



Tecnológico de Monterrey

**TC1004B – IMPLEMENTACIÓN DE INTERNET DE LAS COSAS
GRUPO 6**

Profesores: Ing. Rafael Emilio Dávalos Villarreal
& Ing. Sergio Ramírez San Vicente

Domótica

Equipo #5

Alejandro Treviño García - A00825543

Eduardo Andrés Valerín Vijil - A00830774

Karen Lizette Rodríguez Hernández - A01197734

Brenda Elena Saucedo González - A00829855

Índice

1. Antecedentes de IoT.....	3
2. Introducción de la propuesta del reto.....	3
3. Descripción de cada elemento del proyecto.....	4
3.1. Microcontrolador nodeMCU.....	4
3.2. Sensores y su funcionamiento.....	4
3.2.1. DHT11.....	4
3.2.2. Sensor ultrasónico.....	4
3.2.3. Fotorresistor.....	4
3.2.4. Pushbutton.....	4
3.3. Actuadores y su funcionamiento.....	5
3.3.1. LEDs.....	5
3.3.2. Servomotor.....	5
3.4. Broker MQTT.....	5
3.5. Node-RED.....	5
3.6. Base de Datos.....	5
4. Diseño de los elementos interconectados.....	6
5. Desarrollo del proyecto.....	7
5.1. Imágenes de los circuitos.....	7
5.2. Programas desarrollados.....	10
5.2.1. Programa para configurar el nodeMCU.....	10
5.2.2. Base de datos.....	18
5.2.3. Node-RED.....	19
5.3. Procesos o pruebas que no funcionaron.....	20
5.4. Resultados.....	22
5.4.1. IDE Arduino.....	22
5.4.2. Base de datos.....	23
6. Conclusión.....	24
7. Reflexión Individual.....	25
8. Referencias bibliográficas.....	25

1. Antecedentes de IoT

El IoT ha evolucionado a partir de la convergencia de distintas tecnologías, pasando de la comunicación máquina a máquina (M2M), máquinas conectadas entre sí a través de una red sin interacción humana, a convertirse en una red de sensores de miles de millones de dispositivos inteligentes que conectan personas, sistemas y otras aplicaciones para recopilar y compartir datos.

En 1990, Jhon Romkey y Simon Hacket desarrollaron el primer objeto con conexión a Internet, el cual fue su tostadora inteligente. En 1999, el ingeniero Bill Joy, fue quien profundizó el concepto de Internet de las cosas como "comunicaciones entre dispositivos". Para 2009, el británico Kevin Ashton acuñó por primera vez el nombre de internet de las cosas para explicar una nueva idea relacionada con la conexión de la cadena de suministro de la empresa (Procter & Gamble) a Internet. Durante este período de tiempo, la cantidad de dispositivos conectados a Internet eclipsó a la población mundial. Con más "cosas" conectadas a Internet que personas en el mundo, se avecinaba una nueva era y nació el Internet de las cosas.

En sí, Internet of Things (IoT) describe la red de objetos físicos (cosas) que incorporan sensores, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas.

2. Introducción de la propuesta del reto

Un sistema domótico es capaz de recoger información proveniente de unos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información.

Con esto, la domótica permite dar respuesta a los requerimientos que plantean los cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles.

El objetivo del proyecto consiste principalmente en desarrollar una propuesta a favor de la automatización de viviendas inteligentes haciendo uso del IoT, en donde nosotros haremos uso de un microcontrolador al que irán conectados algunos sensores: sensor de temperatura, ultrasónico, fotorresistor y un pushbutton, para medir diferentes cantidades físicas, generar datos y generar respuestas por medio de actuadores, como lo son los servomotores y los focos LEDs, que nos indicarían el encendido/apagado de algún aparato doméstico u otro tipo de acciones para la automatización de viviendas inteligentes, que al mismo tiempo estarían controladas por ciertos parámetros especificados. Estos datos generados también se almacenarán en una base de datos para poder hacer posible su análisis en etapas posteriores. Por otro lado, por medio de algunas conexiones y transmisión de mensajes, enviaremos datos por HiveMQ para que el microcontrolador los reciba y con ellos también pueda generar respuestas.

3. Descripción de cada elemento del proyecto

3.1. Microcontrolador nodeMCU

Placa de desarrollo que nos permite manejar dispositivos interconectados a través de una red de internet. Sirve para desarrollar proyectos en torno a objetos inalámbricos, donde la transferencia de datos juega un papel importante.

3.2. Sensores y su funcionamiento

Dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas.

3.2.1. DHT11

Es un sensor digital de temperatura y humedad relativa de fácil uso. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos, pero no cuenta con una salida analógica.

3.2.2. Sensor ultrasónico

Es un sensor de distancia que utiliza ultrasonido para determinar la distancia de un objeto. por medio de sus dos transductores: un micrófono y un altavoz. Genera pulsos de alta frecuencia (no perceptible por el ser humano) que rebota en los objetos cercanos y es reflejado hacia el sensor, que es captado por un micrófono. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio.

3.2.3. Fotorresistor

El fotorresistor es un componente electrónico cuya resistencia se modifica con el aumento de intensidad de luz incidente. Su funcionamiento se basa en el efecto fotoeléctrico, este efecto nos dice que los fotones que componen la luz chocan contra los electrones de un metal, de esta forma arrancan o desplazan sus átomos. En este proceso los electrones del metal están en constante movimiento y esto da paso a una corriente eléctrica.

3.2.4. Pushbutton

El botón por lo general actúa como un interruptor eléctrico, que en su interior contiene dos contactos, en el cual podemos accionar uno de ellos pulsando sobre él, en donde mientras lo mantengamos pulsado, el interruptor estará cerrado y generando continuidad, de tal manera que se le pueda atribuir una función, la cual normalmente será inversa a la que se estaba realizando antes de pulsar el botón. Por otro lado, cuando lo soltemos, el interruptor se abrirá y continuará realizando la función que se le había especificado como determinado.

3.3. Actuadores y su funcionamiento

Dispositivo que puede transformar energía eléctrica, hidráulica o neumática cuando se activa un proceso con el objetivo de crear un efecto sobre un determinado proceso automatizado. Dicho de otro modo, funciona proporcionando fuerza para que otro dispositivo actúe.

3.3.1. LEDs

Los LEDs son componentes electrónicos (diodos semiconductores) con una fuente de luz constituida por un material semiconductor dotado de dos terminales. Estos son capaces de emitir luz al ser atravesados por una corriente eléctrica.

3.3.2. Servomotor

Un servomotor es un motor eléctrico al que podemos controlar tanto la velocidad, como la posición del eje que gira. Los servomotores funcionan por medio de modulación de ancho de pulso (PWM). Todos los servos disponen de tres cables, dos para alimentación Vcc y Gnd, y un tercero para aplicar el tren de pulsos de control, que hace que el circuito de control diferencial interno ponga el servo en la posición indicada.

3.4. Broker MQTT

Es un protocolo de comunicación M2M (machine-to-machine) de tipo message queue que está basado en la pila TCP/IP como base para la comunicación. Cuenta con un servicio de mensajería push con patrón publicador/suscriptor (pub-sub). En el caso de MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación. Los clientes inician una conexión TCP/IP con el broker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza.

3.5. Node-RED

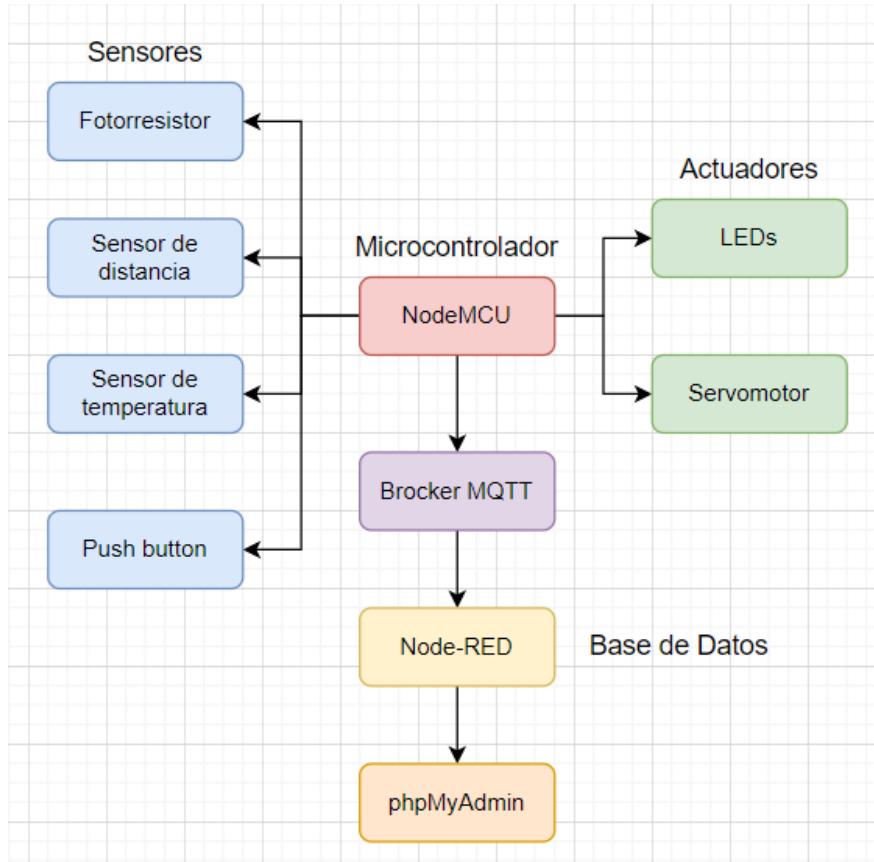
Node-RED es una herramienta de programación visual. Muestra visualmente las relaciones y funciones, y permite al usuario programar sin tener que escribir una línea. Funciona como un editor de flujo basado en el navegador donde se puede añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos.

3.6. Base de Datos

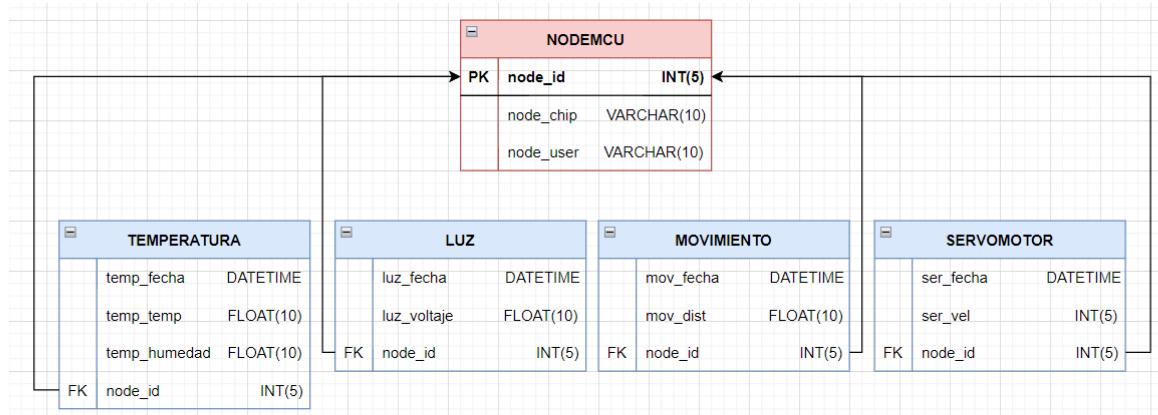
Colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.

4. Diseño de los elementos interconectados

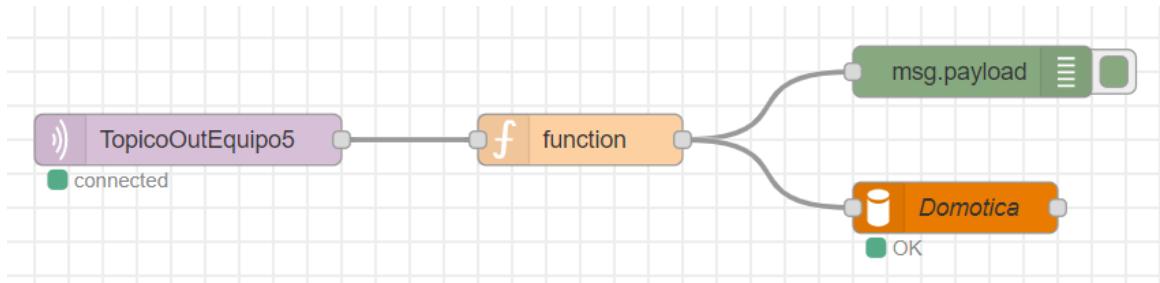
El diseño que seguimos con todos los componentes (sensores, actuadores, microcontrolador y la base de datos) se visualiza en el siguiente esquema.



Para el diseño de la Base de Datos, generamos las posibles tablas a utilizar, en donde en cada una de ellas, se estará almacenando su respectiva información generada por dicho sensor.



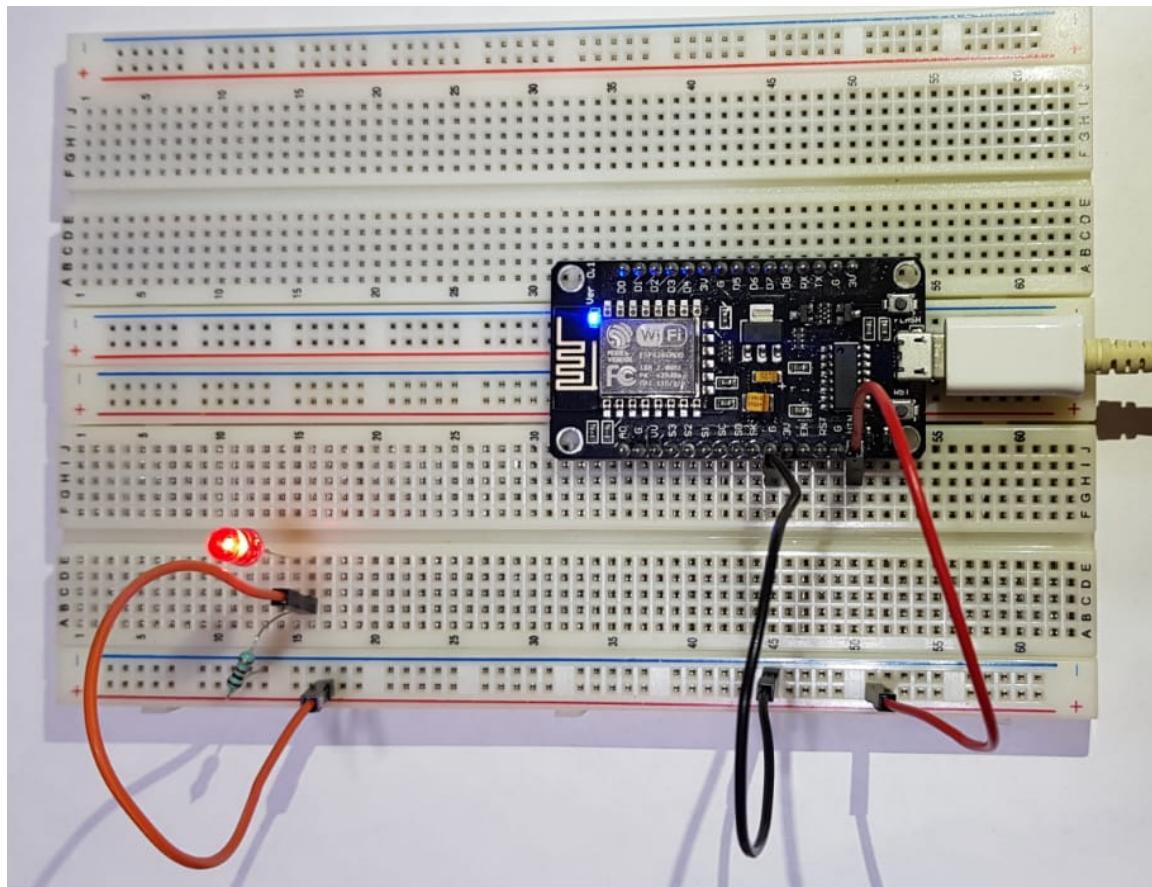
Para el Node-RED, utilizamos los nodos de mqtt, function, mysql y debugger para poder establecer la conexión entre el servidor HiveMQ y la base de datos en phpMyAdmin, para poder enviar los mensajes desde el origen del microcontrolador hasta la base de datos.



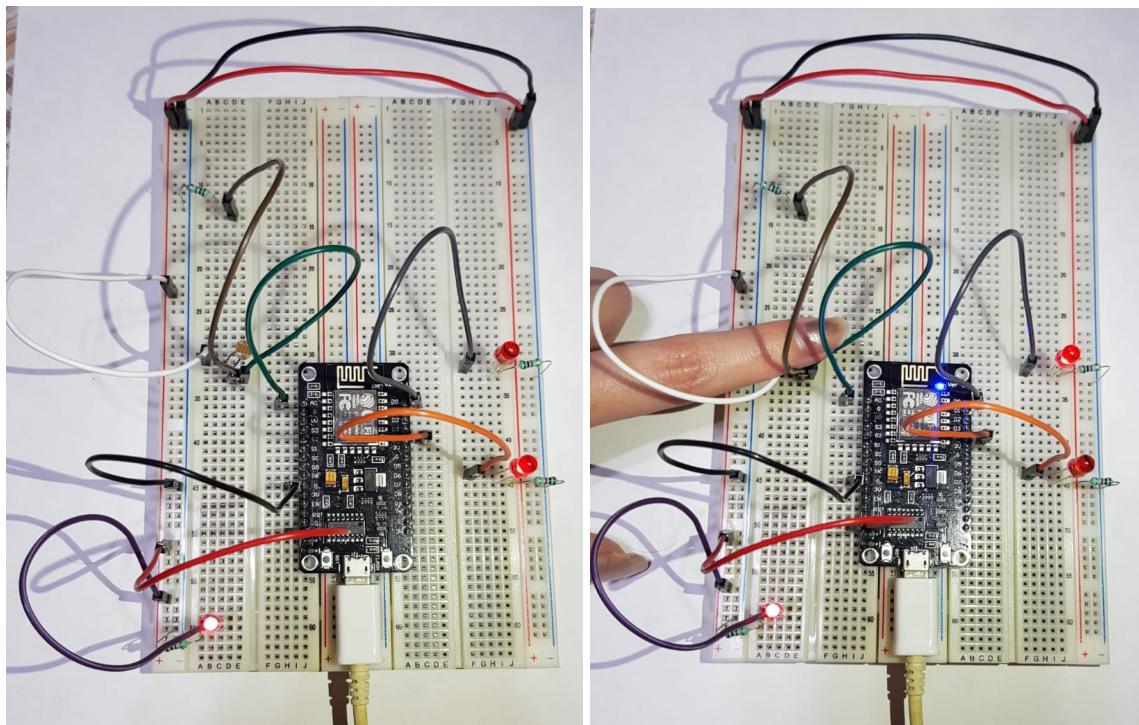
5. Desarrollo del proyecto

5.1. Imágenes de los circuitos

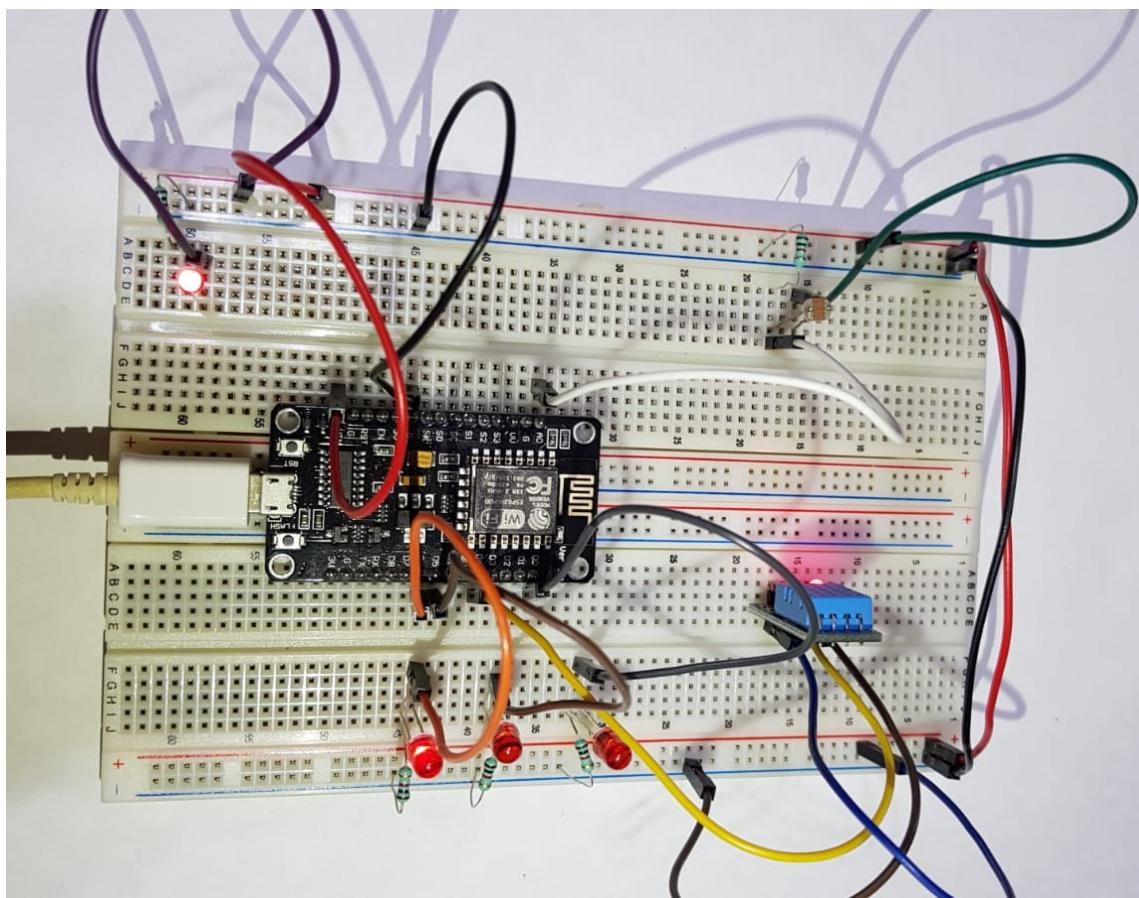
Fuimos desarrollando poco a poco el proyecto, por lo que en la primera práctica conectamos el NodeMCU a un LED parpadeante, para aprender los conceptos básicos necesarios para realizar el reto.



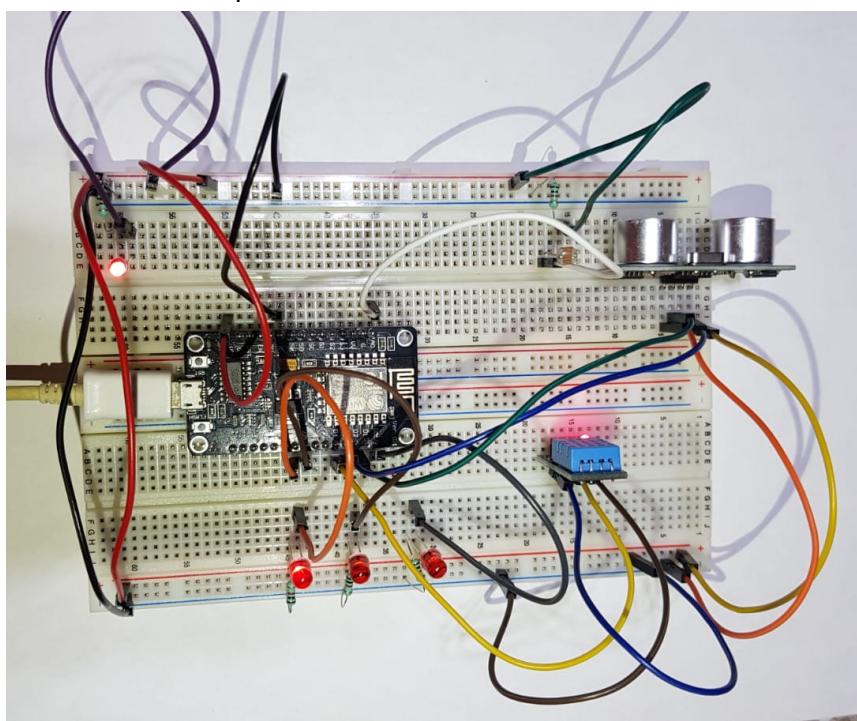
Por consiguiente, agregamos un fotorresistor junto con un LED, de modo que éste se encendiera si el fotorresistor bajaba de cierto valor.



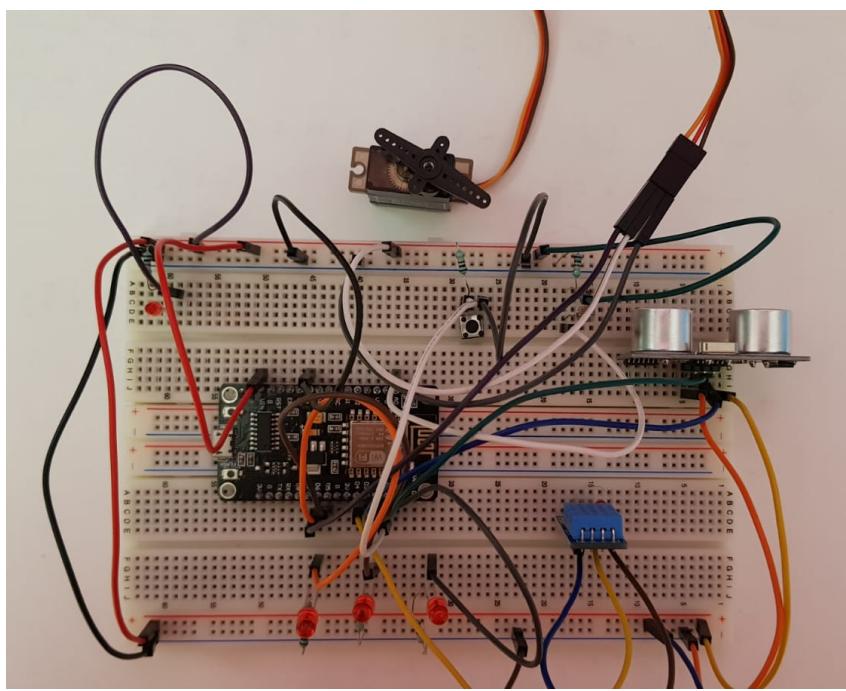
Después, agregamos un sensor de temperatura y humedad junto con un LED, para que éste se encendiera cuando la temperatura alcanzará cierto valor.



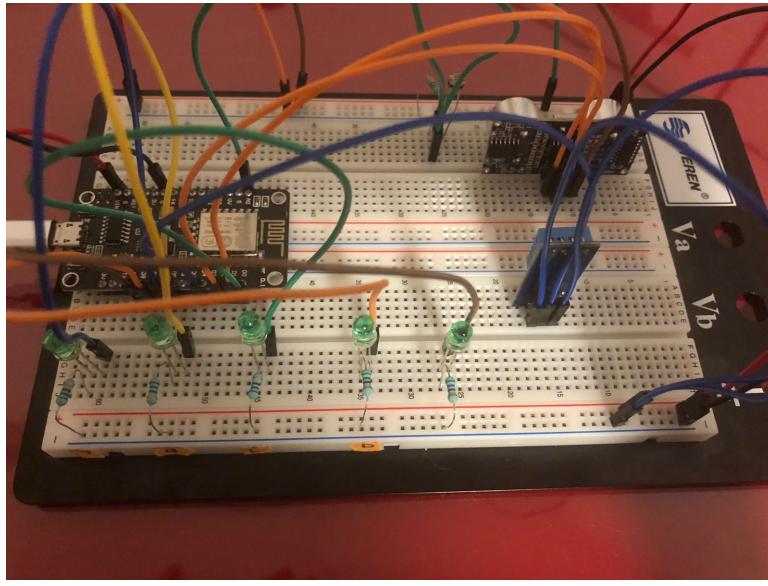
Después agregamos un sensor de distancia, que imprime el valor de la distancia en centímetros en el puerto serial.



Por último, conectamos un sensor de tacto (push button) con el fin de controlar el servomotor



En la otra versión del circuito, se removió el switch y el servomotor, y se agregaron focos LED para cada sensor, inclusive se agregó uno en especial para que éste se encendiera dependiendo del mensaje recibido por HiveMQ.



5.2. Programas desarrollados

5.2.1. Programa para configurar el nodeMCU

Al finalizar el prototipo, pudimos unir todos los componentes junto con sus programas en uno solo, de manera siguiente:

Primero se definen todos los puertos del NodeMCU junto con el identificador de cada placa (es el que diferencia cada NodeMCU).

```
// ----- Definiciones de pins de NodeMCU (se cuenta de arriba hacia abajo)
#define D0 16 // Arriba derecha 1
#define D1 5 // Arriba derecha 2
#define D2 4 // Arriba derecha 3
#define D3 0 // Arriba derecha 4
#define D4 2 // Arriba derecha 5
// Posiciones 6 3V3, 7 GND
#define D5 14 // Arriba derecha 8
#define D6 12 // Arriba derecha 9
#define D7 13 // Arriba derecha 10
#define D8 15 // Arriba derecha 11
#define SD3 10 // Arriba izquierda 4
#define SD2 9 // Arriba izquierda 5
// el RX = GPIO3 y TX = GPIO1
// LED_BUILTIN = 16 //GPIO16

int i = 23; // Identificador (ID)
```

Por consiguiente, definimos las variables para el sensor de temperatura.

```
// ----- Definiciones del sensor de temperatura DHT11
#include "DHT.h"
#define DHTPIN 2 // Posicion equivalente a D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
float t, h;
```

Definimos las variables para el uso del fotorresistor.

```
// ----- Definiciones de fotorresistor
int photoPin = A0;
int ledphotoResistor = 16; // Equivale a D0
int x = -1; //0..1023
```

Definimos las variables para el sensor de distancia.

```
// ----- Definiciones de Sensor distancia
int trigPin = D3; //Conectado a D2,    int trigPin =
int echoPin = D2; //Conectado a D1,    int echoPin =
float v = 331.5+0.6*20; // m/s
float dCm;
```

Definimos las variables para el uso del servomotor con un sensor.

```
// ----- Definiciones del servomotor
#include <Servo.h>
Servo myservo;
int botonPin = D1;
int velocidad = 0;
int pulsacion = 0;
int estado_boton;
int estado_boton_anterior;

int pos = 0; // variable to store the servo position
```

Incluimos las librerías para el uso del WiFi y definimos las variables para que el microcontrolador ESP8266 se conecte a Internet siguiendo el protocolo MQTT.

```
// ----- Definiciones de ESP8266MQTT
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
const char* ssid = "████████";
const char* password = "████████";
const char* mqtt_server = "broker.mqtt-dashboard.com";
const char* topico_salida = "TopicoOutEquipo5";
const char* topico_entrada = "TopicoInEquipo5"; //Clima o aire acondicionado
char sTopicoOutEquipo5[50];

WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (60)
char msg[MSG_BUFFER_SIZE];
int value = 0;
```

Por consiguiente, implementamos las funciones que vamos a utilizar más adelante en el código, la primera es para conectar el microcontrolador al Internet de la casa.

```
// ----- Funcion que Conecta a Wifi del Router
// We start by connecting to a WiFi network
void setup_wifi() {
    delay(10);
    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(".");
    }
    randomSeed(micros());
    Serial.println(""); Serial.println("WiFi connected"); Serial.print("IP address: "); Serial.println(WiFi.localIP());
}
```

Después implementamos funciones para conectar la placa al broker de MQTT.

```
// ----- Funcion que Conecta a Broker MQTT
void setup_mqtt() {
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    if (!client.connected()) {
        reconnect();
    }
}

// ----- Funcion que Conecta a Broker MQTT
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();
    digitalWrite(D5, HIGH); delay(100); digitalWrite(D5, LOW); // Aviso Led azul en D5 Pin 8
    if ((char)payload[0] == '1') {
        digitalWrite(D6, HIGH);
    } else {
        digitalWrite(D6, LOW);
    }
}
```

```

// ----- Funcion que Conecta o reconecta a Broker MQTT
void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        // Attempt to connect
        if (client.connect(clientId.c_str())) {
            Serial.println("connected");
            // Once connected, publish an announcement...
            //client.publish("outTopic", "hello world");
            client.subscribe(topico_entrada);
            client.subscribe(topico_salida); //OJO Quitar el comentario para hacerlo suscribe
            client.publish(topico_salida, sTopicoOutEquipo5);

            // ... and resubscribe
            //client.subscribe("inTopic");

        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

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

    unsigned long now = millis();
    if (now - lastMsg > 5000) {
        lastMsg = now;
        ++value;

        Serial.print(now); Serial.print("Publish message: ");
        Serial.println(sTopicoOutEquipo5);
        client.publish(topico_salida, sTopicoOutEquipo5);
    }
}

```

Por consiguiente, implementamos la función para medir la temperatura y humedad.

```

// ----- Funcion que lee temperatura y humedad con el sensor DHT11
void medirTemperatura() {
    delay(2000);
    h = dht.readHumidity();
    t = dht.readTemperature();
    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.print("Falló al leer el sensor DHT\n");
        return;
    }

    float hif = dht.computeHeatIndex(f, h);
    float hic = dht.computeHeatIndex(t, h, false);

    if (t > 26)
        digitalWrite(D6,HIGH);
    else
        digitalWrite(D6,LOW);
}

```

Implementamos la función para medir la luz del fotorresistor.

```
// ----- Funcion detector de iluminacion mediante un fotorresistor
void medirLuzledPhotoResistor() {
    x = analogRead(photoPin);
    if (x < 100)
        digitalWrite(ledphotoResistor, HIGH);
    else
        digitalWrite(ledphotoResistor, LOW);
}
```

Implementamos la función para medir la distancia.

```
// ----- Funcion para medir la distancia
void medirDistanciaCm() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(3);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(trigPin, LOW);

    //listen for echo
    float tUs = pulseIn(echoPin, HIGH); //microseconds
    float t = tUs / 1000.0 / 1000.0 / 2.0; //s
    float d = t*v; //m
    dCm = d*100; // cm
    delay(200); //ms
}
```

Implementamos la función para controlar la velocidad del servomotor a través de un botón.

```
// ----- Funcion para controlar el servomotor
void detecta_pulsacion() {
    // leo estado boton
    estado_boton = digitalRead(botonPin);

    //Detección pulsación boton
    if (estado_boton != estado_boton_anterior) {
        if (estado_boton == HIGH) {
            //Serial.println("Nueva pulsacion");
            pulsacion++;
            switch (pulsacion) {
                case 1:
                    velocidad = 30;
                    break;
                case 2:
                    velocidad = 20;
                    break;
                case 3:
                    velocidad = 10;
                    break;
                case 4:
                    pulsacion = 0;
                    velocidad = 0;
                    break;
            }
        }
    }
    estado_boton_anterior = estado_boton;
}
```

Inicializamos todos los puertos utilizados para el prototipo.

```
void setup() {  
    pinMode(D0, OUTPUT); // Led del fotoresistor  
    pinMode(D5, OUTPUT); // Led blinking  
    pinMode(D6, OUTPUT); // Led del sensor de temperatura  
  
    // ----- Iniciar consola: Herramientas, Monitor serie  
    Serial.begin(9600); // Iniciar consola  
  
    // ----- Abrir Wifi y MQTT  
    setup_wifi();  
    setup_mqtt();  
  
    // ----- Inicializacion sensor temperatura - humedad  
    Serial.println(F("DHTxx test!"));  
    dht.begin();  
  
    // ----- Inicializacion fotoresistor  
    pinMode(ledphotoResistor, OUTPUT);  
    pinMode(photoPin, INPUT);  
    Serial.println("Iniciando lectura fotorresistor");  
  
    // ----- Inicializacion sensor de distancia  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
    Serial.println("Iniciando Sensor Distancia");  
  
    //----- Inicializacion servomotor  
    myservo.attach(D7); // attaches the servo on pin D7 to the servo object  
    myservo.write(20);  
    pinMode(botonPin, INPUT_PULLUP);  
    estado_boton_anterior = digitalRead(botonPin);  
    Serial.println("Iniciando Servomotor");  
}  
}
```

Por último, realizamos todas las mediciones y publicamos los datos obtenidos dentro de un loop.

```
void loop() {  
    // blinking  
    digitalWrite(D5,HIGH);  
    delay(500);  
    digitalWrite(D5,LOW);  
    delay(500);  
  
    medirTemperatura();  
    conectarMQTT();  
    medirLuzledPhotoResistor();  
    medirDistanciaCm();  
  
    // ----- Servo con botón -----  
    detecta_pulsacion();  
  
    if (pulsacion > 0) {  
        for (pos = 20; pos <= 160; pos += 5) {  
            myservo.write(pos);  
            delay(velocidad);  
            detecta_pulsacion();  
        }  
        for (pos = 160; pos >= 20; pos -= 5) {  
            myservo.write(pos);  
            delay(velocidad);  
            detecta_pulsacion();  
        }  
    }  
    sprintf (sTopicoOutEquipo5, MSG_BUFFER_SIZE, "{\"t\":%f,\"h\":%f,\"v\":%d,\"d\":%f,\"vel\":%d,\"i\":%d}", t,h,x,dCm,velocidad,i);  
}
```

La otra versión del software para el nodeMCU que no tiene el servomotor y el pushbutton, quedó con pocas diferencias, como se muestra a continuación.

Se definió el identificador para esta nueva versión de prototipo, para poder identificarlo de la versión anterior.

```
// Identificador de este nodeMCU  
int i = 12;
```

Se definió una variable “vel” (velocidad) establecida con 0, ya que no se cuenta con el servomotor ni pushbutton, aunque también se hace para evitar errores de inconsistencia de datos al momento de enviar el mensaje desde Node-RED a la base de datos.

```
// Velocidad  
int vel = 0;
```

A continuación, se presenta un LED en específico (D8) que se va a encender cuando el primer carácter del mensaje recibido de TopicInEquipo5 sea 1. Si este carácter es diferente de 1, el LED se apaga.

```
// ----- Funcion que Conecta a Broker MQTT  
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();  
    digitalWrite(D5, HIGH); delay(100); digitalWrite(D5, LOW); // Aviso Led azul en D5 Pin 8  
  
    // Switch on the LED if an 1 was received as first character  
    if ((char)payload[0] == '1') {  
        digitalWrite(D8, HIGH); // 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). No es cierto en D6.  
    } else {  
        digitalWrite(D8, LOW); // Turn the LED off by making the voltage HIGH  
    }  
}
```

Además, la función en donde se hace la reconexión, sólo nos suscribimos en el tópico de entrada, para de esta manera evitar recibir y desplegar los mensajes del tópico al que estamos enviando los datos.

```
// ----- Funcion que Conecta o reconecta a Broker MQTT  
void reconnect() {  
    // Loop until we're reconnected  
    while (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
        // Create a random client ID  
        String clientId = "ESP8266Client-";  
        clientId += String(random(0xffff), HEX);  
        // Attempt to connect  
        if (client.connect(clientId.c_str())) {  
            Serial.println("connected");  
            // Once connected, publish an announcement...  
            // ... and resubscribe  
            client.subscribe(topico_entrada);  
            client.publish(topico_salida, sTopicoOutEquipo5);  
  
            // ... and resubscribe  
            //client.subscribe("inTopic");  
  
        } else {  
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}
```

Aquí se muestra el LED que se usó para el sensor ultrasónico, el cual se enciende dependiendo de las condiciones dadas.

```
// -----
// ----- Funcion que dispara un rayo, lo apaga y espera el echo
// para medir distancia
void medirDistanciaCm() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(3);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(trigPin, LOW);

    //listen for echo
    float tUs = pulseIn(echoPin, HIGH); //microseconds
    float t = tUs / 1000.0 / 1000.0 / 2.0; //s
    float dM = t*v; //m
    d = dM*100; // cm
    Serial.print("Distancia(cm): ");
    Serial.println(d);

    if (d < 100)
        digitalWrite(D7,HIGH);
    else
        digitalWrite(D7,LOW);

    delay(200); //ms
}
```

Se definieron los focos LED extras, en donde uno es para el sensor ultrasónico y el otro es para lo que se reciba del TopicInEquipo5.

```
void setup() {
    pinMode(D0,OUTPUT); // Led del fotoresistor
    pinMode(D5,OUTPUT); // Led blinking
    pinMode(D6,OUTPUT); // Led del sensor de temperatura
    pinMode(D7,OUTPUT); // Led del sensor de distancia
    pinMode(D8,OUTPUT); // Led del Message arrived

    // ----- Iniciar consola: Herramientas, Monitor serie
    Serial.begin(9600);

    // ----- Abrir Wifi y MQTT
    setup_wifi();
    setup_mqtt();

    // ----- Inicializacion sensor temperatura - humedad
    Serial.println(F("DHTxx test!"));
    Serial.println("DHTxx test!");
    dht.begin();

    // ----- Inicializacion fotoresistor
    pinMode(ledphotoResistor, OUTPUT);
    pinMode(photoPin, INPUT);
    Serial.println("Iniciando lectura fotoresistor");

    // ----- Inicializacion sensor de distancia
    pinMode(trigPin,OUTPUT);
    pinMode(echoPin,INPUT);
    Serial.println("Iniciando Sensor Distancia");
}
```

5.2.2. Base de Datos

Primeramente, usamos phpMyAdmin para crear una base de datos. En caso de que exista una base de datos que se llame de manera similar, esta se elimina.

```
1 DROP DATABASE IF EXISTS domotica_eq5;
2 CREATE DATABASE domotica_eq5;
3 USE domotica_eq5;
4
```

Después, se crearon las demás tablas que forman parte de la base de datos, cada una con sus respectivos campos.

```
5 CREATE TABLE NODEMCU(
6     node_id INT(5) NOT NULL,
7     node_chip VARCHAR(10) NOT NULL,
8     node_user CHAR(20) NOT NULL,
9     PRIMARY KEY(node_id)
10 );
11
12 CREATE TABLE TEMPERATURA(
13     node_id INT(5) NOT NULL,
14     temp_fecha DATETIME NOT NULL,
15     temp_temp FLOAT(10) NOT NULL,
16     temp_humedad FLOAT(10) NOT NULL,
17     FOREIGN KEY(node_id) REFERENCES NodeMCU(node_id)
18 );
19
20 CREATE TABLE LUZ(
21     node_id INT(5) NOT NULL,
22     luz_fecha DATETIME NOT NULL,
23     luz_voltaje FLOAT(10) NOT NULL,
24     FOREIGN KEY(node_id) REFERENCES NodeMCU(node_id)
25 );
26
27 CREATE TABLE MOVIMIENTO (
28     node_id INT(5) NOT NULL,
29     mov_fecha DATETIME NOT NULL,
30     mov_dist FLOAT(10) NOT NULL,
31     FOREIGN KEY(node_id) REFERENCES NodeMCU(node_id)
32 );
33
34 CREATE TABLE SERVOMOTOR (
35     node_id INT(5) NOT NULL,
36     ser_fecha DATETIME NOT NULL,
37     ser_vel INT(5) NOT NULL,
38     FOREIGN KEY(node_id) REFERENCES NodeMCU(node_id)
39 );
```

Finalmente, en la tabla de NODEMCU, insertamos los valores correspondientes de las personas del equipo que poseen un NodeMCU.

```
40
41 Insert INTO NODEMCU VALUES(12,'ESP8266','Brenda');
42 Insert INTO NODEMCU VALUES(23,'ESP8266','Karen');
```

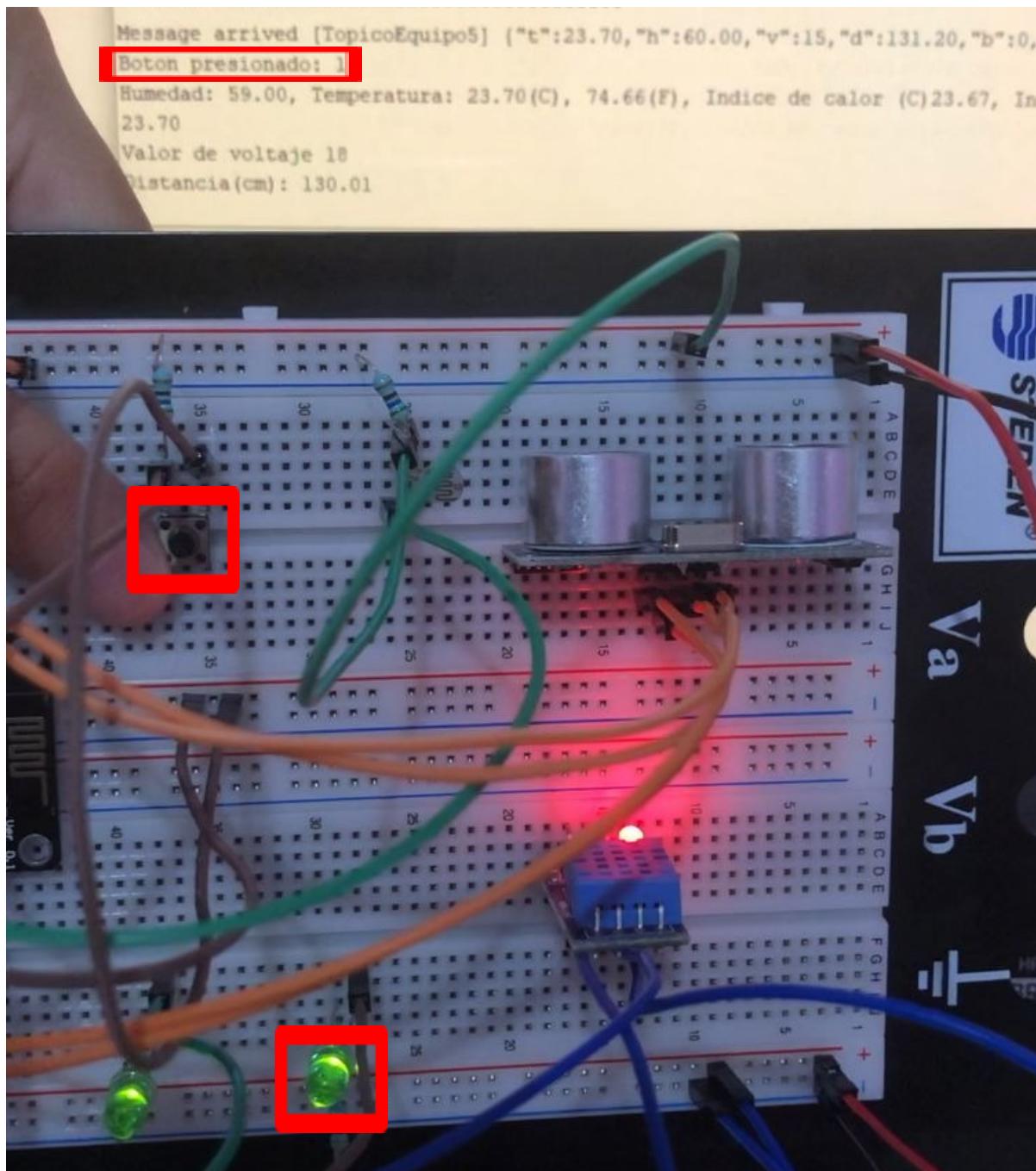
5.2.3. Node-RED

Para establecer la conexión entre el servidor HiveMQ y la base de datos, en node-RED insertamos un nodo de MQTT IN, en el cual nos suscribimos al servidor y al tópico del que vamos a estar recibiendo información. Después insertamos un nodo de function, en el cual hicimos uso de algunas funciones y variables con ayuda del formato JSON para facilitarnos la tarea de dividir y seccionar el mensaje enviado desde el servidor HiveMQ, y poderlo adaptar al formato de phpMyAdmin.

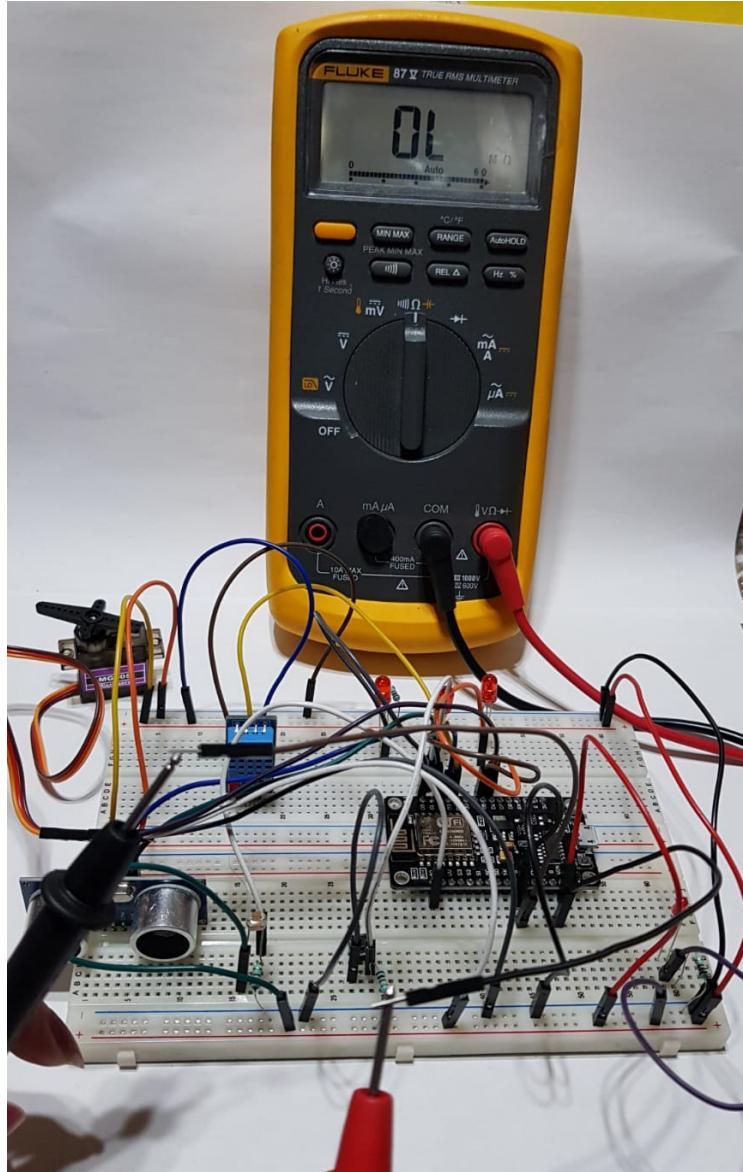
```
1 // Variables
2 var dom = JSON.parse(msg.payload);
3 var temp = dom.t;
4 var hum = dom.h;
5 var vol = dom.v;
6 var dist = dom.d;
7 var vel = dom.vel;
8 var id = dom.i;
9
10
11 // Formato DATETIME
12 var dateNow = new Date();
13 var timeAsString = dateNow.toLocaleTimeString();
14 var tzoffset = dateNow.getTimezoneOffset() * 60000;
15 var fecha = (new Date(dateNow - tzoffset)).toISOString().slice(0, -14);
16 var fechaHora = fecha + " " + timeAsString;
17
18
19 // Almacenar la información en la BD
20 msg.topic = "INSERT INTO temperatura VALUES(" + id + ", '" +
21   + fechaHora + "', " + temp + ", " + hum + ");" +
22
23 "INSERT INTO luz VALUES(" + id + ", '" +
24   + fechaHora + "', " + vol + ");" +
25
26 "INSERT INTO movimiento VALUES(" + id + ", '" +
27   + fechaHora + "', " + dist + ");" +
28
29 "INSERT INTO servomotor VALUES(" + id + ", '" +
30   + fechaHora + "', " + vel + ");"
31
32 return msg;
```

También hicimos uso del nodo de debug para desplegar el contenido de cada mensaje recibido en el node-RED. Finalmente, también agregamos el nodo mysql para enviar el mensaje a la base de datos y esta estuviera almacenando la información que iba recibiendo.

5.3. Procesos o pruebas que no funcionaron



Uno de los procesos que no pudimos eficientizar, fue cuando añadimos un pushbutton al prototipo y al programa, ya que supusimos que debido a las demás funciones, en especial la que hace la conexión con el MQTT, ralentizan el correcto funcionamiento del botón, ya que éste espera a que se ejecuten las demás instrucciones para poder habilitar el siguiente cambio, es decir, al dejar presionado el botón, este se tarda en responder para encender uno de los LED o servomotores que está conectado con este mismo, y lo mismo sucede cuando se quiere apagar o detener.



Otro reto que nos enfrentamos al hacer este prototipo, es que al principio, cuando estábamos haciendo las primeras prácticas, nos dimos cuenta con la ayuda de un multímetro que una tierra (G) del NodeMCU no estaba puenteada, es decir, no funcionaba. Por lo que tuvimos que utilizar otro puerto del NodeMCU para la tierra.

Otra de las cosas, fue que en otra de las placas encontramos que una patita del NodeMCU tampoco estaba puenteada, razón por la cual no funcionaban algunas lecturas de un sensor. Por este tipo de cuestiones, realizamos dos versiones de prototipos con la misma funcionalidad, con la única diferencia de los componentes incluidos en cada placa. En una versión se incluye el servomotor con el switch, y en la otra, como no se cuenta con un servomotor y el switch, se agregó un LED por cada sensor, y también un LED que se activa dependiendo de los datos que reciba de HiveMQ. Para resolver el problema del dato que se envía del servomotor, en esta última versión se declaró la variable con 0.

```


    Connecting to IZZI-C4E3
    .....
    WiFi connected
    IP address: 192.168.1.8
    Attempting MQTT connection...connected
    DHTxx test!
    Iniciando lectura fotorresistor
    Iniciando Sensor Distancia
    Humedad: 58.00, Temperatura: 25.70(C), 78.26(F), Indice de calor (C)25.84, Indice de calor (F)78.51
    25.70
    9631Publish message: {"t":25.70,"h":58.00}
    Valor de voltaje 206
    Distancia(cm): 58.48

    ----- CUT HERE FOR EXCEPTION DECODER -----

Exception (0):
epc1=0x4020164d epc2=0x00000000 epc3=0x00000000 excvaddr=0x00000000 depc=0x00000000

>>>stack>>>

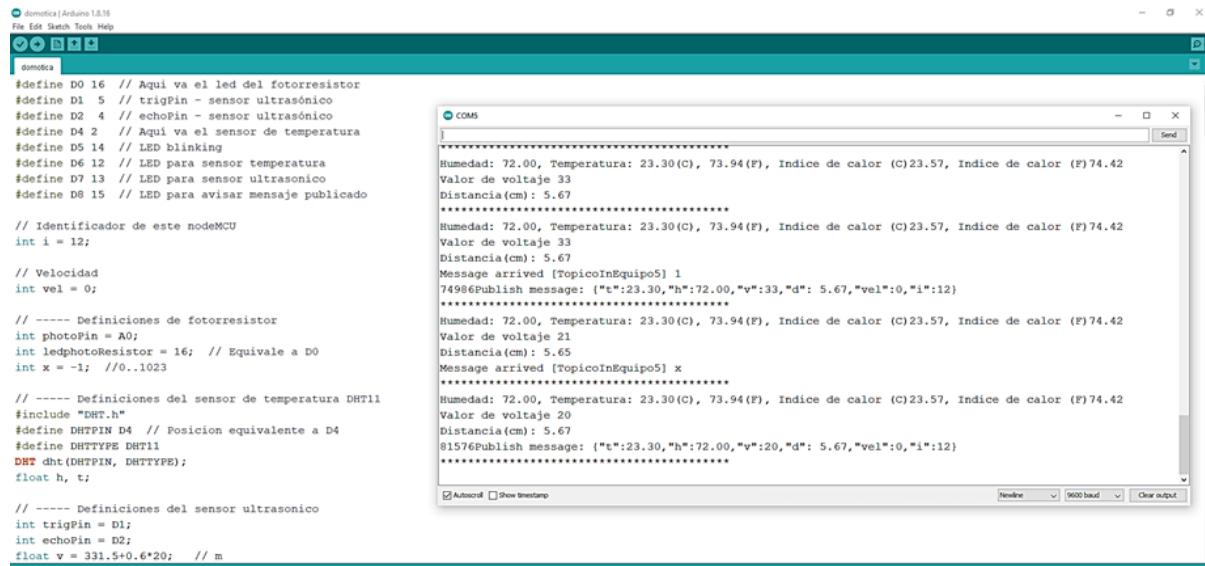
ctx: cont
sp: 3ffffdf0 end: 3fffffc0 offset: 0190
3fffff80: 3fffee7c0 4269ec6b 3fffee814 4020164d
3fffff90: 3ffffdad0 00000000 3fffee890 4020171d
3fffffa0: feefeffe feefeffe 3fffee890 40204cc0
3fffffb0: feefeffe feefeffe 3ffe8600 40100c11
<<<stack<<<


```

Así mismo, al intentar por primera vez conectar el NodeMCU al broker de MQTT por Internet, la placa se conectaba y desconectaba en cada iteración. Al investigar un poco sobre el error, no pudimos comprender la manera de solucionarlo. Sin embargo, en una clase con el profesor Sergio, al externar nuestras dudas y escuchar las de nuestros compañeros, pudimos comprender cuál era nuestro error, el cuál era que una función estaba declarada de tipo flotante, pero en realidad era de tipo void.

5.4. Resultados

5.4.1. IDE Arduino



5.4.2. Base de datos

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0002 segundos.)

```
SELECT * FROM `nodemcu`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar

+ Opciones

	<input type="button" value="←"/>	<input type="button" value="→"/>		node_id	node_chip	node_user
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	12	ESP8266	Brenda
<input type="checkbox"/>	<input type="button" value="Editar"/>	<input type="button" value="Copiar"/>	<input type="button" value="Borrar"/>	23	ESP8266	Karen

Mostrando filas 0 - 24 (total de 287, La consulta tardó 0,0002 segundos.)

```
SELECT * FROM `luz`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [

1 < > >> | Mostrar todo

+ Opciones

node_id	luz_fecha	luz_voltaje
12	2021-12-01 18:49:07	24
12	2021-12-01 18:49:13	24
12	2021-12-01 18:49:20	30
12	2021-12-01 18:49:26	30
12	2021-12-01 18:49:33	30
12	2021-12-01 18:49:59	30
12	2021-12-01 18:50:05	18
12	2021-12-01 18:50:12	19
12	2021-12-01 18:50:19	21
12	2021-12-01 18:50:25	30
23	2021-12-02 09:36:45	158
23	2021-12-02 09:36:50	99
23	2021-12-02 09:36:56	105
23	2021-12-02 09:37:01	100
23	2021-12-02 09:37:06	123
23	2021-12-02 09:37:11	158
12	2021-12-02 10:40:08	38

Consola

Mostrando filas 0 - 24 (total de 287, La consulta tardó 0,0002 segundos.)

```
SELECT * FROM `movimiento`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [

1 < > >> | Mostrar todo

+ Opciones

node_id	mov_fecha	mov_dist
12	2021-12-01 18:49:07	67.74
12	2021-12-01 18:49:13	68.17
12	2021-12-01 18:49:20	67.74
12	2021-12-01 18:49:26	68.15
12	2021-12-01 18:49:33	67.74
12	2021-12-01 18:49:59	68.17
12	2021-12-01 18:50:05	68.6
12	2021-12-01 18:50:12	68.6
12	2021-12-01 18:50:19	66
12	2021-12-01 18:50:25	1209.02
23	2021-12-02 09:36:45	75.36
23	2021-12-02 09:36:50	14.46
23	2021-12-02 09:36:56	68.03
23	2021-12-02 09:37:01	14.58
23	2021-12-02 09:37:06	67.19
23	2021-12-02 09:37:11	72.08
12	2021-12-02 10:40:08	12.8

Consola

The image shows two side-by-side MySQL query results. Both queries return 287 rows.

Left Query:

```
SELECT * FROM `servomotor`
```

Right Query:

```
SELECT * FROM `temperatura`
```

node_id	ser_fecha	ser_vel
12	2021-12-01 18:49:07	0
12	2021-12-01 18:49:13	0
12	2021-12-01 18:49:20	0
12	2021-12-01 18:49:26	0
12	2021-12-01 18:49:33	0
12	2021-12-01 18:49:59	0
12	2021-12-01 18:50:05	0
12	2021-12-01 18:50:12	0
12	2021-12-01 18:50:19	0
12	2021-12-01 18:50:25	0
23	2021-12-02 09:36:45	30
23	2021-12-02 09:36:50	30
23	2021-12-02 09:36:56	30
23	2021-12-02 09:37:01	30
23	2021-12-02 09:37:06	30
23	2021-12-02 09:37:11	30
12	2021-12-02 10:40:08	0
23	2021-12-02 10:40:10	30
12	2021-12-02 10:40:14	0
12	2021-12-02 10:40:15	30

node_id	temp_fecha	temp_temp	temp_humedad
12	2021-12-01 18:49:07	22.2	67
12	2021-12-01 18:49:13	22.2	67
12	2021-12-01 18:49:20	22.2	67
12	2021-12-01 18:49:26	22.2	67
12	2021-12-01 18:49:33	22.3	67
12	2021-12-01 18:49:59	22.3	66
12	2021-12-01 18:50:05	22.4	66
12	2021-12-01 18:50:12	22.4	66
12	2021-12-01 18:50:19	22.4	66
12	2021-12-01 18:50:25	22.4	66
23	2021-12-02 09:36:45	24.8	63
23	2021-12-02 09:36:50	24.6	63
23	2021-12-02 09:36:56	24.2	63
23	2021-12-02 09:37:01	24.2	63
23	2021-12-02 09:37:06	24.4	63
23	2021-12-02 09:37:11	24.6	63
12	2021-12-02 10:40:08	21.7	69

6. Conclusión

Como conclusión, el haber realizado un proyecto para impulsar las viviendas inteligentes, nos ayudó a identificar diversos recursos, elementos y componentes que lo integran. En el desarrollo del proyecto nos topamos con varios inconvenientes, algunos de ellos los pudimos corregir, y otros no, pero aun así pudimos sacar adelante nuestro proyecto, inclusive el poder adaptarlo para dos versiones de dispositivos, en donde a cada uno se le incluyeron o removieron ciertos aspectos, pero que de alguna manera trabajan para el mismo propósito. Esto nos permitió tener buenos resultados en donde se puede identificar los datos que son adquiridos de cada una de las placas NodeMCU. Gracias a este proyecto, pudimos aplicar los conocimientos adquiridos en las clases sobre Internet de las Cosas y Bases de Datos, en un prototipo físico que representa las funcionalidades requeridas. Los aprendizajes alcanzados nos van a ser de gran ayuda y utilidad para futuros proyectos, así como en la vida profesional.

7. Reflexión Individual

Reflexión individual escrita por cada estudiante en su propio documento

Del aprendizaje del reto (mínimo 5 líneas)

De la forma de colaboración o trabajo en equipo (mínimo 5 líneas)

8. Referencias bibliográficas

Oracle. (s.f.). ¿Qué es el IoT?. Noviembre 26, 2021, de Oracle. Sitio web:

<https://www.oracle.com/mx/internet-of-things/what-is-iot/>

Arenas, M. (2016). Internet de las Cosas: Ciberseguridad. Noviembre 26, 2021, de OWASP. Sitio web: https://owasp.org/www-pdf-archive/IoT_CyberSecurity.pdf

Recuero, P. (2020). Breve historia de Internet de las cosas (IoT). Noviembre 26, 2021, de Telefónica Tech. Sitio web:

<https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas-iot/>

Ing. Dávalos, R. (2019). Tema 6.1 Introducción a IoT. Noviembre 26, 2021, de ITESM. Sitio web: <https://experiencia21.tec.mx/courses/182677/files/63749403?wrap=1>

Anónimo. (s.f.). <http://www.cedom.es/sobre-domotica/que-es-domotica>. Noviembre 28, 2021, de Asociación Española de Domótica e Inmótica - CEDOM. Sitio web: <http://www.cedom.es/sobre-domotica/que-es-domotica>

Placa NodeMCU v3 - ESP8266 / CH340G Wifi BricoGeek CH340-V3 |

BricoGeek.com. (2020, 14 diciembre). BricoGeek. Recuperado 2 de diciembre de 2021, de <https://tienda.bricogek.com/wifi/1033-nodemcu-v3-wifi-esp8266-ch340.html>

Definición de sensor — Definicion.de. (s. f.). Definición.de. Recuperado 2 de diciembre de 2021, de <https://definicion.de/sensor/>

¿Qué es actuador? Definición, concepto y significado. (2015, 23 diciembre).

DiccionarioActual. Recuperado 2 de diciembre de 2021, de

<https://diccionarioactual.com/actuador/>

L. (2019, 17 abril). ¿Qué es MQTT? Su importancia como protocolo IoT. Luis Llamas. Recuperado 2 de diciembre de 2021, de

<https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

J. (2020, 5 marzo). Qué es Node-RED. Aprendiendo Arduino. Recuperado 2 de diciembre de 2021, de

<https://aprendiendoarduino.wordpress.com/2020/03/05/que-es-node-red/>

Oracle | What is a Data Base? (s. f.). Oracle. Recuperado 2 de diciembre de 2021, de <https://www.oracle.com/mx/database/what-is-database/>

Colaboradores de Wikipedia. (s.f.). Botón (dispositivo). Diciembre 3, 2021, de Wikipedia. Sitio web: [https://es.wikipedia.org/wiki/Bot%C3%B3n_\(dispositivo\)](https://es.wikipedia.org/wiki/Bot%C3%B3n_(dispositivo))

AbrahamG. (2019). Utilizar Push Button con Arduino. Diciembre 3, 2021, de Automatización para Todos. Sitio web:

<https://www.automatizacionparatodos.com/push-button-con-arduino/>

Colaboradores de Wikipedia. (s.f.). Fotorresistor. Diciembre 3, 2021, de Wikipedia.

Sitio web: <https://es.wikipedia.org/wiki/Fotorresistor>

Luis. (2017). Fotorresistencia: Definición, características y tipos. Diciembre 3, 2021, de Ingeniería Electrónica. Sitio web:

<https://ingenieriaelectronica.org/fotorresistencia-definicion-caracteristicas-y-tipos/>

Anónimo. (s.f.). ¿Qué son los LEDs?. Diciembre 3, 2021, de LEDs International. Sitio web: <https://www.ledsinternational.com/es/que-son-los-leds/>

Colaboradores de Wikipedia. (s.f.). Led. Diciembre 3, 2021, de Wikipedia. Sitio web: <https://es.wikipedia.org/wiki/Led>

Anónimo. (s.f.). SERVOMOTORES. Diciembre 3, 2021, de AREA TECNOLOGIA.

Sitio web: <https://www.atecnologia.com/electricidad/servomotor.html>

Ingeniería Mecafenix. (2017). Servomotor ¿Que es y como funciona?. Diciembre 3, 2021, de Ingeniería Mecafenix. Sitio web:

<https://www.ingmecafenix.com/electricidad-industrial/servomotor/>