ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN RECIPIENTE AUTOMATIZADO PARA DOSIFICACIÓN Y FUMIGACIÓN PARA UN INVERNADERO UBICADO EN LA COMUNA DE MIRANDA CONTROLADO POR WIFI

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

JONATHAN LEONARDO CHAUCA TOSCANO
jonathan.chauca@epn.edu.ec

JEFFERSON IVÁN TANA TOAQUIZA
jefferson.tana@epn.edu.ec

DIRECTOR: ING. PABLO ANDRÉS PROAÑO CHAMORRO
pablo.proano@epn.edu.ec

CODIRECTOR: ING. CARLOS ORLANDO ROMO HERRERA
carlos.romo@epn.edu.ec

Quito, octubre 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr Chauca Toscano Jonathan Leonardo y el Sr Tana Toaquiza Jefferson Iván como requerimiento parcial a la obtención del título de TECNÓLOGOS EN ELECTRÓNICA Y TELECOMUNICACIONES bajo nuestra supervisión:

Ing. Pablo Proaño	**Ing. Carlos Romo**
DIRECTOR DEL PROYECTO	CODIRECTOR DEL PROYECTO

DECLARACIÓN

Nosotros, Chauca Toscano Jonathan Leonardo con CI:1726740010 y Tana Toaquiza Jefferson Iván con CI: 1719592675 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin prejuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

Jonathan Leonardo Chauca Toscano
CI:1726740010
Teléfono:0986735691
Correo: jonathan.chauca@epn.edu.ec

Jefferson Iván Tana Toaquiza
CI: 1719592675
Teléfono: 0989770230
Correo: jefferson.tana@epn.edu.ec

DEDICATORIA

El presente trabajo lo dedico de todo corazón a mis padres José Chauca y Verónica Toscano quienes son el pilar fundamental en mi vida y con su ejemplo me han enseñado a dar batalla cuando las cosas se complican.

A mis hermanos quienes siempre estuvieron apoyándome en el transcurso de la carrera y me inspiraron a seguir adelante. A mi tío Segundo que toda la vida estuvo pendiente de mi formación académica.

Dedico también este logro a Giss quién me acompañó durante este proceso y me ha brindado todo su apoyo incondicional, el tiempo y los consejos para llegar a ser una mejor persona, además de permitirme compartir mis logros junto a ella.

Chauca Jonathan

AGRADECIMIENTO

Quiero agradecer en primer lugar a Dios por brindarme la salud y sabiduría en las decisiones que he tomado en mi vida y de esta manera he logrado cumplir varias metas que me he planteado.

Agradezco de igual manera, a la Escuela Politécnica Nacional y en específico a la ESFOT por permitirme continuar los estudios en esta gran institución, de la cual me llevo lindas experiencias y maravillosas amistades que conservaré toda mi vida.

Doy mi fraterno agradecimiento al Ing. Pablo Proaño director de este proyecto y a cada uno de los docentes que con su vocación compartieron su conocimiento en el ámbito académico, además de inculcarnos valores que hagan de nosotros buenos profesionales.

Brindo también un agradecimiento a mi compañero y amigo Jefferson Tana con quién hemos llevado a cabo este proyecto, y quién ha sido un gran apoyo el transcurso de esta linda carrera.

Chauca Jonathan

DEDICATORIA

En primer lugar, a Dios por fortalecerme cada día en mi camino.

A mi madre Martha Toaquiza, pues gracias a ella lo he conseguido, por cada día darme su bendición, protegerme, llenarme de confianza para afrontar todas las dificultades a lo largo de mi vida.

A mis abuelos José Toaquiza y María Ilaquiche, por darme su apoyo incondicional, además, enseñarme valores vitales para mi formación personal y profesional.

A mis mejores amigos Danny Arguello, Steven Arias, Darío Pilatasig, por siempre acompañarme y darme ánimos en mi camino.

Tana Jefferson

AGRADECIMIENTO

Especialmente a tres personas mi madre Martha Toaquiza, mis abuelos José Toaquiza y María Ilaquiche, gracias a ellos he podido culminar mi etapa universitaria, tan importante en mi vida, a mis hermanos y primos quienes han dado motivación para cumplir con mis metas propuestas.

A mi amigo Jonathan Chauca, un gran compañero, un buen amigo, quienes pasamos todos los desafíos académicos, consiguiendo lo propuesto con buenos resultados.

A mi mejor amiga Melina Rojas, quien me acompaño en muchas etapas de mi vida, además, de darme buenos consejos.

Al Ing. Pablo Proaño, por confiar en nosotros, aconsejarnos durante el desarrollo del proyecto.

A la Escuela Politécnica Nacional, siendo mi segundo hogar durante tantos años, por otorgarme tantos conocimientos.

El camino fue difícil en muchas etapas, sin embargo, gracias a todos ustedes supere cada una de las pruebas.

Tana Jefferson

ÍNDICE DE CONTENIDOS

1 INTRODUCCIÓN 1
1.1 Objetivo general 1
1.2 Objetivos específicos 1
1.3 Fundamentos 2
2 METODOLOGÍA 4
2.1 Descripción de la metodología usada 4
3 RESULTADOS Y DISCUSIÓN 6
3.1 Estudio de requerimientos del sistema implementado 6
Selección de un sensor para el control de nivel de agua del recipiente 7
Selección de la electrobomba 8
Selección del motor-agitador 9
Selección de un módulo para la conexión inalámbrica WiFi 10
Módulo reloj en tiempo real RTC 12
Selección de las electroválvulas 12
Selección del microcontrolador 15
Dimensionamiento de los calibres de cables para el sistema 16
3.2 Construcción del recipiente automatizado de dosificación y fumigación 24
Preparación para el acople de las tuberías al recipiente 24
Instalación del motor 25
Instalación del sensor ultrasónico AJ-SNR04M 25
Instalación de las electroválvulas 26
Instalación del recipiente 26
3.3 Construcción del tablero de control del sistema 27
Selección del Gabinete 28
Montaje de los componentes al tablero de control 28
Diagrama de conexiones de componentes del tablero de control 29
3.4 Control del recipiente automatizado mediante un microcontrolador 34

Programación de los menús del sistema 35
Programación del sensor ultrasónico en el Arduino IDE 39
Creación de las tablas para el registro de las actividades 44
Comunicación entre el tablero y base de datos 46
3.5 Pruebas y Análisis de Resultados 50
Pruebas de continuidad en pulsadores 50
Prueba de alimentación de los módulos. 51
Prueba de funcionamiento de relés 51
Prueba de voltaje en las borneras 52
Calibración de tiempos del sistema 52
Pruebas de verificación del registro en la base datos 54
3.6 Manual de Uso y Mantenimiento 55
Video del funcionamiento del sistema 55
Video del mantenimiento del sistema 55
4 CONCLUSIONES Y RECOMENDACIONES 56
4.1 Conclusiones 56
4.2 Recomendaciones 57
5 REFERENCIAS BIBLIOGRÁFICAS 59
ANEXOS 62
Anexo 1: Certificado de Funcionamiento i
Anexo 2: Certificado de Funcionamiento del dueño ii
Anexo 3: Presentación codigo de arduino iii
Anexo 4: Presentación codigo de esp8266 xiii
Anexo 5: Presentación codigo de php conexión con la base de datos xv
Anexo 6: Presentación codigo de php conexión con esp8266 xvi
Anexo 7: Costo del proyecto implementado xvii

ÍNDICE DE FIGURAS

Figura 3.1 Diagrama de funcionamiento de los sensores ultrasónicos 7
Figura 3.2 Sensor Ultrasónico AJ-SNR04M 8
Figura 3.3 Electrobomba 9
Figura 3.4 Motor reductor DC 10
Figura 3.5 Módulo ESP8266MOD 10
Figura 3.6 Distribución de los pines del ESP8266 11
Figura 3.7 Módulo RTC DS3231 12
Figura 3.8 Diagrama de distribución de las tuberías 13
Figura 3.9 Electroválvula a gravedad 14
Figura 3.10 Electroválvula a presión 14
Figura 3.11 Arduino Mega 15
Figura 3.12 Diagrama Unifilar del sistema 23
Figura 3.13 Ubicación de los componentes para la automatización del recipiente 24
Figura 3.14 Acoplamiento de tuberías al recipiente 24
Figura 3.15 Montaje del motor al recipiente 25
Figura 3.16 Montaje del sensor para medir el nivel 25
Figura 3.17 Instalación de las electroválvulas a las tuberías 26
Figura 3.18 Recipiente automatizado 27
Figura 3.19 Dimensiones del gabinete para el proyecto 28
Figura 3.20 Tablero de control finalizado 29
Figura 3.21 Diagrama de conexiones del tablero de control 30
Figura 3.22 Etiquetado de cables 31
Figura 3.23 Parte delantera del tablero de control 32
Figura 3.24 Distribución de pulsadores en el tablero de control 32
Figura 3.25 Distribución de pines del Arduino Mega 33
Figura 3.26 Distribución de pines del módulo ESP8266 34
Figura 3.27 Diagrama de flujo del programa de control 35
Figura 3.28 Creación de menú 36
Figura 3.29 Creación de cursor de desplazamiento 36
Figura 3.30 Representación del menú principal 37
Figura 3.31 Impresión de la actividad de riego en ejecución 37
Figura 3.32 Riego al instante ejecutándose 38
Figura 3.33 Menú de selección de nivel 38

Figura 3.34 Menú ingresar nutriente 38
Figura 3.35 Dimensiones del recipiente 39
Figura 3.36 Recipiente lleno a la mitad 40
Figura 3.37 Gráfica para hallar volumen del recipiente 41
Figura 3.38 Algoritmo para hallar la cantidad de litros 42
Figura 3.39 Impresión de la cantidad de litros del recipiente 43
Figura 3.40 Activación de cada componente mediante Relés 43
Figura 3.41 Diagrama de flujo del algoritmo de control del módulo ESP8266 44
Figura 3.42 Herramienta XAMPP 45
Figura 3.43 Crear tablas en Heidi 45
Figura 3.44 Configuración de la tabla en HeidiSQL 46
Figura 3.45 Configuración de tabla para historial de actividades 46
Figura 3.46 Configuración del nodeMCU a la red local 47
Figura 3.47 Envío de dígito de Arduino a NodeMCU 47
Figura 3.48 NodeMCU recibe dígito de Arduino mediante comunicación serial 48
Figura 3.49 Archivo conexión PHP 48
Figura 3.50 Código PHP recibo de datos 49
Figura 3.51 Esquema de comunicación entre tablero y base de datos 49
Figura 3.52 Control del riego mediante Blynk 50
Figura 3.53 Medición de la humedad 53
Figura 3.54 Registro de actividades ejecutadas 54

ÍNDICE DE TABLAS

Tabla 3.1 Especificaciones del Tablero General de Alimentación 7
Tabla 3.2 Especificaciones del sensor AJ-SR04M 8
Tabla 3.3 Características de la Electrobomba 9
Tabla 3.4 Especificaciones del Motor DC 10
Tabla 3.5 Especificaciones del Módulo WiFi 11
Tabla 3.6 Especificaciones del Módulo RTC DS3231 12
Tabla 3.7 Especificaciones de las electroválvulas 15
Tabla 3.8 Características del Arduino Mega 15
Tabla 3.9 Conductor seleccionado para las electroválvulas 17
Tabla 3.10 Conductor seleccionado para la electrobomba 18
Tabla 3.11 Conductor calculado para el motor agitador 18
Tabla 3.12 Resumen de los cables calculados para los componentes del sistema 19
Tabla 3.13 Protecciones para el cable 18 AWG 19
Tabla 3.14 Protecciones para la electrobomba y el motor 19
Tabla 3.15 Amperajes por carga del sistema 20
Tabla 3.16 Conductor seleccionado para la acometida del tablero de control 22
Tabla 3.17 Componentes para el Tablero de Control 27
Tabla 3.19 Pruebas de continuidad en pulsadores 50
Tabla 3.20 Valores medidos del voltaje de los dispositivos electrónicos 51
Tabla 3.21 Funcionamiento de relés 51
Tabla 3.22 Valores de voltajes medidos en las borneras de cada componente 52
Tabla 3.23 Tiempos para ejecución de actividades 54

RESUMEN

El presente documento trata acerca de la implementación de un sistema que automatiza las actividades dentro de un invernadero, tales como: riego, dosificación y preparación de pesticidas para la fumigación. De esta manera se brinda una forma más eficiente de distribuir los nutrientes necesarios para el buen desarrollo de los cultivos, reduciendo al mínimo las pérdidas.

El documento se divide en cuatro capítulos que se especifican a continuación:

En el primer capítulo se tiene una introducción en donde se detalla la problemática que tienen los trabajadores agrícolas sobre los cuidados y el tiempo que conlleva la ejecución de las tareas dentro del invernadero. Por tal motivo, se realiza un sistema que solvente las necesidades de los usuarios finales.

En el segundo capítulo se presenta la metodología seleccionada para desarrollar el proyecto. Además, describe cada uno de los objetivos planteados.

En el tercer capítulo se especifican todos los componentes necesarios para la elaboración del proyecto como: microcontrolador, módulos, cables de conexión, electroválvulas, motor y bomba. Además, se muestran los resultados obtenidos de las pruebas de funcionamiento realizadas al sistema.

El cuarto capítulo se compone de las conclusiones y recomendaciones que se consiguieron durante el desarrollo, la ejecución y las pruebas de funcionamiento del proyecto.

PALABRAS CLAVES: riego, dosificación, preparación de pesticidas, microcontrolador, módulo WiFi.

ABSTRACT

This paper deals with the implementation of a system that automates activities within a greenhouse, such as irrigation, dosing and preparation of pesticides for spraying. In this way, a more efficient way of distributing the nutrients necessary for the good development of the crops is provided, reducing losses to a minimum.

The document is divided into four chapters, which are specified below:

In the first chapter there is an introduction where the problems that agricultural workers have about the care and time involved in the execution of tasks inside the greenhouse are detailed. For this reason, a system is developed to meet the needs of the end users.

The second chapter presents the methodology selected to develop the project. In addition, it describes each of the proposed objectives.

In the third chapter, all the components necessary for the development of the project are specified, such as: microcontroller, modules, connection cables, solenoid valves, motor and pump. In addition, the results obtained from the operating tests carried out on the system are shown.

The fourth chapter consists of the conclusions and recommendations obtained during the development, execution and performance tests of the project.

KEY WORDS: irrigation, dosing, pesticide preparation, microcontroller, WiFi module.

1 INTRODUCCIÓN

La comuna de Miranda es un sector que contiene una gran cantidad de invernaderos que, debido a las propiedades del suelo y el clima han visto en esta área una gran oportunidad para el desarrollo de la agricultura, siendo así, emprendimientos que han permitido generar varios productos alimenticios para el consumo de la población. El trabajo manual que llevan a diario los profesionales agrónomos y trabajadores se basa en el riego, dosificación de nutrientes y fumigación de sus cultivos demandando horas de supervisión, además, de un desperdicio excesivo de agua y productos agroquímicos. Teniendo en cuenta todas las actividades que se realiza en un invernadero, se evidenció la necesidad de realizar un sistema automatizado que permita tener un control en cada uno de los procesos antes descritos.

El sistema implementado permite realizar la etapa de riego por goteo, al instante que sea requerido, o a su vez, cuenta con la opción de riego programado, la cual, consiste en regar el cultivo diariamente a una hora establecida en el programa. Por otro lado, permite realizar la preparación de nutrientes mediante un recipiente automatizado y luego distribuirlos hacia el cultivo, así mismo, realiza el proceso de preparación de pesticida para la fumigación cuando ésta sea requerida. El presente proyecto implementado contribuye directamente en el control de cada una de las tareas que se realizan dentro del invernadero, brindando al agricultor mayor seguridad, organización y satisfacción al contar con procesos automatizados y que le permitan mejorar su calidad de producción.

1.1 Objetivo general

Implementar un recipiente automatizado para dosificación y fumigación para un invernadero de la comuna de Miranda controlado mediante conexión inalámbrica.

1.2 Objetivos específicos

- Realizar un estudio de requerimientos del sistema a implementarse.
- Construir un recipiente automatizado de dosificación y fumigación.
- Construir el tablero de control del sistema.
- Controlar el recipiente automatizado mediante un microcontrolador.
- Probar el funcionamiento del recipiente automatizado con actividades dentro del invernadero.
- Elaborar un manual de uso y de mantenimiento del sistema.

1.3 Fundamentos

XAMPP

Es una plataforma de código libre, la cual, integra varios programas como: el servidor *web* Apache, gestor de base de datos MySQL e intérpretes de lenguaje *script* PHP. XAMPP permite probar sin necesidad de internet las páginas *web* desarrolladas en el mismo computador [1].

MySQL

Corresponde a un sistema de gestión de base de datos, el cual, tiene como una principal ventaja su libre distribución, además es de código abierto. La libre distribución permite ser descargado de manera libre desde internet y el código abierto hace referencia a que cualquier persona puede modificar el código para realizar mejoras [2].

Visual Studio Code

Es un programa de código abierto, que permite editar varios lenguajes de programación, además, gestiona atajos para minimizar la dificultad de código. Por otro lado, posee una gran variedad de extensiones para distintos lenguajes como: PHP, Python, C#, C ++, Java [3].

HeidiSQL

Es un entorno gráfico que permite crear tablas y registros mediante la conexión de una base de datos MySQL. El propósito de la aplicación es ayudar a la gestión de los datos de una manera más gráfica y flexible. Esta herramienta, además, permite ejecutar bases de datos como MicrosoftSQL, SQLite y MariaDB [4].

LiquidCrystal_I2C.h

Es una librería que tiene como función controlar pantallas mediante la comunicación I2C, presenta funciones muy parecidas a la librería LiquidCrystal [14]. La gran ventaja de esta librería es que ayuda a reducir la cantidad de pines necesarios para la conexión de la pantalla LCD al microcontrolador debido a que solo usa dos pines para la comunicación.

LiquidMenu.h

Es una biblioteca que está relacionada con la librería LiquidCrystal. Se caracteriza porque permite crear menús, de tal manera que el proceso se vea simplificado. Esta biblioteca desglosa los elementos de un menú mediante clases organizadas de manera

jerárquica [15]. Mediante esta librería se desarrolló los menús y submenús que se muestran en el LCD como una interfaz para el usuario, además, gracias a las funciones que trae esta librería se programó un cursor que permite la navegación entre los datos de cada menú o submenú desplegado.

Wire.h

Esta librería tiene como función realizar una comunicación con dispositivos I2C. Se utiliza los pines SDA conocido como línea de datos y SCL conocido como la línea de reloj. Se debe tener en cuenta que la biblioteca maneja un búfer de 32 *bytes*, por tal motivo, no se debe exceder ese límite a la hora de la comunicación ya que se eliminan los *bytes* que se encuentren fuera del límite [16].

RTClib.h

Es una librería desarrollada para la programación del módulo *Real Time Clock*. Permite establecer la fecha y hora de manera sencilla mediante la función que se muestra a continuación:

DateTime (Año, Mes, Día, Hora, Minuto, Segundo) [8].

SoftwareSerial.h

Esta librería ha sido desarrollada para lograr la comunicación tipo serial en distintos pines digitales del microcontrolador Arduino Mega. Con esta configuración se tiene velocidades de 115200 (bps) [17].

2 METODOLOGÍA

2.1 Descripción de la metodología usada

El presente proyecto se desarrolló en base a las tareas principales que se realizan en el invernadero. Cada una de estas actividades han sido integradas a un tablero de control, desde el cual el agricultor cuenta con una pantalla LCD para la selección de la actividad a ejecutar. El sistema se encuentra controlado mediante un módulo Arduino Mega que a su vez interactúa con un módulo WiFi permitiendo mediante esta interfaz inalámbrica controlar la función de riego. Para el cumplimiento de los objetivos específicos mencionados anteriormente, se realizaron las siguientes actividades:

1. Estudio de requerimientos del sistema.

Se procedió a examinar el modo de ejecución manual de los procesos de riego, dosificación y fumigación en el invernadero. Además, se observó el esquema de distribución de las tuberías, en donde se midió su diámetro. Por otro lado, se verificó la ubicación y el número de válvulas manuales que controlan cada tarea. A continuación, se seleccionó un motor que cumpla con la función de mezclado de nutrientes, una bomba para la distribución, y un sensor ultrasónico para controlar el nivel de agua y los módulos requeridos tanto para la conexión inalámbrica, como para el ajuste de fecha y hora del sistema. Finalmente, se obtuvo las mediciones exactas de acuerdo con la ubicación del tablero de control con respecto a los componentes tales como: electroválvulas, motor y bomba. Según estos datos se realizó el dimensionamiento de cables.

2. Construcción del recipiente automatizado de dosificación y fumigación.

En esta etapa se procedió a realizar el montaje del motor en la parte superior céntrica del recipiente, el cual, cumple la función de mezclado de nutrientes y pesticidas, además, se ubicó un sensor ultrasónico en la parte superior del recipiente, de tal manera que no intervenga con el mezclador y tenga la capacidad de medir sin ningún inconveniente el nivel de agua de acuerdo con la selección del agricultor. A continuación, se realizó el acople para la ubicación de las tuberías en el recipiente y se instaló las electroválvulas que controlan tanto el llenado como el vaciado del recipiente.

3. Elaboración del tablero de control.

Durante esta etapa se procedió a montar en el tablero cada uno de los componentes que permiten el funcionamiento de todo el sistema tales como: fuentes de alimentación,

protecciones, módulos electrónicos, microcontrolador, pulsadores, LCD y canaletas. De igual manera, se realizó la interconexión de los dispositivos utilizando el respectivo cable que fue dimensionado anteriormente, logrando así, la implementación de la parte de control del sistema. Finalmente, se verificó que todo se encuentre conectado de manera correcta mediante pruebas de continuidad.

4. Control del recipiente automatizado mediante un microcontrolador.

Para la elección de la placa Arduino, se tuvo en cuenta las características que posee cada una de ellas, número de pines, indicadores, módulos RTC, Relés, ESP8266. Una vez finalizada la programación en Arduino Entorno de Desarrollo Integrado (IDE) se comprobó su funcionalidad. Con ayuda del programa Proteus se consiguió simular la mayoría de las actividades, por último, se comprobó mediante un *protoboard* el funcionamiento del LCD, módulos RTC, WiFi y Relés.

La comunicación inalámbrica posee una programación que se carga al ESP8266-MCU, el código se elaboró en Arduino IDE, con la finalidad de vincular el microcontrolador para tener acceso a Internet, la forma de interactuar es mediante una página *web*, diseñada en HTTML, mediante el programa *Visual Studio Code*, al mismo tiempo, se generó la base de datos para el registro de actividades mediante el sistema de gestión MySQL, por otra parte, la tabla de actividades se generó con el programa libre HeidiSQL.

5. Prueba del funcionamiento del recipiente automatizado con actividades dentro del invernadero.

Para verificar que el sistema funcione correctamente se hicieron las siguientes pruebas:

a. Pruebas a las protecciones del tablero de control, con el sistema energizado.
b. Comprobación de la comunicación inalámbrica.
c. Verificación del motor en el proceso de mezclado.
d. Medición de la humedad de la tierra, con el método del tacto.

Finalmente, se comprobó que cada una de las actividades sean ejecutadas con normalidad.

6. Elaboración de un manual de uso y de mantenimiento del sistema.

Con la finalización del sistema implementado se realizó un video para instruir al usuario sobre el funcionamiento del sistema, en éste se muestra cómo activar o desactivar las actividades programadas. Por otro lado, se realizó un video sobre el mantenimiento del sistema para evitar futuros fallos.

3 RESULTADOS Y DISCUSIÓN

Para el funcionamiento del sistema, se cuenta con un tablero de control desde el cual, el usuario tiene acceso a las funciones de riego, dosificación de nutrientes y fumigación. Dentro de la función de riego se encuentran establecidos dos parámetros: riego al instante o riego programado. El riego al instante permite activar las electroválvulas de inmediato para dosificar al invernadero de agua, en cambio, al seleccionar el riego programado, el sistema activará el riego del cultivo a una hora específica ya establecida en la programación, cuyo parámetro ha sido recomendado por el usuario. Además, se establece una conexión del sistema a la red WiFi del invernadero para el acceso a la función de riego desde una aplicación *web*.

Por otro lado, el tablero de control contiene una pantalla LCD donde se visualiza un menú con las funciones de: dosificación de nutrientes y fumigación. La dosificación de nutrientes se basa en seleccionar la cantidad de litros para el llenado del recipiente. Una vez concluida esta etapa solicita al usuario el ingreso manual de los nutrientes y posteriormente se realiza la mezcla y mediante una bomba se distribuye al cultivo.

Para la etapa de fumigación se cumple el mismo proceso descrito en la dosificación.

3.1 Estudio de requerimientos del sistema implementado

Para el estudio de los requerimientos y condiciones, se realizó una visita al invernadero ubicado en la provincia de Cotopaxi en la parroquia de Salcedo, en la comuna de Miranda. Se examinó el lugar para tener constancia de cómo se realizan las actividades dentro del invernadero. Por otro lado, se recolectó datos como distancias del tablero de energía hasta el lugar de implementación del proyecto, además, se constató la cobertura de la red WiFi necesaria para el desarrollo del sistema. Finalmente, se elaboró un esquema de las tuberías que están distribuidas en el invernadero, teniendo en cuenta su función y de acuerdo con esto se seleccionó el número de electroválvulas a instalar.

A continuación, se examinó el tablero principal de energía para el dimensionamiento de los componentes obteniendo los datos presentados en la Tabla 3.1.

Tabla 3.1 Especificaciones del Tablero General de Alimentación

Tablero Principal de Alimentación	
Voltaje	113.7
Tipo de Voltaje	AC
Tipo de Corriente	AC
Alimentación	Monofásica

Selección de un sensor para el control de nivel de agua del recipiente

Para el control automatizado del nivel de agua del recipiente se selecciona un sensor ultrasónico, debido a que son capaces de medir la distancia de objetos o líquidos mediante ondas ultrasónicas, el principio de funcionamiento se basa en emitir pulsos acústicos mediante el aire y que estos sean reflejados al encontrar un objeto. De esta manera lo que mide el sensor es el tiempo entre la emisión y recepción del pulso, por consiguiente, al medir la distancia mediante el tiempo de recorrido de las ondas del sonido hace que estos sensores presenten insensibilidad ante el ruido de fondo [5]. En la Figura 3.1 se representa gráficamente el principio de funcionamiento de un sensor ultrasónico.

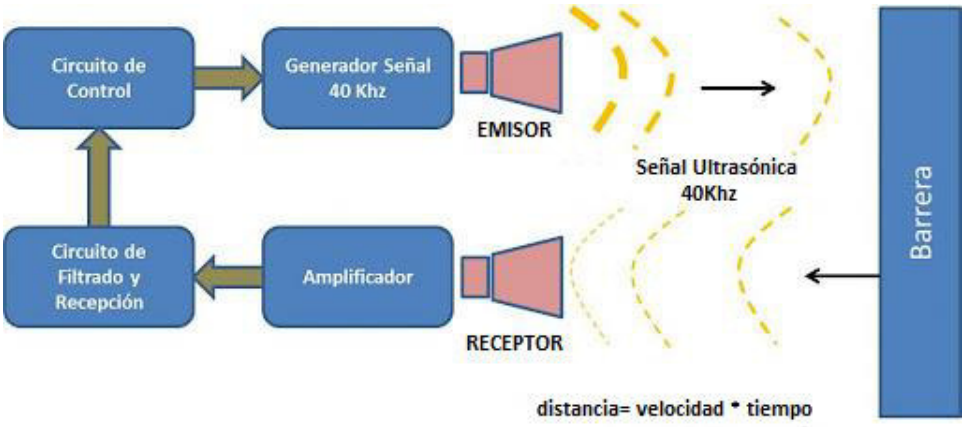

Figura 3.1 Diagrama de funcionamiento de los sensores ultrasónicos [6]

Como se observa en la Figura 3.1 estos sensores no requieren tener un contacto directo con el líquido, por lo que, son capaces de brindar mediciones limpias. Además, este tipo de sensores por su diseño son pequeños y fáciles de instalar. En la Figura 3.2 se presenta el tipo de sensor ultrasónico utilizado en el proyecto.

Figura 3.2 Sensor Ultrasónico AJ-SNR04M [7]

En la Tabla 3.2 se detallan las características del sensor ultrasónico AJ-SR04M utilizado para medir el nivel de agua del recipiente:

Tabla 3.2 Especificaciones del sensor AJ-SR04M [7]

Sensor ultrasónico AJ-SR04M	
Voltaje Operación	5 (V_{DC})
Corriente de trabajo	30 (mA)
Rango de detección	25 a 450 (cm)
Zona muerta	< 20 (cm)
Precisión	3 (mm)
Frecuencia acústica	40 (kHz)
Duración pulso disparo	10 (uS)
Temperatura de trabajo	-10 – 70 (ºC)
A prueba de agua	Parte delantera

Selección de la electrobomba

Para la selección de la bomba se tuvo en cuenta el caudal, es decir, la cantidad de litros a extraer del tanque en un determinado tiempo. Para dicho cálculo se procede de la siguiente manera:

$$Q = \frac{V}{t}$$

Ecuación 3.1 Caudal [4]

Donde:

V : 105 (l) Volumen en litros a extraer del tanque

t : 4 (min) tiempo en minutos de extracción

Usando la Ecuación 3.1 se obtiene:

$$Q = 26{,}25\ (\mathrm{l/min})$$

En base al cálculo obtenido se elige una bomba cuyo caudal aproximado al cálculo es de 35 (l/min).

En la Figura 3.3 se muestra la electrobomba usada en el proyecto.

Figura 3.3 Electrobomba [4]

A continuación, en la Tabla 3.3 se muestran las características de la electrobomba.

Tabla 3.3 Características de la Electrobomba [4]

Electrobomba TRITON TRI-40	
Voltaje Operación	110 (V_{AC})
Corriente Nominal	3.6 (A)
Frecuencia	60 (Hz)
Grado de protección IP	54
Potencia	0.37 (W)
Qmax	35 (l/m)
Hmax	35 (m)

Selección del motor-agitador

Para realizar la mezcla de la materia prima para el proceso de dosificación y fumigación se seleccionó un motor DC que además de su precio y tamaño cumple con la función de mezclado como se muestra en la Figura 3.4 y cuyas especificaciones se muestran en la Tabla 3.4.

Figura 3.4 Motor reductor DC [15]

Tabla 3.4 Especificaciones del Motor DC [15]

Motor DC	
Voltaje Operación	12 - 24 (V_{DC})
Corriente Nominal	4 (A)
RPM	80 - 160

Selección de un módulo para la conexión inalámbrica WiFi

Mediante este módulo se establece la comunicación entre la placa Arduino y la computadora local permitiendo de esta manera el envío de datos para el registro de las actividades ejecutadas. Además, mediante una aplicación móvil se puede seleccionar el riego instantáneo o el riego programado. En la Tabla 3.5 se detalla las características del módulo WiFi seleccionado para el proyecto.

Figura 3.5 Módulo ESP8266MOD [5]

Tabla 3.5 Especificaciones del Módulo WiFi [4]

Módulo WiFi **ESP8266MOD**	
Voltaje Operación	3 – 3.6 (V_{DC})
Corriente Operación	80 (mA)
Módulo WiFi	2.4 (GHz)
Temperatura de operación	-40 a 125 (°C)
Comunicación	UART, SPI, I2C
WiFi IEEE 802.11	b/g/n
Consumo	0.0005 a 170 (mA)
Potencia de Salida	+19.5 dBm en 802.11b

El módulo contiene un Sistema en *chip* (SoC), se trata de un integrado y su funcionamiento es de manera autónoma, tal como si se tratara de un computador. Una desventaja de este *chip* es que no contiene una memoria para el almacenamiento de programas, por tal motivo, parte de sus pines son utilizados para la conexión de una memoria *flash* externa. El ESP8266 es capaz de conectarse a la red WiFi con el propósito de enviar y recibir datos desde Internet [8]. Este módulo contiene pines distribuidos de la siguiente manera:

- Pines digitales D0-D12
- Pin analógico A0
- Pin de 3.3 (V)
- Pin de 5 (V)
- Pin GND

En la Figura 3.6**,** se muestra de manera general la distribución de pines.

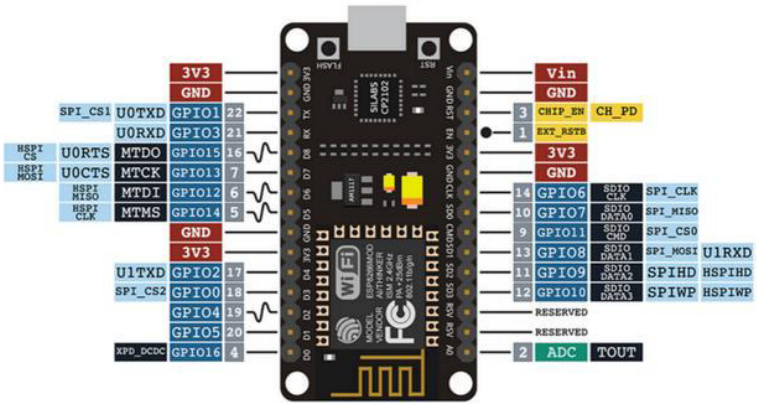

Figura 3.6 Distribución de los pines del ESP8266 [9]

Módulo reloj en tiempo real RTC

Permite que el sistema esté configurado con la fecha y la hora en tiempo real, mediante esta aplicación se lleva a cabo el riego programado. En la Tabla 3.6 se muestran las características que contiene este módulo y en la Figura 3.7 se presenta el módulo.

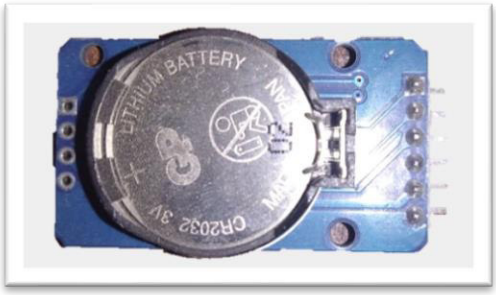

Figura 3.7 Módulo RTC DS3231 [9]

Tabla 3.6 Especificaciones del Módulo RTC DS3231 [10]

Módulo RTC DS3231	
Voltaje Operación	3.3 – 5 (V_{DC})
Exactitud del reloj	2ppm
Interfaz de comunicación	I2C
Batería	10 años

Selección de las electroválvulas

Para la selección de las electroválvulas se realizó un esquema de la distribución de las tuberías del invernadero como se muestra en la Figura 3.8.

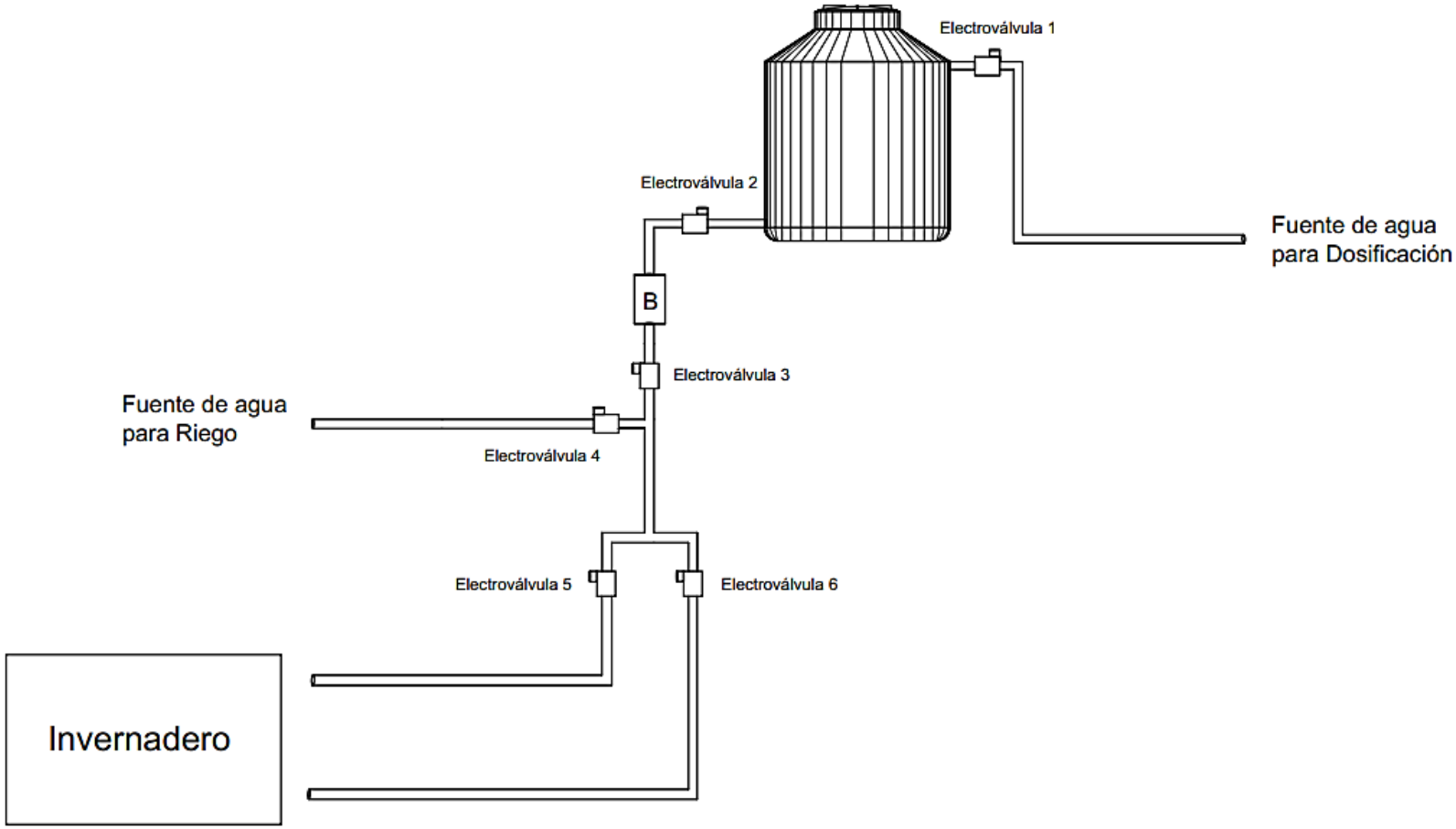

Figura 3.8 Diagrama de distribución de las tuberías

El sistema requiere un total de seis electroválvulas de acuerdo con la distribución de tuberías en todo el invernadero, de las cuales, dos son electroválvulas a gravedad que fueron ubicadas estratégicamente en los puntos donde se presenta menos presión de agua. El principio de funcionamiento de las electroválvulas a gravedad se basa en el modo de activación, ya que no requiere de una presión mínima del agua para su operación. A diferencia de las electroválvulas a presión que para su operación necesitan de una presión mínima. En la Figura 3.9 se visualiza la electroválvula a gravedad y en la Figura 3.10 se muestra la electroválvula a presión.

Figura 3.9 Electroválvula a gravedad [11]

Figura 3.10 Electroválvula a presión [11]

En la Tabla 3.7 se detallan las características de las electroválvulas seleccionadas:

Tabla 3.7 Especificaciones de las electroválvulas [11]

Electroválvulas a gravedad	
Voltaje Operación	110 (V_{AC})
Corriente de Operación	300 (mA)
Presión	0 - 8 (Bar)
Diámetro Tubería	½ (in)
Electroválvulas sin gravedad	
Voltaje Operación	110 (V_{AC})
Corriente de Operación	300 (mA)
Presión	0.02 – 0.8 (Mpa)
Diámetro Tubería	½ (in)

Selección del microcontrolador

El microcontrolador que se seleccionó para el proyecto es el módulo Arduino Mega por la cantidad de pines, además de que presenta mucha facilidad para manejar programas complejos. A continuación, se detallan las especificaciones en la Tabla 3.8 y en la Figura 3.11 se presenta el módulo Arduino.

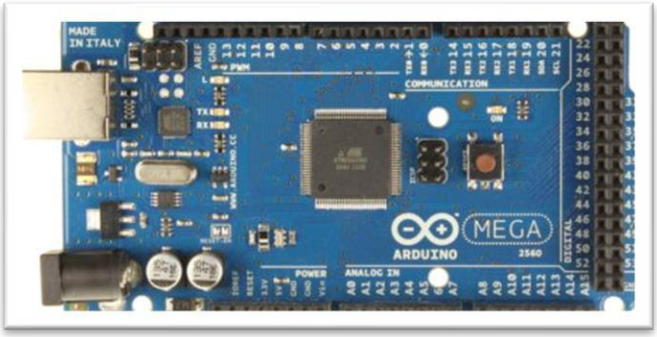

Figura 3.11 Arduino Mega [12]

Tabla 3.8 Características del Arduino Mega [12]

Módulo Arduino Mega	
Voltaje Operativo	5 (V_{DC})
Voltaje Entrada	7 – 12 (V_{DC})
Pines digitales	54
Pines análogos	16
Corriente por cada pin	40 (mA)

Módulo Arduino Mega	
Memoria flash	256 (KB)
Velocidad de reloj	16 (MHz)

Dimensionamiento de los calibres de cables para el sistema

El dimensionamiento del conductor se seleccionó de tal manera que la distancia a la cual son ubicados los componentes cumpla con la caída de tensión máxima del 3%, asegurando de esta manera el voltaje requerido para el funcionamiento de cada dispositivo.

Cálculo del conductor para las electroválvulas

A continuación, se realiza el cálculo de calibre de cable para las electroválvulas, teniendo en cuenta la que se encuentra a mayor distancia del tablero de control. Para dicho cálculo se procede de la siguiente manera:

$$\Delta V = V_{nom} \cdot \%$$

Ecuación 3.2 Caída de tensión máxima admisible [15]

Donde:

ΔV : Máxima caída de tensión admisible

V_{nom} : 113.7 (V_{AC}) Voltaje de la instalación

$\%$: 3 (%) Porcentaje máximo de caída de tensión

Usando la Ecuación 3.2 se obtiene:

$$\Delta V = 3.4\ (V)$$

En base al cálculo obtenido y empleando las ecuaciones 3.2 y 3.3 se halla el valor de la resistencia de cable.

$$R = \frac{\Delta V}{2 \cdot I}$$

Ecuación 3.3 Resistencia en función de la caída de tensión [16]

Donde:

I : 300 (mA) corriente

R : (Ω) resistencia

Usando la Ecuación 3.3 se obtiene:

$$R = 5.69\ (\Omega)$$

Finalmente, en base a la resistencia calculada empleando las ecuaciones 3.3 y 3.4 se halla el área del cable:

$$A = \frac{P \cdot d}{R}$$

Ecuación 3.4 Sección de cable en función de la resistividad del conductor [17]

Donde:

P : $1.71 * 10^{-8}$ resistividad del cobre

R : 5.69 (Ω) resistencia

d : 35 (m) distancia

Por lo tanto:

$$A = 0.104\ (mm^2)$$

Se selecciona el calibre de acuerdo al área obtenida, siendo así 26 AWG el más aproximado cuya sección nominal corresponde a 0,128 mm^2. Sin embargo, la norma de instalaciones eléctricas recomienda un calibre de cable mínimo para electroválvulas de 20 AWG. Basándose en este dato se decidió escoger el calibre 18 AWG debido a la disponibilidad en el mercado. En la Tabla 3.9 se presentan los datos sección y amperaje para el calibre seleccionado.

Tabla 3.9 Conductor seleccionado para las electroválvulas [18]

Calibre AWG	Sección nominal	Amperaje
18	0.8230 (mm^2)	10 (A)

Cálculo del conductor para la electrobomba

A continuación, se realiza el dimensionamiento del cable y la protección de la bomba, se debe tener en cuenta que se realiza el dimensionamiento con el 125% de la corriente nominal, según la recomendación de la *National Electrical Code* (NEC) [10], como se presenta en la Ecuación 3.5.

$$\mathrm{Ip} = \mathrm{In} \cdot 1.25$$

Ecuación 3.5 Corriente de protección [19]

Donde:

Ip : Corriente de protección al 125% de la In

In : 3.6 (A) corriente nominal

Usando la ecuación 3.5 se obtiene:

$$\mathrm{Ip} = 4.5\ (\mathrm{A})$$

Una vez obtenido el cálculo de la corriente, se realizó el cálculo de caída de tensión de la misma manera que en la sección anterior. En la Tabla 3.10 se muestran los resultados.

Tabla 3.10 Conductor seleccionado para la electrobomba [17]

Iprot.	ΔV	Resistencia	Área	Calibre AWG	Sección nominal	Amperaje
4.5 (A)	3.4 (V)	0.38 (Ω)	0.225 (mm^2)	14	2.08 (mm^2)	25 (A)

Se seleccionó el calibre de acuerdo con el área obtenida, según los cálculos se tiene un cable 22 AWG cuya sección nominal corresponde a 0.51 (mm^2). Sin embargo, la norma de instalaciones eléctricas recomienda un calibre de cable mínimo para electrobombas de 14 AWG [20].

Cálculo del conductor para el motor agitador DC

A continuación, se realizó el mismo proceso para el dimensionamiento del motor DC tal como se lo hizo en la electrobomba. En la Tabla 3.11 se muestran los resultados del cálculo realizado.

Tabla 3.11 Conductor calculado para el motor agitador [17]

Iprot.	ΔV	Resistencia	Área	Calibre AWG	Sección nominal	Amperaje
5 (A)	3 (V)	0.3 (Ω)	0.285 (mm^2)	18	0.8230 (mm^2)	10 (A)

A continuación, en la Tabla 3.12 se muestra el resumen del dimensionamiento de calibres de cable de los componentes del sistema.

Tabla 3.12 Resumen de los cables calculados para los componentes del sistema [17]

Componentes	Calibre AWG	Amperaje	Tipo de cable
Electroválvulas	18	10 (A)	THHW
Electrobomba	14	25 (A)	THHW
Motor (Agitador) DC	18	10 (A)	THHW

A continuación, en la Tabla 3.13 se especifica las protecciones del conductor seleccionado para cada electroválvula:

Tabla 3.13 Protecciones para el cable 18 AWG [18]

Componentes	Calibre AWG	Amperaje	Fusible
Electroválvulas	18	10 (A)	7.5 (A)

Además, se elige la protección de la electrobomba y el motor agitador en base a sus corrientes calculadas y se muestran en la Tabla 3.14.

Tabla 3.14 Protecciones para la electrobomba y el motor [17]

Componentes	Corriente de protección	Breaker
Electrobomba	4.5 (A)	6 (A)
Motor DC	5 (A)	6 (A)

Cálculo del cable para la acometida del tablero de control

Para el cálculo se tomó en cuenta el amperaje de todas las cargas y se analizó el factor de uso tomando en cuenta el número de cargas que están conectadas a la vez. El factor de uso del sistema corresponde al 70 %. Además, para la acometida se establece un porcentaje del 1% de caída de tensión según la Norma Ecuatoriana de la Construcción [11]. En la Tabla 3.15, se muestran los valores de amperaje de todas las cargas del sistema.

Tabla 3.15 Amperajes por carga del sistema [17]

	Componentes	Amperaje	Total
Electrónica	Arduino Mega	500 (mA)	0.9 (A)
	Sensor AJ-SR04M	30 (mA)	
	Módulo Relé	160 (mA)	
	Módulo WiFi	80 (mA)	
	Módulo RTC	5 (mA)	
	LCD Display	125 (mA)	
Electroválvulas	Electroválvula 1	300 (mA)	1.8 (A)
	Electroválvula 2	300 (mA)	
	Electroválvula 3	300 (mA)	
	Electroválvula 4	300 (mA)	
	Electroválvula 5	300 (mA)	
	Electroválvula 6	300 (mA)	
	Electrobomba	4.5 (A)	4.5 (A)
	Motor-agitador DC	5 (A)	5 (A)
TOTAL			**12,2 (A)**

A continuación, se procede a realizar el cálculo para el conductor de la acometida del tablero de control:

$$I_A = I_{total} \cdot FD$$

Ecuación 3.6 Corriente de acometida en relación con el factor de uso [20]

Donde:

I_A : Corriente de acometida

I_{total} : 12.2 (A) corriente total del sistema

FD : 70 (%) Factor de uso

Usando la ecuación 3.6 se obtiene:

$$I_A = 9\ (A)$$

Se procede a realizar el método de caída de voltaje:

$$\Delta V = V_{nom} \cdot \%$$

Ecuación 3.7 Caída de tensión máxima admisible [15]

Donde:

ΔV : Máxima caída de tensión admisible

V_{nom} : 113.7 (V) Voltaje de la instalación

$\%$: 1 (%) Porcentaje máximo de caída de tensión

Usando la Ecuación 3.7 se obtiene:

$$\Delta V = 1.137\ (V)$$

En base al cálculo obtenido y empleando las ecuaciones 3.7 y 3.8 se halla el valor de la resistencia de cable.

$$R = \frac{\Delta V}{2 \cdot I_A}$$

Ecuación 3.8 Resistencia en función de la caída de tensión [16]

Donde:

I_A : 9 (A) corriente

R : (Ω) resistencia

Usando la Ecuación 3.8 se obtiene:

$$R = 0.07\ (\Omega)$$

Finalmente, en base a la resistencia calculada empleando las ecuaciones 3.8 y 3.9 se halla el área del cable:

$$A = \frac{P \cdot d}{R}$$

Ecuación 3.9 Sección del cable en función de la resistividad del conductor [17]

Donde:

P : $1.71 * 10^{-8}$ resistividad del cobre

R : 0.07 (Ω) resistencia

d : 12 (m) distancia

Por lo tanto:

$$A = 3.06\ (mm^2)$$

Para la acometida se establece un conductor de calibre 12 AWG cuya sección nominal corresponde a 3.310 (mm^2) y sus especificaciones de presentan en la Tabla 3.16.

Tabla 3.16 Conductor seleccionado para la acometida del tablero de control [18]

Calibre AWG	Sección nominal	Amperaje
12	3.310 (mm^2)	30 (A)

Finalmente, en la Figura 3.12 se muestra el diagrama unifilar correspondiente al sistema.

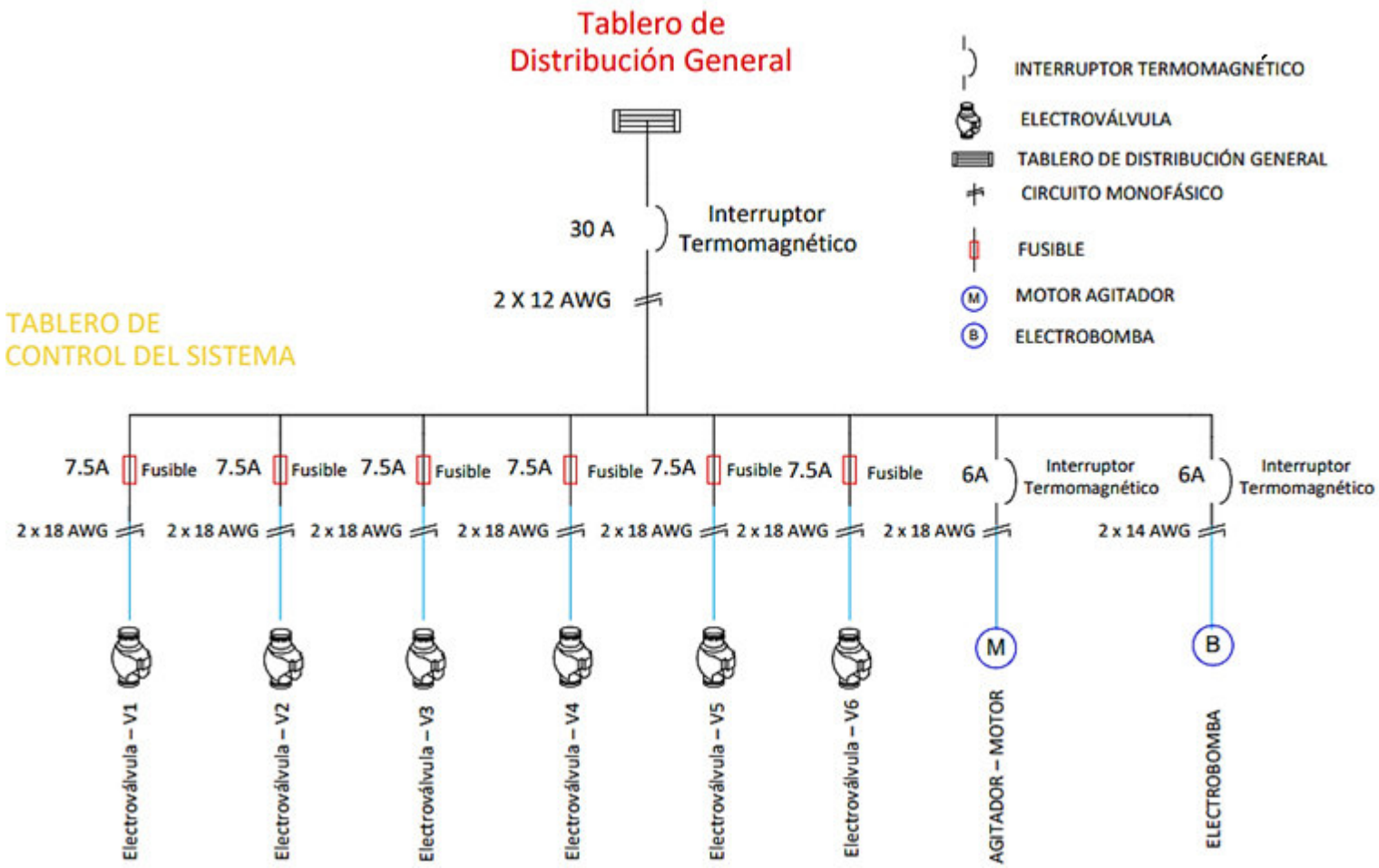

Figura 3.12 Diagrama Unifilar del sistema

3.2 Construcción del recipiente automatizado de dosificación y fumigación

Para el cumplimiento de los procesos de dosificación y fumigación se automatiza un recipiente para un llenado máximo de 210 litros. Se realizó un diseño previo al montaje de los componentes en el recipiente, como se muestra en la Figura 3.13.

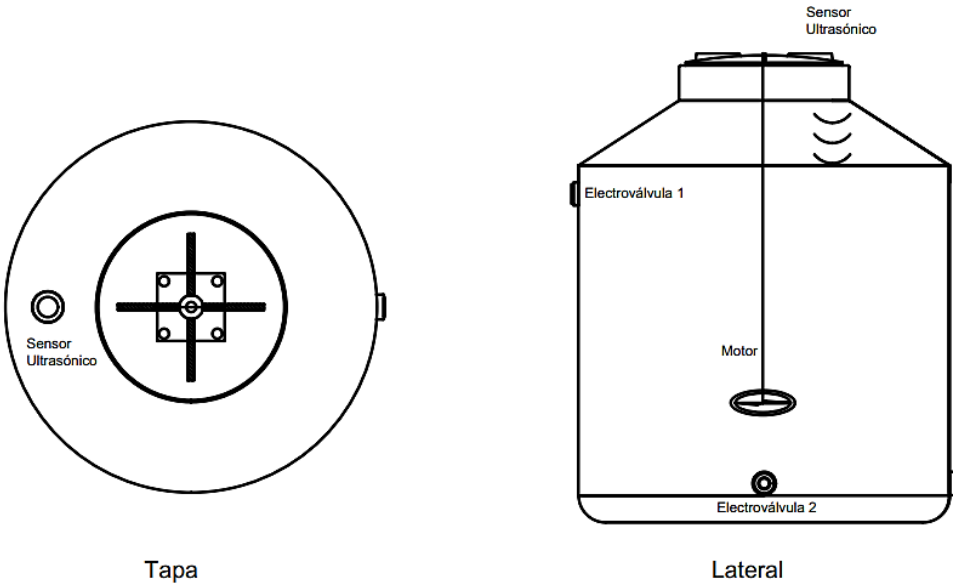

Figura 3.13 Ubicación de los componentes para la automatización del recipiente

Preparación para el acople de las tuberías al recipiente

En esta etapa se realizó las perforaciones para la ubicación de acoples a tuberías de ½ (in), las mismas que permiten el llenado del recipiente y la distribución hacia el invernadero. En la Figura 3.14 se visualiza el recipiente con sus acoples para las tuberías.

Figura 3.14 Acoplamiento de tuberías al recipiente

Instalación del motor

El motor es el encargado de realizar la mezcla de materia prima dentro del recipiente. Como se muestra en la Figura 3.15 el componente se encuentra ubicado en la parte superior céntrica, permitiendo de esta manera realizar una mezcla homogénea.

Figura 3.15 Montaje del motor al recipiente

Instalación del sensor ultrasónico AJ-SNR04M

Para la ubicación del sensor, el cual, permite medir el nivel de agua del recipiente se tuvo en cuenta el diámetro de las hélices de la barra de mezcla, para evitar que las señales ultrasónicas sean intervenidas por éstas y ocasione errores al medir el nivel. Luego de realizar varias pruebas de medición se instaló el sensor tal como se muestra en la Figura 3.16.

Figura 3.16 Montaje del sensor para medir el nivel

Instalación de las electroválvulas

Para la instalación de las electroválvulas se pusieron acoples de 1.5 (in) a 0.5 (in) en cada una de las tuberías, además, se conectó antes de cada electroválvula una válvula manual, cuyo objetivo es bloquear el paso de flujo de agua al invernadero en caso de que el sistema por algún motivo falle y requiera de una acción manual. En la Figura 3.17 se visualiza el resultado de la instalación de las electroválvulas.

Figura 3.17 Instalación de las electroválvulas a las tuberías

Para el cableado de cada una de las electroválvulas que se encuentran a la intemperie se usó ductos rígidos de PVC que por sus siglas se refiere a policloruro de vinilo. Las características que posee este material se destacan su ligereza, resistencia y no corrosión. El escenario de aplicación de estos ductos puede ser tanto en paredes, techos empotrados y también se puede colocar enterrados en el suelo incluso en ambientes húmedos [12].

Instalación del recipiente

Para la instalación del recipiente se tomaron factores como distancia del tablero de control, distancia de las tuberías del invernadero, y que se encuentre en una zona protegida de los rayos del sol y de las lluvias, debido a que, esto puede afectar a los dispositivos electrónicos. Una vez analizados los factores, se escogió el lugar adecuado para el recipiente, el cual, se encuentra cerca del tablero de control permitiendo de esta manera facilidades para el usuario a la hora de manipular el tablero y de insertar la materia prima para la dosificación y fumigación.

A continuación, en la Figura 3.18 se muestra el recipiente instalado.

Figura 3.18 Recipiente automatizado

3.3 Construcción del tablero de control del sistema

Para la construcción del tablero se tiene en cuenta cada uno de los componentes dimensionados en la etapa de requerimientos del sistema. Dentro del tablero se establece la distribución de la parte eléctrica, la cual corresponde a las protecciones del sistema y la parte electrónica que contiene al microcontrolador y módulos.

A continuación, en la Tabla 3.17 se muestra la lista de los componentes utilizados para la construcción del tablero de control.

Tabla 3.17 Componentes para el Tablero de Control [26]

Componente	Cantidad	Modelo-Marca	Especificaciones
Interruptor *On - Off*	1		30 (A)
Barras de Distribución	1		Fase - Neutro
Breaker	2		127 (V_{AC}) y 10 (A)
Fusibles	6		10 (A)
Fuente Industrial	1		120 a 12 (V_{DC}) y 30 (A)
Reductores V_{DC}	3		40 a 0 (V_{DC})
Módulo Arduino	1	Mega 2560	V_{in} = 12 (V_{DC})
Módulo de Relés 1 x 8	1		5 (V_{DC}) y 10 (A)
Módulo RTC	1	DS3231	3.3 a 5 (V_{DC})
Módulo WiFi	1	ESP8266MOD	3.6 (V_{DC})
Pantalla LCD	1	20 x 4	5 (V_{DC})

Selección del Gabinete

Teniendo en cuenta el número de componentes a instalar y sus dimensiones, se seleccionó un gabinete de doble fondo, cuyas dimensiones son: 60 x 40 x 20 (cm) correspondientes al largo, ancho y profundidad. En la Figura 3.19 se representa de manera gráfica las dimensiones del gabinete.

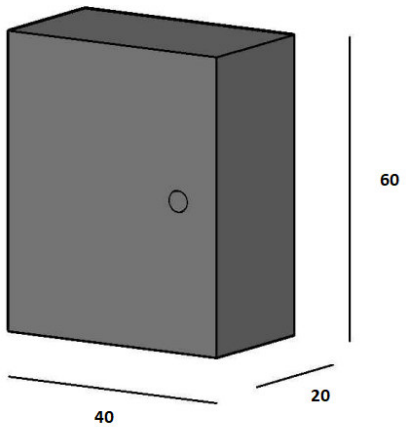

Figura 3.19 Dimensiones del gabinete para el proyecto

Montaje de los componentes al tablero de control

Los componentes fueron ubicados en el tablero de tal manera que las conexiones de cada componente sean establecidas de manera correcta. Para el cableado se utilizó canaletas ranuradas y se comprobó con un multímetro la continuidad de cada conexión. En la Figura 3.20 se muestra el tablero de control finalizado.

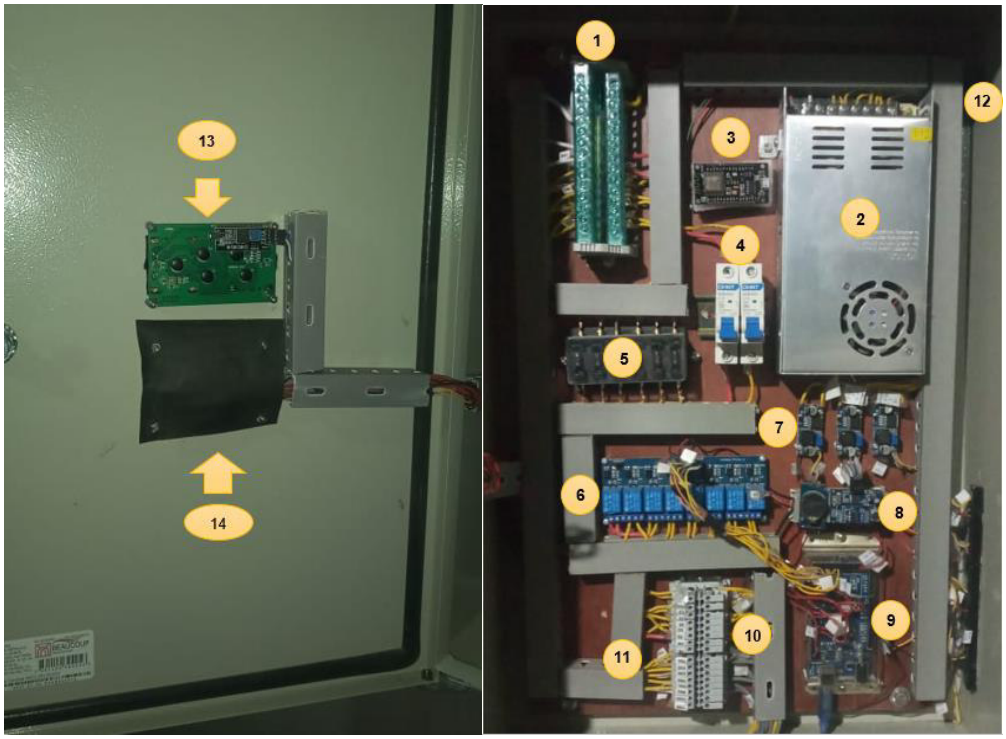

Figura 3.20 Tablero de control finalizado

1. Regleta de distribución Fase - Neutro
2. Fuente Industrial de 120 (V_{AC}) – 12 (V_{DC})
3. Módulo ESP8266 WiFi
4. *Breakers* Termomagnético 6 (A)
5. Fusibles
6. Módulo Relés 1 x 8
7. Reductores de Voltaje DC de 40 (V_{DC}) a 0 (V_{DC})
8. Módulo DS3231 RTC
9. Módulo Arduino Mega 2560
10. Borneras Industriales
11. Canaletas Ranuradas 25 x 25 (cm)
12. Interruptor *ON - OFF*
13. LCD Pantalla 20 x 4
14. Pulsadores

Diagrama de conexiones de componentes del tablero de control

El diagrama de conexiones de los componentes eléctricos del tablero se diseñó mediante el programa AutoCAD, con el objetivo de contar con una guía de los puntos de conexión de cada dispositivo, siendo así, información fundamental tanto para el

mantenimiento como para la reparación del tablero. A continuación, en la Figura 3.21 se muestra el cableado eléctrico interno de cada elemento del tablero.

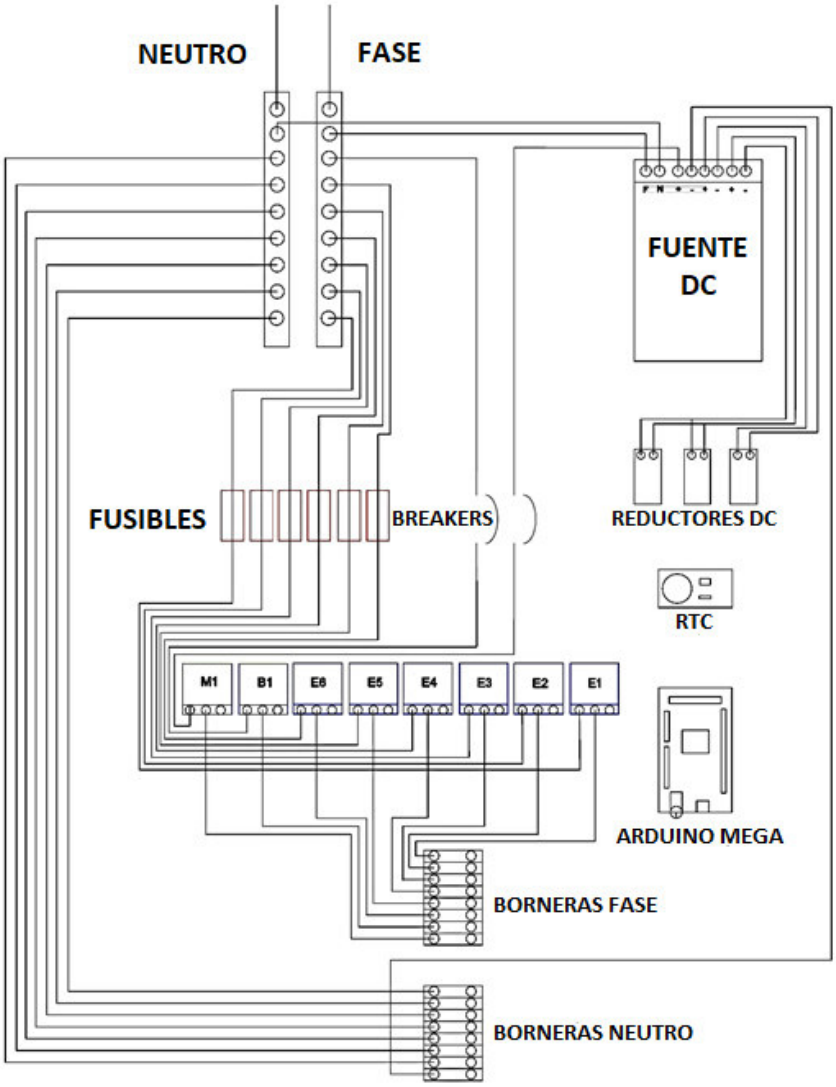

Figura 3.21 Diagrama de conexiones del tablero de control

Dentro de un tablero de control se debe realizar el etiquetado de los cables con la finalidad de poder ubicar, reparar o sustituir algún conductor o dispositivo [13]. Para etiquetar los cables se tuvo en cuenta factores como:

- **Identificación alfanumérica. –** Se debe llevar un orden lógico y firme de tal manera que sea fácil de ser interpretado.

- **Etiquetado en los dos extremos del cable. –** Permite que los cambios o interacciones con el cable sea más sencilla, de esta manera se reduce la probabilidad de cometer algún error o incidente.
- **Ubicación. –** La etiqueta debe estar situada de tal manera que sea visible y fácil de leer, además, debe estar bien adherida garantizando su vida útil.

A continuación, en la Figura 3.22 se muestra el proceso de etiquetado de los conductores y dispositivos. Para la identificación de las electroválvulas se utilizó la letra E seguido del número correspondiente a cada una de ellas. Así mismo, se utilizó la letra M para identificar al motor y la letra B para la electrobomba. Finalmente, para ubicar los neutros de cada componente se estableció la letra N, como por ejemplo EN1 que corresponde al neutro de la electroválvula 1.

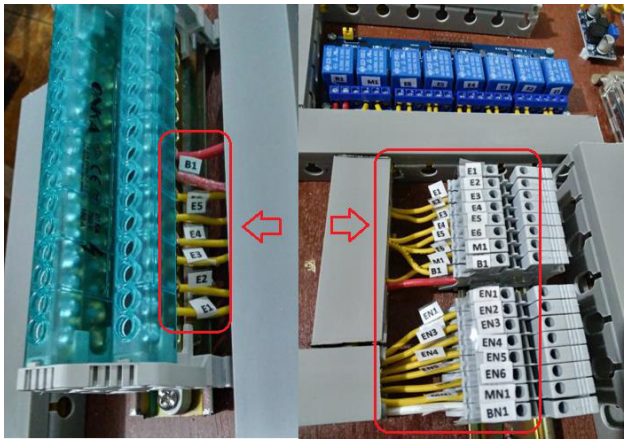

Figura 3.22 Etiquetado de cables

En la parte delantera del tablero de control se ubicó una pantalla LCD de 20 x 4, en la cual se mostrarán los datos al usuario para el proceso de riego, dosificación y fumigación. En la parte inferior de la pantalla LCD se ubicaron 5 pulsadores para la manipulación de la interfaz del sistema. En la Figura 3.23 se muestra el tablero con la pantalla LCD.

Figura 3.23 Parte delantera del tablero de control

A continuación, en la Figura 3.24 se muestra la distribución de los pulsadores instalados en el Tablero.

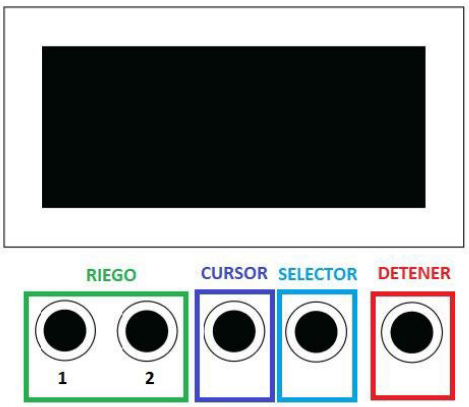

Figura 3.24 Distribución de pulsadores en el tablero de control

Las funciones de cada pulsador se detallan a continuación:

- **Riego 1.-** Su función es ejecutar el proceso de riego programado, esto quiere decir que, al ser presionado dicho pulsador, el sistema activa de manera automática la función de riego todos los días a una hora establecida ya en el programa. Cuando se activa el riego programado se muestra una letra en la pantalla indicando que la función ha sido activada. Para desactivar esta funcionalidad simplemente se pulsa nuevamente el botón y en la pantalla desaparece la letra indicando que ha sido desactivada la función de riego programado.

- **Riego 2.-** Su función permite realizar el riego del invernadero al instante que el usuario pulsa este botón, es decir, el riego se habilitará por un periodo de tiempo.
- **Cursor. -** Permite desplazarse mediante las opciones de los menús presentados en el LCD.
- **Selector. -** Como su nombre lo indica cumple la función de seleccionar las opciones presentadas en la pantalla.
- **Detener. -** Desactiva todos los procesos que el sistema se encuentre ejecutando.

Una vez realizado el cableado de la parte eléctrica del tablero de control se procedió a realizar la conexión de los módulos electrónicos al microcontrolador Arduino. Para esta etapa se diseñó un esquema de conexión de pines mediante el programa AutoCAD. En la Figura 3.25 se muestra la distribución de pines del microcontrolador.

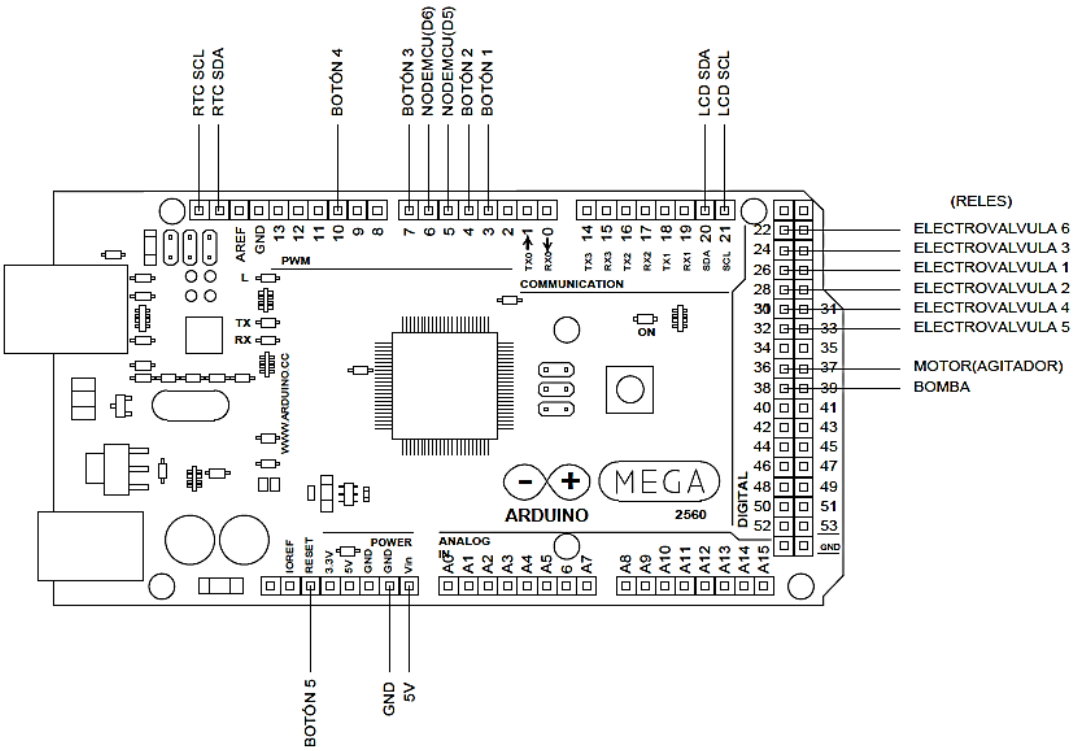

Figura 3.25 Distribución de pines del Arduino Mega

De igual manera se realiza la distribución de pines del módulo ESP8266 como se visualiza en la Figura 3.26**.**

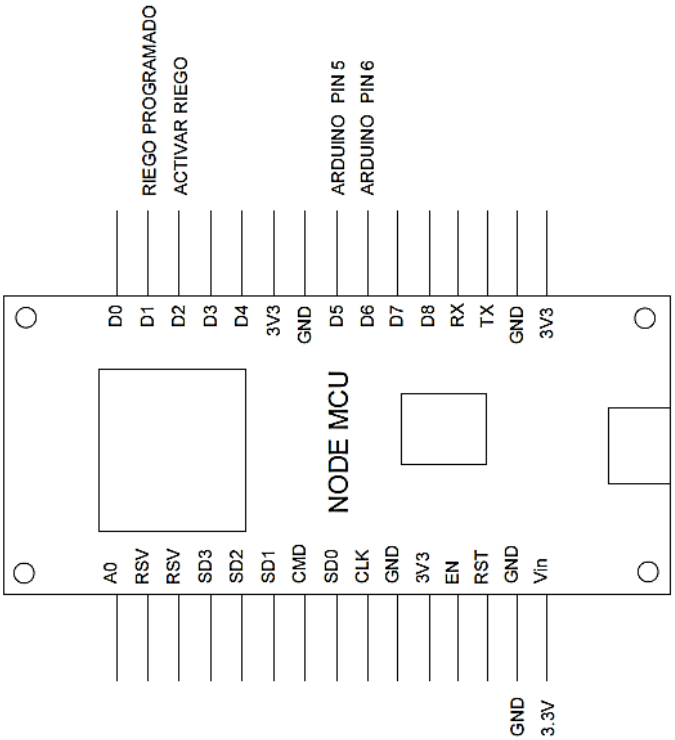

Figura 3.26 Distribución de pines del módulo ESP8266

3.4 Control del recipiente automatizado mediante un microcontrolador

Durante esta etapa se procedió a realizar un diagrama de flujo para el control del sistema, para luego realizar el algoritmo mediante la herramienta Arduino IDE. En la Figura 3.27 se muestra el diagrama de flujo de control del sistema implementado.

