

November 20, 2024

1 Primer Proyecto Integrador

2 Índice

1. Limpieza de datos
 - Carga de datos
 - Duplicacios y valores ausentes
 - Tipo de datos
 - Creacion de columnas
2. Análisis de datos
 - Juegos por año
 - Ventas
 - Rentabilidad
3. Perfil de usuario para cada region
 - Plataformas principales
 - Generos principales
 - Clasificacion ESRB
4. Prueba de hipotesis
 - Calificaciones promedio de los usuarios para las plataformas Xbox One y PC
 - calificaciones promedio de los usuarios para los géneros de Acción y Deportes
5. Conclusion general

2.0.1 Limpieza de datos

```
[45]: import pandas as pd
import numpy as np
import scipy.stats as sts
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[3]: games = pd.read_csv("/datasets/games.csv")
games.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
#   Column                Non-Null Count  Dtype
```

```

0   Name                16713 non-null object
1   Platform            16715 non-null object
2   Year_of_Release     16446 non-null float64
3   Genre               16713 non-null object
4   NA_sales            16715 non-null float64
5   EU_sales            16715 non-null float64
6   JP_sales            16715 non-null float64
7   Other_sales         16715 non-null float64
8   Critic_Score        8137 non-null float64
9   User_Score          10014 non-null object
10  Rating              9949 non-null object

```

dtypes: float64(6), object(5)

memory usage: 1.4+ MB

```
[4]: games.columns = games.columns.str.lower()
```

```
[5]: games['year_of_release'].isna().sum()
```

[5]: 269

```
[6]: games= games.dropna(subset=['year_of_release'])

games['year_of_release'] = games['year_of_release'].astype("int")

print(games.head())
```

	name	platform	year_of_release	genre	na_sales	\
0	Wii Sports	Wii	2006	Sports	41.36	
1	Super Mario Bros.	NES	1985	Platform	29.08	
2	Mario Kart Wii	Wii	2008	Racing	15.68	
3	Wii Sports Resort	Wii	2009	Sports	15.61	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	11.27	

	eu_sales	jp_sales	other_sales	critic_score	user_score	rating
0	28.96	3.77	8.45	76.0	8	E
1	3.58	6.81	0.77	NaN	NaN	NaN
2	12.76	3.79	3.29	82.0	8.3	E
3	10.93	3.28	2.95	80.0	8	E
4	8.89	10.22	1.00	NaN	NaN	NaN

```
[7]: games['year_of_release'] = pd.to_datetime(games['year_of_release'],
        ↪format='%Y').dt.year
print(games['year_of_release'].dtypes)
```

int64

```
[8]: games["critic_score"]=games["critic_score"].fillna(0)
games["critic_score"]=games["critic_score"].astype(int)

games["user_score"]=games["user_score"].fillna(0)
games["user_score"]=games["user_score"].replace("tbd", 0)
games["user_score"]=games["user_score"].astype(float)

games["rating"]=games["rating"].fillna(" ")

games ["genre"] = games["genre"].fillna(" ")
```

Se modificaron las columnas : “critic_score” y “user_score” tenían valores ausentes, mediante el metodo fillna, dichos valores fueron cambiados por cero, en la columna user_score en algunas celdas estaba la abreviatura TBD, que fue tambien reemplaza por ceros y esta columna paso de ser object a ser de tipo float.

igualmente la columna critic_score fue cambiada de float a int.

La columna yer_of_release. los valores ausentes fueron eliminados mediante dropna, ya que eran valores ausentes aleatorios y no hay un gran riesgo de perder informacion valiosa. y paso de ser float a int.

Por ultimo la columna “rating” los valores ausentes se dejaron en blanco.

```
[9]: games ["total_sales"] = games["na_sales"] + games["jp_sales"] + games_
↳ ["eu_sales"] + games["other_sales"]

print(games.head())
```

	name	platform	year_of_release	genre	na_sales	\
0	Wii Sports	Wii	2006	Sports	41.36	
1	Super Mario Bros.	NES	1985	Platform	29.08	
2	Mario Kart Wii	Wii	2008	Racing	15.68	
3	Wii Sports Resort	Wii	2009	Sports	15.61	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	11.27	

	eu_sales	jp_sales	other_sales	critic_score	user_score	rating	\
0	28.96	3.77	8.45	76	8.0	E	
1	3.58	6.81	0.77	0	0.0		
2	12.76	3.79	3.29	82	8.3	E	
3	10.93	3.28	2.95	80	8.0	E	
4	8.89	10.22	1.00	0	0.0		

	total_sales
0	82.54
1	40.24
2	35.52
3	32.77
4	31.38

2.0.2 Analisis de datos

```
[10]: juegos_por_ano = games['year_of_release'].value_counts().sort_index()

print(juegos_por_ano)
```

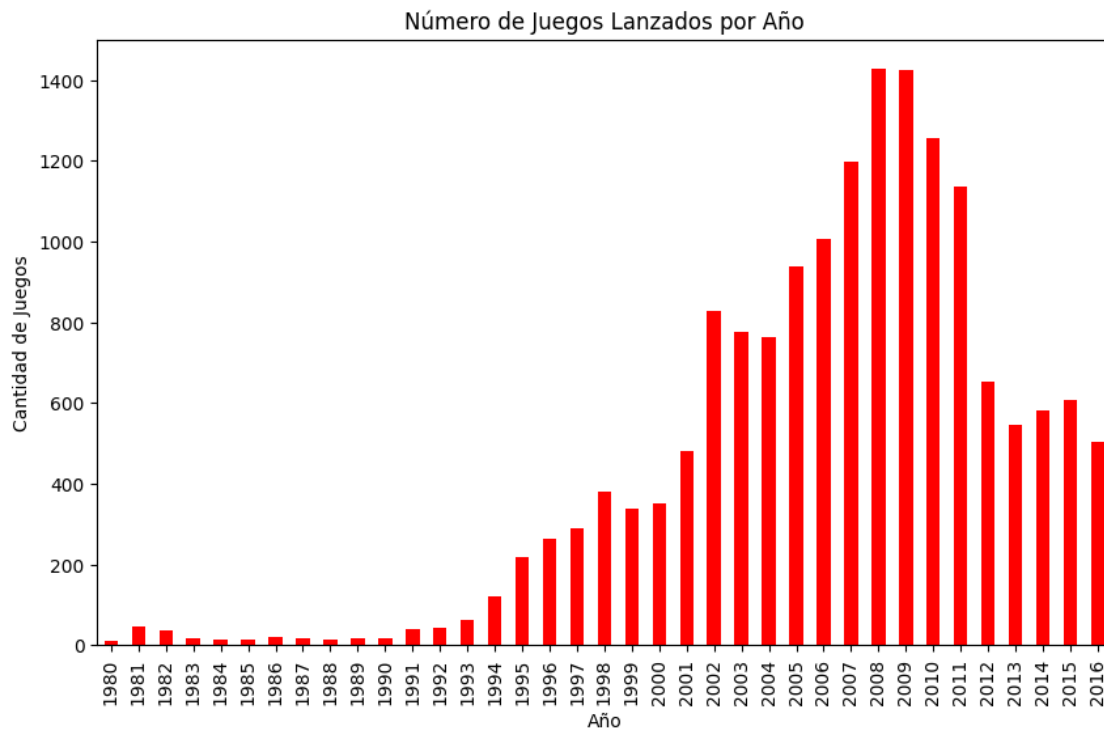
1980	9
1981	46
1982	36
1983	17
1984	14
1985	14
1986	21
1987	16
1988	15
1989	17
1990	16
1991	41
1992	43
1993	62
1994	121
1995	219
1996	263
1997	289
1998	379
1999	338
2000	350
2001	482
2002	829
2003	775
2004	762
2005	939
2006	1006
2007	1197
2008	1427
2009	1426
2010	1255
2011	1136
2012	653
2013	544
2014	581
2015	606
2016	502

Name: year_of_release, dtype: int64

```
[11]: color = "red"

juegos_por_ano.plot(kind='bar',color=color, figsize=(10, 6))
plt.title('Número de Juegos Lanzados por Año')
```

```
plt.xlabel('Año')
plt.ylabel('Cantidad de Juegos')
plt.show()
```



Es significativo el incremento en la cantidad de juegos lanzados por año, estamos hablando que entre el primer año de registro que es 1980 a 2008 (que es el año con mayor numero de lanzamiento) hay un incremento de mas del 15.000%, y tambien es significativo el descenso para el periodo de 2016.

```
[12]: ventas_totales_plataforma = games.groupby('platform')['total_sales'].sum()

print("Ventas totales por plataforma:")
print(ventas_totales_plataforma)
```

Ventas totales por plataforma:

```
platform
2600      86.48
3D0        0.10
3DS     257.81
DC       15.95
DS     802.78
GB     254.43
GBA    312.88
```

GC	196.73
GEN	30.77
GG	0.04
N64	218.01
NES	251.05
NG	1.44
PC	255.76
PCFX	0.03
PS	727.58
PS2	1233.56
PS3	931.34
PS4	314.14
PSP	289.53
PSV	53.81
SAT	33.59
SCD	1.86
SNES	200.04
TG16	0.16
WS	1.42
Wii	891.18
WiiU	82.19
X360	961.24
XB	251.57
XOne	159.32

Name: total_sales, dtype: float64

```
[13]: plataformas_top = ventas_totales_plataforma.sort_values(ascending=False).
      ↪head(5).index.tolist()

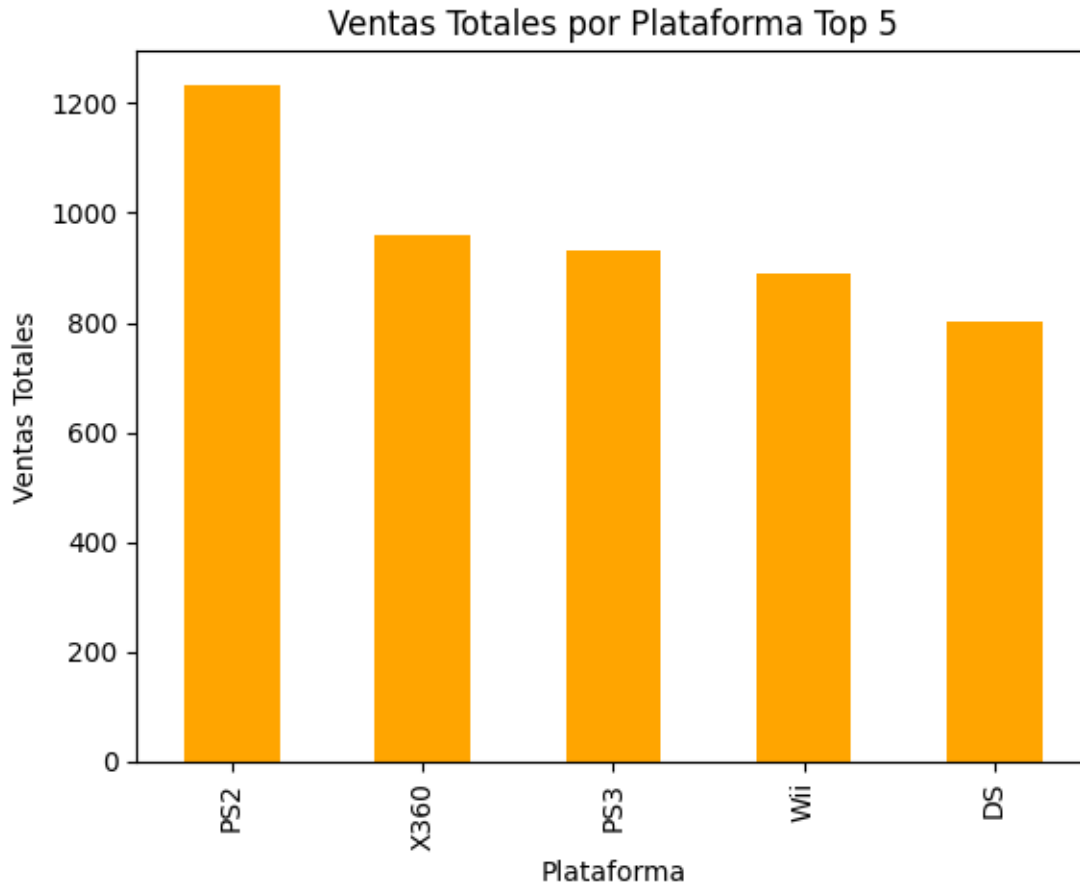
      plataformas_top_df = games[games['platform'].isin(plataformas_top)]

      print(plataformas_top)
```

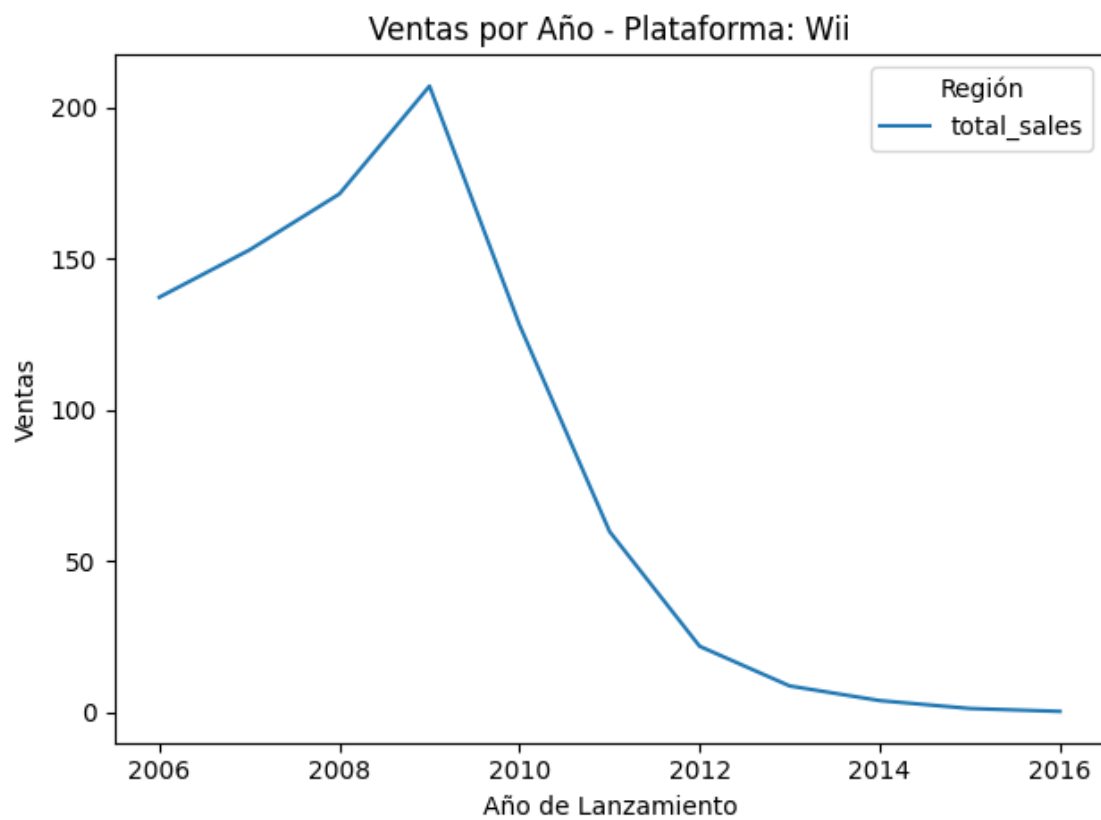
['PS2', 'X360', 'PS3', 'Wii', 'DS']

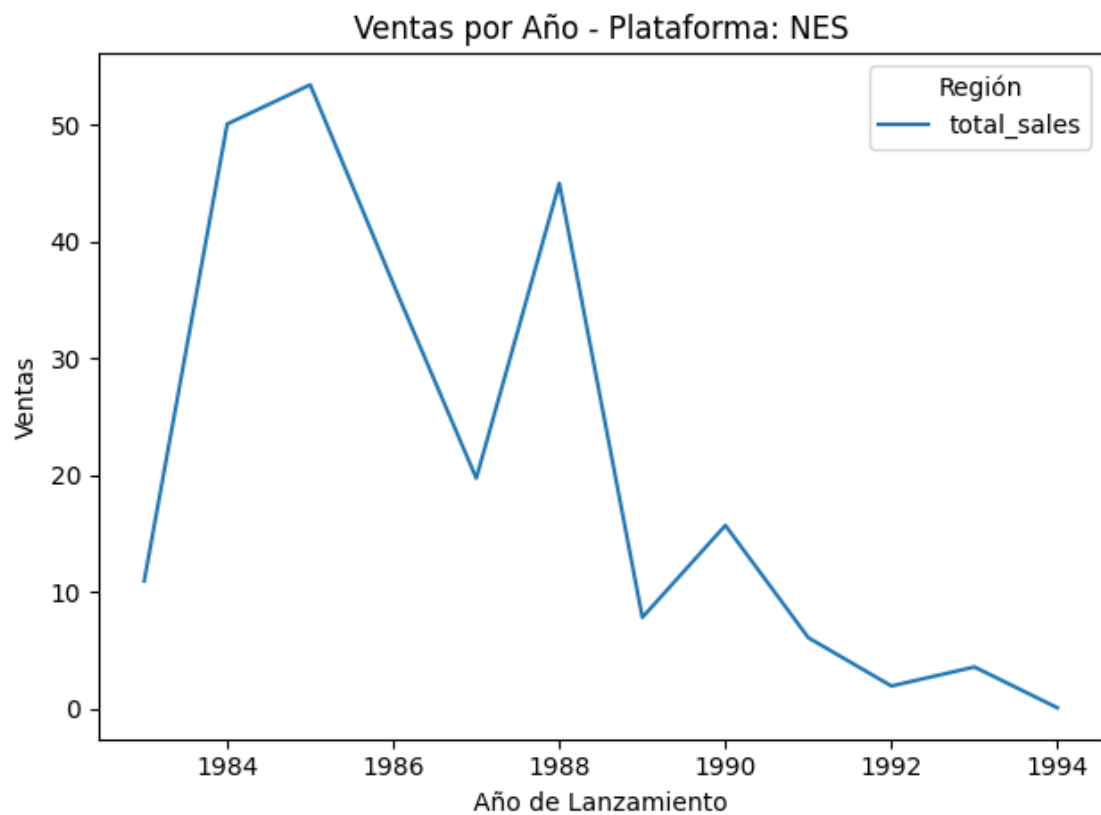
```
[40]: df_top5 = games[games['platform'].isin(plataformas_top)]
      ventas_por_plataforma = df_top5.groupby('platform')['total_sales'].sum().
      ↪sort_values(ascending=False)

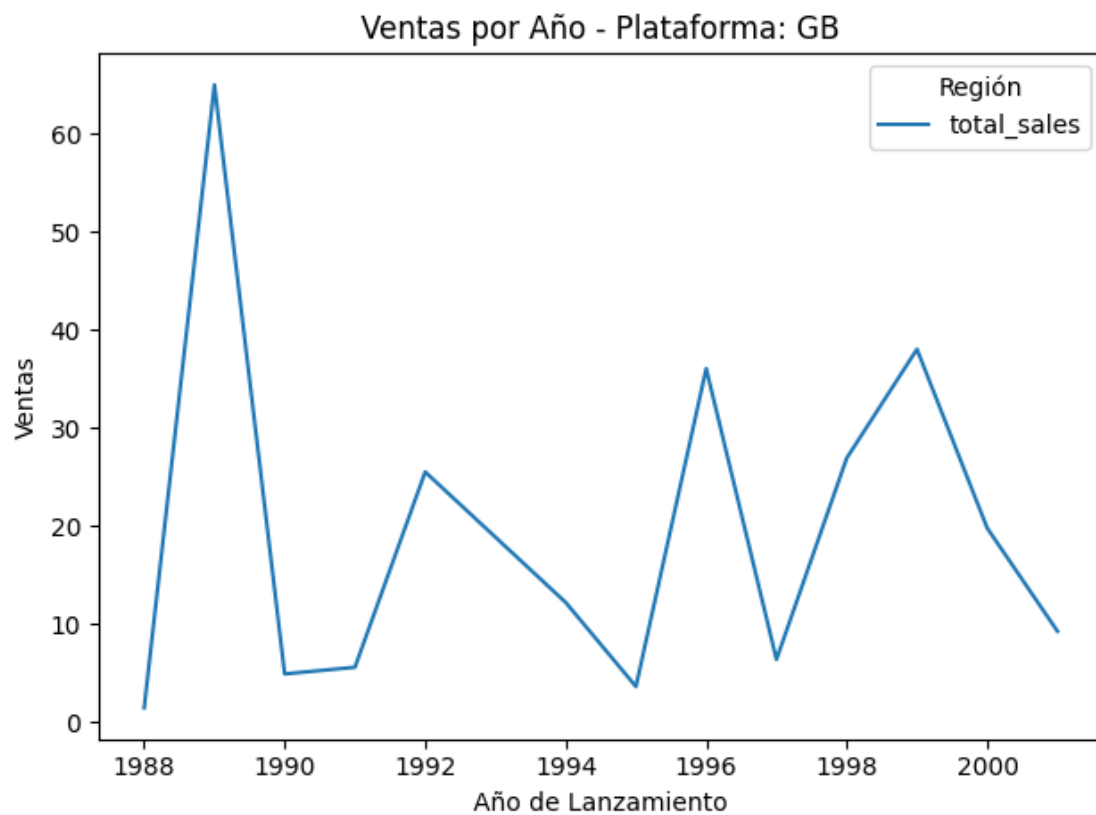
      color = "orange"
      ventas_por_plataforma.plot(kind='bar', color = color)
      plt.title('Ventas Totales por Plataforma Top 5')
      plt.xlabel('Plataforma')
      plt.ylabel('Ventas Totales')
      plt.show()
```

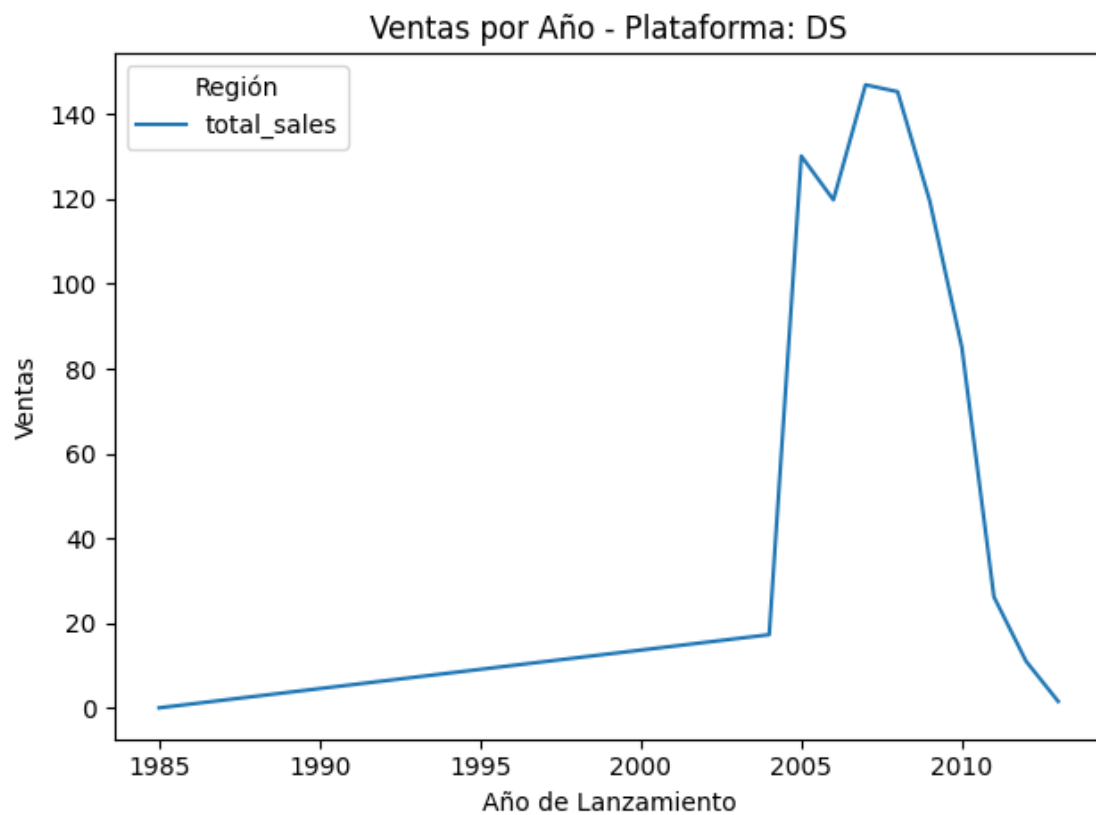


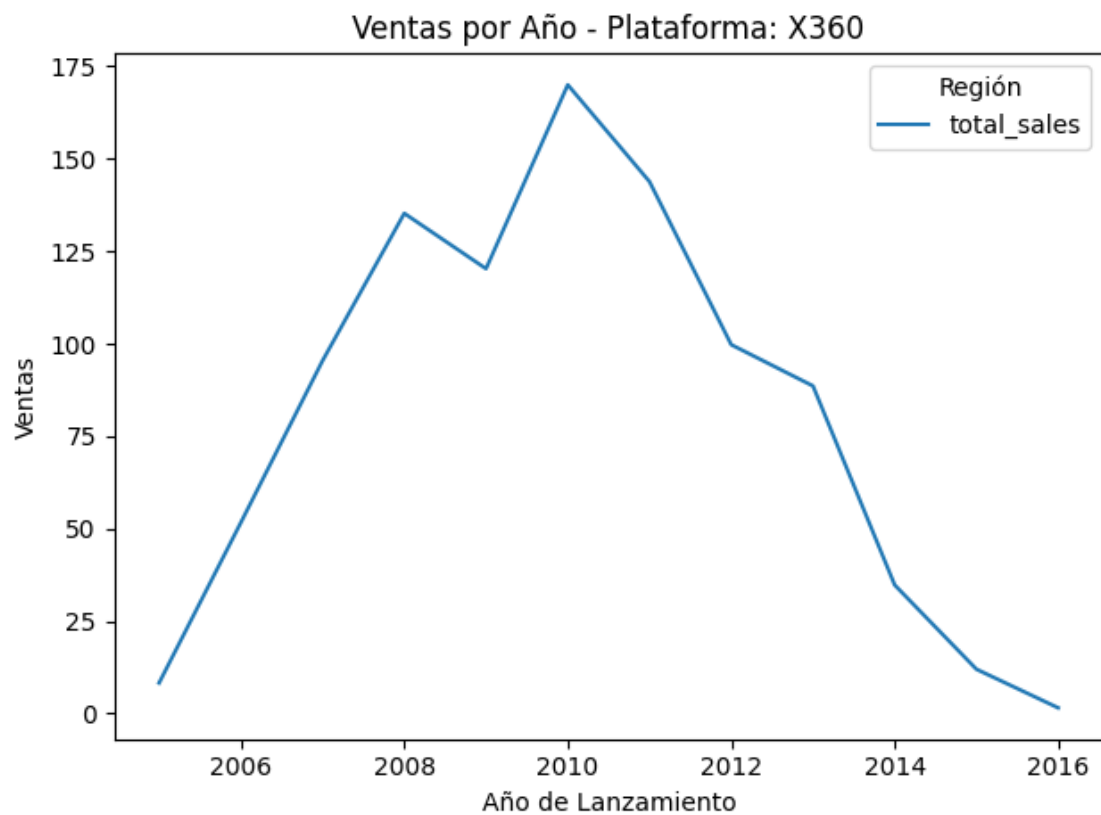
```
[15]: for plataforma in games['platform'].unique():
        plataforma_df = games[games['platform'] == plataforma]
        ventas_por_ano = plataforma_df.groupby('year_of_release')['total_sales'].
        ↪sum()
        ventas_por_ano.plot(kind='line')
        plt.title(f'Ventas por Año - Plataforma: {plataforma}') # Ajusta el tamaño
        ↪de la fuente
        plt.xlabel('Año de Lanzamiento')
        plt.ylabel('Ventas')
        plt.legend(title='Región')
        plt.tight_layout()
        plt.show()
```

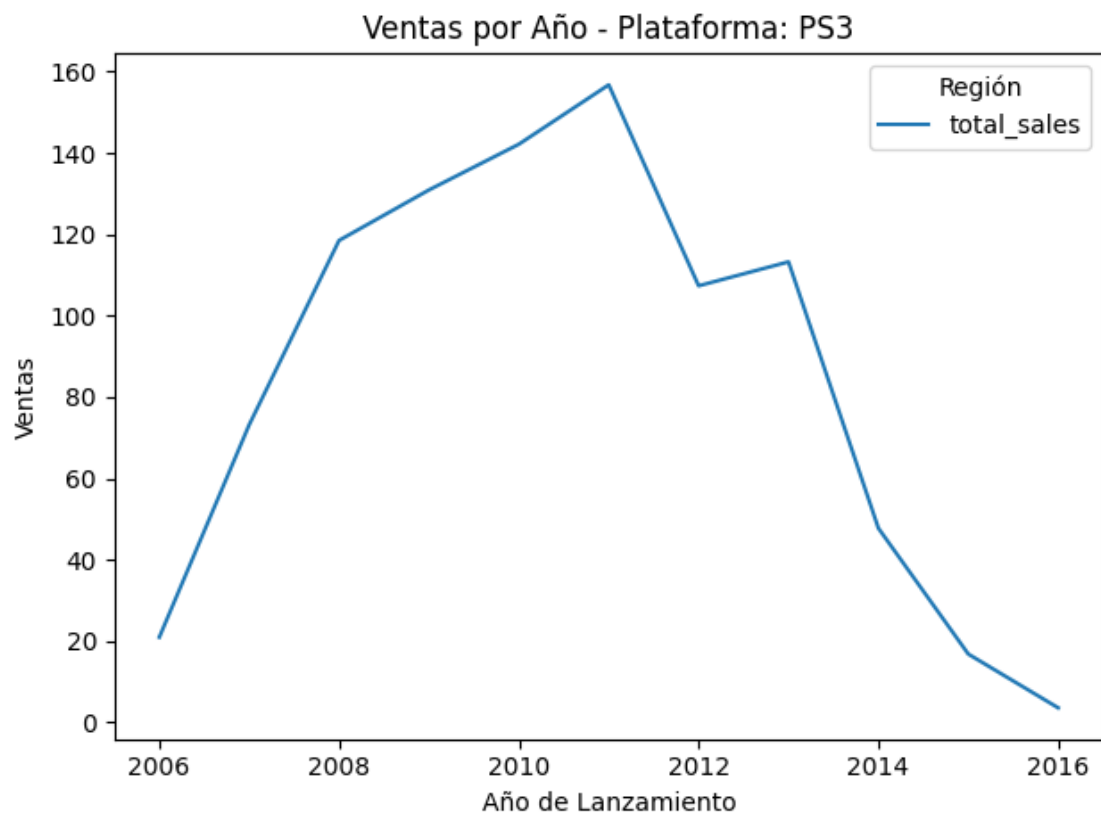


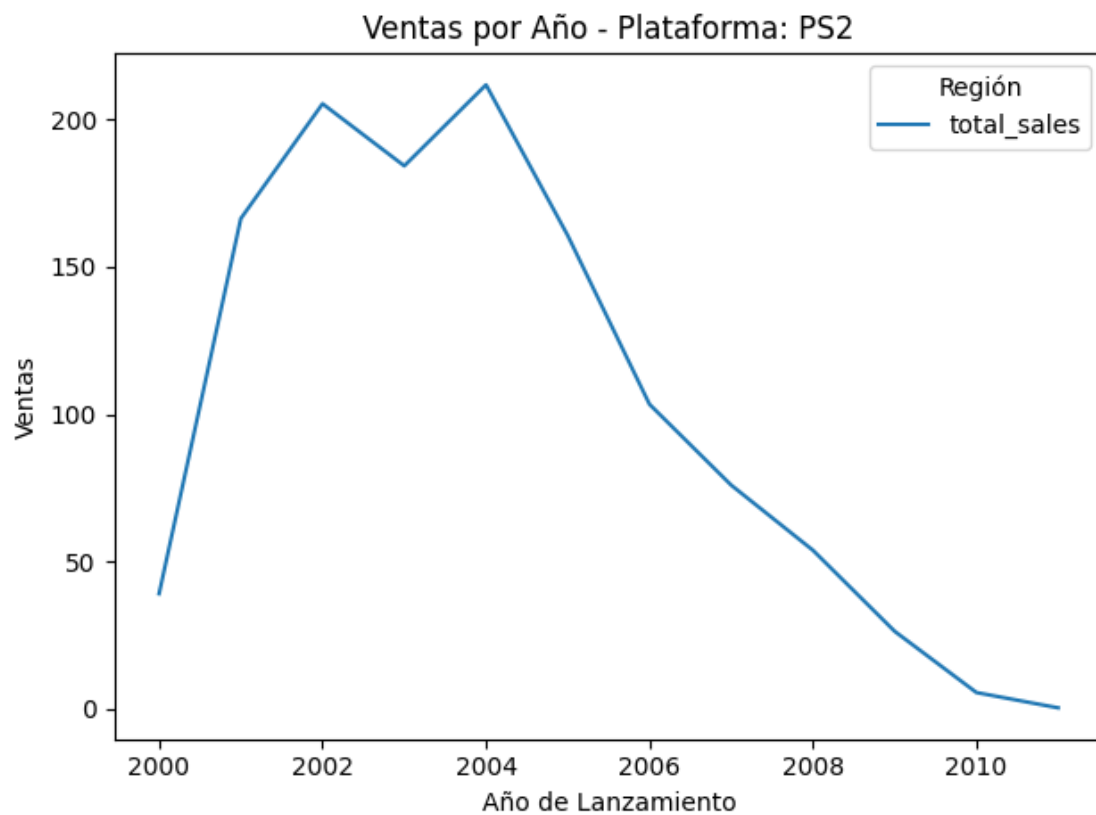


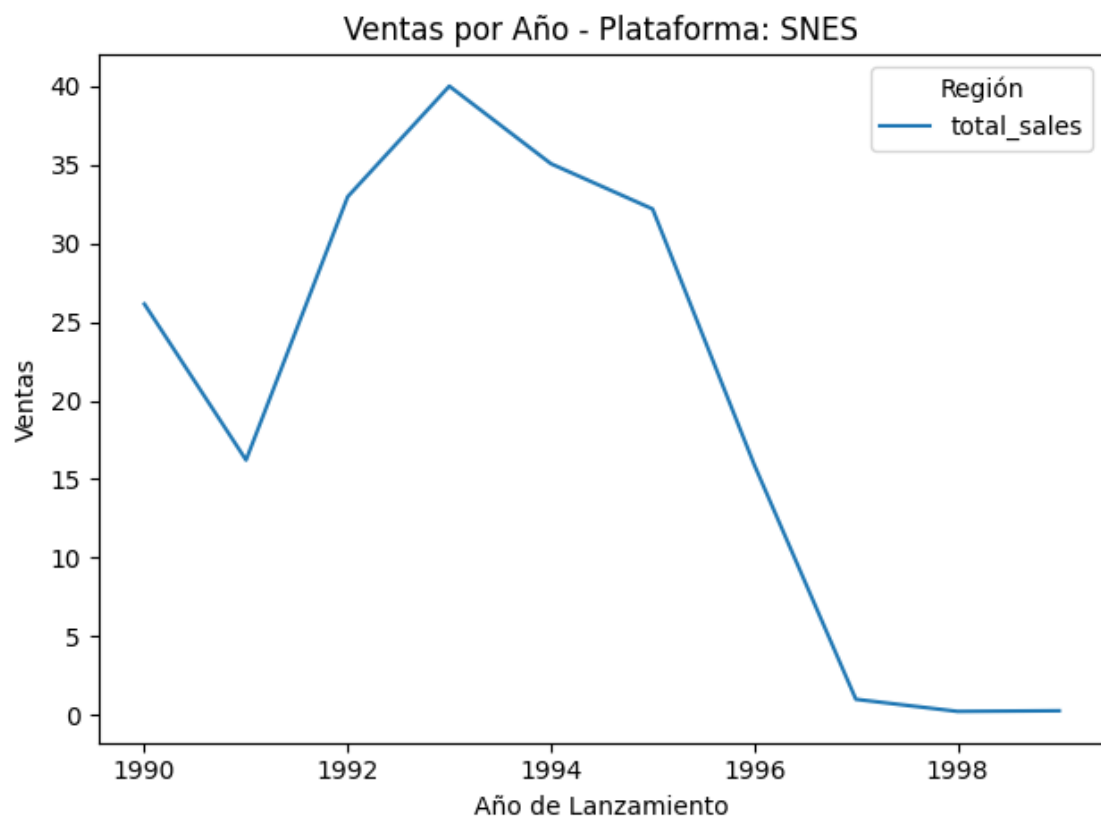


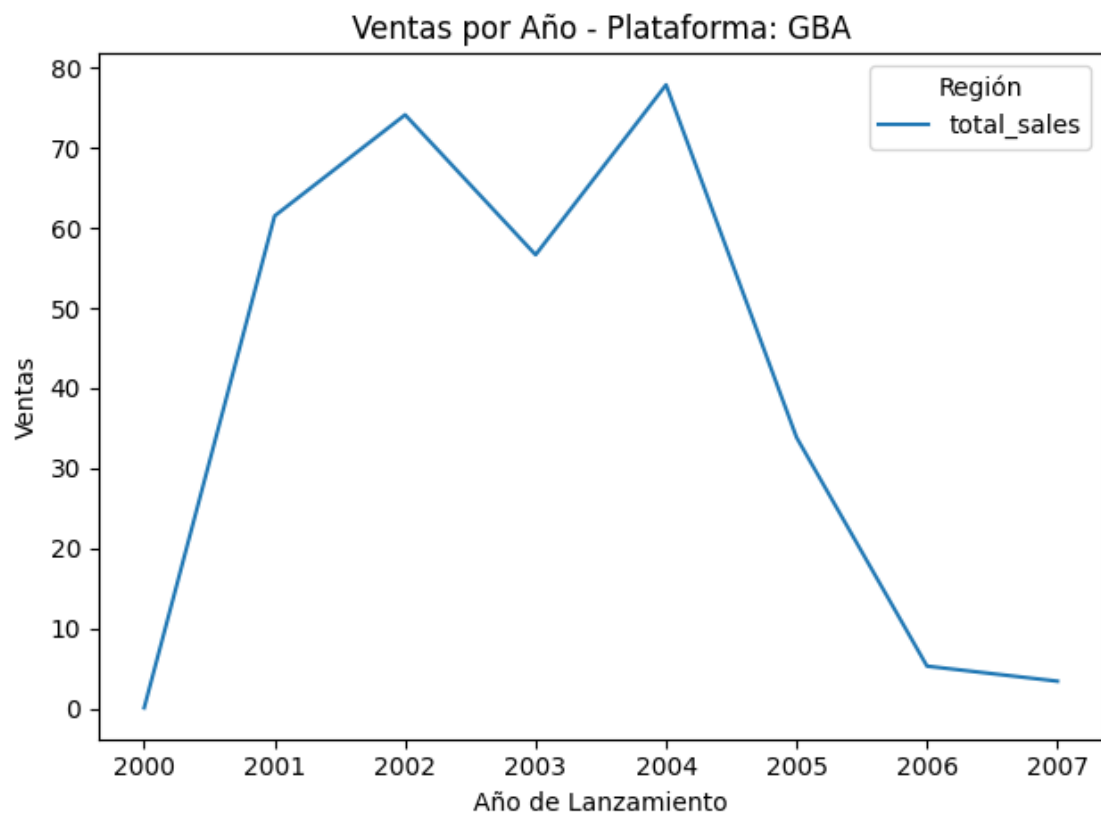


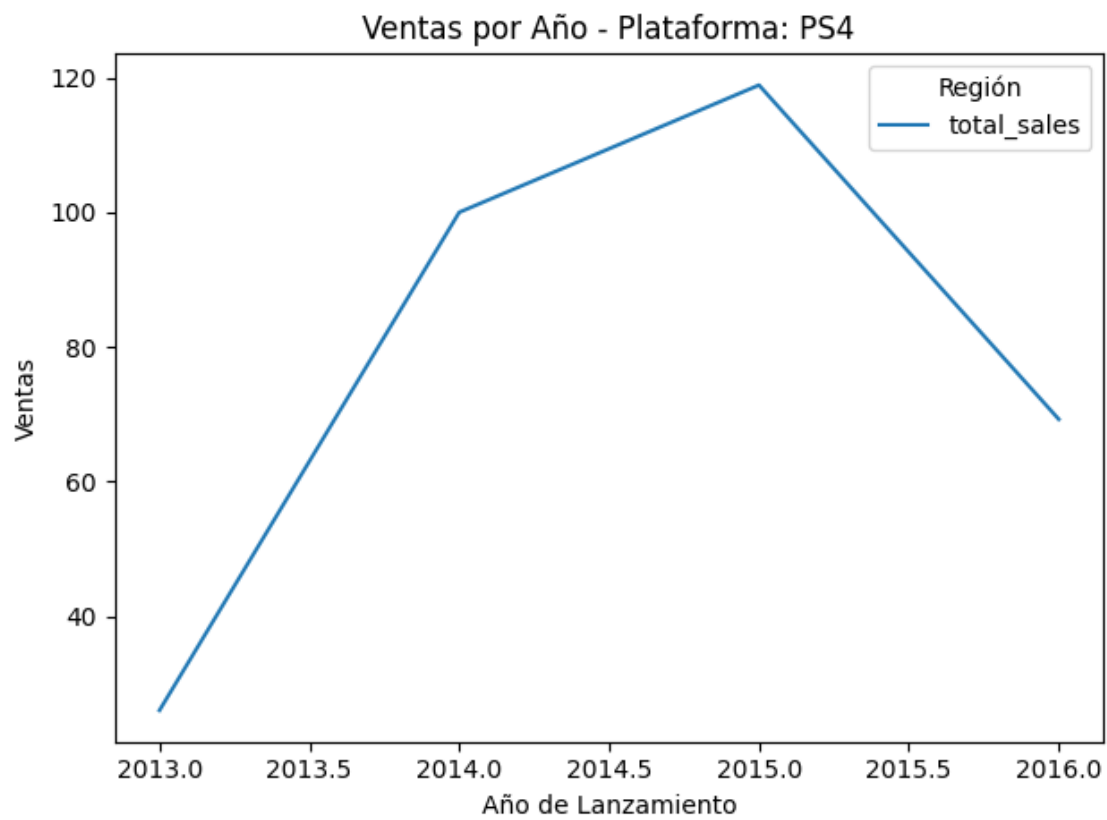


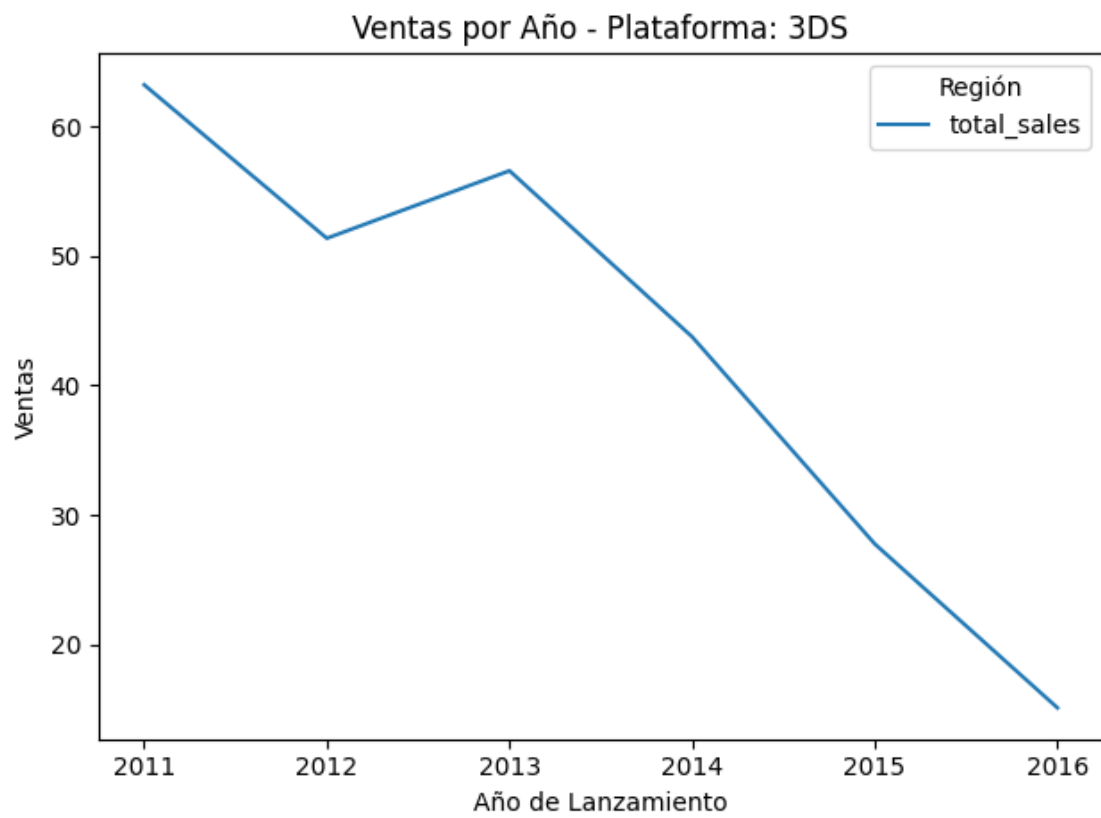


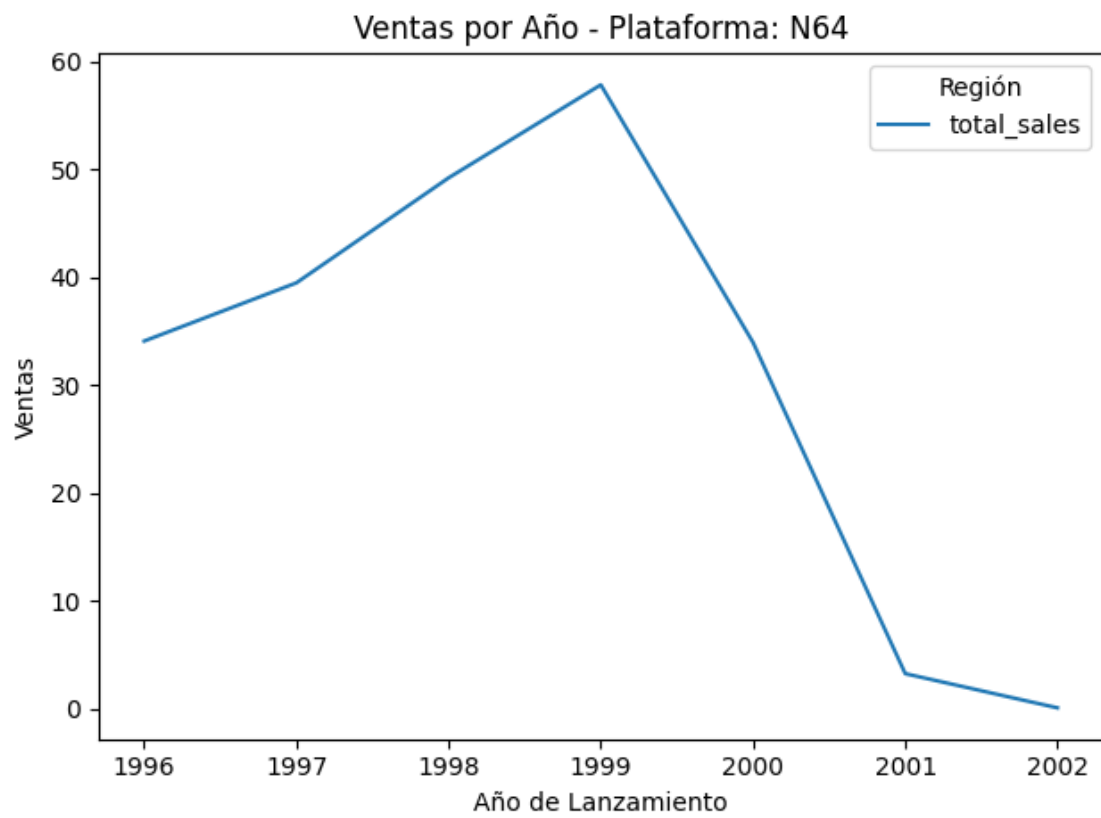


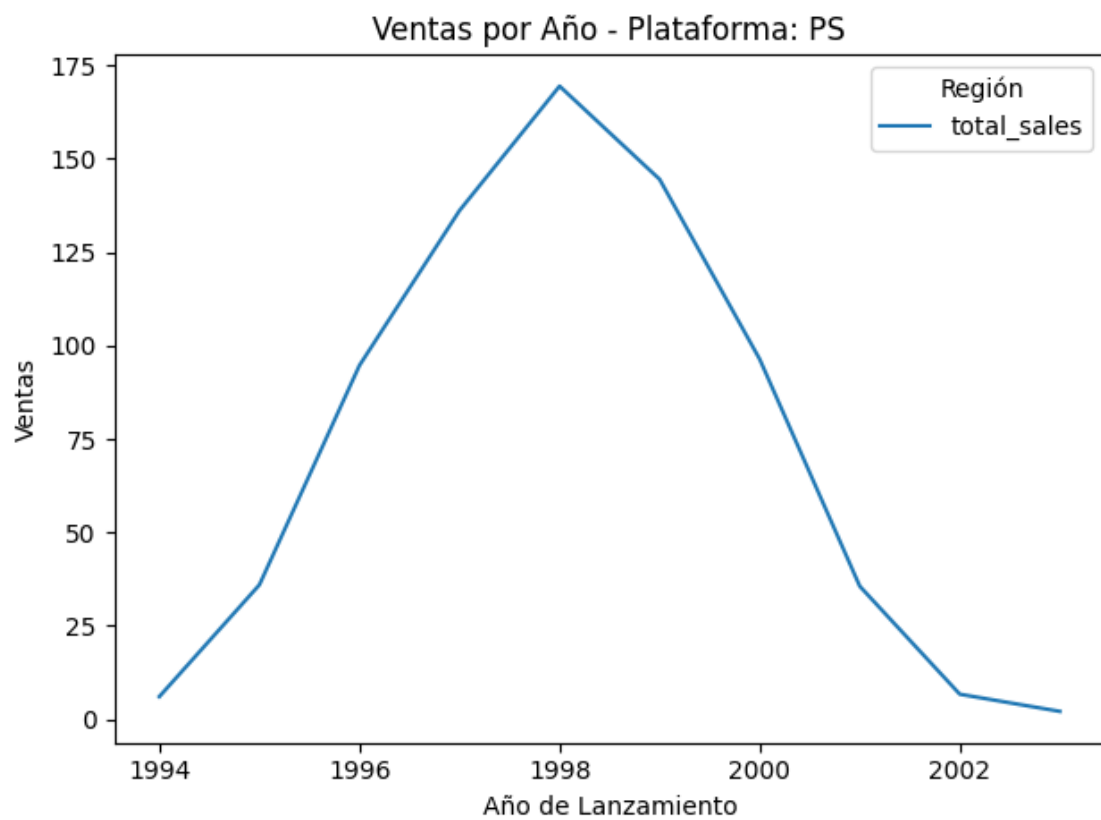


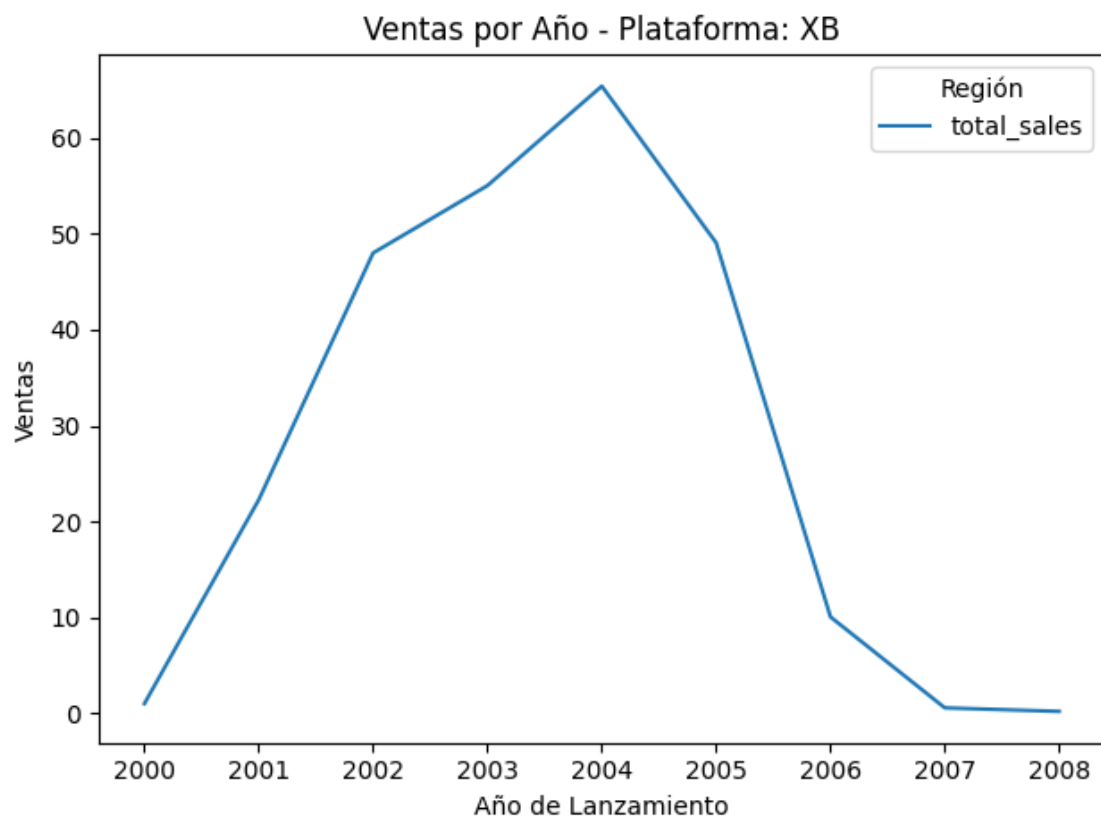


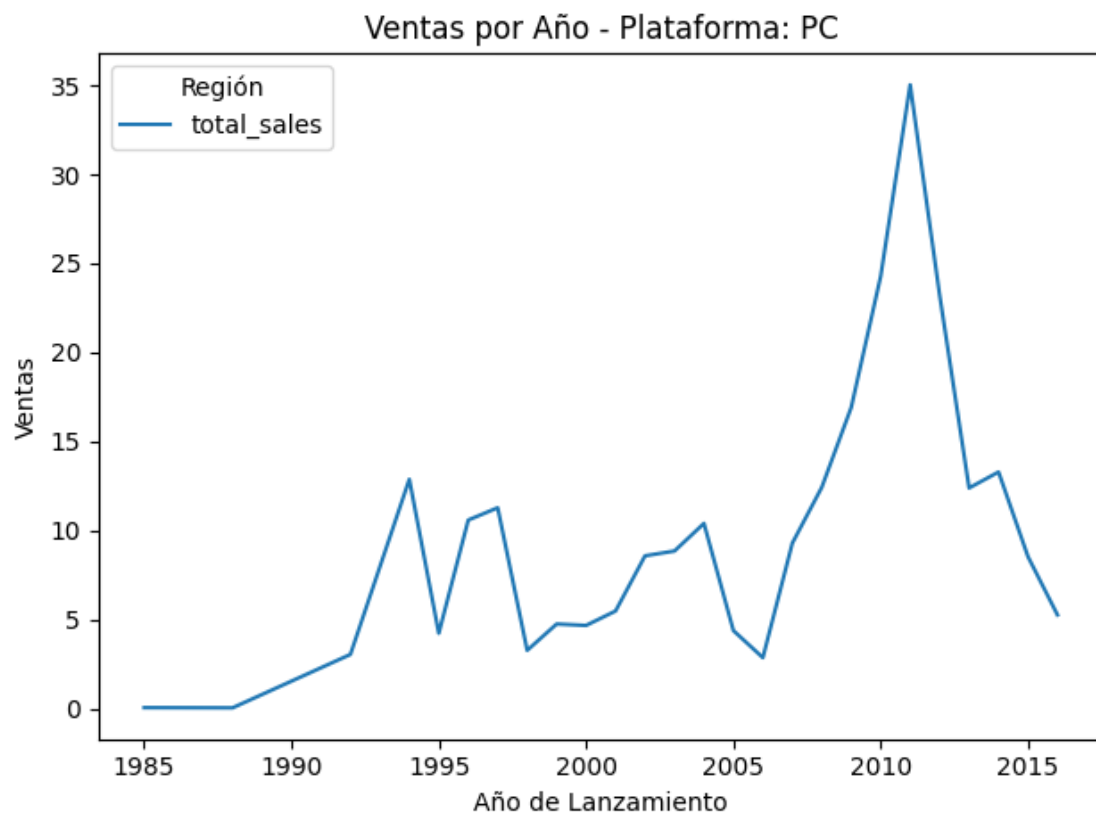


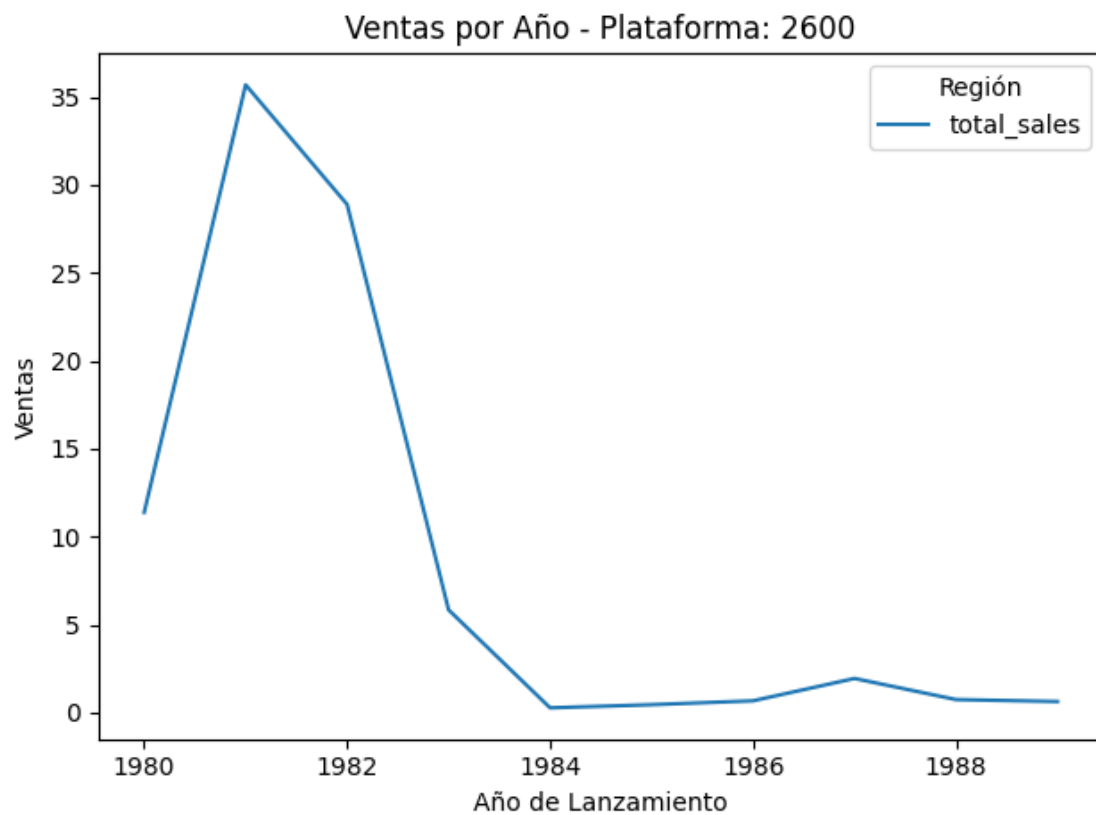


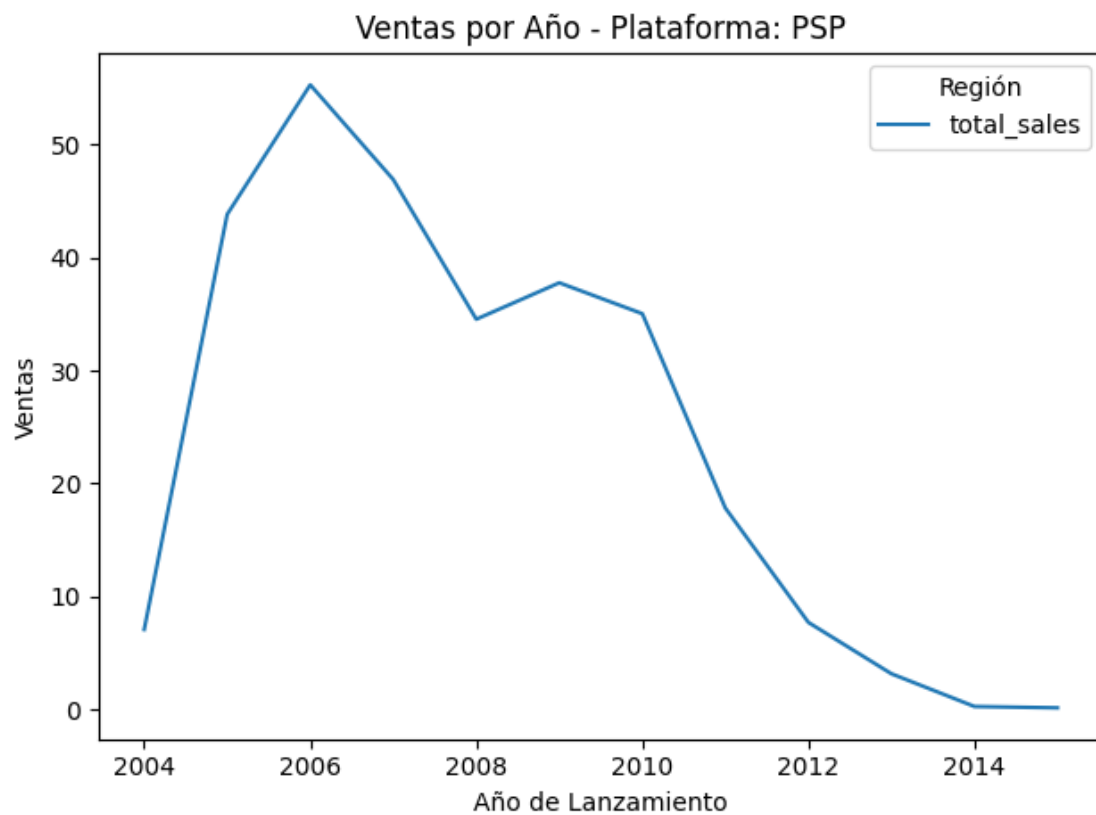


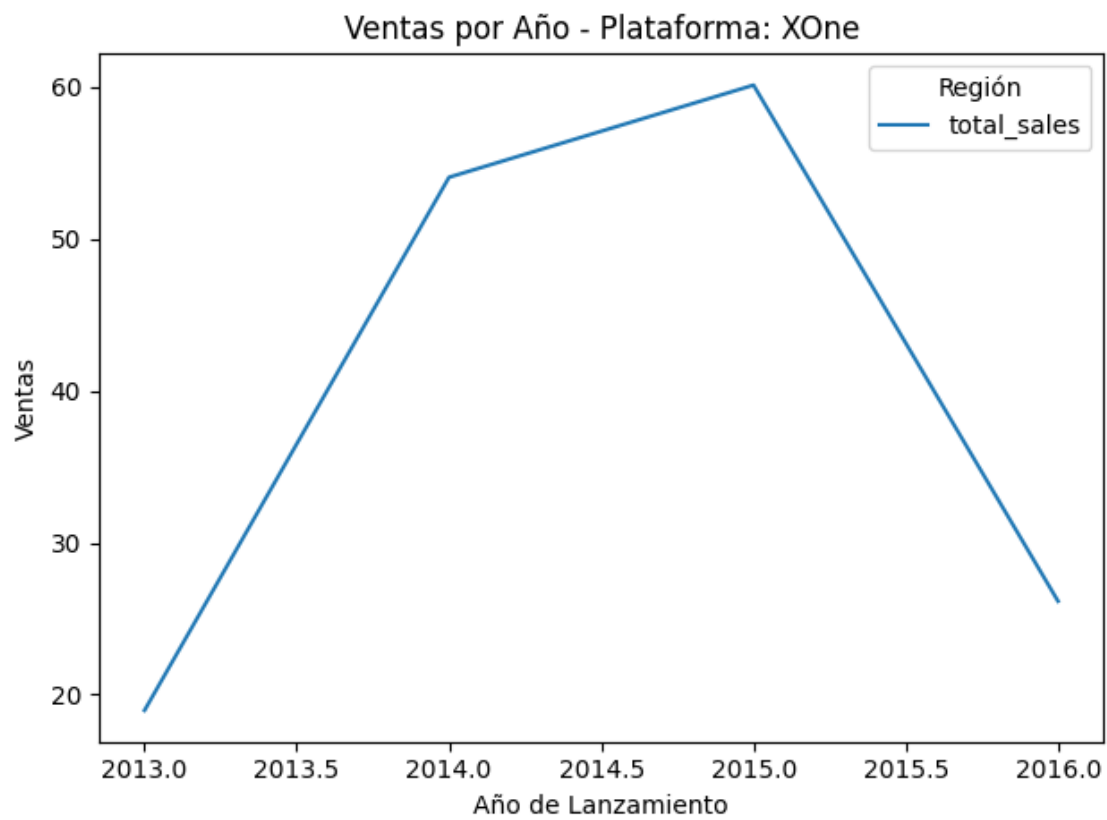


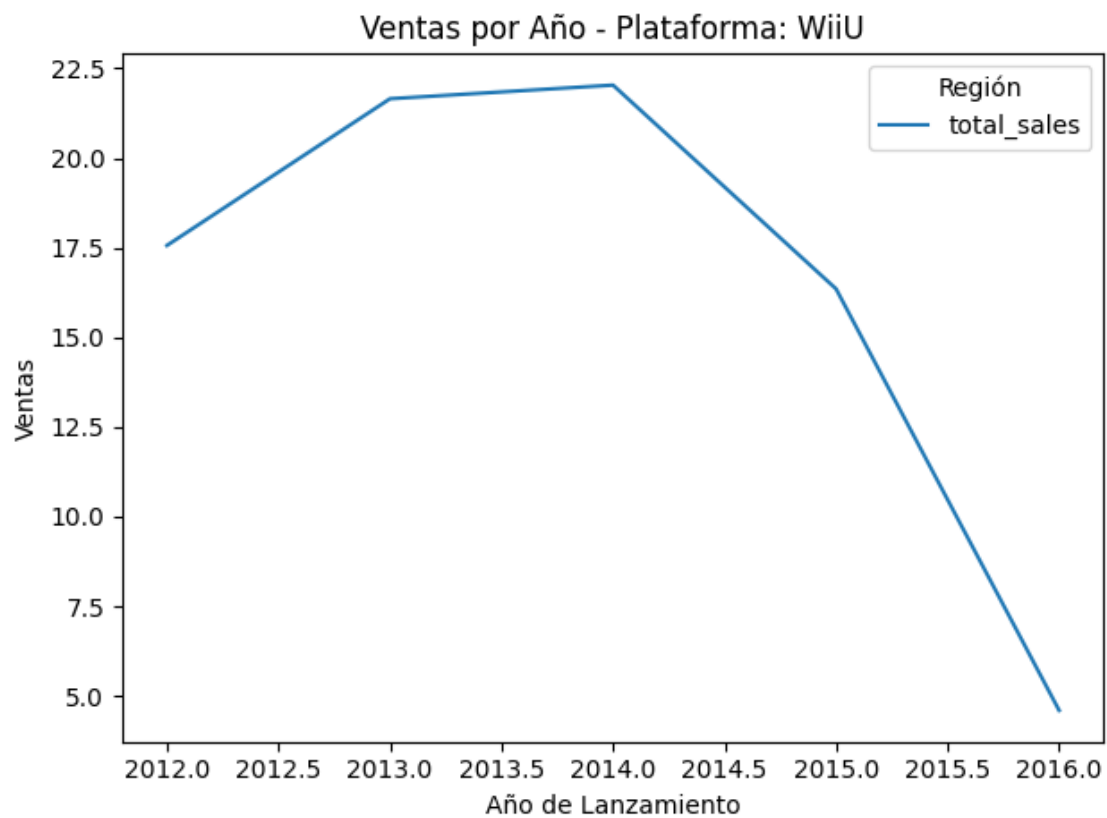


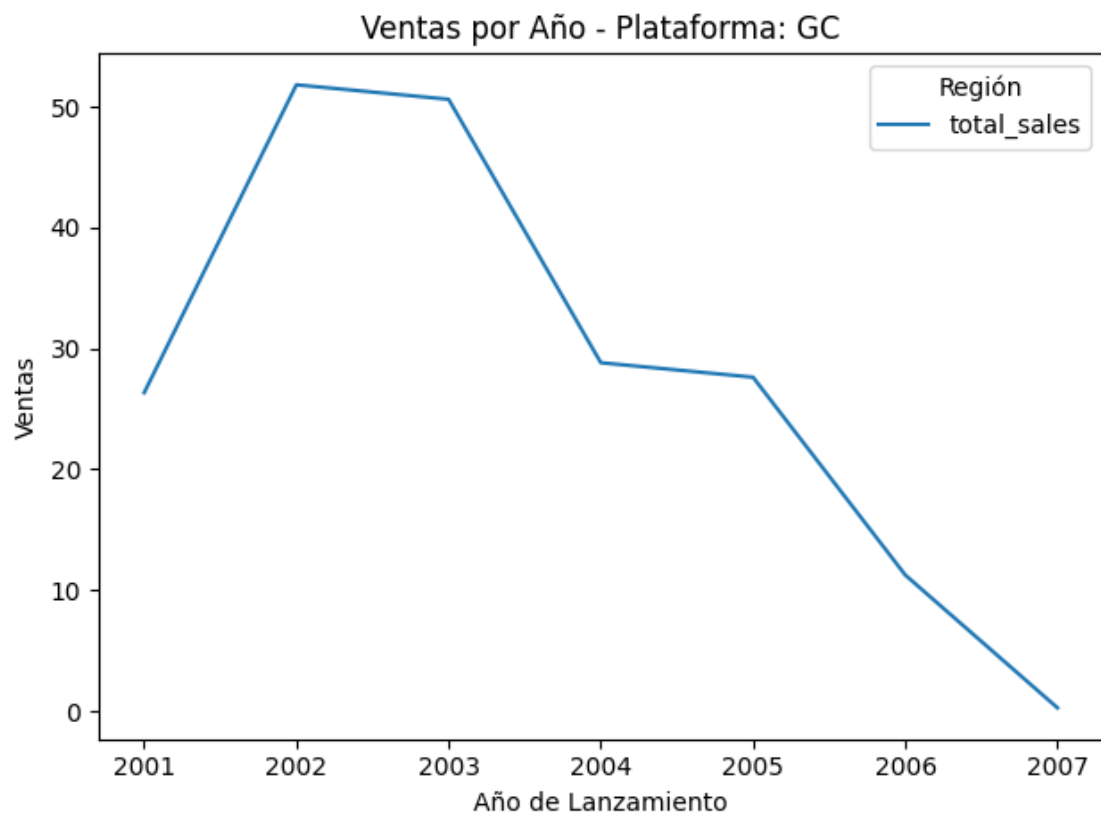


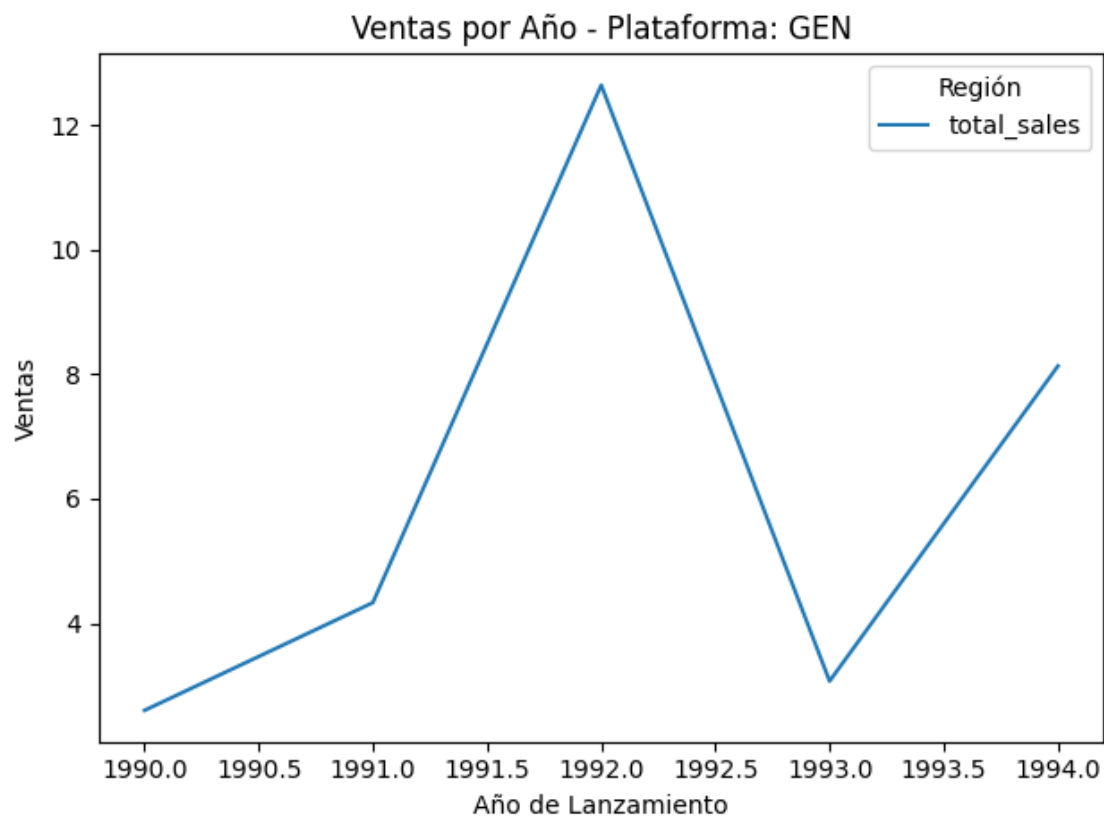


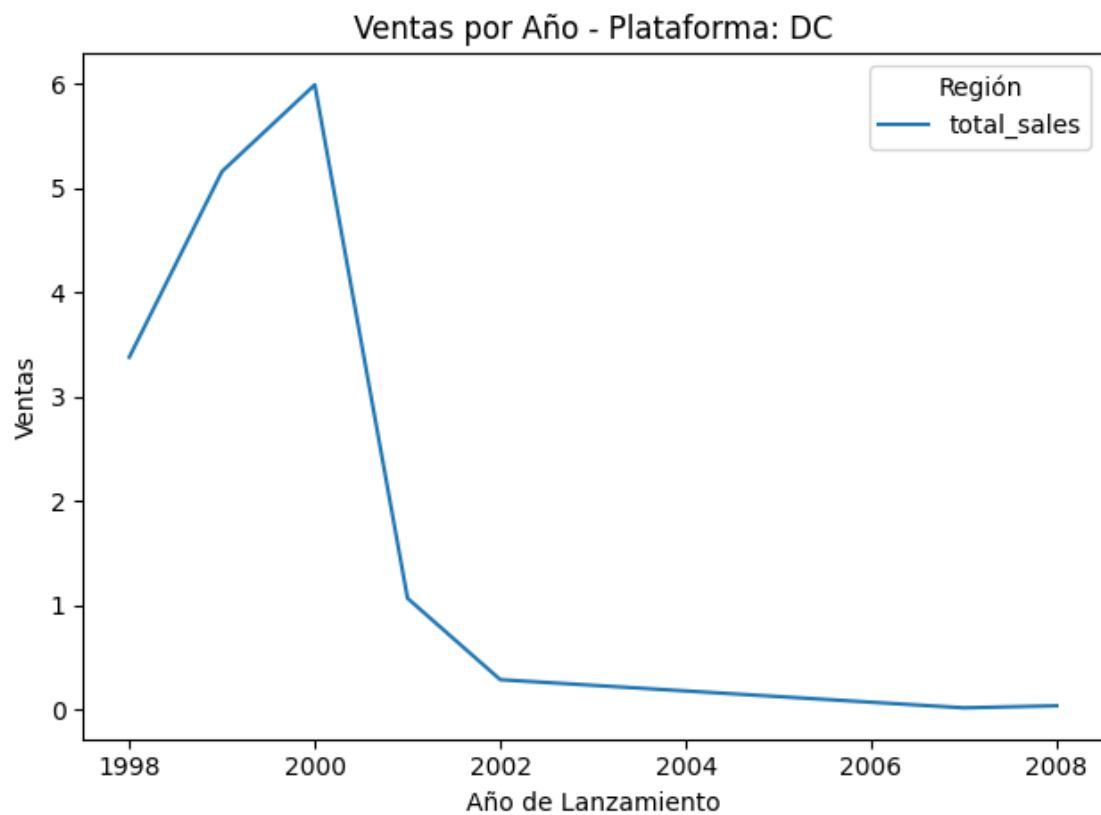


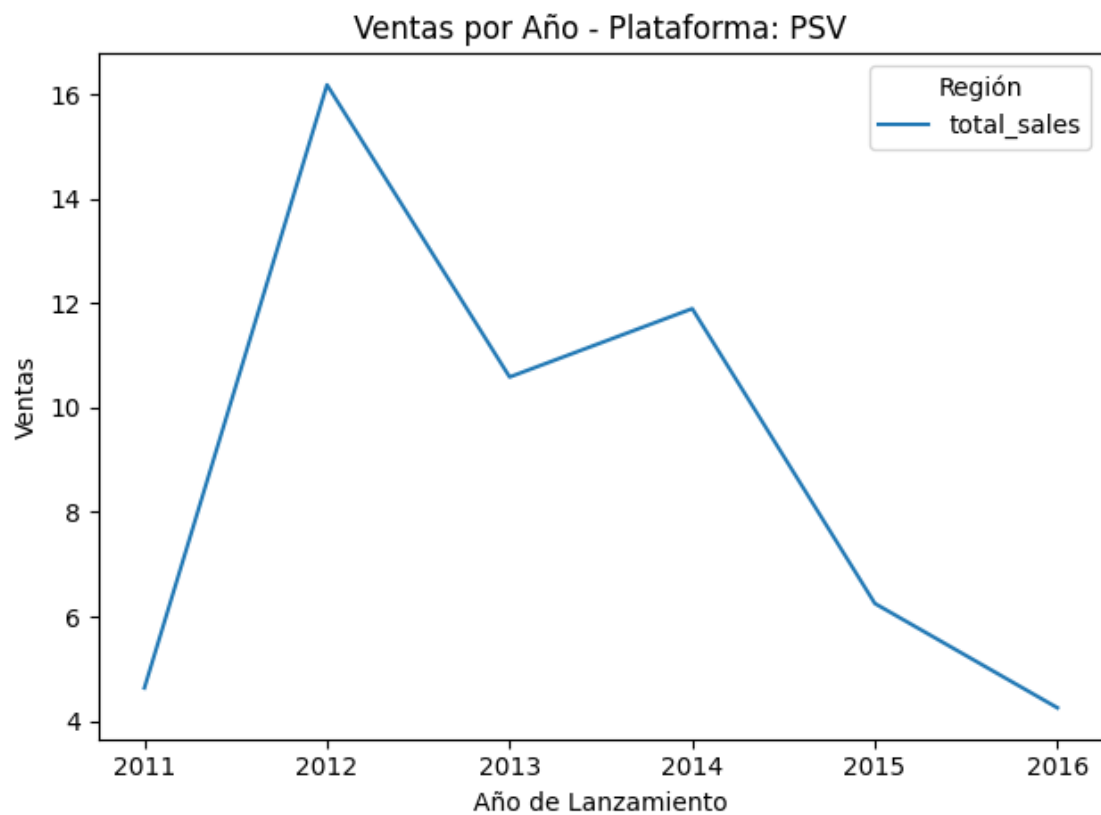


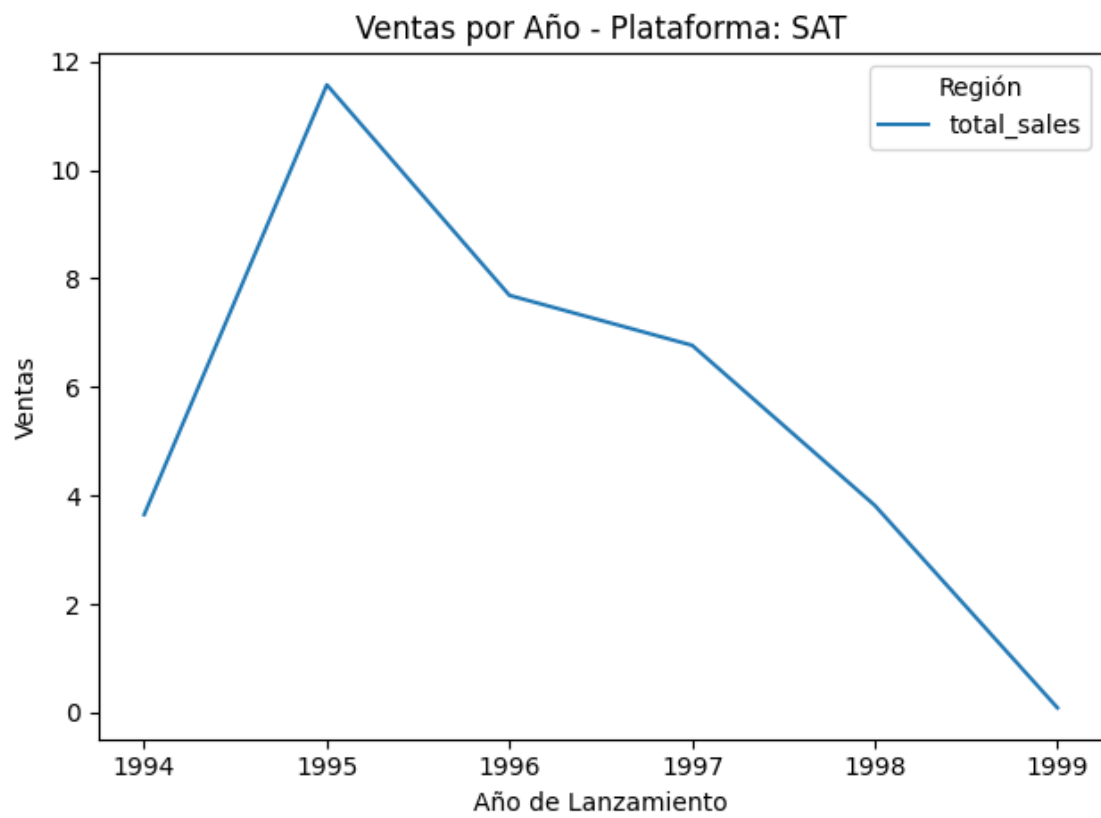


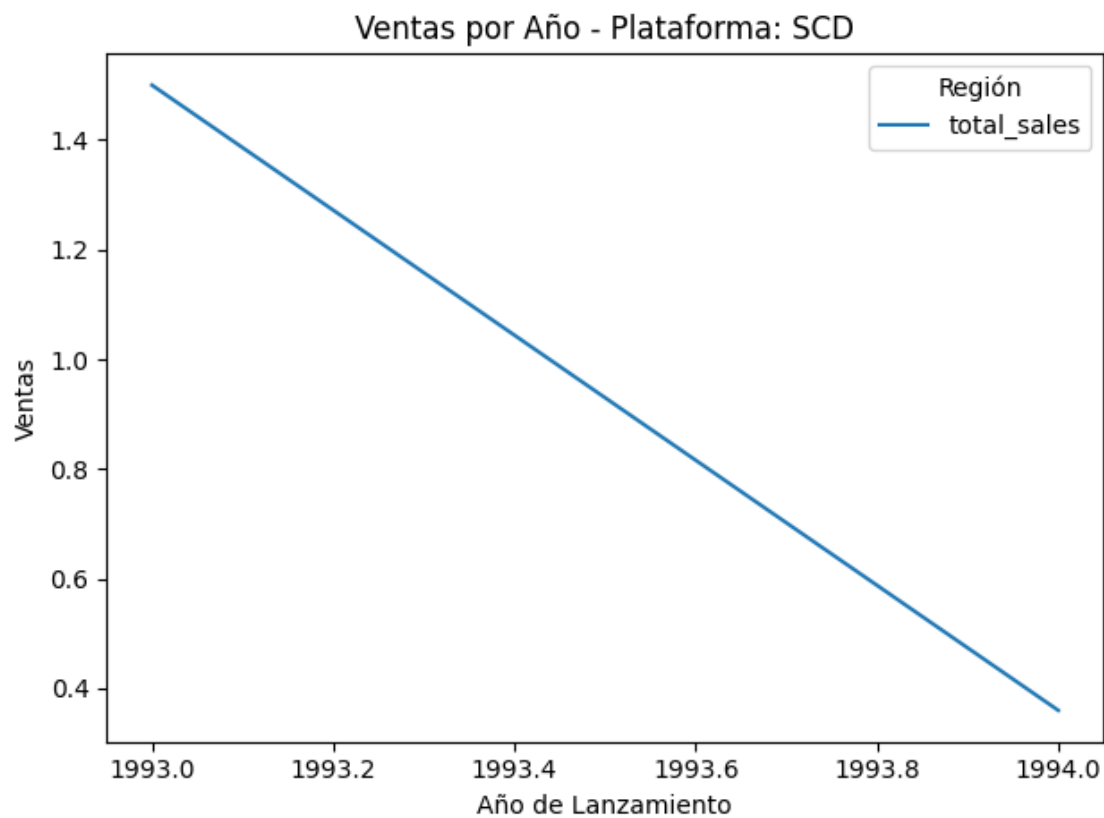


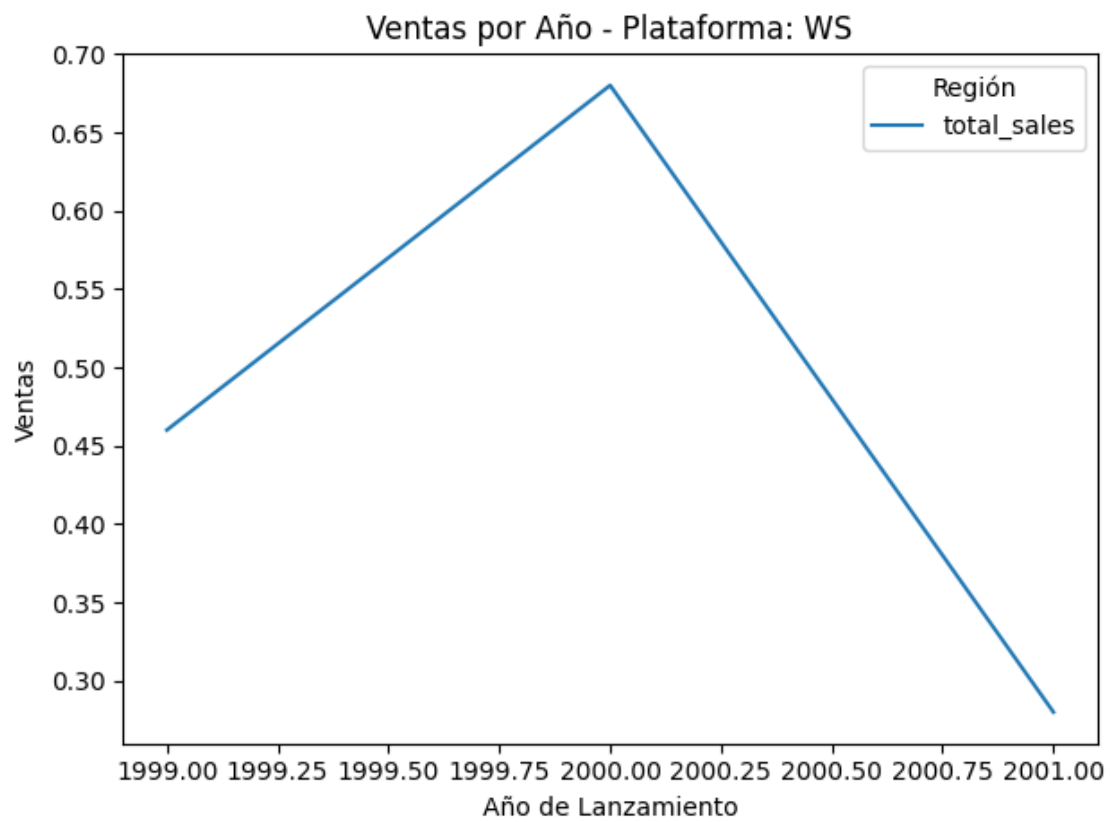


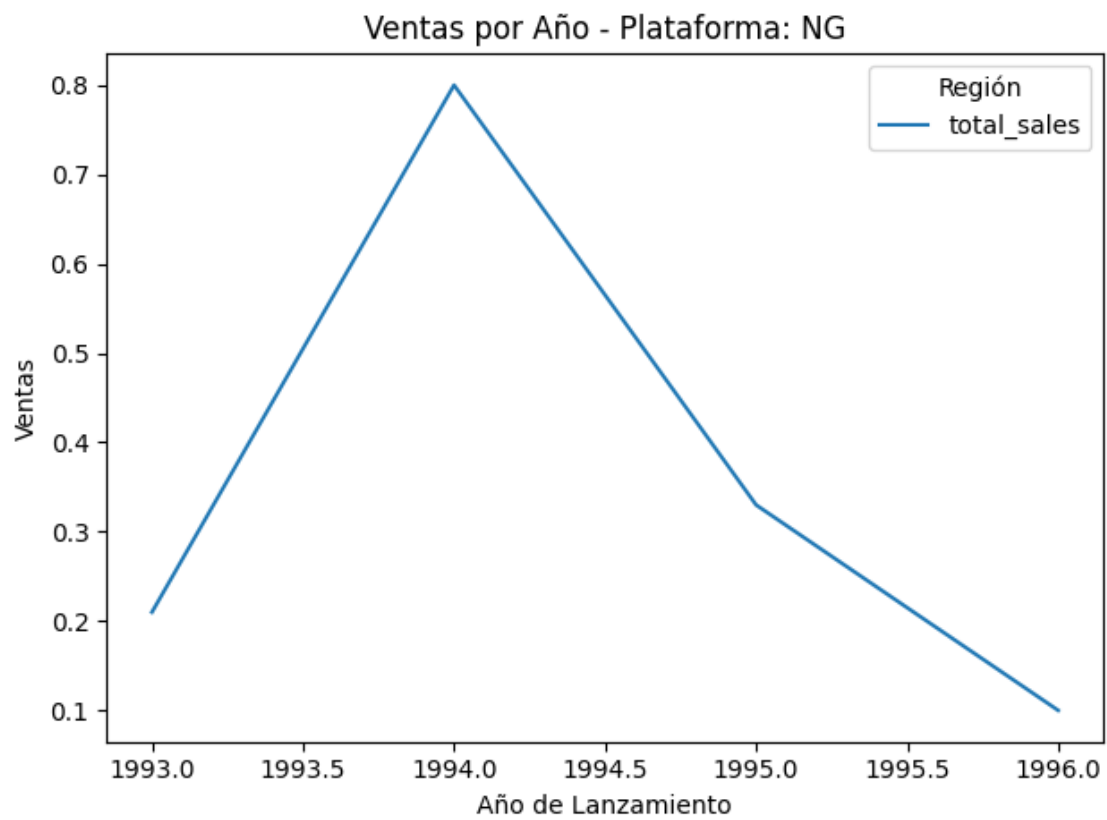


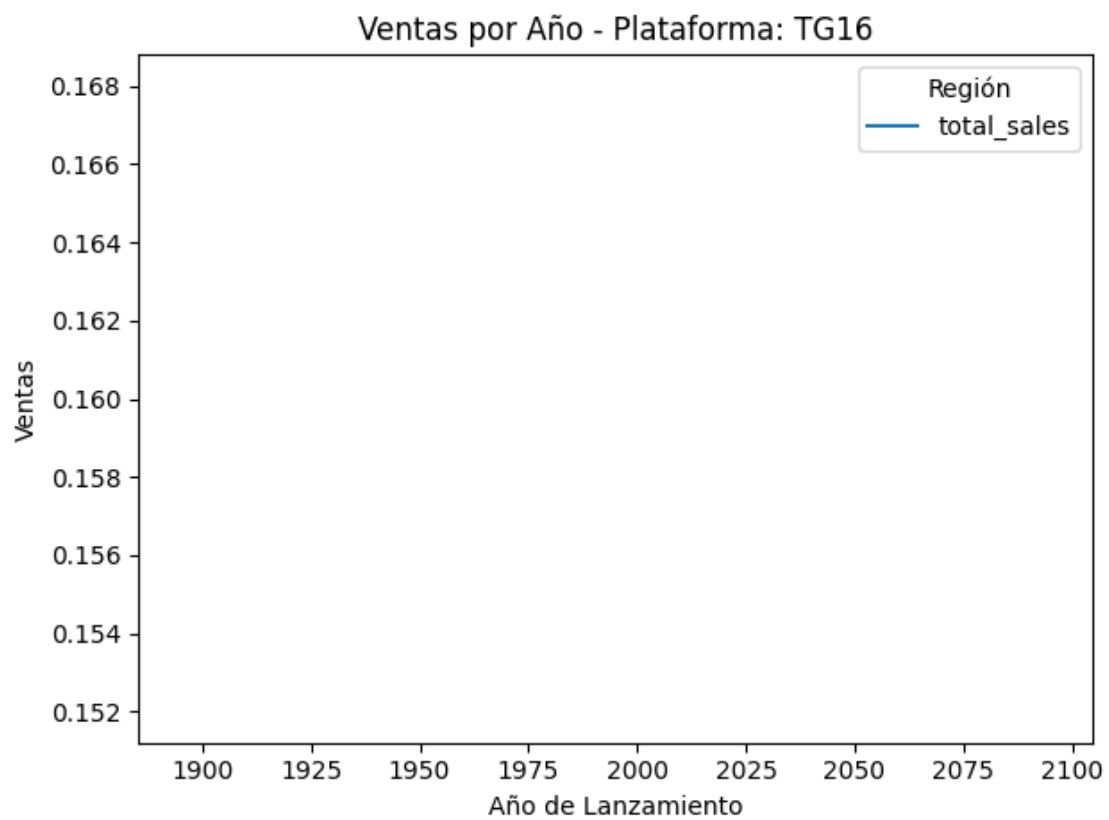


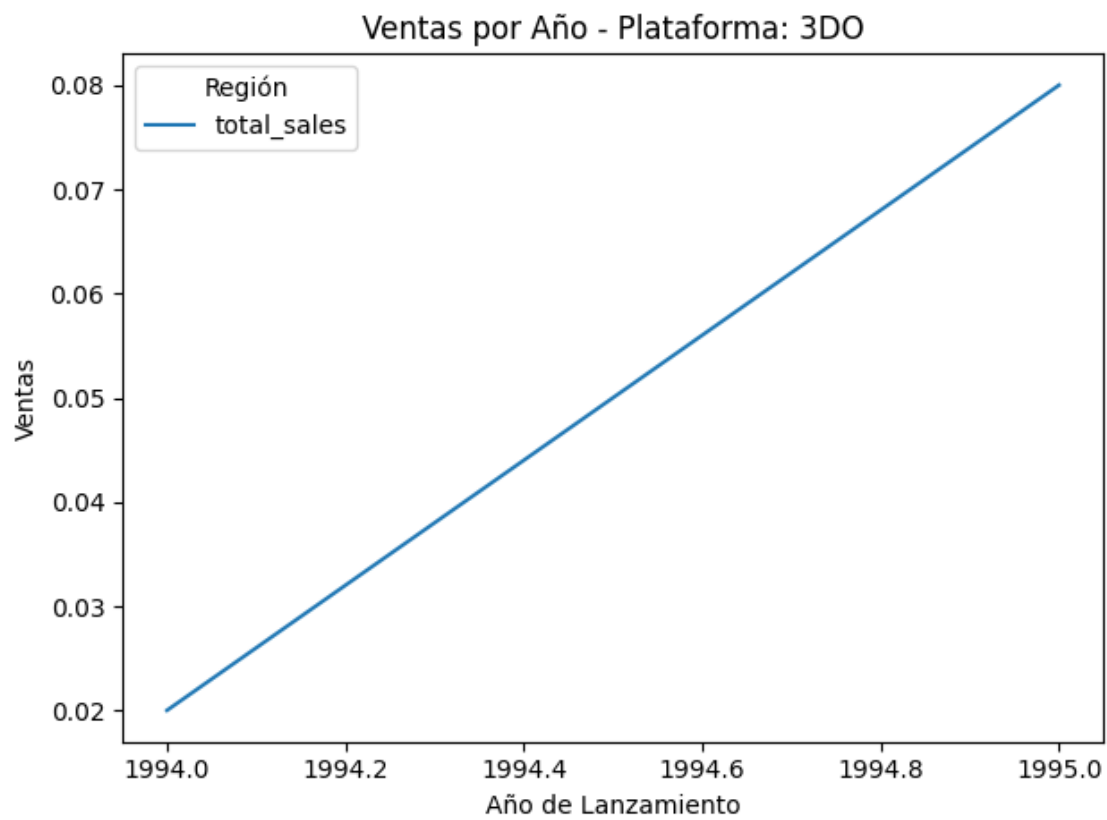


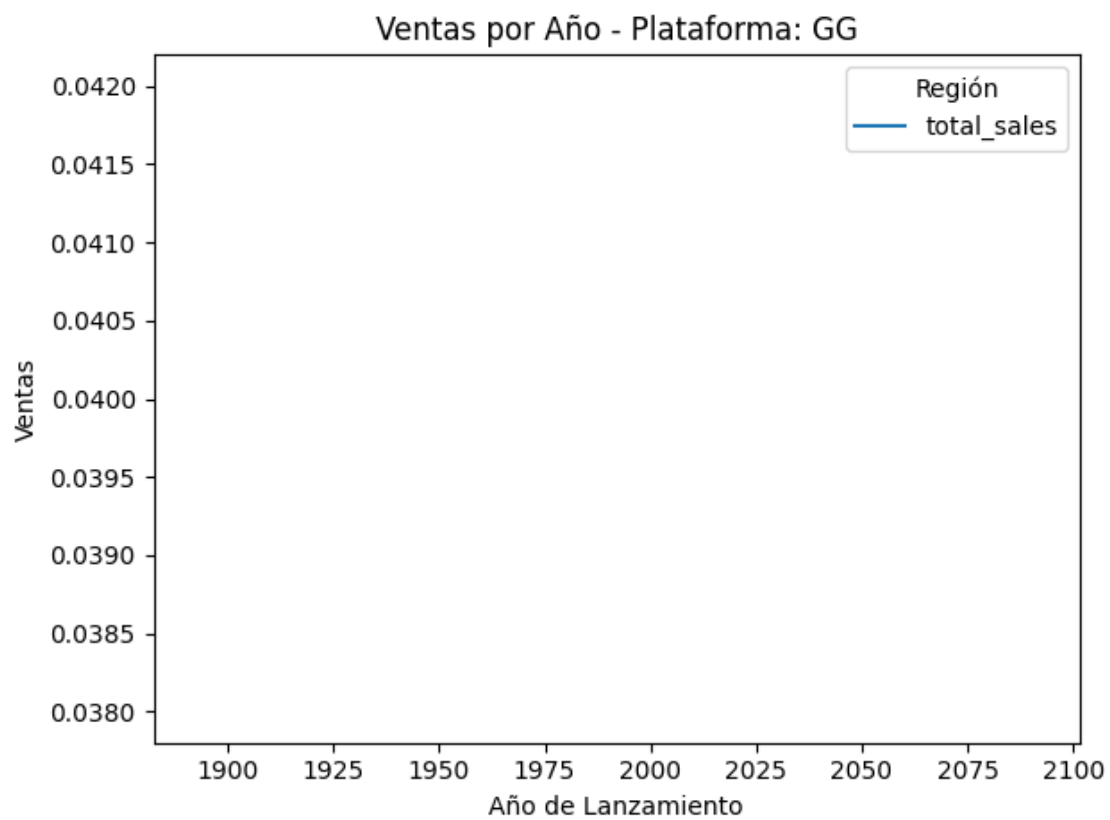


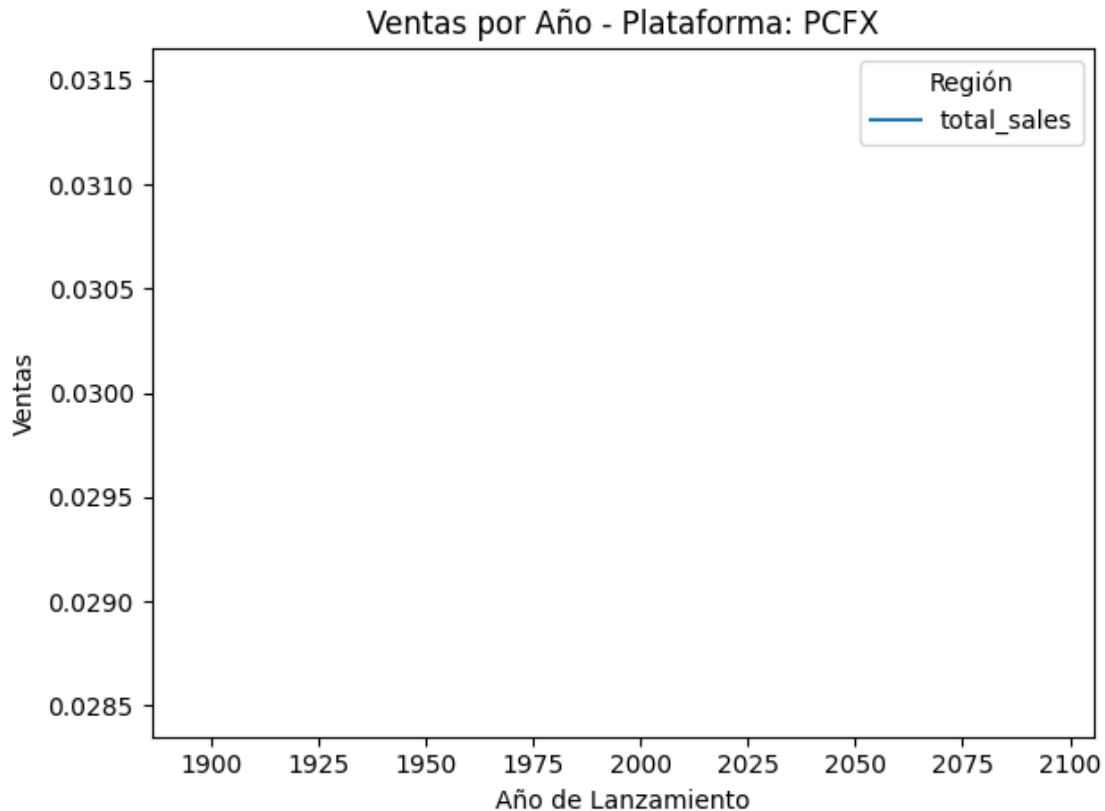












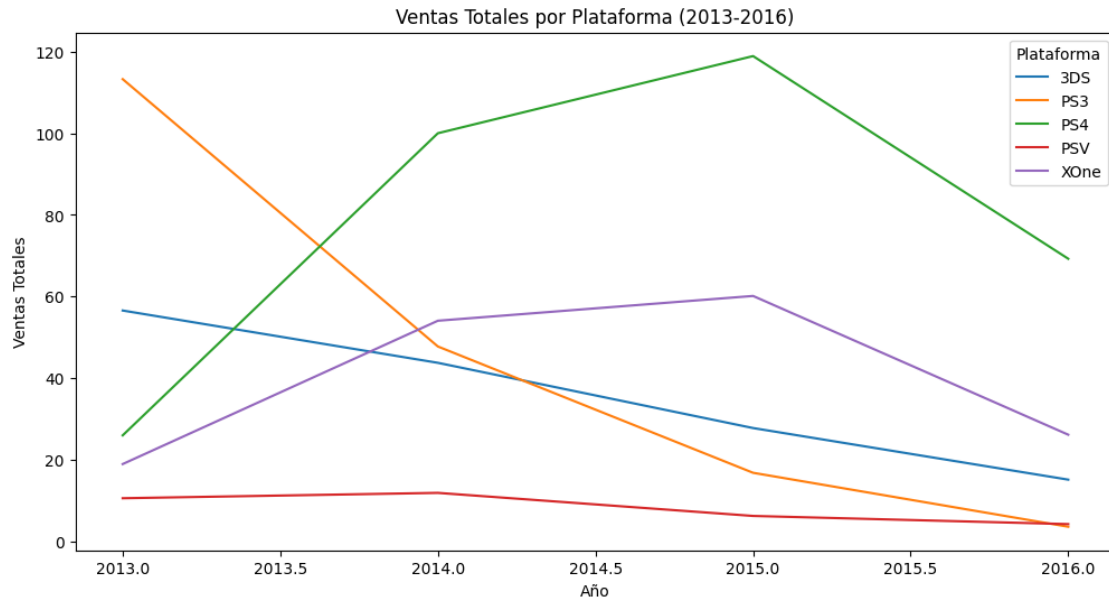
```
[44]: games_filtro = games[(games['year_of_release'] >= 2013) &
    ↪ (games['year_of_release'] <= 2016)]

top_consolas = games_filtro['platform'].value_counts().head(5).index

games_filtro = games_filtro[games_filtro['platform'].isin(top_consolas)]

consolas_ventas = games_filtro.groupby(['platform',
    ↪ 'year_of_release'])['total_sales'].sum().reset_index()

plt.figure(figsize=(12, 6))
sns.lineplot(x='year_of_release', y='total_sales', hue='platform',
    ↪ data=consolas_ventas)
plt.title('Ventas Totales por Plataforma (2013-2016)')
plt.xlabel('Año')
plt.ylabel('Ventas Totales')
plt.legend(title='Plataforma')
plt.show()
```



Podemos observar un patron en las graficas, de picos altos y a su vez descensos bruscos, la causa de estos descensos pueden ocurrir por el estreno de nuevas consolas, por ejemplo las PS, PS2, PS4, claramente su descenso ocurre cuando sale a la venta una nueva consola. Entonces las consolas que fueron creadas antes del 2000 al haber nuevas graficas, con mejor tecnologia, nuevos juegos tienden a dejar de venderse.

```
[16]: plataformas_rentables = games.groupby('platform').agg({'total_sales': 'sum',
    ↪ 'critic_score': 'mean'}).sort_values(by=['total_sales', 'critic_score'],
    ↪ ascending=False)
print(plataformas_rentables)
```

platform	total_sales	critic_score
PS2	1233.56	41.189939
X360	961.24	49.841721
PS3	931.34	43.382083
Wii	891.18	27.727838
DS	802.78	21.272984
PS	727.58	11.790756
PS4	314.14	46.344388
GBA	312.88	35.689273
PSP	289.53	25.813915
3DS	257.81	21.773438
PC	255.76	55.457680
GB	254.43	0.000000
XB	251.57	61.466999
NES	251.05	0.000000
N64	218.01	0.000000

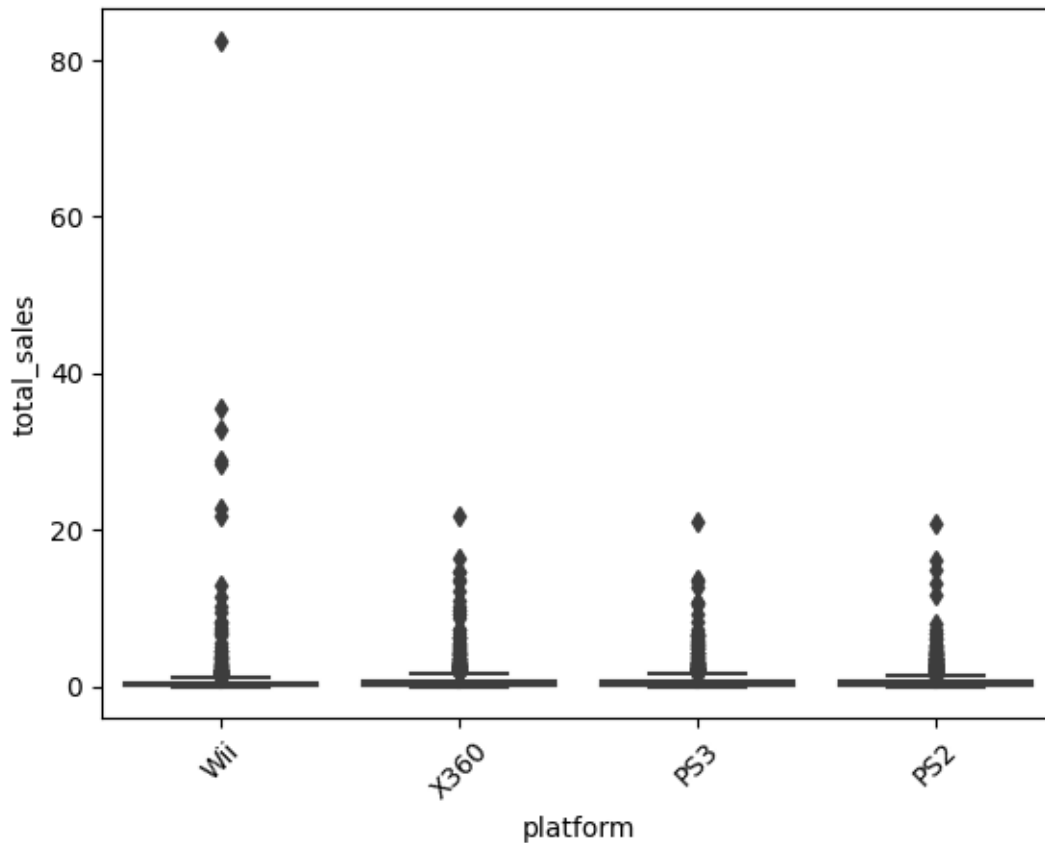
SNES	200.04	0.000000
GC	196.73	56.118081
XOne	159.32	50.170040
2600	86.48	0.000000
WiiU	82.19	43.306122
PSV	53.81	19.620047
SAT	33.59	0.000000
GEN	30.77	0.000000
DC	15.95	23.519231
SCD	1.86	0.000000
NG	1.44	0.000000
WS	1.42	0.000000
TG16	0.16	0.000000
3DO	0.10	0.000000
GG	0.04	0.000000
PCFX	0.03	0.000000

Existe una correlación positiva entre la puntuación promedio de los críticos y las ventas totales en algunas plataformas (como PC, X360, PS4). Esto sugiere que una buena recepción crítica puede impulsar las ventas. Plataformas como PS4 y XOne, con altas ventas totales y buenas puntuaciones de crítica, podrían ser consideradas como plataformas potencialmente rentables.

```
[17]: filtro_consolas = games[games['platform'].isin(['PS2', 'X360', 'PS3', 'Wii'])]

sns.boxplot(
    x = "platform",
    y = "total_sales",
    data=filtro_consolas)

plt.xticks(rotation=45)
plt.show()
```

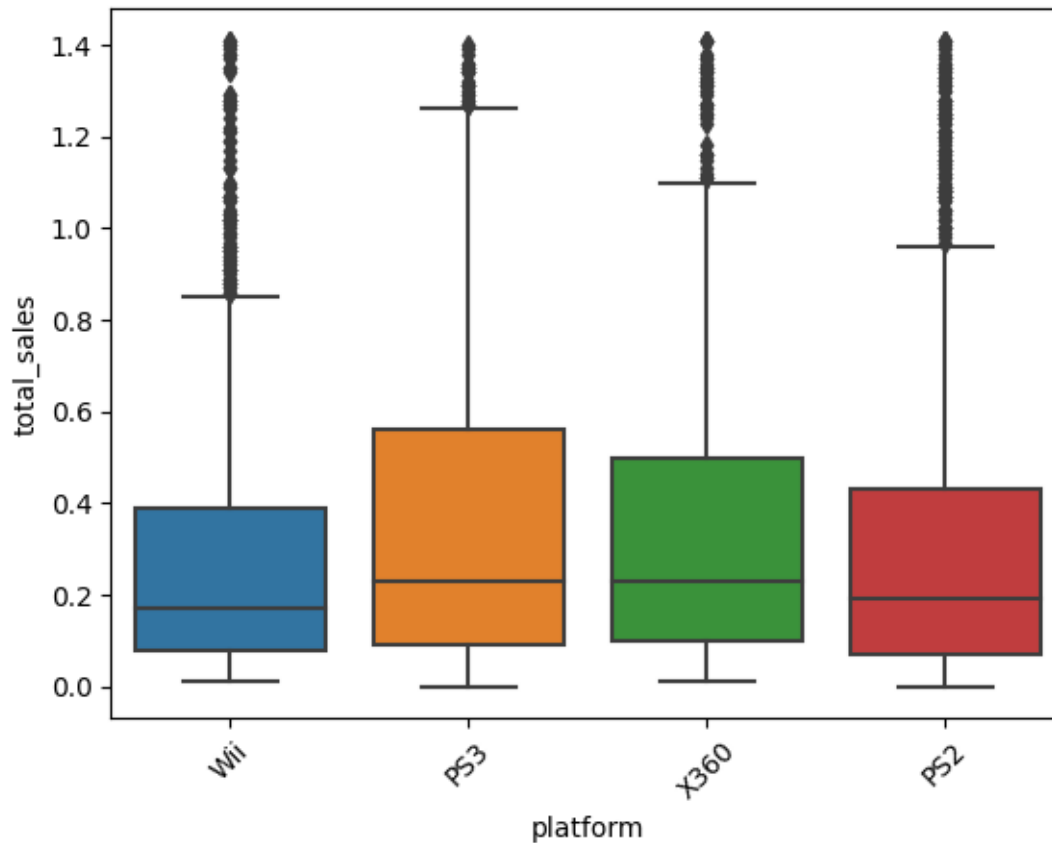
```
[18]: Q1 = filtro_consolas['total_sales'].quantile(0.25)
      Q3 = filtro_consolas['total_sales'].quantile(0.75)
      IQR = Q3 - Q1

      lower_bound = Q1 - 1.5 * IQR
      upper_bound = Q3 + 1.5 * IQR

      filtro_consolas_sin_atipicos = filtro_consolas[
          (filtro_consolas['total_sales'] >= lower_bound) &
          (filtro_consolas['total_sales'] <= upper_bound)]

      sns.boxplot(
          x="platform",
          y="total_sales",
          data=filtro_consolas_sin_atipicos)

      plt.xticks(rotation=45)
      plt.show()
```



```
[19]: promedio_puntuaciones = games.groupby("platform")["critic_score"].mean()
ventas = games.groupby ("platform")["total_sales"].sum()
df_concat = pd.concat([promedio_puntuaciones, ventas], axis="columns")

print(df_concat)
```

	critic_score	total_sales
platform		
2600	0.000000	86.48
3DO	0.000000	0.10
3DS	21.773438	257.81
DC	23.519231	15.95
DS	21.272984	802.78
GB	0.000000	254.43
GBA	35.689273	312.88
GC	56.118081	196.73
GEN	0.000000	30.77
GG	0.000000	0.04
N64	0.000000	218.01
NES	0.000000	251.05

NG	0.000000	1.44
PC	55.457680	255.76
PCFX	0.000000	0.03
PS	11.790756	727.58
PS2	41.189939	1233.56
PS3	43.382083	931.34
PS4	46.344388	314.14
PSP	25.813915	289.53
PSV	19.620047	53.81
SAT	0.000000	33.59
SCD	0.000000	1.86
SNES	0.000000	200.04
TG16	0.000000	0.16
WS	0.000000	1.42
Wii	27.727838	891.18
WiiU	43.306122	82.19
X360	49.841721	961.24
XB	61.466999	251.57
XOne	50.170040	159.32

No hay una correlación directa evidente entre la puntuación media de los críticos y las ventas totales. Algunas plataformas con altas puntuaciones (como PC) tienen ventas moderadas, mientras que otras con puntuaciones bajas (como PS2) lideran en ventas

```
[20]: x360 = games[games["platform"] == "X360"]
x360.loc[:, 'total_score'] = x360['user_score'] + x360['critic_score']

sns.scatterplot(x='total_score', y='total_sales', data=x360)
plt.title('Relación entre Puntuación Total y Ventas (Xbox 360)')
plt.xlabel('Puntuación Total (Usuario + Crítico)')
plt.ylabel('Ventas Totales')
plt.show()

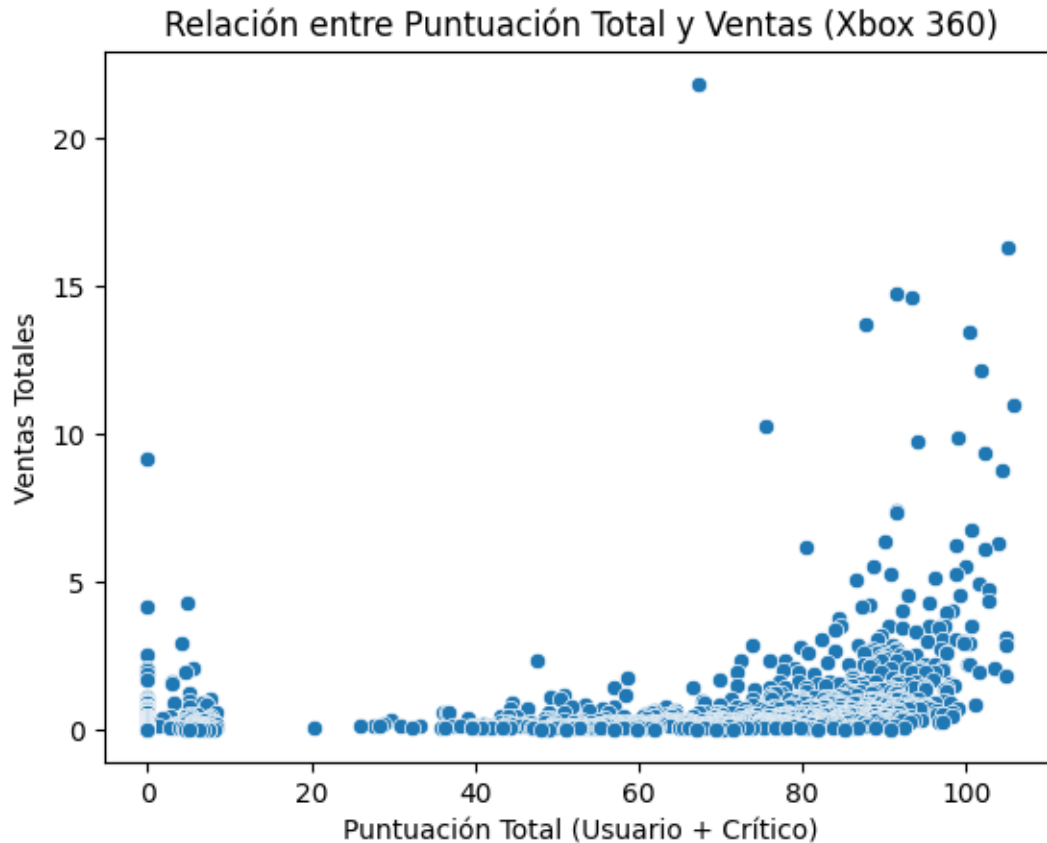
correlacion = x360['total_score'].corr(x360['total_sales'])
print('Correlación:', correlacion)
```

```
/opt/conda/envs/python3/lib/python3.9/site-
packages/pandas/core/indexing.py:1597: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
self.obj[key] = value
/opt/conda/envs/python3/lib/python3.9/site-
packages/pandas/core/indexing.py:1676: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
self._setitem_single_column(ilocs[0], value, pi)



Correlación: 0.30797981145425

Si bien existe una correlación positiva, no se puede afirmar que una puntuación alta garantice unas ventas altas.

```
[21]: unique_names = games['name'].unique()  
      print(unique_names)
```

```
['Wii Sports' 'Super Mario Bros.' 'Mario Kart Wii' ...  
 'Woody Woodpecker in Crazy Castle 5' 'LMA Manager 2007'  
 'Haitaka no Psychedelica']
```

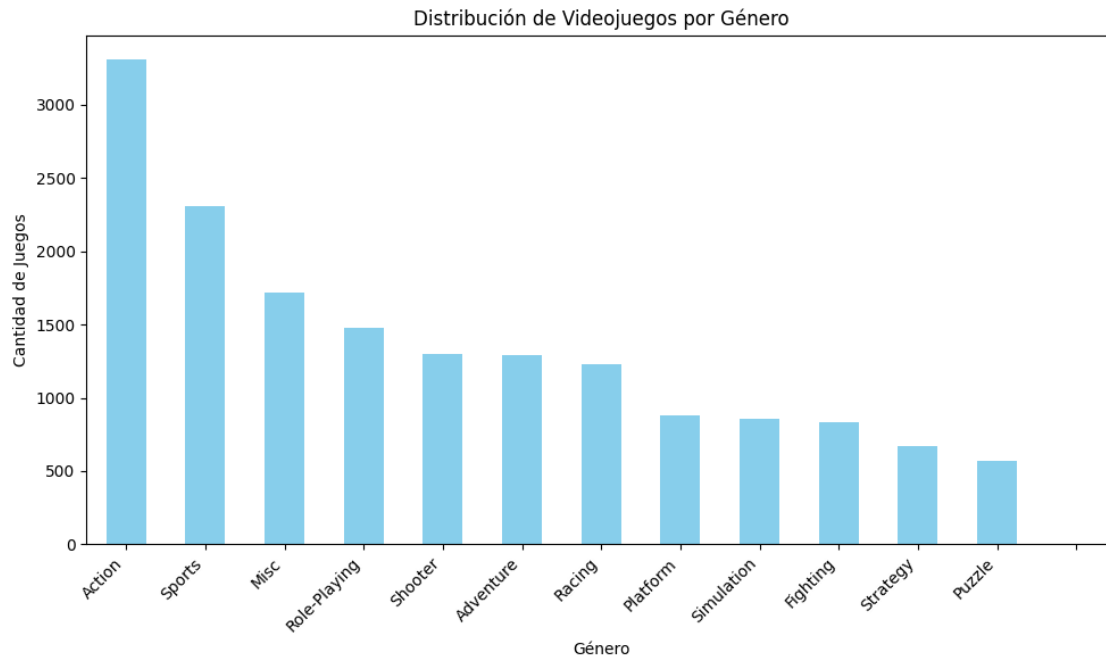
```
[22]: ventas_por_juego = games.groupby(['name', 'platform'])['total_sales'].sum().  
      ↪reset_index()
```

```
ventas_pivot = ventas_por_juego.pivot_table(index='name', columns='platform',  
↳ values='total_sales')
```

```
[23]: distribucion_por_genero = games['genre'].value_counts().  
↳ sort_values(ascending=False)  
  
print(distribucion_por_genero)
```

```
Action          3307  
Sports          2306  
Misc            1721  
Role-Playing    1481  
Shooter         1296  
Adventure       1293  
Racing          1226  
Platform        878  
Simulation      857  
Fighting        837  
Strategy        673  
Puzzle          569  
                2  
Name: genre, dtype: int64
```

```
[24]: plt.figure(figsize=(10, 6))  
distribucion_por_genero.plot(kind='bar', color='skyblue')  
plt.title('Distribución de Videojuegos por Género')  
plt.xlabel('Género')  
plt.ylabel('Cantidad de Juegos')  
plt.xticks(rotation=45, ha='right')  
plt.tight_layout()  
  
plt.show()
```



Los géneros de Acción y Deportes son los más populares, mientras que otros géneros tienen una representación más limitada

2.0.3 Perfil de usuario para cada region

```
[25]: ventas_por_plataforma_na = games.groupby('platform')['na_sales'].sum().
      ↪sort_values(ascending=False)

top_5_plataformas_na = ventas_por_plataforma_na.head(5)

print(top_5_plataformas_na)
```

```
platform
X360    595.74
PS2     572.92
Wii     486.87
PS3     390.13
DS      380.31
Name: na_sales, dtype: float64
```

```
[26]: ventas_por_plataforma_jp = games.groupby('platform')['jp_sales'].sum().
      ↪sort_values(ascending=False)

top_5_plataformas_jp = ventas_por_plataforma_jp.head(5)

print(top_5_plataformas_jp)
```

```
platform
DS      175.02
PS      139.78
PS2     137.54
SNES    116.55
3DS     100.62
Name: jp_sales, dtype: float64
```

```
[27]: ventas_por_plataforma_eu = games.groupby('platform')['eu_sales'].sum().
      ↪sort_values(ascending=False)

top_5_plataformas_eu = ventas_por_plataforma_eu.head(5)

print(top_5_plataformas_eu)
```

```
platform
PS2     332.63
PS3     327.21
X360    268.32
Wii     258.32
PS      212.39
Name: eu_sales, dtype: float64
```

Las variaciones en las cuotas de mercado de las plataformas de videojuegos entre las diferentes regiones reflejan una combinación de factores culturales, económicos y de marketing. Comprender estos patrones puede ser útil para las empresas de videojuegos a la hora de desarrollar estrategias de lanzamiento y marketing.

```
[28]: generos_na = games.groupby('genre')['na_sales'].sum().
      ↪sort_values(ascending=False)

top_5_generos_na = generos_na.head(5)

print(top_5_generos_na)
```

```
genre
Action    863.17
Sports    671.20
Shooter   584.83
Platform  444.44
Misc      399.57
Name: na_sales, dtype: float64
```

```
[29]: generos_jp = games.groupby('genre')['jp_sales'].sum().
      ↪sort_values(ascending=False)

top_5_generos_jp = generos_jp.head(5)
```

```
print(top_5_generos_jp)
```

```
genre
Role-Playing    353.39
Action          160.14
Sports          134.93
Platform        130.71
Misc            107.02
Name: jp_sales, dtype: float64
```

```
[30]: generos_eu = games.groupby('genre')['eu_sales'].sum().
      ↪sort_values(ascending=False)

top_5_generos_eu = generos_eu.head(5)

print(top_5_generos_eu)
```

```
genre
Action    510.99
Sports    371.33
Shooter   314.52
Racing    234.49
Misc      210.60
Name: eu_sales, dtype: float64
```

```
[39]: na_ventas_rating = games.groupby('rating')['na_sales'].sum().
      ↪sort_values(ascending=False)

total_na_ventas = games['na_sales'].sum()

pct_na_ventas_rating = (na_ventas_rating / total_na_ventas) * 100

print("Ventas Totales de NA por Clasificación ESRB:")
print(na_ventas_rating)

print("\nPorcentaje de Ventas Totales de NA para Cada Clasificación ESRB:")
print(pct_na_ventas_rating.round(1))
```

Ventas Totales de NA por Clasificación ESRB:

```
rating
E      1274.24
      1227.62
T      747.60
M      742.89
E10+   345.50
K-A     2.56
EC      1.53
AO      1.26
```



```
RP          0.00
Name: na_sales, dtype: float64
```

Porcentaje de Ventas Totales de NA para Cada Clasificación ESRB:

```
rating
E          29.3
          28.3
T          17.2
M          17.1
E10+       8.0
K-A         0.1
EC          0.0
AO          0.0
RP          0.0
```

```
Name: na_sales, dtype: float64
```

```
[38]: jp_ventas_rating = games.groupby('rating')['jp_sales'].sum().
      ↪sort_values(ascending=False)

total_jp_ventas = games['jp_sales'].sum()

pct_jp_ventas_rating = (jp_ventas_rating / total_jp_ventas) * 100

print("Ventas Totales de JP por Clasificación ESRB:")
print(jp_ventas_rating)

print("\nPorcentaje de Ventas Totales de JP para Cada Clasificación ESRB:")
print(pct_jp_ventas_rating.round(1))
```

Ventas Totales de JP por Clasificación ESRB:

```
rating
      836.77
E       197.96
T       150.70
M        63.58
E10+     40.20
K-A        1.46
AO         0.00
EC         0.00
RP         0.00
```

```
Name: jp_sales, dtype: float64
```

Porcentaje de Ventas Totales de JP para Cada Clasificación ESRB:

```
rating
      64.8
E       15.3
T       11.7
M         4.9
```

```

E10+      3.1
K-A       0.1
AO        0.0
EC        0.0
RP        0.0
Name: jp_sales, dtype: float64

```

```

[37]: eu_ventas_rating = games.groupby('rating')['eu_sales'].sum().
      ↪sort_values(ascending=False)

total_eu_ventas = games['eu_sales'].sum()

pct_eu_ventas_rating = (eu_ventas_rating / total_eu_ventas) * 100

print("Ventas Totales de EU por Clasificación ESRB:")
print(eu_ventas_rating)

print("\nPorcentaje de Ventas Totales de EU para Cada Clasificación ESRB:")
print(pct_eu_ventas_rating.round(1))

```

Ventas Totales de EU por Clasificación ESRB:

```

rating
E      703.87
      610.18
M      480.00
T      421.79
E10+   183.35
AO       0.61
K-A     0.27
EC       0.11
RP       0.03

```

Name: eu_sales, dtype: float64

Porcentaje de Ventas Totales de EU para Cada Clasificación ESRB:

```

rating
E      29.3
      25.4
M      20.0
T      17.6
E10+    7.6
AO       0.0
K-A     0.0
EC       0.0
RP       0.0

```

Name: eu_sales, dtype: float64

Las clasificaciones ESRB desempeñan un papel importante en la industria de los videojuegos, influyendo en las decisiones de compra de los consumidores y, por lo tanto, en las ventas de los

juegos. Las clasificaciones ESRB pueden limitar las opciones de marketing y publicidad para los juegos. Los juegos con clasificaciones más altas pueden tener restricciones en ciertos canales de publicidad, lo que puede afectar su alcance.

Como podemos observar en nuestros datos en Norte America hay mejores ventas a pesar de las clasificacion, pero en Japon, se evidencia como en clasificaciones altas como “Mature” posee muy pocas ventas.

2.0.4 Prueba de hipotesis

Hipotesis Nula (H0): Las calificaciones promedio de los usuarios para las plataformas Xbox One y PC son iguales.

Hipotesis Alternativa (H1): Las calificaciones promedio de los usuarios para las plataformas Xbox One y PC no son las mismas.

```
[36]: alpha = 0.05

xbox = games[games['platform'] == 'XOne']
pc = games[games['platform'] == 'PC']

promedio_xbox = 4.8
promedio_pc = 5.6

results = sts.ttest_ind(xbox["user_score"], pc["user_score"])

print('valor p:', results.pvalue)

if results.pvalue < alpha :
    print("Rechazamos la hipótesis nula")
else:
    print("No podemos rechazar la hipótesis nula")
```

valor p: 0.0007068097266608117
Rechazamos la hipótesis nula

```
[37]: xbox = games[games['platform'] == 'XOne']

promedio_user_score_xbox_one = xbox['user_score'].mean()

print(promedio_user_score_xbox_one)
```

4.805263157894737

```
[38]: pc = games[games['platform'] == 'PC']

promedio_user_score_pc = pc['user_score'].mean()
```

```
print(promedio_user_score_pc)
```

5.5745036572622775

El resultado indica que se rechaza la hipótesis nula. Esto significa que existe una diferencia estadísticamente significativa entre las calificaciones promedio de los usuarios de Xbox One y PC. Específicamente, las calificaciones promedio de PC son significativamente más altas que las de Xbox One.

Se uso para probar esta hipotesis una prueba t para comparar las medias de dos grupos

Hipotesis Nula (H0): Las calificaciones promedio de los usuarios para los géneros de Acción y Deportes son diferentes.

Hipotesis Alternativa (H1): Las calificaciones promedio de los usuarios para los géneros de Acción y Deportes no son diferentes.

```
[40]: accion= games[games['genre'] == 'Action']
      sport = games[games['genre'] == 'Sports']

      results = sts.ttest_ind(accion['user_score'], sport['user_score'])

      print('valor p:',results.pvalue)

      if results.pvalue < alpha :
          print("Rechazamos la hipótesis nula")
      else:
          print("No podemos rechazar la hipótesis nula")
```

valor p: 7.492568776804778e-09

Rechazamos la hipótesis nula

```
[41]: accion = games[games['genre'] == 'Action']

      promedio_user_score_accion = accion['user_score'].mean()

      print(promedio_user_score_accion)
```

3.833202298155427

```
[42]: sport = games[games['genre'] == 'Sports']

      promedio_user_score_sport = sport['user_score'].mean()

      print(promedio_user_score_sport)
```

3.2586730268863833

Se uso para probar esta hipotesis una prueba t para comparar las medias de dos grupos.

Y como resultado, podemos concluir con un alto grado de confianza que las calificaciones promedio de los usuarios para los géneros de Acción y Deportes no son significativamente diferentes.

2.0.5 Conclusion general

Luego del analisis de los datos se puede concluir varias cosas, primero las plataformas como PS2, Xbox 360 y PS3 han dominado el mercado en términos de ventas totales. Sin embargo, se observa una clara evolución con el surgimiento de nuevas consolas como PS4 y Xbox One.

Por otro lado el mercado de videojuegos ha experimentado un crecimiento significativo en las últimas décadas, con picos y valles asociados a lanzamientos importantes y cambios tecnológicos. Asi mismo, los géneros de acción y deportes son los más populares en términos de ventas, aunque la preferencia puede variar según la región.

Existe una cierta correlación entre las puntuaciones de los críticos y las ventas, pero no es una relación perfecta. Otros factores como el marketing y las tendencias de la industria también influyen y por ultimo las preferencias de los jugadores varían significativamente entre las diferentes regiones. Por ejemplo, los juegos de rol son más populares en Japón, mientras que los juegos de acción son más populares en Norteamérica y Europa.