

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [4]: df = pd.read_excel('inmuebles.xlsx')
```

```
In [5]: df.head(5)
```

Out[5]:

	Referencia	FechaAlta	Tipo	Operación	Ciudad	Superficie	Precio Venta	Fecha Venta	Vendedor	Estatus	Días para Vender
0	1	2016-01-18	Estacionamiento	Alquiler	Cancún	215	1154980	2017-01-14	Joaquín	Vendida	362
1	2	2016-01-20	Oficina	Alquiler	Tijuana	287	2851058	2017-01-02	Joaquín	Vendida	348
2	3	2016-01-20	Local	Venta	Monterrey	40	321680	2017-01-11	Jesús	Vendida	357
3	4	2016-01-24	Industrial	Venta	Cancún	178	1142938	2017-01-23	Joaquín	Vendida	365
4	5	2016-02-03	Departamento	Venta	Monterrey	275	2141700	2017-01-08	Jesús	Vendida	340

```
In [6]: df.shape
```

Out[6]: (2651, 11)

```
In [7]: #Información de Los datos
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2651 entries, 0 to 2650
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Referencia            2651 non-null  int64
1   FechaAlta             2651 non-null  datetime64[ns]
2   Tipo                  2651 non-null  object
3   Operación             2651 non-null  object
4   Ciudad                2651 non-null  object
5   Superficie            2651 non-null  int64
6   Precio Venta          2651 non-null  int64
7   Fecha Venta           2651 non-null  datetime64[ns]
8   Vendedor              2387 non-null  object
9   Estatus               2651 non-null  object
10  Días para Vender      2651 non-null  int64
dtypes: datetime64[ns](2), int64(4), object(5)
memory usage: 227.9+ KB
```

```
In [9]: # Campos de La base de datos
for col in df.columns:
    print(col)
```

Referencia  
FechaAlta  
Tipo  
Operación  
Ciudad  
Superficie  
Precio Venta  
Fecha Venta  
Vendedor  
Estatus  
Días para Vender

```
In [10]: +
```

Out[10]:

	Superficie	Precio Venta	Días para Vender
count	2651.000000	2.651000e+03	2651.000000
mean	171.201811	1.211656e+06	365.289702
std	74.937985	6.261026e+05	200.002488
min	40.000000	1.639200e+05	90.000000
25%	106.000000	6.980235e+05	190.000000
50%	172.000000	1.140876e+06	323.000000
75%	237.000000	1.642624e+06	542.000000
max	300.000000	2.977318e+06	730.000000

In [11]:

```
#Precio promedio de venta por ciudad: Examina Los precios de venta promedio para cada ciudad.
df.groupby('Ciudad')['Precio Venta'].mean().round()
```

Out[11]:

Ciudad  
Cancún1180772.0  
Ciudad de México1213952.0  
Monterrey1218331.0  
Tijuana1233933.0  
Name: Precio Venta, dtype: float64

In [16]:

```
valor_minimo = df['Precio Venta'].min()
valor_maximo = df['Precio Venta'].max()
```

In [21]:

```
valor_minimo
```

Out[21]:

163920

In [23]:

```
valor_maximo
```

Out[23]:

2977318

In [25]:

```
# Promedio
promedio = df['Precio Venta'].mean()
print("Promedio:", promedio)

# Mediana
mediana = df['Precio Venta'].median()
print("Mediana:", mediana)

# Moda
moda = df['Precio Venta'].mode()
print("Moda:", moda)
```

Promedio: 1211656.4907582044  
Mediana: 1140876.0  
Moda: 0358670  
1416430  
2419256  
3446940  
4544896  
5574200  
6797215  
7913248  
8917378  
91191044  
101199394  
111342278  
122682225  
Name: Precio Venta, dtype: int64

In [27]:

```
#Grafico de lineas para mostrar Las ventas en cada mes

import matplotlib.pyplot as plt

# Crear el gráfico de línea
ax = df.groupby(df['Fecha Venta'].dt.to_period('M')).size().plot(kind='line')

# Obtener Los datos de X y Y del gráfico
x_values = ax.get_lines()[0].get_xdata()
y_values = ax.get_lines()[0].get_ydata()

# Agregar etiquetas de datos
for x, y in zip(x_values, y_values):
    plt.annotate(f'{int(y)}', xy=(x, y), xytext=(0, 5), textcoords='offset points', ha='center')
#Color y grosor de La Linea
ax = df.groupby(df['Fecha Venta'].dt.to_period('M')).size().plot(kind='line', color='red',linewidth=5)

