



INSTITUTO FEDERAL

Brasília

Campus Brasília

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

10/01/2023

Sumário

1. Objetivos	3
2. Descrição do problema	4
3. Desenvolvimento	7
3.1 Armazenamento	7
3.2 Código Implementado	8
3.3 Arquivos CSVs coletados	11
4. Considerações finais	12
Referências	13

1. Objetivos

O objetivo dessa terceira sprint foi de analisar os dados e fazer um gráfico para conseguirmos ver os resultados que foram adquiridos na sprint passada. Agora, por meio de gráficos, mostraremos os resultados obtidos para melhor entendimento daqueles que estão observando os dados. O gráfico com sua legenda, é um meio de simplificar e ajudar a entender o que são os números que foram retirados do sensor de frequência, pois sem ele, seria apenas números, não teriam como uma pessoa entender o que seriam.

2. Descrição do problema

Nosso problema nessa etapa foi a transformação do dado em gráfico, que parece ser algo simples, mas foi trabalhoso pela quantidade de dados e análises que fizemos para no final fazer um gráfico. Também encontramos alguns problemas para a retirada de alguns dados, por exemplo, o dado correndo, que poderíamos ter usado um powerbank de fonte de energia para o arduino, e não ter que usar o computador de fonte de energia. Com o atraso da sprint anterior, ficamos com pouco tempo para essa, mas com o esforço de todos, conseguimos fazer tudo no prazo e não deixar acumular, e entregando o trabalho do jeito que queríamos.

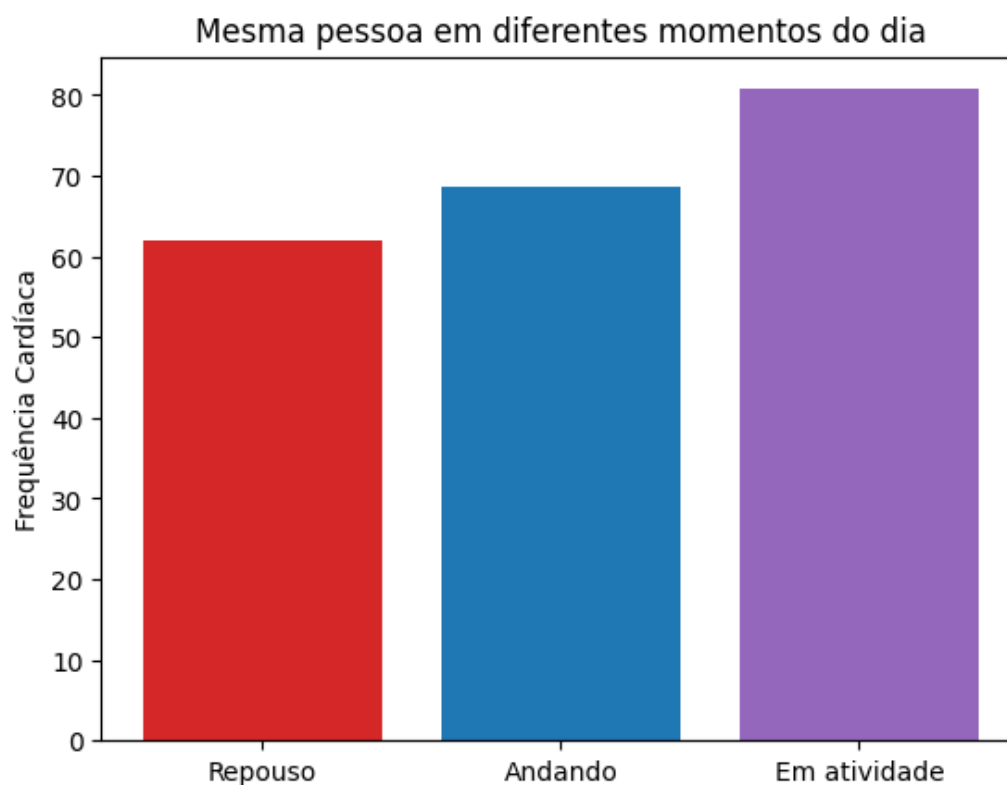
3. Desenvolvimento

Utilizamos a tecnologia do Python e do Google Colab, para fazer a montagem dos gráficos.

