 <b>Estácio</b>	<p style="text-align: right;"><i>Universidade Estácio Campus Polo Casa Amarela Curso de Desenvolvimento Full Stack</i></p>
<i>Disciplina:</i>	DGT2823 Tecnologias para desenv. de soluções de big data
<i>Nome:</i>	João Gilberto dos Santos
<i>Turma:</i>	2022.4

### ***Sumario:***

#### DGT2823 - Tecnologias para Desenvolvimento de Soluções de Big Data

1.	INTRODUÇÃO .....	2
2.	OBJETIVOS .....	2
3.	METODOLOGIA .....	3
3.1	Ambiente de Desenvolvimento .....	3
3.2	Dataset Utilizado .....	3
3.3	Problemas Identificados no Dataset.....	3
4.	MICROATIVIDADES .....	4
4.1	Microatividade 1: Leitura de CSV.....	4
4.2	Microatividade 2: Subconjunto de Dados .....	5
4.3	Microatividade 3: Configuração de Visualização .....	6
4.4	Microatividade 4: Descrever como exibir as primeiras e últimas 'N'.....	7
4.5	Microatividade 5: Informações Gerais .....	8
	Trabalho Prático Final - DGT2823 .....	9
	Passo 1: Preparação do Dataset.....	9
	Passos 2-4: Leitura do CSV .....	10
	Passo 5: Verificação dos Dados .....	10
	Passo 6: Cópia do Dataset .....	11
	Passo 7: Tratamento de Valores Nulos em 'Calories'.....	11
	Passo 8: Tratamento Inicial da Coluna 'Date' .....	11
	Passo 9: Correção do Primeiro Erro .....	12
	Passo 10: Correção de Formato Inconsistente .....	12
	Passo 11: Conversão Final .....	13
	Passo 12: Remoção de Registros Nulos .....	13
	Passo 13: Verificação Final .....	14
	Passo 14 (Extra): Criar novo arquivo .csv com dados já tratados.....	15

## **1 . Introdução**

Este trabalho prático tem como finalidade demonstrar o conhecimento adquirido na disciplina DGT2823 Tecnologias para Desenvolvimento de Soluções de Big Data, com foco específico na manipulação e limpeza de dados utilizando a biblioteca Pandas da linguagem Python.

O trabalho foi desenvolvido no ambiente VScode, permitindo a execução de código Python de forma interativa e colaborativa.

O dataset utilizado contém informações sobre exercícios físicos, incluindo duração, data, pulso, pulso máximo e calorias queimadas. Este conjunto de dados apresenta propositalmente inconsistências e valores nulos, representando um cenário realista de dados que necessitam de tratamento antes de serem utilizados em análises.

## **2. Objetivo da Prática:**

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

### 3. Metodologia

#### 3.1 Ambiente de Desenvolvimento

- **Plataforma:** VScode
- **Linguagem:** Python 3.x
- **Biblioteca Principal:** Pandas
- **Formato de Dados:** CSV (Comma-Separated Values)

#### 3.2 Dataset Utilizado

O dataset contém 32 registros iniciais com as seguintes colunas:

- **ID:** Identificador único do exercício
- **Duration:** Duração do exercício em minutos
- **Date:** Data do exercício
- **Pulse:** Frequência cardíaca durante o exercício
- **Maxpulse:** Frequência cardíaca máxima atingida
- **Calories:** Calorias queimadas

