

Sesión 4

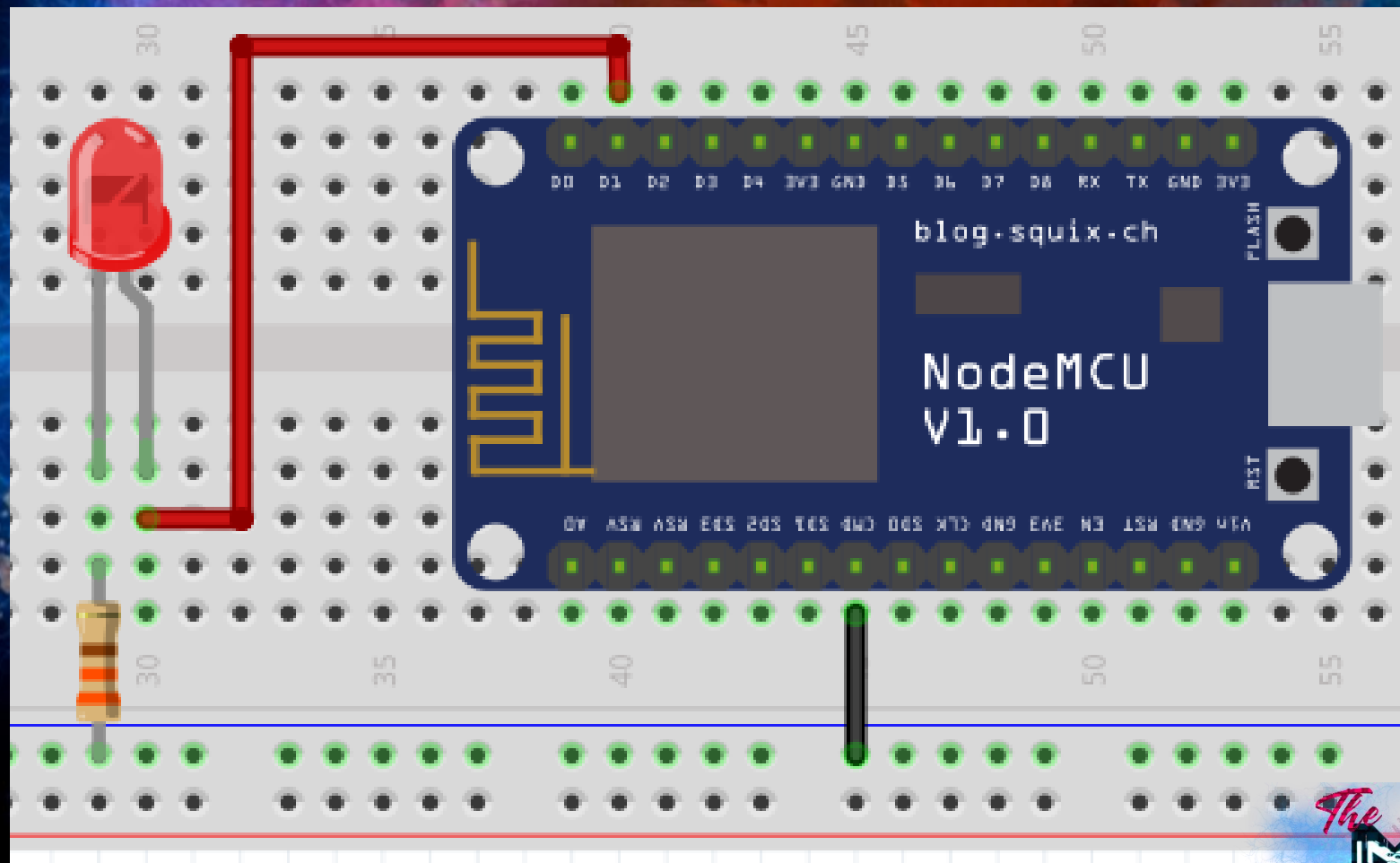
Blynk



PINES PWM

- Pines PWM o pines con modulación de pulso (Pulse Width Modulation).
- En realidad son pines digitales que imitan salidas analógicas modificando la separación entre los diferentes pulsos de la señal.
- La señal PWM puede dar diversos valores hasta 255 incluyendo el 0, se utilizan por ejemplo para variar la intensidad de un led o hacer funcionar un Servomotor.
- Hay que hacer notar que estos pines funcionan como salidas o entradas digitales o como salidas analógicas.

CONEXIÓN DE UN LED CON EL NODE MCU



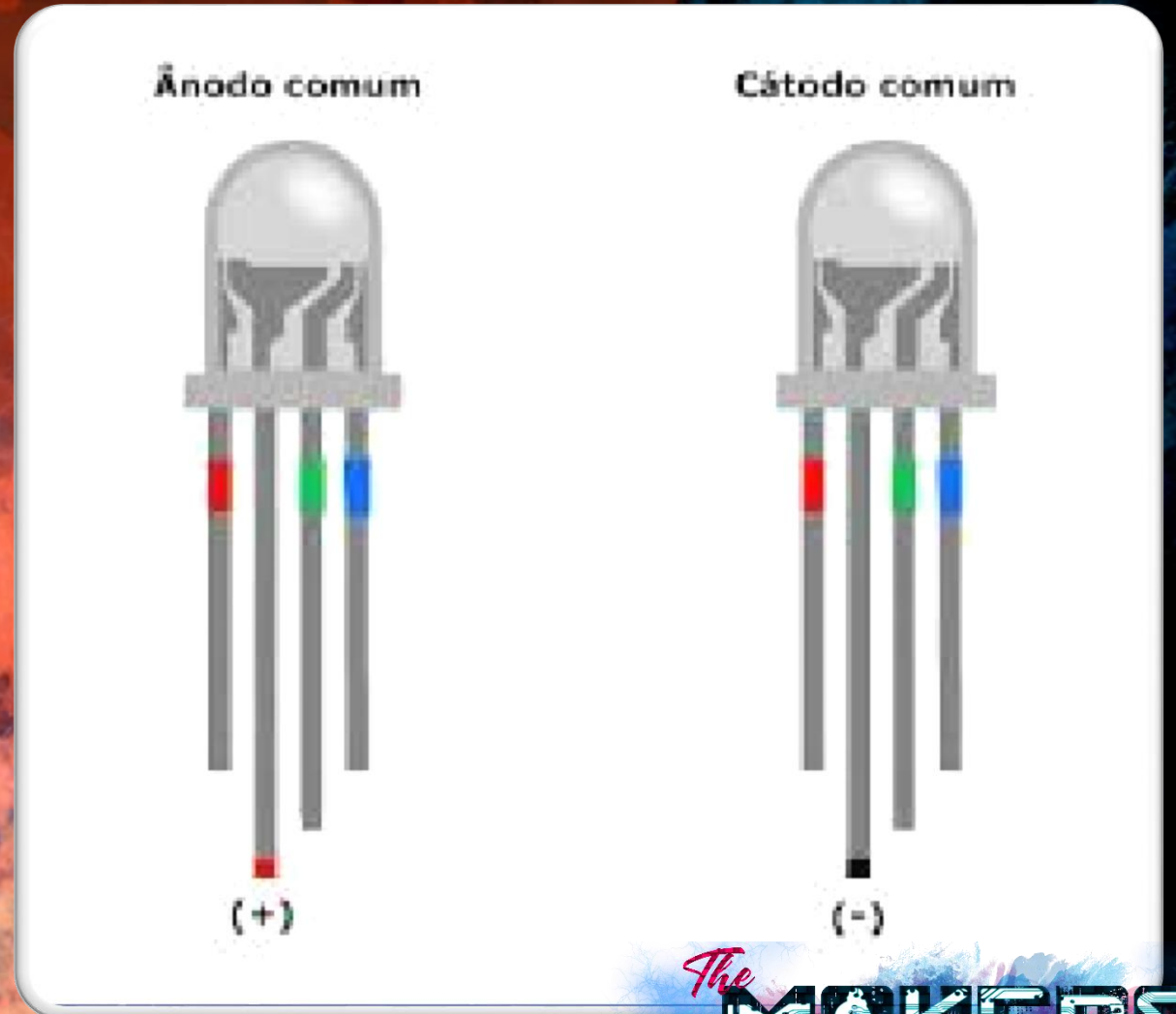
CONOCIENDO EL LED 'RGB'

Un LED RGB es en realidad la unión de tres LED's de los colores básicos, en un encapsulado común, compartiendo el Ground.

Red : Rojo

Green: Verde

Blue: Azul



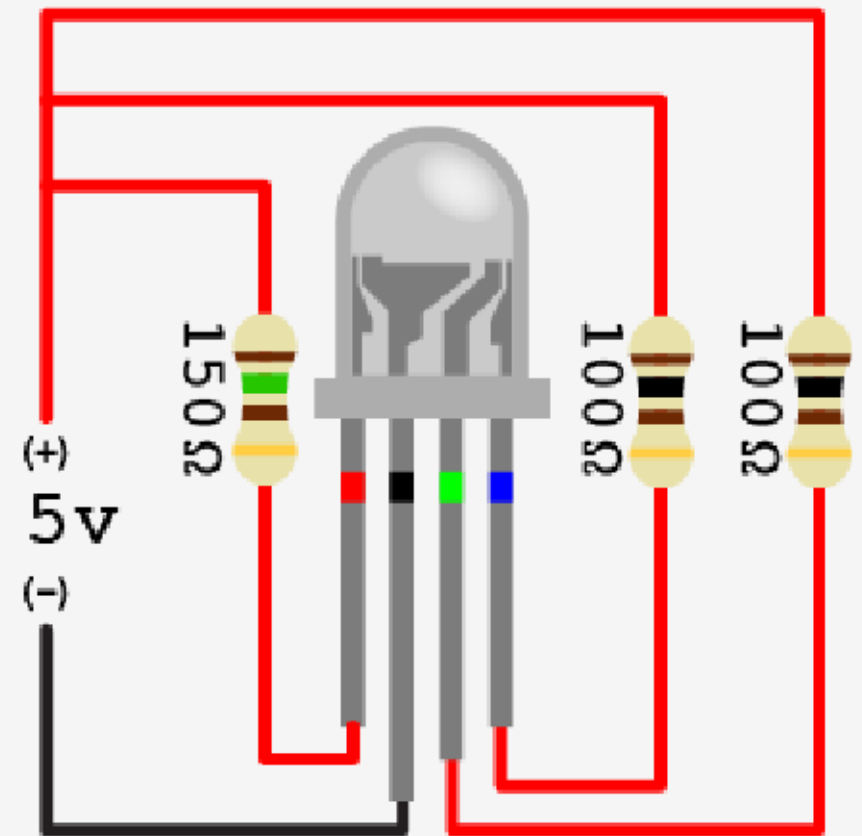
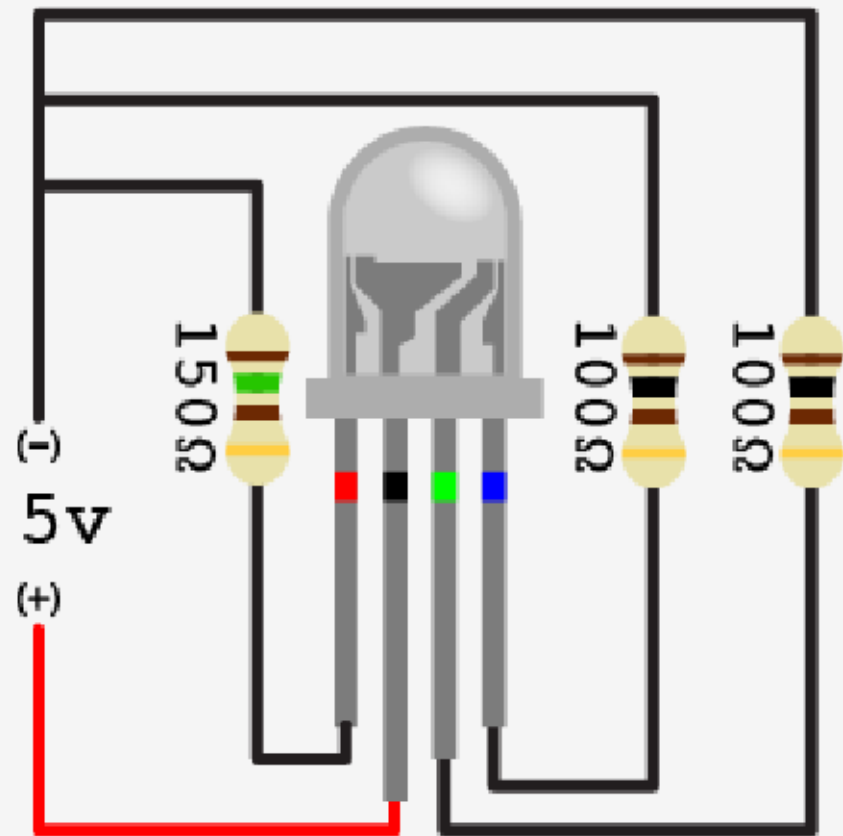
COLORES SECUNDARIOS

Mediante la combinación de los colores primarios **RGB** se pueden generar colores secundarios como el rosado, amarillo y magenta.

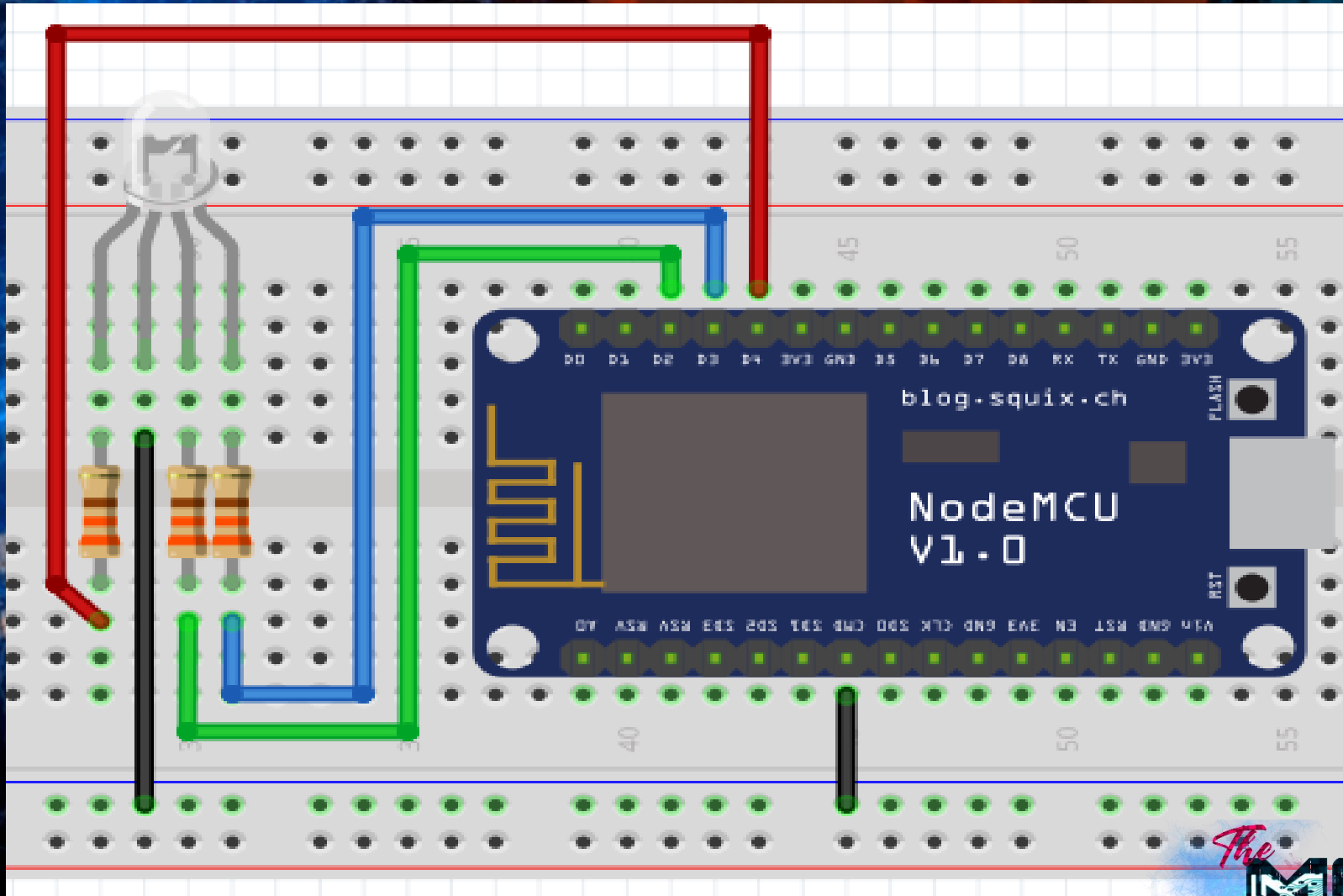
Mediante diferentes métodos se pueden obtener infinidad de colores.



COLORES SECUNDARIOS

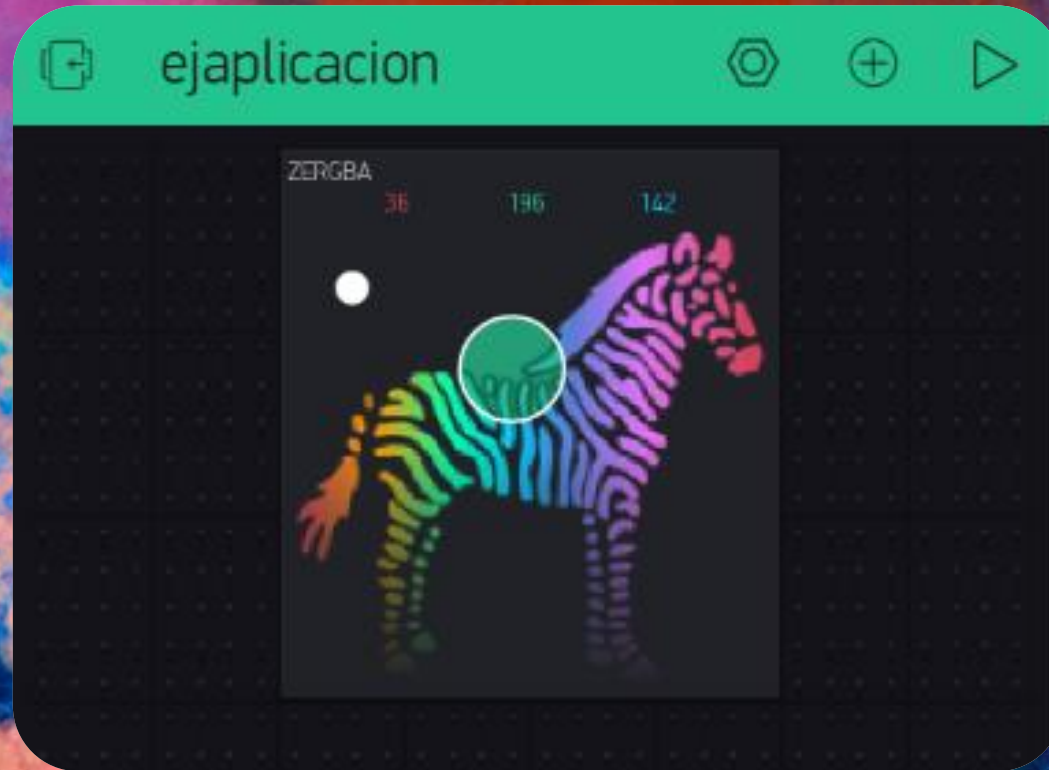


CONEXIÓN DE UN RGB CON EL NODE MCU



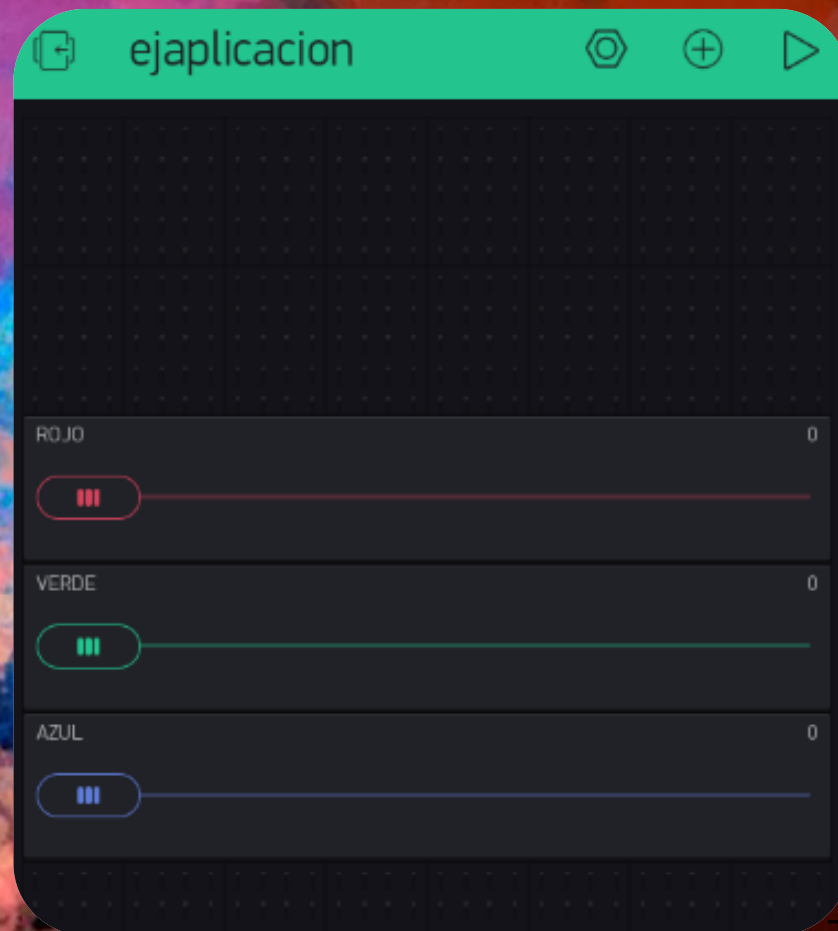
PINES PWM

Encender un led RGB utilizando BLYNK

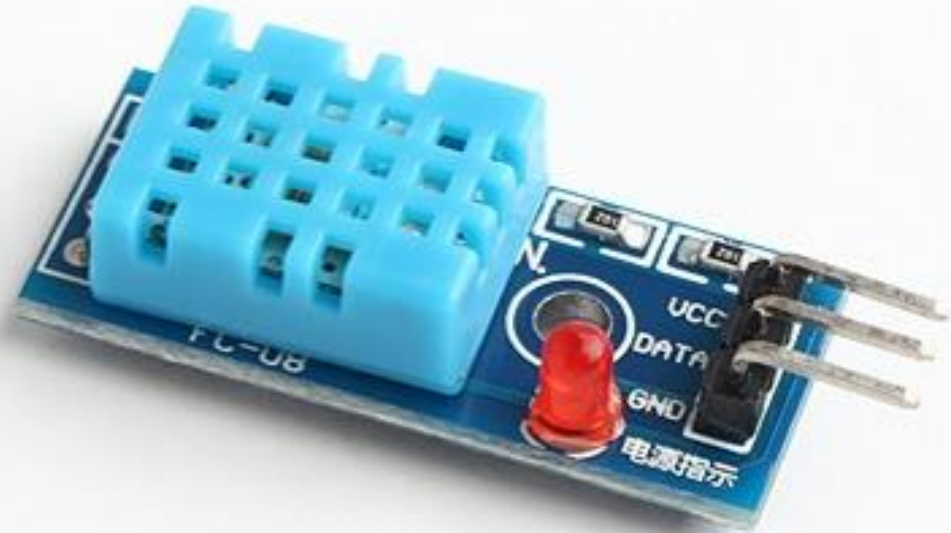


Ejercicio de aplicación

Encender un led RGB utilizando los sliders de blynk (debe enviar la captura de su código .ino y de la aplicación)



SENSOR DHT 11



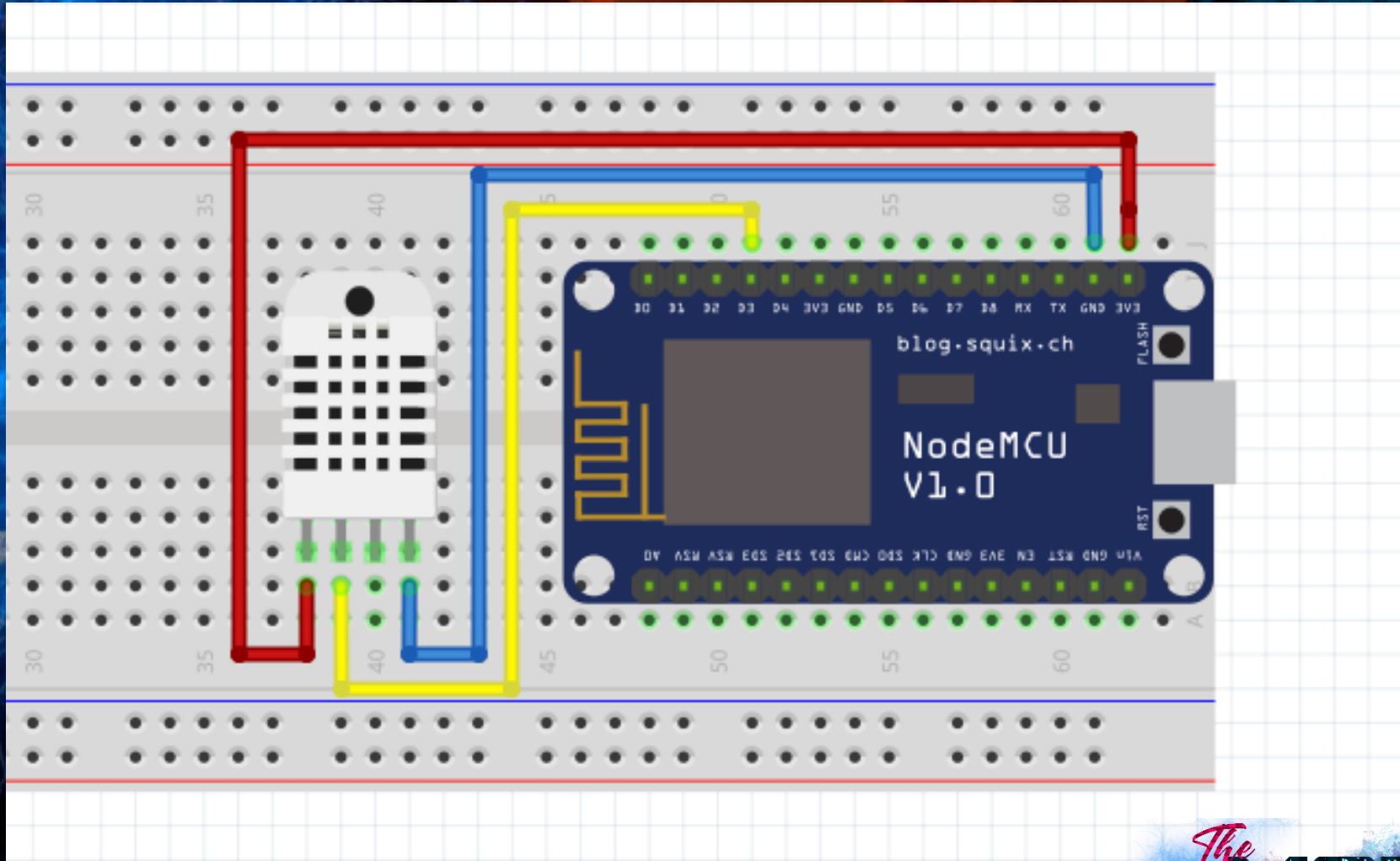
- Voltaje de Operación: 3 – 5V.
- Modo de salida: Digital.
- Rango de medición de temperatura: 0 a 50 °C con $\pm 2^{\circ}\text{C}$ de precisión.
- Rango de medición de humedad: 20% a 80% Rh con $\pm 4\%$ Rh de precisión.
- Tiempo de medición: 1 seg.
- Devuelve la medida en °C sin decimales.

SENSOR DHT 22



- Voltaje de Operación: 3.3 – 6V.
- Modo de salida: Digital.
- Rango de medición de temperatura: -40 a 80 °C con $\pm 5^{\circ}\text{C}$ de precisión.
- Rango de medición de humedad: 0% a 100% Rh con $\pm 2\%$ Rh de precisión.
- Tiempo de medición: 2 seg.
- Devuelve la medida en °C.

