



Semáforo de 2 quadras e 4 sinais por quadra, com acessibilidade.

Alunos: João Victor,
Débora Santos,
Felizberto Macedo,
Christian Williams,
João Vitor do Nascimento



Sumário

1. Explicando o projeto;
2. Matérias utilizados;
3. Imagens;
4. Código I;
5. Código II;
6. Referência.

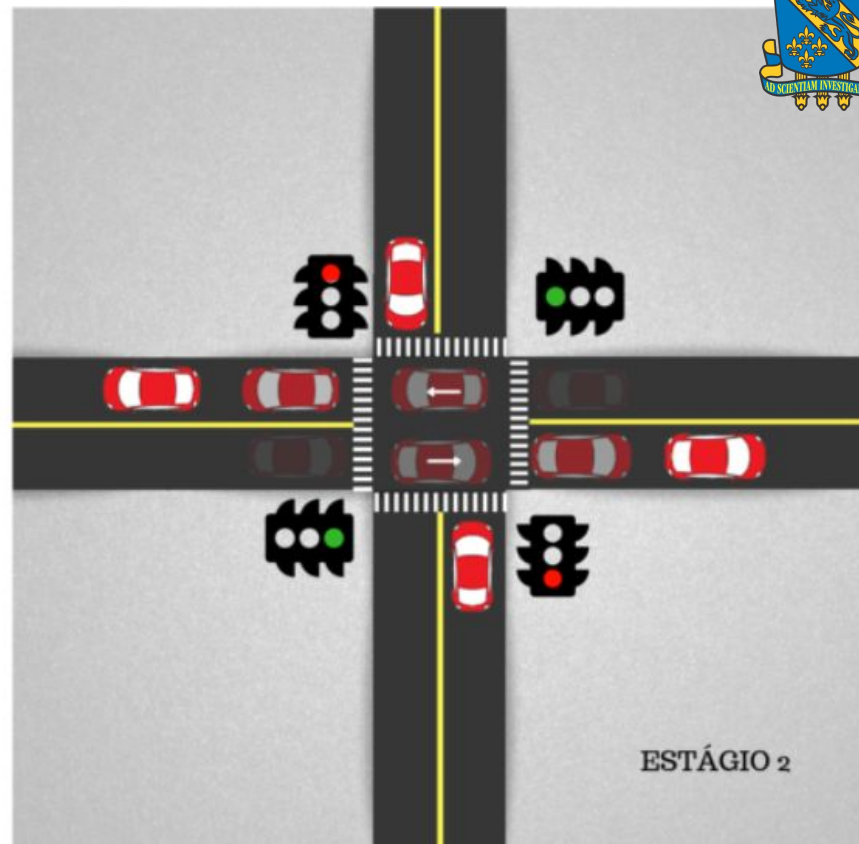
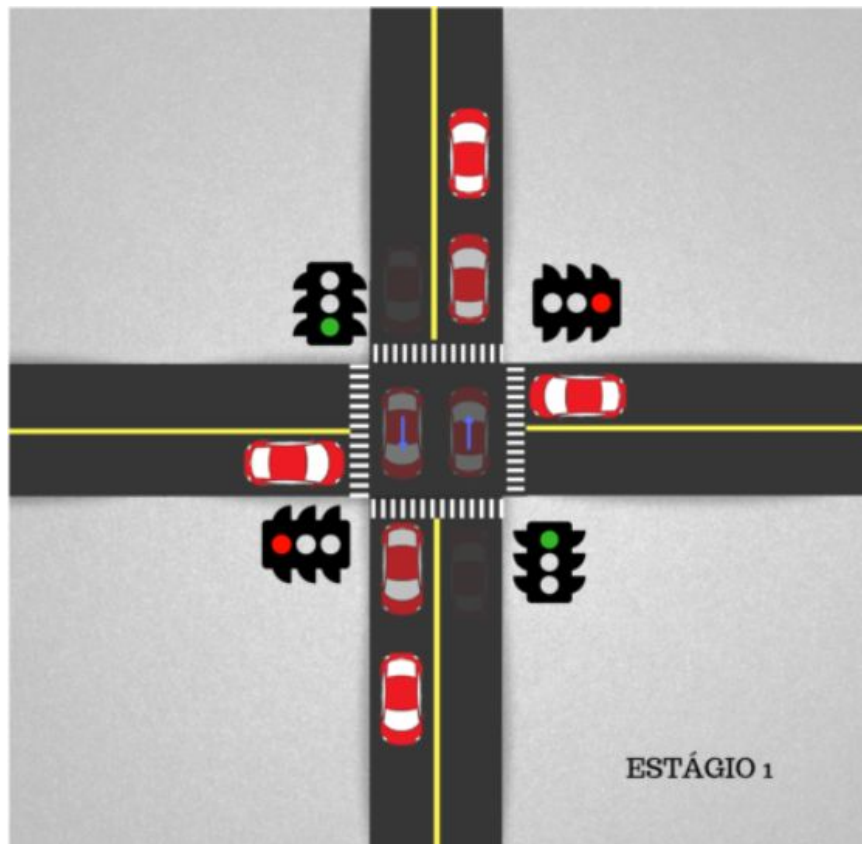


1. Explicando o projeto

- É um sistema de controle de tráfego que gerencia veículos e pedestres em cruzamentos;
- Ele inclui sinais de trânsito para veículos e pedestres;
- Além de recursos acessíveis como sinais sonoros;
- Os semáforos devem ser sincronizados e receber manutenção regular para garantir sua eficácia na segurança e no fluxo do tráfego.

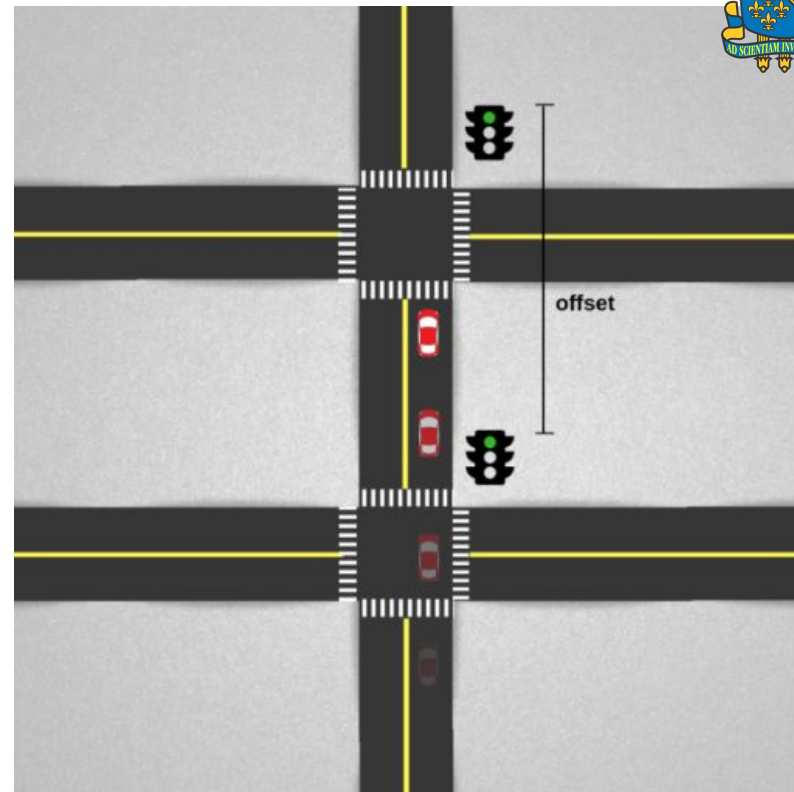


- **Ciclo:** é o tempo em que a sequência verde, amarelo e vermelho aparece por completo.
- Ciclo total de operação, dura 20 segundos;
- O tempo do sinal Amarelo é de 3 segundos, do Verde 7 segundos e do Vermelho 10 segundos;





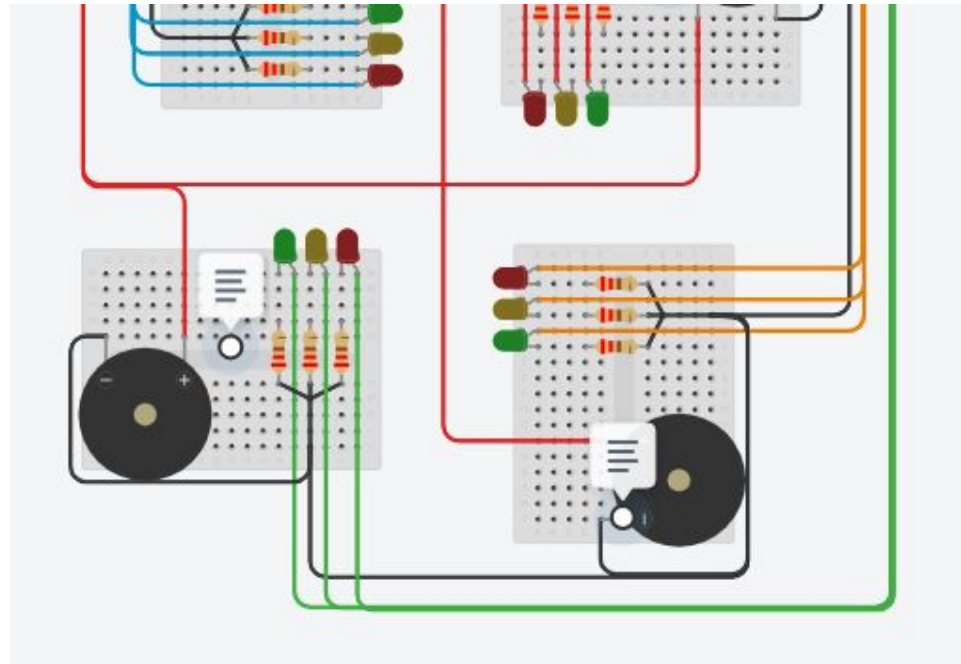
Offset: também conhecido como defasagem, é a diferença, em segundos, entre os verdes de dois semáforos consecutivos de uma mesma via e está relacionado com o sincronismo dos semáforos.

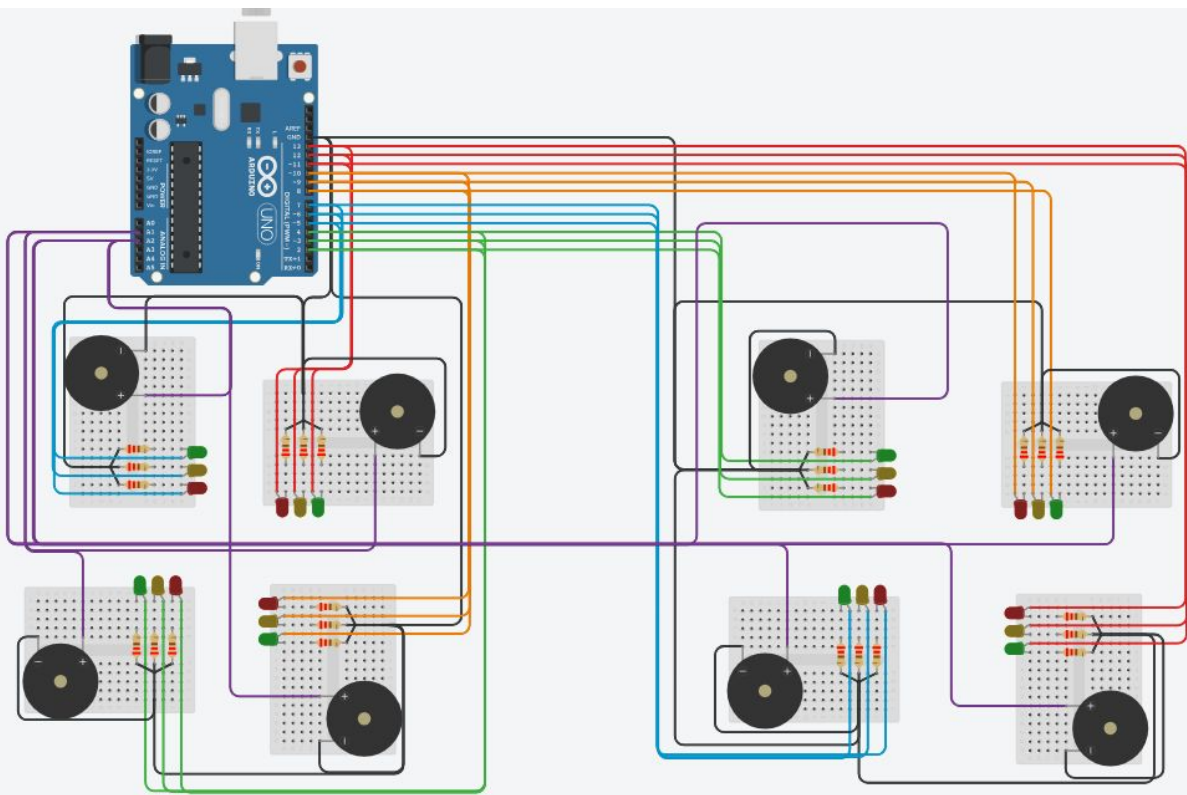




2. Materiais utilizados;

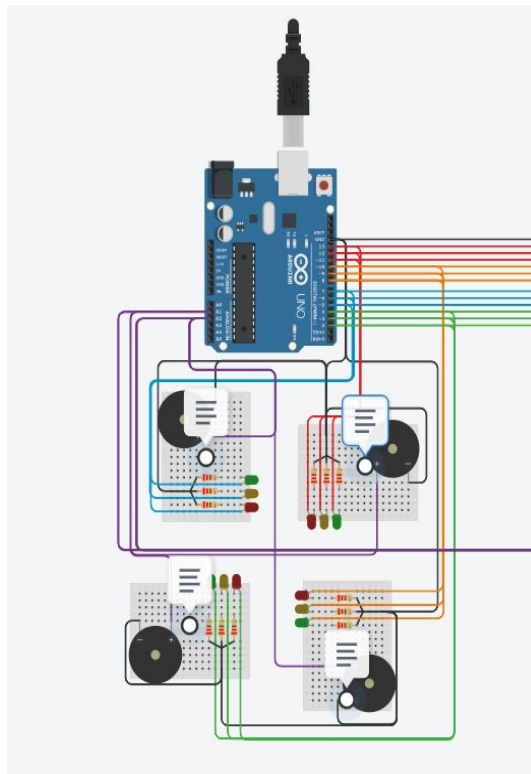
- Arduino Uno R3;
- 4 Placas de ensaio pequenas;
- 4 Led's Amarelos;
- 4 Led's Verdes;
- 4 Led's Vermelhos;
- 12 Resistores de 300 ohms;
- 4x Piezo;







4. Código I



```
//Semaforo 1 quadra 1
#define red1 13
#define yellow1 12
#define green1 11

//Semaforo 2 quadra 1
#define red2 10
#define yellow2 9
#define green2 8

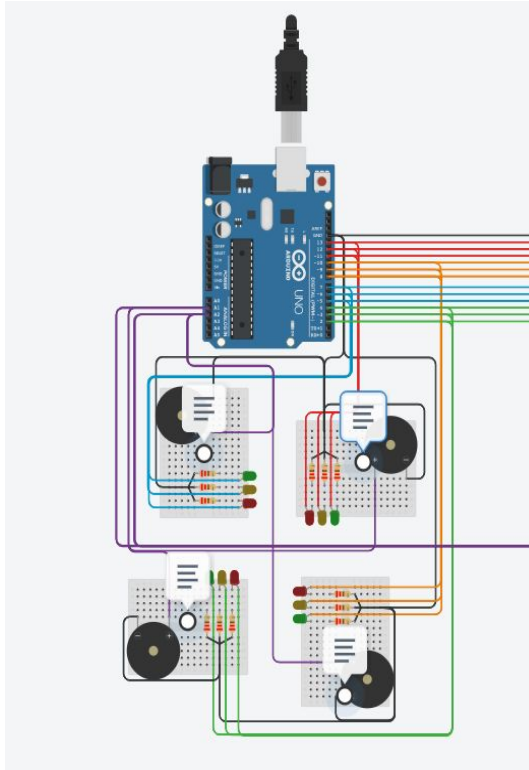
//Semaforo 3 quadra 1
#define red3 5
#define yellow3 6
#define green3 7

//Semaforo 4 quadra 1
#define red4 2
#define yellow4 3
#define green4 4

#define buzzer1 A1 //SEMAPHORO 1 & 4
#define buzzer2 A2 //SEMAPHORO 2 & 3
```



4. Código I



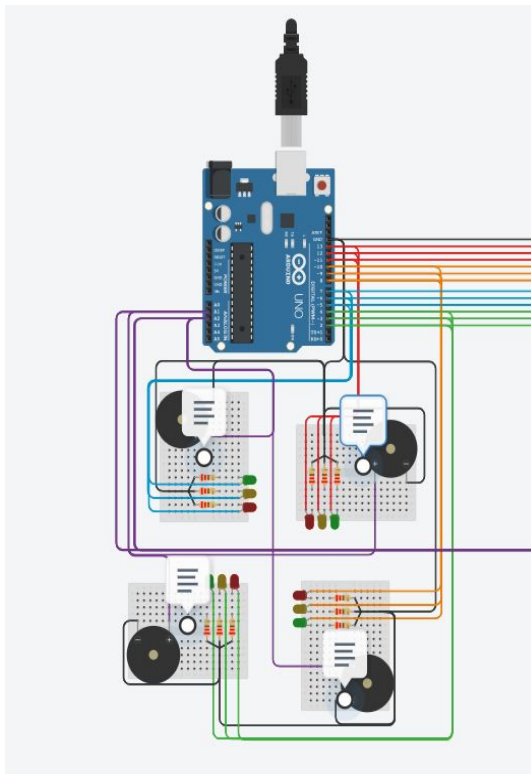
```
void setup()
{
  short leds[12] = {
    red1,
    red2,
    red3,
    red4,
    yellow1,
    yellow2,
    yellow3,
    yellow4,
    green1,
    green2,
    green3,
    green4
  };

  pinMode(buzzer1, OUTPUT);
  pinMode(buzzer2, OUTPUT);

  noTone(buzzer1);
  noTone(buzzer2);

  for(int index = 0; index < 12; index++)
    pinMode(leds[index], OUTPUT);
}
```

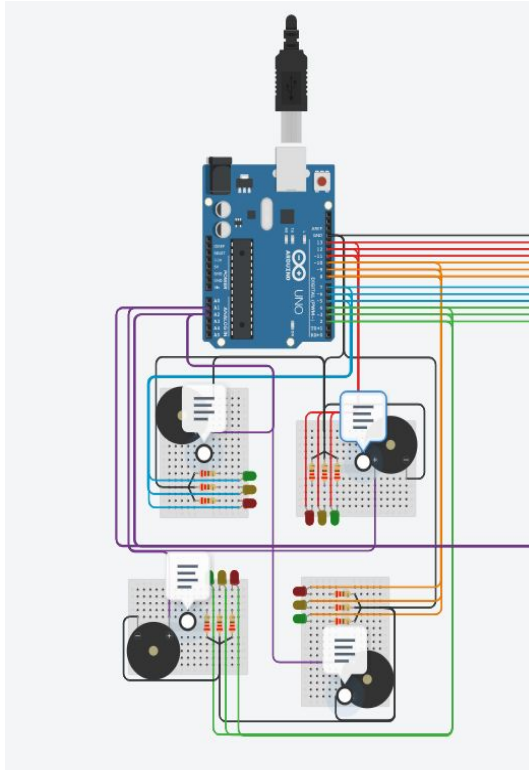
4. Código I



```
void loop()  
{  
  semaphoro1C1();  
  clear();  
  semaphoro2C1();  
  clear();  
}
```



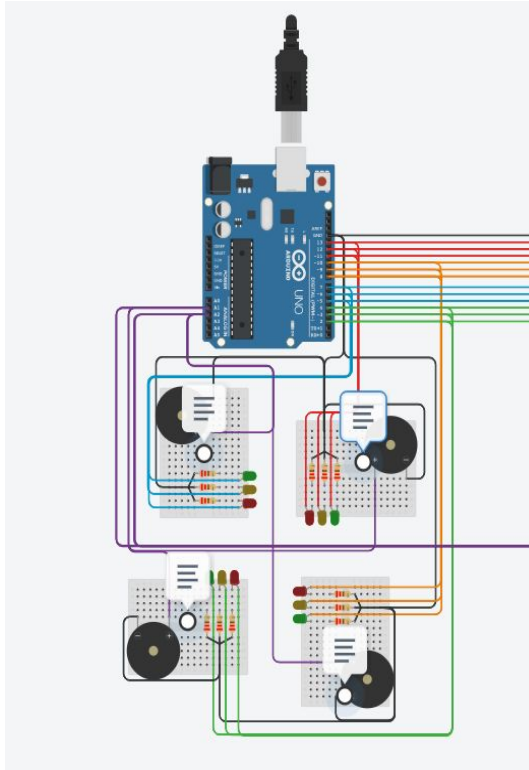
4. Código I



```
int semaphoreIC1() {
  for (int index = 11; index <= 13; index++){
    digitalWrite(index, HIGH);
    if (digitalRead(green1) == HIGH) {
      digitalWrite(red2, HIGH);
      digitalWrite(red3, HIGH);
      digitalWrite(green4, HIGH);
      sinalSound(buzzer1, 10, 200, 500);
    } else if (digitalRead(yellow1) == HIGH){
      digitalWrite(red3, HIGH);
      digitalWrite(red2, HIGH);
      digitalWrite(green4, LOW);
      digitalWrite(yellow4, HIGH);
      sinalSound(buzzer1, 7, 200, 200);
    } else {
      digitalWrite(yellow4, LOW);
      digitalWrite(red4, HIGH);
      noTone(buzzer1);
      noTone(buzzer2);
      return 0;
    }
    digitalWrite(index, LOW);
  }
}
```



4. Código I



```
int semaphoro2C1() {
  for (int index = 8; index <= 10; index++){
    digitalWrite(index, HIGH);
    if (digitalRead(green2) == HIGH) {
      digitalWrite(red1, HIGH);
      digitalWrite(red4, HIGH);
      digitalWrite(green3, HIGH);
      sinalSound(buzzer2, 10, 200, 500);
    } else if (digitalRead(yellow2) == HIGH){
      digitalWrite(red4, HIGH);
      digitalWrite(red1, HIGH);
      digitalWrite(green3, LOW);
      digitalWrite(yellow3, HIGH);
      sinalSound(buzzer2, 7, 200, 200);
    } else {
      digitalWrite(yellow3, LOW);
      digitalWrite(red3, HIGH);
      noTone(buzzer1);
      noTone(buzzer2);
      return 0;
    }
    digitalWrite(index, LOW);
  }
}
```



4. Código I

```
void clear() {  
    for (int index = 2; index <= 13; index++) {  
        digitalWrite(index, LOW);  
    }  
}
```




4. Código I



```
void sinalSound(int buzzer, int times, int delay1, int delay2) {  
    for (int count = 0; count <= times; count++){  
        tone(buzzer, 2000);  
        delay(delay1);  
        noTone(buzzer);  
        delay(delay2);  
    }  
}
```




5. Código II

```
#define red1 13
#define yellow1 12
#define green1 11

#define red2 10
#define yellow2 9
#define green2 8

#define red3 5
#define yellow3 6
#define green3 7

#define red4 2
#define yellow4 3
#define green4 4

#define buzzer1 A1
#define buzzer2 A2

#define button1 A5
#define button2 A0
#define button3 A3
#define button4 A4
```



5. Código II

```
int btn1State;  
int btn2State;  
int btn3State;  
int btn4State;  
  
unsigned long previousMillis = 0;  
unsigned long interval = 0;  
  
enum TrafficLightState {  
    GREEN1,  
    YELLOW1,  
    RED1,  
    GREEN2,  
    YELLOW2,  
    RED2  
};  
  
TrafficLightState currentState = GREEN1;
```



5. Código II

```
void setup() {  
  short leds[12] = {  
    red1, red2, red3, red4,  
    yellow1, yellow2, yellow3, yellow4,  
    green1, green2, green3, green4  
  };  
  
  pinMode(buzzer1, OUTPUT);  
  pinMode(buzzer2, OUTPUT);  
  pinMode(button1, INPUT);  
  pinMode(button2, INPUT);  
  pinMode(button3, INPUT);  
  pinMode(button4, INPUT);  
  
  noTone(buzzer1);  
  noTone(buzzer2);  
  
  for (int index = 0; index < 12; index++) {  
    pinMode(leds[index], OUTPUT);  
  }  
}
```



5. Código II

```
//RESPONSÁVEL PELO SINAL SONORO
void sinalSound(int buzzer, int times, int delay1, int delay2) {
    unsigned long soundMillis = millis();
    for (int count = 0; count < times; count++) {
        if (millis() - soundMillis < delay1) {
            tone(buzzer, 2000);
        } else if (millis() - soundMillis < (delay1 + delay2)) {
            noTone(buzzer);
        } else {
            soundMillis = millis();
        }
    }
}
```



5. Código II

```
void loop() {  
    unsigned long currentMillis = millis();  
  
    int reading1 = digitalRead(button1);  
    int reading2 = digitalRead(button2);  
    int reading3 = digitalRead(button3);  
    int reading4 = digitalRead(button4);  
  
    if (reading1 == HIGH) btn1State = HIGH;  
    if (reading2 == HIGH) btn2State = HIGH;  
    if (reading3 == HIGH) btn3State = HIGH;  
    if (reading4 == HIGH) btn4State = HIGH;
```



5. Código II

```
switch (currentState) {  
  case GREEN1:  
    digitalWrite(red1, LOW);  
    digitalWrite(red4, LOW);  
  
    digitalWrite(green1, HIGH);  
    digitalWrite(red2, HIGH);  
    digitalWrite(red3, HIGH);  
    digitalWrite(green4, HIGH);  
  
    if (btn1State == HIGH || btn4State == HIGH) {  
      signalSound(buzzer1, 10, 200, 500);  
      //btn1State = LOW;  
      //btn4State = LOW;  
    } else {  
      interval = 7000;  
    }  
  
    if (currentMillis - previousMillis >= interval) {  
      previousMillis = currentMillis;  
      currentState = YELLOW1;  
      digitalWrite(green1, LOW);  
      digitalWrite(green4, LOW);  
      digitalWrite(yellow1, HIGH);  
      digitalWrite(yellow4, HIGH);  
    }  
    break;  
}
```



5. Código II

```
case YELLOW1:
    digitalWrite(red1, LOW);
    digitalWrite(red4, LOW);

    if (btn1State == HIGH || btn4State == HIGH) {
        sinalSound(buzzer1, 7, 200, 200);
        btn1State = LOW;
        btn4State = LOW;
    } else {
        interval = 3000;
    }

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        currentState = RED1;
        digitalWrite(yellow1, LOW);
        digitalWrite(yellow4, LOW);
        digitalWrite(red1, HIGH);
        digitalWrite(red4, HIGH);
    }

    btn1State = LOW;
    btn4State = LOW;
    break;
```



5. Código II

```
case RED1:  
    previousMillis = currentMillis;  
    currentState = GREEN2;  
    digitalWrite(red1, LOW);  
    digitalWrite(red4, LOW);  
    noTone(buzzer1);  
    noTone(buzzer2);  
    break;
```




5. Código II

```
case GREEN2:
  digitalWrite(green2, HIGH);
  digitalWrite(red1, HIGH);
  digitalWrite(red4, HIGH);
  digitalWrite(green3, HIGH);
  digitalWrite(red2, LOW);
  digitalWrite(red3, LOW);

  if (btn2State == HIGH || btn3State == HIGH) {
    digitalWrite(buzzer2, 10, 200, 500);
    //btn2State = LOW;
    //btn3State = LOW;
  } else {
    interval = 7000;
  }

  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    currentState = YELLOW2;
    digitalWrite(green2, LOW);
    digitalWrite(green3, LOW);
    digitalWrite(yellow2, HIGH);
    digitalWrite(yellow3, HIGH);
  }
  break;
```



5. Código II

```
case YELLOW2:
    if (btn2State == HIGH || btn3State == HIGH) {
        sinalSound(buzzer2, 7, 200, 200);
        btn2State = LOW;
        btn3State = LOW;
    } else {
        interval = 3000;
    }

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        currentState = RED2;
        digitalWrite(yellow2, LOW);
        digitalWrite(yellow3, LOW);
        digitalWrite(red2, HIGH);
        digitalWrite(red3, HIGH);
    }
    break;
```



5. Código II

```
case RED2:  
    previousMillis = currentMillis;  
    currentState = GREEN1;  
    digitalWrite(red2, LOW);  
    digitalWrite(red3, LOW);  
    noTone(buzzer1);  
    noTone(buzzer2);  
    break;
```



6. Referência

<https://www.guiadaengenharia.com/programacao-semaforo/>