UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores Acatlán

diplomado en técnicas estadísticas y minería de datos

Preferencias del turismo extranjero en México

MÓDULO II. MODELOS ESTADÍSTICOS

Presenta:
Angel Jesus Martinez Briones

17 de diciembre 2025

Índice

1. Introducción

El turismo es una de las actividades económicas más relevantes en México ya que contribuye de manera significativa al empleo y al desarrollo económico del país. De acuerdo con datos de la Secretaría de Turismo (SECTUR, 2024), esta actividad representa aproximadamente el 8.6 % del Producto Interno Bruto (PIB) nacional. Dentro del sector del turismo la industria hotelera desempeña un papel fundamental al ofrecer servicios de hospedaje que influyen directamente en la experiencia del turista, este es un tema que sin importar el año se mantiene vigente puesto es un sector muy importante para el desarrollo del país.

El presente trabajo tiene como objetivo analizar estadísticamente las preferencias hoteleras del turismo en México, considerando variables como la categoría del establecimiento, el destino turístico, la ocupación de cuartos y la llegada de turistas residentes y no residentes. A través de este análisis se busca identificar patrones que permitan comprender mejor las decisiones que toman los turistas extranjeros cuando visitan México.

El estudio de estas variables resulta relevante ya que un mejor entendimiento del comportamiento del turismo puede contribuir a la toma de decisiones de los empresarios e inversionistas, además de favorecer el desarrollo económico de las comunidades que dependen de esta actividad.

2. Descripción de los datos

Los datos utilizados en este estudio fueron obtenidos a través de la plataforma *datos.gob.mx*, una plataforma de libre acceso que almacena bases de datos recopiladas por diversas instituciones mexicanas. En este trabajo, se empleó información correspondiente al sector turismo, recopilada por la Secretaría de Turismo (SECTUR).

La base de datos analizada lleva por nombre *"Ocupación en establecimientos de hospedaje de Categoría Turística"* y forma parte del sistema DataTur. Esta base de datos presenta información sobre la ocupación hotelera en 70 destinos turísticos principales del país.

El conjunto de datos contiene un total de 31,173 registros obtenidos durante los años 2016 y 2024. La información que se incluye en esta base de datos son el año, mes, tipo de centro turístico, categoría del establecimiento, número de cuartos disponibles, así como el número de cuartos ocupados y la llegada de turistas residentes y no residentes.

3. Metodología

Para analizar las preferencias hoteleras del turismo en México, se llevó a cabo una fase de exploración descriptiva. En primer lugar, se realizó la limpieza y preparación de los datos, asegurando la correcta conversión de las variables a sus tipos adecuados (cuantitativas y categóricas) para garantizar la fiabilidad del análisis.

Se emplearon diversos diagramas de comparación para analizar variables clave entre las distintas categorías hoteleras. Este análisis gráfico facilitó la detección de patrones de ocupación diferenciados según la categoría de los hoteles y el tipo de centro (playa frente a ciudad), permitiendo identificar sesgos de preferencia en los segmentos de turistas residentes y no residentes.

Posteriormente se realizó un análisis de correlación entre las variables numéricas y se calcularon intervalos de confianza al 95 % con el fin de ver el grado de correlación lineal entre la disponibilidad de cuartos, la ocupación y el flujo de turistas .

finalmente implementó un modelo de regresión lineal simple, el objetivo de este modelo fue modelar la relación entre la capacidad instalada y la demanda efectiva, permitiendo realizar estimaciones sobre la ocupación y el flujo turístico. La aplicación de este análisis permitió validar los hallazgos descriptivos y confirmar las variables críticas en el ecosistema turístico mexicano

4. Resultados

En primer lugar, se realizó un análisis gráfico de la llegada de turistas a los hoteles, diferenciando entre no residentes (en adelante, turistas extranjeros) y residentes (turistas locales). En la Figura 1 se observa una diferencia en las preferencias según el tipo de destino. Algo notable es que los turistas extranjeros eligen las playas como su principal destino, representando el 71.87 % de sus visitas. Por el contrario, el turista local muestra una marcada preferencia por las ciudades de la República Mexicana, destino que concentra el 65.52 % de sus preferencias.

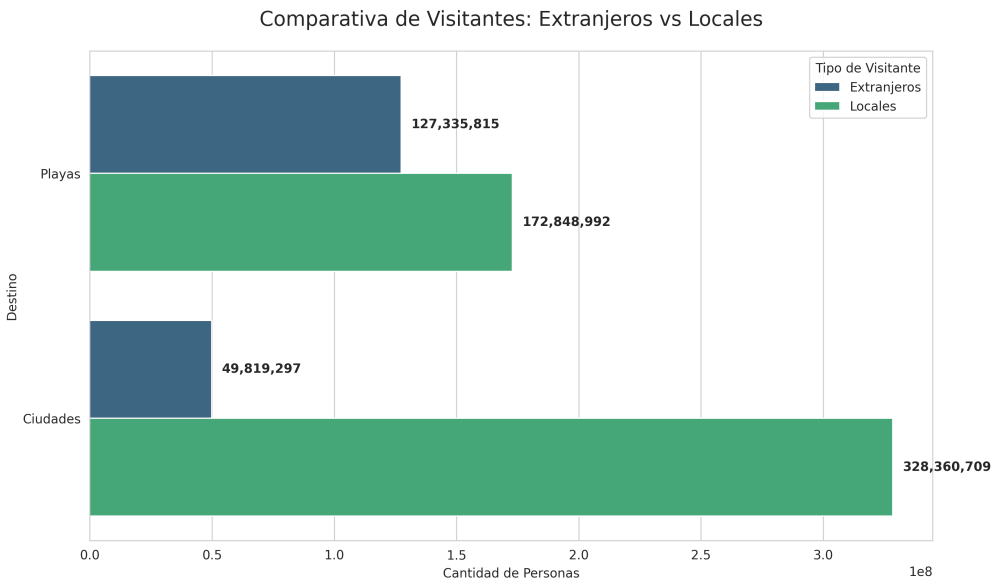


Figura 1: Comparativa de visitantes extranjeros y locales sobre su preferencia a un tipo de centro turístico

Por otra parte se busco una comparativa sobre la preferencia de los turistas sobre la clasificación del recinto donde deseaban hospedarse, lo que llevo a la gráfica visualizada en la figura 2. Podemos observar que el turista extranjero tiene una preferencia más marcada por la categoría del hotel. No obstante, son pocos los turistas locales que realizan una estadía en hoteles de 5 estrellas, hablamos de solo un 27.00 %, lo cual contrasta con el 64.24 % registrado en los turistas extranjeros.

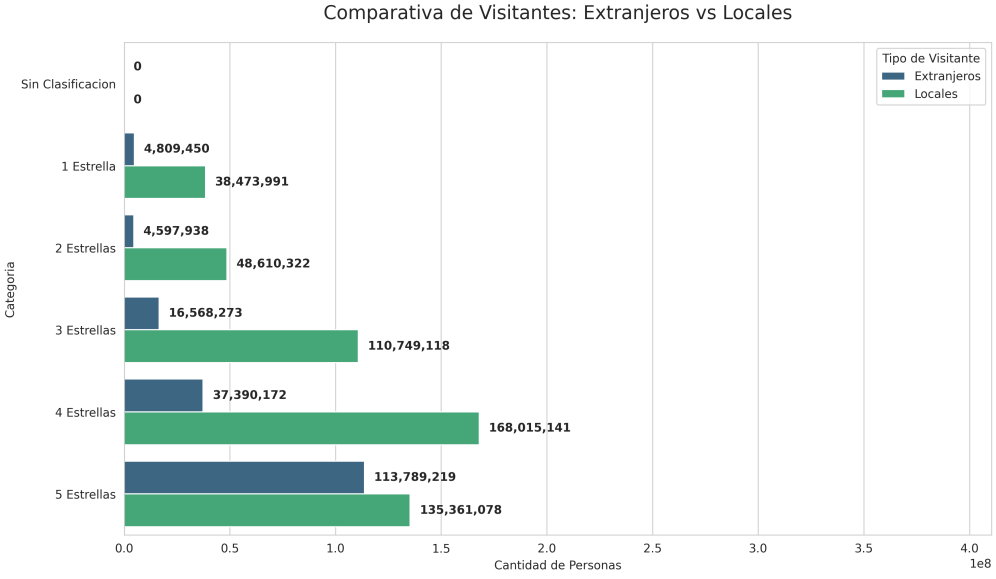


Figura 2: Comparativa de visitantes extranjeros y locales sobre su preferencia a la clasificación de un hotel

Tras concluir la fase de exploración inicial basada en estadística descriptiva, se procedió a ejecutar un análisis de correlación. El objetivo de esta etapa fue verificar el grado de asociación lineal entre las variables cuantitativas presentes en la base de datos. Los resultados de este análisis son los siguientes.

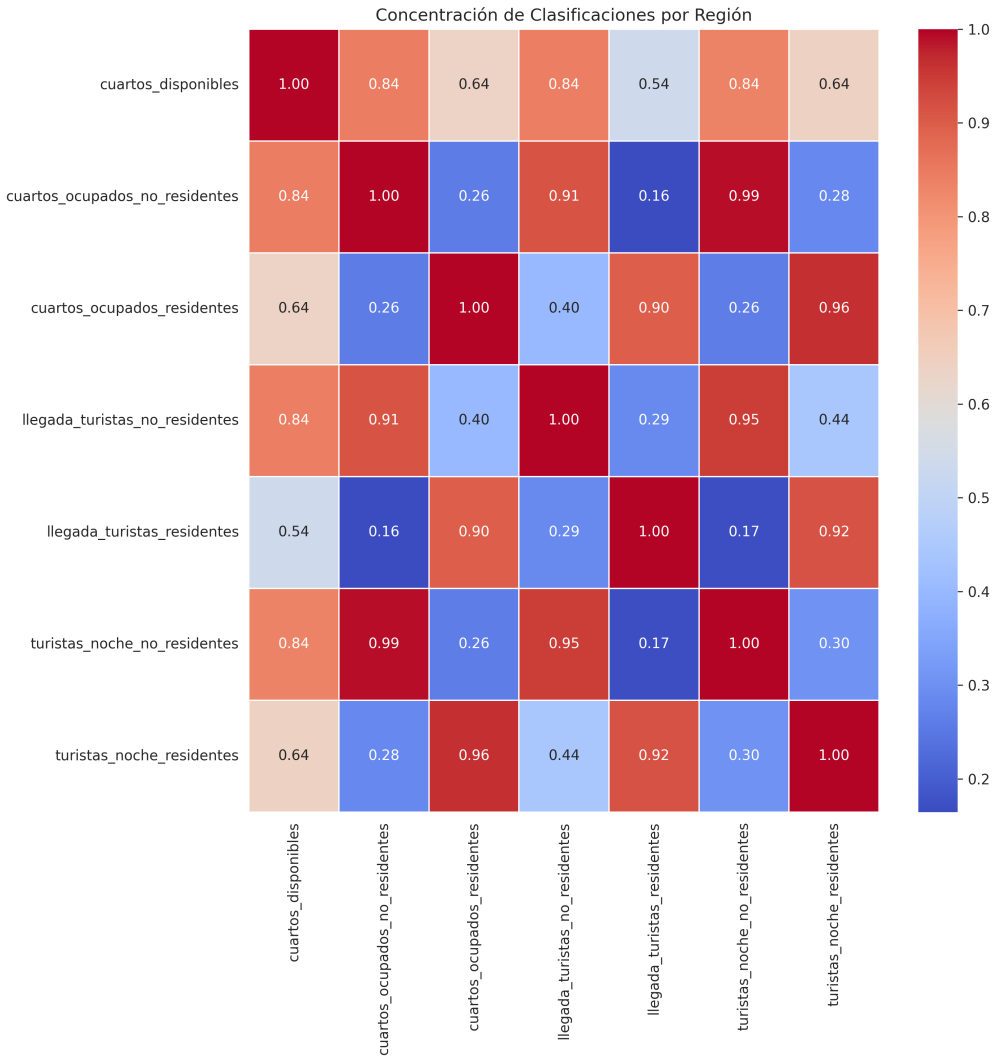


Figura 3: Correlaciones de las variables cuantitativas

Todas las correlaciones son positivas, no hay relaciones negativas, así que no hay variables que se muevan en sentido contrario, se puede observar que los coeficientes mas fuertes se presentan en las relaciones tipo: cuando aumentan turistas entonces aumentan cuartos ocupados y por lo tanto aumentan turistas-noche, lo cual es lógico, no obstante también encontramos una relación fuerte con la cantidad de cuartos disponibles, la llegada de turistas extranjeros, cuartos ocupados por no residentes y su estancia:

- Cuartos disponibles $\longleftrightarrow$ Cuartos ocupados no residentes **0.84**
- Cuartos disponibles $\longleftrightarrow$ Llegada turistas no residentes **0.84**

- Cuartos disponibles $\longleftrightarrow$ Turistas noche no residentes **0.84**

Esto quiere decir que donde hay más infraestructura hotelera, hay más turismo extranjero y mayor estancia, lo cual concuerda con los resultados de la figura 2 donde pudimos observar la preferencia de este tipo de turistas por los hoteles 5 estrellas.

Dada esta información resulta importante conocer la media de cuartos disponibles, los cuartos ocupados por turistas extranjeros, su llegada y la noches que pasaban, este análisis se realizo por mes con un intervalo de confianza del 95 %, lo que se obtuvo fue lo siguiente:

- Cuartos disponibles $\rightarrow$ **42,552.44 - 44,788.67**
- Cuartos ocupados no residentes $\rightarrow$ **8,706.62 - 9,980.53**
- Llegada turistas no residentes $\rightarrow$ **5,376.60 - 5,999.18**
- Turistas noche no residentes $\rightarrow$ **19,352.50 - 22,306.62**

A continuación, se aplicó un modelo de regresión lineal simple para predecir el comportamiento de las variables analizadas. El modelo arrojó los siguientes resultados

$$\textbf{Cuartos Ocupados} = (\textbf{0.4807} \times \textbf{Cuartos Disponibles}) - \textbf{11650.0003}$$

Por lo que si el destino tiene 100 nuevos cuartos disponibles, se puede esperar que aproximadamente 48 de ellos sean ocupados por los turistas extranjeros, podemos validar entonces que la infraestructura hotelera es un factor directo para la captación de turistas extranjeros

$$\textbf{Llegada de Turista} = (\textbf{0.2342} \times \textbf{Cuartos Disponibles}) - \textbf{4540.7017}$$

Con una pendiente de 0.2342, el modelo demuestra que la oferta habitacional garantiza la ocupación, y actúa como la atracción de flujo turístico. Por cada 1,000 cuartos disponibles, el destino puede esperar, en promedio, la llegada de 234 nuevos turistas extranjeros.

$$\textbf{Noches de estancia} = (\textbf{1.1046} \times \textbf{Cuartos Disponibles}) - \textbf{27409.3504}$$

El análisis de regresión para las noches que pasan los turistas extranjeros revela el mayor coeficiente de m = 1.10. Esto demuestra que la disponibilidad de habitaciones influye en el tiempo de estancia en el destino. La capacidad del hotel repercute a la estadística de permanencia.

Otro análisis que se realizo fue el de describir los meses de temporada alta y los meses de temporada baja, para ella se realizo un gráfico de lineas, donde se puede ver los promedios de visitas por mes durante los años de 2016 a 2024, se puede observar en la figura 4 que la temporada alta son los meses de marzo, julio y diciembre, mientras que la temporada baja son los meses de febrero junio y septiembre.

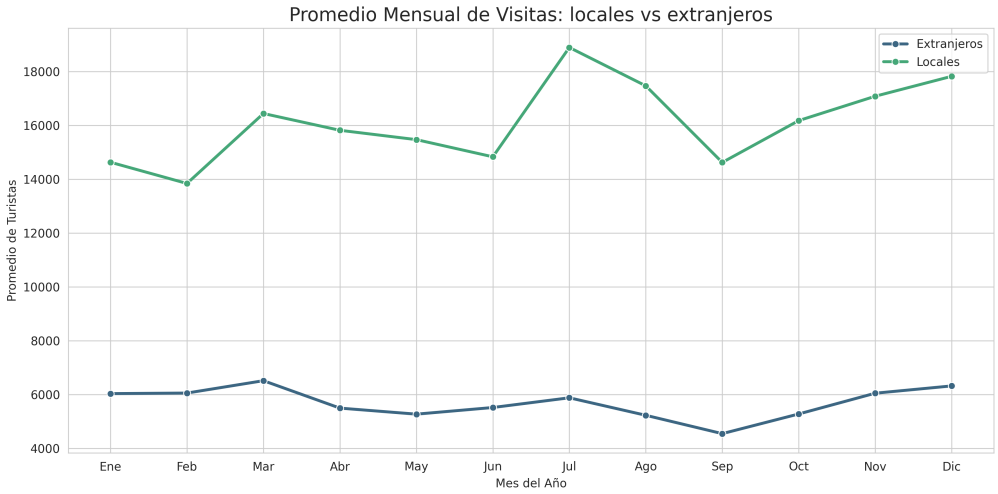


Figura 4: Visitas de los turistas locales y extranjeros

5. Conclusiones

La presente investigación permite concluir que existe una marcada elección del turismo extranjero por los destinos de sol y playa frente a otras alternativas. Mediante el análisis de modelos de regresión lineal simple y coeficientes de correlación, se demostró que esta preferencia está ligada a la búsqueda de confort y exclusividad, lo que explica la alta demanda de alojamientos de cinco estrellas.

Se identificó que el flujo turístico no es uniforme a lo largo del año, presentando la temporada alta en los meses de marzo, julio y diciembre. Estos periodos coinciden con las vacacionales internacionales, donde los hoteles alcanzan buenos niveles de ingresos.

El estudio estadístico en un intervalo de confianza del 95 %, permitió entender el comportamiento mensual de las variables analizadas permitiendo entender mejor como se desenvuelven estas variables a nivel nacional. La investigación sugiere que la calidad de la infraestructura y la capacidad del hotel son factores determinantes en la elección del destino, y actúan como variables que prolongan el tiempo de estancia.

Podemos concluir que la relación entre la oferta de infraestructura de los hoteles y la respuesta del turista extranjero es directa y positiva. Podemos afirmar inversión de estos recintos ubicados en zonas costeras sobre capacidad operativa y servicios de lujo de los hoteles es la estrategia más efectiva para incrementar su operatividad.

6. Fuentes de informacion

Secretaría de Turismo. (21 de noviembre de 2024). PIB Turístico creció 4.4 por ciento; ascendió a 2 billones 582 mil millones de pesos. Gobierno de México.

https://www.gob.mx/sectur/articulos/pib-turistico-crecio-4-4-por-ciento-ascendio-a-2-billones-582-mil-millones-de-pesos

Pandas Development Team. (n.d.). Pandas. Pandas: Python Data Analysis Library. https://pandas.pydata.org/

NumPy Developers. (n.d.). NumPy. NumPy. https://numpy.org/

Matplotlib Development Team. (n.d.). Matplotlib. Matplotlib: Visualization with Python. https://matplotlib.org/

Waskom, M. L. (2021). Seaborn: Statistical data visualization. https://seaborn.pydata.org/

Secretaría de Turismo. (s.f.). Ocupación hotelera 70 destinos principales monitoreados DataTur [Conjunto de datos]. Datos.gob.mx. Recuperado el 20 de diciembre de 2025, de https://datos.gob.mx/dataset/ocupacion_hotelera_70_destinos_principales_monitoreados_datatur

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272. https://doi.org/10.1038/s41592-019-0686-2

7. Anexos

05_Practica_EDA-Copy1

December 20, 2025

1 Práctica 05. Análisis exploratorio de datos

Nombre(s) Apellidos Angel Jesus Martinez Briones

Fecha 20 de diciembre del 2025

Realiza un análisis exploratorio de los datos que has elegido para tu proyecto integrador.

2 Importacion y limpieza de datos

[87]:
```
# Módulos
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import skew, kurtosis
from sklearn.linear_model import LinearRegression
```

[2]:
```
# Cargamos los datos
df = pd.read_csv("Base70centros.csv")
```

[3]:
```
# Verificamos cuantas observaciones tenemos y cuantas columas
df.shape
```

[3]:
```
(31147, 1)
```

[4]:
```
# Como solo hay una quise saber cuantas hay
df.columns
```

[4]:
```
Index(['anio\tmes\ttipo_centro\tsubtipo_centro\tcentro\tcategoria\tcuartos_dispo
nibles\tcuartos_ocupados_no_residentes\tcuartos_ocupados_residentes\tllegada_tur
istas_no_residentes\tllegada_turistas_residentes\tturistas_noche_no_residentes\t
turistas_noche_residentes'], dtype='object')
```

[5]:
```
# Estan unidas en una sola columna
df.head()
```

[5]:
```
  anio\tmes\ttipo_centro\tsubtipo_centro\tcentro\tcategoria\tcuartos_disponibles
\tcuartos_ocupados_no_residentes\tcuartos_ocupados_residentes\tllegada_turistas_
no_residentes\tllegada_turistas_residentes\tturistas_noche_no_residentes\tturist
as_noche_residentes
0  2016\t01\tCentros de Playa\tIntegralmente Plan…
1  2016\t01\tCentros de Playa\tIntegralmente Plan…
2  2016\t01\tCentros de Playa\tIntegralmente Plan…
3  2016\t01\tCentros de Playa\tIntegralmente Plan…
4  2016\t01\tCentros de Playa\tIntegralmente Plan…
```

[6]:
```
#sSelecciono la coluna
columna_unica = df.iloc[:, 0]
```

[7]:
```
#Separamos las palabras de la columna
df_final = columna_unica.str.split("\t", expand=True)
```

[8]:
```
df_final.head()
```

[8]:

	0	1	2	3	4	\
0	2016	01	Centros de Playa	Integralmente Planeados	Ixtapa - Zihuatanejo	
1	2016	01	Centros de Playa	Integralmente Planeados	Ixtapa - Zihuatanejo	
2	2016	01	Centros de Playa	Integralmente Planeados	Ixtapa - Zihuatanejo	
3	2016	01	Centros de Playa	Integralmente Planeados	Bahías de Huatulco	
4	2016	01	Centros de Playa	Integralmente Planeados	Bahías de Huatulco	

	5	6	7	8	9	10	11	12
0	3 estrellas	10819	5333	2421	3888	2023	10116	5175
1	4 estrellas	45090	8114	29303	3941	21399	15301	63565
2	5 estrellas	105015	32190	47030	12240	44790	64356	109299
3	2 estrellas	5549	0	2571	0	4816	0	6320
4	3 estrellas	12896	0	4360	0	8560	0	11464

[9]:
```
# Agregamos los encabezados que se eliminaron
encabezados = ['anio', 'mes', 'tipo_centro', 'subtipo_centro', 'centro',␣
 ↪'categoria', 'cuartos_disponibles',
               'cuartos_ocupados_no_residentes', 'cuartos_ocupados_residentes',␣
 ↪'llegada_turistas_no_residentes',
               'llegada_turistas_residentes', 'turistas_noche_no_residentes',␣
 ↪'turistas_noche_residentes']

df_final.columns = encabezados
```

[10]:
```
# listo ya estan bien separadas
df_final.head()
```

[10]:

	anio	mes	tipo_centro	subtipo_centro	centro	\
0	2016	01	Centros de Playa	Integralmente Planeados	Ixtapa - Zihuatanejo	

```
1  2016  01  Centros de Playa  Integralmente Planeados  Ixtapa - Zihuatanejo
2  2016  01  Centros de Playa  Integralmente Planeados  Ixtapa - Zihuatanejo
3  2016  01  Centros de Playa  Integralmente Planeados    Bahías de Huatulco
4  2016  01  Centros de Playa  Integralmente Planeados    Bahías de Huatulco

     categoria cuartos_disponibles cuartos_ocupados_no_residentes  \
0  3 estrellas               10819                           5333
1  4 estrellas               45090                           8114
2  5 estrellas              105015                          32190
3  2 estrellas                5549                              0
4  3 estrellas               12896                              0

  cuartos_ocupados_residentes llegada_turistas_no_residentes  \
0                        2421                           3888
1                       29303                           3941
2                       47030                          12240
3                        2571                              0
4                        4360                              0

  llegada_turistas_residentes turistas_noche_no_residentes  \
0                        2023                         10116
1                       21399                         15301
2                       44790                         64356
3                        4816                             0
4                        8560                             0

  turistas_noche_residentes
0                      5175
1                     63565
2                    109299
3                      6320
4                     11464
```

[11]:
```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31147 entries, 0 to 31146
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   anio                            31147 non-null  object
 1   mes                             31147 non-null  object
 2   tipo_centro                     31147 non-null  object
 3   subtipo_centro                  31147 non-null  object
 4   centro                          31147 non-null  object
 5   categoria                       31147 non-null  object
 6   cuartos_disponibles             31147 non-null  object
 7   cuartos_ocupados_no_residentes  31147 non-null  object
```

```
 8   cuartos_ocupados_residentes     31147 non-null  object
 9   llegada_turistas_no_residentes  31147 non-null  object
 10  llegada_turistas_residentes     31147 non-null  object
 11  turistas_noche_no_residentes    31147 non-null  object
 12  turistas_noche_residentes       31147 non-null  object
dtypes: object(13)
memory usage: 3.1+ MB
```

[12]:
```
print("\n años: \n",df_final['anio'].unique())
```

```
 años:
 ['2016' '2017' '2018' '2019' '2020' '2021' '2022' '2023' '2024']
```

[13]:
```
# Cambiamos de tipo a las columnas que son numericas
columnas_numericas =␣
 ↪['cuartos_disponibles','cuartos_ocupados_no_residentes','cuartos_ocupados_residentes','lleg
   ␣
 ↪'llegada_turistas_residentes','turistas_noche_no_residentes','turistas_noche_residentes']

df_final[columnas_numericas] = df_final[columnas_numericas].apply(pd.
 ↪to_numeric, errors="coerce")

# Unificamos una sola columa tipo fecha a la columna anio y mes
df_final['fecha'] = pd.to_datetime(dict(year=df_final['anio'],␣
 ↪month=df_final['mes'], day=1))
df_final.drop(columns=['anio', 'mes'], inplace=True)
```

[14]:
```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31147 entries, 0 to 31146
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   tipo_centro                     31147 non-null  object
 1   subtipo_centro                  31147 non-null  object
 2   centro                          31147 non-null  object
 3   categoria                       31147 non-null  object
 4   cuartos_disponibles             31147 non-null  int64
 5   cuartos_ocupados_no_residentes  31147 non-null  int64
 6   cuartos_ocupados_residentes     31147 non-null  int64
 7   llegada_turistas_no_residentes  31147 non-null  int64
 8   llegada_turistas_residentes     31147 non-null  int64
 9   turistas_noche_no_residentes    31147 non-null  int64
 10  turistas_noche_residentes       31146 non-null  float64
 11  fecha                           31147 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(6), object(4)
```

```
memory usage: 2.9+ MB
```

[15]:
```
# Verificamos si hay espacios nulos
df_final.isna().sum()
```

[15]:
```
tipo_centro                      0
subtipo_centro                   0
centro                           0
categoria                        0
cuartos_disponibles              0
cuartos_ocupados_no_residentes   0
cuartos_ocupados_residentes      0
llegada_turistas_no_residentes   0
llegada_turistas_residentes      0
turistas_noche_no_residentes     0
turistas_noche_residentes        1
fecha                            0
dtype: int64
```

[16]:
```
# Eliminamos espacios nulos y volvemos a verificar
df_final = df_final.dropna()
df_final.isna().sum()
```

[16]:
```
tipo_centro                      0
subtipo_centro                   0
centro                           0
categoria                        0
cuartos_disponibles              0
cuartos_ocupados_no_residentes   0
cuartos_ocupados_residentes      0
llegada_turistas_no_residentes   0
llegada_turistas_residentes      0
turistas_noche_no_residentes     0
turistas_noche_residentes        0
fecha                            0
dtype: int64
```

3 Medidas Descriptivas

[17]:
```
# Me gustaria saber cuatos grupos pueden hacerse por categoria cualitativa asi␣
 ↪que imprimo para ver su contenido
print("\n Tipo de Centro: \n",df_final['tipo_centro'].unique())
print("\n Sub Tipo de Centro: \n",df_final['subtipo_centro'].unique())
print("\n Centro: \n",df_final['centro'].unique())
print("\n Categorias: \n",df_final['categoria'].unique())
```

```
 Tipo de Centro:
```

```
['Centros de Playa' 'Ciudades']

Sub Tipo de Centro:
['Integralmente Planeados' 'Otros' 'Tradicionales' 'Grandes' 'Interior'
'Fronterizas']

Centro:
['Ixtapa - Zihuatanejo' 'Bahías de Huatulco' 'Cancún' 'Loreto' 'Los Cabos'
'Isla Mujeres' 'Nuevo Nayarit' 'Playas de Rosarito' 'Puerto Escondido'
'Riviera Maya' 'San Felipe' 'Tonalá Puerto Arista' 'Acapulco' 'Cozumel'
'La Paz' 'Manzanillo' 'Mazatlán' 'Veracruz-Boca del Río'
'Puerto Vallarta' 'Ciudad de México' 'Guadalajara' 'Monterrey'
'Aguascalientes' 'Campeche' 'Celaya' 'Chihuahua' 'Coatzacoalcos' 'Colima'
'Comitán de Domínguez' 'Culiacán' 'Durango' 'El Fuerte' 'Guanajuato'
'Hermosillo' 'Irapuato' 'León' 'Los Mochis' 'Mérida' 'Morelia' 'Oaxaca'
'Pachuca' 'Palenque' 'Puebla' 'Querétaro' 'Salamanca'
'San Cristóbal de las Casas' 'San Juan de los Lagos' 'San Juan del Río'
'San Luis Potosí' 'San Miguel de Allende' 'Taxco' 'Tequisquiapan'
'Tlaxcala' 'Toluca' 'Tuxtla Gutiérrez' 'Valle de Bravo' 'Villahermosa'
'Xalapa' 'Zacatecas' 'Ciudad Juárez' 'Mexicali' 'Piedras Negras' 'Tecate'
'Tijuana']

Categorias:
['3 estrellas' '4 estrellas' '5 estrellas' '2 estrellas' '1 estrella'
'Sin categoría']
```

[18]:
```
# Vamos a aplicar un describe para poder ver unas breves medidas descriptivas
df_final.describe()
```

[18]:

	cuartos_disponibles	cuartos_ocupados_no_residentes	\
count	3.114600e+04	3.114600e+04	
mean	4.367056e+04	9.343579e+03	
min	8.000000e+00	0.000000e+00	
25%	7.560000e+03	0.000000e+00	
50%	1.592800e+04	2.735000e+02	
75%	3.806800e+04	2.154000e+03	
max	1.317790e+06	1.107395e+06	
std	1.006751e+05	5.735170e+04	

	cuartos_ocupados_residentes	llegada_turistas_no_residentes	\
count	31146.000000	31146.000000	
mean	14162.060232	5687.892892	
min	0.000000	0.000000	
25%	1744.000000	0.000000	
50%	4823.000000	285.000000	
75%	13503.000000	2023.000000	
max	294189.000000	542220.000000	

```
std                   27376.033527                    28028.730036

       llegada_turistas_residentes  turistas_noche_no_residentes  \
count                 31146.000000                  3.114600e+04
mean                  16092.265492                  2.082957e+04
min                       0.000000                  0.000000e+00
25%                    2503.000000                  0.000000e+00
50%                    6450.000000                  4.820000e+02
75%                   16576.500000                  3.977750e+03
max                  373532.000000                  2.444202e+06
std                   29199.800061                  1.329950e+05

       turistas_noche_residentes                          fecha
count               3.114600e+04                          31146
mean                2.896731e+04  2020-05-26 18:46:12.644962560
min                 0.000000e+00            2016-01-01 00:00:00
25%                 3.424000e+03            2018-03-01 00:00:00
50%                 9.146000e+03            2020-05-01 00:00:00
75%                 2.611400e+04            2022-09-01 00:00:00
max                 1.004741e+06            2024-12-01 00:00:00
std                 5.806651e+04                            NaN
```

[19]:
```
df_final.describe(include='object')
```

[19]:
```
       tipo_centro subtipo_centro          centro    categoria
count        31146          31146           31146        31146
unique           2              6              64            6
top       Ciudades       Interior  Valle de Bravo  3 estrellas
freq         21977          18370             552         6868
```

[20]:
```
def coef_variation(x):
    return np.std(x, ddof=1) / np.mean(x)

def asim(x):
    """Calcula la asimetría (skewness) ignorando valores NaN."""
    return skew(x, nan_policy='omit')

def curt(x):
    """Calcula la curtosis (kurtosis) ignorando valores NaN."""
    return kurtosis(x, nan_policy='omit')
```

[21]:
```
summary_funcs = [
    ('n', 'count'),
    ('media', 'mean'),
    ('std', 'std'),
    ('min', 'min'),
    ('25%', lambda x: x.quantile(0.25)),
```

```
    ('50%', 'median'),
    ('75%', lambda x: x.quantile(0.75)),
    ('max', 'max'),
    ('asim', asim),
    ('curt', curt),
    ('cv', coef_variation)
]
summary_centro = df_final.groupby('centro',␣
 ↪observed=False)['llegada_turistas_no_residentes'].agg(summary_funcs)
```

[22]:
```
summary_centro
```

[22]:

centro	n	media	std	min	25%	50%	75%	max	asim	curt	cv
Acapulco	494	1935.459514	3064.167526	0	0.0	46.0	3088.50	15137	1.800661	2.830144	1.583173
Aguascalientes	540	1461.379630	1655.660475	0	0.0	1216.0	2116.00	10445	1.824641	4.249962	1.132943
Bahías de Huatulco	432	1203.622685	1960.624600	0	0.0	55.5	1655.25	9338	2.060656	3.756607	1.628936
Campeche	527	1195.743833	1654.342380	0	0.0	355.0	1894.00	7379	1.622542	1.949591	1.383526
Cancún	539	80362.515770	126536.171304	0	851.0	20820.0	82609.50	542220	1.779812	1.986198	1.574567
…	…	…	…	…	…	…	…	…	…	…	…
Valle de Bravo	552	139.920290	485.328453	0	0.0	0.0	0.00	3853	4.445077	20.957524	3.468607
Veracruz-Boca del Río	539	1578.942486	3071.750388	0	0.0	11.0	1151.50	15832	2.292915	4.788030	1.945448
Villahermosa	540	837.824074	954.982190	0	0.0	564.5	1366.50	5106	1.453375	2.533115	1.139836
Xalapa	502	364.589641	672.674516	0	5.0	61.5	351.50	3725	2.594310	6.764902	1.845018
Zacatecas	538	442.027881	638.445815	0	0.0	53.0	794.75	6707	3.048066	20.187534	1.444356

[64 rows x 11 columns]

[23]:
```
# Valores atípicos

def val_atipicos(df):
    col_num = df.select_dtypes(include='number').columns
```

```
    Q = df[col_num].quantile([0.25, 0.75])
    Q1 = Q.loc[0.25]
    Q3 = Q.loc[0.75]

    IQR = Q3 - Q1

    return ((df[col_num] < (Q1 - 1.5 * IQR)) |
            (df[col_num] > (Q3 + 1.5 * IQR))).sum()


val_atipicos(df_final)
```

[23]:
```
cuartos_disponibles               3440
cuartos_ocupados_no_residentes    4480
cuartos_ocupados_residentes       3625
llegada_turistas_no_residentes    4689
llegada_turistas_residentes       3165
turistas_noche_no_residentes      4554
turistas_noche_residentes         3612
dtype: int64
```

[24]:
```
columnas_numericas =␣
 ↪['cuartos_disponibles','cuartos_ocupados_no_residentes','cuartos_ocupados_residentes','lleg
   ␣
 ↪'llegada_turistas_residentes','turistas_noche_no_residentes','turistas_noche_residentes']


for var in columnas_numericas:
    df_final.boxplot(column=var, by='categoria')
    plt.title(f'Diagrama de caja de {var} por categoría')
    plt.suptitle('')
    plt.xlabel('Categoría')
    plt.ylabel(var)
    plt.xticks(rotation=45)
    plt.show()
```

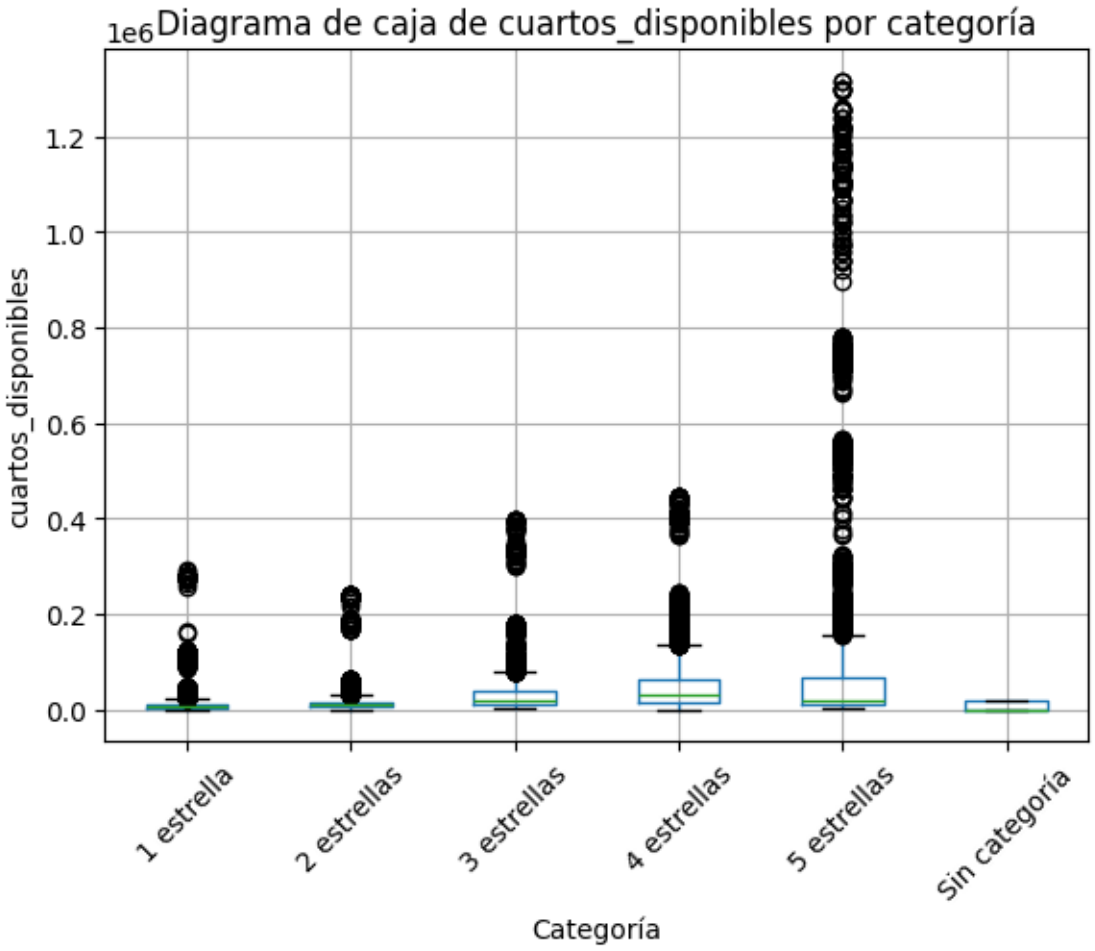


Diagrama de caja de cuartos_disponibles por categoría
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
cuartos_disponibles
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría

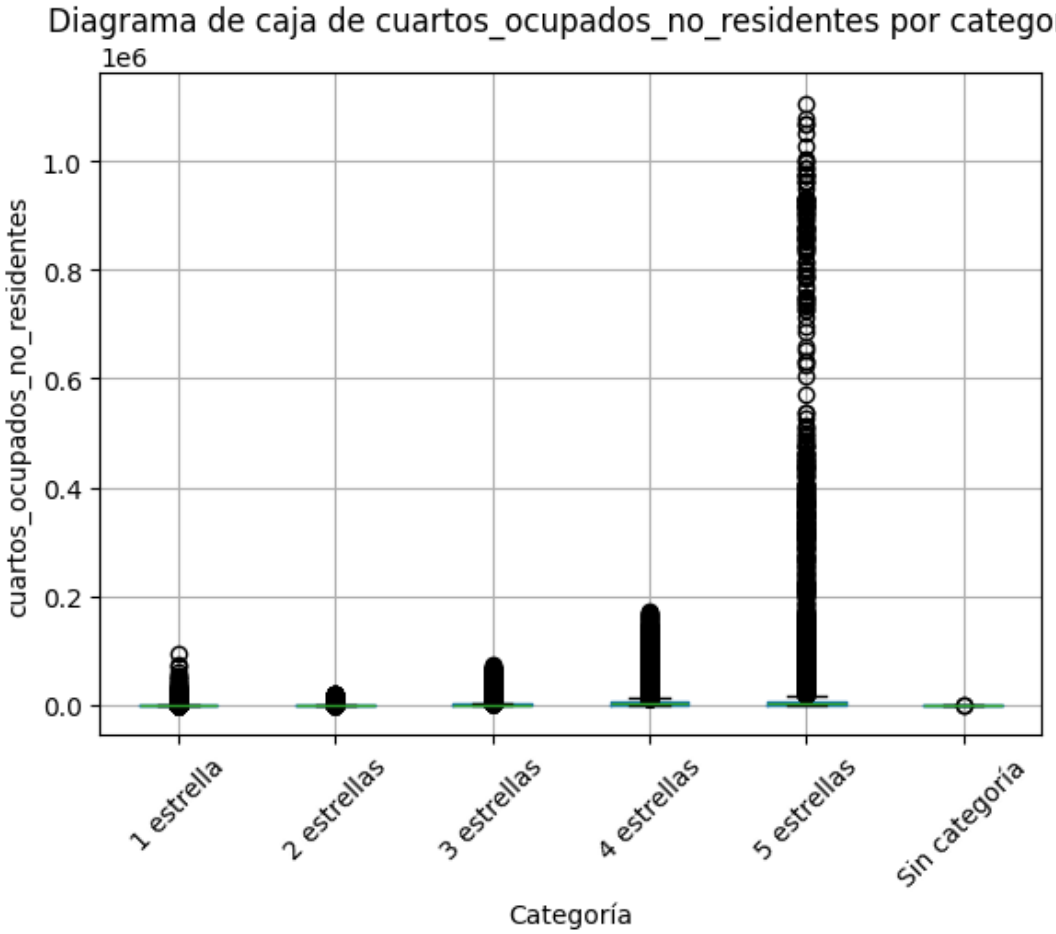


Diagrama de caja de cuartos_ocupados_no_residentes por categoría
1e6
1.0
0.8
0.6
0.4
0.2
0.0
cuartos_ocupados_no_residentes
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría

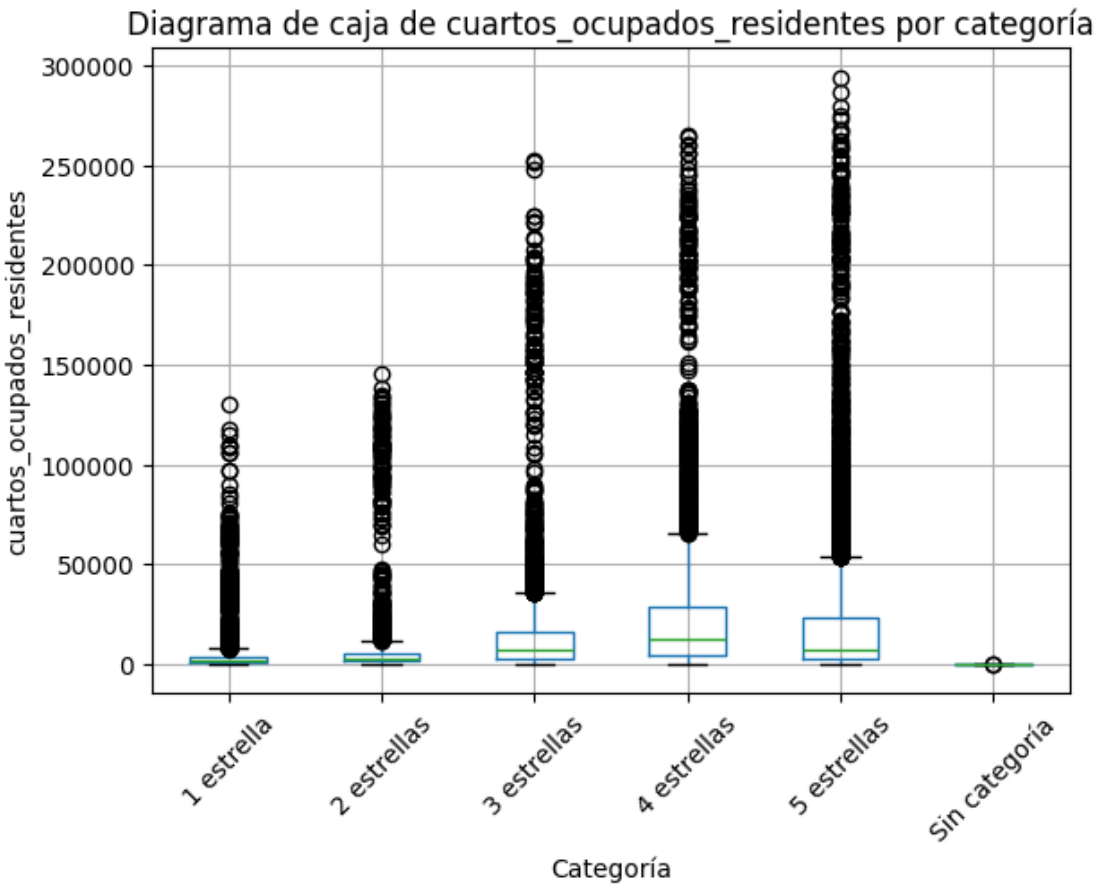


Diagrama de caja de cuartos_ocupados_residentes por categoría
cuartos_ocupados_residentes
300000
250000
200000
150000
100000
50000
0
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría

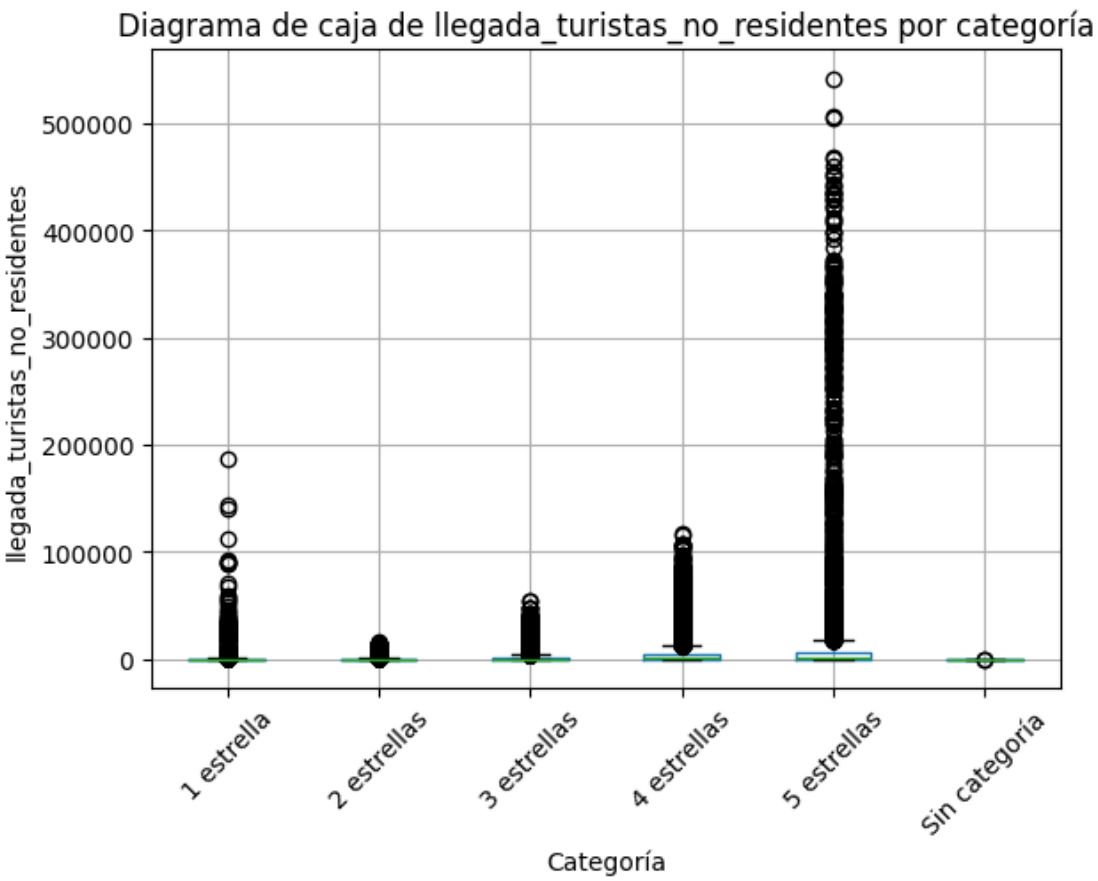


Diagrama de caja de llegada_turistas_no_residentes por categoría
llegada_turistas_no_residentes
500000
400000
300000
200000
100000
0
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría

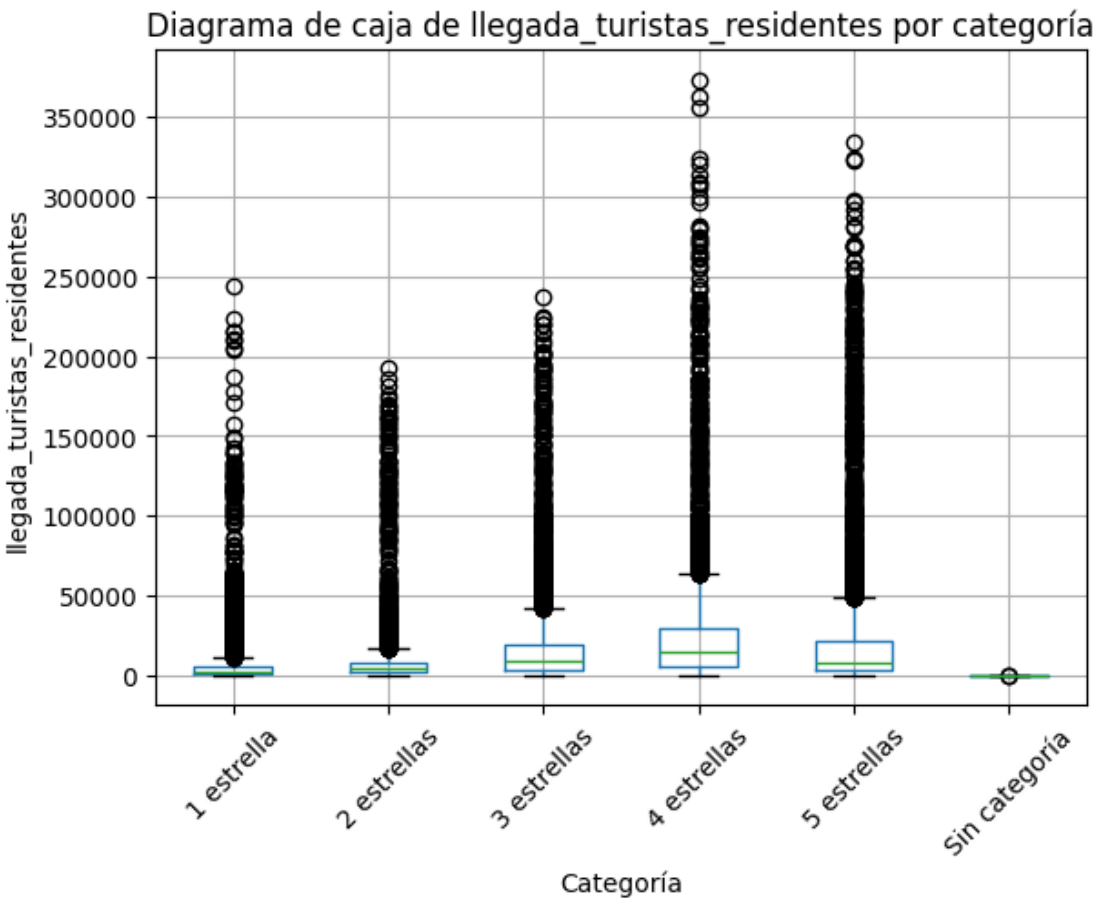


Diagrama de caja de llegada_turistas_residentes por categoría
llegada_turistas_residentes
350000
300000
250000
200000
150000
100000
50000
0
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría

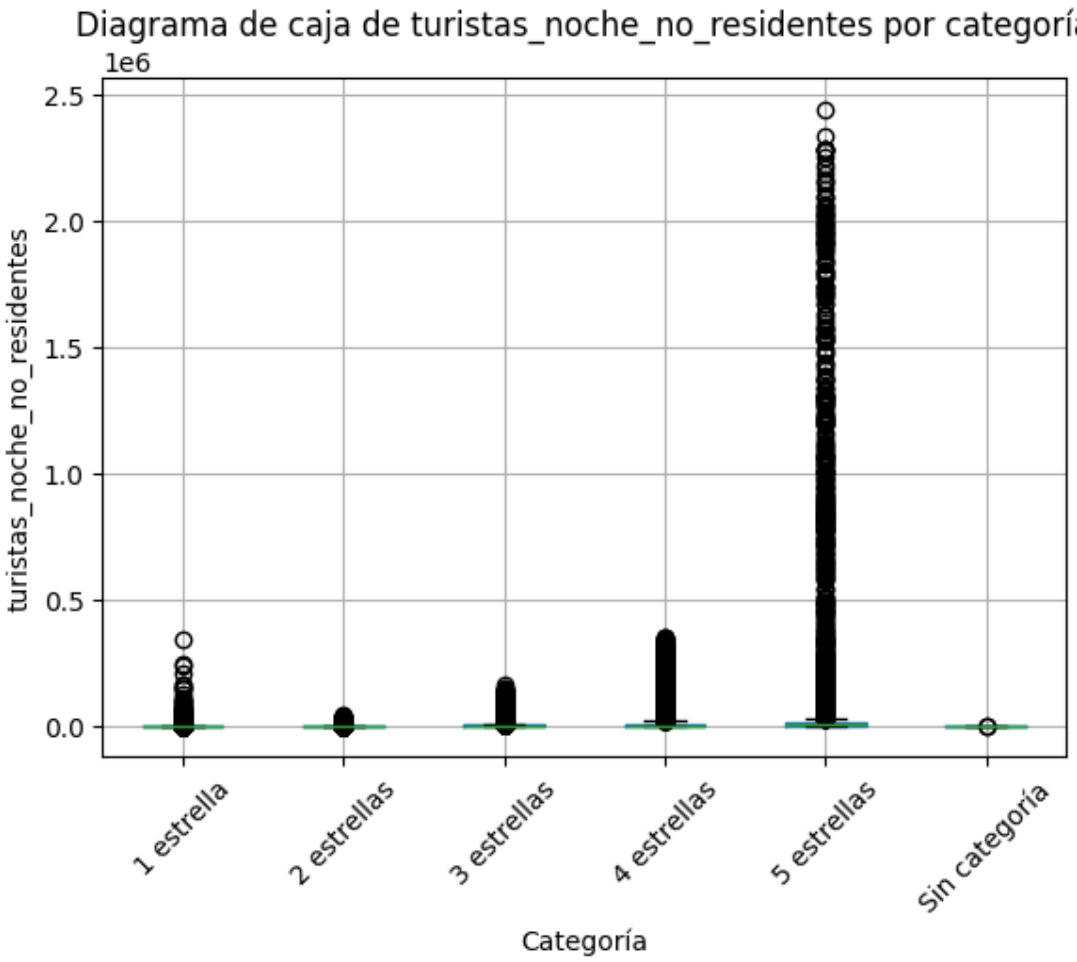
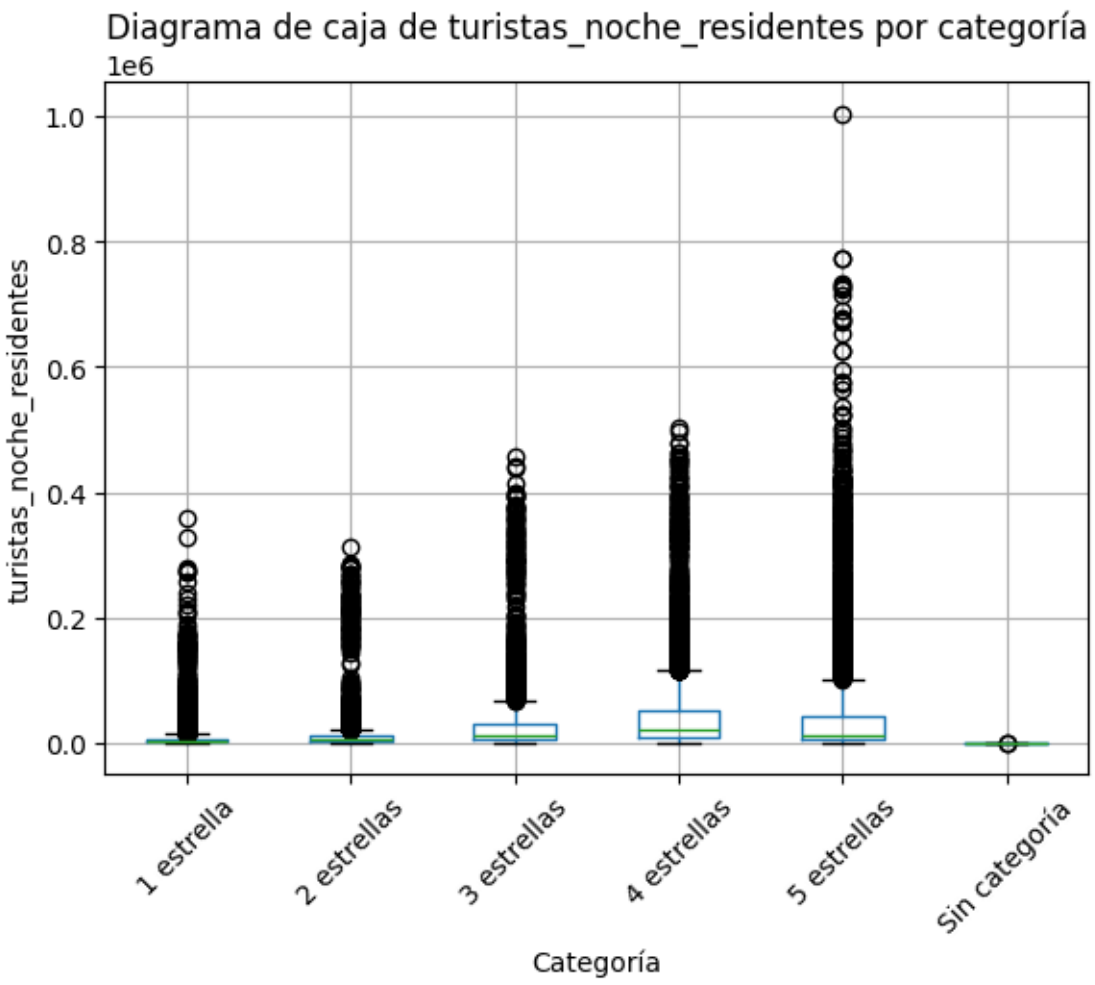
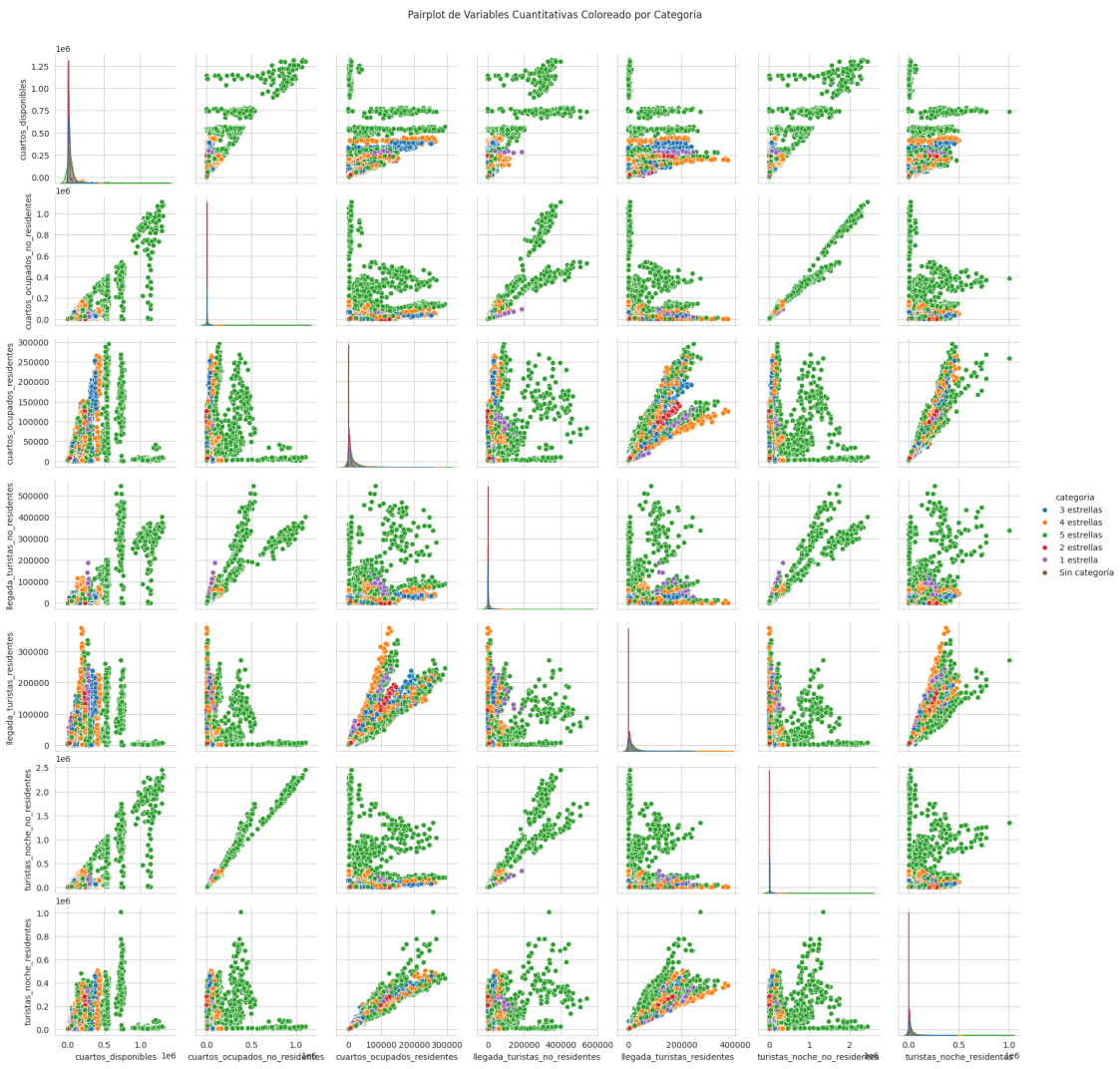


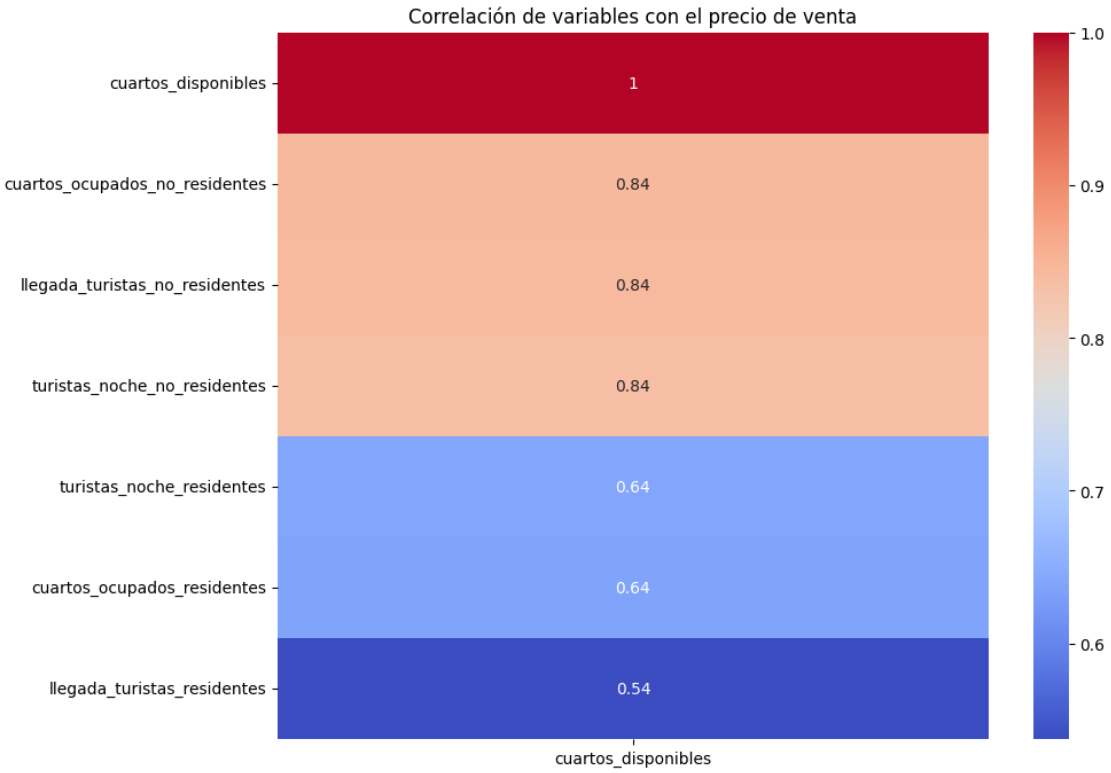
Diagrama de caja de turistas_noche_no_residentes por categoría
1e6
2.5
2.0
1.5
1.0
0.5
0.0
turistas_noche_no_residentes
1 estrella
2 estrellas
3 estrellas
4 estrellas
5 estrellas
Sin categoría
Categoría



[25]:
```
sns.set_style("whitegrid")

sns.pairplot(df_final,vars=columnas_numericas, hue='categoria', kind='scatter')

plt.suptitle("Pairplot de Variables Cuantitativas Coloreado por Categoria", y=1.
↪02)
plt.show()
```



[25]:
```
# Correlación
columnas_numericas_df= df_final[columnas_numericas]
corr = columnas_numericas_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr[['cuartos_disponibles']].sort_values(by='cuartos_disponibles',␣
 ↪ascending=False),
            annot=True, cmap='coolwarm')
plt.title("Correlación de variables con el precio de venta")
plt.show()
```



[36]:
```
df_playas = df_final[df_final['tipo_centro'] == 'Centros de Playa']
df_ciudades = df_final[df_final['tipo_centro'] == 'Ciudades']

print(df_playas['centro'].unique())
print(df_playas['llegada_turistas_no_residentes'].sum())
print(df_playas['llegada_turistas_residentes'].sum())
```

```
['Ixtapa - Zihuatanejo' 'Bahías de Huatulco' 'Cancún' 'Loreto' 'Los Cabos'
 'Isla Mujeres' 'Nuevo Nayarit' 'Playas de Rosarito' 'Puerto Escondido'
 'Riviera Maya' 'San Felipe' 'Tonalá Puerto Arista' 'Acapulco' 'Cozumel'
 'La Paz' 'Manzanillo' 'Mazatlán' 'Veracruz-Boca del Río'
 'Puerto Vallarta']
127335815
172848992
```

[37]:
```
print(df_ciudades['centro'].unique())
print(df_ciudades['llegada_turistas_no_residentes'].sum())
print(df_ciudades['llegada_turistas_residentes'].sum())
```

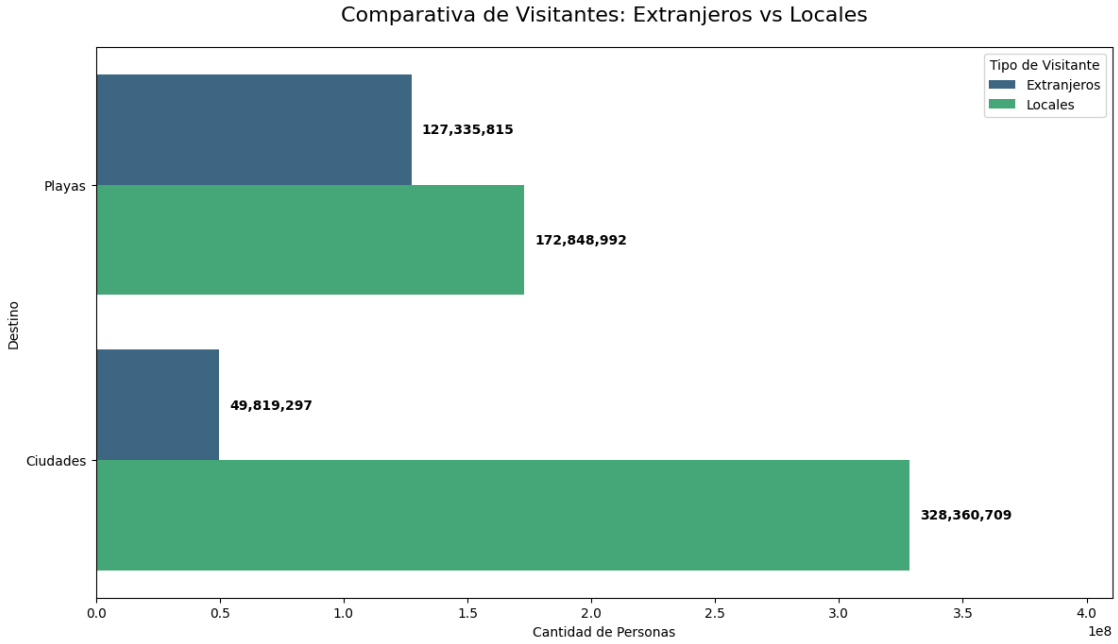
```
['Ciudad de México' 'Guadalajara' 'Monterrey' 'Aguascalientes' 'Campeche'
 'Celaya' 'Chihuahua' 'Coatzacoalcos' 'Colima' 'Comitán de Domínguez'
 'Culiacán' 'Durango' 'El Fuerte' 'Guanajuato' 'Hermosillo' 'Irapuato'
 'León' 'Los Mochis' 'Mérida' 'Morelia' 'Oaxaca' 'Pachuca' 'Palenque'
```

```
 'Puebla' 'Querétaro' 'Salamanca' 'San Cristóbal de las Casas'
 'San Juan de los Lagos' 'San Juan del Río' 'San Luis Potosí'
 'San Miguel de Allende' 'Taxco' 'Tequisquiapan' 'Tlaxcala' 'Toluca'
 'Tuxtla Gutiérrez' 'Valle de Bravo' 'Villahermosa' 'Xalapa' 'Zacatecas'
 'Ciudad Juárez' 'Mexicali' 'Piedras Negras' 'Tecate' 'Tijuana']
49819297
328360709
```

[38]:
```
# Módulos
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import skew, kurtosis
from sklearn.linear_model import LinearRegression

datos = {'Destino': ['Playas', 'Playas', 'Ciudades', 'Ciudades'],'Tipo':␣
 ↪['Extranjeros', 'Locales', 'Extranjeros', 'Locales'],
    'Total': [df_playas['llegada_turistas_no_residentes'].sum(),␣
 ↪df_playas['llegada_turistas_residentes'].sum(),
    df_ciudades['llegada_turistas_no_residentes'].sum(),␣
 ↪df_ciudades['llegada_turistas_residentes'].sum()]}

df_plot = pd.DataFrame(datos)
```

[29]:
```
plt.figure(figsize=(12, 7))

ax = sns.barplot(data=df_plot, x='Total', y='Destino', hue='Tipo',␣
 ↪palette='viridis')


for container in ax.containers:

    valores_formateados = [f'{val:,.0f}' for val in container.datavalues]
    ax.bar_label(container, labels=valores_formateados, padding=8, fontsize=10,␣
 ↪weight='bold')


plt.title('Comparativa de Visitantes: Extranjeros vs Locales', fontsize=16,␣
 ↪pad=20)
plt.xlabel('Cantidad de Personas')
plt.ylabel('Destino')
plt.legend(title='Tipo de Visitante')
plt.savefig("heatmap_correlacion_turismo.png", dpi=300, bbox_inches="tight")
plt.xlim(0, df_plot['Total'].max() * 1.25)
```

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



[30]:
```
df_sin_clas_extranjeros = df_final[df_final['categoria'] == 'Sin␣
 ↪Categoria'][['llegada_turistas_no_residentes']]
df_sin_clas_locales = df_final[df_final['categoria'] == 'Sin␣
 ↪Categoria'][['llegada_turistas_residentes']]

df_clas1_extranjeros = df_final[df_final['categoria'] == '1␣
 ↪estrella'][['llegada_turistas_no_residentes']]
df_clas1_locales = df_final[df_final['categoria'] == '1␣
 ↪estrella'][['llegada_turistas_residentes']]

df_clas2_extranjeros = df_final[df_final['categoria'] == '2␣
 ↪estrellas'][['llegada_turistas_no_residentes']]
df_clas2_locales = df_final[df_final['categoria'] == '2␣
 ↪estrellas'][['llegada_turistas_residentes']]

df_clas3_extranjeros = df_final[df_final['categoria'] == '3␣
 ↪estrellas'][['llegada_turistas_no_residentes']]
df_clas3_locales = df_final[df_final['categoria'] == '3␣
 ↪estrellas'][['llegada_turistas_residentes']]

df_clas4_extranjeros = df_final[df_final['categoria'] == '4␣
 ↪estrellas'][['llegada_turistas_no_residentes']]
```

```
df_clas4_locales = df_final[df_final['categoria'] == '4␣
 ↪estrellas'][['llegada_turistas_residentes']]

df_clas5_extranjeros= df_final[df_final['categoria'] == '5␣
 ↪estrellas'][['llegada_turistas_no_residentes']]
df_clas5_locales= df_final[df_final['categoria'] == '5␣
 ↪estrellas'][['llegada_turistas_residentes']]
```

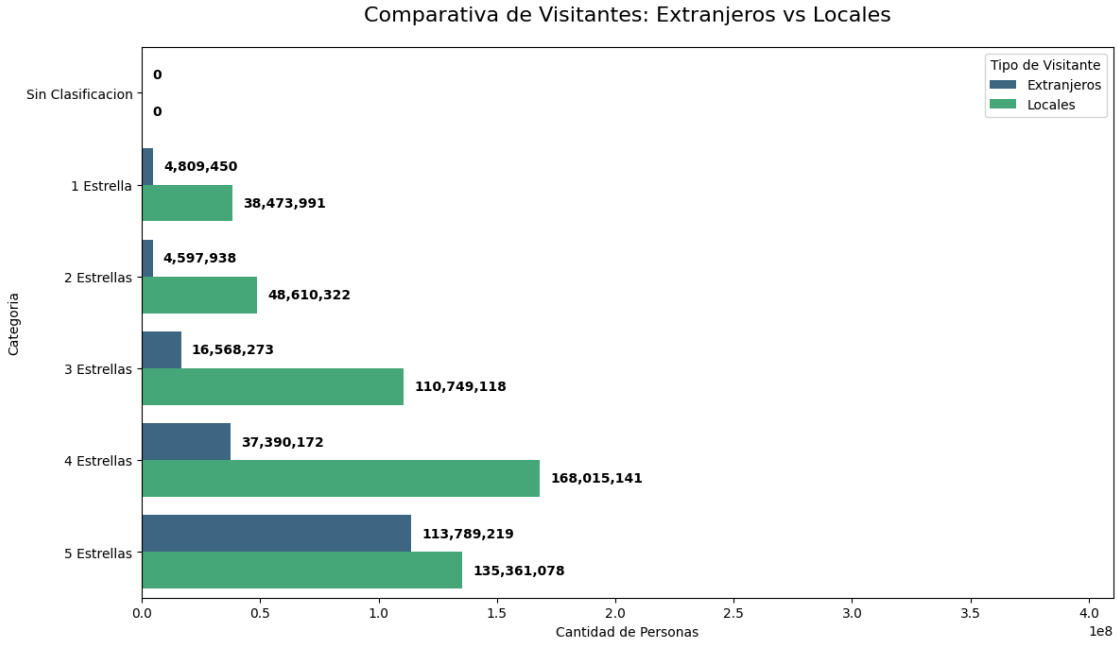
[31]:
```
df_clas1_extranjeros['llegada_turistas_no_residentes'].sum()
df_clas1_extranjeros.sum()
```

[31]:
```
llegada_turistas_no_residentes    4809450
dtype: int64
```

[32]:
```
datos2 = {'Destino': ['Sin Clasificacion', 'Sin Clasificacion', '1 Estrella',␣
 ↪'1 Estrella', '2 Estrellas', '2 Estrellas',
                      '3 Estrellas', '3 Estrellas','4 Estrellas', '4␣
 ↪Estrellas','5 Estrellas', '5 Estrellas'],
        'Tipo': ['Extranjeros', 'Locales', 'Extranjeros',␣
 ↪'Locales','Extranjeros', 'Locales', 'Extranjeros', 'Locales','Extranjeros'
                  , 'Locales', 'Extranjeros', 'Locales'],
    'Total': [df_sin_clas_extranjeros.sum().iloc[0], df_sin_clas_locales.sum().
 ↪iloc[0]
             ,df_clas1_extranjeros.sum().iloc[0], df_clas1_locales.sum().iloc[0]
             ,df_clas2_extranjeros.sum().iloc[0], df_clas2_locales.sum().iloc[0]
             ,df_clas3_extranjeros.sum().iloc[0], df_clas3_locales.sum().iloc[0]
             ,df_clas4_extranjeros.sum().iloc[0], df_clas4_locales.sum().iloc[0]
             ,df_clas5_extranjeros.sum().iloc[0], df_clas5_locales.sum().
 ↪iloc[0]]}

df_plot2 = pd.DataFrame(datos2)
```

[33]:
```
plt.figure(figsize=(12, 7))

ax = sns.barplot(data=df_plot2, x='Total', y='Destino', hue='Tipo',␣
 ↪palette='viridis')


for container in ax.containers:

    valores_formateados = [f'{val:,.0f}' for val in container.datavalues]
    ax.bar_label(container, labels=valores_formateados, padding=8, fontsize=10,␣
 ↪weight='bold')


plt.title('Comparativa de Visitantes: Extranjeros vs Locales', fontsize=16,␣
 ↪pad=20)
```

```
plt.xlabel('Cantidad de Personas')
plt.ylabel('Categoria')
plt.legend(title='Tipo de Visitante')

plt.xlim(0, df_plot['Total'].max() * 1.25)

plt.tight_layout()
plt.savefig("heatmap_correlacion_turismo.png", dpi=300, bbox_inches="tight")
plt.show()
```



[34]:
```
#Coeficientes de corelacion
vars_num = [
    'cuartos_disponibles',
    'cuartos_ocupados_no_residentes',
    'cuartos_ocupados_residentes',
    'llegada_turistas_no_residentes',
    'llegada_turistas_residentes',
    'turistas_noche_no_residentes',
    'turistas_noche_residentes'
]

corr = df_final[vars_num].corr()
corr
```

[34]:

	cuartos_disponibles
cuartos_disponibles	1.000000
cuartos_ocupados_no_residentes	0.843866
cuartos_ocupados_residentes	0.636859
llegada_turistas_no_residentes	0.841290
llegada_turistas_residentes	0.537180
turistas_noche_no_residentes	0.836172
turistas_noche_residentes	0.639473

	cuartos_ocupados_no_residentes
cuartos_disponibles	0.843866
cuartos_ocupados_no_residentes	1.000000
cuartos_ocupados_residentes	0.258059
llegada_turistas_no_residentes	0.914431
llegada_turistas_residentes	0.164491
turistas_noche_no_residentes	0.990993
turistas_noche_residentes	0.282691

	cuartos_ocupados_residentes
cuartos_disponibles	0.636859
cuartos_ocupados_no_residentes	0.258059
cuartos_ocupados_residentes	1.000000
llegada_turistas_no_residentes	0.400446
llegada_turistas_residentes	0.898223
turistas_noche_no_residentes	0.260590
turistas_noche_residentes	0.962590

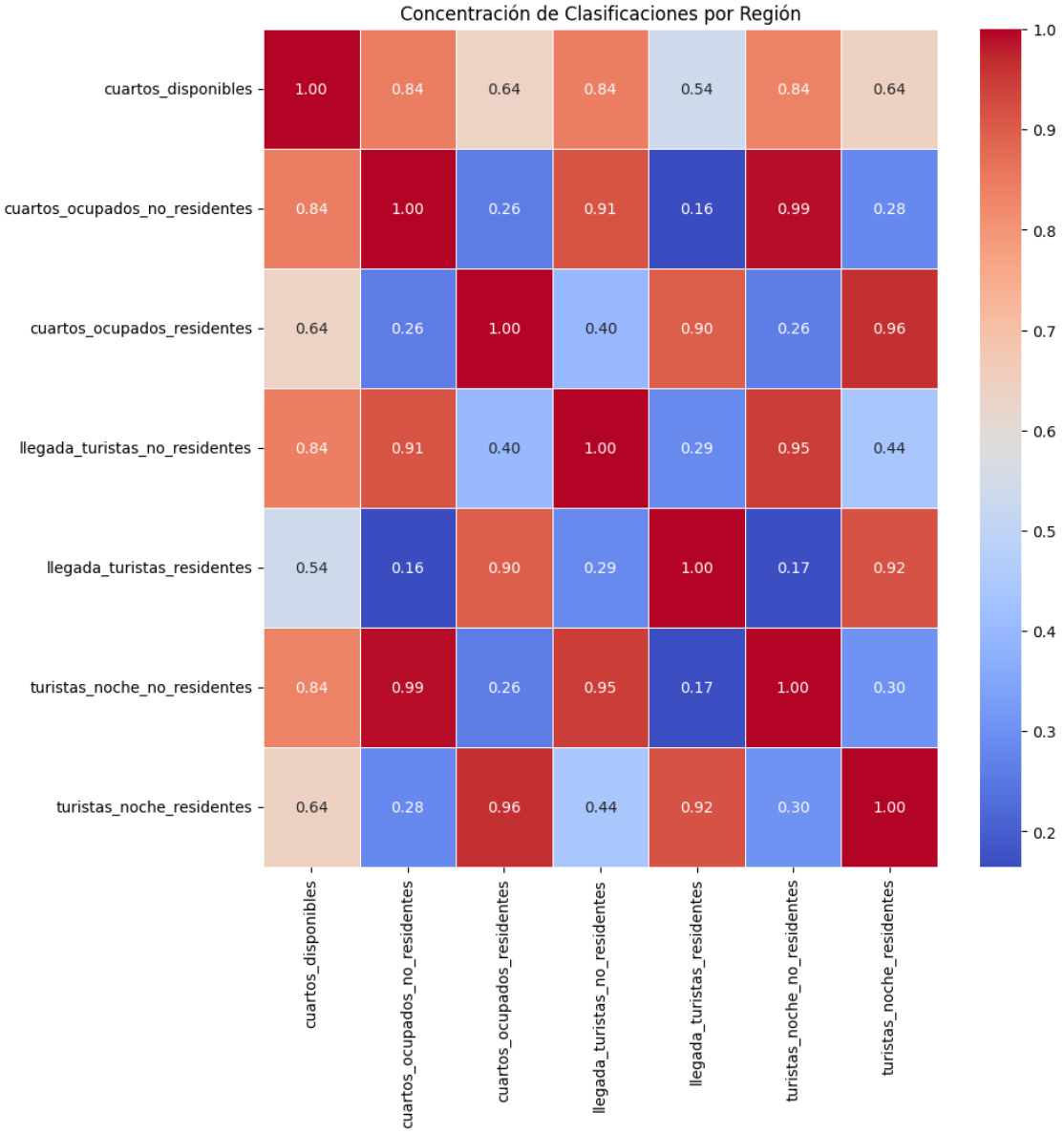
	llegada_turistas_no_residentes
cuartos_disponibles	0.841290
cuartos_ocupados_no_residentes	0.914431
cuartos_ocupados_residentes	0.400446
llegada_turistas_no_residentes	1.000000
llegada_turistas_residentes	0.288384
turistas_noche_no_residentes	0.947079
turistas_noche_residentes	0.441504

	llegada_turistas_residentes
cuartos_disponibles	0.537180
cuartos_ocupados_no_residentes	0.164491
cuartos_ocupados_residentes	0.898223
llegada_turistas_no_residentes	0.288384
llegada_turistas_residentes	1.000000
turistas_noche_no_residentes	0.169260
turistas_noche_residentes	0.916931

	turistas_noche_no_residentes
cuartos_disponibles	0.836172

```
cuartos_ocupados_no_residentes                    0.990993
cuartos_ocupados_residentes                       0.260590
llegada_turistas_no_residentes                    0.947079
llegada_turistas_residentes                       0.169260
turistas_noche_no_residentes                      1.000000
turistas_noche_residentes                         0.302458

                                  turistas_noche_residentes
cuartos_disponibles                                0.639473
cuartos_ocupados_no_residentes                     0.282691
cuartos_ocupados_residentes                        0.962590
llegada_turistas_no_residentes                     0.441504
llegada_turistas_residentes                        0.916931
turistas_noche_no_residentes                       0.302458
turistas_noche_residentes                          1.000000
```

[35]:
```
plt.figure(figsize=(10, 10))
sns.heatmap(corr, annot=True, fmt=".2f", linewidths=.5,cmap="coolwarm")
plt.title("Concentración de Clasificaciones por Región")
plt.savefig("heatmap_correlacion_turismo.png", dpi=300, bbox_inches="tight")
plt.show()
```



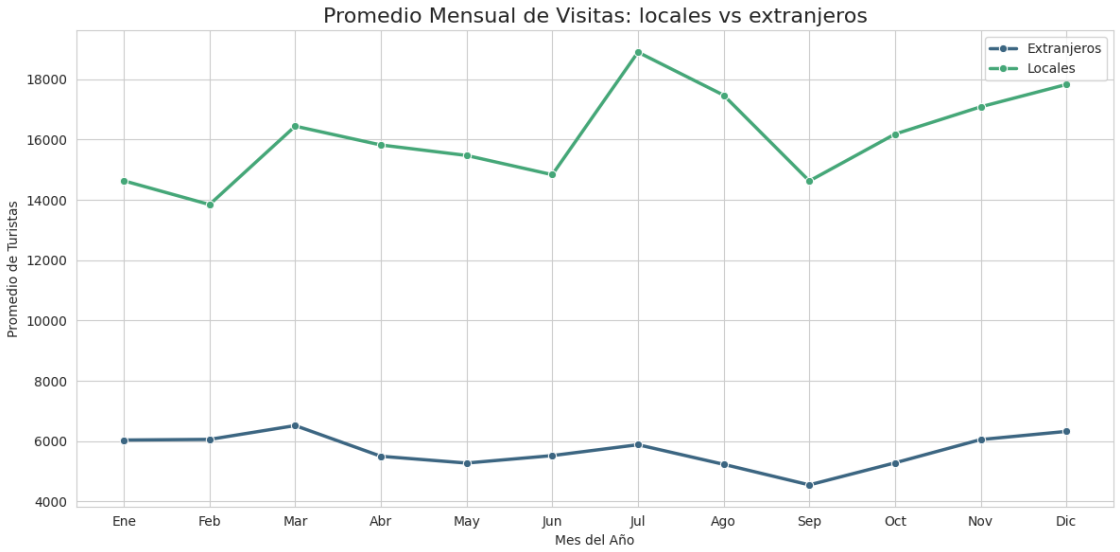
[39]:
```
df_final['anio'] = df_final['fecha'].dt.year
df_final['mes'] = df_final['fecha'].dt.month
```

[41]:
```
tabla = df_final.
↪pivot_table(index='anio',values='llegada_turistas_no_residentes',aggfunc='sum')
```

[43]:
```
tabla2 = df_final.
↪pivot_table(index='anio',values='llegada_turistas_residentes',aggfunc='sum')
```

[83]:
```
promo_extranjeros = df_final.groupby('mes')['llegada_turistas_no_residentes'].
↪mean()
promo_locales = df_final.groupby('mes')['llegada_turistas_residentes'].mean()
```

[84]:
```
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")


meses = ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct',␣
↪'Nov', 'Dic']


sns.lineplot(x=meses, y=promo_extranjeros.values, marker='o',␣
↪label='Extranjeros', linewidth=2.5 , color='#3c6682')
sns.lineplot(x=meses, y=promo_locales.values, marker='o', label='Locales',␣
↪linewidth=2.5 ,color='#45a778')


plt.title('Promedio Mensual de Visitas: locales vs extranjeros', fontsize=16)
plt.ylabel('Promedio de Turistas')
plt.xlabel('Mes del Año')
plt.legend()
plt.tight_layout()
plt.savefig("linea_meses.png", dpi=300, bbox_inches="tight")
plt.show()
```



[100]:
```
#interbalos de confianza
```

```
#variables agrupadas por mes para cuartos_disponibles,␣
 ↪cuartos_ocupados_no_residentes, llegada_turistas_no_residentes,
#turistas_noche_no residentes.
x2 = df_final['llegada_turistas_no_residentes']

x = x2.values
mean = x.mean()
se = x.std(ddof=1) / np.sqrt(len(x))

ic = stats.t.interval(0.95, df=len(x)-1, loc=mean, scale=se)
ic
```

[100]: (np.float64(5376.601400300798), np.float64(5999.184382785313))

[93]:
```
x3 = df_final['cuartos_ocupados_no_residentes']

x = x3.values
mean = x.mean()
se = x.std(ddof=1) / np.sqrt(len(x))

ic = stats.t.interval(0.95, df=len(x)-1, loc=mean, scale=se)
ic
```

[93]: (np.float64(8706.622122815212), np.float64(9980.536099749484))

[95]:
```
x3 = df_final['turistas_noche_no_residentes']

x = x3.values
mean = x.mean()
se = x.std(ddof=1) / np.sqrt(len(x))

ic = stats.t.interval(0.95, df=len(x)-1, loc=mean, scale=se)
ic
```

[95]: (np.float64(19352.503188580336), np.float64(22306.62928428937))

[96]:
```
x3 = df_final['cuartos_disponibles']

x = x3.values
mean = x.mean()
se = x.std(ddof=1) / np.sqrt(len(x))

ic = stats.t.interval(0.95, df=len(x)-1, loc=mean, scale=se)
ic
```

[96]: (np.float64(42552.446209666676), np.float64(44788.673356248684))

[109]:
```
X = df_final['cuartos_disponibles'].values.reshape(-1, 1)
y = df_final['cuartos_ocupados_no_residentes'].values


modelo_turismo = LinearRegression()
modelo_turismo.fit(X, y)


pendiente = modelo_turismo.coef_[0]
interseccion = modelo_turismo.intercept_

print(f"Pendiente (m): {pendiente:.4f}")
print(f"Intersección (b): {interseccion:.4f}")
```

```
Pendiente (m): 0.4807
Intersección (b): -11650.0003
```

[110]:
```
X = df_final['cuartos_disponibles'].values.reshape(-1, 1)
y = df_final['llegada_turistas_no_residentes'].values


modelo_turismo = LinearRegression()
modelo_turismo.fit(X, y)


pendiente = modelo_turismo.coef_[0]
interseccion = modelo_turismo.intercept_

print(f"Pendiente (m): {pendiente:.4f}")
print(f"Intersección (b): {interseccion:.4f}")
```

```
Pendiente (m): 0.2342
Intersección (b): -4540.7017
```

[111]:
```
X = df_final['cuartos_disponibles'].values.reshape(-1, 1)
y = df_final['turistas_noche_no_residentes'].values


modelo_turismo = LinearRegression()
modelo_turismo.fit(X, y)


pendiente = modelo_turismo.coef_[0]
interseccion = modelo_turismo.intercept_

print(f"Pendiente (m): {pendiente:.4f}")
print(f"Intersección (b): {interseccion:.4f}")
```

```
Pendiente (m): 1.1046
```

```
Intersección (b): -27409.3504
```

[ ]: