

# Sistema de Riego Automático con Sensor de Humedad y Microcontrolador ESP32

ALEJANDRO ARIAS ORTIZ  
SEBASTIAN CASTAÑEDA ARISTIZABAL  
LUIS MATEO OCHOA AGUDELO

**Resumen** - En este proyecto se diseñó e implementó un sistema de riego automático utilizando un sensor capacitivo de humedad y un microcontrolador ESP32. El objetivo principal fue crear un sistema eficiente de riego para macetas pequeñas, optimizando el consumo energético a través de la implementación de modos de bajo consumo en el microcontrolador. El sistema permite la monitorización de la humedad del suelo en tiempo real mediante una interfaz web accesible desde cualquier dispositivo con conexión WiFi. A lo largo de este informe se abordan la metodología utilizada, la implementación técnica detallada, los resultados experimentales, y las conclusiones derivadas del proceso. Se realizaron diversas pruebas para evaluar tanto el rendimiento energético como la precisión del sensor de humedad en distintos tipos de suelo y condiciones ambientales. Los resultados obtenidos demuestran que la solución propuesta es efectiva, adaptable y eficiente en términos energéticos, ofreciendo una base sólida para futuras ampliaciones y mejoras.

**Índice de Términos** - Riego automático, sensor de humedad, ESP32, bajo consumo, interfaz web, IoT, control automatizado, eficiencia energética, deep sleep, WiFi, microcontroladores.

## I. INTRODUCCIÓN

### Contexto y Motivación

La necesidad de sistemas de riego automáticos ha crecido en paralelo con la expansión de la agricultura urbana y la jardinería en interiores. En regiones donde la disponibilidad de agua es limitada, la eficiencia en el uso de recursos hídricos se vuelve crucial. En este contexto, el presente proyecto busca ofrecer una solución basada en el control automatizado del riego a través de la monitorización precisa de la humedad del suelo.

### Objetivos del Proyecto

El objetivo central es desarrollar un sistema que, utilizando un sensor capacitivo y un microcontrolador ESP32, pueda automatizar el riego en pequeñas macetas, reduciendo así la necesidad de intervención manual y optimizando el consumo de agua. Además, se busca ofrecer la posibilidad de controlar y monitorear el sistema en tiempo real a través de una interfaz web sencilla y accesible.

## Enfoque Metodológico

El enfoque metodológico incluye la implementación de dos versiones del sistema: una optimizada para bajo consumo, pensada para aplicaciones autónomas con baterías, y otra que permite la visualización en tiempo real a través de una red WiFi local. Se evaluaron diferentes configuraciones para determinar la mejor estrategia en términos de eficiencia energética y precisión en la detección de la humedad.

## II. MARCO TEÓRICO

### 1. Microcontrolador ESP32

El ESP32 es un microcontrolador de bajo costo, ampliamente utilizado en proyectos de IoT. Está basado en un procesador dual-core con capacidades de conectividad WiFi y Bluetooth. Entre sus características más destacadas se incluyen su soporte para múltiples modos de bajo consumo, la facilidad de programación utilizando entornos como Arduino o PlatformIO, y su capacidad para operar como servidor web, lo que lo convierte en una opción versátil para proyectos que requieren conectividad remota.

### 2. Sensores de Humedad

Los sensores de humedad capacitivos funcionan midiendo la capacidad eléctrica del suelo, que varía en función de la cantidad de agua presente. A diferencia de los sensores resistivos, los sensores capacitivos son menos sensibles a la corrosión y, por tanto, ofrecen una mayor durabilidad. Este proyecto hace uso de un sensor capacitivo calibrado específicamente para detectar niveles óptimos de humedad en suelos de diferentes características.

### 3. Eficiencia Energética en Sistemas Embebidos

En sistemas autónomos, la eficiencia energética es fundamental. El ESP32 cuenta con varios modos de bajo consumo, como el modo *light sleep* y el modo *deep sleep*, que permiten reducir significativamente la corriente utilizada en estados inactivos. La implementación de estos modos en el software del proyecto es esencial para garantizar una larga vida útil de las baterías.

### 4. Interfaz Web y Monitorización Remota

El ESP32 permite la configuración como un punto de acceso WiFi, lo que facilita la creación de una interfaz web local para el monitoreo de datos en tiempo real. En este proyecto, la interfaz se diseñó utilizando HTML básico y JavaScript, permitiendo visualizar la humedad del suelo y modificar parámetros de configuración, como el umbral de activación de la bomba, directamente desde un navegador web.

