***Año del Fortalecimiento de la Soberanía Nacional”***

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**CURSO:**

Internet de las cosas

**TEMA:**

**Diseño e implementación de un sistema IoT mediante la plataforma ESP32 para la monitorización del proceso de compostaje de residuos orgánicos domésticos.**

**INTEGRANTES:**

\_ Tardeo Segovia Ronald Edison

\_ Ramos Cahuana Wily Hugo

\_ Cancán Damián Kevin Anderson

\_ Huamani Ramos Angel Manuel

\_ Serrano Bendezu Fernando Jose

**2023-0**

[**CAPÍTULO I: INTRODUCCIÓN**](#_shtj1ysxzm81)[**4**](#_shtj1ysxzm81)

[1.1 Revisión del estado del arte](#_xtjql2np0tpi) [4](#_xtjql2np0tpi)

[1.2 Planteamiento del problema](#_p49yt9wjgzst) [2](#_p49yt9wjgzst)

[1.3 Objetivos del proyecto](#_nqqs7c4djr8w) [3](#_nqqs7c4djr8w)

[**CAPÍTULO II: MARCO TEÓRICO**](#_tk8qq5jiq347)[**4**](#_tk8qq5jiq347)

[2.1 Compostera](#_5n28ta5xxeb5) [4](#_5n28ta5xxeb5)

[2.2 Sistema de aireación](#_w3h97i7tfua6) [4](#_w3h97i7tfua6)

[2.3 Sistema de goteo](#_1i1msdvi1k30) [5](#_1i1msdvi1k30)

[2.4 Internet de las cosas (IoT):](#_mrvxshr38jyo) [5](#_mrvxshr38jyo)

[2.5 Sensores:](#_gs09ms5kjghh) [5](#_gs09ms5kjghh)

[2.6 Protocolo MQTT:](#_p0q4pjonu9h) [6](#_p0q4pjonu9h)

[2.7 Node-RED:](#_bmmxe827v6jr) [6](#_bmmxe827v6jr)

[2.8 Base de datos](#_asxv0v87sfqd) [6](#_asxv0v87sfqd)

[2.9 Broker Mosquitto](#_6iq8yo4ags9s) [7](#_6iq8yo4ags9s)

[2.10 Arduino IDE:](#_adyqaqrchznn) [7](#_adyqaqrchznn)

[2.11 ESP32:](#_leq2amnsu3w3) [8](#_leq2amnsu3w3)

[**CAPÍTULO III: COMPONENTES DEL SISTEMA**](#_hsygkcbbgsr7)[**9**](#_hsygkcbbgsr7)

[3.1 ESP32](#_nrt16savraf6) [9](#_nrt16savraf6)

[3.2 Sensor de humedad y temperatura ambiente DHT11](#_ronqphg1ol8k) [10](#_ronqphg1ol8k)

[3.3 Sensor de temperatura del compost: Sonda DS18B20](#_3i2rr3kio5x9) [11](#_3i2rr3kio5x9)

[3.4 Sensor de humedad del compost: YL-69](#_v0uyucvn6geo) [12](#_v0uyucvn6geo)

[3.5 Sensor de metano MQ-4](#_sag1dontolpz) [13](#_sag1dontolpz)

[3.6 Sensor ultrasonidos HC-SR04:](#_i1tp831c2evi) [13](#_i1tp831c2evi)

[3.7 Protoboard](#_e4rxvvgsnw2r) [14](#_mdsdq5ftjfx1)

[3.8 Actuadores](#_e2vv7k2ea4hr) [16](#_e2vv7k2ea4hr)

[**CAPÍTULO IV: IMPLEMENTACIÓN DEL SISTEMA**](#_luyzexsjb8wf)[**24**](#_jbipd6vub8s6)

[**CAPÍTULO V: RESULTADOS**](#_h24ofjfih0k8)[**27**](#_h24ofjfih0k8)

[**CAPÍTULO VI: CONCLUSIONES**](#_a1zrvca2fdlt)[**29**](#_a1zrvca2fdlt)

[**7. Bibliografía**](#_h5vi7yhtcx32)[**30**](#_h5vi7yhtcx32)

[**8. Anex**](#_32snzcdxavv)**os** [**31**](#_32snzcdxavv)

**Índice de figuras**

Imagen 1: protocolo MQTT

Figura 2: Mosquitto. imagen propia

Imagen 3: Esp32.

Figura 4: Sensor DHT11

Figura 5: Sonda del sensor de temperatura DS18B20

Figura 6: Sonda YL-69 y tarjeta comparadora LM393

Figura 7: Sensor de metano MQ4

Figura 8: Módulo sensor de ultrasonidos HC-SR04

Imagen 9: Protoboard

Figura 10: Relé

Figura 11: Maceta

Figura 12: Bomba de aire para acuario

Figura 13: Gotero difusor

Figura 14: Led

Figura 15: Resistencias

Figura 16: Cables de protoboard

# **CAPÍTULO I: INTRODUCCIÓN**

## 1.1 Revisión del estado del arte

**"Automatización del proceso de compostaje de residuos orgánicos domésticos mediante la plataforma ESP32" de los autores Rivas et al. (2020)**

**OBJETIVO:** Desarrollar un sistema de automatización del proceso de compostaje de residuos orgánicos domésticos utilizando la plataforma ESP32.

**RESULTADOS:** El sistema permite la monitorización y control de variables críticas del proceso, como la temperatura y la humedad, y utiliza un algoritmo de control para optimizar el proceso de compostaje.

**CONCLUSIONES:** Los autores demuestran la viabilidad de la tecnología IoT para la automatización de procesos sostenibles y amigables con el medio ambiente.

**"Diseño de un sistema de control para la gestión de residuos orgánicos mediante IoT" de los autores García et al. (2019)**

**OBJETIVO:** Diseñar un sistema de control mediante IoT que sea capaz de controlar de forma remota mediante el uso de internet la gestión de residuos orgánicos

**FUNCIONAMIENTO:** El sistema utiliza la plataforma ESP32 para la adquisición de datos y la comunicación con una aplicación móvil. Una vez obtenidos los datos la aplicación envía de forma automática a un sistema de control para que se haga la toma de decisiones.

**CONCLUSIONES:** Los sistemas de control ofrecen una valiosa ventaja para la gestión de residuos orgánicos pudiendo extenderse a otros tipos de residuos. Se recomienda optar por usar este sistema ya que mejora la eficiencia en gestión de residuos y reducir el impacto ambiental

**"An IoT-Based Intelligent Composting System for Organic Waste Management"** **de los autores Parveen et al. (2021)**

**OBJETIVO:** Presenta un sistema inteligente de compostaje basado en IoT para la gestión de residuos orgánicos.

**FUNCIONAMIENTO:** El sistema utiliza la plataforma ESP32 para la adquisición de datos y la gestión de la información en tiempo real.

**CONCLUSIONES:** Los autores demuestran cómo se puede utilizar la tecnología IoT para mejorar la eficiencia y la sostenibilidad en la gestión de residuos orgánicos.

## 1.2 Planteamiento del problema

El problema a abordar en este proyecto es la falta de un sistema eficiente y sostenible para el tratamiento de residuos orgánicos domésticos. La acumulación de residuos orgánicos en los vertederos causa problemas ambientales, como la emisión de gases de efecto invernadero y la contaminación del suelo y del agua. Además, el proceso de compostaje de los residuos orgánicos puede ser tedioso y requiere de conocimientos específicos.

Para solucionar este problema, se propone el diseño e implementación de un sistema IoT (Internet de las cosas) utilizando la plataforma ESP32 para monitorizar el compostaje de residuos orgánicos domésticos. El sistema permitirá monitorear y controlar el proceso de compostaje de manera remota, lo que facilitará su gestión y aumentará su eficiencia. Además, se podrán obtener datos en tiempo real sobre la temperatura, humedad y nivel de oxígeno del compost, lo que permitirá ajustar las condiciones para obtener un compost de alta calidad. En resumen, el objetivo de este proyecto es desarrollar un sistema sostenible y eficiente para el tratamiento de residuos orgánicos domésticos, mediante el uso de tecnología IoT y la plataforma ESP32.

## 1.3 Objetivos del proyecto

# 

* Diseñar y desarrollar un sistema IoT mediante la plataforma ESP32 para monitorear el compostaje de residuos orgánicos domésticos.
* Implementar un sistema de monitoreo y control remoto del proceso de compostaje, que permita obtener datos en tiempo real sobre la temperatura, humedad y nivel de oxígeno del compost.
* Seleccionar e integrar los sensores necesarios para medir variables críticas en el proceso de compostaje, tales como la temperatura, la humedad y el nivel de oxígeno, y transmitir esta información a una base de datos.
* Promover la gestión sostenible de los residuos orgánicos domésticos, reduciendo su impacto ambiental y contribuyendo a la conservación del medio ambiente.

# 

# **CAPÍTULO II: MARCO TEÓRICO**

# 

Este capítulo contempla una descripción detallada del proyecto, en primer lugar se definirá sobre la compostera, el sistema de aireación, sistema de goteo así como sus beneficios en los residuos orgánicos. Así mismo se explicarán los requerimientos de la ……y por último la tecnología implementada para su desarrollo.

## 2.1 Compostera

En la actualidad existen muchas soluciones comerciales diseñadas para producir residuos orgánicos durante el proceso de compostaje, estas suelen ser de varios materiales tanto de plástico como de madera y pueden tener diversas formas y tamaños. Sin embargo, para mantener el bajo costo debemos optar por el más económico. Para ello el diseño de la compostera debe ser de fácil uso, en donde se utilizan componentes comerciales de fácil acceso.

Para cumplir estos requisitos de tamaño compacto y de bajo costo, se utilizará una maceta de polietileno de 10 Litros.

## 2.2 Sistema de aireación

Para el sistema de aireación se partió de un diseño complejo que implicaba todo un

entramado de tubos por los que circula el aire. La idea estaba basada en la primera

versión del Compost Professor, un proyecto de Darian Johnson que persigue el mismo

objetivo que este pero que se aproxima al problema desde una perspectiva distinta

Al estudiar dicha solución, se dudó de la eficacia que pudiera tener usando únicamente ventiladores y sin utilizar un compresor. Además, los conductos de aire se podían atascar fácilmente y la humedad podría condensarse en su interior. Buscando un compresor de pequeño tamaño, se llegó a la solución final.

Para el sistema de aireación se utilizara una bomba de aire para acuario, por su precio accesible y volumen adecuado.

## 2.3 Sistema de goteo

Para el sistema de goteo se requiere de un depósito de agua, una bomba pequeña y tuberías para riego por goteo. Se decide utilizar un sistema de riego por goteo por ser más eficiente en el uso del agua y requerir de menos presión que otras soluciones.

El depósito utilizado será el macetero de polipropileno de 5 L el cual se escoge por su bajo coste y por tener una capacidad suficiente como para dar autonomía en el riego durante varios días.

Se utilizará un único gotero regulable de hasta 60L/h con capacidad de difusión para humedecer toda la superficie de la pila de compost por igual. El caudal máximo del gotero determina la elección de la bomba, que deberá ser capaz de aportar como mínimo 60L/h. Teniendo esto en cuenta, se elige una bomba sin escobillas con caudal de 500L/h y presión de 2,5m. La bomba trabaja a 12V DC y se puede conectar a un enchufe de 22V de alterna utilizando una fuente de alimentación.

## 2.4 Internet de las cosas (IoT):

El IoT es una tecnología que permite la conexión de objetos y dispositivos a Internet, permitiendo la recopilación y procesamiento de datos en tiempo real. Según Gubbi, Buyya, Marusic y Palaniswami (2013), el IoT puede ser aplicado en diversas áreas, incluyendo la agricultura, la salud y la gestión ambiental, entre otras.

## 2.5 Sensores:

Los sensores son dispositivos que permiten la medición de variables físicas, como temperatura, humedad y nivel de oxígeno. Según Li, Li y Yu (2019), los sensores son esenciales en la implementación de sistemas IoT, ya que permiten la adquisición de datos en tiempo real y la monitorización de variables críticas.

## 2.6 Protocolo MQTT:

Es un protocolo de conectividad máquina a máquina (M2M) de internet de las cosas.

Fue diseñado como un transporte de mensajes tanto de publicación como de suscripción extremadamente ligero. Es útil para conexiones con ubicaciones donde se requiere una pequeña huella de código y el ancho de banda de la red es muy importante.



Imagen 1: protocolo MQTT

## 2.7 Node-RED:

Node-RED es un editor visual de flujos de datos, que permite la creación de aplicaciones IoT de manera sencilla y rápida. Según Guinard, Trifa y Wilde (2016), Node-RED es una herramienta útil para la creación de aplicaciones IoT basadas en la nube, gracias a su facilidad de uso y flexibilidad.

## 2.8 Base de datos

Es una recopilación de datos estructurados que se almacena en un sistema informático; su funcionamiento está controlado por un sistema de gestión de base de datos(DBMS) que requiere del uso de un software que sirva como interfaz entre la base de datos y programas a los usuarios finales.

## 2.9 Broker Mosquitto

MQTT es un protocolo controlado por eventos donde no hay transmisión de datos periódica o continua. Este protocolo de mensajería se utiliza principalmente para comunicaciones de máquina a máquina (M2M) o conexiones del tipo internet de las cosas. Un broker es el servidor con el que se comunican los clientes.

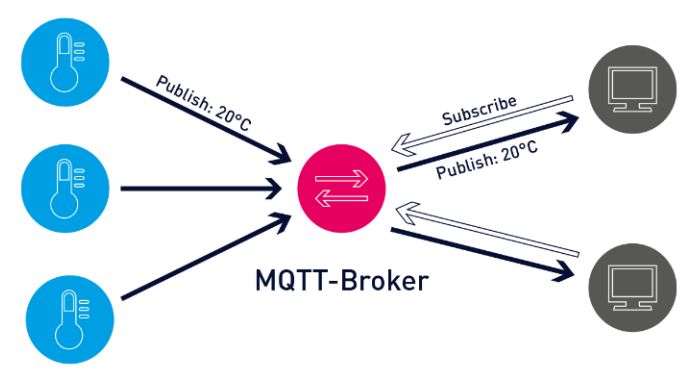


Figura 2: Mosquitto. imagen propia

## 2.10 Arduino IDE:

El Arduino IDE es un entorno de programación que permite la programación de microcontroladores basados en la plataforma Arduino. Según Banzi y Shiloh (2014), el Arduino IDE es una herramienta fácil de usar y flexible, que permite la creación de proyectos electrónicos desde cero.

## 2.11 ESP32:

El ESP32 es un microcontrolador de bajo costo y alto rendimiento que cuenta con conectividad Wi-Fi y Bluetooth integrada. Según González-Hernández, González-Hernández y González-Castro (2020), el ESP32 es una plataforma adecuada para la implementación de sistemas IoT, gracias a su capacidad de procesamiento y conectividad.

# 

# 

# **CAPÍTULO III: COMPONENTES DEL SISTEMA**

***Descripción del sistema.***

Se utiliza la plataforma ESP32 para conectar los sensores y dispositivos necesarios al sistema, tales como sensores de temperatura, humedad y nivel de oxígeno. Estos sensores permiten medir las condiciones dentro del compostador y proporcionar datos en tiempo real.

Luego, los datos recopilados por los sensores son enviados a una base de datos en la nube a través de la conexión Wi-Fi del ESP32 con el servidor MQTT. Una vez que los datos están almacenados en la base de datos, se pueden acceder para realizar análisis y visualizaciones.