CONEXIÓN SENSOR DHT Y NODE MCU



Pines Virtuales en Blynk

Virtuales

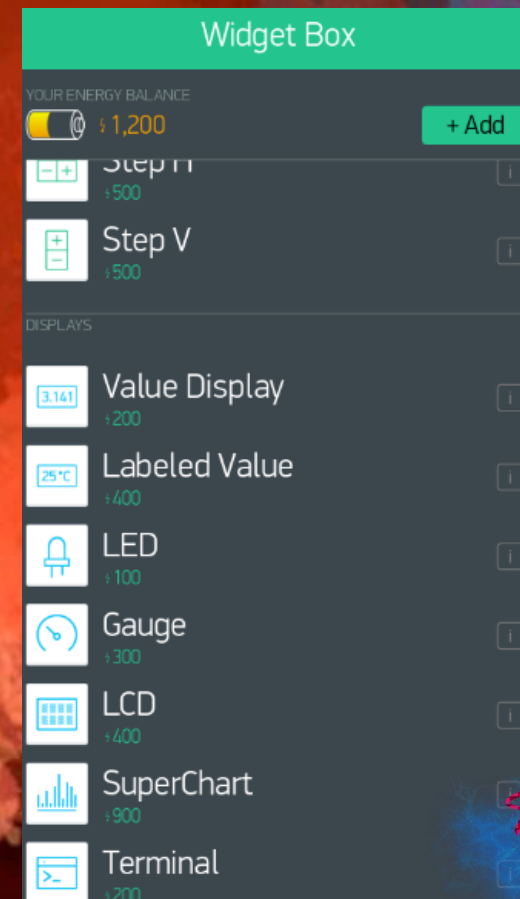
son los pines con los que blynk nos permite programar nuestra placa o cambiar valores dentro la app.

```
BLYNK_WRITE(V1) {  
  //PROGRAMA DE CONTROL  
}
```

```
BLYNK_WRITE(V1)  
{  
  String action = param.asStr();  
  
  Blynk.setProperty(V1, "label", action);  
  Serial.print(action);  
  Serial.println();  
}
```

Ejercicio de aplicación

utilizando los pines virtuales de blynk mostrar la temperatura y el tiempo en la app.
(debe enviar la captura de su código .ino y de la aplicación)



Ejercicio de aplicación

```
#include <SimpleTimer.h>    //libreria que nos permite crear intervalos de tiempo sin pausar el sistema
#include <Adafruit_Sensor.h> //libreria que nos permite tener una mejor toma de datos e interacción con nuestro de DHT
#include <DHT.h>             //libreria especifica para el manejo de sensores de tipo DHT
#include <DHT_U.h>           //extensión de la libreria que nos permite trabajar de mejor manera con nuestro DHT

#define BLYNK_PRINT Serial //Variable de tipo SERIAL especifica para conexion con Blynk

#include <ESP8266WiFi.h>     //libreria especifica para el manejo del ESP8266
#include <BlynkSimpleEsp8266.h> //libreria que nos permite interactuar entre blynk y ESP8266

#define DHTPIN D4           //pin donde se alojara la toma de datos de nuestro DHT
#define DHTTYPE DHT11       //definimos el tipo de sensor DHT que utilizaremos en esta ocasion es un DHT22
                             //para su caso usar DHT11
DHT_Unified dht(DHTPIN, DHTTYPE); //se crea por defecto el objeto de tipo DHT donde se envia el pin de toma de
                             //datos y el tipo de sensor DHT respectivamente
SimpleTimer timer;          //creamos un objeto de tipo SimpleTimer

char auth[] = "4fcc0685876846e387555c71807cdca2"; //autenticación que se envia al correo
char ssid[] = "Makers"; //Identificador de nuestra red WiFi (nombre)
char pass[] = "M@kersInformatica"; //Contraseña de nuestra red WiFi
```

Ejercicio de aplicación

```
WidgetLED led1(V1);

void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(5000L, sendUptime);
}

void loop() {
  Blynk.run();
  timer.run();
}
```

```
void sendUptime()
{
  Blynk.virtualWrite(V5, millis() / 1000);
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (!isnan(event.temperature)) {
    Blynk.virtualWrite(V6, event.temperature);
  }
  dht.humidity().getEvent(&event);
  if (!isnan(event.relative_humidity)) {
    Serial.println(event.relative_humidity);
    if(event.relative_humidity>=20){
      led1.on();
    }
    else{
      led1.off();
    }
    Blynk.virtualWrite(V7, event.relative_humidity);
  }
}
```


Ejercicio de aplicación

Utilizando los pines virtuales y un botón (de tipo push) de blynk realice la secuencia del semáforo; el led rojo debe encenderse a razón de 600 milisegundos, el amarillo a razón de 400 milisegundos y el verde a razón de un 500 milisegundos. (debe enviar la captura de su código .ino y de la aplicación)

