

**CENTRO UNIVERSITÁRIO DE BRASÍLIA**  
**DISCIPLINA: CIRCUITOS ELETRÔNICOS**  
**PROFESSOR: IVANDRO DA SILVA RIBEIRO**

**TRABALHO PRÁTICO DA DISCIPLINA DE**  
**CIRCUITOS ELETRÔNICOS**

EDUARDO AFONSO DA SILVA INÁCIO, 21908507  
MATHEUS BARCELOS DE CARVALHO, 21907159  
VINÍCIUS DE OLIVEIRA PERPÉTUO, 21908298

## ● INTRODUÇÃO

Esse trabalho foi feito com a finalidade de desenvolver um sistema de automação através da ferramenta virtual Tinkercad, fazendo o uso da placa Arduino UNO e a programando para que o sistema funcione corretamente.

Os componentes utilizados foram:

- Placa LCD;
- Potenciômetro;
- Buzzer;
- Phototransistor (sensor de luz);
- TMP36 (sensor de temperatura);
- Transistor bipolar;
- Ventoinha 12V (no Tinkercad foi utilizado um Hobby Gear Motor para simular o cooler).

Desse modo, ao prontificar todo o circuito com os respectivos componentes, e executar o código no Arduino UNO, a temperatura ambiente e a necessidade ou não de iluminação apareceram na placa LCD. O Phototransistor tem o papel de controlar a iluminação, transmitindo para o Arduino UNO qual led deve ser ligado de acordo com a necessidade do sistema. Assim, caso o ambiente necessite de iluminação do sistema, o led branco é aceso simulando o acionamento de uma lâmpada. O led RGB funciona do seguinte modo, quando o sistema é ligado ele emite uma luz verde e quando o sensor de temperatura bate 40°C, o led fica vermelho.

O TMP36 é o equipamento utilizado para medir a temperatura ambiente, que é mostrada na placa LCD. Quando a temperatura atingir ou ultrapassar 40°C, um buzzer é acionado para emitir um alerta sonoro, que fica 5 segundos ligado e 5 desligado enquanto a temperatura não volta para menos que isso. Para que isso aconteça, o Hobby Gear Motor é acionado para reduzir a temperatura e só é desligado quando volta para menos de 40°C.

## • DESENVOLVIMENTO

Primeiramente, dividimos o projeto em partes separadas e unificamos em seguida para funcionar como um todo. Assim, conectamos o LED RGB na protoboard e testamos o código com as cores. Utilizamos 3 resistores de 300Ω em cada cor, ligando diretamente no Arduino UNO, nas entradas 11, 10 e 9, respectivamente.

