

Sistema de monitoreo inteligente para plantas

Pedro Manuel Baldovino Soto (T00082206),
Karol Michel Alvarez Morales (T00082261),
Daniel Eduardo Rengifo Acosta (T00082476),
Diego Andrés Lafont Paternina (T00083627),
Jose David Quiñonez Rehenals (T00083600).

Facultad de Ciencia Básicas, Universidad Tecnológica de Bolívar.
Grupo: R Subgrupo: Feria_2
Cartagena, Colombia.

drengifo@utb.edu.co, pbaldovino@utb.com, dlafont@utb.edu.co, karoalvarez@utb.edu.co,
quinonezj@utb.edu.co.

RESUMEN

El sistema de monitoreo y riego inteligente para plantas aborda la problemática de la deshidratación de la flora, el exceso o falta de luz, la detección de plagas y los problemas de humedad del suelo. Utiliza inteligencia artificial para reconocer diversos tipos de plantas y cultivos, evaluando sus necesidades específicas de temperatura, luz y humedad. Implementa un modelo basado en las ecuaciones del movimiento parabólico, principio de Bernoulli y ecuación de continuidad para ajustar, en tiempo real, una gráfica del movimiento del agua, proyectada con un ángulo variable. Para ello, el sistema requiere un modelo de aprendizaje capaz de identificar la planta y calcular los ángulos necesarios, junto con una velocidad específica, para un riego óptimo. El sistema de riego, controlado por servomotores, ajusta una trayectoria parabólica precisa según las necesidades de cada planta.

Palabras claves: sistema hidráulico, principio de Bernoulli, ecuación de continuidad.

1. OBJETIVO GENERAL

Desarrollar un sistema de monitoreo independiente que permita reconocer las necesidades, características y condición de las plantas, utilizando principios de física para entender el movimiento y predecir trayectorias, optimizando así el sistema de riego basado en dichas ecuaciones.

2. OBJETIVOS ESPECÍFICOS

2.1 OBJETIVO ESPECÍFICO 1:

Implementar un sistema de sensores que permita medir las variables ambientales y las condiciones de la planta, tales como humedad y la temperatura. Para realizar un análisis en tiempo real de sus necesidades.

2.2 OBJETIVO ESPECÍFICO 2:

Aplicar las ecuaciones del movimiento parabólico y principios físicos relacionados, para calcular y ajustar la trayectoria del agua en el sistema de riego, optimizando el uso de recursos según las características específicas de cada planta.

3. MARCO TEÓRICO

La física mecánica es esencial para comprender y optimizar los sistemas de riego, ya que proporciona las herramientas necesarias para analizar el movimiento de fluidos y las fuerzas que influyen en su comportamiento. La dinámica de fluidos, que estudia cómo se comportan los líquidos y gases en movimiento, es crucial en el diseño de sistemas de riego eficientes que puedan optimizar el uso del agua en diversos ambientes.

Uno de los conceptos clave en la física del riego es el movimiento parabólico, que describe la trayectoria que sigue un objeto lanzado bajo la influencia de la gravedad. Este fenómeno se puede modelar mediante las ecuaciones de movimiento en dos dimensiones, que permiten predecir la distancia y la altura que alcanzará el agua al ser proyectada. Estas ecuaciones son esenciales para calcular los ángulos óptimos de lanzamiento que garantizan una distribución efectiva del agua sobre las plantas, donde nos guiaremos del libro *"Physics for Scientists and Engineers"* de Serway & Jewett.

Además, el principio de Bernoulli se puede aplicar en el contexto del riego, relacionando la presión, la velocidad y la altura del flujo de agua. Este principio menciona que, en un fluido en movimiento, la suma de la energía cinética, la energía potencial y la presión es constante. La comprensión de este principio permite diseñar sistemas que maximicen la presión del agua al salir por las boquillas, asegurando un riego uniforme y eficiente.

La ecuación de continuidad, que establece que el producto del área de la sección transversal de un conducto y la velocidad del fluido es constante, es otra herramienta esencial. Esto permite que los diseñadores de sistemas de riego ajusten las dimensiones de las tuberías y boquillas para mantener un flujo constante de agua, evitando pérdidas innecesarias.

El uso de sensores para monitorear las variables ambientales, como la temperatura y la humedad del suelo, introduce una dimensión adicional al control del riego. La recopilación y el análisis de datos en tiempo real permiten una gestión más precisa del riego, optimizando y mejorando la salud de las plantas.

Este marco teórico se basa en estos principios físicos y tecnológicos, proporcionando la base necesaria para entender el diseño y la implementación del sistema de monitoreo y riego inteligente que se propone en este proyecto.

3.1 ECUACIONES FUNDAMENTALES EN EL SISTEMA DE RIEGO INTELIGENTE:

El sistema implementado en este proyecto se basa en una serie de principios físicos que determinan cómo el agua se desplaza desde las bombas hasta las plantas, y cómo ajustar la trayectoria de este movimiento para así optimizar el riego según las necesidades detectadas por los sensores. Las ecuaciones presentadas a continuación reflejan los fenómenos físicos que intervienen en el diseño.

MOVIMIENTO PARABÓLICO:

El movimiento parabólico describe la trayectoria que es seguida un objeto cuando es lanzado en un ángulo respecto a la horizontal y está sujeto únicamente a la aceleración de la gravedad. Este fenómeno es un ejemplo de movimiento en dos dimensiones, donde el objeto posee tanto una velocidad horizontal como una vertical. Mientras que la velocidad en la dirección horizontal permanece constante (en ausencia de fricción del aire), la velocidad en la dirección vertical se ve afectada por la aceleración gravitacional. Para el modelado de la ecuación en cualquier instante debemos usar las fórmulas que describen el movimiento en horizontal y vertical para este tipo de parábolas y eliminar el tiempo t en ambas ecuaciones de la siguiente manera:

Ecuación del movimiento en el eje horizontal:

$$x = v_0 \cdot \cos(\phi) \cdot t$$

Ecuación del movimiento en el eje vertical:

$$y = v_0 \cdot \sin(\phi) \cdot t - \frac{1}{2} g \cdot t^2$$

De la ecuación de movimiento horizontal despejamos t :

$$t = \frac{x}{v_0 \cdot \cos(\phi)}$$

Ahora, reemplazamos este valor t en la ecuación del movimiento vertical:

$$y = v_0 \cdot \sin(\phi) \cdot \left(\frac{x}{v_0 \cdot \cos(\phi)} \right) - \frac{1}{2} g \cdot \left(\frac{x}{v_0 \cdot \cos(\phi)} \right)^2$$

Simplificando obtenemos que:

$$y = v_0 \cdot \tan(\phi) - \frac{g \cdot x^2}{2 \cdot v_0^2 \cdot \cos^2(\phi)}$$

Esta es la ecuación que describe la trayectoria parabólica del agua o cualquier otro proyectil lanzado sin resistencia del aire Donde:

y : Altura del chorro de agua en metros.

x : Distancia horizontal en metros.

ϕ : Angulo de lanzamiento del agua en radianes.

v_0 : Velocidad inicial del agua en metros por segundo (m/s).

g : Aceleración debido a la gravedad en metros sobre segundo al cuadrado (m/s^2).

PRINCIPIO DE BERNOULLI:

El principio de Bernoulli es una ley fundamental de la dinámica de fluidos que relaciona la presión, la velocidad y la altura en un fluido en movimiento. Según este principio, en un fluido incompresible y en flujo estacionario, la suma de la presión, la energía cinética y la energía potencial a lo largo de una línea de corriente es constante.

La ecuación de Bernoulli se expresa como:

$$P_1 + \frac{1}{2} \rho v_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho v_2^2 + \rho g h_2$$

Donde:

P_1 y P_2 son las presiones en los puntos 1 y 2 del flujo.

$\frac{1}{2} \rho v_1^2$ y $\frac{1}{2} \rho v_2^2$ son las energías cinéticas del fluido en los puntos 1 y 2 donde v son las velocidades respectivas.

$\rho g h_1$ y $\rho g h_2$ Son las energías potenciales gravitacionales del flujo en los puntos mientras h representa las alturas.

Y gracias a esas relaciones podemos saber que este principio implica que en un punto donde la velocidad del fluido aumenta, la presión disminuye, y viceversa. Este fenómeno es crucial para comprender cómo se comportan los fluidos cuando fluyen a través de diferentes secciones de una tubería con variaciones en el diámetro.

ECUACIÓN DE CONTINUIDAD:

La ecuación de continuidad es otro principio esencial en la dinámica de fluidos, el cual describe la conservación de la masa en un sistema de fluido que fluye. Si un fluido es incompresible, el producto del área de la sección transversal del conducto por la velocidad del flujo debe ser constante en todo el sistema.

La ecuación se expresa como:

$$A_1 v_1 = A_2 v_2$$

Donde:

A_1 y A_2 son las áreas de las secciones transversales en dos puntos diferentes del flujo.

v_1 y v_2 son las velocidades del fluido en esas secciones.

Esta ecuación refleja que, si el área por donde fluye el líquido disminuye, la velocidad del fluido debe aumentar para mantener un flujo constante. Es la base de muchos principios en la hidráulica y la aerodinámica.

3.2Tecnologías de Monitoreo y Control:

El sistema de monitoreo y control utiliza una combinación de sensores, placas de desarrollo y algoritmos para realizar una gestión eficiente del riego de las plantas. Este apartado describe las tecnologías clave involucradas en el sistema.

SENSORES DE MONITOREO:

Los sensores son dispositivos que convierten magnitudes físicas, como la temperatura o la humedad, en señales eléctricas que pueden ser interpretadas y procesadas por un sistema. En el contexto de un sistema de riego, los sensores permiten monitorizar continuamente las condiciones tanto del entorno como del suelo, proporcionando información crucial para tomar decisiones automáticas de riego.

- **Sensores DHT11 y DHT22:** Estos sensores se utilizan para medir la temperatura y humedad relativa del aire. El DHT11 es un sensor básico que ofrece lecturas de temperatura con una precisión de $\pm 2^\circ\text{C}$ y humedad con una precisión de $\pm 5\%$. En cambio, el DHT22 tiene una precisión más alta ($\pm 0.5^\circ\text{C}$ en temperatura y $\pm 2\%$ en humedad), lo que lo hace más adecuado

para aplicaciones que requieren mayor exactitud. Estos sensores son ampliamente utilizados en proyectos de monitoreo ambiental debido a su bajo costo y facilidad de uso.

- **Funcionamiento:** Utilizan un componente capacitivo para medir la humedad del aire y un termistor para medir la temperatura. Los cambios en la capacitancia o resistencia se convierten en una señal digital que puede ser procesada por un microcontrolador.
- **Sensores de Humedad del Suelo(higrómetro):** Estos sensores miden la humedad del suelo, lo que es fundamental para determinar cuándo regar las plantas. Los sensores de humedad utilizan dos electrodos que miden la resistencia eléctrica en el suelo. A medida que el suelo se seca, la resistencia entre los electrodos aumenta, lo que indica una menor humedad. La tecnología de estos sensores permite obtener una estimación directa de la cantidad de agua presente en el suelo.

PLACAS DE DESARROLLO:

Las placas de desarrollo son microcontroladores que permiten la integración de sensores y actuadores, y la ejecución de programas para controlar sistemas automatizados. En este proyecto, se emplean dos placas principales: el ESP32 y el Arduino Uno.

- **ESP32:** Es una plataforma de desarrollo con capacidad Wi-Fi y Bluetooth, lo que la hace ideal para aplicaciones de Internet de las Cosas (IoT). En este caso, el ESP32 se utiliza para recopilar datos de los sensores y enviar esta información a un servidor externo para su procesamiento. Además de su capacidad de comunicación inalámbrica, el ESP32 posee un procesador de 32 bits y varios pines de entrada/salida, lo que lo convierte en una herramienta poderosa para aplicaciones de control y monitoreo.
 - **Wi-Fi y Comunicación:** El ESP32 permite la transmisión de datos mediante Wi-Fi, lo que facilita la conexión con una aplicación en Python a través de un socket. Este tipo de comunicación es esencial para mantener el sistema actualizado en tiempo real sin necesidad de cables, reduciendo la latencia y permitiendo la monitorización remota.
- **Arduino Uno:** El Arduino Uno es una de las placas de desarrollo más comunes y versátiles en proyectos de control. Con un procesador ATmega328, el Arduino Uno es ideal para controlar actuadores como servomotores y bombas de agua, gestionando el hardware físico del sistema. En este proyecto, se encarga de recibir las instrucciones de Python y ajustar los actuadores según las necesidades de riego de la planta.

Actuadores y Control del Flujo:

Un actuador se define como un dispositivo que convierte señales eléctricas en movimiento o acciones físicas. En el sistema de riego propuesto, los actuadores utilizados incluyen servomotores y bombas de agua sumergibles, que permiten ajustar la trayectoria del agua y regular la cantidad de riego.

- **Servomotores:** Los servomotores son motores que permiten un control preciso del ángulo de rotación. En este caso, se utilizan para ajustar la trayectoria parabólica del agua mediante el control de los ángulos, que varían en función de las necesidades hídricas de cada planta. El ángulo de salida del agua determina la cobertura del riego, y puede ajustarse en tiempo real con base en los datos recibidos por los sensores.
 - **Funcionamiento:** Los servomotores funcionan mediante modulación por ancho de pulso (**PWM**), donde la duración del pulso determina el ángulo de rotación del motor. Esto es fundamental en aplicaciones que requieren un control preciso del movimiento, como el ajuste de la dirección de los chorros de agua en un sistema de riego.
- **Bombas de Agua:** Las bombas de agua sumergibles son responsables de extraer el agua y conducirla a través de las tuberías hasta las plantas. En un sistema de riego automatizado, las bombas se activan y desactivan en función de los datos de humedad del suelo y otros parámetros definidos. Estas bombas operan con **voltaje constante** y su capacidad para generar presión es fundamental para que el agua alcance la distancia y altura necesarias.

