UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACSISTEL

INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

Prototipo IoT de monitoreo y control de variables ambientales para un cultivo hidropónico de hortalizas en sectores suburbanos de Ballenita

Víctor Richard Yépez Cevallos

Dirigido por:
Ing. Edisson Stalin Coral Salinas, Mgt.

LA LIBERTAD – ECUADOR

2026

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. Rovira Jurado Ronald Humberto, Ph. D.
DIRECTOR DE LA CARRERA

Ing. Edisson Stalin Coral Salinas, Mgtr.
TUTOR

Ing. Fernando Chamba Macas, Mgtr.
DOCENTE ESPECIALISTA

Ing. Luis Miguel Antaya Fariño, Mgtr.
DOCENTE GUÍA UIC

Ing. Corina Raquel Gonzabay De La A, Mgtr.
SECRETARIA

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Yépez Cevallos Víctor Richard, como requerimiento para la obtención del título de Ingeniero en Telecomunicaciones.

La Libertad, a los 21 días del mes de Agosto del año 2026

TUTOR

Ing. Edisson Stalin Coral Salinas, Mgtr.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, Víctor Richard Yépez Cevallos

DECLARO QUE:

El trabajo de integración curricular, (**Prototipo IoT de Monitoreo y Control de Variables Ambientales para un Cultivo Hidropónico de hortalizas en sectores Suburbanos de Ballenita**) previo a la obtención del título de Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de integración curricular referido.

La Libertad, a los 21 días del mes de Agosto del año 2026

EL AUTOR

Víctor Richard Yépez Cevallos

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTI-PLAGIO

Certifico que después de revisar el documento final del trabajo de integración curricular denominado (Prototipo IoT de Monitoreo y Control de Variables Ambientales para un Cultivo Hidropónico de hortalizas en sectores Suburbanos de Ballenita), presentado por el estudiante, Yépez Cevallos Víctor Richard fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 6%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

TUTOR

Ing. Edisson Stalin Coral Salinas, Mgtr.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **VÍCTOR RICHARD YÉPEZ CEVALLOS**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de integración curricular o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de integración curricular con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

La Libertad, a los 21 días del mes de Agosto del año 2026

EL AUTOR

Víctor Richard Yépez Cevallos

AGRADECIMIENTO

A Dios, por darme salud, fortaleza y la sabiduría necesaria para enfrentar los retos que encontré a lo largo de este camino.

A mis amados padres, quienes con su amor incondicional, sacrificios y enseñanzas, me motivan a esforzarme en cada paso que doy. Su perseverancia y valores son los cimientos con los que he levantado mis sueños.

A mis hermanas, por su compañía, apoyo y palabras de ánimo, que me brindaron la energía y motivación necesarias para seguir adelante, incluso cuando las circunstancias se tornaron más difíciles.

A mis profesores, por su paciencia, dedicación y orientación durante la realización de mis estudios. Sus conocimientos, enseñanzas y compromiso han sido fundamentales para mi desarrollo académico y personal.

Y en general a todas aquellas personas que, de una u otra manera contribuyeron a la realización de este trabajo de integración curricular, les expreso mi más sincero agradecimiento. Este logro no habría sido posible sin el apoyo, la confianza y las palabras de ánimo que recibí a lo largo de este proceso.

Víctor Richard, Yépez Cevallos

DEDICATORIA

A mis padres, siempre benévolos y por ser el motor de mi vida y enseñarme que con esfuerzo y perseverancia todo es alcanzable, A Sonia Cevallos y Víctor Yépez Bazán por mostrarme que nada era inalcanzable cuando se trabaja con dedicación y por brindarme su amor incondicional que es la razón de que pudiese superar cualquier tropiezo en el camino.

A mis hermanas, por su apoyo infalible y sus palabras de aliento, y porque son mi refugio cuando más lo necesito; su compañía ha sido la mayor alegría y la mejor inspiración.

A mis profesores, que han sabido compartir sus conocimientos con generosidad y, a la vez, han sido pilares para guiarme en mi camino, dedico con gratitud y respeto este esfuerzo. A ellos y a quienes creyeron en mí, les dedico también este logro, pues es el fruto del apoyo y la confianza que me brindaron.

Víctor Richard, Yépez Cevallos

I. DATOS GENERALES

Apellidos: Yépez Cevallos
Nombres: Víctor Richard
Número de cédula: 092708788-2
Número de matrícula: 20220300001
Carrera: Ingeniería en Telecomunicaciones
Correo electrónico: vyepezcevallos@gmail.com
Teléfono: 0980409940

Apellidos: Coral Salinas
Nombres: Edisson Stalin
Formación de Tercer Nivel: Ingeniero en Mecatrónica
Formación de Cuarto Nivel: Magister en electrónica y automatización
Correo electrónico: ecoral8368@upse.edu.ec

RESUMEN

La producción de hortalizas a través de la tecnología hidropónica es una realidad eficiente y de actualidad. Entre sus ventajas permite un mejor aprovechamiento del agua, reduce el efecto de factores ambientales y posibilita el control de las condiciones del cultivo.

En este contexto, el presente trabajo pretende mostrar el diseño e implementación de un prototipo IoT para el monitoreo y control inteligente de las variables ambientales de un sistema hidropónico. Se fusionaron las tecnologías de comunicación y los sistemas embebidos para vigilancia remota y automatización de procesos.

El sistema con una tarjeta ESP32 basada en el microcontrolador fue diseñado como nodo de adquisición única para recolectar, procesar y enviar datos. Sensores fueron añadidos para la medición de la temperatura, humedad relativa, potencia del hidrógeno (pH) y nivel del agua.

Se añadió además el control de los actuadores y se implementó el control automático de una bomba de recirculación, de un sistema de ventilación. Esto se hizo usando reglas de decisión dentro de un conjunto predefinido de rangos de operación. La infraestructura de comunicaciones fue diseñada sobre redes inalámbricas Wi-Fi basada en el estándar IEEE 802.11n, operando en la banda de 2.4 GHz mediante el módulo inalámbrico integrado del ESP32. También, se empleó el protocolo MQTT bajo el modelo Publicador/Suscriptor, utilizando una arquitectura de múltiples brokers para incrementar la disponibilidad del servicio y la continuidad de la transmisión de datos.

La información generada por el nodo sensor es distribuida en tiempo real hacia una aplicación de escritorio desarrollada en Python y una aplicación móvil para Android, permitiendo la supervisión simultánea del cultivo desde diferentes dispositivos.

La validación del prototipo se llevó a cabo con las pruebas funcionales de adquisición de datos, comunicaciones inalámbricas, transmisión vía MQTT, sincronización con otras aplicaciones cliente, operación automática y control manual.

Los resultados mostraron una performance estable de la infraestructura de comunicaciones, una sincronización apropiada de la información entre ambas plataformas, y una respuesta adecuada a modificaciones en las variables ambientales monitoreadas. En definitiva, el prototipo constituye una solución tecnológicamente viable para el desarrollo de sistemas inteligentes para agricultura, evidenciando la convergencia de tecnologías IoT, protocolos de comunicación, aplicaciones multiplataforma, y soluciones de inteligencia artificial como herramientas que potencian la generación de soluciones disruptivas en el ámbito de la Ingeniería en Telecomunicaciones.

Palabras clave: Internet de las Cosas (IoT), ESP32, MQTT, Wi-Fi, cultivo hidropónico, monitoreo ambiental, agricultura inteligente.

ABSTRACT

Hydroponic agriculture represents an efficient technological alternative for crop production by optimizing water use, minimizing the impact of environmental factors, and facilitating continuous monitoring of cultivation conditions.

In this context, the objective of this research was to design and implement an Internet of Things (IoT)-based prototype for the intelligent monitoring and control of environmental variables in a hydroponic cultivation system, integrating communication technologies and embedded systems to enable remote supervision and process automation.

The proposed system was developed using an ESP32 microcontroller as the central node for data acquisition, processing, and transmission.

It integrates sensors for measuring temperature, relative humidity, pH, and water level.

In addition, automatic control of actuators, including a water recirculation pump and a ventilation system, was implemented through decision rules based on predefined operating ranges.

The communication infrastructure was designed based on an IEEE 802.11n Wi-Fi wireless network, operating in the 2.4 GHz band through the ESP32's integrated wireless module. In addition, the MQTT protocol was employed under the Publish/Subscribe model, using a multi-broker architecture to increase service availability and ensure the continuity of data transmission.

The information generated by the sensor node is distributed in real time to a Python-based desktop application and an Android mobile application, enabling simultaneous monitoring of the hydroponic system from multiple devices.

The prototype was validated through functional test involving data acquisition, wireless connectivity, MQTT-based communication, synchronization among client applications, automatic, and manual operation.

The results demonstrated stable communication performance, efficient information synchronization between the developed platforms, and an adequate system response to changes in the monitored environmental variables.

In general, the proposed prototype proved to be a technically viable solution for the realization of smart agriculture systems, emphasizing the convergence of IoT technologies, communication protocols, cross-platform applications, and artificial intelligence as key elements for the development of innovative solutions in the domain of Telecommunications Engineering.

Keywords: Internet of Things (IoT), ESP32, MQTT, Wi-Fi, hydroponic cultivation, environmental monitoring, smart agriculture.

ÍNDICE GENERAL

Tabla de contenido

TRIBUNAL DE SUSTENTACIÓN .....**¡Error! Marcador no definido.**

CERTIFICACIÓN .....**¡Error! Marcador no definido.**

DECLARACIÓN DE RESPONSABILIDAD.....**¡Error! Marcador no definido.**

DECLARO QUE: .....**¡Error! Marcador no definido.**

CERTIFICACIÓN DE ANTI-PLAGIO.....**¡Error! Marcador no definido.**

AUTORIZACIÓN .....**¡Error! Marcador no definido.**

AGRADECIMIENTO 7

DEDICATORIA.....8

CAPÍTULO I 22

1. TÍTULO 22

1.1 Antecedentes.....22

1.2 Planteamiento del problema. 25

1.3 Descripción del proyecto.....25

1.4 Integración de la infraestructura de comunicaciones IoT.....26

1.5 Funcionamiento del sistema. 27

1.6 Objetivos.....29

1.7 Justificación.....30

1.8 Alcance del proyecto 32

1.9 Metodología.....34

CAPÍTULO II 36

2. MARCO TEÓRICO DE LA PROPUESTA.....36

2.1 Marco contextual.....36

2.2 Marco conceptual.....36

2.3 Componentes del sistema (hardware).....38

2.3.1 Sensores de adquisición de datos.....39

2.4 Infraestructura de software y comunicaciones.....41

2.5 Lógica de control y valores ideales.....42

2.6 Tecnologías de comunicación para sistemas IoT.....43

2.7 Aplicaciones del IoT en la agricultura inteligente.....47

2.8 Tendencias actuales del IoT aplicado a la agricultura.....50

CAPÍTULO III.....52

3. DESARROLLO DE LA PROPUESTA. 52

3.1. Arquitectura del sistema IoT.....52

3.2. Hardware y programación del nodo sensor.....56

3.3 Infraestructura de comunicaciones.....60

3.3.2 Configuración de la red WI-FI. 60

3.4 Desarrollo del firmware en arduino IDE. 62

3.5 Procesamiento de datos y centro de control de python. 63

3.6 Compilación y empaquetado con buildozer en google colab.....65

3.7 Generación del archivo APK. 66

3.8 Integración física del prototipo.....71

CAPÍTULO IV.....79

4. RESULTADOS Y ANÁLISIS 79

4.1. Validación del sistema 79

4.2. Discusión de resultados 88

4.3. Limitaciones del sistema 90

4.4. Cumplimiento de los objetivos del proyecto 92

CONCLUSIONES 94

RECOMENDACIONES 95

REFERENCIAS BIBLIOGRÁFICAS 96

ANEXOS 100

Código fuente del sistema desarrollado 100

Creación y pruebas del prototipo 111

Pruebas del prototipo instalado en la estructura 112

ÍNDICE DE FIGURAS

Figura 1: Esquema Aplicado del Modelo OSI. 33
Figura 2: Diagrama de Interacción y Flujo de Información IoT. 37
figura 3: Diagrama de Bloques de Control de Lazo Cerrado [26]. 37
Figura 4: Microcontrolador ESP32 Devkit V1 y Distribución. 38
Figura 5: Shield de Conexiones y Etapa de Potencia [28] 39
Figura 6: Sensor de Temperatura y Humedad DHT22 [29]. 39
Figura 7: Sensor de pH Analógico con Interfaz de Control [30]. 40
Figura 8: Sensor de Nivel de Líquido tipo Flotador [31]. 40
Figura 9: Arquitectura de Comunicación MQTT y Manejo de Hilos [32] 41
Figura 10: Despliegue con Buildozer [33] 42
Figura 11: Organigrama de Redes IoT según Capas Funcionales 47
Figura 12: Diagrama de Arquitectura IoT en Capas. 51
Figura 13: Arquitectura General del Sistema IoT Propuesto. 52
Figura 14: Arquitectura de Comunicaciones Basados en MQTT. 55
Figura 15: Diagrama Esquemático de captura de datos usando Proteus. 57
Figura 16: Diagrama Esquemático de Actuadores. 59
Figura 17: Fragmento del Código utilizado para Configuración de la Conexión Wi-Fi del ESP32 61
Figura 18: Prueba de Conectividad Wi-Fi en Monitor Serial 62
Figura 19: Diagrama de Procesamiento de Datos y Control de Python [33]. 64
Figura 20: Gestión de Hilos y Flujo MQTT [33]. 65
Figura 21: Instalación de Dependencias de Google Colab. 66
Figura 22: Carga del Script principal del Sistema. 67
Figura 23: Inicialización de la Herramienta Buildozer 68
Figura 24: Configuración del Archivo Buildozer. Spec. 68
Figura 25: Ejecución del Empaquetado Final 69
Figura 26: Ilustración de Control Automático en Interfaz. 70
Figura 27: Interfaz de Sensores Activos. 70
Figura 28: Panel Principal de Monitoreo 71
Figura 29: Uso del Calibrador Vernier para tomar Medidas Correctamente 72
Figura 30: Creación del Diseño del Box según las Medidas 72
Figura 31: Inserción de bloques según Distribución. 72
Figura 32: Diseño Final del Box incluida la tapa. 73

Figura 33: Colocación del Módulo Relé, ESP32 y Módulo Sensor de pH. 73

Figura 34: Instalación Acorde la Distribución de los Dispositivos. 74

Figura 35: Cableado de los de los Módulos y la ESP32. 74

Figura 36: Encapsulado de los Componentes Electrónicos del Sistema IoT 75

Figura 37: Vista Lateral de las Entradas de Alimentación de Energía del Prototipo. 75

Figura 38: Vista Frontal de las salidas de los Sensores, Actuador por cableado y por módulo de pH. 76

Figura 39: Estructura donde se Implementará las Pruebas del Prototipo. 76

Figura 40: Prueba del Funcionamiento del Prototipo. 77

Figura 41: Previa Instalación y Colocaciones Sensores 77

Figura 42: Adecuación del lugar para la ubicación del Prototipo. 78

Figura 43: Visualización final de cómo quedaría Conectado. 78

Figura 44: Incorporación de lechuguines de 2 semanas, visualizando óptimo crecimiento. 78

Figura 45: Cultivo Hidropónico de lechuguines de 5 días 81

Figura 46: Envío de Datos con muy poca Latencia. 84

Figura 47: Control Funcional Ventilador Apagado. 84

Figura 48: Conexión Wi-Fi excelente, Broker conectado. 85

Figura 49: Activación de Ventilación por Temperatura Elevada. 86

Figura 50: Sensor de nivel de agua antes de las Pruebas. 90

Figura 51: Sensor Luego de la exposición para controlar el pH del agua. 91

Figura 52: Correcto Funcionamiento del Apk. 92

Figura 53: Evidente Pérdida de la Transmisión de Datos debido a Mala Conexión de Wi-Fi 92

ÍNDICE DE TABLAS

Tabla 1. Rangos óptimos para el sistema hidropónico

Tabla 2. Resumen de configuración técnica

Tabla 3. Seguimiento del cultivo hidropónico

Tabla 4. Comparación de un sistema hidropónico convencional y el prototipo IoT.

Tabla 5. Cumplimiento de los objetivos específicos del proyecto

ÍNDICE DE ABREVIATURAS

ABREVIATURA	SIGNIFICADO
AC	**Corriente Alterna (Alternating Current)**
ADC	**Convertidor Analógico-Digital (Analog to Digital Converter)**
AI	**Inteligencia Artificial (Artificial Intelligence)**
APK	**Paquete de Aplicación para Android (Android Package Kit)**
API	**Interfaz de Programación de Aplicaciones (Application Programming Interface)**
BLE	**Bluetooth Low Energy**
CPU	**Unidad Central de Procesamiento (Central Processing Unit)**
DAC	**Convertidor Digital-Analógico (Digital to Analog Converter)**
DC	**Corriente Continua (Direct Current)**
DHT22	**Sensor Digital de Temperatura y Humedad**
ESP32	**Microcontrolador de Espressif con Wi-Fi y Bluetooth integrados**
FAO	**Organización de las Naciones Unidas para la Alimentación y la Agricultura**
GPIO	**Entrada/Salida de Propósito General (General Purpose Input/Output)**
GPRS	**Servicio General de Paquetes vía Radio (General Packet Radio Service)**
HTTP	**Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol)**
IDE	**Entorno de Desarrollo Integrado (Integrated Development Environment)**
IEEE	**Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers)**
IP	**Protocolo de Internet (Internet Protocol)**
ISM	**Banda Industrial, Científica y Médica (Industrial, Scientific and Medical)**
IoT	**Internet de las Cosas (Internet of Things)**
JSON	**Notación de Objetos de JavaScript (JavaScript Object Notation)**
MQTT	**Transporte de Telemetría por Publicación/Suscripción de Mensajes (Message Queuing Telemetry Transport)**
NDK	**Native Development Kit**
NFT	**Técnica de Película Nutritiva (Nutrient Film Technique)**
OSI	**Interconexión de Sistemas Abiertos (Open Systems Interconnection)**

pH	**Potencial de Hidrógeno**
RTU	**Unidad Terminal Remota (Remote Terminal Unit)**
SDK	**Kit de Desarrollo de Software (Software Development Kit)**
SPI	**Interfaz Periférica Serie (Serial Peripheral Interface)**
SSID	**Identificador del Conjunto de Servicios (Service Set Identifier)**
TCP	**Protocolo de Control de Transmisión (Transmission Control Protocol)**
TCP/IP	**Protocolo de Control de Transmisión / Protocolo de Internet**
UART	**Receptor/Transmisor Asíncrono Universal (Universal Asynchronous Receiver/Transmitter)**
USB	**Bus Serie Universal (Universal Serial Bus)**
UV	**Ultravioleta**
Wi-Fi	**Fidelidad Inalámbrica (Wireless Fidelity)**

CAPÍTULO I

1. TÍTULO

Prototipo IoT de Monitoreo y Control de Variables Ambientales para un Cultivo Hidropónico de Hortalizas en Sectores Suburbanos de Ballenita.

1.1 Antecedentes

La hidroponía, entendida como una técnica de cultivo sin suelo, tiene sus primeras referencias en civilizaciones antiguas como la babilónica y la azteca, donde se utilizaron métodos rudimentarios para cultivar en medios acuosos.

Pero fue apenas a principios del siglo XX cuando la hidroponía empezó a desarrollarse científicamente, sobre todo debido a los trabajos del profesor William F. Gericke de la Universidad de California, que demostró que era posible cultivar plantas utilizando solo soluciones nutritivas y no tierra. Desde entonces, esta técnica ha avanzado hasta situarse como una alternativa real para la producción agrícola en ambientes urbanos, zonas secas y espacios confinados [1].

Al mismo tiempo, la automatización de las máquinas agrícolas fue estimulada por el desarrollo de la electrónica y telecomunicaciones en la segunda mitad del siglo XX. Durante la década de 1990 se desarrolló el concepto de Internet de las Cosas (IoT, por sus siglas en inglés), acuñado por Kevin Ashton en 1999, que consiste en la conectividad de objetos y dispositivos a través de Internet, para que éstos puedan comunicarse y generar redes inteligentes [2].

Esta integración tecnológica hace viable la incorporación de sensores y microcontroladores con plataformas digitales para la adquisición de datos y control de variables como la temperatura, humedad, etc., sentando las bases de lo que ahora se conoce como agricultura de precisión.

Los prototipos son versiones preliminares de un sistema que cuentan con funcionalidades básicas y se desarrollan para validar conceptos antes de realizar su desarrollo completo.

Sirven para experimentar con conceptos y evaluar una idea o proyecto en concreto.

En agricultura los prototipos basados en el IoT son ahora importantes, ya que permiten monitorizar parámetros como humedad, temperatura, pH o luz, entre otros, para potenciar el uso de recursos y aumentar el rendimiento en los cultivos hidropónicos. La Organización para la Agricultura y la Alimentación (FAO) en su informe 2015 advirtió que será necesario un aumento sustancial en la producción alimentaria para satisfacer las demandas generadoras de una cada vez mayor población [3].

Se estima que en el año 2029 la población mundial alcanzará los 10.000 millones de personas, lo que representa una presión creciente sobre el sector agrícola. En este sentido, la agricultura enfrenta limitaciones estructurales como la reducción de áreas de cultivo debido al avance de las zonas urbanas, el uso de suelos para biocombustibles, la escasez de agua y el impacto del cambio climático.

Ante esta situación, surge la necesidad de producir más alimentos con menos mano de obra, lo que exige la incorporación de tecnologías innovadoras capaces de garantizar seguridad alimentaria y sostenibilidad en el tiempo [4].

En países latinoamericanos, la aplicación de dispositivos IoT en la agricultura inteligente ofrece beneficios significativos, como la recopilación de datos en la nube, el monitoreo en tiempo real y la generación de alertas personalizadas para garantizar condiciones óptimas de los cultivos.

Esto contribuye a los Objetivos de Desarrollo Sostenible, en especial al de “Hambre Cero”, dado que alrededor del 80 % de los alimentos en el mundo proviene de pequeñas fincas agrícolas. Sin embargo, la adopción de estas tecnologías enfrenta barreras, como la resistencia al cambio y los retos en la gestión empresarial [5].

Sudamérica es uno de los mayores mercados con potencial en la agroindustria.

A pesar de que las leyes han evolucionado recientemente y existen iniciativas de innovación propias, resulta difícil que la gente se adapte a esta nueva era de actualizaciones tecnológicas debido al proceso de gestión de la empresa y de las áreas de mercado [6].

En Ecuador, el prototipado IoT para cultivos hidropónicos se ha popularizado en los últimos años como una solución para aumentar la producción y para lidiar con retos relacionados al medio ambiente.

Recientes investigaciones indicaron que estos sistemas pueden aumentar la productividad de agua y nutrientes y la seguridad alimentaria. Un caso es el proyecto desarrollado en la Escuela Marieta de Veintimilla, en Loja, que plantea un sistema IoT con sensores de humedad, temperatura y radiación UV, haciendo uso de la red Sigfox para la transmisión de los datos en tiempo real a través de la plataforma ThingSpeak [7].

El sistema incluso integró modelos de predicción basados en redes neuronales para anticipar el comportamiento de variables críticas.

De manera complementaria, en otros países de la región y del mundo se han desarrollado iniciativas que evidencian el potencial del IoT en la hidroponía.

En Colombia, uno de los ejemplos en donde se han llevado a cabo proyectos con controladores lógicos programables es en la Universidad Tecnológica de Pereira, titulado "Automatización de la técnica de hidroponía NFT en invernadero, con monitoreo web", el cual evidencia el uso de sistemas de control interconectados comunicándose mediante el protocolo MODBUS RTU para el envío de datos; El sistema realizó acciones de acuerdo con ciertas entradas y salidas digitales configuradas para la operación de los actuadores [8].

Por último, dado el uso masivo a nivel global de sistemas embebidos para la transmisión de datos y señales en tiempo real.

En Indonesia, estudiantes de la Universidad de Sumatra Utara crearon un sistema para monitorear cultivos hidropónicos a través de comunicación GPRS para enviar la información.

El sistema consta de un hardware de procesamiento que se conecta a sensores de temperatura, pH y conductividad eléctrica. Los datos se transmiten a un servidor para el monitoreo remoto de los parámetros ambientales y nutricionales de la planta. Los autores recomendaron realizar el cultivo en interiores o en biomas protegidos para obtener resultados superiores [9].

1.2 Planteamiento del Problema.

En áreas rurales, la disponibilidad limitada de las tecnologías de IoT y herramientas de monitoreo limita el rendimiento eficiente de los recursos agrícolas. La introducción de un sistema basado en IoT para monitorizar variables ambientales tales como temperatura, humedad, pH y nivel de agua dotaría a los agricultores de la información necesaria para que adopten mejores decisiones. Por medio de la conectividad y las telecomunicaciones, estos datos también son transmitidos, facilitando la producción agrícola y contribuyendo a la solución de los problemas ambientales en estas zonas.

Esta problemática implica la necesidad de poder desarrollar un prototipo IoT viable, que emplee protocolos y servicios de comunicación que sean de bajo costo y fácil implementación, y que pueda ser utilizado en el medio rural o en lugares con pocas posibilidades. La combinación de tecnologías inalámbricas en bandas ISM, con el protocolo MQTT y aplicaciones de monitoreo, permite una solución técnica que podría acercar la agricultura controlada a regiones en vías de desarrollo.

1.3 Descripción del Proyecto

El proyecto se centra en el diseño de un prototipo de Internet de las Cosas (IoT) para monitoreo de variables ambientales (temperatura, humedad) y control de actuadores en un cultivo hidropónico de hortalizas en un área suburbana de Ballenita. La hidroponía, un método de cultivo sin suelo, necesita un control riguroso de múltiples parámetros para garantizar el desarrollo adecuado de las plantas. El presente trabajo persigue el diseño e implementación de un sistema integral para el monitoreo en tiempo real, y el acondicionamiento climatológico de

las condiciones ambientales del cultivo. Desde el punto de vista de la ingeniería en telecomunicaciones, este trabajo se centra en el análisis de la tecnología utilizada en agricultura, mediante la utilización de elementos que mejoran la conectividad entre los dispositivos y la transmisión de datos. El microcontrolador ESP32 es el corazón del sistema, dado su alta capacidad y conectividad Wi-Fi y Bluetooth. Este aparato no solamente controla la comunicación entre todos los sensores, sino que además se acumula y transmite los datos de una manera eficiente utilizando el protocolo MQTT a las aplicaciones que se desarrollen para el monitoreo y control del cultivo, lo cual es esencial para la integración de soluciones IoT en ambientes agrícolas. El sistema tiene múltiples sensores críticos de monitoreo ambiental.

Un sensor de temperatura ambiental permite medir la temperatura del ambiente en que se cultivan las plantas, un factor clave para su crecimiento y salud.

Esta información se utiliza para modificar el ambiente y mantener unas condiciones de temperatura aceptables.

Un sensor de nivel de agua también se incluye con el fin de informar en tiempo real de la cantidad de agua que hay en el sistema hidropónico, detectando así situaciones que puedan comprometer el funcionamiento del cultivo. Además, el prototipo dispone de un sensor de pH que evalúa la acidez o alcalinidad del agua, y que es fundamental en el acceso a los nutrientes. Este sensor proporciona lecturas continuas para monitorear el nivel de pH y mantener un ambiente compatible con el cultivo.

1.4 Integración de la Infraestructura de comunicaciones IoT

La infraestructura de comunicaciones del sistema fue diseñada utilizando el protocolo MQTT sobre redes inalámbricas Wi-Fi IEEE 802.11n, permitiendo el intercambio de información entre el nodo sensor basado en Esp32 y las aplicaciones desarrolladas para la supervisión del cultivo hidropónico.

Para incrementar la disponibilidad del servicio y mejorar la continuidad de la transmisión de datos, se implementó una arquitectura de múltiples brokers MQTT, conformada por test.mosquitto.org, broker.hivemq.com y broker.emqx.io, los cuales actúan como intermediarios entre el nodo publicador y las aplicaciones cliente, esta información se encuentra más detallada en el capítulo II y III respectivamente.

El Esp32 procesa la información obtenida por los sensores de temperatura, humedad relativa, pH y nivel de agua, publicando periódicamente estos datos mediante tópicos MQTT.

Posteriormente, tanto la aplicación de escritorio desarrollada en Python como la aplicación móvil para Android se suscriben a dichos tópicos para recibir y visualizar la información en tiempo real.

Esta arquitectura es suficiente para cubrir la comunicación en ambas direcciones, y además permite enviar órdenes para controlar manualmente los actuadores cuando se requiera.

La aplicación del patrón de diseño Publicador/Suscriptor ofrece una solución escalable y desacoplada, de modo que se pueden añadir nuevos dispositivos o aplicaciones cliente sin que ello requiera modificar el nodo sensor o la infraestructura de comunicaciones.

1.5 Funcionamiento del Sistema.

El funcionamiento del prototipo inicia con la adquisición de las variables ambientales mediante los sensores instalados en el cultivo hidropónico.

El Microcontrolador Esp32 realiza el procesamiento de las mediciones obtenidas y ejecuta la lógica de control establecida para el accionamiento automático de actuadores como un sistema de ventilación, de acuerdo con los rangos definidos para cada variable.

Una vez procesada la información, el Esp32 transmite los datos mediante el protocolo MQTT hacia la infraestructura de brokers, desde donde son distribuidos simultáneamente a la aplicación de escritorio y a la aplicación móvil Android para su monitoreo en tiempo real.

Ambas aplicaciones permiten visualizar el estado del sistema, supervisar las variables ambientales y ejecutar acciones de control manual cuando el usuario lo requiera.

1.5.1 Impacto Esperado

El prototipo desarrollado contribuye a la incorporación de tecnologías de Internet de las Cosas e inteligencia artificial en sistemas de agricultura inteligente, proporcionando una solución de bajo costo para el monitoreo y control de cultivos hidropónicos.

Gracias a la integración de redes Wi-Fi, el protocolo MQTT, aplicaciones multiplataforma y técnicas de procesamiento inteligente se logra obtener una mejor monitorización del cultivo, un uso más eficiente de los recursos y una toma de decisiones más rápida. Desde la visión de la Ingeniería en Telecomunicación, el proyecto acredita la utilización de arquitecturas de comunicación IoT, protocolos de mensajería y sistemas distribuidos para la transmisión fiable de información, como una base tecnológica escalable para desarrollos posteriores orientados a la automatización agropecuaria y la transformación digital del sector productivo.

1.6 Objetivos

1.6.1 Objetivo General

Desarrollar un prototipo de sistema de control y monitoreo de variables ambientales (pH, temperatura, humedad y nivel de agua) que permita la recolección, análisis y visualización de datos en tiempo real, con el fin de optimizar la gestión y conservación de recursos naturales en un entorno específico.

1.6.2 Objetivos Específicos

- Implementar un sistema para la medición de variables ambientales críticas del cultivo, tales como temperatura, humedad relativa y pH mediante la integración de sensores analógicos para la obtención de información en tiempo real que permita el monitoreo.

- Diseñar de un sistema de control automatizado utilizando un Microcontrolador ESP32, sensores de nivel y dispositivos eléctricos para energizar o desenergizar equipos.

- Construir una infraestructura de comunicación IoT para transportar los datos obtenidos en tiempo real a una plataforma de monitoreo central que permita la supervisión remota y el registro histórico utilizando protocolos de comunicación IoT y microcontrolador.

- Implementar una interfaz de usuario sencilla y amigable que permita visualizar y almacenar las variables monitoreadas y que permita también mediante comandos enviando comandos MQTT controlar manualmente los actuadores.

1.7 Justificación

El prototipo IoT para monitoreo y control de variables ambientales en cultivos hidropónicos de hortalizas en sectores suburbanos es factible no sólo para aumentar la producción agrícola, sino también por su importancia en la ingeniería de telecomunicaciones.

Esta perspectiva combina diferentes tecnologías que son esenciales para que esta área evolucione y se enfrenta a retos actuales en la agricultura urbana. El Internet de las Cosas (IoT) representa una convergencia significativa entre la tecnología de la información y las telecomunicaciones.

Según Mishra "IoT permite la interconexión de dispositivos a través de redes inalámbricas, facilitando la recolección y análisis de datos en tiempo real" [9].

El seguimiento y control agrícola a través de IoT brinda a los ingenieros en telecomunicaciones la oportunidad de aplicar sus conocimientos en redes, comunicación inalámbrica y procesamiento de datos.

La recolección y transmisión de datos en tiempo real de variables críticas como la temperatura, humedad, nivel del agua y pH, es solo un ejemplo de cómo las telecomunicaciones pueden aumentar la eficiencia en la agricultura [10].

La automatización, por medio de tecnologías IoT, del monitoreo de cultivos es necesaria para la maximización de recursos en áreas suburbanas, con escaso tiempo y atención humana.

Según Zhang," la automatización en agricultura no solo mejora la eficiencia operativa, sino que también permite un uso más sostenible de los recursos" [11].

En esta línea, los ingenieros de telecomunicaciones tienen una importancia fundamental en el diseño y desarrollo de las redes que posibilitan la comunicación entre dispositivos, y en que éstas aseguren la transmisión eficiente y segura de los datos [12].

Así se refuerza la gestión del cultivo y también se impulsa el desarrollo de nuevas soluciones que podrán aplicarse en otros sectores.

La posibilidad de disponer de los datos de los sensores desde cualquier lugar permite una gestión remota eficaz, lo que atesora un valor en alza dada la rápida expansión de las áreas urbanas.

Esta tendencia, además de fomentar la agricultura sostenible, pone en valor la figura del ingeniero en telecomunicaciones como un actor clave para la transformación digital del sector del agro [13].

En consecuencia, el proyecto favorece el interés de los estudiantes de ingeniería en telecomunicaciones para buscar aplicaciones reales de sus conocimientos.

Mediante la participación en el diseño y la implementación de soluciones IoT para la agricultura, los estudiantes obtienen una serie de habilidades que son muy valoradas en la industria, [14].

No solo fortalece su formación académica, sino que impacta en el desarrollo tecnológico local con una visión innovadora en la resolución de problemas de actualidad.

Por lo tanto, el prototipo IoT para monitoreo y control de cultivos de hidroponía no solo va dirigido a atender una necesidad prioritaria de mejorar la producción agrícola en medios suburbanos, sino que además constituye para la ingeniería en telecomunicaciones una oportunidad para desarrollar e innovar.

La integración de tecnologías avanzadas con criterios de sostenibilidad hace de este proyecto una referencia para posibles desarrollos en cuanto a tecnología en el campo.

1.8 Alcance del proyecto

El proyecto de titulación se basa en el diseño y desarrollo de un prototipo para monitoreo y control de variables ambientales en un cultivo hidropónico de hortalizas en áreas suburbanas de Ballenita a través de IoT. El objetivo general es diseñar un sistema de comunicación para el monitoreo remoto y en tiempo real de temperatura, humedad, ph y nivel de agua para aumentar el rendimiento del cultivo. En este proyecto, se utilizará una infraestructura de comunicación basada en el protocolo MQTT para la transmisión y visualización de datos mediante aplicaciones desarrolladas, facilitando la toma de decisiones basadas en información en tiempo real [15].

Los sensores IoT enviarán información clave sobre las condiciones del cultivo a través de una red inalámbrica Wi-Fi IEEE 802.11n en la banda de 2.4 GHz, garantizando una transmisión eficiente y constante con un mínimo de recursos energéticos.

Las aplicaciones desarrolladas serán responsables de recibir los datos transmitidos y presentarlos en una interfaz gráfica intuitiva, donde los usuarios podrán visualizar las condiciones del cultivo en tiempo real y el estado actual de los actuadores [16].

En cuanto a la estructura de red, el proyecto seguirá el modelo OSI.

En la capa física (Capa 1), se encargará de la comunicación inalámbrica, transmitiendo los datos mediante Wi-Fi 802.11n en la banda de 2.4 GHz.

En la capa de enlace de datos (Capa 2), se gestionará la confiabilidad de la transmisión de datos entre los sensores y la estación base.

En la capa de red (Capa 3), se transportarán los datos mediante los protocolos TCP/IP hacia los brokers MQTT gestionados y distribuidos a los clientes suscritos [17].

En las capas superiores del modelo OSI (Capas 5, 6 y 7), las aplicaciones desarrolladas desempeñarán un rol crucial, permitiendo a los usuarios interactuar con los datos de manera dinámica, tal como se muestra en el esquema de la **Figura 1**.

Desde su interfaz, el agricultor puede consultar la visualización de las variables monitorizadas y utilizar los actuadores para que tome decisiones informadas sobre el manejo del cultivo.

Esta integración haría disponer de una potente herramienta para el control remoto y eficiente del cultivo hidropónico, lo cual tiene mucha importancia en un escenario donde la escasez de agua y las dificultades climáticas ponen en riesgo la seguridad alimentaria y la sostenibilidad agrícola [18].

La habilidad para observar y controlar estos recursos es también crucial para satisfacer las crecientes necesidades de producción de alimentos en un ambiente de agua limitada [19].

De la misma manera, aplicar tecnologías IoT en agricultura permite mejorar la gestión del agua y aumentar la productividad, que es clave para asegurar el acceso a alimentos nutritivos y asequibles [20].

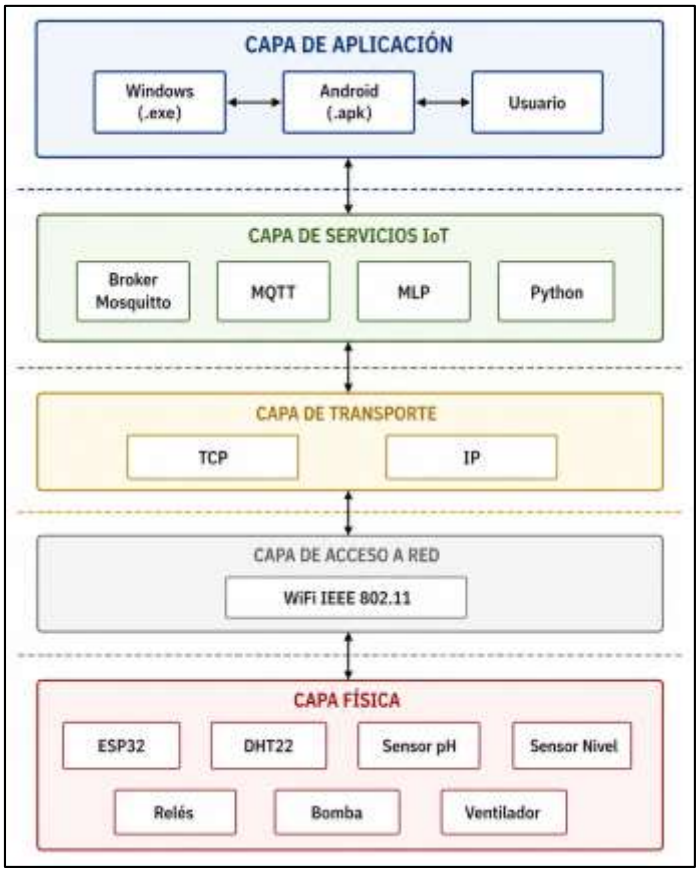


Figura 1: Esquema aplicado del modelo OSI.

1.9 Metodología

1.9.1 Investigación descriptiva

Esta investigación será de tipo descriptivo, que tiene como propósito describir las características del fenómeno de estudio, atendiendo al desarrollo de un prototipo IoT en el área de telecomunicaciones para agricultura. El trabajo está orientado a una selección de las variables ambientales que influyen en el cultivo hidropónico (temperatura, humedad, pH y nutrientes) y tendrá un punto adicional sobre cómo las redes de comunicación IoT pueden hacer seguimiento de esta variable. También serán tomadas en cuenta las condiciones particulares del entorno del sitio en Ballenita, Santa Elena que podrían afectar al rendimiento de hortalizas.

Dos métodos se utilizarán: observación de primera mano en la plantación para registrar las condiciones ambientales y las prácticas actuales, y una encuesta a los agricultores locales para conocer de primera mano sus experiencias y problemas en el monitoreo de cultivos mediante sistemas IoT.

1.9.2 Investigación documental

Esta metodología se basa en la consulta de fuentes ya existentes, para obtener información relevante sobre la teoría y práctica del IoT en Telecomunicaciones Aplicadas a la Agricultura, y también para conocer estudios previos relacionados con el empleo de sistemas de monitoreo y control en cultivos hidropónicos.

Las fuentes a tener en cuenta son artículos de investigación y libros que tratan sobre tecnologías de telecomunicaciones, por ejemplo Redes IoT y Comunicación Inalámbrica, aplicadas al monitoreo agrícola, con ello se espera conocer sobre tales tecnologías y sobre las técnicas de implementación de las mismas, así también informes técnicos y documentos gubernamentales que traten sobre la aplicación de tecnologías IoT en Ecuador.

1.9.3 Investigación aplicada

La investigación aplicada que se desarrolla en este trabajo se centra en la solución de problemas concretos en telecomunicaciones, se presenta el desarrollo de un prototipo IoT para la observación y control de variables ambientales en cultivos de hidroponía. eficiente en energía artículo relacionado Proceso: múltiples etapas se prevé que el proceso tendrá varias etapas: en primer lugar, la construcción del prototipo, la selección de componentes (sensores, microcontroladores y software de comunicación);

Posteriormente, la etapa de desarrollo del prototipo, que contempla la instalación de sensores para temperatura, humedad, pH, nivel de agua plus actuadores para el control ON/OFF de los dispositivos del sistema en un ambiente controlado a fin de realizar pruebas preliminares; y, en tercero, análisis de datos para determinar la eficacia y realizar modificaciones que mejoren el rendimiento del cultivo.

CAPÍTULO II

2. MARCO TEÓRICO DE LA PROPUESTA

2.1 Marco Contextual

La parroquia Ballenita, en el cantón Santa Elena, posee una vocación turística que genera una importante actividad económica estacional. La introducción de sistemas hidropónicos controlados por IoT surge como una alternativa de diversificación productiva. Debido a la escasez de suelos fértiles y recursos hídricos en la zona, la técnica NFT (Nutrient Film Technique) representa una opción eficiente para el cultivo en espacios reducidos y entornos suburbanos [21].

2.2 Marco Conceptual

2.2.1 Internet de las Cosas (IoT)

El IoT se define como una red de objetos físicos equipados con sensores y software que permiten el intercambio de datos. En el presente proyecto, esta tecnología se emplea para el monitoreo constante de variables críticas como el pH, la temperatura y la humedad, facilitando la ejecución de acciones automáticas mediante actuadores gestionados desde la nube. La interacción entre estos componentes y el flujo de información se detalla en la **Figura 2.**

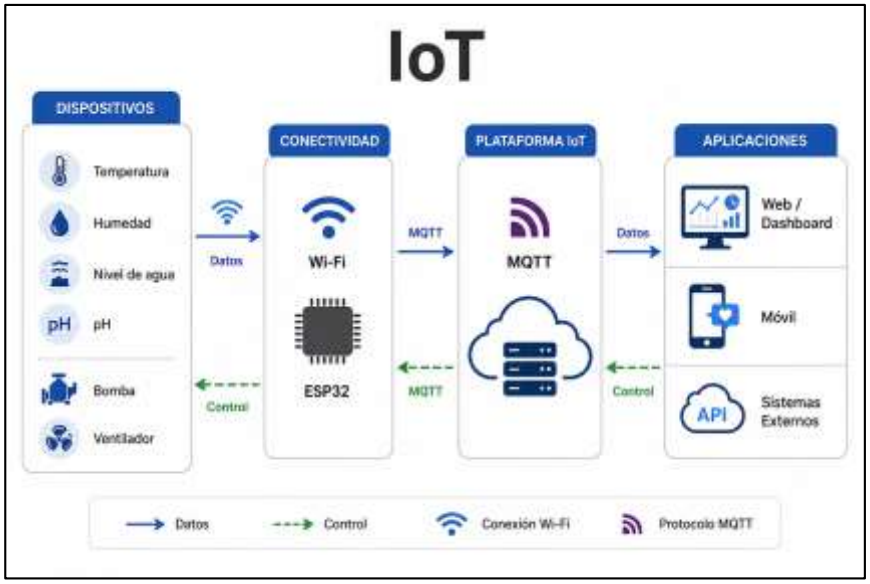


Figura 2: Diagrama de interacción y flujo de información IoT.

2.2.2 Arquitectura de Control de Lazo Cerrado.

El sistema implementa un control de lazo cerrado (feedback), donde los datos recolectados por los sensores son comparados con valores de consigna o setpoints predefinidos. La estructura de este flujo de control, que permite la transición entre la autonomía del sistema y la intervención del usuario, se presenta en la **Figura 3**.

Modo Automático: El microcontrolador ejecuta algoritmos de decisión autónomos basándose en rangos óptimos de cultivo.

Modo Manual (Override): El usuario interviene a través de una interfaz de supervisión para forzar estados de los actuadores, otorgando prioridad a la gestión humana sobre la lógica programada.

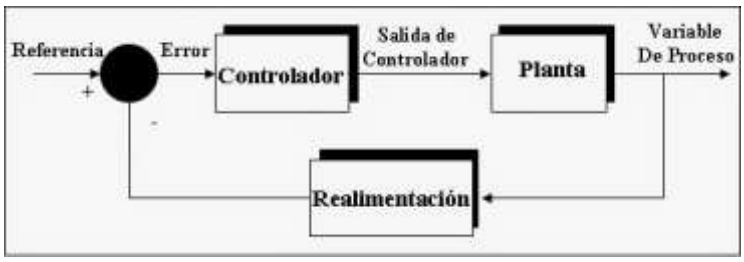


Figura 3: Diagrama de bloques de control de lazo cerrado [22].

2.3 Componentes del Sistema (Hardware)

2.3.1 Microcontrolador ESP32 DevKit V1

Es el núcleo del sistema. Se destaca por su arquitectura de doble núcleo a 240 MHz y su conectividad Wi-Fi nativa, permitiendo procesar tareas de control y comunicación de forma asíncrona. En la **Figura 4** se detallan los componentes y la distribución de pines del dispositivo utilizado.

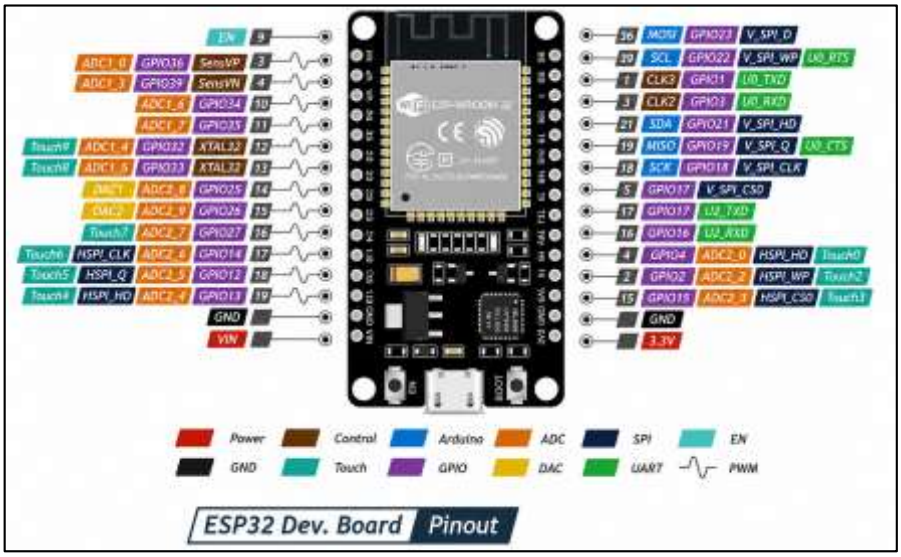


Figura 4: Microcontrolador Esp32 devkit v1 y distribución.

2.3.2 Shield de Expansión y Actuadores

El uso de una shield de expansión permite centralizar las conexiones, garantizando la estabilidad eléctrica de los sensores y la distribución de potencia hacia los actuadores. El sistema emplea una fuente conmutada de 12V DC para alimentar la electrobomba de nutrientes y el ventilador de control térmico. Los componentes y la etapa de potencia se detallan en la **Figura 5.**

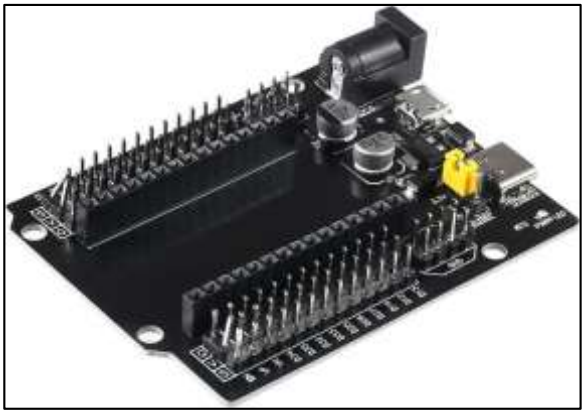


Figura 5: Shield de conexiones y etapa de potencia [23].

2.3.1 **Sensores de Adquisición de Datos**

Para la recolección de parámetros físicos y químicos, se han seleccionado dispositivos de precisión que permiten el monitoreo en tiempo real:

DHT22: Sensor digital para el monitoreo de temperatura y humedad ambiental.

Figura 6: Sensor de temperatura y humedad DHT22 [24].

Sensor de pH: Dispositivo analógico que determina la acidez de la solución, vital para la biodisponibilidad de nutrientes.

Figura 7: Sensor de pH analógico con interfaz de control [25].

Sensor de Nivel: Mecanismo de seguridad que previene el funcionamiento en seco de la bomba.

Figura 8: Sensor de nivel de líquido tipo flotado [26].

2.4 Infraestructura de Software y Comunicaciones

2.4.1 Protocolo MQTT y Procesamiento Multihilo

MQTT es un protocolo ligero basado en la publicación y suscripción de mensajes. Para gestionar esta comunicación en entornos de alta demanda, se emplea la técnica de *threading* (hilos), que permite a la aplicación de supervisión recibir datos en segundo plano sin interrumpir la fluidez de la interfaz de usuario. En la **Figura 9** se ilustra el flujo de datos entre el *broker* y los procesos paralelos.

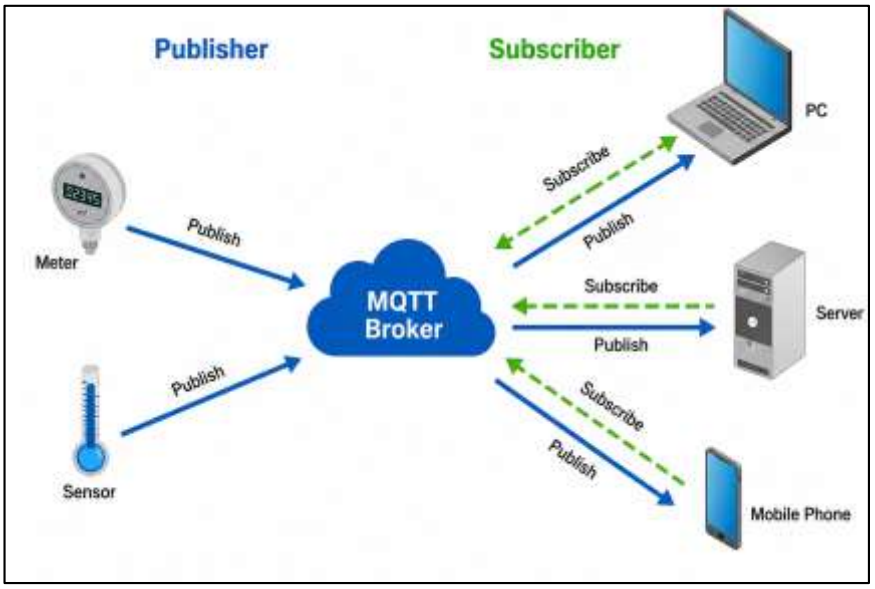


Figura 9: Arquitectura d comunicación MQTT y manejo de hilos [27].

2.4.2 Entorno de Desarrollo y Compilación.

Para el procesamiento inteligente y la interfaz de usuario, se utilizan librerías científicas como Scikit-learn (para la RNA) y el *framework* KivyMD. El despliegue en dispositivos Android se fundamenta en herramientas de compilación cruzada como Buildozer, que empaqueta el código Python en binarios nativos (.apk). El flujo de compilación se compila con el logo en la **Figura 10**.

Figura 10: Despliegue con Buildozer [28].

2.5 Lógica de Control y Valores Ideales

El control automático se rige por parámetros agronómicos que definen el estado de salud de la planta. Cuando el sistema detecta una desviación, envía señales digitales para retornar a los niveles de consigna, de acuerdo con los rangos establecidos en la Tabla 2.1.

Tabla 1 Rangos óptimos para el sistema hidropónico

Variable	Rango Ideal	Acción de Control
Temperatura	18°C – 26°C	Activa Ventilador si T 26°C
pH	5.5 - 6.5	Alerta visual en aplicación móvil
Humedad	60 % - 80 %	Notificación de ventilación
Nivel de Agua	Lleno / Medio	Bloqueo de bomba en nivel crítico

2.6 Tecnologías de Comunicación Para Sistemas IoT

2.6.1 Redes Inalámbricas Para Dispositivos IoT

Las redes inalámbricas constituyen uno de los componentes fundamentales del Internet de las Cosas, ya que permiten establecer comunicación entre dispositivos sin necesidad de utilizar enlaces cableados. Su uso minimiza la complejidad de instalación, facilita la escalabilidad de sistema, y posibilita desplegar soluciones en ambientes donde el cableado se considera impráctico o demasiado costoso.

Entre las diversas tecnologías inalámbricas disponibles para aplicaciones IoT se incluyen Wi-Fi, Bluetooth Low Energy (BLE), Zigbee, LoRaWAN, Sigfox y las redes celulares. Cada una de éstas presenta diferentes características en cuanto a alcance, consumo energético, velocidad de transmisión o tipo de aplicación para las que fueron diseñadas. En sistemas de monitoreo y control para cultivos hidropónicos de pequeña y mediana escala, la tecnología **Wi-Fi** representa una de las alternativas más utilizadas debido a su amplia disponibilidad, facilidad de integración y capacidad para transmitir datos hacia servidores locales o plataformas en la nube.

Wi-Fi corresponde a la familia de estándares **IEEE 802.11**, desarrollados por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), los cuales especifican los mecanismos de acceso al medio, las características físicas de transmisión y los procedimientos necesarios para establecer comunicación inalámbrica entre dispositivos dentro de una red local (WLAN). Las diferentes versiones del estándar 802.11 presentan características específicas en cuanto a velocidad de transmisión, alcance y rendimiento de la comunicación. Entre estas versiones se encuentran 802.11b, 802.11g y 802.11n, las cuales han incorporado mejoras en la capacidad de transmisión y en el desempeño de redes inalámbricas, permitiendo su utilización en diferentes escenarios de comunicación.

En aplicaciones IoT, el uso de Wi-Fi permite aprovechar las redes inalámbricas disponibles en casa, laboratorio o institución educativa, sin la obligación de instalar infraestructura de comunicación adicional. Facilita el acceso directo a servicios en la nube vía TCP/IP para obtener la supervisión online de las variables medidas por los sensores y para enviar las órdenes de control a los actuadores del sistema. Y es por eso, que entre las principales ventajas que ofrece Wi-Fi para soluciones de IoT tenemos: Amplia disponibilidad de infraestructura de red.

- Transferencia rápida de datos
- Se integra fácilmente con servicios en línea
- Compatible con IoT y aplicaciones móviles
- Fácil de configurar y mantener

No obstante, esta tecnología también presenta algunas limitaciones, entre ellas un mayor consumo energético respecto a tecnologías como Zigbee o LoRaWAN, así como un alcance condicionado por la cobertura del punto de acceso inalámbrico.

La elección de Wi-Fi responde, además, a las condiciones de operación del prototipo, ya que el cultivo hidropónico se encuentra ubicado dentro de un entorno con cobertura inalámbrica disponible. En estas circunstancias, esta tecnología proporciona una solución confiable para el intercambio de datos entre el sistema de monitoreo y las aplicaciones desarrolladas para la supervisión y el control de las variables ambientales.

2.6.2 Protocolos y Formatos de Comunicación en IoT

Los protocolos de comunicación constituyen un componente esencial en los sistemas IoT, ya que establecen las reglas para el intercambio de información entre dispositivos, aplicaciones y servicios conectados a la red. Su utilización garantiza que los datos sean transmitidos de manera organizada, confiable y compatible entre los diferentes elementos que

conforman la arquitectura del sistema.

En el presente proyecto se emplean diversos protocolos y formatos de intercambio de información, cada uno con funciones específicas dentro del proceso de comunicación.

TCP/IP (Transmission Control Protocol / Internet Protocol)

Constituye la base de las comunicaciones en Internet y en las redes locales. El direccionamiento a los dispositivos conectados lo realiza el protocolo IP, y TCP se ocupa de que los datos se transmitan correctamente hasta el destino a través de procedimientos para el control de errores y de la retransmisión en caso de haber errores. Con este protocolo de conjunto el Esp32 puede enviar y recibir en otros dispositivos por medio de la red Wi-Fi.

HTTP (Hypertext Transfer Protocol)

Es un protocolo de nivel de aplicación para el intercambio de información entre cliente y servidor. Se opera bajo un modelo solicitud-respuesta, a través de éste un dispositivo solicita recursos o envía datos a otro. Si bien en aplicaciones con IoT se emplea por lo general para operaciones relacionadas con configuración, consulta o integración con servicios web, el mismo consume mayores recursos que otros protocolos ideados para dispositivos embebidos.

MQTT (Message Queuing Telemetry Transport)

Es un protocolo ligero diseñado para aplicaciones IoT y sistemas con recursos limitados. Basa su interacción en un modelo de comunicación pub-sub (publicador/suscriptor) en el que un agente llamado *broker* se ocupa de gestionar la comunicación, recepción y envío de mensajes, entre las máquinas que están conectadas. Este proceso atenúa el tráfico de red y hace posible la transmisión de datos eficientemente aun existiendo varios nodos que se comunican al mismo tiempo.

Es un formato reducido para el intercambio de información en datos estructurados.

La información está estructurada como pares clave–valor, por lo que los microcontroladores y aplicaciones pueden fácilmente procesar y entender la información dentro de los datos. Su simplicidad y tamaño pequeño hace que sea uno de los formatos más utilizados en aplicaciones del IoT. En alguna medida estas capas y protocolos juntos es lo que hace que la comunicación dentro del sistema propuesto sea efectiva. En este proyecto el Esp32 se conecta a la red a través de Wi-Fi, se sirve de la pila de protocolos TCP/IP para enviar la información, utiliza MQTT como método de comunicación principal, y estructura la información captada por los sensores en formato JSON para realizar el intercambio con ambas aplicaciones de monitoreo y control desarrolladas para el sistema operativo Android y Windows.

2.6.3 Arquitectura de Redes IoT.

La arquitectura de redes IoT se organiza en capas funcionales para distribuir las tareas del sistema de forma ordenada y eficiente, como se indica en la **Figura 11**, cada capa cumple una función específica, desde la captura de datos hasta su transmisión, procesamiento y visualización.

