

Área de Conocimiento de Tecnología de la  
Información y Comunicación

# **“Prototipo de sistema electrónico automatizado de invernadero dedicado a la producción de hortalizas basado en la tecnología de Internet de las cosas (IoT) con sistema de control de acceso”**

Trabajo Monográfico para optar al título de  
Ingeniero Electrónico

**Elaborado por:**

Br. Jhonatan David  
Martínez Calero  
Carnet: 2018-0353U

Br. Haminton Ecsequiel  
Blanco Bustillo  
Carnet: 2018-0057U

**Tutor:**

MSc. Fernando  
Antonio Flores Guido





Secretaría Académica  
DACTIC

**SECRETARIA DE ÁREA ACADÉMICA**

**F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA**

El Suscrito Secretario del **ÁREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

**MARTÍNEZ CALERO JHONATAN DAVID**

Carné: **2018-0353U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA ELECTRÓNICA**, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los trece días del mes de noviembre del año dos mil veinte y cinco.

**Atentamente,**



*Luisa Mercado*

\_\_\_\_\_  
MSc. Luisa Massiel Mercado Gutiérrez  
**SECRETARIO DE ÁREA ACADÉMICA**

📞 Móvil: (505) 83803517



Recinto Universitario Simón Bolívar  
Avenida Universitaria.  
Managua, Nicaragua.  
Apdo: 5595



Secretaria Académica  
DACTIC

**SECRETARIA DE ÁREA ACADÉMICA**

**F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA**


El Suscrito Secretario del **ÁREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

**BLANCO BUSTILLO HAMINTON ECSEQUIEL**

Carné: **2018-0057U** Turno: **Nocturno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA ELECTRÓNICA**, en el año 2022 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los trece días del mes de noviembre del año dos mil veinte y cinco.

Atentamente,

  
MSc. Luisa Massiel Mercado Gutiérrez  
SECRETARIO DE ÁREA ACADÉMICA



📞 Móvil: (505) 83803517



Recinto Universitario Simón Bolívar  
Avenida Universitaria.  
Managua, Nicaragua.  
Apdo: 5595

Managua, 12 de enero de 2026

**MSc. Claudia Lucía Benavidez Rugama**

**Directora DAC-TIC**

**Sus manos.**

Estimada Maestra Benavidez:

Reciba cordiales saludos y deseos de éxitos en el desempeño de sus funciones.

El motivo de la presente es para solicitarle sus atentas gestiones para llevar a cabo el proceso de defensa para la monografía que actualmente estoy como tutor con el tema: **“Prototipo de sistema electrónico automatizado de invernadero dedicado a la producción de hortalizas basado en la tecnología del Internet de las Cosas (IoT) con sistema de control de acceso.”**, presentado por los Bachilleres:

- Br. Haminton Ecsequiel Blanco Bustillo      Carné: 2018 – 0057U
- Br. Jhonatan David Martínez Calero      Carné: 2018 – 0353U

Según la revisión del informe final, los bachilleres se encuentran listos para la presentación de su trabajo en el proceso de defensa.

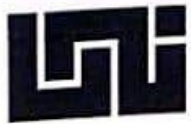
Agradezco de antemano su atención a la presente comunicación y me despido enviándole un cordial saludo.

Atentamente,

  
MSc. Fernando Flores Guido

Tutor del trabajo monográfico

Cc.: Archivo



Managua, 20 de febrero 2025

**Br. Jhonatan David Martínez Calero**      **2018-0353U**  
**Br. Haminton Ecsequiel Blanco Busto**      **2018-0057U**  
Egresados Programa académico Ingeniería Electrónica  
Sus manos.-

Estimados Egresados:

Reciban cordiales saludos y éxito en sus actividades.

Por medio de la presente, les comunico la aprobación del Protocolo de trabajo monográfico titulado: **"Prototipo de sistema electrónico automatizado de invernadero dedicado a la producción de hortalizas basado en la tecnología de Internet de las cosas (IoT) con sistema de control de acceso"**, el cual cumple con los requisitos establecidos en el capítulo II de la normativa de trabajos monográficos de la UNI como forma de culminación de estudios, por lo que queda oficialmente aprobado por esta Dirección.

El docente responsable de acompañarle en el proceso de desarrollo de su tema monográfico es el MSc. Fernando Antonio Flores Guido.

A partir de la fecha de aprobación del protocolo monográfico, tienen un máximo de doce meses para presentar los documentos correspondientes para la coordinación del proceso de pre defensa.

Atentamente,

**Msc. Claudia Lucía Benavidez Rugama**  
**Directora Área de Conocimiento de**  
**Tecnología de la Información y Comunicación**



Cc. MSc. Fernando Antonio Flores Guido – Tutor  
MSc. Hazzely Orozco Miranda - Secretaria Académica.  
Archivo DACTIC.

☎ Móvil: (505) 8889 0903

📍 Recinto Universitario Simón Bolívar  
Avenida Universitaria,  
Managua, Nicaragua.  
Apdo: 5595

✉ claudia.benavidez@dactic.uni.edu.ni  
www.uni.edu.ni

# **DEDICATORIA**

## **A Dios nuestro creador:**

Antes que todo, dedicamos este logro al omnipotente y todo poderoso Dios creador, porque de él procede nuestra vida, fortaleza, salud, sabiduría, inteligencia y amor.

## **A nuestras familias amadas:**

Nuestros padres, madres y hermanos por brindarnos su apoyo absoluto, confianza y comprensión en nuestra formación, nuestros logros y en el curso de este trabajo monográfico de nuestra carrera.

**Br. Haminton Ecsequiél Blanco Bustillo**

**Br. Jhonatan David Martínez Calero**

# **AGRADECIMIENTOS**

## **A Dios nuestro proveedor:**

Primeramente, agradecemos infinitamente a nuestro Dios, dador de vida, quien nos dotó de habilidades, capacidades y actitudes para finalizar de manera excelente nuestros estudios de ingeniería.

## **A nuestras familias queridas:**

Por su apoyo incondicional, cariño, amor, paciencia y responsabilidad. Muchas gracias por estar presente siempre a nuestro lado y ser una base principal en nuestra formación como ingenieros.

## **Al Ing. Fernando Antonio Flores Guido:**

Muchas gracias por su confianza, comprensión y esfuerzo por la tutoría de nuestro trabajo monográfico, aportándonos sus valiosas indicaciones y recomendaciones para la realización del mismo.

**Br. Haminton Ecsequiel Blanco Bustillo**

**Br. Jhonatan David Martínez Calero**

## **RESUMEN**

El presente trabajo monográfico tiene como objetivo desarrollar un prototipo de sistema de control y monitoreo para invernaderos, con el fin de asegurar condiciones ambientales óptimas en su interior, tales como la temperatura del suelo, la humedad relativa y el equilibrio higrométrico, necesarios para el cultivo eficiente del tomate.

En Nicaragua, uno de los principales obstáculos para el cultivo de hortalizas radica en la falta de condiciones climáticas adecuadas. En muchas ocasiones, las temperaturas alcanzan niveles muy altos, acompañadas de bajos niveles de humedad relativa. Además, otro problema recurrente es la falta de monitoreo y control adecuados durante el mantenimiento del cultivo, lo que provoca encharcamientos, daños en las raíces de las plantas, proliferación de hongos y un uso ineficiente de los recursos hídricos.

Este documento propone una solución alternativa mediante el diseño y desarrollo de un prototipo de sistema de riego automatizado y monitoreo ambiental, que garantice condiciones óptimas durante las diferentes etapas de crecimiento del cultivo. El sistema incluirá un mecanismo de control de acceso mediante tarjeta de identificación única, así como la posibilidad de gestionar y visualizar datos de sensores y actuadores a través de un panel web y una aplicación móvil.

## CONTENIDO

I.	INTRODUCCIÓN.....	1
II.	JUSTIFICACIÓN.....	2
III.	OBJETIVOS.....	4
3.1.	OBJETIVO GENERAL.....	4
3.2.	OBJETIVOS ESPECÍFICOS.....	4
IV.	MARCO TEÓRICO.....	5
	CAPÍTULO I.....	5
1.1.	INVERNADERO.....	5
1.2.	TIPOS DE INVERNADEROS.....	5
1.3.	EL TOMATE Y SU CULTIVO BAJO INVERNADERO.....	6
	CAPÍTULO II.....	9
2.1.	VARIABLES FÍSICAS DE INTERÉS EN UN INVERNADERO.....	9
2.1.2.	LOS CUATRO FACTORES DEL CONTROL AMBIENTAL DE LOS INVERNADEROS.....	9
2.1.3.	SISTEMA DE RIEGO.....	12
2.1.4.	RIEGO POR GOTEO.....	12
2.1.5.	REQUISITOS DE AGUA EN EL TOMATE DE INDUSTRIA.....	13
	CAPÍTULO III.....	14
3.1.	SISTEMA DE CONTROL.....	14
3.1.1.	¿CÓMO SELECCIONAR LOS COMPONENTES ELECTRÓNICOS?.....	14
3.1.2.	¿QUÉ TARJETA DE DESARROLLO ELEGIR?.....	14
3.1.3.	PLACAS DE DESARROLLO PARA PROYECTOS BASADOS EN INTERNET DE LAS COSAS.....	15
3.1.4.	SENSORES.....	18
3.1.5.	SENSOR DE HUMEDAD HIGRÓMETRO.....	22

3.1.6.	RESISTENCIA DEPENDIENTE DE LA LUZ LDR: FOTORRESISTENCIA	25
3.1.7.	SENSOR ULTRASÓNICO .....	27
3.1.8.	ACTUADORES .....	31
3.1.9.	OTROS COMPONENTES .....	34
	CAPÍTULO IV .....	41
4.	CONECTIVIDAD WiFi.....	41
4.1.	COMUNICACIONES POR INTERNET .....	41
4.2.	CONEXIÓN MÓDULO WiFi.....	41
4.3.	SERVIDOR WEB .....	42
4.3.1.	PROTOCOLO DE APLICACIÓN MQTT .....	42
4.4.	PROGRAMACIÓN INALÁMBRICA ESP32 (OTA): ACTUALIZADOR WEB ARDUINO IDE .....	50
4.5.	FREERTOS Y ESP32 .....	53
	CAPÍTULO V .....	56
5.1.	DISEÑO DEL SISTEMA – VISIÓN GLOBAL DEL HARDWARE .....	56
5.2.	EI SOFTWARE - VISIÓN GENERAL .....	60
5.2.1.	COMUNICACIÓN MQTT CON SERVIDOR.....	73
V.	CONCLUSIONES.....	91
VI.	RECOMENDACIONES.....	92
VII.	BIBLIOGRAFÍA .....	93
VIII.	ANEXOS .....	i

## ÍNDICE DE FIGURAS

Figura 1. Tipos de invernaderos [6].....	6
Figura 2. Invernadero totalmente equipado que controla completamente las condiciones de crecimiento [6].....	6
Figura 3. Parámetros ambientales que influyen en el crecimiento del tomate [8].....	11
Figura 4. Invernadero para el cultivo del tomate [9]. .....	12
Figura 5. ESP32 frente a ESP8266 [13]. .....	15
Figura 6. ESP32 DEVKIT V1 DOIT [13]. .....	16
Figura 7. Diagrama de bloque de funcionamiento del ESP32 [14]. .....	17
Figura 8. Sensor DHT11 Module [16].....	19
Figura 9. Diagrama del circuito del módulo DHT11 [16]. .....	20
Figura 10. Funcionamiento del sensor DHT [16]. .....	20
Figura 11. Ilustración 16 - Pines del sensor DHT22 [16]. .....	22
Figura 12. Sensor de humedad del suelo FC-28 [17].....	23
Figura 13. Sensor de humedad del suelo Higrómetro [17]. .....	24
Figura 14. Resistencia típica dependiente de luz con plomo [18]. .....	25
Figura 15. Estructura del fotorresistor [18].....	26
Figura 16. Circuito LDR básico usando un divisor de potencial [18]. .....	27
Figura 17. voltaje de salida usando la fórmula [18].....	27
Figura 18. Sensor ultrasónico de distancia [19].....	29
Figura 19. Pines del sensor de ultrasonidos [19]. .....	30
Figura 20. Principio de funcionamiento del sensor ultrasónico [19]. .....	30
Figura 21. Diagrama de temporización del módulo [19]. .....	30
Figura 22. Medición del nivel de agua del tanque [20]. .....	31
Figura 23. Funcionamiento básico de un actuador [21].....	32

Figura 24.	Bomba de agua 12V DC [22].	33
Figura 25.	Módulo adaptador LCD a I2C [23].	34
Figura 26.	Módulo adaptador LCD a I2C [56]	34
Figura 27.	Módulo adaptador LCD a I2C [23].	35
Figura 28.	Módulo de dos relés [25].	36
Figura 29.	Pines de un relé [25].	37
Figura 30.	Módulo RFID RC522 [61]	37
Figura 31.	Lector RFID [26].	39
Figura 32.	Acceso mediante aproximación al lector RFID [26].	40
Figura 33.	Gestor de Librerías IDE de Arduino. Elaboración propia.	42
Figura 34.	Protocolo de comunicación MQTT – Esquema ilustrativo [30].	45
Figura 35.	Funcionamiento MQTT [30].	46
Figura 36.	Suscripción MQTT [30].	47
Figura 37.	MQTT es un protocolo pensado para IoT que está al mismo nivel que HTTP o CoAP [30].	48
Figura 38.	Suscripción y publicación de datos [30].	48
Figura 39.	Broker encargado de gestionar la red y transmitir los mensajes [30].	49
Figura 40.	Visión general de la arquitectura OTA que muestra un proceso de ejemplo para actualizar el firmware de la aplicación sobre el terreno en los dispositivos desplegados [32].	51
Figura 41.	Organización FreeRTOS [33].	54
Figura 42.	Sistema operativo RTOS [34].	55
Figura 43.	Diagrama de bloques del sistema. Elaboración Propia.	56
Figura 44.	Diagrama esquemático de conexiones del sistema. Elaboración Propia.	59
Figura 45.	Diagrama de conexiones del sistema (vista protoboard). Elaboración Propia.	59

Figura 46.	Visión general del software. Elaboración propia.....	60
Figura 47.	Plataforma Adafuit. Elaboración propia. ....	75
Figura 48.	Paquete de uso gratuito del Servidor Adafruit IO. Elaboración propia.....	76
Figura 49.	Configuración tradicional de las credenciales WIFI. Elaboración propia.....	76
Figura 50.	Nombre visible del Access Point. Elaboración propia. ....	77
Figura 51.	Página Web de WiFiManager – Menú principal. Elaboración propia. ....	78
Figura 52.	Página Web de WiFiManager – Configuración de credenciales WIFI. Elaboración propia.....	78
Figura 53.	Página Web de actualizaciones OTA – Inicio de sesión. Elaboración propia. 80	
Figura 54.	Página Web de actualizaciones OTA – Selección de archivo. Elaboración propia. 80	
Figura 55.	Página Web de actualizaciones OTA – actualizando firmware de la ESP32. Elaboración propia.....	81
Figura 56.	Plataforma MQTT de Adafruit – Feeds. Elaboración propia. ....	82
Figura 57.	Plataforma MQTT de Adafruit – Feeds. Elaboración propia. ....	83
Figura 58.	Dashboard construido en interfaz web de Adafruit. Elaboración propia. ....	83
Figura 59.	Dashboard construido en interfaz web de Adafruit. Elaboración propia. ....	84
Figura 60.	App MQTT Dash. Elaboración propia. ....	85
Figura 61.	Pruebas de funcionamiento del prototipo construido. Elaboración propia.....	86
Figura 62.	Pruebas de funcionamiento del prototipo construido. Elaboración propia.....	86
Figura 63.	Pruebas de funcionamiento del prototipo construido. Elaboración propia.....	87
Figura 64.	Pruebas de funcionamiento prototipo construido. Elaboración propia. ....	87
Figura 65.	Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia. 88	

Figura 66.	Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.	88
Figura 67.	Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.	89
Figura 68.	Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.	89
Figura 69.	Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.	90
Figura 70.	Pruebas de funcionamiento de app MQTT Dash. Elaboración propia. ....	90

## ÍNDICE DE TABLAS

Tabla 1	Horas de riego por semana requeridas para aplicar 68 onzas de agua por planta de tomate por día según las poblaciones de plantas variables y las tasas de flujo del tubo de goteo [9]. ....	13
Tabla 2	Alimentación y modos de consumo, ESP8266 versus ESP32 [14]....	18
Tabla 3	Características y comparativa sensores DHT [16]]. ....	21
Tabla 4	Valores del sensor analógico para las diferentes combinaciones [17] .....24	
Tabla 5	Descripción de pines del módulo de relé de un solo canal [25] .....	24
Tabla 6	Configuración de pines RC522 [26] .....	24

## ANEXOS

ANEXO 1.	Placa PCB del Sistema en 2D.....	i
ANEXO 2.	Placa PCB del Sistema en 3D.....	i
ANEXO 3.	Estimación de costos del sistema .....	ii

## **I. INTRODUCCIÓN**

En las últimas décadas, ha habido un rápido avance en los sistemas agrícolas inteligentes. En el pasado, los sistemas de riego solían depender de molinos para regar las fincas mediante métodos convencionales, sin garantizar una adecuada cantidad de agua para los cultivos. Estos sistemas precursores sentaron las bases para automatizar los procesos. Sin embargo, con los recientes desarrollos tecnológicos, han surgido sistemas innovadores de riego que eliminan la necesidad de intervención constante por parte del agricultor.

El desarrollo en el sector agrícola exige alta precisión en los procesos para optimizar el rendimiento y la calidad de los cultivos, además de reducir los costos de producción. Para cumplir con estos requisitos, se deben implementar sistemas de automatización. Desde hace algunos años, se han desarrollado sistemas de control y automatización en medios controlados como los invernaderos. Estos sistemas brindan un alto grado de independencia, reduciendo la necesidad de supervisión constante, además de aprovechar las posibilidades que ofrece el Internet de las cosas (IoT).

Este proyecto busca abordar los problemas de riego, como los errores humanos y el consumo excesivo de agua, que pueden favorecer el crecimiento de hongos y bacterias perjudiciales para los cultivos. En Nicaragua, la agricultura siempre ha sido un componente esencial de la economía. La implementación de invernaderos ha permitido reducir el consumo de agua y mantener la producción sin depender de las condiciones climáticas. Sin embargo, las técnicas utilizadas en los invernaderos tradicionales siguen siendo convencionales [1].

Con este trabajo se pretende explorar el uso de tecnologías avanzadas para mejorar los métodos de producción de hortalizas en invernadero. En este documento se detalla el diseño de un sistema de automatización para el riego y la climatización de un invernadero, basado en la recolección de variables ambientales como la humedad del suelo, la temperatura y la humedad relativa del aire, el nivel de agua y la luminosidad. Los datos de estas variables serán monitoreados a través de una aplicación móvil y un servidor web.

## II. JUSTIFICACIÓN

En Nicaragua, la siembra bajo invernadero se ha implementado para revolucionar la producción agrícola. Con estos sistemas, se ha experimentado un auge en las siembras de hortalizas a gran escala, logrando productos de alta calidad, especialmente en cultivos como la chiltoma y los tomates híbridos [2].

Sin embargo, la producción en los invernaderos nicaragüenses conserva métodos tradicionales para el cuidado de los cultivos. Esto ha limitado el desarrollo de este sector económico, una situación que podría cambiar al incluir aplicaciones de tecnologías emergentes en la agricultura, fomentando la innovación en la producción a través de conceptos como la “agricultura de precisión” [3].

Esto sirve como referencia para analizar el estado actual de la implementación de sistemas de automatización en invernaderos. En Nicaragua, aún no existe ninguna empresa o microempresa que ofrezca instalaciones de sistemas de automatización personalizados para invernaderos de pequeño o mediano tamaño.

Debido a la creciente demanda de alimentos entre la población y a la necesidad de mantener bajo control la calidad y cantidad de la producción, el desarrollo de invernaderos ha cobrado auge. Sin embargo, la producción en estos espacios controlados todavía depende de la atención manual del personal, lo que genera costos por consumo de agua y remuneraciones, además de implicar tareas repetitivas y poco tecnificadas. Por otro lado, el uso del agua podría ser más eficiente, y no se llevan registros de las actividades realizadas en el invernadero para una posterior evaluación.

Esta falta de control y manejo adecuado de los cultivos provoca que no se generen condiciones óptimas para el desarrollo y crecimiento de las plantas, lo que puede resultar en producciones que no cumplen las expectativas, con tiempos prolongados y resultados no muy alentadores.

La construcción de invernaderos busca proporcionar condiciones propicias para los cultivos establecidos en su interior, promoviendo un uso eficiente del agua

y un adecuado desarrollo de las plantas para obtener hortalizas y productos de calidad. Además, el productor o inversionista desea estar informado en todo momento sobre el estado de los cultivos y asegurarse de que se cumplan las condiciones prometidas con el uso de un invernadero.

Nicaragua posee una economía sustentada en actividades del sector primario, entre las que destaca la agricultura. En este contexto, existe una gran cantidad de pequeños y medianos productores, y una pequeña proporción de grandes productores, de los cuales solo algunos implementan tecnologías en sus cultivos, mientras que la mayoría continúa utilizando métodos convencionales [3].

En respuesta a esta situación, el presente trabajo académico propone el diseño de un sistema electrónico de automatización de riego y climatización, basado en la recolección de variables ambientales e integrado con tecnología IoT y un sistema de control de acceso al invernadero. Este sistema permitirá monitorear los datos emitidos por una red de sensores a través de una aplicación móvil, ofreciendo una solución eficiente y adaptable para invernaderos. Así, se podrán controlar las condiciones de temperatura, humedad del suelo y del aire, manteniéndolas dentro de un rango óptimo que garantice un ambiente adecuado para el desarrollo de las plantas.

### **III. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Implementar un prototipo de sistema electrónico automatizado de invernadero dedicado a la producción de hortalizas.

#### **3.2. OBJETIVOS ESPECÍFICOS**

- Identificar los parámetros ambientales críticos que satisfagan las necesidades de los productores y permitan la optimización del prototipo de sistema de riego automatizado y control de clima dentro del vivero.
- Diseñar un prototipo de sistema de control climático y automatización del riego basado en datos recopilados por sensores especializados, incorporando, además, un sistema de control de acceso para la gestión eficiente del ingreso al invernadero.
- Integrar la tecnología del Internet de las Cosas (IoT) para el monitoreo y control remoto del invernadero, accesible a través de un servidor web y una aplicación móvil para una gestión más eficiente y moderna del sistema electrónico.

## **IV. MARCO TEÓRICO**

### **CAPÍTULO I**

#### **1.1. INVERNADERO**

Se puede definir un INVERNADERO, como una instalación cubierta y cerrada artificialmente con materiales transparentes, en la mayoría de los casos, con el fin de proteger las plantas de las malas condiciones climáticas (bajas temperaturas, fuertes vientos, granizo, tormentas, baja humedad del aire o excesiva radiación solar) [4].

Los cultivos se benefician durante su ciclo productivo de un microclima creado dentro del invernadero, que ayuda no solo a cultivar en periodos diferentes al origen climático del invernadero, sino que también aumenta la producción en comparación con los cultivos al aire libre, debido entre otras cosas a un mayor control sobre las condiciones climáticas [4].

#### **1.2. TIPOS DE INVERNADEROS**

Según diferentes parámetros, se podrían tener clasificaciones diferentes de los invernaderos, como las siguientes, algunos son tipos que hoy en día ya no se utilizan.

Según el rango de temperaturas a mantener en el invernadero: Invernaderos fríos, invernaderos templados, invernaderos cálidos, según materiales de cubierta, según su estructura, según su forma [5].

Actualmente se podría hacer una clasificación que mezcla varias de las clasificaciones anteriores y que son los modelos más comunes a nivel mundial [5].

**A.** Invernaderos de cristal: Invernaderos Tipo Velo.

**B.** Invernaderos de plástico: Invernaderos tipo Parral o tipo Almería (Raspa y Amagao), invernaderos tipo capilla, invernaderos tipo Gótico, invernaderos tipo Asimétrico o Tropical, invernaderos tipo Túnel.

**C.** Invernaderos de malla, invernaderos tipo Malla Sombra.



**Figura 1. Tipos de invernaderos [6]**



**Figura 2. Invernadero totalmente equipado que controla completamente las condiciones de crecimiento [6]**

### **1.3. EL TOMATE Y SU CULTIVO BAJO INVERNADERO.**

El cultivo del tomate (*Solanum Lycopersicum L.*) es una de las hortalizas de mayor importancia a nivel mundial, esto se debe a sus cualidades nutricionales altos contenido de antioxidantes y vitamina. Entre los países de América Latina de mayor producción se encuentran; Argentina, Brasil y Chile con producciones anuales de 150 t. países que contribuyen aproximadamente en el 70 % de la producción mundial. (MIFIC, 2017). Según (FAO, 2018), Centroamérica representa el 15.3% de la producción mundial de tomate [7].

Dentro de la región, se puede destacar que el mayor productor de tomate es Guatemala con 331,338.00 kg ha (58.56% de la producción regional), en segundo lugar, Nicaragua con 80,723.00 kg ha (14.27%), en tercer lugar, Honduras con 76,567.00 kg ha (13.53%), en cuarto lugar, Costa rica con 57,238.00 kg ha (10.12%) y en último lugar. El salvador con 19,915.00 kg ha (3.52%) [7].

En Nicaragua, el cultivo de tomate constituye una actividad de gran importancia económica para pequeños y medianos productores. El consumo nacional de tomate se estima en 50 millones de kilogramos y para este mismo se cosechó 1,637 hectáreas. (MIFIC, 2017). Según (FAO, 2018) en Nicaragua se establecen 1,706 ha, con rendimientos de 41,963.9 kg ha<sup>-1</sup> lo que conlleva a una producción final de 71,577 t, las zonas productoras de tomate de mayor importancia en Nicaragua se encuentran en los departamentos de Matagalpa, Jinotega y Estelí, particularmente en los Valles de Sébaco y Tomatoya [7].

El método principal de siembra utilizado es el de almácigo, que consiste en sembrar las semillas en un determinado lugar para trasplantarlas posteriormente al sitio destinado para su crecimiento, aunque últimamente se hace uso del invernadero. [7].

### **1.3.1. CARACTERÍSTICAS DE LA PRODUCCIÓN**

Para la siembra de este rubro se recomiendan los meses de julio hasta agosto, para que la cosecha salga en los meses de mayor demanda en el mercado internacional, desde octubre hasta junio. Las mejores áreas para sembrar tomate en el país están ubicadas en los lugares frescos con temperaturas promedio de 27°, Estelí, Matagalpa y Jinotega [7].

La Planta de Tomate se adapta a casi todo tipo de suelo, pero con preferencia suelos sueltos de textura silíceo arcillosa, y rica en materia orgánica. Cuando los suelos están enarenados, en cuanto al PH, los suelos pueden ser desde ligeramente ácidos hasta ligeramente alcalinos [7].

Los niveles adecuados de nutrientes son muy importantes para la producción óptima de tomate. Los principales requerimientos de nutrientes son el nitrógeno, fósforo, potasio, magnesio, calcio, azufre y boro [7].

Las especies cultivadas en invernadero toleran mejor las condiciones de salinidad, tanto en el suelo, como en el agua de riego. Se suelen sembrar con una distancia de 1.5 metros entre línea, y 0.5 metros entre planta, aunque cuando se trata de plantas de porte medio, es común aumentar la densidad de plantación a 2 plantas por metro cuadrado con marcos de 1m x 0.5 metros. [3]

En cuanto al clima, la temperatura óptima de desarrollo oscila entre 20 y 30° durante el día y entre 1° y 17° durante la noche. La maduración del fruto está muy influida por la temperatura en lo referente tanto a la precocidad como a la coloración, de forma que valores cercanos a los 10°C, así como superiores a los 30°C originan tonalidades amarillentas. El ciclo vegetativo es de 4.5 meses según la variedad [7].

### **1.3.2. PRODUCTIVIDAD**

Bajo el sistema de Invernadero, que se ha venido popularizando rápidamente entre los productores nicaragüenses, la producción se incrementa en un 50% comparado con el método tradicional, obteniendo a su vez una mejor calidad, color y sabor del producto [7].

Los productores nicaragüenses de Tomate se encuentran en un proceso en que incursionan y perfeccionan sus prácticas de producción como, por ejemplo, mediante el riego por goteo, los invernaderos y el manejo integrado de plagas, a la vez que mejoran la calidad y obtienen mejores precios en el mercado nacional. El abrirse paso en los Supermercados nacionales ha representado un gran logro y a la vez un aprendizaje que en el futuro les ayudará a incursionar en los mercados internacionales con prácticas más exigentes, especialmente cuando se trata de productos perecederos. Los tomates y la salsa de Tomate han obtenido un tratamiento favorable en los tratados de libre Comercio vigentes, con la perspectiva de apoyar el potencial de este rubro [7].

## **CAPÍTULO II**

### **2.1. VARIABLES FÍSICAS DE INTERÉS EN UN INVERNADERO**

#### **2.1.1. DEFINICIÓN**

El control del clima se refiere a mantener el valor deseado del invernadero dentro de parámetros como temperatura, luz, humedad y concentración de CO<sub>2</sub> para proporcionar condiciones óptimas para las plantas [8].

#### **2.1.2. LOS CUATRO FACTORES DEL CONTROL AMBIENTAL DE LOS INVERNADEROS**

Más que sólo la temperatura del aire exterior constituye un entorno de crecimiento. Es una combinación totalmente integrada de numerosos elementos que afectan el crecimiento de las plantas, siendo los cuatro principales el calor, la humedad, los niveles de luz y la ventilación [8].

Es posible proporcionar la combinación o "entorno" ideal para que sus plantas tengan éxito en su desarrollo, si se comprende cada uno de estos factores y se toma el control individual de cada uno de ellos [8].

##### **2.1.2.1. TEMPERATURA AMBIENTE**

Los controles de temperatura interior ayudan a bajar la temperatura durante los calurosos meses de verano, cuando demasiado calor puede provocar agotamiento por calor y otras enfermedades causadas por las altas temperaturas que podrían dañar su capacidad de producir. La temperatura interior de su invernadero se puede reducir para mantener sus plantas a la temperatura ideal para el éxito con la ayuda de cubiertas de sombra, plástico resistente al sol y buena ventilación [8].

### **2.1.2.2. HUMEDAD**

Además de ser crucial para el éxito de los cultivos y frecuentemente relacionada con la temperatura, la humedad puede tener un impacto significativo en las condiciones dentro de un área de cultivo si no se controla [8].

Aunque algunas plantas pueden prosperar en un ambiente ligeramente húmedo (pimientos, los tomates y otros vegetales ricos en agua), la humedad excesiva puede promover el crecimiento de hongos poco atractivos e incluso albergar enfermedades o parásitos, que pueden causar daños irreparables al cultivo de una planta [8].

Para garantizar que todo esté en el lugar correcto para minimizar el crecimiento de moho y maximizar el crecimiento de las plantas, se puede monitorear y controlar cuidadosamente la humedad en el cultivo dentro del invernadero utilizando la combinación adecuada de ventilación y gestión del calor [8].

### **2.1.2.3. VENTILACIÓN**

Durante muchos años, la ventilación ha sido simplemente un aspecto incontrolable del cultivo al aire libre para los productores que trabajan duro en las ventosas llanuras. Sin embargo, gracias a los avances en la tecnología de preparación para el invierno, los productores ahora pueden gestionar completamente sus esfuerzos de ventilación. Esto significa que ya no necesitarán esperar a que llegue la brisa adecuada para que sus plantas crezcan con fuerza [8].

Desde la regulación de la temperatura hasta el control de la humedad, la ventilación juega un papel clave en todo. Incluso puede ser necesario un flujo de aire regular para proporcionar CO<sub>2</sub> fresco para que las plantas de interior puedan seguir respirando fácilmente [8].

La mayoría de los fabricantes modernos creen que se puede lograr una ventilación adecuada utilizando una combinación de ventiladores y lengüetas de ventilación integradas en la pared de su almacén de invierno. Al promover el flujo de aire en días tranquilos y cerrar espacios para evitar el flujo de aire en días ventosos, se puede regular la temperatura de una manera más natural, agregar o eliminar humedad y mantener las plantas bien abastecidas con CO<sub>2</sub> respirable para que puedan crecer vigorosamente durante todo el año [8].

#### 2.1.2.4. LUZ DE SOL

Aunque la luz solar ha estado fuera del control de los productores que operan en campos abiertos bajo el sol desnudo, la capacidad de elegir la cubierta de su invernadero para regular los niveles de luz solar, e incluso filtrar la radiación UV mientras mantiene altos los niveles de luz, ha tenido un enorme impacto en productores de todo el mundo que buscan extender sus temporadas de cultivo [8].

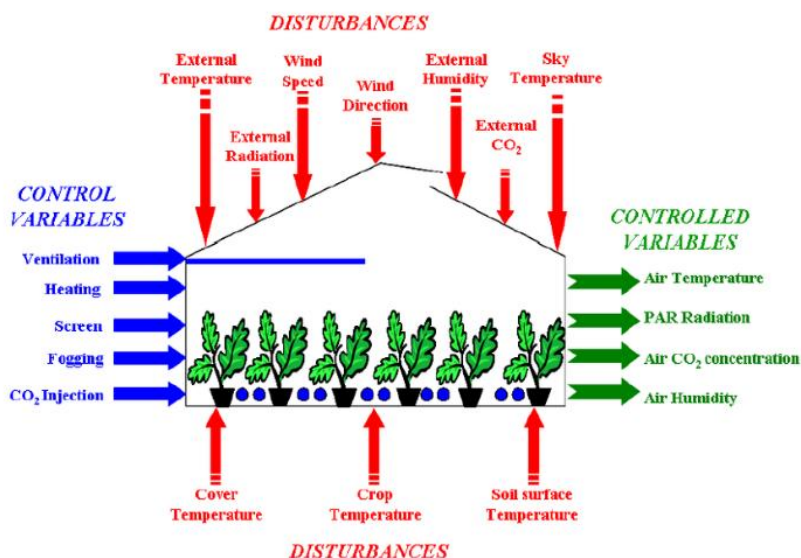


Figura 3. Parámetros ambientales que influyen en el crecimiento del tomate [8].

### 2.1.3. SISTEMA DE RIEGO

El riego y la fertilización adecuados son esenciales para producir tomates en un túnel alto. Los túneles altos impiden la entrada de lluvias naturales, por lo que es necesario un riego oportuno. Además, los productores deben considerar las necesidades especiales del suelo de túnel alto para garantizar la disponibilidad de niveles adecuados de nutrientes [9].



Figura 4. Invernadero para el cultivo del tomate [9].

### 2.1.4. RIEGO POR GOTEO

El riego por goteo es el método más eficiente para suministrar agua y nutrientes a los tomates de túnel alto. Las plantas se riegan lentamente a través de un pequeño tubo plegable de  $\frac{3}{4}$  de pulgada de diámetro sin mojar el follaje. La cinta de goteo, generalmente de 8 a 10 milímetros de espesor, se entierra a una profundidad de 1 a 2 pulgadas. El espacio entre goteros o emisores es típicamente de 4 a 12 pulgadas. Los tomates requieren una sola línea de goteo por fila, desplazada aproximadamente 2 pulgadas de la planta. Los caudales de las cintas de goteo varían. La mayoría de los productores eligen una cinta de flujo medio, que entrega medio galón por minuto (gpm) por cada 100 pies. La cinta de alto flujo, que entrega de 0,8 a 1,0 gpm, es útil para evitar obstrucciones y reducir el tiempo de riego [9].

**Tabla 1 Horas de riego por semana requeridas para aplicar 68 onzas de agua por planta de tomate por día según las poblaciones de plantas variables y las tasas de flujo del tubo de goteo [9].**

Caudal del tubo de goteo		Plantas de tomate por túnel alto		
Gph/100 pies <sup>1</sup>	gpm/100 pies <sup>2</sup>	300	400	500
8	0.13	21	28	35
10	0.17	17	22	28
12	0.20	14	19	23
dieciséis	0.27	11	14	18
18	0.30	9	12	dieciséis
20	0.33	8	11	14
24	0.40	7	9	12
30	0.50	6	8	9
36	0.60	5	6	8
40	0.67	4	6	7
42	0.70	4	5	7
48	0.80	3.5	5	6
60	1.00	2.8	4	5

<sup>1</sup> galón de agua por hora por tramo de 100 pies de cinta de goteo.  
<sup>2</sup> galones de agua por minuto por tramo de 100 pies de cinta de goteo.

### 2.1.5. REQUISITOS DE AGUA EN EL TOMATE DE INDUSTRIA

Las necesidades hídricas del tomate son muy altas, ya que estas varían entre los 5.000 y 7.000 m<sup>3</sup>/ha. Estas necesidades deben evaluarse en base a un análisis climático (precipitación y evapotranspiración de referencia) de la estación agroclimática más próxima a la ubicación de la finca, y en base al tipo de suelo y a los rendimientos esperados en la explotación. La filosofía es aportar justo el volumen hídrico que la planta necesita, con el fin de reducir las pérdidas de agua por percolación, evaporación y escorrentía, y esto sólo se consigue mediante el uso de un sistema de riego que permita la máxima eficiencia de aplicación a baja presión, es decir, mediante el riego por goteo. [10].

## **CAPÍTULO III**

### **3.1. SISTEMA DE CONTROL**

#### **3.1.1. ¿CÓMO SELECCIONAR LOS COMPONENTES ELECTRÓNICOS?**

La selección depende de varios factores, como si se necesita un diseño compacto, un diseño fácil de fabricar, el más asequible, el más eficiente en energía, la menor cantidad de componentes electrónicos, el más confiable, etc. Entonces, al final es una compensación, en la que está tratando de decidir qué componente electrónico se ajusta a la mayoría de sus requisitos [11].

Los requisitos de diseño fueron el paso clave en la selección de las piezas. El verdadero problema que se está tratando de resolver no es qué piezas elegir, sino qué parámetros y funciones se necesitarían para cumplir con los requisitos del proyecto; las piezas sólo ayudan a cumplir esos requisitos [11].

#### **3.1.2. ¿QUÉ TARJETA DE DESARROLLO ELEGIR?**

Para proyectos electrónicos de menor complejidad, las tarjetas Arduino representan una opción ideal por su facilidad de uso y bajo costo; mientras que, para tareas que requieren mayor potencia de procesamiento y compatibilidad con sistemas operativos como Linux, las computadoras de placa única (SBC), como la Raspberry Pi, resultan más adecuadas. Por su parte, las placas ESP, como ESP8266 y ESP32, se han vuelto muy populares para proyectos de Internet de las Cosas (IoT) gracias a sus módulos integrados de Wi-Fi y Bluetooth, que permiten conectar dispositivos y redes de manera sencilla y económica [12].

### 3.1.3. PLACAS DE DESARROLLO PARA PROYECTOS BASADOS EN INTERNET DE LAS COSAS

#### 3.1.3.1. ESP32 y ESP8266

ESP32 y ESP8266 son módulos Wi-Fi muy rentables, ideales para proyectos en el campo de Internet de las cosas (IoT) [13].

Ambos chips son procesadores de 32 bits. El ESP32 es una CPU de doble núcleo de 160 MHz a 240 MHz, mientras que el ESP8266 es un procesador de un solo núcleo que funciona a 80 MHz [13].

Todos estos módulos WiFi vienen con GPIO y admiten varios protocolos como SPI, I2C, UART, etc. La mejor parte es que vienen con redes inalámbricas, lo que la diferencia de otros microcontroladores como Arduino. Esto significa que puede controlar y monitorear fácilmente el dispositivo de forma remota a través de Wi-Fi o Bluetooth (ESP32) por un precio muy bajo [13].



Figura 5. ESP32 frente a ESP8266 [13].

#### 3.1.3.2. ESPECIFICACIONES: ESP32 VS ESP8266

El ESP32 es el sucesor del ESP8266. Agrega núcleos de CPU adicionales, Wi-Fi más rápido, más GPIO y es compatible con Bluetooth 4.2 y Bluetooth Low Energy. Además, el ESP32 viene con pines de detección táctil que se pueden usar para despertar al ESP32 del modo de suspensión profunda, un sensor de efecto Hall incorporado y un sensor de temperatura incorporado [13].

### 3.1.3.3. MÁS GPIO EN ESP32

El ESP32 tiene más GPIO que el ESP8266, puede decidir qué pines UART, I2C, SPI, solo necesita configurar el código. Esto es posible gracias a las capacidades de multiplexación del chip ESP32 que le permiten asignar múltiples funciones al mismo pin [13].

Si no se configuran en el código, estarán en los pines definidos de manera predeterminada, como se muestra en la imagen a continuación (este es un ejemplo para la placa ESP32 DEVKIT V1 DOIT; las ubicaciones de los pines pueden variar según el fabricante) [13].

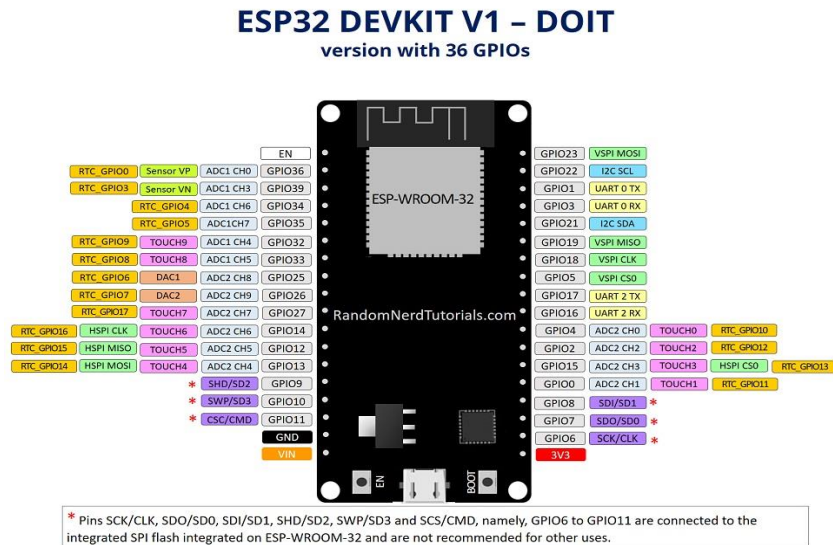


Figura 6. ESP32 DEVKIT V1 DOIT [13].

### 3.1.3.4. PWM, ADC, ETC

Puede configurar la señal PWM en cualquier GPIO y configurar la frecuencia configurable y el ciclo de trabajo en el código [13].

Para los pines analógicos, estos son estáticos, pero el ESP32 admite mediciones en 18 canales (pines de habilitación analógica), mientras que el

ESP8266 solo tiene un pin ADC de 10 bits. El ESP32 también admite dos canales DAC de 8 bits [13].

Además, el ESP32 contiene 10 GPIO de detección capacitiva que detectan el tacto y se pueden usar para activar eventos o despertar el ESP32 del modo de suspensión profunda [13].

### 3.1.3.5. ARDUINO IDE - ESP32 FRENTE A ESP8266

Hay muchas formas de programar las placas ESP32 y ESP8266. Actualmente, ambas placas se pueden programar utilizando el entorno de programación Arduino IDE [13].

En este caso, cada una de las opciones discutidas anteriormente fue investigada a fondo, lo cual derivó en la elección del ESP32 porque ofrece beneficios superiores en comparación con las otras placas de desarrollo. El principal beneficio de esta placa es que permite la conexión a internet, lo cual es fundamental para el funcionamiento del sistema, porque es necesario monitorear todas las variables agroclimáticas utilizando el servidor de Adafruit

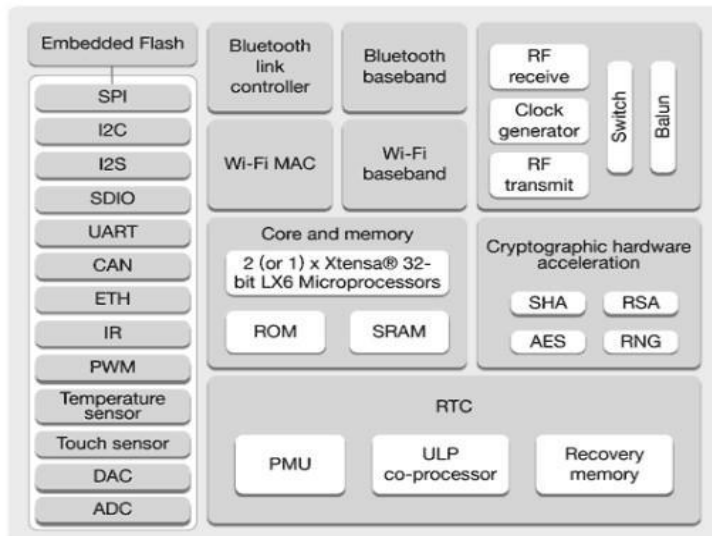


Figura 7. Diagrama de bloque de funcionamiento del ESP32 [14].

En la siguiente tabla se muestra un resumen de sus características técnicas. Se aprovecha también para compararlo con el ESP8266 para ver las diferencias [14].

**Tabla 2 Alimentación y modos de consumo, ESP8266 versus ESP32 [14].**

Bluetooth	No	v4.2 BR/EDR y BLE
UART	2* (En una de ellas solo puede utilizarse el pin Tx)	3
I2C	1	2
SPI	2	4
GPIO (utilizables)	32	11
PWM	8	16
ADC	1 (10 bit)	18 (12 bit)
ADC con preamplificador	No	Sí (Bajo ruido) Hasta 60 dB
DAC	No	2 (8 bit)
1-Wire	Implementado por software	Implementado por software
I2S	1	2
CAN bus	No	1 x 2.0
Ethernet	No	10/100 Mbps MAC
Sensor de temperatura	No	Sí
Sensor efecto HALL	No	Sí
IR	Sí	Sí
Temporizadores	3	4 (64 bits)
Encriptación por hardware	No (TLS 1.2 por software)	Sí (AES, SHA, RSA, ECC)
Gen. de núm. aleatorios	No	Sí
Encriptación de la flash	No	Sí
Arranque seguro	No	Sí

### 3.1.4. SENSORES

#### 3.1.4.1. SONDAS DE SENSORES

Se denomina sonda sensora a cualquier instrumento o dispositivo que mide ciertas propiedades físicas o químicas del medio ambiente y transmite los resultados como una señal eléctrica para ser recibida e interpretada por la computadora principal automatizada con fines de control y toma de decisiones. Algunos de los parámetros clave para medir en un ambiente invernal incluyen el microclima (es decir, temperatura del aire, humedad relativa), humedad del suelo, temperatura del suelo y nivel de luz (o radiación solar). Las mediciones de estos sensores pueden revelar la presencia de moho o condensación [15].

#### 3.1.4.2. SENSOR TEMPERATURA Y HUMEDAD

El módulo DHT11 y DHT22 presenta un complejo de sensores de temperatura y humedad con una salida de señal digital calibrada. La exclusiva

técnica de adquisición de señales digitales y la tecnología de detección de temperatura y humedad garantizan una alta confiabilidad y una excelente estabilidad a largo plazo. Este sensor incluye un NTC para medir la temperatura y un componente de medición de humedad de tipo resistivo para medir la humedad. Estos están conectados a un microcontrolador de 8 bits de alto rendimiento, que ofrece excelente calidad, respuesta rápida, capacidad anti-interferencias y rentabilidad [16].

El DHT11 es un sensor de temperatura y humedad digital básico y de muy bajo costo. Utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante y envía datos a través de un protocolo de 1 cable. DHT11 puede medir la temperatura de 0 a 50 °C con una precisión del 2 % y la humedad relativa del 20 al 80 % con una precisión del 5 % [16].

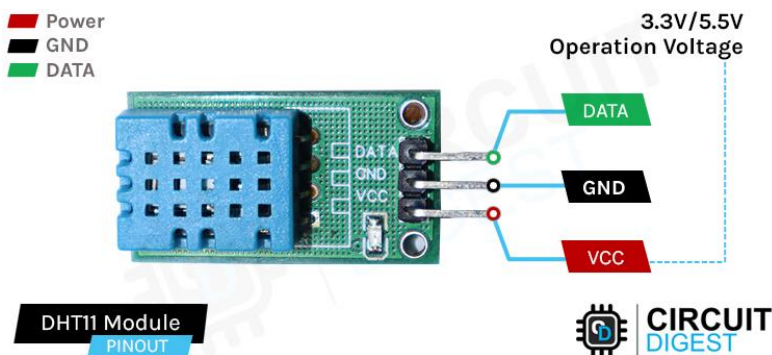


Figura 8. Sensor DHT11 Module [16].

DATOS Pin de datos para comunicación de 1 hilo.

TIERRA Conexión a tierra al pin de tierra del ESP32.

CCV Proporciona energía para el módulo, conéctese al pin de 5V del ESP32.

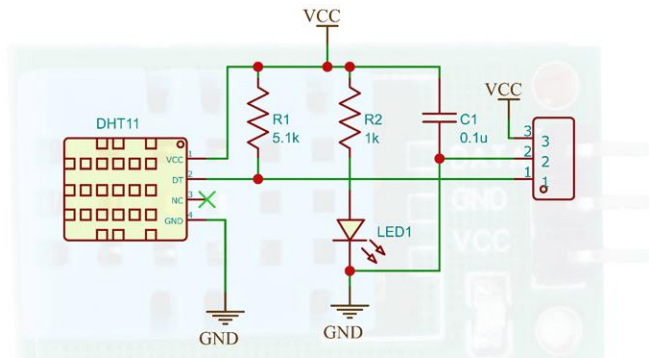


Figura 9. Diagrama del circuito del módulo DHT11 [16].

### 3.1.4.3. ¿CÓMO FUNCIONA DHT11?

Como ya se mencionó, el DHT11 tiene un termistor NTC y componentes de detección de humedad. Cuando la temperatura cambia, la resistencia del NTC también cambia. Este cambio en la resistencia se mide y la temperatura se calcula a partir de él [16].

El componente de detección de humedad consta de un sustrato que retiene la humedad intercalada entre dos electrodos. Cuando el sustrato absorbe contenido de agua, la resistencia entre los dos electrodos disminuye. El cambio de resistencia entre los dos electrodos es proporcional a la humedad relativa. Una humedad relativa más alta disminuye la resistencia entre los electrodos, mientras que una humedad relativa más baja aumenta la resistencia entre los electrodos. Este cambio en la resistencia se mide con el ADC de la MCU integrada y se calcula la humedad relativa [16].

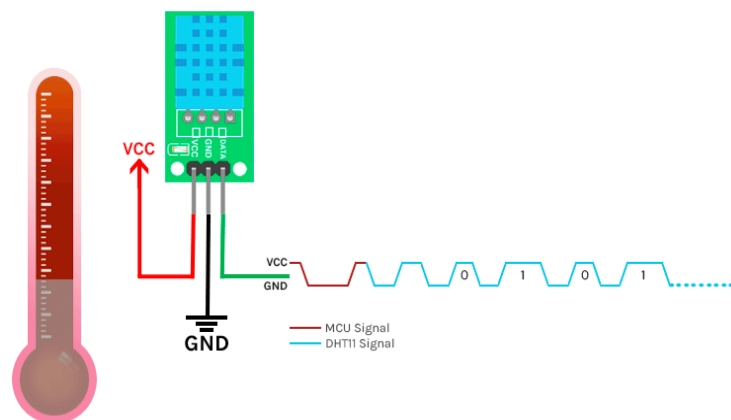


Figura 10. Funcionamiento del sensor DHT [16].

Cada elemento DHT11 está estrictamente calibrado en un laboratorio, lo que es extremadamente preciso en la calibración de la humedad. Los coeficientes de calibración se almacenan como programas en la memoria OTP, que son utilizados por el proceso de detección de señales internas del sensor [16].

La principal diferencia entre ambos es que el ciclo de operación es menor en el DHT11 que en el DHT22, y que el DHT22 tiene rangos de medida más amplios y mayor resolución. La diferencia en precio es pequeña, algo más caro el DHT22, pero merece la pena decantarse por este. Ambos sensores están calibrados en laboratorio y tienen una buena fiabilidad. Esto supone una gran ventaja, ya que simplifican las conexiones a realizar en la placa [16].

A continuación, se muestra una tabla con las características de estos sensores y la comparativa entre ellos, obtenida de la referencia.

**Tabla 3 Características y comparativa sensores DHT [16].**

Parámetro	DHT11	DHT22
Alimentación	3Vdc ≤ Vcc ≤ 5Vdc	3.3Vdc ≤ Vcc ≤ 6Vdc
Señal de Salida	Digital	Digital
Rango de medida Temperatura	De 0 a 50 °C	De -40°C a 80 °C
Precisión Temperatura	±2 °C	<±0.5 °C
Resolución Temperatura	0.1°C	0.1°C
Rango de medida Humedad	De 20% a 90% RH	De 0 a 100% RH
Precisión Humedad	4% RH	2% RH
Resolución Humedad	1%RH	0.1%RH
Tiempo de sensado	1s	2s
Tamaño	12 x 15.5 x 5.5mm	14 x 18 x 5.5mm

En lo que se refiere al pinout, los pines del DHT22 siguen el mismo orden:

De izquierda a derecha se tiene:

- VCC: Alimentación del sensor
- Señal: Información de temperatura y humedad

- NC: No Conexión
- GND: Masa del circuito

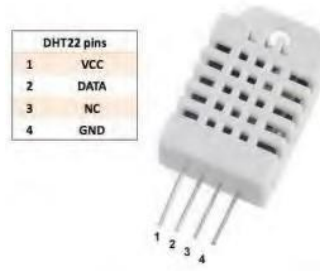


Figura 11. Ilustración 16 - Pines del sensor DHT22 [16].

### 3.1.5. SENSOR DE HUMEDAD HIGRÓMETRO

El Sensor de humedad de Suelo FC-28 permite medir de forma sencilla la humedad del suelo por medio de 2 electrodos resistivos. Compatible con Arduino, PIC, ESP8266/NodeMCU/NodeMCU-32, ESP32. El sensor es ideal para monitorear el nivel de humedad de las plantas y así recordar cuando necesitan ser regadas o incluso para realizar un sistema totalmente automatizado de riego añadiendo una válvula o una bomba de agua [17].

Básicamente este sensor encarga de medir el nivel de humedad presente en el suelo donde se encuentre instalado. A través del pin D0, transmite la señal correspondiente al valor de humedad detectado. Esta señal varía en función de los cambios en la resistencia eléctrica del terreno, la cual depende directamente de la cantidad de humedad existente en el mismo [17].

El sensor está compuesto por dos piezas: la placa electrónica (a la derecha) y la sonda con dos almohadillas, que detecta el contenido de agua (a la izquierda).



Figura 12. Sensor de humedad del suelo FC-28 [17].

El sensor tiene incorporado un potenciómetro para el ajuste de la sensibilidad de la salida digital (D0), un LED de alimentación y un LED de salida digital [17].

La sonda dispone de 2 terminales separados adecuadamente y un módulo que contiene un circuito comparador LM393 SMD (de soldado superficial) muy estable, un led de encendido y otro de activación de salida digital. Este último presenta 2 pines de conexión hacia el módulo FC-28, 2 pines para la alimentación y 2 pines de datos. VCC, GND, D0, A0 [17].

Este sensor tiene la capacidad de medir la humedad del suelo. Aplicando una pequeña tensión entre los terminales del módulo FC-28 hace pasar una corriente que depende básicamente de la resistencia que se genera en el suelo y ésta depende mucho de la humedad. Por lo tanto, al aumentar la humedad la corriente crece y al bajar la corriente disminuye [17].

#### 3.1.5.1. CARACTERÍSTICAS:

- Modelo: Sensor de Humedad del Suelo (Tierra) Resistivo
- Voltaje de Alimentación DC (VCC): 3.3V ~ 5V
- Voltaje de salida: 0V ~ 4.2V
- Corriente: 35 mA

- A0: Salida analógica que entrega una tensión proporcional a la humedad. Puede ser medida directamente desde un puerto analógico en un Microcontrolador, Arduino, entre otros.
- D0: Salida digital; este módulo permite ajustar cuándo el nivel lógico en esta salida pasa de bajo a alto mediante el potenciómetro

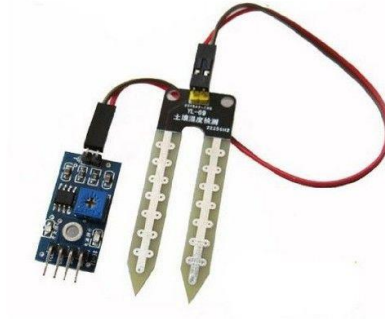


Figura 13. Sensor de humedad del suelo Higrómetro [17].

Se debe tener en cuenta que el valor del sensor analógico depende de la tensión de funcionamiento. La siguiente tabla muestra el valor del sensor analógico para todas las diferentes combinaciones de voltaje de funcionamiento, humedad y tipo de sensor de humedad del suelo [17].

Tabla 4 Valores del sensor analógico para las diferentes combinaciones [17].

Tensión de funcionamiento	3,3V	5V		
Seco/Húmedo	Seco	Húmedo	Seco	Húmedo
Sensor capacitivo de humedad del suelo	2...8	14...21	821...824	490...549
Sensor de humedad del suelo resistivo	1024	483...505	1023	344...358

En teoría, se puede usar cualquiera de estas combinaciones, pero se debe calibrar el sensor antes de declarar el suelo húmedo o seco [17].

### 3.1.6. RESISTENCIA DEPENDIENTE DE LA LUZ LDR: FOTORRESISTENCIA

Las resistencias dependientes de la luz, LDR o fotorresistores son componentes electrónicos que se utilizan para detectar la luz y cambiar el funcionamiento de un circuito en función de los niveles de luz [18].

Los LDR son muy diferentes a otras formas de resistencia, como la resistencia de película de carbono, la resistencia de película de óxido metálico, la resistencia de película metálica y similares, que se utilizan ampliamente en otros diseños electrónicos. Están diseñados específicamente para su sensibilidad a la luz y el cambio en la resistencia que esto provoca [18].



Figura 14. Resistencia típica dependiente de luz con plomo [18].

#### 3.1.6.1. FUNCIONAMIENTO

Un LDR o fotorresistor está hecho de cualquier material semiconductor con una alta resistencia. Tiene una alta resistencia porque hay muy pocos electrones libres y capaces de moverse; la gran mayoría de los electrones están encerrados en la red cristalina y no pueden moverse. Por lo tanto, en este estado hay una alta resistencia LDR [18].

A medida que la luz cae sobre el semiconductor, los fotones de luz son absorbidos por la red del semiconductor y parte de su energía se transfiere a los electrones. La cantidad de energía transferida a los electrones les da a algunos de ellos la energía suficiente para liberarse de la red cristalina y luego conducir la electricidad. Esto da como resultado una disminución de la resistencia del semiconductor y, por lo tanto, de la resistencia LDR general [18].

El proceso es progresivo, y a medida que más luz incide sobre el semiconductor LDR, se liberan más electrones para conducir la electricidad y la resistencia cae aún más. Estructuralmente, el fotorresistor es un resistor sensible a la luz que tiene un cuerpo horizontal que está expuesto a la luz [18].

El formato básico para un fotorresistor es el que se muestra a continuación:

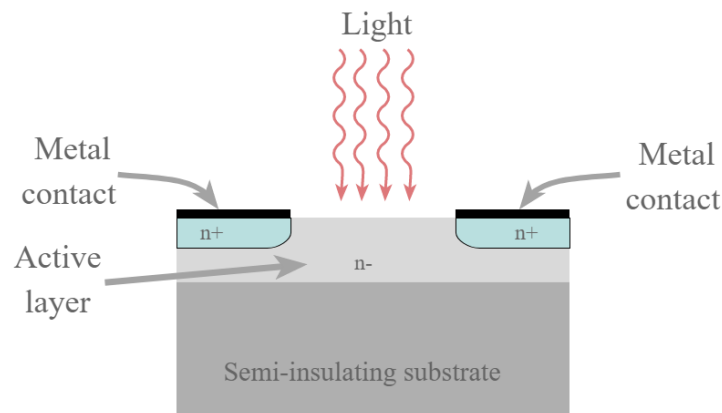


Figura 15. Estructura del fotorresistor [18].

### 3.1.6.2. CIRCUITOS LDR

Hay muchos circuitos que se utilizan para resistencias dependientes de la luz. Estos circuitos LDR pueden basarse en transistores bipolares, amplificadores operacionales FET, etc [18].

Sin embargo, la base de la mayoría de los circuitos LDR es un divisor de potencial, y luego se puede usar con varios otros circuitos para procesar el voltaje según sea necesario. Un divisor de potencial básico consta de dos resistencias en serie donde un extremo generalmente está conectado a un potencial fijo y el otro a tierra [18].

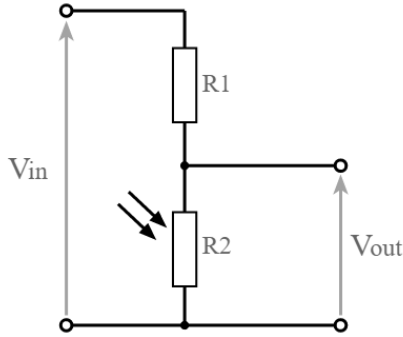


Figura 16. Circuito LDR básico usando un divisor de potencial [18].

Es bastante sencillo calcular el voltaje de salida usando la fórmula a continuación.

$$V_{out} = V_{in} \left( \frac{R_2}{R_1 + R_2} \right)$$

Figura 17. voltaje de salida usando la fórmula [18].

Esto supone que el circuito divisor de potencial no tiene una carga en la salida que afecte materialmente el voltaje. Por lo general, una carga de alta impedancia significará que el circuito funcionará como se espera; de lo contrario, la carga y R2 deben calcularse en paralelo para formar la resistencia general del extremo inferior del divisor de potencial. Puede verse que, si la resistencia dependiente de la luz es, por ejemplo, R2, entonces, a medida que varía, variará la tensión de salida del divisor de potencial [18].

### 3.1.7. SENSOR ULTRASÓNICO

El sensor 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. Posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación.

El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de

sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar) [19].

La distancia se puede calcular utilizando la siguiente fórmula:

$$\text{Distancia(m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\}/2$$

El sensor **US-016** es similar al HC-SR04 pero con salida de tipo analógico, otro sensor ultrasonido es el sensor **US-100** con salida de tipo uart/serial [19].

#### 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
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)

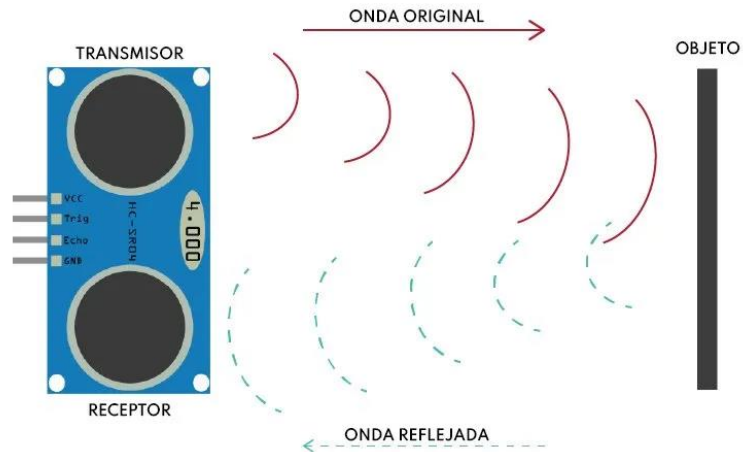


Figura 18. Sensor ultrasónico de distancia [19].

## COMUNICACIÓN CON EL MICROCONTROLADOR

- De acuerdo con [19], el procedimiento de medición es realizado por un microcontrolador que interactúa con el HC-SR04 mediante dos pines digitales y funciona de la siguiente forma:
- **El pin TRIGGER** recibe un pulso de habilitación de parte del microcontrolador, mediante el cual se le indica al módulo que comience a realizar la medición de distancia. Este pulso debe durar 10 microsegundos.
- **En el pin ECHO** el sensor devuelve al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de vuelta al módulo.
- Mediante una sencilla formula puede estimarse entonces la distancia entre el sensor y el obstáculo si se conoce el tiempo de viaje del sonido, así como la velocidad de propagación del sonido:
- Para obtener la distancia en centímetros hay que dividir el tiempo en microsegundos entre 58
- Para obtener la distancia en pulgadas será entre 148.



Figura 19. Pines del sensor de ultrasonidos [19].

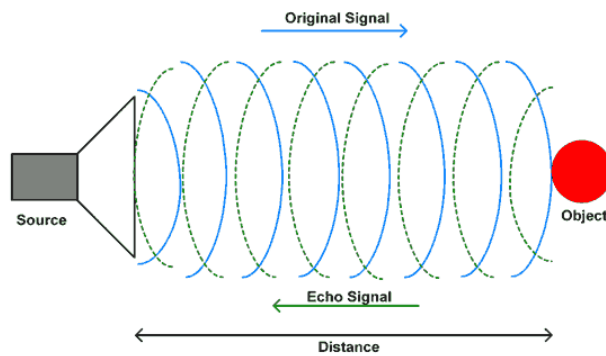


Figura 20. Principio de funcionamiento del sensor ultrasónico [19].

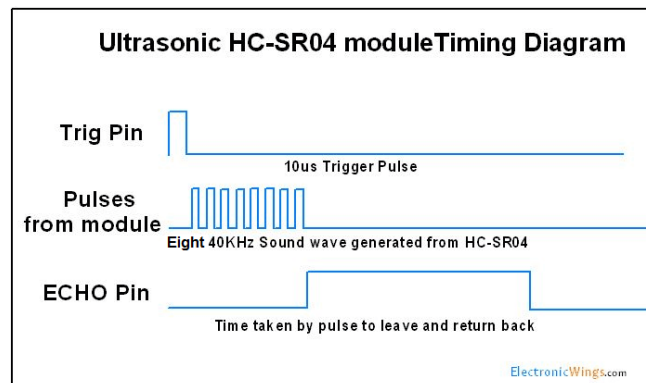


Figura 21. Diagrama de temporización del módulo [19].

### 3.1.7.1. CAPTURA DEL NIVEL DEL AGUA

Según [20], para medir el nivel de agua del tanque se puede utilizar el sensor ultrasónico HC-SR04. Este sensor dispara un ping de ultrasonido en la dirección de un objeto, y al medir el tiempo que tarda en su eco para regresar, se puede encontrar la distancia entre el sensor y el objeto, como se muestra en la Figura.

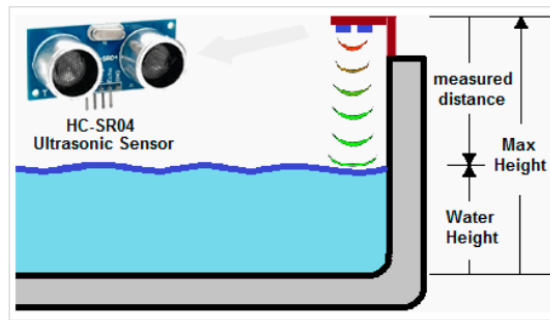


Figura 22. Medición del nivel de agua del tanque [20].

Cuando el sensor se coloca de cara al agua se puede encontrar la altura del agua. La frecuencia del ping ultrasónico se puede configurar mediante programación, Para la parte del sensor, el tipo de sensor no es lo más destacado de este trabajo [20].

La idea detrás de todos los métodos sin contacto es medir la distancia entre el transceptor y el fluido. Como se dijo antes, se transmite un pulso ultrasónico corto y se mide el tiempo de viaje de ese pulso desde el transceptor al líquido y de regreso al transceptor. El pulso ultrasónico rebotará desde el nivel del líquido debido al cambio de densidad del medio de viaje del pulso ultrasónico (el pulso ultrasónico primero viaja a través del aire y rebota del líquido con mayor densidad que el aire). Debido a que el agua tiene mayor densidad, la mayoría del pulso rebotará [20].

### 3.1.8. ACTUADORES

Un actuador es un dispositivo que produce un movimiento al convertir la energía y las señales que ingresan al sistema. El movimiento que produce puede ser rotatorio o lineal. Los actuadores lineales eléctricos, como su nombre lo indica, producen movimiento lineal [21].

Esto significa que los actuadores lineales pueden moverse hacia adelante o hacia atrás en un plano lineal establecido, una distancia establecida que pueden recorrer en cualquier dirección antes de que deban detenerse. Los actuadores

giratorios, por otro lado, producen un movimiento giratorio, lo que significa que el actuador gira en un plano circular. A diferencia del actuador lineal, el actuador giratorio no está limitado por una trayectoria establecida, lo que significa que puede seguir girando en la misma dirección durante el tiempo que sea necesario. [21].

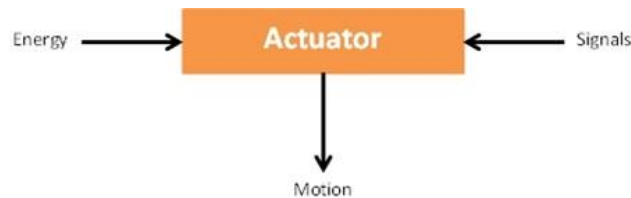


Figura 23. Funcionamiento básico de un actuador [21].

### 3.1.8.1. BOMBA DE AGUA PARA RIEGO

Para controlar una bomba de agua con un ESP32, se conecta la bomba a los pines de salida digital del ESP32. La bomba normalmente tiene dos cables, uno para alimentación y otro para tierra. Estos cables se pueden conectar a los pines 5V y GND del ESP32, respectivamente. Una vez que se realizan las conexiones, puede usar los pines de salida digital del ESP32 para controlar la bomba [22].

Una bomba de agua que funciona a 12 V es un tipo de bomba diseñada para funcionar con una fuente de alimentación de CC de 12 voltios. Estos tipos de bombas se usan comúnmente en aplicaciones automotrices, marinas y de vehículos recreativos, así como también en riego a pequeña escala, tratamiento de agua y otros sistemas similares [22].

El uso de una bomba de agua de 12 V con el ESP32 requería una fuente de alimentación externa, ya que el ESP32 no puede proporcionar la energía necesaria para operar la bomba. En este caso, se optó por usar un relé para encender y apagar la bomba usando los pines digitales del ESP32, y también se puede usar un regulador de voltaje para reducir el voltaje de la fuente de alimentación externa a 12V [22].



Figura 24. Bomba de agua 12V DC [22].

### Especificaciones:

- Material de la bomba: ABS
- Condición de uso: Continuo
- Diámetro de salida de la bomba: 8 mm
- Temperatura máxima de funcionamiento: 60 grados
- Consumo de energía: 4.2W
- Tensión nominal: 12V CC
- Corriente máxima de carga: 350mA
- Caudal máximo: 4L/MIN (1.06G/MIN)
- Clase a prueba de agua: IP68 (se puede instalar sumergible)
- Vida útil: más de 30000 horas
- Fuente de alimentación: módulos de alimentación de CC.
- Fluidos: Agua, solución ácida y alcalina

Básicamente lo que se hizo fue realizar la activación de la bomba de 12 v mediante un relé, el cual era alimentado por el ESP32, enviando de esta forma el pulso para la activación de esta, el propósito con el diseño primario de este sistema es que pueda extrapolarse a gran escala, pudiendo ser utilizado en invernaderos que disponga de mayores dimensiones.

### 3.1.9. OTROS COMPONENTES

#### 3.1.9.1. HARDWARE DEL MÓDULO ADAPTADOR LCD A I2C



Figura 25. Módulo adaptador LCD a I2C [23].

El adaptador LCD a I2C tiene los pines ordenados para conectar directamente al LCD, esto se puede hacer a través de un protoboard o soldando directamente al LCD [23].

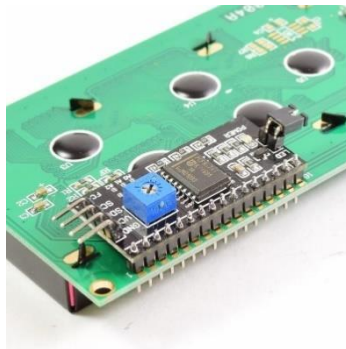


Figura 26. Módulo adaptador LCD a I2C [56]

#### **Características:**

- Solo 2 pines ESP32 están ocupados (usar interfaz I2C).
- Admite el modo I2C estándar (100 Kbit/s) y el modo I2C rápido (400 Kbit/s).
- Compatible con múltiples niveles lógicos de comunicación: 2.8~5VDC.
- Compatible con la biblioteca Arduino, usa una línea de código para completar la pantalla.
- Integra 7 tamaños de fuentes ASCII, 5 funciones gráficas.
- Proporciona un software de conversión de datos de imagen dedicado (Convertidor de mapa de bits).

- La mayor parte de la operación compleja es procesada por el controlador independiente I2C\_LCD, ahorrando recursos del controlador de usuario.
- Admite la función de cursor, puede configurar 16 frecuencias de parpadeo del cursor.
- Admite ajuste de luminosidad de retroiluminación de 128 niveles.
- Admite ajuste de contraste de pantalla de 64 niveles.
- Admite la modificación de la dirección del dispositivo.
- Admite el trabajo 127 I2C\_LCD en paralelo.
- Al depurar el código, puede tomar el lugar del monitor en serie para monitorear el estado de ejecución del programa.
- Se proporcionan dos métodos de recuperación anormales: restablecer y restaurar la configuración de fábrica.
- Compatible con la interfaz Grove y la interfaz 4Pin-100mil (debajo del zócalo Grove).
- Diseño de 4 orificios fijos simétricos para una fácil instalación por parte del usuario [23].



Figura 27. Módulo adaptador LCD a I2C [23].

### 3.1.9.2. MÓDULO RELÉS

Un relé es un interruptor electrónico que puede ser controlado por un microcontrolador Arduino para encender y apagar un actuador. Cuando el relé se energiza, cierra un circuito eléctrico, permitiendo que la electricidad fluya a través

de la bomba y la enciende. Cuando el relé se desactiva, abre el circuito, cortando la electricidad y apagando la bomba [24].

Para usar un relé para controlar una bomba de agua con un ESP32, se deberá conectar el relé a los pines de salida digital del Arduino. El relé normalmente tiene tres pines, VCC, GND e IN. VCC debe conectarse a 5V del ESP32, GND a GND e IN a un pin digital del ESP32 [24].



Figura 28. Módulo de dos relés [25].

Tabla 5 Descripción de pines del módulo de relé de un solo canal [25].

Número de PIN	Nombre del pin	Descripción
1	Disparador de relé	Entrada para activar el relé
2	Suelo	referencia 0V
3	CCV	Entrada de alimentación para alimentar la bobina del relé
4	Normalmente abierto	Terminal normalmente abierto del relé
5	Común	Terminal común del relé
6	Normalmente cerrado	Contacto normalmente cerrado del relé

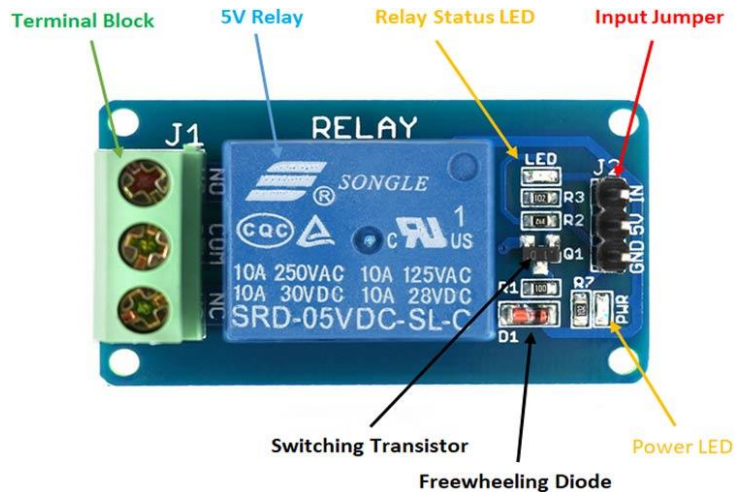


Figura 29. Pines de un relé [25].

### 3.1.9.3. MÓDULO RFID RC522



Figura 30. Módulo RFID RC522 [61]

El RC522 es un módulo RFID de 13,56 MHz que se basa en el controlador MFRC522 de NXP semiconductors. El módulo puede admitir I2C, SPI y UART y normalmente se envía con una tarjeta RFID y un llavero. Se usa comúnmente en sistemas de asistencia y otras aplicaciones de identificación de personas/objetos [26].

**Tabla 6 Configuración de pines RC522 [26].**

Número de PIN	Nombre del pin	Descripción
1	vcc	Se utiliza para alimentar el módulo, por lo general se utilizan 3,3 V
2	PRIMERA	Pin de reinicio: se utiliza para reiniciar o apagar el módulo
3	Suelo	Conectado a tierra del sistema
4	IRQ	Pin de interrupción: se utiliza para activar el módulo cuando un dispositivo entra en el rango
5	MISO/SCL/TX	El pin MISO cuando se usa para comunicación SPI, actúa como SCL para I2c y Tx para UART.
6	MOSI	Master out esclavo en pin para comunicación SPI
7	SCK	Pin de reloj en serie: se utiliza para proporcionar una fuente de reloj
8	SS/SDA/Rx	Actúa como entrada serial (SS) para comunicación SPI, SDA para IIC y Rx durante UART

### **3.1.9.3.1. CARACTERÍSTICAS RC522**

- Módulo RFID de 13,56 MHz
- Voltaje de funcionamiento: 2,5 V a 3,3 V
- Comunicación: SPI, protocolo I2C, UART
- Velocidad máxima de datos: 10Mbps
- Rango de lectura: 5 cm
- Consumo de corriente: 13-26mA
- Consumo en modo apagado: 10uA (min) [26].

El módulo funciona en 13,56 MHz, que es una banda industrial (ISM) y, por lo tanto, se puede utilizar sin ningún problema de licencia. Funciona típicamente a 3,3 V y, por lo tanto, se usa comúnmente en diseños de 3,3 V. Normalmente se usa en aplicaciones donde cierta persona/objeto debe identificarse con una identificación única [26].

El llavero tiene 1kB de memoria que se puede utilizar para almacenar datos únicos. El módulo lector RC522 puede leer y escribir datos en estos elementos de memoria. El lector solo puede leer datos de etiquetas pasivas que funcionan en 13,56 MHz [26].

Dado que, en la aplicación, la mayor parte del tiempo el módulo lector estará esperando que la etiqueta se acerque. El Lector se puede poner en modo de apagado para ahorrar energía en aplicaciones que funcionan con batería. Esto se puede lograr usando el pin IRQ en el módulo. La corriente mínima consumida por el módulo durante el modo de apagado será solo de 10 uA [26].

### 3.1.9.3.2. APLICACIONES

- Sistemas de facturación automática
- Sistemas de asistencia
- Sistema de verificación/identificación
- Sistemas de control de acceso [26].

### 3.1.9.3.3. MODELO 2D DEL MÓDULO RFID RC522

El lector consta de un módulo de radiofrecuencia y una antena que genera un campo electromagnético de alta frecuencia. Mientras que la etiqueta suele ser un dispositivo pasivo (no tiene batería). Consiste en un microchip que almacena y procesa información, y una antena para recibir y transmitir una señal [26].

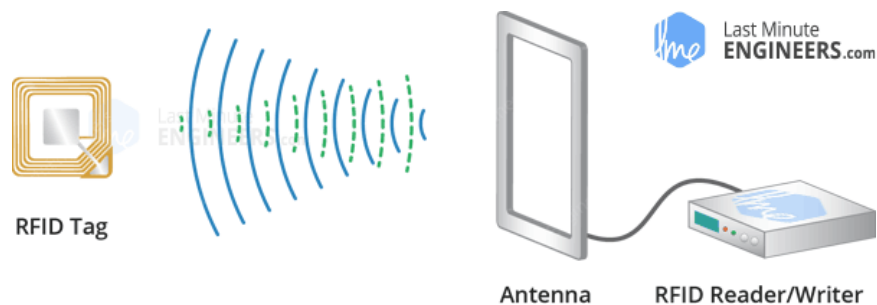


Figura 31. Lector RFID [26].

Cuando la etiqueta se acerca al lector, el lector genera un campo electromagnético. Esto hace que los electrones se muevan a través de la antena de la etiqueta y, posteriormente, alimenta el chip [26].

Luego, el chip responde enviando su información almacenada al lector en forma de otra señal de radio. Esto se llama retrodispersión. El lector detecta e interpreta esta retrodispersión y envía los datos a una computadora o microcontrolador [26].



**Figura 32. Acceso mediante aproximación al lector RFID [26].**

# CAPÍTULO IV

## 4. CONECTIVIDAD WiFi

### 4.1. COMUNICACIONES POR INTERNET

Existen diversos protocolos involucrados en el establecimiento de conexiones a internet, y su elección depende del tipo de comunicación que se necesite. Sin embargo, la transmisión de datos no se limita únicamente al entorno de internet: también se da en otros tipos de redes y sistemas. En todos los casos, los protocolos de red son los encargados de definir las reglas y características de la conexión, asegurando una comunicación efectiva entre los dispositivos. [27].

Un protocolo es un conjunto de reglas: los **protocolos de red** son estándares y políticas formales, conformados por restricciones, procedimientos y formatos que definen el intercambio de **paquetes** de información para lograr la comunicación entre dos **servidores** o más dispositivos a través de una red [27].

Los **protocolos de red** incluyen mecanismos para que los dispositivos se identifiquen y establezcan conexiones entre sí, así como reglas de formato que especifican cómo se forman los **paquetes** y los datos en los mensajes enviados y recibidos. Algunos protocolos admiten el reconocimiento de mensajes y la compresión de datos diseñados para una comunicación de red confiable de alto rendimiento [27].

### 4.2. CONEXIÓN MÓDULO WiFi

Existen una serie de pasos que se deben seguir para la instalación de cualquier módulo ESP32. Este proceso es un poco más difícil que la instalación del ESP8266 porque este último utiliza el administrador de la placa Arduino y solo requiere especificar una dirección web para descargar los controladores del dispositivo directamente [28].

Luego de poder acceder a la placa de desarrollo (se selecciona la placa "ESP Dev Module" en Herramientas), se instala el software necesario para poder conectar el ESP32 a una red WiFi. Se debe incluir la biblioteca "WiFi.h". En el menú

IDE, ir a Programa > Incluir una biblioteca > Administrar bibliotecas. En el buscador escribe WiFi. La librería instalada se muestra en la imagen. Este paso siempre se realizará cada vez que se instale una nueva biblioteca para su uso [28].

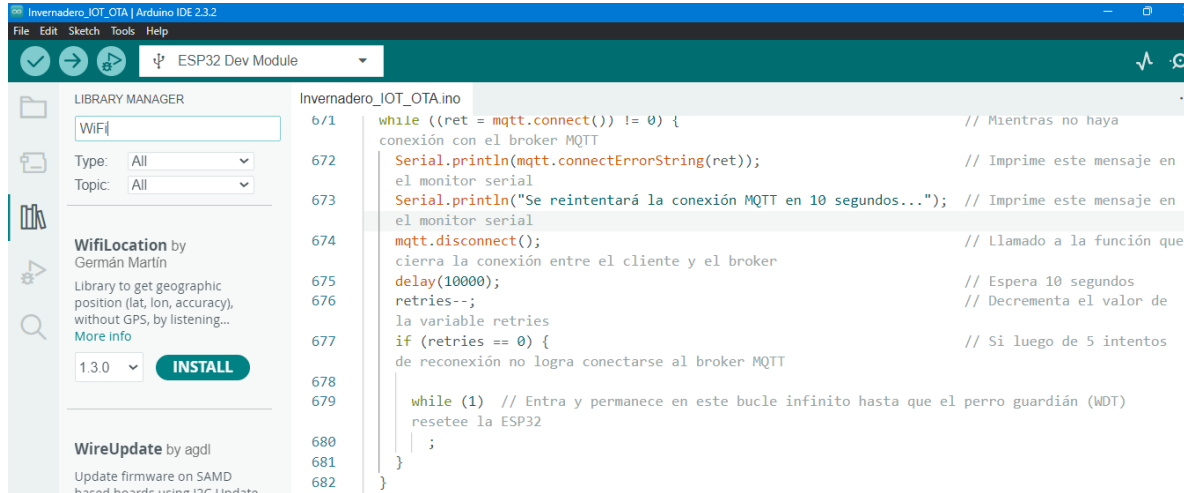


Figura 33. Gestor de Librerías IDE de Arduino. Elaboración propia.

Una vez completado esto, el siguiente paso será ingresar algún código en la función de Configuración para conectar el dispositivo a la red WiFi. En este punto, se configura el sistema de tal manera que la conexión se produzca inmediatamente después de la inicialización del código, sin necesidad de finalizar el módulo [28].

Es importante tener en cuenta que el intento de conexión se cancelará si no se puede establecer una conexión en menos de 25 segundos, lo que generalmente se considera tiempo suficiente [28].

Por otro lado, aunque se entienda bien qué es el protocolo IP, es necesario hacer un esfuerzo por aprender cuáles son las direcciones IP, ya que una de ellas se utilizará para conectar el dispositivo a la red [28].

### 4.3. SERVIDOR WEB

#### 4.3.1. PROTOCOLO DE APLICACIÓN MQTT

##### 4.3.1.1. ¿QUÉ ES MQTT?

MQTT es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para la comunicación de un equipo a otro. Los sensores

inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado. Estos dispositivos IoT utilizan MQTT para la transmisión de datos, ya que resulta fácil de implementar y puede comunicar datos IoT de manera eficiente. MQTT admite la mensajería entre dispositivos a la nube y la nube al dispositivo [29].

El protocolo MQTT se ha convertido en un estándar para la transmisión de datos de IoT, ya que ofrece los siguientes beneficios:

#### **4.3.1.2. LIGERO Y EFICIENTE**

La implementación de MQTT en el dispositivo IoT requiere recursos mínimos, por lo que se puede usar incluso en pequeños microcontroladores. Por ejemplo, un mensaje de control MQTT mínimo puede tener tan solo dos bytes de datos. Los encabezados de los mensajes MQTT también son pequeños para poder optimizar el ancho de banda de la red [29].

#### **4.3.1.3. ESCALABLE**

La implementación de MQTT requiere una cantidad mínima de código que consume muy poca energía en las operaciones. El protocolo también tiene funciones integradas para admitir la comunicación con una gran cantidad de dispositivos IoT. Por tanto, puede implementar el protocolo MQTT para conectarse con millones de estos dispositivos [29].

#### **4.3.1.4. FIABLE**

Muchos dispositivos IoT se conectan a través de redes celulares poco fiables con bajo ancho de banda y alta latencia. MQTT tiene funciones integradas que reducen el tiempo que tarda el dispositivo IoT en volver a conectarse con la nube. También define tres niveles diferentes de calidad de servicio a fin de garantizar la fiabilidad para los casos de uso de IoT: como máximo una vez (0), al menos una vez (1) y exactamente una vez (2) [29].

#### **4.3.1.5. SEGURO**

MQTT facilita a los desarrolladores el cifrado de mensajes y la autenticación de dispositivos y usuarios mediante protocolos de autenticación modernos, como OAuth, TLS1.3, certificados administrados por el cliente, etc [29].

#### **4.3.1.6. ADMITIDO**

Varios lenguajes, como Python, tienen un amplio soporte para la implementación del protocolo MQTT. Por lo tanto, los desarrolladores pueden implementar rápidamente con una codificación mínima en cualquier tipo de aplicación [29].

#### **4.3.1.7. ¿CUÁL ES EL PRINCIPIO EN EL QUE SE BASA MQTT?**

El protocolo MQTT funciona según los principios del modelo de publicación o suscripción. En la comunicación de red tradicional, los clientes y servidores se comunican directamente entre sí. Los clientes solicitan recursos o datos del servidor, a continuación, el servidor procesa y envía una respuesta. Sin embargo, MQTT utiliza un patrón de publicación o suscripción para desacoplar el remitente del mensaje (editor) del receptor del mensaje (suscriptor). En lugar de ello, un tercer componente, denominado agente de mensajes, controla la comunicación entre editores y suscriptores. El trabajo del agente consiste en filtrar todos los mensajes entrantes de los editores y distribuirlos correctamente a los suscriptores [29].

#### **4.3.1.8. CLIENTE MQTT**

Un cliente MQTT es cualquier dispositivo, desde un servidor hasta un microcontrolador, que ejecuta una biblioteca MQTT. Si el cliente envía mensajes, actúa como editor, y si recibe mensajes, actúa como receptor. Básicamente, cualquier dispositivo que se comunique mediante MQTT a través de una red puede denominarse dispositivo cliente MQTT [29].

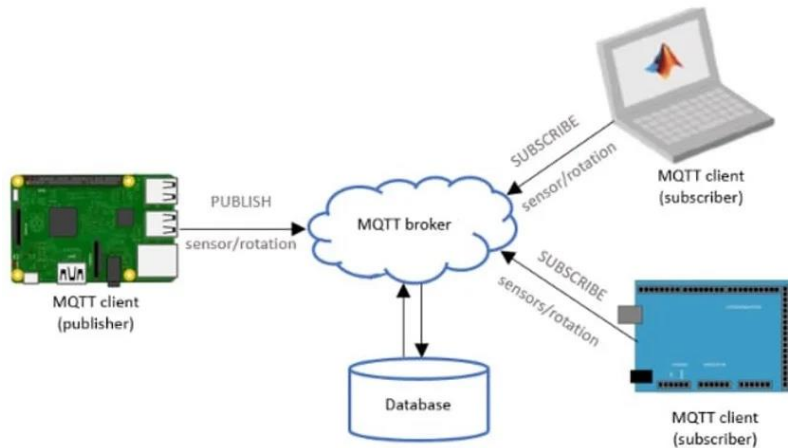


Figura 34. Protocolo de comunicación MQTT – Esquema ilustrativo [30].

#### 4.3.1.9. AGENTE MQTT

El agente MQTT es el sistema de back-end que coordina los mensajes entre los diferentes clientes. Las responsabilidades del agente incluyen recibir y filtrar mensajes, identificar a los clientes suscritos a cada mensaje y enviarles los mensajes. También se encarga de otras tareas como:

- La autorización y autenticación de clientes MQTT
- Pasar mensajes a otros sistemas para su posterior análisis
- El control de mensajes perdidos y sesiones de clientes [29].

#### 4.3.1.10. CONEXIÓN MQTT

Los clientes y los agentes comienzan a comunicarse mediante una conexión MQTT. Los clientes inician la conexión al enviar un mensaje CONECTAR al agente MQTT. El agente confirma que se ha establecido una conexión al responder con un mensaje CONNACK. Tanto el cliente MQTT como el agente requieren una pila TCP o IP para comunicarse. Los clientes nunca se conectan entre sí, solo con el agente [29].

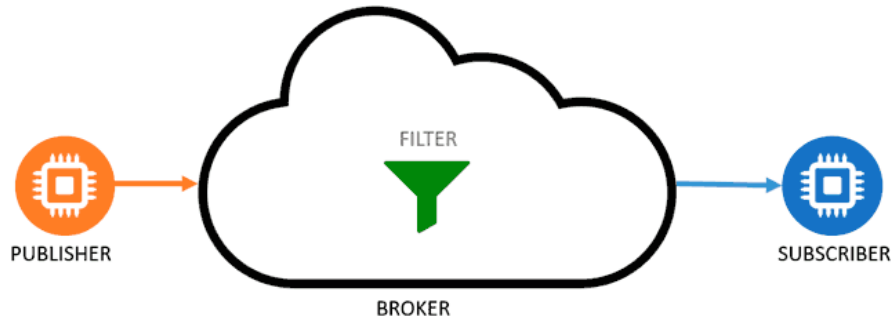


Figura 35. Funcionamiento MQTT [30].

A continuación, se proporciona una descripción general del funcionamiento de MQTT.

1. Un cliente MQTT establece una conexión con el agente MQTT.
2. Una vez conectado, el cliente puede publicar mensajes, suscribirse a mensajes específicos o hacer ambas cosas.
3. Cuando el agente MQTT recibe un mensaje, lo reenvía a los suscriptores que están interesados [29].

#### 4.3.1.11. TEMA DE MQTT

De acuerdo con [29], el término “tema” se refiere a las palabras clave que utiliza el agente MQTT a fin de filtrar mensajes para los clientes de MQTT. Los temas están organizados jerárquicamente, de forma similar a un directorio de archivos o carpetas. Por ejemplo, considere un sistema doméstico inteligente que opera en una casa de varios pisos que tiene diferentes dispositivos inteligentes en cada uno de ellos. En ese caso, es posible que el agente MQTT organice temas como:

ourhome/groundfloor/livingroom/light

ourhome/firstfloor/kitchen/temperatura

#### 4.3.1.12. PUBLICACIÓN MQTT

Los clientes MQTT publican mensajes que contienen el tema y los datos en formato de bytes. El cliente determina el formato de los datos, como datos de texto,

datos binarios, archivos XML o JSON. Por ejemplo, es posible que una lámpara del sistema doméstico inteligente publique un mensaje sobre el tema salón o luz [29].

#### 4.3.1.13. SUSCRIPCIÓN MQTT

Los clientes MQTT envían un mensaje para SUBSCRIBE (SUBSCRIBIRSE) al agente MQTT para recibir mensajes sobre temas de interés. Este mensaje contiene un identificador único y una lista de suscripciones. Por ejemplo, la aplicación de hogar inteligente en su teléfono quiere mostrar cuántas luces están encendidas en casa. Se suscribirá a la luz del tema y aumentará el contador para todos los mensajes activados [29].

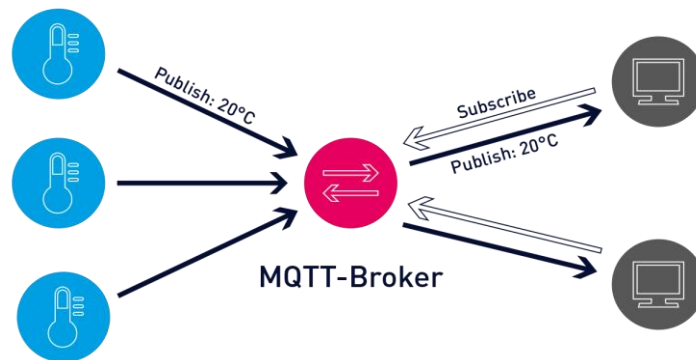


Figura 36. Suscripción MQTT [30].

#### 4.3.1.14. ¿MQTT ES SEGURO?

La comunicación MQTT utiliza el protocolo SSL para proteger los datos confidenciales que transmiten los dispositivos IoT. Puede implementar la identidad, la autenticación y la autorización entre los clientes y el agente mediante certificados SSL y contraseñas. El agente MQTT normalmente autentica a los clientes mediante sus contraseñas, así como los identificadores de cliente únicos que asigna a cada cliente. En la mayoría de las implementaciones, el cliente autentica el servidor con

certificados o búsquedas de DNS. También puede implementar protocolos de cifrado con MQTT [29].

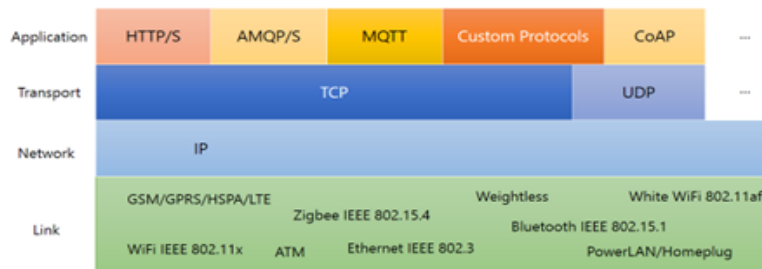


Figura 37. MQTT es un protocolo pensado para IoT que está al mismo nivel que HTTP o CoAP [30].

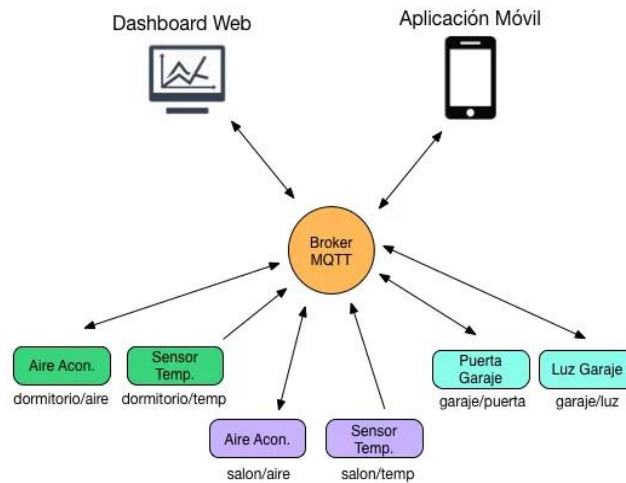


Figura 38. Suscripción y publicación de datos [30].

En esta forma de comunicación se desacoplan los clientes que publican (**Publisher**) de los que consumen los datos (**Suscribers**). Eso significa que los clientes no se conocen entre ellos unos publican la información y otros simplemente la consumen, simplemente todos tienen que conocer al message broker [29].

El desacoplamiento se produce en tres dimensiones:

- **En el espacio:** El publicador y el suscriptor no tienen por qué conocerse. No hace falta saber la dirección IP del contrario, ubicación, ni nada, el publicador manda el dato y lo entrega el broker. Muy interesante si cambio el equipo suscriptor de IP o de ubicación.

- **En el tiempo:** El publicador y el suscriptor no tienen por qué estar conectados en el mismo momento. Como el email, que lee el dato publicado cuando el suscriptor está disponible, pero necesita el retained u otro método con persistencia de los admitidos por MQTT.
- **En la sincronización:** las operaciones en cualquiera de los dos componentes no quedan interrumpidas mientras se publican o se reciben mensajes.

Es precisamente el broker el elemento encargado de gestionar la red y transmitir los mensajes [29].

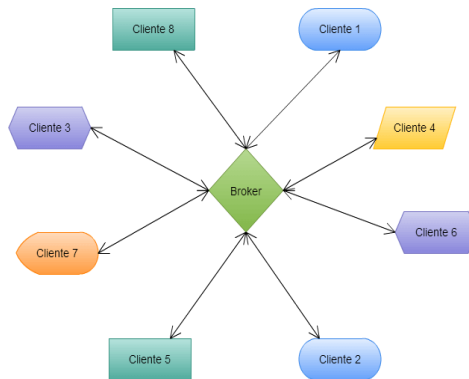


Figura 39. Broker encargado de gestionar la red y transmitir los mensajes [30].

Tabla 7 Tipos de mensajes del protocolo MQTT [30].

Control packet	Direction of flow	Description
CONNECT	Client to Server	Client request to connect to Server
CONNACK	Server to Client	Connect acknowledgment
PUBLISH	Client to Server or Server to Client	Publish message
PUBACK	Client to Server or Server to Client	Publish acknowledgment
PUBREC	Client to Server or Server to Client	Publish received (assured delivery part 1)
PUBREL	Client to Server or Server to Client	Publish release (assured delivery part 2)
PUBCOMP	Client to Server or Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	Client to Server	Client subscribe request
SUBACK	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	Client to Server	Unsubscribe request
UNSUBACK	Server to Client	Unsubscribe acknowledgment
PINGREQ	Client to Server	PING request
PINGRESP	Server to Client	PING response
DISCONNECT	Client to Server	Client is disconnecting

El identificador de cliente (ClientId) identifica a cada cliente MQTT que se conecta a un broker MQTT. El broker usa el ClientID para identificar al cliente y el

estado actual del cliente. Por lo tanto, este ID debe ser único por cliente y broker. En MQTT 3.1.1, puede enviar un ClientId vacío, si no necesita que el broker mantenga un estado. El ClientId vacío da como resultado una conexión sin ningún estado. En este caso, el indicador “clean session” debe establecerse en verdadero o el broker rechazará la conexión [30].

El parámetro de ClientId es muy importante cuando se requiere tener persistencia en datos y que el cliente pueda recuperar datos no obtenidos mientras estaba desconectado, usado junto la QoS2. Hay que recordar que cada cliente que se conecte a un broker debe tener un ClientId diferente [30].

#### **4.4. PROGRAMACIÓN INALÁMBRICA ESP32 (OTA): ACTUALIZADOR WEB ARDUINO IDE**

La programación OTA es útil cuando necesita actualizar el código de placas ESP32 a las que no se puede acceder fácilmente. El ejemplo que mostraremos aquí funciona cuando el ESP32 y su navegador están en su red local [31].

La única desventaja del Actualizador web OTA es que debe agregar el código de OTA en cada boceto que cargue, para poder utilizar OTA en el futuro [31].

##### **4.1.1. REALIZAR ACTUALIZACIONES POR AIRE (OTA) UTILIZANDO EL MICROCONTROLADOR ESP32 Y SU ESP-IDF**

Los diseñadores de productos del Internet de las Cosas (IoT) deben evaluar continuamente la selección de plataformas y componentes con vistas a reducir el costo y la potencia, al tiempo que mejoran el rendimiento y aceleran el diseño de las aplicaciones de conectividad. En la actualidad hay bastantes soluciones entre las que elegir, pero los diseñadores se enfrentan al reto de realizar actualizaciones inalámbricas por aire (OTA) para mantener el firmware del dispositivo al día, una vez desplegado [32].

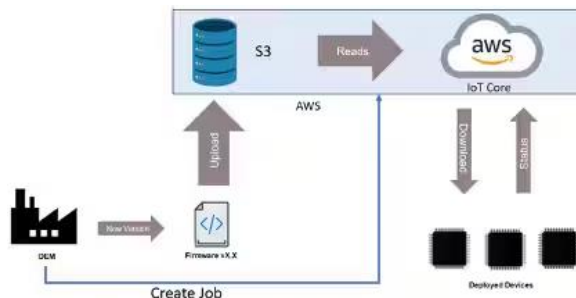
##### **4.4.2. INTRODUCCIÓN A LAS ACTUALIZACIONES OTA**

El objetivo principal de la mayoría de los equipos de desarrollo es implementar las características específicas de su producto, es decir, la lógica

empresarial que lo diferencia. Sin embargo, todos los productos del IoT tienen un conjunto de características básicas que deben ser desplegadas, configuradas y mantenidas a lo largo de la vida útil del dispositivo. Las actualizaciones de seguridad son un buen ejemplo. Dada la necesidad de llevar a cabo estas actualizaciones, una característica importante, pero que fácilmente se pasa por alto a la hora de evaluar una plataforma de desarrollo adecuada es la capacidad de actualización del cargador de arranque o del firmware OTA (a veces simplemente denominado OTA) [32].

La OTA ofrece a los ingenieros la posibilidad de mantener y actualizar a distancia sus productos en respuesta a los requisitos técnicos y empresariales sin necesidad de enviar personal de mantenimiento al dispositivo o de que el cliente final haga algo activamente con el dispositivo para actualizarlo. En cambio, todos esos costes pueden eliminarse haciendo que los dispositivos actualicen silenciosamente su firmware en segundo plano, o durante las horas de "inactividad" operativa, como la mitad de la noche [32].

Las arquitecturas de la OTA pueden adoptar muchas formas y configuraciones diferentes, desde soluciones personalizadas hasta implementaciones estándar suministradas por el proveedor de la nube [32].



**Figura 40. Visión general de la arquitectura OTA que muestra un proceso de ejemplo para actualizar el firmware de la aplicación sobre el terreno en los dispositivos desplegados [32].**

En este ejemplo, un fabricante de equipos originales utiliza Amazon Web Services (AWS) IoT Core para cargar nuevas versiones de firmware y, a continuación, utiliza las funciones de trabajo integradas para desplegar las

actualizaciones en los dispositivos sobre el terreno. Este es sólo uno de los muchos ejemplos, y casi todos los proveedores de la nube tienen una solución similar [32].

Hay muchas opciones de microcontroladores disponibles hoy en día que soportan OTA. Un microcontrolador muy popular tanto para sistemas de bajo coste como entre los fabricantes es el ESP32. Hay varias razones por las que el ESP32 ha sido tan popular, entre ellas:

- Tiene un microcontrolador integrado con módulos de certificación Wi-Fi/Bluetooth disponibles
- Bajo costo
- Entorno de desarrollo de código abierto y marcos de software como ESP-IDF y ESP Audio Development Framework (ESP-ADF)
- Numerosos ejemplos de aplicaciones existentes disponibles gratuitamente en la web [32].

#### **4.4.3. EL FLUJO DE TRABAJO DE DESARROLLO DE LA OTA**

El flujo de trabajo de desarrollo de la OTA para el ESP32 variará ligeramente en función de las necesidades del negocio y de la selección de componentes del producto. Por ejemplo, un equipo que utilice AWS probablemente se basará en las guías de inicio y en los ejemplos de AWS para implementar su solución OTA en el ESP32. En cambio, una empresa que personalice su propia solución aprovechará la documentación oficial del ESP32. En este trabajo se examinan las piezas a nivel del ESP32, y no en la nube, dado que son componentes genéricos aplicables a la OTA con el ESP32, independientemente del proveedor de nube o la solución empleada [32].

En general, según [32], el proceso para configurar una actualización OTA en el ESP32 implica los siguientes pasos:

- Configurar la tabla de particiones del ESP32
- Descargar el firmware que soporta OTA
- Desarrollar una herramienta que actúe como servidor y empuje el nuevo firmware

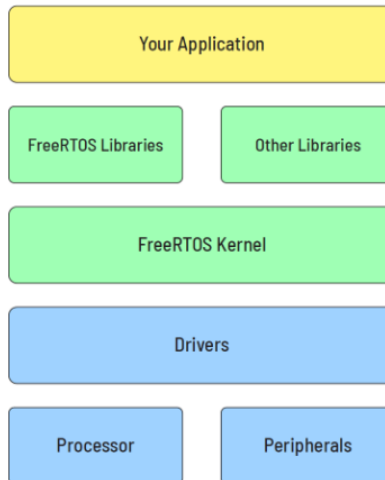
- Descargue el último firmware en el ESP32
- Cambiar a la nueva aplicación

Las actualizaciones OTA son una característica crítica para un número cada vez mayor de sistemas IoT e integrados. Los desarrolladores deben saber cómo hacerlo de forma eficaz para ahorrar tiempo durante el proceso de diseño y desarrollo, y después de que el producto haya sido enviado [32].

El microcontrolador inalámbrico ESP32 se ha abierto camino en una amplia gama de dispositivos, y como se ha mostrado tiene una solución OTA lista. Al aprovechar el ESP-IDF y los módulos y plataformas asociados, y utilizando algunos consejos y trucos basados en la experiencia, los desarrolladores pueden aligerar drásticamente su tiempo de diseño y poner en marcha su propia solución OTA [32].

#### **4.5. FREERTOS Y ESP32**

ESP32 es un SoC potente y rico en funciones de Espressif. Parte de la razón por la que nos gusta tanto es que es un SoC de doble núcleo con un reloj operativo rápido y una gran capacidad de memoria. Dos núcleos significan que hay dos procesadores separados dentro del chip SoC. Para aprovechar al máximo estas increíbles funciones de hardware, la típica programación secuencial no es suficiente [33].



**Figura 41. Organización FreeRTOS [33].**

La programación o ejecución secuencial es cuando se ejecuta una tarea mientras todas las demás esperan a que finalice la tarea en ejecución. Por el contrario, si piensa en su computadora personal, sabrá que puede realizar múltiples tareas a la vez, como si cada tarea/aplicación se ejecutara en paralelo e independientemente de la otra. Si bien la programación secuencial puede hacer eso si se ejecuta lo suficientemente rápido, existen ventajas reales al utilizar el "verdadero paralelismo" con múltiples núcleos [33].

FreeRTOS es un sistema operativo en tiempo real ligero y de código abierto. Por lo tanto, se adapta perfectamente a los problemas de gestión de tareas en función del tiempo o de ocurrencia de eventos. Espressif ha integrado el sistema operativo en las últimas versiones del SDK. Por lo tanto, es posible utilizar la funcionalidad de este sistema operativo para explotar el potencial de los dos núcleos del NodeMCU ESP32 con el IDE de Arduino. [33].

La parte central de un RTOS es un kernel (un programa de supervisión con los privilegios más altos del sistema). El kernel se encarga de todo, incluidos tiempos, colas, sincronización de tareas, prioridades, interrupciones y más. FreeRTOS puede funcionar tanto en un entorno de un solo núcleo como de varios núcleos [33].



**Figura 42. Sistema operativo RTOS [34].**

En caso de disponer de varios núcleos, se pueden ejecutar tareas diferentes en cada uno de ellos. Pero si solo hay un núcleo, FreeRTOS divide el tiempo del procesador en diferentes cuantos (una unidad mínima de tiempo, generalmente un intervalo de tiempo muy pequeño) y permite que cada tarea consuma los intervalos de tiempo periódicamente. Eso significa que se podrán ejecutar múltiples tareas RTOS paralelas en el mismo núcleo sin ninguna preocupación, tal como se ejecuta el `loop()` y `loop2()` en el mismo núcleo [33].

## CAPÍTULO V

### 5.1. DISEÑO DEL SISTEMA – VISIÓN GLOBAL DEL HARDWARE

En el siguiente diagrama, se presentan cada uno de los bloques que integra el sistema:

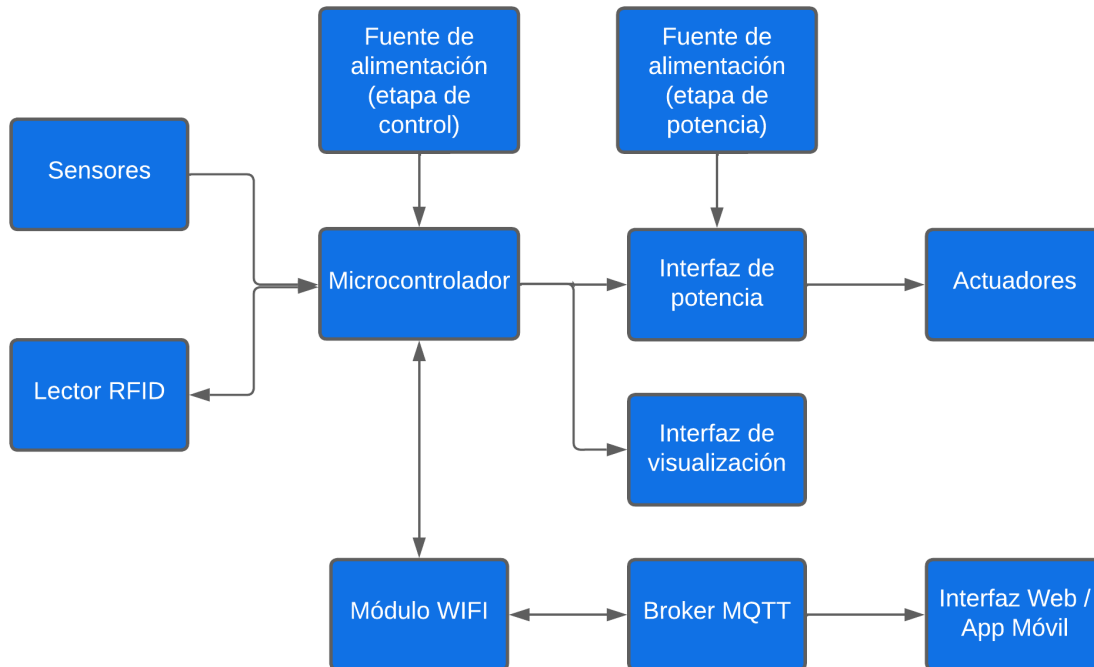


Figura 43. Diagrama de bloques del sistema. Elaboración Propia.

Como se observa en el diagrama anterior, el sistema está conformado por bloques funcionales que incluyen **módulos de entrada**, una **unidad de procesamiento**, **módulos de salida** y **fuentes de alimentación**. A continuación, se detalla el funcionamiento de cada bloque o etapa:

- **Sensores:** este bloque agrupa un conjunto de dispositivos encargados de adquirir datos de variables ambientales, los cuales serán procesados por el microcontrolador para ejecutar acciones de control en el invernadero. En el sistema propuesto se emplearán un sensor de temperatura y humedad relativa (DHT22), un sensor de humedad de suelo (FC-28), un sensor de nivel

de luz (LDR) y un sensor ultrasónico (HC-SR04) para la medición del nivel de agua. También se dispone de un botón pulsador para abrir la puerta desde el interior del invernadero.

- **Lector RFID:** esta etapa incorpora un módulo lector RFID (RC522) para la gestión del control de acceso seguro al interior del invernadero. Únicamente se permitirá el ingreso mediante un teléfono, llavero o tarjeta RFID previamente registrada en el sistema.
- **Microcontrolador:** es el núcleo de procesamiento encargado de recibir y analizar los datos provenientes de los sensores y del lector RFID, para posteriormente generar señales de control hacia los actuadores, de acuerdo con los parámetros y configuraciones definidos en el programa del sistema. La plataforma de desarrollo seleccionada es la ESP32, por su capacidad de procesamiento y conectividad integrada.
- **Módulo WiFi:** proporciona conectividad inalámbrica al sistema, permitiendo su monitoreo y control remoto desde cualquier ubicación con acceso a Internet. En este proyecto se utiliza el módulo WiFi incorporado en la placa ESP32.
- **Interfaz de visualización:** compuesta por una pantalla LCD 16x2, permite mostrar información relevante del sistema, como mensajes asociados al control de acceso, indicando si la autorización fue concedida o denegada.
- **Interfaz de potencia:** integrada por un módulo de relés, establece un aislamiento eléctrico seguro entre la etapa de control y la etapa de potencia, posibilitando la activación de actuadores que operan con tensiones superiores a 5 V DC.
- **Actuadores:** el sistema cuenta con dos electroválvulas de 12 V DC, una destinada al control del riego y otra al llenado del tanque; una cerradura eléctrica de 12 V DC para la puerta del invernadero; y un ventilador de 12 V DC para la regulación de la temperatura del microclima interno. También se dispone un buzzer activo que alerta cuando se deniega el acceso al invernadero a un *tag* RFID no registrado.

- **Broker MQTT:** actúa como servidor central en la arquitectura de comunicación bajo el protocolo MQTT, cuya función es recibir, filtrar y distribuir mensajes entre los diferentes clientes (sensores, aplicaciones y microcontroladores), conforme al modelo publicador–suscriptor. En este caso, se utiliza el servicio de Adafruit IO como broker MQTT, permitiendo el envío de datos de sensores a la nube y el control remoto del sistema.
- **Interfaz web / aplicación móvil:** estas interfaces posibilitan la creación de un *dashboard* para la visualización gráfica de las variables ambientales y el control remoto del invernadero, ya sea en modo manual o automático. Se emplea la interfaz web de Adafruit IO y la aplicación móvil MQTT Dash, lo que permite el acceso desde teléfonos inteligentes, tabletas o computadoras personales.
- **Fuentes de alimentación:** suministran la energía necesaria para el funcionamiento del sistema. Se dispone de una fuente de 5 V DC para la alimentación de la etapa de control (microcontrolador y módulos de baja tensión) y una fuente de 12 V DC para la etapa de potencia (actuadores).

En las siguientes figuras se presenta el diagrama de conexiones del sistema, tanto en su vista esquemática como en su implementación sobre protoboard. Estas representaciones permiten identificar la disposición de los componentes, sus interconexiones y la correspondencia entre el diseño conceptual y el montaje físico utilizado durante la etapa de pruebas.

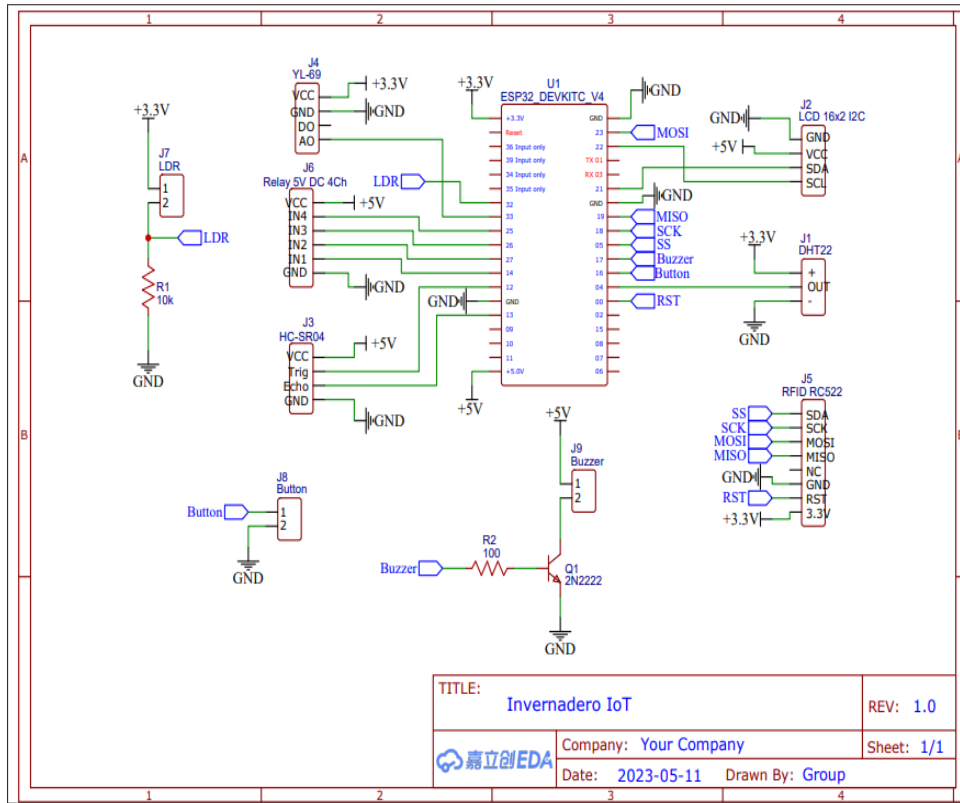


Figura 44. Diagrama esquemático de conexiones del sistema. Elaboración Propia.

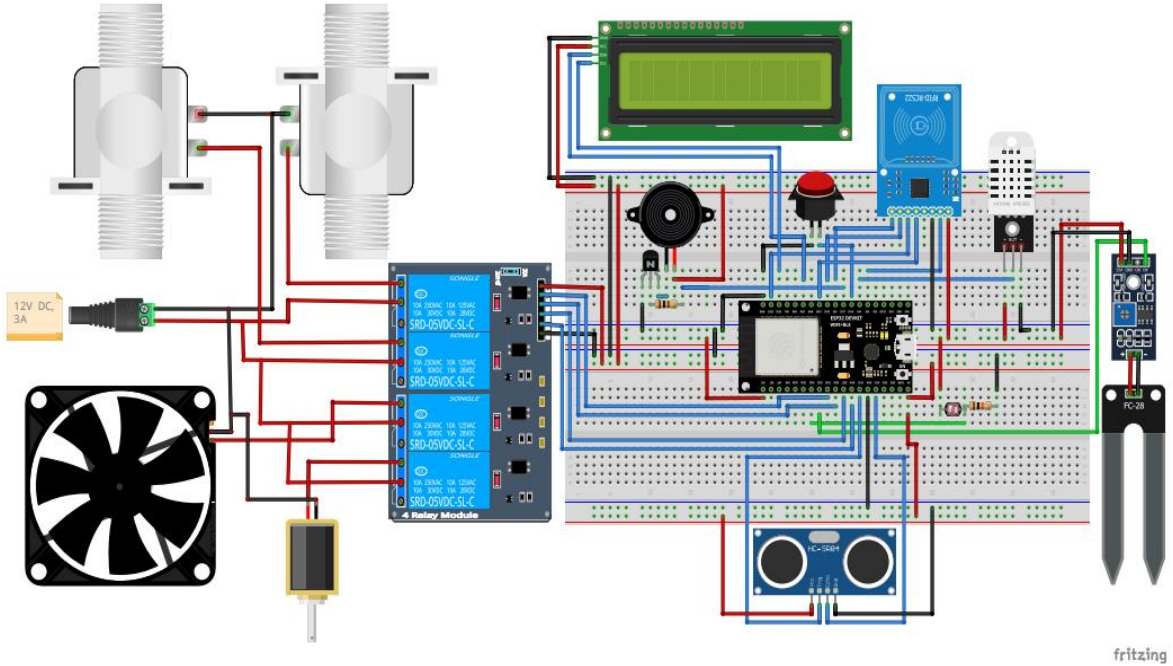


Figura 45. Diagrama de conexiones del sistema (vista protoboard). Elaboración Propia.

## 5.2. EI SOFTWARE - VISIÓN GENERAL

Para tener una visión más clara y precisa del desarrollo del código, este puede ser agrupado en varias etapas que permitan entender el funcionamiento del programa y el proceso general del mismo. En este sentido, dentro del código

Se pueden identificar distintos subprocesos, subrutinas y tareas que conforman el software del proyecto. A continuación, se presentan las diferentes secciones que constituyen el esquema general del programa.

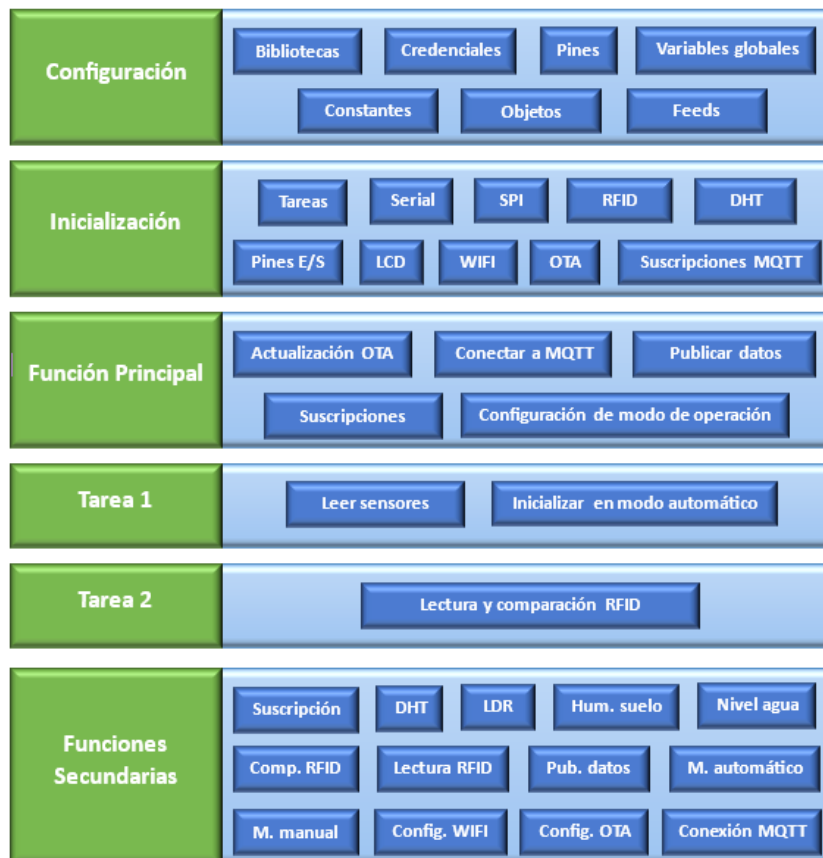


Figura 46. Visión general del software. Elaboración propia

### CONFIGURACIÓN

En esta parte inicial del código, se establece la configuración principal del mismo; es decir, se incluyen las bibliotecas o librerías, se definen las credenciales MQTT y los pines, se declaran las variables globales y las constantes, se crean los objetos y, finalmente, se configuran los feeds de la plataforma MQTT. A

continuación, se detallan cada uno de los elementos de programación que constituyen la etapa de configuración del código de este proyecto.

## **BIBLIOTECAS**

Una biblioteca o librería es un conjunto de archivos de código prescritos que pueden ser utilizados por los programadores para facilitar el desarrollo de software. Estas bibliotecas proporcionan a los usuarios una serie de métodos, funciones y clases predefinidas para realizar tareas comunes o complejas, como leer los datos de un sensor digital o controlar un módulo de pantalla electrónica.

En el presente proyecto, se emplean 10 bibliotecas con las siguientes funcionalidades:

- Configurar y establecer la conexión WIFI de la ESP32 a una red desde un servidor web, así como almacenar las credenciales WIFI en la memoria EEPROM.
- Generar un cliente WIFI.
- Crear un servidor web.
- Actualizar el firmware de la ESP32 vía OTA desde un servidor web.
- Establecer conexión al Servidor Adafruit vía MQTT.
- Generar un cliente MQTT.
- Controlar un módulo de pantalla LCD con I2C.
- Establecer comunicación vía SPI.
- Manejar un módulo lector RFID MFRC522.
- Leer datos de temperatura y humedad relativa de un sensor DHT.

## **CREDENCIALES DE ADAFRUIT**

Se configuran las credenciales del Servidor Adafruit IO. Como se abordó anteriormente, se define la dirección del servidor, el puerto, el usuario y la llave de

seguridad para acceder a los feeds creados en el tablero a través del cliente (en este caso, la placa ESP32).

## **PINES**

En esta instancia, se definen los pines digitales y analógicos empleados de la placa de desarrollo ESP32, donde estarán conectados los sensores, módulos y actuadores.

Los pines utilizados son los siguientes:

- Sensor DHT22 – GPIO 4 / ADC2\_0 / Empleado como entrada digital
- Fotorresistencia o LDR – GPIO 32 / ADC1\_4 / Empleado como entrada analógica
- Sensor de humedad de suelo – GPIO 33 / ADC1\_5 / Empleado como entrada analógica
- Sensor ultrasónico – GPIO 12 / ADC2\_5 / Empleado como salida digital (Trigger)
- Sensor ultrasónico – GPIO 13 / ADC2\_4 / Empleado como entrada digital (Echo)
- Módulo relé electroválvula 1 – GPIO 25 / ADC2\_8 / Empleado como salida digital
- Módulo relé electroválvula 2 – GPIO 26 / ADC2\_9 / Empleado como salida digital
- Módulo relé ventilador – GPIO 27 / ADC2\_7 / Empleado como salida digital
- Módulo RFID MFRC522 – GPIO 0 / ADC2\_1 / Pin de reinicio (RST)
- Módulo RFID MFRC522 – GPIO 5 / SS / Pin de datos SPI (SS o SDA)
- Módulo RFID MFRC522 – GPIO 18 / SCK / Pin de reloj SPI (SCK)

- Módulo RFID MFRC522 – GPIO 23 / MOSI / Pin de entrada de datos SPI (MOSI)
- Módulo RFID MFRC522 – GPIO 19 / MISO / Pin de salida datos SPI (MISO)
- Buzzer – GPIO 17 / TXD 2 / Empleado como salida digital
- Pulsador – GPIO 16 / RXD 2 / Empleado como salida digital
- Display LCD 16x2 – GPIO 21 / SDA / Pin de datos I2C (SDA)
- Display LCD 16x2 – GPIO 22 / SCL / Pin de reloj I2C (SCL)

## **VARIABLES GLOBALES**

Una variable es un espacio reservado en memoria que se utiliza para almacenar un valor que puede ser modificado durante la ejecución de un programa. Las variables tienen un nombre que se utiliza para referirse a ellas en el código y un tipo de datos que determina el tipo de valor que se puede almacenar en ellas, como números, cadenas de texto, valores booleanos, entre otras [35].

Las variables pueden ser de tipo global o local. La diferencia principal entre ambas es que la primera puede ser accedida desde cualquier parte del programa, mientras que la segunda solo puede ser accedida dentro del bloque de código donde se ha definido [35]. En este proyecto, se declararon variables globales con diferentes funcionalidades, que se pueden agrupar de la siguiente manera:

- **MEDICIÓN O LECTURA:**

Son variables para almacenar los datos de los sensores o del lector RFID. Estos datos podrán ser de tipo “int” (entero) y de tipo “byte” (8 bits). Las variables de tipo entero almacenan los datos proporcionados por los sensores y fueron inicializadas en cero. Por su parte, el array de tipo byte guarda el identificador único (UID, por sus siglas en inglés) de la tarjeta leída por el módulo RFID y tiene una longitud de 4 elementos, es decir, es una colección de 4 variables del mismo tipo.

- **CONTROL DE TIEMPO**

Se han declarado variables para almacenar y controlar los tiempos, en milisegundos, de publicación de datos en los diferentes tópicos del Servidor MQTT. Estas variables son de tipo “unsigned long” (entero sin signo) y fueron inicializadas en cero.

- **SUSCRIPCIÓN (ESTADO DE BOTONES)**

Son variables de tipo “String” (cadena de caracteres) y se utilizan para almacenar el valor de los tópicos a los que está suscrita la ESP32, siendo estos el estado de los botones de la plataforma MQTT. Dicho estado puede ser “ON” (encendido) u “OFF” (apagado). Cabe mencionar que estas variables fueron inicializadas en estado “OFF”.

- **IDENTIFICACIÓN DE TAREAS**

Son variables de tipo TaskHandle\_t, el cual es un tipo de dato definido en la biblioteca de FreeRTOS, que se utiliza para manejar y controlar tareas en sistemas embebidos. Estas variables se utilizan para almacenar los identificadores de las tareas que se van a crear en el sistema. Una vez que se han declarado estas variables, se pueden utilizar para iniciar las tareas y asignarles una función y parámetros específicos.

## **CONSTANTES**

Como su nombre lo indica, una constante es un valor que no cambia durante la ejecución del programa. En esta sección del código, se definieron dos tipos de constantes que cumplen funciones distintas, las cuales se detallan a continuación.

- **UID DE USUARIOS**

Son constantes de tipo “byte” y fueron definidas como un array o arreglo de 4 elementos. Estas se emplean para almacenar el identificador único o UID de los usuarios (dos, en este caso) que podrán acceder al invernadero.

- **PÁGINAS WEB DE ACTUALIZACIONES OTA**

Estas constantes son de tipo “const char\*” y se utilizan para almacenar el contenido de la página web de inicio de sesión, así como de la página web que permite actualizar el firmware de la ESP32 usando OTA. El contenido de tales páginas se compone de código en HTML, que define la estructura de la página, y código en CSS, que define el estilo y la apariencia de la página.

## **OBJETOS**

Un objeto es una instancia de una clase. A su vez, una clase es una plantilla que define las propiedades y los métodos que tendrán los objetos que se creen a partir de ella [36]. Por ejemplo, en Arduino es posible usar la biblioteca DHT para leer los datos de temperatura y humedad relativa de un sensor DHT. Para crear un objeto que represente al sensor DHT, se puede declarar una variable de tipo DHT, como sigue: `DHT dht(DHTPIN, DHTTYPE);`

En este caso, `dht` es nombre del objeto y `DHT` es la clase a la que pertenece. Además, se debe especificar el pin al que está conectado el sensor y el tipo de sensor utilizado. Luego de crear el objeto, se pueden usar los métodos de la clase `DHT` para leer los datos temperatura y humedad relativa provistos por el sensor.

En esta etapa del código, se crearon 7 objetos con las siguientes finalidades:

- Objeto de la clase `WiFiManager` para configurar la red WIFI.
- Objeto de la clase `WebServer` para crear un servidor web en el puerto 80.
- Objeto de la clase `WiFiClient` para generar un cliente WIFI.
- Objeto de la clase `Adafruit_MQTT_Client` para generar un cliente MQTT que permita conectarse al servidor de Adafruit.
- Objeto de la clase `LiquidCrystal_I2C` para controlar la pantalla LCD.
- Objeto de la clase `DHT` para leer los valores de temperatura y humedad relativa del sensor DHT22.
- Objeto de la clase `MFRC522` para controlar el módulo lector RFID MFRC522.

## FEEDS

Se configuran los feeds o rutas de comunicación para realizar los procedimientos de publicación y suscripción en el broker MQTT. Las rutas MQTT de Adafruit deben seguir la siguiente forma: <username>/feeds/<feedname>. En esta instancia es donde se hará uso de las bibliotecas citadas anteriormente para la comunicación MQTT con el servidor web de Adafruit.

## INICIALIZACIÓN

En esta parte del código, se realiza la inicialización global del sistema cuando se energiza o se reinicia el mismo. Todo este proceso es llevado a cabo dentro de la función Setup, característica del IDE de Arduino para la configuración inicial de pines como entradas y salidas y otras inicializaciones, como la del puerto Serial para visualizar datos o gráficas en pantalla.

Para mayor comprensión del código implementado en la función Setup, se han distinguido diferentes inicializaciones que se describen a continuación:

- **Tareas:** Se inicializan dos tareas para su posterior ejecución en el programa. Se debe especificar la función que ejecutará cada tarea, el nombre de la tarea, el tamaño de la pila (en bytes) para la función, la prioridad de la tarea, el nombre de la variable de tipo TaskHandle\_t y, finalmente, el núcleo de la ESP32 donde se ejecutará la tarea.
- **Serial:** Se inicia la comunicación serial a 115200 baudios para mostrar datos por pantalla.
- **SPI:** Se inicializa la Interfaz en Serie para Periféricos (SPI, por sus siglas en inglés), mediante la cual se establecerá comunicación con el módulo RFID.
- **RFID:** Se inicializa el módulo lector RFID MFRC522.
- **DHT:** Se inicializa el sensor de temperatura y humedad relativa DHT22.
- **Pines de entrada y salida:** Se inicializan los pines de entrada de datos de los sensores, así como los pines de salida para controlar los actuadores.

Cabe destacar que en el IDE de Arduino no es necesario inicializar los pines analógicos de entrada de datos. Además, algunas de las bibliotecas empleadas en este proyecto se encargan de inicializar los pines utilizados por las mismas, como es el caso de la biblioteca DHT.

- **LCD:** Se inicializa la pantalla LCD con módulo I2C y se imprime un mensaje de presentación para la etapa de control de acceso del invernadero.
- **Conexión WIFI:** Se realiza un llamado a la función “WiFi\_config()”, donde se intenta conectar la ESP32 a una red WIFI cercana.
- **Actualizaciones OTA:** Se efectúa un llamado a la función “OTA\_config()”, que configura la programación o actualización de firmware de la placa vía OTA, mediante un servidor web local.
- **Canales de suscripción:** Se generan los canales de suscripción a los botones de Adafruit IO por MQTT. En este caso, se crean cuatro: el primero, para la activación del modo manual (modo); el segundo, para activación del riego (riego); el tercero, para la activación del llenado del tanque (llenado), y el cuarto, para la activación de la ventilación (ventilación). Cada uno de ellos posee dos estados: “ON” (activado) y “OFF” (desactivado). Cabe mencionar que, si el botón de modo se encuentra desactivado, el sistema se controlará de forma automática, según lo establecido en la programación.

## **FUNCIÓN PRINCIPAL**

La función principal o “loop”, en el caso del IDE de Arduino, es la parte del código que se ejecuta continuamente en un bucle infinito después de que el programa de Arduino se haya iniciado correctamente. Dentro de esta función, se suelen incluir las instrucciones que se ejecutan de forma repetitiva mientras el dispositivo está encendido.

En esta sección del código fuente, se han definido una serie de instrucciones que se ejecutarán para cumplir propósitos específicos, que se describen a continuación.

## **Actualizaciones OTA**

Se ejecuta repetidamente la función “server.handleClient()”, que forma parte de la biblioteca WebServer. Esta función se utiliza para manejar las solicitudes de clientes web entrantes en el servidor web local que permite iniciar sesión y posteriormente realizar actualizaciones en el software de la ESP32 de forma inalámbrica, usando OTA.

La función “server.handleClient()” debe ser llamada repetidamente dentro del bucle principal “void loop()” del programa para garantizar que el servidor web esté siempre disponible para procesar las solicitudes entrantes de los clientes. Si esta función no se llama de forma regular, el servidor web no podrá procesar las solicitudes entrantes y los clientes web no podrán conectarse ni recibir respuestas.

## **Establecer conexión con el broker MQTT**

Esta instrucción realiza un llamado a la función “MQTT\_connect()” y sólo se ejecuta cuando la ESP32 se encuentra conectada a una red WIFI (al igual que las otras instrucciones del “loop” que se detallarán más adelante). La función anterior se utiliza para establecer una conexión entre el cliente MQTT y el servidor o broker MQTT de Adafruit, para enviar y recibir datos a través de la red, empleando el protocolo MQTT.

## **Publicar Datos**

Esta instrucción invoca a la función “Publicar\_Datos()”, que permite publicar los datos de los sensores en el broker MQTT de Adafruit. Aunque la instrucción se ejecuta repetidamente dentro del “loop”, los datos de los sensores se publicarán cada 15 segundos, según lo definido el código de la subrutina “Publicar\_Datos()”.

## **Suscripciones MQTT**

Se ejecuta de forma repetida, cada 500 milisegundos, la función “mqtt.processPackets()”, que forma parte de la biblioteca Adafruit\_MQTT. Esta función se utiliza para procesar los paquetes entrantes y salientes en una conexión

MQTT, así como para garantizar que los paquetes se manejen de manera oportuna y que la conexión con el servidor MQTT se mantenga activa.

Además, la función “`mqtt.processPackets()`” invoca dentro del “loop” a las funciones Callback de suscripción a los botones de la plataforma MQTT, que recogen el estado de los mismos para el control de los actuadores.

### **Configuración del modo de operación**

En esta parte del “loop”, se establece el modo de operación del invernadero en función del valor de la variable “selec”, la cual almacena el estado del botón que permite activar o desactivar el modo manual. Si el estado de “selec” es “ON”, los actuadores del invernadero se controlarán de forma manual desde la plataforma MQTT de Adafruit. Por su parte, si el estado de “selec” es “OFF”, los actuadores del invernadero se controlarán de manera automática, de acuerdo con lo establecido en la programación.

### **Tareas**

Una tarea es una unidad de trabajo que realiza una parte específica de un programa. En programación, las tareas se utilizan para dividir un programa en unidades más pequeñas, facilitar la ejecución simultánea de múltiples tareas y mejorar la modularidad y el mantenimiento del código [37].

La ESP32 es dual core, ya que posee dos microprocesadores Xtensa LX6 de 32 bits: core 0 y core 1. Cuando se ejecuta código en Arduino IDE, este es ejecutado en el núcleo 1 de forma predeterminada. Sin embargo, se podrían crear tareas que se ejecuten simultáneamente en ambos núcleos y hacer que la ESP32 sea multitarea. Cabe destacar que no es necesario ejecutar tareas en ambos núcleos para lograr la multitarea.

Arduino IDE es compatible con FreeRTOS para ESP32, que es un sistema operativo en tiempo real. Esto permite manejar varias tareas en paralelo que se ejecutan de forma independiente en un sólo núcleo o en ambos.

En el código fuente del sistema, se aprovecha la capacidad multitarea de la ESP32, dado que, empleando FreeRTOS, se crearon dos tareas que se ejecutan en núcleo 0 de forma paralela a la ejecución de la función principal o “loop” (que se ejecuta en el núcleo 1). A continuación, se detallan las tareas creadas en el núcleo 0 de la ESP32 y las funciones que estas cumplen.

### **Tarea 1**

En esta primer tarea, se ejecuta repetidamente la función “void loop1()”, donde se hace una llamado a las funciones secundarias “Temp\_Hum()”, “Hum\_Suelo()”, “Luminosidad()” y “Nivel\_Agua()”, que se encargan de recoger los datos de las variables ambientales proporcionados por los sensores empleados en este proyecto. Además, si la ESP32 no se encuentra conectada a una red WIFI, se invoca a la subrutina “Automatico()” para controlar los actuadores del sistema de manera automática, de acuerdo a los umbrales establecidos en la programación para cada variable ambiental.

### **Tarea 2**

En esta segunda tarea, se ejecuta de forma repetida la función “void loop2()”, que hace un llamado a la función “Lectura\_Comp\_RFID()”. Esta última permite leer el UID de las etiquetas RFID de los usuarios que intenten acceder al invernadero, así como comparar los identificadores únicos leídos con los almacenados previamente para controlar el acceso al interior del invernadero.

### **Funciones secundarias**

Se han creado una serie de funciones secundarias, cada una con un conjunto de instrucciones concreto para realizar una tarea específica. Estas funciones fueron creadas para ser llamadas en la función principal, en las funciones ejecutadas por las tareas 1 y 2, e incluso dentro de otra función secundaria.

A continuación, se presentan las funciones secundarias creadas y se explica su funcionamiento de forma breve.

- **Funciones callback de suscripción:** Las funciones callback son aquellas que ejecutan cuando ocurre un evento específico. En este proyecto, se crearon cuatro funciones callback: “modoCallback()”, “riegoCallback()”, “llenadoCallback()” y “ventilacionCallback()”, las cuales se ejecutan, respectivamente, cuando ocurre un cambio en el estado de los botones de la plataforma MQTT que permiten activar o desactivar el modo manual, el riego del cultivo, el llenado del tanque y la ventilación. Básicamente, estas funciones almacenan en variables el estado de los botones (“ON” u “OFF”), que luego se utiliza para llevar a cabo acciones de control mediante los actuadores.
- **DHT – void Temp\_Hum():** Recoge los datos de temperatura y humedad relativa proporcionados por el sensor DHT22.
- **Porcentaje de luminosidad – void Luminosidad ():** Calcula el porcentaje de luminosidad en función de los valores analógicos provistos por la forresistencia o LDR.
- **Porcentaje de humedad de suelo – void Hum\_Suelo():** Calcula el porcentaje de humedad en el suelo del cultivo basándose en los valores de voltaje del sensor de humedad de suelo.
- **Nivel de agua – void Nivel\_Agua():** Calcula el porcentaje de nivel de agua presente en el tanque a partir de los valores de distancia proporcionados por el sensor ultrasónico.
- **Comparación de usuarios RFID – boolean comparaUID(byte lectura[], byte usuario[]):** Esta subrutina es llamada dentro de la función “Lectura\_Comp\_RFID()”, con el fin de comparar el UID de la etiqueta RFID leída con los UID autorizados para acceder al invernadero.
- **Lectura y comparación RFID – Void Lectura\_Comp\_RFID():** Lee los UID de las etiquetas RFID, luego los compara con los UID previamente registrados para permitir o denegar el acceso al invernadero. Además, muestra mensajes en la pantalla LCD relacionados con el control de acceso.

Asimismo, enciende un LED que indica acceso permitido o activa un zumbador cuando el acceso es denegado.

- **Publicación de datos – void Publicar\_Datos():** Se publican los datos de temperatura, humedad relativa, luminosidad, humedad de suelo y nivel de agua cada 15 segundos en el broker MQTT de Adafruit, para ser visualizados de forma remota.
- **Modo automático – void Automatico():** Se llevan a cabo las acciones de control (actuaciones) de forma automática, siendo estas: el riego, el llenado del tanque y la ventilación. La lógica de control se basa en umbrales preestablecidos para cada una de las variables medidas en el sistema.
- **Modo manual – void Manual():** Se controlan los actuadores del sistema de forma manual, según la elección del usuario mediante los botones de riego, llenado y ventilación creados en la plataforma MQTT de Adafruit.
- **Configuración WIFI – WIFI\_config():** Esta función permite a la ESP32 conectarse a una red WIFI previamente registrada. Si no hay una red registrada o la red que fue registrada no está disponible, crea un Access Point (Punto de Acceso) y la ESP32 sirve una página web desde donde se pueden visualizar las redes disponibles y conectarse a alguna de ellas introduciendo sus credenciales. Dichas credenciales WIFI son almacenadas en la memoria EEPROM del microcontrolador.
- **Configuración de actualizaciones OTA – void OTA\_config():** OTA es la sigla de "Over-The-Air" (sobre el aire, en español). Se refiere a la capacidad de actualizar el firmware o software de un dispositivo electrónico de manera inalámbrica a través de una conexión a internet por medio de una red de área local (LAN, por sus siglas en inglés). En esta función, se configura la programación de la placa vía OTA, por medio de un servidor web local al que se puede acceder mediante la dirección IP de la ESP32. Para ingresar a la página web de actualizaciones OTA, primero se debe iniciar sesión

ingresando el nombre de usuario y la contraseña establecidos en la programación.

- **Conexión MQTT – void MQTT\_connect():** Función para conectarse y reconectarse, según sea necesario, al servidor de Adafruit.io a través del protocolo MQTT. Cuando se pierda la conexión, esta ejecutará hasta cinco intentos de reconexión que se harán cada 10 segundos. Se llamará en la función principal y controlará la conexión al broker MQTT.

### 5.2.1. COMUNICACIÓN MQTT CON SERVIDOR

Elegir el broker más adecuado para este proyecto **condiciona el buen funcionamiento y éxito de este**. A continuación, se presenta un repaso rápido de algunos de los principales MQTT disponibles.

- **Mosquitto.** Es el broker MQTT más conocido en el sector doméstico/#maker. Es un broker Open Source desarrollado por la fundación Eclipse y distribuido bajo licencia EPL/EDL. Está programado en C, y es multiplataforma. Es un broker liviano y adecuado para uso en servidores de baja potencia.
- **Mosca.** Es un broker MQTT Open Source para Node.js, desarrollado en Javascript por Matteo Collina. Puede ser empleado como aplicación independiente o embebido en cualquier proyecto de Node.js
- **Aedes.** Del mismo autor que Mosca, Aedes es un servidor broker MQTT Open Source para Node.js diseñado para ser un reemplazo de Mosca.
- **HBMQTT.** Es un broker MQTT Open Source escrito en Python que funciona sobre asyncio, introducido en Python 3.4.
- **EMQTT.** Erlan MQTT broker es Open Source, desarrollado en Erlang/OTP, está diseñado para aplicaciones con grandes exigencias en escalabilidad.
- **RabbitMQ.** Es un popular broker de mensajería AMQP Open Source, que también permite emplear el protocolo MQTT a través de un Adaptador.

- **HiveMQ CE.** La versión Community del popular HiveMQ es un broker Open Source basado en Java.
- **ActiveMQ.** Es un broker de mensajería JMS (Java Message Script) Open Source desarrollado por Apache, que también admite protocolo MQTT.
- **Moquette.** Un broker MQTT Open Source escrito en Java desarrollado por Eclipse, que destaca por su livieza.
- **MQTTnet.** Un broker Open-Source para .NET.

Son sólo algunos ejemplos de los broker más conocidos, pero, por supuesto, hay muchos más (Mystique, Awesome MQTT, SurgeMQ, VerneMQ), existe todo un ecosistema, con un broker para cada lenguaje y cada proyecto. [69]

### **Plataforma Adafruit IO para dispositivos IoT**

Existen numerosas plataformas que le permiten conectar un sistema IoT completo y al mismo tiempo recopilar, almacenar y analizar datos en tiempo real de los dispositivos conectados. En esta tesis, se empleó la plataforma Adafruit IO, desarrollada y mantenida por la reconocida empresa Adafruit Industries, entre varias de estas plataformas.

Adafruit IO es un servicio de mensajería que utiliza el protocolo MQTT con un modelo de publicador/suscriptor y un intermediario que acepta mensajes publicados por el cliente y los distribuye a los clientes suscritos.

FIRST NAME

LAST NAME

EMAIL

USERNAME

Username is viewable to the public on the forums, Adafruit IO, and elsewhere.

PASSWORD

HAVE AN ADAFRUIT ACCOUNT?

**Figura 47. Plataforma Adafruit. Elaboración propia.**

La plataforma sobre la que se está diseñando este proyecto se llama Adafruit; Es fácil de usar y práctico, visualmente más atractivo que los que ofrecen otros proveedores de servicios y totalmente compatible con el ESP32. Es importante señalar que, a diferencia de otros servidores gratuitos basados en la nube, su paquete gratuito se encuentra entre los más completos. Sin embargo, está severamente limitado en términos de número de tablas (5 como máximo) y feeds (10 como máximo), publicación de datos en tiempo real (30 datos por minuto) y almacenamiento de datos durante un máximo de 30 días. Debido a estas limitaciones, se establecen las prioridades del proyecto sobre los temas instalados (feeds), los tiempos de publicación y suscripción de datos, entre otros factores.

Se optó por utilizar este servidor para el desarrollo del sistema porque ofrece más comodidades que las otras opciones disponibles. Uno de los principales beneficios es que era compatible con la placa ESP32, y además tiene un paquete gratuito que brinda todas las funciones requeridas para realizar el monitoreo de todas las variables. Sin embargo, si bien es cierto que la versión gratuita del servidor sólo brinda acceso a cinco gráficos de datos, y un máximo de 10 feeds sigue ofreciendo muchas más funciones que el resto de las opciones.

Con esta plataforma se podrá agregar, almacenar y visualizar datos en tiempo real en la nube desde dispositivos IoT conectados. También permitirá interactuar con los dispositivos desde paneles de control (dashboards) basados en la web. Es compatible con dispositivos como **Arduino**, **Raspberry Pi** y **ESP8266/32** entre otros.



Figura 48. Paquete de uso gratuito del Servidor Adafruit IO. Elaboración propia.

## CONFIGURACIÓN DE CREDENCIALES WIFI

Usualmente, cuando se requiere conectar la ESP32 a una red WIFI, se utiliza la popular biblioteca WiFi. Sin embargo, al usar esta, es necesario establecer las credenciales de la red (SSID y contraseña) directamente en el código. Esto, además de ser poco práctico, podría ocasionar problemas de seguridad.

```
#include <WiFi.h>

const char* ssid    = "your-ssid"; // Change this to your WiFi SSID
const char* password = "your-password"; // Change this to your WiFi password
```

Figura 49. Configuración tradicional de las credenciales WIFI. Elaboración propia.

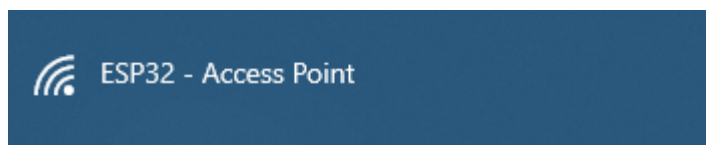
Como se puede notar, empleando la librería WiFi, el usuario del sistema sólo podría cambiar las credenciales de la red modificando el código y volviendo a programar la ESP32. Asimismo, al dejar expuestas las credenciales en el código, la red se volvería fácilmente vulnerable.

Por tales razones, en el diseño del sistema se optó por hacer uso de una biblioteca más robusta, llamada WiFiManager. Esta biblioteca, desarrollada por

tzapu, permite configurar las credenciales de la red WIFI mediante una página web servida por la ESP32.

A continuación, se describe el funcionamiento de la biblioteca en cuestión, así como los pasos a seguir para conectar la ESP32 a una red WIFI.

Cuando la ESP32 empieza a ejecutar el código, luego de haber sido energizada o reiniciada, WiFiManager verifica si hay credenciales WIFI previamente almacenadas en la memoria EEPROM de la ESP32. Si las hay, entonces establece conexión con la red especificada. Si no las hay o si las credenciales almacenadas no corresponden con ninguna red WIFI disponible, entonces la ESP32 se configura como un Access Point (Punto de acceso), al cual el usuario se puede conectar para establecer nuevas credenciales. El nombre del Access Point fue establecido como “ESP32 – Access Point”, con la contraseña “wificonfig”.



**Figura 50. Nombre visible del Access Point. Elaboración propia.**

Una vez establecida la conexión, empleando un navegador web, el usuario deberá acceder a la dirección IP 192.168.4.1, donde la ESP32 sirve la página web que permite configurar las credenciales WIFI. Cabe destacar que esta conexión ocurre a nivel de área local, por lo que no se requiere estar conectado a Internet.

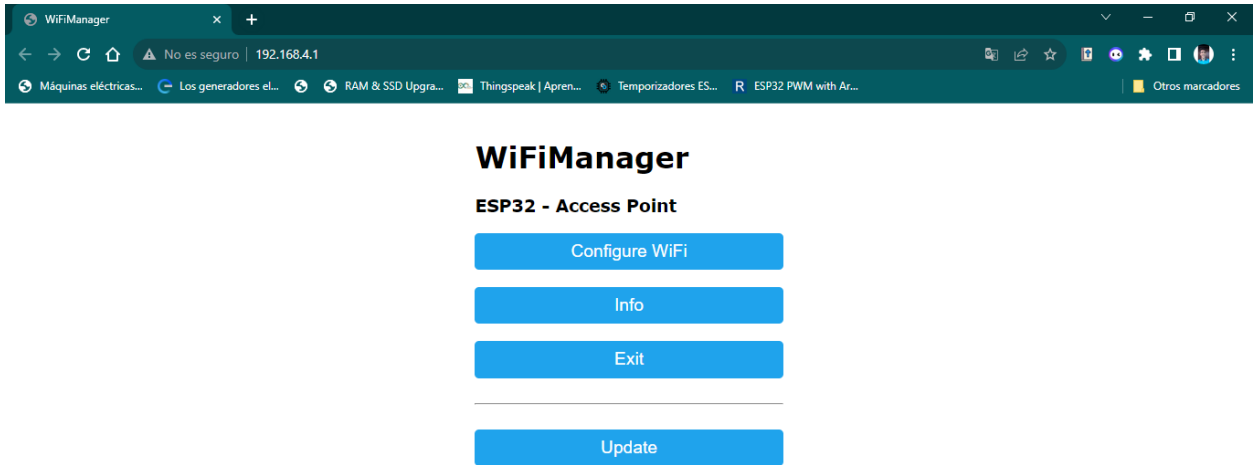


Figura 51. Página Web de WiFiManager – Menú principal. Elaboración propia.

Para cambiar las credenciales WIFI, se debe hacer click sobre el botón “Configure WiFi”. Una vez hecho esto, se mostrarán las redes WIFI disponibles y el usuario podrá establecer el SSID y la contraseña de la red a la que desea conectarse. Cuando las credenciales hayan sido establecidas, se debe hacer click sobre el botón “Save” para almacenarlas en la memoria EEPROM de la ESP32.

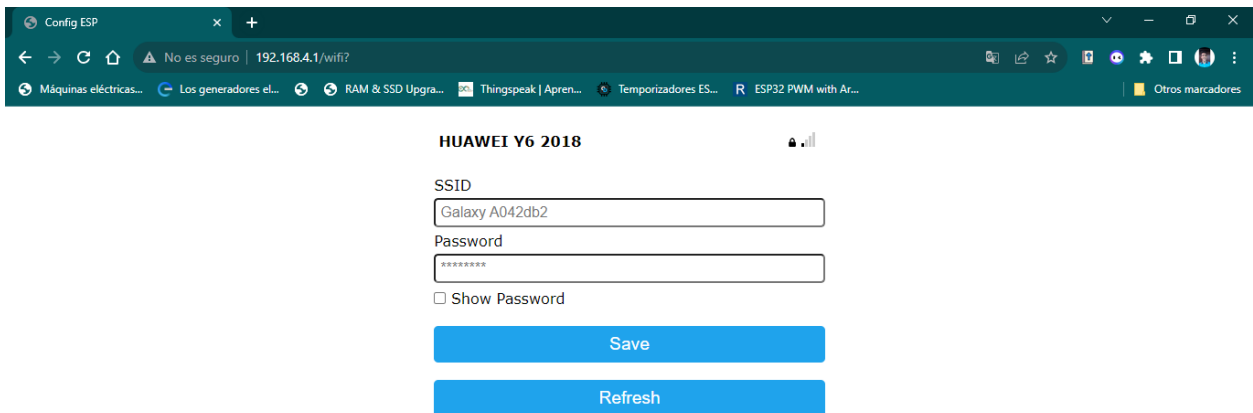


Figura 52. Página Web de WiFiManager – Configuración de credenciales WIFI. Elaboración propia.

Si las credenciales WIFI fueron introducidas correctamente y la ESP32 logra conectarse a la red WIFI, esta pasará de modo “Access Point” a “WIFI Client” (Cliente WIFI), para finalmente establecer comunicación con el broker MQTT de

Adafruit. Si no, la ESP32 seguirá funcionando en modo “Access Point” hasta que logre conectarse a una red WIFI.

## **CONFIGURACIÓN DE ACTUALIZACIONES OTA**

Para actualizar el firmware de la ESP32 mediante la tecnología OTA, se puede emplear directamente el Arduino IDE. No obstante, esta práctica resulta muy incómoda y complicada para el usuario final, puesto que se debe configurar el Arduino IDE de una forma específica para que permita la programación vía OTA.

Por ello, se eligió una opción más simple y práctica que consiste en realizar las actualizaciones OTA desde una página web servida por la ESP32. A continuación, se describe el procedimiento para llevar a cabo las actualizaciones OTA mediante el servidor web, empleando la biblioteca Update.

En primera instancia, se debe verificar que la ESP32 y el dispositivo desde donde se accederá al servidor web se encuentren conectados a la misma red WIFI. Posteriormente, se requiere conocer la dirección IP asignada por el router a la ESP32. Una vez hecho esto, se debe introducir dicha dirección IP en un navegador web.

Para acceder al servidor web, primeramente, por seguridad, se debe iniciar sesión, ingresando un nombre de usuario y una contraseña. En este caso, para efectos de prueba, se estableció en la programación la palabra “OTAadmin” como nombre de usuario y “OTAUpdates14” como contraseña.

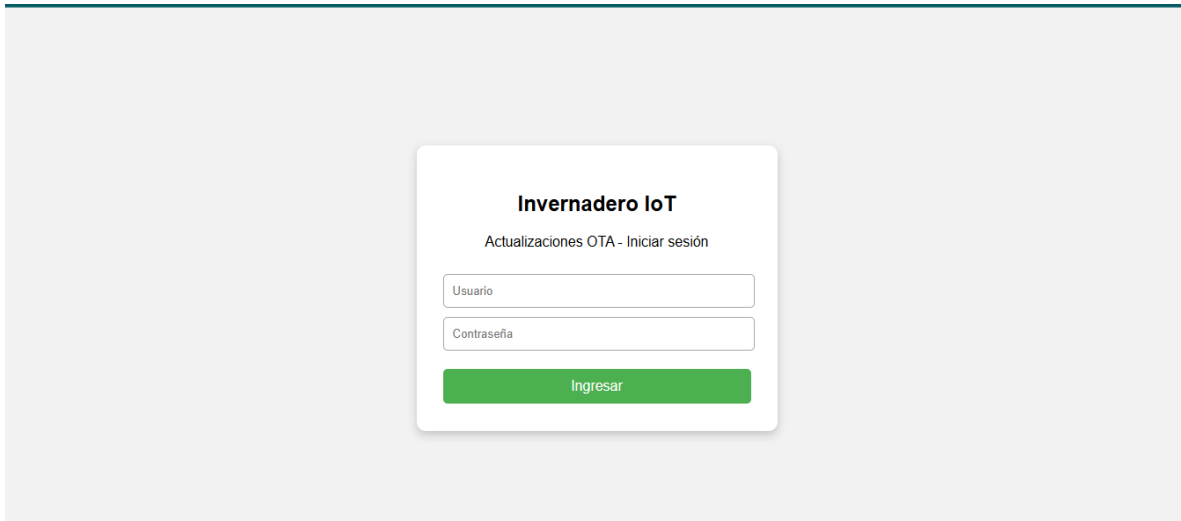


Figura 53. Página Web de actualizaciones OTA – Inicio de sesión. Elaboración propia.

Si el usuario y la contraseña son correctos, se abrirá una nueva pestaña en el navegador que permite acceder al servidor web de actualizaciones OTA. En cambio, si son incorrectos, sólo aparecerá un mensaje indicando un error en el nombre de usuario o en la contraseña.

Una vez en el servidor web, haciendo click sobre el botón “Seleccionar archivo”, se podrá cargar el archivo que contiene el código (previamente compilado) que se desea subir a la placa ESP32.

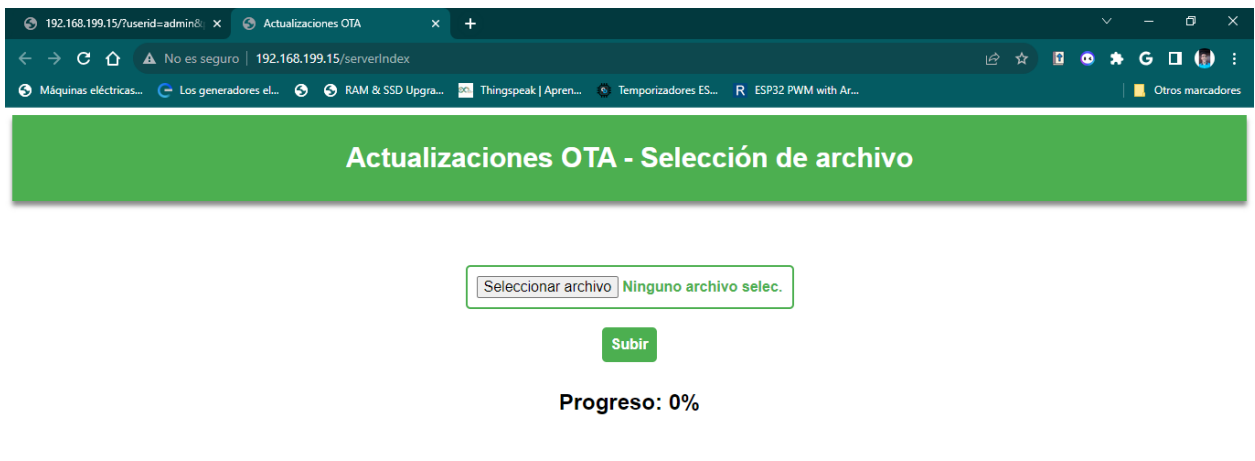


Figura 54. Página Web de actualizaciones OTA – Selección de archivo. Elaboración propia.

Finalmente, se debe hacer click en botón “Subir” para iniciar el proceso actualización de firmware. Este proceso termina cuando el progreso es equivalente al 100 %.



**Figura 55. Página Web de actualizaciones OTA – actualizando firmware de la ESP32. Elaboración propia.**

### **5.3. Interfaz web de Adafruit y app MQTT Dash**

A continuación, se presentan los dashboards diseñados en las plataformas Adafruit y MQTT Dash, que permiten monitorear y controlar el sistema IoT de forma remota desde dispositivos móviles, tablets y computadoras. Gracias al protocolo MQTT, se logra una comunicación eficiente y en tiempo real entre el hardware y las plataformas de monitoreo, garantizando la transferencia segura y rápida de datos.

Estas interfaces gráficas ofrecen una visualización clara de las variables y estados del prototipo, facilitando la supervisión continua sin necesidad de presencia física. Además, permiten la interacción directa con el sistema para realizar ajustes y comandos que optimizan su funcionamiento según las condiciones observadas.

La implementación de esta solución IoT demuestra la efectiva integración entre el sistema físico y las plataformas digitales, destacando la importancia del protocolo MQTT para la gestión remota y en tiempo real.

Para el monitoreo y control remoto del sistema, se utilizó la plataforma Adafruit IO, un servicio en la nube diseñado para proyectos IoT que permite la visualización y gestión de datos de manera sencilla y eficiente. Adafruit IO ofrece una interfaz intuitiva y personalizable, con dashboards que facilitan la integración de sensores y actuadores para el control en tiempo real.

Esta plataforma destaca por su facilidad de uso, estabilidad y compatibilidad con múltiples dispositivos y protocolos, incluyendo MQTT, lo que garantiza una comunicación fluida y segura entre el hardware y la nube. Gracias a estas características, Adafruit IO se convierte en una herramienta ideal para la supervisión y administración remota de sistemas IoT.

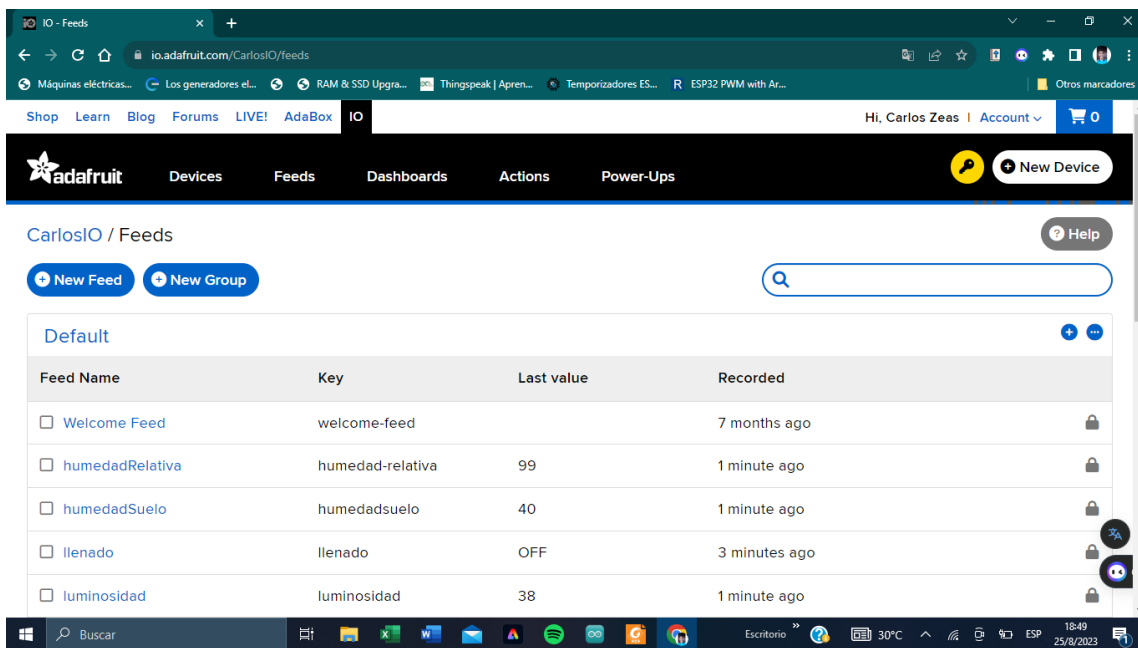


Figura 56. Plataforma MQTT de Adafruit – Feeds. Elaboración propia.

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> Welcome Feed	welcome-feed		7 months ago
<input type="checkbox"/> humedadRelativa	humedad-relativa	99	1 minute ago
<input type="checkbox"/> humedadSuelo	humedadsuelo	41	1 minute ago
<input type="checkbox"/> llenado	llenado	OFF	3 minutes ago
<input type="checkbox"/> luminosidad	luminosidad	54	1 minute ago
<input type="checkbox"/> modo	modo	ON	4 minutes ago
<input type="checkbox"/> nivelAgua	nivelagua	86	1 minute ago
<input type="checkbox"/> riego	riego	OFF	3 minutes ago
<input type="checkbox"/> temperatura	temperatura	26	1 minute ago
<input type="checkbox"/> ventilacion	ventilacion	ON	2 minutes ago

Figura 57. Plataforma MQTT de Adafruit – Feeds. Elaboración propia.

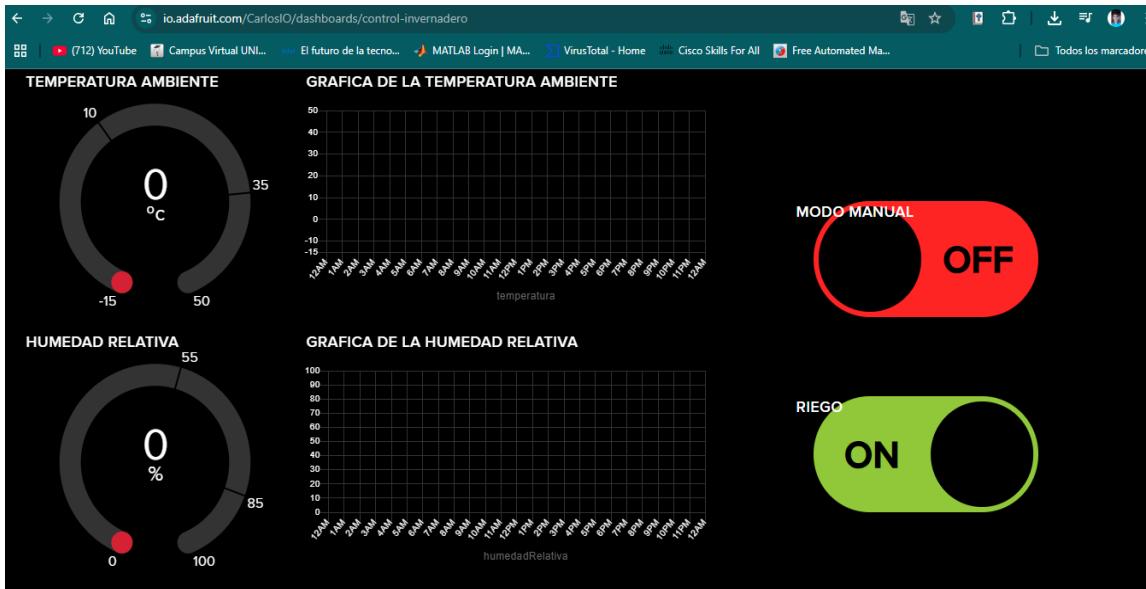


Figura 58. Dashboard construido en interfaz web de Adafruit. Elaboración propia.



Figura 59. Dashboard construido en interfaz web de Adafruit. Elaboración propia.

Por otro lado, la aplicación utilizada para este sistema es MQTT Dash (IoT, Smart Home), una app gratuita para Android disponible en la Play Store. Aunque existen otras opciones similares como IoT MQTT Panel, IoT MQTT Dashboard o Linear MQTT Dashboard, se optó por MQTT Dash debido a sus características destacadas, como su alta usabilidad, facilidad de manejo y buen rendimiento.

Si bien la interfaz de usuario puede parecer poco estética en un primer vistazo, la aplicación ofrece diversas opciones de personalización en diseño, incluyendo formas, colores y widgets, que permiten adaptar el dashboard a las necesidades del usuario de manera eficiente.

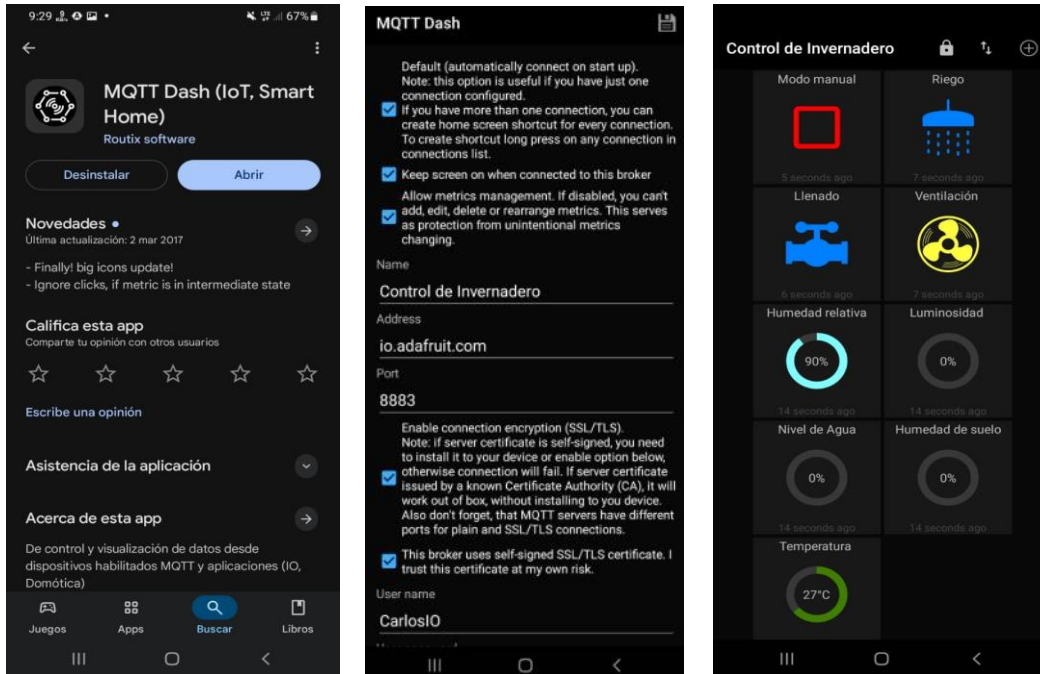
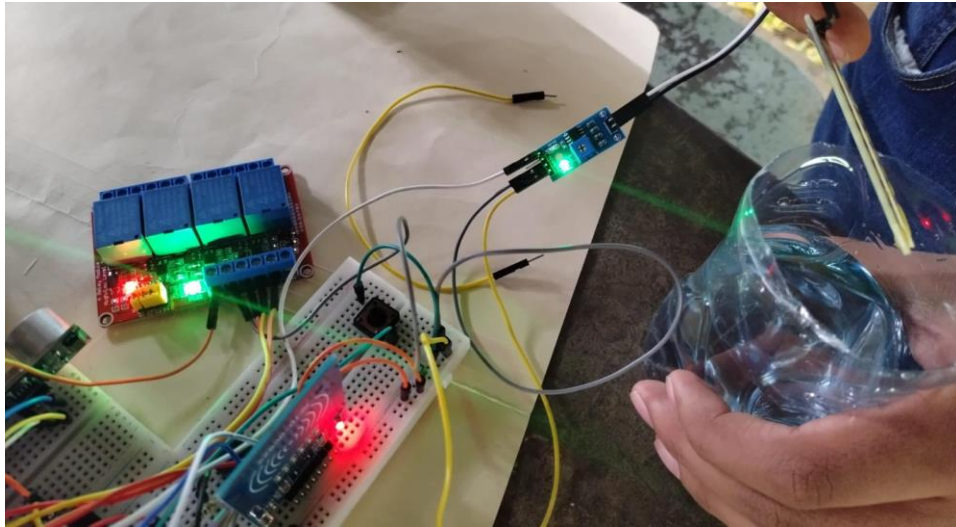


Figura 60. App MQTT Dash. Elaboración propia.

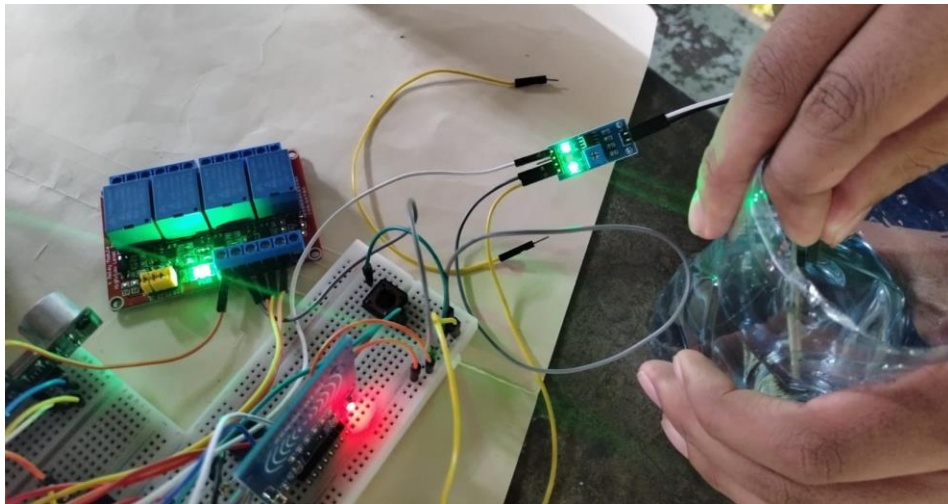
#### 5.4. Pruebas de funcionamiento

A continuación, se presentan las pruebas de funcionamiento realizadas sobre la plataforma de protoboard, donde se observa el sistema operando bajo condiciones reales de prueba. Durante estas pruebas, se verificó el correcto desempeño de cada módulo conforme a las especificaciones establecidas en el diseño y la programación.

Los resultados obtenidos fueron satisfactorios, evidenciando la adecuada interacción entre los componentes y la respuesta esperada del sistema frente a las distintas condiciones de operación planteadas. Estas evidencias permiten confirmar que el prototipo cumple con los objetivos funcionales propuestos, garantizando la viabilidad y efectividad del diseño implementado.



**Figura 61.** Pruebas de funcionamiento del prototipo construido. Elaboración propia.



**Figura 62.** Pruebas de funcionamiento del prototipo construido. Elaboración propia.

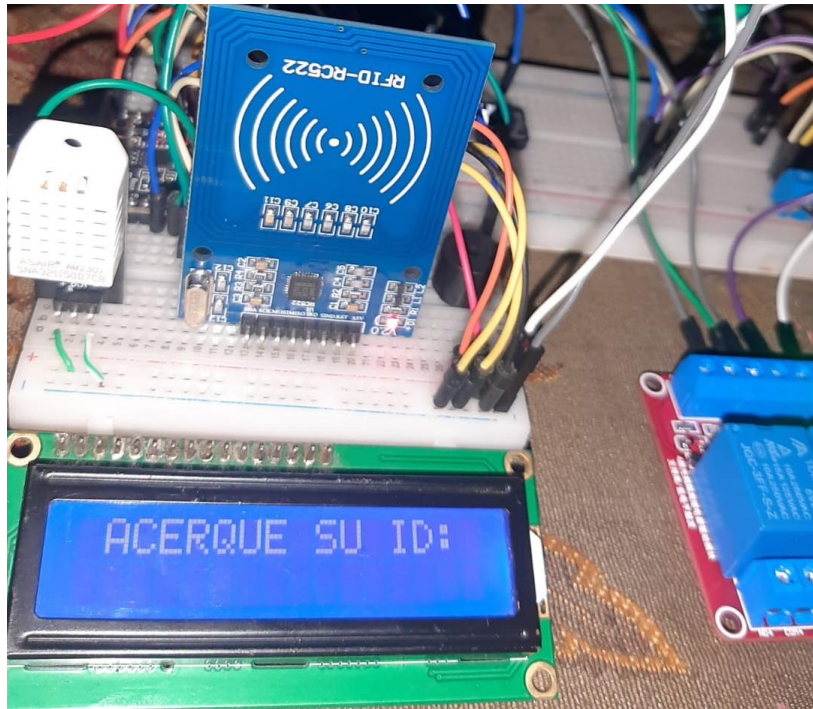


Figura 63. Pruebas de funcionamiento del prototipo construido. Elaboración propia.



Figura 64. Pruebas de funcionamiento prototipo construido. Elaboración propia.

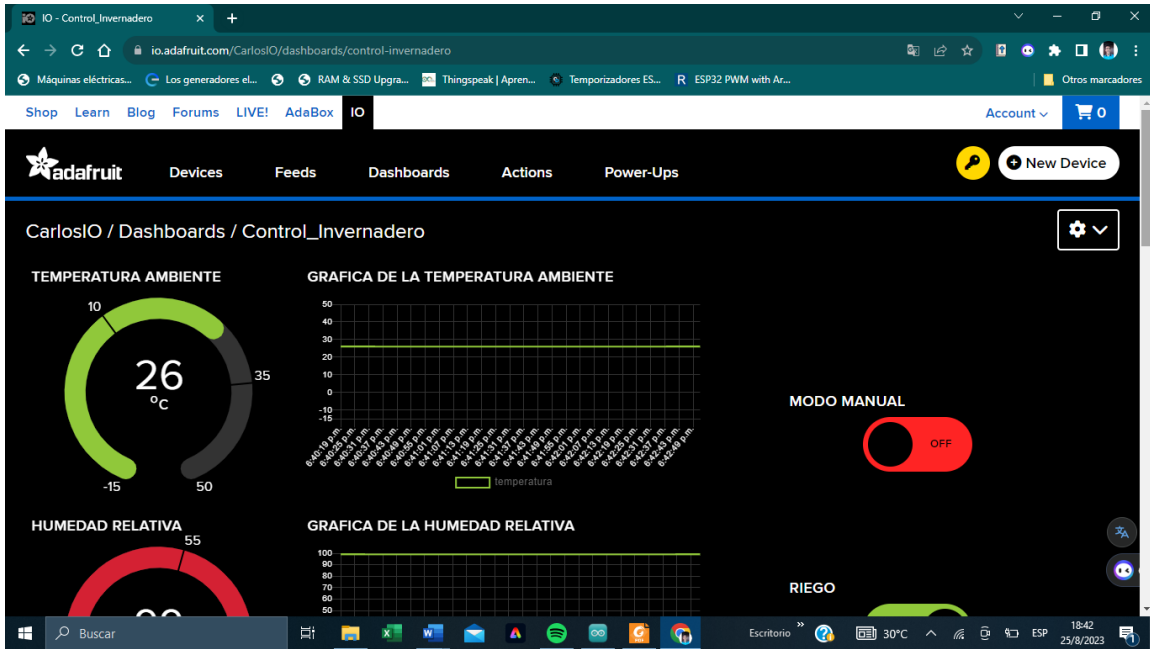


Figura 65. Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.

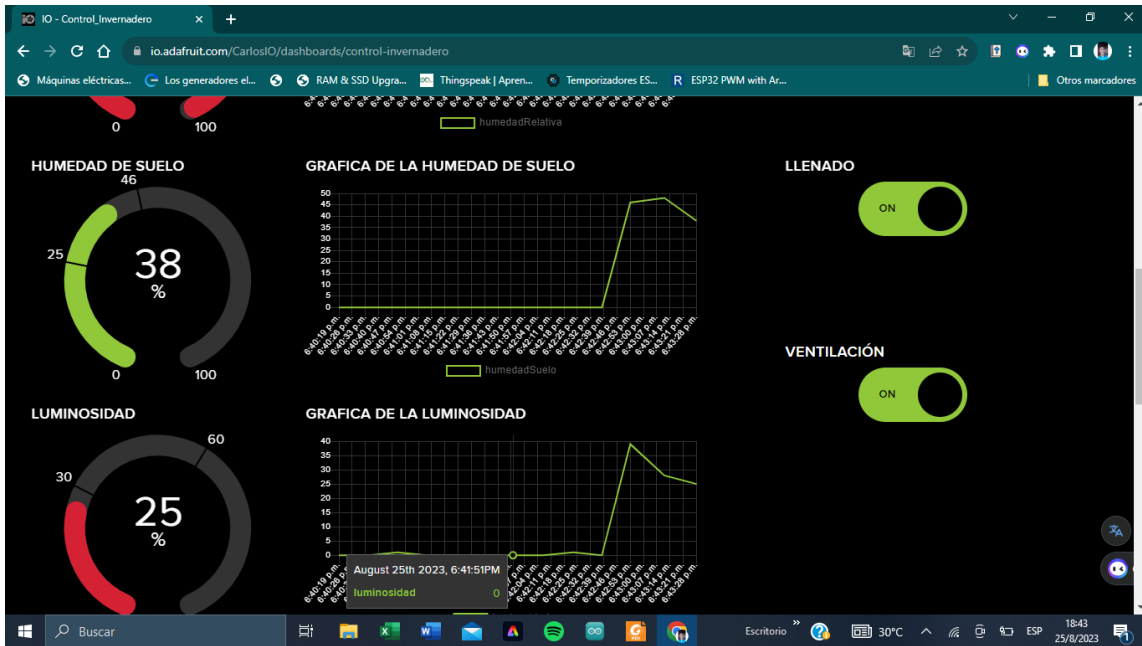


Figura 66. Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.

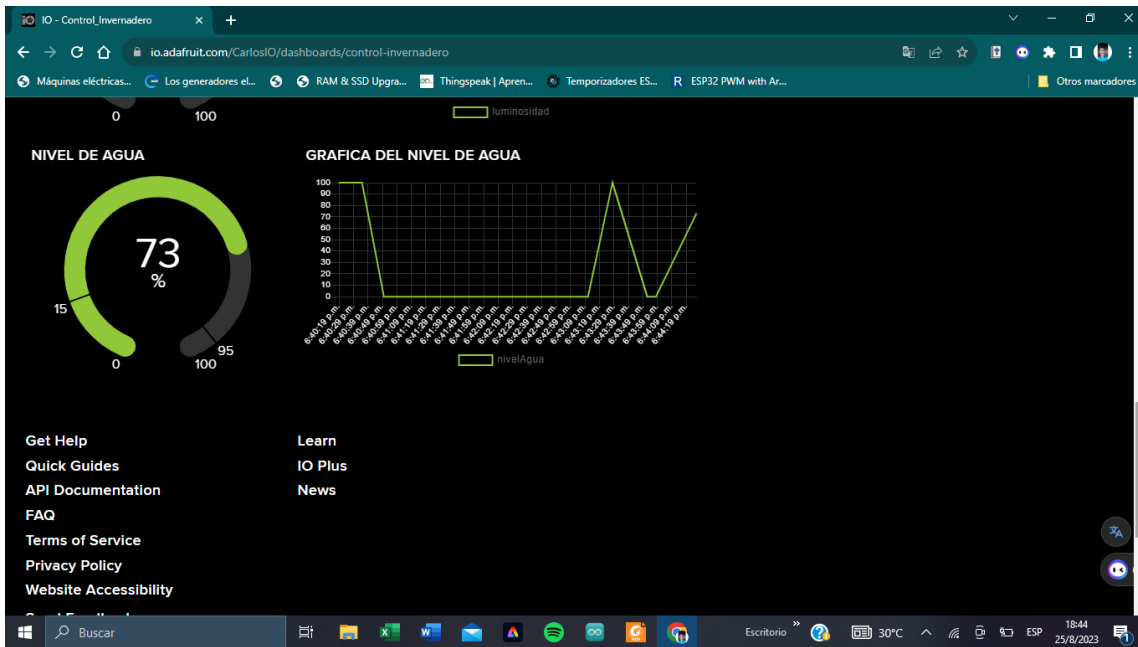


Figura 67. Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.

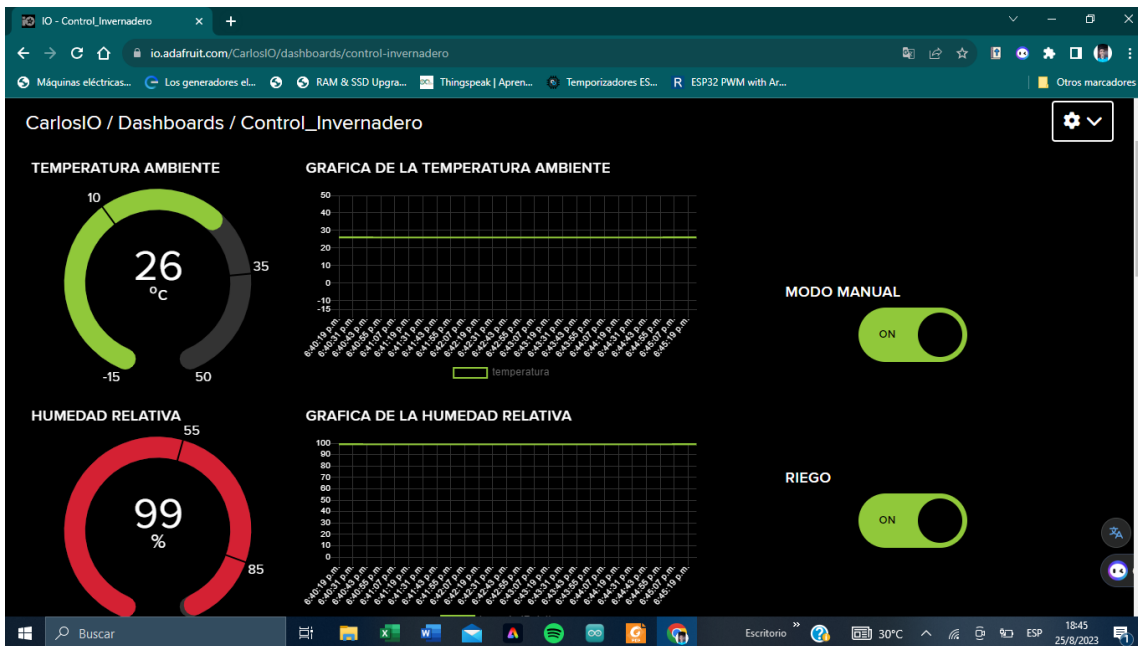


Figura 68. Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.

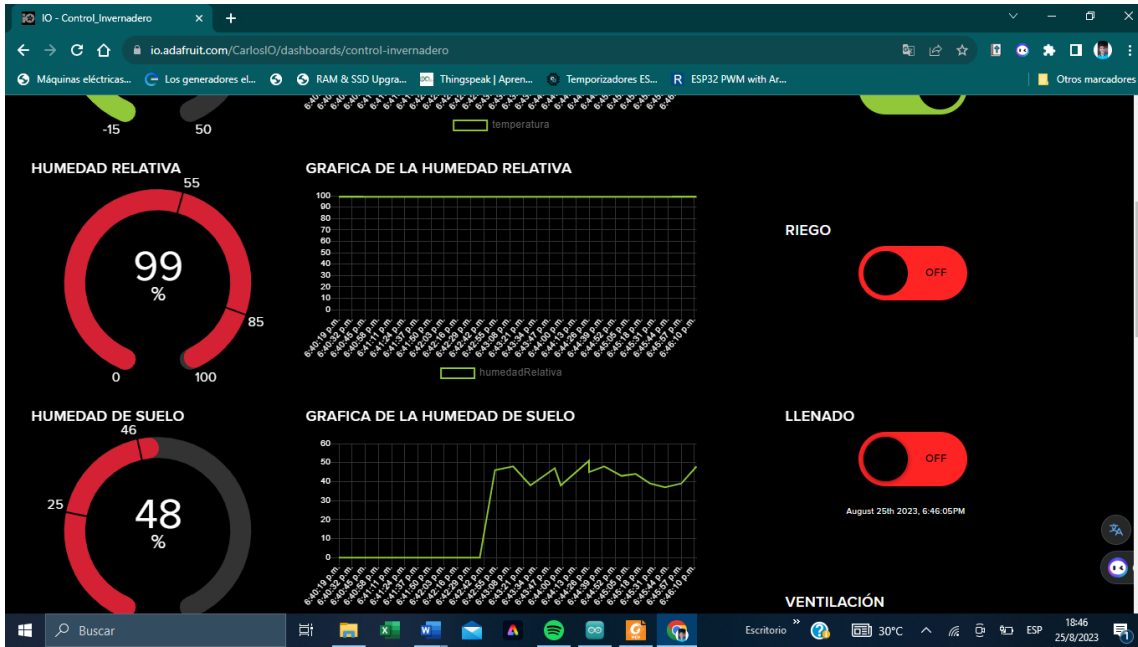


Figura 69. Pruebas de funcionamiento de interfaz web de Adafruit. Elaboración propia.



Figura 70. Pruebas de funcionamiento de app MQTT Dash. Elaboración propia.

## V. CONCLUSIONES

El desarrollo de este proyecto culminó en la creación de un prototipo funcional de sistema electrónico automatizado para invernadero, validando el cumplimiento de todos los objetivos planteados. Este prototipo representa una contribución significativa a la aplicación de soluciones tecnológicas en el sector agrícola, demostrando que es posible optimizar la producción de hortalizas mediante la automatización.

La flexibilidad del sistema, garantizada por su configuración a través del código, permite adaptar parámetros cruciales como los umbrales de las variables ambientales, las credenciales de conexión y la configuración del servidor broker. Esta característica es esencial para su aplicabilidad en diferentes escenarios con condiciones variables.

La implementación de este sistema podría facilitar la producción agrícola a lo largo del año, asegurando la calidad de los cultivos al protegerlos de factores climáticos adversos. Esta solución es particularmente relevante para el contexto agrícola nicaragüense, que enfrenta desafíos inherentes al cambio climático.

Finalmente, la integración del Internet de las Cosas (IoT) en este proyecto demuestra que la adopción de tecnologías accesibles y económicamente viables es una oportunidad significativa para aumentar la eficiencia y la sostenibilidad de la producción. A pesar de los costos iniciales de la infraestructura, la automatización del control climático, el riego y la gestión remota contribuyen a una operación más eficiente y rentable a largo plazo.

## VI. RECOMENDACIONES

Las siguientes recomendaciones se derivan de los hallazgos de este estudio y tienen como objetivo optimizar y expandir la viabilidad del prototipo a futuro.

- **Implementación de un Servidor MQTT Local:** Se propone la configuración de un servidor MQTT propio para mejorar la seguridad de los datos, reducir la latencia en la comunicación y eliminar la dependencia de servicios de terceros. Esto proporcionaría una solución más robusta y controlada para el sistema.
- **Integración de Sensores Adicionales:** Se recomienda la incorporación de sensores de dióxido de carbono (CO<sub>2</sub>) y velocidad del viento para enriquecer el monitoreo ambiental. Esta adición permitiría al sistema tomar decisiones de control más complejas y eficientes.
- **Desarrollo de Algoritmos de Aprendizaje Automático:** La recopilación continua de datos habilita la implementación de modelos de *machine learning* para predecir las necesidades de los cultivos y optimizar proactivamente las condiciones del invernadero, maximizando la producción y la eficiencia.
- **Pruebas en Entorno de Producción:** Para validar la robustez y escalabilidad del prototipo, se sugiere su prueba en un invernadero comercial. Esto permitiría evaluar su rendimiento bajo condiciones operativas reales y obtener datos valiosos para futuras mejoras.
- **Análisis Económico:** Se recomienda realizar un estudio de costos y beneficios para la implementación del prototipo a gran escala. Dicho análisis es fundamental para demostrar la viabilidad comercial y el retorno de inversión para pequeños y medianos agricultores.
- **Riego Personalizado y Energía Renovable:** A largo plazo, se podría investigar la creación de un sistema de riego que se adapte a las necesidades específicas de cada cultivo y la integración de fuentes de energía renovable, como paneles solares, para lograr un sistema autosuficiente y sostenible.

## VII. BIBLIOGRAFÍA

- [1] CATIE Nicaragua, «Agricultura bajo techo reduce consumo de agua a la mitad,» 2020. [En línea]. Available: <https://www.catie.ac.cr/nicaragua/es/81-agricultura-bajo-techo-reduce-consumo-de-agua-a-la-mitad.html>.
- [2] J. Peralta, «Estructura de costos en cultivo de chiltoma bajo invernaderos y determinación de la rentabilidad financiera, en la Universidad Católica del Trópico Seco Estelí periodo 2015,» 2016. [En línea]. Available: <https://repositorio.unan.edu.ni/3218/1/17603.pdf>.
- [3] Unidad de Seguimiento y Evaluación IICA. , «Uso de túneles en la producción hortícola de Nicaragua a junio 2004,» [En línea]. Available: <http://www.renida.net.ni/renida/iica/f01-i59-121.pdf>.
- [4] A. Álvarez, « Tipos de Invernaderos ¿Qué es un invernadero? y Usos,» Hydro Environment - Inovación Agrícola En Un Click, 18 diciembre 2024. [En línea]. Available: <https://hydroenv.com.mx/id44/>.
- [5] I. Fertri, «¿Qué tipos de invernaderos existen? | Fertri Invernaderos,» Fertri Invernaderos, 31 agosto 2023. [En línea]. Available: <https://fertri.com/2023/08/tipos-de-invernaderos-existen/>.
- [6] Horticultivos , [En línea]. Available: <https://horticultivos.com/articulo/>.
- [7] R. Lacayo , L. López, Z. Zamora y E. Padilla, «El Higo - Revista Tecnológica,» 2021. [En línea]. Available: <https://camjol.info/index.php/elhigo/article/download/13012/15091>.
- [8] C. Andrade, «Control climático de un invernadero mediante lógica borrosa,» [En línea]. Available: [https://oa.upm.es/47865/1/TFG\\_CESAR\\_MAZQUIARAN\\_ANDRADE.pdf](https://oa.upm.es/47865/1/TFG_CESAR_MAZQUIARAN_ANDRADE.pdf).

- [9] L. W. Jett, «Watering and fertilizing tomatoes in a high tunnel | MU Extension,» 01 enero 2025. [En línea]. Available: <https://extension.missouri.edu/publications/g6462>.
- [10] I. Iberia, «Riego por goteo en Tomate - Irritec Iberia,» 30 abril 2024. [En línea]. Available: <https://irritec.es/cultivos/riego-por-goteo-de-tomate/>.
- [11] P. Aggarwal, «How to select electronic components?,» Pallav Aggarwal, 15 octubre 2022. [En línea]. Available: <https://pallavaggarwal.in/how-to-select-electronic-components/#overview>.
- [12] «¿Qué tarjeta de desarrollo elegir? (Parte 1),» Hacedores.com | Maker Community, [En línea]. Available: <https://hacedores.com/que-tarjeta-de-desarrollo-elegir-parte-1/>.
- [13] «What are the advantages and disadvantages of ESP32 compared to ESP8266?,» PCB HERO, 30 septiembre 2022. [En línea]. Available: <https://www.pcb-hero.com/blogs/lisas-column/what-are-the-advantages-and-disadvantages-of-esp32-compared-to-esp8266>.
- [14] «Configurando ESP32 con IDE de Arduino,» ICStation.com – RogerBit, [En línea]. Available: <https://rogerbit.com/wprb/2017/08/configurando-esp32-con-ide-de-arduino-subiendo-datos-a-thendspeak-con-el-aporte-de-icst>.
- [15] R. R. H. I. A. T. K. R. B. S. K. S. S. F. M. S. M. M. B. & S. S. Shamshiri, «Greenhouse Automation Using Wireless Sensors and IoT Instruments Integrated with Artificial Intelligence.,» In IntechOp eBooks, 2021. [En línea]. Available: <https://doi.org/10.5772/intechopen.97714>.
- [16] J. Joseph, «Tutorial sensor de temperatura y humedad DHT11 y DHT22,» CircuitDigest, 17 mayo 2022. [En línea]. Available: <https://circuitdigest.com/microcontroller-projects/interfacing-dht11-sensor-with-arduino>.

- [17] «Sensor de Humedad del Suelo FC-28,» Naylamp Mechatronics - Perú, [En línea]. Available: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/47-sensor-de-humedad-de-suelo-fc-28.html>.
- [18] E. Notes, «Light Dependent Resistor LDR, Photo-Resistor » Electronics Notes,» Electronics Notes, [En línea]. Available: [https://www.electronic-notes.com/articles/electronic\\_components/resistors/light-dependent-resistor-ldr.php](https://www.electronic-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php).
- [19] «Ultrasonic Sensor HC-SR04 and Arduino – complete guide,» How to Mechatronics, 18 febrero 2022. [En línea]. Available: <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>.
- [20] «HC-SR04 Ultrasonic Sensor Guide with Arduino Interfacing | Sensor,» © 2018 ElectronicWing, [En línea]. Available: <https://www.electronicwings.com/sensors-modules/ultrasonic-module-hc-sr04>.
- [21] «Linear actuators you can trust,» REAC, [En línea]. Available: <https://reac-group.com/actuators/>.
- [22] «Control a water pump by the ESP32 card,» [En línea]. Available: <https://www.robotique.tech/robotics/control-a-water-pump-by-the-esp32-card/>.
- [23] «Tutorial LCD con I2C, controla un LCD con solo dos pines,» Naylamp Mechatronics, [En línea]. Available: [https://naylampmechatronics.com/blog/35\\_tutorial-lcd-con-i2c-controla-un-lcd-con-solo-dos-pines.html](https://naylampmechatronics.com/blog/35_tutorial-lcd-con-i2c-controla-un-lcd-con-solo-dos-pines.html).
- [24] «Control a water pump by Arduino,» Robotique, 22 marzo 2024. [En línea]. Available: <https://www.robotique.tech/robotics/control-a-water-pump-by-arduino/#:~:text=a%20water%20pump%20controlled%20by,and%20direction%20of%20the%20water>}.

- [25] «5V Single-Channel Relay Module,» Components101, [En línea]. Available: <https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet>.
- [26] LME Editorial Staff, «What is RFID? How It Works? Interface RC522 RFID Module with Arduino,» Last Minute Engineers, 07 abril 2025. [En línea]. Available: <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>.
- [27] C. Rodríguez, «¿Qué es un protocolo de comunicación y cuál es el más utilizado?,» SDI Industrial, 23 junio 2023. [En línea]. Available: <https://sdindustrial.com.mx/blog/protocolos-de-comunicacion-que-son/>.
- [28] K. Magdy, «ESP32 WiFi Tutorial & Library Examples (Arduino IDE),» DeepBlue, 17 agosto 2023. [En línea]. Available: <https://deepbluembedded.com/esp32-wifi-library-examples-tutorial-arduino/>.
- [29] «¿Qué es el MQTT? - Explicación del protocolo MQTT - AWS,» Amazon Web Services, Inc., [En línea]. Available: <https://aws.amazon.com/es/what-is/mqtt/>.
- [30] Jecrespom, «Protocolo MQTT,» Aprendiendo Arduino, 22 abril 2022. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2022/04/22/protocolo-mqtt/>.
- [31] S. Santos , «ESP32 Over-the-air (OTA) programming | Random nerd tutorials,» Random Nerd Tutorials, 12 junio 2020. [En línea]. Available: <https://randomnerdtutorials.com/esp32-over-the-air-ota-programming/>.
- [32] «How to Perform Over-the-Air (OTA) Updates Using the ESP32 Microcontroller and its ESP-IDF,» DigiKey, [En línea]. Available: <https://www.digikey.com/en/articles/how-to-perform-ota-updates-using-esp32-microcontroller-and-esp-idf>.
- [33] V. Mohanan, «How to Write Parallel Multitasking Applications for ESP32 using FreeRTOS & Arduino - CIRCUITSTATE,» CIRCUITSTATE Electronics, [En

línea]. Available: <https://www.circuitstate.com/tutorials/how-to-write-parallel-multitasking-applications-for-esp32-usin>.

- [34] «ESP32 Dual Core with FreeRTOS and Arduino IDE,» Microcontrollers Lab, [En línea]. Available: <https://microcontrollerslab.com/esp32-dual-core-freertos-arduino-ide/>.
- [35] E. Sandoval, «¿Qué es una variable en programación?,» Ebac, [En línea]. Available: <https://ebac.mx/blog/variable-en-programacion>.
- [36] E. Sandoval, «¿Qué es un objeto en programación?,» Ebac, [En línea]. Available: <https://ebac.mx/blog/objeto-en-programacion>.
- [37] «Task Programming,» ScienceDirect, [En línea]. Available: <https://www.sciencedirect.com/topics/computer-science/task-programming>.
- [38] G. Alemán y K. Sánchez, «Implementación de una red de sensores inalámbricos (WSN) para el cultivo de rábano utilizando protocolo ZigBee ubicado en invernadero de la UNI,» Universidad Nacional de Ingeniería, 2021.
- [39] M. Flores y J. Rodríguez, «Diseño e implementación de un sistema de riego por goteo automatizado para el cultivo del café usando tecnología de microcontrolador y estableciendo un sistema de control inalámbrico en la comunidad de Totogalpa,» Universidad Nacional de Ingeniería, 2018.
- [40] K. Ortega, «Riego por goteo convencional y automatizado en la producción de tomate (*Solanum lycopersicum* L.), a campo abierto, UNA, Managua, 2020-2021,» Universidad Nacional Agraria, 2021.
- [41] W. Corrales y A. Morazán, «Desarrollo de un sistema de información para el control de válvulas y estadísticas de riego en el sector agrícola,» Universidad Nacional de Ingeniería., 2017.

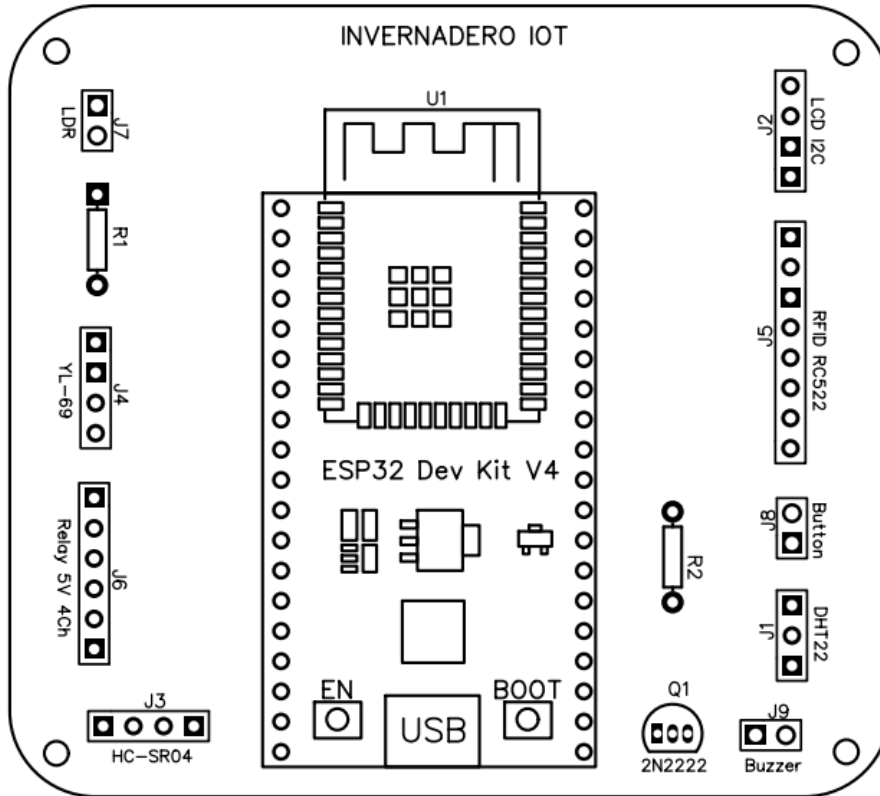
- [42] M. Hernández , A. Parrila y A. Triminio, « Prototipo de un sistema de riego y fertilización mediante sensores de humedad para la finca “La Gaviota”, ASODECOM, Somoto, Madriz,» Universidad Nacional de Ingeniería, 2020.
- [43] C. Crespo, «Factores ambientales y el potencial de rendimiento de hortalizas - PortalFruticola.com.,» Portal Frutícola. , 15 Enero 2025. [En línea]. Available: <https://www.portalfruticola.com/noticias/2023/01/06/factores-ambientales-y-el-potencial-de-rendimiento-de-hortalizas/>.
- [44] «Sistema de riego,» Ferrovial, 19 octubre 2023. [En línea]. Available: <https://www.ferrovial.com/es/recursos/sistema-de-riego/#:~:text=%C2%BFQu%C3%A9%20es%20un%20sistema%20de,tubos%2C%20bombas%20hidr%C3%A1ulicas%20y%20aspersores.>
- [45] Jecrespom, «Electrónica, Sensores, Actuadores y Periféricos,» Aprendiendo Arduino, 06 noviembre 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/11/06/electronica-sensores-actuadores-y-perifericos/>.
- [46] F. Sebastiany, «Qué es un lenguaje de programación y sus aplicaciones,» Salesforce, 04 julio 2025. [En línea]. Available: <https://www.salesforce.com/mx/blog/lenguaje-de-programacion/>.
- [47] Jecrespom, «IDE Arduino,» Aprendiendo Arduino, 11 diciembre 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>.
- [48] «Plataforma del Internet de las Cosas,» www.cognizant.com, [En línea]. Available: <https://www.cognizant.com/es/es/glossary/internet-of-things-platform.>
- [49] L. Calvo, «¿Qué es una app? Guía sobre aplicaciones móviles,» GoDaddy Resources - Spain, 26 noviembre 2024. [En línea]. Available: <https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza.>

- [50] Comunicación Herogra, «El tomate, la hortaliza más conocida del mundo,» Herogra Especiales, 27 enero 2021. [En línea]. Available: <https://herograespeciales.com/el-tomate-la-hortaliza-mas-conocida-del-mundo/>.
- [51] «El cultivo del tomate (Parte I),» infoAgro.com, [En línea]. Available: [https://www.infoagro.com/documentos/el\\_cultivo\\_del\\_tomate\\_\\_parte\\_i\\_.asp](https://www.infoagro.com/documentos/el_cultivo_del_tomate__parte_i_.asp).
- [52] AdminTextil, «Invernaderos: ¿cómo funciona?,» Textil Agrícola, 05 julio 2023. [En línea]. Available: <https://www.textilagricola.com/invernaderos-como-funciona/#:~:text=Un%20invernadero%20est%C3%A1%20formado%20por,tipo%20y%20uso%20del%20invernadero..>
- [53] C. Vargas, «¿Por qué elegir un sistema de riego por goteo?,» Grupo Hidráulica, 01 marzo 2025. [En línea]. Available: <https://grupohidraulica.com/noticias/2024/12/19/por-que-elegir-un-sistema-de-riego-por-goteo/>.
- [54] S. Lucero, «Managing winter humidity in a greenhouse - Growing spaces greenhouses.,» Growing Spaces Greenhouses, 25 febrero 2025. [En línea]. Available: <https://growingspaces.com/blog/managing-winter-humidity-in-a-greenhouse/>.
- [55] V. Cherlinka, « Punto de marchitez permanente y temporal en las plantas,» EOS Data Analytics, 7 marzo 2025. [En línea]. Available: <https://eos.com/es/blog/punto-de-marchitez-permanente/>.
- [56] c. fbustos@yc, «La capacidad de campo de un suelo: técnicas para su medición - PortalFruticola.com,» Portal Frutícola, 10 febrero 2025. [En línea]. Available: <https://www.portalfruticola.com/noticias/2018/01/25/la-capacidad-de-campo-de-un-suelo-tecnicas-para-su-medicion/>.
- [57] «Balance Hídrico | lfv-aulavirtual,» Lfv-aulavirtual, [En línea]. Available: <https://sebapereyra.wixsite.com/lfv-aulavirtual/bce-hidrico>.

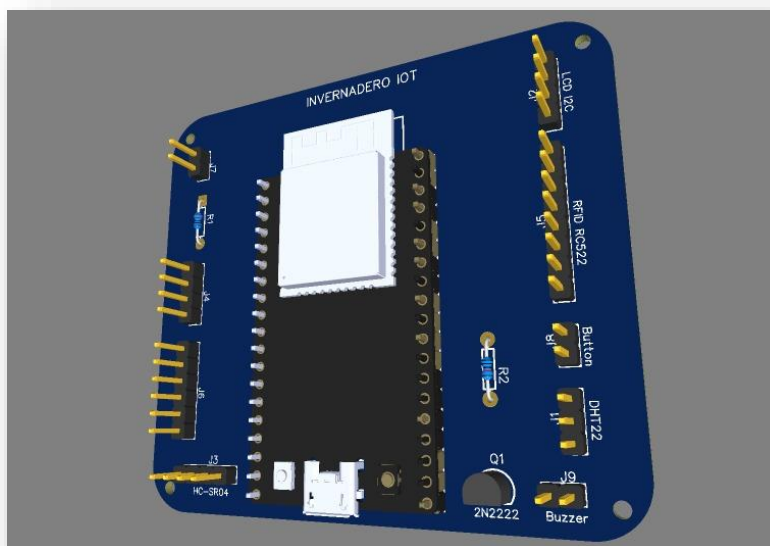
- [58] R. iAgua, « ¿Qué es el estrés hídrico?,» iAgua, 23 mayo 2024. [En línea]. Available: <https://www.iagua.es/respuestas/que-es-estres-hidrico>.
- [59] AdminRotoplas, « ¿Qué es el balance hídrico?,» Rotoplas Agroindustria., 22 noviembre 2024. [En línea]. Available: <https://rotoplas.com.ar/agroindustria/que-es-el-balance-hidrico/#:~:text=El%20balance%20h%C3%ADdrico%20en%20la,para%20su%20crecimiento%20y%20desarrollo..>
- [60] Vegetables Australia, 18 septiembre 2019. [En línea]. Available: <https://www.vegetables.bayer.com/au/en-au/resources/growing-tips/cultivation-insights/temperature--humidity--and-water-in-protected-culture-tomatoes.html>.
- [61] ManageEngine, «Protocolos de red | Tipos de protocolos de red,» ManageEngine OpManager, [En línea]. Available: <https://www.manageengine.com/latam/network-monitoring/protocolos-de-red.html>.

## VIII. ANEXOS

ANEXO 1. Placa PCB del Sistema en 2D:



ANEXO 2. Placa PCB del Sistema en 3D:



**ANEXO 3. Estimación de costos del sistema:**

ELEMENTOS	CANTIDAD	COSTO UNITARIO	MONTO
Sensor de temperatura y humedad relativa DHT22	1	C\$ 345.00	C\$ 345.00
ESP32-DevKitC V4	1	C\$ 480.00	C\$ 480.00
Sensor de humedad de suelo FC - 28	1	C\$ 108.00	C\$ 108.00
Fotorresistencia o LDR	1	C\$ 20.00	C\$ 20.00
Sensor ultrasónico HC-SR04	1	C\$ 148.00	C\$ 148.00
Módulo RFID RC522	1	C\$ 148.00	C\$ 148.00
Pantalla LCD 16x2 con I2C	1	C\$ 200.00	C\$ 200.00
Módulo relé 5V DC (4 Canales)	1	C\$ 350.00	C\$ 350.00
Electroválvulas	2	C\$ 296.00	C\$ 592.00
FAN 12V DC	1	C\$ 250.00	C\$ 250.00
Cerradura eléctrica 12V DC	1	C\$ 213.00	C\$ 213.00
Resistor 100 Ohms	1	C\$ 4.00	C\$ 4.00
Resistor 10K Ohms	1	C\$ 4.00	C\$ 4.00
Buzzer 5V DC	1	C\$ 30.00	C\$ 30.00
Transistor NPN 2N2222	1	C\$ 5.00	C\$ 5.00
Fuente de alimentación 12V DC, 3A	1	C\$ 350.00	C\$ 350.00
Fuente de alimentación 5V DC, 2A	1	C\$ 250.00	C\$ 250.00
Caja plexo cuadrada pvc legrand	2	C\$ 113.00	C\$ 226.00
Placa PCB	1	C\$ 300.00	C\$ 300.00
Estaño	1	C\$ 38.00	C\$ 38.00
Pasta para soldar	1	C\$ 50.00	C\$ 50.00
Cautín	1	C\$ 383.00	C\$ 383.00
Cable UTP Cat 5	4	C\$ 12.00	C\$ 48.00
Pulsador	1	C\$ 6.00	C\$ 6.00
<b>COSTO TOTAL</b>			<b>C\$ 4,548.00</b>