



SEMAFORO
INTELIGENTE

AGENDA

- 01** **QUEM SOMOS**
- 02** **SEMÁFORO BÁSICO**
- 03** **PROGRAMA**
- 04** **SEMÁFORO COM BOTÃO DE PEDESTRE**
- 05** **PROGRAMA**
- 06** **SENSOR ULTRASSONICO**
- 07** **PROGRAMA**



Start Dev

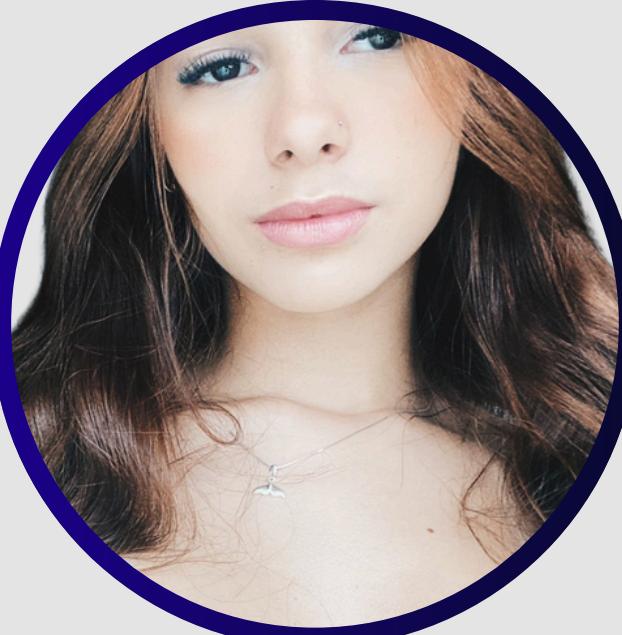
Uma empresa de Desenvolvimento (T.I)
focada em iniciar o seu sistema e
fomos encarregados desse projeto.



Apresentação dos desenvolvedores



Eduardo Zanetti
Desenvolvedor



Nicoly Ribeiro
Scrum master



Rafael Brecci
Desenvolvedor

Mariana Ribeiro
Desenvolvedor



Nicolas Fernandes
Desenvolvedor



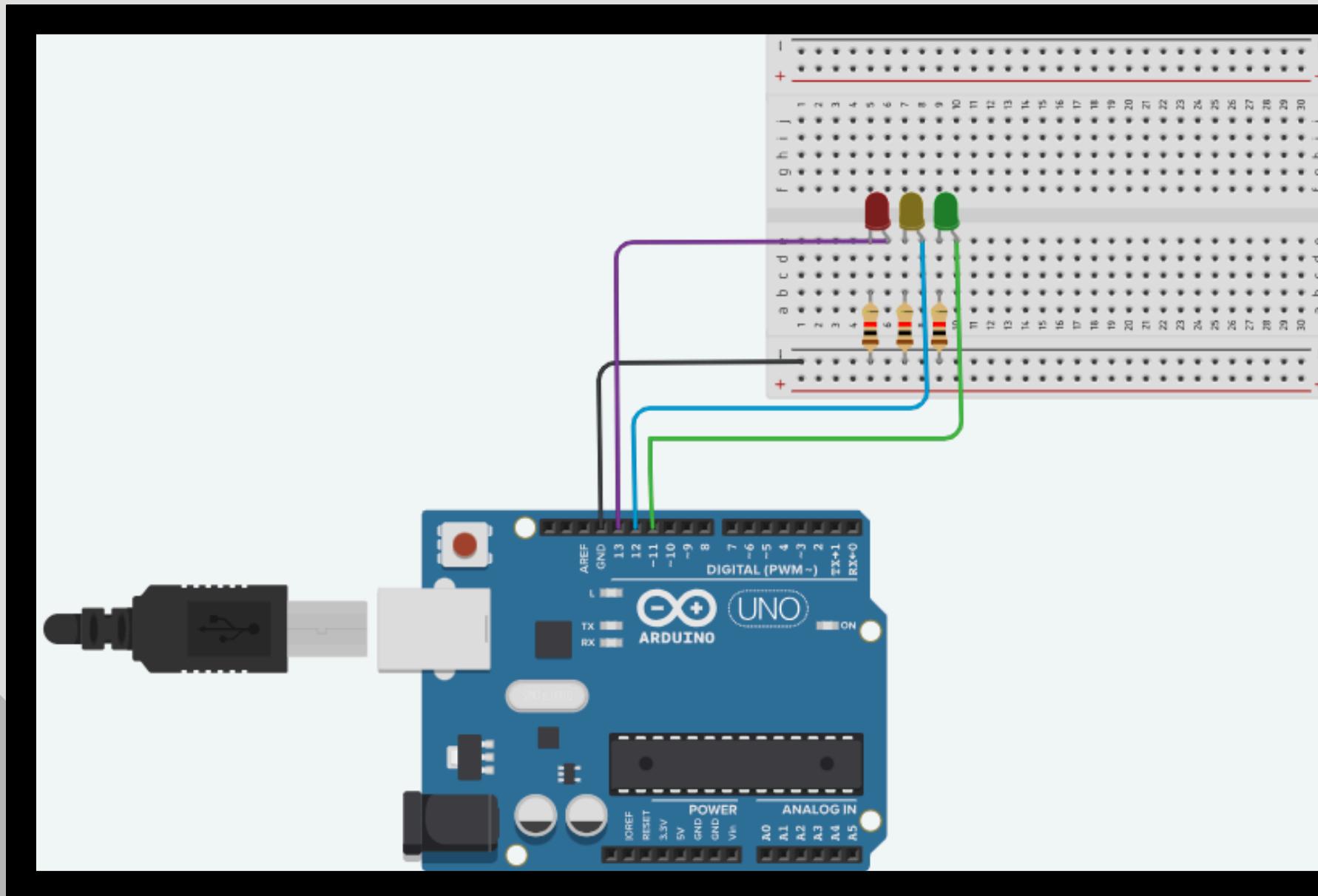
Semáforo básico

Sequência de um semáforo
com LEDs vermelho, amarelo e verde.

Componentes: Arduino Uno R3, LED
vermelho, LED amarelo, LED verde, 3
resistores (220 ohms cada).



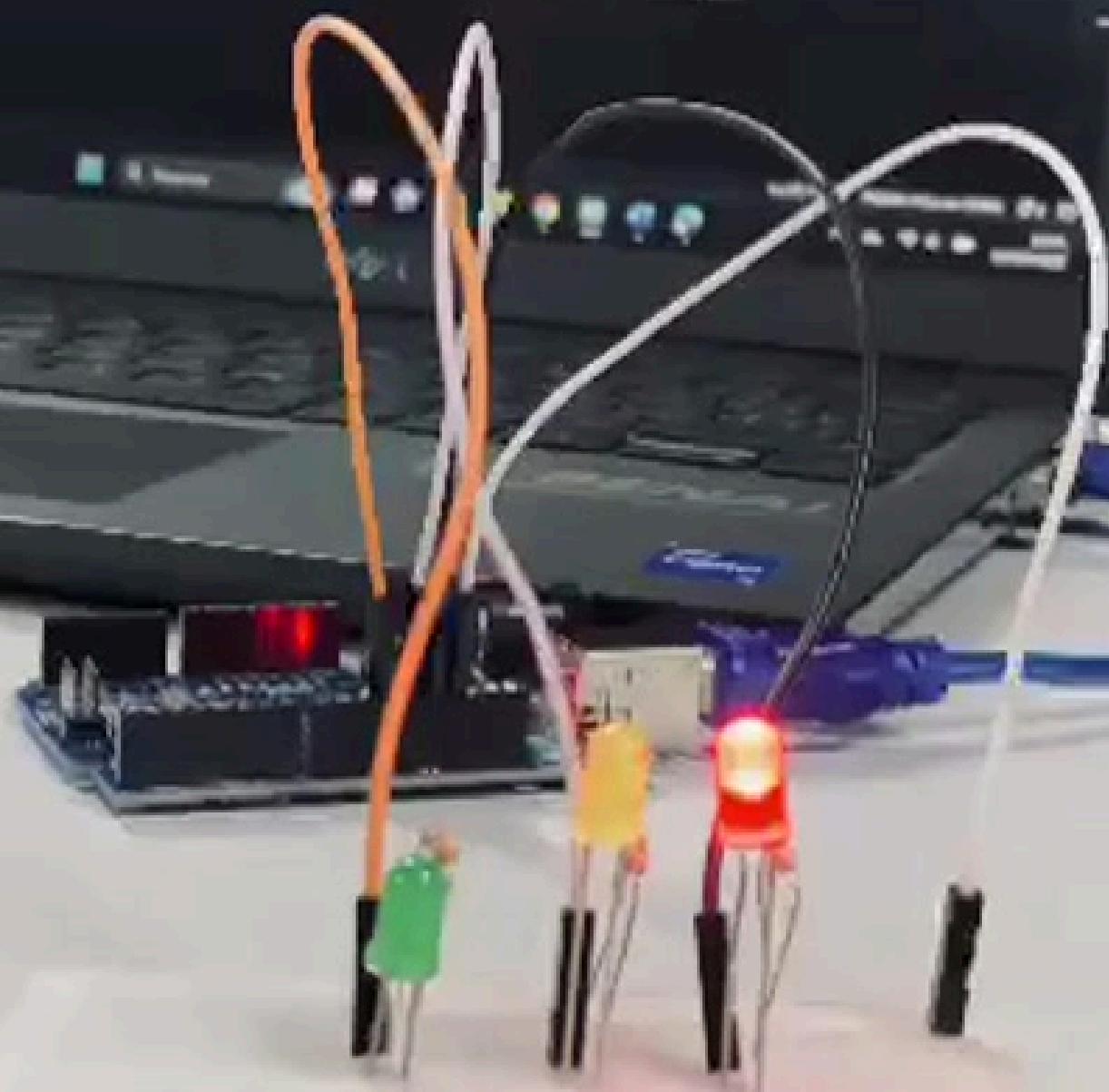
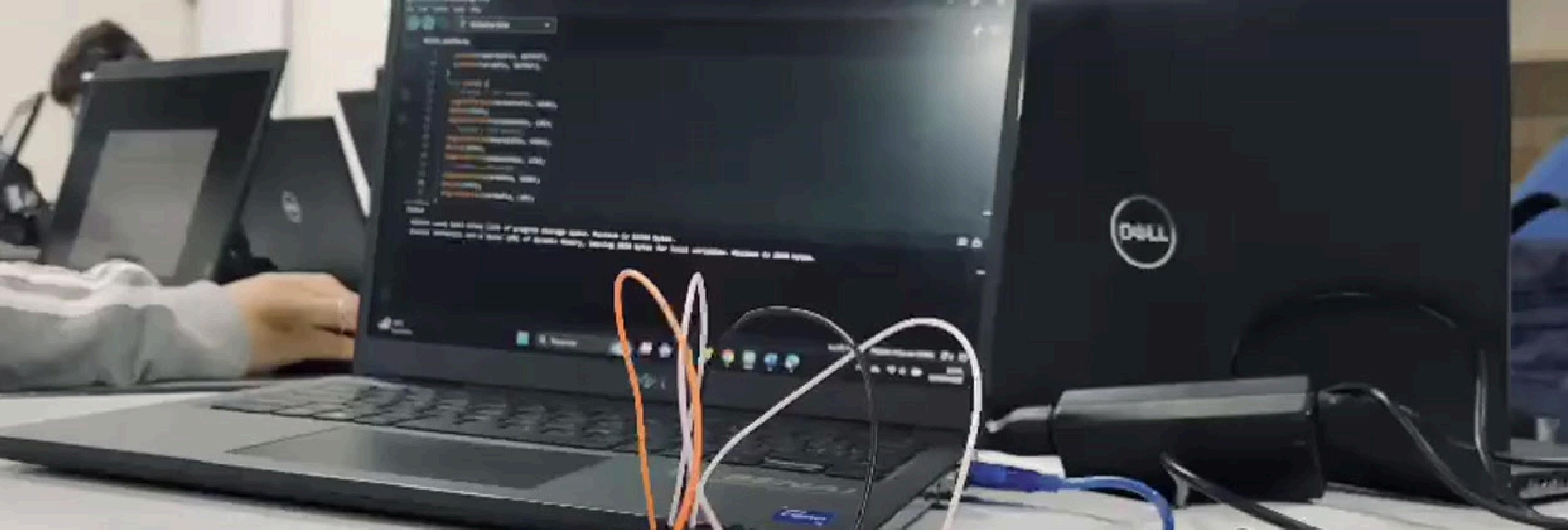
Tinkercad: Imagem e código



```
const int vermelhoPin = 13;
const int amareloPin = 12;
const int verdePin = 11;
const int tempoVermelho = 5000; // 5 segundos
const int tempoAmarelo = 2000; // 2 segundos
const int tempoVerde = 5000; // 5 segundos

void setup() {
    pinMode(vermelhoPin, OUTPUT);
    pinMode(amareloPin, OUTPUT);
    pinMode(verdePin, OUTPUT);
}

void loop() {
    digitalWrite(vermelhoPin, HIGH);
    delay(tempoVermelho);
    digitalWrite(vermelhoPin, LOW);
    digitalWrite(amareloPin, HIGH);
    delay(tempoAmarelo);
    digitalWrite(amareloPin, LOW);
    digitalWrite(verdePin, HIGH);
    delay(tempoVerde);
    digitalWrite(verdePin, LOW);
}
```



Semáforo com botão de pedestre

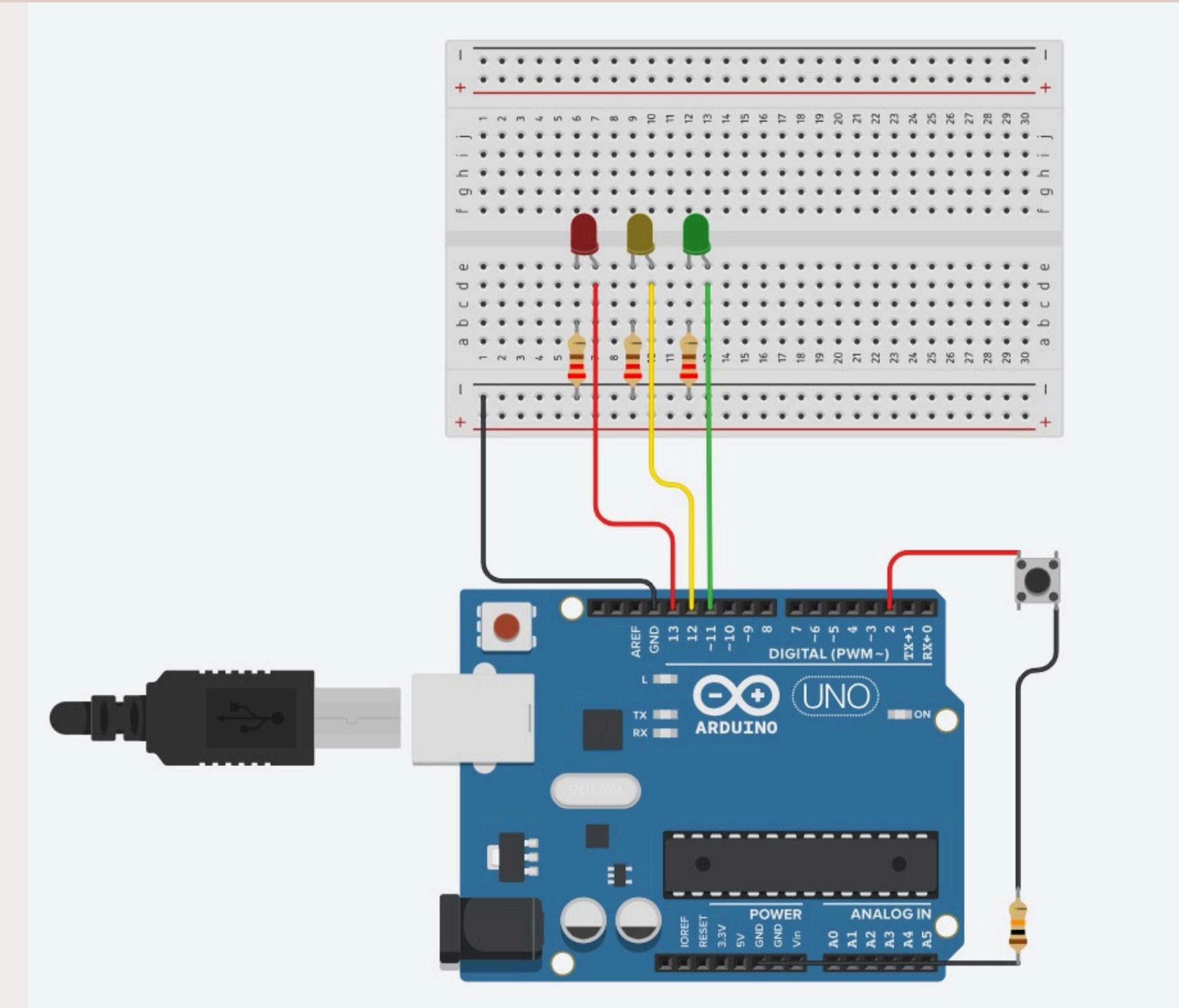
Botão para pedestres que altera o ciclo do semáforo.

Componentes: Arduino Uno R3, LED vermelho, LED amarelo, LED verde, 3 resistores (220 ohms), Botão de pressão, Resistor de pull-up ou pull-down (10k ohms).





Tinkercad: Imagem





Tinkercad: Código

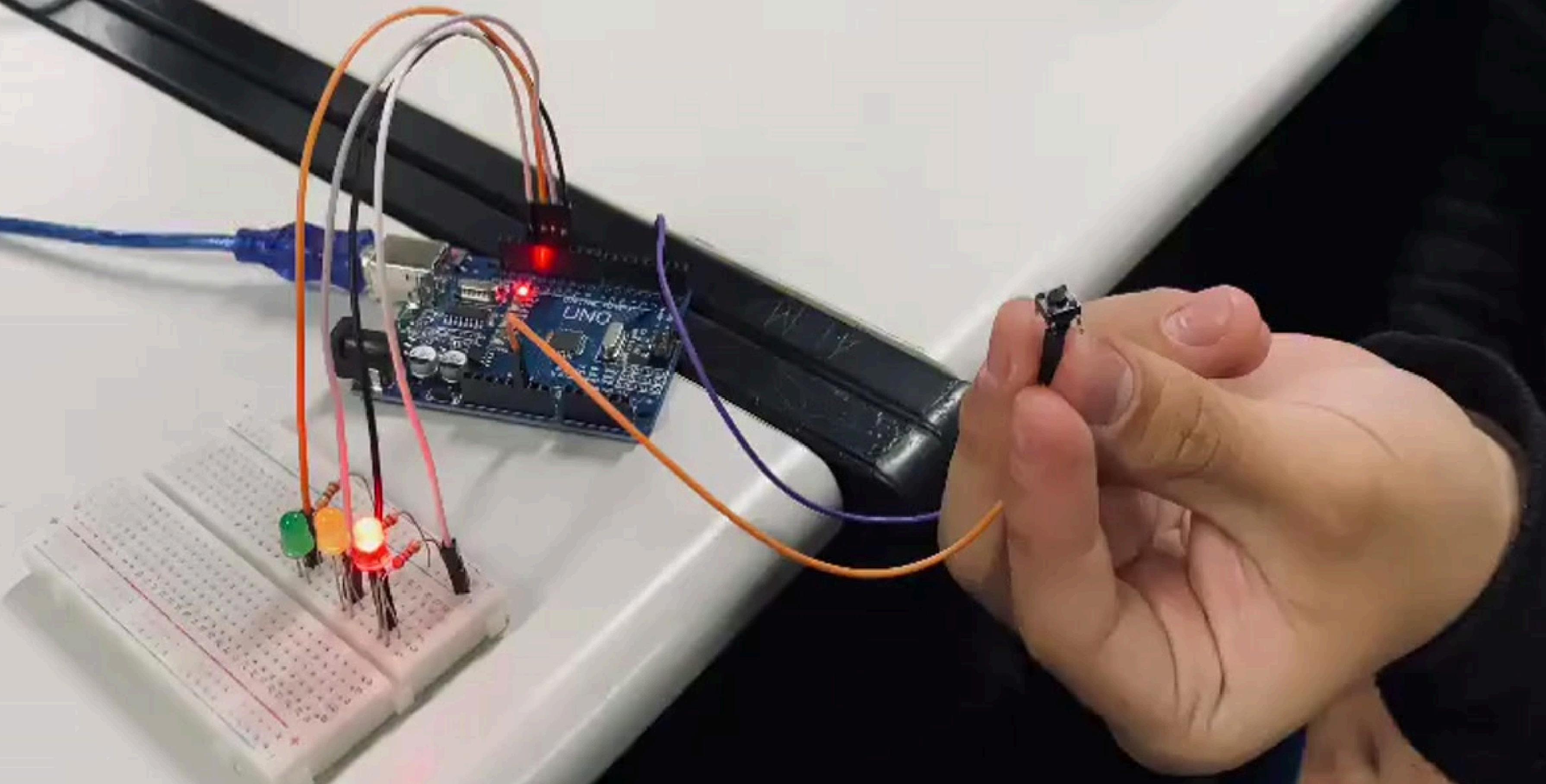
```
const int vermelhoPin = 13;
const int amareloPin = 12;
const int verdePin = 11;
const int botaoPin = 2;
volatile bool pedidoPedestre = false;

void setup() {
pinMode(vermelhoPin, OUTPUT);
pinMode(amareloPin, OUTPUT);
pinMode(verdePin, OUTPUT);
pinMode(botaoPin, INPUT_PULLUP); // Usando resistor de
pull-up interno
attachInterrupt(digitalPinToInterrupt(botaoPin),
botaoPressionado, FALLING); // Interrupção na queda (quando
pressionado)
// Inicialmente o semáforo para carros está verde
digitalWrite(verdePin, HIGH);
}

void loop() {
if (!pedidoPedestre) {
// Sequência normal do semáforo para carros
// ... (implemente a lógica como no Exercício 4) ...
} else {
// Sequência para pedestres
digitalWrite(verdePin, LOW);
digitalWrite(amareloPin, HIGH);
```

```
delay(2000);
digitalWrite(amareloPin, LOW);
digitalWrite(vermelhoPin, HIGH);
delay(5000);
// Sinaliza verde para pedestres (se houver um LED para
pedestres)
delay(5000); // Tempo para pedestres atravessarem
digitalWrite(vermelhoPin, LOW);
pedidoPedestre = false; // Reseta o pedido
digitalWrite(verdePin, HIGH); // Volta ao verde para
carros
}
delay(100); // Pequeno delay
}

void botaoPressionado() {
pedidoPedestre = true;
}
```



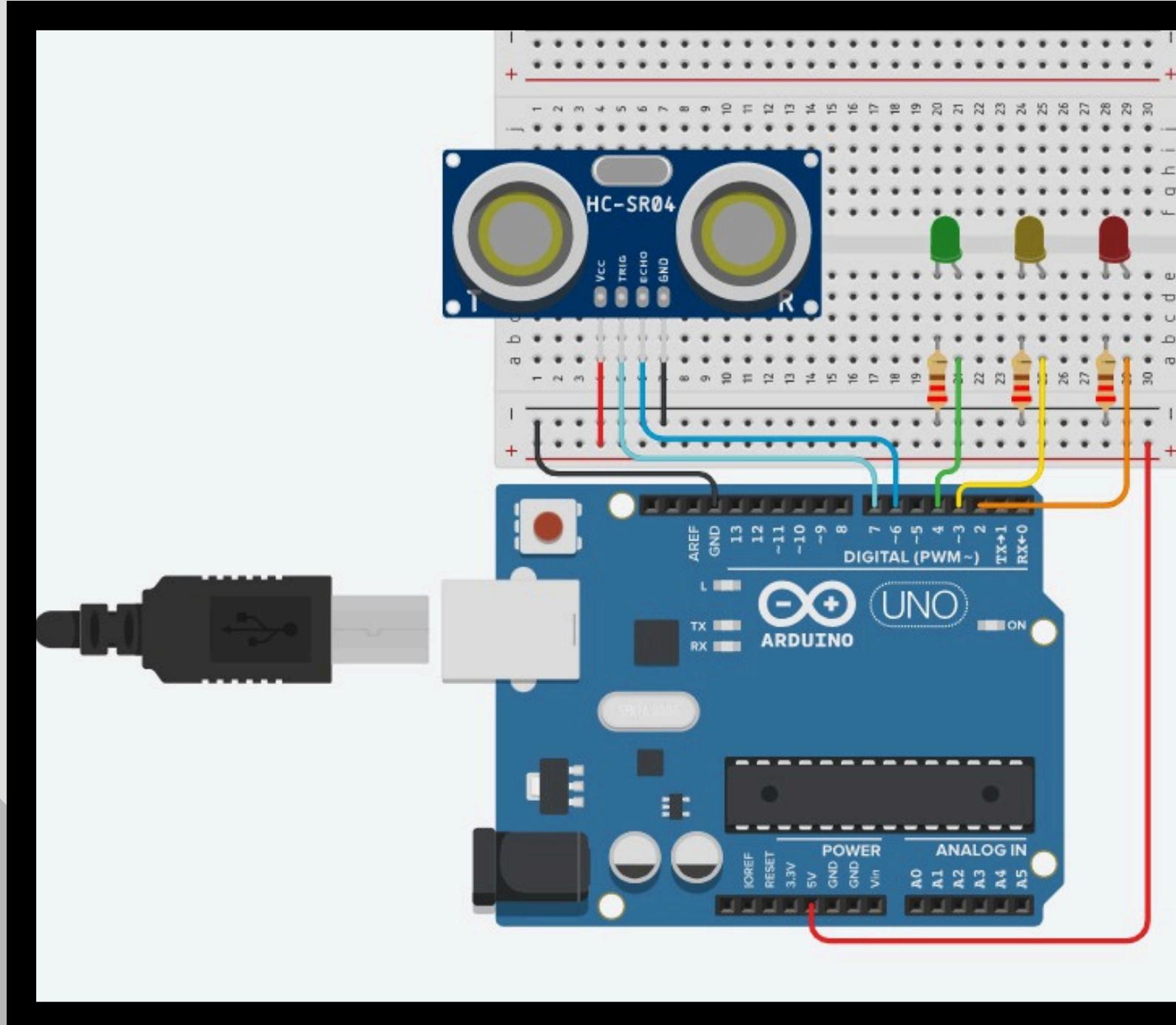
Sensor ultrassônico para ajuste automático do semáforo

Sensor de luz para ajustar o tempo das cores do semáforo de acordo com a luminosidade.

Componentes: Arduino Uno R3, LED vermelho, LED amarelo, LED verde, 3 resistores (220 ohms), Sensor ultrassônico (HC-SR04), Resistor (10k ohms).



Tinkercad: Imagem e código

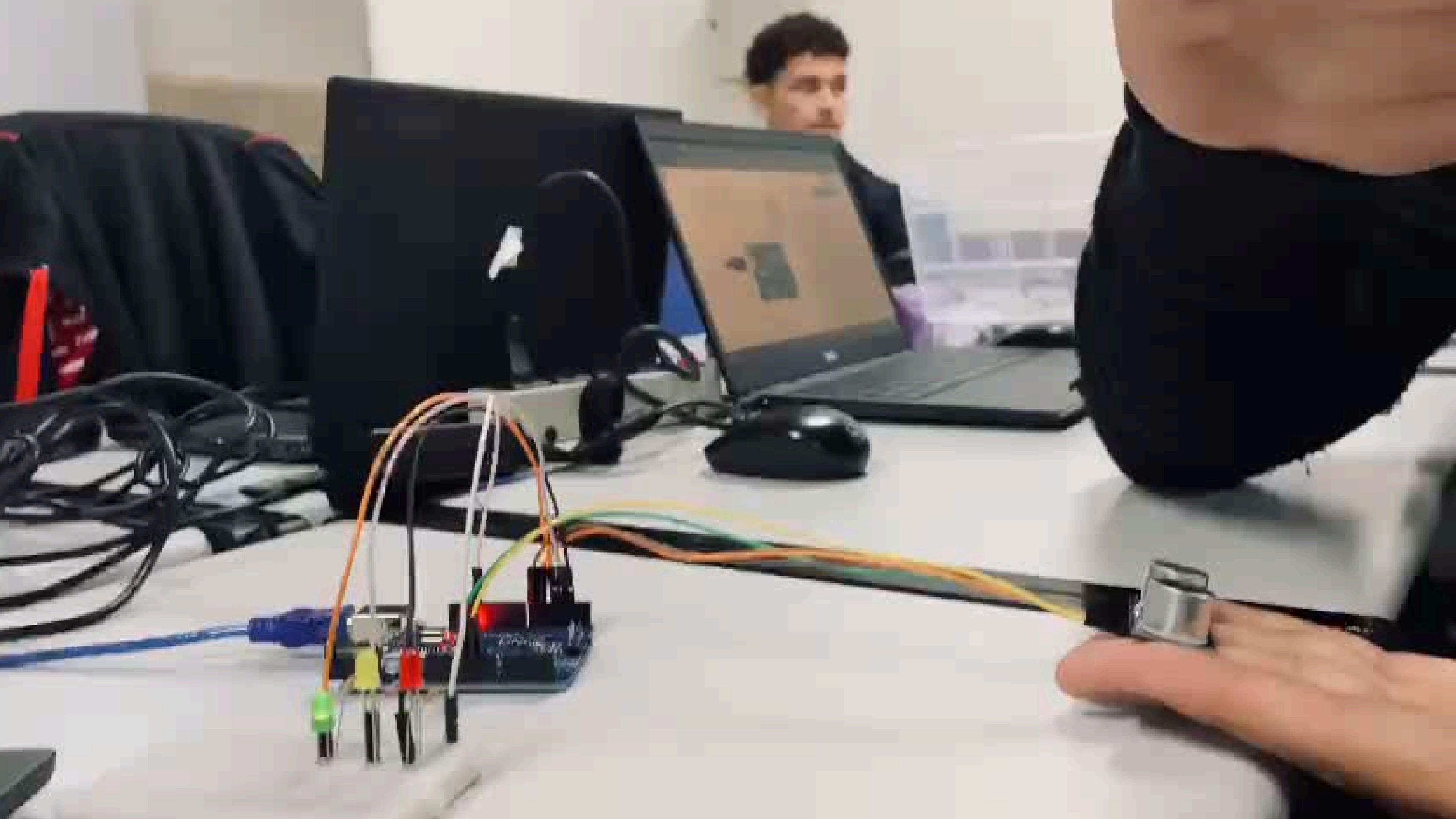


```
const int vermelhoPin = 13;
const int amareloPin = 12;
const int verdePin = 11;
const int sensorLuzPin = A0;

void setup() {
  pinMode(vermelhoPin, OUTPUT);
  pinMode(amareloPin, OUTPUT);
  pinMode(verdePin, OUTPUT);
}

void loop() {
  int luminosidade = analogRead(sensorLuzPin); // Lê o valor
  do sensor de luz (0-1023)
  int tempoVerde = map(luminosidade, 0, 1023, 10000, 2000);
  // Mapeia a luminosidade para o tempo do verde (quanto mais
  claro, menos tempo)

  digitalWrite(vermelhoPin, HIGH);
  delay(5000);
  digitalWrite(vermelhoPin, LOW);
  digitalWrite(amareloPin, HIGH);
  delay(2000);
  digitalWrite(amareloPin, LOW);
  digitalWrite(verdePin, HIGH);
  delay(tempoVerde);
  digitalWrite(verdePin, LOW);
}
```





Obrigada pela atenção!!