FACULTAD 50 AÑOS DE INGENIERÍA UNaM
JIDeTEV 2024 ISSN 2591-4219
XIV Jornadas de Investigación, Desarrollo Tecnológico, Extensión, Vinculación y Muestra de la Producción

Sistema de monitoreo y control de variables ambientales utilizando nodos remotos de bajo coste y Raspberry Pi3 como servidor central

Corina M. Feltan [a *], Angelo Silke [a], Gerardo M. Iurinic[a], Aldo L, Caballero[a], Leandro J. Corrado[a], Cristian M. Meinl [a], Cristian Cegeski[a] , Eduardo A. Cirera[b]

[a] *Universidad Nacional de Misiones, Facultad de Ingeniería, Oberá, Misiones, Argentina.*
[b] *Universidad Nacional del Nordeste, Facultad de Ingeniería, Resistencia, Chaco, Argentina..*
e-mails: corina.feltan@fio.unam.edu.ar

Resumen

Este trabajo presenta los avances sobre un sistema de monitoreo y control de la humedad del suelo, temperatura y humedad ambiental. A través del mismo se dispone de reportes en vivo de las condiciones del suelo y permite la corrección automática a valores requeridos, además de poder visualizar esta información desde cualquier parte del mundo. El propósito del sistema es proporcionar una solución eficiente de medición y monitoreo del suelo mediante tecnología de Internet de las Cosas (IoT), y mostrar estos datos a través de una aplicación web colaborativa de fácil acceso y rápida consulta.

En este proyecto se utiliza una microcomputadora Raspberry Pi 3 B+ con un sistema operativo basado en Linux. Se han configurado contenedores Docker que alojan diversos servicios encargados de recolectar, administrar y visualizar los datos enviados por sensores autónomos remotos, específicamente ESP32 equipados con sensores de temperatura y humedad del suelo (DHT11). Esto permite monitorear de manera eficiente las condiciones del suelo en tiempo real.

Palabras Clave *– IOT, Raspberry Pi 3, ESP32, Monitoreo remoto, Temperatura y humedad del suelo.*

1 Introducción

Los cambios climáticos tienen un impacto significativo en una amplia gama de actividades humanas. Para el sector agrícola es crucial realizar un monitoreo riguroso del suelo debido a su profunda influencia en el crecimiento, desarrollo y productividad de los cultivos. Además, afecta la incidencia de plagas y enfermedades, así como las necesidades de agua y fertilizantes {1]. Por lo tanto, es imperativo que el sistema agrícola disponga de datos actualizados en tiempo real para maximizar la producción de cultivos y frutas. Además el uso de sensores de humedad del suelo para la irrigación puede ayudar a reducir el consumo de agua y energía, así como los riesgos de contaminación de las aguas subterráneas [2-3]. En la provincia de Misiones, el sector agrícola no solo es una fuente significativa de ingresos, sino que también desempeña un papel crucial en la economía general de la provincia y la región.

2 Metodología

Este trabajo tiene como objetivo contribuir al diseño y desarrollo de un sistema de monitoreo y control de variables ambientales mediante nodos remotos económicos y un servidor central basado en la microcomputadora Raspberry Pi 3 B+. El sistema está diseñado para la recolección de datos de humedad del suelo y de temperatura y humedad ambiental actualizados en tiempo real en una

*corina.feltan@fio.unam.edu.ar

aplicación web y para actuar en consecuencia corrigiendo a valores óptimos configurados por el usuario, con el propósito de mejorar la producción de cultivos y frutas.

Para este propósito, se emplea un sensor remoto autónomo que mide las condiciones del suelo en el área de cultivo y presenta los resultados en una interfaz web intuitiva. Esto facilita a los agricultores la preparación activa para alcanzar altos rendimientos en la producción agrícola.

En la Fig. 1 se presenta un diagrama del funcionamiento del sistema. El nodo remoto, conectado a la misma red que el servidor mediante WiFi, recibe los datos de los sensores de temperatura y humedad incorporados en la placa de desarrollo y los envía al servidor central a través del protocolo MQTT. Estos datos se procesan y se almacenan en una base de datos SQL.

Posteriormente, un usuario puede acceder a un dominio HTTPS desde un navegador para visualizar toda la información enviada por los sensores. Esta información es almacenada y gestionada por el servidor central.

