

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú, Decana de América

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA ACADÉMICA DE INGENIERÍA DE SISTEMAS



INTERNET DE LAS COSAS

PROYECTO FINAL

SISTEMA DE RIEGO AUTOMÁTICO

GRUPO 03

INTEGRANTES

Arce Carhuamaca, Elias Ruben	19200159
Cristobal Artica, Luis Enrique	19200011
Estrella Condor, Joshep Douglas	19200146
Gonza Santibañez, Pamela Sofia	19200017
Guzmán Nolasco, Enrique Eduardo	19200022
Herrera Yzaguirre, Leonardo Deivy	19200225

PROFESORA

Rosas Cueva, Yessica

Lima, Perú

2022

PARTICIPACIÓN DE INTEGRANTES				
Nº	Integrante		Código de estudiante	Porcentaje de participación
	Apellidos	Nombres		
1	Arce Carhuamaca	Elias Ruben	19200159	87%
2	Cristobal Artica	Luis Enrique	19200011	89%
3	Estrella Condor	Joshep Douglas	19200146	90%
4	Gonza Santibañez	Pamela Sofia	19200017	87%
5	Guzmán Nolasco	Enrique Eduardo	19200022	90%
6	Herrera Yzaguirre	Leonardo Deivy	19200225	89%

Índice

1.	Introducción.....	5
1.1.	Revisión del estado del arte.....	5
1.2.	Planteamiento del problema.....	7
1.3.	Objetivos	8
1.4.	Alcances	9
2.	Marco teórico.....	10
2.1.	Internet de las Cosas (IoT)	10
2.2.	Automatización	10
2.3.	Sistema de Riego Automático	10
2.4.	Arduino IDE.....	10
2.5.	Wi-Fi	10
2.6.	MQTT.....	11
2.7.	Node-Red	11
2.8.	MySQL.....	11
2.9.	Librerías de Arduino	11
3.	Componentes del sistema de riego	12
3.1.	Sensor ultrasonido HC-SR04	12
3.2.	Sensor DHT22.....	12
3.3.	Sensor de humedad del suelo FC-28.....	13
3.4.	Relé.....	13
3.5.	Display LCD I2C	14
3.6.	LED	14
3.7.	LED RGB	15
3.8.	Resistencias Eléctricas	15
3.9.	Microcontrolador ESP32.....	16
3.10.	Protoboard	16

3.11.	Bomba de agua	17
3.12.	Adaptador de corriente	17
3.13.	Descripción del sistema de riego	17
4.	Implementación del sistema	18
4.1.	Diseño metodológico del sistema.....	18
4.2.	Lugar de trabajo	25
4.3.	Implementación de Node-RED	27
4.4.	Ejecución de Node-RED	32
4.5.	Visualización a través de gráficos	33
4.6.	Implementación de base de datos en Node-RED	34
4.7.	Flujo completo del Node-RED.....	42
5.	Resultados.....	42
5.1.	Impresión de datos en Arduino IDE.....	43
5.2.	Escenarios de uso del sistema de riego	45
5.3.	Gráficos en Node-RED	49
6.	Conclusiones.....	54
7.	Bibliografía.....	55
	Anexo N° 1	57
	Anexo N° 2	67

1. Introducción

El constante crecimiento de la población mundial ha traído consigo uno de los problemas más serios para el ser humano, como un mayor consumo energético, además del uso de bastantes recursos hídricos teniendo como efecto negativo su desuso en otras áreas como el consumo. La constante mejora de las soluciones tecnológicas relacionadas con IOT (Internet of Things), el sector agrario, en muchos casos y tipos de explotación, ha quedado desbancado en cuanto a la modernización del resto de sectores, como la industria, lo que conlleva una peor optimización sobre los recursos de los que se dispone. Dentro de la variedad de actuaciones que se pueden llevar a cabo para la puesta al día del sector agrario, se han elegido las relacionadas con el uso eficiente del agua, un bien de primera necesidad cuya prioridad siempre será el abastecimiento humano.

Uno de los factores más importantes en la agricultura, para obtener una buena producción, es la optimización de la irrigación dependiendo del cultivo a tratar. Entre los factores que afectan directamente a una irrigación eficaz se encuentran: el horario en que se realiza, la humedad en el terreno y la temperatura ambiente. Si no se lleva un control sobre estos valores, la eficacia y eficiencia de la irrigación no es óptima originando una pérdida de recursos.

El proyecto tiene por finalidad crear un sistema de riego automático que le permita al usuario poder optimizar el uso del agua y optimizar el tiempo del usuario. El sistema será realizado con el uso de ESP32 y sistema de comunicación inalámbrica MQTT, además del uso de sensores que permitan analizar su entorno actual como sensor de humedad del suelo, sensor de distancia, etc. El sistema permitirá al usuario regar una maceta cada vez que lo necesite es decir, cuando la tierra esté seca. También tendrá una interfaz para poder visualizar de manera gráfica los datos recolectados por el sistema en una base de datos.

1.1. Revisión del estado del arte

En el estudio de Guijarro et al. (2018), tiene como objetivo el riego automático para huertos domésticos con el fin de evitar que estos se sequen por la falta de hidratación, al mismo tiempo se quiere optimizar el recurso hídrico, para lograr este objetivo se hizo uso de tecnologías como sensores de humedad y temperatura, una placa Arduino y un dispositivo Bluetooth, esto se complementó con el desarrollo de una aplicación Android utilizando ANDROID STUDIO, para desarrollar la lógica de este programa se utilizaron como entrada la información climática obtenida por los sensores estos son tres parámetros,

la humedad de la tierra, la temperatura y la humedad del aire. Como resultados se evaluó el sistema de riego por un mes en donde se observó que ya no se producía una resequedad o sobrehidratación, además se informaba correctamente por la aplicación móvil los estados en los que se encontraba el suelo. Finalmente, se concluyó que la implementación de este sistema ayuda a optimizar el tiempo y la forma de realizar el riego de un huerto doméstico a bajo coste, además se resalta el fácil uso de este mismo, por lo que cumple con sus objetivos planteados inicialmente.

En la tesis de grado de Cañón y Cifuentes (2018), se identifica como problemática para los pequeños jardines domésticos, en donde por un inadecuado riego en donde no se consideran factores como la temperatura o humedad del ambiente llevan a un estado de deshidratación o congelamiento de las plantas, por ello se plantean como objetivo realizar un prototipo de sistema automatizado de riego para jardines, para ello hicieron uso de sensores y Arduino. Adicionalmente, se desarrolló una aplicación móvil para ver en tiempo real gráficas y estadísticas. El tipo de investigación utilizada es descriptiva debido a que se observan las necesidades existentes en cuanto al sistema de riego en los hogares, para utilizar esta información en su sistema de riego. También se utilizó la metodología ágil SCRUM, para el desarrollo del software. Se realizaron pruebas de caja blanca y caja negra en donde probaron cada funcionalidad del sistema, como resultado se obtuvo un correcto funcionamiento de estas. Finalmente, como conclusión se obtuvo que la implementación de un sistema de riego automatizado, facilitan y optimizan esta acción, además que es más eficiente ya que se toman datos climáticos para realizar el riego, manteniendo informado al usuario mediante una aplicación móvil.

En el trabajo de fin de grado realizado por Castillo (2021), se identifica como problemática la falta de tecnologías para el riego de los cultivos, existiendo pocos estudios sobre el riego tecnificado, además de ineficiente monitoreo de la humedad de los suelos de los cultivos, todo ello conlleva a un mal uso del agua el cual es un recurso hídrico escaso. Por ello se plantea como objetivo diseñar un sistema automatizado de riego por goteo para el cultivo de pimiento haciendo uso de Arduino UNO y Tinkercad enfocado en la humedad de suelo, se hicieron estudios para saber los requisitos de la humedad necesaria para este suelo y la estimación del agua que será necesaria para el riego del cultivo en cuestión. Como resultado se obtuvo la simulación del proyecto implementado en Tinkercad utilizando el lenguaje de programación usado por Arduino en donde se realizaron diversas simulaciones

en distintos suelos y variando las variables climáticas y la humedad de suelo para evaluar su correcto funcionamiento. Finalmente, se concluyó que se obtuvo una óptima optimización del riego haciendo uso de en total 12 sensores de humedad con lo cual se optimiza el recurso hídrico y se incentiva el uso de este tipo de tecnologías en diferentes cultivos.

Como hemos visto en los estudios anteriores se destaca la importancia de los sistemas de riego por diferentes factores entre ellos se puede destacar la optimización de recurso hídrico, ya que solo se procede con el riego si cumple con ciertos parámetros como la humedad del suelo, aire y temperatura. Otra importancia se refleja en el mayor cuidado del cultivo debido a que se evita la deshidratación o la sobrehidratación de las plantas, ellos también conlleva a evitar enfermedades de las plantas por exceso de humedad.

1.2. Planteamiento del problema

1.2.1. Situación Problemática

En la agricultura o incluso en cultivos domésticos existen diversos factores que pueden poner a estas plantas en situaciones de riesgo. El exceso de agua en los suelos debido a un mal manejo del riego, es uno de los factores que contribuyen a una salinidad en los suelos, esta salinización en los suelos evita que una planta absorba de manera adecuada sus nutrientes, acumula iones que pueden resultar tóxicos, desestabiliza el ph de las plantas, entre otras consecuencias, por lo cual se debe tener en cuenta para asegurar la vitalidad de las plantas.

El sector agrícola consume un tercio de las reservas de agua, si a esto sumamos el incremento de la demanda de usos particulares por nuestro estilo de vida y el descenso de precipitaciones por el cambio climático, es probable que, en un futuro, encontremos dificultades para satisfacer las necesidades hídricas en todos los ámbitos, incluida la agricultura. Esto sucede por el uso ineficiente del agua por parte de los propios agricultores al no tener en cuenta el uso de artefactos que le permitan medir el nivel de agua que tiene una planta en cierto momento.

En cuanto a técnicas de riego, actualmente los trabajadores del sector agrícola en su mayoría no usan un riego localizado principalmente por la falta de tiempo y tecnicismo que eso requiere. Con el objetivo de aumentar la eficiencia en las aplicaciones de agua a un cultivo, se debería utilizar el riego localizado, para tener un

mejor aprovechamiento del agua y poder ahorrar entre un 40-60 % de agua con respecto a otros sistemas de riego, también permite reducir la mano de obra, sobre todo porque disminuyen las malas hierbas al no humedecer la totalidad del suelo

En cuanto a optimización de recursos, los agricultores al tener cientos de cultivos en su tierra, tienen el problema de usar mano de obra excesiva para poder regar justamente estas plantas. Adicionalmente, en cuanto a cultivos domésticos, debido a la falta de tiempo no se cuida de manera adecuada estos cultivos. Por estos motivos se necesita optimizar estos problemas y poder hacer el riego de una manera más rápida, autónoma y con menos uso de mano de obra.

Finalmente, las tareas en el campo de asistencia técnica son múltiples, consolidar el crecimiento agrario exigirá el desarrollo de factores productivos y el impulso de la innovación tecnológica, la falta de uso de tecnología en la agricultura trae consigo serias desventajas como alto uso de mano de obra, disminución de la producción, desperdicio de agua, entre otros mencionados anteriormente.

1.2.2. Formulación del problema

¿Cómo un sistema de riego automático, con un sistema de gestión de datos inalámbrico, impacta de manera positiva al área de la agricultura en la optimización de recursos imprescindibles del ser humano como el tiempo, el agua y la mano de obra?

1.3. Objetivos

1.3.1. General

Implementar un sistema de riego automático con un sistema de gestión de datos inalámbrico para optimizar el uso de recursos de tiempo, costos e insumos en el área de la agricultura.

1.3.2. Específicos

- Utilizar el riego automático para regar los cultivos de forma óptima solo cuando sea necesario, es decir, cuando los sensores utilizados nos lo indiquen.
- Implementar el riego para cultivos utilizando la técnica de riego localizado.

- Crear el sistema de riego completamente automático y evitar el uso de mano de obra excesiva.
- Implementar una comunicación inalámbrica entre los dispositivos y un sistema de recepción de datos mediante WiFi.
- Generar una base de datos y la exportación de sus datos registrados.
- Desarrollo de gráficas de información para el usuario final.
- Cumplir los objetivos 6 y 15 del desarrollo sostenible.

1.4. Alcances

- El sistema es un prototipo y permitirá el riego automático de cualquier planta en una maceta que cumpla la distancia de 48 cm de ancho y 18 cm de largo.
- La comunicación entre los dispositivos es mediante WiFi usando el protocolo MQTT.
- El sistema de gestión de base de datos es de manera local y se hizo usando la tecnología de Node-RED y MySQL.
- El dispositivo se conecta a energía mediante un adaptador.
- El código e información del proyecto será puesto en un repositorio público.

2. Marco teórico

En este apartado, se presentan los conceptos útiles que ayudan a comprender el proyecto.

2.1. Internet de las Cosas (IoT)

Es una infraestructura global de la sociedad de la información, el cual ofrece servicios avanzados mediante la interconexión de objetos, ya sean estos físicos o virtuales, gracias a la interoperabilidad de tecnologías de la información y la comunicación (TIC)(Ramirez y Rodriguez, 2016).

2.2. Automatización

Consiste en el uso de la tecnología para realizar tareas con una mínima o nula intervención humana, su implementación y aplicación es muy variada desde la fabricación de cosas hasta la robótica o software que toma decisiones empresariales (Red Hat, 2022).

2.3. Sistema de Riego Automático

Es un sistema que provee de agua a los cultivos de manera automatizada en la cual emplea técnicas de aspersión o el goteo, su aplicación presenta muchas ventajas como el ahorro de tiempo y esfuerzo, optimiza el uso del agua, riego programado y fácil mantenimiento (maher, 2022).

2.4. Arduino IDE

Conjunto de herramientas de software que permite desarrollar y almacenar el código necesario para el funcionamiento de la placa Arduino, permite escribir, depurar, editar y grabar los programas de manera fácil ya que tiene una gran accesibilidad (Software Arduino, 2018).

2.5. Wi-Fi

Wi-Fi es una marca de Wi-Fi Alliances (Alianza Wi-Fi en español). Se trata de una organización sin ánimo de lucro que promueve y certifica la tecnología y productos Wi-Fi, comprobando que se ajustan a los estándares de interconectividad compatibles con IEEE 802.11 (especifica las normas de funcionamiento de una red de área local inalámbrica) (Profesional review, 2017).

2.6. MQTT

Son las siglas de MQ Telemetry Transport, es un protocolo de comunicación machine to machine de tipo message queue, basado en la pila TCP/IP como base para la comunicación (Luis Llamas, 2019).

2.7. Node-Red

Es una herramienta de programación visual, que conecta dispositivos de hardware, API y servicios en línea de formas nuevas, en donde su editor se basa en un navegador que facilita la conexión de flujos mediante una amplia gama de nodos que pueden implementarse en tiempo de ejecución (Node-RED, 2022).

2.8. MySQL

Es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (ComputerWeekly.es, 2022).

2.9. Librerías de Arduino

2.9.1. WiFi.h

Es una biblioteca o librería que permite la conexión a la red (local), además puede crear instancias de servidores, clientes y enviar/ recibir paquetes UDP a través de WiFi.

2.9.2. PubSubClient.h

Librería compatible con Arduino y ESP8266, hace que la placa de se comporte como un cliente MQTT, en donde se puede publicar mensajes y suscribirse a un tópico para recibir mensajes.

2.9.3. LyquidCrystal_I2C.h

Una biblioteca para pantallas LCD I2C. La biblioteca permite controlar pantallas I2C con funciones extremadamente similares a la biblioteca LiquidCrystal.

2.9.4. DHTesp.h

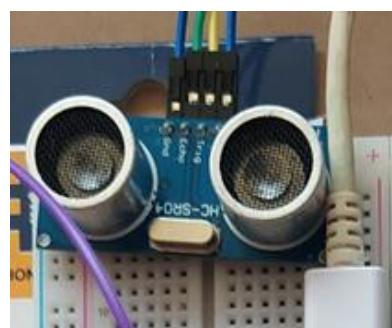
La librería DHT proporciona un conjunto de funciones para el control correcto de sensores de humedad y temperatura.

3. Componentes del sistema de riego

En este apartado, se describe el sistema de riego y sus componentes electrónicos principales.

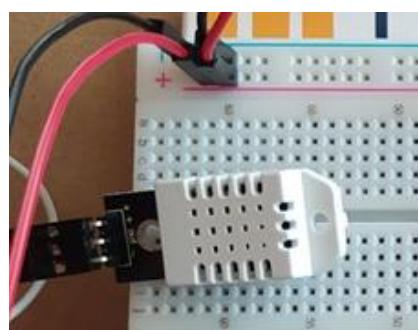
3.1. Sensor ultrasonido HC-SR04

El sensor HC-SR04 compuesto por un emisor y un receptor de ultrasonidos que permiten medir la distancia a la se encuentra un objeto frente a él, enviando un pulso ultrasónico y midiendo el tiempo del transcurso del pulso (Solectro, 2021). El funcionamiento en el proyecto consiste en medir el nivel del agua para verificar la disponibilidad de este recurso.



3.2. Sensor DHT22

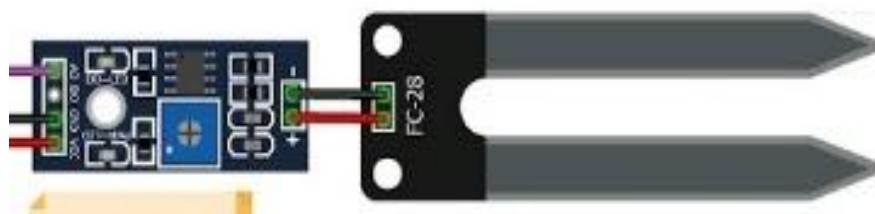
El sensor DHT22 es un sensor digital de temperatura y humedad relativa. Este integra un sensor capacitivo de humedad y un termistor para medir el aire circundante y muestra datos mediante una señal digital en el pin de datos. El funcionamiento en el proyecto consiste en medir si el cultivo se encuentra en una humedad óptima y el otro funcionamiento es medir la temperatura que ayudará a decidir si se riega o no la planta.



3.3. Sensor de humedad del suelo FC-28

Los sensores de humedad miden una señal eléctrica que calcula la cantidad de agua en el suelo, lo cual permite determinar el volumen de agua almacenado en este. Por lo general los sensores de humedad se componen de una sonda con electrodos en un extremo y una tarjeta de acondicionamiento. Los electrodos van insertados en el suelo. Conectando los electrodos a la tarjeta de acondicionamiento, el cual tiene la finalidad de entregar una salida digital “DO” y otra analógica “A0”. Además, de poder disponer con el pin de fuente de energía que viene a ser el de voltaje de alimentación “VCC” y otro pin a tierra “GND”.

