

## 9 - Parte EDA & Limpieza - Sección NPCJP

October 31, 2022

### 1 Análisis de la Encuesta Multiproposito

#### 1.1 Importación de Paquetes y carga del archivo

```
[52]: import pandas as pd
      from matplotlib import pyplot as plt
      import seaborn as sns
      from scipy import stats
      import numpy as np
```

```
[53]: data = pd.read_excel('Encuesta_Multiproposito_Suba.xlsx')
```

```
[54]: data.shape
```

```
[54]: (24536, 498)
```

Hay en total 24536 encuestados en la Localidad de Suba

### 2 Revisión de cada columna

#### 2.1 Novena Sección (NPCJP)

#### 2.2 NPCJP7

##### 2.2.1 5. ¿Por qué razón principal no pertenece a alguna organización?

1. Porque no le genera confianza
2. Porque no conoce quién la lidera
3. Porque no conoce organizaciones
4. Porque es costoso participar
5. Porque no le interesa o no le ve utilidad
6. Porque no lo/a han invitado
7. Por falta de tiempo
8. Otra razón
9. No sabe, no responde

Datos: 19891

```
[55]: data['NPCJP7'].count()
```

```
[55]: 19891
```

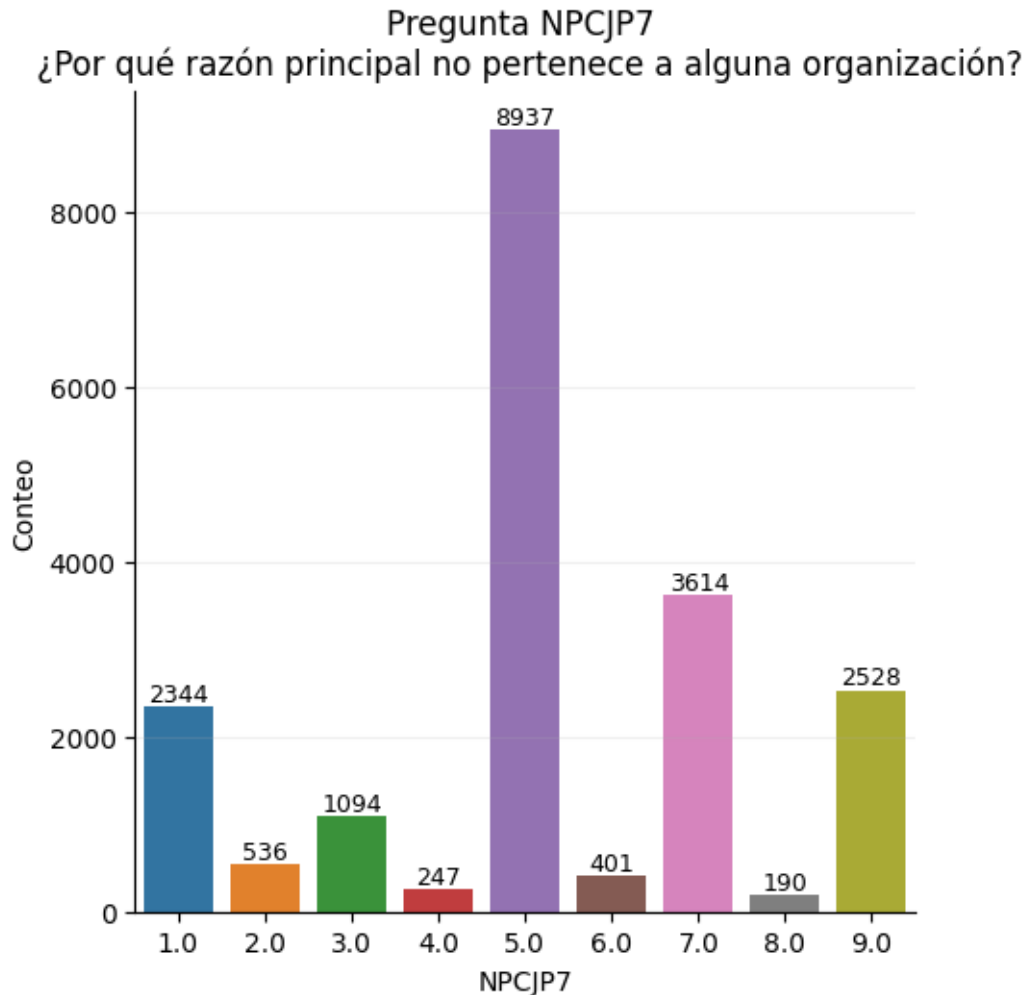
```
[56]: data['NPCJP7'].value_counts()
```

```
[56]: 5.0    8937
      7.0    3614
      9.0    2528
      1.0    2344
      3.0    1094
      2.0     536
      6.0     401
      4.0     247
      8.0     190
      Name: NPCJP7, dtype: int64
```

```
[57]: plt.figure(figsize=(10,5))
      g = sns.catplot(data = data, x = 'NPCJP7', kind='count')
      plt.title('Pregunta NPCJP7 \n ¿Por qué razón principal no pertenece a alguna_
      ↪organización?')
      plt.ylabel('Conteo')
      ax = g.facet_axis(0, 0)
      plt.grid(alpha = 0.2, axis = 'y')
      # iterate through the axes containers
      for c in ax.containers:
          labels = [f'{{(v.get_height()):.0f}}' for v in c]
          ax.bar_label(c, labels = labels, label_type = 'edge', fontsize=9)

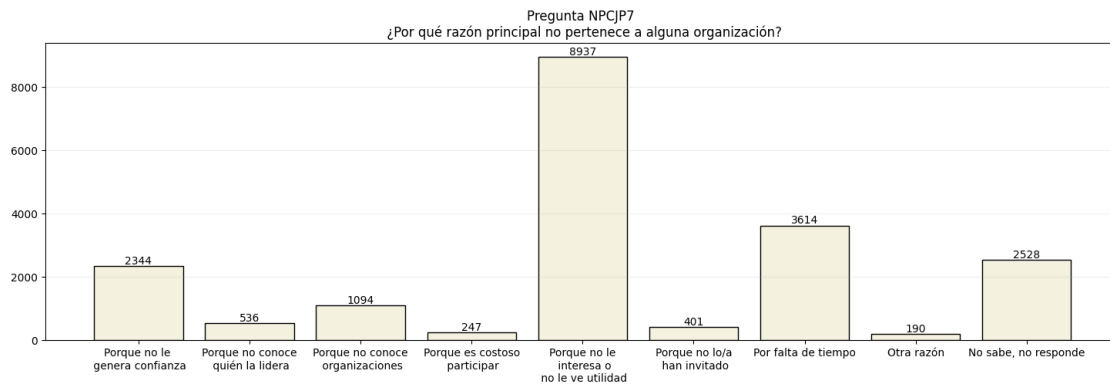
      plt.show()
```

<Figure size 1000x500 with 0 Axes>



```
[58]: plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCJP7'].value_counts().index.tolist(),data['NPCJP7'].
    ↳value_counts().tolist(), edgecolor = 'k', color = '#f4f1de')
plt.xticks([1,2,3,4,5,6,7,8,9], ['Porque no le \n genera confianza',
    'Porque no conoce\n quién la lidera',
    'Porque no conoce\n organizaciones',
    'Porque es costoso\n participar',
    'Porque no le\n interesa o\n no le ve utilidad',
    'Porque no lo/a\n han invitado',
    'Por falta de tiempo',
    'Otra razón',
    'No sabe, no responde'])
plt.title('Pregunta NPCJP7 \n ¿Por qué razón principal no pertenece a alguna_
    ↳organización?')
plt.bar_label(bars)
```

```
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



## 2.3 NPCJP9{A-J}

### 2.3.1 6. ¿A quién recurre cuando tiene problemas económicos?

- NPCJP9A : Familiares de otro hogar
- NPCJP9B : Vecinos y/o amigos
- NPCJP9C : Personas del hogar
- NPCJP9D : Compañero(a)s del trabajo
- NPCJP9E : Banco, cooperativa de ahorro
- NPCJP9F : Iglesia, congregación o grupo espiritual
- NPCJP9G : Compraventa, prestamista o gota a gota
- NPCJP9H : Otro
- NPCJP9I : No tiene a quien recurrir
- NPCJP9J : A nadie

```
[59]: eco = ['NPCJP9A',
            'NPCJP9B',
            'NPCJP9C',
            'NPCJP9D',
            'NPCJP9E',
            'NPCJP9F',
            'NPCJP9G',
            'NPCJP9H',
            'NPCJP9I',
            'NPCJP9J']
```

```
[60]: for i in eco:
        print(i, data[i].count())
```

```
NPCJP9A 11330
NPCJP9B 2169
```

```
NPCJP9C 14020
NPCJP9D 418
NPCJP9E 2253
NPCJP9F 143
NPCJP9G 57
NPCJP9H 59
NPCJP9I 178
NPCJP9J 1463
```

## 2.4 NPCJP8{A-H}

### 2.4.1 7. ¿Quién ayuda a ...cuando tiene problemas personales?

- NPCJP8A : Alguien del hogar
- NPCJP8B : Familiares de otro hogar
- NPCJP8C : Vecino/as o amigo/as
- NPCJP8D : Un/a profesional especializado/a
- NPCJP8E : Compañero/as de trabajo
- NPCJP8F : Iglesia, congregación o grupo espiritual
- NPCJP8G : No tiene quien le ayude
- NPCJP8H : Nadie, lo soluciona solo

```
[61]: pro2 = [i for i in data.columns if 'NPCJP8' in str(i)]
      pro2
```

```
[61]: ['NPCJP8A',
      'NPCJP8B',
      'NPCJP8C',
      'NPCJP8D',
      'NPCJP8E',
      'NPCJP8F',
      'NPCJP8G',
      'NPCJP8H']
```

```
[62]: for i in pro2:
      print(i,data[i].count())
```

```
NPCJP8A 16716
NPCJP8B 9481
NPCJP8C 2919
NPCJP8D 634
NPCJP8E 175
NPCJP8F 339
NPCJP8G 67
NPCJP8H 1662
```

## 2.5 NPCJP9A{A-k}

2.5.1 8. En una escala de 0 a 10, donde 0 es ““nada satisfecho”” y 10 es ““muy satisfecho””, ¿qué tan satisfecho está usted con:

- su vida?
- vivienda?
- su ingreso?
- su salud?
- su trabajo?
- su seguridad en los sitios que frecuenta?
- sus amigos?
- sus relaciones familiares?
- su educación?
- su posibilidad de tomar decisiones y tener control sobre su propia vida?
- su barrio o comunidad?

Datos: 20878

```
[63]: list_col = ['NPCJP9AA',
                 'NPCJP9AB',
                 'NPCJP9AC',
                 'NPCJP9AD',
                 'NPCJP9AE',
                 'NPCJP9AF',
                 'NPCJP9AG',
                 'NPCJP9AH',
                 'NPCJP9AI',
                 'NPCJP9AJ',
                 'NPCJP9AK']

list_names = ['su vida?',
              'su vivienda?',
              'su ingreso?',
              'su salud?',
              'su trabajo?',
              'su seguridad en los sitios que frecuenta?',
              'sus amigos?',
              'sus relaciones familiares?',
              'su educación?',
              'su posibilidad de tomar decisiones y tener control sobre su propia vida?',
              'su barrio o comunidad?']
```

