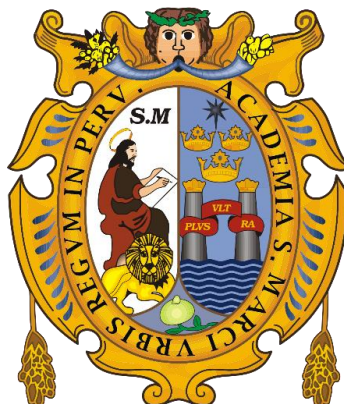


**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**(Universidad del Perú, DECANA DE AMÉRICA)**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**Escuela Profesional De Ingeniería De Sistemas**



**SISTEMA DE RIEGO AUTOMATIZADO - IOT**

**ASIGNATURA**

Internet de las Cosas

**GRUPO 8**

**INTEGRANTES**

Ccolcca Avalos, Mariluz – 19200177

Dominguez Acosta, Prish Antony – 19200054

Linares Castañeda, Bruno Luis - 19200140

Malca Ramirez, Jhonattan David – 19200030

Ramirez Teran, Brithany Antonella - 19200168

Solis Rivera, Angel Jeanpierre - 19200194

**DOCENTE**

Rosas Cuevas, Yessica

**LIMA – PERÚ**  
**2022**

## Cuadro de participación

<b>Integrantes</b>	<b>Evaluación</b>
Ccolcca Avalos, Mariluz	<b>100%</b>
Dominguez Acosta, Prish Antony	<b>100%</b>
Linares Castañeda, Bruno Luis	<b>100%</b>
Malca Ramirez, Jhonattan David	<b>100%</b>
Ramirez Teran, Brithany Antonella	<b>100%</b>
Solis Rivera, Angel Jeanpierre	<b>100%</b>

## ÍNDICE

1. INTRODUCCIÓN	6
2. REVISIÓN DEL ESTADO DEL ARTE	7
2.1. Diseño e implementación de un sistema de regado con funciones smart	7
2.2. Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante IOT en los cultivos urbanos de la fundación “Mujeres empresarias Marie Poussepin”	7
2.3. Evaluación de la eficiencia del riego en los jardines de la ciudad de Arequipa empleando un sistema automatizado de bajo costo con energía cero y una plataforma IOT.	8
3. PROBLEMÁTICA	9
4. OBJETIVOS	9
4.1. General	9
4.2. Específicos	9
5. MARCO TEÓRICO	10
5.1. Internet de las cosas	10
5.2. Comunicación inalámbrica	10
5.3. Automatización	11
5.4. Sistemas de control automático	11
5.5. NODE-RED	12
5.6. Sistema de riego	13
5.7. Componentes del sistema de riego	13
5.7.1. ESP8266	13
5.7.2. Sensor ultrasonido	14
5.7.3. DHT22	14
5.7.4. Sensor humedad del suelo salida analógica	14
5.7.5. Relé	15
5.7.6. Bomba de agua	15
5.7.7. LED	16
5.7.8. Led RGB	16
5.7.9. Resistencia	17
5.7.10. Protoboard	17
5.7.11. Cables para protoboard	17
6. IMPLEMENTACIÓN DEL SISTEMA	18
6.1. Descripción general	18
6.2. Funcionamiento del sistema	18
6.3. Diagrama del circuito	19

6.4.	Ensamblado del circuito	20
6.5.	Implementación de NODE RED	24
6.6.	Base de datos	28
7.	RESULTADOS	31
8.	CONCLUSIONES	34
9.	BIBLIOGRAFÍA	35
	ANEXOS	36

## ÍNDICE DE FIGURAS

Figura 1. ESP8266	13
Figura 2. Sensor ultrasonido	14
Figura 3. DHT22	14
Figura 4. Sensor humedad del suelo salida analógica	15
Figura 5. Relé	15
Figura 6. Bomba de agua	16
Figura 7. LED	16
Figura 8. Led RGB	16
Figura 9. Resistencia	17
Figura 10. Protoboard	17
Figura 11. Cables para protoboard	18
Figura 12. Diagrama del circuito	19
Figura 13. Implementación de NODE RED	25

## **1. NTRODUCCIÓN**

Actualmente, dentro del tema de manejo de medios de información que incluyen de voz, datos y últimamente el internet, la comunicación inalámbrica es considerada y aceptada como medio de enlace entre diferentes dispositivos como en sistemas domóticos que tienen la particularidad de conectarse vía bluetooth, radio frecuencia y lo más importante para la época, a través de wifi. Precisamente, el uso del wifi implica la movilización sin los molestos cables tradicionales dentro de la distancia permitida, es decir, gracias a su fácil manipulación tuvo éxito en estos últimos años.

Con la comunicación inalámbrica, la interrelación del puerto serie y algunos dispositivos, se pudo crear el proyecto de sistema de riego automatizado utilizando Internet de las Cosas que será necesario para la medición de agua y otros factores en una maceta, que serán recibidos en una base de datos para la presentación y comprobación del correcto funcionamiento.

Este tipo de simulación se realiza a grande escala en muchas chacras del litoral peruano dada su importancia y necesidad en la agricultura junto a la correcta ubicación y proceso donde es necesario un riego controlado y constante, para que el cultivo no se dañe reduciendo la productividad y consumo, en ese sentido, surge la necesidad de automatizar algunos sectores a través del sistema de riego automatizado que controla algunas áreas gracias a los sensores de temperatura, humedad, etc. Esta idea del control de riego utilizando tecnología surge como analogía en el campo de los sistemas domóticos bajo la premisa de “si la domótica se utiliza para el beneficio de las personas dentro del hogar”, esta también puede ayudar a las personas en el campo para permitir no estar tan pendientes de estos para poder realizar tareas adicionales o paralelas al mismo tiempo.

Finalmente, el presente informe presenta una introducción para darle contexto al uso del sistema automatizado, una justificación en el marco de la agricultura, problemática actual, objetivos, marco teórico de los conceptos aplicados y dispositivos usados, el desarrollo del sistema en Node Red y las conexiones adecuadas junto a gráficos y conclusiones.

## **2. REVISIÓN DEL ESTADO DEL ARTE**

### **2.1. Diseño e implementación de un sistema de regado con funciones smart**

Roca Roca, Jose Manuel (2020) en la tesis sustentada en la “Universidad Politécnica de Valencia” con el título de “Diseño e implementación de un sistema de regado con funciones Smart” tuvo como objetivo principal crear un sistema de riego que permita un funcionamiento autónomo y hacer un seguimiento de factores y control de estos con tecnología. Respecto al diseño, no es necesario utilizar ninguna metodología de desarrollo ágil, por lo que, simplemente, se ha utilizado un diseño evolutivo, en el que, tras desarrollar cada etapa del proyecto, se testea y se construye sobre ella. Además, se utilizó Git para permitir la sincronización del código entre distintos dispositivos y respecto al hardware se tiene el Raspberry pi, Arduino, sensores, electroválvulas de riego, relés electromagnéticos y dentro del software se tiene a Node-red, Arduino e IFTTT. Su principal resultado respecto a las funcionalidades es la creación de un sistema de riego autónomo con conexión a internet que dentro de sus acciones tiene la de bloquear riego en caso de previsión de lluvia, notificación con la información de los sensores y se obtiene seguridad gracias a Node-Red. Finalmente, se concluye que a lo largo de este trabajo se han detallado una serie de requisitos motivados por la necesidad de crear un sistema de riego con conexión a Internet. Estos requisitos se han visto cubiertos gracias al uso del hardware y software elegidos, pudiendo realizar todas las funciones que eran necesarias.

