



## **Missão Prática | Nível 3 | Mundo 5**

João Gabriel Cesconetto - 202208324053

### **Tratando a Imensidão dos Dados**

Objetivos da prática:

- 1) Descrever como ler um arquivo CSV usando a biblioteca Pandas;**
- 2) Descrever como criar um subconjunto de dados a partir de um conjunto existente;**
- 3) Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados;**
- 4) Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados;**
- 5) Descrever como exibir informações gerais sobre as colunas, linhas e dados de um conjunto de dados.**

## Microatividade 1: Descrever como ler um arquivo CSV usando a biblioteca Pandas

```
[2] import pandas as pd
# Variável contendo o caminho do arquivo CSV
file_path = '/content/data.csv'
# Ler o conteúdo do arquivo CSV
dados = pd.read_csv(file_path, sep=';', engine='python')

print(dados)
```

Saída:


	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

**Microatividade 2: Descrever como criar um subconjunto de dados a partir de um conjunto existente usando a biblioteca Pandas**

```
[3] # Variável contendo subconjunto de colunas, selecionando ID, duração e calorias
    subconjunto_dados = dados[['ID', 'Duration', 'Calories']]

    print(subconjunto_dados)
```

Saída:




	ID	Duration	Calories
0	0	60	4091
1	1	60	4790
2	2	60	3400
3	3	45	2824
4	4	45	4060
5	5	60	3000
6	6	60	3740
7	7	450	2533
8	8	30	1951
9	9	60	2690
10	10	60	3293
11	11	60	2507
12	12	60	2507
13	13	60	3453
14	14	60	3793
15	15	60	2750
16	16	60	2152
17	17	60	3000
18	18	45	NaN
19	19	60	3230
20	20	45	2430 2
21	1	60	3642
22	22	45	2820
23	23	60	3000
24	24	45	2460
25	25	60	3345
26	26	60	2500
27	27	60	2410
28	28	60	NaN
29	29	60	2800
30	30	60	3803
31	31	60	2430

### Microatividade 3: Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados usando a biblioteca Pandas

```
[4] # Define o novo valor para a propriedade max_rows
pd.set_option('display.max_rows', 9999)

print(dados.to_string())
```

Saída:



	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

**Microatividade 4: Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados usando a biblioteca Pandas**

```
[5] print("Primeiras 10 linhas:")
    print(dados.head(10))

    print("\nÚltimas 10 linhas:")
    print(dados.tail(10))
```

Saída:

```
⇒ Primeiras 10 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
0   0         60 '2020/12/01'   110       130     4091
1   1         60 '2020/12/02'   117       145     4790
2   2         60 '2020/12/03'   103       135     3400
3   3         45 '2020/12/04'   109       175     2824
4   4         45 '2020/12/05'   117       148     4060
5   5         60 '2020/12/06'   102       127     3000
6   6         60 '2020/12/07'   110       136     3740
7   7        450 '2020/12/08'   104       134     2533
8   8         30 '2020/12/09'   109       133     1951
9   9         60 '2020/12/10'    98       124     2690

Últimas 10 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
22  22         45      NaN    100       119     2820
23  23         60 '2020/12/23'   130       101     3000
24  24         45 '2020/12/24'   105       132     2460
25  25         60 '2020/12/25'   102       126     3345
26  26         60 20201226    100       120     2500
27  27         60 '2020/12/27'    92       118     2410
28  28         60 '2020/12/28'   103       132      NaN
29  29         60 '2020/12/29'   100       132     2800
30  30         60 '2020/12/30'   102       129     3803
31  31         60 '2020/12/31'    92       115     2430
```

**Microatividade 5: Descrever como exibir informações gerais sobre as colunaslinhas e dados de um conjunto de dados usando a biblioteca Pandas**

```
[6] print("Informações gerais do conjunto de dados:")
    dados.info()

    total_linhas = dados.shape[0]
    total_colunas = dados.shape[1]
    quantidade_nulos = dados.isnull().sum()
    tipos_dados = dados.dtypes
    memoria_utilizada = dados.memory_usage(deep=True).sum()

    print(f"\nTotal de linhas: {total_linhas}")
    print(f"Total de colunas: {total_colunas}")
    print(f"Quantidade de dados nulos:\n{quantidade_nulos}")
    print(f"Tipos de dados de cada coluna:\n{tipos_dados}")
    print(f"Quantidade de memória utilizada: {memoria_utilizada} bytes")
```

Saída:

```
↔ Informações gerais do conjunto de dados:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   ID          32 non-null    int64
 1   Duration    32 non-null    int64
 2   Date        31 non-null    object
 3   Pulse       32 non-null    int64
 4   Maxpulse    32 non-null    int64
 5   Calories    30 non-null    object
dtypes: int64(4), object(2)
memory usage: 1.6+ KB

Total de linhas: 32
Total de colunas: 6
Quantidade de dados nulos:
ID          0
Duration    0
Date        1
Pulse       0
Maxpulse    0
Calories    2
dtype: int64
Tipos de dados de cada coluna:
ID          int64
Duration    int64
Date        object
Pulse       int64
Maxpulse    int64
Calories    object
dtype: object
Quantidade de memória utilizada: 5215 bytes
```

## Missão Prática

```
[8] # Variável contendo a cópia dos dados
    dados_copia = dados.copy()

    # Substituição de todos os valores nulos da coluna 'Calories' por 0
    dados_copia['Calories'].fillna(0, inplace=True)

    print(dados_copia)
```

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	0
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	0
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

```
[9] # Substituição de todos os valores nulos da coluna 'Date' por '1900/01/01'
dados_copia['Date'].fillna('1900/01/01', inplace=True)

print(dados_copia)
```

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	0
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	1900/01/01	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	0
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

```
# Transforma os dados da coluna 'Date' em datetime
dados_copia['Date'] = pd.to_datetime(dados_copia['Date'])

dados_copia['Date'].replace('1900/01/01', pd.NA, inplace=True)

dados_copia['Date'] = pd.to_datetime(dados_copia['Date'], errors='coerce')

print(dados_copia)
```





	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	2020-12-01	110	130	4091
1	1	60	2020-12-02	117	145	4790
2	2	60	2020-12-03	103	135	3400
3	3	45	2020-12-04	109	175	2824
4	4	45	2020-12-05	117	148	4060
5	5	60	2020-12-06	102	127	3000
6	6	60	2020-12-07	110	136	3740
7	7	450	2020-12-08	104	134	2533
8	8	30	2020-12-09	109	133	1951
9	9	60	2020-12-10	98	124	2690
10	10	60	2020-12-11	103	147	3293
11	11	60	2020-12-12	100	120	2507
12	12	60	2020-12-12	100	120	2507
13	13	60	2020-12-13	106	128	3453
14	14	60	2020-12-14	104	132	3793
15	15	60	2020-12-15	98	123	2750
16	16	60	2020-12-16	98	120	2152
17	17	60	2020-12-17	100	120	3000
18	18	45	2020-12-18	90	112	0
19	19	60	2020-12-19	103	123	3230
20	20	45	2020-12-20	97	125	2430 2
21	1	60	2020-12-21	108	131	3642
22	22	45	NaT	100	119	2820
23	23	60	2020-12-23	130	101	3000
24	24	45	2020-12-24	105	132	2460
25	25	60	2020-12-25	102	126	3345
26	26	60	NaT	100	120	2500
27	27	60	2020-12-27	92	118	2410
28	28	60	2020-12-28	103	132	0
29	29	60	2020-12-29	100	132	2800
30	30	60	2020-12-30	102	129	3803
31	31	60	2020-12-31	92	115	2430

```
# Substituí o valor "20201226" no formato correto
dados_copia['Date'] = dados_copia['Date'].astype(str).replace('20201226', '2020/12/26')

dados_copia['Date'] = pd.to_datetime(dados_copia['Date'], errors='coerce')

print("\nApós transformar a coluna 'Date' para datetime:")
print(dados_copia)

# Remove os registros contendo valores nulos
dados_copia.dropna(inplace=True)

print("\nDataframe final após remover registros com valores nulos:")
print(dados_copia)
```

Após transformar a coluna 'Date' para datetime:

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	2020-12-01	110	130	4091
1	1	60	2020-12-02	117	145	4790
2	2	60	2020-12-03	103	135	3400
3	3	45	2020-12-04	109	175	2824
4	4	45	2020-12-05	117	148	4060
5	5	60	2020-12-06	102	127	3000
6	6	60	2020-12-07	110	136	3740
7	7	450	2020-12-08	104	134	2533
8	8	30	2020-12-09	109	133	1951
9	9	60	2020-12-10	98	124	2690
10	10	60	2020-12-11	103	147	3293
11	11	60	2020-12-12	100	120	2507
12	12	60	2020-12-12	100	120	2507
13	13	60	2020-12-13	106	128	3453
14	14	60	2020-12-14	104	132	3793
15	15	60	2020-12-15	98	123	2750
16	16	60	2020-12-16	98	120	2152
17	17	60	2020-12-17	100	120	3000
18	18	45	2020-12-18	90	112	0
19	19	60	2020-12-19	103	123	3230
20	20	45	2020-12-20	97	125	2430 2
21	1	60	2020-12-21	108	131	3642
23	23	60	2020-12-23	130	101	3000
24	24	45	2020-12-24	105	132	2460
25	25	60	2020-12-25	102	126	3345
27	27	60	2020-12-27	92	118	2410
28	28	60	2020-12-28	103	132	0
29	29	60	2020-12-29	100	132	2800
30	30	60	2020-12-30	102	129	3803
31	31	60	2020-12-31	92	115	2430

Dataframe final após remover registros com valores nulos:

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	2020-12-01	110	130	4091
1	1	60	2020-12-02	117	145	4790
2	2	60	2020-12-03	103	135	3400
3	3	45	2020-12-04	109	175	2824
4	4	45	2020-12-05	117	148	4060
5	5	60	2020-12-06	102	127	3000
6	6	60	2020-12-07	110	136	3740
7	7	450	2020-12-08	104	134	2533
8	8	30	2020-12-09	109	133	1951
9	9	60	2020-12-10	98	124	2690
10	10	60	2020-12-11	103	147	3293
11	11	60	2020-12-12	100	120	2507
12	12	60	2020-12-12	100	120	2507
13	13	60	2020-12-13	106	128	3453
14	14	60	2020-12-14	104	132	3793
15	15	60	2020-12-15	98	123	2750
16	16	60	2020-12-16	98	120	2152
17	17	60	2020-12-17	100	120	3000
18	18	45	2020-12-18	90	112	0
19	19	60	2020-12-19	103	123	3230
20	20	45	2020-12-20	97	125	2430 2
21	1	60	2020-12-21	108	131	3642
23	23	60	2020-12-23	130	101	3000
24	24	45	2020-12-24	105	132	2460
25	25	60	2020-12-25	102	126	3345
27	27	60	2020-12-27	92	118	2410
28	28	60	2020-12-28	103	132	0
29	29	60	2020-12-29	100	132	2800
30	30	60	2020-12-30	102	129	3803
31	31	60	2020-12-31	92	115	2430