



## A.3.3 Actividad de aprendizaje

---

Circuito de medición de temperatura a través de un NodeMCU ESP32



---

### Instrucciones

- Basado en la figura 1, ensamblar un sistema, capaz de detectar la temperatura y humedad del ambiente, a través de un circuito electrónico, utilizando un NodeMCU **ESP32**, y un **Sensor DHT11/DHT22**.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A3.3\_NombreApellido\_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
- readme.md
- blog
  - C3.1_TituloActividad.md
  - C3.2_TituloActividad.md
  - C3.3_TituloActividad.md
  - C3.4_TituloActividad.md
  - C3.5_TituloActividad.md
  - C3.6_TituloActividad.md
  - C3.7_TituloActividad.md
  - C3.8_TituloActividad.md
- img
- docs
  - A3.1_TituloActividad.md
  - A3.2_TituloActividad.md
  - A3.3_TituloActividad.md
```

Fuentes de apoyo para desarrollar la actividad:

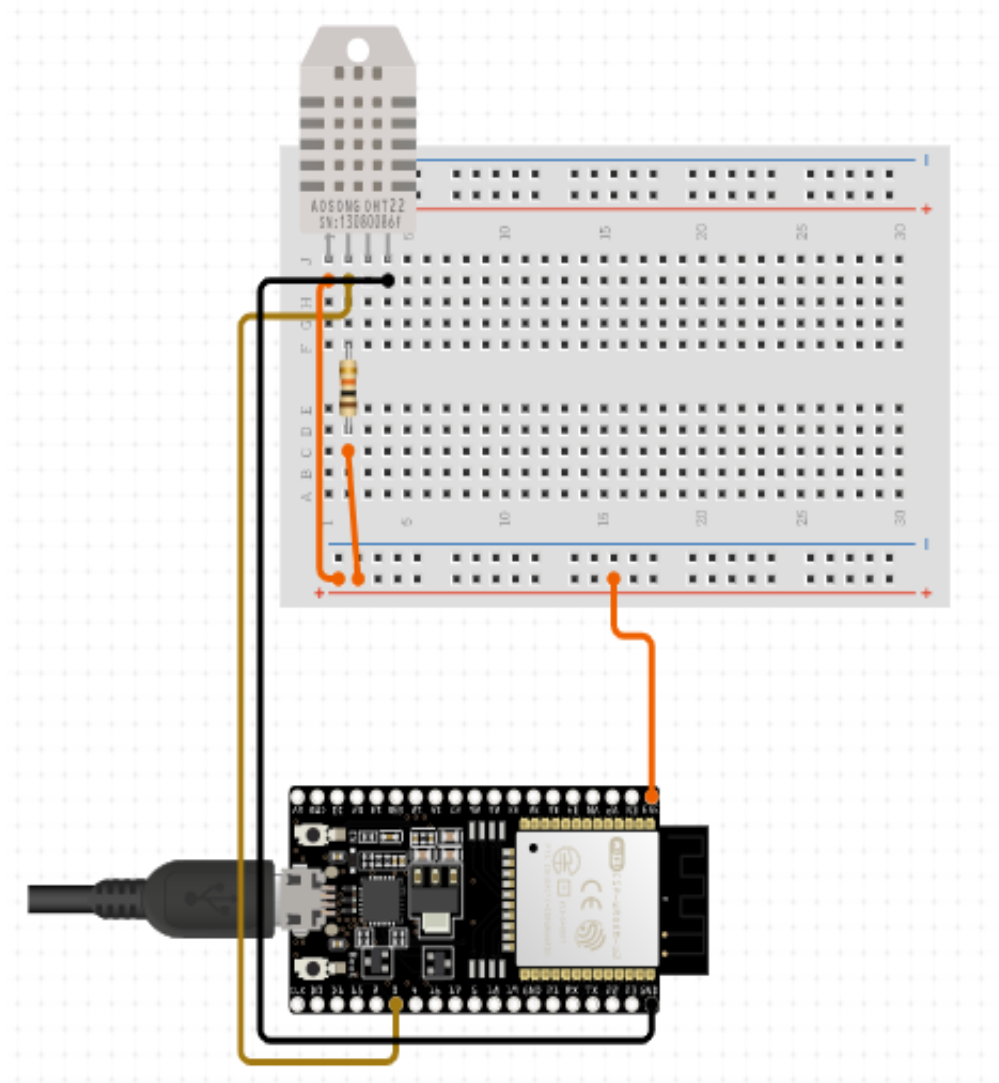
-  [Random Nerd Tutorial Touch pin](#)
  -  [Ejemplo de circuito con LED RGB Led](#)
- 

## Desarrollo

1. Utilice el siguiente listado de materiales para la elaboración de la actividad

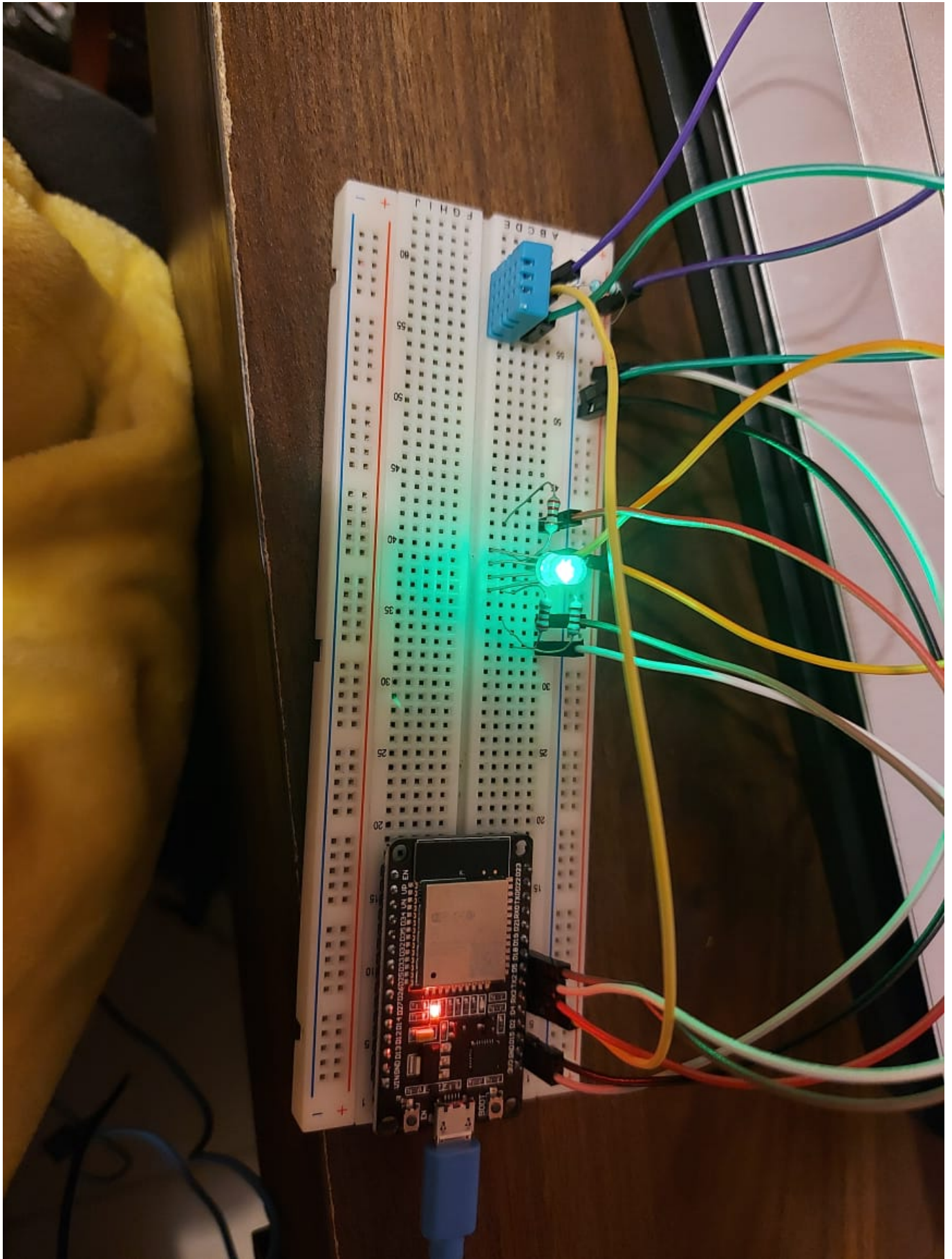
Cantidad	Descripción
1	<a href="#">Sensor temperatura y humedad DHT11 o DHT22</a>
1	<a href="#">Diodo led RGB</a>
1	<a href="#">Resistencia 4.7k ohms</a>
3	<a href="#">Resistencias 1k ohm</a>
1	<a href="#">Fuente de voltaje de 5V</a>
1	<a href="#">NodeMCU ESP32</a>
