

PROJETO CUIDADOA+

Sistema inteligente de alarme preventivo para detecção de presença em áreas de risco, como varandas e sacadas, voltado à proteção de crianças e idosos em residências.



Sumário

1. INTRODUÇÃO	01
2. CONSTRUINDO ALARME.....	02
2.1 LISTA DE MATERIAIS	03
2.2 MONTAGEM.....	04
3. PROGRAMANDO O ALARME.....	05
3.1 Linguagem de programação.....	06
4. FUNCIONAMENTO.....	07
5. REFERÊNCIAS.....	08



1. INTRODUÇÃO

A segurança residencial é uma das principais preocupações de famílias que convivem com crianças pequenas ou idosos, especialmente em ambientes que possuem áreas de risco, como varandas, sacadas, janelas ou escadas sem proteção. Esses locais, representam potenciais de perigos quando não há supervisão constante ou barreiras físicas suficientes. Situações de descuido ou desatenção podem resultar em acidentes graves, que poderiam ser evitados com medidas preventivas simples e eficazes.



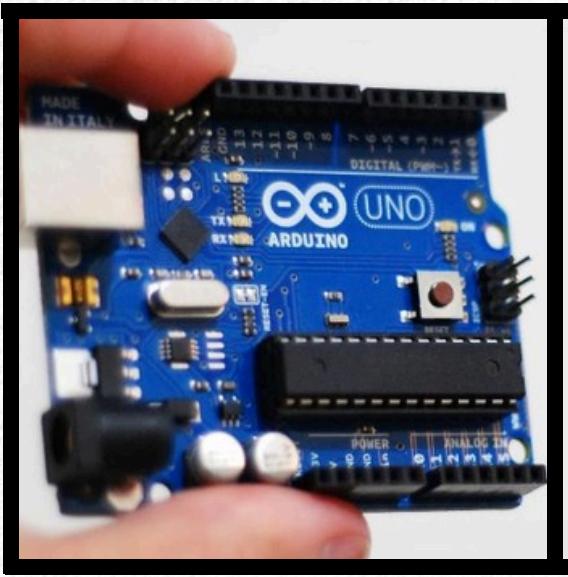
Este projeto apresenta o desenvolvimento de um sistema de alarme preventivo com detecção de presença, baseado em um sensor de movimento PIR conectado a um microcontrolador Arduino Uno. A proposta tem como objetivo identificar, de forma automática, a aproximação de uma pessoa em áreas sensíveis da casa e emitir alertas sonoros e visuais imediatos, proporcionando uma resposta rápida e reforçando a vigilância do ambiente.



2. CONSTRUINDO ALARME

- 2.1 LISTA DE MATERIAIS

1. Arduino Uno R3



1. LCD 16 x 2



1. Resistor 220 Ω



1. Cabo USB



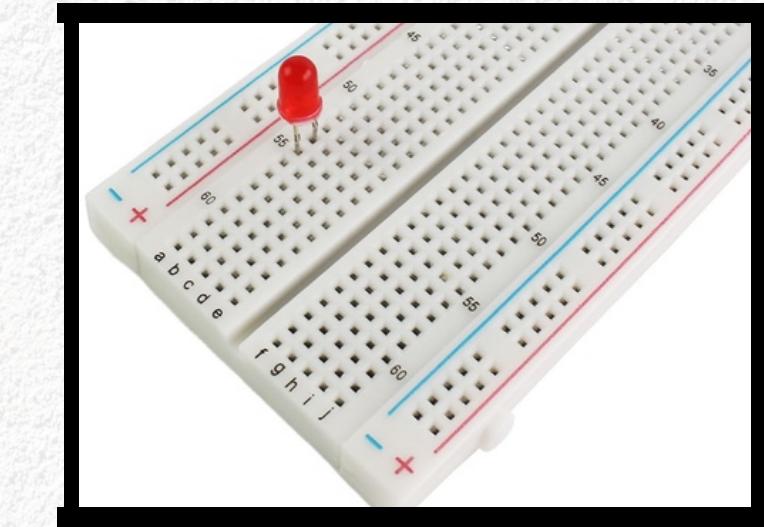
1. Piezo buzzer



1. Sensor PIR

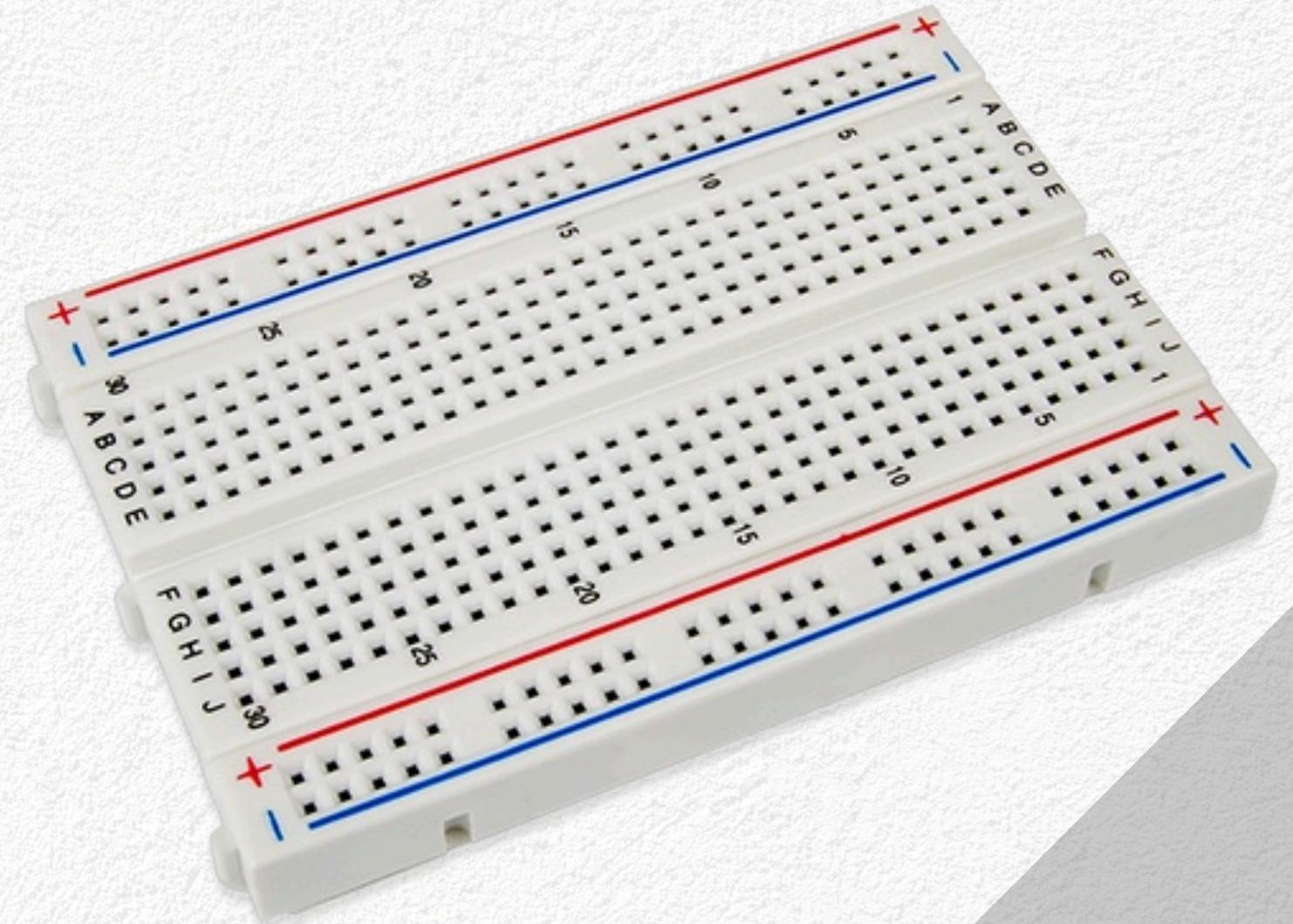


1. Placa Protoboard

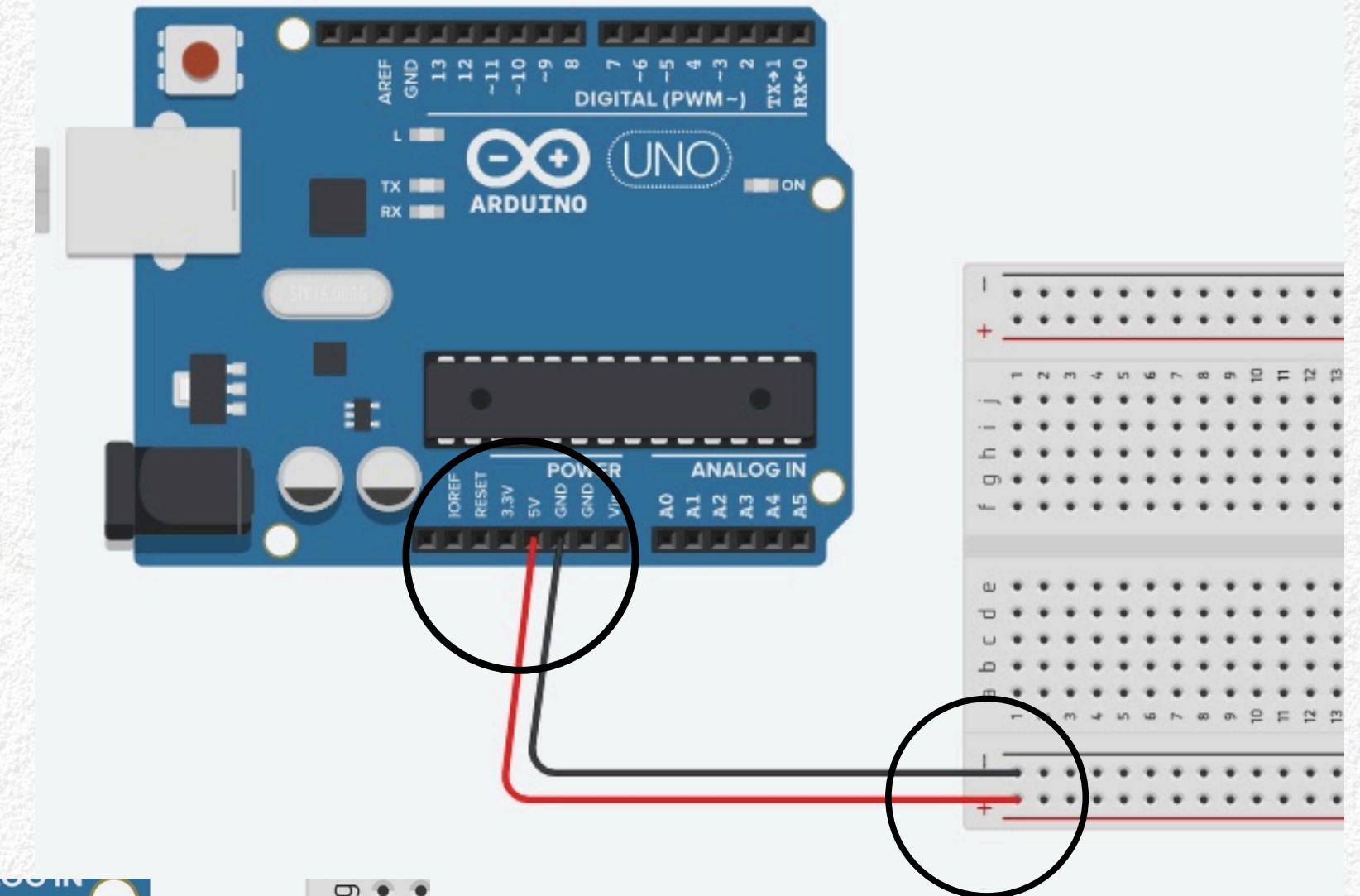
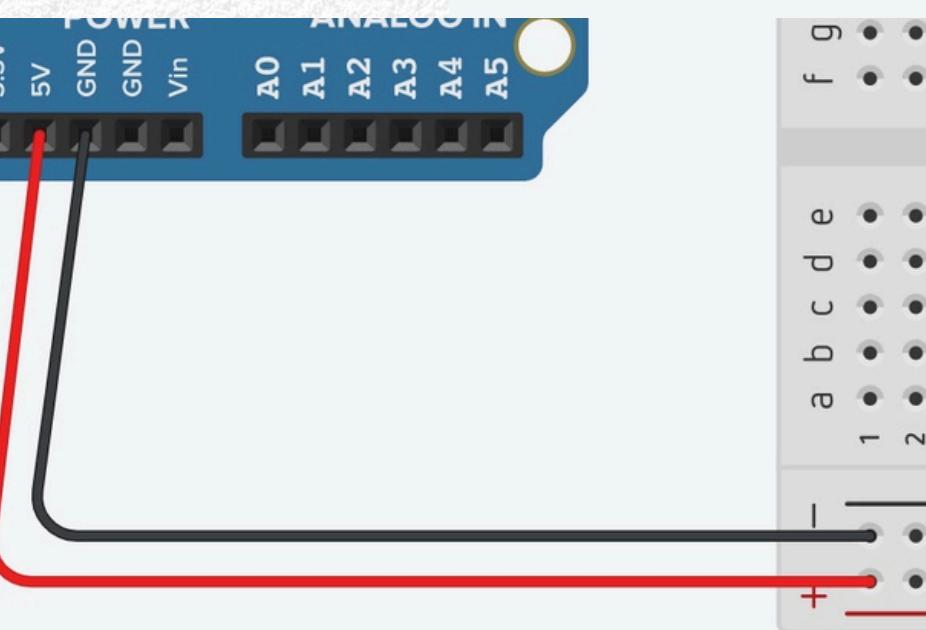


2.2 MONTAGEM

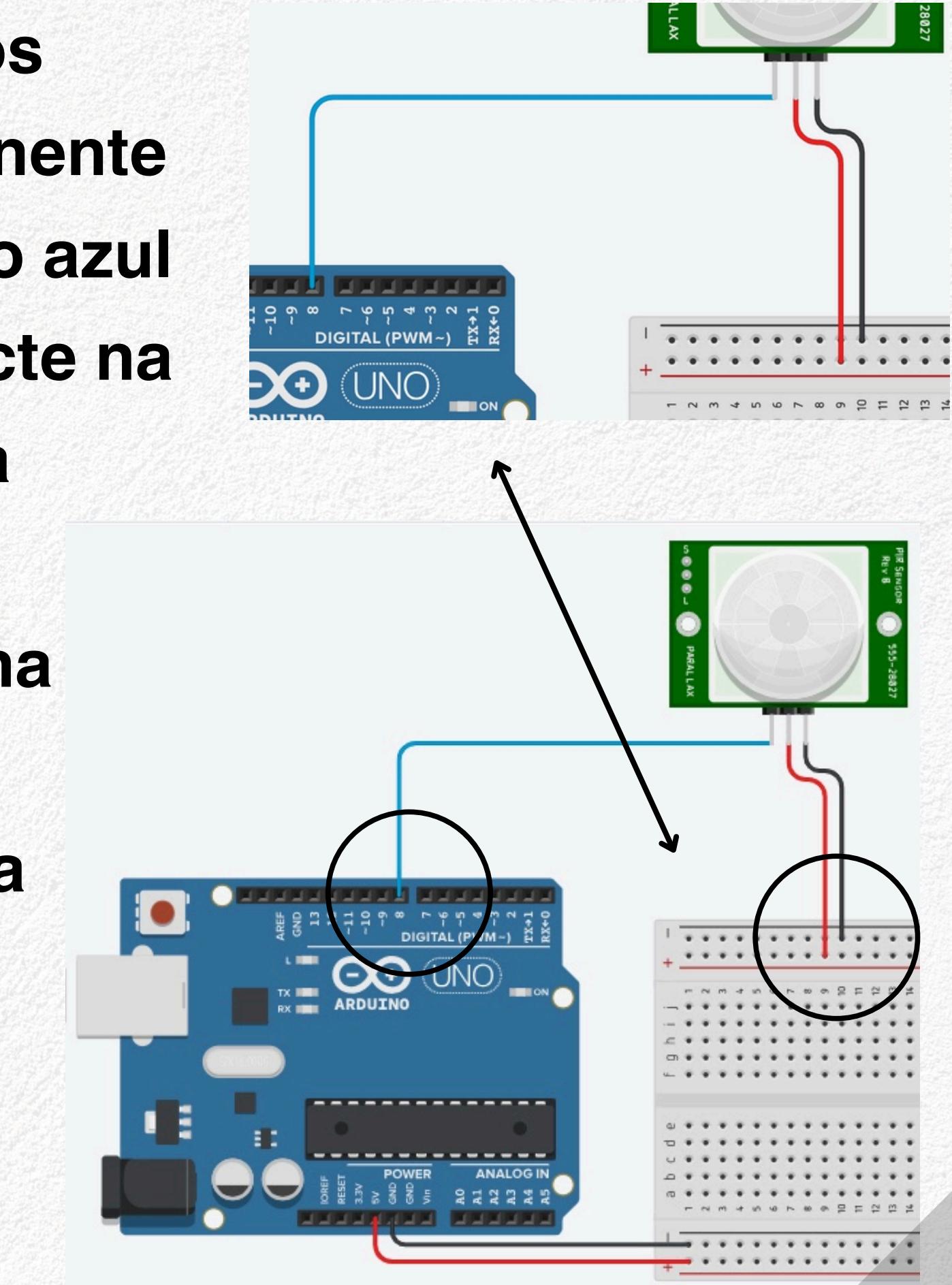
Com todos os materiais reunidos, iniciamos a montagem do protótipo. O primeiro passo é a organização dos componentes eletrônicos sobre a protoboard, uma placa de ensaio que permite a criação de circuitos sem a necessidade de solda, utilizando conexões temporárias entre os pinos e terminais dos dispositivos.



