

TECNOLOGIA EM SISTEMAS PARA INTERNET

Mateus Lopes da Silva
Ruan Mateus de Souza Nunes
Wendel Maxuel Ribeiro Pereira

**RELATÓRIO DE PRÁTICA INTEGRADA
DE
CIÊNCIA DE DADOS E INTERNET DAS COISAS**

Brasília - DF

04/01/2023

Sumário

1. Objetivos	3
2. Descrição do problema	4
3. Desenvolvimento	5
3.1 Código implementado	7
3.2 Link do repositório	11
4. Considerações finais	11
Referências	12

1. Objetivos

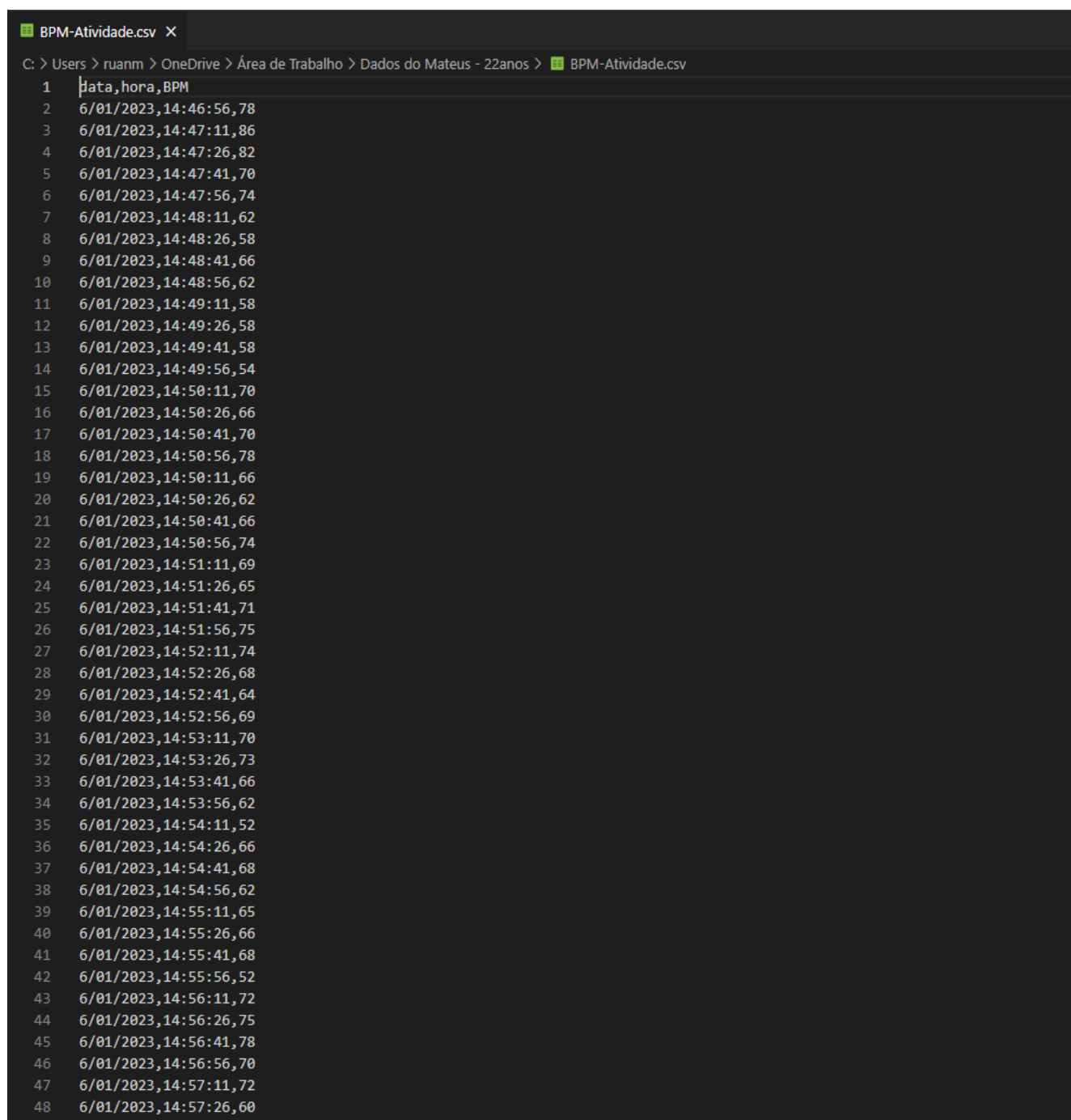
Nessa segunda sprint o objetivo foi coletar dados e analisá-los, com isso, nosso objetivo foi coletar diversos dados onde a frequência cardíaca da pessoa pode alterar como, por exemplo:(pessoa em repouso, andando, em atividade física, adultos sedentários, diferença entre adultos e jovens e crianças) identificando pelos resultados provenientes dos sensores, e depois analisá-los.