Se desarrolla un dashboard en el node-red para visualizar los datos del sistema.

En conjunto, este sistema IoT permite a los usuarios monitorear el proceso de compostaje de residuos orgánicos domésticos de manera efectiva, lo que les permite reciclar de manera más efectiva y reducir su impacto ambiental. Además, la plataforma ESP32 es una opción económica y fácil de usar para desarrollar sistemas IoT, lo que hace que este proyecto sea accesible para una amplia gama de usuarios.

**Condiciones del sistema.**

Las condiciones y rangos de los sensores en la cual deben de estar se presentan para el monitoreo del proceso de compostaje:

**Sensor DHT11**: Este sensor es utilizado para medir la temperatura y la humedad relativa del aire en el entorno del compost. El rango de temperatura que puede medir es de 0°C a 50°C, con una precisión de ±2°C.

**Sensor de temperatura para compost**: Este sensor es utilizado para medir la temperatura del compost en sí mismo. El rango de temperatura que se debe medir está entre 15°C y 70°C para asegurar que el proceso de compostaje se esté llevando a cabo de manera efectiva.

**Sensor de humedad para compost:** Este sensor es utilizado para medir la humedad del compost. El rango de humedad adecuado para el proceso de compostaje está entre el 50% y el 60%.

**Sensor de ultrasonido:** Este sensor es utilizado para medir el nivel de llenado del contenedor de compost. El rango de medición puede variar, pero en general, debe ser capaz de medir una altura de entre 5 y 20 cm con una precisión de ±0,5 cm.

**Sensor de metano MQ-4:** Este sensor es utilizado para medir los niveles de gas metano en el aire del entorno del compost. El rango de medición debe ser bajo, generalmente menor al 1%, ya que los niveles altos de metano pueden ser peligrosos.

**Componentes:**

## 3.1 ESP32

El módulo ESP32 es un microcontrolador de bajo consumo y bajo costo. Este posee un microprocesador Tensilica Xtensa LX6 con una frecuencia hasta de 240 MHz. Además tiene conexión Wi-Fi y modo dual con Bluetooth. De acuerdo a Bruno (2019): “ESP32 está altamente integrado con switch de antena , balun para RF, amplificador de potencia, amplificador de recepción con bajo nivel de ruido, filtros y módulos de administración de energía, totalmente integrados dentro del mismo chip!!“

# 

Figura 3: Esp32.

**Características principales:**

● Procesador principal: Tensilica Xtensa LX6 de 32 bits.

● Wi-Fi: 802.11 b / g / n / e / i (802.11n @ 2.4 GHz hasta 150 Mbit / s).

● Bluetooth: v4.2 BR / EDR y Bluetooth Low Energy (BLE).

● Frecuencia de Clock: Programable, hasta 240 MHz.

● Rendimiento: hasta 600 DMIPS.

● ROM: 448KB, para arranque y funciones básicas.

● SRAM: 520KiB, para datos e instrucciones

# 

## 3.2 Sensor de humedad y temperatura ambiente DHT11

Es un sensor digital que capta la humedad y temperatura relativa. Posee una gran calidad y una respuesta rápida en las medidas, el rango para las medidas de la humedad es de 20% - 95% y para la temperatura de 0°C - 50°C.

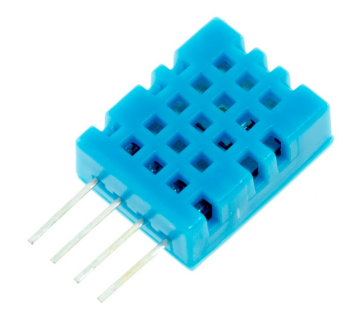


Figura 4: Sensor DHT11 . Imagen obtenida de google images

ESPECIFICACIONES TÉCNICAS

● Voltaje de Operación: 3V - 5V DC

● Rango de medición de temperatura: 0 a 50 °C

● Precisión de medición de temperatura: ±2.0 °C

● Resolución Temperatura: 0.1°C

● Rango de medición de humedad: 20% a 90% RH.

● Precisión de medición de humedad: 5% RH.

● Resolución Humedad: 1% RH

● Tiempo de sensado: 1 seg.

● Interface digital: Single-bus (bidireccional)

● Modelo: DHT11

● Dimensiones: 16\*12\*5 mm

● Peso: 1 gr.

● Carcasa de plástico celeste

## **3.3 Sensor de temperatura del compost: Sonda DS18B20**

Es un sensor digital que capta la temperatura. Es ideal para medir la temperatura dentro del compost debido a que cuenta con un encapsulamiento que lo hace resistente al agua, humedad y suciedad.



Figura 5: Sonda del sensor de temperatura DS18B20. Imagen obtenida de google images

## 3.4 Sensor de humedad del compost: YL-69

Es un sensor que tiene la capacidad de medir la humedad del compost. Aplicando un poco de tensión entre los terminales del sensor hace pasar una corriente y dependiendo de la resistencia que genere el compost es que nos arrojará el porcentaje de humedad. Nos permite hacer medidas digitales y analógicas. Se obtendrá medida del porcentaje de humedad 0% - 100%.

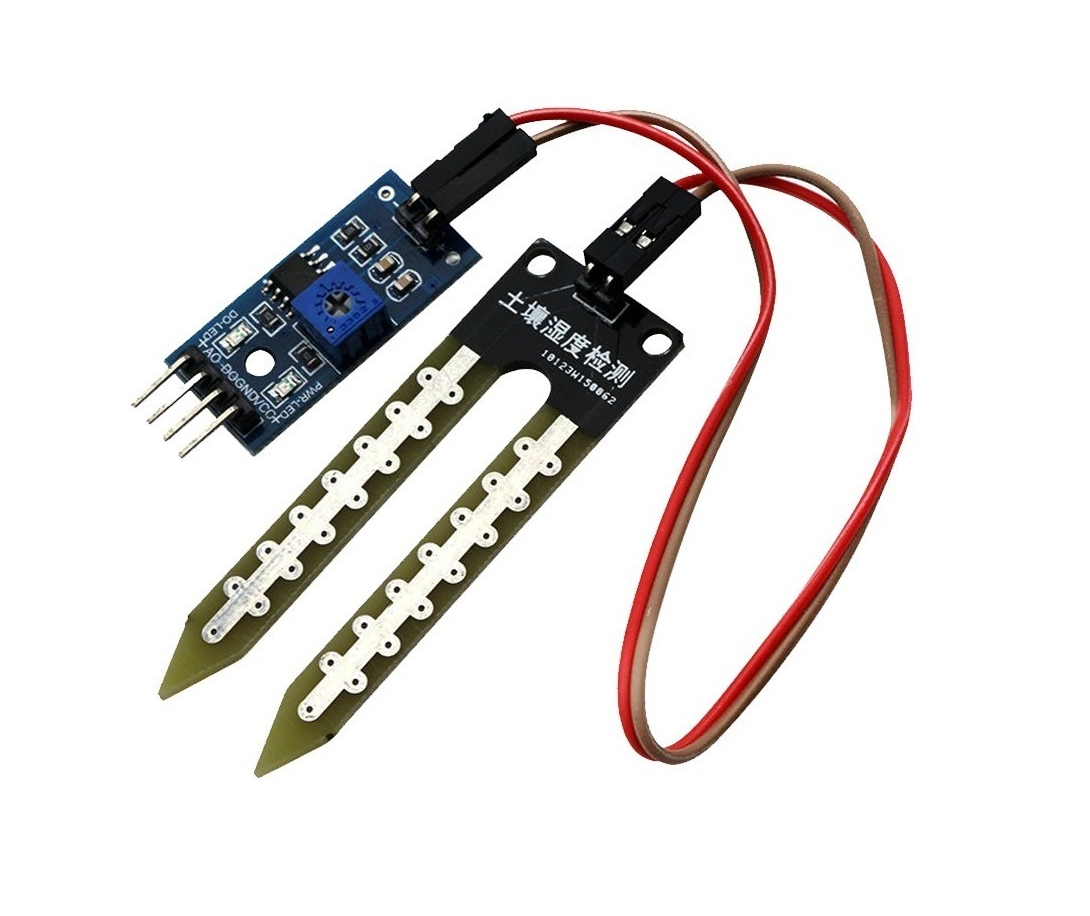


Figura 6: Sonda YL-69 y tarjeta comparadora LM393. Imagen obtenida de google images

## 3.5 Sensor de metano MQ-4

este sensor puede detectar las concentraciones de metano que están entre los 300 ppm - 10000 ppm, posee una respuesta rápida y alta sensibilidad. Funciona mediante señales eléctricas cuando se detecta el gas.

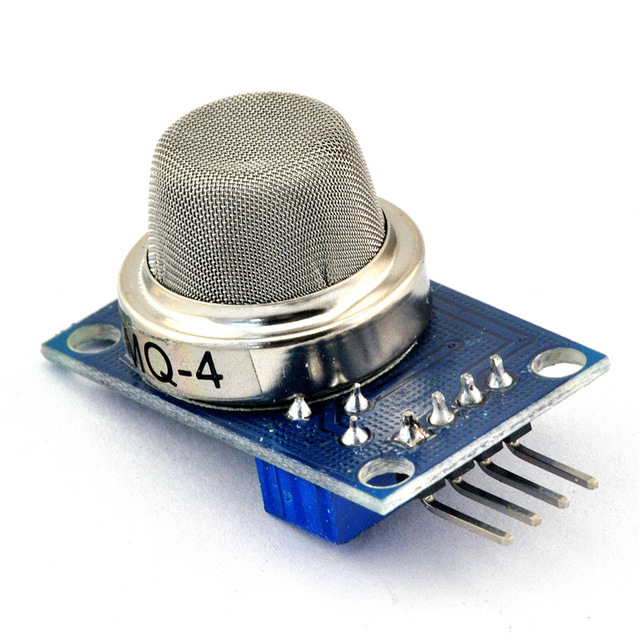


Figura 7: Sensor de metano MQ4. Imagen obtenida de google images

## **3.6 Sensor ultrasonidos HC-SR04:** Nivel del depósito se utilizará este sensor para comprobar si la compostera tiene la suficiente cantidad de agua para regar, se necesita conocer el nivel de agua en el recipiente. Este sensor posee un rango de medición de 2 cm a 400 cm.



Figura 8: Módulo sensor de ultrasonidos HC-SR04. Imagen obtenida de google images

## 3.7 Protoboard

## Es una placa de pruebas llena de orificios en donde se pueden insertar elementos electrónicos y cables para el armado de circuitos sin la necesidad de soldar ninguno de los componentes.



Imagen 9: Protoboard. Imagen obtenida de google images

Características :

● Puntos: 830 puntos.

● Color: Blanco.

● Material: Plástico ABS.

● Compatible con cualquier componente o cable de 20-29 AWG (0.3-0.8mm)

● Longitud: 16.5 cm.

● Ancho: 5.5 cm.

● Altura: 1 cm.

* **Módulo relé de 2 canales con optoacoplador.** Se utilizará para controlar el encendido y apagado de los actuadores. Posee dos canales que se activan a través de señales de 5V.



Figura 10: Relé

* Maceta de polietileno utilizada como composter



Figura 11: Maceta

## 3.9 Actuadores:

* Bomba de aire para el acuario Ireenuo Q7.
* 

Figura 12: Bomba de aire para acuario

* Gotero difusor



Figura 13: Gotero difusor

**Otros materiales:**

* LED



Figura 14: Led

* Resistencias de 220 ohmios



Figura 15: Resistencias

* Cables para protoboard

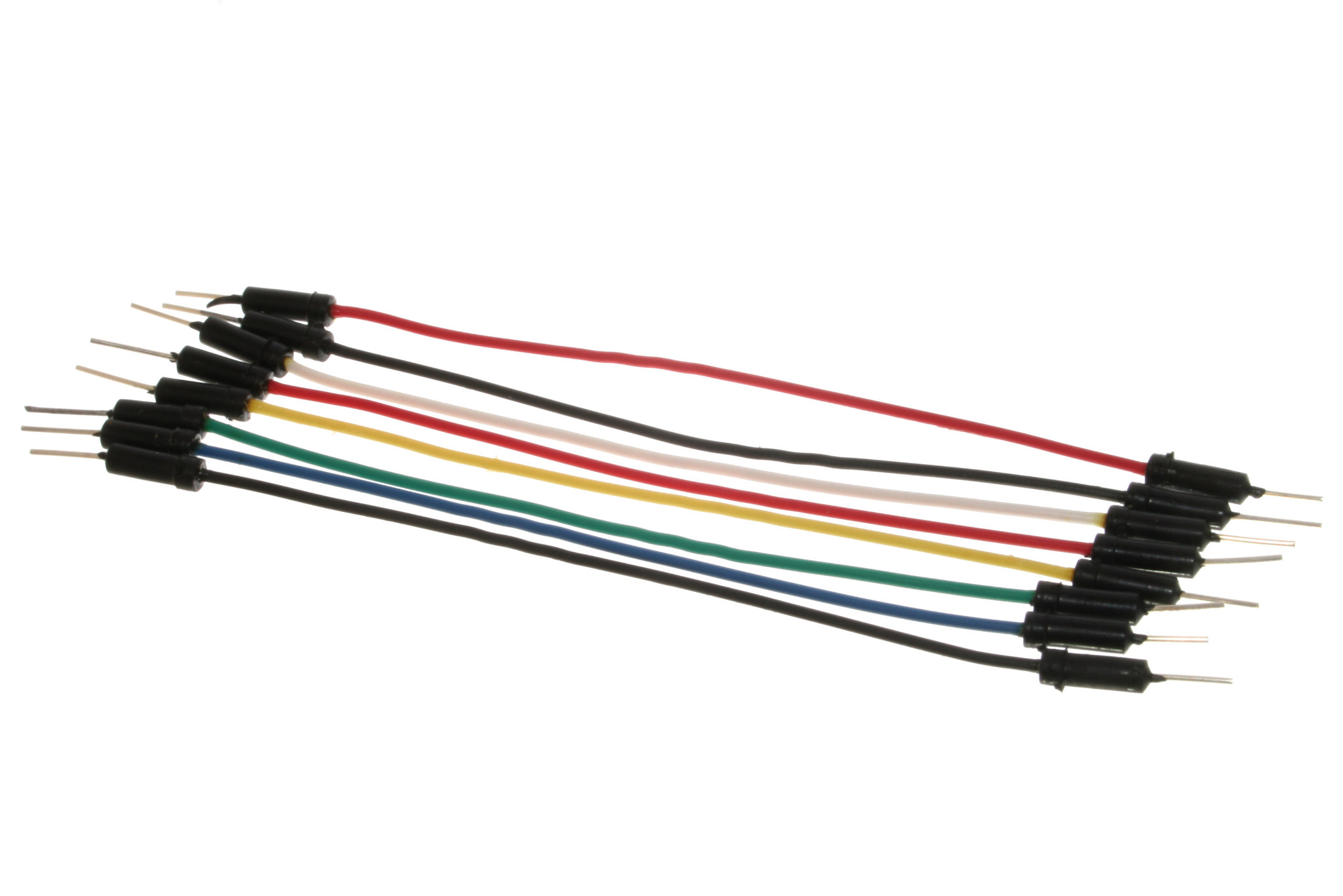
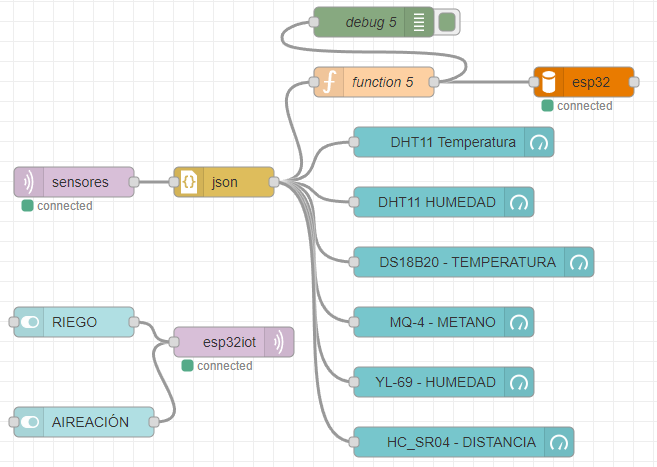