### **2.2. Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante IOT en los cultivos urbanos de la fundación “Mujeres empresarias Marie Poussepin”**

Vargas Garcia, Marco y Cortes Cadavid, Valeria (2020) en la tesis sustentada en la “Universidad Catolica de Colombia” con el título de Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante IOT en los cultivos urbanos de la fundación “Mujeres empresarias Marie Poussepin” tuvo como objetivo principal desarrollar un sistema de riego automatizado y monitoreo de variables ambientales mediante IOT en los cultivos urbanos ubicados en el barrio Altamira, Bogota. Con el fin de cumplir los objetivos planteados se llevó a cabo un plan metodológico dividido en diferentes fases iniciando con la exploración de información, diseño del sistema de control y automatización, implementación del prototipo, pruebas de funcionamiento y capacitación. Dentro de estas

actividades se usan los sensores de humedad, temperatura, la placa arduino acompañado del software Thingspeak. Su principal resultado es la correcta implementación dentro de los cultivos junto al software y sensores proporcionados para la verificación del sistema de control de riego y monitoreo de variables mediante su comportamiento evaluado en Thingspeak. Finalmente, se concluye que se logra automatizar el riego del cultivo y tener un control de las variables principales que influyen directamente en el desarrollo de la siembra, además, se optimizó el consumo de agua ya que así se garantiza la cantidad necesaria y exacta requerida por el tipo de cultivo.

### **2.3. Evaluación de la eficiencia del riego en los jardines de la ciudad de Arequipa empleando un sistema automatizado de bajo costo con energía cero y una plataforma IOT.**

Esquicha Jose (2022) en la tesis sustentada en la “Universidad Nacional de San Agustín de Arequipa” con el título de Evaluación de la eficiencia del riego en los jardines de la ciudad de Arequipa empleando un sistema automatizado de bajo costo con energía cero y una plataforma IOT. para optar el Grado Académico de Doctor en Ciencias Ambientales y Energías Renovables tuvo como objetivo principal evaluar la eficiencia del riego por goteo en comparación con el riego por inundación donde utilizaron un sistema de bajo costo con energía teniendo dos alternativas de plataforma IoT en los jardines de la Ciudad de Arequipa. El desarrollo se llevó a cabo a través de varios pasos iniciando en la Recolección y análisis de los datos, Medición de las zonas de la ciudad de Arequipa, requerimientos de la propuesta, elección del software, ubicación del sistema fotovoltaico, cálculo del sistema fotovoltaico, diseño de la propuesta, elección de la plataforma y finalmente la instalación y configuración de la plataforma IoT utilizando sensores, actuadores y lenguajes de programación como Python y Arduino. Su principal resultado fue que tuvieron un ahorro del 42% de agua potable en las zonas donde se implementó el riego por goteo automatizado. Como conclusión se obtuvo que el riego por goteo automatizado propuesto arrojó un ahorro de agua potable a comparación del sistema de riego por inundación, también que las placas de desarrollo utilizadas fueron de bajo costo, lo cual generó que la población arequipeña se motivara a seguir manteniendo las áreas verdes de sus viviendas.



### **3. PROBLEMÁTICA**

El riego es una pieza fundamental del rompecabezas para el cultivo, tanto para el resultado final como para las plantas mismas. El riego manual es el riego más común que existe, sobre todo por su simpleza; solamente se necesita una manguera, un par de conexiones y, a veces, un terminal; sin embargo, requiere de mucho tiempo y en caso de tener un área de cultivo grande esta actividad se puede convertir en algo muy incómodo. Es por ello que instalar un sistema de riego autónomo le da al cultivador un control mucho más preciso sobre la cantidad de agua y nutrientes que reciben sus plantas en comparación con el riego a mano.

El presente proyecto surge de la necesidad de simular lo que sucede en las chacras del litoral peruano, pero a menor escala dado que analizaremos los datos que proporcione todo tipo de medición en una maceta, ya que en estos lugares no existe un buen control de riego debido a la carencia de recursos, ubicación, malas gestiones, etc.

Tomando en cuenta los estándares, conexiones y códigos que se requiere para implementar este tipo de sistemas y a su vez mejorar la productividad y/o consumo es importante una buena gestión del procedimiento a desarrollar para que el resultado sea óptimo en menor escala y pueda ser amplificado a futuro.

### **4. OBJETIVOS**

#### **4.1. General**

- Desarrollar un Sistema de Riego Automatizado basado en IOT para un mejor control de agua en los cultivos.

#### **4.2. Específicos**

- Diseñar el sistema de control teniendo en cuenta las condiciones del terreno y las variables ambientales.
- Implementar el riego automático cuando sea necesario, según los datos arrojados por los sensores a usar.
- Crear una comunicación entre los diferentes tipos de dispositivos empleando el protocolo MQTT.

- Mostrar la información de los datos arrojados por los sensores en dashboards usando la plataforma node red.
- Almacenar en una base de datos, los datos arrojados por los sensores, usando Arduino para su lectura y Python para su escritura en Excel.

## **5. MARCO TEÓRICO**

### **5.1. Internet de las cosas**

El Consejo de Arquitectura de Internet (IAB) en el artículo titulado “Architectural Considerations in Smart Object Networking” presenta una definición inicial sobre Internet de las Cosas (IoT) referido a un gran número de dispositivos capaces emplean los servicios de comunicación que brindan los protocolos de Internet. A este tipo de dispositivos se les conoce como “objetos inteligentes” que no son manipulados directamente por un ser humano, sino que existen como componentes en muchos entornos como edificios, vehículos, etc.

Para la IEEE Communications Magazine el Internet de las Cosas (IoT) está vinculado con los servicios en la nube ya que es un marco en el que todo lo involucrado tienen una representación y presencia en internet, a partir de ello, el IoT tiene como objetivo ofrecer nuevos servicios y aplicaciones que sirvan de puente entre el mundo físico y el virtual.

Finalmente, la entidad Internet Society dentro de su artículo titulado como “La internet de las cosas - una breve reseña” considera que el término de IoT se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana.

### **5.2. Comunicación inalámbrica**

También llamada comunicación sin el uso de ningún tipo de cables, es aquella donde se ubica un emisor y varios receptores, los cuales no se encuentran vinculados a través de difusión física, sino a través de algún tipo de red que realiza el uso de modulación de ondas electromagnéticas por medio de un área determinada. Por lo tanto, dentro del funcionamiento el emisor es el que se manifiesta mediante un aparato físico que transmite la señal y los receptores

son todos aquellos dispositivos que reciben esa señal emitida por el receptor, entre ellos están los siguientes: antenas, laptops, PDA, teléfonos celulares, entre otros.

La motivación clave para el uso de estas tecnologías es la reducción de costos por instalación y la posibilidad de reutilización en obras sucesivas, ya que en ellos no existe necesidad de tender cableado. Los módulos inalámbricos más extendidos hoy en día son los que utilizan la banda de 868 MHz. Sin embargo, debido al reducido ancho de banda de los mismos, se tiende a sistemas inalámbricos que se desarrollen en la banda sin licencia de 2.4 GHz.

### **5.3. Automatización**

La automatización consiste en usar la tecnología para realizar tareas con muy poca intervención humana. Se puede implementar en cualquier sector en el que se lleven a cabo tareas repetitivas. Sin embargo, es más común en aquellos relacionados con la fabricación, la robótica y los automóviles, así como en el mundo de la tecnología: en el software para la toma de decisiones empresariales y los sistemas de TI.

### **5.4. Sistemas de control automático**

Si este término lo asociamos a nuestra cotidianidad, si nos referimos al ámbito doméstico, se requiere regular la temperatura y humedad de las oficinas y casas, al igual que al transportarnos por la ciudad se necesita manejar un vehículo que siga patrones y evite multas reguladas por señalizaciones y semáforos, entonces en ambas situaciones por analogía se entiende que un sistema de control automático debe verificar el consumo de energía y desarrollar la mejor estrategia para determinado caso. La búsqueda para cumplir tales “objetivos” requiere normalmente de un sistema de control que al ser automatizado ya tiene reacciones o respuestas ante patrones.

Entonces, técnicamente un sistema de control automático es aquel conjunto de técnicas y herramientas que permiten mantener una condición física o cantidad medible en un valor deseado. Para ello, se vale de una medición instantánea de la variable a controlar y de una comparación con un valor patrón o de referencia.

Existen aplicaciones de este tipo de sistemas que se involucran en varios escenarios como en el transporte que se tiene un ABS (sistema antibloqueo) o el sistema de suspensión activa que son encargados de medir o evaluar señales (derrape de ruedas, balanceo y cabeceo del vehículo) y actúa sobre freno para mejorar de manera automática la seguridad a bordo. También dentro de las plantas de producción de productos químicos en general, fabricación de azúcar, procesar minerales, industria automotriz, entre otras, utilizan de manera muy amplia los sistemas de control automático destinado a regular variables físicas como temperatura, presión, caudal, flujo, entre otras variables que intervienen de manera definitiva en el buen desarrollo del proceso de fabricación.

Este tipo de sistemas generalmente se dividen en dos clases, donde la primera es llamada “sistemas de control a lazo abierto” que consiste en un sistema de control automático que en el que la salida no es afectada por la señal de entrada y usualmente se divide en dos partes, el controlador y el proceso controlado, mientras que la segunda es llamada “sistemas de control a lazo cerrado” en el que el controlador se alimenta de la señal de error de desempeño, la cual representa la diferencia entre la señal de entrada y de realimentación con el fin de reducir el error y llevar la salida del sistema a un valor deseado.

## **5.5. NODE-RED**

Es considerado una herramienta de desarrollo open-source basada en programación visual que fue creada por IBM para conectar dispositivos de hardware, APIs y servicios en línea. Diseñada para comunicar hardware, es un sistema que permite simplificar la programación, la conectividad y los servicios, distribuyéndolos de forma eficiente para ganar tiempo en el registro y gestión de la información.

Node-RED permite conectar gráficamente bloques predefinidos, llamados nodos, para desarrollar una tarea concreta. La conexión de los nodos, habitualmente una combinación de nodos de entrada, nodos de procesamiento y nodos de salida, forman lo que conocemos como flow. Entre los nodos disponibles podemos encontrar desde protocolos estándar como MQTT, REST, Modbus, OPC-UA, Bacnet, Websocket hasta integraciones a APIs de terceros como Microsoft Azure, Amazon Web Services, Twitter, Facebook y muchas más.

## 5.6. Sistema de riego

Aunque existen diversas técnicas en la actualidad, el sistema de riego automático es cada vez más frecuente gracias a los múltiples beneficios que presenta. Se trata de un sistema de riego que provee de agua a los cultivos de manera automatizada y que emplea normalmente la aspersión o el goteo. Existen sistemas de riego automático que combinan tanto la aspersión como el goteo y que permiten combinar las ventajas de ambas técnicas.

Antes de proceder con la instalación de un sistema de riego automático, se deberá realizar un estudio previo del terreno y el tipo de cultivo, para definir de manera adecuada cómo se deberá distribuir el agua para optimizar al máximo los resultados. Asimismo, deberás tener clara su ubicación, la cantidad de agua a emplear o la frecuencia con la que se producirá el riego automático.

Los sistemas de riego automático se han convertido en grandes aliados para las cosechas y cultivos. Asimismo, se trata de una tecnología en constante evolución y que, gracias a los últimos avances e innovaciones, incorporan cada vez más novedades para su uso más eficiente.

## 5.7. Componentes del sistema de riego

### 5.7.1. ESP8266

El ESP8266 es un chip Wi-Fi de bajo coste con pila TCP/IP completa y capacidad de MCU (Micro Controller Unit) producida por el fabricante chino Espressif Systems, con sede en Shanghai.

Figura 1. ESP8266



Voltaje	3.3 V
Consumo de corriente	10 $\mu$ A – 170 mA
Memoria Flash	16 MB máx. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analógico a digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r
Máximas conexiones simultáneas	5

### 5.7.2. Sensor ultrasonido

Son dispositivos que generan o detectan energía ultrasónica. Se pueden dividir en tres amplias categorías: transmisores, receptores y transeptores.

Figura 2. Sensor ultrasonido



### 5.7.3. DHT22

El sensor DHT22 (también conocido como AM2302) es un sensor de bajo costo digital que mide la temperatura y la humedad. Utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante, y suministra una señal digital en el pin de datos.

Figura 3. DHT22



### 5.7.4. Sensor humedad del suelo salida analógica

El Sensor de humedad de Suelo FC-28 permite medir de forma sencilla la humedad del suelo por medio de 2 electrodos resistivos. El sensor es ideal para monitorear el nivel de humedad de tus plantas y así recordar cuando necesitan ser

regadas o incluso para realizar un sistema totalmente automatizado de riego añadiendo una válvula o una bomba de agua.

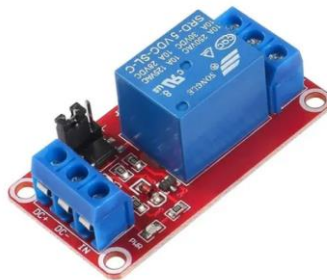
Figura 4. Sensor humedad del suelo salida analógica



#### 5.7.5. Relé

Un relé, o relevador, es un componente eléctrico que permite abrir o cerrar un circuito en función de una bobina y un electroimán. Este dispositivo permite cerrar y abrir circuitos eléctricos, que pueden ser dependientes o independientes entre sí, para generar movimientos/funciones.

Figura 5. Relé



#### 5.7.6. 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.

Figura 6. Bomba de agua



#### **5.7.7. LED**

Es un diodo emisor de luz, en su interior hay un semiconductor que, al ser atravesado por una tensión continua, emite luz, lo que se conoce como electroluminiscencia.

Figura 7. LED



#### **5.7.8. Led RGB**

Este tipo de led hace uso de los colores rojo, azul y verde que se combinan para producir más de 16 millones de tonos de luz. La conexión de este tipo de led se realiza mediante cuatro pines, uno por cada color y otro para GND.

Figura 8. Led RGB





### 5.7.9. Resistencia

También llamado resistor, es un componente ubicado en casi todos los circuitos eléctricos y tiene como objetivo modificar o alterar el paso de la corriente y en otros casos para generar calor.

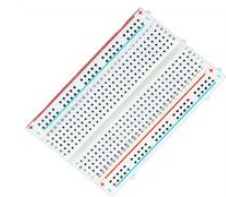
Figura 9. Resistencia



### 5.7.10. Protoboard

Es similar a un circuito impreso temporal con una forma y tamaño generalizado. Utilizada comúnmente para pruebas y prototipos, se usa insertando los terminales de los dispositivos electrónicos en los orificios de tal manera que tengan continuidad.

Figura 10. Protoboard



### 5.7.11. Cables para protoboard

Cables Dupont para Protoboard M/M son cables jumper de 15cm de largo terminados como macho a hembra y son de muchos colores. Estos se usan para conectarse desde cualquier header hembra o macho en cualquier placa. Múltiples jumpers se pueden conectar uno al lado del otro en un conector de 2.54mm.

Figura 11. Cables para protoboard



## **6. IMPLEMENTACIÓN DEL SISTEMA**

### **6.1. Descripción general**

El presente proyecto consiste en un sistema de riego que provee agua a una maceta contenida de tierra, de una manera automatizada. Para el riego, se consideraron factores ambientales como la temperatura, humedad ambiental y humedad de suelo. Los datos de cada factor serán capturados por los sensores implementados y según los datos recibidos, el sistema determinará si la maceta debe ser regado o no.

### **6.2. Funcionamiento del sistema**

El sistema hace uso del sensor de ultrasonido para medir la cantidad de agua que hay en el recipiente para regar la planta si este está bajo entonces se activara el led rojo y se enviara un correo, utilizando el node red, indicando que se debe llenar el recipiente de agua para el riego.

El sistema tiene dos formas de regar la planta la primera se ejecutará cada día para ello el node red enviará un mensaje al sistema para activar el riego, y la segunda forma será de acuerdo a los valores obtenidos por los sensores.

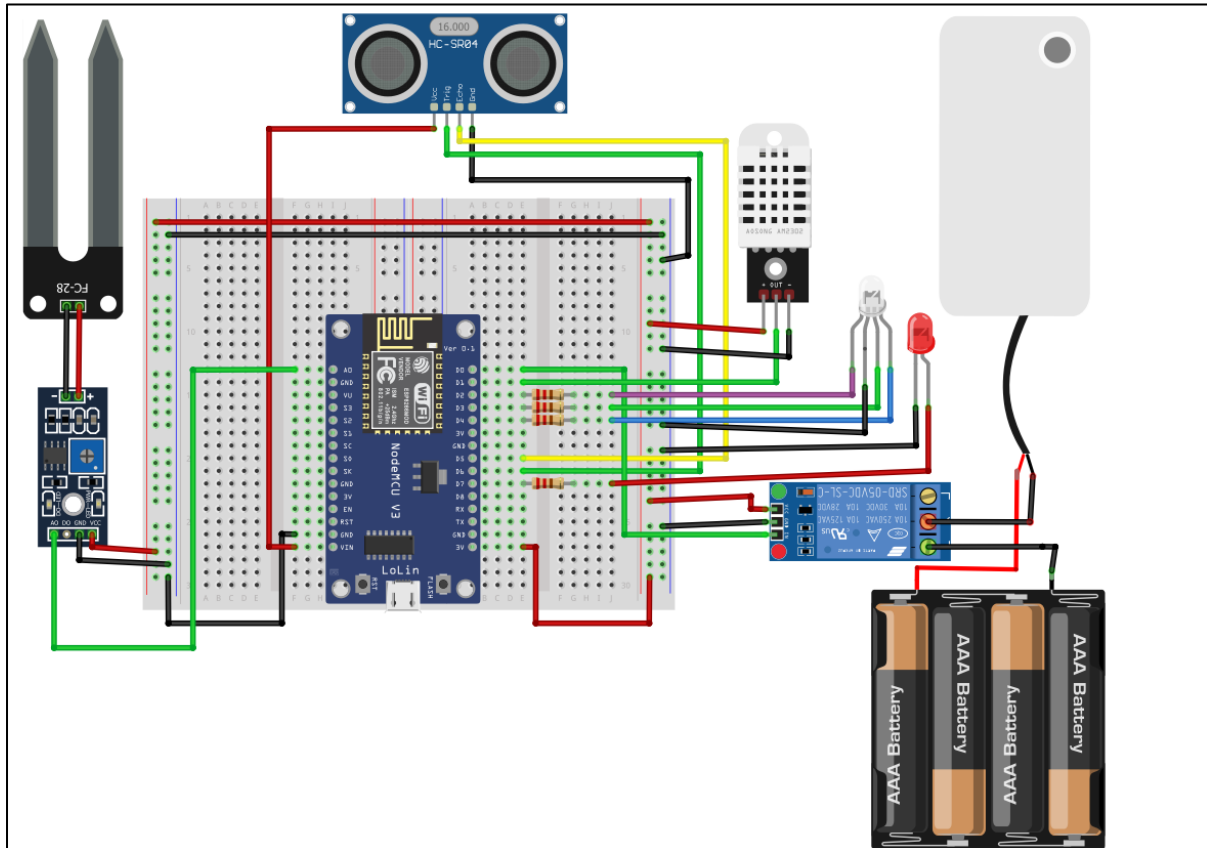
El sensor del suelo capta la humedad del suelo de la planta y el sensor de temperatura capta la humedad y la temperatura del ambiente, según estos datos cuando la humedad sea baja o cuando la temperatura sea elevada se iniciará el riego.

Para realizar el riego el sistema envía una señal al relé que activa al motor de agua el cual realizara el riego de la planta durante un minuto.

### 6.3. Diagrama del circuito

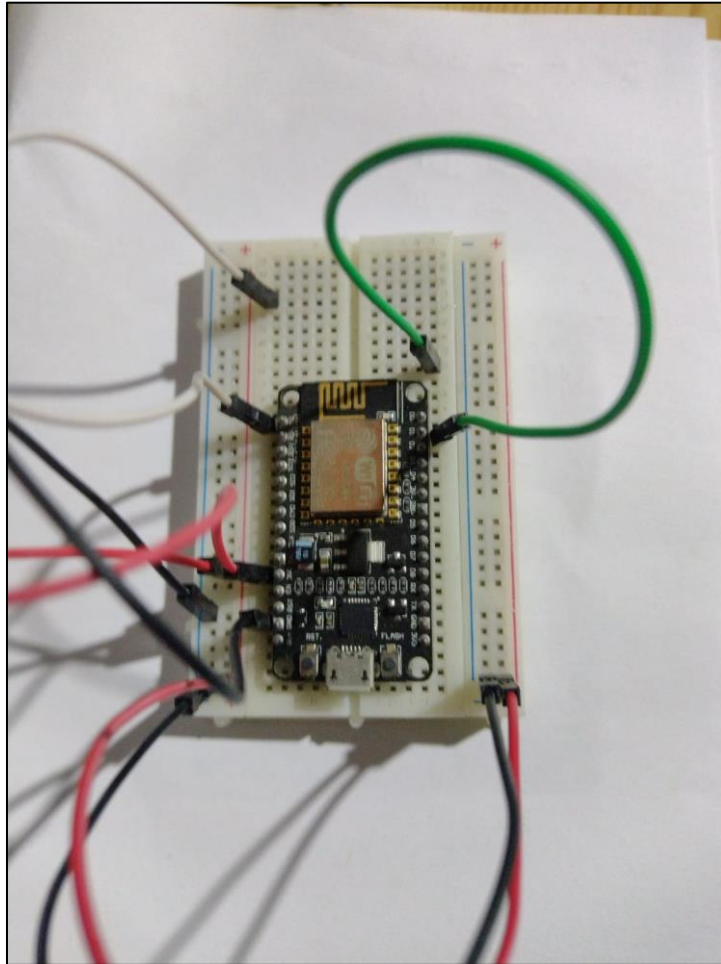
Se implemento el sistema según el diagrama que se muestra en la imagen usando los componentes anteriormente mencionados.

Figura 12. Diagrama del circuito

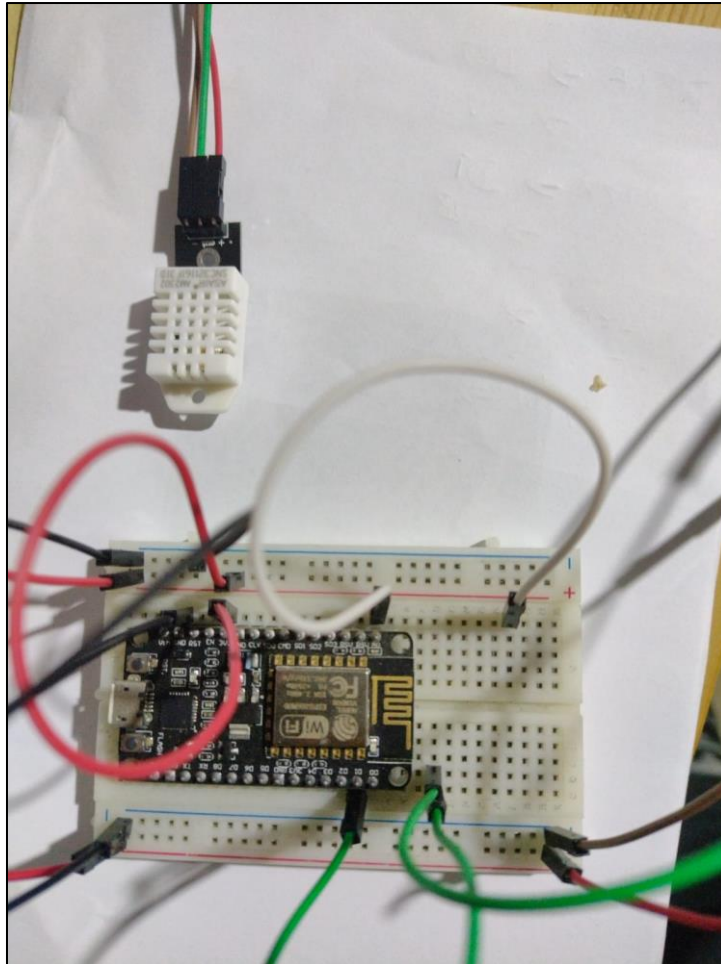


#### 6.4. Ensamblado del circuito

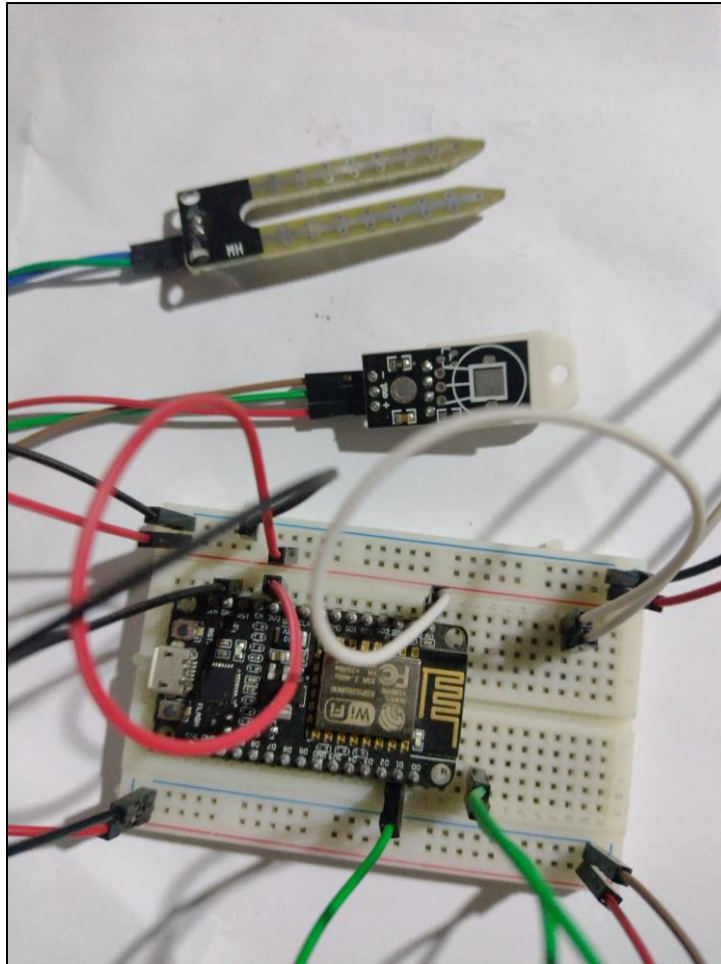
En la siguiente imagen se muestra al ESP8266 en el protoboard para comenzar con el ensamblado del sistema



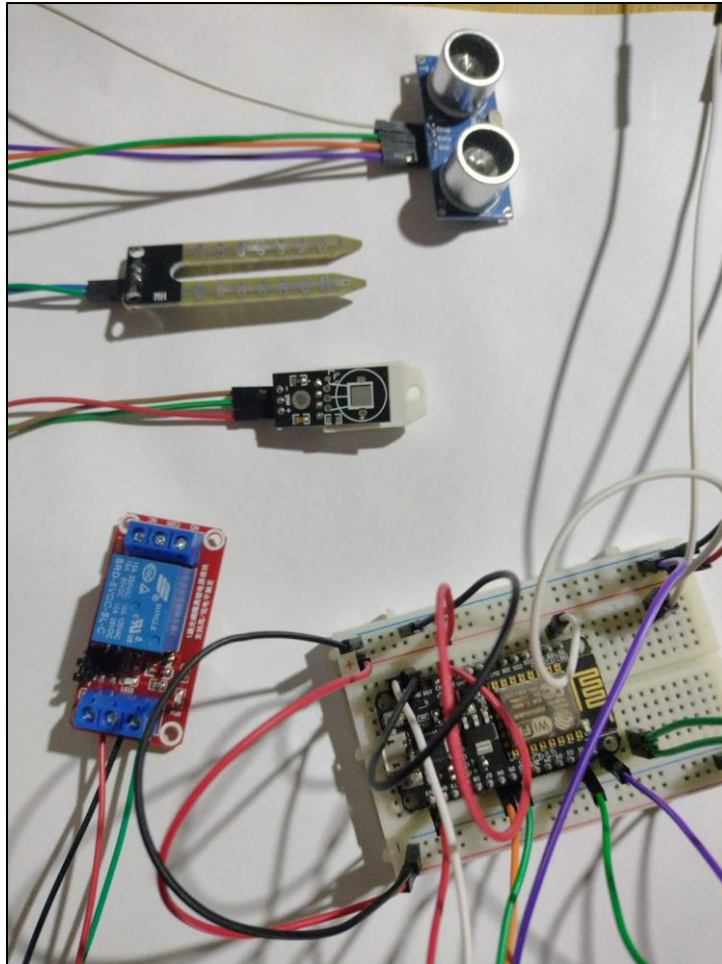
Luego se le conecta el sensor DHT22 el cual cuenta con tres pines, dos para la energía y tierra, y el tercer pin para los datos que va al pin D1 del ESP8266.



Se conecta el sensor de humedad del suelo el cual tiene tres pines dos para la energía y tierra, y el tercer pin para los datos que va al pin A0 del ESP8266, el cual es una entrada analógica.



Luego se conectan el sensor de ultrasonido el cual cuenta con cuatro pines dos para la energía y tierra, un pin llamado “trigger” y un pin “echo” que van al pin D5 y D6 respectivamente. También se conecta el relé el cual tiene cinco pines, por un lado tiene tres pines de los cuales dos son para la energía y tierra, y el tercero que recibe la señal del pin D0 del ESP8266 y de los tres pines restantes solo usaremos dos el “com” que va conectado al lado negativo del motor y el pin “NO” que va conectado al lado negativo del cargador de 5V, los pines positivos del motor y del cargador van conectados juntos.



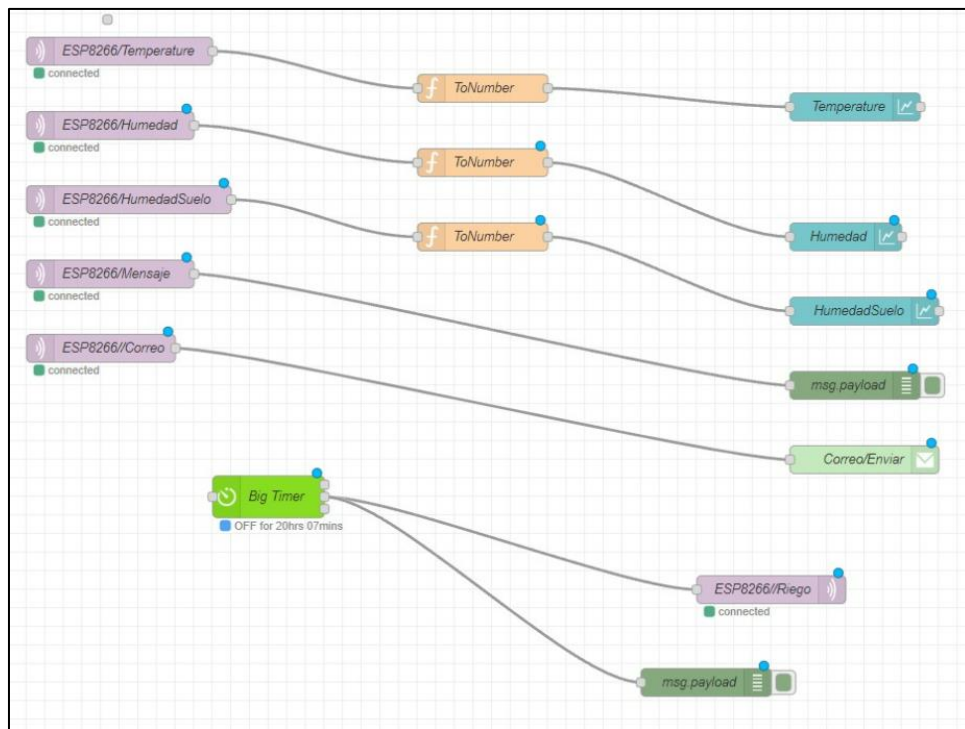
Finalmente se conecta un led común y un led RGB, con el led común indicamos que se está realizando el regado de las plantas y con el led RGB se indica que el recipiente se encuentra vacío y necesita llenarse







Figura 13. Implementación de NODE RED



Los nodos MQTT de entrada tienen la siguiente estructura.

- **Nodo MQTT de entrada.**

Se asigna un servidor broker y también el nombre del tópico por el cual se publicarán los mensajes, el QoS hace referencia a que el nodo MQTT es de entrada y la salida se establece como “detectada automáticamente”.

Cada tópico deberá estar asociado a determinado valor que obtengamos de un sensor.

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server 192.168.1.25:1883

Action Subscribe to single topic

Topic Temperature

QoS 2

Output auto-detect (parsed JSON object, string or buf)

Name ESP8266/Temperature

- **Nodo MQTT de salida.**

Se establecen los mismos parámetros del nodo MQTT entrada a diferencia del QoS. Para este caso el QoS se establece como 0, lo cual nos indica que se trata de un nodo MQTT de salida.

**Edit mqtt out node**

Delete Cancel Done

**Properties**

Server 192.168.1.25:1883

Topic Riego

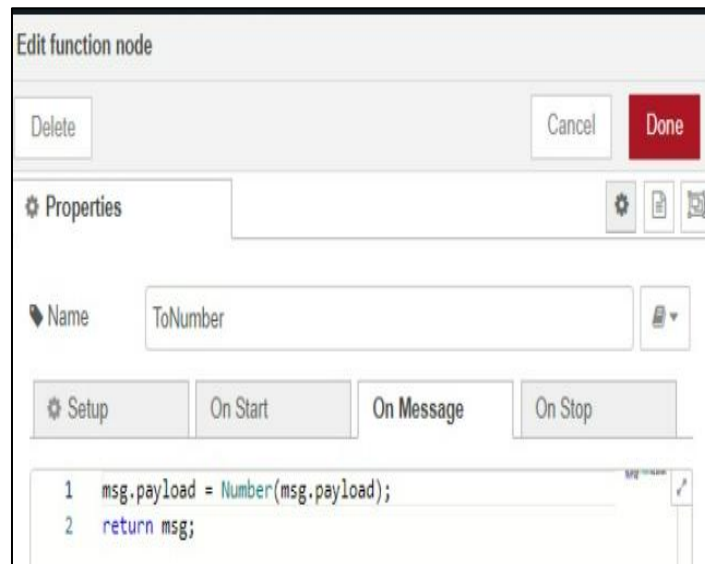
QoS 0 Retain

Name ESP8266/Riego

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

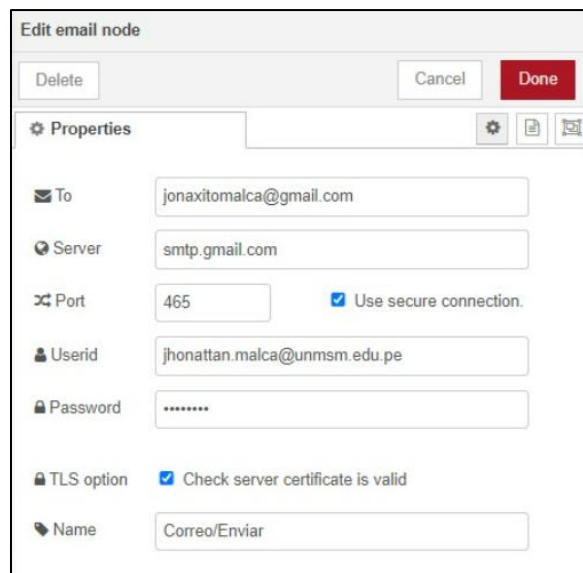
- **Nodo ToNumber:**

Todos los nodos MQTT tienen la particularidad de enviarnos valores de tipo cadena, por lo que al momento de enviarlos a mostrar en un dashboard no sería posible ya que solo aceptan datos de tipo numérico, por lo que a través de esta función se realizaría el cambio de variable de cadena a número.



- **Nodo Correo/Enviar:**

A través del protocolo simple de transferencia de correo, nos permitirá realizar el envío de correos hacia un destinatario previamente definido (jonaxitomalca@gmail.com), este correo se enviará con el mensaje que previamente hayamos enviado al tópico Mensaje



### ▪ **Nodo Bigtimer:**

Este nodo nos permite establecer el tiempo en el cual se realizará automáticamente el riego, ya que esto permitirá enviar un valor HIGH o LOW al pin en el cual se encuentra conectado el Relé, durante determinado tiempo establecido en este nodo.

En el ejemplo se establece que la salida sea:

- ON: 18:37hrs - 18:38hrs
- OFF El resto del día

Por lo cual el Relé se activará una vez por un minuto.

## 6.6. Base de datos

Se realizará la persistencia de la data obtenida por cada sensor, y será mostrado en un archivo xlsx.

```

//Inicializamos los valores de cada una de las salidas de los sensores
int hum = 0;
int temp = 0;
int humedad =0;
int i =0;

#define DHTpin 5 // declaramos el pin que va al sensor
DHTesp dht; // declaramos una variable tipo DHTesp

void setup() {
  pinMode(BUILTIN_LED, OUTPUT); //Inicializa el BUILTIN_LED como salida
  Serial.begin(115200);
}

void loop() {
  TempAndHumidity data = dht.getTempAndHumidity();

  hum = data.humidity; //Calibracion de los valores del sensor de humedad
  temp = data.temperature; //Calibracion de los valores del sensor de temperatura
  humedad = map(analogRead(A0), 0, 1024, 100, 0); //Se realiza la calibracion de los valores
  //que el sensor de humedad del suelo obtiene.
  //Imprimir los valores que irán en el excel
  Serial.print(i);
  Serial.print(",");
  Serial.print(hum);
  Serial.print(",");
  Serial.print(temp);
  Serial.print(",");
  Serial.print(humedad);
  Serial.println(";");
  i++;
  delay(3000);
}

```

Para ello se hace uso del lenguaje Python como se muestra en la siguiente imagen.

```

import serial
import xlwt
from datetime import datetime

class SerialToExcel:

    def __init__(self,port,speed):

        self.port = port
        self.speed = speed

        self.wb = xlwt.Workbook()
        self.ws = self.wb.add_sheet("Data from Serial",cell_overwrite_ok=True)
        self.ws.write(0, 0, "Data from Serial")
        self.columns = ["Date Time"]
        self.number = 100

    def setColumns(self,col):
        self.columns.extend(col)

    def setRecordsNumber(self,number):
        self.number = number

    def readPort(self):
        ser = serial.Serial(self.port, self.speed, timeout=1)
        c = 0
        for col in self.columns:
            self.ws.write(1, c, col)
            c = c + 1
        self.fila = 2
        i = 0
        while(i<self.number):
            line = str(ser.readline())
            if(len(line) > 0):
                now = datetime.now()
                date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
                print(date_time,line)
                if(line.find(",")>0):
                    c = 1
                    self.ws.write(self.fila, 0, date_time)
                    columnas = line.split(",")
                    for col in columnas:
                        self.ws.write(self.fila, c, col)
                        c = c + 1

                    i = i + 1
                    self.fila = self.fila + 1

    def writeFile(self,archivo):
        self.wb.save(archivo)

