

**Aplicacion  
es de IoT**

# **PRÁCTICA THINGSBOARD**

**Realizado por:**

Bahena Castillo Luis Eduardo  
Estrada Hernández Andrea Michelle  
Garcia Araujo Ximena Natalia  
Hernandez De la Cruz Axel

**5A DSM**

## ÍNDICE

<b>Introducción</b>	3
<b>Objetivo</b>	3
<b>Materiales</b>	3
<b>Desarrollo</b>	5
• Esquema del Circuito	6
• Código fuente en Arduino	6
• Construcción del Circuito	10
<b>Análisis del Resultado</b>	12
<b>Conclusión</b>	13

## Introducción

Diseñar un sistema físico de monitoreo y control para una casa inteligente, donde se puedan controlar el encendido de leds dependiendo la luz de la habitación y la presencia de objetos, además de la temperatura y humedad para enviar alertas.

Mostrar en ThingsBoard los datos de los sensores (un widget por cada sensor), además de dos leds indicadores, uno que encienda cuando la humedad sea menor a 50% y el otro cuando la luz sea menor al 25 % y el sensor de distancia detecte movimiento a menos de 20 cm.

## Objetivo

Comunicación a través de WiFi con Arduino y el servidor de ThingsBoard.

## Materiales



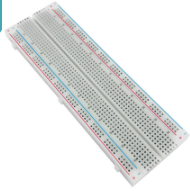
Resistencia

Es una medida de la oposición al flujo de corriente en un circuito eléctrico. La resistencia se mide en ohmios, que se simbolizan con la letra griega omega ( $\Omega$ ).



LED

"Light Emitting Diode" es un elemento capaz de recibir una corriente eléctrica moderada y emitir una radiación electromagnética transformada en luz.



Protoboard

Prácticamente una PCB temporal con una forma y tamaño generalizados. Utilizada comúnmente para pruebas y prototipos temporales de circuitos. Se usa insertando las terminales de los dispositivos electrónicos en los orificios de la protoboard de la forma en que tengan continuidad.

El cable jumper, o cable puente, suele venir en multitud de kits de electrónica, desde algunos de robótica, hasta los de Arduino, etc. Además, son cables muy prácticos para un gran número de proyectos electrónicos. Variaciones de cables con extremos homogéneos o heterogéneos.

Es decir:

- Hembra-hembra en los dos extremos.
- Hembra-macho.
- Macho-macho en ambos extremos.



JumpersMM



JumpersMH



Módulo Wifi

Es un chip integrado con conexión WiFi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red. Son dispositivos minimizados que permiten la comunicación inalámbrica entre dispositivos. Cuando se utilizan ondas de radio, garantizan la transmisión y recepción de la señal.

Es un dispositivo utilizado en espacios de interior a fin de controlar la temperatura y humedad del aire. Las magnitudes medidas por el sensor son transformadas en una señal eléctrica (con una intensidad entre 4 y 20 mA).



Sensor Humedad



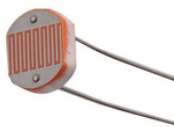
Sensor  
Temperatura

Es un sensor que nos dará valores analógicos de temperatura, convirtiendo la temperatura en un voltaje análogo. Es de bajo costo y ofrece bajo consumo, con una precisión de  $\pm 4^{\circ}\text{C}$  desde  $0^{\circ}\text{C}$  a  $+70^{\circ}\text{C}$ . Este sensor provee una solución de bajo costo para aplicaciones que requieran mediciones del cambio de temperatura relativa.

Es un circuito que detecta o mide la distancia y es compatible con Arduino. Básicamente, el sensor puede detectar objetos, distancia o nivel en un rango mínimo de 2 cm a un máximo de 400 cm.



Sensor Ultrasonico



Fotoresistencia

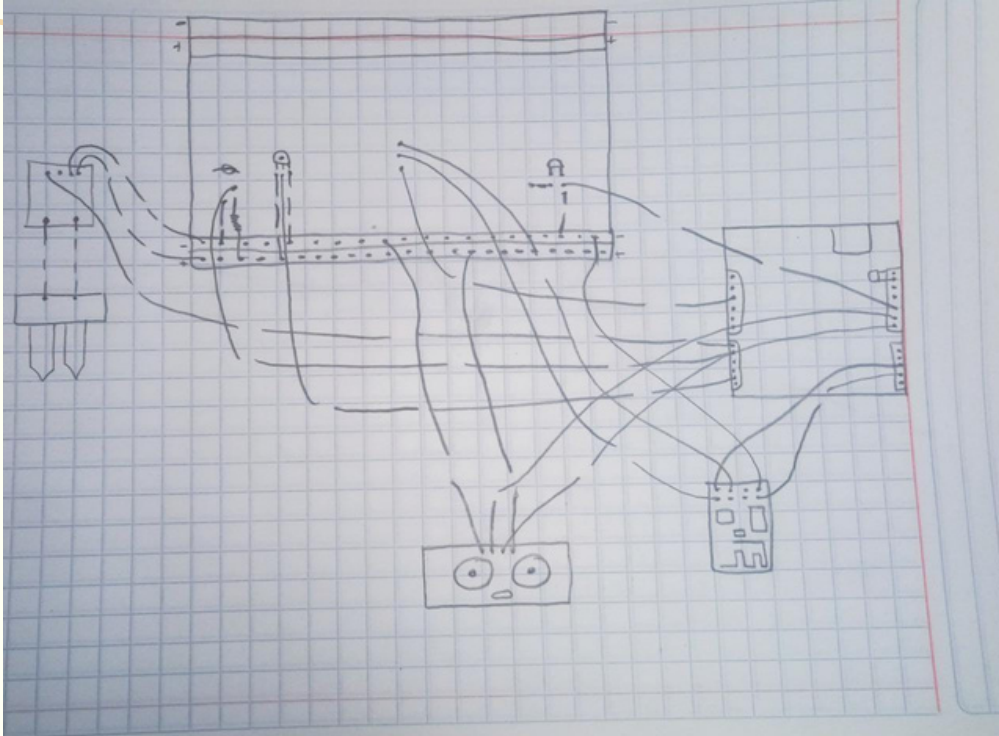
Componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente.

## Desarrollo



Para realizar esta práctica, vamos a intentar conectar de Arduino a la página de ThingsBoard con la verificación de un token, así para acceder a los valores de los sensores y mostrar esos mismos en los widgets correspondientes de la plataforma

- Esquema del Circuito



- Código Fuente Arduino

```
#include <WiFiEspClient.h>
#include <WiFiEsp.h>
#include "SoftwareSerial.h"

#define WIFI_AP          "LuisVad"
#define WIFI_PASSWORD    "laurasefue"

// See https://thingsboard.io/docs/getting-started-guides/helloworld/
// to understand how to obtain an access token
#define TOKEN            "1KJXMRc3tOKIjVAMrtYB"
#define THINGSBOARD_SERVER "thingsboard.cloud"

const int humedad = 0;
const int luz = A1;
const int led1 = 13;
const int led2 = 10;
const int Trigger = 8;
const int Echo = 9;
float tempC ;
int entTemp = A4;
int hum2, luz2;
int distancia , time , valor =0;
bool humFinal = false;
bool luzFinal = false;
```

```
#define SERIAL_DEBUG_BAUD    9600
// Baud rate for communicating with ESP chip
#define SERIAL_ESP8266_BAUD 9600

// Serial driver for ESP
SoftwareSerial soft(2, 3); // RX, TX
// Initialize the Ethernet client object
WiFiEspClient espClient;
// Initialize ThingsBoard instance
ThingsBoard tb(espClient);

void setup() {
    uint8_t wifiStatus;

    // initialize serial for debugging
    Serial.begin(SERIAL_DEBUG_BAUD);
    // initialize serial for ESP module
    soft.begin(SERIAL_ESP8266_BAUD);

    // initialize ESP module
    WiFi.init(&soft);
    // check for the presence of the shield
    if (WiFi.status() == WL_NO_SHIELD) {
        Serial.println("WiFi shield not present");
        // don't continue
        while (true);
    }
}
```

```
Serial.println("Connecting to AP ...");
// attempt to connect to WiFi network
while (wifiStatus != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(WIFI_AP);
    // Connect to WPA/WPA2 network
    wifiStatus = WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    delay(500);
}

Serial.println("Connected to AP");

pinMode(humedad, INPUT);
pinMode(luz, INPUT);
pinMode(Triquer, OUTPUT);
pinMode(Echo, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
}
```



```
void loop() {  
    delay(1000);  
    if (!tb.connected()) {  
        // Connect to the ThingsBoard  
        Serial.print("Connecting to: ");  
        Serial.print(THINGSBOARD_SERVER);  
        Serial.print(" with token ");  
        Serial.println(TOKEN);  
        if (!tb.connect(THINGSBOARD_SERVER, TOKEN)) {  
            Serial.println("Failed to connect, retrying ...");  
            return;  
        }  
    }  
  
    Serial.println("Sending telemetry data...");  
  
    int temp = analogRead(luz);  
    int hum = analogRead(humedad);  
  
    const int data_items = 4;
```

```
while(temp !=50 && hum !=100){  
  
    //Ultra  
    digitalWrite(Triquer, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(Triquer,LOW);  
  
    time = pulseIn(Echo, HIGH);  
    distancia = time/2 * 0.034;  
  
    //Temp  
    tempC = analogRead(entTemp);  
    tempC = (5.0* tempC * 100.0) /1024.0;  
  
    //Luz y Hum  
    temp = analogRead(luz);  
    hum = analogRead(humedad);  
    luz2 = map(temp,0,1023, 0, 100);  
    hum2 = map(hum, 0 ,1023, 100, 0);  
  
    if(hum2 < 50){  
        humFinal = true;  
        digitalWrite(led1, HIGH);  
    }else{  
        humFinal= false;  
        digitalWrite(led1, LOW);  
    }  
}
```



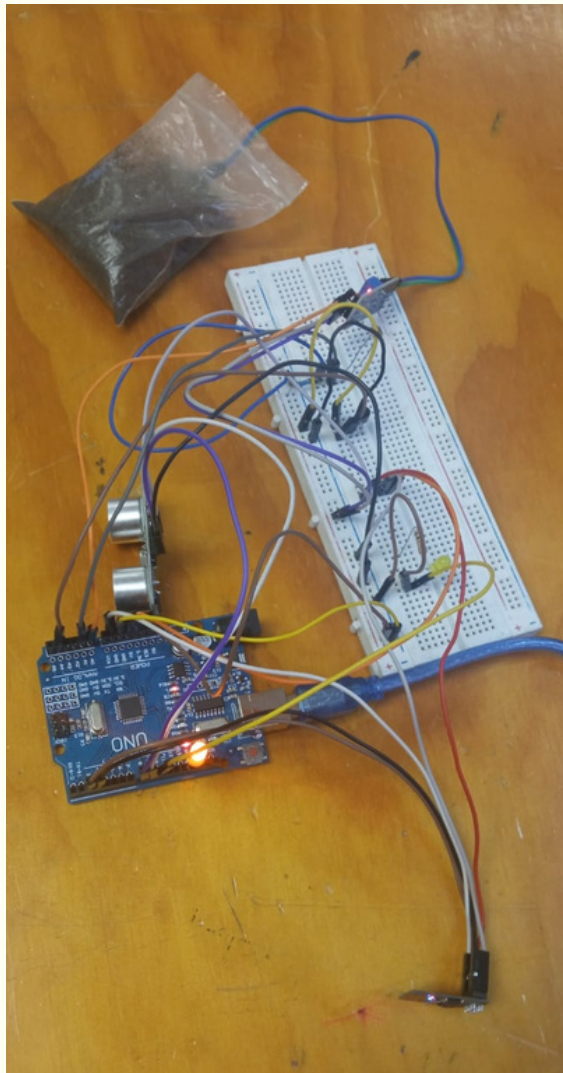
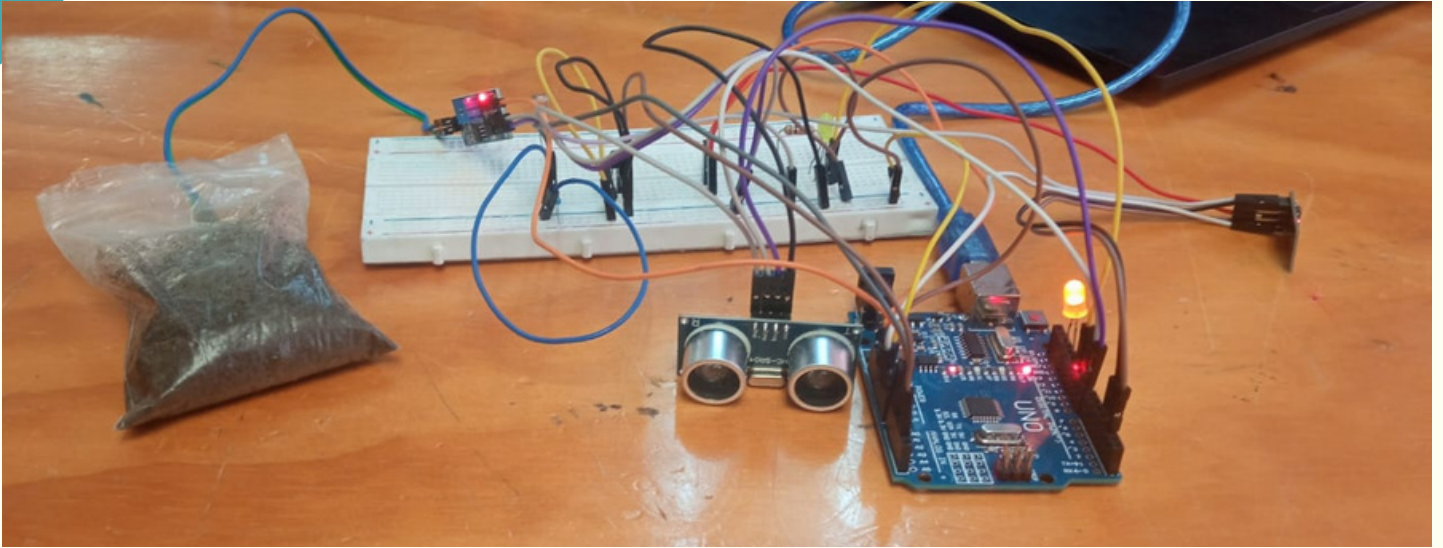
```
if( distancia <20 && luz2 <50 ){
    luzFinal = true;
    digitalWrite(led2, HIGH);
}else{
    luzFinal = false;
    digitalWrite(led2, LOW);
}

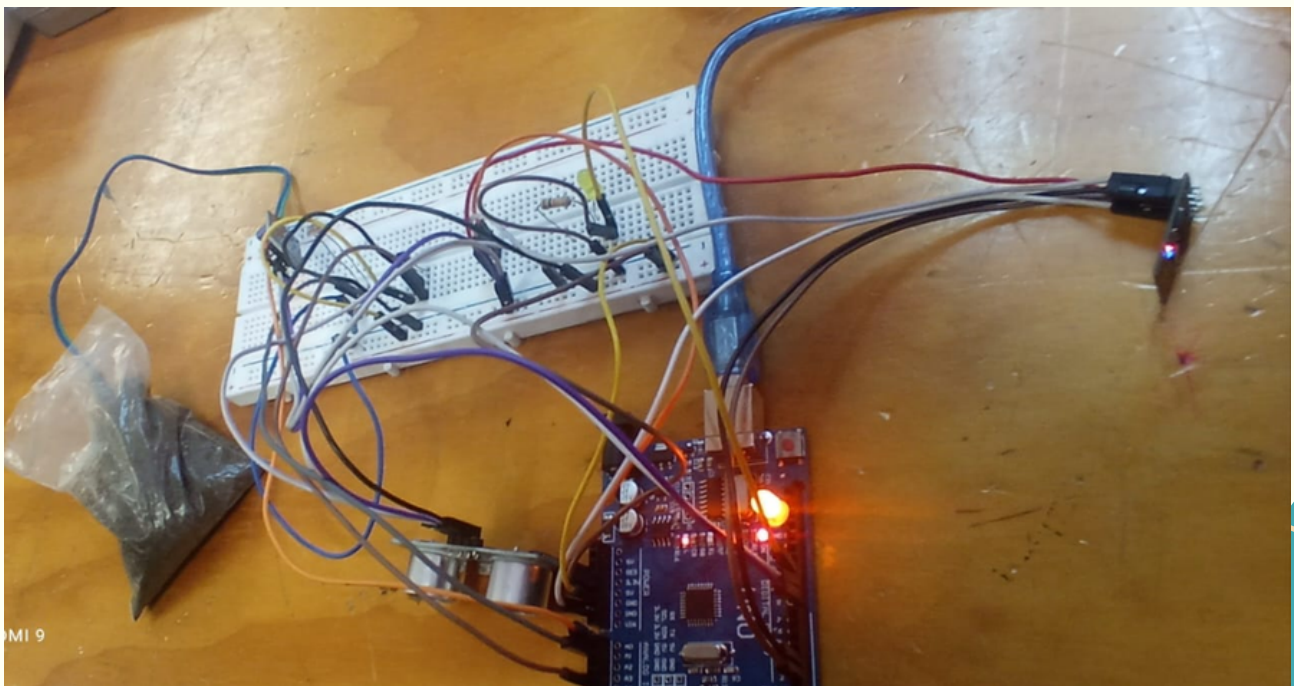
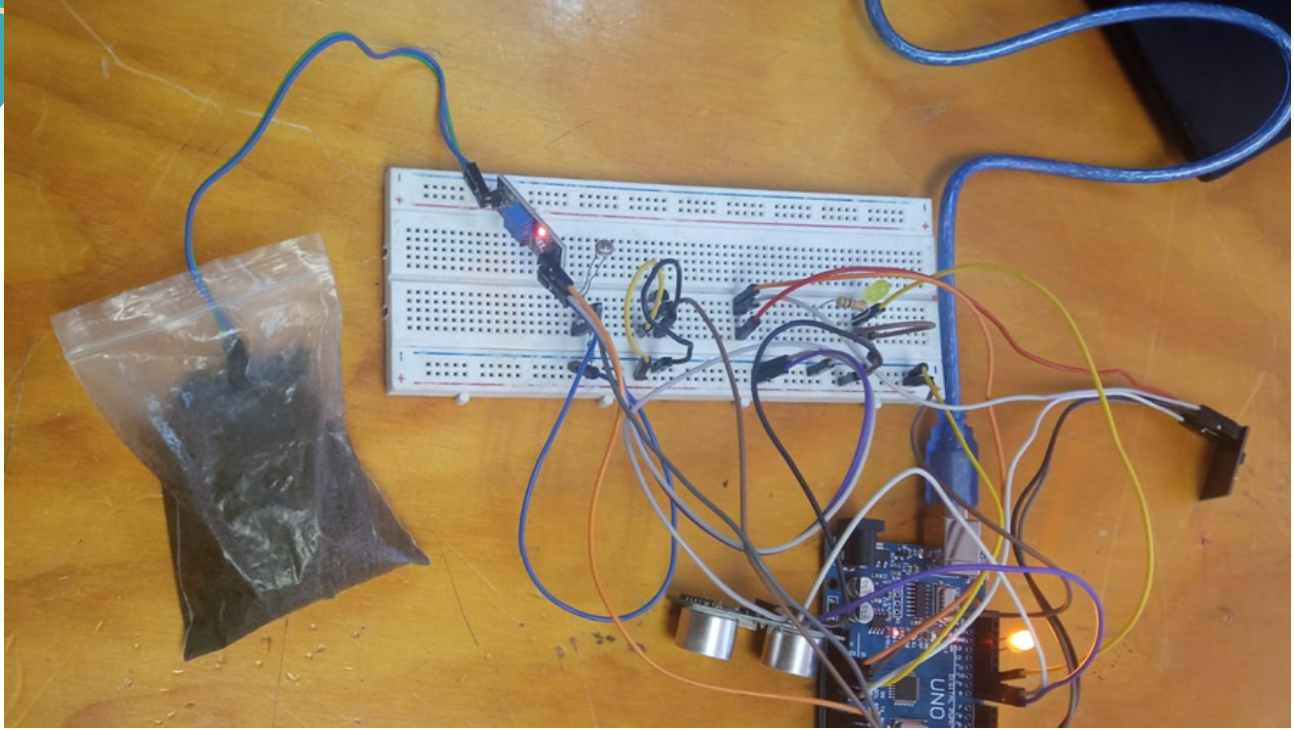
tb.sendAttributeBool("humFinal", humFinal);
tb.sendAttributeBool("luzFinal", luzFinal);

Telemetry data[data_items] = {
    { "light", luz2 },
    { "temperature", tempC },
    { "humidity", hum2 },
    { "ultrasonic", distancia },
};

tb.sendTelemetry(data, data_items);
}
tb.loop();
}
```

