

Monitoramento de Frequência Cardíaca

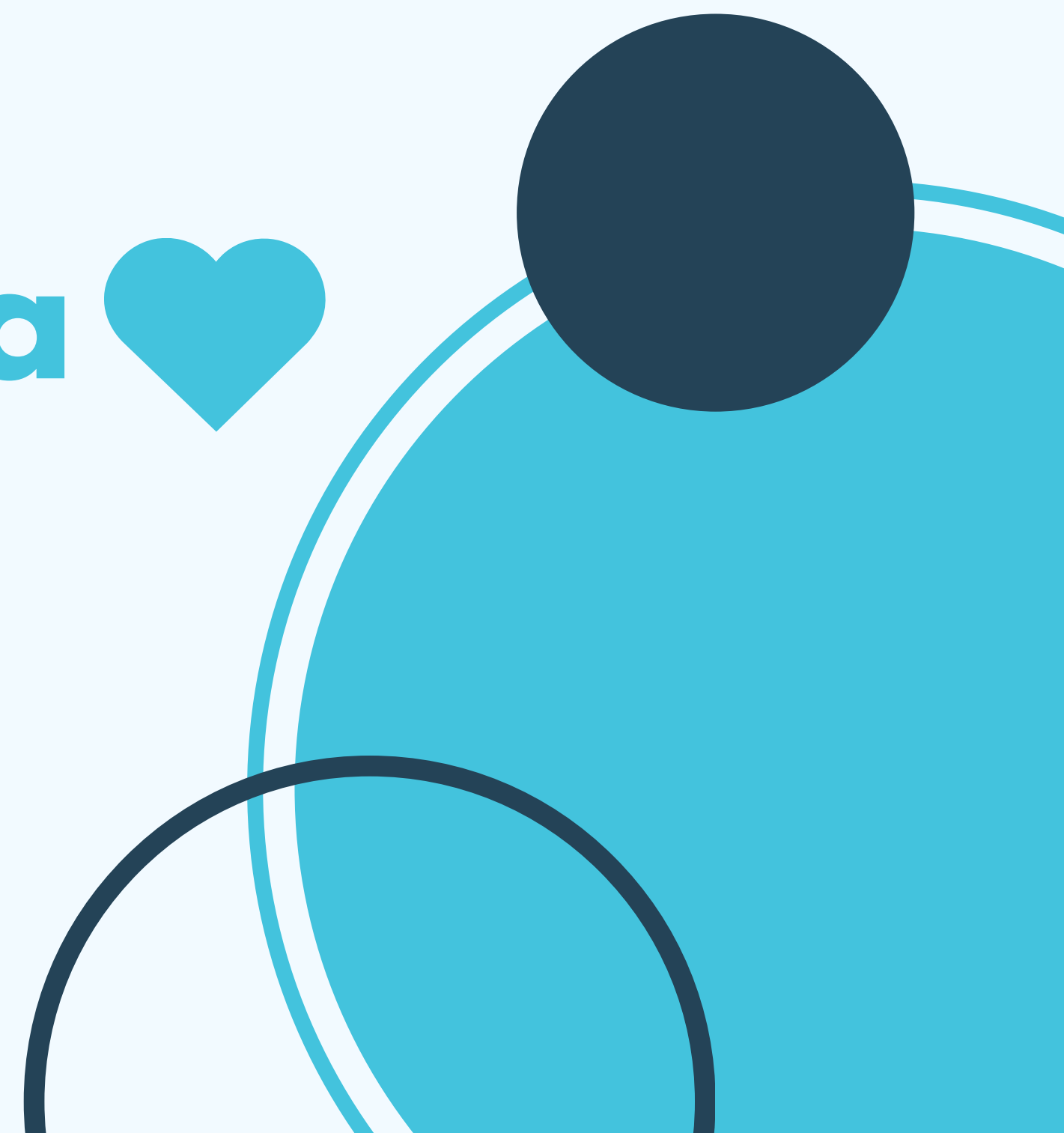


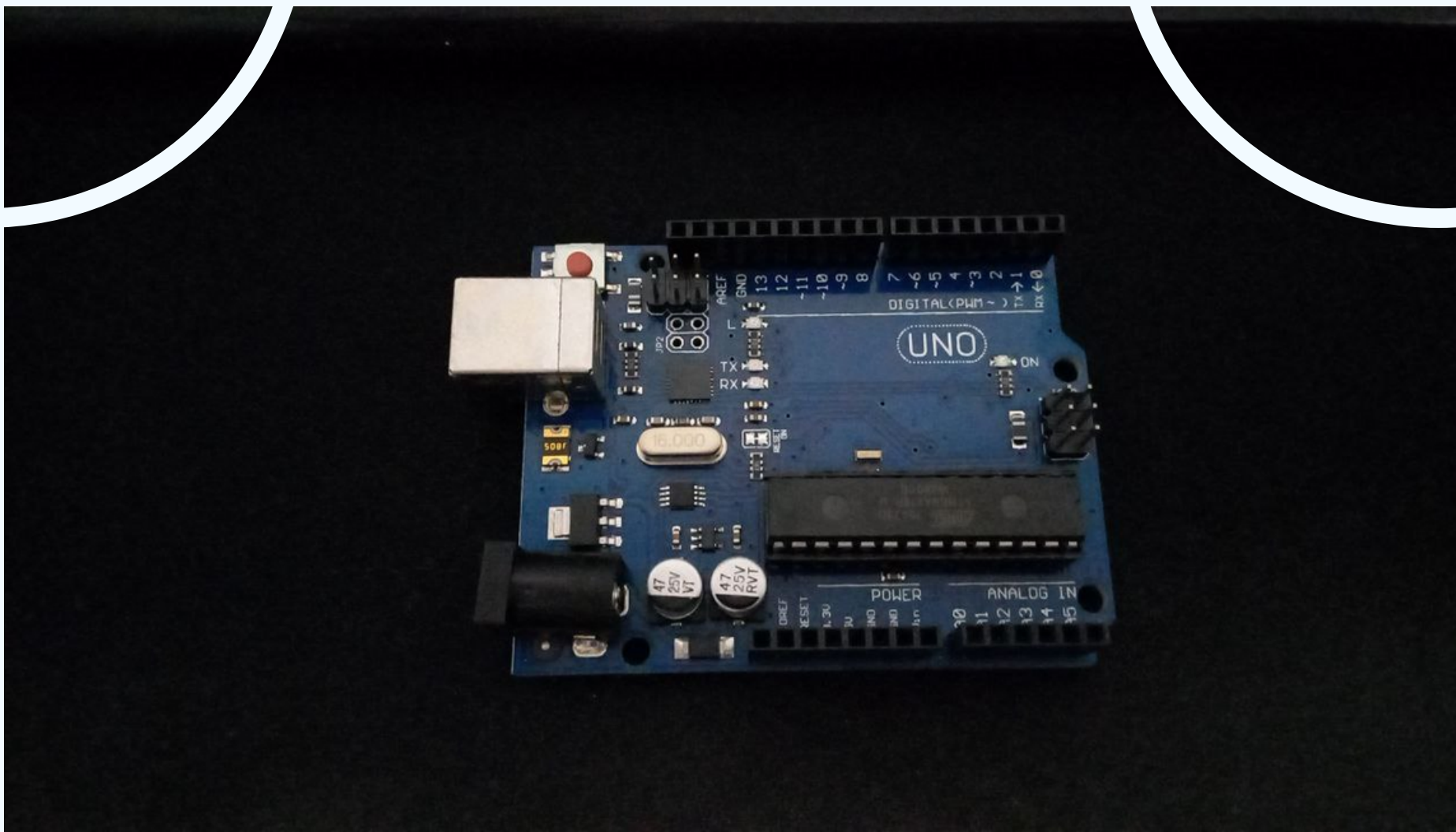
Prática Integrada - G5

Mateus Lopes da Silva

Ruan Mateus de Souza Nunes

Wendel Maxuel Ribeiro Pereira





SENSOR DE FREQUÊNCIA CARDÍACA

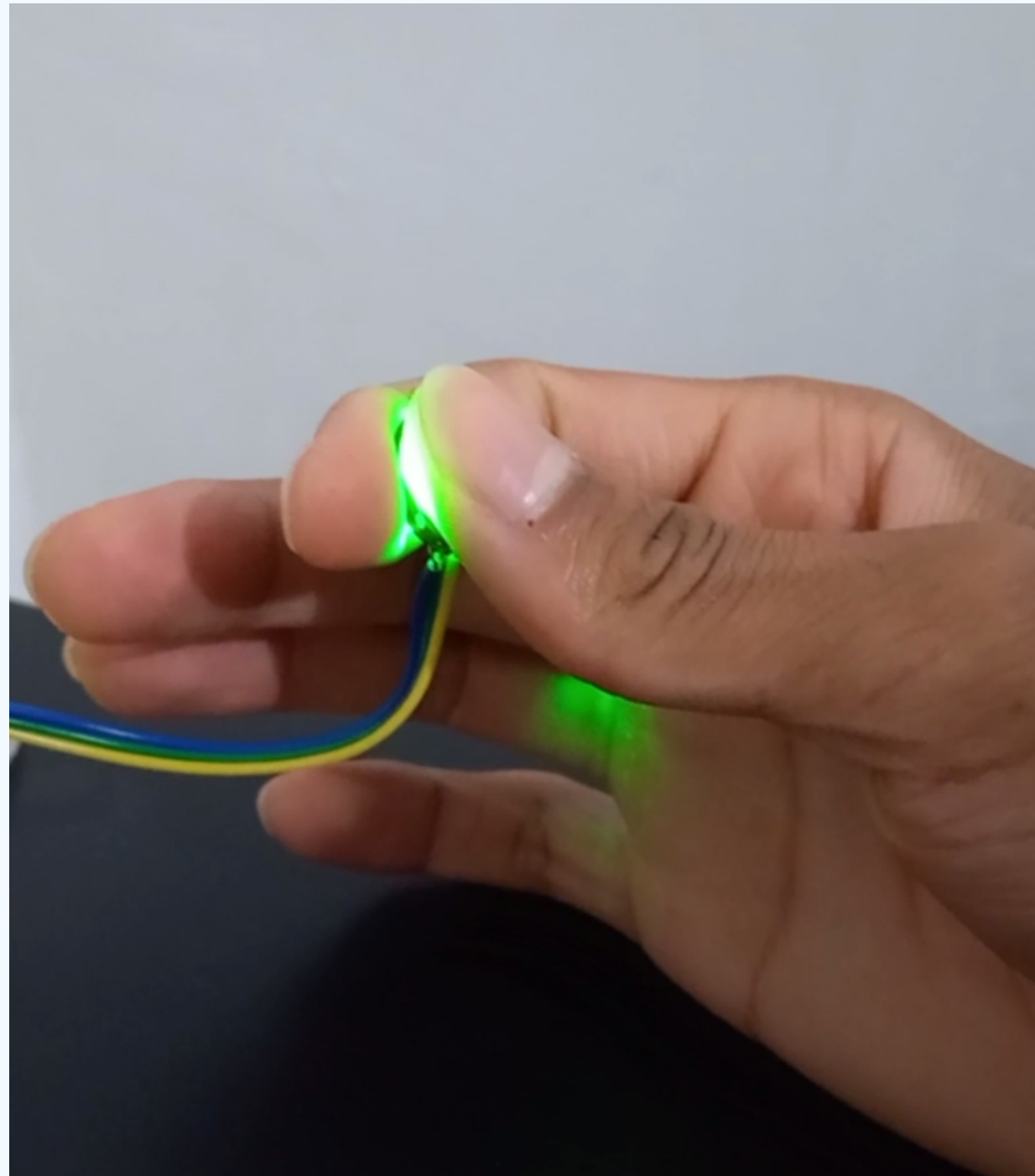
- Utilizado em conjunto com Arduino;
- Realiza a leitura da taxa de batimentos;
- Possui 3 conexões:
S = Sinal,
+ = Vcc (3,3V ou 5V) e
- = GND;

ARDUINO

- Plataforma de prototipagem eletrônica;
- Barato e funcional e simples de programar;
- Um minicomputador.



COMO É UTILIZADO



1. O sensor deve estar corretamente conectado ao Arduino;
2. O Arduino deve estar conectado a porta USB do Pc;
3. Com a IDE do Arduino previamente instalada, implementar o código e carregá-lo;
4. O sensor deve estar em contato com a ponta do dedo;
5. Por fim, testar o funcionamento do sensor na Serial Monitor ou Serial Plotter da IDE.

FUNCIONAMENTO DO SENSOR

sensorPulso | Arduino IDE 2.0.3

Arquivo Editar Rascunho Ferramentas Ajuda

Arduino Uno Mini

sensorPulso.ino

```
11 long entreBatidas = millis(); // Definindo a variável local que armazenará o tempo entre os batimentos
12 long tempoBPM = millis(); // Definindo a variável local "tempo de batimentos por minuto em milissegundos"
13
14 void setup() {
15     Serial.begin(9600); // Inicializando o monitor serial
16     Serial.println("BPM"); // Imprime no monitor serial a mensagem que está sendo passada como parâmetro
17 }
18
19 void loop() {
20     int valorLido = analogRead(pin); // Realizando a leitura do pino denominado A0
21
22     float valorFiltrado = fatorFiltro * valorAnterior + (1 - fatorFiltro) * valorLido; // Realizando a filtragem do sinal analógico
23     float valorDiferenca = valorFiltrado - valorAnterior; // Calculando a diferença entre a variavel valorFiltrado e valorAnterior
24
25     valorAnterior = valorFiltrado; // Atualizando a variavel valorAnterior com o valor da variável valorFiltrado
26
27     if ((valorDiferenca >= valorMaximo) && // Verificando se a variavel valorDiferenca é maior que a variável valorMaximo
28         (millis() > entreBatidas + minimoEntreBatidas)) // E se o tempo atual é maior que a soma do tempo da última batida detectada e o tempo mínimo entre as batidas
29     ) {
30         valorMaximo = valorDiferenca; // Armazena na variável valorMaximo o valor da variável valorDiferenca
31         entreBatidas = millis(); // Armazena o momento atual em milissegundos na variável entreBatidas
32         quantidadeBatidas++; // Incrementa mais um ao valor armazenado na variável quantidadeBatidas
33     }
34
35     valorMaximo = valorMaximo * 0.97; // Atualizando o valor da variável valorMaximo com 97% do valor da variavel valorDiferenca
36
37     if (millis() >= tempoBPM + 15000) { // Verificando se já se passaram 15 segundos para mostrar a quantidade de batimentos por minuto
38         Serial.println(quantidadeBatidas * 4); // Imprime no monitor serial a quantidade de batidas multiplicadas por 4, pois a cada 15 segundos temos 1/4 do minuto
39         tempoBPM = millis(); // Armazena o momento atual em milissegundos na variável tempoBPM
40         quantidadeBatidas = 0; // Atualizando a variavel quantidadeBatidas com o valor zero para iniciar uma nova contagem
41     }
42
43     delay(50); // Parando a execução do código por 50 milissegundos
44 }
45
```

Saída Monitor Serial x

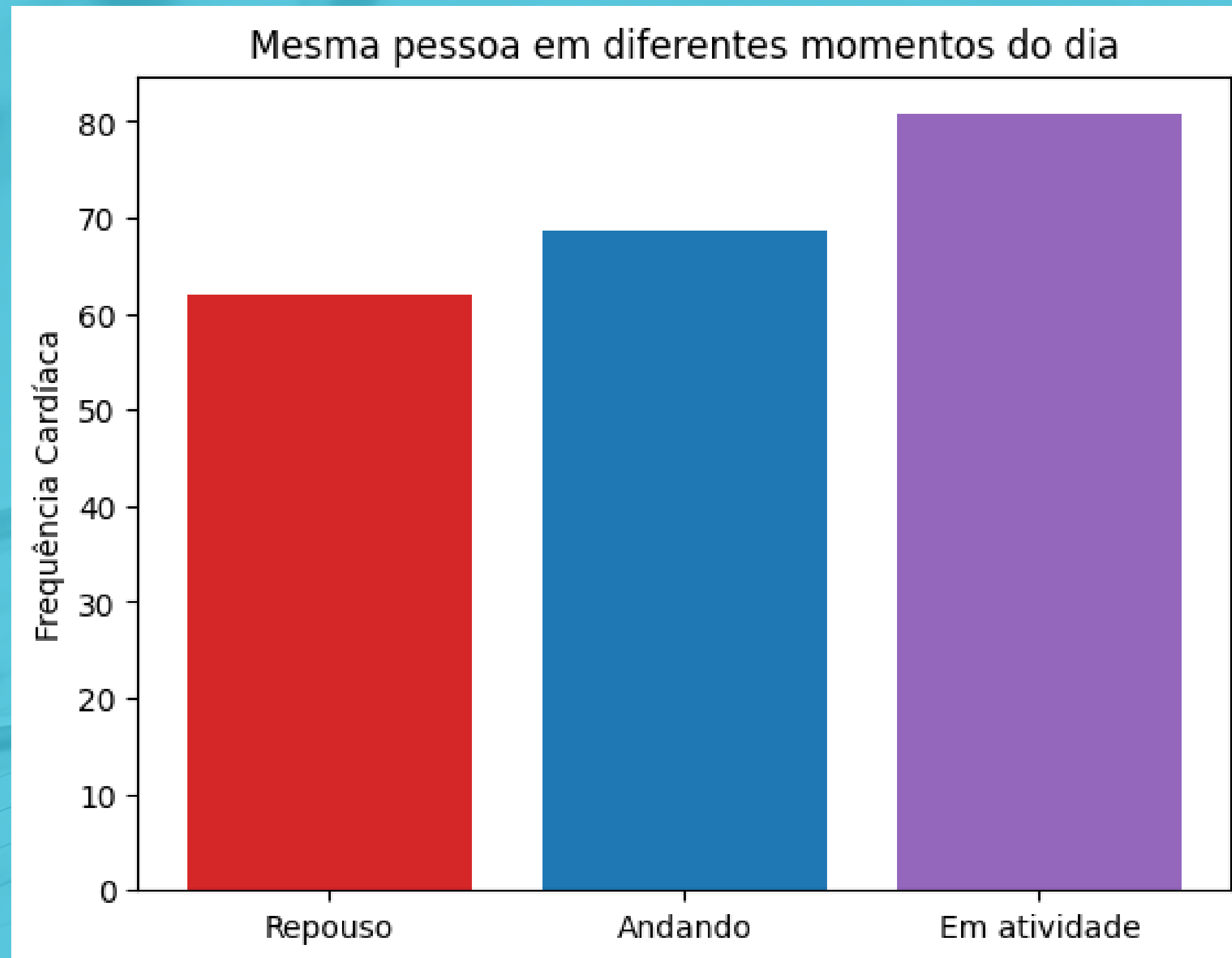
Message (Enter to send message to 'Arduino Uno Mini' on 'COM3')

Nova linha 9600 baud

10:49:40.772 -> BPM
10:49:55.802 -> 48
10:50:10.819 -> 60
10:50:25.834 -> 48
10:50:40.849 -> 52
10:50:55.850 -> 56
10:51:10.850 -> 48
10:51:25.897 -> 60

Ln 1, Col 1 UTF-8 Arduino Uno Mini em COM3 2

ANÁLISE DOS DADOS COLETADOS



REFERÊNCIAS

- <https://www.casadarobotica.com/sensores-modulos/sensores/outros/sensor-de-batimento-cardiaco-monitor-de-pulso>
- <https://arqia.com.br/post/arduino/>