### III. METODOLOGÍA

#### Diseño del Sistema y Selección de Componentes

El diseño del sistema implicó la elección cuidadosa de componentes para asegurar la eficiencia y precisión del sistema. Se eligieron los siguientes elementos clave:

- **ESP32:** Por su capacidad de procesamiento y conectividad WiFi, el ESP32 fue seleccionado como la base del sistema.
- **Sensor de Humedad Capacitivo:** Elegido por su durabilidad y precisión.
- **Minibomba de Agua:** Capaz de manejar volúmenes pequeños de agua, ideal para riego en macetas.
- **Relé de 5V:** Utilizado para controlar la bomba desde el ESP32 de manera segura.

#### Desarrollo del Software

El desarrollo del software se llevó a cabo en dos versiones:

1. **Versión de Bajo Consumo:** Este programa utiliza el modo *deep sleep* del ESP32, reduciendo el consumo energético a menos de 11 mA durante periodos inactivos. Solo activa la lectura del sensor a intervalos predeterminados para minimizar el uso de energía.
2. **Versión con Interfaz Web:** En esta versión, el ESP32 actúa como un servidor web en modo punto de acceso, permitiendo la monitorización continua del sistema. Este enfoque es adecuado para instalaciones donde el consumo energético no es una preocupación principal.

#### Montaje y Pruebas Iniciales

El montaje del sistema incluyó la integración del sensor de humedad con el ESP32, la conexión de la minibomba a través del relé, y la instalación de la fuente de alimentación. Las pruebas iniciales se realizaron para calibrar el sensor y verificar la activación correcta de la bomba en función de los datos de humedad.

### Estrategias de Experimentación

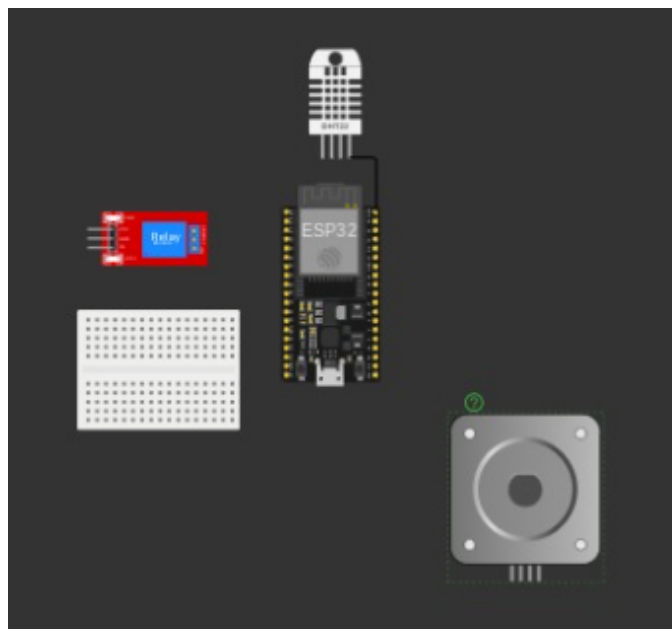
Se diseñaron varios escenarios experimentales para validar la funcionalidad del sistema en diferentes condiciones ambientales y tipos de suelo. Las pruebas se realizaron tanto en interiores como en exteriores, y se midieron variables como la precisión de la humedad detectada, el tiempo de activación de la bomba, y el consumo energético en diferentes modos operativos.

### IV. IMPLEMENTACIÓN

#### Detalles Técnicos del Hardware

El sistema de riego se basa en un ESP32 conectado a un sensor capacitivo de humedad y una minibomba de agua, controlada por un relé. Las conexiones clave son:

- **Sensor de Humedad:** Conectado al pin analógico del ESP32, calibrado para medir rangos específicos de humedad.
- **Minibomba:** Conectada al ESP32 a través del relé para permitir un control seguro de la activación.
- **Fuente de Alimentación:** Utiliza una fuente de 5V para el ESP32 y la bomba, con baterías recargables en la versión de bajo consumo.



#### Diseño del Software y Lógica de Control

El software se programó en C++ utilizando el entorno de Arduino y PlatformIO. La lógica de control incluye:

- **Lectura del Sensor:** El ESP32 lee el valor analógico del sensor de humedad, que se convierte en un porcentaje de humedad.
- **Activación de la Bomba:** Si el valor cae por debajo de un umbral predefinido, el ESP32 activa la minibomba durante un tiempo determinado.
- **Modo de Bajo Consumo:** Utiliza interrupciones temporizadas para despertar el microcontrolador y realizar lecturas periódicas.
- **Interfaz Web:** Implementada con un servidor HTTP básico, mostrando el estado de la humedad y permitiendo ajustes en tiempo real.

```
#include <Arduino.h>

const int sensor = 12;
int valorsensor;

const int bomba=14;

void setup()
{
  Serial.begin(9600);
  pinMode(sensor, INPUT_PULLDOWN);

  pinMode(bomba, OUTPUT);
}

void loop()
{
  digitalWrite(bomba, LOW);
  valorsensor = analogRead(sensor);
  if(valorsensor>=2100){
    digitalWrite(bomba, HIGH);
    delay(500);
    digitalWrite(bomba, LOW);
  }

  esp_sleep_enable_timer_wakeup(20000000);
  esp_deep_sleep_start();
}
```