Figura 11: Organigrama de redes IoT según capas funcionales

2.7 Aplicaciones del IoT en la Agricultura Inteligente

2.7.1 Monitoreo remoto de cultivos.

El monitoreo a distancia de cultivos es la recopilación, transmisión y monitoreo de información sobre las condiciones de los cultivos a través de tecnologías de comunicación y dispositivos IoT. Su finalidad es entregar información en tiempo real para hacer más sencillo el seguimiento del estado del cultivo y ayudar en la toma de decisiones sin que sea necesaria la realización de inspecciones continuas en terreno.

En función de cómo se adquiere y administra la información, el monitoreo remoto puede clasificarse en:

• Monitoreo periódico

Aquí se trata de tomar datos de las variables del cultivo a intervalos regulares de tiempo. Los sensores miden de manera programada y envían la información a un servidor para su almacenamiento y análisis. Este tipo de vigilancia es apropiada cuando las variables presentan cambios lentos y no necesita de acción inmediata.

• Monitoreo en tiempo real

Esto da como resultado que las condiciones de la planta pueden ser monitoreadas en tiempo real a través de la adquisición continua y constante de datos, que luego pueden ser enviados de vuelta para el análisis. Las alteraciones detectadas por los sensores son visibles de manera inmediata, lo que posibilita una rapidez de actuación frente a las modificaciones ambientales.

Este método es popular en sistemas hidropónicos ya que calefacción, pH y nivel de agua son ejemplos de parámetros que pueden cambiar a lo largo del uso del sistema.

• Monitoreo basado en eventos

En este tipo de monitoreo la transmisión de información ocurre únicamente cuando una variable supera un valor previamente establecido o cuando se presenta una condición específica. De esta manera se reduce el tráfico de datos y el consumo de recursos de comunicación.

Aplicación en el presente proyecto

En el presente proyecto se implementa un sistema de monitoreo remoto en tiempo real del nodo IoT que está basado en el microcontrolador ESP32 que debe adquirir de manera continua las variables temperatura, humedad, pH y nivel de agua. Los datos procesados son

enviados por medio de protocolos IoT a las aplicaciones hechas para Windows y Android que dan la posibilidad de controlar el estado de la planta hidropónica desde cualquier sitio con conexión a Internet.

2.7.2 Automatización en sistemas hidropónicos

La automatización en sistemas hidropónicos consiste en utilizar sensores, dispositivos de control y algoritmos de procesamiento para ejecutar acciones sobre el cultivo sin requerir intervención manual constante. Este método permite hacer que las condiciones ambientales se mantengan dentro de un rango apropiado, maximizando la eficiencia del agua, energía y recursos disponibles.

De acuerdo con el nivel de intervención en el sistema, la automatización puede ser:

- **Automatización manual asistida**

El sistema se limita a controlar las variables del cultivo y a alertar al usuario, que es el encargado de tomar acciones.

Este tipo de automatización requiere la intervención activa del operador

- **Automatización semiautomática**

Integra la supervisión automática con la intervención del usuario. Cuando se activa una condición específica, el sistema genera recomendaciones o alertas, la decisión para ejecutar o no la acción recae en el operador.

Esta metodología da lugar a una operación de cultivo mucho más flexible.

• Automatización automática

Actúa con autonomía, realizando de manera automática, a partir de la información captada por sensores, las medidas de control programadas con anterioridad sin necesidad de intervención humana.

Las acciones más habituales son las siguientes:

- Bombeo de agua.
- Riego por goteo.
- Controlar los ventiladores.
- Monitorear el nivel del tanque.
- Monitorización de los parámetros ambientales.
- Utiliza en el presente proyecto

El prototipo diseñado realiza un esquema de automatización directa, donde el microcontrolador ESP32 está procesando constantemente la información de los sensores y tomando acciones de control para el sistema hidropónico, para tal fin. Este funcionamiento permite mantener condiciones adecuadas para el cultivo y disminuir la necesidad de intervención manual durante la operación.

2.8 Tendencias Actuales del IoT Aplicado a la Agricultura

2.8.1 Edge Computing y procesamiento distribuido.

El Edge Computing es un paradigma de procesamiento distribuido que permite ejecutar parte del análisis de la información cerca del lugar donde se generan los datos. En lugar de enviar toda la información a un servidor externo para su procesamiento, el dispositivo IoT realiza un procesamiento inicial de forma local, reduciendo la latencia, el tráfico de red y el consumo de ancho de banda.

En el presente proyecto, el Esp32 actúa como un nodo inteligente al adquirir las variables ambientales, realizar un procesamiento inicial y transmitir únicamente la información necesaria hacia las aplicaciones de monitoreo, como se detalla en la **Figura 12** mediante un diagrama. Las principales funciones del Edge Computing en el sistema son:

- Adquirir las variables ambientales mediante los sensores.
- Ejecutar el procesamiento inicial de la información en el Esp32.
- Reducir la cantidad de datos transmitidos por la red.
- Disminuir la latencia en la comunicación.
- Optimizar el consumo de ancho de banda.
- Mantener el funcionamiento del sistema aun cuando existan interrupciones temporales en la conectividad.

En el sistema desarrollado, el Esp32 ejecuta estas funciones antes de enviar la información a las aplicaciones de monitoreo, mejorando la eficiencia de la arquitectura IoT.

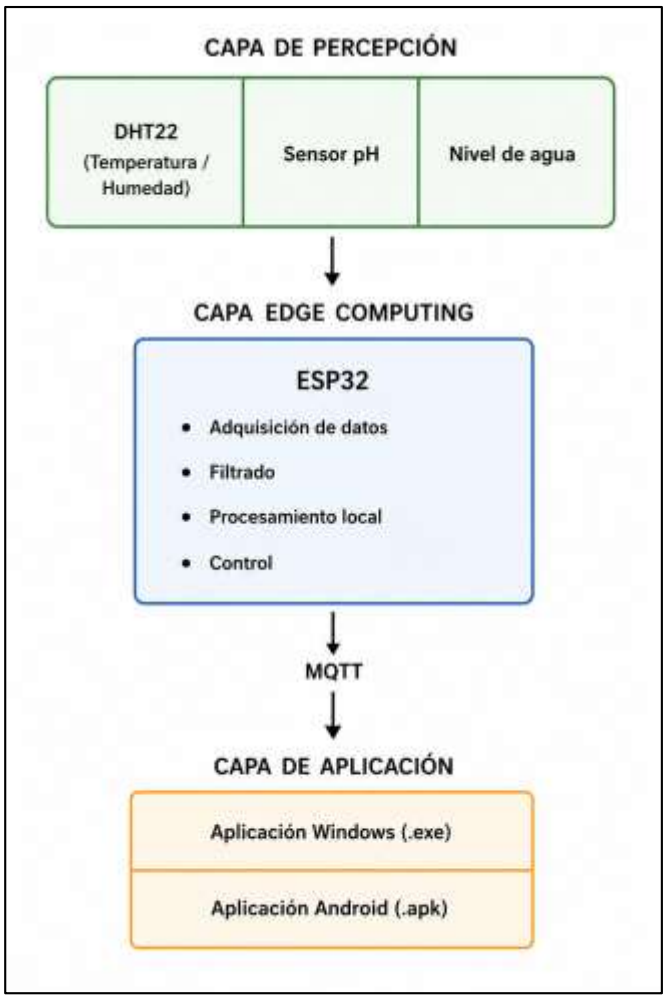


Figura 12: Diagrama de arquitectura IoT en capas.

CAPÍTULO III

3. DESARROLLO DE LA PROPUESTA.

3.1. Arquitectura del Sistema IoT

La propuesta se fundamenta en una estructura de tres capas, tal como se muestra en la imagen de la **Figura 13**, diseñada para optimizar el monitoreo y control de precisión en cultivos hidropónicos.

Figura 13: Arquitectura general del sistema IoT propuesto.

3.1.1 Arquitectura General del Sistema.

La arquitectura general del sistema fue concebida bajo un enfoque modular, permitiendo separar las funciones de adquisición de datos, procesamiento, comunicaciones, inteligencia artificial y supervisión en bloques independientes pero interconectados. Este diseño facilita el mantenimiento, la escalabilidad y la incorporación de nuevos dispositivos sin modificar la estructura principal del sistema.

El primer bloque corresponde a la adquisición de datos, conformado por los sensores encargados de medir las variables ambientales del cultivo hidropónico, entre ellas la temperatura, humedad relativa, nivel de agua y potencial de hidrógeno (pH). Estos sensores realizan mediciones periódicas que son enviadas al microcontrolador para su procesamiento.

El segundo bloque está constituido por el ESP32, que actúa como nodo inteligente dentro de la arquitectura IoT. Este dispositivo recibe la información proveniente de los sensores, ejecuta el procesamiento inicial de los datos, administra la lógica de control del sistema y publica la información mediante el protocolo MQTT hacia el broker Mosquitto. Además, recibe comandos provenientes de las aplicaciones cliente para el accionamiento de los actuadores cuando el sistema opera en modo manual.

El tercer bloque es la infraestructura de las comunicaciones con red Wi-Fi y el broker MQTT Mosquitto que es el servidor de mensajes. Este bloque es el núcleo en el intercambio de información con todos los dispositivos conectados, y se encarga de distribuir los mensajes publicados por ESP32 hacia las aplicaciones de supervisión.

El cuarto bloque corresponde a la unidad de procesamiento y está basado en una aplicación desarrollada en Python. La aplicación toma los datos transmitidos a partir de un broker MQTT, procesa la información.

Finalmente, bloque cinco representa al cliente, tanto escritorio para Windows, como móvil para Android. Ambas apps dan al usuario la posibilidad de monitorear el estado del cultivo, leer variables ambientales y enviar comandos al ESP32 cuando es necesaria la operación manual del sistema.

La modularización de estos bloques permite una arquitectura flexible, distribuida y escalable, que es una característica indispensable para un sistema IoT, el cual es común utilizarlo para aplicaciones de monitoreo y control.

3.1.2 Arquitectura de Comunicaciones.

Una parte clave en la propuesta es la infraestructura de comunicaciones, basada en una arquitectura IoT con el modelo Publicador/Suscriptor (Publish/Subscribe) mediante el protocolo MQTT sobre la pila TCP/IP. En esta arquitectura El ESP32 es el publicador porque los datos captados por los sensores los publica en el broker Mosquitto, mientras que este último se encarga de distribuir dicha información a los clientes aplicaciones suscritas, indistintamente si son Windows o Android. Asimismo, los comandos enviados por el usuario son retransmitidos por el broker hacia el ESP32, permitiendo una comunicación bidireccional para el monitoreo y control del cultivo.

El intercambio de información se realiza mediante MQTT sobre la red Wi-Fi bajo el estándar IEEE 802.11n, proporcionando una comunicación eficiente con bajo consumo de ancho de banda, baja latencia y desacoplamiento entre dispositivos. Además, la arquitectura presenta un diseño escalable, ya que permite incorporar nuevos dispositivos, aplicaciones o servicios mediante la suscripción a los tópicos MQTT existentes, sin necesidad de realizar modificaciones significativas en la infraestructura de comunicaciones, véase la **Figura 14** del siguiente esquema.

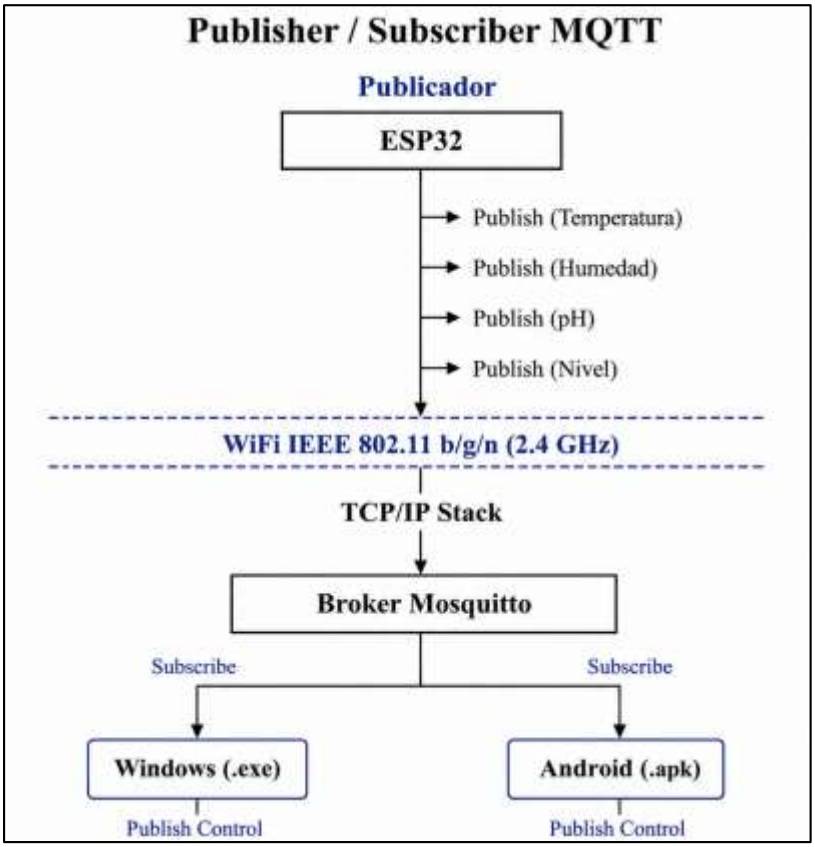


Figura 14: Arquitectura de comunicaciones basados en MQTT.

3.1.3 Flujo de Información del Sistema.

El flujo de información del sistema inicia con la adquisición de datos por parte de los sensores instalados en el cultivo hidropónico. El ESP32 recibe estas mediciones, realiza un procesamiento inicial y organiza la información para su transmisión mediante el protocolo MQTT a través de la red Wi-Fi.

Posteriormente, la información es publicada en el broker Mosquitto, el cual la distribuye a las aplicaciones cliente bajo el Modelo de Comunicación Publicador/Suscriptor. Tanto la aplicación de escritorio, desarrollada en Python y distribuida como un archivo ejecutable (.exe) para Windows, como la aplicación Android reciben y presentan las variables del cultivo en tiempo real, permitiendo al usuario supervisar el estado del sistema desde cualquiera de las dos plataformas.

Cuando el usuario envía un comando de control desde cualquiera de las aplicaciones, este es transmitido al broker MQTT y posteriormente al ESP32, que acciona los dispositivos correspondientes, como la bomba o el ventilador. Este flujo bidireccional garantiza una comunicación en tiempo real, sincronización entre los dispositivos y una arquitectura IoT escalable basada en el modelo Publicador/Suscriptor.

3.2. Hardware y Programación del Nodo Sensor

Con el objetivo de garantizar la eficiencia del microcontrolador ESP32, se implementó una programación de tipo modular mediante el uso de archivos de cabecera (.h) e implementación (.cpp). Esta arquitectura permite realizar una separación clara entre la gestión de la red inalámbrica Wi-Fi y el protocolo MQTT, de la lógica dedicada a la adquisición de datos de los sensores y el control de los actuadores finales.

3.2.1. Instrumentación y Captura de Datos.

El nodo sensor recolecta variables críticas mediante los siguientes componentes:

- **Sensor DHT22 (GPIO 4):** Temperatura y humedad.
- **Sensor de pH (GPIO 34):** ADC de 12 bits.
- **Sensor de Nivel (GPIO 35)**

Diagrama (**Figura 15**).

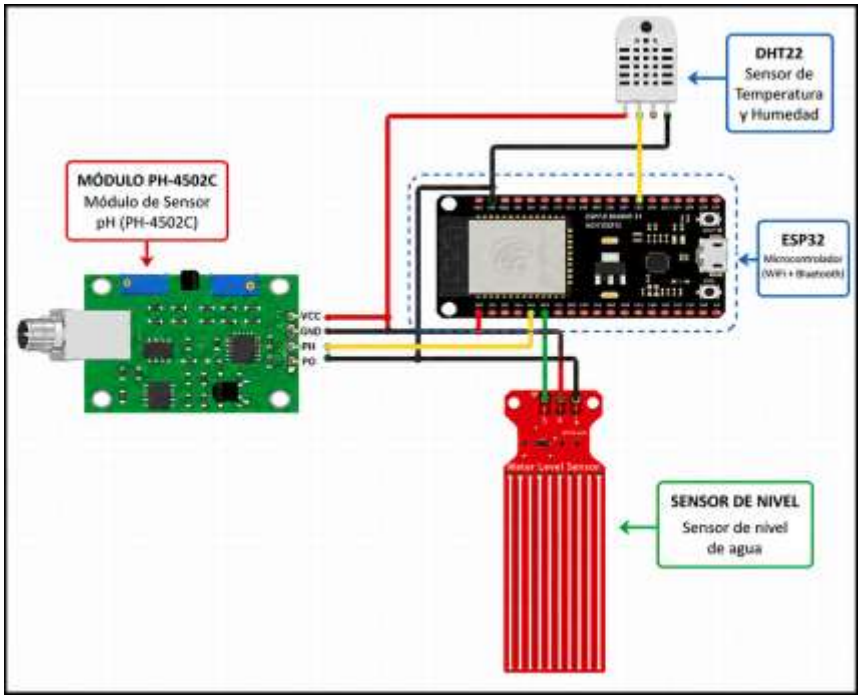


Figura 15: Diagrama esquemático de captura de datos usando proteus.

3.2.2. Control de Potencia y Lógica de Histéresis

El control de actuadores puede verse en la figura 16, en específico el ventilador está conectado al pin GPIO 18 y la electrobomba al GPIO 19, la gestión de los actuadores se realiza con una etapa de potencia basada en módulos relé. Para control térmico se desarrolló un algoritmo de histéresis (On-Off con banda muerta) en el firmware del ESP32. Esta táctica es esencial para evitar el fenómeno conocido como saltos o chattering, que se presenta cuando la variable medida realiza variaciones mínimas en un único umbral, lo que ocasiona desgaste prematuro de los contactos del relé electrome-cánico.

El ajuste de los umbrales de histéresis se fijó teniendo en cuenta las condiciones climatológicas de la provincia de Santa Elena, donde se realizó la validación experimental del sistema. Estos valores fueron establecidos para la estación caliente o lluviosa (Diciembre a Mayo), la cual está caracterizada por temperaturas altas. En la temporada seca (de junio a noviembre) dichos umbrales pueden ser reajustados para minimizar el desperdicio del sistema de ventilación. Por lo tanto, la lógica de control implementada es flexible y puede ser ajustada

a las condiciones climáticas del entorno de la aplicación, lo que permite un uso más eficiente de los actuadores.

La lógica de funcionamiento es definida con las siguientes restricciones:

Umbral de Activación (Límite Superior): Si este umbral es alcanzado o superado el procesador envía una señal HIGH al relé y el extractor es prendido para que empiece a extraer calor por convección forzada en el lugar donde se están creciendo las plantas.

Umbral de Desactivación (Límite Inferior): El ventilador permanece encendido hasta que la temperatura reportada por el sensor DHT22 sea menor a 31.0°C para mantener un buen balance entre estabilidad térmica y eficiencia del mismo. En este punto la señal es fijada a bajo (LOW) desactivando el actuador.

Diferencial térmico (Banda Muerta): Se utiliza una banda/intervalo de 3.0 °C para el ventilador con relación a sus puntos de inicio y parada. Esta política reduce la frecuencia de conmutación del relé, aumenta la vida útil del actuador y proporciona condiciones térmicas más estables para cultivo hidropónico.

