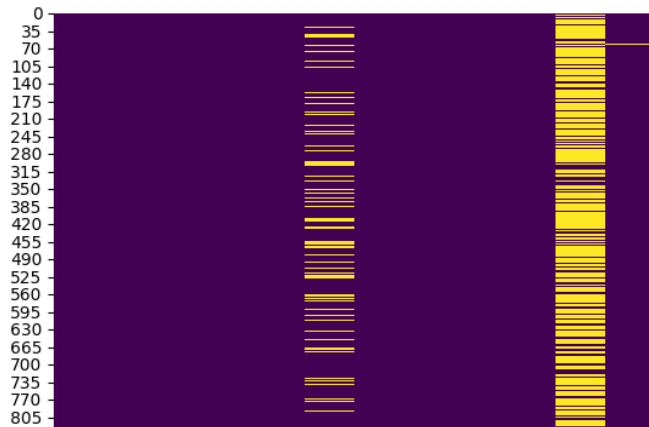


```
# LECTURA DE UN ARCHIVO CSV
import pandas as pd
from math import isnan
import seaborn as sns

nombreArch = input("Nombre de archivo CSV: ")
datosOrigen = pd.read_csv(nombreArch+'.csv', encoding='UTF8')
datosOrigen.head()
datosOrigen.info()
datosOrigen.describe()
sns.heatmap(datosOrigen.isnull(), cbar=False, cmap='viridis')
```

```
Nombre de archivo CSV: titanic
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age         714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Cabin        204 non-null    object
11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
<Axes: >
```



```
datosOrigen.head()
datosOrigen.info()
datosOrigen.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column             Non-Null Count  Dtype
---  -
0   PassengerId         891 non-null    int64
1   Survived            891 non-null    int64
2   Pclass              891 non-null    int64
3   Name                891 non-null    object
4   Sex                 891 non-null    object
5   Age                 891 non-null    float64
6   SibSp               891 non-null    int64
7   Parch               891 non-null    int64
8   Ticket              891 non-null    object
9   Fare                891 non-null    float64
10  Cabin               891 non-null    object
11  Embarked            891 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	116.000000	0.383838	2.308612	23.790203	0.523008	0.381504	32.204205

```
# Obtener los nombres de las columnas
nombres_columnas = datosOrigen.columns

# Imprimir los nombres de las columnas
print("Nombres de las columnas:")
for columna in nombres_columnas:
    print(columna)

# Nombres de las columnas:
PassengerId
Survived
Pclass
Name
Sex
Age
SibSp
Parch
Ticket
Fare
Cabin
Embarked

# Calcular el promedio de la columna
promedio = datosOrigen['Age'].mean()

round_promedio = round(promedio)

# Imprimir el resultado
print("El promedio de la columna 'Age' es:", round_promedio)

El promedio de la columna 'Age' es: 24

#rellenamos los valores nulos de la edad con la edad promedio
datosOrigen['Age'].fillna(promedio, inplace=True)
#relleno los valores nulos de la cabina con 0
datosOrigen['Cabin'].fillna(0, inplace=True)

datosOrigen.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0
1	2	1	1	Cumings, Mrs. John	female	35.0	1	0

```
#hombres con hijos
# Filtrar las filas donde 'sex' sea igual a 'male' y 'parch' sea igual o mayor a 1
condicion = (datosOrigen['Sex'] == 'male') & (datosOrigen['Survived'] == 1) & (datosOrigen['Parch'] >= 1)
```

```
#Contar las filas que cumplen con la condición
cantidad_resultados = len(datosOrigen[condicion])

# Imprimir la cantidad de resultados
print("Cantidad de resultados con 'sex' igual a 'male' y 'parch' igual o mayor a 1:", cantidad_resultados)

    Cantidad de resultados con 'sex' igual a 'male' y 'parch' igual o mayor a 1: 29

#Personas que sobrevivieron por clase
#Clase 1
Clase1 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 1)
Cantidad_Clase1= len(datosOrigen[Clase1])
print("Clase 1 =", Cantidad_Clase1)

#Personas que sobrevivieron por clase
#Clase 2
Clase2 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 2)
Cantidad_Clase2= len(datosOrigen[Clase2])
print("Clase 2 =", Cantidad_Clase2)

#Personas que sobrevivieron por clase
#Clase 3
Clase3 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 3)
Cantidad_Clase3= len(datosOrigen[Clase3])
print("Clase 3 =", Cantidad_Clase3)

    Clase 1 = 136
    Clase 2 = 87
    Clase 3 = 119

#Personas que sobrevivieron por clase
#Clase 1
Clase1 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 1) & (datosOrigen['Sex'] == 'male')
Cantidad_Clase1= len(datosOrigen[Clase1])
print("Clase 1 =", Cantidad_Clase1)

#Personas que sobrevivieron por clase
#Clase 2
Clase2 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 2) & (datosOrigen['Sex'] == 'male')
Cantidad_Clase2= len(datosOrigen[Clase2])
print("Clase 2 =", Cantidad_Clase2)

#Personas que sobrevivieron por clase
#Clase 3
Clase3 = (datosOrigen['Survived'] == 1) & (datosOrigen['Pclass'] == 3) & (datosOrigen['Sex'] == 'male')
Cantidad_Clase3= len(datosOrigen[Clase3])
print("Clase 3 =", Cantidad_Clase3)

    Clase 1 = 45
    Clase 2 = 17
    Clase 3 = 47
```