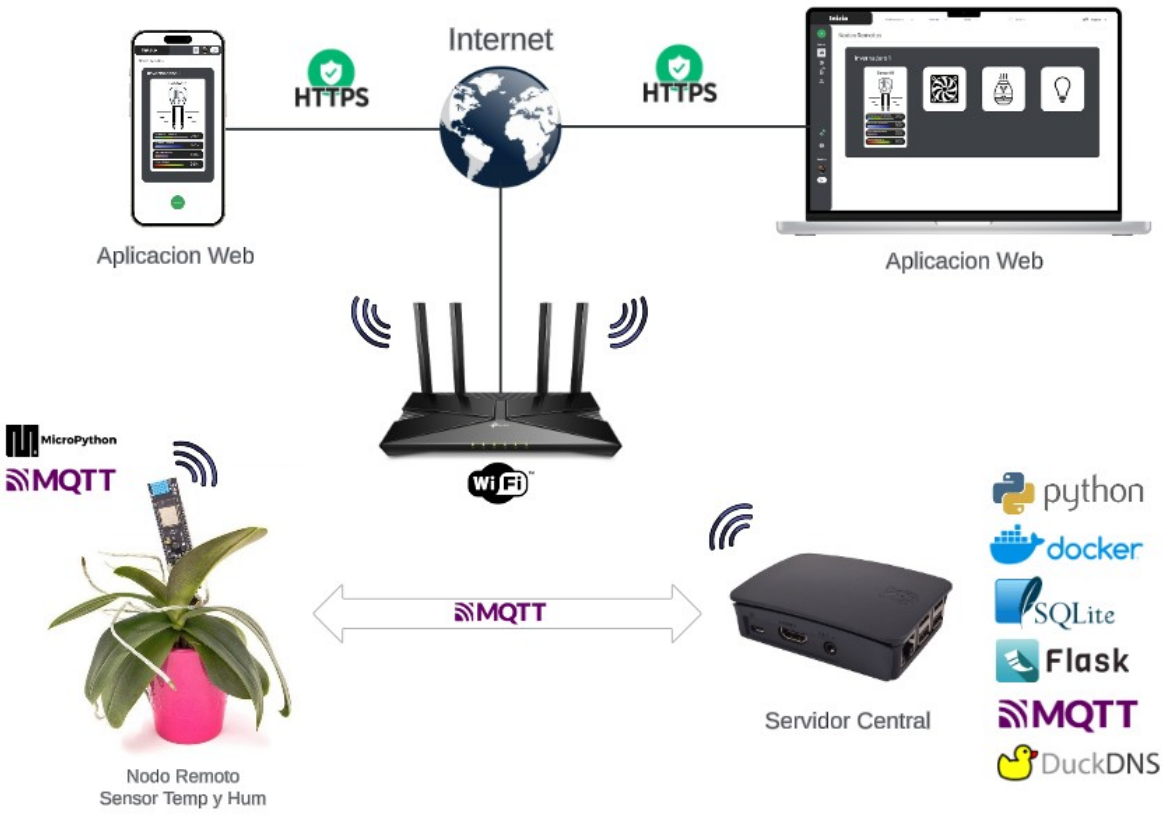

Fig. 1. Diagramado del sistema completo.

2.1 Servidor Central

El servidor central es un dispositivo encargado de recopilar la información de los nodos remotos, almacenarla y mostrarla a través de una aplicación web.

- Hardware: RaspBetty Pi 3 B+

La Raspberry Pi 3 B + que se muestra en la Fig. 2 es una computadora desarrollada por la Fundación Raspberry Pi, reconocida por su tamaño compacto, costo accesible y gran versatilidad. Está equipado con un procesador ARM Cortex-A53 de cuatro núcleos a 1.2 GHz y 1 GB de RAM, ofreciendo un equilibrio óptimo entre rendimiento y eficiencia energética. Además, integra WiFi (802.11n) y Bluetooth 4.1 BLE, lo que simplifica la conectividad inalámbrica y la interacción con dispositivos externos. Con puertos USB 2.0, HDMI, Ethernet y ranura para tarjeta microSD.

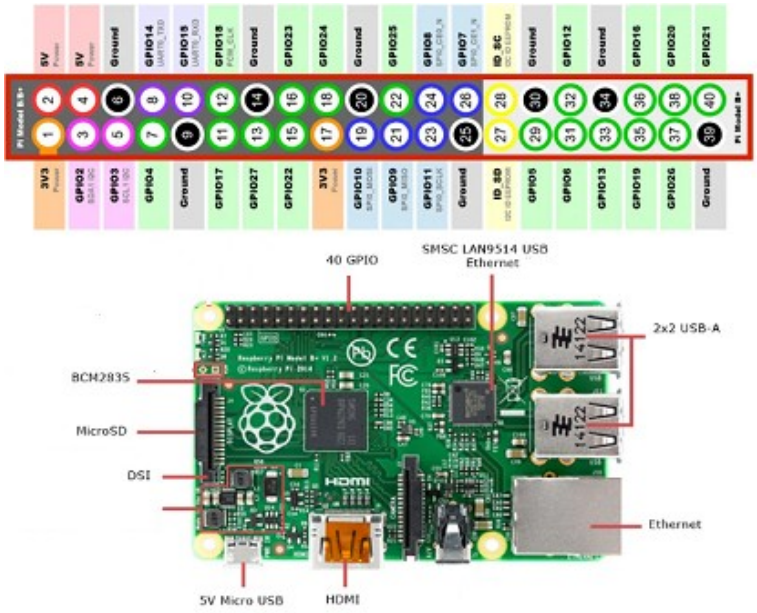

Fig. 2. Placa Raspberry Pi 3.

- Software:

Se instaló el sistema operativo Raspberry Pi OS Lite en la tarjeta de memoria SD para iniciar la configuración del sistema. Python se utilizó como herramienta principal para desarrollar el servidor MQTT, administrar la base de datos SQL y crear scripts para la recolección, procesamiento y visualización de datos. Se implementaron funciones asincrónicas con una librería especializada para manejar múltiples tareas de manera eficiente.

Primero, se instaló Docker para gestionar y ejecutar contenedores, lo que permite aislar las aplicaciones y sus dependencias, asegurando un entorno de ejecución consistente. A continuación, se desplegó un servidor o broker MQTT en uno de los contenedores, que maneja la comunicación de datos entre los sensores ESP32 y el servidor central, utilizando el protocolo MQTT para la transmisión eficiente de datos.

Luego, se implementó una base de datos SQL en otro contenedor para almacenar los datos de humedad del suelo, temperatura y humedad ambiental recolectados por los sensores. Esta base de datos estructurada permite realizar consultas eficientes tanto de datos históricos como en tiempo real.

Con esta configuración, el sistema puede recolectar datos de los sensores, almacenarlos en la base de datos y ponerlos a disposición a través del servidor MQTT o para su visualización en tiempo real mediante la aplicación. Los scripts desarrollados, junto con las funciones asincrónicas, facilitan la

integración y gestión de las diversas tareas del sistema, garantizando una operación eficiente y confiable.

2.2 **Nodos Remotos**

Sensor de temperatura y humedad ambiental y humedad del suelo (ESP32 DHT11)

- Hardware:

  Esta placa de desarrollo que se puede observar en la Fig 3. llamada HiGrow diseñada y fabricada por WEMOS integra un sensor capacitivo de humedad de suelo, un sensor de temperatura y humedad ambiente DHT11 y un ESP32, un chip desarrollado por Expressif Systems que incorpora capacidades integradas de WiFi y Bluetooth. Este dispositivo de bajo consumo cuenta con un procesador de doble núcleo Tensilica Xtensa, soporte para Bluetooth clásico y BLE, así como diversas interfaces periféricas, lo que lo hace ideal para proyectos IoT [5].

  El ESP32 sirve como un controlador robusto para aplicaciones IoT agrícolas y ambientales, complementando diversos sensores de humedad del suelo para facilitar el monitoreo agrícola y la automatización del riego [6].

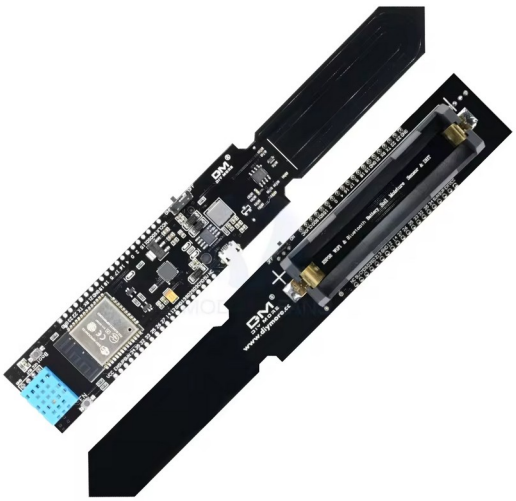

Fig. 3. Módulo sensor ESP32 (DHT11).