El funcionamiento del sensor en el proyecto consiste en establecer si la tierra está seca, húmeda o mojada de acuerdo a los rangos establecidos, que son los siguientes: de 0-300 tierra mojada, de 300-700 tierra húmeda y de 700-1023 es tierra seca. Para ello se utiliza su entrada analógica.



3.4. Relé

Es un interruptor mecánico operado eléctricamente para poder encender o apagar, dejando pasar o no la corriente, y se puede controlar con voltajes bajos. El funcionamiento en el proyecto consiste en conectar y otorgar voltaje a la bomba de agua para su funcionamiento en el sistema de riego.



3.5. Display LCD I2C

El adaptador para LCD 16×2 por I2C con Arduino nos permite ahorrar bastante tiempo en las conexiones, pues ya incluye el potenciómetro para regular el contraste de la pantalla. También incluye todo lo necesario para el funcionamiento del backlight (retroiluminación). Para controlar el contraste de los dígitos en el LCD solo necesitamos girar el potenciómetro que se encuentra en el módulo, hasta quedar conformes con el contraste mostrado.

El funcionamiento consiste en mostrar la temperatura, la humedad ambiental y la humedad del suelo. Además también se encarga de mostrar mensajes de la situación del sistema, como “Agua insuficiente”.



3.6. LED

Dispositivo que permite el paso de corriente en un solo sentido y que al ser polarizado emite un haz de luz, al recibir corriente eléctrica emite luz. Consta de dos pines: ánodo para energía y cátodo para tierra El funcionamiento en el proyecto consiste en servir como un medio para alertar sobre el nivel de agua en el recipiente de la bomba, el cual el LED emite luz si se encuentra parcialmente llena.



3.7. LED RGB

Significa LED Red, Green y Blue que significan en español rojo, verde y azul respectivamente. El dispositivo LED RGB combina estos tres colores para producir una diversidad de tonos de luz. A diferencia de LED convencional LED RGB consta de cuatro pines: R para el rojo, B para el azul, G para el verde y GND para tierra. El funcionamiento en el proyecto consiste en la verificación de disponibilidad de agua en el recipiente de la bomba emitiendo una luz de color verde, y sobre el funcionamiento correcto de la bomba para regar las plantas emitiendo una luz azul.



3.8. Resistencias Eléctricas

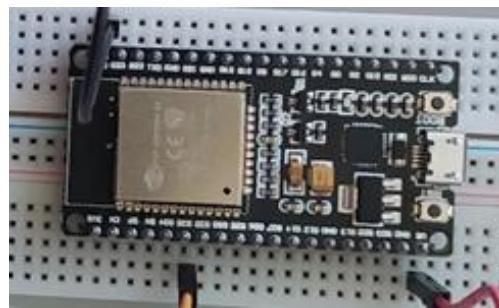
La resistencia eléctrica es la oposición al paso de la corriente eléctrica. Su funcionamiento en el proyecto consiste en disminuir la corriente eléctrica al LED RGB y al LED. Las resistencias son de 220 ohmios.



3.9. Microcontrolador ESP32

El microcontrolador ESP32 es una solución de Wi-Fi/Bluetooth todo en uno, integrada y certificada que proporciona además de una radio inalámbrica y dispone con un procesador integrado con interfaces para conectarse con varios periféricos.

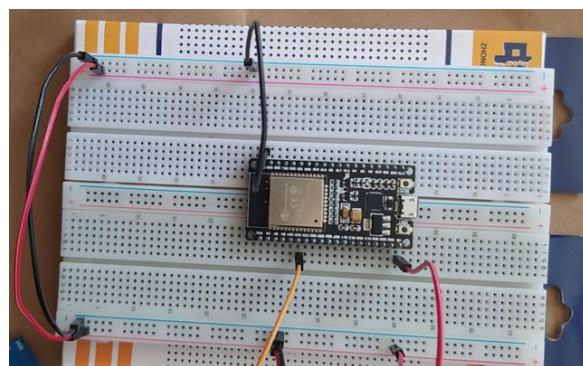
Su funcionamiento en el proyecto consiste en ser un controlador entre los componentes electrónicos utilizados y un operador para el intercambio de información.



3.10. Protoboard

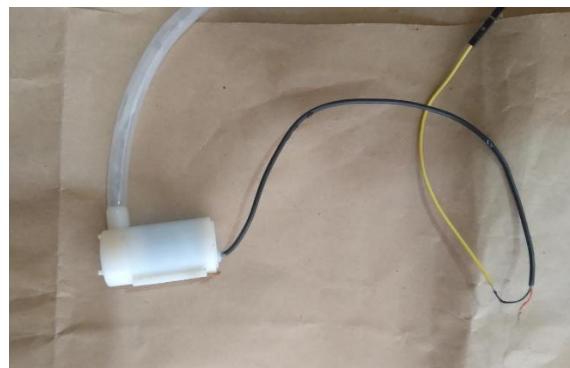
El protoboard es una tablilla de pruebas, que internamente está dividida por columnas verticales de material conductor, que te permite construir prototipos, realizar prácticas y simulaciones de circuitos electrónicos.

En el proyecto se usó 2 protoboards para realizar todo el circuito. Además el ESP32 va conectado al protoboard para que facilite el uso de sus pines.



3.11. Bomba de agua

Denominado mini bomba de agua sumergible que cuenta con un voltaje entre 2.5V - 6V DC, que permite tener un flujo de agua de hasta 2L en un determinado tiempo dependiendo del voltaje de alimentación. Su funcionamiento consiste en fluir una corriente de agua para el regado de las plantas a través de un pequeño tubo en la salida de la bomba, debe ser alimentada con el voltaje deseado por ESP32, y sumergida completamente en líquido, en un recipiente de plástico.



3.12. Adaptador de corriente

El adaptador de corriente sirve para alimentar con corriente a aparatos electrónicos. Su funcionalidad en el proyecto fue alimentar al módulo ESP32. Esta alimentación debe ser a lo mucho 5 voltios, en caso contrario se corre el riesgo de que se queme los componentes electrónicos. Es por esto que se utilizó un adaptador de corriente de 5 voltios.



3.13. Descripción del sistema de riego

Se trata de un sistema de riego que provee agua a un área localizada, en una maceta contenida de plantas, de manera automatizada y que emplea el goteo como tipo de riego. Para nuestro proyecto, se consideraron factores ambientales como la temperatura,

humedad, humedad de suelo, captados por los sensores, que determinan si el cultivo debe ser regado o no y si la planta se encuentra en condiciones ambientales óptimos.

En nuestro proyecto se dispone de una bomba de agua para el flujo de agua, esta es almacenada en un recipiente de plástico y fluye a través de una tubo pequeño insertado en la bomba, en la cual será activada acorde a las condiciones dadas por el sensor de humedad de suelo, en la cual indicará si está en una cierta medida 0-1023 u lo cual nos indica el estado de la tierra para poder o no el regado correspondiente. Se definieron tres estados para el estado de suelo: tierra seca (700-1023 u), tierra húmeda (300-700 u) y tierra mojada (0-300 u). Además, debemos tener en cuenta que cuando se presente el estado de tierra seca se hace el riego por goteo del cultivo. Además, cuando se presente el caso de tierra húmeda se evalúa la temperatura ambiental, para lo cual nos ayudará el sensor DHT22, esta temperatura debe ser mayor a 23°C para que se pueda regar. En caso se presente el estado de tierra mojada no se realizará ningún riego. Además el nivel de ambiente óptimo que deben estar las plantas será verificado por el sensor DHT22, la humedad debe estar en un nivel de rango 50%-75% y la temperatura entre 18°C - 23 °C.

Además en el proyecto se dispone de un display LCD que muestra los niveles ambientales del cultivo. Y con la ayuda de los LED Y LED RGB permiten saber cuándo se necesita llenar el contenedor con agua, en ese caso se prende la luz roja, cuando se necesita cambiar de lugar el cultivo ya que hay exceso de humedad prende la luz de color multicolor y cuando se está regando prende la luz de color azul.

4. Implementación del sistema

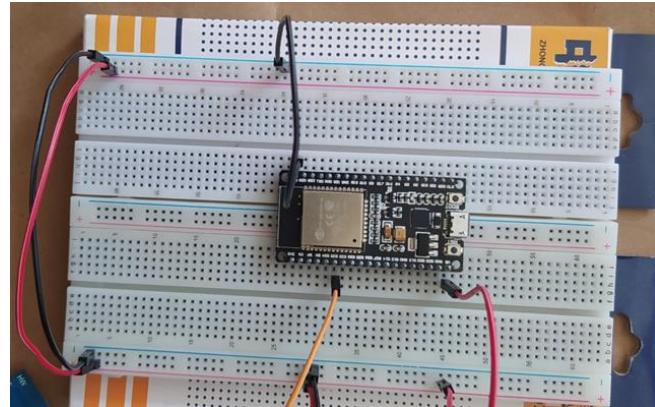
4.1. Diseño metodológico del sistema

4.1.1. Construcción del circuito

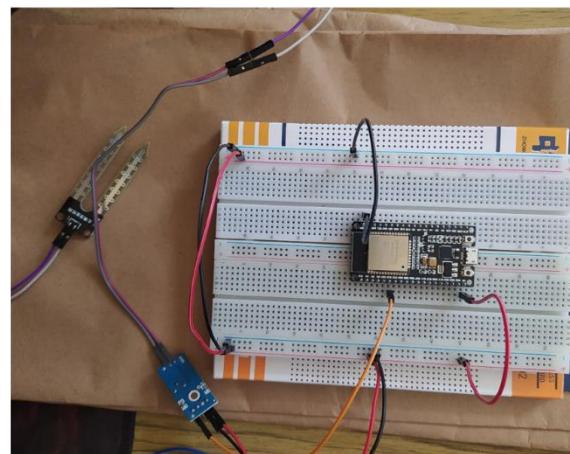
Para la construcción del Sistema de Riego, se utilizaron los diversos componentes electrónicos mencionados en el apartado anterior. Estos componentes van a ser relacionados en un circuito, para esto se realizó el cableado necesario para el funcionamiento del sistema.

A continuación se presentarán los pasos para realizar el cableado del circuito.

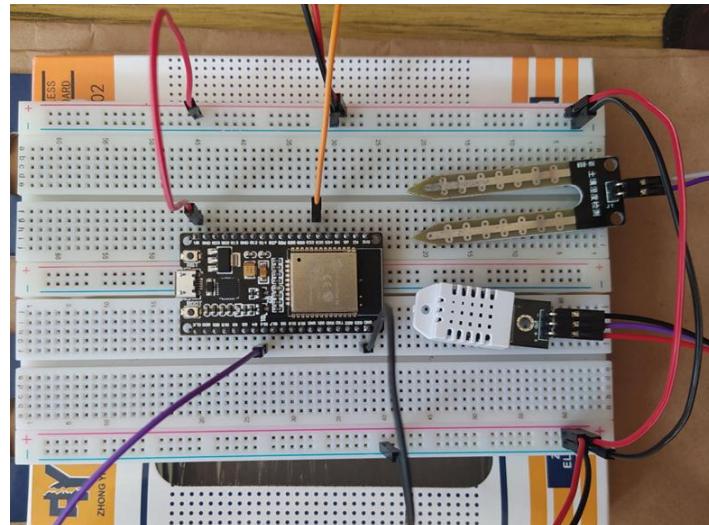
Para empezar se juntó dos protoboards para que pueda alcanzar todo el cableado necesario del sistema. Luego, el ESP32 se conectó en el protoboard unificado. Y se alimentó el protoboard con un voltaje de 3.3v y con tierra



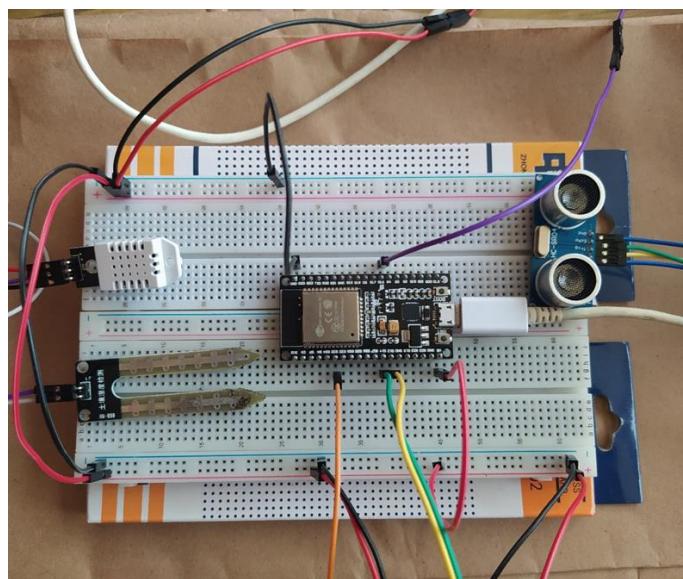
Después de haber alimentado el protoboard con corriente y tierra, se conectó el sensor de humedad del suelo. El sensor de humedad del suelo cuenta con un convertidor el cual tiene dos lados, uno de 4 pines y otro de dos. En el lado de 4 pines se debe de recibir una cantidad de voltaje en el pin VCC y tierra en su pin GND. El voltaje y la tierra lo obtenemos del protoboard que se encuentra alimentado. Además en ese lado también se encuentra el pin analógico (A0) el cual se conecta con el pin 35(G35) del módulo ESP32. Ahora bien, el lado donde hay 2 pines se conecta con los electrodos, en el cual cada pin representa energía y tierra.



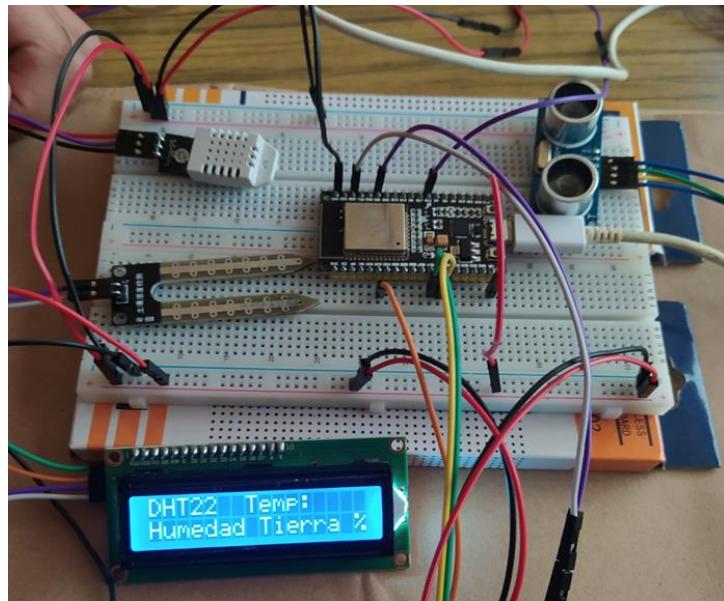
El sensor de temperatura y humedad relativa, DHT22, necesita de tres conexiones. Una de energía, de tierra y de un pin digital el cual sirva para la lectura. El voltaje va en el pin “+” del DHT22 y la tierra va en el pin “-” del DHT22. Y el pin OUT del DHT22 se conecta con el pin 16 (G16).



El sensor ultrasónico se conecta a dos pines digitales en el ESP32, uno para el Trigger o disparador y otro para el Echo o receptor. En nuestro caso se conectó en el pin G12 el Trigger y en el pin G14 el pin Echo. Además el sensor ultrasónico debe de recibir una cantidad de voltaje y tierra en los pines VCC y GND respectivamente. El voltaje y la tierra se obtienen del protoboard.



Después de incluir todos los sensores al circuito, incluiremos un LCD 16x2 I2C para la salida de datos. La ventaja de usar una pantalla LCD I2C es que el cableado es realmente simple. Solo necesita conectar los pines SDA y SCL. El SDA se conectó en el pin digital 21(G21) del módulo ESP32 y el SCL se conectó en el pin 22(G22) del ESP32. Además se debe conectar tierra y corriente al lcd en sus respectivos pines. La corriente debe ser de 5 voltios.

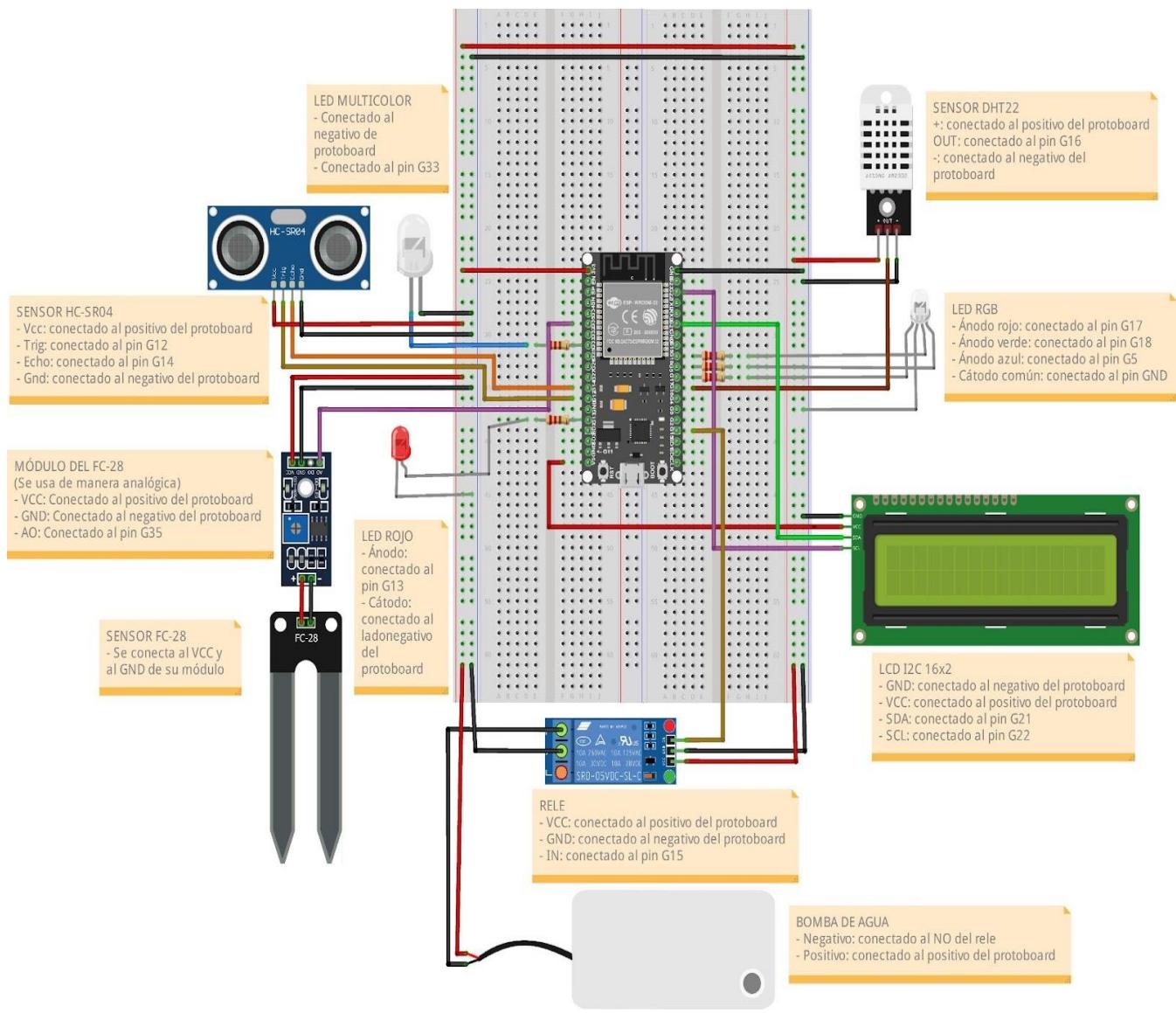


Ahora bien el lado negativo o tierra de la bomba del agua va conectada al relé, en el pin NO. Y el positivo de la bomba va conectado al positivo del protoboard. En el caso de relé la entrada llamada “COMUN” va conectado a tierra. Y los pines que están entre los focos que son el VCC, y GND se conectan al positivo y negativo del protoboard, respectivamente. Y el pin IN se conecta al pin 15 (G15) del ESP32.