```

Luego con la clase “example.py” se imprimen los datos en un archivo Excel.

```

example.py X
serialToExcel-master > example.py > ...
1  from serialToExcel import SerialToExcel
2
3  serialToExcel = SerialToExcel("COM3",115200)
4
5  #columnas = ["Nro Lectura","Valor"]
6
7  serialToExcel.setColumns(["Nro Lectura","Humedad","Temperatura","Humedad Suelo"])
8  serialToExcel.setRecordsNumber(30)
9  serialToExcel.readPort()
10
11  serialToExcel.writeFile("archivoPrueba.xls")

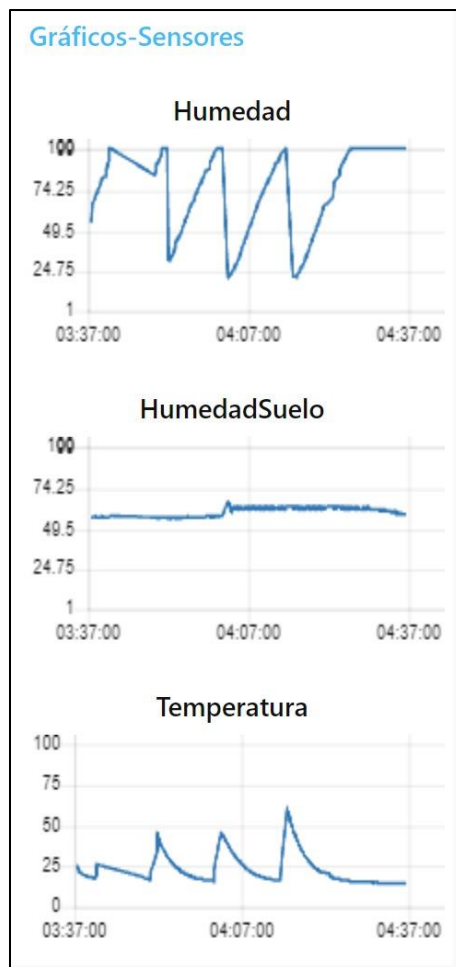
```

Los datos obtenidos de los sensores se almacenan en un archivo Excel como se muestra es la imagen.

	A	B	C	D	E	F	G	H	I	J	K
1	Data from Serial										
2	Date Time	Nro Lectura	Humedad	Temperatura	Humedad Suelo						
3	08/29/2022, (b'57	99	16	52:\r\n'							
4	08/29/2022, (b'58	99	16	53:\r\n'							
5	08/29/2022, (b'59	99	16	53:\r\n'							
6	08/29/2022, (b'60	99	16	53:\r\n'							
7	08/29/2022, (b'61	99	16	52:\r\n'							
8	08/29/2022, (b'62	99	16	53:\r\n'							
9	08/29/2022, (b'63	99	16	53:\r\n'							
10	08/29/2022, (b'64	99	16	52:\r\n'							
11	08/29/2022, (b'65	99	16	52:\r\n'							
12	08/29/2022, (b'66	99	16	53:\r\n'							
13											
14											
15											

## 7. RESULTADOS

Como resultados se muestran los datos obtenidos de los sensores en el dashboard creado con el node red el cual obtiene estos datos mediante el protocolo de comunicación mqtt. Se puede observar en la imagen las lecturas de la humedad, la humedad del suelo y la temperatura y las variaciones en la ejecución.

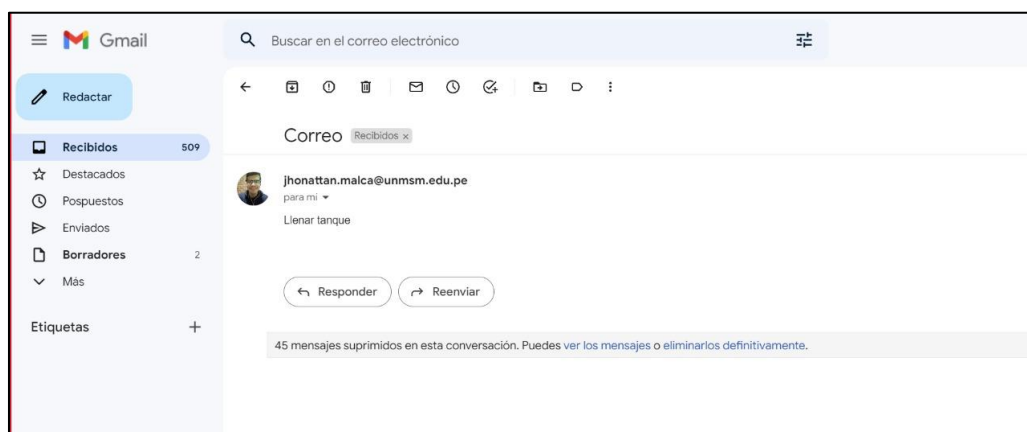


Y es de acuerdo a la variación de la temperatura ya si esta es mayor a 25 °C o humedad del ambiente baja a menos de 50% o la humedad del suelo es menor a 80% que se activa el motor regando a la planta como se muestra en la imagen





Así también cuando el sensor detecta que el recipiente de agua se queda vacío, se enciende la luz roja del circuito y se envía un mensaje al usuario para avisarle que se debe llenar el recipiente con agua.



## 8. CONCLUSIONES

- El desarrollo del sistema logró implementar correctamente el riego automático para regar la maceta solo en situaciones que sea necesario, según los datos arrojados por los sensores.
- La generación de dashboard logró mostrar la información arrojada por los sensores, lo cual hace una forma sencilla de interpretar por los usuarios.
- Los datos arrojados por los sensores se logran almacenar en una base de datos, lo cual beneficia a los usuarios, ya que podrán realizar informes según las fechas correspondientes.

## 9. BIBLIOGRAFÍA

- Carvajal, R. G., de Ingenieros, E. E. T. S., REBOLLO, M. D. D., & Superior, P. R. L. (2012). Tecnologías inalámbricas. *Segur. y Medio Ambient*, 10-21.
- Cortes-Cadavid, V., & Vargas-García, M. F. (2021). Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante Iot en los cultivos urbanos de la fundación mujeres empresarias Marie Poussepin.
- Esquicha Tejada, J. D. (2022). Evaluación de la eficiencia del riego en los jardines de la ciudad de Arequipa empleando un sistema automatizado de bajo costo con energía cero y una plataforma IOT.
- Roca Roca, J. M. (2020). *Diseño e implementación de un sistema de riego con funciones Smart* (Doctoral dissertation, Universitat Politècnica de València).
- Salazar, J., & Silvestre, S. (2016). Internet de las cosas. *Techpedia. České vysoké učení technické v Praze Fakulta elektrotechnická*.
- Tabaa, M., Chouri, B., Saadaoui, S., & Alami, K. (2018). Industrial communication based on modbus and node-RED. *Procedia computer science*, 130, 583-588.

## ANEXOS

- Código Arduino del sistema:

```
-  
/***** Librerías usadas *****/  
//Libreria para conectarse a una red Wifi con el ESP8266  
#include <ESP8266WiFi.h>  
//Libreria para que el microcontrolador se comporte como un cliente MQTT  
#include <PubSubClient.h>  
//Libreria para realizar la calibracion de los sensores de temperatura y  
humedad  
#include "DHTesp.h"  
  
/***** Credenciales para la conexión MQTT *****/  
const char* ssid = "MOVISTAR-MalcaR."; //Nombre de red  
const char* password = "I8k7l911@MalcaRamirez25.Com"; // Contraseña de la red  
const char* mqtt_server = "192.168.1.25"; //ip de broker  
const char* mqtt_user = "grupo8";  
const char* mqtt_password = "grupo8";  
  
/***** Definimos los pines usados *****/  
//Se define los pines para los diferentes sensores  
const int TRIG = 14; // trigger en pin D5  
const int ECHO = 12; // echo en pin D6  
const int relay = 16; // relay en PIN D0  
#define DHTpin 5 // declaramos el pin nro 5 - D1 para el sensor de  
temperatura y humedad  
const int ledRojo = 2; //pin D4 LED-RGB  
const int ledVerde = 0; //pin D3 LED-RGB  
const int ledAzul = 4; //pin D2 LED-RGB  
const int led = 13; //pin D7 LED  
  
const float VelSon = 34000.0; // Constante velocidad sonido en cm/s  
  
bool validar= true; //Auxiliar para accionar solo una vez el envio de correo  
bool validarriego= true; //Auxiliar para accionar como máximo dos veces el  
riego  
  
/***** Inicializamos los valores de cada una de las salidas de los sensores  
*****/  
float hum = 0;  
float temp = 0;  
float humedad = 0;  
float distancia;
```

```

String regar;

/***** Vector de caracteres para enviar el mensaje al topico *****/
char datos[100];
char datos2[100];
char datos3[100];
char datos4[100];
char datos5[100];

DHTesp dht; // declaramos una variable tipo DHTTest

WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

void setup_wifi() {
  //Configuracion del wifi para establecer la conexion
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  //Wifi en modo estación (STA)
  WiFi.mode(WIFI_STA);
  //Conectarse a la red wifi
  WiFi.begin(ssid, password);
  //Mientras no nos conectemos a la red wifi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  //Inicializa el generador de numeros pseudo aleatorios con una semilla
  //establecida por el numero de ms en el cual arduino comenzo a ejecutar el
programa
  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

```

```

void callback(char* topic, byte* payload, unsigned int length) {
    String cadena;
    Serial.print("Message arrived [");
    Serial.print(topic); //Topico al cual nos suscribimos
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        cadena += (char)payload[i];
    }
    regar = cadena; //Obtenemos el mensaje de salida del topico "Regar"

}

void reconnect() {
    //Codigo que se ejecutara mientras no se establezca la conexion con el
    protocolo MQTT.
    while (!client.connected()) { //Verifica si el cliente esta conectado al
    servidor
        Serial.print("Attempting MQTT connection...");
        //Crea un cliente con un id aleatorio, el cual se usara para conectar al
        servidor
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        //Si la conexion se realiza de forma correcta.
        if (client.connect(clientId.c_str())) {
            Serial.println("connected");
            //Publica un mensaje en el topico "outTopic"
            client.publish("outTopic", "hello world");
            //Realiza una suscripción a un topico especifico "Riego"
            client.subscribe("Riego");
        } else {
            //Mensaje que se imprime en caso la conexion no se pudo realizar
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            //Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void setup() {
    //Se definen los pines del LED RGB como salida
    pinMode(ledRojo, OUTPUT);
}

```

```

pinMode(ledVerde,OUTPUT);
pinMode(ledAzul,OUTPUT);
//Se define LED como salida
pinMode(led,OUTPUT);

pinMode(TRIG, OUTPUT); // trigger como salida(sensorultrasonido)
pinMode(ECHO, INPUT); // echo como entrada(sensorultrasonido)
pinMode(relay, OUTPUT); // Configurar relay como salida o OUTPUT
dht.setup(DHTpin, DHTesp::DHT22);
pinMode(A0,INPUT); //El pin A0 establecido como entrada
Serial.begin(115200);
setup_wifi();//Configuracion del wifi para establecer la conexion
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

void loop() {

TempAndHumidity data = dht.getTempAndHumidity(); //Implementacion de la
funcion que permitira realizar la calibracion de sensores.

if (!client.connected()) {//Si el cliente no esta conectado al servidor
reconnect();//Llamada a funcion reconnect
}
client.loop();//Funcion que permite que el cliente procese los mensajes
entrantes y mantenga la conexion con el servidor.

unsigned long now = millis();
if (now - lastMsg > 2000) {
lastMsg = now;
hum = data.humidity;//Calibracion de los valores del sensor de humedad
temp = data.temperature;//Calibracion de lo valores del sensor de
temperatura
humedad=map(analogRead(A0),0,1024,100,0);//Se realiza la calibracion de
los valores que el sensor de humedad del suelo obtiene.
//Obtener valores del sensor ultrasonido
iniciarTrigger();

unsigned long tiempo = pulseIn(ECHO, HIGH);// La función pulseIn obtiene
el tiempo que tarda en cambiar entre estados, en este caso a HIGH

distancia = tiempo * 0.000001 * VelSon / 2.0; // Obtenemos la distancia
en cm, hay que convertir el tiempo en segundos ya que está en microsegundos por
eso se multiplica por 0.000001

```

```

//Publicaciones para cada uno de los topicos
sprintf(datos, "%.2f", temp);
client.publish("Temperature",datos);

sprintf(datos2, "%.2f", hum);
client.publish("Humedad",datos2);

sprintf(datos3, "%.2f", humedad);
client.publish("HumedadSuelo",datos3);
}

//Riego opcional, en caso se cumplan con las condiciones de las lecturas de
los sensores
if ((humedad <50 || temp > 20 || hum <80)&& validarriego){
    strcpy(datos4, "inicia riego ");
    digitalWrite(relay, HIGH); // envia señal alta al relay
    validarriego=false;
    digitalWrite(led, HIGH);//
    delay(60000);
    digitalWrite(led, LOW);
    digitalWrite(relay, LOW); // envia señal baja al relay
}
else{
    strcpy(datos4, "no regar");
}

delay(1000);
if(distancia<=6){
    //Hacer color rojo
    digitalWrite(ledRojo,0);
    digitalWrite(ledVerde,0);
    digitalWrite(ledAzul,0);
    validar=true;
}
//Condicion para que se envíe el correo
if (distancia>6 && validar){
    strcpy(datos5, "Llenar tanque");
    //Hacer color rojo
    digitalWrite(ledRojo,0);
    digitalWrite(ledVerde,0);
    digitalWrite(ledAzul,255);

    validar= false; //validar distancia
    client.publish("Correo",datos5);
}
client.publish("Mensaje",datos4);

```



```

//Imprimir los valores que irán en el excel
Serial.print(i);
Serial.print(",");
Serial.print(hum);
Serial.print(",");
Serial.print(temp);
Serial.print(",");
Serial.print(humedad);
Serial.println(";");
i++;

//Riego obligatorio diario.
if(regar == "1")
{
    validarriego = true; //riego dos veces por dia (maximo)
    digitalWrite(relay, HIGH); // envia señal alta al relay
}
else if(regar == "0"){
    digitalWrite(relay, LOW); // envia señal baja al relay
}
    delay(5000);
}
// Método que inicia la secuencia del Trigger para comenzar a medir
void iniciarTrigger()
{
    // Ponemos el Triiger en estado bajo y esperamos 2 ms
    digitalWrite(TRIG, LOW);
    delayMicroseconds(2);

    // Ponemos el pin Trigger a estado alto y esperamos 10 ms
    digitalWrite(TRIG, HIGH);
    delayMicroseconds(10);

    // Comenzamos poniendo el pin Trigger en estado bajo
    digitalWrite(TRIG, LOW);
}

```

- Código serialToExcel.py para la persistencia.

```

import serial
import xlwt
from datetime import datetime
class SerialToExcel:
    def __init__(self,port,speed):
        self.port = port
        self.speed = speed
        self.wb = xlwt.Workbook()
        self.ws = self.wb.add_sheet("Data from Serial",cell_overwrite_ok=True)
        self.ws.write(0, 0, "Data from Serial")

```

```

        self.columns = ["Date Time"]
        self.number = 100
    def setColumns(self,col):
        self.columns.extend(col)
    def setRecordsNumber(self,number):
        self.number = number
    def readPort(self):
        ser = serial.Serial(self.port, self.speed, timeout=1)
        c = 0
        for col in self.columns:
            self.ws.write(1, c, col)
            c = c + 1
        self.fila = 2
        i = 0
        while(i<self.number):
            line = str(ser.readline())
            if(len(line) > 0):
                now = datetime.now()
                date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
                print(date_time,line)
                if(line.find(",")>0):
                    c = 1
                    self.ws.write(self.fila, 0, date_time)
                    columnas = line.split(",")
                    for col in columnas:

                        self.ws.write(self.fila, c, col)
                        c = c + 1
                    i = i + 1
                    self.fila = self.fila + 1
    def writeFile(self,archivo):
        self.wb.save(archivo)

```

- Example.py

```

from serialToExcel import SerialToExcel
serialToExcel = SerialToExcel("COM3",115200)
#columnas = ["Nro Lectura","Valor"]
#Indicamos las columnas que se mostraran en la tabla
serialToExcel.setColumns(["Nro Lectura","Humedad","Temperatura","Humedad
Suelo"])
#Indicamos el número de registros que queremos obtener
serialToExcel.setRecordsNumber(20)
serialToExcel.readPort()
#Se indica el archivo excel que se va a generar con la tabla creada
serialToExcel.writeFile("archivo2.xls")

```