



# UNIRIO

**TARC 1**

**Projeto - Sistema de Segurança  
SimpleSec**

**Breno Phillip Mendes Baronte**

## 1. Introdução

O seguinte trabalho tem como proposta apresentar um sistema de segurança simples e com baixo custo que pode ser utilizado em residências ou pequenos comércios. O projeto desenvolvido tende a ser um bom começo para aqueles que tem como objetivo aumentar sua segurança sem precisar de uma quantia elevada.

O trabalho está dividido em: introdução, motivação, objetivos, desenvolvimento, testes e conclusões. A seção de desenvolvimento mostrará não só os componentes utilizados, mas também código programado para o funcionamento do sistema de segurança. Os títulos das demais seções são autoexplicativos. O trabalho foi realizado com sua prototipagem realizada na plataforma TinkerCad. [Link do protótipo.](#)

## 2. Motivação

A constante violência enfrentada pela população brasileira traz consigo a preocupação e a vontade de tentarmos nos blindar cada vez mais de possíveis episódios negativos que são enfrentados por muitos no dia a dia.

O sistema de segurança apresentado neste trabalho tem como objetivo alertar ao seu dono sempre que alguém estiver em um local onde não deveria existir circulação de pessoas ou objetos.

## 3. Objetivos

O desenvolvimento realizado neste trabalho tem como objetivo apresentar uma solução acessível para pessoas e comerciantes que desejam prevenir ou inibir furtos. Foi utilizado neste trabalho a placa de prototipagem Arduino UNO R3. A próxima seção será inteiramente dedicada a explicar a camada eletrônica do trabalho.

O sistema de segurança desenvolvido pode ser utilizado residencialmente em casas compartilhadas, onde um morador gostaria de se assegurar de que seu quarto não foi habitado em sua ausência. Também pode ser utilizado

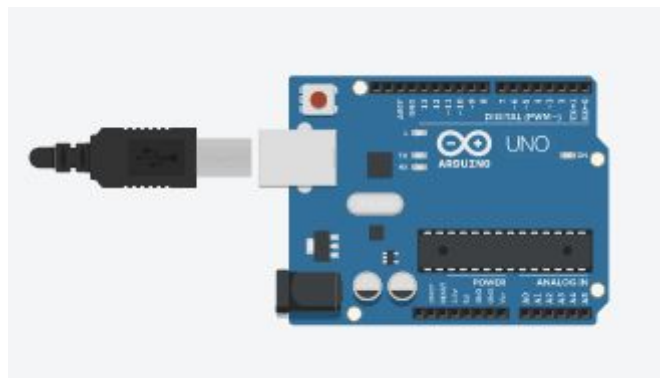
comercialmente, em situações de não atividade da loja em que o comerciante pretende se assegurar que seu estoque não foi invadido.

## 4. Desenvolvimento

Primeiramente teremos um breve descrição de cada componente utilizado e posteriormente uma descrição mais detalhada de como esses foram utilizados em conjunto, assim como o código desenvolvido em linguagem de programação para o funcionamento do sistema.

Todo o desenvolvimento da prototipação realizada neste trabalho foi realizada na plataforma [TinkerCad](#), como mencionado anteriormente na introdução.

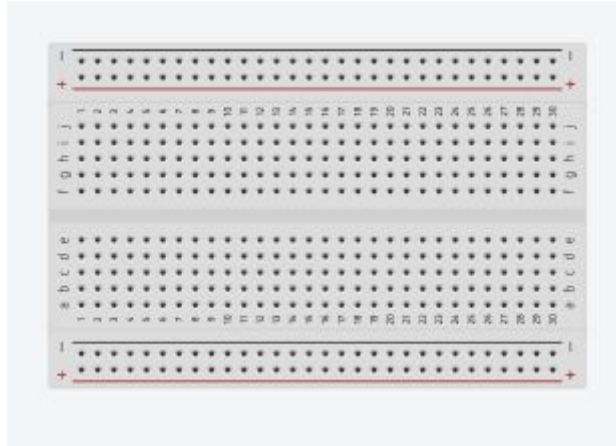
### 4.1. Descrição dos componentes



Arduino Uno R3 - é um sistema de prototipagem que serve para projetos eletrônicos sem um grande investimento.



Led - dispositivo com capacidade de emissão de luz de forma barata e eficaz.



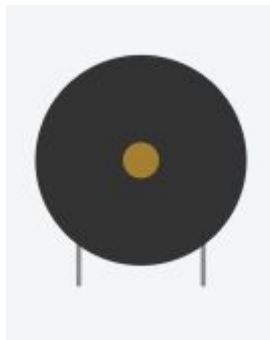
Placa de ensaio - placa com furos e conexões que conduzem energia elétrica que é utilizada para prototipagem.



Resistor - dispositivo usado em projetos elétricos devido a sua propriedade de resistência.



Botão - dispositivo analógico de comando com função de estabelecer ou interromper a carga em circuitos.



Piezo - também conhecido como “buzzer”, é um dispositivo capaz de emitir som.



Sensor PIR - dispositivo capaz de detectar movimentos realizados em uma determinada área.

#### 4.2. Funcionamento esperado

Antes de mostrarmos a montagem do protótipo, será dada uma descrição de qual é o funcionamento desejado para o circuito desenvolvido.

O sistema contará com três sensores PIR, capazes de detectar movimentos. Cada sensor PIR possui um LED correspondente que será ligado durante, e somente durante, o movimento no raio de ação do sensor. O Piezo, ou buzzer, será acionado assim que qualquer um dos sensores PIR for acionado e tocará com pequenos intervalos de pausa indefinidamente. O botão disposto no circuito é o mecanismo responsável por reiniciar o alarme e interromper o funcionamento do Piezo.

Além das funcionalidades atuais, uma mensagem é emitida no monitor serial indicando qual foi a área na qual o sensor detectou movimento e também sempre que o sistema de segurança é reiniciado.

### 4.3. Montagem do protótipo

Variáveis iniciais do circuito:

```
// General
int sensorState = LOW;
bool isAlarmActive = false;
int buttonPreviousState = 1;
int buttonCurrentState = 1;

// Leds
const int upLedPin = 2;
const int rightLedPin = 3;
const int downLedPin = 4;

// Sensors
const int upSensorPin = 13;
const int rightSensorPin = 12;
const int downSensorPin = 11;
```

Configurações iniciais dos componentes:

```
void setup()
{
  pinMode(upLedPin, INPUT);
  pinMode(upSensorPin, OUTPUT);
  pinMode(rightLedPin, INPUT);
  pinMode(rightSensorPin, OUTPUT);
  pinMode(downLedPin, INPUT);
  pinMode(downSensorPin, OUTPUT);

  pinMode(7, OUTPUT);

  pinMode(10, INPUT_PULLUP);

  Serial.begin(9600);
}
```

O objetivo principal é detectar o movimento, por isso começaremos com os sensores PIR. Eles serão ligados ao arduino e a placa de ensaio através de

fios. Assim que um movimento for detectado, iremos emitir luz através do Led correspondente a direção do sensor PIR. Para isso iremos conectar os leds na placa de ensaio e também no arduino, contando com um resistor para cada led.

Além do sinal luminoso, devemos emitir a mensagem de detecção de movimento para o monitor serial e “ligar” o alarme. Para que depois que o alarme for ativado, o buzzer seja capaz de emitir o som esperado. A nossa função que irá verificar o sensor PIR e ligar o led correspondente é:

```
void checkSensor(int sensorPin, int ledPin, String direction){  
  
    sensorState = digitalRead(sensorPin);  
  
    if (sensorState == HIGH) {  
        digitalWrite(ledPin, HIGH);  
  
        sendAlarmMessage(direction);  
  
        isAlarmActive = true;  
    }  
    else {  
        digitalWrite(ledPin, LOW);  
    }  
}
```

A função `sendAlarmMessage` que é utilizada no último trecho é a responsável pela mensagem no monitor serial.

```
void sendAlarmMessage(String direction) {  
    Serial.print("Sensor ");  
    Serial.print(direction);  
    Serial.println(" activated");  
}
```

Uma vez que o alarme for ativado, teremos uma função para verificar se é necessário acionar o buzzer.

```
void checkPiezo(){
  if (isAlarmActive == true){
    playAlarmSound();
  }
}
```

E a função de ativação do buzzer:

```
void playAlarmSound(){
  tone(8, 220, 100);
  delay(250);
}
```

Com o código até aqui conseguimos a ativação do nosso alarme porém sem poder reiniciá-lo. Para desativar o alarme precisaremos conectar o botão à placa de ensaio e ao arduino. A função que verifica se o alarme deve ser reiniciado é:

```
void checkResetButton(){

  buttonCurrentState = digitalRead(10);

  if (buttonPreviousState == LOW
      && buttonCurrentState == HIGH){
    resetAlarm();
  }

  buttonPreviousState = buttonCurrentState;
}
```

E a função para reiniciar o alarme:

```
void resetAlarm(){
  isAlarmActive = false;
  Serial.println("Alarm reseted");
}
```

Com todos esses trechos só precisamos da nossa função loop(), padrão do arduino que será:



```

void loop()
{
  checkSensor(upLedPin, upSensorPin, "Up");

  checkSensor(rightLedPin, rightSensorPin, "Right");

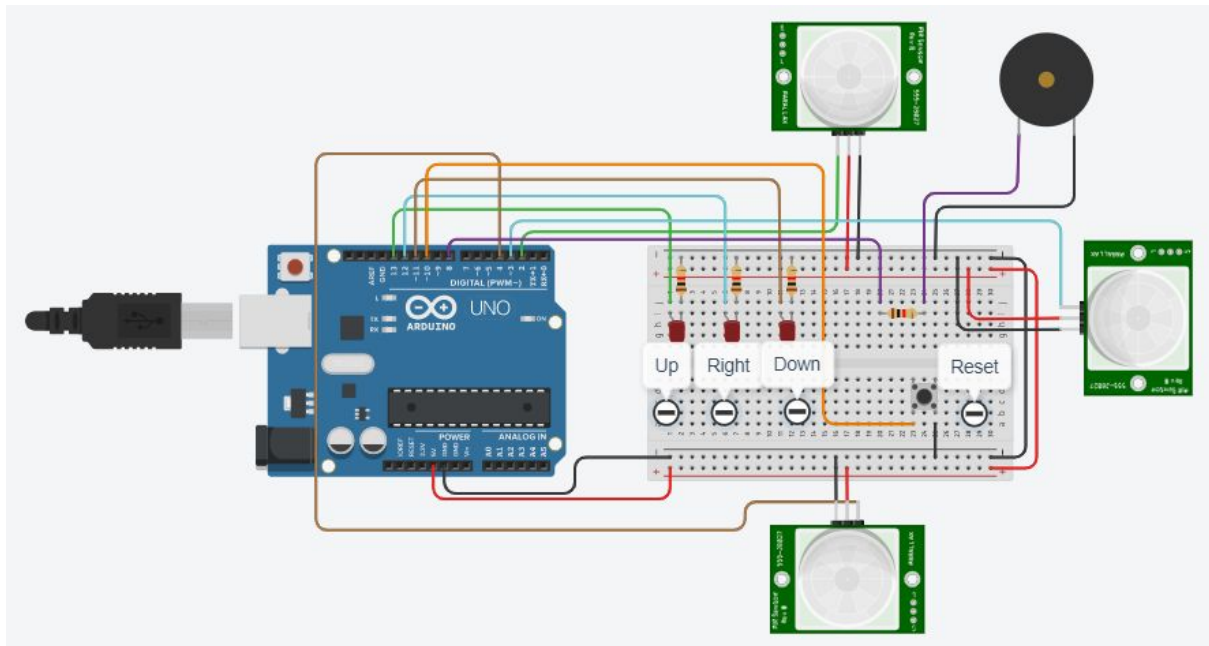
  checkSensor(downLedPin, downSensorPin, "Down");

  checkPezo();

  checkResetButton();
}

```

Imagem do circuito completo no TinkerCad:



## 5. Testes

Os testes realizados através do TinkerCad mostraram que o sistema funciona corretamente de acordo com o seu propósito. Não houveram falhas reconhecidas durante os testes.

## 6. Conclusões e Trabalhos Futuros

O sistema de segurança SimpleSec se mostrou uma alternativa funcional e barata como sistema de segurança básica.

Todavia, possíveis melhorias, que podem ser feitas em trabalhos futuros, foram detectadas, como por exemplo:

### 6.1. Sensores em todas as direções;

Isso faria com que a efetividade do projeto de segurança fosse ainda maior, já que aumentaria seu raio de atuação.

### 6.2. Botão para iniciar o sistema ao invés de somente botão para reiniciar;

Atualmente, uma vez que o projeto está ligado na eletricidade não é possível desligar. Isso faz com que sempre que o alarme precisa ser desligado ele deverá ser retirado da tomada ou bateria.

### 6.3. Controle Remoto

Um controle remoto seria muito útil em um alarme como esse, já que seu atual funcionamento necessita de um ligamento manual e para o alarme não ser acionado depois de seu ativamento a pessoa que o ligou deverá sair com cautela pela direção que o alarme não detecta.

### 6.4. Integração com sistema de gestão de segurança

Integrar com outro sistema de gestão de segurança mais completo, visto que é possível adicionar a integração na mesma parte do código que mandamos a mensagem para o monitor serial. Dessa forma, os dados gerados pelo alarme poderiam ser coletados e tratados, gerando informações de alto valor.