

```
In [1]: #Craga librerias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: #Cargar base de datos
df = sns.load_dataset('titanic')

# Mostrar las primeras filas
df.head()
```

Out[2]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class
0	0	3	male	22.0	1	0	7.2500	S	Third
1	1	1	female	38.0	1	0	71.2833	C	First
2	1	3	female	26.0	0	0	7.9250	S	Third
3	1	1	female	35.0	1	0	53.1000	S	First
4	0	3	male	35.0	0	0	8.0500	S	Third

Variables categóricas

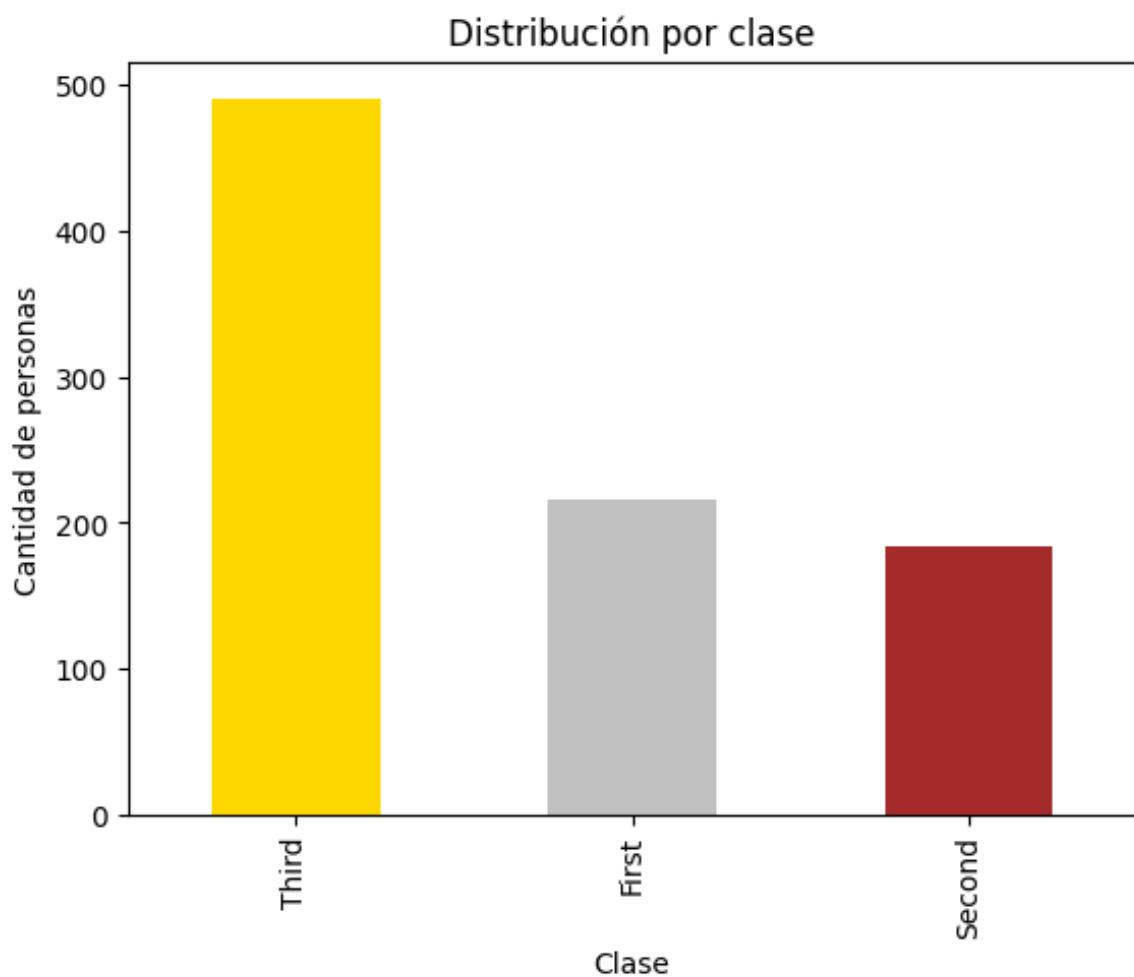
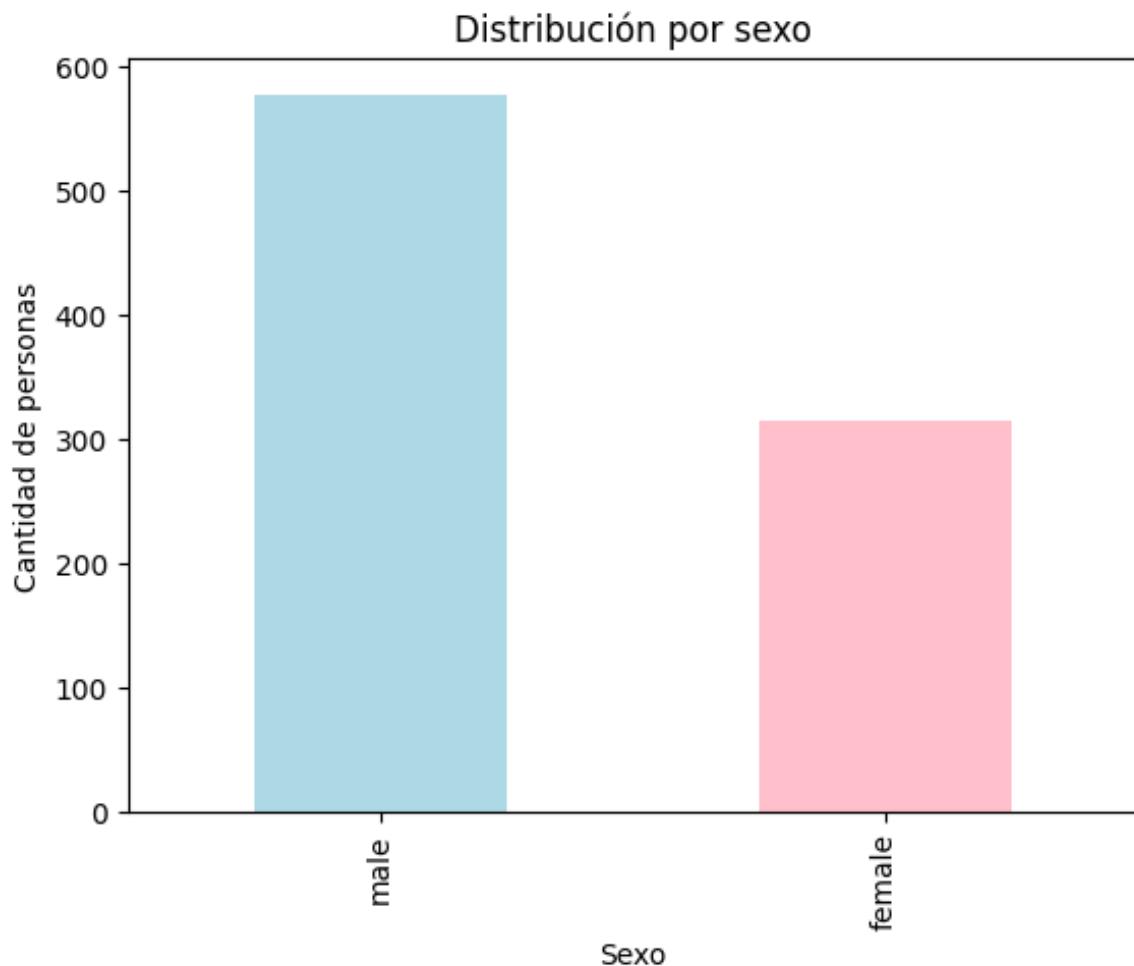
'survived', 'sex', 'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town', 'alive', 'alone'

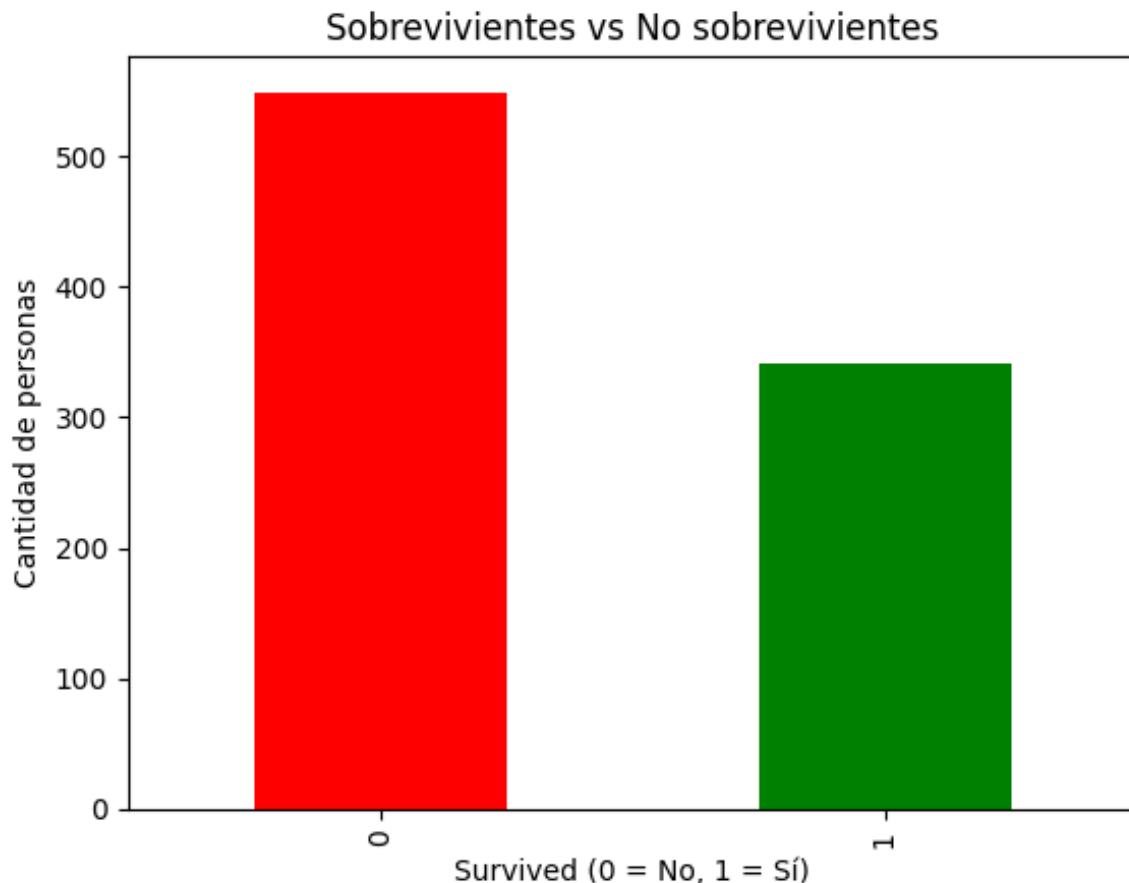
Graficas de Barras

```
In [3]: # Conteo de personas por sexo
df['sex'].value_counts().plot(kind='bar', color=['lightblue', 'pink'])
plt.title('Distribución por sexo')
plt.xlabel('Sexo')
plt.ylabel('Cantidad de personas')
plt.show()

# Conteo de personas por clase
df['class'].value_counts().plot(kind='bar', color=['gold', 'silver'])
plt.title('Distribución por clase')
plt.xlabel('Clase')
plt.ylabel('Cantidad de personas')
plt.show()

# Conteo de sobrevivientes
df['survived'].value_counts().plot(kind='bar', color=['red', 'green'])
plt.title('Sobrevivientes vs No sobrevivientes')
plt.xlabel('Survived (0 = No, 1 = Sí)')
plt.ylabel('Cantidad de personas')
plt.show()
```





Variables numéricas

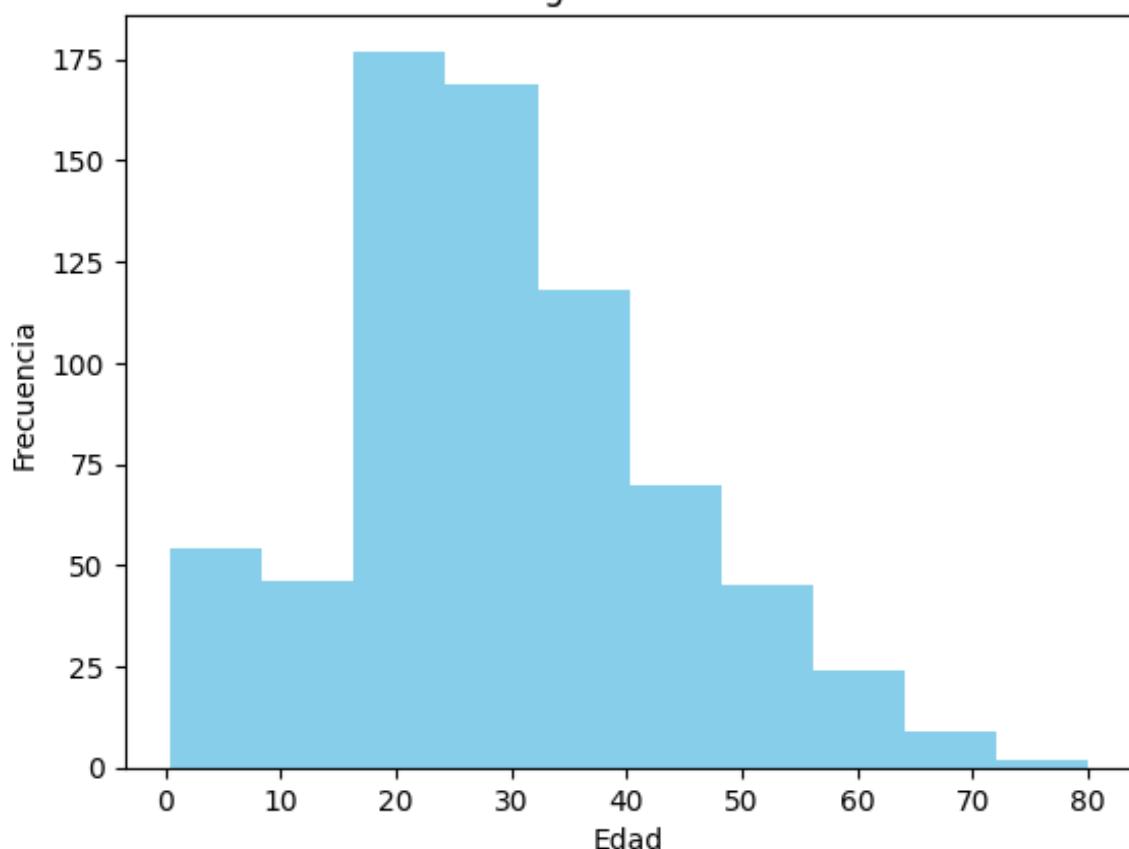
'pclass', 'age', 'sibsp', 'parch', 'fare'

Historogramas

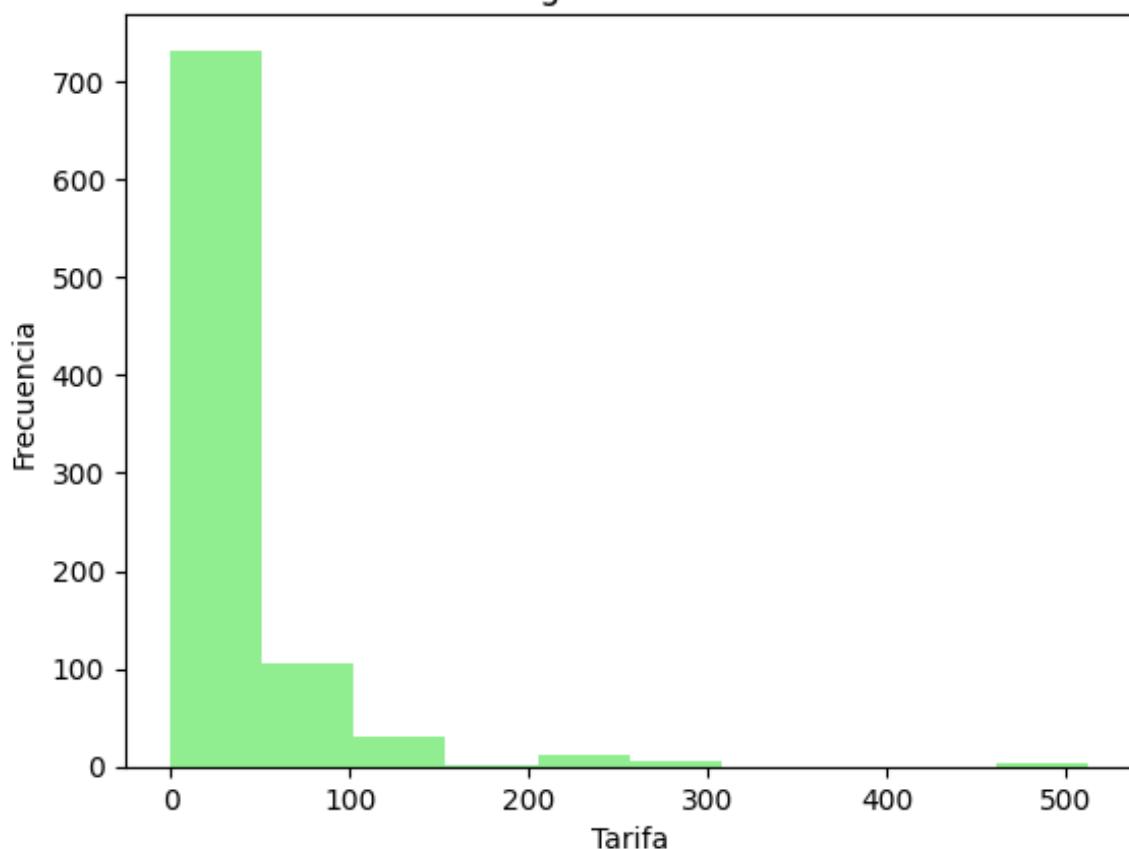
```
In [4]: # Edad
plt.hist(df['age'].dropna(), bins=10, color='skyblue')
plt.title('Histograma de Edad')
plt.xlabel('Edad')
plt.ylabel('Frecuencia')
plt.show()

# Tarifa (fare)
plt.hist(df['fare'], bins=10, color='lightgreen')
plt.title('Histograma de Tarifa')
plt.xlabel('Tarifa')
plt.ylabel('Frecuencia')
plt.show()
```

Histograma de Edad



Histograma de Tarifa

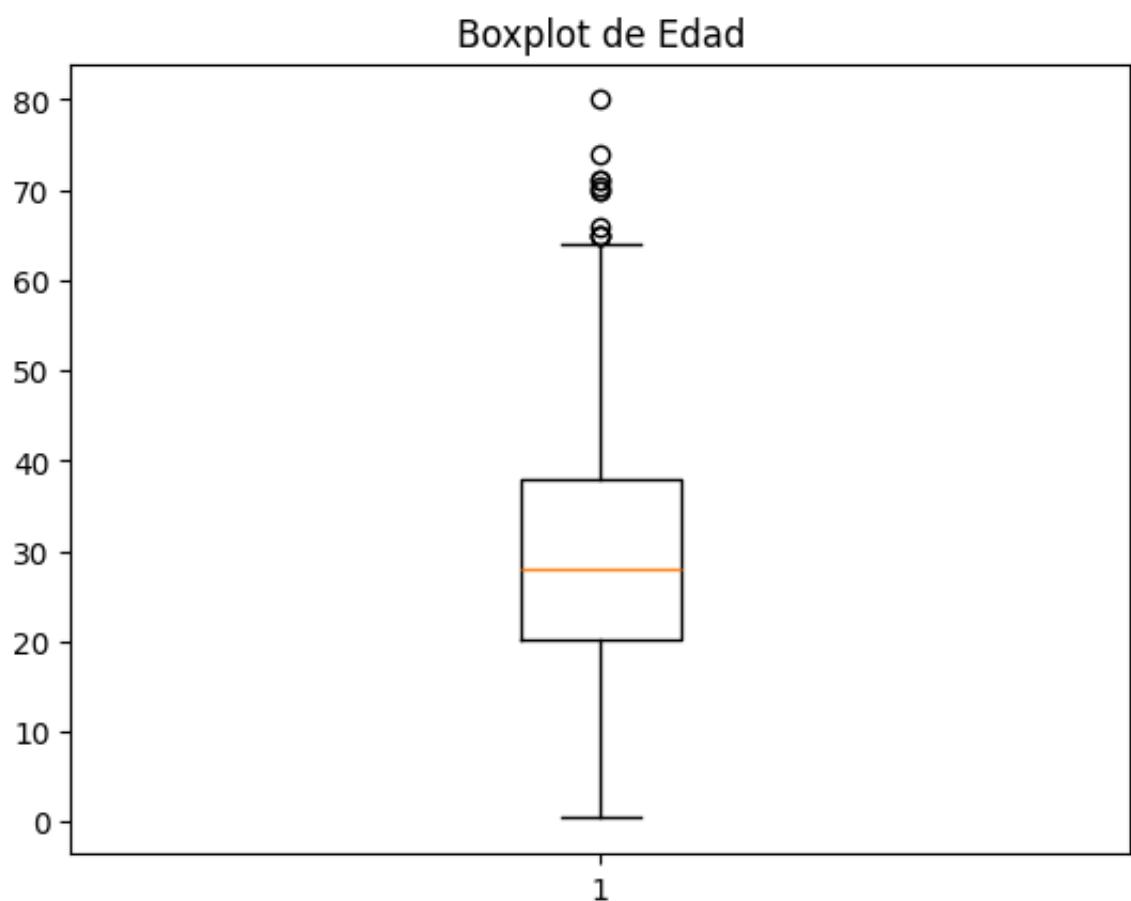


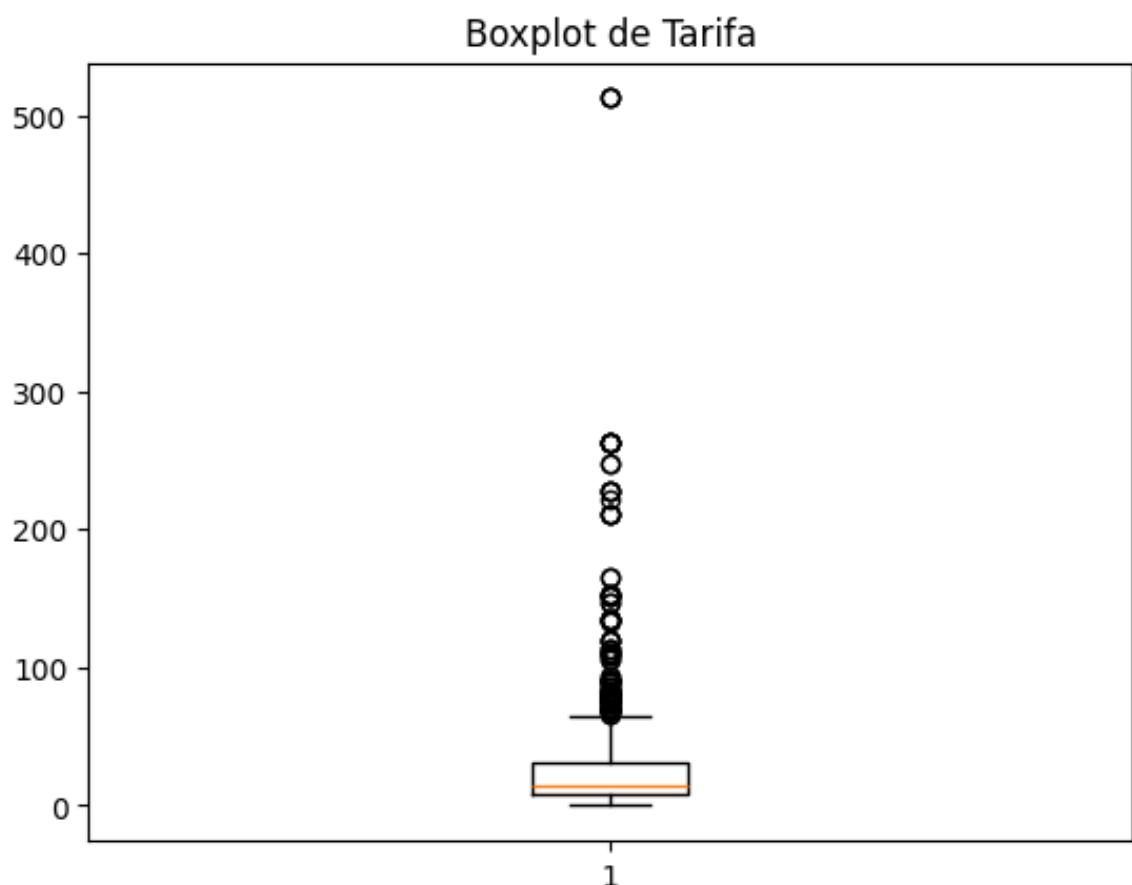
Boxplots

```
In [5]: plt.boxplot(df['age'].dropna())
```

```
plt.title('Boxplot de Edad')
plt.show()

plt.boxplot(df['fare'])
plt.title('Boxplot de Tarifa')
plt.show()
```





Mapa de Correlación

In [6]: `# Calcular correlación (solo columnas numéricas)`
`corr = df.corr(numeric_only=True)`

```
# Gráfica de mapa de calor
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Mapa de calor de correlaciones')
plt.show()
```

	survived	pclass	age	sibsp	parch	fa
re \						
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.2573
07						
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.5495
00						
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.0960
67						
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.1596
51						
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.2162
25						
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.0000
00						
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.1820
24						
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.2718
32						
		adult_male	alone			
survived	-0.557080	-0.203367				
pclass	0.094035	0.135207				
age	0.280328	0.198270				
sibsp	-0.253586	-0.584471				
parch	-0.349943	-0.583398				
fare	-0.182024	-0.271832				
adult_male	1.000000	0.404744				
alone	0.404744	1.000000				

