

### **Desafio 3 - Aula 4:**

Código utilizado:

```
int led = 3;
int botao = 2;
int contador = 0;

bool estado_anterior_botao = LOW;
bool estado_atual_botao = LOW;
bool estado_atual_led = true;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop()
{
  estado_atual_botao = debounce(estado_anterior_botao);

  if(estado_anterior_botao == LOW && estado_atual_botao == HIGH){
    estado_atual_led = !estado_atual_led;
    Serial.println(contador);
    contador++;
  }
  estado_anterior_botao = estado_atual_botao;
  digitalWrite(led, estado_atual_led);
}

// Função debounce:
boolean debounce(boolean anterior){
  boolean atual = digitalRead(botao);
  if(anterior!=atual){
    delay(5);
    atual = digitalRead(botao);
  }
  return atual;
}
```

#### **Desafio 4 - Aula 5:**

Código utilizado:

```
int Potenciometro = A0;
int LED = 3;
int leitura;
int intensidade;
int PWM1 = 5;
int PWM2 = 6;
int PWM3 = 9;

void setup(){
  pinMode(PWM1, OUTPUT);
  pinMode(PWM2, OUTPUT);
  pinMode(PWM3, OUTPUT);
}
void loop(){
  leitura = analogRead(Potenciometro);
  //LED RGB:
  if (leitura == 0){
    analogWrite(PWM1,0);
    analogWrite(PWM2,0);
    analogWrite(PWM3,0);
  }
  // verde
  if (leitura > 0 && leitura < 256){
    analogWrite(PWM1,0);
    analogWrite(PWM2,0);
    analogWrite(PWM3,255);
  }
  // azul
  if (leitura >= 256 && leitura < 512){
    analogWrite(PWM1,0);
    analogWrite(PWM2,255);
    analogWrite(PWM3,0);
  }
  // rosa
  if (leitura >= 512 && leitura < 768){
    analogWrite(PWM1,255);
    analogWrite(PWM2,255);
    analogWrite(PWM3,0);
  }
  // vermelho
  if (leitura >= 768 && leitura < 1024){
    analogWrite(PWM1,255);
    analogWrite(PWM2,0);
    analogWrite(PWM3,0);
  }
}
```

```

//LED azul:
intensidade = map(leitura, 0, 1023, 0, 255);
analogWrite(LED, intensidade);
}

```

### **Desafio 5 (final) - Aula 6:**

Código utilizado:

```

#include <Servo.h>
#define pinoServo 8
Servo Servo_1;

int Interruptor = 7;
int Potenciometro = A0;
int PWM1 = 5;
int PWM2 = 6;
int PWM3 = 9;
int i = 0;
bool leitura_Interruptor;
int leitura_Potenciometro;
int angulo_Servo;

void setup()
{
  pinMode(Interruptor, INPUT);
  pinMode(PWM1, OUTPUT);
  pinMode(PWM2, OUTPUT);
  pinMode(PWM3, OUTPUT);

  Servo_1.attach(pinoServo);
}
void loop()
{
  leitura_Interruptor = digitalRead(Interruptor);
  leitura_Potenciometro = analogRead(Potenciometro);

  if (leitura_Interruptor == true){

    // o laço while foi usado para o led ficar na cor verde assim que ligamos a
    máquina e logo depois de 5s já se pode alterar a cor usando o
    potenciometro, pois assim não haverá mistura da cor verde de quando se
    liga a máquina com as outras cores quando o mesmo for usado

    while(i<1){
      analogWrite(PWM1,0);
      analogWrite(PWM2,0);
      analogWrite(PWM3,255);
      delay(5000);
      i += 1;
    }
  }
}

```

```

    }
    // verde
    if (leitura_Potenciometro >= 0 && leitura_Potenciometro < 256){
        analogWrite(PWM1,0);
        analogWrite(PWM2,0);
        analogWrite(PWM3,255);
    }

    // azul
    if (leitura_Potenciometro >= 256 && leitura_Potenciometro < 512){
        analogWrite(PWM1,0);
        analogWrite(PWM2,255);
        analogWrite(PWM3,0);
    }
    // rosa
    if (leitura_Potenciometro >= 512 && leitura_Potenciometro < 768){
        analogWrite(PWM1,255);
        analogWrite(PWM2,255);
        analogWrite(PWM3,0);
    }
    // vermelho
    if (leitura_Potenciometro >= 768 && leitura_Potenciometro < 1024){
        analogWrite(PWM1,255);
        analogWrite(PWM2,0);
        analogWrite(PWM3,0);
    }
    angulo_Servo = map(leitura_Potenciometro, 0, 1023, 0, 180);
    Servo_1.write(angulo_Servo);
} else {
    analogWrite(PWM1,0);
    analogWrite(PWM2,0);
    analogWrite(PWM3,0);

    angulo_Servo = 0;
    Servo_1.write(angulo_Servo);
}
}

```