Desconsideramos os pontos fora da curva, os pontos muito baixos e muito altos. Os pontos muito baixo, depois de vários dados, ficou evidente que foi o sensor que não estava encostando completamente na superfície de contato, vimos que logo após encostar completamente os dados não apareceram mais. Os pontos muito alto, desconsideramos por se tratar de uma anomalia, um bug do sensor, que de vez em quando aparecia, pois se tratava de um número muito alta e a pessoas que esta sendo medida, falava que não tinha sentido alteração. Utilizamos as médias dos dados para serem colocados no gráfico.

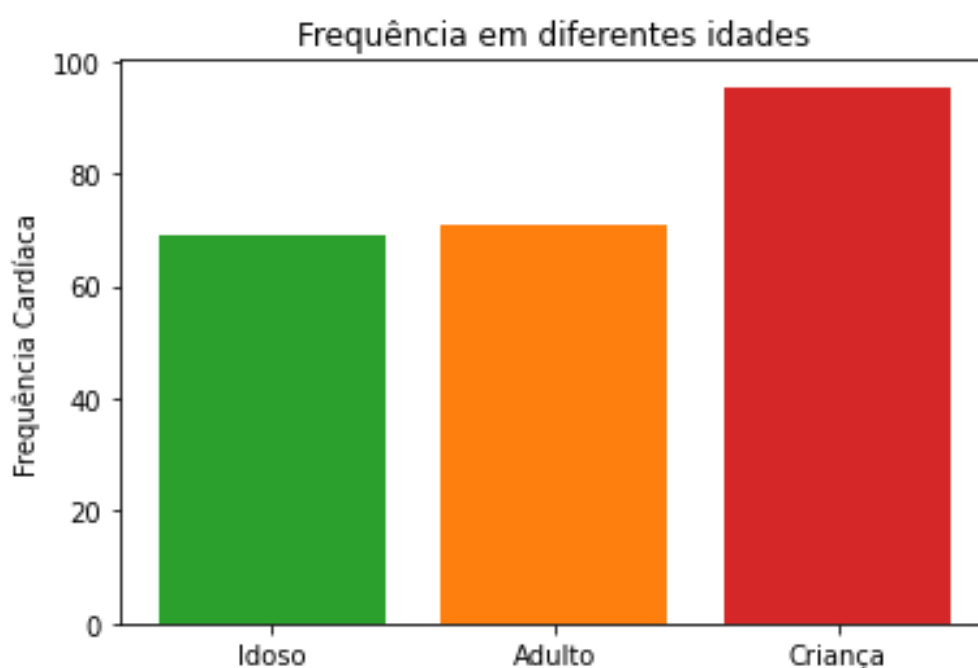
Coisas que podem alterar os dados, como temperatura, fazendo a frequência aumentar em dias frios e diminuir em dias quentes, se a pessoa está emocionalmente bem, foram consideradas e filtradas.

Na primeira imagem, imagem 1, que está logo abaixo, pegamos os dados de uma pessoa e analisamos ela por diversos momentos do seu dia, com isso podemos ver como a frequência vai se alterando ao decorrer do dia e do esforço que a pessoa está fazendo.



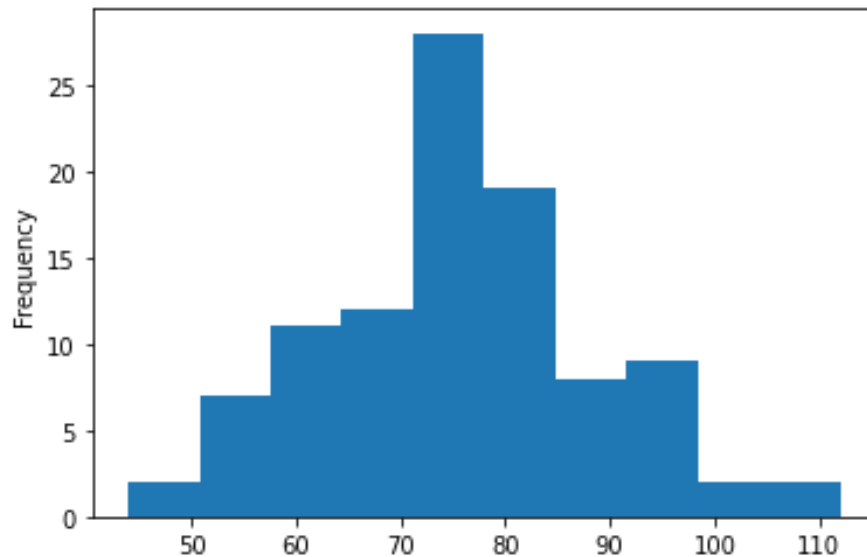
(Imagem 1 - Mesma pessoa em diferentes momentos do dia)

Aqui podemos diferenciar o tanto que frequência muda dependendo da sua atividade, do seu esforço e abaixo podemos observar na imagem 2, o quanto a idade também altera a sua frequência, por meio de análise, percebemos que frequência e idade são inversamente proporcionais, quanto mais novo, maior será sua frequência cardíaca.



(Imagem 2 - Frequência em diferentes idades)

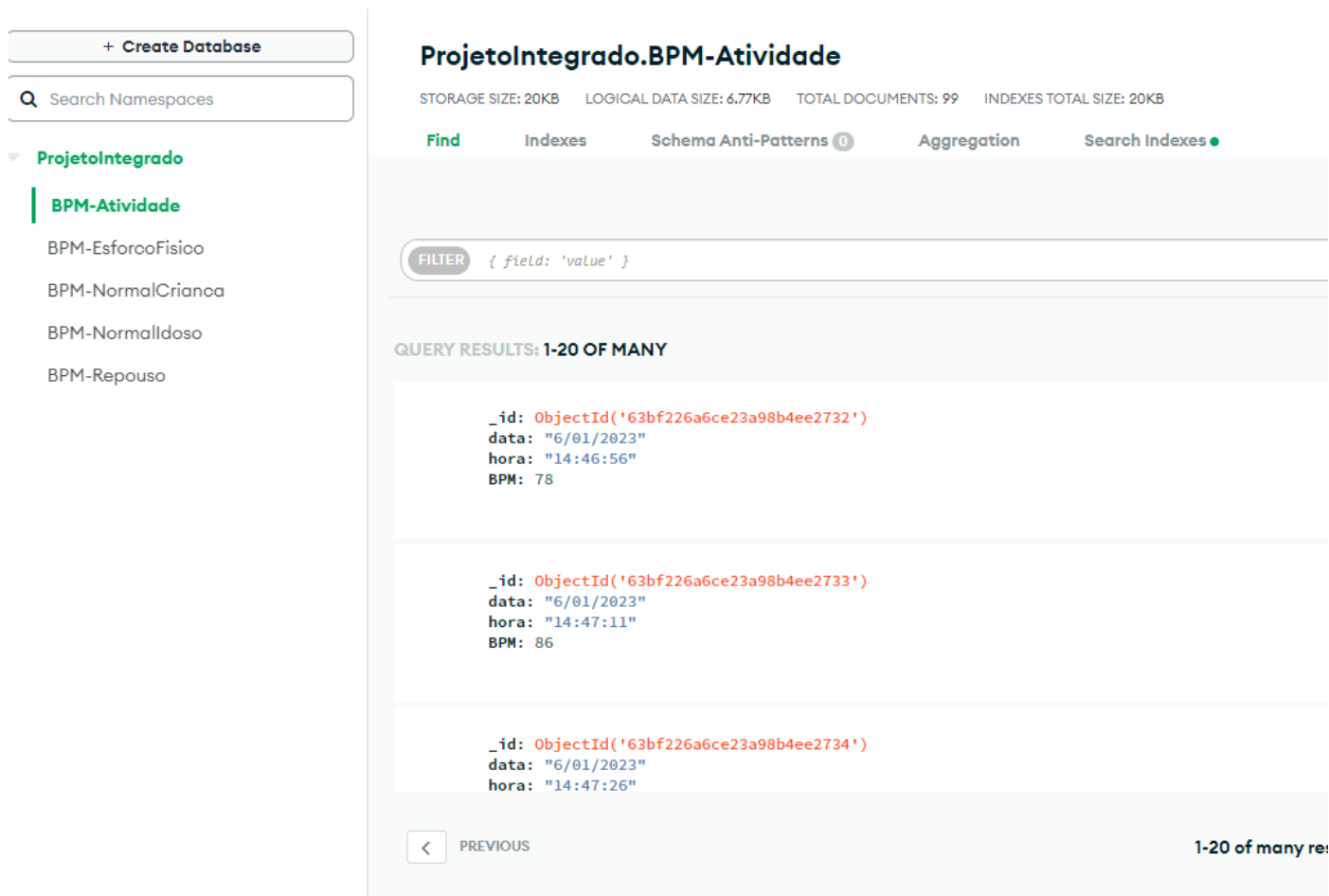
Nesse terceiro gráfico, podemos ver todos os dados do csv de esforço físico de uma determinada pessoa, com o eixo X mostrando a frequência e o Y quantas vezes aquela frequência foi repetida, com isso podemos ver os pontos fora da curva e a concentração dos outros pontos aonde mais ocorreram aquelas frequências.



(Imagem 3 - Analisando dados do CSV)

3.1 Armazenamento

Para armazenar todos essas dados que foram coletados na sprint passada, escolhemos o SGBD da MongoDB, por ser um banco de dados não relacional, como não terá relacionamento entre as tabelas do banco, ele se encaixou perfeitamente. Usamos o Google Colab com o Pymongo para efetuar esse armazenamento. Logo abaixo na imagem 4, podemos ver o banco MongoDB com os csvs armazenados. Os códigos usados para realizar tanto o transporte, quanto a criação dos gráficos está logo abaixo, no item 3.2.



(imagem 4 - Dados armazenados)

3.2 Código implementado

Aqui está os códigos dos gráficos que utilizamos no Google Colab.

- Imagem 1:

```
import matplotlib.pyplot as plt          #Importando a biblioteca matplotlib

fig, ax = plt.subplots()                #definindo uma função que retornará uma figura

estado = ['Repouso', 'Andando', 'Em atividade']    #Definindo o nome das
variáveis que ficaram no eixo X mais para frente
frequencia = [64, 70.71, 75.72,]    #Média dos valores coletados pelo CSV de cada
variável
bar_colors = ['tab:red', 'tab:blue', 'tab:purple' ]    #Definindo a cor das figuras

ax.bar(estado, frequencia, color=bar_colors)    #Colocando as variáveis no eixoX
```



```
ax.set_ylabel('Frequência Cardíaca')      #Colocando nome para o eixoY
ax.set_title('Mesma pessoa em diferentes momentos do dia') #Título do gráfico

plt.show()                                #fechamento da biblioteca
```

```
import pandas as pd                        #Importando a biblioteca pandas

batimentos = pd.read_csv("BPM-Repouso.csv") #Declarando batimentos do CSV
batimentos.mean()                          #Pegando a média para usar no código acima

batimentos = pd.read_csv("BPM-Atividade.csv") #Declarando batimentos do CSV
batimentos.mean()                          #Pegando a média para usar no código acima

batimentos = pd.read_csv("BPM-EsforcoFisico.csv") #Declarando batimentos do CSV
batimentos.mean()                          #Pegando a média para usar no código acima
```

- Imagem 2:

```
import matplotlib.pyplot as plt           #Importando a biblioteca matplotlib

fig, ax = plt.subplots()                  #definindo uma função que retornará uma figura

estado = ['Idoso', 'Adulto', 'Criança']  #Definindo o nome das variáveis que ficaram no
eixo X mais para frente
frequencia = [68.83, 70.71, 95.37,]      #Média dos valores coletados pelo CSV de cada
variável
bar_colors = ['tab:green', 'tab:orange', 'tab:red' ] #Definindo a cor das figuras

ax.bar(estado, frequencia, color=bar_colors) #Colocando as variáveis no eixoX

ax.set_ylabel('Frequência Cardíaca')      #Colocando nome para o eixoY
ax.set_title('Frequência em diferentes idades') #Título do gráfico

plt.show()                                #fechamento da biblioteca
```

```

import pandas as pd                                #Importando a biblioteca pandas
batimentos = pd.read_csv("BPM-Atividade.csv")      #Declarando batimentos do CSV
batimentos.mean()                                  #Pegando a média para usar no código acima

batimentos = pd.read_csv("BPM-NormalCrianca.csv")  #Declarando batimentos do CSV
batimentos.mean()                                  #Pegando a média para usar no código acima

batimentos = pd.read_csv("BPM-NormalIdoso.csv")    #Declarando batimentos do CSV
batimentos.mean()                                  #Pegando a média para usar no código acima

```

- Imagem 3:

```

import pandas as pd                                #importando a biblioteca pandas

batimentos = pd.read_csv("BPM-EsforcoFisico.csv") #definindo batimentos do csv
batimentos.columns = ["data", "hora", "BPM",]      #Observando as colunas do csv
batimentos['BPM'].unique()                          #Mostrando apenas o BPM e definindo
com batimentos
batimentos.BPM.plot(kind='hist')                   #Colocando os batimentos em um gráfico do tipo hist

```

- Código usado para armazenar os dados em csv no banco:

```

import pymongo                                     #Estamos importando a biblioteca do pymongo
import pandas as pd                                #Estamos importando a biblioteca do pandas

myclient=pymongo.MongoClient("mongodb+srv://admin:CGC2kNOmZ5gcGbdj@
cluster0.tkktqcr.mongodb.net/?retryWrites=true&w=majority") #Colocando o local
onde esta localizado o banco, para os códigos do colab terem um endereço

mydb = myclient["ProjetoIntegrado"]                #Nome do database

mycol = mydb["BPM-Repouso"]                         #Nome da coleção

df = pd.read_csv("BPM-Repouso.csv")                 #Variável onde o código vai ler os dados

```

```
mylist = df.to_dict('records') #Função transporta e adiciona os dados no daatabase
```

```
x = mycol.insert_many(mylist) #Função para adicionar uma grande quantidade  
de dados no database
```

```
print(x.inserted_ids) #Print dos ids que foram adicionados
```

3.3 Arquivos CSVs coletados

Os dados coletados estão localizados no GitHub e do Colab do projeto, no:

- <https://colab.research.google.com/drive/1n4wrPfBm5fVvhcUNcGmbe2--nhivzAt8#scrollTo=nmjLTrr7L6p3>
- <https://github.com/infocbra/pratica-integrada-cd-e-ic-2022-2-g5-rwmv>

4. Considerações finais

Encontramos diversas dificuldades ao decorrer do projeto, como montar um arduino, o código para fazer o mesmo funcionar, achar um sensor de frequência cardíaca, fazer o sensor e o arduino conversarem, transportar o resultado para um armazenamento, coletar diversos tipos de dados. E no final conseguimos chegar e gostamos do resultado, deu muito trabalho, mas foi gratificante no final em saber tudo que passamos. Gostamos demais do projeto, pois no curso não tem nada parecido com o que tivemos, no curso os projetos são mais teóricos ou fictícios, mas nesse realmente desenvolvemos as coisas e fizemos as coisas acontecerem. No começo estávamos muito perdidos, mas ao decorrer foi nos encontrando e tudo foi fluindo naturalmente.

O nosso principal e mais repetitivo problema, foi a falta de comunicação adequada e o atraso das entregas, mas foi consertada no final. E o que ficou foi o sentimento de alegria de conseguir concluir o projeto e de superação pelos diversos momentos difíceis que passamos.

Referências

- https://www.w3schools.com/python/python_mongodb_getstarted.asp
- <https://www.w3schools.com/python/pandas/default.asp>
- https://www.w3schools.com/python/matplotlib_intro.asp