- Software: La programación del ESP32 se realiza en código Python, utilizando el firmware de MicroPython instalado previamente en el dispositivo. Este firmware facilita la programación y la implementación de funciones avanzadas.
  La comunicación de datos se realiza mediante el protocolo MQTT (Message Queuing Telemetry Transport), un estándar ligero y eficiente ideal para dispositivos IoT. Los datos recolectados por los sensores son publicados en tópicos MQTT y enviados al servidor central, donde se procesan y almacenan para luego poder ser visualizados en la aplicación web.

El software del ESP32 está diseñado con rutinas asincrónicas de python, permitiendo manejar múltiples tareas de manera eficiente sin bloquear la ejecución del código. Esto asegura una recolección y transmisión de datos fluida y continua. Para optimizar el consumo de energía, se recomienda el uso del modo DeepSleep que integra el ESP32. Este modo permite que el dispositivo entre en un estado de bajo consumo cuando no está recolectando o transmitiendo datos, prolongando así la vida útil de la batería y mejorando la eficiencia del sistema.

3 Resultados y discusión

Una vez que se programaron el nodo remoto y el servidor central, se estableció correctamente la interconexión entre los dispositivos, lo que permitió la recopilación de datos recibidos a través de MQTT, como se muestra en la Fig. 4.

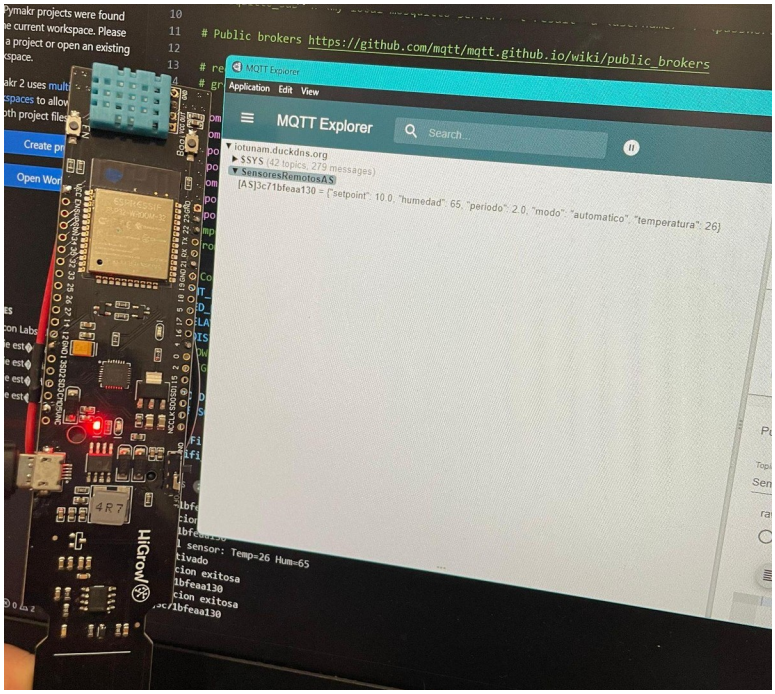

Fig. 4. ESP32 enviando datos por MQTT al servidor.

Como se ilustra en la Fig. 5, se utilizó el software Figma para diseñar una maqueta de la aplicación web destinada a la visualización de datos y el control de los nodos remotos. Esta aplicación, que se aloja en la Raspberry Pi, permitirá la visualización de los datos del sistema a través de un navegador web..

Cabe destacar que esta maqueta representa una fase preliminar del desarrollo de la aplicación web, la cual se construirá utilizando tecnologías como Flask, React y Bootstrap.

Fig. 5. Maquetado de la aplicación web utilizando Figma.

Modificaciones hechas en el nodo remoto:

- **Medición de la Batería:** Dado que la placa de desarrollo utilizada no admite baterías de gran capacidad, es crucial monitorear continuamente el estado de la batería para evitar interrupciones en el funcionamiento y prevenir su degradación prematura. Para ello, se implementó el módulo descrito en la Fig. 6, que emplea el circuito integrado MAX17043. Este componente permite medir con precisión el estado de la batería y activar interrupciones cuando el nivel de carga alcanza un umbral crítico. La información sobre el estado de la batería se transmite al microcontrolador ESP32 a través de la interfaz I2C.
  A diferencia de métodos alternativos, como el uso de un divisor resistivo para la medición de la batería, este módulo presenta un consumo energético prácticamente nulo, lo que contribuye a mejorar la autonomía del sistema en su conjunto. La Fig. 7 ilustra la implementación del módulo en el sistema.

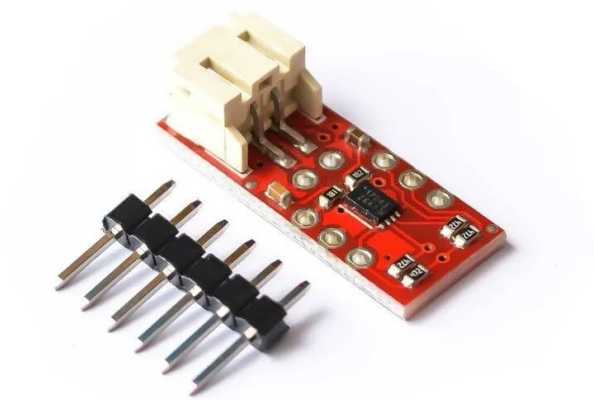

Fig. 6. Módulo Fuel Gauge Li-po (MAX 17043).

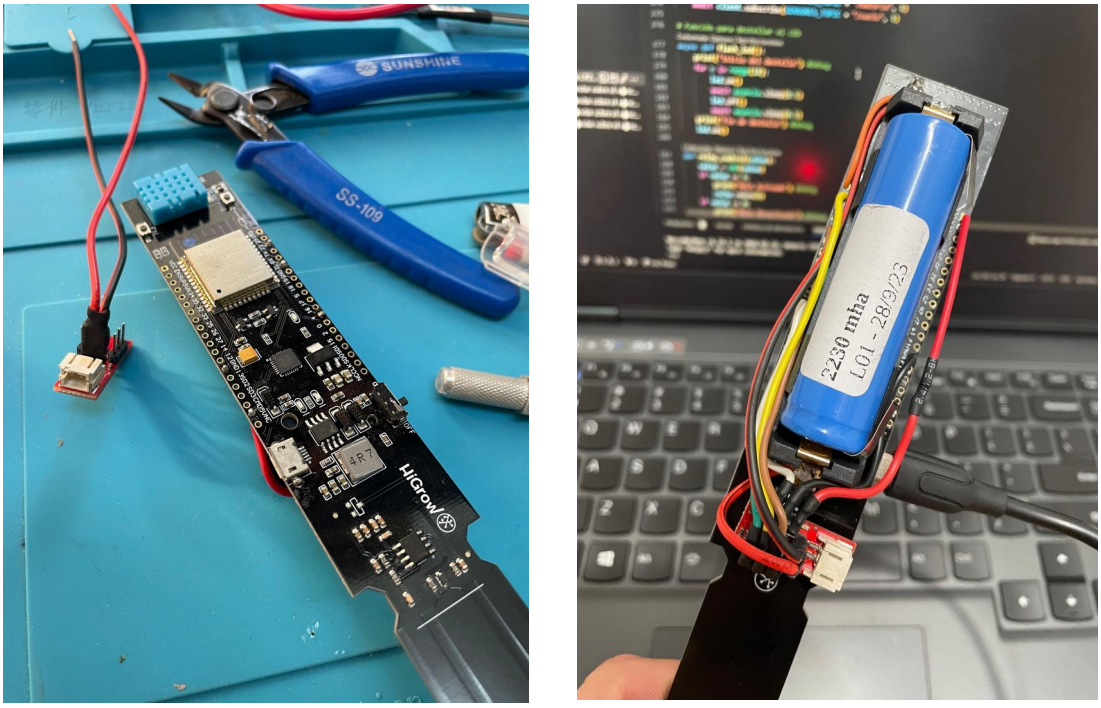

Fig. 7. Módulo sensor ESP32 con medidor de batería incorporado.

