



A.3.3 Actividad de aprendizaje

Círculo 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 extensión .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuenta 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:

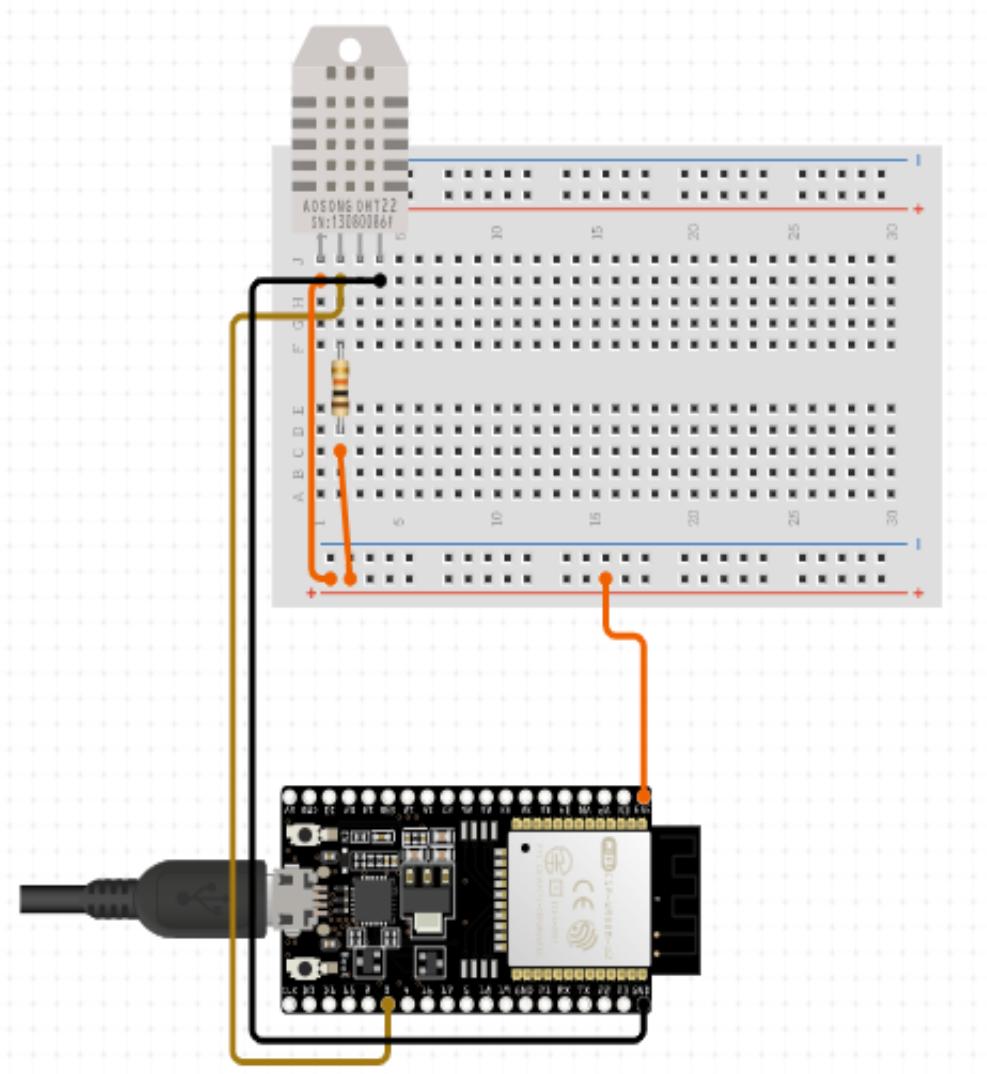
- Random Nerd Tutorial Touch pin
 - Ejemplo de circuito con LED RGB Led
-

Desarrollo

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

Cantidad	Descripción
1	Sensor temperatura y humedad DHT11 o DHT22
1	Diodo led RGB
1	Resistencia 4.7 kohms
3	Resistencias 1 kohm
1	Fuente de voltaje de 5V
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M

2. 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 registrar 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.

- Evidencia física realizada por Jose Alfredo Venegas Medina
- [Presentacion del circuito \(VIDEO\)](#)

Archivo Editar Programa Herramientas Ayuda



DHT11 §

```
//Pin de LEDS
const int LEDR = 17;
const int LEDG = 18;
const int LEDB = 19;

//Canales de LEDS
const int Canal_R = 0;
const int Canal_G = 1;
const int Canal_B = 2;

const int frecuencia = 5000;

const int resolucion = 8; //8 bit

// Biblioteca Sensor de Temperatura
#include <Adafruit_Sensor.h>
#include "DHT.h"

//Pin del sensor DHT
const int DHTPIN = 16;

//Define el modelo del DHT
#define DHTTYPE DHT11

//Variables para el DHT
DHT dht(DHTPIN, DHTTYPE);
float humedad;
float humedad_actual;
float temperatura;
float temperatura_actual;

void setup() {
    Serial.begin(115200);
    dht.begin(); //Metodo de libreria

    ////Se declaran los Pin para ser controlados
    ledcAttachPin(LEDR, Canal_R);
    ledcAttachPin(LEDG, Canal_G);

    ////Se declaran los Pin para ser controlados
    ledcAttachPin(LEDR, Canal_R);
    ledcAttachPin(LEDG, Canal_G);
    ledcAttachPin(LEDB, Canal_B);

    //Se configura el comportamiento de los LEDs a una frecuencia de 5000
    ledcSetup(Canal_R, frecuencia, resolucion);
    ledcSetup(Canal_G, frecuencia, resolucion);
```

```
ledcSetup(Canal_A, frecuencia, resolucion),  
ledcSetup(Canal_B, frecuencia, resolucion);  
}  
  
void loop() {  
  
    //Lectura del DHT  
    humedad = dht.readHumidity();  
    temperatura = dht.readTemperature();  
  
    //Si no se detecta ningun valor, mandara mensaje  
    if (isnan(humedad) || isnan(temperatura)) {  
        Serial.println("No se ha detectado DHT");  
        delay(1000);  
    }  
  
    //Si existe un cambio de temperatura o humedad, imprimira los nuevos valores  
    else if (humedad_actual != humedad || temperatura_actual != temperatura) {  
        humedad_actual = humedad;  temperatura_actual = temperatura;  
  
        //Imprime mensaje segun la temperatura  
        if(temperatura >= 45){  
            Serial.print("Temperatura alta: ");  
            Serial.print(temperatura);  
            Serial.print("°");  
            Serial.println("C");  
        }  
  
        if(temperatura <= 5){  
            Serial.print("Temperatura baja: ");  
            Serial.print(temperatura);  
            Serial.print("°");  
        }  
    }  
}
```

```
}

if((temperatura > 5) && (temperatura < 45)) {
    Serial.print("Temperatura ambiente: ");
    Serial.print(temperatura);
    Serial.print("°");
    Serial.println("C");
}

Serial.print("Humedad:");
Serial.print(humedad);
Serial.println("%");
Serial.println("=====");

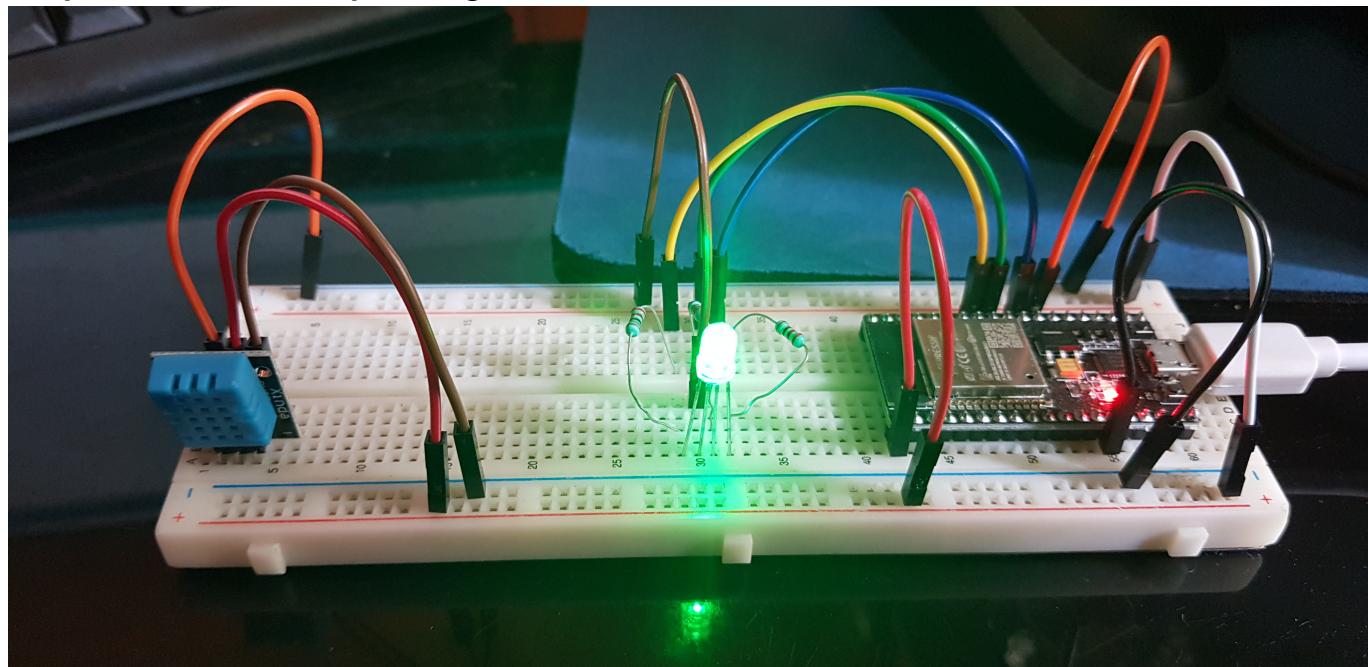
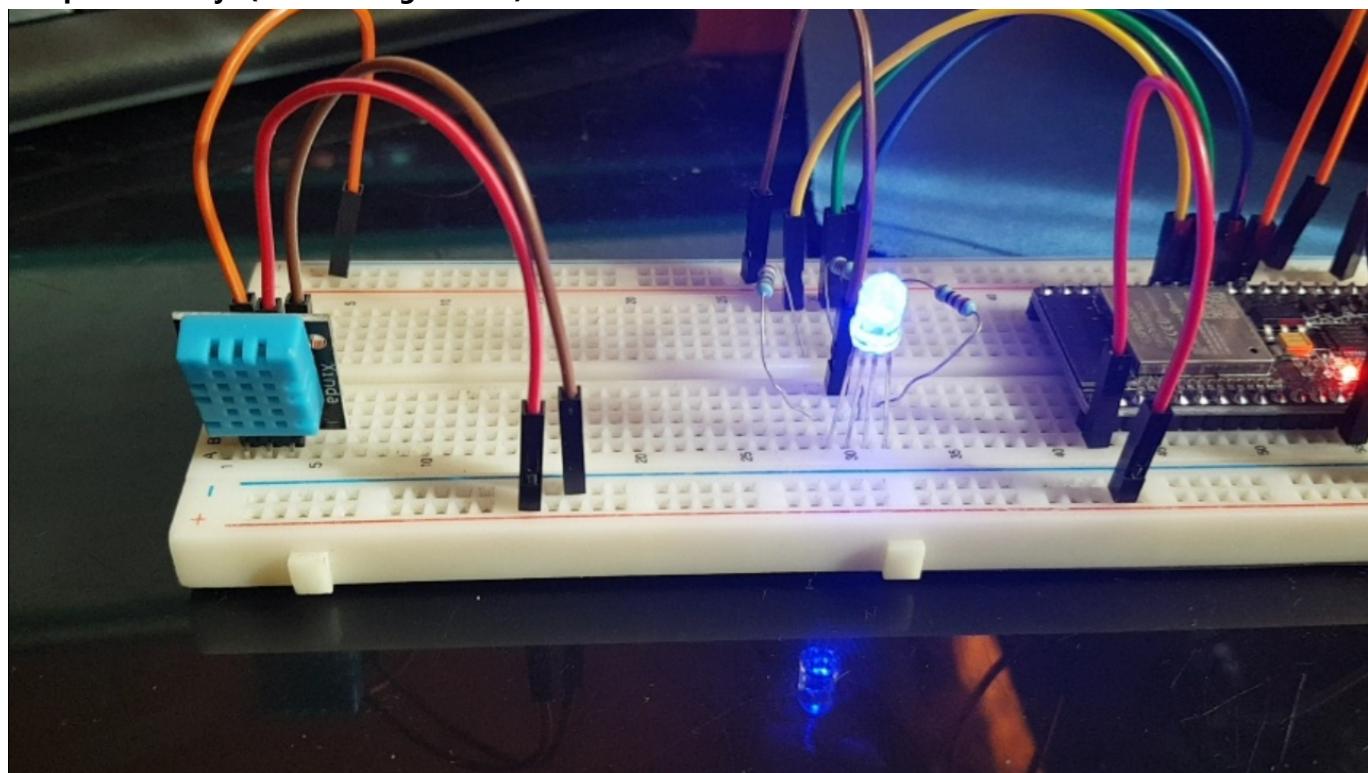
//Si la temperatura supera los 45, el LED se prendera Rojo
if(temperatura >= 45){
    RGB_Color(255,255,0); //Rojo
}

//Si la temperatura es menor a 5, el LED se prendera Azul
if(temperatura <= 5){
    RGB_Color(255,0,255); //Azul
}

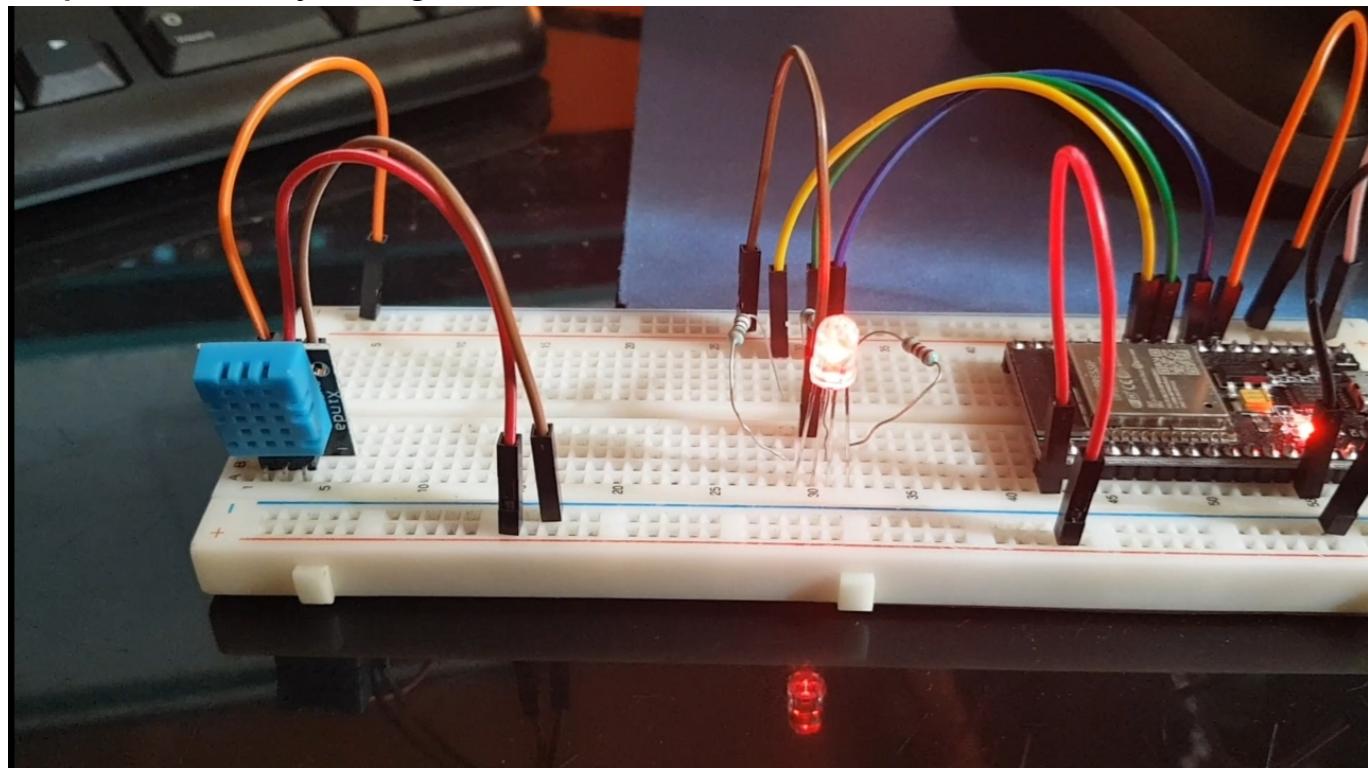
//Si la temperatura se mantiene entre 5 y 45, el LED se prendera Verde
if((temperatura > 5) && (temperatura < 45)) {
    RGB_Color(0,255,255); //Verde
}

}

//Metodo para cambiar el color del LED
void RGB_Color(int i, int j, int k)
{
    ledcWrite(Canal_R, i);
    ledcWrite(Canal_G, j);
    ledcWrite(Canal_B, k);
}
```

Temperatura Ambiente (Aprox. 25 grados C):**Temperatura Baja (Menor a 5 grados C):**

Temperatura Alta (Mayor a 45 grados C):



3. Evidencias reuniones.

Screenshot of a Google Meet video conference interface. The top bar shows tabs for Facebook, Clases, Gestión de proyectos - Google, Meet - idr-cufa-wei, and Sistemas Programables: A3.3. The main area shows three participants: Jose Venegas (presenting), CESAR MANUEL RAMIREZ CERVANTES, and Jose Venegas (another participant). The left side of the screen displays a large amount of Arduino C code for a temperature control project, including definitions for pins, constants, and functions for reading DHT sensors and controlling an LCD and LEDs. The code is organized into sections for pin definitions, sensor definitions, and various control functions.

```

//A3.3 Alumno: J.A.V.
//Autor: Jose Alfredo Venegas Medina Ayala
//Unidad: Sistemas Programables
//Materia: Microcontroladores Avanzados

//Pin
#define Pin_Temp 27
const int LEDR = 17;
const int LEDG = 18;
const int LEDB = 19;
const int LEDY = 20;

//Definiciones de DHT
const int Canal_0 = 0;
const int Canal_1 = 1;
const int Canal_2 = 2;
const int Canal_3 = 3;

const int Frecuencia = 5000;

const int retrasoMotor = 50 //0 ms

// Variables Sensor de temperatura
#include <DHT.h>
#include "DHT.h"

//Pin del sensor DHT
const int DHTPIN = 18;
//const int DHTTYPE = DHT11;
//const int DHTTYPE = DHT22;
//const int DHTTYPE = DHT11;
//const int DHTTYPE = DHT22;

//Variables para el LCD
DHT dht(DHTPIN, DHTTYPE);
level humedad;
float humedadActual;
float temperatura;
level temperaturaActual;

void setup() {
  Serial.begin(115200);
  dht.begin(); //Este es el libreria
}

//Este declaran los Pin para ser controlados
ledtachibana(LCD, Canal_R);
lcdControl(LCD, Canal_G);
ledtachibana(LCD, Canal_B);
ledtachibana(LCD, Canal_Y);

//Pulse con el teclado la configuracion de los LED's en una frecuencia de 5000
ledtachibana(Canal_R, frecuencia, resolution);
ledtachibana(Canal_G, frecuencia, resolution);

```

Jose Venegas está presentando

19:06:07.474 -> Temperatura ambiente: 21.00°C
 19:06:07.474 -> Humedad:34.00%
 19:06:07.474 -> ======
 19:06:09.492 -> Temperatura ambiente: 21.00°C
 19:06:09.492 -> Humedad:39.00%
 19:06:09.492 -> ======
 19:06:11.470 -> Temperatura ambiente: 21.00°C
 19:06:11.504 -> Humedad:43.00%
 19:06:11.504 -> ======
 19:06:13.475 -> Temperatura ambiente: 21.00°C
 19:06:13.475 -> Humedad:42.00%
 19:06:13.475 -> ======
 19:06:15.490 -> Temperatura ambiente: 21.00°C
 19:06:15.490 -> Humedad:41.00%
 19:06:15.490 -> ======
 19:06:17.499 -> Temperatura ambiente: 21.00°C
 19:06:17.499 -> Humedad:40.00%
 19:06:17.499 -> ======
 19:06:19.480 -> Temperatura ambiente: 21.00°C
 19:06:19.480 -> Humedad:39.00%
 19:06:19.480 -> ======
 19:06:21.491 -> Temperatura ambiente: 21.00°C
 19:06:21.491 -> Humedad:38.00%
 19:06:21.491 -> ======
 19:06:23.477 -> Temperatura ambiente: 21.00°C
 19:06:23.477 -> Humedad:37.00%
 19:06:23.477 -> ======
 19:06:29.474 -> Temperatura ambiente: 21.00°C
 19:06:29.474 -> Humedad:36.00%
 19:06:29.474 -> ======

Autoscroll Mostrar marca temporal Ambos NL y CR 115200 baudio Limpiar salida

Type here to search

Jose Venegas está presentando

19:09:00.000 -> [1] JoseAlfredoVenegasMedina_Verde.md : docx 3 / A3.3_JoseAlfredoVenegasMedina_Verde.md : troquel A3.3 Actividad de aprendizaje ...

1. Basado en las imágenes que se muestran en la **Figura 1**, ensamble el circuito en la sala de circuitos electrónicos, de tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones antes solicitadas para esta actividad.

```
align="center">


```

2. Una vez ensamblado el circuito anterior, a través de un LCD RGB y elabora el programa que le permite al LCD RGB funcionar como indicador para las siguientes temperaturas:

- El sensor de Temperatura en todo momento estará enviando, mediante el valor registrado por la terminal serial, por ejemplo "Temperatura: 25 grados C" y el LED RGB estará encendido de color verde.
- El sensor de Temperatura en todo momento, en caso de que el valor de la temperatura ambiente, deberá mostrar el mensaje "Temperatura alta: > 30 grados C".
- El sensor de Temperatura en todo momento, en caso de que el valor del <30 por debajo de la temperatura ambiente, deberá mostrar el mensaje "Temperatura baja: < 20 grados C".

3. Coloca aquí evidencias que consideras importantes durante el desarrollo de la actividad.

- Evidencia 1: Imagen del Circuito (Vídeo 1)(https://youtu.be/luhRmKgG8b)
- Código 1: ./Img/A3.3 Código1.png
- Código 2: ./Img/A3.3 Código2.png
- Código 3: ./Img/A3.3 Código3.png

Temperatura Ambiente (Aprox. 25 grados C):

!Evidence 1: ./Img/A3.3 Evidence1.jpg

Temperatura Baja (Menor a 5 grados C):

!Evidence 2: ./Img/A3.3 Evidence2.jpg

Temperatura Alta (Mayor a 45 grados C):

!Evidence 3: ./Img/A3.3 Evidence3.jpg

Type here to search

Jose Venegas está presentando

19:09:00.000 -> [1] JoseAlfredoVenegasMedina_Verde.md : docx 3 / A3.3_JoseAlfredoVenegasMedina_Verde.md : troquel A3.3 Actividad de aprendizaje ...

1. Basado en las imágenes que se muestran en la **Figura 1**, ensamble el circuito en la sala de circuitos electrónicos, de tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones antes solicitadas para esta actividad.

```
align="center">


```

2. Una vez ensamblado el circuito anterior, a través de un LCD RGB y elabora el programa que le permite al LCD RGB funcionar como indicador para las siguientes temperaturas:

- El sensor de Temperatura en todo momento estará enviando, mediante el valor registrado por la terminal serial, por ejemplo "Temperatura: 25 grados C" y el LED RGB estará encendido de color verde.
- El sensor de Temperatura en todo momento, en caso de que el valor de la temperatura ambiente, deberá mostrar el mensaje "Temperatura alta: > 30 grados C".
- El sensor de Temperatura en todo momento, en caso de que el valor del <30 por debajo de la temperatura ambiente, deberá mostrar el mensaje "Temperatura baja: < 20 grados C".

3. Coloca aquí evidencias que consideras importantes durante el desarrollo de la actividad.

- Evidencia 1: Imagen del Circuito (Vídeo 1)(https://youtu.be/luhRmKgG8b)
- Código 1: ./Img/A3.3 Código1.png
- Código 2: ./Img/A3.3 Código2.png
- Código 3: ./Img/A3.3 Código3.png

Temperatura Ambiente (Aprox. 25 grados C):

!Evidence 1: ./Img/A3.3 Evidence1.jpg

Temperatura Baja (Menor a 5 grados C):

!Evidence 2: ./Img/A3.3 Evidence2.jpg

Temperatura Alta (Mayor a 45 grados C):

!Evidence 3: ./Img/A3.3 Evidence3.jpg

Type here to search

Jose Venegas está presentando

19:09:00.000 -> [1] JoseAlfredoVenegasMedina_Verde.md : docx 3 / A3.3_JoseAlfredoVenegasMedina_Verde.md : troquel A3.3 Actividad de aprendizaje ...

1. Basado en las imágenes que se muestran en la **Figura 1**, ensamble el circuito en la sala de circuitos electrónicos, de tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones antes solicitadas para esta actividad.

```
align="center">


```

2. Una vez ensamblado el circuito anterior, a través de un LCD RGB y elabora el programa que le permite al LCD RGB funcionar como indicador para las siguientes temperaturas:

- El sensor de Temperatura en todo momento estará enviando, mediante el valor registrado por la terminal serial, por ejemplo "Temperatura: 25 grados C" y el LED RGB estará encendido de color verde.
- El sensor de Temperatura en todo momento, en caso de que el valor de la temperatura ambiente, deberá mostrar el mensaje "Temperatura alta: > 30 grados C".
- El sensor de Temperatura en todo momento, en caso de que el valor del <30 por debajo de la temperatura ambiente, deberá mostrar el mensaje "Temperatura baja: < 20 grados C".

3. Coloca aquí evidencias que consideras importantes durante el desarrollo de la actividad.

- Evidencia 1: Imagen del Circuito (Vídeo 1)(https://youtu.be/luhRmKgG8b)
- Código 1: ./Img/A3.3 Código1.png
- Código 2: ./Img/A3.3 Código2.png
- Código 3: ./Img/A3.3 Código3.png

Temperatura Ambiente (Aprox. 25 grados C):

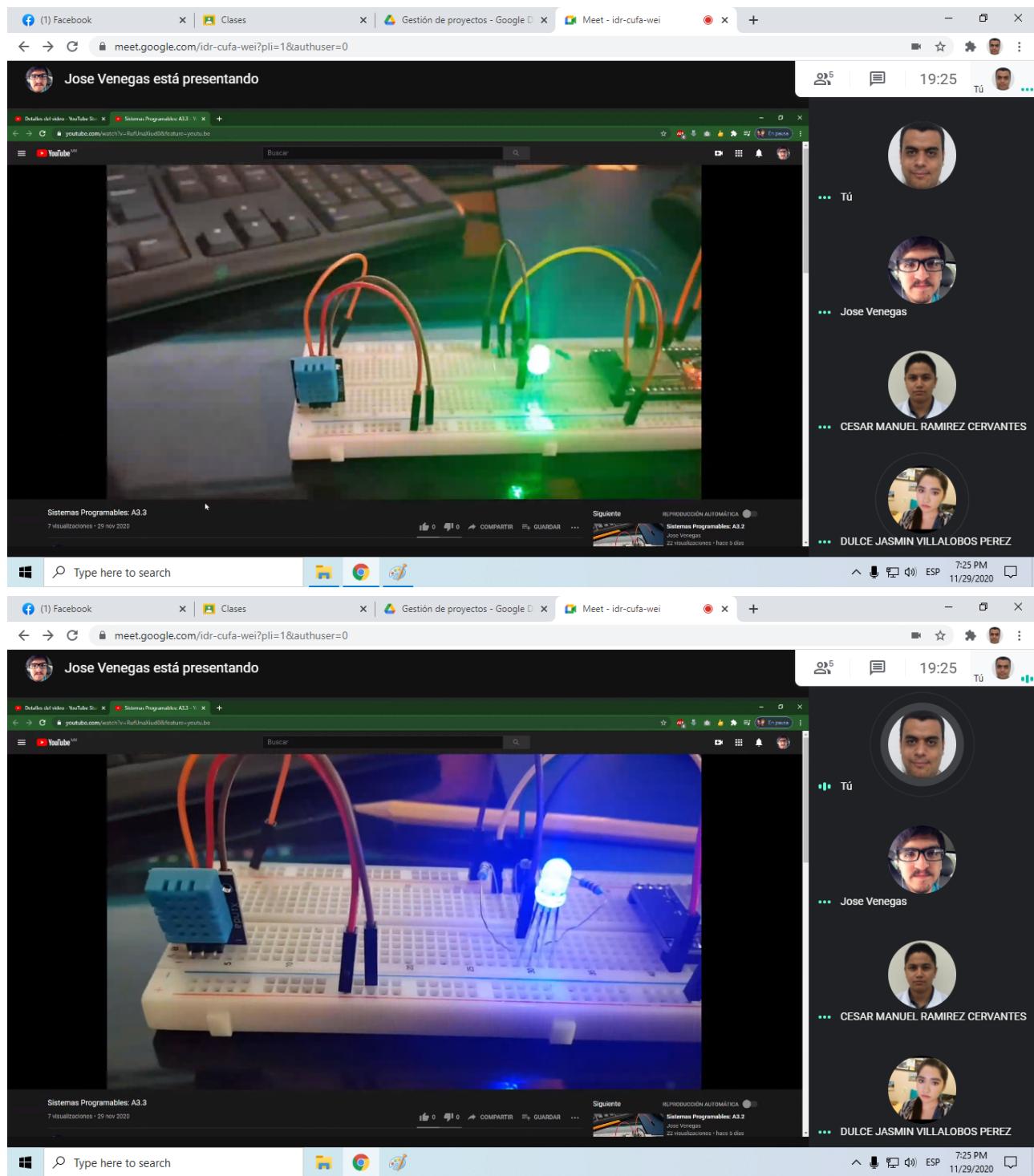
!Evidence 1: ./Img/A3.3 Evidence1.jpg

Temperatura Baja (Menor a 5 grados C):

!Evidence 2: ./Img/A3.3 Evidence2.jpg

Temperatura Alta (Mayor a 45 grados C):

!Evidence 3: ./Img/A3.3 Evidence3.jpg



4. Conclusiones:

- Acevedo Ensiso Pedro Gabriel:**

La utilización de un sensor de temperatura es muy útil para diversas aplicaciones en la industria, con esta práctica compruebo que lo que hay detrás de un sistema detector de temperatura más que nada es la escalación apropiada de los valores medidos a valores que sean concretos para nuestros objetivos, mediante el uso de código podemos traducir las mediciones a temperaturas con las cuales al ser detectadas nos indiquen una función a realizar, en este caso fue el simple uso de un LED RGB con el cual mostramos que temperatura se a detectado. Pero un sistema de este tipo fácilmente puede ser utilizado en sistemas de enfriamiento, control de temperatura e incluso en componentes digitales.

- Ramirez Cervantes Cesar Manuel:**

Gracias a los circuitos que nos ayudan a detectar la temperatura podemos hacer muchas aplicaciones

en el área de automatización con esto podemos controlar la temperatura de un refrigerador, también podemos usar un circuito como este en sistemas de aviso de emergencia dentro de lugares peligrosos, como las plantas de energía eléctrica o nuclear. Dentro de esta práctica realizamos el simple acto de encender un LED rgb de tres colores distintos verde, azul y rojo para esta practica verde determina la temperatura ambiente, rojo determina una temperatura alta y azul un temperatura baja, aunque la aplicación es sencilla en base a esto podemos utilizar la salida de este sistema para realizar alguna de las diferentes aplicaciones que hay en el mercado.

- **Venegas Medina Jose Alfredo:**

En esta práctica pudimos observar la utilización de un sensor DHT11 con un ESP32, para esto utilizamos un LED RGB que cambiaría de color según el cambio de temperatura. Se utilizó una librería DHT para poder monitorear los estados del sensor, temperatura y humedad. Para una mejor visualización de los cambios de estado se implementó un LED RGB que estuviera encendido en una luz verde a temperatura ambiente, si el sensor subía demasiado su temperatura, entonces el LED RGB cambiaría a rojo, si el sensor bajaba de temperatura, entonces el LED RGB cambiaría a azul. Al principio, existió un problema con el LED RGB, ya que se desconocía si era de Ánode común o Cátodo común, pero luego de probarlo se concluyó que era de Ánode común el cual debía estar conectado al positivo. De igual manera, en esta práctica se decidió utilizar solamente 3.3 V para evitar quemar los componentes con más voltaje. Luego el LED RGB no cambiaba al color solicitado, ya que, en vez de ser rojo, verde y azul, el LED RGB iluminaba rosa, amarillo y azul claro, luego de revisarlo, se concluyó que al ser de Ánode común los valores RGB eran distintos. Por último, para variar aumentar la temperatura del sensor se decidió utilizar una secadora de pelo, y para disminuir su temperatura se decidió meter el sensor en una bolsa de plástico 5 minutos en el congelador.

- **Villalobos Perez Dulce Jasmin:**

Como sabemos para la realización de estas actividades nuestro compañero a cargo de los materiales es él qué arma el circuito, entonces mientras observaba como se iban conectando, qué pines del ESP32 se usaban, como sabemos usamos un sensor de temperatura DHT, fue interesante realizar esta práctica ya qué el usar el sensor y hacerlo qué dependiendo de los grados de temperatura a los que se expuso se puedan representar a través del color de un LED RGB, la verdad aprendí como se usa, qué existe una secuencia para encenderlo de un color específico, es la primera vez qué veo uno y también qué veo como funciona dentro del código, sobre el código se declaró los pines Led qué son el número del pin al cual están vinculados y sus respectivos canales, para hacer uso del sensor se debe hacer uso de las librerías del sensor y definir de qué tipo es el modelo del DHT con el que se va a trabajar, como sabemos en las indicaciones decía qué dependiendo de qué temperatura detectara el sensor se imprimiría un mensaje y a la vez el led Rgb tomaría el color indicado, por ejemplo si la temperatura registrada era de 21 grados, entonces el Led se pondría de color verde, y se imprime: temperatura ambiente: 21 grados, para eso en el código se especificaron 3 condicionales para las 3 posibles condiciones como lo es temperatura baja, ambiente y alta, las condiciones fueron si era menor a 5 grados es baja, si está dentro del rango de mayor de 5 a 45 entonces ambiente, si excede los 45 grados es alta, y eso mismo para el Led Rgb de esas condiciones dependía su color.



Rubrica

Criterios	Descripción	Puntaje
-----------	-------------	---------

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

EQUIPO VERDE

Acevedo Ensiso Pedro Gabriel:

 [Ir a mi Github](#)

Ramirez Cervantes Cesar Manuel:

 [Enlace a mi repositorio](#) 

Venegas Medina Jose Alfredo:

 [Mi Github](#)

Villalobos Perez Dulce Jasmin:

  [ENLACE - MI GITHUB](#)