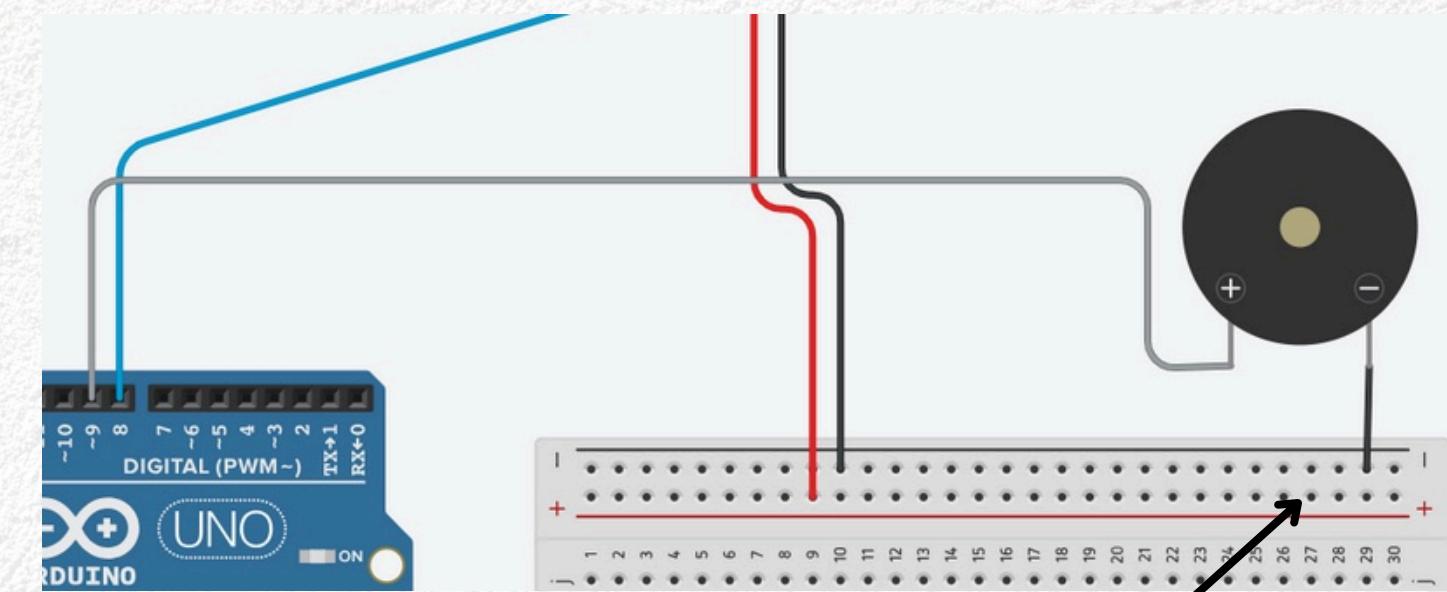
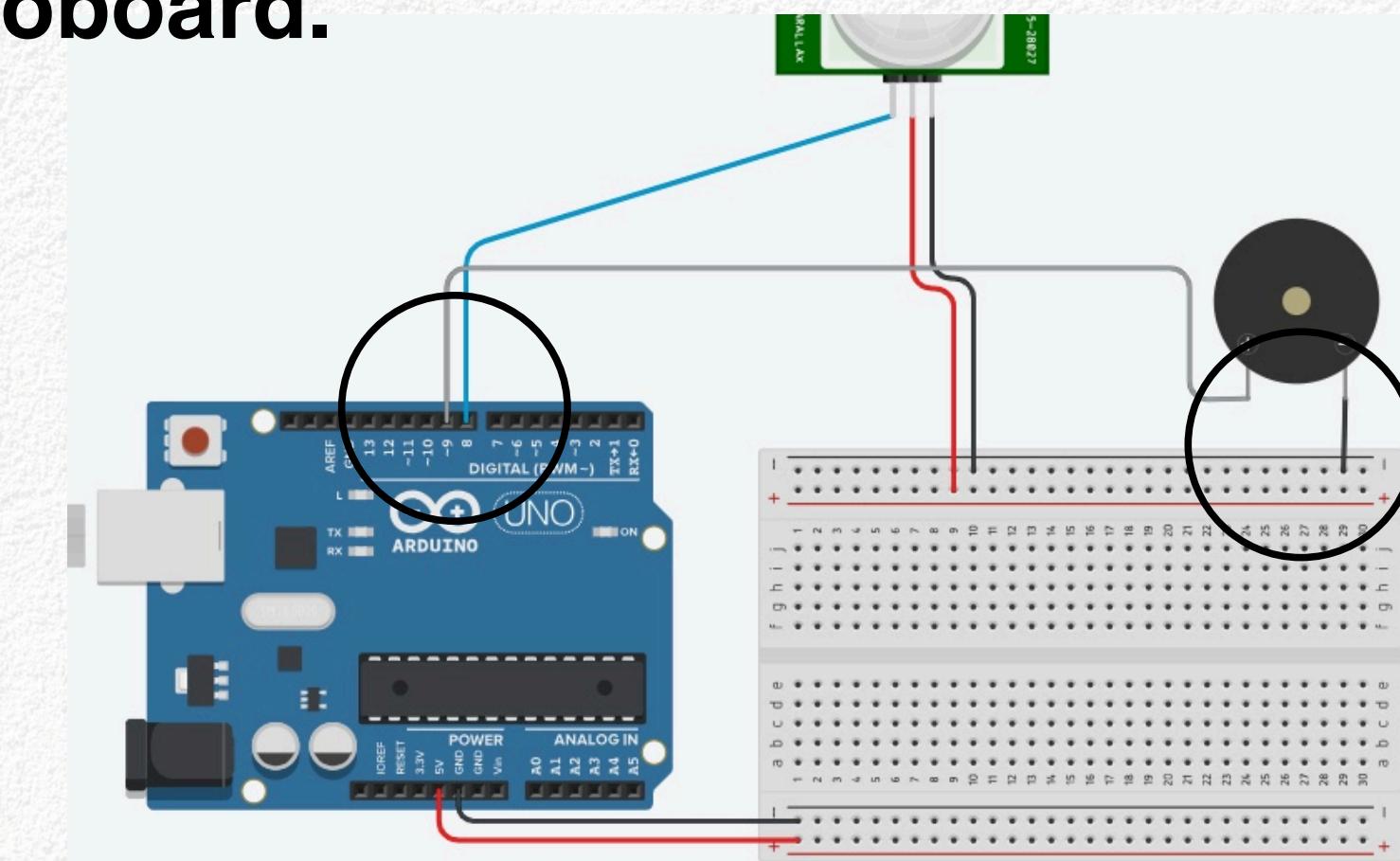
O primeiro componente eletrônico a ser conectado é a placa Arduino Uno, responsável por controlar todo o sistema. Em seguida, é feita a distribuição de energia por meio da placa Protoboard, conectando-se o terminal 5V do Arduino ao trilho de alimentação positiva (vermelho) e o GND ao trilho negativo (azul), permitindo que os demais componentes sejam alimentados corretamente.



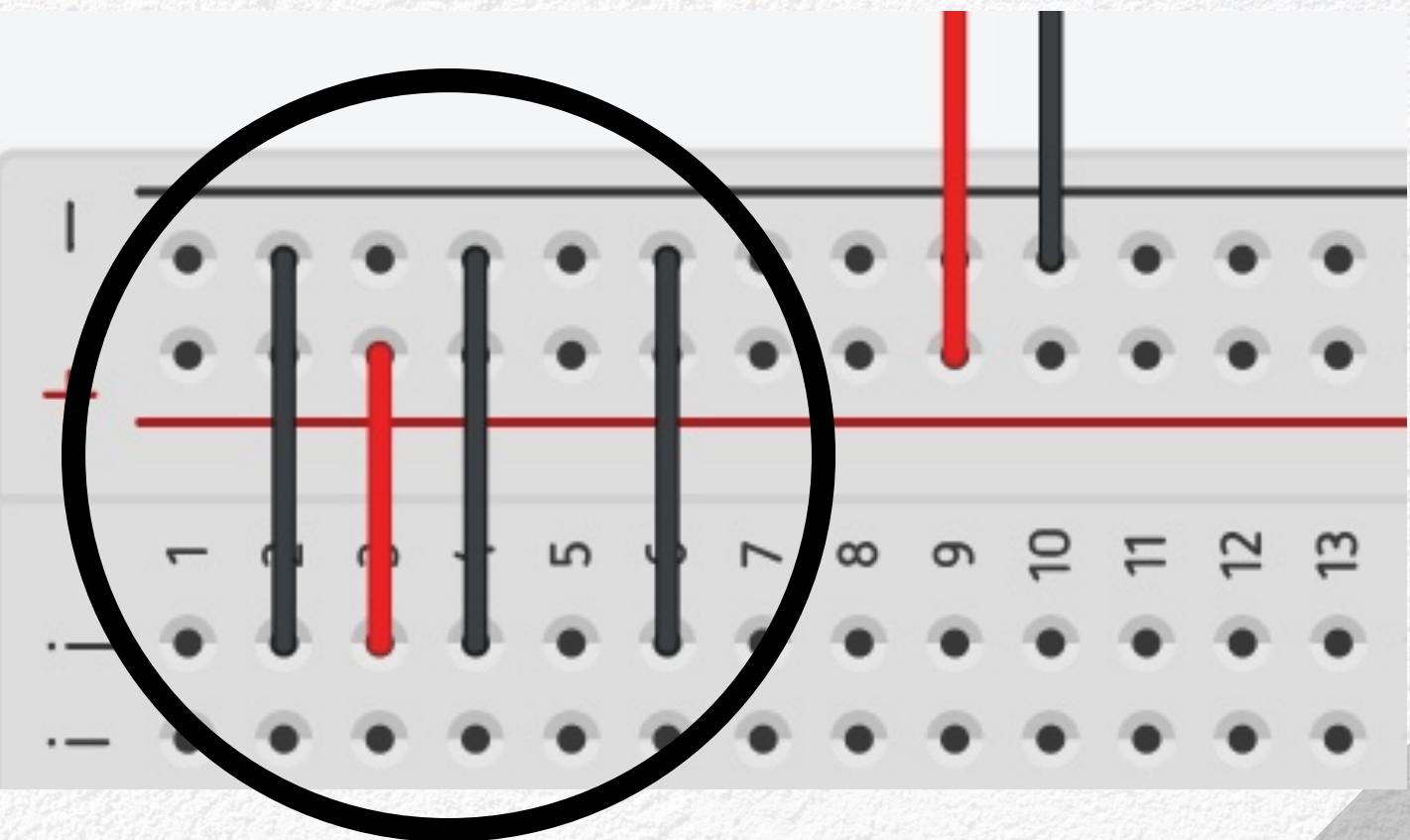
Com a base de alimentação montada, vamos para instalação do sensor PIR. Este componente possui três terminais, primeiro conecte o fio azul na entrada de (Sinal) do sensor PIR e conecte na entrada (número 8 do Arduino), em seguida conecte o fio vermelho, ele é conectado na entrada (Potência) do sensor e conectado na placa Protoboard na alimentação (positiva número 9), o fio preto (Solo) e conectado na alimentação (negativa número 10).



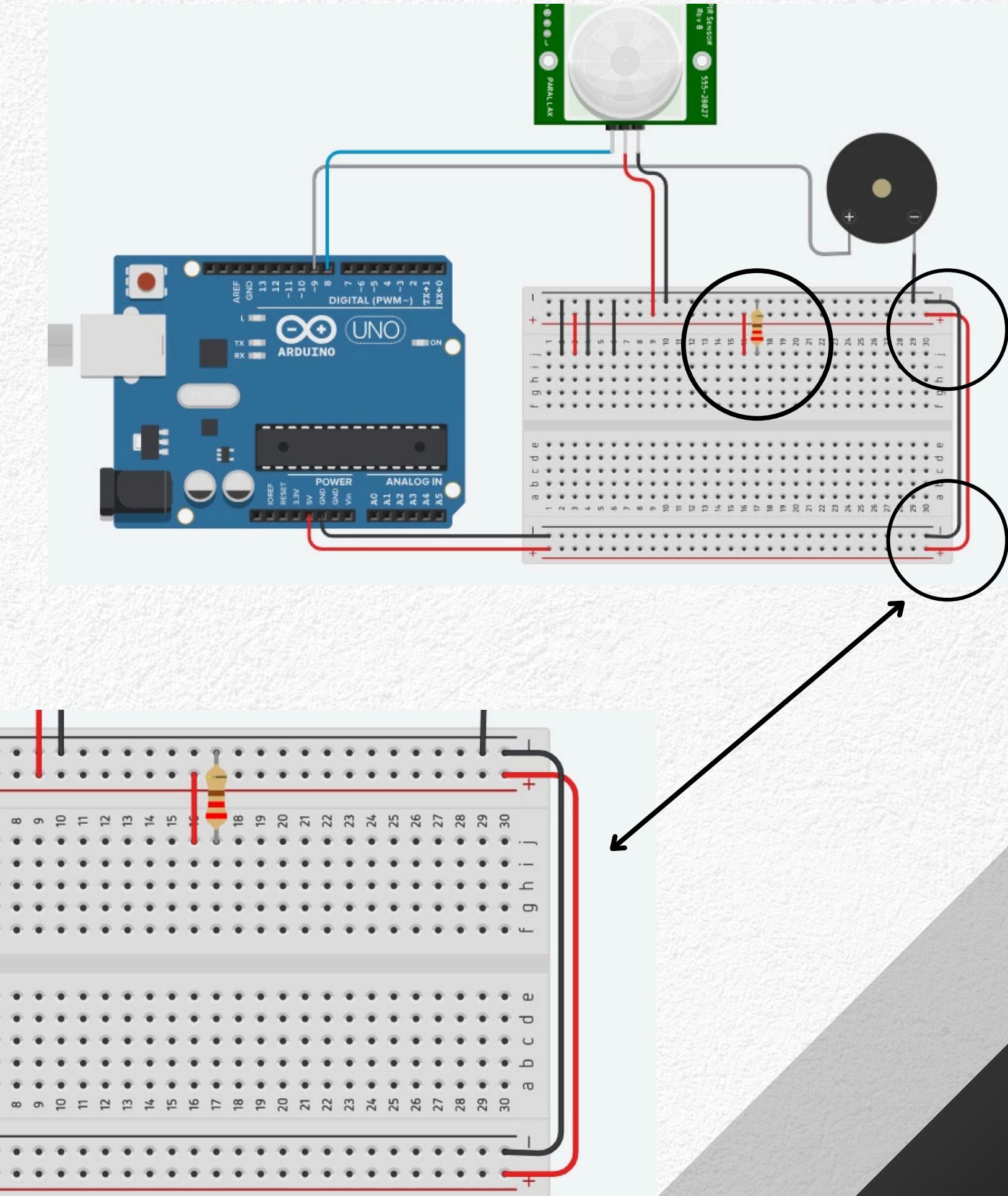
O próximo componente inserido é o piezo buzzer, que será responsável pelo aviso sonoro. Para o funcionamento puxe um fio cinza do pino positivo do buzzer e ligue ao (pino ~9) do Arduino, em seguida puxe um fio preto do buzzer pino negativo e conecte a entrada (negativa número 29) da protoboard.



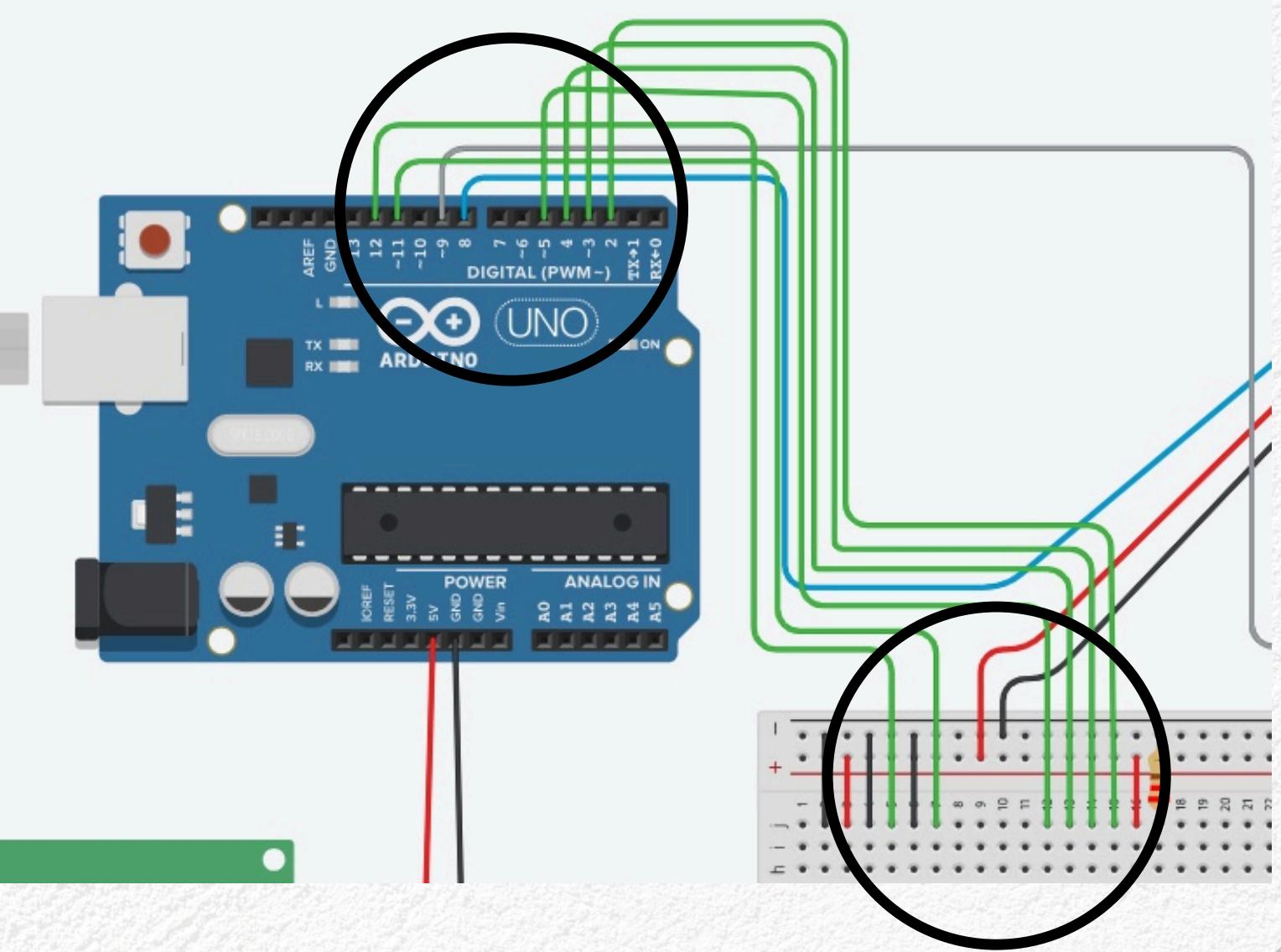
Por fim, vamos realizar a montagem do restante dos fios que vão servir para o funcionamento do display LCD 16x2, que exibirá mensagens conforme a presença ou ausência de movimento. Primeiro puxe um fio (preto da entrada negativa número 2) da protoboard para a (entrada 2/j) da protoboard, que será ligado no display LCD 16x2 na (entrada GND) . Em seguida puxe um fio positivo vermelho da protoboard número 3 para a (entrada 3/j) da protoboard, que será ligado display LCD na (entrada VCC). O próximo passo será puxar um fio preto negativo 4 da protoboard para a (entrada 4/j) que será ligado no display LCD na (entrada VO). E depois puxe um fio preto negativo para entrada 6/j.



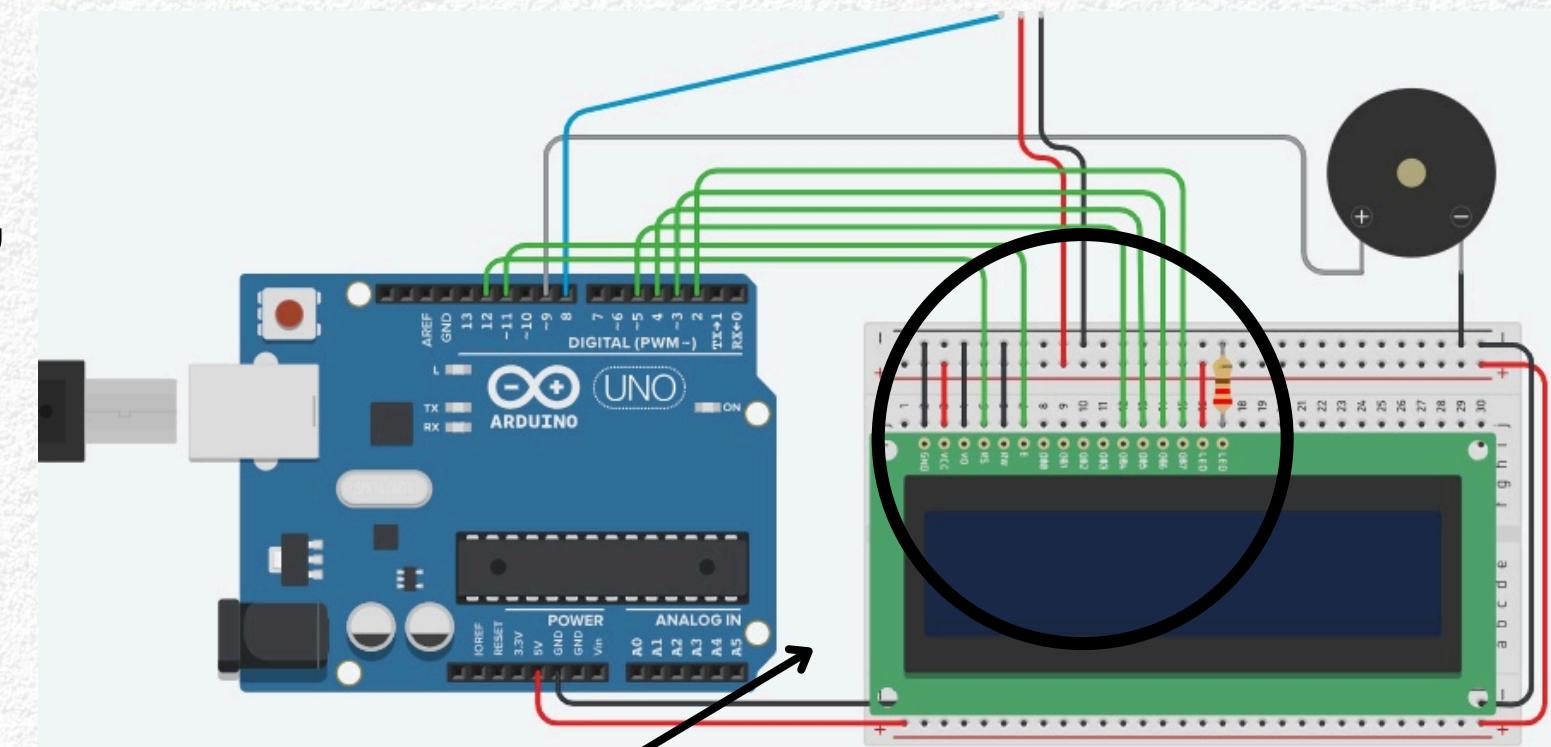
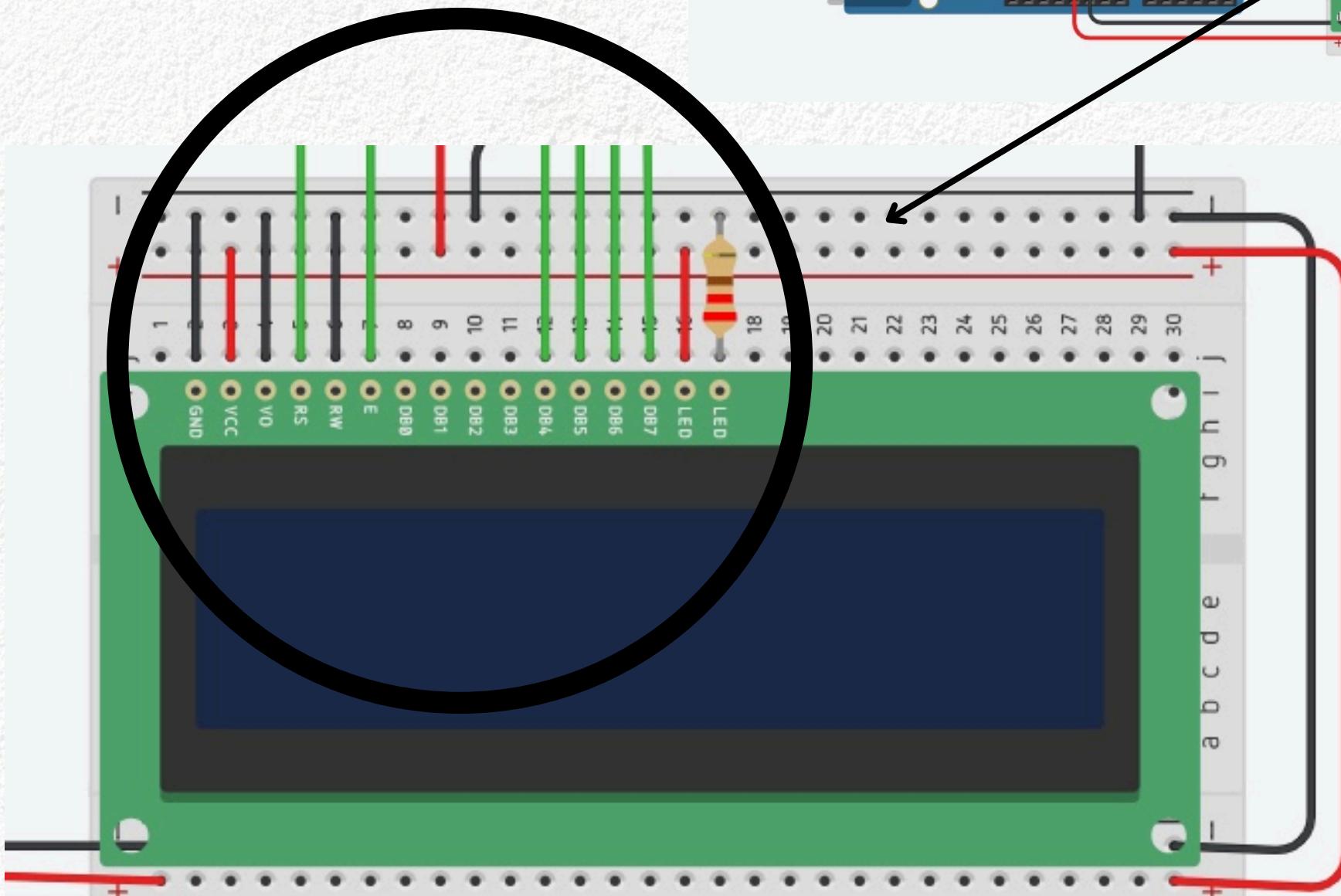
O próximo passo será colocar o resistor, ele será colocado na entrada negativa da placa protoboard e na entrada número 17 da protoboard, em seguida puxe um fio vermelho da entrada positiva para entrada numero 16 da protoboard em seguida vamos interligar os lados da protoboard, O barramento positivo do lado superior está conectado ao do lado inferior com um fio vermelho (positivo com positivo) e o barramento negativo está interligado com um fio preto (negativo com negativo).



Agora vamos para o ultimo passo que será fazer a ligação dos fios verdes e do display LCD, puxe um fio verde da entrada número 12 do Arduino e ligue na (entrada número 5/j) da placa, depois puxe um fio da entrada ~11 do Arduino para a (entrada 7/j) da placa, em seguida vamos puxar um fio do Arduino número ~5 para a (entrada 12/j) da placa protoboard lembrando que todos esses fios são verdes, depois vamos puxar um fio da entrada número 4 do Arduino para a (entrada 13/j)da placa, agora vamos para os dois últimos fios que um será puxado da entrada ~3 do Arduino para a entrada 14 da placa e o ultimo será da entrada 2 do Arduino para a entrada 15/j da placa.



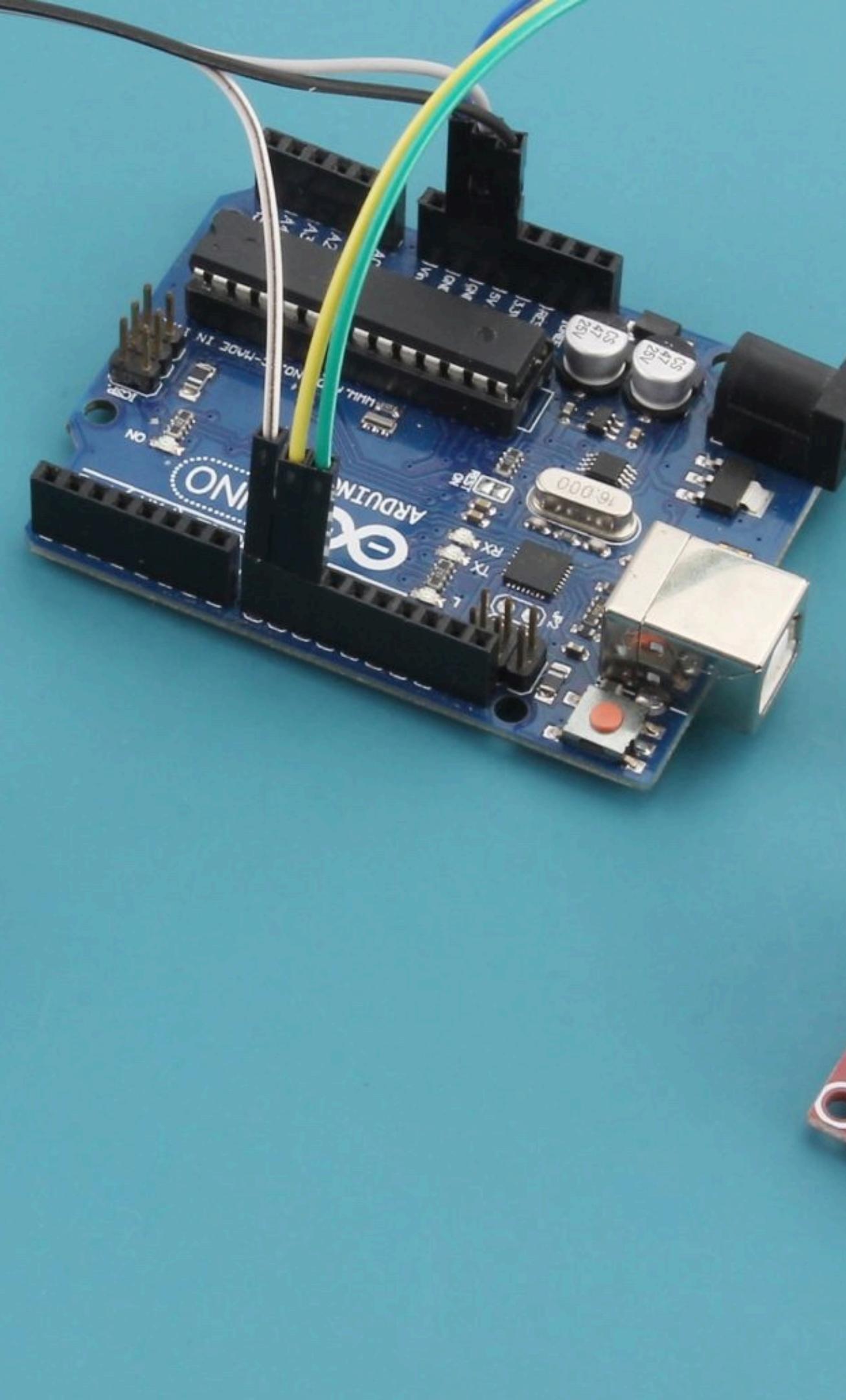
Por ultimo vamos conectar o display LCD 16x2, conecte as entradas (GND, VCC, VO, RS, RW, E, DB4, DB4, DB6, DB7, LED, LED) nos fios que conectamos a placa protoboard, na imagem ao lado mostramos mais detalhadamente.



3. PROGRAMANDO O ALARME

Nessa etapa vamos fazer a programação do alarme, para iniciar com a programação caso você esteja fazendo o projeto físico conecte a placa Arduino ao computador, através de um Cabo USB, em seguida baixe e instale o Arduino IDE (Integrated Development Environment) é o programa onde você escreve e envia os códigos pro Arduino.

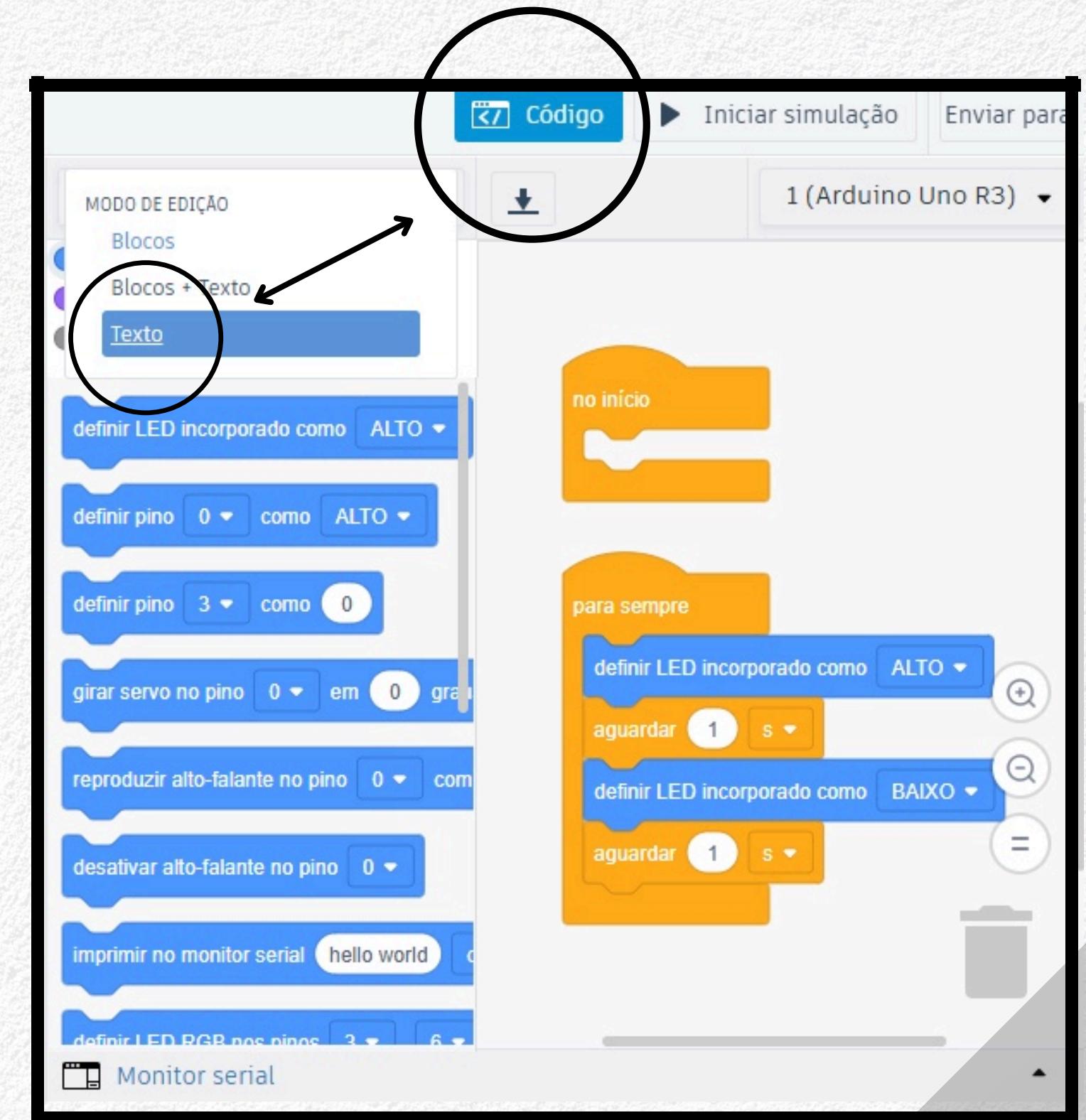
- 1. Abra a Arduino IDE**
- 2. Vá em Ferramentas > Placa > Selecione sua placa**
- 3. Vá em Ferramentas > Porta > Selecione a porta COM correta**



Se caso for montar o projeto no tinkercard faca os seguintes passo para fazer a progamação

- 1. Clique no botão " código" no canto superior direito da área de montagem**
- 2. Ele vai abrir o editor com blocos (tipo Scratch)**
- 3. Troque para "Texto", no menu suspenso onde diz "Blocos"**
- 4. Vai aparecer um aviso — clique em "Continuar"**

Agora você pode colar qualquer código C++/Arduino padrão no editor.



Código da programação na linguagem do Arduino



```
#include <LiquidCrystal.h>
#define PIR_Input 8
#define buzzer_Output 9
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define LINHAS 2
#define COLUNAS 16

void setup() {
    pinMode(buzzer_Output, OUTPUT);
    pinMode(PIR_Input, INPUT);
    Serial.begin(9600);
    lcd.begin(COLUNAS, LINHAS);
}

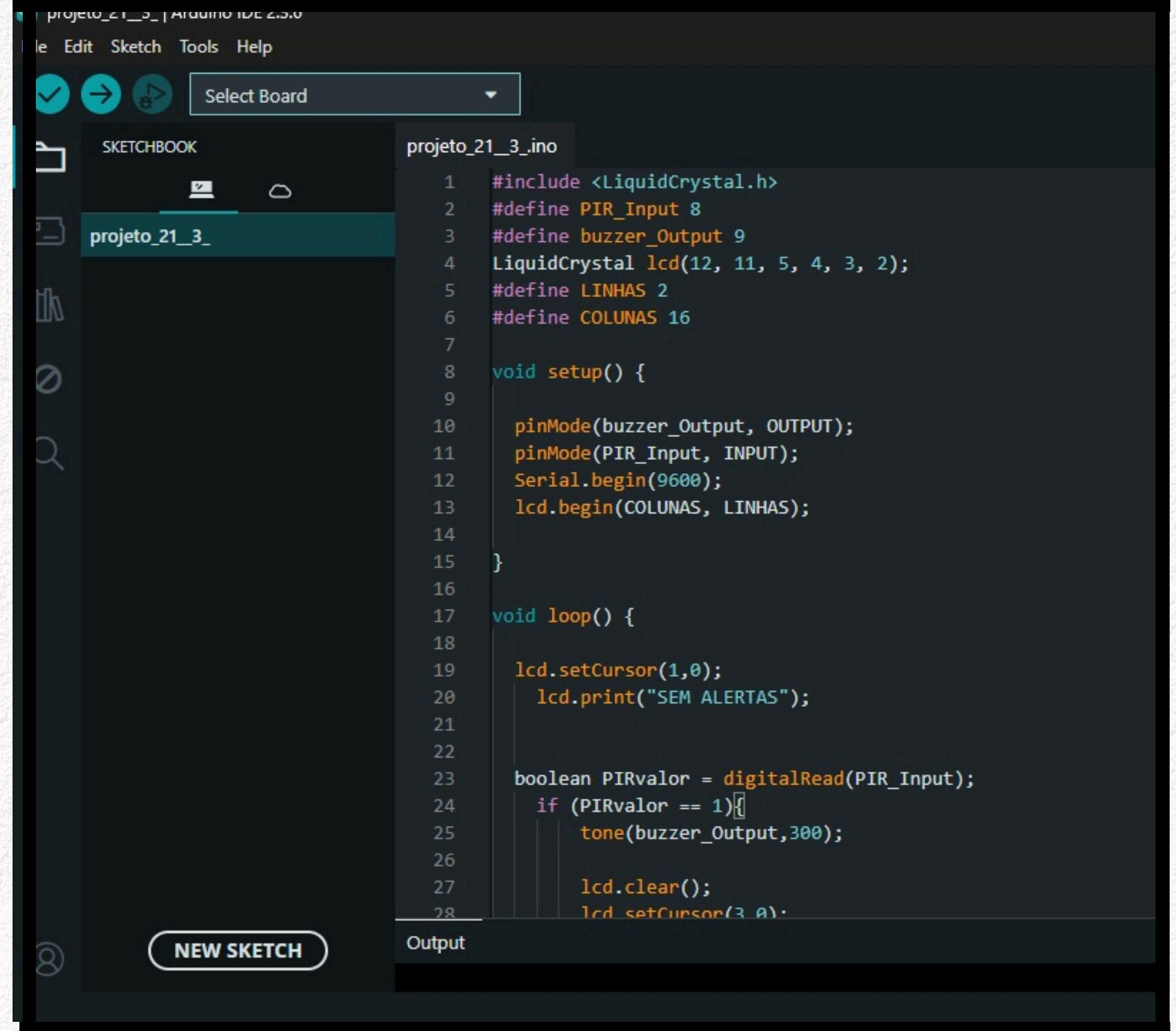
void loop() {
    lcd.setCursor(1,0);
    lcd.print("SEM ALERTAS");

    boolean PIRvalor = digitalRead(PIR_Input);
    if (PIRvalor == 1){
        tone(buzzer_Output,300);

        lcd.clear();
        lcd.setCursor(3,0);
        lcd.print("ALERTA");
        lcd.setCursor(5,1);
        lcd.print("PERIGO");
        delay(1500);
        noTone(buzzer_Output);

        lcd.clear();
    }
    delay(10);
}
```

Código da programação no Arduino IDE



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** projeto_21_3_.ino | Arduino IDE 2.5.0
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Checkmark, Run, Stop, Select Board
- Sidebar:** SKETCHBOOK, showing a folder named "projeto_21_3_" which is highlighted.
- Code Editor:** The main area displays the following C++ code for "projeto_21_3_.ino":

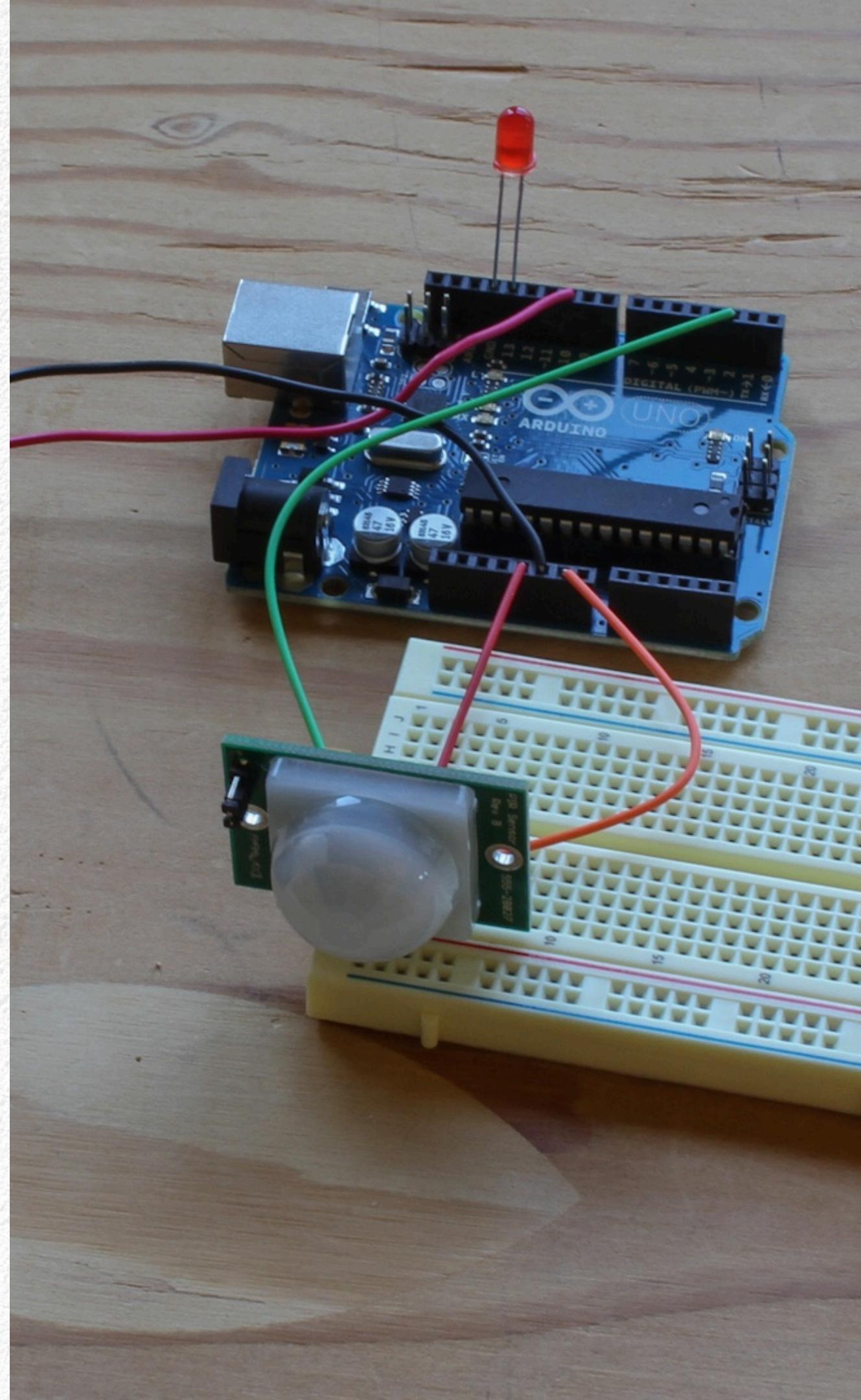
```
1 #include <LiquidCrystal.h>
2 #define PIR_Input 8
3 #define buzzer_Output 9
4 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
5 #define LINHAS 2
6 #define COLUNAS 16
7
8 void setup() {
9
10    pinMode(buzzer_Output, OUTPUT);
11    pinMode(PIR_Input, INPUT);
12    Serial.begin(9600);
13    lcd.begin(COLUNAS, LINHAS);
14
15 }
16
17 void loop() {
18
19    lcd.setCursor(1,0);
20    lcd.print("SEM ALERTAS");
21
22
23    boolean PIRvalor = digitalRead(PIR_Input);
24    if (PIRvalor == 1){
25        tone(buzzer_Output,300);
26
27        lcd.clear();
28        lcd.setCursor(3,0);
```

- Bottom Buttons:** NEW SKETCH, Output

4. FUNCIONAMENTO

Após a finalização da montagem e a devida programação do Arduino, o protótipo está apto a operar conforme o objetivo proposto. O sistema é projetado para identificar movimentos no ambiente por meio do sensor PIR, que, ao detectar variações associadas à presença de pessoas, envia um sinal digital ao microcontrolador.

O Arduino, ao receber esse sinal, interpreta a presença de movimento e aciona duas respostas imediatas: o buzzer é ativado por um período tempo até que o indivíduo saia do local de perigo, emitindo um sinal sonoro, e uma mensagem é exibida no display LCD, informando que foi detectada movimentação.



5. REFERÊNCIAS

ARDUINO BRASIL. Guia prático de sensores de presença para automação residencial. 2023. Disponível em:

<https://www.arduinoecia.com.br/sensor-presenca-arduino/>.

Acesso em: 11 abr. 2025.

Tutorial em vídeo demonstrando como adicionar e configurar um sensor de presença no Tinkercad. Disponível

em:<https://www.youtube.com/watch?v=QdjhJhORJMI>

OLIVEIRA, Euler. Como usar com Arduino – Módulo Piezoelétrico Sensor de Vibração e Toque. Blog MasterWalker Shop, 2018. Disponível em:

<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-piezoelectrico-sensor-de-vibracao-e-toque>.

Acesso em: 11 abr. 2025.