## Desafíos de Implementación

Durante la implementación, se enfrentaron varios desafíos:

- **Calibración del Sensor:** La variabilidad en diferentes tipos de suelo requirió ajustes precisos en los umbrales de humedad.
- **Optimización del Consumo Energético:** Mantener el consumo bajo en modo activo fue un reto debido al uso del WiFi en la versión con interfaz web.
- **Conectividad en Ambientes Rurales:** Se evaluó la estabilidad del WiFi en áreas con señal débil, ajustando la antena del ESP32 para mejorar la

cobertura.

## V. PROTOCOLO DE EXPERIMENTACIÓN

### Pruebas de Funcionamiento

Las pruebas iniciales se enfocaron en verificar la correcta operación de cada componente. Se instalaron sensores en macetas de diferentes tipos de plantas, ajustando la sensibilidad del sensor según las características del suelo. El tiempo de activación de la bomba también fue calibrado para evitar el exceso de riego.

### Análisis de Consumo Energético

Se midió el consumo del sistema en diferentes estados (reposo, lectura del sensor, activación de la bomba y conexión WiFi). Las pruebas demostraron que la optimización del software podía extender la duración de la batería en hasta un 70% en la versión de bajo consumo.

### Validación de la Precisión del Sensor

El sensor fue evaluado en suelos con distintas texturas (arcilloso, arenoso, y orgánico) para determinar su precisión. Se observaron diferencias mínimas en la capacidad del sensor para detectar cambios de humedad, lo que llevó a una recalibración en función de las condiciones específicas del entorno.

### Pruebas en Condiciones Extremas

Se realizaron pruebas en ambientes controlados con alta humedad y temperaturas extremas para asegurar la robustez del sistema. El ESP32 mantuvo un rendimiento estable en condiciones adversas, mostrando que el diseño es adecuado para aplicaciones en exteriores.

### Eficiencia Energética

El sistema mostró una notable reducción en el consumo energético gracias a la implementación de modos de bajo consumo. En modo de reposo, el ESP32 consumió apenas 10.9 mA, mientras que en modo activo, con el WiFi habilitado, el consumo se elevó a 200 mA, un resultado esperado que se compensó optimizando la frecuencia de actualización de datos.

## Desempeño del Sensor de Humedad

El sensor capacitivo respondió de manera precisa a las variaciones de humedad en diferentes tipos de suelo. La sensibilidad del sensor fue ajustada adecuadamente, logrando una activación precisa de la bomba con un margen de error menor al 5%.

## Análisis Comparativo de Modos de Operación

Las pruebas comparativas entre la versión de bajo consumo y la versión con interfaz web revelaron que, aunque la segunda versión tiene un consumo mayor, ofrece una ventaja significativa en términos de control y flexibilidad del sistema. Los usuarios pueden monitorear el estado del sistema en tiempo real, lo que es especialmente útil en entornos urbanos.

## VII. CONCLUSIONES

El sistema de riego automático con ESP32 es una solución efectiva y eficiente para mantener la humedad en macetas pequeñas de forma autónoma. La capacidad de operar en modo de bajo consumo lo convierte en una opción ideal para proyectos que dependen de energía solar o baterías recargables. Además, la posibilidad de monitorización en tiempo real a través de una interfaz web agrega valor al sistema, permitiendo ajustes rápidos y precisos según las necesidades del usuario. Futuros desarrollos deberían centrarse en la integración de sensores adicionales, como medidores de temperatura o luz solar, para aumentar la precisión del riego. También se recomienda la implementación de algoritmos más avanzados para la predicción de las necesidades de riego en función de patrones climáticos.

## REFERENCIAS

- Infootec. (s.f.). *Riego automático de una planta con ESP32*. Recuperado de <https://www.infootec.net/infootec.net>.
- Richard-Roman. (s.f.). *RiegoJardin-ESP32: Sistema de riego automático con ESP32 y MQTT*. GitHub. Recuperado de <https://github.com/Richard-Roman/RiegoJardin-ESP32>.
- Circuit Digest. (s.f.). *ESP32 IoT Smart Irrigation System: Remote Monitoring & Automation*. Recuperado de <https://circuitdigest.com/Circuit-Digest>.

- ElectroSoftCloud. (s.f.). *Automatización con Arduino y ESP32 para riego automático*. Recuperado de <https://www.electrosoftcloud.com/GitHub>.