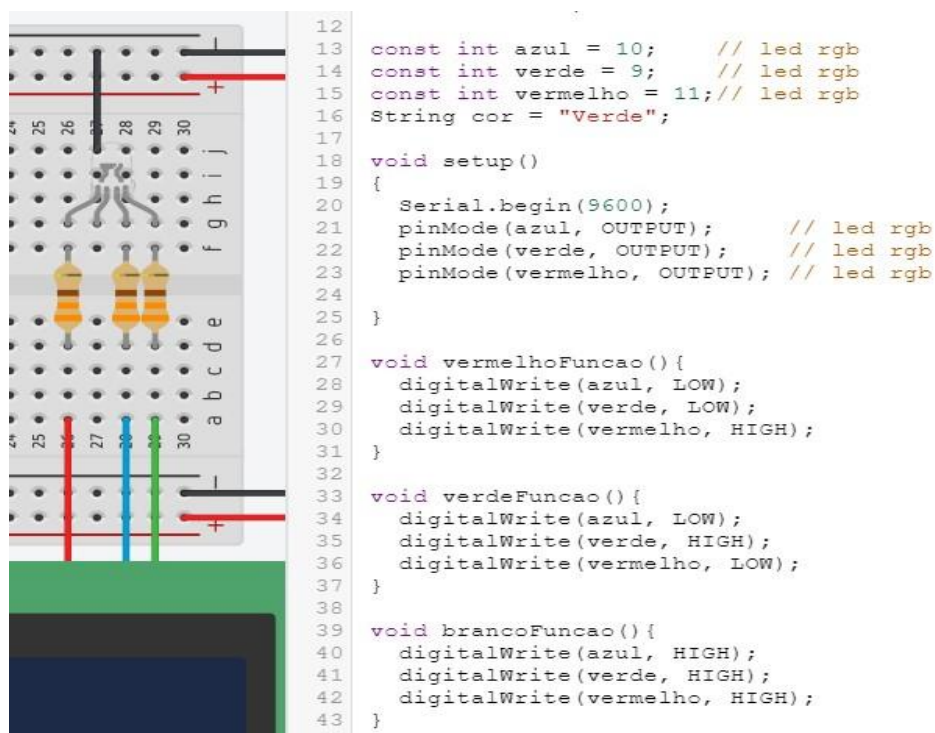


Figura 1 - Setup do LED RGB

Após isso, configuramos o sensor TMP36 (que foi utilizado para aferir a temperatura ambiente), conectando sua saída na entrada analógica A0 do Arduino UNO e utilizamos um cálculo que converte o valor que o sensor lê em graus Celsius. No caso do sensor de luz, utilizamos o Phototransistor junto a um resistor de 10KΩ, conectando na entrada analógica A1 do Arduino UNO.

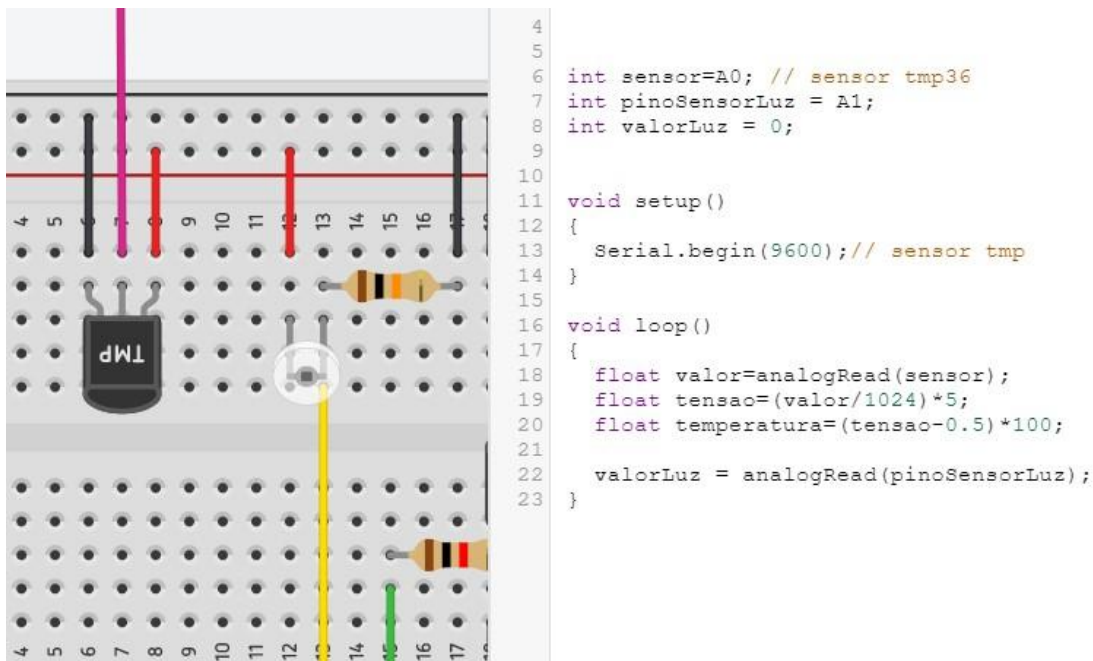


Figura 2 - Setup dos sensores de temperatura e de luz

Depois de colocarmos os dois sensores e o LED RGB, já podíamos entrar com a placa LCD para mostrar na tela os valores. Fizemos as seguintes conexões: GND no negativo da protoboard; VCC no positivo da protoboard; VO no negativo da protoboard com resistor de 1K $\Omega$ ; RS na porta 12; RW no negativo; E na porta 13; DB4 na porta 7; DB5 na porta 6; DB6 na porta 5; DB7 na porta 4; e os LEDs no potenciômetro e no ground. Para regular a iluminação da placa, o potenciômetro foi conectado com um resistor de 200 $\Omega$  no positivo.

No nosso projeto, fizemos com que o LED RGB ligasse verde e instantaneamente já mudasse para branco, já que o sensor no início da simulação aponta um ambiente sem iluminação.

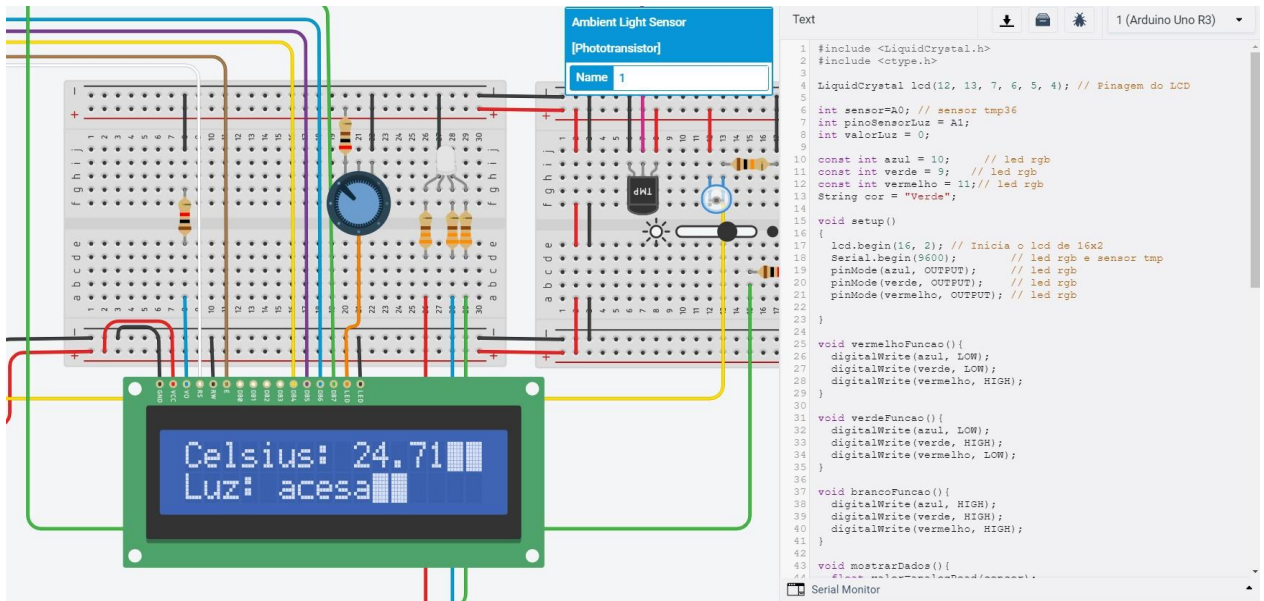


Figura 3 - Circuito LCD com lâmpada branca acesa e primeira parte do código

Durante a implementação da luz vermelha, chegamos a conclusão de que ela deveria ter máxima prioridade dentre todas as outras cores, já que é uma luz de alerta.

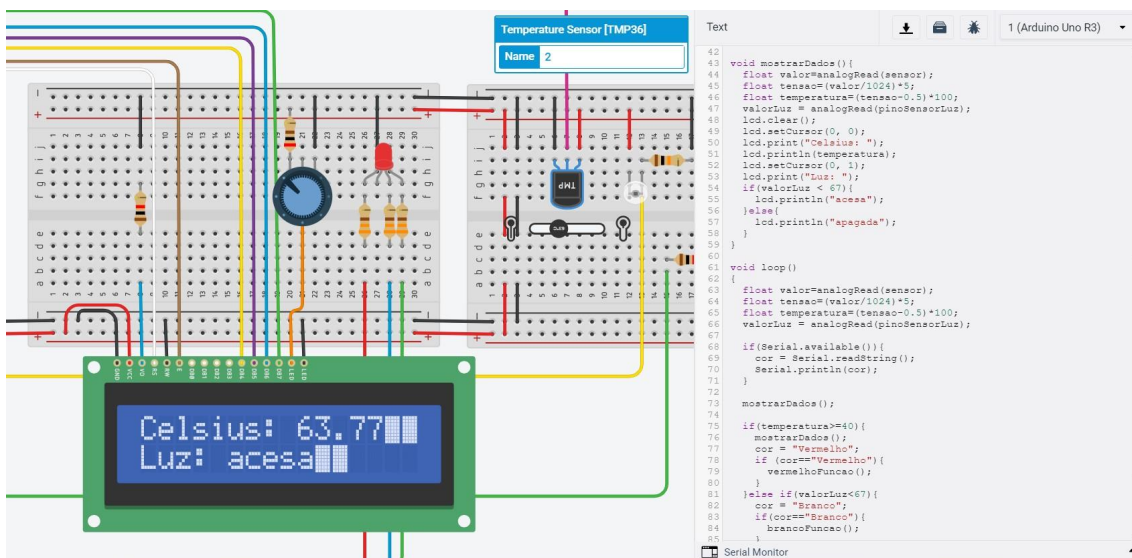


Figura 4 - Circuito LCD com luz vermelha de alerta acesa e segunda parte do código