2. Descrição do problema

Na segunda sprint o problema foi utilizar o sensor nas coletas, pois tinha momentos em que o sensor não pegava a frequência ou seu resultado era fora da curva. Um problema da primeira sprint também veio para essa, a pouca comunicação e tempo, resultado das festas de fim de ano e calendário apertado do instituto, acarretando uma bola de neve, que vai se resolver para não ocorrer novos atrasos e virar uma bola de neve. Nessa sprint um dos componentes do grupo acabou desistindo do projeto.

3. Desenvolvimento

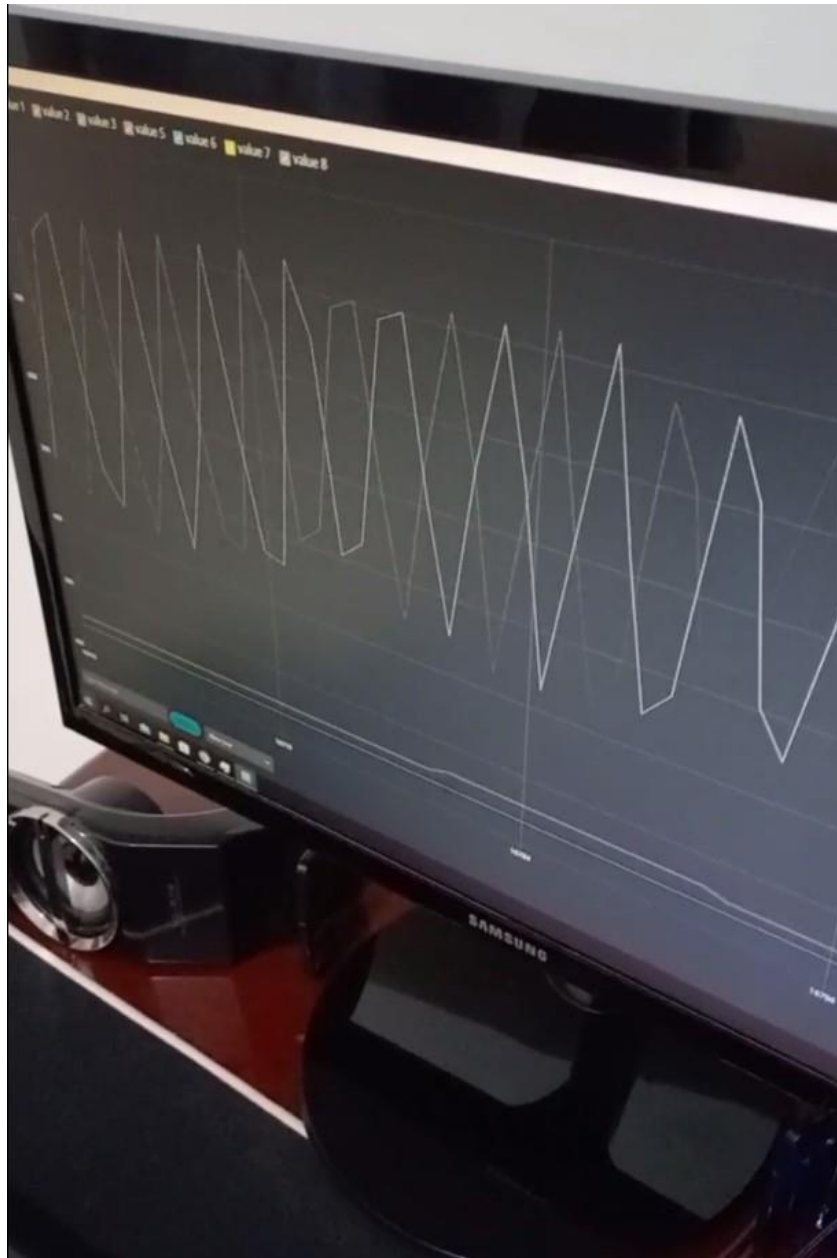
Utilizamos a linguagem de programação Python para fazer as duas etapas, coleta e preparação dos dados. Nessa primeira imagem, imagem 1, podemos ver o dado sendo transferido do arduino para o console, com isso já temos nossa primeira coleta. A Preparação do dado está sendo em CSV para depois, na próxima etapa, ser colocado em um gráfico.



```
BPM-Atividade.csv X
C: > Users > ruanm > OneDrive > Área de Trabalho > Dados do Mateus - 22anos > BPM-Atividade.csv
1 data,hora,BPM
2 6/01/2023,14:46:56,78
3 6/01/2023,14:47:11,86
4 6/01/2023,14:47:26,82
5 6/01/2023,14:47:41,70
6 6/01/2023,14:47:56,74
7 6/01/2023,14:48:11,62
8 6/01/2023,14:48:26,58
9 6/01/2023,14:48:41,66
10 6/01/2023,14:48:56,62
11 6/01/2023,14:49:11,58
12 6/01/2023,14:49:26,58
13 6/01/2023,14:49:41,58
14 6/01/2023,14:49:56,54
15 6/01/2023,14:50:11,70
16 6/01/2023,14:50:26,66
17 6/01/2023,14:50:41,70
18 6/01/2023,14:50:56,78
19 6/01/2023,14:50:11,66
20 6/01/2023,14:50:26,62
21 6/01/2023,14:50:41,66
22 6/01/2023,14:50:56,74
23 6/01/2023,14:51:11,69
24 6/01/2023,14:51:26,65
25 6/01/2023,14:51:41,71
26 6/01/2023,14:51:56,75
27 6/01/2023,14:52:11,74
28 6/01/2023,14:52:26,68
29 6/01/2023,14:52:41,64
30 6/01/2023,14:52:56,69
31 6/01/2023,14:53:11,70
32 6/01/2023,14:53:26,73
33 6/01/2023,14:53:41,66
34 6/01/2023,14:53:56,62
35 6/01/2023,14:54:11,52
36 6/01/2023,14:54:26,66
37 6/01/2023,14:54:41,68
38 6/01/2023,14:54:56,62
39 6/01/2023,14:55:11,65
40 6/01/2023,14:55:26,66
41 6/01/2023,14:55:41,68
42 6/01/2023,14:55:56,52
43 6/01/2023,14:56:11,72
44 6/01/2023,14:56:26,75
45 6/01/2023,14:56:41,78
46 6/01/2023,14:56:56,70
47 6/01/2023,14:57:11,72
48 6/01/2023,14:57:26,60
```

(Imagem 1 – Dados de batimentos cardíacos com data e hora, já armazenados em um arquivo CSV pelo Python no VSCode.

Nessa segunda imagem podemos ver como fica a tela do console com o sensor de frequência cardíaca em funcionamento, basta a pessoa colocar o sensor em sua pele, em um ponto que posso medir sua pulsação.



(Imagem 2 - Tela com sensor em contato com a pele)

Também estamos utilizando um dispositivo que também tem um sensor de frequência cardíaca para comparar os resultados, mostrado na imagem 1, isso está ajudando bastante na hora de efetuar as análises.



(Imagem 3 - Comparando os valores)

3.1 Código implementado

Os códigos que utilizamos durante todo o projeto e o referencial para o GitHub onde está o código:

- **Código Python para criação do csv:**

```
import serial    # importe da biblioteca Pyserial
```

```
from datetime import datetime    # importe do módulo datetime
```

```
porta = "COM3"    # Definindo a variável porta no qual o arduino está conectado. Obs.: COM3  
e a porta USB do PC mostrado pela IDE do arduino
```

```
baud = 9600      # Definindo a taxa de dados chamada de "baud rate" com o valor armazenado
                  em bits por segundo
```

```
arquivo = "BPM-EsforcoFisico.csv"  # Definindo o arquivo csv que irá receber os dados
```

```
ser = serial.Serial(porta, baud)    # Inicializando o objeto serial com parâmetros
```

```
ser.flushInput()                    # Limpando dados da serial
```

```
print("Abrindo Serial")             # Imprime a mensagem em parâmetro
```

```
amostra = 100      # Definindo o valor de amostras que se deseja obter na variável amostra
```

```
linha = 0           # Definindo o controle de linha com a variável linha
```

```
while linha <= amostra:              # loop para ler os dados
```

```
data = str(ser.readline().decode("utf-8"))  # Lendo os dados (Batimentos) enviados do
                                              arduino e transformando para string
```

```
file = open(arquivo, "a")             # Abrindo o arquivo csv com parâmetro de criação
```

```
now = datetime.now()                  # Mostra a data e hora atual em tempo de execução
```

```
dataHora = now.strftime("%d/%m/%Y,%H:%M:%S")  # Armazenando a data e hora
                                              atual na variável dataHora formatado dessa forma, exemplo: 01/01/2000,01:01:01
```

```
print(data)                           # Imprime no terminal o dado do Batimento Cardíaco enviado pelo arduino
```

```
file.write(dataHora + "," + data)      # Adiciona os dados ao arquivo csv
```

```
linha = linha + 1                      # Iteração da variável linha
```

```
print("Fim da Leitura!")              # Imprime a mensagem em parâmetro
```

```
file.close()                           # Fecha o arquivo csv
```

```
ser.close()                            # Fecha a serial
```


- **Código da coleta de dados (Batimentos cardíacos) no Arduino:**

```
int pin = A0;      // Definido o pino A0 como "pin"

float valorAnterior = 0;    // Definindo a variável que armazenaráo valor da leitura anterior
float valorMaximo = 0.0;    // Definindo a variável que armazenará97% do valor máximo
                             obtido
int quantidadeBatidas = 0;  // Definindo a variável que armazenaráa quantidade de
                             batimentos

float fatorFiltro = 0.75;   // Coeficiente para o filtro do valor analógico
                             obtido durante a leitura
int minimoEntreBatidas = 300;      // Valor mínimo de tempo entre os
                                     batimentos cardíacos

long entreBatidas = millis();      // Definindo a variável local quearmazenará o tempo
                                     entre os batimentos

long tempoBPM = millis();          // Definindo a variável local “tempo debatimentos por
                                     minuto” em milissegundos

void setup() {
    Serial.begin(9600);    // Inicializando o monitor serial

    Serial.println("BPM"); // Imprime no monitor serial a mensagem que está
                             sendo passada como parâmetro
}

void loop() {
    int valorLido = analogRead(pin); // Realizando a leitura do pinodenominado A0

    float valorFiltrado = fatorFiltro * valorAnterior + (1 - fatorFiltro) *valorLido;
        // Realizando a filtragem do sinal analógico

    float valorDiferenca = valorFiltrado - valorAnterior;    // Calculando a diferença
        entre a variável valorFiltrado e valorAnterior
```

```
valorAnterior = valorFiltrado;           // Atualizando a variável valorAnterior com o valor da
variável valorFiltrado
```

```
if ((valorDiferenca >= valorMaximo) &&    // Verificando se a variável valorDiferenca
    é maior que a variável valorMaximo
```

```
(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
) {
```

```
    valorMaximo = valorDiferenca;           // Armazena na variável valorMaximo o
    valor da variável valorDiferenca
```

```
    entreBatidas = millis();               // Armazena o momento atual em milissegundos
    na variável entreBatidas
```

```
    quantidadeBatidas++;                   // Incrementa mais um ao valor armazenado na
    variável quantidadeBatidas
}
```

```
valorMaximo = valorMaximo * 0.97;         // Atualizando o valor da variável
    valorMaximo com 97% do valor da variável valorDiferenca
```

```
if (millis() >= tempoBPM + 15000) {      // Verificando se já se passaram 15 segundos para
    mostrar a quantidade de batimentos por minuto
```

```
    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
```

```
    tempoBPM = millis();                     // Armazena o momento atual em milissegundos na
    variável tempoBPM
```

```
    quantidadeBatidas = 0;                   // Atualizando a variável quantidadeBatidas com o
    valor zero para iniciar uma nova contagem
```

```
    }
    delay(50);                               // Parando a execução do código por 50 milissegundos
}
```

3.2 Link do repositório

Os dados coletados estão localizados no GitHub do projeto, no link:

- <https://github.com/infocbra/pratica-integrada-cd-e-ic-2022-2-g5-rwmv>

4. Considerações finais

Encontramos muita dificuldade para terminar a montagem do hardware, deixando as coisas para última hora e dificultando nos próximos passos, melhoramos isso e no final dessa sprint. A coleta e análise de dados, ficaram atrasadas, poderíamos melhorar adiantando as coisas para não acontecer mais atrasos.

Essa sprint foi desorganizada, demos o melhor como grupo e as coisas estão acontecendo. A falta de um integrante é muito grande, o trabalho poderia ser bem mais fácil com mais um elemento, mas estamos conseguindo. Mas nas próximas sprints as tarefas serão entregues no dia certo.