

RESULTADOS DE LAS HISTORIAS CLÍNICAS

Proyecto 14INV-457 "Programa de diagnóstico precoz del cáncer de próstata"

Informe técnico de avance núm. 8

El análisis de datos se realizó con Python 3.8 (Anaconda Distribution 2020-07, Anaconda, Inc., Austin, TX), usando las librerías *Pandas* y *Numpy* para el análisis estadístico, y las librerías *Matplotlib* y *Seaborn* para los gráficos. Los datos fueron analizados por el investigador principal del proyecto Prof. Dr. Alcides Chaux.

- **Fecha de cierre del periodo de análisis:** 10 de septiembre de 2019
- **Número de historias clínicas a la fecha:** 447 pacientes

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

from matplotlib import pyplot as plt
import seaborn as sns
sns.set_theme(rc={'figure.figsize':[12,8]})
```

```
In [2]: df = pd.read_pickle('../Datos/ITA_8.pkl')
```

```
In [3]: df.shape
```

```
Out[3]: (447, 37)
```

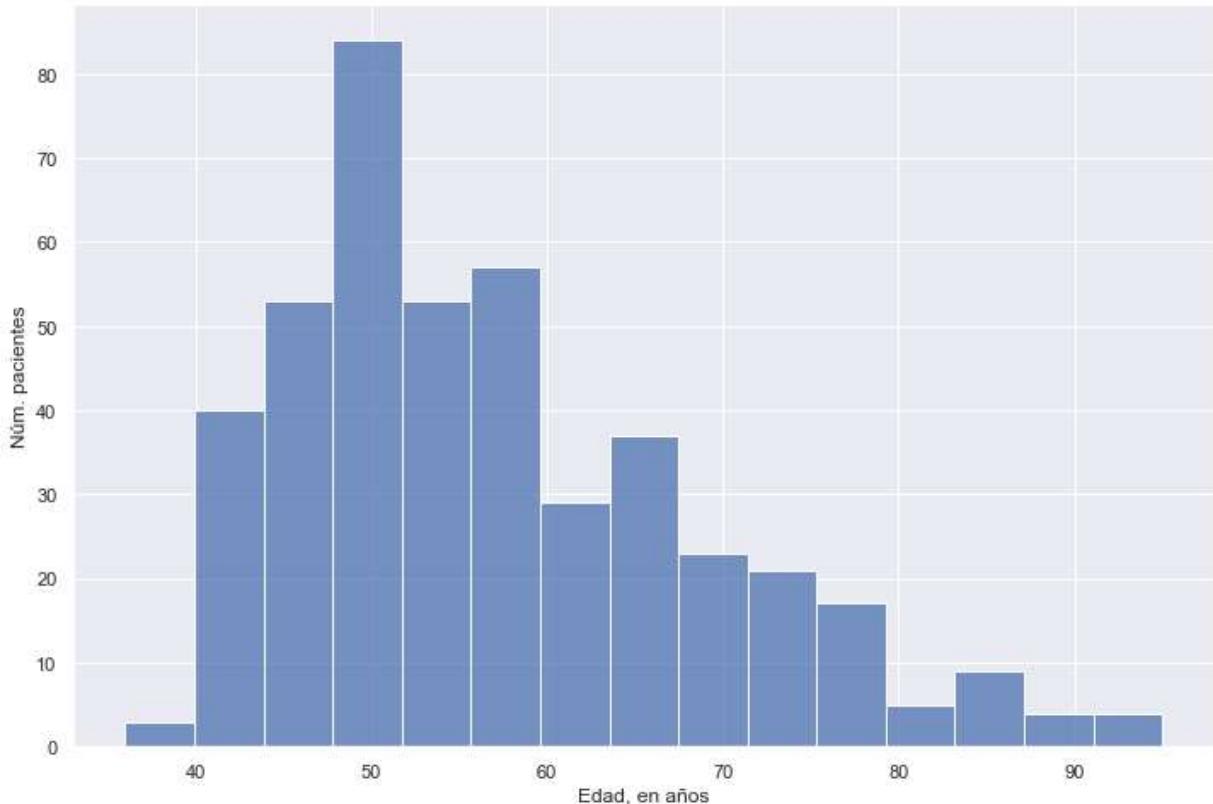
Datos demográficos

Edad

```
In [4]: df['edad'].describe().round()
```

```
Out[4]: count    439.0
mean      57.0
std       12.0
min      36.0
25%      48.0
50%      54.0
75%      64.0
max     95.0
Name: edad, dtype: float64
```

```
In [5]: sns.histplot(x='edad', data=df)
plt.xlabel('Edad, en años')
plt.ylabel('Núm. pacientes')
plt.show()
```



Procedencia

```
In [6]: # Las 10 ciudades de procedencia más frecuentes  
df['ciudad'].value_counts().head(10)
```

```
Out[6]: Asunción      134  
Luque          52  
San Lorenzo    41  
Capiatá        30  
Ferndo de la Mora 29  
Lambaré         26  
Villa Elisa     18  
Mariano Roque Alonso 15  
Ñemby           13  
Itá              11  
Name: ciudad, dtype: int64
```

Datos antropométricos

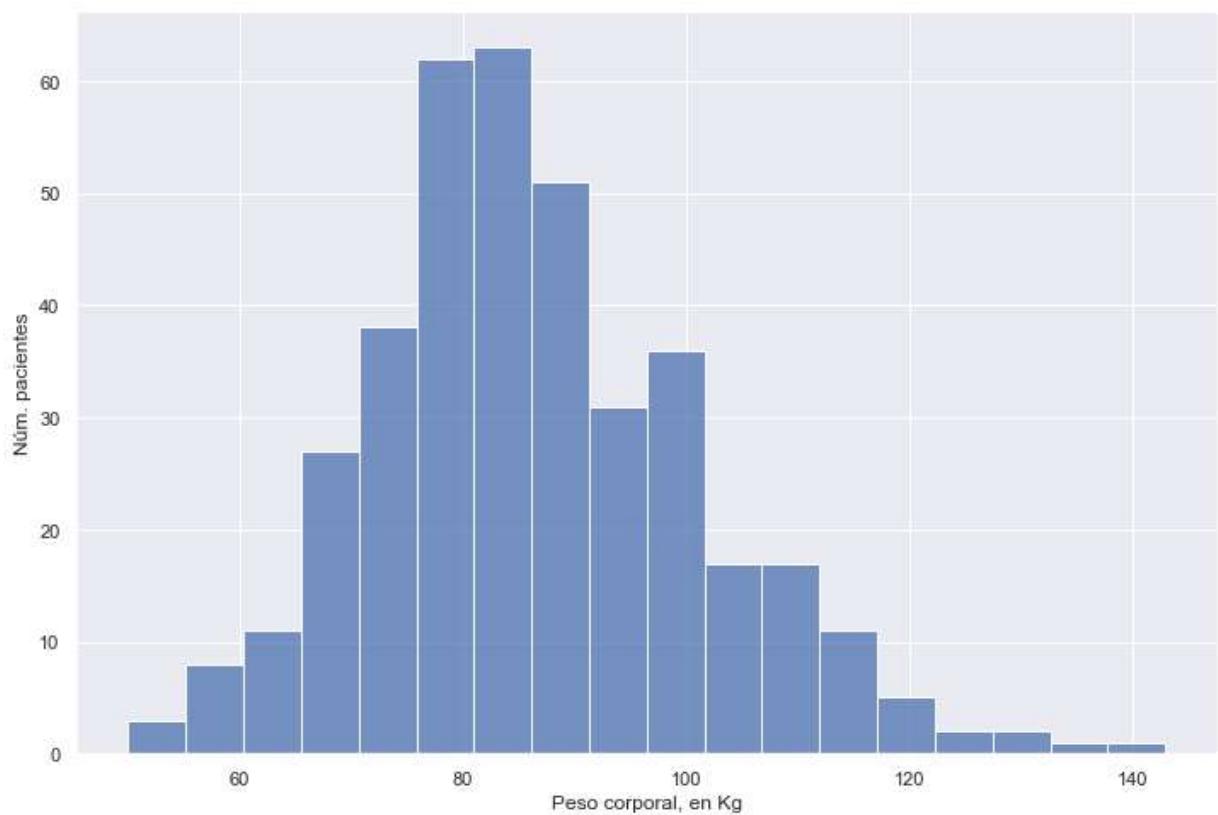
Peso, en Kg

```
In [7]: df['peso'].describe().round()
```

```
Out[7]: count    386.0  
mean     87.0  
std      15.0  
min      50.0  
25%     77.0  
50%     85.0  
75%     96.0  
max     143.0  
Name: peso, dtype: float64
```

```
In [8]:
```

```
sns.histplot(x='peso', data=df)
plt.xlabel('Peso corporal, en Kg')
plt.ylabel('Núm. pacientes')
plt.show()
```

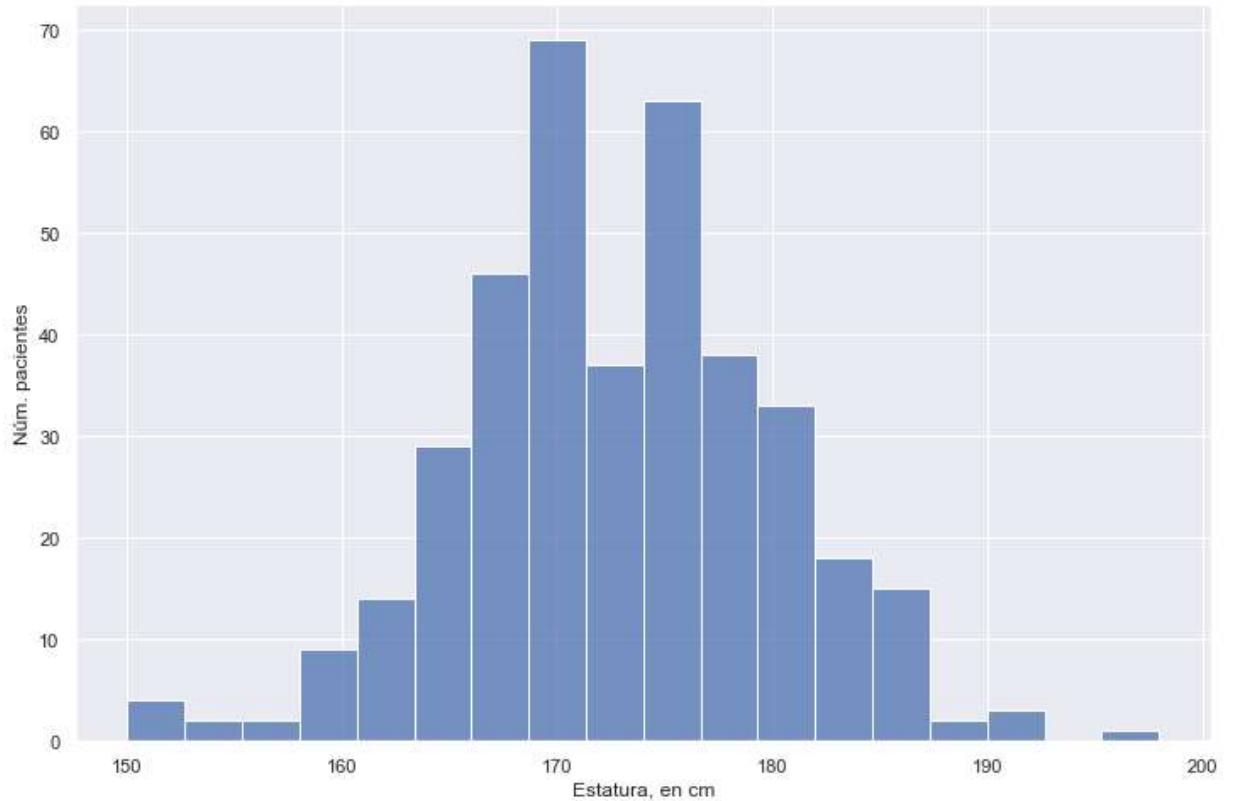


Estatura, en cm

