

---

# PROTOTIPO DE UN SISTEMA DE MONITOREO AGRARIO CON VISUALIZACIÓN DE DATOS EN SERVIDOR WEB

---

**Flores Quispe, Carlos**  
Estudiante de Ingeniería Electrónica  
Universidad Privada Antenor Orrego

**Huamán Kemper, Ricardo**  
Estudiante de Ingeniería Electrónica  
Universidad Privada Antenor Orrego

**Alva Alarcón, Jorge**  
Ingeniero Electrónico  
Universidad Privada Antenor Orrego  
Asesor de Proyecto

## RESUMEN

En el presente proyecto se propone la elaboración de un sistema de monitoreo agrario a escala basado en el Socket on Chip (SoC), NodeMCU ESP8266; este sistema tendrá la capacidad de captar valores de temperatura y humedad de un plantín, para luego enviarlos a través de internet a una laptop remota. La información recibida se guardará en una base de datos para luego ser graficada en tiempo real en una página web que será accesible desde cualquier dispositivo con internet. En colaboración con estudiantes de Ingeniería Agrónoma de la Universidad Privada Antenor Orrego se espera que este trabajo contribuya con un mejor cuidado de los cultivos.

**Palabras Clave:** NodeMCU ESP8266, Sistema de Monitoreo Agrario, Servidor Web.

## ABSTRACT

The present project proposes the elaboration of an agricultural monitoring system based on the Socket on Chip (SoC), NodeMCU ESP8266; this system will have the ability to capture temperature and humidity values from a seedling, and then send them over the internet to a remote laptop. The information received will be saved in a database and then plotted in real time on a web page that will be accessible from any device with internet. In collaboration with students of Agronomic Engineering from the Antenor Orrego Private University, it is expected that this work will contribute to better crop care.

**Keywords:** NodeMCU ESP8266, Agriculture Monitoring System, Web Server.

## 1. Introducción

Con la creciente tasa de automatización en las industrias, distintos procesos se han ido confiando a robots concomitante-mente al advenimiento de nuevas tecnologías; la industria agraria es un sector visto de forma principalmente tradicional, pero que también está descubriendo que gracias a las mejoras tecnológicas es capaz de avanzar de una manera notable [1]. De los diferentes proyectos enfocados en el sector, el de este trabajo es uno dirigido al monitoreo titulado "Prototipo de un Sistema de Monitoreo Agrario con Visualización de Datos en Servidor Web".

Monitoreo es el proceso sistemático de recolectar, analizar y utilizar información para hacer seguimiento al progreso de un programa en pos de la consecución de sus objetivos, y para guiar las decisiones de gestión. El monitoreo generalmente se dirige a los procesos en lo que respecta a cómo, cuándo y dónde tienen lugar las actividades, quién las ejecuta y a cuántas personas o entidades beneficia. El sistema de monitoreo para cultivos es un sistema que permite al agricultor realizar un seguimiento en tiempo real de plantines en los diferentes cultivos y campos, con el fin de determinar su evolución negativa y positiva [2].

Usando como punto de inicio los pasos y estrategias utilizadas para la construcción de este proyecto se espera que otros puedan imitar e incluso mejorar esta versión, convirtiéndose en una gran alternativa para el cuidado de los cultivos basado en el monitoreo.

## 2. Contexto del Proyecto

Siendo este trabajo dirigido a estudiantes de ingeniería electrónica y/o adeptos al tema, en el presente informe se desarrollará el "Prototipo de un Sistema de Monitoreo Agrario con Visualización de Datos en Servidor Web" para el curso de Sistemas Embebidos.

Para la elaboración de este proyecto se tendrán en cuenta dos fases; en la primera se desarrollará el diseño de un circuito encargado de capturar, controlar y enviar la información de un plantín. Para el mencionado circuito se indicará cada componente usado y sus características para así saber su rol en el circuito general, se mostrarán además los diagramas circuitales como guía para el cableado. Este será un paso importante para que la información sea la correcta y evitar datos erróneos que podrían resultar irreversibles para el cuidado del plantín.

La segunda fase tratará todo lo relacionado a la codificación, aquí se mencionarán los lenguajes de programación empleados y cuál fue su función. Estos códigos en su mayoría se encargarán de todo el proceso desde que se tienen los datos hasta la gráfica la cual posteriormente será analizada, dichas gráficas y datos obtenidos se mostrarán en tiempo real en una página web [3].

Las herramientas clave para lograr el proyecto planteado son en el caso del circuito el microcontrolador NodeMCU ESP8266 y en la codificación la herramienta imprescindible es el lenguaje de programación Python.

## 3. Objetivos

- Diseñar un modelo circuital usando el NodeMCU ESP8266 y sensores para captar información del plantín.
- Emplear el lenguaje de programación Arduino para el NodeMCU ESP8266 y digitalizar datos correspondientes a los sensores.
- Emplear el lenguaje de programación Python para analizar y organizar datos.
- Aplicando IoT lograr mostrar la información en tiempo real en una página web.
- Dar a conocer este sistema de monitoreo agrario como una alternativa de cuidado más riguroso para plantines.
- Investigar y conocer más acerca de sistemas embebidos e IoT.

## 4. Desarrollo del Proyecto

### 4.1. Prototipo de un Sistema de Monitoreo Agrario con Visualización de Datos en Servidor Web

El sistema de monitoreo se implementará a escala aplicado a uno o varios plantines, teniendo como proyección el monitoreo dirigido a campos enteros de cultivos.

## 4.2. Componentes

Se explicará qué función tiene cada componente utilizado en este proyecto, de esta manera su disponibilidad no sería problema y podría sustituirse teniendo en cuenta su aplicación en el proyecto.

### 4.2.1. NodeMCU ESP8266

NodeMCU ESP8266 es una plataforma de desarrollo similar a Arduino especialmente orientada al Internet de las cosas (IoT). La placa NodeMCU v2 ESP8266 tiene como núcleo al SoM ESP-12E que a su vez está basado en el SoC Wi-Fi ESP8266, integra además el conversor USB-Serial TTL CP2102 y conector micro-USB necesario para la programación y comunicación a PC. NodeMCU v2 ESP8266 está diseñado especialmente para trabajar montado en protoboard o soldado sobre una placa. Posee un regulador de voltaje de 3.3V en placa, esto permite alimentar la placa directamente del puerto micro-USB o por los pines 5V y GND. Los pines de entradas/salidas (GPIO) trabajan a 3.3V por lo que para conexión a sistemas de 5V es necesario utilizar convertidores de nivel.

El SoC (System On a Chip) ESP8266 de Espressif Systems es un chip especialmente diseñado para las necesidades de un mundo conectado, integra un potente microcontrolador con arquitectura de 32 bits (más potente que el Arduino Due) y conectividad Wi-Fi. El SoM (System on Module) ESP-12E fabricado por Ai-Thinker integra en un módulo el SoC ESP8266, memoria FLASH, cristal oscilador y antena WiFi en PCB [4].

La plataforma ESP8266 permite el desarrollo de aplicaciones en diferentes lenguajes como: Arduino, Lua, MicroPython, C/C++, Scratch.

### Conectividad

- SDIO 2.0, SPI, UART.
- Integra RF switch, balun, 24dBm PA, DCXO y PMU.
- Posee un procesador RISC, memoria en chip e interface para memoria externa.
- Procesador MAC/Baseband integrado.
- Interface I2S para aplicaciones de audio de alta calidad.
- Reguladores de voltaje lineales "low-dropout" en chip.
- Arquitectura propietaria de generación de clock "spurious free".
- Módulos WEP, TKIP, AES y WAPI integrados.

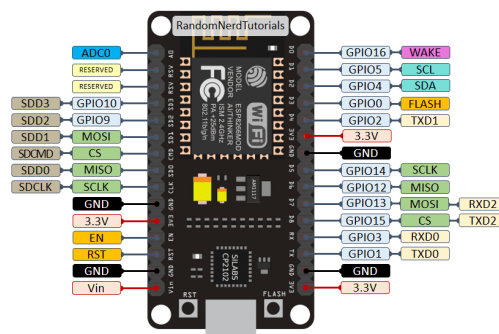


Figura 1. NodeMCU ESP8266.

### 4.2.2. Sensor de temperatura y humedad relativa DHT22 (AM2302)

Es un sensor digital de temperatura y humedad relativa de buen rendimiento y bajo costo. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Utilizado en aplicaciones de control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.

El protocolo de comunicación entre el sensor y el microcontrolador emplea un único hilo o cable, la distancia máxima

recomendable de longitud de cable es de 20m., de preferencia utilizar cable apantallado. Proteger el sensor de la luz directa del sol (radiación UV). El DHT22 presenta mejores prestaciones respecto al sensor DHT11, como mejor resolución, mayor precisión y un empaque más robusto [5].

### Pines

- 1- Alimentación: +5V (VCC).
- 2- Datos (DATA).
- 3- No Usado (NC).
- 4- Tierra (GND).

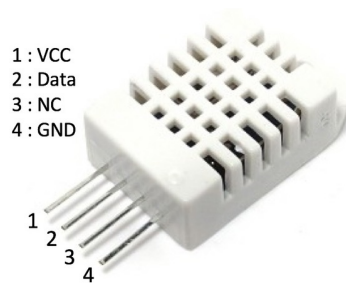


Figura 2. Sensor DHT22.

### 4.2.3. Módulo Buzzer Pasivo

Un buzzer es el elemento más usado como indicador de sonido, puede usarse al presionar una tecla, reproducir melodías, alarmas. El módulo incluye un transistor que cumple la función de amplificador de señal, de esta forma podemos conectar el módulo directamente a nuestro Arduino y no preocuparnos por dañar algún componente. El buzzer piezoeléctrico es de tipo pasivo por lo que puede reproducir melodías utilizando la función TONE de Arduino [6].

### Especificaciones

- Voltaje de Operación: 3.3V - 5V DC.
- Tipo: Piezo eléctrico pasivo.
- Incluye el transistor S8550.
- Pines: VCC, GND y Señal.

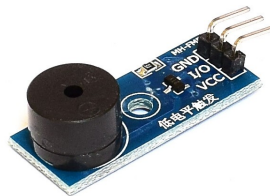


Figura 3. Módulo Buzzer. Pasivo.

#### 4.2.4. Módulo PIR HC-SR501

Los sensores PIR tienen como función detector movimiento (de personas), normalmente se busca detectar el movimiento de una persona dentro del rango del sensor. Son baratos, pequeños, de bajo consumo y fáciles de utilizar, además no se desgastan. Normalmente los podemos encontrar en electrodomésticos y gadgets para la oficina o el hogar. Son conocidos como PIR, “Sensores Infrarrojos” o “Sensores de movimiento”.

Este módulo contiene un sensor piroeléctrico, el cual puede detectar cambios de radiación infrarroja. Todo objeto (o cuerpo humano) emite cierto nivel de radiación, y entre mayor temperatura tenga, mayor radiación emitirá. El sensor dentro del detector de movimientos está dividido en 2 mitades o 2 lados. La razón para esto es que estamos buscando la diferencia en el movimiento, no el promedio. Las dos mitades están unidas por cables de modo que se cancelan una a otra. Si una mitad recibe más o menos radiación IR, la salida cambiará a alto o bajo. El módulo PIR HC-SR501 cuenta con 2 potenciómetros que permiten regular la sensibilidad y el tiempo de duración del pulso [7].

#### Especificaciones

- Voltaje de operación: 4.5VDC - 20VDC.
- Rango de detección: 3 a 7 metros, ajustable mediante Trimmer (Sx).
- Ángulo de detección: <100° (cono).
- Tiempo de retardo: 5-200 S (puede ser ajustado (Tx), por defecto 5S  $\pm$  3 %).
- Tiempo de bloqueo: 2.5 S (por defecto).
- Temperatura de trabajo: -20°C hasta 80°C.
- Dimensión: 3.2 cm x 2.4 cm x 1.8 cm (aprox.).
- Redisparo configurable mediante jumper de soldadura.



Figura 4. Módulo PIR HC-SR501.

#### 4.2.5. Cable Jumper

Es un tipo de socket rectangular de plástico que a su vez tiene en su interior dos o más sockets metálicos con un espacio entre ellos de 0.2mm hechos de fósforo-bronce, de una aleación de cobre-níquel, de estaño o de latón y con un color dorado o cromado, de tal manera que cuando se introducen y se empujan hacia los pines de un circuito, éstos cierran el circuito cubriendo completamente los pines, resultando en una conexión temporal [8].



Figura 5. Cable Jumper.

#### 4.2.6. Protoboard

El Protoboard sirve para probar circuitos sin tener que fabricar un PCB. Todos los pines están espaciados por un estándar de 0.1". Los dos conjuntos de cinco filas están separadas por aproximadamente 0.3", perfecto para integrados tipo DIP [9].

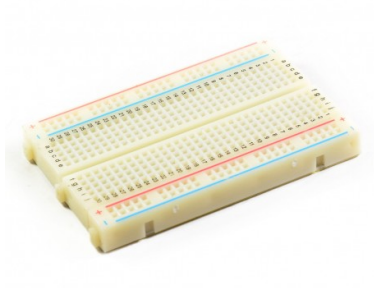


Figura 6. Protoboard.

#### 4.2.7. Diodos LED

Un diodo LED es un diodo que además de permitir el paso de la corriente solo en un sentido (en el sentido en el que la corriente pasa por el diodo), emite luz [10].

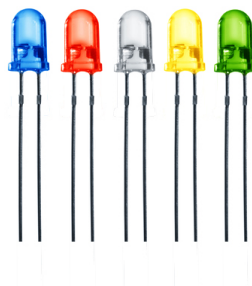


Figura 7. Diodo LED.

#### 4.2.8. Maqueta

Esto puede variar ya que no implica ni afecta de manera importante al proyecto; la forma, material y diseño dependerá del gusto del investigador.



Figura 8. Maqueta.



Figura 9. Maqueta.

### **4.3. Herramientas de Desarrollo**

En los últimos años, se han desarrollado diferentes tipos de plataformas de programación y plataformas de simulación, estas son muy útiles a la hora de desarrollar nuestros proyectos, ya que permite revisar su funcionamiento antes de implementarlo. En este proyecto se utilizaron las siguientes herramientas para su desarrollo.

#### **4.3.1. Proteus Design Lab**

Proteus es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción [11].

#### **4.3.2. Arduino (Software)**

La plataforma Arduino se programa con un lenguaje propio basado en el lenguaje de programación de alto nivel Processing, lo que significa que es similar a C++. Se trata de un lenguaje de programación de propósito general asociado a un sistema operativo llamado UNIX.

Este lenguaje de medio nivel, trata con objetos básicos como caracteres, números, bits y direcciones de memoria, entre otros. Este tipo de lenguaje posee una gran portabilidad. Gracias a ello se suele utilizar para la programación de sistemas como la construcción de intérpretes, compiladores, y editores de texto [12].

#### **4.3.3. Visual Studio Code**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial esta bajo software privativo desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online) [13].

#### **4.3.4. Python**

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License [14].

### **Herramientas Para el Desarrollo de la Página Web**

#### **4.3.5. Javascript**

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional) [15].



#### 4.3.6. HTML

Siglas en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros [16].

#### 4.3.7. CSS

CSS (siglas en inglés de Cascading Style Sheets), en español "Hojas de estilo en cascada", es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Te puede ayudar a crear tu propio sitio web. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS) [17].

#### 4.3.8. SQLAlchemy

La filosofía de SQLAlchemy es que las bases de datos relacionales se comportan menos como colecciones de objetos a medida que la escala aumenta y el rendimiento comienza a ser una preocupación, mientras que las colecciones de objetos se comportan menos como tablas y filas a medida que se diseña más abstracción en ellas. Por esta razón, ha adoptado el patrón de mapeador de datos (similar a Hibernate para Java) en lugar del patrón de registro activo utilizado por varios otros mapeadores relacionales de objetos. Sin embargo, los complementos opcionales permiten a los usuarios desarrollar usando sintaxis declarativa.

SQLAlchemy se lanzó por primera vez en febrero de 2006 y rápidamente se ha convertido en una de las herramientas de mapeo relacional de objetos más utilizadas en la comunidad de Python, junto con el ORM de Django [18].

### 4.4. Esquemas

#### 4.4.1. Diagramas de Bloques

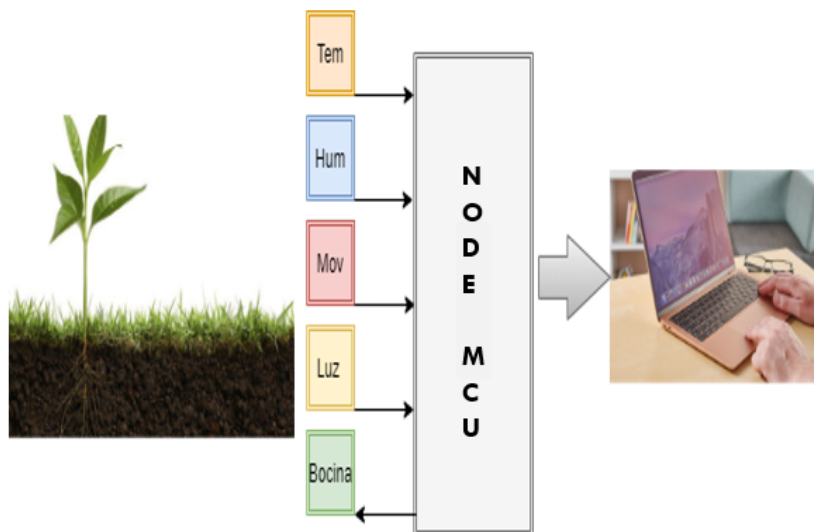


Figura 10. Diagrama de Bloques del Circuito.



Figura 11. Diagrama de Bloques del Flujo de Datos.

#### 4.4.2. Proteus Design Lab

Este circuito en Proteus es como se diseñó y se cableó el circuito físico, en este esquema no incluye el módulo PIR HC-SR501 también usado en el proyecto (El pin de datos de este módulo va conectado al pin D2 del circuito mostrado en la Figura 12).

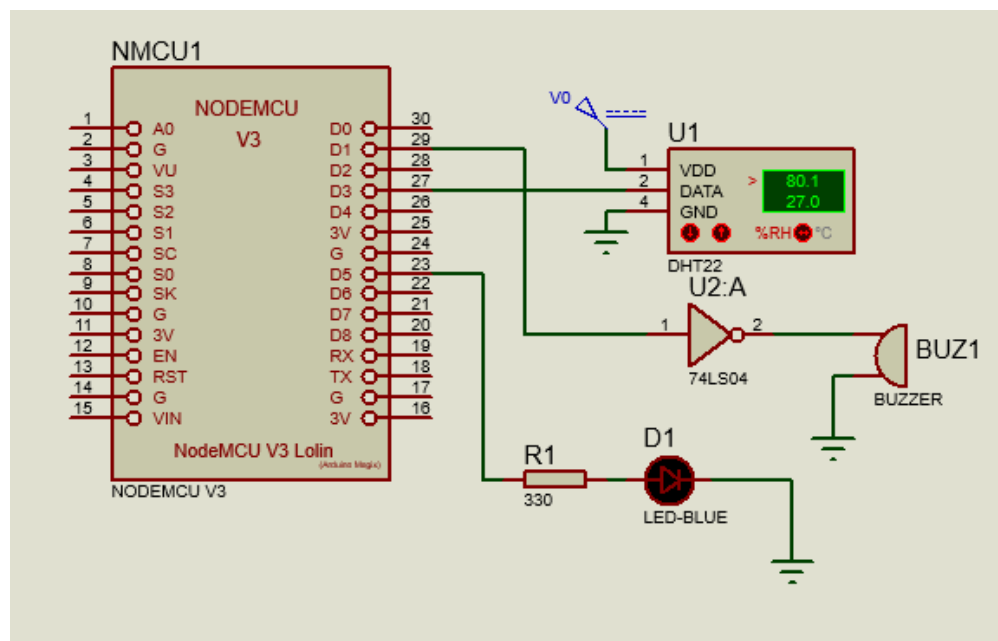


Figura 12. Circuito en Proteus.

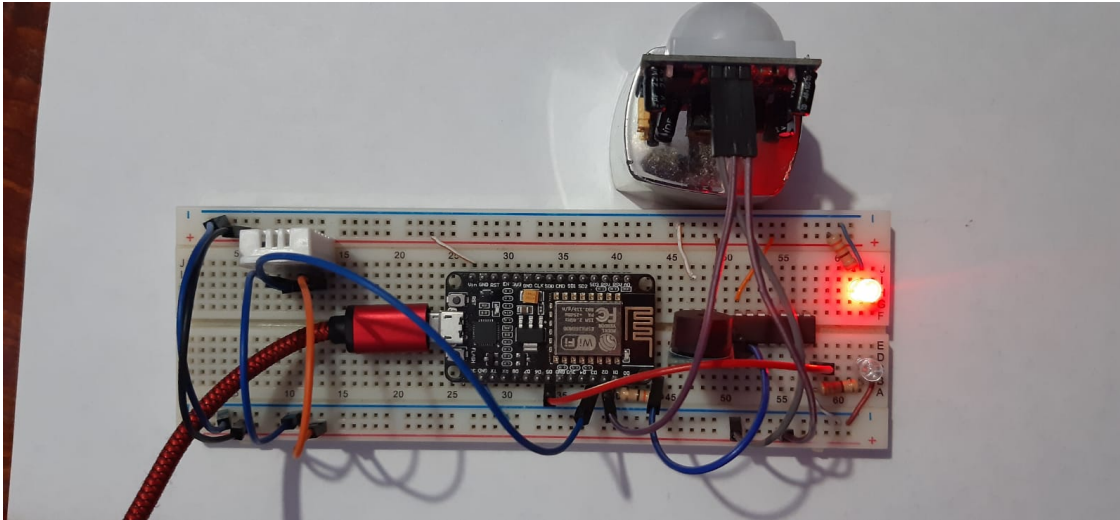


Figura 13. Circuito implementado.

#### 4.5. Código en Arduino

Este código va dirigido para el microcontrolador NodeMCU ESP8266, con el fin de recibir datos de los sensores en el puerto COM4 de Arduino. También aquí se codifica el funcionamiento del módulo PIR y el buzzer.

```
prueba01 $  
1 #include <DHT.h>  
2 #include <ESP8266WiFi.h>  
3 #include <WiFiClient.h>  
4  
5 #define DHTPIN 0  
6 #define DHTTYPE DHT22  
7 #define pirPin 4  
8 #define ledPin 14  
9 #define BUZZER 5  
10  
11 int val = 0;  
12 bool motionState = false;  
13  
14 DHT dht(DHTPIN, DHTTYPE);  
15  
16 void setup() {  
17   Serial.begin(9600);  
18   pinMode(ledPin, OUTPUT);  
19   pinMode(pirPin, INPUT);  
20   pinMode(BUZZER, OUTPUT);  
21   dht.begin();  
22
```

```

23 }
24
25 void loop() {
26   float h = dht.readHumidity(); //Se lee la humedad
27   float t = dht.readTemperature(); //Se lee la temperatura
28   //Se imprimen las variables
29   Serial.print(h);
30   Serial.print(" ");
31   Serial.println(t);
32   delay(3000); //Se espera 3 segundos para seguir leyendo datos
33
34   // Lee el valor de pirpin y lo guarda en val:
35   val = digitalRead(pirPin);
36   // Si se detecta movimiento (pirPin = HIGH), se ejecuta lo siguiente:
37   if (val == HIGH) {
38     digitalWrite(ledPin, HIGH); // enciende el LED.
39     // Cambia el estado de motion a true (movimiento detectado):
40     if (motionState == false) {
41       digitalWrite(BUZZER, HIGH);
42       motionState = true;
43     }
44   }
45
46   // Si ningun movimiento es detectado (pirPin = LOW), se ejecuta lo siguiente:
47   else {
48     digitalWrite(ledPin, LOW); // Apaga el LED.
49     // Cambia el estado de motion a false (no hay movimiento):
50     if (motionState == true) {
51       digitalWrite(BUZZER, LOW);
52       motionState = false;
53     }
54   }

```

#### 4.6. Código en Python

En esta codificación se toman los datos del puerto COM4 de Arduino, con el fin de conectarse a un broker y enviar dicha información al servidor que se guardará en formato CSV para posteriormente mostrarlas de manera gráfica.

```

ArduinoPython.py X
ArduinoPython.py > ...
1  import serial
2  import time
3  import random
4
5  from paho.mqtt import client as mqtt_client
6
7  ser = serial.Serial('COM4',9600)
8  ser.flushInput()
9
10 broker = 'broker.emqx.io'
11 port = 1883
12 topic = "dandelion"
13 # generate client ID with pub prefix randomly
14 client_id = f'python-mqtt-{random.randint(0, 1000)}'
15 def connect_mqtt():
16     def on_connect(client, userdata, flags, rc):
17         if rc == 0:
18             print("Connected to MQTT Broker!")
19         else:
20             print("Failed to connect, return code %d\n", rc)
21
22     client = mqtt_client.Client(client_id)
23     client.on_connect = on_connect
24     client.connect(broker, port)
25     return client

```

```

26
27 def publish(client):
28     while True:
29         time.sleep(3)
30         ser_bytes = ser.readline()
31         msg = str(ser_bytes.decode("utf-8")).replace("\r\n", "")
32         result = client.publish(topic, msg)
33         # result: [0, 1]
34         status = result[0]
35         if status == 0:
36             print(f"Send `{msg}` to topic `{topic}`")
37         else:
38             print(f"Failed to send message to topic {topic}")
39
40 def run():
41     client = connect_mqtt()
42     client.loop_start()
43     publish(client)
44
45
46 if __name__ == '__main__':
47     run()
48

```

## 5. Página Web

A continuación se muestra la página web denominada "Dandelion", su distribución de contenido referente al proyecto, inicio de sesión y la sala de monitoreo donde se mostrarán los datos de manera gráfica.

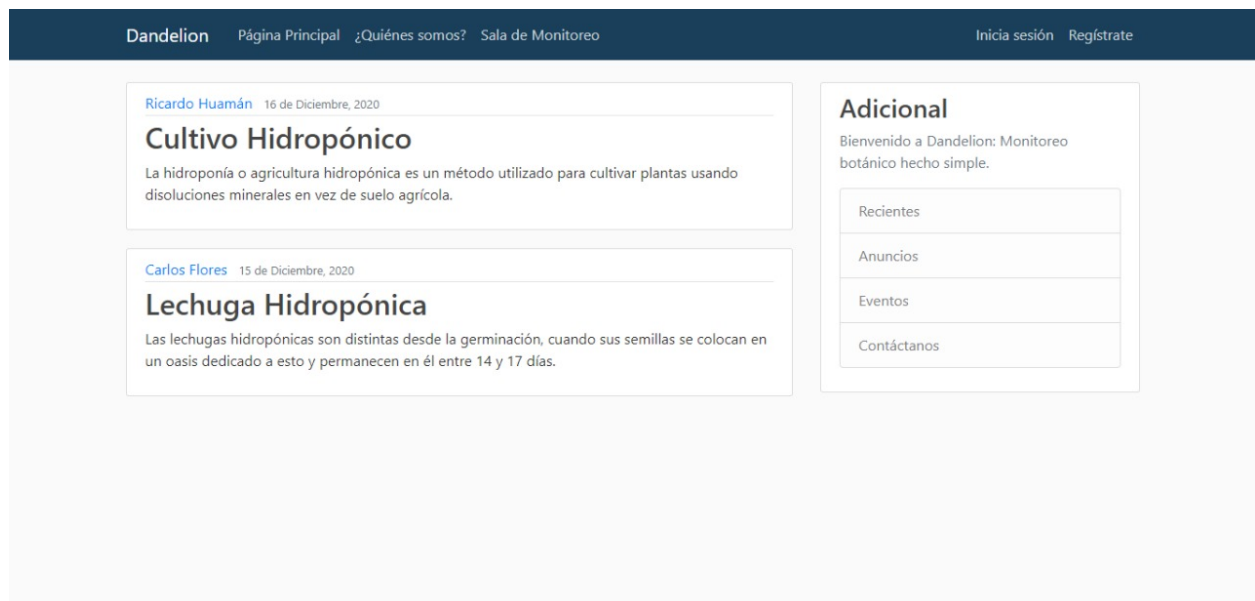


Figura 14. Página principal.

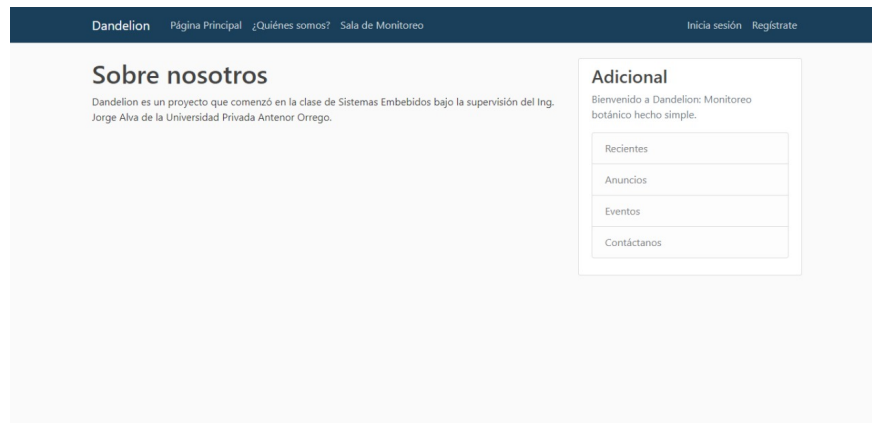


Figura 15. Información de la pagina.

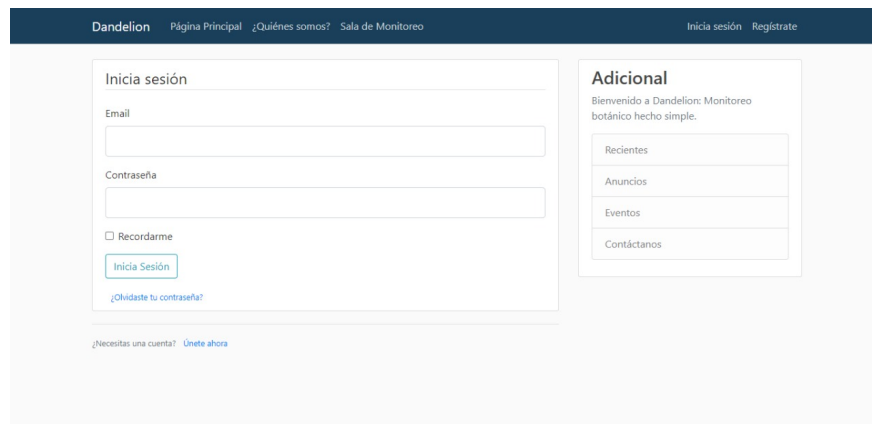


Figura 16. Inicio de sesión.

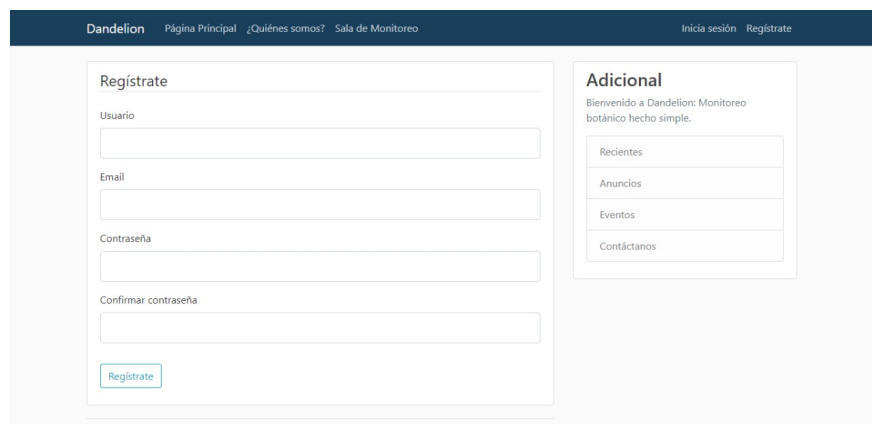


Figura 17. Registro con correo electrónico.

