

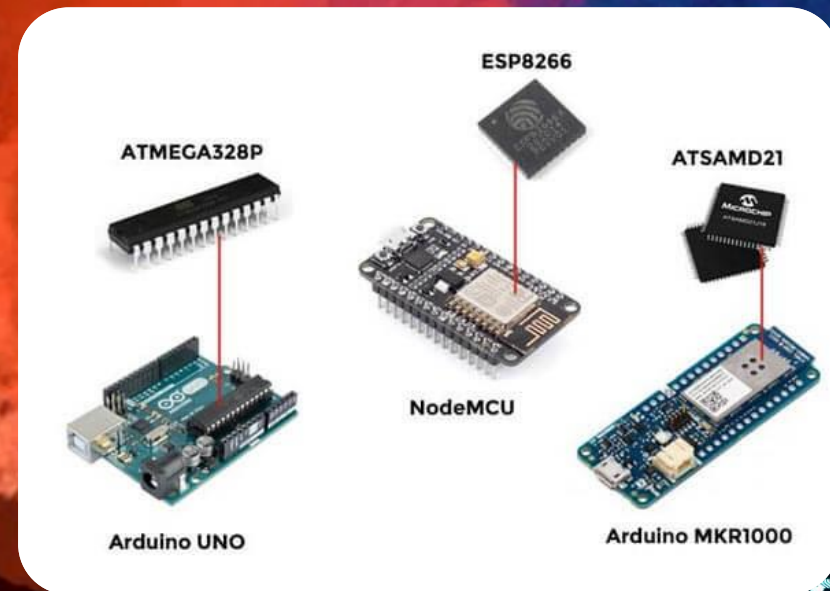


NodeMCU

The **MAKERS**
INFORMATICA

CARACTERÍSTICAS

- NodeMCU es la placa de desarrollo basada en el ESP8266, utiliza el chip ESP-12E.
- Incorpora una MCU de 32-bit de bajo consumo.
- Módulo WiFi de 2.4 GHz.
- 1 entrada analógica.
- 13 pines de entrada y salida GPIO.



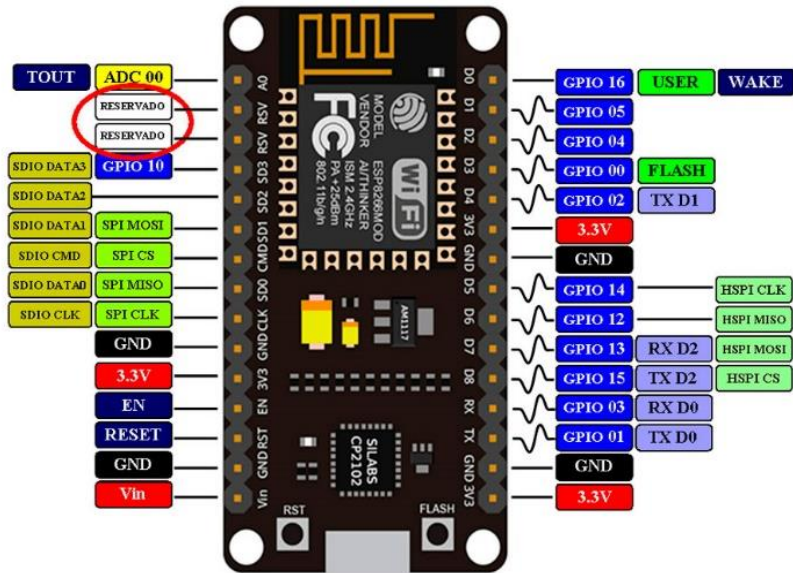
CARACTERÍSTICAS

- Voltaje de entrada (USB): 5V
- Voltaje de salida en los pines: 3.3V
- Corriente nominal por pin: 12mA
- Frecuencia de procesador: 80MHz (160MHz max.)
- 4MB Flash
- Consumo de corriente en stand-by @80MHz: 80mA
- Velocidad de transmisión óptima en baudios: 115200

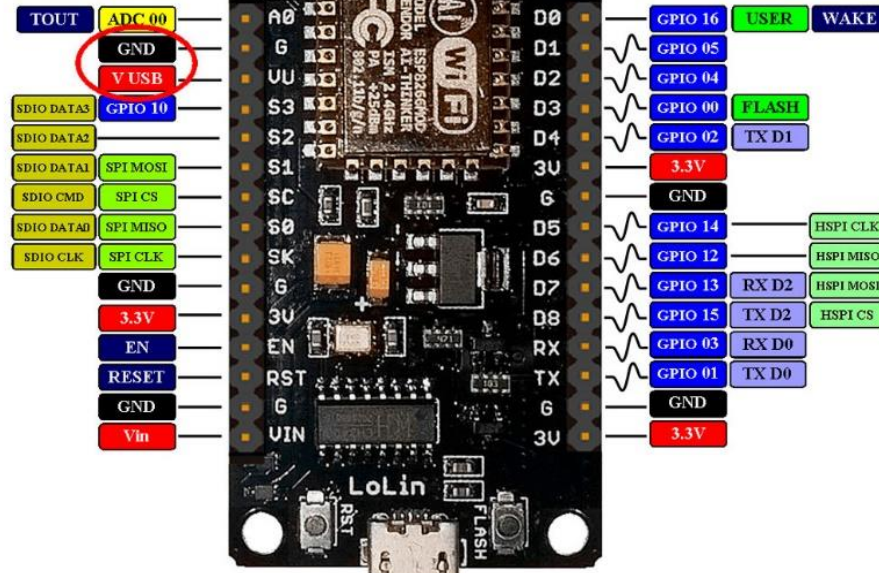


PINOUT DEL NODEMCU

AMICA / DOIT (V2)



LOLIN (V3)

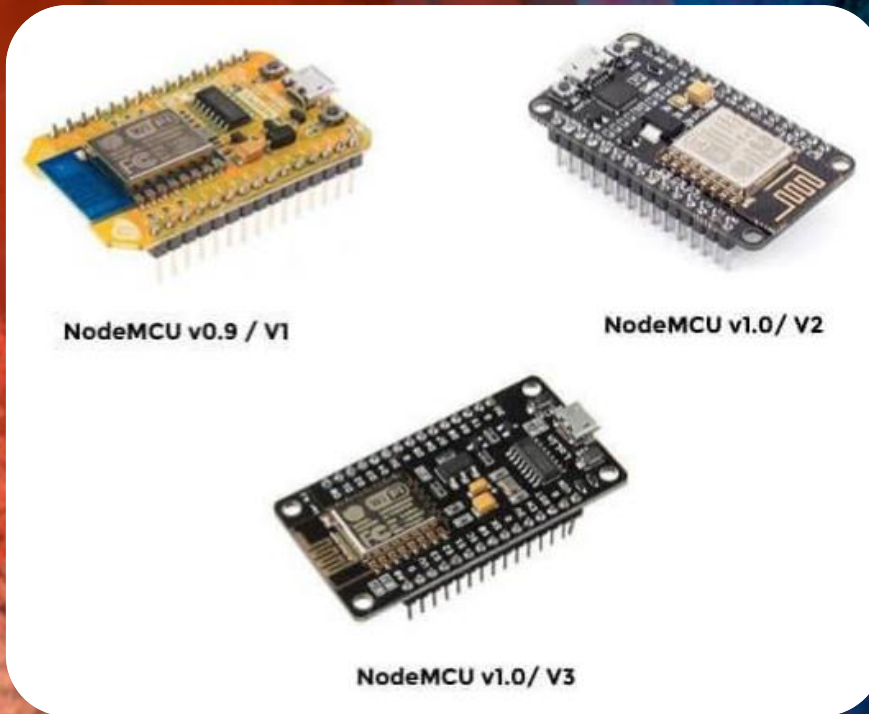


Vin Внешнее питание (5-10 вольт)
3.3 V Внутренняя шина питания (после стабилизатора)
V USB Питание с MiniUSB
GND Земля
GPIO Логические пины, +3.3 вольта максимум
 Цифровые — ШИМ ~

ADC Аналоговый вход
SPI SPI
HSPI HSPI
SDIO SD
TX/RX Последовательный (Serial, UART) порт

NOTA.- Los pines
 D0(GPIO16) Y
 D4 (GPIO2)
 controlan los led's
 integrados del
 NodeMCU.

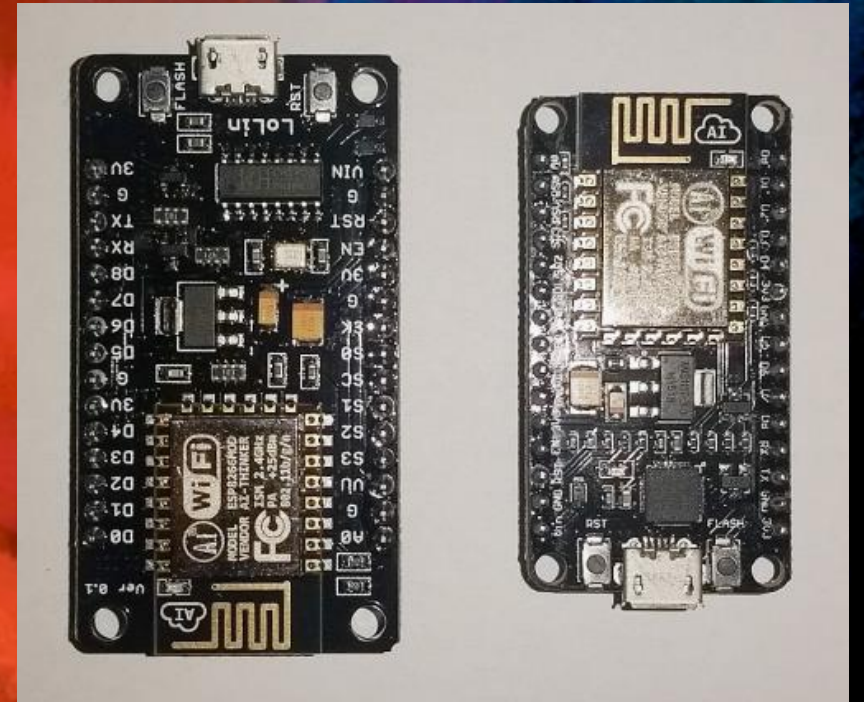
VERSIONES



NodeMCU v1.0 / V2

DIFERENCIA ENTRE AMICA Y LOLIN

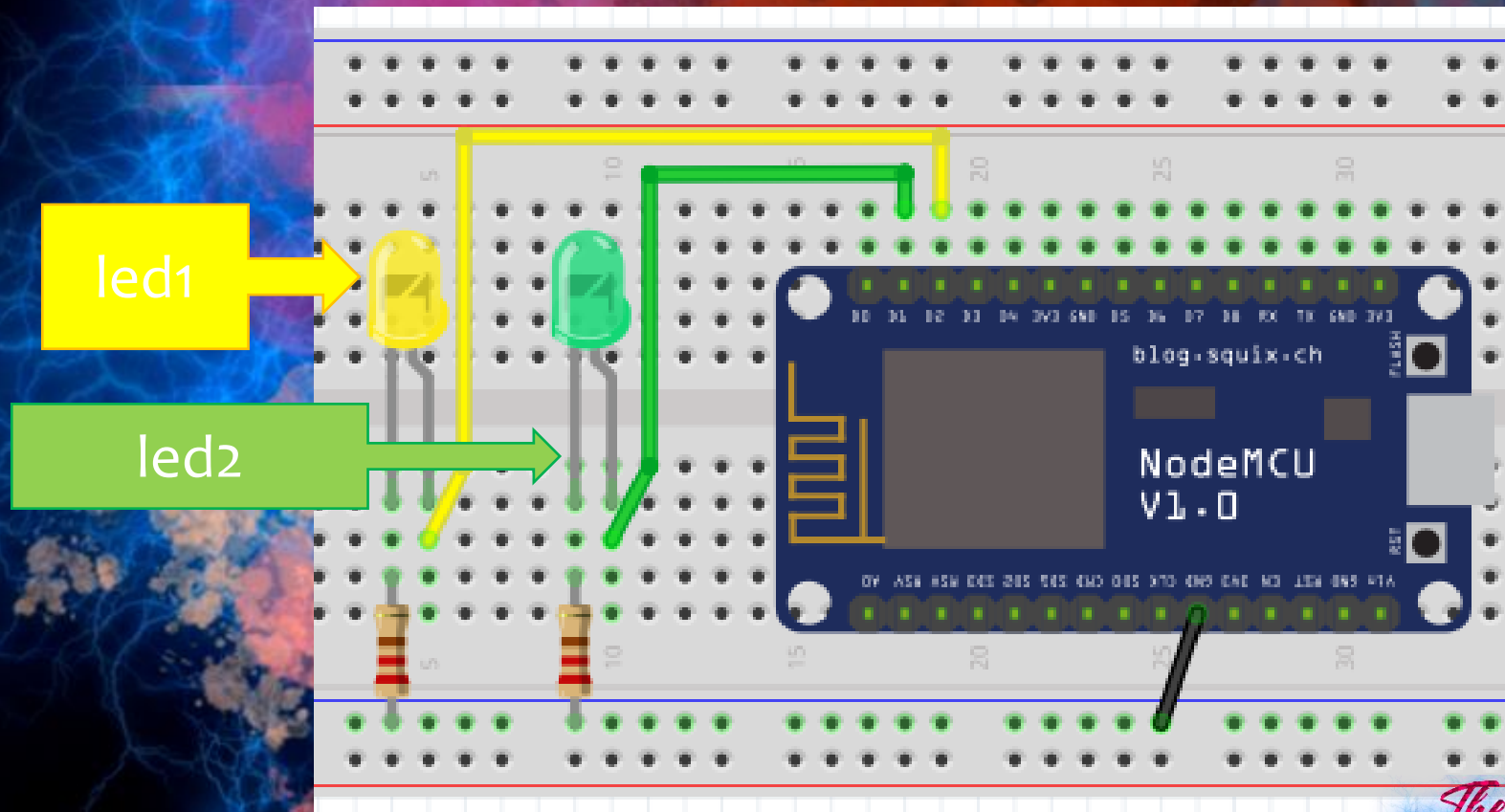
- ✓ Tamaño
- ✓ Fabricantes
- ✓ 2 pines de energía extra en lolin
- ✓ Amica es la versión oficial del NodeMCU v2.



EJERCICIOS DE APLICACIÓN

The
MAKERS
INFORMATICA

Conexión



1. Salidas digitales con el NodeMCU Amica

```
#define led1 D2  // definimos el pin digital 2
int led2 = 5;   // definimos el GPIO 5
void setup() {
    pinMode(led1,OUTPUT); //pin digital 2 de salida
    pinMode(led2,OUTPUT); //pin GPIO 5 de salida
}

void loop() {
    digitalWrite(led1,HIGH);
    delay(500);
    digitalWrite(led1,LOW);
    delay(300);
    digitalWrite(led2,HIGH);
    delay(600);
    digitalWrite(led2,LOW);
    delay(400);
}
```

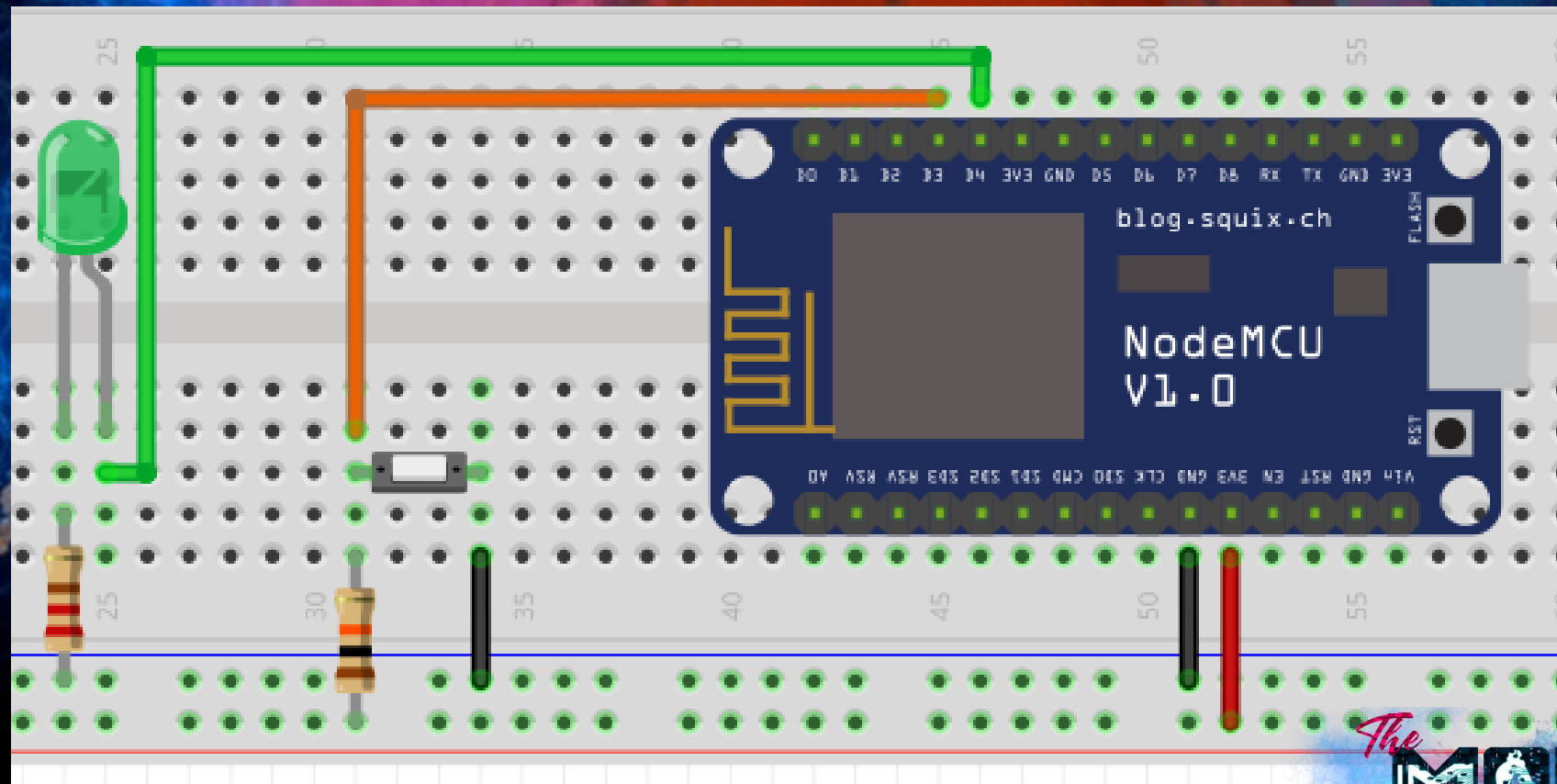

PULSADORES

Un pulsador es aquel componente electrónico que por lo general es activado al ser pulsado, es decir que básicamente este funciona como un interruptor eléctrico.

En su interior tiene dos contactos, NA(Normalmente Abierto) –NC (Normalmente Cerrado). Al ser pulsado maneja lo que es la lógica inversa de la que en ese momento esta



Conexión



2. Entradas digitales con el NodeMCU Amica

```
#define led D4//definimos el digital 5 para el led
#define b D3 //definimos el digital 6 para el boton
int v=0;      //definimos la variable de estado del pulsador
void setup() {
    pinMode(led,OUTPUT);
    pinMode(b,INPUT);
    Serial.begin(115200);
}

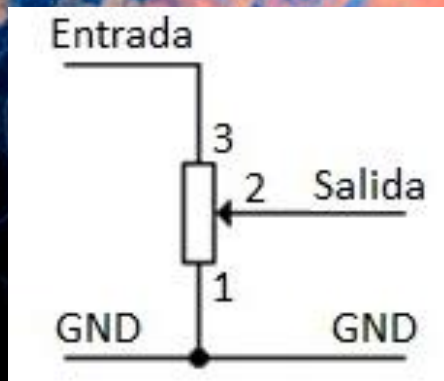
void loop() {
    v=digitalRead(b);
    if(v==HIGH){
        Serial.println("encendido");
        digitalWrite(led,HIGH);
        delay(200);
    }else{
        Serial.println("apagado");
        digitalWrite(led,LOW);
        delay(200);
    }
}
```


POTENCIOMETRO

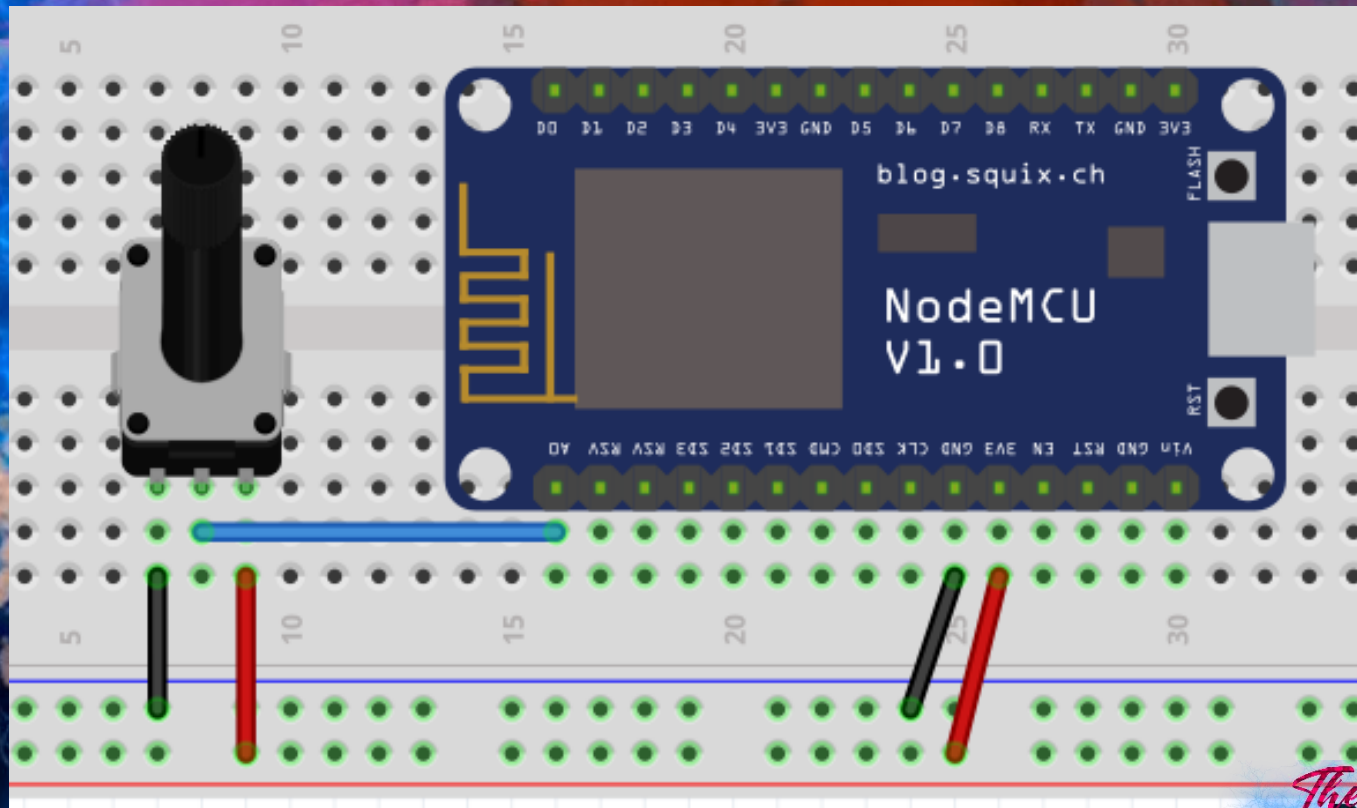
Un potenciómetro es un dispositivo conformado por 2 resistencias en serie, las cuales poseen valores que pueden ser modificados por el usuario. Existen múltiples tipos de potenciómetros, variando su forma y el método cómo modifican los valores de las resistencias.

Los potenciómetros que encontramos en el mercado vienen con un valor de resistencia determinado; por ejemplo 1K, 5K, 10k, 50k, 100k, etc.

Este valor de resistencia lo podemos medir entre las terminales 1 y 3 del potenciómetro



Conexión

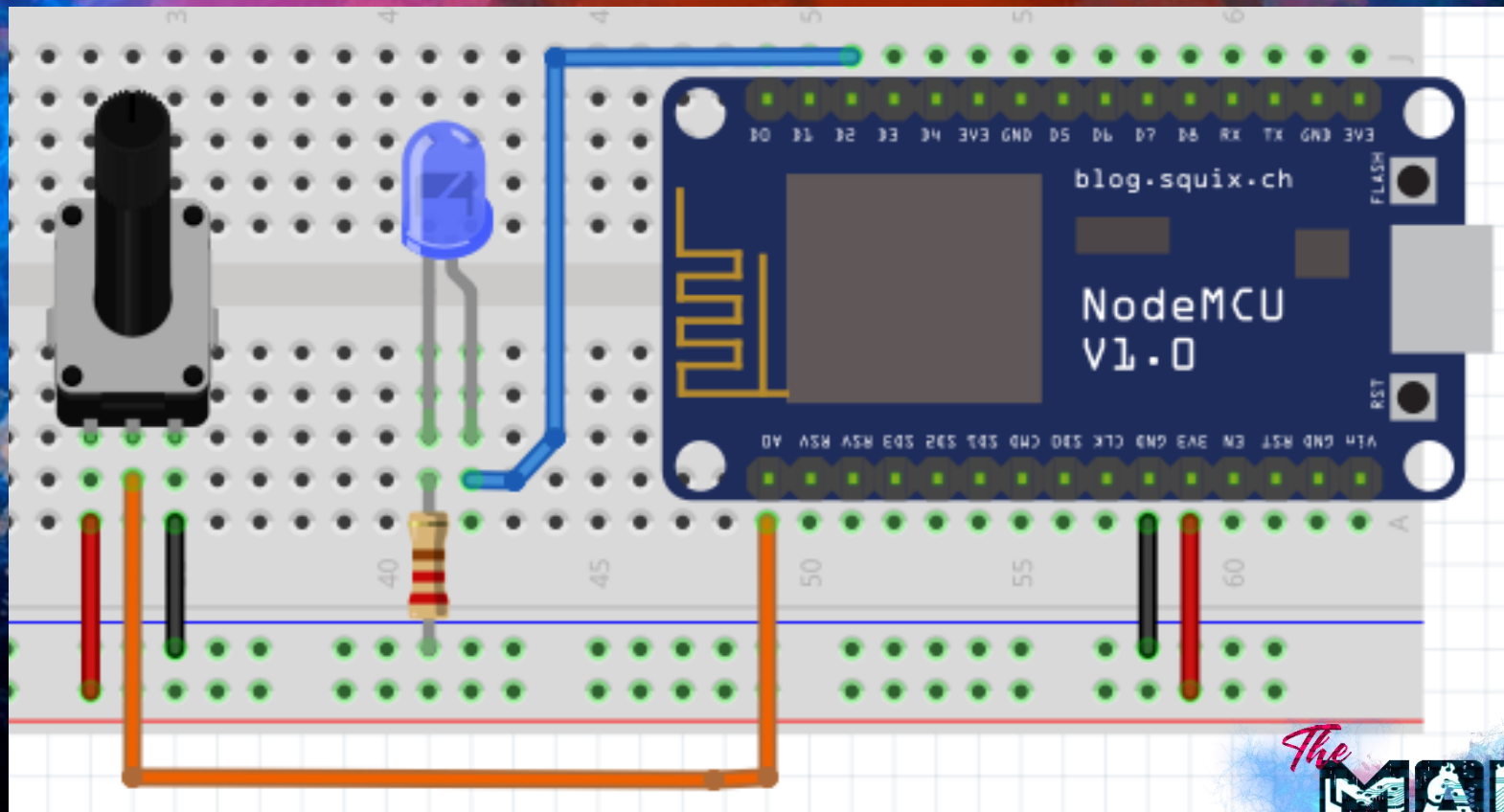


3. Lectura analógica con el NodeMCU Amica

```
#define pot A0 // definimos el pin analogico
void setup() {
    Serial.begin(115200); //velocidad de comunicacion optima para el node mcu
}

void loop() {
    Serial.println(analogRead(pot)); // realizar la lectura analógica
    delay(500);
}
```


Conexión



3. Salida PWM con el NodeMCU Amica

```
#define pot A0
#define led 4
int v;
void setup() {
    pinMode(pot, INPUT);
    pinMode(led, OUTPUT);
    Serial.begin(115200);
}

void loop() {
    v=map(analogRead(pot), 0, 1023, 0, 255);
    analogWrite(led, v);
    Serial.print("intensidad: ");
    Serial.println(v);
    delay(200);
}
```