In [63]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

pd.set_option('display.max_rows', None)  # Muestra todas las filas
```

In [91]:
```
datos1=pd.read_excel('datos1.xlsx')
datos2=pd.read_excel('datos2.xlsx')
datos3=pd.read_excel('datos3.xlsx')
datos4=pd.read_excel('datos4.xlsx')
datos5=pd.read_excel('datos5.xlsx')

datos = pd.concat([datos1,
                   datos2,
                   datos3,
                   datos4,
                   datos5]).drop_duplicates('Matricula', keep = 'last').reset_index(drop=True)
    ['Versión','Distintivo Ambiental','Historial de revisiones','Estándar de calidad'], axis=1
```

In [92]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2411 entries, 0 to 2410
Data columns (total 28 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Marca                             2411 non-null   object
 1   Modelo                            2411 non-null   object
 2   Precio                            2411 non-null   int64
 3   Primera matriculación             2411 non-null   object
 4   Kilometraje                       2411 non-null   object
 5   Carburante                        2411 non-null   object
 6   Transmisión                       2411 non-null   object
 7   Potencia                          2411 non-null   object
 8   Tracción                          2389 non-null   object
 9   Tipo de vehículo                  2411 non-null   object
 10  Puertas                           2409 non-null   float64
 11  Número de asientos                2411 non-null   int64
 12  Color                             2410 non-null   object
 13  Tapicería                         2408 non-null   object
 14  Tipo de ruedas                    2241 non-null   object
 15  Motor original                    2411 non-null   object
 16  Cilindrada                        2411 non-null   object
 17  Consumo                           2151 non-null   object
 18  Clase de eficiencia CO2           2411 non-null   object
 19  Emisiones de CO2                  2241 non-null   object
 20  País de origen                    2411 non-null   object
 21  Número de llaves                  2411 non-null   int64
 22  Coche accidentado y reparado      2241 non-null   object
 23  La última revisión se realizó el  2148 non-null   object
 24  Tipo de IVA                       2411 non-null   object
 25  ITV válida hasta                  2241 non-null   object
 26  Matricula                         2410 non-null   object
 27  Número de inventario              2411 non-null   object
dtypes: float64(1), int64(3), object(24)
memory usage: 527.5+ KB
```

In [93]:
```
datos.isnull().sum()
```

Out[93]:
```
Marca                                  0
Modelo                                 0
Precio                                 0
Primera matriculación                  0
Kilometraje                            0
Carburante                             0
Transmisión                            0
Potencia                               0
Tracción                              22
Tipo de vehículo                       0
Puertas                                2
Número de asientos                     0
Color                                  1
Tapicería                              3
Tipo de ruedas                       170
Motor original                         0
Cilindrada                             0
Consumo                              260
Clase de eficiencia CO2                0
Emisiones de CO2                     170
País de origen                         0
Número de llaves                       0
Coche accidentado y reparado         170
La última revisión se realizó el     263
Tipo de IVA                            0
ITV válida hasta                     170
Matricula                              1
Número de inventario                   0
dtype: int64
```

In [94]:
```
datos = datos.dropna()
datos.reset_index(drop=True, inplace=True)
```

In [95]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1924 entries, 0 to 1923
Data columns (total 28 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Marca                             1924 non-null   object 
 1   Modelo                            1924 non-null   object 
 2   Precio                            1924 non-null   int64  
 3   Primera matriculación             1924 non-null   object 
 4   Kilometraje                       1924 non-null   object 
 5   Carburante                        1924 non-null   object 
 6   Transmisión                       1924 non-null   object 
 7   Potencia                          1924 non-null   object 
 8   Tracción                          1924 non-null   object 
 9   Tipo de vehículo                  1924 non-null   object 
 10  Puertas                           1924 non-null   float64
 11  Número de asientos                1924 non-null   int64  
 12  Color                             1924 non-null   object 
 13  Tapicería                         1924 non-null   object 
 14  Tipo de ruedas                    1924 non-null   object 
 15  Motor original                    1924 non-null   object 
 16  Cilindrada                        1924 non-null   object 
 17  Consumo                           1924 non-null   object 
 18  Clase de eficiencia CO2           1924 non-null   object 
 19  Emisiones de CO2                  1924 non-null   object 
 20  País de origen                    1924 non-null   object 
 21  Número de llaves                  1924 non-null   int64  
 22  Coche accidentado y reparado      1924 non-null   object 
 23  La última revisión se realizó el  1924 non-null   object 
 24  Tipo de IVA                       1924 non-null   object 
 25  ITV válida hasta                  1924 non-null   object 
 26  Matricula                         1924 non-null   object 
 27  Número de inventario              1924 non-null   object 
dtypes: float64(1), int64(3), object(24)
memory usage: 421.0+ KB
```

In [96]:
```
#primer paso: modificar variables para poder tratarlas
#las fechas se van a convertir en fechas de pandas y luego en días hasta hoy, para que sean núm
datos['Primera matriculación'] = pd.to_datetime(datos['Primera matriculación'], format = '%d.%
datos['Días desde matriculación'] = (pd.Timestamp.now() - datos['Primera matriculación']).dt.d
datos['La última revisión se realizó el'] = pd.to_datetime(datos['La última revisión se realiz
datos['Días desde revisión'] = (pd.Timestamp.now() - datos['La última revisión se realizó el']
datos['ITV válida hasta'] = pd.to_datetime(datos['ITV válida hasta'], format = '%d.%m.%Y')
datos['Días hasta ITV'] = (datos['ITV válida hasta'] - pd.Timestamp.now()).dt.days
```

In [97]:
```
#las variables numéricas
datos['Kilometraje'] = datos['Kilometraje'].str.replace(' km', '').str.replace('.','').astype(
datos['Potencia CV'] = datos['Potencia'].str.extract(r'(\d+) CV \/ (\d+) kW').iloc[:,0].astype
datos['Cilindrada']  = datos['Cilindrada'].str.replace(' ccm', '').astype(float)
datos['Emisiones de CO2'] = datos['Emisiones de CO2'].str.replace(' g/km', '').astype(float)
datos['Puertas'] = datos['Puertas'].astype(int)
datos['Precio'] = datos['Precio'].astype(float)

#El consumo hay que partirlo en 3
partes = datos['Consumo'].str.split(')', expand=True)
datos['Consumo ciudad'] = partes.iloc[:,0].str.extract(r'(\d+(?:\.\d+)?) l\/100 km \(En la ciu
datos['Consumo combinado'] = partes.iloc[:,1].str.extract(r'(\d+(?:\.\d+)?) l\/100 km \(Combin

#arreglo de combinados que aparecen en la primera columna
incluir = partes.iloc[:,0].str.extract(r'(\d+(?:\.\d+)?) l\/100 km \(Combinado').astype(float)
datos['Consumo combinado'] = datos['Consumo combinado'].combine_first(incluir)
datos['Consumo fuera'] = partes.iloc[:,2].str.extract(r'(\d+(?:\.\d+)?) l\/100 km \(Fuera de l
```

In [98]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1924 entries, 0 to 1923
Data columns (total 35 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Marca                               1924 non-null   object
 1   Modelo                              1924 non-null   object
 2   Precio                              1924 non-null   float64
 3   Primera matriculación               1924 non-null   datetime64[ns]
 4   Kilometraje                         1924 non-null   float64
 5   Carburante                          1924 non-null   object
 6   Transmisión                         1924 non-null   object
 7   Potencia                            1924 non-null   object
 8   Tracción                            1924 non-null   object
 9   Tipo de vehículo                    1924 non-null   object
 10  Puertas                             1924 non-null   int32
 11  Número de asientos                  1924 non-null   int64
 12  Color                               1924 non-null   object
 13  Tapicería                           1924 non-null   object
 14  Tipo de ruedas                      1924 non-null   object
 15  Motor original                      1924 non-null   object
 16  Cilindrada                          1924 non-null   float64
 17  Consumo                             1924 non-null   object
 18  Clase de eficiencia CO2             1924 non-null   object
 19  Emisiones de CO2                    1924 non-null   float64
 20  País de origen                      1924 non-null   object
 21  Número de llaves                    1924 non-null   int64
 22  Coche accidentado y reparado        1924 non-null   object
 23  La última revisión se realizó el    1924 non-null   datetime64[ns]
 24  Tipo de IVA                         1924 non-null   object
 25  ITV válida hasta                    1924 non-null   datetime64[ns]
 26  Matricula                           1924 non-null   object
 27  Número de inventario                1924 non-null   object
 28  Días desde matriculación            1924 non-null   int64
 29  Días desde revisión                 1924 non-null   int64
 30  Días hasta ITV                      1924 non-null   int64
 31  Potencia CV                         1924 non-null   float64
 32  Consumo ciudad                      1900 non-null   float64
 33  Consumo combinado                   1924 non-null   float64
 34  Consumo fuera                       1900 non-null   float64
dtypes: datetime64[ns](3), float64(8), int32(1), int64(5), object(18)
memory usage: 518.7+ KB
```

In [99]:
```
datos.isnull().sum()
```

Out[99]:
```
Marca                                  0
Modelo                                 0
Precio                                 0
Primera matriculación                  0
Kilometraje                            0
Carburante                             0
Transmisión                            0
Potencia                               0
Tracción                               0
Tipo de vehículo                       0
Puertas                                0
Número de asientos                     0
Color                                  0
Tapicería                              0
Tipo de ruedas                         0
Motor original                         0
Cilindrada                             0
Consumo                                0
Clase de eficiencia CO2                0
Emisiones de CO2                       0
País de origen                         0
Número de llaves                       0
Coche accidentado y reparado           0
La última revisión se realizó el       0
Tipo de IVA                            0
ITV válida hasta                       0
Matricula                              0
Número de inventario                   0
Días desde matriculación               0
Días desde revisión                    0
Días hasta ITV                         0
Potencia CV                            0
Consumo ciudad                        24
Consumo combinado                      0
Consumo fuera                         24
dtype: int64
```

Eliminamos los campos de Consumo y nos quedamos solo con el combinado

In [100]:
```
datos = datos.drop(['Consumo ciudad','Consumo fuera','Primera matriculación',
                    'La última revisión se realizó el','ITV válida hasta','Número de inventari
```

In [101]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1924 entries, 0 to 1923
Data columns (total 29 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Marca                         1924 non-null   object
 1   Modelo                        1924 non-null   object
 2   Precio                        1924 non-null   float64
 3   Kilometraje                   1924 non-null   float64
 4   Carburante                    1924 non-null   object
 5   Transmisión                   1924 non-null   object
 6   Potencia                      1924 non-null   object
 7   Tracción                      1924 non-null   object
 8   Tipo de vehículo              1924 non-null   object
 9   Puertas                       1924 non-null   int32
 10  Número de asientos            1924 non-null   int64
 11  Color                         1924 non-null   object
 12  Tapicería                     1924 non-null   object
 13  Tipo de ruedas                1924 non-null   object
 14  Motor original                1924 non-null   object
 15  Cilindrada                    1924 non-null   float64
 16  Consumo                       1924 non-null   object
 17  Clase de eficiencia CO2       1924 non-null   object
 18  Emisiones de CO2              1924 non-null   float64
 19  País de origen                1924 non-null   object
 20  Número de llaves              1924 non-null   int64
 21  Coche accidentado y reparado  1924 non-null   object
 22  Tipo de IVA                   1924 non-null   object
 23  Matricula                     1924 non-null   object
 24  Días desde matriculación      1924 non-null   int64
 25  Días desde revisión           1924 non-null   int64
 26  Días hasta ITV                1924 non-null   int64
 27  Potencia CV                   1924 non-null   float64
 28  Consumo combinado             1924 non-null   float64
dtypes: float64(6), int32(1), int64(5), object(17)
memory usage: 428.5+ KB
```

In [102]:
```
datos.select_dtypes(exclude=['datetime64']).describe().T
```

Out[102]:

	count	mean	std	min	25%	50%	75%	max
Precio	1924.0	15630.185031	5969.871875	5799.0	11199.00	14399.0	18424.000	43899.0
Kilometraje	1924.0	73950.833160	38826.959899	202.0	42852.25	69649.0	102723.000	159645.0
Puertas	1924.0	4.810811	0.582790	2.0	5.00	5.0	5.000	6.0
Número de asientos	1924.0	5.008836	0.575478	2.0	5.00	5.0	5.000	9.0
Cilindrada	1924.0	1487.669958	372.391397	875.0	1199.00	1498.0	1598.000	2998.0
Emisiones de CO2	1924.0	118.062370	21.697904	1.0	106.00	115.0	129.000	226.0
Número de llaves	1924.0	1.823805	0.385157	1.0	2.00	2.0	2.000	4.0
Días desde matriculación	1924.0	2275.527547	897.584942	234.0	1634.50	2179.0	2897.000	4509.0
Días desde revisión	1924.0	124.205821	298.187100	1.0	30.00	58.0	100.000	3454.0
Días hasta ITV	1924.0	337.408004	239.957758	-673.0	150.00	322.5	526.000	1226.0
Potencia CV	1924.0	124.905925	39.019149	60.0	100.00	120.0	140.000	340.0
Consumo combinado	1924.0	4.825104	0.979700	0.6	4.10	4.7	5.325	9.2

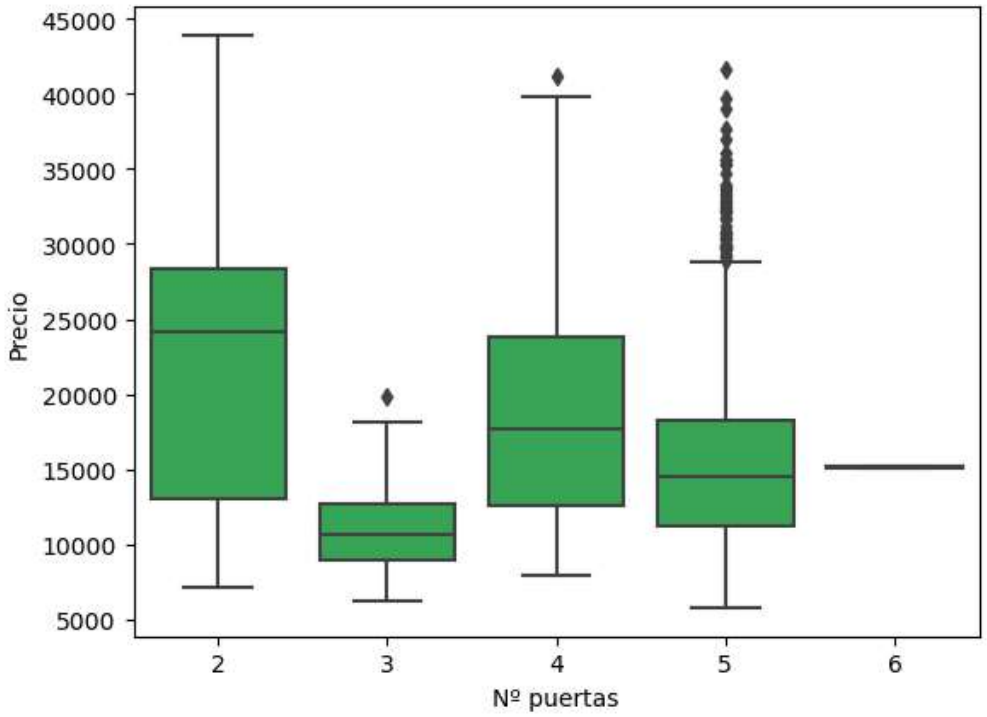
In [103]:
```
datos.groupby('Puertas').agg(frequency=('Puertas', "count"))
```

Out[103]:

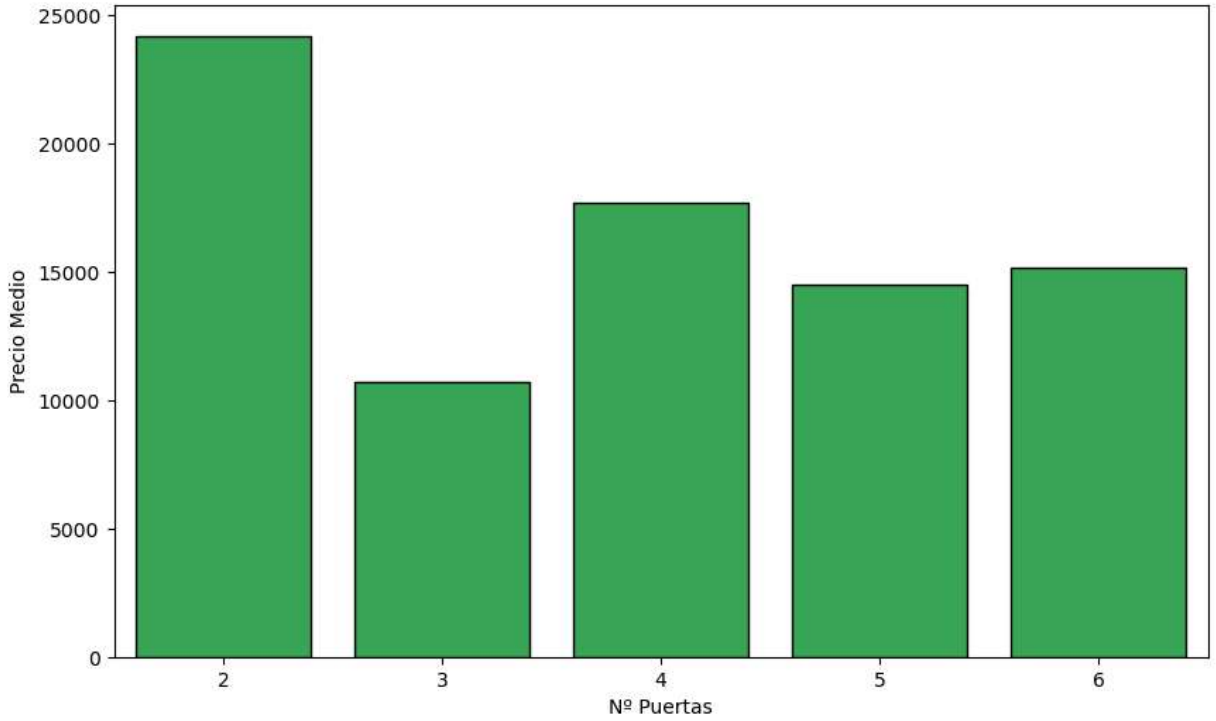
Puertas	frequency
2	29
3	90
4	99
5	1704
6	2

In [104]:
```
ax = sns.boxplot(x = 'Puertas', y = 'Precio', data = datos, color = '#28b84f')
ax.set( xlabel = 'Nº puertas')
```

Out[104]:
```
[Text(0.5, 0, 'Nº puertas')]
```



In [122]:
```
media_precios = datos.groupby('Puertas')['Precio'].median().reset_index()

# Crear el gráfico de barras
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Puertas', y='Precio', data=media_precios, color='#28b84f', edgecolor='blac

ax.set_xlabel('Nº Puertas')
ax.set_ylabel('Precio Medio')

# Mostrar el gráfico
plt.show()
```



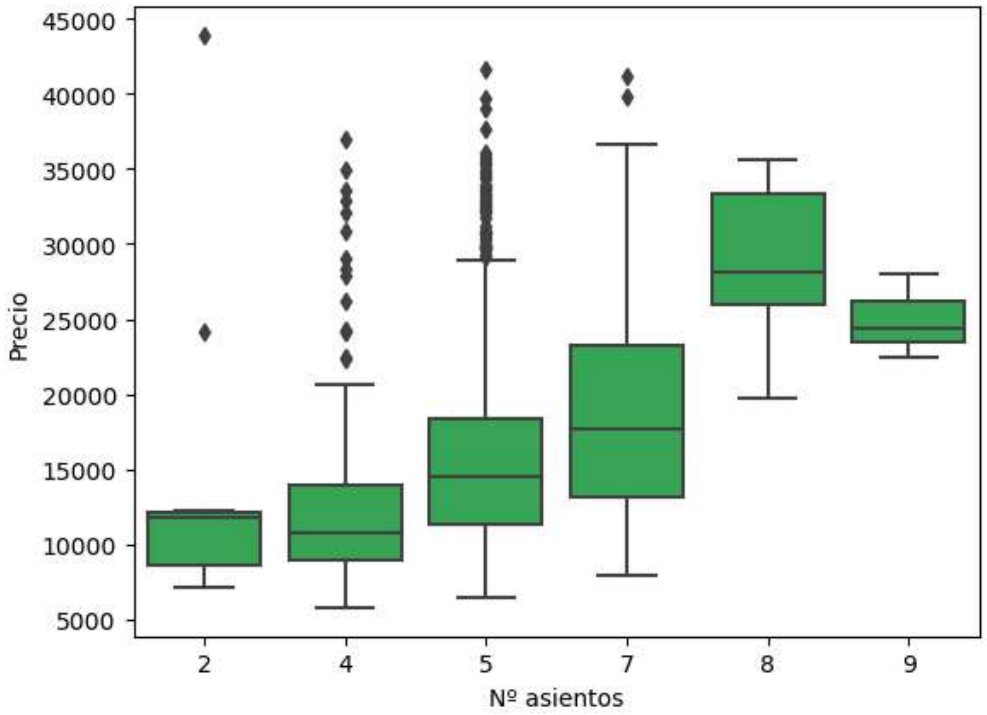
In [105]:
```
datos.groupby('Número de asientos').agg(frequency=('Número de asientos', "count"))
```

Out[105]:

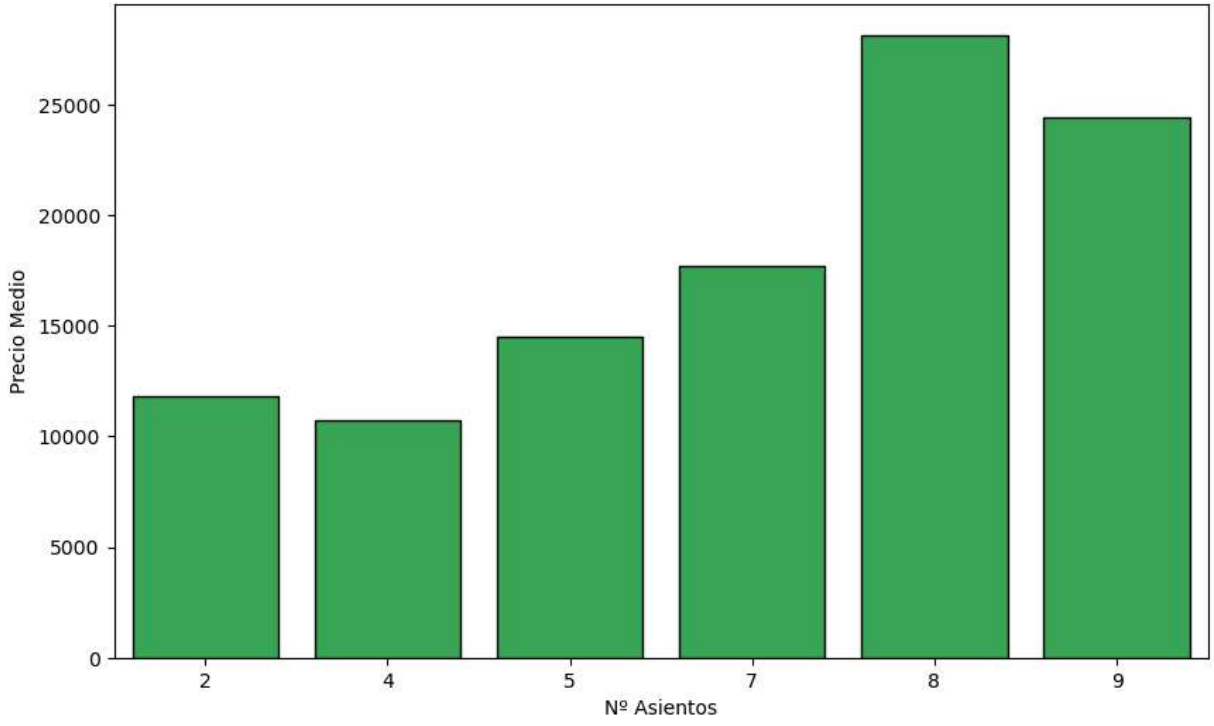
Número de asientos	frequency
2	13
4	122
5	1706
7	74
8	6
9	3

In [106]:
```
ax = sns.boxplot(x = 'Número de asientos', y = 'Precio', data = datos, color = '#28b84f')
ax.set( xlabel = 'Nº asientos')
```

Out[106]:
```
[Text(0.5, 0, 'Nº asientos')]
```



In [121]:
```
media_precios = datos.groupby('Número de asientos')['Precio'].median().reset_index()

# Crear el gráfico de barras
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Número de asientos', y='Precio', data=media_precios, color='#28b84f', edge

ax.set_xlabel('Nº Asientos')
ax.set_ylabel('Precio Medio')

# Mostrar el gráfico
plt.show()
```



El caso extremo de dos asientos es un BWM deportivo. ¿Deberíamos quitarlo?

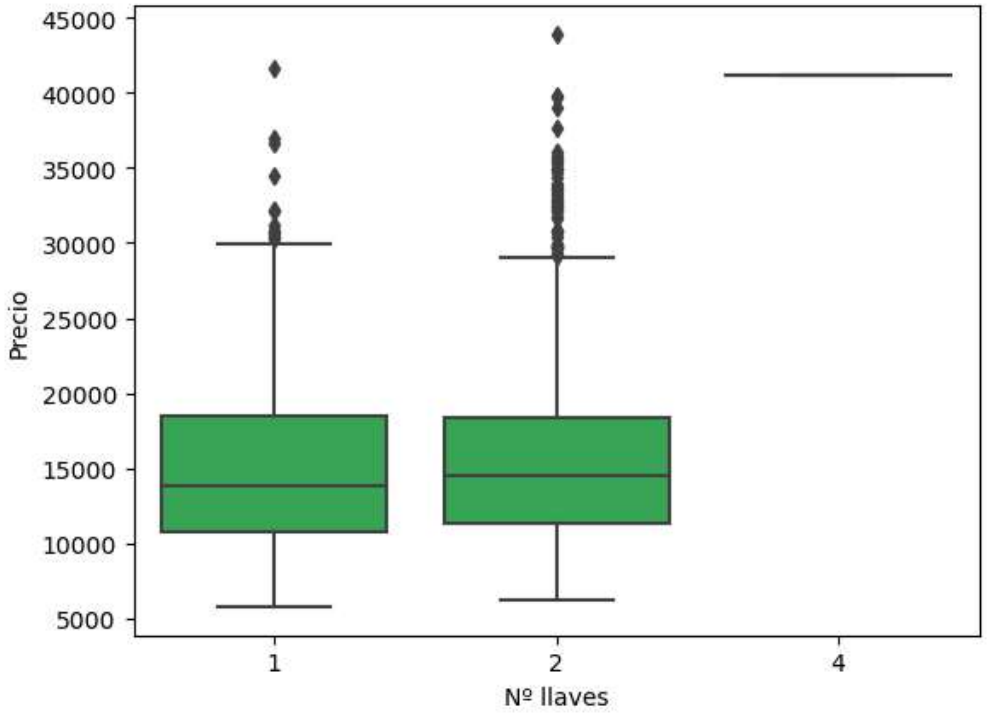
In [107]:
```
datos.groupby('Número de llaves').agg(frequency=('Número de llaves', "count"))
```

Out[107]:

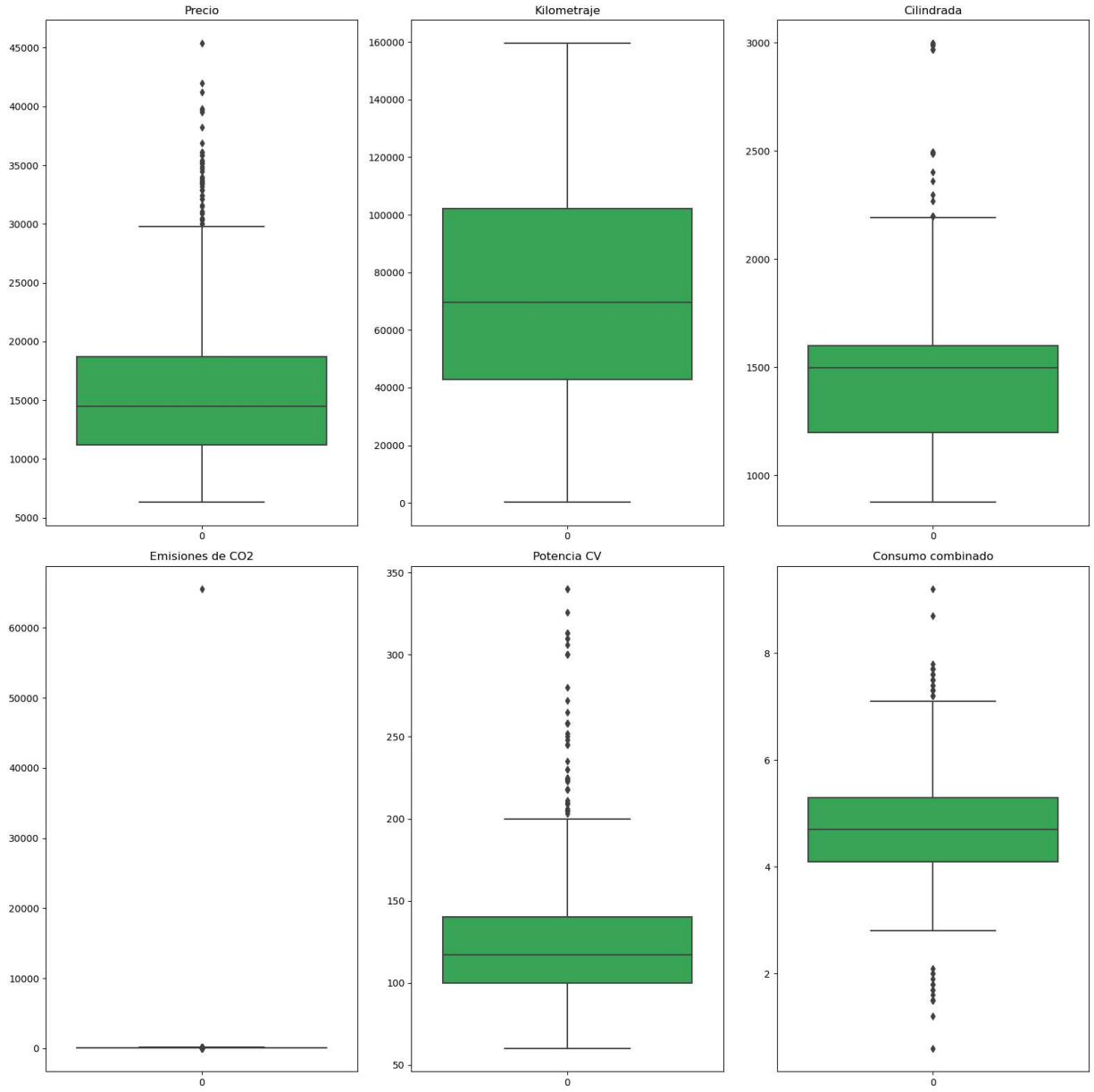
Número de llaves	frequency
1	341
2	1582
4	1

In [108]:
```
ax = sns.boxplot(x = 'Número de llaves', y = 'Precio', data = datos, color = '#28b84f')
ax.set( xlabel = 'Nº llaves')
```

Out[108]:
```
[Text(0.5, 0, 'Nº llaves')]
```



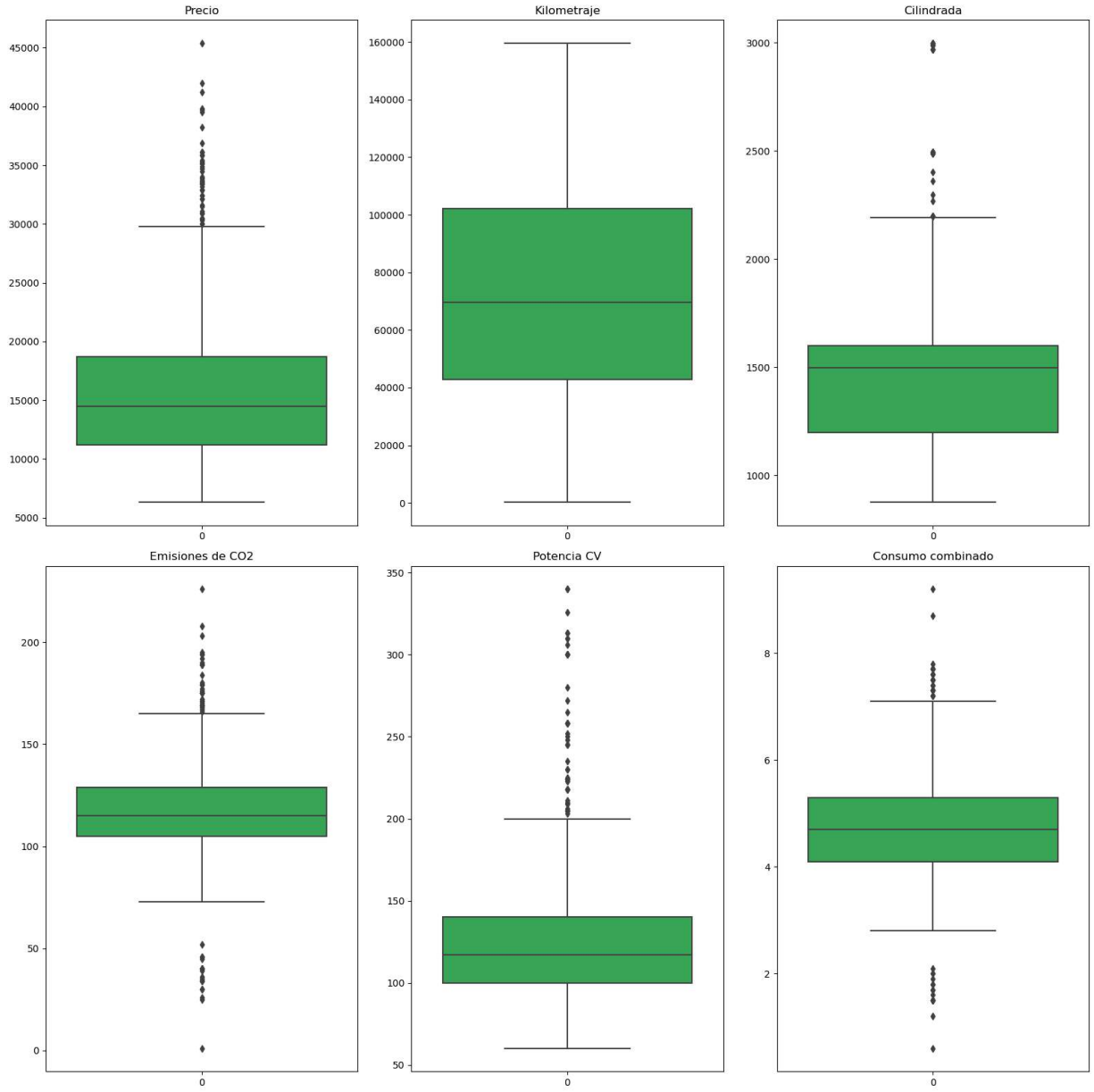
In [81]:
```
fig, axes = plt.subplots(2, 3, figsize=(16,16))
k = 0
for i in datos.select_dtypes(exclude=['object', 'datetime64', 'int']).columns:
    sns.boxplot(datos[i],ax = axes.flatten()[k], color = '#28b84f').set_title(i)
    k = k+1

plt.tight_layout()
plt.show()
```



Viendo los gráficos anteriores, sería conveniente descartar los valores extremos de emisiones de CO2 (ronda los 65000 y no tiene sentido) y de Consumo fuera (mayor que 35)

In [82]:
```
datos = datos[datos['Emisiones de CO2'] <= 60000]
```

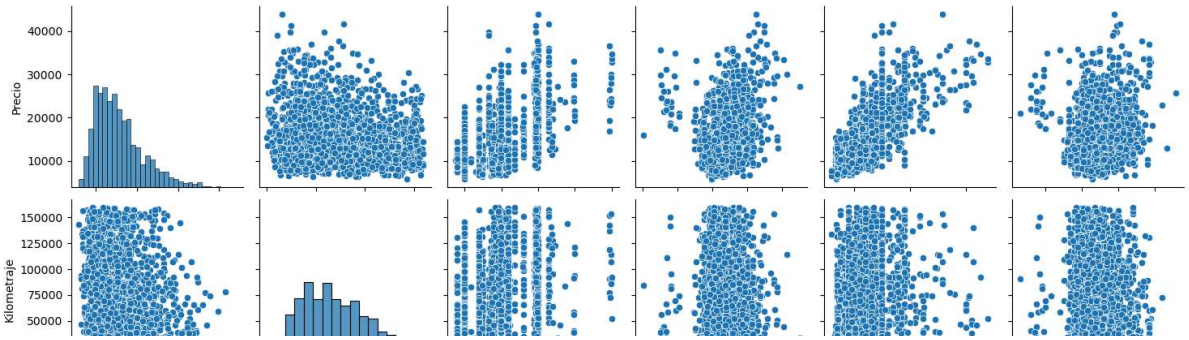
In [83]:
```
fig, axes = plt.subplots(2, 3, figsize=(16,16))
k = 0
for i in datos.select_dtypes(exclude=['object', 'datetime64', 'int']).columns:
    sns.boxplot(datos[i],ax = axes.flatten()[k], color = '#28b84f').set_title(i)
    k = k+1

plt.tight_layout()
plt.show()
```



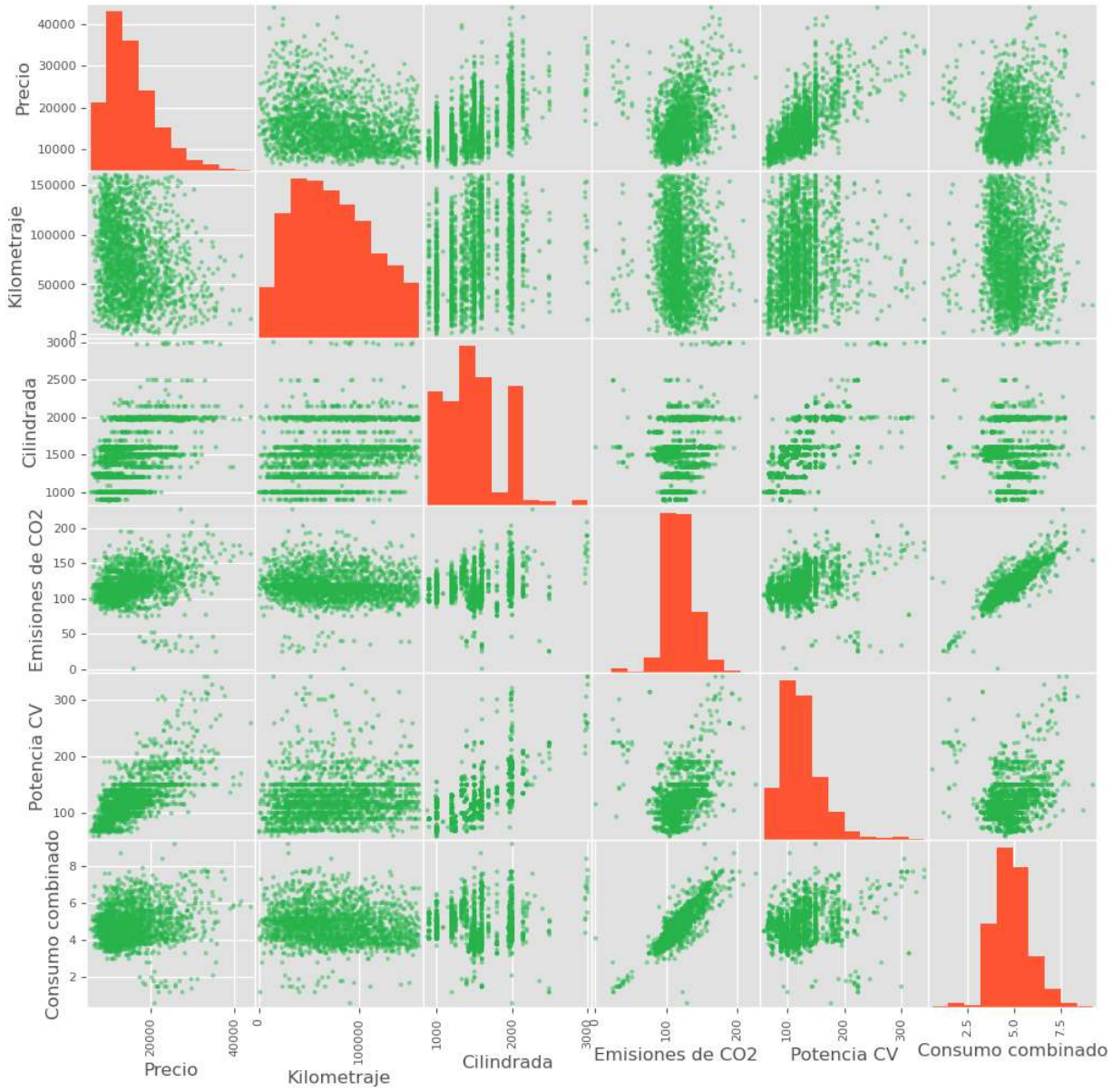
Seguimos teniendo algunos valores muy altos y muy bajos pero nada tan extremo

In [133]:
```
sns.pairplot(datos.select_dtypes(exclude=['object', 'datetime64','int']))
plt.show()
```

```
  palette` because no `hue` variable has been assigned.
  func(x=x, y=y, **kwargs)
C:\Users\antdu\anaconda3\Lib\site-packages\seaborn\axisgrid.py:1609: UserWarning: Ignoring
`palette` because no `hue` variable has been assigned.
  func(x=x, y=y, **kwargs)
C:\Users\antdu\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure
layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [141]:
```
from pandas.plotting import scatter_matrix
plt.style.use('ggplot')


# Crear matriz de diagramas de dispersión
scatter_matrix(datos.select_dtypes(exclude=['object', 'datetime64','int']), figsize=(12, 12),

plt.show()
```



Ahora haremos dummies de las variables categóricas que son interesantes en el estudio y convertimos en 0 y 1 las variables binarias:

- Marca
- Modelo
- Versión
- Carburante
- Transmisión
- Tracción
- Tipo de vehículo
- Color
- Tapicería
- Tipo de ruedas
- **Motor original**
- Eficiencia
- País de origen

- **Coche accidentado y reparado**
- **Tipo de IVA**

In [8]:
```
datos
```

Out[8]:

	Marca	Modelo	Precio	Primera matriculación	Kilometraje	Carburante	Transmisión	Potencia	Tracción
0	Audi	Q2	21199	2019-12-30	82489	Gasolina	Cambio tipo manual	116 CV / 85 kW	Tracción delantera
1	Seat	Arona	17199	2021-05-19	23640	Gasolina	Cambio tipo manual	110 CV / 81 kW	NaN
2	Opel	Adam	8799	2014-11-28	17256	Gasolina	Cambio tipo manual	87 CV / 64 kW	Tracción delantera
3	Ford	Fiesta	13899	2019-10-21	27990	Gasolina	Cambio tipo manual	100 CV / 74 kW	Tracción delantera
4	Seat	León	16499	2016-07-21	116873	Diésel	Cambio tipo manual	185 CV / 135 kW	Tracción delantera
5	Mercedes-Benz	Clase CLA	34599	2020-01-07	37561	Diésel	Cambio tipo automático	150 CV / 110 kW	Tracción delantera
							Cambio tipo	110 CV /	Tracción

In [54]:
```
#Binarias
datos['Motor original'] = datos['Motor original'].map({'Sí': 1, 'No': 0})
datos['Coche accidentado y reparado'] = datos['Coche accidentado y reparado'].map({'Sí': 1, 'N
datos['Tipo de IVA'] = datos['Tipo de IVA'].map({'IVA no deducible': 0, 'IVA deducible': 1})
```

In [55]:
```
#Categóricas
datos = pd.get_dummies(datos, columns=['Marca','Modelo',
                                       'Carburante','Transmisión','Tracción','Tipo de vehícu
                                       'Color','Tapicería','Tipo de ruedas','Clase de eficie
                                      'País de origen'],dtype=int,drop_first=True)
```

In [56]:
```
datos
```

Out[56]:

	Precio	Kilometraje	Potencia	Puertas	Número de asientos	Motor original	Cilindrada	Consumo	Emisiones de CO2	Estándar de calidad
0	21199.0	82489.0	116 CV / 85 kW	5	5	1	999.0	6.3 l/100 km (En la ciudad)5.4 l/100 km (Combi...	118.0	Ver estándar de calidad
1	8799.0	17256.0	87 CV / 64 kW	3	4	1	1398.0	6.6 l/100 km (En la ciudad)5 l/100 km (Combina...	119.0	Ver estándar de calidad
2	19899.0	28670.0	125 CV / 92 kW	5	5	1	999.0	5.6 l/100 km (Combinado)	122.0	Ver estándar de calidad

In [57]:
```
datos = datos.select_dtypes(exclude=['object', 'datetime64'])
```

In [58]:
```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1575 entries, 0 to 1575
Columns: 340 entries, Precio to Clase de eficiencia CO2_EURO 6d
dtypes: float64(6), int32(326), int64(8)
memory usage: 2.2 MB
```

In [16]:
```
datos.isnull().sum()
```

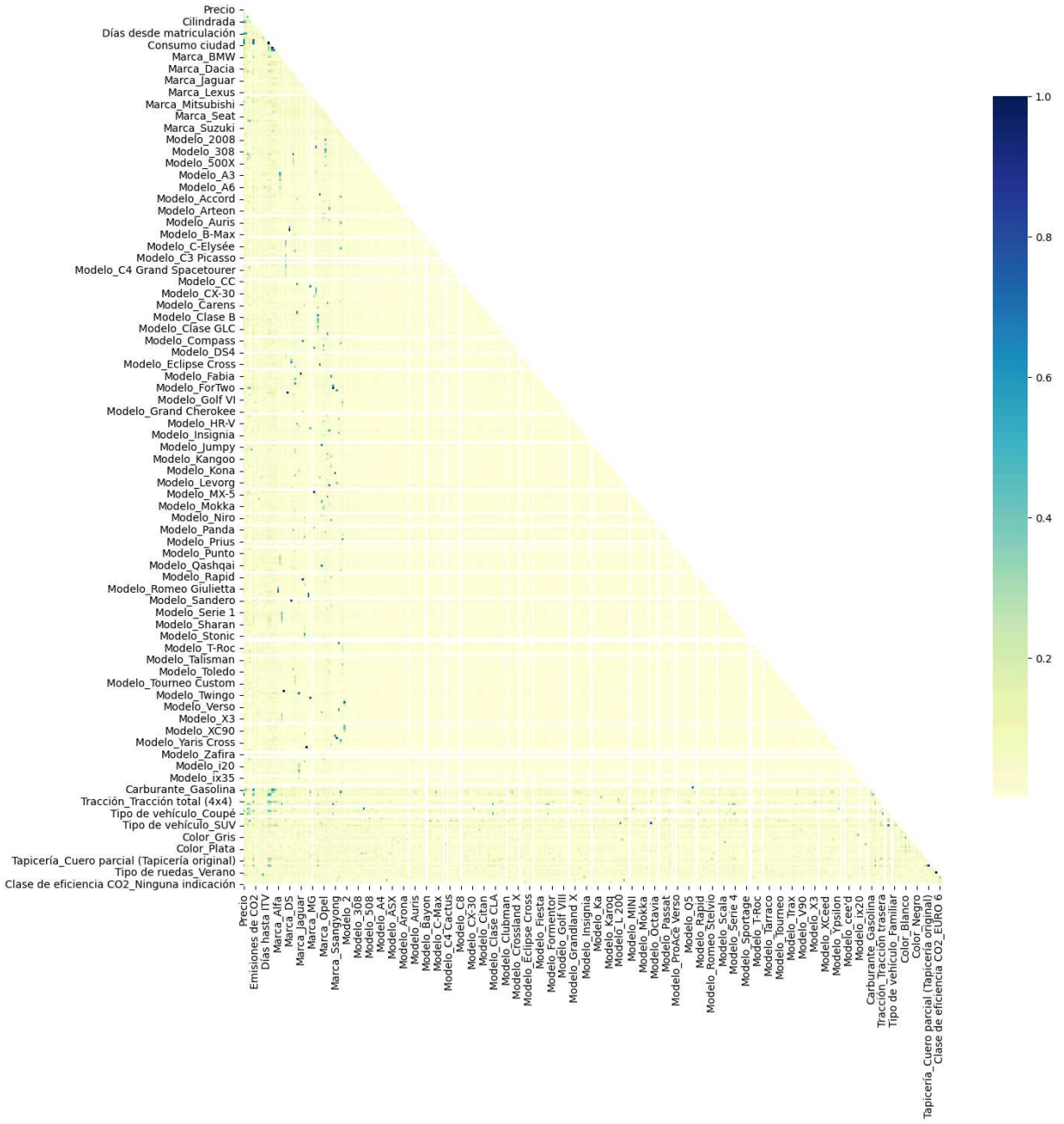
Out[16]:
```
Precio                              0
Kilometraje                         0
Puertas                             2
Número de asientos                  0
Motor original                      0
Cilindrada                          0
Emisiones de CO2                  185
Número de llaves                    0
Coche accidentado y reparado      185
Tipo de IVA                         0
Días desde matriculación            0
Días desde revisión               291
Días hasta ITV                    185
Potencia CV                         0
Potencia kW                         0
Consumo ciudad                    342
Consumo combinado                 342
Consumo fuera                     342
Marca_Alfa                          0
```

In [17]:
```
datos = datos.dropna()
```

In [18]:
```
datos.isnull().sum()
```

Out[18]:
```
Precio                            0
Kilometraje                       0
Puertas                           0
Número de asientos                0
Motor original                    0
Cilindrada                        0
Emisiones de CO2                  0
Número de llaves                  0
Coche accidentado y reparado      0
Tipo de IVA                       0
Días desde matriculación          0
Días desde revisión               0
Días hasta ITV                    0
Potencia CV                       0
Potencia kW                       0
Consumo ciudad                    0
Consumo combinado                 0
Consumo fuera                     0
Marca_Alfa                        0
```

In [33]:
```
correlacion = np.abs(datos.corr())
mask = np.zeros_like(correlacion)

mask[np.triu_indices_from(mask)] = True
plt.subplots(figsize=(15,15))
sns.heatmap(correlacion,mask=mask, cmap="YlGnBu", cbar_kws={"shrink": .8})
plt.show()
```



In [34]:
```
correlacion['Precio'].sort_values(ascending=False).head(10)
```

Out[34]:
```
Precio                                  1.000000
Potencia CV                             0.715906
Potencia kW                             0.715080
Transmisión_Cambio tipo manual          0.550729
Cilindrada                              0.522576
Días desde matriculación                0.504716
Tipo de vehículo_Coche pequeño          0.360304
Tapicería_Tejido (Tapicería original)   0.355276
Marca_Mercedes-Benz                     0.317953
Tracción_Tracción total (4x4)           0.302416
Name: Precio, dtype: float64
```

In [ ]:
```
datos.describe()
```