

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL
DE HUAMANGA**

FACULTAD DE CIENCIAS AGRARIAS

ESCUELA PROFESIONAL DE INGENIERÍA AGRÍCOLA



TESIS:

**Implementación y monitoreo con microcontrolador Arduino
para automatizar el riego por goteo en parcela de hortalizas,
Tambo, Ayacucho-2024**

Para optar el título profesional de:
INGENIERA AGRÍCOLA

PRESENTADO POR:
Bach. Diana Marisella LOAYZA PAREDES

ASESOR:
M.Sc. César Vidal GUTIÉRREZ NINAHUAMÁN

AYACUCHO - PERÚ

2025

DEDICATORIA

A Dios por haberme permitido llegar hasta este punto para así lograr mis objetivos, por darme fuerzas para superar obstáculos y dificultades a lo largo de este camino.

A mi madre Rufina Paredes, cuyo sacrificio y amor incondicional son la base de cada paso que doy y a mi padre Nolberto Loayza por su constante apoyo.

También a mi familia en general por su apoyo incondicional y motivación en cada momento de mi vida.

AGRADECIMIENTO

Por haber sido pilares fundamentales en mi desarrollo profesional, expreso mis sinceros agradecimientos:

- A la Universidad Nacional San Cristóbal de huamanga, a mi alma máter, a la Facultad de Ciencias Agrarias; en especial a mi gloriosa Escuela Profesional de Ingeniería Agrícola por brindarme la oportunidad de formarme profesionalmente, por forjar mi carácter y a su vez darme las herramientas para llegar a ser una gran profesional.
- A mi Asesor de Tesis M.Sc. Ing. César Vidal Gutiérrez Ninahuamán, por su sabia experiencia y conocimiento compartido durante la elaboración de mi tesis.
- A los miembros del jurado de esta investigación por sus valiosas observaciones que enriquecieron significativamente este trabajo.
- A mis amigos, Jhour ayudarme a desarrollar mis habilidades tecnológicas en el lenguaje de programación y Whitman, por darme la oportunidad de desarrollar mi tesis en su invernadero.

ÍNDICE GENERAL

DEDICATORIA	i
AGRADECIMIENTO.....	ii
RESUMEN	9
ABSTRACT	10
INTRODUCCION.....	11
Objetivo general.....	11
Objetivos específicos	11
CAPÍTULO I MARCO TEORICO	13
1.1. Antecedentes.....	13
<i>1.1.1. Antecedente a nivel internacional</i>	<i>13</i>
<i>1.1.2. Antecedente a nivel nacional.....</i>	<i>14</i>
<i>1.1.3. Antecedente a nivel local.....</i>	<i>16</i>
1.2 Bases teóricas.....	16
<i>1.2.1. Invernadero</i>	<i>16</i>
<i>1.2.2. Invernadero inteligente</i>	<i>17</i>
<i>1.2.3. Hortaliza zanahoria</i>	<i>17</i>
<i>1.2.4. Riego.....</i>	<i>18</i>
<i>1.2.5. Tipos de riego.....</i>	<i>18</i>
<i>1.2.6. Componentes del sistema de riego por goteo</i>	<i>20</i>
<i>1.2.7. Tiempo de riego.....</i>	<i>22</i>
<i>1.2.8. Frecuencia o intervalo de riego</i>	<i>23</i>
<i>1. 1.2.9. Automatización del sistema riego por goteo.....</i>	<i>23</i>
CAPÍTULO II METODOLOGIA	36
2.1. Aspectos generales del proyecto	36
<i>2.1.1. Ubicación</i>	<i>36</i>
<i>2.1.2. Vías de acceso</i>	<i>38</i>
<i>2.1.3. Clima.....</i>	<i>39</i>
<i>2.1.4. Antecedentes del sistema de riego.....</i>	<i>39</i>
2.2. Materiales y equipos	42
<i>2.2.1. Gabinete</i>	<i>42</i>

2.2.2. <i>Campo</i>	42
2.3. Problemas específicos	44
2.4. Metodología	45
2.4.1. <i>Recopilación de información sobre el sistema de riego del invernadero (tomar muestras para el análisis de suelo – agua y prueba de coeficiente de uniformidad)</i>	46
2.4.2. <i>Selección de sensores, actuador y demás materiales para la automatización del sistema de riego por goteo del invernadero.</i>	51
2.4.3. <i>Programación de los sensores y actuador en el sketch de Arduino con sus respectivos códigos para su funcionamiento.</i>	54
2.4.4. <i>Implementación de sensores y actuadores en el sistema de riego por goteo del invernadero y realizar las pruebas exhaustivas.</i>	65
2.4.5. <i>Calibración y ajustes de los sensores para una correcta lectura.</i>	79
2.4.6. <i>Creación de un aplicativo en la plataforma Blynk para tener lecturas en tiempo real.</i>	85
2.4.7. <i>Cosecha de la Hortaliza zanahoria.</i>	89
CAPÍTULO III RESULTADOS	90
3.1. Resultado de la Implementación y monitoreo con microcontrolador Arduino para la automatización de riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024.	90
3.1.1. <i>Resultado sobre el Análisis de los componentes de la Automatización para el riego por goteo.</i>	90
3.1.2. <i>Resultado sobre el desarrollo de un sistema efectivo de comunicación entre el sensor y el microcontrolador Arduino.</i>	91
3.1.3. <i>Resultado sobre el desarrollo del código fuente para automatizar el riego por goteo empleando la tecnología del Arduino.</i>	141
3.1.4. <i>Resultado de estimación de la cantidad de zanahorias que se cosecharon al finalizar la campaña agrícola.</i>	153
DISCUSIONES	154
CONCLUSIONES	161
RECOMENDACIONES	163
REFERENCIAS BIBLIOGRÁFICAS	164
ANEXOS	166

ÍNDICE DE TABLAS

Tabla 1. 1 Especificaciones Técnica de la placa Arduino Mega 2560 Rev3.....	24
Tabla 1. 2 Interfaz IDE Arduino	25
Tabla 1. 3 Tipos de sensores	28
Tabla 1. 4 Especificaciones Técnicas del Sensor Ultrasonido HC-SR04.....	29
Tabla 1. 5 Especificaciones Técnicas del Sensor presión HK 3025.....	30
Tabla 1. 6 Especificaciones técnicas del Sensor flujo de agua 2” YF-DN50.....	31
Tabla 1. 7 Especificaciones técnicas del sensor DHT11 de temperatura	32
Tabla 1. 8 Especificaciones técnicas del Módulo higrómetro	33
Tabla 1. 9 Tipos de actuadores.....	34
Tabla 1. 10 Especificaciones técnicas de la electroválvula Serie PGA 2”	35
Tabla 2. 1 Vías de comunicación al Distrito Tambo-Pampa Hermosa	38
Tabla 2. 2 Resultado del análisis Físico de suelos.....	46
Tabla 2. 3 Resultado del análisis del agua.....	47
Tabla 2. 4 Datos recolectados del CU en riego sin automatizar.....	49
Tabla 2. 5 Datos recolectados del CU en riego automatizado.....	50
Tabla 3. 1 Resultado de la comunicación efectiva del sensor de Humedad.....	92
Tabla 3. 2 Resultado de la comunicación efectiva del sensor de flujo	101
Tabla 3. 3 Resultado de la comunicación efectiva de los sensores de presión.....	111
Tabla 3. 4 Resultado de la comunicación efectiva del sensor ultrasónico	121
Tabla 3. 5 Resultado de la comunicación efectiva del sensor de temperatura	131

ÍNDICE DE FIGURAS

Figura 1. 1 Hortaliza zanahoria.....	18
Figura 1. 2 Riego por gravedad.....	18
Figura 1. 3 Riego por aspersión	19
Figura 1. 4 Riego por goteo.....	20
Figura 1. 5 Sistema de riego por goteo	22
Figura 1. 6 Placa Arduino	24
Figura 1. 7 Lenguaje de programación	26
Figura 1. 8 Entradas y salidas analógicas del Arduino	27
Figura 1. 9 Sensor Ultrasonido HC-SR04.....	29
Figura 1. 10 Sensor de presión HK-3025.....	30
Figura 1. 11 Sensor de presión HK-3025.....	31
Figura 1. 12 Sensor DHT11 de temperatura	32
Figura 1. 13 Módulo higrómetro sensor de humedad del suelo	33
Figura 1. 14 Electroválvula Serie PGA 2”	35
Figura 2. 1 Ubicación Satelital del Invernadero.....	36
Figura 2. 2 Ubicación política del proyecto	37
Figura 2. 3 Mapa de acceso al proyecto.....	38
Figura 2. 4 Pampa Hermosa - Lorensayoc.....	39
Figura 2. 5 Reservorio de agua revestido con geomembrana.....	40
Figura 2. 6 Cabezal de riego	41
Figura 2. 7 Toma de medidas de la subunidad de riego.....	41
Figura 2. 8 Invernadero	41
Figura 2. 9 Toma de muestra de suelo para el análisis en laboratorio	46
Figura 2. 10 Análisis de la muestra de agua y suelo	47
Figura 2. 11 Prueba de coeficiente de uniformidad	48
Figura 2. 12 Esquema del sistema eléctrica	53
Figura 2. 13 Esquema de los sensores y actuador para la programación	53
Figura 2. 14 Diagrama de flujo de trabajo de la electroválvula en función del sensor ultrasónico y humedad del suelo.....	54
Figura 2. 15 Programación de código del sensor HC-SR04 en Arduino Mega	56
Figura 2. 16 Código de programación del sensor de humedad de suelo en Arduino Mega.....	57
Figura 2. 17 Diagrama de flujo del sensor de presión antes y después del filtro.	58
Figura 2. 18 Código de programación del sensor de presión Hk3025 en la placa Arduino.....	59

Figura 2. 19	Diagrama de flujo del sensor de flujo de agua 2" YF-DN50.....	60
Figura 2. 20	Código de programación del sensor de flujo de agua en la placa Arduino.....	61
Figura 2. 21	Diagrama de flujo del sensor de temperatura	62
Figura 2. 22	Código de programación del sensor de temperatura en la placa Arduino	63
Figura 2. 23	Código de programación del módulo relé de 4 canales en la placa Arduino.....	64
Figura 2. 24	Código de programación módulo Ethernet Shield W5100	65
Figura 2. 25	Limpieza del reservorio	66
Figura 2. 26	Instalación de la caseta de control	67
Figura 2. 27	Caja de protección en el cabezal de riego.....	67
Figura 2. 28	Caja de protección para la electroválvula	68
Figura 2. 29	Comunicación entre el sensor ultrasonido HC-SR04 y placa Arduino.....	68
Figura 2. 30	Instalación del sensor ultrasónico HC-SR04	69
Figura 2. 31	Comunicación entre el Sensor de presión Hk3025 y placa Arduino	69
Figura 2. 32	Instalación del sensor de presión Hk3025	70
Figura 2. 33	Comunicación entre Sensor de flujo de agua YF-DN50 y placa Arduino.....	70
Figura 2. 34	Instalación del sensor de flujo YF-DN50	71
Figura 2. 35	Comunicación del Sensor DHT11 de temperatura y la placa Arduino.....	71
Figura 2. 36	Instalación del Sensor DHT11 de temperatura	72
Figura 2. 37	Comunicación entre el sensor de humedad y placa Arduino.....	72
Figura 2. 38	Instalación del sensor de humedad	73
Figura 2. 39	Comunicación entre el Módulo Relé de 4 canales y la placa Arduino	73
Figura 2. 40	Instalación del Módulo Relé de 4 canales	74
Figura 2. 41	Comunicación del Módulo de Ethernet y la placa Arduino.....	74
Figura 2. 42	Instalación del Módulo Ethernet Shield.....	75
Figura 2. 43	Instalación de la placa Arduino Mega 2560 R3.....	75
Figura 2. 44	Comunicación entre el MicroSD Car Adapter y la placa Arduino	76
Figura 2. 45	Instalación del MicroSD Card Adapter	76
Figura 2. 46	Esquema del sistema eléctrico en la caja hermética	77
Figura 2. 47	Instalación de la alimentación eléctrica	77
Figura 2. 48	Instalación de la electroválvula.....	78
Figura 2. 49	Sembrado de la Zanahoria	79
Figura 2. 50	Toma de datos para ajustar las lecturas del sensor ultrasónico HC-SR04	79
Figura 2. 51	Calibración del sensor de ultrasonido HC-SR04	80
Figura 2. 52	Toma de datos para ajustar las lecturas del sensor presión HK-3025.....	81
Figura 2. 53	Calibración del sensor de presión Hk-3025 antes del filtro	81

Figura 2. 54	Calibración del sensor de presión Hk-3025 después del filtro	82
Figura 2. 55	Toma de datos para ajustar las lecturas del sensor de flujo de agua 2" YF-DN50	82
Figura 2. 56	Calibración del sensor de flujo de agua 2" YF-DN50	83
Figura 2. 57	Toma de datos para ajustar las lecturas del sensor de temperatura.....	83
Figura 2. 58	Calibración del sensor DHT11 de temperatura.....	84
Figura 2. 59	Toma de datos para ajustar las lecturas del sensor de humedad	84
Figura 2. 60	Calibración del sensor de humedad	85
Figura 2. 61	Conexión de los sensores y actuador para el monitoreo en Blynk.....	86
Figura 2. 62	Widgets para el encendido y apagado de la electroválvula	86
Figura 2. 63	Widgets para el control de presión antes y después del filtro de anilla	87
Figura 2. 64	Widgets para el control del nivel y volumen del reservorio	87
Figura 2. 65	Widgets para el control del caudal de agua	88
Figura 2. 66	Widgets para el control Widgets para el control de la humedad del suelo	88
Figura 2. 67	Widgets para el control Widgets para el control de la Temperatura.....	88
Figura 2. 68	Cosecha de zanahorias	89
Figura 3. 1	Datos recolectados del sensor de humedad.....	100
Figura 3. 2	Datos recolectados del sensor de flujo.....	110
Figura 3. 3	Datos recolectados del sensor de presiones	120
Figura 3. 4	Datos recolectados del sensor ultrasónico	130
Figura 3. 5	Datos recolectados del sensor de temperatura	140
Figura 3. 6	Código fuente de la automatización del sistema de riego por goteo.....	142

ÍNDICE DE ANEXOS

Anexo A Panel fotográfico

Anexo B Resultados de laboratorio de análisis Físico – Químico del agua

Anexo C Resultados de laboratorio de análisis Físico y caracterización del suelo

Anexo D Resultados de laboratorio de Humedad actual del suelo

Anexo E Ficha técnica de los emisores

Anexo F Presupuesto de la investigación de riego automatizado

Anexo G Planos

ÍNDICE DE ABREVIATURAS

IDE: Entorno de Desarrollo Integrado

PVC: Policloruro de vinilo

PE: Polietileno

mA = miliamperios

MHz = Megahercios

mm: Milímetros

cm: centímetros

KB: Kilobytes

V: Voltios (Volts)

PWM: Pulse Width Modulation (Modulación por Ancho de Pulso)

LCD: Liquid Crystal Display

SD: Secure Digital (digital segura)

W : watts

A: Amperios

μS : microsiemens

MPa : megapascals

RH : Humedad Relativa

Hz : Hertz

UTM : Universal Transverse Mercator

WGS 84: World Geodetic System 1984

m.s.n.m: metros sobre el nivel del mar

RAS: Relación de adsorción de sodio

Meq/L: mili equivalentes por litro

ppm: partes por millón

IoT: Internet de las Cosas

pH: Potencial de hidrogeno

CU: Coeficiente de uniformidad

RESUMEN

En el distrito de Tambo, provincia de La Mar, del departamento de Ayacucho, los métodos tradicionales de riego empleados en la producción de hortalizas han provocado un uso ineficiente del recurso hídrico y pérdidas de tiempo en el manejo y control del riego. Esta situación, junto con la necesidad de modernizar las prácticas agrícolas y aumentar la competitividad del sector, motivó la presente investigación. Este estudio tiene como objetivo general implementar y monitorear un sistema automatizado de riego por goteo basado en un microcontrolador Arduino en parcela dedicada a hortalizas. Para alcanzar este objetivo, se establecieron objetivos específicos que permitieron una comprensión detallada del sistema. Se recopiló información básica mediante la toma de muestras de suelo y agua, y se evaluó el coeficiente de uniformidad del riego. Asimismo, se seleccionaron, programaron, implementaron y calibraron sensores de humedad, caudal, presión, temperatura y nivel de agua, así como de la electroválvula. Los resultados mostraron que el sistema automatizado funcionó correctamente según los parámetros definidos, manteniendo una comunicación estable entre los sensores y el microcontrolador Arduino. El sistema permitió optimizar el uso del agua, reducir el tiempo en el monitoreo y alcanzar una producción estimada de 245 kg de zanahorias en 35 m², equivalente a un rendimiento de 7 kg/m², superando significativamente el rendimiento del riego convencional, que es de 2.5 kg/m², lo que evidencia la eficacia del sistema. En conclusión, la automatización del riego por goteo mediante Arduino constituye una solución viable y replicable en zonas rurales, ya que mejora el rendimiento agrícola, reduce el consumo de agua y promueve el uso sostenible de tecnologías accesibles.

Palabras clave: Riego automatizado, Arduino, Sensores, Tecnología, Calibración.

ABSTRACT

In the district of Tambo, province of La Mar, in the department of Ayacucho, traditional irrigation methods used in vegetable production have led to inefficient water use and time losses in irrigation management and control. This situation, along with the need to modernize agricultural practices and increase the competitiveness of the sector, motivated the present research. The general objective of this study was to implement and to monitor an automated drip irrigation system based on an Arduino microcontroller in a vegetable cultivation plot. To achieve this, specific objectives were established to provide a detailed understanding of the system. Basic information was collected through soil and water sampling, and the irrigation uniformity coefficient was evaluated. Sensors for soil moisture, flow, pressure, temperature, and water level were selected, programmed, implemented, and calibrated, along with the solenoid valve. The results showed that the automated system operated correctly according to the defined parameters, maintaining stable communication between the sensors and the Arduino microcontroller. The system optimized water use, reduced monitoring time, and achieved an estimated production of 245 kg of carrots in 35 m², equivalent to a yield of 7 kg/m², significantly surpassing the conventional irrigation yield of 2.5 kg/m², thus demonstrating the system's effectiveness. In conclusion, the automation of drip irrigation using Arduino is a viable and replicable solution in rural areas, as it improves agricultural yield, reduces water consumption, and promotes the sustainable use of accessible technologies.

Keywords: Automated irrigation, Arduino, Sensors, Technology, Calibration

INTRODUCCION

Para maximizar los rendimientos de los cultivos, aprovechar eficientemente los recursos hídricos y optimizar el uso de tecnologías que incrementen la competitividad y rentabilidad en la actividad agrícola, es esencial garantizar un manejo óptimo de los sistemas de riego.(Martínez, 2001)

El suministro de agua, a través de diversas técnicas de irrigación como el riego superficial, por aspersión, micro aspersión y goteo, desempeña un papel fundamental en la producción agrícola al influir directamente en el crecimiento y desarrollo de las plantas (Hargreaves & Merkle G.P., 2000).

Sin embargo, los métodos de riego convencionales utilizados en zonas agrícolas suelen llevar a un uso ineficiente del agua, lo que genera pérdidas de tiempo para los agricultores, rendimientos subóptimos en el cultivo de zanahoria y, en consecuencia, pérdidas económicas en la producción final. Para abordar estos problemas, esta tesis propone una alternativa, que es la implementación de esta tecnología Arduino, ya que proporciona al agricultor un control preciso sobre el suministro de agua, mediante la monitorización constante de la humedad del suelo a través de sensores y contribuye a mejorar el rendimiento del cultivo de zanahoria y reduce significativamente el tiempo de trabajo.

Finalmente, este trabajo de tesis tuvo un paso significativo hacia el avance tecnológico en el ámbito agropecuario del distrito Tambo, la localidad de Pampa Hermosa, anexo Lorensayoq, y se espera que sirva como modelo y catalizador para futuras iniciativas similares en la región y a nivel nacional. La combinación de la tecnología Arduino con el riego por goteo ofrece una solución innovadora y adaptable a las necesidades específicas de la agricultura en la zona, mejorando así la calidad de vida de los agricultores y promoviendo la sostenibilidad en el uso de los recursos naturales.

Objetivo general

- a. Implementar y monitorear con microcontrolador Arduino para la automatización de riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024.

Objetivos específicos

1. Analizar los componentes de la Automatización para el riego por goteo.

2. Desarrollar un sistema efectivo de comunicación entre el sensor y el microcontrolador Arduino.
3. Desarrollar el código fuente para automatizar el riego por goteo empleando la tecnología del Arduino.
4. Estimar el rendimiento de las zanahorias que se cosecharán al finalizar la campaña agrícola.

CAPÍTULO I

MARCO TEÓRICO

1.1. Antecedentes

1.1.1. Antecedente a nivel internacional

Cervantes (2016) , en su investigación titulada "Diseño e implementación de un sistema de riego automático controlado por placa Arduino para la finca La Lucía", tuvo como objetivo general crear un sistema de riego automatizado, utilizando la placa Arduino Mega, para optimizar la distribución del agua en la finca y así mejorar el desarrollo de los cultivos y, en consecuencia, la producción de las cosechas, impactando positivamente en la economía de los habitantes. Para ello utilizó una técnica de encuesta basada en un método deductivo, lo cual le permitió profundizar en los problemas de riego y así plantear una solución para optimizar el uso del agua. La solución incluyó la tecnificación del sistema de riego, para lo cual se emplearon componentes de hardware como la tarjeta Arduino Mega, una protoboard, cables, una electroválvula, relé, reloj RTC, tarjeta MicroSD, pantalla LCD y un teclado matricial. Tras la implementación y automatización del sistema, se encontró que el 86 % de los habitantes manifestaron interés en adoptar un sistema de riego automatizado para un control óptimo del agua, mientras que el 14 % consideró innecesaria la automatización. Finalmente, el sistema automatizado permitió una entrega oportuna y eficiente del agua hacia los cultivos, mejorando el rendimiento de la producción y satisfaciendo diversas necesidades agrícolas en la finca "La Lucía".

Nina (2020) , en su investigación titulada “Evaluación de tres variedades de zanahoria (*daucus carota* l.) en invernadero y a campo abierto en el municipio de la ciudad de La Paz” tuvo como objetivo evaluar el comportamiento de tres variedades de zanahoria (*Daucus carota* L.): morada, rojo rosáceo y chantenay, bajo condiciones de invernadero y a campo abierto en el Municipio de La Paz. Se buscó determinar las diferencias de producción, rendimiento y viabilidad económica en ambos ambientes. El problema central radica en la falta de estudios comparativos sobre la adaptabilidad y producción de estas variedades en diferentes condiciones de cultivo. Debido a la creciente demanda de zanahoria en el altiplano, es fundamental identificar las condiciones óptimas que maximicen su rendimiento y rentabilidad. La metodología se desarrolló en el vivero “Orgánico Los Almendros”, ubicado en el Macro Distrito Sur de la Ciudad de La Paz. Se evaluaron seis tratamientos, combinando tres variedades y dos ambientes de cultivo: T1 (zanahoria morada-invernadero), T2 (zanahoria morada-campo abierto), T3

(zanahoria rojo rosáceo-invernadero), T4 (zanahoria rojo rosáceo-campo abierto), T5 (zanahoria chantenay-invernadero) y T6 (zanahoria chantenay-campo abierto). Se midieron variables como porcentaje de emergencia, altura del follaje, diámetro y altura de la raíz, peso sin follaje, rendimiento y parámetros económicos (beneficio bruto, neto y relación beneficio/costo). Los resultados mostraron que el mayor porcentaje de emergencia se obtuvo en T5 (zanahoria chantenay-invernadero) con un 100%. En altura de follaje, T1 (zanahoria morada-invernadero) presentó el mayor desarrollo con 37,08 cm. En diámetro de raíz, T5 alcanzó 2,65 cm en promedio. La mayor altura de raíz se observó en T2 (zanahoria morada-campo abierto) con 13,49 cm, mientras que el mayor peso sin follaje correspondió a T2 con 43,90 g. El rendimiento más alto obtuvo zanahoria chantenay-invernadero (T5) con 6,19 kg/m². Se concluyó que los tratamientos con mejor desempeño fueron T2 (zanahoria morada-campo abierto) y T5 (zanahoria chantenay-invernadero), evidenciando que ambas variedades se adaptan favorablemente a sus respectivos ambientes de cultivo. Esto demuestra la viabilidad de comparar variedades en distintas condiciones de temperatura y manejo agronómico para optimizar su producción en el altiplano

1.1.2. Antecedente a nivel nacional

Huamán (2023) , desarrolló la tesis titulada "Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima", tuvo como objetivo integrar tecnología y agricultura para mejorar la distribución y dosificación del agua según las necesidades de los cultivos. El proyecto se enfocó en seleccionar materiales adecuados e implementar tecnología accesible y de bajo costo en términos de hardware, programación y operación, con el fin de facilitar su masificación. Se diseñó un prototipo de riego automatizado aplicando principios agroecológicos para la selección, diversificación, distribución y cuidado de los cultivos. Se realizaron estudios de clima, suelo y agua para identificar los cultivos adecuados y se desarrollaron los diseños agronómico e hidráulico para determinar las necesidades hídricas y el dimensionamiento de los materiales a emplear. La automatización se logró a través de un microcontrolador Arduino, conectado a sensores y actuadores que monitorean la humedad del suelo, el flujo y nivel de agua, así como el funcionamiento de válvulas y bombas. La programación del código en el IDE de Arduino incluyó librerías, funciones y variables para automatizar el sistema de riego. Los resultados obtenidos mostraron que los sensores de humedad y flujo de agua, una vez calibrados,

permitieron una gestión eficiente del agua. El sistema de riego se programó para activar las electroválvulas cuando la humedad del suelo en cualquiera de las cuatro áreas es menor al 50%, y cerrarlas cuando el nivel de humedad es igual o superior al 50%. Si todas las áreas registran una humedad superior al 50%, el sistema envía una señal al Arduino para apagar la bomba y cerrar las electroválvulas, optimizando así el consumo de agua. Finalmente, el sistema de riego fue habilitado para operar de manera automatizada, mediante la aplicación ArduinoDroid, o manualmente en caso de fallos en la placa Arduino o dificultades en el uso de la automatización. Este prototipo cumple con el objetivo de mejorar la eficiencia en el uso del agua en huertos agroecológicos urbanos, presentándose como una solución viable y escalable para zonas desérticas.

Gonzales (2022), en su tesis titulada "Diseño y automatización vía remota de un sistema de riego por goteo para el cultivo de paprika", tuvo como objetivo principal diseñar y automatizar un sistema de riego por goteo para el cultivo de paprika en la localidad de Zaña, Chiclayo-Lambayeque. El proyecto se centró en desarrollar una alternativa de automatización que permitiera el monitoreo y control remoto de válvulas hidráulicas mediante unidades de telecontrol vía radiofrecuencia. La metodología incluyó el cálculo del balance hídrico y el diseño agronómico, en el que se definieron el coeficiente de uniformidad y la eficiencia de riego esperada. Posteriormente, se realizó el diseño hidráulico de la red principal utilizando el método de velocidades. Se seleccionó un filtro de grava en el cabezal de riego debido a la procedencia del agua, y se escogió una bomba basada en los puntos de operación necesarios para los turnos de riego. Se evaluó el coeficiente de uniformidad para asegurar que coincidiera con lo propuesto en el diseño. Para la automatización, se configuró el sistema Dreon II de acuerdo con los requerimientos de la finca. Asimismo, se empleó una unidad remota de transmisión base y varias unidades esclavas para automatizar las válvulas hidráulicas, seleccionando los modelos de acuerdo con la distribución de válvulas en campo, la topografía, los turnos de riego, la distancia y el tamaño de las válvulas. Los resultados mostraron que los diámetros de la red matriz variaron entre 90 y 250 mm, con una velocidad promedio de 1.6 m/s. El caudal de filtración en los cinco tanques de grava fue de 57.75 m³/h, y la bomba seleccionada tiene una potencia de 55 kilovatios. El coeficiente de uniformidad en campo alcanzó un 94.38 %, mientras que el diseño contemplaba un 98.01 %. Además, se seleccionaron cinco unidades remotas de transmisión (RTU) que mejoraron la eficiencia en la gestión del agua para el riego del cultivo de paprika. En conclusión, el sistema de riego automatizado desarrollado por

Gonzales permitió una gestión optimizada y eficaz del agua, logrando una uniformidad y eficiencia que contribuyen al mejor rendimiento del cultivo de paprika en la región de Zaña, Chiclayo-Lambayeque.

1.1.3. Antecedente a nivel local

Ludeña (2021), en su investigación titulada "Implementación y monitoreo con tecnología Arduino para automatización de riego tecnificado en el Centro Experimental Canaán UNSCH - Ayacucho 2021", tuvo como objetivo implementar y monitorear un sistema de riego tecnificado con tecnología Arduino en dicho centro experimental, donde se identificó el problema del mal manejo del agua durante el riego y se propuso una solución automatizada. La metodología empleada fue de tipo cuantitativo y de diseño experimental aplicada, utilizando teorías tecnológicas para controlar los procesos de riego de forma más precisa. Se seleccionaron, programaron y calibraron sensores y actuadores, como el sensor de presión, el sensor de flujo, el sensor de ultrasonido, el sensor de humedad del suelo, la electroválvula y la electrobomba. Una vez instalados, los sensores y actuadores optimizaron la operación, el tiempo de trabajo y la eficiencia del riego mediante la programación de un código que permitía tanto la ejecución de las actividades de riego como el monitoreo de la humedad del suelo, asegurando una dosificación adecuada de agua para mantener el cultivo hidratado. Los resultados mostraron que los sensores y actuadores pudieron implementarse con éxito en la placa Arduino, logrando un coeficiente de determinación de 0.97 a 0.99 en la calibración de los sensores, lo que indica una alta correlación entre las variables físicas y eléctricas. La humedad en la subunidad de riego se mantuvo entre el 28.9 % (mínima) y el 32.7 % (máxima), valores cercanos a la capacidad de campo de 30.6 %. En conclusión, la automatización y el monitoreo del sistema de riego tecnificado en el Centro Experimental Canaán fueron realizados satisfactoriamente, permitiendo la interacción en tiempo real con los sensores y actuadores de la subunidad de riego desde cualquier lugar, lo cual optimiza el uso del agua y mejora la eficiencia en el riego del cultivo.

1.2 Bases teóricas

1.2.1. Invernadero

Un invernadero se puede definir como:

Según Maroto (2008) “un recinto delimitado por una estructura de madera o de metal, recubierta por vidrio o cualquier material plástico de naturaleza transparente, en cuyo interior suelen cultivarse hortalizas y plantas ornamentales en épocas durante las cuales las condiciones climáticas externas no permitirían obtener el producto deseado.”

1.2.2. Invernadero inteligente

El invernadero inteligente presenta la característica de hacer uso de sistemas embebidos los cuales son "sistemas electrónicos diseñados específicamente para realizar unas determinadas funciones, habitualmente formando parte de un sistema de mayor entidad"(De rosas, 2014).

El uso de sensores, actuadores y controladores permite la implementación de sistemas de tipo mecánico, eléctricos y electrónicos para automatizar el proceso de crecimiento, desarrollo, cuidado y monitoreo de los cultivos bajo techo. Estos sistemas se aplican de manera conjunta para generar sistemas de riego, ventilación, suministro de nutrientes, calefacción, entre otros.(Cepeda et al., 2010)

1.2.3. Hortaliza zanahoria

La zanahoria (*Daucus carota* L.) pertenece al grupo de hortalizas de raíces. Requiere de climas templados para su buen desarrollo, por lo que se adapta muy bien a los valles altos interandinos y a condiciones de invierno de la costa peruana. Tiene un alto valor nutritivo, y un alto contenido de vitamina A, que es una de las limitaciones nutricionales del país. La parte útil es la raíz la cual se consume cruda en ensaladas, licuada en jugos o cocida de varias formas. Su cultivo demanda una gran cantidad de mano de obra por lo que tiene un papel socio económico importante (INIA, 2009).

a) Morfología de la Zanahoria

- Altura de planta: 0,55 - 0,65 m
- Color del producto: Naranja intenso
- Forma: Cilíndrica
- Tamaño del producto: 12 - 20 cm (raíz)
- Peso del producto: 180 - 210 g (raíz)
- Período vegetativo:
 - ✓ 90 - 110 días (costa central)
 - ✓ 110 - 120 días (sierra central)

Figura 1. 1 Hortaliza zanahoria



Fuente: Agraria.pe, 2021.

1.2.4. Riego

Castañon (1991) , refiere que el riego es un método de suministro artificial de agua a los cultivos para que crezcan y se desarrollen; este método se usa en la agricultura para ayudar a que las plantas crezcan cuando no hay suficiente lluvia, lo cual es común en lugares áridos. También se utiliza en áreas menos secas para proporcionar a las plantas el agua que necesitan cuando se están sembrando.

1.2.5. Tipos de riego

a) Gravedad

Fuentes (1999) , El riego por superficie es un sistema de riego en donde fluye debido a la fuerza de gravedad, utilizándose la superficie del suelo agrícola como parte del sistema de distribución del agua.

Figura 1. 2 Riego por gravedad



Fuente: Instituto Mexicano de Tecnología del Agua, 2010.

b) Aspersión

Fuentes (1999), El riego por aspersión es una técnica en donde el agua se aplica en forma de lluvia por medio de unos aparatos de aspersión alimentados por agua a presión. Estos aparatos deberán asegurar el reparto uniforme sobre la superficie que se pretende regar.

Figura 1. 3 Riego por aspersión



Fuente: Grupo hidráulica, 2022

c) Goteo

Los sistemas de riego por goteo posibilitan el transporte del agua mediante una red de mangueras y emisores que suministran pequeñas cantidades de agua de manera regular. La aplicación del agua se lleva a cabo mediante dispositivos de goteo. Para llevar a cabo este proceso, es esencial contar con un sistema presurizado que dirija y distribuya el agua a través de conductos cerrados (Liotta et al, 2015).

Desde la perspectiva agronómica, se clasifican como riegos localizados debido a su capacidad para humedecer una porción específica del volumen del suelo, lo cual es suficiente para favorecer el crecimiento adecuado del cultivo. También se conocen como riegos de alta frecuencia, ya que permiten la aplicación de agua una o dos veces al día, con regularidad diaria o interdiaria, dependiendo de las características del suelo y las demandas del cultivo. La capacidad de realizar riegos frecuentes contribuye significativamente a reducir el riesgo de estrés hídrico, ya que facilita el mantenimiento de niveles óptimos de humedad en el suelo a lo largo de todo el ciclo de cultivo, mejorando así las condiciones para el desarrollo de las plantas (Liotta et al, 2015).

Figura 1. 4 Riego por goteo



Fuente: Agrohuerto, 2021

1.2.6. Componentes del sistema de riego por goteo

La mayoría de sistemas están compuestos por diversos componentes como:

1.2.6.1 Equipo de bombeo

Se compone de una o más bombas cuyas dimensiones y potencia están determinadas por la extensión del área a regar. El diseño de la bomba debe asegurar que la presión generada sea adecuada para superar las variaciones de altura y las pérdidas de carga en todo el sistema (Palomino K., 2009).

1.2.6.2. Cabezal de riego

Se trata del conjunto de componentes que controla toda la instalación y tiene la función de proporcionar presión y caudal al sistema, filtrar el agua, administrar la inyección de fertilizantes y medir el volumen (Palomino K., 2009).

1.2.6.3. Filtro

Debido a las diminutas aberturas presentes en los goteros, es esencial que el agua esté libre de partículas que puedan obstruirlas. Esto demanda una filtración de alta calidad, lo cual implica la utilización de equipos costosos y la implementación de un mantenimiento adecuado (Briones & García I., 2007).

1.2.6.4. Aparatos de control y medición

a) Medidor de Flujo

El medidor de flujo constituye una herramienta de supervisión que proporciona al operador la capacidad de cuantificar el caudal efectivo requerido por una sección de riego, al mismo tiempo que facilita la identificación de posibles inconvenientes, tales como obstrucciones o rupturas en las líneas.

b) Válvulas reguladoras de presión

Los sistemas de riego por goteo se fundamentan en la emisión uniforme de todos los dispositivos emisores. A pesar de que las bombas suministran una presión y un caudal básicos, son necesarios múltiples ajustes mínimos para garantizar el funcionamiento eficiente y seguro del sistema.

Tienen la función de mantener presiones constantes aguas abajo, incluso en presencia de variaciones ascendentes de presión. Estas válvulas no influyen directamente en el caudal de agua. Su utilidad es significativa para limitar las diferencias de presión en el campo, especialmente cuando se instalan al inicio de cada sección.

c) Válvula de control de campo

Estas válvulas, por lo general, se emplean como interruptores para encender o apagar el flujo, con reguladores de presión en línea o válvulas reductoras de presión que se usan para controlar la presión en cada sector.

d) Válvula de purga de aire / alivio de vacío

La Válvula de purga de aire es fundamental ante la acumulación de aire en las líneas principales y secundarias puede causar restricciones en el flujo y resultar en daños ocasionados por golpes de ariete. Situaciones de vacío, que pueden manifestarse en los laterales de sistemas de riego por goteo cuando este se encuentra apagado, pueden ocasionar la succión de contaminantes hacia los emisores, dando lugar a obstrucciones. Además, la formación de vacío aguas abajo de las válvulas de control al cerrarse abruptamente puede ocasionar daños a las tuberías o las propias válvulas.

1.2.6.5. Red de distribución

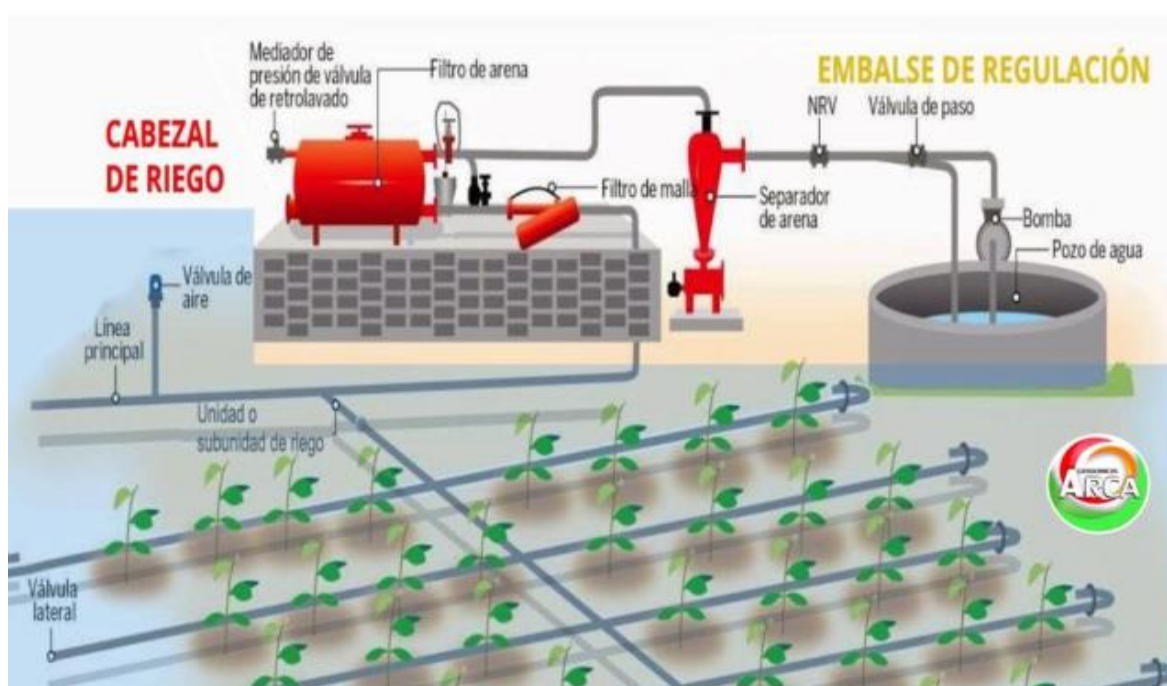
Son responsables del transporte del agua hacia las unidades de riego y suelen instalarse comúnmente en PVC o PE, aunque la preferencia recae en el PVC debido a su mayor economía, especialmente para diámetros considerables. La red principal, generalmente colocada en zanjas enterradas, se encarga de la distribución del agua en la unidad de riego y sirve como conexión

para los ramales que llevan los emisores. Optar por la instalación en PE es usualmente más conveniente, ya que proporciona mayor flexibilidad a la red y no requiere enterrarse, resultando en una opción más económica (Osorio U., 1999)

1.2.6.6. Laterales con emisores

Las líneas laterales de riego, son tuberías que se ubican dentro del cultivo a lo largo de la hilera de plantas. Normalmente son de 16 y 20 mm en función del caudal a distribuir y la longitud de riego. El material es polietileno de baja densidad.

Figura 1. 5 Sistema de riego por goteo



Fuente: Agroquímicos Arca,2020

1.2.7. Tiempo de riego

Los sensores instalados en diferentes profundidades permiten monitorear el nivel de humedad en todo el perfil del suelo, desde la superficie hasta las raíces. Esto ayuda a determinar el tiempo de riego con mayor precisión.

Según Evans et al. (2013), este tipo de monitoreo continuo permite ajustar el tiempo de riego de manera precisa, asegurando que se aplique el agua cuando realmente es necesario.

1.2.8. Frecuencia o intervalo de riego

La frecuencia o intervalo de riego se puede ajustar automáticamente en función de los umbrales predefinidos de humedad del suelo.

Según Jones (2004) sugiere que “Los sistemas automatizados basados en sensores ajustan la frecuencia del riego cuando la humedad cae por debajo de un nivel crítico, asegurando un suministro adecuado de agua para el cultivo”.

1. 1.2.9. Automatización del sistema riego por goteo

La automatización del sistema de riego se revela como una alternativa más ventajosa, ya que posibilita la programación y gestión secuencial de las operaciones, proporcionando un control más preciso y permitiendo la toma de decisiones respecto al inicio o finalización del riego (Liotta et al, 2015).

Gracias a los sistemas de riego inteligentes, es posible estimar y medir la disminución de la humedad en las plantas haciendo posible operar un sistema de riego y así poder restaurar y minimizar el uso excesivo de agua (Darshna et al., 2015)

1.2.9.1. Arduino

Según Artero (2013), un Arduino se define como “Una placa de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines- hembra (los cuales están unidos internamente a las pastillas del microcontrolador) que permiten conectar allí de forma sencilla y cómoda diferentes sensores y actuadores.

a) Placa Arduino Mega 2560 Rev3.

Es una tarjeta de desarrollo mucho más potente que Uno R3, es utilizada en aplicaciones con mayor necesidad de recursos como entradas/salidas, memoria e interfaces. Mega 2560 Rev3 es una tarjeta de desarrollo que utiliza el microcontrolador ATmega2560.

Figura 1. 6 Placa Arduino



Fuente: Electromania,2024

Tabla 1. 1 Especificaciones Técnica de la placa Arduino Mega 2560 Rev3

Especificaciones Técnicas	
Microcontrolador	ATmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (Recomendado)	7-12V
Voltaje de entrada (Límite)	6-20V
Pines de E/S Digital	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente CC por PIN de E/S	20 mA
Corriente Dc para PIN 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB utilizados por el gestor de arranque (Bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52 mm
Ancho	53.3 mm
Peso	37g

Fuente: Electromania,2024

1.2.9.2. Microcontrolador

Según De rosas (2014), el término Microcontrolador tienen por significado “pequeño en tamaño y en maniobrar o controlar procesos los cuales son definidos mediante programación”. Esta es constituido en su interior por tres principales unidades funcionales de una computadora, las cuales son: unidad central de procesamiento, memoria y periféricos de entrada y salida.

1.2.9.3. Entorno de desarrollo integrado

Se trata de un conjunto de herramientas de software que posibilita la creación y prueba de programas, ofreciendo la capacidad de editar los programas propios en caso de que se hayan cometido errores en el "sketch" (la plataforma de edición). En caso de que la edición sea correcta, se procede a grabar el programa en la memoria del microcontrolador de la placa Arduino, permitiendo que la placa funcione de manera autónoma como ejecutor del programa (Torrente, 2013).

Tabla 1. 2 Interfaz IDE Arduino

Pestañas	Descripción
Archivo	Se puede: crear, abrir un archivo, abrir archivos utilizados, ejemplos de programas de Arduino, elegir letra, tamaño, idioma, etc.
Editar	Se puede: copiar, pegar, comentar código, aumentar sangría y fuente código
Programar	Se puede: verificar escritura de código, cargar el código a la placa, buscar y descargar librerías.
Herramientas	Se puede: seleccionar tipo de placa Arduino, tipo de puerto de conexión, visualizar el monitor serial.
Ayuda	Se puede: solicitar ayuda en línea

1.2.9.4. Lenguaje de programación

El lenguaje es un sistema de comunicación que posee una determinada estructura, contenido y uso. La programación es un vocabulario propio de la informática, el procedimiento de escritura del código fuente de un software.

Es un conjunto simplificado de instrucciones derivadas de los lenguajes C y C++ diseñadas específicamente para facilitar el desarrollo de programas en microcontroladores. Dado que los lenguajes C y C++ utilizados por Arduino son difíciles de aprender y dominar debido a su extenso contenido y detalles que pueden resultar innecesarios para los programadores aficionados que desean realizar proyectos con Arduino, el "lenguaje" Arduino actúa como un medio para simplificar la comprensión y evitar la complejidad de C y C++ (Torrente, 2013).

Figura 1. 7 Lenguaje de programación



Fuente: Administración de sistemas, 2024

1.2.9.5. Librería de Arduino

Es un conjunto de funciones diseñado para simplificar la programación y abstractizar los detalles técnicos de un hardware, haciendo que el trabajo en el código sea más fácil y accesible. Estas funciones incluyen la capacidad de controlar pantallas LCD, leer y escribir datos en la memoria EEPROM, acceder a tarjetas SD, establecer conexiones entre Arduino y programas externos, entre otras (Torrente, 2013).

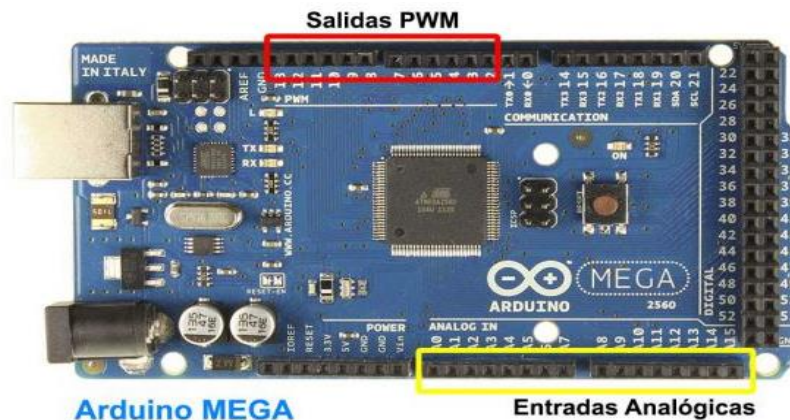
1.2.9.6. Entrada y salida digital

Para Arduino la entrada y salida digital es el pin donde irá conectada un sensor o actuador que detecte algún cambio o variación del mundo real, los valores detectados están entre 0V y 5V.

1.2.9.7. Entrada y salida analógica

Las señales analógicas generalmente son de tipo continua con lo cual se tiene que necesitar un convertidor analógico a digital para poder leerlas. Estos valores son emitidos por sensores de temperatura, presión, caudal, humedad, intensidad de corriente, intensidad lumínica, etc.

Figura 1. 8 Entradas y salidas analógicas del Arduino



Fuente: Zonamake,2014

1.2.9.8. Lector de MicroSD

Es un dispositivo diseñado para leer tarjetas de memoria MicroSD, que son un tipo de tarjetas de almacenamiento de tamaño reducido, también permite la conexión a otros dispositivos, como computadoras, para transferir archivos de manera rápida y eficiente.

1.2.9.9. Fuente de alimentación conmutada AC/DC

Esta fuente de alimentación tiene la función de transformar la corriente alterna (AC) en corriente continua (DC) mediante el uso de transistores conmutadores. Destaca por su diseño compacto, reducido tamaño y una eficiencia notable del 81%. Con una potencia máxima de 250W, acepta un voltaje de entrada de 200V-240V AC, proporcionando un voltaje de salida de 12V DC y permitiendo una corriente máxima de 20A. Esta fuente es especialmente indicada para alimentar dispositivos como motores DC, servomotores, luces LED, entre otros (Naylamp Mechatronics., 2021).

1.2.9.10. Sensores

Los sensores poseen la capacidad de identificar magnitudes físicas o químicas, denominadas variables de instrumentación, con el propósito de convertirlas posteriormente en variables eléctricas que resulten legibles.

El sensor de humedad en este caso mide la temperatura del cultivar y la humedad relativa. Con la incorporación de circuitos integrados le permite a este enviar una señal acondicionada, también cuentan con un punto sensible que capta las señales del ambiente. Este punto es fabricado con polímeros y electrodos de platino. La mayoría son calibrados por láser, tiene un buen rendimiento y un margen de error mínimo (Goyal, 2017).

Tabla 1. 3 Tipos de sensores

Tipo	Descripción	Imagen
Humedad del suelo	Compatible con Arduino, Mide la humedad en función de la resistencia eléctrica , además posee unas puntas anticorrosivas que alarga su vida útil.	
Presión	Es un instrumento compuesto por un elemento detector de presión con el que se determina la presión real aplicada al sensor (utilizando distintos principios de funcionamiento) y otros componentes que convierten esta información en una señal de salida.	
Flujo de agua	Compatible con Arduino, cuantifica el flujo volumétrico de manera aproximada en L/min, l/h o m3/h; se coloca directamente en las tuberías.	
Ultra sonido	Compatible con Arduino, mide la distancia a un rango de 2 a 450 cm por medio de señales de sonido.	

En el desarrollo de la tesis se usaron los siguientes modelos de sensores:

a) Sensor ultrasonido HC-SR04

Es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio.

Figura 1. 9 Sensor Ultrasonido HC-SR04



Fuente: Naylamp mechatronics,2024

Tabla 1. 4 Especificaciones Técnicas del Sensor Ultrasonido HC-SR04

Especificaciones Técnicas	
Voltaje de Operación:	5V DC
Corriente de reposo:	< 2mA
Corriente de trabajo:	15mA
Rango de medición:	2cm a 450cm
Precisión:	+/- 3mm
Ángulo de apertura:	15°
Frecuencia de ultrasonido:	40KHz
Duración mínima del pulso de disparo TRIG (nivel TTL):	10 μS
Duración del pulso ECO de salida (nivel TTL):	100-25000 μS
Dimensiones: 45*20*15 mm	45*20*15 mm
Tiempo mínimo de espera entre una medida y el inicio de otra	(recomendable 50ms)
	20ms

Fuente: Naylamp mechatronics,2024

Conexión	
VCC	(+5V DC)
TRIG	(+5V DC)
ECHO	(Recepción del ultrasonido)
GND	(Tierra: 0V)

Fuente: Naylamp mechatronics,2024

b) Sensor presión Hk3025

Es un sensor que trabaja en el rango de 0 a 5 MPa, un rango común para aplicaciones experimentales en neumática y sistemas de presión constante. La salida del sensor es un voltaje analógico que varía linealmente desde 0.5V para 0 MPa hasta 4.5V para 5 Mpa. Esta salida analógica es compatible con las entradas analógicas (ADC) de Arduino.

Figura 1. 10 Sensor de presión HK-3025



Fuente: Naylamp mechatronics,2024

Tabla 1. 5 Especificaciones Técnicas del Sensor presión HK 3025

Especificaciones Técnicas	
Voltaje de operación:	5.0 VDC
Voltaje de salida:	5.0 VDC
Corriente de funcionamiento:	≤ 10 mA
Rango de presión de trabajo:	0-5MPa (0-50 Bar ó 0-725psi)
Presión Max:	7.5 Mpa
Presión de ruptura:	10.0 Mpa
Tiempo de respuesta: ≤ 2.0 ms	≤ 2.0 ms
Error de medición:	± 1 %FSO

Error por temperatura:	+3.5 %FSO
Temperatura de trabajo:	0-85°C
Temperatura de almacenamiento:	-20 - 85°C
Protección: IP65	IP65
Rosca de conexión: Macho G1/4 (BSP):	Rosca Whitworth para tubos cilíndrica)
Fluido de trabajo:	Gas, Aire comprimido, Agua, Aceite.
Dimensiones:	26*26*60 mm
Conexiones:	Rojo (5V), Negro(tierra), Amarillo (Salida)

Fuente: Naylamp mechatronics,2024

c) Sensor flujo de agua 2" YF-DN50

El sensor sirve para medir caudal de agua en tuberías de 2" de diámetro. Es un caudalímetro electrónico de tipo turbina. Compatible con sistemas digitales como Arduino, Pic, etc. El sensor posee tres cables: rojo (VCC: 5VDC), negro (tierra) y amarillo (salida de pulsos del sensor de efecto Hall).

Figura 1. 11 Sensor de presión HK-3025



Fuente: Naylamp mechatronics,2024

Tabla 1. 6 Especificaciones técnicas del Sensor flujo de agua 2" YF-DN50

Especificaciones Técnicas	
Modelo:	YF-DN50
Voltaje de operación:	5V - 18V DC
Consumo de corriente:	15mA máx.
Salida:	Onda cuadrada pulsante

Rango de Flujo:	10 - 200L/min
Volumen promedio por pulso:	83.3mL
Pulsos por litro:	12
Factor de conversión:	0.2
Rosca externa:	2" NPS
Presión de trabajo máx.:	1.75MPa (17 bar)
Temperatura de trabajo:	5°-80°C
Resistencia de insulación:	>100M Ohm
Conexiones:	Rojo (5V), Negro (tierra), Amarillo (Salida)

Fuente: Naylamp mechatronics,2024

d) Sensor DHT11 de temperatura

Es un sensor que mide la temperatura, y muestra los datos mediante una señal digital en el pin de datos.

Figura 1. 12 Sensor DHT11 de temperatura



Fuente: Programar fácil.com, 2024

Tabla 1. 7 Especificaciones técnicas del sensor DHT11 de temperatura

Especificaciones técnicas	
Modelo	DHT11
Consumo	2,5 mA
Señal de salida	Digital
Temperatura	

Rango	de 0°C a 50°C
Precisión	a 25°C ± 2°C
Resolución	1°C (8-bit)
Humedad	
Rango	de 20% RH a 90% RH
Precisión	entre 0°C y 50°C ± 5% RH
Resolución	1% RH

Fuente: Programar fácil.com, 2024

Pines	
Alimentación:	5V (VCC)
Datos	(DATA)
No Usado	(NC)
Tierra	(GND)

Fuente: Programar fácil.com, 2024

e) Módulo higrómetro sensor de humedad del suelo

Sensor de humedad del suelo con sonda de dos terminales metálicos y módulo de procesamiento. Proporciona salidas analógica y digital para monitoreo. Las puntas están tratadas para resistir la oxidación y prolongar su vida útil.

Figura 1. 13 Módulo higrómetro sensor de humedad del suelo



Fuente: Electromanía,2024

Tabla 1. 8 Especificaciones técnicas del Módulo higrómetro

Especificaciones Técnicas	
Modelo:	HD-38
Corriente de operación:	< 20 ma
Corriente de salida:	< 30 ma
Voltaje de alimentación:	3.3-12VDC

Puerto de conexión:	Vcc, GND, DO, AO
DO:	Salida digital
AO:	Salida analógica (voltaje)




Fuente: Naylamp mechatronics,2024

1.2.9.11. Actuadores

Son dispositivos responsables de controlar el flujo de agua, regular la energía entrante y saliente, así como activar o desactivar el flujo de electricidad de los aparatos para garantizar su correcto funcionamiento.

Carit et al. (2002), menciona que los actuadores son dispositivos capaces de efectuar acciones físicas, determinadas y controladas. Estas acciones pueden tener un movimiento lineal o un movimiento circular (rotatorio) de acuerdo a la aplicación.

Tabla 1. 9 Tipos de actuadores

Tipo	Descripción	Imagen
Modulo Relé	Dispositivo permite controlar el voltaje reduciéndolo para poder ser utilizado por el Arduino.	
Válvula solenoide	Electroválvula que funciona dejando pasar o cerrando el flujo, por medio de los impulsos eléctricos emitidos desde un microcontrolador.	
Bomba de agua	Bomba tipo centrifuga compatible con Arduino, con entrada y salida tipo rosca, silenciosa y bajo consumo de energía	

Nota :Se muestran actuadores compatibles con el Arduino

En el desarrollo de la tesis se usaron los siguientes modelos de actuadores:

a) **Electroválvula Serie PGA 2” de 24 VAC NPT Hembra**

Es un actuador utilizado comúnmente en sistemas de riego automático para controlar el flujo de agua.

Figura 1. 14 Electroválvula Serie PGA 2”



Fuente: AibiTech,2024

Tabla 1. 10 Especificaciones técnicas de la electroválvula Serie PGA 2”

Especificaciones Técnicas	
Caudal:	0,5 a 34,05 m ³ /h
Presión:	1,0 a 10,4 bar (23° C)
Temperatura:	1,0 a 10,4 bar (23° C)
Solenoides:	24 VAC - 50 Hz
Corriente de arranque:	0,41 A (9,9 VA)
Corriente de régimen:	0,23 A (5,5 VA)

Fuente: AibiTech,20

CAPÍTULO II METODOLOGÍA

2.1. Aspectos generales del proyecto

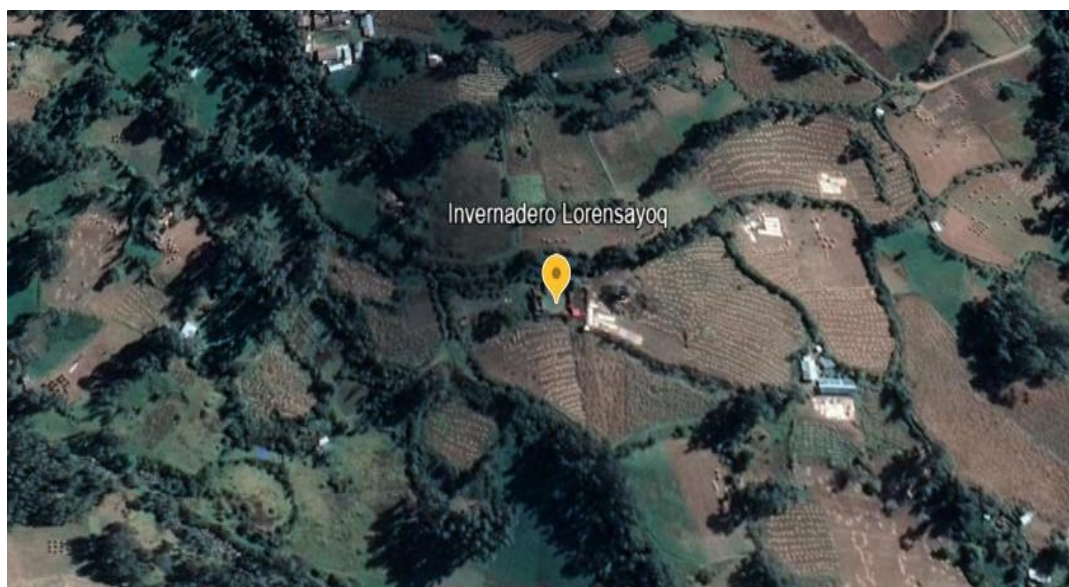
2.1.1. Ubicación

a) Ubicación geográfica

La ubicación geográfica del área de estudio está ubicada en las coordenadas UTM WGS 84, zona 18 L:

- Este: 603913.20 m
- Norte: 8571470.34 m
- Altitud: 3266 m.s.n.m

Figura 2. 1 Ubicación Satelital del Invernadero

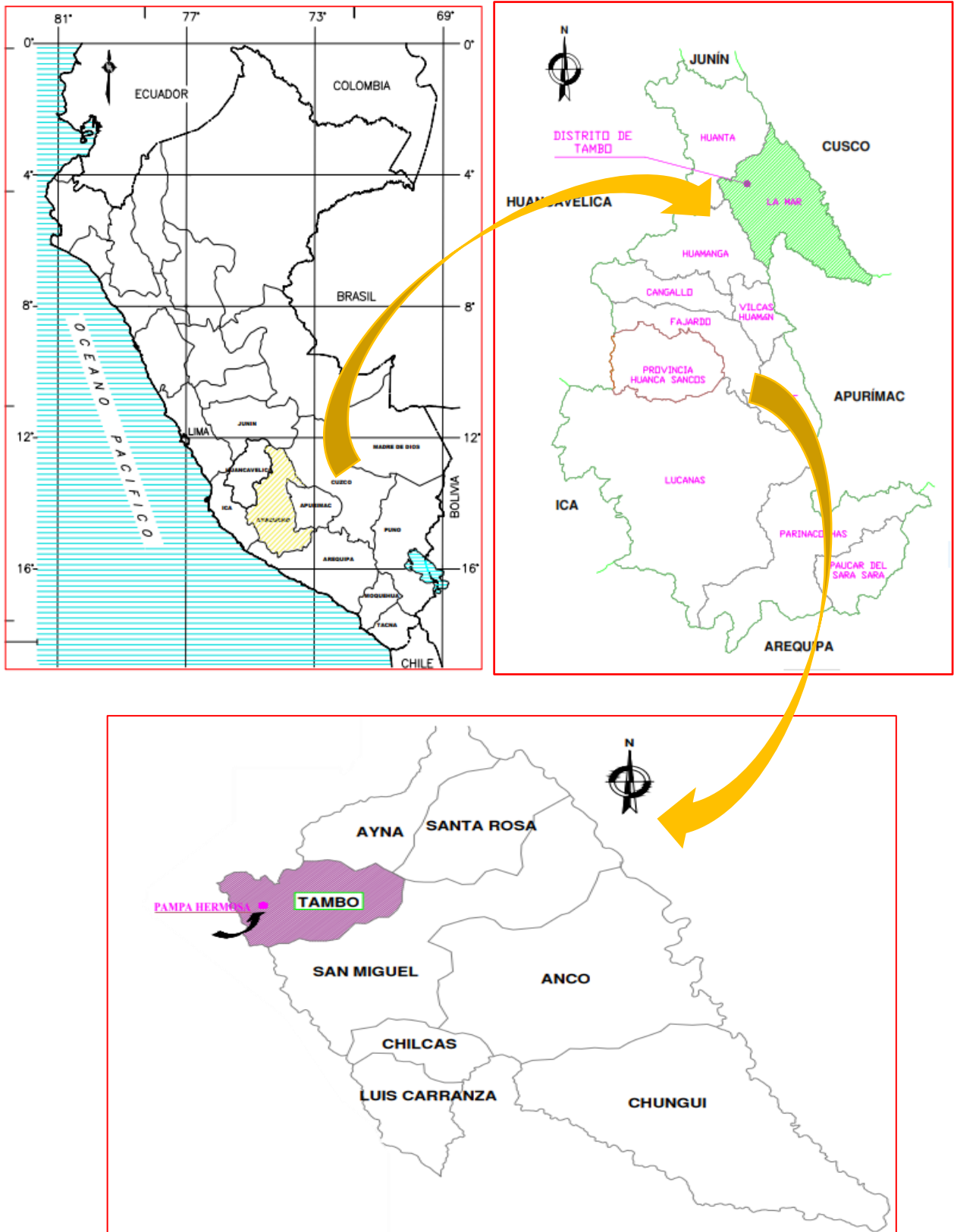


Fuente: Google earth

b) Ubicación política

- Departamento : Ayacucho
- Provincia : La Mar
- Distrito : Tambo
- Localidad : Pampa Hermosa
- Anexo : Lorensayoc

Figura 2. 2 Ubicación política del proyecto



2.1.2. Vías de acceso

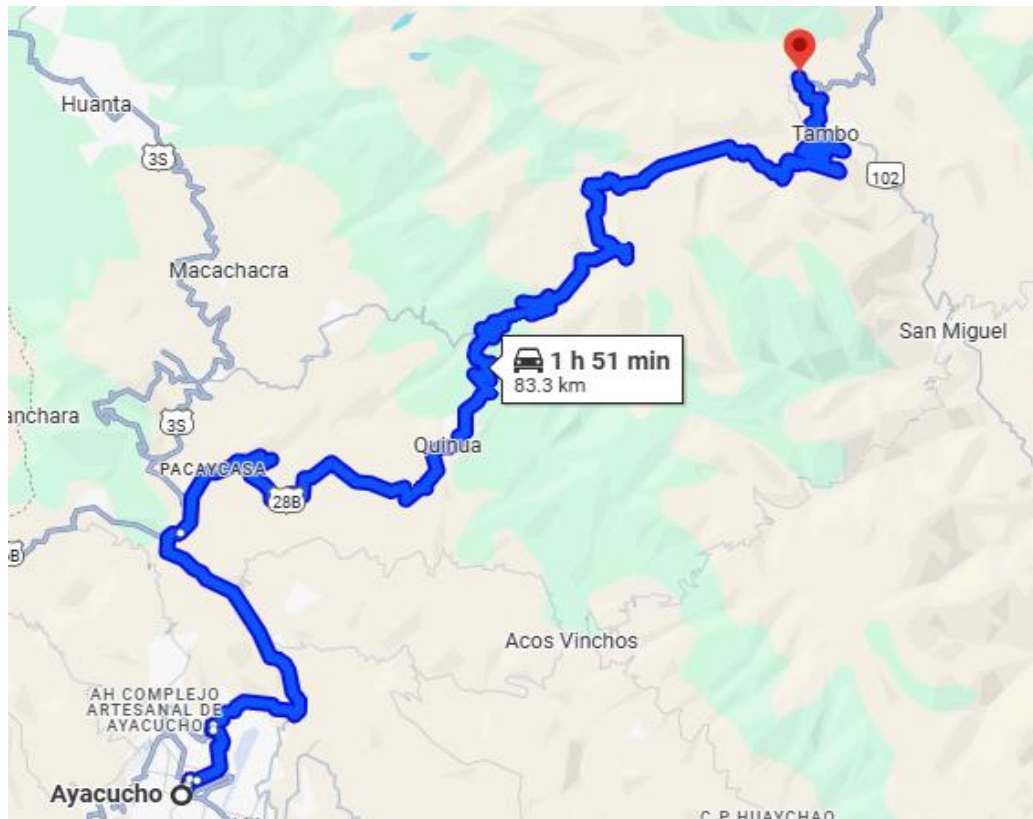
El acceso al proyecto, se realiza mediante la utilización de la vía terrestre que parte desde Ayacucho a Tambo es de 71 Km, y de Tambo hasta Pampa Hermosa es de 8.8 km. El tiempo promedio de viaje es de 3 horas.

Tabla 2. 1 Vías de comunicación al Distrito Tambo-Pampa Hermosa

Descripción	Distancia	Tiempo	Tipo de Vía
Ayacucho-Quinua	37 km	55 min	Asfaltado
Quinua-Tambo	34 km	1 h 10 min	Asfaltado
Tambo-Acco	5.2 km	40 min	Asfaltado
Acco -Pampa Hermosa	3.6 km	15 min	Trocha Carrozable
Total	79.8	3 horas	

Fuente : DRTCA – Ayacucho

Figura 2. 3 Mapa de acceso al proyecto



Nota. Fuente : Google Maps

2.1.3. Clima

El clima es uno de los aspectos de suma importancia ya que repercute directamente en los cultivos. El área geográfica del proyecto presenta un clima predominantemente templado y seco con temperatura media anual de 18°C, con temperaturas máxima media de 24.90 °C y temperaturas medias menores a 5°C, además con una precipitación promedio anual de 831.20 mm, que se registra con la Estación meteorológica de Quinua. Las precipitaciones se dan en el período de verano siendo los meses de intensidad, diciembre enero y febrero disminuyendo paulatinamente en el mes de marzo.

Figura 2. 4 Pampa Hermosa - Lorensayoc



2.1.4. Antecedentes del sistema de riego

Esta investigación se realizó en un sistema de riego existente del Invernadero ubicado en Anexo de Lorensayoc de la Localidad Pampa Hermosa, la cual fue aprovechado para su automatización, para lo cual fue necesario realizar el cambio de los accesorios del cabezal de riego ya que presentaban fugas la cual traía consigo las pérdidas de presión y posteriormente la poca entrega de agua, por ende, los cultivos se veían afectados en su desarrollo. Para la automatización del sistema de riego se contó con los siguientes componentes como son:

- Línea de purga.
- Reservorio de 25 m³.

- Filtro de anilla de 2 pulgadas.
- Válvula de control de 2 pulgadas.
- Cintas de riego de clase 15 mil.
- Laterales de riego principal 2 pulgadas

Las cuales fueron examinados para una correcta implementación de los sensores y actuador permitiendo monitorear en tiempo real mediante el uso del aplicativo instalado en un celular.

a) Fuente de agua

La fuente de abastecimiento de agua es un reservorio de forma de tronco pirámide revestido de geomembrana con una pendiente del 1.5 %, con una entrada de caudal de 1.74 l/s del rio Pampa Hermosa- Acco, así mismo el reservorio tiene las siguientes medidas de fondo 4m largo, ancho 2 m y la medida del espejo del agua son largo 6 m y ancho 4 m con una altura de 1.5 m la cual hace que tenga un volumen de agua de 25 m³ y un bordo libre de 30 cm, pero no contaba con un rebose.

Figura 2. 5 Reservorio de agua revestido con geomembrana



b) Cabezal de riego

El cabezal de riego está compuesto por los siguientes componentes como filtro de anilla, válvula de control, cabe mencionar que el cabezal de riego funciona en forma manual.

Figura 2. 6 Cabezal de riego



c) Área de riego

El área o subunidad de riego es de 77.14 m² con un largo de 13.3 m y un ancho de 5.8 m.

Figura 2. 7 Toma de medidas de la subunidad de riego



Figura 2. 8 Invernadero



2.2. Materiales y equipos

2.2.1. Gabinete

- Papel bond.
- Laptop de Windows 10
- Impresora

- Software

- Arduino IDE 1.8.19.
- Blynk.IO.
- AutoCAD 2020.
- Microsoft Word 2019.
- Microsoft Excel 2019.
- Lucidchart

2.2.2. Campo

- Libreta de campo.
- Lapicero.
- Flexómetro.
- Cemento +ladrillo + Agregados
- Caseta de control
- Semilla de zanahoria (Chantenay Red Core)
- Herramientas manuales + eléctricos

-Equipo para la automatización

- Sensor de ultrasonido HC-SR04.
- 02 sensores de presión Hk3025 5.0MPA.
- Sensor de flujo de agua 2" YF-DN50.
- Sensor DHT11 de temperatura
- 02 módulo higrómetro sensor de humedad del suelo

-Equipo eléctrico

- Modulo Relé de 4 canales 5v.
- Placa Arduino Mega 2560 Rev3.
- Modulo Ethernet shield W5100.
- Módulo Lector de memoria Micro SD Card 74LVC125A

- Cable Dupont macho macho.
- Cable Dupont macho hembra.
- Cable Dupont hembra hembra.
- Cable internet Red UTP Cat. 5e
- Protoboard Blanco de 400 puntos.
- LED rojo + Resistencia + pulsador de Arduino
- Electroválvula de 24 VAC
- 02 cable eléctrico #14 AWG rollo x100m
- Tablero metálico 1P 67 Hermética (500*400*200mm)
- Canaleta ranurada 25*25 PVC
- 02 interruptores termomagnéticos 2*20A ABB
- Contactor
- Relé térmico
- Selector de 3 posiciones MOA
- Pulsador doble partida/ stop
- Transformador de 220 VAC a 24 VAC.
- Pack bornero de conexión $4mm^2/6mm^2$
- Cintillo nylon 110*25 mm
- Paquete tornillo wafer phillips
- Pack de terminales tipo pm
- Espiral PVC 4mm Ø
- Conector hermético de 1/2"
- 02 cajas de paso 15*15 PVC
- 01 caja metálica de 10*10
- 05 cinta eléctrica negro
- 2 tomacorrientes dobles
- 02 cargadores

-Calibración

- 1 medidor de presión/ manómetro digital
- Medidor de flujo de agua digital
- Laboratorio / medir humedad del suelo
- Termohigrómetro

-Accesorios

- 03 enlaces Rosca/Hembra 2"
- 02 niple PVC
- 02 reducciones bushines 1 a 3/4"
- 02 reducciones bushines 2 a 1"
- 02 abrazaderas PVC
- Unión universal de 2".
- 02 bushines de 16 mm
- 02 uniones de metal
- Codo PVC de 90° de 1".
- 10 Cintas teflones

-Dispositivo móvil

- Celular smartphone

- Red

- Internet + Router Modem Para Chip 4g Liberados

-Almacenamiento

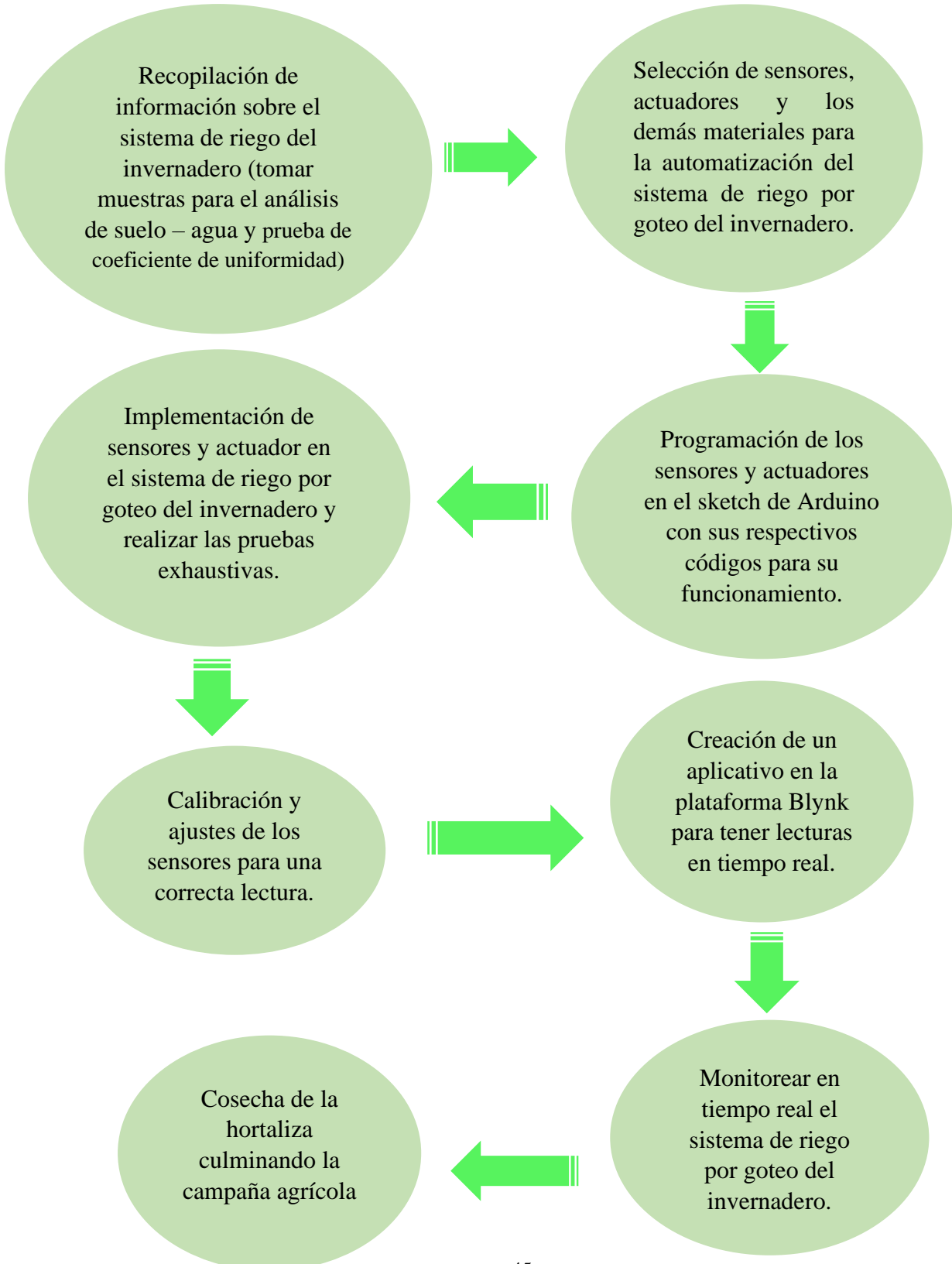
- Memoria de 64 GB

2.3. Problemas específicos

1. ¿Cuáles son los componentes que se utilizan al automatizar mediante microcontrolador Arduino para el riego por goteo?
2. ¿De qué manera los sensores enviaran la información obtenida al microcontrolador Arduino?
3. ¿Cómo es la programación del Arduino en el sistema automatizado en riego por goteo?
4. ¿Cuál será el rendimiento de la hortaliza de zanahoria finalizado la campaña agrícola?

2.4. Metodología

El presente trabajo se inició desde el 20 de noviembre del 2024 hasta 04 de abril del 2025 para lo cual se siguió las siguientes etapas que a continuación se menciona:



2.4.1. Recopilación de información sobre el sistema de riego del invernadero (tomar muestras para el análisis de suelo – agua y prueba de coeficiente de uniformidad)

Es importante destacar que el aprovechamiento eficiente del agua es fundamental para garantizar la salud de las plantas, optimizar su rendimiento y maximizar la rentabilidad. Para ello, se requieren tecnologías agrícolas innovadoras, como la implementación de sistemas basados en Arduino. Estas tecnologías no solo favorecen una mayor productividad, sino que también reducen los costos asociados a insumos como materiales, energía, mano de obra y el agua, un recurso natural de gran valor.

El punto de partida para la implementación de esta tecnología fue una visita de reconocimiento al invernadero de Lorensayoc, ubicado en Pampa Hermosa. Durante esta inspección, se evaluó el estado actual del sistema de riego por goteo manejado de forma manual por el dueño Sr. Oscar, donde identificaron diversos problemas, entre ellos un control inadecuado del nivel de agua en el reservorio, presión y caudal. Además, se observó una gestión poco eficiente del agua durante el riego, evidenciada en casos de sobreestimación y subestimación del recurso, debido a la falta de conocimiento sobre la humedad del suelo y el tiempo óptimo de riego.

Como parte del estudio, se recolectaron muestras de suelo para análisis físico y caracterización, los cuales se realizaron en el laboratorio “AGROLAB”. Los resultados indicaron que el suelo tiene:

Tabla 2. 2 Resultado del análisis Físico de suelos

Clase Textural	Capacidad de campo (%)	Punto de marchitez (%)	Análisis granulométrico		
			Arena(%)	Limo(%)	Arcilla (%)
Franco Arcilloso	27.45	14.09	41	21	38

Figura 2. 9 Toma de muestra de suelo para el análisis en laboratorio



Adicionalmente, se tomó una muestra de agua del reservorio para un análisis físico y químico en el laboratorio. Este análisis proporcionó información detallada sobre las características del agua.

Figura 2. 10 Análisis de la muestra de agua y suelo



Existen múltiples factores que, en conjunto, influyen en la calidad del agua destinada al riego de las plantas. Entre ellos, los componentes químicos del agua pueden tener un impacto directo en el desarrollo de las plantas, ya sea causando toxicidad o generando deficiencias, o indirectamente al alterar la disponibilidad de nutrientes para las plantas.

Cabe resaltar que los resultados del Laboratorio AGROLAB clasificó: Agua de salinidad media válida para riego.

Tabla 2. 3 Resultado del análisis del agua

Análisis de Agua	
N° Laboratorio	A01428
N° de campo	Río Acco
pH	8.34
C.E. ($\mu\text{s}/\text{Cm}$)	594
CATIONES (meq/l)	
Calcio	3.35
Magnesio	1.58
Potasio	0.12
Sodio	0.89
Suma de Cationes (meq/l)	5.94

ANIONES (meq/l)	
Nitratos	0.62
Carbonatos	0.13
Bicarbonatos	2.79
Cloruros	0.97
Sulfatos	1.18
Suma de Aniones(meq/l)	5.69
Na (%)	14.98
Boro(ppm)	0.02
RAS	0.57
Sales Solubles Totales (ppm)	380.16
Dureza (G.H.F.)	24.64
CLASIFICACIÓN:	C2-S1

Además, la eficiencia con la que se distribuye el agua en un sistema de riego, representa el grado de homogeneidad en la aplicación del agua sobre el suelo y es fundamental para garantizar un crecimiento uniforme de los cultivos.

Figura 2. 11 Prueba de coeficiente de uniformidad



Por ello se logró verificar antes y después de la automatización, la distribución uniforme del sistema de riego por goteo mediante la fórmula de J.E. Christiansen:

$$CU = 100 * \left(1 - \frac{\sum_i^n |Xi - X|}{n * X} \right) \dots \dots \dots Ecuación (1)$$

CU: Coeficiente de uniformidad %

n: Números de puntos de control

X: Promedio de todos los datos (ml)

Xi: Cantidad de gotas en cada punto de control (ml)

Duración del riego manual (antes)

Horario inicio: 10:14 am

Horario finalizado: 10:18 am

Tabla 2. 4 Datos recolectados del CU en riego sin automatizar

Recipiente	Cantidad (ml)	Xi -X
1	100	15.69
2	100	15.69
3	108	23.69
4	75	9.31
5	76	8.31
6	88	3.69
7	82	2.31
8	60	24.31
9	80	4.31
10	64	20.31
11	58	26.31
12	100	15.69
13	70	14.31
14	90	5.69
15	60	24.31
16	62	22.31
17	68	16.31
18	82	2.31
19	90	5.69
20	90	5.69
21	82	2.31
22	90	4.17
23	98	13.69
24	100	15.69
25	98	13.69
26	80	4.31

27	70	14.31
28	86	1.69
29	90	5.69
30	92	7.69
31	84	0.31
32	80	4.31
33	94	9.69
34	96	11.69
35	100	15.69
36	92	7.69
Promedio	84.31	398.86

$$CU = 100 * \left(1 - \frac{398.86}{36 * 84.31}\right)$$

$$CU = 86.8 \%$$

Duración del riego automatizado (después)

Horario inicio: 1:42 pm

Horario finalizado: 1:48 pm

Tabla 2. 5 Datos recolectados del CU en riego automatizado

Recipiente	Cantidad (ml)	Xi -X
1	140	15.83
2	130	5.83
3	138	13.83
4	132	7.83
5	130	5.83
6	130	5.83
7	116	8.17
8	132	7.83
9	126	1.83
10	120	4.17
11	100	24.17
12	130	5.83
13	138	13.83
14	100	24.17
15	138	13.83
16	136	11.83
17	138	13.83

18	134	9.83
19	110	14.17
20	130	5.83
21	140	15.83
22	120	4.17
23	98	26.17
24	140	15.83
25	138	13.83
26	112	12.17
27	140	15.83
28	140	15.83
29	98	26.17
30	98	26.17
31	108	16.17
32	110	14.17
33	118	6.17
34	126	1.83
35	116	8.17
36	120	4.17
Promedio	124.17	437.00

$$CU = 100 * \left(1 - \frac{437}{36 * 124.17}\right)$$

$$CU = 90.2 \%$$

2.4.2. Selección de sensores, actuador y demás materiales para la automatización del sistema de riego por goteo del invernadero.

La implementación de un sistema de riego por goteo automatizado en un invernadero, basado en tecnologías IoT, requirió una selección cuidadosa de sensores, actuadores y materiales que no solo garanticen la funcionalidad, sino también la sostenibilidad y la asequibilidad del sistema.

Los sensores de humedad del suelo juegan un papel fundamental para monitorear en tiempo real las condiciones del sustrato y activar el riego solo cuando sea necesario. La elección de sensores de bajo costo, como los basados en tecnología capacitiva, permite mantener el equilibrio entre precisión y accesibilidad económica.

Para el control del sistema, plataformas de hardware libre como Arduino presenta como opciones viables debido a su versatilidad y compatibilidad con múltiples componentes IoT.

Estas plataformas permiten la integración con actuadores, como válvulas solenoides, que regulan la apertura y cierre de las líneas de riego, y bombas de agua que aseguran un flujo adecuado en la red de goteo. Además, es crucial considerar materiales complementarios como tuberías de polietileno, goteros regulables y sistemas de filtración para garantizar un suministro eficiente y libre de obstrucciones.

La automatización de un sistema de riego por goteo mediante un microcontrolador Arduino involucra varios componentes electrónicos y sensores para gestionar el riego de manera eficiente. A continuación, se muestran los componentes utilizados en el proyecto:

- Sensor de ultrasonido HC-SR04.
- Sensor de presión Hk3025 5.0MPA.
- Sensor de flujo de agua 2" YF-DN50.
- Sensor DHT11 de temperatura
- Módulo higrómetro sensor de humedad del suelo
- Módulo Relé de 4 canales 5v.
- Placa Arduino Mega 2560 R3.
- Módulo Ethernet shield W5100.
- Electroválvula de 24 VAC
- LED rojo + Resistencia + pulsador de Arduino
- Componentes eléctricos
- Internet

Figura 2. 12 Esquema del sistema eléctrica

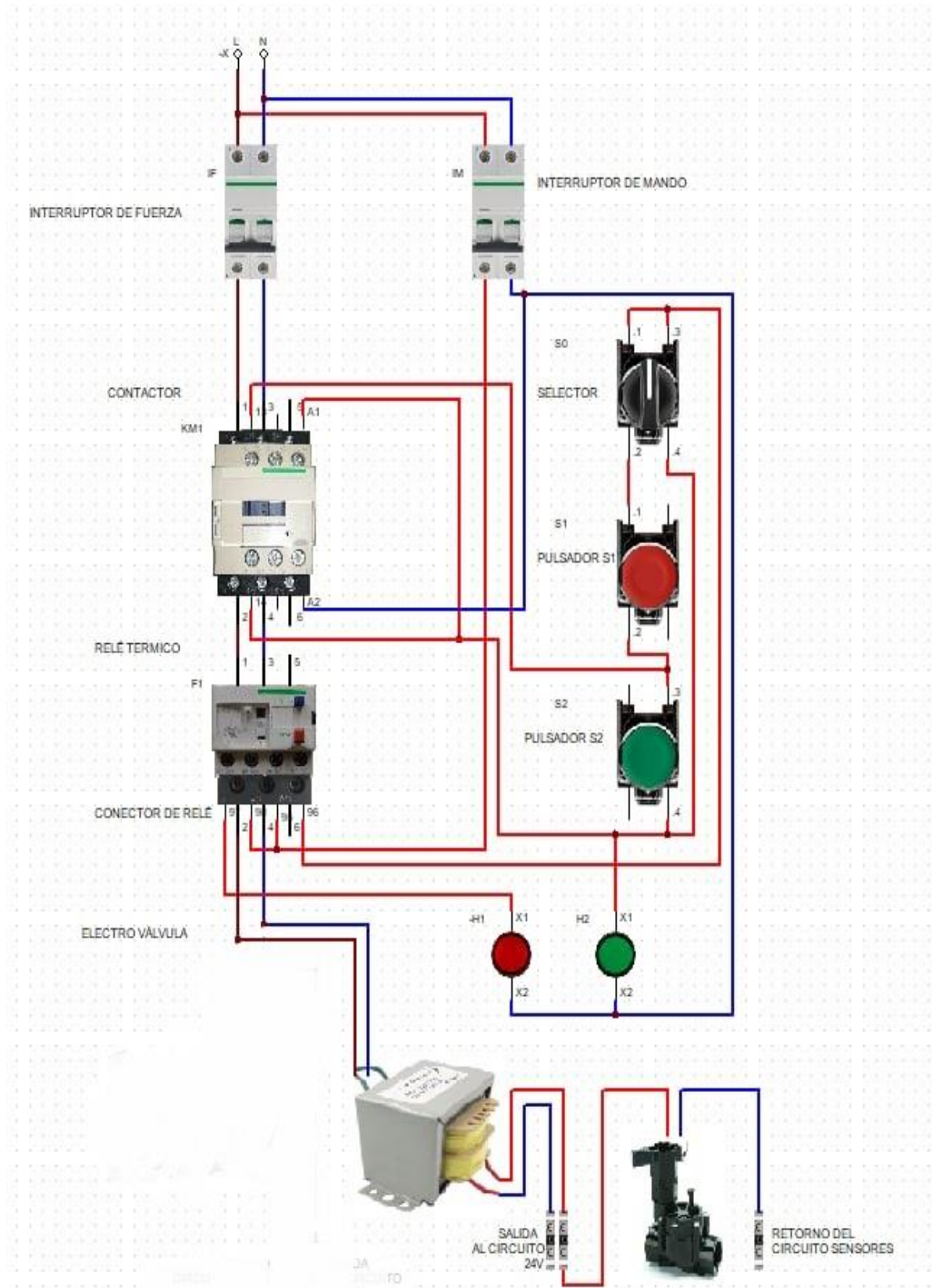
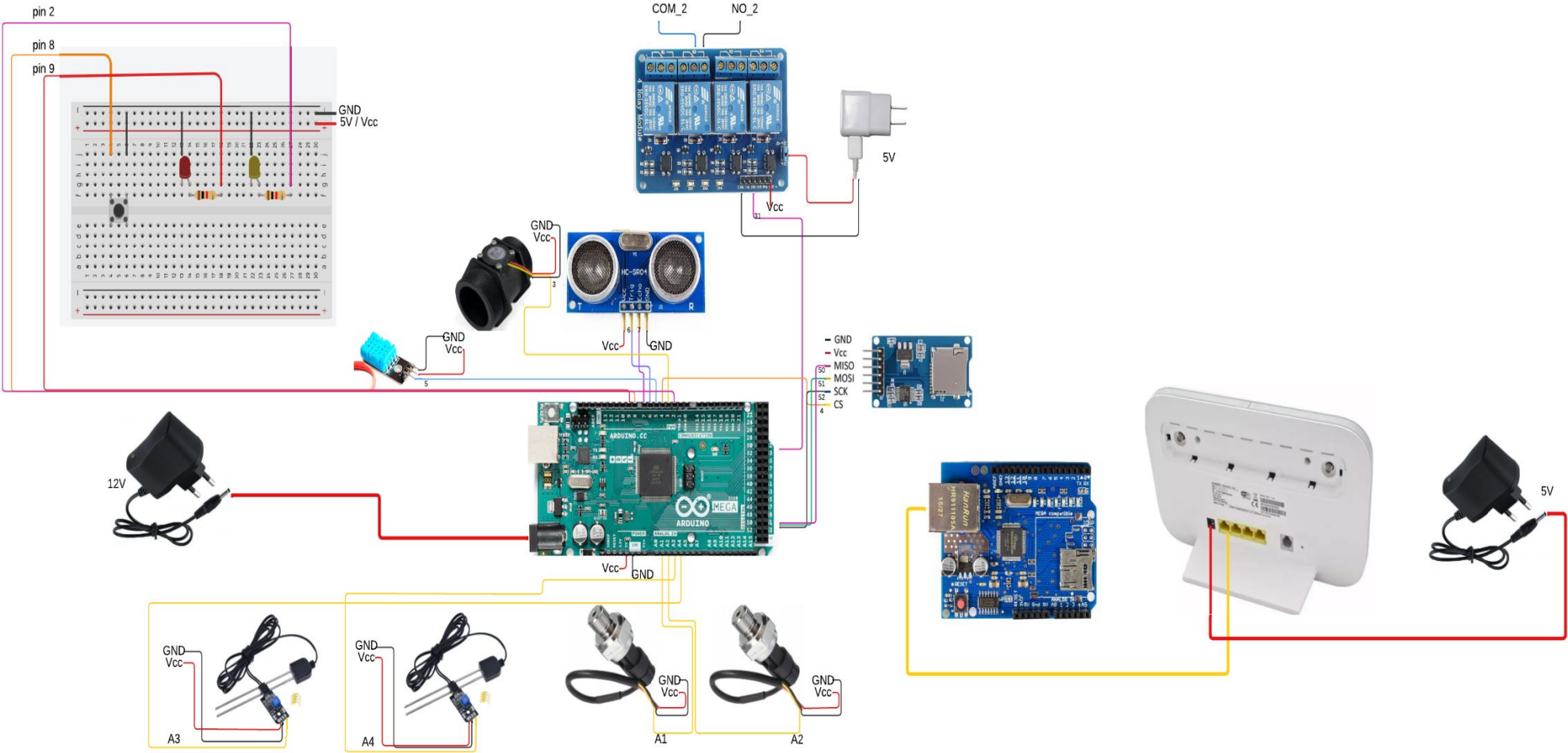


Figura 2. 13 Esquema de los sensores y actuador para la programación



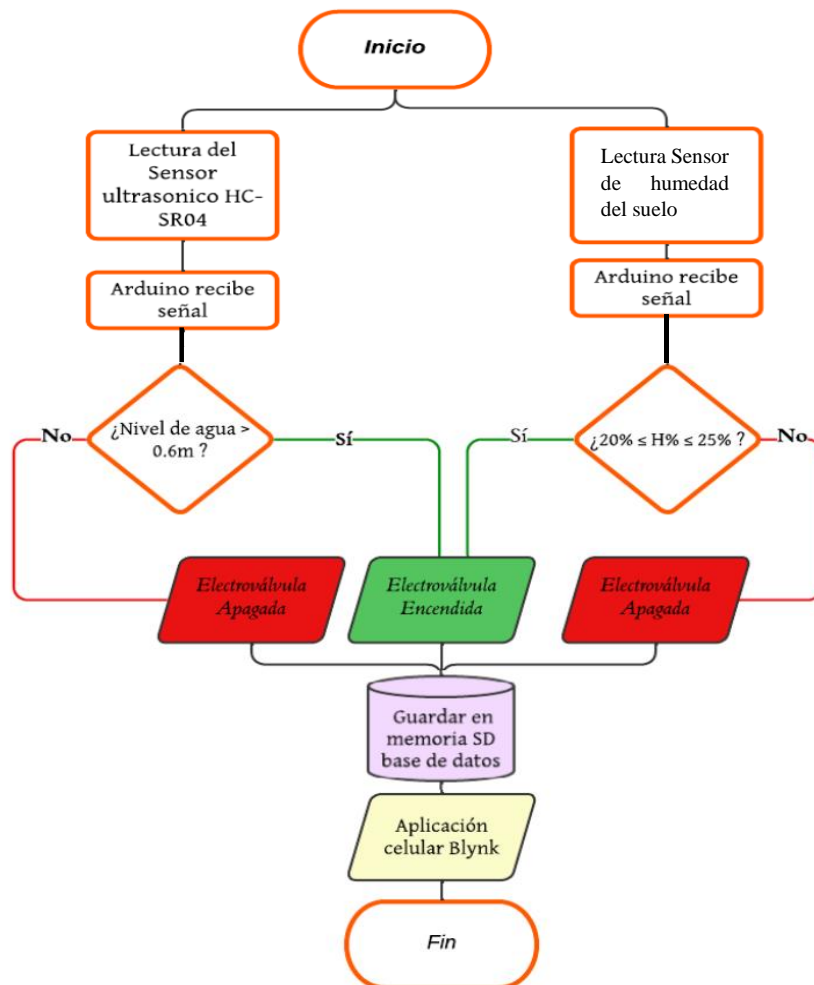
2.4.3. Programación de los sensores y actuador en el sketch de Arduino con sus respectivos códigos para su funcionamiento.

Antes de programar los sensores en Arduino, primero diseñé un flujograma para organizar y visualizar el funcionamiento del sistema. Esto me ayudó a planificar la lógica antes de escribir el código y posterior realizar la programación.

El proceso comienza con la lectura del sensor ultrasónico HC-SR04 y el sensor de humedad de suelo. Luego, el Arduino evalúa las mediciones: si el nivel de agua supera los 0.6 metros, activa la electroválvula; de lo contrario, la mantiene apagada. Después, verifica la humedad del suelo: si está entre el 20% y el 25%, la electroválvula se enciende; si no, se apaga.

Finalmente, los datos se almacenan en una memoria SD y se envían a la aplicación Blynk.

Figura 2. 14 Diagrama de flujo de trabajo de la electroválvula en función del sensor ultrasónico y humedad del suelo



a) Sensor de ultrasonido HC-SR04

En la **figura 2.15** se muestra el código de programación elaborado, donde primero se definieron los pines del sensor ultrasónico, (trigPin) para enviar señales y otro (echoPin) para recibirlas. Se estableció la velocidad del sonido en $0.000343 \text{ m}/\mu\text{s}$, necesaria para calcular la distancia, y se declaran variables para almacenar la distancia medida, el nivel de agua y el volumen del reservorio. El pozo tiene una altura total de 1.5 m, con un nivel crítico de seguridad a 0.6 m, y se define que el área base del reservorio total.

En la configuración inicial (setup), se dio inicio a la comunicación serie a 9600 bps para monitorear datos en tiempo real y se configuraron los pines del sensor (trigPin como salida y echoPin como entrada).

Para medir la distancia, se emplearon una función que activa el trigPin durante 10 microsegundos y utiliza pulseIn (echoPin, HIGH) para determinar el tiempo de respuesta del eco. La distancia se calcula con la fórmula: $\text{distancia} = (\text{duración} \times \text{velocidad del sonido}) / 2$, debido a que el sonido recorre el trayecto de ida y vuelta. Posteriormente, se ajustó la medición de la siguiente forma: $\text{nivelAgua} = \text{nivelTotalPozo} - (\text{distancia} - 0.36)$ y se aplicó una corrección adicional respecto a la calibración del sensor para una lectura correcta: $\text{nivelAgua} = (1.0209 \times \text{nivelAgua}) + 0.0072$. El volumen se determinó mediante una fórmula que tiene en cuenta la variación del área de la base según el nivel de agua. Finalmente, en el bucle principal (loop), la función de medición se ejecutaba continuamente, actualizándose los datos cada 30 segundos.

Una vez realizado el código de programación se procedió a subir en el Arduino Mega.

Figura 2. 15 Programación de código del sensor HC-SR04 en Arduino Mega

```
// ----- Definiciones -----
// Sensor ultrasónico
#define trigPin 6          // Pin Trigger
#define echoPin 7         // Pin Echo
const float velocidadSonido = 0.000343; // Velocidad del sonido en 343m/s = 0.000343 m/us
float distancia;
float nivelAgua;
float volumenReservorio;
const float nivelTotalPozo = 1.5; //m
const float nivelCriticoAgua = 0.6; //m
const float AreabaseMinima = 8.; //m2
const float AreaBaseMaxima = 24.; //m2
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");

  // Configuración sensor ultrasónico
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
// ----- Funciones -----
//Sensor ultrasónico
void medirDistancia() {
  long duracion;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duracion = pulseIn(echoPin, HIGH);
  distancia = (duracion * velocidadSonido) / (2);
  nivelAgua = nivelTotalPozo - (distancia-0.36); //total altura del pozo - lectura sensor-altura extra
  nivelAgua = (1.0209*nivelAgua)+0.0072;

  volumenReservorio = (nivelAgua / 3) *
    (AreabaseMinima +
     AreaBaseMaxima * pow((nivelAgua / nivelTotalPozo), 2) +
     sqrt(AreabaseMinima * AreaBaseMaxima * pow((nivelAgua / nivelTotalPozo), 2)));
  Serial.print("NivelAgua: ");
  Serial.print(nivelAgua);
  Serial.println(" m");
  Serial.print("VolumenReservorio: ");
  Serial.print(volumenReservorio);
  Serial.println(" m3");
}
// ----- Bucle principal -----
void loop() {
  medirDistancia(); // Leer y procesar distancia
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

b) Sensor de humedad del suelo

En la **figura 2.16** se muestra el código de programación elaborado del sensor de humedad del suelo, primero se definió los pines analógicos A3 y A4 para conectar los sensores. También se estableció los valores de referencia para interpretar las lecturas: 1000 indicaba suelo completamente seco y 100 representaba suelo totalmente húmedo. Además, se fijó un rango ideal de humedad entre el 20% y e 25% y se declaró variables para almacenar las mediciones.

Se asignó los pines de los sensores como entradas para que la placa pudiera recibir información de ellos. Después, se creó la función `controlarHumedad()`, que leía los valores de los sensores, los convertía en porcentajes utilizando `map()`, y se aplicó una corrección a través de una ecuación matemática proveniente de la calibración de los sensores de humedad para mejorar la precisión de las lecturas y el promedio entre ambos sensores, se llevaba un control más preciso. Finalmente, en el bucle principal, programé el sistema para que tomara una nueva lectura cada 30 segundos.

Figura 2. 16 Código de programación del sensor de humedad de suelo en Arduino Mega

```
// ----- Definiciones -----
// Sensor de humedad
#define pinHumedad1 A3 // Pin analógico para el sensor de humedad 1
#define pinHumedad2 A4 // Pin analógico para el sensor de humedad 2
const int valorAire1 = 1000;
const int valorAgua1 = 100;
const int valorAire2 = 1000;
const int valorAgua2 = 100;
const int limiteMinimoHumedad = 20;
const int limiteMaximoHumedad = 25;
int humedad1, humedad2; //PONER SI HUMEDAD <= 0% == 0%
int porcentajeHumedad1, porcentajeHumedad2, promedioPorcentajeHumedad;
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");

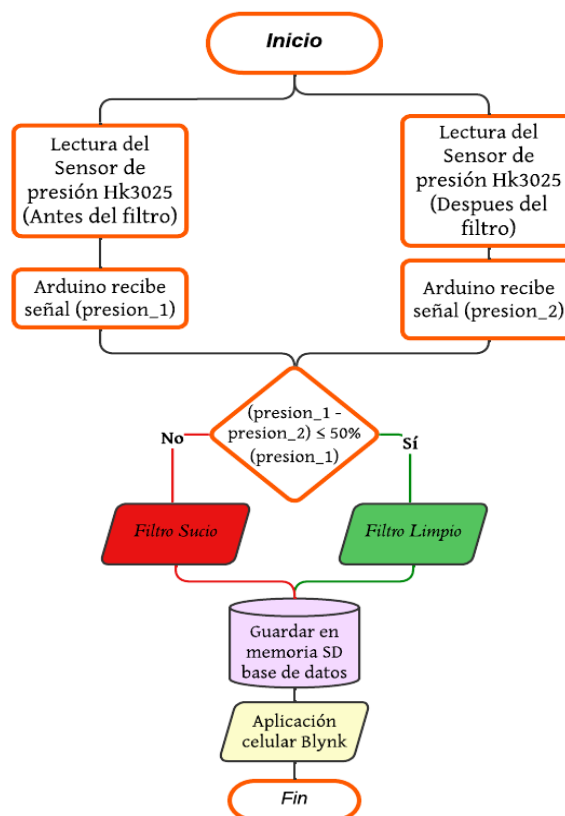
  // Configuración sensores de humedad
  pinMode(pinHumedad1, INPUT);
  pinMode(pinHumedad2, INPUT);
}
// ----- Funciones -----
// Sensor humedad suelo
void controlarHumedad() {
  humedad1 = analogRead(pinHumedad1);
  humedad2 = analogRead(pinHumedad2);
  porcentajeHumedad1 = map(humedad1, valorAire1, valorAgua1, 0, 100);
  porcentajeHumedad1 = (0.4072*porcentajeHumedad1)-0.3306;
  porcentajeHumedad2 = map(humedad2, valorAire2, valorAgua2, 0, 100);
  porcentajeHumedad2 = (0.4072*porcentajeHumedad2)-0.3306;
  promedioPorcentajeHumedad = (porcentajeHumedad1 + porcentajeHumedad2) / 2;
  Serial.print("Humedad 1: ");
  Serial.print(humedad1);
  Serial.print(", Humedad 2: ");
  Serial.print(humedad2);
  Serial.print(", Humedad 1: ");
  Serial.print(porcentajeHumedad1);
  Serial.print("%, Humedad 2: ");
  Serial.print(porcentajeHumedad2);
  Serial.print("%, Promedio: ");
  Serial.print(promedioPorcentajeHumedad);
  Serial.println("%");
}
// ----- Bucle principal -----
void loop() {
  controlarHumedad(); // Leer y procesar humedad
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

c) Sensor de presión Hk3025

El proceso inició con la lectura del sensor de presión HK3025 antes del filtro, cuyo valor fue registrado por el Arduino como "presion_1". Luego, se tomó una segunda lectura del sensor después del filtro, almacenando esta medición como "presion_2". A continuación, ambas señales fueron comparadas. Si la diferencia entre "presion_1" y "presion_2" no superaba el 50% del valor inicial, se determinaba que el filtro estaba en buen estado. En caso contrario, se consideraba que estaba sucio y requería mantenimiento.

Para llevar un registro, los datos fueron almacenados en una memoria SD. Además, se utilizó la aplicación móvil Blynk para monitorear los resultados en tiempo real y facilitar el monitoreo del sistema.

Figura 2. 17 Diagrama de flujo del sensor de presión antes y después del filtro.



En la **figura 2.18** se muestra el código de programación elaborado para los sensores de presión, se definió los pines presión1 A1 y presión2 A2. Además, se definió que la presión máxima sería de 5.0 MPa y se creó variables para almacenar las lecturas y el estado del filtro del sistema.

Después, se configuró los pines de los sensores como entradas (pinPresion1, INPUT); y (pinPresion2, INPUT), lo que permitió que la placa Arduino recibiera la información de presión de cada sensor.

Luego se programó la función medirPresion(), encargada de leer los valores de cada sensor utilizando analogRead(). Primero, convierte la lectura analógica en voltaje con la fórmula $(\text{valorAnalogico} * 5.0) / 1023.0$, y luego transforma ese voltaje en presión en KPa. Si esta diferencia no superaba el 50% de la presión de entrada, se consideró que el filtro estaba en buen estado (asigné el valor 1 a estadoFiltro). En caso contrario, marqué el filtro como sucio (valor 0). En el bucle principal (loop()), la función medirPresion() se ejecuta cada 30 segundos.

Figura 2. 18 Código de programación del sensor de presión Hk3025 en la placa Arduino

```
// ----- Definiciones -----
// Sensor de presión
#define pinPresion1 A1 // Pin analógico para el sensor de presión 1
#define pinPresion2 A2 // Pin analógico para el sensor de presión 2
const float voltajeMin = 0.5; // Voltaje a presión mínima (0 MPa)
const float voltajeMax = 4.5; // Voltaje a presión máxima (5.0 MPa)
const float presionMax = 5.0; // Presión máxima en MPa
float presion1=0.0, presion2=0.0;
int estadoFiltro;
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");

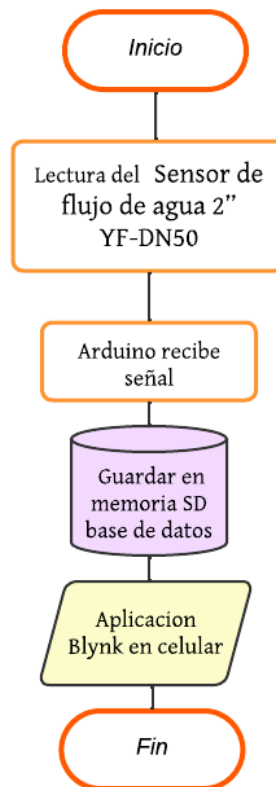
  // Configuración sensores de presión
  pinMode(pinPresion1, INPUT);
  pinMode(pinPresion2, INPUT);
}
// ----- Funciones -----
// Sensor de presión
void medirPresion() {
  // Sensor 1
  int valorAnalogico1 = analogRead(pinPresion1);
  float voltaje1 = (valorAnalogico1 * 5.0) / 1023.0; // Conversión de ADC a voltaje
  presion1 = ((voltaje1 - voltajeMin) / (voltajeMax - voltajeMin))*presionMax*1000.; //KPa
  presion1 = (0.0444*presion1)-0.5650;
  // Sensor 2
  int valorAnalogico2 = analogRead(pinPresion2);
  float voltaje2 = (valorAnalogico2 * 5.0) / 1023.0; // Conversión de ADC a voltaje
  voltaje2=voltaje2+0.065; //Calibración con voltaje1
  presion2 = ((voltaje2 - voltajeMin) / (voltajeMax - voltajeMin))*presionMax*1000.; //KPa
  presion2= (0.1211*presion2)-3.6842;
  if(presion1<presion2){
    presion2=presion1;
  }
  Serial.print("Presión voltaje 1: ");
  Serial.println(voltaje1);
  Serial.print("Presión voltaje 2: ");
  Serial.println(voltaje2);
  Serial.print("Presión Sensor 1: ");
  Serial.println(presion1);
  Serial.println(" KPa");
  Serial.print("Presión Sensor 2: ");
  Serial.println(presion2);
  Serial.println(" KPa");

  float diferenciaPresion = presion1-presion2;
  if(diferenciaPresion<=(presion1/2)){ //si la diferencia es mayor al 50% de la entrada
    estadoFiltro=1; //FILTRO OK
  }else{
    estadoFiltro=0; //FILTRO SUCIO
  }
}
// ----- Bucle principal -----
void loop() {
  medirPresion(); // Leer y procesar presión de agua3
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

d) Sensor de flujo de agua 2" YF-DN50

El proceso inició con la lectura del sensor de flujo de agua de 2" YF-DN50. Luego, el Arduino recibió la señal correspondiente a esta lectura. Después, se guardaron los datos obtenidos en una memoria SD para mantener una base de datos. Finalmente, se utilizó la aplicación Blynk en el celular para monitorear y visualizar los datos registrados

Figura 2. 19 Diagrama de flujo del sensor de flujo de agua 2" YF-DN50



En la **figura 2.20** se muestra el código de programación elaborado que permitió medir el flujo de agua conectado al pin 3. Para ello, primero se definieron las variables necesarias: el pin del sensor, un contador de pulsos, un factor de conversión de 0.2 pulsos por litro y variables para calcular la frecuencia, el caudal en litros por minuto y el tiempo de la última medición.

Se configuró el pin del sensor como entrada con resistencia interna activada y se asignó una interrupción para contar los pulsos cada vez que se detectara un flanco de subida.

El código también incluyó una función principal encargada de medir el flujo de agua. Esta verificaba si había pasado al menos 1 segundo desde la última medición y, si era así, desactivaba las interrupciones temporalmente para leer con precisión el número de pulsos. Con estos datos

se calculaba la frecuencia en pulsos por segundo y, a partir de ella, se determinaba el caudal en litros por minuto. Para obtener una medición precisa, se aplicaba la fórmula de calibración del sensor (**$1.0335 \times \text{caudal} - 16.488$**).

Por último, en el bucle principal, el sistema repetía la medición continuamente, dejando una pausa de 30 segundos entre cada medición.

Figura 2. 20 Código de programación del sensor de flujo de agua en la placa Arduino

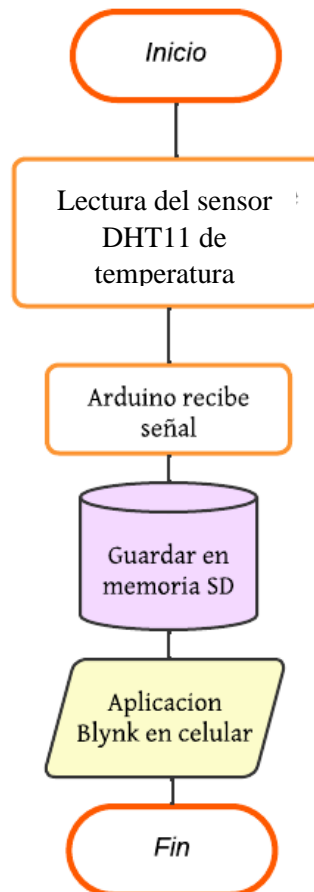
```
// ----- Definiciones -----
// Sensor de flujo de agua
#define pinFlujoAgua 3
volatile int conteoPulsos = 0; // Número de pulsos
float frecuencia, flujoLitrosMinuto;
const float factorConversion = 0.2; // Pulsos por litro (según sensor)
unsigned long tiempoAnterior = 0;
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");

  // Configuración sensor de flujo de agua
  pinMode(pinFlujoAgua, INPUT_PULLUP);
  // Se configura una interrupción para registrar cada pulso (contarPulsos) generado por el sensor (pin 3)
  attachInterrupt(digitalPinToInterrupt(pinFlujoAgua), contarPulsos, RISING);
}
// ----- Funciones -----
//Sensor flujo de agua
void medirFlujoAgua() {
  unsigned long tiempoActual = millis();
  // Calcular flujo cada segundo
  if (tiempoActual - tiempoAnterior >= 1000) {
    tiempoAnterior = tiempoActual;
    // Calcular la frecuencia
    noInterrupts(); // Detén las interrupciones temporalmente para leer el conteo
    frecuencia = conteoPulsos;
    conteoPulsos = 0; // Reiniciar conteo
    interrupts(); // Habilita las interrupciones nuevamente
    // Convertir frecuencia a caudal (L/min) - datasheet
    flujoLitrosMinuto = (frecuencia / factorConversion);
    flujoLitrosMinuto = (1.0335*flujoLitrosMinuto)-16.488;
    if(flujoLitrosMinuto<0){
      flujoLitrosMinuto=0.0;
    }
    // Mostrar datos
    Serial.print("Frecuencia: ");
    Serial.print(frecuencia);
    Serial.print(" Hz\tFlujo: ");
    Serial.print(flujoLitrosMinuto);
    Serial.println(" L/min ");
  }
}
// Función de interrupción para contar pulsos
void contarPulsos() {
  conteoPulsos++;
}
// ----- Bucle principal -----
void loop() {
  medirFlujoAgua(); // Leer y procesar flujo de agua
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

e) **Sensor DHT11 de temperatura**

El proceso inició con la lectura del sensor DHT11 de temperatura. Luego, el Arduino recibió la señal correspondiente a esta lectura. Después, se guardaron los datos obtenidos en una memoria SD para mantener una base de datos. Finalmente, se utilizó la aplicación Blynk en el celular para monitorear y visualizar los datos registrados

Figura 2. 21 Diagrama de flujo del sensor de temperatura



En la **figura 2.22** se muestra el código que fue diseñado para medir la humedad ambiental y la temperatura utilizando un sensor DHT11 conectado al pin 5 del Arduino. Primero, se incluyó la librería necesaria y se definieron las variables para almacenar los valores de temperatura y humedad. Luego, en la configuración inicial, se estableció la comunicación serial a 9600 bps.

En el desarrollo del programa, se creó una función para leer los datos del sensor, donde se obtuvo la humedad y la temperatura en (°C). A la temperatura se le aplicó una corrección mediante una ecuación lineal debido a la calibración del sensor para que realice una lectura correcta. En caso de que la lectura fallara, se asignó un valor de -1.0 para indicar un error.

Figura 2. 22 Código de programación del sensor de temperatura en la placa Arduino

```
// ----- Definiciones -----
//Sensor de humedad
#include <DHT.h>
#define DHTPIN 5 // Pin al que está conectado el sensor
#define DHTTYPE DHT11 // Tipo de sensor (DHT11)
DHT dht(DHTPIN, DHTTYPE);
float humedadAmbiente, temperatura;
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");
  // Inicializar el sensor DHT11
  delay(1000);
  dht.begin();
}
// ----- Funciones -----
// Sensor DHT11
void leerSensorDHT() {
  // Leer humedad ambiental
  humedadAmbiente = dht.readHumidity();
  // Leer temperatura en Celsius
  temperatura = dht.readTemperature();
  temperatura = (0.5503*temperatura)+7.8711;
  // Verificar si hubo error en la lectura
  if (isnan(humedadAmbiente)) {
    humedadAmbiente=-1.0;
  }
  if (isnan(temperatura)) {
    temperatura=-1.0;
  }
  // Mostrar valores en el monitor serial
  Serial.print("Humedad Ambiental: ");
  Serial.print(humedadAmbiente);
  Serial.print(" %\t");
  Serial.print("Temperatura Ambiental: ");
  Serial.print(temperatura);
  Serial.println(" °C");
}
// ----- Bucle principal -----
void loop() {
  leerSensorDHT(); // Leer y procesar DHT11
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

f) Módulo relé de 4 canales

El código de programación elaborado para el módulo relé permitió tener el control de la electroválvula que se instaló en la subunidad de riego; Cabe mencionar que para su funcionamiento se tuvieron que hacer uso del terminal In2 del módulo relé llegando a conectarse en el pin 31 de la placa Arduino para electroválvula.

En la **figura 2.23** se muestra que en la sección de configuración (setup), se estableció la comunicación serial a una velocidad de 9600 bps y se configuró el pin del relé como salida.

Las funciones principales se encargaban de controlar los relés según el nivel de agua y los valores de humedad. Había una lógica de seguridad para asegurarse de que la válvula no estuviera abierta por más de 8 minutos consecutivos, en cuyo caso se consideraba una anomalía y se cerraba la válvula. Además, si la válvula estaba abierta por más de 2 minutos seguidos, se cerraba y se esperaba 10 minutos antes de intentar abrirla nuevamente.

Figura 2. 23 Código de programación del módulo relé de 4 canales en la placa Arduino

```
// ----- Definiciones -----
// Relés
#define RELE_ON 0
#define RELE_OFF 1
#define rele_2 31
int estadoValvula = 0;
// ----- Configuración -----
void setup() {
  // Comunicación serial
  Serial.begin(9600);
  Serial.println("Sistema iniciado.");

  // Configuración relés
  pinMode(rele_2, OUTPUT);
  digitalWrite(rele_2, HIGH); // Iniciar relé apagado (relé activado por LOW)
}
// ----- Funciones -----
// Funciones de control de relés
int tiempoValvulaAbierta = 0;
int formaActivacion = 0; //0=automatico, 1>manual
void controlarReles() {
  // Verificar nivel crítico de agua, aplica a ambos modos
  if (nivelAgua <= nivelCriticoAgua) {
    cerrarValvula();
    return;
  }
}
//Verificar que no esté abierto por 2 min seguidos
if(estadoValvula==2){
  cerrarValvula();
  delay(600000); //600000 = 10 min
  return;
}
//Modo automático
if(formaActivacion==0){
  if (promedioPorcentajeHumedad >= limiteMinimoHumedad && promedioPorcentajeHumedad <= limiteMaximoHumedad) {
    abrirValvula();
  } else {
    cerrarValvula();
  }
}
//Modo manual
else if (formaActivacion == 1){
  abrirValvula();
}
}
void abrirValvula() {
  digitalWrite(rele_2, RELE_ON); //activarRele
  estadoValvula = 1;
  tiempoValvulaAbierta++;
  // Verificar si la válvula lleva demasiado tiempo abierta
  if (tiempoValvulaAbierta >= 16) { //16 = 8 minutos -> 480'000segundos/30'000 (segundos loop).
    estadoValvula = 2; //Valvula estuvo abierta por mucho tiempo (anomalía)
  }
}
void cerrarValvula() {
  digitalWrite(rele_2, RELE_OFF); //desactivarRele
  estadoValvula = 0;
  tiempoValvulaAbierta = 0;
}
// ----- Bucle principal -----
void loop() {
  controlarReles(); // Controlar los relés
  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

g) Módulo Ethernet Shield W5100

Una vez se culminaron las programaciones del actuador y los sensores, se procedió con la del Módulo de Ethernet Shield W5100, encargado de enviar información a la nube.

En la **figura 2.24** se muestra cómo se estableció la conexión con el servidor mediante la dirección IP de la red, lo que permite enviar toda la información a la nube de internet. Además, la información se actualizaba cada 20 minutos.

Figura 2. 24 Código de programación módulo Ethernet Shield W5100

```
void loop() {
  actualizarFechaHora();
  contadorEnvioBlynk++;
  if(contadorEnvioBlynk >=40){ //40, cada 20 minutos enviar a la app Blynk
    Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
    Blynk.virtualWrite(V1, nivelAgua);
    Blynk.virtualWrite(V11, volumenReservorio);
    Blynk.virtualWrite(V3, estadoValvula);
    Blynk.virtualWrite(V4, flujoLitrosMinuto);
    Blynk.virtualWrite(V5, presion1);
    Blynk.virtualWrite(V6, presion2);
    Blynk.virtualWrite(V12, estadoFiltro);
    Blynk.virtualWrite(V7, humedadAmbiente);
    Blynk.virtualWrite(V8, temperatura);
    Blynk.virtualWrite(V9, 1);
    String fechaHora = fecha + " " + hora;
    Blynk.virtualWrite(V10, fechaHora);
    contadorEnvioBlynk=0;
  }
  Blynk.run();

  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1 minuto)
  Serial.println("-----");
}
```

2.4.4. Implementación de sensores y actuadores en el sistema de riego por goteo del invernadero y realizar las pruebas exhaustivas.

La Internet de las Cosas (IoT) es clave en la implementación de sistemas de riego inteligentes, ya que permite la integración y gestión remota de sensores y actuadores a través de la red. La IoT se basa en conectar dispositivos electrónicos habilitados para ser gestionados a distancia mediante protocolos de comunicación que, idealmente, siguen estándares abiertos.

En el sistema de riego, los sensores miden variables como la humedad del suelo, la temperatura ambiente, o el nivel de agua en el reservorio. Estos dispositivos recopilan datos que son enviados a través de Internet a un sistema centralizado o a la nube, donde son analizados para tomar decisiones. Por otro lado, los actuadores, como electroválvulas, reciben instrucciones para activar o desactivar el flujo de agua en función de las necesidades del cultivo determinadas por los datos de los sensores.

Para dicho fin, primero fue necesario realizar el mantenimiento y la limpieza de algunos componentes del sistema de riego por goteo, así como llevar a cabo la siembra de la hortaliza. Posteriormente, se implementaron los sensores y actuadores de manera exitosa, sin que se presentaran inconvenientes durante el proceso.

a) Limpieza y mantenimiento del reservorio

Se llevó a cabo una limpieza exhaustiva del reservorio debido a que el canal de entrada de agua se encontraba en construcción, lo que provocó la acumulación de suciedad y sedimentos en el interior del sistema. Este mantenimiento fue crucial para garantizar que el agua almacenada estuviera libre de impurezas que pudieran obstruir el funcionamiento del sistema de riego por goteo, especialmente en los emisores y conductos.

Figura 2. 25 Limpieza del reservorio



b) Colocación de la caseta de control

Se procedió a instalar la caseta de control, un espacio destinado para albergar todo el sistema de automatización. Dentro de la caseta, se colocó una caja hermética, que tiene como propósito proteger los componentes electrónicos del sistema de posibles daños causados por la humedad o la lluvia. Esta caja asegura que todos los circuitos, conexiones y dispositivos estén resguardados en un entorno seguro y seco, garantizando el buen funcionamiento del sistema a pesar de las condiciones climáticas adversas. Además, la caseta fue ubicada en un lugar estratégico, de fácil acceso para su mantenimiento y monitoreo, pero a la vez protegida de factores externos que pudieran interferir con su operatividad.

Figura 2. 26 Instalación de la caseta de control



c) Cajas de protección

Para implementar los sensores y actuadores, se construyeron cajas de ladrillo para protegerlos de la lluvia y otros factores climáticos que podrían afectar su funcionamiento, garantizando así su correcto desempeño.

Figura 2. 27 Caja de protección en el cabezal de riego



Figura 2. 28 Caja de protección para la electroválvula

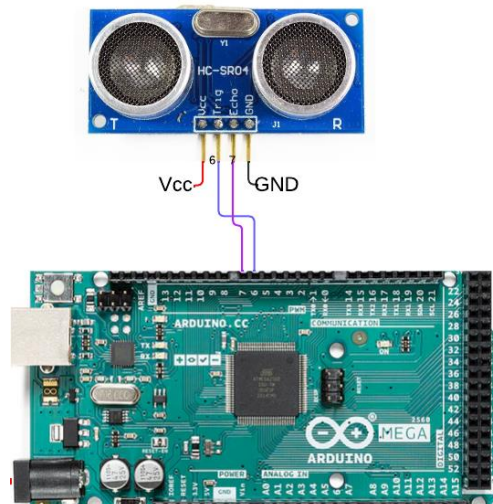


d) Implementación de sensores y actuadores

-Sensor de Ultrasonido HC-SR04

El sensor de ultrasonido HC-SR04 fue implementado con el fin de medir el nivel del agua del reservorio, mediante el uso de ondas ultrasónicas enviamos una onda ultrasónica mediante el disparo del pin Trigger (6) la cual viaja a través del aire y al encontrar un objeto este disparo fue rebotado y recepcionado por el sensor a través del pin Echo (7).

Figura 2. 29 Comunicación entre el sensor ultrasonido HC-SR04 y placa Arduino



El sensor se instaló en el reservorio utilizando un listón de madera como soporte, lo que garantizó su estabilidad y evitó que cayera al agua, previniendo posibles daños. Asimismo, se diseñó y colocó una caja de protección que lo resguardó de las inclemencias del clima, como la lluvia, y de la intensa radiación solar propia del distrito de Tambo.

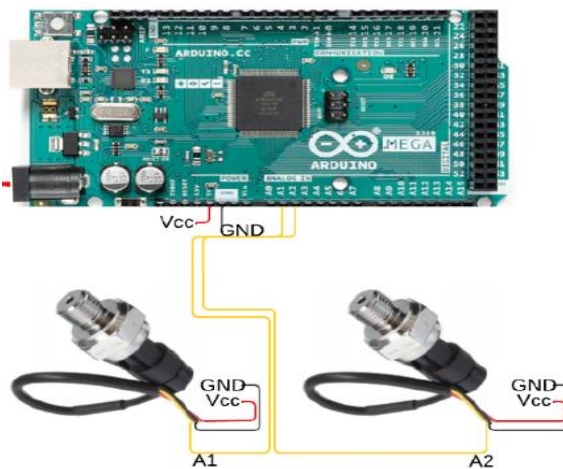
Figura 2. 30 Instalación del sensor ultrasónico HC-SR04



-Sensor de presión Hk3025

El sensor fue instalado en el cabezal de riego con el objetivo de medir la presión tanto antes como después del filtro. Mediante el pin A1 y pin A2, permitió enviar los valores de presión necesarios para tomar decisiones informadas.

Figura 2. 31 Comunicación entre el Sensor de presión Hk3025 y placa Arduino



Al detectarse una pérdida de presión, se procedió a realizar la limpieza del filtro de anilla, lo que garantizó que las presiones en el sistema de riego se mantuvieran dentro de los niveles adecuados, para una mejor distribución de agua en la subunidad de riego.

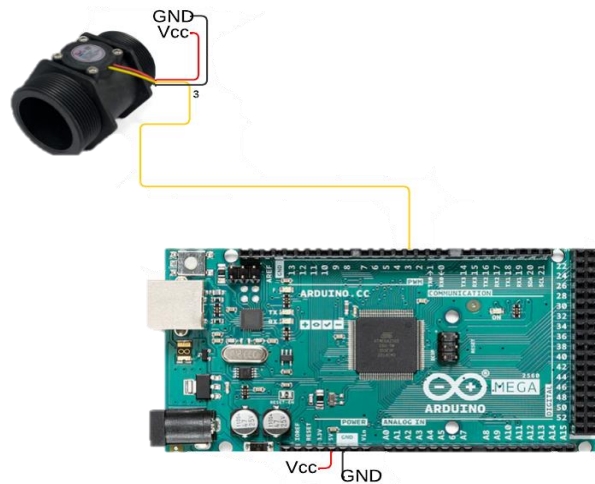
Figura 2. 32 Instalación del sensor de presión Hk3025



-Sensor de flujo de agua YF-DN50

El sensor de flujo de agua YF-DN50 se instaló en el cabezal de riego para medir el caudal de salida en la tubería. Este sensor cuenta con tres cables de distintos colores: rojo (alimentación) y negro (tierra) las cuales fueron conectados al protoboard y el amarillo (salida) al pin 3 para toma de datos.

Figura 2. 33 Comunicación entre Sensor de flujo de agua YF-DN50 y placa Arduino



Este sensor de flujo fue implementado en el cabezal del sistema de riego para monitorear y controlar el caudal de agua que circula por la tubería y se suministra al cultivo. Su instalación permite obtener mediciones precisas en tiempo real.

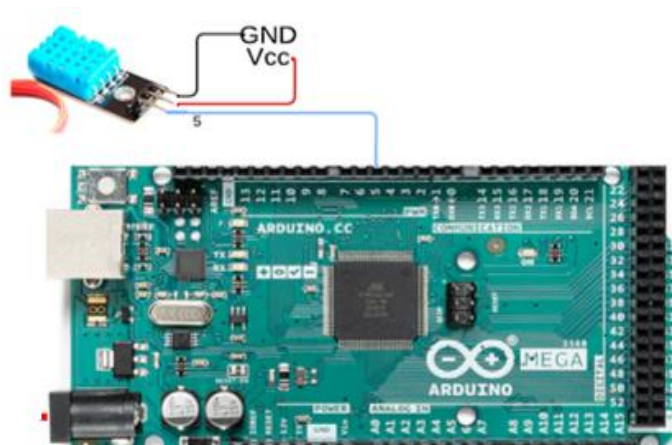
Figura 2. 34 Instalación del sensor de flujo YF-DN50



-Sensor DHT11 de temperatura

El sensor fue conectado al microcontrolador Arduino, con el objetivo de monitorear la temperatura en el invernadero. Además, cuenta con los pines correspondientes, DATA (señal) fue conectado al pin 5, VCC (alimentación) y GND (tierra).

Figura 2. 35 Comunicación del Sensor DHT11 de temperatura y la placa Arduino



El sensor fue instalado a una altura adecuada cerca del cultivo, garantizando que las mediciones de temperatura reflejaran con precisión las condiciones reales dentro del invernadero.

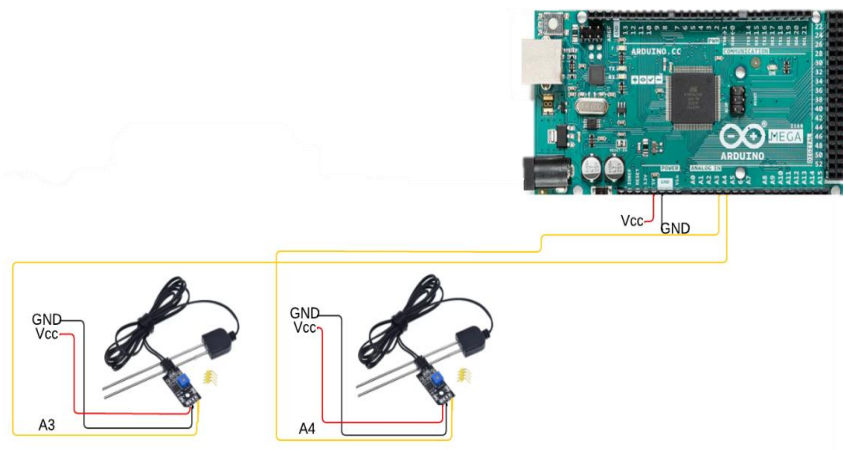
Figura 2. 36 Instalación del Sensor DHT11 de temperatura



-Módulo higrómetro sensor de humedad del suelo

El sensor de humedad del suelo cuenta con varios pines fundamentales para su funcionamiento: VCC (suministra la energía) , GND (tierra) ; y el pin A3 y A4, encargados de transmitir la señal. Este sensor mide la humedad del suelo a través de su resistencia eléctrica, lo que permite determinar el nivel de humedad presente en la tierra de manera precisa.

Figura 2. 37 Comunicación entre el sensor de humedad y placa Arduino



Se instaló el sensor de humedad de suelo en la subunidad de riego, ubicada a 60 metros del cabezal y a 10 metros del microcontrolador Arduino. Este sensor se implementó con el objetivo de asegurar una entrega adecuada de agua a la hortaliza zanahoria.

Figura 2. 38 Instalación del sensor de humedad



-Modulo Relé de 4 canales

Se instaló un módulo relé en la caseta de control para manejar altos voltajes con una señal pequeña desde una placa Arduino Mega 2560 R3. Además, se utilizó un transformador para reducir de 220V AC a 24V AC y alimentar la electroválvula. El cargador se conectó a los pines GND y VCC, mientras que el terminal IN2 se asignó al pin 31 para la señal de control.

Figura 2. 39 Comunicación entre el Módulo Relé de 4 canales y la placa Arduino

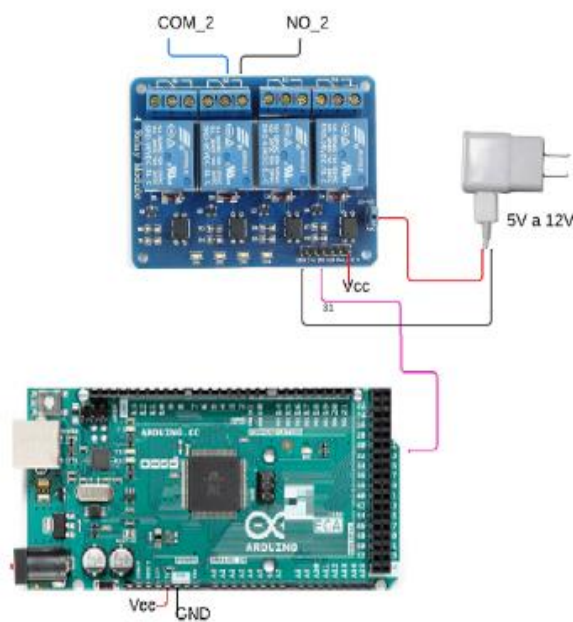
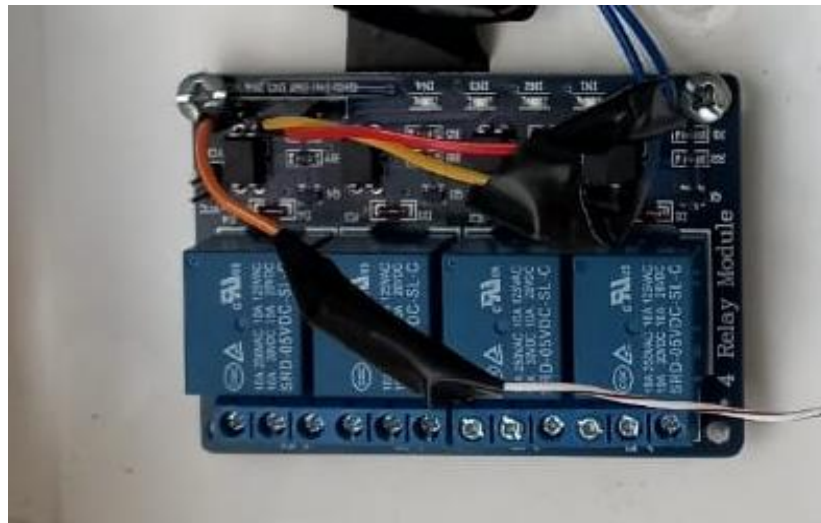


Figura 2. 40 Instalación del Módulo Relé de 4 canales



-El módulo Ethernet Shield W5100

Se instaló en la caseta de control, con el fin de facilitar la comunicación en tiempo real con todos los sensores y actuadores ubicados. Para lograr monitorear los distintos sensores y actuador, fue necesario instalar Internet y un router. Es importante destacar que el módulo Ethernet Shield se montó en la placa Arduino Mega 2560 Rev3, y luego se conectó a la red, lo que permitió establecer comunicación con la aplicación Blynk y así monitorear las acciones en tiempo real.

Figura 2. 41 Comunicación del Módulo de Ethernet y la placa Arduino

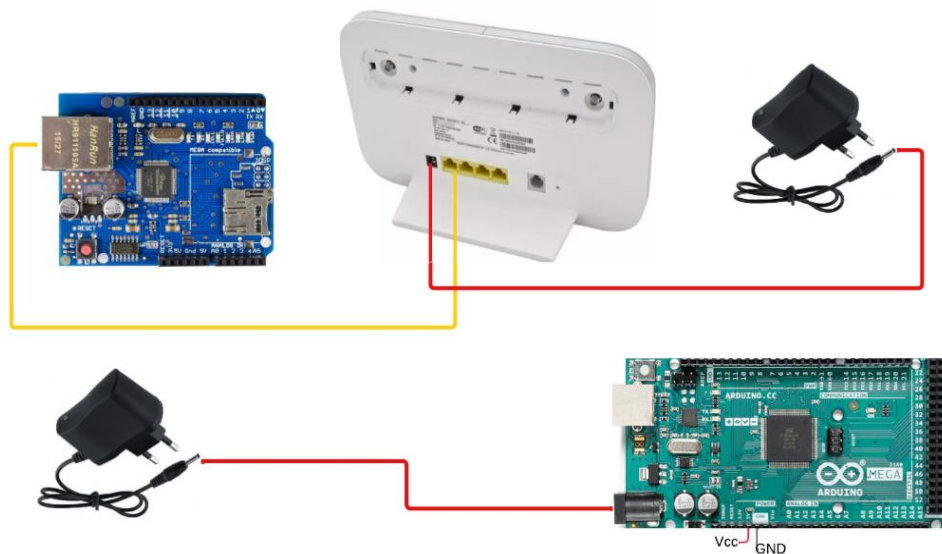
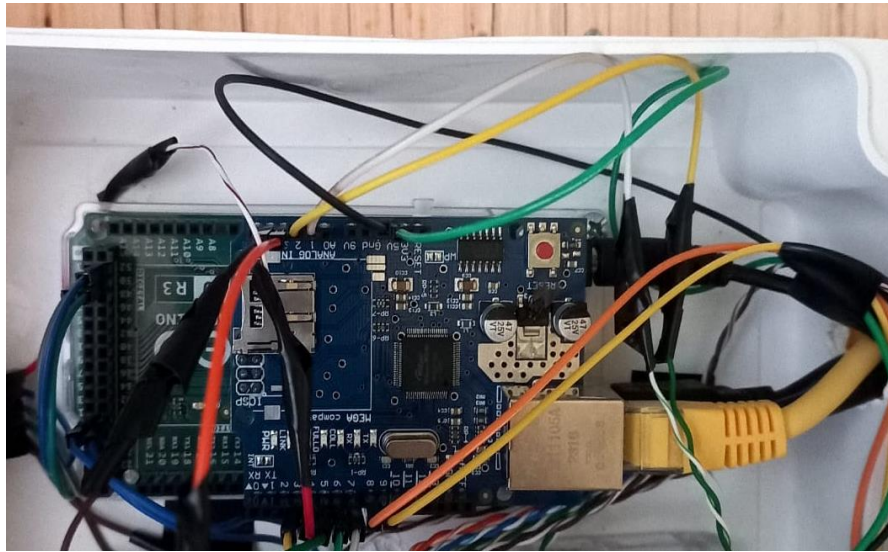


Figura 2. 42 Instalación del Módulo Ethernet Shield



-Arduino Mega 2560 Rev3

La placa Arduino mega 2560 Rev3 se instaló en la caseta a 8 m del invernadero y así poder controlar todos los sensores implementados; Así mismo posee el microcontrolador ATmega 2560 y tiene una conexión USB, conector ICSP y un botón reset. Se usó esta tarjeta ya que es compatible con la mayoría de los Shields.

Figura 2. 43 Instalación de la placa Arduino Mega 2560 R3



-MicroSD Card Adapter

El MicroSD Card Adapter fue instalado junto al Arduino con el propósito de almacenar los datos registrados, los cuales se actualizan cada 30 segundos. Esta implementación permite guardar de forma continua y organizada las mediciones de todos los sensores y actuador, asegurando que la información esté disponible para su posterior análisis.

Figura 2. 44 Comunicación entre el MicroSD Car Adapter y la placa Arduino

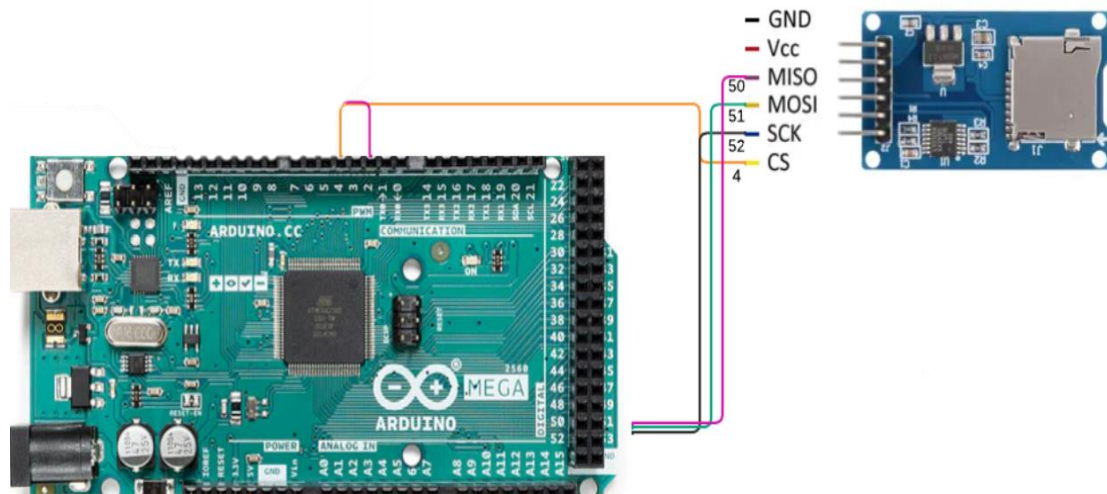


Figura 2. 45 Instalación del MicroSD Card Adapter



-Conexión eléctrica

Se llevó a cabo la instalación eléctrica dentro de la caja hermética para alimentar y controlar la electroválvula, y fuera de ella se realizaron las conexiones necesarias para los cargadores que suministran energía al sistema de automatización.

Figura 2. 46 Esquema del sistema eléctrico en la caja hermética

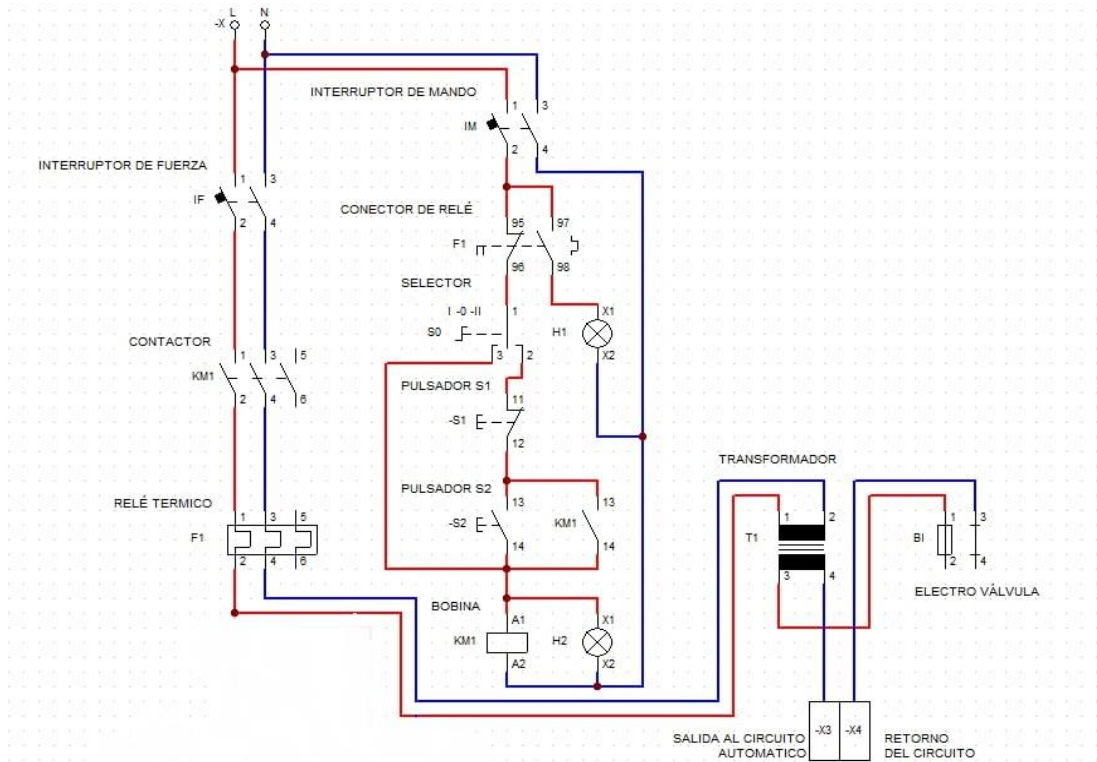
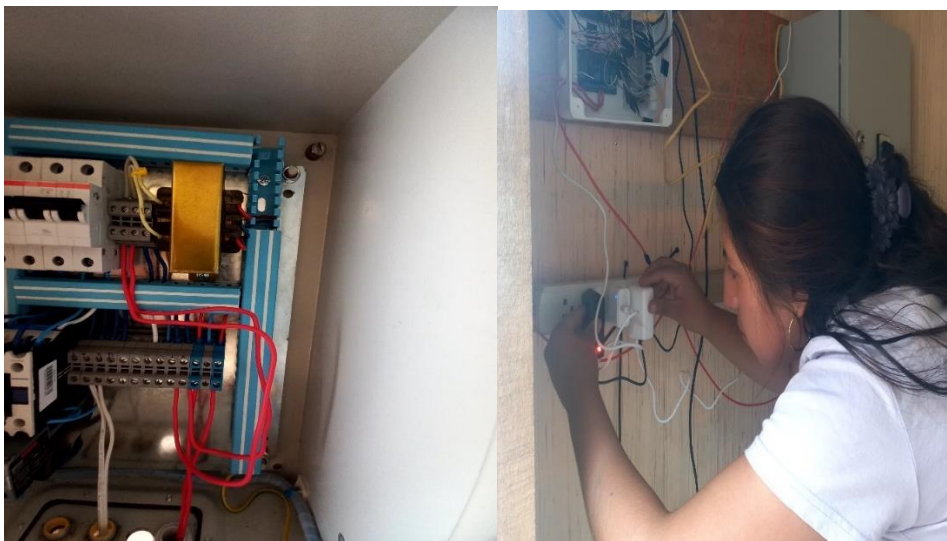


Figura 2. 47 Instalación de la alimentación eléctrica



- Electroválvula de 24V AC

Se instaló una electroválvula de 24V AC en la subunidad de riego para controlar la entrega de agua a los cultivos. Esta electroválvula fue de 2" y de funcionamiento normalmente cerrado, cuenta con una bobina solenoide que regula el paso y cierre del agua. Para su operación, se utilizó un transformador que convierte 220V AC a 24V AC. Además, se conectó al pin 31 de la placa Arduino Mega 2560 R3 para el mando y monitoreo del sistema.

Figura 2. 48 Instalación de la electroválvula



-Sembrío de la Hortaliza zanahoria

El 15 de diciembre se llevó a cabo la siembra de zanahorias en el invernadero situado en la localidad de Pampa Hermosa- Lorensayoq. Para ello, se preparó previamente el terreno dentro del invernadero, asegurando condiciones ideales para el desarrollo del cultivo, como un nivel adecuado de humedad, la fertilización del suelo y la limpieza de malezas. Asimismo, se utilizaron semillas de alta calidad para garantizar un crecimiento uniforme y saludable de las plantas. Este proyecto tuvo como propósito fomentar un cultivo sostenible en un entorno controlado, optimizando tanto la producción como la calidad de la cosecha en beneficio del propietario.

Figura 2. 49 Sembrado de la Zanahoria



2.4.5. Calibración y ajustes de los sensores para una correcta lectura.

a) Sensor de ultrasonido HC-SR04

Para calibrar el sensor ultrasónico HC-SR04, se realizaron lecturas en un punto de referencia y se compararon con mediciones reales obtenidas mediante un flexómetro. Este proceso se llevó a cabo antes de la instalación en el reservorio.

Figura 2. 50 Toma de datos para ajustar las lecturas del sensor ultrasónico HC-SR04



Funcionamiento: Se basó en poder calcular de manera exacta el tiempo de envío y recepción que hace el sensor a través de un pulso, cabe mencionar que la velocidad del sonido es 343 m/s.

$$Velocidad = \frac{Distancia}{Tiempo} \dots \dots \dots Ecuación (2)$$

$$343 \frac{m}{s} * 100 \frac{cm}{m} * \frac{1}{10000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s} \dots \dots \dots Ecuación (3)$$

$$Velocidad = \frac{2 * Distancia}{tiempo} \dots \dots \dots Ecuación (4)$$

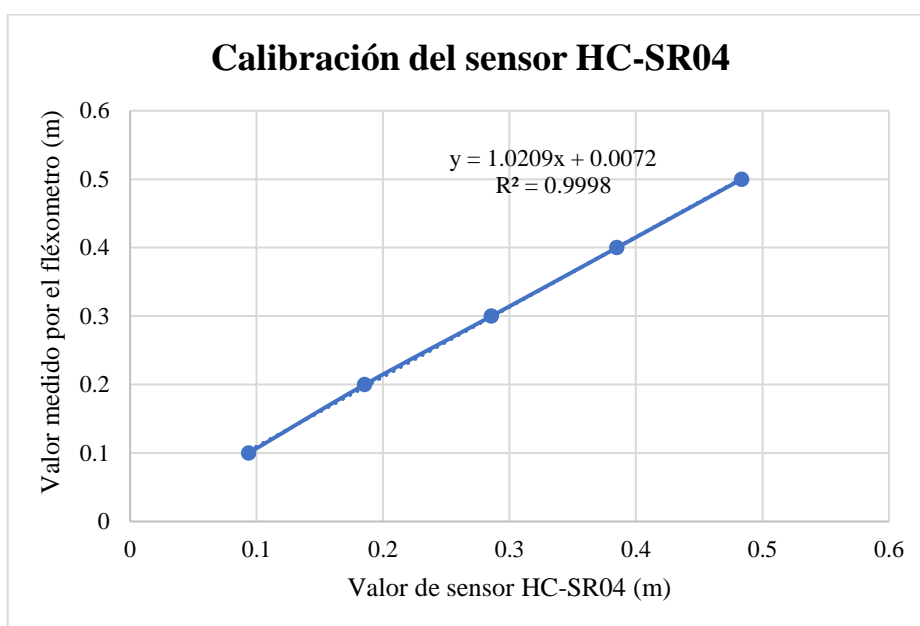
De la ecuación 3 se obtiene que el sonido tarda 29.2 μs en recorrer 1 cm. Con este principio, es posible calcular el tiempo que el sensor HC-SR04 emplea en emitir y recibir el pulso ultrasónico, determinando así la distancia mediante la siguiente fórmula:

$$Distancia (cm) = \frac{Tiempo (\mu s)}{29.2 * 2} \dots \dots \dots Ecuación (5)$$

Dividimos entre 2 porque el tiempo medido corresponde al recorrido de ida y vuelta del pulso ultrasónico. Sin embargo, solo nos interesa la distancia en un solo trayecto.

La calibración se realizó mediante una regresión simple, que relaciona la distancia de la lectura del sensor ultrasónico con la lectura del flexómetro, obteniendo un coeficiente de determinación alta el cual nos hace saber que la calibración fue correcta.

Figura 2. 51 Calibración del sensor de ultrasonido HC-SR04



b) Sensor de presión HK- 3025 5MPA

Para calibrar el sensor de presión, se realizaron lecturas antes y después del filtro utilizando un manómetro digital.

Figura 2. 52 Toma de datos para ajustar las lecturas del sensor presión HK-3025



La calibración se llevó a cabo mediante una regresión simple, que relaciona las lecturas del sensor (señales analógicas convertidas a kPa) con las mediciones del manómetro digital (kPa). El alto coeficiente de determinación obtenido confirma la precisión y fiabilidad de la calibración.

Figura 2. 53 Calibración del sensor de presión Hk-3025 antes del filtro

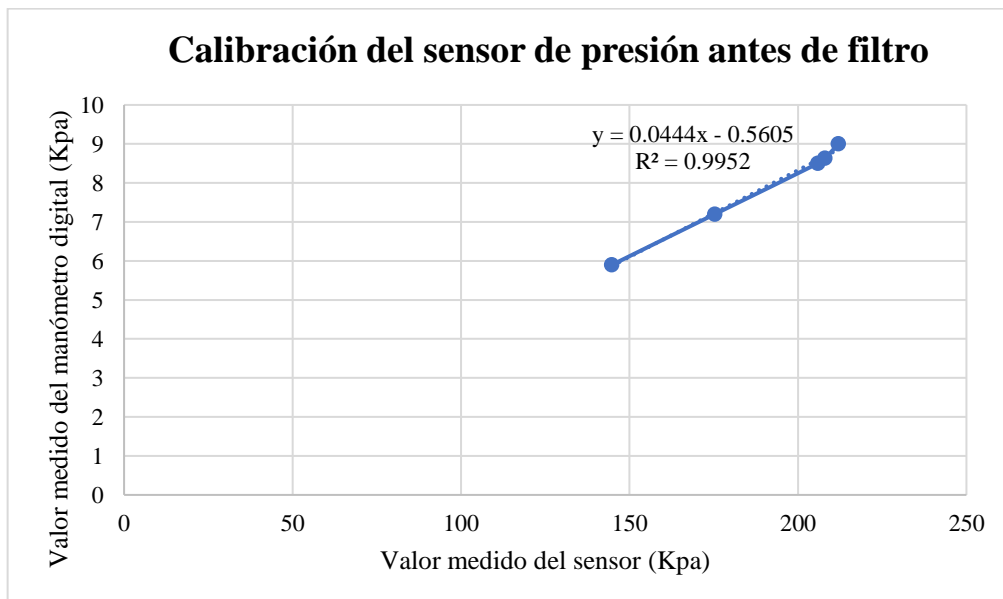
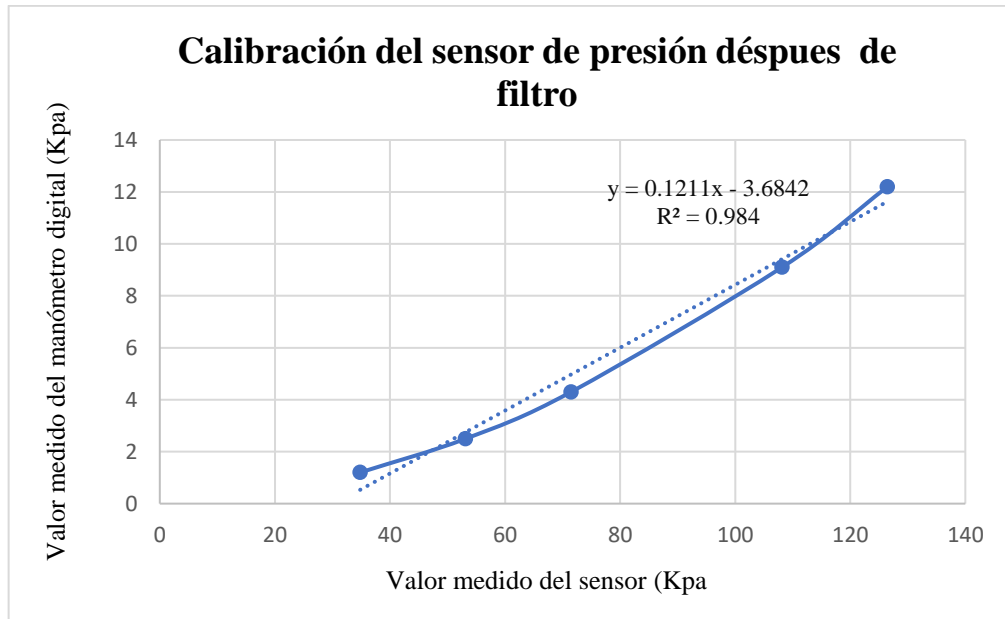


Figura 2. 54 Calibración del sensor de presión Hk-3025 después del filtro



c) Sensor de flujo de agua 2" YF-DN50

Para la calibración del flujo de agua fue necesario el uso de un caudalímetro.

Figura 2. 55 Toma de datos para ajustar las lecturas del sensor de flujo de agua 2" YF-DN50



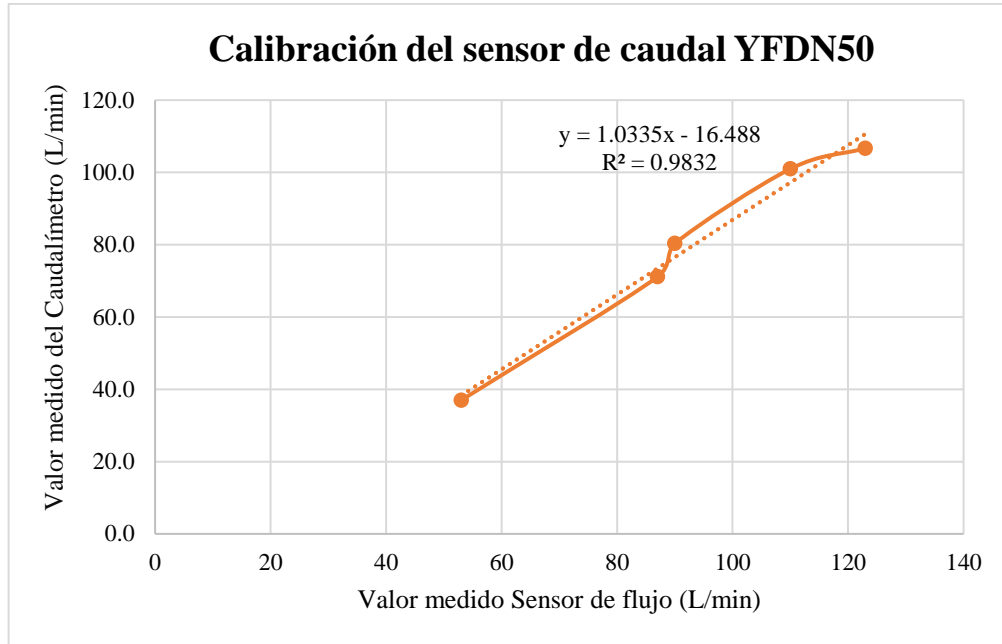
Funcionamiento:

El cálculo para convertir la lectura de frecuencia o el número de pulsos del sensor de flujo en caudal se realiza utilizando la siguiente fórmula:

$$Q_{(l/min)} = \frac{\text{Pulsaciones del sensor (Hz)}}{0.2} \dots \dots \dots \text{Ecuación (6)}$$

Una vez obtenido los valores del sensor y del caudalímetro se pudo hacer la calibración, mediante una regresión simple, obteniendo una correlación y un coeficiente de determinación alta, el cual nos hace saber que la calibración fue correcta.

Figura 2. 56 Calibración del sensor de flujo de agua 2” YF-DN50



d) Sensor DHT11 de temperatura

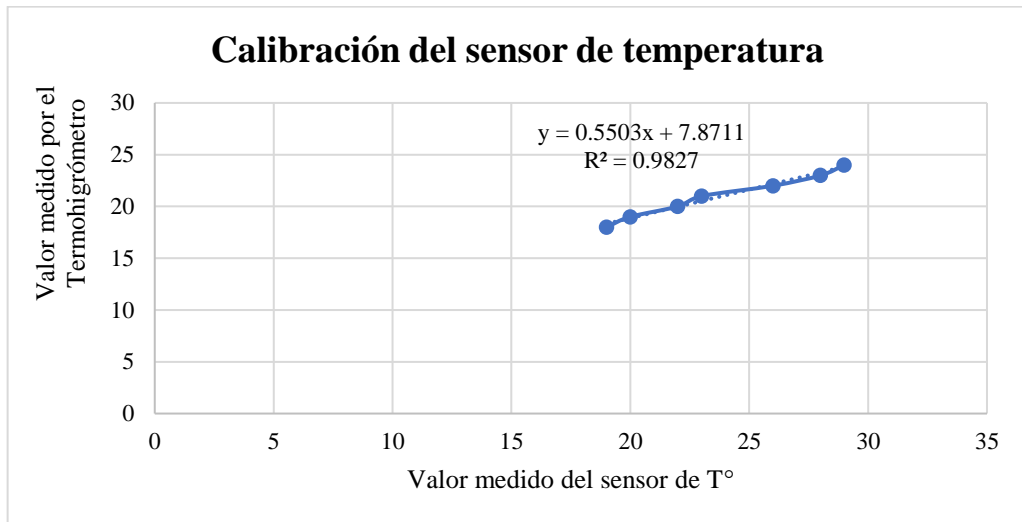
Para la calibración del sensor DHT11 de temperatura se realizó haciendo uso de con un termohigrómetro.

Figura 2. 57 Toma de datos para ajustar las lecturas del sensor de temperatura



Mediante una regresión simple, se compararon los valores, obteniendo una correlación y un coeficiente de determinación alta, el cual nos hace saber que la calibración fue correcta.

Figura 2. 58 Calibración del sensor DHT11 de temperatura



e) Módulo higrómetro sensor de humedad del suelo

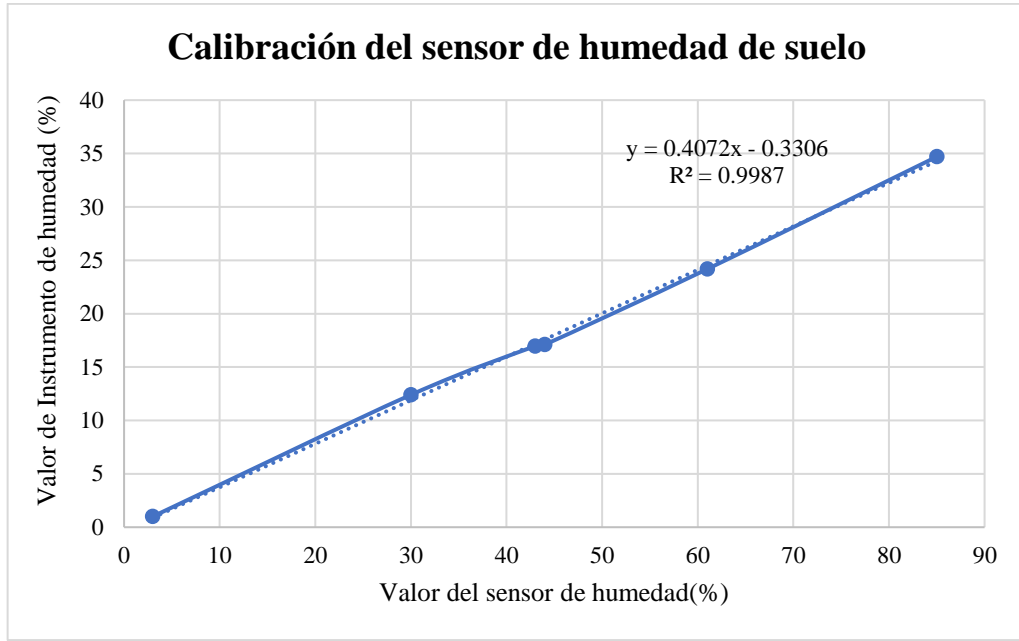
Para la calibración del sensor de humedad se tuvo que instalar a 10cm de profundidad del suelo; este sensor nos da las lecturas ya convertidas en porcentaje, la cual varía según la cantidad de humedad que existe en el suelo, cabe mencionar que también se sacó muestras del suelo; estas muestras fueron llevados al laboratorio para conocer la humedad gravimétrica de suelo, Además, se empleó un medidor de humedad del suelo para realizar mediciones *in situ*.

Figura 2. 59 Toma de datos para ajustar las lecturas del sensor de humedad



Mediante la ecuación lineal se relacionó las mediciones con los resultados del laboratorio de suelos. Como resultado, obtuvimos una alta correlación y un coeficiente de determinación elevado, lo que confirmó la precisión y confiabilidad de la calibración.

Figura 2. 60 Calibración del sensor de humedad



2.4.6. Creación de un aplicativo en la plataforma Blynk para tener lecturas en tiempo real.

Se eligió la plataforma Blynk para la monitorización en tiempo real de los sensores y el actuador debido a su simplicidad, flexibilidad y amplia compatibilidad con diversos dispositivos. Su facilidad de uso la hace ideal tanto para pequeños proyectos de IoT como para sistemas a gran escala. Además, permite la conexión a la nube sin costos adicionales, garantizando una interacción segura y en tiempo real.

Para comenzar a usar Blynk, simplemente hay que descargar la aplicación desde el Play Store y registrarse con un correo electrónico. Al hacerlo, se genera un token de autenticación, el cual es clave para vincular dispositivos como el Arduino Mega 2560 Rev3 al sistema. Una vez autenticado, se configuró un nuevo proyecto en la aplicación, llamado "**Riego Automatizado**", aprovechando la conexión a internet para su correcto funcionamiento.

Figura 2. 61 Conexión de los sensores y actuador para el monitoreo en Blynk

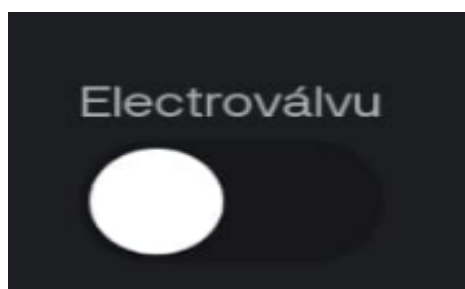


- **Creación de widgets para los sensores y actuador para el monitoreo en tiempo real del sistema de riego**

Para el monitoreo en tiempo real del sistema de riego, se diseñaron widgets personalizados en la aplicación Blynk, lo cual es clave para tomar decisiones rápidas y efectivas, la actualización de la aplicación se da cada 20 minutos.

- a) Para el control de la electroválvula se tuvo que crear otro botón con el nombre de Electroválvula este se conectó en el pin virtual número V 13 del aplicativo Blynk, se configuró en el modo Switch para que pueda trabajar como un interruptor para su encendido y apagado correspondiente.

Figura 2. 62 Widgets para el encendido y apagado de la electroválvula



- b) Para controlar los sensores de presión ubicados antes y después del filtro de anillas, se crearon dos medidores. Estos medidores se conectaron a los pines virtuales V5 y V6 en la aplicación Blynk.

Figura 2. 63 Widgets para el control de presión antes y después del filtro de anilla



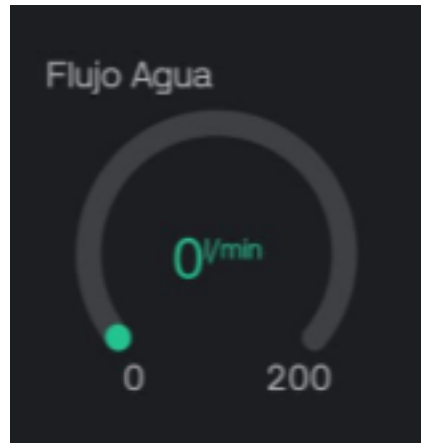
- c) Para controlar el nivel del reservorio, se utilizó un sensor HC-SR04 que se conectó al pin virtual número V1, mientras que el volumen pin virtual número V11 en la aplicación Blynk. Es importante destacar que las mediciones se realizaron en metros y el volumen en metros cúbicos.

Figura 2. 64 Widgets para el control del nivel y volumen del reservorio



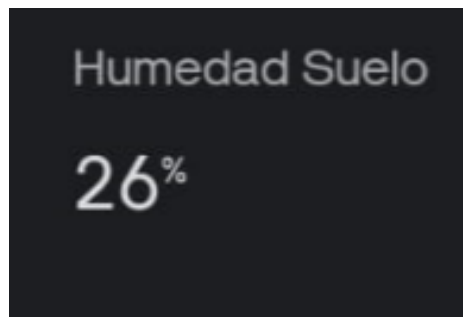
- d) Para controlar el sensor de caudal, se creó un medidor que se conectó al pin virtual número V4 en la aplicación Blynk, las mediciones del sensor de caudal se realizaron en litros por segundo (l/min).

Figura 2. 65 Widgets para el control del caudal de agua



- e) Para controlar el sensor de humedad del suelo, se conectó al pin virtual número V0 en la aplicación Blynk, para mostrar las mediciones del sensor en porcentaje (%).

Figura 2. 66 Widgets para el control Widgets para el control de la humedad del suelo



- f) Para controlar el sensor de temperatura, se conectó al pin virtual número V8 en la aplicación Blynk, para mostrar las mediciones de temperatura en Celsius (°C).

Figura 2. 67 Widgets para el control Widgets para el control de la Temperatura



2.4.7. Cosecha de la Hortaliza zanahoria

El 04 de abril se llevó a cabo la cosecha de las zanahorias, obteniendo una producción de alta calidad y buen rendimiento.

Figura 2. 68 Cosecha de zanahorias



CAPÍTULO III

RESULTADOS

3.1. Resultado de la Implementación y monitoreo con microcontrolador Arduino para la automatización de riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024.

Este trabajo permitió implementar y monitorear con éxito un sistema de riego automatizado con Arduino en una parcela de hortalizas ubicada en Tambo, Ayacucho, durante el año 2024.

Gracias al uso del microcontrolador Arduino, se integraron sensores de humedad del suelo, temperatura, presión y caudal para monitoreo el riego de manera eficiente. Estos se conectaron a la plataforma Blynk, permitiendo el monitoreo remoto. El sistema también incorporó un control automatizado de la electroválvula, regulando su apertura o cierre según las lecturas de los sensores de humedad y nivel de agua. Esto garantizó un riego preciso, evitando el exceso o la escasez de agua, optimizando su uso y favoreciendo el crecimiento saludable de la zanahoria.

Asimismo, se ha comprobado en el campo que un sistema de riego automatizado favorece un mejor desarrollo y rendimiento de los cultivos en comparación con los sistemas tradicionales. Por ello, se presenta como una alternativa clave para fortalecer la agricultura en la región, contribuyendo también a la preservación de los recursos naturales.

3.1.1. Resultado sobre el Análisis de los componentes de la Automatización para el riego por goteo.

Los componentes utilizados en la automatización del riego por goteo permitieron comprender cómo cada elemento desempeña un papel esencial en el correcto funcionamiento del sistema.

- a) Se emplearon sensores de alta precisión para garantizar un control óptimo de las variables críticas del riego.
 - ✓ El sensor de ultrasonido HC-SR04 aseguró un nivel adecuado de almacenamiento de agua en el reservorio.
 - ✓ El sensor de presión Hk3025 permitió mantener limpio el anillo de filtro, asegurando una distribución uniforme del agua.

- ✓ El sensor de flujo de agua 2" YF-DN50 proporcionó información sobre el caudal de agua circulante en el sistema durante la activación del riego.
 - ✓ Adicionalmente, el sensor DHT11 y los sensores de humedad del suelo monitorearon la temperatura y la humedad en el invernadero, facilitando la toma de decisiones para un riego eficiente.
- b) El módulo relé de 4 canales 5V fue fundamental para gestionar el encendido y apagado de dispositivos de mayor potencia, como la electroválvula de 24 VAC, encargada de regular el flujo de agua en el sistema.
- c) La placa Arduino Mega 2560 R3, junto con el módulo Ethernet Shield W5100, proporcionó la plataforma de control y comunicación, permitiendo la conexión a Internet para un monitoreo remoto a través de un smartphone y acceso mediante la red UTP.
- d) El manómetro digital, el caudalímetro y el medidor de humedad del suelo/laboratorio ofrecieron datos en tiempo real, permitiendo al sistema ajustar sus parámetros para una programación óptima.

3.1.2. Resultado sobre el desarrollo de un sistema efectivo de comunicación entre el sensor y el microcontrolador Arduino.

Se logró con éxito el desarrollo de un sistema de comunicación eficaz entre los sensores y el microcontrolador Arduino, lo que permitió una interacción precisa y eficiente entre ambos componentes. Gracias a esta integración, fue posible obtener los siguientes datos recolectados:

- En cuanto al sensor de humedad, cabe mencionar que se logró hacer el monitoreo de la cantidad de humedad en el suelo, la cual tuvo valor mínimo de 24 % y un valor máximo de 28 %; cuyos valores se encontraron cercanos de la capacidad de campo, lo que favoreció al buen desarrollo y rendimiento del cultivo de la zanahoria.

Tabla 3. 1 Resultado de la comunicación efectiva del sensor de Humedad

Fecha	HORA	Porcentaje de humedad (%)	Capacidad de campo	Punto de marchitez	Estado de la electroválvula
16/12/2024	16:06:07	27	27.45	14.09	Apagado
16/12/2024	23:59:58	27	27.45	14.09	Apagado
17/12/2024	00:00:09	27	27.45	14.09	Apagado
17/12/2024	23:59:57	27	27.45	14.09	Apagado
18/12/2024	00:00:08	27	27.45	14.09	Apagado
18/12/2024	23:59:52	27	27.45	14.09	Apagado
19/12/2024	00:00:03	27	27.45	14.09	Apagado
19/12/2024	23:59:57	26	27.45	14.09	Apagado
20/12/2024	00:00:07	26	27.45	14.09	Apagado
20/12/2024	23:59:54	26	27.45	14.09	Apagado
21/12/2024	00:00:05	26	27.45	14.09	Apagado
21/12/2024	10:21:26	24	27.45	14.09	Encendido
21/12/2024	10:21:56	24	27.45	14.09	Encendido
21/12/2024	10:22:26	24	27.45	14.09	Encendido
21/12/2024	10:22:56	25	27.45	14.09	Encendido
21/12/2024	10:23:27	25	27.45	14.09	Encendido
21/12/2024	10:24:01	25	27.45	14.09	Encendido
21/12/2024	10:24:31	25	27.45	14.09	Encendido
21/12/2024	10:25:02	25	27.45	14.09	Encendido
21/12/2024	23:59:59	27	27.45	14.09	Apagado
22/12/2024	00:00:09	27	27.45	14.09	Apagado
22/12/2024	23:59:49	27	27.45	14.09	Apagado
23/12/2024	00:00:00	27	27.45	14.09	Apagado
23/12/2024	23:59:49	27	27.45	14.09	Apagado
24/12/2024	00:00:00	27	27.45	14.09	Apagado
24/12/2024	23:59:53	27	27.45	14.09	Apagado
25/12/2024	00:00:04	26	27.45	14.09	Apagado
25/12/2024	23:59:58	26	27.45	14.09	Apagado
26/12/2024	00:00:08	26	27.45	14.09	Apagado
26/12/2024	23:59:53	26	27.45	14.09	Apagado
27/12/2024	00:00:03	26	27.45	14.09	Apagado
27/12/2024	17:23:30	25	27.45	14.09	Encendido
27/12/2024	17:24:01	24	27.45	14.09	Encendido
27/12/2024	17:24:29	24	27.45	14.09	Encendido
27/12/2024	17:25:00	25	27.45	14.09	Encendido
27/12/2024	17:30:27	25	27.45	14.09	Encendido
27/12/2024	17:30:58	25	27.45	14.09	Encendido
27/12/2024	23:59:56	26	27.45	14.09	Apagado
28/12/2024	00:00:06	27	27.45	14.09	Apagado

28/12/2024	23:59:57	27	27.45	14.09	Apagado
29/12/2024	00:00:07	27	27.45	14.09	Apagado
29/12/2024	23:59:51	27	27.45	14.09	Apagado
30/12/2024	00:00:01	27	27.45	14.09	Apagado
30/12/2024	23:59:57	27	27.45	14.09	Apagado
31/12/2024	00:00:08	26	27.45	14.09	Apagado
31/12/2024	23:59:51	26	27.45	14.09	Apagado
01/01/2025	00:00:01	26	27.45	14.09	Apagado
01/01/2025	23:59:56	26	27.45	14.09	Apagado
01/01/2025	14:38:03	24	27.45	14.09	Encendido
01/01/2025	14:38:34	24	27.45	14.09	Encendido
01/01/2025	14:39:05	24	27.45	14.09	Encendido
01/01/2025	14:39:37	25	27.45	14.09	Encendido
01/01/2025	14:40:08	25	27.45	14.09	Encendido
01/01/2025	14:40:39	25	27.45	14.09	Encendido
01/01/2025	14:41:10	25	27.45	14.09	Encendido
01/01/2025	14:41:41	25	27.45	14.09	Encendido
02/01/2025	00:00:06	28	27.45	14.09	Apagado
02/01/2025	23:59:58	27	27.45	14.09	Apagado
03/01/2025	00:00:08	27	27.45	14.09	Apagado
03/01/2025	23:59:51	27	27.45	14.09	Apagado
04/01/2025	00:00:01	27	27.45	14.09	Apagado
04/01/2025	23:59:55	27	27.45	14.09	Apagado
05/01/2025	00:00:05	27	27.45	14.09	Apagado
05/01/2025	23:59:59	27	27.45	14.09	Apagado
06/01/2025	00:00:10	27	27.45	14.09	Apagado
06/01/2025	23:59:51	27	27.45	14.09	Apagado
07/01/2025	00:00:01	28	27.45	14.09	Apagado
07/01/2025	23:59:49	27	27.45	14.09	Apagado
08/01/2025	00:00:20	27	27.45	14.09	Apagado
08/01/2025	23:59:42	26	27.45	14.09	Apagado
09/01/2025	00:00:14	26	27.45	14.09	Apagado
09/01/2025	09:06:21	25	27.45	14.09	Encendido
09/01/2025	09:06:52	24	27.45	14.09	Encendido
09/01/2025	09:07:24	24	27.45	14.09	Encendido
09/01/2025	09:07:56	25	27.45	14.09	Encendido
09/01/2025	09:08:27	25	27.45	14.09	Encendido
09/01/2025	09:08:59	25	27.45	14.09	Encendido
09/01/2025	09:09:31	25	27.45	14.09	Encendido
09/01/2025	23:59:42	27	27.45	14.09	Apagado
10/01/2025	00:00:14	27	27.45	14.09	Apagado
10/01/2025	23:59:49	27	27.45	14.09	Apagado
11/01/2025	00:00:20	27	27.45	14.09	Apagado
11/01/2025	23:59:42	28	27.45	14.09	Apagado

12/01/2025	00:00:14	27	27.45	14.09	Apagado
12/01/2025	23:59:42	28	27.45	14.09	Apagado
13/01/2025	00:00:14	27	27.45	14.09	Apagado
13/01/2025	06:07:06	24	27.45	14.09	Encendido
13/01/2025	06:07:37	25	27.45	14.09	Encendido
13/01/2025	06:08:09	25	27.45	14.09	Encendido
13/01/2025	06:08:40	24	27.45	14.09	Encendido
13/01/2025	06:09:11	25	27.45	14.09	Encendido
13/01/2025	06:09:43	25	27.45	14.09	Encendido
13/01/2025	06:10:14	25	27.45	14.09	Encendido
13/01/2025	23:59:42	28	27.45	14.09	Apagado
14/01/2025	00:00:14	27	27.45	14.09	Apagado
14/01/2025	23:59:42	27	27.45	14.09	Apagado
15/01/2025	00:00:14	27	27.45	14.09	Apagado
15/01/2025	23:59:42	27	27.45	14.09	Apagado
16/01/2025	00:00:14	27	27.45	14.09	Apagado
16/01/2025	23:59:42	27	27.45	14.09	Apagado
17/01/2025	00:00:14	27	27.45	14.09	Apagado
17/01/2025	15:17:02	24	27.45	14.09	Encendido
17/01/2025	15:17:34	24	27.45	14.09	Encendido
17/01/2025	15:18:05	25	27.45	14.09	Encendido
17/01/2025	15:18:36	25	27.45	14.09	Encendido
17/01/2025	15:19:08	25	27.45	14.09	Encendido
17/01/2025	15:19:39	25	27.45	14.09	Encendido
17/01/2025	23:59:56	26	27.45	14.09	Apagado
18/01/2025	00:00:27	27	27.45	14.09	Apagado
18/01/2025	19:47:28	27	27.45	14.09	Apagado
19/01/2025	08:38:13	26	27.45	14.09	Apagado
19/01/2025	23:59:31	28	27.45	14.09	Apagado
20/01/2025	00:00:02	27	27.45	14.09	Apagado
20/01/2025	23:59:42	27	27.45	14.09	Apagado
21/01/2025	00:00:13	27	27.45	14.09	Apagado
21/01/2025	23:59:55	27	27.45	14.09	Apagado
22/01/2025	00:00:58	27	27.45	14.09	Apagado
22/01/2025	10:45:09	25	27.45	14.09	Encendido
22/01/2025	10:45:40	25	27.45	14.09	Encendido
22/01/2025	10:46:11	25	27.45	14.09	Encendido
22/01/2025	10:46:43	25	27.45	14.09	Encendido
22/01/2025	10:47:14	25	27.45	14.09	Encendido
22/01/2025	23:59:48	26	27.45	14.09	Apagado
23/01/2025	00:00:19	26	27.45	14.09	Apagado
23/01/2025	23:59:48	26	27.45	14.09	Apagado
24/01/2025	00:00:19	26	27.45	14.09	Apagado
24/01/2025	23:59:51	26	27.45	14.09	Apagado

25/01/2025	00:00:22	26	27.45	14.09	Apagado
25/01/2025	23:59:32	26	27.45	14.09	Apagado
26/01/2025	00:00:03	26	27.45	14.09	Apagado
26/01/2025	23:59:28	26	27.45	14.09	Apagado
27/01/2025	00:00:00	26	27.45	14.09	Apagado
27/01/2025	15:30:56	24	27.45	14.09	Encendido
27/01/2025	15:31:27	24	27.45	14.09	Encendido
27/01/2025	15:31:59	25	27.45	14.09	Encendido
27/01/2025	15:32:31	25	27.45	14.09	Encendido
27/01/2025	23:59:50	27	27.45	14.09	Apagado
28/01/2025	00:00:22	28	27.45	14.09	Apagado
28/01/2025	23:59:50	27	27.45	14.09	Apagado
29/01/2025	00:00:22	28	27.45	14.09	Apagado
29/01/2025	23:59:36	28	27.45	14.09	Apagado
30/01/2025	00:00:07	27	27.45	14.09	Apagado
30/01/2025	23:59:55	27	27.45	14.09	Apagado
31/01/2025	00:00:26	26	27.45	14.09	Apagado
31/01/2025	10:40:03	25	27.45	14.09	Encendido
31/01/2025	10:40:34	25	27.45	14.09	Encendido
31/01/2025	10:41:05	25	27.45	14.09	Encendido
31/01/2025	10:41:36	25	27.45	14.09	Encendido
31/01/2025	23:59:45	27	27.45	14.09	Apagado
01/02/2025	00:00:16	27	27.45	14.09	Apagado
01/02/2025	23:59:38	28	27.45	14.09	Apagado
02/02/2025	00:00:10	28	27.45	14.09	Apagado
02/02/2025	23:59:45	28	27.45	14.09	Apagado
03/02/2025	00:00:16	27	27.45	14.09	Apagado
03/02/2025	23:59:38	28	27.45	14.09	Apagado
04/02/2025	00:00:10	28	27.45	14.09	Apagado
04/02/2025	10:18:51	24	27.45	14.09	Encendido
04/02/2025	10:19:54	24	27.45	14.09	Encendido
04/02/2025	10:20:25	24	27.45	14.09	Encendido
04/02/2025	10:20:57	25	27.45	14.09	Encendido
04/02/2025	10:21:28	25	27.45	14.09	Encendido
04/02/2025	10:21:59	25	27.45	14.09	Encendido
04/02/2025	10:22:31	25	27.45	14.09	Encendido
04/02/2025	10:23:02	25	27.45	14.09	Encendido
04/02/2025	23:59:38	28	27.45	14.09	Apagado
05/02/2025	00:00:10	28	27.45	14.09	Apagado
05/02/2025	23:59:59	27	27.45	14.09	Apagado
06/02/2025	00:00:30	27	27.45	14.09	Apagado
06/02/2025	23:59:42	28	27.45	14.09	Apagado
07/02/2025	00:00:14	27	27.45	14.09	Apagado
07/02/2025	23:59:49	27	27.45	14.09	Apagado

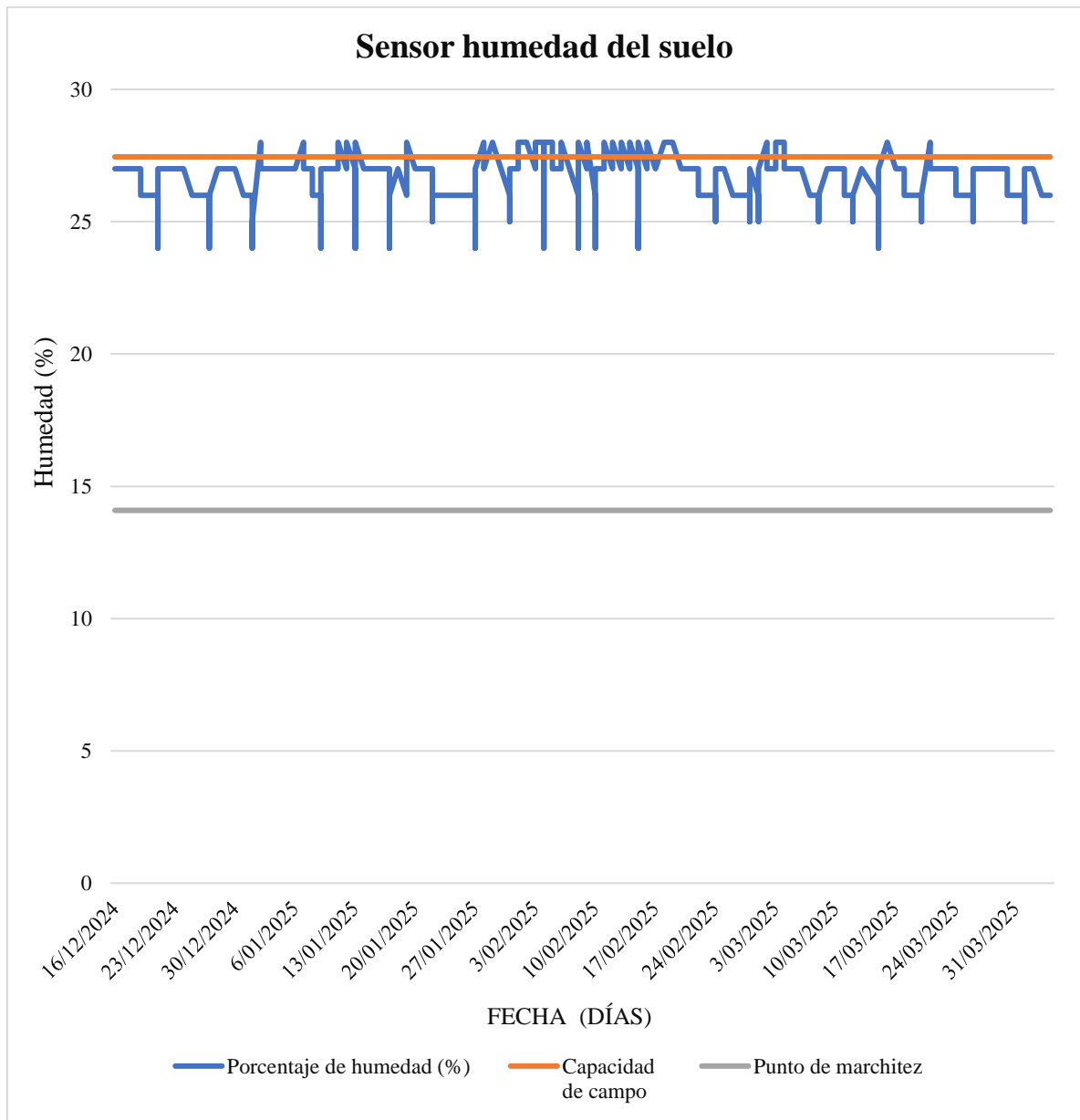
08/02/2025	00:00:20	26	27.45	14.09	Apagado
08/02/2025	08:10:20	24	27.45	14.09	Encendido
08/02/2025	08:10:51	24	27.45	14.09	Encendido
08/02/2025	08:11:23	24	27.45	14.09	Encendido
08/02/2025	08:11:54	25	27.45	14.09	Encendido
08/02/2025	08:12:26	25	27.45	14.09	Encendido
08/02/2025	08:12:57	25	27.45	14.09	Encendido
08/02/2025	08:13:28	25	27.45	14.09	Encendido
08/02/2025	23:59:42	28	27.45	14.09	Apagado
09/02/2025	00:00:14	27	27.45	14.09	Apagado
09/02/2025	23:59:42	28	27.45	14.09	Apagado
10/02/2025	00:00:14	26	27.45	14.09	Apagado
10/02/2025	15:04:56	24	27.45	14.09	Encendido
10/02/2025	15:06:40	24	27.45	14.09	Encendido
10/02/2025	15:07:10	25	27.45	14.09	Encendido
10/02/2025	15:07:42	25	27.45	14.09	Encendido
10/02/2025	15:08:13	25	27.45	14.09	Encendido
10/02/2025	15:08:45	25	27.45	14.09	Encendido
10/02/2025	23:59:49	27	27.45	14.09	Apagado
11/02/2025	00:00:20	27	27.45	14.09	Apagado
11/02/2025	23:59:42	28	27.45	14.09	Apagado
12/02/2025	00:00:14	27	27.45	14.09	Apagado
12/02/2025	23:59:42	28	27.45	14.09	Apagado
13/02/2025	00:00:14	27	27.45	14.09	Apagado
13/02/2025	23:59:42	28	27.45	14.09	Apagado
14/02/2025	00:00:14	27	27.45	14.09	Apagado
14/02/2025	23:59:42	28	27.45	14.09	Apagado
15/02/2025	00:00:14	27	27.45	14.09	Apagado
15/02/2025	05:49:47	25	27.45	14.09	Encendido
15/02/2025	05:50:19	24	27.45	14.09	Encendido
15/02/2025	05:50:50	25	27.45	14.09	Encendido
15/02/2025	05:51:23	24	27.45	14.09	Encendido
15/02/2025	05:51:54	24	27.45	14.09	Encendido
15/02/2025	05:52:26	25	27.45	14.09	Encendido
15/02/2025	05:52:57	25	27.45	14.09	Encendido
15/02/2025	23:59:42	28	27.45	14.09	Apagado
16/02/2025	00:00:14	27	27.45	14.09	Apagado
16/02/2025	23:59:42	28	27.45	14.09	Apagado
17/02/2025	00:00:14	27	27.45	14.09	Apagado
17/02/2025	23:59:56	27	27.45	14.09	Apagado
18/02/2025	00:00:27	28	27.45	14.09	Apagado
18/02/2025	19:47:28	28	27.45	14.09	Apagado
19/02/2025	08:38:13	28	27.45	14.09	Apagado
19/02/2025	23:59:31	28	27.45	14.09	Apagado

20/02/2025	00:00:02	27	27.45	14.09	Apagado
20/02/2025	23:59:42	27	27.45	14.09	Apagado
21/02/2025	00:00:13	27	27.45	14.09	Apagado
21/02/2025	23:59:55	27	27.45	14.09	Apagado
22/02/2025	00:00:58	27	27.45	14.09	Apagado
22/02/2025	23:59:48	26	27.45	14.09	Apagado
23/02/2025	00:00:19	26	27.45	14.09	Apagado
23/02/2025	23:59:48	26	27.45	14.09	Apagado
24/02/2025	00:00:19	26	27.45	14.09	Apagado
24/02/2025	17:05:10	25	27.45	14.09	Encendido
24/02/2025	17:05:42	25	27.45	14.09	Encendido
24/02/2025	17:06:14	25	27.45	14.09	Encendido
24/02/2025	23:59:51	27	27.45	14.09	Apagado
25/02/2025	00:00:22	27	27.45	14.09	Apagado
25/02/2025	23:59:32	27	27.45	14.09	Apagado
26/02/2025	00:00:03	26	27.45	14.09	Apagado
26/02/2025	23:59:28	26	27.45	14.09	Apagado
27/02/2025	00:00:00	26	27.45	14.09	Apagado
27/02/2025	23:59:38	26	27.45	14.09	Apagado
28/02/2025	00:00:09	26	27.45	14.09	Apagado
28/02/2025	19:28:29	25	27.45	14.09	Encendido
28/02/2025	19:29:03	25	27.45	14.09	Encendido
28/02/2025	19:29:38	25	27.45	14.09	Encendido
28/02/2025	19:30:10	25	27.45	14.09	Encendido
28/02/2025	19:30:41	25	27.45	14.09	Encendido
28/02/2025	23:59:40	27	27.45	14.09	Apagado
01/03/2025	00:00:11	26	27.45	14.09	Apagado
01/03/2025	15:29:18	25	27.45	14.09	Encendido
01/03/2025	15:29:51	25	27.45	14.09	Encendido
01/03/2025	15:30:22	25	27.45	14.09	Encendido
01/03/2025	15:30:54	25	27.45	14.09	Encendido
01/03/2025	15:31:25	25	27.45	14.09	Encendido
01/03/2025	23:59:57	27	27.45	14.09	Apagado
02/03/2025	00:00:28	28	27.45	14.09	Apagado
02/03/2025	23:59:55	27	27.45	14.09	Apagado
03/03/2025	00:00:26	27	27.45	14.09	Apagado
03/03/2025	23:59:28	28	27.45	14.09	Apagado
04/03/2025	00:00:00	28	27.45	14.09	Apagado
04/03/2025	23:59:38	27	27.45	14.09	Apagado
05/03/2025	00:00:09	27	27.45	14.09	Apagado
05/03/2025	23:59:40	27	27.45	14.09	Apagado
06/03/2025	00:00:11	27	27.45	14.09	Apagado
06/03/2025	23:59:46	27	27.45	14.09	Apagado
07/03/2025	00:00:17	26	27.45	14.09	Apagado

07/03/2025	23:59:30	26	27.45	14.09	Apagado
08/03/2025	00:00:02	26	27.45	14.09	Apagado
08/03/2025	17:34:03	25	27.45	14.09	Encendido
08/03/2025	17:34:37	25	27.45	14.09	Encendido
08/03/2025	17:34:47	25	27.45	14.09	Encendido
08/03/2025	17:35:21	25	27.45	14.09	Encendido
08/03/2025	17:35:52	25	27.45	14.09	Encendido
08/03/2025	23:59:28	26	27.45	14.09	Apagado
09/03/2025	00:00:00	27	27.45	14.09	Apagado
09/03/2025	23:59:38	27	27.45	14.09	Apagado
10/03/2025	00:00:09	27	27.45	14.09	Apagado
10/03/2025	23:59:40	27	27.45	14.09	Apagado
11/03/2025	00:00:11	27	27.45	14.09	Apagado
11/03/2025	23:59:46	26	27.45	14.09	Apagado
12/03/2025	00:00:17	26	27.45	14.09	Apagado
12/03/2025	18:30:47	25	27.45	14.09	Encendido
12/03/2025	18:31:18	25	27.45	14.09	Encendido
12/03/2025	18:31:50	25	27.45	14.09	Encendido
12/03/2025	18:32:24	25	27.45	14.09	Encendido
12/03/2025	18:32:56	25	27.45	14.09	Encendido
12/03/2025	18:33:28	25	27.45	14.09	Encendido
12/03/2025	18:34:06	25	27.45	14.09	Encendido
12/03/2025	18:37:54	26	27.45	14.09	Apagado
13/03/2025	09:18:47	27	27.45	14.09	Apagado
13/03/2025	23:20:46	27	27.45	14.09	Apagado
15/03/2025	09:38:53	26	27.45	14.09	Apagado
15/03/2025	10:16:27	24	27.45	14.09	Encendido
15/03/2025	10:17:01	24	27.45	14.09	Encendido
15/03/2025	10:17:32	25	27.45	14.09	Encendido
15/03/2025	10:17:43	25	27.45	14.09	Encendido
15/03/2025	10:18:17	25	27.45	14.09	Encendido
15/03/2025	10:20:20	25	27.45	14.09	Encendido
15/03/2025	10:20:51	25	27.45	14.09	Encendido
15/03/2025	12:05:59	27	27.45	14.09	Apagado
16/03/2025	11:02:03	28	27.45	14.09	Apagado
16/03/2025	13:08:44	28	27.45	14.09	Apagado
17/03/2025	09:06:15	27	27.45	14.09	Apagado
17/03/2025	23:59:47	27	27.45	14.09	Apagado
18/03/2025	00:00:19	27	27.45	14.09	Apagado
18/03/2025	23:59:43	26	27.45	14.09	Apagado
19/03/2025	00:00:14	26	27.45	14.09	Apagado
19/03/2025	23:59:31	26	27.45	14.09	Apagado
20/03/2025	00:00:03	26	27.45	14.09	Apagado
20/03/2025	10:23:33	25	27.45	14.09	Encendido

20/03/2025	10:24:05	25	27.45	14.09	Encendido
20/03/2025	10:24:36	25	27.45	14.09	Encendido
20/03/2025	10:25:08	25	27.45	14.09	Encendido
20/03/2025	11:25:27	26	27.45	14.09	Apagado
21/03/2025	12:54:10	28	27.45	14.09	Apagado
21/03/2025	23:59:57	27	27.45	14.09	Apagado
22/03/2025	00:00:29	27	27.45	14.09	Apagado
22/03/2025	23:59:28	27	27.45	14.09	Apagado
23/03/2025	00:00:00	27	27.45	14.09	Apagado
23/03/2025	23:59:35	27	27.45	14.09	Apagado
24/03/2025	00:00:07	27	27.45	14.09	Apagado
24/03/2025	23:59:48	26	27.45	14.09	Apagado
25/03/2025	00:00:19	26	27.45	14.09	Apagado
25/03/2025	23:59:48	26	27.45	14.09	Apagado
26/03/2025	00:00:19	26	27.45	14.09	Apagado
26/03/2025	21:05:20	25	27.45	14.09	Encendido
26/03/2025	21:05:51	25	27.45	14.09	Encendido
26/03/2025	21:06:23	25	27.45	14.09	Encendido
26/03/2025	21:06:57	25	27.45	14.09	Encendido
26/03/2025	21:07:29	25	27.45	14.09	Encendido
26/03/2025	23:59:41	27	27.45	14.09	Apagado
27/03/2025	00:00:12	27	27.45	14.09	Apagado
27/03/2025	23:59:52	27	27.45	14.09	Apagado
28/03/2025	00:00:23	27	27.45	14.09	Apagado
28/03/2025	17:33:50	27	27.45	14.09	Apagado
29/03/2025	07:55:59	27	27.45	14.09	Apagado
29/03/2025	23:59:39	27	27.45	14.09	Apagado
30/03/2025	00:00:11	27	27.45	14.09	Apagado
30/03/2025	23:59:29	26	27.45	14.09	Apagado
31/03/2025	00:00:01	26	27.45	14.09	Apagado
31/03/2025	23:59:47	26	27.45	14.09	Apagado
01/04/2025	00:00:18	26	27.45	14.09	Apagado
01/04/2025	08:20:15	25	27.45	14.09	Encendido
01/04/2025	08:20:47	25	27.45	14.09	Encendido
01/04/2025	08:21:18	25	27.45	14.09	Encendido
01/04/2025	08:21:53	25	27.45	14.09	Encendido
01/04/2025	08:22:24	25	27.45	14.09	Encendido
01/04/2025	23:59:47	27	27.45	14.09	Apagado
02/04/2025	00:00:18	27	27.45	14.09	Apagado
02/04/2025	23:59:59	27	27.45	14.09	Apagado
03/04/2025	00:00:31	26	27.45	14.09	Apagado
03/04/2025	23:59:35	26	27.45	14.09	Apagado
04/04/2025	00:00:07	26	27.45	14.09	Apagado
04/04/2025	09:25:28	26	27.45	14.09	Apagado

Figura 3. 1 Datos recolectados del sensor de humedad



- El sensor YF-DN50 funcionó con un rotor que giraba al pasar el agua. Este rotor tenía paletas magnéticas que, al moverse, generaban pulsos detectados por un sensor de efecto Hall. Luego, esos pulsos eran enviados a un Arduino, que los convertía en una medición del flujo de agua expresada en litros/min. Durante el funcionamiento del sistema de riego, se registró un flujo promedio de 4.5 l/s.

Tabla 3. 2 Resultado de la comunicación efectiva del sensor de flujo

Fecha	HORA	Flujo de agua (L/min)	Flujo de agua (L/s)
16/12/2024	16:06:07	0	0
16/12/2024	23:59:58	0	0
17/12/2024	00:00:09	0	0
17/12/2024	23:59:57	0	0
18/12/2024	00:00:08	0	0
18/12/2024	23:59:52	0	0
19/12/2024	00:00:03	0	0
19/12/2024	23:59:57	0	0
20/12/2024	00:00:07	0	0
20/12/2024	23:59:54	0	0
21/12/2024	00:00:05	0	0
21/12/2024	10:21:26	267.72	4.5
21/12/2024	10:21:56	267.72	4.5
21/12/2024	10:22:26	278.06	4.6
21/12/2024	10:22:56	312.9	5.2
21/12/2024	10:23:27	319.4	5.3
21/12/2024	10:24:01	309.06	5.2
21/12/2024	10:24:31	300.08	5.0
21/12/2024	10:25:02	300.08	5.0
21/12/2024	23:59:59	0	0
22/12/2024	00:00:09	0	0
22/12/2024	23:59:49	0	0
23/12/2024	00:00:00	0	0
23/12/2024	23:59:49	0	0
24/12/2024	00:00:00	0	0
24/12/2024	23:59:53	0	0
25/12/2024	00:00:04	0	0
25/12/2024	23:59:58	0	0
26/12/2024	00:00:08	0	0
26/12/2024	23:59:53	0	0
27/12/2024	00:00:03	0	0

27/12/2024	17:23:30	221.22	3.7
27/12/2024	17:24:01	319.4	5.3
27/12/2024	17:24:29	319.4	5.3
27/12/2024	17:25:00	319.4	5.3
27/12/2024	17:30:27	309.06	5.2
27/12/2024	17:30:58	309.06	5.2
27/12/2024	23:59:56	0	0
28/12/2024	00:00:06	0	0
28/12/2024	23:59:57	0	0
29/12/2024	00:00:07	0	0
29/12/2024	23:59:51	0	0
30/12/2024	00:00:01	0	0
30/12/2024	23:59:57	0	0
31/12/2024	00:00:08	0	0
31/12/2024	23:59:51	0	0
01/01/2025	00:00:01	0	0
01/01/2025	23:59:56	0	0
01/01/2025	14:38:03	278.06	4.6
01/01/2025	14:38:34	278.06	4.6
01/01/2025	14:39:05	312.9	5.2
01/01/2025	14:39:37	319.4	5.3
01/01/2025	14:40:08	309.06	5.2
01/01/2025	14:40:39	309.06	5.2
01/01/2025	14:41:10	309.06	5.2
01/01/2025	14:41:41	309.06	5.2
02/01/2025	00:00:06	0	0
02/01/2025	23:59:58	0	0
03/01/2025	00:00:08	0	0
03/01/2025	23:59:51	0	0
04/01/2025	00:00:01	0	0
04/01/2025	23:59:55	0	0
05/01/2025	00:00:05	0	0
05/01/2025	23:59:59	0	0
06/01/2025	00:00:10	0	0
06/01/2025	23:59:51	0	0
07/01/2025	00:00:01	0	0
07/01/2025	23:59:49	0	0
08/01/2025	00:00:20	0	0
08/01/2025	23:59:42	0	0
09/01/2025	00:00:14	0	0
09/01/2025	09:06:21	272.89	4.55
09/01/2025	09:06:52	272.89	4.55
09/01/2025	09:07:24	272.89	4.55
09/01/2025	09:07:56	371.07	6.18

09/01/2025	09:08:27	267.72	4.46
09/01/2025	09:08:59	371.07	6.18
09/01/2025	09:09:31	371.07	6.18
09/01/2025	23:59:42	0	0
10/01/2025	00:00:14	0	0
10/01/2025	23:59:49	0	0
11/01/2025	00:00:20	0	0
11/01/2025	23:59:42	0	0
12/01/2025	00:00:14	0	0
12/01/2025	23:59:42	0	0
13/01/2025	00:00:14	0	0
13/01/2025	06:07:06	371.07	6.2
13/01/2025	06:07:37	371.07	6.2
13/01/2025	06:08:09	267.72	4.5
13/01/2025	06:08:40	371.07	6.2
13/01/2025	06:09:11	267.72	4.5
13/01/2025	06:09:43	267.72	4.5
13/01/2025	06:10:14	267.72	4.5
13/01/2025	23:59:42	0	0
14/01/2025	00:00:14	0	0
14/01/2025	23:59:42	0	0
15/01/2025	00:00:14	0	0
15/01/2025	23:59:42	0	0
16/01/2025	00:00:14	0	0
16/01/2025	23:59:42	0	0
17/01/2025	00:00:14	0	0
17/01/2025	15:17:02	107.53	1.79
17/01/2025	15:17:34	138.54	2.31
17/01/2025	15:18:05	128.2	2.14
17/01/2025	15:18:36	107.53	1.79
17/01/2025	15:19:08	138.54	2.31
17/01/2025	15:19:39	128.2	2.14
17/01/2025	23:59:56	0	0
18/01/2025	00:00:27	0	0
18/01/2025	19:47:28	0	0
19/01/2025	08:38:13	0	0
19/01/2025	23:59:31	0	0
20/01/2025	00:00:02	0	0
20/01/2025	23:59:42	0	0
21/01/2025	00:00:13	0	0
21/01/2025	23:59:55	0	0
22/01/2025	00:00:58	0	0
22/01/2025	10:45:09	107.53	1.792
22/01/2025	10:45:40	138.54	2.309

22/01/2025	10:46:11	128.2	2.137
22/01/2025	10:46:43	128.2	2.137
22/01/2025	10:47:14	128.2	2.137
22/01/2025	23:59:48	0	0
23/01/2025	00:00:19	0	0
23/01/2025	23:59:48	0	0
24/01/2025	00:00:19	0	0
24/01/2025	23:59:51	0	0
25/01/2025	00:00:22	0	0
25/01/2025	23:59:32	0	0
26/01/2025	00:00:03	0	0
26/01/2025	23:59:28	0	0
27/01/2025	00:00:00	0	0
27/01/2025	15:30:56	371.07	6.2
27/01/2025	15:31:27	267.72	4.5
27/01/2025	15:31:59	371.07	6.2
27/01/2025	15:32:31	371.07	6.2
27/01/2025	23:59:50	0	0
28/01/2025	00:00:22	0	0
28/01/2025	23:59:50	0	0
29/01/2025	00:00:22	0	0
29/01/2025	23:59:36	0	0
30/01/2025	00:00:07	0	0
30/01/2025	23:59:55	0	0
31/01/2025	00:00:26	0	0
31/01/2025	10:40:03	312.9	5.2
31/01/2025	10:40:34	319.4	5.3
31/01/2025	10:41:05	309.06	5.2
31/01/2025	10:41:36	309.06	5.2
31/01/2025	23:59:45	0	0
01/02/2025	00:00:16	0	0
01/02/2025	23:59:38	0	0
02/02/2025	00:00:10	0	0
02/02/2025	23:59:45	0	0
03/02/2025	00:00:16	0	0
03/02/2025	23:59:38	0	0
04/02/2025	00:00:10	0	0
04/02/2025	10:18:51	312.9	5.22
04/02/2025	10:19:54	312.9	5.22
04/02/2025	10:20:25	312.9	5.22
04/02/2025	10:20:57	319.4	5.32
04/02/2025	10:21:28	309.06	5.15
04/02/2025	10:21:59	309.06	5.15
04/02/2025	10:22:31	267.72	4.46

04/02/2025	10:23:02	371.07	6.18
04/02/2025	23:59:38	0	0
05/02/2025	00:00:10	0	0
05/02/2025	23:59:59	0	0
06/02/2025	00:00:30	0	0
06/02/2025	23:59:42	0	0
07/02/2025	00:00:14	0	0
07/02/2025	23:59:49	0	0
08/02/2025	00:00:20	0	0
08/02/2025	08:10:20	226.38	3.77
08/02/2025	08:10:51	371.07	6.18
08/02/2025	08:11:23	371.07	6.18
08/02/2025	08:11:54	267.72	4.46
08/02/2025	08:12:26	267.72	4.46
08/02/2025	08:12:57	226.38	3.77
08/02/2025	08:13:28	226.38	3.77
08/02/2025	23:59:42	0	0
09/02/2025	00:00:14	0	0
09/02/2025	23:59:42	0	0
10/02/2025	00:00:14	0	0
10/02/2025	15:04:56	226.38	3.8
10/02/2025	15:06:40	226.38	3.8
10/02/2025	15:07:10	226.38	3.8
10/02/2025	15:07:42	371.07	6.2
10/02/2025	15:08:13	371.07	6.2
10/02/2025	15:08:45	371.07	6.2
10/02/2025	23:59:49	0	0
11/02/2025	00:00:20	0	0
11/02/2025	23:59:42	0	0
12/02/2025	00:00:14	0	0
12/02/2025	23:59:42	0	0
13/02/2025	00:00:14	0	0
13/02/2025	23:59:42	0	0
14/02/2025	00:00:14	0	0
14/02/2025	23:59:42	0	0
15/02/2025	00:00:14	0	0
15/02/2025	05:49:47	226.38	3.8
15/02/2025	05:50:19	226.38	3.8
15/02/2025	05:50:50	226.38	3.8
15/02/2025	05:51:23	371.07	6.2
15/02/2025	05:51:54	371.07	6.2
15/02/2025	05:52:26	371.07	6.2
15/02/2025	05:52:57	267.72	4.5
15/02/2025	23:59:42	0	0

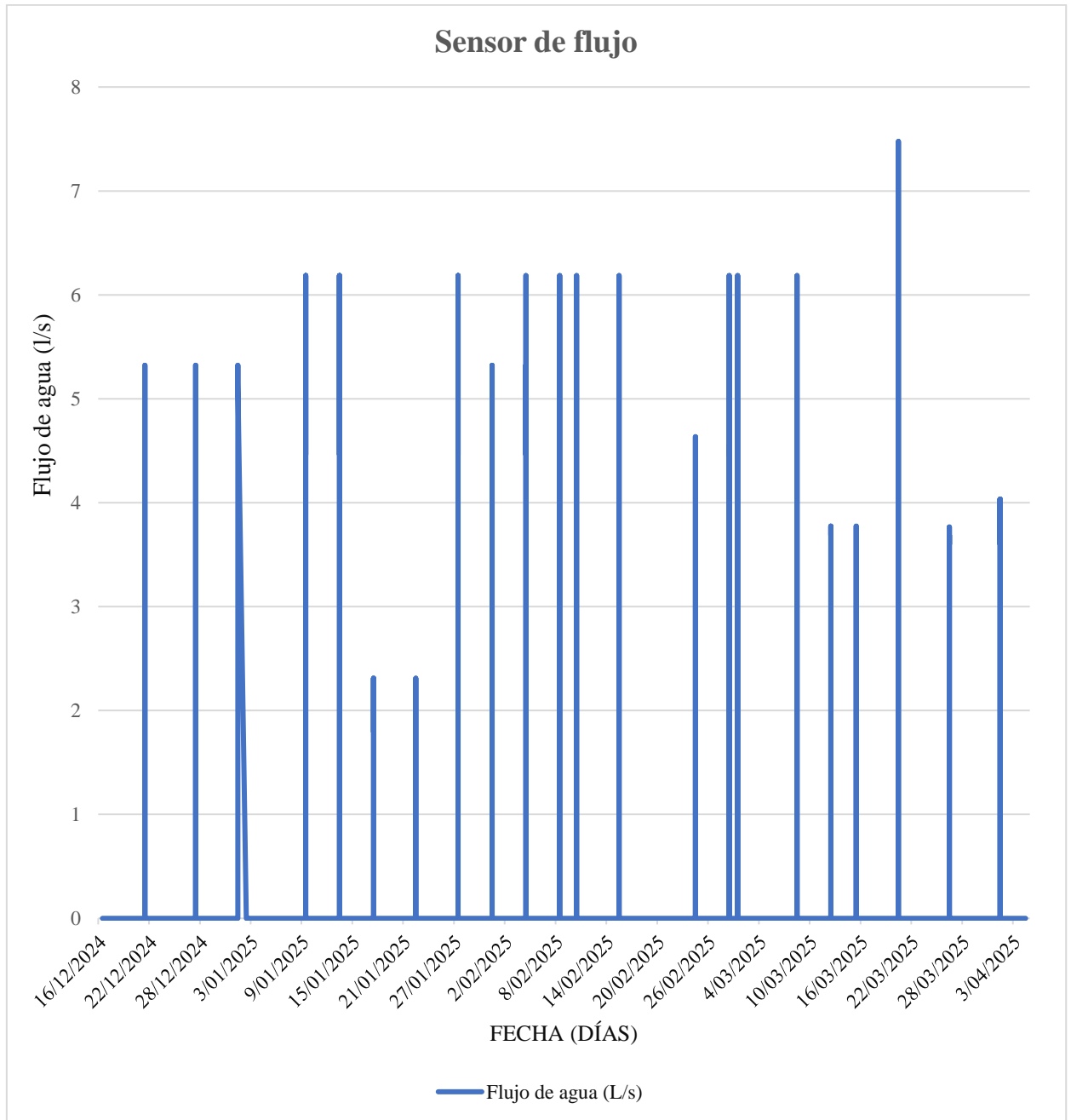
16/02/2025	00:00:14	0	0
16/02/2025	23:59:42	0	0
17/02/2025	00:00:14	0	0
17/02/2025	23:59:56	0	0
18/02/2025	00:00:27	0	0
18/02/2025	19:47:28	0	0
19/02/2025	08:38:13	0	0
19/02/2025	23:59:31	0	0
20/02/2025	00:00:02	0	0
20/02/2025	23:59:42	0	0
21/02/2025	00:00:13	0	0
21/02/2025	23:59:55	0	0
22/02/2025	00:00:58	0	0
22/02/2025	23:59:48	0	0
23/02/2025	00:00:19	0	0
23/02/2025	23:59:48	0	0
24/02/2025	00:00:19	0	0
24/02/2025	17:05:10	278.06	4.6
24/02/2025	17:05:42	226.38	3.8
24/02/2025	17:06:14	226.38	3.8
24/02/2025	23:59:51	0	0
25/02/2025	00:00:22	0	0
25/02/2025	23:59:32	0	0
26/02/2025	00:00:03	0	0
26/02/2025	23:59:28	0	0
27/02/2025	00:00:00	0	0
27/02/2025	23:59:38	0	0
28/02/2025	00:00:09	0	0
28/02/2025	19:28:29	272.89	4.5
28/02/2025	19:29:03	272.89	4.5
28/02/2025	19:29:38	267.72	4.5
28/02/2025	19:30:10	267.72	4.5
28/02/2025	19:30:41	371.07	6.2
28/02/2025	23:59:40	0	0
01/03/2025	00:00:11	0	0
01/03/2025	15:29:18	267.72	4.5
01/03/2025	15:29:51	272.89	4.5
01/03/2025	15:30:22	272.89	4.5
01/03/2025	15:30:54	371.07	6.2
01/03/2025	15:31:25	267.72	4.5
01/03/2025	23:59:57	0	0
02/03/2025	00:00:28	0	0
02/03/2025	23:59:55	0	0
03/03/2025	00:00:26	0	0

03/03/2025	23:59:28	0	0
04/03/2025	00:00:00	0	0
04/03/2025	23:59:38	0	0
05/03/2025	00:00:09	0	0
05/03/2025	23:59:40	0	0
06/03/2025	00:00:11	0	0
06/03/2025	23:59:46	0	0
07/03/2025	00:00:17	0	0
07/03/2025	23:59:30	0	0
08/03/2025	00:00:02	0	0
08/03/2025	17:34:03	226.38	3.8
08/03/2025	17:34:37	272.89	4.5
08/03/2025	17:34:47	371.07	6.2
08/03/2025	17:35:21	272.89	4.5
08/03/2025	17:35:52	221.22	3.7
08/03/2025	23:59:28	0	0
09/03/2025	00:00:00	0	0
09/03/2025	23:59:38	0	0
10/03/2025	00:00:09	0	0
10/03/2025	23:59:40	0	0
11/03/2025	00:00:11	0	0
11/03/2025	23:59:46	0	0
12/03/2025	00:00:17	0	0
12/03/2025	18:30:47	226.38	3.8
12/03/2025	18:31:18	221.22	3.7
12/03/2025	18:31:50	221.22	3.7
12/03/2025	18:32:24	221.22	3.7
12/03/2025	18:32:56	226.38	3.8
12/03/2025	18:33:28	221.22	3.7
12/03/2025	18:34:06	221.22	3.7
12/03/2025	18:37:54	0	0
13/03/2025	09:18:47	0	0
13/03/2025	23:20:46	0	0
15/03/2025	09:38:53	0	0
15/03/2025	10:16:27	216.05	3.6
15/03/2025	10:17:01	221.22	3.7
15/03/2025	10:17:32	221.22	3.7
15/03/2025	10:17:43	226.38	3.8
15/03/2025	10:18:17	221.22	3.7
15/03/2025	10:20:20	221.22	3.7
15/03/2025	10:20:51	216.05	3.6
15/03/2025	12:05:59	0	0
16/03/2025	11:02:03	0	0
16/03/2025	13:08:44	0	0

17/03/2025	09:06:15	0	0
17/03/2025	23:59:47	0	0
18/03/2025	00:00:19	0	0
18/03/2025	23:59:43	0	0
19/03/2025	00:00:14	0	0
19/03/2025	23:59:31	0	0
20/03/2025	00:00:03	0	0
20/03/2025	10:23:33	216.05	3.6
20/03/2025	10:24:05	371.07	6.2
20/03/2025	10:24:36	448.59	7.5
20/03/2025	10:25:08	216.05	3.6
20/03/2025	11:25:27	0	0
21/03/2025	12:54:10	0	0
21/03/2025	23:59:57	0	0
22/03/2025	00:00:29	0	0
22/03/2025	23:59:28	0	0
23/03/2025	00:00:00	0	0
23/03/2025	23:59:35	0	0
24/03/2025	00:00:07	0	0
24/03/2025	23:59:48	0	0
25/03/2025	00:00:19	0	0
25/03/2025	23:59:48	0	0
26/03/2025	00:00:19	0	0
26/03/2025	21:05:20	226	3.8
26/03/2025	21:05:51	216.05	3.6
26/03/2025	21:06:23	216.05	3.6
26/03/2025	21:06:57	221.22	3.7
26/03/2025	21:07:29	198.05	3.3
26/03/2025	23:59:41	0	0
27/03/2025	00:00:12	0	0
27/03/2025	23:59:52	0	0
28/03/2025	00:00:23	0	0
28/03/2025	17:33:50	0	0
29/03/2025	07:55:59	0	0
29/03/2025	23:59:39	0	0
30/03/2025	00:00:11	0	0
30/03/2025	23:59:29	0	0
31/03/2025	00:00:01	0	0
31/03/2025	23:59:47	0	0
01/04/2025	00:00:18	0	0
01/04/2025	08:20:15	242	4.0
01/04/2025	08:20:47	216.05	3.6
01/04/2025	08:21:18	242	4.0
01/04/2025	08:21:53	221.22	3.7

01/04/2025	08:22:24	159	2.7
01/04/2025	23:59:47	0	0
02/04/2025	00:00:18	0	0
02/04/2025	23:59:59	0	0
03/04/2025	00:00:31	0	0
03/04/2025	23:59:35	0	0
04/04/2025	00:00:07	0	0
04/04/2025	09:25:28	0	0

Figura 3. 2 Datos recolectados del sensor de flujo



- La presión del agua fue monitoreada con sensores HK3025. Se consideró que el filtro estaba en buen estado mientras la diferencia de presión no superara el 50% de la presión de entrada; de lo contrario, indicaba la necesidad de limpieza. Por esta razón, el filtro se limpió en promedio 10 veces

Tabla 3. 3 Resultado de la comunicación efectiva de los sensores de presión

Fecha	HORA	Presión 1 (KPa)	Presión 2 (kPa)	Estado de filtro
16/12/2024	16:06:07	20.81	12.99	Limpio
16/12/2024	23:59:58	20.81	13.99	Limpio
17/12/2024	00:00:09	20.81	9.89	Sucio
17/12/2024	23:59:57	22.3	11.01	Sucio
18/12/2024	00:00:08	22.3	11.01	Sucio
18/12/2024	23:59:52	22.3	11.01	Sucio
19/12/2024	00:00:03	22.3	11.01	Sucio
19/12/2024	23:59:57	20.81	13.99	Limpio
20/12/2024	00:00:07	20.81	12.99	Limpio
20/12/2024	23:59:54	20.81	12.99	Limpio
21/12/2024	00:00:05	22.3	11.01	Sucio
21/12/2024	10:21:26	20.81	11.01	Limpio
21/12/2024	10:21:56	22.3	11.01	Sucio
21/12/2024	10:22:26	20.81	11.01	Limpio
21/12/2024	10:22:56	20.81	12.99	Limpio
21/12/2024	10:23:27	22.3	12.99	Limpio
21/12/2024	10:24:01	20.81	12.99	Limpio
21/12/2024	10:24:31	20.81	12.99	Limpio
21/12/2024	10:25:02	20.81	12.99	Limpio
21/12/2024	23:59:59	22.3	12.99	Limpio
22/12/2024	00:00:09	22.3	13.99	Limpio
22/12/2024	23:59:49	22.3	12.99	Limpio
23/12/2024	00:00:00	22.3	12.99	Limpio
23/12/2024	23:59:49	20.81	12.99	Limpio
24/12/2024	00:00:00	22.3	12.99	Limpio
24/12/2024	23:59:53	20.81	13.99	Limpio
25/12/2024	00:00:04	20.81	12.99	Limpio
25/12/2024	23:59:58	20.81	11.01	Limpio
26/12/2024	00:00:08	22.3	11.01	Sucio
26/12/2024	23:59:53	20.81	11.01	Limpio
27/12/2024	00:00:03	17.83	12	Limpio
27/12/2024	17:23:30	5.05	5.05	Limpio
27/12/2024	17:24:01	4.78	4.78	Limpio

27/12/2024	17:24:29	5.32	5.32	Limpio
27/12/2024	17:25:00	5.32	5.32	Limpio
27/12/2024	17:30:27	4.78	4.78	Limpio
27/12/2024	17:30:58	4.78	4.78	Limpio
27/12/2024	23:59:56	5.05	1.44	Sucio
28/12/2024	00:00:06	5.05	2.01	Sucio
28/12/2024	23:59:57	5.05	1.44	Sucio
29/12/2024	00:00:07	5.05	1.44	Sucio
29/12/2024	23:59:51	5.05	1.44	Sucio
30/12/2024	00:00:01	4.78	2.01	Sucio
30/12/2024	23:59:57	5.05	1.44	Sucio
31/12/2024	00:00:08	4.78	0.87	Sucio
31/12/2024	23:59:51	4.51	0.87	Sucio
01/01/2025	00:00:01	5.05	1.44	Sucio
01/01/2025	23:59:56	4.78	4.78	Limpio
01/01/2025	14:38:03	6.68	6.68	Limpio
01/01/2025	14:38:34	4.78	4.78	Limpio
01/01/2025	14:39:05	5.32	5.32	Limpio
01/01/2025	14:39:37	4.78	4.78	Limpio
01/01/2025	14:40:08	4.78	4.78	Limpio
01/01/2025	14:40:39	4.78	4.78	Limpio
01/01/2025	14:41:10	5.32	5.32	Limpio
01/01/2025	14:41:41	4.78	4.78	Limpio
02/01/2025	00:00:06	4.51	4.51	Limpio
02/01/2025	23:59:58	8.31	8.31	Limpio
03/01/2025	00:00:08	8.03	8.03	Limpio
03/01/2025	23:59:51	9.12	9.12	Limpio
04/01/2025	00:00:01	8.58	8.58	Limpio
04/01/2025	23:59:55	8.31	8.31	Limpio
05/01/2025	00:00:05	8.31	8.31	Limpio
05/01/2025	23:59:59	7.22	7.22	Limpio
06/01/2025	00:00:10	6.68	6.68	Limpio
06/01/2025	23:59:51	7.49	7.49	Limpio
07/01/2025	00:00:01	7.76	7.76	Limpio
07/01/2025	23:59:49	4.78	4.78	Limpio
08/01/2025	00:00:20	5.05	5.05	Limpio
08/01/2025	23:59:42	5.32	5.32	Limpio
09/01/2025	00:00:14	5.05	5.05	Limpio
09/01/2025	09:06:21	4.51	4.51	Limpio
09/01/2025	09:06:52	4.24	4.24	Limpio
09/01/2025	09:07:24	4.51	4.51	Limpio
09/01/2025	09:07:56	4.78	4.78	Limpio
09/01/2025	09:08:27	4.51	4.51	Limpio
09/01/2025	09:08:59	4.51	4.51	Limpio

09/01/2025	09:09:31	4.51	4.51	Limpio
09/01/2025	23:59:42	5.32	5.32	Limpio
10/01/2025	00:00:14	5.05	5.05	Limpio
10/01/2025	23:59:49	4.78	4.78	Limpio
11/01/2025	00:00:20	5.05	5.05	Limpio
11/01/2025	23:59:42	5.32	5.32	Limpio
12/01/2025	00:00:14	5.05	5.05	Limpio
12/01/2025	23:59:42	5.32	5.32	Limpio
13/01/2025	00:00:14	5.05	5.05	Limpio
13/01/2025	06:07:06	5.32	5.32	Limpio
13/01/2025	06:07:37	5.32	5.32	Limpio
13/01/2025	06:08:09	5.32	5.32	Limpio
13/01/2025	06:08:40	5.32	5.32	Limpio
13/01/2025	06:09:11	5.32	5.32	Limpio
13/01/2025	06:09:43	5.32	5.32	Limpio
13/01/2025	06:10:14	5.32	5.32	Limpio
13/01/2025	23:59:42	5.32	5.32	Limpio
14/01/2025	00:00:14	5.05	5.05	Limpio
14/01/2025	23:59:42	5.32	5.32	Limpio
15/01/2025	00:00:14	5.05	5.05	Limpio
15/01/2025	23:59:42	5.32	5.32	Limpio
16/01/2025	00:00:14	5.05	5.05	Limpio
16/01/2025	23:59:42	5.32	5.32	Limpio
17/01/2025	00:00:14	5.05	5.05	Limpio
17/01/2025	15:17:02	4.51	4.51	Limpio
17/01/2025	15:17:34	4.51	4.51	Limpio
17/01/2025	15:18:05	4.24	4.24	Limpio
17/01/2025	15:18:36	4.24	4.24	Limpio
17/01/2025	15:19:08	4.24	4.24	Limpio
17/01/2025	15:19:39	4.24	4.24	Limpio
17/01/2025	23:59:56	5.59	5.59	Limpio
18/01/2025	00:00:27	5.59	5.59	Limpio
18/01/2025	19:47:28	6.41	6.41	Limpio
19/01/2025	08:38:13	5.86	5.86	Limpio
19/01/2025	23:59:31	5.32	5.32	Limpio
20/01/2025	00:00:02	5.05	5.05	Limpio
20/01/2025	23:59:42	5.86	5.86	Limpio
21/01/2025	00:00:13	5.86	5.86	Limpio
21/01/2025	23:59:55	6.14	6.14	Limpio
22/01/2025	00:00:58	6.14	6.14	Limpio
22/01/2025	10:45:09	5.32	5.32	Limpio
22/01/2025	10:45:40	5.32	5.32	Limpio
22/01/2025	10:46:11	5.32	5.32	Limpio
22/01/2025	10:46:43	5.59	5.59	Limpio

22/01/2025	10:47:14	5.32	5.32	Limpio
22/01/2025	23:59:48	6.14	6.14	Limpio
23/01/2025	00:00:19	6.14	6.14	Limpio
23/01/2025	23:59:48	6.68	6.68	Limpio
24/01/2025	00:00:19	6.41	6.41	Limpio
24/01/2025	23:59:51	6.95	6.95	Limpio
25/01/2025	00:00:22	6.95	6.95	Limpio
25/01/2025	23:59:32	6.95	6.95	Limpio
26/01/2025	00:00:03	6.95	6.95	Limpio
26/01/2025	23:59:28	6.14	6.14	Limpio
27/01/2025	00:00:00	6.41	6.41	Limpio
27/01/2025	15:30:56	7.76	7.76	Limpio
27/01/2025	15:31:27	6.14	6.14	Limpio
27/01/2025	15:31:59	6.14	6.14	Limpio
27/01/2025	15:32:31	6.14	6.14	Limpio
27/01/2025	23:59:50	5.86	5.86	Limpio
28/01/2025	00:00:22	5.86	5.86	Limpio
28/01/2025	23:59:50	5.86	5.86	Limpio
29/01/2025	00:00:22	5.86	5.86	Limpio
29/01/2025	23:59:36	5.32	5.32	Limpio
30/01/2025	00:00:07	5.32	5.32	Limpio
30/01/2025	23:59:55	5.32	5.32	Limpio
31/01/2025	00:00:26	5.32	5.32	Limpio
31/01/2025	10:40:03	6.41	6.41	Limpio
31/01/2025	10:40:34	6.41	6.41	Limpio
31/01/2025	10:41:05	5.32	5.32	Limpio
31/01/2025	10:41:36	5.32	5.32	Limpio
31/01/2025	23:59:45	6.41	6.41	Limpio
01/02/2025	00:00:16	6.14	6.14	Limpio
01/02/2025	23:59:38	5.86	5.86	Limpio
02/02/2025	00:00:10	5.86	5.86	Limpio
02/02/2025	23:59:45	6.41	6.41	Limpio
03/02/2025	00:00:16	6.41	6.41	Limpio
03/02/2025	23:59:38	5.32	5.32	Limpio
04/02/2025	00:00:10	5.32	5.32	Limpio
04/02/2025	10:18:51	6.14	6.14	Limpio
04/02/2025	10:19:54	6.14	6.14	Limpio
04/02/2025	10:20:25	6.14	6.14	Limpio
04/02/2025	10:20:57	6.14	6.14	Limpio
04/02/2025	10:21:28	5.86	5.86	Limpio
04/02/2025	10:21:59	5.86	5.86	Limpio
04/02/2025	10:22:31	6.14	6.14	Limpio
04/02/2025	10:23:02	5.32	5.32	Limpio
04/02/2025	23:59:38	5.32	5.32	Limpio

05/02/2025	00:00:10	5.32	5.32	Limpio
05/02/2025	23:59:59	5.05	5.05	Limpio
06/02/2025	00:00:30	5.32	5.32	Limpio
06/02/2025	23:59:42	5.32	5.32	Limpio
07/02/2025	00:00:14	5.05	5.05	Limpio
07/02/2025	23:59:49	4.78	4.78	Limpio
08/02/2025	00:00:20	5.05	5.05	Limpio
08/02/2025	08:10:20	5.32	5.32	Limpio
08/02/2025	08:10:51	5.32	5.32	Limpio
08/02/2025	08:11:23	5.05	5.05	Limpio
08/02/2025	08:11:54	5.05	5.05	Limpio
08/02/2025	08:12:26	5.05	5.05	Limpio
08/02/2025	08:12:57	5.05	5.05	Limpio
08/02/2025	08:13:28	5.32	5.32	Limpio
08/02/2025	23:59:42	5.32	5.32	Limpio
09/02/2025	00:00:14	5.05	5.05	Limpio
09/02/2025	23:59:42	5.32	5.32	Limpio
10/02/2025	00:00:14	5.05	5.05	Limpio
10/02/2025	15:04:56	4.51	4.51	Limpio
10/02/2025	15:06:40	4.24	4.24	Limpio
10/02/2025	15:07:10	4.24	4.24	Limpio
10/02/2025	15:07:42	3.97	3.97	Limpio
10/02/2025	15:08:13	4.24	4.24	Limpio
10/02/2025	15:08:45	4.24	4.24	Limpio
10/02/2025	23:59:49	4.78	4.78	Limpio
11/02/2025	00:00:20	5.05	5.05	Limpio
11/02/2025	23:59:42	5.32	5.32	Limpio
12/02/2025	00:00:14	5.05	5.05	Limpio
12/02/2025	23:59:42	5.32	5.32	Limpio
13/02/2025	00:00:14	5.05	5.05	Limpio
13/02/2025	23:59:42	5.32	5.32	Limpio
14/02/2025	00:00:14	5.05	5.05	Limpio
14/02/2025	23:59:42	5.32	5.32	Limpio
15/02/2025	00:00:14	5.05	5.05	Limpio
15/02/2025	05:49:47	5.32	5.32	Limpio
15/02/2025	05:50:19	5.05	5.05	Limpio
15/02/2025	05:50:50	5.32	5.32	Limpio
15/02/2025	05:51:23	5.32	5.32	Limpio
15/02/2025	05:51:54	5.32	5.32	Limpio
15/02/2025	05:52:26	5.05	5.05	Limpio
15/02/2025	05:52:57	5.32	5.32	Limpio
15/02/2025	23:59:42	5.32	5.32	Limpio
16/02/2025	00:00:14	5.05	5.05	Limpio
16/02/2025	23:59:42	5.32	5.32	Limpio

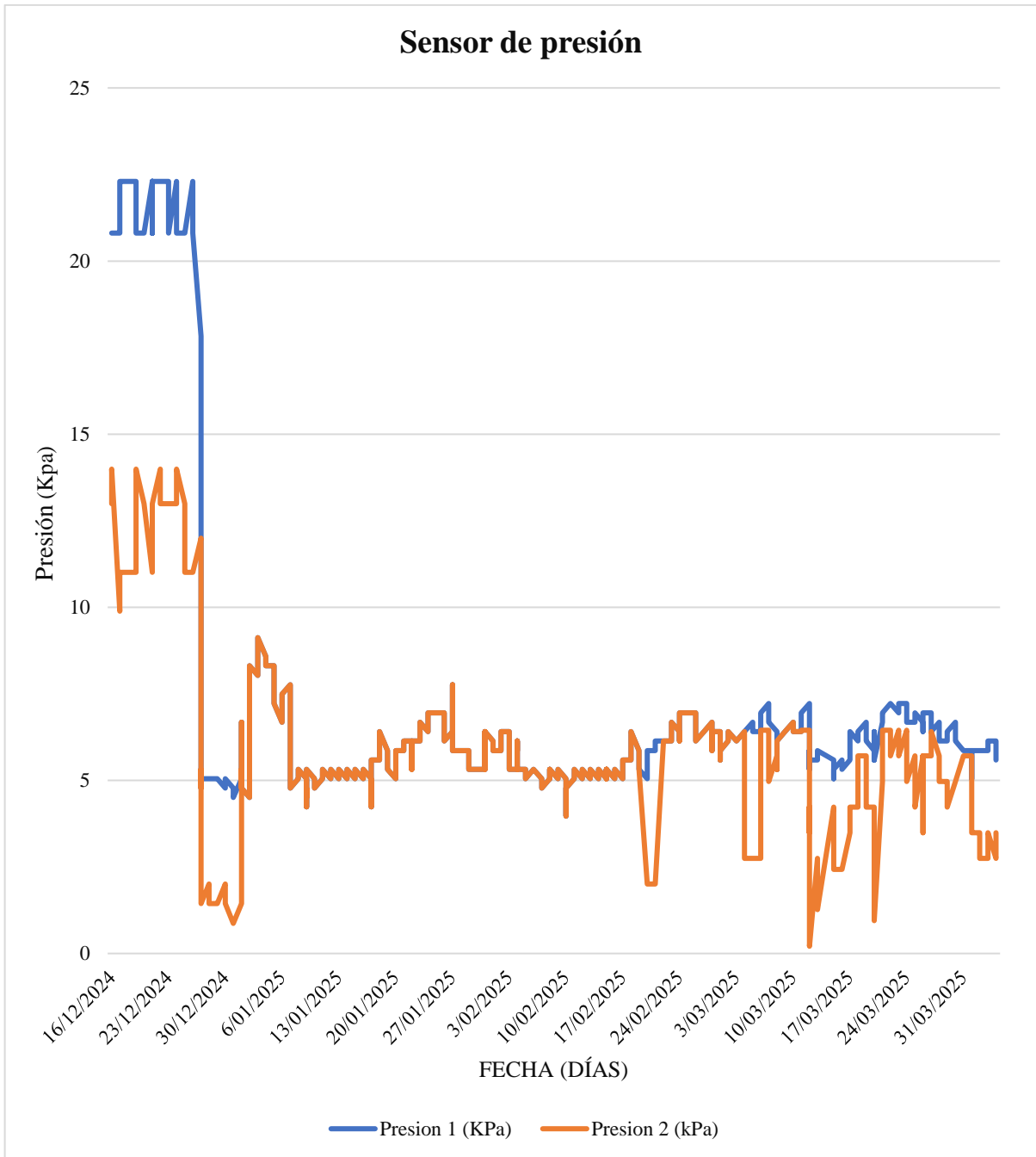
17/02/2025	00:00:14	5.05	5.05	Limpio
17/02/2025	23:59:56	5.59	5.59	Limpio
18/02/2025	00:00:27	5.59	5.59	Limpio
18/02/2025	19:47:28	6.41	6.41	Limpio
19/02/2025	08:38:13	5.86	5.86	Limpio
19/02/2025	23:59:31	5.32	5.32	Limpio
20/02/2025	00:00:02	5.05	2.01	sucio
20/02/2025	23:59:42	5.86	2.01	sucio
21/02/2025	00:00:13	5.86	2.01	sucio
21/02/2025	23:59:55	6.14	2.01	sucio
22/02/2025	00:00:58	6.14	6.14	Limpio
22/02/2025	23:59:48	6.14	6.14	Limpio
23/02/2025	00:00:19	6.14	6.14	Limpio
23/02/2025	23:59:48	6.68	6.68	Limpio
24/02/2025	00:00:19	6.41	6.41	Limpio
24/02/2025	17:05:10	6.14	6.14	Limpio
24/02/2025	17:05:42	6.14	6.14	Limpio
24/02/2025	17:06:14	6.14	6.14	Limpio
24/02/2025	23:59:51	6.95	6.95	Limpio
25/02/2025	00:00:22	6.95	6.95	Limpio
25/02/2025	23:59:32	6.95	6.95	Limpio
26/02/2025	00:00:03	6.95	6.95	Limpio
26/02/2025	23:59:28	6.14	6.14	Limpio
27/02/2025	00:00:00	6.41	6.41	Limpio
27/02/2025	23:59:38	6.41	6.41	Limpio
28/02/2025	00:00:09	6.68	6.68	Limpio
28/02/2025	19:28:29	6.41	6.41	Limpio
28/02/2025	19:29:03	6.14	6.14	Limpio
28/02/2025	19:29:38	6.14	6.14	Limpio
28/02/2025	19:30:10	6.41	6.41	Limpio
28/02/2025	19:30:41	5.86	5.86	Limpio
28/02/2025	23:59:40	6.41	6.41	Limpio
01/03/2025	00:00:11	6.41	6.41	Limpio
01/03/2025	15:29:18	5.59	5.59	Limpio
01/03/2025	15:29:51	5.59	5.59	Limpio
01/03/2025	15:30:22	5.59	5.59	Limpio
01/03/2025	15:30:54	5.59	5.59	Limpio
01/03/2025	15:31:25	5.59	5.59	Limpio
01/03/2025	23:59:57	5.86	5.86	Limpio
02/03/2025	00:00:28	6.14	6.14	Limpio
02/03/2025	23:59:55	6.41	6.41	Limpio
03/03/2025	00:00:26	6.14	6.14	Limpio
03/03/2025	23:59:28	6.14	6.14	Limpio
04/03/2025	00:00:00	6.41	6.41	Limpio

04/03/2025	23:59:38	6.41	2.75	sucio
05/03/2025	00:00:09	6.68	2.75	sucio
05/03/2025	23:59:40	6.41	2.75	sucio
06/03/2025	00:00:11	6.41	2.75	sucio
06/03/2025	23:59:46	6.95	6.45	Limpio
07/03/2025	00:00:17	7.22	6.45	Limpio
07/03/2025	23:59:30	6.68	4.97	Limpio
08/03/2025	00:00:02	6.41	5.71	Limpio
08/03/2025	17:34:03	5.32	5.32	Limpio
08/03/2025	17:34:37	5.59	5.59	Limpio
08/03/2025	17:34:47	5.59	5.32	Limpio
08/03/2025	17:35:21	5.32	5.32	Limpio
08/03/2025	17:35:52	5.59	5.59	Limpio
08/03/2025	23:59:28	6.14	6.14	Limpio
09/03/2025	00:00:00	6.41	6.41	Limpio
09/03/2025	23:59:38	6.41	6.41	Limpio
10/03/2025	00:00:09	6.68	6.68	Limpio
10/03/2025	23:59:40	6.41	6.41	Limpio
11/03/2025	00:00:11	6.41	6.41	Limpio
11/03/2025	23:59:46	6.95	6.45	Limpio
12/03/2025	00:00:17	7.22	6.45	Limpio
12/03/2025	18:30:47	5.86	3.49	Limpio
12/03/2025	18:31:18	5.32	3.49	Limpio
12/03/2025	18:31:50	5.86	3.49	Limpio
12/03/2025	18:32:24	5.86	3.49	Limpio
12/03/2025	18:32:56	5.32	4.23	Limpio
12/03/2025	18:33:28	5.86	3.49	Limpio
12/03/2025	18:34:06	5.86	4.23	Limpio
12/03/2025	18:37:54	5.59	0.21	sucio
13/03/2025	09:18:47	5.59	2.75	sucio
13/03/2025	23:20:46	5.86	1.27	sucio
15/03/2025	09:38:53	5.59	4.23	Limpio
15/03/2025	10:16:27	5.32	3.49	Limpio
15/03/2025	10:17:01	5.32	3.49	Limpio
15/03/2025	10:17:32	5.05	3.49	Limpio
15/03/2025	10:17:43	5.32	3.49	Limpio
15/03/2025	10:18:17	5.32	3.49	Limpio
15/03/2025	10:20:20	5.32	3.49	Limpio
15/03/2025	10:20:51	5.32	3.49	Limpio
15/03/2025	12:05:59	5.32	2.43	sucio
16/03/2025	11:02:03	5.59	2.43	sucio
16/03/2025	13:08:44	5.32	2.43	sucio
17/03/2025	09:06:15	5.59	3.49	Limpio
17/03/2025	23:59:47	6.41	4.23	Limpio

18/03/2025	00:00:19	6.14	4.23	Limpio
18/03/2025	23:59:43	6.41	5.71	Limpio
19/03/2025	00:00:14	6.68	5.71	Limpio
19/03/2025	23:59:31	6.14	4.23	Limpio
20/03/2025	00:00:03	5.86	4.23	Limpio
20/03/2025	10:23:33	5.59	4.23	Limpio
20/03/2025	10:24:05	6.41	4.23	Limpio
20/03/2025	10:24:36	6.14	3.49	Limpio
20/03/2025	10:25:08	5.86	3.49	Limpio
20/03/2025	11:25:27	5.59	0.95	Sucio
21/03/2025	12:54:10	6.68	4.97	Limpio
21/03/2025	23:59:57	6.95	6.45	Limpio
22/03/2025	00:00:29	7.22	6.45	Limpio
22/03/2025	23:59:28	7.22	5.71	Limpio
23/03/2025	00:00:00	6.95	6.45	Limpio
23/03/2025	23:59:35	7.22	5.71	Limpio
24/03/2025	00:00:07	7.22	6.45	Limpio
24/03/2025	23:59:48	6.68	4.97	Limpio
25/03/2025	00:00:19	6.68	5.71	Limpio
25/03/2025	23:59:48	6.95	4.23	Limpio
26/03/2025	00:00:19	6.68	5.71	Limpio
26/03/2025	21:05:20	6.68	5.71	Limpio
26/03/2025	21:05:51	6.41	4.97	Limpio
26/03/2025	21:06:23	6.68	5.71	Limpio
26/03/2025	21:06:57	6.95	3.49	Limpio
26/03/2025	21:07:29	6.95	5.71	Limpio
26/03/2025	23:59:41	6.95	5.71	Limpio
27/03/2025	00:00:12	6.95	5.71	Limpio
27/03/2025	23:59:52	6.41	6.41	Limpio
28/03/2025	00:00:23	6.68	5.71	Limpio
28/03/2025	17:33:50	6.14	4.97	Limpio
29/03/2025	07:55:59	6.14	4.97	Limpio
29/03/2025	23:59:39	6.41	4.23	Limpio
30/03/2025	00:00:11	6.68	4.97	Limpio
30/03/2025	23:59:29	6.14	4.97	Limpio
31/03/2025	00:00:01	5.86	5.71	Limpio
31/03/2025	23:59:47	5.86	5.71	Limpio
01/04/2025	00:00:18	5.86	5.71	Limpio
01/04/2025	08:20:15	5.32	3.49	Limpio
01/04/2025	08:20:47	5.05	3.49	Limpio
01/04/2025	08:21:18	5.32	3.49	Limpio
01/04/2025	08:21:53	5.32	3.49	Limpio
01/04/2025	08:22:24	5.59	3.49	Limpio
01/04/2025	23:59:47	5.86	3.49	Limpio

02/04/2025	00:00:18	5.86	3.49	Limpio
02/04/2025	23:59:59	5.86	2.75	sucio
03/04/2025	00:00:31	5.86	2.75	sucio
03/04/2025	23:59:35	6.14	3.49	Limpio
04/04/2025	00:00:07	6.14	2.75	sucio
04/04/2025	09:25:28	5.59	3.49	Limpio

Figura 3. 3 Datos recolectados del sensor de presiones



- El nivel de agua en el reservorio fue monitoreado de forma eficiente utilizando el sensor ultrasónico HC-SR04. Este enviaba lecturas al microcontrolador Arduino Mega 2560, lo que permitió un control preciso del almacenamiento. Se mantuvo un nivel mínimo de 0.6 m para asegurar la disponibilidad de agua cuando fuera necesario para el riego.

Tabla 3. 4 Resultado de la comunicación efectiva del sensor ultrasónico

Fecha	HORA	Nivel de agua (m)	Volumen (m3)
16/12/2024	16:06:07	1.5	23.04
16/12/2024	23:59:58	1.5	23.04
17/12/2024	00:00:09	1.5	23.04
17/12/2024	23:59:57	1.5	23.04
18/12/2024	00:00:08	1.5	23.04
18/12/2024	23:59:52	1.5	23.04
19/12/2024	00:00:03	1.5	23.04
19/12/2024	23:59:57	1.5	23.04
20/12/2024	00:00:07	1.5	23.04
20/12/2024	23:59:54	1.5	23.03
21/12/2024	00:00:05	1.5	23.03
21/12/2024	10:21:26	1.5	23.03
21/12/2024	10:21:56	1.5	23.03
21/12/2024	10:22:26	1.5	23.03
21/12/2024	10:22:56	1.49	22.73
21/12/2024	10:23:27	1.48	22.31
21/12/2024	10:24:01	1.49	22.44
21/12/2024	10:24:31	1.48	22.32
21/12/2024	10:25:02	1.43	20.39
21/12/2024	23:59:59	1.43	20.39
22/12/2024	00:00:09	1.42	20.18
22/12/2024	23:59:49	1.42	20.24
23/12/2024	00:00:00	1.43	20.39
23/12/2024	23:59:49	1.42	20.11
24/12/2024	00:00:00	1.42	20.11
24/12/2024	23:59:53	1.42	20.11
25/12/2024	00:00:04	1.42	20.11
25/12/2024	23:59:58	1.42	20.11
26/12/2024	00:00:08	1.42	20.11
26/12/2024	23:59:53	1.42	20.11
27/12/2024	00:00:03	1.42	20.11
27/12/2024	17:23:30	0.87	12.19
27/12/2024	17:24:01	0.87	12.19

27/12/2024	17:24:29	0.86	12.05
27/12/2024	17:25:00	0.86	12.05
27/12/2024	17:30:27	0.86	12.05
27/12/2024	17:30:58	0.86	12.05
27/12/2024	23:59:56	0.85	11.91
28/12/2024	00:00:06	0.85	11.91
28/12/2024	23:59:57	0.85	11.91
29/12/2024	00:00:07	0.85	11.91
29/12/2024	23:59:51	0.83	11.63
30/12/2024	00:00:01	0.83	11.63
30/12/2024	23:59:57	0.83	11.63
31/12/2024	00:00:08	0.83	11.63
31/12/2024	23:59:51	0.83	11.63
01/01/2025	00:00:01	0.83	11.63
01/01/2025	23:59:56	0.83	11.63
01/01/2025	14:38:03	0.83	11.63
01/01/2025	14:38:34	0.83	11.63
01/01/2025	14:39:05	0.82	11.49
01/01/2025	14:39:37	0.82	11.49
01/01/2025	14:40:08	0.82	11.49
01/01/2025	14:40:39	0.82	11.49
01/01/2025	14:41:10	0.82	11.49
01/01/2025	14:41:41	0.82	11.49
02/01/2025	00:00:06	0.88	12.33
02/01/2025	23:59:58	0.89	12.47
03/01/2025	00:00:08	0.9	12.61
03/01/2025	23:59:51	0.89	12.47
04/01/2025	00:00:01	0.89	12.47
04/01/2025	23:59:55	0.9	12.61
05/01/2025	00:00:05	0.9	12.61
05/01/2025	23:59:59	0.9	12.61
06/01/2025	00:00:10	0.9	12.61
06/01/2025	23:59:51	0.9	12.61
07/01/2025	00:00:01	0.9	12.61
07/01/2025	23:59:49	0.9	12.61
08/01/2025	00:00:20	0.9	12.61
08/01/2025	23:59:42	0.9	12.61
09/01/2025	00:00:14	0.9	12.61
09/01/2025	09:06:21	1.5	23.04
09/01/2025	09:06:52	1.5	23.04
09/01/2025	09:07:24	1.49	22.80
09/01/2025	09:07:56	1.49	22.80
09/01/2025	09:08:27	1.48	22.64
09/01/2025	09:08:59	1.48	22.64

09/01/2025	09:09:31	1.47	22.50
09/01/2025	23:59:42	1.45	21.19
10/01/2025	00:00:14	1.45	21.19
10/01/2025	23:59:49	1.45	21.19
11/01/2025	00:00:20	1.45	21.19
11/01/2025	23:59:42	1.45	21.19
12/01/2025	00:00:14	1.44	21.18
12/01/2025	23:59:42	1.44	21.18
13/01/2025	00:00:14	1.44	21.18
13/01/2025	06:07:06	1.48	22.22
13/01/2025	06:07:37	1.47	21.87
13/01/2025	06:08:09	1.47	21.87
13/01/2025	06:08:40	1.47	21.87
13/01/2025	06:09:11	1.46	21.52
13/01/2025	06:09:43	1.46	21.52
13/01/2025	06:10:14	1.46	21.52
13/01/2025	23:59:42	1.45	21.19
14/01/2025	00:00:14	1.44	21.18
14/01/2025	23:59:42	1.45	21.18
15/01/2025	00:00:14	1.45	21.18
15/01/2025	23:59:42	1.25	15.09
16/01/2025	00:00:14	1.25	15.09
16/01/2025	23:59:42	0.94	13.17
17/01/2025	00:00:14	0.94	13.17
17/01/2025	15:17:02	0.94	13.17
17/01/2025	15:17:34	0.94	13.17
17/01/2025	15:18:05	0.93	13.03
17/01/2025	15:18:36	0.93	13.03
17/01/2025	15:19:08	0.93	13.03
17/01/2025	15:19:39	0.89	12.47
17/01/2025	23:59:56	0.88	12.33
18/01/2025	00:00:27	0.88	12.33
18/01/2025	19:47:28	0.88	12.33
19/01/2025	08:38:13	0.88	12.33
19/01/2025	23:59:31	0.88	12.33
20/01/2025	00:00:02	0.61	8.54
20/01/2025	23:59:42	1.5	23.04
21/01/2025	00:00:13	1.5	23.04
21/01/2025	23:59:55	1.5	23.04
22/01/2025	00:00:58	1.5	23.04
22/01/2025	10:45:09	1.5	23.04
22/01/2025	10:45:40	1.48	22.36
22/01/2025	10:46:11	1.48	22.36
22/01/2025	10:46:43	1.46	21.66

22/01/2025	10:47:14	1.46	21.66
22/01/2025	23:59:48	1.46	21.66
23/01/2025	00:00:19	1.46	21.66
23/01/2025	23:59:48	1.46	21.66
24/01/2025	00:00:19	1.46	21.66
24/01/2025	23:59:51	1.46	21.66
25/01/2025	00:00:22	1.46	21.66
25/01/2025	23:59:32	1.46	21.66
26/01/2025	00:00:03	1.46	21.66
26/01/2025	23:59:28	1.45	21.13
27/01/2025	00:00:00	1.45	21.13
27/01/2025	15:30:56	1.15	16.75
27/01/2025	15:31:27	1.14	16.61
27/01/2025	15:31:59	1.14	16.61
27/01/2025	15:32:31	1.13	16.46
27/01/2025	23:59:50	1.11	16.17
28/01/2025	00:00:22	1.11	16.17
28/01/2025	23:59:50	1.11	16.17
29/01/2025	00:00:22	1.11	16.17
29/01/2025	23:59:36	1.11	16.17
30/01/2025	00:00:07	1.11	16.17
30/01/2025	23:59:55	1.11	16.17
31/01/2025	00:00:26	1.11	16.17
31/01/2025	10:40:03	1.1	6.02
31/01/2025	10:40:34	1.09	15.87
31/01/2025	10:41:05	1.09	15.87
31/01/2025	10:41:36	1.08	15.73
31/01/2025	23:59:45	1.11	16.17
01/02/2025	00:00:16	1.11	16.17
01/02/2025	23:59:38	1.12	16.32
02/02/2025	00:00:10	1.12	16.32
02/02/2025	23:59:45	1.11	16.17
03/02/2025	00:00:16	1.11	16.17
03/02/2025	23:59:38	1.12	16.32
04/02/2025	00:00:10	1.12	16.32
04/02/2025	10:18:51	1.12	16.32
04/02/2025	10:19:54	1.12	16.32
04/02/2025	10:20:25	1.11	16.17
04/02/2025	10:20:57	1.11	16.17
04/02/2025	10:21:28	1.1	16.02
04/02/2025	10:21:59	1.1	16.02
04/02/2025	10:22:31	1.09	15.88
04/02/2025	10:23:02	1.09	15.88
04/02/2025	23:59:38	1.05	15.3

05/02/2025	00:00:10	1.05	15.3
05/02/2025	23:59:59	1.05	15.3
06/02/2025	00:00:30	1.05	15.3
06/02/2025	23:59:42	1.05	15.3
07/02/2025	00:00:14	1.05	15.3
07/02/2025	23:59:49	1.05	15.3
08/02/2025	00:00:20	1.05	15.3
08/02/2025	08:10:20	1.2	17.48
08/02/2025	08:10:51	1.2	17.48
08/02/2025	08:11:23	1.19	17.34
08/02/2025	08:11:54	1.19	17.34
08/02/2025	08:12:26	1.18	17.19
08/02/2025	08:12:57	1.18	17.19
08/02/2025	08:13:28	1.17	17.04
08/02/2025	23:59:42	1.17	17.04
09/02/2025	00:00:14	1.17	17.04
09/02/2025	23:59:42	1.17	17.04
10/02/2025	00:00:14	1.17	17.04
10/02/2025	15:04:56	1.18	17.19
10/02/2025	15:06:40	1.18	17.19
10/02/2025	15:07:10	1.17	17.04
10/02/2025	15:07:42	1.17	17.04
10/02/2025	15:08:13	1.16	12.91
10/02/2025	15:08:45	1.16	16.90
10/02/2025	23:59:49	1.16	16.90
11/02/2025	00:00:20	1.16	16.90
11/02/2025	23:59:42	1.16	16.90
12/02/2025	00:00:14	1.16	16.90
12/02/2025	23:59:42	1.16	16.90
13/02/2025	00:00:14	1.16	16.90
13/02/2025	23:59:42	1.45	21.19
14/02/2025	00:00:14	1.44	21.18
14/02/2025	23:59:42	1.45	21.18
15/02/2025	00:00:14	1.45	21.18
15/02/2025	05:49:47	1.44	20.84
15/02/2025	05:50:19	1.44	20.84
15/02/2025	05:50:50	1.43	20.51
15/02/2025	05:51:23	1.43	20.51
15/02/2025	05:51:54	1.43	20.51
15/02/2025	05:52:26	1.42	20.18
15/02/2025	05:52:57	1.42	20.18
15/02/2025	23:59:42	1.42	20.18
16/02/2025	00:00:14	1.42	20.18
16/02/2025	23:59:42	1.42	20.18

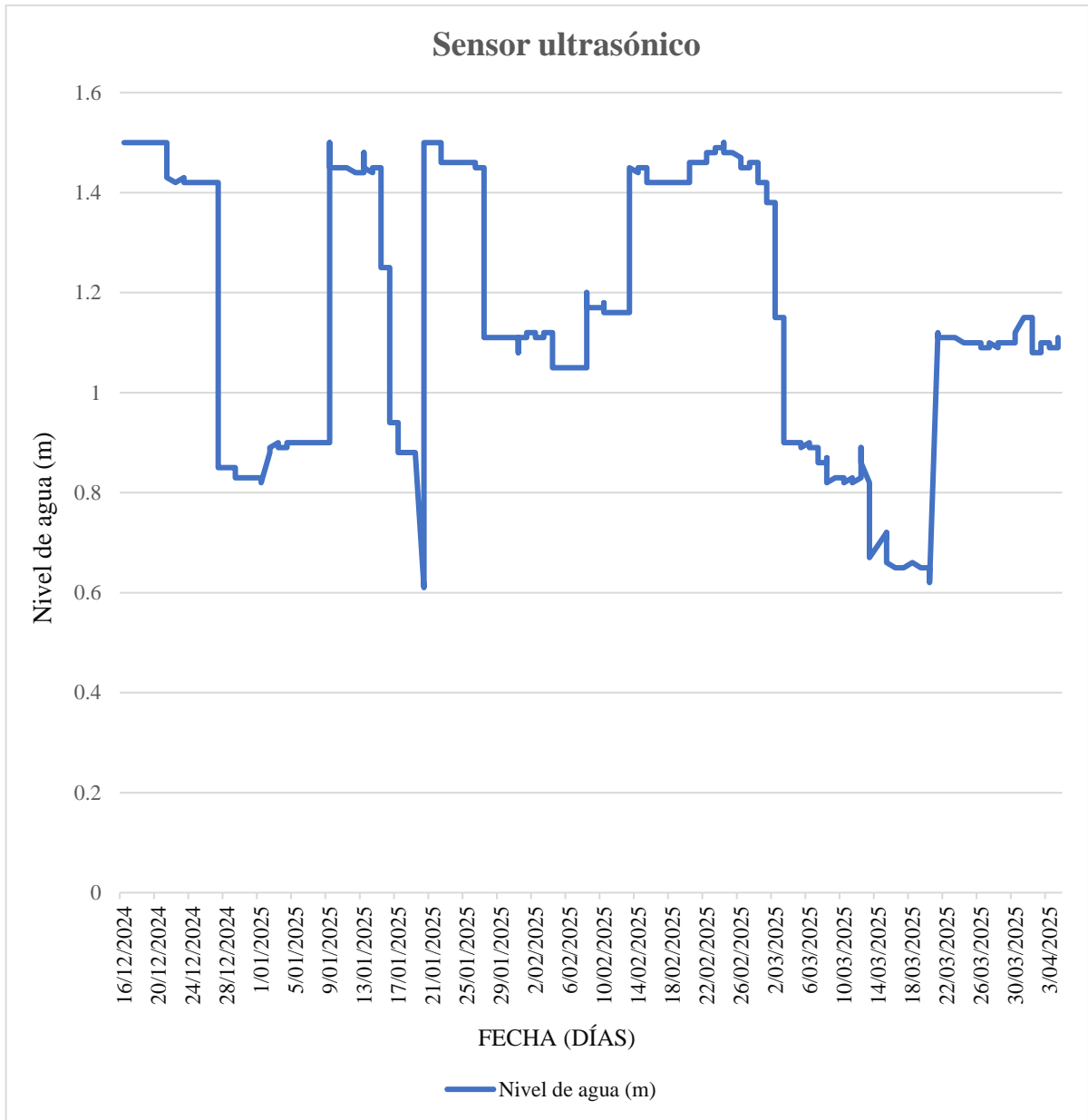
17/02/2025	00:00:14	1.42	20.18
17/02/2025	23:59:56	1.42	20.18
18/02/2025	00:00:27	1.42	20.18
18/02/2025	19:47:28	1.42	20.18
19/02/2025	08:38:13	1.42	20.18
19/02/2025	23:59:31	1.42	20.18
20/02/2025	00:00:02	1.42	20.18
20/02/2025	23:59:42	1.46	21.41
21/02/2025	00:00:13	1.46	21.56
21/02/2025	23:59:55	1.46	21.62
22/02/2025	00:00:58	1.46	21.59
22/02/2025	23:59:48	1.48	22.36
23/02/2025	00:00:19	1.48	22.36
23/02/2025	23:59:48	1.49	22.57
24/02/2025	00:00:19	1.49	22.56
24/02/2025	17:05:10	1.49	22.47
24/02/2025	17:05:42	1.5	22.85
24/02/2025	17:06:14	1.49	22.64
24/02/2025	23:59:51	1.48	22.19
25/02/2025	00:00:22	1.48	22.23
25/02/2025	23:59:32	1.48	22.19
26/02/2025	00:00:03	1.47	21.81
26/02/2025	23:59:28	1.45	21.13
27/02/2025	00:00:00	1.45	21.13
27/02/2025	23:59:38	1.46	21.6
28/02/2025	00:00:09	1.46	21.6
28/02/2025	19:28:29	1.46	21.6
28/02/2025	19:29:03	1.44	20.69
28/02/2025	19:29:38	1.43	20.39
28/02/2025	19:30:10	1.43	20.39
28/02/2025	19:30:41	1.43	20.38
28/02/2025	23:59:40	1.42	20.03
01/03/2025	00:00:11	1.42	20.04
01/03/2025	15:29:18	1.42	20.1
01/03/2025	15:29:51	1.42	20.26
01/03/2025	15:30:22	1.42	20.13
01/03/2025	15:30:54	1.41	19.93
01/03/2025	15:31:25	1.4	19.62
01/03/2025	23:59:57	1.38	18.8
02/03/2025	00:00:28	1.38	18.8
02/03/2025	23:59:55	1.15	16.11
03/03/2025	00:00:26	1.15	16.11
03/03/2025	23:59:28	0.9	12.61
04/03/2025	00:00:00	0.9	12.61

04/03/2025	23:59:38	0.9	12.61
05/03/2025	00:00:09	0.9	12.61
05/03/2025	23:59:40	0.89	12.47
06/03/2025	00:00:11	0.9	12.61
06/03/2025	23:59:46	0.89	12.47
07/03/2025	00:00:17	0.89	12.47
07/03/2025	23:59:30	0.86	12.05
08/03/2025	00:00:02	0.86	12.05
08/03/2025	17:34:03	0.87	12.19
08/03/2025	17:34:37	0.87	12.19
08/03/2025	17:34:47	0.87	12.19
08/03/2025	17:35:21	0.86	12.05
08/03/2025	17:35:52	0.86	12.05
08/03/2025	23:59:28	0.82	11.49
09/03/2025	00:00:00	0.83	11.63
09/03/2025	23:59:38	0.83	11.63
10/03/2025	00:00:09	0.83	11.63
10/03/2025	23:59:40	0.82	11.49
11/03/2025	00:00:11	0.83	11.63
11/03/2025	23:59:46	0.82	11.49
12/03/2025	00:00:17	0.83	11.63
12/03/2025	18:30:47	0.89	12.47
12/03/2025	18:31:18	0.88	12.33
12/03/2025	18:31:50	0.87	12.19
12/03/2025	18:32:24	0.87	12.19
12/03/2025	18:32:56	0.87	12.19
12/03/2025	18:33:28	0.87	12.19
12/03/2025	18:34:06	0.86	12.05
12/03/2025	18:37:54	0.86	12.05
13/03/2025	09:18:47	0.82	11.49
13/03/2025	23:20:46	0.67	9.38
15/03/2025	09:38:53	0.72	10.09
15/03/2025	10:16:27	0.72	10.09
15/03/2025	10:17:01	0.72	10.09
15/03/2025	10:17:32	0.71	9.95
15/03/2025	10:17:43	0.72	10.09
15/03/2025	10:18:17	0.72	10.09
15/03/2025	10:20:20	0.71	9.95
15/03/2025	10:20:51	0.71	9.95
15/03/2025	12:05:59	0.66	9.24
16/03/2025	11:02:03	0.65	9.1
16/03/2025	13:08:44	0.65	9.1
17/03/2025	09:06:15	0.65	9.1
17/03/2025	23:59:47	0.65	9.1

18/03/2025	00:00:19	0.66	9.24
18/03/2025	23:59:43	0.66	9.24
19/03/2025	00:00:14	0.65	9.1
19/03/2025	23:59:31	0.65	9.1
20/03/2025	00:00:03	0.65	9.1
20/03/2025	10:23:33	0.65	9.1
20/03/2025	10:24:05	0.64	8.96
20/03/2025	10:24:36	0.64	8.96
20/03/2025	10:25:08	0.63	8.83
20/03/2025	11:25:27	0.62	8.68
21/03/2025	12:54:10	1.12	15.69
21/03/2025	23:59:57	1.11	15.55
22/03/2025	00:00:29	1.11	15.55
22/03/2025	23:59:28	1.11	15.55
23/03/2025	00:00:00	1.11	15.55
23/03/2025	23:59:35	1.11	15.55
24/03/2025	00:00:07	1.1	15.41
24/03/2025	23:59:48	1.1	15.41
25/03/2025	00:00:19	1.1	15.41
25/03/2025	23:59:48	1.1	15.41
26/03/2025	00:00:19	1.1	15.41
26/03/2025	21:05:20	1.1	15.41
26/03/2025	21:05:51	1.1	15.41
26/03/2025	21:06:23	1.1	15.41
26/03/2025	21:06:57	1.09	15.27
26/03/2025	21:07:29	1.09	15.27
26/03/2025	23:59:41	1.09	15.27
27/03/2025	00:00:12	1.09	15.27
27/03/2025	23:59:52	1.1	15.41
28/03/2025	00:00:23	1.09	15.27
28/03/2025	17:33:50	1.1	15.41
29/03/2025	07:55:59	1.1	15.41
29/03/2025	23:59:39	1.1	15.41
30/03/2025	00:00:11	1.1	15.41
30/03/2025	23:59:29	1.12	15.69
31/03/2025	00:00:01	1.15	16.11
31/03/2025	23:59:47	1.15	16.11
01/04/2025	00:00:18	1.15	16.11
01/04/2025	08:20:15	1.15	16.11
01/04/2025	08:20:47	1.12	15.69
01/04/2025	08:21:18	1.1	15.41
01/04/2025	08:21:53	1.1	15.41
01/04/2025	08:22:24	1.09	15.27
01/04/2025	23:59:47	1.08	15.13

02/04/2025	00:00:18	1.08	15.13
02/04/2025	23:59:59	1.1	15.41
03/04/2025	00:00:31	1.1	15.41
03/04/2025	23:59:35	1.09	15.27
04/04/2025	00:00:07	1.09	15.27
04/04/2025	09:25:28	1.11	15.55

Figura 3. 4 Datos recolectados del sensor ultrasónico



- La temperatura dentro del invernadero se registró de manera efectiva mediante el sensor, el cual proporcionó datos precisos y continuos durante todo el periodo del cultivo. Esto permitió monitorear las variaciones térmicas en el interior del invernadero y evidenciar la influencia directa de la temperatura en la evapotranspiración. A mayor temperatura, mayor fue la necesidad de riego. La temperatura más alta registrada fue de 30.76 °C y la más baja de 12.11 °C

Tabla 3. 5 Resultado de la comunicación efectiva del sensor de temperatura

Fecha	HORA	Temperatura (°C)
16/12/2024	16:06:07	18.55
16/12/2024	23:59:58	13.87
17/12/2024	00:00:09	13.87
17/12/2024	23:59:57	14.36
18/12/2024	00:00:08	14.36
18/12/2024	23:59:52	14.80
19/12/2024	00:00:03	14.80
19/12/2024	23:59:57	14.36
20/12/2024	00:00:07	14.36
20/12/2024	23:59:54	13.87
21/12/2024	00:00:05	13.87
21/12/2024	10:21:26	30.49
21/12/2024	10:21:56	30.21
21/12/2024	10:22:26	30.54
21/12/2024	10:22:56	30.76
21/12/2024	10:23:27	30.76
21/12/2024	10:24:01	30.76
21/12/2024	10:24:31	30.49
21/12/2024	10:25:02	30.49
21/12/2024	23:59:59	15.63
22/12/2024	00:00:09	15.63
22/12/2024	23:59:49	15.41
23/12/2024	00:00:00	15.41
23/12/2024	23:59:49	13.87
24/12/2024	00:00:00	13.87
24/12/2024	23:59:53	14.8
25/12/2024	00:00:04	14.8
25/12/2024	23:59:58	15.63
26/12/2024	00:00:08	15.63
26/12/2024	23:59:53	15.41
27/12/2024	00:00:03	15.41

27/12/2024	17:23:30	16.51
27/12/2024	17:24:01	16.51
27/12/2024	17:24:29	16.51
27/12/2024	17:25:00	16.51
27/12/2024	17:30:27	16.51
27/12/2024	17:30:58	16.51
27/12/2024	23:59:56	12.55
28/12/2024	00:00:06	12.55
28/12/2024	23:59:57	13.65
29/12/2024	00:00:07	13.65
29/12/2024	23:59:51	13.43
30/12/2024	00:00:01	13.37
30/12/2024	23:59:57	15.19
31/12/2024	00:00:08	15.19
31/12/2024	23:59:51	13.87
01/01/2025	00:00:01	13.87
01/01/2025	23:59:56	15.02
01/01/2025	14:38:03	28.23
01/01/2025	14:38:34	28.78
01/01/2025	14:39:05	28.78
01/01/2025	14:39:37	28.78
01/01/2025	14:40:08	28.78
01/01/2025	14:40:39	28.23
01/01/2025	14:41:10	28.23
01/01/2025	14:41:41	28.23
02/01/2025	00:00:06	15.02
02/01/2025	23:59:58	13.37
03/01/2025	00:00:08	13.37
03/01/2025	23:59:51	12.55
04/01/2025	00:00:01	12.55
04/01/2025	23:59:55	13.87
05/01/2025	00:00:05	13.87
05/01/2025	23:59:59	13.87
06/01/2025	00:00:10	13.87
06/01/2025	23:59:51	14.36
07/01/2025	00:00:01	14.36
07/01/2025	23:59:49	14.8
08/01/2025	00:00:20	14.8
08/01/2025	23:59:42	12.33
09/01/2025	00:00:14	12.33
09/01/2025	09:06:21	20.09
09/01/2025	09:06:52	20.09
09/01/2025	09:07:24	19.87
09/01/2025	09:07:56	19.87

09/01/2025	09:08:27	19.87
09/01/2025	09:08:59	19.87
09/01/2025	09:09:31	19.98
09/01/2025	23:59:42	12.33
10/01/2025	00:00:14	12.33
10/01/2025	23:59:49	14.8
11/01/2025	00:00:20	14.8
11/01/2025	23:59:42	12.33
12/01/2025	00:00:14	12.33
12/01/2025	23:59:42	12.33
13/01/2025	00:00:14	12.33
13/01/2025	06:07:06	24.49
13/01/2025	06:07:37	24.49
13/01/2025	06:08:09	24.49
13/01/2025	06:08:40	24.49
13/01/2025	06:09:11	24.49
13/01/2025	06:09:43	24.49
13/01/2025	06:10:14	24.49
13/01/2025	23:59:42	12.33
14/01/2025	00:00:14	12.33
14/01/2025	23:59:42	12.33
15/01/2025	00:00:14	12.33
15/01/2025	23:59:42	12.33
16/01/2025	00:00:14	12.33
16/01/2025	23:59:42	12.33
17/01/2025	00:00:14	12.33
17/01/2025	15:17:02	17.67
17/01/2025	15:17:34	17.67
17/01/2025	15:18:05	17.67
17/01/2025	15:18:36	17.67
17/01/2025	15:19:08	17.67
17/01/2025	15:19:39	17.67
17/01/2025	23:59:56	14.36
18/01/2025	00:00:27	14.36
18/01/2025	19:47:28	13.65
19/01/2025	08:38:13	18.55
19/01/2025	23:59:31	14.58
20/01/2025	00:00:02	14.58
20/01/2025	23:59:42	14.03
21/01/2025	00:00:13	14.09
21/01/2025	23:59:55	13.21
22/01/2025	00:00:58	13.37
22/01/2025	10:45:09	21.52
22/01/2025	10:45:40	21.52

22/01/2025	10:46:11	21.52
22/01/2025	10:46:43	21.35
22/01/2025	10:47:14	21.35
22/01/2025	23:59:48	14.36
23/01/2025	00:00:19	14.36
23/01/2025	23:59:48	14.09
24/01/2025	00:00:19	14.09
24/01/2025	23:59:51	12.77
25/01/2025	00:00:22	12.55
25/01/2025	23:59:32	14.09
26/01/2025	00:00:03	14.09
26/01/2025	23:59:28	12.77
27/01/2025	00:00:00	12.77
27/01/2025	15:30:56	25.37
27/01/2025	15:31:27	25.65
27/01/2025	15:31:59	25.37
27/01/2025	15:32:31	24.82
27/01/2025	23:59:50	12.99
28/01/2025	00:00:22	12.99
28/01/2025	23:59:50	12.99
29/01/2025	00:00:22	12.99
29/01/2025	23:59:36	14.36
30/01/2025	00:00:07	14.36
30/01/2025	23:59:55	14.09
31/01/2025	00:00:26	14.09
31/01/2025	10:40:03	17.23
31/01/2025	10:40:34	17.23
31/01/2025	10:41:05	17.23
31/01/2025	10:41:36	17.23
31/01/2025	23:59:45	14.09
01/02/2025	00:00:16	14.09
01/02/2025	23:59:38	15.02
02/02/2025	00:00:10	15.02
02/02/2025	23:59:45	14.09
03/02/2025	00:00:16	14.09
03/02/2025	23:59:38	13.65
04/02/2025	00:00:10	13.65
04/02/2025	10:18:51	17.01
04/02/2025	10:19:54	17.01
04/02/2025	10:20:25	17.01
04/02/2025	10:20:57	17.01
04/02/2025	10:21:28	17.01
04/02/2025	10:21:59	17.01
04/02/2025	10:22:31	16.73

04/02/2025	10:23:02	16.73
04/02/2025	23:59:38	13.65
05/02/2025	00:00:10	13.65
05/02/2025	23:59:59	14.36
06/02/2025	00:00:30	14.36
06/02/2025	23:59:42	12.33
07/02/2025	00:00:14	12.33
07/02/2025	23:59:49	14.8
08/02/2025	00:00:20	14.8
08/02/2025	08:10:20	14.36
08/02/2025	08:10:51	14.58
08/02/2025	08:11:23	14.58
08/02/2025	08:11:54	14.58
08/02/2025	08:12:26	14.58
08/02/2025	08:12:57	14.36
08/02/2025	08:13:28	16.12
08/02/2025	23:59:42	12.33
09/02/2025	00:00:14	12.33
09/02/2025	23:59:42	12.33
10/02/2025	00:00:14	12.33
10/02/2025	15:04:56	19.81
10/02/2025	15:06:40	19.81
10/02/2025	15:07:10	19.43
10/02/2025	15:07:42	19.21
10/02/2025	15:08:13	19.21
10/02/2025	15:08:45	19.21
10/02/2025	23:59:49	14.8
11/02/2025	00:00:20	14.8
11/02/2025	23:59:42	12.33
12/02/2025	00:00:14	12.33
12/02/2025	23:59:42	12.33
13/02/2025	00:00:14	12.33
13/02/2025	23:59:42	12.33
14/02/2025	00:00:14	12.33
14/02/2025	23:59:42	12.33
15/02/2025	00:00:14	12.33
15/02/2025	05:49:47	12.11
15/02/2025	05:50:19	12.11
15/02/2025	05:50:50	12.11
15/02/2025	05:51:23	12.11
15/02/2025	05:51:54	12.11
15/02/2025	05:52:26	12.11
15/02/2025	05:52:57	12.11
15/02/2025	23:59:42	12.33

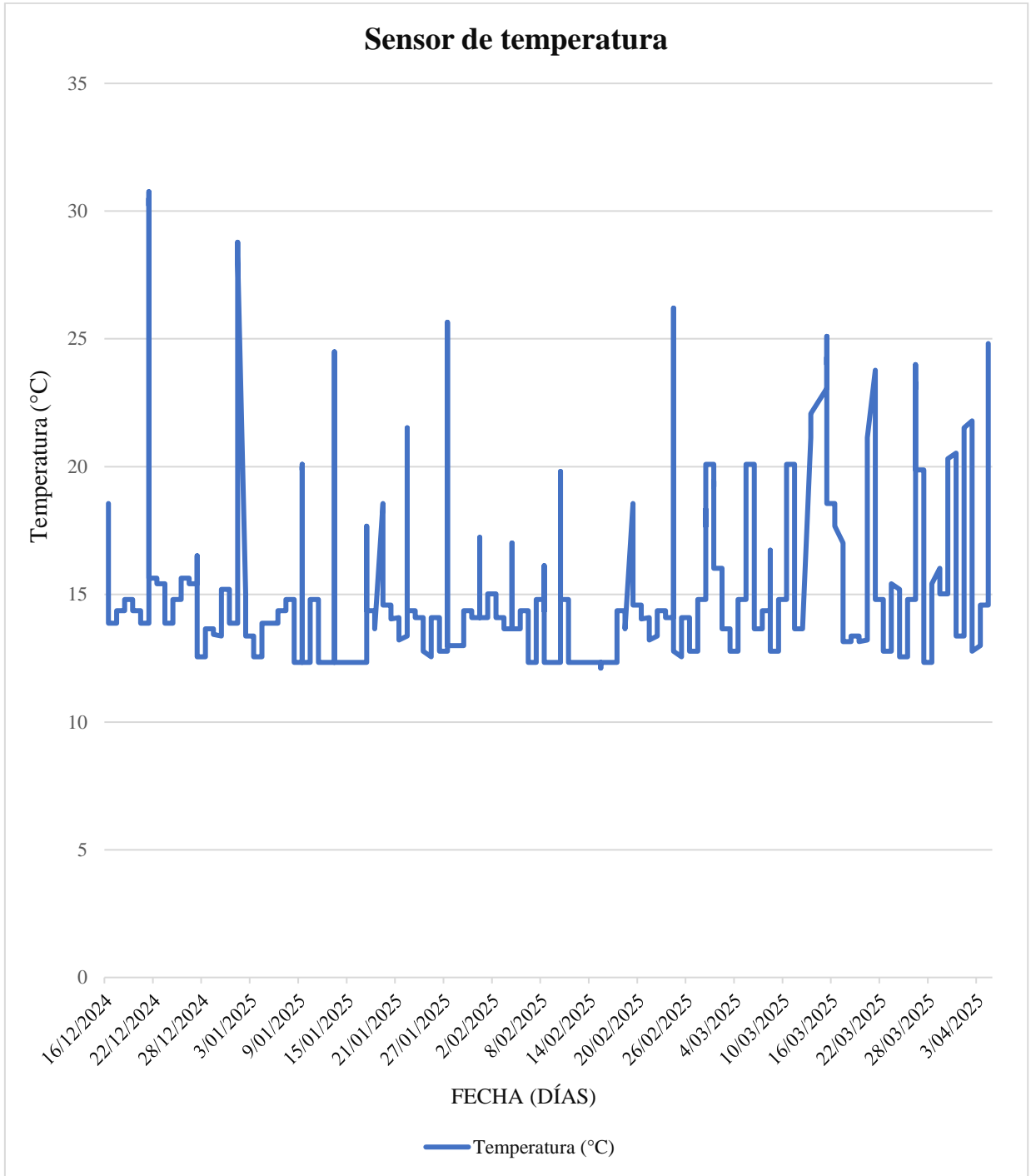
16/02/2025	00:00:14	12.33
16/02/2025	23:59:42	12.33
17/02/2025	00:00:14	12.33
17/02/2025	23:59:56	14.36
18/02/2025	00:00:27	14.36
18/02/2025	19:47:28	13.65
19/02/2025	08:38:13	18.55
19/02/2025	23:59:31	14.58
20/02/2025	00:00:02	14.58
20/02/2025	23:59:42	14.03
21/02/2025	00:00:13	14.09
21/02/2025	23:59:55	13.21
22/02/2025	00:00:58	13.37
22/02/2025	23:59:48	14.36
23/02/2025	00:00:19	14.36
23/02/2025	23:59:48	14.09
24/02/2025	00:00:19	14.09
24/02/2025	17:05:10	26.2
24/02/2025	17:05:42	26.2
24/02/2025	17:06:14	25.92
24/02/2025	23:59:51	12.77
25/02/2025	00:00:22	12.55
25/02/2025	23:59:32	14.09
26/02/2025	00:00:03	14.09
26/02/2025	23:59:28	12.77
27/02/2025	00:00:00	12.77
27/02/2025	23:59:38	14.8
28/02/2025	00:00:09	14.8
28/02/2025	19:28:29	18.33
28/02/2025	19:29:03	18.11
28/02/2025	19:29:38	18
28/02/2025	19:30:10	17.89
28/02/2025	19:30:41	17.67
28/02/2025	23:59:40	20.09
01/03/2025	00:00:11	20.09
01/03/2025	15:29:18	19.21
01/03/2025	15:29:51	19.43
01/03/2025	15:30:22	19.43
01/03/2025	15:30:54	19.21
01/03/2025	15:31:25	19.21
01/03/2025	23:59:57	16.02
02/03/2025	00:00:28	16.02
02/03/2025	23:59:55	13.65
03/03/2025	00:00:26	13.65

03/03/2025	23:59:28	12.77
04/03/2025	00:00:00	12.77
04/03/2025	23:59:38	14.8
05/03/2025	00:00:09	14.8
05/03/2025	23:59:40	20.09
06/03/2025	00:00:11	20.09
06/03/2025	23:59:46	13.65
07/03/2025	00:00:17	13.65
07/03/2025	23:59:30	14.36
08/03/2025	00:00:02	14.36
08/03/2025	17:34:03	16.73
08/03/2025	17:34:37	16.51
08/03/2025	17:34:47	16.51
08/03/2025	17:35:21	16.51
08/03/2025	17:35:52	16.51
08/03/2025	23:59:28	12.77
09/03/2025	00:00:00	12.77
09/03/2025	23:59:38	14.8
10/03/2025	00:00:09	14.8
10/03/2025	23:59:40	20.09
11/03/2025	00:00:11	20.09
11/03/2025	23:59:46	13.65
12/03/2025	00:00:17	13.65
12/03/2025	18:30:47	13.65
12/03/2025	18:31:18	13.65
12/03/2025	18:31:50	13.65
12/03/2025	18:32:24	13.65
12/03/2025	18:32:56	13.65
12/03/2025	18:33:28	13.65
12/03/2025	18:34:06	13.65
12/03/2025	18:37:54	13.65
13/03/2025	09:18:47	21.13
13/03/2025	23:20:46	22.07
15/03/2025	09:38:53	23.06
15/03/2025	10:16:27	24.27
15/03/2025	10:17:01	25.1
15/03/2025	10:17:32	25.1
15/03/2025	10:17:43	24.82
15/03/2025	10:18:17	23.99
15/03/2025	10:20:20	24.27
15/03/2025	10:20:51	24.27
15/03/2025	12:05:59	18.55
16/03/2025	11:02:03	18.55
16/03/2025	13:08:44	17.67

17/03/2025	09:06:15	17.01
17/03/2025	23:59:47	13.15
18/03/2025	00:00:19	13.15
18/03/2025	23:59:43	13.37
19/03/2025	00:00:14	13.37
19/03/2025	23:59:31	13.15
20/03/2025	00:00:03	13.21
20/03/2025	10:23:33	17.23
20/03/2025	10:24:05	17.23
20/03/2025	10:24:36	17.23
20/03/2025	10:25:08	17.23
20/03/2025	11:25:27	21.13
21/03/2025	12:54:10	23.77
21/03/2025	23:59:57	14.8
22/03/2025	00:00:29	14.8
22/03/2025	23:59:28	12.77
23/03/2025	00:00:00	12.77
23/03/2025	23:59:35	15.41
24/03/2025	00:00:07	15.19
24/03/2025	23:59:48	12.55
25/03/2025	00:00:19	12.55
25/03/2025	23:59:48	14.8
26/03/2025	00:00:19	14.8
26/03/2025	21:05:20	23.99
26/03/2025	21:05:51	23.55
26/03/2025	21:06:23	23.28
26/03/2025	21:06:57	23.06
26/03/2025	21:07:29	23.28
26/03/2025	23:59:41	19.87
27/03/2025	00:00:12	19.87
27/03/2025	23:59:52	12.33
28/03/2025	00:00:23	12.33
28/03/2025	17:33:50	15.41
29/03/2025	07:55:59	16.02
29/03/2025	23:59:39	15.02
30/03/2025	00:00:11	15.02
30/03/2025	23:59:29	20.31
31/03/2025	00:00:01	20.53
31/03/2025	23:59:47	13.37
01/04/2025	00:00:18	13.37
01/04/2025	08:20:15	17.23
01/04/2025	08:20:47	17.23
01/04/2025	08:21:18	17.23
01/04/2025	08:21:53	17.23

01/04/2025	08:22:24	17.23
01/04/2025	23:59:47	21.52
02/04/2025	00:00:18	21.79
02/04/2025	23:59:59	12.77
03/04/2025	00:00:31	12.99
03/04/2025	23:59:35	14.58
04/04/2025	00:00:07	14.58
04/04/2025	09:25:28	24.82

Figura 3. 5 Datos recolectados del sensor de temperatura



3.1.3. Resultado sobre el desarrollo del código fuente para automatizar el riego por goteo empleando la tecnología del Arduino.

Como parte de la implementación, se desarrolló un código en la plataforma Arduino para recopilar y procesar los datos de los sensores y el actuador, almacenándolos posteriormente en una tarjeta Micro SD. Gracias a este código, el sistema opera de manera autónoma, asegurando que las zanahorias reciban siempre la cantidad de agua necesaria para su crecimiento.

Figura 3. 6 Código fuente de la automatización del sistema de riego por goteo

```
// ----- Definiciones -----
// Sensor de humedad
#define pinHumedad1 A3 // Pin analógico para el sensor de humedad 1
#define pinHumedad2 A4 // Pin analógico para el sensor de humedad 2
const int valorAire1 = 1000;
const int valorAgua1 = 100;
const int valorAire2 = 1000;
const int valorAgua2 = 100;
const int limiteMinimoHumedad = 20;
const int limiteMaximoHumedad = 25;
int humedad1, humedad2; ///PONER SI HUMEDAD <= 0% == 0%
int porcentajeHumedad1, porcentajeHumedad2, promedioPorcentajeHumedad;

// Sensor ultrasónico
#define trigPin 6 // Pin Trigger
#define echoPin 7 // Pin Echo
const float velocidadSonido = 0.000343; // Velocidad del sonido en
343m/s = 0.000343 m/us
float distancia;
float nivelAgua;
float volumenReservorio;
const float nivelTotalPozo = 1.5; //m
const float nivelCriticoAgua = 0.6; //m
const float AreabaseMinima = 8.; //m2
const float AreaBaseMaxima = 24.; //m2

// Relés
#define RELE_ON 0
#define RELE_OFF 1
#define rele_2 31
int estadoValvula = 0;

// Sensor de flujo de agua
#define pinFlujoAgua 3
volatile int conteoPulsos = 0; // Número de pulsos
float frecuencia, flujoLitrosMinuto;
const float factorConversion = 0.2; // Pulsos por litro (según sensor)
unsigned long tiempoAnterior = 0;

// Sensor de presión
#define pinPresion1 A1 // Pin analógico para el sensor de presión
1
#define pinPresion2 A2 // Pin analógico para el sensor de presión 2
const float voltajeMin = 0.5; // Voltaje a presión mínima (0 MPa)
const float voltajeMax = 4.5; // Voltaje a presión máxima (5.0 MPa)
const float presionMax = 5.0; // Presión máxima en MPa
float presion1=0.0, presion2=0.0;
int estadoFiltro;

//Sensor de humedad
#include <DHT.h>
#define DHTPIN 5 // Pin al que está conectado el sensor
#define DHTTYPE DHT11 // Tipo de sensor (DHT11)
DHT dht(DHTPIN, DHTTYPE);
```

```

float humedadAmbiente, temperatura;

//SD
#include <SPI.h>
#include <SD.h>
#define chipSelect 4 //CS
#define pinBoton 8 // Pin donde conecta el botón para expulsar SD
#define pinLed 9 // Pin donde conecta el LED
#define pinLed2 2 // Pin donde conecta el LED
File archivo;

//Ethernet Shield
#include <Ethernet.h>
#include <TimeLib.h> // Biblioteca Time para manejar la conversión del
tiempo
String fecha = ""; // Variable para la fecha
String hora = ""; // Variable para la hora

// Configuración de Ethernet
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Dirección MAC
única
IPAddress ip(192, 168, 1, 102); // Cambia según tu red

//BLYNK
#define BLYNK_TEMPLATE_ID "TMPL2-DNrZ08"
#define BLYNK_TEMPLATE_NAME "Riego Automatizado"
#define BLYNK_AUTH_TOKEN "W_TWklq_CwNmifBnrQUkHLGR7OUSbWoR"
#define BLYNK_PRINT Serial
#include <BlynkSimpleEthernet.h>
#include <WidgetRTC.h>
WidgetRTC rtc; //Para capturar fecha y hora de servidor Blynk

// Variable para contar iteraciones del bucle principal
int contadorEnvioBlynk= 38; //38 Para que envíe a Blynk pasado 1 minuto
la primera vez que se enciende, luego envía cada 20 min

//REINICIAR ARDUINO
#include <EEPROM.h> // Librería para manejar la EEPROM
#include <avr/wdt.h> // Librería para el Watchdog Timer
#define ADDR_CONTADOR 0 // Dirección en EEPROM para almacenar el
contador

void reinicioForzoso() {
  Serial.println("Reiniciando...");
  int contadorReinicioForzoso = EEPROM.read(ADDR_CONTADOR);
  if (contadorReinicioForzoso >= 12) {
    EEPROM.write(ADDR_CONTADOR, 0); // Reiniciar el contador
    while (true) {
      delay(1000);
    }
  }
  // Incrementar el contador
  contadorReinicioForzoso++;
}

```

```

EEPROM.update(ADDR_CONTADOR, contadorReinicioForzoso); // Guardar en
EEPROM

// Forzar el reinicio
delay(20000); // Esperar 20 segundos antes del reinicio
wdt_enable(WDTO_15MS); // Activar el watchdog con un tiempo de
espera corto
while (true){ // Bucle infinito para forzar el reinicio
}

// ----- Configuración -----
void setup() {
// Comunicación serial
Serial.begin(9600);
Serial.println("Sistema iniciado.");
delay(1000);
// Inicializar SD
int contadorReinicioSD=0;
while (!inicializarSD()) {
if(contadorReinicioSD>=12){
reinicioForzoso();
}
delay(9000);
reinicializarModuloSD();
contadorReinicioSD++;
}
Serial.println("SD iniciado.");

delay(60000); //delay de 1 minuto, esperando que el wifi se conecte
correctamente ante un apagón
//Shield Ethernet
// Inicializar Ethernet
if (Ethernet.begin(mac) == 0) {
Serial.println("Error al iniciar Ethernet. Verifique la conexión.");
Ethernet.begin(mac, ip);
if (Ethernet.linkStatus() == LinkOFF) {
Serial.println("El cable Ethernet no está conectado.");
reinicioForzoso();
}
}
Serial.println("Ethernet iniciado.");

// Conexión a Blynk
delay(1000);
Serial.println("Conectando a Blynk...");
Blynk.begin(BLYNK_AUTH_TOKEN);
Serial.println("Conectado a Blynk!");
// Sincronizar RTC de Blynk
delay(1000);
rtc.begin();

// Inicializar el sensor DHT11
delay(1000);
dht.begin();

```

```

// Configuración sensores de humedad
pinMode(pinHumedad1, INPUT);
pinMode(pinHumedad2, INPUT);

// Configuración sensor ultrasónico
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

// Configuración relés
pinMode(rele_2, OUTPUT);
digitalWrite(rele_2, HIGH); // Iniciar relé apagado (relé activado por
LOW)

// Configuración sensor de flujo de agua
pinMode(pinFlujoAgua, INPUT_PULLUP);
// Se configura una interrupción para registrar cada pulso
(contarPulsos) generado por el sensor (pin 3)
attachInterrupt(digitalPinToInterrupt(pinFlujoAgua), contarPulsos,
RISING);

// Configuración sensores de presion
pinMode(pinPresion1, INPUT);
pinMode(pinPresion2, INPUT);

//Para expulsar SD
pinMode(pinBoton, INPUT_PULLUP); // Configura el botón con resistencia
pull-up
pinMode(pinLed, OUTPUT); // Configura el LED como salida
digitalWrite(pinLed, LOW); // LED apagado inicialmente
pinMode(pinLed2, OUTPUT);
digitalWrite(pinLed2, LOW);

//Apagar Rele
digitalWrite(rele_2, HIGH); // Apaga el relé al establecer la conexión
}

// ----- Funciones -----
// Sensor humedad suelo
void controlarHumedad() {
  humedad1 = analogRead(pinHumedad1);
  humedad2 = analogRead(pinHumedad2);
  porcentajeHumedad1 = map(humedad1, valorAire1, valorAgua1, 0, 100);
  porcentajeHumedad1 = (0.4072*porcentajeHumedad1)-0.3306;
  porcentajeHumedad2 = map(humedad2, valorAire2, valorAgua2, 0, 100);
  porcentajeHumedad2 = (0.4072*porcentajeHumedad2)-0.3306;
  promedioPorcentajeHumedad = (porcentajeHumedad1 + porcentajeHumedad2) /
2;

  Serial.print("Humedad 1: ");
  Serial.print(humedad1);
  Serial.print(", Humedad 2: ");
  Serial.print(humedad2);
  Serial.print(", Humedad 1: ");

```

```

Serial.print (porcentajeHumedad1);
Serial.print ("%, Humedad 2: ");
Serial.print (porcentajeHumedad2);
Serial.print ("%, Promedio: ");
Serial.print (promedioPorcentajeHumedad);
Serial.println("%");

}

//Sensor ultrasónico
void medirDistancia() {
    long duracion;

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duracion = pulseIn(echoPin, HIGH);
    distancia = (duracion * velocidadSonido) / (2);
    nivelAgua = nivelTotalPozo - (distancia-0.36); //total altura del pozo
- lectura sensor-altura extra
    nivelAgua = (1.0209*nivelAgua)+0.0072;

    volumenReservorio = (nivelAgua / 3) *
                        (AreabaseMinima +
                        AreaBaseMaxima * pow((nivelAgua / nivelTotalPozo),
2) +
                        sqrt(AreabaseMinima * AreaBaseMaxima *
pow((nivelAgua / nivelTotalPozo), 2))));

    Serial.print ("NivelAgua: ");
    Serial.print (nivelAgua);
    Serial.println(" m");
    Serial.print ("VolumenReservorio: ");
    Serial.print (volumenReservorio);
    Serial.println(" m3");

}

// Funciones de control de relés
int tiempoValvulaAbierta = 0;
int formaActivacion = 0; //0=automatico, 1>manual
void controlarReles() {
    // Verificar nivel crítico de agua, aplica a ambos modos
    if (nivelAgua <= nivelCriticoAgua) {
        cerrarValvula();
        return;
    }

//Verificar que no esté abierto por 2 min seguidos
    if(estadoValvula==2){

```

```

    cerrarValvula();
    Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
    Blynk.virtualWrite(V3, 0);
    delay(600000); //600000 = 10 min
    reiniciarSistema();
    return;
}

//Modo automático
if(formaActivacion==0){
    if (promedioPorcentajeHumedad >= limiteMinimoHumedad &&
promedioPorcentajeHumedad <= limiteMaximoHumedad) {
        if(estadoValvula==0){ //Si el estado anterior era 0, significa que
se acaba de activar. Solamente envia info a Blynk cuando cambia de
estado, si estuvo en el mismo estado mucho tiempo, no envia nada.
            Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
            Blynk.virtualWrite(V3, 1);
        }
        abrirValvula();

    } else {
        if(estadoValvula==1){ //Si el estado anterior era 1, significa que
se acaba de desactivarse, enviar a Blynk.
            Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
            Blynk.virtualWrite(V3, 0);
        }
        cerrarValvula();
    }
}

//Modo manual
else if (formaActivacion == 1){
    abrirValvula();
}
}

void abrirValvula() {
    digitalWrite(rele_2, RELE_ON); //activarRele
    estadoValvula = 1;
    tiempoValvulaAbierta++;

    // Verificar si la válvula lleva demasiado tiempo abierta
    if (tiempoValvulaAbierta >= 16) { //16 = 8 minutos ->
480'000segundos/30'000 (segundos loop). En caso la valvula se abra por
más de 8 minutos seguidos, significa que hubo un problema.
        estadoValvula = 2; //Valvula estuvo abierta por mucho tiempo
(anomalía)
    }
}

void cerrarValvula() {
    digitalWrite(rele_2, RELE_OFF); //desactivarRele
    estadoValvula = 0;
}

```

```

    tiempoValvulaAbierta = 0;
}

//Sensor flujo de agua
void medirFlujoAgua() {
    unsigned long tiempoActual = millis();
    // Calcular flujo cada segundo
    if (tiempoActual - tiempoAnterior >= 1000) {
        tiempoAnterior = tiempoActual;

        // Calcular la frecuencia
        noInterrupts(); // Detén las interrupciones temporalmente para leer
el conteo
        frecuencia = conteoPulsos;
        conteoPulsos = 0; // Reiniciar conteo
        interrupts(); // Habilita las interrupciones nuevamente

        // Convertir frecuencia a caudal (L/min) - datasheet
        flujoLitrosMinuto = (frecuencia / factorConversion);
        flujoLitrosMinuto = (1.0335*flujoLitrosMinuto)-16.488;

        if(flujoLitrosMinuto<0){
            flujoLitrosMinuto=0.0;
        }
        // Mostrar datos
        Serial.print("Frecuencia: ");
        Serial.print(frecuencia);
        Serial.print(" Hz\tFlujo: ");
        Serial.print(flujoLitrosMinuto);
        Serial.println(" L/min ");
    }
}

// Función de interrupción para contar pulsos
void contarPulsos() {
    conteoPulsos++;
}

// Sensor de presión
void medirPresion() {
    // Sensor 1
    int valorAnalogico1 = analogRead(pinPresion1);
    float voltaje1 = (valorAnalogico1 * 5.0) / 1023.0; // Conversión de ADC
a voltaje
    presion1 = ((voltaje1 - voltajeMin) / (voltajeMax -
voltajeMin))*presionMax*1000.; //KPa
    presion1 = (0.0444*presion1)-0.5650;

    // Sensor 2
    int valorAnalogico2 = analogRead(pinPresion2);
    float voltaje2 = (valorAnalogico2 * 5.0) / 1023.0; // Conversión de ADC
a voltaje
    voltaje2=voltaje2+0.065; //Calibración con voltaje1
}

```

```

    presion2 = ((voltaje2 - voltajeMin) / (voltajeMax -
voltajeMin))*presionMax*1000.; //KPa
    presion2= (0.1211*presion2)-3.6842;
    if (presion1<presion2){
        presion2=presion1;
    }

    Serial.print("Presión voltaje 1: ");
    Serial.println(voltaje1);
    Serial.print("Presión voltaje 2: ");
    Serial.println(voltaje2);

    Serial.print("Presión Sensor 1: ");
    Serial.print(presion1);
    Serial.println(" KPa");

    Serial.print("Presión Sensor 2: ");
    Serial.print(presion2);
    Serial.println(" KPa");

    float diferenciaPresion = presion1-presion2;
    if(diferenciaPresion<=(presion1/2)){ //si la diferencia es mayor al 50%
de la entrada
        estadoFiltro=1; //FILTRO OK
    }else{
        estadoFiltro=0; //FILTRO SUCIO
    }

}

// Sensor DHT11
void leerSensorDHT() {
    // Leer humedad ambiental
    humedadAmbiente = dht.readHumidity();
    // Leer temperatura en Celsius
    temperatura = dht.readTemperature();
    temperatura = (0.5503*temperatura)+7.8711;

    // Verificar si hubo error en la lectura
    if (isnan(humedadAmbiente)) {
        humedadAmbiente=-1.0;
    }
    if (isnan(temperatura)) {
        temperatura=-1.0;
    }
    // Mostrar valores en el monitor serial
    Serial.print("Humedad Ambiental: ");
    Serial.print(humedadAmbiente);
    Serial.print(" %\t");

    Serial.print("Temperatura Ambiental: ");
    Serial.print(temperatura);

```

```

    Serial.println(" °C");
}

// Función para inicializar SD y abrir archivo
bool inicializarSD() {
    if (!SD.begin(chipSelect)) {
        Serial.println("Error al inicializar la tarjeta SD.");
        return false;
    }

    archivo = SD.open("DATA.txt", FILE_WRITE);
    if (!archivo) {
        Serial.println("Error al abrir el archivo.");
        return false;
    }

    Serial.println("Tarjeta SD y archivo inicializados correctamente.");

    // Enviar al Pin Virtual
    Blynk.virtualWrite(V9, 1);
    return true;
}

void reinicializarModuloSD() {
    digitalWrite(chipSelect, HIGH); // Desactiva el CS
    delay(500); // Espera para estabilizar
    digitalWrite(chipSelect, LOW); // Activa el CS
    delay(500); // Espera para que el módulo se
estabilice
}

// Función para escribir datos en SD
void escribirEnSD(const String& fecha, const String& hora, float
nivelAgua, float volumenReservorio, float frecuencia, float
flujoLitrosMinuto, int porcentajeHumedad1, int porcentajeHumedad2, int
promedioPorcentajeHumedad, float presion1, float presion2, int
estadoFiltro, float humedadAmbiente, float temperatura, int
estadoValvula) { //&: referencia directa, sin copias, mayor eficiencia,
*:puntero, permite almacenar cadenas de texto y no 1 letra. //int
estadoBomba
    if (archivo) {
        archivo.print(fecha); // Escribe la fecha y hora actual
        archivo.print(",");
        archivo.print(hora); // Escribe la fecha y hora actual
        archivo.print(",");
        archivo.print(nivelAgua);
        archivo.print(",");
        archivo.print(volumenReservorio);
        archivo.print(",");
        archivo.print(frecuencia);
        archivo.print(",");
        archivo.print(flujoLitrosMinuto);
        archivo.print(",");
        archivo.print(porcentajeHumedad1);
    }
}

```

```

archivo.print(",");
archivo.print(porcentajeHumedad2);
archivo.print(",");
archivo.print(promedioPorcentajeHumedad);
archivo.print(",");
archivo.print(presion1);
archivo.print(",");
archivo.print(presion2);
archivo.print(",");
archivo.print(estadosFiltro);
archivo.print(",");
archivo.print(humedadAmbiente);
archivo.print(",");
archivo.print(temperatura);
archivo.print(",");
archivo.println(estadosValvula);
archivo.flush(); // Asegura que los datos se escriban inmediatamente

digitalWrite(pinLed2, HIGH); // Encender el LED
delay(1000);
digitalWrite(pinLed2, LOW);
} else {
  Serial.println("No se pudo escribir en el archivo.");
  Blynk.virtualWrite(V9, 0);
}
}

//Función expulsar SD
void expulsarSD(){
  // Si el botón se presiona, cerrar el archivo y la SD
  if (digitalRead(pinBoton) == LOW) { // Botón presionado
    delay(1000);
    digitalWrite(pinLed, HIGH); // Encender el LED
    archivo.close();
    Serial.println("Archivo cerrado. Puede extraer la SD.");
    delay(3000);
    digitalWrite(pinLed, LOW);

    // Enviar al Pin Virtual
    Blynk.virtualWrite(V9, 0); //como la función se llama explícitamente
    en el loop, entonces el valor es enviado a Blynk

    // Detener el programa para evitar escribir más
    while (true) {
      delay(1000);
    }
  }
}

// Función para sincronizar el RTC cuando Blynk se conecta
BLYNK_CONNECTED() {
  rtc.begin();
  Blynk.syncAll(); // Sincroniza los datos del servidor
  Serial.println("RTC sincronizado con el servidor de Blynk.");
}

```

```

}

// Actualizar Fecha y Hora desde RTC (Blynk)
void actualizarFechaHora() {
    time_t tiempoActual = now(); // Ajuste a UTC-5 - (5 * 3600)

    // Construir la fecha
    fecha = String(year(tiempoActual)) + "-" +
             String(month(tiempoActual) < 10 ? "0" : "") +
String(month(tiempoActual)) + "-" +
             String(day(tiempoActual) < 10 ? "0" : "") +
String(day(tiempoActual));

    // Construir la hora
    hora = String(hour(tiempoActual) < 10 ? "0" : "") +
String(hour(tiempoActual)) + ":" +
             String(minute(tiempoActual) < 10 ? "0" : "") +
String(minute(tiempoActual)) + ":" +
             String(second(tiempoActual) < 10 ? "0" : "") +
String(second(tiempoActual));
}

void reiniciarSistema(){
    //Expulsar SD
    delay(1000);
    digitalWrite(pinLed, HIGH); // Encender el LED
    archivo.close();
    Serial.println("Archivo cerrado. Puede extraer la SD.");
    delay(3000);
    digitalWrite(pinLed, LOW);
    Blynk.virtualWrite(V9, 0); //como la función se llama explícitamente
en el loop, entonces el valor es enviado a Blynk
    //REINICIAR
    Serial.println("Reiniciando Arduino...");
    delay(1000); // Esperar 1 segundo antes del reinicio
    wdt_enable(WDTO_15MS); // Activar el watchdog con un tiempo de
espera corto
    while (true) {} // Bucle infinito para forzar el reinicio
}

//REINICIAR ARDUINO DESDE BLINK
BLYNK_WRITE(V2) {
    int botonReinicio = param.asInt(); // Obtiene el valor desde el widget
(0 o 1)
    if (botonReinicio == 1) { // Verifica si el botón virtual está
presionado
        reiniciarSistema();
    }
}

//Activar o desactivar valvula de forma manual
BLYNK_WRITE(V13) {
    int botonValvulaManual = param.asInt(); // Obtiene el valor desde el
widget (0 o 1)

```

```

    tiempoValvulaAbierta = 0; // Reiniciar contador al activar o desactivar
modo manual
    if (botonValvulaManual == 1) {          // Modo Manual
        formaActivacion=1;
    }else{
        formaActivacion=0;
    }
}

// ----- Bucle principal -----
void loop() {
    expulsarSD();
    controlarHumedad();          // Leer y procesar humedad
    medirDistancia();           // Leer y procesar distancia
    controlarReles();           // Controlar los relés
    medirFlujoAgua();           // Leer y procesar flujo de agua
    medirPresion();             // Leer y procesar presión de agua3
    leerSensorDHT();            // Leer y procesar DHT11
    actualizarFechaHora();
    escribirEnSD( fecha, hora, nivelAgua, volumenReservorio, frecuencia,
    flujoLitrosMinuto, porcentajeHumedad1, porcentajeHumedad2,
    promedioPorcentajeHumedad, presion1, presion2,
    estadoFiltro, humedadAmbiente, temperatura, estadoValvula); //estadoBomba,

    contadorEnvioBlynk++;
    if(contadorEnvioBlynk >=40){ //40, cada 20 minutos enviar a la app
Blynk
        Blynk.virtualWrite (V0, promedioPorcentajeHumedad);
        Blynk.virtualWrite (V1, nivelAgua);
        Blynk.virtualWrite (V11, volumenReservorio);
        Blynk.virtualWrite (V3, estadoValvula);
        Blynk.virtualWrite (V4, flujoLitrosMinuto);
        Blynk.virtualWrite (V5, presion1);
        Blynk.virtualWrite (V6, presion2);
        Blynk.virtualWrite (V12, estadoFiltro);
        Blynk.virtualWrite (V7, humedadAmbiente);
        Blynk.virtualWrite (V8, temperatura);
        Blynk.virtualWrite (V9, 1);
        String fechaHora = fecha + " " + hora;
        Blynk.virtualWrite (V10, fechaHora);
        contadorEnvioBlynk=0;
    }

    Blynk.run();

    delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1
minuto)
    Serial.println("-----");
}

```

3.1.4. Resultado de estimación de la cantidad de zanahorias que se cosecharon al finalizar la campaña agrícola.

Al concluir la campaña agrícola, se realizó la estimación de la producción total de zanahorias, tomando como referencia el peso promedio de las raíces comerciales por planta. El rendimiento estimado fue de 7 kg/m², considerando una superficie cultivada de 35 m². De esta manera, se obtuvo una producción total estimada de 245 kilogramos de zanahorias.

DISCUSIONES

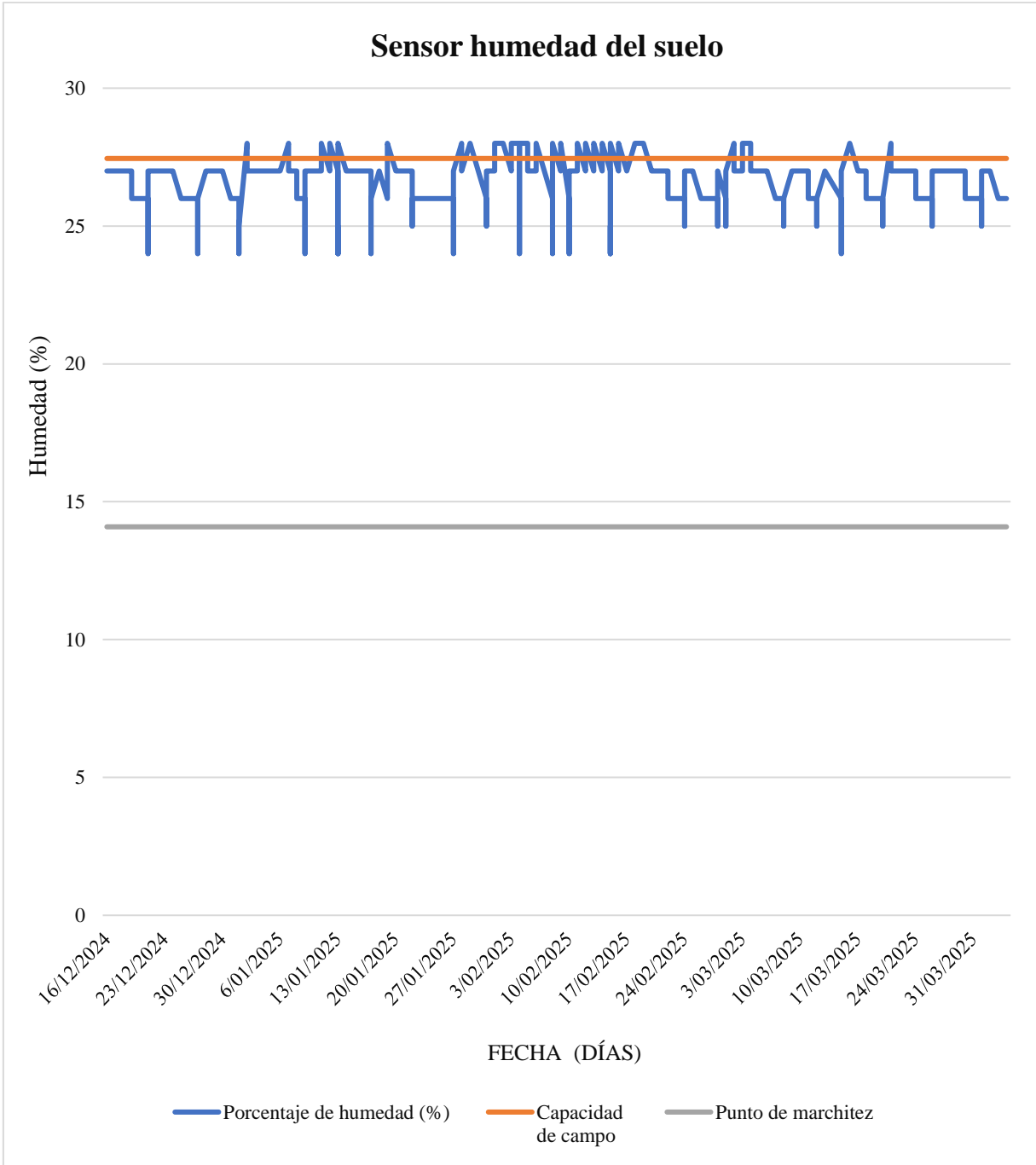
a. La implementación del riego por goteo automatizado con Arduino en la parcela de hortalizas en Tambo, La Mar, Ayacucho, optimizó el uso del agua y mejoró la producción. Se instalaron sensores de humedad que enviaban datos en tiempo real al microcontrolador, activando la electroválvula cuando el suelo lo requería. Durante el monitoreo, surgieron desafíos como la calibración de sensores y ajustes en la programación para evitar activaciones innecesarias. Tras pruebas y modificaciones, el sistema funcionó de manera eficiente, reduciendo el desperdicio de agua y asegurando una irrigación adecuada.

1. Tomando como base los resultados obtenidos se acepta la hipótesis específica 1 en base a los resultados obtenidos, de que los componentes de la automatización para el sistema de riego por goteo fueron de fácil instalación, el cual permitieron mejorar la eficiencia del invernadero de Pampa Hermosa-Lorensayoq, debido a que guarda una relación alta con lo que menciona Ludeña (2021) en su trabajo de investigación titulado *"Implementación y monitoreo con tecnología Arduino para automatización de riego tecnificado"*, se evidenció que la adecuada selección de sensores y actuadores se integró exitosamente, logrando una programación eficiente para el sistema.

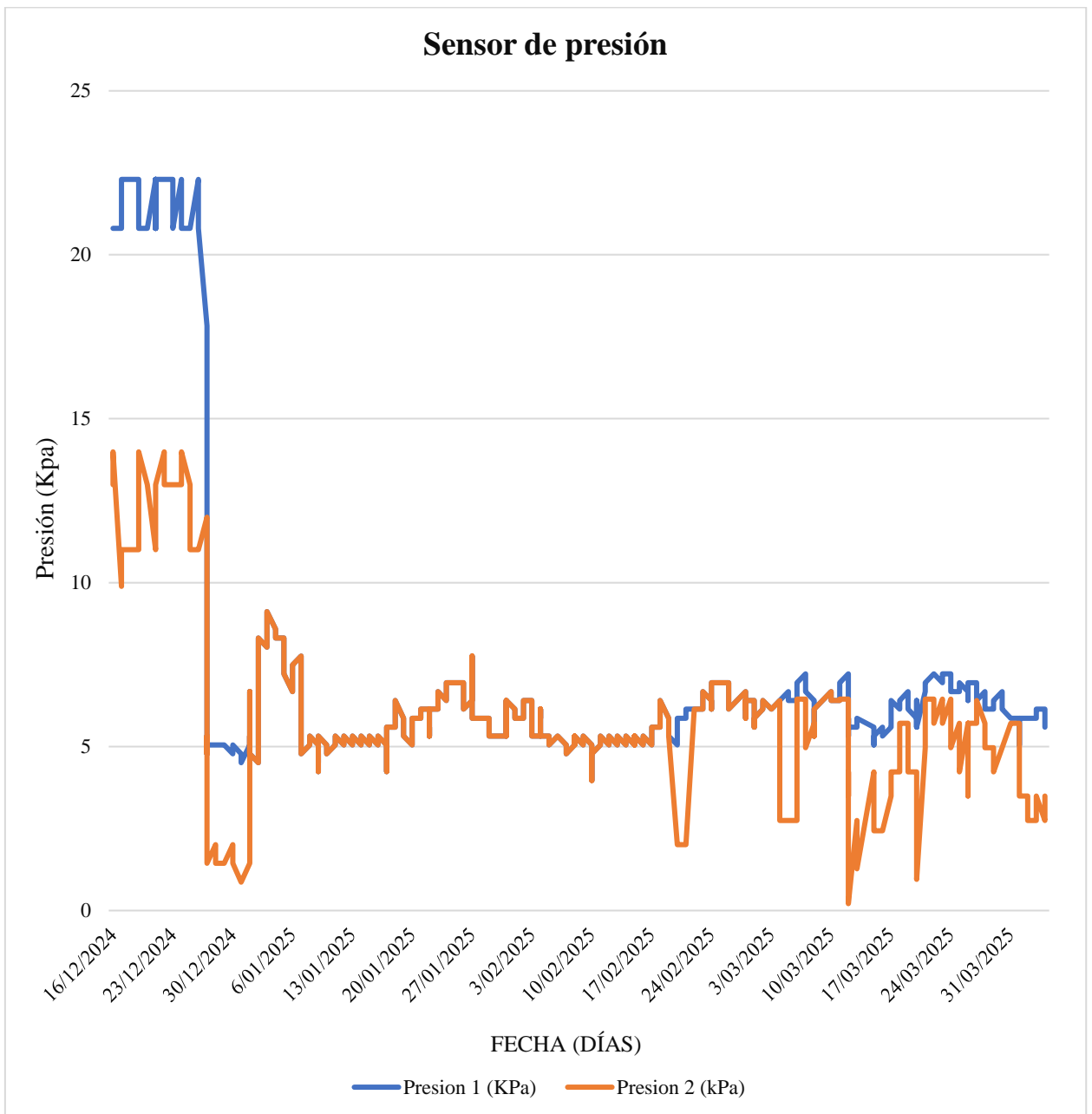
2. A partir de los resultados obtenidos aceptamos la hipótesis específica 2 que establece que los sensores transmitieron la información al microcontrolador Arduino de manera rápida, lo que facilitó una comunicación efectiva permitiendo detectar señales digitales y convertirlas en magnitudes físicas.

a) El gráfico muestra el registro del porcentaje de humedad del suelo desde el 16 de diciembre de 2024 hasta el 31 de marzo de 2025. Durante este periodo, la humedad se mantuvo entre 24% y 28%, valores que se mantuvieron por encima del punto de marchitez permanente y cercanos a la capacidad de campo. Esto nos indica que el suelo permaneció en condiciones adecuadas para el desarrollo del cultivo.

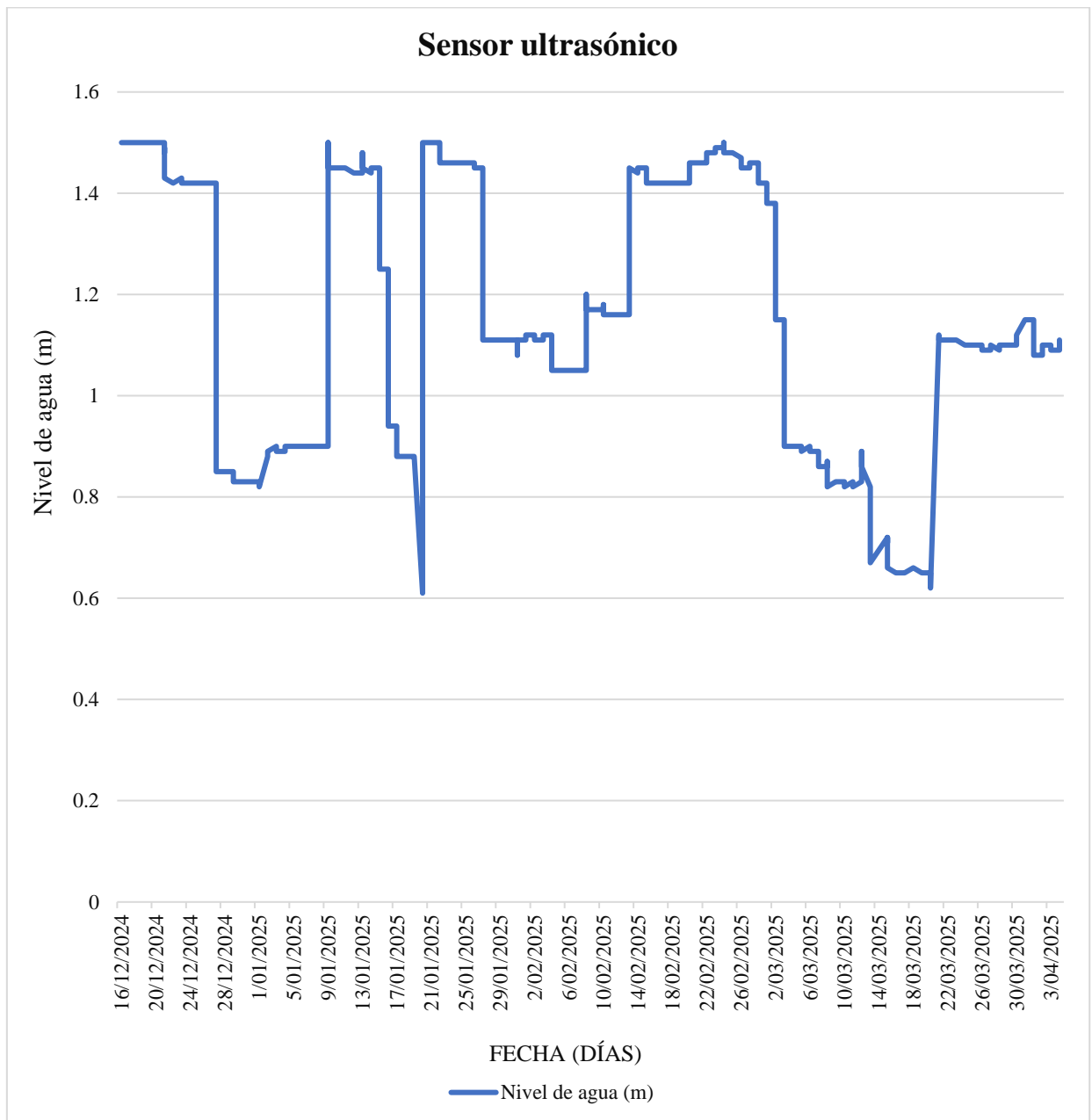
El comportamiento observado refleja un manejo eficiente del riego, favorecido por el sistema automatizado implementado.



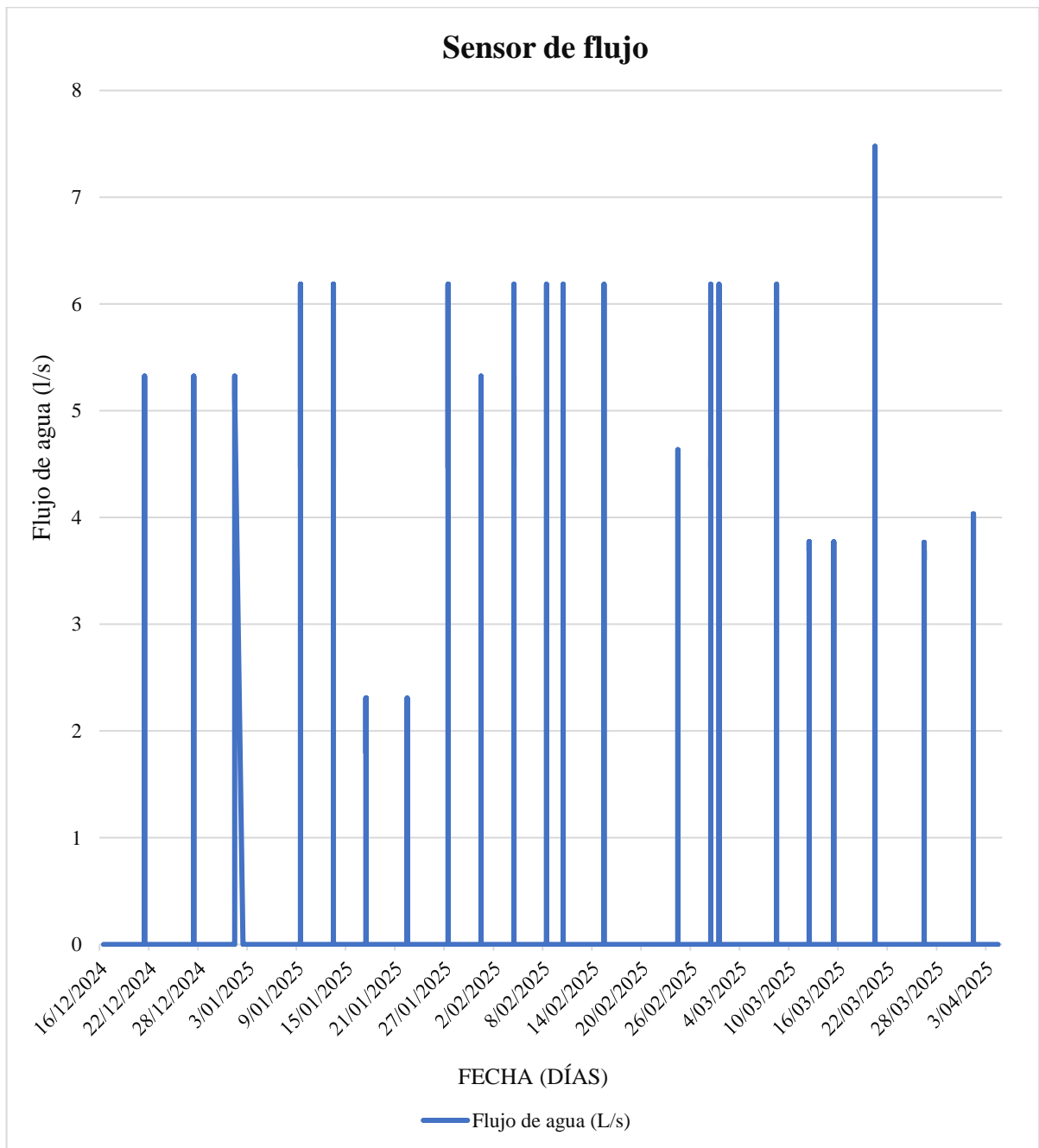
b) El gráfico nos indica los valores registrados durante el monitoreo de los sensores de presión instalados antes y después del filtro. Un diferencial de presión reducido indicó un filtro limpio, mientras que un aumento progresivo evidenció obstrucción por acumulación de partículas. Esta variación permitió programar mantenimientos oportunos, evitando caídas de presión en el sistema y posibles daños en los emisores. De este modo, se aseguró una entrega uniforme de agua al cultivo, lo cual es fundamental para mantener la eficiencia del riego.



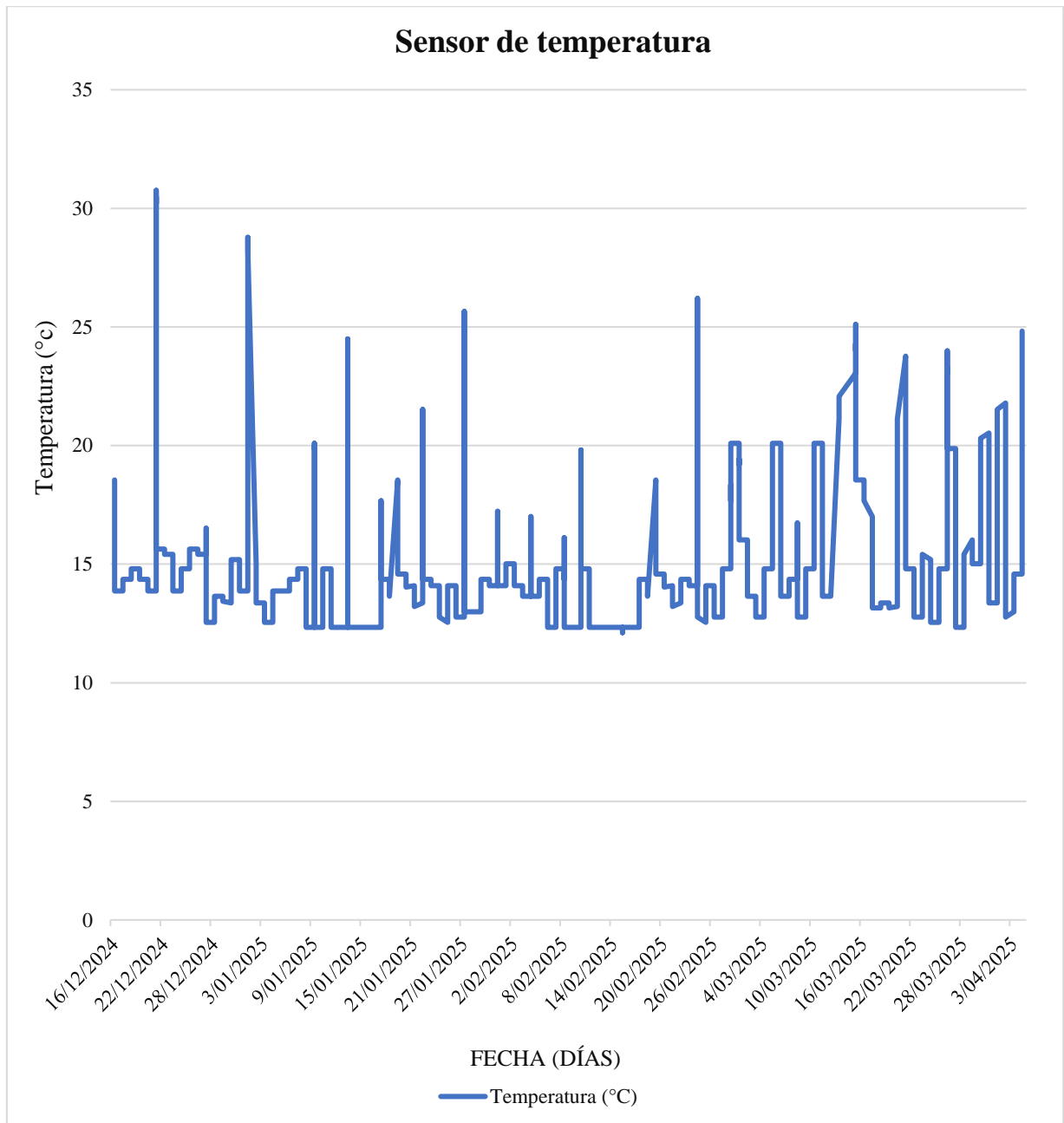
- c) El grafico nos indica que el sensor ultrasónico registró variaciones significativas en el nivel de agua. Se detectaron descensos abruptos en enero y marzo de 2025, lo que indica eventos de vaciado o disminución en la disponibilidad del recurso, sin que el nivel descendiera por debajo de 0.6 m. Además, se observaron incrementos del nivel de agua, posiblemente por las lluvias o llenados programados. Estos resultados permiten evaluar el comportamiento dinámico del sistema y confirman la capacidad del sensor para detectar cambios con precisión en tiempo real.



d) La gráfica del caudal registrado mediante el sensor de flujo muestra picos que corresponden a eventos de riego, mientras que los valores en 0 indican ausencia de flujo en el sistema. El caudal aplicado tuvo con un valor máximo de aproximadamente 7 L/s y un promedio de 4.5 L/s. El sensor permitió registrar de forma continua y precisa el caudal durante el funcionamiento del sistema, lo que facilitó la verificación de los eventos de riego y la identificación de posibles anomalías. Esta información contribuye a mejorar la eficiencia del sistema de riego.



- e) El análisis del gráfico térmico muestra que la temperatura se mantuvo mayormente entre 13 °C y 18 °C, con algunos picos por encima de los 25 °C. Estos aumentos aceleraron la pérdida de humedad en el suelo, lo que activó el sistema de riego automático al detectar niveles bajos. Este comportamiento confirma que la temperatura ambiental influye directamente en la evaporación del agua y, por tanto, en el funcionamiento del sistema de riego.



3. Con base en los resultados obtenidos, se acepta la hipótesis específica 3, la cual establece que el código fuente, desarrollado en Arduino IDE, se integró con éxito al microcontrolador Arduino. Esto permitió el correcto funcionamiento del sistema de automatización del riego, incluyendo el monitoreo eficiente del sistema. Por otra parte, estos resultados guardan una relación con lo expuesto por Huamán (2023) en su tesis titulada *"Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima"*, donde se evidencia que la programación del código en el IDE de Arduino, mediante el uso de librerías, funciones y variables, permitió el funcionamiento de la automatización.

4. Con los resultados obtenidos, se acepta la hipótesis 4, que plantea que la aplicación de prácticas agrícolas sostenibles, incluida la automatización del sistema de riego junto con condiciones climáticas favorables, generó un impacto positivo en el rendimiento del cultivo de zanahoria Chantenay Red Core, alcanzando los 7 kg/m², siendo un rendimiento mayor al de Nina (2020) en su tesis: *"Evaluación de tres variedades de zanahoria (Daucus carota L.) en invernadero y a campo abierto en el municipio de la ciudad de La Paz"*, en donde se demuestra que la mayor producción en rendimiento la obtuvo el tratamiento en la zanahoria chantenay en un invernadero que fue de 6,19 kg/m².

CONCLUSIONES

a. La implementación del microcontrolador Arduino para automatizar el riego por goteo ha demostrado ser una solución eficiente para optimizar el uso del agua en la parcela de hortalizas en Tambo, La Mar. A través del monitoreo del sistema, se ha logrado garantizar que el riego se realice de manera puntual y precisa, ajustándose a las condiciones del suelo y el clima. Además, el sistema es fácil de mantener, lo que lo convierte en una opción viable para pequeños y medianos agricultores.

1. El análisis de los componentes del sistema de riego automatizado permite identificar la función específica de cada elemento en su rendimiento global. Sensores de humedad, flujo, presión, ultrasonido y temperatura, junto con el microcontrolador Arduino, la electroválvula y otros complementos, conforman una estructura integrada que asegura un riego preciso y adaptado a las condiciones del cultivo. Los sensores de humedad son clave, ya que activan el riego solo cuando el suelo lo necesita, optimizando el uso del agua y evitando excesos o carencias. La integración tecnológica mejora la operatividad del sistema y promueve prácticas agrícolas sostenibles y de alto rendimiento.

2. El sistema desarrollado permitió una comunicación eficiente y en tiempo real entre los sensores y el microcontrolador Arduino, lo que aseguró un control preciso del riego automatizado. Los niveles de humedad del suelo se mantuvieron dentro del rango óptimo (24 %–28 %) gracias al monitoreo continuo. Los sensores de presión facilitaron labores de mantenimiento preventivo al detectar oportunamente obstrucciones en el filtro, mientras que el sensor ultrasónico permitió un control adecuado del nivel de agua. La temperatura en el invernadero osciló entre los 13 y 18 °C, registrándose picos superiores a 25 °C, los cuales provocaron una disminución en la humedad del suelo. En conjunto, estos resultados evidencian la efectividad del sistema automatizado para el manejo eficiente del riego en condiciones variables dentro del invernadero.

3. Se desarrolló un código fuente funcional que automatiza un sistema de riego por goteo utilizando la plataforma Arduino, integrando múltiples sensores para una gestión eficiente y precisa. El sistema realiza lecturas del sensor ultrasónico HC-SR04 para medir el nivel de agua

en el reservorio, así como del sensor de humedad del suelo. Cuando el nivel de agua supera los 0.6 metros y la humedad del suelo se encuentra entre el 20 % y el 25 %, se activa automáticamente la electroválvula; en caso contrario, permanece desactivada. Además, se implementó la programación de sensores adicionales como el de presión HK3025, flujo YF-DN50 y la temperatura DHT11. Todos los datos recolectados por el sistema se almacenan en una tarjeta de memoria SD y se visualizan en tiempo real a través de la aplicación móvil Blynk, lo que permite un monitoreo remoto, eficiente y confiable del sistema de riego.

4. La implementación del sistema de riego automatizado con control continuo incrementó significativamente el rendimiento del cultivo de zanahoria Chantenay Red Core, alcanzando los 7 kg/m². Este resultado superó el promedio nacional de 2,52 kg/m² reportado por la Dirección General Agrícola del MIDAGRI (2020) y el rendimiento de 6,19 kg/m² registrado por Nina (2020) en condiciones de invernadero. Los datos obtenidos confirman la eficacia de esta tecnología como una opción sostenible para optimizar la productividad agrícola.

RECOMENDACIONES

1. Se recomienda emplear la tecnología Arduino para el riego, de modo que puedan controlar en tiempo real los distintos parámetros necesarios para lograr un rendimiento satisfactorio en el momento de la cosecha
2. Antes de implementar el sistema automatizado por completo, realiza pruebas en el campo para ajustarlo a las condiciones locales.
3. Se recomienda el uso fuentes de energía tales como los paneles solares para que sea amigable con el medio ambiente.
4. Se debe elegir componentes que sean duraderos y fáciles de mantener en el caso del sensor de humedad, debido que en las áreas rurales el acceso a repuestos puede ser limitado.
5. Para una comunicación eficiente, asegúrate de que las conexiones sean fiables y tengan buen alcance.
6. También es importante ofrecer la opción de control manual para ajustes en situaciones de emergencia.
7. Lleva un registro detallado del crecimiento de las zanahorias y del consumo de agua durante la campaña. Este análisis permitirá evaluar el impacto del riego automatizado y optimizar su eficiencia en futuras siembras.
8. Capacitar a los agricultores en el uso, mantenimiento y reparación de sistemas automatizados mediante talleres prácticos y asesoría técnica continua.
9. Implementar sistemas de riego automatizado en cultivos de agricultura extensiva como una estrategia clave para optimizar el uso eficiente del agua, mejorar la productividad y fomentar la sostenibilidad del sistema agrícola.

REFERENCIAS BIBLIOGRÁFICAS

- Artero, O. T. (2013). *Arduino curso práctico de formación*. México: Alfaomega.
<https://www.latecnicalf.com.ar/descargas/material/proyectointegrador/Arduino%20Curso%20Pr%C3%A1ctico%20de%20formaci%C3%B3n.pdf>
- Briones, G., & García I. (2007). *Sistema de riego por aspersión y goteo. 2a ed.* México. Trillas UAAAN.
- Carit et al. (2002). *Marcos A. Carit, Marvin Coto, Mario E. Fung. Sistemas de control aplicados a la robótica. Universidad de Costa Rica, Facultad de Ingeniería -Escuela de Ingeniería Eléctrica, Departamento de Automática (2002).*
- Castañon, G. (1991). *Manual de Riego por Aspersión*. Mérida, Venezuela: CIDIAT.
- Cepeda et al. (2010). *Invernadero Inteligente basado en un Enfoque Sustentable para la Agricultura Mexicana*.
- Cervantes, W. R. (2016). *Diseño e implementación de sistema de riego automatizado y controlado por una placa Arduino para la finca "LA LUCIA" . CALCETA.*
<http://repositorio.espam.edu.ec/handle/42000/301>
- Darshna et al. (2015). *Smart Irrigation System, vol. 10.* <https://doi.org/10.9790/2834-10323236>
- De rosas, F. (2014). *Microcontroladores: Universidad Autónoma De Baja California - Facultad de Ciencias Químicas e Ingeniería.*
- Evans et al. (2013). *Adoption of site-specific variable rate sprinkler irrigation systems. Irrigation Science, 31(4), 871-887.* <https://doi.org/10.1007/s00271-012-0365-x>
- Fuentes, J. (1999). *Técnicas de riego. Ediciones Mundi-Prensa México.*
- Gonzales, J. E. (2022). *Diseño y automatización vía remota de un sistema de riego por goteo para el cultivo de Paprika.* <https://repositorio.lamolina.edu.pe/handle/20.500.12996/5654>
- Goyal, M. (2017). *Riego por goteo. Capítulo IV sistemas de riego.*
- Hargreaves, G. H., & Merkle G.P. (2000). *Fundamentos del Riego. Estados Unidos. Water Resources Publications.*
- Huamán, L. M. (2023). *Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima.*
<https://repositorio.lamolina.edu.pe/handle/20.500.12996/5971>
- INIA. (2009). *Zanahoria INIA 101 Perú. Ministerio de Agricultura: Instituto Nacional de Innovación Agraria.*
- Jones, H. (2004). *Irrigation scheduling: Advantages and pitfalls of plant-based methods. Journal of Experimental Botany, 55(407), 2427-2436.* 55(407).
<https://doi.org/10.1093/jxb/erh213>

- Liotta et al. (2015). *Manual de capacitación Riego por goteo. 1 er ed. Argentina.*
- Ludeña, J. (2021). *Implementación y monitoreo con tecnología Arduino para automatización de riego tecnificado en el Centro Experimental Canaán UNSCH - Ayacucho.*
- Maroto, J. (2008). *Elementos de Horticultura General, 3a. Madrid: Mundi-Prensa. Mundi-Prensa.*
- Martínez, L. (2001). *Manual de operación y mantención de equipos de riego presurizado. Gobierno Regional de Atacama. Comisión Nacional de Riego e Instituto de Investigaciones Agropecuarias. Huasco, Chile: Vallenar.*
- Naylamp Mechatronics. (2021). *Fuente de alimentación Conmutada 12v 20a. Google.*
<https://naylampmechatronics.com/fuentes-switching-ac-dc/628-fuente-de-alimentacion-conmutada-acdc-250w-12v-20a-wode.html>
- Nina, D. (2020). *Evaluación de tres variedades de zanahoria (Daucus carota l.) en invernadero y a campo abierto en el municipio de la ciudad de La Paz. Universidad Mayor San Andrés: Facultad de Agronomía.*
- Osorio U., A. y T. C. , F. (1999). *Elementos de riego tecnificado [en línea]. La Serena: Disponible.*
<https://biblioteca.inia.cl/bitstream/handle/20.500.14001/28965/NR29540.pdf?sequenc>
- Palomino K. (2009). *Riego por goteo. Starbook. Editorial.*
- Torrente, O. (2013). *ARDUINO: Curso práctico de formación (Primera ed). Alfaomega Grupo Editor, S.A. De C.V., México.*

ANEXOS

Anexo A: Panel Fotográfico



Figura 1 Toma de datos del área de estudio



Figura 2 Limpieza del reservorio



Figura 3 Construcción de la caja de protección del cabezal de riego



Figura 4 Limpieza para la instalación de la caseta de control



Figura 5 Implementación de los sensores y electroválvula al sistema de riego por goteo



Figura 6 Implementación del microcontrolador Arduino y el sistema eléctrico en la caseta de control



Figura 7 Prueba de coeficiente de uniformidad



Figura 8 Siembra de la zanahoria



Figura 9 Prueba del sistema de riego por goteo automatizado



Figura 10 Calibración de los sensores



Figura 11 Inspección de mi asesor sobre la instalación de los sensores



Figura 12 Inspección de mi asesor sobre la instalación de la electroválvula



Figura 13 Inspección del Arduino y sistema eléctrico en la caseta de control



Figura 14 Explicación breve de la implementación



Figura 15 Recibiendo algunas sugerencias de mi asesor



Figura 16 Visita de campo junto a mi asesor



Figura 17 Seguimiento de las etapas de desarrollo de la zanahoria



Figura 18 Revisiones periódicas para garantizar una producción saludable y de calidad.



Figura 19 Sacando medidas para calcular rendimiento por M2



Figura 20 Cosechando el cultivo de la zanahorias

Anexo B:
Resultados de laboratorio de
análisis Físico – Químico del
agua



MULTISERVICIOS AGROLAB

INGENIEROS TRABAJANDO POR UN AGRO SOSTENIBLE

LABORATORIO DE ANÁLISIS DE SUELOS, PLANTAS, AGUAS Y FERTILIZANTES

ASESORIA Y CAPACITACIÓN EN:

- EVALUACIÓN Y MUESTREO DE SUELOS. - INTERPRETACIÓN DE RESULTADOS DEL ANÁLISIS AGRÍCOLA
- USO, MANEJO Y CONSERVACIÓN DE SUELOS - ESTUDIOS DE IMPACTO AMBIENTAL
- AGRICULTURA SUSTENTABLE

ANÁLISIS DE AGUA

1150951

SOLICITANTE: Srta. Diana Marisella Loayza Paredes

PROYECTO: Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2025

REGIÓN: Ayacucho **PROVINCIA:** La Mar **DISTRITO:** Tambo **LOCALIDAD:** Pampa Hermosa

FUENTE: Rio Acco

FECHA DE INGRESO: 15-11-24

FECHA DE RESULTADOS: 23-11-24

Nº Laboratorio	A01428
Nº Campo	Rio Acco
pH	8.34
C.E. (µS/cm)	594.00
CATIONES (meq/l)	
Calcio	3.35
Magnesio	1.58
Potasio	0.12
Sodio	0.89
Suma de Cationes (meq/l)	5.94
ANIONES (meq/l)	
Nitratos	0.62
Carbonatos	0.13
Bicarbonatos	2.79
Cloruros	0.97
Sulfatos	1.18
Suma de Aniones (meq/l)	5.69
Na (%)	14.98
Boro (ppm)	0.02
RAS	0.57
Sales Solubles Totales (ppm)	380.16
Dureza (G.H.F.)	24.64
CLASIFICACIÓN	C2-S1*

(*) Se trata de agua de salinidad media. Válida para riego. En ciertos casos, puede ser necesario emplear volúmenes de agua en exceso y utilizar cultivos tolerantes a la salinidad. Sin embargo, pueden presentarse problemas con cultivos muy sensibles al sodio. Muestra riesgo de precipitación por ende obturación de emisores, al tratarse de agua medianamente dura. Existe ligero a moderado riesgo de daño, en el uso por aspersión.

09-11-2024



Ph.D. Marhleni Cerda Gómez
Responsable de Laboratorio

VºPº

Urb. Mariscal Cáceres Mz. "G-12" - Ayacucho / 966938028 - 966631889

982781298 agrolab107@gmail.com

Anexo C:
Resultados de laboratorio de
análisis Físico y
caracterización del suelo



MULTISERVICIOS AGROLAB

INGENIEROS TRABAJANDO POR UN AGRO SOSTENIBLE

LABORATORIO DE ANÁLISIS DE SUELOS, PLANTAS, AGUAS Y FERTILIZANTES

ASISORIA Y CAPACITACIÓN EN:

- EVALUACIÓN Y MUESTREO DE SUELOS - INTERPRETACIÓN DE RESULTADOS DEL ANÁLISIS AGRÍCOLA
- USO, MANEJO Y CONSERVACIÓN DE SUELOS - ESTUDIOS DE IMPACTO AMBIENTAL
- AGRICULTURA SUSTENTABLE

ANÁLISIS FÍSICO DE SUELOS

1350964

SOLICITANTE: Srta. Diana Marisella Loayza Paredes

PROYECTO: Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2025

REGIÓN: Ayacucho PROVINCIA: La Mar DISTRITO: Tambo LOCALIDAD: Pampa Hermosa

FECHA DE INGRESO: 15-11-24

FECHA DE RESULTADOS: 23-11-24

Nº Laboratorio	AS 111983
Nº Campo	Pampa Hermosa
Capacidad de campo (%)	27.45
Punto de marchites (%)	14.09
Densidad aparente (g cc ⁻¹)	1.22
Arena (%)	41
Limo (%)	21
Arcilla (%)	38
Clase Textural	Franco Arcilloso

23-11-2024



Ph.D. Marheleni Cerda Gómez
Responsable de Laboratorio

VºBº

Urb. Mariscal Cáceres Mz. "G-12" - Ayacucho / 966938028 - 966631889
982781298 agrolab107@gmail.com



MULTISERVICIOS AGROLAB

INGENIEROS TRABAJANDO POR UN AGRO SOSTENIBLE

LABORATORIO DE ANÁLISIS DE SUELOS, PLANTAS, AGUAS Y FERTILIZANTES

ANÁLISIS DE SUELOS : CARACTERIZACIÓN

ASESORÍA Y CAPACITACIÓN EN:

- EVALUACIÓN Y MUESTREO DE SUELOS
- INTERPRETACIÓN DE RESULTADOS DEL ANÁLISIS AGRÍCOLA
- USO, MANEJO Y CONSERVACIÓN DE SUELOS.
- ESTUDIOS DE IMPACTO AMBIENTAL
- AGRICULTURA SUSTENTABLE.

Solicitante	Sra. DIANA MARISELLA LOAYZA PAREDES		Fecha	23/11/2024
Proyecto	Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas. Tambo, La Mar, Ayacucho-2025			
Departamento	Ayacucho	Provincia:	La Mar	Tambo
Localidad	Pampa Hermosa-Lorensayoc	Altitud (m.s.n.m.)	N/A	Superficie: N/A
Código de muestra	13548	Referencia de Campo:	Pampa Hermosa-Lorensayoc Cultivo anterior: Cultivo a sembrar:	

Parámetro	Resultado	Unidad	Clasificación
1	7.84	Unidad pH	Moderadamente Alcalino

2	0.71	dS m.-1	No salino
---	------	---------	-----------

3	4.94	%	Bajo
4	5.49	%	Muy Alto
5	67.10	ppm	Muy Alto
6	740	ppm	Muy Alto
7	0.27	%	Muy Alto
8	28.27	Cmol (+).kg-1	Alto

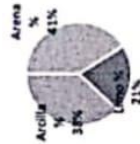
Parámetro	Resultado	Unidad	Clasificación
10	21.80	Cmol (+).kg-1	Alto
11	4.12	Cmol (+).kg-1	Alto
12	2.10	Cmol (+).kg-1	Muy Alto
13	0.24	Cmol (+).kg-1	Muy bajo
14	0.00	Cmol (+).kg-1	Muy bajo
15	100	%	Alto

Acido	Neutro	Básico
7.84		

Muy bajo	Medio	Muy alto
4.94	5.49	67.10
740	0.27	28.27

Bajo	Medio	Alto
21.80	4.12	2.10
0.24	0.00	100

Clase textural		
Arena	Limo	Arcilla
%	%	%
41	21	38
		Fr.Ar.



Parámetro	Resultado	Valoración
Ca / K	10.39	Deficiencia de Calcio
Ca / Mg	5.30	Ideal
Mg / K	1.96	Ideal
(Ca + Mg + K)/Al		

23/11/2024 Ph. D. Maribel Yérez Gómez
 Laboratorio de Laboratorio

A = arena, A.Fr = Arena franca; Fr.A. = Franco arenoso; Fr = Franco; Fr.L = Franco limoso; L = Limoso; FrArA = Franco arcillo arenoso; FrAr = Franco arcilloso; FrArL = Franco arcillo limoso; ArA = Arcillo arenoso; ArL = Arcillo limoso; Ar = Arcilloso.

Anexo D:
Resultados de laboratorio de
Humedad actual del suelo



MULTISERVICIOS AGROLAB

INGENIEROS TRABAJANDO POR UN AGRO SOSTENIBLE

LABORATORIO DE ANÁLISIS DE SUELOS, PLANTAS, AGUAS Y FERTILIZANTES

ASESORIA Y CAPACITACIÓN EN:

- EVALUACIÓN Y MUESTREO DE SUELOS - INTERPRETACIÓN DE RESULTADOS DEL ANÁLISIS AGRÍCOLA
- USO, MANEJO Y CONSERVACIÓN DE SUELOS - ESTUDIOS DE IMPACTO AMBIENTAL
- AGRICULTURA SUSTENTABLE

ANÁLISIS FÍSICO DE SUELOS

1350970

SOLICITANTE: Srta. Diana Marisella Loayza Paredes

PROYECTO: Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2025

REGIÓN: Ayacucho **PROVINCIA:** La Mar **DISTRITO:** Tambo **LOCALIDAD:** Pampa Hermosa

FECHA DE INGRESO: 21-12-24

FECHA DE RESULTADOS: 26-12-24

Nº Laboratorio	AS 111990	AS 111991	AS 111992
Nº Campo	M-01	M-02	M-03
Humedad actual (%)	16.96	17.10	12.40

26-12-2024



Ph.D. Marileni Cerda Gómez
Responsable de Laboratorio

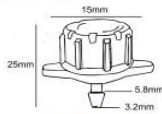
Anexo E: Ficha técnica de los emisores



Grupo
AGROTECH
Soluciones Agrícolas

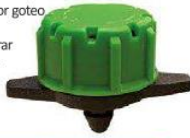
GOTERO REGULABLE

- MARCA:** PIMTAS
- MATERIAL:** Caucho Natural
- COLOR:** Verde - Negro



CARACTERÍSTICAS

Gotero regulable se utiliza en tuberías por goteo no perforadas, es fácil de instalar. las tuberías de riego por goteo sin perforar se perforan a 3mm y se fijan fácilmente. también se utilizan en tubos Spaguetti de 7mm. Son extraíbles y facilitan su limpieza.



CÓDIGO SISTEMA	CÓDIGO FABRICANTE	DIAMETRO
19502	EXDRP45000001	0-100 L/H

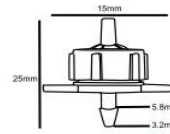
GOTERO AUTORREGULADO

- MARCA:** PIMTAS
- MATERIAL:** Polipropileno
- COLOR:** Azul - Morado - Verde

CARACTERÍSTICAS

Gotero autoregulado se utiliza en tuberías de riego por goteo no perforadas, proporciona facilidad de instalación en las áreas de uso. Las tuberías de riego por goteo sin perforar se perforan a 3 mm y se fijan fácilmente. También se utiliza en tubos Spaghetti de 7 mm. son extraíbles y facilitan el proceso de limpieza.

CÓDIGO SISTEMA	CÓDIGO FABRICANTE	L/H
19499	EXDRP43000001	2
19500	EXDRP43000002	4
19501	EXDRP43000003	8



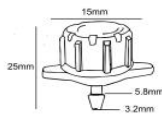
Tiene una superficie interna homogénea.



Protección UV.

GOTERO REGULABLE

- MARCA:** PIMTAS
- MATERIAL:** Caucho Natural
- COLOR:** Verde - Negro



CARACTERÍSTICAS

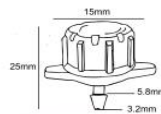
Gotero regulable se utiliza en tuberías por goteo no perforadas, es fácil de instalar. las tuberías de riego por goteo sin perforar se perforan a 3mm y se fijan fácilmente. también se utilizan en tubos Spaguetti de 7mm. Son extraíbles y facilitan su limpieza.



CÓDIGO SISTEMA	CÓDIGO FABRICANTE	L/H
----	EXDRP44000001	0-150 L/H

GOTERO REGULABLE

- MARCA:** PIMTAS
- MATERIAL:** Caucho Natural
- COLOR:** Verde - Negro



CARACTERÍSTICAS

Gotero regulable se utiliza en tuberías por goteo no perforadas, es fácil de instalar. las tuberías de riego por goteo sin perforar se perforan a 3mm y se fijan fácilmente. también se utilizan en tubos Spaguetti de 7mm. Son extraíbles y facilitan su limpieza.



CÓDIGO SISTEMA	CÓDIGO FABRICANTE	L/H
----	EXDRP46000001	0-70 L/h

MICROTUBO (ESPAQUETTI)

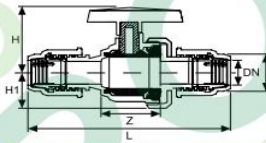
- MARCA:** PIMTAS
- MATERIAL:** Polipropileno
- COLOR:** Negro



CÓDIGO SISTEMA	CÓDIGO FABRICANTE	DIAMETRO (mm)	ROLLO
----	EXDRP47000001	5	250 mt
----	EXDRP47000002	7	250 mt
----	EXDRP47000003	5	60 cm
----	EXDRP47000004	7	60 cm
----	EXDRP47000005	5	100 cm
----	EXDRP47000006	7	100 cm

ENLACE VÁLVULA BOLA DESMONTABLE

- MARCA:** PIMTAS
- MATERIAL:** Polipropileno
- COLOR:** Azul - Negro



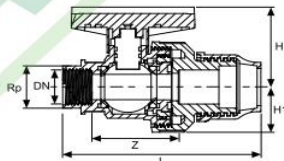
CARACTERÍSTICAS

Válvula de esfera desmontable en polipropileno, interior totalmente liso. Mínima pérdida de carga. No existe estrechamiento por su paso por la bola, lo que evita obturaciones y permite el paso de líquidos, esta cualidad la hace idónea para la industria agroalimentaria. Permiten fácilmente realizar montajes especiales. Están hechos de caucho natural y proporcionan hermeticidad, resisten a la corrosión no causa calcificación y óxido.

CÓDIGO SISTEMA	CÓDIGO FABRICANTE	DIAMETRO	DN (mm)	Z (mm)	L (mm)	H (mm)	H1 (mm)
----	233300202	20	15	52	129	47	27
----	233300252	25	20	58	140	51	31
----	233300322	32	25	61	150	63	33
----	233300502	50	40	82	200	86	46
----	233300632	63	50	95	224	101	58

ENLACE VÁLVULA DE BOLA ROSCA HEMBRA

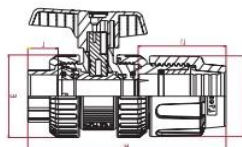
- MARCA:** PIMTAS
- MATERIAL:** Polipropileno
- COLOR:** Azul - Negro



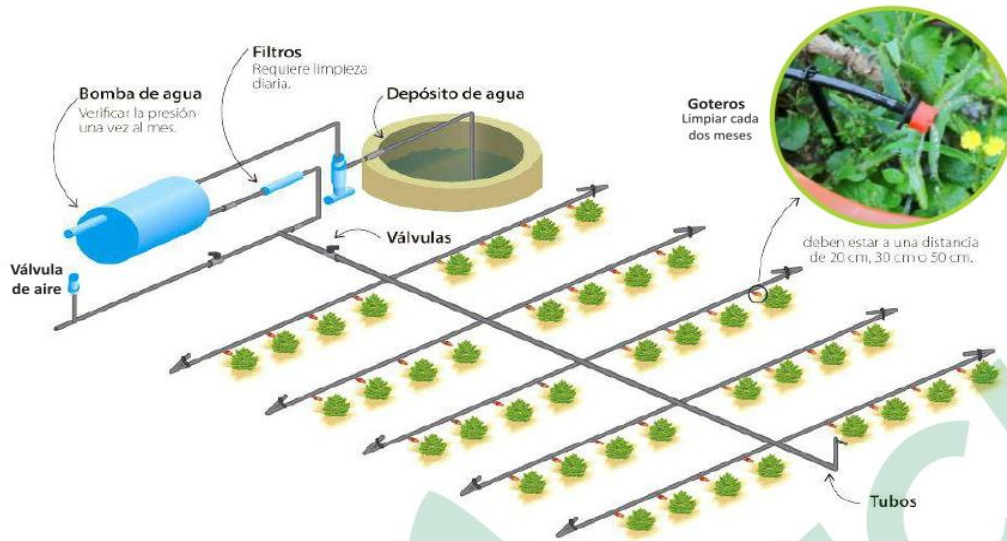
CÓDIGO SISTEMA	CÓDIGO FABRICANTE	DIAMETRO	DN (mm)	Z (mm)	L (mm)	H (mm)	H1 (mm)
----	233100252	25	20	58	108	51	31
----	233100322	32	25	61	120	63	33
----	233100502	50	40	82	175	86	46
----	233100632	63	50	95	190	101	58

CARACTERÍSTICAS

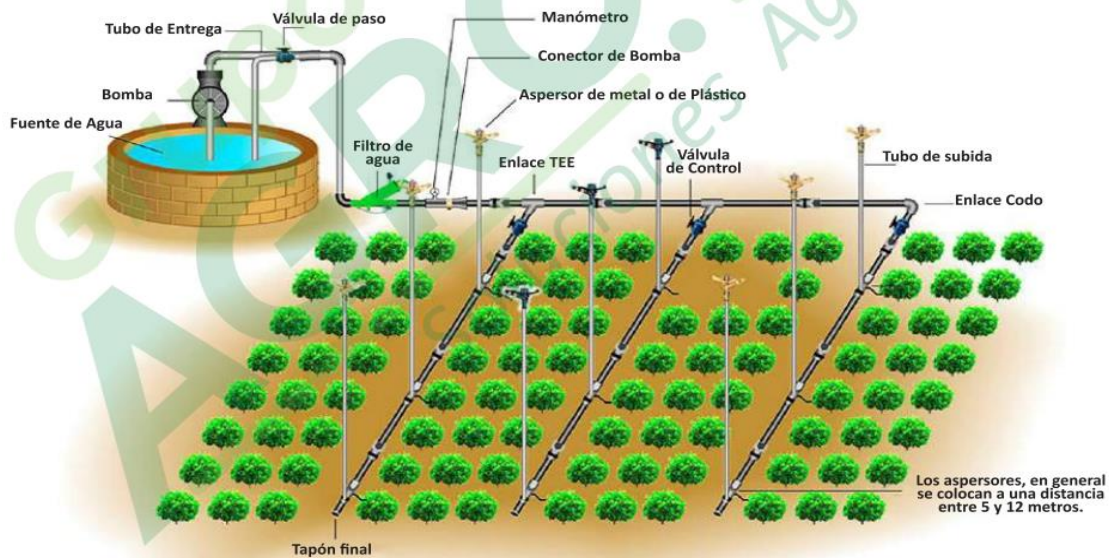
La válvula de bola tiene un mecanismo de llave de paso que sirve para regular el flujo de un fluido canalizado y se caracteriza por que el mecanismo regulador, situado en el interior tiene forma de esfera perforada, se abre mediante el giro del eje unido a la esfera, cuando la válvula está cerrada, el agujero estará perpendicular a la entrada y salida. Están hechos de caucho natural y proporcionan hermeticidad, resisten a la corrosión no causa calcificación y óxido.



RIEGO POR GOTEO



RIEGO POR ASPERSIÓN



RAIN BIRD

 **PIMTAS**[®]

 **NETAFIM**

Sunbeam
Simple and secure connection.


Grupo AGROTECH
Soluciones Agrícolas

Quiver

 **VYRSA**



K-PLASTIC

 **Rivulis**

 **jacto**

HONDA

 **TRUPER**[®]

 ventas@grupoagrotech.com

 946 420 128 / 922 030 485

 Jr Huancavelica 1196 Mz "C" Lt. 11 Cercado de Lima - Perú

Grupo Agrotech



www.agrominplast.com



Goteros Ajustables



TS ISO 9001:2015



Características



- Fabricados a partir de materias primas de polipropileno de alta calidad; altamente resistentes a los impactos.
- Son fáciles de instalar.
- Se usan en mangueras de polietileno de baja densidad (LDPE) (sin gotero) de Ø16 mm, Ø20 mm, Ø25 mm y Ø32 mm.
- Se instalan fácilmente con solo perforar un agujero de 3 mm en la manguera.
- La capacidad mínima de filtración del filtro utilizado en el sistema deberá ser de 125 mesh.
- La cubierta desmontable simple del gotero facilita el proceso de limpieza.

Código	Medida	Cantidad por Caja	Cantidad por Bolsa
70101002	0-70 Lt/h	8000	100
70101001	0-100 Lt/h	4000	100

Gotero azul (clic)	Valores de descarga (L/h)					
	0.5 bar	1.0 bar	1.5 bar	2.0 bar	2.5 bar	3.0 bar
2	4.0	9.8	12.2	15.2	17.3	20.2
8	9.9	15.7	22.8	26.4	27.9	33.0
12	18.7	30.0	38.9	45.1	49.7	57.0
16 (1 vuelta)	25.4	38.9	53.1	60.4	68.1	75.7
20	29.6	45.0	61.8	69.7	78.9	88.4
24	33.9	52.4	71.2	78.0	90.5	101.0

Gotero verde (clic)	Valores de descarga (L/h)					
	0.5 bar	1.0 bar	1.5 bar	2.0 bar	2.5 bar	3.0 bar
2	0.3	0.4	0.5	0.6	0.8	1.0
8	9.6	15.2	19.0	21.8	25.4	30.1
12	13.7	22.3	28.3	33.6	37.9	42.6
16 (1 vuelta)	31.0	48.2	62.5	72.3	79.6	84.7
20	56.2	84.9	110.9	125.7	137.0	146.0
24	81.7	116.5	147.6	172.5	187.8	206.3
28	84.8	122.9	153.4	179.0	202.4	216.4
32	86.0	125.3	154.1	179.4	202.9	217.3



☎ 01 784 1625 ✉ ventas@iesta.pe 🌐 www.riego.pe 📘 [/iestasac](https://www.facebook.com/iestasac) 📷 [/iestasac](https://www.instagram.com/iestasac)

Distribuidor exclusivo de:



Turkey

Discover
the potential

Anexo F:
Presupuesto de la investigación
de riego automatizado

1. Bienes

Descripción	Unidad	Cantidad	Precio Unitario (S/.)	Precio Total (S/.)
Sensor de ultrasonido HC-SR04.	Und	1	8	8
02 sensores de presión Hk3025 5.0MPA.	Und	2	180	360
Sensor de flujo de agua 2" YF-DN50.	Und	1	270	270
Sensor DHT11 de temperatura	Und	1	12	12
02 módulo higrómetro sensor de humedad del suelo	Und	2	32	64
Modulo Relé de 4 canales 5v.	Und	1	24	24
Placa Arduino Mega 2560 Rev3.	Und	2	215	430
Modulo Ethernet shield W5100.	Und	2	65	130
Módulo Lector de memoria Micro SD Card 74LVC125A	Und	2	12	24
Electroválvula de 24 VAC	Und	1	438	438
Protoboard Blanco	Und	1	7	7
Cable Dupont Macho Macho.	Und	2	5.5	11
Cable Dupont Macho Hembra.	Und	2	5.5	11
Cable Dupont Hembra Hembra.	Und	2	5.5	11
Cable internet Red UTP Cat. 5e	Und	1	260	260
Cable eléctrico #14 AWG rollo x100m	Und	2	148	296
Medidor de flujo de agua digital	Und	1	165	165
Medidor de Temperatura y humedad	Und	1	125	125
Router Modem Para Chip 4g Liberados	Und	1	340	340
Materiales de Tablero eléctrico	Und	1	550	550
Semilla zanahoria	gr.	50	10	10
Memoria de 64 GB	Und	1	45	45
Materiales para la caseta de control	Und	1	160	160
Materiales para las cajas de protección	Und	1	210	210
Accesorios para la conexión de los sensores	Und	1	250	250
Total				4211

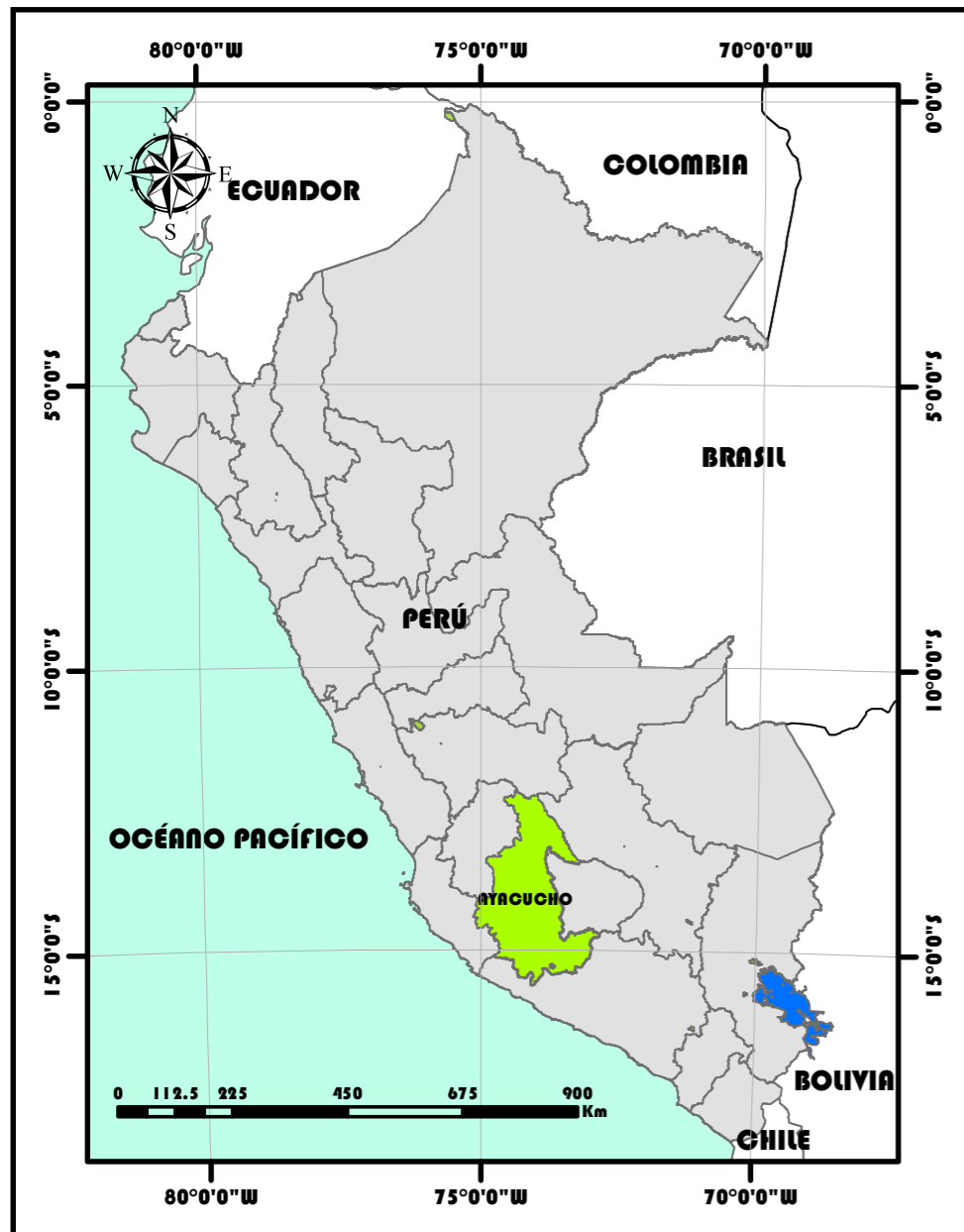
2. Servicios

Descripción	Cantidad	Precio Unitario (S/.)	Precio Total (S/.)
Mano de obra			2580
Construcción del circuito eléctrico	1	160	160
Programación del sistema automatizado	1	800	800
Laboratorio análisis suelo y agua	1	260	260
Instalación de caseta control y cajas de protección	1	200	200
Movilidad		1000	1000
Internet (mensual)	4	40	160

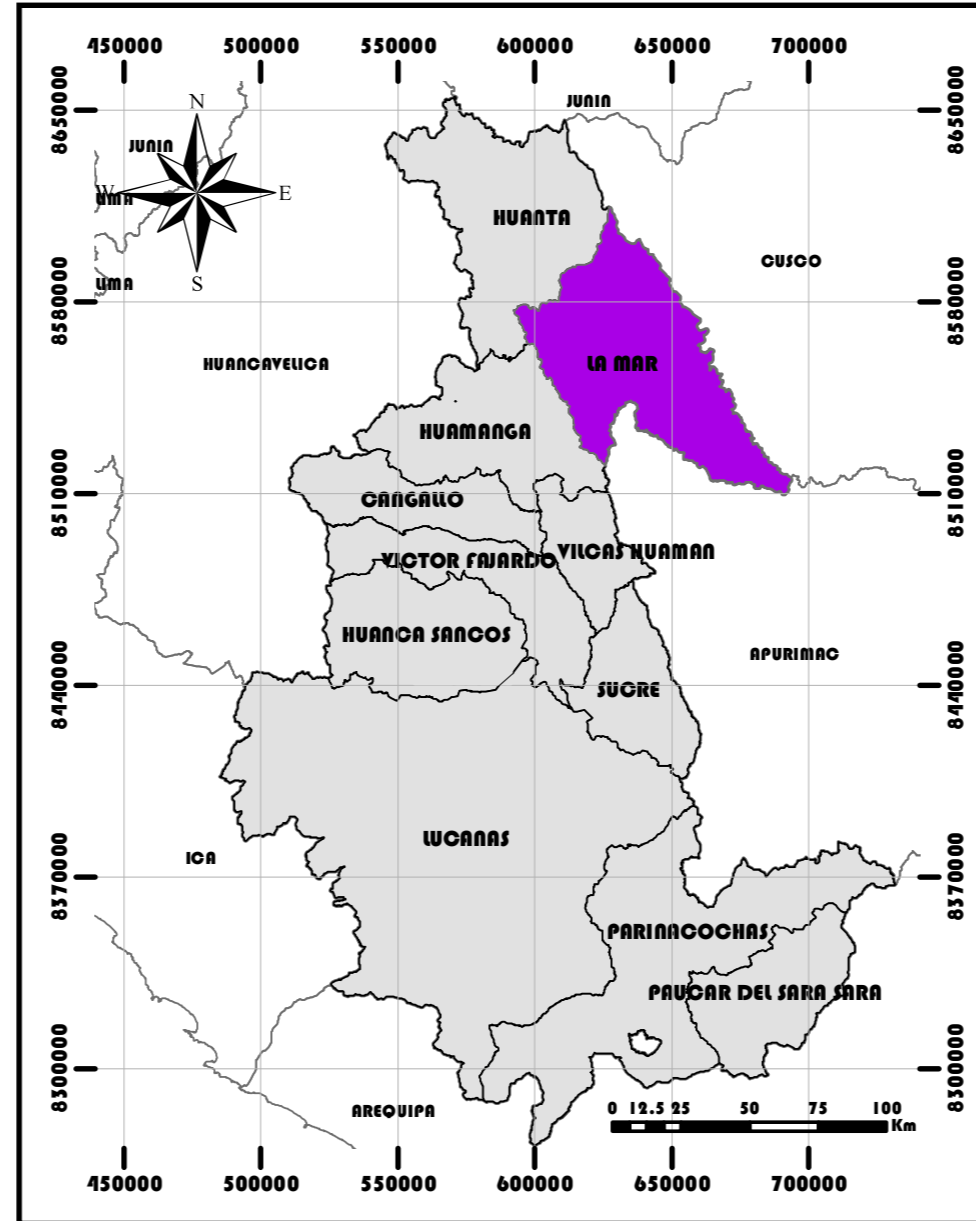
El presupuesto fue de carácter autofinanciado

Anexo G: Planos

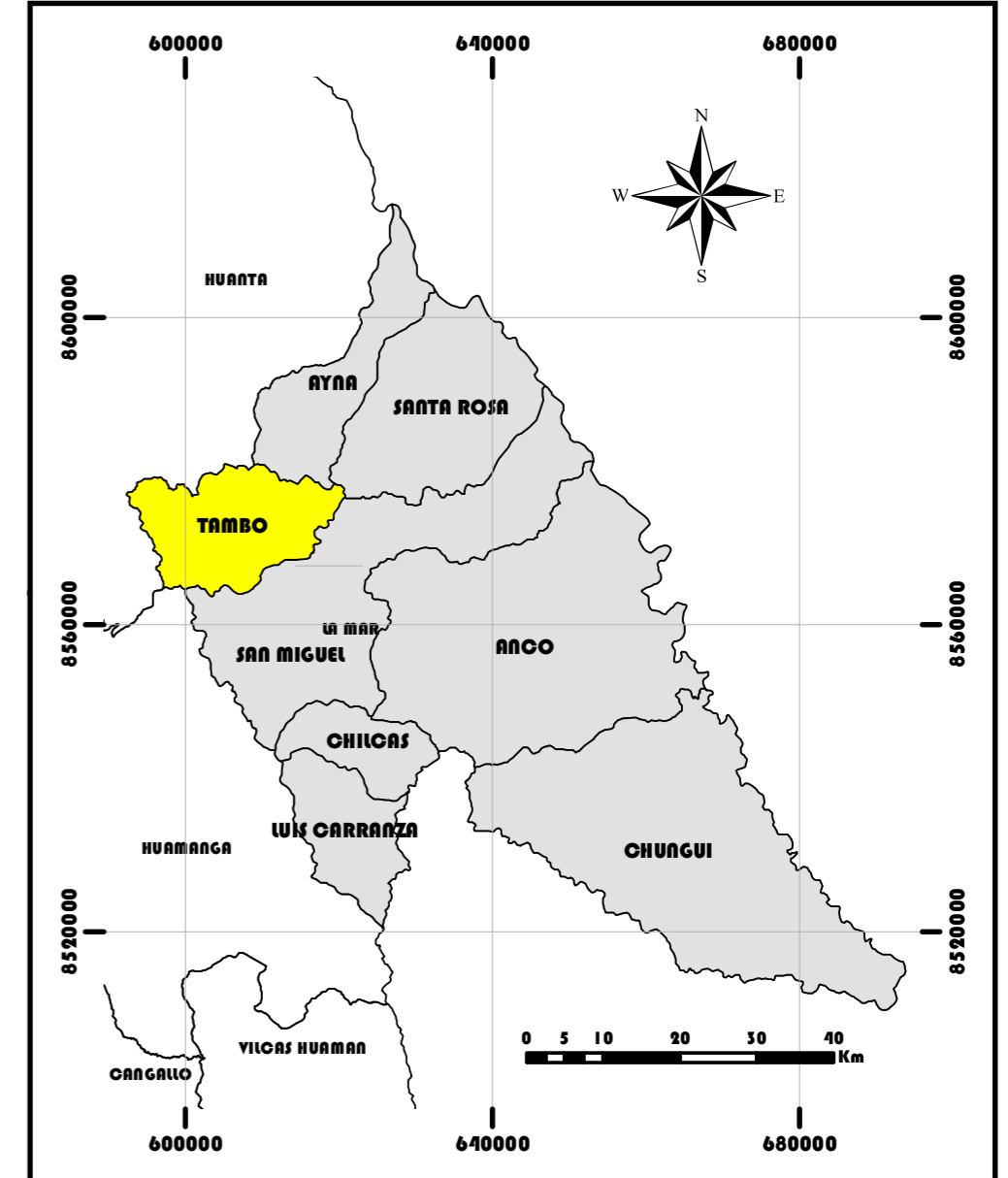
1. MAPA DEPARTAMENTAL



2. MAPA PROVINCIAL



3. MAPA DISTRITAL





4. MAPA UBICACIÓN DEL PROYECTO



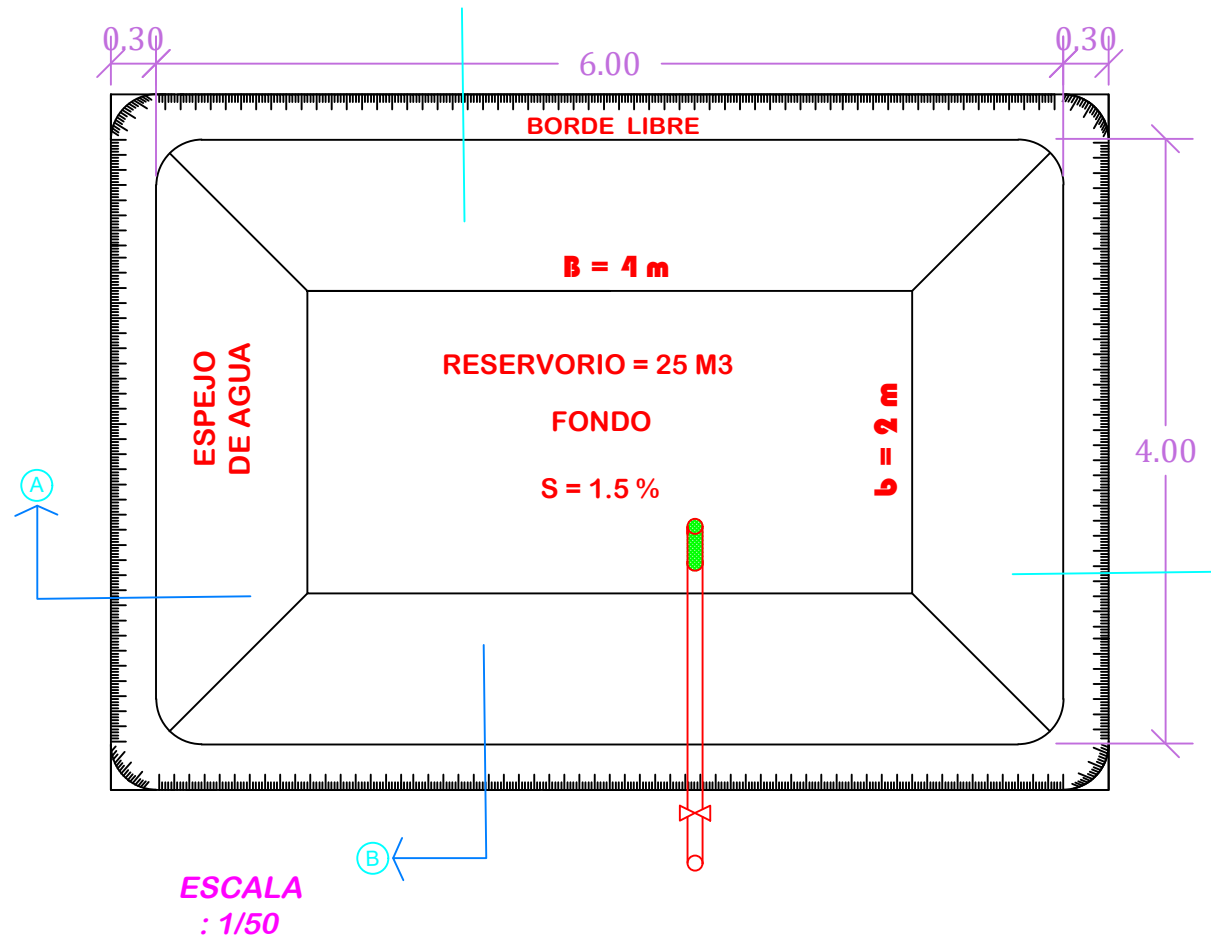
UBICACIÓN SATELITAL DEL PROYECTO



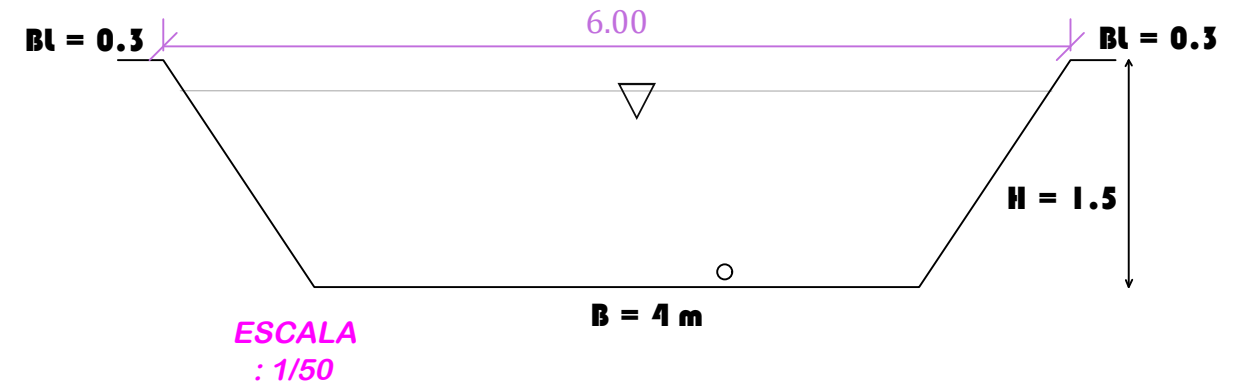
UBICACIÓN POLÍTICA DE LA ZONA

	UNIVERSIDAD NACIONAL SAN CRISTÓBAL DE HUAMANGA FACULTAD CIENCIAS AGRARIAS ESCUELA PROFESIONAL DE INGENIERIA AGRICOLA		
	UBICACIÓN: REGIÓN : AYACUCHO PROVINCIA : LA MAR DISTRITO : TAMBO CC.PP : PAMPA HERMOSA	PROYECTO : Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2024	
AUTOR: Diana Marisella Loayza Paredes		FECHA: Febrero-2025	ESCALA: INDICADA

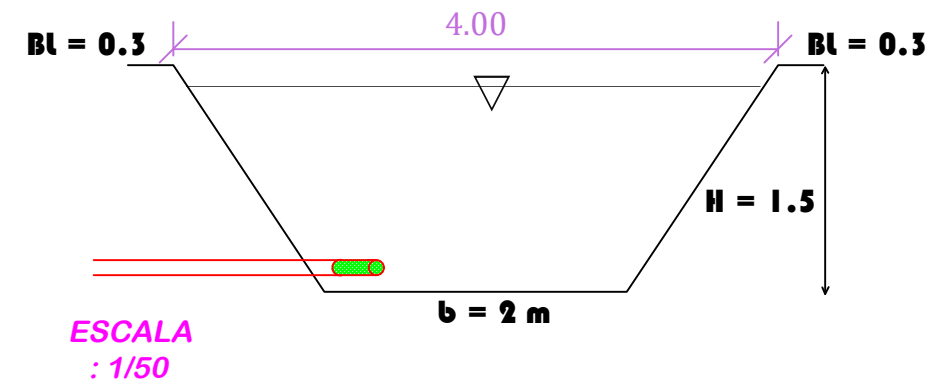
RESERVORIO - PLANTA





CORTE A - A

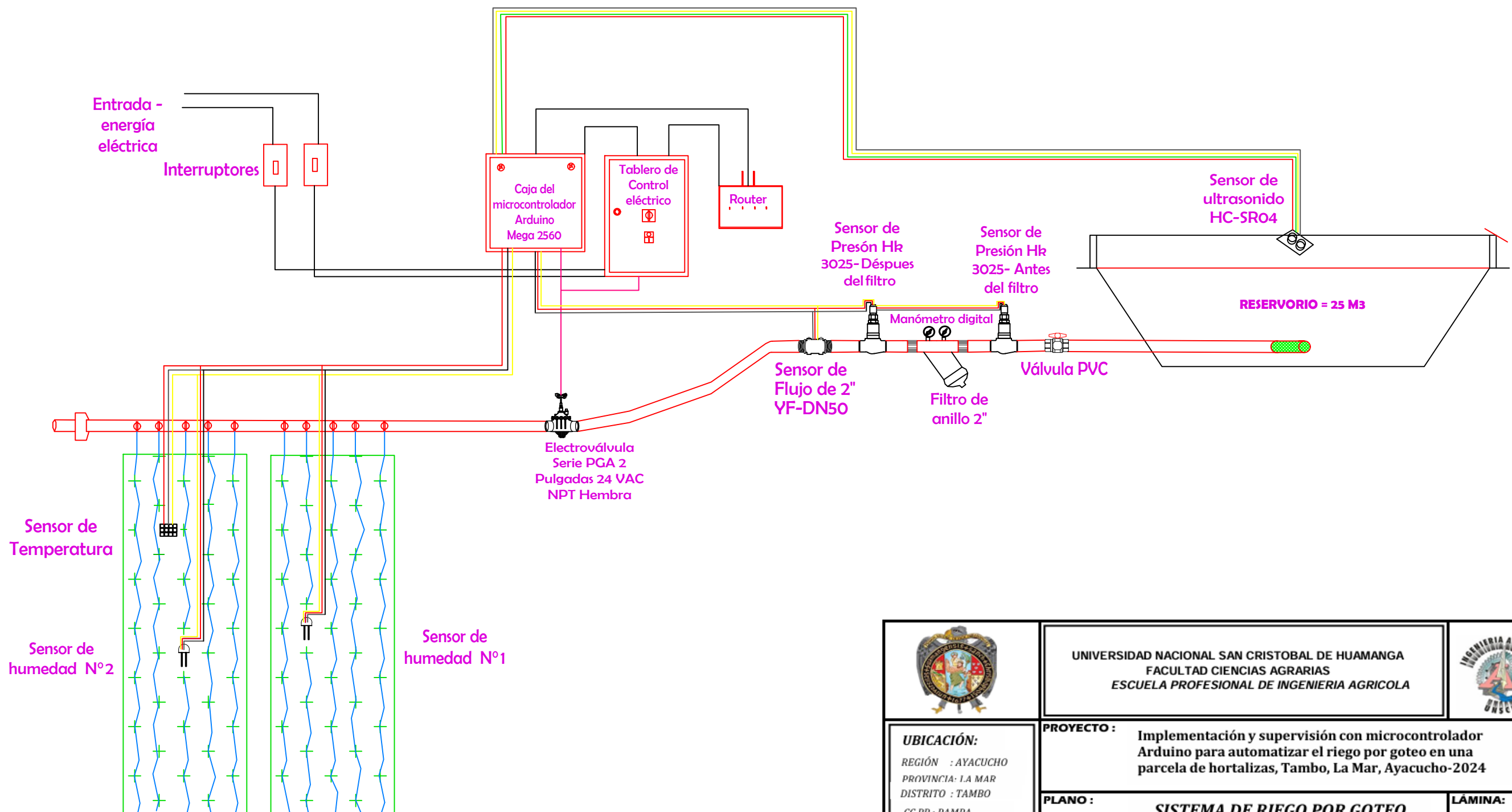




CORTE B - B



	UNIVERSIDAD NACIONAL SAN CRISTOBAL DE HUAMANGA FACULTAD CIENCIAS AGRARIAS ESCUELA PROFESIONAL DE INGENIERIA AGRICOLA		
	PROYECTO : Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2024		
UBICACIÓN: REGIÓN : AYACUCHO PROVINCIA: LA MAR DISTRITO : TAMBO CC.PP : PAMPA HERMOSA	PLANO : RESERVORIO		LÁMINA: 02
AUTOR: Diana Marisella Loayza Paredes	FECHA: Febrero-2025	ESCALA: INDICADA	

SISTEMA DE RIEGO POR GOTEO - AUTOMATIZADO CON MICROCONTROLADOR ARDUINO



	UNIVERSIDAD NACIONAL SAN CRISTOBAL DE HUAMANGA FACULTAD CIENCIAS AGRARIAS ESCUELA PROFESIONAL DE INGENIERIA AGRICOLA		
	PROYECTO : Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2024		
UBICACIÓN: REGIÓN : AYACUCHO PROVINCIA: I.A MAR DISTRITO : TAMBO CC.PP : PAMPA HERMOSA	PLANO : SISTEMA DE RIEGO POR GOTEO		LÁMINA:
	AUTOR: Diana Marisella Loayza Paredes		FECHA: Marzo-2025
			03

**ACTA DE SUSTENTACIÓN DE TESIS
Bach. DIANA MARISELLA LOAYZA PAREDES****R.D. N° 139-2025-UNSCH-FCA-D**

En la ciudad de Ayacucho a los seis días del mes de junio del año dos mil veinticinco, siendo las diez horas, se reunieron en el auditorio de la Facultad de Ciencias Agrarias, bajo la presidencia del Dr. Felipe Escobar Ramírez Decano de la Facultad de Ciencias Agrarias; los miembros del jurado conformado por el Ing. Orlando Fidel Sulca Castilla, M.Sc. César Vidal Gutiérrez Ninahuamán como asesor, Mtro. Jaime Villar Rodríguez y el Ing. Herbert Núñez Alfaro; actuando como secretario de actas el Mtro. Rodolfo Alca Mendoza, para recibir la sustentación de la Tesis titulado: **Implementación y supervisión con microcontrolador Arduino para automatizar el riego por goteo en una parcela de hortalizas, Tambo, La Mar, Ayacucho-2024**, para obtener el Título Profesional de Ingeniera Agrícola presentado por la Bachiller **DIANA MARISELLA LOAYZA PAREDES**.

El señor Decano previa verificación de los documentos exigidos solicitó se proceda con la sustentación y posterior defensa de la tesis en un periodo de cuarenta y cinco minutos de acuerdo al reglamento de grados y títulos vigente. Terminado la exposición, los miembros del Jurado, formularon sus preguntas, aclaraciones y/o observaciones correspondientes. Luego se invito a los miembros del jurado pasar a otra aula para la deliberación y calificación del trabajo de tesis, teniendo el siguiente resultado:

Jurado evaluador	Exposición	Respuestas a las preguntas	Generación de conocimiento	Promedio
Ing. Orlando Fidel Sulca Castilla	16	16	16	16
M.Sc. César Vidal Gutiérrez Ninahuamán	17	16	18	17
Mtro. Jaime Villar Rodríguez	14	16	15	15
Ing. Herbert Núñez Alfaro	16	16	16	16
PROMEDIO GENERAL				16

OBSERVACIÓN: Por acuerdo unánime de los miembros del jurado, el título del trabajo de investigación debe ser: **Implementación y monitoreo con microcontrolador Arduino para automatizar el riego por goteo en parcela de hortalizas, Tambo, Ayacucho - 2024**

Acto seguido se invita al sustentante y publico en general para dar a conocer el resultado final. Firman el acta.

Ing. Orlando Fidel Sulca Castilla
*Presidente***M.Sc. César Vidal Gutiérrez Ninahuamán**
*Asesor***Mtro. Jaime Villar Rodríguez**
*Jurado***Ing. Herbert Núñez Alfaro**
*Jurado***Mtro. Rodolfo Alca Mendoza**
Secretario Docente



UNSCH

FACULTAD DE CIENCIAS
AGRARIAS

CONSTANCIA DE CONTROL DE ORIGINALIDAD DE TRABAJO DE TESIS

El que suscribe, miembro de la comisión de docentes instructores responsables de operativisar, verificar, garantizar y controlar la originalidad de los trabajos de **TESIS** de la Facultad de Ciencias Agrarias, de la Universidad Nacional de San Cristóbal de Huamanga, autorizado por R.C.F. N° N° 005-2024-UNSCH-FCA-CF; hace constar que el trabajo titulado;

Implementación y monitoreo con microcontrolador Arduino para automatizar el riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024

Autor : Diana Marisella Loayza Paredes

Asesor : César Vidal Gutiérrez Ninahuamán

Ha sido sometido al control de originalidad mediante el software TURNITIN UNSCH, acorde al Reglamento de originalidad de trabajos de Tesis, aprobado mediante la RCU N° 039-2021-UNSCH-CU, arrojando un resultado de Siete por ciento (**7 %**) de índice de similitud, realizado con **depósito de trabajos estándar**.

En consecuencia, se otorga la presente Constancia de Originalidad para los fines pertinentes.

Nota: Se adjunta el resultado con Identificador de la entrega: 2698451268

Ayacucho, 13 de junio de 2025

UNIVERSIDAD NACIONAL DE
SAN CRISTÓBAL DE HUAMANGA
Facultad de Ciencias Agrarias
Ing. Edgar Tenorio Mancilla
Coordinador de Control de originalidad de
trabajo de Investigación - Tesis - FCA

Implementación y monitoreo con microcontrolador Arduino para automatizar el riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024

por Diana Marisella LOAYZA PAREDES

Fecha de entrega: 13-jun-2025 11:05a.m. (UTC-0500)

Identificador de la entrega: 2698451268

Nombre del archivo: TESIS_LOAYZA_PAREDES__DIANA_M..pdf (17.9M)

Total de palabras: 37821

Total de caracteres: 164727

Implementación y monitoreo con microcontrolador Arduino para automatizar el riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024

INFORME DE ORIGINALIDAD

7%

INDICE DE SIMILITUD

7%

FUENTES DE INTERNET

1%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.lamolina.edu.pe Fuente de Internet	1%
2	hdl.handle.net Fuente de Internet	1%
3	naylampmechatronics.com Fuente de Internet	<1%
4	repositorio.inia.gob.pe Fuente de Internet	<1%
5	repositorio.umsa.bo Fuente de Internet	<1%
6	repositorio.utn.edu.ec Fuente de Internet	<1%
7	repositorio.unsch.edu.pe Fuente de Internet	<1%
8	repositorio.una.edu.ni Fuente de Internet	<1%

9	Submitted to Universidad Nacional de San Cristóbal de Huamanga	<1 %
Trabajo del estudiante		
10	bibdigital.epn.edu.ec	<1 %
Fuente de Internet		
11	dspace.unl.edu.ec	<1 %
Fuente de Internet		
12	repositorio.usmp.edu.pe	<1 %
Fuente de Internet		
13	repositorio.upao.edu.pe	<1 %
Fuente de Internet		
14	Submitted to Universidad Internacional de la Rioja	<1 %
Trabajo del estudiante		
15	redivia.gva.es	<1 %
Fuente de Internet		
16	repositorio.unasam.edu.pe	<1 %
Fuente de Internet		
17	Submitted to Universidad Técnica Nacional de Costa Rica	<1 %
Trabajo del estudiante		
18	repositorio.utp.edu.pe	<1 %
Fuente de Internet		
19	repositorio.ucp.edu.pe	<1 %
Fuente de Internet		

20

www.electrogeekshop.com

Fuente de Internet

<1 %

21

repositorio.unp.edu.pe

Fuente de Internet

<1 %

22

repositorio.unica.edu.pe

Fuente de Internet

<1 %

23

www.automatizacionparatodos.com

Fuente de Internet

<1 %

24

repositorio.unal.edu.co

Fuente de Internet

<1 %

Excluir citas

Activo

Excluir coincidencias

< 30 words

Excluir bibliografía

Activo

Implementación y monitoreo con microcontrolador Arduino para automatizar el riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024

Implementation and Monitoring with Arduino Microcontroller for Automating Drip Irrigation in a Vegetable Plot, Tambo, Ayacucho – 2024

Diana Marisella Loayza Paredes¹ diana.loayza.21@unsch.edu.pe

Cesar Vidal Gutierrez Ninahuaman ² cesar.gutierrez@unsch.edu.pe

Resumen

En el distrito de Tambo, provincia de La Mar, del departamento de Ayacucho, los métodos tradicionales de riego empleados en la producción de hortalizas han provocado un uso ineficiente del recurso hídrico y pérdidas de tiempo en el manejo y control del riego. Esta situación, junto con la necesidad de modernizar las prácticas agrícolas y aumentar la competitividad del sector, motivó la presente investigación. Este estudio tiene como objetivo general implementar y monitorear un sistema automatizado de riego por goteo basado en un microcontrolador Arduino en parcela dedicada a hortalizas. Para alcanzar este objetivo, se establecieron objetivos específicos que permitieron una comprensión detallada del sistema. Se recopiló información básica mediante la toma de muestras de suelo y agua, y se evaluó el coeficiente de uniformidad del riego. Asimismo, se seleccionaron, programaron, implementaron y calibraron sensores de humedad, caudal, presión, temperatura y nivel de agua, así como de la electroválvula. Los resultados mostraron que el sistema automatizado funcionó correctamente según los parámetros definidos, manteniendo una comunicación estable entre los sensores y el microcontrolador Arduino. El sistema permitió optimizar el uso del agua, reducir el tiempo en el monitoreo y alcanzar una producción estimada de 245 kg de zanahorias en 35 m², equivalente a un rendimiento de 7 kg/m², superando significativamente el rendimiento del riego convencional, que es de 2.5 kg/m², lo que evidencia la eficacia del sistema. En conclusión, la automatización del riego por goteo mediante Arduino constituye una solución viable y replicable en zonas rurales, ya que mejora el rendimiento agrícola, reduce el consumo de agua y promueve el uso sostenible de tecnologías accesibles.

Palabras clave: Riego automatizado, Arduino, Sensores, Tecnología, Calibración.

Abstract

In the district of Tambo, province of La Mar, in the department of Ayacucho, traditional irrigation methods used in vegetable production have led to inefficient water use and time losses in irrigation management and control. This situation, along with the need to modernize agricultural practices and increase the competitiveness of the sector, motivated the present research. The general objective of this study was to implement and to monitor an automated drip irrigation system based on an Arduino microcontroller in a vegetable cultivation plot. To achieve this, specific objectives were established to provide a detailed understanding of the system. Basic information was collected through soil and water sampling, and the irrigation uniformity coefficient was evaluated. Sensors for soil moisture, flow, pressure, temperature, and water level were selected, programmed, implemented, and calibrated, along with the solenoid valve. The results showed that the automated system operated correctly according to the defined parameters, maintaining stable communication between the sensors and the Arduino microcontroller. The system optimized water use, reduced monitoring time, and achieved an estimated production of 245 kg of carrots in 35 m², equivalent to a yield of 7 kg/m², significantly surpassing the conventional irrigation yield of 2.5 kg/m², thus demonstrating the system's effectiveness. In conclusion, the automation of drip irrigation using Arduino is a viable and replicable solution in rural areas, as it improves agricultural yield, reduces water consumption, and promotes the sustainable use of accessible technologies.

Keywords: Automated irrigation, Arduino, Sensors, Technology, Calibration

I. INTRODUCCIÓN

Para maximizar los rendimientos de los cultivos, aprovechar eficientemente los recursos hídricos y optimizar el uso de tecnologías que incrementen la competitividad y rentabilidad en la actividad agrícola, es esencial garantizar un manejo óptimo de los sistemas de riego. (Martínez, 2001)

El suministro de agua, a través de diversas técnicas de irrigación como el riego superficial, por aspersión, micro aspersión y goteo, desempeña un papel fundamental en la producción agrícola al influir directamente en el crecimiento y desarrollo de las plantas (Hargreaves & Merkle G.P., 2000).

Sin embargo, los métodos de riego convencionales utilizados en zonas agrícolas suelen llevar a un uso ineficiente del agua, lo que genera pérdidas de tiempo para los agricultores, rendimientos subóptimos en el cultivo de zanahoria y, en consecuencia, pérdidas económicas en la producción final. Para abordar estos problemas, esta tesis propone una alternativa, que es la implementación de esta tecnología Arduino, ya que proporciona al agricultor un control preciso sobre el suministro de agua, mediante la monitorización constante de la humedad del suelo a

través de sensores y contribuye a mejorar el rendimiento del cultivo de zanahoria y reduce significativamente el tiempo de trabajo.

Finalmente, este trabajo de tesis tuvo un paso significativo hacia el avance tecnológico en el ámbito agropecuario del distrito Tambo, la localidad de Pampa hermosa, anexo Lorensayoq, y se espera que sirva como modelo y catalizador para futuras iniciativas similares en la región y a nivel nacional. La combinación de la tecnología Arduino con el riego por goteo ofrece una solución innovadora y adaptable a las necesidades específicas de la agricultura en la zona, mejorando así la calidad de vida de los agricultores y promoviendo la sostenibilidad en el uso de los recursos naturales.

El objetivo principal de este trabajo de investigación consistió en Implementar y monitorear con microcontrolador Arduino para la automatización de riego por goteo en parcela de hortalizas, Tambo, Ayacucho-2024.

II. MATERIALES Y MÉTODOS

2.1 Materiales

2.1.1. Equipo para la automatización

- Sensor de ultrasonido HC-SR04.
- 02 sensores de presión Hk3025.
- Sensor de flujo de agua 2" YF-DN50.

- Sensor DHT11 de temperatura
- 02 módulo higrómetro sensor de humedad del suelo
- Electroválvula

2.2. Metodología

2.2.1 Recopilación de información sobre el sistema de riego del invernadero (tomar muestras para el análisis de suelo – agua y prueba de coeficiente de uniformidad)

Como parte del estudio, se recolectaron muestras de suelo para su análisis físico en el laboratorio "AGROLAB". Los resultados obtenidos fueron fundamentales para ajustar la programación del sensor de humedad

Tabla 1

Resultado del análisis del agua

Clase Textural	Capacidad de campo (%)	Punto de marchitez (%)	Análisis granulométrico		
			Arena (%)	Limo (%)	Arcilla (%)
Franco Arcilloso	27.45	14.09	41	21	38

Los componentes químicos del agua pueden tener un impacto directo en el desarrollo de las plantas. Cabe resaltar que los resultados del Laboratorio AGROLAB clasificó : **C2-S1** (Agua de salinidad media válida para riego).

Además , la eficiencia con la que se distribuye el agua en un sistema de riego, representa el grado de homogeneidad en la aplicación del agua sobre el suelo y es fundamental para garantizar un crecimiento uniforme de los cultivos.

Por ello se logró verificar si el sistema de goteo cumple con la distribución uniforme después de la automatización según la fórmula de J.E. Christiansen:

$$CU = 100 * \left(1 - \frac{\sum_i^n |X_i - X|}{n * X} \right) \dots Ecuación (1)$$

- CU: Coeficiente de uniformidad %
- n: Números de puntos de control
- X: Promedio de todos los datos (ml)
- Xi: Cantidad de gotas en cada punto de control (ml)

$$CU = 100 * \left(1 - \frac{437}{36 * 124.17} \right)$$

$$CU = 90.2 \%$$

2.2.2. Selección de sensores, actuador y los demás materiales para la automatización del sistema de riego por goteo del invernadero.

La implementación de un sistema de riego por goteo automatizado en un invernadero, basado en tecnologías IoT, requirió una selección cuidadosa de sensores, actuador y materiales que no solo garanticen la funcionalidad, sino

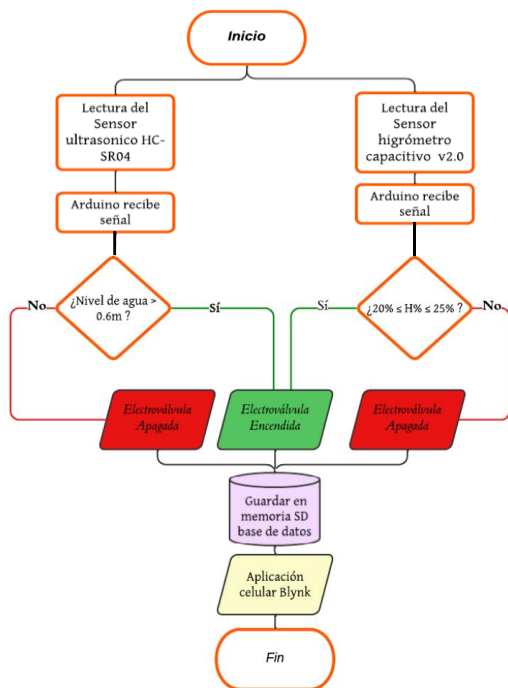
también la sostenibilidad y la asequibilidad del sistema.

2.2.3. Programación de los sensores y actuador en el sketch de Arduino con sus respectivos códigos para su funcionamiento.

Antes de programar los sensores en Arduino, primero se diseñó un flujograma para organizar y visualizar el funcionamiento del sistema. Esto ayudó a planificar la lógica antes de escribir el código y posterior realizar la programación.

Figura 1

Diagrama de flujo de trabajo de la electroválvula en función del sensor ultrasónico y humedad del suelo



a) Sensor de ultrasonido HC-SR04

Ayudo a monitorear el nivel y volumen del agua donde era almacenada

en un reservorio de geomembrana de 25 m3.

b) Sensor de humedad del suelo

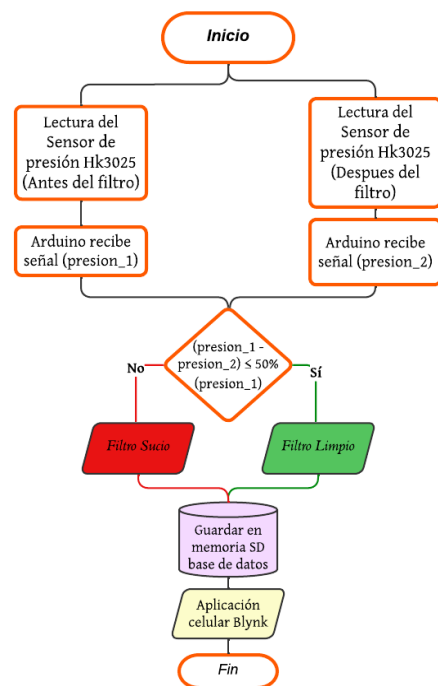
Este sensor definió en qué momento se activaba el riego, según la humedad que identificaba en el suelo del área cultivada.

c) Sensor de presión Hk3025

Este sensor permitió monitorear el estado del filtro de anilla, asegurando que el agua llegara de manera uniforme al cultivo.

Figura 2

Diagrama de flujo del sensor de presión antes y después del filtro

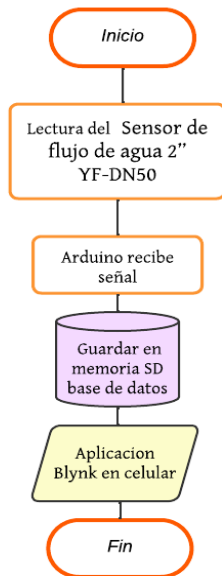


d) Sensor de flujo de agua 2” YF-DN50

Este sensor permitió medir el caudal como parte del monitoreo en la entrega de agua para el riego.

Figura 3

Diagrama de flujo del sensor de flujo de agua 2'' YF-DN50



e) Sensor DHT11 de temperatura

Ayudo a monitorear y registrar las condiciones ambientales dentro del invernadero, facilitando un control preciso del entorno de cultivo.

Figura 4

Diagrama de flujo del sensor DHT11 de temperatura



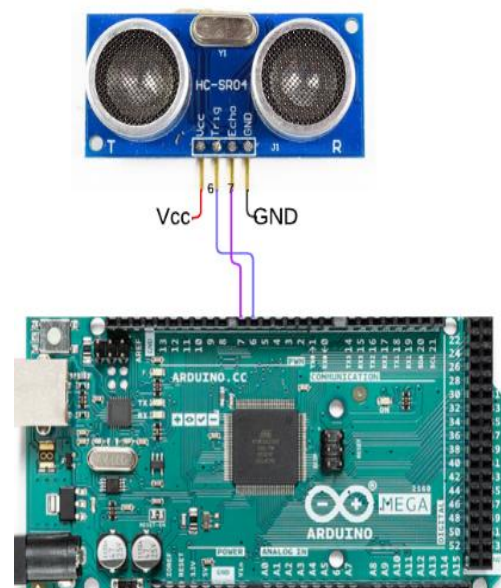
2.2.4. Implementación de sensores y actuador en el sistema de riego por goteo del invernadero y realizar las pruebas exhaustivas.

a) Sensor de Ultrasonido HC-SR04

mediante el disparo del pin Trigger (6) se media la distancia respecto al espejo del agua, donde es rebotado y decepcionado por el sensor a través del pin Echo (7) , dato necesario para medir el nivel y volumen del agua del reservorio.

Figura 5

Comunicación entre el sensor ultrasonido HC-SR04 y placa Arduino

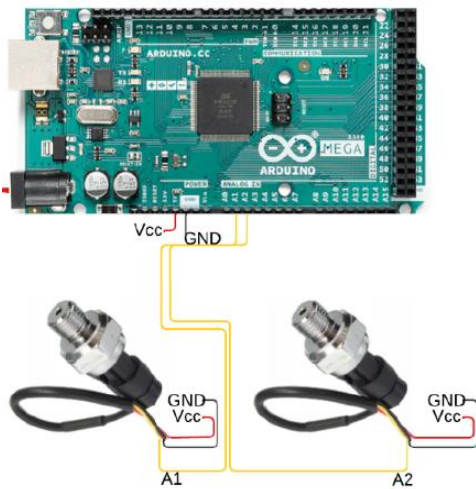


b) Sensor de presión Hk3025

Mediante el pin A1 y pin A2 , permitió enviar los valores de presión antes y después del filtro, necesarios para evaluar el estado del anillo del filtro.

Figura 6

Comunicación entre el Sensor de presión Hk3025 y placa Arduino

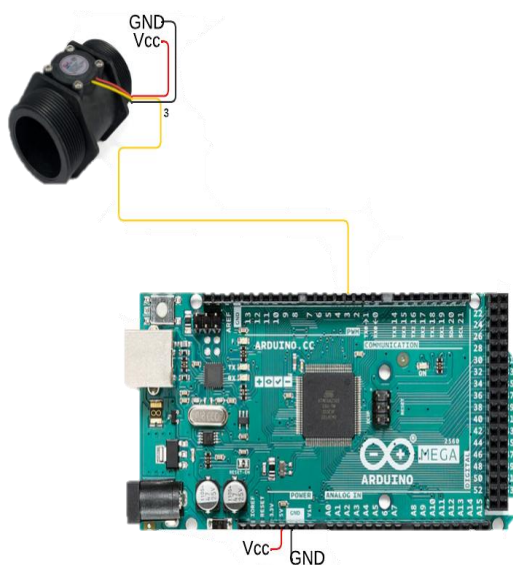


c) Sensor de flujo de agua YF-DN50

Este sensor cuenta con tres cables de distintos colores: rojo (alimentación) y negro (tierra) las cuales fueron conectados al protoboard y el amarillo (salida) al pin 3 para toma de datos.

Figura 7

Comunicación entre Sensor de flujo de agua YF-DN50 y placa Arduino

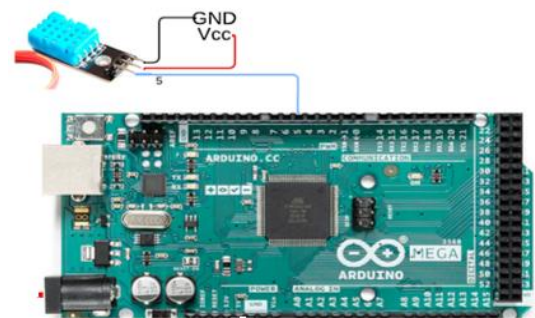


d) Sensor DHT11 de temperatura

cuenta con los pines correspondientes, DATA (señal) fue conectado al pin 5, VCC (alimentación) y GND (tierra).

Figura 8

Comunicación del Sensor DHT11 de temperatura y la placa Arduino

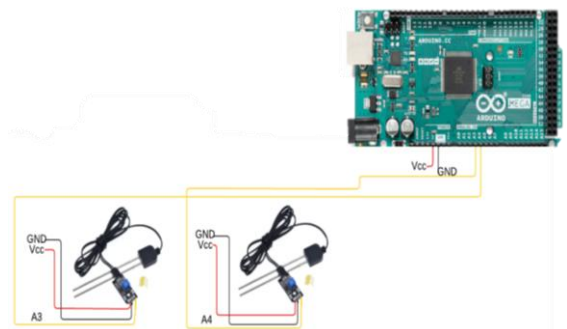


e) Módulo higrómetro sensor de humedad del suelo

El sensor de humedad del suelo cuenta con varios pines fundamentales para su funcionamiento: VCC (suministra la energía), GND (tierra); y el pin A3 y A4, son los encargados de transmitir la señal.

Figura 9

Comunicación entre el sensor de humedad y placa Arduino

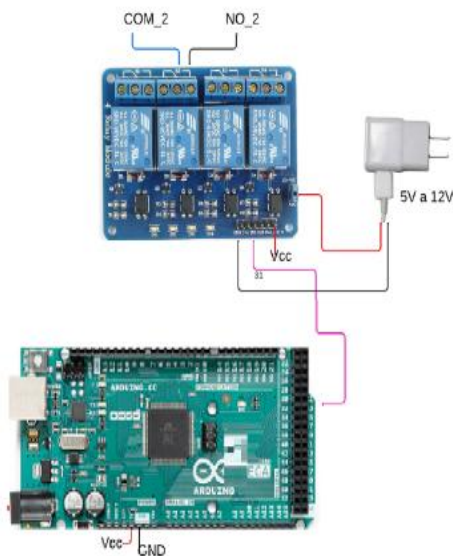


f) Módulo Relé de 4 canales

Se instaló un módulo relé en la caseta de control para manejar altos voltajes con una señal pequeña desde una placa Arduino Mega 2560 R3. Además, se utilizó un transformador para reducir de 220V AC a 24V AC y alimentar la electroválvula. El cargador se conectó a los pines GND y VCC, mientras que el terminal IN2 se asignó al pin 31 para la señal de control.

Figura 10

Comunicación entre el Módulo Relé de 4 canales y la placa Arduino



2.2.5. Calibración y ajustes de los sensores para una correcta lectura

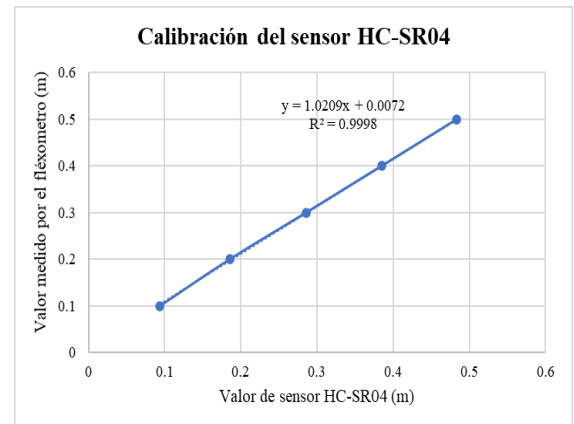
a) Sensor de ultrasonido HC-SR04

Para calibrar el sensor ultrasónico HC-SR04, se realizaron lecturas en un punto de referencia y se compararon con mediciones reales obtenidas mediante un

flexómetro. Este proceso se llevó a cabo antes de la instalación en el reservorio.

Figura 11

Calibración del sensor de ultrasonido HC-SR04



b) Sensor de presión HK- 3025 5MPA

Para calibrar el sensor de presión, se realizaron lecturas antes y después del filtro utilizando un manómetro digital.

Figura 12

Calibración del sensor de presión Hk-3025 antes del filtro

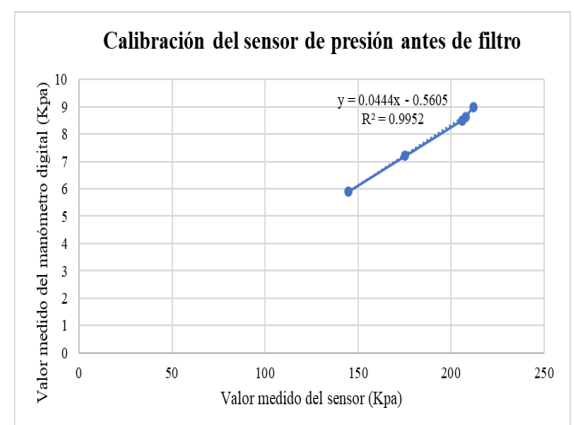
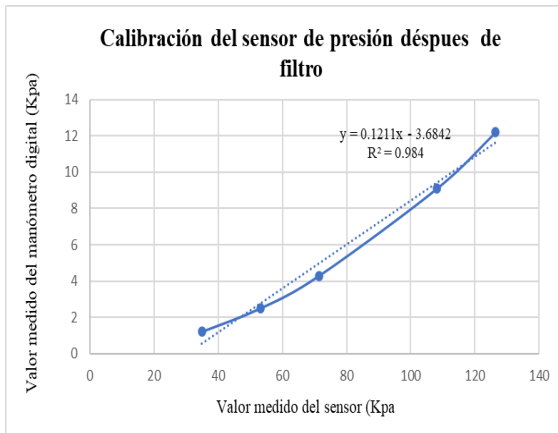


Figura 13

Calibración del sensor de presión Hk-3025 después del filtro

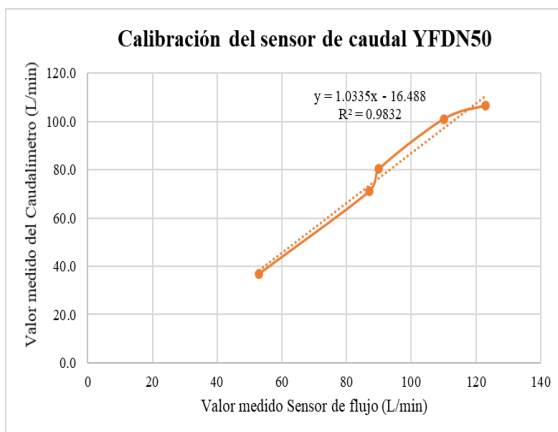


c) Sensor de flujo de agua 2" YF-DN50

Para la calibración del flujo de agua fue necesario el uso de un caudalímetro.

Figura 14

Calibración del sensor de flujo de agua 2" YF-DN50

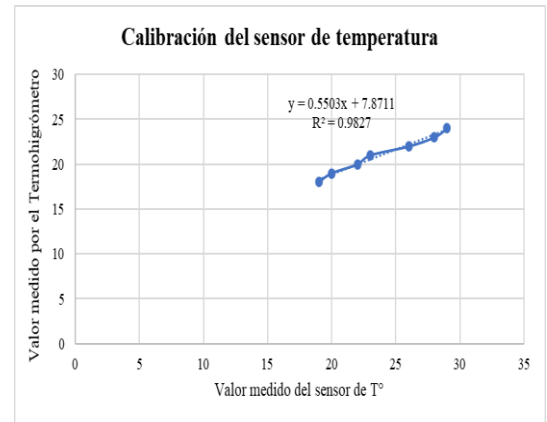


d) Sensor DHT11 de temperatura

Para la calibración del sensor DHT11 de temperatura se realizó haciendo uso de con un termohigrómetro.

Figura 15

Calibración del sensor DHT11 de temperatura

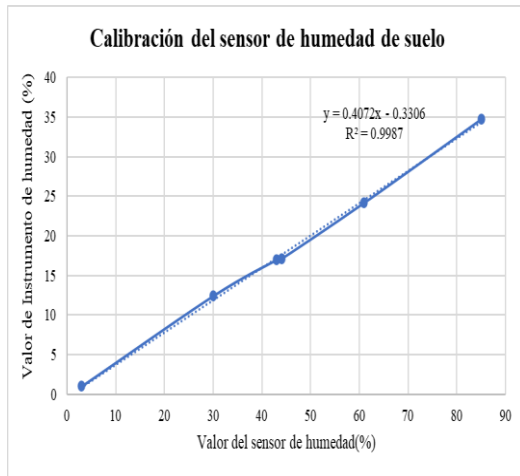


e) Módulo higrómetro sensor de humedad del suelo

Para la calibración del sensor de humedad se tuvo que instalar a 10cm de profundidad del suelo; este sensor nos da las lecturas ya convertidas en porcentaje, la cual varía según la cantidad de humedad que existe en el suelo, cabe mencionar que también se sacó muestras del suelo; estas muestras fueron llevados al laboratorio para conocer la humedad gravimétrica de suelo, Además, se empleó un medidor de humedad del suelo para realizar mediciones *in situ*.

Figura 16

Calibración del sensor de humedad



2.2.6. Creación de un aplicativo en la plataforma Blynk para tener lecturas en tiempo real.

La plataforma Blynk fue elegida para la monitorización en tiempo real por su simplicidad, flexibilidad y compatibilidad con diversos dispositivos. Su facilidad de uso y conexión a la nube sin costos adicionales la hacen ideal para proyectos IoT de cualquier escala.

Para el monitoreo en tiempo real del sistema de riego, se diseñaron widgets personalizados en la aplicación Blynk, lo cual es clave para tomar decisiones rápidas y efectivas, la actualización de la aplicación se da cada 20 minutos.

Figura 17

Widgets personalizados en la aplicación Blynk



III. RESULTADOS

3.1. Análisis de los componentes de la Automatización para el riego por goteo

Los componentes utilizados en la automatización del sistema de riego por goteo permitieron comprender la función específica y esencial de cada elemento en el correcto desempeño del sistema. Se integraron sensores de alta precisión como el HC-SR04 para controlar el nivel del reservorio, el Hk3025 para mantener limpio el anillo de filtro, y el YF-DN50 para monitorear el caudal de agua. Además, el sensor DHT11 y los sensores de humedad del suelo proporcionaron datos ambientales clave. El módulo relé

de 4 canales controló dispositivos de mayor potencia, como la electroválvula, mientras que la placa Arduino Mega 2560 R3 con el Ethernet Shield W5100 permitió el monitoreo remoto. Complementaron el sistema instrumentos como el manómetro digital, el caudalímetro y medidores de humedad, asegurando una gestión eficiente y precisa del riego.

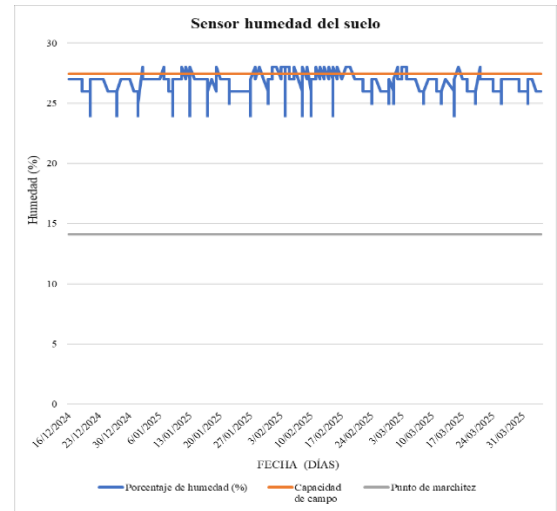
3.2. Desarrollo de un sistema efectivo de comunicación entre el sensor y el microcontrolador Arduino.

Se logró con éxito el desarrollo de un sistema de comunicación eficaz entre los sensores y el microcontrolador Arduino, lo que permitió una interacción precisa y eficiente entre ambos componentes. Gracias a esta integración, fue posible obtener los siguientes datos recolectados

En cuanto al sensor de humedad, cabe mencionar que se logró hacer el monitoreo de la cantidad de humedad en el suelo, la cual tuvo valor mínimo de 24 % y un valor máximo de 28 %; cuyos valores se encontraron cercanos de la capacidad de campo, lo que favoreció al buen desarrollo y rendimiento del cultivo de la zanahoria.

Figura 18

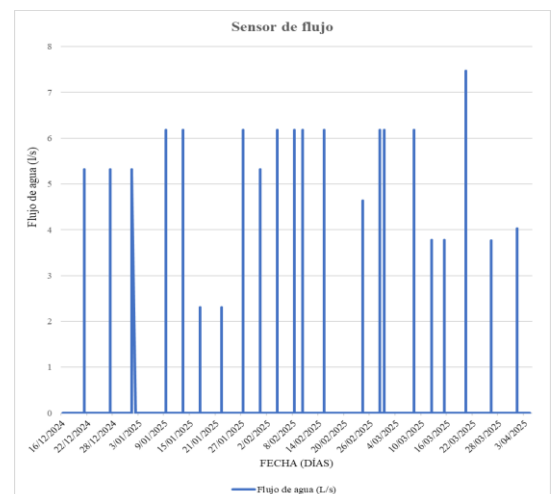
Datos recolectados del sensor de humedad



El sensor YF-DN50 funcionó con un rotor que giraba al pasar el agua. Este rotor tenía paletas magnéticas que, al moverse, generaban pulsos detectados por un sensor de efecto Hall. Luego, esos pulsos eran enviados a un Arduino, que los convertía en una medición del flujo de agua expresada en litros/min. Durante el funcionamiento del sistema de riego, se registró un flujo promedio de 4.5 l/s.

Figura 19

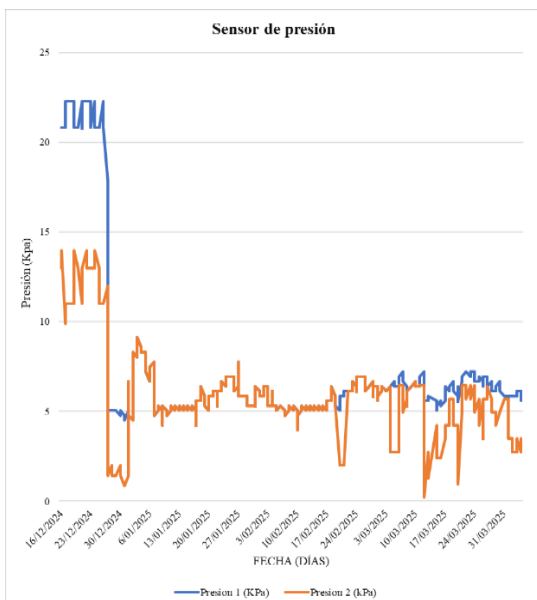
Datos recolectados del sensor de flujo



La presión del agua fue monitoreada con sensores HK3025. Se consideró que el filtro estaba en buen estado mientras la diferencia de presión no superara el 50% de la presión de entrada; de lo contrario, indicaba la necesidad de limpieza. Por esta razón, el filtro se limpió en promedio 10 veces.

Figura 20

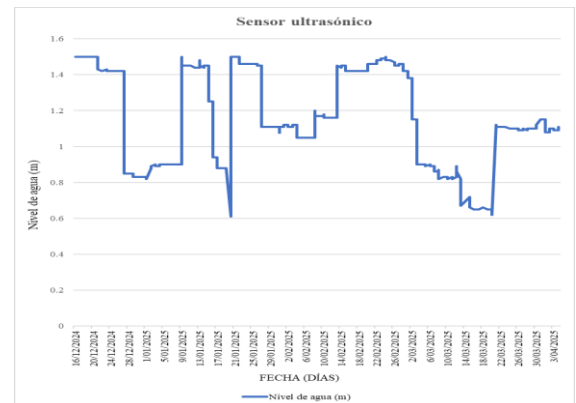
Datos recolectados del sensor de presiones



El nivel de agua en el reservorio fue monitoreado de forma eficiente utilizando el sensor ultrasónico HC-SR04. Este enviaba lecturas al microcontrolador Arduino Mega 2560, lo que permitió un control preciso del almacenamiento. Se mantuvo un nivel mínimo de 0.6 m para asegurar la disponibilidad de agua cuando fuera necesario para el riego.

Figura 21

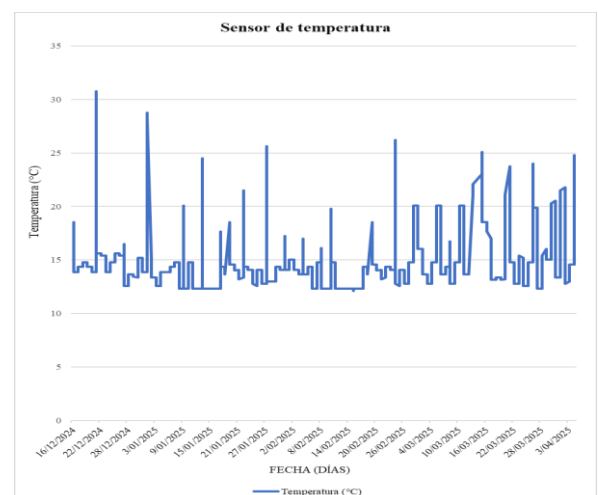
Datos recolectados del sensor ultrasónico



La temperatura dentro del invernadero se registró de manera efectiva mediante el sensor, el cual proporcionó datos precisos y continuos durante todo el periodo del cultivo. Esto permitió monitorear las variaciones térmicas en el interior del invernadero y evidenciar la influencia directa de la temperatura en la evapotranspiración. A mayor temperatura, mayor fue la necesidad de riego. La temperatura más alta registrada fue de 30.76 °C y la más baja de 12.11 °C.

Figura 22

Datos recolectados del sensor de temperatura



3.3. Desarrollo del código fuente para automatizar el riego por goteo empleando la tecnología del Arduino.

Como parte de la implementación, se desarrolló un código en la plataforma Arduino para recopilar y procesar los datos de los sensores y el actuador, almacenándolos posteriormente en una tarjeta Micro SD. Gracias a este código, el sistema opera de manera autónoma, asegurando que las zanahorias reciban siempre la cantidad de agua necesaria para su crecimiento.

Figura 23

Código fuente de la automatización

```
// ----- Definiciones -----
// Sensor de humedad
#define pinHumedad1 A3 // Pin analógico para el sensor de humedad 1
#define pinHumedad2 A4 // Pin analógico para el sensor de humedad 2
const int valorAire1 = 1000;
const int valorAire2 = 1000;
const int valorAire1 = 100;
const int valorAire2 = 100;
const int limiteMinimoHumedad = 20;
const int limiteMaximoHumedad = 25;
int humedad1, humedad2; ///PONER SI HUMEDAD <= 0% == 0%
int porcentajeHumedad1, porcentajeHumedad2, promedioPorcentajeHumedad;

// Sensor ultrasónico
#define trigPin 6 // Pin Trigger
#define echoPin 7 // Pin Echo
const float velocidadSonido = 0.000343; // Velocidad del sonido en
343m/s = 0.000343 m/us
float distancia;
float nivelAgua;
float volumenReservorio;
const float nivelTotalPozo = 1.5; //m
const float nivelCriticoAgua = 0.6; //m
const float AreaBaseMinima = 8.; //m2
const float AreaBaseMaxima = 24.; //m2

// Relés
#define RELE_ON 0
#define RELE_OFF 1
#define rele_2 31
int estadoValvula = 0;

// Sensor de flujo de agua
#define pinFlujoAgua 3
volatile int conteoPulsos = 0; // Número de pulsos
float frecuencia, flujoLitrosMinuto;
const float factorConversion = 0.2; // Pulsos por litro (según sensor)
unsigned long tiempoAnterior = 0;

// Sensor de presión
#define pinPresion1 A1 // Pin analógico para el sensor de presión
1
#define pinPresion2 A2 // Pin analógico para el sensor de presión 2
const float voltajeMin = 0.5; // Voltaje a presión mínima (0 MPa)
const float voltajeMax = 4.5; // Voltaje a presión máxima (5.0 MPa)
const float presionMax = 5.0; // Presión máxima en MPa
float presion1=0.0, presion2=0.0;
int estadoFiltro;

//Sensor de humedad
#include <DHT.h>
#define DHTPIN 5 // Pin al que está conectado el sensor
#define DHTTYPE DHT11 // Tipo de sensor (DHT11)
DHT dht(DHTPIN, DHTTYPE);
```

```
float humedadAmbiente, temperatura;

//SD
#include <SPI.h>
#include <SD.h>
#define chipSelect 4 //CS
#define pinBoton 8 // Pin donde conecta el botón para expulsar SD
#define pinLed 9 // Pin donde conecta el LED
#define pinLed2 2 // Pin donde conecta el LED
File archivo;

//Ethernet Shield
#include <Ethernet.h>
#include <TimeLib.h> // Biblioteca Time para manejar la conversión del
tiempo
String fecha = ""; // Variable para la fecha
String hora = ""; // Variable para la hora

// Configuración de Ethernet
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Dirección MAC
única
IPAddress ip(192, 168, 1, 102); // Cambia según tu red

//BLYNK
#define BLYNK_TEMPLATE_ID "TMPL2-DNrZ08"
#define BLYNK_TEMPLATE_NAME "Riego Automatizado"
#define BLYNK_AUTH_TOKEN "W_TWK1q_CwNmifBnrQKRLGR700sbW0R"
#define BLYNK_PRINT Serial
#include <BlynkSimpleEthernet.h>
#include <WidgetRTC.h>
WidgetRTC rtc; //Para capturar fecha y hora de servidor Blynk

// Variable para contar iteraciones del bucle principal
int contadorEnvioBlynk= 38; //38 Para que envíe a Blynk pasado 1 minuto
la primera vez que se enciende, luego envía cada 20 min

//REINICIAR ARDUINO
#include <EEPROM.h> // Librería para manejar la EEPROM
#include <avr/wdt.h> // Librería para el Watchdog Timer
#define ADDR_CONTADOR 0 // Dirección en EEPROM para almacenar el
contador

void reinicioForzoso() {
Serial.println("Reiniciando...");
int contadorReinicioForzoso = EEPROM.read(ADDR_CONTADOR);
if (contadorReinicioForzoso >= 12) {
EEPROM.write(ADDR_CONTADOR, 0); // Reiniciar el contador
while (true) {
delay(1000);
}
}
// Incrementar el contador
contadorReinicioForzoso++;

EEPROM.update(ADDR_CONTADOR, contadorReinicioForzoso); // Guardar en
EEPROM

// Forzar el reinicio
delay(20000); // Esperar 20 segundos antes del reinicio
wdt_enable(WDTO_15MS); // Activar el watchdog con un tiempo de
espera corto
while (true){ // Bucle infinito para forzar el reinicio
}

// ----- Configuración -----
void setup() {
// Comunicación serial
Serial.begin(9600);
Serial.println("Sistema iniciado.");
delay(1000);
// Inicializar SD
int contadorReinicioSD=0;
while (!inicializarSD()) {
if (contadorReinicioSD==12){
reinicioForzoso();
}
delay(9000);
reinitializarModuloSD();
contadorReinicioSD++;
}
Serial.println("SD iniciado.");

delay(60000); //delay de 1 minuto, esperando que el wifi se conecte
correctamente ante un apagón
//Shield Ethernet
// Inicializar Ethernet
if (Ethernet.begin(mac) == 0) {
Serial.println("Error al iniciar Ethernet. Verifique la conexión.");
Ethernet.begin(mac, ip);
if (Ethernet.linkStatus() == LinkOFF) {
Serial.println("El cable Ethernet no está conectado.");
reinicioForzoso();
}
}
Serial.println("Ethernet iniciado.");

// Conexión a Blynk
delay(1000);
Serial.println("Conectando a Blynk...");
Blynk.begin(BLYNK_AUTH_TOKEN);
Serial.println("Conectado a Blynk!");
// Sincronizar RTC de Blynk
delay(1000);
rtc.begin();

// Inicializar el sensor DHT11
delay(1000);
dht.begin();
```

```

// Configuración sensores de humedad
pinMode(pinHumedad1, INPUT);
pinMode(pinHumedad2, INPUT);

// Configuración sensor ultrasónico
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

// Configuración relés
pinMode(relé_2, OUTPUT);
digitalWrite(relé_2, HIGH); // Iniciar relé apagado (relé activado por LOW)

// Configuración sensor de flujo de agua
pinMode(pinFlujoAgua, INPUT_PULLUP);
// Se configura una interrupción para registrar cada pulso
(contarPulsos) generado por el sensor (pin 3)
attachInterrupt(digitalPinToInterrupt(pinFlujoAgua), contarPulsos, RISING);

// Configuración sensores de presión
pinMode(pinPresion1, INPUT);
pinMode(pinPresion2, INPUT);

// Para expulsar SD
pinMode(pinBoton, INPUT_PULLUP); // Configura el botón con resistencia pull-up
pinMode(pinLed, OUTPUT); // Configura el LED como salida
digitalWrite(pinLed, LOW); // LED apagado inicialmente
pinMode(pinLed2, OUTPUT);
digitalWrite(pinLed2, LOW);

// Apagar Relé
digitalWrite(relé_2, HIGH); // Apaga el relé al establecer la conexión
}

// ----- Funciones -----
// Sensor humedad suelo
void controlarHumedad() {
  humedad1 = analogRead(pinHumedad1);
  humedad2 = analogRead(pinHumedad2);
  porcentajeHumedad1 = map(humedad1, valorAire1, valorAgua1, 0, 100);
  porcentajeHumedad1 = (0.4072*porcentajeHumedad1)-0.3306;
  porcentajeHumedad2 = map(humedad2, valorAire2, valorAgua2, 0, 100);
  porcentajeHumedad2 = (0.4072*porcentajeHumedad2)-0.3306;
  promedioPorcentajeHumedad = (porcentajeHumedad1 + porcentajeHumedad2) / 2;

  Serial.print("Humedad 1: ");
  Serial.print(humedad1);
  Serial.print(", Humedad 2: ");
  Serial.print(humedad2);
  Serial.print(", Humedad 1: ");

  Serial.print(porcentajeHumedad1);
  Serial.print("%, Humedad 2: ");
  Serial.print(porcentajeHumedad2);
  Serial.print("%, Promedio: ");
  Serial.print(promedioPorcentajeHumedad);
  Serial.println("%");
}

// Sensor ultrasónico
void medirDistancia() {
  long duracion;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duracion = pulseIn(echoPin, HIGH);
  distancia = (duracion * velocidadSonido) / (2);
  nivelAgua = nivelTotalPozo - (distancia-0.36); //total altura del pozo - lectura sensor-altura extra
  nivelAgua = (1.0209*nivelAgua)+0.0072;

  volumenReservorio = (nivelAgua / 3) *
    (AreaBaseMinima +
     AreaBaseMaxima * pow((nivelAgua / nivelTotalPozo),
2) +
     sqrt(AreaBaseMinima * AreaBaseMaxima *
pow((nivelAgua / nivelTotalPozo), 2)));

  Serial.print("NivelAgua: ");
  Serial.print(nivelAgua);
  Serial.println(" m");
  Serial.print("VolumenReservorio: ");
  Serial.print(volumenReservorio);
  Serial.println(" m3");
}

// Funciones de control de relés
int tiempoValvulaAbierta = 0;
int formaActivacion = 0; //0=automatico, 1>manual
void controlarRelés() {
  // Verificar nivel crítico de agua, aplica a ambos modos
  if (nivelAgua <= nivelCriticoAgua) {
    cerrarValvula();
    return;
  }

  // Verificar que no esté abierto por 2 min seguidos
  if (estadoValvula==2) {
    cerrarValvula();
    Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
    Blynk.virtualWrite(V3, 0);
    delay(600000); //600000 = 10 min
    reiniciarSistema();
    return;
  }

  //Modo automático
  if (formaActivacion==0) {
    if (promedioPorcentajeHumedad >= limiteMinimoHumedad &&
promedioPorcentajeHumedad <= limiteMaximoHumedad) {
      if (estadoValvula==0) { //Si el estado anterior era 0, significa que se acaba de activar. Solamente envia info a Blynk cuando cambia de estado, si estuvo en el mismo estado mucho tiempo, no envia nada.
        Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
        Blynk.virtualWrite(V3, 1);
      }
      abrirValvula();
    } else {
      if (estadoValvula==1) { //Si el estado anterior era 1, significa que se acaba de desactivarse, enviar a Blynk.
        Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
        Blynk.virtualWrite(V3, 0);
      }
      cerrarValvula();
    }
  }

  //Modo manual
  else if (formaActivacion == 1) {
    abrirValvula();
  }
}

void abrirValvula() {
  digitalWrite(relé_2, RELE_ON); //activarRelé
  estadoValvula = 1;
  tiempoValvulaAbierta++;

  // Verificar si la válvula lleva demasiado tiempo abierta
  if (tiempoValvulaAbierta >= 16) { //16 = 8 minutos ->
480'000segundos/30'000 (segundos loop). En caso la valvula se abra por más de 8 minutos seguidos, significa que hubo un problema.
  estadoValvula = 2; //Valvula estuvo abierta por mucho tiempo (anomalía)
}
}

void cerrarValvula() {
  digitalWrite(relé_2, RELE_OFF); //desactivarRelé
  estadoValvula = 0;

  tiempoValvulaAbierta = 0;
}

//Sensor flujo de agua
void medirFlujoAgua() {
  unsigned long tiempoActual = millis();
  // Calcular flujo cada segundo
  if (tiempoActual - tiempoAnterior >= 1000) {
    tiempoAnterior = tiempoActual;

    // Calcular la frecuencia
    noInterrupts(); // Detén las interrupciones temporalmente para leer el conteo
    frecuencia = conteoPulsos;
    conteoPulsos = 0; // Reiniciar conteo
    interrupts(); // Habilita las interrupciones nuevamente

    // Convertir frecuencia a caudal (L/min) - datasheet
    flujoLitrosMinuto = (frecuencia / factorConversion);
    flujoLitrosMinuto = (1.0335*flujoLitrosMinuto)-16.488;

    if (flujoLitrosMinuto<0){
      flujoLitrosMinuto=0.0;
    }
    // Mostrar datos
    Serial.print("Frecuencia: ");
    Serial.print(frecuencia);
    Serial.print(" Hz\tFlujo: ");
    Serial.print(flujoLitrosMinuto);
    Serial.println(" L/min ");
  }
}

// Función de interrupción para contar pulsos
void contarPulsos() {
  conteoPulsos++;
}

// Sensor de presión
void medirPresion() {
  // Sensor 1
  int valorAnalogico1 = analogRead(pinPresion1);
  float voltaje1 = (valorAnalogico1 * 5.0) / 1023.0; // Conversión de ADC a voltaje
  presion1 = ((voltaje1 - voltajeMin) / (voltajeMax - voltajeMin))*presionMax*1000.; //kPa
  presion1 = (0.0444*presion1)-0.5650;

  // Sensor 2
  int valorAnalogico2 = analogRead(pinPresion2);
  float voltaje2 = (valorAnalogico2 * 5.0) / 1023.0; // Conversión de ADC a voltaje
  voltaje2=voltaje2-0.065; //Calibración con voltaje1
}

```

```

presion2 = ((voltaje2 - voltajeMin) / (voltajeMax -
voltajeMin))*presionMax*1000.; //KPa
presion2= (0.1211*presion2)-3.6842;
if (presion1<presion2){
  presion2=presion1;
}

Serial.print("Presión voltaje 1: ");
Serial.println(voltaje1);
Serial.print("Presión voltaje 2: ");
Serial.println(voltaje2);

Serial.print("Presión Sensor 1: ");
Serial.print(presion1);
Serial.println(" KPa");

Serial.print("Presión Sensor 2: ");
Serial.print(presion2);
Serial.println(" KPa");

float diferenciaPresion = presion1-presion2;
if (diferenciaPresion<=(presion1/2)){ //si la diferencia es mayor al 50%
de la entrada
  estadoFiltro=1; //FILTRO OK
}else{
  estadoFiltro=0; //FILTRO SUCIO
}
}

// Sensor DHT11
void leerSensorDHT() {
  // Leer humedad ambiental
  humedadAmbiente = dht.readHumidity();
  // Leer temperatura en Celsius
  temperatura = dht.readTemperature();
  temperatura = (0.5503*temperatura)+7.8711;

  // Verificar si hubo error en la lectura
  if (isnan(humedadAmbiente)) {
    humedadAmbiente=-1.0;
  }
  if (isnan(temperatura)) {
    temperatura=-1.0;
  }
  // Mostrar valores en el monitor serial
  Serial.print("Humedad Ambiental: ");
  Serial.print(humedadAmbiente);
  Serial.print(" %\t");

  Serial.print("Temperatura Ambiental: ");
  Serial.print(temperatura);

  Serial.println(" °C");
}

// Función para inicializar SD y abrir archivo
bool inicializarSD() {
  if (!SD.begin(chipSelect)) {
    Serial.println("Error al inicializar la tarjeta SD.");
    return false;
  }

  archivo = SD.open("DATA.txt", FILE_WRITE);
  if (!archivo) {
    Serial.println("Error al abrir el archivo.");
    return false;
  }

  Serial.println("Tarjeta SD y archivo inicializados correctamente.");

  // Enviar al Pin Virtual
  Blynk.virtualWrite(V9, 1);
  return true;
}

void reinicializarModuloSD() {
  digitalWrite(chipSelect, HIGH); // Desactiva el CS
  delay(500); // Espera para estabilizar
  digitalWrite(chipSelect, LOW); // Activa el CS
  delay(500); // Espera para que el módulo se
estabilice
}

// Función para escribir datos en SD
void escribirEnSD(const String& fecha, const String& hora, float
nivelAgua, float volumenReservorio, float frecuencia, float
flujoLitrosMinuto, int porcentajeHumedad1, int porcentajeHumedad2, int
promedioPorcentajeHumedad, float presion1, float presion2, int
estadoFiltro, float humedadAmbiente, float temperatura, int
estadoValvula) { //&: referencia directa, sin copias, mayor eficiencia,
*puntero, permite almacenar cadenas de texto y no 1 letra. //int
estadoBomba
  if (archivo) {
    archivo.print(fecha); // Escribe la fecha y hora actual
    archivo.print(",");
    archivo.print(hora); // Escribe la fecha y hora actual
    archivo.print(",");
    archivo.print(nivelAgua);
    archivo.print(",");
    archivo.print(volumenReservorio);
    archivo.print(",");
    archivo.print(frecuencia);
    archivo.print(",");
    archivo.print(flujoLitrosMinuto);
    archivo.print(",");
    archivo.print(porcentajeHumedad1);
    archivo.print(",");
    archivo.print(porcentajeHumedad2);
    archivo.print(",");
    archivo.print(promedioPorcentajeHumedad);
    archivo.print(",");
    archivo.print(presion1);
    archivo.print(",");
    archivo.print(presion2);
    archivo.print(",");
    archivo.print(estadoFiltro);
    archivo.print(",");
    archivo.print(humedadAmbiente);
    archivo.print(",");
    archivo.print(temperatura);
    archivo.print(",");
    archivo.print(estadoValvula);
    archivo.println(); // Asegura que los datos se escriban inmediatamente

    digitalWrite(pinLed2, HIGH); // Encender el LED
    delay(1000);
    digitalWrite(pinLed2, LOW);
  } else {
    Serial.println("No se pudo escribir en el archivo.");
    Blynk.virtualWrite(V9, 0);
  }
}

//Función expulsar SD
void expulsarSD(){
  // Si el botón se presiona, cerrar el archivo y la SD
  if (digitalRead(pinBoton) == LOW) { // Botón presionado
    delay(1000);
    digitalWrite(pinLed, HIGH); // Encender el LED
    archivo.close();
    Serial.println("Archivo cerrado. Puede extraer la SD.");
    delay(3000);
    digitalWrite(pinLed, LOW);

    // Enviar al Pin Virtual
    Blynk.virtualWrite(V9, 0); //como la función se llama explícitamente
en el loop, entonces el valor es enviado a Blynk

    // Detener el programa para evitar escribir más
    while (true) {
      delay(1000);
    }
  }
}

// Función para sincronizar el RTC cuando Blynk se conecta
BLYNK_CONNECTED() {
  rtc.begin();
  Blynk.syncAll(); // Sincroniza los datos del servidor
  Serial.println("RTC sincronizado con el servidor de Blynk.");
}

}

// Actualizar Fecha y Hora desde RTC (Blynk)
void actualizarFechaHora() {
  time_t tiempoActual = now(); // Ajuste a UTC-5 - (5 * 3600)

  // Construir la fecha
  fecha = String(year(tiempoActual)) + "-" +
String(month(tiempoActual) < 10 ? "0" : "") +
String(month(tiempoActual)) + "-" +
String(day(tiempoActual) < 10 ? "0" : "") +
String(day(tiempoActual));

  // Construir la hora
  hora = String(hour(tiempoActual) < 10 ? "0" : "") +
String(hour(tiempoActual)) + ":" +
String(minute(tiempoActual) < 10 ? "0" : "") +
String(minute(tiempoActual)) + ":" +
String(second(tiempoActual) < 10 ? "0" : "") +
String(second(tiempoActual));
}

void reiniciarSistema(){
  //Expulsar SD
  delay(1000);
  digitalWrite(pinLed, HIGH); // Encender el LED
  archivo.close();
  Serial.println("Archivo cerrado. Puede extraer la SD.");
  delay(3000);
  digitalWrite(pinLed, LOW);
  Blynk.virtualWrite(V9, 0); //como la función se llama explícitamente
en el loop, entonces el valor es enviado a Blynk
  //REINICIAR
  Serial.println("Reiniciando Arduino...");
  delay(1000); // Esperar 1 segundo antes del reinicio
  wdt_enable(WDTO_15MS); // Activar el watchdog con un tiempo de
espera corto
  while (true) {} // Bucle infinito para forzar el reinicio
}

//REINICIAR ARDUINO DESDE BLYNK
BLYNK_WRITE(V2) {
  int botonReinicio = param.asInt(); // Obtiene el valor desde el widget
(0 o 1)
  if (botonReinicio == 1) { // Verifica si el botón virtual está
presionado
    reiniciarSistema();
  }
}

//Activar o desactivar valvula de forma manual
BLYNK_WRITE(V13) {
  int botonValvulaManual = param.asInt(); // Obtiene el valor desde el
widget (0 o 1)
}

```

```

tiempoValvulaAbierta = 0; // Reiniciar contador al activar o desactivar
modo manual
if (botonValvulaManual == 1) { // Modo Manual
  formaActivacion=1;
}else{
  formaActivacion=0;
}
}

// ----- Bucle principal -----
void loop() {
  expulsarSD();
  controlarHumedad(); // Leer y procesar humedad
  medirDistancia(); // Leer y procesar distancia
  controlarReles(); // Controlar los relés
  medirFlujoAgua(); // Leer y procesar flujo de agua
  medirPresion(); // Leer y procesar presión de agua3
  leerSensorDHT(); // Leer y procesar DHT11
  actualizarFechaHora();
  escribirEnSD(fecha, hora, nivelAgua, volumenReservorio,frecuencia,
  flujoLitrosMinuto, porcentajeHumedad1, porcentajeHumedad2,
  promedioPorcentajeHumedad, presion1, presion2,
  estadoFiltro,humedadAmbiente, temperatura, estadoValvula); //estadoBomba,

  contadorEnvioBlynk++;
  if(contadorEnvioBlynk >=40){ //40, cada 20 minutos enviar a la app
  Blynk
  Blynk.virtualWrite(V0, promedioPorcentajeHumedad);
  Blynk.virtualWrite(V1, nivelAgua);
  Blynk.virtualWrite(V11, volumenReservorio);
  Blynk.virtualWrite(V3, estadoValvula);
  Blynk.virtualWrite(V4, flujoLitrosMinuto);
  Blynk.virtualWrite(V5, presion1);
  Blynk.virtualWrite(V6, presion2);
  Blynk.virtualWrite(V12, estadoFiltro);
  Blynk.virtualWrite(V7, humedadAmbiente);
  Blynk.virtualWrite(V8, temperatura);
  Blynk.virtualWrite(V9, 1);
  String fechaHora = fecha + " " + hora;
  Blynk.virtualWrite(V10, fechaHora);
  contadorEnvioBlynk=0;
  }

  Blynk.run();

  delay(30000); //30000 = 30 segundos (blynk se desconecta cada 1
  minuto)
  Serial.println("-----");
}

```

3.4 Estimación de la cantidad de zanahorias que se cosecharon al finalizar la campaña agrícola.

Al concluir la campaña agrícola, se realizó la estimación de la producción total de zanahorias, tomando como referencia el peso promedio de las raíces comerciales por planta. El rendimiento estimado fue de 7 kg/m², considerando una superficie cultivada de 35 m². De esta manera, se obtuvo una producción total estimada de 245 kilogramos de zanahorias.

IV. DISCUSIÓN

4.1. Discusión sobre el análisis de los componentes de la Automatización para el riego por goteo

Tomando como base los resultados obtenidos se acepta la hipótesis específica 1 en base a los resultados obtenidos, de que los componentes de la automatización para el sistema de riego por goteo fueron de fácil instalación, el cual permitieron mejorar la eficiencia del invernadero de Pampa Hermosa-Lorensayoq, debido a que guarda una relación alta con lo que menciona Ludeña (2021) en su trabajo de investigación titulado "*Implementación y monitoreo con tecnología Arduino para automatización de riego tecnificado*", se evidenció que la adecuada selección de sensores y actuadores se integró exitosamente, logrando una programación eficiente para el sistema.

4.2. Discusión sobre el desarrollo de un sistema efectivo de comunicación entre el sensor y el microcontrolador Arduino.

A partir de los resultados obtenidos aceptamos la hipótesis específica 2 que establece que los sensores transmitieron la información al microcontrolador Arduino de manera rápida, lo que facilitó una comunicación efectiva permitiendo detectar señales digitales y convertirlas en magnitudes físicas.

3.3. Discusión sobre el desarrollo del código fuente para automatizar el riego por goteo empleando la tecnología del Arduino.

Con base en los resultados obtenidos, se acepta la hipótesis específica 3, la cual establece que el código fuente, desarrollado en Arduino IDE, se integró con éxito al microcontrolador Arduino. Esto permitió el correcto funcionamiento del sistema de automatización del riego, incluyendo el monitoreo eficiente del sistema. Por otra parte, estos resultados guardan una relación con lo expuesto por Huamán (2023) en su tesis titulada "*Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima*", donde se evidencia que la programación del código en el IDE de Arduino, mediante el uso de librerías, funciones y variables, permitió el funcionamiento de la automatización.

3.4 Discusión sobre la estimación de la cantidad de zanahorias que se cosecharon al finalizar la campaña agrícola.

Con los resultados obtenidos, se acepta la hipótesis 4, que plantea que la aplicación de prácticas agrícolas sostenibles, incluida la automatización del sistema de riego junto con

condiciones climáticas favorables, generó un impacto positivo en el rendimiento del cultivo de zanahoria Chantenay Red Core, alcanzando los 7 kg/m², siendo un rendimiento mayor al de Nina (2020) en su tesis: "*Evaluación de tres variedades de zanahoria (Daucus carota L.) en invernadero y a campo abierto en el municipio de la ciudad de La Paz*", en donde se demuestra que la mayor producción en rendimiento la obtuvo el tratamiento en la zanahoria chantenay en un invernadero que fue de 6,19 kg/m².

V. CONCLUSIONES

El análisis del sistema de riego automatizado permitió identificar la función específica de cada componente en su rendimiento global. Los sensores de humedad, flujo, presión, ultrasonido y temperatura, junto con el microcontrolador Arduino, la electroválvula y otros elementos, conforman una estructura integrada que asegura un riego preciso y adaptado a las condiciones del cultivo. Los sensores de humedad activan el riego solo cuando el suelo lo requiere, optimizando el uso del agua.

El sistema desarrollado permitió una comunicación eficiente y en tiempo real entre los sensores y el microcontrolador Arduino, asegurando

un control preciso del riego automatizado. Los niveles de humedad del suelo se mantuvieron dentro del rango óptimo (24 %–28 %). Los sensores de presión detectaron obstrucciones en el filtro y el sensor ultrasónico controló el nivel de agua. La temperatura en el invernadero osciló entre 13 y 18 °C, con picos superiores a 25 °C que disminuyeron la humedad del suelo. El sistema demostró efectividad en condiciones variables.

Se desarrolló un código funcional que automatiza el riego por goteo mediante Arduino, integrando sensores para una gestión eficiente. El sistema activa la electroválvula cuando el nivel de agua supera los 0.6 metros y la humedad del suelo está entre 20 % y 25 %. También se programaron los sensores de presión HK3025, flujo YF-DN50 y temperatura DHT11. Los datos registrados por cada uno de ellos se almacenan en una tarjeta SD y se visualizan en tiempo real mediante la aplicación Blynk.

La implementación del sistema incrementó el rendimiento del cultivo de zanahoria Chantenay Red Core a 7 kg/m², superando el promedio nacional de 2,52 kg/m² (MIDAGRI, 2020) y el rendimiento de 6,19 kg/m² reportado por Nina (2020). Los resultados confirman la

eficacia del sistema para optimizar la productividad agrícola.

VI. RECOMENDACIONES

Se recomienda emplear la tecnología Arduino para el riego, de modo que puedan controlar en tiempo real los distintos parámetros necesarios para lograr un rendimiento satisfactorio en el momento de la cosecha.

Antes de implementar el sistema automatizado por completo, realiza pruebas en el campo para ajustarlo a las condiciones locales.

Se recomienda el uso fuentes de energía tales como los paneles solares para que sea amigable con el medio ambiente.

Se debe elegir componentes que sean duraderos y fáciles de mantener en el caso del sensor de humedad, debido que en las áreas rurales el acceso a repuestos puede ser limitado.

También es importante ofrecer la opción de control manual para ajustes en situaciones de emergencia.

Lleva un registro detallado del crecimiento de las zanahorias y del consumo de agua durante la campaña. Este análisis permitirá evaluar el impacto

del riego automatizado y optimizar su eficiencia en futuras siembras.

Capacitar a los agricultores en el uso, mantenimiento y reparación de sistemas automatizados mediante talleres prácticos y asesoría técnica continua.

VII. REFERENCIAS BIBLIOGRÁFICAS

- Artero, O. T. (2013). *Arduino curso práctico de formación*. México: Alfaomega.
<https://www.latecnicalf.com.ar/descargas/material/proyectointegrador/Arduino%20Curso%20Pr%C3%A1ctico%20de%20formaci%C3%B3n.pdf>.
- Briones, G., & García I. (2007). *Sistema de riego por aspersión y goteo. 2ª ed.* México. Trillas UAAAN.
- Carit et al. (2002). Marcos A. Carit, Marvin Coto, Mario E. Fung. *Sistemas de control aplicados a la robótica*. Universidad de Costa Rica, Facultad de Ingeniería -Escuela de Ingeniería Eléctrica, Departamento de Automática (2002).
- Castañón, G. (1991). *Manual de Riego por Aspersión*. Mérida, Venezuela: CIDIAT.
- Cepeda et al. (2010). *Invernadero Inteligente basado en un Enfoque Sustentable para la Agricultura Mexicana*.
- Cervantes, W. R. (2016). *Diseño e implementación de sistema de riego automatizado y controlado por una placa Arduino para la finca*
Implementar sistemas de riego automatizado en cultivos de agricultura extensiva como una estrategia clave para optimizar el uso eficiente del agua, mejorar la productividad y fomentar la sostenibilidad del sistema agrícola.
- “LA-LUCIA”.CALCETA.
<http://repositorio.espam.edu.ec/handle/42000/301>
- Darshna et al. (2015). *Smart Irrigation System*, vol. 10.
<https://doi.org/10.9790/2834-10323236>
- De rosas, F. (2014). *Microcontroladores: Universidad Autónoma De Baja California - Facultad de Ciencias Químicas e Ingeniería*.
- Evans et al. (2013). *Adoption of site-specific variable rate sprinkler irrigation systems*. *Irrigation Science*, 31(4), 871-887.
<https://doi.org/10.1007/s00271-012-0365-x>
- Fuentes, J. (1999). *Técnicas de riego*. Ediciones Mundi-Prensa México.
- Gonzales, J. E. (2022). *Diseño y automatización vía remota de un sistema de riego por goteo para el cultivo de Paprika*.
<https://repositorio.lamolina.edu.pe/handle/20.500.12996/5654>
- Goyal, M. (2017). *Riego por goteo. Capítulo IV sistemas de riego*.
- Hargreaves, G. H., & Merkle G.P. (2000). *Fundamentos del Riego*. Estados Unidos. *Water Resources Publications*.

- Huamán, L. M. (2023). *Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima*. <https://repositorio.lamolina.edu.pe/handle/20.500.12996/5971>
- INIA. (2009). *Zanahoria INIA 101 Perú*. Ministerio de Agricultura: Instituto Nacional de Innovación Agraria.
- Jones, H. (2004). *Irrigation scheduling: Advantages and pitfalls of plant-based methods*. *Journal of Experimental Botany*, 55(407), 2427-2436. 55(407). <https://doi.org/10.1093/jxb/erh213>
- Liotta et al. (2015). *Manual de capacitación Riego por goteo. 1 er ed. Argentina*.
- Ludeña, J. (2021). *Implementación y monitoreo con tecnología Arduino para automatización de riego tecnificado en el Centro Experimental Canaán UNSCH - Ayacucho*.
- Maroto, J. (2008). *Elementos de Horticultura General, 3a*. Madrid: Mundi-Prensa. Mundi-Prensa.
- Martínez, L. (2001). *Manual de operación y mantención de equipos de riego presurizado*. Gobierno Regional de Atacama. Comisión Nacional de Riego e Instituto de Investigaciones Agropecuarias. Huasco, Chile: Vallenar.
- Naylamp Mechatronics. (2021). *Fuente de alimentación Conmutada 12v 20a*. Google. <https://naylampmechatronics.com/fuentes-switching-ac-dc/628-fuente-de-alimentacion-conmutada-acdc-250w-12v-20a-wode.html>
- Nina, D. (2020). *Evaluación de tres variedades de zanahoria (Daucus carota l.) en invernadero y a campo abierto en el municipio de la ciudad de La Paz*. Universidad Mayor San Andrés: Facultad de Agronomía.
- Osorio U., A. y T. C., F. (1999). *Elementos de riego tecnificado [en línea]*. La Serena: Disponible. <https://biblioteca.inia.cl/bitstream/handle/20.500.14001/28965/NR29540.pdf?sequence>
- Palomino K. (2009). *Riego por goteo*. Starbook. Editorial.
- Torrente, O. (2013). *ARDUINO: Curso práctico de formación (Primera ed)*. Alfaomega. Grupo Editor, S.A. De C.V., México.