

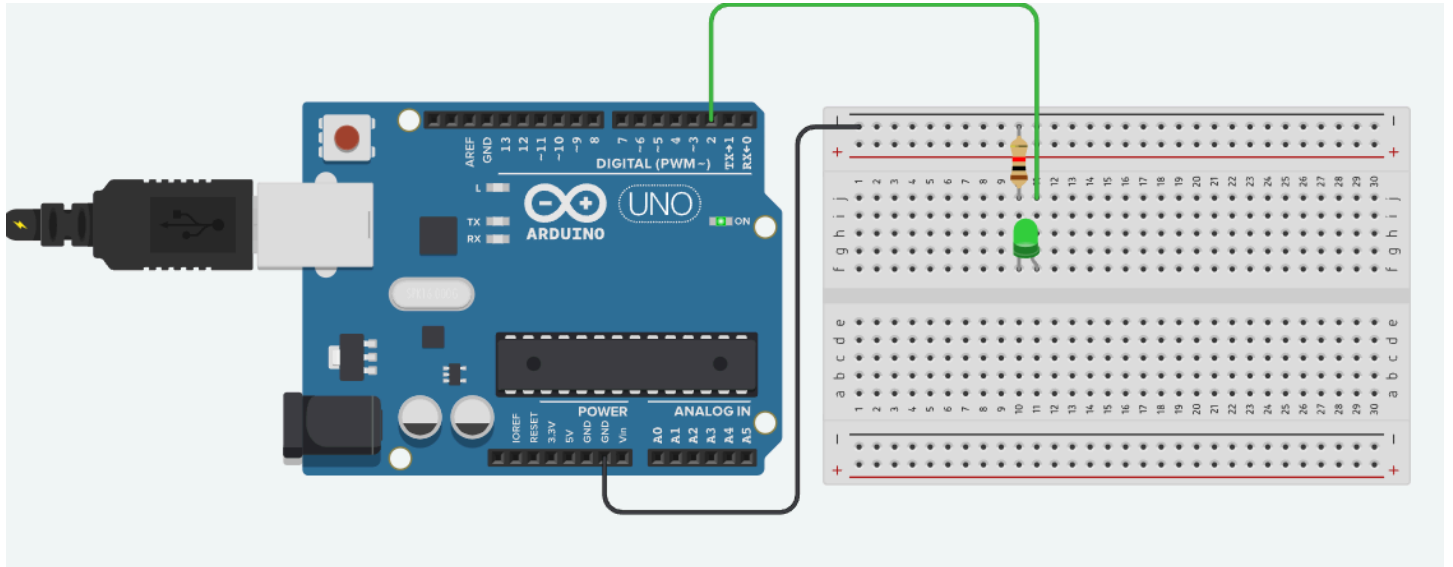
Nome: Gabriel Henrique Vieira de Oliveira

Data: 06/04/2025

Matéria: Laboratório de Introdução à Engenharia da Computação

RELATÓRIO 6

Prática 1 - Pisca-Pisca: Na primeira prática, iremos controlar um led pelo Esp32 (Arduino no simulador Tinkercad).



Exercícios:

1. Teste o código acima e explique o que cada comando faz

R: No código acima, declaramos uma variável constante led1, que possui seu valor 2, indicando que essa porta está ligada a uma das pernas do led e será responsável por dar o OUTPUT, como descrito na primeira função do código. Em seguida, temos a presença de um loop que só vai terminar quando o arduino desligar. Nesse loop, o led recebe uma informação virtual para ligar (DigitalWrite(led1, 1)), em seguida ele vai ter um delay e ficar ligado por 1 segundo (delay(1000)), depois vai receber outra informação digital para apagar (DigitalWrite(led1, 0)) e por fim vai ficar apagado também por 1 segundo (delay(1000)). Tudo isso acontece em loop enquanto o programa estiver rodando, originando então, um pisca-pisca.

2. Altere para que o led pisque de 5 em 5 segundos.

```
const int led1 = 2;

void setup()
{
  pinMode(led1, OUTPUT);
}

void loop()
{
  digitalWrite(led1, 1);
  delay(5000);
  digitalWrite(led1, 0);
  delay(5000);
}
```

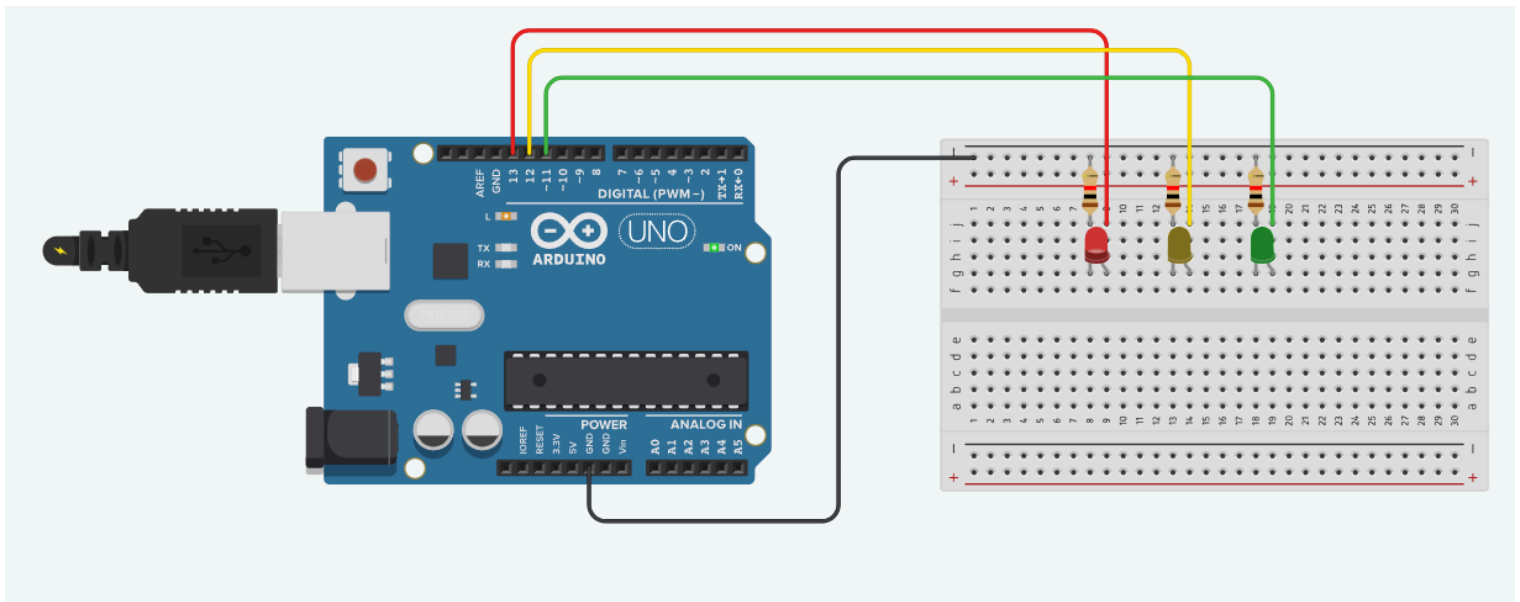
3. Altere para ficar ligado por 10 segundos e desligado por 5 segundos.

```
const int led1 = 2;

void setup()
{
  pinMode(led1, OUTPUT);
}

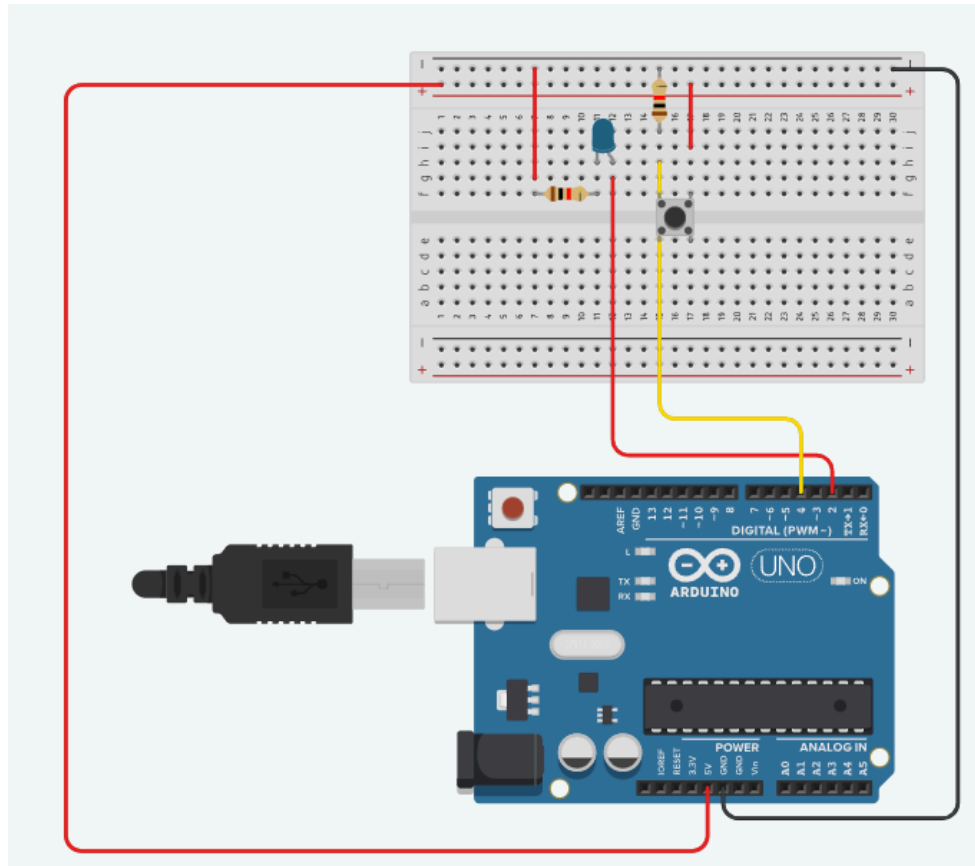
void loop()
{
  digitalWrite(led1, 1);
  delay(10000);
  digitalWrite(led1, 0);
  delay(5000);
}
```

4. Monte um novo circuito com 3 leds (1 verde, 1 amarelo e 1 vermelho). Simule o funcionamento de um semáforo, sendo que o led verde fica ligado por 4 segundos, amarelo por 1 segundo e vermelho por 5 segundos).



```
1 int led11 = 11; // LED VERDE
2 int led12 = 12; // LED AMARELO
3 int led13 = 13; // LED VERMELHO
4
5 void setup(){
6   // configura os pinos como saidas digitais
7   pinMode(led11,OUTPUT);
8   pinMode(led12,OUTPUT);
9   pinMode(led13,OUTPUT);
10 }
11
12 void loop(){
13
14   // Vermelho fica 3 segundos piscando
15   digitalWrite(led13, HIGH); // nivel logico alto
16   delay(3500);
17   digitalWrite(led13,LOW); // nivel logico baixo
18   delay(1000);
19
20   // Amarelo fica 2 segundos piscando
21   digitalWrite(led12,HIGH); // nivel logico alto
22   delay(2400);
23   digitalWrite(led12,LOW); // nivel logico baixo
24   delay(1000);
25
26   // Verde fica 4 segundos piscando
27   digitalWrite(led11,HIGH); // nivel logico alto
28   delay(4500);
29   digitalWrite(led11,LOW); // nivel logico baixo
30   delay(1000);
31 }
32
33
```

Prática 2 - Pisca Pisca com Botão: Na segunda prática, iremos controlar um led e um botão pelo Esp32 (ou Arduino).



Exercícios:

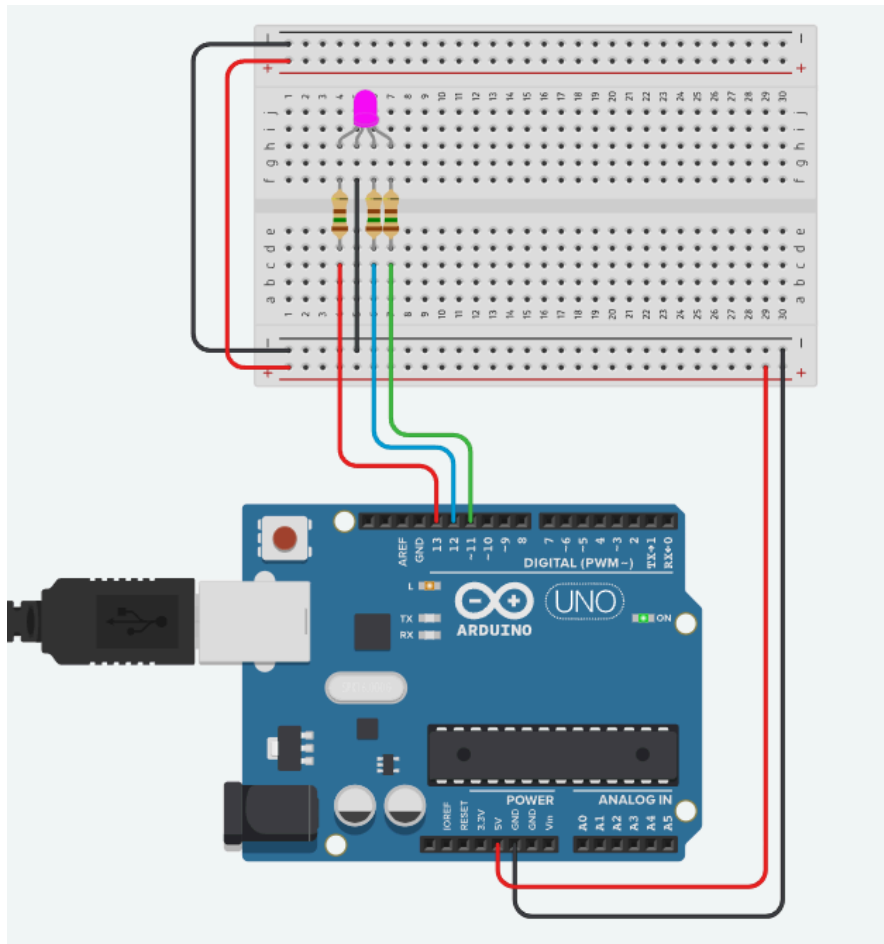
5. Teste o código acima e explique o que cada comando faz.