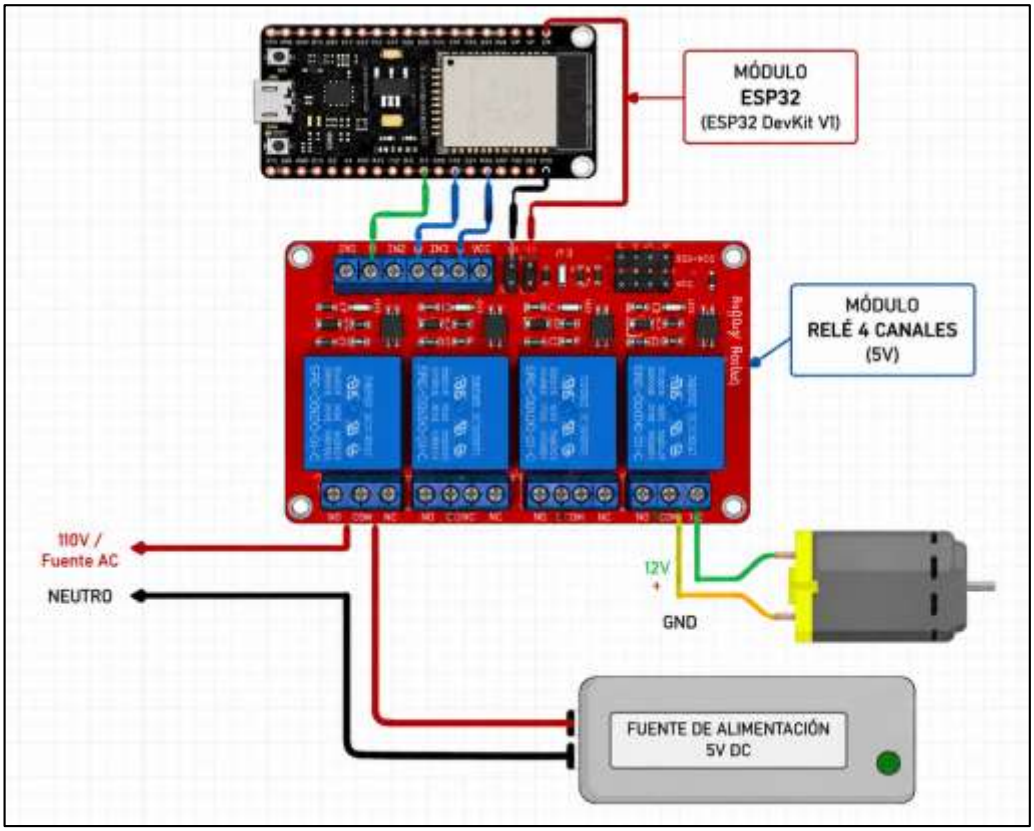


Figura 16: Diagrama esquemático de actuadores.

3.2.3 Integración del Hardware.

Terminados los módulos hardware de adquisición, procesamiento y control de potencia, se procedió a integrar físicamente el hardware en un solo prototipo. La ubicación de los componentes está orientada a un fácil mantenimiento, minimizando el cableado y manteniendo los elementos electrónicos organizados adecuadamente dentro de la caja del sistema.

Se estableció el ESP32 como la unidad central del nodo sensor, y se conectó a los sensores de temperatura y humedad, pH y nivel de agua, y a los módulos de relé para el control de la bomba y el ventilador. Todos los componentes se energizaron con una sola fuente para garantizar la integridad del sistema.

La integración del hardware permitió verificar el correcto funcionamiento de

comunicación entre todos los dispositivos, tanto en la adquisición de datos, como en la realización de acciones de control en tiempo-real. La disposición física del prototipo también permite la posibilidad de futuras expansiones al sistema, incorporando nuevos sensores y/o actuadores sin necesidad de modificar la arquitectura principal. Desde la perspectiva de la ingeniería en comunicaciones y telecomunicaciones, la integración hardware es el sustrato físico sobre el que se apoya la infraestructura de comunicaciones IoT para posibilitar que la información generada en el entorno del cultivo sea transmitida, procesada y empleada en la toma de decisiones a distancia.

3.3 Infraestructura de Comunicaciones

3.3.1 Protocolo de Comunicación.

La transferencia de datos se realiza mediante el protocolo MQTT, través del cual el ESP32 publica un objeto JSON cada 3 segundos.

Tabla 2: Resumen de configuración técnica

Componente	**Tecnología / Pin**	**Especificación**
Microcontrolador	ESP32 DevKit V1	Arquitectura Modular
Control de Bomba	GPIO 19	pH < 5.5 o > 6.5
Seguridad Nivel	GPIO 35	Umbral > 350

3.3.2 Configuración de la Red WI-FI.

El nodo sensor basado en el microcontrolador ESP32 fue configurado para establecer la conexión inalámbrica con la red Wi-Fi disponible antes de iniciar la comunicación mediante MQTT. Para aumentar la disponibilidad del sistema, se introdujo un mecanismo de búsqueda automática de redes con las credenciales de varias redes inalámbricas autorizadas almacenadas previamente. Durante la inicio, el ESP32 ejecuta un escaneo de las redes disponibles con la función "**`Wi-Fi.scanNetworks(),`**" y compara los SSID encontrados con los almacenados en la lista de redes configuradas. Cuando detecta un encuentro, comienza el proceso de autentificación **`Wi-Fi.begin()`** y monitorea constantemente el

estado de la conexión hasta recibir una dirección IP válida. Una vez establecida la conexión, el dispositivo muestra en el Monitor Serial la dirección IP asignada por el servidor DHCP del router, confirmando que el nodo sensor se encuentra preparado para iniciar la comunicación con el broker MQTT , como se muestra en la **Figura 17** y **Figura 18.**

C++

```
// 🔐 Credenciales WiFi
const char* redes[] = {"XTRIM_DELGADO", "LIAMYENCE", "prueba1"};
const char* claves[] = {"0928351600", "68928223YE", "1234567890"};
const int totalRedes = 3;
```

y la función:

C++

```
void iniciarWiFi() {
   Serial.println("\nEscaneando redes...");
   int n = WiFi.scanNetworks();
   ...
   WiFi.begin(redes[j], claves[j]);
   ...
   if (WiFi.status() == WL_CONNECTED) {
       Serial.println("\n¡Conectado!");
       Serial.print("IP: ");
       Serial.println(WiFi.localIP());
   }
}
```

Figura 17: Fragmento del código utilizado para configuración de la conexión Wi-Fi del ESP32

Figura 18: Prueba de conectividad Wi-Fi en monitor serial

3.4 Desarrollo del Firmware en Arduino IDE.

La lógica de hardware y de control eran desarrolladas con el microcontrolador ESP32 en el entorno Arduino IDE. La arquitectura utilizada es modular, el código se divide en archivos de cabecera (.h) y de implementación (.cpp), esto permite manejar eficientemente los recursos del sistema y ayuda en el debugging del sistema.

3.4.1 Estructura Modular del Código.

El firmware se organiza en módulos que interactúan de forma coordinada en el ciclo de ejecución principal:

Módulo de Sensores (sensores.cpp): Centraliza la lectura del sensor DHT22 para temperatura y humedad. También controla la entrada analógica para el sensor pH y detección de nivel de agua, que se trata como es un umbral de activación en 350 unidades de ADC (lectura analógica).

Módulo de Actuadores (control.cpp): Se definen las funciones para los pines GPIO 18 y 19. Este módulo garantiza que los periféricos terminan en un estado seguro a la parada del sistema.

Módulo de Comunicación (mqtt.cpp): Gestiona la conexión a la red local y la vinculación con el broker. Se encarga de empaquetar los parámetros del sistema en una cadena de texto con formato JSON para su transmisión inalámbrica.

3.4.2 Algoritmo de Control y Toma de Decisiones.

El programa principal coordina la lógica de control basada en las necesidades del cultivo:

Control Térmico por Histéresis: Para la gestión del ventilador, se configuró un umbral de encendido a los 34.0 °C y un umbral de apagado a los 31.0 °C.

Regulación de pH: El sistema monitorea constantemente el nivel de acidez. Si el valor de pH se encuentra fuera del rango ideal entre 5.5 y 6.5, se activa la bomba de corrección, siempre y cuando el sensor de nivel confirme la presencia de líquido.

3.4.3 Modo de control Manual

Complementando la lógica automatizada, se integró una función de control manual gestionada desde el script de Python. Esta funcionalidad permite al usuario forzar el estado de los actuadores (encendido/apagado de bomba y ventilador) de manera remota a través del protocolo MQTT, priorizando la intervención humana sobre los umbrales automáticos en situaciones de mantenimiento o pruebas de hardware.

3.5 Procesamiento de Datos y Centro de Control de Python.

Para la gestión centralizada y supervisión del sistema hidropónico, se desarrolló una aplicación robusta utilizando el lenguaje de programación Python 3. El software actúa como el nodo de procesamiento superior, permitiendo la visualización de métricas en tiempo real y

la toma de decisiones basada en inteligencia artificial. Se integró el *framework* KivyMD para la interfaz gráfica y la librería Paho-MQTT para la gestión de mensajería asíncrona, como se describe en la **Figura 19.**

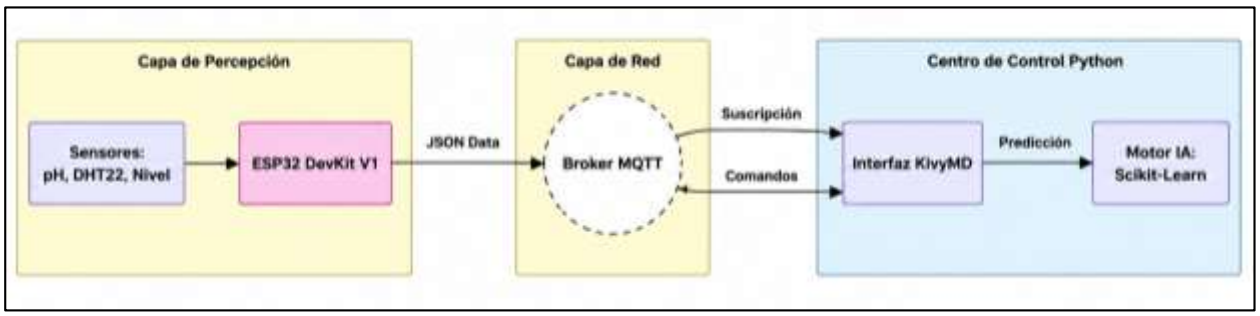


Figura 19: Diagrama de procesamiento de datos y control de Python [29].

3.5.1 Gestión de Hilos (Threading).

Dado que la aplicación debe procesar un flujo constante de datos sin comprometer la fluidez de la interfaz de usuario, se implementó una arquitectura basada en ***multithreading*.** Mediante el uso de la librería **`threading`**, el cliente MQTT opera en un hilo secundario independiente, véase **Figura 20.**

Esta separación es crítica para la estabilidad del sistema: mientras el hilo secundario mantiene la escucha activa de mensajes, el hilo principal (*Main Thread*) se dedica exclusivamente a la renderización visual. La sincronización se realiza mediante **`Clock.schedule_once`**, evitando congelamientos en la pantalla.

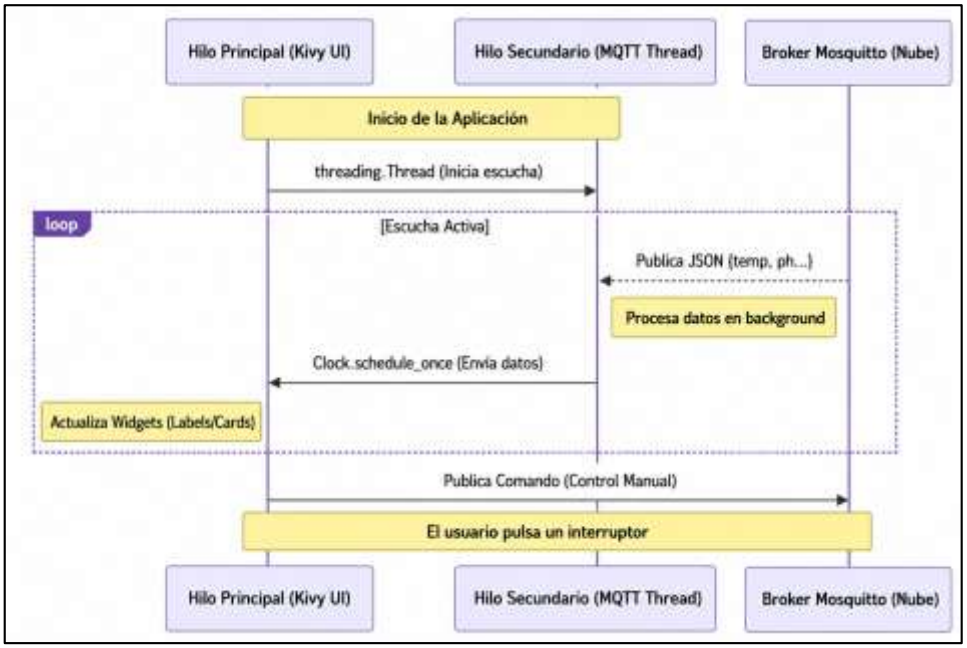


Figura 20: Gestión de hilos y flujo MQTT [30].

3.6 Compilación y Empaquetado con Buildozer en Google Colab

El proceso de conversión a binario ejecutable (.apk) se llevó a cabo con la herramienta Buildozer en el entorno de nube de Google Colab. Esta decisión de tecnología proporcionó un entorno Linux para satisfacer las necesidades de compilación de aplicaciones desarrolladas con kivy.

Las ventajas principales de usar Google Colab fueron:

Gestión de Dependencias:. Google Colab hizo más fácil instalar el SDK y NDK de Android, así como las librerías para compilar la aplicación con Buildozer.

Configuración del archivo buildozer.spec: Para definir los requisitos de la app, como los permisos de acceso a la red (INTERNET), y las dependencias necesarias como Kivy, KivyMD, Paho-MQTT y Pandas.

Optimización del Entorno: La construcción en la nube ayudó a contener al intérprete de Python, junto con las bibliotecas y los archivos usados por la aplicación, lo que permite que el resultado sea un solo archivo installable (.apk), que puede correr de manera

independiente en dispositivos Android, sin requerir configuraciones adicionales durante su instalación.

3.7 Generación del Archivo APK.

Para garantizar la movilidad del operador y la supervisión remota del sistema, el software fue portado a la plataforma Android. Este proceso de ingeniería consistió en transformar los scripts de Python en un paquete ejecutable independiente que integra la interfaz gráfica.

3.7.1 Configuración del Entorno y Dependencias.

El proceso de compilación se realizó en el entorno de nube Google Colab. Inicialmente, se configuró el sistema operativo Linux del contenedor mediante la instalación de dependencias críticas de compilación, herramientas de Python y librerías científicas necesarias para el funcionamiento de Scikit-learn en dispositivos móviles, véase la **Figura 21**.

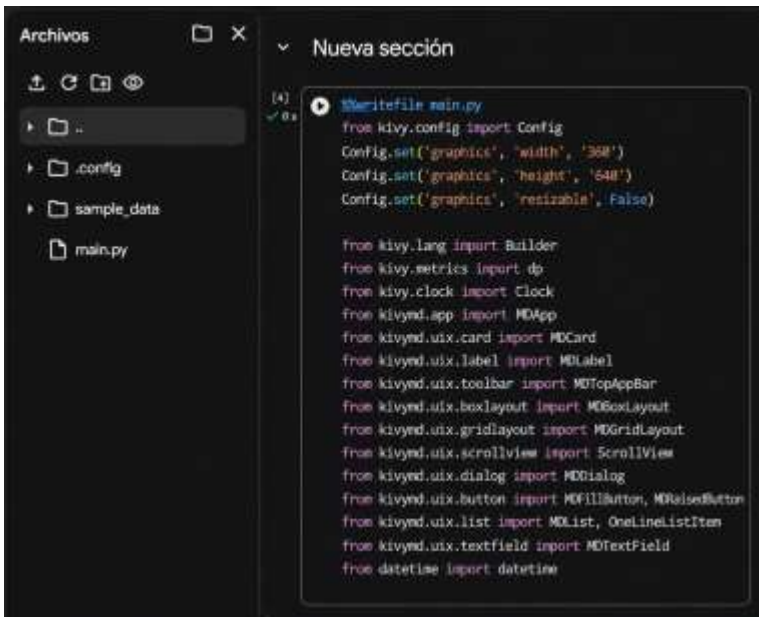


Figura 21: Instalación de dependencias de Google Colab.

3.7.2 Estructura del Proyecto y Archivo Principal.

Se procedió a la creación del archivo main.py dentro del entorno de trabajo como se observa en la **Figura 22**. Este script centraliza la lógica de la interfaz desarrollada en KivyMD, gestionando simultáneamente la recepción de tramas JSON vía MQTT y el procesamiento de los datos a través del modelo de inteligencia artificial previamente entrenado.

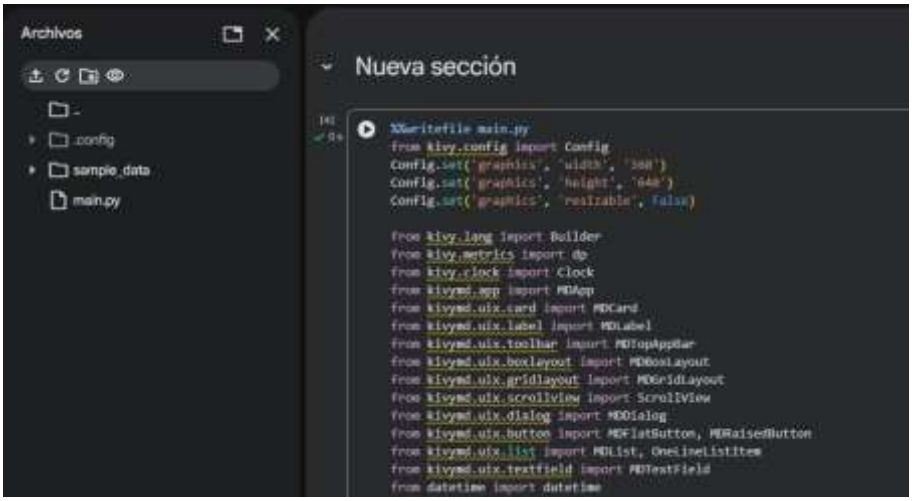


Figura 22: Carga del script principal del sistema.

3.7.3 Inicialización y Configuración de Buildozer

Para automatizar el empaquetado, se utilizó la herramienta Buildozer. El primer paso consistió en la generación del archivo de especificaciones buildozer.spec, el cual define cómo debe construirse la aplicación, véase **Figura 23.**

Figura 23: Inicialización de la herramienta buildozer

Posteriormente, se editaron los parámetros del archivo buildozer.spec para declarar formalmente el nombre de la aplicación (Hidroponico AI), los permisos de red requeridos y la inclusión de todas las librerías necesarias (KivyMD, Paho-MQTT, Scikit-learn, NumPy, entre otras), véase **Figura 24.**

Figura 24: Configuración del archivo buildozer. spec.

3.7.4 Compilación y Empaquetado Final.

Finalmente, se ejecutó el proceso de compilación mediante el comando de depuración de Android se evidencia en la **Figura 25**. Durante esta etapa, Buildozer descarga el NDK y SDK de Android y genera el archivo .apk instalable.

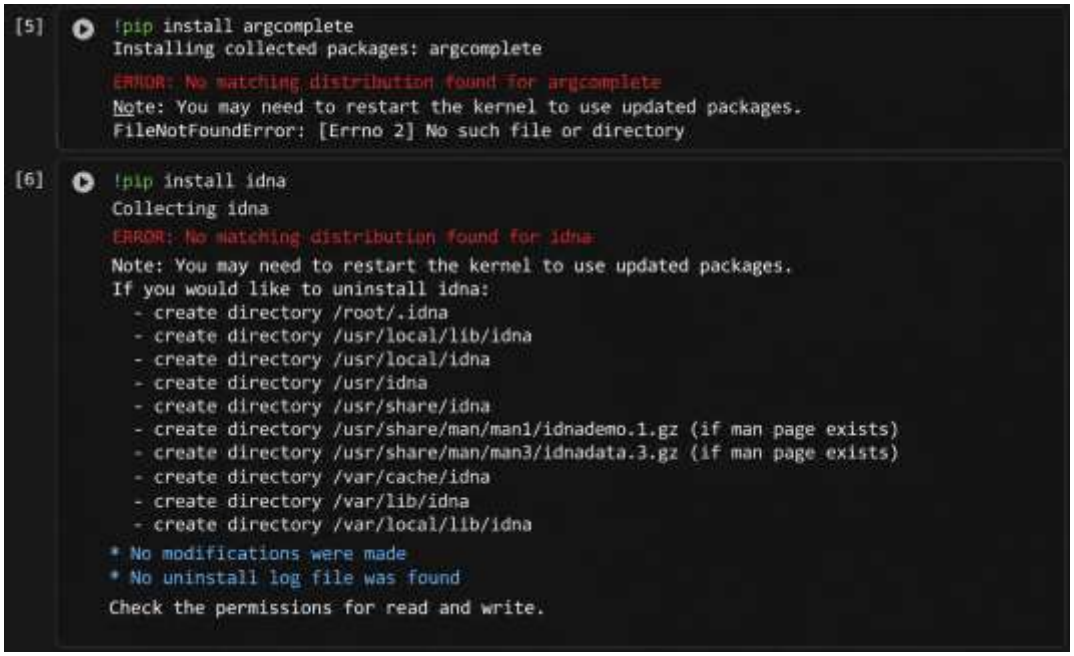


Figura 25: Ejecución del empaquetado final

3.7.5 Lógica y Control Automático.

El sistema basa su funcionamiento principal en un control por umbrales predefinidos que se activa de manera automática e inmediata al iniciar la aplicación. El software evalúa constantemente las variables de temperatura, humedad y pH recibidas vía MQTT; si los valores exceden los rangos óptimos establecidos, el sistema despacha comandos hacia los actuadores (bomba y ventilador) de forma autónoma. Esta lógica de control permanece operativa de manera predeterminada, garantizando la estabilidad del cultivo sin requerir intervención inicial del usuario, como se muestra en la **Figura 26**.

Figura 26: Ilustración de control automático en interfaz.

3.7.6 Interfaz e Supervisión y Control Manual.

La aplicación que se observa en la **Figura 27** permite una interacción bidireccional entre el usuario y el microcontrolador. Se deserializan las tramas JSON recibidas para el monitoreo de parámetros y se incluyen interruptores lógicos (*MDSwitch*) para el control manual. Al activar este modo, el usuario tiene prioridad jerárquica sobre la lógica automática del sistema.

Figura 27: Interfaz de sensores activos.

3.7.9 Visualización y Diagnóstico en Tiempo Real

La interfaz de usuario centraliza las métricas de los sensores y el veredicto de la red neuronal. El software implementa una lógica de alertas visuales donde los parámetros críticos, como el nivel de pH, cambian de color para facilitar el diagnóstico rápido por parte del operador (ver **Figura 28**).

Figura 28: Panel principal de monitoreo

3.8 Integración Física del Prototipo.

3.8.1 Diseño de la Caja Box

Con el propósito de proteger e integrar los componentes electrónicos del sistema, se diseñó una caja (Box) mediante el software **SolidWorks**. Para ello, se tomaron las dimensiones del ESP32 y de los demás componentes utilizando un calibrador Vernier, como se puede ver en la **Figura 29**, lo que permitió desarrollar un modelo tridimensional con los espacios necesarios para el montaje de los dispositivos, el paso del cableado y los conectores, que se pueden ver en las siguientes figuras (**Fig. 30, 31 y 32**). Posteriormente, el diseño fue fabricado mediante impresión 3D para su implementación en el prototipo.

Figura 29: Uso del calibrador VERNIER para tomar medidas correctamente

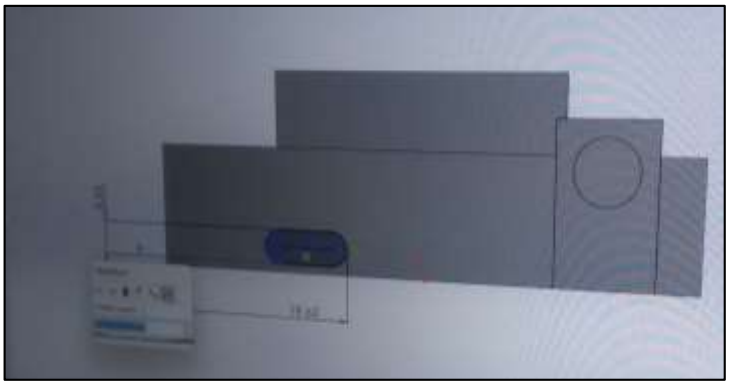


Figura 30: Creación del diseño del box según las medidas

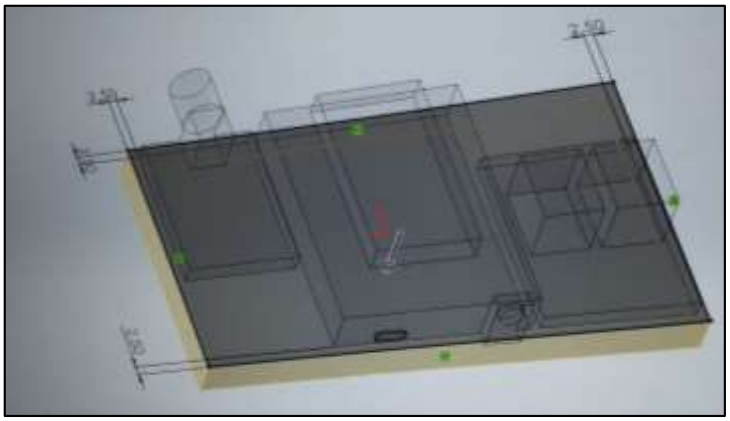


Figura 31: Inserción de Bloques según distribución.

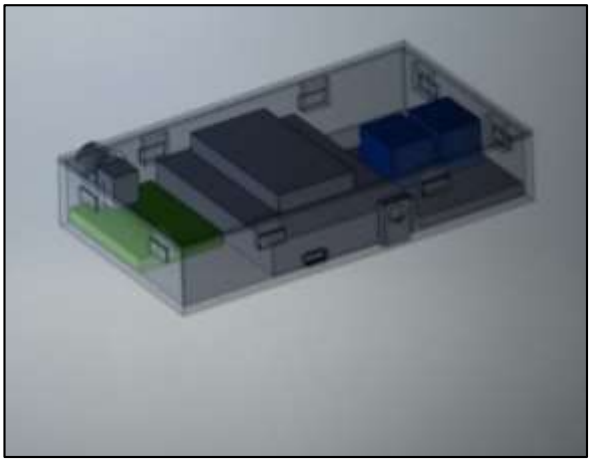


Figura 32: Diseño final del box incluida la tapa.

3.8.2 Distribución Interna de Componentes.

Una vez fabricada la caja, se procedió a la instalación y organización de los componentes electrónicos, ubicando el ESP32, el módulo de relés, la fuente de alimentación y el cableado de forma ordenada para optimizar el espacio disponible y facilitar las conexiones, el mantenimiento y el correcto funcionamiento del sistema, que se pueden visualizar en las siguientes Imágenes (**Fig. 33, 34, 35** y **36**).

Figura 33: Colocación del módulo relé, ESP32 y módulo sensor de pH.

Figura 34: Instalación acorde la distribución de los dispositivos.

Figura 35: Cableado de los de los módulos y el ESP32.

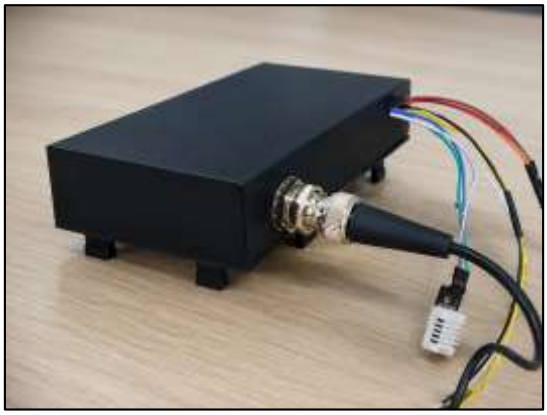


Figura 36: Encapsulado de los componentes electrónicos del sistema IoT

3.8.3 Ensamblaje del Sistema.

Finalizada la instalación de los componentes internos, se realizó el ensamblaje completo de la caja y su integración con el sistema hidropónico NFT. Se verificó el correcto ajuste de la estructura, la salida de los conductores y la conexión de los sensores y actuadores, obteniendo un módulo compacto que brinda protección a la electrónica durante su operación, que se visualiza en las siguientes Figuras (**Fig. 37, 38** y **39**).

Figura 37: Vista lateral de las entradas de alimentación de energía del prototipo.

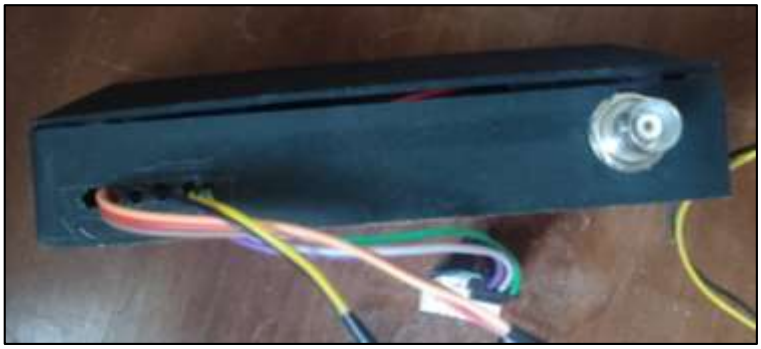


Figura 38: Vista frontal de las salidas de los sensores, actuador por cableado y por módulo de pH.

Figura 39: Estructura donde se implementará las pruebas del prototipo.

3.8.4 Fotografías del prototipo terminado

Finalmente, se integraron todos los elementos del sistema hidropónico, incluyendo la estructura NFT, el depósito de la solución nutritiva, la bomba de recirculación, los sensores y la caja de control. Las fotografías muestran el prototipo completamente ensamblado y en funcionamiento, evidenciando la integración del sistema IoT con el cultivo hidropónico desarrollado, como se muestran en la **Figura 40, 41, 42, 43** y **44** respectivamente.

Figura 40: Prueba del funcionamiento del prototipo.

Figura 41: Previa instalación y colocaciones sensores

Figura 42: Adecuación del lugar para la ubicación del prototipo.

Figura 43: Visualización final de cómo quedaría conectado.

Figura 44: Incorporación de lechuguines de 2 Semanas, visualizando óptimo crecimiento.

CAPÍTULO IV

4. RESULTADOS Y ANÁLISIS

4.1. Validación del sistema

La validación del sistema desarrollado se realizó con el propósito de comprobar el cumplimiento de los objetivos planteados durante la investigación y verificar el correcto funcionamiento de cada uno de los módulos que integran el prototipo IoT para el monitoreo y control de un cultivo hidropónico. En contraste con una validación enfocada a procesos de certificación industrial, ésta evaluación fue funcional y experimental, orientándose en mostrar la integración correcta entre el hardware, el software embebido, la infraestructura de comunicaciones y las aplicaciones desarrolladas para la supervisión remota.

La validación consistió en pruebas unitarias y de integración en los distintos elementos del sistema, sobre la adquisición de las variables ambientales, el procesamiento local con el ESP32, envío de datos usando el protocolo MQTT sobre redes Wi-Fi, con el broker Mosquitto, y conexión con aplicaciones para Windows y Android. También se evaluó el rendimiento de los actuadores del controlador automático del cultivo, probando la respuesta del sistema ante diversas situaciones de operación.

Los resultados obtenidos permitieron demostrar la estabilidad durante las pruebas experimentales que fueron en tiempo real para el monitoreo y control del cultivo. Por lo tanto, la solución proporciona dos vías de comunicación: una destinada a establecer monitoreo y otra al control en tiempo real desde las aplicaciones informáticas, cumpliendo así con los objetivos planteados en el desarrollo del prototipo.

4.1.1. Validación funcional del hardware

Como primera etapa del proceso de validación se verificó el funcionamiento individual y conjunto de todos los componentes físicos que conforman el sistema. Esta

evaluación permitió comprobar la correcta integración entre el microcontrolador ESP32, los sensores DHT22, pH y nivel, el módulo de relés, la bomba de recirculación y el ventilador.

Durante las pruebas se verificó el reconocimiento de los sensores, la estabilidad de las lecturas y el accionamiento correcto de los relés. También se comprobó la adecuada alimentación del sistema, evidenciándose un funcionamiento estable durante las pruebas.

4.1.2. Validación del software embebido

Se validó el firmware desarrollado para el ESP32, comprobando la adquisición de datos, el procesamiento de las variables ambientales, la publicación y recepción de mensajes MQTT y la ejecución de los algoritmos de control. El sistema ejecutó correctamente la inicialización, la lectura de sensores y la reconexión automática a la red y al broker MQTT.

4.1.3. Validación de la conectividad IoT

Las pruebas confirmaron que el ESP32 publicó correctamente las variables ambientales hacia el broker Mosquitto mediante MQTT. Las aplicaciones para Windows y Android recibieron la información en tiempo real y enviaron comandos de control que fueron ejecutados correctamente por el ESP32.

4.1.4. Validación de los sensores DHT22, ph y nivel

Se comprobó la estabilidad de las lecturas de temperatura, humedad, pH y nivel de agua. Las mediciones fueron consistentes durante las pruebas, permitiendo el correcto funcionamiento del sistema de monitoreo y control.

4.1.5. Validación del sistema control bomba-ventilador

Se verificó el accionamiento automático y manual de la bomba de agua y del ventilador mediante los relés controlados por el ESP32. Los actuadores respondieron correctamente tanto a las condiciones programadas como a los comandos enviados desde las aplicaciones.

4.1.6. Resultados experimentales

Las pruebas realizadas en el cultivo hidropónico demostraron la correcta adquisición de datos, la transmisión continua mediante MQTT y el funcionamiento coordinado entre sensores, actuadores y aplicaciones cliente. El sistema respondió de forma estable durante el monitoreo y control en tiempo real, tal como se aprecia en la Figura 45, en donde se muestra el comportamiento de los lechuguines de unos 5 días , así como las anomalías respectivas cuando no está controlado la temperatura y el nivel de agua.

Figura 45: Cultivo hidropónico de lechuguines de 5 días

4.1.7 Análisis de estabilidad del Sistema

Con el fin de probar la confiabilidad del sistema desarrollado, se hicieron corridas continuas monitoreando la estabilidad del enlace, la adquisición de datos y la circulación de los algoritmos de control en varias sesiones de su uso.

Los resultados demostraron que el ESP32 siempre mantuvo una comunicación estable con el broker Mosquitto a través de Wi-Fi, sin que se presentaran pérdidas considerables de mensajes MQTT durante las pruebas. También, las aplicaciones clientes recuperaron con éxito la información publicada por el microcontrolador, y los actuadores reaccionaron de forma oportuna a los comandos que fueron enviados desde el sistema de monitoreo, lo que

permitió la sincronización entre el hardware y las aplicaciones creadas. Durante las pruebas experimentales también se identificaron pequeñas variaciones asociadas a algunos sensores utilizados. En el caso del sensor de nivel de agua, el uso continuo durante las diferentes etapas de experimentación produjo un desgaste progresivo de sus pistas conductoras, ocasionando que en determinadas ocasiones la lectura presentara ligeras diferencias respecto al nivel real del depósito. Debido a ello, el sistema complementa esta información considerando el tiempo de funcionamiento de la bomba de recirculación, permitiendo estimar el comportamiento del nivel de agua y reducir el efecto de dichas variaciones.

De manera similar, el sensor de pH empleado corresponde a un módulo analógico de bajo costo, por lo que las mediciones pueden presentar pequeñas fluctuaciones y un tiempo de estabilización antes de alcanzar un valor definitivo. Tales desviaciones son típicas de este tipo de sensores y pueden causar un pequeño retraso durante la adquisición del dato, no obstante, no afectaron el comportamiento del sistema ni en su ejecución ni en la toma de decisiones diseñada en el prototipo.

En conjunto las pruebas realizadas conllevan al hecho del funcionamiento estable del sistema y sincrónico para aplicaciones de monitoreo y control de sustratos hidropónicos. Las variaciones observadas son más atribuidas a las características de los propios sensores que a fallos de la arquitectura IoT desarrollada, que realizó una comunicación persistente entre los nodos considerados a lo largo de toda la experimentación.

Tabla 3 seguimiento del cultivo hidropónico.

Día	Altura promedio (cm)	Temperatura (°C)	Humedad (%)	pH	Nivel de agua	Estado del sistema
0	4.5	27.8	69	6.2	Alto	Normal
3	5.8	28.0	70	6.1	Alto	Normal
6	7.3	27.6	71	6.0	Medio	Normal
9	9.4	27.9	70	6.2	Medio	Normal
12	11.2	28.1	69	6.1	Bajo	Bomba activada
14	13.1	27.8	70	6.1	Alto	Normal

Nota: los resultados muestran un crecimiento progresivo del cultivo bajo condiciones ambientales controladas. El sistema mantuvo monitoreo continuo de variables ambientales y activo automáticamente la bomba de recirculación cuando el nivel de agua descendió aun valor bajo.

La observación durante dos semanas posibilitó evaluar el comportamiento del cultivo hidropónico bajo el seguimiento continuo por el sistema IoT implementado. Las mediciones realizadas cada 3 días mostraron que las plantas de lechuga crecían de manera constante, sin síntomas de marchitez o estrés hídrico durante el periodo de evaluación, habiendo un aumento continuo en la altura promedio de las plantas.

Las variables ambientales registradas por el sistema (temperatura, humedad relativa), junto con el pH y el nivel de agua, se mantuvieron dentro de los límites definidos para el cultivo. Durante las pruebas, el Esp32 envió continuamente los datos a través del protocolo MQTT al broker Mosquitto, lo que hizo posible observar en tiempo real el estado del sistema mediante las aplicaciones creadas para Windows y Android, como se evidencia en las pruebas de muestra en las **Figuras 46, 47 y 48** en ese orden. El interruptor también permitio activar la bomba de recirculación cuando el nivel de agua descendió por debajo del nivel preestablecido, proporcionando al cultivo la solución nutritiva necesaria.

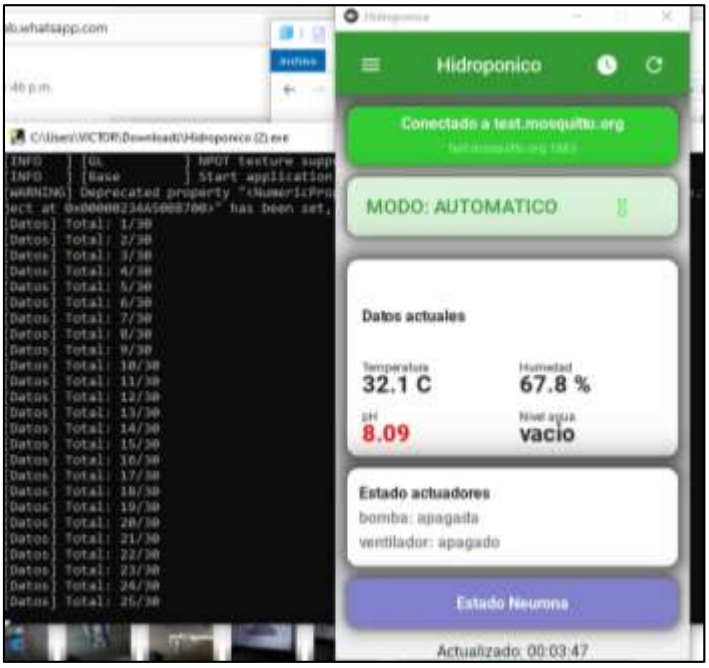


Figura 46: Envío de datos con muy poca latencia.

Figura 47: Control funcional ventilador apagado.

Figura 48: Conexión Wi-Fi excelente, broker conectado.

Aunque se observaron pequeñas fluctuaciones en las lecturas del sensor de nivel y ligeras oscilaciones en el sensor de pH, debido a las características propias de estos dispositivos de bajo costo, estas desviaciones no afectaron al crecimiento de las plantas ni al desempeño general del sistema. En conjunto, los resultados obtenidos evidencian que la arquitectura IoT implementada permitió mantener un monitoreo continuo y un control eficiente de las variables ambientales como se visualiza mediante la interfaz en la **Figura 49**, proporcionando condiciones adecuadas para el desarrollo del cultivo hidropónico durante el periodo experimental.

Figura 49: Activación de ventilación por temperatura elevada.

4.1.8. Análisis comparativo de un cultivo hidropónico convencional

Con el propósito de evaluar el desempeño del sistema desarrollado, se realizó un análisis comparativo entre el prototipo IoT implementado y el funcionamiento de un sistema hidropónico convencional. Esta comparación fue diseñada para estimar en el sistema la capacidad de monitorear continuamente-condiciones ambientales y ejecutar acciones de control automático en el medio ambiente durante el crecimiento de la planta de lechuga.

A diferencia de un sistema hidropónico convencional, donde el operador debe realizar inspecciones manuales y monitorear las variables, el sistema desarrollado adopta una arquitectura basada en Internet de las Cosas (IoT) para conseguir datos en tiempo real mediante sensores de temperatura, humedad, pH y nivel de agua que se conectan a un nodo

Esp32. La información recabada es transmitida utilizando el protocolo MQTT hacia el broker Mosquitto y luego es visualizada desde un cliente desarrollado para Windows y Android. Durante el período de prueba se constató que se podía monitorear el estado del cultivo en tiempo real, con lo cual se podían detectar tempranamente cambios en el nivel de agua o en el pH de la solución nutritiva. De igual forma, el sistema de control automático permitió la activación de la bomba de recirculación al momento que ésta fue requerida, manteniendo condiciones necesarias para el desarrollo de las plantas sin necesidad de una supervisión constante del operador.

Hay que tener en cuenta que la finalidad de esta comparación no fue demostrar un mayor crecimiento de las plantas frente a otros sistemas hidropónicos, sino confirmar la viabilidad de la arquitectura IoT desarrollada como herramienta para el monitoreo y control. Las diferencias encontradas estaban relacionadas con el proceso que era más automatizado y que disponía de información en tiempo real, además de la capacidad de ser accedido desde cualquier lugar para ver el estado del cultivo usando los dispositivos conectados a Internet. Los resultados indicaron que la introducción del sistema IoT potencia la vigilancia y el control del cultivo hidropónico, permitiendo la automatización de procesos que en un sistema tradicional requieren de inspección manual. Por tanto, la propuesta realizada constituye una solución tecnológica factible para aplicaciones de monitoreo en el campo agrícola a pequeña escala, y sirve como plataforma para futuras ampliaciones mediante inclusión de nuevos sensores, algoritmos inteligentes o plataformas de análisis de datos.

La **Tabla 4** resume las principales diferencias entre un sistema hidropónico convencional y el prototipo IoT desarrollado.

Tabla 4: Comparación de un sistema hidropónico convencional y el prototipo IoT desarrollado.

Parámetros del Sistema	Sistema hidropónico convencional	Prototipo IoT desarrollado
Monitoreo de temperatura	Manual	Automático mediante DHT22
Monitoreo de humedad	Manual	Automático mediante DHT22
Medición de pH	Manual y periódica	Automática mediante sensor de pH
Monitoreo del nivel de agua	Inspección visual	Sensor de nivel conectado al ESP32
Activación de la bomba	Manual	Automática y remota mediante MQTT
Control del ventilador	Manual	Automático y remoto
Comunicación	No disponible	Wi-Fi IEEE 802.11
Protocolo de comunicación	No aplica	MQTT sobre TCP/IP
Monitoreo remoto	No disponible	Aplicaciones Windows y Android
Escalabilidad	Limitada	Alta, mediante nuevos nodos IoT

Aunque no se realizó una comparación cuantitativa con otro sistema comercial, el análisis desarrollado permitió comprobar que el prototipo implementado incorpora funcionalidades de monitoreo remoto, automatización y comunicación IoT que no se encuentran presentes en un cultivo hidropónico convencional. Esto demuestra la viabilidad técnica de la arquitectura propuesta y su potencial para mejorar la gestión de cultivos hidropónicos mediante tecnologías de Internet de las Cosas.

4.2. Discusión de resultados

Los resultados obtenidos durante la fase de implementación y validación demuestran que el prototipo IoT desarrollado cumplió satisfactoriamente con los objetivos planteados en la investigación. La integración entre el hardware, el software embebido, la infraestructura de comunicaciones y las aplicaciones de monitoreo permitió verificar el funcionamiento coordinado del sistema durante las pruebas realizadas en el cultivo hidropónico, garantizando la adquisición, transmisión, procesamiento y visualización de la información en tiempo real.

Desde la perspectiva de las telecomunicaciones, la arquitectura implementada basada en MQTT sobre la pila TCP/IP tiene un rendimiento estable en la comunicación entre el Esp32 y el broker Mosquitto con las aplicaciones cliente en los SO Windows y Android. El modelo Publicador / Suscriptor permitió un desacoplamiento entre los dispositivos de adquisición y los clientes de monitoreo y posibilitó la distribución paralela de los datos y los comandos de control sin interrumpir la comunicación. Con estas pruebas experimentales, los sensores de temperatura, humedad, pH y nivel de agua facilitaron la obtención de datos suficiente para monitorear los estados del cultivo. Se detectaron pequeñas fluctuaciones en las mediciones del sensor de pH, y también algunas variaciones en el sensor de nivel del agua debido al desgaste provocado por el uso en la fase experimental, sin embargo, estos eventos no comprometieron el funcionamiento general del sistema. El desarrollo de la Lógica de control permite conservar en operación el cultivo, manteniendo en forma automática las condiciones definidas para el bombeo de recirculación y del ventilado. La monitorización del cultivo durante dos semanas se pudo ver el desarrollo de las plantas de lechuga bajo condiciones regidas por el sistema IoT. Las mediciones tomadas y observaciones realizadas a intervalos de tres días probaron que el prototipo fue capaz de realizar un monitoreo consistente de las variables ambientales y apoyar la administración del cultivo con la automatización de los procesos de monitoreo y control, disminuyendo la carga de intervención manual sobre el operador.

En conjunto, estos resultados avalan la factibilidad técnica de la arquitectura propuesta y demuestra que el prototipo es una plataforma funcional para la monitorización y control de un pequeño cultivo hidropónico. Por otro lado, la estructura modular que se utilizó permite agregar nuevos sensores, algoritmos de análisis o actuadores, lo que posibilita extender las funciones del sistema sin cambiar en demasiado la infraestructura de comunicaciones.

4.3. Limitaciones del sistema

Como en todo desarrollo experimental, el prototipo implementado presenta ciertas limitaciones que deben considerarse al interpretar los resultados obtenidos. La investigación tuvo un enfoque académico y de validación tecnológica, por lo que el sistema fue diseñado como una plataforma de monitoreo y control para un cultivo hidropónico de pequeña escala, sin pretender sustituir soluciones comerciales destinadas a la producción agrícola intensiva.

Una de las principales limitaciones identificadas corresponde a los sensores empleados durante el desarrollo. El sensor de pH utilizado es un dispositivo de bajo costo cuya precisión depende de una adecuada calibración y de condiciones estables de medición, pudiendo presentar pequeñas fluctuaciones y tiempos de estabilización antes de entregar un valor definitivo. De manera similar, el sensor de nivel de agua evidenció desgaste progresivo debido a su utilización continua durante las pruebas experimentales, ocasionando variaciones ocasionales en las lecturas obtenidas como se puede en la Figura 50 y 51.

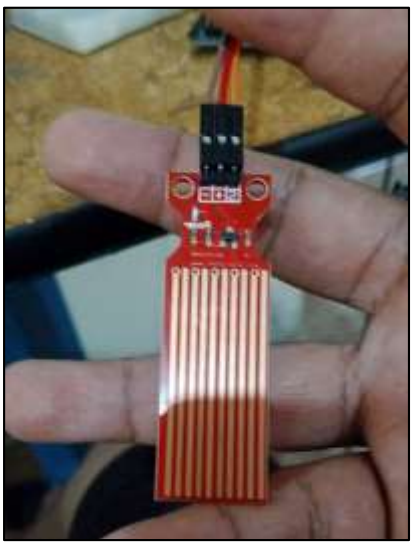


Figura 50: Sensor de nivel de agua antes de las pruebas.

Figura 51: Sensor luego de la exposición para controlar el pH del agua.

Otra limitación está relacionada con el tiempo de evaluación del cultivo. Aunque el seguimiento realizado permitió verificar el correcto funcionamiento del sistema durante dos semanas, un periodo de monitoreo más prolongado permitiría analizar con mayor profundidad el comportamiento del cultivo frente a cambios estacionales, variaciones climáticas y diferentes etapas de crecimiento de las plantas.

Finalmente, el desempeño general del sistema depende de la disponibilidad de la red inalámbrica y de la estabilidad de la comunicación con el broker MQTT. Aunque durante las pruebas no se presentaron fallos importantes de conectividad, la calidad de la red Wi-Fi puede influir en la pérdida de transmisión de los datos por falta de conexión y en el tiempo de respuesta de los comandos enviados hacia los actuadores, como se muestra en la Figura 53. A pesar de que el sistema puede encontrarse conectado y funcional, una conexión Wi-Fi deficiente o fallas en la comunicación con los brokers pueden impedir la actualización de datos en la interfaz de la aplicación Android, aspecto que debe considerarse en futuras implementaciones de mayor escala, véase Figura 52.

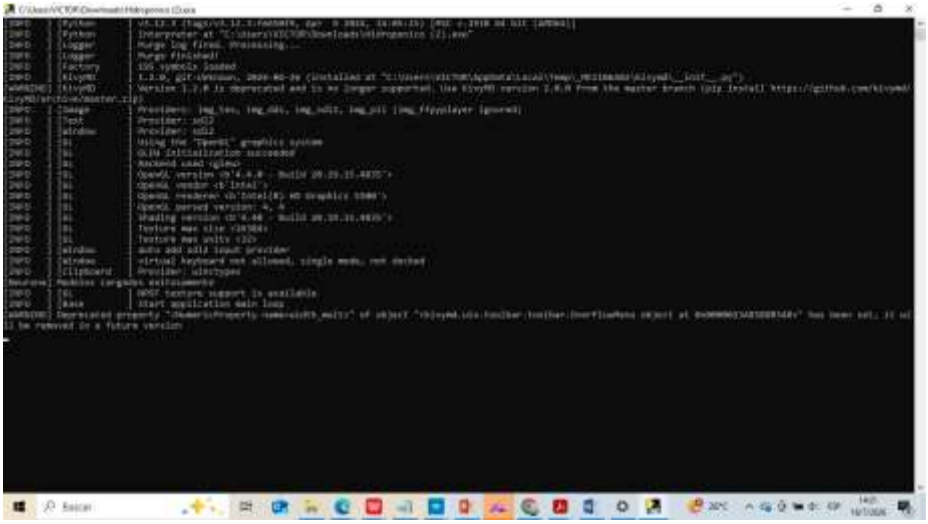


Figura 52: Correcto funcionamiento del APK.

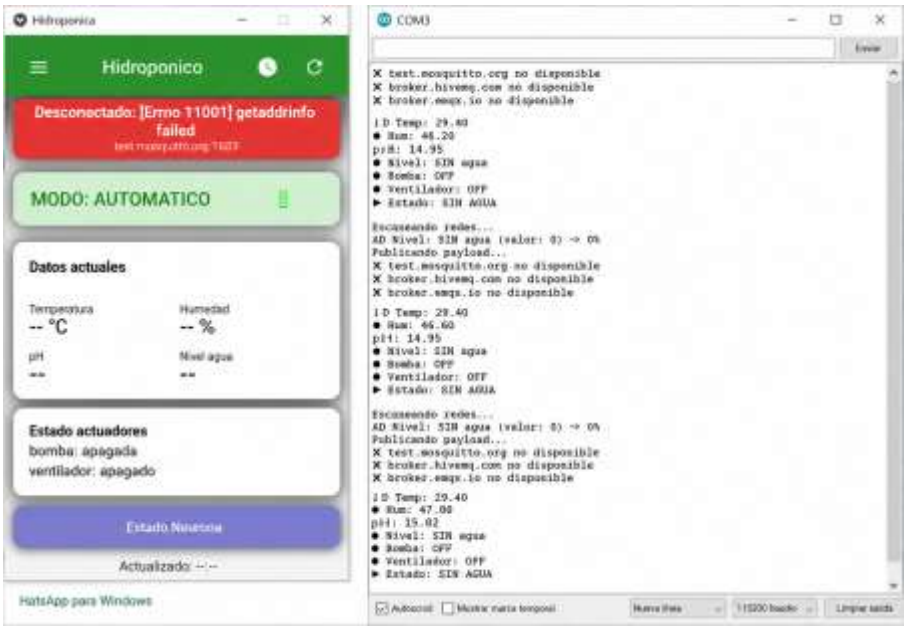


Figura 53: Evidente pérdida de la transmisión de datos debido a mala conexión de Wi-Fi

4.4. Cumplimiento de los objetivos del proyecto

Una vez finalizadas las etapas de diseño, implementación, integración y validación del prototipo IoT, se realizó una evaluación del grado de cumplimiento de los objetivos planteados al inicio de la investigación. Para ello, se analizaron los resultados obtenidos durante las pruebas experimentales, verificando que las actividades desarrolladas permitieron alcanzar las metas propuestas dentro del alcance definido para el proyecto.

La **Tabla 5** presenta la relación entre los objetivos específicos formulados y las

evidencias obtenidas durante el desarrollo del sistema, permitiendo demostrar el cumplimiento de cada uno de ellos mediante los resultados experimentales alcanzados

Tabla 5: Cumplimiento de los objetivos especifico del proyecto

Objetivo específico	**Evidencia de cumplimiento**	**Resultado**
Diseñar y desarrollar un sistema que permita monitorear las variables ambientales de un cultivo hidropónico usando sensores de temperatura, humedad, pH y nivel del agua.	Se ha desarrollado un sistema basado en el ESP32 que integra los sensores DHT22, pH y nivel de agua, que permite adquirir y procesar de modo continuo las variables ambientales del cultivo.	Cumplido
Implementar una infraestructura de comunicaciones IoT utilizando MQTT.	Se propuso una arquitectura basada en MQTT y Mosquitto que facilita la comunicación entre ESP32 y las plataformas Windows y Android a través de Wi-Fi (IEEE 802.11) para dos vías mediante.	Cumplido
Desarrollar un sistema de monitoreo y control automático.	El sistema permitió monitorear las variables ambientales y controlar la bomba y el ventilador de forma automática.	Cumplido

Nota: el cumplimiento de los objetivos específicos fue validado mediante pruebas de funcionamiento, la implementación del sistema y los resultados experimentales obtenidos.

Se ha desarrollado un sistema basado en ESP32 combinando los sensores DHT22, pH y nivel de agua para obtener de forma continua las variables ambientales del cultivo. Con el ESP32 fue posible obtener las variables ambientales necesarias para monitorear el estado del cultivo, y la infraestructura de comunicaciones basada en MQTT facilitó el intercambio constante de datos entre el nodo IoT y las aplicaciones desarrolladas.

Del mismo modo, el sistema de control desarrollado también permitió la automatización de los actuadores según las condiciones definidas para el cultivo, disminuyendo la necesidad de intervención por parte del usuario. Por lo tanto, el

logro de los objetivos específicos permitió alcanzar el objetivo general de la investigación.

CONCLUSIONES

Se desarrolló e implementó un prototipo IoT para el monitoreo y control de variables ambientales en un cultivo hidropónico, integrando un ESP32 con sensores de temperatura, humedad, pH y nivel de agua para la adquisición y procesamiento de datos en tiempo real.

La comunicación inalámbrica mediante **Wi-Fi (IEEE 802.11n)** y el protocolo **MQTT** permitió la transmisión confiable de información hacia las aplicaciones desarrolladas para Windows y Android, facilitando el monitoreo y el control remoto del sistema.

Las pruebas realizadas demostraron el correcto funcionamiento del prototipo, permitiendo el monitoreo continuo de las variables ambientales y el control automático. Aunque se observaron pequeñas variaciones en algunos sensores de bajo costo, estas no afectaron el desempeño general del sistema.

El prototipo constituye una alternativa de bajo costo y fácil implementación para el monitoreo de cultivos hidropónicos, con posibilidades de ampliación mediante nuevos sensores, almacenamiento en la nube o integración con otras tecnologías IoT.

RECOMENDACIONES

Extender el período de evaluación del cultivo mediante la inclusión de diferentes ciclos productivos y especies vegetales con el fin de estudiar el desempeño del sistema bajo diversas condiciones.

• Utilizar la naturaleza modular del sistema para la integración de nuevos sensores y actuadores, aumentando así sus capacidades de monitoreo y control de acuerdo a las necesidades de la aplicación. También se sugiere implementar un protocolo para calibrar y dar mantenimiento preventivo a los sensores a fin de asegurar la estabilidad y confiabilidad de las mediciones en el tiempo.

• Integrar nuevas variables ambientales y algoritmos de inteligencia artificial que permitan mejorar la toma de decisiones y la eficiencia del sistema automático de control.

• La arquitectura modular basada en Esp32, MQTT y Mosquitto puede seguir para permitir la integración de más sensores, actuadores, clientes aplicaciones en posteriores desarrollos.

REFERENCIAS BIBLIOGRÁFICAS

[1] FAO, *The State of Food and Agriculture 2020: Overcoming the crisis of food systems*. Dirección: https://doi.org/10.4060/cb1209es

[2] FAO, *The State of Food and Agriculture 2020: Overcoming water challenges in agriculture*. FAO Publications, 2020.

[3] S. Kumar, M. García y J. Rodríguez, «Internet de las Cosas (IoT) aplicado a la agricultura inteligente en Latinoamérica: Beneficios y desafíos», *Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 16, n.º 2, págs. 145-152, 2021.

[4] S. A. A. Abu Sneineh Anees, «Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things», *ScienceDirect*, 2023.

[5] A. K. N. B. D. S. S. G. Vasantha Lakshmi C., «Hydroponic Farm Monitoring System Using IoT», *RE Journals*, vol. 3, n.º 10, abr. de 2020.

[6] M. B. K. P. K. J. S. K. Pavan Kumar R., «Reliable IoT-Based Monitoring and Control of Hydroponic Systems», *Technologies*, vol. 10, n.º 1, 2022.

[7] A. S. R. Mishra y P. Kumar, «Role of IoT in Smart Agriculture: A Review on Applications and Challenges», *International Journal of Computer Applications*, n.º 8887, pág. 975, 2020.

[8] Y. B. Y. LeCun y P. Huang, «Gradient-Based Learning Applied to Document Recognition», *Proceedings of the IEEE*, vol. 86, n.º 11, págs. 2278-2324, 2020.

[9] W. X. Zhang y J. Li, «Automation in Agriculture: A Review on Current Technologies and Future Directions», *Agricultural Systems*, n.º 172, págs. 33-45, 2021.

[10] S. Qazi, B. A. Khawaja, and Q. U. Farooq, "IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends," IEEE Access, vol. 10, pp. 21219–21235, 2022, doi: **10.1109/ACCESS.2022.3152544.**

[11] A. M. J. González y L. Pérez, «Skills for the Future: The Role of Telecommunications in Agricultural Innovation», *Journal of Engineering Education*, vol. 110, n.º 3, págs. 456-472, 2021.

[12] R. T. S. Khadijah Febriana R., «Enhancing Hydroponic Farming Productivity Through IoT-Based Multi-Sensor Monitoring System», en *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, 2024.

[13] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming: An Overview," *IEEE Access*, vol. 10, pp. 998–1024, 2022**.**

[14] R. H. K. Jhaveri, N. M. Patel, G. Srivastava, and V. R. Gadekallu, "The Role of Internet of Things in Education: A Systematic Review and Future Trends," *Education and Information Technologies*, vol. 27, no. 8, pp. 11061–11094, 2022.

[15] S. K. Kassab, A. M. Alhassan, and M. H. Ahmed, "MQTT-Based Internet of Things Applications: Architecture, Challenges, and Solutions," *IEEE Access*, vol. 9, pp. 165547–165569, 2021, doi: **10.1109/ACCESS.2021.3135333**.

[16] V. P. Kour and S. Arora, "Recent Developments of the Internet of Things in Agriculture: A Survey," IEEE Access, vol. 8, pp. 129924–129957, 2020, doi: **10.1109/ACCESS.2020.3009298**.

[17] OASIS, *MQTT Version 3.1.1*, OASIS Standard, Dec. 2015. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html**.**

[18] M. **Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune**, "**Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk**," *IEEE Access*, vol. 7, pp. 129551–129583, 2019, doi: **10.1109/ACCESS.2019.2932609**.

[19] Espressif Systems, «ESP32 Series Datasheet», 2024. [En línea]. Disponible en: https://www.espressif.com

[20] C. Parra-López, S. Ben Abdallah, G. Garcia-Garcia, A. Hassoun, H. Trollman, S. Jagtap, S. Gupta, A. Aït-Kaddour, S. Makmuang, and C. Carmona-Torres, "**Digital technologies for water use and management in agriculture: Recent applications and future outlook**," *Agricultural Water Management*, vol. 309, Art. no. 109347, 2025, doi: **10.1016/j.agwat.2025.109347**.

[21] Palmitessa, O. D., Signore, A., & Santamaria, P. (2024). Advancements and future perspectives in nutrient film technique hydroponic system: A comprehensive review and bibliometric analysis. *Frontiers in Plant Science, 15, 1504792. https://doi.org/10.3389/fpls.2024.1504792*.

[22] C. Álvarez G., A. Soto P. y F. Watkins O., *"Simulación de controladores digitales,"* Ingeniare. Revista Chilena de Ingeniería*, vol. 17, no. 3, pp. 309–316, 2009, doi: 10.4067/S0718-33052009000300004.*

[23] Arduino, «Arduino IDE Documentation», 2025. [En línea]. Disponible en: https://docs.arduino.cc/software/ide/.

[24] J. Portley, «Todo lo que necesitas saber sobre los escudos Arduino», 18-oct-2023. [En línea]. Disponible en: https://knowhow.distrelec.com/electronics/everything-you-need-to-know-about-arduino-shields/. [Último acceso: 2026].

[25] Megatronic, «Módulo sensor temperatura y humedad DHT22 con cable Arduino», 2022. [En línea]. Disponible en: https://megatronica.cc/producto/modulo-sensor-temperatura-y-humedad-dht22-con-cable-arduino/. [Último acceso: 2026].

[26] Daruifuno, «Sensor de pH analógico de 4-20 mA», mayo de 2024. [En línea]. Disponible en: https://es.daruifuno.com/analog-ph-sensor/. [Último acceso: jul. 2026].

[27] IFM Electronics, «Sensores de nivel», 2026. [En línea]. Disponible en: https://www.ifm.com/es/es/category/200_020_020#/. [Último acceso: 2026].

[28] KivySpaceGame, «Buildozer», junio de 2015. [En línea]. Disponible en: https://kivyspacegame.wordpress.com/2014/06/30/tutorial-how-to-build-python-for-android-with-ubuntu-and-buildozer/. [Último acceso: 2026].

[29] ORING, «What is MQTT and how it works?,» 2026. [En línea]. Disponible en: https://oringnet.com/en/knowledge-base/what-is-mqtt-and-how-it-works. [Último acceso: 2026].

[30] ProcessOn, «Diagrama de arquitectura de componentes MVC,» 2026. [En línea]. Disponible en: https://www.processon.io/es/view/mvc-component-architecture-diagram/6a14f9c851bbd3339a79fdae. [Último acceso: 2026].

ANEXOS

Código fuente del sistema desarrollado

```
#include "sensores.h"

#include "control.h"
#include "mqtt.h"

// Variables para control de ventilador con histéresis (ajustado para clima
cálido)
float tempVentiladorOn = 34.0;   // Se enciende a 34°C
float tempVentiladorOff = 31.0;  // Se apaga a 31°C
bool ventiladorEstado = false;

void setup() {
  Serial.begin(115200);

  initSensores();
  initActuadores();
  initWi-Fi();
  initMQTT();

  Serial.println("✅ Sistema iniciado");
  Serial.println("💡 Ventilador se encenderá a 34°C y se apagará a 31°C");
  Serial.println("💧 Sensor de nivel: HAY agua cuando valor > 350");
}

void loop() {

  // 🔁 Mantener conexión MQTT
  loopMQTT();

  // 📡 Lectura de sensores
  float temp = leerTemperatura();
  float hum  = leerHumedad();
  float ph   = leerPH();
  int nivel  = leerNivel();  // 1 = HAY agua, 0 = SIN agua

  // ========================================
  // 🟢 CONTROL DE BOMBA por pH
  // ========================================
  int bomba = 0;
  String estado;
 // Control de BOMBA por pH (rango ideal: 5.5 - 6.5 para hidroponía)
```

```
if (ph < 5.5 || ph > 6.5) {
  // pH fuera de rango, activar bomba para corregir
  if (nivel == 1) {  // Solo si HAY agua
    activarSistema();
    bomba = 1;
    estado = "CORRIGIENDO pH";
  } else {
    desactivarSistema();
    bomba = 0;  // 🔄 CAMBIADO: antes era 1, ahora es 0
    estado = "SIN AGUA";
  }
} else {
  // pH en rango normal, bomba apagada
  desactivarSistema();
  bomba = 0;
  estado = "NORMAL";
}
  // ========================================

  // ========================================
  // ⚫ CONTROL DEL VENTILADOR CON HISTÉRESIS
  // ========================================
  int ventilador = 0;

  if (!ventiladorEstado && temp > tempVentiladorOn) {
    // Ventilador apagado y temperatura superó el umbral → ENCENDER
    encenderVentilador();
    ventiladorEstado = true;
    ventilador = 1;
    Serial.println("🌬️ Ventilador ENCENDIDO (temp: " + String(temp) + "°C)");

  } else if (ventiladorEstado && temp < tempVentiladorOff) {
    // Ventilador encendido y temperatura bajó del umbral → APAGAR
    apagarVentilador();
    ventiladorEstado = false;
    ventilador = 0;
    Serial.println("⚫ Ventilador APAGADO (temp: " + String(temp) + "°C)");
  }

  // Actualizar valor del ventilador según estado
  if (ventiladorEstado) {
    ventilador = 1;
  } else {
    ventilador = 0;
  }
```

```
// ========================================

// ========================================
// 📡 Envío por MQTT (CON 7 PARÁMETROS)
// ========================================
int bombaMQTT = (bomba == 1) ? 0 : 1;  // Invertir si es necesario
enviarDatos(ph, temp, hum, nivel, ventilador, bombaMQTT, estado);
  // 📟 Monitor serie (debug)
  Serial.println("-----");
  Serial.print("🌡️ Temp: "); Serial.println(temp);
  Serial.print("💧 Hum: "); Serial.println(hum);
  Serial.print("🧪 pH: "); Serial.println(ph);
  Serial.print("🫧 Nivel: "); Serial.println(nivel == 1 ? "CON agua" : "SIN
agua");
  Serial.print("💧 Bomba: "); Serial.println(bomba == 1 ? "ON" : "OFF");
  Serial.print("🌬️ Ventilador: "); Serial.println(ventilador == 1 ? "ON" :
"OFF");
  Serial.print("📢 Estado: "); Serial.println(estado);

  delay(3000);
}

 #include "control.h"
#include <Arduino.h>

#define RELE_VENTILADOR 18
#define RELE_BOMBA 19

void initActuadores() {
  pinMode(RELE_VENTILADOR, OUTPUT);
  pinMode(RELE_BOMBA, OUTPUT);

  // Inicializar: ventilador apagado, bomba apagada
  digitalWrite(RELE_VENTILADOR, HIGH);
  digitalWrite(RELE_BOMBA, LOW);    // 🔄 CAMBIADO: LOW = bomba apagada
}

void activarSistema() {
  digitalWrite(RELE_BOMBA, HIGH);   // 🔄 CAMBIADO: HIGH = activar bomba
}

void desactivarSistema() {
  digitalWrite(RELE_BOMBA, LOW);    // 🔄 CAMBIADO: LOW = desactivar bomba
```

```
void encenderVentilador() {
  digitalWrite(RELE_VENTILADOR, LOW);
}

void apagarVentilador() {
  digitalWrite(RELE_VENTILADOR, HIGH);
}

void encenderBomba() {
  digitalWrite(RELE_BOMBA, HIGH);   // 🔄 CAMBIADO: HIGH = encender bomba
}

void apagarBomba() {
  digitalWrite(RELE_BOMBA, LOW);    // 🔄 CAMBIADO: LOW = apagar bomba
}
```

```
#ifndef CONTROL_H
#define CONTROL_H

void initActuadores();
void activarSistema();
void desactivarSistema();
void encenderVentilador();
void apagarVentilador();
void encenderBomba();
void apagarBomba();

#endif
```

```
#include "mqtt.h"
#include <Wi-Fi.h>
#include <PubSubClient.h>

// ----------------------
// 🔐 Credenciales Wi-Fi (Multired)
// ----------------------
const char* redes[] = {"XTRIM_DELGADO", "LIAMYENCE", "prueba1"};
const char* claves[] = {"0928351600", "68928223YE", "1234567890"};
const int totalRedes = 3;
```

```
}

// ----------------------
// 🌐 Brokers MQTT (Redundancia Triple)
// ----------------------
const char* servers[] = {"test.mosquitto.org", "broker.hivemq.com",
"broker.emqx.io"};
Wi-FiClient espClients[3];
PubSubClient clients[3] = { PubSubClient(espClients[0]),
PubSubClient(espClients[1]), PubSubClient(espClients[2]) };

unsigned long lastReconnectAttempt = 0;

// ----------------------
// 📡 Wi-Fi con Escaneo
// ----------------------
void initWi-Fi() {
  Serial.println("\nEscaneando redes...");
  int n = Wi-Fi.scanNetworks();
  bool conectado = false;

  for (int i = 0; i < n; ++i) {
    for (int j = 0; j < totalRedes; j++) {
      if (Wi-Fi.SSID(i) == redes[j]) {
        Serial.printf("Intentando conectar a: %s\n", redes[j]);
        Wi-Fi.begin(redes[j], claves[j]);

        int intentos = 0;
        while (Wi-Fi.status() != WL_CONNECTED && intentos < 20) {
          delay(500);
          Serial.print(".");
          intentos++;
        }

        if (Wi-Fi.status() == WL_CONNECTED) {
          Serial.println("\n¡Conectado!");
          Serial.print("IP: ");
          Serial.println(Wi-Fi.localIP());
          conectado = true;
          break;
        }
      }
    }
```

```
    if (conectado) break;
  }
}

// ----------------------
// 🔗 MQTT - Inicialización
// ----------------------
void initMQTT() {
  for (int i = 0; i < 3; i++) {
    clients[i].setServer(servers[i], 1883);
  }
}

// ----------------------
// 🔄 Reconexión
// ----------------------
void loopMQTT() {
  if (Wi-Fi.status() != WL_CONNECTED) {
    initWi-Fi();
    return;
  }

  unsigned long now = millis();
  if (now - lastReconnectAttempt > 5000) {
    lastReconnectAttempt = now;

    for (int i = 0; i < 3; i++) {
      if (!clients[i].connected()) {
        String clientId = "ESP32-Cultivo-" + String(random(0xffff), HEX);
        if (clients[i].connect(clientId.c_str())) {
          Serial.printf("Broker %s conectado\n", servers[i]);
        }
      }
    }
  }

  for (int i = 0; i < 3; i++) {
    if (clients[i].connected()) clients[i].loop();
  }
}

// ----------------------
```

```
// 📤 Enviar datos a todos los brokers activos
// ----------------------
void enviarDatos(float ph, float temp, float hum, int nivel, int ventilador,
int bomba, String estado) {

  // Inversión de lógica para bomba (como pediste)
  int bombaMQTT = (bomba == 1) ? 0 : 1;

  String payload = String("{\"temp\":") + String(temp,2) +
                   ",\"hum\":" + String(hum,2) +
                   ",\"ph\":" + String(ph,2) +
                   ",\"nivel\":" + String(nivel) +
                   ",\"bomba\":" + String(bombaMQTT) +
                   ",\"ventilador\":" + String(ventilador) +
                   ",\"alerta\":\"" + estado + "\"}";

  Serial.println("Publicando payload...");

  for (int i = 0; i < 3; i++) {
    if (clients[i].connected()) {
      if (clients[i].publish("cultivo/datos", payload.c_str())) {
        Serial.printf("✅ Enviado a %s\n", servers[i]);
      }
    } else {
      Serial.printf("❌ %s no disponible\n", servers[i]);
    }
  }
}
```

```
#ifndef MQTT_H
#define MQTT_H

#include <Arduino.h>
#include <PubSubClient.h>

// --- Funciones principales ---

void initWi-Fi();

void initMQTT();
```

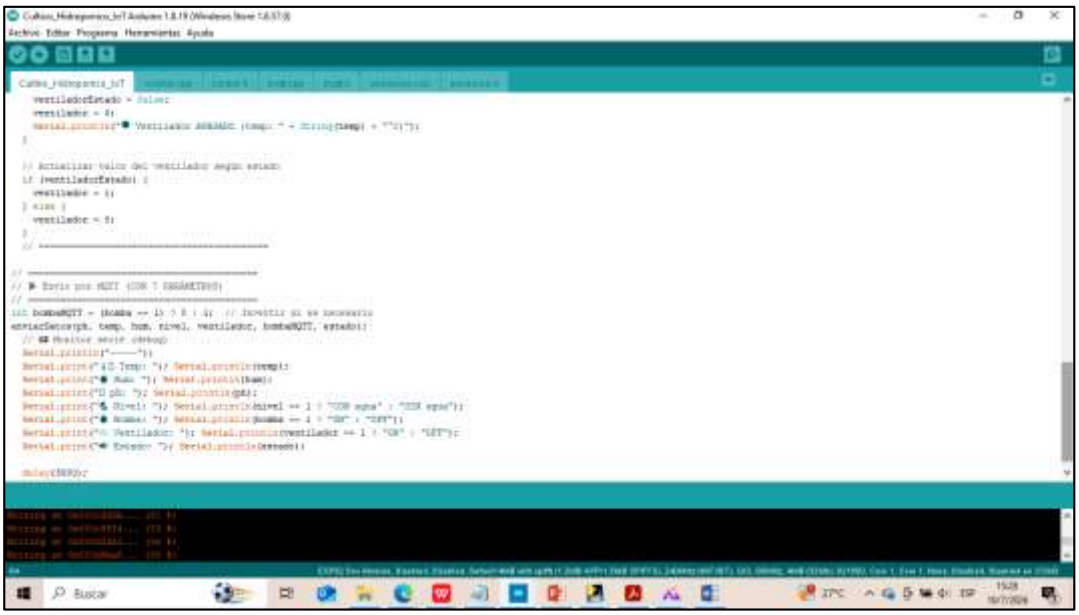
```
void loopMQTT();


// --- Función de envío de datos ---
// Asegúrate de que los parámetros coincidan con la implementación en mqtt.cpp
void enviarDatos(float ph, float temp, float hum, int nivel, int ventilador, int bomba, String estado);


#endif


#include "sensores.h"
#include <DHT.h>


#define DHTPIN 4
#define DHTTYPE DHT22
#define PH_PIN 34
#define NIVEL_PIN 35   // 🟢 pin sensor de nivel (ANALÓGICO)


DHT dht(DHTPIN, DHTTYPE);


// ----------------------
// Inicialización
// ----------------------
void initSensores() {
  dht.begin();
  pinMode(NIVEL_PIN, INPUT);

}
```

```
// ----------------------
// Temperatura
// ----------------------
float leerTemperatura() {
  float t = dht.readTemperature();

  if (isnan(t)) {
    Serial.println("Error leyendo temperatura");
    return -1;
  }

  return t;
}

// ----------------------
// Humedad
// ----------------------

float leerHumedad() {
  float h = dht.readHumidity();

  if (isnan(h)) {
    Serial.println("Error leyendo humedad");
    return -1;
  }

  return h;
}
```

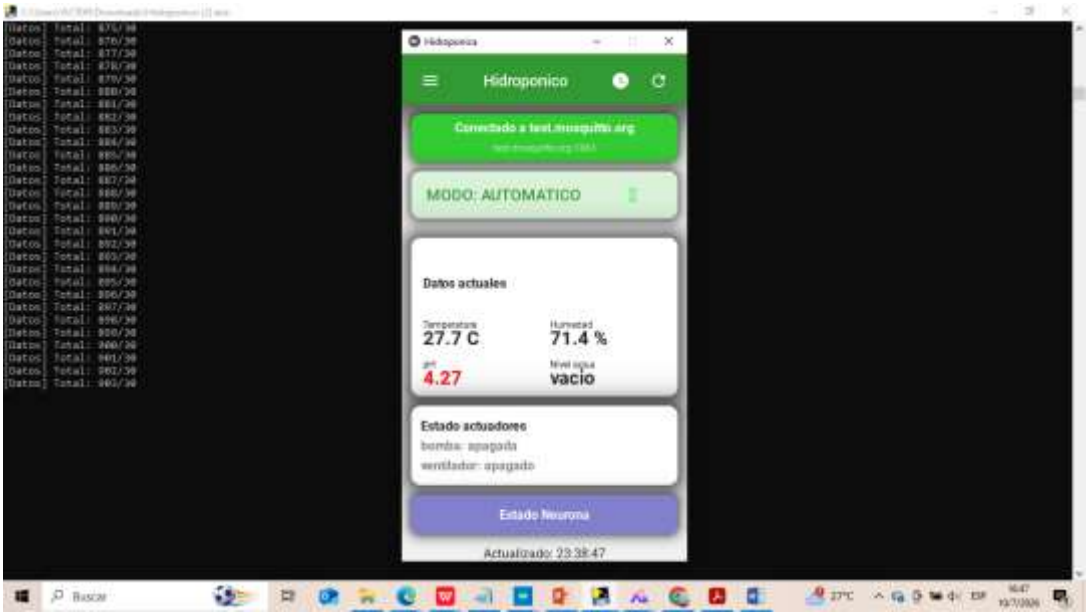
```
// ----------------------
// pH
// ----------------------
float leerPH() {
  int raw = analogRead(PH_PIN);
  float voltaje = (raw / 4095.0) * 3.3;
  float ph = 7 + ((2.5 - voltaje) / 0.18);
  return ph;
}

// ----------------------
// Nivel de agua (ENVÍA 0% o 100%)
// ----------------------
int leerNivel() {
  int valor = analogRead(NIVEL_PIN);  // Lee valor 0-4095 (ESP32)

  // Umbral para determinar si hay agua (ajústalo según tu sensor)
  int umbral = 350;


  if (valor > umbral) {
    Serial.print("● Nivel: HAY agua (valor: ");
    Serial.print(valor);
    Serial.println(") → 100%");
    return 100;  // ✓ ENVÍA 100% cuando hay agua
  } else {
    Serial.print("⚠□ Nivel: SIN agua (valor: ");
    Serial.print(valor);
    Serial.println(") → 0%");
```

```
    return 0;    // ✓ ENVÍA 0% cuando no hay agua
  }
}

#ifndef SENSORES_H
#define SENSORES_H

void initSensores();

float leerTemperatura();
float leerHumedad();
float leerPH();
int leerNivel();

#endif
```

Código de Arduino IDE



Creación y pruebas del prototipo



Pruebas del Prototipo instalado en la estructura