Procesamiento y Visualización de Datos

Una parte esencial de los sistemas inteligentes de monitoreo y control es el procesamiento y análisis de los datos obtenidos por los sensores. En este proyecto, se emplea Python como plataforma principal para realizar los cálculos necesarios y visualizar las trayectorias y condiciones ambientales en tiempo real.

- **Python:** Es un lenguaje de programación de alto nivel, ampliamente utilizado en el ámbito científico y técnico por su simplicidad y versatilidad. Python ofrece una gran cantidad de librerías que permiten realizar tareas avanzadas, como el procesamiento de datos, la simulación numérica y la generación de gráficos. En este proyecto, Python se encarga de recibir los datos de los sensores enviados por el **ESP32** y procesar la información en tiempo real.

Python también facilita la integración de diferentes tipos de tecnologías y dispositivos mediante el uso de sockets (Un **socket** es un punto de comunicación que permite la transferencia de datos entre dispositivos en una red) para comunicación Wi-Fi y puerto serial para la interacción con Arduino. Además, Python se destaca por ser una herramienta eficiente en términos de cálculo

y procesamiento de datos, lo que lo convierte en una opción ideal para sistemas como el empleado en este tipo de proyectos.

- **Matplotlib:** Es una librería de Python especializada en la creación de gráficos y visualizaciones. En este proyecto, se utiliza para generar gráficos en tiempo real que muestran las trayectorias parabólicas del agua proyectadas por los servomotores, así como las variables ambientales clave, como la humedad del suelo y la temperatura. Esta librería permite visualizar estos datos de manera continua, lo que es fundamental para analizar el comportamiento dinámico del sistema de riego y hacer ajustes en tiempo real.

El uso de **Matplotlib** facilita la comprensión de los datos a través de representaciones gráficas, lo que mejora la capacidad de identificar patrones y tomar decisiones sobre el riego adecuado para las plantas.

En conjunto, Python y Matplotlib proporcionan una potente plataforma de desarrollo para realizar simulaciones y monitoreo en tiempo real, ofreciendo herramientas avanzadas para el análisis de datos y la visualización, aspectos clave en un sistema de riego automatizado.

4. MONTAJE EXPERIMENTAL Y SIMULACIÓN

El montaje experimental del sistema de monitoreo y riego inteligente tiene como objetivo implementar y validar, en un entorno controlado, los principios teóricos descritos en el marco teórico del proyecto. Se integran tecnologías de sensores, microcontroladores y actuadores con fundamentos físicos aplicados al movimiento de fluidos, para optimizar el riego de las plantas según sus necesidades específicas.

Este montaje corresponde al presentado a continuación:

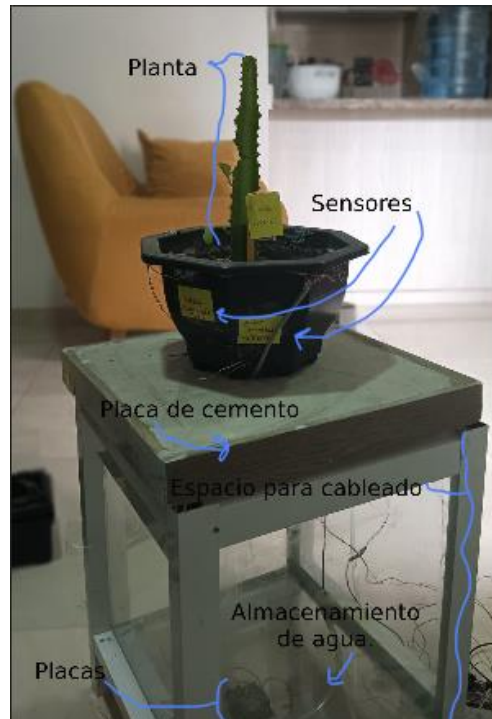


Imagen 1. Montaje experimental de la propuesta.

Desglosando el montaje desde la parte superior de este hasta la inferior, se tiene:

- **Planta monitoreada:** La planta a monitorear con el sistema es una trigona.
- **Sensores:** Los sensores dispuestos en el montaje recogen los datos para posteriormente ser analizados y usados para llevar a cabo la acción más adecuada para la planta.
- **Placa de cemento:** La placa de cemento da soporte y rigidez al montaje. Se escogió este material debido a la facilidad de preparación en una forma definida. Esta placa tiene un orificio en el centro para el filtrado de agua y reúso de esta. También cabe resaltar que la placa puede causar un desnivel que facilita el flujo del agua hacia el orificio.
- **Espacio para cableado:** El cableado se encuentra dentro de los soportes metálicos del montaje para mejorar la presentación del proyecto y evitar accidentes
- **Espacio para almacenamiento de agua:** El espacio dentro de las láminas de cristal se dispone para almacenar agua.
- **Placas:** Las placas se encuentran bajo el almacenamiento de agua en una plataforma de cristal. Los cables se conectan a las placas y recorren el espacio dispuesto para ellos, es decir, los soportes del montaje, como se había aclarado anteriormente.

Se realizaron las conexiones necesarias para el funcionamiento del sistema, utilizando el microcontrolador ESP32 y las bombas de agua. El ESP32 se programó para controlar las operaciones del sistema y fue alimentado a través de un adaptador conectado a la red eléctrica de 110V. Las

bombas de agua también fueron alimentadas mediante el mismo adaptador, asegurando un suministro de energía constante para el funcionamiento óptimo del sistema.

Esta configuración permite que las bombas sean controladas de manera remota y automática, minimizando la intervención manual y optimizando la eficiencia energética del dispositivo.

El sistema se divide en tres partes fundamentales:

- **Obtención de Datos:** Captura de variables ambientales y del suelo mediante sensores.
- **Procesamiento:** Análisis de los datos y cálculo de acciones necesarias utilizando software.
- **Acción:** Ejecución de las tareas de riego ajustadas a través de actuadores.

Esquema General del Sistema:

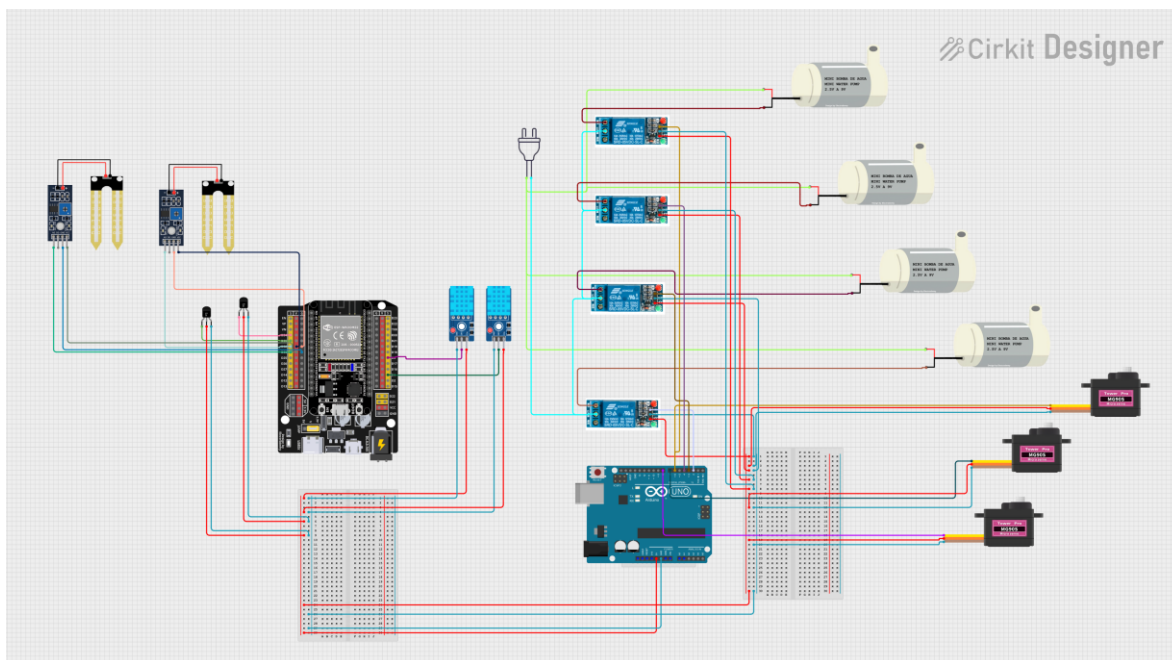


Figura 1. Diseño esquemático del sistema general diseñado en Cirkuit Designer

4.1. Instalación y Configuración de Sensores

4.1.1. Sensor DHT11 (Temperatura y Humedad)

Descripción:

El DHT11 es un sensor digital que proporciona lecturas de temperatura y humedad relativa. Adecuado para aplicaciones básicas de monitoreo ambiental, este se incluyó en el montaje de la siguiente manera.



Imagen 2. Sensor DHT11.

- **Especificaciones Técnicas:**
 - **Rango de Temperatura:** 0°C a +50°C
 - **Precisión de Temperatura:** $\pm 2^{\circ}\text{C}$
 - **Rango de Humedad Relativa:** 20% a 90% RH
 - **Precisión de Humedad:** $\pm 5\% \text{ RH}$
 - **Voltaje de Operación:** 3V a 5.5V

- **Instalación Física:**

El sensor debe colocarse en una ubicación representativa del ambiente donde se encuentran las plantas, evitando exposición directa a la luz solar intensa o a corrientes de aire que puedan distorsionar las mediciones.

- **Conexiones Eléctricas:**

Pinout del DHT11:

PIN	Función	Conexión
1	VCC	3.3V a 5V
2	DATA	Pin Digital
3	GND	Tierra / negativo

Tabla 1. Pinout referente al dht11

Resistencia Pull-up: Se recomienda conectar una resistencia de 10kΩ entre VCC y DATA para asegurar una señal estable.

- Esquema de Conexión:**

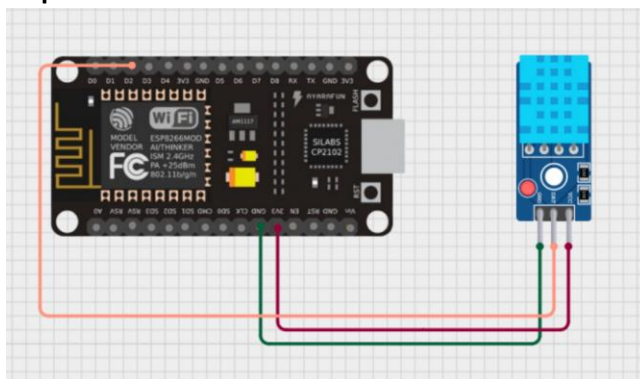


Figura 2. Conexiones DHT11 a ESP32

4.1.2. Sensor de Humedad del Suelo

- Descripción:**

El sensor de humedad del suelo es un dispositivo que mide la cantidad de agua presente en el sustrato mediante la detección de cambios en la conductividad eléctrica entre dos electrodos.

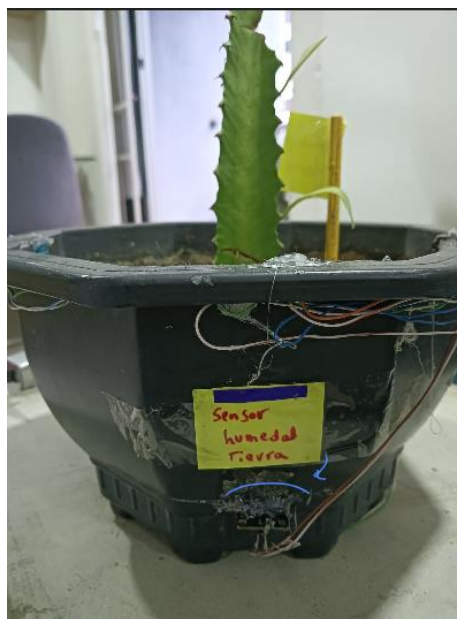


Imagen 3. Sensor de humedad tierra.

- **Especificaciones Técnicas:**

- **Tipo:** Sensor resistivo
- **Voltaje de Operación:** 3.3V o 5V
- **Salida:** Analógica (voltaje proporcional a la humedad)

- **Ubicación Óptima:**

- Se insertan las sondas en el suelo cerca de las raíces de la planta, evitando zonas con piedras o materiales que puedan dañar los electrodos.
- Hay que asegurar que las sondas estén completamente cubiertas por el sustrato para obtener mediciones precisas.

- **Conexiones Eléctricas:**

Pinout del Sensor de Humedad del Suelo:

PIN	Función	Conexión
1	VCC	3.3V
2	GND	Tierra / negativo
3	AOUT	Pin Analógico

Tabla 1. Pinout referente al sensor de humedad a tierra

- **Esquema de Conexión:**

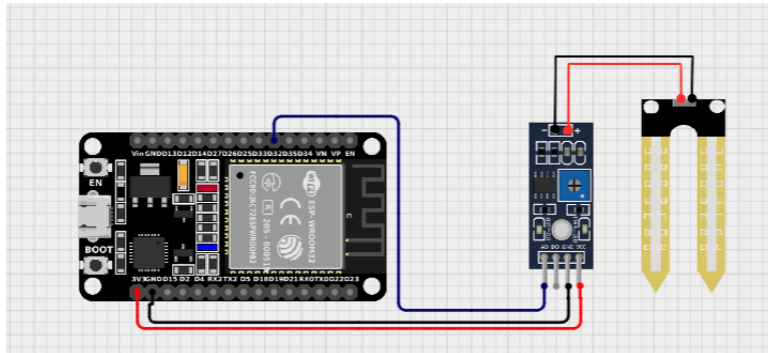


Figura 3. Conexiones sensor humedad a ESP32

4.1.3 Configuración del Microcontrolador para Lectura de Sensores ESP32

- **Configuración Inicial:**
 - **Entorno de Desarrollo:** Utilizar Arduino IDE.
 - **Configuración del ESP32 en Arduino IDE:**
 - Añadir el URL del gestor de tarjetas:
`https://dl.espressif.com/dl/package_esp32_index.json` en las preferencias.
 - Instalar las placas ESP32 desde el gestor de tarjetas.
- **Programación para Adquisición de Datos:**
 - **Lectura del DHT22:**
 - Incluir la librería DHT.h.
 - Definir el tipo de sensor y el pin de datos.
 - Inicializar el sensor en el setup().
 - Leer temperatura y humedad en el loop().



FACULTAD DE CIENCIAS BÁSICAS



Universidad Tecnológica de Bolívar

```
#include <DHT.h>

#define DHTPIN 15
#define DHTTYPE DHT11 // Cambiado a DHT11

DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  float humedad = dht.readHumidity();
  float temperatura = dht.readTemperature();

  // Verificar si las lecturas son válidas
  if (isnan(humedad) || isnan(temperatura)) {
    Serial.println("Error al leer el DHT11");
  } else {
    // Enviar datos
  }
  delay(2000);
}
```

Imagen 1. Código usado para procesar DHT11 desde ARDUINO IDE

- **Lectura del Sensor de Humedad del Suelo:**
 - Definir el tipo de sensor y el pin de datos.
 - Leer humedad en el loop().
 - Si se desea obtener el valor de %de humedad se deberá mapear de digital a %de humedad

```
int pinHumedadSuelo = 34; // GPIO 34

void loop() {
    int valorHumedadSuelo = analogRead(pinHumedadSuelo);
    // Mapear o convertir el valor si es necesario
    // Enviar datos
    delay(2000);
}
```

Imagen 2. Código usado para procesar sensor de humedad desde ARDUINO IDE

4.2. Desarrollo del Software de Procesamiento:

4.2.1. Recepción de Datos desde el ESP32

- **Implementación de la Comunicación Wi-Fi en Python:**

Para que el sistema funcione correctamente, es esencial establecer una comunicación eficiente entre el ESP32 y el software de procesamiento en Python que se ejecuta en el ordenador. Esta comunicación se realizará mediante sockets TCP/IP a través de la red Wi-Fi local.

Pasos para implementar la comunicación:

1. **Configuración del Servidor en Python:**

El primer paso es crear un servidor en Python que escuche en un puerto específico y esté preparado para aceptar conexiones entrantes desde el ESP32.

```
import socket
import json

# Definir la dirección y el puerto en el que el servidor escuchará
HOST = '' # Escucha en todas las interfaces disponibles
PORT = 12345 # Puerto que utilizaremos para la comunicación

# Crear el socket del servidor
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((HOST, PORT))
server_socket.listen(1) # Escuchar hasta 1 conexión entrante

print('Servidor iniciado. Esperando conexiones...')
```

Imagen 3. Código usado crear servidor en Python

2. **Aceptación de Conexiones y Recepción de Datos:**

Una vez que el servidor está en escucha, debe aceptar conexiones y recibir datos enviados por el ESP32.


```
while True:
    conn, addr = server_socket.accept() # Aceptar una nueva conexión
    print(f'Conectado por {addr}')
    data = conn.recv(1024) # Recibir datos (máximo 1024 bytes)
    if not data:
        break # Si no se reciben datos, cerrar la conexión
    try:
        # Decodificar los datos recibidos
        datos = json.loads(data.decode())
        temperatura = datos['temperatura']
        humedad = datos['humedad']
        humedad_suelo = datos['humedadSuelo']
        print(f"Temperatura: {temperatura} °C, Humedad: {humedad} %, Humedad
    except json.JSONDecodeError:
        print("Error al decodificar los datos JSON")
    conn.close() # Cerrar la conexión
```

Imagen 4. Código para leer valores de sensores en Python

3. Manejo de Excepciones y Errores:

Es importante manejar posibles errores en la comunicación, como interrupciones en la conexión o datos mal formateados.

```
try:
    # Código para aceptar conexiones y recibir datos
except KeyboardInterrupt:
    print("Servidor detenido por el usuario")
    server_socket.close()
except Exception as e:
    print(f'Ocurrió un error: {e}')
    server_socket.close()
```

Imagen 5. Código para hacer manejar posibles errores

4.2.2. Configuración del Cliente en el ESP32:

En el ESP32, se debe configurar el código para conectarse al servidor y enviar los datos de los sensores en formato JSON.



FACULTAD DE CIENCIAS BÁSICAS



Universidad Tecnológica de Bolívar

```
#include <WiFi.h>
#include <DHT.h>

// Configuración del Wi-Fi
const char* ssid = "Tu_SSID";
const char* password = "Tu_Contraseña";

// Dirección IP y puerto del servidor
const char* host = "IP_DEL_SERVIDOR"; // Reemplazar con la IP del ordenador
const uint16_t port = 12345;

// Configuración del DHT11
#define DHTPIN 15
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Pin del sensor de humedad del suelo
const int pinHumedadSuelo = 34;

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

  // Conexión al Wi-Fi
  WiFi.begin(ssid, password);
  Serial.print("Conectando al Wi-Fi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConectado al Wi-Fi");
}
```

Imagen 6. Setup para enviar leer datos y enviarlos desde esp32

```
void loop() {
    // Lectura de los sensores
    float temperatura = dht.readTemperature();
    float humedad = dht.readHumidity();
    int valorHumedadSuelo = analogRead(pinHumedadSuelo);

    // Verificación de las lecturas
    if (isnan(temperatura) || isnan(humedad)) {
        Serial.println("Error al leer el DHT11");
        return;
    }

    // Creación del JSON con los datos
    String datosJSON = "{";
    datosJSON += "\"temperatura\":" + String(temperatura, 2) + ",";
    datosJSON += "\"humedad\":" + String(humedad, 2) + ",";
    datosJSON += "\"humedadSuelo\":" + String(valorHumedadSuelo);
    datosJSON += "}";

    // Conexión al servidor
    WiFiClient client;
    if (!client.connect(host, port)) {
        Serial.println("Conexión al servidor fallida");
        delay(2000);
        return;
    }

    // Envío de los datos
    client.print(datosJSON);
    Serial.println("Datos enviados: " + datosJSON);

    // Cierre de la conexión
    client.stop();

    // Esperar antes de la siguiente lectura
    delay(5000); // Lecturas cada 5 segundos
}
```

Imagen 7. Loop donde se leen los datos y se envían

4.2.3. Procesamiento de Datos y Cálculos Físicos:

En esta sección, se detallará el procesamiento físico necesario para determinar la velocidad inicial del agua (V_0) que permitirá alcanzar la distancia deseada al regar la planta, aplicando la ecuación de continuidad y el principio de Bernoulli. Además, se utilizará la ecuación del movimiento parabólico para graficar la trayectoria del agua.

