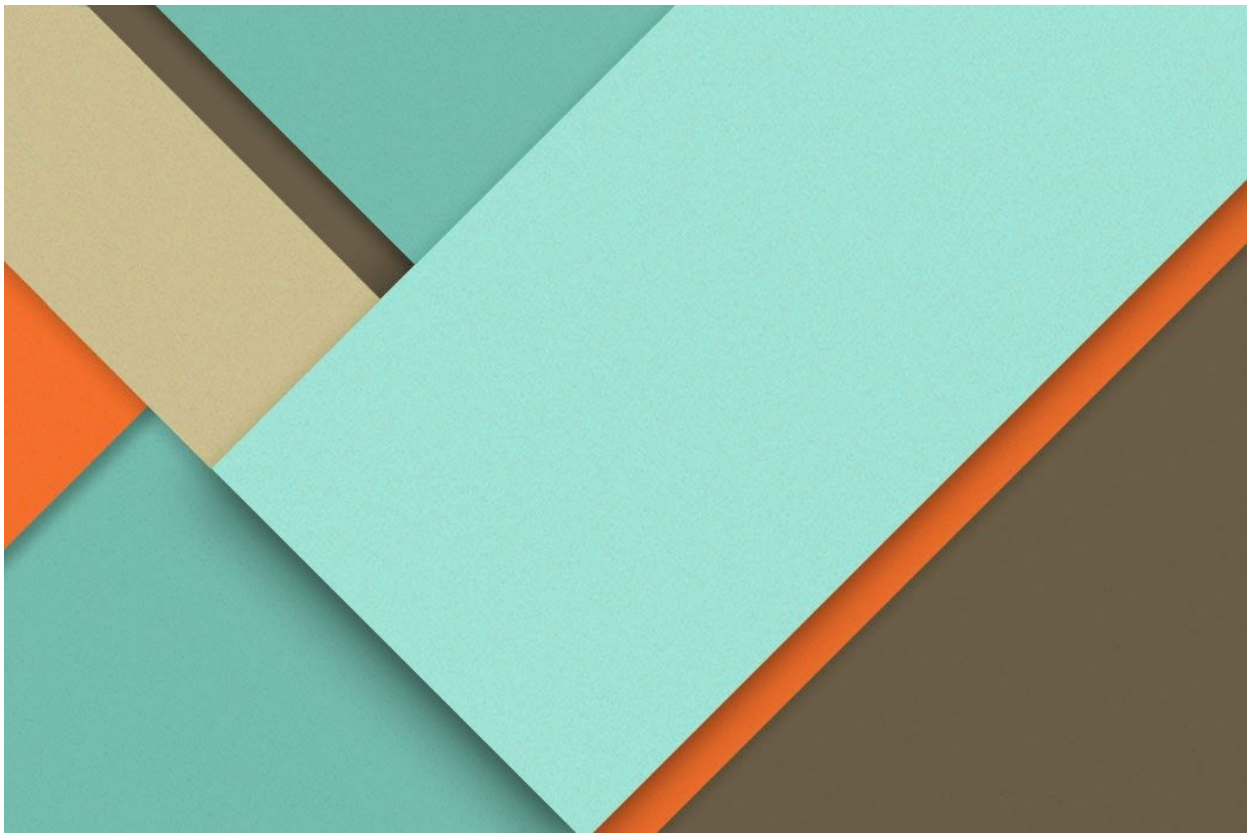
..



# Automatización de un huerto

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Grecia Pacheco A01366730

Jorge Flores A01769059

Rubén Ruiz A01366792

Miguel Santamaria A01366795

Alberto Navarrete A01422954

TC1004B.1 (Gpo 1) CAMPUS TOLUCA

Implementación de internet de las cosas

SEMESTRE AGOSTO-DICIEMBRE 2020

# 1. Introducción

El Internet de las Cosas (Internet of Things), también llamado Internet de todo o Internet industrial, es un nuevo paradigma tecnológico concebido como una red global de máquinas y dispositivos capaces de interactuar entre sí. El IoT es reconocido como una de las áreas más importantes de la tecnología del futuro y está recibiendo la atención de una amplia gama de industrias debido al potencial que tiene de automatizar procesos y con ello ser más eficientes, reducir costos y mantenerse competitivos. Sin embargo, la industria no es el único sector que puede beneficiarse de esta tecnología ya que, como lo ejemplifica el desarrollo de este proyecto, también es aplicable para ayudar a resolver problemáticas del crecimiento urbano.

Uno de los Objetivos de Desarrollo Sostenible (ODS) de la ONU es lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles. “El mundo se está urbanizando cada vez más. Desde 2007, más de la mitad de la población mundial ha estado viviendo en ciudades, y se prevé que esa proporción aumente al 60% para 2030” (United Nations Statistics Division Development Data and Outreach Branch, 2020). En los países en vías de desarrollo, el proceso de urbanización está relacionado con el aumento de la pobreza y la contaminación del medio ambiente por la falta de planeación urbana; la mala distribución de recursos provoca inseguridad alimentaria y desnutrición, especialmente para los grupos vulnerables como lo son los niños, las mujeres embarazadas y lactantes; y el aumento del desempleo por el cambio en el uso de suelo para continuar con el crecimiento de las ciudades (Orsini, Kahane, Nono-Womdim & Gianquinto, 2013).

La agricultura urbana representa una oportunidad para mejorar el suministro de alimentos, las condiciones de salud, la economía local, la integración social y la sostenibilidad ambiental en conjunto. Además, la agricultura urbana también aporta beneficios ecológicos ya que promueve la reducción de los desechos de la ciudad; mejora la biodiversidad urbana, el paisaje y la calidad del aire; y, en general, reduce el impacto ambiental relacionado con el transporte y el almacenamiento de alimentos ya que al ser productos locales se reduce el consumo de combustibles que requerirían si fueran transportados desde una ubicación lejana y tener más variedad de perecederos.

En los próximos años, la agricultura urbana dejará de ser una alternativa y se convertirá en una necesidad para hacer frente al crecimiento urbano, lo que trae consigo la creación de una rama más competitiva de la agricultura ya que también concierne el desarrollo económico de las ciudades. Esto trae como consecuencia la búsqueda de un uso más eficiente de los insumos agrícolas, formas de incrementar el valor de los productos,

conservación de bienes perecederos, un manejo más eficiente del agua, entre otras variables que pueden beneficiarse del uso de IoT para ser competitivos y vigentes.

## 2. Descripción General

El objetivo del proyecto es el diseño e implementación de un sistema de huerto urbano que permita mantener dentro de rangos específicos la temperatura, la luminosidad y la humedad que necesita la planta para crecer y de manera automática. Para el monitoreo de estas variables se deben almacenar en una base de datos que pueda accederse desde cualquier parte del mundo utilizando internet.

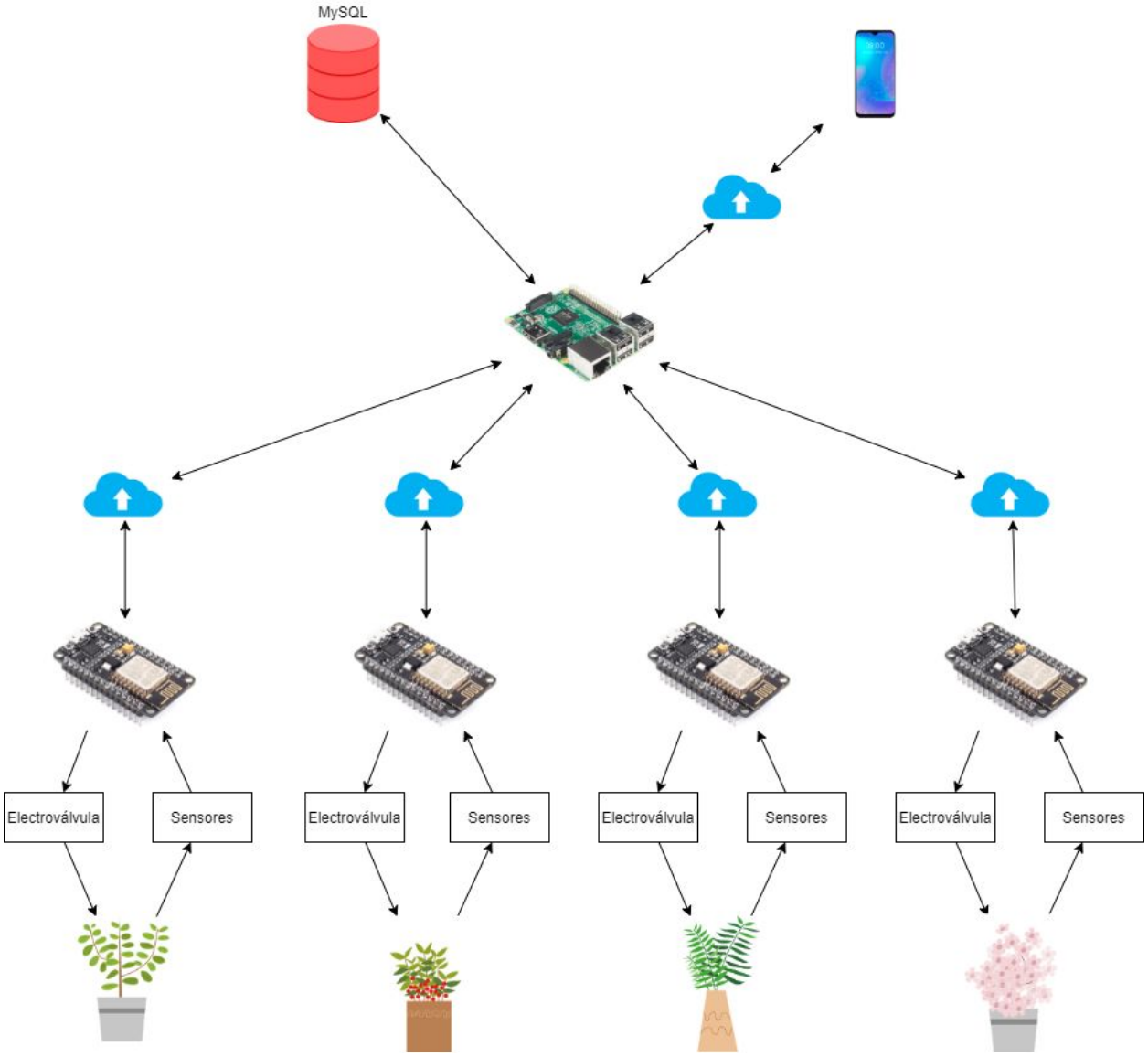
En lo que concierne al diseño de software, el sistema debe tener una estructura tipo estrella donde el hub principal será la computadora Raspberry en dónde se almacenará la base de datos MySQL. Los dispositivos NodeMCU comunicarán las mediciones de temperatura, luminosidad y humedad a través de la red a la Raspberry donde se conectarán a la base de datos. Originalmente, la Raspberry también contaría con su propio huerto y recibiría información de los sensores, pero ya que los huertos iban a estar al aire libre se optó por que no tuviera un huerto para procurar la integridad del dispositivo.

Para el servidor de la página donde se desplegarán las lecturas en tiempo real, la Raspberry se encargará de hostear el servidor APACHE. Dicha página debe desplegar los datos en un formato que sea pertinente (de fácil lectura y agradable) para el usuario con CCS. Adicionalmente, se debe crear otra página que pueda recibir instrucciones para el control de la electroválvula, específicamente debe realizar las siguientes acciones:

I. Configurar si la electroválvula se debe abrir de forma manual o automática cuando se alcance cierto nivel de humedad en la tierra
II. Mostrar un botón para la apertura manual de la electroválvula
III. Desplegar un indicador que marque el estado actual (abierto o cerrado) de la electroválvula

En lo que concierne al diseño de hardware, se hicieron varias modificaciones para procurar la integridad de los componentes más esenciales (como la Raspberry) y hacer frente a limitaciones físicas en los tanques de riego. La primera de ellas fue la sustitución del cultivo principal de la Raspberry por uno secundario, teniendo un total de 4 cultivos secundarios.

Inicialmente se consideró poner un tanque independiente debido a que se solicitaba el uso de un potenciómetro que desempeñaría la función de un flotador digital, pero la presión que se tenía no era suficiente para el riego óptimo de la planta por lo que se decidió como alternativa conectar las electroválvulas directactamente a la llave de paso.




Img 2.1 Esquema del sistema huertos

# 3. Metodología

## 3.1 Materiales

1x Raspberry

5x Node MCU 8266

1x 555 (empaquetado DIP)

1x dip switch de 8

1x metro de cable UTP

2x 74ls00

2x 74ls04

3x 74ls08

3x 74ls32

1x 74ls86

1x 74ls85

2x 74ls47

1x 74ls193

1x74ls74

1x 74ls76

1x 74LS174

1x 74LS279

2x display de 7 segmentos de ánodo común

1x LCD de 2x16

2x Protoboard grande

Paquete de 40 jumper macho hembra

Paquete de 40 jumper macho macho

4x Led rojo

4x Led verde

4x Led amarillo

8x Botón de presión (push button) de pata larga

16x resistencias de 1k ohm

8 x resistencia de 330 ohm

5x resistencia de 10k ohm

5x resistencia de 33k ohm

5x resistencia de 330k ohm

5x resistencia de 100k ohm

9x capacitores de 10 microfaradios a 10V ó 16V

1x Cargador de celular

1x módulo de 2 o 4 relevadores para arduino

2x termistor de 10k ohm

2x Sensor de humedad para arduino

1x potenciómetro de 10k ohm

2x metro de cable dúplex calibre 22 (POT22)

1x clavija

1x metro de cable UTP

1x electroválvula para lavadora, 127V 60Hz

1x metro de manguera del diámetro mayor de la electroválvula

1x metro de manguera del diámetro menor de la electroválvula

2x fotoresistencia

## 3.2 Herramientas

Multímetro

Alicate de corte

Pinzas de presión

Pinzas de punta

Pinzas pela cables

Cutter

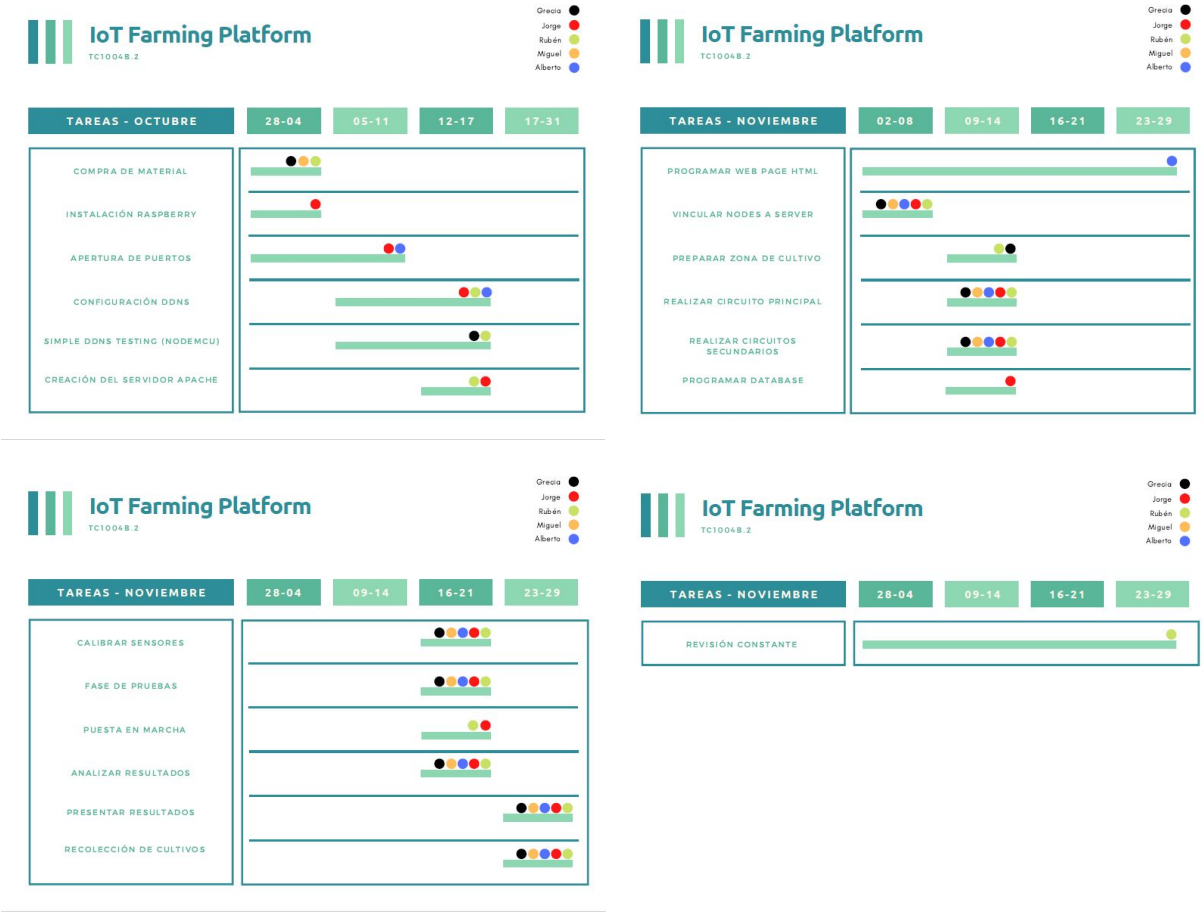
Pistola de silicón

Cautín, pasta de soldar y soldadura de estaño

Recipiente de plástico de 3 o 4 litros

Caja metálica o plástica de 20x20x20 cm

## 3.3 Cronograma de actividades




Img 3.3.1 Cronograma de actividades

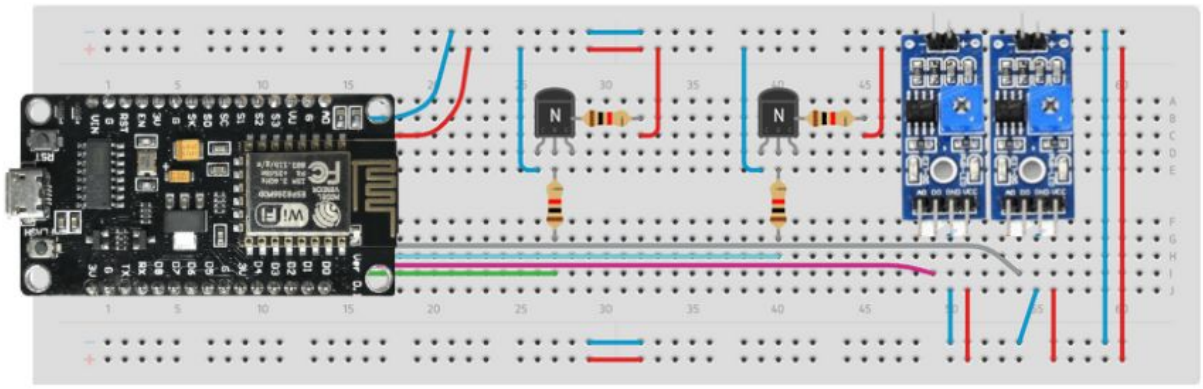
## 4. Hardware

Para la implementación física de los cuatro huertos se realizaron dos circuitos, cada uno de ellos se compone de dos nodeMCU, cada uno de los cuales estaba conectado a una protoboard completamente independiente con la finalidad de tener un mejor rendimiento del voltaje y corriente con el que estos se alimentaban.

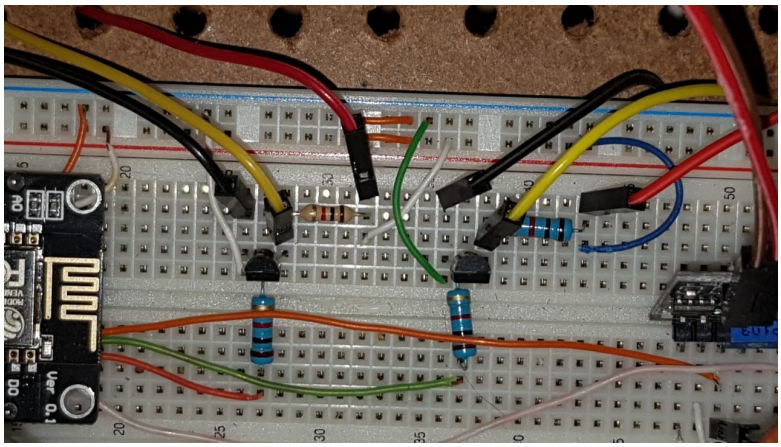
Uno de los dispositivos se encarga del control del riego de la planta a través de la recepción de los valores de los sensores de humedad y la activación de las electroválvulas; mientras que el otro recibía los valores análogos de temperatura y luminosidad y despliega a través de leds el estado de cada uno de los cultivos con el siguiente código de color.


Img 4.0.1 LEDS

En términos generales podemos visualizar el circuito de la siguiente manera:



Img 4.0.2 Circuito A



Img 4.0.3 Implementación del Circuito A

```
/// RECIBE MENSAJE
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == '1' && (strcmp(topic,"N3VAL_E")==0) ) {
    Serial.println("PRUEBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAX");
    client.publish("Prueba", "CIERRA N3VAL_E");
    digitalWrite(16, LOW);
    delay(2000);



    digitalWrite(BUILTIN_LED, LOW);   // Turn the LED on (Note that LOW is the voltage level
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
  }

    if (((char)payload[0] == '0') && (strcmp(topic,"N3VAL_E")==0) ){
      digitalWrite(BUILTIN_LED, HIGH);  // Turn the LED off by making the voltage HIGH
      client.publish("Prueba", "ABRE N3VAL_E");

      digitalWrite(16,  HIGH);
      digitalWrite(5,  LOW);
      delay(2000);
  }

  if (((char)payload[0] == '1') && (strcmp(topic,"N4VAL_E")==0) ) {

    client.publish("Prueba", "CIERRA N4VAL_E");
    digitalWrite(5, LOW);
```

```
void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  if(!digitalRead(humedad1)){
    Serial.println("Hay humedad1");
    digitalWrite(16,LOW);
    client.publish("N3Hum", "HU31");
    client.publish("N3VAL_S", "VS31");
    delay(2000);
   }
  client.loop();
  if(digitalRead(humedad1)){
      Serial.println("No Hay humedad1");


      digitalWrite(16,HIGH);
      client.publish("N3VAL_S", "VS31");
      client.publish("N3Hum", "HU30");
      delay(2000);
  }
  client.loop();

  if(digitalRead(humedad2)){
      Serial.println("No Hay humedad2");
      digitalWrite(5,HIGH);
      client.publish("N4VAL_S", "VS41");
      client.publish("N4Hum", "HU40");
      delay(2000);
  }
  client.loop();
  if(!digitalRead(humedad2)){
    Serial.println("Hay humedad2");
    digitalWrite(5,LOW);
    client.publish("N4VAL_S", "VS40");
    client.publish("N4Hum", "HU41");
    delay(2000);

  }
  client.loop();
}
```

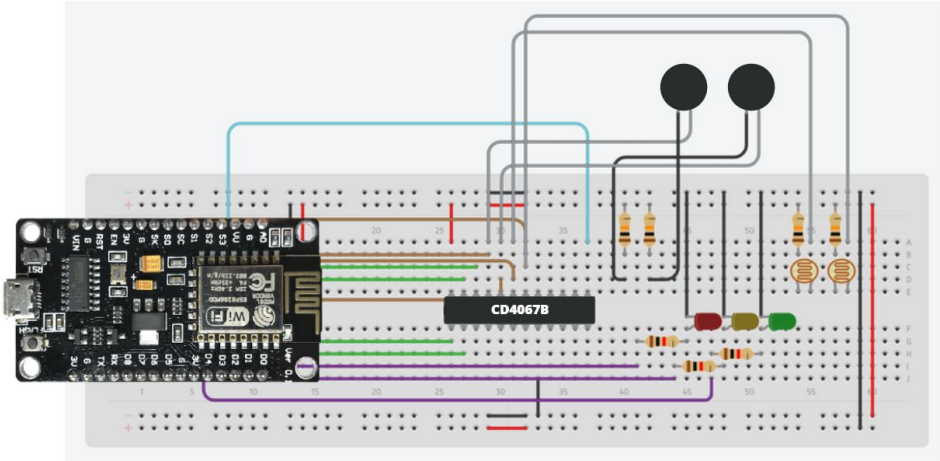
Img 4.0.4 Software Circuito A

En este podemos observar que en la primera parte se utilizaron dos transistores NPN 2222, uno por cada electroválvula a utilizar, esto debido a que los relevadores necesitan un voltaje de 5V para ser activados y el ESP8266 tiene como voltaje de salida alrededor de 3.3V, por lo cual era necesario aumentar de manera considerable el mismo.

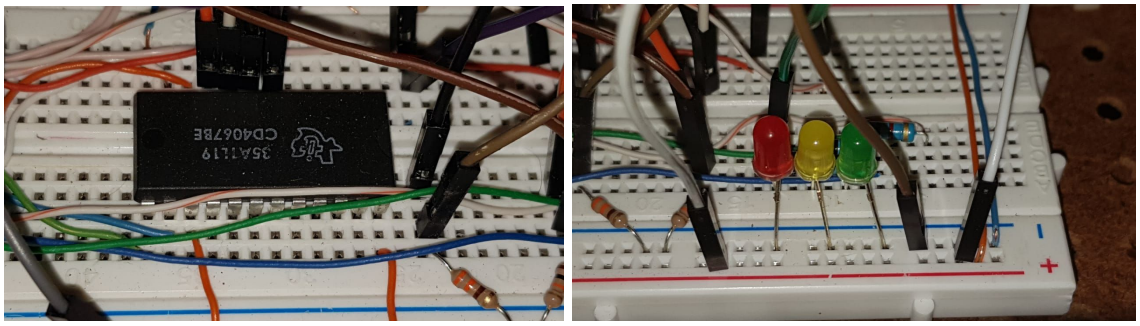
Dentro de este primer circuito se colocaron los sensores de humedad correspondientes a cada planta, los cuales mandan una señal digital a dos puertos diferentes del nodeMCU y a dos puertos del segundo como parte de las condiciones de estado.

A través de los puertos se D0, D1, D2 Y D3 se reciben y se mandan las señales necesarias para cumplir con el funcionamiento; los puertos 0 y 2 nos permiten la activación de los relevadores mientras que los puertos 1 y 3 se encargan de recibir una señal digital por parte de los sensores de humedad.

Por otro lado el segundo circuito se puede visualizar de la siguiente manera:




Img 4.0.5 Circuito B



Img 4.0.6 Implementación del Circuito B

El circuito B por su parte, se encarga de manejar la lógica del multiplexor analógico CD4067B, el cual realiza un barrido de todas las señales cada 4 segundos, leyendo a su paso en primera instancia los sensores de temperatura y en segundo lugar los sensores de luminosidad. Se tiene que el puerto A0 recibe la señal que va cambiando durante el ciclo, D0, D1, D2 y D3 actúan los seleccionadores del multiplexor, mientras que D4, D5, D6, D7 forman parte de las respuestas lógicas para las condiciones de los sensores de luminosidad y temperatura. Por último tenemos que D8 y D9 reciben la señales de humedad del circuito A.

A su vez, el Circuito B se encarga de manejar la lógica de los LEDs y Buzzers dada por la siguiente tabla de verdad previamente simplificada con Mapas de Karnaugh.

| A | B | C | | | | | |
|---|---|---|---|---|---|---|---|
| DIGITAL | ANALOG | ANALOG | = A | =AB'C' | =AC + AB | = A' | = A' |
| Humidity Sensor | Temperature Sensor | Light Sensor | REGAR | LED VERDE | LED AMARILLO | LED ROJO | BUZZER |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

Img 4.0.7 Circuito B

Para poder realizar la conversión de analógico a digital se programó un rango de valores desde el IDE, estos valores corresponden a los requerimientos mínimos para el desarrollo

de una planta de acuerdo a estándares internacionales. Por ejemplo, una vez que se superan los 30 Celsius, hay una alerta por la temperatura alta.

Sin embargo, como se puede observar en Img 4.0.4, el factor crítico es la humedad que ante su ausencia opera el LED rojo de emergencia y el Buzzer.

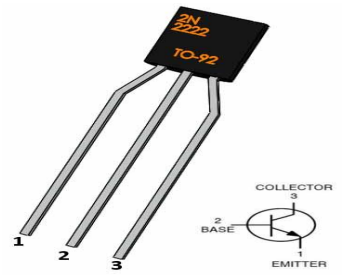
```
mux_v6
// Input pins
float temp1=0;
float temp2=0;
int luz1=0;
int luz2=0;

// for Bin 0 means it is okay
int temp1Bin = 0;
int temp2Bin = 0;
int luz1Bin = 0;
int luz2Bin = 0;

void setup_wifi() {

  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
void loop() {

  if (!client.connected()) {
    reconnect();
  }

  int data;

  for (byte i = 0; i < 4; i++) {
    data = mux.read(i);


    Serial.println("--------------------");
    Serial.print("Reading port no. ");
    Serial.println(i);
    Serial.print("Raw data:  ");
    Serial.println(data);

    if (i == 0) {
      temp1 = (0.07992*(data))-40.36;

      Serial.print("Temp1 (C):  ");
      Serial.println(temp1);

      String mqttVar = "TE1";
      String temp1Str = String(temp1, 0);
      mqttVar = mqttVar + temp1Str;
      client.publish("N1TE", mqttVar.c_str());
       Serial.println("Envie N1TE");

      if (temp1 > 20.0) { temp1Bin = 1;}
      else {temp1Bin = 0;}
```

Img 4.0.8 Software Circuito B

## 4.1 Conexiones

### 4.1.1 Transistores




| 2N2222A | |
|---|---|
| 1 | Emitter |
| 2 | Base |
| 3 | Collector |

Img 4.1.1 transistor

Para realizar las conexiones de los transistores es necesario conocer que estos se componen por tres pines principales que son: emisor base y colector, como se muestra en la imagen.

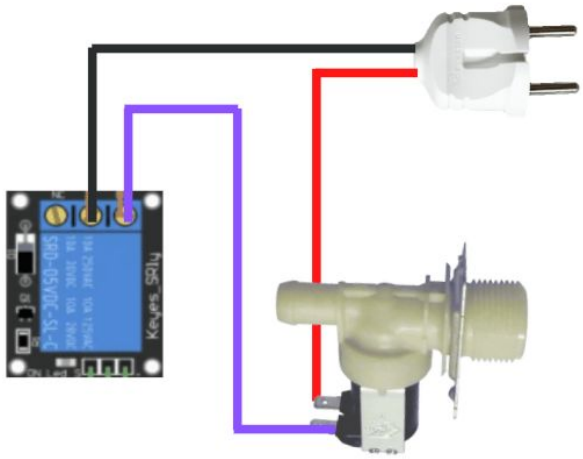
En el circuito, el emisor es conectado a tierra, la base se conectará a la señal mandada por el Node MCU a través de una resistencia de un 1kΩ, por último el collector se conectará a una resitencia de 1kΩ el voltaje general que serán los 5V con los que está electrificada la placa de pruebas.

A partir de el emisor se le brindará una señal de tierra al relevado, el colector es el encargado de alimentar con un voltaje de 5V al mismo para que este pueda ser activado de manera eficiente, y

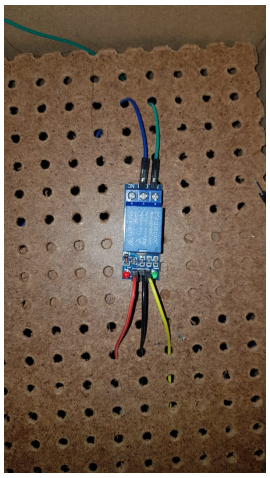
### 4.1.2 Electroválvula

Para poder manejar una electroválvula comercial fue necesario era elevar el voltaje a 125VAC, esto fue logrado con la ayuda del módulo relevador 125AC-220AC para arduino el cual contiene una pequeña bobina que induce cierto voltaje dependiendo si es administrada corriente o no, esta administración de corriente puede verse como una salida digital del NodeMCU ESP8266.

La siguiente imagen muestra de forma ilustrativa la conexión básica con una bombilla eléctrica.



Img 4.1.2 Conexión electroválvula-relevador

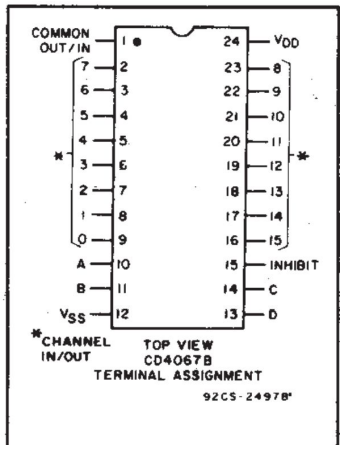


Img 4.1.3 Implementación

Para nuestro caso particular usamos la electroválvula que va conectada de la siguiente forma:

I. VCC Relevador -> 5V ESP8266
II. GND Relevador -> GND ESP8266
III. IN Relevador -> D1 ESP8266
IV. COMMON Relevador -> Clavija Extremo 1
V. NORMALMENTE OPEN Relevador -> Electroválvula
VI. Clavija Extremo 2 -> Electroválvula

### 4.1.2 Multiplexor

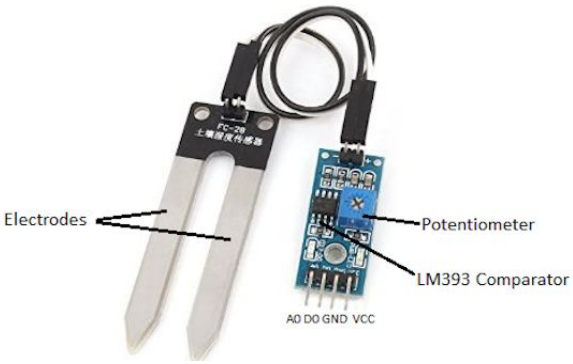



El multiplexor que se utilizó para la recepción de las señales analógicas brindadas por los sensores fue el CD4067, el cual cuenta con 16 pines de entrada, 4 seleccionadores, dos dedicados a la electrificación de la compuerta, un activador y, por último, un pin que dará la salida deseada; dicha composición se puede observar en la imagen 4.1.4.

Para poder hacer uso de este dispositivo el puerto 24 se conectará a voltaje, mientras que el 12 será tierra, los puertos marcados con las letras A,B,C,D se conocen como seleccionadores funcionan ; el puerto de activación en este caso es el número 15 y necesita recibir una señal de 0 por lo tanto se conectará a tierra.

Img 4.1.4 Multiplexor

### 4.1.2 Sensor de Humedad




Img 4.1.5 sensor de humedad

Se hizo uso de un sensor de humedad de suelo genérico para Arduino el cual mide la concentración de humedad de acuerdo a la conductividad eléctrica de la superficie donde esté situado.

Su conexión en el circuito A  queda de la siguiente manera:

VCC -> 5V ESP8266

GND -> GND ESP8266

D0-> D2 ESP8266

### 4.1.2 Termistores y Fotoresistencias


Img 4.1.6 sensor de humedad

Ambas entran dentro del espectro de las resistencias por lo tanto su funcionamiento primordial es limitar la corriente de acuerdo a ciertas variables física por un lado el

termistor dependiendo de la temperatura y por otro lado la fotoresistencia dependiendo de la cantidad de luz.

Su conexión en el circuito B queda de la siguiente manera:

TERMINAL 1 -> 5V ESP8266

TERMINAL 2 -> RESISTENCIA 10KOHMS -> GND ESP8266

-> INPUT SEÑAL.

## 4.2 Calibraciones

Existe un número importante de compañías que se dedican a la fabricación y comercialización de módulos y sensores para microcontroladores, por lo que es aconsejable que por cuenta propia calibre los sensores dependiendo sus condiciones características y especificaciones de la casa fabricadora. A continuación expondremos de forma general el procedimiento que llevamos a cabo.

### 4.2.1 Sensores de humedad de suelo

Para realizar la calibración de los sensores de humedad fue necesario directamente ajustar los parámetros con el potenciómetro incluido en el módulo.

Fue entonces que con la ayuda de un destornillador pequeño y dependiendo de la dirección se ajustaría la salida digital del dispositivo, funcionando este pequeño potenciómetro como un convertidor ADC.

### 4.2.2 Sensores de luminosidad (Fotoresistencias)

Para realizar una calibración correspondiente se registró la lectura del sensor en tres escenarios diferentes, el primero de ellos a plena luz del día, el segundo bajo la sombra y el último en el anochecer.

Se concluyó que el parámetro crítico para la salud de la planta era el exceso de luminosidad; por lo que la lectura a plena luz del día fue registrada para más adelante colocarla como condicional dentro del programa de riego.

| | |
|---|---|
| *Lectura del Sensor a plena luz del día (12:00 hrs)* | *1009* |
| Lectura del Sensor por el atardecer (17:00 hrs) | 550 |
| Lectura del Sensor en el anochecer (21:00 hrs) | 25 |

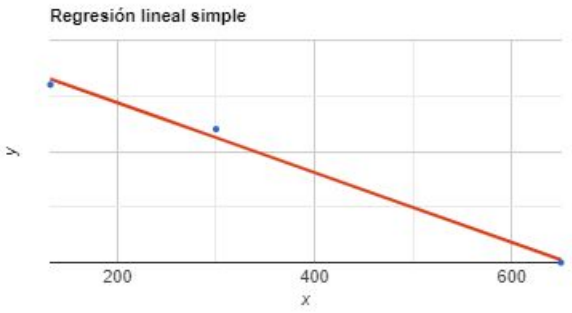
Tabla 4.2.2 Lecturas luminosidad

### 4.2.3 Sensores de temperatura (Termistores)

Para la correcta calibración de estos sensores fue necesario aplicar un modelo matemático de regresión lineal el cual tuvo una muestra de 3 datos, por un lado la variable dependiente era la temperatura en Celsius mientras que la variable independiente resultaría la lectura del sensor de temperatura.

| Lectura del termistor (X) | Temperatura en Celsius (Y) |
|---|---|
| 132 | 16° |
| 300 | 24° |
| 650 | 0° |

Tabla 4.2.3 Lecturas temperatura




**Ecuación resultante de la regresión lineal**

$$y = 41.320 - 0.063x$$

**Parámetros interesantes:**

$\bar{x} = 360.667$ (media de los datos en x)

$\bar{y} = 18.667$ (media de los datos en y)

$\sigma = 31.708$ (desviación estándar)

Img 4.2.3 calibración de sensores

# 5. Software

## 5.1 Código principal para recepción de datos en la base de datos MySQL con la Raspberry en Python.

```python
import time
import paho.mqtt.client as paho
from datetime import datetime
import mysql.connector
import json

def mysqlfunc(ide,estado,formatofh,tabla):
    conexion = mysql.connector.connect(user="root", password="root", database="iot")
    query = f"INSERT INTO {tabla} Values('{ide}','{estado}','{formatofh}') ;"
    statement = conexion.cursor()
    statement.execute(query)
    conexion.commit()
    statement.close()
    conexion.close()

broker="rubenruiz.hopto.org"
#define callback
def on_message(client, userdata, message):
    #TIEMPO
    global time1,TL, TC, VS1,VS2, VS3, VS4, H1, H2, H3, H4, L1, L2, L3, L4, T1, T2, T3, T4, dicc
    time2= time.time()
    if ((time2-time1)>=TIEMPO): TL=1
    #print("TIEMPO : ",(time2-time1))

    #TERMINA TIEMPO

    #time.sleep(1)
    message = str(message.payload.decode("utf-8"))
    #print("EL MENSAJE ES: ",message)
    if ((message)=="1" or (message)=="0"):
        pass

    else:

        ide = int(message[2])
        estado = int(message[3:])

        now = datetime.now()
        formatofh = now.strftime("%Y-%m-%d %H:%M:%S")
        #client.publish("PRASP", "ENTRA message")
        #"V": {"VS1": "0","VS2": "0","VS3": "0","VS4": "0"}
        if (message[0:2]=="VE"):
            dicc[f'{message[0]}'][f'VS{message[2]}']=f'{message[3:]}'
        else:
            dicc[f'{message[0]}'][f'{message[0:3]}']=f'{message[3:]}'


        with open('/var/www/html/RTD.json', 'w') as outfile:

            json.dump(dicc, outfile)

    #print("received message = "+message)
```

Img 5.1-1 Cliente MQTT Raspberry

```
elif (message[0:2]=="TE"):

    if (TL==1):
        if (T1==0 and message[2]=="1"):

            mysqlfunc(ide,estado,formatofh,"S_temperatura")

            T1 = 1
        if (T2==0 and message[2]=="3"):

            mysqlfunc(ide,estado,formatofh,"S_temperatura")

            T2 = 1
        if (T3==0 and message[2]=="3"):

            mysqlfunc(ide,estado,formatofh,"S_temperatura")

            T3 = 1
        if (T4==0 and message[2]=="4"):

            mysqlfunc(ide,estado,formatofh,"S_temperatura")

            T4 = 1
elif (message[0:2]=="LU"):

    if (TL==1):
        if (L1==0 and message[2]=="1"):

            mysqlfunc(ide,estado,formatofh,"S_iluminacion")

            L1 = 1
        if (L2==0 and message[2]=="3"):

            mysqlfunc(ide,estado,formatofh,"S_iluminacion")

            L2 = 1
        if (L3==0 and message[2]=="3"):

            mysqlfunc(ide,estado,formatofh,"S_iluminacion")

            L3 = 1
        if (L4==0 and message[2]=="4"):

            mysqlfunc(ide,estado,formatofh,"S_iluminacion")

            L4 = 1
```

Img 5.1-2 Cliente MQTT condiciones (mensaje)

```
TC = VS1+VS2+VS3+VS4+H1+H2+H3+H4+T1+T2+T3+T4+L1+L2+L3+L4

if (TC>=16 and TL==1):
    #print("ENTRA")
    VS1 = 0
    VS2 = 0
    VS3 = 0
    VS4 = 0
    H1 = 0
    H2 = 0
    H3 = 0
    H4 = 0
    T1 = 0
    T2 = 0
    T3 = 0
    T4 = 0
    L1 = 0
    L2 = 0
    L3 = 0
    L4 = 0

    TC = 0
    TL = 0
    time1 = time.time()
```

Img 5.1-3 Cliente MQTT Variables condicionales

```
TIEMPO = 1

time1 = time.time()


Node1_VE = "N1Val_E"
Node1_VS = "N1Val_S"
Node1_H = "N1Hum"
Node1_I = "N1Ilum"
Node1_I = "N1Ilum"
Node1_T = "N1TE"
VS1 = 0
H1 = 0
L1 = 0
T1 = 0


Node2_VE = "N2Val_E"
Node2_VS = "N2Val_S"
Node2_H = "N2Hum"
Node2_I = "N2Ilum"
Node2_T = "N2TE"

VS2 = 0
H2 = 0
L2 = 0
T2 = 0

Node3_VE = "N3Val_E"
Node3_VS = "N3Val_S"
Node3_H = "N3Hum"
Node3_I = "N3Ilum"
Node3_T = "N3TE"

VS3 = 0
H3 = 0
L3 = 0
T3 = 0

Node4_VE = "N4Val_E"
Node4_VS = "N4Val_S"
Node4_H = "N4Hum"
Node4_I = "N4Ilum"
Node4_T = "N4TE"

VS4 = 0
H4 = 0
L4 = 0
T4 = 0

dicc = {"V": {"VS1": "0","VS2": "0","VS3": "0","VS4": "0"} ,"H": {"HU1": " ","HU2": " ","HU3": " ","HU4": " "},"T": {"TE1": " ","TE2": " ","TE3": " ","TE4": " "},"L": {"LU1": " ","LU2": " ","LU3": " ","LU4": " "}}

TL=0
TC= 0
```

Img 5.1-4 Cliente MQTT Variables condicionales

```
client= paho.Client("RASPABABY")
######Bind function to callback
client.on_message=on_message
#####
print("connecting to broker ",broker)
client.connect(broker)#connect
#client.loop_start() #start loop to process received messages

client.subscribe(Node1_VS)#subscribe
client.subscribe(Node1_VE)
client.subscribe(Node1_H)
client.subscribe(Node1_I)
client.subscribe(Node1_T)

client.subscribe(Node2_VS)
client.subscribe(Node2_VE)
client.subscribe(Node2_H)
client.subscribe(Node2_I)
client.subscribe(Node2_T)

client.subscribe(Node3_VS)
client.subscribe(Node3_VE)
client.subscribe(Node3_H)
client.subscribe(Node3_I)
client.subscribe(Node3_T)


client.subscribe(Node4_VS)
client.subscribe(Node4_VE)
client.subscribe(Node4_H)
client.subscribe(Node4_I)
client.subscribe(Node4_T)


time1= time.time()

#while True:

    #time.sleep(1)

    #client.publish("PRASP","on")#publish
    #client.disconnect() #disconnect
client.loop_forever() #stop loop
```

Img 5.1-5 Cliente MQTT  Raspberry Subscribes

## 5.2 Estructura del sitio web

Para el desarrollo de la página web se utilizó el siguiente software y herramientas:

- PHP
- Python
- JSON
- JS
- HTML

### 5.2.1 Estructura de la Página

La página está definida con la siguiente estructura:

4 páginas principales:

- Index
- Válvulas
- Realtime Data
- Full Tables (Tablas Enteras)

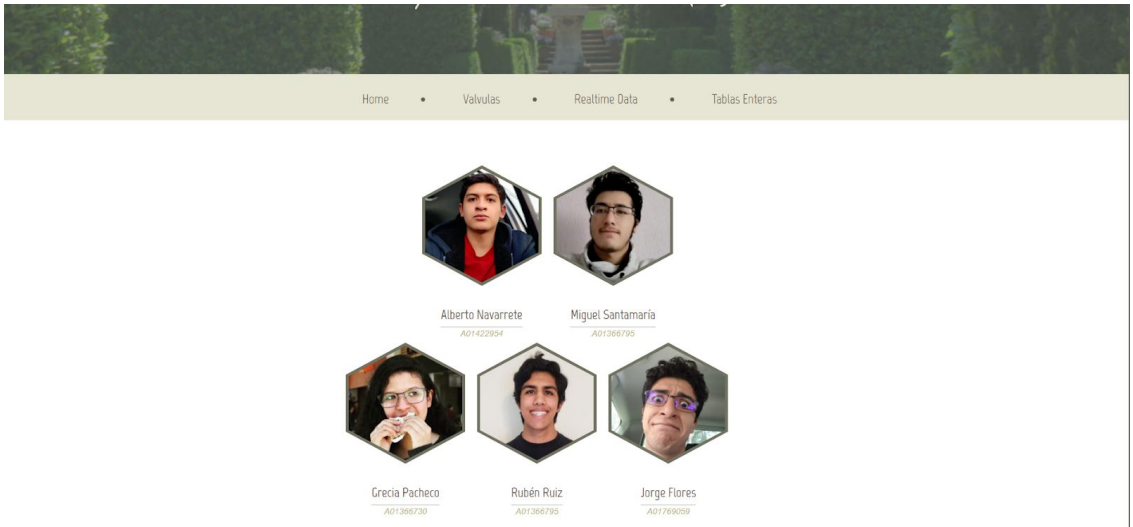
Elementos individuales:

- Header (Encabezado)
- Footer (Pie de página)

Cada página está estructurada de la siguiente manera:

### 5.2.2 Index

- Index.php
    - Libraries.php
    - Header.php
    - Index_body.php
    - Footer.php



### 5.2.3 Valvulas

- Valvulas.php
    - Libraries.php
    - Header.php
    - Valvulas_body.php
        - Botonespag.html
            - Botones.php
                - Botones.py
    - Footer.php

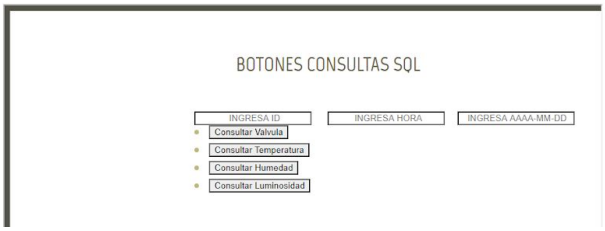


### 5.2.4 Realtime data

- Realtimedata.php
    - Libraries.php
    - Header.php
    - Realtimedata_body.php
        - Rtd.php (llama a RTD.json)
    - Footer.php


**5.2.5 Tablas Enteras**

- Fulltables.php
  - Libraries.php
  - Header.php
  - Fulltables_body.php
    - Bfulltables.html
      - Bfulltables.php
        - Bfulltables.py
  - Footer.php


## 5.3 Monitorización de variables analógicas o digitales con dispositivos NodeMCU y Raspberry

### 5.3.1 NodeMCU

### 5.3.2 Raspberry

#### 5.3.2.1 Información en tiempo real


Img 5.3.2 on message (CLiente MQTT Raspberry)

Para la monitorización de las variables analógicas o digitales del huerto, se utilizó programación en Python con las siguientes bibliotecas:

1.- Time

2.- Paho,mqtt.client

3.- Datetime

4.- Mysql.connector

5.- Json

La biblioteca de “MQTT” es fundamental para nuestro sistema, ya que es el método que utilizamos para poder enviar y recibir datos desde los NodeMCU a la Raspberry y viceversa.