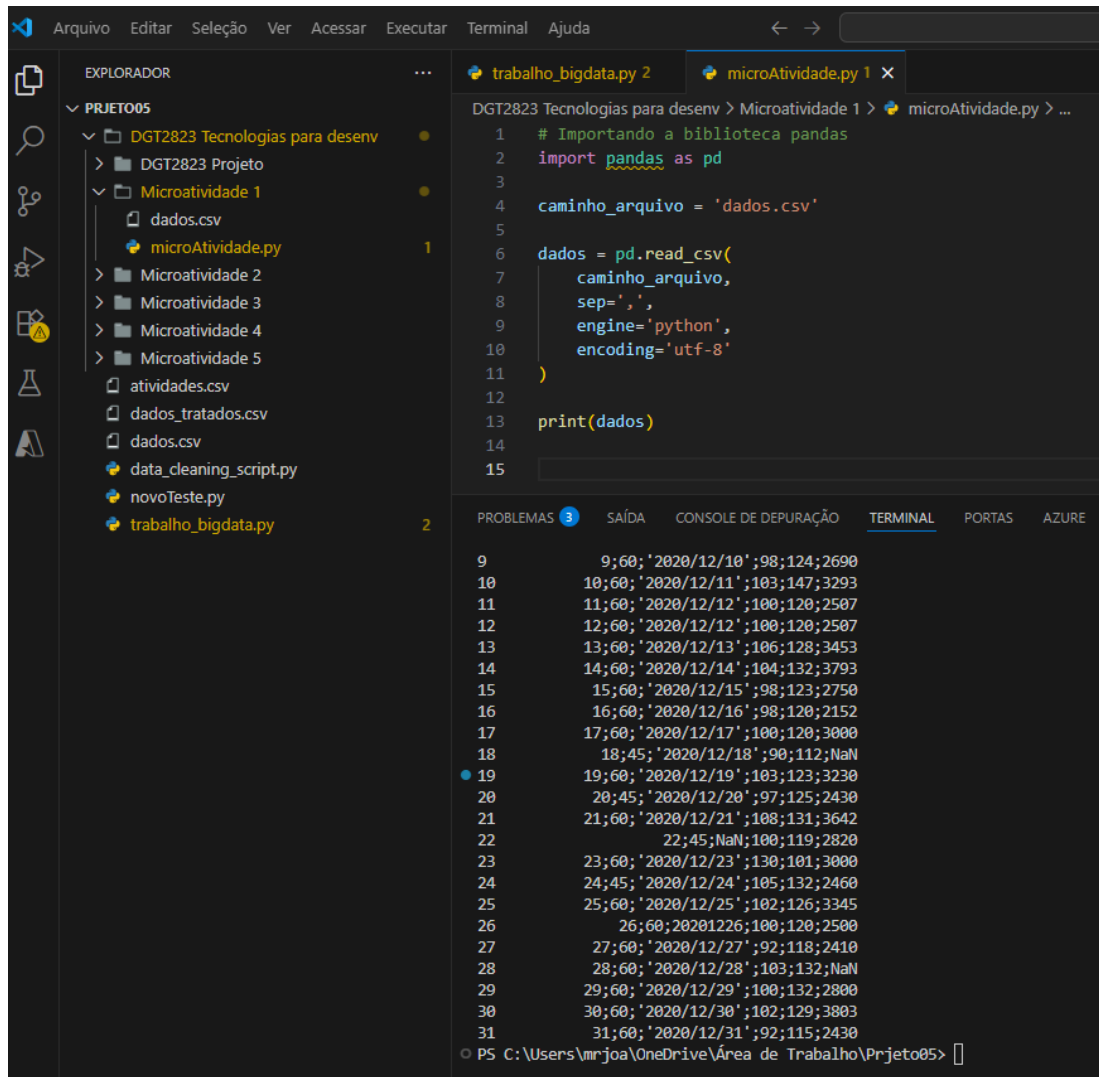
#### 3.3 Problemas identificados no Dataset

- Valores nulos (NaN) nas colunas Calories e Date
- Formato inconsistente da data (linha 26: "20201226")
- Necessidade de conversão de tipos de dados

Link GitHub <https://github.com/J-Gilberto/Prjeto05.git>

## 4. Micro Atividades

### Micro Atividade 4.1



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure for 'PRJETO05' with folders 'DGT2823 Tecnologias para desenv', 'Microatividade 1' through '5', and files 'atividades.csv', 'dados\_tratados.csv', 'dados.csv', 'data\_cleaning\_script.py', 'novoTeste.py', and 'trabalho\_bigdata.py'. The main editor shows a file named 'microAtividade.py' with the following Python code:

```
1 # Importando a biblioteca pandas
2 import pandas as pd
3
4 caminho_arquivo = 'dados.csv'
5
6 dados = pd.read_csv(
7     caminho_arquivo,
8     sep=',',
9     engine='python',
10    encoding='utf-8'
11 )
12
13 print(dados)
14
15
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
9;60;'2020/12/10';98;124;2690
10;60;'2020/12/11';103;147;3293
11;60;'2020/12/12';100;120;2507
12;60;'2020/12/12';100;120;2507
13;60;'2020/12/13';106;128;3453
14;60;'2020/12/14';104;132;3793
15;60;'2020/12/15';98;123;2750
16;60;'2020/12/16';98;120;2152
17;60;'2020/12/17';100;120;3000
18;45;'2020/12/18';90;112;NaN
19;60;'2020/12/19';103;123;3230
20;45;'2020/12/20';97;125;2430
21;60;'2020/12/21';108;131;3642
22;45;NaN;100;119;2820
23;60;'2020/12/23';130;101;3000
24;45;'2020/12/24';105;132;2460
25;60;'2020/12/25';102;126;3345
26;60;20201226;100;120;2500
27;60;'2020/12/27';92;118;2410
28;60;'2020/12/28';103;132;NaN
29;60;'2020/12/29';100;132;2800
30;60;'2020/12/30';102;129;3803
31;60;'2020/12/31';92;115;2430
```

The terminal prompt is 'PS C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05>'.

Nesta microatividade, tive a oportunidade de aplicar conceitos fundamentais do ecossistema Python, voltados para a análise de dados. O foco principal foi:

- Configurar o ambiente de desenvolvimento, seja local (como VSCode ou Jupyter) ou remoto (como Google Colab);
- Importar a biblioteca pandas, essencial para manipulação de dados em tabelas;
- Realizar a leitura de um arquivo CSV, utilizando parâmetros como separador (sep), motor de leitura (engine='python') e codificação (encoding);
- Armazenar o conteúdo lido em uma variável, garantindo acesso fácil e organizado aos dados;
- Exibir os dados na tela, como forma de verificar o sucesso da operação e validar visualmente o conteúdo importado.

## Micro Atividade 4.2

The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project structure with folders for 'DGT2823 Tecnologias para desenv', 'Microatividade 1', 'Microatividade 2', 'Microatividade 3', 'Microatividade 4', and 'Microatividade 5'. The code editor displays a Python script named 'microAtividade2.py' with the following code:

```
1 # Adicionando as 3 tabelas de dados
2 import pandas as pd
3
4 caminho_arquivo = 'dados.csv'
5
6 dados = pd.read_csv(
7     caminho_arquivo, sep=';',
8     engine='python',
9     encoding='utf-8'
10 )
11
12
```

The terminal output shows the columns available and a subset of data:

Colunas disponíveis:  
Index(['ID', 'Duration', 'Date', 'Pulse', 'Maxpulse', 'Calories'], dtype=object)

Subconjunto de dados:

	ID	Duration	Calories
0	0	60	4091.0
1	1	60	4790.0
2	2	60	3400.0
3	3	45	2824.0
4	4	45	4060.0
5	5	60	3000.0
6	6	60	3740.0
7	7	450	2533.0
8	8	30	1951.0
9	9	60	2690.0
10	10	60	3293.0
11	11	60	2507.0
12	12	60	2507.0
13	13	60	3453.0
14	14	60	3793.0
15	15	60	2750.0
16	16	60	2152.0
17	17	60	3000.0
18	18	45	NaN
19	19	60	3230.0
20	20	45	2430.0
21	21	60	3642.0
22	22	45	2820.0
23	23	60	3000.0
24	24	45	2460.0
25	25	60	3345.0
26	26	60	2500.0
27	27	60	2410.0
28	28	60	NaN
29	29	60	2800.0
30	30	60	3803.0
31	31	60	2430.0

The terminal prompt is PS C:\Users\mirjoa\OneDrive\Área de Trabalho\Prjeto05>.

Carregando os dados.csv, filtrei informações específicas com condições lógicas usando `loc[ ]`, apliquei operadores como `&` e `|` e exibi só os dados que atendiam aos critérios definidos.

## Micro Atividade 4.3

The screenshot shows the Visual Studio Code interface with a project named 'PRJETO05'. The file explorer on the left shows a directory structure with files like 'dados.csv', 'microAtividade.py', 'microAtividade2.py', 'microAtividade3.py', 'dados\_cleaning\_script.py', and 'novoTeste.py'. The main editor displays the code for 'microAtividade3.py', which imports pandas, sets display options, reads a CSV file, and prints the data as a string.

```
1 # Importar biblioteca pandas
2 import pandas as pd
3
4 pd.options.display.max_rows = 9999
5
6 dados = pd.read_csv(
7     'dados.csv',
8     sep=';',
9     engine='python',
10    encoding='utf-8')
11
12 print(dados.to_string())
```

The terminal at the bottom shows the command prompt output, displaying a table of data with columns: ID, Duration, Date, Pulse, Maxpulse, and Calories. The data is printed as a string, showing 31 rows of data.

ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	'2020/12/01'	110	130	4091.0
1	1	'2020/12/02'	117	145	4790.0
2	2	'2020/12/03'	103	135	3400.0
3	3	'2020/12/04'	109	175	2824.0
4	4	'2020/12/05'	117	148	4060.0
5	5	'2020/12/06'	102	127	3000.0
6	6	'2020/12/07'	110	136	3740.0
7	7	'2020/12/08'	104	134	2533.0
8	8	'2020/12/09'	109	133	1951.0
9	9	'2020/12/10'	98	124	2690.0
10	10	'2020/12/11'	103	147	3293.0
11	11	'2020/12/12'	100	120	2507.0
12	12	'2020/12/12'	100	120	2507.0
13	13	'2020/12/13'	106	128	3453.0
14	14	'2020/12/14'	104	132	3793.0
15	15	'2020/12/15'	98	123	2750.0
16	16	'2020/12/16'	98	120	2152.0
17	17	'2020/12/17'	100	120	3000.0
18	18	'2020/12/18'	90	112	NaN
19	19	'2020/12/19'	103	123	3230.0
20	20	'2020/12/20'	97	125	2430.0
21	21	'2020/12/21'	108	131	3642.0
22	22	NaN	100	119	2820.0
23	23	'2020/12/23'	130	101	3000.0
24	24	'2020/12/24'	105	132	2460.0
25	25	'2020/12/25'	102	126	3345.0
26	26	'2020/12/26'	100	120	2500.0
27	27	'2020/12/27'	92	118	2410.0
28	28	'2020/12/28'	103	132	NaN
29	29	'2020/12/29'	100	132	2800.0
30	30	'2020/12/30'	102	129	3803.0
31	31	'2020/12/31'	92	115	2430.0

Após abri o script, importei o pandas, configurei o max rows pra 9999, carreguei o dados.csv, corriji um erro de caminho e exibi o dataframe completo com to string(). Tudo funcio nando certinho!

## Micro Atividade 4.4

The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'PRJETO05' with a folder 'DGT2823 Tecnologias para desenv' containing subfolders 'Microatividade 1' through 'Microatividade 5'. The file 'microAtividade4.py' is selected in 'Microatividade 4'. The code editor shows the following Python code:

```
1 # Importa a biblioteca pandas
2 import pandas as pd
3
4 dados = pd.read_csv(
5     'dados.csv',
6     sep=';',
7     engine='python',
8     encoding='utf-8'
9 )
10
11 print("Primeiras 10 linhas:")
12 print(dados.head(10))
13
14 print("\nÚltimas 10 linhas:")
15 print(dados.tail(10))
```

The terminal shows the command prompt output:

```
PS C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05> c:: cd 'C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05' & python -u -x py-2025.10.0-win32-x64\bundled\libs\debugpy\launcher '58347' '--' 'C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05\microAtividade4.py'
Primeiras 10 linhas:
```

ID	Duration	Date	Pulse	Maxpulse	Calories	
0	0	60	'2020/12/01'	110	130	4091.0
1	1	60	'2020/12/02'	117	145	4790.0
2	2	60	'2020/12/03'	103	135	3400.0
3	3	45	'2020/12/04'	109	175	2824.0
4	4	45	'2020/12/05'	117	148	4060.0
5	5	60	'2020/12/06'	102	127	3000.0
6	6	60	'2020/12/07'	110	136	3740.0
7	7	450	'2020/12/08'	104	134	2533.0
8	8	30	'2020/12/09'	109	133	1951.0
9	9	60	'2020/12/10'	98	124	2690.0

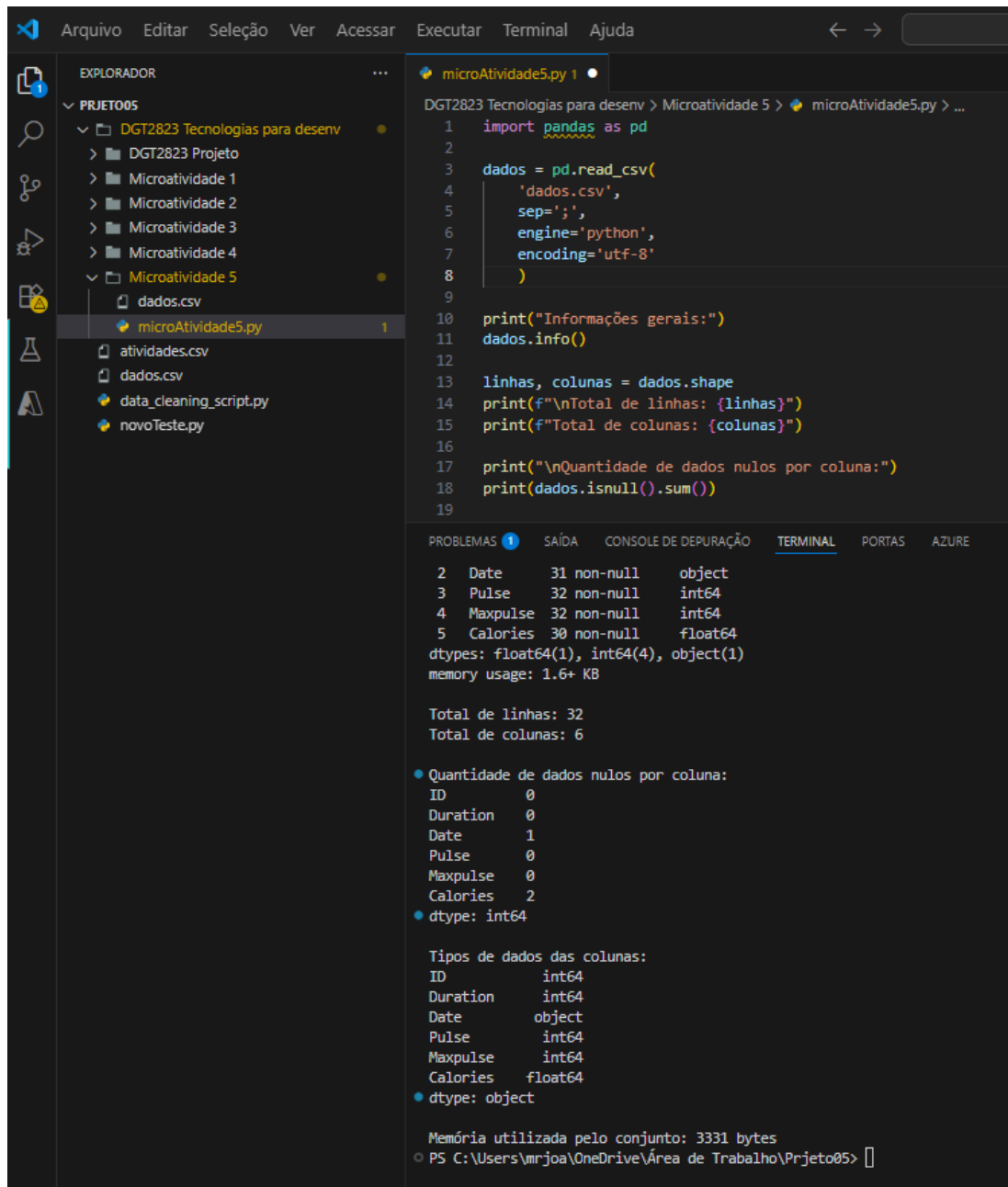
Últimas 10 linhas:

ID	Duration	Date	Pulse	Maxpulse	Calories	
22	22	45	NaN	100	119	2820.0
23	23	60	'2020/12/23'	130	101	3000.0
24	24	45	'2020/12/24'	105	132	2460.0
25	25	60	'2020/12/25'	102	126	3345.0
26	26	60	20201226	100	120	2500.0
27	27	60	'2020/12/27'	92	118	2410.0
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800.0
30	30	60	'2020/12/30'	102	129	3803.0
31	31	60	'2020/12/31'	92	115	2430.0

PS C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05> [ ]

Mais uma vez, carreguei os dados.csv, usei head(10) e tail(10) para mostrar as 10 primeiras e 10 últimas linhas do dataframe, explorando os métodos básicos de visualização de dados com pandas.

## Micro Atividade 4.5



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'PRJETO05' with a folder 'DGT2823 Tecnologias para desenv' containing subfolders 'Microatividade 1' through 'Microatividade 5'. The file 'microAtividade5.py' is selected. The code editor shows the following Python code:

```
1 import pandas as pd
2
3 dados = pd.read_csv(
4     'dados.csv',
5     sep=';',
6     engine='python',
7     encoding='utf-8'
8 )
9
10 print("Informações gerais:")
11 dados.info()
12
13 linhas, colunas = dados.shape
14 print(f"\nTotal de linhas: {linhas}")
15 print(f"Total de colunas: {colunas}")
16
17 print("\nQuantidade de dados nulos por coluna:")
18 print(dados.isnull().sum())
19
```

The terminal output shows the following information:

```
2 Date      31 non-null    object
3 Pulse     32 non-null    int64
4 Maxpulse  32 non-null    int64
5 Calories  30 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 1.6+ KB

Total de linhas: 32
Total de colunas: 6

• Quantidade de dados nulos por coluna:
ID          0
Duration    0
Date        1
Pulse       0
Maxpulse    0
Calories    2
• dtype: int64

Tipos de dados das colunas:
ID          int64
Duration    int64
Date        object
Pulse       int64
Maxpulse    int64
Calories    float64
• dtype: object

Memória utilizada pelo conjunto: 3331 bytes
PS C:\Users\mrj\OneDrive\Trabalho\Prjeto05>
```

Após carregar os dados.csv, usei `.info()`, `.shape`, `.isnull().sum()`, `.dtypes` e `.memory_usage()` para inspecionar colunas, tipos, dados nulos, dimensões e memória ocupada pelo dataset.



## Trabalho Prático Final - DGT2823

### Passo 1: Preparação do Dataset

```
Arquivo  Editar  Seleção  ...  ←  →  🔍 Projeto05

trabalho_bigdata.py ●
DGT2823 Tecnologias para desenv > DGT2823 Projeto > trabalho_bigdata.py > ...

1  # tratamento_dados.py
2  # Autor: João Gilberto e com ajuda da ferramenta Git Copilot
3  # Objetivo: Limpeza e preparação de dados para análise usando Pandas
4
5  import pandas as pd
6  import numpy as np
7
8  # PASSO 1: Leitura do CSV
9  sample_data = """ID;Duration;Date;Pulse;Maxpulse;Calories
10 0;60;'2020/12/01';110;130;4091
11 1;60;'2020/12/02';117;145;4790
12 2;60;'2020/12/03';103;135;3400
13 3;45;'2020/12/04';109;175;2824
14 4;45;'2020/12/05';117;148;4060
15 5;60;'2020/12/06';102;127;3000
16 6;60;'2020/12/07';110;136;3740
17 7;450;'2020/12/08';104;134;2533
18 8;30;'2020/12/09';109;133;1951
19 9;60;'2020/12/10';98;124;2690
20 10;60;'2020/12/11';103;147;3293
21 11;60;'2020/12/12';100;120;2507
22 12;60;'2020/12/12';100;120;2507
23 13;60;'2020/12/13';106;128;3453
24 14;60;'2020/12/14';104;132;3793
25 15;60;'2020/12/15';98;123;2750
26 16;60;'2020/12/16';98;120;2152
27 17;60;'2020/12/17';100;120;3000
28 18;45;'2020/12/18';90;112;NaN
29 19;60;'2020/12/19';103;123;3230
30 20;45;'2020/12/20';97;125;2430
31 21;60;'2020/12/21';108;131;3642
32 22;45;NaN;100;119;2820
33 23;60;'2020/12/23';130;101;3000
34 24;45;'2020/12/24';105;132;2460
35 25;60;'2020/12/25';102;126;3345
36 26;60;20201226;100;120;2500
37 27;60;'2020/12/27';92;118;2410
38 28;60;'2020/12/28';103;132;NaN
39 29;60;'2020/12/29';100;132;2800
40 30;60;'2020/12/30';102;129;3803
41 31;60;'2020/12/31';92;115;2430"""
42
43 # Gravar CSV na pasta local
44 with open('dados.csv', 'w', encoding='utf-8') as f:
45     f.write(sample_data)
```

### Passos 2-3: Leitura do CSV =====

```
df_original = pd.read_csv('dados.csv', sep=';', engine='python', encoding='utf-8')

# PASSO 2: Novo arquivo/script (já criado)
print("\n--- Passo 2: Novo script criado ---")
# 2 Informações gerais do dataset
print("\n Informações gerais do DataFrame original:")
print(df_original.info())
print("\n Primeiras 5 linhas:")
print(df_original.head())
print("\n Últimas 5 linhas:")
print(df_original.tail())

# PASSO 3: Leitura dos dados
print("\n--- PASSO 3: Leitura dos dados ---")
# 3 Cópia dos dados originais
df_tratado = df_original.copy()
```

### Resultados:

```
Primeiras 5 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
0   0         60 '2020/12/01'   110      130    4091.0
1   1         60 '2020/12/02'   117      145    4790.0
2   2         60 '2020/12/03'   103      135    3400.0
3   3         45 '2020/12/04'   109      175    2824.0
4   4         45 '2020/12/05'   117      148    4060.0

Últimas 5 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
27  27         60 '2020/12/27'    92      118    2410.0
28  28         60 '2020/12/28'   103      132      NaN
29  29         60 '2020/12/29'   100      132    2800.0
30  30         60 '2020/12/30'   102      129    3803.0
31  31         60 '2020/12/31'    92      115    2430.0
```

### Passos 4 Verificação da coluna Calories =====

```
67 # PASSO 4: Verificação da coluna 'Calories'
68 print("\n--- PASSO 4: Tratamento da coluna 'Calories' ---")
69 print("\nColunas disponíveis:\n", list(df_tratado.columns))
70 # 4 Substituir nulos da coluna 'Calories' por 0
71 df_tratado['Calories'].fillna(0, inplace=True)
72
```

### Passo 5 Verificação da coluna Date =====

```
73 # PASSO 5: Verificação da coluna 'Date'
74 print("\n--- PASSO 5: Tratamento da coluna 'Date' ---")
75 # 5 Substituir nulos da coluna 'Date' por '1900/01/01'
76 df_tratado['Date'].fillna('1900/01/01', inplace=True)
77
```

#### Passo 6 Tentativa de conversão da coluna 'Date' para datetime =====

```
78 #PASSO 6: Tentativa de conversão da coluna 'Date' para datetime
79 print("\n--- PASSO 6: Tentativa de conversão da coluna 'Date'---")
80 # 6 Substituir '1900/01/01' por NaN
81 df_tratado['Date'].replace('1900/01/01', np.nan, inplace=True)
82
```

#### Passo 7 Conversão da coluna 'Date' para datetime =====

```
83 #PASSO 7: Conversão da coluna 'Date' para datetime
84 print("\n--- PASSO 7: Conversão da coluna 'Date' para datetime ---")
85 # 7 Corrigir valor '20201226' para formato datetime
86 df_tratado['Date'] = df_tratado['Date'].replace('20201226', '2020-12-26')
87
```

#### Passo 8 Conversão final da coluna 'Date' =====

```
88 #PASSO 8: Conversão final da coluna 'Date'
89 print("\n--- PASSO 8: Conversão final da coluna 'Date' ---")
90 # 8 Detectar formatos automaticamente e converter para datetime
91 df_tratado['Date'] = pd.to_datetime(df_tratado['Date'], errors='coerce', infer_datetime_format=True)
92
```

## Passo 9 – 10 Conversão final da coluna ‘Date’ =====

```
93 # PASSO 9: Verificação final das datas tratadas
94 print("\n--- PASSO 9: Verificação final das datas tratadas ---")
95 # 9 Verificar resultados após conversão
96 print("\n Datas tratadas:")
97 print(df_tratado['Date'])
98
99 # PASSO 10: Exibir DataFrame tratado
100 print("\n DataFrame tratado:")
101 # Confirmar conversão de todas as datas (nova tentativa já feita acima)
102 # (Etapa já concluída com o uso de infer_datetime_format)
103
```

## Resultados

```
Datas tratadas:
0    2020-12-01
1    2020-12-02
2    2020-12-03
3    2020-12-04
4    2020-12-05
5    2020-12-06
6    2020-12-07
7    2020-12-08
8    2020-12-09
9    2020-12-10
10   2020-12-11
11   2020-12-12
12   2020-12-12
13   2020-12-13
14   2020-12-14
15   2020-12-15
16   2020-12-16
17   2020-12-17
18   2020-12-18
19   2020-12-19
20   2020-12-20
21   2020-12-21
22         NaT
23   2020-12-23
24   2020-12-24
25   2020-12-25
26         NaT
27   2020-12-27
28   2020-12-28
29   2020-12-29
30   2020-12-30
31   2020-12-31
Name: Date, dtype: datetime64[ns]
```

### Passo 11 Conversão final da coluna 'Date' =====

```
104 #PASSO 11: Exibir DataFrame completo tratado antes da remoção de nulos
105 print("\n DataFrame tratado antes da limpeza final:")
106 # 11 Exibir DataFrame completo tratado antes da remoção de nulos
107 print(df_tratado)
108
```

### Resultados:

```
DataFrame tratado antes da limpeza final:
   ID  Duration    Date  Pulse  Maxpulse  Calories
0   0         60 2020-12-01   110      130    4091.0
1   1         60 2020-12-02   117      145    4790.0
2   2         60 2020-12-03   103      135    3400.0
3   3         45 2020-12-04   109      175    2824.0
4   4         45 2020-12-05   117      148    4060.0
5   5         60 2020-12-06   102      127    3000.0
6   6         60 2020-12-07   110      136    3740.0
7   7        450 2020-12-08   104      134    2533.0
8   8         30 2020-12-09   109      133    1951.0
9   9         60 2020-12-10    98      124    2690.0
10 10         60 2020-12-11   103      147    3293.0
11 11         60 2020-12-12   100      120    2507.0
12 12         60 2020-12-12   100      120    2507.0
13 13         60 2020-12-13   106      128    3453.0
14 14         60 2020-12-14   104      132    3793.0
15 15         60 2020-12-15    98      123    2750.0
16 16         60 2020-12-16    98      120    2152.0
17 17         60 2020-12-17   100      120    3000.0
18 18         45 2020-12-18    90      112     0.0
19 19         60 2020-12-19   103      123    3230.0
20 20         45 2020-12-20    97      125    2430.0
21 21         60 2020-12-21   108      131    3642.0
22 22         45      NaT   100      119    2820.0
23 23         60 2020-12-23   130      101    3000.0
24 24         45 2020-12-24   105      132    2460.0
25 25         60 2020-12-25   102      126    3345.0
26 26         60      NaT   100      120    2500.0
27 27         60 2020-12-27    92      118    2410.0
28 28         60 2020-12-28   103      132     0.0
29 29         60 2020-12-29   100      132    2800.0
30 30         60 2020-12-30   102      129    3803.0
31 31         60 2020-12-31    92      115    2430.0

--- PASSO 12: Removendo registros com valores nulos ---
```

### Passo 12 Remover registros com valores nulos =====

```
109 # PASSO 12: Remover registros com valores nulos
110 print("\n--- PASSO 12: Removendo registros com valores nulos ---")
111 # 12 Remover registros com valores nulos (como datas inválidas)
112 df_final = df_tratado.dropna()
113
```

### Passo 13 exibir DataFrame final tratado =====

```
114 #PASSO 13: Exibir DataFrame final tratado
115 print("\n--- PASSO 13: DataFrame final tratado ---")
116 # 13 Exibir e salvar DataFrame final tratado
117 print("\n DataFrame final após remoção de nulos:")
118 print(df_final)
119
```

### Resultados:

```
--- PASSO 13: DataFrame final tratado ---

DataFrame final após remoção de nulos:
   ID  Duration      Date  Pulse  Maxpulse  Calories
0   0         60 2020-12-01   110       130    4091.0
1   1         60 2020-12-02   117       145    4790.0
2   2         60 2020-12-03   103       135    3400.0
3   3         45 2020-12-04   109       175    2824.0
4   4         45 2020-12-05   117       148    4060.0
5   5         60 2020-12-06   102       127    3000.0
6   6         60 2020-12-07   110       136    3740.0
7   7        450 2020-12-08   104       134    2533.0
8   8         30 2020-12-09   109       133    1951.0
9   9         60 2020-12-10    98       124    2690.0
10  10         60 2020-12-11   103       147    3293.0
11  11         60 2020-12-12   100       120    2507.0
12  12         60 2020-12-12   100       120    2507.0
13  13         60 2020-12-13   106       128    3453.0
14  14         60 2020-12-14   104       132    3793.0
15  15         60 2020-12-15    98       123    2750.0
16  16         60 2020-12-16    98       120    2152.0
17  17         60 2020-12-17   100       120    3000.0
18  18         45 2020-12-18    90       112         0.0
19  19         60 2020-12-19   103       123    3230.0
20  20         45 2020-12-20    97       125    2430.0
21  21         60 2020-12-21   108       131    3642.0
23  23         60 2020-12-23   130       101    3000.0
24  24         45 2020-12-24   105       132    2460.0
25  25         60 2020-12-25   102       126    3345.0
27  27         60 2020-12-27    92       118    2410.0
28  28         60 2020-12-28   103       132         0.0
29  29         60 2020-12-29   100       132    2800.0
30  30         60 2020-12-30   102       129    3803.0
31  31         60 2020-12-31    92       115    2430.0
```

#### Passo 14 - Extra Exibir DataFrame final tratado e gerar novo csv =====

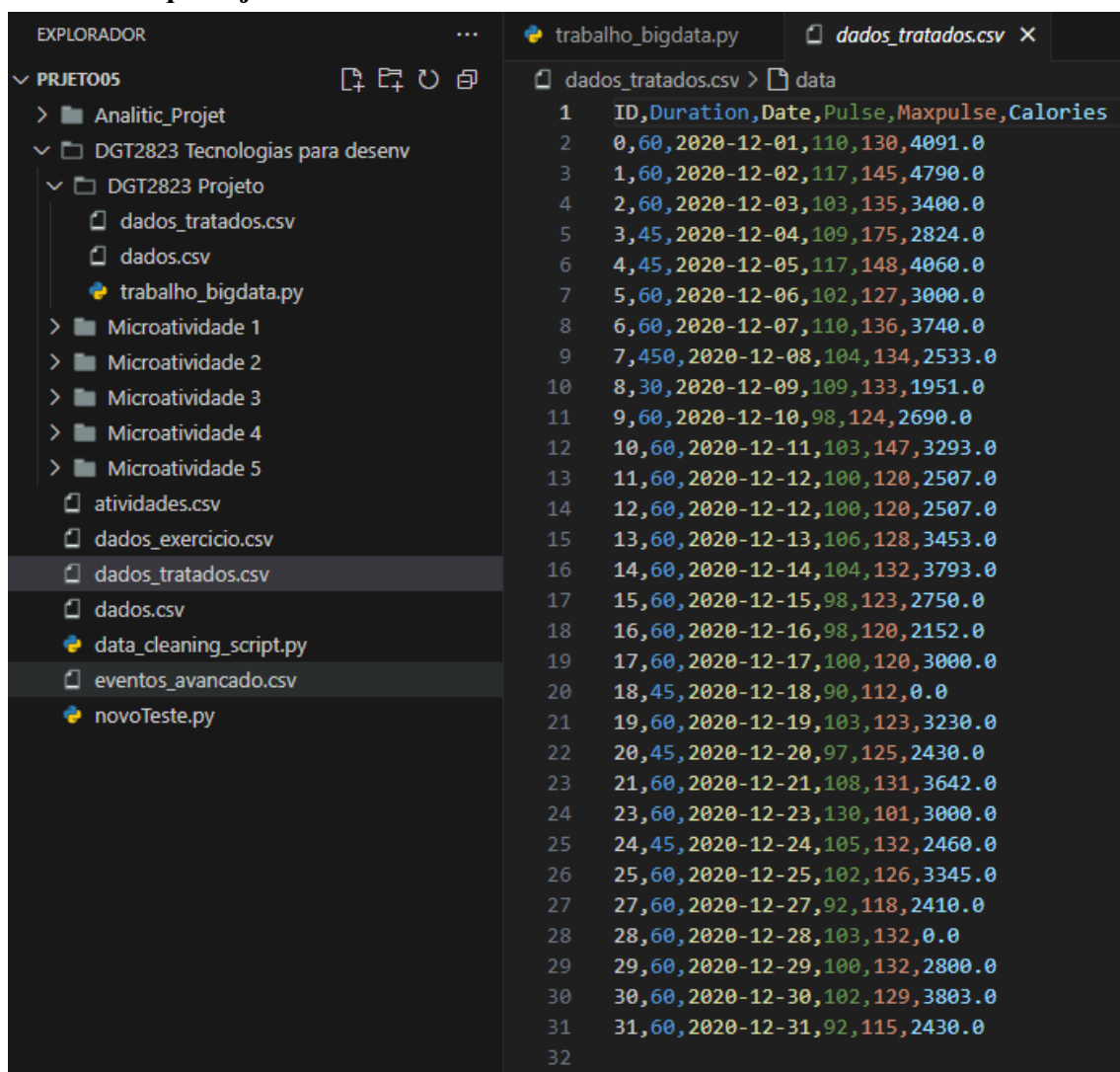
```
120 #PASSO 14: Exportar DataFrame tratado para CSV
121 print("\n--- PASSO 14: Exportando DataFrame tratado para CSV ---")
122 # (Extra) Exportar para novo arquivo CSV
123 df_final.to_csv('dados_tratados.csv', index=False)
124 print("\n Arquivo 'dados_tratados.csv' gerado com sucesso!")
```

#### Resultados:

```
--- PASSO 14: Exportando DataFrame tratado para CSV ---

Arquivo 'dados_tratados.csv' gerado com sucesso!
PS C:\Users\mrjoa\OneDrive\Área de Trabalho\Prjeto05>
```

#### Print do arquivo já tratado.



	ID	Duration	Date	Pulse	Maxpulse	Calories
1	0	60	2020-12-01	110	130	4091.0
2	1	60	2020-12-02	117	145	4790.0
3	2	60	2020-12-03	103	135	3400.0
4	3	45	2020-12-04	109	175	2824.0
5	4	45	2020-12-05	117	148	4060.0
6	5	60	2020-12-06	102	127	3000.0
7	6	60	2020-12-07	110	136	3740.0
8	7	450	2020-12-08	104	134	2533.0
9	8	30	2020-12-09	109	133	1951.0
10	9	60	2020-12-10	98	124	2690.0
11	10	60	2020-12-11	103	147	3293.0
12	11	60	2020-12-12	100	120	2507.0
13	12	60	2020-12-12	100	120	2507.0
14	13	60	2020-12-13	106	128	3453.0
15	14	60	2020-12-14	104	132	3793.0
16	15	60	2020-12-15	98	123	2750.0
17	16	60	2020-12-16	98	120	2152.0
18	17	60	2020-12-17	100	120	3000.0
19	18	45	2020-12-18	90	112	0.0
20	19	60	2020-12-19	103	123	3230.0
21	20	45	2020-12-20	97	125	2430.0
22	21	60	2020-12-21	108	131	3642.0
23	23	60	2020-12-23	130	101	3000.0
24	24	45	2020-12-24	105	132	2460.0
25	25	60	2020-12-25	102	126	3345.0
26	27	60	2020-12-27	92	118	2410.0
27	28	60	2020-12-28	103	132	0.0
28	29	60	2020-12-29	100	132	2800.0
29	30	60	2020-12-30	102	129	3803.0
30	31	60	2020-12-31	92	115	2430.0
31						
32						