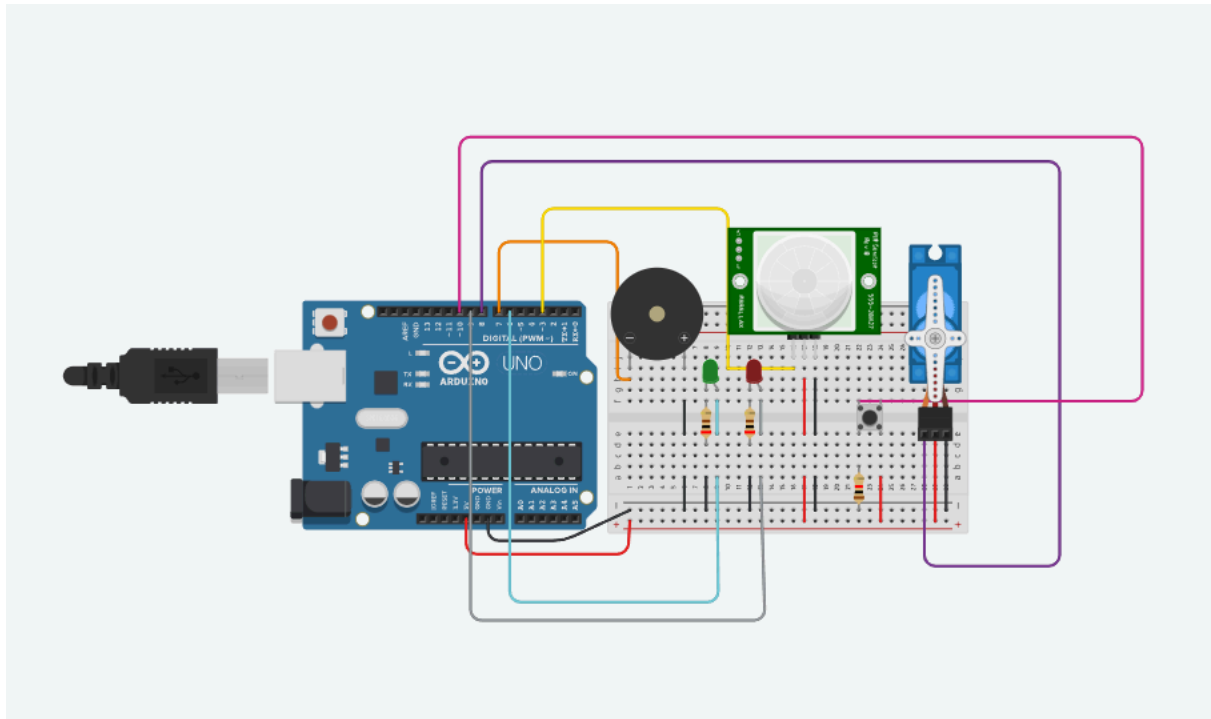


# Alarme e trava de segurança



```
#include <Servo.h>
```

```
// Inclui a biblioteca Servo, que permite controlar servomecanismos.
```

```
Servo myservo;
```

```
// Cria uma variavel 'myservo' para controlar o servo motor.
```

```
#define BUZ 7
```

```
#define SEN 3
```

```
#define LED 6
```

```
#define LEDR 9
```

```
// Define os pinos usados no circuito:
```

```
// BUZ - Pino para o buzzer.
```

```
// SEN - Pino para o sensor digital.
```

```
// LED - Pino para um LED verde.
```

```
// LEDR - Pino para um LED vermelho.
```

```
int valorbotao = 0;
```

```
int LeituraSensor;
```

```
int tempo = 0;
```

```
int frequencia = 0;
```

```
// Declara variáveis:
```

```
// valorbotao: Para armazenar o estado do botão.
```

```

// LeituraSensor: Para armazenar a leitura do sensor.
// tempo: Atualmente não é utilizado (pode ser removido ou usado se
necessário).
// frequencia: Para armazenar a frequência do buzzer.

void setup()
{
  pinMode(BUZ, OUTPUT);
  // Configura o pino BUZ como saída (para o buzzer).
  pinMode(SEN, INPUT);
  // Configura o pino SEN como entrada (para o sensor).
  pinMode(LED, OUTPUT);
  // Configura o pino LED como saída (para o LED verde).
  pinMode(LEDV, OUTPUT);
  // Configura o pino LEDV como saída (para o LED vermelho).
  pinMode(10, INPUT);
  // Configura o pino 10 como entrada (para o botão).
  myservo.attach(8);
  // Associa o objeto 'myservo' ao pino 8, onde está conectado o servo motor.
}

void loop()
{
  LeituraSensor = digitalRead(SEN);
  // Lê o valor do pino SEN e o armazena em LeituraSensor.
  valorbotao = digitalRead(10);
  // Lê o valor do pino 10 e o armazena em valorbotao.

  if(valorbotao == LOW)
  // Verifica se o estado do botão (valorbotao) é baixo (LOW).
  {
    if(LeituraSensor == LOW)
    // Se a leitura do sensor (LeituraSensor) for baixa (LOW).
    {
      digitalWrite(BUZ, LOW);
      // Desliga o buzzer (define o pino BUZ como LOW).
      noTone(BUZ);
      // Garante que não há tom no buzzer.
      digitalWrite(LED, HIGH);
      // Acende o LED verde (define o pino LED como HIGH).
    }
  }
}

```

```

    digitalWrite(LED, LOW);
    // Desliga o LED vermelho (define o pino LED como LOW).
    myservo.write(0);
    // Move o servo motor para 0 graus.
}
else
// Se a leitura do sensor não for baixa (é HIGH).
{
    digitalWrite(LED, LOW);
    // Desliga o LED verde (define o pino LED como LOW).
    digitalWrite(LED, HIGH);
    // Acende o LED vermelho (define o pino LED como HIGH).
    tone(BUZ, 1500);
    // Emite um tom de 1500 Hz no buzzer.
    delay(500);
    // Espera 500 ms.
    noTone(BUZ);
    // Para o tom no buzzer.
    delay(500);
    // Espera mais 500 ms.
    myservo.write(200);
    // Move o servo motor para 200 graus.
    delay(100);
    // Espera 100 ms.
}
}

if(valorbotao == HIGH)
// Se o estado do botão (valorbotao) for alto (HIGH).
{
    myservo.write(200);
    // Move o servo motor para 200 graus.
    digitalWrite(LED, HIGH);
    // Acende o LED verde (define o pino LED como HIGH).
    digitalWrite(LED, HIGH);
    // Acende o LED vermelho (define o pino LED como HIGH).

    for (frequencia = 150; frequencia < 1800; frequencia += 1)
    // Incrementa a frequência do buzzer de 150 Hz a 1800 Hz em passos de 1
    Hz.

```

```
{
  tone(BUZ, frequencia, tempo);
  // Emite um tom no buzzer com a frequência atual.
  delay(1);
  // Espera 1 ms antes de mudar a frequência.
}

for (frequencia = 1800; frequencia > 150; frequencia -= 1)
  // Decrementa a frequência do buzzer de 1800 Hz a 150 Hz em passos de
  1 Hz.
  {
    tone(BUZ, frequencia, tempo);
    // Emite um tom no buzzer com a frequência atual.
    delay(1);
    // Espera 1 ms antes de mudar a frequência.
  }
}
```