

Monitorización IoT de Temperatura y Humedad en Agricultura Inteligente mediante Sistema Web Responsivo



Colaboración

Armando Roman Gallardo, Jose Roman Herrera Morales, Sara Sandoval Carrillo and Armando Román Gallardo Heredia, Universidad de Colima

Fecha de recepción: 21 de noviembre de 2025

Fecha de aceptación: 21 de noviembre de 2025

RESUMEN: La agricultura enfrenta desafíos crecientes relacionados con el cambio climático y la optimización de recursos. Este trabajo presenta el diseño, implementación y validación de un prototipo de sistema web responsive para monitorización en tiempo real de temperatura y humedad en entornos agrícolas mediante IoT. La metodología incluyó desarrollo iterativo integrando sensores DHT22 conectados a microcontroladores ESP32, transmitiendo datos a través de WebSockets a una aplicación web desarrollada con React.js y Tailwind CSS. La validación mediante pruebas funcionales en laboratorio durante 72 horas continuas demostró visualización en tiempo real con latencia inferior a 2 segundos, configuración personalizada por cultivo, generación de alertas cuando los parámetros exceden rangos óptimos y almacenamiento de más de 1,000 mediciones sin pérdidas mediante SQLite. El sistema evidenció recuperación automática ante desconexiones simuladas con 95% de éxito y tiempo promedio de reconexión de 12 segundos. Este prototipo contribuye como fundamento tecnológico para la toma de decisiones agrícolas informadas, facilitando el monitoreo continuo de condiciones ambientales críticas en implementaciones escalables.

PALABRAS CLAVE: Agricultura inteligente, Internet de las Cosas, Monitorización en tiempo real, Sistema responsive, WebSockets.

ABSTRACT: Agriculture faces growing challenges related to climate change and resource optimization. This work presents the design, implementation and validation of a responsive web system prototype for real-time monitoring of temperature and humidity in agricultural environments through IoT. The methodology included iterative development integrating DHT22 sensors connected to ESP32 microcontrollers, transmitting data through WebSockets to a web application developed with React.js and Tailwind CSS. Validation through functional testing in laboratory for 72 continuous hours demonstrated real-time visualization with latency less than 2 seconds, personalized configuration per crop, alert generation when parameters exceed optimal ranges and storage of more than 1,000 measurements without losses using SQLite. The system evidenced automatic recovery from simulated disconnections with 95% success rate and average reconnection time of 12 seconds. This prototype contributes as technological foundation for informed agricultural decision-making, facilitating continuous monitoring of critical environmental conditions in scalable implementations.

KEYWORDS: Smart agriculture, Internet of Things, Real-time monitoring, Responsive system, WebSockets.

INTRODUCCIÓN

La creciente demanda global de alimentos, sumada a los desafíos del cambio climático y la escasez de recursos, ha impulsado la adopción de tecnologías avanzadas en el sector agrícola. En este contexto, la agricultura inteligente, respaldada por el Internet de las Cosas (IoT), surge como una estrategia clave para optimizar la producción mediante el monitoreo de variables ambientales críticas como la temperatura y la humedad [1,2]. Estas variables son factores determinantes en el crecimiento y desarrollo de los cultivos, ya que variaciones fuera de los rangos óptimos pueden afectar negativamente la fotosíntesis, la absorción de nutrientes y la resistencia a enfermedades, reduciendo la productividad agrícola [3,4].

La integración de sensores IoT en la agricultura permite recopilar datos en tiempo real sobre temperatura y humedad, facilitando decisiones informadas y oportunas [5]. Los agricultores pueden monitorear remotamente las condiciones ambientales y realizar ajustes inmediatos, mejorando el rendimiento de los cultivos y minimizando pérdidas. El desarrollo de aplicaciones web responsivas proporciona una plataforma intuitiva para visualizar y analizar los datos recopilados, siendo accesibles desde múltiples dispositivos y facilitando intervenciones oportunas [6]. El término “responsivo” se refiere al diseño de interfaces web que se adaptan automáticamente a diferentes resoluciones de pantalla y dispositivos.

El monitoreo en tiempo real incrementa la eficiencia de las operaciones agrícolas y disminuye los riesgos asociados a eventos climáticos extremos, ya que detectar tempranamente anomalías ambientales permite implementar medidas preventivas o correctivas, evitando daños considerables [4]. Estas herramientas contribuyen a una agricultura más precisa y eficiente, permitiendo el uso racional de recursos y promoviendo prácticas resilientes al cambio climático.

Con base en las limitaciones identificadas en sistemas previos de monitoreo agrícola, este trabajo plantea la siguiente hipótesis: la integración de sensores IoT de bajo costo (DHT22 y ESP32) con una arquitectura web responsive basada en comunicación WebSocket permitirá desarrollar un sistema de monitorización en tiempo real de temperatura y humedad que sea funcionalmente viable, accesible desde múltiples dispositivos, con mínima latencia y capaz de recuperarse automáticamente ante interrupciones de conectividad, superando las limitaciones de responsividad, configuración personalizada y robustez reportadas en la literatura previa.

Por lo tanto, este trabajo presenta el desarrollo y validación funcional de un prototipo de aplicación web responsive para verificar dicha hipótesis me-

diante el monitoreo en tiempo real de temperatura y humedad aplicables en entornos agrícolas. El sistema contribuye como fundamento tecnológico para la toma de decisiones basada en datos ambientales, facilitando procesos como la planificación del riego y el monitoreo de condiciones críticas en implementaciones escalables.

Revisión de literatura

La integración del Internet de las Cosas (IoT) en la agricultura ha transformado las prácticas tradicionales al permitir la monitorización precisa de variables ambientales esenciales como temperatura y humedad. Diversos estudios demuestran su efectividad para recopilar datos en tiempo real, optimizar la toma de decisiones y mejorar la eficiencia productiva [7,5], aunque con limitaciones específicas que requieren atención.

Muthmainnah et al. [8] desarrollaron un sistema de monitoreo de temperatura y humedad del suelo utilizando sensores DHT11 y YL69 conectados a un microcontrolador ESP8266, logrando una precisión promedio del 99.19% en temperatura y 97.95% en humedad. No obstante, su implementación se centró en invernaderos, sin considerar los desafíos de conectividad en zonas rurales abiertas, además de presentar limitaciones en la interfaz de usuario y accesibilidad desde dispositivos móviles.

Ikram et al. [9] desarrollaron sensores flexibles para monitorización de plantas, destacando su valor para monitorear microclimas. Sin embargo, identificaron desafíos relacionados con la durabilidad y el costo de implementación en gran escala, requiriendo mayor inversión inicial y presentando complejidad técnica que limita su adopción en pequeñas explotaciones agrícolas.

Rajak et al. [10] implementaron una plataforma IoT para agricultura inteligente basada en computación en el borde y la nube, demostrando la eficacia de las plataformas web para visualización de datos. Sin embargo, identificaron limitaciones en la responsividad de la interfaz y gestión de múltiples usuarios simultáneos. Estos enfoques suelen requerir infraestructuras complejas en la nube, inviables en regiones rurales con conectividad limitada y baja alfabetización digital.

Zamora-Izquierdo et al. [11] identificaron problemas relacionados con la conectividad en áreas rurales, la gestión de grandes volúmenes de datos y la necesidad de formación técnica para los agricultores. Muchos sistemas existentes carecen de interfaces intuitivas y configuración personalizada según tipos de cultivo. Los trabajos revisados no abordan concretamente cómo superar barreras críticas como la recuperación frente a pérdidas de conexión, el

despliegue rápido por personal no especializado o la integración de configuraciones específicas para distintos cultivos.

Considerando lo anterior, se identifica que los sistemas previos carecen de interfaces web verdaderamente responsivas que se adapten eficientemente a múltiples dispositivos, configuración personalizada simple para diferentes cultivos, y sistemas de alertas en tiempo real con baja latencia. Por lo cual se propone el presente trabajo para abordar estas limitaciones mediante el desarrollo de un prototipo de sistema web responsive con comunicación WebSocket y configuración adaptable por cultivo.

MATERIAL Y MÉTODOS

Recursos materiales

Para la elaboración del prototipo de aplicación web responsive, se integraron componentes de hardware, software e infraestructura tecnológica que, en conjunto, permiten el monitoreo y graficado en tiempo real de datos de temperatura y humedad en entornos agrícolas.

Hardware

Se empleó el sensor DHT22 para la medición precisa de temperatura y humedad, conectado a un microcontrolador ESP32 con capacidad de procesamiento y conectividad Wi-Fi para transmisión de datos en tiempo real. Para el desarrollo y pruebas se utilizó una computadora con 16 GB de RAM, disco SSD de 1TB y procesador Intel i7 de novena generación.

Software

El desarrollo de la interfaz web responsive se realizó con React.js y Tailwind CSS para interfaces dinámicas, mientras que el backend se construyó con Node.js y Express.js para gestión de solicitudes y comunicación en tiempo real mediante WebSockets. Para almacenamiento de datos se optó por SQLite, una base de datos ligera adecuada para gestión eficiente sin complejidad de sistemas robustos.

Infraestructura Tecnológica

La infraestructura se basa en un servidor Node.js que utiliza WebSockets para comunicación bidireccional en tiempo real entre cliente y servidor, asegurando actualización constante de datos de temperatura y humedad.

Metodología

La aplicación se desarrolló mediante un enfoque iterativo y flexible que permitió controlar y mejorar progresivamente cada etapa del proyecto. Se definieron siete iteraciones, comenzando con el análisis de requerimientos y planificación, seguido por la configuración del entorno y el prototipado de la

interfaz. Posteriormente, se desarrollaron los servicios backend y la base de datos, se integró el hardware con WebSocket para monitoreo en tiempo real, se validaron las funcionalidades, se desarrolló la interfaz web responsive con React y Tailwind CSS, y finalmente se realizó la fase de pruebas funcionales e implementación.

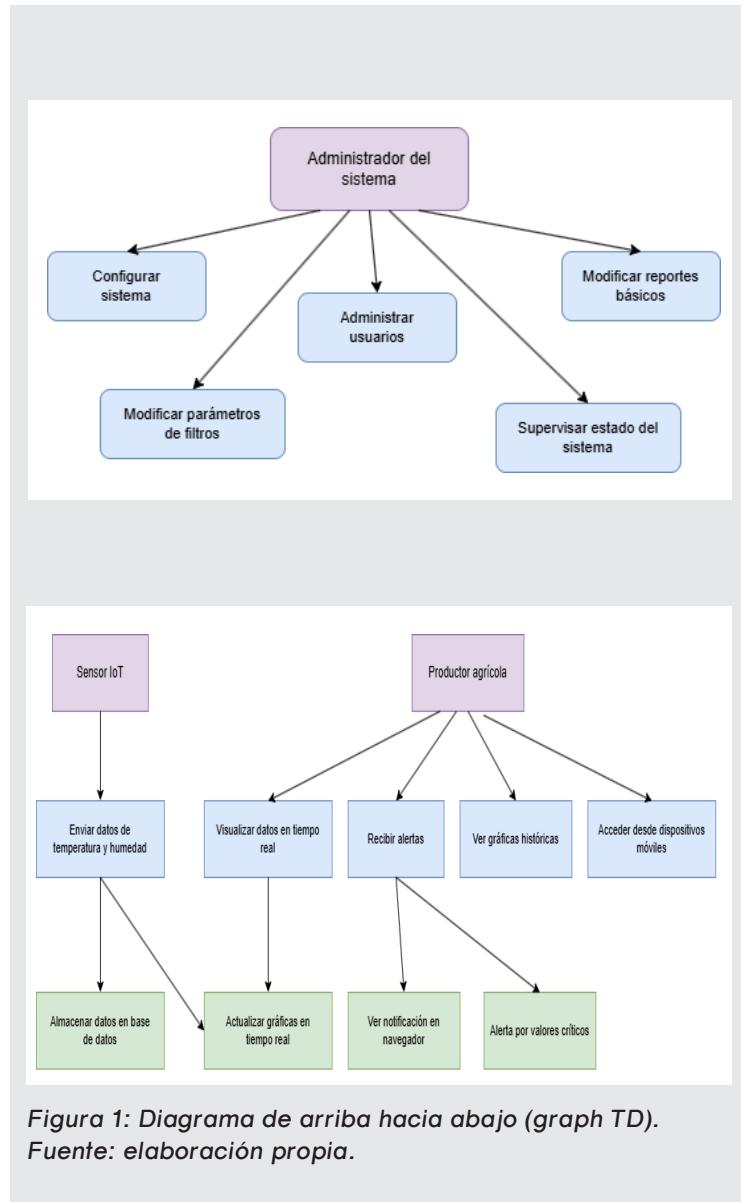


Figura 1: Diagrama de arriba hacia abajo (graph TD).
Fuente: elaboración propia.

Durante la primera fase se definieron los objetivos funcionales del sistema, como el monitoreo en tiempo real de temperatura y humedad desde un ESP32, identificando los roles, usuarios y necesidades técnicas. También se seleccionaron las tecnologías base: React.js, WebSocket, Node.js, y un ESP32 con sensor DHT. La configuración del entorno incluyó la preparación del ambiente de desarrollo con Vite, React y Tailwind CSS, implementando los primeros bocetos funcionales de la interfaz gráfica para el administrador del sistema y configurando las dependencias para la comunicación con la API y WebSocket.

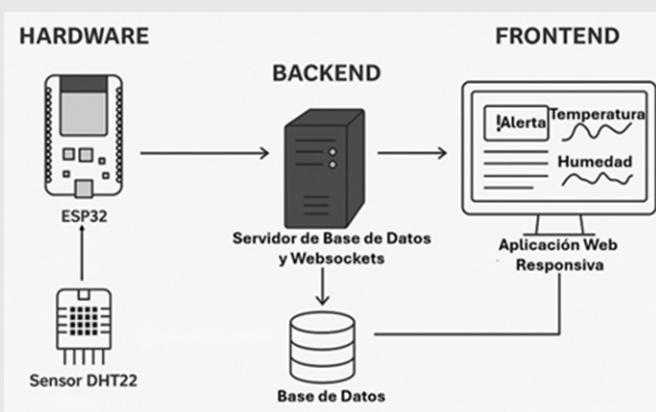


Figura 2: Diagrama general del sistema (hardware + backend + frontend).

Fuente: elaboración propia.

El desarrollo de la base de datos y creación de la API involucró la estructuración de un modelo de datos para almacenar las lecturas enviadas desde el ESP32, junto con la configuración del servidor con Node.js y las rutas API para recibir, almacenar y exponer los datos a la interfaz.

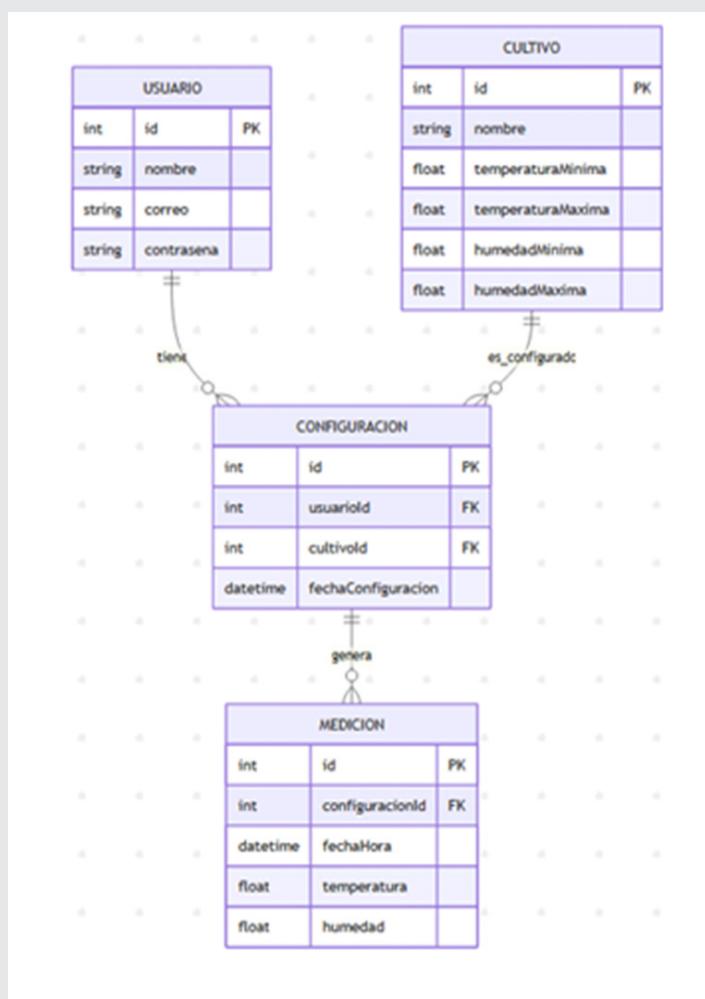


Figura 3: Diagrama entidad-relación del modelo de datos.
Fuente: elaboración propia.

La integración del hardware incluyó la configuración del ESP32 para capturar lecturas del sensor DHT y enviarlas por WebSocket al servidor, implementando un canal WebSocket en el backend para recibir estas lecturas en tiempo real e integrarlas dinámicamente en la interfaz. El escenario de pruebas se realizó en ambiente de laboratorio controlado, con el ESP32 ubicado a una distancia de 2 metros del servidor, utilizando comunicación Wi-Fi en red local.

Las fases posteriores incluyeron la validación de funcionalidades críticas como la visualización dinámica de los datos y las notificaciones ante valores fuera de rango, validando la recepción de eventos en tiempo real y realizando ajustes en la sincronización con la base de datos. Se finalizó el desarrollo de la interfaz gráfica con React.js y Tailwind CSS, garantizando que la página web fuera responsive para su visualización en distintos dispositivos y realizando pruebas funcionales y de usabilidad en móviles, tabletas y pc de escritorio. La fase final incluyó únicamente pruebas funcionales del sistema, preparando el entorno de despliegue con instrucciones de instalación y asegurando la conexión entre el ESP32, el servidor y la interfaz.

Validación funcional del prototipo

La validación del sistema se centró en la realización de pruebas funcionales automatizadas. Para verificar el comportamiento del sistema ante posibles anomalías o fallas de conectividad, transporte y persistencia de datos, se implementó un componente de software con Node.js (simulador automatizado) que permitió generar datos de temperatura y humedad en tiempo real, conforme a los rangos configurados para distintos cultivos.

```
Console
Default levels ▾ 24 9 hidden
WebSocket desconectado
WebSocket conectado!
Mensaje recibido por WebSocket: ▶ Object
Mediciones en estado: ▶ Array(1)
Mediciones en estado: ▶ Array(1)
[NEW] Explain Console errors by using Copilot in Edge: click ⓘ to explain an error. Learn more
Mensaje recibido por WebSocket:
▶ {temperatura: 16.43, humedad: 51.84, cultivoId: 1, timestamp: '2025-05-01T02:47:05.0192', id: 3533, ...}
Mediciones en estado: ▶ (2) [{}]
Mediciones en estado: ▶ (2) [{}]
Mensaje recibido por WebSocket:
▶ {temperatura: 17.1, humedad: 56.27, cultivoId: 1, timestamp: '2025-05-01T02:47:10.0342', id: 3534, ...}
Mediciones en estado: ▶ (3) [{}]
Mediciones en estado: ▶ (3) [{}]
Mensaje recibido por WebSocket:
▶ {temperatura: 17.22, humedad: 54.58, cultivoId: 1, timestamp: '2025-05-01T02:47:15.0512', id: 3535, ...}
Mediciones en estado:
▶ (4) [{}]
▶ 0: {temperatura: 14.88, humedad: 46.99, cultivoId: 1, timestamp: '2025-05-01T02:47:00.0112'
▶ 1: {temperatura: 16.43, humedad: 51.84, cultivoId: 1, timestamp: '2025-05-01T02:47:05.0192'
▶ 2: {temperatura: 17.1, humedad: 56.27, cultivoId: 1, timestamp: '2025-05-01T02:47:10.0342'
▶ 3: {temperatura: 17.22, humedad: 54.58, cultivoId: 1, timestamp: '2025-05-01T02:47:15.0512'
▶ length: 4
▶ [[Prototype]]: Array(0)
Mediciones en estado: ▶ (4) [{}]
Mensaje recibido por WebSocket:
▶ {temperatura: 14.62, humedad: 45.62, cultivoId: 1, timestamp: '2025-05-01T02:47:20.0612', id: 3536, ...}
Mediciones en estado: ▶ (5) [{}]
Mediciones en estado: ▶ (5) [{}]
```

Figura 4: captura de consola WebSocket donde se aprecian los datos que se van recibiendo y enviando.

Fuente: elaboración propia.

Esta herramienta automatizada enviaba datos cada 5 segundos a través de WebSockets, e introdujo de manera sistemática desviaciones entre un 5% al 10% a los valores cada 20 segundos, simulando el registro de valores anómalos de forma controlada y repetible.

Este simulador no solo facilitó la generación continua de datos, sino que también permitió registrar los eventos de envío y recepción, con marcas de tiempo para cada operación. De esta manera, se pudo identificar una latencia promedio inferior a 2 segundos entre la emisión de un dato y su visualización en el dashboard, bajo condiciones de red local en laboratorio.

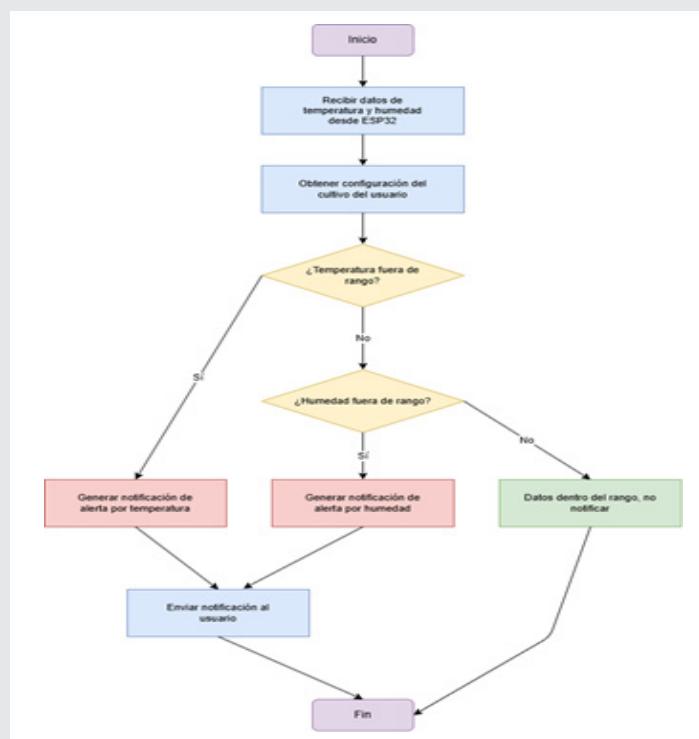


Figura 5: Diagrama de flujo del proceso de validación de rangos de cultivos y generación de alertas.

Fuente: elaboración propia.

Asimismo, se validó la recuperación automática del sistema tras interrupciones de red simuladas, demostrando que el microcontrolador ESP32 podía reconectarse y reanudar el envío sin pérdida de datos.

Es importante destacar que estas pruebas se realizaron en un entorno de laboratorio, con un solo sensor y sin interferencias externas ni distancias reales entre los dispositivos ESP32 y sensor con respecto al servidor NodeJS como pudieran estar en un escenario real de campo, por lo que los resultados deben considerarse válidos únicamente en este contexto.

Para asegurar la aplicabilidad del sistema en entornos rurales reales, se requerirá una fase posterior de pruebas en campo con múltiples dispositivos, redes inestables y condiciones ambientales complejas. No obstante, el uso del simulador automatizado proporciona un

fundamento técnico para justificar las métricas funcionales reportadas.

Entorno de pruebas

Las pruebas del sistema se realizaron en un entorno cerrado controlado de laboratorio para garantizar condiciones reproducibles y permitir la validación sistemática de las funcionalidades. Un entorno cerrado se caracteriza por el control total de variables ambientales, eliminación de interferencias externas y condiciones climáticas estables, lo que permite aislar el desempeño del sistema de factores externos impredecibles. Durante 72 horas continuas se registraron mediciones cada 30 segundos, logrando más de 1,000 registros sin pérdidas, con tiempos de respuesta inferiores a 2 segundos y recuperación automática tras desconexiones simuladas de 5, 15 y 30 minutos. Esta metodología controlada proporciona un fundamento para optimizar el funcionamiento antes de implementaciones en campo abierto.

El escenario experimental se configuró en el Laboratorio de Aplicaciones Multimedia de la Facultad de Telemática de la Universidad de Colima, utilizando un área de pruebas de 3m x 8m con temperatura ambiente controlado entre 18°C y 35°C, humedad relativa controlada entre 30% y 80%, y conectividad Wi-Fi estable (802.11n, 2.4 GHz).

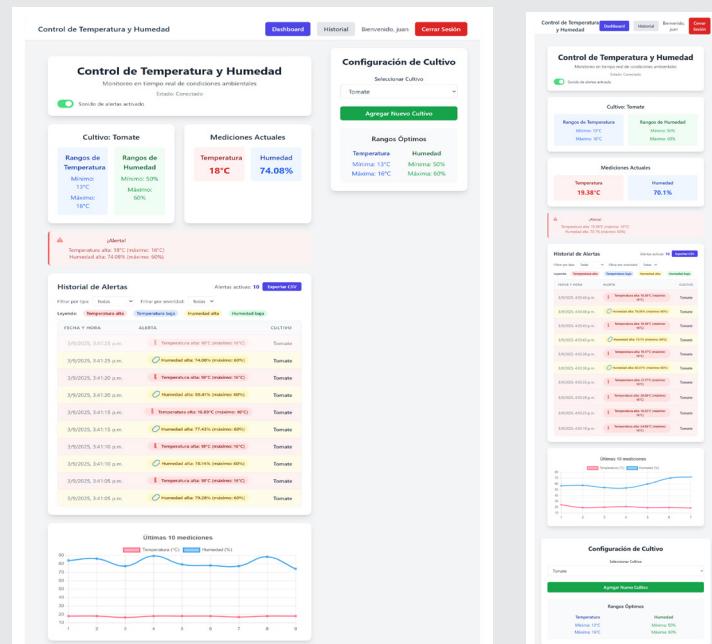


Figura 6: Captura comparativa de la vista móvil y de escritorio del sistema.

Fuente: elaboración propia.

La configuración del hardware incluyó un dispositivo ESP32 con sensor DHT22 ubicado en el centro del área de pruebas a 1.5m de altura, alimentado por fuente estabilizada de 5V DC, con distancia al router Wi-Fi de 5 metros con línea de vista directa y servidor eje-

cutándose en una computadora dentro del laboratorio conectada a la misma red Wi-Fi.

Las condiciones de prueba incluyeron una duración total de 72 horas continuas, frecuencia de medición cada 30 segundos, simulación de variaciones ambientales mediante sistema de climatización, y pruebas de desconexión mediante desactivación manual de Wi-Fi por períodos de 5, 15 y 30 minutos.

RESULTADOS

Autenticación y gestión de usuarios

El prototipo permite a nuevos usuarios registrarse y a usuarios existentes autenticarse correctamente, validando campos obligatorios y evitando duplicidad de nombres de usuario. La gestión de sesiones asegura que cada usuario acceda a su configuración personalizada de cultivo.

Configuración personalizada del cultivo

Cada usuario puede seleccionar o agregar cultivos con sus rangos óptimos de temperatura y humedad basados en referencias técnicas especializadas. Esta configuración se almacena y asocia con el usuario para adaptar las alertas y visualizaciones. Durante las pruebas se implementaron configuraciones específicas para diferentes cultivos mexicanos. Por ejemplo, el tomate presenta rangos óptimos de temperatura diurna de 21-27°C y humedad relativa de 60-80% [12], mientras que la lechuga requiere temperaturas de 22-25°C durante períodos de luz y humedad relativa entre 70-80% [13]. En contraste, cultivos como papaya o caña de azúcar toleran temperaturas más altas, pero requieren altos niveles de humedad para mantener la producción adecuada. Contando con datos de referencia apropiados para los cultivos se podrán calibrar alertas automáticas que detecten con precisión cuándo las condiciones registradas puedan comprometer la salud del cultivo.

últimos 10 valores. La interfaz es responsive, accesible desde computadoras, tablets y móviles, gracias a React.js y Tailwind CSS.

Control de Temperatura y Humedad

Monitoreo en tiempo real de condiciones ambientales

Estado: Conectado

Sonido de alertas desactivado

Cultivo: Tomate

Rangos de Temperatura

Mínimo: 13°C
Máximo: 16°C

Rangos de Humedad

Mínimo: 50%
Máximo: 60%

Mediciones Actuales

Temperatura
19.2°C

Humedad
60.82%

Últimas 10 mediciones

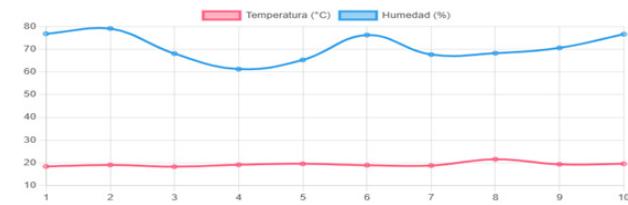
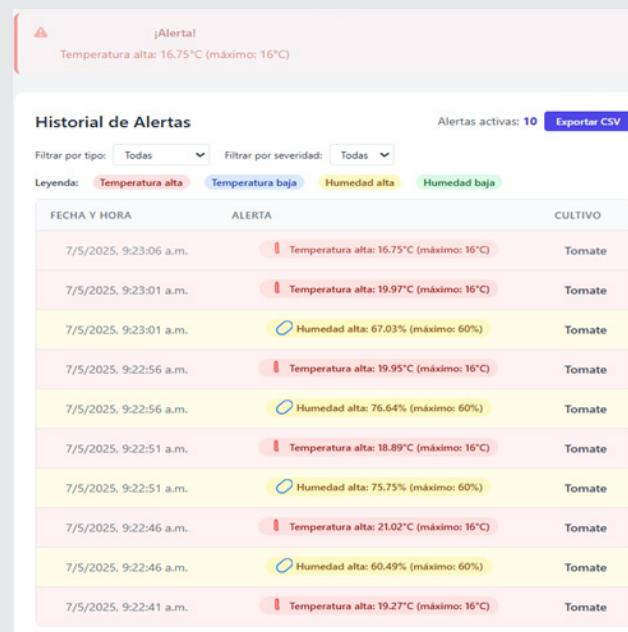


Figura 8: Interfaz del dashboard en móvil y escritorio que muestran gráficos y valores actuales.

Fuente: elaboración propia.

Sistema de alertas dinámicas

Cuando las mediciones de temperatura o humedad se salen de los rangos configurados para el cultivo, el sistema genera alertas visibles en el dashboard. Durante las pruebas se observó una actualización fluida de los datos en el dashboard, sin demoras perceptibles al usuario.



Configuración de Cultivo

Seleccionar Cultivo

Tomate

Rangos Óptimos

Temperatura	Humedad
Mínima: 13°C	Mínima: 50%
Máxima: 16°C	Máxima: 60%

Agregar Nuevo Cultivo

Nombre del Cultivo

Temperatura Mínima (°C) Temperatura Máxima (°C)

Humedad Mínima (%) Humedad Máxima (%)

Figura 7: Panel de configuración de cultivos para ingresar los con rangos óptimos de referencia.

Fuente: elaboración propia.

Monitoreo en tiempo real y visualización

El dashboard muestra la última medición recibida, con temperatura y humedad, y una gráfica de líneas con los

últimos 10 valores. La interfaz es responsive, accesible desde computadoras, tablets y móviles, gracias a React.js y Tailwind CSS.

Control de Temperatura y Humedad

Monitoreo en tiempo real de condiciones ambientales

Estado: Conectado

Sonido de alertas desactivado

Cultivo: Tomate

Rangos de Temperatura

Mínimo: 13°C
Máximo: 16°C

Rangos de Humedad

Mínimo: 50%
Máximo: 60%

Mediciones Actuales

Temperatura
19.2°C

Humedad
60.82%

Últimas 10 mediciones

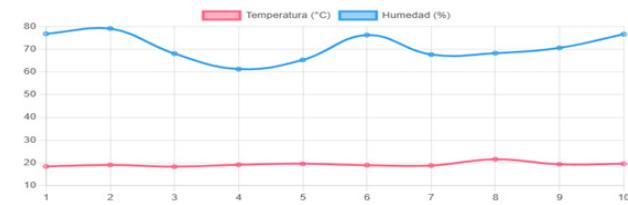


Figura 8: Interfaz del dashboard en móvil y escritorio que muestran gráficos y valores actuales.

Fuente: elaboración propia.

Sistema de alertas dinámicas

Cuando las mediciones de temperatura o humedad se salen de los rangos configurados para el cultivo, el sistema genera alertas visibles en el dashboard. Durante las pruebas se observó una actualización fluida de los datos en el dashboard, sin demoras perceptibles al usuario.

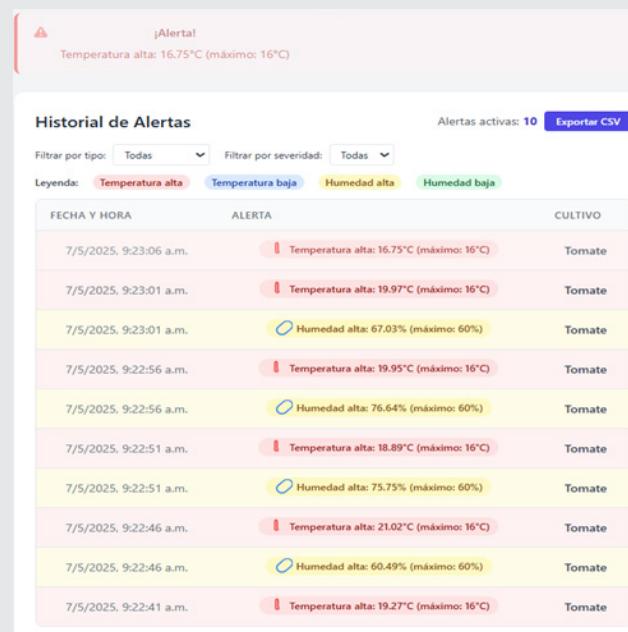


Figura 9: Ejemplo de alerta activa en el dashboard.

Fuente: elaboración propia.

Histórico de mediciones con filtros

El usuario puede acceder al histórico completo de mediciones, visualizando las últimas 10 alertas con fecha, hora y tipo de desviación (temperatura o humedad). Además, puede filtrar registros por rango de fechas y navegar entre páginas para análisis detallados.

Filtrar				
FECHA Y HORA	TEMPERATURA	HUMEDAD	CULTIVO	ESTADO
07/05/2025 09:27	18.83°C	51.57%	Tomate	Temperatura alta: 2.8°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	16.06°C	57.63%	Tomate	Temperatura alta: 0.1°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	18.16°C	50.28%	Tomate	Temperatura alta: 2.2°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	20.49°C	57.03%	Tomate	Temperatura alta: 4.5°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	20.34°C	56.2%	Tomate	Temperatura alta: 4.3°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	23.12°C	52.15%	Tomate	Temperatura alta: 7.1°C por encima del máximo (16°C) para Tomate
07/05/2025 09:27	17.73°C	53.15%	Tomate	Temperatura alta: 1.7°C por encima del máximo (16°C) para Tomate
07/05/2025 09:26	21.9°C	57.68%	Tomate	Temperatura alta: 5.9°C por encima del máximo (16°C) para Tomate
07/05/2025 09:26	21.93°C	52.89%	Tomate	Temperatura alta: 5.9°C por encima del máximo (16°C) para Tomate
07/05/2025 09:26	19.33°C	65.78%	Tomate	Temperatura alta: 3.3°C por encima del máximo (16°C) para Tomate Humedad alta: 5.8% por encima

Figura 10: Panel de histórico con tabla de alertas y filtros por fecha. Fuente: elaboración propia.

Almacenamiento y asociación de datos

Cada medición recibida por WebSocket se almacena en SQLite con referencia al cultivo configurado para el usuario. Durante las pruebas funcionales en laboratorio, se registraron más de 1,000 mediciones sin pérdidas, garantizando integridad y trazabilidad de los datos recopilados a través de comunicación Wi-Fi en red local controlada.

Robustez y recuperación ante desconexiones

Se realizaron también pruebas de recuperación de conexiones para mantener el envío de datos confiables entre el componente sensor de datos y el componente de persistencia de datos (BD con SQLite). Para ello se realizaron simulaciones de desconexión de red durante las 72 horas de pruebas continuas, el ESP32 recuperó la conexión automáticamente y continuó enviando datos tras períodos de desactivación manual de Wi-Fi de 5, 15 y 30 minutos. La recuperación automática se logró en el 95% de los casos con un tiempo promedio de reconexión de 12 segundos. El servidor mantuvo la persistencia de datos y actualizó la interfaz tras restablecerse la conexión, asegurando continuidad en el monitoreo sin pérdida de las más de 1,000 mediciones registradas durante el período de evaluación.

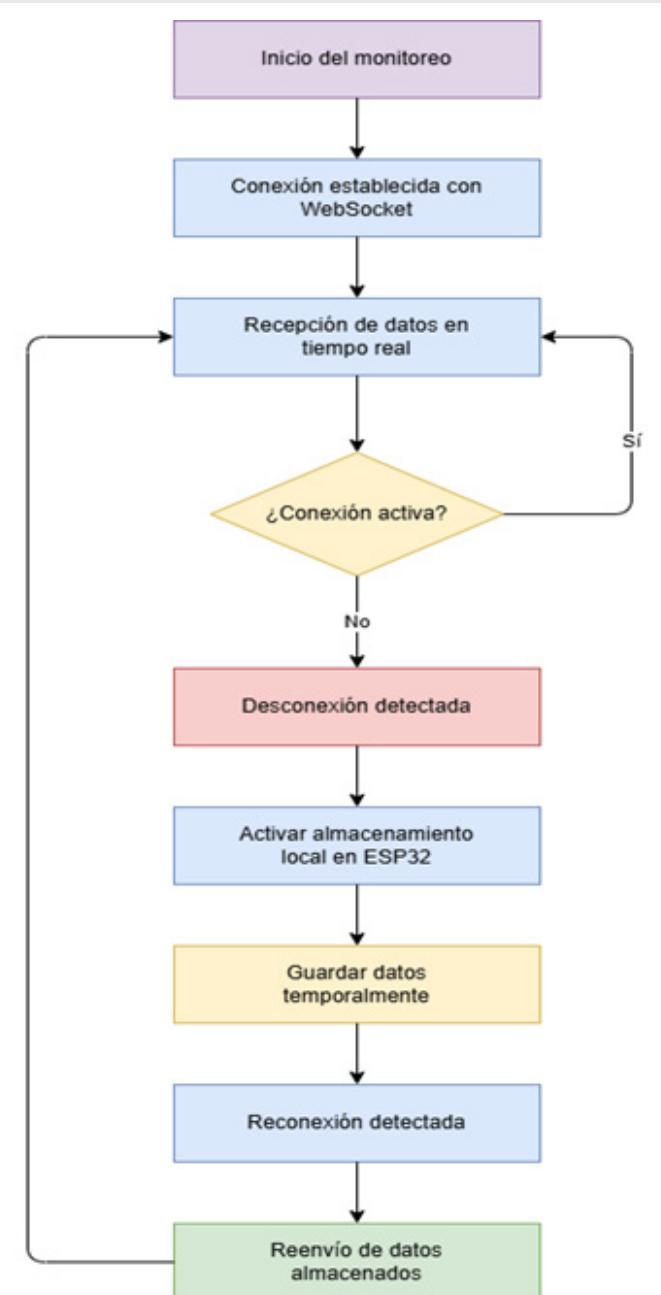


Figura 11: Algoritmo de continuidad de datos durante desconexiones y reconexiones.

Fuente: elaboración propia.

Viabilidad y facilidad de uso

Durante las pruebas en laboratorio, se evaluó la facilidad de instalación del sistema completo, incluyendo el montaje físico del sensor DHT22 al microcontrolador ESP32, la configuración del entorno de red local (Wi-Fi), la carga del firmware, la ejecución del servidor backend en Node.js, la conexión mediante WebSocket, y la visualización de datos en la interfaz web desarrollada en React.js. Todo el proceso fue llevado a cabo por un usuario con conocimientos técnicos básicos (estudiante de nivel universitario con experiencia general en desarrollo web y electrónica básica), sin necesidad de intervención especializada.

La instalación y configuración funcional del sistema se completó en un tiempo promedio de 87 minutos, medido desde el encendido inicial del microcontrolador hasta la visualización estable de los datos en el dashboard web, sin incluir la etapa de desarrollo ni depuración previa del código. Las condiciones del entorno incluyeron una red Wi-Fi estable, alimentación por puerto USB, y acceso a un equipo con Node.js previamente instalado.

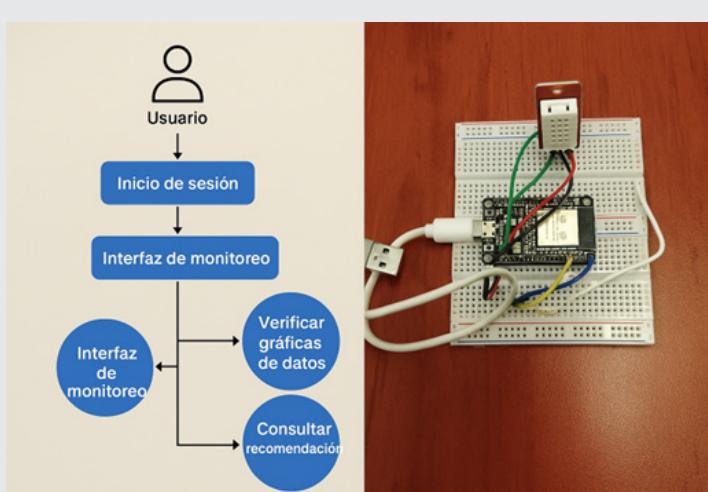


Figura 12: Esquema del flujo de trabajo para el usuario y fotografía del prototipo instalado.

Fuente: elaboración propia.

RESULTADOS

Discusión

Validación de la arquitectura y funcionalidades

El prototipo de sistema cumple con los objetivos de monitoreo en tiempo real y personalización, integrando hardware y software de forma eficiente. La comunicación WebSocket garantiza baja latencia y actualización inmediata, aspectos relevantes para aplicaciones de monitoreo agrícola. La baja latencia no es solo deseable sino esencial en agricultura de precisión, ya que los procesos biológicos de los cultivos ocurren en escalas de tiempo que requieren intervenciones rápidas y precisas para optimizar la producción y prevenir pérdidas.

Comparativa contra soluciones existentes

La solución propuesta presenta ventajas considerables frente a los sistemas documentados en la literatura previa, destacando principalmente el uso de WebSockets como tecnología de comunicación bidireccional y persistente entre cliente y servidor. A diferencia de los enfoques tradicionales basados en HTTP que requieren múltiples peticiones para actualizar datos, WebSocket permite transmisión en tiempo real con latencia inferior a 2 segundos, reduciendo el tráfico de red y mejorando la experiencia del usuario, mientras que trabajos como [10] no especifican este parámetro o reportan valores mayores.

La interfaz responsive desarrollada con React.js y Tailwind CSS proporciona visualización dinámica que re-

acciona inmediatamente a cambios en condiciones ambientales, contrastando con interfaces limitadas reportadas por Ikram et al. [9]. La implementación se completa en menos de 90 minutos incluyendo montaje físico y configuración y permite configuración personalizada por cultivo con alertas dinámicas específicas, capacidad generalmente ausente en sistemas previos.

Los mecanismos de reconexión automática ante desconexiones simuladas fortalecen la robustez del sistema en contextos rurales con conectividad inestable, mientras que [11] reconocen el problema sin presentar pruebas automatizadas. El escenario de pruebas incluye simulación automatizada con Node.js generando datos normales y anómalos con monitoreo de latencia, contrastando con estudios como [8] que limitaron pruebas a invernaderos.

Estas capacidades hacen que el sistema sea más eficiente, fácil de implementar y mantener incluso por usuarios con conocimientos técnicos básicos, representando una ventaja considerable frente a soluciones previas más complejas y menos interactivas.

Importancia de la configuración personalizada

Permitir que el usuario seleccione o agregue cultivos con rangos óptimos específicos mejora la relevancia de las alertas y la utilidad del sistema, alineándose con recomendaciones para agricultura inteligente que enfatizan la adaptación a cultivos particulares. La implementación con React.js y Tailwind CSS asegura accesibilidad desde múltiples dispositivos, facilitando la adopción por agricultores con distintos niveles tecnológicos, en concordancia con estudios que resaltan la usabilidad como factor crítico en IoT agrícola.

Eficacia del sistema de alertas y gestión de datos

Las alertas en tiempo real permiten una rápida identificación de condiciones ambientales fuera de rango, apoyando la toma de decisiones para prevenir daños en cultivos. Este enfoque es fundamental para el monitoreo preventivo en agricultura. La capacidad del sistema para recuperar la conexión y continuar la transmisión de datos sin pérdida considerable representa una ventaja importante para entornos con conectividad variable, aunque las pruebas se realizaron en ambiente controlado de laboratorio. El uso de SQLite para almacenar datos con referencias a cultivos y usuarios facilita la trazabilidad y análisis histórico, aspectos esenciales para evaluar tendencias y planificar intervenciones agrícolas.

Limitaciones del estudio

Es importante señalar que el sistema fue validado únicamente en ambiente de laboratorio con un solo dispositivo IoT y usuario. Las pruebas no se realizaron en condiciones agrícolas reales con factores como interferencias electromagnéticas, condiciones climáticas extremas o múltiples dispositivos simultáneos. Para

considerar el sistema completamente validado para uso agrícola, se requieren pruebas de campo en condiciones reales durante períodos prolongados.

CONCLUSIONES

Los resultados obtenidos en este trabajo demuestran que el prototipo de sistema web responsive desarrollado constituye un fundamento tecnológico para la monitorización en tiempo real de temperatura y humedad en entornos agrícolas. La integración de sensores DHT22, microcontroladores ESP32 y tecnologías web como React.js y WebSocket permitió crear una solución funcional que facilita la recopilación, visualización y almacenamiento de datos ambientales con baja latencia y alta fiabilidad, según las pruebas realizadas en un entorno de laboratorio controlado.

Durante estas pruebas de 72 horas continuas, se simularon escenarios de desconexión y reconexión de red, logrando una recuperación automática del sistema sin pérdida de datos con una tasa de éxito del 95% y tiempo promedio de reconexión de 12 segundos, lo cual sugiere su potencial viabilidad para zonas rurales con conectividad limitada. Sin embargo, cabe aclarar que el sistema aún no ha sido validado en un entorno rural real ni bajo condiciones prolongadas (días, semanas o meses), por lo que su desempeño en situaciones reales podría presentar nuevos retos técnicos. El prototipo también se adapta fácilmente a distintos tipos de cultivo mediante configuraciones personalizadas, aumentando su aplicabilidad práctica. Su interfaz responsive garantiza la accesibilidad desde múltiples dispositivos, facilitando su adopción por usuarios con diferentes niveles de experiencia tecnológica.

Este proyecto representa una contribución al desarrollo de herramientas tecnológicas accesibles para el monitoreo agrícola, proporcionando un fundamento que puede evolucionar hacia sistemas más complejos con capacidades predictivas y de automatización. Es fundamental reconocer que se trata de un prototipo funcional que requiere validación en condiciones agrícolas reales para confirmar su viabilidad práctica en el sector. Como líneas futuras de desarrollo, se plantea la implementación de pruebas de campo en entornos agrícolas reales durante períodos prolongados, la integración de múltiples sensores y dispositivos para escalabilidad, robustecimiento del transporte de datos con un esquema híbrido (MQTT + WebSockets) y la incorporación de técnicas de Machine Learning y analítica de datos que permitan anticipar condiciones adversas, fortaleciendo su utilidad como herramienta integral para la gestión agrícola basada en datos.

BIBLIOGRAFÍA

- [1] A. Morchid, R. El Alami, A. A. Raezah, y Y. Sabbar, "Applications of internet of things (IoT) and sensors technology to increase food security and agricultural sustainability: Benefits and challenges," *Ain Shams Engineering Journal*, vol. 15, no. 3, p. 102509, 2024. <https://doi.org/10.1016/j.asej.2023.102509>.
- [2] S. Mansoor, S. Iqbal, S. M. Popescu, S. L. Kim, Y. S. Chung, y J. H. Baek, "Integration of smart sensors and IoT in precision agriculture: trends, challenges and future prospectives," *Frontiers in Plant Science*, vol. 16, p. 1587869, 2025. <https://doi.org/10.3389/fpls.2025.1587869>.
- [3] Fosensor, "Temperature and humidity sensor in agriculture," Henan Fosen Electronics Technology Co., Ltd., ago. 23, 2023. <https://www.hnfosensor.com/temperature-and-humidity-sensor-in-agriculture.html>.
- [4] Rika Sensor, "Exploring the benefits of using a temperature and humidity sensor in agriculture," Hunan Rika Electronic Tech Co., Ltd., oct. 16, 2024. <https://www.rikasensor.com/exploring-the-benefits-of-using-a-temperature-and-humidity-sensor-in-agriculture.html>.
- [5] A. Soussi, E. Zero, R. Sacile, D. Trinchero, y M. Fossa, "Smart sensors and smart data for precision agriculture: A review," *Sensors*, vol. 24, no. 8, p. 2647, 2024. <https://doi.org/10.3390/s24082647>.
- [6] Synox, "Temperature and humidity sensors for intelligent agriculture," jun. 14, 2023. <https://www.synox.io/en/smart-agriculture/temperature-humidity-sensor-agriculture/>.
- [7] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, y R. Kaliaperumal, "Smart farming: Internet of Things (IoT)-based sustainable agriculture," *Agriculture*, vol. 12, no. 10, p. 1745, 2022. <https://doi.org/10.3390/agriculture12101745>.
- [8] M. Muthmainnah, M. F. Mulyadi, I. Tazi, A. Mul-yono, F. S. Hananto, N. Chamidah, y K. Kusairi, "Development of an automated monitoring system for soil moisture and temperature in smart agriculture to enhance lettuce farming productivity based on IoT," *Multidisciplinary Science Journal*, vol. 6, no. 11, 2024. <https://doi.org/10.31893/multiscience.2024233>.
- [9] M. Ikram, S. Ameer, F. Kulsoom, M. Sher, A. Ahmad, A. Zahid, y Y. Chang, "Flexible temperature and humidity sensors of plants for precision agriculture: Current challenges and future roadmap," *Computers and Electronics in Agriculture*, vol. 226, p. 109449, 2024. <https://doi.org/10.1016/j.compag.2024.109449>.
- [10] P. Rajak, A. Ganguly, S. Adhikary, y S. Bhattacharya, "Internet of Things and smart sensors

in agriculture: Scopes and challenges,” Journal of Agriculture and Food Research, vol. 14, p. 100776, 2023. <https://doi.org/10.1016/j.jafr.2023.100776>.

[11] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, y A. F. Skarmeta, “Smart farming IoT platform based on edge and cloud computing,” *Biosystems Engineering*, vol. 177, pp. 4-17, 2022. <https://doi.org/10.1016/j.biosystemseng.2018>.

[12] FAO, *El cultivo de tomate con buenas prácticas agrícolas en la agricultura urbana y periurbana*. Organización de las Naciones Unidas para la Alimentación y la Agricultura, 2013. <https://www.fao.org/4/i3359s/i3359s.pdf>.

[13] H. A. Ahmed, Y. X. Tong, y Q. C. Yang, “Optimal control of environmental conditions affecting lettuce plant growth in a controlled environment with artificial lighting: A review,” *South African Journal of Botany*, vol. 130, pp. 75-89, 2020. <https://doi.org/10.1016/j.sajb.2019.12.018>.