```
[64]: for i in list_col:
        print(i, data[i].count())
```

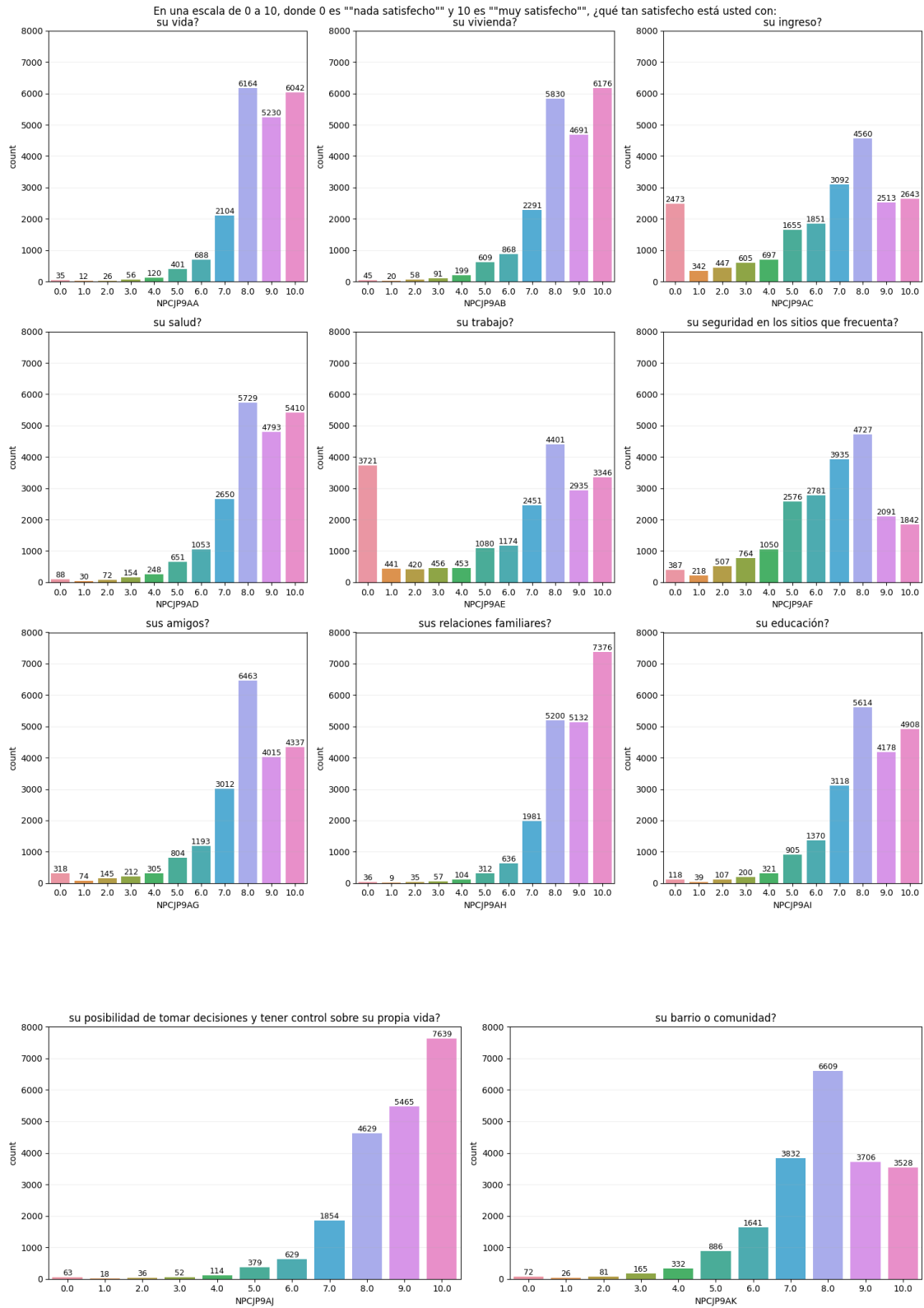
```
NPCJP9AA 20878
NPCJP9AB 20878
NPCJP9AC 20878
NPCJP9AD 20878
```

NPCJP9AE 20878  
NPCJP9AF 20878  
NPCJP9AG 20878  
NPCJP9AH 20878  
NPCJP9AI 20878  
NPCJP9AJ 20878  
NPCJP9AK 20878

```
[65]: fig, axes = plt.subplots(3,3, figsize = (15,15), squeeze=False)
axli = axes.flatten()
fig.subplots_adjust(top=0.9)
fig.suptitle('En una escala de 0 a 10, donde 0 es "nada satisfecho" y 10 es
↳ "muy satisfecho", ¿qué tan satisfecho está usted con:')
for ax,cols,names in zip(axli,list_col,list_names):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.set_ylim(0,8000)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title(f'{names}')
    ax.margins(y=0.1) # make room for the labels
    for bars in ax.containers:
        ax.bar_label(bars, fmt='%.0f', fontsize=9)
plt.tight_layout()

fig, axes = plt.subplots(1,2, figsize = (15,5))
for ax,cols,names in zip(axes,list_col[9:],list_names[9:]):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.set_ylim(0,8000)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title(f'{names}')
    ax.margins(y=0.1) # make room for the labels
    for bars in ax.containers:
        ax.bar_label(bars, fmt='%.0f', fontsize=9)
plt.tight_layout()

plt.show()
```





## 2.6 NPCJP10

2.6.1 9. En una escala de 0 a 10, donde 0 es “nada feliz” y 10 “muy feliz”, ¿El día de ayer qué tan feliz se sintió?

Datos: 19891

```
[66]: data['NPCJP10'].count()
```

```
[66]: 20878
```

```
[67]: data['NPCJP10'].value_counts()
```

```
[67]: 8.0      5781
      10.0     4874
      9.0     3850
      7.0     2984
      6.0     1481
      5.0     1055
      4.0       389
      3.0       173
      0.0       148
      2.0       102
      1.0        41
      Name: NPCJP10, dtype: int64
```

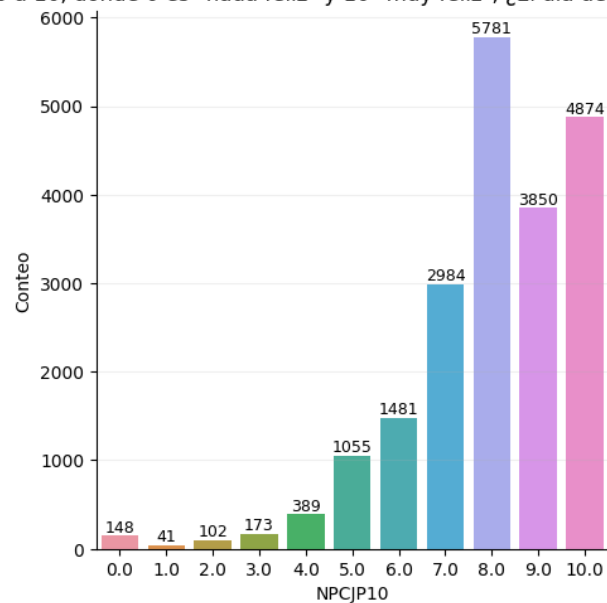
```
[68]: plt.figure(figsize=(10,5))

g = sns.catplot(data = data, x = 'NPCJP10', kind='count')
plt.title('Pregunta NPCJP10 \n En una escala de 0 a 10, donde 0 es "nada feliz"
↪ y 10 "muy feliz", ¿El día de ayer qué tan feliz se sintió?')
plt.ylabel('Conteo')
plt.grid(alpha = 0.2, axis = 'y')
for ax in g.axes.ravel():

    # iterate through the axes containers
    for c in ax.containers:
        labels = [f'{{(v.get_height()):.0f}}' for v in c]
        ax.bar_label(c, labels = labels, label_type = 'edge', fontsize=9)
plt.show()
```

<Figure size 1000x500 with 0 Axes>

Pregunta NPCJP10  
En una escala de 0 a 10, donde 0 es "nada feliz" y 10 "muy feliz", ¿El día de ayer qué tan feliz se sintió?



## 2.7 NPCJP11

### 2.7.1 10. En una escala de 0 a 10, donde 0 es “nada preocupado” y 10 “muy preocupado”, ¿El día de ayer qué tan preocupado se sintió?

Datos: 20878

```
[69]: data['NPCJP11'].count()
```

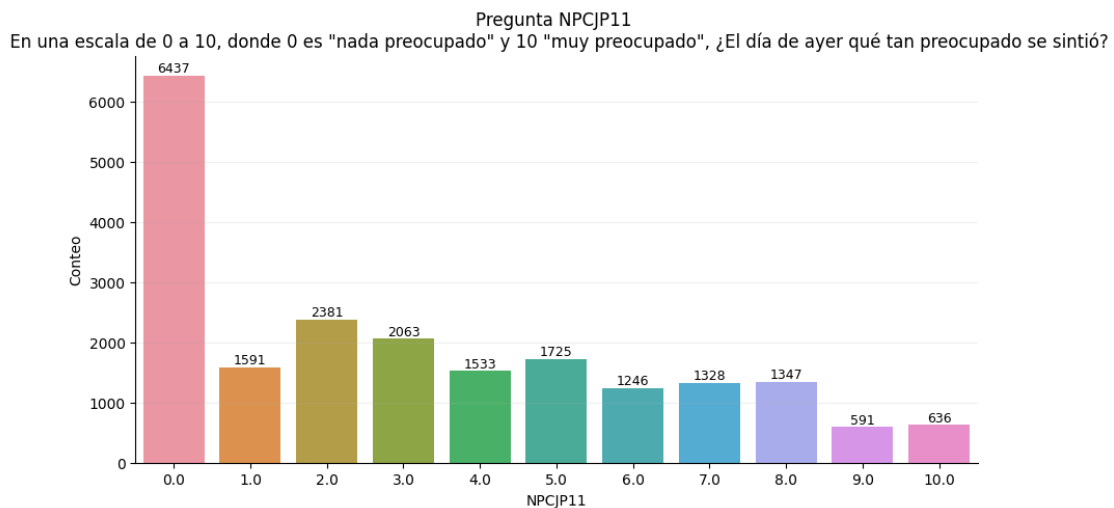
```
[69]: 20878
```