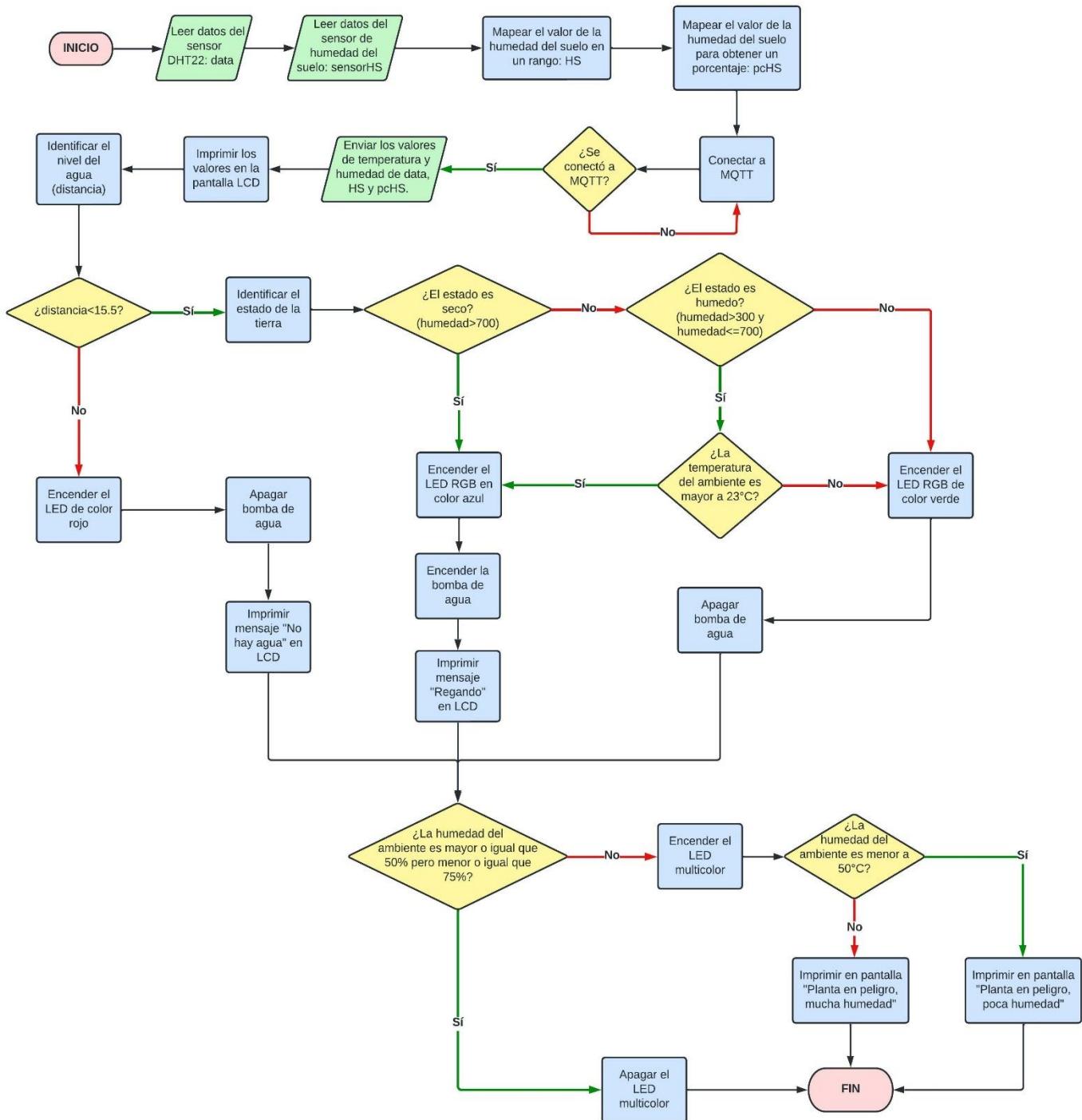
Y por último se realizó el cableado de los led. Para el LED RGB, se conecta el cátodo común al negativo del protoboard, el cátodo común es el pin más largo. En caso de los ánodos, la distribución es la siguiente: el ánodo verde al pin 18 (G18), el ánodo azul al pin 5 (G5) y el ánodo rojo al pin 17 (G17). En el caso del LED rojo se conectó su ánodo al pin 13 (G13) y su cátodo al negativo del protoboard. Y el LED Multicolor se conectó el cátodo al negativo de protoboard y el ánodo al pin 33 (G33).

4.1.2. Diagrama del circuito

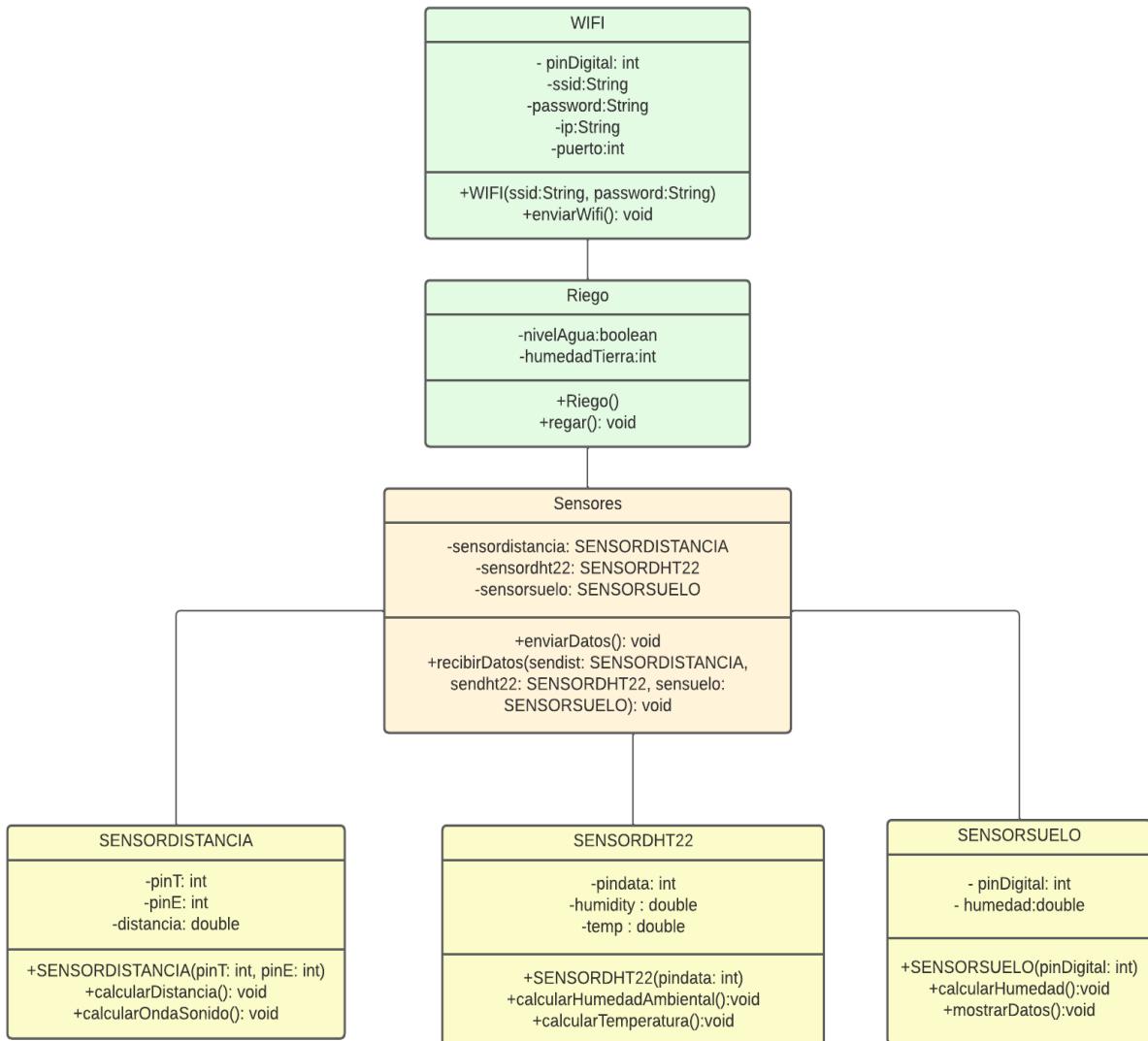


fritzing

4.1.3. Diagrama de flujo de funcionamiento



4.1.4. Diagrama de clases



4.2. Lugar de trabajo

4.2.1. Caracterización del cultivo

Para el uso del sistema debe ser implementado en un cultivo. Es por este motivo que en el proyecto se usó una maceta rectangular, este contiene tres plantas de la misma especie. El tamaño de la maceta es mediano, sus medidas son alrededor de 45 cm de largo y 18 cm de ancho.

Por otro lado, la especie de las plantas es la suculenta. Las plantas suculentas se distinguen de las demás por su gran capacidad de almacenar agua en los tallos, hojas o raíces (Reyes, 2009).

Además Reyes et al. (2011), nos menciona que este tipo de plantas abren sus estomas (poros o aberturas regulables) durante la noche, purificando el ambiente, lo cual le permite maximizar su eficiencia para el uso de agua.

4.2.2. Construcción de la maqueta

Para diseñar la maqueta o un huerto a pequeña escala se usó una maceta rectangular mediana.

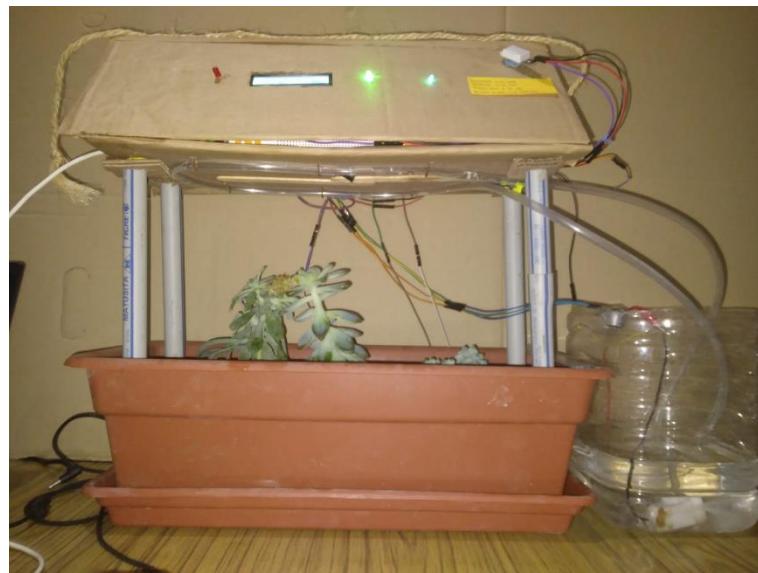


Para implementar todo el circuito se diseñó un techo para la maceta, en donde se colocó la manguera de plástico que va regar el cultivo en forma de goteos dependiendo de las condiciones ambientales que se tienen registrados. Para realizar este techo se usó cartones y cuatro tubos como soporte del techo.



En el techo generado, se colocaron los dispositivos para el funcionamiento del sistema. Por último, al lado derecho de la maceta, se puso un contenedor de agua.

A continuación, la imagen de cómo quedó el sistema de riego.



4.3. Implementación de Node-RED

Para esta sección una precondición se considera ya tener instalado el NODE-RED en el ordenador, entonces a continuación presentaremos la implementación de NODE-RED en nuestro proyecto, se tienen una secuencia de pasos:

4.3.1. Acceder a Node-RED

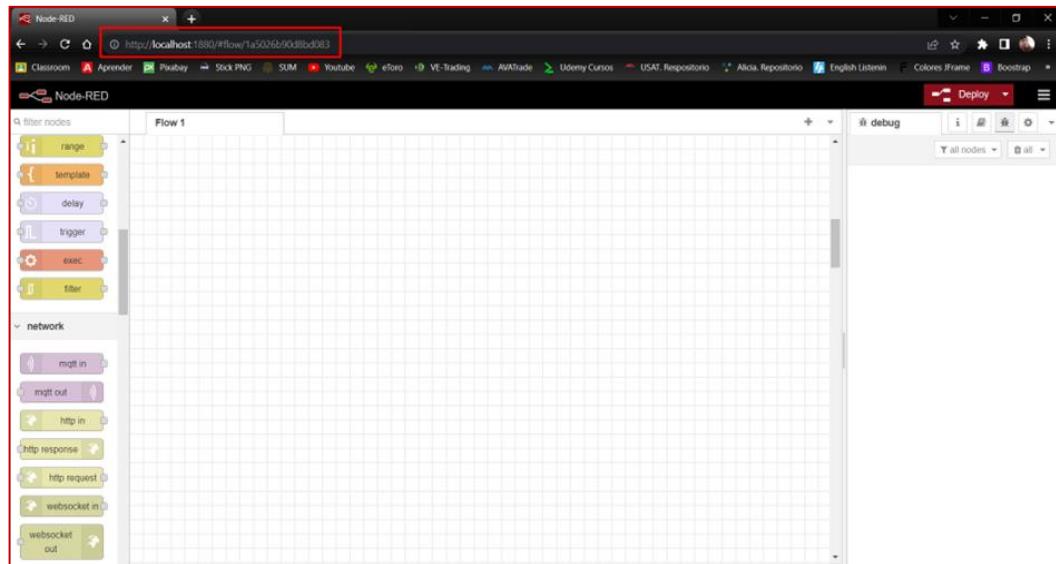
Primero para acceder a NODE-RED en nuestro navegador, debemos activar a través de nuestro terminal (símbolo de sistema) mediante el comando “node-red”, luego podremos acceder a NODE-RED mediante el URL: <http://localhost:1880>



```
node-red
Microsoft Windows [Versión 10.0.19044.1889]
(c) Microsoft Corporation. Todos los derechos reservados.

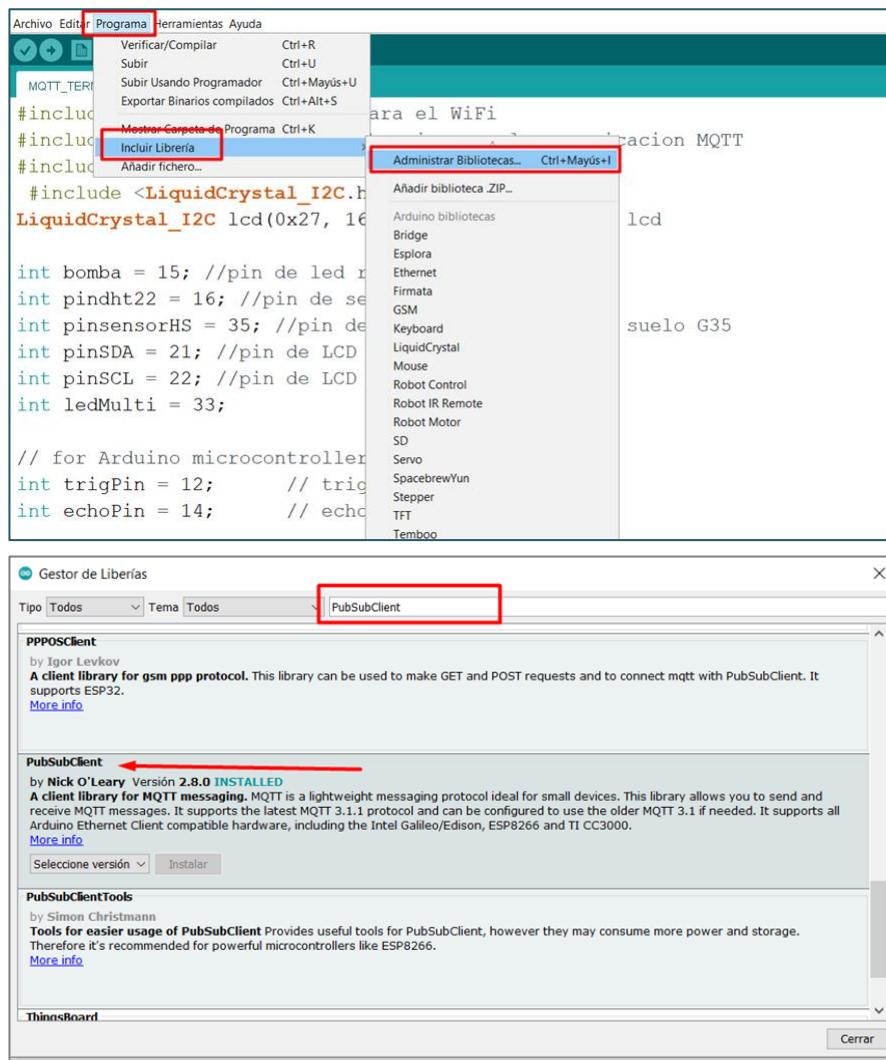
C:\Users\Luis Enrique>node-red
27 Aug 21:15:16 - [info]

Welcome to Node-RED
=====
```



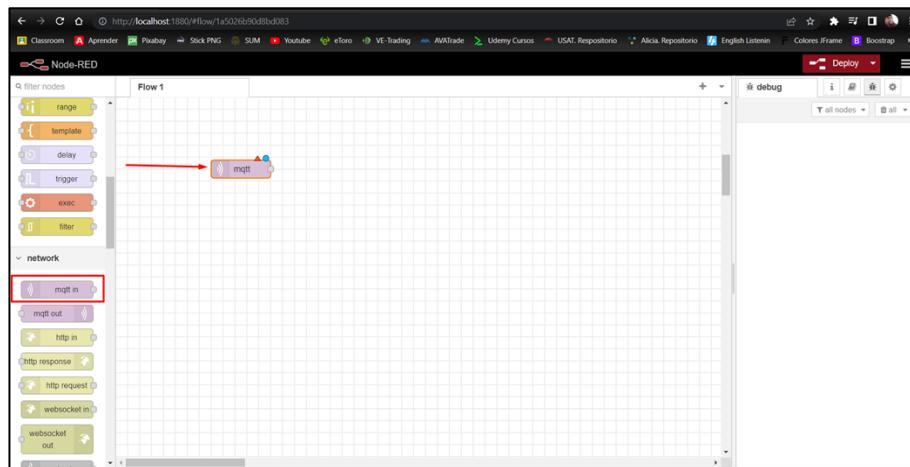
4.3.2. Instalación de la librería PubSubClient

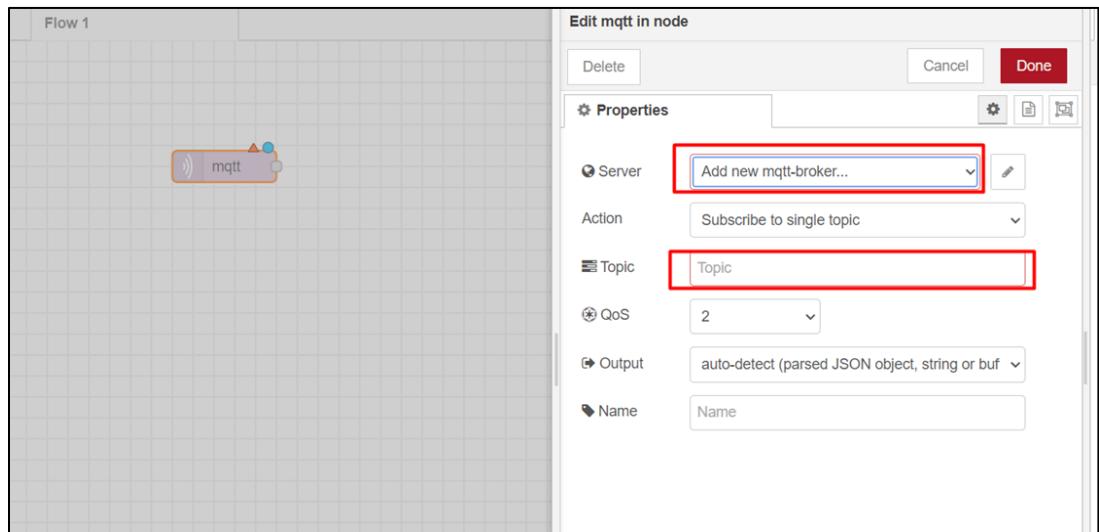
En el Arduino IDE, procedemos a instalar la librería correspondiente PubSubClient. Ingresando a la sección de administración de librerías, buscamos e instalamos la librería.



4.3.3. Configuración de Node-RED

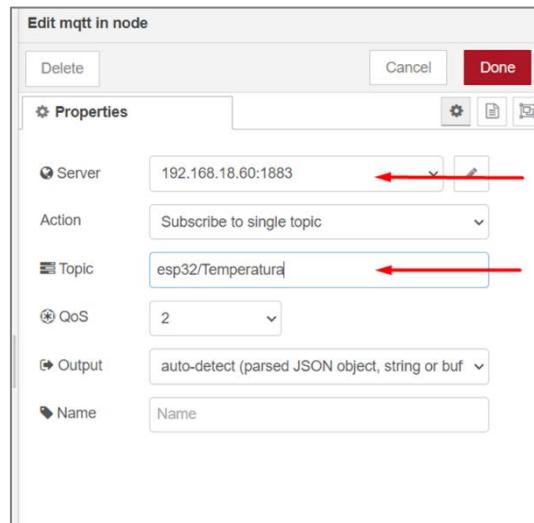
Agregamos un Nodo de entrada mqtt. Le damos doble clic en el Nodo para configurar el Server y Topic que nos están solicitando.



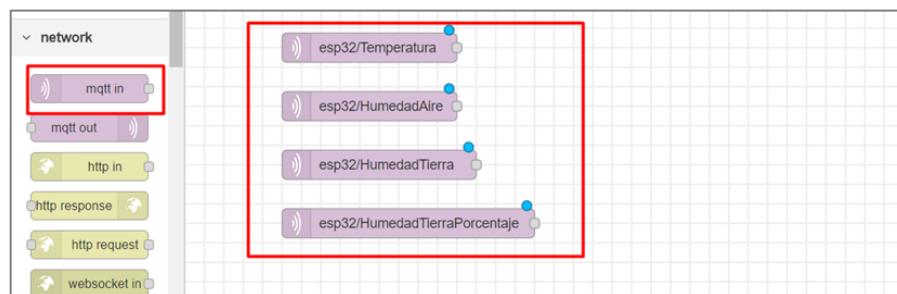


4.3.4. Configuramos el Nodo de entrada mqtt

Configuramos el Nodo de entrada mqtt con el respectivo servidor “192.168.18.60.1883” y el topic “esp32/Temperatura” (para la variable temperatura).

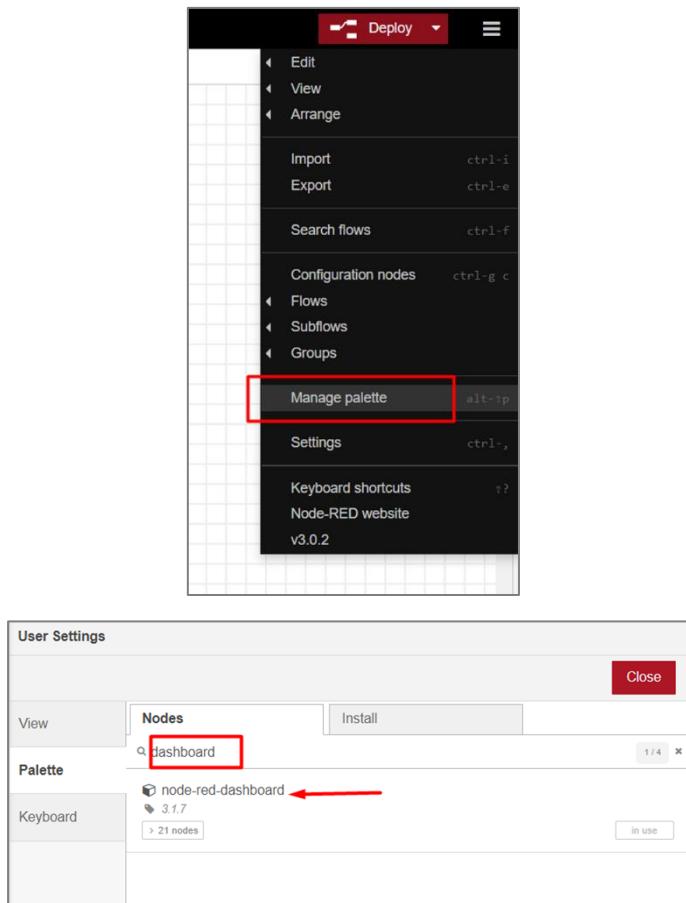


De esta manera realizamos la operación para las variables de humedad de aire, humedad de suelo y porcentaje de humedad de suelo.



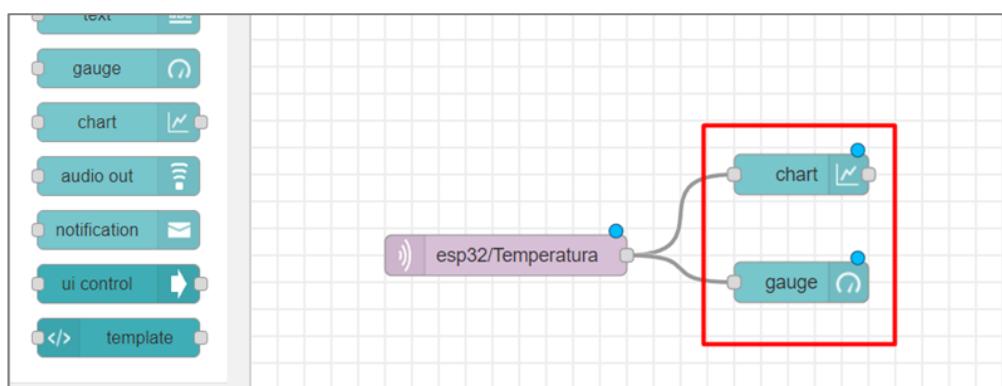
4.3.5. Configuraciones para realización del Dashboard de los resultados

Para utilizar los nodos para realizar los dashboards debemos instalar la librería “node-red-dashboard”. Nos dirigimos a configuraciones e instalamos. De esta manera, podemos utilizar los nodos para las gráficas: “chart” y “gauge”.



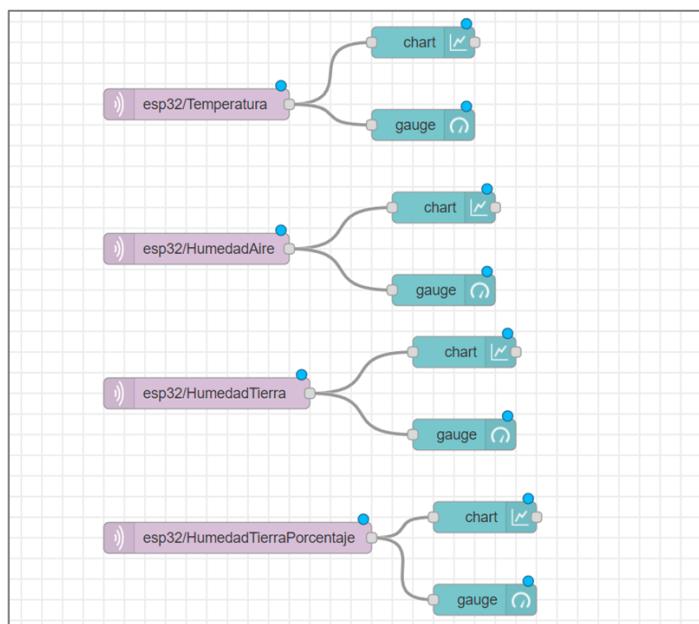
4.3.6. Inserción de los nodos para realizar Dashboard

Ahora que visualizamos en la sección de los Nodos los nodos respectivos para realizar las gráficas. Debemos instalarlos, para relacionarlos y configurarlos dando clic con los nodos de entrada mqtt de la siguiente manera.

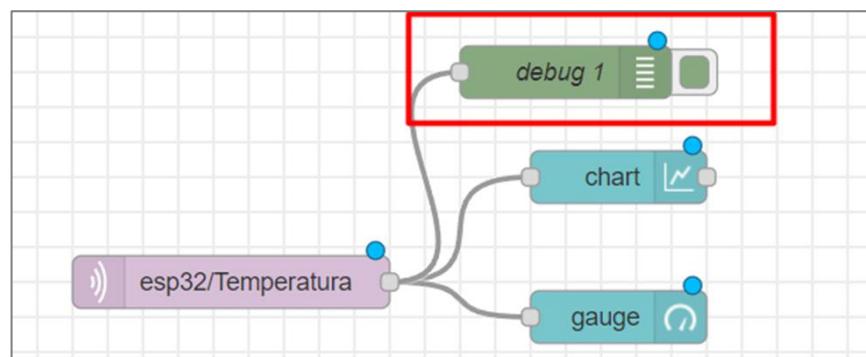




De esta manera realizamos el proceso anterior para las variables restantes: humedad de aire, humedad de suelo y porcentaje de humedad de suelo.

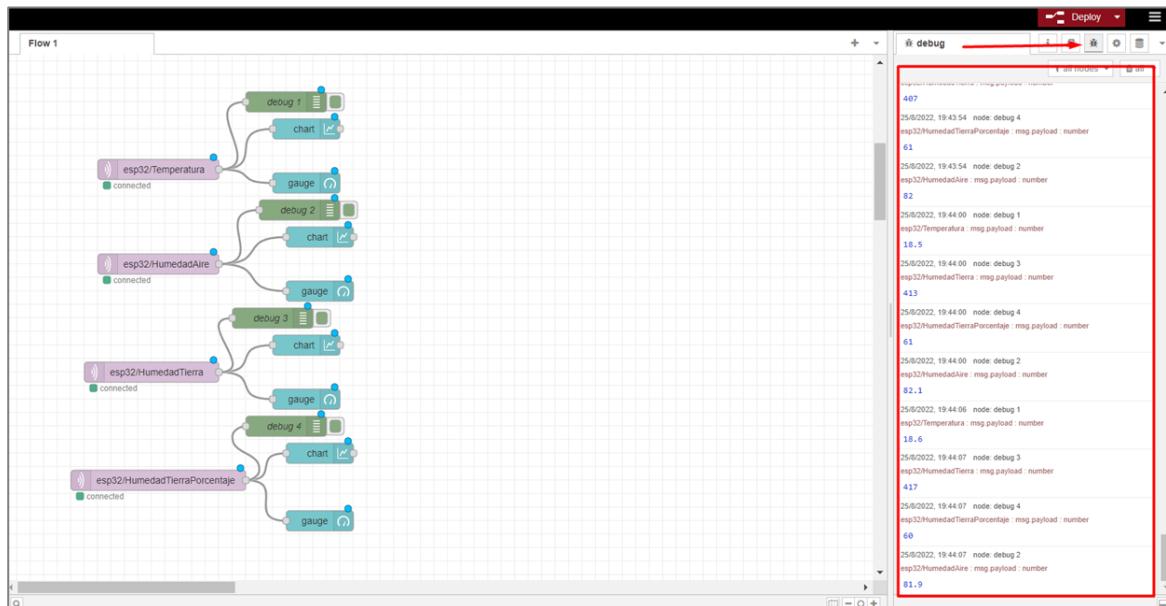


Para verificar que se está llegando los datos, mostraremos estos resultados en consola en NODE-RED agregando un Node “debug”. Lo relacionamos como se observa en la figura posterior. De la misma manera para todos los Nodos de entrada mqtt.

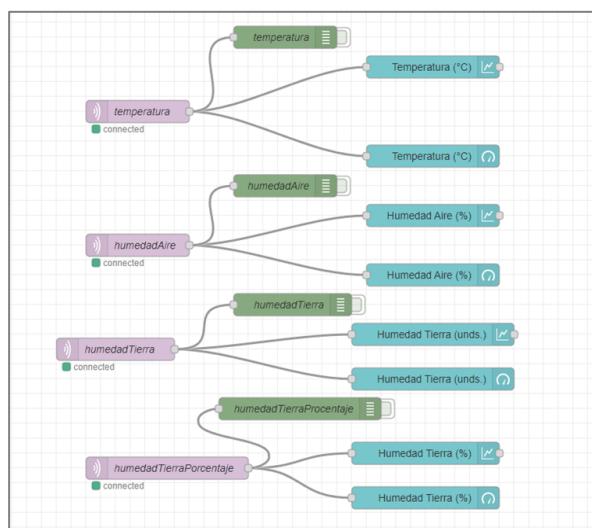


4.4. Ejecución de Node-RED

Ejecutamos el NODE-RED dando clic en “Deploy”. Luego, podremos observar los resultados en la consola en la parte derecha.

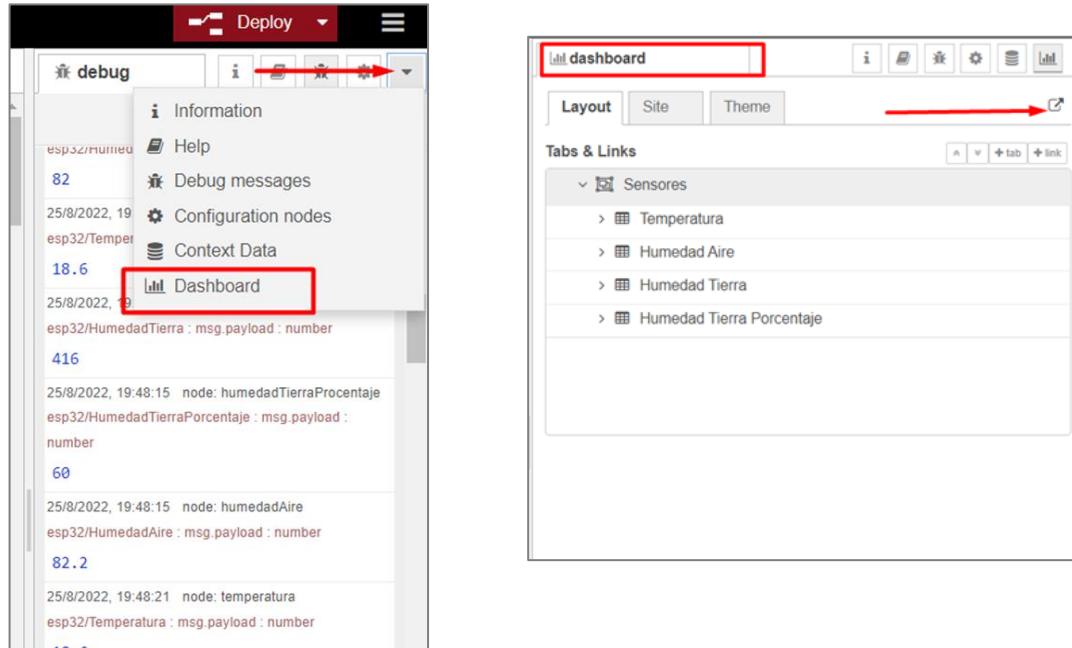


Renombramos todos los nodos de la manera mejor correspondiente para poder tener un mejor enfoque de la siguiente manera.

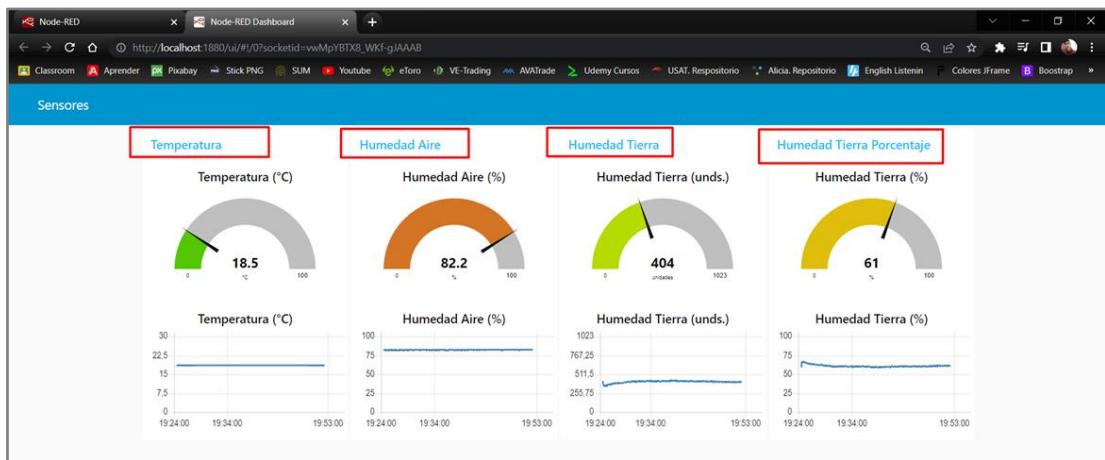


4.5. Visualización a través de gráficos

Para visualizar las gráficas creadas, dashboard, debemos dirigirnos a la opción de “dashboard” entrando en la configuración para cambiar de sección.



De esta manera nos dirige a una nueva pestaña donde se pueden visualizar las gráficas determinadas.

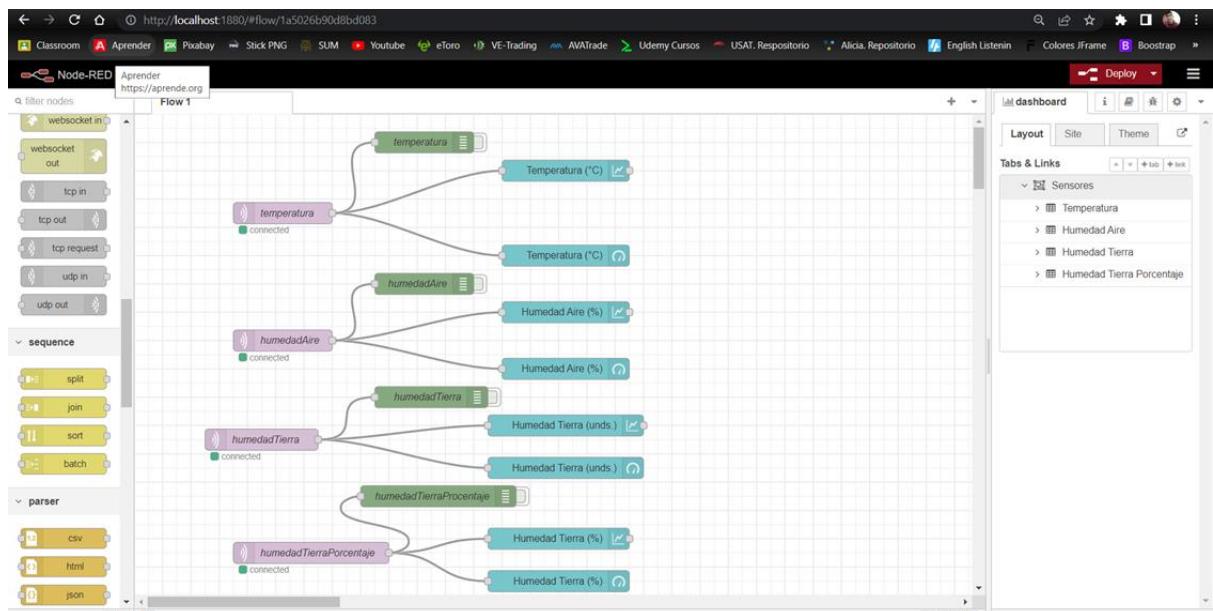


4.6. Implementación de base de datos en Node-RED

Para la implementación de Base de Datos en NODE-RED se necesitará realizar las actividades involucradas en el apartado anterior.

4.6.1. Acceder a NODE-RED

Primero para acceder a NODE-RED en nuestro navegador, debemos activar a través de nuestro terminal (símbolo de sistema) mediante el comando “node-red”, luego podremos acceder a NODE-RED mediante el URL: <http://localhost:1880>



4.6.2. Inserción de un Nodo “join” para unir los mensajes

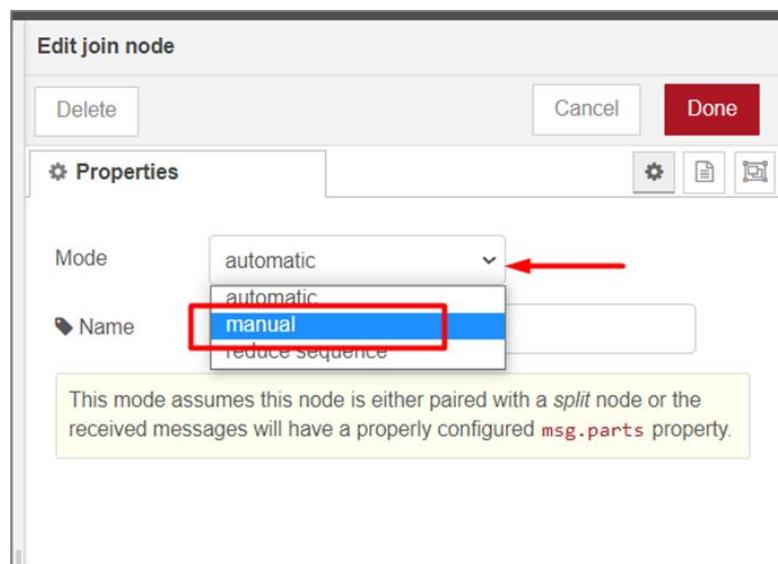
Luego insertamos un nodo “join” para agrupar una secuencia de mensajes en un solo mensaje. Para así agilizar el proceso de guardar los datos en la base de datos. Después de insertarlo relacionamos el nodo “join” con los nodos de entradas como se puede observar en la figura.



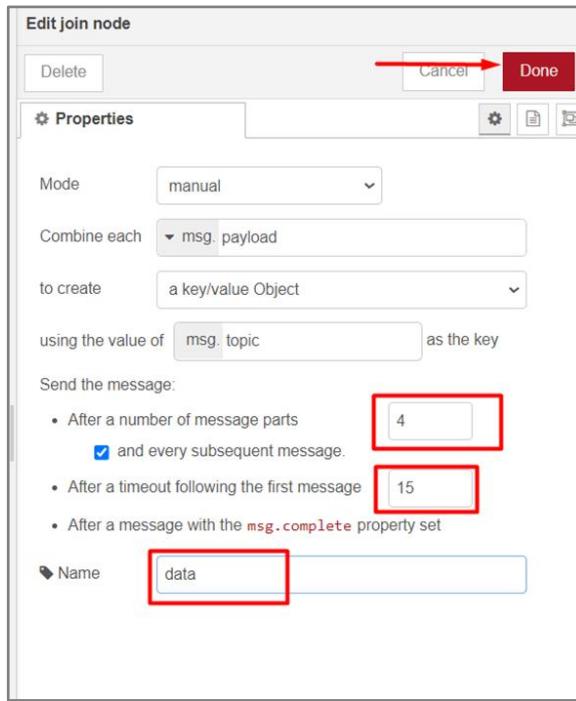


4.6.3. Configuración del Nodo “join”

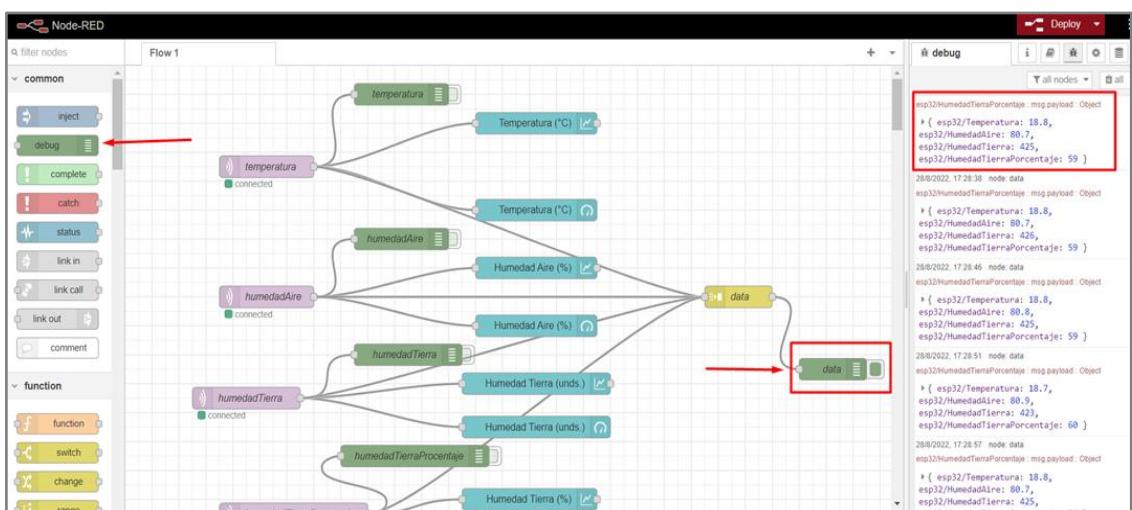
Para configurar el Nodo “join” le damos doble clic y cambiamos el modo por manual.



En la sección de “Enviar el Mensaje” configuramos el número de mensaje que vienen ser cuatro: temperatura, humedad de aire, humedad de tierra y porcentaje de humedad de tierra, asimismo, configuramos el intervalo de tiempo para la salida de datos que serán 15 s. Finalmente, configuramos el nombre del Nodo por “data” y damos clic en “done”.

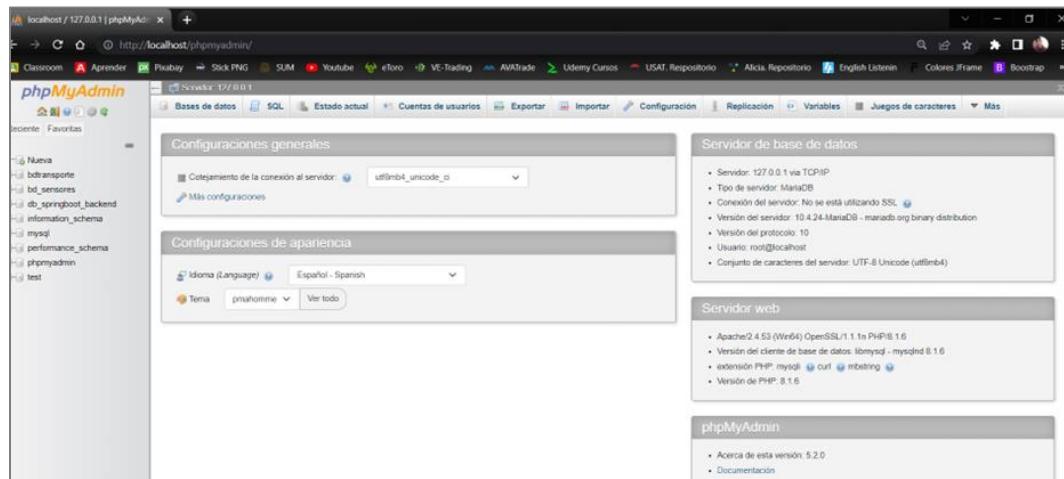
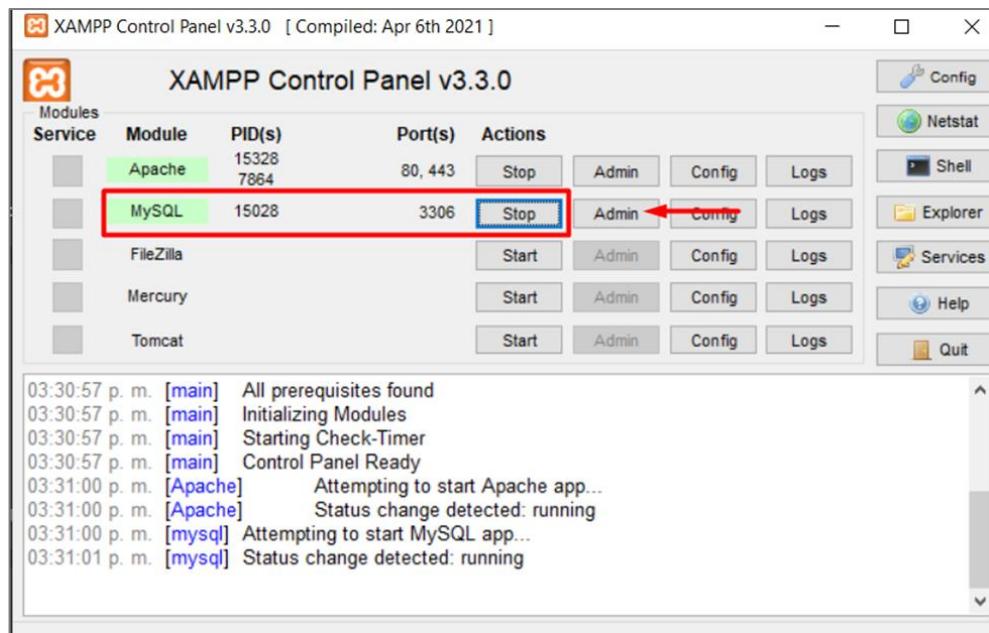


Para verificar que se ha realizado la correcta configuración se imprimirá en consola los datos obtenidos en los Nodos de entrada mqtt utilizando un Nodo “debug” al que denominaremos “data”.



4.6.4. Instalación de una herramienta de desarrollo XAMPP

Para utilizar el motor de base datos MySQL, utilizaremos una herramienta de desarrollo XAMPP, lo cual se necesitará descargarlo, luego de instalarlo activaremos el MySQL. Damos clic en “Start” y luego en “Admin” para ingresar al phpMyAdmin para utilizar MySQL.



4.6.5. Creación de la base de datos

Una vez ingresando en el phpMyAdmin. Crearemos un nueva base de datos lo cual denominaremos “db_sensores”, configuraremos el nombre y damos clic en “Crear”.

The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. In the left sidebar, there's a tree view of databases: bd_transporte, bd_sensores, db_springboot_backend, information_schema, mysql, performance_schema, phpmyadmin, and test. A red arrow points to the 'Nueva' (New) button. The main area is titled 'Bases de datos' (Databases). A modal window titled 'Crear base de datos' (Create database) is open, showing the input field 'db_sensores' and a 'Create' button, both highlighted with a red box. Below the modal, a table lists existing databases with their character sets and collations. The table has three columns: 'Base de datos', 'Cotejamiento', and 'Acción'. The 'db_sensores' entry is listed under 'Base de datos'. At the bottom of the table, it says 'Total: 8'.

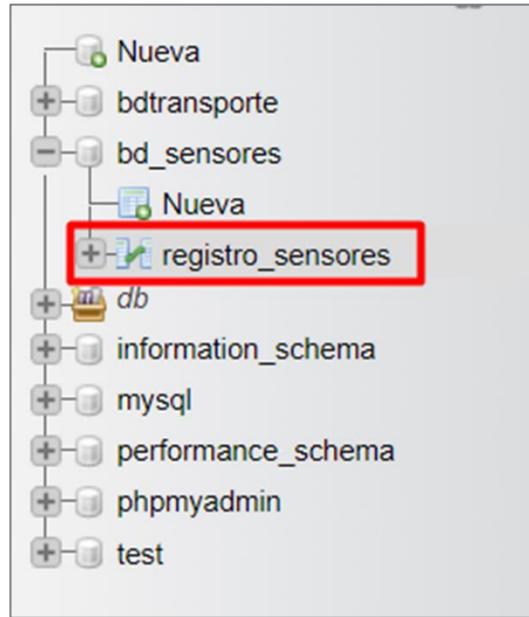
4.6.6. Creación de tabla en la base de datos

Luego crearemos una tabla a la que denominaremos registro_sensores y un total de seis columnas: id, temperatura, humedad_aire, huemedad_suelo, humedad_por y fecha_registro.

The image shows a three-step process for creating a table:

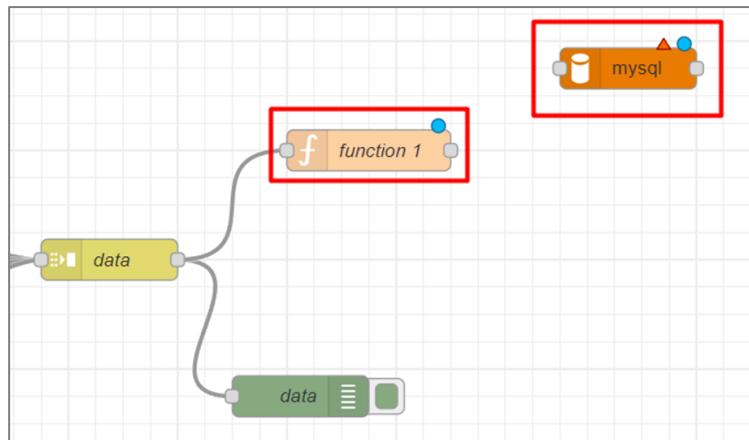
- Step 1: 'Crear nueva tabla' (Create new table) screen.** It has two main fields: 'Nombre de la tabla' (Table name) containing 'registro_sensores' and 'Número de columnas' (Number of columns) set to '6'. Both fields are highlighted with a red box. A 'Crear' (Create) button is to the right.
- Step 2: 'Estructura' (Structure) screen for the 'registro_sensores' table.** This screen shows the table structure with six columns. The columns are: 'id' (Type: INT), 'temperatura' (Type: DOUBLE), 'humedad_aire' (Type: DOUBLE), 'humedad_suelo' (Type: DOUBLE), 'humedad_suelo_por' (Type: DOUBLE), and 'fecha_registro' (Type: DATETIME). The 'id' column is marked as 'PRIMARY' with a checked checkbox in the 'A_I' (Auto Increment) column. A red box highlights the 'id' column and the 'PRIMARY' checkbox.
- Step 3: Confirmation and save screen.** It features two buttons: 'Previsualizar SQL' (Preview SQL) and 'Guardar' (Save), with a red arrow pointing to the 'Guardar' button.

Luego se podrá ver en la parte izquierda superior la base de datos creada con la tabla correspondiente.

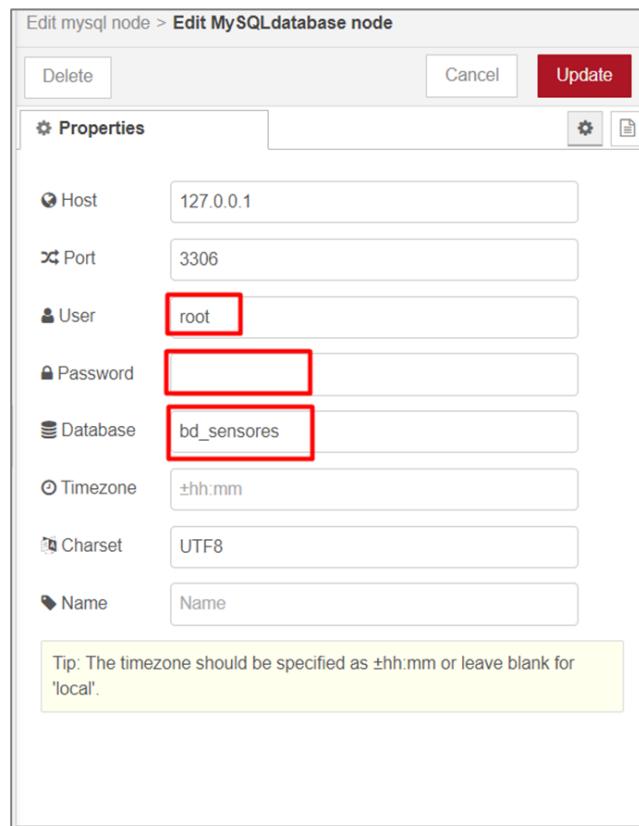


4.6.7. Configuración de Node-RED para la base de datos

Luego en el NODO-RED insertamos un Nodo “function” en la cual escribiremos un bloque de código para registrar los datos y Nodo “mysql” para conectarlo con la base de datos.



Luego configuramos el Nodo “mysql” con los datos correspondientes como se muestra en la figura posterior: el usuario, la contraseña, en este caso como no hay se cuenta y se le deja en vacío, y el nombre de la base de datos. Luego damos en “Update”.



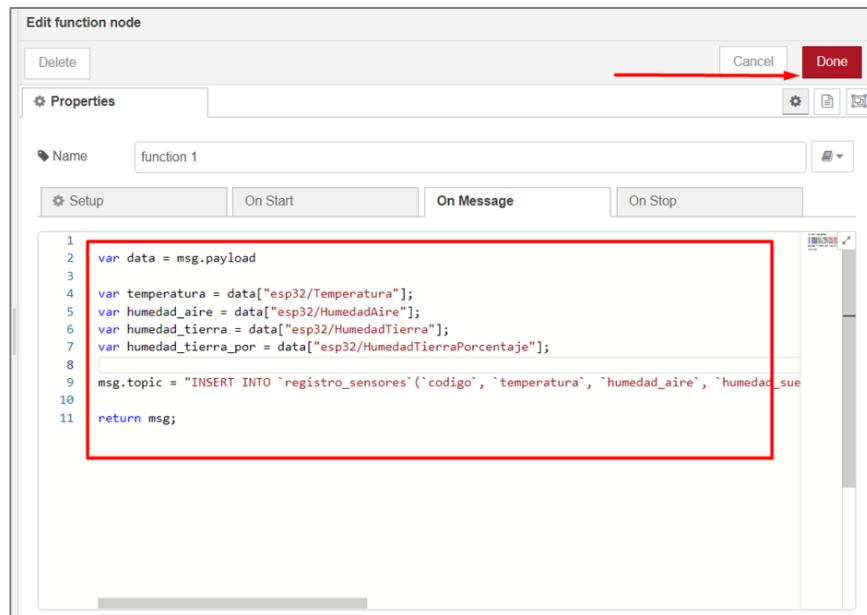
De manera simultánea configuramos el Nodo “function” con los comandos correspondientes para obtener los datos de los Nodos de entrada y para registrar los datos en la base de datos en la tabla “registro_sensores”

Bloque de código:

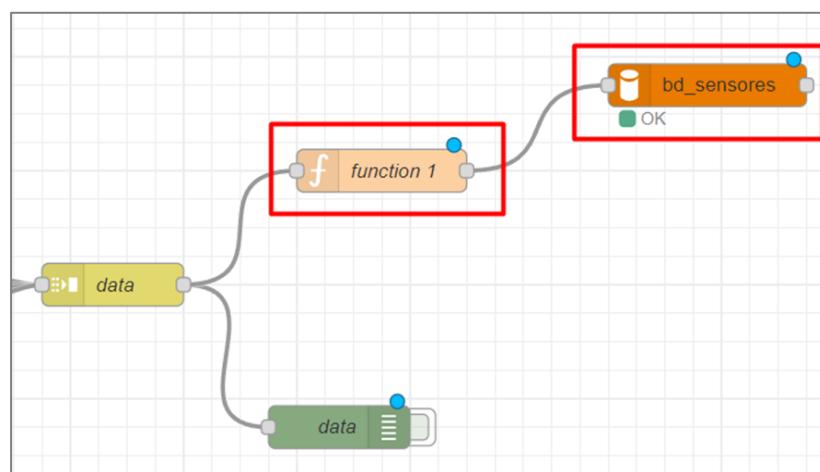
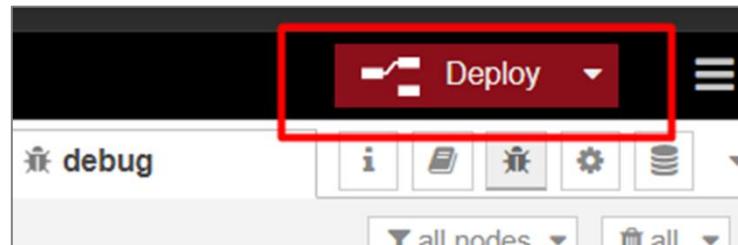
```
//Variable para obtener los datos de los nodos de entrada mqtt
var data = msg.payload

//Variables para obtener los datos de las variables
var temperatura = data["esp32/Temperatura"];
var humedad_aire = data["esp32/HumedadAire"];
var humedad_tierra = data["esp32/HumedadTierra"];
var humedad_tierra_por = data["esp32/HumedadTierraPorcentaje"];
```

Damos clic en “Done” para realizar la configuración.

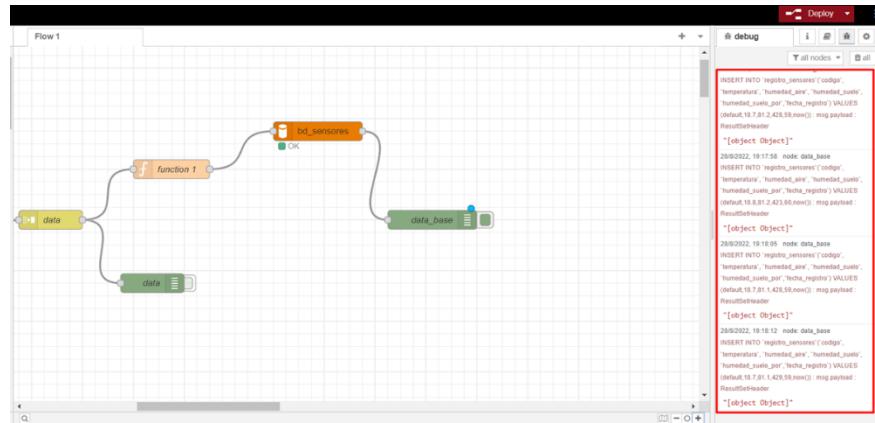


Luego de relacionar los Nodos “function” y “mysql” agregados debemos verificar las correctas configuraciones de estos damos clic en “Deploy” para ejecutar el NODE-RED.

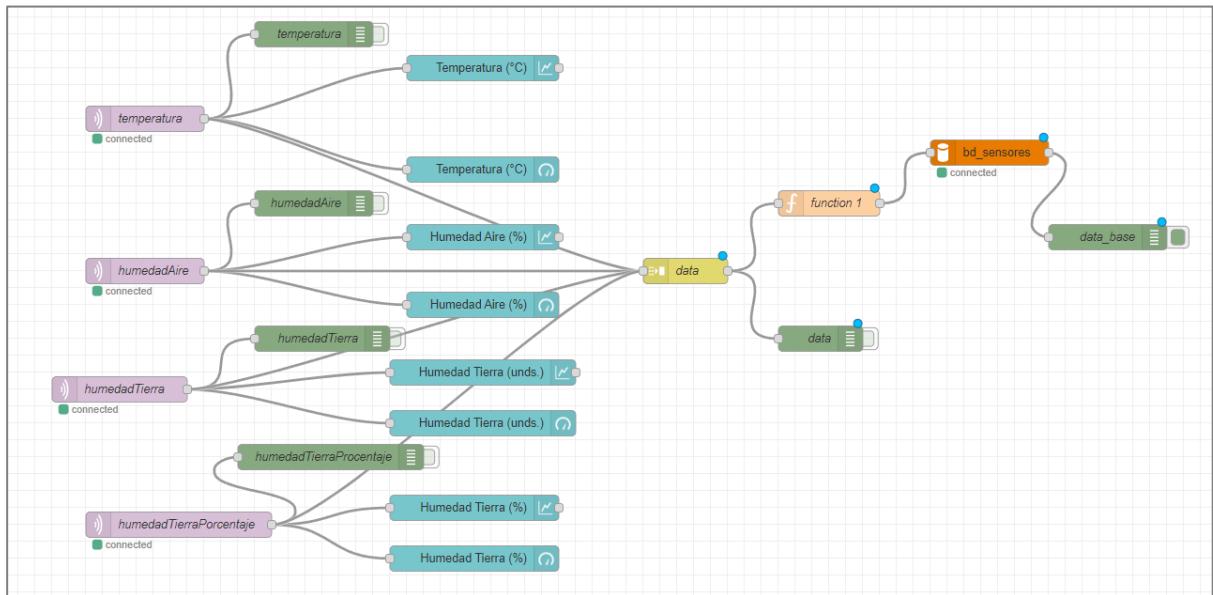


4.6.8. Verificar la configuración

Para verificar que está correcto agregamos un Nodo “debug” que denominaremos “data_base” y los datos que muestra en consola nos indica que el proceso es correcto.



4.7. Flujo completo del Node-RED



5. Resultados

El sistema de riego automático es útil para cualquier persona o empresa que tenga un jardín de plantas o un espacio de cultivo en el cual desea automatizar el proceso de riego debido a la facilidad de uso e implementación es posible la implementación a nivel escalable por ahora solo nos permite abarcar el área una mediana maceta para el riego automatizado. Se observa que la maqueta es intuitiva y reconocible como se puede ver en las imágenes de la maqueta general. Los gráficos estadísticos como gráfico lineal de Node-RED también ayudan a que el

sistema sea entendible para el usuario final, se logra gracias a la comunicación WiFi con los componentes. El Node-RED también ayudó a generar un conjunto de datos, estos datos se almacenarán en la base de datos para posteriormente exportarlo en un archivo con extensión “csv” para que sea pueda ser usado para investigación de análisis de datos para los investigadores externos, toda esta parte se observa en el apartado de Node-RED. Finalmente los objetivos de desarrollo sostenible como es el ahorro de agua se cumple en el apartado uno y dos donde se observa cómo se riega la planta cuando la humedad del suelo esté seca y se regara solo en zonas localizadas, evitando así el desperdicio de agua.

5.1. Impresión de datos en Arduino IDE

En esta sección se mostrarán los datos que se imprimen en el serial de Arduino conforme van variando los parámetros ambientales y del suelo.

5.1.1. Nivel del agua

```
Distancia: 18.73cm
Temperatura: 21.50 Celsius
Humedad Aire: 75.70 %
Humedad Suelo : 465.000 u
Humedad Suelo en % : 56.00 %

=====
Distancia: 19.18cm
Temperatura: 21.30 Celsius
Humedad Aire: 75.20 %
Humedad Suelo : 465.000 u
Humedad Suelo en % : 56.00 %
```

De esta imagen, analizaremos la distancia que nos marca, vemos que es mayor a 15.5 el cual es nuestro umbral con el cual si el sensor de distancia detecta el agua mayor a esta distancia nos indica que el nivel de agua es bajo con lo cual no se puede permitir el riego.

5.1.2. Humedad del suelo

```
Distancia: 12.22cm
Temperatura: 19.00 Celsius
Humedad Aire: 79.20 %
Humedad Suelo : 265.000 u
Humedad Suelo en % : 75.00 %
```

De esta imagen, analizaremos la humedad del suelo cuando este el alto como vemos tenemos dos valores para este parámetro, el primero nos indica el valor según nos da el sensor y el segundo es el valor en porcentaje. En este caso la humedad es alta 265.0 (75%).

Del mismo modo que la anterior imagen, se analizará la humedad del suelo. En este caso la humedad nos marca 1023.0 (1%), es decir que en ese instante el suelo está muy seco por lo que se tiene que comenzar a regar la planta.

```
Distancia: 12.13cm
Temperatura: 19.90 Celsius
Humedad Aire: 78.70 %
Humedad Suelo : 1023.000 u
Humedad Suelo en % : 1.00 %
=====
```

5.1.3. Humedad del ambiente

```
=====
Distancia: 12.13cm
Temperatura: 19.80 Celsius
Humedad Aire: 78.70 %
Humedad Suelo : 468.000 u
Humedad Suelo en % : 55.00 %
=====
```

De esta imagen, analizaremos la humedad del aire para ello tenemos un intervalos de 50% a 75% que nos indica la humedad óptima para las plantas, como vemos en este caso, nos marca 78.7% lo cual significa que la planta se encuentra en peligro debido a esta excesiva humedad.

5.2. Escenarios de uso del sistema de riego

5.2.1. Nivel de agua

Este parámetro nos indica si el nivel del agua es suficiente o no para que haya una disponibilidad de riego.

- Nivel de agua bajo: Como podemos observar en la siguiente imagen, el led rojo se enciende lo cual nos indica que el nivel del agua es bajo. Adicionalmente, se imprime en el LCD que no hay agua y que es necesario llenar agua para que se pueda continuar con el funcionamiento normal.



- Nivel de agua óptimo: Como podemos observar en la siguiente imagen, mientras el nivel del agua se mantenga dentro del límite, la luz del led rojo se mantendrá apagada y el LCD seguirá imprimiendo lo correspondiente.



5.2.2. Humedad del suelo

Como hemos explicado anteriormente la humedad del suelo se separa en tres niveles: seco, húmedo, mojado.

- Estado seco y hay un nivel de agua óptimo: Se comienza con el riego de la planta. Adicionalmente como podemos observar en la siguiente imagen, la humedad de suelo marca 1023.0 (1%), lo que nos indica un nivel de humedad muy seco, por lo que se tiene que regar la planta y adicionalmente la luz del LED RGB cambia al color azul que indica que la planta está siendo regada.



- Estado húmedo: se tiene que evaluar la temperatura del ambiente, si la temperatura es mayor a 23° se riega, en caso contrario no se procede con el riego. Por ejemplo, tenemos dos imágenes la primera nos indica la humedad del suelo la cual marca 481.0 (54%), es decir se encuentra húmeda por lo que se tiene que evaluar la temperatura, para ello evaluamos la segunda imagen en donde nos marca 21.8 °, cómo esta es menor a 23°C no se tiene que regar; por ello, se mantiene el LED RGB en verde.



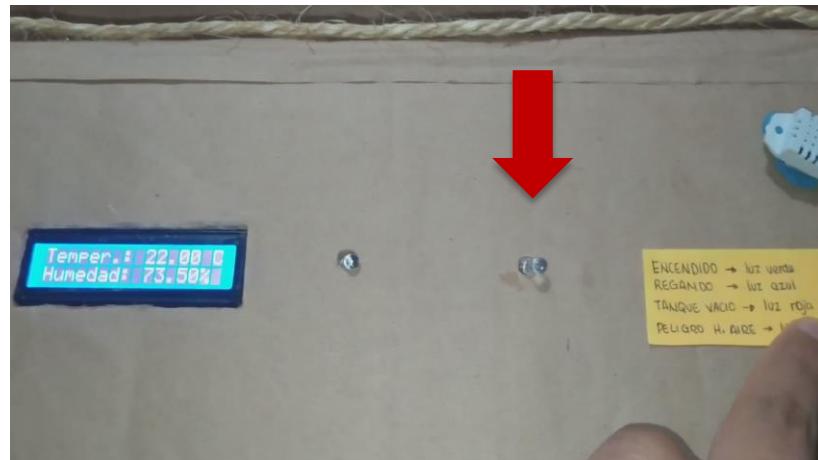
- Estado mojado: No se riega por ningún motivo. Por ejemplo, se observa en la siguiente imagen, la humedad de suelo marca 246.0 (77%), lo que nos indica que la tierra está mojada por ende, el LED RGB se mantiene encendida de color verde y el LCD sigue informando la temperatura y humedad del ambiente, además de la humedad del suelo correspondiente.



5.2.3. Humedad del ambiente

Cuando la humedad del ambiente se encuentra dentro del intervalo 50%-75% es considerado un nivel óptimo para que una planta se mantenga fuera de algún riesgo, es por ello que en este caso se tiene 2 maneras de interpretar, cuando la humedad del ambiente está fuera o dentro del intervalo.

- Dentro del rango: Como podemos observar en la siguiente imagen la humedad marca 73.5% eso nos indica que está dentro del rango óptimo por lo que el LED multicolor se mantiene apagado.



- Fuera del rango: Cuando la humedad del ambiente está fuera del rango, se informa del peligro en la pantalla del LCD con el mensaje "Planta en peligro, mucha humedad", adicionalmente, se enciende la luz del LED multicolor, para dar más alerta a los usuarios, esto se puede observar en la siguiente imagen.



Aquí podremos darnos una idea general de cómo es el sistema de riego automático, en donde se puede apreciar los LED RGB, y el motorcito de agua sumergible en el reservorio de plástico.

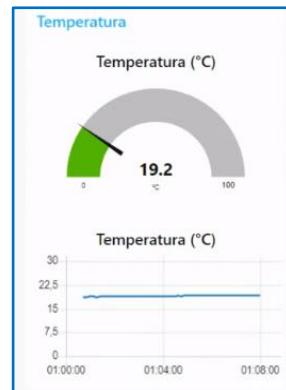


5.3. Gráficos en Node-RED

Los gráficos que obtuvimos de la aplicación fueron divididos en 4 partes, la primera trata sobre las mediciones de la temperatura, la segunda es sobre la humedad del aire, la tercera es sobre la humedad de la tierra y finalmente la última es sobre la humedad de la tierra en porcentaje.

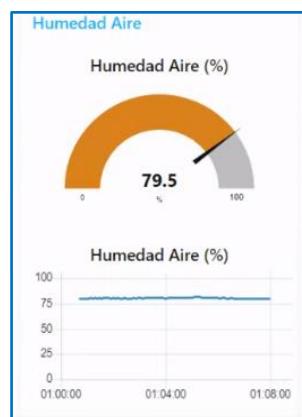
5.3.1. Temperatura

La primera gráfica de la temperatura nos indica que en la hora actual tiene una temperatura de 19.2 °C. La segunda gráfica nos indica que los valores de temperatura no sobrepasan el valor de 22.5 °C y además no baja del valor 15 °C. Los valores expuestos anteriormente nos indican que el ambiente está en condiciones normales.



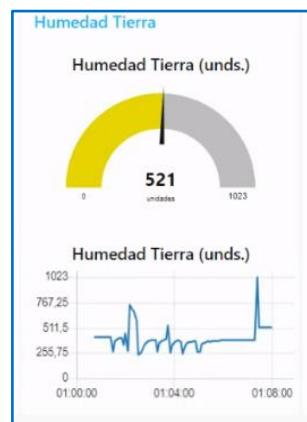
5.3.2. Humedad Aire

La primera gráfica de la humedad del aire nos indica que en la hora actual tiene una humedad relativa de 79.5%. La segunda gráfica nos indica que los valores de humedad no se acercan al valor de 100% y además no baja del valor 75%. Los valores expuestos anteriormente nos indican que el ambiente se encuentra en una humedad considerable para la planta debido a que la humedad óptima es entre 50-75%.



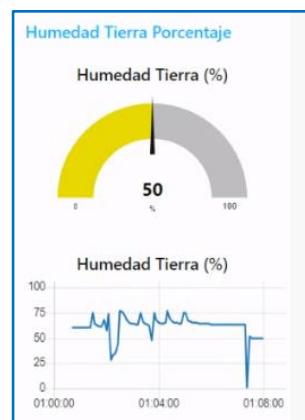
5.3.3. Humedad de Tierra

La primera gráfica de la humedad de la tierra nos indica que en la hora actual tiene una humedad de 521 que indica una tierra húmeda. La segunda gráfica nos indica que el valor mínimo alcanzado es de 255.75 y tiene un valor máximo alcanzado de 1023, los valores cercanos a 255.75 indican que la planta está muy húmeda mientras que el valor de 1023 indica que la planta está muy seca.

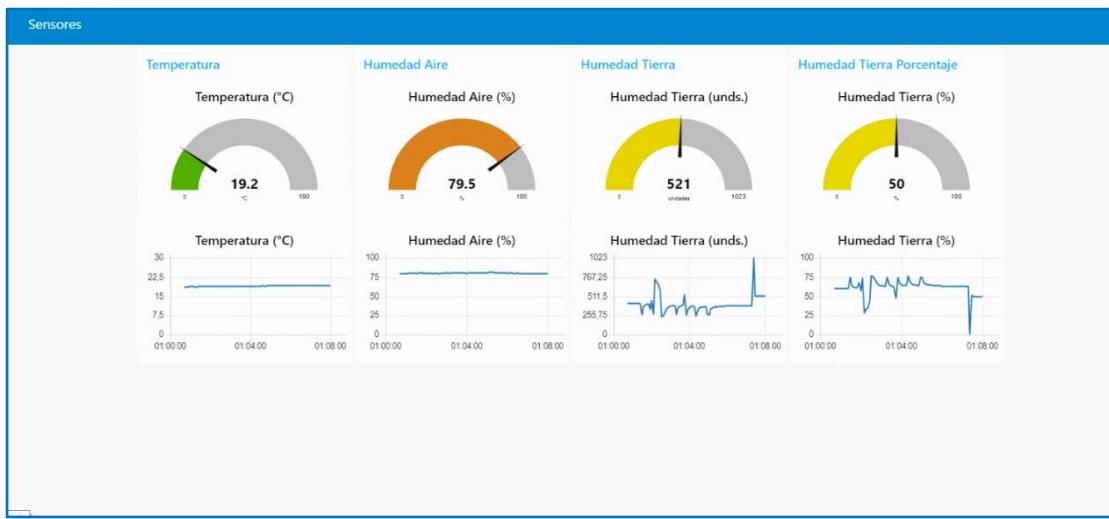


5.3.4. Humedad de Tierra en porcentaje

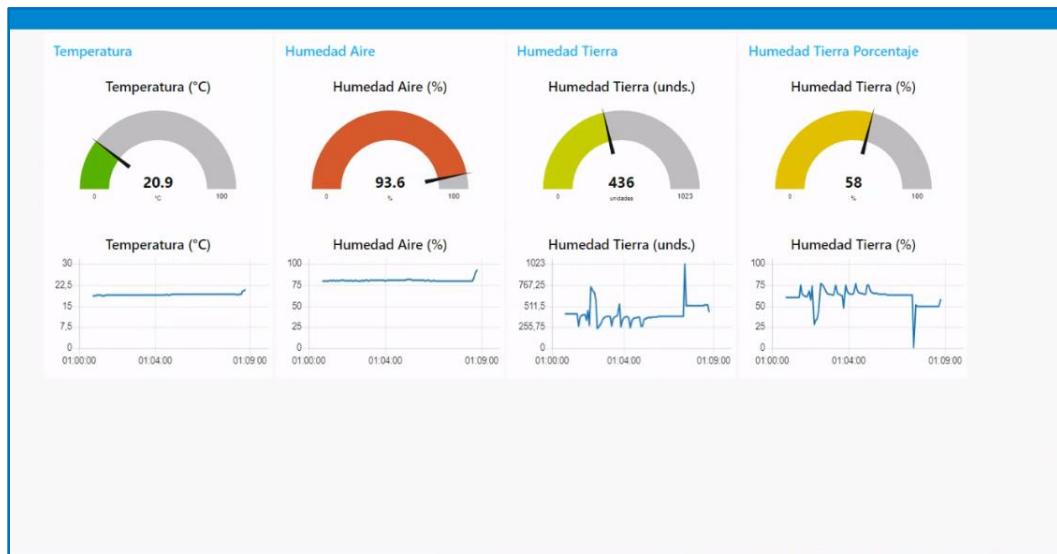
La gráfica de porcentaje expresa la humedad de la tierra en porcentaje donde 0 significa muy seco y 100 indica muy húmedo. En la hora actual se muestra una humedad del 50% lo cual indicaría que la planta aún tiene agua suficiente y no sería necesario un riego. Además se puede observar que mientras en la gráfica anterior salía 1023, en esta gráfica sale 0%, comprobando así los valores obtenidos.



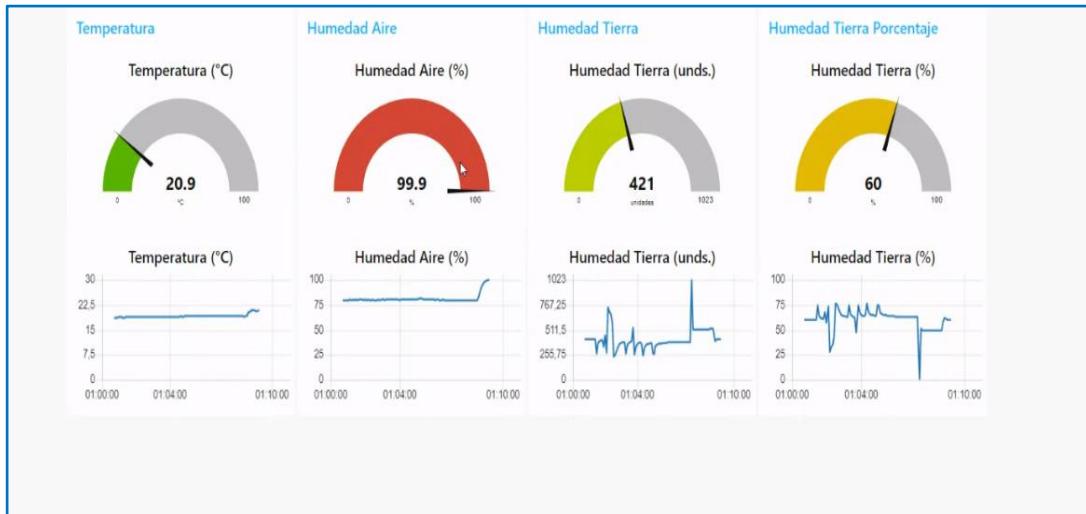
Un panorama general de los 4 gráficos en estado óptimo se muestra a continuación:



Cuando los sensores van subiendo de valores, también se modifica en los dashboard. Para comprobarlo, aumentaremos la humedad del aire y la temperatura poniendo aire caliente al costado del sensor de humedad. En la primera gráfica se observa que la humedad del aire pasó de 79.5% a 93.6 %, y que la temperatura pasó de 19.2°C a 20.9°C.



Ahora se mostrará un caso donde el sensor de humedad del aire detecta valores muy altos como 99%. Se observa que la gráfica se obtiene un pico, cuando la gráfica marca 99 % de humedad.



Todos estos valores se guardan en la base de datos mediante las variables que se muestran en la imagen derecha, se puede comprobar también que los valores son similares a los de la gráfica del Node-RED. El proceso de comprobación para insertar los datos se muestran en el gráfico de la izquierda donde se observa la sentencia de insertar datos en el la base de datos creada mediante MySQL.

```

28/8/2022, 1:11:28 node: debug 5
INSERT INTO `registro_sensores`(`codigo`, `temperatura`, `humedad_aire`, `humedad_suelo`, `humedad_suelo_por`, `fecha_registro`) VALUES
(default,19.4,86,421,60,now()) : msg.payload :
ResultSetHeader

```

```

28/8/2022, 1:10:36 node: data
esp32/HumedadAire : msg.payload: Object
  * { esp32/Temperatura: 19.6,
    esp32/HumedadTierra: 425,
    esp32/HumedadTierraPorcentaje: 59,
    esp32/HumedadAire: 92.8 }

28/8/2022, 1:10:36 node: data
msg : string(22)

```

```

"Database not connected"
28/8/2022, 1:10:41 node: data
esp32/HumedadAire : msg.payload: Object
  * { esp32/Temperatura: 19.6,
    esp32/HumedadTierra: 429,
    esp32/HumedadTierraPorcentaje: 59,
    esp32/HumedadAire: 92.1 }

```

Finalmente, el resultado guardado en la base de datos quedaría como la siguiente imagen.

		codigo	temperatura	humedad_aire	humedad_suelo	humedad_suelo_por	fecha_registro
<input type="checkbox"/>	Editar Copiar Borrar	2291	18.2	90.8	432	59	2022-08-28 15:31:32
<input type="checkbox"/>	Editar Copiar Borrar	2290	18.2	80.6	431	59	2022-08-28 15:31:26
<input type="checkbox"/>	Editar Copiar Borrar	2289	18.2	90.7	427	59	2022-08-28 15:31:20
<input type="checkbox"/>	Editar Copiar Borrar	2288	19.1	81	359	66	2022-08-28 01:59:40
<input type="checkbox"/>	Editar Copiar Borrar	2287	18.9	81	358	66	2022-08-28 01:59:34
<input type="checkbox"/>	Editar Copiar Borrar	2286	18.9	80.8	361	66	2022-08-28 01:59:29
<input type="checkbox"/>	Editar Copiar Borrar	2285	18.8	80.9	360	66	2022-08-28 01:59:24
<input type="checkbox"/>	Editar Copiar Borrar	2284	19	80.9	361	66	2022-08-28 01:59:19
<input type="checkbox"/>	Editar Copiar Borrar	2283	18.9	81.2	359	66	2022-08-28 01:59:13
<input type="checkbox"/>	Editar Copiar Borrar	2282	18.9	81	360	66	2022-08-28 01:59:08
<input type="checkbox"/>	Editar Copiar Borrar	2281	18.9	81.1	360	66	2022-08-28 01:59:03
<input type="checkbox"/>	Editar Copiar Borrar	2280	18.9	81.2	357	66	2022-08-28 01:58:58
<input type="checkbox"/>	Editar Copiar Borrar	2279	18.9	81.4	364	65	2022-08-28 01:58:53
<input type="checkbox"/>	Editar Copiar Borrar	2278	18.9	81.3	366	65	2022-08-28 01:58:47

Los datos exportados en el archivo con extensión “csv” se verían así:

```
codigo,"temperatura","humedad_aire","humedad_suelo","humedad_suelo_por","fecha_registro"
1,"18.8","81.9","457","56","2022-08-25 00:35:25"
2,"18.9","82","476","55","2022-08-25 00:35:32"
3,"18.8","82.1","473","55","2022-08-25 00:35:38"
4,"18.8","81.9","479","54","2022-08-25 00:35:44"
5,"18.8","82.1","479","54","2022-08-25 00:35:50"
6,"18.8","82","483","54","2022-08-25 00:35:57"
7,"18.8","82","484","54","2022-08-25 00:36:03"
8,"18.8","82","485","54","2022-08-25 00:36:09"
9,"18.8","82.1","480","54","2022-08-25 00:36:15"
10,"18.8","81.8","489","53","2022-08-25 00:36:21"
11,"18.9","81.9","489","53","2022-08-25 00:36:28"
12,"18.8","81.7","492","53","2022-08-25 00:36:59"
13,"18.8","81.7","485","54","2022-08-25 00:36:59"
14,"18.9","81.8","492","53","2022-08-25 00:36:59"
15,"18.9","82.1","493","53","2022-08-25 00:37:20"
16,"18.8","81.9","489","53","2022-08-25 00:37:26"
17,"18.7","82.1","498","52","2022-08-25 00:37:32"
18,"18.8","82.1","498","52","2022-08-25 00:37:39"
19,"18.7","81.9","500","52","2022-08-25 00:37:45"
20,"18.8","81.8","493","53","2022-08-25 00:37:51"
```

6. Conclusiones

El sistema de riego automático logró lo propuesto por los objetivos, este ayudará a contribuir con los Objetivos de Desarrollo Sostenible (ODS), Consumo y producción sostenibles. Nuestro prototipo logró utilizar el riego automático para regar los cultivos solo cuando sea necesario, es decir, cuando los sensores utilizados nos lo indiquen y así cumplir con el objetivo número 15 que se basa en apoyar la seguridad alimentaria y del consumo del agua a nivel mundial y con el objetivo 6 que se basa en asegurar el agua asequible y potable a nivel mundial. Se creó el sistema de riego completamente automático y se evitará el uso de mano de obra excesiva, también se logró Implementar una comunicación inalámbrica entre los dispositivos y un sistema de recepción de datos mediante WiFi y poder así hacer la generación de la base de datos, todo esto permite que nuestro sistema sea usable por diferentes personas que tengan conocimiento mínimo de base de datos y programación. La generación del desarrollo de dashboard y gráficas de información para el usuario final se logró implementar y esto permitirá de manera general a que todos los usuarios sin conocer mucho de sistemas, puedan entender los gráficos e ideas.

7. Bibliografía

Ramírez, D. Rodríguez, E. (2016). Diseño de un método para identificar necesidades y oportunidades para la implementación de Internet de las cosas (IoT) aplicable a oficinas de trabajo donde permanezcan entre 30 y 70 personas y planteamiento de un caso práctico de solución en las oficinas de la Agencia Nacional del Espectro [Tesis de Bachiller, Universidad Distrital Francisco José de Caldas]. Repositorio Institucional - Universidad Distrital Francisco José de Caldas.

Software de Arduino. (2022, 8 julio). Arduino.cl - Compra tu Arduino en Línea. Recuperado 24 de agosto de 2022, de <https://arduino.cl/programacion/>

Node-RED. (2022, 14 julio). Node-RED Programación de código bajo aplicaciones basadas en eventos. Recuperado 24 de agosto de 2022, de <https://nodered.org>

Red Hat. (2022, 10 mayo). La Automatización. Recuperado 25 de agosto de 2022, de <https://www.redhat.com/es/topics/automation>

Maher. (2022). Sistema de Riego Automático: en qué consiste y cuáles son sus ventajas. Recuperado 25 de agosto de 2022, de <https://www.maherelectronica.com/sistema-riego-automatico/>

Profesional review. (2017, 18 noviembre). ¿Cuáles son los principales protocolos Wifi? Todo lo que necesitas saber. Recuperado 23 de agosto de 2022, de <https://www.profesionalreview.com/2017/11/18/cuales-principales-protocolos-wifi>

Luis Llamas. (2019, 17 abril). ¿Qué es MQTT? Su importancia como protocolo IOT. Recuperado 25 de agosto, de <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot>

ComputerWeekly.es. (2022). MYSQL. Recuperado 25 de agosto de 2022, de <https://www.computerweekly.com/es/definicion/MySQL>

Guijarro, A., Cevallos, J., Preciado, K., & Zambrano, N. (2018). Sistema de riego automatizado con arduino. *Revista Espacios*, 39(37), 27. <https://www.revistaespacios.com/a18v39n37/18393727.html>

Cañon, A. & Cifuentes, E.(2018). *Prototipo de un sistema automatizado de riego para jardines* [Tesis de grado, Los Libertadores]. Repositorio intitucional Los Libertadores
<http://hdl.handle.net/11371/1533>

Castillo, C. (2021). *Diseño de un sistema de riego automatizado para cultivos de ciclo corto con arduino. Estudio de caso pimiento* [Tesis de grado, Universidad Estatal Peninsula de Santa Elena]. Repositorio Universidad Estatal Peninsula de Santa Elena
<https://repositorio.upse.edu.ec/handle/46000/6306>

¿Cómo funciona el sensor de ultrasonidos medidor de distancia? (2021, 12 octubre). Solectro. Recuperado 24 de agosto de 2022, de <https://solectroshop.com/es/blog/como-funciona-el-sensor-de-ultrasonidos-medidor-de-distancia--n99>

Reyes, S.J., 2009. Manual práctico de Conservación y Restauración de Cactáceas y otras Plantas Suculentas. Comisión Nacional Forestal. Secretaría de Medio Ambiente y Recursos Naturales. México, 108 pp.

Reyes, S. J., Islas-Luna, M. A., & González-Zorzano, O. (2014). Guía práctica de propagación y cultivo de las especies del género Echeveria. Editorial Universidad Nacional Autónoma de México: Mexico City, Mexico, 111.

Anexo N° 1

Código Implementado

```
/**INCLUIR LIBRERIAS*/
#include <WiFi.h> //Libreria para el WiFi
#include <PubSubClient.h> //Libreria para la comunicacion MQTT
#include "DHTesp.h" //libreria del sensor DHT22
#include <LiquidCrystal_I2C.h> //Libreria para usar LCD I2C
/**PINES*/
//Sensores
int pindht22 = 16; //pin G16 conectado al sensor DHT22
int pinsensorHS = 35; //pin G35 conectado al sensor de humedad de suelo
(FC-28)
//Rele
int rele = 15; //pin G15 conectado al rele
//LCD I2C
int pinSDA = 21; //pin G21 conectado al sistema de data del LCD I2C
int pinSCL = 22; //pin G22 conectado al sistema de reloj del LCD I2C

//Sensor de distancia HC-SR04
int trigPin = 12; //pin G12 conectado al trigger del sensor
int echoPin = 14; //pin G14 conectado al echo del sensor

//LED RGB
int pinRed = 17; //pin G17 conectado al anodo rojo
int pinGreen = 18; //pin G18 conectado al anodo verde
int pinBlue = 5; //pin G5 conectado al anodo azul

//LED Rojo
int ledrojo = 13; //pin G13 conectado al LED rojo

//LED multicolor
int ledMulti = 33; //pin G33 conectado al LED multicolor

/**VARIABLES GLOBALES*/
/* Crea un objeto de la clase LiquidCrystal_I2C
   con dirección, columnas y filas indicadas.*/
LiquidCrystal_I2C lcd(0x27, 16, 2);
DHTesp dhtSensor; //variable que representa al sensor DHT22
WiFiClient esp32Client; //Instancia de cliente para WiFi
PubSubClient mqttClient(esp32Client); //Instancia de cliente para MQTT

/**DATOS DE LA RED WIFI*/
const char* ssid = "HUAWEI-2.4G-YpsU"; //Nombre del WiFi
const char* password = "5F32kT8q"; //Contrasenia del WiFi
```

```

/***DATOS DEL SERVIDOR BROKER MQTT***/
char *server = "192.168.18.60"; //IP del servidor
int port = 1883; //Puerto para la conexion
String resultS = ""; //Variable que representa el mensaje

/********************************************/


/***PROTOTIPOS DE FUNCIONES***/
void wifiInit();
void callback(char* topic, byte* payload, unsigned int length);
void reconnect();
long readUltrasonicDistance(int triggerPin, int echoPin);
double sensorDistancia();
void sensorSuelo(double HS, double pcHS);
void sensorHumedadTemperatura(double humedad, double temp);
void linea();
void imprimirLCD(double humedad, double temperatura, double hs, double pcHS);
bool nivelAgua();
int tierraSeca(double hs);
void MQQT_conexion(double h, double t, double hs, double p_hs);

/********************************************/


/***SETUP***/
//Inicializar los sensores y configuraciones
void setup()
{
    Serial.begin(9600); //Inicializa la consola

    wifiInit(); //Se inicializa el WiFi
    //Establece los detalles del servidor: IP del servidor y puerto de
    conexion
    mqttClient.setServer(server, port);
    //Establece el procedimiento para la devolucion de llamada de mensaje
    mqttClient.setCallback(callback);

    //Inicializar los pines
    pinMode(rele, OUTPUT); //pin G15 como salida
    pinMode(pinsensorHS, INPUT); //pin G35 como entrada
    pinMode(pinRed, OUTPUT); //pin G17 como salida
    pinMode(pinGreen, OUTPUT); //pin G18 como salida
    pinMode(pinBlue, OUTPUT); //pin G5 como salida
    pinMode(ledrojo, OUTPUT); //pin G13 como salida
    pinMode(ledMulti, OUTPUT); //pin G33 como salida

    //Inicializar el sensor DHT22 especificando el pin G16 y el tipo de
    sensor
    dhtSensor.setup(pindht22, DHTesp::DHT22);
}

```

```

/*Funciones para configurar el LCD I2C*/
lcd.init(); //Configura e inicializa el I2C y el LCD.
lcd.backlight(); //Enciende la luz del fondo del LCD
}
/*****************************************/
/**LOOP**/
//Ejecuta el codigo repetitivamente
void loop()
{

    TempAndHumidity data = dhtSensor.getTempAndHumidity(); //Captura
valores del sensor DHT22
    double sensorHS = analogRead(pinsensorHS); //Captura valor del sensor de
humedad del suelo (sensorHS)
    double HS = map(sensorHS, 0, 4096, 0, 1023); //Mapea el valor de
sensorHS en un rango de 0 a 1023
    double pcHS = map(sensorHS, 0, 4096, 100, 0); //Mapea el valor de
sensorHS en un rango de 100% a 0%
    int humedadTierra = tierraSeca(HS); //Asigna el estado del agua
(seca/humeda/mojada)

    MQTTConexion(data.humidity, data.temperature, HS, pcHS); //Conecta a
MQTT
    imprimirLCD(data.humidity, data.temperature, HS, pcHS); //Imprime
valores en el LCD

    if (nivelAgua()) { //Si hay agua suficiente
        if (humedadTierra == 3) { //Si la tierra esta seca
            //Encender el color azul del LED RGB
            digitalWrite(pinRed, 0);
            digitalWrite(pinGreen, 0);
            digitalWrite(pinBlue, 255);
            //Apagar el LED rojo
            digitalWrite(ledrojo, LOW);
            digitalWrite(rele, LOW); // Enviar un valor bajo al rele y enciende
la bomba de agua
            lcd.clear(); //Limpia el LCD
            lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones
(columna:0, fila:0)
            lcd.print("Regando..."); //Imprime la frase en el LCD
        }
        else { //Sino, no esta seca
            if (humedadTierra == 2) { //Si la tierra esta humeda
                if (data.temperature > 23) { //Si la temperatura del ambiente es
mayor a 23°C
                    //Encender el color azul del LED RGB
                    digitalWrite(pinRed, 0);
                    digitalWrite(pinGreen, 0);

```

```

        digitalWrite(pinBlue, 255);
        //Apagar el LED rojo
        digitalWrite(ledrojo, LOW);
        digitalWrite(rele, LOW); // Enviar un valor bajo al rele y
enciende la bomba de agua
        lcd.clear(); //Limpia el LCD
        lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones
(columna:0, fila:0)
        lcd.print("Regando..."); //Imprime la frase en el LCD
    }
    else { //Si no, la temperatura del ambiente es menor que 23°C
        //Encender el color verde del LED RGB
        digitalWrite(pinRed, 0);
        digitalWrite(pinGreen, 255);
        digitalWrite(pinBlue, 0);
        //Apagar el LED rojo
        digitalWrite(ledrojo, LOW);
        digitalWrite(rele, HIGH); //Envia un valor alto al rele y apaga
la bomba de agua
    }
}
else { //Si no, la tierra esta mojada
    //Encender el color verde del LED RGB
    digitalWrite(pinRed, 0);
    digitalWrite(pinGreen, 255);
    digitalWrite(pinBlue, 0);
    //Apagar el LED rojo
    digitalWrite(ledrojo, LOW);
    digitalWrite(rele, HIGH); //Envia un valor alto al rele y apaga la
bomba de agua
}
}
}
else { //Si no, no hay agua suficiente o el tanque esta vacío
    digitalWrite(ledrojo, HIGH); //Encender el LED rojo
    //Apagar LED RGB
    digitalWrite(pinRed, 0);
    digitalWrite(pinGreen, 0);
    digitalWrite(pinBlue, 0);
    //-----
    digitalWrite(rele, HIGH); //Envia un valor alto al rele y apaga la
bomba de agua
    //Imprimir en el LCD
    lcd.clear(); //Limpia el LCD
    lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones (columna:0,
fila:0)
    lcd.print("No hay agua,"); //Imprime la frase en el LCD
}

```

```

    lcd.setCursor(0, 1); //Se ubica el cursor en las posiciones (columna:0,
fila:1)
    lcd.print("rellene de agua"); //Imprime la frase en el LCD
}

if (data.humidity >= 50 && data.humidity <= 75) { //Si la humedad
detectada por el sensor DHT22 esta entre 50°C y 75°C
    digitalWrite(ledMulti, LOW); //Apagar el LED multicolor
}
else { //Si no cumple la condicion anterior
    delay(2000); //Espera por 2000 milisegundos = 2 segundos
    digitalWrite(ledMulti, HIGH); //Encender el LED multicolor
    if (data.humidity < 50) {
        //Imprimir en el LCD
        lcd.clear(); //Limpia el LCD
        lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones
(columna:0, fila:0)
        lcd.print("Planta en peligro"); //Imprime la frase en el LCD
        lcd.setCursor(0, 1); //Se ubica el cursor en las posiciones
(columna:0, fila:1)
        lcd.print("poca humedad"); //Imprime la frase en el LCD
    }
    else {
        //Imprimir en el LCD
        lcd.clear(); //Limpia el LCD
        lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones
(columna:0, fila:0)
        lcd.print("Planta en peligro"); //Imprime la frase en el LCD
        lcd.setCursor(0, 1); //Se ubica el cursor en las posiciones
(columna:0, fila:1)
        lcd.print("muchas humedad"); //Imprime la frase en el LCD
    }
}
delay(2000); //Espera por 2000 milisegundos = 2 segundos
}
/*****************************************/
/**IMPLEMENTACION DE FUNCIONES**/
//Procedimiento para inicializar la configuracion del WiFi
void wifiInit() {

    Serial.print("Conectándose a "); //Imprimir en consola la cadena de
caracteres enviada
    Serial.println(ssid); //Imprimir el nombre del WiFi en consola

    WiFi.begin(ssid, password); //Inicializa la configuracion de red de la
libreria WiFi tomando en cuenta el nombre y la contrasenia del WiFi

    //Bucle para conectar el WiFi
}

```

```

while (WiFi.status() != WL_CONNECTED) { //Continua hasta que se haya
conectado al WiFi
    Serial.print(".");
    delay(500);
}
//Imprimir en consola
Serial.println("");
Serial.println("Conectado a WiFi");
Serial.println("Dirección IP:");
Serial.println(WiFi.localIP()); //Imprime la direccion IP en consola
}

//Procedimiento que maneja los mensajes recibidos
void callback(char* topic, byte* payload, unsigned int length) {
    //Imprime en consola
    Serial.print("Mensaje recibido [");
    Serial.print(topic);
    Serial.print("] ");

    char payload_string[length + 1];

    memcpy(payload_string, payload, length); //Concatenar cadenas de
caracteres
    payload_string[length] = '\0'; //El ultimo caracter es el final de
cadena
    resultS = "";
    //Bucle para unir los caracteres
    for (int i = 0; i < length; i++) {
        resultS = resultS + (char)payload[i]; //Se concatena los caracteres en
resultS, es decir el mensaje recibido
    }
    Serial.println(); //Hace un salto de linea
}

//Procedimiento para reconectarse al MQTT en caso sea necesario
void reconnect() {
    while (!mqttClient.connected()) { //Verifica que no este conectado
        Serial.println("Intentando conectarse MQTT..."); //Imprime en consola
        String client_id = "esp32-client-"; //ID de cliente de esp32
        /*Se obtiene la direccion MAC del WiFi,
         * se hace un cast para que sea de tipo String
         * y se le concatena a la variable client_id.*/
        client_id += String(WiFi.macAddress());
        //Conectar al cliente usando su respectivo ID
        if (mqttClient.connect(client_id.c_str())) {
            Serial.println("Conectado"); //Imprime en consola
            mqttClient.subscribe("test"); //Suscribir los mensajes publicados a
"test"
    }
}

```

```

    } else {//Si no se conecta
        Serial.print("Fallo, rc=");
        Serial.print(mqttClient.state()); //Imprime el estado actual del
cliente MQTT
        Serial.println(" intentar de nuevo en 5 segundos");
        delay(5000); // Espera por 5000 milisegundos = 5 segundos
    }
}

//Retornar la onda de sonido por el sensor HC-SR04
long readUltrasonicDistance(int triggerPin, int echoPin)
{
    pinMode(triggerPin, OUTPUT); //Configura el pin G12 como salida
    digitalWrite(triggerPin, LOW); //Determina el pin G12 con un valor bajo
    delayMicroseconds(2); //Retrasa el programa por 2 microsegundos
    digitalWrite(triggerPin, HIGH); //Determina el pin G12 con un valor alto
    delayMicroseconds(10); //Retrasa el programa por 10 microsegundos
    digitalWrite(triggerPin, LOW); //Determina el pin G12 con un valor bajo
    pinMode(echoPin, INPUT); //Configura el pin G14 como entrada
    // Lee el pin echo y retorna la onda de sonido en microsegundos
    return pulseIn(echoPin, HIGH);
}

//Retorna la distancia segun el sensor HC-SR04
double sensorDistancia() {
    double cm = 0;
    //Obtiene la distancia en cm usando la onda de sonido del HC-SR04
    cm = 0.01723 * readUltrasonicDistance(trigPin, echoPin);
    Serial.println("Distancia: " + String(cm) + "cm");
    return cm;
}

//Mostrar valores del sensor de humedad del suelo en LCD y el monitor del
IDE
void sensorSuelo(double HS, double pcHS) {
    //LCD
    lcd.setCursor(0, 0); //Se ubica el cursor en las posiciones (columna:0,
fila:0)
    lcd.print("H. Suelo: "); //Imprime un string
    lcd.print(String(HS, 3)); //Convertir HS a string con 3 decimales y lo
imprime
    lcd.print(" u"); //Imprime un string
    lcd.setCursor(0, 1); //Se ubica el cursor en las posiciones (columna:0,
fila:1)
    lcd.print("H.S.(%)"); //Imprime un string
    lcd.print(String(pcHS)); //Convertir pcHS a string y lo imprime
}

```

```

lcd.print("%"); //Imprime un string
//MONITOR SERIAL
Serial.println("Humedad Suelo : " + String(HS, 3) + " u "); //Imprime un
string
Serial.println("Humedad Suelo en % : " + String(pcHS) + " % ");
//Imprime un string
}

//Mostrar valores segun el sensor DHT22
void sensorHumedadTemperatura(double humedad, double temp) {

    //LCD
    lcd.print("Temper.: ");
    lcd.print(String(temp, 2)); //Convierte temp a string con 2 decimales y
lo imprime
    lcd.print(" Celcius");
    lcd.setCursor(0, 1); //Se ubica el cursor en las posiciones (columna:0,
fila:1)
    lcd.print("Humedad: ");
    lcd.print(String(humedad)); //Convertir humedad a string y lo imprime
    lcd.print("%");

    //MONITOR SERIAL
    Serial.println("Temperatura: " + String(temp, 2) + " Celsius ");
    //Imprime valores
    Serial.println("Humedad Aire: " + String(humedad) + " % "); //Imprime
valores
}

//Imprime una linea de dobles rayas en el monitor
void linea() {
    Serial.println("=====");
}

//Imprimir los valores en el LCD de forma general
void imprimirLCD(double humedad, double temperatura, double hs, double
pcHS) {
    lcd.clear(); //Limpiar la pantalla del LCD
    sensorHumedadTemperatura(humedad, temperatura); //Imprime los valores
del DHT22
    delay(1000); //Pausa el programa por 1 segundo

    lcd.clear(); //Limpiar la pantalla del LCD
    sensorSuelo(hs, pcHS); //Imprime el valor del sensor de humedad del
suelo
    delay(1000); //Pausa el programa por 1 segundo

    linea(); //Imprime la linea en el monitor
}

```

```

}

//Determina el nivel del agua en el tanque
bool nivelAgua() {
    //Variables
    double distancia = 0;
    bool agua = false; //La cantidad de agua no es suficiente o ya no hay
    distancia = sensorDistancia(); //Asigna la disntancia por el HC-SR04

    if (distancia < 15.5) { //Si la distancia es menor que 15.5 cm
        agua = true; //La cantidad de agua es suficiente
    }

    return agua; //Retorna true o false
}

//Determina si la tierra esta seca/humeda/mojada
int tierraSeca(double hs) {
    //Se realiza de acuerdo al valor obtenido de forma analogica del sensor
    de humedad del suelo
    int estado;
    if (hs <= 300) {
        estado = 1; // Tierra mojada
    } else {
        if (hs <= 700) {
            estado = 2; // Tierra humeda
        } else {
            estado = 3; // Tierra seca
        }
    }
    return estado; //Retorna un numero entero de acuerdo al estado de la
tierra
}

//Procedimiento de conexion al MQTT
void MQQTConexion(double h, double t, double hs, double p_hs) {
    if (!mqttClient.connected()) { //Verifica si el cliente para MQTT no
esta conectado
        reconnect(); //Reconectar en caso no este conectado
    }

    mqttClient.loop(); //Permite que el cliente procese los mensajes
entrantes y mantenga la conexion al servidor

    /*Sensor DHT22*/
    // Se pasa el valor de la temperatura de DHT22 al servidor MQTT
    mqttClient.publish("esp32/Temperatura", String(t).c_str());
}

```

```
//Se pasa el valor de la humedad del aire de DHT22 al servidor MQTT
mqttClient.publish("esp32/HumedadAire", String(h).c_str());  
  
/*Sensor Humedad de Tierra*/
//Se pasa el valor del sensor de humedad del suelo al servidor MQTT
mqttClient.publish("esp32/HumedadTierra", String(hs).c_str());  
  
//Se pasa el valor de la sensor de humedad del suelo en formato de % al
servidor MQTT
mqttClient.publish("esp32/HumedadTierraPorcentaje",
String(p_hs).c_str());  
  
delay(2000); //Espera por 2000 milisegundos = 2 segundos
}
```

Anexo N° 2

Estructura de la Tabla

REGISTRO_SENSORES	
PK	codigo temperatura humedad_aire humedad_suelo humedad_suelo_por fecha_registro

REGISTRO_SENSORES		
Descripción	Descripción	
Campo	Tipo	Descripción
<u>codigo</u>	INT	Identificador de un registro de la tabla.
temperatura	DOUBLE	Temperatura del ambiente obtenido por el sensor DHT22.
humedad_aire	DOUBLE	Humedad del ambiente obtenido por el sensor DHT22.
humedad_suelo	DOUBLE	Valor de humedad del suelo obtenido de manera analógica por el sensor FC-28.
humedad_suelo_por	DOUBLE	Porcentaje de humedad del suelo obtenido por el sensor FC-28.
fecha_registro	DATETIME	Fecha y hora del momento en que un registro ha sido ingresado.