R: Diferentemente do anterior, neste temos a presença de um botão para controlar a piscada do LED. Declaramos duas variáveis constantes alocadas a suas devidas portas no arduino, só que com um porém: enquanto o LED e sua porta continuam sendo um OUTPUT, o botão agora é um INPUT, ou seja, vai transmitir informação para o circuito. Também tem uma variável padrão para determinar o estado do botão durante o processamento do código. Começamos um loop incluindo um if em que, caso o botão seja apertado, tudo que está codificado dentro dele acontecerá, que no caso é se apertarmos o botão, o LED acende, e se deixarmos de apertar esse botão, o led apaga. Tudo isso enquanto o arduino estiver ligado.

6. Criar um programa que quando pressionarmos o botão, um led deverá piscar 5 vezes e de meio em meio segundo.

```
1  const int led = 2;
2  const int botao = 4;
3  int estadoBotao;
4
5  void setup()
6  {
7    pinMode(led, OUTPUT);
8    pinMode(botao, INPUT);
9  }
10
11 void loop()
12 {
13   estadoBotao = digitalRead(botao);
14
15   if (estadoBotao == 1){
16     digitalWrite(led, 1);
17     delay(500);
18     digitalWrite(led, 0);
19     delay(500);
20     digitalWrite(led, 1);
21     delay(500);
22     digitalWrite(led, 0);
23     delay(500);
24     digitalWrite(led, 1);
25     delay(500);
26     digitalWrite(led, 0);
27     delay(500);
28     digitalWrite(led, 1);
29     delay(500);
30     digitalWrite(led, 0);
31     delay(500);
32     digitalWrite(led, 1);
33     delay(500);
34     digitalWrite(led, 0);
35     delay(500);
36     digitalWrite(led, 1);
37     digitalWrite(led, 0);
38   }
39 }
```

Prática 3 - LED RGB: Na terceira prática, iremos controlar um led RGB pelo Esp32 (ou Arduino).



Exercícios:

7. Crie um programa que mostre cor a cor do led RGB (mostrando todas as cores) e altere cada cor a cada 1 segundo.

```
1 const int vermelho = 13;
2 const int azul = 12;
3 const int verde = 11;
4
5 void setup(){
6   pinMode(verde, OUTPUT);
7   pinMode(azul, OUTPUT);
8   pinMode(vermelho, OUTPUT);
9 }
10
11 void loop(){
12   digitalWrite(vermelho, 1);
13   digitalWrite(azul, 0);
14   digitalWrite(verde, 0);
15   delay(1400);
16
17   digitalWrite(vermelho, 0);
18   digitalWrite(azul, 1);
19   digitalWrite(verde, 0);
20   delay(1400);
21
22   digitalWrite(vermelho, 0);
23   digitalWrite(azul, 0);
24   digitalWrite(verde, 1);
25   delay(1400);
26
27   digitalWrite(vermelho, 0);
28   digitalWrite(azul, 1);
29   digitalWrite(verde, 1);
30   delay(1400);
31
32   digitalWrite(vermelho, 1);
33   digitalWrite(azul, 0);
34   digitalWrite(verde, 1);
35   delay(1400);
36
37   digitalWrite(vermelho, 1);
38   digitalWrite(azul, 1);
39   digitalWrite(verde, 0);
40   delay(1400);
41
42   digitalWrite(vermelho, 1);
43   digitalWrite(azul, 1);
44   digitalWrite(verde, 1);
45   delay(1400);
46 }
```

8. Utilize o led RGB para simular o funcionamento de um semáforo, sendo que o led verde fica ligado por 5 segundos, amarelo por 1 segundo e vermelho por 3 segundos).

```
1  const int vermelho = 13;
2  const int azul = 12;
3  const int verde = 11;
4
5  void setup(){
6    pinMode(verde, OUTPUT);
7    pinMode(azul, OUTPUT);
8    pinMode(vermelho, OUTPUT);
9  }
10
11 // SIMULANDO UM SEMÁFORO
12 void loop(){
13   // Verde fica ligado por 5 segundos
14   digitalWrite(vermelho, 0);
15   digitalWrite(azul, 0);
16   digitalWrite(verde, 1);
17   delay(5400);
18
19   // Amarelo fica ligado por 1 segundo
20   digitalWrite(vermelho, 1);
21   digitalWrite(azul, 0);
22   digitalWrite(verde, 1);
23   delay(1400);
24
25   // Vermelho fica ligado por 3 segundos
26   digitalWrite(vermelho, 1);
27   digitalWrite(azul, 0);
28   digitalWrite(verde, 0);
29   delay(3400);
30 }
```