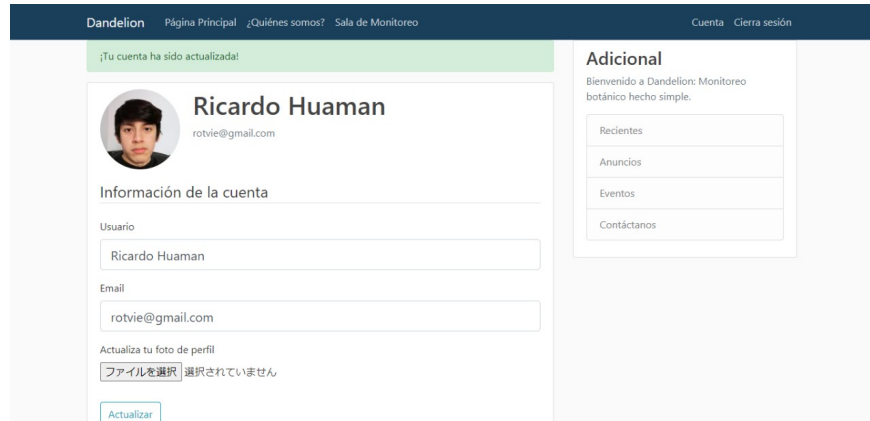


Figura 18. Ejemplo de cuenta registrada.



Figura 19. Gráfica obtenida con datos de humedad.



Figura 20. Gráfica obtenida con datos de temperatura.

## 6. Conclusiones

- La información de los sensores fue extraída con éxito y enviada al puerto COM4 de Arduino.
- Usando los lenguajes de programación mencionados se consiguió diseñar una página web.
- Se logró enviar los datos a la página web "Dandelion", donde se visualizarán gráficas de línea, permitiendo un mejor análisis para toma de decisiones más efectivas.
- La idea inicial de desarrollar un sistema de monitoreo agrario que cumpla con los requerimientos que le otorguemos, se ha llevado a cabo de manera satisfactoria.

## 7. Anexos

### Códigos en Python

Envío de datos: [https://github.com/Rotvie/Embedded-Systems/blob/main/MQTT\\_Send](https://github.com/Rotvie/Embedded-Systems/blob/main/MQTT_Send).

Recepción de datos: [https://github.com/Rotvie/Embedded-Systems/blob/main/MQTT\\_Receive](https://github.com/Rotvie/Embedded-Systems/blob/main/MQTT_Receive).

### Código Arduino

<https://github.com/Rotvie/Embedded-Systems/blob/main/Arduino>.



Anexo 1.



## Referencias

- [1] <https://blogespanol.se.com/industria/2019/03/26/la-automatizacion-como-lider-en-la-industria-este-2019/>
- [2] <https://acolita.com/sistema-de-monitoreo-satelital-de-cultivos/#:~:text=El%20sistema%20de%20monitoreo%20de,negativa%20y%20positiva%20de%20los>
- [3] <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html>
- [4] [https://www.naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html?search\\_query=node+mcu&results=43](https://www.naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html?search_query=node+mcu&results=43)
- [5] <https://www.naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>
- [6] <https://www.naylampmechatronics.com/interfaz-de-usuario/251-modulo-buzzer-pasivo.html>
- [7] <https://www.naylampmechatronics.com/sensores-proximidad/55-modulo-de-deteccion-pir-hc-sr501.html>
- [8] <https://www.naylampmechatronics.com/accesorios-y-prototipado/26-cable-para-protoboard.html>
- [9] [https://www.naylampmechatronics.com/protoboard/14-protoboard-400.html?search\\_query=protoboard&results=40](https://www.naylampmechatronics.com/protoboard/14-protoboard-400.html?search_query=protoboard&results=40)
- [10] [https://www.naylampmechatronics.com/luces-iluminacion/91-led-difuso-5mm.html?search\\_query=leds&results=40](https://www.naylampmechatronics.com/luces-iluminacion/91-led-difuso-5mm.html?search_query=leds&results=40)
- [11] [https://es.wikipedia.org/wiki/Proteus\\_Design\\_Suite](https://es.wikipedia.org/wiki/Proteus_Design_Suite)
- [12] <https://www.bejob.com/que-es-la-programacion-con-arduino-y-para-que-sirve/>
- [13] [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code)
- [14] <https://es.wikipedia.org/wiki/Python>
- [15] <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [16] <https://es.wikipedia.org/wiki/HTML>
- [17] [https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)
- [18] <https://en.wikipedia.org/wiki/SQLAlchemy>