```
[70]: data['NPCJP11'].value_counts()
```

```
[70]: 0.0      6437
      2.0      2381
      3.0      2063
      5.0      1725
      1.0      1591
      4.0      1533
      8.0      1347
      7.0      1328
      6.0      1246
      10.0      636
      9.0       591
      Name: NPCJP11, dtype: int64
```

```
[71]: g = sns.catplot(data = data, x = 'NPCJP11', kind='count')
g.fig.set_size_inches(10, 5)
g.fig.subplots_adjust(top=0.9)
plt.title('Pregunta NPCJP11 \n En una escala de 0 a 10, donde 0 es "nada_
↳ preocupado" y 10 "muy preocupado", ¿El día de ayer qué tan preocupado se_
↳ sintió?')
plt.ylabel('Conteo')
plt.grid(alpha = 0.2, axis = 'y')
for ax in g.axes.ravel():

    # iterate through the axes containers
    for c in ax.containers:
        labels = [f'{(v.get_height()):.0f}' for v in c]
        ax.bar_label(c, labels = labels, label_type = 'edge', fontsize=9)
plt.show()
```



## 2.8 NPCJP12

2.8.1 11. En una escala de 0 a 10, donde 0 es “nada enojado” y 10 “muy enojado”, ¿El día de ayer qué tan enojado se sintió?

Datos: 20878

```
[72]: data['NPCJP12'].count()
```

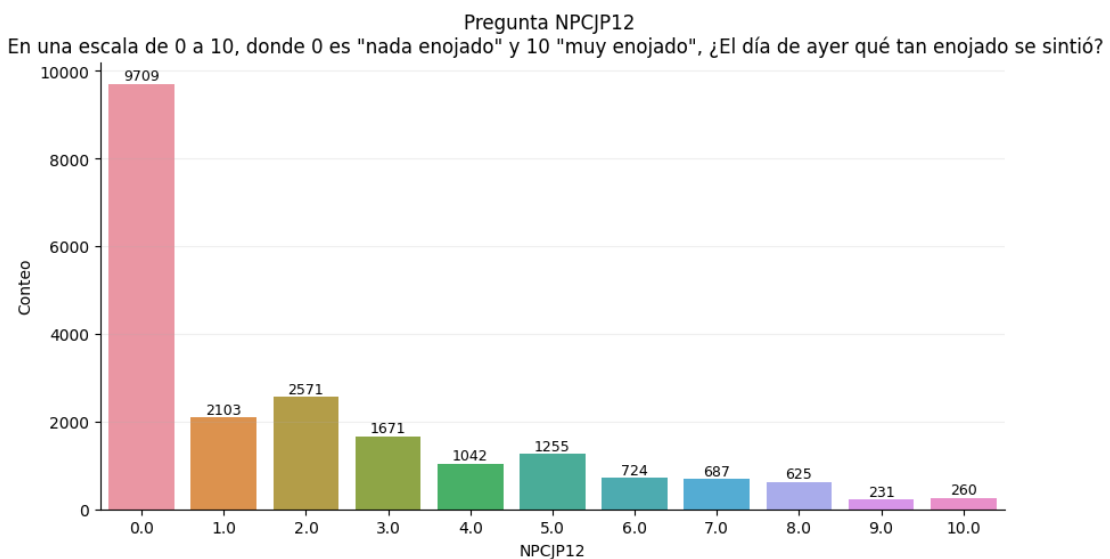
```
[72]: 20878
```

```
[73]: data['NPCJP12'].value_counts()
```

```
[73]: 0.0    9709
      2.0    2571
      1.0    2103
      3.0    1671
      5.0    1255
      4.0    1042
      6.0     724
      7.0     687
      8.0     625
     10.0     260
      9.0     231
      Name: NPCJP12, dtype: int64
```

```
[74]: g = sns.catplot(data = data, x = 'NPCJP12', kind='count')
      g.fig.set_size_inches(10, 5)
      g.fig.subplots_adjust(top=0.9)
      plt.title('Pregunta NPCJP12 \n En una escala de 0 a 10, donde 0 es "nada_
      ↪enojado" y 10 "muy enojado", ¿El día de ayer qué tan enojado se sintió?')
      plt.ylabel('Conteo')
      plt.grid(alpha = 0.2, axis = 'y')
      for ax in g.axes.ravel():

          # iterate through the axes containers
          for c in ax.containers:
              labels = [f'{(v.get_height()):.0f}' for v in c]
              ax.bar_label(c, labels = labels, label_type = 'edge', fontsize=9)
      plt.show()
```



## 2.9 NPCJP13

**2.9.1 12.** Imagine una escalera con escalones numerados de 0 a 10, donde cero es el escalón más bajo y 10, el escalón más alto. El más alto representa la mejor vida posible para usted y el más bajo la peor vida posible para usted.

¿En cuál escalón diría usted que se encuentra parado/a en este momento?

**Datos: 20878**

```
[75]: data['NPCJP13'].count()
```

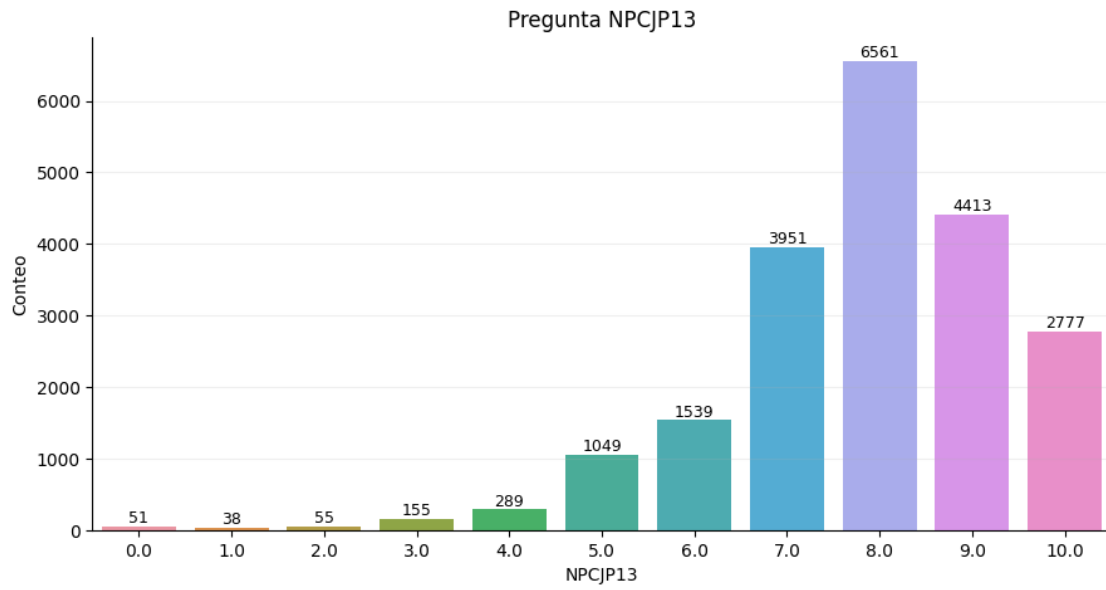
```
[75]: 20878
```

```
[76]: data['NPCJP13'].value_counts()
```

```
[76]: 8.0    6561
      9.0    4413
      7.0    3951
      10.0   2777
      6.0    1539
      5.0    1049
      4.0     289
      3.0     155
      2.0      55
      0.0      51
      1.0      38
      Name: NPCJP13, dtype: int64
```

```
[77]: g = sns.catplot(data = data, x = 'NPCJP13', kind='count')
      g.fig.set_size_inches(10, 5)
      g.fig.subplots_adjust(top=0.9)
      plt.title('Pregunta NPCJP13')
      plt.ylabel('Conteo')
      plt.grid(alpha = 0.2, axis = 'y')
      for ax in g.axes.ravel():

          # iterate through the axes containers
          for c in ax.containers:
              labels = [f'{{(v.get_height()):.0f}}' for v in c]
              ax.bar_label(c, labels = labels, label_type = 'edge', fontsize=9)
      plt.show()
```



```
[78]: data.to_excel('Encuesta_Multiproposito_Suba.xlsx', index = False)
```