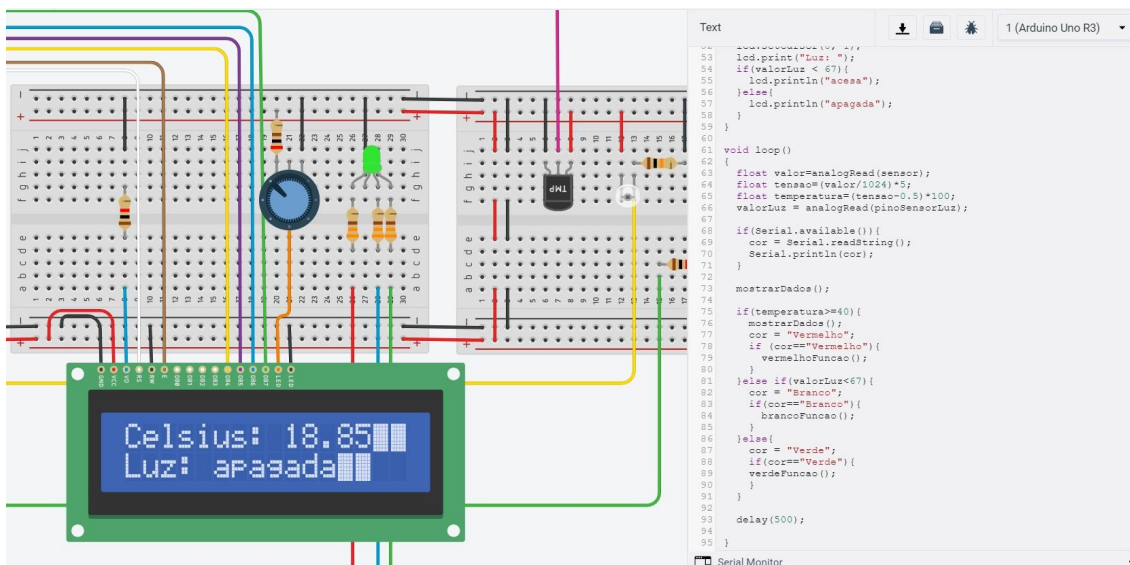


Figura 5 - Circuito LCD com luz verde acesa e terceira parte do código

Finalmente, depois de configurar todo o sistema, com os sensores e o display, o que faltava era o buzzer com um transistor bipolar e adicionar no código. Para isso, utilizamos o buzzer provido pelo Tinkercad chamado Piezo conectando seu negativo por um resistor de 5KΩ e seu positivo na porta 3, e um transistor NPN (BJT), sendo o transistor bipolar conectado a porta 2 do Arduino Uno com um resistor de 1KΩ e seu coletor conectado ao motor do sistema. Nesse sistema, nosso motor servirá para representar uma ventoinha, apesar de o mesmo não ser uma. Aqui, utilizamos um Hobby Gear Motor apenas para ligar e desligar quando for necessário.

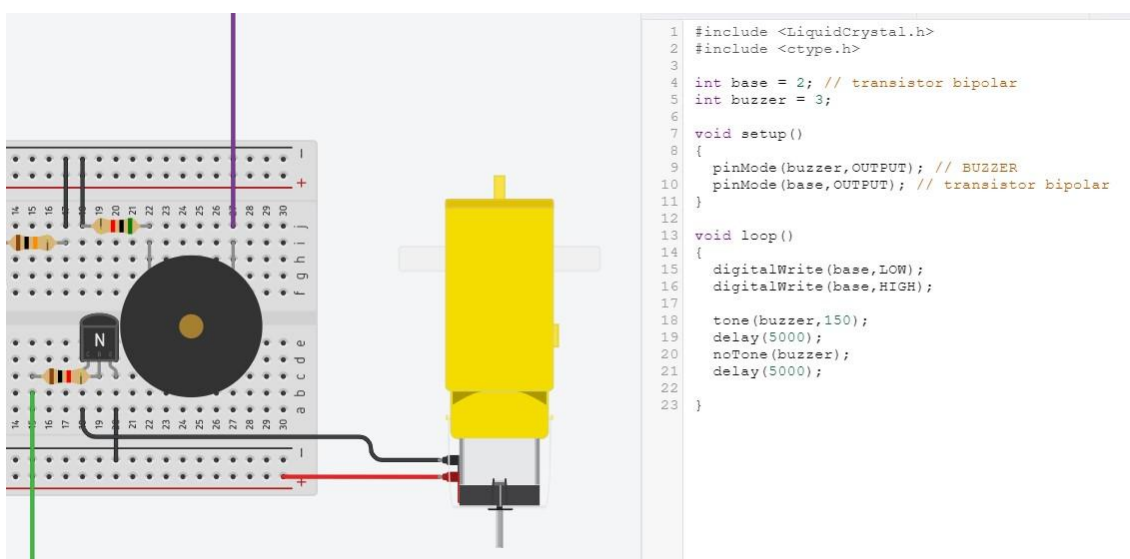


Figura 6 - Sistema do buzzer e ventoinha com transistor bipolar



Dessa forma, após programar os códigos e juntá-los, juntamos todas as peças e colocamos o projeto para funcionar. O display LCD mostrará o valor da temperatura em graus Celsius. O sensor de luz estará sempre regulando a lâmpada branca e o sensor de temperatura estará sempre regulando o buzzer juntamente com o motor e a luz vermelha. Sempre que a temperatura detectar 40°C ou acima, acionará um sistema de duração de 10 segundos onde a luz vermelha acenderá, a ventoinha irá ligar e o buzzer ficará emitindo som por 5 segundos ligado e 5 desligado. Se a temperatura continuar acima de 40°C, esse sistema continuará ativo até que a temperatura esteja no valor desejado, no caso, abaixo de 40°C. No mais, o sistema apresentará apenas uma luz verde, indicando que o sistema está ligado ou uma luz branca caso o ambiente esteja escuro.

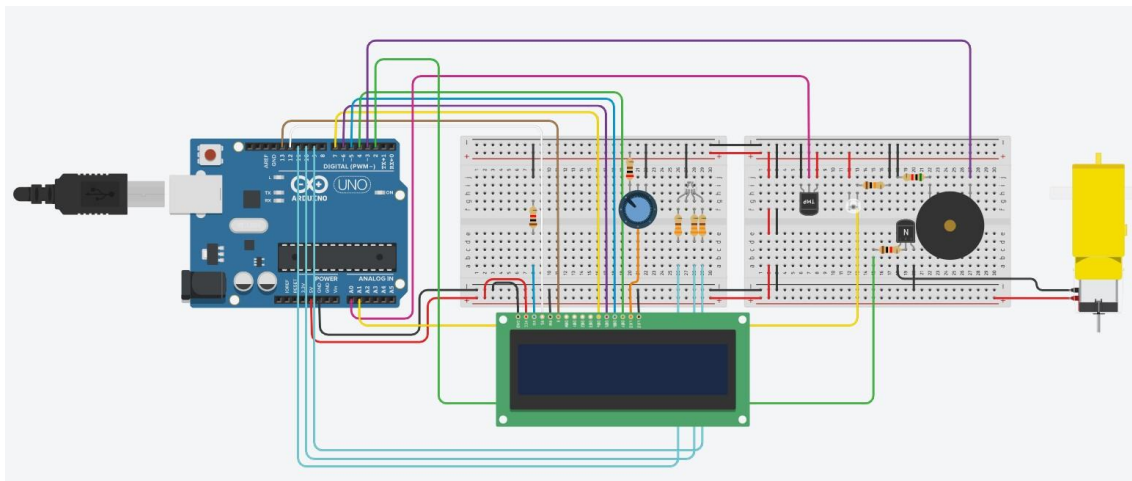


Figura 7 - Circuito completo

### **Código Fonte:**

```
#include <LiquidCrystal.h>
```

```
#include <ctype.h>
```

```
LiquidCrystal lcd(12, 13, 7, 6, 5, 4); // Pinagem do LCD
```

```
int base = 2; // transistor bipolar
```

```
int sensor=A0; // sensor tmp36
```

```
int pinoSensorLuz = A1;
```

```
int valorLuz = 0;

int buzzer = 3;

const int azul = 10;    // led rgb
const int verde = 9;    // led rgb
const int vermelho = 11; // led rgb

String cor = "Verde";

void setup()
{
    cor = "Verde";

    if(cor=="Verde"){
        verdeFuncao();
    }

    lcd.begin(16, 2); // Inicia o lcd de 16x2

    Serial.begin(9600);    // led rgb e sensor tmp

    pinMode(azul, OUTPUT);    // led rgb
    pinMode(verde, OUTPUT);    // led rgb
    pinMode(vermelho, OUTPUT); // led rgb
    pinMode(buzzer, OUTPUT); // BUZZER
    pinMode(base, OUTPUT); // transistor bipolar
}

void vermelhoFuncao(){
    digitalWrite(azul, LOW);
    digitalWrite(verde, LOW);
    digitalWrite(vermelho, HIGH);
}
```



```
void verdeFuncao(){
    digitalWrite(azul, LOW);
    digitalWrite(verde, HIGH);
    digitalWrite(vermelho, LOW);
}

void brancoFuncao(){
    digitalWrite(azul, HIGH);
    digitalWrite(verde, HIGH);
    digitalWrite(vermelho, HIGH);
}

void apagadoFuncao(){
    digitalWrite(azul, LOW);
    digitalWrite(verde, LOW);
    digitalWrite(vermelho, LOW);
}

void mostrarDados(){
    float valor=analogRead(sensor);
    float tensao=(valor/1024)*5;
    float temperatura=(tensao-0.5)*100;
    valorLuz = analogRead(pinoSensorLuz);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Celsius: ");
    lcd.println(temperatura);
    lcd.setCursor(0, 1);
```

```
lcd.print("Luz: ");  
if(valorLuz < 67){  
  lcd.println("acesa");  
}else{  
  lcd.println("apagada");  
}  
}  
  
void loop()  
{  
  float valor=analogRead(sensor);  
  float tensao=(valor/1024)*5;  
  float temperatura=(tensao-0.5)*100;  
  valorLuz = analogRead(pinoSensorLuz);  
  if(Serial.available()){  
    cor = Serial.readString();  
    Serial.println(cor);  
  }  
  mostrarDados();  
  digitalWrite(base,LOW);  
  if(temperatura>=40){  
    mostrarDados();  
    digitalWrite(base,HIGH);  
    cor = "Vermelho";  
    if (cor=="Vermelho"){  
      vermelhoFuncao();  
    }  
  }  
}
```

```
}  
    tone(buzzer,150);  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
noTone(buzzer);  
delay(500);  
mostrarDados();
```

```
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
delay(500);  
mostrarDados();  
    }else if(valorLuz<67){  
cor = "Branco";  
if(cor=="Branco"){  
    brancoFuncao();  
}  
}else{  
    cor = "Verde";
```

```
if(cor=="Verde"){  
  verdeFuncao();  
}  
}  
delay(500);  
}
```

- **CONCLUSÃO**

Portanto, ao realizar esse trabalho, foi possível simular o funcionamento do circuito como um todo e viabilizou por em prática conhecimentos teóricos da disciplina de Circuitos Eletrônicos adquiridos no decorrer do semestre. Como por exemplo, como o uso de resistores, evitando danos nos componentes; uso e leitura de sensores, para analisar e desempenhar as funções do sistema todo e o uso de transistores em um circuito, controlando o acionamento do motor.

## ● REFERÊNCIAS

- <https://www.embarcados.com.br/arduino-uno/#:~:text=A%20placa%20Arduino%20UNO%20é,feita%20através%20do%20protocolo%20STK500.>
- <https://multilogica-shop.com/sensor-de-temperatura-tmp36#:~:text=O%20TMP36%20é%20um%20sensor.C%20a%20%2B125°C.>
- <https://doc-0o-b4-apps-viewer.googleusercontent.com/viewer/secure/pdf/hrqrn1qa3845lv2lok83i3u0sv3o5egi/i20vdkprbeaqpci56576ipbdvs0uucei/1624579200000/drive/08526946867258917380/ACFrOgChDMtXdt-1tDW89B9jIPOruEUNDiCgY4N6qxR26M6FEdpW-YZDDk7hUZnTKYCbSJanveofs2pK3FfNjoUvzOvHsehO1rVWwtggclnrtoWiLIjEiT60jw2ZHZ4si6YHgQZ0THi36ZWxil0?print=true&nonce=300cto81ep59s&user=08526946867258917380&hash=3u54829b9unn24u07j1b2kh4paboapvk>
- <https://portal.vidadesilicio.com.br/usando-o-buzzer-com-arduino-transdutor-piezo-eletrico/>