```
In [9]: df['estatura'].describe().round()
```

```
Out[9]: count    385.0
mean     173.0
std      7.0
min     150.0
25%    168.0
50%    172.0
75%    178.0
max    198.0
Name: estatura, dtype: float64
```

```
In [10]: sns.histplot(x='estatura', data=df)
plt.xlabel('Estatura, en cm')
plt.ylabel('Núm. pacientes')
plt.show()
```

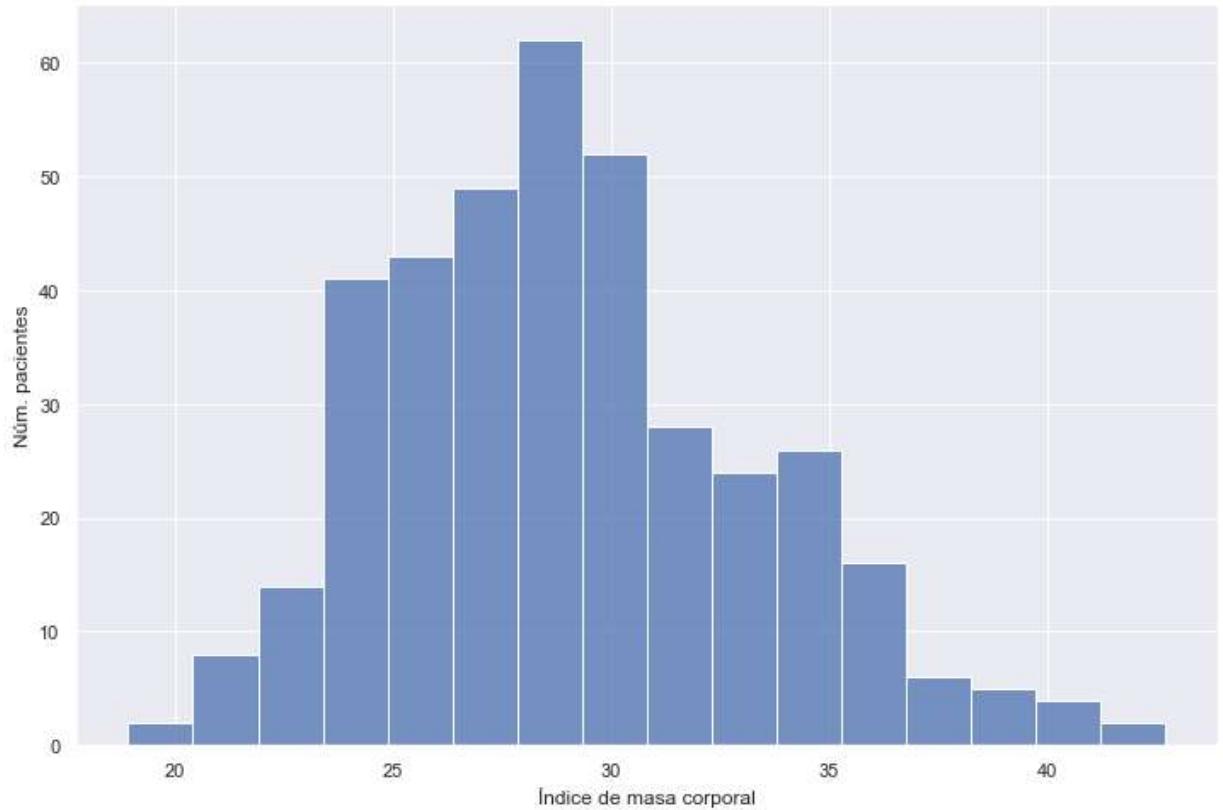


Índice de masa corporal

```
In [11]: df['imc'] = df['peso']/(df['estatura']/100)**2  
df['imc'].describe().round()
```

```
Out[11]: count    382.0  
mean     29.0  
std      4.0  
min     19.0  
25%    26.0  
50%    29.0  
75%    31.0  
max    43.0  
Name: imc, dtype: float64
```

```
In [12]: sns.histplot(x='imc', data=df)  
plt.xlabel('Índice de masa corporal')  
plt.ylabel('Nºm. pacientes')  
plt.show()
```



Antecedentes patológicos

Hipertensión arterial

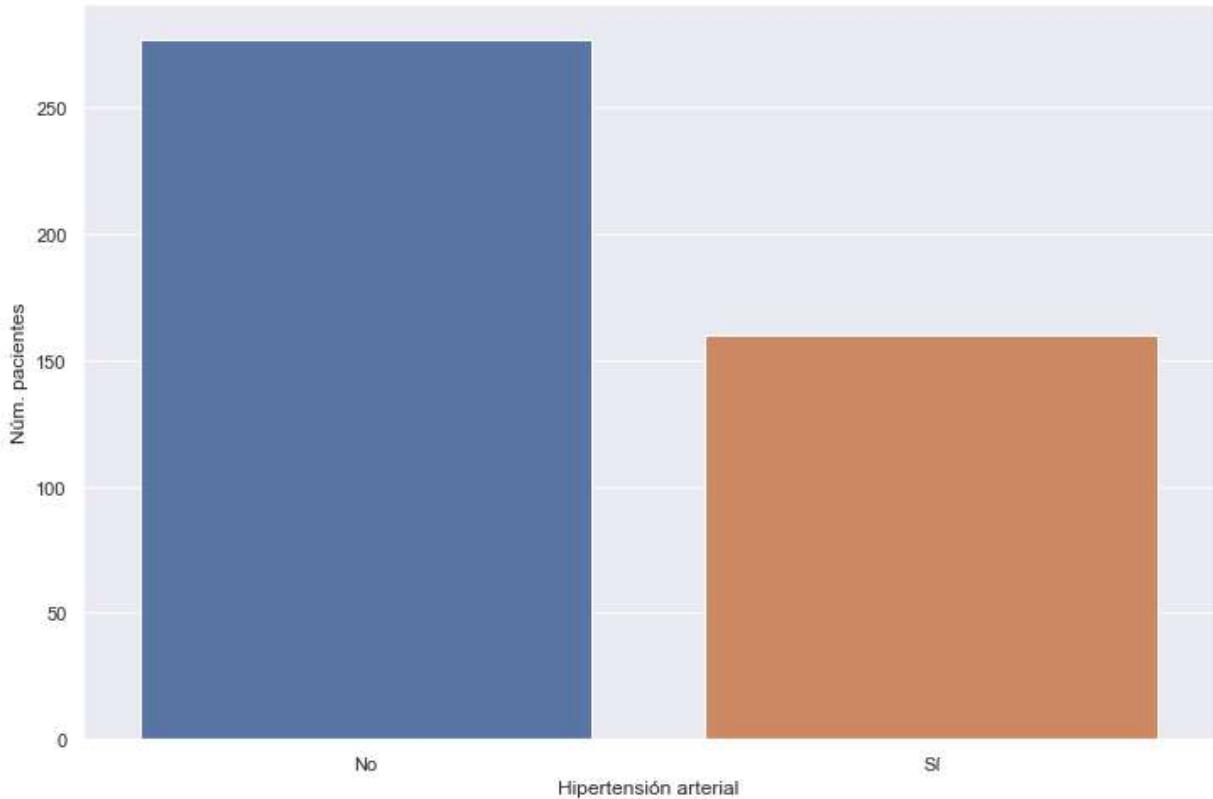
```
In [13]: df['hta'].value_counts(sort=False)
```

```
Out[13]: No    277  
Sí     160  
Name: hta, dtype: int64
```

```
In [14]: df['hta'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[14]: No     0.63  
Sí     0.37  
Name: hta, dtype: float64
```

```
In [15]: sns.countplot(x='hta', data=df)  
plt.xlabel('Hipertensión arterial')  
plt.ylabel('Número de pacientes')  
plt.show()
```



Diabetes mellitus

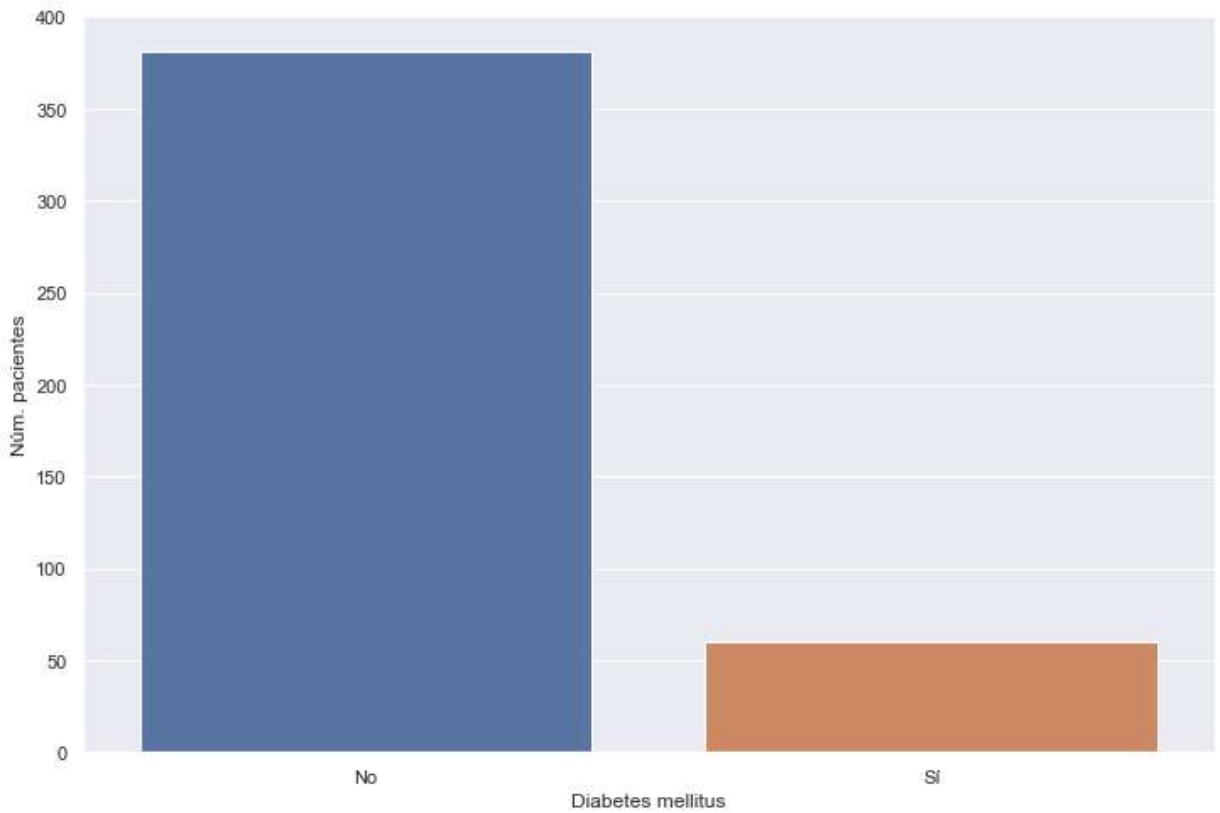
```
In [16]: df['dm'].value_counts(sort=False)
```

```
Out[16]: No    381
          Sí    60
          Name: dm, dtype: int64
```

```
In [17]: df['dm'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[17]: No     0.86
          Sí     0.14
          Name: dm, dtype: float64
```

```
In [18]: sns.countplot(x='dm', data=df)
plt.xlabel('Diabetes mellitus')
plt.ylabel('Núm. pacientes')
plt.show()
```



Tumor no prostático

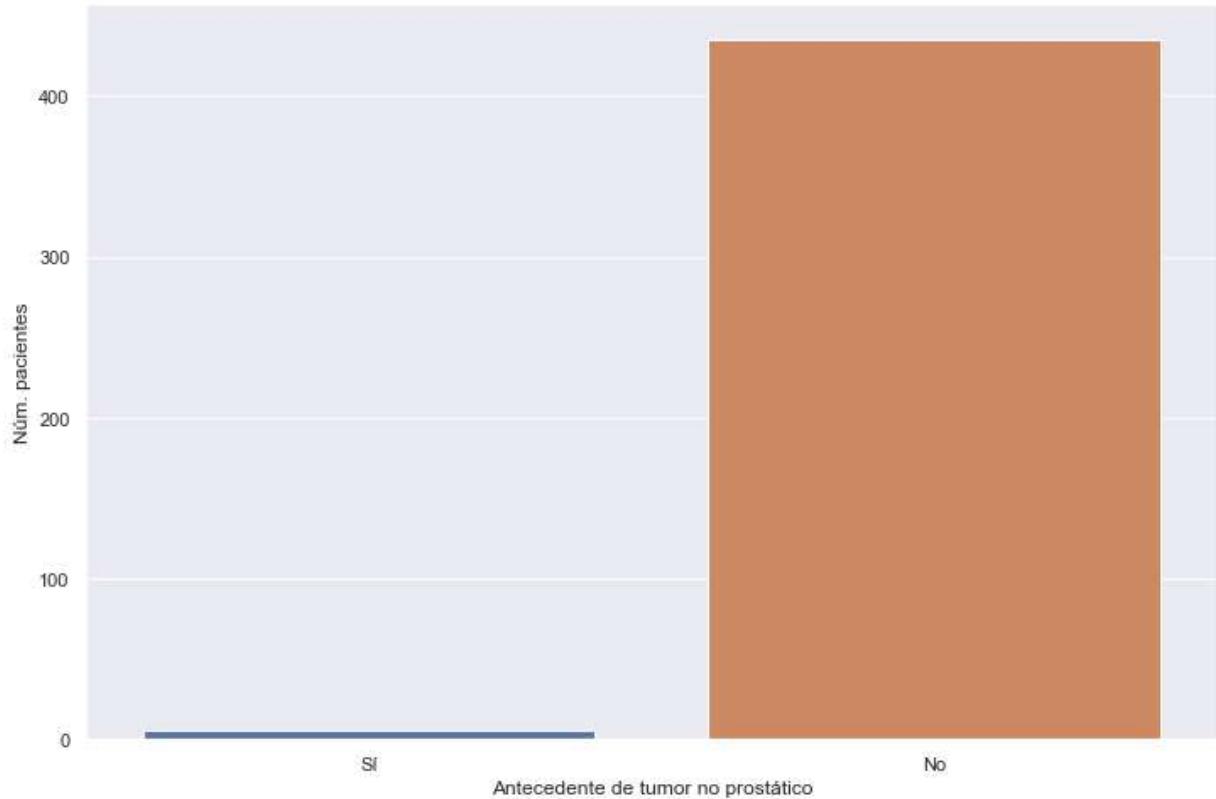
```
In [19]: df['tumor'].value_counts(sort=False)
```

```
Out[19]: No    435
          Sí     6
          Name: tumor, dtype: int64
```

```
In [20]: df['tumor'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[20]: No    0.99
          Sí    0.01
          Name: tumor, dtype: float64
```

```
In [21]: sns.countplot(x='tumor', data=df)
          plt.xlabel('Antecedente de tumor no prostático')
          plt.ylabel('Núm. pacientes')
          plt.show()
```



Tipo de tumor no prostático

```
In [22]: df['tumor_tipo'].value_counts()
```

```
Out[22]: Hipófisis      1  
Pólipos colónicos    1  
Mieloma múltiple     1  
Tiroïdes             1  
Name: tumor_tipo, dtype: int64
```

Hiperplasia prostática

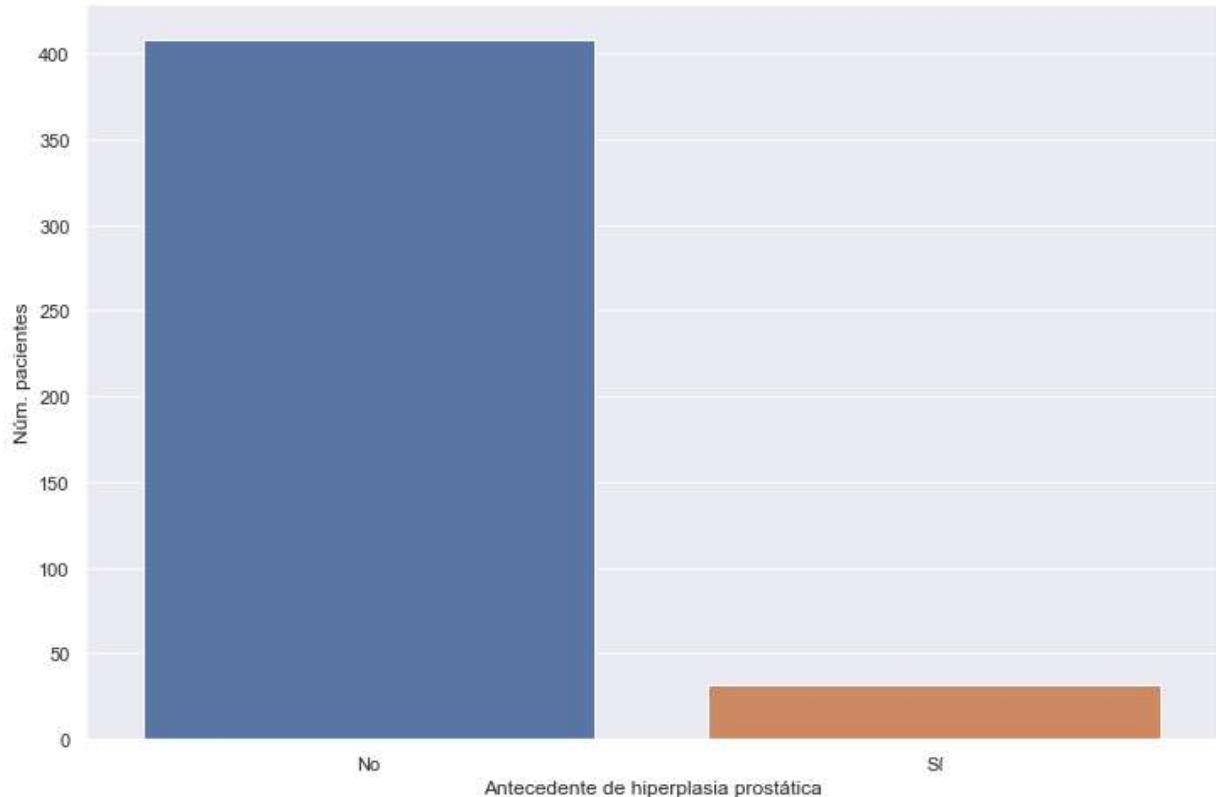
```
In [23]: df['hp'].value_counts(sort=False)
```

```
Out[23]: No    408  
Sí     32  
Name: hp, dtype: int64
```

```
In [24]: df['hp'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[24]: No    0.93  
Sí     0.07  
Name: hp, dtype: float64
```

```
In [25]: sns.countplot(x='hp', data=df)  
plt.xlabel('Antecedente de hiperplasia prostática')  
plt.ylabel('Nº. pacientes')  
plt.show()
```



Cáncer de próstata familiar

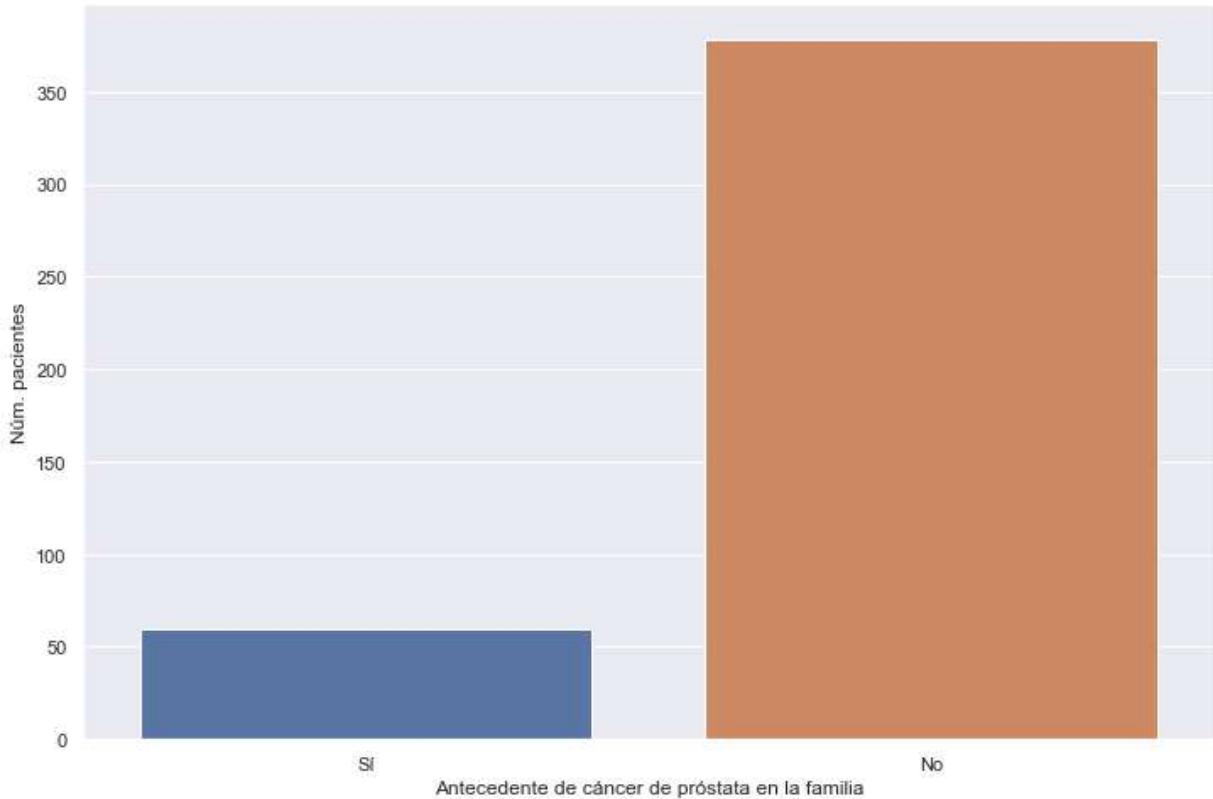
```
In [26]: df['tumor_fam'].value_counts(sort=False)
```

```
Out[26]: No    378  
Sí     60  
Name: tumor_fam, dtype: int64
```

```
In [27]: df['tumor_fam'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[27]: No    0.86  
Sí     0.14  
Name: tumor_fam, dtype: float64
```

```
In [28]: sns.countplot(x='tumor_fam', data=df)  
plt.xlabel('Antecedente de cáncer de próstata en la familia')  
plt.ylabel('Nºm. pacientes')  
plt.show()
```



Familiar con cáncer de próstata

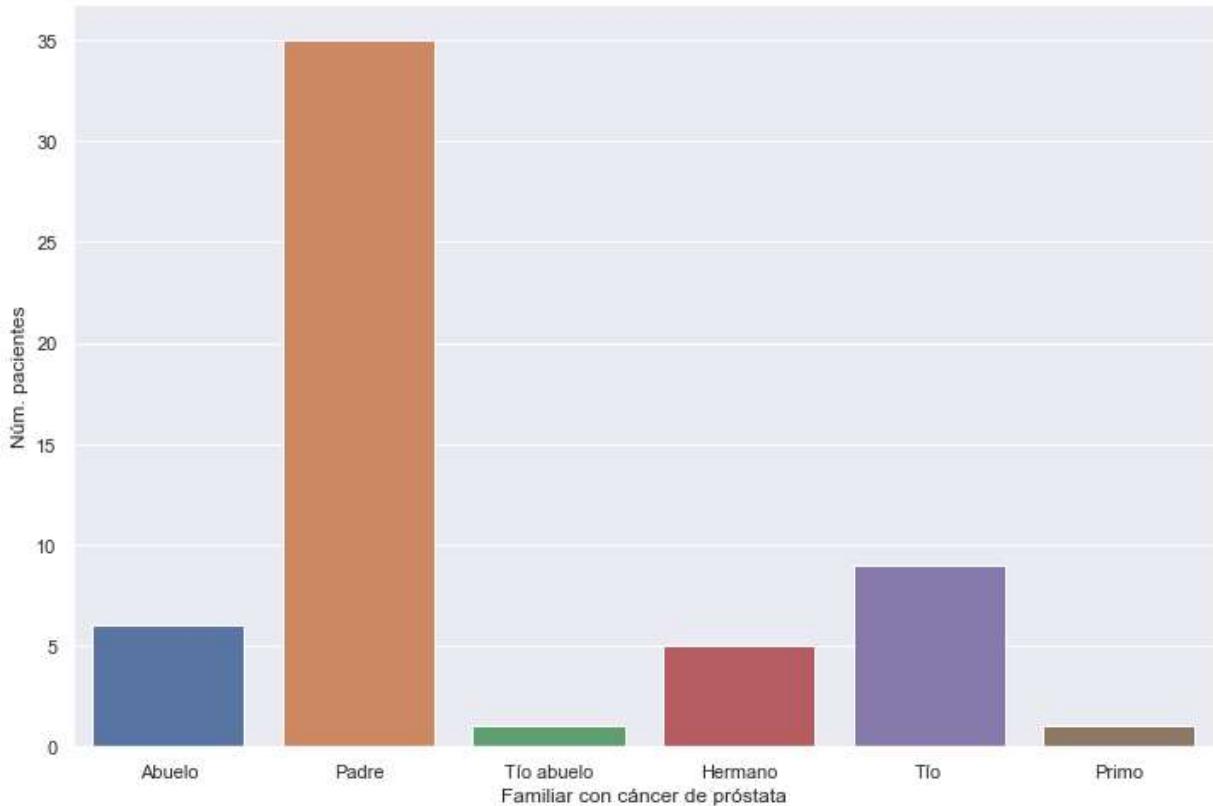
```
In [29]: df['tumor_fam_tipo'].value_counts()
```

```
Out[29]: Padre      35  
Tío          9  
Abuelo       6  
Hermano      5  
Primo         1  
Tío abuelo   1  
Name: tumor_fam_tipo, dtype: int64
```

```
In [30]: df['tumor_fam_tipo'].value_counts(normalize=True).round(2)
```

```
Out[30]: Padre      0.61  
Tío          0.16  
Abuelo       0.11  
Hermano      0.09  
Primo         0.02  
Tío abuelo   0.02  
Name: tumor_fam_tipo, dtype: float64
```

```
In [31]: sns.countplot(x='tumor_fam_tipo', data=df)  
plt.xlabel('Familiar con cáncer de próstata')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



Hábitos personales

Consumo de tabaco

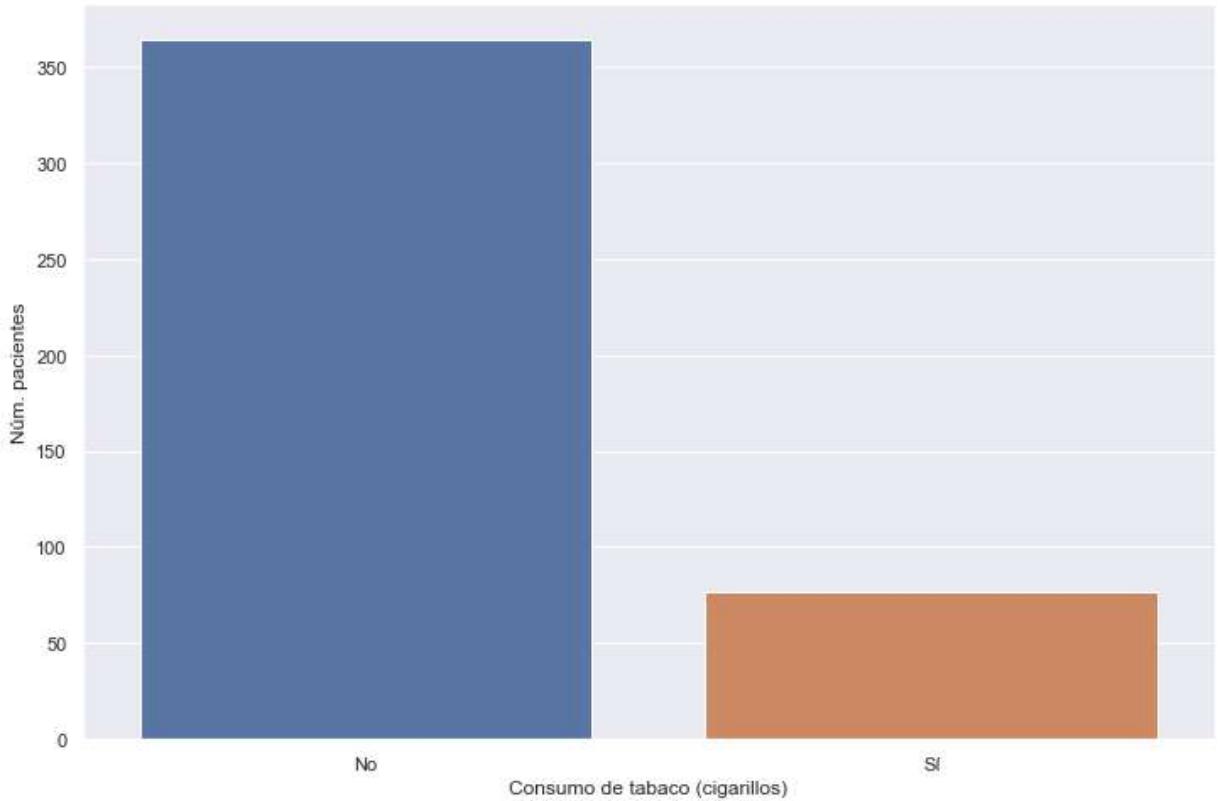
```
In [32]: df['tabaco'].value_counts(sort=False)
```

```
Out[32]: No      364  
Sí       77  
Name: tabaco, dtype: int64
```

```
In [33]: df['tabaco'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[33]: No      0.83  
Sí       0.17  
Name: tabaco, dtype: float64
```

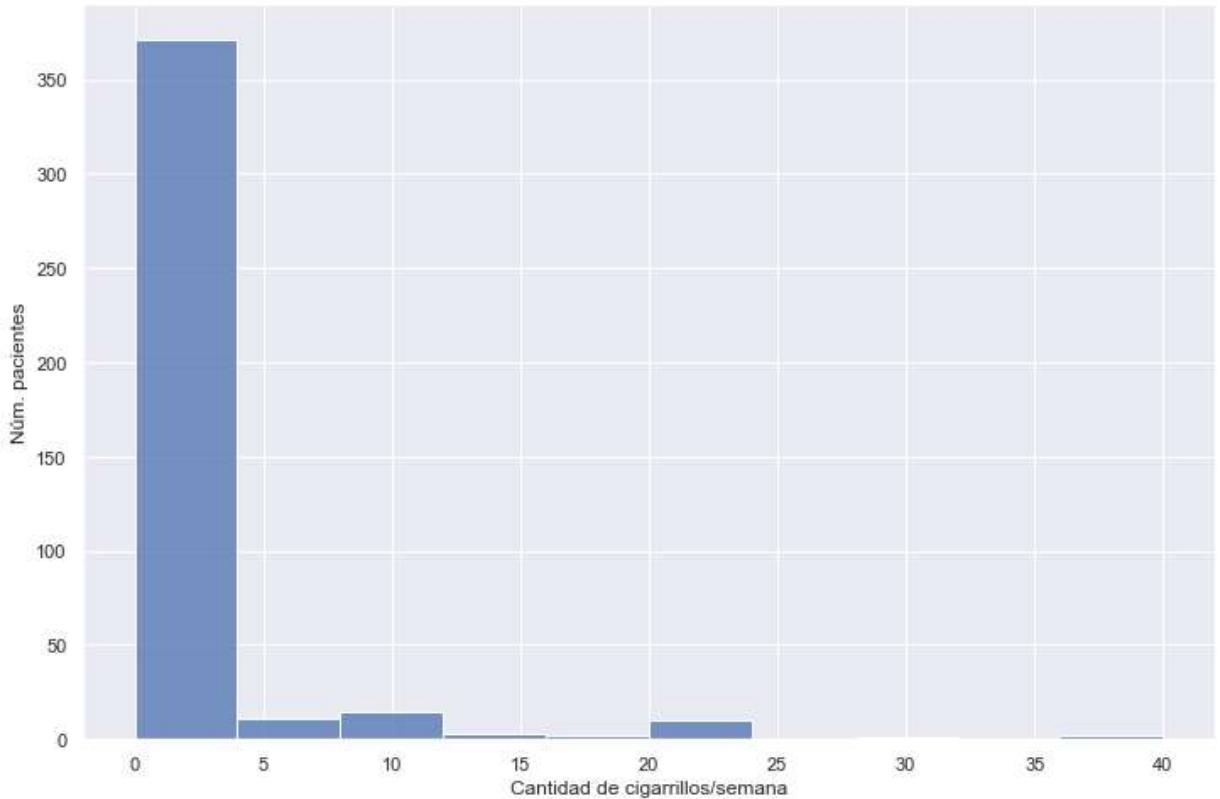
```
In [34]: sns.countplot(x='tabaco', data=df)  
plt.xlabel('Consumo de tabaco (cigarillos)')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



```
In [35]: df['tabaco_cantidad'].describe().round()
```

```
Out[35]: count    415.0
mean      2.0
std       5.0
min      0.0
25%     0.0
50%     0.0
75%     0.0
max     40.0
Name: tabaco_cantidad, dtype: float64
```

```
In [36]: sns.histplot(x='tabaco_cantidad', data=df)
plt.xlabel('Cantidad de cigarrillos/semana')
plt.ylabel('Nºm. pacientes')
plt.show()
```



Consumo de carne

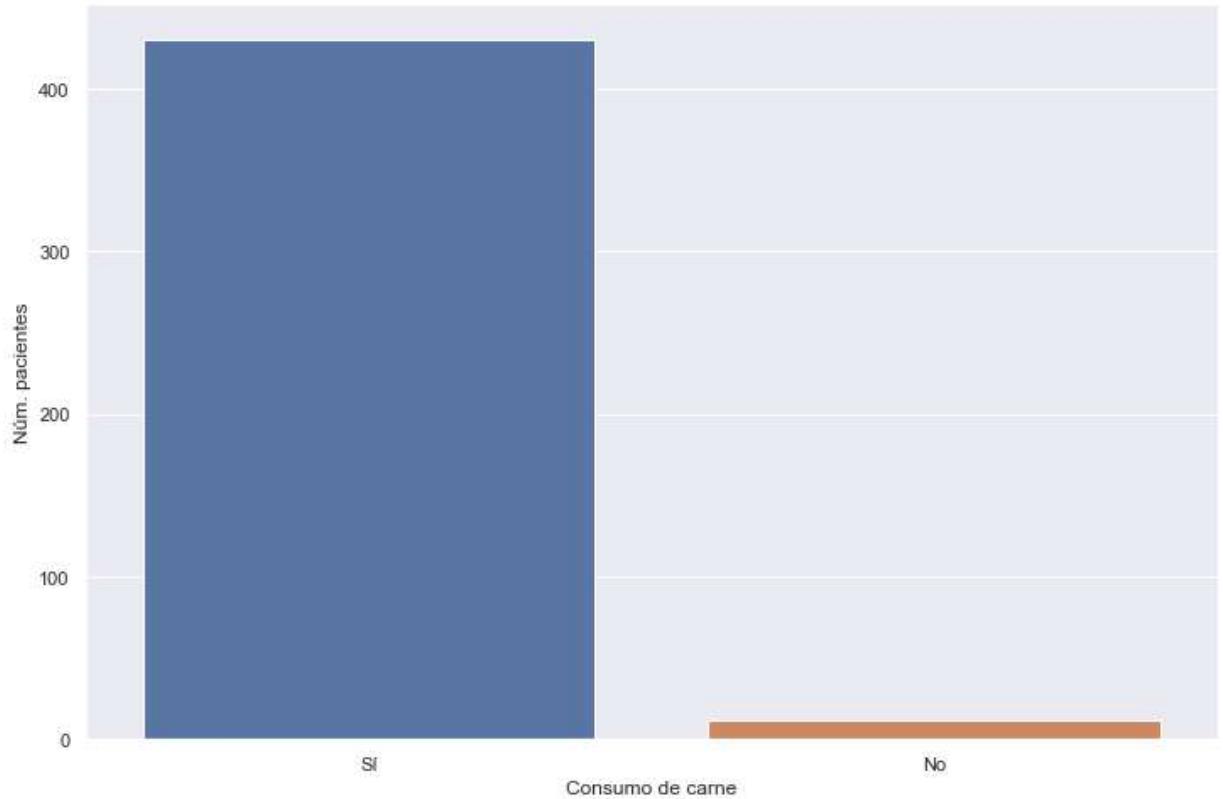
```
In [37]: df['carne'].value_counts(sort=False)
```

```
Out[37]: No      12
Sí     430
Name: carne, dtype: int64
```

```
In [38]: df['carne'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[38]: No      0.03
Sí     0.97
Name: carne, dtype: float64
```

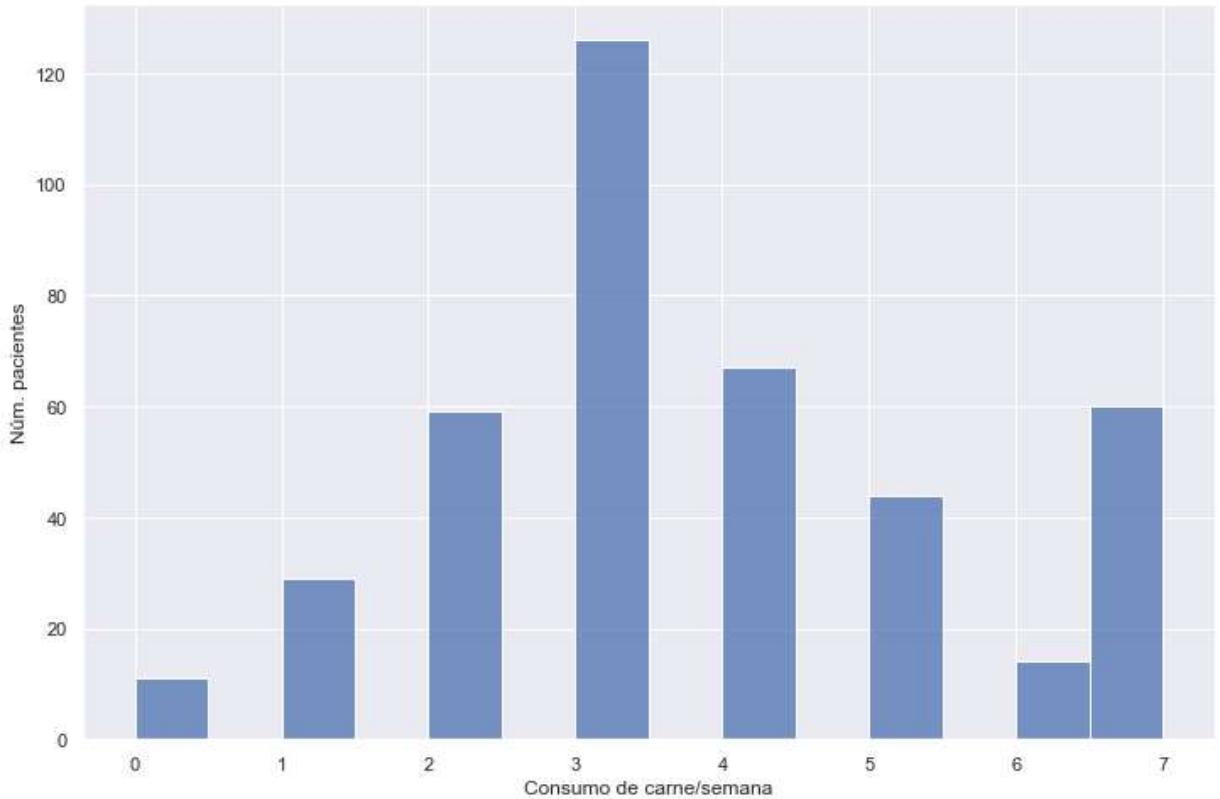
```
In [39]: sns.countplot(x='carne', data=df)
plt.xlabel('Consumo de carne')
plt.ylabel('Núm. pacientes')
plt.show()
```



```
In [40]: df['carne_frec'].describe().round()
```

```
Out[40]: count    410.0
mean      4.0
std       2.0
min       0.0
25%      3.0
50%      3.0
75%      5.0
max      7.0
Name: carne_frec, dtype: float64
```

```
In [41]: sns.histplot(x='carne_frec', data=df)
plt.xlabel('Consumo de carne/semana')
plt.ylabel('Nº. pacientes')
plt.show()
```



Consumo de frutas/verduras

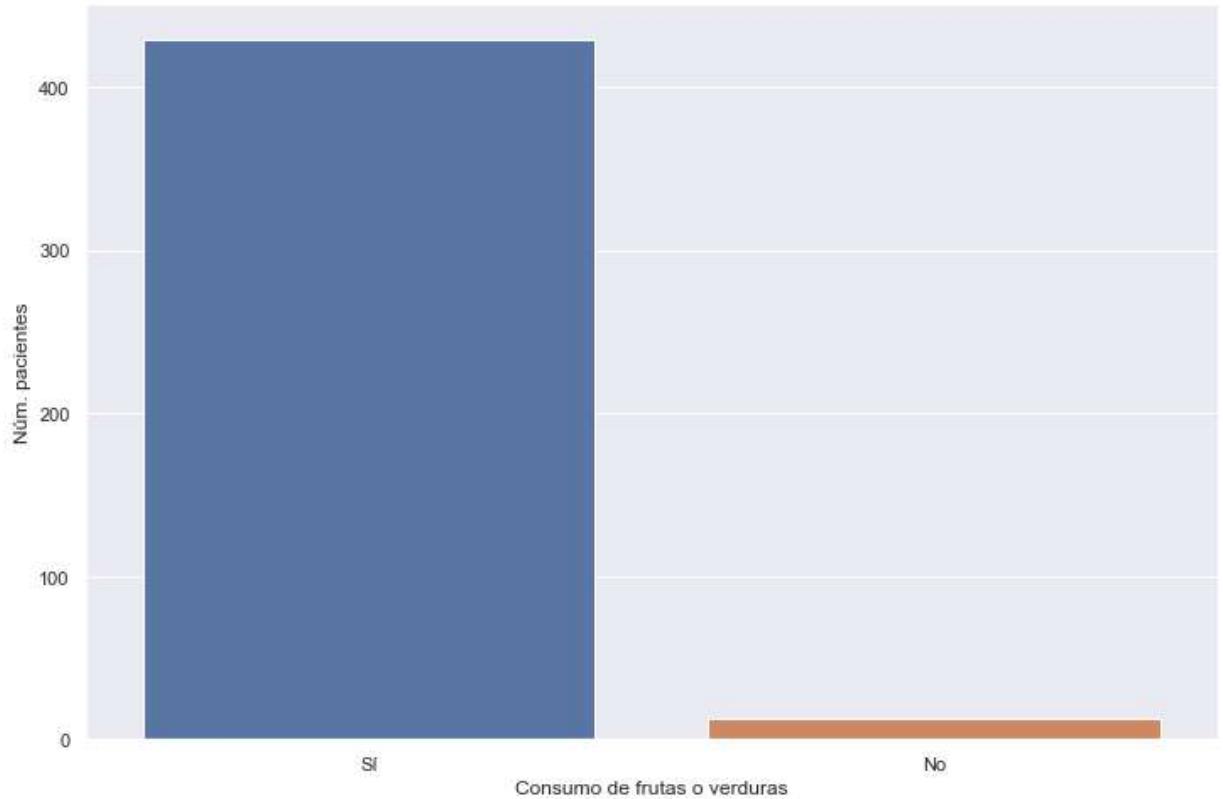
```
In [42]: df['verdura'].value_counts(sort=False)
```

```
Out[42]: No      13
Sí     429
Name: verdura, dtype: int64
```

```
In [43]: df['verdura'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[43]: No      0.03
Sí     0.97
Name: verdura, dtype: float64
```

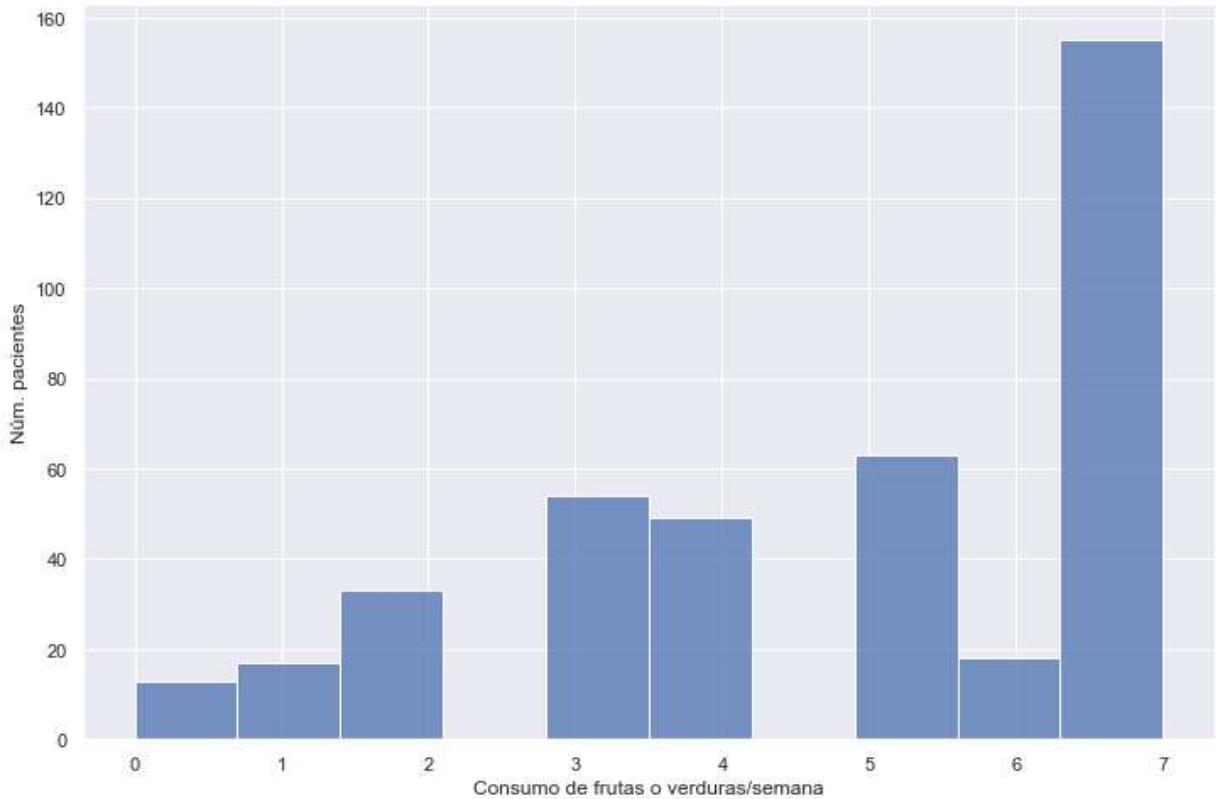
```
In [44]: sns.countplot(x='verdura', data=df)
plt.xlabel('Consumo de frutas o verduras')
plt.ylabel('Núm. pacientes')
plt.show()
```



```
In [45]: df['verdura_frec'].describe().round()
```

```
Out[45]: count    402.0
mean      5.0
std       2.0
min       0.0
25%      3.0
50%      5.0
75%      7.0
max      7.0
Name: verdura_frec, dtype: float64
```

```
In [46]: sns.histplot(x='verdura_frec', data=df)
plt.xlabel('Consumo de frutas o verduras/semana')
plt.ylabel('Nº. pacientes')
plt.show()
```



Actividad física

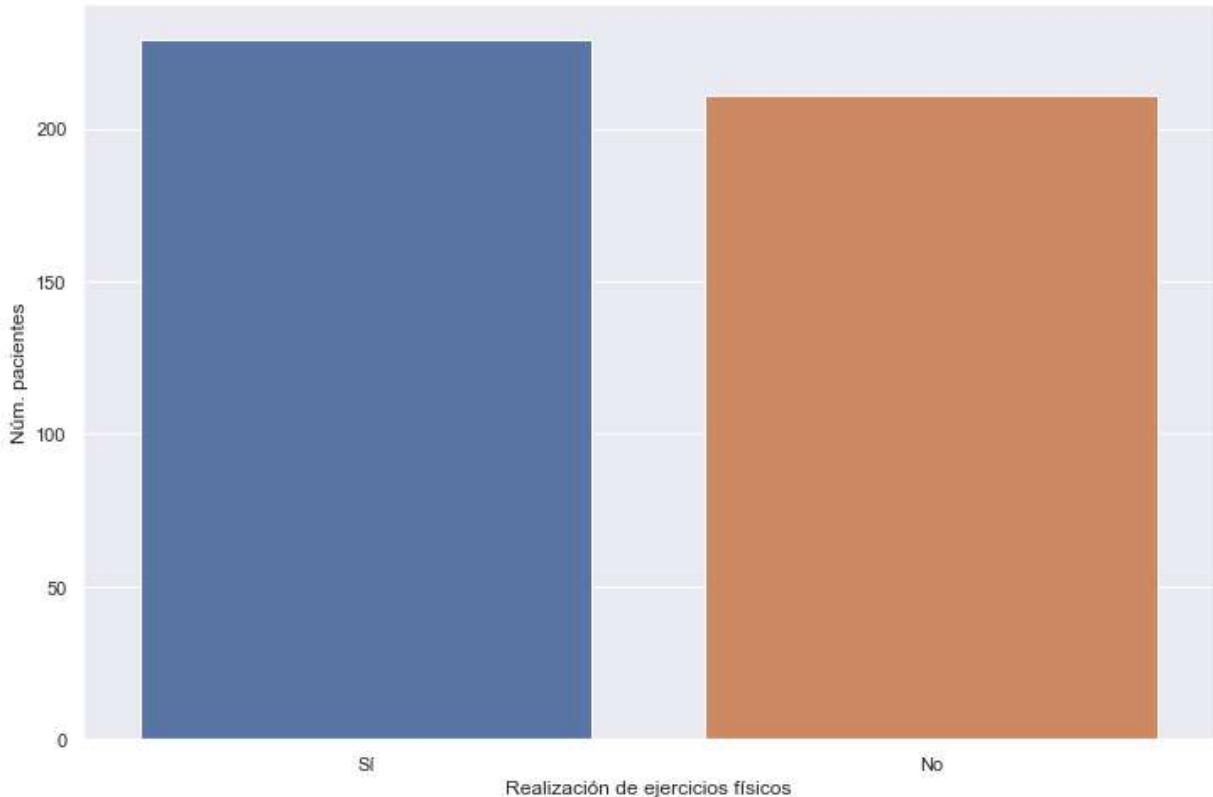
```
In [47]: df['ejercicio'].value_counts(sort=False)
```

```
Out[47]: No    211  
Sí    229  
Name: ejercicio, dtype: int64
```

```
In [48]: df['ejercicio'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[48]: No    0.48  
Sí    0.52  
Name: ejercicio, dtype: float64
```

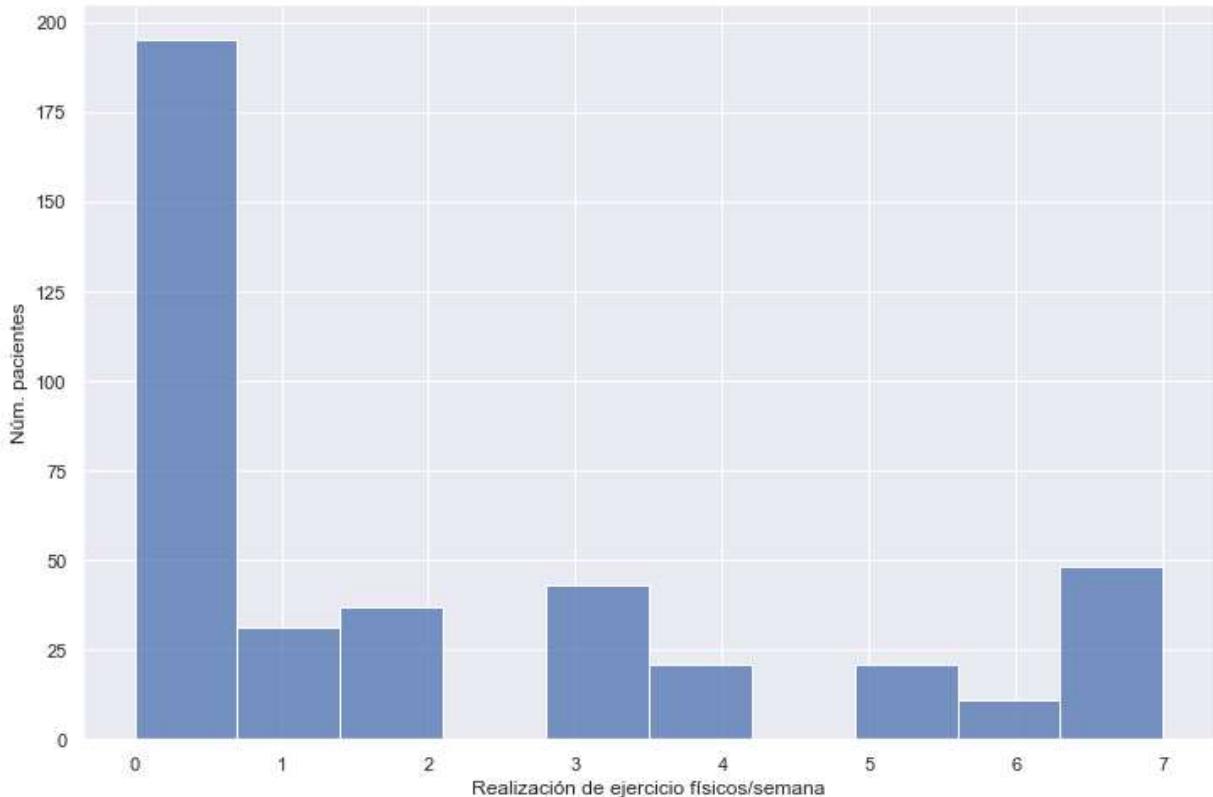
```
In [49]: sns.countplot(x='ejercicio', data=df)  
plt.xlabel('Realización de ejercicios físicos')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



```
In [50]: df['ejercicio_frec'].describe().round()
```

```
Out[50]: count    407.0
mean      2.0
std       2.0
min      0.0
25%      0.0
50%      1.0
75%      3.0
max      7.0
Name: ejercicio_frec, dtype: float64
```

```
In [51]: sns.histplot(x='ejercicio_frec', data=df)
plt.xlabel('Realización de ejercicio físicos/semana')
plt.ylabel('Nº. pacientes')
plt.show()
```



Síntomas urinarios

Dificultad para empezar a orinar

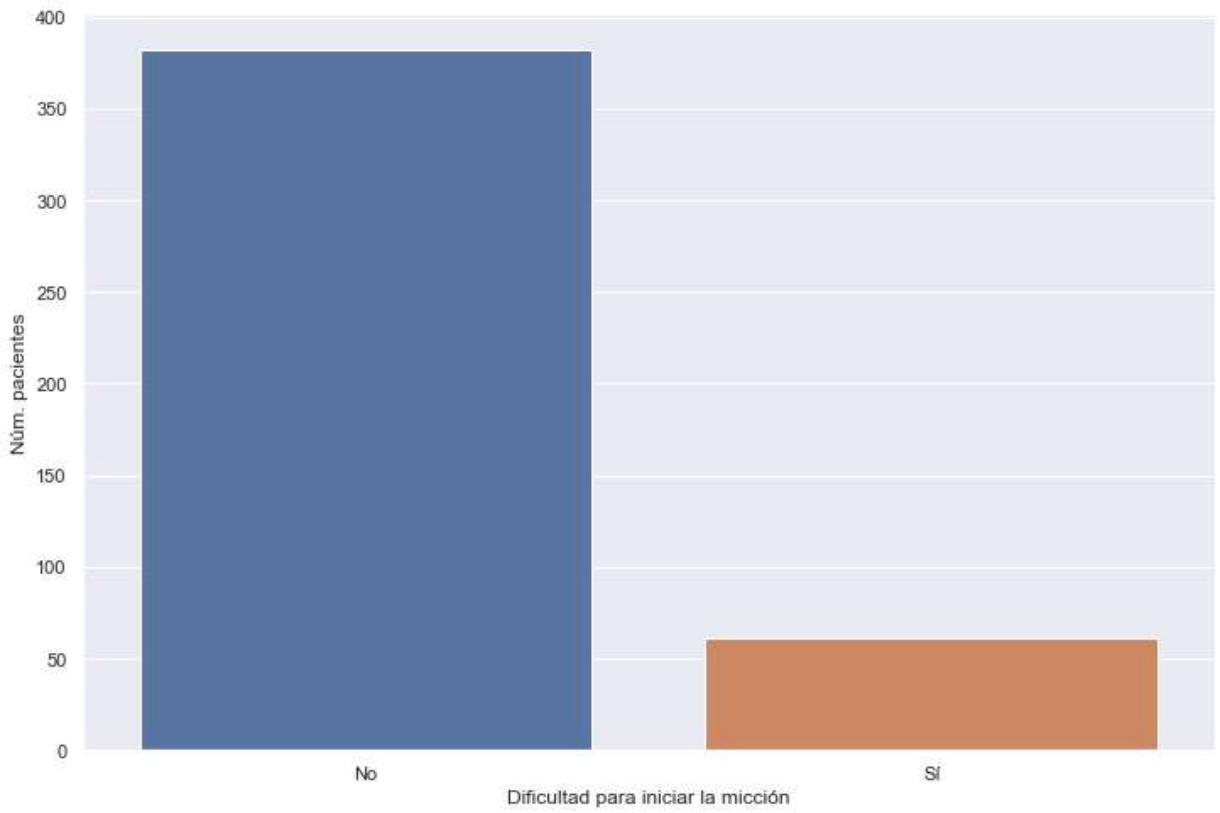
```
In [52]: df['miccion_inicio'].value_counts(sort=False)
```

```
Out[52]: No    382  
Sí     61  
Name: miccion_inicio, dtype: int64
```

```
In [53]: df['miccion_inicio'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[53]: No     0.86  
Sí     0.14  
Name: miccion_inicio, dtype: float64
```

```
In [54]: sns.countplot(x='miccion_inicio', data=df)  
plt.xlabel('Dificultad para iniciar la micción')  
plt.ylabel('Nºm. pacientes')  
plt.show()
```



Chorro de orina lento o débil

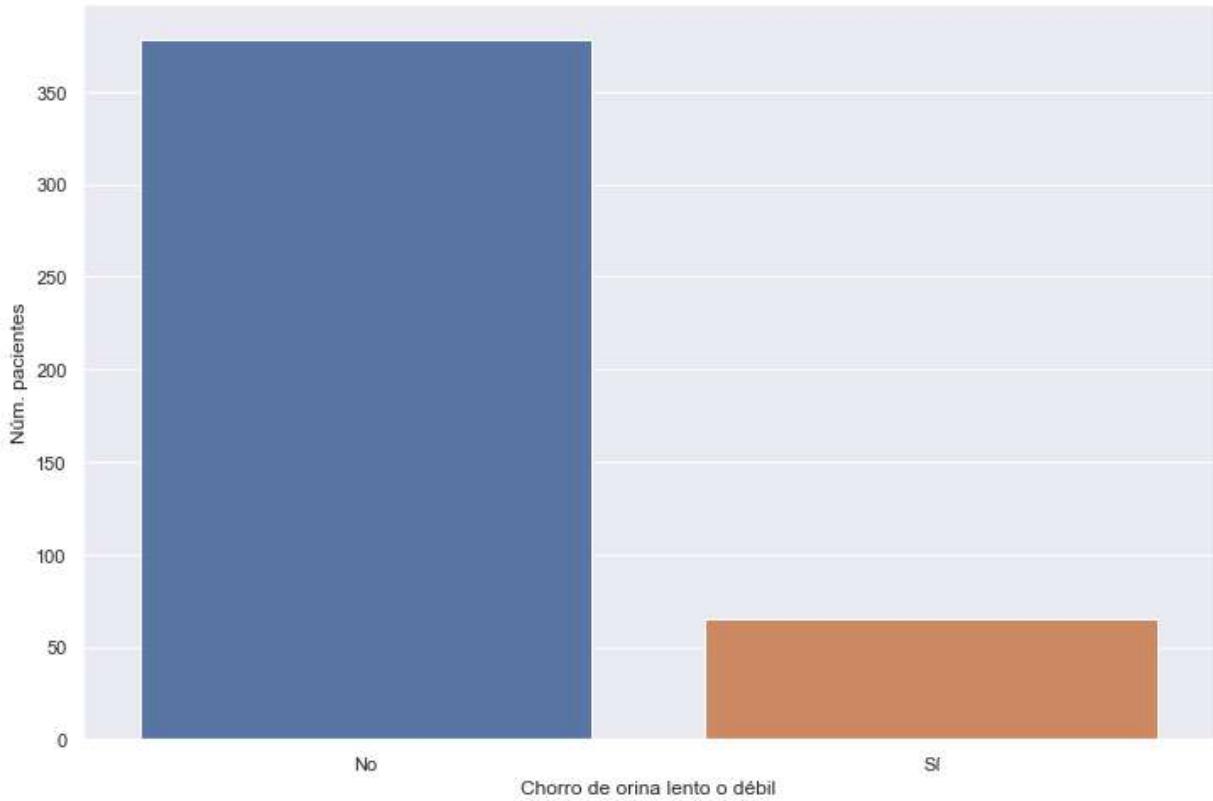
```
In [55]: df['miccion_lenta'].value_counts(sort=False)
```

```
Out[55]: No    378
          Sí    65
          Name: miccion_lenta, dtype: int64
```

```
In [56]: df['miccion_lenta'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[56]: No     0.85
          Sí     0.15
          Name: miccion_lenta, dtype: float64
```

```
In [57]: sns.countplot(x='miccion_lenta', data=df)
          plt.xlabel('Chorro de orina lento o débil')
          plt.ylabel('Núm. pacientes')
          plt.show()
```



Orinar frecuentemente (>10 veces/día)

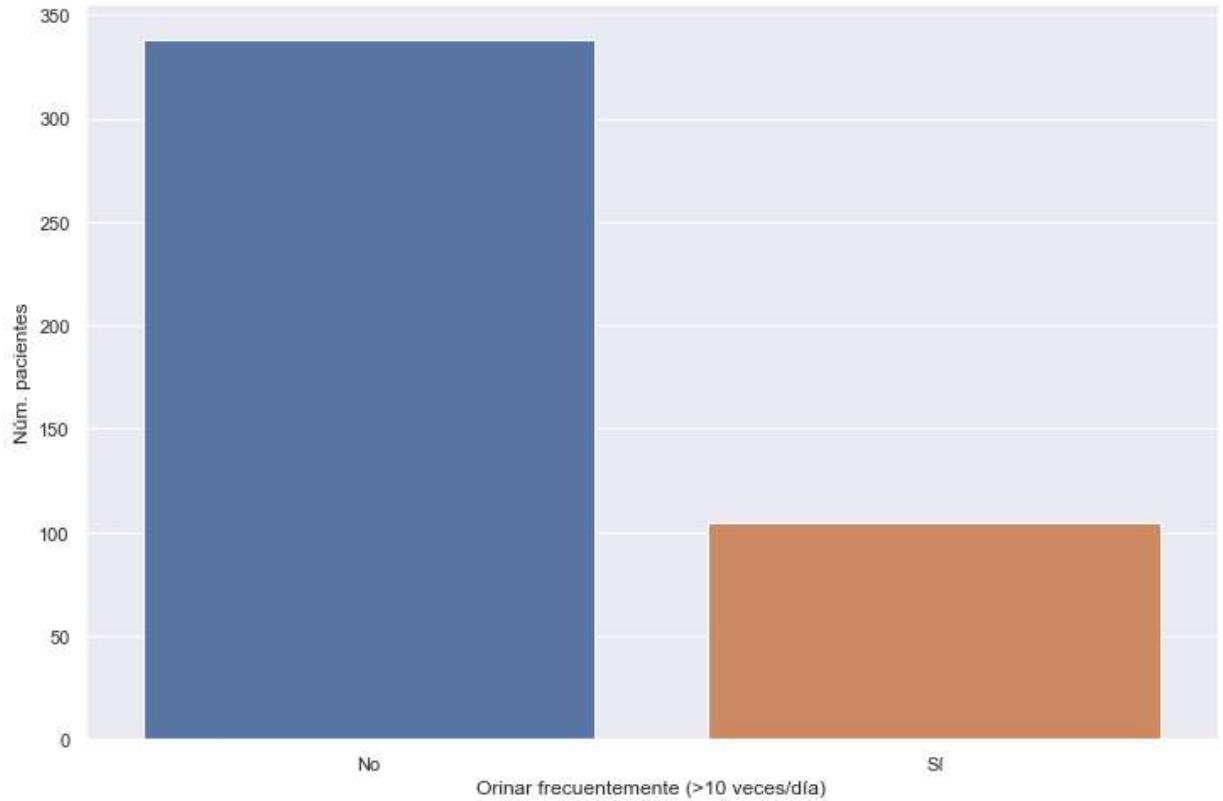
```
In [58]: df['miccion_frec'].value_counts(sort=False)
```

```
Out[58]: No    338
          Sí    105
          Name: miccion_frec, dtype: int64
```

```
In [59]: df['miccion_frec'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[59]: No     0.76
          Sí     0.24
          Name: miccion_frec, dtype: float64
```

```
In [60]: sns.countplot(x='miccion_frec', data=df)
          plt.xlabel('Orinar frecuentemente (>10 veces/día)')
          plt.ylabel('Nº. pacientes')
          plt.show()
```



Nicturia

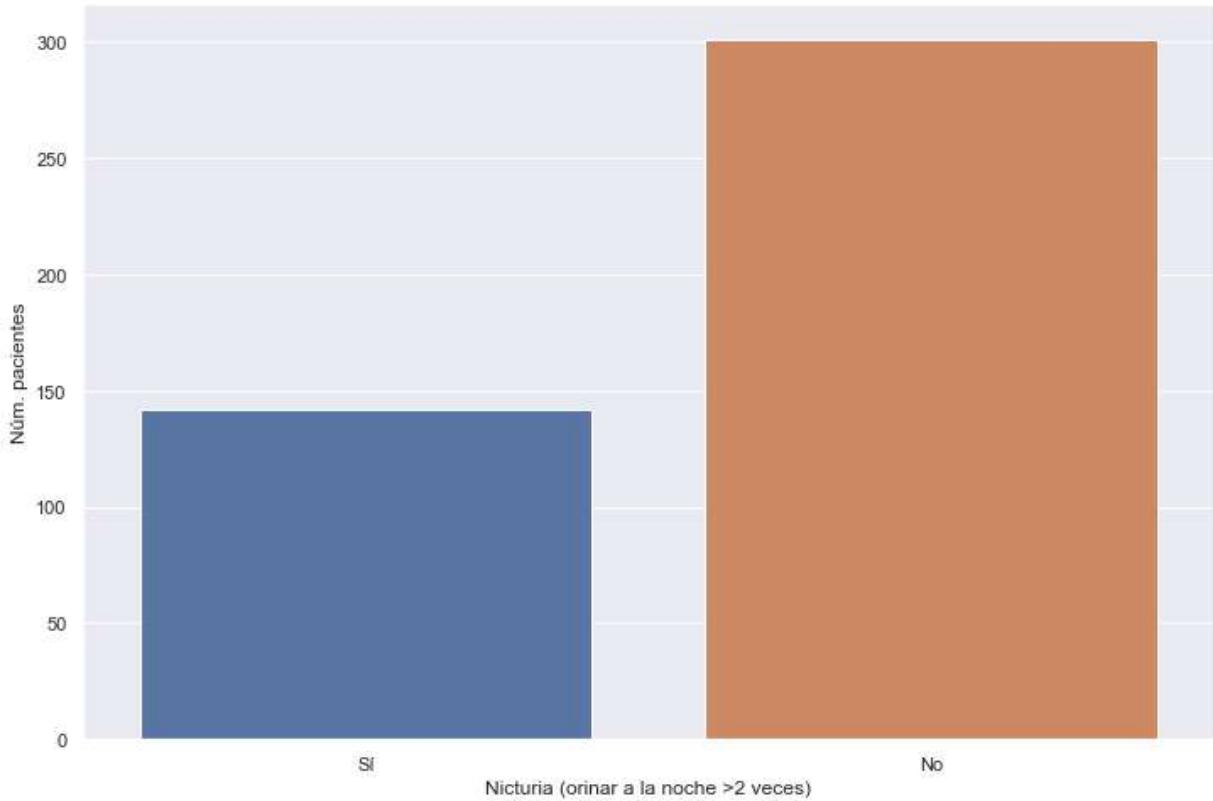
```
In [61]: df['nicturia'].value_counts(sort=False)
```

```
Out[61]: No    301  
Sí    142  
Name: nicturia, dtype: int64
```

```
In [62]: df['nicturia'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[62]: No     0.68  
Sí     0.32  
Name: nicturia, dtype: float64
```

```
In [63]: sns.countplot(x='nicturia', data=df)  
plt.xlabel('Nicturia (orinar a la noche >2 veces)')  
plt.ylabel('Nº. pacientes')  
plt.show()
```



Hematuria

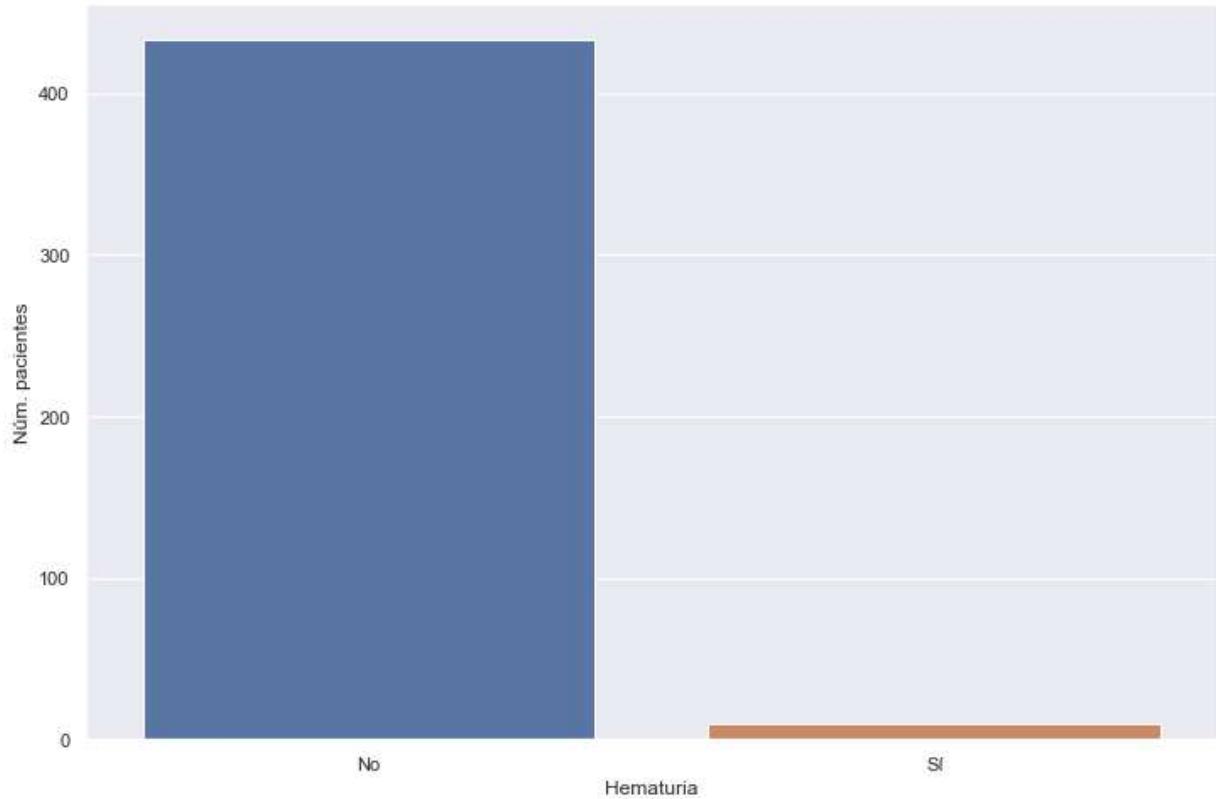
```
In [64]: df['hematuria'].value_counts(sort=False)
```

```
Out[64]: No    433
          Sí     10
          Name: hematuria, dtype: int64
```

```
In [65]: df['hematuria'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[65]: No      0.98
          Sí      0.02
          Name: hematuria, dtype: float64
```

```
In [66]: sns.countplot(x='hematuria', data=df)
          plt.xlabel('Hematuria')
          plt.ylabel('Núm. pacientes')
          plt.show()
```



Dolor lumbar bajo

```
In [67]: df['dolor_oseo'].value_counts(sort=False)
```

```
Out[67]: No    351
          Sí    92
          Name: dolor_oseo, dtype: int64
```

```
In [68]: df['dolor_oseo'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[68]: No     0.79
          Sí     0.21
          Name: dolor_oseo, dtype: float64
```

```
In [69]: sns.countplot(x='dolor_oseo', data=df)
          plt.xlabel('Dolor en la región lumbar baja')
          plt.ylabel('Nº. pacientes')
          plt.show()
```