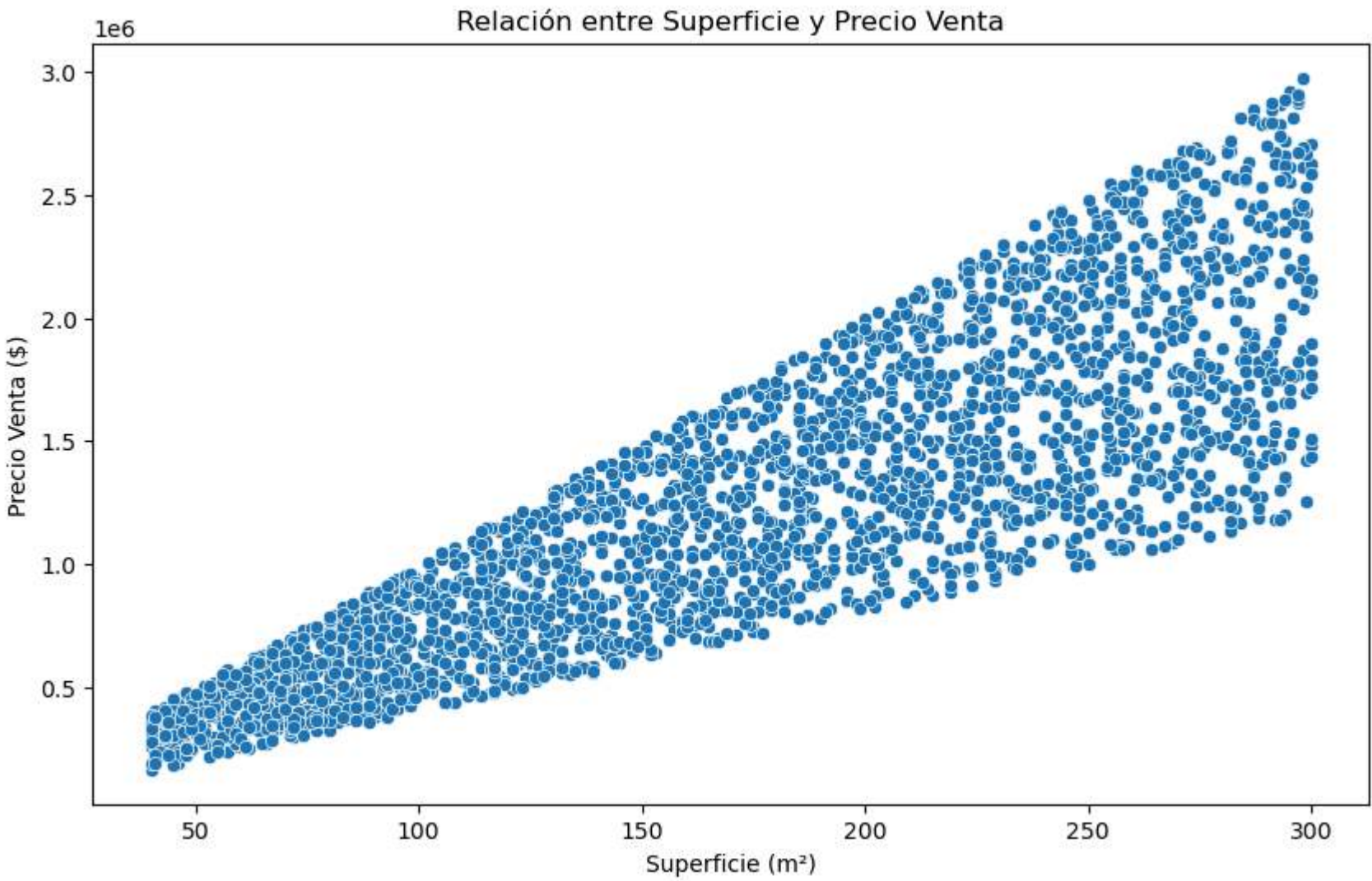
# Personalizar el gráfico
```

```
plt.title('Ventas por Meses')
plt.xlabel('Meses')
plt.ylabel('Número de Ventas')
plt.grid(True)
plt.show()
```



In [86]: *#Existe una variabilidad en Las ventas mensuales que probablemente puede estar relacionado con aspectos economicos, #estacionales o del mercado. Es importante resaltar que hay una tendencia de crecimiento finalizando el año de 2018 y empezand #Despues de puede observar una caida fuerte. Lo que implica un analisis mas profundo para conocer posibles causas.*

```
In [33]: # Gráfico de dispersión
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Superficie', y='Precio Venta')
plt.title('Relación entre Superficie y Precio Venta')
plt.xlabel('Superficie (m²)')
plt.ylabel('Precio Venta ($)')
plt.show()
```



In [34]: *#Este es un estilo de consulta con el numero que ponga. Lo va a buscar a la base de datos y selecciona toda la fila*  
`df.iloc[874]`

```
Out[34]: Referencia      875
FechaAlta      2017-03-20 00:00:00
Tipo      Oficina
Operación      Alquiler
Ciudad      Tijuana
Superficie      274
Precio Venta      1152718
Fecha Venta      2017-07-31 00:00:00
Vendedor      Pedro
Estatus      Vendida
Días para Vender      133
Name: 874, dtype: object
```

```
In [35]: #Busca desde el 18 hasta el 20
df.iloc[18:20]
```

```
Out[35]:
```

	Referencia	FechaAlta	Tipo	Operación	Ciudad	Superficie	Precio Venta	Fecha Venta	Vendedor	Estatus	Días para Vender
18	19	2016-03-02	Oficina	Venta	Cancún	172	1063648	2017-02-13	María	Vendida	348
19	20	2016-03-06	Estacionamiento	Venta	Monterrey	169	1370759	2017-01-14	Luisa	Vendida	314

```
In [36]: #Busca en la columna Vendedor, que este en la fila 500
df['Vendedor'][500]
```

```
Out[36]: 'Luisa'
```

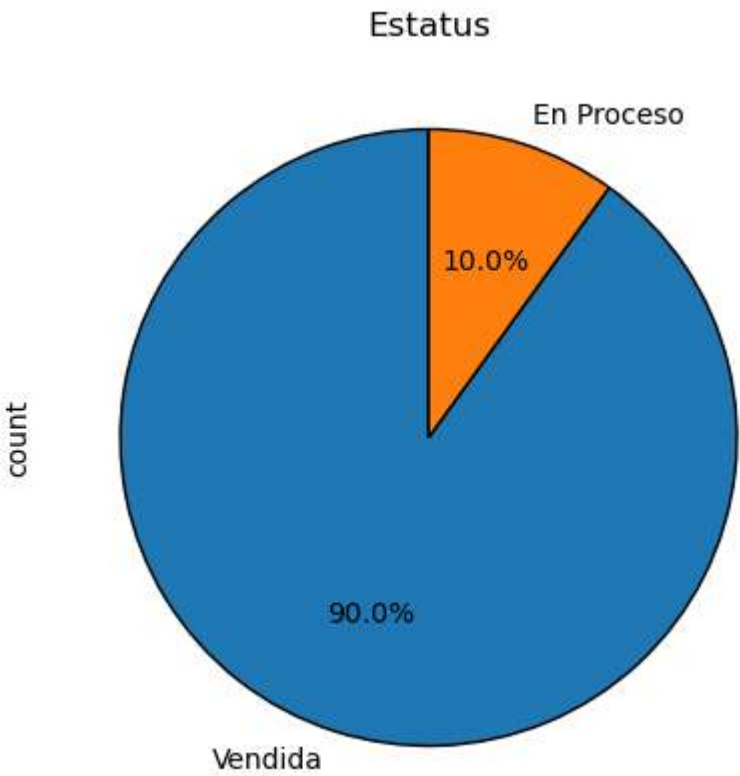
```
In [37]: #Muestra aleatoria
df.sample(5)
```

```
Out[37]:
```

	Referencia	FechaAlta	Tipo	Operación	Ciudad	Superficie	Precio Venta	Fecha Venta	Vendedor	Estatus	Días para Vender
1597	1598	2016-12-14	Casa	Venta	Ciudad de México	216	2046384	2018-07-03	Pedro	Vendida	566
320	321	2016-08-30	Oficina	Venta	Ciudad de México	274	1577692	2017-05-11	Joaquín	Vendida	254
2282	2283	2017-08-13	Local	Venta	Cancún	98	957264	2018-11-11	NaN	En Proceso	455
1138	1139	2016-06-22	Casa	Venta	Monterrey	274	2594780	2018-02-23	María	Vendida	611
284	285	2016-08-20	Industrial	Venta	Monterrey	119	711382	2017-08-01	Luisa	Vendida	346

```
In [40]: df['Estatus'].value_counts().plot(kind='pie',
      title='Estatus',
      autopct='%1.1f%%', # Muestra el porcentaje con 1 decimal
      startangle=90, # Para que el gráfico comience desde arriba
      figsize=(5, 5), # Tamaño del gráfico
      wedgeprops={'edgecolor': 'black'}) # Bordes de los sectores
```

```
Out[40]: <Axes: title={'center': 'Estatus'}, ylabel='count'>
```





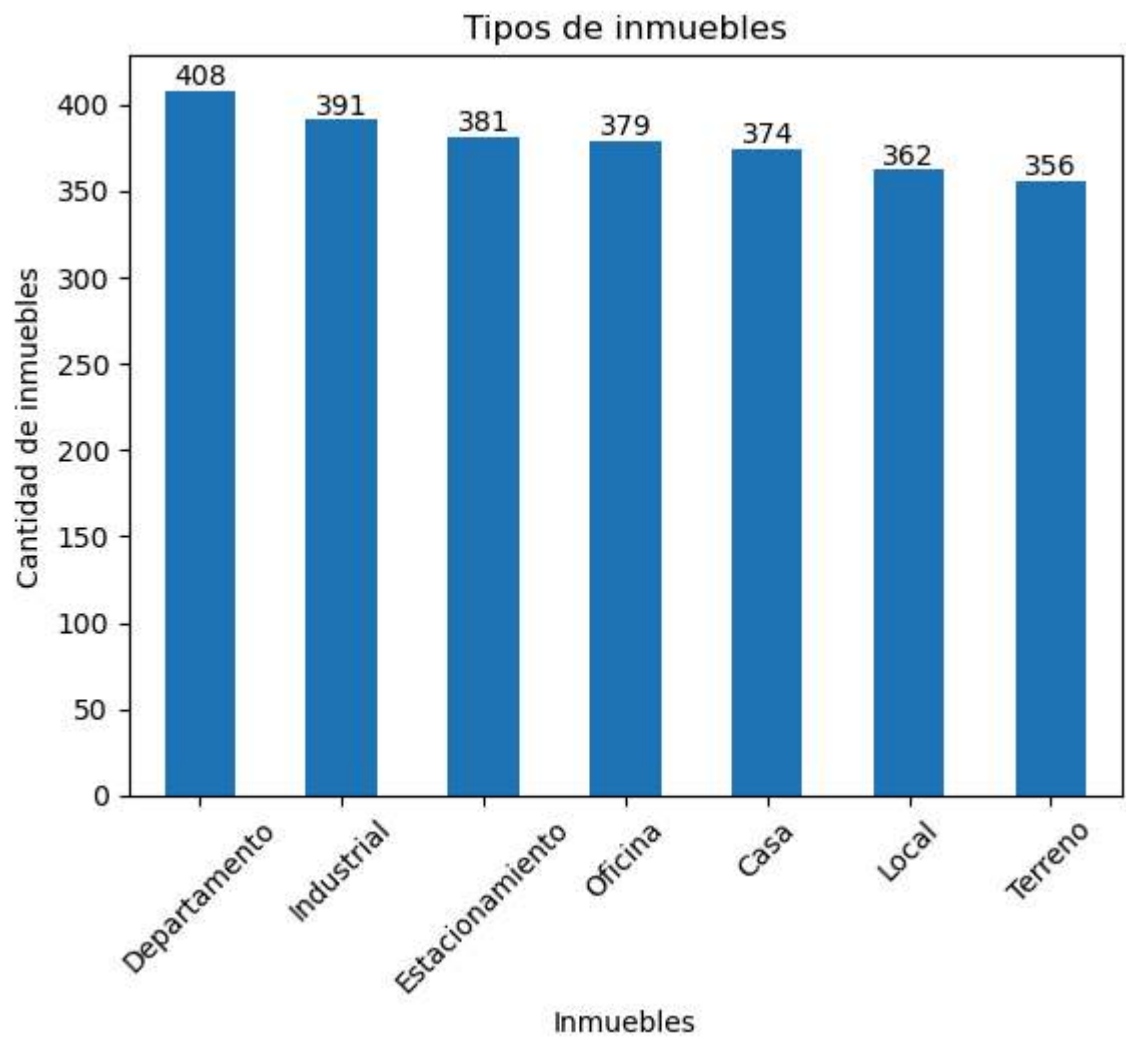
```
In [76]: import matplotlib.pyplot as plt
import pandas as pd

# Contar La frecuencia de cada tipo de inmueble
conteo_tipos = df['Tipo'].value_counts()

# Crear el gráfico de barras
conteo_tipos.plot(kind='bar', title='Tipos de inmuebles')

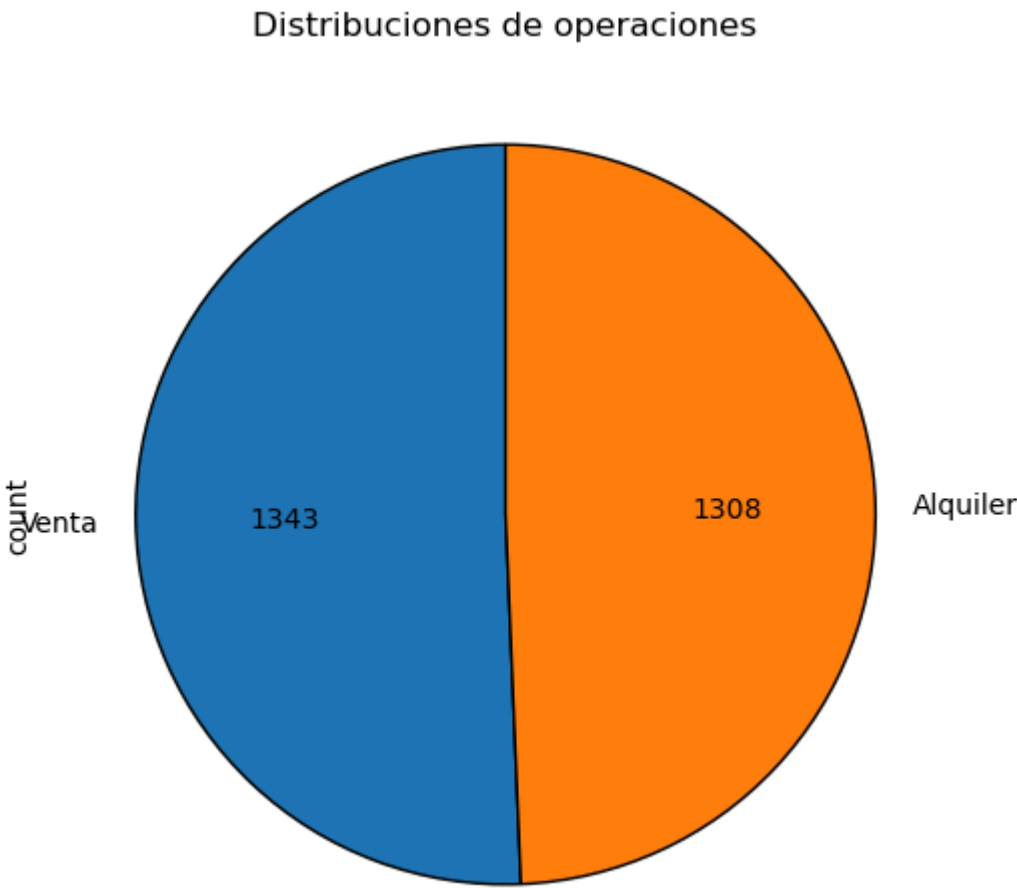
# Agregar etiquetas a Las barras
for i in range(len(conteo_tipos)):
    plt.text(i, conteo_tipos.iloc[i], str(conteo_tipos.iloc[i]), ha='center', va='bottom')

# Etiquetas de Los ejes
plt.xlabel('Inmuebles')
plt.ylabel('Cantidad de inmuebles')
plt.xticks(rotation=45)
# Mostrar el gráfico
plt.show()
```



```
In [45]: #Alquiler vs Ventas
df['Operación'].value_counts().plot(kind='pie',
                                   title='Distribuciones de operaciones',
                                   autopct=lambda p: '{:.0f}'.format(p * sum(df['Operación'].value_counts()) / 100), # Muest
                                   startangle=90,
                                   figsize=(6, 6),
                                   wedgeprops={'edgecolor': 'black'})
```

Out[45]: <Axes: title={'center': 'Distribuciones de operaciones'}, ylabel='count'>



In [78]: *#Este grafico de pie nos muestra un breve resumen de las ventas vs los alquileres. Donde las ventas superan ligeramente los que son de alquiler.*

```
In [ ]: #Grafico de Barras horizontales de vendedores

import matplotlib.pyplot as plt

# Contar la frecuencia de cada tipo de inmueble
conteo_tipos = df['Vendedor'].value_counts()

# Crear el gráfico de barras horizontales
conteo_tipos.plot(kind='barh', title='Vendedores')

# Agregar etiquetas a las barras (ahora en el eje horizontal)
for i, v in enumerate(conteo_tipos):
    plt.text(v, i, str(v), va='center')

# Etiquetas de los ejes
plt.xlabel('Cantidad de ventas')
plt.ylabel('Nombre del vendedor')

# Mostrar el gráfico
plt.show()
```

In [ ]: Todos los vendedores tienen un desempeño bastante cercano, donde el primero hasta el último solo hay una diferencia de 45 unidades, esta brecha puede ser por diferentes factores como lo son la experiencia, la red de contactos, la atención al cliente o las capacidades de negociación. Para mejorar estos aspectos se puede hacer:

- Bridar capacitaciones en técnicas de ventas, manejo de objeciones y cierre de ventas.
- Juntar los mejores vendedores con aquellos que están por debajo, soluciones personalizadas e incentivos.

Podemos destacar a Luisa y a Carmen como vendedoras estrellas, ya que sus ventas superan el promedio en comparación con sus compañeros.

```
In [80]: import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Seleccionar las características relevantes
X = df[['Superficie', 'Precio Venta']]

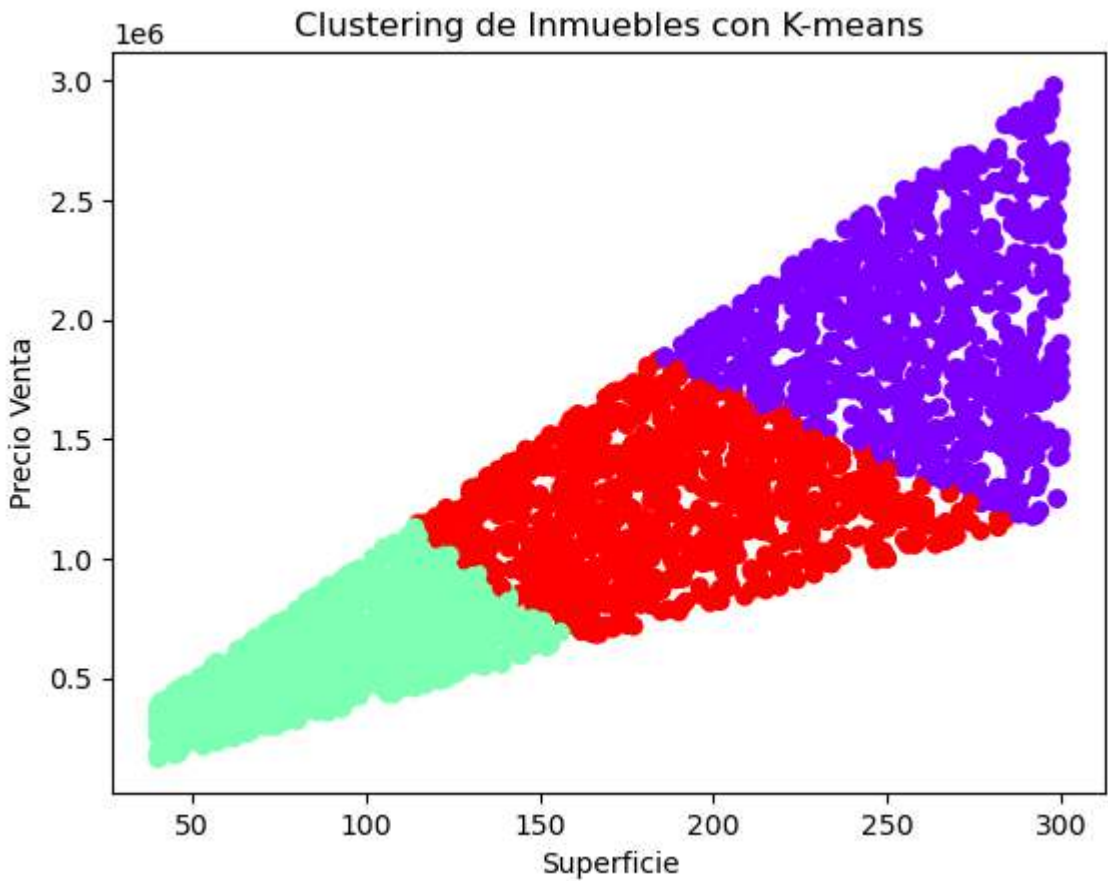
# Escalar los datos
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Numero de clusters
num_clusters = 3

# Crear y ajustar el modelo K-means
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X_scaled)

# Agregar los labels de los clusters al DataFrame original
df['Cluster'] = kmeans.labels_
```

```
# Visualizar los clusters
plt.scatter(df['Superficie'], df['Precio Venta'], c=df['Cluster'], cmap='rainbow')
plt.xlabel('Superficie')
plt.ylabel('Precio Venta')
plt.title('Clustering de Inmuebles con K-means')
plt.show()
```



In [ ]: El grafico de K-means podemos resaltar tres grupos, con características distintas en cuanto tamaño y precio. Podemos concluir que podemos tener tres nichos como por ejemplo viviendas de lujo, viviendas con tamaño medio y por ultimo casas pequeñas. Tenemos una relacion positiva debido a que ambas variables "Superficie" y "Precio Venta" aumentan, no es un patron exacto pero hay una tendencia.

```
In [ ]: df.head(1)
```

```
In [ ]: import matplotlib.pyplot as plt
import seaborn as sns

# Descripción de los datos
print(df['Precio Venta'].describe())

# Configuración de la figura
plt.figure(figsize=(6, 5))

# Trazar el histograma y la curva de densidad
sns.histplot(df['Precio Venta'], color='blue', bins=100, kde=True, stat="density", alpha=0.4)

# Mostrar el gráfico
plt.title('Histograma de Precio Venta')
plt.xlabel('Precio Venta')
plt.ylabel('Densidad')
plt.show()
```

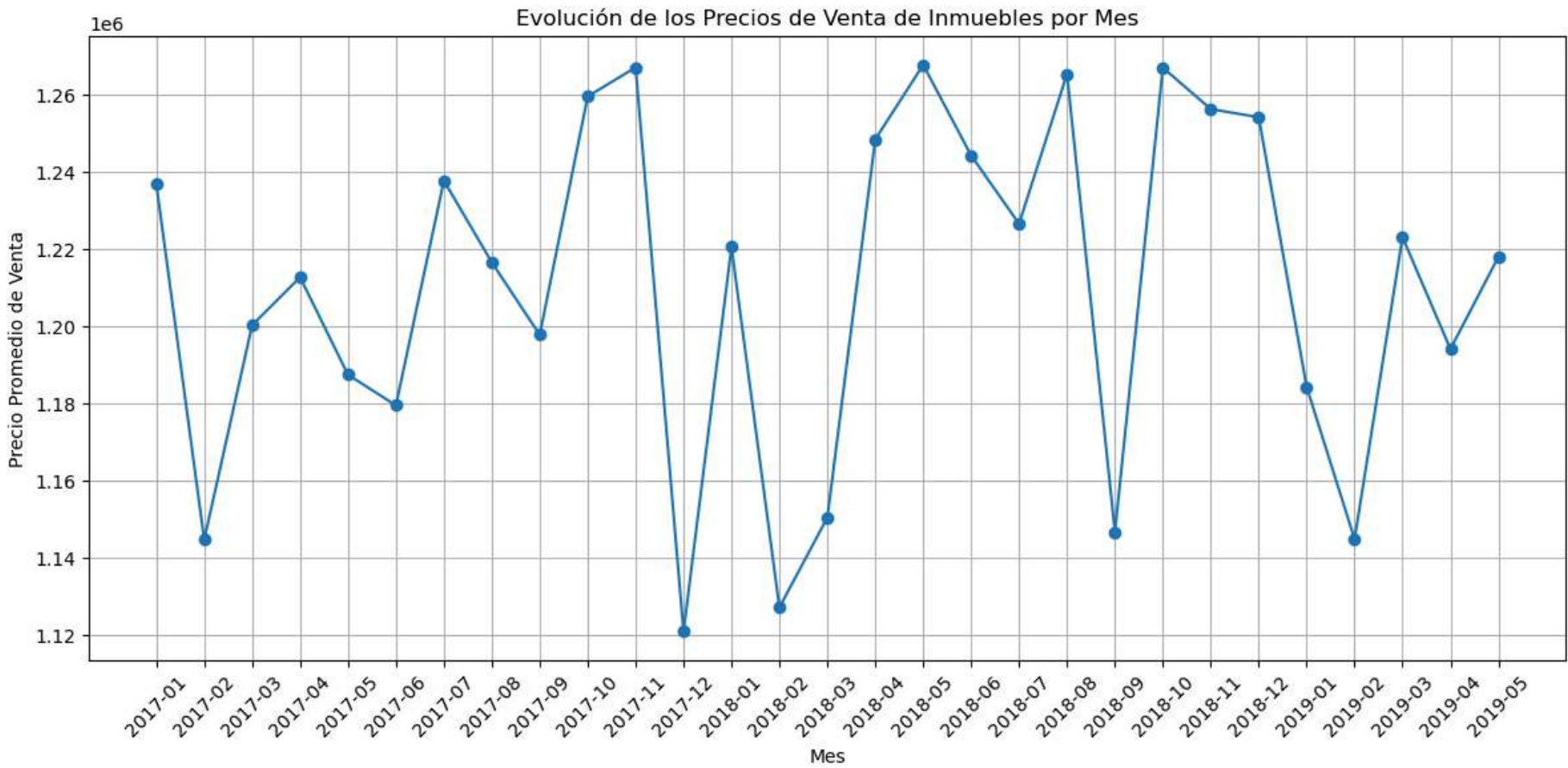
```
In [54]: #Propiedades que mas se demoraron en vender
propiedades_lentas = df.sort_values(by='Días para Vender', ascending=False)
print(propiedades_lentas.head(15).to_string())
```

	Referencia	FechaAlta	Tipo Operación		Ciudad	Superficie	Precio Venta	Fecha Venta	Vendedor	Estatu
s	Días para Vender									
1427	1428	2016-10-06	Terreno	Alquiler	Ciudad de México	132	882420	2018-10-06	Carmen	Vendid
a	730									
1484	1485	2016-10-28	Terreno	Venta	Tijuana	249	1878954	2018-10-28	Luisa	Vendid
a	730									
1482	1483	2016-10-28	Departamento	Alquiler	Monterrey	164	1421060	2018-10-28	Luisa	Vendid
a	730									
898	899	2016-03-30	Industrial	Alquiler	Monterrey	167	1191044	2018-03-30	Joaquín	Vendid
a	730									
929	930	2016-04-09	Local	Venta	Monterrey	290	1702590	2018-04-09	Pedro	Vendid
a	730									
1046	1047	2016-05-21	Casa	Alquiler	Tijuana	97	834394	2018-05-20	Jesús	Vendid
a	729									
893	894	2016-03-27	Casa	Venta	Cancún	196	849464	2018-03-26	Jesús	Vendid
a	729									
721	722	2016-01-28	Oficina	Venta	Cancún	110	943800	2018-01-26	Jesús	Vendid
a	729									
1610	1611	2016-12-20	Local	Venta	Ciudad de México	281	2671748	2018-12-18	Carmen	Vendid
a	728									
1554	1555	2016-11-25	Estacionamiento	Alquiler	Cancún	295	2613405	2018-11-23	Carmen	Vendid
a	728									
1395	1396	2016-09-29	Industrial	Alquiler	Tijuana	121	490897	2018-09-27	María	Vendid
a	728									
1488	1489	2016-10-30	Estacionamiento	Alquiler	Ciudad de México	103	536733	2018-10-28	María	Vendid
a	728									
1326	1327	2016-08-31	Industrial	Venta	Tijuana	158	1473982	2018-08-29	Carmen	Vendid
a	728									
1606	1607	2016-12-17	Local	Venta	Tijuana	165	1393095	2018-12-15	Carmen	Vendid
a	728									
1337	1338	2016-09-07	Departamento	Alquiler	Cancún	99	938916	2018-09-04	Pedro	Vendid
a	727									

In [68]: df['Fecha Venta'] = pd.to\_datetime(df['Fecha Venta'])

In [70]: df['Mes'] = df['Fecha Venta'].dt.to\_period('M') # Agrupar por mes  
precios\_promedio\_mensuales = df.groupby('Mes')['Precio Venta'].mean().reset\_index()

In [74]: plt.figure(figsize=(12, 6))  
plt.plot(precios\_promedio\_mensuales['Mes'].astype(str), precios\_promedio\_mensuales['Precio Venta'], marker='o')  
plt.title('Evolución de los Precios de Venta de Inmuebles por Mes')  
plt.xlabel('Mes')  
plt.ylabel('Precio Promedio de Venta')  
plt.xticks(rotation=45) # Rotar las etiquetas del eje x para mejor legibilidad  
plt.grid()  
plt.tight\_layout() # Ajustar el diseño para que no se superpongan elementos  
plt.show()



In [ ]: #¿Cómo han evolucionado los precios de venta de los inmuebles en los últimos años?  
# Para analizar la evolución de los precios de las ventas se agrupa los promedios por meses,  
# esto permite visualizar la evolución de los precios a lo largo del tiempo.

In [88]: df\_ventas = df[df['Operación'] == 'Venta']  
precios\_promedio = df\_ventas.groupby(['Ciudad', 'Tipo'])['Precio Venta'].mean().reset\_index()  
precios\_promedio.rename(columns={'Precio Venta': 'PrecioPromedio'}, inplace=True)  
  
# Formatear la columna 'PrecioPromedio'  
precios\_promedio['PrecioPromedio'] = precios\_promedio['PrecioPromedio'].apply(lambda x: f"{x:,.2f}")



```
# Imprimir el DataFrame
print(precios_promedio)
```

	Ciudad	Tipo	PrecioPromedio
0	Cancún	Casa	1,106,847.25
1	Cancún	Departamento	1,032,479.60
2	Cancún	Estacionamiento	1,229,344.48
3	Cancún	Industrial	1,307,819.64
4	Cancún	Local	1,278,478.36
5	Cancún	Oficina	1,122,743.72
6	Cancún	Terreno	1,026,276.72
7	Ciudad de México	Casa	1,228,839.05
8	Ciudad de México	Departamento	1,168,701.49
9	Ciudad de México	Estacionamiento	1,149,324.59
10	Ciudad de México	Industrial	1,283,821.59
11	Ciudad de México	Local	1,300,523.68
12	Ciudad de México	Oficina	1,207,923.62
13	Ciudad de México	Terreno	1,309,398.07
14	Monterrey	Casa	1,282,966.78
15	Monterrey	Departamento	1,365,772.85
16	Monterrey	Estacionamiento	1,180,556.28
17	Monterrey	Industrial	1,006,846.96
18	Monterrey	Local	1,248,935.69
19	Monterrey	Oficina	1,305,124.52
20	Monterrey	Terreno	1,350,894.64
21	Tijuana	Casa	1,260,817.66
22	Tijuana	Departamento	1,291,555.63
23	Tijuana	Estacionamiento	1,207,011.12
24	Tijuana	Industrial	1,177,282.88
25	Tijuana	Local	1,211,232.09
26	Tijuana	Oficina	1,265,099.91
27	Tijuana	Terreno	1,200,037.53

In [ ]: