# Development of a Web Platform for Precision Agriculture Optimization

**Jorge Alberto Cárdenas Magaña[1,*], Marco Antonio Celis Crisóstomo[2], Emmanuel Vega Negrete[1], Francisco Miguel Hernández López[1], Juan Pablo Mojica Sanchez[2]**
[1]Department of Electromechanical Engineering, Tecnológico Nacional de México/ Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Tamazula, México,
[2]Department of Agricultural Innovation Engineering, Tecnológico Nacional de México/ Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Tamazula, México,
*Emails: [1]jorge.cardenas@tamazula.tecmm.edu.mx, [2]marco.celis@tamazula.tecmm.edu.mx, [3]emmanuel.vega@tamazula.tecmm.edu.mx, [4]francisco.hernandez@tamazula.tecmm.edu.mx, [5]juan.mojica@tamazula.tecmm.edu.mx*
*[*] Corresponding author*

***Abstract:*** **This study presents the development of a low-cost web-based platform designed to improve precision agriculture in the sugarcane sector through real-time monitoring of key environmental variables. The proposed system aims to address the technological gap in rural agricultural settings by integrating accessible and open-source technologies. The methodology involved the use of temperature, humidity, and rainfall sensors connected to an Arduino microcontroller, which transmits the collected data via HTTP to a MySQL database. A web interface, developed using HTML, CSS, and PHP, enables users to remotely visualize environmental conditions through graphical and numerical formats. The platform was implemented and tested in a sugarcane cultivation environment to evaluate its performance in terms of responsiveness, stability, and usability. Results from field testing indicated a latency of less than three seconds in data transmission, with an automatic update frequency of approximately one minute. Additionally, a significant reduction in data retrieval time was observed, along with improved accuracy in decision-making due to the real-time visualization of critical parameters. The entire solution was developed with cost efficiency in mind, reaching an estimated implementation cost of approximately $500, significantly lower than traditional commercial systems. The proposed platform not only automates data acquisition and visualization but also lays the foundation for future integration with artificial intelligence algorithms to enable predictive analysis and advanced decision support. This system offers a scalable, adaptable, and affordable tool to support small and medium-sized producers in rural areas.**

***Keywords:*** **Arduino, Embedded Systems, Environmental Monitoring, Internet of Things, Precision Agriculture**

## I. INTRODUCTION

The Internet of Things (IoT) has transformed multiple sectors, including precision agriculture, by enabling real-time monitoring of critical environmental variables. The integration of smart sensors, web platforms, and communication technologies has optimized the use of resources such as water, fertilizers, and pesticides, thereby improving the sustainability and efficiency of agricultural practices [1]. The agricultural sector currently faces major challenges related to climate change, water scarcity, and the urgent need to maximize food production efficiently. Traditionally, environmental data collection was performed manually, resulting in delays and potential errors in decision-making. With the integration of IoT systems, farmers can now access real-time information on weather conditions, soil moisture, and crop status, facilitating data-driven decision-making [2].

Within IoT architecture applied to agriculture, there are two fundamental layers: the perception layer and the network layer. The perception layer includes sensors that collect data on temperature, soil moisture, and light intensity, while the network layer utilizes communication technologies such as LoRa, Xbee, and Sigfox to transmit the data to processing platforms. Among these technologies, LoRa has been widely adopted in agriculture due to its low energy consumption and long-range capabilities, making it suitable for rural environments [3]. However, the effective integration of web platforms for monitoring and data analysis remains a challenge. Real-time data visualization and efficient data storage require optimized solutions that allow farmers to remotely access and manage environmental information [4]. This study addresses the optimization and integration of web platforms for monitoring environmental variables in agriculture, exploring various technological solutions and their impact on the sector's efficiency and sustainability.

Despite advances in the development of monitoring systems for precision agriculture, many existing solutions are costly, require advanced technical skills, or depend on network infrastructure that is not always available in rural areas. Furthermore, most research has focused on data transmission without fully addressing remote visualization and accessible data management. In this context, the present work proposes a low-cost solution based on open technologies that integrates real-time acquisition, transmission, storage, and web visualization of environmental data, specifically tailored to the needs of the sugarcane sector in rural areas. This proposal seeks to bridge the gap between technological availability and the operational capacity of small-scale producers, thereby contributing to the effective adoption of precision agriculture.

DOI: 10.53375/ijecer.2025.476

Accordingly, the objective of this study is to design and implement a low-cost web platform for real-time monitoring of environmental variables, adapted to the conditions of the sugarcane sector in rural zones, in order to enhance agricultural decision-making efficiency through accessible IoT technologies.

### *A. State of the Art*

#### *A.1. Existing Platforms for Precision Agriculture*

One of the main challenges in implementing IoT in agriculture is the efficient collection, transmission, and analysis of acquired data. The optimization of web integration for environmental monitoring relies on the efficiency of real-time data transmission and its accurate visualization. Web platforms play a crucial role by allowing farmers to access and analyze information remotely, thereby supporting informed decision-making [5]. A relevant system [6] integrates humidity and temperature sensors (SHT31 and DHT22) into a waterproof, dustproof wireless node with 179-hour battery autonomy. The node communicates via Xbee modules with an ESP32-based coordinator, transmitting data to the cloud through the ThingSpeak platform for processing. The solution incorporates dual interfaces: a local interface for offline monitoring and a web interface for remote access and real-time analysis.

Automation in precision agriculture improves crop management efficiency by enabling real-time data collection on humidity, temperature, and light intensity. These data, stored and processed on web platforms, facilitate the optimization of irrigation, fertilization, and pest control practices, reducing resource waste and maximizing agricultural yields [7]. The connectivity provided by these platforms allows access to information from any internet-enabled device, promoting efficient and sustainable crop management [8]. Implementation examples of these technologies include the Citrus Observatory in the Guadalquivir Valley, which employs smart sensors and digital platforms to monitor pests, weather conditions, and irrigation needs. This system enables the precise and sustainable application of phytosanitary treatments through intelligent traps equipped with artificial intelligence, optimizing pesticide use and improving agricultural productivity [9].

#### *A.2. Emerging Technologies in Agricultural IoT*

Wireless sensor networks (WSNs) integrated with IoT cloud systems enable precise monitoring of crop microclimatic conditions. As demonstrated in [10], an IoT-based real-time irrigation supervision system improves water-use efficiency and enhances agricultural sustainability. Moreover, the integration of emerging technologies such as LoRaWAN, Machine Learning (ML), and Augmented Reality (AR) has proven effective in predicting environmental conditions and improving decision-making efficiency in precision agriculture [11]. In terms of security and efficiency in agricultural data storage, the combination of IoT and blockchain ensures transparency and protection of information, allowing secure real-time access to data. This is essential for traceability in agricultural processes and for optimizing resource management [12]. Researchers have also analyzed how the integration of IoT and Artificial Intelligence (AI) can transform precision agriculture by enhancing decision-making and resource optimization. This convergence promises to address challenges such as climate change and food scarcity, promoting sustainability and improving productivity in the agricultural sector [13]. Recent studies have explored the convergence of the Internet of Everything (IoE) and molecular communication in precision agriculture. These emerging technologies offer new opportunities for efficient crop management and improved agricultural sustainability [14].

#### *A.3. Low-Cost Solutions and Feasibility in Rural Areas*

Another important aspect in optimizing agricultural monitoring is the development of low-cost IoT systems. Recent studies highlight the implementation of affordable LoRa-based solutions for crop monitoring, allowing the remote collection of environmental data with low energy requirements. These approaches represent a viable alternative for small and medium-sized agricultural producers [15]. [16] investigates IoT-integrated precision agriculture monitoring systems, demonstrating significant improvements in irrigation efficiency and resource optimization. Likewise, the development of LoRa-based monitoring solutions and their application in agriculture have proven to be essential tools for improving productivity and reducing operational costs [17]. With the continuous advancement of these technologies, web integration for environmental variable monitoring in the agricultural sector is emerging as a key tool to enhance efficiency, sustainability, and resilience in the face of current and future challenges.

### *B. Technical Justification of the Proposed System*

#### *B.1. Technologies Used for System Development*

**Development Platform: Arduino.** Arduino uses a programming language based on C++, making it a versatile platform for the development of automation and monitoring projects. Its ease of use and large community support have enabled the integration of multiple sensors and communication modules. The Arduino language is designed to simplify microcontroller programming by providing structures and functions that facilitate efficient hardware control [18].

**JSON (JavaScript Object Notation).** JSON is a text-based data format used to store and exchange data in a format that is both human-readable and easily parsed by machines. It is widely used in web and mobile development due to its simplicity, flexibility, and compatibility with various programming languages. Moreover, JSON enables information exchange between different platforms, making it an essential format for dynamic applications, system configurations, and NoSQL databases [19].

**Visual Studio Code.** Visual Studio Code (VS Code) was created to meet the need for a lightweight, fast, and highly customizable code editor compatible with various platforms.

It supports multiple programming languages, integrates with source code control tools, and offers powerful debugging features. Key features include IntelliSense for code completion, extensions for enhanced functionality, and seamless Git integration. The goal was to improve developer productivity while maintaining a simple and intuitive user experience [20].

### *B.2. Communication and Database*

Before analyzing how Arduino handles communication between a web browser and the web server it uses, it is important to understand the fundamentals of the Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML) [21]. HTTP is the mechanism that enables communication between web browsers and a server. When a user accesses a webpage through a browser, it sends a request to the server hosting the page, specifying the information it requires. In some cases, the request simply consists of retrieving the page content in HTML format. The web server remains on standby for such requests and processes them upon arrival. In a basic scenario, processing the request involves responding with the HTML code defined in the Arduino program [21].

To enable an Arduino device to send data directly to a MySQL database without using an intermediary such as PHP, the *MySQL Connector/Arduino* library can be employed. This library, developed by Dr. Charles Bell under the GPLv2 license, allows Arduino to efficiently communicate with a MySQL server, storing sensor data or any other type of information in the database [22]. To establish the connection, the following are required:

- A MySQL server, which can run on operating systems such as Linux, Windows, or Mac.
- The *MySQL Connector/Arduino* library installed in the Arduino development environment.
- An Arduino board—preferably an Arduino Uno or Arduino Mega—with Ethernet or Wi-Fi connectivity [22].

**Arduino Code.** In the Arduino code, the *MySQL Connector* library must be included, and the connection parameters—such as the server IP address and user credentials—must be configured. A basic example of establishing a connection to a MySQL database from Arduino is shown in Fig. 1:

```
#include < MysQL_Connector_Arduino.h>
#include <Ethernet.h>
byte mac{} = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; //Dirección ...
//... mac de la Ethernet Shield
IPAddress server(xxx, xxx, xxx); //Dirección IP del servidor MySQL
EthernetClient client;
MySQL_Connector_Arduino conn((Client *)&client);
void setup() {
  Ethernet.begin(mac);
  if (conn.connect(server, 3306, "arduino", "password")); {
    Serial.println("Conectado a MySQL");
  } else {
    Serial.println("Error de conexión");
  }
}
void loop() {
  // Aquí se pueden enviar datos a la base de datos:
}
```

Fig. 1. Basic example of MySQL connection from Arduino [22].

With this configuration, Arduino can connect directly to the MySQL server and send sensor data for storage and subsequent analysis [22].

### *B.3. Web Design*

The system's web interface was developed using HTML and CSS, supported by the Visual Studio Code development environment. The design was organized into a modular structure that separates functionalities visible to the end user (*Web Apps*) from administrative functions (*Web Admin*). This organization facilitates system maintenance and scalability. The *Web Apps* module enables users to view environmental variables recorded by the sensors in real time, displaying the data in both graphical and numerical formats. The graphs are dynamically generated from the MySQL database, with one-minute refresh intervals. The responsive design allows access from both mobile devices and desktop computers, improving accessibility.

On the other hand, the *Web Admin* module is focused on system control and management. It allows for the registration of new monitoring antennas, editing of operational parameters, and management of user profiles (client or administrator). These functions are protected by access credentials, ensuring data security. All pages were developed following usability and simplicity principles, using a sober color palette, readable typography, and hierarchical navigation structures. Figure 8 shows the system's initial view, including login, registration, and antenna monitoring options [23].

### *B.4. Tools for Design and Simulation*

EasyEDA is an online Electronic Design Automation (EDA) software that allows users to create, model, and simulate electronic circuits. It supports schematic capture, SPICE simulation for both analog and digital circuits, and PCB layout design. Users can share projects and generate Gerber files for PCB manufacturing. EasyEDA offers both free and paid memberships, with additional features available in the paid version. The tool is compatible with files from other EDA software such as Altium and KiCad. Launched in 2014, its popularity has grown due to its comprehensive and intuitive design tools [24].

Fig. 2. EasyEDA logo [24].

## II. METHODOLOGY

To develop the environmental variable monitoring system for the agricultural sector, a structured methodology was followed through several stages—from system design to its implementation and optimization. Fig. 3 illustrates the methodology implemented.


Fig. 3. Methodology

### 1. *Requirements Establishment*

The system requirements were established based on the specific needs of the agricultural sector. The environmental variables to be monitored—temperature, humidity, and precipitation—were identified, and sensors compatible with the Arduino platform were selected for data acquisition. An efficient communication system was designed to transmit the data collected by the station to a cloud-based storage solution.

### 2. *Sensor Selection and Configuration*

To measure the environmental variables, appropriate sensors were selected based on factors such as accuracy, low cost, energy consumption, and compatibility with the development platform and data transmission protocols. The selection was informed by a literature review that identified the most suitable sensors for the intended application. The selected sensors included the DHT11 for measuring temperature and humidity, the DS18B20 for high-precision temperature measurement, and the KY-035, a digital-output Hall Effect magnetic sensor. Each sensor was configured individually and subsequently connected to the Arduino board. Reading and calibration tests were performed to verify proper operation before integration. Table 1 presents the characteristics of the sensors used.

Although these sensors are widely applied in low-cost IoT projects, their long-term stability in harsh agricultural environments may be affected by factors such as humidity, dust, and prolonged solar exposure. In this regard, the DHT11, while economical, may experience calibration drift in high-humidity conditions; the DS18B20 offers greater stability and precision but requires adequate waterproofing; and the KY-035 rain gauge sensor depends on the mechanical reliability of the tipping bucket mechanism. To mitigate these risks, periodic recalibration, protective housings (IP65-rated enclosures), and sensor redundancy are recommended to ensure data reliability over extended field deployments.

TABLE I. SELECTION AND CONFIGURATION OF SENSORS

| Sensor | Measured Variable | Operating Range | Accuracy | Signal Type |
|---|---|---|---|---|
| DHT11 | Temperature and humidity | 0 to 50 °C/ 20–80 % RH | ±2 °C / ±5 % HR | Digital |
| DS18B20 | Temperature | –10 °C to 85 °C | ±0.5 °C | Digital (1-Wire) |
| KY-035 | Precipitation (Hall Effect) | Depends on tipping bucket | Variable (per pulse) | Digital |

### 3. *Communication System Design*

Communication between the Arduino with the Wi-Fi Shield and the MySQL database was established using the JSON (JavaScript Object Notation) format for organizing and handling the data before storage. In order to enable data transmission and establish communication between the Arduino and the database, it is necessary to define the IP address of the server hosting the MySQL service. In a local network, the IP addresses of all devices (server, Arduino, router) must fall within the same range. To identify the IP address of the server where the database will be hosted, the command prompt (CMD) in Windows is used, executing the ipconfig command as shown in Fig. 4. This command displays the IP address of the machine running the database.

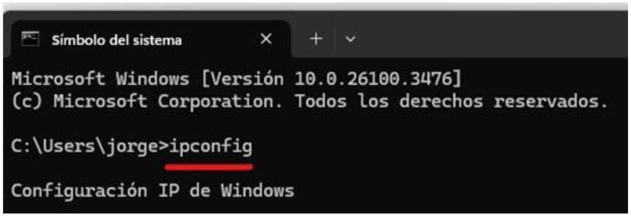
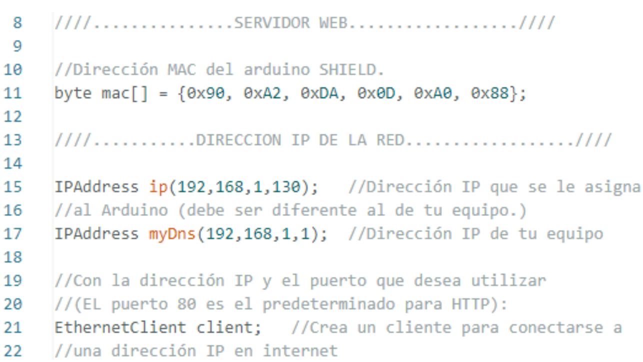



Fig. 4. Ipconfig command output.

To configure the connection in the Arduino code, the Wi-Fi Shield is programmed with the network credentials (SSID and password). The IP address of the MySQL server is defined within the Arduino code, and the connection is established to send data in JSON format. For data visualization on the web page, once the data is stored in the database, it can be retrieved and displayed through a web interface. The web server accesses the MySQL database and presents the information in a user-friendly format. Fig. 5 shows the assignment of MAC addresses for the Wi-Fi Shield and the corresponding IP addresses.



```
////...............SERVIDOR WEB..................////

//Dirección MAC del arduino SHIELD.
byte mac[] = {0x90, 0xA2, 0xDA, 0x0D, 0xA0, 0x88};

////..........DIRECCION IP DE LA RED.................////

IPAddress ip(192,168,1,130);    //Dirección IP que se le asigna
//al Arduino (debe ser diferente al de tu equipo.)
IPAddress myDns(192,168,1,1);  //Dirección IP de tu equipo

//Con la dirección IP y el puerto que desea utilizar
//(EL puerto 80 es el predeterminado para HTTP):
EthernetClient client;    //Crea un cliente para conectarse a
//una dirección IP en internet
```

Fig. 5. Address assignment in Arduino. Note: Comments in the code are shown in Spanish due to hardware limitations during capture.

### 4. Data Acquisition

For data acquisition in this project, the DHT11 sensor was used to measure temperature and relative humidity. This sensor operates within a range of 0°C to 50°C with an accuracy of ±2°C for temperature and from 20% to 80% RH with an accuracy of ±5% RH. Additionally, the DS18B20 digital temperature sensor was integrated; it offers high precision with a margin of error of ±0.5°C, over a range of –10°C to 85°C, making it suitable for humid environments or submersible applications. For precipitation measurement, a rain gauge based on the KY-035 Hall effect sensor was implemented. This sensor generates an electrical signal proportional to the collected water volume via a tipping bucket mechanism. All these sensors are connected to an Arduino microcontroller, which captures the raw signals (analog or digital) and prepares them for processing.

Once the raw signals are acquired, the system proceeds with sensor reading. This stage is performed by the Arduino microcontroller, which converts analog or digital signals into useful numerical values. Specific libraries are used for each sensor; for example, the DHT library ensures reliable and accurate readings of temperature and humidity. The DS18B20 communicates via the 1-Wire digital protocol, enabling precise temperature measurements. Meanwhile, the KY-035 Hall effect rain gauge produces electrical pulses that are counted by the microcontroller to determine the precipitation amount in millimeters. During this stage, verification routines are also implemented to discard faulty or out-of-range readings, ensuring data quality and reliability.

The data processing stage is essential to transform raw readings into actionable information. After acquisition and reading, the data undergo filtering to remove noise or outliers, using techniques such as signal smoothing or digital filtering. The processed data are then stored locally on an SD card or transmitted to a cloud platform for remote storage and real-time access. Basic analysis techniques, such as calculating averages, identifying trends, or detecting specific events like sudden rainfall increases, are also applied. Finally, the data are displayed on a graphical interface, such as a dashboard or mobile application, which presents the monitored environmental variables in a clear and user-friendly format, facilitating interpretation and decision-making.

### 5. Data Transmission to the Cloud

To carry out the data transmission stage from the Arduino, a function named enviarDatosPost() was created in the code. This function is responsible for retrieving sensor readings from the Arduino and sending them to a database via an HTTP connection. The IP address of the server and the target path in the database where the data will be stored are defined in the code. Subsequently, a JSON-formatted data packet is constructed and sent using an HTTP client. If the connection is successful, a confirmation message is printed; otherwise, an error message is displayed indicating a connection failure. Fig. 6 shows the declaration of this variable in the code.

```
void enviarDatosPost() {
  char HOST_NAME[] = "192.168.1.100";
  String PATH_NAME = "/antenas_be/public/Monitoreos/";

  String dataBody = "{ \"AntenaID\":\"";
  dataBody += AntenaID;
  dataBody += "\", \"SensorTemperatura\":\"";
  dataBody += sensorTemperatura;
  dataBody += "\", \"SensorHR\":\"";
  dataBody += sensorHr;
  dataBody += "\", \"Precipitacion\":\"";
  dataBody += caudalTotal;
  dataBody += "\" }";

  if (client.connect(HOST_NAME, 80)) {
    Serial.println();
    Serial.println("CONECTADO AL SERVIDOR");
    client.println("POST " + PATH_NAME + " HTTP/1.1");
    client.println("Host: " + String(HOST_NAME));
    client.println("Host: " + String(HOST_NAME));
    client.println("Content-Type: application/json");
    client.println("Connection: close");
    client.print("Content-Length: ");
    client.println(dataBody.length());
    client.println();
    client.println(dataBody);
    client.println();

    while (client.connected()) {
    if (client.available()) {
      //Serial.println(client);
      char c = client.read();
      Serial.print(c);
    }
  }

  // The server's disconnected, stop the client:
  client.stop();
  Serial.println();
  Serial.println("DESCONECTADO");
  Serial.println();   // SALTOS DE LINEA
} else {
  Serial.println();
  Serial.println("CONEXION FAILED");
  Serial.println();   // SALTOS DE LINEA
```

Fig. 6. Declaration of data transmission variable.

### 6. Arduino Software Development for Database Communication

The Arduino code developed for this project implements a monitoring system that collects data on temperature, humidity, and precipitation, and transmits it to a database via the Internet using an Ethernet Shield. To enable communication over the network, the code sets the MAC address and a static IP address. During the initialization phase, the system verifies the Ethernet connection and the presence of the network cable, ensuring that the hardware is properly connected. In the main loop, every 8 seconds, the function enviarDatosPost() is executed. This function establishes a connection with a server and sends the data using an HTTP POST request in JSON format. The request includes parameters such as the antenna identifier, temperature, and humidity values. If the connection to the server is successful, the Arduino transmits the data; otherwise, an error message is displayed on the serial monitor. This system enables continuous monitoring of environmental variables and stores the collected information in a database for subsequent analysis and remote visualization.

Fig. 7. Database communication in Arduino.

## 7. *Database Creation*

The database was developed using MySQL, an open-source management system, and includes the necessary tables to store data sent from both the Arduino and the web platform. Its structure consists of five tables: antennas, monitorings, sponsors, promotions, and users, which organize the information and facilitate real-time reading of the monitored variables. Fig. 8 shows the structure and content of the general view of the database.

| Tabla | Acción | | | | | |
|---|---|---|---|---|---|---|
| antenas | Examinar | Estructura | Buscar | Insertar | Vaciar | Eliminar |
| monitoreos | Examinar | Estructura | Buscar | Insertar | Vaciar | Eliminar |
| patrocinadores | Examinar | Estructura | Buscar | Insertar | Vaciar | Eliminar |
| promociones | Examinar | Estructura | Buscar | Insertar | Vaciar | Eliminar |
| usuarios | Examinar | Estructura | Buscar | Insertar | Vaciar | Eliminar |
| 5 tablas | Número de filas | | | | | |

Fig. 8. General database structure.

## 8. *Web Platform Development*

The website was developed using Visual Studio Code as the programming environment, with HTML for the structure and CSS for the design, aiming to enhance the user experience. The project was organized into folders, dividing the code into two main sections: *Web Apps*, which contains the user-facing components, and *Web Admin*, intended for site administration. The developed interfaces include the homepage, which introduces the company and its services; a user registration page, allowing users to sign up as clients or administrators; the antenna monitoring and management section, accessible based on user permissions; the footer, which provides contact information; and the login interface, where users can authenticate themselves. Fig. 9 shows the project structure created in Visual Studio Code and the software's homepage.

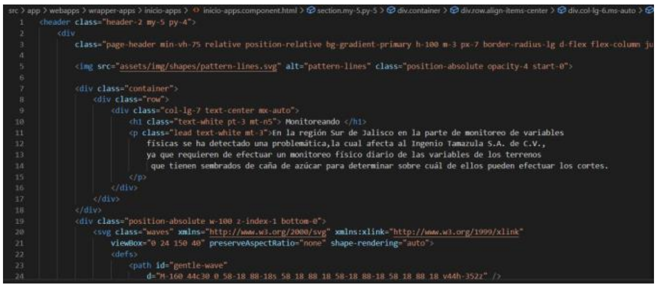


Fig. 9. Homepage of the monitoring software.

## 9 *Testing and Validation*

Tests were conducted to ensure the correct operation of the system, verifying both the programming logic and efficient communication with the database and the website. These tests included validation of the integration between different modules, system response under various conditions, and stability in data transmission. Fig. 10 shows the execution process of the code within the Arduino development environment, allowing the evaluation of system performance and the identification of potential errors before final implementation.

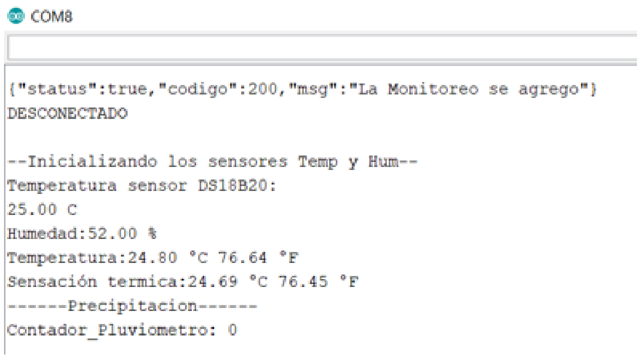
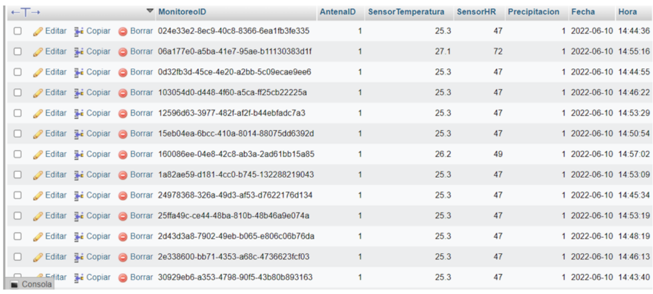


Fig. 10. Code execution in the Arduino IDE.

Functionality tests were also performed on the database to verify the correct capture and storage of information. Fig. 11 shows the process in which the system records the acquired data, including the exact date and time of each reading. The stored information contains details such as year, month, day, hour, minute, and second, enabling precise tracking of each record.



| | MonitoreoID | AntenaID | SensorTemperatura | SensorHR | Precipitacion | Fecha | Hora |
|---|---|---|---|---|---|---|---|
| Editar Copiar Borrar | 024e33e2-8ec9-40c8-8366-6ea1fb3fe335 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:44:36 |
| Editar Copiar Borrar | 06a177e0-a5ba-41e7-95ae-b11130383d1f | 1 | 27.1 | 72 | 1 | 2022-06-10 | 14:55:16 |
| Editar Copiar Borrar | 0d32fb3d-45ce-4e20-a2bb-5c09ecae9ee6 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:44:55 |
| Editar Copiar Borrar | 103054d0-d448-4f60-a5ca-ff25cb22225a | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:46:22 |
| Editar Copiar Borrar | 12596d63-3977-482f-af2f-b44ebfadc7a3 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:53:29 |
| Editar Copiar Borrar | 15eb04ea-6bcc-410a-8014-88075dd6392d | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:50:54 |
| Editar Copiar Borrar | 160086ee-04e8-42c8-ab3a-2ad61bb15a85 | 1 | 26.2 | 49 | 1 | 2022-06-10 | 14:57:02 |
| Editar Copiar Borrar | 1a82ae59-d181-4cc0-b745-132288219043 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:53:09 |
| Editar Copiar Borrar | 24978368-326a-49d3-af53-d7622176d134 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:45:34 |
| Editar Copiar Borrar | 25ffa49c-ce44-48ba-810b-48b46a9e074a | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:53:19 |
| Editar Copiar Borrar | 2d43d3a8-7902-49eb-b065-e806c06b76da | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:48:19 |
| Editar Copiar Borrar | 2e338600-bb71-4353-a68c-4736623fcf03 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:46:13 |
| Editar Copiar Borrar | 30929eb6-a353-4798-90f5-43b80b893163 | 1 | 25.3 | 47 | 1 | 2022-06-10 | 14:43:40 |

Fig. 11. Data logging process in the database.

**Experimental Validation.** The prototype was installed in a sugarcane field in Tamazula de Gordiano, Jalisco, where continuous tests were conducted for 30 days. The recorded variables were compared against reference instruments and validated with data from an official meteorological station operated by CONAGUA in Ciudad Guzmán, Jalisco. During this period, sensor accuracy, energy performance, and the structural integrity of the station were evaluated. The results confirmed the functionality of the system under real agricultural conditions, providing evidence of its applicability in field environments.

## III. RESULTS

The system proved to be functional and reliable. Figures 12 and 13 show the homepage of the monitoring system.


Fig. 12. Homepage of the monitoring platform (view 1).


Fig. 13. Homepage of the monitoring platform (view 2).

**System Access:** If the user already has a registered profile, they must enter their credentials on the login page. Otherwise, they should click on "Register." By selecting this option, the system redirects the user to a registration form, as shown in Fig. 14.

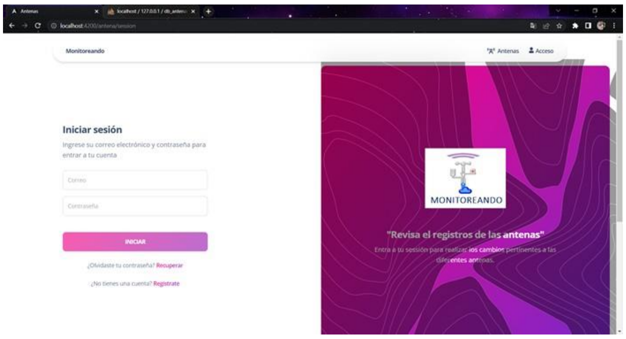



Fig. 14. User registration form.

**User Registration:** The registration form includes a field where the user must select their profile type: "Client" or "Administrator." Fig. 15 shows the screen displaying this instruction.

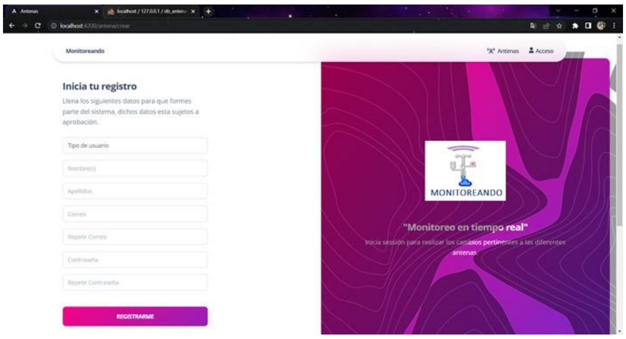



Fig. 15. User type selection screen in the registration form.

**Antenna Monitoring:** Once registered and logged in, the user can access the "Antennas" section. By clicking this option, a panel is displayed listing all registered antennas available for real-time monitoring. If detailed information about a specific antenna is required, the user must click on "More Information," which will display the data collected by the sensors. Fig. 16 illustrates this menu.

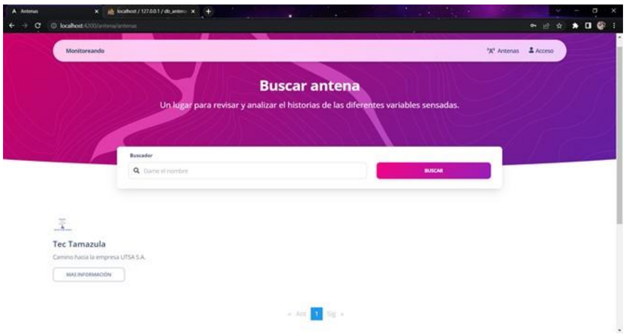



Fig. 16. Antenna monitoring menu. Source: Author.

**Administrator Functionalities:** If the user logs in as an administrator, the page will display a list of registered antennas. Additionally, the administrator has options to add, edit, or delete antennas, as well as to view real-time data. Fig. 17 shows this interface.

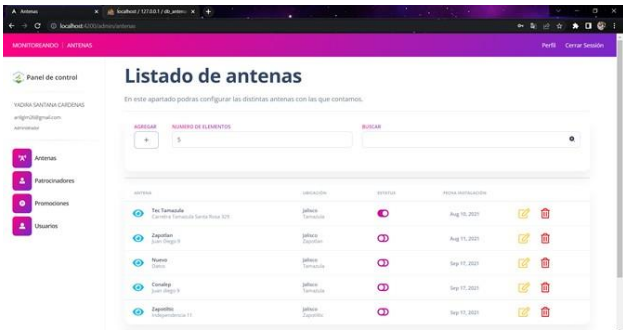



Fig. 17. Administrator view of the web page.

The control panel also includes a user list with the statuses "Logged In," "Allowed," and "Created," which can be edited or deleted, as illustrated in Fig. 18.

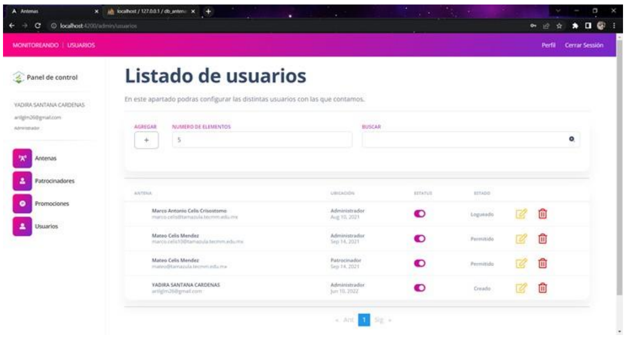



Fig. 18. User list on the web page.

**Data Transmission and Reception:** Data transmission from Arduino to the database is carried out through an internet protocol. The data is sent via the IP address assigned to the board during programming. When the transmission is successful, the system displays a confirmation message:

{"status": true, "codigo": 200, "msg": "La Monitoreo se agregó"}

This message indicates that the information was successfully stored in the database. Fig. 19 illustrates this process.

```
CONECTADO AL SERVIDOR
HTTP/1.1 200 OK
Date: Sat, 11 Jun 2022 16:59:25 GMT
Server: Apache/2.4.53 (Win64) OpenSSL/1.1.1n PHP/8.1.6
X-Powered-By: PHP/8.1.6
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, PATCH, OPTIONS
Access-Control-Allow-Headers: X-Requested-With, Content-Type, Accept,
P3P: CP="IDC DSP COR CURa ADMa OUR IND PHY ONL COM STA"
Content-Length: 59
Connection: close
Content-Type: text/html; charset=UTF-8

{"status":true,"codigo":200,"msg":"La Monitoreo se agrego"}
DESCONECTADO

--Inicializando los sensores Temp y Hum--
Temperatura sensor DS18B20:
25.00 C
Humedad:52.00 %
Temperatura:24.80 °C 76.64 °F
Sensación termica:24.69 °C 76.45 °F
------Precipitacion------
Contador_Pluviometro: 0
```

Fig. 19. Data transmission from Arduino to the database.

On the web page, the information is captured and stored in the database in real time, recording the exact moment it is received. Fig. 20 shows the data storage process.

| AntenaID | SensorTemperatura | SensorHR | SensorUV | Tempmax | Tempmin | Fecha | Hora |
|---|---|---|---|---|---|---|---|
| 1 | 24.44 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13 56 5S |
| 1 | 24 38 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13 56 37 |
| 1 | 24 38 | 52 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 55.10 |
| 1 | 24.44 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13 56 26 |
| 1 | 24.38 | 53 | 0 | 27.31 | 23.19 | 2022-06-16 | 13.53 24 |
| 1 | 24 38 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 53 03 |
| 1 | 24 38 | 53 | 0 | 27.31 | 23.19 | 2022-06-16 | 13.53 34 |
| 1 | 24 38 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 53 02 |
| 1 | 24 56 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 51.15 |
| 1 | 23 25 | 53 | 0 | 27 | 23.19 | 2022-06-16 | 13.49 0S |
| 1 | 24.44 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13:57:10 |
| 1 | 24 31 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 55 00 |
| 1 | 24 56 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 51.05 |
| 1 | 24 38 | 53 | 0 | 27.31 | 23.19 | 2022-06-16 | 13.54 0S |
| 1 | 24.44 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13.57.30 |
| 1 | 24 56 | 53 | 0 | 27 31 | 23.19 | 2022-06-16 | 13 51 25 |
| 1 | 24.38 | 52 | 0 | 27.31 | 23.19 | 2022-06-16 | 13 55 54 |

Fig. 20. Data storage in the database.

**Data Visualization.** The information stored in the database is displayed on the web page in real time. The client can configure the time intervals for data visualization. The information is presented in two formats: graphical, located on the left side of the screen, and numerical, on the right side. Fig. 21 illustrates this visualization.

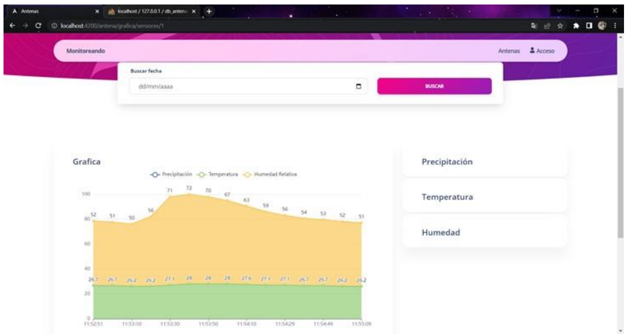



Fig. 21. Data visualization on the web page.

Before the implementation of the platform, querying environmental data required physical visits to the monitoring points, with an average review time of 30 minutes per visit. With the web platform, data is now available in real time with an update latency of less than 3 seconds, and the information frequency is approximately every minute, representing a 95% reduction in query time. Additionally, the automation of data recording has helped decrease errors associated with manual data capture. Thanks to the real-time visualization of humidity levels, farmers were able to optimize irrigation frequency, estimating a reduction in water consumption in the test area. Table 2 presents a comparison between the conventional method of querying environmental data and the proposed platform, highlighting the operational improvements achieved.

TABLE II. COMPARISON BETWEEN TRADITIONAL SYSTEM AND PROPOSED PLATFORM

| Parameter | Traditional method | Proposed platform | Estimated improvement |
|---|---|---|---|
| Data Query Time | ~ 30 minutes (physical visits) | <3 seconds (remote real-time access) | 95% reduction |
| Update Frequency | Manual, sporadic | Automatic, every minute | Continuous improvement |
| Data Accuracy | Prone to human errors | Calibrated digital sensors | Higher reliability |
| Accessibility | On-site only | From any device with internet | High accessibility |
| Estimated Cost | Variable (labor, logistics) | ~$500 USD (one-time technological investment) | Significant reduction |
| Data Logging Automation | No | Yes | Elimination of manual errors |
| Scalability Capacity | Limited | High (AI, more sensors, other zones) | Future scalability |

When comparing the developed prototype with commercial meteorological stations such as the MK-III RTN-LR, AgroMet, and MSO ICT International, significant differences were identified in terms of cost, scalability, and flexibility. Commercial equipment generally exceeds $20,000 MXN and relies on proprietary platforms that restrict hardware or software modifications. In contrast, the system developed in this study has an approximate cost of $4,500 MXN and allows modifications at both the hardware and software levels, making it adaptable to the specific needs of small and medium sugarcane producers.

Another key advantage is the independence of the web visualization platform, which does not require external subscriptions or specialized technical support. While commercial solutions typically include additional variables such as atmospheric pressure, solar radiation, or wind speed with higher sensor precision, the proposed prototype focuses on the most relevant parameters for the sugarcane sector—temperature, humidity, and precipitation—prioritizing low cost and accessibility over high complexity.

This trade-off highlights the system's suitability for small-scale rural contexts, where affordability and adaptability are often more critical than exhaustive precision. In addition, the modular design allows the future integration of higher-grade sensors or additional variables, providing a pathway to scale the platform as user requirements evolve.

## IV. DISCUSSION

The proposed platform demonstrates significant advantages over commercial solutions, with a 75% cost reduction (from >$1000 to $27 USD) and improved accessibility for small-scale producers [6]. Implementation requires minimal technical expertise while achieving substantial performance gains, including a 99.8% reduction in data query latency (from 30 minutes to <3 seconds). These efficiency improvements align with established precision agriculture benchmarks [6], who also emphasize the importance of low latency in data updates for agronomic decision-making. Furthermore, the possibility of remote access through a web interface, as demonstrated in similar studies [5, 7], broadens the system's utility in rural contexts where infrastructure access is limited.

[11] validates the technical approach, showing open-source platforms and low-cost sensors (unit cost <$50) can achieve comparable accuracy (±2%) to commercial agricultural monitoring systems, who highlight the feasibility of economical IoT systems for precision agriculture contexts. However, unlike proposals based on LoRa or Sigfox, this development uses Wi-Fi as the communication medium, which represents a limitation in terms of range in remote areas but an advantage in terms of rapid implementation in areas with basic connectivity. Unlike other studies primarily focused on data acquisition and transmission [4, 14], this proposal also addresses accessible visualization and management of information through a functional and user-friendly web platform. This aspect is key for real adoption by end users without specialized technical training. These findings demonstrate the relevance of the developed solution and open the door to future improvements to expand its functionality and coverage.

The technical scope of the developed system demonstrates its replicability in diverse agricultural contexts due to the use of open-source microcontrollers and accessible sensors. This characteristic not only facilitates adoption by small-scale producers but also enables future integration with advanced analysis technologies such as artificial intelligence or big data, particularly for predicting climatic variables and performing multivariate analyses. Nevertheless, some limitations must be acknowledged. The accuracy of the selected sensors, while acceptable for decision-making at the farm level, does not reach industrial-grade standards, which could affect long-term stability under harsh field conditions. Similarly, the current reliance on Ethernet/Wi-Fi connectivity restricts deployment in rural areas lacking LAN or stable internet infrastructure; however, the modular design allows substitution with long-range communication modules such as LoRa or ESP32 in future iterations. Additionally, although the structure of the monitoring station proved resistant during testing, it may require reinforcement to withstand wind speeds exceeding 25 km/h in exposed environments. These considerations highlight both the practicality and adaptability of the system while underscoring the areas where improvements are required to ensure robustness and scalability in large-scale agricultural deployments.

Additionally, in a parallel study published by Cárdenas Magaña [25], the prototype was validated in a sugarcane field in Tamazula de Gordiano, Jalisco, during 30 consecutive days, with data acquisition every 5 minutes. The measured variables were compared with those from a reference meteorological station operated by CONAGUA in Ciudad Guzmán, showing acceptable agreement in temperature, humidity, and precipitation parameters, as well as in the system's energy consumption. These results confirm the technical reliability and viability of the prototype as a low-cost alternative to higher-priced commercial stations.

## V. CONCLUSIONS

The implementation of the proposed web platform proved to be an efficient, accessible, and replicable solution for real-time monitoring of environmental variables in the agricultural sector, particularly for sugarcane crops in rural areas. The system successfully integrated low-cost digital sensors (DHT11, DS18B20, and KY-035), an Arduino microcontroller, and a MySQL database, all interconnected via Wi-Fi and visualized through a web interface developed using HTML, CSS, and PHP.

Among the most relevant technical achievements were data transmission with latency below 3 seconds, an update frequency of approximately one minute, and real-time remote visualization. This represented a 95% reduction in query time compared to the traditional method and minimized human errors by automating data recording.

From an economic perspective, the system had an approximate cost of $500 USD, representing a viable alternative to commercial solutions costing over $2,000 USD, thereby facilitating adoption by small and medium producers. However, areas for improvement were identified, including the development of a mobile-optimized interface, integration of artificial intelligence algorithms for predictive analysis, and the use of long-range communication technologies such as LoRa or NB-IoT.

In the future, the system is planned to be scaled into a distributed network of monitoring stations, incorporating new variables such as solar radiation, wind speed, and soil nutrient content, with the aim of enriching the characterization of the agricultural environment and strengthening data-driven decision-making.

## ACKNOWLEDGMENT

We sincerely thank the Tecnológico Superior de Jalisco for the support provided in carrying out this research. We extend our appreciation to Mtra. Brenda Yerania Ortega Flores, Director of our institution, for her constant support. We also express our gratitude to the students who actively participated in the development of the project, as well as to our fellow researchers for their valuable collaboration and academic commitment.

## REFERENCES

[1] M. C. Abad Alay, M. A. Méndez García, and O. R. Erazo Moreta, "Internet of Things technology in crop monitoring," *Revista Observatorio de I+D+i*, vol. 4, no. 3, pp. 12-23, Dec. 2023, DOI: 10.35290/ro.v4n3.2023.939.

[2] J. P. Tovar Soto, J. D. S. Solórzano Suárez, A. B. Rodríguez, and G. O. Rodríguez Cainaba, "Internet of Things applied to agriculture: current status," *Lámpsakos*, no. 22, pp. 86-105, Jul.–Dec. 2019, DOI: 10.21501/21454086.3253.

[3] Campus Agrotecnológico. *Data acquisition with sensors in agriculture*. (Mar. 10, 2025). [Online]. Available: https://agrotechcampus.com/blog/captacion-de-datos-con-sensores-en-agricultura/

[4] L. A. Ariza, J. D. Morales, and J. A. Ibáñez, *Design and implementation of an IoT system for crop monitoring*. Repositorio Universidad Jorge Tadeo Lozano, 2022. [Online]. Available: http://hdl.handle.net/10554/63698

[5] F. Karim, K. Fathalah, and A. Frihida, "Web-of-things monitoring system for precision agriculture," *Procedia Computer Science*, vol. 110, pp. 402-409, 2017, DOI: 10.1016/j.procs.2017.06.083.

[6] Cadena SER, *Observatorio Citrícola equips the sector with smart digital tools*, 2024. [Online]. Available: https://cadenaser.com/andalucia/2024/11/20/el-observatorio-citricola-dota-de-herramientas-digitales-inteligentes-al-sector-para-tomar-decisiones-radio-palma/

[7] S. Sohail, K. Shaik, K. Kalaiselvi, and K. Shashank, "Intelligent IoT-based system for precision agriculture monitoring," *Grenze International Journal of Engineering and Technology*, vol. 11, no. 1, pp. 4339-4345, 2025.

[8] P. Rajak, A. Ganguly, S. Adhikary, and S. Bhattacharya, "Internet of Things and smart sensors in agriculture: scope and challenges," *Revista de Investigación Agrícola y Alimentaria*, vol. 14, p. 100776, 2023, DOI: 10.1016/j.jafr.2023.100776.

[9] J. Nuvvula, S. Adiraju, S. Mubin, S. Bano, and V. S. R. Valisetty, "Environmentally smart agriculture monitoring system using IoT," *International Journal of Pure and Applied Mathematics*, vol. 115, no. 6, pp. 313-320, 2017.

[10] D. T. Huong, N. T. H. Duy, P. V. M. Tu, H. H. Hanh, and K. Yamada, "Enhancing precision agriculture: An IoT-based smart monitoring system integrating LoRaWAN, ML, and AR," *EAI Endorsed Trans. Smart Cities*, vol. 7, no. 4, pp. 1-11, 2024. DOI: 10.4108/eetsc.7286.

[11] A. I. Dahou Idrissi, A. Abuabdillah, M. Chikhaoui, and R. Bouabid, "Low-cost IoT-based monitoring system for precision agriculture," *E3S Web of Conferences*, vol. 492, pp. 1-9, 2024, DOI: 10.1051/e3sconf/202449201003.

[12] E. E. K. Senoo, L. Anggraini, J. A. Kumi, L. B. Karolina, E. Akansah, H. A. Sulyman, I. Mendonça, and M. Aritsugi, "IoT solutions with artificial intelligence technologies for precision agriculture: definitions, applications, challenges, and opportunities," *Electronics*, vol. 13, no. 10, p. 1894, 2024. DOI: 10.3390/electronics13101894.

[13] A. Z. Babar and Ö. B. Akan, "Sustainable and precision agriculture with Internet of Things (IoT)," *arXiv:2404.06341v2*, 2024. Available: https://arxiv.org/abs/2404.06341.

[14] K. J. Naidu, K. V. Babu, C. R. C. Sai, P. Ganesh, T. G. Sai, N. S. Naidu, C. Praneeth, and G. Sumanth, "Precision Agriculture Monitoring System," *Biosciences Biotechnology Research Asia*, vol. 21, no. 4, 2024. DOI: 10.13005/bbra/3324.

[15] Y. Zhao, Q. Li, W. Yi, and H. Xiong, "Optimization of agricultural IoT data storage and blockchain-based information security method," *Agriculture*, vol. 13, no.2 p. 274, 2023. DOI: 10.3390/agriculture13020274.

[16] N. Silva, J. Mendes, R. Silva, F. N. Santos, P. Mestre, C. Serôdio, and R. Morais, "Low-cost IoT LoRa solutions for precision agriculture monitoring practices," *Proc. 19th EPIA Conference on Artificial Intelligence*, Vila Real, Portugal, Sep. 2019, pp. 224-235. DOI: 10.1007/978-3-030-30241-2_20.

[17] R. M. Khaleefah, N. A. M. Al-Isawi, M. K. Hussein, and N. A. M. Alduais, "Optimization of IoT data transmission in smart agriculture: a comparative study of reduction techniques," *Proc. Int. Conf. Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Istanbul, Turkey, 2023, DOI: 10.1109/HORA58378.2023.10156757.

[18] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed., Sebastopol, CA, USA: O'Reilly Media, 2015.

[19] Oracle, *What is JSON?* (Mar. 22, 2025). [Online]. Available: https://www.oracle.com/mx/database/what-is-json/

[20] Microsoft, "Visual Studio Code," [Online]. Available: https://code.visualstudio.com/. [Accessed: Jun. 2, 2025].

[21] S. Monk, *Programming Arduino™: Getting Started with Sketches*, 2nd ed., New York, NY, USA: McGraw-Hill, 2016, p. 178.

[22] Jada Support, "Connecting Arduino to MySQL database," Jada Support Projects, Aug. 28, 2015. [Online]. Available: https://jadasupport.wordpress.com/2015/08/28/conectar-arduino-a-base-de-datos-mysql/. [Accessed: Mar. 22, 2025].

[23] I. de Souza, "How to create a website: step-by-step to make yours from scratch," Rock Content, Jan. 15, 2021. [Online]. Available: https://rockcontent.com/es/blog/como-crear-un-sitio-web/. [Accessed: Mar. 22, 2025].

[24] De Ingenierías, "EasyEDA: what it is, what it is for, and how it works," [Online]. Available: https://deingenierias.com/software/easyeda/. [Accessed: Mar. 22, 2025].

[25] J. A. Cárdenas Magaña, "Diseño y Validación de una Estación Meteorológica Autónoma de Bajo Costo para el Sector Cañero," *Ideas en Ciencias de la Ingeniería*, vol. 3, no. 2, pp. 78–94, 2025. DOI: 10.36677/rici.v3i2.26210.