Debido a la forma en la que la biblioteca y servicio “MQTT” funciona, el código está principalmente en la función que maneja la recepción de mensajes “on_message” y establecimos un formato para recepción de mensajes o “payload” el cual es de la siguiente manera:

Los primeros 2 caracteres definen el tipo de sensor/entrada de dato que va a recibir:

I. “VS” = Válvula Salida
II. “VE” = Válvula Entrada
III. “TE” = Temperatura Estado
IV. “LU” = Luminosidad

El siguiente caracter define el número de NodeMCU que lo está enviando

I. “1” = NodeMCU 1
II. “2” = NodeMCU 2
III. “3” = NodeMCU 3
IV. “4” = NodeMCU 4

Y los siguientes dígitos representan el valor que el NodeMCU devuelve del sensor o valor en cuestión. Definiéndose así los mensajes recibidos en formato:

| | | | |
|---|---|---|---|
| “VS1#” | “VS2#” | “VS3#” | “VS4#” |
| “VE1#” | “VE2#” | “VE3#” | “VE4#” |
| “TE1#” | “TE2#” | “TE3#” | “TE4#” |
| “LU1#” | “LU2#” | “LU3#” | “LU4#” |

Y se implementa un sistema para guardar los datos en la base de datos MySQL cada cierto tiempo del cual se explica más adelante

```
def on_message(client, userdata, message):
    #TIEMPO
    global time1,TL, TC, VS1,VS2, VS3, VS4, H1, H2, H3, H4, L1, L2, L3, L4, T1, T2, T3, T4, dicc
    time2= time.time()
    if ((time2-time1)>=TIEMPO): TL=1
    #print("TIEMPO : ",(time2-time1))

    #TERMINA TIEMPO

    #time.sleep(1)
    message = str(message.payload.decode("utf-8"))
    #print("EL MENSAJE ES: ",message)
    if ((message)=="1" or (message)=="0"):
        pass

    else:

        ide = int(message[2])
        estado = int(message[3:])

        now = datetime.now()
        formatofh = now.strftime("%Y-%m-%d %H:%M:%S")
```

Img 5.3.2-1 on message (CLiente MQTT Raspberry)

En este fragmento de código, se definen como globales los candados que definimos para nuestro algoritmo temporizado para insertar información en las bases de datos, obtenemos el mensaje que llega del tópico del MQTT, y empezamos a segmentar los mensajes que van llegando.

Para simplificar la entrada de datos a los NodeMCU, omitimos la nomenclatura definida arriba y mandamos un solo dígito. Es por ello, que fue añadida una excepción si el mensaje era igual a “1” o igual a “0”; si esto no fuera así, segmentamos el mensaje en la nomenclatura establecida,los primeros 2 caracteres del mensaje recibido, es el tipo de sensor o dato que se recibe, el tercer caracter del mensaje recibido será el identificador del NodeMCU que mandó el mensaje y los consiguientes es el estado del sensor/dato que se obtuvo lectura.

Después de segmentar el mensaje recibido, se obtiene la fecha y hora del sistema para poder insertar los datos correspondientes a la base de datos con el formato establecido de cada tabla.

Posteriormente,se establece nuevamente una excepción relacionada a la de arriba, para cambiar el diccionario de datos a tiempo real que se va a estar consultando constantemente para mostrar en la página web.

```
        if (message[0:2]=="VE"):
            dicc[f'{message[0]}'][f'VS{message[2]}']=f'{message[3:]}'
        else:
            dicc[f'{message[0]}'][f'{message[0:3]}']=f'{message[3:]}'
```

Img 5.3.2-2 on message (CLiente MQTT Raspberry)

En este fragmento de código, se observa la excepción que contiene el proceso para actualizar el diccionario que se define al inicio del código en el siguiente fragmento:

```
 dicc = {"V": {"VS1": "0","VS2": "0","VS3": "0","VS4": "0"} ,"H": {"HU1": " ","HU2": " ","HU3": " ","HU4": " "},
"T": {"TE1": " ","TE2": " ","TE3": " ","TE4": " "},"L": {"LU1": " ","LU2": " ","LU3": " ","LU4": " "}}
```

Img 5.3.2-3 on message (CLiente MQTT Raspberry)-diccionario

Y el en fragmento siguiente, una vez actualizado el diccionario, se dumpea en un archivo de formato .json para su lectura con php

```
        with open('/var/www/html/RTD.json', 'w') as outfile:

            json.dump(dicc, outfile)
```

Img 5.3.2-4 on message (CLiente MQTT Raspberry)-write,json

A continuación se explicará el código php junto con el html que hace posible el despliegue de los datos en tiempo real:

```php
<?php


$archivo= file_get_contents("RTD.json");
$json_a = json_decode($archivo, true);

$V1= $json_a['V']['VS1'];
$V2= $json_a['V']['VS2'];
$V3= $json_a['V']['VS3'];
$V4= $json_a['V']['VS4'];

$H1= $json_a['H']['HU1'];
$H2= $json_a['H']['HU2'];
$H3= $json_a['H']['HU3'];
$H4= $json_a['H']['HU4'];

$T1= $json_a['T']['TE1'];
$T2= $json_a['T']['TE2'];
$T3= $json_a['T']['TE3'];
$T4= $json_a['T']['TE4'];

$L1= $json_a['L']['LU1'];
$L2= $json_a['L']['LU2'];
$L3= $json_a['L']['LU3'];
$L4= $json_a['L']['LU4'];

?>
```

Img 5.3.2-5 on message (CLiente MQTT Raspberry)-json

En este segmento de código, se define dentro de la variable “archivo” el contenido que generó previamente el programa de python en formato json.

Se decodifica el archivo json y se traduce para que php lo pueda leer de forma correcta, y se guardan en distintas variables los datos correspondientes del diccionario que contiene los estados de las válvulas y de los sensores.

Y luego, se despliegan los datos en el sitio web con HTML como se muestra en el siguiente fragmento.

```
<div class="container_12">
    <div class="grid_12">

        <div class="slogan">Valvula 1 </div>
        <div class="slogan">Estado de la valvula <input type="text" size="50" value="<?php echo $V1?>" readonly></div>
        <div class="slogan">Humedad Valvula <input type="text" size="50" value="<?php echo $H1?>" readonly></div>
        <div class="slogan">Tempertura Valvula <input type="text" size="50" value="<?php echo $T1?>" readonly></div>
        <div class="slogan">Luminosidad Valvula <input type="text" size="50" value="<?php echo $L1?>" readonly></div>

    </div>
    <div>.</div>
    <div>.</div>
    <div>.</div>
    <div class="grid_12">
        <div class="slogan">Valvula 2 </div>
        <div class="slogan">Estado de la valvula <input type="text" size="50" value="<?php echo $V2?>" readonly></div>
        <div class="slogan">Humedad Valvula <input type="text" size="50" value="<?php echo $H2?>" readonly></div>
        <div class="slogan">Tempertura Valvula <input type="text" size="50" value="<?php echo $T2?>" readonly></div>
        <div class="slogan">Luminosidad Valvula <input type="text" size="50" value="<?php echo $L2?>" readonly></div>

    </div>
    <div>.</div>
    <div>.</div>
    <div>.</div>
    <div class="grid_12">
        <div class="slogan">Valvula 3 </div>
        <div class="slogan">Estado de la valvula <input type="text" size="50" value="<?php echo $V3?>" readonly></div>
        <div class="slogan">Humedad Valvula <input type="text" size="50" value="<?php echo $H3?>" readonly></div>
        <div class="slogan">Tempertura Valvula <input type="text" size="50" value="<?php echo $T3?>" readonly></div>
        <div class="slogan">Luminosidad Valvula <input type="text" size="50" value="<?php echo $L3?>" readonly></div>

    </div>
    <div>.</div>
    <div>.</div>
    <div>.</div>
    <div class="grid_12">
        <div class="slogan">Valvula 4 </div>
        <div class="slogan">Estado de la valvula <input type="text" size="50" value="<?php echo $V4?>" readonly></div>
        <div class="slogan">Humedad Valvula <input type="text" size="50" value="<?php echo $H4?>" readonly></div>
        <div class="slogan">Tempertura Valvula <input type="text" size="50" value="<?php echo $T4?>" readonly></div>
        <div class="slogan">Luminosidad Valvula <input type="text" size="50" value="<?php echo $L4?>" readonly></div>

    </div>
</div>
```

Img 5.3.2-6 pagina

## 5.4 Puesta en marcha de un servidor APACHE Tomcat y programación con PHP

Para poner en marcha el servidor APACHE en la Raspberry se debe instalar (montar) el servidor apache en la Raspberry con los respectivos comandos en la terminal, además de configurar el firewall e instalar PHP dentro del mismo Apache. Para comprobar que el apache estaba completamente configurado e instalado, ingresamos al dominio principal y la pagina esta hosteada y con acceso a la misma en:

**http://smartfarmingiot.ddns.net/**

Todas las páginas tienen la estructura previamente mostrada y el código se implementa de la siguiente manera:

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
<link rel="icon" href="images/favicon.ico">
<link rel="shortcut icon" href="images/favicon.ico" />
<link rel="stylesheet" href="css/owl.carousel.css">
<link rel="stylesheet" href="css/style.css">
<script src="js/jquery.js"></script>
<script src="js/jquery-migrate-1.1.1.js"></script>
<script src="js/script.js"></script>
<script src="js/jquery.ui.totop.js"></script>
<script src="js/superfish.js"></script>
<script src="js/sForm.js"></script>
<script src="js/jquery.equalheights.js"></script>

<script src="js/jquery.easing.1.3.js"></script>
<script src="js/owl.carousel.js"></script>
<script>
$(document).ready(function(){
    $().UItoTop({ easingType: 'easeOutQuart' });
    /*carousel*/
    var owl=$("#owl");
    owl.owlCarousel({
        items : 1, //10 items above 1000px browser width
        navigation : true,
        pagination :  false
    });
    var owl=$("#owl1");
    owl.owlCarousel({
        items : 1, //10 items above 1000px browser width
        navigation : true,
        pagination :  false
    });
})
</script>
<title> IoT Smart Farm</title>
```

Img 5.4-1 libraries.php(Bibliotecas,  referencias y scripts CSS, JS  (En todas las páginas al inicio))

## Header.php (En todas las páginas al inicio)

```
<html>
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
    <header>
            <div class="container_12">
                <div class="grid_12">
                    <h1>
                        <a href="index.php">
                            <img src="images/logo.png" alt="IoT">
                        </a>
                    </h1>
                    <div class="slogan">Implementación de Internet de las Cosas (IoT)</div>
                </div>
            </div>
            <div class="clear"></div>
            <div class="menu_block">
                <div class="container_12" style="text-align:center;">
                    <div class="grid_12">
                        <nav class="horizontal-nav full-width horizontalNav-notprocessed">
                            <ul class="sf-menu" style="text-align:center;">
                                <li><a href="index.php">Home</a></li>
                                <li><a href="valvulas.php">Valvulas</a></li>
                                <li><a href="realtimedata.php">Realtime Data</a></li>
                                <li><a href="fulltables.php">Tablas Enteras </a></li>

                            </ul>
                        </nav>
                        <div class="clear"></div>
                    </div>
                    <div class="clear"></div>
                </div>
            </div>
        </header>

</html>
```

Img 5.4-2 libraries.php(referencias a las páginas php)

## Footer.php (En todas las páginas al final)

```
<div class="container_12">
    <div class="grid_12 ">
        <div class="copy">
            IoT Internet de las Cosas; Tecnológico de Monterrey (C) <span id="copyright-year"></span>  |  | Diseño parcialmente de TemplateMonster
        </div>
    </div>
    <div class="clear"></div>
</div>
```

Img 5.4-3 footer.php(footer de pagina)

## Index.php

```php
<html>
    <head>
    <?php include('libraries.php'); ?>
        <?php include('header.php'); ?>
    </head>
    <body class="page1">
        <?php include('index_body.php'); ?>


    </body>
    <footer>
        <?php include('footer.php'); ?>
    </footer>



</html>
```

Img 5.4-4 libraries.php(footer de página)

## Index_body.php

```php
<div  class="content">
            <div style="width: 100% text-align: center;" class="container_12">
            <div style="margin-left: 39%" class="grid_3" >
                    <div class="block1">
                        <div class="img_block">
                            <img src="images/anvrrt.jpg" alt="" class="img_inner">
                            <span class="l"></span>
                            <span class="ll"></span>
                            <span class="r"></span>
                            <span class="rr"></span>
                            <span class="lb"></span>
                            <span class="llb"></span>
                            <span class="rb"></span>
                            <span class="rrb"></span>
                        </div>
                        <div class="text1">Alberto Navarrete</div><br>
                        <i class="col1">A01422954</i>
                    </div>
                </div>
            </div>
            <div style="width: 100% text-align: center;" class="container_12">
                <div style="display: inline-block;" class="grid_3">
                    <div class="block1">
                        <div class="img_block">
                            <img src="images/mike.jpg" alt="" class="img_inner">
                            <span class="l"></span>
                            <span class="ll"></span>
                            <span class="r"></span>
                            <span class="rr"></span>
                            <span class="lb"></span>
                            <span class="llb"></span>
                            <span class="rb"></span>
                            <span class="rrb"></span>
                        </div>
                        <div class="text1">Miguel Santamaría</div><br>
                        <i class="col1">A01366795</i>
                    </div>
                </div>
```

```
                <div class="grid_3">
                    <div class="block1">
                        <div class="img_block">
                            <img src="images/grecia.jpg" alt="" class="img_inner">
                            <span class="l"></span>
                            <span class="ll"></span>
                            <span class="r"></span>
                            <span class="rr"></span>
                            <span class="lb"></span>
                            <span class="llb"></span>
                            <span class="rb"></span>
                            <span class="rrb"></span>
                        </div>
                        <div class="text1">Grecia Pacheco</div><br>
                        <i class="col1">A01366730</i>
                    </div>
                </div>
                <div class="grid_3">
                    <div class="block1">
                        <div class="img_block">
                            <img src="images/rubs.jpg" alt="" class="img_inner">
                            <span class="l"></span>
                            <span class="ll"></span>
                            <span class="r"></span>
                            <span class="rr"></span>
                            <span class="lb"></span>
                            <span class="llb"></span>
                            <span class="rb"></span>
                            <span class="rrb"></span>
                        </div>
                        <div class="text1">Rubén Ruiz</div><br>
                        <i class="col1">A01366795</i>
                    </div>
                </div>
                <div class="grid_3">
                    <div class="block1">
                        <div class="img_block">
                            <img src="images/yorch.jpg" alt="" class="img_inner">
                            <span class="l"></span>
                            <span class="ll"></span>
                            <span class="r"></span>
                            <span class="rr"></span>
                            <span class="lb"></span>
                            <span class="llb"></span>
                            <span class="rb"></span>
                            <span class="rrb"></span>
                        </div>
                        <div class="text1">Jorge Flores</div><br>
                        <i class="col1">A01769059</i>
                    </div>
                </div>

                <div class="clear"></div>


        </div>
</div>
```

Img 5.4-5 Index.php(pagina principal)

## Valvulas.php

```php
<html>
    <head>
        <?php include('libraries.php'); ?>
        <?php include('header.php'); ?>
    </head>
    <body class="page1" >

        <?php include('valvulas_body.php'); ?>

        <div style="margin: 80px auto;" class="drop"></div>
        <div style="margin: 80px auto;" class="wave"></div>

    </body>
    <footer>
        <?php include('footer.php'); ?>
    </footer>



</html>
```

Img 5.4-6 valvulass.php(margenes)

## Valvulas_body.php

```php
<div  class="content">
        <script>
        window.setInterval("reloadIFrame();", 5000);
        function reloadIFrame() {
         document.getElementById("myIframe").src="botonespag.html";
        }
        </script>
        <div style="width: 100%; text-align: center;" class="container_12">

            <div id="result"></div>
            <iframe id="myIframe"  src="botonespag.html" style="overflow: hidden; height: 500px;width:900px; auto" title="Prueba"></iframe>

            <div class="clear"></div>

    </div>
</div>
```

Img 5.4-7 valvulas.php(margenes 2)

## botonespag.html

```html
<html>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
    <link rel="icon" href="images/favicon.ico">
    <link rel="shortcut icon" href="images/favicon.ico" />
    <link rel="stylesheet" href="css/owl.carousel.css">
    <link rel="stylesheet" href="css/style.css">
    <script src="js/jquery.js"></script>
    <script src="js/jquery-migrate-1.1.1.js"></script>
    <script src="js/script.js"></script>
    <script src="js/jquery.ui.totop.js"></script>
    <script src="js/superfish.js"></script>
    <script src="js/sForm.js"></script>
    <script src="js/jquery.equalheights.js"></script>

    <script src="js/jquery.easing.1.3.js"></script>
    <head>
    </head>
        <div class="content" >
        <h2 style="text-align: center;">BOTONES PARA ACTIVAR VALVULAS</h2>
            <form action="botones.php" method="get">
                <input type="hidden" name="IB1" value="1">
                <div style="margin-left: 29%">
                    <ul class="list">
                        <li><a href="#">Valvula 1.........:</a><input type="submit" name="boton" value="boton1"></li>
                        <li><a href="#">Valvula 2.........:</a><input type="submit" name="boton" value="boton2"></li>
                        <li><a href="#">Valvula 3.........:</a><input type="submit" name="boton" value="boton3"></li>
                        <li><a href="#">Valvula 4.........:</a><input type="submit" name="boton" value="boton4"></li>
                    </ul>
                </div>
            </form>
        </div>
</html>
```

Img 5.4-7(botones)

## Botones.php

```php
<?php
//$_POST['IB1'];
$V = $_REQUEST['IB1'];
//echo $_POST['boton'];
$B = $_REQUEST['boton'];

system("python3 Botones.py '$V' '$B'");

?>
```

Img 5.4-8 botones 2( requests)

## Botones.py

```python
#!/usr/bin/env python3.7
import time
import paho.mqtt.client as paho
import sys

class ClienteController:
    def procesarTransaccion(self):
        if(B == "boton1"):
            client= paho.Client("Boton1")
            client.connect(broker)#connect
            client.publish(Node1_VE,"0")#publish
            client.publish(Node1_VE,"VE11")
            time.sleep(1)
            client.publish(Node1_VE,"1")
            client.publish(Node1_VE,"VE10")
            client.disconnect() #disconnect
            #AQUI MANDA MOSQUITO
        elif (B == "boton2"):
            client= paho.Client("Boton2")
            client.connect(broker)#connect
            client.publish(Node2_VE,"0")#publish
            client.publish(Node2_VE,"VE21")
            time.sleep(1)
            client.publish(Node2_VE,"1")
            client.publish(Node2_VE,"VE20")
            client.disconnect() #disconnect
            #AQUI MANDA MOSQUITO
        elif (B == "boton3"):
            client= paho.Client("Boton3")
            client.connect(broker)#connect
            client.publish(Node3_VE,"0")#publish
            client.publish(Node3_VE,"VE31")
            time.sleep(1)
            client.publish(Node3_VE,"1")
            time.sleep(0.3)
            client.publish(Node3_VE,"VE30")
            client.disconnect() #disconnect
            #AQUI MANDA MOSQUITO
        elif (B == "boton4"):
            client= paho.Client("Boton4")
            client.connect(broker)#connect
            client.publish(Node4_VE,"0")#publish
            client.publish(Node4_VE,"VE41")
            time.sleep(1)
            client.publish(Node4_VE,"1")
            client.publish(Node4_VE,"VE40")
            client.disconnect() #disconnect
            #AQUI MANDA MOSQUITO
```

Img 5.4-9 botones 2( conexiones)

```
# Crear el objeto de la clase Primero
broker="rubenruiz.hopto.org"
Node1_VS = "N1Val_S"
Node2_VS = "N2Val_S"
Node3_VS = "N3Val_S"
Node4_VS = "N4Val_S"


Node1_VE = "N1Val_E"
Node2_VE = "N2Val_E"
Node3_VE = "N3Val_E"
Node4_VE = "N4Val_E"


V = sys.argv[1]
B = sys.argv[2]


cliente = ClienteController()
#cliente.respuestaServer2()
cliente.procesarTransaccion()
```

Img 5.4-10 mqtt 2(lógica de mensajes)

En esta parte se encuentran las respuestas dependiendo del botón que fue presionado, utilizando MQTT se publican los diferente mensajes para que se prenda y apague la electroválvula, el mensaje de la válvula 1 de prendido sería VE11 y de apagado sería VE10. Esto se hace en intervalos de 1 segundo para no gastar agua o ahogar la planta

## Realtimedata.php

```
<html>
    <head>
        <?php include('libraries.php'); ?>
        <?php include('header.php'); ?>
    </head>
    <body class="page1">

        <?php include('realtimedata_body.php'); ?>


    </body>
    <footer>
        <?php include('footer.php'); ?>
    </footer>



</html>
```

Img 5.4-11 rtd(include)

## Realtimedata_body.php

```
<div  class="content">
            <script>
            window.setInterval("reloadIFrame();", 1000);
            function reloadIFrame() {
             document.getElementById("myIframe").src="rtd.php";
            }
            </script>
            <div style="width: 100% text-align: center;" class="container_12">

                <div id="result"></div>
                <iframe id="myIframe"  src="rtd.php" style="height: 1000px; width:1000px;" title="Prueba"></iframe>
                <div class="clear"></div>

            </div>
</div>
```

Img 5.4-12 Realtimedata_body

rtd.php

```php
<?php

$archivo= file_get_contents("RTD.json");
$json_a = json_decode($archivo, true);


//dicc = {'V': {'VS1': ' ','VS2': ' ','VS3': ' ','VS4': ' '} ,
//'H': {'HU1': ' ','HU2': ' ','HU3': ' ','HU4': ' '},
//'T': {'TE1': ' ','TE2': ' ','TE3': ' ','TE4': ' '},
//'L': {'LU1': ' ','LU2': ' ','LU3': ' ','LU4': ' '}}

$V1= $json_a['V']['VS1'];
$V2= $json_a['V']['VS2'];
$V3= $json_a['V']['VS3'];
$V4= $json_a['V']['VS4'];

$H1= $json_a['H']['HU1'];
$H2= $json_a['H']['HU2'];
$H3= $json_a['H']['HU3'];
$H4= $json_a['H']['HU4'];

$T1= $json_a['T']['TE1'];
$T2= $json_a['T']['TE2'];
$T3= $json_a['T']['TE3'];
$T4= $json_a['T']['TE4'];

$L1= $json_a['L']['LU1'];
$L2= $json_a['L']['LU2'];
$L3= $json_a['L']['LU3'];
$L4= $json_a['L']['LU4'];

if($V1 == 1) $V1 = "Abierto";
if($V1 == 0) $V1 = "Cerrado";
if($H1 == 1) $H1 = "Hay humedad";
if($H1 == 0) $H1 = "No hay humedad";

if($V2 == 1) $V2 = "Abierto";
if($V2 == 0) $V2 = "Cerrado";
if($H2 == 1) $H2 = "Hay humedad";
if($H2 == 0) $H2 = "No hay humedad";

if($V3 == 1) $V3 = "Abierto";
if($V3 == 0) $V3 = "Cerrado";
if($H3 == 1) $H3 = "Hay humedad";
if($H3 == 0) $H3 = "No hay humedad";

if($V4 == 1) $V4 = "Abierto";
if($V4 == 0) $V4 = "Cerrado";
if($H4 == 1) $H4 = "Hay humedad";
if($H4 == 0) $H4 = "No hay humedad";

?>
```

Img 5.4-13 rtd(variables)

```php
<?php include('libraries.php'); ?>
<div class="container_12">
    <div class="grid_12">

        <div class="slogan2">Huerto 1 </div>
        <div class="slogan2">Estado de la valvula <input type="text" size="50" value="<?php echo $V1?>" readonly></div>
        <div class="slogan2">Humedad <input type="text" size="50" value="<?php echo $H1?>" readonly></div>
        <div class="slogan2">Tempertura Valvula <input type="text" size="50" value="<?php echo $T1?>" readonly></div>
        <div class="slogan2">Luminosidad Valvula <input type="text" size="50" value="<?php echo $L1?>" readonly></div>

    </div>

    <div class="grid_12">
        <div class="slogan2">Huerto 2 </div>
        <div class="slogan2">Estado de la valvula <input type="text" size="50" value="<?php echo $V2?>" readonly></div>
        <div class="slogan2">Humedad <input type="text" size="50" value="<?php echo $H2?>" readonly></div>
        <div class="slogan2">Tempertura <input type="text" size="50" value="<?php echo $T2?>" readonly></div>
        <div class="slogan2">Luminosidad <input type="text" size="50" value="<?php echo $L2?>" readonly></div>

    </div>

    <div class="grid_12">
        <div class="slogan2">Huerto 3 </div>
        <div class="slogan2">Estado de la valvula <input type="text" size="50" value="<?php echo $V3?>" readonly></div>
        <div class="slogan2">Humedad <input type="text" size="50" value="<?php echo $H3?>" readonly></div>
        <div class="slogan2">Tempertura <input type="text" size="50" value="<?php echo $T3?>" readonly></div>
        <div class="slogan2">Luminosidad <input type="text" size="50" value="<?php echo $L3?>" readonly></div>

    </div>

    <div class="grid_12">
        <div class="slogan2">Huerto 4 </div>
        <div class="slogan2">Estado de la valvula <input type="text" size="50" value="<?php echo $V4?>" readonly></div>
        <div class="slogan2">Humedad <input type="text" size="50" value="<?php echo $H4?>" readonly></div>
        <div class="slogan2">Tempertura <input type="text" size="50" value="<?php echo $T4?>" readonly></div>
        <div class="slogan2">Luminosidad <input type="text" size="50" value="<?php echo $L4?>" readonly></div>

    </div>
</div>
```

Img 5.4-14 botones(setup)

## Fulltables.php

```php
<html>
    <head>
        <?php include('libraries.php'); ?>
        <?php include('header.php'); ?>
    </head>
    <body class="page1" >

        <?php include('fulltables_body.php'); ?>



    </body>
    <footer>
        <?php include('footer.php'); ?>
    </footer>



</html>
```

Img 5.4-15 tables(full tables)

## Fulltables_body.php

```php
<div  class="content">

        <div style="width: 100%; text-align: center;" class="container_12">

            <div id="result"></div>
            <iframe id="myIframe"  src="bfulltables.html" style="overflow: hidden; height: 500px;width:900px; auto" title="Prueba"></iframe>

            <div class="clear"></div>

        </div>
</div>
```

Img 5.4-16 tables(full tables)

## Bfulltables.html

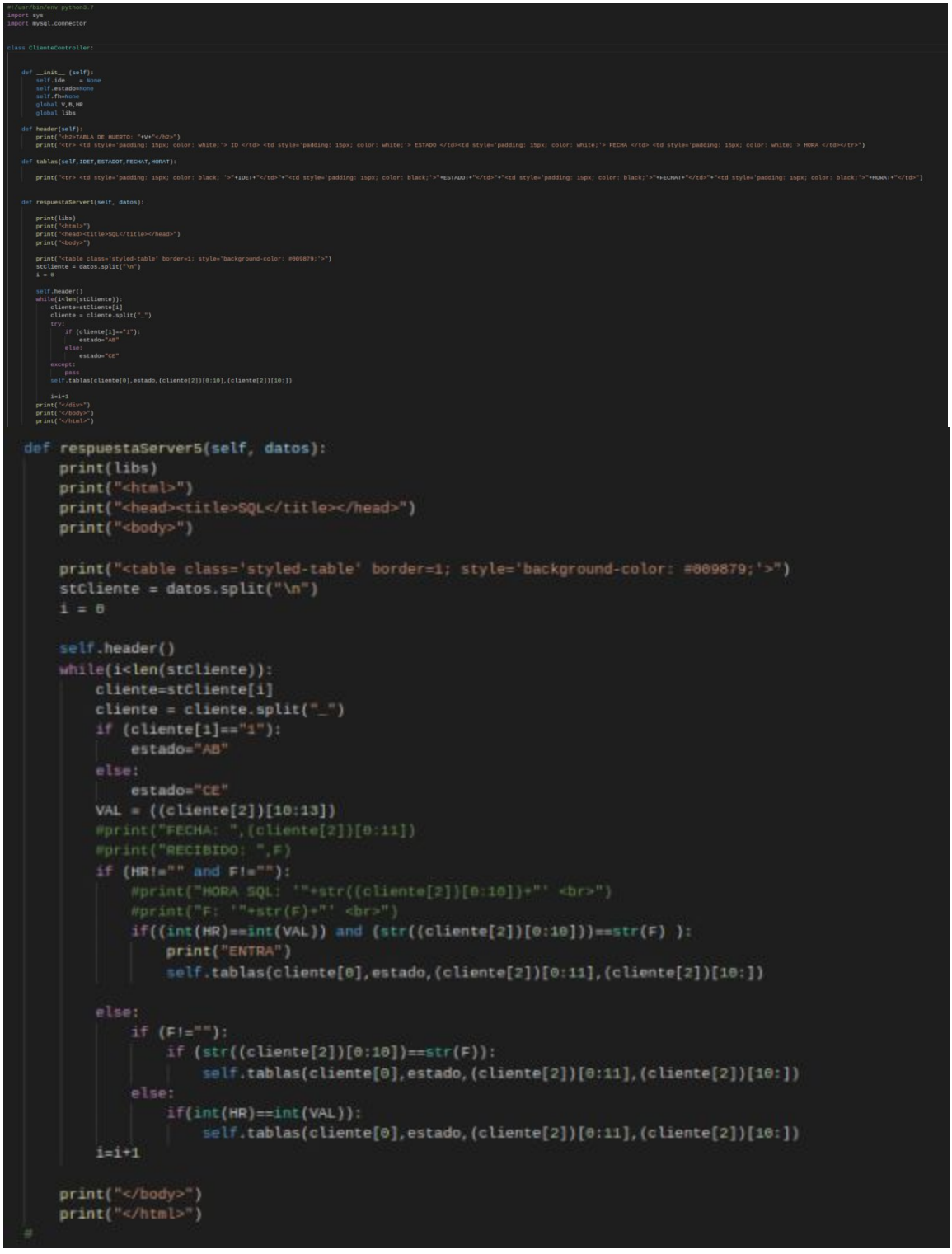

Img 5.4-17 botones(request)

## Bfulltables.php

```php
<?php

$V = $_REQUEST['ide'];

$H = $_REQUEST['Hora'];
$F = $_REQUEST['Fecha'];

$B = $_REQUEST['boton'];

system("python3 bfulltables.py '$V' '$H' '$F' '$B'");

?>
```

Img 5.4-18 botones(request)

## Bfulltables.py



```
#!/usr/bin/env python3.7
import sys
import mysql.connector

class ClienteController:


    def __init__ (self):
        self.ide    = None
        self.estado=None
        self.fh=None
        global V,B,HR
        global libs

    def header(self):
        print("<h2>TABLA DE HUESTO: "+v+"</h2>")
        print("<tr> <td style='padding: 15px; color: white;'> ID </td> <td style='padding: 15px; color: white;'> ESTADO </td><td style='padding: 15px; color: white;'> FECHA </td> <td style='padding: 15px; color: white;'> HORA </td></tr>")

    def tablas(self,IDET,ESTADOT,FECHAT,HORAT):

        print("<tr> <td style='padding: 15px; color: black; '>"+IDET+"</td>"+"<td style='padding: 15px; color: black;'>"+ESTADOT+"</td>"+"<td style='padding: 15px; color: black;'>"+FECHAT+"</td>"+"<td style='padding: 15px; color: black;'>"+HORAT+"</td>")


    def respuestaServer1(self, datos):

        print(libs)
        print("<html>")
        print("<head><title>SQL</title></head>")
        print("<body>")

        print("<table class='styled-table' border=1; style='background-color: #009879;'>")
        stCliente = datos.split("\n")
        i = 0

        self.header()
        while(i<len(stCliente)):
            cliente=stCliente[i]
            cliente = cliente.split("_")
            try:
                if (cliente[1]=="i"):
                    estado="AB"
                else:
                    estado="CE"
            except:
                pass
            self.tablas(cliente[0],estado,(cliente[2])[0:10],(cliente[2])[10:])

            i=i+1
        print("</div>")
        print("</body>")
        print("</html>")
```

```
def respuestaServer5(self, datos):
    print(libs)
    print("<html>")
    print("<head><title>SQL</title></head>")
    print("<body>")

    print("<table class='styled-table' border=1; style='background-color: #009879;'>")
    stCliente = datos.split("\n")
    i = 0

    self.header()
    while(i<len(stCliente)):
        cliente=stCliente[i]
        cliente = cliente.split("_")
        if (cliente[1]=="i"):
            estado="AB"
        else:
            estado="CE"
        VAL = ((cliente[2])[10:13])
        #print("FECHA: ",(cliente[2])[0:11])
        #print("RECIBIDO: ",F)
        if (HR!="" and F!=""):
            #print("HORA SQL: '"+str((cliente[2])[0:10])+"' <br>")
            #print("F: '"+str(F)+"' <br>")
            if((int(HR)==int(VAL)) and (str((cliente[2])[0:10]))==str(F) ):
                print("ENTRA")
                self.tablas(cliente[0],estado,(cliente[2])[0:11],(cliente[2])[10:])

        else:
            if (F!=""):
                if (str((cliente[2])[0:10])==str(F)):
                    self.tablas(cliente[0],estado,(cliente[2])[0:11],(cliente[2])[10:])
            else:
                if(int(HR)==int(VAL)):
                    self.tablas(cliente[0],estado,(cliente[2])[0:11],(cliente[2])[10:])
        i=i+1

    print("</body>")
    print("</html>")
#
```

Imgs 5.4-19 respuesta server

```
def respuestaServer2(self, datos):
    print(libs)

    print("<html>")
    print("<head><title>SQL</title></head>")
    print("<body>")


    print("<table class='styled-table' border=1; style='background-color: #009879;'>")
    stCliente = datos.split("\n")
    i = 0
    self.header()
    while(i<len(stCliente)):
        cliente2=stCliente[i]
        cliente2 = cliente2.split(" ")

        self.tablas(cliente2[0],cliente2[1],(cliente2[2]),(cliente2[3]))
        i=i+1

    print("</body>")
    print("</html>")

def respuestaServer3(self, datos):
    print(libs)
    print("Content-type: text/html\n\n")
    print("<html>")
    print("<head><title>SQL</title></head>")
    print("<body>")


    print("<table class='styled-table' border=1; style='background-color: #009879;'>")
    stCliente = datos.split("\n")
    i = 0
    self.header()
    while(i<len(stCliente)):
        cliente2=stCliente[i]
        cliente2 = cliente2.split(" ")
        try:
            if (HR!="" and F!=""):
            #print("HORA SQL: '"+str((cliente[2])[0:10])+"' <br>")
            #print("F: '"+str(F)+"' <br>")
                if((int(HR)==int(VAL)) and str((cliente2[2]))==str(F) ):
                    self.tablas(cliente2[0],cliente2[1],(cliente2[2]),(cliente2[3]))
            else:
                if (F!=""):
                    if (str((cliente2[2]))==str(F)==F):
                        self.tablas(cliente2[0],cliente2[1],(cliente2[2]),(cliente2[3]))
                else:

                    if (((cliente2[3])[0:2])==HR):
                        self.tablas(cliente2[0],cliente2[1],(cliente2[2]),(cliente2[3]))
        except:
            pass
        i=i+1

    print("</body>")
    print("</html>")
```

Imgs 5.4-20 respuesta server

```
def procesarTransaccion(self):
    conexion = mysql.connector.connect(user="root", password="root", database="iot")

    if(B == "Consultar Valvula"):

        query = "SELECT * FROM Valvula WHERE id='"+V+"'"

        statement = conexion.cursor()
        statement.execute(query)


        datos=""
        st = datos.split("_")
        tupla = statement.fetchone()
        # while tupla is not None:
        while(tupla != None):
            ide=tupla[0]
            estado=tupla[1]
            fh=tupla[2]

            datos = datos + str(ide) +"_"+ str(estado) +"_"+ str(fh) + "\n"
            # print(tupla)
            tupla = statement.fetchone()


        statement.close()
        conexion.close

        if (HR!="" or F!=""):
            self.respuestaServer5(datos)
        else:
            self.respuestaServer1(datos)
```

Img 5.4-21 libraries.php(procesar botones)

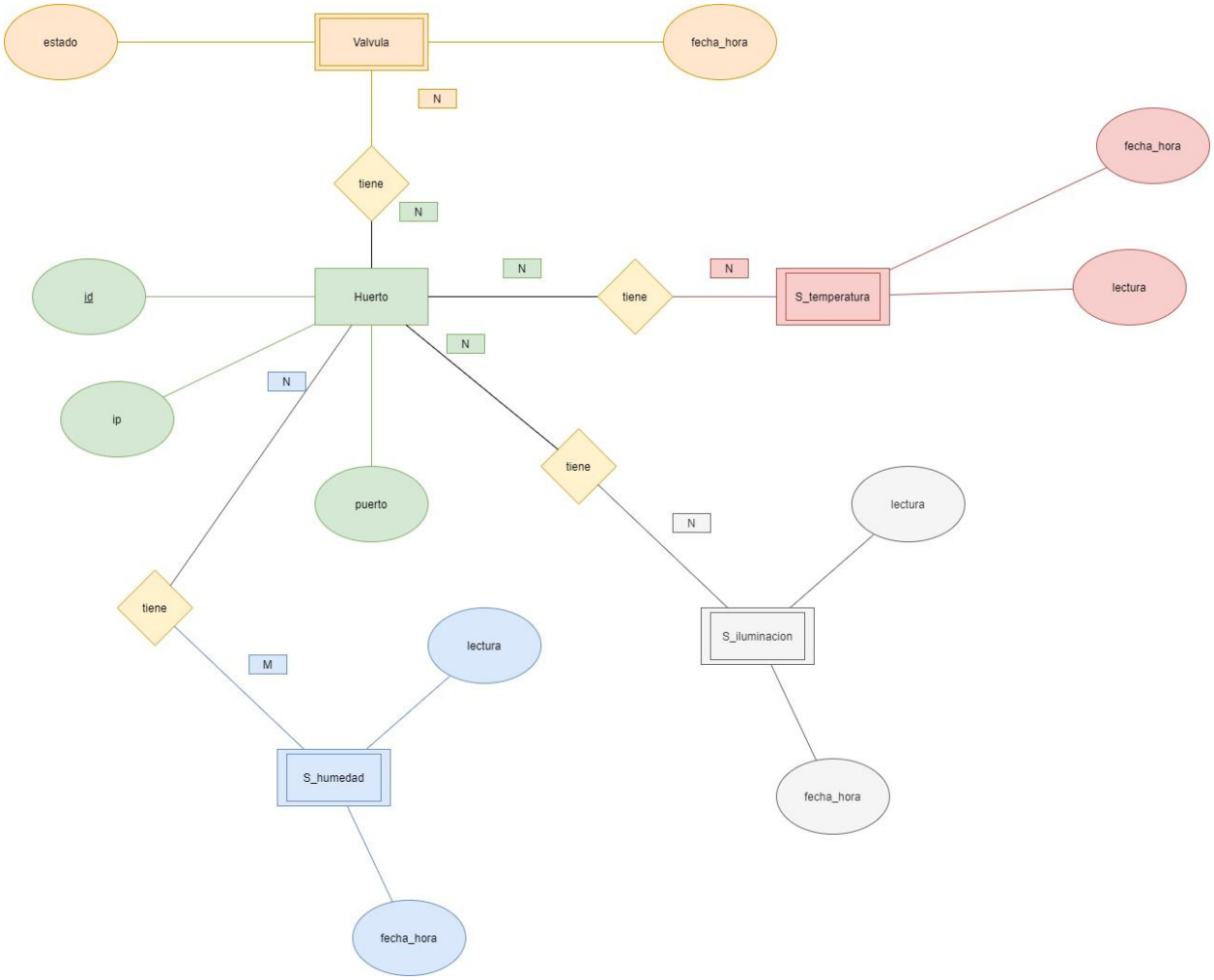
```
    elif (B == "Consultar Temperatura"):

        query = "SELECT * FROM S_temperatura WHERE id='"+V+"'"

        statement = conexion.cursor()
        statement.execute(query)


        datos=""
        st = datos.split("_")
        tupla = statement.fetchone()
        # while tupla is not None:
        while(tupla != None):

            ide=tupla[0]
            estado=tupla[1]
            fh=tupla[2]

            datos = datos + str(ide) +" "+ str(estado) +" "+ str(fh) + "\n"
            # print(tupla)
            tupla = statement.fetchone()


        statement.close()
        conexion.close
        if (HR!="" or F!=""):
            self.respuestaServer3(datos)
        else:
            self.respuestaServer2(datos)

    elif (B == "Consultar Humedad"):

        query = "SELECT * FROM Humedad WHERE id='"+V+"'"

        statement = conexion.cursor()
        statement.execute(query)

        # Procesar los datos de la tabla resultante
        datos=""
        st = datos.split("_")
        tupla = statement.fetchone()
        # while tupla is not None:
        while(tupla != None):

            ide=tupla[0]
            estado=tupla[1]
            fh=tupla[2]

            datos = datos + str(ide) +" "+ str(estado) +" "+ str(fh) + "\n"
            # print(tupla)
            tupla = statement.fetchone()


        statement.close()
        conexion.close
        if (HR!="" or F!=""):
            self.respuestaServer3(datos)
        else:
            self.respuestaServer2(datos)
```

Img 5.4-22 libraries.php(cliente controller)

```
        elif (B == "Consultar Luminosidad"):

            query = "SELECT * FROM S_iluminacion WHERE id='"+V+"'"

            statement = conexion.cursor()
            statement.execute(query)

            # Procesar los datos de la tabla resultante
            datos=""
            st = datos.split("_")
            tupla = statement.fetchone()
            # while tupla is not None:
            while(tupla != None):

                ide=tupla[0]
                estado=tupla[1]
                fh=tupla[2]

                datos = datos + str(ide) +" "+ str(estado) +" "+ str(fh) + "\n"
                # print(tupla)
                tupla = statement.fetchone()


            statement.close()
            conexion.close
            if (HR!="" or F!=""):
                self.respuestaServer3(datos)
            else:
                self.respuestaServer2(datos)


# Crear el objeto de la clase Primero
HR= None
V = sys.argv[1]
HR = sys.argv[2]
F = sys.argv[3]
B = sys.argv[4]
libs= """
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
        <link rel="icon" href="images/favicon.ico">
        <link rel="shortcut icon" href="images/favicon.ico" />
        <link rel="stylesheet" href="css/owl.carousel.css">
        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.js"></script>
        <script src="js/jquery-migrate-1.1.1.js"></script>
        <script src="js/script.js"></script>
        <script src="js/jquery.ui.totop.js"></script>
        <script src="js/superfish.js"></script>
        <script src="js/sForm.js"></script>
        <script src="js/jquery.equalheights.js"></script>

        <script src="js/jquery.easing.1.3.js"></script>
        <div class="content" >


        """

cliente = ClienteController()
cliente.procesarTransaccion()
```

Img 5.4-23 libraries.php(cliente controller)

## 5.5 Uso y mantenimiento de la base de datos




Img 5.5-1 libraries.php(Diagrama ER)

```
if(B == "Consultar Valvula"):

    query = "SELECT * FROM Valvula WHERE id='"+V+"'"

elif (B == "Consultar Humedad"):

    query = "SELECT * FROM Humedad WHERE id='"+V+"'"

elif (B == "Consultar Luminosidad"):

    query = "SELECT * FROM S_iluminacion WHERE id='"+V+"'"
```

```
def tablas(self,IDET,ESTADOT,FECHAT,HORAT):

    print("<tr> <td style='padding: 15px; color: black; '>"+IDET+"</td>"+"<td style='padding: 15px; color: black;'>"+ESTADOT+"

+ESTADOT+"</td>"+"<td style='padding: 15px; color: black;'>"+FECHAT+"</td>"+"<td style='padding: 15px; color: black;'>"+HORAT+"</td>")
```

```
elif (B == "Consultar Temperatura"):

    query = "SELECT * FROM S_temperatura WHERE id='"+V+"'"
```

Imgs 5.5-2 libraries.php(Queries)

## 5.6 Conexión de distintos dispositivos a través de protocolos especiales para IoT

Para conectar los NodeMCU a la Raspberry utilizamos el software MQTT para la comunicación entre dispositivos, esto nos permitió realizar conexiones rápidas,versátiles y sencillas a través de tópicos de comunicación.

Se creó un tópico distinto para cada valor de sensor que los nodes obtienen y se envían al broker para su manejo; posteriormente, se reciben y segmentan con el script en python para su asignación a los datos en tiempo real y a la base de datos MySQL.

# 6. Reflexión y conclusiones

Durante el diseño e implementación del huerto automatizado se presentaron varios desafíos para su correcta implementación. Entre ellos se pueden destacar problemas con la apertura de los puertos debido a que el proveedor de internet no las autorizaba. Tres de los cinco integrantes del equipo tenían servicio de internet TotalPlay, el cual no permitía dicha apertura; se decidió entonces tener dos pares de huertos, cada uno en las ubicaciones de los otros dos integrantes donde su proveedor sí permitía la apertura.

Otro de los problemas que también se presentaron fue que la presión de salida de los tanques que alimentaban a los cultivos no era suficiente para el buen funcionamiento de las electroválvulas y, como ya se mencionó en el desarrollo de este documento, se optó por conectarlas directamente a la llave de paso.

Además, se consideró crear un presupuesto holgado para la compra de materiales extra en caso de falla de fábrica o estropeo y no salir a comprar constantemente.

Hablando de las ocasiones en las que fue necesario reunirse en físico, cabe mencionar que se tomaron las medidas de higiene pertinentes (gel antibacterial, cubrebocas y sana distancia) velando por garantizar la seguridad e integridad de cada uno de los miembros del equipo.

A pesar de que este proyecto de un huerto automatizado estuvo enfocado a sólo 4 cultivos, es posible expandirlo de forma modular a escalas mayores con la ayuda de placas de circuito impreso (PCV) ya que uno de los problemas más frecuentes fue que los componentes se desoldaban constantemente por mantener una posición de mucho estrés para la soldadura, a pesar de tener soporte con thermofit y cinta de aislar para mantenerlos en la posición que les correspondía.

La importancia de escalar este proyecto recae en que no sólo se quede como una práctica de integración de Internet de las Cosas, si no que pueda ser usado como una alternativa sustentable y competitiva para la agricultura dentro de las ciudades y contribuir a alcanzar el Objetivo 11 de Desarrollo Sostenible de la ONU sobre Ciudades y Comunidades Sostenibles. Esto porque promueve la reducción de los desechos con la posibilidad de usarlos como macetas; mejora la biodiversidad y paisaje urbano; y, en general, reduce la huella ecológica ya que no se usarían tantos recursos para importar perecederos.

# 7. Referencias

Orsini, F., Kahane, R., Nono-Womdim, R., & Gianquinto, G. (2013). *Urban agriculture in the developing world: a review*. Agronomy for Sustainable Development. 33, 695–720. Recuperado en diciembre 02, 2020 de https://doi.org/10.1007/s13593-013-0143-z

United Nations Statistics Division Development Data and Outreach Branch. (2020). *11 Sustainable cities and communities.* Department of Economic and Social Affairs: Statistics Division. Recuperado en diciembre 02, 2020 de https://unstats.un.org/sdgs/report/2019/goal-11/