- **Determinación de las necesidades de riego:**

Establecer un **umbral de humedad** para decidir cuándo es necesario regar. Por ejemplo, si la humedad del suelo es menor al 40%, se activará el sistema de riego.

```

UMBRALE_HUMEDAD_SUELO = 40.0 # Umbral en porcentaje

if humedad_suelo < UMBRALE_HUMEDAD_SUELO:
    necesita_riego = True
    print("La planta necesita riego")
else:
    necesita_riego = False
    print("La planta no necesita riego")

```

Imagen 8. Establecer umbral de humedad en Python

4.2.4. Cálculo de la Velocidad Inicial V_0 Aplicando el Principio de Bernoulli y la Ecuación de Continuidad

Calcular la velocidad inicial V_0 del agua que sale de la boquilla para alcanzar la distancia horizontal X necesaria para regar la planta.

- **Aplicación de la Ecuación de Continuidad**

La ecuación establece que, para un fluido incompresible en flujo estacionario, el caudal (Q) es constante en todas las secciones del conducto:

$$Q = A_1 v_1 = A_2 v_2$$

$$v_2 = 4.71 \text{ m/s}$$

- **Aplicación del Principio de Bernoulli**

El principio relaciona la presión, la velocidad y la altura en dos puntos de un fluido en movimiento:

$$P_1 + \frac{1}{2} \rho v_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho v_2^2 + \rho g h_2$$

Despejando v_2 ,

$$v_2 = \sqrt{2g(h_1 - h_2)}$$

h_1 es la altura de agua en el depósito

h_2 es la altura de la boquilla

Si se utiliza una bomba:

La velocidad inicial V_0 dependerá de la presión generada por la bomba y del diámetro de la boquilla.

Cálculo del Caudal Necesario

Podemos relacionar el caudal (Q) con el área de la boquilla (A) y la velocidad inicial

$$Q = A \cdot v_0 = 2l/min$$

Si conocemos el caudal que proporciona la bomba (Q), podemos calcular V_0 o ajustar el área de la boquilla (A).

Al considerar la presión generada por la bomba (P):

$$v_0 = \sqrt{\frac{2P}{\rho}}$$

Siendo ρ la densidad del fluido

Ya con una velocidad inicial podemos iniciar a graficar usando la fórmula de movimiento parabólico:

$$y = v_0 \cdot \tan(\phi) - \frac{g \cdot x^2}{2 \cdot v_0^2 \cdot \cos^2(\phi)}$$

Una vez obtenido todos los datos podemos conseguir hacer la gráfica de manera sencilla con Python:

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros
g = 9.81 # Aceleración de la gravedad en m/s^2
v0 = 5.0 # Velocidad inicial en m/s (puede ajustarse según el sistema)
theta_grados = 45 # Ángulo de lanzamiento en grados (puede ajustarse)
theta = np.radians(theta_grados) # Convertir a radianes

# Generar un rango de X (distancia horizontal) desde 0 hasta una distancia máxima
X = np.linspace(0, 5, num=500) # Por ejemplo, de 0 a 5 metros

# Calcular Y (altura) en función de X
Y = X * np.tan(theta) - (g * X**2) / (2 * v0**2 * np.cos(theta)**2)

# Filtrar valores válidos (donde Y >= 0)
indices_validos = Y >= 0
X = X[indices_validos]
Y = Y[indices_validos]

# Graficar la trayectoria
plt.figure(figsize=(10, 6))
plt.plot(X, Y)
plt.xlabel('Distancia Horizontal X (m)')
plt.ylabel('Altura Y (m)')
plt.title(f'Trayectoria del Agua para  $\theta = \{theta\_grados\}^\circ$ ,  $v_0 = \{v0\}$  m/s')
plt.grid(True)
plt.show()
```

Imagen 9. Graficar usando matplotlib

Ya con la lógica de la grafica hecha, solo queda pasar el angulo optimo a los servomotores del mediante el Arduino y pc para ajustarlos y tener esa grafica en nuestro sistema de riego.

```
if angulo_optimo is not None and necesita_riego:
    angulo_envio = angulo_optimo # En grados
    tiempo_riego = 5 # Duración del riego en segundos (puede ajustarse)

    # Enviar comandos al Arduino
    comando = f"{angulo_envio},{tiempo_riego}\n"
    arduino.write(comando.encode())
    print(f"Comando enviado al Arduino: {comando}")
else:
    print("No se enviaron comandos al Arduino.")
```

Imagen 10. Envío de ángulo optimo a Arduino.

Recibimos los datos en el Arduino de la siguiente manera:

```
void loop() {
    if (Serial.available()) {
        String input = Serial.readStringUntil('\n');
        int indice = input.indexOf(',');
        if (indice != -1) {
            String anguloStr = input.substring(0, indice);
            String tiempoStr = input.substring(indice + 1);
            float angulo = anguloStr.toFloat();
            int tiempoRiego = tiempoStr.toInt();

            // Mover el servomotor al ángulo calculado
            servo.write(angulo);

            // Activar la bomba durante el tiempo especificado
            digitalWrite(pinBomba, HIGH);
            delay(tiempoRiego * 1000); // Convertir segundos a milisegundos
            digitalWrite(pinBomba, LOW);
        }
    }
}
```

Imagen 11. Recibo de datos en el loop del arduino.

4.3. Control de Actuadores:

En esta sección, detallaremos el funcionamiento y la integración de los actuadores en el sistema de riego inteligente: las bombas de agua sumergibles y los servomotores.

Proporcionaremos las especificaciones técnicas, las conexiones eléctricas y el control de estos dispositivos, tal como lo hicimos con los sensores al comienzo del documento.

4.3.1. Bombas Sumergibles:

Las bombas de agua sumergibles son dispositivos eléctricos diseñados para impulsar agua desde un depósito hacia el sistema de riego. En este proyecto, se utilizan para proporcionar el flujo y la presión necesarios para el riego de las plantas, controlando la cantidad de agua suministrada según las necesidades detectadas.

Especificaciones Técnicas

- **Modelo:** Mini Bomba de Agua Sumergible DC 3V-6V
- **Voltaje de Operación:** 3V - 6V DC
- **Consumo de Corriente:** 130mA - 220mA (dependiendo del voltaje)
- **Caudal Máximo:** 80 - 120 litros/hora
- **Altura Máxima de Bombeo:** 40 - 110 cm
- **Diámetro de Salida:** 5 - 7 mm
- **Dimensiones:** Aproximadamente 45mm x 24mm x 30mm
- **Cable de Alimentación:** Dos conductores (rojo para positivo y negro para negativo)

Conexiones Eléctricas

Las bombas de agua sumergibles requieren una fuente de alimentación externa que proporcione el voltaje y corriente adecuados, ya que el Arduino no puede suministrar la potencia necesaria directamente. Para controlar la bomba desde el Arduino, utilizaremos un módulo de relé que actuará como interruptor, permitiendo encender y apagar la bomba mediante señales de control de bajo voltaje.

Control mediante Relés

Un relé es un dispositivo electromecánico que permite controlar circuitos de alta potencia con señales de baja potencia. En este caso, el relé aislará el circuito de control (Arduino) del circuito de potencia (bomba), proporcionando seguridad y protegiendo los componentes electrónicos.

Conexiones:

1. **Conexión del Relé al Arduino:**
 - **IN (Señal de Control):** Conectar al pin digital 8 del Arduino.
 - **VCC del Relé:** Conectar al pin de 5V del Arduino.
 - **GND del Relé:** Conectar al pin GND del Arduino.
2. **Conexión de la Bomba al Relé y Fuente de Alimentación:**

- **Terminal COM (Común) del Relé:** Conectar al positivo (+) de la fuente de alimentación externa.
- **Terminal NO (Normalmente Abierto) del Relé:** Conectar al cable positivo (+) de la bomba.
- **Cable Negativo (-) de la Bomba:** Conectar al negativo (-) de la fuente de alimentación.
- **Negativo de la Fuente de Alimentación:** También se puede conectar al GND del Arduino para referencia común (opcional y solo si las tensiones son compatibles).

Diagrama de conexión:

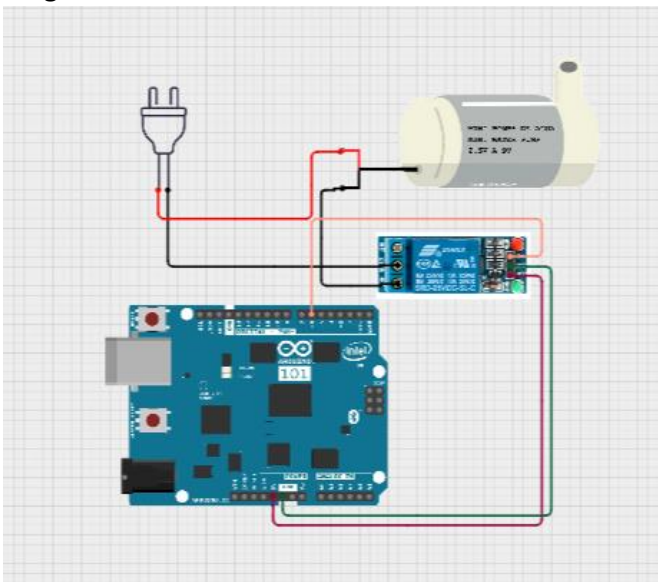


Figura 4. Montaje de bomba de agua con modulo relé.

Control desde el Arduino:

El Arduino controlará la bomba enviando señales al módulo de relé. Cuando el pin de control está en ALTO (HIGH), el relé activa el circuito de la bomba, permitiendo su funcionamiento. Cuando está en BAJO (LOW), el circuito se abre y la bomba se apaga.

```
const int pinBomba = 8; // Pin digital conectado al IN del relé

void setup() {
  pinMode(pinBomba, OUTPUT);
  digitalWrite(pinBomba, LOW); // Asegurar que la bomba esté apagada al inicio
}

void loop() {
  // Encender la bomba
  digitalWrite(pinBomba, HIGH); // Activar el relé
  delay(5000); // Mantener la bomba encendida durante 5 segundos

  // Apagar la bomba
  digitalWrite(pinBomba, LOW); // Desactivar el relé
  delay(10000); // Esperar 10 segundos antes de repetir el ciclo
}
```

Imagen 12. Código para manejo de bombas con Arduino.

4.3.2. Servomotores

Descripción:

Los servomotores son dispositivos que permiten controlar la posición angular de su eje con precisión. En este proyecto, se utilizan para ajustar el ángulo de la boquilla de riego, modificando la dirección y alcance del chorro de agua según los cálculos realizados en el procesamiento.

Especificaciones Técnicas:

- **Modelo:** Servomotor MG90 Micro Servo
- **Voltaje de Operación:** 4.8V - 6V DC
- **Ángulo de Rotación:** 0° - 180°
- **Par Máximo:** 2.2 kg-cm (4.8V), 2.5 kg-cm (6.0V)
- **Velocidad de Operación:** 0.10 sec/60° (4.8V), 0.08 sec/60° (6.0V)
- **Dimensiones:** 22.2 mm x 11.8 mm x 31 mm
- **Peso:** 13.4 gramos

Conexiones:

PIN	Función	Conexión
1	GND	Tierra
2	VCC	4.8V a 6V
3	Señal	Pin PWM

Tabla 3. Pinout sobre los servomotores

Conexiones Eléctricas

El servomotor requiere una señal PWM para controlar su posición y una alimentación adecuada. Aunque el Arduino puede proporcionar el voltaje necesario, es recomendable usar una fuente de alimentación externa si se utilizan varios servos o si el consumo supera la capacidad del regulador del Arduino.

Diagrama de conexión:

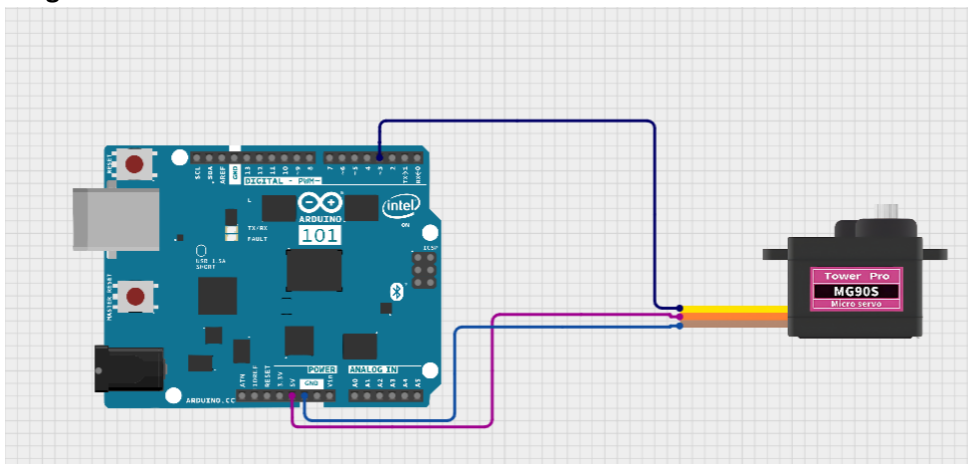


Figura 4. Conexiones del servomotor.

Control desde el Arduino

El Arduino utiliza la librería **Servo.h** para controlar el servomotor, enviando señales PWM que ajustan la posición del eje según los ángulos especificados.

