

Capturas de pantalla.

Estado de DataFrame antes:

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xed in position 62: invalid continuation byte
PS C:/Users/maria/OneDrive/Escritorio/Académico/Programación I/github/T_RestrepoPorras_Chau> & C:/Users/maria/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/maria/OneDrive/Escritorio/Académico/Programación I/github/T_RestrepoPorras_Chau/main.py"
ID;Edad;Sexo;Peso;Talla;PAS;PAD;Glucosa;Colesterol;Actividad física (min/sem);Fuma;Región;Fecha de tam
0 1;75;F;49.2;135.0;171;69;54.2;226.0;130;S;Orinoquia;2/10/2024
1 2;50;M;94.4;170.0;169;76;123.4;199.0;113;No;Pacífica;4/03/2025
2 3;68;F;102.4;157.0;121;81;171.5;204.9;237;No;Andina;25/03/2025
3 4;43;M;69.9;162.0;125;95;112.5;150.5;268;No;Caribe;30/09/2024
4 5;27;F;83.8;162.0;119;91;57.7;217;Si;Amazonia;9/10/2024
...
1015 469;26;M;52.9;170.0;116;96;119.0;225.7;43;No;A...
1016 357;32;F;58.1;159.0;178;95;243.3;168;No;Orino...
1017 878;31;M;64.4;164.0;176;93;148.7;270.6;51;Si;O...
1018 850;36;F;71.1;164.0;109;95;118.4;258.1;290;No;...
1019 651;69;F;51.6;168.0;160;92;89.0;192.8;22;No;An...
[1020 rows x 1 columns]
```

El primer problema fue leerlo, que se intentó solucionar poniendo `encoding='utf-8'` como sugería la misma consola, pero no funcionó, por lo que se le preguntó a la inteligencia artificial Aria, de Opera, qué otras opciones se podían poner cuando el utf-8 no funcionaba.

## 2. Prueba con otras codificaciones

- Si `utf-8` no funciona, intenta con otras codificaciones comunes, como:
  - `latin-1` (ISO-8859-1)
  - `utf-16`
  - `ascii`
- Ejemplo:

```
python

with open('archivo.txt', 'r', encoding='latin-1') as f:
    contenido = f.read()
```



## ¿Cómo funciona?

1. La columna `fecha_de_tamizaje` ya está convertida en formato de fecha gracias a `pd.to_datetime()`. Eso significa que cada fila tiene una fecha real, como `"14/05/2025"`, y no solo texto plano.
2. `.dt.month`
  - La parte `.dt` permite acceder a las propiedades de la fecha, como el año, el mes o el día.
  - Cuando usamos `.month`, estamos obteniendo solamente el número del mes de cada fecha.
  - Por ejemplo, si la fecha es `"14/05/2025"`, `df['fecha_de_tamizaje'].dt.month` devolverá `5`, porque mayo es el quinto mes del año.

## Ejemplo práctico:

Imagina que tienes una tabla con estas fechas:

fecha_de_tamizaje
10/01/2025
23/03/2025
14/05/2025

File Edit Selection ...

T\_RestrepoPorras\_Chau

main.py M x biometria\_pacientes.csv

main.py > ...

Open Changes

EXPLORER

T\_RESTREPOPORRAS\_CHA...

biometria\_pacientes.c...

main.py M

README.md

1 import pandas as pd

2 import matplotlib.pyplot as plt

3

4 #punto 1

5 df = pd.read\_csv("biometria\_pacientes.csv", encoding='latin1', sep=';')

6

7 df.columns = df.columns.str.strip().str.lower().str.replace(' ', '\_')

8

9 print("Primeras 10 filas:")

10 print(df.head(10))

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Python +

Python

hon312/python.exe "c:/Users/Usuario/Desktop/Ing. Biomédica/U 2025-1/Programación/T\_RestrepoPorras\_Chau/T\_RestrepoPorras\_Chau/main.py"

Primeras 10 filas:

	id	edad	sexo	...	fuma	región	fecha_de_tamizaje
0	1	75	F	...	Sí	Orinoquía	2/10/2024
1	2	50	M	...	No	Pacífica	4/03/2025
2	3	68	F	...	No	Andina	25/03/2025
3	4	43	M	...	No	Caribe	30/09/2024
4	5	27	F	...	Sí	Amazonía	9/10/2024
5	6	35	M	...	No	Amazonía	30/11/2024
6	7	60	M	...	No	Caribe	20/02/2025
7	8	34	F	...	No	Andina	11/03/2025
8	9	60	F	...	Sí	Amazonía	29/06/2024
9	10	47	M	...	SI	Orinoquía	29/04/2025

[10 rows x 13 columns]

PS C:\Users\Usuario\Desktop\Ing. Biomédica\U 2025-1\Programación\T\_RestrepoPorras\_Chau> T\_RestrepoPorras\_Chau\T\_RestrepoPorras\_Chau>

> TIMELINE

main\* 0L 11 0 0 0

Not Committed Yet

Ln 10, Col 19

Spaces: 4

UTF-8

CRLF

{ } Python

3.12.4

Go Live

```
T_RestrepoPorras_Chaux
main.py M x biometria_pacientes.csv
main.py > ...
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 #punto 1
5 df = pd.read_csv("biometria_pacientes.csv", encoding='latin1', sep=';')
6
7 df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
8
9 print("Primeras 10 filas:")
10 print(df.head(10))
11
12 #punto 2
13 duplicados = df.duplicated().sum()
14 print(f"Duplicados: {duplicados}")
15
16 nulos = df.isnull().sum()
17 print("Valores nulos por columna:")
18 print(nulos)
19
20 print("\nValores únicos en la columna 'fuma':")
21 print(df['fuma'].unique())
22
23
Terminal (Ctrl+ñ)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] [ ] [ ] [ ] [ ]
8 9 60 F ... Sí Amazonía 29/06/2024
9 10 47 M ... SI Orinoquía 29/04/2025

[10 rows x 13 columns]
Duplicados: 17
Valores nulos por columna:
id 0
edad 0
sexo 0
peso 51
talla 50
pas 0
pad 0
glucosa 50
colesterol 49
actividad_fisica_(min/sem) 0
fuma 0
región 0
fecha_de_tamizaje 0
dtype: int64

Valores únicos en la columna 'fuma':
['Sí' 'No' 'SI' 'SI' 'NO' 'desconocido' 'True' 'TRUE' 'FALSE' 'No sé']
PS C:\Users\Usuario\Desktop\Ing. Biomédica\U 2025-1\Programación\T_RestrepoPorras_Chaux>T_RestrepoPorras_Chaux
```

File Edit Selection ...

T\_RestrepoPorras\_ChauX

main.py M x biometria\_pacientes.csv

EXPLORER

T\_RESTREPOPORRAS\_CHA...  
biometria\_pacientes.c...  
main.py M  
README.md

main.py > ...  
31  
32 #punto 5  
33 df['fuma'] = df['fuma'].str.lower().str.strip()  
34 df['fuma'] = df['fuma'].replace({  
35 'sí': True, 'si': True, 'fuma': True,  
36 'no': False, 'desconocido': False, 'nunca': False, 'no fuma': False  
37 })  
38  
39 #punto 6  
40 df['fecha\_de\_tamizaje'] = pd.to\_datetime(df['fecha\_de\_tamizaje'], dayfirst=  
41 df['mes'] = df['fecha\_de\_tamizaje'].dt.month  
42 print(df['fecha\_de\_tamizaje'])  
43 print(df['mes'])  
44  
45 #Parte 2  
46  
47 #Punto 1  
48  
49

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + - - - ^ x

```
[ 'Sí' 'No' 'SI' 'SI' 'NO' 'desconocido' 'True' 'TRUE' 'FALSE' 'No sé']  
0      2024-10-02  
1      2025-03-04  
2      2025-03-25  
3      2024-09-30  
5      2024-11-30  
...  
997    2024-11-24  
999    2024-06-27  
1002    2025-03-16  
1005    2025-03-29  
1014    2024-10-10  
Name: fecha_de_tamizaje, Length: 817, dtype: datetime64[ns]  
0      10  
1       3  
2       3  
3       9  
5      11  
...  
997    11  
999     6  
1002     3  
1005     3  
1014    10  
Name: mes, Length: 817, dtype: int32  
PS C:\Users\Usuario\Desktop\Ing. Biomédica\U 2025-1\Programación\T_RestrepoPorras_ChauX\
```

> TIMELINE

main\* 0 0 0

Ln 49, Col 1 Spaces: 4 UTF-8 CRLF () Python 3.12.4 Go Live

File Edit Selection ...

T\_RestrepoPorras\_Chau

main.py x biometria\_pacientes.csv

main.py x

biometria\_pacientes.c...

main.py M

README.md

73 (df['colesterol'] > 240).astype(int) +

74 (df['imc'] > 30).astype(int) +

75 (df['sedentario']).astype(int)) >= 2

76 )

77

78 print(df)

79

80 #Parte 3

81

82 #Punto 12

83 riesgo\_por\_region = df.groupby('región')['metabólicamente\_alterado'].mean

84 print("\nRiesgo metabólico por región:")

85 print(riesgo\_por\_region)

86

87 #Punto 13

88 sedentarismo\_por\_mes = df.groupby('mes')['sedentario'].mean().sort\_value

89 print("\nSedentarismo por mes:")

90 print(sedentarismo\_por\_mes)

91

92

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + -

Riesgo metabólico por región:

región

Caribe 0.224242

Amazonía 0.215569

Andina 0.210191

Orinoquía 0.192982

Pacífica 0.152866

Name: metabólicamente\_alterado, dtype: float64

Sedentarismo por mes:

mes

12 0.263889

4 0.253333

10 0.245902

2 0.213115

6 0.200000

9 0.189406

3 0.184211

7 0.178571

8 0.171875

5 0.154930

1 0.150000

11 0.148148

Name: sedentario, dtype: float64

PS C:\Users\Usuario\Desktop\Ing. Biomédica\U 2025-1\Programación\T\_RestrepoPorras\_Chau\

main\*

Ln 92, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 Go Live

File Edit Selection ... T\_RestrepoPorras\_Chau

EXPLORER ... main.py M X biometria\_pacientes.csv

T\_RESTREPOPORRAS\_CHA...  
biometria\_pacientes.c...  
main.py M  
README.md

```
52 def clasificar_imc(imc):  
53     if imc < 18.5:  
54         return 'Bajo peso'  
55     elif imc < 25:  
56         return 'Normal'  
57     elif imc < 30:  
58         return 'Sobrepeso'  
59     else:  
60         return 'Obesidad'  
61  
62 df['clasificación_imc'] = df['imc'].apply(clasificar_imc)  
63  
64 #Punto 9  
65 df['sedentario'] = df['actividad_física_(min/sem)'] < 60  
66  
67 #punto 10  
68 df['hipertenso'] = (df['pas'] >= 140) | (df['pad'] >= 90)  
69  
70 #Punto 11  
71 df['metabólicamente_alterado'] = (  
72     ((df['glucosa'] > 126).astype(int) +  
73     (df['colesterol'] > 240).astype(int) +  
74     (df['imc'] > 30).astype(int) +  
75     (df['sedentario']).astype(int)) >= 2  
76 )
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + -

```
2      3  
3      9  
5     11  
..  
997    11  
999      6  
1002     3  
1005     3  
1014    10  
Name: mes, Length: 817, dtype: int32  
   id  edad  sexo  ...  sedentario  hipertenso  metabólicamente_alterado  
0    1    75    F  ...      False      True             False  
1    2    50    M  ...      False      True             False  
2    3    68    F  ...      False      False             True  
3    4    43    M  ...      False      True             False  
5    6    35    M  ...      False      False             False  
...  ...  ...  ...  ...      ...      ...             ...  
997  998    20    F  ...      False      True             False  
999 1000    76    M  ...      False      True             False  
1002 198    24    M  ...      False      True             False  
1005  33    73    M  ...      False      True             False  
1014 613    58    M  ...      False      True             False  
[817 rows x 19 columns]  
PS C:\Users\Usuario\Desktop\Ing. Biomédica\U 2025-1\Programación\T_RestrepoPorras_Chau\  
T_RestrepoPorras_Chau>
```

> TIMELINE

main\* 0 0 0 Ln 80, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 Go Live