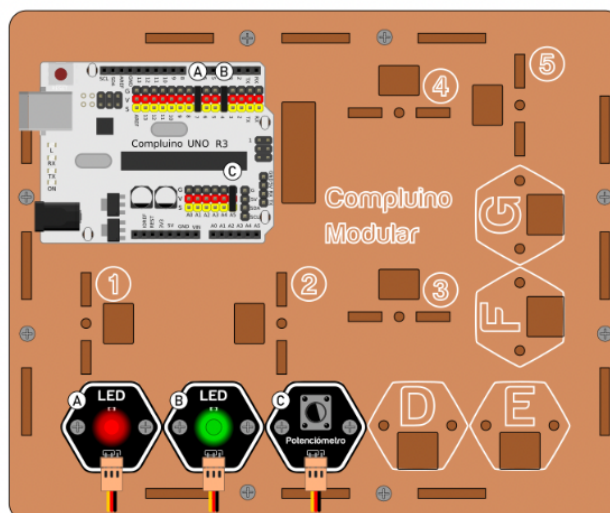
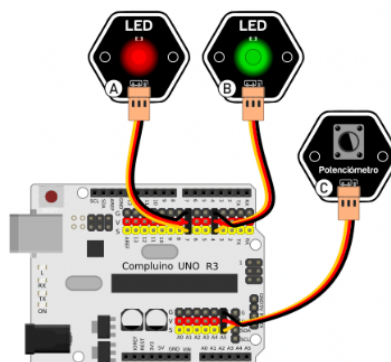


Curso 2022-2023

Prácticas de COMPLUINO

Polos Creativos
IES A Xunqueira 1

POLOS
CREATIVOS



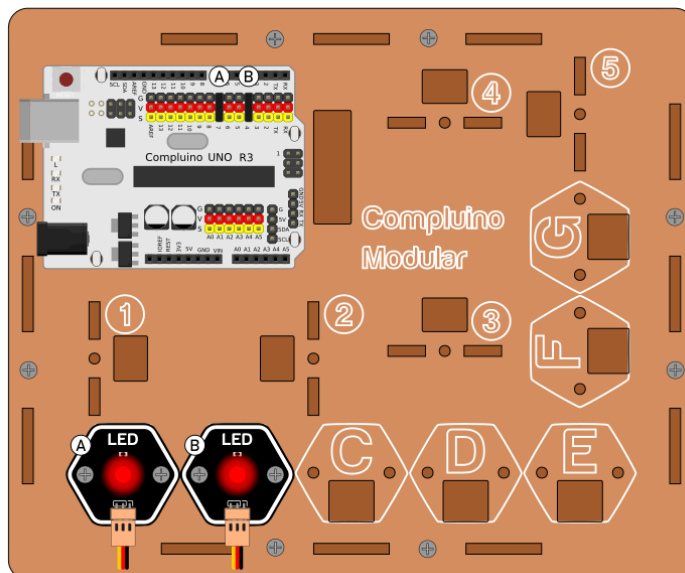
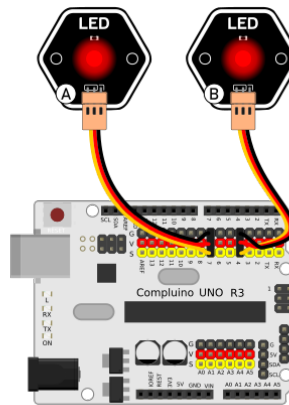
TALLER 1: PARPADEO DE DOUS LEDS VERMELLOS CADA 2000 MILISEGUNDOS NUNHA NAVE ESPACIAL

OBXECTIVO

Empregar o material asignado para o proxecto “Polos Creativos”
“Welcome to Mars”

WELCOME TO MARS

O kit de traballo COMPLUINO Modular está formado por sensores e actuadores que se conectan directamente cun único cable de 3 pins coa placa COMPLUINO UNO.



CÓDIGO COMPLUINO

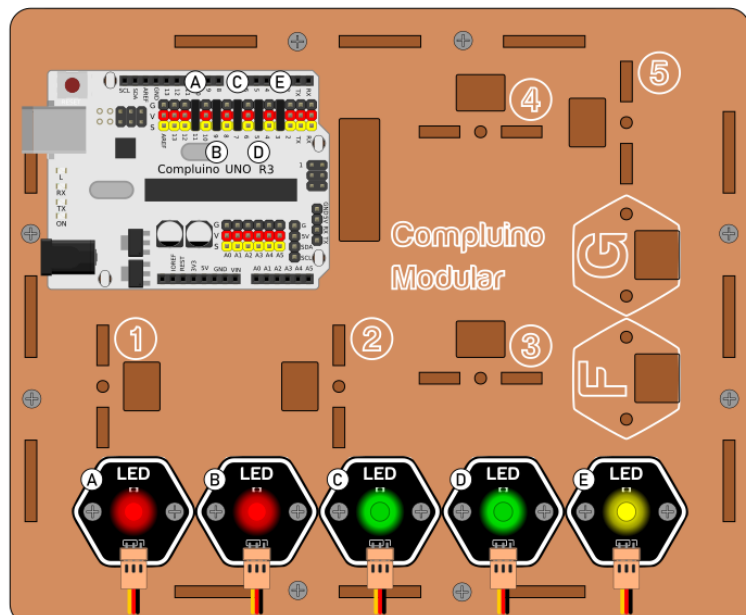
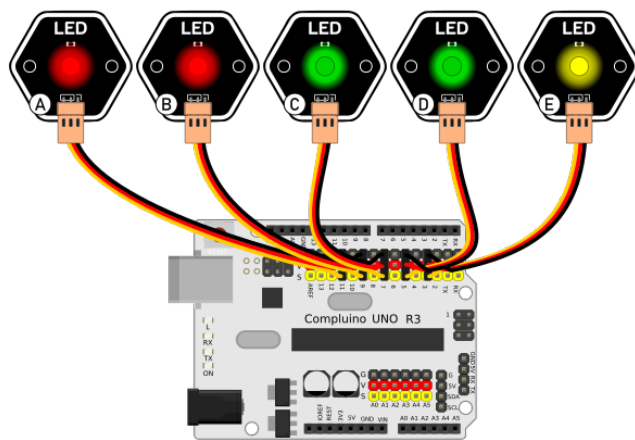
```
#define LEDA 7

#define LEDB 4

void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 7 como salida.
  pinMode(LEDB, OUTPUT); // Definir el pin 4 como salida.
}

void loop()
{
  digitalWrite (LEDA, HIGH); // Encender el led A
  digitalWrite (LEDB, HIGH); // Encender el led B
  delay (2000); // Esperar dos segundos
  digitalWrite (LEDA, LOW); // Apagar el led A
  digitalWrite (LEDB, LOW); // Apagar el led B
  delay (2000); // Esperar dos segundos
}
```

TALLER 2: ILUMINACIÓN SECUENCIAL DE CINCO LEDS NUNHA SONDA EN MARTE



CÓDIGO COMPLUINO

```
#define LEDA 11

#define LEDB 9

#define LEDC 7

#define LEDD 5

#define LEDE 3

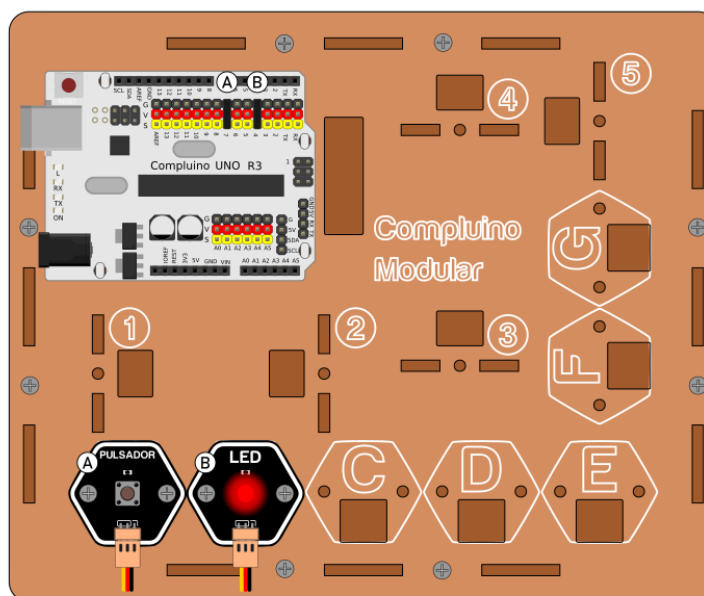
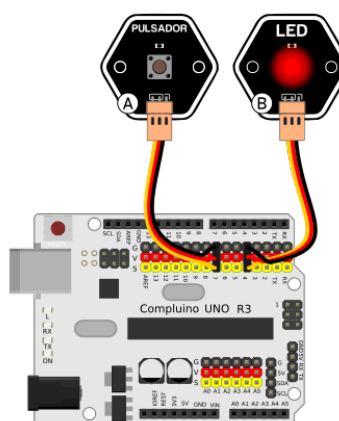
#define TIEMPO 125

void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 11 como salida
  pinMode(LEDB, OUTPUT); // Definir el pin 9 como salida
  pinMode(LEDC, OUTPUT); // Definir el pin 7 como salida
  pinMode(LEDD, OUTPUT); // Definir el pin 5 como salida
  pinMode(LEDE, OUTPUT); // Definir el pin 3 como salida
}

void loop()
{
  digitalWrite (LEDA, HIGH); // Encender el led A
```

```
delay (TIEMPO); // Esperar 0.125 segundos
digitalWrite (LEDA, LOW); // Apagar el led A
digitalWrite (LEDB, HIGH); // Encender el led B
delay (TIEMPO); // Esperar 0.125 segundos
digitalWrite (LEDB, LOW); // Apagar el led B
digitalWrite (LEDC, HIGH); // Encender el led C
delay (TIEMPO); // Esperar 0.125 segundos
digitalWrite (LEDC, LOW); // Apagar el led C
digitalWrite (LEDD, HIGH); // Encender el led D
delay (TIEMPO); // Esperar 0.125 segundos
digitalWrite (LEDD, LOW); // Apagar el led D
digitalWrite (LEDE, HIGH); // Encender el led E
delay (TIEMPO); // Esperar 0.125 segundos
digitalWrite (LEDE, LOW); // Apagar el led E
}
```

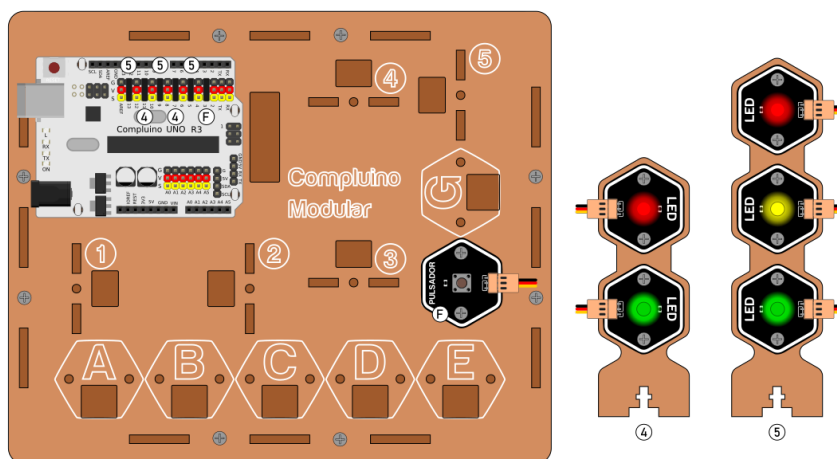
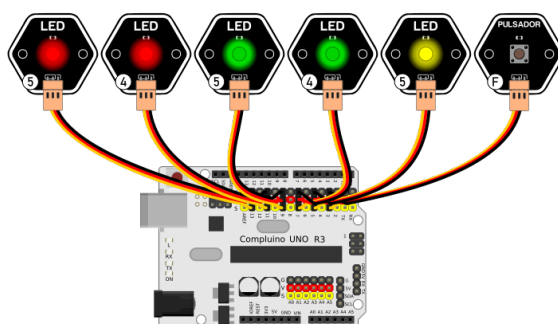
TALLER 3: ACENDER OU APAGAR UN LED
DEPENDENDO DO ESTADO DUN PULSADOR



CÓDIGO COMPLUINO

```
#define LED 4
#define PULSADOR 7
void setup()
{
  pinMode(LED, OUTPUT); // Definir el pin 4 como salida
  pinMode(PULSADOR, INPUT); // Definir el pin 7 como entrada
}
void loop()
{
  if (digitalRead(PULSADOR)==HIGH) //Comparar el estado que se lee
  en el pin del pulsador
  {
    digitalWrite(LED,HIGH); // Si se cumple la condición enciende el led
  }
  else
  {
    digitalWrite(LED,LOW); // Si no se cumple la condición apaga el led
  }
}
```

TALLER 4: SEMÁFORO PARA PEÓNS E
VEHÍCULOS CON PULSADOR NUNHA CIDADE
DE MARTE

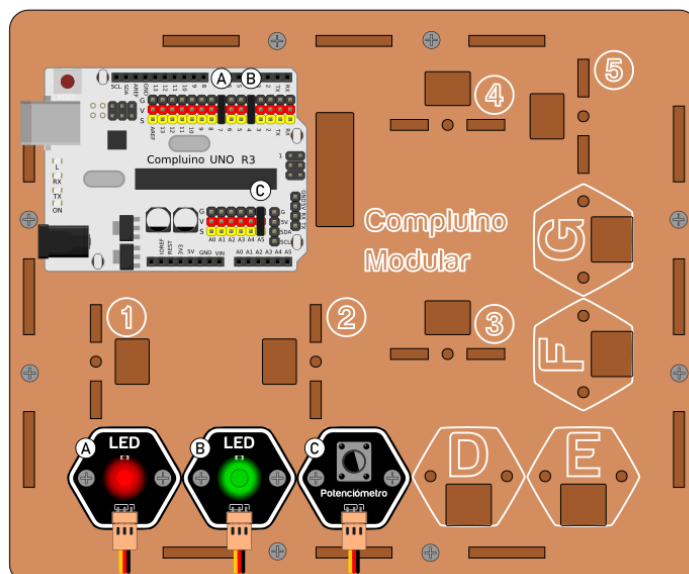
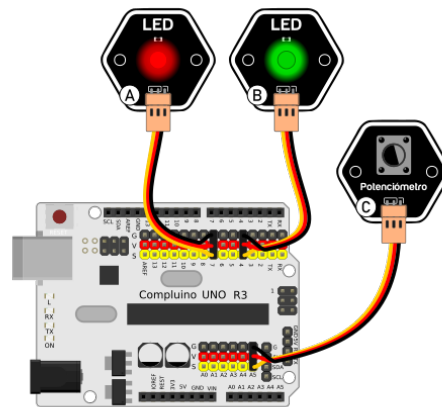


CÓDIGO COMPLUINO

```
#define C_ROJO 13
#define C_AMBAR 5
#define C_VERDE 9
#define P_ROJO 11
#define P_VERDE 7
#define PULSADOR 3
void setup()
{
  pinMode(C_ROJO, OUTPUT); // Definir el pin 13 como salida
  pinMode(C_AMBAR, OUTPUT); // Definir el pin 5 como salida
  pinMode(C_VERDE, OUTPUT); // Definir el pin 9 como salida
  pinMode(P_ROJO, OUTPUT); // Definir el pin 11 como salida
  pinMode(P_VERDE, OUTPUT); // Definir el pin 7 como salida
  pinMode(PULSADOR, INPUT); // Definir el pin 3 como entrada
}
void loop()
{
  digitalWrite(P_ROJO, HIGH); // Vehículos circulan y los peatones
esperan en rojo hasta que se pulse
  digitalWrite(C_VERDE, HIGH);
  while (digitalRead(PULSADOR) == LOW);
  delay(1000);
  digitalWrite(C_VERDE, LOW); //Vehículos en ámbar
  digitalWrite(C_AMBAR, HIGH);
  delay(1000);
}
```

```
digitalWrite(C_AMBAR, LOW); //Vehículos en rojo
digitalWrite(C_ROJO, HIGH);
delay(500);
digitalWrite(P_VERDE, HIGH); //Peatones en verde mientras los
vehículos esperan en rojo
delay(2000);
for (int x = 0; x < 5; x++) //Parpadea el semáforo verde de los
peatones cinco veces
{
    digitalWrite(P_VERDE, HIGH);
    delay(250);
    digitalWrite(P_VERDE, LOW);
    delay(250);
}
digitalWrite(P_ROJO, HIGH); //Ambos semáforos en rojo para no
atropellar a los peatones lentos
delay(500);
}
```

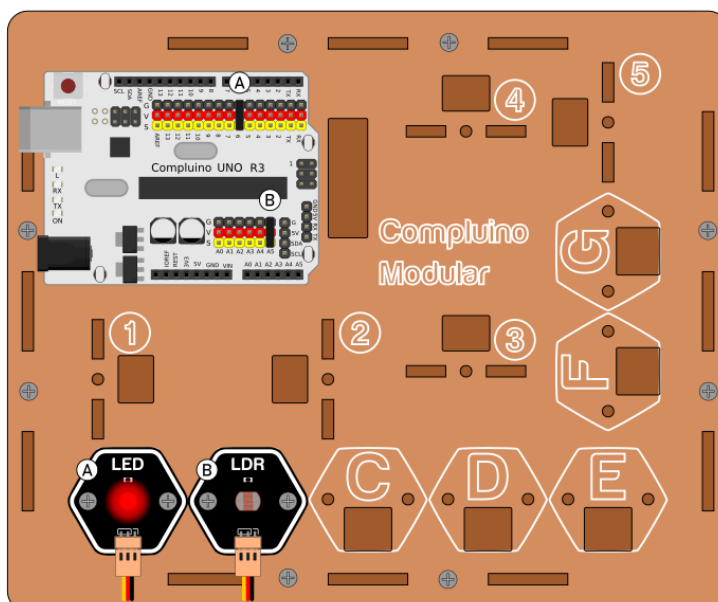
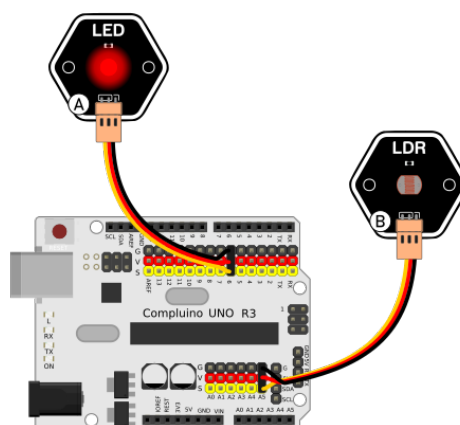
TALLER 5: ACENDER O LED VERDE OU O
VERMELHO DEPENDENDO DO VALOR DO
POTENCIÓMETRO



CÓDIGO COMPLUINO

```
#define POT A5
#define LEDA 7
#define LEDB 4
int lectura_pot = 0;
void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 7 como salida
  pinMode(LEDB, OUTPUT); // Definir el pin 4 como salida
}
void loop()
{
  lectura_pot = analogRead(POT); // Guardar el valor leído en el pin A5
  en una variable
  if (lectura_pot >= 512) // Comparar la lectura del potenciómetro con un valor de corte
  {
    digitalWrite(LEDA, HIGH);
    digitalWrite(LEDB, LOW);
  }
  else
  {
    digitalWrite(LEDB, HIGH);
    digitalWrite(LEDA, LOW);
  }
}
```

TALLER 6: CONTROLAR O BRILLO DUN LED CA
LDR



CÓDIGO COMPLUINO

```
#define LDR A5

#define LED 6

int lectura_ldr = 0;

int brillo = 0;

void setup()

{

  pinMode(LED, OUTPUT);

}

void loop()

{

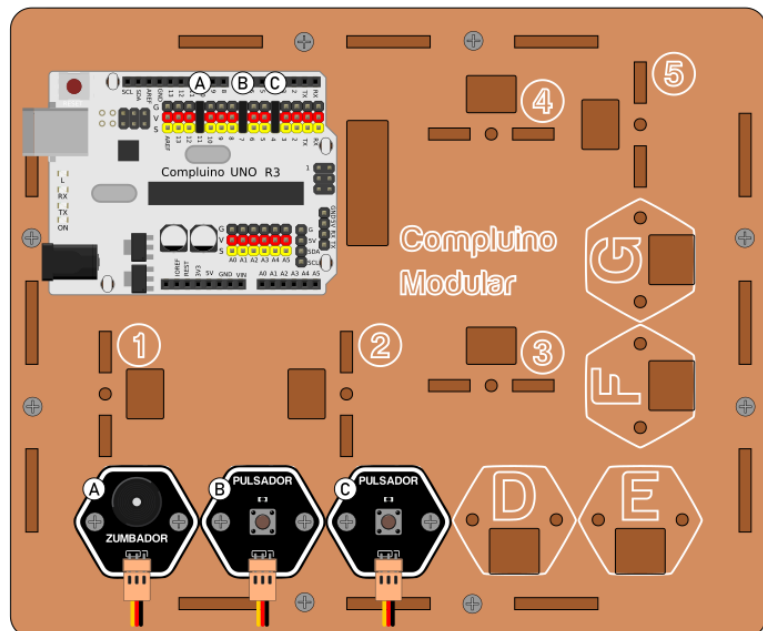
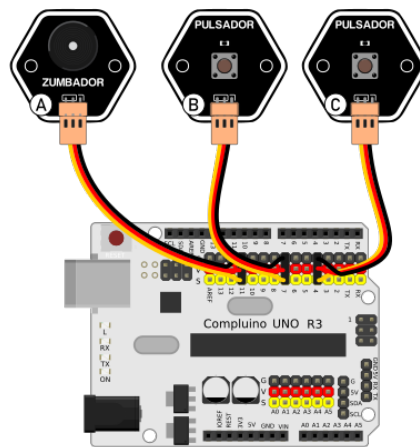
  lectura_ldr = analogRead(LDR); //Guardar el valor leído en el pin A5
  en una variable

  brillo = map(lectura_ldr, 0, 1023, 0, 255); //Convertir el valor a un
  rango de 0 a 255

  analogWrite(LED, brillo);

}
```

TALLER 7: DOUS PULSADORES ACTÍVANSE
CON TONOS DIFERENTES NO ZOADOR

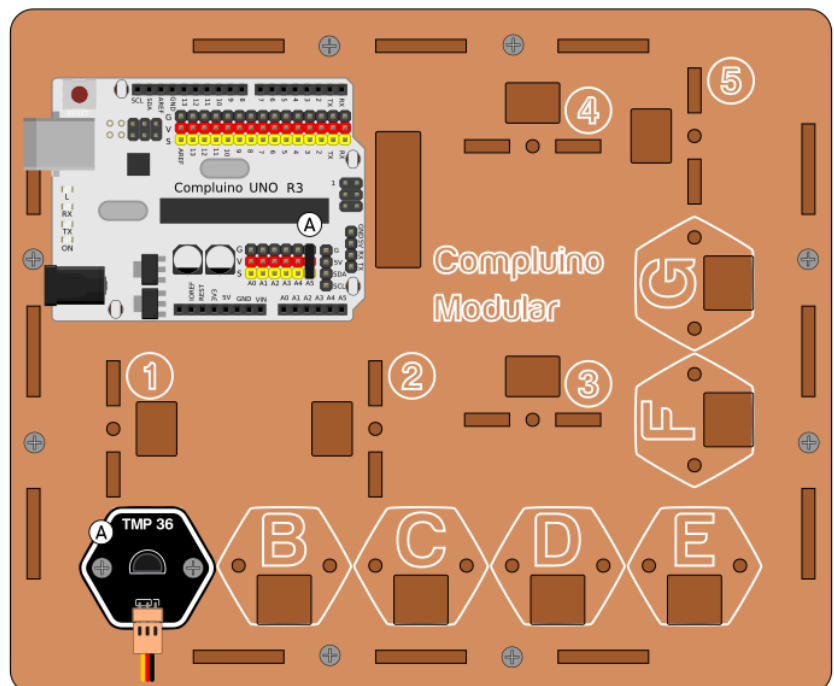
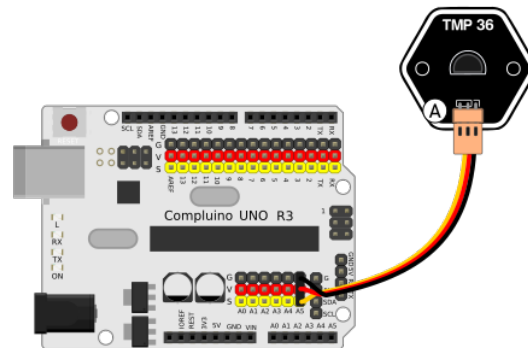


CÓDIGO COMPLUINO

```
#define PULSADORB 7
#define PULSADORC 4
#define ZUMBADOR 11
void setup()
{
  pinMode(PULSADORB, INPUT);
  pinMode(PULSADORC, INPUT);
  pinMode(ZUMBADOR, OUTPUT);
  tone(ZUMBADOR, 293); //Nota RE
  delay(200);
  tone(ZUMBADOR, 440); //Nota LA
  delay(200);
  noTone(ZUMBADOR);
}
void loop()
{
  if (digitalRead(PULSADORB) == HIGH)
  {
    tone(ZUMBADOR, 659); //Nota MI
  }
  else
  {
    if (digitalRead(PULSADORC) == HIGH)
    {
```

```
tone(ZUMBADOR, 392); //Nota SOL
}
else
{
noTone(ZUMBADOR);
}
}
}
```

TALLER 8: MEDIR A TEMPERATURA E
MOSTRALA NO MONITOR SERIE



CÓDIGO COMPLUINO

```
#define TEMP A5

float temperatura = 0, medida_sensor = 0, voltaje = 0;

void setup()
{
  Serial.begin(9600);
}

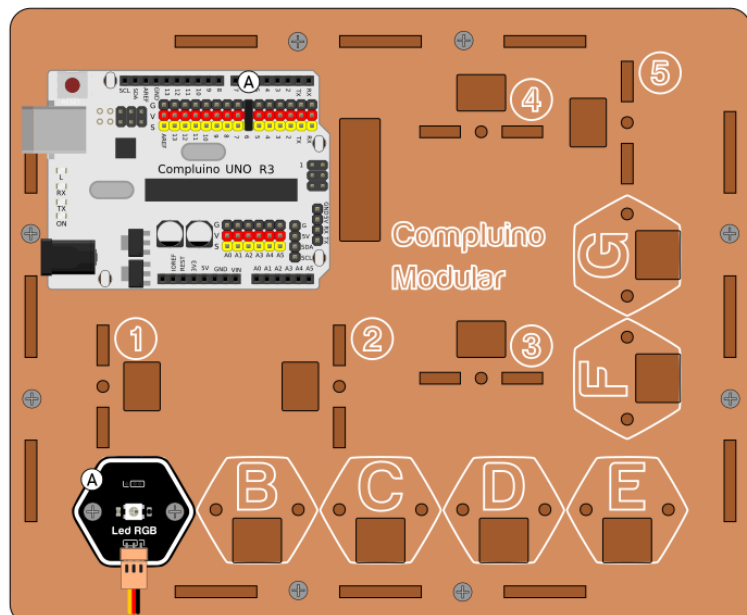
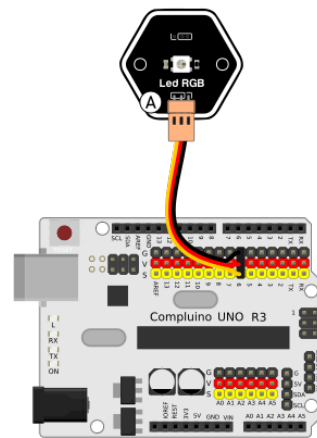
void loop()
{
  medida_sensor = analogRead(TEMP); //Actualizamos el valor del
  sensor de temperatura

  voltaje = medida_sensor * 5 / 1024; //convierte cuentas a voltios
  temperatura = (voltaje * 100) - 50; //convierte de voltios a grados C°

  Serial.print("La temperatura actual es: ");
  Serial.print(temperatura);
  Serial.println(" grados");

  delay(500);
}
```

TALLER 9: SECUENCIA DE OITO CORES CO
LED RGB E LA LIBRERÍA NEOPIXEL



CÓDIGO COMPLUINO

```
#include <Adafruit_NeoPixel.h>
#define PIN_RGB 6 //Pin del rgb
Adafruit_NeoPixel rgb(1, PIN_RGB, NEO_GRB + NEO_KHZ400); //Conf
igura 1 pixel RGB
void setup()
{
  rgb.begin();
}
void loop()
{
  rgb.setPixelColor(0, rgb.Color(255, 0, 0)); //Muestra el color rojo
  rgb.show();
  delay(2000);
  rgb.setPixelColor(0, rgb.Color(0, 255, 0)); //Muestra el color verde
  rgb.show();
  delay(2000);

  rgb.setPixelColor(0, rgb.Color(0, 0, 255)); //Muestra el color azul
  rgb.show();
  delay(2000);

  rgb.setPixelColor(0, rgb.Color(255, 255, 0)); //Muestra el color
  amarillo
  rgb.show();
```

```
delay(2000);
```

```
rgb.setPixelColor(0, rgb.Color(255, 0, 255)); //Muestra el color  
magenta
```

```
rgb.show();
```

```
delay(2000);
```

```
rgb.setPixelColor(0, rgb.Color(0, 255, 255)); //Muestra el color cian  
rgb.show();
```

```
delay(2000);
```

```
rgb.setPixelColor(0, rgb.Color(255, 255, 255)); //Muestra el color  
blanco
```

```
rgb.show();
```

```
delay(2000);
```

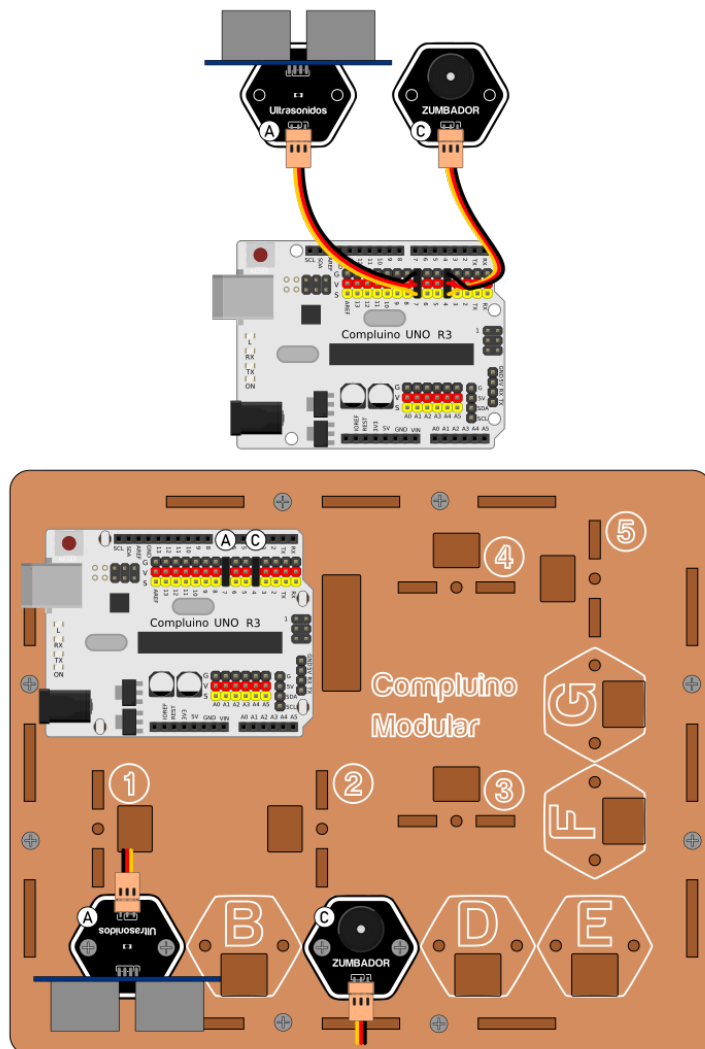
```
rgb.clear(); //Apaga el RGB
```

```
rgb.show();
```

```
delay(2000);
```

```
}
```

TALLER 10: MEDIR DISTANCIA CO SENSOR
ULTRASÓNICO E EMPREGALAS PARA
MODIFICAR O SON DO ZOADOR



CÓDIGO COMPLUINO

```
#define PIN_US 7 //Pin del ultrasonido

#define ZUMBADOR 4 //Pin del zumbador

float tiempo; //Variable para guardar el tiempo

float distancia; //Variable para guardar la distancia

void setup()

{

  pinMode(ZUMBADOR,OUTPUT); //Configuramos el zumbador como
  salida

}

void loop()

{

  pinMode(PIN_US,OUTPUT); //Configuramos el pin como salida

  digitalWrite(PIN_US,LOW);

  delayMicroseconds(5);

  digitalWrite(PIN_US,HIGH); //Emitimos un echo por el ultrasonidos

  delayMicroseconds(10);

  digitalWrite(PIN_US,LOW);
```

```
//Medimos el tiempo que tarda el sonido en ir y volver  
  
pinMode(PIN_US,INPUT); //Cambiamos el pin a entrada  
  
tiempo = pulseIn(PIN_US,HIGH);  
  
//Calculamos la distancia en cm  
  
distancia = tiempo/58;  
  
//Cambiamos la frecuencia del pitido en funcion de la distancia  
  
tone(ZUMBADOR,440);  
  
delay(map(distancia,0,255,50,1000)); //La espera va desde los 50 ms  
a 1s  
  
noTone(ZUMBADOR);  
  
delay(map(distancia,0,255,50,1000));  
  
}
```