Adicionando os imports

In [1]:
```
import pandas as pd
import numpy as np
from tabulate import tabulate
```

Carregar dados

In [2]:
```
df_original = pd.read_csv('dados.csv', sep=';', engine='python')
```

Visualizar informações iniciais

In [3]:
```
print("Informações gerais:")
print(df_original.info())
```

```
Informações gerais:
<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
None
```

In [4]:
```
print("\nPrimeiras 5 linhas:")
df_original.head()
```

```
Primeiras 5 linhas:
```

Out[4]:

	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

In [5]:
```
print("\nÚltimas 5 linhas:")
df_original.tail()
```

```
Últimas 5 linhas:
```

Out[5]:

	ID	Duration	Date	Pulse	Maxpulse	Calories
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

Limpeza da coluna 'Calories'

In [6]:
```
df = df_original.copy()

df['Calories'] = df['Calories'].replace(r'^\s*$', np.nan, regex=True)
df['Calories'] = pd.to_numeric(df['Calories'], errors='coerce')
df['Calories'] = df['Calories'].fillna(0)

df
```

Out[6]:

	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	0.0
21	1	60	'2020/12/21'	108	131	3642.0
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	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

Limpeza da coluna 'Date'

In [7]:
```
df['Date'] = df['Date'].replace(r'^\s*$', np.nan, regex=True)
df['Date'] = df['Date'].str.replace('"', '', regex=False)
df['Date'] = df['Date'].str.replace("'", '', regex=False)
```

```
df['Date'] = df['Date'].replace('20201226', '2020/12/26')
df['Date'] = df['Date'].fillna('1900/01/01')

df
```

Out[7]:

	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	0.0
21	1	60	2020/12/21	108	131	3642.0
22	22	45	1900/01/01	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	2020/12/26	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

Conversão da coluna Date

In [9]:
```
df['Date'] = pd.to_datetime(df['Date'], format='%Y/%m/%d', errors='coerce')
df['Date'] = df['Date'].replace(pd.Timestamp('1900-01-01'), pd.NaT)

df
```

Out[9]:

	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	0.0
21	1	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	2020-12-26	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

Remover valores nulos

In [10]:
```
df = df.dropna()
df
```

Out[10]:

	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	0.0
21	1	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	26	60	2020-12-26	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

Exibir tabela formatada com tabulate

In [11]:

```
from IPython.display import Markdown

# Formatar tabela com `tabulate` e exibir como Markdown para melhor visualização no notebook
formatted_table = tabulate(df, headers='keys', tablefmt='grid', showindex=False)
display(Markdown(f'```\n{formatted_table}\n```'))
```

ID	Duration	Date	Pulse	Maxpulse	Calories
0	60	2020-12-01 00:00:00	110	130	4091
1	60	2020-12-02 00:00:00	117	145	4790
2	60	2020-12-03 00:00:00	103	135	3400
3	45	2020-12-04 00:00:00	109	175	2824
4	45	2020-12-05 00:00:00	117	148	4060
5	60	2020-12-06 00:00:00	102	127	3000
6	60	2020-12-07 00:00:00	110	136	3740
7	450	2020-12-08 00:00:00	104	134	2533
8	30	2020-12-09 00:00:00	109	133	1951
9	60	2020-12-10 00:00:00	98	124	2690
10	60	2020-12-11 00:00:00	103	147	3293
11	60	2020-12-12 00:00:00	100	120	2507
12	60	2020-12-12 00:00:00	100	120	2507
13	60	2020-12-13 00:00:00	106	128	3453
14	60	2020-12-14 00:00:00	104	132	3793
15	60	2020-12-15 00:00:00	98	123	2750
16	60	2020-12-16 00:00:00	98	120	2152
17	60	2020-12-17 00:00:00	100	120	3000
18	45	2020-12-18 00:00:00	90	112	0
19	60	2020-12-19 00:00:00	103	123	3230
20	45	2020-12-20 00:00:00	97	125	0
1	60	2020-12-21 00:00:00	108	131	3642
23	60	2020-12-23 00:00:00	130	101	3000
24	45	2020-12-24 00:00:00	105	132	2460

```
+------+------------+---------------------+---------+------------+------------+
|   25 |         60 | 2020-12-25 00:00:00 |     102 |        126 |       3345 |
+------+------------+---------------------+---------+------------+------------+
|   26 |         60 | 2020-12-26 00:00:00 |     100 |        120 |       2500 |
+------+------------+---------------------+---------+------------+------------+
|   27 |         60 | 2020-12-27 00:00:00 |      92 |        118 |       2410 |
+------+------------+---------------------+---------+------------+------------+
|   28 |         60 | 2020-12-28 00:00:00 |     103 |        132 |          0 |
+------+------------+---------------------+---------+------------+------------+
|   29 |         60 | 2020-12-29 00:00:00 |     100 |        132 |       2800 |
+------+------------+---------------------+---------+------------+------------+
|   30 |         60 | 2020-12-30 00:00:00 |     102 |        129 |       3803 |
+------+------------+---------------------+---------+------------+------------+
|   31 |         60 | 2020-12-31 00:00:00 |      92 |        115 |       2430 |
+------+------------+---------------------+---------+------------+------------+
```

Informações finais do DataFrame

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 31 entries, 0 to 31
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype         
---  ------    --------------  -----         
 0   ID        31 non-null     int64         
 1   Duration  31 non-null     int64         
 2   Date      31 non-null     datetime64[ns]
 3   Pulse     31 non-null     int64         
 4   Maxpulse  31 non-null     int64         
 5   Calories  31 non-null     float64       
dtypes: datetime64[ns](1), float64(1), int64(4)
memory usage: 1.7 KB
```

In [ ]: