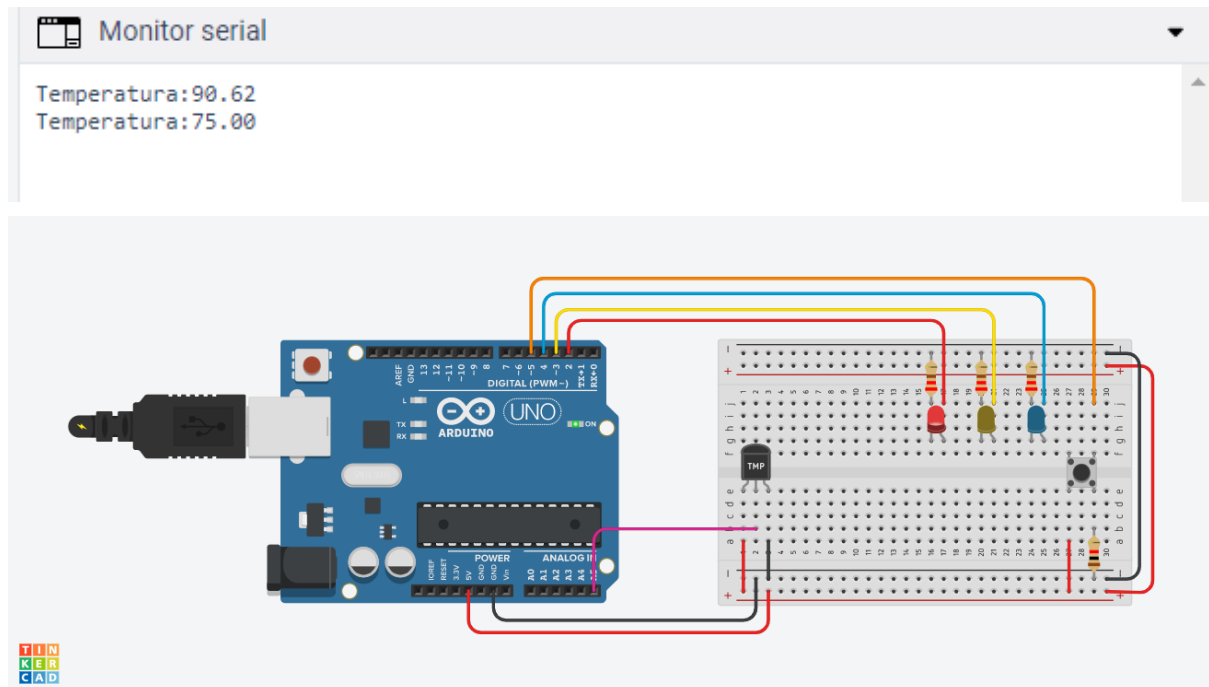


Simulação:

<https://www.tinkercad.com/things/dXHkJJOZ9kh>

Simulando temperaturas entre 60°C e 100°C, sinalizado pelo led de cor vermelha:



Código:

```
// Aula Prática 7
// - Entre -40°C e 25°C o LED azul deve permanecer totalmente aceso;
// - Entre 25°C e 60°C o LED amarelo deve aumentar seu brilho
gradualmente conforme o aumento da temperatura;
// - Entre 60°C e 100°C o LED vermelho deve piscar à uma frequência de
5 Hz indicando alerta;
// - Se ultrapassar os 100°C todos os LED's devem acender e o sistema
deverá ser desativado;
// - O sistema só retornará quando o botão for pressionado e a
temperatura estiver abaixo dos 25°C;
// - Utilize o canal serial para exibir a temperatura de leitura no
monitor serial;

// Definir os pinos dos botões e do LED
#define LED_AZUL 4           // pino do LED azul
#define LED_AMARELO 3        // pino do LED amarelo
#define LED_VERMELHO 2       // pino do LED vermelho
#define botao 5              // pino do botão
int sensorTemperatura = A5; // pino do sensor de temperatura
```

```

// Variáveis
int s = 1; // estado do sistema
int estadoBotao = 0; // estado do botão
float temperatura = 0; // temperatura
float temperaturaAnterior = 0; // temperatura anterior

void setup()
{
    pinMode(LED_AZUL, OUTPUT); // define o pino do LED azul como
saída
    pinMode(LED_AMARELO, OUTPUT); // define o pino do LED amarelo como
saída
    pinMode(LED_VERMELHO, OUTPUT); // define o pino do LED vermelho
como saída
    pinMode(botao, INPUT); // define o pino do botão como
entrada
    Serial.begin(9600); // inicia a comunicação serial
}

void loop()
{
    estadoBotao = digitalRead(botao); // lê o estado do
botão
    int valorSensor = analogRead(sensorTemperatura); // lê o valor do
sensor de temperatura
    float voltage = (valorSensor * 5.0) / 1024.0; // converte o
valor lido para tensão
    temperatura = (voltage - 0.5) * 100; // converte a
tensão para temperatura em graus Celsius
    if (temperatura != temperaturaAnterior) // se a
temperatura for diferente da temperatura anterior
    {
        temperaturaAnterior = temperatura; // atualiza a temperatura
anterior
        Serial.print("Temperatura:"); // exibe a mensagem no
monitor serial
        Serial.println(temperatura); // exibe a temperatura no
monitor serial
    }

    if (s == 1) // se o sistema estiver ativo
    {

```

```

    if (temperatura >= -40 && temperatura <= 25) // se a
temperatura estiver entre -40°C e 25°C
    {
        digitalWrite(LED_AZUL, HIGH);    // acende o LED azul
        analogWrite(LED_AMARELO, 0);    // apaga o LED amarelo
        digitalWrite(LED_VERMELHO, LOW); // apaga o LED vermelho
    }
    else if (temperatura > 25 && temperatura <= 60) // se a
temperatura estiver entre 25°C e 60°C
    {
        digitalWrite(LED_AZUL, LOW);
// apaga o LED azul
        analogWrite(LED_AMARELO, map(temperatura, 25, 60, 0, 255));
// aumenta gradativamente o brilho do LED amarelo conforme o aumento da
temperatura
        digitalWrite(LED_VERMELHO, LOW);
// apaga o LED vermelho
    }
    else if (temperatura > 60 && temperatura <= 100) // se a
temperatura estiver entre 60°C e 100°C
    {
        digitalWrite(LED_AZUL, LOW);    // apaga o LED azul
        analogWrite(LED_AMARELO, 0);    // apaga o LED amarelo
        digitalWrite(LED_VERMELHO, HIGH); // acende o LED vermelho
        delay(100);
        digitalWrite(LED_VERMELHO, LOW); // apaga o LED vermelho
        delay(100);
    }
    else if (temperatura > 100) // se a temperatura ultrapassar os
100°C
    {
        digitalWrite(LED_AZUL, HIGH);    // acende o LED azul
        analogWrite(LED_AMARELO, 1023);  // acende o LED amarelo
        digitalWrite(LED_VERMELHO, HIGH); // acende o LED vermelho
        s = 0;                          // desativa o sistema
    }
}

// se o botão for pressionado e a temperatura estiver abaixo de
25°C
if (estadoBotao == HIGH && temperatura < 25)
{
    s = 1;
}

```

```
}  
}
```

Conclusão

Neste relatório, apresentamos o desenvolvimento de um exercício prático de programação de um sistema de controle de temperatura com Arduino e LEDs. O objetivo era simular diferentes faixas de temperatura e acionar os LEDs de acordo com as condições estabelecidas. Utilizamos um sensor de temperatura, um botão e três LEDs de cores diferentes para representar as variações de temperatura. O código foi escrito em linguagem C++ e testado no simulador online Tinkercad. Os resultados mostraram que o sistema funcionou conforme o esperado, acendendo, apagando ou piscando os LEDs conforme a temperatura lida pelo sensor. O sistema também foi capaz de desativar-se quando a temperatura ultrapassou os 100°C e reativar-se quando o botão foi pressionado e a temperatura estava abaixo de 25°C. A comunicação serial permitiu exibir a temperatura no monitor serial. Este trabalho permitiu compreender os conceitos básicos de programação com Arduino, bem como aplicar os conhecimentos de eletrônica e física na construção de um circuito simples e funcional.