

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



**IMPLEMENTACION DE UN SISTEMA DE CONTROL Y
ADMINISTRACION DE RIEGO DE INVERNADERO BASADO EN
TECNOLOGÍAS DEL INTERNET DE LAS COSAS PARA EL
MONITOREO DE VARIABLES AMBIENTALES**

PRESENTADO POR:

JOSUÉ ADONAI OSORIO MORALES

MOISÉS HUMBERTO QUINTANILLA MENJÍVAR

EDWIN LEONARDO ROMERO VÁSQUES

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, JULIO DE 2024

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. JUAN ROSA QUINTANILLA

SECRETARIO GENERAL:

LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

ING. LUIS SALVADOR BARRERA MANCÍA

SECRETARIO:

ARQ. RAÚL ALEXANDER FABIÁN ORELLANA

DIRECTOR INTERINO:

ING. WERNER DAVID MELÉNDEZ VALLE

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al grado de:

INGENIERO ELECTRICISTA

Titulo:

**IMPLEMENTACION DE UN SISTEMA DE CONTROL Y
ADMINISTRACION DE RIEGO DE INVERNADERO BASADO EN
TECNOLOGÍAS DEL INTERNET DE LAS COSAS PARA EL
MONITOREO DE VARIABLES AMBIENTALES**

Presentado por:

JOSUÉ ADONAI OSORIO MORALES

MOISÉS HUMBERTO QUINTANILLA MENJÍVAR

EDWIN LEONARDO ROMERO VÁSQUES

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSC. HUGO MIGUEL COLATO RODRÍGUEZ

SAN SALVADOR, JULIO DE 2024

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSC. HUGO MIGUEL COLATO RODRÍGUEZ

Agradecimientos

En primer lugar, quiero agradecer a mi mamá, por su amor incondicional, sacrificio y apoyo constante han sido la base para poder alcanzar mis logros. Gracias por creer en mí y por brindarme la fuerza necesaria para superar cada obstáculo.

A mis compañeros de tesis, gracias por darme la oportunidad de compartir este último proceso juntos. Gracias a su dedicación y apoyo hicieron que este desafío se lograra de la mejor manera. Aprendí mucho de ustedes durante los momentos que compartimos en el diseño y culminación de este proyecto.

Finalmente, me gustaría agradecer de manera especial a Reina Vides, una persona que nos ha ayudado muchísimo en esta última etapa. Su apoyo incondicional y su disposición para ayudarnos en todo lo posible han sido de mucha ayuda para poder culminar con éxito nuestro proyecto.

JOSUÉ ADONAI OSORIO MORALES

Al señor de los espacios infinitos.

Por permitirme estar aquí con vida y disfrutando este logro más para mí y para mi familia.

A mi madre.

Por ser un apoyo incondicional a lo largo de todo este tiempo en los momentos buenos y malos, por haber confiado y haber puesto su fe en mí, este logro es para ti también.

A mi padre.

A pesar de no estar físicamente, sigues siendo una fuente de inspiración y una fuerza que impulsó mis ganas de continuar, a pesar de todo lo sucedido, donde sea que estes, este logro es para ti.

A mi esposa.

Tanto que agradecer, y tan pocas las palabras. Por los muchos sacrificios que hiciste para poder estar en donde nos encontramos ahora, por haberme brindado un motivo más para poder continuar y seguir adelante, por caminar junto a mí en este camino, este logro es para ti.

A mi hija.

A mi combustible principal, al motivo de mi perseverancia, a esa personita que ve alguien grande en mí, este logro también es para ti.

A mi familia.

A todos los que creyeron en mí de corazón, que me brindaron su granito de arena y que me acompañaron, este gran mérito es para ustedes.

A mis amigos.

Que, con su apoyo dentro de la carrera o mi vida personal, me apoyaron, me estrecharon la mano, con los que pasamos buenos y malos momentos y además me acompañaron para que pudiera seguir adelante, este mérito es también de ustedes.

A mis compañeros de grupo

Y también grandes amigos, a quienes les quite la paciencia, tuvimos peleas, pero a pesar de todo logramos encontrar solución a los problemas como buen equipo, y finalmente lo logramos. Es un mérito nuestro.

Y a cada una de las personas que por buena fortuna fueron colocadas en mi vida, que brindaron su apoyo sin pedir nada a cambio y de las que aprendí mucho, les agradezco de todo corazón, este merito también es de ustedes.

MOISÉS HUMBERTO QUINTANILLA MENJÍVAR

Un agradecimiento especial a mi progenitora, la Licda. Francisca Concepción Vásquez de Romero, por su apoyo incondicional tanto en mi vida en general como en el proceso y culmen de esta carrera universitaria. Igualmente, al resto de mi familia por su compañía y apoyo.

Mi gratitud hacia Reina Vides, por su alto apoyo, comentarios y críticas constructivas que fueron fundamentales para el culmen de este y otros procesos académicos.

Sin olvidar a mi estimado Carlos Mejía, un colega compañero y sobre todo amigo que a pesar de las idas y venidas dentro y fuera de la vida estudiantil, esperando que mantener una relación de amistad y trabajo a lo largo del tiempo.

A un viejo compañero Ronaldo Salgado que con nuestras fortalezas y debilidades se pudieron lograr objetivos comunes en la carrera y también ver el crecimiento en la vida de cada uno.

A mi querido equipo de trabajo en este proyecto que a pesar de las diferencias, de la distancia y todo los problemas ajenos a nosotros, pudimos dar culmen a este y así pasar como equipo a una nueva etapa en la vida profesional.

Y por último agradecido por todo el que apoyo de manera directa e indirecta en el largo y difícil proceso de lo que fue lograr el título de ingeniero, a los amigos que están y a los que se fueron, a los compañeros de una materia o de una sola presentación y a los profesores que pasaron su conocimiento y lograr llegar al final del camino.

EDWIN LEONARDO ROMERO VÁSQUES

NOTA Y DEFENSA FINAL


En esta fecha, lunes 29 de abril de 2024, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 4:30 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Werner David Meléndez Valle
Director Interino


Firma

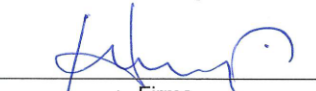


2. MSc. José Wilber Calderón Urrutia
Secretario


Firma

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

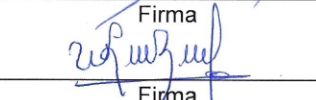
- MSC. HUGO MIGUEL COLATO RODRIGUEZ
(Docente Asesor)


Firma

- ING. WERNER DAVID MELENDEZ VALLE


Firma

- ING. WALTER LEOPOLDO ZELAYA CHICAS


Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RIEGO DE INVERNADEROS BASADO EN TECNOLOGÍAS DEL INTERNET DE LAS COSAS PARA EL MONITOREO DE VARIABLES AMBIENTALES

A cargo de los Bachilleres:

- OSORIO MORALES JOSUÉ ADONAI
- QUINTANILLA MENJÍVAR MOISÉS HUMBERTO
- ROMERO VÁSQUEZ EDWIN LEONARDO

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.5

(ocho punto cinco)

Tabla de Contenido

Índice de Figuras	1
Introducción.....	2
Objetivos.....	4
Alcances	5
Glosario.....	6
Capítulo I: Marco Teórico	8
1.1 Agricultura.....	9
1.1.1 Tipos de Agricultura	10
1.2 Sistemas de Riego.....	11
1.2.1 Red general de Riego	11
1.2.2 Elementos de Control de la Red de Riego.....	12
1.2.3 Métodos de Riego	13
1.2.3.1 Riego por Goteo	13
1.2.3.2 Riego Exudante	14
1.2.3.3 Sistemas de Control de Riego Automatizados	15
1.2.3.4 Riego por Goteo Subterráneo.....	15
1.3 Invernaderos	16
1.3.1 Tipos de Invernaderos	16
1.4 Internet de las Cosas (IoT)	21
1.4.1 Beneficios del Internet de las Cosas.....	22
1.5 Raspberry Pi	22
1.5.1 ¿Para qué sirve Raspberry Pi?.....	23
1.6 Automatización.....	24
1.6.1 Tipos de Automatización.....	25
1.6.2 Ventajas.....	26
1.7 Servicios Web para Sistemas de Control Automáticos.....	28
1.8 Bases de Datos	29
1.8.1 ¿Qué son las Bases de Datos?	29
1.8.2 Utilidad de las Bases de Datos	29
1.8.3 Tipos de Bases de Datos:	30
1.8.4 Ventajas de las Bases de Datos en Sistemas Automatizados:	30

Capitulo II: Diseño del Sistema	32
2.1 Diseño del hardware	33
2.1.1 Raspberry Pi en el proyecto	33
2.1.2 Sensores.....	34
2.1.2.1 Sensor de luminosidad (LDR).....	35
2.1.2.2 Sensor de Temperatura y Humedad Relativa del Ambiente (DHT11)	36
2.1.2.3 Sensor humedad en tierra (Capacitive Soil moisture sensor v1.2)	38
2.1.3 PCF8591 (Conversor ADC y DAC)	39
2.1.3.1 Tecnología I2C	41
2.1.3.1.1 Características.....	42
2.1.3.1.2 Funcionamiento del bus I2C	42
2.1.3.1.3 Fundamentos y Operación del Protocolo de Comunicación I2C....	43
2.1.3.1.4 Conversión Analógica Digital.....	44
2.1.4 Salida de Control de Riego	44
2.1.4.1 Optoacoplador (6N136).....	44
2.1.4.2 Relé	45
2.1.4.3 Electroválvulas y Sistema de Bombeo.....	45
2.2 Software	46
2.2.1 Estructura de Servicios	46
2.2.2 Docker	47
2.2.2.1 Tiempos de Ejecución	47
2.2.2.2 Comparación de los Contenedores de Docker con las Máquinas Virtuales	48
2.2.2.3 Docker dentro del Proyecto	49
2.2.2.3.1 Imágenes en los Contenedores	50
2.2.2.3.1.1 MariaDB.....	50
2.2.2.3.1.2 Node	50
2.2.2.3.1.3 Nginx.....	50
2.3 Estructura Final del Proyecto.....	50
2.4 Diseño del Prototipo.....	51
Capitulo III: Implementación.....	53
3.1 Explicación física del Prototipo	54

3.1.1 Proceso de armado del Prototipo	54
3.2 Hardware	59
3.2.1 Diagrama de Conexión	59
3.2.2 Modulo PCF8591	61
3.2.3 Modulo DHT11.....	61
3.2.4 Sistema de Control	62
3.2.5 Diagrama Esquemático Completo	62
3.3 Software	63
3.3.1 Estructura general del software utilizado y su configuración para el proyecto.....	64
Recomendaciones	69
Conclusiones.....	70
Bibliografía.....	72
Anexos	74
A.1 Preparación preliminar de Raspberry	74
A.1.1 OS (Operating System)	74
A.1.2 Instalación y configuración inicial de la Raspberry	74
A.1.3 Instalación de Servicios de Red Inalámbricos:	75
A.1.4 Creación del usuario a utilizar (moss):.....	76
A.2 Preparación preliminar e instalación de Docker	76
A.3 Preparación de Contenedores.....	78
A.4 Configuración de MySQL.....	79
A.4.1.1 Habilidad del Acceso Externo de la Base de Datos	79
A.4.1.2 Configuración de MySQL:.....	80
A.4.1.3 Creando tabla	81
A.4.1.4 Respaldo de base de datos.	82
A.5 Instalación y uso de Node.js.....	84
A.6 Instalación y uso de Nginx	87
A.7 Software de Terceros	87
A.7.1 TailScale	87
A.7.2 Insomnia.....	88

A.7.3 Instalación, configuración y pruebas con Software de Terceros en equipos ajenos a la Raspberry.	88
A.8 Enlace	91
A.8.1 Conexión a la red inalámbrica	91
A.8.2 Configuración servidor con IP estática	92
A.8.3 Enlace entre dispositivos	94
A.8.3.1 Ingreso al servicio SSH.	94
A.8.3.2 Envío de archivos por SCP	94
A.9 Implementación de Python en Raspberry y lectura de puertos	95
Referencias	97

Índice de Figuras

Figura 1. LDR diagrama esquemático y forma física.....	35
Figura 2. Diagrama esquemático del sensor de luminosidad.	36
Figura 3. DHT11 Aspecto físico y diagrama esquemático.	37
Figura 4. Sensor en su forma física y su circuito esquemático.....	38
Figura 5. IC PCF8591 en forma física y su diagrama esquemático.....	40
Figura 6. Diagrama de bloques de PCF8591	40
Figura 7. Forma física de 6N136 y diagrama esquemático	45
Figura 8. Forma física y diagrama esquemático del relé de 5 voltios.	45
Figura 9. Electroválvula.....	46
Figura 10. Diagrama de bloques del sistema de proyecto.	51
Figura 11. Capturas del diseño de prototipo en AutoCAD.	51
Figura 12. Diseño del prototipo en SketchUp.	52
Figura 13. Postes de tabla roca para la construcción del prototipo.	54
Figura 14. Perfil de tabla roca para la construcción del prototipo.	55
Figura 15. Canales de PVC para los cultivos.	55
Figura 16. Primeras fases de construcción del prototipo.....	56
Figura 17. Integración elementos eléctricos y electrónicos al prototipo.	57
Figura 18. Fase final de la construcción del prototipo.	57
Figura 19. Muestra del interior del prototipo armado.....	58
Figura 20 . Diagrama de Conexión.....	60
Figura 21. Diagrama esquemático del módulo PCF8591.	61
Figura 22. Diagrama esquemático control de electroválvulas.	62
Figura 23. Diagrama esquemático completo del proyecto.	63
Figura 24. Diagrama de bloques de la estructura general del software.	66
Figura 25. Diagrama de la estructura de la Web APP.	67
Figura 26. Sistema Operativo.....	74
Figura 27. Arquitectura de Contenedores.....	76
Figura 28. Especificaciones de MariaDB.....	77
Figura 29. Contenedores levantados.	78
Figura 30. Verificación de servicios levantados.....	78
Figura 31. Tabla resultante en MySQL.....	82
Figura 32. QUERY ingreso de datos.	85
Figura 33. Obtención de base de datos.....	86
Figura 34. Archivo de Backend.	86
Figura 35. Otro Software Utilizado.....	88
Figura 36. Interfaz de conexión de DBeaver.....	89
Figura 37. Conexión de DBeaver.....	90
Figura 38. Base de datos consultada por DBeaver.	90
Figura 39. Enlace	91

Introducción

En la era moderna, la integración de la tecnología desempeña un papel crucial en la evolución y eficiencia de la agricultura. Este documento se centra en explorar cómo las tecnologías emergentes pueden modernizar el sector agrícola, mejorando la gestión de recursos y aumentando la productividad.

En el Capítulo I, se proporciona una base sólida sobre los fundamentos de la agricultura, abordando la clasificación de diferentes tipos de agricultura y sistemas de riego, así como la variedad de invernaderos disponibles. Se examina el potencial del Internet de las Cosas (IoT) y se destacan sus beneficios en la monitorización y control de condiciones ambientales. Además, se discute la automatización y la importancia de las bases de datos para la gestión integral de información agrícola.

El Capítulo II se dedica al diseño detallado del sistema propuesto, comenzando con la especificación de los componentes de hardware necesarios. Se destaca la integración estratégica de la Raspberry Pi como núcleo del sistema, junto con la descripción de los sensores empleados y la técnica de conversión de señales mediante ADC y DAC. Se analiza la implementación de la tecnología I2C para la comunicación entre dispositivos, y se detalla el diseño del sistema de control de riego. Finalmente, se presenta un diagrama completo del sistema integrado, así como el diseño del prototipo desarrollado.

El Capítulo III se enfoca en la implementación práctica del sistema diseñado. Se ofrece una guía detallada del montaje físico del prototipo, incluyendo el proceso de ensamblaje y los diagramas de conexión necesarios. Se describe exhaustivamente el sistema de control agrícola, complementado con un

diagrama esquemático completo del sistema implementado. Por último, se abordan los aspectos clave relacionados con la estructura y configuración del software esencial para el funcionamiento óptimo del sistema automatizado en el entorno agrícola.

En conjunto, este trabajo no solo busca explorar las aplicaciones prácticas de la tecnología en la agricultura moderna, sino también ofrecer un modelo integral y replicable que pueda mejorar significativamente la eficiencia y sostenibilidad de las prácticas agrícolas contemporánea

Objetivos

General

- Diseñar un sistema de control y monitoreo de variables ambientales en invernaderos utilizando tecnologías de Internet de las cosas (IoT)

Específicos

- Implementar un sistema de monitoreo y supervisión de cultivos.
- Integrar la placa programable Raspberry Pi 2B como parte del sistema de control automático.
- Desarrollar un sistema de control automático interactuando desde un cliente web.
- Utilizar tecnología IoT para desarrollar una interfaz de control de un invernadero.
- Diseñar una base de datos que permita la administración eficiente del sistema.

Alcances

- Diseñar y desarrollar un sistema de control y monitoreo de variables ambientales en invernaderos utilizando IoT, incluyendo la selección de sensores, la implementación de una red de sensores, la adquisición y procesamiento de datos, y la visualización de los resultados.
- Implementar un sistema de riego automatizado con dispositivos de control como válvulas y actuadores, garantizando un suministro óptimo de agua basado en parámetros ambientales.
- Desarrollar una interfaz de usuario basada en web (HMI) para monitorear y supervisar cultivos, permitiendo a los agricultores visualizar y controlar el estado de los cultivos mediante gráficos interactivos y alertas personalizadas.
- Evaluar y validar el sistema mediante pruebas de campo y análisis de datos, asegurando su efectividad y eficiencia en el control y monitoreo de variables ambientales.

Glosario

IoT: De las siglas del inglés “Internet of Things” o en español Internet de las Cosas.

ADC: De las siglas del inglés “Analog to Digital Converter” o en español convertidor analógico-digital.

SCL: Es la línea por donde se transmite la señal de reloj del sistema.

SDA: Es la línea por la que se transmite la señal de datos.

LDR: Un dispositivo cuya resistencia varía de acuerdo con la cantidad de luz que recibe.

Docker: Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Server: Conjunto de computadoras capaces de atender las peticiones de un cliente y devolverle una respuesta en concordancia.

Kernel: Parte fundamental del sistema operativo que se encarga de conceder el acceso al hardware de forma segura para todo el software que lo solicita.

Host: Cualquier computadora o máquina conectada a una red mediante un número de IP definido y un dominio, que ofrece recursos, información y servicios a sus usuarios.

Raspberry Pi: Computadora muy económica que ejecuta GNU/Linux, pero también proporciona un conjunto de pines GPIO (general purpose input/output), lo que le permite controlar componentes electrónicos para la computación física y explorar el Internet de las Cosas (IoT).

GPIO: De las siglas en inglés “General Purpose Input/Output” o en español sistema de entrada y salida de propósito general.

Tecnología I2C: El término I2C da nombre a un estándar para bus de comunicaciones serie. El nombre proviene de “Inter-Integrated Circuits”.

Servidor SSH: Acrónimo de Secure Shell, es un protocolo que se utiliza en el manejo de servidores de forma remota, permitiendo a un usuario realizar toda clase de tareas sobre el mismo.

Capítulo I: Marco Teórico

1.1 Agricultura

La agricultura, actividad esencial del sector primario, se centra en el cultivo de plantas para producir alimentos, fibras y medicinas, elementos fundamentales para la vida humana. Esta actividad ha sido crucial en el desarrollo de las civilizaciones, facilitando el establecimiento de comunidades y el crecimiento urbano (Smith, 2020).

Con el avance tecnológico, la agricultura ha experimentado transformaciones significativas. La introducción de sistemas de riego automatizados y sensores de humedad ha permitido optimizar el uso del agua y mejorar la eficiencia en la producción agrícola (Jones & Brown, 2018).

Un sistema de riego automatizado es una tecnología que utiliza sensores y controladores para gestionar el riego de los cultivos de manera eficiente. Este sistema monitorea continuamente la humedad del suelo y ajusta el riego según las necesidades de las plantas, reduciendo el desperdicio de agua y mejorando la productividad (Lee et al., 2019).

Los sensores de humedad del suelo, como los sensores capacitivos, miden la capacidad del suelo para retener agua. Estos sensores envían datos en tiempo real al controlador del sistema de riego, que regula la cantidad de agua suministrada a los cultivos. Su precisión y confiabilidad son esenciales para el correcto funcionamiento del sistema (García & Martínez, 2021).

Los controladores programables, como los microcontroladores Arduino, se utilizan para procesar la información recibida de los sensores y ejecutar las acciones necesarias para el riego. Estos dispositivos pueden ser programados para ajustar automáticamente los tiempos y la cantidad de riego, optimizando así el uso de recursos (Williams, 2017).

1.1.1 Tipos de Agricultura

La agricultura se puede clasificar en diferentes tipos según varios criterios de análisis:

Por los volúmenes de producción:

- **Agricultura de subsistencia.** Tiene como objetivo obtener un nivel bajo de producción, con la única finalidad de alimentar a una comunidad estable y reducida de personas. Por lo tanto, no produce un gran desgaste en el suelo.
- **Agricultura Industrial.** Consiste en la producción de grandes volúmenes de alimentos provenientes del suelo, propio de los países industrializados y en vías de desarrollo. Esta práctica tiene el objetivo no simplemente de garantizar la satisfacción de las necesidades de la comunidad, sino también el comercializar el excedente, como ocurre con las exportaciones de alimentos agrícolas en el extranjero.

Por la importancia del agua en la producción:

- **De Regadío.** En este tipo de agricultura es fundamental un sistema de riego aportada por el agricultor, utilizando métodos naturales o artificiales.
- **De Secano.** La humedad necesaria para la producción es suministrada por las lluvias y por el suelo, sin intervención del agricultor.

Según los medios de producción utilizados y su rendimiento:

- **Agricultura Extensiva.** El objetivo tenido en cuenta no es tanto el rédito económico sino el cuidado del suelo. Se utilizan superficies amplias de tierra, pero se obtienen niveles bajos de producción.
- **Agricultura Intensiva.** Se busca la producción en masa en un espacio reducido de suelo, lo cual puede ser perjudicial para el medio ambiente. Se utiliza generalmente en países industrializados.

De acuerdo a la técnica utilizada y su objetivo:

- **Agricultura Industrial.** Esta forma de producción tiene como finalidad obtener magnitudes enormes de alimentos, teniendo en miras su comercialización.
- **Agricultura Ecológica.** Esta forma de agricultura tiene como prioridad la no alteración del medio ambiente y el cuidado del suelo, utilizando métodos y tecnologías adecuadas.
- **Agricultura Tradicional.** Se caracteriza por utilizar técnicas y procedimientos autóctonos de una región determinada, extendiéndose en el tiempo y formando parte de la cultura del lugar.

1.2 Sistemas de Riego

Los sistemas de riego ofrecen diversas ventajas que permiten un uso más eficiente del agua disponible. Es esencial realizar un estudio previo para determinar el sistema de riego más adecuado, considerando factores como el tipo de vegetación y la distribución del agua para lograr el mejor rendimiento. Los instrumentos de control de riego, tales como programadores, higrómetros y detectores de lluvia, deben ubicarse según la orografía, la capacidad hídrica del suelo y las plantaciones.

Existen múltiples sistemas de riego, los cuales están en constante revisión debido a que esta es una tecnología joven que ha evolucionado junto con el desarrollo de la sociedad del bienestar. Las zonas verdes, que antes eran un lujo, ahora son una necesidad, y el riego es la operación más crucial para su mantenimiento.

1.2.1 Red general de Riego

Distribuidores de agua. Elementos destinados a distribuir el agua de acuerdo con una pluviometría predeterminada: aspersores, difusores, bocas de riego, goteros.

La presión de red se obtiene de la red general de la ciudad. Sin embargo, también es frecuente usar las bombas conectadas a depósitos, albercas o estanques para asegurar una presión adecuada.

1.2.2 Elementos de Control de la Red de Riego

Los elementos que mejoran la automatización de las redes de riego permiten regular y controlar los caudales, los tiempos y otras características importantes. Los equipos que forman parte de las instalaciones de riego ofrecen grandes ventajas:

- Mayor exactitud y seguridad en el control de la instalación.
- Operaciones mecanizadas exentas de errores.
- Reducción de mano de obra.
- Telemando y facilidad de programación y manejo.
- Registro de datos para su análisis posterior.

Para la realización de todas estas operaciones y alcanzar los objetivos propuestos, son necesarios varios equipos individuales conectados de manera adecuada:

- **Electroválvulas:** Regulan el paso del agua a través de la canalización de manera automática. El sistema de accionamiento puede ser eléctrico, hidráulico o mixto.
- **Pluviómetros:** Funciona por impulsos eléctricos y desconectan el programa de riego si llueve. Una pequeña cubeta de PVC recoge el agua de lluvia, y en el interior hay dos electrodos que funcionan como un interruptor por el efecto conductor del agua que se acumulada.
- **Higrómetros:** Controla el riego con mayor precisión que los pluviómetros, ya que miden el grado de humedad del suelo en cada momento mediante sondas.

- **Programadores:** Regulan el riego y funcionan como el cerebro del sistema, ajustando el riego según las necesidades de las plantas y minimizando el consumo de agua. Existen tres tipos principales:
 - Electromecánicos: Tienen un pequeño motor eléctrico que mueve diversos relojes mecánicos donde se determinan los horarios.
 - Electrónicos: Son precisos en sus órdenes y adecuados para pequeñas instalaciones.
 - Híbridos: Combinan las ventajas de los programadores electrónicos y la facilidad de uso de los electromecánicos, ofreciendo exactitud y simplicidad.
- **Válvulas:** Suelen estar construidas de latón, fundición, o plásticos, especialmente PVC. Su función específica es regular el paso del agua a través de una canalización. Las válvulas manuales se denominan de control, mientras que las que actúan de acuerdo a un parámetro de la propia agua se llaman de regulación.

1.2.3 Métodos de Riego

Entre los diferentes métodos de riego se pueden enlistar los siguientes:

1.2.3.1 Riego por Goteo

El riego por goteo se destaca como una técnica ideal para el control preciso del agua en invernaderos, debido a su eficiencia y capacidad de optimizar el uso del recurso hídrico. Este método consiste en la aplicación de agua de forma localizada y controlada, directamente a la zona radicular de las plantas, evitando el desperdicio por evaporación o escorrentía.

Ventajas:

- **Ahorro de agua:** Reduce el consumo de agua en un 50% o más en comparación con otros métodos.

- **Control preciso:** Permite la aplicación de agua de forma precisa y eficiente, adaptándose a las necesidades específicas de cada planta.
- **Nutrición localizada:** Facilita la aplicación de fertilizantes y nutrientes directamente a las raíces, mejorando la absorción y el crecimiento.
- **Minimización de enfermedades:** Reduce la humedad en el suelo, lo que disminuye el riesgo de enfermedades fúngicas y bacterianas.

Desventajas:

- **Mantenimiento:** Requiere un mantenimiento regular para evitar obstrucciones en los goteros.
- **Instalación:** Puede ser más costosa de instalar que otros métodos de riego.

1.2.3.2 Riego Exudante

El riego exudante, también conocido como "tubería exudante", es una variante del riego por goteo que utiliza una red de tuberías porosas para distribuir el agua de forma uniforme y constante. El agua se filtra a través de las paredes porosas de la tubería, humedeciendo el suelo de manera gradual y controlada.

Ventajas:

- **Riego uniforme:** Proporciona una distribución uniforme del agua a lo largo de la tubería, asegurando una hidratación constante para las plantas.
- **Ahorro de agua:** Reduce el consumo de agua en un 35% a 45% en comparación con otros métodos.
- **Fácil instalación:** Es relativamente fácil de instalar, especialmente en comparación con sistemas de riego por goteo más complejos.

Desventajas:

- **Presión de agua:** Requiere una presión de agua específica para un funcionamiento óptimo.
- **Mantenimiento:** Puede requerir limpieza periódica para evitar obstrucciones en las tuberías.

1.2.3.3 Sistemas de Control de Riego Automatizados

La automatización del riego en invernaderos es fundamental para optimizar el uso del agua y garantizar un crecimiento óptimo de las plantas. Los sistemas de control de riego automatizados utilizan sensores para monitorear las condiciones ambientales, como la humedad del suelo, la temperatura y la humedad relativa, y ajustan el riego de forma automática para satisfacer las necesidades de las plantas.

Ventajas:

- **Optimización del riego:** Ajusta el riego en tiempo real, adaptándose a las necesidades específicas de las plantas y las condiciones ambientales.
- **Ahorro de agua:** Reduce el desperdicio de agua al aplicar solo la cantidad necesaria.
- **Eficiencia:** Libera tiempo y esfuerzo al automatizar el proceso de riego.
- **Control remoto:** Permite monitorear y controlar el riego de forma remota, desde cualquier lugar con conexión a internet.

Desventajas:

- **Costo:** Los sistemas de control de riego automatizados pueden ser costosos de instalar.
- **Mantenimiento:** Requieren mantenimiento regular para garantizar su correcto funcionamiento.

1.2.3.4 Riego por Goteo Subterráneo

El riego por goteo subterráneo es una técnica que utiliza goteros instalados debajo de la superficie del suelo, lo que permite una distribución más eficiente del agua y reduce la evaporación.

Ventajas:

- **Mayor eficiencia:** Reduce la pérdida de agua por evaporación y escorrentía.
- **Menor riesgo de enfermedades:** Reduce la humedad en la superficie del suelo, disminuyendo el riesgo de enfermedades fúngicas y bacterianas.

- **Estética:** No afecta la estética del invernadero, ya que los goteros no son visibles.

Desventajas:

- **Instalación:** Puede ser más complejo de instalar que otros métodos de riego.
- **Mantenimiento:** Requiere un mantenimiento regular para evitar obstrucciones en los goteros.

Es decir, los métodos de riego descritos, especialmente el riego por goteo y los sistemas de control automatizados, son esenciales para el desarrollo de un sistema de control de riego de invernadero basado en IoT. La selección del método más adecuado dependerá de las características del invernadero, las necesidades de las plantas y los recursos disponibles.

1.3 Invernaderos

Un invernadero es una estructura diseñada para el cultivo de plantas en un ambiente controlado. Se caracterizan por ser estructura alta y fija que permite controlar de manera eficiente factores ambientales como la temperatura, la humedad, la luz y la ventilación para obtener un buen desarrollo de los cultivos. Esto a su vez posibilita la producción de cultivos en condiciones óptimas durante todo el año, independientemente de las estaciones climáticas externas.

1.3.1 Tipos de Invernaderos

Dentro de los tipos de invernaderos más comunes en el mundo se encuentran:

1. Invernadero-Túnel.

Se compone de uno o varios módulos que incluyen arcos contruidos a partir de tubos cilíndricos galvanizados. Estos arcos no requieren de cimientos de hormigón, lo que facilita su movilidad y sencilla instalación. Su forma permite alojar un volumen mayor de aire en su interior y proporciona resistencia a lluvia.

Ventajas

- Se trata de un tipo de invernadero barato y sencillo.
- Ofrecen una mejor capacidad de control del clima que el invernadero plano.
- Permite la instalación de sistemas de climatización.
- Mayor capacidad de estanqueidad que el invernadero plano.
- Buen reparto de la luminosidad en el interior del invernadero.
- Reduce considerablemente el problema de la condensación y el goteo del agua en los cultivos debido a la cubierta curva, la cual favorece la evacuación hacia las paredes del agua proveniente de la condensación en la cubierta plástica.
- Permite la instalación de ventanas cenitales y laterales.
- Facilita las operaciones agrícolas con maquinaria.
- Montaje rápido y sin soldaduras.

Características Estándares

Las dimensiones Estándares de este tipo de invernaderos son las siguientes:

- Ancho: 8- 9.60 m.
- Altura al cenit: 4 - 5 m.
- Distancia entre arcos: 2,50 m. (externas).
- Bastidores de refuerzo perimetrales.

2. Invernaderos Capilla

El invernadero tipo capilla o también denominado multi-capilla, se caracteriza por la forma de su cubierta formado por arcos curvos semicirculares y por su estructura totalmente metálica.

Este tipo de invernadero entraría dentro de la categoría de los comúnmente denominados multi-tunel, junto a los invernaderos de tipo Gótico e Invernaderos de tipo Asimétrico. El montaje se realiza a modo de Mecano. Las diferentes partes se unen con grapas, tuercas y tornillos, por lo que no es necesario soldar.

Ventajas de los invernaderos tipo Capilla

- Pocos obstáculos en su estructura.
- Buena ventilación.
- Buena estanqueidad a la lluvia y al aire.
- Permite la instalación de ventilación cenital, así como ventilación perimetral
- Buen reparto de la luminosidad en el interior del invernadero.
- Fácil instalación.

Dimensiones Estándares de este tipo de invernaderos son las siguientes:

- Ancho: 8 m - 9.60 m
- Altura bajo canal: 4 m - 5 m – 5.50 m
- Altura al zenit: 5.80 m - 6.30 m - 6.80 m
- Separación entre pilares: 5 m (interior) – 2,50 m (exterior)

3. Invernaderos Malla Sombra

Los invernaderos malla sombra o también denominados casa sombra constan de las estructuras más simples dentro de los diferentes tipos de invernaderos.

Consiste en un invernadero compuesto por tubos galvanizados o perfiles metálicos y dos mallas de alambre superpuestas que portan y sujetan la lámina de plástico, cuya cubierta es plana. Se puede adaptar a cualquier tipo de terreno con el objetivo de maximizar el uso del terreno.

Ventajas

- Se trata del invernadero más barato y sencillo.
- Gran adaptabilidad a la geometría del terreno.
- Presenta una gran uniformidad luminosa.
- Montaje rápido y sin soldaduras.

Dimensiones estándares de este tipo de invernaderos son las siguientes:

- Ancho: 8- 9.60 -10 m.
- Altura al cenit: 7 m
- Distancia entre arcos: 8 - 9.60 - 10 m. (internos) 5 m. (externos)
- Bastidores de refuerzo perimetrales
- Materiales: Malla

4. Invernaderos Góticos

El tipo de Invernadero Gótico se diferencia del tipo capilla en el diseño de los arcos, siendo estos de tipo ojival, permite albergar un mayor volumen de aire, proporcionando un mejor microclima e iluminación interior.

Está diseñado para adaptarse a todo tipo de cultivos, particularmente a cultivos suspendidos y su construcción está orientada a climas extremos.

Son estructuras diseñadas para soportar grandes cargas además de exigir ciertos cuidados y condiciones ambientales para el cultivo. Al ser la cumbre de tipo gótico, nos permite construir naves más anchas, con la ventaja que supone el aumento de superficie de cultivo.

Ventajas

- Alta duración y resistencia a la corrosión.
- Eficacia de fijación del plástico de cubierta (buena hermeticidad).
- Montaje rápido y sin soldaduras.
- Mayor distancia de la ventilación al cultivo.
- Fácil deslizamiento de la condensación.
- Mayor entrada de luz.
- Mayor ventilación.
- Permite realizar labores agrícolas mecanizadas en su interior.

Dimensiones Estándares de este tipo de invernaderos son las siguientes:

- Ancho: 8 m - 9.60 m dependiendo del modelo.
- Altura bajo canal: 4 - 4,50 – 5 m.
- Altura al cenit: entre 6 y 7,40 metros dependiendo del modelo.
- Distancia entre pilares: 5 m internas – 2,50 m externas.

5. Invernadero Tropical o Asimétrico

Se denominan Invernaderos Tropicales porque su uso está muy extendido en estas regiones, y Asimétrico, porque, a diferencia de los invernaderos tipo capilla y góticos, su geometría es asimétrica, siendo uno de los lados de la cubierta más inclinado que el otro.

La inclinación de la cubierta se estudia en función de la incidencia perpendicular sobre la misma de la luz al medio día solar, durante el invierno, con el objetivo de aprovechar al máximo la radiación solar incidente.

La ventilación de este invernadero suele ser fija y es resuelta a través de las aperturas localizadas en el centro de cada uno de los arcos estructurales que corren a lo largo de todo el techo. Las aperturas permiten ventilación natural y la salida de aire caliente.

Ventajas

- Buen aprovechamiento de la luz en la época invernal.
- Elevada inercia térmica debido a su gran volumen unitario.
- Es estanco a la lluvia y al aire.
- Buena ventilación debido a su elevada altura.
- Permite la instalación de ventilación cenital a sotavento.
- Resistencia a fuertes vientos.
- Montaje rápido y sin soldaduras.

Dimensiones Estándares de este tipo de invernaderos son las siguientes:

- Ancho: 9.60 m.
- Altura del pilar: 6,4 - 6,90 - 7,40 m.
- Altura debajo canal: 4 - 4,50 - 5 m.
- Distancia entre pilares: 4 - 5 m. (internos). 2 – 2,50 m. (externos).
- Bastidores de refuerzo perimetrales.

1.4 Internet de las Cosas (IoT)

El Internet de las cosas es un sistema tecnológico que permite que los objetos se conecten a Internet y entre sí. Consiste en sistemas ciber físicos, computación integrada que usan la infraestructura de Internet y las aplicaciones y servicios que la utilizan.

Se trata de una nueva tecnología que conecta casi todo lo que hacemos. Facilita un ecosistema interconectado de dispositivos y máquinas, lo que permite a los usuarios controlar sus dispositivos desde cualquier lugar. Hay muchos beneficios de esta nueva tecnología, como mejorar la productividad y reducir el impacto ambiental.

Internet de las Cosas (IoT) es un término amplio que se utiliza para la interconexión de objetos cotidianos con Internet o entre sí, contemplado en los estudios en informática. Estos dispositivos incluyen teléfonos inteligentes, automóviles, televisores, relojes, electrodomésticos y mucho más.

1.4.1 Beneficios del Internet de las Cosas

- Facilidad de acceso a la información

Internet de las cosas ha recibido mucha más prensa a medida que esta nueva tecnología se está abriendo camino lentamente en los hogares y negocios. Un dispositivo que está conectado a Internet puede considerarse un dispositivo de IoT.

- Eficiencia y productividad

IoT se refiere a una red creciente de objetos físicos que están conectados a Internet. Estos dispositivos inteligentes son cada vez más frecuentes en la vida cotidiana, ya que pueden ayudar a administrar el tiempo, energía, recursos, bienes, servicios y mucho más.

- Ahorro

Con esta tecnología usada en los hogares o en las empresas, es posible saber de forma sencilla si uno de los dispositivos necesita arreglarse o cambiarlo. Esto permite poder planificar y evitar males mayores.

1.5 Raspberry Pi

Raspberry Pi es una placa de microordenador, que como su propio nombre indica, es de pequeñas dimensiones a la cual se le pueden dar multitud de usos como veremos más adelante. La Raspberry Pi apareció en febrero de 2012, 6 años después de que comenzara el proyecto principal de esta placa, y para agosto del 2012 ya habrían vendido 500.000 unidades, y un mes después ya se había realizado la primera revisión "B" de la placa original. La primera unidad tenía 256 MB de RAM y un procesador a 700 MHz, tenía el característico conector de 26 pines GPIO y

salida de video por HDMI o RCA además de un conector de 3.5mm para el audio, el primer modelo carecía de puerto ethernet.

1.5.1 ¿Para qué sirve Raspberry Pi?

La Raspberry Pi sirve para llevar la informática a todo el mundo, con ella se pueden realizar las tareas más comunes de un ordenador, pero por tan solo un precio aproximado de \$35 con una Raspberry Pi y un sistema Raspbian se puede navegar por internet, consultar el correo electrónico, reproducir videos, usar las aplicaciones de mensajería instantánea, etc. Todo esto en un muy reducido tamaño y con un coste muy inferior a cualquier otro ordenador de escritorio.

También se puede usar como un ordenador portable, aunque no portátil, ya que depende de una toma de red eléctrica, de una pantalla, ya sea monitor o televisión, y de un teclado y ratón (aunque no siempre) donde conectarla para hacerla funcionar, pero dadas sus diminutas dimensiones, incluso si se coloca en el interior de una de las muchas cajas que hay para esta, se puede llevar en un bolsillo o una mochila, junto con el adaptador de corriente necesario, y lista para conectarla en cualquier pantalla junto con un teclado y ratón.

La fundación Raspberry Pi quería hacer llegar la informática a todos los usuarios, con esta sencilla placa, pero suficiente para hacer las tareas más comunes de cualquier ordenador, la informática está al alcance de todos los usuarios e incluso bolsillos. También se ha introducido para la enseñanza de informática en escuelas, se puede comprar en lotes para el sector de la educación desde su propia página Web que nos redirigirá al distribuidor asociado para esto.

El microordenador Raspberry Pi ha pasado por varias versiones de mejora, pero continua con su reducido precio de aproximadamente 35\$, para funcionar utiliza un sistema operativo, Raspbian, basado en Linux y especial para aprovechar el hardware de este microordenador. Además de Raspbian, multitud de sistemas han

aparecido que hacen de la Raspberry Pi un microordenador para diferentes fines, centros multimedia, emuladores de máquinas recreativas, etc.

La Raspberry Pi también dispone de una amplia gama de accesorios, como cajas con refrigeración pasiva de Cooler Master y periféricos para conectar a sus pines GPIO con los que se pueden hacer muchísimas más cosas. Se pueden conectar cámaras NoIR, sintonizadores de TDT, pantallas táctiles, módulos WI-FI, etc. También está disponible en otros formatos, especiales para sistemas embebidos, para programación o de reducido tamaño para pequeños proyectos, como las Raspberry Pi Zero con y sin WI-FI. Incluso hay una Raspberry Pi integrada dentro de un teclado de la marca, la Raspberry Pi 400.

Particularmente se eligió la Placa Raspberry debido a su tamaño su fácil transporte y la complejidad que esta puede albergar, a diferencia de un ARDUINO que solamente corre un código pre establecido, esta corre directamente un sistema operativo del cual se pueden instalar una infinidad de scripts, servicios, etc. Sirviendo de mucho a la hora de utilizarla dentro del proyecto debido a la cantidad de información que esta procesara y de los servicios necesarios para el funcionamiento completo del proyecto.

1.6 Automatización

El concepto de automatización está enfocado en una disciplina de control que se basa en el uso de sistemas electromecánicos para controlar de forma automatizada diversos procesos industriales. Abarca control, sistemas digitales, supervisión, gestión de datos, accionamientos, instrumentación, comunicaciones, producción, interacciones y muchos otros.

La automatización incorpora elementos y dispositivos tecnológicos que aseguran tener un control específico sobre los procesos y sus evidentes comportamientos. Esta automatización debe ser capaz de poder controlar efectivamente todo el

conjunto de posibles eventos previstos frente a posibles ocurrencias; buscando siempre lograr la situación más favorable según la determinación de recursos asignados.

La automatización es muy usada en diferentes áreas de trabajo. Los elementos y características de la automatización han tenido importante impacto en el área industrial, mecánica, informática, máquinas y programables.

La alta competitividad empresarial y la necesidad de aumentar eficazmente los procesos de producción mediante la incorporación de la robótica, los robots y automatizar los procesos, han implicado un mayor nivel de integración entre los sistemas productivos y la decisión política empresarial en áreas de fabricación, gestión de procesos, servicios y la gestión de la información.

1.6.1 Tipos de Automatización

- **Automatización Mecánica:** Es un tipo de automatización donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos mecánicos, por sistemas automatizados con partes de mando y parte operativas.
- **Automatización Industrial:** Es un tipo de automatización que involucra la incorporación y uso sistemático de varios tipos de sistemas o elementos computarizados, electromecánicos, electro neumáticos y electrohidráulicos con fines directamente industriales.
- **Automatización Informática:** Es un tipo de automatización que consiste principalmente en el uso de sistemas de software que permiten generar y crear instrucciones, con la utilización de diversos procesos repetibles para reemplazar o reducir la interacción humana en tecnologías de información.
- **Automatización de Máquinas:** Esta definida como el proceso de automatización en donde se transfieren diversas tareas de producción, que

son generalmente ejecutadas por operadores humanos hacia un conjunto de elementos de forma tecnológica.

- **Automatización Programable:** Es un tipo de automatización que incorpora dentro de un sistema secuencia las operaciones de proceso y ensamblaje, que son determinadas por la configuración del equipo de forma programada.

La historia y evolución de la automatización ha estado necesariamente ligada a generar una reducción sostenida en los posibles costes de fabricación de cualquier producto o servicio. Ha sido una tarea que ha tenido como principal enfoque mejorar los medios de producción en función de reducir posibles tareas que sean tediosas y peligrosas. Por ello la evolución de la automatización, ha ido creciendo de forma más importante en la medida que ha sucedido una incorporación cada vez mayor de la competitividad empresarial y las demandas de mercado.

La historia y la evolución de la automatización en la actualidad ha estado enfocada o privilegiada por los avances en el desarrollo de la electrónica, los semiconductores, procesos programables, microprocesadores, robótica automatizada, comunicación celular y finalmente procesos de inteligencia artificial para la automatización.

1.6.2 Ventajas

Entre las principales importancia y objetivos que busca desarrollar la automatización están:

- Incrementar la productiva de la empresa que implique reducir los propios costes de los procesos de producción mejorando la calidad de los productos.
- Optimizar las propias condiciones del propio trabajo personal que se ejecuta. Eliminando los procesos tediosos, riesgosos, con incremento de la seguridad laboral.

- Ejecutar nuevas actividades y funciones que no eran posible realizar de forma manual. Las automatizaciones permiten simplificar procesos complejos manuales.
- Aumentar la producción y disponibilidad de cualquier producto o servicio, según las cantidades necesarias determinadas por los objetivos de las empresas.
- Lograr una integración efectiva de los diversos procesos de gestión operacional y producción, creando racionalización de los procesos y los insumos.
- Permite generar un retorno de inversión mucho más rápido y productivos, creando una trazabilidad adecuada que influye en la disminución de posibles errores.
- Contribuye directamente a generar una completa reducción en la simplificación de las actividades, que puedan implicar que un trabajador no necesite o requiera grandes cantidades de conocimientos y capacitación para la manipulación del proceso productivo.
- Contribuir a desarrollar una menor inversión en un sistema automatizado diseñado bajo requerimientos específicos para las empresas, esto puede aumentar la producción y generación de retornos de inversión más elevados.

En algunos casos la importancia de la automatización implica que los sistemas de producción pueden estar completamente diseñados para ejecutar importantes actividades de procesamiento, ensamble, manejo de materiales e inspección, con una reducida o escasa participación de los humanos.

1.7 Servicios Web para Sistemas de Control Automáticos

Los servicios Web utilizados en sistemas de control automático suelen ser parte integral de la arquitectura de estos sistemas para habilitar la comunicación y el intercambio de datos entre los componentes. Algunos de los servicios Web comunes utilizados en sistemas de control automático incluyen:

Servicios Web Rest (Representational State Transfer): Los servicios Web RESTful se basan en los principios de REST, que utilizan operaciones HTTP estándar (GET, POST, PUT, DELETE) para permitir la comunicación entre sistemas. Son ampliamente utilizados en sistemas de control automático debido a su simplicidad y facilidad de implementación. Estos servicios permiten la recuperación y actualización de datos, lo que es esencial en el control y monitoreo de sistemas.

SOAP (Simple Object Access Protocol): SOAP es un protocolo de comunicación basado en XML utilizado para intercambiar información estructurada en la implementación de servicios Web. Aunque SOAP es más pesado que REST, se utiliza en sistemas donde se requiere una comunicación más robusta y controlada, como en sistemas críticos de control industrial.

MQTT (Message Queuing Telemetry Transport): MQTT es un protocolo de mensajería ligero y eficiente que se utiliza comúnmente en sistemas de control automático para la comunicación entre dispositivos y sistemas de supervisión. Es especialmente útil en aplicaciones de IoT (Internet de las Cosas) y sistemas de automatización industrial, donde se necesita una comunicación de bajo ancho de banda y alta eficiencia.

WebSockets: WebSockets es un protocolo que permite la comunicación bidireccional en tiempo real entre un cliente Web y un servidor. Es útil en sistemas de control automático donde se requiere una comunicación continua y en tiempo real, como en sistemas de monitoreo de procesos industriales.

UPnP (Universal Plug and Play): UPnP es un conjunto de protocolos y servicios utilizados para descubrir y controlar dispositivos en una red local. En sistemas de control automático, UPnP puede utilizarse para descubrir dispositivos conectados a la red y facilitar su comunicación.

Servicios Web RESTful personalizados: En algunos casos, se pueden desarrollar servicios Web personalizados utilizando REST u otros protocolos para satisfacer las necesidades específicas de un sistema de control automático. Estos servicios pueden ser diseñados para proporcionar acceso a datos de sensores, actuadores, sistemas de control y bases de datos.

1.8 Bases de Datos

1.8.1 ¿Qué son las Bases de Datos?

Una base de datos es un conjunto organizado de información que se almacena y se gestiona de manera estructurada para facilitar su acceso, búsqueda, recuperación y actualización. Está diseñada para contener datos que se relacionan entre sí de alguna manera y se utilizan en diversas aplicaciones.

1.8.2 Utilidad de las Bases de Datos

Las bases de datos se utilizan para gestionar y almacenar datos de manera eficiente, lo que facilita la recuperación y manipulación de la información. Se utilizan en una variedad de aplicaciones, como sistemas de gestión de inventarios, sistemas de reservas, sistemas de control de inventario, sistemas de información empresarial, sistemas de control de producción, sistemas de seguimiento de clientes y muchas otras aplicaciones.

1.8.3 Tipos de Bases de Datos:

- Bases de Datos Relacionales: Utilizan tablas para organizar los datos y establecer relaciones entre ellos. Ejemplos incluyen MySQL, PostgreSQL, Oracle, y Microsoft SQL Server.
- Bases de Datos NoSQL: Son bases de datos diseñadas para manejar datos no estructurados o semiestructurados. Pueden ser útiles en aplicaciones donde la escalabilidad y la velocidad son críticas. Ejemplos incluyen MongoDB y Cassandra.
- Bases de Datos de Gráficos: Están diseñadas para representar y gestionar datos relacionales de manera eficiente. Son útiles en aplicaciones donde las relaciones son fundamentales, como redes sociales y sistemas de recomendación. Ejemplos incluyen Neo4j y Amazon Neptune.
- Bases de Datos en Memoria: Almacenan datos en la memoria principal de la computadora para un acceso ultrarrápido. Son adecuadas para aplicaciones que requieren una alta velocidad de lectura/escritura, como sistemas de control en tiempo real.
- Bases de Datos de Series Temporales: Se utilizan para almacenar y analizar datos de series temporales, como registros de sensores y datos de monitoreo. Ejemplos incluyen InfluxDB y TimescaleDB.

1.8.4 Ventajas de las Bases de Datos en Sistemas Automatizados:

- Organización de Datos: Las bases de datos ayudan a organizar los datos de manera estructurada, lo que facilita la gestión y el acceso a la información.
- Recuperación Eficiente: Permite recuperar datos de manera rápida y precisa, lo que es esencial en sistemas de control automático donde la velocidad es crítica.

- Seguridad y Control de Acceso: Las bases de datos ofrecen mecanismos para garantizar la seguridad de los datos y controlar quién puede acceder y modificar la información.
- Escalabilidad: Muchas bases de datos son escalables, lo que permite manejar grandes volúmenes de datos a medida que un sistema automatizado crece.
- Consistencia de Datos: Las bases de datos garantizan la integridad de los datos y evitan inconsistencias en la información.
- Soporte para Transacciones: En sistemas automatizados críticos, es fundamental que las transacciones se ejecuten de manera consistente y confiable, y las bases de datos brindan soporte para ello.
- Análisis y Reporting: Las bases de datos permiten realizar análisis de datos y generar informes, lo que es esencial en sistemas automatizados para la toma de decisiones informadas.

Capitulo II: Diseño del Sistema

2.1 Diseño del hardware

Durante el proceso de concepción y diseño del sistema, se seleccionaron meticulosamente los componentes más adecuados para asegurar una integración armónica y eficiente. La elección se basó en criterios rigurosos y consideraciones específicas, con el objetivo de ejecutar el diseño de manera eficiente y efectiva.

Este enfoque ha implicado una cuidadosa evaluación de las capacidades y características de los distintos elementos que componen el sistema. Se ha prestado una atención minuciosa a aspectos como la compatibilidad, la interoperabilidad y la sinergia entre estos componentes, lo que ha permitido una cohesión óptima en el diseño global.

Al considerar con detenimiento la elección de estos elementos, se ha logrado establecer una sólida base sobre la cual se construirá el sistema de manera robusta y confiable. Además, se ha tenido en cuenta la idoneidad de estos componentes para cumplir con los requisitos y objetivos predefinidos del proyecto, garantizando que la integración resultante sea la más adecuada.

2.1.1 Raspberry Pi en el proyecto

En la arquitectura de nuestro proyecto, hemos seleccionado la potencia y adaptabilidad de Raspberry Pi como el núcleo vital de nuestro sistema. La preferencia por Raspberry Pi se basa en su flexibilidad sin paralelo y su amplia gama de aplicaciones. Frente a opciones menos versátiles, los beneficios que aporta esta plataforma son destacables, proporcionando un cimiento robusto para nuestros objetivos y requerimientos específicos.

Para nuestro proyecto, hemos seleccionado el modelo Raspberry Pi 2B, una elección estratégica respaldada por un conjunto de características técnicas sobresalientes. Esta decisión se basa en un análisis detallado de nuestras

necesidades técnicas, financieras y de rendimiento, asegurando así una base sólida y confiable para el desarrollo exitoso de nuestro proyecto. A continuación, se muestran las características más destacables.

- Procesador: Broadcom BCM2836 ARM Cortex-A7 de cuatro núcleos a 900 MHz¹²³.
- RAM: 1 GB SDRAM¹²³.
- Puertos USB: 4 puertos USB 2.0¹².
- Red: Ethernet 10/100 Mbit/s¹².
- Almacenamiento: Tarjeta MicroSD para el Sistema Operativo³.
- GPU: VideoCore IV de doble núcleo con soporte de Open GL ES 2.0, hardware acelerado OpenVG hasta 1080p30 H.264³.
- Tamaño: 85,60 mm x 56,5 mm¹.
- Peso: 45 g¹.

2.1.2 Sensores.

El primer apartado a destacar en el diseño del hardware serán los sensores o técnicamente conocidos como transductores, que son los encargados de transformar una magnitud física en impulsos eléctricos.

En nuestro caso, las magnitudes más importantes a tomar en cuenta son las siguientes:

- Temperatura ambiente
- Humedad relativa del aire
- Humedad relativa bajo tierra
- Luminosidad del ambiente

Los valores a manejar serán determinados por el tipo de cultivo, el clima, el área y otros factores que se tomarán en cuenta para tal calibración.

2.1.2.1 Sensor de luminosidad (LDR)

La luminosidad del ambiente puede ser medida por diferentes medios: resistivos, de electrónica activa o sensores ópticos complejos. Para el desarrollo del proyecto se ha decidido el uso de medios resistivos, lo que se conoce como LDR (Light Dependence Resistor) Resistencia dependiente de la luz. Este elemento cambia su valor de resistividad a medida se le incide una mayor cantidad de lúmenes o luxes. Con la ayuda de un circuito básico de división de voltaje se puede calibrar para obtener los valores máximos y mínimos requeridos. Ver *Figura 1*.

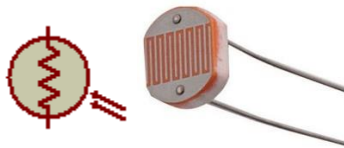


Figura 1. LDR diagrama esquemático y forma física.

De forma general, el circuito que medirá la luminosidad se conforma de dos resistores: un resistor convencional de $1k\Omega$, conectado en serie al LDR de $10k\Omega$, creando así un circuito divisor de voltaje. El valor a medir será el voltaje entregado por la LDR, ya que el valor del primer resistor es fijo. La alimentación del circuito será a 5 voltios de corriente continua, proporcionando un rango de voltaje no muy amplio, pero suficiente para lograr un nivel de precisión adecuado para detectar cambios en el ambiente. Ver *Figura 2*.

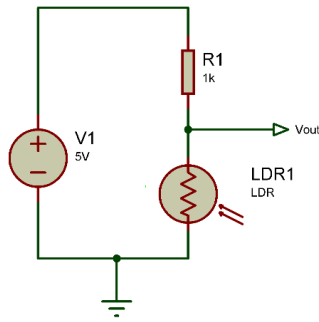


Figura 2. Diagrama esquemático del sensor de luminosidad.

2.1.2.2 Sensor de Temperatura y Humedad Relativa del Ambiente (DHT11)

Otra variable que se toma en cuenta al trabajar con un invernadero son 2 de las variables más importantes a destacar: la temperatura y la humedad. La humedad es relativa, ya que se relaciona con un valor estándar de referencia, el cual se usa para calibrar el sensor capacitivo. La temperatura del ambiente se mide mediante un termistor que se encuentra dentro del encapsulado del sensor. Ambos sensores están conectados a un convertidor ADC en formato ASIC (Application-Specific Integrated Circuit) que cuenta con un transmisor de datos por bus serial.

Algunas de las especificaciones del sensor son:

- Consumo de corriente de 2,5mA
- Señal de salida digital (transmisión serial)
- Rango de temperatura de 0°C a 50°C
- Precisión para medir temperatura a 25°C de unos 2°C de variación
- Resolución de medición de temperatura es de 8-bit, 1°C
- El rango de humedad es desde 20% RH hasta los 90% RH
- con precisión para la humedad del 5% RH para temperaturas que se encuentren entre 0-50°C
- Resolución es de 1% RH

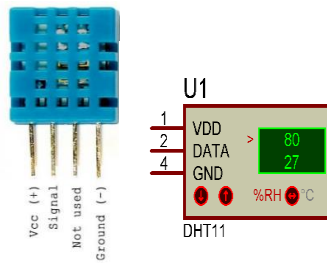


Figura 3. DHT11 Aspecto físico y diagrama esquemático.

Se puede visualizar en la *Figura 3* el modelo utilizado en Proteus y sus semejanzas con el aspecto físico del sensor. Además, la única diferencia entre ambos es la existencia de un pin no utilizado en el modelo físico.

La transmisión de datos captada por el DHT11 se realiza en una trama de 40 bits que corresponden a la información de humedad y temperatura. Los primero dos grupos de 8 bits (16 bits en total) representan la humedad. Los siguientes dos grupos de 8 bits (16 en total)) representan la temperatura. En resumen, se utiliza un byte para la parte entera y otro byte para la parte decimal tanto de la humedad como de la temperatura. Por ejemplo:

0011 0101 0000 0010 0001 1000 0000 0000 0011 1001
 HUMEDAD RELATIVA TEMPERATURA AMBIENTE PARIDAD

En este caso, 0011 0101 0000 0010 es el valor de la humedad, y 0001 1000 0000 0000 para la temperatura. Una primera parte es para la parte entera y la segunda parte es para decimales. En cuanto a 0011 1001, es decir, los últimos 8-bit son de paridad para evitar errores, Corresponde a la suma de los bits anteriores, por tanto, si la suma es igual a la paridad estará correcto. En la hoja técnica que se encuentra en los anexos se explica a más detalle el funcionamiento de la transmisión digital.

2.1.2.3 Sensor humedad en tierra (Capacitive Soil moisture sensor v1.2)

Es primordial que para monitorizar y controlar el sistema de riego es necesario saber además de las variables ambientales, el estado de la tierra que se está usando para el cultivo, de tal manera que para ese fin se necesitara revisar la cantidad de la humedad dentro de la misma, por lo que necesitaremos una interfaz que además de poder medir la humedad relativa como se daba en el sensor anterior, tiene que ser dentro de la tierra misma ya que en la mayoría de las ocasiones, no concordaran entre sí, por lo que se ha optado por utilizar otro sensor de humedad, este será el:

Este sensor funciona mediante un pequeño circuito oscilatorio integrado. Ver *Figura 4*. Utilizando el suelo como capacitor y mide las variaciones de capacitancia, devolviendo una salida analógica en forma de voltaje. El nivel de humedad se refleja en el voltaje de salida: a mayor humedad, menor es el voltaje. Este sensor es ideal para proporcionar datos precisos sobre la humedad en el suelo, esenciales para un control eficiente de riego. Además, el sensor capacitivo es más duradero y resistente al ambiente en comparación con los sensores resistivos lo que hace más adecuado entornos agrícolas y condiciones adversas.

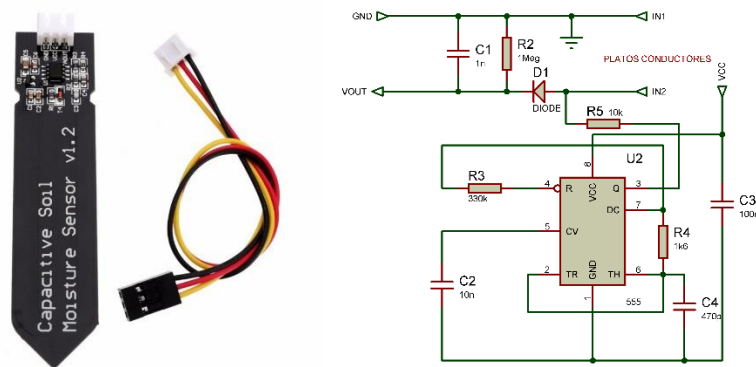


Figura 4. Sensor en su forma física y su circuito esquemático.

En el diagrama esquemático se puede ver un oscilador de frecuencia fija que se construye utilizando un IC temporizador 555 de 8 pines (NE555) un generador de

pulso de onda cuadrada. la señal de onda cuadrada está conectada al sensor que funciona como un condensador sabemos que los condensadores tendrá una reactancia (esta almacena la energía extra no deseada en las placas y la libera cuando se descarga sin disiparse como calor en el caso de la resistencia), por lo que se crea un divisor de voltaje conectando una resistencia de 10k ohmios desde el pin 3, cuanto mayor sea la humedad del suelo en el sensor, la capacitancia es mayor en el sensor, entonces, cuando se observa una capacitancia más alta en el sensor, menor es la reactancia a la onda cuadrada. produciendo un bajo voltaje en la línea de señal.

Especificaciones técnicas del sensor:

- Funciona de 3.3v a 5.5v
- Voltaje analógico de salida: 0 ~ 3.0 VDC
- Soporta interfaz de sensor de 3 pines
- Interfaz: PH2.0-3P

2.1.3 PCF8591 (Conversor ADC y DAC)

El PCF8591 es un circuito integrado que funciona como convertidor ADC y DAC con una interfaz de bus de comunicación con protocolo I2C, el cual se encuentra en serie tanto con el convertidor ADC y el DAC.

Este dispositivo posee un convertidor analógico digital de 8 bits, adecuado para el rango de voltajes utilizado (0-5V). Además, cuenta con una entrada de tierra separada que permite mantener las parte analógicas y digital aisladas, conectándola a la tierra la fuente de alimentación. Esto ayuda a evitar que el ruido digital afecte al circuito analógico, mejorando la precisión al eliminar interferencias.

Se puede visualizar en la *Figura 5* el modelo utilizado en Proteus y sus semejanzas con el aspecto físico del sensor.

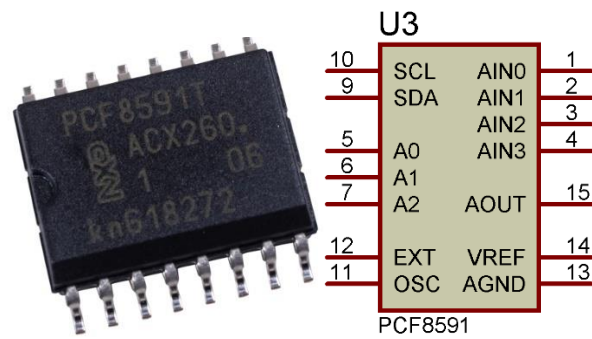


Figura 5. IC PCF8591 en forma física y su diagrama esquemático.

Cabe mencionar que el chip cuenta con una salida DAC a 256 pasos, los mismos que el ADC, el DAC es necesario para realizar analógico a la conversión digital, generando una señal analógica que se compara con la señal de entrada, y aunque el DAC se utiliza para parte del proceso de conversión de ADC, la salida tiene un amplificador “track and hold” que mantiene la salida constante mientras que la salida DAC se cambia a un valor diferente, siendo factible su uso en separado de la entrada ADC. Ver Figura 6.

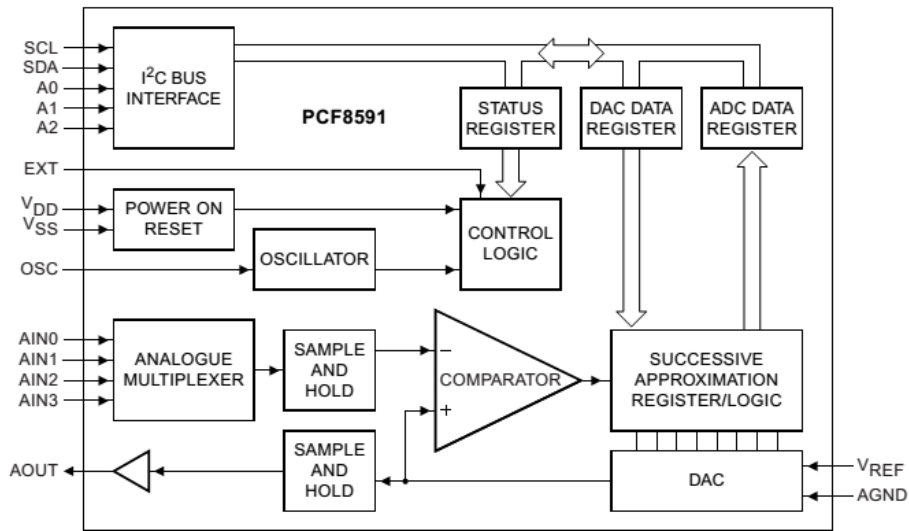


Figura 6. Diagrama de bloques de PCF8591

También es de destacar que por su facilidad de adquirirlo y su precio este es una ADC multiplexado en las entradas, generando que este conversor sea lento, a comparación de otros medios más comunes de conversión, sumándole que para

poder leer los valores de entrada es necesario acceder vía I2C generando más retardo a la hora de su lectura.

Especificaciones del convertidor:

- Alimentación: 2.5V a 6V DC
- Valores máximos de trabajo: -0.5V a 8V DC
- Interfaz: I2C
- Frecuencia de trabajo en I2C: 100kHz
- Resolución: 8bits
- Entradas analógicas: 4 entradas separadas/ 2 entradas diferenciales
- Corriente de alimentación máxima: 250µA a 1mA
- Error de medición ADC: 20 a 50 mV
- Tiempo de conversión: 90µs
- Frecuencia de muestreo: 11.1kHz
- Direcciones I2C: 0x48 a 0x4F
- Temperatura de funcionamiento: -40 a 85 °C

2.1.3.1 Tecnología I2C

I2C (Inter-Integrated Circuit) es un protocolo de comunicación serial que permite la interconexión de múltiples dispositivos utilizando solo dos cables: uno para datos (SDA) y otro para el reloj (SCL). Es comúnmente utilizado en sistemas embebidos y otros dispositivos electrónicos¹. Además, I2C permite conectar hasta 127 dispositivos esclavos en el mismo bus, lo que lo hace ideal para comunicarse con sensores digitales y otros componentes.

2.1.3.1.1 Características

Las características más salientes del bus I2C son

- Requiere solo dos líneas: datos (SDA) y reloj (SCL).
- Cada dispositivo tiene un código de dirección seleccionable mediante software.
- Relación Master/Slave permite entre el microcontrolador y los dispositivos conectados.
- Permite varios masters y detecta colisiones.
- Protocolo de transferencia de datos y direccionamiento definido por software.
- Datos y direcciones transmitidos en palabras de 8 bits.

2.1.3.1.2 Funcionamiento del bus I2C

Las líneas SDA y SCL transportan información entre los dispositivos conectados. Cada dispositivo se reconoce por su dirección y puede operar como transmisor o receptor. Además, pueden actuar como Master o Slave:

- Master: Inicia la transferencia y genera la señal de reloj.
- Slave: Es el dispositivo direccionado.

Las líneas SDA y SCL son bidireccionales y están conectadas al positivo de la alimentación mediante resistencias de pull-up. Cuando el bus está libre, ambas líneas están en nivel alto. La transmisión de datos bidireccional (8 bits) puede realizarse a 100 Kbits/s en modo estándar o 400 Kbits/s en modo rápido. La cantidad de dispositivos conectables al bus está limitada por la capacidad máxima permitida de 400 pF.

2.1.3.1.3 Fundamentos y Operación del Protocolo de Comunicación I2C

Condiciones de START y STOP

- Condición de START: El Master inicia la comunicación haciendo que la línea SDA caiga a cero mientras SCL permanece en alto. Esto marca el comienzo de la transferencia de datos.
- Condición de STOP: Al finalizar la comunicación, el Master eleva la línea SDA a nivel alto mientras SCL permanece en alto, indicando el fin de la transferencia.

Transferencia de Datos

- El Master genera la condición de START.
- Cada palabra en el bus SDA debe tener 8 bits. La primera palabra contiene la dirección del Esclavo.
- El Master lee el estado de la línea SDA. Si es 0 (ACK), la transferencia continúa. Si es 1, el Master genera una condición de STOP.
- Al finalizar la transmisión, el Master genera una condición de STOP y libera el bus, llevando las líneas SDA y SCL a estado alto.

Fallas en el Bus

- Verificar las resistencias de pull-up.
- Usar una punta lógica para comprobar que las líneas SDA y SCL estén en estado alto cuando el bus está inactivo.
- Verificar que el Master direcciona algún integrado usando una punta lógica, aunque esto solo detecta actividad en el bus.
- Para un diagnóstico más detallado, utilizar un osciloscopio digital o construir un probador con componentes básicos.

2.1.3.1.4 Conversión Analógica Digital.

Una de las desventajas de utilizar la plataforma Raspberry Pi es la falta de entradas analógicas integradas. Esto presenta un desafío para aplicaciones que requieren la lectura y manejo de señales analógicas. Para solucionar este problema, es esencial disponer de una interfaz que convierta los valores analógicos en digitales, permitiendo que la Raspberry Pi pueda interpretarlos y procesarlos adecuadamente. Esta necesidad nos lleva a abordar el proceso de conversión de valores analógicos a digitales.

2.1.4 Salida de Control de Riego

Dentro del sistema de control automático, como se ha revisado en apartados anteriores, los dispositivos actuadores o de salida se describen a continuación.

2.1.4.1 Optoacoplador (6N136)

En la etapa final del sistema de riego, es necesario conectar las salidas digitales de la SBC (Single Board Computer) con el manejo electromecánico. Para lograr esta conexión, se utiliza un optoacoplador, específicamente el modelo 6N136. Este componente consta de dos partes diferenciadas: un Led y un transistor.

El Led, de bajo consumo, recibe la entrada digital. El transistor, con un arreglo de alta velocidad, garantiza una respuesta rápida y precisa, lo que es crucial para la elección de este modelo en particular. Además, el optoacoplador permite que la salida de potencia sea alimentada con un voltaje mayor al manejado por la Raspberry Pi (3.3V), en este caso de 5 a 24 V. Esto asegura que el sistema de riego, incluyendo las electroválvulas, se active y desactive sin demoras, manteniendo una alta fidelidad en la respuesta del sistema.

Se puede visualizar en la *Figura 7* el modelo utilizado en Proteus y sus semejanzas con el aspecto físico del sensor.

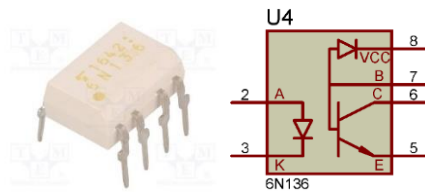


Figura 7. Forma física de 6N136 y diagrama esquemático

2.1.4.2 Relé

Por último, para el manejo de electroválvulas y motores (bombas) se utilizará un viejo conocido el relé, un sistema de switch electromecánico activado por medio de pulso eléctrico, este cuenta con contactos que están destinados a ser utilizado tanto en corriente directa o corriente alterna según sea la necesidad, por lo que lo hace el medio idóneo para trabajar con este tipo de carga de potencia.

Se puede visualizar en la *Figura 8* el modelo utilizado en Proteus y sus semejanzas con el aspecto físico del sensor.

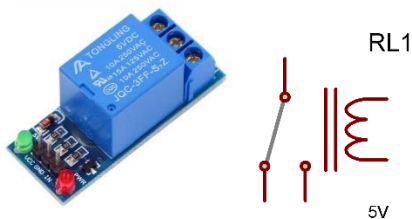


Figura 8. Forma física y diagrama esquemático del relé de 5 voltios.

2.1.4.3 Electroválvulas y Sistema de Bombeo

Una electroválvula es un dispositivo electromecánico diseñado para controlar el flujo de líquidos o gases en un sistema. Consiste en un cuerpo principal que alberga un mecanismo de válvula y una bobina electromagnética. Cuando se aplica una

corriente eléctrica a la bobina, esta genera un campo magnético que activa el mecanismo de la válvula, permitiendo o bloqueando el paso del fluido a través de la válvula.

Para el control del sistema de riego, se tomó la decisión de implementar una electroválvula con especificaciones de 2 watts de potencia y operación a 110 voltios. Esta elección se basa en la necesidad de un dispositivo eficiente energéticamente y compatible con la red eléctrica estándar. La electroválvula seleccionada ofrecerá un control preciso del flujo de agua en el sistema de riego, asegurando una distribución óptima del recurso hídrico en función de las necesidades de cada área de cultivo. Se puede ver una muestra de su forma física en la *Figura 9*.



Figura 9. Electroválvula.

2.2 Software

2.2.1 Estructura de Servicios

A pesar de que Raspberry Pi es un ordenador con muchas características, sus capacidades no son suficientes para ejecutar todo tipo de servidores mediante máquinas virtuales. Sin embargo, el uso de contenedores nos brinda un ahorro significativo de memoria y procesos dentro de la Raspberry Pi. Es por eso que se ha adoptado este tipo de estructura en el dispositivo.

2.2.2 Docker

Docker es un proyecto de código abierto que automatiza la implementación de aplicaciones como contenedores portátiles y autosuficientes, ejecutables tanto en la nube como localmente. Docker, la empresa detrás de esta tecnología, colabora con proveedores de la nube, Linux y Windows, incluyendo a Microsoft.

Los contenedores de Docker pueden ejecutarse en cualquier entorno: localmente en el centro de datos del cliente, en un proveedor de servicios externo o en la nube, como en Azure. Las imágenes de Docker son nativas de Linux y Windows; sin embargo, las imágenes de Windows solo pueden ejecutarse en hosts de Windows, mientras que las imágenes de Linux pueden ejecutarse en hosts de Linux y en hosts de Windows mediante una máquina virtual Linux de Hyper-V.

Los desarrolladores pueden usar entornos de desarrollo en Windows, Linux o macOS. En su equipo de desarrollo, el desarrollador ejecuta un host de Docker donde se implementan imágenes de Docker, incluyendo la aplicación y sus dependencias. Los desarrolladores en Linux o macOS usan un host de Docker basado en Linux y pueden crear imágenes solo para contenedores de Linux. Los desarrolladores en Windows pueden crear imágenes para contenedores de Windows o Linux.

Para hospedar contenedores en entornos de desarrollo y proporcionar herramientas adicionales, Docker ofrece Docker Desktop para Windows y macOS. Estos productos instalan la máquina virtual necesaria (el host de Docker) para hospedar los contenedores.

2.2.2.1 Tiempos de Ejecución

Para ejecutar contenedores de Windows, hay dos tipos de tiempos de ejecución:

- Los contenedores de Windows Server ofrecen aislamiento de aplicaciones a través de tecnología de aislamiento de proceso y de espacio de nombres. Un

contenedor de Windows Server comparte el kernel con el host de contenedor y con todos los contenedores que se ejecutan en el host.

- Los contenedores de Hyper-V amplían el aislamiento que ofrecen los contenedores de Windows Server mediante la ejecución de cada contenedor en una máquina virtual altamente optimizada. En esta configuración, el kernel del host del contenedor no se comparte con los contenedores de Hyper-V, lo que proporciona un mejor aislamiento.

Las imágenes de estos contenedores se crean y funcionan de la misma manera. La diferencia radica en cómo se crea el contenedor desde la imagen ejecutando un contenedor de Hyper-V que requiere un parámetro adicional. Para más información, vea Contenedores de Hyper-V.

2.2.2.2 Comparación de los Contenedores de Docker con las Máquinas Virtuales

Para las máquinas virtuales, hay tres niveles de base en el servidor host, de manera ascendente: infraestructura, sistema operativo host y un hipervisor y, encima de todo eso, cada máquina virtual tiene su propio sistema operativo y todas las bibliotecas necesarias. En el caso de Docker, el servidor host solo tiene la infraestructura y el sistema operativo y, encima de eso, el motor de contenedor, que mantiene el contenedor aislado, pero con el uso compartido de los servicios del sistema operativo de base.

Dado que los contenedores requieren muchos menos recursos (por ejemplo, no necesitan un sistema operativo completo), se inician rápidamente y son fáciles de implementar. Esto permite tener una mayor densidad, lo que significa que se pueden ejecutar más servicios en la misma unidad de hardware, reduciendo así los costos.

Como efecto secundario de la ejecución en el mismo kernel, obtiene menos aislamiento que las máquinas virtuales.

El objetivo principal de una imagen es que hace que el entorno (dependencias) sea el mismo entre las distintas implementaciones. Esto significa que puede depurarlo en su equipo y, a continuación, implementarlo en otra máquina con el mismo entorno garantizado.

Una imagen de contenedor es una manera de empaquetar una aplicación o un servicio e implementarlo de forma confiable y reproducible. Podría decir que Docker no solo es una tecnología, sino también una filosofía y un proceso.

El uso de Docker elimina la frase común entre los desarrolladores: "Si funciona en mi máquina, ¿por qué no en producción?". Con Docker, se puede decir simplemente "Se ejecuta en Docker", porque la aplicación de Docker empaquetada puede ejecutarse en cualquier entorno de Docker compatible, y se ejecuta de la forma prevista en todos los destinos de implementación (como desarrollo, control de calidad, ensayo y producción).

2.2.2.3 Docker dentro del Proyecto

El proyecto implementó Docker debido a las ventajas que ofrece, especialmente en un entorno con recursos limitados como Debian en la Raspberry Pi. Los contenedores de Docker permiten un uso eficiente de los recursos, optimizando el sistema.

Docker, como se mencionó anteriormente, facilita la rápida implementación de aplicaciones. En este proyecto, forma parte del sistema central, ya que los contenedores de Docker permiten utilizar los servicios necesarios sin necesidad de virtualización de sistemas externos. Docker es una herramienta popular para optimizar el uso de recursos.

Dentro de Docker se encuentran tres servicios importantes del proyecto: MariaDB, NodeJS y Nginx.

2.2.2.3.1 Imágenes en los Contenedores

Durante el diseño del sistema, se determinó qué servicios se ejecutarían en contenedores y cuáles se ejecutarían de forma nativa en la Raspberry Pi. Se decidió utilizar tres imágenes dentro del proyecto: MariaDB, NodeJS y Nginx, que corresponden al servicio de base de datos, el Back End y el Front End, respectivamente.

2.2.2.3.1.1 MariaDB

MariaDB es una base de datos popular entre los usuarios de Linux. Dado que el sistema operativo de la Raspberry Pi se basa en un núcleo Linux, MariaDB se convirtió en una opción ideal para el proyecto.

2.2.2.3.1.2 Node

NodeJS fue la opción elegida, seguido posteriormente por Nginx. Esta estructura basada en JavaScript se seleccionó debido a su amplia comunidad y su uso prevalente en la Web contemporánea. La amplia comunidad de NodeJS facilita la búsqueda de ayuda y la resolución de problemas, además de minimizar la probabilidad de incompatibilidad con el hardware utilizado.

2.2.2.3.1.3 Nginx

Nginx es una herramienta de servicio Web, estando a un nivel de como estuvo Apache en su momento, la implementación de códigos CSS, Java y otros dentro de Nginx y amplia comunidad fueron los motivos al igual que Node para elegir esta plataforma

2.3 Estructura Final del Proyecto

La estructura de bloques del sistema se presenta en la Figura 10 y se retomará posteriormente en la implementación de cada módulo del sistema de invernadero.

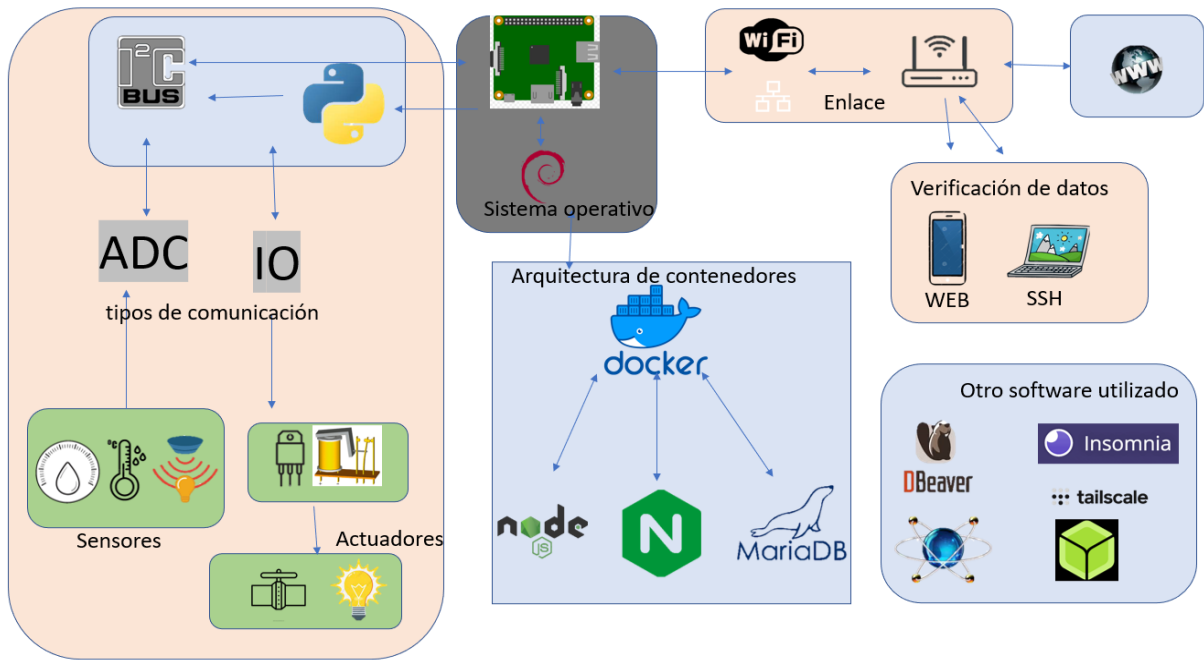


Figura 10. Diagrama de bloques del sistema de proyecto.

2.4 Diseño del Prototipo

Se eligió AutoCAD, un software de diseño asistido por computadora (CAD), para crear el modelo del prototipo. Este software fue fundamental en la fase de diseño. Con su interfaz intuitiva y sus potentes funciones de dibujo, se pudo diseñar los diferentes perfiles del prototipo. (Ver Figura 11).



Figura 11. Capturas del diseño de prototipo en AutoCAD.

Además de AutoCAD, se utilizó SketchUp, una herramienta de modelado 3D conocida por su facilidad de uso y su capacidad para crear diseños detallados rápidamente. SketchUp fue especialmente útil para generar representaciones visuales del prototipo en las primeras etapas del diseño, lo que permitió visualizar cómo se vería el prototipo antes de su construcción física. (Ver Figura 12, captura del prototipo en 3D representado en SketchUp).

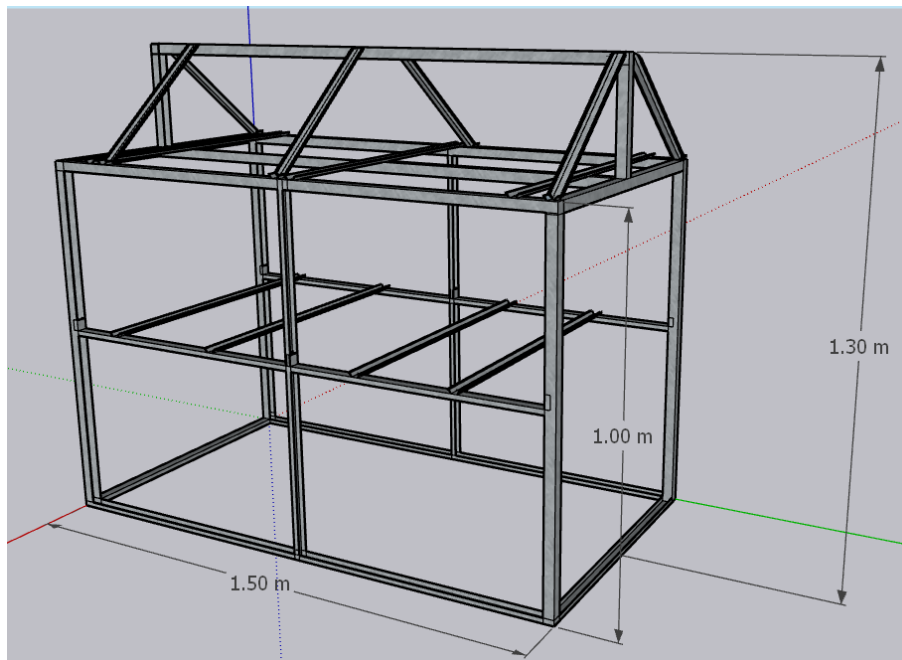


Figura 12. Diseño del prototipo en SketchUp.

Capitulo III: Implementación

3.1 Explicación física del Prototipo

En los siguientes apartados se describirá el proceso de construcción y armado del prototipo.

3.1.1 Proceso de armado del Prototipo

En el proceso de construcción del prototipo, se exploraron diversas opciones de materiales con el objetivo de encontrar la solución más idónea para las necesidades. Tras realizar un análisis, se llegó a la conclusión de que las estructuras metálicas de tabla roca son la elección más adecuada para el proyecto. Este material proporciona la combinación perfecta de ligereza y resistencia, lo que resulta esencial para el diseño y la funcionalidad del prototipo. Con las estructuras metálicas de tabla roca, se puede asegurar tanto la durabilidad como la flexibilidad necesaria para cumplir con los requerimientos del proyecto de manera óptima. Como se puede apreciar en las siguientes capturas, se muestra el proceso de armado de la estructura.

Con base en los diseños realizados en AutoCAD y SketchUp, se cortaron las piezas metálicas para comenzar el ensamblaje del prototipo. (Ver Figuras 13 y 14). Las siguientes imágenes muestran algunos de los perfiles utilizados, según los requerimientos establecidos en los diseños preliminares.



Figura 13. Postes de tabla roca para la construcción del prototipo.



Figura 14. Perfil de tabla roca para la construcción del prototipo.

Con el propósito de resguardar y almacenar los cultivos, se ha establecido un depósito que no solo facilitará el tratamiento de los mismos, sino que también permitirá el control preciso de las variables que inciden en su calidad y desarrollo. (Ver Figura 15).



Figura 15. Canales de PVC para los cultivos.

Una vez que se han delineado las distintas partes y elementos de los perfiles de tabla roca conforme al diseño preliminar en AutoCAD, el siguiente paso consiste en llevar a cabo el ensamblaje del prototipo. (Ver Figura 16).



Figura 16. Primeras fases de construcción del prototipo.

Después de completar una parte del ensamblaje inicial del prototipo, avanzamos con la integración de sus distintos elementos. Esto incluye el sistema de riego, compuesto por una electroválvula y su conjunto de mangueras que suministran agua al sistema. Asimismo, se incorporaron una serie de luminarias incandescentes para proporcionar luz al entorno y, en caso necesario, calor. También se instaló un extractor para regular la temperatura. Además, se integraron varios sensores para monitorear los parámetros ambientales del entorno, incluyendo un sensor de temperatura y humedad DHT11, un sensor de luminosidad LDR y un sensor capacitivo para medir la humedad en la tierra. (Ver Figura 17).



Figura 17. Integración elementos eléctricos y electrónicos al prototipo.

Una vez que todos los elementos fueron integrados al prototipo, se procedió al proceso final de construcción del mismo. (Ver Figura 18).



Figura 18. Fase final de la construcción del prototipo.

También es importante destacar que, para la protección del cultivo, se utilizó una cubierta plástica fabricada con un material especial tratado con filtro anti rayos ultravioleta. Esta medida se implementó con el fin de garantizar resultados óptimos y proporcionar una protección adecuada para nuestro diseño. En la siguiente imagen se presenta una fotografía del sistema con una vista interna, donde se puede observar cómo la capa protectora envuelve el prototipo. Además, se aprecian los cultivos, específicamente tomates, que ya han comenzado a crecer dentro del sistema. Está cubierta especial no solo protege el interior del prototipo, sino que también contribuye al desarrollo óptimo de los cultivos. (Ver Figura 19).



Figura 19. Muestra del interior del prototipo armado.

Después de integrar tanto la parte física, que incluye la estructura completa del prototipo con todos sus componentes, como la incorporación del hardware para el control y la administración, así como la integración de todo con el software, procedimos al proceso de control y gestión a través de una aplicación Web. Esta aplicación sirve tanto para el monitoreo como para el control de los diferentes sistemas que componen el prototipo. En ella, es posible visualizar los parámetros ambientales o variables como la temperatura y humedad las cuales son medidas por los sensores y enviadas y guardadas en la base de datos y finalmente

presentadas en la Web App, también en esta se puede hacer lo que es el control del riego, las luminarias, y el extractor de aire.

3.2 Hardware

Explicados en el capítulo, ahora se presentará como se explicará la interacción entre ellos y el centro del sistema que es la Raspberry.

3.2.1 Diagrama de Conexión

La Figura 20 muestra un diagrama de las conexiones físicas que componen el hardware de control y monitoreo.

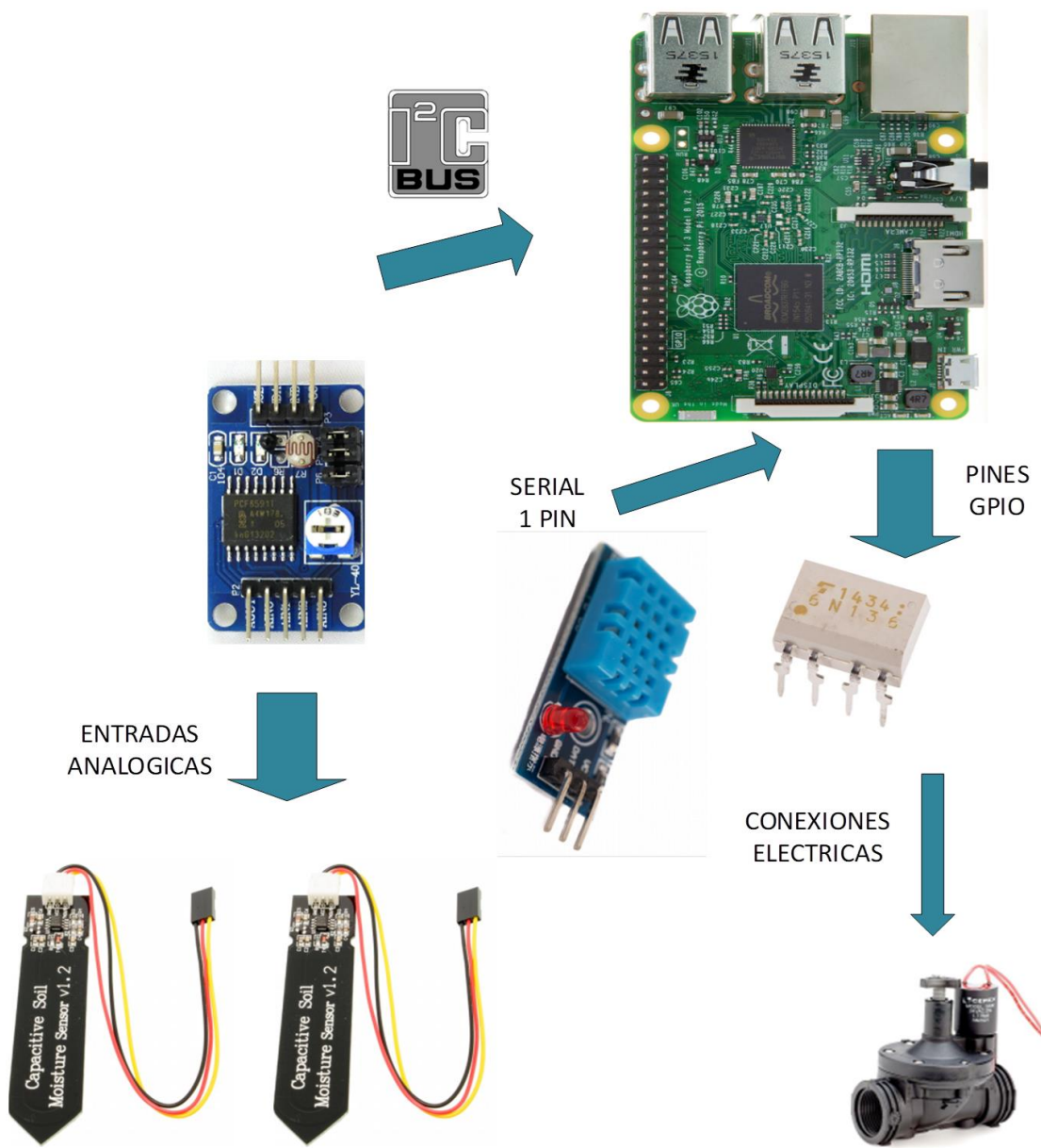


Figura 20 . Diagrama de Conexión.

3.2.2 Modulo PCF8591

El PCF8591, comercialmente no está disponible solo como IC, ya que está pensado para la parte didáctica, por lo que este viene en un módulo integrado con unos elementos extras como se puede apreciar en el circuito esquemático. Ver *Figura 21*.

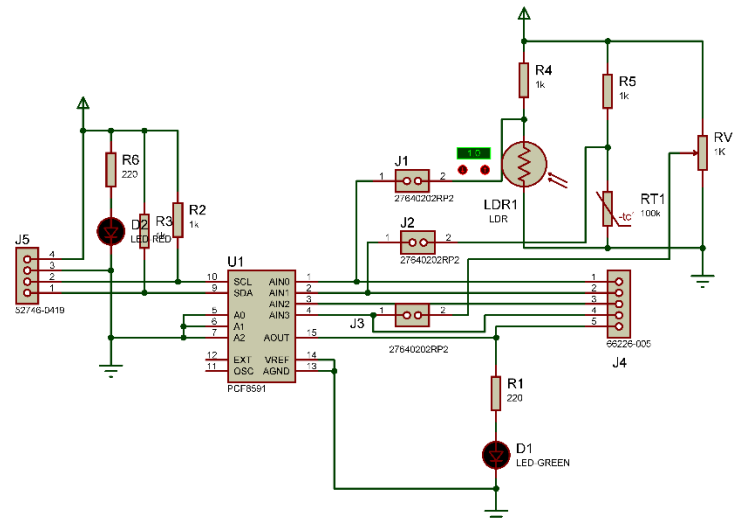


Figura 21. Diagrama esquemático del módulo PCF8591.

Como se puede observar que el módulo cuenta con las resistencias PULL-UP necesarias para evitar problemas de ruido en el canal I²C, además, se encuentra un potenciómetro, una LDR y un NTC estos conectados a las entradas analógicas, 0 1 y 3 pero conectadas a unos jumpers que se pueden desconectar y utilizar las entradas analógicas de manera externa o con estos elementos para calibración o uso.

Para nuestro caso serán utilizadas la entrada AN0 con su LDR como sensor de luminosidad, AN3 y AN4, se utilizar para la lectura de los sensores de humedad en tierra.

3.2.3 Modulo DHT11

El módulo no es más que la entrada de alimentación del sensor, donde se descarta uno de los pines que en realidad no funciona para nada, este sensor se conectara a unos de los GPIO de la Raspberry, específicamente en el pin 4.

3.2.4 Sistema de Control

El sistema de control es la combinación del uso del optoacoplador junto un circuito de relé con transistor conectado a la base de la salida del optoacoplador 6N136, el transistor de acople puede ser cualquiera NPN, en este caso será el TIP35, por si es necesario usar un voltaje mayor a los 5V entregados por la Raspberry.

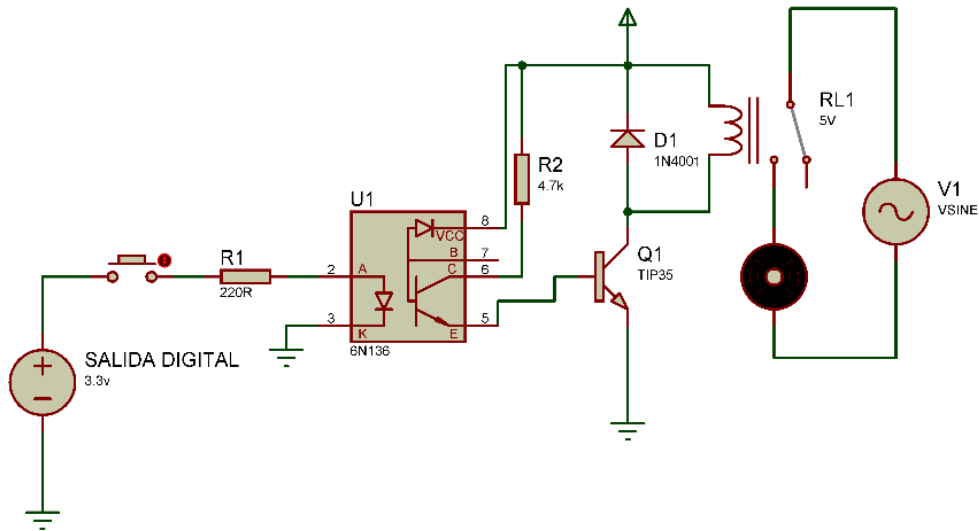


Figura 22. Diagrama esquemático control de electroválvulas.

Se puede observar en la Figura 22 el esquema de conexiones de dicho circuito.

3.2.5 Diagrama Esquemático Completo

En la Figura 23 del diagrama esquemático, se detallan las conexiones de todos los elementos del sistema de control y monitoreo. Los sensores digitales DTH11 se conectan directamente a los pines GPIO de la Raspberry Pi mediante su pin de señal, independientemente del número del puerto GPIO utilizado. Además, se selecciona otro pin GPIO para controlar el transistor que gestiona las válvulas.

Además, los pines GPIO 2 y 3 están asignados al bus I2C, el cual se conecta a los pines correspondientes del IC PCF9581. A su vez, los pines analógicos de este IC se conectan a las salidas de los sensores de humedad en el suelo.

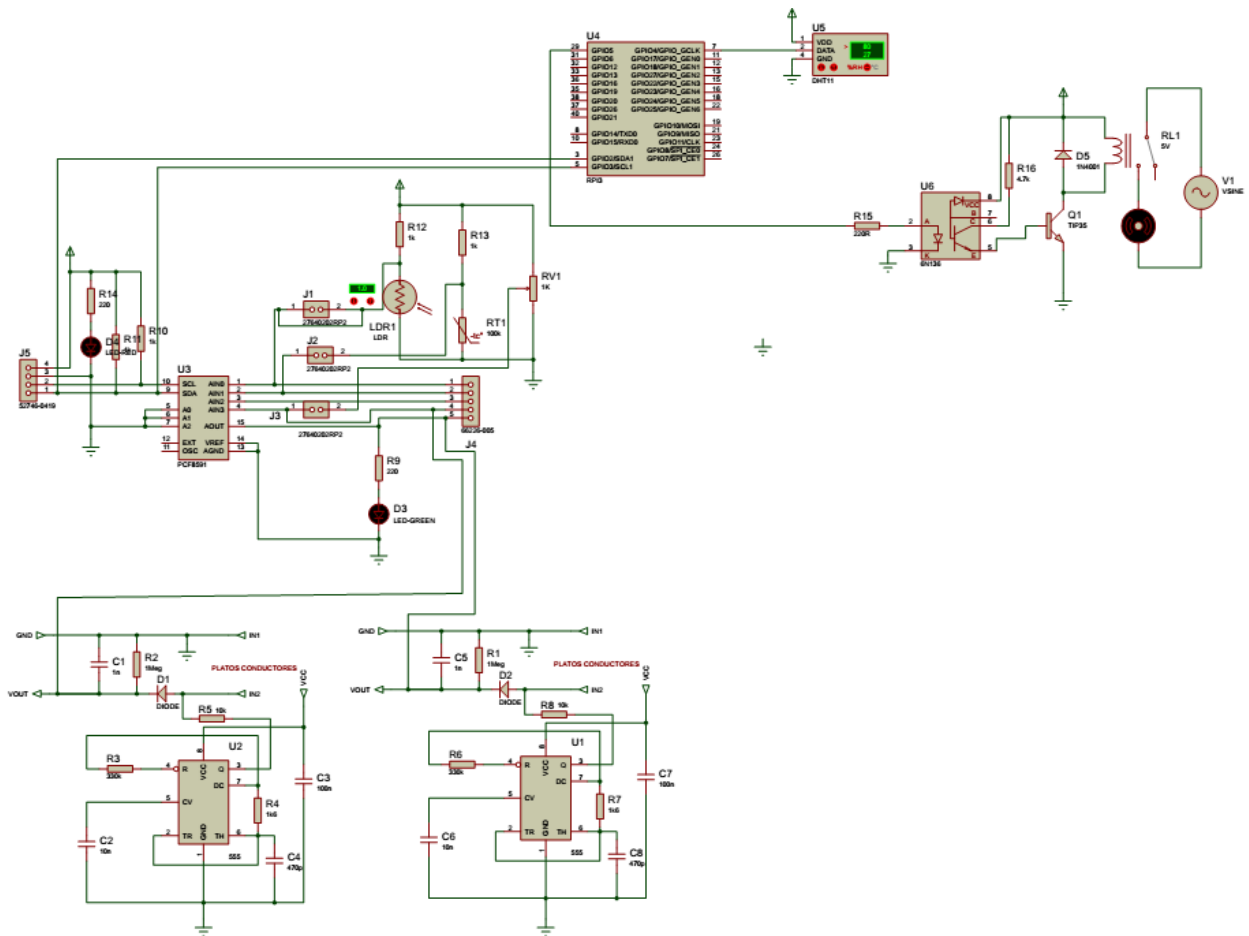


Figura 23. Diagrama esquemático completo del proyecto.

3.3 Software

El software utilizado en el proyecto se puede clasificar en dos categorías: el software utilizado indirectamente y el software utilizado directamente para el desarrollo de la Web App. La primera categoría incluye herramientas y aplicaciones que facilitan la implementación general del proyecto. La segunda categoría se refiere a las herramientas específicas empleadas para el desarrollo de la aplicación web.

3.3.1 Estructura general del software utilizado y su configuración para el proyecto

La placa programable Raspberry Pi funciona como un miniordenador, utilizando un sistema operativo al igual que una computadora. Esta pequeña computadora puede ejecutar diversos scripts, ejecutar software, utilizarse como servidor de diversos servicios y además utilizar cada una de sus entradas GPIO como entradas analógicas y digitales (según sea el caso) para la lectura y escritura de información. Esta última característica fue el motivo principal para su uso en el presente proyecto.

El sistema utilizado por parte de la placa programable es Raspbian, un sistema operativo con Kernel Linux basado en Debian que tiene como una de sus funciones presentar la información capturada por sus puertos GPIO. Debido a las limitantes en hardware de la tarjeta programable Raspberry fue necesario utilizar un sistema operativo sin interfaz gráfica.

Teniendo la tarjeta Raspberry con su respectivo sistema operativo instalado y con su interfaz de WI-FI USB conectado se procedió a realizar las primeras configuraciones del dispositivo a Red, realizando actualizaciones a repositorios y actualizaciones al sistema operativo de la Raspberry. Se realizó la instalación del software que facilitaría la conexión remota vía SSH, siendo este TailScale.

El software de TailScale es de suma importancia dentro del proyecto, ya que este facilita la creación de redes virtuales privadas y permite la conexión remota a la tarjeta programable Raspberry. Estas conexiones se utilizaron no solo para la implementación final del software, sino que también se utilizó para el desarrollo y prueba dentro del proyecto. Las conexiones que se realizaron hacia la tarjeta programable Raspberry se realizaron por el protocolo SSH o Secure Shell, conectándose dentro de la red local o por medio de la VPN.

Una vez establecido el enlace de comunicación con el dispositivo, se instaló el software Docker para crear cada uno de los contenedores necesarios, a saber, el servidor web, el servidor de base de datos y el servidor de la aplicación web.

De los 3 contenedores utilizados, únicamente el contenedor de base de datos se trabajó directamente en el dispositivo. Se realizó la creación de base de datos que se encargarían del almacenamiento de la información y de las ordenes de control. Estas bases de datos eran consultadas para su muestra en la aplicación Web.

Los contenedores Nginx y NodeJS se desarrollaron de manera independiente, utilizando un equipo dedicado. Los códigos fuente fueron compilados y alojados en repositorios de GitHub. El desarrollo se llevó a cabo con la herramienta Microsoft Visual Studio. La documentación detallada del código fuente de la aplicación web se encuentra en los anexos.

Estos contenedores cumplían la función de servicio web, proporcionando la interfaz de usuario de la aplicación. La aplicación presentaba los datos almacenados en la base de datos y registraba el estado de los dispositivos del invernadero, incluyendo la electroválvula y el sistema de iluminación.

Una vez finalizada la configuración del software, se procedió a la implementación. Para ello, se realizaron pruebas de lectura de cada uno de los sensores GPIO, verificando su funcionamiento como entrada o salida. Se desarrollaron scripts en lenguaje Python para validar la correcta lectura de los sensores, considerando que la comunicación entre los sensores y la placa programable podía variar según el tipo de sensor. Estos scripts se ejecutaban en la placa Raspberry Pi, utilizando las librerías necesarias.

Como herramienta adicional, se utilizó el programa DBeaver para la gestión de bases de datos. Si bien no forma parte del sistema principal, DBeaver fue fundamental para el desarrollo del proyecto. Con DBeaver, se verificó la estructura de las bases de datos y las tablas, asegurando que se mantuvieran como se

esperaba. Posteriormente, se confirmó que los datos se almacenaban correctamente en la tabla designada.

Para la fase de pruebas, se utilizó el software *Insomnia*. Esta herramienta permitió verificar, durante el desarrollo y las pruebas de la aplicación web, el envío y recepción de solicitudes, así como la validación de los datos según el puerto de trabajo. Esto facilitó el establecimiento de la comunicación entre la interfaz visual de la aplicación web y la parte de procesamiento y gestión de datos. La Figura 24 muestra un resumen de este diagrama de comunicación.

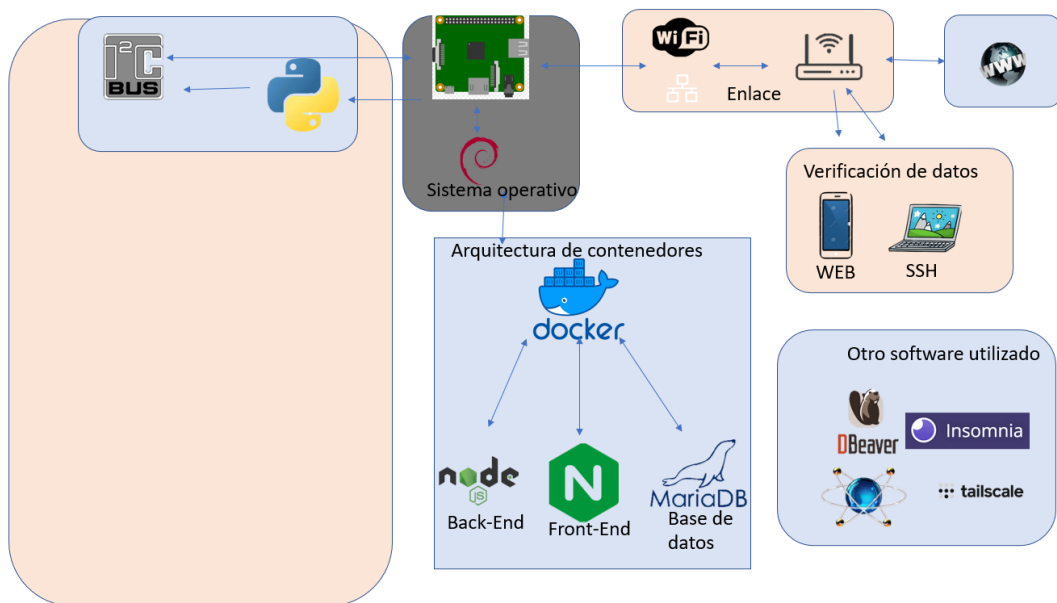


Figura 24. Diagrama de bloques de la estructura general del software.

3.3.2 Estructura y funcionamiento de Web App

El diagrama básico del funcionamiento de la Web App se puede ver en la *Figura 25*.

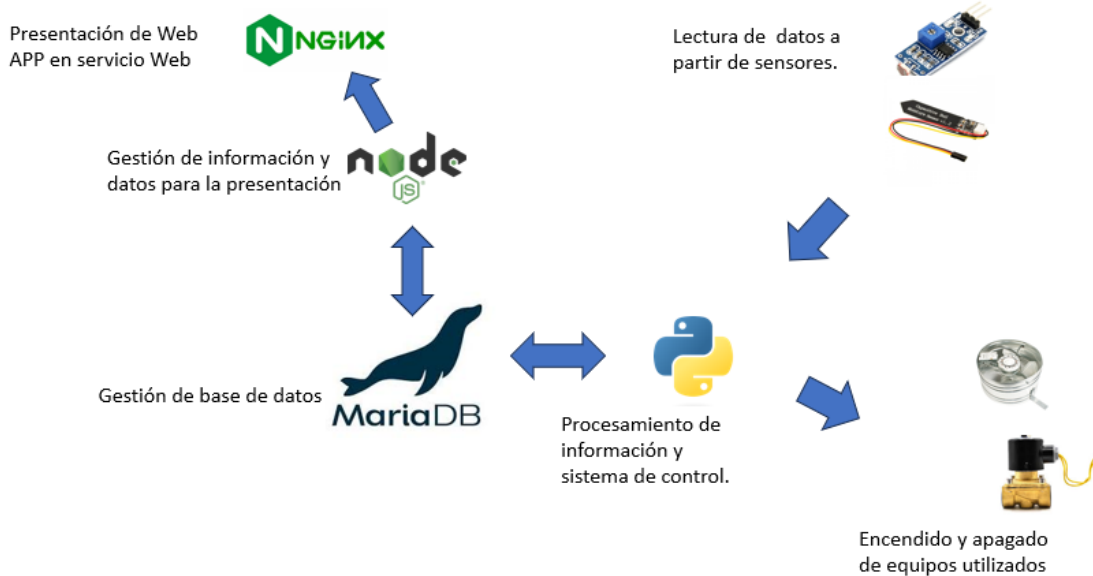


Figura 25. Diagrama de la estructura de la Web APP.

La aplicación desarrollada integra un sistema web de captura y visualización de datos, junto con el control de encendido y apagado de diversos dispositivos. La importancia radica en los datos almacenados, ya que constituyen el núcleo de la funcionalidad del sistema.

Como se mencionó previamente, las tablas de la base de datos almacenan información que es leída y escrita por scripts en lenguaje Python. Estos scripts incluyen pre-sets configurables desde el servidor web, permitiendo una rápida configuración de encendido y apagado del sistema. La captura de datos puede ser ampliada en futuros desarrollos, incluyendo la selección y procesamiento de datos para un sistema con mayor complejidad. Esta información podría utilizarse para optimizar el comportamiento del cultivo.

El script tiene la capacidad de capturar datos en cualquier momento que el usuario lo requiera. Sin embargo, es crucial considerar que una captura de datos con retraso puede resultar en acciones tardías, generando posibles pérdidas. Por ejemplo, un retraso en la captura de datos podría ocasionar un desperdicio de agua o un sobrecalentamiento debido a la activación o desactivación tardía de los dispositivos de control ambiental.

La información almacenada en la base de datos se lee en tiempo real desde la aplicación web. El back-end de la aplicación gestiona la información y la presenta al usuario, completando así la funcionalidad de la aplicación.

Recomendaciones

- Debido a su diseño flexible y adaptable, este sistema puede ser utilizado en diversas áreas. Sin embargo, se recomienda especialmente implementarlo en grandes invernaderos, donde su capacidad de automatización puede ser aprovechada al máximo al integrar otros sistemas que permitan un control más preciso del ambiente de los cultivos.
- Para el desarrollo de proyectos similares, es recomendable mantener el uso de estándares industriales en cada fase del desarrollo. Esto asegura la creación de aplicaciones y utilidades con herramientas de modificación universal. Por ejemplo, utilizar software y lenguajes de programación ampliamente adoptados es crucial, ya que el uso de tecnologías menos comunes podría dificultar el desarrollo y la ejecución debido a la falta de precedentes sobre su uso.
- Es fundamental destacar el uso de Docker en la implementación de sistemas como el presentado, debido a sus ventajas significativas. Docker permite la creación de contenedores, entornos aislados donde se ejecutan las aplicaciones, lo que ayuda a prevenir fallos que puedan afectar al sistema en su totalidad. Además, facilita la realización de respaldos de estos contenedores. En caso de fallo del software principal, estos respaldos pueden restaurar rápidamente el servicio, minimizando el tiempo de inactividad y el impacto en los usuarios finales.

Conclusiones

- El uso de sistemas programables, dispositivos IoT y otros tipos de dispositivos de control puede beneficiar enormemente la producción agrícola. Estos sistemas permiten un control más eficiente de los recursos y del estado de los cultivos, lo que no solo incrementa la cantidad de productos cultivados, sino que también mejora su calidad.
- El uso de la placa programable Raspberry Pi para la automatización de los sistemas de riego presenta una gran oportunidad y ofrece múltiples opciones. Esto se debe a que es un ordenador con bajo consumo de energía, capacidad de conexión a la Web y la posibilidad de establecer puntos de entrada y salida para la captura de datos y la implementación de sistemas de control.
- Para monitorear de manera eficiente los cultivos, se seleccionó diversos tipos de sensores que nos permiten verificar los parámetros ambientales clave, como temperatura, humedad ambiental, humedad del suelo y luminosidad. Sin embargo, era imprescindible establecer una interfaz entre los dispositivos y el operador. Para lograr esto, se desarrolló una Web App que gestiona tanto los dispositivos utilizados como los recursos esenciales para el crecimiento de las plantas.
- El uso de una interfaz amigable es crucial para establecer un sistema de control efectivo. Por lo tanto, se optó por crear un sistema intuitivo y fácil de manejar, accesible a través de dispositivos cotidianos como ordenadores y celulares. Se implementó un diseño de Web App que cumple con los estándares de interfaz aceptados. Esta plataforma permite a los usuarios verificar datos almacenados por los sensores y controlar los módulos dentro del invernadero de manera sencilla e interactiva.

- El uso de bases de datos para la gestión de información ha sido fundamental en el invernadero, permitiendo registrar cada cambio en la ambientación. Esto facilita la identificación de configuraciones que podrían afectar el crecimiento de los cultivos, ya sea mediante scripts automatizados o configuraciones manuales a través de nuestra aplicación Web.

Bibliografía

El taller del huerto. (2020). Sensor de humedad capacitivo vs resistivo. <https://eltallerdelhuerto.com/sensor-capacitivo-vs-resistivo> (Consultado el 30 de abril de 2023).

Hostingplus. (2020). Qué es MariaDB y cuáles son sus características. <https://www.hostingplus.com.es/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/> (Consultado el 24 de marzo de 2023).

INFOAGRO. (2018). Climatización de invernaderos. <https://mexico.infoagro.com/climatizacion-de-invernaderos/> (Consultado el 22 de marzo de 2023).

INFOAGRO. (2022). Control climático en invernaderos. https://www.infoagro.com/industria_auxiliar/control_climatico.htm (Consultado el 24 de marzo de 2023).

Kinsta. (2022). ¿Qué es Nginx y cómo funciona?. <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/> (Consultado el 30 de marzo de 2023).

MariaDB Foundation. (2009). <https://mariadb.org/es/> (Consultado el 30 de marzo de 2023).

Mouser Electronics. (s.f.). DHT11 humidity & temperature sensor. <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf> (Consultado el 5 de mayo de 2023).

Nginx. (s.f.). <https://nginx.org/en/> (Consultado el 8 de mayo de 2023).

NodeJS. (s.f.). <https://nodejs.org/en> (Consultado el 8 de mayo de 2023).

PROBOTS. (s.f.). How to use soil moisture sensor capacitive V2.0 module with Arduino. <https://tutorials.probots.co.in/using-soil-moisture-sensor-capacitive-v2-0-module-fo> (Consultado el 5 de mayo de 2023).

Raspberry Pi 2 Modelo B. (s.f.). <https://electronicamade.com/raspberry-pi-2-modelo-b/> (Consultado el 28 de febrero de 2023).

Soloelectronicos. (2018). Control de GPIO de Raspberry Pi sobre MQTT usando ThingsBoard. <https://soloelectronicos.com/tag/automatizacion-con-raspberry-pi/> (Consultado el 28 de febrero de 2023).

TronicsBench. (s.f.). PCF8591. <https://www.best-microcontroller-projects.com/pcf8591.html> (Consultado el 5 de mayo de 2023).

Uelectronics. (2011). RAS series. <https://uelectronics.com/wp-content/uploads/2017/06/AR0037-Modulo-Relevador-5V-KY-019-Datasheet.pdf> (Consultado el 15 de agosto de 2023).

Electroson. (s.f.). High speed optocoupler, 1 MBd, photodiode with transistor output. https://www.electroson.com/documentos/D_6N136_0_6N136_OPTOACOPLADOR_DOCUMENTACION.pdf (Consultado el 10 de septiembre de 2023).

Anexos

A.1 Preparación preliminar de Raspberry

En primer lugar, se describirán los pasos de preparación tanto de software para la utilización de la plataforma Raspberry Pi.

A.1.1 OS (Operating System)



Figura 26. Sistema Operativo

Se eligió un sistema operativo sin interfaz gráfica con el fin de ahorrar los recursos, para aprovecharlos de la mejor manera en las capacidades de la memoria, para el procesamiento y comunicación de los datos del sistema, para este caso se procedió a utilizar una versión del Debian GNU/LINUX que es un sistema operativo libre que se puede usar en la Raspberry, se seleccionó una versión testada siendo la versión Debian 12: Bookworm con fecha de construcción 2022.08.08. Ver Figura 26.

A.1.2 Instalación y configuración inicial de la Raspberry

Para la instalación de OS fue necesario un software de terceros en este caso se utilizó Balena Escher versión de 64bits, se eligió de objetivo la memoria SD y se procedió a instalar el sistema operativo en un tiempo aproximado de 20 min.

Muchos de los servicios y funciones no vienen instalados dentro del sistema. Por tanto, es necesario primero hacer una instalación de repositorios y del sistema en general.

Se enlista los comandos a utilizar.

Actualización de Repositorios:

```
#Apt get update
```

```
#Apt get upppgrade
```

Instalación de súper usuario:

```
#Apt get install sudo.
```

Instalación de if config y Utilidades de Red:

```
#sudo apt install net-tool
```

Con esta herramienta se llevará la gestión de las IP para el caso personal se localizará la dirección MAC de Eth0

Nota: Por defecto la contraseña #root viene vacía.

A.1.3 Instalación de Servicios de Red Inalámbricos

Estos servicios podrán ser necesarios en caso de necesitar conectar por medio del WI-FI la Raspberry, esta versión posee una antena de WI-FI 2.4GHz que puede utilizarse para conectarse de una red aislada. Para la instalación de los servicios de red inalámbrica se hace de la siguiente forma.

```
#sudo apt install wireless-tools
```

Esta versión de debían trae por defecto el servidor SSH, sin embargo, es necesario que este servicio inicie con el sistema, por tanto, se hará uso del siguiente comando para verificar los datos de arranque.

```
#sudo systemctl status ssh
```

Si antes de eso se necesita verifican si está instalado juntos con el comando anterior se puede verificar el estado del servicio con el siguiente comando.

```
#sudo service ssh status
```

Luego de una serie de pruebas fue necesario crear un nuevo usuario para tener acceso al servidor ssh, ya que por defecto el usuario root no tenía acceso debido a las credenciales por defecto.

A.1.4 Creación del usuario a utilizar (moss)

```
#add User moss
```

```
#nano -c /etc/sudoers
```

En este último archivo se le garantiza los permisos de súper usuario al ~~nuevo~~ usuario moss y de le dio una nueva contraseña siendo esta 1234 que es utilizado como un usuario para acceder a servicios.

A.2 Preparación preliminar e instalación de Docker

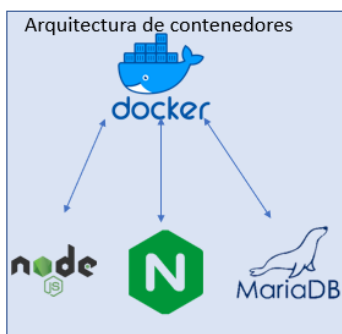


Figura 27. Arquitectura de Contenedores

Para la utilización de Docker en nuestro entorno fue necesario instalar dos paquetes principales, los cuales eran Docker y Docker Compose, ya que ambos trabajan de forma conjunta para poder brindar el servicio de contenedores.

A continuación, se muestran los comandos utilizados para la instalación de Docker.

```
$ sudo apt-get update
```

```
$ sudo apt install docker
```

Inmediatamente después de instalar Docker fue necesario instalar Docker-Compose para implementar la instalación de los contenedores.

```
$ sudo apt install docker-compose
```

```

version: '3.7'
services:
  nginx: #servidor web para servir paginas web
    image: nginx:alpine
    ports:
      - 80:80
    volumes:
      - ./default.conf:/etc/nginx/conf.d/default.conf
      - ./dist:/usr/share/nginx/html
  mariadb:
    image: mariadb
    ports:
      - 3306:3306
    volumes:
      - mariadb:/var/lib/mysql
      - ./mariadb:/home/moss/Proyecto_invernadero/mariadb
    environment:
      - MARIADB_ROOT_PASSWORD=1234
      - MARIADB_USER=moss
      - MARIADB_PASSWORD=1234
      - MARIADB_DATABASE=invernadero
  node:
    ports:
      - 3003:3003
    build:
      context: .
      dockerfile: Dockerfile_node
volumes:
  mariadb:
    .:

```

Figura 28. Especificaciones de MariaDB.

Para el armado y el servicio de los paquetes, luego de la instalación y ubicación en el archivo, se ejecutaron los siguientes comandos, estando en el mismo directorio de todos los archivos antes mencionados:

```
$ sudo docker-compose up -d
```

Se descargarán todos los contenedores especificados en el archivo luego de eso se levantarán los servicios. (Ver Figura 29).

```
moss@rp13-20220807:~/Proyecto_invernadero$ sudo docker-compose up -d
sudo: unable to resolve host rp13-20220807: Name or service not known
[sudo] password for moss:
Starting proyecto_invernadero_nginx_1 ... done
Starting proyecto_invernadero_node_1 ... done
Starting proyecto_invernadero_mariadb_1 ... done
```

Figura 29. Contenedores levantados.

Para verificar cuales contenedores están levantados se utilizó el comando

```
$ sudo docker ps
```

Mostrando la versión y el nombre de las imágenes que se han iniciado. (Ver Figura 30).

```
moss@rp13-20220807:~/Proyecto_invernadero$ sudo docker ps
sudo: unable to resolve host rp13-20220807: Name or service not known
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
012f227a7a6c  nginx:alpine  "/docker-entrypoint..." 47 hours ago  Up About a minute  0.0.0.0:80->80/tcp, :::80->80/tcp  proyecto_invernadero_nginx_1
1697004a13d0  mariadb       "docker-entrypoint.s..." 47 hours ago  Up About a minute  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp  proyecto_invernadero_mariadb_1
0af3013f2289  proyecto_invernadero_node  "docker-entrypoint.s..." 47 hours ago  Up About a minute  0.0.0.0:3003->3003/tcp, :::3003->3003/tcp  proyecto_invernadero_node_1
```

Figura 30. Verificación de servicios levantados.

A.3 Preparación de Contenedores

Con lo anterior se ha logrado establecer las imágenes de contenedores y levantarlas, pero como se mencionó anteriormente en el caso de Node y Nginx ya se estaban enviando archivos pre-compilados para que formaran parte del contenedor, únicamente fue necesario el siguiente comando:

```
$ sudo docker-compose up -d --build
```

Con eso se construyen las imágenes utilizando los datos dentro de las carpetas además que se terminan de configurar los últimos detalles dentro de los contenedores.

A.4 Configuración de MySQL

En el caso particular MariaDB necesita ciertas configuraciones para el funcionamiento de la base de datos y en este caso particular se puede crear las tablas en otra base de datos u otro servidor para posteriormente hacer una restauración de datos en el servidor ubicado dentro del contenedor de Docker, sin embargo es necesario revisar el funcionamiento completo dentro del entorno Debian de Raspberry para tener en cuenta cualquier eventualidad antes de tener listo el contenedor con la base de datos funcional, es por eso mismo que inicialmente se creó la base de datos de forma local y se hizo test del funcionamiento para luego exportar los datos a la nueva base de datos ubicada dentro del contenedor.

A.4.1.1 Habilitación del Acceso Externo de la Base de Datos

Por defecto, MariaDB no permite conexiones externas a la base de datos, restringiendo el acceso solo a localhost. Para habilitar el acceso externo, es necesario modificar un archivo de configuración.

1. Editar el archivo de configuración del servidor:

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

2. Ubicar la línea:

```
bind-address = 127.0.0.1
```

3. Reemplazarla por:

```
bind-address = 0.0.0.0
```

De esta forma se garantizará acceso externo a la base de datos además de los permisos a los usuarios creados anteriormente.

A.4.1.2 Configuración de MySQL:

Para ingresar a MySQL, se utiliza el siguiente comando:

```
$ mysql -h localhost -u root -p
```

Una vez dentro de MySQL, se procedió a crear los usuarios y otorgarles los respectivos permisos.

1. Crear un usuario para acceso local:

```
CREATE USER 'moss'@'localhost';
```

2. Crear un usuario para acceso externo:

```
CREATE USER 'moss'@'%' IDENTIFIED BY '1234';
```

3. Otorgar permisos a estos usuarios:

```
GRANT ALL PRIVILEGES ON *.* TO 'moss'@'localhost';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'moss'@'%';
```

Se pueden asignar permisos de acceso a los usuarios, incluyendo la posibilidad de agregar contraseñas para realizar modificaciones menores con permisos restringidos.

Los asteriscos en *.* se refieren a permisos sobre todas las tablas de la base de datos y a todas las bases de datos. El primer asterisco indica todas las tablas y el segundo todas las bases de datos.

Luego de la creación del usuario, si este no tiene contraseña o se le quiere asignar otra contraseña, se utiliza el siguiente comando:

```
SET PASSWORD FOR 'usuario'@'host' = 'nueva_contraseña';
```

Para eliminar usuarios, tablas o bases de datos, se utiliza el comando DROP:

```
DROP USER 'nombreusuario'@'localhost';
```

Para eliminar privilegios dentro de la base de datos, se utiliza el comando REVOKE:

```
REVOKE ALL PRIVILEGES ON *.* FROM 'usuario'@'host';
```

Para utilizar una base de datos específica, se emplea el comando USE:

```
USE invernadero;
```

MySQL gestiona toda la configuración de objetos en tablas de la base de datos nombrada: 'mysql', lo que permite verificar datos específicos, como los usuarios y sus permisos entre otros.

A.4.1.3 Creando tabla

Para crear una tabla, se utiliza el siguiente comando:

```
CREATE TABLE nombre_tabla (  
  
    nombre_columna1 tipo_de_dato,  
  
    nombre_columna2 tipo_de_dato,  
  
    ...  
  
    nombre_columnaN tipo_de_dato  
  
);
```

En este caso específico se creó la tabla sensores, con seis columnas. La primera columna corresponde a un dato identificador de nombre: ID, que actúa como llave primaria, seguida por una columna de fecha para registrar el momento de ingreso

de los datos, en el resto de las columnas se guardan valores específicos provenientes desde los sensores (Ver Figura 31).

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [invernadero]> show tables;
+-----+
| Tables_in_invernadero |
+-----+
| control                |
| sensores                |
| usuarios                |
+-----+
3 rows in set (0.001 sec)

MariaDB [invernadero]> describe sensores;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL             | auto_increment |
| fecha | timestamp     | NO   |     | current_timestamp() |
| sensor1 | decimal(7,3) | YES  |     | NULL             |
| sensor2 | decimal(7,3) | YES  |     | NULL             |
| sensor3 | decimal(7,3) | YES  |     | NULL             |
| sensor4 | decimal(7,3) | YES  |     | NULL             |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.006 sec)

MariaDB [invernadero]> exit;
Bye
noss@rpi3-20220807:~$ sudo mysql -u leo -p
```

Figura 31. Tabla resultante en MySQL.

Luego de la creación de las tablas, se verificó el ingreso de datos (se mencionará posteriormente el ingreso de datos por parte de otro de los contenedores). Al confirmar que el funcionamiento de la base de datos cumplía con las características esperadas, se realizó un respaldo de esta para prepararla para su exportación al contenedor.

A.4.1.4 Respaldo de base de datos.

Para realizar el respaldo de la base de datos, es necesario iniciar sesión como el usuario root y configurar el acceso sin solicitud de contraseña. Para ello, se crea un archivo denominado `.my.cnf` en el directorio del usuario root:

1. Iniciar sesión como usuario root en la Raspberry Pi.
2. Crear y editar el archivo: `.my.cnf`:

```
#nano .my.cnf
```

```
# nano /root/.my.cnf
```

3. Agregar las siguientes líneas al archivo:

```
[client]

user=root

password=1234

[client]

user=root

password=1234
```

La contraseña del usuario root de la base de datos MySQL se incluye en este archivo. Para asegurar la privacidad, se otorgan permisos de lectura y escritura únicamente al propietario del archivo:

```
#chmod 600 .my.cnf

chmod 600 /root/.my.cnf
```

Con esta configuración, se puede acceder al usuario root sin necesidad de contraseña, lo cual es necesario para hacer el respaldo:

```
mysqldump --add-drop-table --databases invernadero >
/home/moss/proyecto/$(date +%Y-%m-%d).sql.bak
```

Esto crea un respaldo de la base de datos en el usuario moss, que se inyectará en el contenedor de Docker. Para ello, es necesario ubicarse en el directorio donde se encuentran los datos de la base de datos en Docker (por ejemplo, /moss/Proyecto_invernadero/MariaDB).

Una vez que el servicio está levantado en Docker, se puede ingresar a una bash dentro del entorno de MariaDB mediante el siguiente comando:

```
$ sudo docker-compose exec MariaDB bash
```

En algunas situaciones, puede ser necesario instalar programas adicionales, ya que la bash está configurada con lo mínimo necesario para funcionar. Una vez dentro de la bash de MariaDB, se ubica el archivo de respaldo y se ejecuta el siguiente comando:

```
$ mysql -u root invernadero < (fecha_de_backup).sql.bak
```

De esta manera, se procedió a la restauración de la base de datos dentro del contenedor.

A.5 Instalación y uso de Node.js

Node.js es un entorno de ejecución multiplataforma de JavaScript que se utiliza en el back-end de la aplicación, junto con otros paquetes. Debido a la necesidad de ejecutar servicios, se decidió integrarlo dentro de un contenedor Docker. Inicialmente, esta parte del desarrollo se llevó a cabo en un entorno Ubuntu, ya que Node.js permite la ejecución remota de servicios desde otro terminal para realizar los procesos necesarios del back-end. Estos procesos consisten principalmente en el ingreso de datos a la base de datos y el procesamiento de estos datos.

Se utilizó un proyecto base encontrado en la plataforma GitHub, que incluía la configuración de usuarios y acceso a una plataforma web para realizar modificaciones. Este proyecto se adaptó a las necesidades del proyecto actual.

La instalación inicial se realizó en Ubuntu, se armó el proyecto y se exportó mediante STP una vez que se compiló y se logró el acceso a la base de datos. Posteriormente, la instalación se replicó tanto en Ubuntu como en Debian, utilizando los mismos comandos.

La instalación de Node.js se llevó a cabo dentro de la carpeta del proyecto, utilizando los siguientes comandos:

```
#sudo apt update
```

```
#sudo apt install NodeJS
```

Además de la instalación del paquete Node.js, se instalaron las librerías NPM y NODEMON, que son necesarias para la ejecución de elementos esenciales del proyecto.

```
#sudo apt install npm
```

Después de instalar Node.js, se utilizó npm para instalar nodemon:

```
#sudo apt install nodemon
```

Una vez instalados todos los programas, se procedió a modificar ciertos archivos del proyecto, ya que este tenía una base establecida. Se realizaron modificaciones en el archivo .env, donde se ingresaron los siguientes datos:

```
1  PORT=3003
2  DB_DATABASE=invernadero
3  DB_USER=moss
4  DB_PASSWORD=1234
5  DB_HOST=mariadb
6  DB_PORT=3306
7  DB_DIALECT=mariadb
8
9  CADUCIDAD_TOKEN=30d
10
11 SECRET_KEY=none
12
```

Figura 32. QUERY ingreso de datos.

La obtención de datos desde la base de datos se realiza en el archivo mediciones.js, que se muestra en la Figura 33.

```

1  const { queryRun } = require('../db/query');
2  const getUserQuery = async (req, res) => {
3    const results = await queryRun("select * from Users");
4    res.json({
5      results,
6    });
7  }
8  const sensoresquery = async (req, res) => {
9    const results = await queryRun("select * from sensores order by id desc limit 1");
10   res.json({
11     results,
12   });
13 }
14 module.exports = {
15   sensoresquery,
16   getUserQuery
17 };

```

Figura 33. Obtención de base de datos

Además, se colocó un archivo en HTML para verificar que había comunicación con el puerto establecido con el backend. Ver *Figura 34*.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8  </head>
9  <body>
10   <h1>Hola</h1>
11 </body>
12 </html>

```

Figura 34. Archivo de Backend.

Para comprobar el funcionamiento se utilizó el software INSOMNIA para enviar solicitudes y probar el funcionamiento de esta parte del software. Este documento se envió tal y como se había codificado al contenedor para su inicio en el sistema

A.6 Instalación y uso de Nginx

De manera similar a Node.js, se utilizó un proyecto base de Vuetify, que incluye librerías propias para la compilación. Se realizaron modificaciones en los archivos CSS, JavaScript y HTML para adaptar el proyecto a las necesidades específicas.

Los archivos modificados incluyen:

- HelloWorld.vue y App.vue: Estos archivos contienen la página principal del proyecto.
- Index.js: Este archivo trabaja con el objeto base de datos.
- Index.html: Este archivo contiene las viñetas de cada una de las aplicaciones anteriores para la presentación.

Para compilar estos documentos, se utilizó el siguiente comando:

```
#npm run build
```

Previo a la compilación, se creó una carpeta llamada "dist" que contenía los archivos compilados. Esta carpeta fue exportada al proyecto principal.

A.7 Software de Terceros

A.7.1 TailScale

TailScale es un software que sirve como alternativa para realizar conexiones por VPN (Virtual Proxy Network), este software configura de manera automática los puertos además de configurar el firewall para acceso remoto. TailScale ha sido una herramienta que facilitó la conexión remota vía SSH con el dispositivo permitiendo la prueba remota sin tener acceso físico a la tarjeta Raspberry Pi.

A.7.2 Insomnia

Insomnia es una herramienta que facilita la interacción con APIs de forma rápida y visual. Permite consumir APIs de manera sencilla, anidar peticiones y centrarse en las pruebas o análisis que se desean realizar, minimizando el tiempo de configuración del cliente. Es una herramienta útil para la estructura del proyecto actual y para realizar pruebas a las APIs implementadas.

A.7.3 Instalación, configuración y pruebas con Software de Terceros en equipos ajenos a la Raspberry.



Figura 35. Otro Software Utilizado

Estos software de terceros fueron necesarios para testeo verificación de datos y simulación. Además, que se utilizaron otro tipo de software que no es de testeo pero que vienen con el sistema como serían los IDE para la lectura modificación y compilación de los archivos, las terminales, etc.

Estos software se pueden ven la Figura 35.

Uso y pruebas con TailScale

Para la instalación de TailScale, se debe ejecutar el siguiente comando en la terminal:

```
curl -fsSL https://tailscale.com/install.sh | sh
```

Es importante considerar que para el uso de TailScale es necesario disponer de una cuenta de Google, Microsoft, entre otros proveedores de identidad. Además, se debe instalar TailScale en la PC o dispositivo de desarrollo que se utilizará para las pruebas de las aplicaciones.

En el contexto del proyecto, la utilización de TailScale resultó particularmente beneficiosa, ya que inicialmente se disponía de una única Raspberry Pi y era

imperativo conectarse a ella desde diferentes ubicaciones. TailScale facilita una conexión directa a SSH, lo que permitió trabajar con la Raspberry Pi sin necesidad de estar en la misma red local. Para ello, se debieron considerar todos los detalles pertinentes para su instalación en el sistema operativo correspondiente y asegurar su correcto funcionamiento dentro de la Raspberry Pi.

Uso y pruebas con DBeaver

Con DBeaver fue posible entablar enlace de dos formas, inicialmente se limitó a un enlace ssh para verificar la base de datos. (Ver Figura 36).

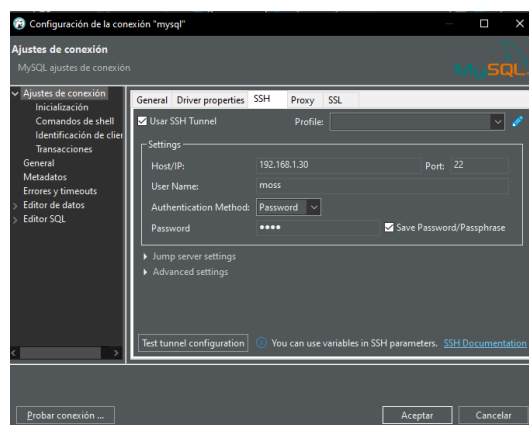


Figura 36. Interfaz de conexión de DBeaver.

Luego de verificar que todo está en orden y dar acceso remoto a la base de dato se puede hacer una conexión directa usando el puerto 3306 e ingresando las credenciales, como se puede ver en la Figura 37.

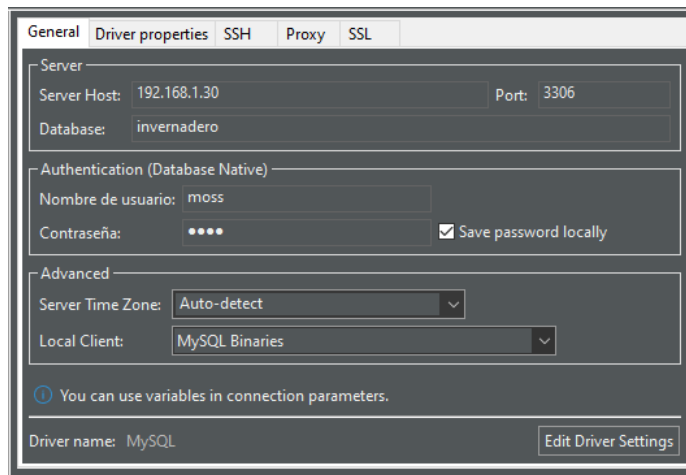


Figura 37. Conexión de DBeaver.

De esta forma se logró la verificación de ingreso de datos además del ingreso a la base de datos desde un host diferente al local. Se hace la verificación con el software DBeaver como se puede ver en la Figura 38.

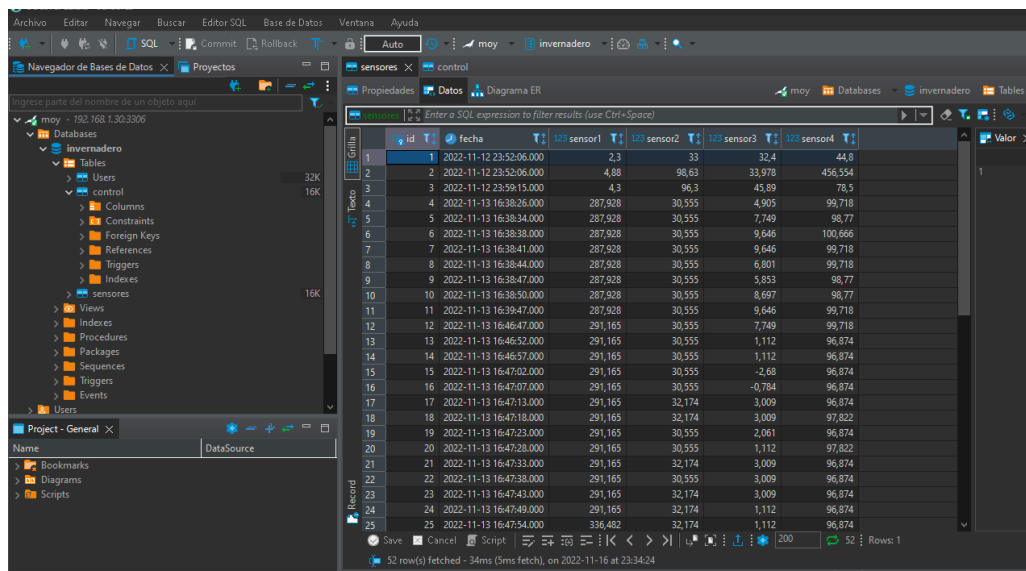


Figura 38. Base de datos consultada por DBeaver.

A.8 Enlace



Figura 39. Enlace

Durante el desarrollo del proyecto, fue necesario el envío y recepción de documentos, ya que muchos de ellos fueron tomados a partir de un proyecto base. Estos documentos se descargaron en el ordenador y se realizaron pruebas antes de ser migrados al entorno de la Raspberry Pi.

Otro aspecto fundamental fue la conexión a Internet. Anteriormente se mencionó uno de los programas que fue útil para la conexión remota; sin embargo, inicialmente fue necesario establecer una conexión a la red para que esto funcionara. Con el enlace vía Ethernet no hubo dificultad alguna, ya que simplemente se asignó una IP estática. Sin embargo, la conexión mediante Wi-Fi requirió configuraciones adicionales. Esto fue esencial para el envío y recepción de paquetes compilados, además de la actualización de repositorios y la descarga de archivos desde la web.

El funcionamiento de la aplicación se puede observar en la Figura 39.

A.8.1 Conexión a la red inalámbrica

Conexión WI-FI del dispositivo (WPA-WPA2)

```
$ sudo ifconfig wlan0 up
```

Inicialmente, es necesario activar el módulo WLAN utilizando el comando previamente mostrado. Posteriormente, se debe iniciar el módulo WLAN 0. Para verificar las señales Wi-Fi disponibles, se puede utilizar el siguiente comando:

```
$ sudo iwlist scan | grep ESSID
```

Este comando permitirá listar las redes Wi-Fi detectadas por el módulo WLAN, facilitando la selección y configuración de la red adecuada para el proyecto.

Una vez identificada la red a la que se va a conectar, es importante asegurarse de que el tipo de conexión utilice seguridad WPA o WPA2. Para facilitar la inserción de la contraseña, se creará un archivo de configuración con el siguiente comando:

```
$ sudo wpa_passphrase nombre_de_la_red contraseña > mired.conf
```

Este comando generará un archivo mired.conf con los atributos necesarios de la red. A continuación, se utilizarán los datos de este archivo para enviar una solicitud mediante el protocolo WPA con el siguiente comando:

```
$ sudo wpa_supplicant -B -i wlan0 -c mired.conf
```

Cuando aparezca el mensaje "Successfully initialized WPA supplicant", la conexión se habrá establecido correctamente. Sin embargo, es necesario activar el servicio DHCP con el siguiente comando:

```
$ sudo dhclient -v wlan0
```

De esta forma, se habrá establecido una conexión con una IP asignada vía DHCP.

A.8.2 Configuración servidor con IP estática

Inicialmente, se estableció una conexión Wi-Fi desde la Raspberry Pi utilizando DHCP, con el objetivo de actualizar repositorios, instalar paquetes, entre otras tareas. Sin embargo, dado que el uso de servidores requiere IPs estáticas, fue necesario configurar una IP fija para la Raspberry Pi, ya que muchos servicios necesitan estar configurados para evitar cambios frecuentes debido a DHCP. Por esta razón, se estableció la IP de la Raspberry Pi como 192.168.1.50 en la red local de la siguiente manera:

1. Se ubicó el archivo de configuración de interfaces de red:

```
cd /etc/network
```

2. Se agregaron los datos de la interfaz de red en el archivo correspondiente:

```
nano interfaces
```

Y se añadieron las siguientes líneas:

```
auto wlan0  
  
interface wlan0 inet static  
  
address 192.168.1.50  
  
netmask 255.255.255.0  
  
network 192.168.1.0  
  
broadcast 192.168.1.255  
  
gateway 192.168.1.1
```

3. Se reinició la Raspberry Pi y se ejecutó nuevamente el comando para establecer la conexión:

```
$sudo wpa_supplicant -B -i wlan0 -c mired.conf
```

Para verificar conexión se utilizó el comando ping apuntando al DNS de Google.

Para verificar la conexión, se utilizó el comando ping apuntando al DNS de Google:

```
ping 8.8.8.8
```

A.8.3 Enlace entre dispositivos

A.8.3.1 Ingreso al servicio SSH.

Una vez que el servicio SSH está habilitado en la Raspberry Pi, es necesario comprobar que hay conexión. En este caso, la IP de la Raspberry Pi es 192.168.1.32. Para la conexión remota desde otro dispositivo con kernel de Linux (en este caso, Ubuntu 22.04), se utilizó el siguiente comando:

```
#ssh moss@192.168.1.32
```

Al solicitar la contraseña de acceso, se ingresa a la shell del sistema. Una vez establecida la conexión, se pueden instalar los paquetes necesarios.

A.8.3.2 Envío de archivos por SCP

Luego de establecer una IP estática y probar el software en las computadoras, fue necesario enviar archivos a la Raspberry Pi utilizando SCP (Secure Copy Protocol). SCP permite la transferencia segura de archivos entre un host local y un host remoto. En este caso:

1. Se comprimió la carpeta en un archivo, ubicándose en el directorio correspondiente:

```
# tar -czvf proyecto.tgz invernadero
```

El directorio invernadero contiene el proyecto.

2. Para enviar el archivo por SCP, se utilizó el siguiente comando:

```
# scp proyecto.tgz moss@192.168.1.50:/home/moss
```

Luego de ingresar la contraseña, se descomprimió el archivo en la Raspberry Pi:

```
# tar -xzf proyecto.tgz
```

A.9 Implementación de Python en Raspberry y lectura de puertos

Instalación de Python

Inicialmente se hizo la instalación del paquete de Python 3 utilizando el siguiente comando:

```
$ sudo apt install python3
```

Una vez instalado pip3, se procedió a instalar una serie de complementos necesarios para la lectura de puertos:

Instalación de Gestor de Paquetes de Python

El gestor de paquetes pip3 se instaló de la siguiente manera:

```
$ sudo apt install python3-pip -y
```

Instalación de Funciones dentro de Python

Una vez instalado pip3 ya se pueden instalar una serie de complementos necesarios para la lectura de los puertos.

Instalación de GPIO.RPi:

```
$ sudo pip3 install RPi.GPIO
```

Instalación de MySQL Connector:

```
$ sudo pip3 install mysql-connector-python
```

Instalación de SMBus:

```
# sudo pip3 install smbus
```

Instalación de i2c-tools:

```
# sudo pip3 install i2c-tools
```

Instalación de Snapd:

```
# sudo pip3 install snapd
```

Instalación de Snapd con herramientas para i2c:

```
#sudo pip3 install snapd-i2c-tools
```

Instalación de Adafruit CircuitPython:

```
#sudo pip3 install adafruit-circuitpython-pcf8591
```

Instalación de Adafruit DHT:

```
# sudo pip3 install adafruit-circuitpython-dht
```

Referencias

- García, P., & Martínez, L. (2021). Tecnologías de sensores en la agricultura moderna. Editorial AgroTech.
- Jones, M., & Brown, T. (2018). Avances tecnológicos en la agricultura: Un enfoque práctico. GreenField Publishers.
- Lee, K., Park, S., & Kim, J. (2019). Sistemas automatizados de riego para la agricultura del futuro. TechFarm Press.
- Smith, J. (2020). Historia y evolución de la agricultura. Rural Development Press.
- Williams, D. (2017). Programación de microcontroladores para sistemas agrícolas. Electronic Farming Publishers.