



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RIEGO IOT PARA SEMBRIOS
DE MAÍZ**

Trabajo de titulación previo a la obtención del
título de Ingeniera Electrónica

AUTORA: MARTHA ESTEFANNY ASTUDILLO CASTILLO

TUTOR: ING. JUAN DIEGO JARA SALTOS, MgT.

Cuenca - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Martha Estefanny Astudillo Castillo con documento de identificación N° 1104778780, manifiesto que:

Soy la autora y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 14 de febrero del 2024

Atentamente,



Martha Estefanny Astudillo Castillo

1104778780

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

Yo, Martha Estefanny Astudillo Castillo con documento de identificación N° 1104778780, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del Proyecto técnico: “Diseño e implementación de un sistema de riego IOT para sembríos de maíz”, el cual ha sido desarrollado para optar por el título de: Ingeniera Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Cuenca, 14 de febrero del 2024

Atentamente,



Martha Estefanny Astudillo Castillo

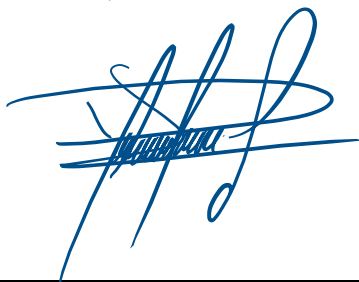
1104778780

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Diego Jara Saltos con documento de identificación N° 0103543658, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACION DE UN SISTEMA DE RIEGO IOT PARA SEMBRIOS DE MAÍZ, realizado por Martha Estefanny Astudillo Castillo con documento de identificación N° 1104778780, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 14 de febrero del 2024

Atentamente,



Ing. Juan Diego Jara Saltos, MgT.

0103543658

AGRADECIMIENTO

Agradezco en primer lugar a Dios por permitirme alcanzar esta significativa meta en mi vida, y por ser mi compañía constante en cada desafío que ha surgido en mi camino tanto en el ámbito universitario como personal.

Quiero expresar mi más profundo agradecimiento a mis padres, Juan, Martha, Rosario y así como a mi hermana Mishel. Mi camino ha sido guiado por sus valores, su apoyo incondicional y su constante esfuerzo.

No puedo pasar por alto el recuerdo de mi tío, Germán Castillo, quien, aunque ya no está entre nosotros, sé que desde el cielo siente un inmenso orgullo por mi logro. Le agradezco por haber confiado en mí y en mi potencial.

Un agradecimiento especial a mi estimado tutor, el Ing. Juan Diego Jara, por su apoyo absoluto y por confiar en mí ante cualquier adversidad. Agradezco su tiempo y sus valiosos conocimientos que han sido fundamentales en este proceso.

También quiero expresar mi gratitud a mis amigos que han convertido mi camino universitario en una travesía llena de bendiciones. A mi estimada Ing. Dianita, Ing. Jonnathan, Ing. Andrés, Ing. Xavier, Ing. Steven, Juan Gonzales, Santiago Aguirre, Ernesto Jimbo que han contribuido y han estado en los buenos y malos momentos en este viaje de mi vida. ¡Gracias a todos por ser parte de este logro!

¡Con gratitud y cariño!

Martha Estefanny Astudillo Castillo

DEDICATORIAS

Dedico este logro a Dios y a la Virgen del Cisne, por ser mi eterna guía y constante soporte a lo largo de este arduo trayecto. Sus bendiciones y protección han sido mi faro en cada desafío y mi aliento en cada logro alcanzado.

A mis padres, Juan Castillo, Martha Abendaño y Rosario Castillo quienes han sido mi guía constante, mi apoyo inquebrantable y el amor que ilumina mi vida. Gracias por ser mis pilares y por inspirarme a alcanzar cada meta que me he propuesto, a mi abuela, Delia Viñamagua, quienes han transmitido sabiduría y amor de generación en generación.

Finalmente, a mi hermana, mi eterna compañera, mi princesa. Tu cariño y aliento han sido un motor constante en mi viaje. A mis amigos, quienes han compartido risas, lágrimas y momentos inolvidables en este trayecto. Su amistad es un regalo que atesoro profundamente.

Martha Estefanny Astudillo Castillo

ÍNDICE GENERAL

AGRADECIMIENTO	I
DEDICATORIAS	II
ÍNDICE GENERAL	III
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	IX
RESUMEN	XI
ABSTRACT	XII
INTRODUCCIÓN	XIII
ANTECEDENTES DEL PROBLEMA DE ESTUDIO	XIV
JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)	XV
OBJETIVOS	XVI
OBJETIVO GENERAL	XVI
OBJETIVOS ESPECÍFICO	XVII
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE	1
FUNDAMENTACIÓN TEÓRICA	1
1.1 El maíz y su origen.....	1
1.1.1 Importancia del maíz en el mundo y en el Ecuador	1
1.1.2 Clasificación botánica del maíz.....	1
1.1.3 Importancia del maíz en la parroquia de Vilcabamba.....	2
1.1.4 Características botánicas del maíz.....	2
1.1.5 Variables climáticas que afectan la producción del maíz	4
1.1.6 Tipos suelo y sus nutrientes	5
1.1.7 Consideraciones de humedad y temperatura de la planta	6
1.1.8 Evaporaciones y transpiración	6

1.2 Riego	7
1.2.1 Sistema de riego	7
1.2.2 Tipos de riego.....	7
1.3 IOT (Internet de las cosas)	8
1.3.1 Tecnologías lpwan.....	8
1.3.2 Lora	8
1.3.4 Aplicaciones de la tecnología Lora	9
1.3.5 Modulación radio Lora.....	9
1.3.6 Agricultura inteligente.....	9
1.4 Redes de sensores aplicadas en la agricultura	10
1.4.1 Sensores.....	10
1.4.2 Sensor de temperatura	11
1.4.3 Sensores de humedad del suelo	12
1.5 Tegnologias en el sector agricola	14
1.5.1 Tipos de teganologias en el sector agricola	14
1.6 Arquitecturas de sistemas de redes en el sector agricola	16
1.6.1 Arquitectura de sensores y dispositivos IoT (Internet de las Cosas).....	16
1.6.2 Arquitectura de sistemas de información geográfica (SIG)	17
1.6.3 Arquitectura de redes de comunicación	18
1.6.4 Arquitectura de robótica agrícola.....	19
1.6.5 Arquitectura de blockchain	20

1.6.6	Arquitectura de sistemas de gestión agrícola	20
1.7	Dispósitos a utilizar en el riego de maíz por asperción utilizando IoT	21
1.7.1	Sensores de temperatura DS18B20.....	21
1.7.2	Sensores de humedad de suelo HD-38 higometro	21
1.7.3	Placa Wifi LoRa de Heltec.....	22
1.7.4	Heltec WiFi LoRa V3	24
1.7.5	Módulo rele 5v	25
1.7.6	Válvula electromagnética de 12 V	26
1.7.6	Cinta de aislante 19mm x 18m truper expert	27
1.7.7	Fuente conmutada 12V/10/P	27
1.7.8	Manguera polietileno Negra CED-40 de 1/2" 13 MM.....	28
1.7.9	Aspersor plastico Excel Wobbler 1/2.....	29
1.7.10	Caja de derivación de 105x105x55mm.....	30
1.7.11	Caja de derivación de dimensiones: 80x80x45mm.....	31
1.7.12	Cable de cobre aislado paralelo tipo SPT 300V	31
1.8	Aplicación web.....	32
1.8.1	Modelo de cliente-servidor.....	32
1.8.2	Servidor Tcp.....	32
1.8.3	Amazon Elastic Compute Cloud (Amazon EC2).....	32
1.8.4	Service web REST (Representational State Transfer)	33

1.8.5 Sencha Ext JS	33
CAPÍTULO 2: DISEÑO DEL SISTEMA DE RIEGO POR ASPERCIÓN	33
2.1 Ubicación del proyecto de riego por aspersión	33
2.2 Líneas de vista del terreno sembrado	34
2.3 Línea de vista de la casa del usuario y el terreno Lora.....	35
2.4 Esquema del sistema de riego por aspersión	35
2.5 Diagrama de bloques para calcular el número de nodos de transmisión y recepción, así como el número de sensores de humedad y temperatura	36
2.6 Herramientas de software	37
2.6.1 Embarcadero Delphi.....	37
2.6.2 Arduino IDE.....	38
2.7 Nodo transmisor	38
2.8 Diseño PCB para el nodo transmisor.....	41
2.9 Nodo receptor	43
2.10 Diseño de la aplicación web.....	45
2.10.1 Especificaciones para el desarrollo de la aplicación web	45
2.10.2 Diagrama de bloques de la aplicación web	45
2.10.3 Base de datos	46
2.10.4 Interfaz gráfica de la aplicación web.....	51
CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS	53
3.1 Armado de nodo transmisor	53
3.2 Armado del nodo receptor.....	58
3.3 Ubicación de los sensores de temperatura y humedad.....	59
3.4 Presupuesto.....	60

3.5 Ingreso a la aplicación web	61
3.6 Resultados	61
3.6.1 Activación y desactivación del riego manual.....	67
CAPÍTULO 4: CONCLUSIONES.....	68
CAPÍTULO 5: RECOMENDACIONES.....	69
TRABAJOS FUTUROS.....	70
REFERENCIAS BIBLIOGRÁFICAS	70
APÉNDICES	78
A. Programación de número de nodos y sensores en matlab	78
B. Programación del nodo transmisor en Arduino IDE 2.2.1	79
C. Programación del nodo receptor en Arduino IDE 2.2.1	88
D. Programación de la aplicación web en Delphi	96
Programación de Unit Da_base servidor tcp.....	96
Programación de interfaz de usuario de la aplicación web	107
Programación del consumo de datos de la interfaz de la aplicación web	114

ÍNDICE DE FIGURAS

<i>FIGURA 1 PARTES DEL MAÍZ</i>	4
<i>FIGURA 2 TIPOS DE SENSORES</i>	11
<i>FIGURA 3 INTERNET DE LAS COSAS.....</i>	17
<i>FIGURA 4 MAPAS SIG EN 2D Y 3D UTILIZADOS PARA DESARROLLAR Y PRESENTAR ALTERNATIVAS DE DISEÑO Y ALGUNOS DE LOS ANÁLISIS EMPLEADOS COMO ENTRADAS EN LAS DECISIONES. ..</i>	18
<i>FIGURA 5 REPRESENTACIÓN VISUAL DE UNA RED</i>	19
<i>FIGURA 6 AGRICULTORES ROBÓTICOS</i>	19

<i>FIGURA 7 ESTRUCTURA BLOCKCHAIN EN LA INDUSTRIA AGRÍCOLA ALIMENTARIA</i>	<i>20</i>
<i>FIGURA 8 SENSOR DE TEMPERATURA "GIKFUN DS18B20"</i>	<i>21</i>
<i>FIGURA 9 SENSOR DE HUMEDAD "HD-38 HIGRÓMETRO"</i>	<i>22</i>
<i>FIGURA 10 ESQUEMA DE PINES DE LA PLACA WIFI LORA V3</i>	<i>24</i>
<i>FIGURA 11 WIFI LORA V3</i>	<i>25</i>
<i>FIGURA 12 MÓDULO RELÉ</i>	<i>25</i>
<i>FIGURA 13 VÁLVULA ELECTROMAGNÉTICA BEDUAN 12V 1/2.....</i>	<i>27</i>
<i>FIGURA 14 CINTA AISLANTE TRUPER EXPERT.....</i>	<i>27</i>
<i>FIGURA 15 FUENTE CONMUTADA 12V/10A</i>	<i>28</i>
<i>FIGURA 16 MANGUERA DE RIEGO 1/2" 13MM.....</i>	<i>29</i>
<i>FIGURA 17 ASPERSOR EXCEL WOBBLER 1/2</i>	<i>29</i>
<i>FIGURA 18 CAJA DE DERIVACIÓN DE 105x1050x55MM</i>	<i>30</i>
<i>FIGURA 19 CAJA DE DERIVACIÓN DE 80x80x45MM</i>	<i>31</i>
<i>FIGURA 20 CABLE AISLADO PARALELO TIPO SPT.....</i>	<i>32</i>
<i>FIGURA 21 UBICACIÓN DEL TERRENO</i>	<i>34</i>
<i>FIGURA 22 LÍNEA DE VISTA DE ÁREA SEMBRADA.....</i>	<i>34</i>
<i>FIGURA 23 LÍNEA DE VISTA DE LA CASA DEL USUARIO Y EL TERRENO</i>	<i>35</i>
<i>FIGURA 24 ESQUEMA DEL SISTEMA DE RIEGO POR ASPERSIÓN</i>	<i>35</i>
<i>FIGURA 25 DIAGRAMA DE BLOQUES PARA EL CÁLCULO DE NODOS Y SENSORES</i>	<i>36</i>
<i>FIGURA 26 TOTAL DE NODOS Y SENSORES DE HUMEDAD-TEMPERATURA.....</i>	<i>37</i>
<i>FIGURA 27 DELPHI.....</i>	<i>37</i>
<i>FIGURA 28 ARDUINO</i>	<i>38</i>
<i>FIGURA 29 DIAGRAMA DE BLOQUES TRANSMISOR</i>	<i>40</i>
<i>FIGURA 30 CAPA INTERIOR DEL PCB</i>	<i>41</i>
<i>FIGURA 31 DISEÑO DE PLACA PCB EN 3D</i>	<i>42</i>
<i>FIGURA 32 ESQUEMA ELECTRÓNICO</i>	<i>42</i>
<i>FIGURA 33 DIAGRAMA DE BLOQUES RECEPTOR</i>	<i>44</i>
<i>FIGURA 34 DIAGRAMA DE BLOQUES DE LA APLICACIÓN WEB</i>	<i>45</i>
<i>FIGURA 35 ENVIÓ DE DATOS DE LAS NODOS TRANSMISOR Y RECEPTOR.....</i>	<i>46</i>
<i>FIGURA 36 OBTENCIÓN DE DATOS DE LOS SENSORES</i>	<i>47</i>
<i>FIGURA 37 ENVIÓ DE DATOS DESDE EL SERVIDOR A LA APLICACIÓN WEB</i>	<i>47</i>
<i>FIGURA 38 DATOS DE LOS SENSORES Y ESTADO EN</i>	<i>48</i>

<i>FIGURA 39 CONDICIONES DE LOS SENSORES</i>	49
<i>FIGURA 40 CONEXIÓN AL RECEPTOR</i>	50
<i>FIGURA 41 DISEÑO DE LA INTERFAZ GRÁFICA DE LA APLICACIÓN WEB.....</i>	51
<i>FIGURA 42 PROGRAMACIÓN PARA CONTROLAR LOS DATOS DE LA INTERFAZ GRAFICA</i>	52
<i>FIGURA 43 APLICACIÓN WEB</i>	53
<i>FIGURA 44 MONTAJE DE NODO TRANSMISOR</i>	54
<i>FIGURA 45 PROTECCIÓN DE CABLES DEL NODO TRANSMISOR</i>	55
<i>FIGURA 46 CASETA DEL NODO TRANSMISOR.....</i>	55
<i>FIGURA 47 VISTA INTERNA DESDE LA CASETA DEL NODO TRANSMISOR.....</i>	56
<i>FIGURA 48 VÁLVULA ELECTROMAGNÉTICA Y FUNCIONAMIENTO CON EL RELÉ</i>	57
<i>FIGURA 49 CONEXIÓN DE LA ELECTROVÁLVULA A LA FUENTE Y AL RELÉ [.....</i>	57
<i>FIGURA 50 MONTAJE DE NODO RECEPTOR</i>	58
<i>FIGURA 51 VISTA EXTERNA DEL NODO RECEPTOR.....</i>	59
<i>FIGURA 52 DISTRIBUCIÓN DE SENSORES EN EL TERRENO</i>	59
<i>FIGURA 53 APLICACIÓN WEB</i>	61
<i>FIGURA 54 FUNCIONAMIENTO EN LA SEMANA 09/12/2023 AL 13/12/2023</i>	64
<i>FIGURA 55 FUNCIONAMIENTO EN LA SEMANA 13/12/2023 AL 17/12/2023.....</i>	64
<i>FIGURA 56 FUNCIONAMIENTO EN LA SEMANA 17/12/2023 AL 21/12/2023</i>	65
<i>FIGURA 57 FUNCIONAMIENTO EN LA SEMANA 21/12/2023 AL 25/12/2023.....</i>	65
<i>FIGURA 58 FUNCIONAMIENTO EN LA SEMANA 25/12/2023 AL 29/12/2023.....</i>	66
<i>FIGURA 59 FUNCIONAMIENTO EN LA SEMANA 29/12/2023 AL 02/01/2024.....</i>	67
<i>FIGURA 60 ACTIVACIÓN Y DESACTIVACIÓN DE RIEGO MANUAL DESDE LA APLICACIÓN WEB ...</i>	68

ÍNDICE DE TABLAS

TABLA 1 MICRONUTRIENTES PRESENTES EN EL SUEÑO	6
TABLA 2 TIPOS DE SENSORES DE TEMPERATURA PARA SUELO	11
TABLA 3 TIPOS DE SENSORES DE HUMEDAD PARA SUELO	13
TABLA 4 CARACTERÍSTICAS DE LAS DIFERENTES PLACAS DE HELTEC WIFI LORA	22
TABLA 5 CARACTERÍSTICAS DEL MÓDULO RELÉ DE 1 CANAL DE 5VDC CON OPTOACOPLADOR	26
TABLA 6 TABLA ESPECIFICACIONES DE LA FUENTE 12V-10A.....	28

TABLA 7 TABLA CARACTERÍSTICAS DEL ASPERSOR EXCEL WOBBLER 1/2	29
TABLA 8 TABLA CONDICIONES DE TEMPERATURA Y HUMEDAD	39
TABLA 9 TABLA PRESUPUESTO	60
TABLA 10 TABLA PORCENTAJES DE CADA ESTADO DE HUMEDAD	62

RESUMEN

Con el propósito de atender las necesidades de los agricultores dedicados al cultivo de maíz, se ha concebido y materializado el “Diseño e implementación de un sistema de riego IoT para sembríos de maíz”. La finalidad principal es establecer un mecanismo que permita el monitoreo y controlar manualmente el riego por aspersión mediante una aplicación web. La consideración central en este proyecto fue desarrollar un modelo más económico, ya que, en la actualidad, muchos sistemas de riego con tecnología IoT disponibles en el mercado poseen costos significativos.

La concepción de este sistema abarcó tanto el aspecto hardware como el software. En la implementación de hardware, se utilizaron sensores de humedad y temperatura, relés, placas WiFi Lora, fuentes de alimentación y electroválvula. Para el software, se emplearon herramientas como Delphi, MariaDB y Arduino. La disposición estratégica de estos componentes se planificó meticulosamente para garantizar su funcionamiento eficiente y durabilidad.

Antes de iniciar con la programación y ensamblaje de los nodos transmisor y receptor, se llevaron a cabo pruebas de funcionamiento con los sensores en el área designada para obtener datos reales, esenciales para el funcionamiento del sistema. Se realizaron escalas de valores tanto para los sensores de humedad como para los sensores de temperatura, proporcionando una visión más precisa de las características del terreno destinado al cultivo de maíz.

La efectividad del sistema de riego por aspersión se determinó considerando una temperatura mínima de 20°C y una máxima de 30°C. Asimismo, el rango de humedad se definió en función de las categorías de riego necesario, suelo seco, exceso de humedad y suelo húmedo. En cuanto a la programación, se optó por dos placas WiFi Lora V3 debido a su tecnología, acceso WiFi y versatilidad en la programación, permitiendo que el equipo funcione tanto como nodo transmisor o receptor, facilitando así la comunicación a larga distancia.

La fase de implementación incluyó la instalación de una caseta en la que se encuentra el nodo transmisor. Este nodo recibe señales de los sensores de temperatura y humedad, controla el relé que activa y desactiva la electroválvula. Todas estas señales obtenidas se transmiten al

nodo receptor, el cual envía la información cada tres minutos a un servidor TCP para su posterior almacenamiento en una base de datos y visualización a través de una aplicación web.

ABSTRACT

In order to cover the needs of maize farmers, the "Design and Implementation of an IoT Irrigation System for Corn Crops" has been conceived and realized. The main purpose is to establish a mechanism that allows the monitoring and manual control of sprinkler irrigation through a web application. The central consideration in this project was to develop a more affordable model, as many currently available IoT irrigation systems in the market have significant costs.

The conception of this system covered both hardware and software aspects. In the hardware implementation, moisture and temperature sensors, relays, LoRa WiFi boards, power sources, and solenoid valves were used. For the software, tools such as Delphi, MariaDB, and Arduino were employed. The strategic arrangement of these components was meticulously planned to ensure their efficient operation and durability.

Before proceeding with the programming and assembly of the transmitter and receiver nodes, functionality tests were conducted with the sensors in the designated area to obtain real data essential for the system's operation. Value scales were established for both moisture and temperature sensors, providing a more accurate insight into the characteristics of the maize cultivation area.

The effectiveness of the sprinkler irrigation system was determined considering a minimum temperature of 20°C and a maximum of 30°C. Likewise, the humidity range was defined based on categories of required irrigation: dry soil, excess moisture, and moist soil. Regarding programming, two LoRa WiFi V3 boards were chosen due to their technology, WiFi access, and versatility in programming, allowing the equipment to function as both a transmitter and receiver node, facilitating long-distance communication.

The implementation phase included the installation of a cabin where the transmitter node is located. This node receives signals from temperature and humidity sensors, controls the relay that activates and deactivates the solenoid valve. All obtained signals are transmitted to

the receiver node, which sends the information every three minutes to a TCP server for subsequent storage in a database and visualization through a web application.

INTRODUCCIÓN

La agricultura es una piedra angular de la subsistencia humana y al mismo tiempo una de las actividades más susceptibles a las variaciones climáticas y ambientales. En este contexto, el cultivo de maíz, un alimento fundamental en la dieta global, se ha convertido en un pilar económico y alimentario en la parroquia de Vilcabamba, ubicada en la provincia de Loja. Sin embargo, en los últimos años, los efectos del cambio climático han provocado importantes desafíos para los agricultores de esta región, afectando tanto la cantidad como la calidad de sus cultivos debido a las variaciones de temperatura y humedad [1].

En esta encrucijada, donde la necesidad de producir más con menos recursos es apremiante, emerge la tecnología como un aliado inesperado en el campo agrícola. La agricultura de precisión, impulsada por la Internet de las Cosas (IoT), ofrece soluciones innovadoras para abordar la problemática de la escasez de agua y la optimización del riego en cultivos cruciales como el maíz. El presente proyecto surge con el propósito de diseñar e implementar un sistema de riego IoT en la finca del señor Juan Castillo, propietario de un terreno dedicado al cultivo de maíz híbrido.

En medio de este enigma, donde la necesidad de producir más con menos recursos es primordial, la tecnología emerge como un aliado inesperado en el campo agrícola. Motivada por el Internet de las Cosas (IoT), la agricultura de precisión proporciona soluciones innovadoras para lidiar con la problemática de la escasez de agua y optimizar el riego en cultivos cruciales como el maíz. El presente proyecto tiene como objetivo diseñar e implementar un sistema de riego IoT en la propiedad del Señor Juan Castillo, quien posee una parcela de maíz dedicado al cultivo del mismo

Este sistema de riego basado en IoT no solo busca mejorar la gestión de los recursos hídricos en la agricultura, sino también contribuir al bienestar económico de los agricultores locales y por extensión, a la seguridad alimentaria de la comunidad. La implementación de sensores estratégicamente ubicados en el terreno, junto con el uso de tecnología LoRa para el monitoreo y control de las variables climáticas, promete revolucionar la forma en que se cultiva

el maíz en esta región. Con una menor intervención humana, mayor precisión en el riego y la posibilidad de tomar decisiones informadas ante cambios climáticos, se espera obtener un aumento significativo en la producción y calidad del maíz.

Este proyecto de investigación se propone, en última instancia, impulsar la adopción de tecnologías IoT en la agricultura, proporcionando a los trabajadores del señor Juan Castillo una herramienta valiosa para enfrentar los desafíos climáticos y mejorar la eficiencia de sus operaciones. Además, se espera que esta tesis contribuya conocimientos en el área académica al demostrar cómo la tecnología puede transformar y modernizar la agricultura, un sector esencial para la subsistencia de las comunidades rurales y la seguridad alimentaria global.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

Existen diversas actividades que contribuyen al desarrollo económico de las poblaciones en todo el mundo, y una de las más significativas es la agricultura. Esta actividad es fundamental, ya que se relaciona con la alimentación de millones de personas. Entre los productos agrícolas más importantes destaca el maíz un grano nutritivo y antiguo que ofrece una gran versatilidad tanto para seres humanos como para animales y que se produce de una manera potencial en países como Estados Unidos, China, Brasil, Argentina, México, Canadá, India y Indonesia los cuales cubren más de dos tercios de la producción mundial de maíz [1], [2].

El maíz juega un papel crucial en Ecuador, este cereal es cultivado en varias provincias del país, incluyendo Guayas, que contribuye con el 21% de la producción, Los Ríos con un 33%, Manabí con un 22%, Loja con un 8%, Pichincha con un 1%, Imbabura con un 1%, mientras que el resto de las provincias contribuyen con el 14% restante [3].

En la actualidad, se cultiva en diversos ambientes, que van desde terrenos tropicales hasta zonas montañosas e incluso zonas desérticas. La producción de maíz incorpora muy bien el conocimiento sobre las condiciones naturales de los suelos y de los fenómenos meteorológicos de sus propiedades alimenticias y algunos casos medicinales [4],[5].

Para que el cultivo de maíz alcance su máximo potencial de producción, es esencial que reciba una cantidad adecuada de agua distribuida uniformemente durante su ciclo de crecimiento, que oscila entre 500 y 800 mm [6].

Un sistema de monitoreo y riego inteligente con instrumentación IoT para la agricultura pudo reducir el consumo de recursos como el agua en el riego de cultivos, la cual se vio limitada por el uso de Wifi, pero se tuvo resultados muy favorables en el sistema el cual se llevó a cabo en el año 2020 por el departamento de Electrónica de la universidad de Vijan [7].

En el ámbito de las granjas inteligentes, según el análisis de Hyuk Lee en 2020 destaca la preferencia por la tecnología LoRa sobre APRS (Sistema de Reporte Automático de Posición). Es esencial señalar que APRS, siendo una solución viable, presenta limitaciones notables en alcance y eficiencia energética, particularmente en comparación con LoRa, son evidentes ofrece una mayor flexibilidad en la gestión de la red, permitiendo ajustes dinámicos en la potencia de transmisión y el ancho de banda, lo cual puede ser esencial para optimizar la comunicación en tiempo real en amplias áreas agrícolas y a largo alcance [8].

En la parroquia de Vilcabamba ubicada a 45 km de la ciudad de Loja los productos que se cultivan son la caña de azúcar, café, frejol, zarandaja , yuca, tomate de riñón , hortalizas y el maíz siendo una parroquia muy rica en diversidad de suelo y con un clima tropical acogedor donde sus aguas son denominadas de eterna juventud sin embargo la producción de dichos productos en los últimos años han disminuido por los diversos cambios climáticos y porque muchos de los productores que se dedican a la agricultura tienen una edad avanzada y por lo tanto el nivel de producción ya no es el mismo ya que, en ciertos lugares, los terrenos para el cultivo se encuentran a largas distancias por lo tanto el riego de los sembríos es inconstante y a su vez la producción disminuye ya que uno de los recursos esenciales para la producción de maíz es el agua que este en una etapa crucial ayudara al desarrollo de la planta y de su fruto o mazorca [9], [10].

JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

En vista de los antecedentes presentados se realizó una investigación que parte desde la importancia del maíz y de la incorporación de la tecnología en la agricultura con el objetivo de mejorar la utilización de los recursos naturales necesarios para el cultivo. En la parroquia de Vilcabamba ubicada en la provincia de Loja, la agricultura es la principal actividad comercial y el cultivo de maíz es uno de los más destacados. A pesar de esto, los cambios en la temperatura y la humedad que se han presentado como resultado de las variaciones climáticas han tenido un

impacto negativo en el desarrollo de las plantas, disminuyendo tanto su producción como su calidad [10].

Como resultado, los agricultores de la zona han sufrido pérdidas económicas importantes. El maíz es un cultivo que se ve afectado negativamente por la escasez de agua, especialmente durante la fase de floración, lo que reduce el rendimiento de los granos. Para maximizar la producción y minimizar los efectos del estrés hídrico, se deben buscar métodos más eficientes de manejo del agua, como el sistema de riego basado en IoT [10].

Por lo tanto, para el desarrollo de este proyecto y el recurso a utilizarse (Agua), se dará en la finca del señor Juan Castillo propietario ; por ende, se implementara un sistemas de riego por aspersión que permita monitorear y controlar el uso de este recurso en el cultivo, se tomara en cuenta dos variables climáticas como lo es la temperatura y la humedad utilizando la tecnología Lora que permite un control preciso de los parámetros del cultivo a largas distancias y posibilita que los agricultores puedan tomar decisiones bien informadas ante cualquier variación.

La introducción de sensores en lugares estratégicos del terreno ayudara a mejorar la producción y de esta manera a obtener mayores beneficios. Como bien se sabe el rendimiento de los cultivos de maíz dependen mucho del cuidado y riego donde la mitad del agua que se pierde en evaporación y afecta al desarrollo de este [10].

El sistema requiere menos intervención humana, lo que aumenta su seguridad y confiabilidad. Además, el sistema de telecontrol del cultivo recolecta datos mediante la intervención manual, mejorando el desarrollo y el rendimiento del cultivo de maíz al proporcionar la cantidad necesaria de agua en el momento adecuado.

OBJETIVOS

OBJETIVO GENERAL

- Desarrollar un sistema de riego basado en IOT en una parcela destinada al cultivo de maíz híbrido.

OBJETIVOS ESPECÍFICO

- Investigar las tecnologías basadas en el sector agrícola, aplicaciones y funcionamiento de sensores para el monitoreo y riego del suelo.
- Controlar el estado de riego por aspersión mediante el desarrollo de un algoritmo que permita la activación del sistema para el cultivo de maíz por medio de una aplicación web.
- Evaluar la efectividad y eficiencia del sistema de riego mediante pruebas de campo para el sembrío de maíz.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

FUNDAMENTACIÓN TEÓRICA

1.1 EL MAÍZ Y SU ORIGEN

El origen del maíz se encuentra en América, y esto ha sido confirmado por medio de la presencia de los fósiles de polen más antiguos encontrados en la ciudad de México. Después de la llegada de Colón, el cultivo del maíz se extendió a América del Norte y al resto del mundo. Aunque las evidencias arqueológicas de la evolución del maíz son más limitadas y recientes en Sudamérica, especialmente en las regiones costeras de Perú [11].

1.1.1 IMPORTANCIA DEL MAÍZ EN EL MUNDO Y EN EL ECUADOR

En Ecuador, el maíz es considerado como un elemento vital para los pueblos indígenas, ya que forma parte de sus creencias religiosas, celebraciones y, sobre todo, de su alimentación. A lo largo de los años, cada generación ha transmitido la sabiduría tradicional para el cultivo de este grano, lo que ha permitido mantener viva la práctica ancestral en el país [11].

El trigo, el arroz y el maíz son las especies más sembradas a nivel mundial, siendo el maíz el que ocupa la posición de liderazgo en términos de área de cultivo y producción de cosecha. Durante los últimos 50 años, su cultivo ha experimentado un notable crecimiento en línea con el aumento de la población y la demanda de alimentos, con una producción de más de 600 millones de toneladas métricas cultivadas en 150 millones de hectáreas, y siendo nuestro continente el de mayor producción [11].

1.1.2 CLASIFICACIÓN BOTÁNICA DEL MAÍZ

Según Garcés N (1987), a clasificación botánica es la siguiente [11]:

- Reino: Plantae (Plantas)
- División: Magnoliophyta (Plantas con flor)
- Clase: Liliopsida (Monocotiledóneas)
- Orden: Poales (Poales)

- Familia: Poaceae (Gramíneas)
- Género: Zea
- Especie: Zea mays

1.1.3 IMPORTANCIA DEL MAÍZ EN LA PARROQUIA DE VILCABAMBA

El maíz es un cultivo de gran relevancia para Vilcabamba, ya sea desde el punto de vista alimentario, cultural o económico. En primer lugar, el maíz es una parte importante de la dieta de los residentes locales. Se utiliza en platos tradicionales en muchas cocinas locales. Las tortillas de maíz, las humitas, los tamales y otros alimentos se elaboran con el maíz como ingrediente principal. El maíz aporta los hidratos de carbono, fibra, vitaminas y minerales necesarios en una dieta equilibrada.

Desde el punto de vista cultural el maíz tiene un gran valor en Vilcabamba. La planta está estrechamente asociada con las fiestas y celebraciones tradicionales de la comunidad. Por ejemplo, Fiesta del Maíz es un festival que conmemora y celebra la importancia del maíz en la cultura local. Durante esta festividad se realizan actividades relacionadas con la cosecha y preparación del maíz y se destaca la importancia histórica y cultural de este cultivo ancestral.

1.1.4 CARACTERÍSTICAS BOTANICAS DEL MAÍZ

El maíz es una planta de crecimiento sencillo y resistente, que puede alcanzar alturas de entre 2.70 y 3 metros y se compone de raíces, tallos, hojas, flores y granos [11].

Raíz

Las raíces del maíz crecen en racimo y su función principal es proporcionar una base sólida a la planta. En algunas situaciones, las raíces secundarias o adventicias pueden sobresalir de los nudos del tallo al nivel del suelo [12].

En las primeras etapas de desarrollo del maíz, las raíces primarias emergen de la semilla y tienen una función limitada. Sin embargo, se degeneran rápidamente y son reemplazadas por raíces secundarias o adventicias que surgen desde el octavo o décimo nudo en la base del tallo, por debajo de la superficie del suelo. Estas raíces forman un sistema de cabellera que se extiende a profundidades diversas [12].

Tallo

La planta de maíz tiene un tallo compuesto por nudos y entrenudos donde se desarrollan las hojas y se produce la elongación. Cada nudo es el punto de intersección de una hoja, y el tallo principal emerge del suelo cuando la planta tiene cinco hojas, a partir de la sexta hoja, el tallo comienza a crecer rápidamente en altura y puede llegar a medir más de 4 metros, con entre 12 y 24 nudos aéreos dependiendo de la variedad de maíz cultivada [12].

Hojas

Las hojas del maíz son largas y de gran tamaño, con forma de lanza y se unen al tallo de la planta. En la superficie superior de las hojas se pueden observar pelos, mientras que la vaina de la hoja forma un cilindro alrededor del entrenudo. Por lo general, las hojas son verdes, aunque en algunos casos pueden presentarse con rayas de color blanco o púrpura debido a deficiencias nutricionales. El número de hojas por planta puede oscilar entre 8 y 25 [12].

Flores

La fase de floración masculina en el maíz está compuesta por una estructura amarilla denominada panícula, la cual es una fuente abundante de polen, conteniendo de 20 a 25 millones de granos de polen. En cada flor de la panícula, se encuentran tres estambres que producen el polen. Por otro lado, las flores femeninas aparecen en grupos dentro de una mazorca rodeada por brácteas largas. En la mazorca, se encuentra una parte central denominada zuro o "corazón" [12].

Fruto

El fruto del maíz es la mazorca, que se compone de una sección central conocida como zuro o corazón, en la que se adhieren cientos de granos de maíz. Esta parte representa entre el 15% y el 30% del peso total de la espiga [12].

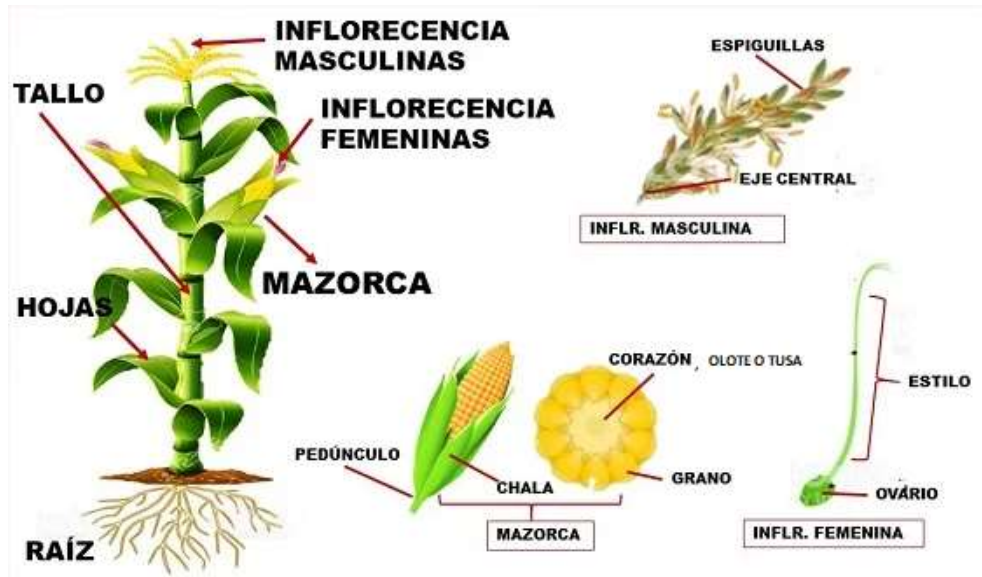


Figura 1 Partes del maíz. [12]

1.1.5 VARIABLES CLIMATICAS QUE AFECTAN LA PRODUCCION DEL MAÍZ

El cambio climático el mundo está afectando ampliamente al campo de la agricultura, ya que las variables climáticas que influyen en el desarrollo del crecimiento del cultivo de maíz son las lluvias, temperatura y la radiación solar los principales factores del desarrollo de las plantaciones de maíz. La mazorca y su contenido de granos conforman una sola entidad, que se asemeja a una cariósida brillante que puede variar en color entre el amarillo, rojo, morado o blanco. A menudo se les llama "granos dentro del fruto". El ovario maduro es el que contiene las semillas, que son óvulos fecundados y maduros [13].

Temperatura: El maíz necesita mantenerse en el rango de 25 a 30°C para crecer adecuadamente, por lo que la temperatura es el factor determinante en su desarrollo. Aunque puede soportar temperaturas tan bajas, pero a temperaturas altas, puede tener dificultades para absorber agua y nutrientes, afectando negativamente su crecimiento [14].

Radiación Solar: La incidencia de la luz solar es crucial para el rápido y eficiente crecimiento del maíz y la planta también requiere grandes cantidades de agua, ya que la fotosíntesis depende de ella. Además, se necesita una cantidad significativa de agua para producir 1 kilogramo de granos de maíz, específicamente 1000 litros [14].

Humedad: La reducción de la absorción de agua afecta directamente al crecimiento de las plantas, debido a que también se reducirá la capacidad de absorción de nutrientes, lo que se traducirá en una reducción en la tasa de crecimiento de los cultivos [14].

1.1.6 TIPOS SUELO Y SUS NUTRIENTES

Es el medio en el que crecen las plantas y es el resultado de cambios naturales y de la intervención biológica y de la piedra. Los diferentes tipos de suelo tienen diferentes propiedades.

Suelo Arenoso: Los suelos arenosos tienen una textura gruesa porque se componen principalmente de partículas de arena. Aunque son sueltos y facilitan el drenaje, carecen de capacidad para retener nutrientes y agua [15].

Suelo Arcilloso: La alta proporción de partículas de arcilla en los suelos arcillosos les da una textura fina y sedosa. Cuando los suelos se secan, pueden volverse compactos y difíciles de trabajar, pero mantienen agua y nutrientes de manera efectiva [15].

Suelo Hemisférico: El término "suelo hemisférico" no se refiere a un tipo de suelo ampliamente aceptado. Puede ser un término local o especializado que se usa en un contexto particular sin una definición estándar [15].

Suelo Calcáreo: El carbonato de calcio (cal) está presente en niveles elevados en los suelos calcáreos. Suelen tener un pH alcalino o básico y, debido al calcio, pueden ser blanquecinas. Con frecuencia, estos suelos tienen una estructura sólida y una alta capacidad para retener nutrientes [15].

Suelo Árido: Los suelos áridos se encuentran en áreas secas o desérticas donde no hay agua disponible. La falta de materia orgánica y nutrientes en ellos los hace inadecuados para la agricultura sin riego adecuado. La evaporación del agua puede causar altas concentraciones de sales en los suelos áridos [15].

Los elementos esenciales necesarios en grandes cantidades para el crecimiento, desarrollo y supervivencia de las plantas se conocen como macronutrientes. Los macronutrientes se pueden clasificar en dos grupos principales, como se muestra en la Tabla 1 [16].

Tabla 1 Micronutrientes presentes en el suelo [3]

Primarios	Secundarios
Nitrógeno [N]	Calcio [Ca]
Fosforo [P]	Magnesio [Mg]
Potasio[K]	Azufre [S]

1.1.7 CONSIDERACIONES DE HUMEDAD Y TEMPERATURA DE LA PLANTA

La cantidad de agua que una planta necesita para crecer adecuadamente incluye la cantidad de agua que pierde por evaporación y transpiración, así como la cantidad de agua que recibe de las precipitaciones [17].

Dependiendo del clima, las temperaturas entre 20 y 30°C grados centígrados son ideales para el crecimiento del maíz; sin embargo, después de los 30°C grados centígrados, no se recomienda regar porque la planta puede sufrir daños para absorber nutrientes. En términos de humedad, la planta necesita 6 mm de agua al día. Hay que tener en cuenta que, aunque la planta necesita más agua durante la etapa de floración para producir frutos adecuados, necesita más recursos hídricos y humedad en su suelo [17].

1.1.8 EVAPORACIONES Y TRANSPIRACIÓN

El proceso natural en el que el agua cambia de estado de líquido a gas o vapor se conoce como evaporación. La humedad del aire, la velocidad del viento y la temperatura ambiente son algunos de los factores que influyen en la evaporación. La transpiración también libera vapor de agua de los tejidos internos de las hojas a la atmósfera. La evapotranspiración es la mezcla de evaporación y transpiración [18].

El riego es importante para proporcionar agua al suelo y permitir el crecimiento de plantas cultivadas. A lo largo de la historia, se han utilizado diferentes técnicas de riego, muchas de las cuales son artesanales [18].

1.2 RIEGO

El riego se le atribuye la capacidad de dotar agua al sustrato (suelo) para que así las plantas derivadas de la agricultura o floricultura tengan la capacidad de desarrollarse y crecer.

1.2.1 SISTEMA DE RIEGO

Dado que tener áreas verdes ya no es un lujo sino una necesidad para conservar la vida en el planeta, existen varios sistemas de riego que se adaptan al cultivo de cada región. Dichos sistemas mejoran el uso del agua permitiendo tener un control y uso eficiente del agua disponible.

1.2.2 TIPOS DE RIEGO

Riego por aspersión: Se trata de un método de riego en el que el agua se dispersa en pequeñas gotas sobre las plantas a través de dispositivos como aspersores o boquillas. Este sistema imita la lluvia y garantiza que el agua se distribuya uniformemente sobre el suelo y las plantas [19].

Riego por goteo: Este método de riego proporciona agua precisa a las raíces de las plantas a través de tuberías que tienen pequeñas aberturas o goteros. Al darle solo a las plantas lo que necesitan, reduce el desperdicio de agua [19].

Riego por microaspersión: Este método, que es similar al riego por aspersión, utiliza aspersores de menor tamaño para pulverizar el agua sobre las plantas en forma de gotas finas. Esto permite un mejor control del consumo de agua y es adecuado para cultivos más delicados [19].

Cultivo hidropónico: El cultivo hidropónico crece las plantas en soluciones nutritivas en lugar del suelo. El agua enriquecida con nutrientes se administra directamente a las raíces, lo que permite un riego y una regulación precisas de los nutrientes [19].

Riego por nebulización: Este sistema usa alta presión para pulverizar agua sobre las plantas en forma de niebla o gotas finas. Su utilidad radica en el aumento de la humedad en invernaderos o en la refrigeración del ambiente en climas extremadamente cálidos [19].

Riego por gravedad: Utiliza la inclinación natural del terreno para llevar el agua desde fuentes más altas a los campos o cultivos. Este método es simple y utiliza la fuerza gravitatoria para distribuir el agua [19].

Riego por inundación: Consiste en inundar controladamente el área de cultivo con agua para que el suelo se llene de agua y las plantas se alimenten. Es particularmente frecuente en los campos de arroz y el cultivo de plantas acuáticas [19].

1.3 IOT (INTERNET DE LAS COSAS)

Es un sistema que enlaza dispositivos a la red, facilita la interconexión entre el ámbito digital y el físico, además de posibilitar el tratamiento de datos y la supervisión integral de los sistemas. Tras el análisis, los datos recolectados se guardan en la nube y se emplean para la toma de decisiones. [20].

1.3.1 TECNOLOGÍAS LPWAN

Se refiere a las redes de baja potencia y amplias áreas que se utilizan en Internet de las cosas, que pueden comunicarse inalámbricamente con un gran alcance, bajo consumo de energía y baja tasa de transmisión de datos. A pesar de estas características, esta tecnología no es buena para transmitir video y audio. En este sentido, la modulación de banda angosta o de espectro ensanchado en la capa física es esencial para esta tecnología. La modulación de espectro ensanchado permite la integración de múltiples dispositivos y ofrece una transmisión más confiable y resistente a interferencias o ataques [20].

1.3.2 LORA

La tecnología LoRa, que envía mensajes mediante modulación de radiofrecuencia. La empresa SEMTECH posee la propiedad intelectual de esta tecnología, que es parte de las redes LPWAN. La capacidad de enviar datos a largas distancias en áreas urbanas y rurales con menos interferencias y su bajo consumo de energía son algunas de las muchas ventajas significativas de LoRa para su uso en proyectos de Internet de las cosas [21].

1.3.4 APLICACIONES DE LA TECNOLOGIA LORA

Las aplicaciones que puede ejecutar la tecnología Lora tienen la capacidad de extraer y enviar información, lo que permite el control remoto de varios dispositivos conectados a la red Lora. A continuación, se enumeran las aplicaciones de esta tecnología [21]:

- Ciudades Inteligentes
- Redes de sensores
- Automatización Industrial
- Medidores Inteligentes
- Agricultura Inteligente
- Seguimiento de flotas
- Domótica

1.3.5 MODULACIÓN RADIO LORA

Debido a su resistencia a las interferencias y su capacidad para transmitir datos a largas distancias, la técnica de modulación Chirp Spread Spectrum (Espectro Extendido por Chirrido) es ampliamente utilizada por los militares. Para codificar la información, esta técnica utiliza tonos de frecuencia con una variación lineal a lo largo del tiempo. Esto hace que el decodificador sea más fácil de eliminar las compensaciones de frecuencia y tiempo. Además, el método es inmune al efecto Doppler, lo que permite un desplazamiento de frecuencia de hasta el 20% del ancho de banda sin afectar el rendimiento de la decodificación. Al no requerir una precisión extrema en la fabricación de los cristales utilizados en los transmisores, estas características hacen que los transmisores LoRa sean menos costosos [22].

1.3.6 AGRICULTURA INTELIGENTE

Con una población creciente año tras año y los avances en el ámbito tecnológico en campos de la digitalización, aportan al agro y a la ganadería para solventar la demanda de alimentos, a esta revolución de la tecnología se lo denomina agricultura 4.0 o agricultura inteligente que integra diferentes técnicas de distintas disciplinas para transformar de forma

significativa la producción de la agricultura para en un futuro poseer la capacidad de producción necesaria para la población. La eficiente utilización de los recursos naturales indispensables para el cultivo se ha tomado en cuenta para introducir la tecnología en las labores del día a día del agricultor. Entonces la monitorización y registro de datos permiten tener un control para analizarlos y tomar decisiones frente alguna variación de los parámetros. Además de que por medio de la introducción de sensores en puntos estratégicos del terreno permite al agricultor mejorar la producción obteniendo mejores beneficios [23].

1.4 REDES DE SENSORES APLICADAS EN LA AGRICULTURA

Los nodos son conjuntos de dispositivos que están adquiriendo cada vez más importancia en diversos campos como la industria, la domótica y la agricultura. Estos nodos se pueden utilizar en agricultura para monitorear diversas condiciones, como variables ambientales que afectan el crecimiento de los cultivos. [23].

Las redes de sensores están compuestas por nodos router que se encargan de recolectar información y luego envían esos datos al nodo coordinador, que es responsable de recibir y almacenar la información [23].

1.4.1 SENSORES

Los sensores son herramientas electrónicas o mecánicas diseñadas para detectar y medir magnitudes físicas o químicas como la temperatura, la presión, la humedad, la luz, el sonido y el movimiento, entre otras. Estos dispositivos convierten las señales detectadas en señales eléctricas o digitales, que pueden procesarse para el control de sistemas o la recolección de datos [24].



Figura 2 Tipos de sensores [25]

1.4.2 SENSOR DE TEMPERATURA

Un sensor de temperatura es un dispositivo utilizado para medir la temperatura, como un termómetro. Surgió en 1874 con el avance de la tecnología de onda corta. Con el tiempo, se han desarrollado diversos tipos de sensores de temperatura para diferentes aplicaciones en el Siglo XX y XXI en este caso los que son utilizados para medir la temperatura del suelo tenemos [26]:

Tabla 2 Tipos de sensores de temperatura para suelo [27]

Tipo de Sensor de Temperatura del Suelo	Ventajas	Desventajas	Aplicaciones
Sensores de Resistencia (RTDs)	<ul style="list-style-type: none"> - Alta precisión. - Estabilidad a largo plazo. - Linealidad. 	<ul style="list-style-type: none"> - Costo moderado. - Rango de temperatura limitado. 	<ul style="list-style-type: none"> - Agricultura de precisión. - Investigación agrícola. - Monitoreo ambiental.
Termopares	<ul style="list-style-type: none"> - Amplio rango de temperatura. - Buena respuesta dinámica. 	<ul style="list-style-type: none"> - Precisión moderada. - Requieren amplificación. 	<ul style="list-style-type: none"> - Monitoreo de temperatura del suelo en tiempo real. - Control de temperatura en

			invernaderos. -Investigación en suelos y geología.
Fibra Óptica	- Sin contacto físico. - Pueden medir a diferentes profundidades. - Inmunidad al ruido eléctrico.	- Costo más alto. -Requieren conocimiento técnico para instalación y uso.	- Monitoreo de temperatura en suelos de difícil acceso. -Investigación geotécnica. - Monitoreo de temperatura en suelos contaminados.
Sensores de Núcleo Caliente	- Respuesta rápida. - Adecuados para suelos húmedos y secos.	- Precisión limitada. -Requieren calibración.	- Agricultura de precisión. - Control de temperatura en viveros. -Investigación en suelos y geotermia.
Sensores Semiconductores	- Bajo costo. - Respuesta rápida.	- Precisión moderada. - Sensibles a la contaminación.	- Monitoreo de temperatura en sistemas de riego. - Control de temperatura en invernaderos. -Investigación agrícola y geotécnica.

1.4.3 SENSORES DE HUMEDAD DEL SUELO

Los sensores de humedad son dispositivos que cuantifican la cantidad de líquido o la humedad relativa en una ubicación específica, lo que facilita la regulación de la humedad del entorno y la temperatura. Las mediciones realizadas por estos sensores se convierten en una señal eléctrica entre ellos tenemos los siguientes sensores [28]:

Tabla 3 Tipos de sensores de humedad para suelo [29]

Tipo de Sensor de Humedad del Suelo	Ventajas	Desventajas	Aplicaciones
Sensores de Resistencia (Higrómetros)	<ul style="list-style-type: none"> -Costo relativamente bajo. - Robustos y duraderos. - Se pueden utilizar en diferentes tipos de suelo. 	<ul style="list-style-type: none"> -Menos precisos en suelos salinos o con alta conductividad -Requieren calibración periódica. 	<ul style="list-style-type: none"> - Agricultura de precisión. - Monitoreo de la humedad en cultivos y suelos. - Control del riego y la irrigación.
Sensores Capacitivos	<ul style="list-style-type: none"> -Buena precisión. -Menor necesidad de calibración. -Adecuados para suelos salinos. 	<ul style="list-style-type: none"> -Costo moderado. -Algunos modelos pueden ser sensibles a la salinidad. 	<ul style="list-style-type: none"> - Agricultura de precisión. - Monitoreo de la humedad en suelos agrícolas y forestales. Investigación ambiental y climatología.
Sensores TDR (Reflectometría en Dominio Temporal)	<ul style="list-style-type: none"> - Alta precisión. - Pueden medir diferentes tipos de suelo. - No requieren contacto con el suelo. 	<ul style="list-style-type: none"> - Costo más alto. -Requieren conocimiento técnico para su instalación y uso. 	<ul style="list-style-type: none"> - Investigación agrícola y geotécnica. - Estudios de hidrología y gestión del agua. - Agricultura de precisión y control de la humedad del suelo.
Sensores Ópticos	<ul style="list-style-type: none"> - Sin contacto físico. - Pueden medir a diferentes profundidades. - Precisos. - Inmunidad al ruido eléctrico. 	<ul style="list-style-type: none"> - Costo más alto. - Sensibles a la contaminación de la óptica. 	<ul style="list-style-type: none"> - Monitoreo de la humedad del suelo en entornos críticos. - Estudios de contaminación de suelos. - Agricultura de precisión en invernaderos y viveros.

<p>Sensores de Núcleo Caliente</p>	<p>Respuesta rápida.</p> <ul style="list-style-type: none"> - Adecuados para suelos húmedos y secos. 	<ul style="list-style-type: none"> - Precisión limitada. - Requieren calibración y mantenimiento. 	<ul style="list-style-type: none"> - Agricultura de precisión y control de la humedad del suelo. - Investigación de suelos y geotermia. - Monitoreo de la humedad en suelos de cultivos y áreas verdes.
<p>Sensores de Semiconductores</p>	<ul style="list-style-type: none"> - Bajo costo. - Respuesta rápida. - Pequeños y compactos. - Baja sensibilidad a la contaminación. 	<ul style="list-style-type: none"> - Precisión moderada. - Sensibles la contaminación. - Pueden requerir calibración periódica. - Menor rango de temperatura que otros tipos de sensores. 	<ul style="list-style-type: none"> - Monitoreo de la humedad del suelo en agricultura. - Control del riego y la irrigación en jardinería. - Investigación en suelos y geología.

1.5 TECNOLOGIAS EN EL SECTOR AGRICOLA

En la agricultura, la tecnología agrícola se utiliza para mejorar la producción de alimentos de manera más eficiente y sostenible. Sus aplicaciones incluyen aumentar la producción de cultivos, mejorar su calidad, optimizar el uso de recursos como el agua y los productos químicos, automatizar las tareas agrícolas, recopilar datos para tomar decisiones informadas, mejorar el desarrollo de cultivos y una gestión agrícola eficiente [30].

1.5.1 TIPOS DE TECNOLOGIAS EN EL SECTOR AGRICOLA

Agricultura de Precisión: La agricultura de precisión aprovecha los sistemas de información geográfica, los sensores y la tecnología GPS para recopilar datos sobre el suelo, el clima y los cultivos. Esto permite a los agricultores tomar decisiones más lógicas sobre temas como la siembra, la fertilización y el riego con el fin de maximizar el uso de recursos [30],[31].

Drones Agrícolas: Los drones son un instrumento volador no tripulado esencial para la agricultura. Permiten inspecciones aéreas de cultivos, identificación de áreas de estrés en las plantas, detección de plagas y enfermedades y supervisión eficiente del crecimiento [30],[31].

Robótica Agrícola: La robótica agrícola va más allá de los drones, incluidos los robots terrestres diseñados para realizar tareas específicas en el campo. Estos robots pueden eliminar la maleza, la cosecha, la clasificación y la siembra de productos agrícolas, lo que reduce la dependencia de la mano de obra humana y mejora la eficiencia [31].

Sensores de Suelo y Clima: Los sensores especializados monitorean el suelo y el clima en tiempo real. Esto ayuda a los agricultores a ajustar adecuadamente la cantidad de agua y fertilizantes utilizados en sus cultivos, lo que podría ahorrar recursos y aumentar los rendimientos [30].

Sistemas de Riego Inteligente: Los sistemas de riego avanzados, basados en sensores y automatización, permiten una distribución de agua más precisa en los campos, evitando desperdicios y mejorando la eficacia general del riego [31].

Bioteología y Modificación Genética: La bioteología ha permitido la creación de cultivos transgénicos que son resistentes a plagas, enfermedades y condiciones climáticas adversas. Esto puede conducir a una mayor producción y una disminución de la necesidad de pesticidas [30].

Aplicaciones Móviles y Software de Gestión Agrícola: Existen numerosas aplicaciones y programas de software diseñados para asistir a los agricultores en la administración de sus operaciones, el control de inventarios, la planificación de la siembra y el riego, así como en el acceso a información meteorológica en tiempo real [30],[31].

Agricultura Vertical: Esta tecnología posibilita el cultivo de plantas en estructuras apiladas o estanterías en ambientes controlados, como invernaderos. Utiliza sistemas de iluminación LED y riego automatizado para optimizar el crecimiento y ahorrar espacio [31].

Blockchain en la Cadena de Suministro Agrícola: La tecnología blockchain se emplea para rastrear y verificar el origen de los productos agrícolas, lo que contribuye a aumentar la transparencia y la confiabilidad en la cadena de suministro [31].

Inteligencia Artificial y Aprendizaje Automático: Estas tecnologías se utilizan para analizar grandes volúmenes de datos agrícolas y proporcionar recomendaciones precisas a los agricultores para mejorar sus prácticas y tomar decisiones más acertadas [31].

1.6 ARQUITECTURAS DE SISTEMAS DE REDES EN EL SECTOR AGRICOLA

Las tecnologías en el sector agrícola pueden utilizar una variedad de arquitecturas entre ellas tenemos:

1.6.1 ARQUITECTURA DE SENSORES Y DISPOSITIVOS IOT (INTERNET DE LAS COSAS)

Los dispositivos utilizados en la agricultura, como los sensores que monitorizan el suelo, la humedad, la temperatura y las condiciones climáticas, generalmente se integran mediante redes de Internet de las Cosas (IoT). Estos dispositivos recopilan información en tiempo real y la transmiten a sistemas de gestión agrícola, donde se procesa para su análisis y para respaldar la toma de decisiones [32].

Las arquitecturas de IoT y sus aplicaciones se exploran en una variedad de entornos, desde el uso en hogares inteligentes hasta el desarrollo de entornos de computación en la nube. En un estudio realizado por Ungurean, Gaitan y Gaitan (2014), se propuso una arquitectura IoT basada en la especificación OPC.NET (Comunicaciones de plataforma abierta) para entornos industriales y edificios inteligentes. Los autores intentan proporcionar una estructura que consta de componentes clave: un servidor de datos y una aplicación de interfaz hombre-máquina (HMI). El servidor recopila datos de la red de sensores y controla actuadores conectados en buses de campo incluyendo la adquisición de datos inalámbricos [32].

Se han desarrollado varios proyectos en torno a la interacción entre IoT y la agricultura; donde se ubica el proyecto desarrollado por la Universidad de Pekin, en donde han diseñado una arquitectura con plataforma para información, gestiona y monitorea la producción agrícola [32].



Figura 3 Internet de las cosas [33]

1.6.2 ARQUITECTURA DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

La Arquitectura de Geoarquitectura de Información del Sistema (SIG) es un conjunto de pautas y estándares que definen el marco organizacional requerido para el despliegue y administración de sistemas que recuperan y visualizan datos espaciales, como la distribución de cultivos y los datos del suelo. Esenciales para la planificación y la toma de decisiones agrarias locales son estos sistemas. Decisiones en la agricultura y otros campos que dependen de la información geográfica se ven más informadas debido a la coherencia y eficacia así por la arquitectura SIG [34].



Figura 4 Mapas SIG en 2D y 3D utilizados para desarrollar y presentar alternativas de diseño y algunos de los análisis empleados como entradas en las decisiones. [35]

1.6.3 ARQUITECTURA DE REDES DE COMUNICACIÓN

El uso de una variedad de redes de comunicación, incluidas las redes móviles, los sistemas Lora WAN y los sistemas satelitales, es necesario para transferir datos de áreas rurales a instalaciones de gestión. La elección de la red correcta depende de factores como la distancia a cubrir y la disponibilidad de recursos. En este contexto, la arquitectura de la red de comunicaciones es esencial para garantizar la seguridad y la eficiencia de la transferencia de datos. Por lo tanto, elegir y diseñar cuidadosamente la arquitectura de red es esencial para lograr una transmisión de datos exitosa en entornos agrícolas y similares [36].



Figura 5 Representación visual de una red [37]

1.6.4 ARQUITECTURA DE ROBÓTICA AGRÍCOLA

La Arquitectura de Robótica Agrícola es un conjunto de principios y pautas que organizan la implementación de robots y tecnologías avanzadas en la agricultura. Su objetivo es automatizar las tareas agrícolas, mejorar la eficiencia y la sostenibilidad, y abarca temas como sensores, control, comunicación y autonomía de los robots agrícolas. Esto ayuda a modernizar las prácticas agrícolas y a administrar eficientemente los recursos del sector [38].



Figura 6 Agricultores robóticos [39]

1.6.5 ARQUITECTURA DE BLOCKCHAIN

La arquitectura de blockchain se centra en la creación y organización de una red específica en el contexto de la cadena de suministro agrícola. Esta red se distingue por su estructura descentralizada y segura, en la que múltiples participantes, así como bloques de datos y contratos inteligentes, juegan un papel crucial. El objetivo principal de esta arquitectura es garantizar que los productos agrícolas sean rastreables y auténticos a lo largo de todas las etapas de la cadena de suministro. Al mismo tiempo, garantiza que la información sea auténtica y transparente durante todo el proceso de distribución agrícola [40].



Figura 7 Estructura Blockchain en la industria agrícola alimentaria [41]

1.6.6 ARQUITECTURA DE SISTEMAS DE GESTIÓN AGRÍCOLA

La arquitectura de sistemas de gestión agrícola es un marco organizativo que guía la incorporación de tecnologías en la gestión de actividades agrícolas, incluida la toma de decisiones y la recopilación de datos. Además, las aplicaciones y software de gestión agrícola usan con frecuencia estructuras de cliente-servidor o web para proporcionar interfaces de usuario amigables y facilitar el acceso a datos y análisis agrícolas. Esto ayuda a los agricultores a tomar decisiones basadas en información y a administrar sus operaciones de manera más eficiente. Tiene como objetivo mejorar la gestión y la productividad agrícolas mediante la integración de tecnologías adecuadas [42].

1.7 DISPÓSIVOS A UTILIZAR EN EL RIEGO DE MAÍZ POR ASPERCIÓN UTILIZANDO IOT

1.7.1 SENSORES DE TEMPERATURA DS18B20

El sensor de temperatura digital Gikfun DS18B20 mide la temperatura en una variedad de aplicaciones. Es resistente al agua y puede transmitir datos y energía utilizando el protocolo One-Wire. Se pueden obtener lecturas precisas de temperatura en grados Celsius con una resolución de hasta 12 bits. El sensor tiene una memoria interna en la que se almacena una dirección única para identificar. Su fuente de alimentación de 3.3V a 5V funciona con plataformas como Arduino y se puede usar en proyectos electrónicos y automatizados. En condiciones adversas, es confiable y versátil gracias a su encapsulación impermeable [43].



Figura 8 Sensor de temperatura "Gikfun DS18B20" [43]

1.7.2 SENSORES DE HUMEDAD DE SUELO HD-38 HIGOMETRO

Este dispositivo ha sido desarrollado con el propósito de monitorear la humedad del suelo o el terreno en el que crecen las plantas. Está compuesto por una sonda que incorpora dos terminales diseñados específicamente para resistir la oxidación. Estos terminales permiten que fluya una corriente eléctrica a través del suelo. Cuando el contenido de agua en el suelo es elevado, se facilita la conducción de electricidad, lo que resulta en una resistencia eléctrica baja. Por otro lado, si el suelo está seco, no se produce conducción eléctrica y, en consecuencia, se registra una resistencia eléctrica alta. Al medir estos valores de resistencia eléctrica, se obtiene el porcentaje de humedad del suelo. Debido a su facilidad de uso, este sensor se convierte en una opción óptima para automatizar el riego en cultivos, plantas y otros contextos similares [44].



Figura 9 Sensor de humedad “HD-38 Higrómetro” [44]

1.7.3 PLACA WIFI LORA DE HELTEC

Las placas WiFi LoRa de Heltec son dispositivos de desarrollo que combinan conectividad WiFi y LoRa para desarrollar aplicaciones IoT. Se utilizan en una variedad de aplicaciones de Internet de las cosas, como el seguimiento de activos y el monitoreo ambiental, y permiten la comunicación inalámbrica de largo alcance y baja potencia. También incluyen un microcontrolador. Para proyectos que requieren eficiencia energética y comunicación inalámbrica en largas distancias entre ellas tenemos la V1, V2, V3 [45].

Tabla 4 Características de las diferentes placas de Heltec Wifi Lora [46]

Característica/Especificación	Heltec WiFi LoRa V1	Heltec WiFi LoRa V2	Heltec WiFi LoRa V3
Microcontrolador	ESP8266	ESP32	ESP32-S3FN8
Módulo LoRa	SX1262	SX1262	SX1262
Frecuencia LoRa	470MHz~510MHz o 863/923MHz	470~510MHz o 863~923 MHz	470~510 MHz o 863~928 MHz
Conexión USB	Micro USB	Micro USB	USB-C
Pantalla OLED	0.96 pulgadas	0.96 pulgadas	0.96 pulgadas
Batería	Conector JST para batería LiPo	Conector JST para batería LiPo	Conector JST para batería LiPo, protección

			contra sobrecarga y USB/batería
Voltaje de Operación	3.7V	3.7V	3.7V,5V
Compatibilidad con Arduino	Sí	Sí	Sí
Compatibilidad con LoRaWAN	Sí	Sí	Sí
Potencia de Procesamiento	Moderada	Mayor	Mayor
Antena Externa	Sí	Sí	Sí
Número de GPIO disponibles	Variado	Variado	Variado
Memoria	128kb SPI FLASH; 16KB internal SRAM	8MB (64M- bits) SPI FLASH; 520KB interna SRAM	ROM de 384 KB; RAM de 512 KB; SRAM RTC de 16 KB; Flash SiP de 8MB

Después de evaluar minuciosamente las características técnicas y las capacidades de las placas Heltec WiFi LoRa V1, V2 y V3, he llegado a la conclusión de que la placa Heltec WiFi LoRa V3 es la mejor elección para mi proyecto de tesis debido a su potente microcontrolador ESP32-S3FN8, que ofrece un mayor rendimiento y capacidad de procesamiento en comparación con las versiones anteriores, V1 y V2. Esto me permitirá ejecutar aplicaciones más avanzadas y complejas para cumplir con los objetivos de mi investigación de manera eficiente.

Es importante destacar que la elección de la placa depende de las necesidades específicas del proyecto, y las otras versiones, como la V1 o V2, pueden ser más adecuadas según el caso. Sin embargo, en mi caso, considerando los requisitos de mi tesis, la Heltec WiFi LoRa V3 se presenta como la opción óptima.

1.7.4 HELTEC WIFI LORA V3

Es una placa de desarrollo que combina capacidades de comunicación WiFi, LoRa y Bluetooth. En su núcleo, se encuentra el chip ESP32-S3FN8, que ofrece un procesador de doble núcleo con frecuencia de hasta 240MHz, lo que lo convierte en una elección ideal para aplicaciones de Internet de las cosas (IoT). El módulo SX1262 habilita la comunicación de largo alcance mediante la tecnología LoRa mediante una antena de resorte y una interfaz IPEX (U. FL) reservada para uso de la misma en la banda de frecuencia de 863-928MHz, la cual es común en redes LoRaWAN. [47].

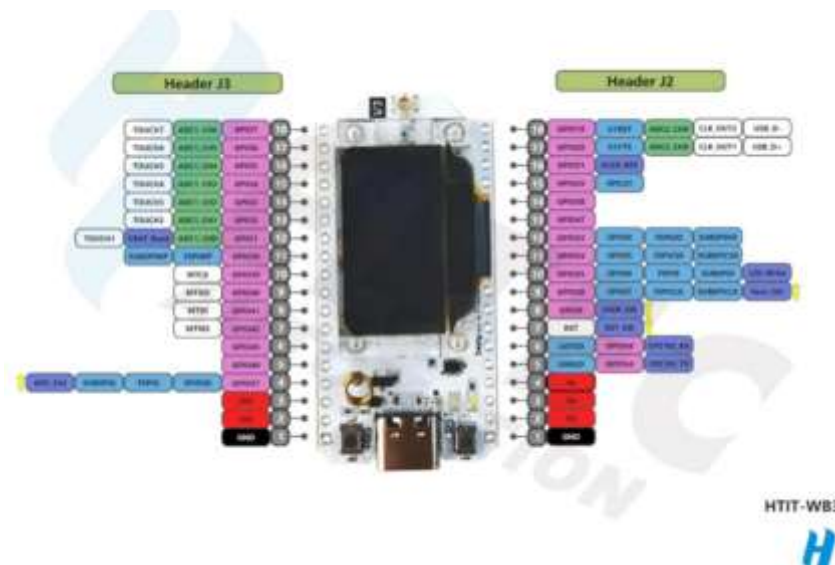


Figura 10 Esquema de pines de la placa Wifi Lora V3 [47]

Además, el ESP32 de este panel permite la comunicación a través de la conexión WiFi utilizando los protocolos TCP (Protocolo de control de transmisión) y UDP (Protocolo de datagramas de usuario). Esto facilita la transferencia eficiente de datos a través de redes IP, lo cual es esencial para muchas aplicaciones de Internet de las cosas [48].



Figura 11 Wifi Lora V3 [48]

1.7.5 MÓDULO RELE 5V

Es un dispositivo que puede enviar una señal de control eléctrico de baja potencia para controlar circuitos eléctricos de alta potencia. Este módulo tiene una bobina de control y un interruptor electromagnético, también conocido como relé. Cuando se energiza la bobina, este último se activa y permite o interrumpe el flujo de corriente en el circuito conectado. Se utiliza en aplicaciones que requieren el control de dispositivos de alto voltaje o potencia, como electrodomésticos, luces y motores [49].



Figura 12 Módulo Relé [50]

Tabla 5 Características del módulo relé de 1 canal de 5VDC con optoacoplador [50]

Característica	Descripción
Relé de 1 canal y 5VDC	Módulo de relé con un único canal de control y voltaje de activación de 5VDC.
Aislamiento fotoeléctrico mediante un optoacoplador	Utiliza un optoacoplador para lograr un aislamiento fotoeléctrico.
Permite seleccionar el trigger high/low	Capacidad de elegir entre activación en alto (high) o en bajo (low).
Permite controlar una gran cantidad de dispositivos en AC y DC	Adecuado para controlar una variedad de dispositivos en corriente alterna (AC) y corriente continua (DC).
Voltaje de alimentación: 5VDC	La fuente de alimentación requerida para el módulo es de 5VDC.
Voltaje señal de entrada: 3.3VDC o 5VDC	Se puede utilizar una señal de entrada de 3.3VDC o 5VDC para activar el relé.
Salida: 250VAC 10A o 30VDC 10A	La salida del relé puede manejar cargas de hasta 250VAC a 10A o 30VDC a 10A.
Led indicador de encendido y led indicador de activación de relé	Incluye LEDs que indican el estado de encendido y la activación del relé.
Permite controlar desde cualquier microcontrolador (Arduino, Raspberry Pi, PIC)	Puede ser controlado por una variedad de microcontroladores, como Arduino, Raspberry Pi y PIC.

1.7.6 VÁLVULA ELECTROMAGNÉTICA DE 12 V

Una válvula electromagnética eléctrica de latón Beduan, diseñada para controlar el flujo de fluidos, como agua o aire, en sistemas automatizados. Su construcción principal es de latón y opera con un voltaje de (12V). Se utiliza en aplicaciones que requieren una regulación precisa del flujo en conductos de medio pulgada de diámetro, como sistemas de riego y aplicaciones industriales. Su funcionamiento se basa en la activación de una bobina eléctrica que controla la apertura o cierre de la válvula [51].



Figura 13 Válvula Electromagnética Beduan 12V 1/2 [51]

1.7.6 CINTA DE AISLANTE 19MM X 18M TRUPER EXPERT

La Cinta Aislante Truper Expert TRU-19954 destaca por su excelente resistencia a la intemperie y condiciones extremas. Hecho de PVC y adhesivo acrílico de alta calidad, es dos veces más fuerte que la competencia y proporciona una adhesión duradera a una variedad de superficies. Mide 19 mm x 18 m y es retardante de llama, lo que lo hace más seguro en una variedad de aplicaciones. Proporcionando opciones confiables y versátiles para una variedad de necesidades de aislamiento [52].



Figura 14 Cinta aislante Truper Expert [52]

1.7.7 FUENTE CONMUTADA 12V/10/P

Esta fuente de alimentación es de tipo conmutada, lo que significa que tiene un diseño eficiente que utiliza tecnología para controlar la salida de voltaje de manera más efectiva que las fuentes de alimentación lineales convencionales [53].



Figura 15 Fuente conmutada 12V/10A [53]

Tabla 6 Tabla Especificaciones de la fuente 12V-10A [53]

Característica	Descripción
Tipo de Fuente de Alimentación	Conmutada
Tensión de Alimentación	230 V AC
Voltaje de Salida	12 V DC
Ajuste de Tensión de Salida	10.2 V a 15 V
Carga Máxima de la Fuente de Alimentación	10 A
Potencia de la Fuente de Alimentación	120 W
Número de Salidas	2 unidades
Protecciones	Contra cortocircuitos y sobrecargas
Temperatura de Funcionamiento	-10 °C a 50 °C
Humedad Relativa de Funcionamiento	20 % a 90 %
Peso	0.452 kg
Dimensiones	200 x 98 x 42 mm

1.7.8 MANGUERA POLIETILENO NEGRA CED-40 DE 1/2" 13 MM

La "Manguera Negra CED-40 de 1/2" 13 mm" es un conducto flexible para riego que cumple con la norma CED-40. Con un diámetro interno de 13 milímetros, tiene un diámetro exterior de medio pulgada. En sistemas de riego, esta manguera se utiliza comúnmente para canalizar el agua hacia áreas que necesitan ser irrigadas, como jardines o cultivos. La construcción negra lo hace adecuado para el uso en el exterior porque es duradero y resistente

a los cambios climáticos y al desgaste. La conformidad con la norma CED-40 indica que la manguera satisface estándares de calidad y rendimiento específicos para aplicaciones de riego [54].



Figura 16 Manguera de Riego 1/2" 13mm [54]

1.7.9 ASPERSOR PLASTICO EXCEL WOBBLER 1/2

El aspersor Xcel-Wobbler se destaca por su capacidad para generar un patrón de aplicación instantáneo uniforme sobre áreas extensas. Dado que usa una menor cantidad de agua que los aspersores convencionales, ofrece mayor protección contra las heladas y una irrigación más eficiente para los cultivos, su eficiencia en el uso del agua es notable [55].



Figura 17 Aspersor Excel Wobbler 1/2 [55]

Tabla 7 Tabla Características del aspersor Excel Wobbler 1/2 [55]

Características	Descripción
Contrapeso	Reduce vibración para un funcionamiento suave y estable.

Parte móvil única	Garantiza mayor durabilidad.
Conexiones posibles	3/4" o 1/2" NPT macho.
Caudales	De 0,78 a 6,97 GPM (De 177 a 1.583 litros/hora).
Presiones de funcionamiento	De 10 a 25 PSI (De 0,69 a 1,72 Bar).
Diámetros de riego	De 8,1 a 16,2 metros.
Baja deriva por viento y mínima pérdida de agua	Ofrece baja deriva por viento y pérdida mínima por evaporación a bajas presiones.
Boquillas codificadas por colores	Facilita la identificación del tamaño, garantizando el mantenimiento correcto del orificio.
Garantía de mantenimiento del orificio durante 5 años	Boquillas garantizadas para mantener el tamaño correcto del orificio durante cinco años.

1.7.10 CAJA DE DERIVACIÓN DE 105X105X55MM

La caja de conexiones Plexo es una solución robusta y resistente con cierre rápido de un cuarto de vuelta. Con grado de protección IP55 y resistencia al impacto IK07, se puede utilizar en temperaturas de -25°C a +40°C. Dispone de siete entradas regulables de 4 a 25 mm y una tapa que protege los tornillos de fijación y puede utilizarse en gran variedad de instalaciones eléctricas y electrónicas. Sus dimensiones son 105x105x55 mm y su color es gris (RAL7035), por lo que es una buena opción para proteger conexiones eléctricas en diversos entornos. [56].



Figura 18 Caja de derivación de 105x105x55mm [56]

1.7.11 CAJA DE DERIVACIÓN DE DIMENSIONES: 80X80X45MM

La caja de conexiones Plexo cuadrada resistente a la intemperie tiene clasificación IP55 e IK07. Está fabricado en PVC gris (RAL7035) con medidas de 80x80x45 mm. Cumple con los estándares de seguridad Clase II y cuenta con una cubierta con cierre a presión con siete entradas ajustables para cables de 4 a 20 mm de diámetro, así como tapones que ocultan los tornillos de montaje. Protege conexiones eléctricas y electrónicas para una amplia gama de aplicaciones desde -25°C a +40°C, desde entornos comerciales hasta residenciales e industriales [57].



Figura 19 Caja de derivación de 80x80x45mm [57]

1.7.12 CABLE DE COBRE AISLADO PARALELO TIPO SPT 300V

Conductor de cobre electrolítico suave aislado con PVC resistente. Se presenta como un cable paralelo flexible que tiene características como resistencia a la humedad, retardante de llama y ausencia de metales pesados. Según el Código Eléctrico Nacional, su aplicación abarca la conexión de electrodomésticos, extensiones y conexiones colgantes, lo que lo convierte en un cable portátil adecuado para ambientes no demasiado exigentes. Con una temperatura de operación de hasta 60 °C y un voltaje de 300 V, también es adecuado para su uso en entornos secos y húmedos [58].



Figura 20 Cable aislado paralelo tipo SPT [59]

1.8 APLICACIÓN WEB

1.8.1 MODELO DE CLIENTE-SERVIDOR

La arquitectura de software conocida como modelo cliente-servidor divide una aplicación en dos partes principales: el cliente y el servidor. El servidor maneja la lógica comercial y los datos, mientras que el cliente es la parte que interactúa con el usuario y presenta la interfaz de usuario. Estas dos partes se comunican a través de una red utilizando protocolos estándar como HTTP o TCP/IP. Este enfoque permite distribuir la carga de trabajo, facilita la escalabilidad y se utiliza en aplicaciones de escritorio, web, móviles y otros dispositivos que acceden a servicios del servidor [60], [61].

1.8.2 SERVIDOR TCP

Un cliente y un servidor TCP son componentes esenciales para la comunicación a través de redes de computadoras e Internet. El servidor funciona como un punto de escucha en un puerto determinado y espera conexiones de clientes. Por otro lado, el cliente proporciona una dirección IP y un número de puerto específicos para iniciar una conexión. Luego, ambos comparten información de manera segura. El protocolo TCP es confiable y permite una comunicación bidireccional eficiente porque garantiza una entrega precisa y sin errores de datos [62], [63]

1.8.3 AMAZON ELASTIC COMPUTE CLOUD (AMAZON EC2)

Es una plataforma de cómputo en la nube proporcionada por Amazon Web Services (AWS). Este servicio permite crear y gestionar servidores virtuales, conocidos como instancias.

La característica más destacada de EC2 es su escalabilidad, lo que significa que puedes ajustar la capacidad de cómputo según las necesidades cambiantes de tu aplicación. Esto resulta en un ahorro de costos significativo, ya que solo pagas por la capacidad que utilizas. Además, proporciona opciones de seguridad y control de tráfico para garantizar la protección de tus instancias y facilita la administración de recursos de cómputo en la nube [64].

1.8.4 SERVICE WEB REST (REPRESENTATIONAL STATE TRANSFER)

Es un estilo arquitectónico para el diseño de sistemas de software que se basa en principios y restricciones específicas. En un servicio web REST, los recursos (como objetos, datos o funciones) se identifican mediante URLs, y las operaciones se realizan utilizando los métodos HTTP estándar (GET, POST, PUT, DELETE, etc.). La comunicación entre el cliente y el servidor se realiza mediante solicitudes y respuestas HTTP, y la representación de los recursos puede estar en formato JSON o XML, entre otros [65].

1.8.5 SENCHA EXT JS

Es un marco de desarrollo de código abierto para la creación de aplicaciones web ricas y escalables en JavaScript. Ext JS proporciona una amplia gama de componentes de interfaz de usuario, herramientas de desarrollo y capacidades de manipulación de datos que facilitan la creación de aplicaciones web empresariales complejas y atractivas. Se destaca por su enfoque en la creación de interfaces de usuario ricas y su arquitectura orientada a objetos [66].

CAPÍTULO 2: DISEÑO DEL SISTEMA DE RIEGO POR ASPERCIÓN

2.1 UBICACIÓN DEL PROYECTO DE RIEGO POR ASPERSIÓN

El sistema se implementará en Vilcabamba, a 20 km de la provincia de Loja, en coordenadas geográficas latitud: 4°13'3.33 "S y longitud: 79°13'29.12"O. Este terreno tiene un área de 10 m de ancho por 10m de largo cuya propiedad es el Sr. Juan Castillo.