- Pin 22 (I2C): En la configuración inicial de la placa de desarrollo, el pin 22 del ESP32 se utiliza para conectar el sensor de humedad y temperatura DHT11. El problema es que este pin también se utiliza para la comunicación I2C necesaria para el módulo medidor de batería. Por lo tanto, se deberá modificar el pin de entrada de señal de datos del sensor para que funcione en otro pin que sea digital.

- Módulo DHT22: También se optó por reemplazar el módulo medidor de temperatura y humedad ambiente DHT11 por su versión mejorada y más precisión el DHT22.

Tras realizar diversas pruebas, se identificaron los principales desafíos asociados con el uso de esta placa de desarrollo y las estrategias más efectivas para abordarlos:

Escasa Información Documental: La falta de documentación oficial detallada complicó la identificación de los pines en uso y las funciones específicas de la placa de desarrollo. Para superar esta limitación, se llevó a cabo un meticuloso seguimiento de las pistas y una investigación exhaustiva para determinar la funcionalidad de cada componente.

Batería de Baja Duración: La placa utiliza una celda 18650 con un voltaje nominal de 3.7V y una capacidad de 2200mAh. Considerando que el microcontrolador ESP32 consume aproximadamente 200mA durante la transmisión de datos, las pruebas realizadas indicaron que la duración aproximada de la batería es de 24 horas. Este tiempo de operación es problemático, dado que uno de los objetivos del diseño es que el módulo funcione de manera remota, prescindiendo de una conexión eléctrica directa.

Para abordar este inconveniente, se proponen dos soluciones:

Modo DeepSleep: La activación del modo DeepSleep en el ESP32 puede mejorar significativamente la duración de la batería, logrando hasta 30 días de autonomía si el módulo se enciende y transmite datos una vez por hora. Esta modalidad minimiza el consumo de energía durante los períodos de inactividad, prolongando la vida útil de la batería.

Panel Solar: Se está evaluando el uso de un panel solar de 5V y 10W, como se muestra en la Fig. 8, para recargar la batería durante las horas de exposición a la luz solar. Esta solución permite adaptar el sistema para que se mantenga operativo y autosuficiente en términos de energía, reduciendo la dependencia de la batería y aumentando la sostenibilidad del módulo.

Fig. 8. Panel solar de 10w con salida 5v micro usb.

- Circuito Expuesto: Dado que se trata de una placa de desarrollo, no es posible utilizarla tal como viene de fábrica, ya que está en riesgo de sufrir daños por condiciones adversas como el sol y la lluvia.

Para mitigar este problema se propone el diseño de un gabinete a medida utilizando una impresora 3D, que luego será rellenado con resina epoxi u otro material impermeable para evitar la entrada de humedad que pueda dañar el circuito eléctrico.
Se dispone de un diseño básico que requiere pocas modificaciones, ilustrado a continuación en la Fig. 9.

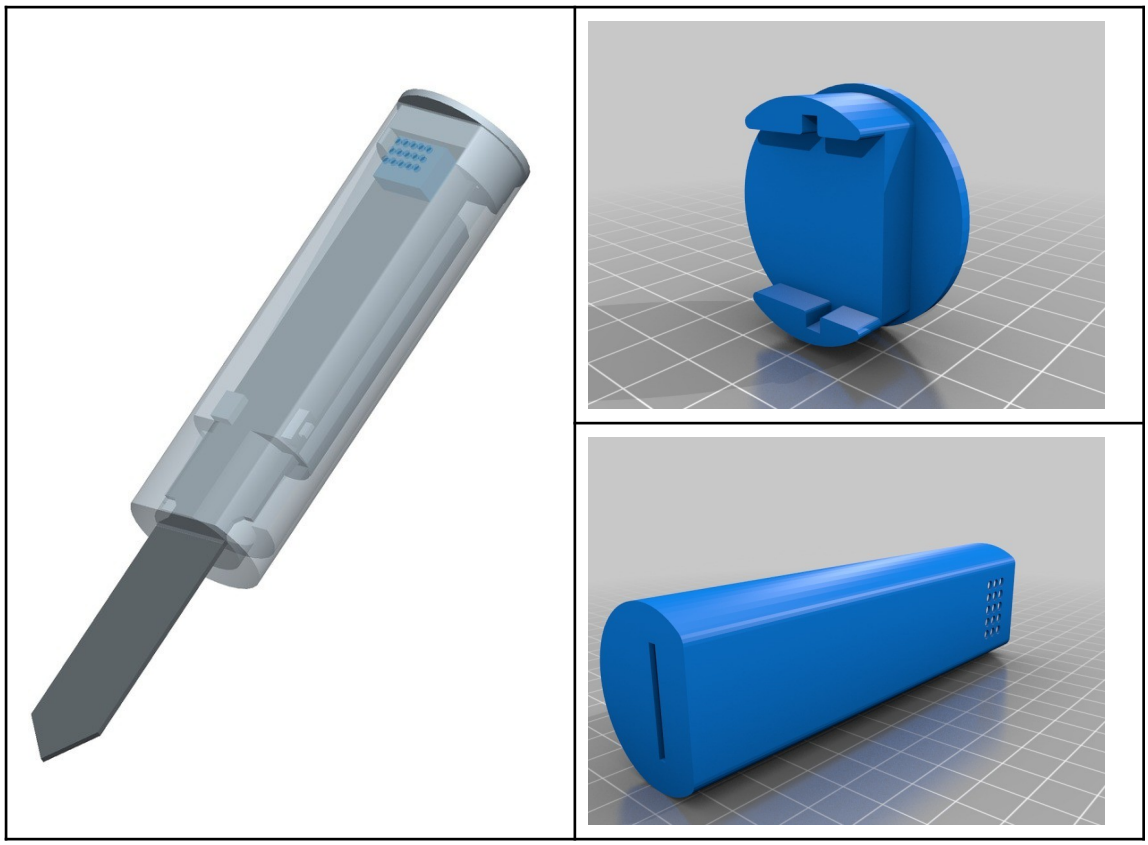

Fig 9. Modelo 3D de la carcasa para modulo ESP 32(DHT11.

4 Conclusiones

Este trabajo se enfoca en el desarrollo de un sistema de recolección de datos del suelo de bajo costo, basado en el microcontrolador Raspberry Pi 3, para aplicaciones en el Internet de las Cosas (IoT). El proyecto tiene como objetivo la recopilación en tiempo real de parámetros del suelo mediante nodos remotos autónomos, que transmiten los datos al servidor central utilizando el protocolo MQTT.

En el futuro, se prevé la incorporación de módulos actuadores, como electroválvulas para la automatización del riego y relés para la activación periódica de luces. Además, se planea el desarrollo de una aplicación web con una interfaz más intuitiva para facilitar la interacción con el sistema.

La programación del sistema se realizó utilizando los recursos proporcionados en la asignatura de Internet de las Cosas.

Agradecimientos

Este trabajo ha sido realizado con el apoyo de la Facultad de Ingeniería de la Universidad Nacional de Misiones (UNaM). El proyecto IIOT: Internet de las cosas aplicado a la industria 4.0, en el cual se enmarca este trabajo, fue financiado por la Universidad Nacional de Misiones en la convocatoria de Proyectos de Ciencia y Tecnología, en la modalidad de temas estratégicos.

Referencias

[1] Kodali, Ravi K.; Valdas, Aditya. Mqtt based monitoring system for urban farmers using esp32 and raspberry pi. En 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, 2018. p. 395-398.

[2] Pithadiya, Bhavesh, et al. An IoT Based Greenhouse Control System Employing Multiple Sensors, for Controlling Soil Moisture, Ambient Temperature and Humidity. En Proceedings of the 2nd International Conference on Electronics, Biomedical Engineering, and Health Informatics: ICEBEHI 2021, 3–4 November, Surabaya, Indonesia. Singapore: Springer Nature Singapore, 2022. p. 405-416..

[3] Chanchí G, Gabriel Elías; Ospina A, Manuel Alejandro; Campo M, Wilmar Yesid. IoT Architecture for Monitoring Variables of Interest in Indoor Plants. Computación y Sistemas, 2021, vol. 25, no 4, p. 695-705.

[4] Placa Raspberry Pi 3. [online] disponible: Chrome extension://efaidnbmnnnibpcajpcglclefindmkaj/https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf

[5] Sensor de temperatura y humedad ambiental y humedad del suelo (ESP32 DHT11). [online] disponible: https://www.taloselectronics.com/blogs/tutoriales/leer-temperatura-y-humedad-esp32s-y-dht11

[6] HiGrow plants monitoring sensor. [online] disponible: https://hackaday.io/project/25253-higrow-plants-monitoring-sensor