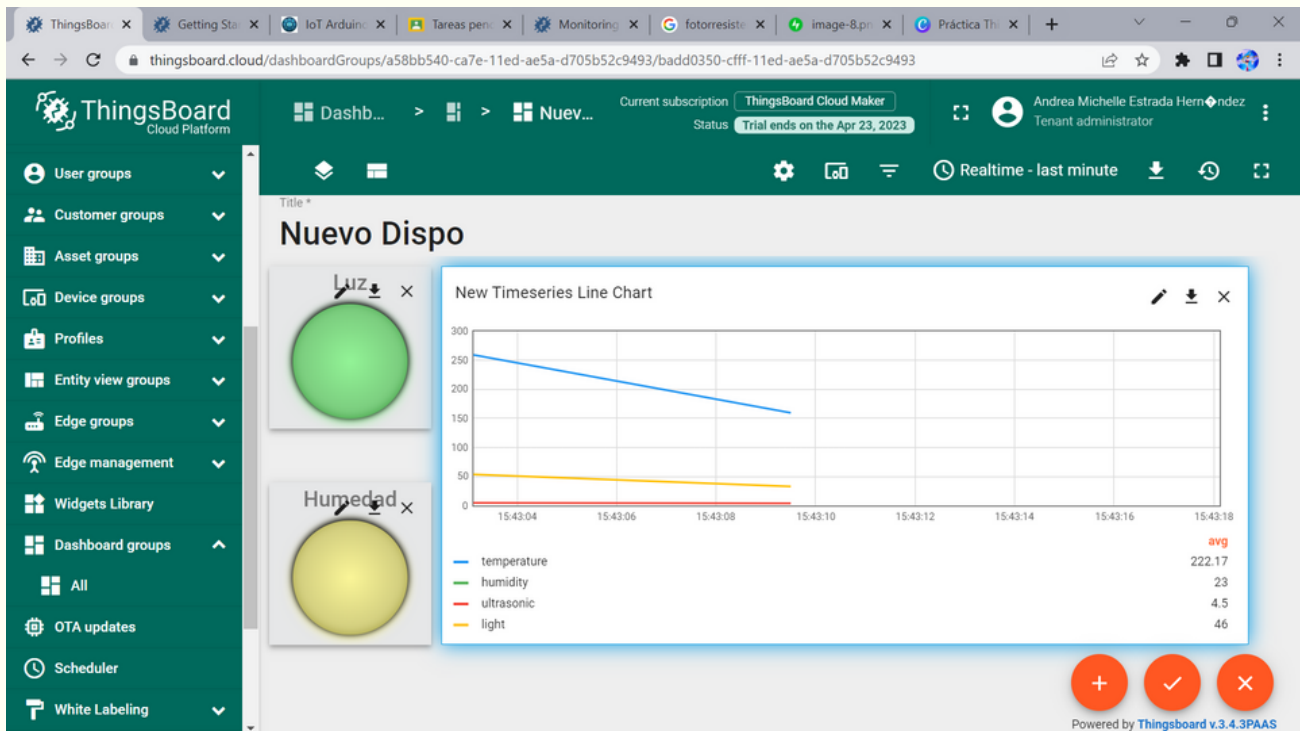
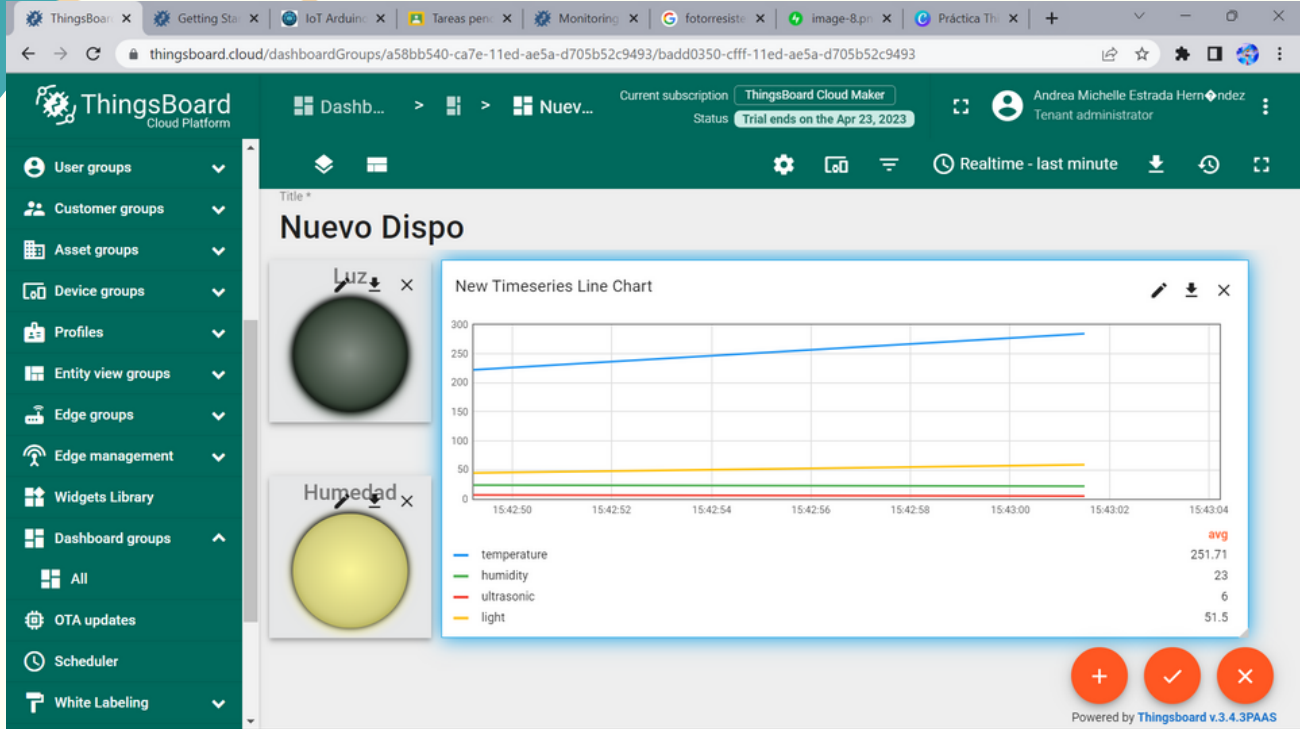
- Construcción del Circuito







## Análisis del Resultado:



## **Conclusión:**

Consideramos que fue un reto o más bien una práctica compleja ya que se usó nuevamente el módulo de wifi, pero en esta ocasión sin el uso total de los comandos AT por lo que la configuración con el ThingsBoard tenía que ser más detallada, usando el modo 3 y una red estable.

Dentro de la lógica del programa, no hubo tanta dificultad pues eran prácticas ya realizadas, sólo lo adaptamos a las indicaciones establecidas y con la sintaxis de esta misma plataforma.

Fué un verdadero reto pero se logró. Se demostró la capacidad de diseñar un sistema de control y monitoreo físico para una casa inteligente, capaz de ajustar la iluminación de los LED en función de los datos del sensor y también enviar alertas cuando sea necesario. Además, la plataforma ThingsBoard es una herramienta útil para la visualización de los datos en tiempo real.