1	<a href="#">BreadBoard</a>
1	<a href="#">Jumpers M/M</a>

1. Basado en las imágenes que se muestran en la **Figuras 1**, ensamble el circuito en un solo circuito electrónico, ide tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones antes solicitadas para esta actividad.

**Figura 1 Circuito ESP32 y Sensor DHT**

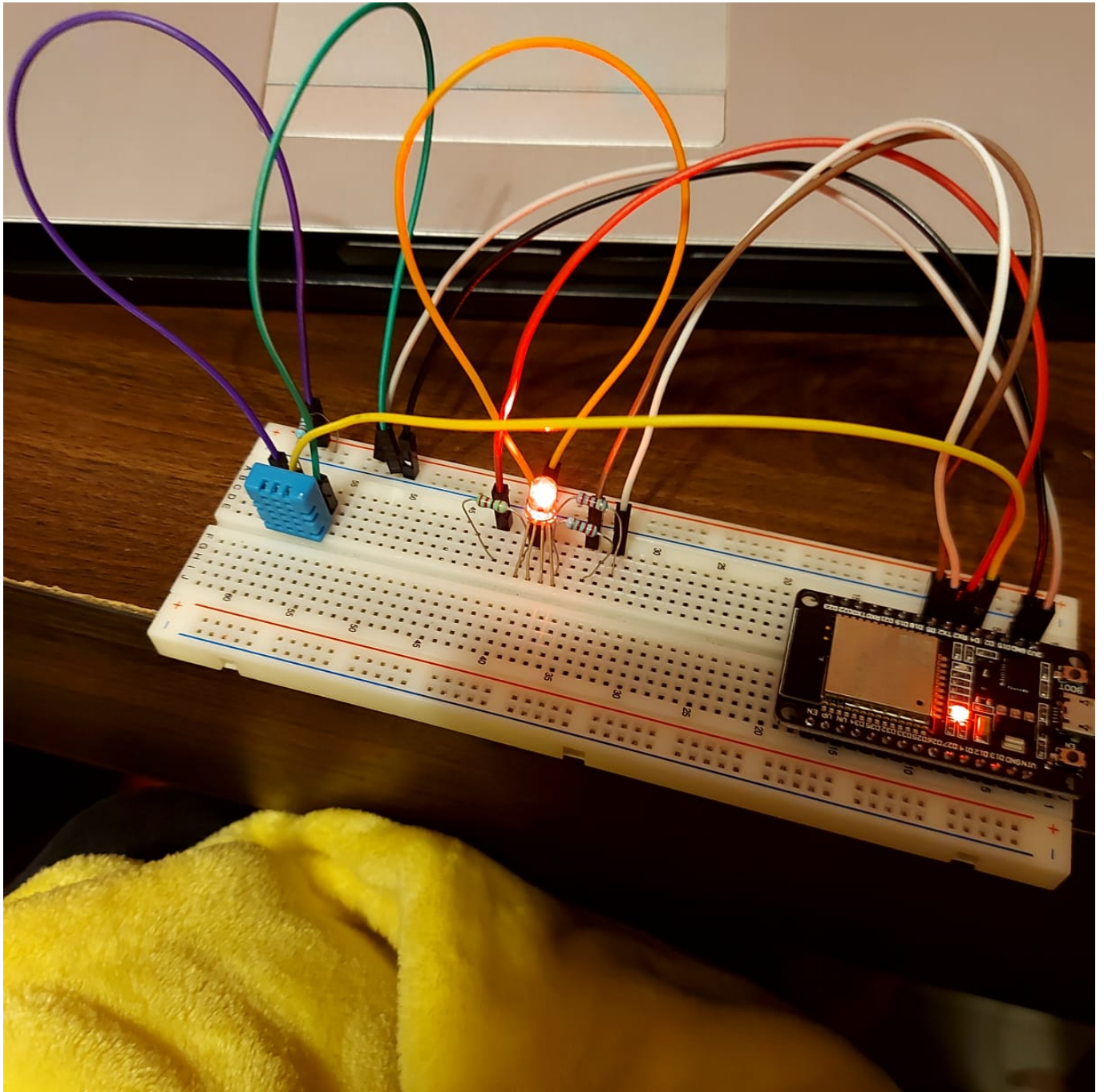
1. Una vez ensamblado el circuito anterior, agregue un LED RGB y elabore el programa que le permita al LED RGB funcionar como indicador para las siguientes condiciones:
  - El sensor de temperatura en todo momento estará sensando, enviando el valor registrado por la terminal serial, por ejemplo "Temperatura ambiente: 25 grados" y el **LED RGB** estará encendido de color verde.
  - El sensor de temperatura al registrar un valor del ~20% por arriba de la temperatura ambiental, deberá mostrar el mensaje "Temperatura alta: ? grados" y el **LED RGB** se encenderá de color rojo.
  - El sensor de temperatura al registra un valor del ~20% por abajo de la temperatura ambiental, deberá mostrar el mensaje "Temperatura baja: ? grados" y el **LED RGB** se encenderá de color azul.
2. Coloque aquí evidencias que considere importantes durante el desarrollo de la actividad.

## Circuito armado y funcionando

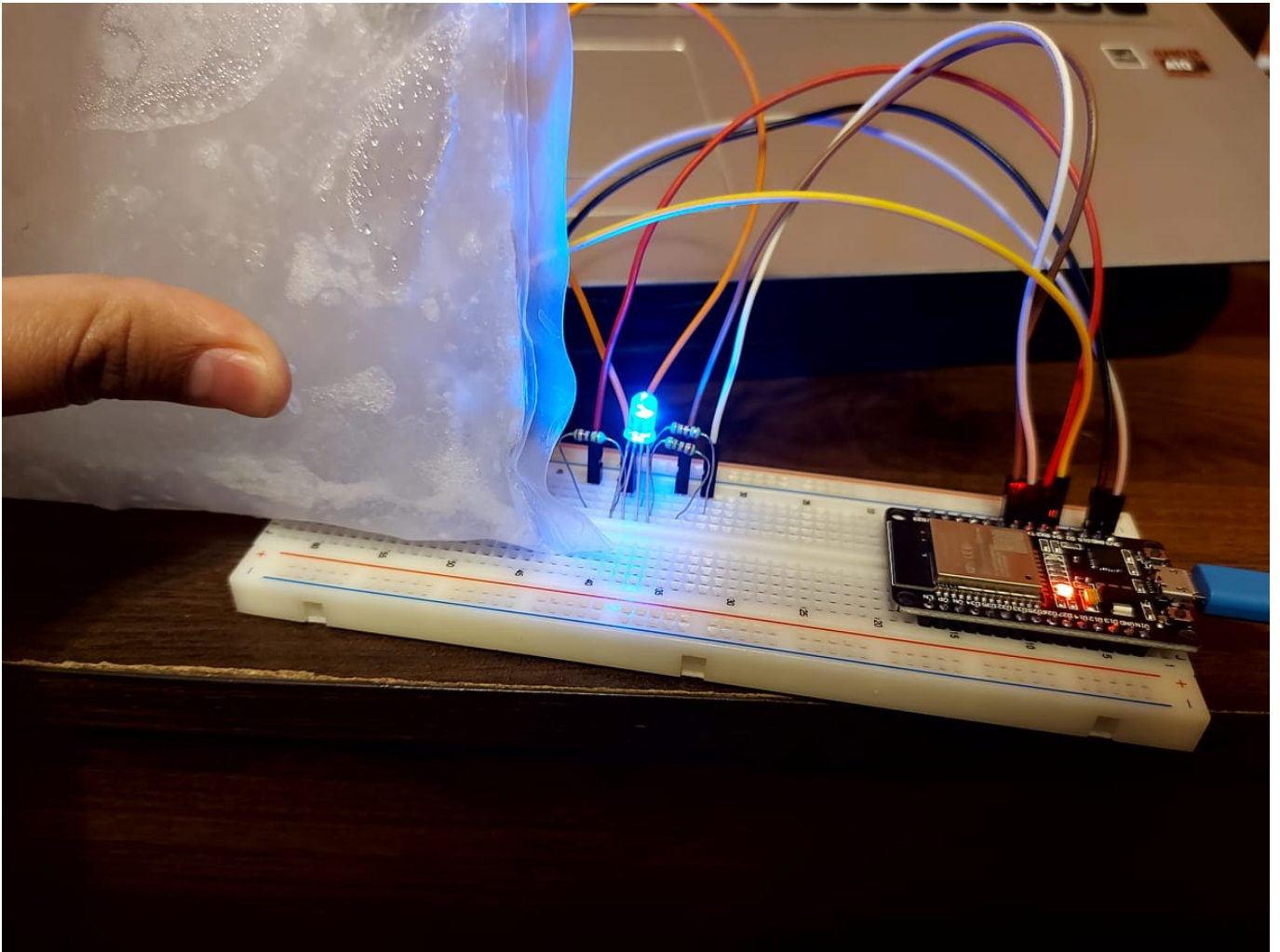


Temperatura ambiental





Temperatura Alta



Temperatura Baja

Codigo



```

#include "DHT.h" //Libreria

//pines que vamos a usar
const int led1 = 16;
const int led2 = 17;
const int led3 = 5;

//Establecemos los pines PWM
const int freq = 5000;
const int resolution = 8;

//Canales
const int red = 1;
const int green = 2;
const int blue = 3;

//Declaramos el pin del sensor
#define DHTPIN 4
//Declaramos el sensor que se utilizara
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {

  Serial.begin(115200);
  //Inicializamos la libreria dht
  dht.begin();
  //Se configura la señal PWM
  ledcSetup(red, freq, resolution);
  ledcSetup(green, freq, resolution);
  ledcSetup(blue, freq, resolution);
  //Se le asigna la señal al GPIO que estemos utilizando
  ledcAttachPin(led1, red);
  ledcAttachPin(led2, green);
  ledcAttachPin(led3, blue);

},

void loop() {
  // put your main code here, to run repeatedly:
  delay(1000);
  //Se lee el sensor
  float celsius = dht.readTemperature();
  //Si la temperatura esta entre 20 y 30 grados es normal
  if (celsius >= 20 && celsius <=30)
  {
    Serial.print("Temperatura Normal ");
    ledcWrite(red, 0);
    ledcWrite(green, 256);
  }
}

```



```
    ledcWrite(blue, 0);

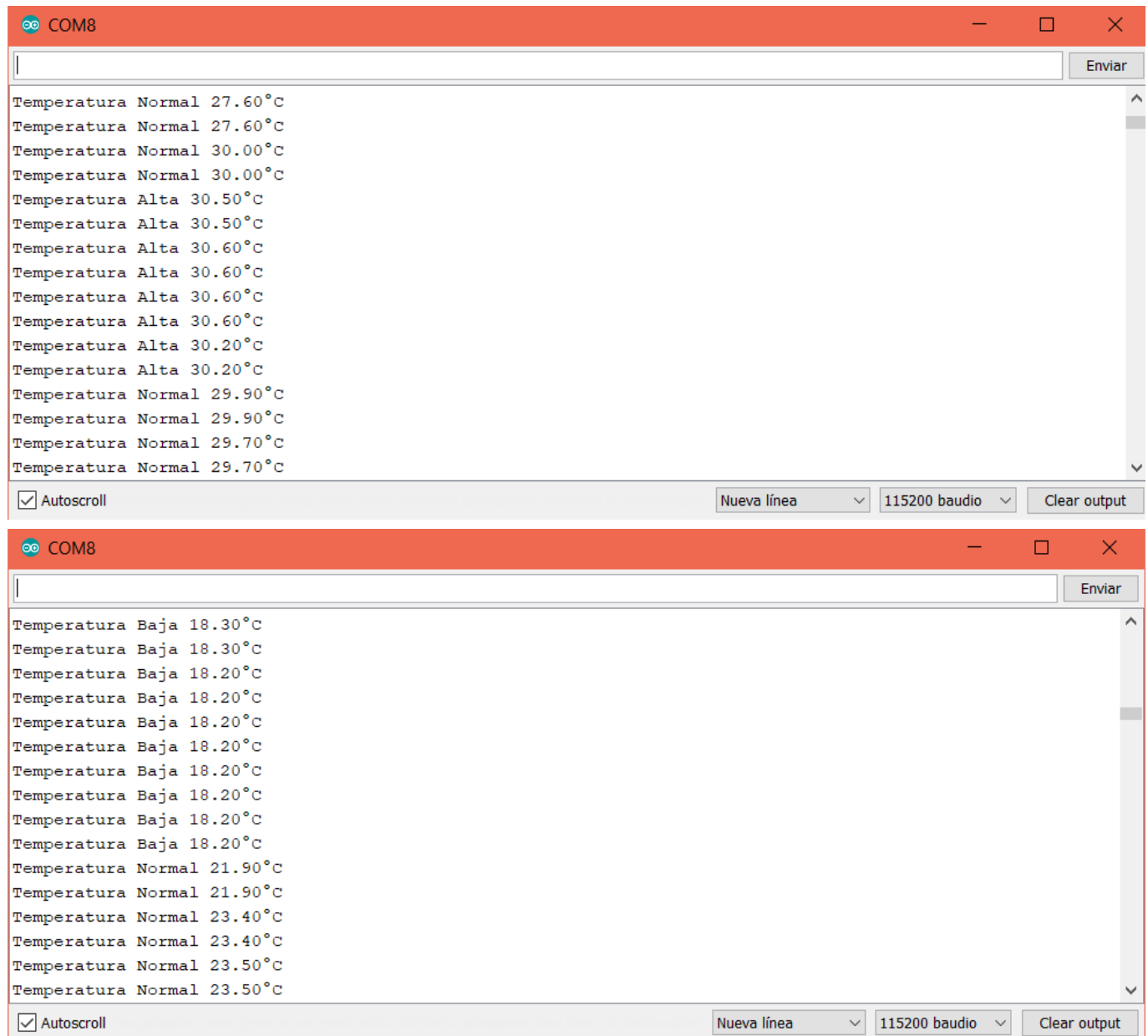
}
//Por debajo de los 20 grados
if (celsius < 20)
{
    Serial.print("Temperatura Baja ");
    ledcWrite(red, 0);
    ledcWrite(green, 0);
    ledcWrite(blue, 256);

}
//Por arriba de los 30 grados
if (celsius > 30)
{
    Serial.print("Temperatura Alta ");
    ledcWrite(red, 256);
    ledcWrite(green, 0);
    ledcWrite(blue, 0);

}
//Imprimir los grados
Serial.print(celsius);
Serial.println("°C");
}
```

---

## Monitor Serial



## Video Demostrativo

[A3.3 Circuito de medicion de temperatura a traves de un NodeMCU ESP32](#)

Slack y Meet

zerox

Añadir un tema

Hoy

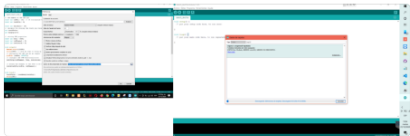
Artículo | Omniblog

Aprende a utilizar el sensor de temperatura y humedad con arduino en tus proyectos

ALEJANDRO DIAZ NAVA

13:54

image.png



ALEJANDRO DIAZ NAVA

15:29

[http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/TutorialDHT11\\_Nodemcu.pdf](http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/TutorialDHT11_Nodemcu.pdf)

VANESSA MARLENNE RODRIGUEZ BAEZ

16:47

9 archivos

30.00°C	30°C	27.60°C
30.00°C	30°C	27.60°C
.70°C	20°C	30.00°C
.70°C	20°C	30.00°C
.50°C	20°C	.50°C
.50°C	20°C	.50°C
.20°C	20°C	.60°C
.20°C	20°C	.60°C
.80°C	20°C	.60°C
.50°C	20°C	.60°C
.50°C	11.90°C	.20°C
.10°C	13.40°C	29.90°C
.10°C	13.40°C	29.90°C
29.80°C	13.50°C	29.70°C
29.80°C	13.50°C	29.70°C

Enviar mensaje a zerox

B I S </> @ ☺ 📎

Aa @ ☺ 📎 ➡

Meet - opg-mpyf-mbu

meet.google.com/opg-mpyf-mbu

AliExpress Facebook Booking.com Google ¡Bienvenido a Face... YouTube - Broadcas... Juegos PC Descarga... Cinemex Materias GitHub - wigar/topi...

**VANESSA MARLENNE RODRIGUEZ BAEZ** está presentando

```
#37
#include "Pwm.h" //libreria

//pines que vamos a usar
const int led1 = 16;
const int led2 = 17;
const int led3 = 5;

//Establecemos los pines PWM
const int freq = 3000;
const int resolution = 8;

//Canales
const int red = 1;
const int green = 2;
const int blue = 3;

//Declaramos el pin del sensor
#define DETECTOR 4
//Declaramos el sensor que se utilizara
#define DETECTOR GPIO1
//Pin del DETECTOR, DETECTOR.

void setup() {
  Serial.begin(115200);
  //Inicializamos la libreria pwm
  pwm.begin();
  //Se configura la señal PWM
  ledSetup(red, freq, resolution);
  ledSetup(green, freq, resolution);
  ledSetup(blue, freq, resolution);
  //Se le asigna la señal al GPIO que estemos utilizando
  ledAttachPin(led1, red);
  ledAttachPin(led2, green);
  ledAttachPin(led3, blue);
}

void loop() {
  //Se enciende el LED rojo
  digitalWrite(led1, HIGH);
  //Se enciende el LED verde
  digitalWrite(led2, HIGH);
  //Se enciende el LED azul
  digitalWrite(led3, HIGH);
  //Se enciende el sensor
  digitalWrite(DETECTOR, HIGH);
  delay(1000);
  //Se apaga el LED rojo
  digitalWrite(led1, LOW);
  //Se apaga el LED verde
  digitalWrite(led2, LOW);
  //Se apaga el LED azul
  digitalWrite(led3, LOW);
  //Se apaga el sensor
  digitalWrite(DETECTOR, LOW);
  delay(1000);
}
```

05:14 p.m. 01/12/2020

- **Rodríguez Báez Vanessa Marlenne:** En esta practica se volvió a utilizar el ESP32 pero ahora con un sensor DHT11 que fue el que nosotros utilizamos y un led RGB, con el ESP32 ya sabemos como funciona pero en mi caso jamás había utilizado el sensor DHT11 ni el led RGB por lo que tuvimos que investigar mucho sobre esos 2 componentes, Una de las cosas en las que mas se nos complico fue hacer que la computadora y el IDE de Arduino detectara el ESP32 en la computadora y al igual que las librerías, después de una investigación y de instalar de nuevo el Arduino funciono, otro contratiempo fue que el led RGB me prendía todos los colores y no podía hacer que solo 1 se quedara prendido pero ya una vez solucionado funciono como indica la practica. En esta practica aprendí a utilizar otro sensor y como programarlo al igual de la utilización del led RGB.
- **Soria Márquez Guillermo:** En esta práctica utilizamos en sensor de temperatura DHT11 junto con el ESP32 para analizar el cambio de temperatura mediante un diodo RGB. Sobre el ESP32 ya teníamos un poco de noción de su funcionamiento gracias a la practicas pasadas, pero el sensor DHT11 era completamente nuevo para nosotros, al momento de ensamblar tuvimos problemas ya que el IDE del arduino de la computadora en la que se realizo no detectaba ya que era de una versión diferente, igual tuvimos problema con la conexión del led. Tuvimos que investigar mucho ya que desconocíamos varios componentes con los que se trabajaron, pero al final se obtuvo el resultado esperado.

## Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10

 [Link Díaz Navarro Alejandro](#)

 [Link Rodríguez Báez Vanessa Marlenne](#)

 [Link Soria Márquez Guillermo](#)