Figura 21 Ubicación del terreno [Autor]

2.2 LINEAS DE VISTA DEL TERRENO SEMBRADO

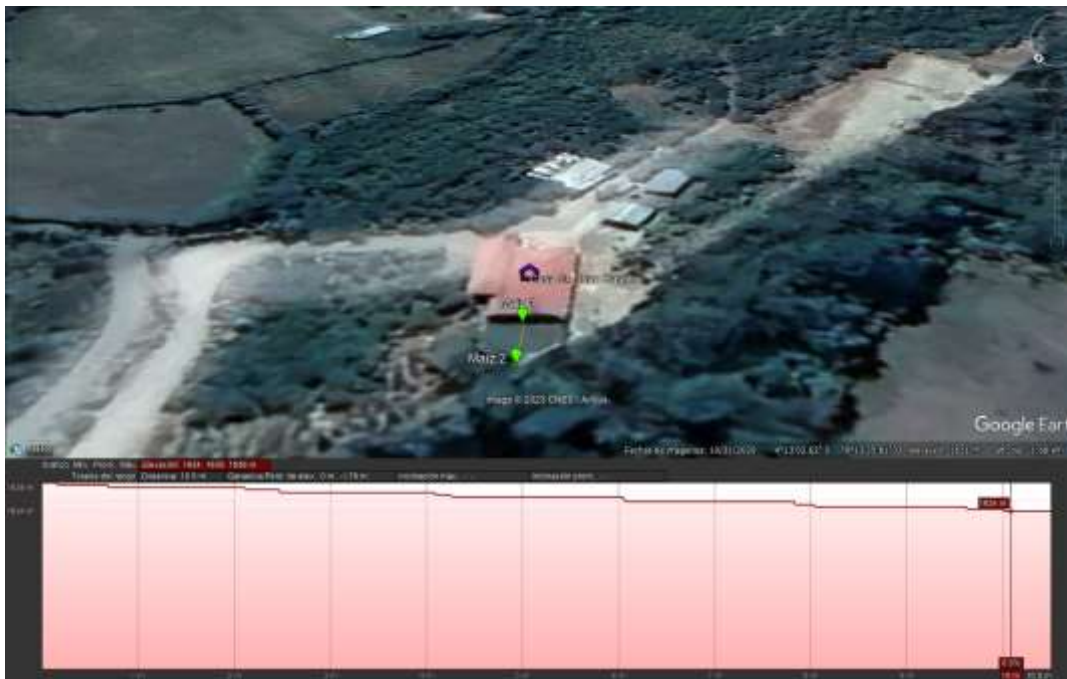


Figura 22 Línea de vista de área sembrada [Autor]

2.3 LINEA DE VISTA DE LA CASA DEL USUARIO Y EL TERRENO LORA



Figura 23 Línea de vista de la casa del usuario y el terreno [Autor]

2.4 ESQUEMA DEL SISTEMA DE RIEGO POR ASPERSIÓN

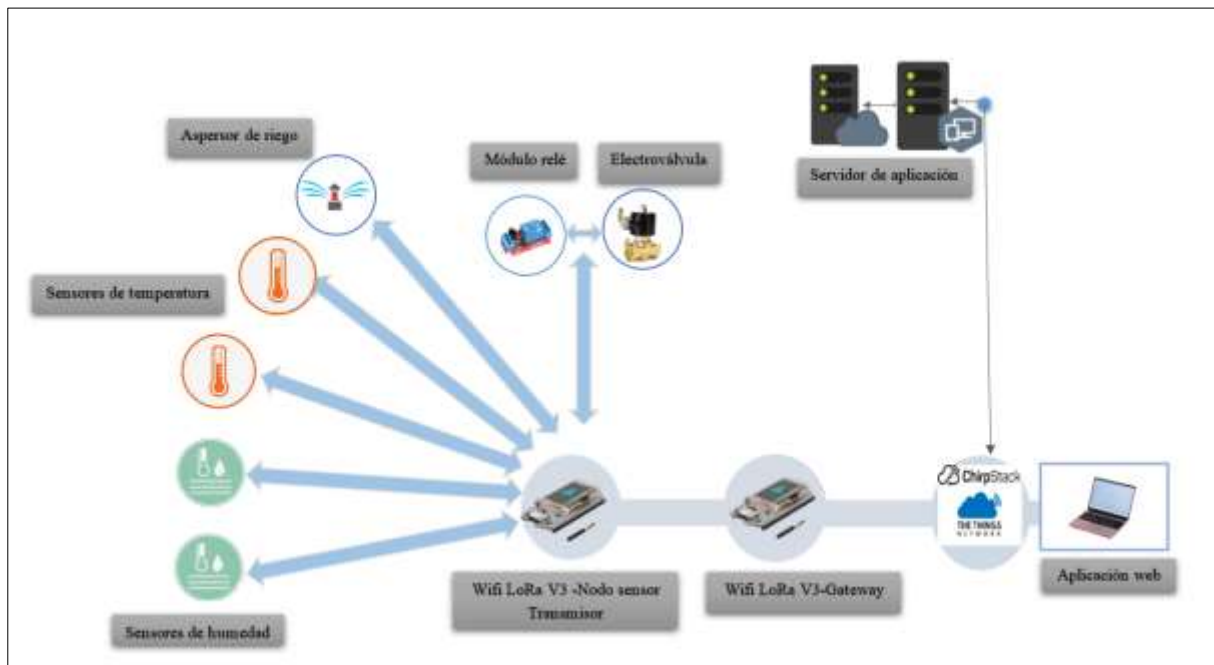


Figura 24 Esquema del sistema de riego por aspersión [Autor]

2.5 DIAGRAMA DE BLOQUES PARA CALCULAR EL NÚMERO DE NODOS DE TRANSMISIÓN Y RECEPCIÓN, ASÍ COMO EL NÚMERO DE SENSORES DE HUMEDAD Y TEMPERATURA

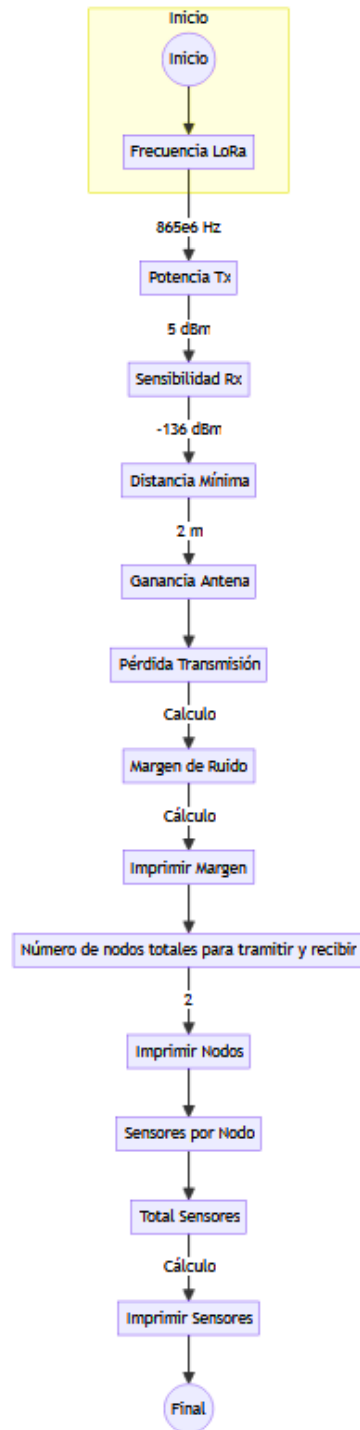


Figura 25 Diagrama de bloques para el cálculo de nodos y sensores [Autor]

```
Command Window
>> untitled3
Margen de ruido: 103.53 dB
Número de nodos totales para transmisión y recepción: 2
Sensores de humedad en nodo transmisor: 2
Sensores de temperatura en nodo transmisor: 2
Sensores de humedad en nodo receptor: 0
Sensores de temperatura en nodo receptor: 0
fx >>
```

Figura 26 Total de Nodos y Sensores de Humedad-Temperatura [Autor]

2.6 HERRAMIENTAS DE SOFTWARE

2.6.1 EMBARCADERO DELPHI

Desarrollado por Embarcadero Technologies, es un entorno de desarrollo de software eficiente que proporciona a los programadores un conjunto completo de herramientas. Utiliza el lenguaje de programación Object Pascal y destaca por su enfoque en el desarrollo rápido de aplicaciones (RAD). Esto permite a los desarrolladores crear soluciones de software de forma rápida y eficiente. Delphi proporciona una amplia gama de componentes visuales, bibliotecas y funciones integradas para crear fácilmente aplicaciones para diversas plataformas, como Windows, macOS, iOS y Android. Además, su versatilidad ha sido probada en el desarrollo de aplicaciones de negocio, soluciones cliente-servidor e implementaciones integradas en el campo de las tecnologías de la información [67].



Figura 27 Delphi [68]

2.6.2 ARDUINO IDE

Es un software que proporciona una interfaz para programar placas Arduino. Está diseñado específicamente para escribir, cargar y administrar código que controla la funcionalidad y el funcionamiento de las placas Arduino. El entorno facilita la programación de proyectos electrónicos, permitiendo a los usuarios desarrollar código que permitan controlar sensores, actuadores y otros componentes sin necesidad de hardware físico [69].



Figura 28 Arduino [69]

2.7 NODO TRANSMISOR

El nodo transmisor se encarga de medir la temperatura y la humedad a través de dos sensores Dallas DS18B20 y dos sensores de humedad análogos HD-38 HIGOMETRO. Luego de eso, procesa estos datos y envía información al receptor a través de comunicación LoRa, que utiliza radio frecuencia para comunicarse directamente con el receptor.

El transmisor realiza las siguientes tareas:

Configuración Inicial:

- Se configuran parámetros LoRa, como frecuencia de 865000000 MHz, potencia de salida de transmisión de 5dBm.
- Se inicializan los sensores de temperatura, humedad y otros periféricos.

Adquisición de Datos:

- Se leen los valores de temperatura y humedad de los sensores.
- Se controla un relé basándose en las lecturas de temperatura y humedad.
- Se actualiza la interfaz de la placa Lcd con los valores de temperatura, humedad y el estado del relé.

Comunicación LoRa:

- Se construye un mensaje que incluye temperatura, humedad y estado del suelo sacando las temperaturas promedio y las humedades promedio de cada uno de los sensores
- El mensaje se envía a través de LoRa.
- Se comunica con el receptor para enviar los datos y a su vez tomar cualquier orden que se envíe desde la página web para realizar la activación manual del riego.

Conexión con el receptor:

- Se establece la conexión con el receptor que permite visualizar los datos enviados.
- Se actualiza la interfaz web con información en tiempo real.

También se tomó las siguientes condiciones para la programación de los sensores de temperatura y humedad:

Tabla 8 Tabla Condiciones de temperatura y humedad [Autor]

CONDICIONES DE HUEMDAD		
Estado	Escala	Relé
Exceso de Humedad	$1200 \geq \text{Humedad} \leq 1800$	0
Suelo Humedo	$1801 > \text{Humedad} \leq 3020$	0
Suelo Seco	$3021 > \text{Humedad} \leq 4000$	1
Riego Necesario	$\text{Humedad} > 4002$	1
CONDICIONES DE TEMPERATURA		
Temperatura alta	$\text{Temperatura} \geq 30$	0
Temperatura baja	$\text{Temperatura} \leq 20$	0

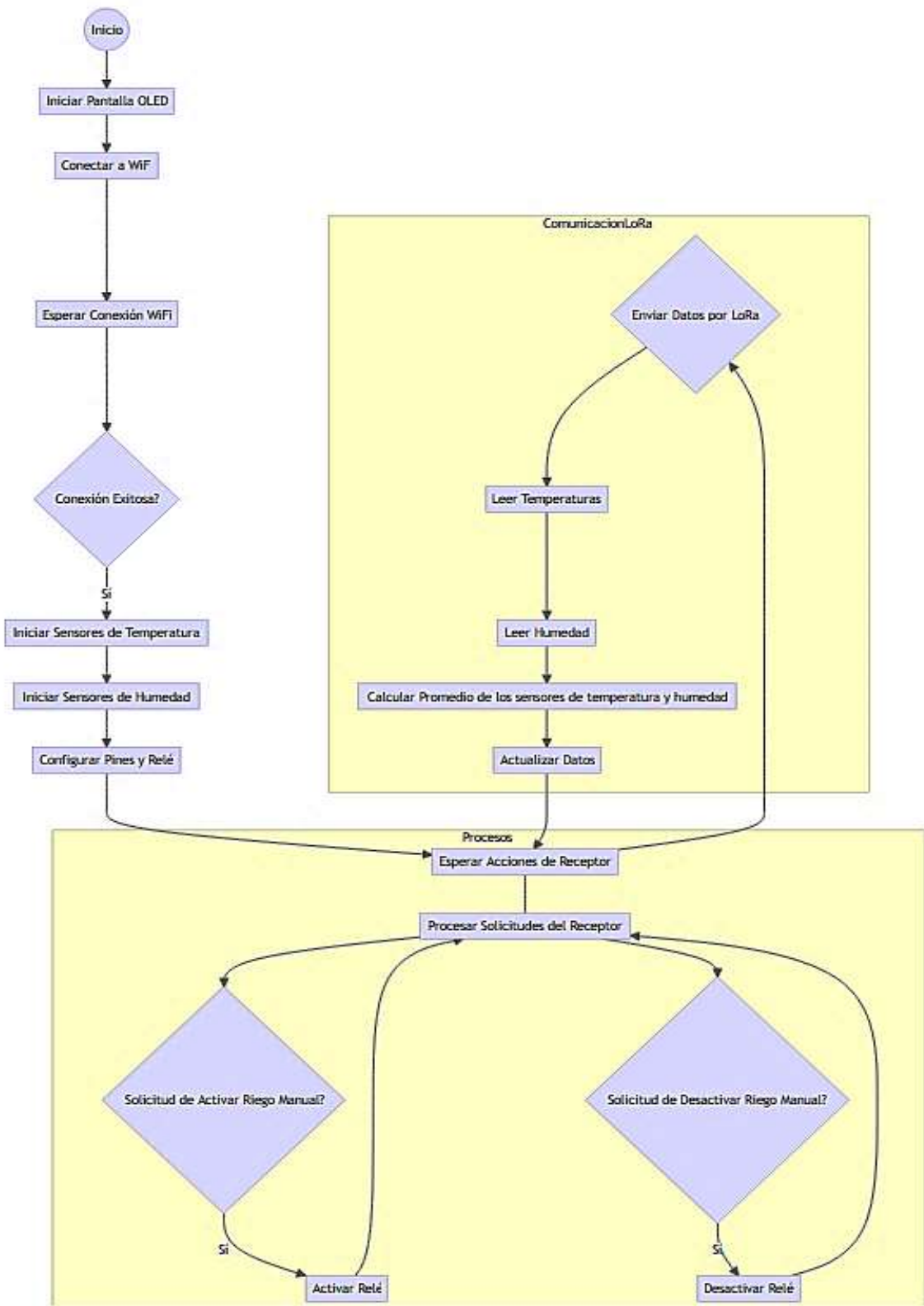


Figura 29 Diagrama de bloques Transmisor [Autor]

2.8 DISEÑO PCB PARA EL NODO TRANSMISOR

En el nodo transmisor, donde se encuentran las conexiones de sensores, alimentación y relé, se vio la necesidad de desarrollar un diseño de placa de circuito impreso (PCB) que establezca un medio sólido para la conexión eléctrica de todos los componentes tanto como para los sensores de temperatura, humedad, relé y que proporciona una estructura fija que garantiza la correcta adquisición de las señales. Las pistas conductoras permitirán conectar de manera eficiente los pines de los componentes, facilitando el flujo de corriente a lo largo del circuito.

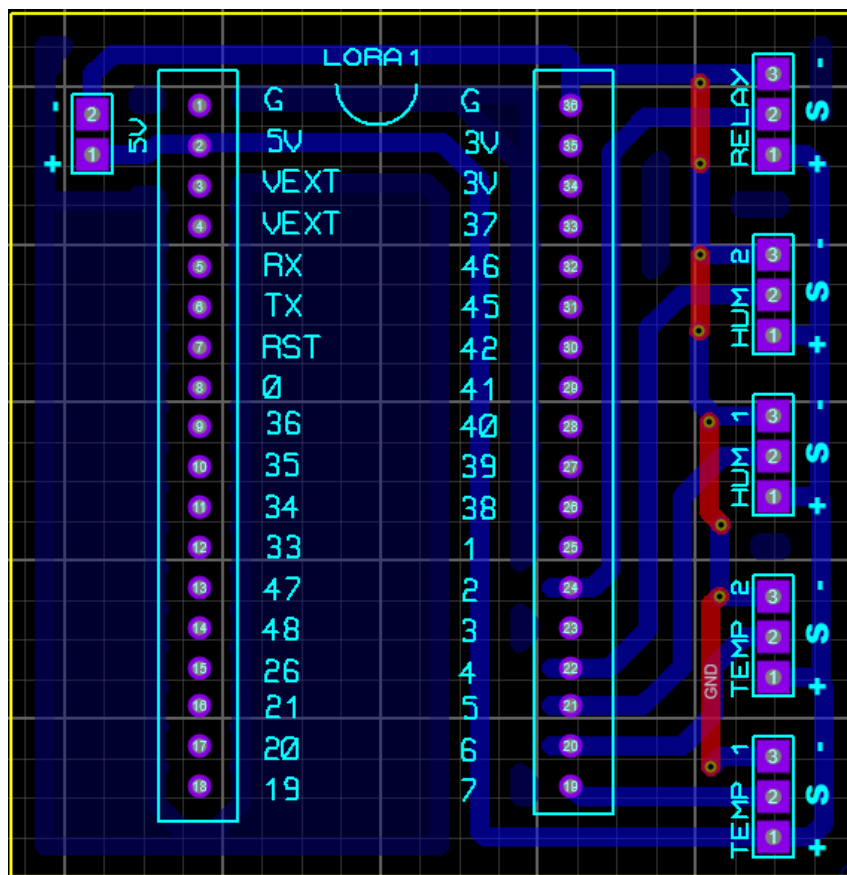


Figura 30 Capa interior del PCB [Autor]

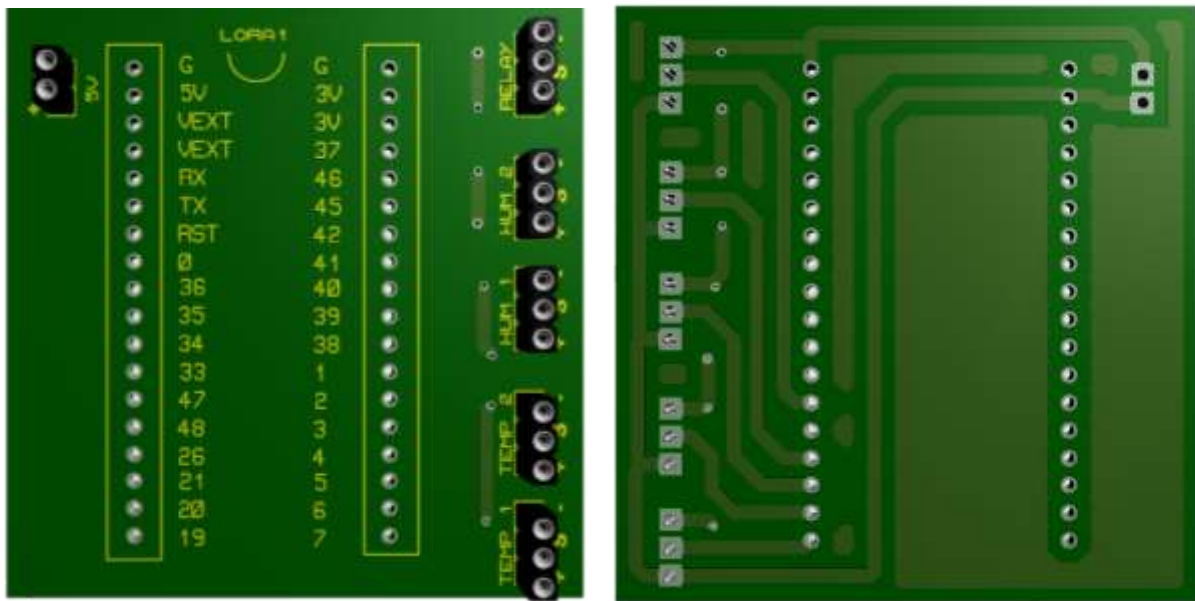


Figura 31 Diseño de placa PCB en 3D [Autor]

Para poder realizar las conexiones de sensores de humedad, temperatura, fuente de alimentación y relé, nos dirigimos ahora hacia la esencia lógica del sistema. Mediante el esquema electrónico, se presenta de manera visual la interconexión precisa de cada componente.

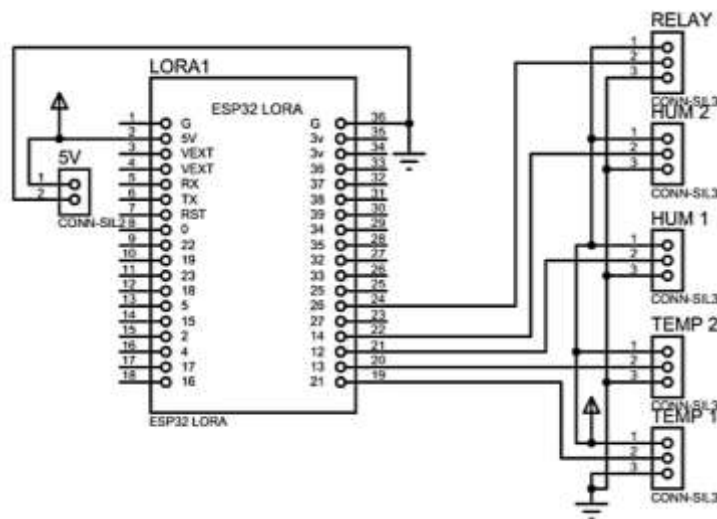


Figura 32 Esquema electrónico [Autor]

2.9 NODO RECEPTOR

El nodo receptor se encarga de recibir los datos enviados por el transmisor. Posteriormente, mostrará los datos en una pantalla OLED y facilitará el manejo del relé mediante una interfaz web.

El receptor realiza las siguientes tareas:

Configuración Inicial:

- Se configuran parámetros LoRa de manera similar al transmisor.
- **Recepción de Datos:**
- El receptor está en modo de espera para recibir datos a través de LoRa.
- Cuando recibe datos, extrae la temperatura, humedad y el estado del suelo y los envía al servidor cada tres minutos.
- Puede recibir desde la página web una variable **a** de activar el riego manual y una **i** de desactivar el riego. Este le enviará esta información al transmisor el cual permitirá la utilización del riego manual.
- **Comunicación TCP:**
- Se establece una conexión con el servidor TCP.
- Se envían los datos de temperatura y humedad al servidor.

Interfaz OLED:

- Se actualiza la pantalla OLED con los valores de temperatura y humedad recibidos.

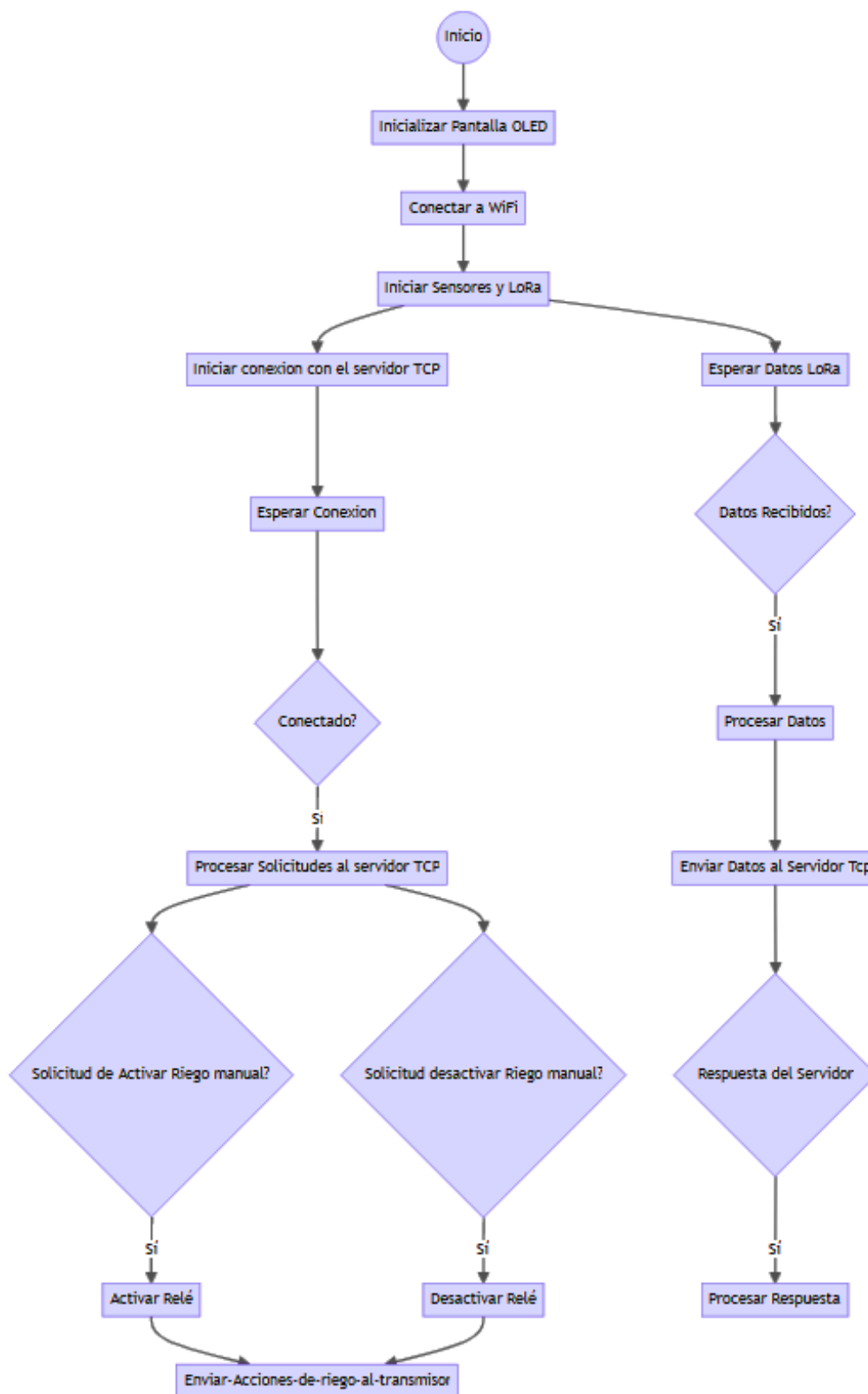


Figura 33 Diagrama de bloques Receptor [Autor]

2.10 DISEÑO DE LA APLICACION WEB

2.10.1 ESPECIFICACIONES PARA EL DESARROLLO DE LA APLICACIÓN WEB

La creación de la aplicación web surge de la necesidad de monitorear y controlar el riego de maíz, presentando los datos de temperatura y humedad recopilados por los sensores en el nodo transmisor en una interfaz gráfica. Para realizar el desarrollo eficiente de la aplicación, es crucial contar con una base de datos que sirva como estructura organizativa y de almacenamiento de los mismos. Asimismo, se necesita un servicio de hosting que acoja tanto la aplicación como la base de datos. Este servicio de alojamiento desempeñará un papel central al facilitar la interacción entre la aplicación y la base de datos, posibilitando la recepción y almacenamiento de datos en las tablas correspondientes.

2.10.2 DIAGRAMA DE BLOQUES DE LA APLICACIÓN WEB

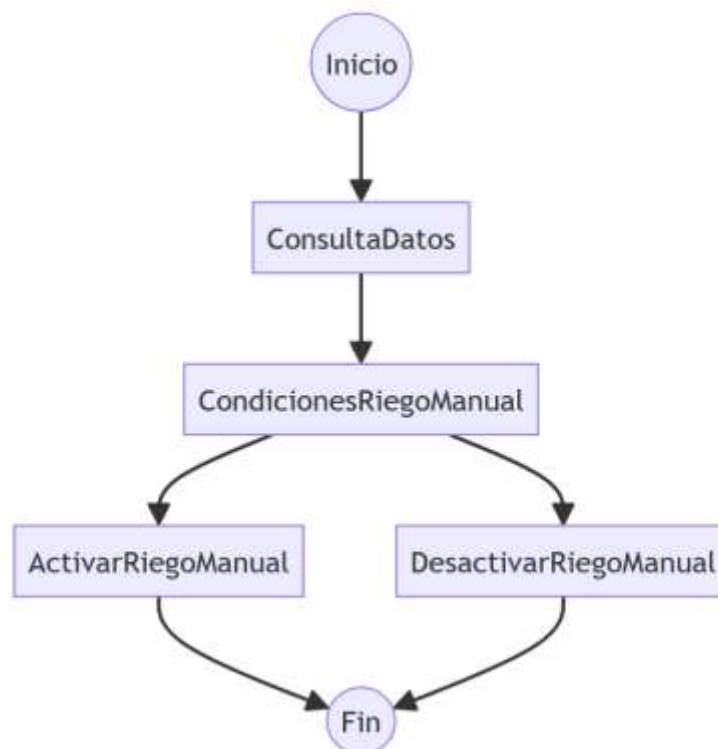


Figura 34 Diagrama de bloques de la aplicación web [Autor]

2.10.3 BASE DE DATOS

Para recibir datos de los sensores, se ha implementado un servidor en AWS (Amazon Web Services) el cual aloja el servidor web y el servidor TCP. Posteriormente, estos datos se envían a una base de datos MariaDB, que se encarga de almacenar información sobre los estados de los sensores de temperatura y humedad como los valores de los sensores, estado del suelo, fecha en la que se recibió el dato.

En este escenario, el servidor web AWS sirve como punto de acceso para la interfaz del usuario y permite el control remoto del sistema y la visualización, el servidor TCP está diseñado para recibir datos de los sensores y servir como un intermediario entre el nodo receptor y la base de datos.

Los datos se reciben mediante una conexión TCP en el puerto 20000 y se vinculan con la dirección IP del servidor 3.141.41.125, siendo almacenados en una base de datos MariaDB.

En la Figura 35 se puede observar los sensores envían datos desde el nodo transmisor al nodo receptor que están delimitados por punto y coma. Estos datos llegan al servidor como una cadena completa, la cual se descompone en variables según los valores de temperatura y humedad.

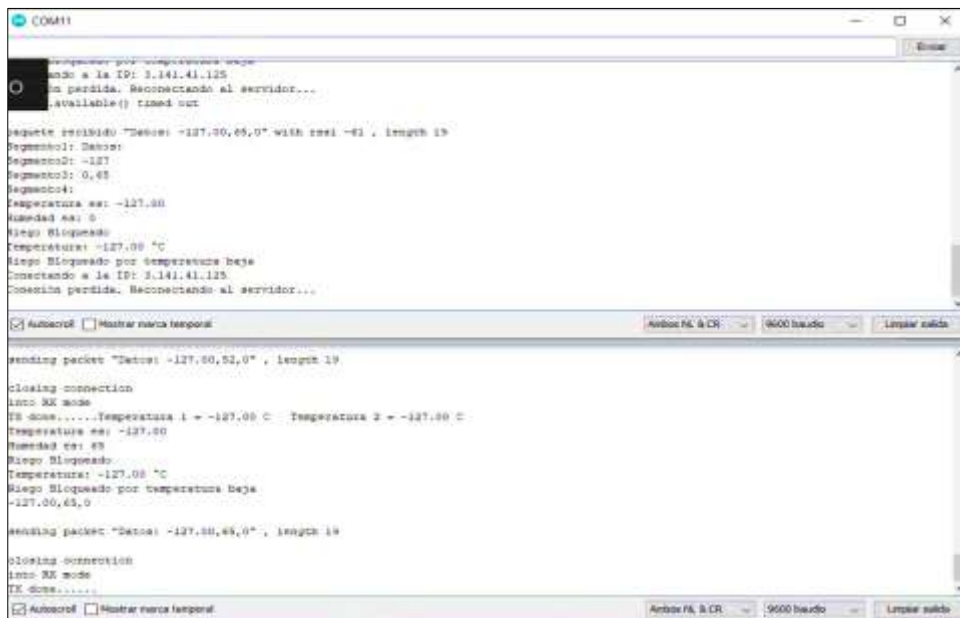


Figura 35 Envío de datos de los nodos transmisor y receptor [Autor]

```

Welcome Page  appweb_riego  Main  Da_consumo
70  F_cliente.BaseURL := urldesactivar;
    F_req.Method := rmGET;
    F_req.Execute;
end;

- Eprocedure TDat_consumo.obtenerDatos;
var
    tmp, resp : string;
    lista : TJSONArray;
    dsensor : TJSONObject;
    i, j : integer;
    fecha : string;
    id_envio : string;
    contenido : string;
    arrstatus : TJSONArray;
    status : TJSONObject;
    locid, deliveryStatus : string;
    dsfecha : string;
    f_zona : TDateTime;
begin
90  mt_datos.EmptyDataSet;
    F_cliente.BaseURL := urldatos;
    F_req.Method := rmGET;
    F_req.Execute;
    resp := F_req.Response.Content;
    lista := TJSONObject.ParseJSONValue(resp) as TJSONArray;
    if Assigned(lista) then begin
100     for i := 0 to lista.Count - 1 do begin
        dsensor := lista.Items[i] as TJSONObject;
        mt_datos.Append;
        mt_datos.ID.Value := dsensor.GetValue<string>('ID');
        fecha := dsensor.GetValue<string>('FECHA');
        fecha := StringReplace(fecha, '\', '/', [rfReplaceAll]);
        f_zona := StrToDateTime(fecha);
        mt_datos.FECHA.Value := f_zona;
        mt_datos.TEMPERATURA.Value := dsensor.GetValue<double>('TEMPERATURA');
        mt_datos.HUMEDAD.Value := dsensor.GetValue<double>('HUMEDAD');
        mt_datos.Post;
    end;
end;

```

Figura 36 Obtención de datos de los sensores [Autor]

Se puede observar en la Figura 36 la programación en la base de datos de la página web donde mostrará la fecha, los datos de temperatura y los datos de humedad recibidos mediante el servidor Tcp.

```

- Eprocedure TDat_base.activarAsensor(id: string);
var
    list_temp : TList<Tsensor>;
    i : integer;
begin
begin
try
list_temp := conexiones.LockList;
80  for i := 0 to list_temp.Count - 1 do begin
    if list_temp[i].id = id then begin
        list_temp[i].context.Connection.Socket.Write('a');
        Dat_globales.registrar_log('Comando a enviado IP ' + list_temp[i].context.Connection.ToString);
    end;
end;
finally
conexiones.UnlockList;
end;
end;
90  Eprocedure TDat_base.DataModuleCreate(Sender: TObject);
begin
try
- servidor.Active := true;
conexiones := TThreadList<Tsensor>.Create;
Dat_globales.registrar_log('Servidor activo', 'trp.txt');
except on e:Exception do begin
Dat_globales.registrar_error('activando servidor', e)
end;
end;
end;
- E function TDat_base.datosSensor(f_inicio, f_fin: TDateTime): string;
var
    tmp : string;
    jv : TJsonValue;
    ja : TJSONArray;
    jo_interno : TJSONObject;
begin
110  ja := TJSONArray.Create;
    Q_datos.ParamByName('inicio').Value := f_inicio;
    Q_datos.ParamByName('fin').Value := f_fin;
    Q_datos.Execute;
    Q_datos.First;
    while not Q_datos.EOF do begin
        jo_interno := TJSONObject.Create;
        jo_interno.AddPair('FECHA', DateTimeToStr(Q_datos.FECHA.Value));
        jo_interno.AddPair('ID', '0001');
        jo_interno.AddPair('TEMPERATURA', TJSONNumber.Create(Q_datos.TEMPERATURA.Value));
120  jo_interno.AddPair('HUMEDAD', TJSONNumber.Create(Q_datos.HUMEDAD.Value));
        jo_interno.AddPair('ESTADO', Q_datos.ESTADO.Value);
        jo_interno.AddPair('RITGO', Q_datos.RITGO.Value);
        ja.Add(jo_interno);
        Q_datos.Next;
    end;
end;

```

Figura 37 Envío de datos desde el servidor a la aplicación web [Autor]

Para la visualización de los datos en la aplicación web se realiza la respectiva programación el cual el servidor Tcp tiene que estar activo para obtener los datos respectivos y poderlos distribuir de acuerdo a la interacción con cada uno de los botones que se han puesto en la interfaz gráfica, en la Figura 37 muestra la programación para la configuración y visualización en cada uno de los botones con los datos de temperatura, humedad, estado actual del suelo además incluye líneas de código que permite la selección de las fechas de inicio y de fin permitiendo el comportamiento de los sensores.

CODIGO	FECHA	TEMPERATURA	HUMEDAD	ESTADO	ABIDO
946	2023-11-11 11:03:04	26.2	4.095	Wago conectado	1
947	2023-11-11 11:04:00	26.8	4.095	Wago conectado	1
948	2023-11-11 11:05:13	26.5	4.095	Wago conectado	1
949	2023-11-11 11:06:38	26.5	4.095	Wago conectado	1
950	2023-11-11 11:08:41	26.3	4.095	Wago conectado	1
951	2023-11-11 11:08:54	26.6	4.095	Wago conectado	1
952	2023-11-11 11:07:08	26.5	4.095	Wago conectado	1
953	2023-11-11 11:07:21	26.6	4.095	Wago conectado	1
954	2023-11-11 11:07:28	26.8	4.095	Wago conectado	1
955	2023-11-11 11:07:39	26.6	4.095	Wago conectado	1
956	2023-11-11 11:08:02	26.5	4.095	Wago conectado	1
957	2023-11-11 11:08:15	26.5	4.095	Wago conectado	1
958	2023-11-11 11:08:28	26.6	4.095	Wago conectado	1
959	2023-11-11 11:08:42	26.6	4.095	Wago conectado	1
960	2023-11-11 11:08:58	26.6	4.095	Wago conectado	1
961	2023-11-11 11:09:10	26.6	4.095	Wago conectado	1
962	2023-11-11 11:09:23	26.6	4.095	Wago conectado	1
963	2023-11-11 11:09:39	26.6	4.095	Wago conectado	1
964	2023-11-11 11:09:48	26.6	4.095	Wago conectado	1
965	2023-11-11 11:10:01	26.3	4.095	Wago conectado	1
966	2023-11-11 11:10:11	26.6	4.095	Wago conectado	1
967	2023-11-11 12:30:41	27.2	4.095	Wago conectado	1
968	2023-11-11 12:30:18	27.1	4.095	Wago conectado	1
969	2023-11-11 12:31:00	27.2	4.095	Wago conectado	1
970	2023-11-11 12:32:29	27.2	4.095	Wago conectado	1
971	2023-11-11 12:33:08	27.1	4.095	Wago conectado	1
972	2023-11-11 12:33:50	27.1	4.095	Wago conectado	1
973	2023-11-11 12:40:20	27.1	4.095	Wago conectado	1
974	2023-11-11 12:41:09	27.1	4.095	Wago conectado	1
975	2023-11-11 12:41:59	27.1	4.095	Wago conectado	1

Figura 38 Datos de los sensores y estado en HeidiSQL [Autor]

Como se muestra en la Figura 38 se usó HeidiSQL, un software que ayudo a la administración de base de datos donde se podrán visualizar los valores de temperatura, humedad, el estado del suelo y fecha en el transcurso del día los cuales ayudarán a generar información para que sea visualizada en la aplicación web y también las gráficas respectivas del comportamiento de los sensores.

```

var
  datos: TIDBytes;
  cadena: string;
  dsensor : TStringList;
  msg,suelo : string;
  temp , hum :double;
  riego : boolean;
begin
try
218  suelo := '';
  if AContext.Connection.IHandler.Readable then
  AContext.Connection.IHandler.ReadBytes(datos,-1, False);
  cadena := Trim(BytesToString(datos,InoyTextEncoding_UTF8));
  dsensor := TStringList.Create;
  dsensor.Delimiter := ',';
  dsensor.StrictDelimiter := true;
  dsensor.DelimitedText := cadena;

  temp := StrToFloat(trim(dsensor.Strings[0]));
  hum := StrToFloat(trim(dsensor.Strings[1]));
220  if temp = 0 then exit;
  C.insertar.ParamByName('FECHA').Value := now;
  C.insertar.ParamByName('TEMPERATURA').Value := temp;
  C.insertar.ParamByName('HUMEDAD').Value := hum;
  riego := false;
  if ( temp >= 20 ) and (temp <=30) then begin
  if (hum >= 1200) and (hum <= 1600) then begin
  suelo := 'Exceso de humedad';
  end;
230  if (hum > 1600) and (hum <= 2200) then begin
  suelo := 'suelo humedo';
  end;
  if (hum > 2200) and (hum <= 2800) then begin
  suelo := 'Suelo seco';
  end;
  if (hum > 2800) then begin
  suelo := 'Riego necesario';
  riego := true;
  end;
240  end else begin
  if temp < 20 then
  suelo := 'Temperatura baja'
  else
  if temp > 30 then
  suelo := 'Temperatura alta';
  end;
  C.insertar.ParamByName('ESTADO').Value := suelo;
  C.insertar.ParamByName('RIEGO').Value := riego;
  C.insertar.Execute;
250  // Dat_globales.registrar_log('Datos guardados ' + IntToStr(mt_datos.RecordCount)+ ' '-dsensor.Strings[0],'mt_datos.txt');
  except on E:Exception do
  Dat_globales.registrar_log('execute '+E.Message)
  end;
end;
end;

```

Figura 39 Condiciones de los sensores [Autor]

Para poder obtener una buena configuración y distribución de los datos de los sensores de humedad y temperatura se realiza una programación que nos permita clasificar de la mejor manera la información que estamos obteniendo desde el nodo receptor en la Figura 39 tenemos la declaración de la variables como lo es datos, riego, temp y msg,suelo ,se establecen los valores de las escalas de los sensores de temperatura menor a 20°C temperatura baja y valores mayores a 30 °C temperatura alta y en la humedad como exceso de humedad, suelo seco , suelo húmedo y riego necesario cada uno con su respectivas escalas para una mejor distribución de nuestros datos en la aplicación web.

```

+ Result := Js.ToString;
+ end;
+
+ procedure TDat_base.DesactivarAspersor(id: string);
139 var
+   list_temp: TList<TSensor>;
+   i: Integer;
+ begin
+   try
+     list_temp := conexiones.LockList;
+     for I := 0 to list_temp.Count-1 do begin
+       if list_temp[I].id = id then begin
+         list_temp[I].context.Connection.Socket.Write('i');
+         Dat_globales.registrar_log('Aspersor Inactivo');
140       end;
+     end;
+   finally
+     conexiones.UnlockList;
+   end;
+ end;
+
+ procedure TDat_base.ServidorConnect(AContext: TIdContext);
+ var
150 datos: TIdBytes;
+   cadena: string;
+   list_temp: TList<TSensor>;
+   sensor: TSensor;
+   dsensor: TStringList;
+   msg: string;
+ begin
+   if AContext.Connection.IDHandler.Readable then
+     AContext.Connection.IDHandler.ReadBytes(datos,-1, False);
+   cadena := Trim(BytesToString(datos, IndyTextEncoding_UTF8));
160 try
+     try
+       sensor.fecha := now;
+       sensor.id := '0001';
+       sensor.context := AContext;
+       list_temp := conexiones.LockList;
+       list_temp.Add(sensor);
+       Dat_globales.registrar_log('Conexion creada ', 'tcp.txt');
168     finally
+       conexiones.UnlockList;
+     end;
170 except on E:exception do
+   Dat_globales.registrar_log('conexion_tcp_'+E.Message);
+ end;
+ end;
+
+ procedure TDat_base.ServidorDisconnect(AContext: TIdContext);
+ var
+   list_temp: TList<TSensor>;
+   i: Integer;
180 begin

```

Figura 40 Conexión al receptor [Autor]

Para enviar las interacciones entre la aplicación web con respecto a la activación y desactivación de los aspersores procedemos a realizar la programación respectiva como muestra en la Figura 40 donde para inactivar el aspersor list_temp.Com =1 es igual a la variable **id** y si se encuentra esta coincidencia se envía una **i** que quiere decir que el aspersor está desactivado y consecuentemente lo mismo pasa con la activación de los aspersor pero en este caso se da cuando se envía la letra **a**, luego se procede a la conexión con el servidor Tcp el cual recibe los valores de temperatura y humedad a su vez también se encarga de enviar el carácter **a** o **i** mediante el servidor Tcp al nodo receptor y este lo envíe al transmisor y tome acciones sobre el rele y pueda activar o desactivar el riego por aspersión.

2.10.4 INTERFAZ GRAFICA DE LA APLICACIÓN WEB

Para el diseño de la interfaz gráfica se tomó en cuenta las necesidades y preferencias que le gustaría al agricultor visualizar con elementos claros e intuitiva que facilite el manejo e interacción en la aplicación web.

La interfaz gráfica muestra en tiempo real el estado de los sensores que monitorean la temperatura y la humedad del cultivo de maíz. Además, contiene dos botones que permitirán activar o desactivar el riego por aspersión manualmente en caso de ser necesaria para el suelo sembrado.

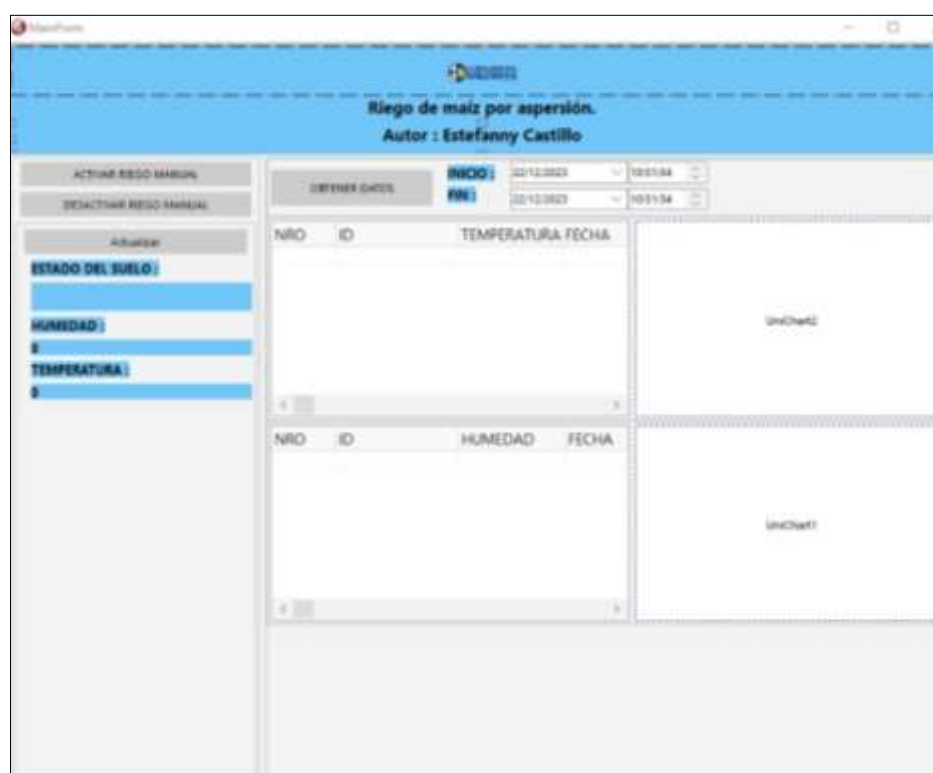


Figura 41 Diseño de la interfaz gráfica de la aplicación web [Autor]

Contiene reportes gráficos donde se podrá visualizar el comportamiento de los sensores basándose en una fecha de inicio, fecha final, hora de inicio y hora final seleccionada en el calendario integrado en la interfaz, de esta forma permitiendo analizar el rendimiento de los sensores y un botón actualizar que tendrá la función de refrescar la información del estado de los sensores su es necesario , todo esto se dará en un periodo de prueba de 24 días desde la implementación del sistema como se puede visualizar en la Figura 41.


```

interface
-
-
uses
SysUtils, Classes, REST.Types, REST.Client, Data.Bind.Components,
Data.Bind.ObjectScope, FireDAC.Stan.Intf, FireDAC.Stan.Option,
FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS, FireDAC.Phys.Intf,
FireDAC.DApt.Intf, FireDAC.Stan.StorageBin, Data.DB, FireDAC.Comp.DataSet,
10 FireDAC.Comp.Client, ison;

type
[+] TDat_consumo = class(TDataModule)
-
-
R_cliente: TRESTClient;
R_req: TRESTRequest;
R_res: TRESTResponse;
mt_datos: TFDMemTable;
mt_datosID: TStringField;
mt_datosHUMEDAD: TFloatField;
20 mt_datosTEMPERATURA: TFloatField;
mt_datosFECHA: TDateTimeField;
procedure DataModuleCreate(Sender: TObject);
private
{ Private declarations }
-
public
{ Public declarations }
procedure obtenerDatos();
function activar():boolean;
function desactivar():boolean;
30 end;
const
urldatos = 'http://3.141.41.125:8895/datos';
urlactivar = 'http://3.141.41.125:8895/activar';
urldesactivar = 'http://3.141.41.125:8895/desactivar';
- function Dat_consumo: TDat_consumo;

[+] implementation
-
{$R *.dfm}

```

Figura 42 Programación para controlar los datos de la interfaz grafica [Autor]

En el presente código de la Figura 42 se define un módulo de datos llamado TDat_consumo el cual incluye componentes relacionados con accesos a servicios RESTful, en el cual se puede manipular datos de memoria y acceso a base de datos utilizando FireDAC. El tiene un cliente REST (R_cliente) que utiliza un conjunto de componentes FireDACMemoryTable (mt:datos) para almacenar diferentes tipos de datos como cadenas, números de punto flotantes y fechas. También tenemos funciones y procedimientos para realizar diversas operaciones como obtención de datos, activación, desactivación y actualización de datos y por último establece constantes que contienen URL para acceder a servicios REST específicos en el servidor, para la obtención de datos, activar, desactivar y actualizar.



Figura 43 Aplicación web [Autor]

A continuación, en la figura 43 se puede observar la interfaz gráfica desde la aplicación web con los datos del estado del suelo, las gráficas de acuerdo a la fechas de inicio y final en este caso se ha escogido para graficar desde el 28/12/2023 al 02/01/2024 con sus respectivas horas de inicio luego dando clic en el botón obtener datos se obtendrán las gráficas respectivas, también se muestran los datos recopilados durante esos días en 17 páginas las cuales se pueden ir cambiando para visualizar el comportamiento de los sensores y por último los botones de activación/desactivación de riego manual que podrán ser utilizados si así lo requiere el usuario.

CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

3.1 ARMADO DE NODO TRANSMISOR

En el nodo transmisor como se observa en la Figura 43, se ocupó una placa wifi Lora de la marca Heltec, una tarjeta PCB que se diseñó previamente con el fin de albergar de una manera más estética y fijar los sensores de temperatura DS18B20, sensores de humedad HD-38, relé, y una fuente de alimentación de 5v. Las conexiones desde la placa PCB hacia los sensores se

hicieron con jumper macho-macho y para proteger el equipo se utilizó una caja de conexiones Plexo 105x105x55mm con propiedades resistentes al clima, alta temperatura y humedad el cual también permite distribuir por sus aberturas laterales los cables de los sensores y a su vez acoplar la antena que viene incorporada en la placa wifi Lora.

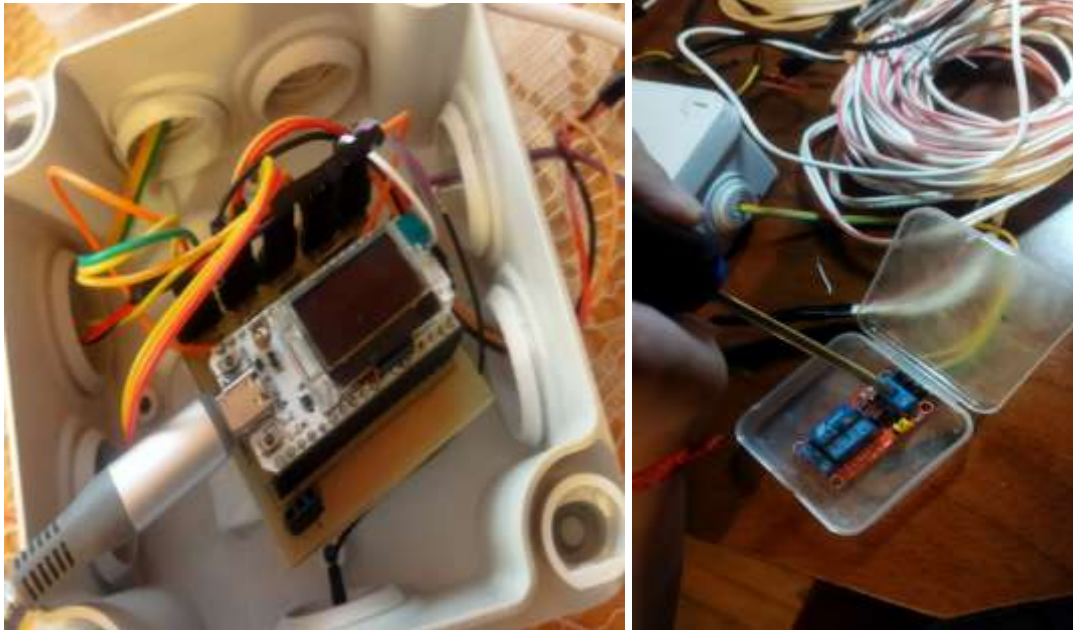


Figura 44 Montaje de nodo transmisor [Autor]

Para los cables que sobresalen del nodo transmisor a los cuales van conectados los sensores de humedad, temperatura, fuente de alimentación y rele se utilizó la cinta Truper Expert TRU-19954, resistente a la humedad y altas temperaturas esto garantizará una sólida protección y durabilidad al sistema de riego por aspersión como se puede observar en la Figura 44.



Figura 45 Protección de cables del nodo transmisor [Autor]

Con el fin de proporcionar una mayor protección a los equipos contra las condiciones adversas del clima, se construyó una caseta de policarbonato como se muestra en la Figura 45 con dimensiones 60 cm de largo por 15 cm de ancho y para el techo una plancha de 70 cm inclinada para albergar el nodo transmisor, así como las fuentes de alimentación y el relé.



Figura 46 Caseta del nodo transmisor [Autor]

Una vez construida la caseta integramos de la mejor manera los equipos en su interior, como se puede visualizar en la Figura 46 el nodo transmisor, rele y las fuentes de alimentación de 5v y 12v. Para facilitar distribución de los sensores conectados al nodo se realizaron aberturas a los laterales de la caseta con dimensiones 5x5 cm estas aberturas permitirán el manejo eficiente de los cables en el terreno del sembrío de maíz.



Figura 47 Vista interna desde la caseta del nodo transmisor [Autor]

En la Figura 47, se aprecia la válvula electromagnética alimentada a 12V el cual se encuentra conectada a una manguera Negra CED-40 de 1/2" de 12m por el cual pasa un caudal de agua proveniente de una llave de agua y se une a un aspersor Xcel-Wobbler. Para ajustar los dos laterales de la manguera se utilizó abrazaderas para manguera 1/2 pulgada y conectores plásticos que permitirán la conexión entre estas dos.



Figura 48 Válvula electromagnética y funcionamiento con el relé [Autor]

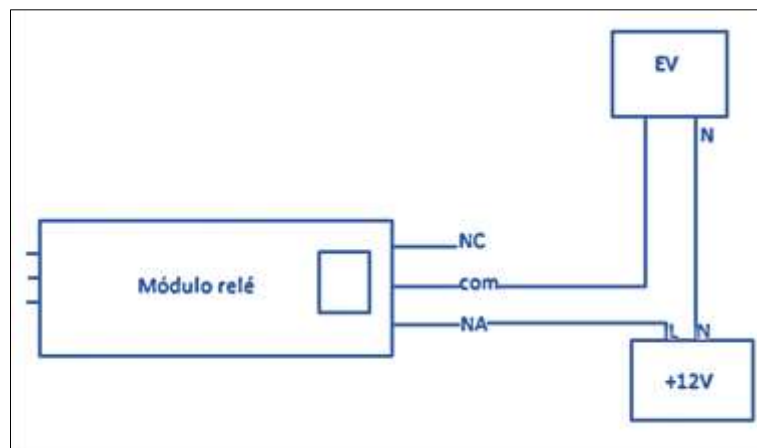


Figura 49 Conexión de la electroválvula a la fuente y al relé [Autor]

En la presente Figura 48 se puede observar la conexión del módulo rele donde el **com** va conectado directamente a la electroválvula (cable rojo) y el NA que hace referencia al contacto normalmente abierto se conecta directamente a la línea de 12V de la fuente de alimentación y por último empalmamos los neutros de la fuente con el neutro de la electroválvula.

3.2 ARMADO DEL NODO RECEPTOR

Para el nodo receptor como se visualiza en la Figura 50, se utilizó una placa WiFi Lora de la marca Heltec la cual ira conectada a una fuente de alimentación de 5V y protegida por una caja de conexiones Plexo 80x80x45mm. Su función principal es recopilar los datos transmitidos por radiofrecuencia desde el transmisor y luego enviar la información al servidor TCP y, finalmente, a una base de datos.



Figura 50 Montaje de nodo receptor [Autor]

En la presente Figura 51, se muestra el nodo receptor desde el exterior de la caja, el mismo que será ubicado en el interior de la casa del señor Juan Castillo a una distancia de 2m del trasmisor la alimentación será directamente desde un toma corriente utilizando un cargador con voltaje de 5V.



Figura 51 Vista externa del nodo receptor [Autor]

3.3 UBICACIÓN DE LOS SENSORES DE TEMPERATURA Y HUMEDAD

Para la colocación de los sensores de humedad y temperatura en el terreno, se consideró una inclinación de 2 grados aproximadamente en un terreno de 10x10m. Se llevó a cabo una distribución en forma de "V", combinando y ubicando los dispositivos de manera proporcional en toda el área. Este diseño permite adquirir con precisión los valores necesarios.



Figura 52 Distribución de sensores en el terreno [Autor]

3.4 PRESUPUESTO

Tabla 9 Tabla Presupuesto [Autor]

Componentes	Cantidad	Precio	Total
Placas Lora Wifi 32 (Heltec)	2	25\$	50\$
Sensor de humedad de higrómetro	2	14\$	28\$
Sensor Temperatura DS8B20	2	4.49\$	8.98\$
Válvula electromagnética	1	21\$	21\$
Modulo rele de 5v	1	3.66\$	3.66
Fuente conmutada de 12v	1	11\$	11\$
Caja de conexiones Plexo cuadrada	2	5\$	10\$
Cable paralelo flexible	50m	12.50\$	12.50\$
Aspersor	1	2.50\$	2.50\$
Cinta Aislante Truper Expert TRU-19954	4	1.30\$	5.20\$
Manguera	30m	1.20\$	36\$
Codos, conectores etc.	-	-	10
Impresión de placa Pcb	1	15\$	15\$
Mano de obra	1	450\$	450\$
Total			663.84 \$

3.5 INGRESO A LA APLICACIÓN WEB

Para acceder a la aplicación web, se inicia abriendo un navegador convencional como Google Chrome o Mozilla Firefox. Posteriormente, se introduce la dirección web específica (<http://3.141.41.125:8077/>) en la barra de direcciones. Una vez en la página principal dedicada al Riego de maíz por aspersión, se accede a la pantalla principal donde observara la siguiente Figura 53 donde el usuario puede monitorear en tiempo real el estado del suelo y variables climáticas, como la temperatura y la humedad. Además, tiene la capacidad de visualizar las gráficas que representan el comportamiento de los sensores según las fechas seleccionadas. Asimismo, se le brinda la opción de activar o desactivar el riego manual, según las necesidades del usuario.



Figura 53 Aplicación Web [Autor]

3.6 RESULTADOS

En el funcionamiento del sistema se obtuvo los siguientes resultados, donde se puede evidenciar el comportamiento del sistema del riego por aspersión y también el estado de los sensores de humedad, temperatura.

En la siguiente Tabla 9 podemos verificar el porcentaje que equivale cada estado de humedad.

Tabla 10 Tabla Porcentajes de cada estado de humedad [Autor]

Intervalo de Humedad	Escala Original (1200-5000)	Porcentaje en la Escala Original (Original)	Porcentaje Ajustado (Ajustado)
Exceso de Humedad	1200 a 1800	$\frac{1800 - 1200}{5000 - 1200} \times 100 =$ 15.79%	$\frac{15.79}{118.84} \times 100 =$ 13.34%
Suelo Húmedo	1081 a 3020	$\frac{3020 - 1081}{5000 - 1200} \times 100 =$ 51.03%	$\frac{51.03}{118.84} \times 100 =$ 42.96%
Suelo Seco	3021 a 4000	$\frac{4000 - 3021}{5000 - 1200} \times 100 =$ 25.76%	$\frac{25.76}{118.84} \times 100 =$ 21.68%
Riego Necesario	4002 en adelante	$\frac{5000 - 4002}{5000 - 1200} \times 100 =$ 26.26%	$\frac{26.26}{118.84} \times 100 =$ 22.02%
Porcentaje total		118.84%	100%

Definición de la Escala Original:

- Se estableció una escala original que abarca desde 1200 hasta 5000, con 1200 como el valor mínimo y 5000 como el valor máximo de humedad.

Definición de Intervalos Significativos:

- Identifico intervalos específicos en esta escala para categorizar los niveles de humedad, como Exceso de Humedad, Suelo Húmedo, Suelo Seco y Riego Necesario.

Cálculos Rigurosos de Porcentajes Originales:

- Se realizo cálculos precisos para determinar los porcentajes originales de cada Intervalo utilizando la fórmula:
$$\frac{\text{Longitud del intervalo}}{\text{Longitud Total de la Escala Original}} \times 100$$

Verificación de la Suma Total de Porcentajes Originales:

- La suma total de los porcentajes originales es 118.84%.

Introducción de la Fórmula de Ajuste Proporcional:

- Fórmula de ajuste que garantiza la preservación de las proporciones Originales:

$$\frac{\text{Porcentaje original}}{\text{Suma Total de porcentajes Originales}} \times 100$$

Esta fórmula asegura que la escala ajustada tenga una suma total del 100%.

Justificación del Ajuste y Normalización:

- La justificación se basa en la importancia de mantener relaciones proporcionales y normalizar los datos para expresarlos de manera efectiva en una escala universal del 0 al 100%.

La fórmula de ajuste de porcentajes se integra para proporcionar una representación más comprensible de los niveles de humedad del suelo. Su aplicación mejora la interpretación de los datos y facilita la comunicación efectiva de los hallazgos en el marco de la investigación.

A continuación, el análisis del funcionamiento del sistema de riego por aspersión que se dio por 25 días en el cual se pudo realizar las pruebas de riego manual y también verificar el funcionamiento de los sensores de humedad y temperatura.

Los datos visualizados en la aplicación son adquiridos cada 3 minutos lo que nos da un total de 20 muestras por hora y en el día 480 muestras el cual nos permitirá analizar las siguientes graficas que se podrán observar el estado de los sensores tanto de temperatura como de humedad, en el eje de las x tenemos el número de muestras y en el eje de las y los valores variables de los sensores.

Semana del 09/12/2023 al 13/12/2023

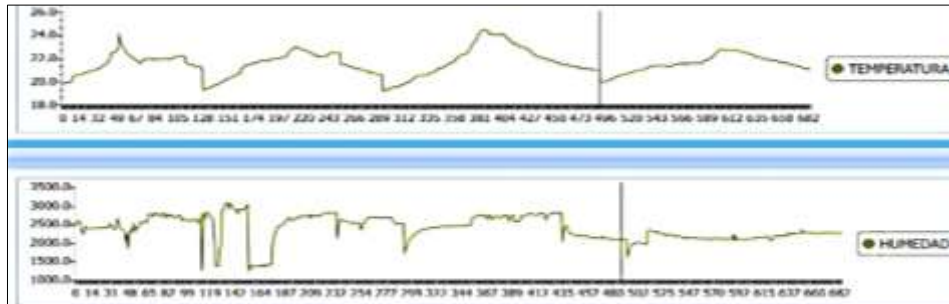


Figura 54 Funcionamiento en la semana 09/12/2023 al 13/12/2023 [Autor]

En el siguiente Figura 54 con datos de los sensores de temperatura y humedad, cada una representando el comportamiento de dichos parámetros. En este análisis, se observa que el sensor de temperatura comienza con un valor de 20 °C por la mañana, lo que provoca el bloqueo del sistema de riego. En cuanto al sensor de humedad, se mantiene en la escala de 3020 durante la tarde. En el transcurso de la tarde, la irrigación se activa al superar la escala de 3021, acompañada por una temperatura de 23 °C. El día culmina con una temperatura de 24 °C y un suelo húmedo.

Semana del 13/12/2023 al 17/12/2023

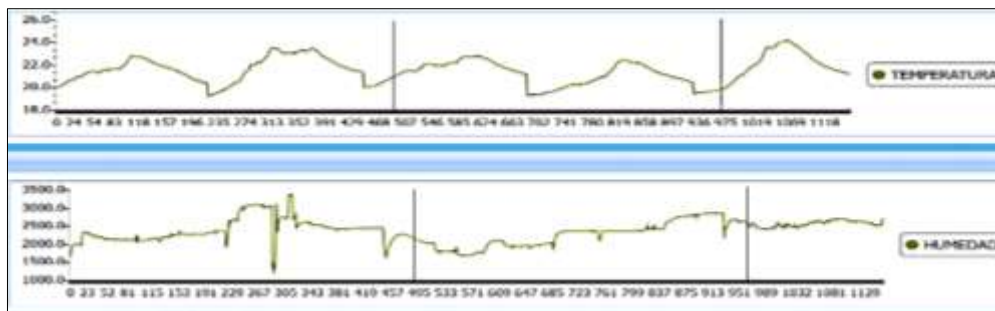


Figura 55 Funcionamiento en la semana 13/12/2023 al 17/12/2023 [Autor]

Los sensores exhiben en la Figura 55 un patrón durante la mañana, con una temperatura de 20 °C que ocasiona el bloqueo del sistema de riego. Por la tarde, se registra un estado de suelo húmedo de 1999, y el riego por aspersión se activa al superar la escala 3500, indicando que el suelo se encuentra seco, con una temperatura de 24 °C durante todo el periodo vespertino. Posteriormente, el riego se desactiva debido al estado normal del suelo durante la noche y madrugada, donde vuelve a bloquearse debido a las bajas temperaturas del suelo.

Semana del 17/12/2023 al 21/12/2023

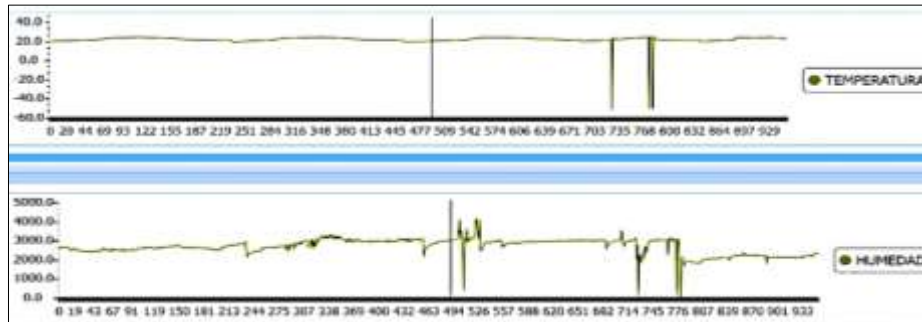


Figura 56 Funcionamiento en la semana 17/12/2023 al 21/12/2023 [Autor]

En la Figura 56, se aprecia que el riego permanece bloqueado durante gran parte de los dos días, debido a las bajas temperaturas que se mantienen en el rango de 20°C. En cuanto a los sensores de humedad, inician con una escala de 2099, ubicándose dentro del rango de suelo húmedo a lo largo del día. Este comportamiento se atribuye al inicio de la temporada de invierno en estos días, influyendo en los valores registrados por los sensores.

Al inicio del segundo día, en la madrugada, los sensores muestran una escala correspondiente a suelo seco, pero el riego no se activa debido a que la temperatura del suelo es inferior a 20°C. A mediodía, las intensas lluvias elevan la escala de los sensores, indicando un exceso de humedad, y la jornada culmina con suelo húmedo y riego bloqueado debido a la temperatura.

Semana del 21/12/2023 al 25/12/2023

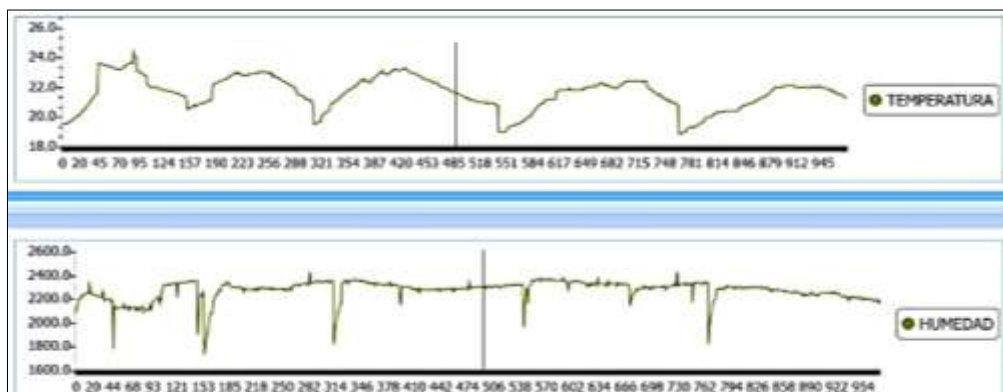


Figura 57 Funcionamiento en la semana 21/12/2023 al 25/12/2023 [Autor]

En estas semanas, los sensores iniciaron su operación registrando una temperatura de 19.5°C y una humedad de 2025, ubicada en el rango de suelo húmedo. Este nivel de humedad persistió casi durante todo el periodo, variando también hacia un estado excesivo de humedad. En cuanto a la temperatura, experimentó variaciones a lo largo del día, alcanzando valores de 23°C, 24°C y 22°C.

Al día siguiente, el suelo comenzó con un estado húmedo y se presentó un bloqueo en el riego debido a la baja temperatura, la cual fluctuó entre 22°C y 20°C a lo largo del día, llegando a su punto más bajo al final de la noche y durante la madrugada. Estos datos fueron registrados desde las 8:15 a. m. hasta las 7:00 p. m como se puede visualizar en la Figura 57.

Semana del 25/12/2023 al 29/12/2023

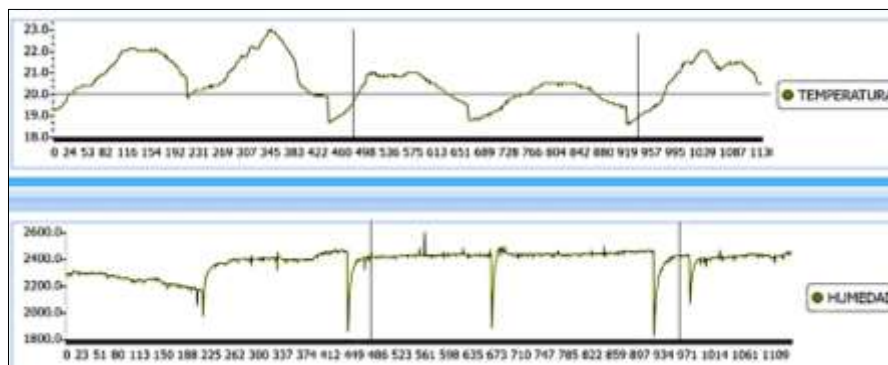


Figura 58 Funcionamiento en la semana 25/12/2023 al 29/12/2023 [Autor]

En estas penúltimas semanas en la Figura 58, los sensores han reflejado un comportamiento dinámico, comenzando con una temperatura de 19.5°C y una humedad que se encuentra en el rango de suelo húmedo. A medida que avanza la tarde, la temperatura oscila hasta alcanzar los 23°C, para luego descender a 19°C en la noche. El sensor de temperatura indica un exceso de humedad durante la noche, resultado de lluvias intensas, ya que estos datos fueron recolectados durante la temporada de invierno.

En el segundo día, el riego inicia bloqueado debido a la baja temperatura de 19.8°C, que se mantiene constante a lo largo del día. El rango del sensor de temperatura indica un exceso de agua en el suelo, manteniendo las condiciones de bloqueo del riego a lo largo de la jornada.

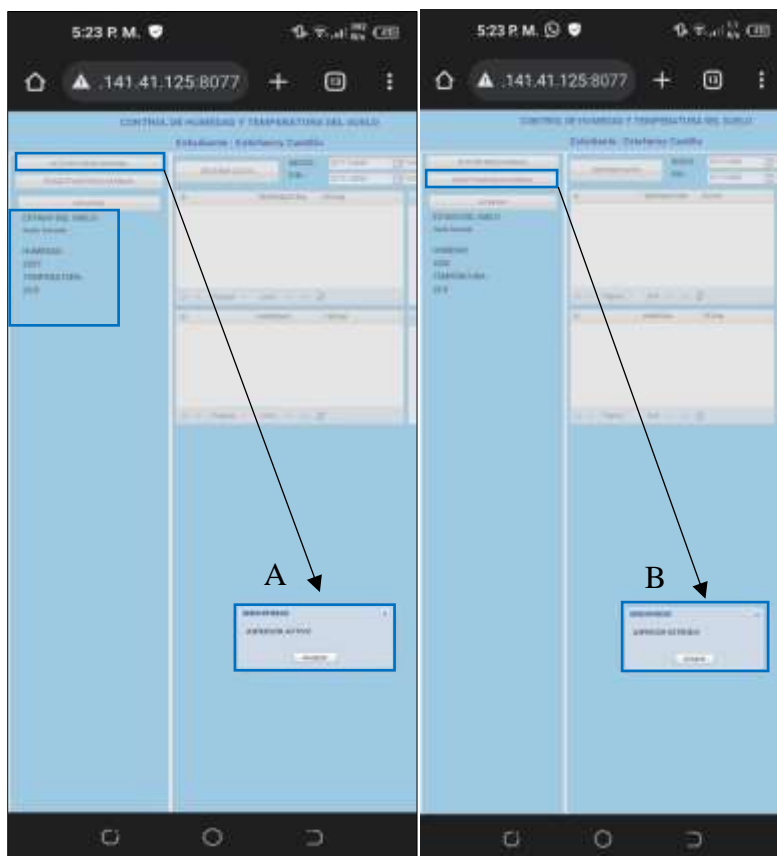


Figura 60 Activación y desactivación de riego manual desde la aplicación Web [Autor]

En la Figura 60, se puede contemplar a la aplicación web realizar la activación manual del riego mediante el botón ACTIVAR RIEGO MANUAL y a su vez dando clic en el botón aceptar y así poner en funcionamiento el aspersor esto se puede visualizar en la parte A de la imagen y la misma manera sucede en la sección B al desactivar el riego manual. También se monitorea el estado actual del suelo que está en estado húmedo y con los datos de los sensores de temperatura con un valor de 22.8°C y de humedad con un valor 2207, con referencia a las gráficas de la aplicación web no se visualiza ninguna curva en el comportamiento de los sensores dado a que no se escogió ninguna fecha ni hora para analizar los datos.

CAPÍTULO 4: CONCLUSIONES

Durante la investigación y el desarrollo de un sistema de riego basado en IoT en parcelas utilizadas para cultivar maíz híbrido, surgieron una serie de conclusiones sólidas que subrayan la relevancia significativa y el impacto potencial de esta tecnología en el sector agrícola. La motivación inicial de ofrecer a los agricultores una solución más asequible se tradujo en un

diseño integral que aborda los componentes de hardware y software de manera equilibrada. Sensores de humedad y temperatura, relés, placas LoRa WiFi y herramientas Delphi, MariaDB y Arduino cuidadosamente seleccionados dan como resultado un sistema eficiente y adaptable.

Las pruebas preliminares en el área de cultivo revelaron la necesidad de ajustar las escalas de los sensores para reflejar con precisión las condiciones del terreno, resaltando la importancia de un enfoque empírico en el diseño. La definición de condiciones operativas basadas en temperaturas óptimas y rangos de humedad proporcionó directrices cruciales para el desempeño eficaz del sistema de riego por aspersión. La elección estratégica de placas WiFi LoRa V3 subraya la versatilidad del sistema, y la arquitectura nodal, junto con la transmisión de datos a través de un servidor TCP y toma de decisiones mediante una aplicación web, consolida la eficiencia del proceso.

Además, se destacan un cuidadoso ajuste estacional en la aplicación del sistema. En invierno, la aplicación web detiene el riego manual debido a una humedad demasiado alta o temperaturas demasiado bajas, lo que ilustra una adaptación consciente al cambio climático. Asimismo, las medidas de precaución para detener el riego cuando las temperaturas superan los 30 grados Celsius pueden prevenir el estrés por calor, la evaporación excesiva y el crecimiento irregular de las plantas de maíz, lo que demuestra un enfoque integrado para promover la salud y el crecimiento de los cultivos.

En general, se consolida la eficacia del sistema, se destaca su viabilidad económica y se revela su potencial para transformar las prácticas agrícolas en desarrollo sostenible. La integración exitosa de las innovaciones tecnológicas y las necesidades agrícolas demuestra un enfoque prometedor y respetuoso con el medio ambiente del sector agrícola.

CAPÍTULO 5: RECOMENDACIONES

La integración de fuentes de energía renovables como paneles solares será una opción clave para mejorar la sostenibilidad del proyecto. Reducir la dependencia de las fuentes de energía tradicionales no solo mejora la eficiencia energética, sino que también respalda la integridad ambiental de los sistemas de riego de IoT.

Implementación de sensores adicionales como los de luz solar específicos para el cultivo de maíz. Esta estrategia brinda la oportunidad de enriquecer la información recopilada

y proporcionar datos adicionales sobre el medio ambiente, ayudando a tomar decisiones más precisas en función de las necesidades específicas del cultivo.

Finalmente, la colaboración activa con expertos agrícolas, especialmente agrónomos y agricultores especializados en el cultivo de maíz, se convierte en un elemento esencial para garantizar que el sistema sea consistente con las últimas prácticas agrícolas. Los valiosos comentarios de estos expertos no solo guían mejoras específicas para optimizar la eficiencia del riego, sino que también garantizan una adaptación exitosa a las necesidades reales de los agricultores.

TRABAJOS FUTUROS

Para trabajos futuros se propone implementar este riego en áreas de terreno más extensos y también para cultivos variados

REFERENCIAS BIBLIOGRÁFICAS

- [1] EL MAIZ EN LOS TRPICOS: Mejoramiento y produccion. (s/f). Fao.org. Recuperado el 30 de diciembre de 2022, de <https://www.fao.org/3/x7650s/x7650s02.htm>
- [2] El aumento de la producci n mundial de ma z ser impulsado principalmente por Am rica Latina. (27 de agosto de 2019). VirtualPro.co. <https://www.virtualpro.co/noticias/el-aumento-de-la-produccion-mundial-de-maiz-sera-impulsado-principalmente-por-america-latina->
- [3] Euronews. (2022, mayo 22). Faltan fertilizantes en América Latina como consecuencia de la guerra en Ucrania. Euronews. <https://es.euronews.com/2022/05/22/faltan-fertilizantes-en-america-latina-como-consecuencia-de-la-guerra-en-ucrania>
- [4] Zambrano, J. L., Yáñez, C., Sangoquiza, C., Limongi, R., Alarcón, D., Zambrano, E., Caicedo, M., Cartagena, Y., Parra, I., Azaquibay, I., Quimbiamba, I., Nieto, I., López, I. Tapia I., Tacán, I., Villacrés, I., Garcés, I., Cañarte, I., Navarrete, 3., ... Pinargote, 4. (s/f). Situación del cultivo de maíz en Ecuador: investigación y desarrollo de tecnologías en el Iniap. Gob.ec. Recuperado el 30 de diciembre de 2022, de <https://repositorio.iniap.gob.ec/bitstream/41000/5457/1/iniapeppdf62.pdf>

- [5] Cultivos de maíz en Ecuador. (s/f). Com.ec. Recuperado el 30 de diciembre de 2022, de <http://www.sancamillo.com.ec/maiz.html>
- [6] Cultivo de maíz .Recuperado el 30 de diciembre de 2022, de <http://file:///C:/Users/MARTHA~1/AppData/Local/Temp/MicrosoftEdgeDownloads/b60bc2ab-8989-49eb-80b3-ca9ce78c6326/GUIA%20CULTIVO%20DE%20MAIZ%202021-1.pdf>
- [7] Vijay, S. Anil Kumar, B. Susmita y N. Himanshu, «An IoT Instrumented Smart Agricultural Monitoring and Irrigation System, » IEEE, p. 4, 2020.
- [8] Tecnología LoRA y LoRAWAN . Catsensors.com. Recuperado el 22 de enero de 2023, de <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>
- [9] Bibliografía Vilcabamba. Gob.ec. Recuperado el 24 de marzo de 2023, de <https://www.loja.gob.ec/contenido/vilcabamba>
- [10] Jiménez, S., Castro, L., Yépez, J., & Wittmer, C. (s/f). Impacto del cambio climático en la agricultura de subsistencia en el Ecuador. Fundacioncarolina.es. Recuperado el 25 de enero de 2024, de <https://www.fundacioncarolina.es/wp-content/uploads/2014/08/AI66.pdf>
- [11] Brizuela, L. 1999. Guía Técnica para la Producción de Maíz en Honduras. Programa Nacional de Maíz. Omonita, Cortés. pp. 27-30.
- [12] Zea mays. (s/f). Ecured.cu. Recuperado el 26 de enero de 2024, de https://www.ecured.cu/Zea_mays
- [13] Parsons, D. 1998. Manual para la Educación Agropecuaria. Edición Trillas. pp. 36, 37, 38, 71, 72
- [14] CIMMYT, “Centro internacional de mejoramiento del maíz y trigo” 1994. Manejo de Ensayos e Informes de Datos de Ensayos Internacionales de Maíz. México D.F. CIMMYT. p. 23.

- [15] TOMi.digital - Ciencias Naturales - Sexto - Tipos de Suelo. TOMi.digital. Recuperado el 19 de septiembre de 2023, de https://tomi.digital/es/182445/ciencias-naturales-sexto-tipos-de-suelo?utm_source=google&utm_medium=seo
- [16] Junta de Comunidades de Castilla-La Mancha. Recuperado el 18 de enero de 2024, de https://www.castillalamancha.es/sites/default/files/documentos/pdf/20180829/informe_28_08_2018.pdf
- [17] Gestor, P. (2020, marzo 3). Claves del cultivo de maíz. SILOS del Cinca. <https://www.silosdelcinca.com/cereales/claves-del-cultivo-maiz/>
- [18] Suelos Salinos y Sódicos: Identificación, Mitigación y (2024). recuperado January 26, 2024, de attra.ncat.org
- [19] StackPath. Jacto.com. Recuperado el 20 de septiembre de 2023, de <https://bloglatam.jacto.com/tipos-de-riego/>
- [20] ¿Qué es el Internet de las cosas (IoT) y cómo funciona? (2024). recuperado January 26, 2024, de www.redhat.com/es/topics/internet-of-things/what-is-iot
- [21] LoRaWAN - AlfaIoT. (2022, noviembre 17). AlfaIoT - Creamos futuro: Desarrollamos, evolucionamos y optimizamos tu actividad con tecnología IoT. <https://alfaiot.com/tecnologiasiot/lora/>
- [22] A. A. William Townsley, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things, » vol. 16, nº 9, p. 1466, 2016.
- [23] O. Iriarte, «ALGUNOS APUNTES CONCEPTUALES SOBRE LOS MÉTODOS Y TIPOS DE RIEGO CAMPESINO Y SU RELACIÓN CON EL DISEÑO DE SISTEMAS DE RIEGO,» vol. 2, nº 1, pp. 2000-2020, 2005.
- [24] Pallás-Areny, R., & Webster, J. G. (2002). Sensores y Acondicionadores de Señal. Pearson.
- [25] Que son los sensores? Miperiodicodigital.com. Recuperado el 26 de enero de 2024, de <https://www.miperiodicodigital.com/2020/grupos/internationalope-11/que-son-sensores-872.html>

- [26] Sensor de Temperatura: Qué es y para qué sirve. (2022, diciembre 6). SDI. <https://sdiindustrial.com.mx/blog/sensor-de-temperatura-que-es/>
- [27] Sensor de temperatura. (2018, septiembre 20). Proyecto Arduino. <https://proyectoarduino.com/sensor-de-temperatura/>
- [28] Latam, M. (2020, febrero 21). Sensor de humedad. Mecatrónica LATAM. <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-humedad/>
- [29] Ventilación, S. (2018, noviembre 8). Sensor de humedad, tu aliado para combatir el vapor de agua en el aire. El blog de la ventilación inteligente. <https://www.siberzone.es/blog-sistemas-ventilacion/sensor-humedad/>
- [30] StackPath. (s/f). Jacto.com. Recuperado el 20 de septiembre de 2023, de <https://bloglatam.jacto.com/tecnologias-en-el-sector-agricola/>
- [31] Sergieieva, K. (2023, junio 14). Tecnología Agrícola: Innovación En El Cultivo De Cosechas. EOS Data Analytics. <https://eos.com/es/blog/tecnologias-en-la-agricultura/>
- [32] De una arquitectura Io T basada en tecnologías Open Source., D. e. I. (s/f). Trabajo Fin de Máster. Biblus.us.es. Recuperado el 20 de septiembre de 2023, de <https://biblus.us.es/bibing/proyectos/abreproy/70884/fichero/TFM-Antonio+Cobos+Dominguez.pdf>
- [33] Niam, J., Sokol, A., SurfUpVector, Butusava, A., S-S-S, Art, A., Srinusen, R., Scharfsinn, YasnaTen, Strelnikova, O., forest_strider, Designer, bsd studio, ONYXprj, Praditcharoenkul, J., Sikorka, G., palau, alphabetMN, FrankRamspott, ... FANDSRabutan. (s/f). 3.400+ Agricultura De Precisión Ilustraciones de Stock, gráficos vectoriales libres de derechos y clip art - iStock. Istockphoto.com. Recuperado el 14 de febrero de 2024, de <https://www.istockphoto.com/es/ilustraciones/agricultura-de-precisi%C3%B3n>
- [34] Ingeoexpert.com. Recuperado el 20 de septiembre de 2023, de <https://ingeoexpert.com/articulo/que-son-los-sistemas-de-informacion-geografica-sig-y-como-funcionan/>

- [35] Introducción a SIG.Arcgis.com. Recuperado el 20 de septiembre de 2023, de <https://resources.arcgis.com/es/help/getting-started/articles/026n0000000t000000.htm>
- [36] What is network Architecture? (2022, septiembre 20). VMware. <https://www.vmware.com/es/topics/glossary/content/network-architecture.html>
- [37] Conzultek. (s/f). ¿Qué es arquitectura de red y cuál es su importancia? Conzultek.com. Recuperado el 20 de septiembre de 2023, de <https://blog.conzultek.com/arquitectura-de-red>
- [38] Marketing. (2021, noviembre 12). Agricultura automatizada y robótica agrícola. EDS Robotics. <https://www.edsrobotics.com/blog/agricultura-automatizada-y-robotica-agricola/>
- [39] Pamoon, S. (2022, marzo 22). Descargar concepto de agricultores robóticos inteligentes, agricultores de robots, tecnología agrícola, automatización agrícola. gratis. Vecteezy. <https://es.vecteezy.com/foto/6686196-robot-inteligente-agricultores-concepto-robot-agricultores-agricultura-tecnologia-granja-automatizacion>
- [40] La agricultura tiene a la tecnologia blockchain para potenciar el sector. (2023, junio 26). Deloitte Ecuador. <https://www2.deloitte.com/ec/es/pages/Audit/articles/la-agricultura-tiene-a-la-tecnologia-blockchain-para-potenciar-al-sector.html>
- [41] Aplicaciones de Blockchain en la industria alimentaria. Trazabilidad, calidad y gestión de productos. (2021, enero 14). Inesdi. <https://www.inesdi.com/blog/aplicaciones-de-Blockchain-en-la-industria-alimentaria/>
- [42] Disner, E., & MyFarm. (2023, enero 1). Cómo puede contribuir el sistema de gestión agrícola a aumentar la productividad. MyFarm - Software de Gestão Agrícola - Uma Solução Aliare. <https://www.myfarm.com.br/sistema-de-gestion-agricola/>
- [43] Valle Hernández, L. (2017, septiembre 12). DS18B20 sensor de temperatura para líquidos con Arduino. Programar fácil con Arduino. <https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>
- [44] Vistronica.com. Recuperado el 12 de septiembre de 2023, de <https://www.vistronica.com/sensores/humedad/modulo-sensor-de-humedad-del-suelo->

hddetail.html#:~:text=Modulo%20sensor%20de%20humedad%20del%20suelo%20HD%2D38%20resistente%20a,adecuadamente%20resistentes%20a%20la%20oxidaci%C3%B3n.

- [45] (Comunicación de larga distancia de dispositivos IoT: Heltec LoRa 32» El Laboratorio del Dr. Yuri. Eniac2000.com. Recuperado el 9 de octubre de 2023, de <https://bitacora.eniac2000.com/?p=4364>
- [46] Hardware Update Logs — esp32 latest documentation. (s/f). Heltec.Cn. Recuperado el 18 de enero de 2024, de https://docs.heltec.cn/en/node/esp32/hardware_update_log.html
- [47] WiFi LoRa 32(V3). (s/f). Heltec Automation. Recuperado el 18 de enero de 2024, de <https://heltec.org/project/wifi-lora-32-v3/>
- [48] MakerFocus ESP32 LoRa Development Board WIFI Bluetooth with 0.96inch OLED Display and 2dBi Antenna. (s/f). MakerFocus. Recuperado el 10 de octubre de 2023, de <https://www.makefocus.com/collections/esp32/products/makerfocus-esp32-lora-development-board-wifi-bluetooth-with-0-96inch-oled-display-and-2dbi-antenna-1>
- [49] MÓDULO RELE DE 1 CANAL 5V 1CH. SSDIELECT ELECTRONICA SAS. Recuperado el 10 de octubre de 2023, de <https://ssdielect.com/rele-relay-o-relevo/992-md-rele-1ch.html>
- [50] Módulo relé relay de 5V, 1 canal Arduino con optoacoplador. PLEXYLAB. Recuperado el 10 de octubre de 2023, de <https://www.plexylab.com/product-page/m%C3%B3dulo-rel%C3%A9-relay-de-5v-1-canal-arduino-con-optoacoplador>
- [51] Amazon.com: Beduan Válvula electromagnética eléctrica de latón, 1/2" válvula de aire 12V normalmente colsed para el aceite combustible del aire del agua: Industrial y Científico. (n.d.). Wwww.amazon.com. Retrieved October 10, 2023, from https://www.amazon.com/-/es/solenoide-el%C3%A9ctrica-pulgadas-normalmente-combustible/dp/B07N6246YB/ref=sr_1_2?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=1E2VRUR9J0OY6&keywords=valvula%2Bsenosoidal&qid=1684974704&srefix=valvula%2Bsenosoidal%2Caps%2C194&sr=8-2&th=1

- [52] Grupo Ferretero CHC. Cinta de aislar 19mm x 18M truper expert CHC. Available at: <https://www.grupoferreterochc.com.mx/products/detail/55870> (Accessed: 21 December 2023).
- [53] Monitoring-Delta. (2002, julio 25). FUENTE CONMUTADA 12V/10A/P. Shopdelta.eu. https://shopdelta.eu/fuente-conmutada-12v10ap_l6_p3961.html
- [54] Manguera Polietileno Negra CED-40 de 1/2" 100 M Diametro 13 MM – RACSA. Retrieved October 11, 2023, from <https://www.racsariego.com/product/manguera-polietileno-negra-ced-40-de-1-2-100-m-diametro-13-mm/?v=0b98720dcb2c#:~:text=Producido%20a%20partir%20de%20polietileno>
- [55] Traxco. (2023, 12 julio). Aspensor XCel Wobbler Senninger. El Blog de Traxco. <https://www.traxco.es/blog/productos-nuevos/xcel-wobbler>
- [56] ASPERSOR PLASTICO EXCEL WOBBLER - Grupo Hidraulica. (2023, 10 julio). Grupo Hidraulica. <https://grupohidraulica.com/producto/aspersor-plastico-excel-wobbler/>
- [57] Caja de derivación para conexiones, Plexo IP55, cuadrada. Dimensiones: 105x105x55mm. 7 conos. Color gris. (s/f). Legrand.es. Recuperado el 26 de octubre de 2023, de <https://www.legrand.es/productos/caja-derivacion-para-conexiones-plexo-ip55-cuadrada-dimensiones-105x105x55mm-7-conos-color-gris-09202>
- [58] Caja de derivación para conexiones, Plexo IP55, cuadrada. Dimensiones: 80x80x45mm. 7 conos. Color gris. (s/f). Legrand.es. Recuperado el 26 de octubre de 2023, de <https://www.legrand.es/productos/caja-derivacion-para-conexiones-plexo-ip55-cuadrada-dimensiones-80x80x45mm-7-conos-color-gris-092012>
- [59] CABLES DE COBRE AISLADO PARALELO TIPO SPT 300V. Incable.com. Recuperado el 18 de enero de 2024, de <https://incable.com/producto?id=46>
- [60] Tanenbaum, A. S., & Van Steen, M. (2016). Distributed Systems: Principles and Paradigms (3rd ed.). Pearson.
- [61] Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). Distributed Systems: Concepts and Design (5th ed.). Pearson.

- [62] Comer, D. E. (2015). Internetworking with TCP/IP: Principles, Protocols, and Architecture (Vol. III). Prentice Hall.
- [63] Stevens, W. R., Fenner, B., & Rudoff, A. M. (2004). TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley.
- [64] Amazon. ¿Qué es Amazon EC2? - Amazon Elastic Compute Cloud. Recuperado el 18 de enero de 2024, de https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html
- [65] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (Tesis de doctorado). Universidad de California, Irvine. Recuperado el 18 de enero de 2024, de <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [66] Sencha. (2015, abril 21). Sencha Ext JS - comprehensive JavaScript framework, UI components. Recuperado el 18 de enero de 2024, de <https://www.sencha.com/products/extjs/>
- [67] Geater, J. ¿Qué Es Delphi (IDE)? (de Embarcadero Technologies). Solvusoft.com. Recuperado el 18 de enero de 2024, de <https://www.solvusoft.com/es/file-extensions/software/embarcadero-technologies/delphi-ide/>
- [68] Software a medida para tu negocio - GSoft Innova. (2018, julio 30). GSoft. <https://www.gsoft.es/software-a-medida/>
- [69] Fernández, Y. (2022, septiembre 23). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

APÉNDICES

A. PROGRAMACIÓN DE NÚMERO DE NODOS Y SENSORES EN MATLAB

```
B. % Definir la frecuencia de operación de LoRa
C. frequency = 865e6; % Hz
D.
E. % Definir la potencia de transmisión
F. tx_power = 5; % dBm
G.
H. % Definir la sensibilidad del receptor
I. rx_sensitivity = -136; % dBm
J.
K. % Definir la distancia mínima deseada entre nodos
L. min_distance = 2; % m
M.
N. % Calcular la ganancia de la antena
O. antenna_gain = 2.5; % dBi
P.
Q. % Calcular la pérdida de transmisión
R. tx_loss = 25 * log10(min_distance) + 32.44; % dB
S.
T. % Calcular el margen de ruido
U. noise_margin = tx_power - rx_sensitivity + antenna_gain - tx_loss;
V.
W. % Imprimir el margen de ruido
X. fprintf('Margen de ruido: %.2f dB\n', noise_margin)
Y.
Z. % Definir el nivel de ruido
AA.noise_level = -0.3 + 10 * log10(125000); % dBm
BB.
CC.% Calcular el número exacto de nodos necesarios para la comunicación
    bidireccional
DD.nodos_transmisor_receptor = 2;
EE.
FF.% Imprimir el número exacto de nodos
GG. fprintf('Número de nodos totales para tramision y recepcion: %d\n',
    nodos_transmisor_receptor);
HH.
II.% Definir la cantidad de sensores de humedad y temperatura necesarios por
    nodo
JJ.sensores_humedad_por_nodo_transmisor = 1;
KK.sensores_temperatura_por_nodo_transmisor = 1;
LL.
MM.sensores_humedad_por_nodo_receptor = 0;
NN.sensores_temperatura_por_nodo_receptor = 0;
OO.
PP.% Calcular el total de sensores requeridos para todos los nodos
QQ.total_sensores_humedad_transmisor = nodos_transmisor_receptor *
    sensores_humedad_por_nodo_transmisor;
RR.total_sensores_temperatura_transmisor = nodos_transmisor_receptor *
    sensores_temperatura_por_nodo_transmisor;
SS.
```

```

TT. total_sensores_humedad_receptor = nodos_transmisor_receptor *
   sensores_humedad_por_nodo_receptor;
UU. total_sensores_temperatura_receptor = nodos_transmisor_receptor *
   sensores_temperatura_por_nodo_receptor;
VV.
WW. % Imprimir la cantidad de sensores requeridos por separado
XX. fprintf('Sensores de humedad en nodo transmisor: %d\n',
   total_sensores_humedad_transmisor);
YY. fprintf('Sensores de temperatura en nodo transmisor: %d\n',
   total_sensores_temperatura_transmisor);
ZZ.
AAA.   fprintf('Sensores de humedad en nodo receptor: %d\n',
   total_sensores_humedad_receptor);
BBB.   fprintf('Sensores de temperatura en nodo receptor: %d\n',
   total_sensores_temperatura_receptor);
CCC.

```

B. PROGRAMACIÓN DEL NODO TRANSMISOR EN ARDUINO IDE 2.2.1

```

//TRANSMISOR FINAL TESIS //10/12/2023
//Autor : Martha Estefanny Astudillo Castillo
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "WiFi.h"
#include <Wire.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include "HT_SSD1306Wire.h" // Libreria para la pantalla oled del fabricante
de la placa
#include <OneWire.h>
#include <DallasTemperature.h>

#define RF_FREQUENCY           865000000 // MHz
#define TX_OUTPUT_POWER      5          // dBm
#define LORA_BANDWIDTH       0          // [0: 125 kHz,
// 1: 250 kHz,
// 2: 500 kHz,
// 3: Reserved]
#define LORA_SPREADING_FACTOR 7          // [SF7..SF12]
#define LORA_CODINGRATE      1          // [1: 4/5,
// 2: 4/6,
// 3: 4/7,
// 4: 4/8]
#define LORA_PREAMBLE_LENGTH 8          // Same for Tx
and Rx
#define LORA_SYMBOL_TIMEOUT  0          // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false

```

```

#define RX_TIMEOUT_VALUE          1000
#define BUFFER_SIZE               30 // Define the payload
size here

char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];

static RadioEvents_t RadioEvents;
void OnTxDone( void );
void OnTxTimeout( void );
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr );

typedef enum
{
    LOWPOWER,
    STATE_RX,
    STATE_TX
} States_t;

int16_t txNumber;
States_t state;
bool sleepMode = false;
int16_t Rssi, rxSize;

/*****
OLED *****/
SSD1306Wire factory_display(0x3c, 500000, SDA_OLED, SCL_OLED,
GEOMETRY_128_64, RST_OLED); // addr , freq , i2c group , resolution , rst

// -----          PARAMETROS RED WIFI          -----
-----
const char * WIFISSID = "FAMILIA ABENDAÑO"; // Nombre de la red wifi
const char * PASSWORD = "16/enero/1950"; // Contraseña del wifi
// -----

// Declaramos el pin de conexión del sensor temperatura
// Pin para el sensor de temperatura

OneWire ourWire1(7);           //Se establece el pin 7 como bus OneWire
OneWire ourWire2(6);           //Se establece el pin 6 como bus OneWire

DallasTemperature sensors1(&ourWire1); //Se declara una variable u objeto para
nuestro sensor1
DallasTemperature sensors2(&ourWire2); //Se declara una variable u objeto para
nuestro sensor2

// Setup a oneWire instance to communicate with any OneWire devices

```

```

// (not just Maxim/Dallas temperature ICs)
//OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
//DallasTemperature sensors(&oneWire);
double temperatura = 0 ;
double temperatura1 = 0 ;
double temperatura2 = 0 ;

// pin de conexión para el sensor de humedad
int sensor_humedad1 = 5;
int sensor_humedad2 = 4;
// variable para almacenar el valor del sensor de humedad
int valor_humedad = 0;
int valor_humedad1 = 0;
int valor_humedad2 = 0;

int RELAY_PIN = 2;
int estado_rele = 0;

int pulsador = 3;
int estado_pulsador = 0;

bool control = true;

////////////////////// INTERFAZ WEB ////////////////////////
WiFiServer server(80);
String header;

void setup() {
  Serial.begin(9600);
  factory_display.init();
  factory_display.clear();
  factory_display.setFont(ArialMT_Plain_16);
  factory_display.display();

  WiFi.mode(WIFI_STA);
  WiFi.setAutoConnect(true);
  // conectamos el equipo a red wifi
  WiFi.begin(WIFISSID, PASSWORD);
  delay(100);
  byte count = 0;
  while (WiFi.status() != WL_CONNECTED && count < 10)
  {
    count ++;
    delay(500);
    factory_display.drawString(0, 0, "Conectando...");
  }
}

```

```

    factory_display.display();
}

factory_display.clear();
if (WiFi.status() == WL_CONNECTED)
{
    factory_display.drawString(0, 0, "Conectado...OK.");
    factory_display.display();
    //    delay(500);
}
else
{
    factory_display.clear();
    factory_display.drawString(0, 0, "Error al conectar wifi");
    factory_display.display();
    //while(1);
}
Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("dirección IP: ");
Serial.println(WiFi.localIP());
// iniciamos el sensor
sensors1.begin(); //Se inicia el sensor 1
sensors2.begin(); //Se inicia el sensor 2

server.begin();
Serial.println("Servidor iniciado");

pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, LOW);
pinMode(pulsador, INPUT);

Mcu.begin();
txNumber = 0;
Rssi = 0;

RadioEvents.TxDone = OnTxDone;
RadioEvents.TxTimeout = OnTxTimeout;
RadioEvents.RxDone = OnRxDone;

Radio.Init( &RadioEvents );
Radio.SetChannel( RF_FREQUENCY );
Radio.SetTxConfig( MODEM_LORA, TX_OUTPUT_POWER, 0, LORA_BANDWIDTH,
                  LORA_SPREADING_FACTOR, LORA_CODINGRATE,
                  LORA_PREAMBLE_LENGTH, LORA_FIX_LENGTH_PAYLOAD_ON,
                  true, 0, 0, LORA_IQ_INVERSION_ON, 3000 );

```

```

Radio.SetRxConfig( MODEM_LORA, LORA_BANDWIDTH, LORA_SPREADING_FACTOR,
                   LORA_CODINGRATE, 0, LORA_PREAMBLE_LENGTH,
                   LORA_SYMBOL_TIMEOUT, LORA_FIX_LENGTH_PAYLOAD_ON,
                   0, true, 0, 0, LORA_IQ_INVERSION_ON, true );
state = STATE_TX;
}
void loop()
{
  WiFiClient client = server.available();
  sensors1.requestTemperatures(); // //Se envía el comando para leer la
temperatura
  temperatura1 = (sensors1.getTempCByIndex(0)); //Se obtiene la temperatura en
°C del sensor 1

  sensors2.requestTemperatures(); //Se envía el comando para leer la
temperatura
  temperatura2 = (sensors2.getTempCByIndex(0)); //Se obtiene la temperatura
en °C del sensor 2

  Serial.print("Temperatura 1 = ");
  Serial.print(temperatura1);
  Serial.print(" C");
  Serial.print("  Temperatura 2 = ");
  Serial.print(temperatura2);
  Serial.println(" C");

  temperatura = (temperatura1 + temperatura2) / 2;

  //presentacion los parametros por el puerto serial
  char tempstring[20];
  dtostrf(temperatura, 3, 1, tempstring);
  String temp(tempstring);

  valor_humedad1 = analogRead(sensor_humedad1);
  valor_humedad2 = analogRead(sensor_humedad2);
  valor_humedad = (valor_humedad1 + valor_humedad2) / 2;

  char humstring[20];
  dtostrf(valor_humedad, 3, 1, humstring);
  String hum(humstring);

  factory_display.drawString(0, 0, "          ");
  factory_display.drawString(0, 20, "          ");
  factory_display.display();

  factory_display.drawString(0, 0, "Temp: ");
  factory_display.drawString(50, 0, temp);

```



```

factory_display.drawString(0, 20, "Hum: ");
factory_display.drawString(50, 20, hum);
factory_display.display();

Serial.print("Temperatura es: ");
Serial.println(temperatura);
Serial.print("Humedad es: ");
Serial.println(valor_humedad);

// si control es 0 el rele se controla con los sensores
if (control == true) {
  // Control del relé basado en la temperatura y la humedad
  if (temperatura >= 20 && temperatura <= 30) {
    if (valor_humedad >= 1200 && valor_humedad <= 1800) {
      Serial.println("Exceso de Humedad");
      digitalWrite(RELAY_PIN, LOW);
      factory_display.drawString(0, 40, "                ");
      factory_display.display();
      factory_display.drawString(0, 40, "Exceso de Humedad");
      factory_display.display();
      estado_rele = 0;
      delay(5000);
    } else if (valor_humedad > 1801 && valor_humedad <= 3020) {
      Serial.println("Suelo Humedo");
      digitalWrite(RELAY_PIN, LOW);
      factory_display.drawString(0, 40, "                ");
      factory_display.display();
      factory_display.drawString(0, 40, "Suelo Humedo");
      factory_display.display();
      estado_rele = 0;
      delay(5000);
    } else if (valor_humedad > 3021 && valor_humedad <= 4000) {
      Serial.println("Suelo Seco");
      digitalWrite(RELAY_PIN, HIGH);
      factory_display.drawString(0, 40, "                ");
      factory_display.display();
      factory_display.drawString(0, 40, "Suelo Seco");
      factory_display.display();
      estado_rele = 1;
      delay(5000);
    } else if (valor_humedad > 4002) {
      Serial.println("Riego Necesario");
      digitalWrite(RELAY_PIN, HIGH);
      factory_display.drawString(0, 40, "                ");

```

```

        factory_display.display();
        factory_display.drawString(0, 40, "Riego Necesario");
        factory_display.display();
        estado_rele = 1;
        delay(5000);

    } else {
        digitalWrite(RELAY_PIN, LOW);
        estado_rele = 0;

    }

} else {
    Serial.println("Riego Bloqueado");
    digitalWrite(RELAY_PIN, LOW);
    estado_rele = 0;

}

// Control del relé basado en la temperatura
if (temperatura <= 20) {
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.println(" °C");
    Serial.println("Riego Bloqueado por temperatura baja");
    digitalWrite(RELAY_PIN, LOW);
    delay(5000);

} else if (temperatura >= 30) {
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.println(" °C");
    Serial.println("Riego Bloqueado por temperatura alta");
    digitalWrite(RELAY_PIN, LOW);
    delay(5000);
}

}

if (control == false) {
    digitalWrite(RELAY_PIN, HIGH);
    Serial.println("Riego activado manualmente");
    factory_display.drawString(0, 40, "                ");
    factory_display.display();
    factory_display.drawString(0, 40, "Control Manual");
    factory_display.display();

}

String msgT = "";

```

```

msgT = String(temperatura) + "," + String(valor_humedad) + "," +
String(estado_rele);
Serial.println(msgT);
sprintf(txpacket, "Datos: %s", msgT); //
Serial.printf("\r\nenviando datos \"%s\" , length %d\r\n", txpacket,
strlen(txpacket));
Radio.Send((uint8_t *)txpacket, strlen(txpacket) ); //envia los datos
Radio.IrqProcess( );

if (client)
{
Serial.println("Web Client connected ");
String request = client.readStringUntil('\r');
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close"); // the connection will be closed
after completion of the response
client.println("Refresh: 10"); // update the page after 10 sec
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<style>html { font-family: Cairo; display: block; margin:
0px auto; text-align: center;color: #333333; background-color: ##f3ffee;}");
client.println("body{margin-top: 50px;}");
client.println("h1 {margin: 50px auto 30px; font-size: 50px; text-align:
center;}");
client.println(".side_adjust{display: inline-block;vertical-align:
middle;position: relative;}");
client.println(".text1{font-weight: 180; padding-left: 15px; font-size:
50px; width: 170px; text-align: left; color: #3498db;}"); // color azul
client.println(".data1{font-weight: 180; padding-left: 70px; font-size:
50px;color: #3498db;}");
client.println(".text2{font-weight: 180;padding-left: 15px; font-size:
50px; width: 170px; text-align: left; color: #ff6600;}"); // color naranja
client.println(".data2{font-weight: 180; padding-left: 120px; font-size:
50px;color: #ff6600;}");
client.println(".text3{font-weight: 180;padding-left: 15px; font-size:
50px; width: 170px; text-align: left; color: #ff3333;}");
client.println(".data3{font-weight: 180; padding-left: 120px; font-size:
50px;color: #ff3333;}"); // color rojo
client.println(".data{padding: 10px;}");
client.println("</style>");
client.println("</head>");
client.println("<body>");
client.println("<div id=\"webpage\">");
client.println("<h1>Estado de los sensores</h1>");
client.println("<div class=\"data\">");

```

```

client.println("<div class=\"side_adjust text1\">Humedad:</div>");
client.println("<div class=\"side_adjust data1\">");
client.print(valor_humedad);
client.println("<div class=\"side_adjust text1\">%</div>");
client.println("</div>");

client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text2\">Temperatura:</div>");
client.println("<div class=\"side_adjust data2\">");
client.print(temperatura);
client.println("<div class=\"side_adjust text2\">*C</div>");
client.println("</div>");

client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text3\">Relay:</div>");
client.println("<div class=\"side_adjust data3\">");
client.print(estado_rele);
client.println("<div class=\"side_adjust text3\"></div>");
client.println("</div>");
client.println("</div>");
client.println("</body>");
client.println("</html>");
delay(4000);
}
Serial.println();
Serial.println("closing connection");
client.stop();
//delay(5000);
factory_display.clear();

// Configuración de WiFi y servidor
if (client) {
  if (client.connected()) {
    client.stop();
  }
}
if (client.available() > 0) {
  String request = client.readStringUntil('\r');
  Serial.println(request);
  if (request == "a") {
    // Si se recibe 'a' desde el receptor, activa el relé
    digitalWrite(RELAY_PIN, HIGH);
    Serial.println("Riego activado manualmente");
  } else if (request == "i") {
    // Si se recibe 'i' desde el receptor, desactiva el relé
    digitalWrite(RELAY_PIN, LOW);
  }
}

```

```

        Serial.println("Riego desactivado manualmente");
    }
    client.stop(); // Cierra la conexión con el receptor
}
delay(1000);
Serial.println("into RX mode");
Radio.Rx(0);
Radio.IrqProcess( );
factory_display.clear();
delay(1000);
}

void OnTxDone( void )
{
    Serial.print("TX done.....");
    state = STATE_RX;
}
void OnTxTimeout( void )
{
    Radio.Sleep( );
    Serial.print("TX Timeout.....");
    state = STATE_TX;
}
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr )
{
    Rssi = rssi;
    rxSize = size;
    memcpy(rxpacket, payload, size );
    rxpacket[size] = '\0';
    Radio.Sleep( );

    Serial.printf("\r\npaquete recibido \"%s\" with Rssi %d , length %d\r\n",
rxpacket, Rssi, rxSize);

    String myString = String(rxpacket);
    Serial.print("Recibido: ");
    Serial.println(myString);
    control = !control;
    state = STATE_TX;
}

```

C. PROGRAMACION DEL NODO RECEPTOR EN ARDUINO IDE 2.2.1

```

// RECEPTOR FINAL TESIS // 10/12/2023
//Autor : Martha Estefanny Astudillo Castillo
#include "LoRaWan_APP.h"
#include "Arduino.h"

```

```

#include "WiFi.h"
#include <Wire.h>
#include <WiFiServer.h>
#include "HT_SSD1306Wire.h" // Libreria para la pantalla oled del fabricante
de la placa

#include <HTTPClient.h>

// Configuración del Servidor
const char* serverUrl = "3.141.41.125"; // URL del servidor
const char* id1 = "temp";// ID
const char* id2 = "hume";// ID

// Configuración TCP
#define SERVER_PORT 20000 // Puerto para la comunicación TCP
#define RELAY_ON_COMMAND "ActivarRiego"
#define RELAY_OFF_COMMAND "DesactivarRiego"

#define RF_FREQUENCY 865000000 // Hz
#define TX_OUTPUT_POWER 5 // dBm
#define LORA_BANDWIDTH 0 // [0: 125 kHz,
// 1: 250 kHz,
// 2: 500 kHz,
// 3: Reserved]
#define LORA_SPREADING_FACTOR 7 // [SF7..SF12]
#define LORA_CODINGRATE 1 // [1: 4/5,
// 2: 4/6,
// 3: 4/7,
// 4: 4/8]
#define LORA_PREAMBLE_LENGTH 8 // Same for Tx
and Rx
#define LORA_SYMBOL_TIMEOUT 0 // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false

#define RX_TIMEOUT_VALUE 1000
#define BUFFER_SIZE 30 // Define the payload
size here

char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];

static RadioEvents_t RadioEvents;
void OnTxDone( void );
void OnTxTimeout( void );
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr );

```

```

typedef enum
{
    LOWPOWER,
    STATE_RX,
    STATE_TX
} States_t;

int16_t txNumber;
States_t state;
bool sleepMode = false;
int16_t Rssi, rxSize;

/*****
OLED *****/
SSD1306Wire factory_display(0x3c, 500000, SDA_OLED, SCL_OLED,
GEOMETRY_128_64, RST_OLED); // addr , freq , i2c group , resolution , rst

// -----          PARAMETROS RED WIFI          -----
-----
const char * WIFISSID = "FAMILIA ABENDAÑO"; // Nombre de la red wifi
const char * PASSWORD = "16/enero/1950"; // Contraseña del wifi
// -----
float temperatura = 0 ;
// variable para almacenar el valor del sensor de humedad
int valor_humedad = 0;
int estado_rele = 0;

int PosicionComa; //Variable que nos dice la posicion de la coma en el
mensaje.
String temp;
String hume;
int control = 0;
WiFiServer server(SERVER_PORT);
WiFiClient client;

unsigned long previousMillis = 0;
unsigned long currentMillis;
unsigned long intervalo = 180000;

void setup() {
    // iniciamos conexion serial
    Serial.begin(9600);
    factory_display.init();
    factory_display.clear();
    factory_display.setFont(ArialMT_Plain_16);

```

```

factory_display.display();

WiFi.mode(WIFI_STA);
WiFi.setAutoConnect(true);
// conectamos el equipo a red wifi
WiFi.begin(WIFISSID, PASSWORD);
delay(100);
byte count = 0;
while (WiFi.status() != WL_CONNECTED && count < 10)
{
    count ++;
    delay(500);
    factory_display.drawString(0, 0, "Conectando...");
    factory_display.display();
}

factory_display.clear();
if (WiFi.status() == WL_CONNECTED)
{
    factory_display.drawString(0, 0, "Conectado...OK.");
    factory_display.display();
    // delay(500);
}
else
{
    factory_display.clear();
    factory_display.drawString(0, 0, "Error al conectar wifi");
    factory_display.display();
    //while(1);
}

Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("dirección IP: ");
Serial.println(WiFi.localIP());

Mcu.begin();
txNumber = 0;
Rssi = 0;

RadioEvents.TxDone = OnTxDone;
RadioEvents.TxTimeout = OnTxTimeout;
RadioEvents.RxDone = OnRxDone;

Radio.Init( &RadioEvents );
Radio.SetChannel( RF_FREQUENCY );
Radio.SetTxConfig( MODEM_LORA, TX_OUTPUT_POWER, 0, LORA_BANDWIDTH,

```



```

        LORA_SPREADING_FACTOR, LORA_CODINGRATE,
        LORA_PREAMBLE_LENGTH, LORA_FIX_LENGTH_PAYLOAD_ON,
        true, 0, 0, LORA_IQ_INVERSION_ON, 3000 );

Radio.SetRxConfig( MODEM_LORA, LORA_BANDWIDTH, LORA_SPREADING_FACTOR,
        LORA_CODINGRATE, 0, LORA_PREAMBLE_LENGTH,
        LORA_SYMBOL_TIMEOUT, LORA_FIX_LENGTH_PAYLOAD_ON,
        0, true, 0, 0, LORA_IQ_INVERSION_ON, true );
state = STATE_TX;
factory_display.clear();
}

void loop()
{
    WiFiClient client = server.available();
    Radio.Rx(0);
    Radio.IrqProcess( );
    factory_display.clear();
    delay(1000);
}

void OnTxDone( void )
{
    Serial.print("TX done.....");
    state = STATE_RX;
}

void OnTxTimeout( void )
{
    Radio.Sleep( );
    Serial.print("TX Timeout.....");
    state = STATE_TX;
}

void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr )
{
    rssi = rssi;
    rxSize = size;
    memcpy(rxpacket, payload, size );
    rxpacket[size] = '\0';
    Radio.Sleep( );
    Serial.printf("\r\npaquete recibido \"%s\" with rssi %d , length %d\r\n",
rxpacket, rssi, rxSize);

    String myString = String(rxpacket);

```

```

String Segmento1 = myString .substring(0, 6);
String Segmento2 = myString .substring(7, 11);
String Segmento3 = myString .substring(13, 17);
String Segmento4 = myString .substring(19, 20);
String Segmento5 = myString .substring(22, 23);

Serial.print("Segmento1: ");
Serial.println(Segmento1);
Serial.print("Segmento2: ");
Serial.println(Segmento2);
Serial.print("Segmento3: ");
Serial.println(Segmento3);
Serial.print("Segmento4: ");
Serial.println(Segmento4);

temperatura = Segmento2.toFloat();
valor_humedad = Segmento3.toInt();
control = Segmento5.toInt();

char tempstring[20];
dtostrf(temperatura, 3, 1, tempstring);
String temp(tempstring);

char humstring[20];
dtostrf(valor_humedad, 3, 1, humstring);
String hum(humstring);

factory_display.drawString(0, 0, "          ");
factory_display.drawString(0, 20, "          ");
factory_display.display();

factory_display.drawString(0, 0, "Temp: ");
factory_display.drawString(50, 0, Segmento2);
factory_display.drawString(0, 20, "Hum: ");
factory_display.drawString(50, 20, Segmento3);
factory_display.display();

Serial.print("Temperatura es: ");
Serial.println(temperatura);
Serial.print("Humedad es: ");
Serial.println(valor_humedad);

if (control == 0) {
  // Control del relé basado en la temperatura y la humedad
  if (temperatura >= 20 && temperatura <= 30) {
    if (valor_humedad >= 1200 && valor_humedad <= 1800) {
      Serial.println("Exceso de Humedad");
    }
  }
}

```

```

        factory_display.drawString(0, 40, "                ");
        factory_display.display();
        factory_display.drawString(0, 40, "Exceso de Humedad");
        factory_display.display();
    } else if (valor_humedad > 1081 && valor_humedad <= 3020) {
        Serial.println("Suelo Humedo");
        factory_display.drawString(0, 40, "                ");
        factory_display.display();
        factory_display.drawString(0, 40, "Suelo Humedo");
        factory_display.display();
    } else if (valor_humedad > 3021 && valor_humedad <= 4000) {
        Serial.println("Suelo Seco");
        factory_display.drawString(0, 40, "                ");
        factory_display.display();
        factory_display.drawString(0, 40, "Suelo Seco");
        factory_display.display();
    } else if (valor_humedad > 4002) {
        Serial.println("Riego Necesario");
        factory_display.drawString(0, 40, "                ");
        factory_display.display();
        factory_display.drawString(0, 40, "Riego Necesario");
        factory_display.display();
    }
} else {
    Serial.println("Riego Bloqueado");
}
// Control del relé basado en la temperatura
if (temperatura <= 20) {
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.println(" °C");
    Serial.println("Riego Bloqueado por temperatura baja");
    factory_display.drawString(0, 40, "                ");
    factory_display.display();
    factory_display.drawString(0, 40, "Riego Bloq temp baja");
    factory_display.display();
    // baja temperatura
} else if (temperatura >= 30) {
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.println(" °C");
    Serial.println("Riego Bloqueado por temperatura alta");
    factory_display.drawString(0, 40, "                ");
    factory_display.display();
    factory_display.drawString(0, 40, "Riego Bloq temp alta");
    factory_display.display();
}

```

```

    }
}
if (control == 1) {
    factory_display.drawString(0, 40, "          ");
    factory_display.display();
    factory_display.drawString(0, 40, "Control Manual");
    factory_display.display();
}

// Envía los datos al servidor
const uint16_t port = 20000;
const char * host = "3.141.41.125"; // ip or dns

Serial.print("Conectando a la IP: ");
Serial.println(host);
client.connected();
// Verifica si la conexión al servidor está activa
if (!client.connected()) {
    Serial.println("Conexión perdida. Reconectando al servidor...");
    if (client.connect(host, port)) {
        Serial.println("Conectado al servidor.");
        Serial.println("Fallo en la reconexión. Intentando nuevamente ...");
        //delay(5000);
        return;
    }
}
//*****
currentMillis = millis();
if ((currentMillis - previousMillis) >= intervalo) {
    previousMillis = currentMillis;
    // Envía los datos al servidor
    String formattedHttp = String(temperatura, 1) + ":" + hum ; // 1 decimal
en la temperatura
    client.print(formattedHttp);
}

if (client.available() > 0) {
    // Lee una línea de respuesta del servidor si es necesario
    String line = client.readStringUntil('\r');
    Serial.println("line: ");
    Serial.println(line);
    String msgT = "";
    msgT = String(line);
    Serial.println(msgT);
    sprintf(txpacket, "C: %s", msgT); //start a package
}

```

```

        Serial.printf("\r\nsending packet \"%s\" , length %d\r\n", txpacket,
strlen(txpacket));
        Radio.Send( (uint8_t *)txpacket, strlen(txpacket) ); //send the package
out
    } else {
        Serial.println("client.available() timed out");
    }
}
}

```

D. PROGRAMACION DE LA APLICACIÓN WEB EN DELPHI

PROGRAMACIÓN DE UNIT DA_BASE SERVIDOR TCP

unit Da_base;

interface

uses

System.SysUtils, System.Classes, Data.DB, DBAccess, Uni, UniProvider,

MySQLUniProvider, MemDS, FireDAC.Stan.Intf, FireDAC.Stan.Option,

FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS, FireDAC.Phys.Intf,

FireDAC.DApt.Intf, FireDAC.Stan.StorageBin, FireDAC.Comp.DataSet,

FireDAC.Comp.Client,system.JSON, IdContext, IdBaseComponent, IdComponent,

IdCustomTCPServer,

IdTCPServer,

Da_globales,System.Generics.Collections,idglobal,dateutils;

type

TSensor = record

fecha : TDateTime;

id : string;

context : TIdContext;

end;

```
TDat_base = class(TDataModule)

  Servidor: TIdTCPServer;

  mt_datos: TFDMemTable;

  mt_datosID: TStringField;

  mt_datosHUMEDAD: TFloatField;

  mt_datosTEMPERATURA: TFloatField;

  mt_datosFECHA: TDateTimeField;

  DB_riego: TUniConnection;

  MySQLUniProvider1: TMySQLUniProvider;

  Q_datos: TUniQuery;

  C_insertar: TUniSQL;

  Q_datosCODIGO: TLongWordField;

  Q_datosFECHA: TDateTimeField;

  Q_datosTEMPERATURA: TFloatField;

  Q_datosHUMEDAD: TFloatField;

  Q_datosESTADO: TStringField;

  Q_datosRIEGO: TShortintField;

  Q_udatos: TUniQuery;

  Q_udatosCODIGO: TLongWordField;

  Q_udatosFECHA: TDateTimeField;

  Q_udatosTEMPERATURA: TFloatField;

  Q_udatosHUMEDAD: TFloatField;
```

```

Q_udatosESTADO: TStringField;

Q_udatosRIEGO: TShortintField;

procedure DataModuleCreate(Sender: TObject);

procedure ServidorConnect(AContext: TIdContext);

procedure ServidorDisconnect(AContext: TIdContext);

procedure ServidorExecute(AContext: TIdContext);

private

    { Private declarations }

public

    { Public declarations }

    ping : string;

    conexiones : TThreadList<TSensor>;

    function datosSensor(f_inicio,f_fin:TdateTime):string;

    function udatos():String;

    procedure activarAspersor(id:string);

    procedure desactivarAspersor(id:string);

end;

var

    Dat_base: TDat_base;

implementation

{%CLASSGROUP 'System.Classes.TPersistent'}

{$R *.dfm}

```

```
{ TDat_base }
```

```
procedure TDat_base.activarAspersor(id: string);
```

```
var
```

```
list_temp: TList<Tsensor>;
```

```
I: integer;
```

```
begin
```

```
try
```

```
list_temp := conexiones.LockList;
```

```
for I := 0 to list_temp.Count-1 do begin
```

```
if list_temp[I].id = id then begin
```

```
list_temp[I].context.Connection.Socket.Write('a');
```

```
Dat_globales.registrar_log('Comando a enviado IP  
:'+list_temp[I].context.Connection.ToString);
```

```
end;
```

```
end;
```

```
finally
```

```
conexiones.UnlockList;
```

```
end;
```

```
end;
```



```

procedure TDat_base.DataModuleCreate(Sender: TObject);

begin

try

    servidor.Active := true;

    conexiones := TThreadList<Tsensor>.Create;

    Dat_globales.registrar_log('Servidor activo','tcp.txt');

except on e:Exception do begin

    Dat_globales.registrar_error('activando servidor',e)

End;

end;

end;

function TDat_base.datosSensor(f_inicio,f_fin:TdateTime): string;

var

    tmp : string;

    jv : TJsonValue;

    ja : TJSONArray;

    jo_interno : TJsonObject;

begin

    ja := TJSONArray.Create;

    Q_datos.ParamByName('inicio').Value := f_inicio;

    Q_datos.ParamByName('fin').Value := f_fin;

    Q_datos.ExecSQL;

```

```

Q_datos.First;

while not Q_datos.Eof do begin

    jo_interno := TJSONObject.Create;

    jo_interno.AddPair('FECHA',DateTimeToStr( Q_datosFECHA.Value));

    jo_interno.AddPair('ID','0001');
jo_interno.AddPair('TEMPERATURA',TJSONNumber.Create(Q_datosTEMPERATURA.Value));

    jo_interno.AddPair('HUMEDAD',TJSONNumber.Create(Q_datosHUMEDAD.Value));

    jo_interno.AddPair('ESTADO',Q_datosESTADO.Value);

    jo_interno.AddPair('RIEGO',Q_datosRIEGO.Value);

    ja.Add(jo_interno);

    Q_datos.Next;

end;

Result := ja.ToString;

end;

procedure TDat_base.desactivarAspensor(id: string);

var

    list_temp: TList<Tsensor>;

    I: integer;

begin

try

    list_temp := conexiones.LockList;

    for I := 0 to list_temp.Count-1 do begin

```

```

if list_temp[I].id = id then begin

    list_temp[I].context.Connection.Socket.Write('i');

    Dat_globales.registrar_log('Aspersor Inactivo');

end;

end;

finally

    conexiones.UnlockList;

end;

end;

procedure TDat_base.ServidorConnect(AContext: TIdContext);

var

    datos: TIdBytes;

    cadena: string;

    list_temp: TList<TSensor>;

    sensor : TSensor;

    dsensor : TStringList;

    msg : string;

begin

if AContext.Connection.IOHandler.Readable then

    AContext.Connection.IOHandler.ReadBytes(datos,-1, False);

    cadena := Trim(BytesToString(datos,IndyTextEncoding_UTF8));

try

```

```

try

    sensor.fecha := now;

    sensor.id := '0001';

    sensor.context := AContext;

    list_temp := conexiones.LockList;

    list_temp.Add(sensor);

    Dat_globales.registrar_log('Conexion creada ', 'tcp.txt')

finally

    conexiones.UnlockList;

end;

Except on E:exception do

    Dat_globales.registrar_log('conexion_tcp_'+e.Message);

end;

end;

procedure TDat_base.ServidorDisconnect(AContext: TIdContext);

var

    list_temp: TList<Tsensor>;

    I: integer;

begin

    try

        try

            list_temp := conexiones.LockList;

```

```

I := 0;

while I < list_temp.Count do

  if list_temp[I].context = acontext then

    list_temp.Delete(I)

  else

    Inc(I);

    Dat_globales.registrar_log('Conexion eliminada ', 'tcp.txt')

  finally

    conexiones.UnlockList;

  end;

Except on E:Exception do

  Dat_globales.registrar_error('eliminar_conexion:', E);

end;

end;

procedure TDat_base.ServidorExecute(AContext: TIdContext);

va

  datos: TIdBytes;

  cadena: string;

  dsensor : TStringList;

  msg,suelo : string;

  temp , hum :double;

  riego : boolean;

```

```

begin

try

suelo := "";

if AContext.Connection.IOHandler.Readable then

    AContext.Connection.IOHandler.ReadBytes(datos,-1, False);

cadena := Trim(BytesToString(datos,IndyTextEncoding_UTF8));

dsensor := TStringList.Create;

dsensor.Delimiter := ':';

dsensor.StrictDelimiter := true;

dsensor.DelimitedText := cadena;

temp := StrToFloat(trim(dsensor.Strings[0]));

hum := StrToFloat(trim(dsensor.Strings[1]));

C_insertar.ParamByName('FECHA').Value := now;

C_insertar.ParamByName('TEMPERATURA').Value := temp;

C_insertar.ParamByName('HUMEDAD').Value := hum;

riego := false;

if (temp >= 20) and (temp <= 30) then begin

    if (hum >= 1200) and (hum <= 1800) then begin

        suelo := 'Exceso de humedad';

    end;

    if (hum > 1800) and (hum <= 3020) then begin

        suelo := 'Suelo humedo';

    end;

end;

```

```

end;

if (hum > 2800) and (hum <= 4000) then begin
    suelo := 'Suelo seco';
end;

if (hum > 4000) then begin
    suelo := 'Riego necesario';
end;

end else begin
    if (temp < 20) then
        suelo := 'Riego Bloqueado por temperatura baja'
    else
        if (temp > 30) then
            suelo := 'Riego Bloqueado por temperatura alta';
        end;
    end;

C_insertar.ParamByName('ESTADO').Value := suelo;

C_insertar.ParamByName('RIEGO').Value := riego;

C_insertar.Execute;

// Dat_globales.registrar_log('Datos guardados ' + IntToStr(mt_datos.RecordCount)+ '
'+dsensor.Strings[0], 'mt_datos.txt');

Except on E:exception do

    Dat_globales.registrar_log('execute '+e.Message)

```

```

end;

end;

function TDat_base.udatos: String;

var

    jo : TJsonObject;

begin

    Q_udatos.ExecSQL;

    Q_udatos.First;

    jo := TJSONObject.Create;

    jo.AddPair('FECHA',DateTimeToStr( Q_udatosFECHA.Value));

    jo.AddPair('ID','0001');

    jo.AddPair('TEMPERATURA',TJSONNumber.Create(Q_udatosTEMPERATURA.Value) );

    jo.AddPair('HUMEDAD',TJSONNumber.Create(Q_udatosHUMEDAD.Value));

    jo.AddPair('ESTADO',Q_udatosESTADO.Value);

    jo.AddPair('RIEGO',Q_udatosRIEGO.Value);

    result := jo.ToString;

end;

end.

```

PROGRAMACION DE INTERFAZ DE USUARIO DE LA APLICACIÓN WEB

```

unit Main;

interface

uses

```


Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, uniGUITypes, uniGUIAbstractClasses,
uniGUIClasses, uniGUIRegClasses, uniGUIForm, uniGUIBaseClasses, uniPanel,
uniLabel, Data.DB, uniBasicGrid, uniDBGrid, uniButton, uniChart,
uniDateTimePicker, uniTimer, uniScrollBar, uniImage, Vcl.Imaging.pngimage, dateutils;

type

TMainForm = class(TUniForm)

UniPanel1: TUniPanel;

UniLabel1: TUniLabel;

UniPanel2: TUniPanel;

UniPanel3: TUniPanel;

UniDBGrid1: TUniDBGrid;

DS_datos: TDataSource;

UniButton1: TUniButton;

UniPanel4: TUniPanel;

UniChart1: TUniChart;

lineahum: TUniLineSeries;

UniPanel5: TUniPanel;

UniDBGrid2: TUniDBGrid;

UniChart2: TUniChart;

lineatem: TUniLineSeries;

UniButton3: TUniButton;

```
UniPanel6: TUniPanel;

UniButton4: TUniButton;

UniPanel7: TUniPanel;

UniButton2: TUniButton;

UniLabel2: TUniLabel;

UniLabel3: TUniLabel;

UniDateTimePicker1: TUniDateTimePicker;

UniDateTimePicker2: TUniDateTimePicker;

udatos: TUniTimer;

lbl_temperatura: TUniLabel;

lbl_estado: TUniLabel;

lbl_humedad: TUniLabel;

UniLabel7: TUniLabel;

UniLabel5: TUniLabel;

UniLabel6: TUniLabel;

UniLabel4: TUniLabel;

UniScrollBar1: TUniScrollBar;

UniPanel8: TUniPanel;

UniImage1: TUniImage;

procedure UniButton3Click(Sender: TObject);

procedure UniButton2Click(Sender: TObject);

procedure UniButton1Click(Sender: TObject);
```

```

procedure UniFormCreate(Sender: TObject);

procedure UniButton4Click(Sender: TObject);

procedure udatosTimer(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

    procedure obtenerDatos();

end;

function MainForm: TMainForm;

implementation

{$R *.dfm}

uses

    uniGUIVars, MainModule, uniGUIApplication, Da_consumo;

function MainForm: TMainForm;

begin

    Result := TMainForm(UniMainModule.GetFormInstance(TMainForm));

end;

procedure TMainForm.obtenerDatos;

begin

    try

        Dat_consumo.mt_datos.DisableControls;

```

```

Dat_consumo.obtenerDatos(
UniDateTimePicker1.DateTime,UniDateTimePicker2.DateTime);

Dat_consumo.mt_datos.First;

lineahum.Clear;

lineatem.Clear;

while not Dat_consumo.mt_datos.Eof do begin

    with lineatem do

        begin

AddValues([Dat_consumo.mt_datosTEMPERATURA.Value],intToStr(Dat_consumo.mt_datosNR
osNRO.Value));

            end;

            with lineahum do

                begin

AddValues([Dat_consumo.mt_datosHUMEDAD.Value],intToStr(Dat_consumo.mt_datosNR
O.Value));

                    end;

                    Dat_consumo.mt_datos.Next;

                end;

            finally

                Dat_consumo.mt_datos.EnableControls;

            end;

        end;

    finally

        Dat_consumo.mt_datos.EnableControls;

    end;

end;

procedure TMainForm.udatosTimer(Sender: TObject);

```

```

var

    estado : string;

    temperatura,humedad : double;

    riego : boolean;

begin

    Dat_consumo.udatos(estado,humedad,temperatura,riego);

    lbl_estado.Caption := estado;

    lbl_humedad.Caption := FloatToStr(humedad);

    lbl_temperatura.Caption := FloatToStr(temperatura);

end;

procedure TMainForm.UniButton1Click(Sender: TObject);

begin

    Dat_consumo.desactivar;

    ShowMessage('ASPERSOR DETENIDO');

end;

procedure TMainForm.UniButton2Click(Sender: TObject);

begin

    obtenerDatos;

    Dat_consumo.mt_datos.First;

end;

procedure TMainForm.UniButton3Click(Sender: TObject);

begin

```

```

    Dat_consumo.activar;

    ShowMessage('ASPERSOR ACTIVO');

end;

procedure TMainForm.UniButton4Click(Sender: TObject);

var

    estado : string;

    temperatura,humedad : double;

    riego : boolean;

begin

    Dat_consumo.udatos(estado,humedad,temperatura,riego);

    lbl_estado.Caption := estado;

    lbl_humedad.Caption := FloatToStr(humedad);

    lbl_temperatura.Caption := FloatToStr(temperatura);

    udatos.Enabled := true;

end;

procedure TMainForm.UniFormCreate(Sender: TObject);

begin

    FormatSettings.DateSeparator := '/';

    FormatSettings.TimeSeparator := ':';

    FormatSettings.ListSeparator := ';';

    FormatSettings.DecimalSeparator := '.';

```

```

FormatSettings.ThousandSeparator:= ',';

FormatSettings.ShortDateFormat := 'DD/MM/YYYY';

FormatSettings.ShortTimeFormat := 'HH:NN';

FormatSettings.LongTimeFormat := 'HH:NN:.';

UniDateTimePicker1.DateTime := IncHour(now,-2);

UniDateTimePicker2.DateTime := now;

end;

initialization

  RegisterAppFormClass(TMainForm);

```

end.

PROGRAMACIÓN DEL CONSUMO DE DATOS DE LA INTERFAZ DE LA APLICACIÓN WEB

```

unit Da_consumo;

interface

uses

  SysUtils, Classes, REST.Types, REST.Client, Data.Bind.Components,

  Data.Bind.ObjectScope, FireDAC.Stan.Intf, FireDAC.Stan.Option,

  FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS, FireDAC.Phys.Intf,

  FireDAC.DApt.Intf, FireDAC.Stan.StorageBin, Data.DB, FireDAC.Comp.DataSet,

  FireDAC.Comp.Client,json;

```

type

```
TDat_consumo = class(TDataModule)
```

```
  R_cliente: TRESTClient;
```

```
  R_req: TRESTRequest;
```

```
  R_res: TRESTResponse;
```

```
  mt_datos: TFDMemTable;
```

```
  mt_datosID: TStringField;
```

```
  mt_datosHUMEDAD: TFloatField;
```

```
  mt_datosTEMPERATURA: TFloatField;
```

```
  mt_datosFECHA: TDateTimeField;
```

```
  mt_datosNRO: TIntegerField;
```

```
  procedure DataModuleCreate(Sender: TObject);
```

```
private
```

```
  { Private declarations }
```

```
public
```

```
  { Public declarations }
```

```
  procedure obtenerDatos(inicio,fin:TDateTime);
```

```
  procedure udatos(out estado : string ; out humedad,temperatuta : double ; out riego : boolean  
);
```

```
  function activar():boolean;
```

```
  function desactivar():boolean;
```

```
end;
```



```

const

// urldatos = 'http://127.0.0.1:8895/datos';

// urlactivarp = 'http://127.0.0.1:8895/activar';

// urldesactivarp = 'http://127.0.0.1:8895/desactivar';

// urludatosp = 'http://127.0.0.1:8895/udatos';

urldatos = 'http://3.141.41.125:8895/datos';

urlactivar = 'http://3.141.41.125:8895/activar';

urldesactivar = 'http://3.141.41.125:8895/desactivar';

urludatos = 'http://3.141.41.125:8895/udatos';

function Dat_consumo: TDat_consumo;

implementation

{$R *.dfm}

uses

    UniGUIVars, uniGUIMainModule, MainModule;

function Dat_consumo: TDat_consumo;

begin

    Result := TDat_consumo(UniMainModule.GetModuleInstance(TDat_consumo));

end;

```

```

function TDat_consumo.activar:boolean;

begin

    r_cliente.BaseURL := urlactivar;

    r_req.Method := rmGET;

    r_req.Execute;

end;

procedure TDat_consumo.DataModuleCreate(Sender: TObject);

begin

    FormatSettings.DateSeparator := '/';

    FormatSettings.TimeSeparator := ':';

    FormatSettings.ListSeparator := ',';

    FormatSettings.DecimalSeparator := '.';

    FormatSettings.ThousandSeparator:= ',';

    FormatSettings.ShortDateFormat := 'DD/MM/YYYY';

    FormatSettings.ShortTimeFormat := 'HH:NN';

    FormatSettings.LongTimeFormat := 'HH:NN:SS';

end;

function TDat_consumo.desactivar: boolean;

begin

```

```

r_cliente.BaseURL := urldesactivar;

r_req.Method := rmGET;

r_req.Execute;

end;

procedure TDat_consumo.obtenerDatos(inicio,fin:TDateTime);

var

tmp,resp : string;

lista : TJSONArray;

dsensor : TJsonObject;

i,j : integer;

j_req : TJsonObject;

fecha : string;

id_envio : string;

contenido : string;

arrstatus : TJSONArray;

status : TJsonObject;

iccid,deliveryStatus : string;

dsfecha : string;

f_zona :TDateTime;

begin

mt_datos.EmptyDataSet;

j_req := TJSONObject.Create;

```

```

j_req.AddPair('inicio', DateTimeToStr(inicio));

j_req.AddPair('fin',DateTimeToStr(fin));

r_cliente.BaseURL := urldatos;

R_req.Body.Add(j_req);

r_req.Method := rmGET;

r_req.Execute;

resp := r_req.Response.Content;

lista := TJSONObject.ParseJSONValue(resp) as TJSONArray;

if Assigned(lista) then begin

    for i := 0 to lista.Count - 1 do begin

        dsensor := lista.Items[i] as TJSONObject;

        mt_datos.append;

        mt_datosID.Value := dsensor.GetValue<string>('ID');

        fecha := dsensor.GetValue<string>('FECHA');

        fecha := StringReplace(fecha, '"', '"', [rfReplaceAll]);

        f_zona := StrToDateTime(fecha);

        mt_datosFECHA.Value := f_zona;

        mt_datosTEMPERATURA.Value := dsensor.GetValue<double>('TEMPERATURA');

        mt_datosHUMEDAD.Value := dsensor.GetValue<double>('HUMEDAD');

        mt_datosNRO.Value := i;

        mt_datos.Post;

    end;
end;

```

```

end;

end;

procedure TDat_consumo.udatos(out estado : string ; out humedad,temperatuta : double ; out
riego : boolean );

var

    tmp,resp : string;

    j_resp : TJsonObject;

begin

    r_cliente.BaseURL := urludatos;

    r_req.Method := rmGET;

    r_req.Execute;

    resp := r_req.Response.Content;

    j_resp := TJSONObject.ParseJSONValue(resp) as TJSONObject;

    estado := j_resp.GetValue<string>('ESTADO');

    humedad := j_resp.GetValue<double>('HUMEDAD');

    temperatuta := j_resp.GetValue<double>('TEMPERATURA');

    riego := j_resp.GetValue<boolean>('RIEGO');

end;

initialization

    RegisterModuleClass(TDat_consumo);

```

end.