Figura 16: Cables de protoboard

# **CAPÍTULO IV: IMPLEMENTACIÓN DEL SISTEMA**

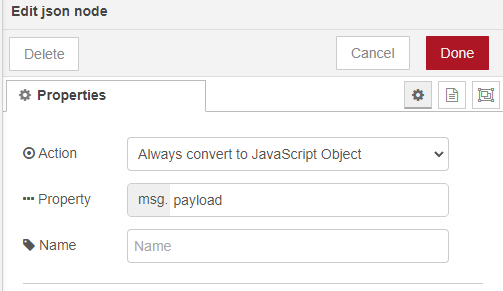
# **NODE RED:**

Flujo:



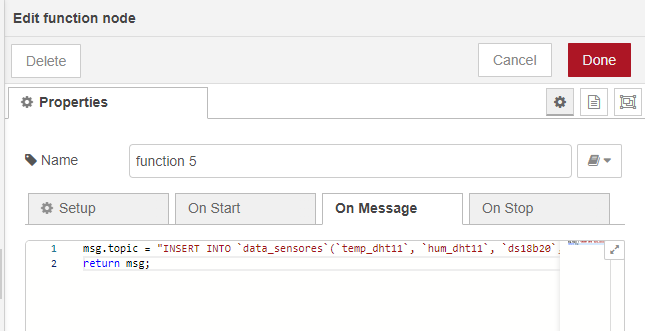
**JSON**





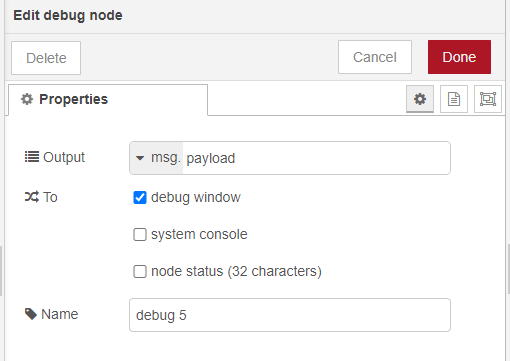
Se editan las propiedades del nodo json.

**FUNCTION**



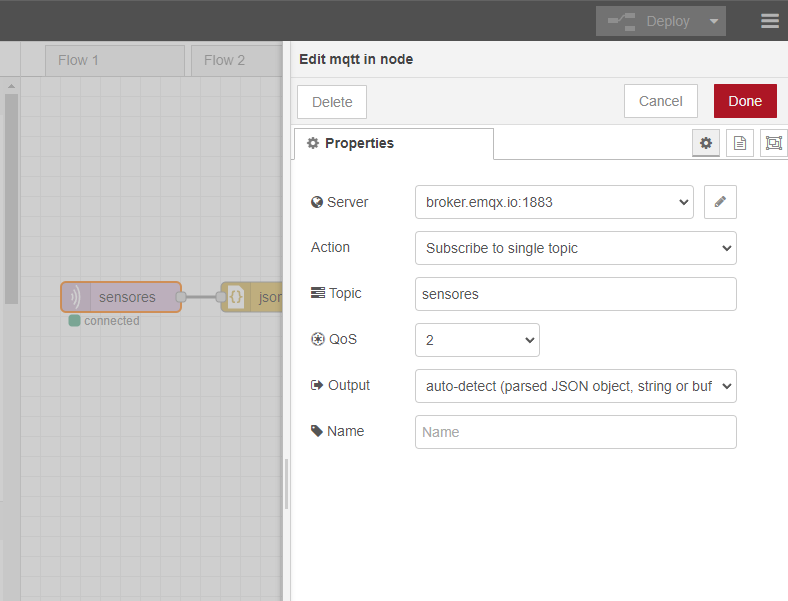
Se editan las propiedades del nodo de función.

**DEBUG**



Se editan las propiedades del nodo de depuración.

**MQTT IN**

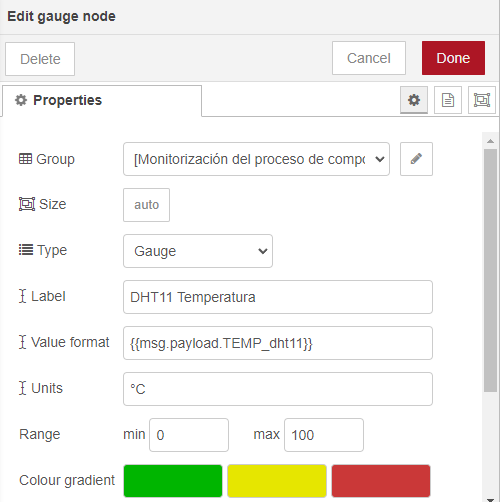


# 

# **configuración de la db**



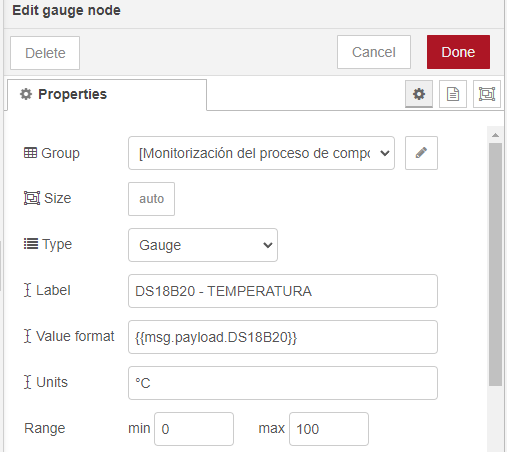
# **Nodo DHT11 Temperatura**



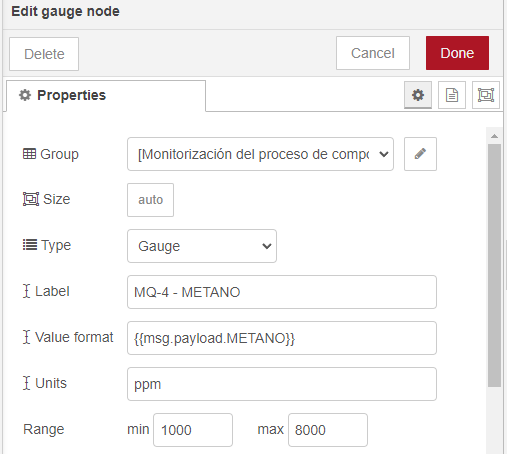
# **Nodo DHT11 Humedad**

# 

**Nodo DS18B20 - Temperatura del compost**

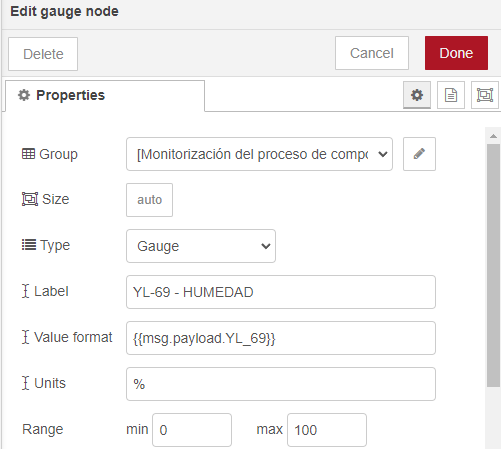


# **Nodo MQ-4 - Metano**

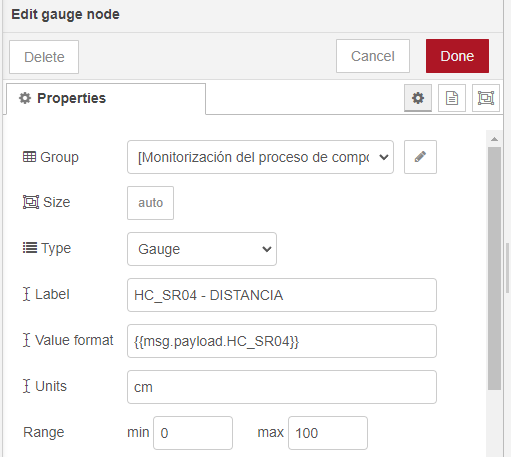


# 

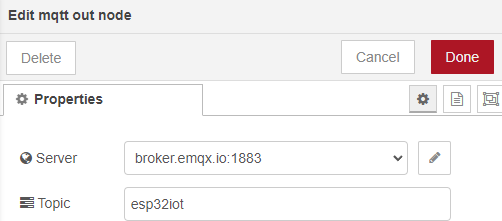
# **Nodo YL-69 - Humedad**



**Nodo HC\_SR04 - Distancia**

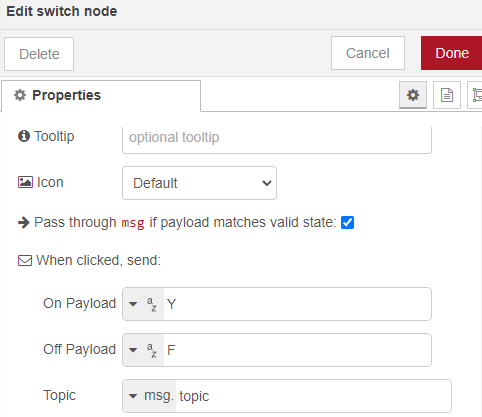


**Nodo MQTT IN**

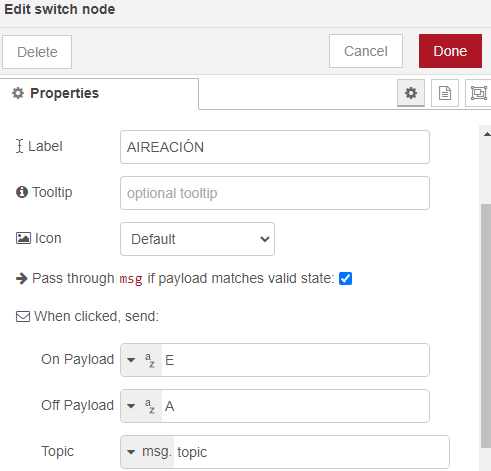


**Nodo Riego**





**Nodo Aireación**



# 

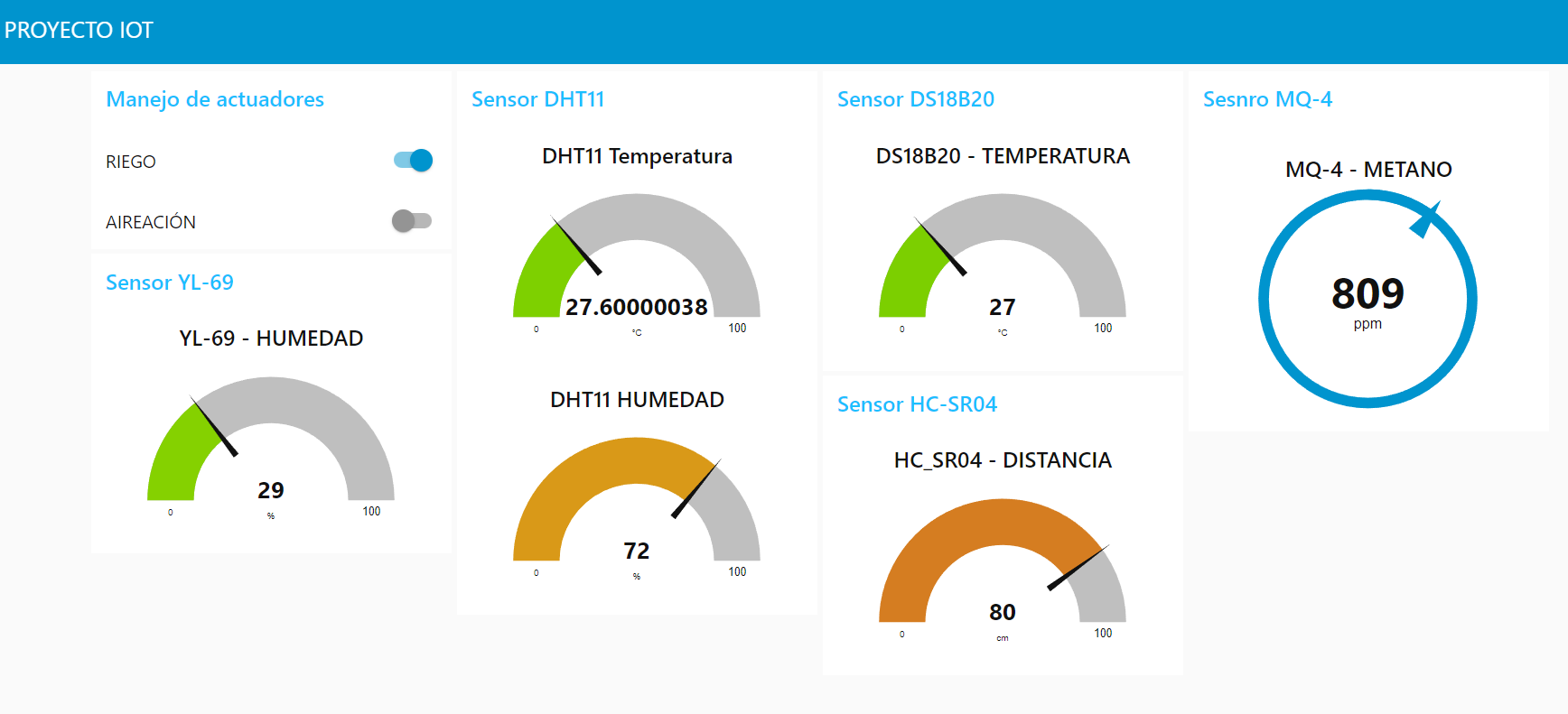
# **CAPÍTULO V: RESULTADOS**

Se obtuvieron los siguientes resultados:

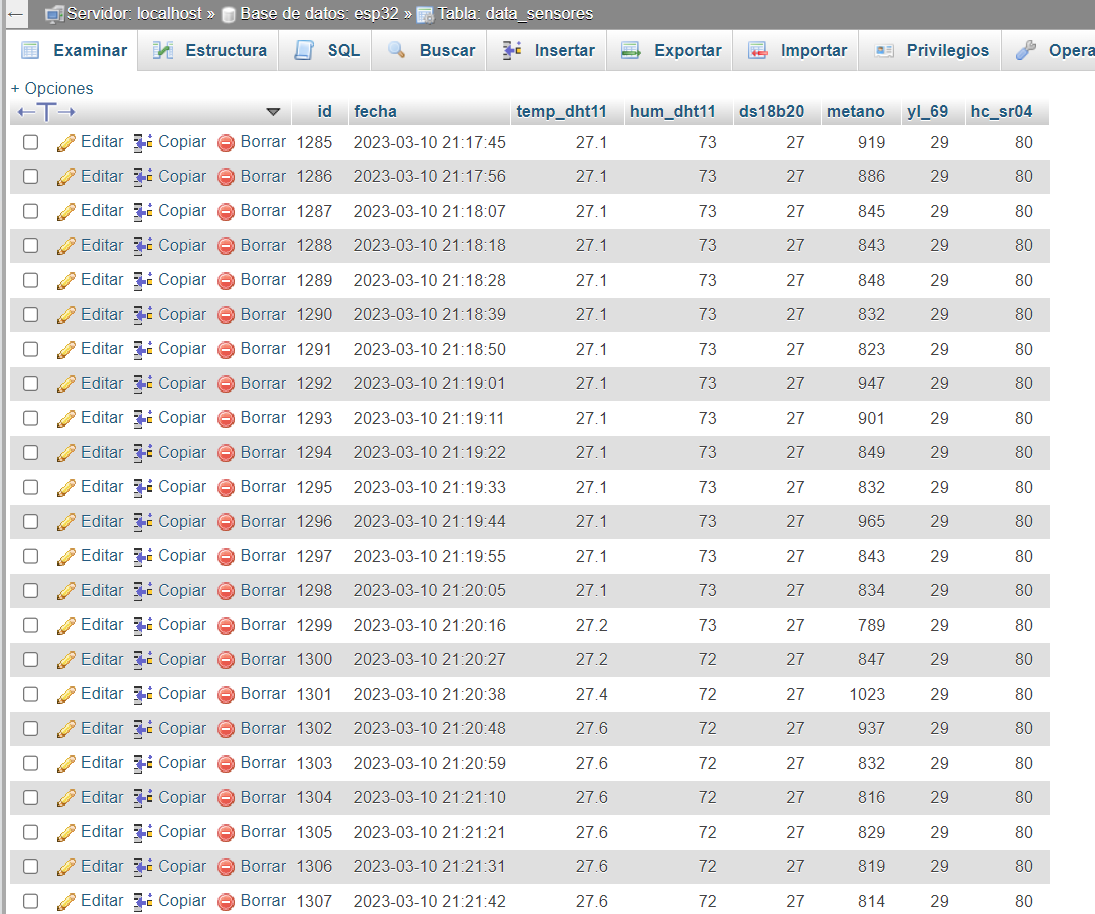
Con la conexión a un servidor con el protocolo MQTT se obtuvo los siguientes

resultados:

**DASHBOARD DEL NODE RED**



**BASE DE DATOS PhpMyAdmin**



Se creó una tabla: data\_sensores donde se almacena toda la información de los sensores.

# 

# **CAPÍTULO VI: CONCLUSIONES**

La aplicación elegida implica crear un sistema funcional de monitoreo y económico

para gestionar el proceso de compostaje a nivel doméstico. También construir un

prototipo replicable por cualquiera interesado en producir compost en su casa.

Tras el desarrollo de este trabajo final se concluye:

* Se ha desarrollado un prototipo funcional para la monitorización del proceso de compostaje de residuos orgánicos domésticos. El sistema riega o airea el compost según las necesidades del proceso de forma automática.
* El sistema almacena los datos que determinan el proceso de compostaje en una base de datos, desde donde el usuario puede descargarlos para poder analizarlos
* Se ha conseguido implementar la visualización de los datos del compost de residuos orgánicos de forma inmediata desde cualquier navegador web de un equipo conectado a la red local. De esta forma el usuario puede saber el estado del proceso en directo y de manera fácil y sencilla.
* Los principales componentes utilizados son de hardware libre. También son de software libre los programas utilizados. El proyecto queda por tanto abierto a mejoras, cambios y es libre de ser reproducido por cualquier persona interesada.
* Este proyecto de IoT ha permitido aprender sobre las posibilidades que ofrece el internet de las cosas y las arquitecturas más habituales de estos sistemas. También ha supuesto el desarrollo de nuevas destrezas relacionadas con la programación e implementación de sistemas IoT, profundizando también en los conocimientos de electrónica adquiridos en el desarrollo del curso.

# 

# **7. Bibliografía**

Rivas, A., Valdivia, G., Ludeña, A., & Pari, R. (2020). Automatización del proceso de compostaje de residuos orgánicos domésticos mediante la plataforma ESP32. Revista de Investigación Académica, 19, 1-10.

García, D., Vásquez, D., & León, E. (2019). Diseño de un sistema de control para la gestión de residuos orgánicos mediante IoT. Revista Científica de Ingeniería y Tecnología, 3(2), 16-24.

Parveen, N., Sharma, N., & Mittal, A. (2021). An IoT-Based Intelligent Composting System for Organic Waste Management. International Journal of Innovative Technology and Exploring Engineering, 10(2), 456-461.

Gubbi, J., Buyya, R., Marusic, S. y Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660. doi: <https://doi.org/10.1016/j.future.2013.01.010>

Guinard, D., Trifa, V. y Wilde, E. (2016). A resource oriented architecture for the Web of Things. IEEE Internet Computing, 20(2), 64-69. doi: 10.1109/MIC.2016.30.

Li, Y., Li, L. y Yu, H. (2019). An IoT-based environmental monitoring system for green buildings. Energy and Buildings, 201, 109389. doi: 10.1016/j.enbuild.2019.109389

Banzi, M. y Shiloh, M. (2014). Getting Started with Arduino (3rd ed.). Maker Media, Inc.

González-Hernández, R., González-Hernández, J., y González-Castro, V. (2020). Implementation of an IoT platform based.

# 

# **8. Anexos**

**8.1 Anexo 1**

**CÓDIGO IMPLEMENTADO PARA CONECTAR A SERVIDOR**

/\*-------Librerias-------------------------\*/

#include "DHTesp.h" //Librería para el sensor de humedad

#include <WiFi.h>

#include <OneWire.h> ///Librería DS18B20 Dallas Temperatura

#include <DallasTemperature.h> ///Librería DS18B20 Dallas Temperatura

#include <ArduinoJson.h> // Libreria para convertir los datos en formato JSON

#include <PubSubClient.h> // Libreria para publicar y suscribir al MQTT

/\*-------Sensor DHT11 Sensor de humedad-------------------------\*/

int pinDHT = 15; //Declaramos el variable que almacena el pin a conectar el DHT11 pin 15

DHTesp dht;//Instanciamos el DHT

/\*-------Sensor MQ4 Sensor de metano------------------------------\*/

int LED1 = 32;//Pin LED definido conectado en D32

int Sensor\_input = 34;//Pin digital 34 para la entrada de sensor en A0

/\*-------Sensor DS18B20 sensor de Temperatura compost-------------\*/

#define DS18B20 14 // Se conecta al pin 14 del esp32

OneWire oneWire(DS18B20); //Instancia el Objeto oneWire en el pin del Sensor para iniciar las lecturas

//Reenvia referencias de oneWire al sensor de Dallas(DS18B20)

DallasTemperature Sensor(&oneWire);

float lectura\_temCompost; // Variable para almacenar datos

/\*-------Sensor YL-69 sensor de Humedad compost-------\*/

const int humsuelo = 33; //Conectado con el PIN A0 del sensor

int valHumsuelo; //Almacena el valor de la humedad en la variable

/\*-------Sensor HC-SR04 sensor de Ultrasonido-------\*/

int DISTANCIA =0;

int pinLed=26; // Declaramos Led en el pin 26

int pinEco=12; // Declaramos Led en el pin 12

int pinGatillo=13; // Declaramos Led en el pin 13

/\*-------Variables para el loop-------\*/

uint32\_t timerUpdate = 0; //Para calcular la diferencia en el loop

const int DELAY\_UPDATE = 1000; //Delay de 1 seg

/\*-------Información de WiFi---------------\*/

const char\* ssid = "WIFI";

const char\* password = "CONTRASEÑA";

/\*-------Información del broker MQTT---------------\*/

const char\* mqtt\_server = "broker.emqx.io";

/\*-------Declaramos las variables para enviar los datos de los sensores.---------------\*/

char data\_temp[12]="";

char data\_humi[12]="";

char data\_humiCompost[12]="";

char data\_meta[12]="";

char data\_ds18b20[12]="";

char data\_hcsr04[12]="";

/\*----- Inicializacion del esp32 para el wifi mqtt---\*/

WiFiClient esp32;

PubSubClient client(esp32);

/\*-----Se declara la el long para calcular la distancia---\*/

long leerDistanciaUltrasonido(int pinGatillo, int pinEco){

pinMode(pinGatillo, OUTPUT); //Iniciamos el pin emisor de ruido en salida

digitalWrite(pinGatillo, LOW); //Apagamos el emisor de sonido

delayMicroseconds(2);//Retrasamos la emision de sonido por 2 milesimas de segundo

digitalWrite(pinGatillo, HIGH);//Comenzamos a emitir sonido

delayMicroseconds(10);//Retrasamos la emision de sonido por 10 milesimas de segundo

digitalWrite(pinGatillo, LOW);//Apagamos el emisor de sonido

pinMode(pinEco, HIGH);//Comenzamos a escuchar el sonido

return pulseIn(pinEco, HIGH);//Calculamos el tiempo que tardo en regresar el sonido

}

// Conexión a nuestro WIFI

void config\_wifi(){

delay(10);

WiFi.mode(WIFI\_STA);

Serial.println("");

Serial.print("conectando ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while(WiFi.status() !=WL\_CONNECTED){

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

/\* ---Conexion al MQTT para reconectar-----\*/

void reconnect() {

// Bucle hasta que estemos reconectados

while (!client.connected()) {

Serial.print("Intentando la conexión MQTT...");

// Crear una identificación de cliente aleatoriaID

String clientId = "ESP32";

// intento de conexión

if (client.connect(clientId.c\_str())) {

Serial.println("conectado");

}

else {

Serial.print("fallido, rc=");

Serial.print(client.state());

Serial.println("inténtalo de nuevo en 3 segundos");

// Espere 5 segundos antes de volver a intentarlo

delay(3000);

}

}

}

/\*------------Sensor DHT11 Sensor de humedad--------------------------\*/

void setup() {

// monitor serial

Serial.begin(115200);

config\_wifi(); // llamada a la funcion para conectar al wifi

/\*Sensor DHT11 Sensor de humedad--------------------------\*/

dhtSensor.setup(DHT\_PIN, DHTesp::DHT22);

/\*Sensor MQ4 Sensor de metano------------------------------\*/

pinMode(pinLedRojo, OUTPUT);

/\*Sensor DS18B20 sensor de Temperatura compost-------------\*/

DS18B20.begin();

/\*Sensor YL-69 sensor de Humedad compost-------------\*/

pinMode(humsuelo, INPUT);

/\*Sensor HC-SR04 sensor de Ultrasonido---------------\*/

pinMode(pinLed, OUTPUT);

/\* Inicializacion a la conexion del servidor mqtt---------------\*/\*/

client.setServer(mqtt\_server,1883);

}

void loop() {

/\* Verificar la conexion al wifi \*/

if (!client.connected()) {

reconnect();

}

client.loop();

/\* Condicional para enviar periodicamente al nodered\*/

if (millis() - timerUpdate >= DELAY\_UPDATE){

timerUpdate = millis();

Serial.println("");

Serial.println("\t\*\*\*\*DATOS DEL COMPOST\*\*\*\*\*\*");

Serial.println("------------");

/\*Sensor DHT11 Sensor de humedad------------------------\*/

TempAndHumidity data = dhtSensor.getTempAndHumidity();

Serial.println("Temperatura: " + String(data.temperature,2) + "°C");

Serial.println("Humedad: " + String(data.humidity,1) + "%");

Serial.println("------------");

/\*---------------Sensor MQ4 Sensor de metano------------------------------\*/

int lectura\_mq4 = analogRead(Sensor\_input);

Serial.print("Sensor de gas metano: ");

Serial.print(lectura\_mq4); /\*Imprime el valor del sensor\*/

Serial.print("\t");

Serial.println("------------");

/\*-----Sensor DS18B20 sensor de Temperatura compost-------------\*/

DS18B20.requestTemperatures(); //Se envía el comando para leer la temperatura

float t\_ds18b20= DS18B20.getTempCByIndex(0);

Serial.print("Temperatura ds18b20: ");

Serial.println(t\_ds18b20);

Serial.println("------------");

/\*Sensor YL-60 sensor de Humedad compost-------------\*/

valHumsuelo = map(analogRead(humsuelo), 4092, 0, 0, 100);

Serial.print("Humedad del suelo: ");

Serial.print(valHumsuelo);

Serial.println(" %");

Serial.println("------------");

/\*Sensor HC-SR04 sensor de Ultrasonido---------------\*/

//calculamos la disctancia en cm

DISTANCIA = 0.01723 \* leerDistanciaUltrasonido(pinGatillo, pinEco);

//Mostramos la distancia

Serial.print("Distancia: ");

Serial.println(DISTANCIA);

//Si la distancia es menor a 20cm encendemos la led

if(DISTANCIA < 20){

digitalWrite(pinLed, HIGH);

}

//Si la distancia es mayor a 20 apagamos el led

else{

digitalWrite(pinLed, LOW);

}

Serial.println("------------");

/\*---------- ALARMAS CON CONTROL DE LEDS-------\*/

/\* Encender el LED si la temperatura es menor a 30 grados\*/

if (temperaturaDHT > 30) {

digitalWrite(LED1, HIGH);

} else if{

digitalWrite(LED1, LOW);

}

/\* Encender el LED si la humedad es menor 50 \*/

if (valHumsuelo < 50) {

digitalWrite(LED2, HIGH);

} else {

digitalWrite(LED2, LOW);

}

/\* Si el valor del metano es mayor a 1800, activar la alarma \*/

if (lectura\_mq4 > 1800) { /\*Si la lectura del sensor pasa el umbral 1800\*/

Serial.println("Hay presencia");

digitalWrite (pinLedRojo, HIGH) ; /\*Enciende el LED en el sistema\*/

}

else {

Serial.println("No hay presencia");/\*Si la lectura del sensor no pasa el umbral 1800\*/

digitalWrite (pinLedRojo, LOW) ; /\*Apaga el LED en el sistema\*/

}

/\* Encender el LED si la distancia es menor a 5 cm \*/

if (valHumsuelo < 5) {

digitalWrite(LED3, HIGH);

} else {

digitalWrite(LED3, LOW);

}

/\* Encender el LED si la temperatura del ds18b20 es menor a 29 \*/

if (["DS18B20"] > 30) {

digitalWrite(LED4, HIGH);

} else if{

digitalWrite(LED4, LOW);

}

/\* CREACION DEL JSON CON LOS DATOS DE LOS SENSORES \*/

StaticJsonDocument<128> doc;

doc["TEMP\_dht11"] = data.temperature;

doc["HUM\_dht11"] = data.humidity;

doc["DS18B20"] = t\_ds18b20;

doc["METANO"] = lectura\_mq4;

doc["YL\_69"] = valHumsuelo;

doc["HC\_SR04"] = DISTANCIA;

String output; //Variable de salida

//Serializacion de los datos del JSON

serializeJson(doc, output);

Serial.print("Publish message: ");

Serial.println(output);

//Se envian los datos serializados al servidor mosquito en el topico Sensores

client.publish("sensores",output.c\_str());

}

delay(10000);

}

**link del video del código fuente:**

[**https://drive.google.com/file/d/1\_1A\_sj51hnrber\_bAzwyduM8c6p2jnA4/view?usp=share\_link**](https://drive.google.com/file/d/1_1A_sj51hnrber_bAzwyduM8c6p2jnA4/view?usp=share_link)

**8.2 Anexo 2: Diagrama de circuito usando 5 sensores**

