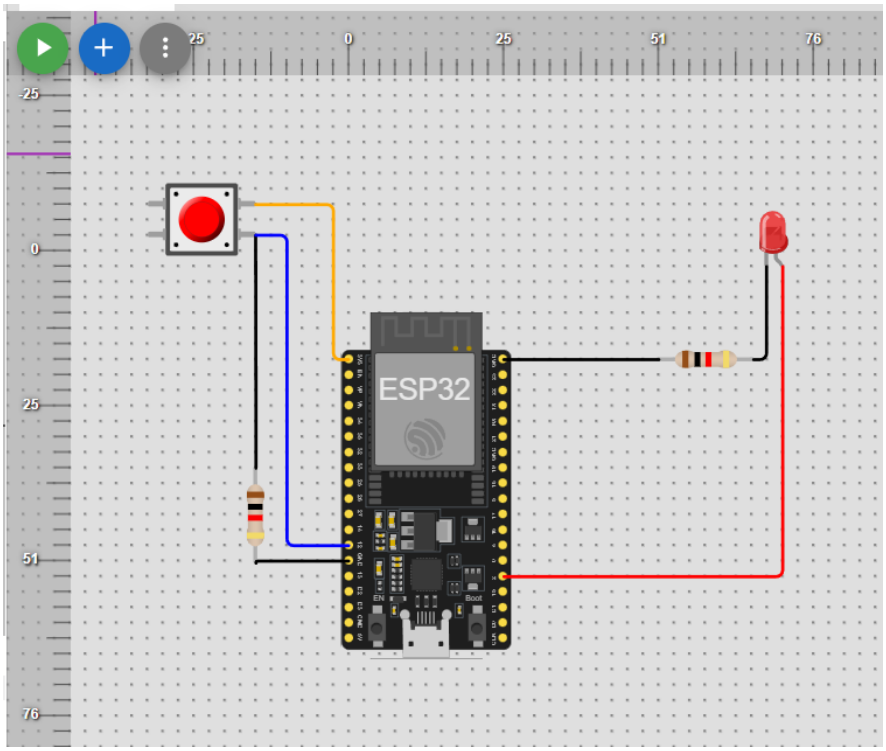
Simulaciones en Wokwi.

• Controlador de Entradas Digitales:
https://wokwi.com/projects/395693182472788993

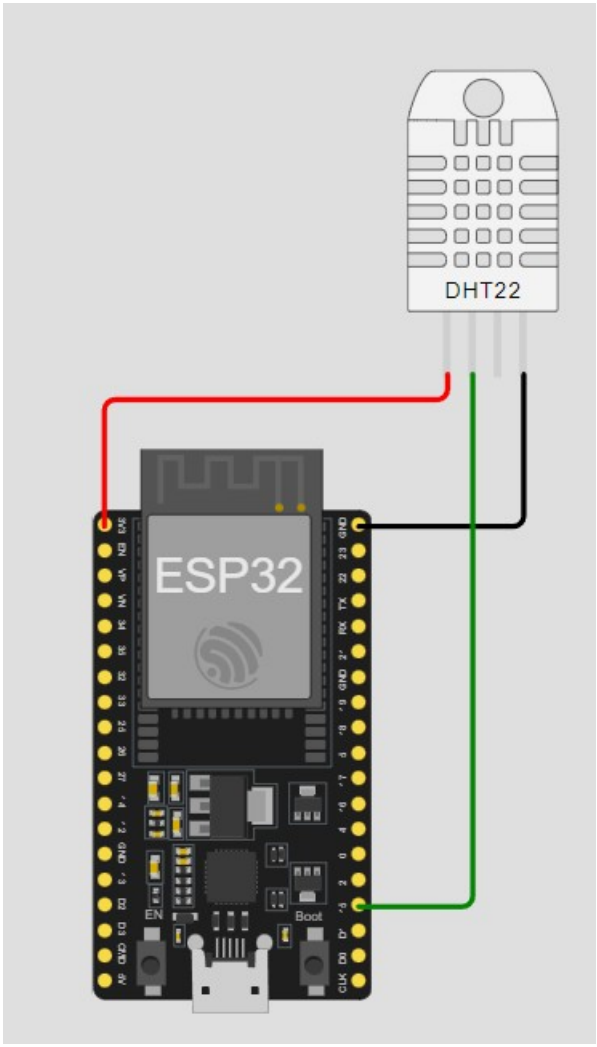



```
#IMPORTAR LOS MODULOS QUE QUE SE VA A TRABAJAR.
from machine import Pin
from time import sleep

#DECLARAR LOS OBJETOS.
button = Pin(12, Pin.IN)#entrada
led = Pin (2, Pin.OUT) #salida



#DESARROLLO DE CICLO Y CODIGO
print("Presiona el botón.")
while True:
    logic_state = button.value()
    if logic_state == True:
        led.value(1)
        sleep(0.5)
        print("Encendido")
    else:
        led.value(0)
```

Controlador de

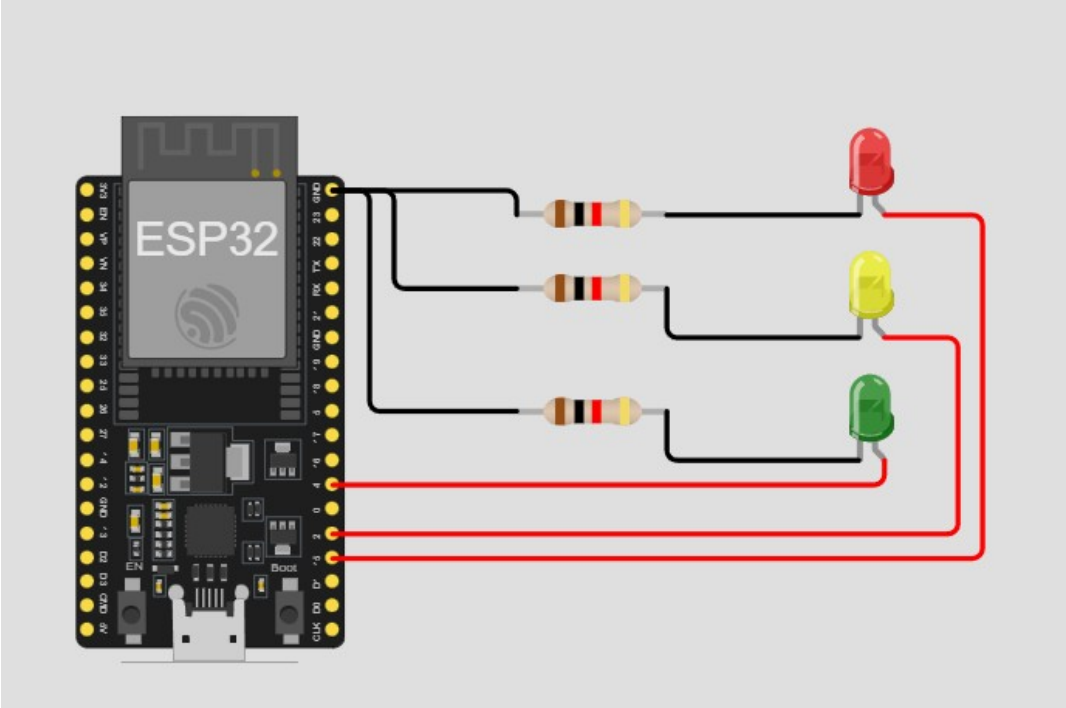
Entradas Analógicas:

https://wokwi.com/projects/395695422598276097




```python
from dht import DHT22 #importar modulo del sensor
from machine import Pin
from time import sleep
#objeto creado para el sensor
sensorDHT = DHT22(Pin(15))

while True:
    sleep(2)
    sensorDHT.measure() #metodo de medicion
    temp = sensorDHT.temperature() #metodo de temperatura
    hum = sensorDHT.humidity() #metodo de humedad
    kel = temp + 273 #metodo la establecido
    far = (temp * 9) / 5 + 32 #formula de temperatura
    print("T={:02}ºC  H={:02}%   K={:02}K   F={:02}F".format(temp, hum, kel, far))

```

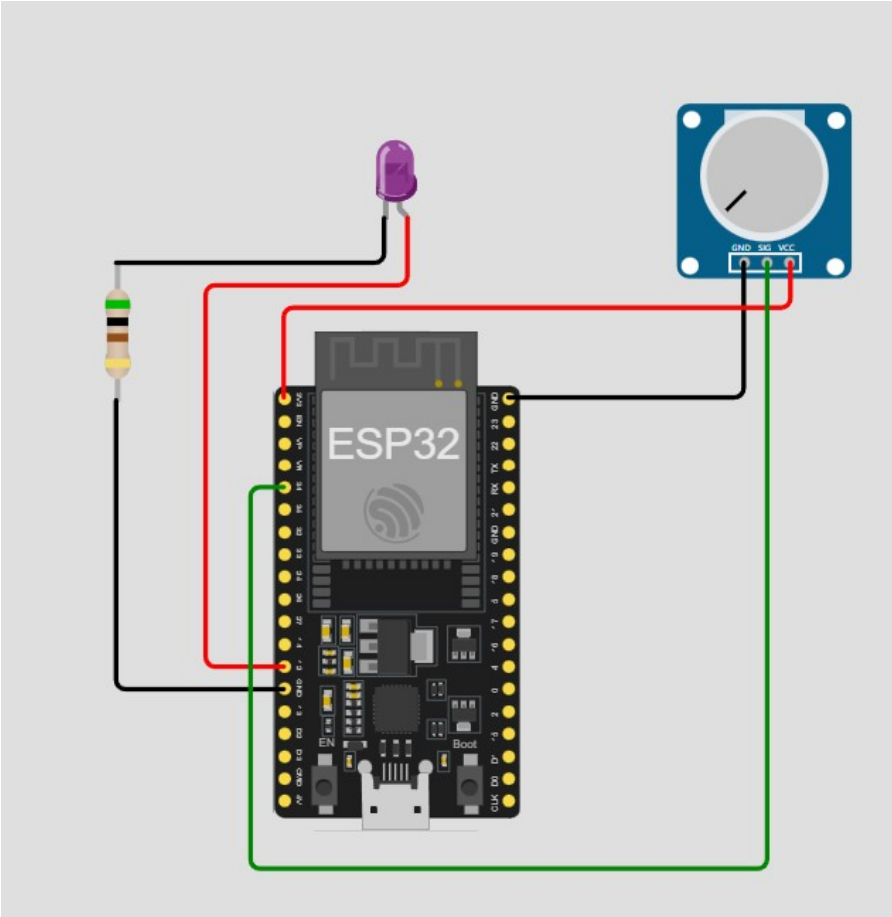
• Controlador de Salidas Digitales:https://wokwi.com/projects/395728090692756481




```
#importar modulos
from machine import Pin
from time import sleep

#crear objetos "leds"
led_rojo = Pin(15, Pin.OUT)
led_amarillo = Pin(2, Pin.OUT)
led_verde = Pin(4, Pin.OUT)

while True:#condicion de intervalos de encendidos
    led_rojo.value(1)
    print("ROJO")
    sleep(3)
    led_rojo.value(0)
    led_amarillo.value(1)
    print("AMARILLO")
    sleep(2)
    led_amarillo.value(0)
    led_verde.value(1)
    print("VERDE")
    sleep(3)
    led_verde.value(0)
    sleep(1)
```

• Controlador de Salidas Analógicas:

https://wokwi.com/projects/395796015724558337




```
# imporatar los modulos
from machine import Pin, ADC, PWM
from time import sleep
# crear objetos
freq = 5000
led = PWM(Pin(12), freq)
pot = ADC(Pin(34))
pot.width(ADC.WIDTH_10BIT)#configurar a 10 bit
pot.atten(ADC.ATTN_11DB)#definir el rango del potenciometro 0v a 3.3v


while True:
    pot_valor = pot.read()
    print(pot_valor)
    led.duty(pot_valor)
    sleep(0.2)

```

Profesor: Gonzalo Vera.

Alumno: Nicolás Barrionuevo.