```
#include <Servo.h>

Servo servoMotor; // Crear un objeto servo

const int pinServo = 9; // Pin digital conectado al servo

void setup() {
    servoMotor.attach(pinServo); // Asociar el servo al pin
}

void loop() {
    // Mover el servo a 0 grados
    servoMotor.write(0);
    delay(1000);

    // Mover el servo a 90 grados
    servoMotor.write(90);
    delay(1000);

    // Mover el servo a 180 grados
    servoMotor.write(180);
    delay(1000);
}
```

Imagen 13. Código para manejo deservomotores con Arduino.

4.3.3. Integración de Actuadores en el sistema:

Después de configurar y probar individualmente la bomba y el servomotor, se procede a integrarlos en el sistema controlado por el Arduino, permitiendo una interacción coordinada según las instrucciones recibidas desde el software de procesamiento.

```
#include <Servo.h>

Servo servoMotor; // Objeto para controlar el servomotor

const int pinServo = 9;    // Pin conectado al servomotor
const int pinBomba = 8;    // Pin conectado al relé de la bomba

void setup() {
    Serial.begin(9600);
    servoMotor.attach(pinServo);
    pinMode(pinBomba, OUTPUT);
    digitalWrite(pinBomba, LOW); // Asegurar que la bomba esté apagada al inicio
}

void loop() {
    if (Serial.available()) {
        String input = Serial.readStringUntil('\n');
        int indice = input.indexOf(',');
        if (indice != -1) {
            String anguloStr = input.substring(0, indice);
            String tiempoStr = input.substring(indice + 1);
            float angulo = anguloStr.toFloat();
            int tiempoRiego = tiempoStr.toInt();

            // Mover el servomotor al ángulo especificado
            servoMotor.write(angulo);

            // Activar la bomba durante el tiempo especificado
            digitalWrite(pinBomba, HIGH); // Encender la bomba
            delay(tiempoRiego * 1000);    // Mantener la bomba encendida
            digitalWrite(pinBomba, LOW);  // Apagar la bomba
        }
    }
}
```

Imagen 14. Código con integración de valores dados por Python para el Arduino.

Con la integración detallada de las bombas de agua sumergibles y los servomotores, se completa el montaje experimental del sistema de riego inteligente. Los actuadores permiten ejecutar las acciones calculadas en el procesamiento, ajustando el riego de manera óptima y automatizada según las necesidades detectadas por los sensores.

5. BIBLIOGRAFÍA

- [1] Rodríguez Rodríguez, Álvaro. Diseño e implementación de un sistema electrónico de riego remoto y automático. 2023. Universitat Politècnica de València.
- [2] <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/> ODS de aquí los objetivos de desarrollo sostenible
- [3] <https://repositorio.utb.edu.co/handle/20.500.12585/9574#page=1>. Abuchar Curi, A., Coronado Hernández, O. E., & Julio Rada, F. E. (2016). Mecánica de Fluidos. Universidad Tecnológica de Bolívar.
- [4] Canal AGRICULTURA ELECTRONICA (24 de enero de 2020). HUERTO INTELIGENTE ARDUINO: DHT11, humedad suelo, bomba agua, sensor de nivel y LCD. [Archivo de Vídeo]. <https://youtu.be/MdCUvPTvpCo?si=bCiVSwSx8kd-HcxN>.
- [5] Canal SalvaTronic (22 de junio de 2024). Sistema de riego automático con ARDUINO. [Archivo de Vídeo]. <https://youtu.be/odD9rxIVZns?si=WwP3uQsJaerjne64>.
- [6] Young, H. D., Sears, F. W., & Freedman, R. A. (2012). Física Universitaria: Volumen 1. Pearson Educación. <https://ebooks724.utb.elogim.com/stage.aspx?il=4142&pg=&ed=>
- [7] Briceño V., Gabriela. (2018). Principio de Bernoulli. Recuperado el 24 febrero, 2024, de Euston96: <https://www.euston96.com/principio-de-bernoulli/>
- [8] AbrahamG. (2020, June 1). Cómo utilizar un sensor de humedad de suelo con Arduino. Automatización para todos. <https://www.automatizacionparatodos.com/sensor-de-humedad-de-suelo-con-arduino/>
- [9] All Data Sheet. (s.f.). DHT11 Humidity & Temperature Sensor. Hoja de datos de DHT11 [Archivo PDF]. <https://www.alldatasheet.com/datasheet-pdf/view/1440068/ETC/DHT11.htm>
- [10] All Data Sheet. (s.f.). [DHT22](#) Humidity & Temperature Sensor. Hoja de datos de [DHT22](#) [Archivo PDF]. <https://www.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>

- [11]Circuit Designer. (s.f.). Servomotor MG90S. How to Use Servomotor MG90S: Pinouts, Specs, and Examples | Circuit Designer. <https://docs.circuitdesigner.com/component/1a6c3c31-c0aa-4bbb-8e18-523d09da0773>
- [12]Circuit Designer. (s.f.). Humedad YL-69. How to Use Humidity YL-69: Pinouts, Specs, and Examples | Circuit Designer. <https://docs.circuitdesigner.com/component/983b91b0-610d-47cb-8b80-a246d92e3608>

6. Anexo

- **Anexo A: Prueba de parábola con bomba de agua según su presión:**



Imagen 15. Prueba de parábola con ángulo fijo y salida de agua constante – Bomba con baja presión.

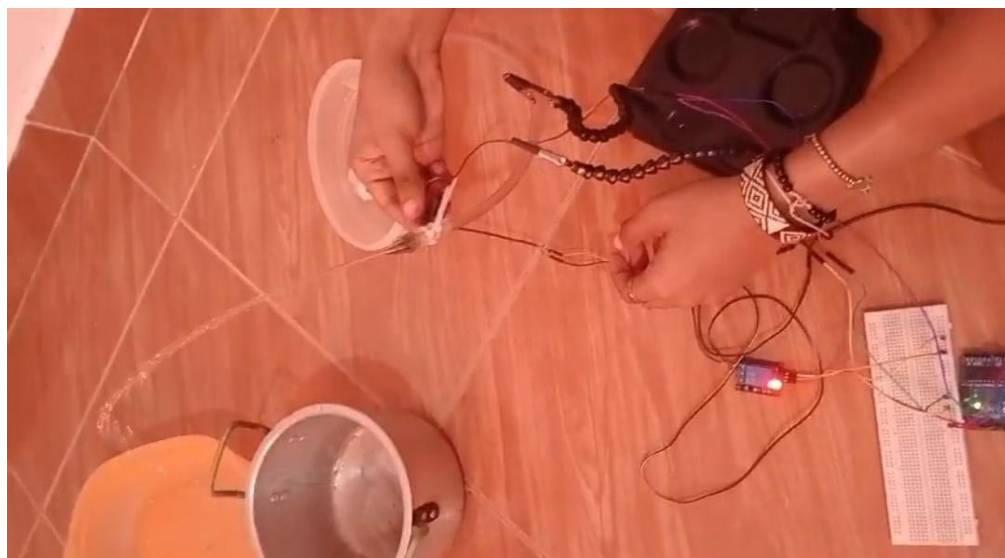


Imagen 16. Prueba de parábola con ángulo fijo y salida de agua intermitente – bomba con presión intermedia.



Imágenes 17 y 18. Prueba de parábola con ángulos variables y salida de agua intermitente – Bomba con alta presión.