamiento del sensor se basa en medir la capacitancia entre 2 electrodos insertados dentro del suelo, la capacitancia entre los electrodos dependerá de la humedad del suelo, por lo que para un suelo muy húmedo (0V) tendremos una capacitancia muy baja y para un suelo muy seco (5V) la capacitancia será muy alta. [2]

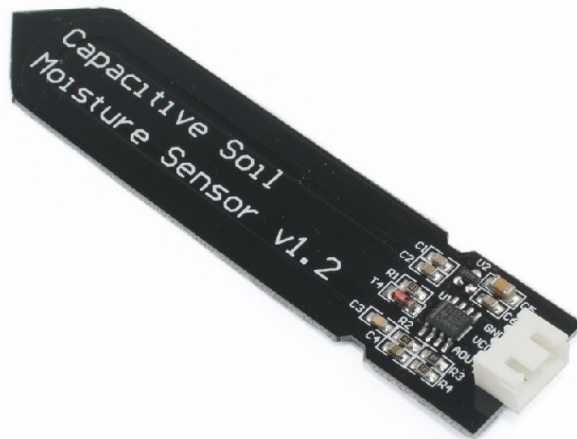


Figura 4.3: Sensor de Humedad Suelo Capacitivo Anticorrosivo.

FC-28 vs SEN-HS-CAP

Ventajas: FC-28

- Costo. Es generalmente más barato y ampliamente disponible.
- Facilidad de uso: Simple de conectar y usar con microcontroladores como Arduino.
- Compatibilidad: Compatible con una amplia gama de proyectos de hobby y educativos.
- Salidas analógica y digital: Proporciona ambas opciones para diferentes tipos de medición.

Desventajas:

- Corrosión: Los sensores de tipo resistivo, como el FC-28, tienden a corroerse con el tiempo debido al contacto constante con la humedad, lo que reduce su vida útil.
- Precisión: Puede ser menos preciso debido a la interferencia de la corrosión y variaciones en el suelo.
- Interferencia eléctrica: Puede ser susceptible a interferencias eléctricas del entorno, afectando la precisión de las lecturas.

SEN-HS-CAP

Ventajas:

- No corrosivo: Al ser capacitivo, no sufre de corrosión, lo que aumenta su durabilidad y confiabilidad a largo plazo.
- Precisión: Tiende a ser más preciso porque no se ve afectado por la conductividad del suelo o la corrosión.

- Menor interferencia: Menos susceptible a interferencias eléctricas y ruido, proporcionando lecturas más estables.
- Calibración: Suele necesitar menos calibración y mantenimiento en comparación con los sensores resistivos.
- Desventajas:
- Costo: Generalmente más caro que los sensores resistivos como el FC-28.
- Compatibilidad: Puede requerir un poco más de conocimiento técnico para saber como es el funcionamiento.

El sensor SEN-HS-CAP destaca por su durabilidad, precisión y estabilidad, lo que lo convierte en una opción superior para diversas aplicaciones. A diferencia de otros sensores, la ausencia de partes metálicas en contacto con el suelo elimina el problema de la corrosión, garantizando una vida útil más larga. Además, su principio de funcionamiento capacitivo ofrece lecturas más precisas y estables, fundamentales en situaciones donde la exactitud es crucial. Otro beneficio del SEN-HS-CAP es su reducido mantenimiento. Su diseño no corrosivo y su menor susceptibilidad a interferencias minimizan la necesidad de mantenimiento regular. Aunque el sensor FC-28 es más económico y fácil de usar, el SEN-HS-CAP es generalmente una mejor opción para aplicaciones a largo plazo debido a su mayor durabilidad, precisión y estabilidad.

4.2.4. Temperatura

La temperatura es uno de los factores más críticos que influyen en el crecimiento y desarrollo de las plantas. Cada especie vegetal tiene un rango de temperatura óptimo en el cual puede prosperar, y desviaciones significativas de este rango pueden tener efectos adversos.

En general, las temperaturas adecuadas permiten una fotosíntesis eficiente, lo cual es crucial para la producción de energía en las plantas. Cuando las temperaturas son demasiado bajas, la actividad enzimática se reduce, ralentizando la fotosíntesis y, en consecuencia, el crecimiento de la planta. En temperaturas extremadamente frías, puede ocurrir daño por congelación, lo que puede matar células vegetales y dañar tejidos.

Por otro lado, temperaturas excesivamente altas pueden causar estrés ambiental, lo que causa una fuerte restricción para el aumento de la productividad y de la extensión de los cultivos. Se estima que únicamente un 10 % de la superficie de la tierra arable se encuentra libre de estrés. Cerca del 20 % de la tierra presenta algún tipo de deficiencia o toxicidad mineral, 26 % es afectada por estrés de sequía y 15 % por congelación. Incluso bajo condiciones de producción protegida como invernadero y túneles se presentan eventos de estrés biótico o abiótico que disminuyen la productividad y calidad de los cultivos. [26]

4.2.5. LM35

El LM35 es un pequeño encapsulado metálico de tres pines, similar a un transistor, lo que facilita su integración en proyectos electrónicos. No requiere de componentes adicionales para su calibración o amplificación de señal, lo que lo convierte en una opción ideal para mediciones de temperatura simples y precisas. La salida de voltaje del LM35 cambia linealmente con la temperatura, con una escala de 10 mV/°C. Por lo tanto, por cada grado Celsius de cambio en la temperatura ambiente, la salida de voltaje del LM35 cambia en 10 milivoltios.

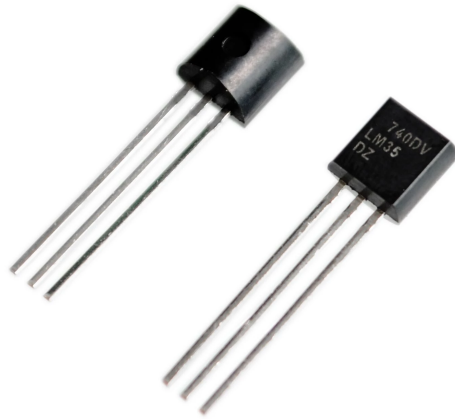


Figura 4.4: Sensor de temperatura LM35.

$$10 \text{ mV} = 1^{\circ}\text{C} \quad (4.1)$$

Debido a su versatilidad y facilidad de uso, el LM35 se encuentra en una amplia gama de aplicaciones, que incluyen sistemas de control de temperatura, monitoreo ambiental, dispositivos médicos, sistemas de climatización, y muchos otros proyectos electrónicos donde se requiere medir la temperatura de manera precisa y confiable.

4.2.6. Solar

Entre los elementos climáticos que afectan el crecimiento de las plantas está la luz del sol. Es un componente esencial para que las plantas puedan realizar la fotosíntesis, proceso del cual dependen directamente para transformar sustancias minerales en materia orgánica. La eficiencia de la fotosíntesis mejora con el aumento de la luz, la temperatura y el CO₂.

A mayor iluminación, más eficiente es la fotosíntesis y más rápido puede desarrollarse la planta. Sin embargo, a medida que la intensidad de la luz aumenta, la velocidad de la fotosíntesis llega a un límite conocido como “punto de saturación de la luz”, donde la velocidad del proceso fotosintético se estabiliza. La cantidad de luz que una planta recibe depende principalmente de dos factores: la intensidad de la luz y el número de horas diarias de exposición.

Los cambios en la iluminación que las plantas experimentan se denominan fotoperiodo, y la duración de la exposición o las horas de luz necesarias para que un cultivo se desarrolle y florezca varía según la especie. Sin embargo, tanto el exceso como la escasez de luz pueden causar efectos negativos en las plantas. Las plantas que reciben poca luz, muestran un crecimiento vegetativo reducido, menor floración, debilitamiento y hojas más pequeñas.

Por otro lado, las plantas expuestas a demasiada luz pueden desarrollar clorosis, una condición en la que las hojas se vuelven amarillas debido a la falta de clorofila. Por lo tanto, para lograr un desarrollo óptimo del cultivo y una mayor productividad, es crucial garantizar una iluminación adecuada. [38]

Para la creación del sensor solar, se utilizó un circuito promediador inversor, diseñado para medir y generar señales invertidas en electrónica y telecomunicaciones. Este circuito emplea un amplificador operacional (op-amp) en configuración inversora, utilizando el LM741. Las señales de entrada se aplican a través de una resistencia (R_n) en la entrada inversora (terminal 3) del op-amp. La salida del op-amp se obtiene del terminal 6 y se conecta al terminal inversor de un segundo op-amp. Además, un arreglo de fotorresistencias se conecta a la entrada inversora del amplificador LM741.

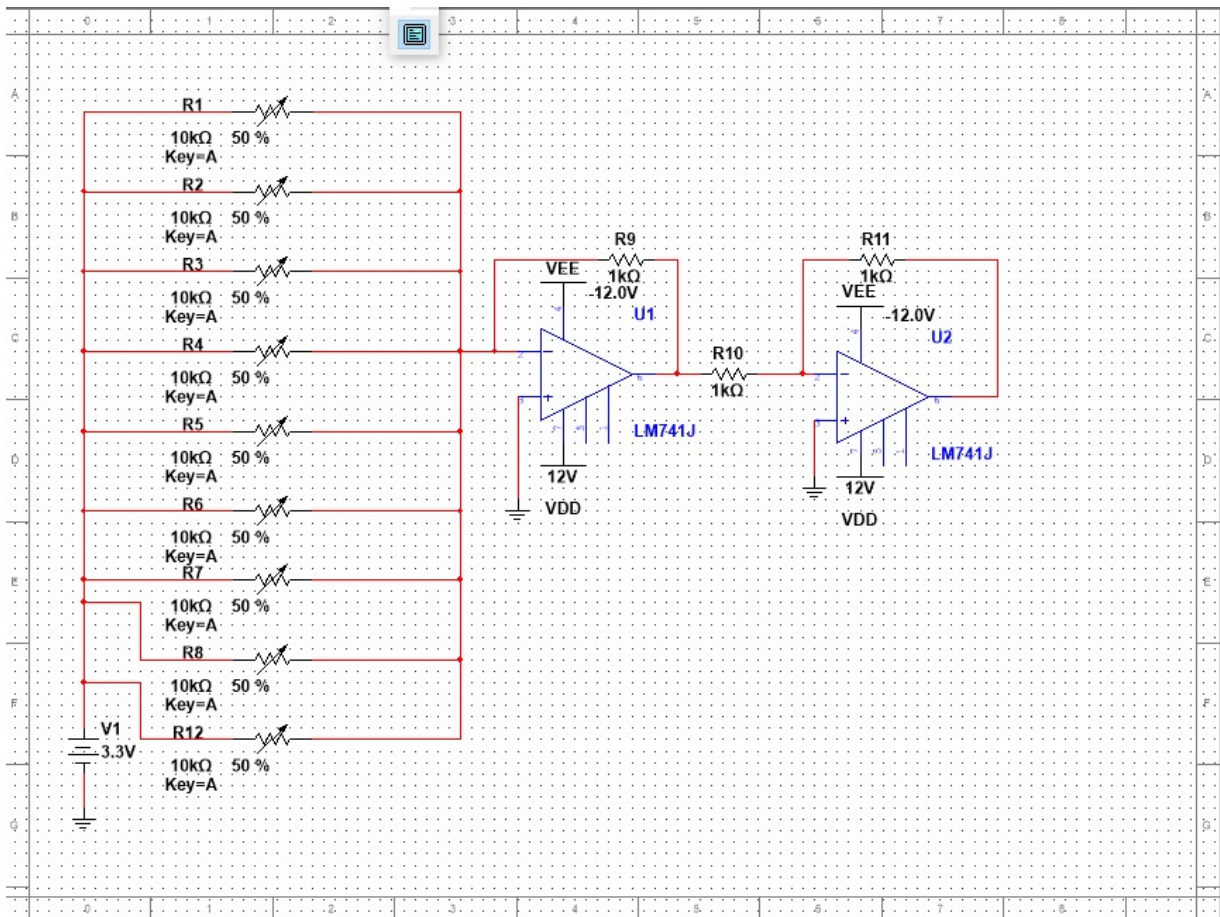


Figura 4.5: Circuito acondicionador para la adquisición de Luminosidad

$$\omega_0 = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (4.2)$$

$$Q = \frac{\sqrt{R_1 R_2 C_1 C_2}}{C_2(R_1 + R_2)} \quad (4.3)$$

Donde:

- R_1, R_2 son los valores de las resistencias.
- C_1, C_2 son los valores de los capacitores.
- ω_0 es el valor de la frecuencia de corte.
- Q es el factor que determina la altura y el ancho del punto más alto de la respuesta en frecuencia del filtro.

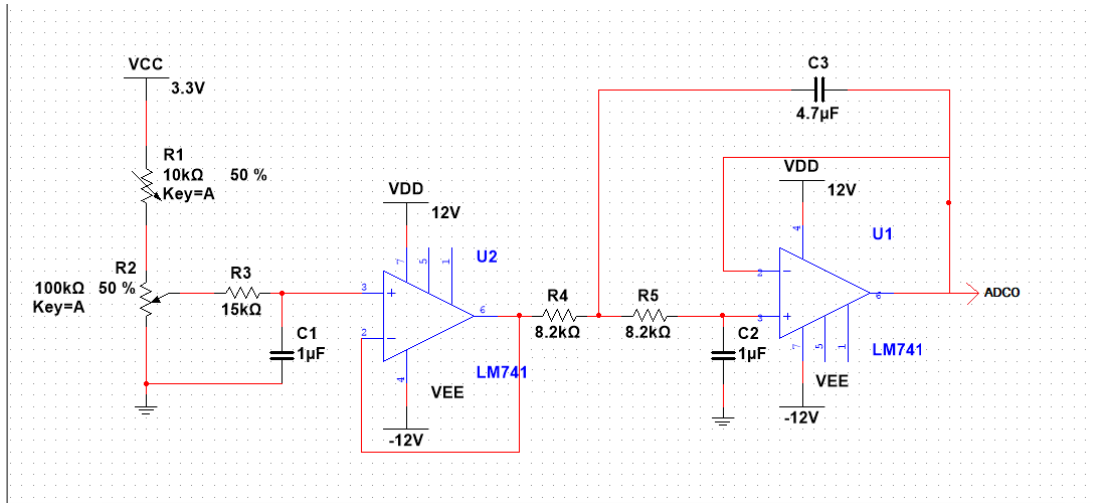


Figura 4.6: circuito sumador elaborado con amplificadores operacionales.

Donde:

- V_o es la salida.
- $V_1, V_2, \dots, V_n$ son los voltajes de entrada.
- n es el número de voltajes de entrada.

Para calcular R_f , se utiliza la siguiente fórmula:

$$R_f = \frac{R_i}{n} \quad (4.4)$$

Donde $R_i = R_1 = R_2 = R_3 = \dots = R_n$ y $R_n \geq R_f$. [4]

Sensor pH

El monitoreo del pH del suelo es un factor clave para el crecimiento y desarrollo saludable de los cultivos en un invernadero, ya que un pH fuera del rango óptimo puede afectar la disponibilidad de nutrientes y causar problemas en la producción agrícola. En agricultura, el pH del suelo es un factor crucial que afecta la disponibilidad de nutrientes para las plantas. Las plantas tienen diferentes requerimientos de pH según el tipo de suelo en el que crecen mejor. Algunas plantas prefieren suelos ácidos, mientras que otras prefieren suelos alcalinos

pH de varias sustancias comunes

Sustancia	pH	Sustancia	pH
Agua pura	7	Agua de lluvia	5.5-5.7
Agua de mar	8.0-8.3	Agua de lluvia ácida	3.0-5.0
Mayoría de suelos ácidos	4.0-6.0	Leche fresca	6.3-6.6
Jugo de naranja	3.4-4.0	Sangre	7.2-7.4
Jugo de limón	2.2-4.0	Solución jabonosa	8.5-10.0
Café	5.0-5.1	Soda cáustica	12.0-12.5
Coca-Cola	2.5-2.8	Vinagre	4.0-4.5

Cuadro 4.3: Valores de pH de diferentes sustancias[21]

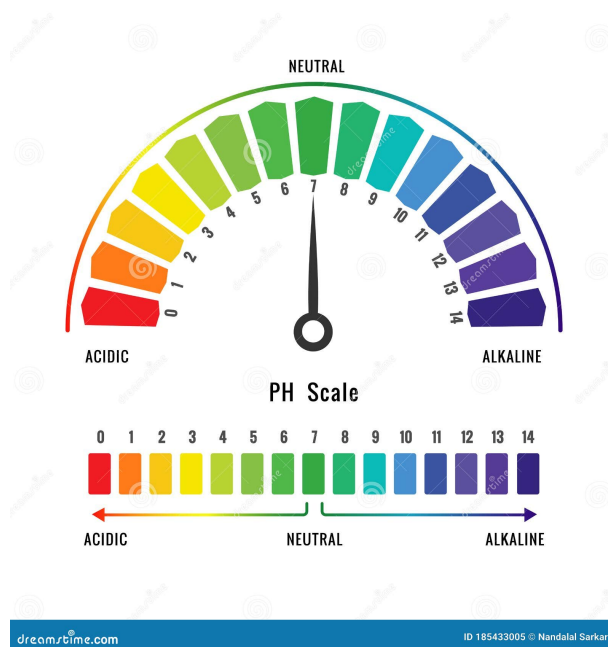


Figura 4.7: Escala vectorial de PH con código de colores. Ilustración vectorial de la escala de PH para medir la acidez o alquilidad de la solución. Indicador universal para medir el ph de la solución [36]

Influencia del pH en la agricultura

El pH del suelo influye en varios aspectos importantes para el crecimiento de las plantas:

- **Disponibilidad de nutrientes:** El pH del suelo afecta la disponibilidad de nutrientes esenciales como el fósforo, el potasio y el calcio.
- **Toxicidad de los elementos:** En suelos con pH extremadamente bajo o alto, ciertos elementos como el aluminio y el manganeso pueden volverse tóxicos para las plantas.
- **Actividad microbiana:** El pH influye en la actividad microbiana, que a su vez afecta la descomposición de la materia orgánica y la disponibilidad de nutrientes.
- **Estructura del suelo:** El pH puede afectar la capacidad de retención de agua, la aireación y la erosión del suelo.

Ajuste del pH en la agricultura

Los agricultores suelen realizar pruebas de pH del suelo y ajustar el pH según sea necesario mediante la aplicación de enmiendas como cal agrícola para aumentar el pH en suelos ácidos, o azufre para reducir el pH en suelos alcalinos. Esto optimiza las condiciones de crecimiento de los cultivos.

Relación entre el pH y los iones de hidrógeno

La relación entre la concentración de iones de hidrógeno y el pH se describe mediante la siguiente ecuación:

$$\text{pH} = -\log_{10} [\text{H}^+] \quad (4.5)$$

Donde:

- pH es el valor del pH de la solución.
- $[\text{H}^+]$ es la concentración de iones de hidrógeno en moles por litro (molaridad).

Esta ecuación muestra que el pH es el logaritmo negativo en base 10 de la concentración de iones de hidrógeno en la solución. El logaritmo negativo se utiliza para convertir la escala lineal de concentración de iones de hidrógeno en una escala logarítmica de pH, donde cada unidad de cambio en el pH representa un cambio de 10 veces en la concentración de iones de hidrógeno. [23]

Para la parte de la medición del PH se optó por usar el sensor PH-4502C. Que es un dispositivo que permite medir el PH con ayuda de una sonda que es la que toma la lectura (electrodo E201) mediante el conector BCN. El PH-4502C es útil para medir el pH del agua, jardín hidropónico, acuario y laboratorio. Para nuestro caso se usó para medir los niveles de pH que tiene el agua, en ciertos tipos de cultivos que requieren distintos tipos de nivel de Ph, por ejemplo, los aguacates

prefieren suelos ligeramente ácidos a neutros, con un pH óptimo generalmente entre 6.0 y 7.0. Un pH del suelo incorrecto puede afectar negativamente el crecimiento y la salud de los aguacates. Los jitomates prefieren suelos ligeramente ácidos con un pH en el rango de 6.0 a 7.0. Un pH del suelo incorrecto puede afectar la disponibilidad de nutrientes clave para el crecimiento de los jitomates, como el fósforo, el potasio y el calcio. En suelos muy alcalinos, la disponibilidad de ciertos nutrientes puede reducirse, lo que podría afectar el rendimiento y la calidad de los jitomates. [9]

PH-4502C

El sensor pH-4502C es un tipo de electrodo utilizado para medir el nivel de acidez o alcalinidad de una solución. Si bien existen diversos métodos para medir el pH este sensor permite obtener una lectura con precisión y fiabilidad en la medición del pH de soluciones líquidas. Opera fundamentalmente gracias a un electrodo de vidrio especializado, que reacciona con los iones de hidrógeno H^+ en la solución, internamente este sensor contiene una solución de cloruro de potasio KCL, la cual al entrar en contacto con otra solución genera una diferencia de potencial, esta es obtenida por medio de un amplificador operacional el cual amplifica dicha diferencia y entrega un voltaje de salida. Cabe mencionar que el sensor también tiene etapas de filtrado de bajo ruido y un circuito para su calibración.

El sensor de pH **PH-4502C** incluye varios componentes electrónicos en su circuito interno que permiten convertir la señal del electrodo de pH en una señal analógica que puede ser leída por un microcontrolador.

1. Electrodo de pH

- **Bulbo de Vidrio:** Parte sensible del electrodo que reacciona con los iones de hidrógeno (H^+) en la solución.
- **Solución de referencia:** Dentro del bulbo de vidrio hay una solución de referencia con un pH conocido.

2. Amplificador Operacional (Op-Amp)

- **Amplificación de la señal:** La señal generada por el electrodo de pH es de muy bajo voltaje. El amplificador operacional se utiliza para amplificar esta señal.
- **Alta impedancia:** Los op-amps en el circuito presentan una alta impedancia de entrada, lo cual es crucial para no cargar el electrodo y alterar la medición.

3. Convertidor de Señal

- **Conversión a voltaje utilizable:** La señal amplificada se convierte en un voltaje proporcional al pH de la solución. Este voltaje es el que se entrega en la salida analógica (AO) del módulo.

4. Potenciómetro de Calibración

- **Ajuste fino:** El circuito incluye uno o más potenciómetros que permiten ajustar el offset y la ganancia de la señal para calibrar el sensor.
- **Compensación de desviaciones:** Este ajuste es necesario para alinear el sensor con las condiciones específicas y las soluciones de calibración utilizadas.

5. Filtro de Ruido

- **Capacitores y resistencias:** Se incluyen componentes pasivos como capacitores y resistencias para filtrar el ruido eléctrico y estabilizar la señal.
- **Eliminación de ruido de alta frecuencia:** Estos filtros ayudan a eliminar interferencias de alta frecuencia que podrían afectar la precisión de la medición.

6. Regulador de Voltaje

- **Estabilidad en la alimentación:** Algunos módulos incluyen un regulador de voltaje para garantizar que el circuito reciba una alimentación estable, mejorando así la precisión de la medición.

7. Salida Analógica (AO)

- **Señal proporcional al pH:** La salida analógica proporciona un voltaje que varía linealmente con el pH de la solución, el cual puede ser leído por un pin analógico de un microcontrolador, como un Arduino.

El sensor pH-4502C transforma una señal de bajo voltaje producida por el electrodo de pH en una señal analógica que se puede utilizar, gracias a un circuito que abarca amplificación, filtrado y calibración. Este enfoque permite medir con precisión el pH de una solución. A continuación, se presenta la etapa de filtrado en la adquisición del sensor pH-4502C, tal como lo sugiere el fabricante. [29]



Figura 4.8: Sensor de PH Liquido con electrodo E201-BNC

conversión del valor analógico a voltaje utilizando la fórmula.

Cálculo del pH utilizando el sensor PH-4502C SIMPLE.

Leer el valor analógico del sensor, que es un valor entre 0 y la resolución ADC (por defecto 4095).

El voltaje se calcula utilizando la siguiente fórmula:

$$\text{voltaje} = \left(\frac{\text{valor analógico}}{\text{resolución del ADC}} \right) \times \text{voltaje de operación}(5 \text{ V}) \quad (4.6)$$

Para calcular el pH, se utiliza la relación entre el voltaje y el pH:

$$\text{pH} = 7 + \frac{\text{voltaje} - \text{voltaje neutral}}{\text{voltaje por pH}} - \text{calibración} \quad (4.7)$$

Donde:

- **voltaje neutral** es el voltaje correspondiente a un pH de 7.
- **voltaje por pH** es el cambio en voltaje por unidad de cambio en pH, cuyo valor es 0,18 V/pH.
- **calibración** es un valor que permite ajustar las lecturas de pH para mayor precisión (por defecto es 14,8).

Cálculo del pH utilizando el sensor PH-4502C.

El voltaje se calcula utilizando la siguiente fórmula:

$$\text{voltaje} = \left(\frac{\text{valor analógico}}{\text{resolución del ADC}} \right) \times \text{voltaje de operación}(5 \text{ V}) \quad (4.8)$$

La fórmula completa para calcular el pH es:

$$\text{pH} = 7 + \frac{\left(\left(\frac{\text{valor analógico}}{\text{resolución del ADC}} \right) \times \text{voltaje de operación} - \text{voltaje neutral} \right)}{\text{voltaje por pH}} - \text{calibración} \quad (4.9)$$

4.3. Módulo LoRa E220

El módulo **LoRa E220** se destaca como una solución eficiente y confiable para la transmisión de datos en aplicaciones de (IoT), especialmente en entornos rurales o áreas remotas. Su objetivo principal es garantizar una comunicación eficaz a largas distancias con un consumo energético reducido. Estas características lo posicionan como una alternativa superior frente a otras tecnologías inalámbricas tradicionales, las cuales enfrentan limitaciones en cobertura y eficiencia.

4.3.1. Trasmision de Datos

Como se menciona anterior mente, el módulo *LoRa* E220, destaca por su capacidad de ser programado a través del protocolo *UART* (Transmisor-Receptor Asincrónico Universal). Este módulo emplea el IC LLCC68, lo que le brinda una flexibilidad notable para ser implementado en diversas unidades de procesamiento, como microcontroladores o plataformas de desarrollo, facilitando su integración en proyectos variados. Además, el E220 es conocido por su bajo costo, lo que lo convierte en una opción accesible para aplicaciones de IoT.

El módulo LoRa E220 opera en bandas de frecuencia de sub-gigahercios, lo que permite alcanzar distancias de comunicación significativas con un bajo consumo de energía. Esto lo hace ideal para aplicaciones en agricultura inteligente, monitoreo ambiental y sistemas de automatización, donde la comunicación a larga distancia es crucial. Además, el módulo incluye características de seguridad como encriptación, lo que ayuda a proteger los datos transmitidos. La combinación de estas características, junto con su fácil programación y costo económico, hace del E220 una solución atractiva para desarrolladores e investigadores en el ámbito de la tecnología de comunicación inalámbrica.



Figura 4.9: Módulo de comunicación E220 [19]

La transmisión de datos mediante módulos de radio es una práctica común en múltiples aplicaciones tecnológicas, abarcando desde el *IoT* hasta sistemas de comunicación industrial. La elección y configuración correctas de estos módulos son fundamentales para asegurar una comunicación efectiva y confiable. Cada módulo de radio funciona en una frecuencia específica, que debe ser seleccionada en función de las necesidades de la aplicación y las regulaciones locales. Por ejemplo, dependiendo de las normativas de cada país, ciertas bandas de frecuencia pueden estar reservadas para aplicaciones específicas, lo que implica que se debe tener en cuenta el entorno operativo y el tipo de datos que se desean transmitir.

Adicionalmente, factores como el rango de transmisión, el consumo energético y la interferencia con otros dispositivos son cruciales para la selección del módulo *LoRa*® adecuado. Al operar en frecuencias más bajas, los módulos RF pueden penetrar mejor en obstáculos y alcanzar mayores distancias, lo que los hace adecuados para aplicaciones en entornos rurales o industriales donde la comunicación a larga distancia es esencial. [37] Algunos módulos están diseñados para operar en frecuencias de sub-GHz, lo que permite largas distancias de comunicación con bajo consumo de energía, ideal para aplicaciones de *IoT* que requieren sensores distribuidos en áreas amplias. La comprensión de estos aspectos garantiza que la transmisión de datos sea no solo efectiva, sino también sostenible y alineada con los requisitos técnicos y normativos del proyecto [33].

4.4. ESP-32

El ESP32 es un microcontrolador moderno y versátil, puesto que está diseñado para aplicaciones que requieren conectividad inalámbrica y alto rendimiento. En efecto, integra en un solo chip Wi-Fi, Bluetooth y un procesador potente, además de una amplia variedad de entradas y salidas, lo que facilita la conexión con sensores y actuadores. Por estas razones, se ha consolidado como una de las plataformas más utilizadas en automatización, domótica e Internet de las Cosas (IoT), ya que permite desarrollar soluciones eficientes, compactas y de bajo costo.

Por otra parte, al comparar distintas placas de desarrollo, se observó que el ESP32 y la Raspberry Pi Pico resultan superiores al Arduino en el manejo de entradas analógicas, debido a que incorporan una resolución ADC de 12 bits, capaz de convertir las señales en 4096 niveles, lo que ofrece mayor precisión en la lectura de sensores. Asimismo, tanto el ESP32 como la Raspberry Pi Pico y el Arduino Mega presentan mayor capacidad de procesamiento y memoria en comparación con modelos básicos como el Arduino Uno, Nano o el ESP8266. En consecuencia, estas características los hacen más adecuados para proyectos complejos que requieren gestionar múltiples entradas analógicas y procesar un mayor volumen de datos.

En la sección de procesamiento, al considerar la amplia variedad de dispositivos disponibles, se decidió utilizar la placa *ESP32-WROOM*. Estas placas están diseñadas específicamente para aplicaciones de *IoT* y tecnología *wearable*. Este *SoC* (System on a Chip) presenta varias características destacadas, entre las cuales se incluyen:

- **Conectividad Dual:** El *ESP32-WROOM* ofrece capacidades de conectividad Wi-Fi y Bluetooth (classic y BLE), lo que permite la comunicación con otros dispositivos y redes sin necesidad de hardware adicional.
- **Rendimiento Potente:** Con un procesador de doble núcleo que puede alcanzar velocidades de hasta 240 MHz, el ESP32 es capaz de manejar tareas complejas y realizar múltiples operaciones simultáneamente, lo que es ideal para aplicaciones que requieren procesamiento intensivo.

- **Amplia Memoria:** El módulo incluye hasta 520 KB de SRAM y soporte para almacenamiento externo, lo que permite la ejecución de aplicaciones más grandes y complejas, así como el almacenamiento de datos temporales.
- **Flexibilidad de Entrada/Salida:** El *ESP32-WROOM* cuenta con múltiples pines de GPIO que permiten la conexión de diversos sensores y actuadores, además de funcionalidades como PWM, ADC y DAC, facilitando la interacción con el entorno.
- **Bajo Consumo de Energía:** Diseñado para aplicaciones de larga duración, el *ESP32-WROOM* ofrece modos de bajo consumo, lo que es esencial para dispositivos alimentados por batería en el ámbito de *IoT* y tecnología wearable.
- **Integración de Seguridad:** Incluye funciones de seguridad como cifrado de hardware y arranque seguro, lo que ayuda a proteger la integridad y la confidencialidad de los datos transmitidos.

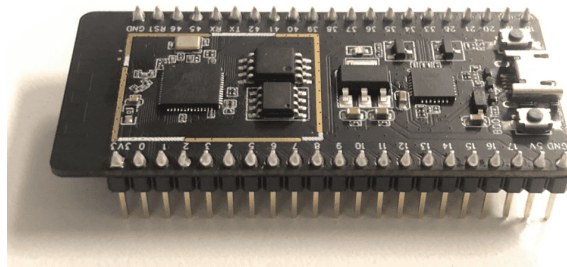


Figura 4.10: Unidad de Procesamiento ESP-32 [27]

4.5. Receptor y Sistema de Procesamiento

Para implementar una base de datos en un proyecto como SIGACI, la Raspberry Pi 4 se presenta como una opción sólida por varias razones clave. Su combinación de capacidad de procesamiento, conectividad y flexibilidad la convierte en una plataforma ideal para manejar bases de datos, tanto para aplicaciones pequeñas como para proyectos más avanzados.

A diferencia de los microcontroladores, la Raspberry Pi 4 ejecuta un sistema operativo completo, como Raspberry Pi OS, basado en Linux, lo que permite utilizar una amplia variedad de sistemas de gestión de bases de datos (DBMS) como MySQL, PostgreSQL, SQLite o MongoDB. Esto facilita la administración de la base de datos y proporciona un entorno conocido para los desarrolladores que trabajan con bases de datos en sistemas Linux.

En este proyecto se emplea una Raspberry Pi 4B como receptor, un sistema embebido que presenta varias características fundamentales:

- **Procesador Potente:** La Raspberry Pi 4B está equipada con un procesador Quad-Core Cortex-A72 a 1.5 GHz, lo que proporciona un rendimiento mejorado en comparación con versiones anteriores, permitiendo ejecutar aplicaciones complejas y múltiples procesos simultáneamente.
- **Memoria RAM:** Está disponible con diferentes configuraciones de RAM, incluyendo 2 GB, 4 GB y 8 GB de LPDDR4, lo que permite manejar tareas exigentes y ejecutar múltiples aplicaciones de manera eficiente.
- **Conectividad:** La Raspberry Pi 4B ofrece conectividad versátil con puertos USB 3.0 y USB 2.0, así como puertos HDMI duales para salida de video, y un puerto Ethernet Gigabit que permite conexiones de red rápidas. También cuenta con conectividad Wi-Fi y Bluetooth, facilitando la integración en redes inalámbricas.
- **Puertos GPIO:** Incluye un conector de 40 pines GPIO que permite la conexión de sensores, actuadores y otros dispositivos, lo que la hace adecuada para proyectos de hardware personalizados y aplicaciones de IoT.
- **Almacenamiento:** Utiliza una tarjeta microSD para el sistema operativo y el almacenamiento de datos, y ofrece la posibilidad de utilizar almacenamiento externo a través de USB, lo que amplía las opciones de capacidad.
- **Compatibilidad de Software:** Soporta una variedad de sistemas operativos, incluyendo Raspbian (ahora Raspberry Pi OS), Ubuntu y otros, lo que proporciona flexibilidad en la elección del entorno de desarrollo.
- **Soporte para Multimedia:** Capaz de reproducir video en 4K, lo que la convierte en una opción ideal para aplicaciones que requieren procesamiento multimedia, como proyectos de video y visualización de datos.
- **Comunidad Activa:** La Raspberry Pi cuenta con una extensa comunidad de usuarios y desarrolladores, lo que facilita el acceso a recursos, tutoriales y soporte técnico, ideal para aquellos que inician proyectos de hardware.

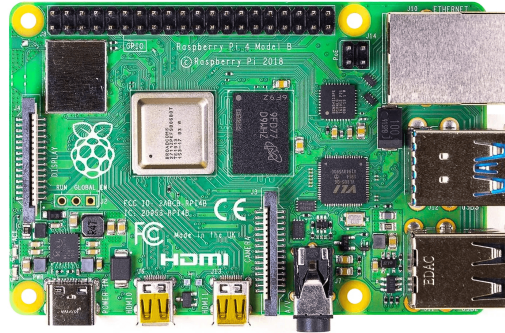


Figura 4.11: Raspberry Pi 4B [24]

Este sistema permite la utilización de varios sistemas operativos basados en el kernel de Linux. Para este trabajo, se optó por la distribución Debian para arquitectura ARM64. Además, como receptor de comunicación *LoRa*®, se empleó un módulo E220, que facilita la recepción de datos transmitidos de manera inalámbrica, aprovechando las ventajas de largo alcance y bajo consumo energético que ofrece esta tecnología. Esto garantiza una comunicación eficiente entre los dispositivos, ideal para aplicaciones de *IoT* y monitoreo remoto.

Conclusiones del capítulo

El desarrollo y la implementación del sistema de monitoreo agrícola propuesto en este capítulo destacan la importancia de la selección adecuada de componentes y tecnologías para garantizar un funcionamiento eficiente. El diseño del sistema, que integra sensores especializados, un nodo Gateway y un servidor Raspberry Pi 4, permite la captura, transmisión y almacenamiento de datos clave para la gestión de cultivos.

La selección de las tarjetas de desarrollo y los sensores fue un paso crítico, ya que cada componente debía cumplir con requisitos específicos de bajo consumo energético, resistencia a condiciones adversas y precisión en la medición de los parámetros físicos y químicos del suelo. Además, la integración de la tecnología *LoRa*® aseguró que la transmisión de los datos recolectados pudiera realizarse a largas distancias, incluso en áreas rurales con poca infraestructura de telecomunicaciones.

En cuanto a la programación y configuración del sistema, se logró un entorno robusto para la transmisión de datos, utilizando software optimizado para el procesamiento en tiempo real. La implementación del sistema de alertas y la interfaz web permiten a los usuarios monitorear las condiciones del cultivo de forma remota y recibir notificaciones cuando se detecten condiciones fuera de los rangos óptimos.

Este capítulo ha demostrado que la correcta implementación del hardware y software es fundamental para que el sistema funcione como una herramienta eficaz en la toma de decisiones agrícolas, optimizando los recursos naturales y mejorando la productividad del cultivo. Las pruebas realizadas confirmaron que el sistema es capaz de operar, lo que valida su aplicabilidad en entornos reales.

Capítulo 5

Diseño y caracterización

5.1. Sensores

Este capítulo describe la implementación y verificación de cada módulo del sistema de monitoreo agrícola, cuyo objetivo es medir diversas variables ambientales que afectan el rendimiento de los cultivos. Se llevaron a cabo pruebas experimentales para analizar el desempeño de los sensores y su precisión en la recolección de datos. A continuación, se presentan los dispositivos empleados en las pruebas:

La programación de los sensores y la configuración del sistema de monitoreo se llevaron a cabo utilizando el entorno de desarrollo **Arduino IDE**. Este entorno fue seleccionado debido a su compatibilidad con los microcontroladores utilizados y su facilidad para integrar los diferentes módulos y sensores implementados en el proyecto.

- **SEN-HS-CAP (Sensor de Humedad y Temperatura):** Sensor capacitivo que proporciona mediciones precisas de la humedad relativa del aire y la temperatura ambiental, fundamentales para el monitoreo de las condiciones climáticas.
- **Sensor Solar:** Empleado para medir la radiación solar incidente, un indicador importante para evaluar la eficiencia fotosintética de los cultivos y el potencial energético.
- **LM35 (Sensor de Temperatura):** Sensor analógico que ofrece lecturas precisas de la temperatura en diversas ubicaciones del sistema agrícola.
- **PH4502C (Sensor de pH):** Utilizado para medir el nivel de pH en el suelo o en el agua de riego, proporcionando información sobre la acidez o alcalinidad del entorno de cultivo.

5.1.1. Pruebas con sensor de humedad capacitivo

El SEN-HS-CAP es un sensor capacitivo de humedad de suelo que mide la humedad en función de la capacidad del suelo. A continuación, se explica cómo se calibro y utilizo el sensor.

El SEN-HS-CAP proporciona una salida analógica que varía en función de la humedad del suelo. La salida analógica está generalmente en el rango de 0 a 3.3V, aunque puede variar según el modelo específico.

Materiales utilizados.

- SEN-HS-CAP.
- ESP-32.
- Cables de conexión
- Suelo seco
- Suelo húmedo
- Recipientes o macetas para pruebas

Conexiones básicas:

- **VCC:** Conexión con el pin 3.3V del ESP-32.
- **GND:** GND del ESP-32
- **OUT:** Conexión al GPIO del ESP-32 (por ejemplo, GPIO 32).

Procedimiento de calibración

1. **Conexión del sensor:** Se conecta el sensor al ESP-32 siguiendo las instrucciones de conexión anteriores.
2. **Lectura en suelo seco:** Se insertó el sensor de humedad en un recipiente con suelo completamente seco.
 - Se observa el valor mostrado en el monitor serial mostrado como valorSeco.

Lectura en suelo húmedo:

- Se limpió el sensor y se aseguró de que estuviera completamente seco antes de continuar.
- Se inserta el sensor en un recipiente con suelo húmedo. El suelo debe estar saturado, pero no inundado.
- Se mostrará en el monitor serial el valor valorHumedo.

Calibración del sensor: Una vez obtenidos los valores de `valorSeco` y `valorHumedo`, se utiliza la siguiente función para obtener el porcentaje de humedad del suelo:

$$\text{humedad} = \text{map}(\text{sensorValue}, \text{valorSeco}, \text{valorHumedo}, 0, 100) \quad (5.1)$$

donde:

- `sensorValue`: Es el valor leído por el sensor.
- `valorSeco`: Es el valor leído cuando el sensor se encuentra en suelo completamente seco.
- `valorHumedo`: Es el valor leído cuando el sensor se encuentra en suelo húmedo.
- El resultado de la fórmula es el porcentaje de humedad del suelo, que se ajusta entre 0 (completamente seco) y 100 (completamente húmedo).

Consejos de calibración y uso

- **Limpiar el sensor:** Después de cada prueba, limpia el sensor para eliminar cualquier residuo de suelo que pueda afectar las lecturas futuras.
- **Evitar condiciones extremas:** Aunque el sensor es resistente, es mejor evitar insertarlo en suelos extremadamente duros o rocosos que puedan dañarlo.
- **Calibración periódica:** Dependiendo del uso, considera recalibrar el sensor periódicamente para mantener la precisión.
- Este sensor es más resistente a la corrosión y duradero que los sensores resistivos tradicionales, ya que no tiene partes metálicas expuestas.

$$l = \begin{cases} 0, & \text{if } 430 \leq \text{humedad} \leq 520, \\ 50, & \text{if } 350 \leq \text{humedad} < 430, \\ 100, & \text{if } 260 \leq \text{humedad} < 350. \end{cases}$$

En la ecuación anterior, l representa el porcentaje de humedad del suelo, que toma valores de 0, 50 o 100, según el rango en el que se encuentre la variable *humedad*. La variable *humedad* corresponde al valor de la señal analógica obtenida del sensor de humedad del suelo, leída a través del ADC (Convertidor Analógico-Digital) del microcontrolador ESP32. Los valores de *humedad* varían entre 0 y 4095, que es el rango de lectura del ADC en el ESP32.

Los intervalos definidos en la ecuación son los siguientes:

- $l = 0$: cuando h se encuentra en el rango de 430 a 520, indicando un suelo completamente seco.
- $l = 50$: cuando h está entre 350 y 430, lo que representa un nivel intermedio de humedad en el suelo.
- $l = 100$: cuando h está entre 260 y 350, indicando que el suelo está completamente húmedo.

Estos umbrales fueron determinados a partir de pruebas experimentales, donde se correlacionaron los valores del ADC del ESP32 con los niveles de humedad en el suelo.

5.1.2. Pruebas con sensor de temperatura

Como se mencionó anterior mente el LM35 es un sensor de temperatura de precisión que proporciona una salida de voltaje linealmente proporcional a la temperatura. La salida del LM35 es de 10 mV por grado Celsius, lo que permite obtener lecturas de temperatura en un amplio rango. A diferencia de algunos otros sensores de temperatura, no requiere de una calibración compleja.

Conexión LM35

Conexión de los pines:

- **Vout:** Pin central del LM35 al GPIO 33.
- **Vcc:** Pin izquierdo del LM35.
- **GND:** Pin derecho del LM35.

Notas adicionales

- **Precisión y exactitud:** El LM35 tiene una precisión típica de ± 0.5 °C en un rango de -55 °C a 150 °C.
- **Entorno y factores externos:** Se debe asegurar que el sensor esté en un lugar adecuado y protegido de factores externos que puedan afectar su lectura, como corrientes de aire o luz directa del sol.
- **Protección del sensor:** Se debe evitar someter el sensor a voltajes superiores a 5V o invertir la polaridad, ya que esto puede dañarlo.

Conversión de valores del ADC

El valor leído por el ADC del ESP32 (que varía entre 0 y 4095 en resolución de 12 bits) se convierte primero en voltaje, y luego ese voltaje se traduce a temperatura en grados Celsius. El Pseudocódigo presentado anteriormente utiliza una resolución de 10 bits, donde el valor máximo del ADC es 1023, adecuado para la mayoría de las aplicaciones básicas.

Para una conversión con resolución de 10 bits, la relación se establece a partir de la salida de 10 mV/°C del LM35 y la referencia de 5V en el microcontrolador, como sigue:

$$T = \frac{\text{sensorValue} \times 5,0}{1023,0} \times 100 \quad (5.2)$$

Donde:

- $T(C)$ es la temperatura en grados Celsius.
- *sensorValue* es el valor leído por el ADC del ESP32.

- 5.0 es el voltaje de referencia utilizado (puede ser ajustado a 3.3V si se usa dicho valor).
- 1023.0 es el valor máximo del ADC cuando se usa una resolución de 10 bits.
- El factor de 100 proviene de la sensibilidad del LM35, que es de 10 mV/°C.

Si se desea aumentar la precisión, se puede configurar el ADC del ESP32 para trabajar a su resolución nativa de 12 bits, en cuyo caso el valor máximo será 4095, y la ecuación de conversión cambiaría de la siguiente forma:

$$T(C) = \frac{\text{sensorValue} \times 3,3}{4095,0} \times 100 \quad (5.3)$$

Este ajuste permite una resolución mayor y más precisa en las mediciones, especialmente en entornos donde la temperatura necesita ser monitoreada con exactitud.

Relación entre temperatura y voltaje

A continuación se presenta una tabla que muestra algunos valores de referencia de la salida de voltaje del LM35 en función de la temperatura medida:

Temperatura (°C)	Voltaje de salida (V)
0	0.012
10	0.108
20	0.201
25	0.254
30	0.307
40	0.406
50	0.503
60	0.607
70	0.703
80	0.805
90	0.908
100	1.011

Cuadro 5.1: Relación entre temperatura y voltaje de salida del LM35.

Como se mencionó en capítulos anteriores, el sensor LM35 genera una señal de salida de 10 mV por cada grado Celsius de variación en la temperatura. De este modo, a 25 °C el voltaje de salida es de aproximadamente 254 mV. Además, a medida que la temperatura aumenta o disminuye, el voltaje se modifica de forma proporcional, manteniendo una relación lineal entre ambas magnitudes.

5.1.3. Pruebas con Sensor de pH

El sensor PH-4502C es un módulo utilizado para medir el pH de una solución líquida. Es un sensor común en proyectos de agricultura, acuarios, tratamiento de agua y laboratorios, donde

la medición precisa del pH es esencial. A continuación, se explica cómo calibrar y utilizar el sensor de pH PH-4502C con Arduino.

Descripción del PH-4502C

El módulo PH-4502C incluye una sonda de pH y un módulo de acondicionamiento de señal que permite conectar la sonda a un microcontrolador como un Arduino. Este módulo puede medir el pH en un rango de 0 a 14.

Materiales necesarios

- Módulo de sensor de pH PH-4502C
- Soluciones de calibración de pH (Jugo, Coca cola, Café y Jabón)
- Cables de conexión
- Un recipiente para las soluciones

PH-4502C

Conexiones básicas:

- **VCC:** pin 5V del ESP32.
- **GND:** pin ESP32.
- **PO:** GPIO 35.
- **T0:** GPIO 34

Ajuste del potenciómetro:

Hay dos potenciómetros en el módulo:

- **Potenciómetro de offset:** Ajusta la lectura de pH.
- **Potenciómetro de rango:** Ajusta el rango de la salida analógica.

Procedimiento de calibración

Preparación:

Se debe asegurar de que la sonda esté limpia y libre de residuos antes de empezar. Sumerge la sonda en la solución de calibración de pH 7.0.

Calibración inicial:

Se debe quitar la sonda y colocar un cable entre GND y la entrada del BNC. Usa el siguiente código para leer el valor de voltaje y con el potenciómetro de rango se ajusta el voltaje. Una vez que se conecte el módulo al Arduino, nos mostrará el valor medio que debe ser 2.5V.

5.1.4. Calibración del sensor PH-4502C con soluciones comunes

La calibración del sensor PH-4502C se realizó utilizando soluciones comunes con valores de pH bien definidos. A continuación, se describen las soluciones utilizadas y el procedimiento de calibración:

- **Leche:** Se utilizó leche, que generalmente tiene un pH alrededor de 6.8. La lectura inicial fue de 6.5, ajustándose el potenciómetro de offset para obtener un valor de pH más preciso.
- **Limón:** El jugo de limón, conocido por su acidez, tiene un pH que varía entre 2 y 3. La lectura inicial fue de 2.7, lo que fue aceptable para los propósitos de calibración.
- **Bicarbonato de sodio:** Se preparó una solución de bicarbonato de sodio, que típicamente tiene un pH de 8.4. La lectura inicial fue de 8.0, ajustando el potenciómetro de rango para que coincidiera con el valor esperado.
- **Coca-Cola:** La Coca-Cola tiene un pH que generalmente se encuentra entre 2.5 y 3.5. La lectura inicial fue de 2.9, que se ajustó correctamente.

El procedimiento de calibración consistió en los siguientes pasos:

1. Se limpió la sonda del sensor para asegurarse de que no hubiera residuos que pudieran afectar la lectura.
2. Se sumergió la sonda en la primera solución (leche) y se registró el valor de pH. Se ajustó el potenciómetro de offset si era necesario para que la lectura coincidiera con el pH esperado.
3. Se repitió el proceso con las otras soluciones (limón, bicarbonato de sodio y Coca-Cola), ajustando los potenciómetros cada vez para obtener lecturas precisas.

Valores de pH de Soluciones Comunes

A continuación, se presenta una tabla que muestra los valores de pH obtenidos durante la calibración:

Solución	Valor de pH Esperado	Valor de pH Medido
Leche	6.8	6.5
Jugo de Limón	2.0 - 3.0	2.7
Bicarbonato de Sodio	8.4	8.0
Coca-Cola	2.5 - 3.5	2.9

Cuadro 5.2: Valores de pH esperados y medidos para soluciones comunes.

Voltaje	pH
1 V	2.8 pH
2.5 V	7 pH
5 V	14 pH

Cuadro 5.3: Relación Ideal Entre Voltaje y pH

5.2. Librerías de comunicación

EBYTE22 y SoftwareSerial

En la programación del ESP32 y el módulo LoRa, es necesario configurar el ESP32 para que pueda comunicarse con el módulo E220. Esto se logra al conectar el módulo LoRa al ESP32 a través de la UART y configurar los pines correspondientes. Además, se deben establecer parámetros como la frecuencia de transmisión y la tasa de baudios.

Se investigaron diversas librerías de código abierto disponibles en GitHub, tales como “EByteLoRaE220SeriesLibrary-master”, “LoRa_220” y “SoftwareSerial”, que son esenciales para la implementación de la comunicación entre el módulo ESP-32 *WROOM* y el módulo inalámbrico E220-400T22D. Para complementar esta investigación, se consultaron tutoriales y videos en YouTube relacionados con el uso del módulo “EBYTE22”. La recopilación de información se centró en comprender cómo estas librerías pueden facilitar la comunicación y el intercambio de datos entre dispositivos a través de conexiones seriales.

5.2.1. Biblioteca EBYTE22

La biblioteca EBYTE22 proporciona un conjunto de funciones que permiten la inicialización, configuración y manejo de la comunicación a través del módulo EBYTE22. **Inicialización y configuración del módulo:** El módulo se inicializa mediante el método `init()`, que configura los pines del microcontrolador necesarios para la operación.

Establecimiento de configuraciones: Para ajustar los parámetros del módulo, se utilizan métodos como `setMode()`, `setAddress()` y `setNetID()`. Estos métodos permiten modificar los valores en la biblioteca local; sin embargo, para aplicar estos cambios al módulo, se debe invocar `writeSettings()` o `writeSettingsWireless()`.

Envío de datos: Una vez configurado el módulo, se pueden enviar datos utilizando métodos específicos como:

- `sendByte(uint8_t TheByte)`: Envía un byte a través del módulo.
- `sendStruct(const void *TheStructure, uint16_t size_)`: Envía una estructura de datos.

Recepción de datos: Para recibir datos, primero se verifica la disponibilidad utilizando `available()`. En caso de que haya datos, se pueden leer con:

- `getBytes()`: Lee un byte del módulo.
- `getStruct(const void *TheStructure, uint16_t size_)`: Lee una estructura de datos.

Finalización de la transmisión: Tras enviar los datos, se puede utilizar `flush()` para garantizar que la transmisión se complete. En caso de que el módulo esté ocupado, se puede esperar su disponibilidad con `completeTask()`, que permite especificar un tiempo de espera.

5.2.2. Biblioteca `SoftwareSerial`

La biblioteca `SoftwareSerial` facilita la comunicación serial entre un dispositivo ESP8266 o ESP32 y otros dispositivos mediante pines GPIO. Esta biblioteca ofrece una interfaz sencilla para la configuración de la comunicación y el envío/recepción de datos a través de la comunicación serial por software. **Definición del espacio de nombres `EspSoftwareSerial`:** Agrupa todas las clases y funciones relacionadas con la comunicación serial por software, evitando conflictos de nombres en el código.

Clase `UARTBase`: Proporciona la funcionalidad fundamental para enviar y recibir datos, estableciendo la base para la comunicación serial por software.

Métodos importantes:

- `begin`: Inicializa la comunicación serial con los parámetros especificados.
- `read`: Lee un byte de datos del puerto serial.
- `write`: Escribe un byte de datos en el puerto serial.
- `available`: Devuelve la cantidad de datos disponibles para ser leídos.
- `flush`: Vacía los datos pendientes en el búfer de salida.

Clase `BasicUART`: Hereda de `UARTBase` y añade funcionalidades específicas para configurar los pines GPIO utilizados en la comunicación.

Clase `SoftwareSerial`: Alias para `EspSoftwareSerial::UART`, que permite la creación de instancias de la clase `BasicUART`. También incluye enumeraciones que facilitan la configuración de la comunicación serial mediante opciones predefinidas.

Explicar el funcionamiento de las funciones y librerías en C++ es crucial para la implementación efectiva de la comunicación en este proyecto.

5.3. Transmisión y Recepción

5.3.1. Transmisión (TX)

En la librería EBYTE22 se utilizan los ejemplos EBYTE22_Send_data y EBYTE_Repeater para la transmisión y recepción de datos. Estos ejemplos son fundamentales para llevar a cabo las pruebas.

Se crea un objeto `SoftwareSerial` denominado `E22Serial` para establecer la comunicación con el módulo a través de un puerto serie virtual. Este objeto se inicia con los pines TX y RX especificados. Además, se crea un objeto EBYTE22 denominado `E22` para interactuar con el módulo EBYTE22, pasando el objeto `SoftwareSerial` previamente creado y los pines de control.

Se definen variables de ejemplo que se enviarán al otro módulo: un byte (`testByte`), una cadena (`testString`) y una estructura (`testStruct`) que contiene diferentes tipos de datos. En la función `setup()`, se inicializa la comunicación tanto con el puerto serie para la depuración como con el puerto serie virtual para comunicarse con el módulo. Luego, se inicializa el módulo EBYTE22 utilizando el método `init()`. Si la inicialización es exitosa, se imprime un mensaje de confirmación; de lo contrario, se imprime un mensaje de error.

En la función `loop()`, se verifica si el módulo está ocupado mediante `getBusy()` y se espera su disponibilidad con un tiempo de espera de 5 segundos (utilizando `completeTask()`). Posteriormente, se envían los datos utilizando los métodos proporcionados por la biblioteca EBYTE22. Primero, se envía la estructura (`testStruct`) con `sendStruct()`, luego se envía un byte (`testByte`) con `sendByte()`, y finalmente se envía una cadena (`testString`) a través de la instancia de `SoftwareSerial`. Se utiliza `flush()` para garantizar que todos los datos se envíen completamente, y se espera un tiempo determinado antes de que el bucle vuelva a ejecutarse con `delay()` para no sobrecargar la comunicación.

5.3.2. Recepción (RX)

De manera similar al código anterior, se definen constantes que especifican los pines a los que están conectados el transmisor (TX), el receptor (RX) y otros pines de control del módulo EBYTE22. Se crea un objeto `SoftwareSerial` llamado `E22Serial` para establecer la comunicación con el módulo, inicializándolo con los pines TX y RX especificados.

Se crea un objeto EBYTE22 denominado `E22` que se utiliza para interactuar con el módulo. Se definen variables de destino donde se guardarán los datos recibidos: un byte (`testByte`), una cadena (`testString`) y una estructura (`testStruct`). En la función `setup()`, se inicializa la comunicación en el puerto serie para depuración y en el puerto serie virtual para comunicarse con el módulo. Se utiliza el método `init()` para inicializar el módulo EBYTE22 y si la inicialización es exitosa, se imprime un mensaje de confirmación.

En el ciclo principal (`loop()`), se verifica si hay datos disponibles para leer utilizando el método `available()`. Si se detectan datos, se leen y se almacenan en las variables de destino utilizando

los métodos correspondientes. Finalmente, se muestran los datos recibidos en el puerto serie para confirmar que la comunicación ha sido exitosa.

NOTA. La biblioteca EBYTE22.h utilizada en este proyecto fue extraída de KrisKasprzak/EBYTE [20] en GitHub. Esta biblioteca proporciona funcionalidades esenciales para la comunicación y el manejo de los módulos EBYTE.

5.4. Pruebas de transmisión y recepción con módulo LoRa

Una vez seleccionados y diseñados los sensores, se procedió a la programación en el entorno de Arduino IDE, asegurando que todos los dispositivos estuvieran debidamente configurados y listos para su integración en el sistema de monitoreo agrícola.

5.4.1. Envío de Datos

Los datos se transmiten a través de la interfaz UART (8N1) a una velocidad de 9600 bps hacia el módulo E220, que se encarga de enviarlos. Se configuro el módulo a una frecuencia por defecto de 433 MHz, permitiendo la transmisión de los datos con un consumo energético de 158.49 mW.

Captura de pantalla que se muestra a continuación, muestra el monitor serial de Arduino IDE: envío y recepción de datos. En el cuadro 5.4 se muestra la interfaz del monitor serial durante la ejecución del programa, donde se puede observar la confirmación del envío y la recepción exitosa de datos entre los módulos. Esta captura destaca la funcionalidad del sistema, demostrando que los sensores están comunicándose eficazmente y transmitiendo la información esperada.

Temp (°C)	Hum (%)	Solar	pH	Temp (°C)	Hum (%)	Solar	pH
22.3	92	49	6.89	22.3	96	49	6.89
21.2	99	49	6.86	21.2	99	49	6.86
22.4	94	47	6.84	22.4	94	47	6.84
21.1	93	48	6.88	21.1	93	48	6.88
22.6	97	46	6.81	22.6	97	46	6.81
21.5	93	47	6.85	21.5	93	47	15.37

Cuadro 5.4: Comparación de datos entre el transmisor (TX) y el receptor (RX). A la izquierda se muestra el transmisor, que envía datos desde el microcontrolador ESP32, y a la derecha se observa el receptor ESP-12E, que recibe la información a través del módulo EBYTE22.

El uso del UART es una ventaja, ya que proporciona una interfaz sencilla y ampliamente soportada, asegurando que ambos dispositivos puedan comunicarse sin la necesidad de complejas configuraciones adicionales. Esto resulta ideal para aplicaciones de monitoreo y control, como las gestionadas por *SIGACI*, donde es crucial una transmisión de datos confiable. [34]

5.5. Procesamiento y almacenamiento de datos en el servidor

Como se menciono anterior mente Se llevó a cabo la instalación de un servidor en una Raspberry Pi 4B, utilizando el Sistema Operativo Debian 12 y un servidor Apache v2.4.57, con el objetivo de alojar los datos obtenidos del módulo ESP32.

Funcionamiento de la base de datos

La base de datos implementada en MySQL, es un sistema relacional que organiza los datos en tablas. Cada tabla contiene columnas para almacenar: Fecha y hora de la medición, variables de los sensores y numero de muestra.

5.5.1. Procesamiento de datos.

Los datos recibidos desde el módulo receptor *LoRa (Raspberry Pi)* se procesan y se insertan en la base de datos mediante consultas SQL. Estas consultas son automatizadas mediante un script que procesa las tramas de datos recibidas.

Consulta y visualización.

Los usuarios acceden a los datos mediante una interfaz web, que realiza consultas a la base de datos para mostrar información en casi tiempo real o histórica según los parámetros de búsqueda seleccionados.

El siguiente pseudocódigo realiza el envío de los datos de los sensores al servidor web mediante una solicitud HTTP GET. A continuación, se explica brevemente. Más adelante se explicará como el servidor maneja los datos.

```
INICIAR URL CON "/hola.php?hum="
```

```
AGREGAR humedad A URL
```

```
AGREGAR "&sol=" A URL
```

```
AGREGAR solar A URL
```

```
AGREGAR "&temp=" A URL
```

```
AGREGAR temperatura A URL
```

```
AGREGAR "&ph=" A URL
```

```
AGREGAR ph A URL
```

```
DEVOLVER URL
```

Construcción de la URL: Se inicializa la variable `url` con la ruta del archivo en el servidor (en este caso, `/hola.php`) y se le añade el parámetro `hum`, que representa el valor de humedad.

Posteriormente, se concatenan los demás parámetros: `sol` para la radiación solar, `temp` para la temperatura y `ph` para el valor de pH. Estos valores se obtienen de la estructura `testStruct`, que previamente almacenó los datos recibidos de los sensores.

Un ejemplo del resultado final de la URL sería el siguiente:

```
/hola.php?hum=45&sol=200.5&temp=22.3&ph=6.7
```

```
INICIAR SOLICITUD HTTP CON "GET "  
AGREGAR URL A SOLICITUD  
AGREGAR " HTTP/1.1\r\n" A SOLICITUD  
AGREGAR "Host: " A SOLICITUD  
AGREGAR host A SOLICITUD  
AGREGAR "\r\n" A SOLICITUD  
AGREGAR "Connection: close \r\n\r\n" A SOLICITUD  
  
ENVIAR SOLICITUD AL CLIENTE
```

5.5.2. Código PHP

Este script en PHP tiene como finalidad la recepción y procesamiento de datos provenientes de sensores ambientales, específicamente los parámetros de temperatura y humedad, así como el identificador del nodo del sensor. Al inicio de la ejecución del script, se extraen los valores correspondientes a la temperatura, la humedad y el nombre del nodo mediante la variable global `$_GET`, la cual permite la captura de parámetros transmitidos a través de una solicitud HTTP GET.

Posteriormente, se establece una conexión con una base de datos MySQL utilizando las credenciales de acceso previamente definidas. Una vez establecida la conexión, se procede a seleccionar la base de datos destinada al almacenamiento de los datos.

El script genera una consulta SQL que inserta los valores obtenidos, junto con una marca temporal, en una tabla designada como `datosb`. Finalmente, se ejecuta esta consulta, lo que permite almacenar la información recopilada en la base de datos. Este enfoque no solo facilita el seguimiento de las condiciones ambientales registradas por los sensores casi en tiempo real, sino que también proporciona una base para el análisis posterior de los datos.

5.5.3. Código de Interfaz usuario

El código presentado tiene como objetivo desarrollar una interfaz web para visualizar los datos de un sistema de sensores. La aplicación se conecta a una base de datos MySQL, denominada "sensor", donde se almacenan las mediciones obtenidas de diversos sensores, como temperatura y humedad.

La primera sección del código establece una conexión con la base de datos utilizando las credenciales de acceso, garantizando que el sistema pueda interactuar con la información almacenada. A continuación, se define la estructura básica del documento HTML, que incluye la inclusión de un archivo CSS para el estilo visual de la aplicación.

La interfaz proporciona un formulario de búsqueda que permite a los usuarios filtrar los datos de los sensores mediante varios criterios, como el identificador del sensor, el tipo de sensor, la temperatura y la humedad. Esto se logra mediante el uso de métodos HTTP GET que envían los parámetros de búsqueda al servidor.

Los resultados se presentan en una tabla, donde se muestran los registros relevantes junto con su correspondiente marca de tiempo, lo que facilita la interpretación de los datos. Adicionalmente, el código implementa una lógica para resaltar las celdas de temperatura según su valor, mejorando la experiencia del usuario al identificar fácilmente las condiciones del sensor.

A través de esta aplicación, se busca facilitar la visualización y el análisis de los datos recolectados por los sensores, contribuyendo al monitoreo eficiente de las condiciones ambientales.

5.5.4. Recopilación de datos en PHP

En el código PHP proporcionado anteriormente, los datos de los sensores se pasan desde el ESP-32 al servidor PHP a través de una solicitud HTTP GET. Los datos se envían como parámetros en la URL. Por ejemplo:

```
/hola.php?humedad=30&ph=7.2&temperatura=25.5&solar=500
```

El servidor PHP recibe esta solicitud y utiliza la función `$_GET` para extraer los valores de humedad, pH, temperatura y solar de la URL. Luego, estos valores se pueden procesar y almacenar en una base de datos.

Envío de la solicitud al servidor: Se construye una solicitud HTTP GET utilizando la URL generada anteriormente. La función `client.print()` envía esta solicitud al servidor web indicado en la variable `host`. La solicitud incluye los siguientes elementos:

- El método HTTP GET, que solicita una acción en el servidor.
- La URL con los parámetros generados.
- El nombre del host, que corresponde a la dirección IP del servidor.
- La indicación de cerrar la conexión después de enviar la solicitud (`Connection: close`).

El mensaje completo que se envía sería similar al siguiente:

```
GET /hola.php?hum=45&sol=200.5&temp=22.3&ph=6.7 HTTP/1.1
Host: 192.168.1.80
Connection: close
```


5.5.5. Control de tiempo de espera (timeout)

El programa espera la respuesta del servidor, pero para evitar que se quede bloqueado indefinidamente, se utiliza un temporizador de espera (timeout). La condición `client.available() == 0` indica que no hay datos disponibles desde el servidor. Si transcurren más de 5000 ms (5 segundos) y no se recibe respuesta, se imprime un mensaje de error en el monitor serial (`Client Timeout !`) y se cierra la conexión mediante la función `client.stop()`.

Este enfoque garantiza que el sistema no se quede esperando indefinidamente en caso de que ocurra algún problema con la conexión al servidor.

5.5.6. Llenado de la Base de Datos

Una vez que el servidor PHP ha obtenido los valores de los sensores, puede insertarlos en una base de datos. En el ejemplo proporcionado, se usa MySQL para la base de datos. Se establece una conexión con la base de datos y luego se ejecuta una consulta SQL para insertar los valores en la tabla `mediciones`. La consulta se realiza utilizando la función `query()` del objeto `mysqli`.

```
$sql = "INSERT INTO mediciones (humedad, ph, temperatura, solar)
VALUES ($humedad, $ph, $temperatura, $solar)";
```

Esta consulta SQL inserta los valores de humedad, pH, temperatura y solar en la tabla `mediciones`. Una vez que se ejecuta la consulta, los datos se almacenan en la base de datos y están disponibles para su posterior análisis o visualización.

5.5.7. Conexión a la Base de Datos

Este código en PHP establece una conexión a una base de datos MySQL utilizando la clase `mysqli`. Primero, define las credenciales de conexión, incluyendo el servidor (`localhost`), el nombre de usuario, la contraseña y el nombre de la base de datos. Luego, crea una instancia de la clase `mysqli` con estos datos. A continuación, verifica si hubo errores durante la conexión utilizando la propiedad `connect_error`. Si hay un error, el script termina con el mensaje `Conexión fallida` seguido del detalle del error; de lo contrario, la conexión queda lista para ser utilizada en operaciones posteriores.

```
// Datos de conexión a la base de datos
$servername = "localhost";
$username = "usuario";
$password = "contraseña";
$dbname = "nombre_base_de_datos";

// Crear conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar la conexión
if ($conn->connect_error) {
```

```

        die("Conexión fallida: " . $conn->connect_error);
    }

```

\$servername, \$username, \$password y \$dbname contienen la información necesaria para conectarse a la base de datos MySQL. Estos valores deben ajustarse según la configuración de MySQL.

Se crea una nueva instancia de la clase `mysqli` y se establece la conexión utilizando los datos proporcionados. Se verifica si la conexión se realizó correctamente. Si hay algún error en la conexión, el script se detiene y muestra un mensaje de error.

5.5.8. Inserción de Datos en la Base de Datos

Una vez establecida la conexión, se pueden insertar los datos de los sensores en la base de datos.

```

// Insertar los valores en la base de datos
$sql = "INSERT INTO mediciones (humedad, ph, temperatura, solar)
        VALUES ($humedad, $ph, $temperatura, $solar)";

if ($conn->query($sql) === TRUE) {
    echo "Datos almacenados correctamente";
} else {
    echo "Error al almacenar datos: " . $conn->error;
}

```

Se construye una consulta SQL que inserta los valores de humedad, pH, temperatura y solar en la tabla `mediciones`. Se ejecuta la consulta utilizando el método `query()` del objeto `mysqli`. Si la consulta se ejecuta correctamente, se muestra un mensaje indicando que los datos se almacenaron correctamente. Si hay algún error en la consulta, se muestra un mensaje de error.

5.5.9. Cierre de la Conexión

Es importante cerrar la conexión a la base de datos una vez que se hayan completado todas las operaciones necesarias. Esto se hace para liberar los recursos y evitar posibles problemas de seguridad. Aquí está el código que cierra la conexión:

```

$conn->close();

```

Este código se coloca al final del script PHP, después de todas las operaciones de inserción de datos.

Con esto, se completa el proceso de conexión a la base de datos y la inserción de datos de los sensores en ella.

Conexión a la base de datos MySQL a través de una conexión WiFi

Se describen los pasos básicos para establecer la conexión e implementación:

- Se incluyen las librerías necesarias, como `SoftwareSerial` para la comunicación serie y las librerías para la conexión WiFi y MySQL.
- Se definen las constantes como el nombre de la red WiFi (SSID), contraseña, dirección IP del servidor MySQL y el puerto de conexión.

Declaración de variables y estructuras de datos

Se define una estructura llamada `myStruct` para almacenar los datos del sensor, como la humedad, pH, energía solar y temperatura.

Configuración del hardware y conexión WiFi

Se inicializa la comunicación serie y se establece la conexión WiFi.

Bucle principal (loop)

- Se comprueba si hay datos disponibles del sensor a través del módulo EBYTE22.
- Se leen los datos del sensor en la estructura `testStruct`.
- Se establece una conexión con el servidor MySQL a través de la conexión WiFi.
- Se construye una URL con los datos del sensor para enviarlos al servidor.
- Se realiza una solicitud HTTP GET al servidor MySQL con la URL construida, enviando así los datos del sensor.
- Se espera una respuesta del servidor y se imprime en el puerto serie.

Conexión al servidor MySQL

La conexión al servidor MySQL se realiza mediante una solicitud HTTP GET, por lo que se espera que el servidor tenga un script (por ejemplo, un script PHP) que procese la solicitud y almacene los datos en la base de datos MySQL. El uso de `millis()` para el control del tiempo garantiza que la conexión al servidor MySQL se realice periódicamente según el intervalo especificado por el *watchdog*.

El código utilizado establece una conexión WiFi y envía datos del sensor a una base de datos MySQL mediante solicitudes HTTP GET. La parte del servidor que recibe los datos y los almacena en la base de datos no está incluida en el código proporcionado y debe implementarse por separado.

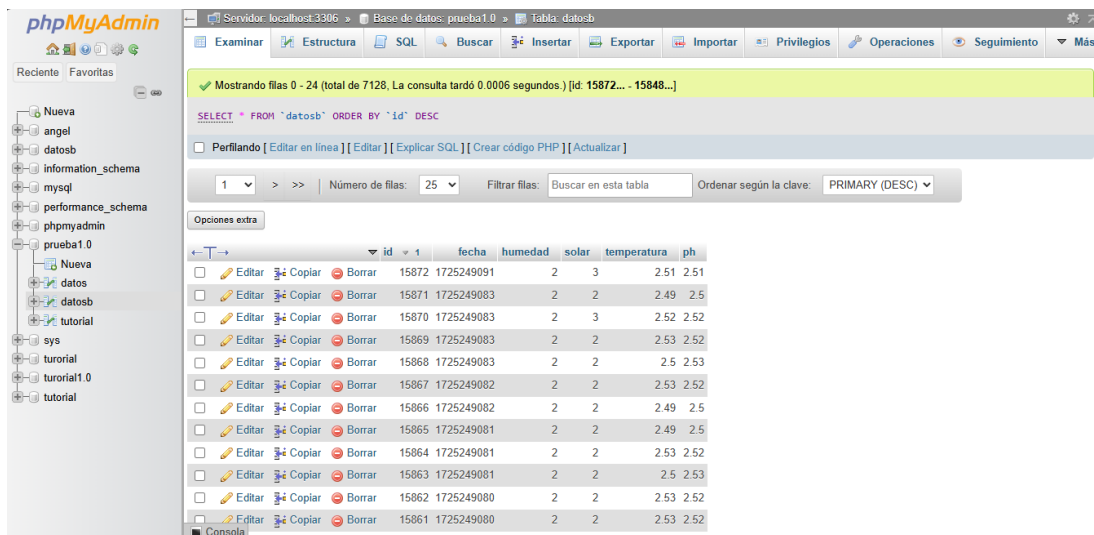


Figura 5.1: Base de datos en funcionamiento

La configuración de un servidor MySQL permitió la gestión eficiente de la información, mejorando su calidad y accesibilidad.

Las pruebas de validación y efectividad se realizaron de manera remota, confirmando el correcto funcionamiento de cada sensor y la confiabilidad en el envío de datos.

Dicha información se muestra en la interfaz WEB diseñada para la consulta de datos por día o lectura por sensor. en la Fig. 5.2.

Id	Fecha	Humedad	Solar	Temperatura	Ph	Mostrar todo
260	08-06-2024	0	330	29.43	5.64	
261	08-06-2024	0	345	29.79	5.64	
262	08-06-2024	0	345	29.55	5.63	
263	08-06-2024	0	320	29.3	5.63	
264	08-06-2024	0	323	29.3	5.63	
265	08-06-2024	0	321	30.04	5.63	
266	08-06-2024	0	352	31.26	5.63	
267	08-06-2024	0	345	30.89	5.63	
268	08-06-2024	1	345	29.3	5.64	
269	08-06-2024	0	206	30.28	5.63	
270	08-06-2024	0	133	29.79	5.63	
271	08-06-2024	0	158	30.04	5.63	
272	08-06-2024	0	289	30.16	5.63	
273	08-06-2024	0	362	29.43	5.65	

Figura 5.2: Entorno WEB del sistema SIGACI.

5.5.10. Rango de envío

Con el objetivo de estimar el alcance del sistema *SIGACI*, se realizaron simulaciones utilizando los modelos de Longley-Rice y Free Space. Estas simulaciones permitieron calcular la distancia máxima en la que se puede recibir datos de manera eficiente, evaluando factores como la propagación de la señal y las posibles pérdidas en el entorno. El modelo de Longley-Rice es ideal para prever el comportamiento de la señal en terrenos irregulares y entornos reales, mientras que el modelo de Free Space proporciona una referencia teórica para condiciones de línea de vista sin obstrucciones. Estos análisis ayudan a determinar el rango efectivo del sistema de monitoreo a través de LoRa en diferentes escenarios de operación.

El modelo Longley-Rice, también conocido como el Modelo de Terreno Irregular (ITM), permite simular el comportamiento de la radiación electromagnética desde un punto de transmisión a través de terrenos con irregularidades, como colinas o edificios. Está diseñado para operar en un rango de frecuencias de 20 MHz a 20 GHz, cubrir distancias de 1 a 2000 km y considerar alturas de antena entre 0.5 y 3000 metros. Este modelo calcula la pérdida de trayectoria al tener en cuenta diversos factores, como la pérdida por espacio libre, la difracción por el terreno y obstáculos, la reflexión en la superficie terrestre, la refracción atmosférica y la dispersión troposférica. Gracias a estos cálculos, es posible predecir con mayor precisión el rendimiento de las comunicaciones en entornos reales donde la señal puede verse afectada por diversos obstáculos naturales y artificiales. [39].

El modelo *Longley-Rice* proporciona estimaciones de la pérdida de señal combinando datos físicos y empíricos sobre la propagación de ondas electromagnéticas. Aunque su implementación completa es muy compleja y va más allá de una simple fórmula, se puede aproximar utilizando una fórmula básica.

$$A = AFS + AA + AH + AM - A_c, \quad (5.4)$$

Donde A representa la atenuación total en dB, AFS es la atenuación por espacio libre (free-space loss), la cual depende de la distancia entre el transmisor y el receptor. AA corresponde a la atenuación adicional provocada por la curvatura de la Tierra, mientras que AH es la atenuación adicional causada por la presencia de obstáculos como edificios y árboles. AM hace referencia a la atenuación adicional ocasionada por la dispersión multimodal de la señal, y finalmente, A_c es la atenuación corregida, que toma en cuenta la variabilidad estadística de la señal.

El modelo de propagación de espacio libre es uno de los más básicos y esenciales en el análisis de la propagación de ondas de radio. Se asume que la señal se desplaza en un entorno ideal, libre de obstáculos, y que la única pérdida de señal proviene de la dispersión de la energía de la onda electromagnética a medida que se expande en el espacio. Este modelo es útil para entender el comportamiento de la señal en condiciones ideales, proporcionando una referencia teórica para la pérdida de señal sin interferencias físicas o ambientales.

$$FSPL_{(dB)} = 20\log(d) + 20\log(f) + 20\log\left(\frac{4\pi}{c}\right), \quad (5.5)$$

Donde:

- L_{dB} es la pérdida de trayecto en decibelios (dB).
- d es la distancia entre el transmisor y el receptor en kilómetros (km).
- f es la frecuencia de la señal en megahercios (MHz).
- $\left(\frac{4\pi}{c}\right)$ es una constante que ajusta las unidades de la fórmula para distancias en kilómetros y frecuencias en megahercios.

El entorno de simulación *MATLab Online* ofrece la posibilidad de examinar una amplia gama de factores que pueden influir en la propagación de señales de radio en una ubicación determinada. Entre estos factores se incluyen la frecuencia de la señal, la altura de las antenas tanto emisoras como receptoras, la topografía del terreno, la conductividad del suelo, las condiciones meteorológicas y otros parámetros relevantes que pueden afectar la calidad de la transmisión.

La altura de las antenas juega un papel crucial en la distancia de transmisión, ya que una mayor elevación puede ayudar a evitar obstáculos en el terreno. Asimismo, la frecuencia de la señal es un aspecto esencial, ya que diferentes frecuencias pueden tener distintos comportamientos de propagación. Por otro lado, la topografía puede crear sombras de señal y reflejos, lo que impacta en la recepción. Además, la conductividad del suelo puede influir en la pérdida de señal y la eficiencia de la transmisión. Las condiciones atmosféricas, como la humedad y la temperatura, también pueden modificar el comportamiento de la señal en su trayecto.

Utilizando este software, es posible generar simulaciones que permiten visualizar y analizar estos factores en detalle, facilitando una mejor comprensión de cómo cada uno puede afectar la propagación de las ondas de radio en un entorno específico. A continuación, se presentan las simulaciones resultantes:

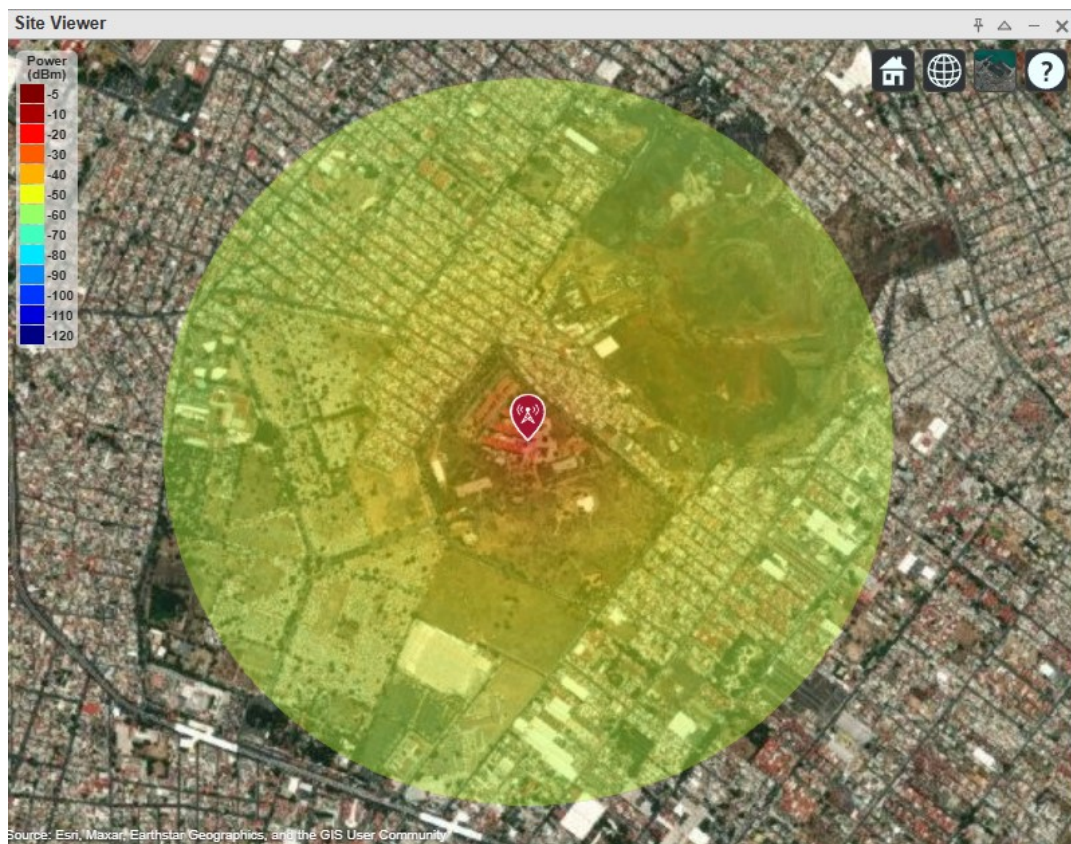


Figura 5.3: Modelo FreeSpace representando la cobertura de transmisión con una potencia de entre -55 dBm y -5 dBm ubicando el TX en los laboratorios de la Universidad Autónoma de la Ciudad de México (UACM) sede San Lorenzo Tezonco (SLT), Coordenadas [19.3128513,-99.0577421].

5.5.11. Desarrollo de Aplicación Android

Para el desarrollo de la aplicación inició la investigación sobre cómo crear una aplicación para dispositivos Android, relacionada con el proyecto.

La aplicación se desarrolla por medio de una serie de pasos utilizando la plataforma **App Inventor**, la cual permite crear aplicaciones móviles para dispositivos Android mediante un entorno visual basado en bloques. Esta herramienta está diseñada para usuarios sin experiencia en programación, y facilita el acceso a funciones del dispositivo como la cámara y el GPS, permitiendo el desarrollo de aplicaciones funcionales sin la necesidad de escribir código.

Creación de la aplicación

La aplicación fue creada en cuatro partes:

- La primera parte consta de los botones inicio de sesión y registro.

- La segunda parte fue crear la parte de registro de usuario donde se le pide al usuario que ingrese una contraseña y que la vuelva a confirmar.
- La tercera parte fue crear el apartado de inicio de sesión, donde el usuario ingresa los datos del registro.
- La cuarta parte se encarga de mostrar al usuario la página donde se visualizarán los datos obtenidos en la base de datos.

1. Diseño de la Interfaz de Usuario

Pantalla de Inicio (Screen 1)

Primero, creamos dos botones: **Iniciar sesión** y **Registrarse**. Estos botones se centran tanto vertical como horizontalmente en la pantalla usando un Layout Vertical. El color de fondo puede ser modificado a uno que prefieras (como rosado). Agregamos etiquetas para guiar al usuario en las siguientes pantallas.

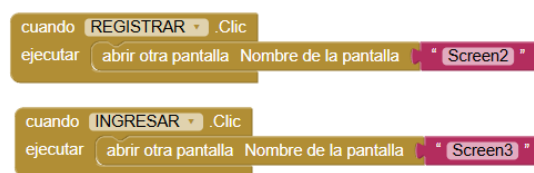


Figura 5.4: Diseño de pantallas.

Pantalla de Registro (Screen 2)

En esta pantalla, se incluyen campos de texto para ingresar el nombre de usuario y la contraseña. También agregamos un botón de **Registrar** para permitir que el usuario se registre y un botón **Volver al inicio** para regresar a la pantalla principal.

Pantalla de Inicio de Sesión (Screen 3)

Similar a la pantalla de registro, se necesitan dos campos de texto: uno para el usuario y otro para la contraseña. Se agrega el botón **Iniciar sesión** para validar los datos y acceder a la aplicación. Un botón **Volver al inicio** permite regresar a la pantalla principal si es necesario.

2. Lógica de la Aplicación

Navegación entre pantallas:

Al presionar **Registrarse**, la aplicación redirige a la pantalla de registro (Screen 2). Al presionar **Iniciar sesión**, la aplicación redirige a la pantalla de inicio de sesión (Screen 3). Al presionar **Volver al inicio** en cualquier pantalla secundaria, la aplicación regresa a la pantalla principal.

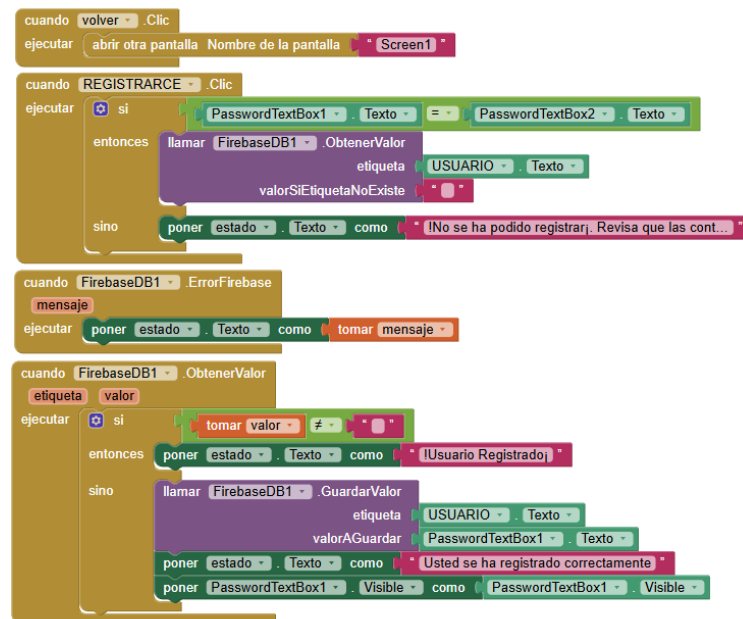


Figura 5.5: Diseño registro.

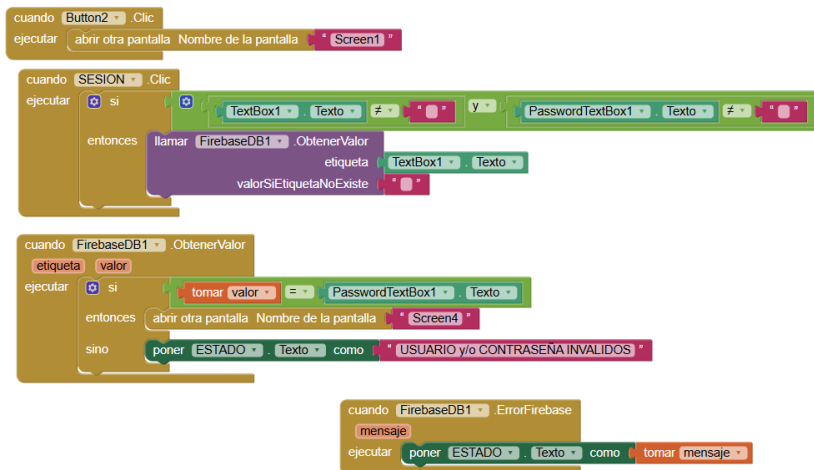


Figura 5.6: Configuración de la primer pantalla.

(Screen 1).

Base de Datos Firebase:

Se utiliza **Firebase** para almacenar los usuarios registrados. Durante el registro, la aplicación verifica si las contraseñas son iguales antes de guardarlas en Firebase. Si las contraseñas no coinciden, se muestra un mensaje de error.

Validación de contraseñas:

Se implementa una lógica condicional para asegurarse de que las contraseñas coincidan. Si las contraseñas coinciden, se registra al usuario en la base de datos; de lo contrario, se muestra un mensaje de error.

Notificación al usuario:

Si el registro es exitoso, la aplicación muestra un mensaje de confirmación: **Te has registrado correctamente**. Si ocurre un error, se muestra **No se ha podido registrar. Revisa que las contraseñas sean iguales**.

3. Manejo de Errores y Excepciones

Se implementó un manejo básico de excepciones para controlar posibles errores en la aplicación, mostrando mensajes adecuados al usuario en caso de fallos. Además, se arregló un error relacionado con el uso incorrecto de las funciones para abrir y cerrar pantallas, cambiando la función **close** por **open**.

4. Proceso de Registro e Inicio de Sesión

El usuario ingresa un nombre de usuario y contraseña, y luego confirma la contraseña. Si las contraseñas coinciden, se guarda la información en la base de datos y se muestra un mensaje de éxito. Si el nombre de usuario ya está registrado, se muestra un mensaje de error y se solicita que intente con otro nombre.

Se valida que las contraseñas coincidan antes de proceder con el registro. Además, la aplicación verifica si el nombre de usuario ya existe en la base de datos de Firebase. Si ya existe, se muestra un mensaje solicitando un nombre diferente.

5. Prueba y Demostración

Si un usuario ya está registrado, y ingresa correctamente la contraseña se permite el inicio de sesión.



Figura 5.7: Inicio de Sesión

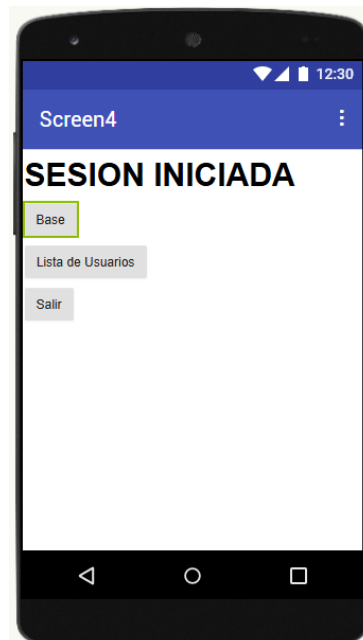


Figura 5.8: Registro de Usuario y contraseña

Id	Fecha	Humedad	Solar	Temperatura
260	08-06-2024	0	330	29.43
261	08-06-2024	0	345	29.79
262	08-06-2024	0	345	29.55
263	08-06-2024	0	320	29.3
264	08-06-2024	0	323	29.3

Figura 5.9: Base de Datos Versión Aplicación

Conclusiones del capítulo

El proceso de implementación y las pruebas detalladas realizadas en este capítulo han permitido validar la funcionalidad integral del sistema de monitoreo agrícola basado en IoT y LoRa®. Las pruebas de los sensores (humedad, pH, temperatura y exposición solar) han demostrado una alta precisión en la medición de los parámetros clave del suelo, lo cual es fundamental para la gestión eficiente de los cultivos. La calibración de los sensores y su integración en el nodo de monitoreo fueron exitosas, asegurando que los datos recolectados sean fiables y representen de manera precisa las condiciones del entorno agrícola.

En cuanto a la transmisión y recepción de datos, la tecnología LoRa® demostró ser una solución efectiva para la comunicación a largas distancias, permitiendo una transmisión continua de datos en tiempo real sin interrupciones significativas. Las pruebas de campo confirmaron que el sistema es capaz de operar bajo distintas condiciones ambientales, y que la transmisión de datos desde los nodos de sensores hacia el nodo Gateway y finalmente al servidor Raspberry Pi fue exitosa, con tiempos de respuesta aceptables.

Además, la implementación de la interfaz web y el sistema de alertas permiten a los usuarios monitorear y gestionar los cultivos de manera remota, facilitando la toma de decisiones en tiempo real. El sistema de alertas es capaz de notificar de forma inmediata cualquier anomalía en los parámetros monitoreados, permitiendo una intervención rápida y eficiente.

En definitiva, el sistema desarrollado ha demostrado ser funcional y eficiente, además de

escalable y adaptable a diferentes condiciones agrícolas. Estos resultados confirman que su aplicación práctica puede mejorar significativamente la gestión de los cultivos, optimizando recursos y aumentando la productividad agrícola, incluso en entornos rurales o de difícil acceso.

Capítulo 6

Discusión / Análisis de Resultados

6.1. Descripción de los análisis realizados para validar el sistema

Para validar el sistema de monitoreo agrícola basado en *IoT* y *LoRa*®, se llevaron a cabo diversos análisis y pruebas experimentales orientados a evaluar el rendimiento, la precisión y la fiabilidad del sistema en condiciones de campo. Estos análisis se centraron en los siguientes aspectos clave:

Precisión de los sensores: Se realizaron pruebas de *calibración y comparación de resultados* de los sensores de humedad, temperatura, pH y exposición solar. Los valores recolectados fueron comparados con instrumentos de medición de referencia para determinar su precisión y margen de error. Esta etapa fue fundamental para asegurar que los sensores proporcionaran datos fiables sobre las condiciones del suelo.

Análisis estadístico

Para verificar que el envío y la recepción de datos se realizan correctamente, se llevaron a cabo diversas pruebas. La primera prueba consistió en recopilar más de 5,000 muestras de datos utilizando sensores conectados a la planta. Estas muestras se registraron y se enviaron a través del módulo **LoRa**®, lo que permitió evaluar la efectividad y precisión de la comunicación en el sistema.

Cuadro 6.1: Datos de medición durante 12 horas.

Medida	Humedad	Lux	Temperatura [°C]	pH
Media	4.26975	142.45	22.9498	6.371
Varianza	1.856	3780.122	22.170	0.9610
Desv. estándar	1.362	61.48	4.7075	0.98035

6.1.1. Análisis de distintos valores de voltaje

Para la segunda prueba se realizaron 500 muestras. Se retiraron los sensores y se colocaron distintos niveles de voltaje con ayuda del regulador AMS1117, debido a su capacidad para proporcionar una tensión de salida constante y estable. También se modificó el código para mostrar únicamente el voltaje. A cada entrada del sensor se aplicó el mismo nivel de voltaje.

Niveles de voltaje utilizados: 1.2V, 1.5V, 1.8V y 2.5V.

Cuadro 6.2: Estadística descriptiva de las mediciones con valores propuestos.

Muestra	Humedad	Lux	Temperatura [°C]	pH
Media	1.2051	1.505	1.818	2.505
Varianza	0.0004	0.0045	0.0000	0.0000
Desv. estándar	0.0212	0.2121	0.0007	0.0007

Como se observa en los resultados, los valores adquiridos por el sistema *SIGACI* presentan una baja varianza, lo que indica una alta precisión en las mediciones realizadas.

6.2. Evaluación de desempeño

Rendimiento de la transmisión de datos: Se analizo las pruebas de *transmisión y recepción de datos*. Las pruebas incluyeron la medición de la calidad de la señal y la pérdida de paquetes en diferentes distancias, tanto en entornos abiertos como en áreas con obstáculos. Se realizo la prueba Okumura en los alrededores del plantel san Lorenzo Tezonco y con ayuda del software *MATLab Online* se realizó el mapa para saber su alcance.

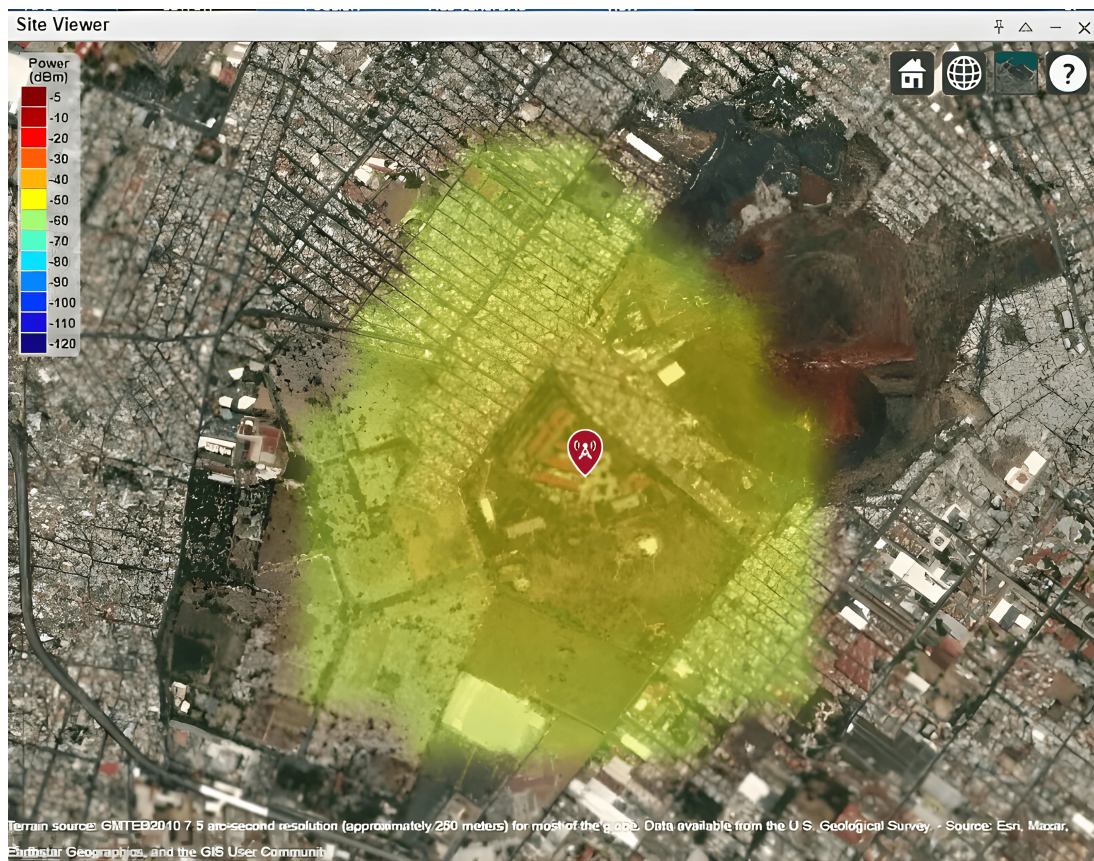


Figura 6.1: Modelo de radiación Longley-Rice con una potencia de radiación de -55dbm a -5dBm ubicando el TX en los laboratorios de la Universidad Autónoma de la Ciudad de México (UACM) sede San Lorenzo Tezonco (SLT), Coordenadas [19.3128513,-99.0577421].

Consumo energético: Se midió el consumo energético de los nodos sensores y el Gateway para evaluar la viabilidad de operar el sistema con baterías en áreas rurales.

Componente	Modo	Consumo (mA)	Consumo diario (Ah)	Consumo diario (W)
2*ESP32	Activo	240	0.0064	0.0211
	En suspensión	10	0.239	0.7887
2*Módulo LoRa E220	Transmisión	120	0.0032	0.0106
	En espera	10	0.239	0.7887
Sensores (promedio)	Continuo	20	0.48	1.584
Total	—	—	0.968	3.1931

Cuadro 6.3: Consumo energético estimado del sistema SIGACI.

Consumo diario total:

- 0.968 Ah/día a 3.3 V, equivalente a 3.19 W/día.

Autonomía estimada con una batería de 3.7 V y 2500 mAh (9.25 Wh):

- **Sin recarga:**

$$\frac{9,25}{3,19} \approx 2,9 \text{ días.}$$

- **Con optimización:** Reduciendo la frecuencia de transmisiones, la autonomía puede extenderse a 4-7 días.

Tiempo de respuesta: Se evaluó el intervalo entre la recolección de datos por los sensores y su visualización en la interfaz web, garantizando retrasos mínimos.

Fiabilidad del sistema de alertas: Se simularon condiciones adversas en cultivos para verificar el funcionamiento de las notificaciones.

Pruebas de campo: Se realizaron ensayos en entornos agrícolas reales para evaluar el desempeño general bajo diversas condiciones climáticas.

Los resultados confirman la viabilidad de implementar *SIGACI* en entornos rurales, contribuyendo a la optimización de recursos y a una mejor gestión agrícola.

Conclusiones del capítulo

El proceso de validación del sistema de monitoreo agrícola basado en IoT y LoRa®, denominado *SIGACI*, demostró su eficacia y viabilidad para su implementación en entornos rurales. Las pruebas realizadas abarcaron diversos aspectos clave como la precisión de los sensores, la fiabilidad en la transmisión de datos, el consumo energético, la autonomía y la respuesta ante condiciones adversas.

Los análisis estadísticos evidenciaron una alta precisión en las mediciones, con márgenes de error mínimos y valores de varianza bajos. Las pruebas de transmisión confirmaron la efectividad de la comunicación, incluso en escenarios con obstáculos, mediante el uso de modelos predictivos como Longley-Rice. Además, la evaluación del consumo energético destacó la capacidad del sistema para operar de manera autónoma por varios días, siendo posible extender su duración con ajustes en la frecuencia de transmisión.

Por último, las simulaciones de condiciones adversas y las pruebas en campo validaron la fiabilidad de las alertas generadas y la adaptación del sistema a diferentes climas y entornos. Estos resultados consolidan a *SIGACI* como una herramienta innovadora para la agricultura inteligente, contribuyendo a la optimización de recursos, la sostenibilidad y la mejora en la gestión de los cultivos.

Capítulo 7

Conclusiones del trabajo

El desarrollo del sistema de monitoreo agrícola IoT ha cumplido plenamente con los objetivos propuestos, demostrando la eficacia de la tecnología IoT y LoRa® en la optimización de la gestión de cultivos. A través de la implementación de un sistema integral compuesto por nodos de sensores, Gateway, servidor y nodo actuador, se consiguió la captura, transmisión, almacenamiento y visualización de datos agrícolas en tiempo casi real. Esta integración automatizó la recolección de información y fortaleció la capacidad de respuesta ante eventos críticos mediante alertas automáticas, facilitando decisiones oportunas y mejorando la eficiencia operativa.

La plataforma web desarrollada cumplió mostrar y monitorear parámetros clave como humedad, temperatura, pH y exposición solar. Esta herramienta facilitó ajustes precisos en los procesos de riego y fertilización, impulsando una planificación más eficiente de los ciclos agrícolas y mejorando la productividad mediante el análisis de datos históricos.

Además, la integración de tecnologías avanzadas, como el microcontrolador ESP32, el módulo LoRa y la Raspberry Pi 4, demostró su eficiencia para la transmisión de datos a largas distancias con bajo consumo energético. Esta implementación validó la viabilidad operativa del sistema en entornos rurales, destacando su capacidad para funcionar en áreas con infraestructura tecnológica limitada.

Durante el proceso, se enfrentaron y resolvieron diversos desafíos técnicos, como interferencias en la transmisión LoRa y dificultades en la calibración de sensores. Estos obstáculos, solucionados mediante ajustes iterativos y pruebas continuas, enriquecieron el conocimiento técnico adquirido y sentaron las bases para futuras mejoras, como la ampliación de la cobertura y la optimización de la estabilidad de la red.

En términos de sostenibilidad, el proyecto cumplió su objetivo al reducir el consumo de recursos naturales. Esta optimización permitirá a los agricultores adoptar prácticas agrícolas más eficientes, favoreciendo el uso responsable del agua y de los fertilizantes.

Además de cumplir con sus metas técnicas, este proyecto representa una valiosa contribución al conocimiento en el ámbito de la agricultura inteligente. Su diseño y resultados ofrecen un

modelo replicable para futuras investigaciones y desarrollos tecnológicos. También abre la posibilidad de integrar nuevas tecnologías, como inteligencia artificial para predicciones climáticas y sistemas autónomos para una automatización avanzada.

En cuanto a los errores y aprendizajes, se observó que cada microcontrolador presenta ventajas y desventajas. Sin embargo, durante las pruebas se identificó una dificultad al intentar realizar la transmisión con Arduino y la recepción con ESP32 y ESP8266, no fue posible obtener resultados correctos. Aunque sí existía comunicación entre los dispositivos, los datos recibidos no coincidían con los enviados. Este inconveniente generó retrasos en el desarrollo del proyecto, ya que no se conocía con certeza la causa del error, a pesar de haber realizado ajustes tanto en el código como en los microcontroladores. Finalmente, la solución adoptada fue utilizar un ESP32 para la transmisión y recepción.

En conclusión, el proyecto no solo alcanzó todos los objetivos planteados, sino que también constituye un aporte significativo para la modernización del sector agrícola, fomentando una gestión más eficiente, sostenible e inteligente de los recursos y proporcionando una base sólida para investigaciones futuras.

Capítulo 8

Recomendaciones y Trabajo Futuro

En base a los resultados obtenidos durante el desarrollo del presente trabajo, se proponen una serie de recomendaciones que pueden mejorar tanto la eficiencia como el impacto de la investigación. Estas recomendaciones están dirigidas a optimizar los procesos actuales y a proporcionar un marco para futuras investigaciones.

8.1. Recomendaciones

Mejora en la precisión de los sensores para obtener datos más confiables y representativos, es recomendable actualizar los sensores utilizados en el sistema de monitoreo agrícola. La implementación de sensores de mayor precisión permitirá una medición más exacta de variables como humedad y pH, lo cual es crucial para la toma de decisiones en tiempo real.

Optimización de la red de comunicación se sugiere explorar nuevas tecnologías o mejorar las configuraciones actuales de LoRa para aumentar la velocidad de transmisión de datos sin comprometer el alcance. Esto contribuirá a una mayor eficiencia en la comunicación entre los dispositivos en zonas agrícolas de gran tamaño.

Fortalecimiento de la seguridad de los datos a medida que el sistema SIGACI se expanda, es fundamental implementar protocolos de seguridad más robustos para garantizar la integridad de los datos transmitidos. Esto incluye cifrado de datos y autenticación de dispositivos.

En cuanto a la capacitación y sensibilización, se recomienda implementar talleres y programas de formación dirigidos a los agricultores, con el fin de que aprendan a utilizar el sistema SIGACI de manera efectiva. De esta forma, se garantizará una adopción más amplia y un uso adecuado de la tecnología..

8.2. Trabajo Futuro

El trabajo futuro en el campo de la agricultura inteligente debe centrarse en la integración de nuevas tecnologías emergentes. Algunas áreas clave a explorar son:

Integración de drones que puedan proporcionar imágenes de alta resolución que complementen los datos obtenidos por los sensores en el terreno.

Implementación de redes neuronales avanzadas

Aunque se ha propuesto una red neuronal para la toma de decisiones, se debe profundizar en el uso de algoritmos de aprendizaje automático para predecir condiciones futuras del cultivo, como plagas o enfermedades, mejorando así la capacidad de anticipación del sistema.

Expansión a nuevos cultivos y regiones.

A medida que se vayan implementando estos sistemas, sería valioso ampliar su aplicación a una mayor diversidad de cultivos y diferentes regiones, adaptando los sensores y algoritmos a las características específicas de cada zona.

Estudio de la sostenibilidad.

En el futuro, será importante investigar cómo estos sistemas pueden contribuir a prácticas agrícolas más sostenibles, con un enfoque en la optimización del uso de recursos como el agua y la energía, además de la reducción de residuos.

Expansión de Sensores para Monitoreo Avanzado.

La incorporación de sensores adicionales fortalecerá las capacidades del sistema SIGACI, proporcionando un monitoreo más completo:

- **Sensor UV:** Mide la radiación ultravioleta para ajustar la exposición y proteger los cultivos sensibles.
- **Sensor de CO₂:** Monitorea el dióxido de carbono para optimizar la fotosíntesis en invernaderos.
- **Sensor de Temperatura del Suelo:** Detecta las condiciones térmicas bajo la superficie para mejorar la siembra y el riego.
- **Sensor de Conductividad Eléctrica (CE):** Evalúa la salinidad del suelo para un manejo eficiente de fertilizantes.

- Sensor de Nutrientes (NPK): Mide niveles de nitrógeno, fósforo y potasio para ajustar la fertilización.
- Sensor de Nivel de Agua: Supervisa las reservas hídricas para evitar desperdicios y garantizar disponibilidad.
- Sensor de Humedad Foliar: Ayuda a prevenir enfermedades detectando niveles altos de agua en las hojas.
- Sensor de Calidad del Aire: Identifica contaminantes para proteger los cultivos.
- Sensor de Radiación PAR: Mide la luz fotosintéticamente activa para maximizar el rendimiento del cultivo.
- Sensor de Presión Atmosférica: Ayuda a predecir cambios climáticos que podrían impactar los cultivos.

Beneficios

Estos sensores optimizan el manejo de recursos, previenen problemas, mejoran la productividad y promueven una agricultura más sostenible.

Bibliografía

- [1] Intro a la tecnología lora, 2024.
- [2] Sensor de humedad de suelo capacitivo v1.2, 2024.
- [3] Sensor de humedad de suelo fc-28, 2024.
- [4] Proyectos Electrónica 3. [entrada de blog], 2012.
- [5] J. M. Aisalla Guanoluiza and C. D. Díaz Díaz. Desarrollo de un micro servidor web con arduino para la monitorización, análisis de calidad de suelos en las parroquias rurales del cantón latacunga, 2023.
- [6] M. A. Altieri and C. I. Nicholls. Cambio climático y agricultura campesina: impactos y respuestas adaptativas. *LEISA revista de agroecología*, 14:5–8, 2009.
- [7] A. Andreu Dólera. Modelado de redes lorawan aplicadas a la conservación preventiva de patrimonio cultural. Master's thesis, 2019.
- [8] SIG Bluetooth. Bluetooth. *Go Faster. Go Further White Paper*. Available online: https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf (accessed on 11 September 2021), 2006.
- [9] J. J. Casa Toctaguano and A. O. Mañay Quinatoa. Implementación de un sistema de monitoreo de ph aplicado a un invernadero de la zona alta sector salache. Master's thesis, Universidad Técnica de Cotopaxi (UTC), Ecuador: Latacunga, 2023.
- [10] Rodrigo Castro. Avanzando en la seguridad de las redes wifi. *Boletín de RedIRIS*, 73:23–33, 2005.
- [11] Instituto Federal de Telecomunicaciones. Inventario de bandas de frecuencias clasificadas como espectro libre, 2023. Accedido el 14 de noviembre de 2024.
- [12] Borja Díaz Mulas. Uart: Universal asynchronous receiver-transmitter. 2015.
- [13] Digilent. Uart explained, s.f. Recuperado de <https://digilent.com/blog/uart-explained/?srsltid=AfmBOoq7Brl0tisxm8h83B5HTNsAGaPRJkz3M93clDCovGy3NcmgR3kZ>.
- [14] Ebyte. Introduction to lora spread spectrum technology, 2024.

- [15] ARCA ELECTRONICA. Sensor de humedad de suelo fc28 - higrómetro arduino, 2024.
- [16] Embedded Lab Vienna for IoT & Security. Jtagulator: Find iot-device's uart interface, 2023.
- [17] A. Franco Mesa. Análisis de las técnicas de prevención y detección de enfermedades y plagas en cultivos agrícolas, desarrolladas bajo tecnologías de iot, 2020. Tesis de licenciatura.
- [18] C. A. Gordillo Gordillo. Red de sensores inalámbricos bajo protocolo lora y gestión de procesos para la analítica de datos mediante meta sistema operativo para monitoreo ambiental en invernaderos. Master's thesis, 2021.
- [19] Ebyte IoT. Wireless lora spread spectrum module 433m radio frequency sx1276 sx1278 sx1262 transparent transmission low power consumption, n.d.
- [20] KrisKasprzak. Ebyte. <https://github.com/KrisKasprzak/EBYTE>. Accedido: [10/3/2020].
- [21] Sanjeev Kumar. Design and implementation of uart using fifo for serial communication. *International Journal of Scientific Engineering and Technology*, 2(7):737–740, 2013.
- [22] Libelium. Connecting sensors to the cloud. IoT Solution Provider - Libelium, jul 2024. Accedido el 30 de julio de 2024.
- [23] J. Love. ph measurement. In *Process Automation Handbook: A Guide to Theory and Practice*, pages 107–112. 2007.
- [24] Raspberry Pi Ltd. Raspberry pi compute module 4. <https://www.raspberrypi.com/products/compute-module-4/>, 2020. Accessed: 2024-10-10.
- [25] Lenny Meléndez, Alba Hernández, and Shirley Fernández. Efecto de la fertilización foliar y edáfica sobre el crecimiento de plantas de maíz sometidas a exceso de humedad en el suelo. *Bioagro*, 18(2):107–114, 2006.
- [26] A. B. Mendoza. *Ecofisiología y bioquímica del estrés en plantas*. Universidad Autónoma Agraria Antonio Narro, Buenavista, México, 2002.
- [27] E. Odunlade. Meet the esp32-s2-saola-1 development board from espressif. <https://www.electronics-lab.com/meet-the-esp32-s2-saola-1-development-board-from-espressif/>, 2020. Accessed: 2024-10-10.
- [28] J. D. Olivares-Auza. Estudio sobre las características del protocolo de transmisión de lora. 2020.
- [29] K. Purwandaru, D. E. Faturrohman, A. R. Putri, R. P. Maghfiroh, M. R. Ridhollah, and L. K. Dewi. Rancang bangun fermentor yogurt susu kambing etawa con sistema control sensor ph 4502c y temperatura ds18b20 berbasis arduino uno r3. *Jurnal Keteknikaan Pertanian Tropis dan Biosistem*, 9(3):281–291, 2021.

- [30] J. L. Rodríguez and S. C. Suero. Modulación de señales digitales. *Universidad Politécnica de Sevilla, Departamento de Tecnología Electrónica*, 1995.
- [31] J.A. ROS MARIN and O. BARRERA DOBLADO. *Sistemas eléctricos y de seguridad y confortabilidad 2.ª edición*. Ciclos Formativos. Ediciones Paraninfo, S.A, 2016.
- [32] F. Ruiz Pérez, S. M. López Estrada, and F. Caballero Briones. El espectro electromagnético. los efectos de la radiación no ionizante y cómo prevenirlos, 2023. Avance y Perspectiva.
- [33] José Carlos Samayoa Santiago. *Integración de sensores y diseño de enlace de datos punto a punto de largo alcance, basado en tecnología LoRa y bluetooth*. PhD thesis, Universidad de San Carlos de Guatemala, 2021.
- [34] Rohde & Schwarz. Entendiendo el uart, s.f. Recuperado de https://www.rohde-schwarz.com/lat/productos/prueba-y-medicion/essentials-test-equipment/digital-oscilloscopes/entendiendo-el-uart_254524.html.
- [35] Atlas Scientific. Atlas scientific, s.f. Accedido: 8 de mayo de 2025.
- [36] Seamartini. Round ph scale indicators chart of acids balance. Imagen vectorial, 2024.
- [37] S. Serna, B. Montoya, and W. Osorio. Monitoreo del ph y fósforo soluble en la rizosfera de árboles de aguacate en el norte y oriente antioqueño. *Suelos Ecuatorianos*, 42(2):186–190, 2012.
- [38] Smnutricontrol. La luz como factor ambiental para las plantas. *Nutricontrol*, 2021.
- [39] A. Tics. Características de los modelos de propagación. Diapositivas de PowerPoint, 2024. Recuperado el 16 de junio de 2024, de https://es.slideshare.net/alfredo_tics/caracteristicas-de-los-modelos-de-propagacion.
- [40] L. A. VERA POLANCO. Prototipo de telecontrol para agricultura inteligente mediante iot y tecnología inalámbrica. Master's thesis, Jipijapa-Unesum, 2023.

Apéndice A

Artículo Publicado en el Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
y constancia.

Título del manuscrito: Transformación agrícola: implementación de un sistema de Monitoreo IoT para la gestión inteligente de cultivos
Title in english: Agriculture transformation: appliance of an IoT monitoring system for smart harvest management

J.A Quiroz-Valdez ^a, J.A. Almaraz-Damian ^b, E. Ramos-Diaz ^{a,*}

^aUniversidad Autónoma de la Ciudad de México, Colegio de Ciencia y Tecnología, ISET, Prolongación San Isidro 151, Col. San Lorenzo Tezonco, Iztapalapa, 09790, Ciudad de México, México.

^bCentro de Investigación Científica y de Educación Superior de Ensenada, Unidad de Transferencia Tecnológica Tecip (CICESE-UT³), Tecip, 63173, Nayarit, México.

Resumen

El avance de las tecnologías contemporáneas está revolucionando diversos sectores, y la agricultura no es una excepción. La internet de las cosas (IoT) ha emergido como una herramienta poderosa para optimizar el uso de los recursos y mejorar la productividad en la agricultura. Este estudio presenta el desarrollo de un sistema de monitoreo agrícola que emplea el microcontrolador ESP32, integrado con una variedad de sensores. Los datos recopilados son transmitidos mediante un módulo LoRa®, lo que permite la monitorización remota y en tiempo real de las condiciones del cultivo. Estos datos se almacenan en una base de datos y se visualizan en una página web, proporcionando una visión del estado de los cultivos, incluyendo niveles de riego y posibles afectaciones.

Palabras Clave: agricultura, IoT, datos, sistema, monitoreo.

Abstract

The advancement of modern technologies is revolutionizing various sectors, and agriculture is no exception. The Internet of Things (IoT) has emerged as a powerful tool to optimize resource use and improve productivity in agriculture. This study presents the development of an agricultural monitoring system that employs the ESP32 microcontroller integrated with various sensors. The collected data is transmitted through a LoRa ®module. Providing remote and monitoring of harvest conditions. This data is stored in a database and displayed on a webpage, providing a near real-time view of the status of the crops, including irrigation levels and their possible effects.

Keywords: harvest, IoT, data, system, monitor.

1. Introducción

La agricultura, siendo uno de los pilares fundamentales de la civilización humana, enfrenta hoy desafíos cruciales que van más allá de la simple producción de alimentos. En medio de la creciente demanda global por productos agrícolas, surge una preocupación cada vez más urgente: los impactos ambientales derivados de las prácticas agrícolas modernas.

El aumento en la producción agrícola ha tenido consecuencias negativas en los recursos naturales, tales como la escasez y la contaminación del agua, además de la contaminación del sue-

lo, el cambio climático y la pérdida de biodiversidad. Abastecer la demanda de alimentos a nivel nacional se ha convertido en un desafío perentorio. La agricultura moderna enfrenta constantes retos, desde la optimización de recursos, hasta la mitigación de riesgos ambientales.

En este contexto, la adopción de tecnologías disruptivas como el Internet de las cosas (*IoT*) ofrece nuevas oportunidades para mejorar la eficiencia y la productividad en el campo. La creación de un sistema de monitoreo agrícola *IoT* que, mediante la recopilación y análisis de datos de forma rápida y eficiente,

*Autor para correspondencia: eduardo.ramos@uacm.edu.mx

Correo electrónico: eduardo.ramos@uacm.edu.mx (Eduardo Ramos Diaz), jose.quiroz@estudiante.uacm.edu.mx (Jose Angel Quiroz Valdez), almaraz@cicese.mx (Jose Agustín Almaraz-Damian)

Historial del manuscrito: recibido el DD/MM/AAAA, última versión-revisada recibida el DD/MM/AAAA, aceptado el DD/MM/AAAA, en línea (postprint) desde el DD/MM/AAAA, publicado el DD/MM/AAAA. **DOI:** <https://doi.org/10.29057/icbi.vXXiYY.ZZZZ>



busca revolucionar la forma en que se gestionan los insumos por parte del agricultor y por ende los cultivos. Este enfoque innovador podría representar un paso fundamental hacia una agricultura más sostenible y respetuosa con el medio ambiente, debido a que los procesos que hemos llevado a cabo por miles de años ahora experimenten un cambio en la eficiencia a partir del análisis de datos que estos sistemas permiten realizar.

Los dispositivos que operan en la *IoT* comúnmente adquieren información y hacen el envío de ella por medio de diferentes protocolos de comunicación y entregan al usuario final reportes dependiendo la aplicación, con el fin de generar servicios significativos, como el monitoreo ambiental, sistemas de vigilancia o la agricultura inteligente.

Estos dispositivos han ganado interés en la última década tanto en el ámbito industrial y de investigación, ya que en la actualidad se puede diseñar, desarrollar e implementar sistemas que cuenten con diversos tipos de sensores, transductores, actuadores y sistemas de procesamiento que sean capaces de realizar una tarea específica. Una característica primordial es la transmisión de los datos adquiridos a un *Gateway* con la finalidad de un posterior procesamiento. Dicho procesamiento actualmente incluir sistemas de inteligencia artificial con la finalidad de que a partir de un evento el sistema inteligente pueda realizar una tarea.

Gordillo y A. (2021), describe un sistema de monitoreo en invernaderos basado en redes *LPWAN* (Low Powered Wireless Area Network) y un *Meta* Sistema Operativo, empleando algoritmos de Aprendizaje de Máquina para analizar variables ambientales como la radiación UV, emisiones CO₂, temperatura y humedad, mejorando la precisión al tomar la decisión de realizar el riego y otras prácticas agrícolas mediante datos procesados tecnológicamente.

El proyecto presenta diversas consideraciones importantes a tener en cuenta. Primeramente, el módulo de transmisión RFM95, al operar en frecuencias libres como las bandas *ISM*, puede ser susceptible a interferencias de otros dispositivos, lo cual afecta la fiabilidad de las comunicaciones.

Por otro lado, el sensor de CO₂, SCD30 requiere de un proceso de calibración periódica para mantener su exactitud, lo cual puede resultar costoso dependiendo del entorno y los estándares de calidad requeridos, finalmente el costo del sensor es elevado. El sensor UV, ML8511, también requiere de una calibración periódica debido a la influencia de las condiciones ambientales, afectando en la precisión de las mediciones debido a la susceptibilidad a interferencias ambientales, afectando la exactitud de los datos obtenidos.

Finalmente, la Raspberry Pi Zero, tiene limitaciones de *Hardware* (HW), ya que cuenta con un procesador de un núcleo y 512MB en RAM y debido a que emplea un Sistema Operativo este, ocupa dichos recursos en diversas tareas.

Serna et al. (2012) diseña un sistema que no requiere comunicaciones inalámbricas de ninguna índole, implicando que los

datos son adquiridos manualmente, por interfaces de usuario o de forma automática mediante *scripts* o aplicaciones dedicadas. Además, los sensores empleados están conectados a la unidad de procesamiento seleccionando la plataforma de Arduino para su envío al servidor.

No obstante, este enfoque presenta algunos inconvenientes significativos, como la dependencia de la proximidad física, debido a la limitante de transmisión por cable de los sensores y su ubicación. El requerimiento de ingreso manual de algunos datos que puede incurrir a errores por parte del encargado y las actualizaciones frecuentes.

Los sensores empleados son el DS18B20 y el sensor FC-28, el primero es un sensor de temperatura digital, el cual tiene como problemática su tiempo de respuesta debido al algoritmo empleado en la comunicación *1-Wire*®, en él que requiere que, el envío de datos se solicite a la memoria *EEPROM* interna del termómetro y este haga una codificación *CRC* para validar que los datos son enviados correctamente. El segundo sensor, adquiere datos de humedad en el suelo, la problemática que se tiene con estos sensores es que dependiendo el tipo de suelo y las condiciones ambientales locales requiere de una calibración específica.

J. y Mañay Quinatoa (2023), propone un sistema de monitoreo del pH del suelo y del agua en invernaderos, empleando sensores de pH y un sistema *SCADA* para la visualización y monitoreo automático, con la finalidad de mejorar el rendimiento de los cultivos y reduciendo pérdidas económicas, beneficiando tanto a agricultores, ya que proporciona una herramienta eficiente para el manejo agrícola.

Al emplear un sistema *SCADA* hace que aumente tanto los costos como la complejidad operativa en comparación con las soluciones modernas basadas en *IoT*, limitando su capacidad para proporcionar monitoreo y optimización de recursos naturales, aspectos esenciales en la agricultura moderna.

Como se puede observar de los métodos encontrados en el *Estado-del-Arte*, algunos carecen de capacidades de adquisición efectivas, debido a que se requiere una constante calibración de los sensores acuerdo al tipo de terreno en el que se esté implementando. Además, algunos carecen de aspectos importantes como el empleo de comunicaciones inalámbricas o el empleo de sistemas de procesamiento que, no solo, se dedican a una tarea, sino que dependen de Sistemas Operativos, los cuales realizan procesos secundarios para la adquisición de los datos. Por ello consideramos la importancia de desarrollar un sistema que sea capaz de poder mitigar los problemas descritos anteriormente, garantizando la correcta adquisición de datos y el envío de estos para su posterior procesamiento.

El presente trabajo trata de mejorar y complementar las propuestas anteriores, las contribuciones principales son:

- Transmisión de datos en cercano al tiempo real y almacenamiento en un servidor, ofreciendo una solución más completa y escalable.

- Mayor simplicidad en la estructura de transmisión y almacenamiento, facilitando la implementación y el uso en diferentes escenarios.
- Proporciona datos casi instantáneos y continuos, mejorando la toma de decisiones informadas sobre el manejo de cultivos.
- Ofrece un monitoreo más integral de múltiples parámetros y una interfaz web para una visualización más accesible y manejable.
- Cobertura y Eficiencia Energética.
- Almacenamiento y Procesamiento Eficiente.

2. Metodología

Para afrontar los desafíos ambientales en la agricultura contemporánea y establecer un sistema de monitoreo agrícola basado en el IoT que nos permita determinar si un cultivo está en condiciones óptimas, es crucial comprender los distintos problemas que pueden surgir durante su desarrollo. Este análisis, nos capacita para diseñar, desarrollar e implementar un prototipo capaz de gestionar y controlar eficientemente cada una de estas problemáticas. La implementación del Sistema de Gestión Agrícola con Control Inteligente (*SIGACI*) véase Fig.1, se llevó a cabo en diversas etapas, las cuales se describirán a continuación.

2.1. Transmisor

De acuerdo a la Fig. 1, se para el nodo transmisor, se requiere primeramente el diseño e implementación de los sensores. Para el sensor de humedad se seleccionó el SEN-HS-CAP debido a que por el método capacitivo el cual implementa, este mide los cambios en la capacitancia causados por el contenido de agua en el suelo teniendo una mejor precisión y proporcionando lecturas más conscientes en sueldos con diferentes niveles de salinidad. Además, es menos susceptible a la variedad de lecturas debido a la corrosión en comparación a otros sensores de humedad como el FC-28.

El SEN-HS-CAP funciona según el principio de la capacitancia. Emplea electrodos o placas capacitivas que interactúan con el medio circundante (suelo, en este caso) para determinar la humedad. Las placas capacitivas del sensor interactúan con la humedad del suelo. La cantidad de agua presente en el suelo afecta la capacitancia del sensor. La humedad del suelo altera la constante dieléctrica del medio entre las placas, lo que a su vez modifica la capacitancia del sensor.

A medida que la humedad del suelo aumenta, la capacitancia del sensor también aumenta debido a la mayor constante dieléctrica del agua en comparación con el suelo seco. Esta variación en la capacitancia se correlaciona directamente con el contenido de humedad del suelo



Figura 2: Sensor de Humedad SEN-HS-CAP.

Para el segundo sensor se diseñó un circuito que pueda medir los niveles de cambio de luz. Usando un amplificador operacional (*Op-Amp*) *LM741* en configuración promediador inversor, y con un arreglo de fotorresistencia se obtienen distintas variaciones de nivel de luz.

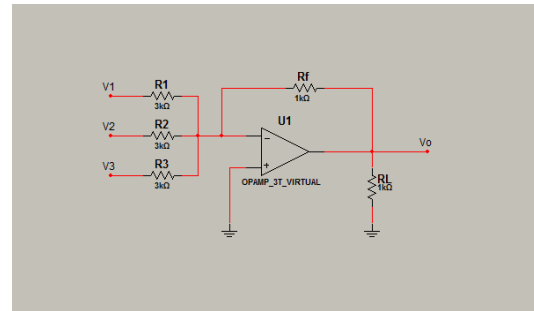


Figura 3: Configuración de Op-Amp LM741 y fotorresistencias empleadas.

El circuito promediador inversor es un tipo configuración de Amplificadores Operacionales empleados en electrónica y telecomunicaciones para medir y generar señales invertidas o con polaridades opuestas respecto a la señal de entrada. El circuito promediador inversor utiliza un amplificador operacional (*op-amp*) configurado en una configuración inversora.

En esta configuración, se utilizó el LM741. Las señales de entrada se aplican a través de una resistencia (R_n) de entrada al terminal inversor (entrada negativa) del Op-Amp. La salida del Op-Amp se toma de su terminal de salida y su terminal inversor (entrada negativa). Proyectos Electrónica (2012)

$$V_{out} = \frac{V_1 + V_2 + V_3 + \dots + V_n}{n}, \quad (1)$$

donde V_{out} es el voltaje de salida, V_1, V_2, V_3, V_n , son los voltajes de entrada, y además se debe calcular el valor de R_f , el cual está dado por:

$$R_f = \frac{R_i}{n}, \quad (2)$$

donde R_i son las resistencias de entrada que deben cumplir

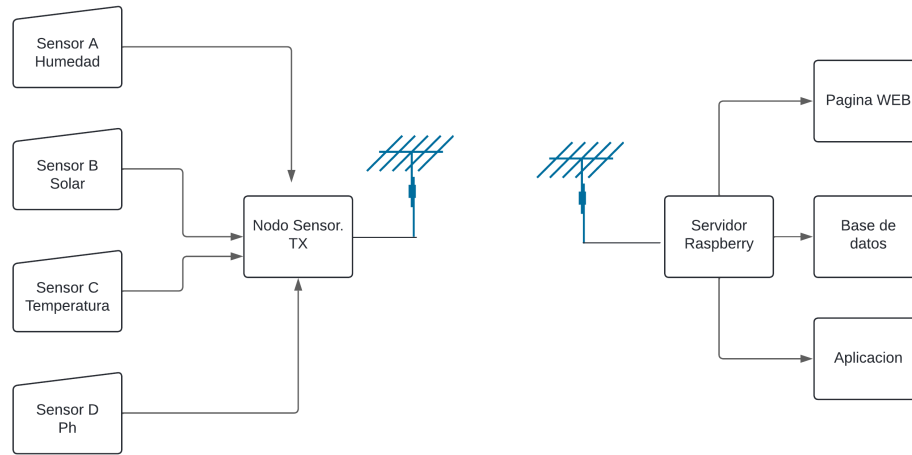


Figura 1: Diagrama a bloques del sistema SIGACI.

$$R_i = R_1 = R_2 = R_3 = R_n \text{ y } R_n > R_f.$$

Para la temperatura, se optó por usar el LM35, ya que ofrece una precisión superior en la medición de temperatura de otros sensores, al ser un sensor lineal, la lectura y el procesamiento de los datos es de forma eficiente en comparación de sensores digitales, además de tener un consumo de energía extremadamente bajo, como características se pueden mencionar:

- Salida Lineal: La salida es de 10 mV por grado Celsius. Por ejemplo, a 25°C, el LM35 produce una salida de 250 mV.
- Precisión: Generalmente tiene una precisión de $\pm 0,5^\circ\text{C}$ a temperatura ambiente.
- Rango de Temperatura: Puede medir temperaturas en un rango de -55°C a $+150^\circ\text{C}$.
- Calibración: El LM35 está calibrado directamente en grados Celsius, lo que simplifica su uso en aplicaciones que requieren mediciones de temperatura en esta escala.

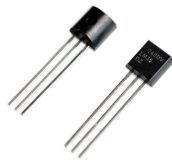


Figura 4: Sensor de temperatura LM35.

Finalmente, se decidió usar el PH-4502C debido a su precisión y facilidad de integración. Este sensor puede conectarse directamente a los pines analógicos de la unidad de procesamiento, lo que simplifica significativamente el proceso de integración, reduciendo la complejidad del sistema.

El sensor PH-4502C es un tipo de electrodo utilizado para medir el nivel de acidez o alcalinidad de una solución. Si bien existen diversos métodos para medir el pH este sensor permite obtener una lectura con precisión y fiabilidad en la medición del pH de soluciones líquidas. Este opera fundamentalmente gracias a un electrodo de vidrio especializado, que reacciona con los iones de hidrógeno H^+ en la solución, internamente este sensor contiene una solución de cloruro de potasio KCL, la cual al entrar en contacto con otra solución genera una diferencia de potencial, esta es obtenida por medio de un amplificador operacional el cual amplifica dicha diferencia y entrega un voltaje de salida. Cabe mencionar que el sensor también tiene etapas de filtrado de bajo ruido y un circuito para su calibración.

En pocas palabras el sensor PH-4502C convierte una señal de muy bajo voltaje generada por el electrodo de pH en una señal analógica utilizable mediante un circuito que incluye amplificación, filtrado, y ajustes de calibración. Este diseño permite medir de manera precisa el pH de una solución.



Figura 5: Sensor de pH PH-4502C.

En la parte de procesamiento al a ver una gran cantidad

de dispositivos disponibles se optó por utilizar la placa *ESP32-WROOM*, estas placas son diseñadas especialmente para aplicaciones de *IoT* y tecnología *wearable*. Este *SoC* (System on a Chip) tiene como principales características:

- Wi-Fi y Bluetooth
- UART, SPI, I^2C , I^2S , ETH
- ADC, DAC
- Timers
- CPU Xtensa Dual-core @ 240MHz

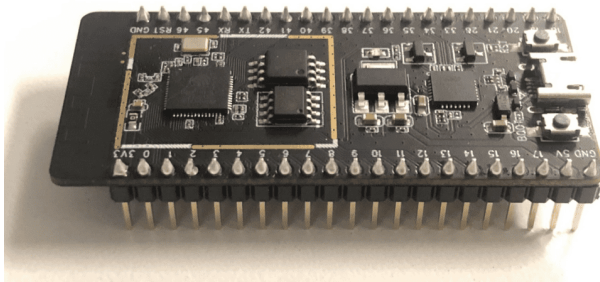


Figura 6: Unidad de Procesamiento ESP-32.

Para la parte de comunicación, se emplea el módulo LoRa E220, el cual permite ser programado por medio del protocolo *UART* (Universal Asynchronous Receiver-Transmitter) debido a que emplea el IC LLCC68, teniendo la flexibilidad de ser implementado en cualquier unidad de procesamiento, además de su bajo costo.



Figura 7: Módulo de comunicación E220.

2.2. Transmisión de Datos

La transmisión de datos a través de módulos de radio es una práctica común en diversas aplicaciones tecnológicas, desde el *IoT* hasta sistemas de comunicación industrial. La selección y configuración adecuadas de estos módulos son esenciales para

garantizar una comunicación efectiva y confiable. Cada módulo de radio opera en una frecuencia específica, que debe seleccionarse de acuerdo con las necesidades de la aplicación y las regulaciones locales. Por ejemplo, los módulos *Bluetooth* operan típicamente en la banda de 2.4 GHz Santiago y C. (2021), mientras que los módulos RF pueden utilizar bandas como 433 MHz o 915 MHz Moreno *et al.* (2019).

2.2.1. LoRa

Como se menciona, el sistema *SIGACI* emplea la tecnología *LoRa* la cual utiliza una modulación de espectro ensanchado (CSS) para transmitir datos a través del aire. Los datos se transmiten en forma de paquetes, utilizando una de las varias técnicas de modulación disponibles en *LoRa*, como *LoRa modulation* o *FSK* (Frequency Shift Keying) Rodríguez y Suero (1995). La señal del módulo LoRa e220 puede alcanzar distancias de 5 Km en entornos rurales, con una potencia de 22 dBm, según las condiciones del terreno y otros factores ambientales Ebyte (sf).

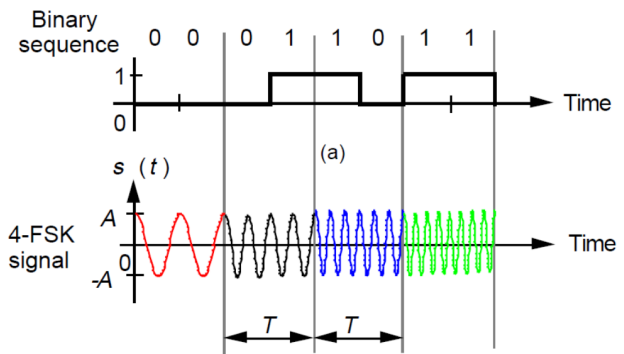


Figura 8: Diagrama de la modulación FSK a partir de una secuencia binaria.

En la Fig.8 se observa una modulación 4-FSK, lo que significa que es un FSK con cuatro símbolos diferentes ('00', '01', '10', '11'), permitiendo la transmisión de dos bits por símbolo. Project (2019)

2.2.2. UART

El sistema *SIGACI*, al contener la unidad de procesamiento ESP-32 es compatible con el módulo E220 debido a que al tener implementado el protocolo *UART* (Universal Asynchronous Receiver-Transmitter) Solectroshop (sf) permite la comunicación del módulo y la unidad de procesamiento en forma de bits. En la *UART*, los datos se envían en paquetes de bits que incluyen:

- Bit de Inicio (Start Bit): Indica el comienzo de una transmisión de datos. Es siempre un bit de valor bajo (0).
- Bits de Datos (Data Bits): La información que se está enviando. Puede ser de 5 a 9 bits de longitud.
- Bit de Paridad (Parity Bit): Opcional. Se utiliza para la detección de errores. Puede ser par, impar, o no utilizado (sin paridad).
- Bits de Parada (Stop Bits): Indican el final de la transmisión de datos. Pueden ser uno o dos bits de valor alto (1).

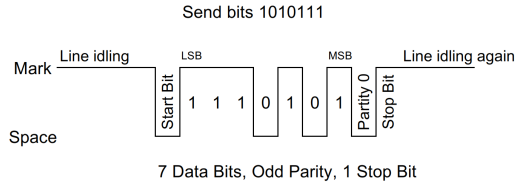


Figura 9: Datagrama del protocolo UART.

2.3. Receptor y Sistema de Procesamiento

Como receptor se emplea una Raspberry Pi 4B, la cual es un sistema embebido el cual contiene:

- CPU BCM2711 @ 1.5GHz
- Wi-Fi y Bluetooth
- UART, SPI, I²C, I²S, ETH
- ADC, DAC
- Timers
- 4, 8 o 16 GB en RAM

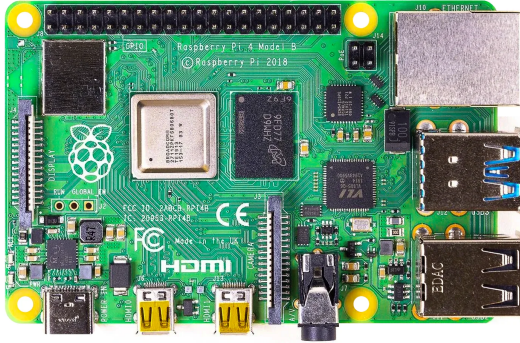


Figura 10: Datagrama del protocolo UART.

Este sistema permite el uso de diversos Sistemas Operativos basados en el kernel de Linux, para este trabajo se empleó el sistema Debian para arquitectura ARM64 y como receptor de comunicación LoRa, se empleó un módulo E220.

3. Experimentación y Resultados

En la Fig.11, se puede apreciar el prototipo del sistema *SI-GACI*, este se encuentra midiendo niveles de pH, temperatura, humedad y luminosidad de una planta. Además, se aprecia el envío de datos por medio del módulo E220 y de la unidad de procesamiento ESP32 el cual programo utilizando el lenguaje de programación C por medio del compilador GNU-GCC y la interfaz Arduino IDE.

3.1. Adquisición de datos

A partir de los sensores seleccionados, estos ingresan a los canales ADC que incluye la unidad de procesamiento ESP32, a continuación se describirá el proceso de lectura y construcción de la trama a enviar al receptor.

Para el sensor de luminosidad se obtiene la siguiente lectura de la siguiente manera:

$$l = 100 * \left(\frac{3.3V}{2^{10}} \right) (ADC_{Read}), \quad (3)$$

donde: 3.3V es el voltaje de referencia empleado, 2^{10} es la cantidad de bits que maneja el ADC, en este caso el ADC es de 10 bits y ADC_{Read} es la lectura que esta adquiriendo el ADC en ese momento.

Adicionalmente se hizo la caracterización del sensor donde finalmente, se envía el siguiente dato a partir de la siguiente regla de decisión:

$$l = \begin{cases} 0, & \text{if } 430 \geq h \geq 520 \\ 50, & \text{if } 430 \geq h \geq 350 \\ 100, & \text{if } 350 \geq h \geq 260 \end{cases} \quad (4)$$

Para el caso de la temperatura, es similar al anterior donde:

$$t = 100 * \left(\frac{3.3V}{2^{10}} \right) (ADC_{Read}), \quad (5)$$

donde: 3.3V es el voltaje de referencia empleado, 2^{10} es la cantidad de bits que maneja el ADC, en este caso el ADC es de 10 bits y ADC_{Read} es la lectura que esta adquiriendo el ADC en ese momento, también esta formula se repite para el caso del sensor de humedad.

Finalmente para la adquisición del pH se realiza la siguiente operación:

$$pH = 7 + \frac{\frac{ADC_{Read}}{2^{10}} * (V_{op} - V_n)}{V_{pH}} - K, \quad (6)$$

donde: el valor de 7 es debido a que suponemos que la solución a analizar es neutra, es decir no es ni ácido ni base, 2^{10} es la cantidad de bits que maneja el ADC, en este caso el ADC es de 10 bits y ADC_{Read} es la lectura que esta adquiriendo el ADC en ese momento, V_{op} , es el voltaje de operación, V_n es el voltaje natural, V_{pH} es el voltaje de pH y K es la calibración establecida.

3.2. Envío de Datos

Estos datos se envían por medio de la UART (8N1) a una velocidad de 9600bps al módulo E220, que es el que se encarga de transmitirlos por medio de la comunicación LoRa la cual como se mencionó tiene una frecuencia de 433MHz. La transmisión LoRa permite enviar datos a largas distancias con bajo consumo de energía de 158.49mW, lo que lo hace ideal para este proyecto de monitoreo de largo alcance.



Figura 11: Prototipo del sistema SIGACI.

3.2.1. Rango de envío

Con la finalidad de estimar el alcance del sistema SIGACI, se procedió a hacer simulaciones de los modelos de Longley-Rice y Free Space con la finalidad de determinar la distancia máxima para poder recibir datos de forma eficiente.

El modelo Longley-Rice nos permite modelar el comportamiento de la radiación electromagnética desde un punto de transmisión a través de terrenos irregulares, incluyendo edificios, utilizando el modelo de Longley-Rice, también conocido como el Modelo de Terreno Irregular (ITM). Está diseñado para frecuencias de 20 MHz a 20 GHz, distancias de 1 a 2000 km y alturas de antena de 0,5 a 3000 m. Además, calcula la pérdida de trayectoria considerando la pérdida de espacio libre, la difracción por el terreno y los obstáculos, la reflexión en la tierra, la refracción atmosférica y la dispersión troposférica MathWorks (sf).

Proporciona estimaciones de pérdida de trayecto mediante la combinación de datos físicos y empíricos.[18] Aunque el modelo es muy complejo y su implementación completa está fuera del alcance de una simple fórmula se puede calcular con la siguiente fórmula:

$$A = AFS + AA + AH + AM - A_c, \quad (7)$$

donde A es la atenuación total en dB, AFS es la atenuación por espacio libre (free-space loss), que depende de la distancia entre el transmisor y el receptor, AA es la atenuación adicional debido a la curvatura de la Tierra AH es la atenuación adicional debido a las pérdidas por obstáculos como edificios y árboles AM es la atenuación adicional debido a la dispersión multimodal de la señal y A_c es la atenuación corregida, que tiene en cuenta la variabilidad estadística de la señal.

El modelo de propagación de espacio libre es uno de los

modelos más simples y fundamentales en el estudio de la propagación de ondas de radio. Se basa en la suposición de que la señal se propaga en un entorno ideal sin obstáculos, donde la única pérdida de señal se debe a la dispersión de la energía de la onda electromagnética en el espacio.

$$FSPL_{(dB)} = 20\log_{10}(d) + 20\log(f) + 20\log\left(\frac{4\pi}{c}\right), \quad (8)$$

En el entorno de simulación *MATLab Online* nos permite observar la variedad de factores que pueden afectar la propagación de las señales de radio como en este caso en una zona específica (poner las coordenadas en la fig.), como la frecuencia de la señal, la altitud de las antenas transmisoras y receptoras, la topografía del terreno, la conductividad eléctrica del suelo, las condiciones atmosféricas y otros parámetros relevantes. Para ello se emplea dicho software para generar las siguientes simulaciones:

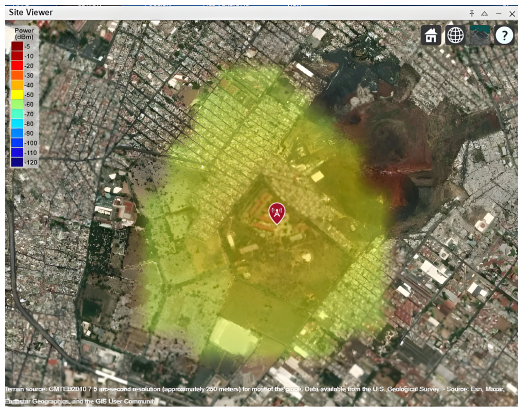


Figura 12: Modelo de radiación Longley-Rice con una potencia de radiación de -55dbm a -5dBm ubicando el TX en los laboratorios de la Universidad Autónoma de la Ciudad de México (UACM) campus San Lorenzo Tezonco (SLT). Coordenadas [19.3128513,-99.0577421]

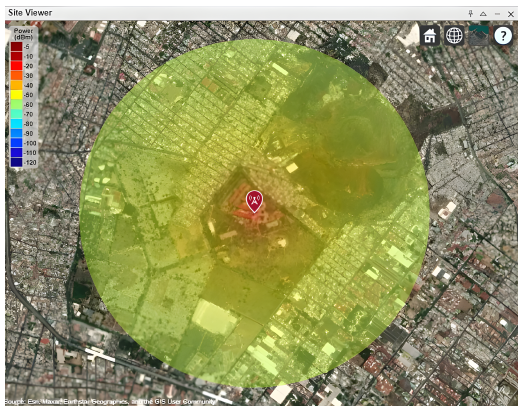


Figura 13: Modelo FreeSpace representando la cobertura de transmisión con una potencia de entre -55 dBm y -5 dBm ubicando el TX en los laboratorios de la Universidad Autónoma de la Ciudad de México (UACM) campus San Lorenzo Tezonco (SLT).Coordenadas [19.3128513,-99.0577421]]

3.3. Procesamiento y almacenamiento de datos en el servidor:

Una vez que los datos son recibidos en el servidor central, pueden ser procesados, almacenados y utilizados según sea necesario. Esto puede incluir la actualización de bases de datos, la generación de informes o la activación de acciones automáticas basadas en los datos recibidos.

Para el servidor se instaló una versión de Debian 12 y un servidor Apache v2.4.57, realizando la configuración de un servidor MySQL para alojar todos los datos obtenidos, permitiendo la gestión de dicha información para mejorar su calidad, en la Fig.14, se muestra la interfaz WEB diseñada para la consulta de datos por día o lectura por sensor.

TABLA DE SENSORES					
ID	Fecha	Humedad	Solar	Temperatura	pH
13101	14-06-2024	4	131	17.08	6.91
13102	14-06-2024	4	131	17.08	6.91
13103	14-06-2024	4	131	17.08	6.91
13104	14-06-2024	4	131	17.08	6.91
13105	14-06-2024	4	131	17.08	6.91
13106	14-06-2024	4	131	17.08	6.91
13107	14-06-2024	4	131	17.08	6.91
13108	14-06-2024	4	131	17.08	6.91
13109	14-06-2024	4	131	17.08	6.91
13110	14-06-2024	4	131	17.08	6.91
13111	14-06-2024	4	131	17.08	6.91
13112	14-06-2024	4	131	17.08	6.91
13113	14-06-2024	4	131	17.08	6.91
13114	14-06-2024	4	131	17.08	6.91
13115	14-06-2024	4	131	17.08	6.91
13116	14-06-2024	4	131	17.08	6.91
13117	14-06-2024	4	131	17.08	6.91
13118	14-06-2024	4	131	17.08	6.91
13119	14-06-2024	4	131	17.08	6.91
13120	14-06-2024	4	131	17.08	6.91

Figura 14: Entorno WEB del sistema SIGACI.

Es importante tener en cuenta la seguridad durante todo el proceso de transmisión de datos. Esto puede implicar el cifrado de los datos transmitidos mediante LoRa y el uso de protocolos de comunicación seguros en el servidor central para proteger los datos recibidos. Además, se deben implementar medidas de seguridad en los dispositivos de campo para protegerlos contra accesos no autorizados y ataques cibernéticos.

Finalmente se muestra la lista de materiales y su costo final (BOM): La implementación del sistema de monitoreo agrícola basado en *IoT* demostró ser altamente eficaz en la captura y transmisión de datos ambientales cruciales para la gestión de cultivos. Los sensores utilizados proporcionaron mediciones precisas y consistentes, cubriendo variables clave como humedad, pH, temperatura y radiación solar. La comunicación inalámbrica, soportada por el módulo *LoRa*, garantizó una transmisión confiable de los datos hacia un servidor central.

Uno de los aspectos destacados del sistema es la interfaz de usuario desarrollada, que facilita el acceso a los datos recopilados y soporta la toma de decisiones informadas por parte de los agricultores. Esta interfaz permite una visualización de las condiciones de los cultivos, lo cual es crucial para optimizar el riego y detectar posibles afectaciones tempranamente.

El sistema también incluyó la instalación de un servidor en una Raspberry Pi 4, con Debian 12 y un servidor Apache para alojar los datos obtenidos del módulo ESP32. La configuración de un servidor MySQL permitió la gestión eficiente de la información, mejorando su calidad y accesibilidad.

Las pruebas de validación y efectividad se realizaron de manera remota, confirmando el correcto funcionamiento de cada sensor y la confiabilidad en el envío de datos. En general, el sistema logró una integración exitosa de las tecnologías *IoT* para mejorar la eficiencia y la sostenibilidad en la agricultura

4. Conclusiones

El presente trabajo resalta cómo la implementación de tecnologías *IoT* puede revolucionar el sector agrícola, ofreciendo herramientas avanzadas para la gestión inteligente y sostenible de los cultivos. A través del desarrollo de un sistema de monitoreo agrícola utilizando el microcontrolador ESP32 y una variedad de sensores, se ha demostrado que es posible optimizar el uso de recursos y mejorar significativamente la productividad. La transmisión de datos mediante el módulo *LoRa* permite una monitorización remota y eficiente, brindando a los agricultores

Tabla 1: Costo de materiales empleados.

Material	Precio (USD)	Precio (MXN)	Cantidad
ESP32	10.00	180.00	2
Resistencias 10K	0.10	1.80	20
Fotoresistencias	0.39	7.00	20
Raspberry Pi	35.00	630.00	1
Módulo LoRa E220	12.00	216.00	2
Sensor de Humedad	1.67	30.00	1
Sensor de Temperatura	1.56	28.00	1
Sensor de pH	7.22	130.00	1
Shield de ESP32	2.78	50.00	1
Amplificadores Operacionales	0.39	7.00	10
Reguladores de Voltaje	0.67	12.00	5
Cables	0.50	9.00	10 metros
Soldadura Estaño	3.00	54.00	5 metros
Total	115.89	2110.00	

una visión detallada del estado de sus cultivos.

El análisis de otros proyectos similares revela que el Sistema de Gestión Agrícola con Control Inteligente (*SIGACI*) ofrece mejoras sustanciales en comparación con enfoques anteriores. Mientras que algunos proyectos dependen de tecnologías más limitadas como WiFi o SCADA, *SIGACI* utiliza una combinación de LoRa y WiFi, proporcionando una mayor cobertura y una mejor eficiencia energética. Además, la arquitectura del sistema, que incluye nodos de sensores, un Gateway, un servidor y un actuador, permite una gestión más compleja y precisa de los datos ambientales.

Las ventajas de *SIGACI* son evidentes en su capacidad para proporcionar datos facilitando la toma de decisiones informadas y oportunas. La integración de una interfaz web moderna e intuitiva mejora la accesibilidad y la interpretación de los datos, haciendo que el sistema sea más útil y eficaz para los usuarios. Asimismo, la inclusión de algoritmos de aprendizaje automático en algunos aspectos del sistema permite una optimización continua de los recursos agrícolas.

Sin embargo, se han identificado algunas desventajas en otros métodos de monitoreo agrícola que deben ser considerados. La dependencia de sistemas como SCADA y LabVIEW puede aumentar los costos y la complejidad operativa, mientras que la falta de integración con tecnologías *IoT* avanzadas limita la capacidad de estos sistemas para proporcionar un monitoreo en tiempo real y una gestión óptima de los recursos naturales. Además, la escalabilidad y la flexibilidad de los sistemas tradicionales son menores en comparación con las soluciones basadas en *IoT* como *SIGACI*.

En conclusión, la adopción de tecnologías *IoT* en la agri-

cultura representa un paso crucial hacia una producción más sostenible y eficiente. La capacidad de monitorear y controlar las condiciones ambientales de los cultivos no solo mejora los rendimientos, sino que también contribuye a la preservación de los recursos naturales. A medida que avanzamos en el desarrollo de soluciones innovadoras basadas en *IoT*, es fundamental seguir investigando y perfeccionando estos sistemas para enfrentar los desafíos emergentes en la agricultura y asegurar la seguridad alimentaria a nivel global.

Referencias

- Ebyte (s.f.). *Introduction to LoRa Spread Spectrum Technology*. Ebyte.
- Gordillo, G. y A., C. (2021). Red de sensores inalámbricos bajo protocolo lora y gestión de procesos para la analítica de datos mediante meta sistema operativo para monitoreo ambiental en invernaderos (bachelor's thesis).
- J., C. T. J. y Mañay Quinatoa, A. O. (2023). Implementación de un sistema de monitoreo de ph aplicado a un invernadero de la zona alta sector salache (bachelor's thesis, ecuador: Latacunga: Universidad técnica de cotopaxi (utc)).
- MathWorks (s.f.). `rfprop.longleyrice`. En *MathWorks*. Recuperado el 14 de junio de 2024, de.
- Moreno, C. A. T., Blasco, R., Nebra, R. C., y Marco, Á. M. (2019). Evaluación de la modulación lora en la banda de 433 mhz para monitorización en exteriores. *DYNA New Technologies*, 6(1):11.
- Project, F. (2019). `iembre 12`. `teleco in a nutshell v8.4`: Modulación por desplazamiento de frecuencia. `flu project`.
- Proyectos Electrónica, . (2012). `zo 26`. [Entrada de blog]. Blogspot. Recuperado el 16 de junio de 2024, de.
- Rodríguez, J. L. y Suero, S. C. (1995). *Modulación de señales digitales*. Universidad Politécnica de Sevilla, Departamento de Tecnología Electrónica.
- Santiago, S. y C., J. (2021). *Integración de sensores y diseño de enlace de datos punto a punto de largo alcance, basado en tecnología LoRa y bluetooth*. Tesis doctoral, Universidad de San Carlos de Guatemala.
- Serna, S., Montoya, B., y Osorio, W. (2012). Monitoreo del ph y fósforo soluble en la rizosfera de árboles de aguacate en el norte y oriente antioqueño. *Suelos Ecuatorianos*, 42(2):186–190.
- Solectroshop (s.f.). Diferencias de los protocolos de comunicación uart vs i2c vs spi. `solectroshop`. Recuperado el 16 de junio de 2024, de.



INSTITUTO POLITÉCNICO NACIONAL



Unidad Profesional Interdisciplinaria de Ingeniería
Campus Hidalgo

Otorga la presente

CONSTANCIA

a:

Dr. Eduardo Ramos Diaz; Dr. José Agustín Almaraz Damián y José Ángel Quiroz Valdez

Por haber participado como ponentes con el tema:

"TRANSFORMACIÓN AGRÍCOLA: IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO IOT PARA LA GESTIÓN INTELIGENTE DE CULTIVOS."

En el marco del "2.º Congreso Nacional Multidisciplinario de Innovación, Ciencia y Tecnología (CoNMICyT)" que se efectuó en la Unidad Profesional Interdisciplinaria de Ingeniería Campus Hidalgo los días 25, 26 y 27 de septiembre de 2024.

"La Técnica al Servicio de la Patria"

Dr. Domingo Hernández García

Director Interino

San Agustín Tlaxiaca, Hidalgo, a 27 de septiembre de 2024

La correcta instalación y configuración de un servidor es un proceso fundamental para asegurar el óptimo desempeño y la fiabilidad del sistema que se implementará. En este apéndice B, se detallan los pasos necesarios para la instalación del servidor web.

Este conjunto de pasos garantiza que el servidor esté correctamente instalado y optimizado para llevar a cabo las tareas de monitoreo y procesamiento de datos de manera eficiente y segura.

Apéndice B

B.1. INSTALACION SERVIDOR

La instalación de un servidor LAMP (Linux, Apache, MySQL, PHP) en una Raspberry Pi permite configurar un entorno de desarrollo robusto para aplicaciones web. A continuación, se describen los pasos necesarios para la instalación y configuración.

Actualización del Sistema

Antes de iniciar la instalación, es fundamental actualizar el sistema operativo. Ejecute el siguiente comando:

```
sudo apt update && sudo apt upgrade
```

Instalación de Apache2

Para instalar el servidor web Apache2, utilice el siguiente comando:

```
sudo apt install apache2
```

Directorio de Páginas Web

Acceda al directorio donde se guardarán las páginas web:

```
cd /var/www/html
```

Instalación de PHP

Para instalar PHP, ejecute el siguiente comando:

```
sudo apt install php
```

Archivo de Prueba de PHP

Para verificar la instalación de PHP, cree un archivo de prueba. Utilice el siguiente comando para crear y editar el archivo:

```
sudo nano /var/www/html/p.php
```

Dentro del archivo, escriba el siguiente código:

```
<?php echo "¡Está vivo!!!!"; ?>
```

Reinicio del Servicio Apache2

Reinicie el servicio de Apache2 para aplicar los cambios:

```
sudo service apache2 restart
```

B.2. Instalación de MySQL

Para instalar el servidor de bases de datos MariaDB y los módulos necesarios para PHP, ejecute el siguiente comando:

```
sudo apt install mariadb-server php-mysql
```

Reinicie el servicio de Apache2 nuevamente:

```
sudo service apache2 restart
```

B.3. Configuración Inicial de MySQL

Después de instalar MySQL, realice la configuración inicial ejecutando:

```
sudo mysql_secure_installation
```

B.3.1. Creación de un Nuevo Usuario

Para crear un nuevo usuario en MySQL, ingrese a la consola de MySQL como root:

```
sudo mysql --user=root --password
```

Luego, ejecute los siguientes comandos en la consola de MySQL:

```
create user 'admin'@'%' identified by 'your_password';  
grant all privileges on *.* to 'admin'@'localhost';  
FLUSH PRIVILEGES;  
exit;
```

B.4. Instalación de phpMyAdmin

Para facilitar la gestión de bases de datos, instale phpMyAdmin utilizando el siguiente comando:

```
sudo apt install phpmyadmin
```

B.5. Enlace de phpMyAdmin con MySQL

Habilite el módulo mysqli para que phpMyAdmin funcione correctamente:

```
sudo phpenmod mysqli  
sudo service apache2 restart
```

B.6. Configuraciones Adicionales en MySQL

Para permitir conexiones remotas, edite el archivo de configuración de MySQL:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Busque la línea `bind-address` y cámbiela a:

```
bind-address = 0.0.0.0
```

B.7. Ajustes en la Configuración de PHP

Para aumentar las capacidades del servidor, ajuste los siguientes parámetros en el archivo de configuración de PHP, que se encuentra en `/etc/php/7.3/apache2/php.ini`:

- Para aumentar el tiempo máximo de ejecución:

```
max_execution_time = 3000
```

- Para aumentar el tamaño de datos POST:

```
post_max_size = 16M
```

- Para aumentar el tamaño máximo de archivos:

```
upload_max_filesize = 20M
```

- Para aumentar el tiempo de entrada de datos:

```
max_input_time = 1000
```

B.8. CODIGO PHP

```
<?php

$temperatura = $_GET['temp'];
$humedad = $_GET['hum'];
$namenodo = $_GET['nodo'];

echo "La temperatura es: ".$temperatura." <br>La humedad es: ".$humedad."

<br>Nodo es: ".$namenodo;

$usuario = "root";
$contrasena = "";
$servidor = "localhost";
$basededatos = "sensor";

$conexion = mysqli_connect( $servidor, $usuario, "" ) or die

("No se ha podido conectar al servidor de Base de datos");

$db = mysqli_select_db( $conexion, $basededatos ) or die

( "No se ha podido seleccionar la base de datos" );

$fecha = time();
$consulta = "INSERT INTO datosb (fecha, temperatura, humedad, sensores)

VALUES ( ".$fecha.", ".$temperatura.", ".$humedad.", ".$namenodo." )";

$resultado = mysqli_query( $conexion, $consulta );

?>
```

Pseudocódigo SEN-HS-CAP para leer el valor analógico:

```
// Definir constantes y variables
constante sensorPin = 32 // Pin analógico donde se conecta el sensor
entero sensorValue = 0
```



```
// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
}

// Función principal que se ejecuta en un bucle
función loop() {
    sensorValue = Leer valor analógico del sensor en sensorPin
    Imprimir "Valor del sensor en suelo seco: " + sensorValue en monitor serial
    Esperar 1000 milisegundos // Esperar 1 segundo antes de la siguiente lectura
}
```

Lectura de temperatura básica LM35

Usando el siguiente Pseudocódigo para leer la temperatura:

```
// Definir constantes y variables
constante lm35Pin = 33 // Pin analógico donde está conectado el LM35

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
}

// Función principal que se ejecuta en un bucle
función loop() {
    entero sensorValue = Leer valor analógico del LM35 en lm35Pin
    flotante voltage = sensorValue * (5.0 / 1023.0) // Convierte el valor a voltaje
    flotante temperature = voltage * 100 // Convierte el voltaje a temperatura en °C

    Imprimir "Temperatura: " + temperature en monitor serial
    Imprimir " °C" en monitor serial

    Esperar 1000 milisegundos // Esperar 1 segundo antes de la siguiente lectura
}
```

B.9. Pseudocódigo PH4502C

```
// Incluir la biblioteca para el sensor PH4502C
Incluir <ph4502c_sensor.h>

// Definir los pines del sensor
```

```

definir PH4502C_PH_LEVEL_PIN como 35 // Pin del nivel de pH
definir PH4502C_TEMP_PIN como 34 // Pin de temperatura

// Crear una instancia del sensor PH4502C
crear objeto ph4502 de tipo PH4502C_Sensor
con PH4502C_PH_LEVEL_PIN y PH4502C_TEMP_PIN

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
    Imprimir "Iniciando sensor PH4502C..."

    // Inicializar la instancia de PH4502
    ph4502.init()
}

// Función principal que se ejecuta en un bucle
función loop() {
    // Leer la temperatura del sensor
    Imprimir "Lectura de temperatura: " + ph4502.read_temp()

    // Leer el nivel de pH por promedio
    Imprimir "Lectura del nivel de pH: " + ph4502.read_ph_level()

    // Leer un único valor de pH
    Imprimir "Lectura única del nivel de pH: " + ph4502.read_ph_level_single()

    Esperar 1000 milisegundos // Esperar 1 segundo antes de la siguiente lectura
}

```

B.10. biblioteca necesaria para el módulo EBYTE22

```

Incluir <ph4502c_sensor.h> // 0 la biblioteca correspondiente según el contexto

// Definir constantes para pines
definir PIN_RX como 14 // Pin de recepción
definir PIN_TX como 12 // Pin de transmisión
definir PIN_M0 como 5 // Pin de modo 0
definir PIN_M1 como 4 // Pin de modo 1
definir PIN_AX como 13 // Pin AUX

```

```

// Crear objeto SoftwareSerial para conectarse al módulo EBYTE22
crear objeto E22Serial de tipo SoftwareSerial con PIN_TX y PIN_RX,

sin invertir lógica

// Crear instancia de la clase EBYTE22
crear objeto E22 de tipo EBYTE22 con E22Serial, PIN_M0, PIN_M1, PIN_AX

// Definir variables de datos a enviar
byte testByte = 123 // Variable de tipo byte
String testString = "TEST" // Variable de tipo cadena

// Definir estructura de datos
estructura myStruct {
    entero days = 365
    flotante pi = 3.14
    largo lightSpeed = 299792458
    caracter ch = 'X'
} testStruct

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
    E22Serial.begin(9600) // Inicializar la conexión con el módulo

    Imprimir "Iniciando Emisor"
    Esperar 200 milisegundos

    // Inicializar el módulo EBYTE22
    si (E22.init()) {
        Imprimir "inicialización OK"
    } sino {
        Imprimir "Error de inicialización"
    }
}

// Función principal que se ejecuta en un bucle
función loop() {
    si (E22.getBusy()) { // Verificar si el módulo está ocupado
        E22.completeTask(5000) //

        Esperar hasta que se libere, con un tiempo de espera de 5 segundos
    }
}

```

```

// Enviar la estructura
E22.sendStruct(&testStruct, sizeof(testStruct))

// Enviar la dirección de la estructura y su tamaño

// Enviar un byte
E22.sendByte(testByte) // Enviar el byte

// Enviar una cadena
E22Serial.print(testString) // Enviar la cadena al módulo

E22.flush() // Esperar a que se complete la transmisión de datos

Esperar 5000 milisegundos // Esperar un poco antes de la siguiente iteración
}

```

B.11. biblioteca necesaria para el módulo EBYTE

```

Incluir <SoftwareSerial.h>
Incluir "EBYTE22.h"

// Definir constantes para pines
definir PIN_TX como 2
definir PIN_RX como 3
definir PIN_M0 como 4
definir PIN_M1 como 5
definir PIN_AX como 6

// Crear objeto SoftwareSerial para conectarse al módulo EBYTE crear objeto
E22Serial de tipo SoftwareSerial con PIN_TX y PIN_RX,

sin invertir lógica

// Crear instancia de la clase EBYTE22
crear objeto E22 de tipo EBYTE22 con E22Serial, PIN_M0, PIN_M1, PIN_AX

// Definir variables para almacenar los valores recibidos
byte testByte
String testString

```

```

// Definir estructura de datos
estructura myStruct {
    entero days
    flotante pi
    largo lightSpeed
    caracter ch
} testStruct

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
    E22Serial.begin(9600) // Inicializar la conexión con el módulo

    Imprimir "Iniciando Emisor"
    Esperar 200 milisegundos

    // Inicializar el módulo EBYTE22
    si (E22.init()) {
        Imprimir "inicialización OK"
    } sino {
        Imprimir "Error de inicialización"
    }
}

// Función principal que se ejecuta en un bucle
función loop() {
    // Comprobar si hay datos no leídos
    si (E22.available()) {
        // Leer la estructura
        E22.getStruct(&testStruct, sizeof(testStruct))
        // Leer la estructura y su tamaño

        // Leer un byte
        testByte = E22.getBytes()

        // Leer una cadena
        testString = E22Serial.readString()
        // Leer la cadena desde el objeto SoftwareSerial

        // Mostrar los datos en el monitor serial
        Imprimir "-----"
        Imprimir "Estructura:"
        Imprimir "Días en el año: " + String(testStruct.days)
        Imprimir "Número Pi: " + String(testStruct.pi)
    }
}

```

```

    Imprimir "Velocidad de la luz m/s: " + String(testStruct.lightSpeed)
    Imprimir "Símbolo: " + String(testStruct.ch)
    Imprimir "-----"

    Imprimir "Byte: " + String(testByte)
    Imprimir "Cadena: " + testString

    // Comprobar y mostrar el nivel de señal RSSI si está habilitado
    si (E22.getRSSIAmbient() == RSSI_ENABLE) {
        Imprimir "Nivel de señal: " + String(E22.getRSSI(RSSI_LAST_RECEIVE))
        Imprimir "Nivel de ruido: " + String(E22.getRSSI(RSSI_AMBIENT))
    } sino {
        Imprimir "Función RSSI deshabilitada"
    }
}

Esperar 1000 milisegundos // Esperar un poco antes de la siguiente iteración
}

```

B.12. pseudocódigo TX

```

// Incluir bibliotecas necesarias
Incluir <SoftwareSerial.h>
Incluir "EBYTE22.h"
Incluir <ph4502c_sensor.h>

// Definir pines de sensores y módulos
definir PH4502C_TEMP_PIN como 0

definir PIN_M0 como 19 // azul
definir PIN_M1 como 18 // verde
definir PIN_RX como 17 // amarillo
definir PIN_TX como 16 // naranja
definir PIN_AX como 15 // rojo
definir Sensor_Hum como 12
definir PH4502C_PH_LEVEL_PIN como 35
definir Sensor_Solar como 36
definir Sensor_Temp como 25

// Crear objeto SoftwareSerial para conectarse al módulo EBYTE
crear objeto E22Serial de tipo SoftwareSerial con PIN_TX y PIN_RX.

```

sin invertir lógica

```
// Crear instancia de la clase EBYTE22
crear objeto E22 de tipo EBYTE22 con E22Serial, PIN_M0, PIN_M1, PIN_AX

// Definir estructura para almacenar datos de los sensores
estructura myStruct {
    entero humedad
    flotante Ph
    flotante solar
    flotante temperatura
} testStruct

// Crear objeto para el sensor de pH
crear objeto ph4502 de tipo
PH4502C_Sensor con PH4502C_PH_LEVEL_PIN y PH4502C_TEMP_PIN

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
    E22Serial.begin(9600) // Inicializar conexión con el módulo
    Imprimir "Starting Sender"
    Esperar 200 milisegundos

    // Inicializar el módulo EBYTE22
    si (E22.init()) {
        Imprimir "init OK"
    } sino {
        Imprimir "init Error"
    }
}

// Función principal que se ejecuta en un bucle
función loop() {
    // Comprobar si el módulo está ocupado
    si (E22.getBusy()) {
        E22.completeTask(5000) // Esperar a que el módulo esté libre
    }

    Imprimir "Estructura:"

    // Leer humedad del sensor
    serial_humedad = analogRead(Sensor_Hum)
    humedad = serial_humedad * (100.0 / 4095) // Convertir lectura a porcentaje
}
```

```

// Actualizar valor de humedad en la estructura
testStruct.humedad = humedad
Imprimir "Valor Humedad: " + humedad + "%"

// Leer radiación solar del sensor
serial_solar = analogRead(Sensor_Solar)
solar = serial_solar * (100.0 / 4095) // Convertir lectura a porcentaje

// Actualizar valor de solar en la estructura
testStruct.solar = solar
Imprimir "Valor solar: " + solar + "%"

// Leer temperatura del sensor
serial_temperatura = analogRead(Sensor_Temp)
temperatura = serial_temperatura * (3.30 / 4095) // Convertir lectura a °C

// Actualizar valor de temperatura en la estructura
testStruct.temperatura = temperatura
Imprimir "Valor temperatura: " + temperatura + "°C"

// Leer pH del sensor
Ph = ph4502.read_ph_level()

// Actualizar valor de pH en la estructura
testStruct.Ph = Ph
Imprimir "pH: " + Ph

// Enviar la estructura con los datos de los sensores
E22.sendStruct(&testStruct, sizeof(testStruct))

Esperar 1000 milisegundos // Esperar un segundo antes de la siguiente lectura
}

```

B.13. PseudoCódigo RX

```

// Incluir bibliotecas necesarias
Incluir <SoftwareSerial.h>
Incluir "EByte22.h"
Incluir <MySQL_Connection.h>
Incluir <MySQL_Cursor.h>

```



```

Incluir <Ethernet.h>
Incluir <ESP8266WiFi.h>

// Definir pines para los sensores y módulos
definir PIN_RX como 14 // D5
definir PIN_TX como 12 // D6
definir PIN_M0 como 5 // D1
definir PIN_M1 como 4 // D2
definir PIN_AX como 13 // D7
definir N como 0
definir h como 0

// Definir credenciales de WiFi y dirección del servidor
const char *ssid = "Totalplay-6A9E"
const char *password = "Angel343"
const char *host = "192.168.1.80" // IP del servidor
const int port = 80 // Puerto del servidor
const int watchdog = 5000 // Frecuencia del watchdog
unsigned long previousMillis = millis()

// Crear objeto SoftwareSerial para conectarse al módulo EBYTE
crear objeto E22Serial de tipo SoftwareSerial con PIN_TX y PIN_RX

sin invertir lógica

// Crear instancia de la clase EBYTE22
crear objeto E22 de tipo EBYTE22 con E22Serial, PIN_M0, PIN_M1, PIN_AX

// Definir estructura para almacenar datos de los sensores
estructura myStruct {
    entero humedad
    flotante ph
    flotante solar
    flotante temperatura
} testStruct

// Función de configuración inicial
función setup() {
    Inicializar comunicación serial a 9600 bps
    E22Serial.begin(9600) // Inicializar conexión con el módulo
    Imprimir "Comenzando Receptor"
    Esperar 200 milisegundos

    // Conectar a la red WiFi

```

```

WiFi.begin(ssid, password)
mientras (WiFi.status() != WL_CONNECTED) {
    Esperar 500 milisegundos
    Imprimir "No hay conexion en el Modem."
}

// Inicializar el módulo EBYTE22
si (E22.init()) {
    Imprimir "Inicialización correcta"
} sino {
    Imprimir "Error de inicialización"
}

Imprimir ""
Imprimir "WiFi connected"
Imprimir "IP address: "
Imprimir WiFi.localIP()
Esperar 700 milisegundos
}

// Función principal que se ejecuta en un bucle
función loop() {
    // Leer datos del módulo EBYTE
    E22.getStruct(&testStruct, sizeof(testStruct))

    // Imprimir valores de la estructura en el monitor serial
    Imprimir "Valor humedad: " + testStruct.humedad
    Imprimir "Valor solar: " + testStruct.solar
    Imprimir "Valor temperatura: " + testStruct.temperatura
    Imprimir "pH: " + testStruct.ph

    // Comprobar el tiempo para el watchdog
    unsigned long currentMillis = millis()
    si (currentMillis - previousMillis > watchdog) {
        previousMillis = currentMillis
        WiFiClient client

        // Intentar conectar al servidor
        si (!client.connect(host, port)) {
            Imprimir "Fallo al conectar"
            retornar
        }

        // Construir la URL para enviar los datos

```

```

String url = "/hola.php?hum="
url += testStruct.humedad
url += "&sol="
url += testStruct.solar
url += "&temp="
url += testStruct.temperatura
url += "&ph="
url += testStruct.ph

// Enviar petición al servidor
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close \r\n\r\n")

// Esperar respuesta del servidor
unsigned long timeout = millis()
mientras (client.available() == 0) {
    si (millis() - timeout > 5000) {
        Imprimir ">>> Client Timeout !"
        client.stop()
        retornar
    }
}

// Leer la respuesta del servidor
mientras (client.available()) {
    String line = client.readStringUntil('\r')
    Imprimir line
}
}
Esperar 800 milisegundos // Esperar un tiempo antes de la próxima iteración
}

```

B.14. Pseudocódigo para la Instalación y Configuración de un Servidor Web

La instalación de un servidor LAMP (Linux, Apache, MySQL, PHP) en una Raspberry Pi permite configurar un entorno de desarrollo robusto para aplicaciones web. A continuación, se describen los pasos necesarios para la instalación y configuración.

Algorithm 1 Instalación y Configuración de un Servidor Web

- 1: **Inicio**
 - 2: Ejecutar “sudo apt install apache2”
 - 3: CambiarDirectorio “/var/www/html”
 - 4: Ejecutar “sudo apt install php”
 - 5: Ejecutar “sudo nano /var/www/html/p.php”
 - 6: Escribir en el archivo “¡?php echo '¡Está vivo!!!!'; ?¿”
 - 7: Ejecutar “sudo service apache2 restart”
 - 8: Ejecutar “sudo apt install mariadb-server php-mysql”
 - 9: Ejecutar “sudo service apache2 restart”
 - 10: Ejecutar “sudo mysql_secure_installation”
 - 11: Ejecutar “sudo mysql –user=root –password”
 - 12: Ejecutar “create user 'admin' @’”
 - 13: Ejecutar “grant all privileges on *.* to 'admin' @'localhost';”
 - 14: Ejecutar “FLUSH PRIVILEGES;”
 - 15: Ejecutar “exit;”
 - 16: Ejecutar “sudo apt install phpmyadmin”
 - 17: Ejecutar “sudo phpenmod mysqli”
 - 18: Ejecutar “sudo service apache2 restart”
 - 19: Ejecutar “sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf”
 - 20: Cambiar “bind-address” a “0.0.0.0”
 - 21: Ejecutar “sudo nano /etc/php/7.3/apache2/php.ini”
 - 22: Establecer “max_execution_time = 3000”
 - 23: Establecer “post_max_size = 16M”
 - 24: Establecer “upload_max_filesize = 20M”
 - 25: Establecer “max_input_time = 1000”
 - 26: **Fin**
-

INICIO

CONEXIÓN A BASE DE DATOS:

CONECTAR a la base de datos 'sensor' con usuario 'root' y contraseña vacía.

INICIO DEL DOCUMENTO HTML:

ESTABLECER el título como "Tabla de sensores".

INCLUIR el archivo CSS 'sensor.css'.

MOSTRAR EL TÍTULO:

ESCRIBIR "TABLA DE SENSORES".

CREAR FORMULARIO DE BÚSQUEDA:

PARA cada tipo de búsqueda (id, sensores, temperatura, humedad):

CREAR un campo de entrada para el número correspondiente.

CREAR un botón de envío para cada tipo de búsqueda.

CREAR un botón para mostrar todos los registros.

INICIO DE LA TABLA DE SENSORES:

DEFINIR encabezados de la tabla: id, fecha, temperatura, humedad, sensores.

OBTENER RESULTADOS DE LA BASE DE DATOS:

INICIALIZAR 'tempStyle' como cadena vacía.

INICIALIZAR 'keyword' como cadena vacía.

INICIALIZAR 'iteration' como 0.

SI se presiona el botón 'search-id':

OBTENER 'keyword-id' del formulario.

PREPARAR consulta SQL para seleccionar datos donde id = keyword.

SINO SI se presiona 'search-sensores':

OBTENER 'keyword-sensores' del formulario.

PREPARAR consulta SQL para seleccionar datos donde sensores = keyword.

SINO SI se presiona 'search-temp':

OBTENER 'keyword-temp' del formulario.

PREPARAR consulta SQL para seleccionar datos donde temperatura = keyword.

SI la temperatura es menor a 11:

ESTABLECER 'tempStyle' a fondo azul y texto blanco.

SINO SI la temperatura está entre 11 y 26:

ESTABLECER 'tempStyle' a fondo gris claro.

SINO:

ESTABLECER 'tempStyle' a fondo rojo y texto blanco.

SINO SI se presiona 'search-hume':

OBTENER 'keyword-hume' del formulario.

PREPARAR consulta SQL para seleccionar datos donde humedad = keyword.

SINO (si no se presionó ningún botón de búsqueda):

PREPARAR consulta SQL para seleccionar todos los datos.

EJECUTAR la consulta SQL y OBTENER los resultados.

MONTAR RESULTADOS EN LA TABLA:

MIENTRAS haya filas en los resultados:

INCREMENTAR 'iteration'.

OBTENER datos de la fila actual.

MOSTRAR la fila en la tabla con estilo de temperatura aplicado.

SI no se encontró ningún resultado (keyword vacío o iteration = 0):

MOSTRAR mensaje "No hay resultados" con una imagen de no resultados.

FIN DEL DOCUMENTO HTML.