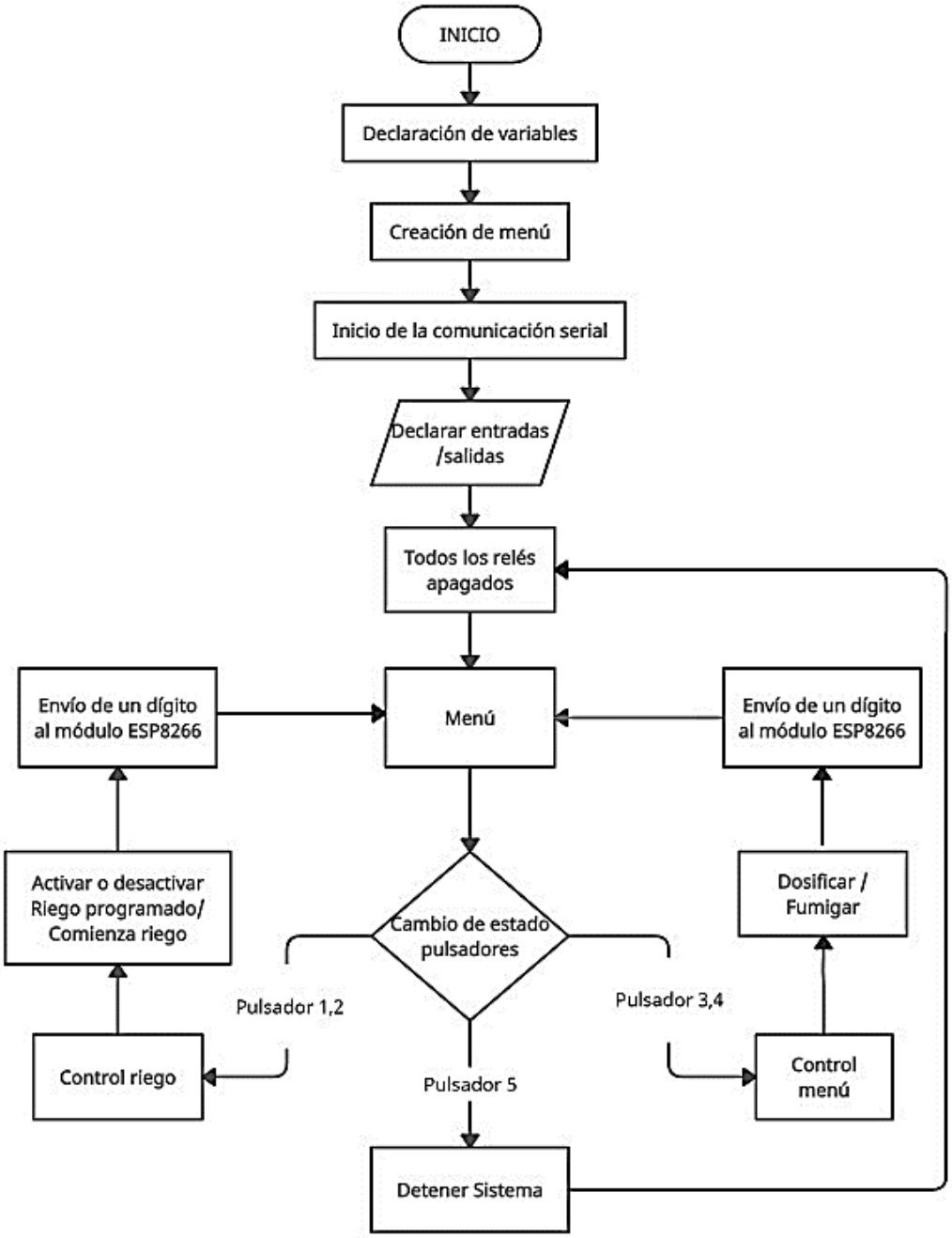

Figura 3.27 Diagrama de flujo del programa de control

Programación de los menús del sistema

Para la creación del menú se utilizó la librería LiquidMenu.h y se definieron 2 submenús. A continuación, en la Figura 3.28 se muestra el código de programación de cada uno de los menús.

```
// Pantalla 1
LiquidLine line1_1(0, 0, "R");
LiquidLine line3_1(1, 2, "Dosificar");
LiquidLine line4_1(1, 3, "Fumigar");
LiquidScreen screenlcd1(line1_1, line3_1, line4_1);

// Pantalla 2
LiquidLine line1_2(1, 0, "     TANQUE");
LiquidLine line2_2(1, 1, "150 l");
LiquidLine line3_2(1, 2, "180 l");
LiquidLine line4_2(1, 3, "210 l");
LiquidScreen screenlcd2(line1_2, line2_2, line3_2, line4_2);

// Pantalla 3
LiquidLine line1_3(1, 0, "     Mezcla");
LiquidLine line2_3(1, 1, "OK");
LiquidLine line3_3(1, 2, "INGRESE");
LiquidLine line4_3(1, 3, "NUTRIENTE");
LiquidScreen screenlcd3(line1_3, line2_3, line3_3, line4_3);
LiquidMenu menu(lcd, screenlcd1, screenlcd2, screenlcd3);
```

Figura 3.28 Creación de menú

Como se muestra en la Figura 3.28 se imprime en cada línea de la pantalla LCD las opciones que tiene el usuario.

Para desplazarse por las opciones se utilizó una función de la librería LiquidMenu.h, la cual permite crear mediante código de programación un cursor como se indica en la Figura 3.29.

```
line3_1.set_focusPosition(Position::LEFT);
line4_1.set_focusPosition(Position::LEFT);
line3_1.attach_function(1, fn_cantidadLitrosDosificar);
line4_1.attach_function(1, fn_cantidadLitrosFumigar);
menu.add_screen(screenlcd1);
```

Figura 3.29 Creación de cursor de desplazamiento

A continuación, en la Figura 3.30 se visualiza el resultado del menú principal programado.

Figura 3.30 Representación del menú principal

El menú principal está conformado por:

- **R:** Es el indicador para el riego programado, si aparece en el LCD R (A) significa que el riego programado ha sido activado, caso contrario solo aparece R.
- **I:** Es el indicador de la cantidad de litros que contiene el recipiente.
- **Dosificar:** Función para realizar el proceso de dosificación de nutrientes al invernadero.
- **Fumigar:** Función que realiza la preparación del pesticida para la fumigación.
- **Fecha y Hora:** Se encuentra establecida en el formato (Año, Mes, Día) y (Hora, Minutos, Segundos).

Cuando el usuario selecciona el riego programado y éste se activa de acuerdo a la hora establecida, se imprime en la pantalla un mensaje de ejecución de la actividad como se muestra en la Figura 3.31**.**

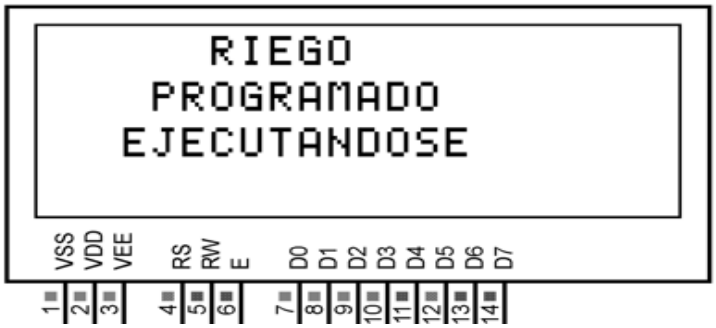

Figura 3.31 Impresión de la actividad de riego en ejecución

De igual manera cuando se presione el botón de riego al instante, se imprime en el LCD la ejecución de la actividad como se muestra en la Figura 3.32.

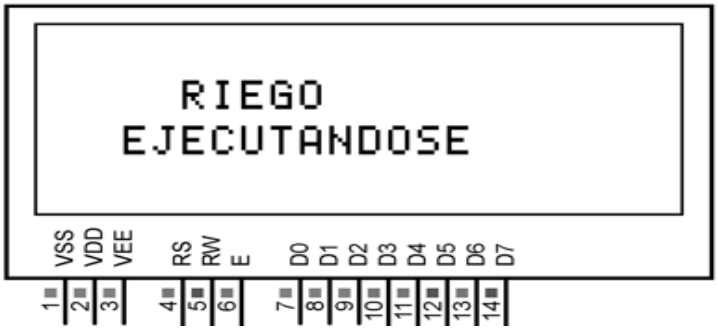

Figura 3.32 Riego al instante ejecutándose

La segunda pantalla del menú que se programó muestra las opciones de los niveles de agua a ser llenados, como se observa en la Figura 3.33.

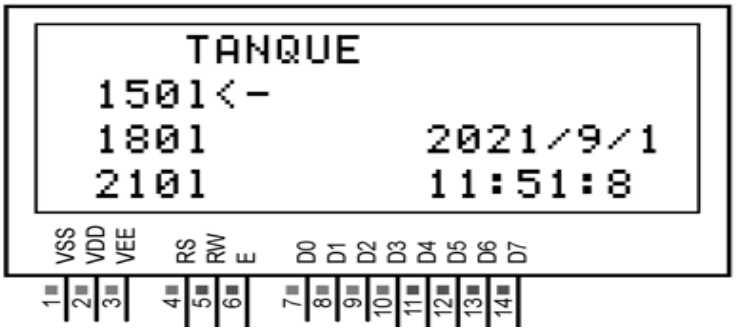

Figura 3.33 Menú de selección de nivel

La tercera pantalla del menú indica cuando se puede ingresar el nutriente para su posterior mezcla y distribución hacia el invernadero y se muestra en la Figura 3.34.

Figura 3.34 Menú ingresar nutriente

Como se puede observar en cada uno de los menús presentados se programó un cursor para desplazarse entre las opciones.

Programación del sensor ultrasónico en el Arduino IDE

A continuación, se muestra el proceso que se realizó para crear la fórmula que permite medir el nivel del agua en litros.

En primer lugar, se tiene un esquema de mediciones del recipiente como se muestra en la Figura 3.35**.** Estos datos son fundamentales para la calibración del sensor.

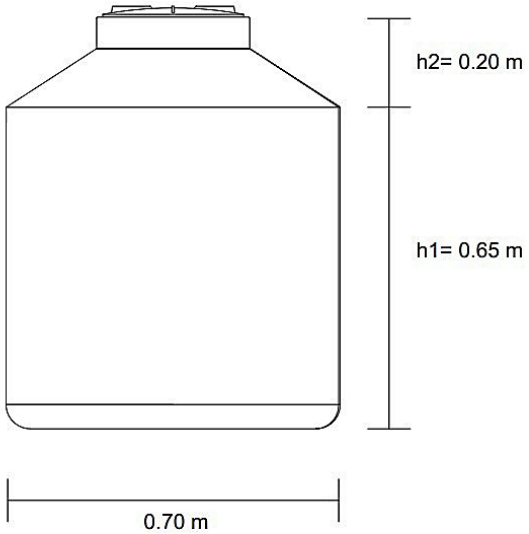

Figura 3.35 Dimensiones del recipiente

Como se observa en la Figura 3.35**,** se cuenta con una altura h1 y h2, las cuales, sumadas se obtiene la altura total del recipiente.

$$\mathrm{Htotal} = \mathrm{h1} + \mathrm{h2}$$

Ecuación 3.10 Altura del envase

Htotal : Altura total del recipiente
$\mathrm{h1}$: (cm) Altura máximo de llenado del tanque
$\mathrm{h2}$: (cm) Altura de ubicación del sensor (tapa)

Usando la ecuación 3.10 se obtiene:

$$\mathrm{Htotal} = 0.85\ (\mathrm{m})$$

Una vez obtenida la altura total del recipiente se procedió a calcular la distancia mediante el sensor ultrasónico verificando que imprima la misma altura. En consecuencia, esto indica que el tanque se encuentra vacío. Si se realiza un análisis se

puede observar en la Figura 3.36 que cuando el sensor imprima una altura menor a 0.85 (m) se afirma que el recipiente contiene un nivel de agua.

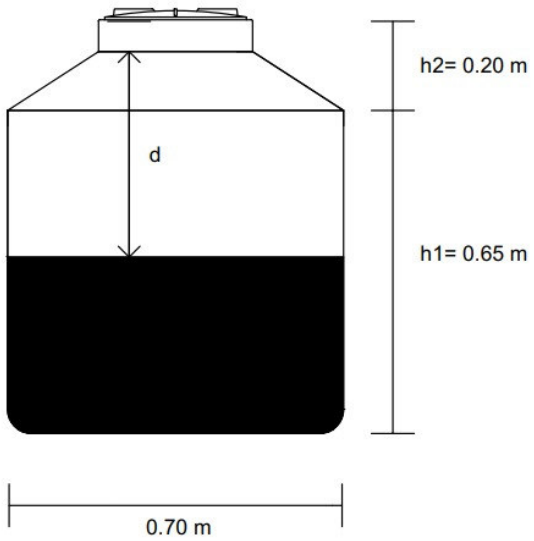

Figura 3.36 Recipiente lleno a la mitad

Para hallar el nivel en litros se utiliza la distancia que imprime el sensor ultrasónico para luego reemplazar este dato en la fórmula de volumen de un cilindro. En primera instancia se tiene que h1 es el nivel máximo de llenado el tanque y la h2 es la altura del sensor con respecto al nivel máximo del tanque. Por tal motivo, cuando el sensor mida la distancia y ésta sea igual a h2 se tiene que el tanque está lleno.

$$\text{Distancia medida por el sensor} = \text{h2}$$

Ecuación 3.11 Tanque a su nivel máximo de llenado

Para hallar el nivel en litros máximo se utiliza la fórmula de volumen de un cilindro. A continuación, se muestra el cálculo en litros del volumen máximo de llenado del recipiente.

$$\text{Vcilindro} = \pi \cdot r^2 \cdot h1$$

Ecuación 3.12 Volumen de un cilindro [33]

Donde:

π : pi

r : 0.35 (m) radio

$h1$: 0.65 (m) Altura máxima de llenado

Usando la Ecuación 3.12 se obtiene:

$$\text{Vcilindro} = 0.250\ (\text{m}^3)$$

A continuación, se hace la conversión de m^3 a litros.

$$1\ \text{m}^3\ = 1000\ \text{litros}$$

Ecuación 3.13 Conversión de unidades de volumen [34]

Por lo tanto, usando la Ecuación 3.13 se tiene:

$$\text{Vcilindro} = \frac{0.250\ \cdot\ 1000\ \text{l}}{1\ \text{m}^3}$$

Ecuación 3.14 Volumen del recipiente en litros

Usando la Ecuación 3.14 se obtiene:

$$\text{Vcilindro} = 250\ (\text{l})$$

Para hallar una ecuación general para el cálculo de volumen se basa en la Figura 3.37.

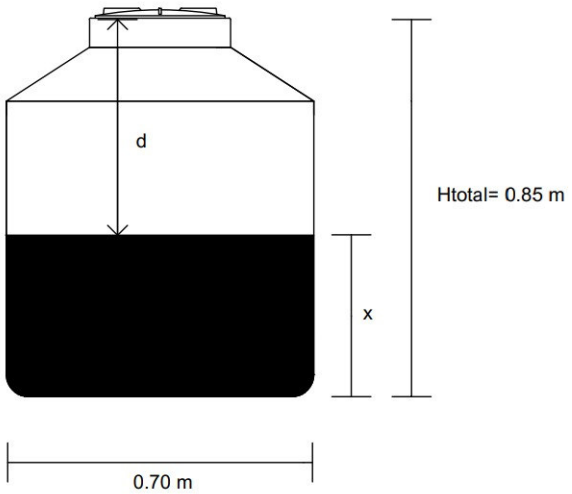

Figura 3.37 Gráfica para hallar volumen del recipiente

Según el gráfico se tiene:

$$\mathrm{Htotal} = \mathrm{x} + \mathrm{d}$$

Ecuación 3.15 Altura del recipiente

Donde:

Htotal	:	0.85 (m) Altura del recipiente
d	:	(m) Distancia medida por el sensor
x	:	Altura a la que se encuentra el nivel del agua

Se despeja x:

$$\mathrm{x} = \mathrm{Htotal} - \mathrm{d}$$

Ecuación 3.16 Altura a la que se encuentra el nivel

Se sustituye la ecuación 3.16 en 3.17:

$$\mathrm{Vcilindro} = \pi \cdot \mathrm{r}^2 \cdot \mathrm{x}$$

Ecuación 3.17 Volumen del recipiente con respecto a la altura del agua [33]

Donde:

π	:	pi
r	:	0.35 (m) radio
x	:	Altura a la que se encuentra el nivel del agua

Finalmente, se tiene la ecuación para medir el nivel de agua en litros.

$$\mathrm{Vcilindro} = \pi \cdot \mathrm{r}^2 \cdot (\mathrm{Htotal} - \mathrm{d})$$

Ecuación 3.18 Fórmula para calcular nivel del recipiente

Una vez hallada la fórmula para el cálculo del nivel de agua se procede a configurar en el algoritmo y cargar al microcontrolador Arduino como se muestra en la Figura 3.38.

```
// Conversión a litros
volumenRecipiente = (0.363168 * (0.79 - (distanciaRecipienteMetros)));
litrosRecipiente = volumenRecipiente * 1000;
Serial.println(litrosRecipiente);
Serial.println(distanciaCM);
```

Figura 3.38 Algoritmo para hallar la cantidad de litros

Por consiguiente, se realiza las respectivas pruebas de nivel y se verifica mediante la pantalla LCD la impresión de la cantidad de litros como se muestra en la Figura 3.39**.**

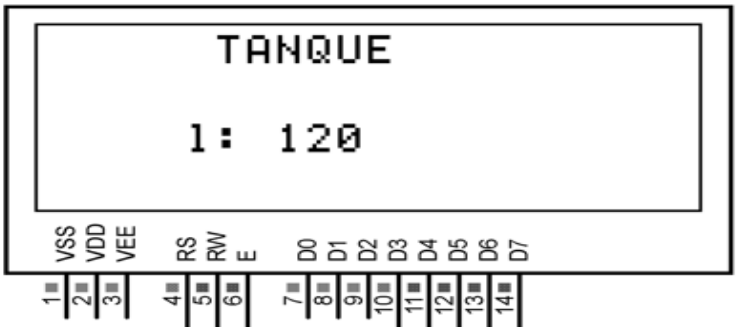

Figura 3.39 Impresión de la cantidad de litros del recipiente

Luego de que se realizó la calibración del sensor ultrasónico se procedió a realizar la etapa de control de cada actividad mostrada en el menú, así como también, de las funciones de riego programado y riego instantáneo. Para el control de cada actividad se activó cada componente mediante relés como: electroválvulas, motor DC y bomba en un orden secuencial de acuerdo con cada función. En la Figura 3.40 se visualiza la activación de los relés mediante el microcontrolador Arduino Mega.

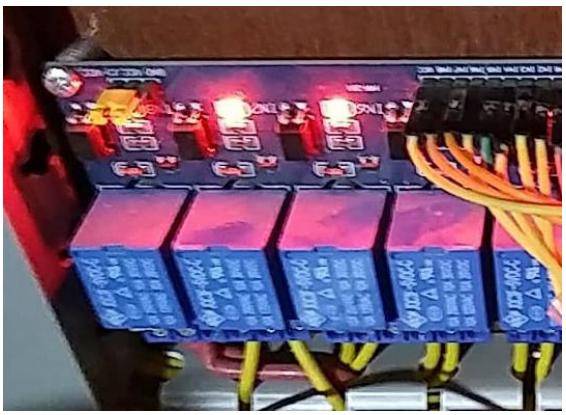

Figura 3.40 Activación de cada componente mediante Relés

A continuación, en la Figura 3.41 se muestra el diagrama de flujo del algoritmo de control para la vinculación del módulo WiFi al sistema.

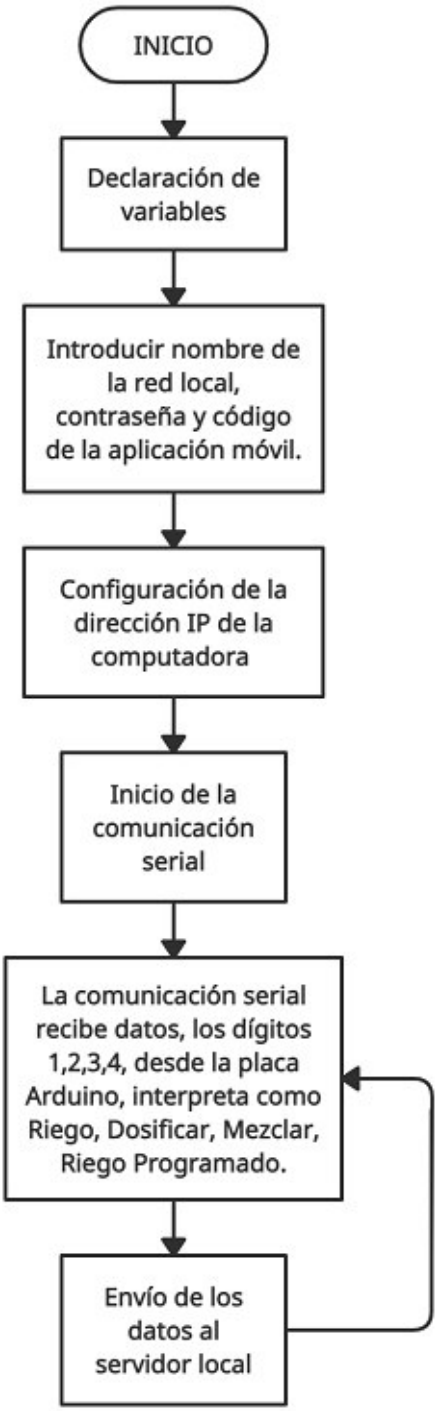

Figura 3.41 Diagrama de flujo del algoritmo de control del módulo ESP8266

Creación de las tablas para el registro de las actividades

Para la elaboración de las tablas que obtendrán la información del registro de las actividades ejecutadas por el microcontrolador, se utiliza dos programas XAMPP y HeidiSQL. El primer programa se utiliza para activar el servidor *web* Apache, el cual, contiene el protocolo HTTP para poder enviar páginas HTML o lo que se conoce como

hipertextos [18]. Además, mediante XAMPP también se activa el gestor de base de datos MySQL. En la Figura 3.42 se puede visualizar la herramienta XAMPP.

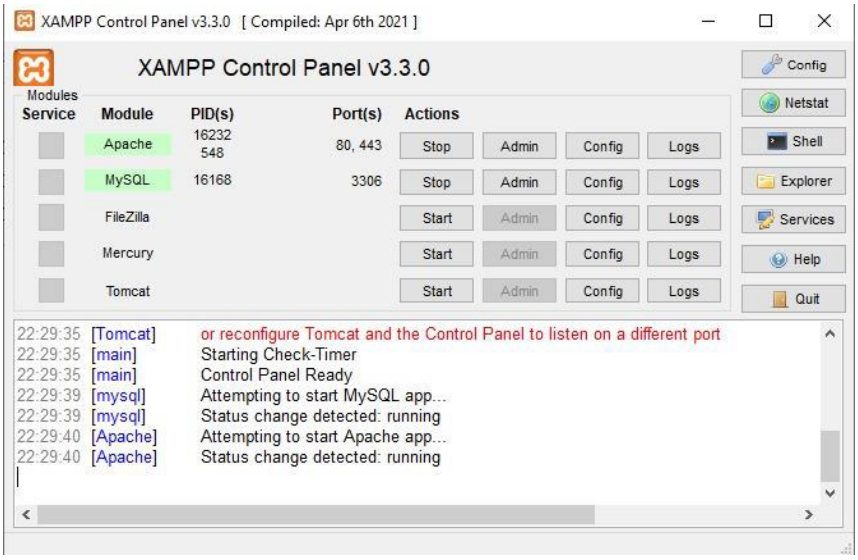

Figura 3.42 Herramienta XAMPP

Por otro lado, se utilizó el programa Heidi para la creación de las tablas de una manera sencilla y rápida como se muestra en la Figura 3.43.

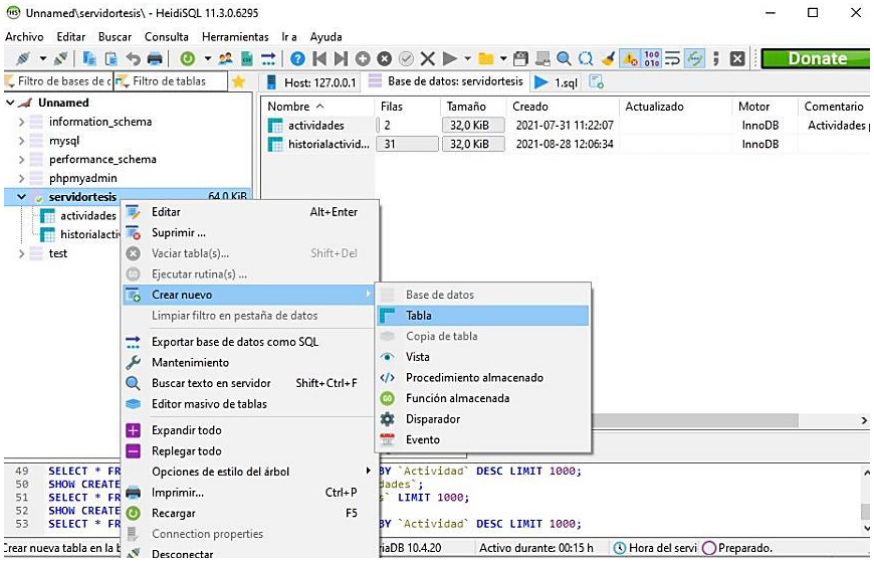

Figura 3.43 Crear tablas en Heidi

Para el proyecto se requieren dos ítems, el primer ítem indica el nombre del microcontrolador, el segundo ítem indica la actividad que se ejecutó. Además, se seleccionó el tipo de dato, debido a que se va a enviar caracteres se escogió la opción de VARCHAR. En la Figura 3.44 se muestra la configuración de las filas y columnas de la tabla.

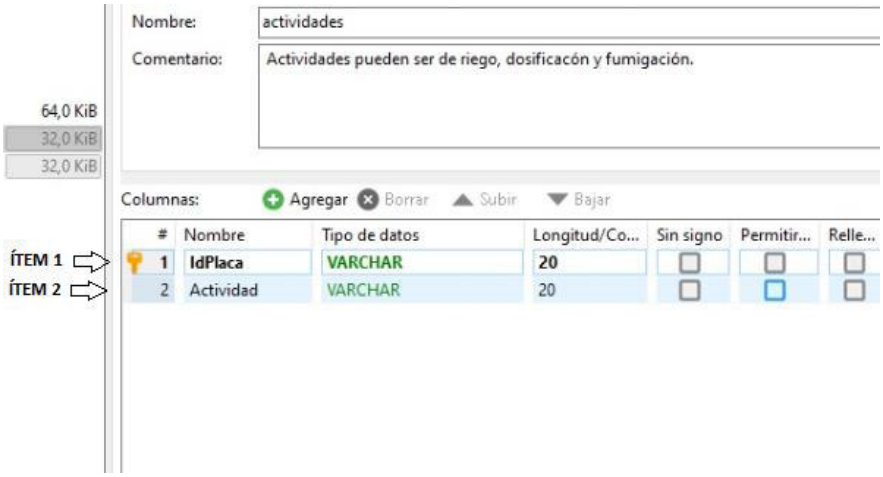

Figura 3.44 Configuración de la tabla en HeidiSQL

Se creó una siguiente tabla para el historial de actividades, en donde, se especificó la actividad y la fecha y hora que se ejecutó. En la Figura 3.45 se muestra la configuración de la tabla de historial de actividades.

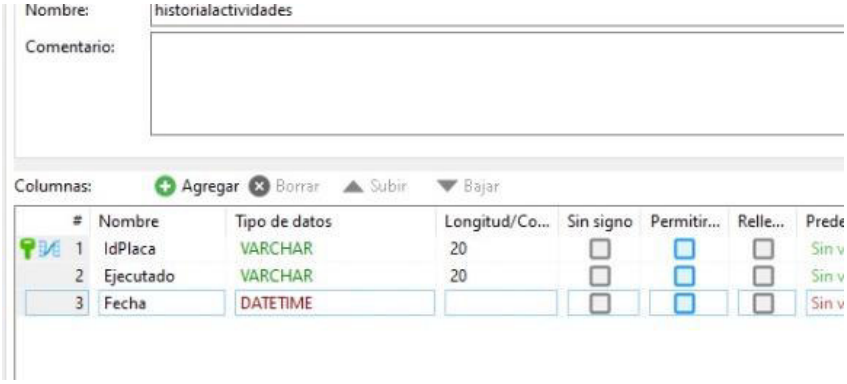

Figura 3.45 Configuración de tabla para historial de actividades

Comunicación entre el tablero y base de datos

Para enviar los datos generados por las actividades del microcontrolador hacia la base de datos se realiza el siguiente proceso:

- Se configura el nodeMCU para que se conecte a la red local y de esta manera se establezca la comunicación con el ordenador que también se encuentra en la misma red como se visualiza en la Figura 3.46.

```
char ssid[] = "TELPRONET_CAROLINA";
char pass[] = "curay1980";
```

Figura 3.46 Configuración del nodeMCU a la red local

- A continuación, se transmite los datos por medio de la comunicación serial existente entre el Arduino y el nodeMCU. Los datos que se envían son numéricos dependiendo de la actividad, por ejemplo: Si se ejecutó el riego envía el 1 como se muestra en la Figura 3.47

```
void ActEnviarRiego() {
  h = 1;
  Serial.print("Enviado: ");
  Serial.println(h);
}
void ActEnviarDosificar() {
  h = 2;
  Serial.print("Enviado: ");
  Serial.println(h);
}
void ActEnviarMezclarListo() {
  h = 3;
  Serial.print("Enviado: ");
  Serial.println(h);
}
```

Figura 3.47 Envío de dígito de Arduino a NodeMCU

- Seguidamente, el módulo WiFi asigna a cada dígito recibido una función que puede ser: riego, dosificar, o mezclar como se muestra en la Figura 3.48. Estos caracteres son enviados hacia la base de datos local mediante la dirección IP del computador del usuario.

```
h = doc["send"];
switch(h){
 case 1:
  a="Riego";
break;
case 2:
  a="Dosificar";
break;
case 3:
  a="Mezclar";
break;
```

Figura 3.48 NodeMCU recibe dígito de Arduino mediante comunicación serial

- Finalmente, para el registro en la base de datos se utilizó dos archivos en PHP. En el primer archivo se indica la ruta en donde se guardan los datos como se observa en la Figura 3.49**.**

```
<?php
class conexion {

    const user ='root'; // Se da la dreccion de la base de datos
    const pass ='';
    const bdatos ='servidortesis';
    const servidor ='localhost';


    public function conectarbdatos(){
        $conectar = new mysqli(self::servidor,self::user,self::pass,self::bdatos);
        if($conectar->connect_error){

            die("Error".$conectar->connect_error);

        }

        return $conectar;
    }
}
```

Figura 3.49 Archivo conexión PHP

- El segundo archivo PHP recibe los datos del nodeMCU y los envía a la dirección configurada anteriormente en el archivo conexión PHP. En la Figura 3.50 se muestra el código de programación PHP.

```
<?php

    require_once('conectado.php'); // Indica el codigo de conectado.php


    $actividadRealizar=$_GET['ActRealizar']; // Espera para almacenar
    //la variable que se recibe del ESP8266

if($actividadRealizar=='Riego' || $actividadRealizar=='Dosificar'
||$actividadRealizar=='Mezclar' ||$actividadRealizar=='RiegoProgramado'){
    // Si la palabra recibida es la que se encuntra en las opnes
    //procede a enviarla a la base de datos
```

Figura 3.50 Código PHP recibo de datos

A continuación, en la Figura 3.51 se muestra el esquema de comunicación.

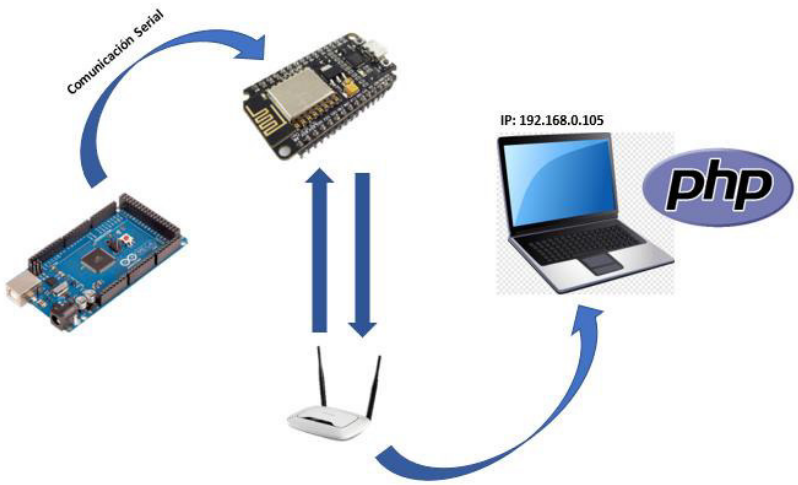

Figura 3.51 Esquema de comunicación entre tablero y base de datos

Comunicación entre la aplicación y el NodeMCU.

Para este fin se utilizó la aplicación Blynk en donde se especifica el módulo ESP8266. Para la configuración de la aplicación se creó un botón para el control del riego especificando el pin del módulo WiFi que activa la función de riego. Una vez programado el pulsador se genera un código, el cual, se carga al algoritmo del nodeMCU para de esta manera vincular la aplicación con el módulo WiFi. En la Figura 3.47 se muestran los botones que permiten activar la función de riego al instante y riego programado.

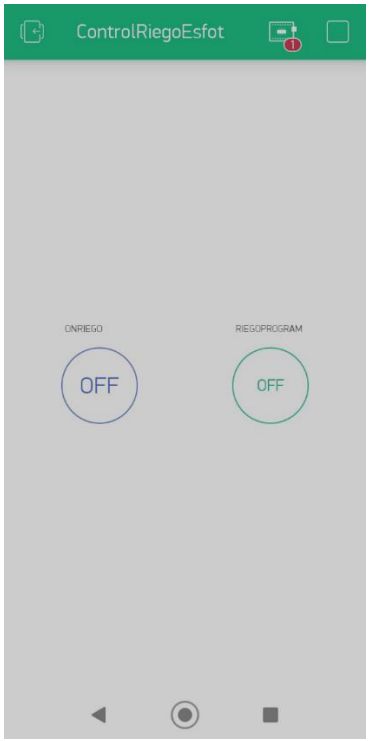

Figura 3.52 Control del riego mediante Blynk

3.5 Pruebas y Análisis de Resultados

Pruebas de continuidad en pulsadores

Los pulsadores son elementos que están en constante interacción con el usuario, por tal motivo se verificó que éstos se encuentren bien adheridos al tablero y que las conexiones se encuentren en los pines asignados del microcontrolador. En la Tabla 3.19 se muestran los resultados de las pruebas.

Tabla 3.18 Pruebas de continuidad en pulsadores

Pulsador	Aprueba	No aprueba
1	✓	-
2	✓	-
3	✓	-
4	✓	-
5	✓	-

Prueba de alimentación de los módulos.

La Placa Arduino Mega y cada módulo que se instaló tiene una alimentación de voltaje DC diferente, que puede variar entre 3.3 (V) a 5 (V). Estos dispositivos están alimentados por una fuente externa regulada con módulo reductor DC-DC de Arduino.

Se utilizó un multímetro para medir que el voltaje entregado por la fuente DC sea de 15 (V). En la Tabla 3.20 se muestran los resultados de las pruebas realizadas.

Tabla 3.19 Valores medidos del voltaje de los dispositivos electrónicos

Dispositivo electrónico	Voltaje especificado (V)	Voltaje medido (V)	Aprueba	No aprueba
Arduino Mega	5	5.1	✓	-
Módulo Relés 1 x 8	5	5.1	✓	-
Módulo DS3231 RTC	3.3	3.34	✓	-
LCD Pantalla 20 x 4	5	5.1	✓	-
Módulo ESP8266 WiFi	3.3	3.29	✓	-
Sensor ultrasónico	5	5.1	✓	-

Se aprobó los voltajes medidos en cada dispositivo debido a que se encuentran dentro del rango de variación del 2%.

Prueba de funcionamiento de relés

Se verificó mediante la ejecución de todas las actividades automatizadas la activación de los relés designados a cada componente del sistema. En la Tabla 3.21 se muestra el resultado de las pruebas.

Tabla 3.20 Funcionamiento de relés

Dispositivo electrónico	Riego	Dosificar	Mezcla	Aprueba	No aprueba
Electroválvula 1	-	✓	✓	✓	-
Electroválvula 2	-	✓	✓	✓	-
Electroválvula 3	-	✓	✓	✓	-
Electroválvula 4	✓	✓	-	✓	-
Electroválvula 5	✓	✓	-	✓	-

Dispositivo electrónico	Riego	Dosificar	Mezcla	Aprueba	No aprueba
Electroválvula 6	✓	✓	-	✓	-
Motor	-	✓	✓	✓	-
Bomba	-	✓		✓	-

Prueba de voltaje en las borneras

Se realizó pruebas de voltaje en las borneras correspondiente a cada componente para asegurar que se encuentren con la alimentación requerida. En la Tabla 3.22 se enlista los voltajes medidos.

Tabla 3.21 Valores de voltajes medidos en las borneras de cada componente

Dispositivo electrónico	Voltaje de operación (V)	Voltaje medido (V)	Aprueba	No aprueba
Electroválvula 1	100-120	117.5	✓	-
Electroválvula 2	100-120	117.5	✓	-
Electroválvula 3	100-120	117.5	✓	-
Electroválvula 4	100-120	117.5	✓	-
Electroválvula 5	100-120	117.5	✓	-
Electroválvula 6	100-120	117.5	✓	-
Motor DC	12-24	15	✓	-
Bomba	100-120	117.5	✓	-

Los valores de voltaje medidos en las borneras se encuentran dentro del rango de operación de cada componente.

Calibración de tiempos del sistema

El sistema debe estar correctamente calibrado para que cumpla con los tiempos adecuados en cada uno de los procesos, y de esta manera, evitar afectaciones en el cultivo del invernadero.

A continuación, se detalla el procedimiento seguido para las pruebas que se realizaron para calibrar los tiempos de cada actividad:

- **Tiempo de riego**

  Para hallar el tiempo ideal para el riego del invernadero se procedió a activar las electroválvulas correspondientes a las tuberías de riego, mediante un cronómetro se empezó a medir el tiempo. Para conocer la humedad ideal del suelo se realizó el método del tacto, el cual, se toma un puño de tierra y se intenta hacer una bola compacta, si se presenta dificultad al hacer y se desmorona quiere decir que la tierra se encuentra muy seca [19]. De esta manera se halló el tiempo adecuado para el riego. En la Figura 3.53 se muestra un ejemplo del test del tacto para medir humedad del suelo.

Figura 3.53 Medición de la humedad [20]

- **Tiempo de dosificación y fumigación**

  Para esta etapa se realizaron pruebas con el motor encargado de realizar la mezcla de nutrientes en el recipiente con el objetivo de tener una disolución homogénea. Se procedió a llenar el tanque y se vertió la materia prima, seguidamente se activó el motor para calcular el tiempo hasta que la mezcla quede homogénea. De esta manera se configuraron los tiempos en el algoritmo de control. Igualmente, se realizó el mismo proceso en la etapa de fumigación.

En la Tabla 3.23 se presenta detallado los tiempos de ejecución de cada una de las actividades.

Tabla 3.22 Tiempos para ejecución de actividades

Actividad	Tiempo de activación
Riego	30 (min)
Mezclado de nutrientes	3 (min)
Dosificación	10 (min)

Pruebas de verificación del registro en la base datos

Para verificar que el tablero de control pueda realizar la comunicación con la base de datos se procedió a ejecutar el riego. Al finalizar se observó el registro de la actividad con la fecha y la hora. En la Figura 3.54 se muestran las pruebas realizadas.

IdPlaca	Ejecutado	Fecha
Arduino	Riego	2021-09-06 19:31:33
Arduino	Dosificar	2021-09-06 19:33:48
Arduino	Riego	2021-09-07 09:46:44
Arduino	Dosificar	2021-09-07 14:21:43
Arduino	Dosificar	2021-09-08 10:41:54
Arduino	Dosificar	2021-09-08 10:49:23
Arduino	Riego	2021-09-08 11:09:02
Arduino	Riego	2021-09-08 11:09:18
Arduino	Riego	2021-09-08 11:09:40

Figura 3.54 Registro de actividades ejecutadas

3.6 Manual de Uso y Mantenimiento

Video del funcionamiento del sistema

Video del mantenimiento del sistema

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Para la implementación del proyecto se realizó una visita al invernadero en donde se conoció las fases de crecimiento de las plantas y la importancia que conlleva realizar correctas actividades de riego, dosificación de nutrientes y fumigación. Este conocimiento fue fundamental para llevar a cabo el proceso de automatización de las actividades antes mencionadas.

- Se elaboró un esquema de todo el sistema, en donde consta el recipiente y la forma en que se encuentran distribuidas las tuberías en el invernadero, esto con el fin de dimensionar el número de electroválvulas y la ubicación de componentes como: sensor ultrasónico, electrobomba y motor agitador.

- Debido a las distancias a las que se encuentran instaladas las electroválvulas se calculó el calibre del cable para que garantice una caída máxima del 3% de voltaje, logrando de esta manera que los componentes reciban una alimentación adecuada y por ende operen con normalidad.

- Para el control de los componentes del sistema se implementó un tablero en donde se ubicaron las protecciones eléctricas, fuente de alimentación y módulos electrónicos, esto de acuerdo con un esquema diseñado previamente con el objetivo de facilitar las conexiones de cada dispositivo.

- Entre las normativas que se emplearon para el dimensionamiento de los conductores eléctricos están la Norma Ecuatoriana de la Construcción que establece los porcentajes máximos de caída de voltaje, los cuales ayudaron a realizar el cálculo de los calibres de los cables. Además, se utilizó el Código Eléctrico Nacional para dimensionar las protecciones eléctricas.

- Se realizó varias pruebas del sistema en donde cada una de las actividades automatizadas fueron calibradas con el tiempo necesario para garantizar el buen desarrollo de las plantas y de esta manera obtener productos de calidad.

- El usuario puede controlar todas las actividades del invernadero mediante pulsadores y una pantalla LCD ubicada en el tablero.

- La actividad de riego fue vinculada a la aplicación móvil Blynk, la cual establece una comunicación inalámbrica con el tablero, permitiendo de esta manera activar la función mencionada desde cualquier zona del invernadero.

- Mediante el módulo ESP8266 se estableció una conexión inalámbrica entre el tablero de control y el servidor local para el registro de las actividades ejecutadas en el invernadero.

- El proyecto fue implementado en Salcedo en la Comuna de Miranda y tuvo un costo aproximado de $750.

- El proyecto se entregó al usuario constatando el funcionamiento de todos los dispositivos y cumpliendo con los requisitos solicitados.

4.2 Recomendaciones

- Verificar que los dispositivos tengan un grado de protección IP de acuerdo a las condiciones del lugar donde serán instalados.

- Para la selección de los componentes se debe tener en cuenta que sus características se encuentren acorde al voltaje de alimentación que dispone el lugar de instalación.

- Para medir el nivel de líquidos mediante sensores ultrasónicos es preferible seleccionar un modelo que cuente con impermeabilidad, especialmente en la zona del sensor que se va a encontrar expuesta.

- Antes de energizar el sistema se debe verificar que los cables se encuentren bien sujetos y es recomendable realizar pruebas de continuidad en cada componente y comprobar que no exista ningún corto.

- Al momento de ejecutar cualquiera de las funciones programadas en el sistema se recomienda al usuario verificar que exista abastecimiento de agua en las tuberías para evitar que las electroválvulas operen sin líquido y se quemen.

- Para activar el riego mediante la aplicación móvil el usuario debe encontrarse en la zona de cobertura de la red local inalámbrica.

- Se debe realizar periódicamente una limpieza a los componentes debido a que pueden acumular polvo ocasionando fallas en el sistema o incluso que los dispositivos se dañen.

5 REFERENCIAS BIBLIOGRÁFICAS

[1] B. Sintes, «XAMPP,» 08 julio 2021. [En línea]. Available: https://www.mclibre.org/consultar/php/otros/xampp.html. [Último acceso: 12 08 2021].

[2] M. C. Server, *Manual Básico de MySQL,* México DF., 2018.

[3] Microsoft, «Visual Studio Code,» 2021. [En línea]. Available: https://code.visualstudio.com/docs. [Último acceso: 14 Agosto 2021].

[4] Embarcadero Delphi, «HeidiSQL,» 2010. [En línea]. Available: https://www.heidisql.com/. [Último acceso: 14 Agosto 2021].

[5] Frank de Brabander, *LiquidCrystal I2C,* Arduino.cc, 2021.

[6] V. Kalchev, *LiquidMenu,* Arduino.cc, 2016.

[7] Arduino, *Wire.h,* Creative Commons Attribution-ShareAlike, 2019.

[8] Adafruit, «RTClib,» Arduino Library, 2021.

[9] Arduino, *SoftwareSerial,* Creative Commons Attribution-ShareAlike, 2019.

[10] M. Gandhi, «¿Qué es un sensor ultrasónico?,» Autycom, 19 12 2019. [En línea]. Available: https://www.autycom.com/que-es-un-sensor-ultrasonico-y-para-que-sirve/. [Último acceso: 15 08 2021].

[11] Ingeniería Mantenimiento, «Gestión del Mantenimiento con tecnologías de control,» 12 junio 2014. [En línea]. Available: http://www.ingenieriamantenimiento.org/medida-de-distancia-por-ultrasonidos/. [Último acceso: 20 agosto 2021].

[12] D. d. v. J.-S. 2.0, «Unit Electronics,» [En línea]. Available: https://uelectronics.com/producto/sensor-ultrasonico-jns-sr04t/. [Último acceso: 15 08 2021].

[13] M. A. M. Redondo, «Fundamentos básicos de hidráulica,» 25 06 2018. [En línea]. Available: https://www.iagua.es/blogs/miguel-angel-monge-redondo/fundamentos-basicos-hidraulica-i. [Último acceso: 20 08 2021].

[14] P. Ecuador, Electrobombas, Quito: Ecuaimco, 2020.

[15] O. Electrónica, «Datasheet Motor DC,» Quito, 2021.

[16] B. Technology, «Módulo Inal{ambrico Esp8266,» 06 08 2021. [En línea]. Available: https://beeggle.com/producto/mdulo-inalmbrico-cp2102-nodemcu-lua-wifi-esp8266-esp-12f/. [Último acceso: 06 08 2021].

[17] E. Systems, «Datasheet ESP8266EX,» 2020. [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Último acceso: 24 08 2021].

[18] L. d. V. Hernández, «Automatización con ESP8266,» Programación Fácil, 2019.

[19] Espressif, *ESP8266EX,* Espressif Systems, 2020.

[20] Tecnopura, «Módulo RTC DS3231,» 18 04 2020. [En línea]. Available: https://www.tecnopura.com/producto/modulo-rtc-ds3231-interface-i2c-reloj-en-tiempo-real-compatible-arduino/. [Último acceso: 12 08 2021].

[21] Maximintegrated, «DS3231,» 2015. [En línea]. Available: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf. [Último acceso: 29 07 2021].

[22] O. Electronics, «Catálogo de Electroválvulas,» 2020.

[23] S. Electronics, «Arduino Mega 2560,» 2017. [En línea]. Available: https://sparkfruit.ph/product/arduino-mega-2560-r3-premuim/. [Último acceso: 19 08 2021].

[24] RobotShop, «Arduino Mega 2560 Datasheet,» 2020. [En línea]. Available: https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf. [Último acceso: 19 08 2021].

[25] Norma Ecuatoriana de la Construcción, *Instalaciones Electromecánicas,* miduvi, 2013.

[26] M. G. Barragan, «Ley de Ohm,» Guadalajara, 2019.

[27] R. Miguel, «Cálculo de líneas y redes eléctricas,» Edicions de la Universitat Politècnica de Catalunya, Barcelona, 2012.

[28] Electricasas, «Cables eléctricos y sus características,» 2019.

[29] NEC, *National Electrical Code,* 2017.

[30] Norma Ecuatoriana de la Construcción, *Instalaciones Eléctricas,* MIDUVI, 2018.

[31] Onulec, «Tipos de canalizaciones eléctricas,» 2021. [En línea]. Available: https://www.onulec.com/blog/novedades-onulec/185-tipos-de-canalizaciones-electricas. [Último acceso: 19 agosto 2021].

[32] V. Masapanta, «Mega Electrónica,» 2021.

[33] «Área y volumen de sólidos geométricos,» Santillana, Barcelona, 2016.

[34] B. Albarenga y A. Máximo, «Física General,» Harla, 2013.

[35] E. Mifsuf Talón, «Curso Mentor Apache,» Ministerio de Educaión Cultural, España, 2017.

[36] C. Muñoz y E. Martin, *Métodos para Medir la Humedad del Suelo para la Programación del Riego,* Tucson, AZ: The University of Arizona , 2017.

[37] L. d. California, «Medir la humedad del suelo al tacto,» 10 enero 2015. [En línea]. Available: http://lombricesdecalifornia.com/blog/medir-la-humedad-del-suelo-al-tacto/. [Último acceso: 20 08 2021].

[38] F. d. Brabander, «Librería LiquidCrystal_I2C,» 2015. [En línea]. Available: https://www.arduinolibraries.info/libraries/liquid-crystal-i2-c. [Último acceso: 26 08 2021].

ANEXOS

ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO

ESCUELA POLITÉCNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 05 de octubre de 2021

CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, Pablo Andrés Proaño Chamorro, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el correcto funcionamiento del recipiente automatizado para dosificación y fumigación para un invernadero ubicado en la comuna de miranda controlado por WiFi, los cuales fueron implementados por los estudiantes Chauca Jonathan y Tana Jefferson.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para la correcta ejecución de las actividades en el invernadero

DIRECTOR

Ing. Pablo Andrés Proaño Chamorro, Msc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 28. EXT: 2729
email: pablo.proano@epn.edu.ec Quito-Ecuador

ANEXO 2: CERTIFICADO DE FUNCIONAMIENTO DEL DUEÑO

CERTIFICADO DE FUNCIONAMIENTO

Yo, *Curay Taipe Luis Gerardo,* con numero de cedula 050275409-6, dueño del invernadero ubicado en la provincia de Cotopaxi en la parroquia de Salcedo, en la comuna de Miranda, certifico que he verificado el correcto funcionamiento del sistema automatizado, además, constatado que el equipo electrónico instalado funciona correctamente, los cuales fueron implementados por los estudiantes Chauca Jonathan y Tana Jefferson.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para ayuda al cuidado de todas las platas que se encuentran dentro del invernadero, además, evitar desperdicio de nutrientes para los cultivos.

DUEÑO

Curay Taipe Luis Gerardo

050275409-6

ANEXO 3: PRESENTACIÓN CODIGO DE ARDUINO

```
// Librerías requeridas
#include <LiquidCrystal_I2C.h> // Realizar la comunicación I2C para LCD
#include <LiquidMenu.h> // Realizar las pantallas de menús
#include <Wire.h> // por default en Arduino
#include "RTClib.h" // librería para RTC
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

LiquidCrystal_I2C lcd(0x27, 20, 4); // LCD especificando el tamaño
SoftwareSerial nodemcu(5, 6); // Se usa los pines para Rx y Tx del ESP8266
RTC_DS3231 RTC; // Se crea la clase para su uso

// Variable para trasmitir datos
int h; // Se guarda un número que se envía al ESP8266

// Variables del sensor ultrasónico
long duracionTiempo;
long distanciaCM; // distancia para obtener centímetros
inti echo = 52;
int trig = 50;
int nivelseleccion = 0;
int nivelenLitros = 0;
float volumenRecipiente;
float ultimaDistanciaCM;
int litrosRecipiente;
float distanciaRecipienteMetros;

// Variables uso para los botones almacenar el dato que se pulso
int btn1estado_act = 0, btn1estado_ant = 1, bot1estado = 0;
int btn2estado_act, btn2estado_ant, bot2estado = 0;
int btn3estado_act, btn3estado_ant, bot3estado = 0;
int btn4estado_act, btn4estado_ant, bot4estado = 0; // Asociado electroválvula 1
inti down; //Almacenar cuando se presione el bot5
inti seleccionAct = 0; // Crear una selección entre dosificar y fumigar

// etiquetar botones
int bot1 = 3; // A/D programación de Riego
int bot2 = 4; // Activar riego en ese momento
int bot3 = 7; // Función controlar el desplazamiento del menú
int bot4 = 10; //Función seleccionar la función del menú

// Declaración de los elementos a controlar por los RELES
int electrovalvula1 = 26;
int electrovalvula2 = 28;
int electrovalvula3 = 24;
int electrovalvula4 = 30;
int electrovalvula5 = 32;
```

```
int electrovalvula6 = 22;
int mezcladorMotor = 36;
int BOMBA = 38;

// Creación de menú que observa el usuario, se crea 3 pantallas con diferente información
LiquidLine line1_1(0, 0, "R");
LiquidLine line3_1(1, 2, "Dosificar"); // Pantalla 1
LiquidLine line4_1(1, 3, "Fumigar");
LiquidScreen screenlcd1(line1_1, line3_1, line4_1); // Se añade a un variable creada con propósito Específico

LiquidLine line1_2(1, 0, "     TANQUE");
LiquidLine line2_2(1, 1, "150 l"); // Pantalla 2
LiquidLine line3_2(1, 2, "180 l");
LiquidLine line4_2(1, 3, "210 l");
LiquidScreen screenlcd2(line1_2, line2_2, line3_2, line4_2);

LiquidLine line1_3(1, 0, "     Mezcla");
LiquidLine line2_3(1, 1, "OK"); // Pantalla 3
LiquidLine line3_3(1, 2, "INGRESE");
LiquidLine line4_3(1, 3, "NUTRIENTE");
LiquidScreen screenlcd3(line1_3, line2_3, line3_3, line4_3);
LiquidMenu menu(lcd, screenlcd1, screenlcd2, screenlcd3); // Se añade al LCD

void setup() {
  Serial.begin(115200);
  nodemcu.begin(115200);
  delay(1000);
  Wire.begin();     // Inicia el puerto I2C
  RTC.begin();
  //RTC.adjust(DateTime(__DATE__, __TIME__)); // Obtiene la hora del ordenador se comenta una vez subida al microcontrolador
  lcd.begin();      // Iniciar LCD
  lcd.backlight();   //Se prende la luz de fondo del LCD
  lcd.clear();
  // Se declara como entradas o salidas
  // Varibales del sensor ultrasónico
  pinMode(trig, OUTPUT); // emisor
  pinMode(echo, INPUT); // Receptor
  // Variables a controlar
  pinMode(electrovalvula1, OUTPUT);
  pinMode(electrovalvula2, OUTPUT);
  pinMode(electrovalvula3, OUTPUT);
  pinMode(electrovalvula4, OUTPUT);
  pinMode(electrovalvula5, OUTPUT);
  pinMode(electrovalvula6, OUTPUT);
  pinMode(mezcladorMotor, OUTPUT);
  pinMode(BOMBA, OUTPUT);
  // Pulsadores
  pinMode(bot1, INPUT);
  pinMode(bot2, INPUT);
  pinMode(bot3, INPUT_PULLUP);   //Al iniciar sistema se encuentra en HIGH
  pinMode(bot4, INPUT_PULLUP);
```

```
 // Cada una de las líneas de LCD puede tener una función dependiendo de la selección
 // ejecuta una acción o cambia de pantalla

line3_1.set_focusPosition(Position::LEFT);  //El seleccionador de LCD está a la izquierda

line4_1.set_focusPosition(Position::LEFT);
 line3_1.attach_function(1, fn_cantidadLitrosDosificar);
 line4_1.attach_function(1, fn_cantidadLitrosFumigar);
 menu.add_screen(screenlcd1);
line1_2.set_focusPosition(Position::RIGHT);
line2_2.set_focusPosition(Position::RIGHT);
line3_2.set_focusPosition(Position::RIGHT);   //El seleccionador de LCD está a la derecha
line4_2.set_focusPosition(Position::RIGHT);
 line1_2.attach_function(1, fn_regresar);
 line2_2.attach_function(1, fn_150l);
 line3_2.attach_function(1, fn_180l);
 line4_2.attach_function(1, fn_210l);
 menu.add_screen(screenlcd2);


line1_3.set_focusPosition(Position::RIGHT);
line2_3.set_focusPosition(Position::RIGHT);
 line1_3.attach_function(1, fn_regresar);
 line2_3.attach_function(1, fn_OK);
 menu.add_screen(screenlcd3);

 screenlcd1.set_displayLineCount(4);
 screenlcd2.set_displayLineCount(4);
 screenlcd3.set_displayLineCount(4);

 menu.set_focusedLine(2);  //  posición del marcador selección
 menu.update(); // Se actualiza LCD para cambiar pantallas creadas

}

void loop() {
 // Obtención e impresión de la hora tiempo asociado a la computadora
 // Se observa en LCD
 DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC
 lcd.setCursor(11, 2);
 lcd.print(now.year(), DEC);
 lcd.print("/");
 lcd.print(now.month(), DEC);
 lcd.print("/");
 lcd.print(now.day(), DEC);
 lcd.print(" ");
 lcd.setCursor(12, 3);       // Se imprime la hora y fecha en LCD
 lcd.print(now.hour(), DEC);
 lcd.print(":");
```

```
  lcd.print(now.minute(), DEC);
  lcd.print(":");
  lcd.print(now.second(), DEC);
  delay(100);
  lcd.setCursor(11, 2);
  lcd.print("        ");
  lcd.setCursor(12, 3);
  lcd.print("       ");

  // Riego Programado Activar o Desactivar
  btn1estado_act = digitalRead(bot1);
  if ((btn1estado_act == LOW) && (btn1estado_ant == HIGH)) {
    bot1estado = 1 - bot1estado; // Depende se selecciona la acción a realizar
  }
  btn1estado_ant = btn1estado_act; // Ayuda a guardar el estado del botón
  if (bot1estado == 1) { //Si se encuentra en 1 imprime un digito LCD para saber que se activo
    int minuto = now.minute();
    int hora = now.hour();
    lcd.setCursor(1, 0);
    lcd.print("(A)"); // Se verifica que la función ha sido activada
    if ( hora == 7 && minuto >= 0 && minuto <= 1) { //Que tiempo se desea que se active la función
      lcd.setCursor(0, 0);
      lcd.print("                    ");
      lcd.setCursor(0, 1);
      lcd.print("                    "); //Elimina todo LCD
      lcd.setCursor(0, 2);
      lcd.print("                    ");
      lcd.setCursor(0, 3);
      lcd.print("                    ");
      delay(800);
      lcd.setCursor(0, 0); //Al ejecutarse se puede visualizar un aviso
      lcd.print("RIEGO");
      lcd.setCursor(0, 1);
      lcd.print("PROGRAMADO");
      lcd.setCursor(0, 2);
      lcd.print("EJECUTANDOSE");
      delay(800);
      bot1estado = 0;
      StaticJsonDocument<1000> doc;
      ActEnviarRiegoProgramado(); // Conoce que valor se a trasmitir
      doc["send"] = h;
      serializeJson(doc, nodemcu);
      digitalWrite(electrovalvula4, LOW); //CONTROL DE RIEGO DEL BOTÓN 1
      delay(1000);
      bot1estado = 0;
      digitalWrite(electrovalvula5, LOW);
      digitalWrite(electrovalvula6, LOW);
      delay(1800 * 1000);

      menu.change_screen(1); //Que pantalla cambiar
      menu.set_focusedLine(1); //Posición del marcador selección
    }
    else {
      delay(100);
      lcd.setCursor(0, 0);
      lcd.print("                    ");
```

```
    lcd.setCursor(0, 1);
    lcd.print("                    ");   //Elimina
todo LCD
    lcd.setCursor(0, 2);
    lcd.print("                    ");
    menu.change_screen(1);        //Que
pantalla cambiar
    menu.set_focusedLine(1);
//Posición del marcador selección
   }
  }
  else {
   delay(100);
   lcd.setCursor(1, 0); // borra el aviso de
presionar el botón
   lcd.print("   ");
  }
  // Fin Riego Programado7
  // Presiona Botón2 - Comienza Riego
en ese momento
  btn2estado_act = digitalRead(bot2);
  if ((btn2estado_act == LOW) &&
(btn2estado_ant == HIGH)) {
   bot2estado = 1 - bot2estado;
  }

  btn2estado_ant = btn2estado_act;
  if (bot2estado == 1) {
   lcd.setCursor(0, 0);
   lcd.print("                    ");
   lcd.setCursor(0, 1);
   lcd.print("                    ");   //Elimina
todo LCD
   lcd.setCursor(0, 2);
   lcd.print("                    ");
   lcd.setCursor(0, 3);
   lcd.print("                    ");
   lcd.setCursor(6, 0);
   lcd.print("  RIEGO");
   lcd.setCursor(6, 1);        // Aviso de
actividad ejecutándose
   lcd.print("EJECUTANDOSE");
   delay(800);
   StaticJsonDocument<1000> doc;
   ActEnviarRiego(); //Conoce que valor
se debe trasmitir
   doc["send"] = h;
   serializeJson(doc, nodemcu);
   delay(800);
   bot1estado = 0;
   digitalWrite(electrovalvula4,     LOW);
//CONTROL DE RIEGO DEL BOTÓN 1
   delay(1000);
   digitalWrite(electrovalvula5, LOW);
   digitalWrite(electrovalvula6, LOW);
   delay(18000 * 1000);
   lcd.setCursor(6, 0);
   lcd.print("      ");
   lcd.setCursor(6, 1);
   lcd.print("           ");   //Elimina aviso
   delay(500);
   menu.change_screen(1);        //Que
pantalla cambiar
   menu.set_focusedLine(1);  //Posición
del marcador selección

  }

  // Fin Botón2 - Fin Riego

  // Los RELES off hasta que se requiera
  digitalWrite(electrovalvula1, HIGH);
```

```
  digitalWrite(electrovalvula2, HIGH);
  digitalWrite(electrovalvula3, HIGH);
  digitalWrite(electrovalvula4, HIGH);
  digitalWrite(electrovalvula5, HIGH);
  digitalWrite(electrovalvula6, HIGH);
  digitalWrite(mezcladorMotor, HIGH);
  digitalWrite(BOMBA, HIGH);


  medidaSensor(); // En LCD me muestra la cantidad de litros en el tanque
  seleccion(); // Si presiona el botón4
  desplazarLCD(); // presiona botón3 y mueve cursor

} // FIN VOID LOOP

void medidaSensor() {
  // Se estabiliza el sensor Ultrasónico
  //Para estabilizar el módulo ultrasónico
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  //disparo de un pulso en el trigger de longitud 10us
  digitalWrite(trig, HIGH);
  delayMicroseconds(50);
  digitalWrite(trig, LOW);

  //Lectura de la duración del pulso HIGH generado hasta recibir el Echo
  duracionTiempo = pulseIn(echo, HIGH);



  //Calculo distancia centímetros/metros
  distanciaCM = duracionTiempo / 58, 4; // (cm)
  distanciaRecipienteMetros = distanciaCM * 0.01;
  // Conversión a litros
  volumenRecipiente = (0.363168 * (0.79 - (distanciaRecipienteMetros))); // Fórmula del volumen (pi*r*r*h) en m3
  litrosRecipiente = volumenRecipiente * 1000;
  Serial.println(litrosRecipiente);
  Serial.println(distanciaCM);

  if (distanciaCM != ultimaDistanciaCM) {

    if (distanciaCM >= 77) {
      litrosRecipiente = 0;
      lcd.setCursor(15, 0);
      lcd.print("l:");
      lcd.print(litrosRecipiente);
      delay(1000);
      lcd.setCursor(15, 0);
      lcd.print("l:   ");
      delay(1000);
      ultimaDistanciaCM = distanciaCM;
    }

    if (76 >= distanciaCM && 15 <= distanciaCM) {



      lcd.setCursor(15, 0);
      lcd.print("l:");
      lcd.print(litrosRecipiente);
      delay(100);
      lcd.setCursor(15, 0);
```

```
      lcd.print("l:   ");
      delay(100);

      ultimaDistanciaCM = distanciaCM;
    }
  }
}


// Botón4 función seleccionar en el menú
void seleccion() {
  if (digitalRead(bot4) == LOW) {
    delay(500);
    menu.call_function(1);

  }
}

//Desplazarse por LCD con el botón3
void desplazarLCD() {
  if (digitalRead(bot3) == LOW ) {
    Serial.println("Pul3");
    menu.switch_focus(true);
  }
}

// Función regresar para el menú
void fn_regresar() {
  menu.change_screen(1);
  menu.set_focusedLine(1);

}

void fn_cantidadLitrosDosificar() {
  seleccionAct = 1;
  menu.change_screen(2);
  menu.set_focusedLine(1);
}

void fn_cantidadLitrosFumigar() {
  seleccionAct = 2;
  menu.change_screen(2);
  menu.set_focusedLine(1);
}

void fn_150l() {
  medidaSensor();
  while (distanciaCM > 38) { //En la fórmula de conversión a litros se conoce
    medidaSensor();          //la distancia puesta es lo mismo de poner 150l
    digitalWrite(electrovalvula1, LOW);
    delay(100);
    lcd.setCursor(0, 0);
    lcd.print("                    ");
    lcd.setCursor(0, 1);
    lcd.print("                    ");
    lcd.setCursor(0, 2);
    lcd.print("                    ");
    lcd.setCursor(0, 3);
    lcd.print("                    ");
    delay(100);
    lcd.setCursor(2, 1);
    lcd.print("LLENANDO");
    lcd.setCursor(3, 2);
    lcd.print("TANQUE");
    delay(100);
  }
  digitalWrite(electrovalvula1, HIGH);
  menu.change_screen(3);
  menu.set_focusedLine(1);
```

```
}
void fn_180l() {
  medidaSensor();
  while (distanciaCM > 30) {
    medidaSensor();
    digitalWrite(electrovalvula1, LOW);
    delay(100);
    lcd.setCursor(0, 0);
    lcd.print("                    ");
    lcd.setCursor(0, 1);
    lcd.print("                    ");
    lcd.setCursor(0, 2);
    lcd.print("                    ");
    lcd.setCursor(0, 3);
    lcd.print("                    ");
    delay(100);
    lcd.setCursor(2, 1);
    lcd.print("LLENANDO");
    lcd.setCursor(3, 2);
    lcd.print("TANQUE");
    delay(100);

  }
  digitalWrite(electrovalvula1, HIGH);
  menu.change_screen(3);
  menu.set_focusedLine(1);

}
void fn_210l() {
  medidaSensor();
  while (distanciaCM > 21) {
    medidaSensor();
    digitalWrite(electrovalvula1, LOW);
    delay(100);
    lcd.setCursor(0, 0);
    lcd.print("                    ");
    lcd.setCursor(0, 1);
    lcd.print("                    ");
    lcd.setCursor(0, 2);
    lcd.print("                    ");
    lcd.setCursor(0, 3);
    lcd.print("                    ");
    delay(100);
    lcd.setCursor(2, 1);
    lcd.print("LLENANDO");
    lcd.setCursor(3, 2);
    lcd.print("TANQUE");
    delay(100);

  }
  digitalWrite(electrovalvula1, HIGH);
  menu.change_screen(3);
  menu.set_focusedLine(1);
}

void fn_OK() {
  lcd.setCursor(0, 0);
  lcd.print("                    ");
  lcd.setCursor(0, 1);
  lcd.print("                    ");
  lcd.setCursor(0, 2);
  lcd.print("                    ");
  lcd.setCursor(0, 3);
  lcd.print("                    ");
  delay(500);
  lcd.setCursor(5, 1);
  lcd.print("AGITADOR ON");
  delay(500);
  digitalWrite(mezcladorMotor, LOW);
  delay(180000);
  lcd.setCursor(5, 1);
  lcd.print("          ");
```

```
lcd.setCursor(5, 2);
lcd.print("AGITADOR OFF");
lcd.setCursor(5, 3);
lcd.print("              ");
delay(500);
digitalWrite(mezcladorMotor, HIGH);
delay(1000);
lcd.setCursor(5, 2);
lcd.print("              ");
medidaSensor();
Serial.println(seleccionAct);

switch (seleccionAct) {
  case 1: //Selecciona Dosificar
    StaticJsonDocument<1000> doc;
    ActEnviarDosificar();
    doc["send"] = h;
    serializeJson(doc, nodemcu);
    delay(800);
    while (distanciaCM < 72) {
      medidaSensor();
      lcd.setCursor(2, 1);
      lcd.print("DOSIFICANDO");
      lcd.setCursor(3, 3);
      lcd.print("BOMBA ON");
      delay(500);
      digitalWrite(electrovalvula2, LOW);
      digitalWrite(electrovalvula3, LOW);
      digitalWrite(electrovalvula4, LOW);
      digitalWrite(electrovalvula5, LOW);
      digitalWrite(electrovalvula6, LOW);
      //delay(180000);
      digitalWrite(BOMBA, LOW);

      medidaSensor();
    }
```

```
    lcd.setCursor(5, 1);
    lcd.print("          ");
    lcd.setCursor(7, 2);
    lcd.print("      ");
    lcd.setCursor(0, 3);
    lcd.print("BOMBA OFF");
    digitalWrite(BOMBA, HIGH);
    digitalWrite(electrovalvula4, HIGH);
    digitalWrite(electrovalvula5, HIGH);
    digitalWrite(electrovalvula6, HIGH);
    digitalWrite(electrovalvula2, HIGH);
    delay(3000);
    digitalWrite(electrovalvula3, HIGH);
    menu.change_screen(1);
    menu.set_focusedLine(1);

    break;

  case 2:
    medidaSensor();
    if (distanciaCM < 70) {
      StaticJsonDocument<1000> doc;
      ActEnviarMezclarListo();
      doc["send"] = h;
      serializeJson(doc, nodemcu);
      delay(800);
      lcd.setCursor(2, 1);
      lcd.print("LISTO");
      lcd.setCursor(3, 3);
      lcd.print("MEZCLA");
      delay(500);
      delay(60000);

      menu.change_screen(1);
      menu.set_focusedLine(1);
```

```
      }
      break;

  }

}

void ActEnviarRiego() {
  h = 1;
  Serial.print("Enviado: ");
  Serial.println(h);
}
void ActEnviarDosificar() {
  h = 2;
  Serial.print("Enviado: ");
  Serial.println(h);
}
void ActEnviarMezclarListo() {
  h = 3;
  Serial.print("Enviado: ");
  Serial.println(h);
}
void ActEnviarRiegoProgramado() {
  h = 4;
  Serial.print("Enviado: ");
  Serial.println(h);
}
```

ANEXO 4: PRESENTACIÓN CODIGO DE ESP8266

```
// Librerías a usar
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <BlynkSimpleEsp8266.h>

// Variables
const String SERVER_ADDRESS =
"http://192.168.0.105/Servidor%20Tesis
";
// Código generado por la aplicación
char auth[] =
"PIieWwiA3gC3nRGq3LdhxpmbEr33X-
tL";
//Identificadores de la red a usar
char ssid[] =
"TELPRONET_CAROLINA";
char pass[] = "curay1980";

//D6 = Rx & D5 = Tx
#define BLYNK_PRINT Serial
SoftwareSerial nodemcu(D6, D5); //
pines de comunicación serial
#define D1 5
#define D2 4
int b;
int h;
int g;
String a;

String mensaje;
int num = 0;

void setup() {
  //Inicia la comunicación Serial
  Serial.begin(115200);
  nodemcu.begin(115200);
  Blynk.begin(auth, ssid, pass); // Se
debe incluir para el uso d la app

  while (!Serial) continue;


}

void loop() {
  Blynk.run();

  StaticJsonDocument<1000> doc;
  DeserializationError error =
deserializeJson(doc, nodemcu); // se
utiliza para recibir el mensaje

  if (error) {
    DeserializationError error =
deserializeJson(doc, nodemcu);
  }
  h = doc["send"];
  switch (h) { // dependiendo del digito
que reciba se genera una palabra

case 1:
      a = "Riego";
      Serial.println(a);

      break;
    case 2:
      a = "Dosificar";
      break;

    case 3:
      a = "Mezclar";
      break;

    case 48:
      a = "RiegoProgramado";
      break;

  }

  if (a == "Riego" || a == "Mezclar" || a
== "Dosificar" || a ==
"RiegoProgramado") { // se envía los
daos si coincide
    ////////////
    WiFiClient client;
    HTTPClient http;
    String full_url = SERVER_ADDRESS
+ "/recibe_datos.php?ActRealizar=" +
a;
    http.begin(client, full_url);
```

```
    int httpCode = http.GET();
    if (httpCode > 0)




    {
      if (httpCode == HTTP_CODE_OK)
      {
        String payload = http.getString();
        Serial.println(payload);
      }
      else
      {
        Serial.println("Error”);
      }
    }
    else
    {
      Serial.println("Request failed: " +
http.errorToString(httpCode));
    }
    http.end(); //Finaliza la comunicación
    a = ""; // Limpia la variable que se
envió


  }
}
```

ANEXO 5: PRESENTACIÓN CODIGO DE PHP CONEXIÓN CON LA BASE DE DATOS

```
<?php
class conexion {
   const user ='root'; // Se da la dirección
de la base de datos
   const pass ='';
   const bdatos ='servidortesis';
   const servidor ='localhost';
   public function conectarbdatos(){
      $conectar = new
mysqli(self::servidor,self::user,self::pass
,self::bdatos); // se verifica los datos
if($conectar->connect_error){
         die("Error".$conectar-
>connect_error);
      }
      return $conectar;
   }
}
?>
```

ANEXO 6: PRESENTACIÓN CODIGO DE PHP CONEXIÓN CON ESP8266

```
<?php
   require_once('conectado.php');        //
Indica el codigo de conectado.php
$actividadRealizar=$_GET['ActRealizar'
]; // Espera para almacenar la variable
que se recibe del ESP8266

if($actividadRealizar=='Riego'          ||
$actividadRealizar=='Dosificar'
||$actividadRealizar=='Mezclar'
||$actividadRealizar=='RiegoProgramad
o'){
   // Si la palabra recibida es la que se
encuentra en las opciones procede a
enviarla a la base de datos
   $connew = new conexion();
      //Actualiza la tabla actividades
      $query = "UPDATE actividades
SET Actividad= '$actividadRealizar'
WHERE IdPlaca ='Arduino'";
      $insert = mysqli_query($connew-
>conectarbdatos(),$query);
      //Introduc los datos en la tabla
historialactividades
      $query = "INSERT INTO
historialactividades(IdPlaca,Ejecutado,
Fecha)                    VALUES
('Arduino','$actividadRealizar', NOW())";
      $insert = mysqli_query($connew-
>conectarbdatos(),$query);
      //Respuesta que se envía al
ESP8266
      echo "Datos_Registrados";
      echo
"{ACT:".$actividadRealizar."}";
}


?>
```

ANEXO 7: COSTO DEL PROYECTO IMPLEMENTADO

Componentes	Cantidad	Precio Unitario	Total
Tanque Botella 250 (LTS)	1	55,00	55,00
Arduino Mega	1	25,00	25,00
Adaptador 12 (V) - 2 (A)	1	5,00	5,00
Módulo Relé 8 Canales	2	14,00	28,00
Sensor Ultrasónico Impermeable	1	20,00	20,00
Case de Arduino Mega	1	5,00	5,00
Electroválvula 1/2 (in) - 110 (V)	4	13,00	52,00
Electroválvula a gravedad 1/2 (in) - 110 (V)	2	43,00	86,00
Módulo ESP8266	1	8,00	8,00
Módulo LM2596	4	3,00	12,00
Módulo RTC DS3231	1	4,00	4,50
Fuente 15 (V) - 30 (A)	1	25,00	25,00
Pulsadores	10	0,20	2,00
LCD 2004 con I2C	1	14,00	14,00
Canaleta ranurada	2	4,00	8,00
Riel din Barra 1 (m)	1	3,50	3,50
Interruptor Termomagnético 30 (A)	1	12,00	12,00
Interruptor Termomagnético 6 (A)	3	6,00	18,00
Gabinete metálico d/fondo 60x40x20	1	60,00	60,00
Base de fondo de gabinete	1	5,00	5,00
Bloque de Distribución Fase - Neutro	1	16,00	16,00
Motor DC 12-24 (V)	1	50,00	50,00
Bomba	1	45,00	45,00
Porta fusibles	1	10,00	10,00
Fusibles	12	0,80	9,60
Borneras Industriales	16	1,25	20,00
Terminales Faston Hembra	30	0,20	6,00
Tornillos	1	2,00	2,00
Hélice mezcladora	1	21,00	21,00
Acoples de tuberías de 1/2 (in)	14	2,00	28,00
Válvulas manuales	6	2,50	15,00
Hilo de Estaño - 3 (m)	1	1,80	1,80
Cable 18 AWG - 100 (m)	1	30,00	30,00
Cable 14 AWG - 15 (m)	1	6,00	6,00
Cable 12 AWG - 30 (m)	1	16,80	16,80
Tubo Flex 3/4 - 1 (m)	20	0,31	6,2
Tubería PVC - 6 (m)	1	9,00	9,00
		TOTAL	740,40