

03_analysis

September 14, 2026

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

```
[2]: #Importamos el tablón analítico desde la base de datos

eng = sqlalchemy.create_engine("sqlite:///../datos/intermedios/
↳mercado_inmobiliario.db")

with eng.connect() as con:
    df = pd.read_sql("SELECT * FROM tablon_analitico", eng)

df
```

```
[2]:
```

		name	neighbourhood \
0		Apartamentos Dana Sol	Sol
1		Apartasol Apartamentos Dana	Universidad
2		MAGIC ARTISTIC HOUSE IN THE CENTER OF MADRID	Justicia
3		Adorable Apartment Malasaña-Gran Vía	Universidad
4		Heart of Malasaña Cozy, Quiet & Sunny Apartment	Universidad
...		...	...
8268		Charming apartment near Santiago Bernabéu	Bellas Vistas
8269		habitación moderna en pleno centro	Valdeacederas
8270		Bonito apartamento a 5 minutos de Atocha.	San Diego
8271		Habitación cerca del aeropuerto, piso compartido!	Apostol Santiago
8272		Nuevo apartamento, vistas a Gran Vía.	Palacio

	latitude	longitude	room_type	precio_noche	minimum_nights \
0	40.414760	-3.704180	Entire home/apt	157.0	5
1	40.422470	-3.705770	Entire home/apt	143.0	5
2	40.418840	-3.696550	Private room	65.0	1
3	40.422520	-3.702500	Entire home/apt	116.0	30
4	40.422520	-3.702500	Entire home/apt	79.0	30
...	...	...	...	...	...
8268	40.450697	-3.704396	Entire home/apt	189.0	1
8269	40.460935	-3.698405	Private room	42.0	1

8270	40.393974	-3.667897	Entire home/apt	154.0	1
8271	40.474829	-3.661487	Private room	34.0	1
8272	40.422480	-3.710080	Entire home/apt	140.0	1

	availability_365		description \
0	342		NaN
1	341		NaN
2	299		INCREDIBLE HOME OF AN ARTIST SURROUNDED BY PAI...
3	305		Beautiful apartment with spacious living room ...
4	317		Cozy Apartment in great location in the center...
...	...		...
8268	314		A unique and charming accommodation with moder...
8269	326		This listing has a strategic location - it wil...
8270	344		Nice apartment with 2 bedrooms, a sofa bed in ...
8271	333		Looking for comfort and a good location during...
8272	337		Can you imagine staying in the center of Spain...

	accommodates	...	precio_noche_total	ingreso_anual	m2_estimado \
0	2	...	157.0	1570.0	50.0
1	2	...	143.0	5720.0	50.0
2	4	...	65.0	15990.0	50.0
3	2	...	116.0	6960.0	50.0
4	2	...	79.0	9480.0	50.0
...	...	...	...	...	...
8268	8	...	189.0	1134.0	65.0
8269	2	...	42.0	504.0	50.0
8270	6	...	154.0	924.0	65.0
8271	2	...	34.0	204.0	50.0
8272	4	...	140.0	840.0	50.0

	coste_adquisicion	atractivo_turistico	bedrooms_disc	bathrooms_disc \
0	290437.5	15.439820	1_hab	0_1_ban
1	290437.5	13.056790	1_hab	0_1_ban
2	290437.5	14.987660	1_hab	1.5-2_ban
3	290437.5	11.883364	1_hab	0_1_ban
4	290437.5	11.883364	1_hab	0_1_ban
...	...	...	...	...
8268	301762.5	0.000000	2_hab	0_1_ban
8269	232125.0	0.000000	1_hab	0_1_ban
8270	169650.0	0.000000	2_hab	0_1_ban
8271	205612.5	0.000000	1_hab	0_1_ban
8272	290437.5	12.185324	1_hab	0_1_ban

	accommodates_disc	beds_disc	margen_bruto
0	02_dos	2_camás	0.54
1	02_dos	3_camás	1.97
2	4_cuatro	2_camás	5.51

3	02_dos	1_cama	2.40
4	02_dos	2_camás	3.26
...	...	...	...
8268	5_cinco_mas	3_camás	0.38
8269	02_dos	1_cama	0.22
8270	5_cinco_mas	4_camás	0.54
8271	02_dos	1_cama	0.10
8272	4_cuatro	2_camás	0.29

[8273 rows x 27 columns]

Es momento de interrogar a los datos con las preguntas semilla del proyecto.

```
[3]: df.describe().T
```

```
[3]:
```

	count	mean	std	min \
latitude	8273.0	40.422423	0.023242	40.33140
longitude	8273.0	-3.693824	0.027328	-3.82997
precio_noche	8273.0	132.419316	92.250608	20.00000
minimum_nights	8273.0	5.994681	15.800381	1.00000
availability_365	8273.0	292.465490	48.480412	181.00000
accommodates	8273.0	3.300254	1.632901	1.00000
bathrooms	8273.0	1.243926	0.541302	0.00000
bedrooms	8273.0	1.317176	0.780855	0.00000
beds	8273.0	1.925420	1.116572	0.00000
review_scores_location	8273.0	4.719433	0.393151	1.00000
estimated_occupancy_l365d	8273.0	123.333495	92.801497	6.00000
Precio_m2	8273.0	7098.257343	1849.716190	3302.00000
precio_noche_total	8223.0	133.000790	92.205277	20.00000
ingreso_anual	8223.0	15894.127083	17003.210312	126.00000
m2_estimado	8273.0	58.357911	17.525255	50.00000
coste_adquisicion	8273.0	313001.030158	135293.654752	123825.00000
atractivo_turistico	8273.0	5.314309	6.122208	0.00000
margen_bruto	8223.0	5.013229	4.576717	0.05000

	25%	50%	75% \
latitude	40.40985	40.420300	40.432410
longitude	-3.70712	-3.700930	-3.685404
precio_noche	76.00000	112.000000	161.000000
minimum_nights	1.00000	1.000000	3.000000
availability_365	254.00000	302.000000	333.000000
accommodates	2.00000	3.000000	4.000000
bathrooms	1.00000	1.000000	1.000000
bedrooms	1.00000	1.000000	2.000000
beds	1.00000	2.000000	2.000000
review_scores_location	4.63000	4.840000	4.950000
estimated_occupancy_l365d	36.00000	102.000000	228.000000
Precio_m2	6190.00000	7745.000000	7745.000000

precio_noche_total	77.00000	113.000000	161.000000
ingreso_anual	3606.00000	10416.000000	22950.000000
m2_estimado	50.00000	50.000000	65.000000
coste_adquisicion	237450.00000	290437.500000	377568.750000
atractivo_turistico	0.00000	1.666192	10.455641
margen_bruto	1.35000	3.710000	7.590000

	max
latitude	4.051417e+01
longitude	-3.545904e+00
precio_noche	1.000000e+03
minimum_nights	3.650000e+02
availability_365	3.650000e+02
accommodates	8.000000e+00
bathrooms	5.000000e+00
bedrooms	5.000000e+00
beds	6.000000e+00
review_scores_location	5.000000e+00
estimated_occupancy_1365d	2.550000e+02
Precio_m2	1.084100e+04
precio_noche_total	1.000000e+03
ingreso_anual	2.101200e+05
m2_estimado	1.400000e+02
coste_adquisicion	1.138305e+06
atractivo_turistico	1.871343e+01
margen_bruto	5.299000e+01

[4]: *#¿Cuál es el precio medio?*

```
print(f"Precio medio {df.precio_noche.median()}?")

print(f"Rango precios ({df.precio_noche.min()}, {df.precio_noche.max()}?)")
```

Precio medio 112.0?

Rango precios (20.0, 1000.0)?

[5]: *#Ahora por distritos*

```
df.groupby("Localizacion")["precio_noche"].median().sort_values(ascending=False)
```

[5]: Localizacion

Salamanca	138.0
Centro	133.0
Chamartín	121.0
Chamberí	121.0
Retiro	119.5
Tetuán	103.0
Moncloa - Aravaca	102.0
Arganzuela	100.0

Hortaleza	91.0
Ciudad Lineal	89.0
Puente de Vallecas	83.5
Fuencarral - El Pardo	83.0
Barajas	81.0
Villa de Vallecas	79.0
Villaverde	78.0
San Blas - Canillejas	76.0
Carabanchel	70.5
Usera	61.0
Vicálvaro	61.0
Moratalaz	56.5
Latina	56.0

Name: precio_noche, dtype: float64

```
[6]: #Por barrios
df.groupby("neighbourhood")["precio_noche"].median().sort_values(ascending=False)
```

```
[6]: neighbourhood
Recoletos      202.5
Castellana     172.0
Goya           156.0
Colina         154.0
Sol            149.5
...
Rosas          37.0
Campamento    37.0
Pradolongo     36.0
Aeropuerto     35.0
Los Angeles    31.0
Name: precio_noche, Length: 124, dtype: float64
```

```
[7]: df.corr(numeric_only=True)["precio_noche"].sort_values(ascending=False)
```

```
[7]: precio_noche      1.000000
precio_noche_total   0.999320
accommodates         0.539335
coste_adquisicion    0.530201
ingreso_anual        0.515250
m2_estimado          0.467590
beds                 0.457172
bedrooms             0.441767
bathrooms            0.366179
margen_bruto         0.313534
Precio_m2            0.291846
atractivo_turistico  0.254210
review_scores_location 0.115647
```

```

availability_365      0.038361
latitude              0.035401
estimated_occupancy_1365d -0.059189
longitude             -0.071817
minimum_nights        -0.093532
Name: precio_noche, dtype: float64

```

```
[8]: df.neighbourhood.value_counts()
```

```

[8]: neighbourhood
Embajadores      910
Universidad       722
Palacio           630
Sol               504
Justicia          388
...
Aeropuerto        3
Corralesjos        2
Pavones            2
Horcajo            2
Valdemarín         1
Name: count, Length: 124, dtype: int64

```

```

[9]: df.groupby("Localizacion")["estimated_occupancy_1365d"].median().
      ↪sort_values(ascending=False)

```

```

[9]: Localizacion
Barajas           198.0
Centro            129.0
Arganzuela        114.0
Moncloa - Aravaca 111.0
Moratalaz         109.0
Villa de Vallecas 105.0
Ciudad Lineal     100.0
Salamanca         96.0
Latina            90.0
Chamberí          90.0
Retiro            84.0
Puente de Vallecas 79.0
Carabanchel       79.0
Usera             78.0
Chamartín         78.0
San Blas - Canillejas 66.0
Tetuán            64.0
Fuencarral - El Pardo 62.0
Vicálvaro         61.0
Hortaleza         60.0

```

```
Villaverde          50.0
Name: estimated_occupancy_l365d, dtype: float64
```

```
[10]: df.groupby("room_type")["precio_noche"].median()
```

```
[10]: room_type
Entire home/apt      128.0
Private room         52.0
Shared room          25.0
Name: precio_noche, dtype: float64
```

```
[11]: df.groupby("Localizacion")["coste_adquisicion"].median().rank()
```

```
[11]:               coste_adquisicion
Localizacion
Arganzuela          16.0
Barajas             10.0
Carabanchel          4.0
Centro              17.0
Chamartín           18.0
Chamberí            20.0
Ciudad Lineal       11.0
Fuencarral - El Pardo 13.0
Hortaleza           12.0
Latina              7.0
Moncloa - Aravaca    15.0
Moratalaz           9.0
Puente de Vallecas   2.0
Retiro              19.0
Salamanca           21.0
San Blas - Canillejas 8.0
Tetuán             14.0
Usera               3.0
Vicálvaro            6.0
Villa de Vallecas     5.0
Villaverde           1.0
```

```
[12]: df.groupby("neighbourhood")["coste_adquisicion", "precio_noche"].corr()
```

```
[12]:               coste_adquisicion  precio_noche
neighbourhood
Abrantes      coste_adquisicion      1.000000      0.080409
              precio_noche          0.080409      1.000000
Acacias       coste_adquisicion      1.000000      0.388626
              precio_noche          0.388626      1.000000
Adelfas       coste_adquisicion      1.000000     -0.040605
...           ...
```

Vinateros	precio_noche	NaN	1.000000
Vista Alegre	coste_adquisicion	1.000000	0.458741
	precio_noche	0.458741	1.000000
Zofío	coste_adquisicion	1.000000	0.080463
	precio_noche	0.080463	1.000000

[248 rows x 4 columns]

```
[13]: dims = ['neighbourhood', 'room_type', 'Localizacion', 'bedrooms_disc',
            ↪ 'bathrooms_disc', 'accommodates_disc', 'beds_disc']

metrics = ['precio_noche_total', 'estimated_occupancy_l365d', 'ingreso_anual',
            ↪ 'coste_adquisicion', 'margen_bruto']
```

```
[14]: minicubo = df[dims + metrics]

minicubo = minicubo.melt(id_vars=metrics, value_vars=dims)

minicubo
```

```
[14]:
```

	precio_noche_total	estimated_occupancy_l365d	ingreso_anual	\
0	157.0	10	1570.0	
1	143.0	40	5720.0	
2	65.0	246	15990.0	
3	116.0	60	6960.0	
4	79.0	120	9480.0	
...	...	...	...	
57906	189.0	6	1134.0	
57907	42.0	12	504.0	
57908	154.0	6	924.0	
57909	34.0	6	204.0	
57910	140.0	6	840.0	

	coste_adquisicion	margen_bruto	variable	value
0	290437.5	0.54	neighbourhood	Sol
1	290437.5	1.97	neighbourhood	Universidad
2	290437.5	5.51	neighbourhood	Justicia
3	290437.5	2.40	neighbourhood	Universidad
4	290437.5	3.26	neighbourhood	Universidad
...	...	...	...	...
57906	301762.5	0.38	beds_disc	3_camas
57907	232125.0	0.22	beds_disc	1_cama
57908	169650.0	0.54	beds_disc	4_camas
57909	205612.5	0.10	beds_disc	1_cama
57910	290437.5	0.29	beds_disc	2_camas

[57911 rows x 5 columns]

```
[15]: minicubo = minicubo.groupby(["variable", "value"])[metrics].agg("median").
      ↪reset_index()
minicubo
```

```
[15]:
```

	variable	value	precio_noche_total \
0	Localizacion	Arganzuela	100.0
1	Localizacion	Barajas	81.0
2	Localizacion	Carabanchel	73.0
3	Localizacion	Centro	133.0
4	Localizacion	Chamartín	121.0
..	...	...	...
161	neighbourhood	Vista Alegre	44.0
162	neighbourhood	Zofío	63.0
163	room_type	Entire home/apt	128.0
164	room_type	Private room	53.0
165	room_type	Shared room	25.0

	estimated_occupancy_l365d	ingreso_anual	coste_adquisicion	margen_bruto
0	114.0	11280.0	237975.0	4.360
1	198.0	10736.0	193912.5	5.400
2	79.0	4788.0	144075.0	3.070
3	129.0	16188.0	290437.5	5.055
4	78.0	9100.0	308325.0	2.510
..	...	...	...	...
161	81.0	2676.0	144075.0	1.855
162	54.0	3360.0	135975.0	2.450
163	112.0	14313.0	290437.5	4.690
164	78.0	4452.0	237975.0	1.975
165	24.0	1200.0	232125.0	0.610

[166 rows x 7 columns]

```
[16]: minicubo.query("variable == 'Localizacion'")
```

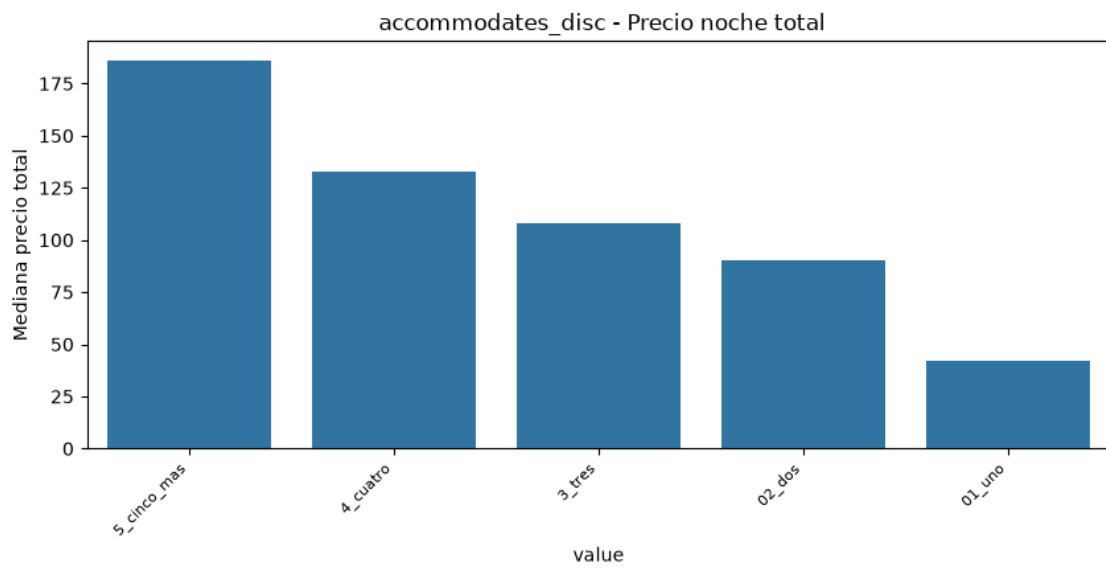
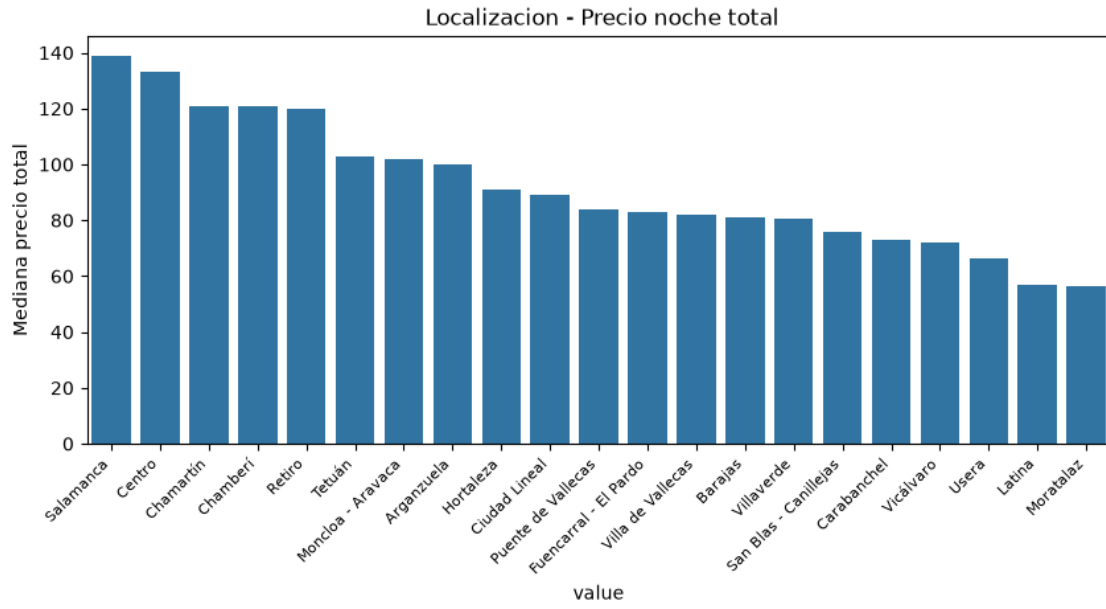
```
[16]:
```

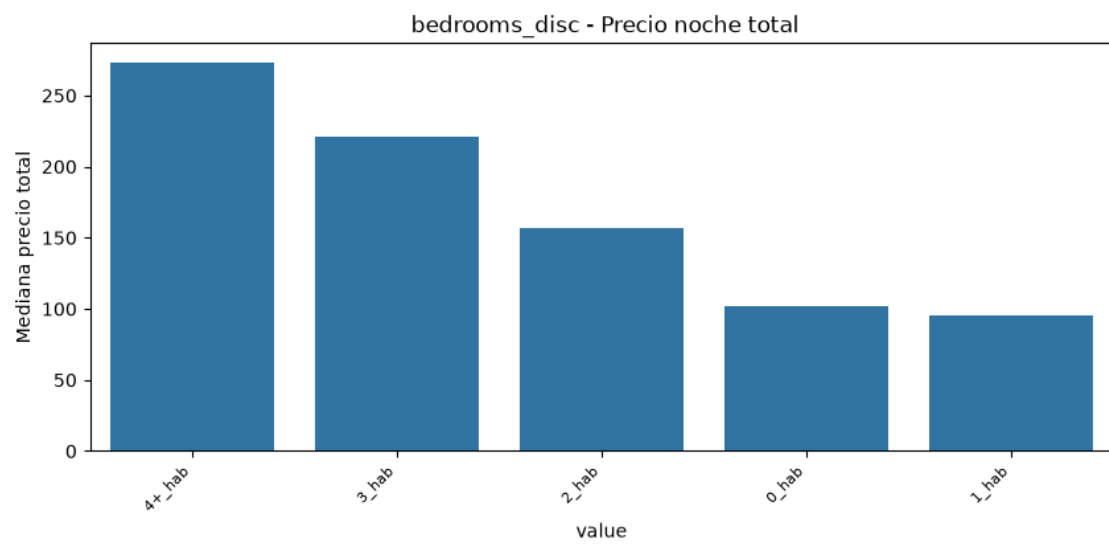
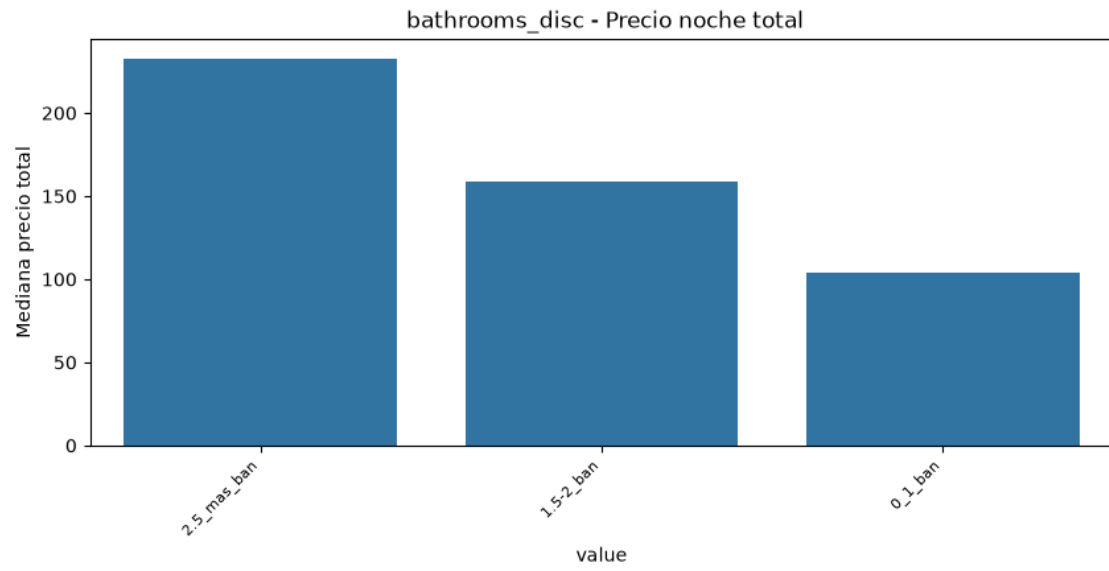
	variable	value	precio_noche_total \
0	Localizacion	Arganzuela	100.0
1	Localizacion	Barajas	81.0
2	Localizacion	Carabanchel	73.0
3	Localizacion	Centro	133.0
4	Localizacion	Chamartín	121.0
5	Localizacion	Chamberí	121.0
6	Localizacion	Ciudad Lineal	89.0
7	Localizacion	Fuencarral - El Pardo	83.0
8	Localizacion	Hortaleza	91.0
9	Localizacion	Latina	57.0
10	Localizacion	Moncloa - Aravaca	102.0
11	Localizacion	Moratalaz	56.5

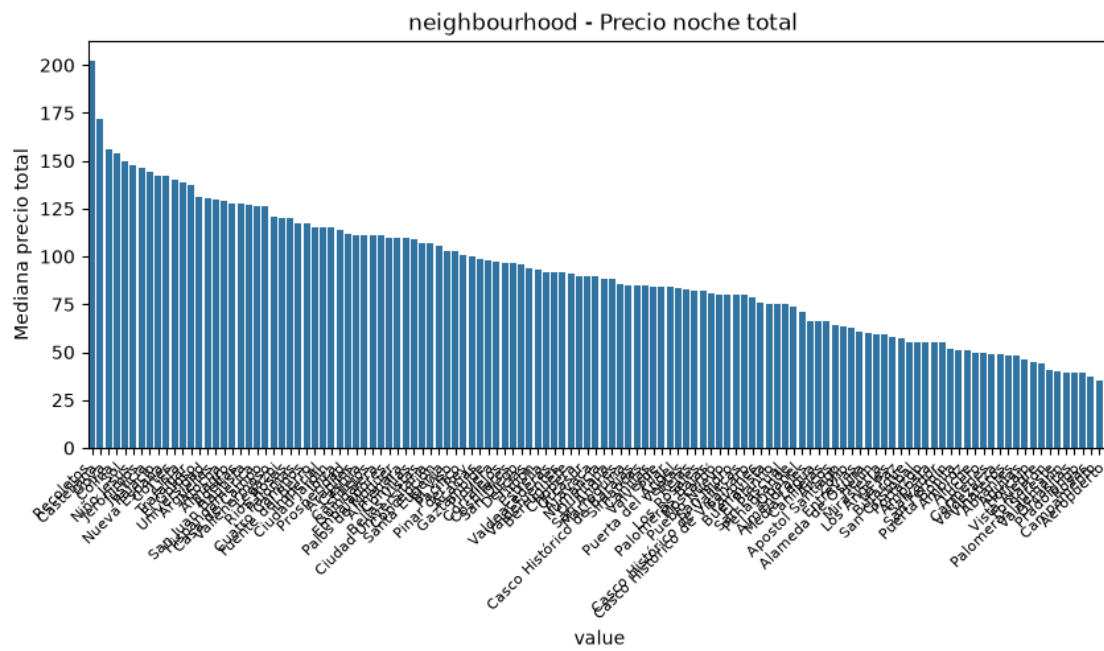
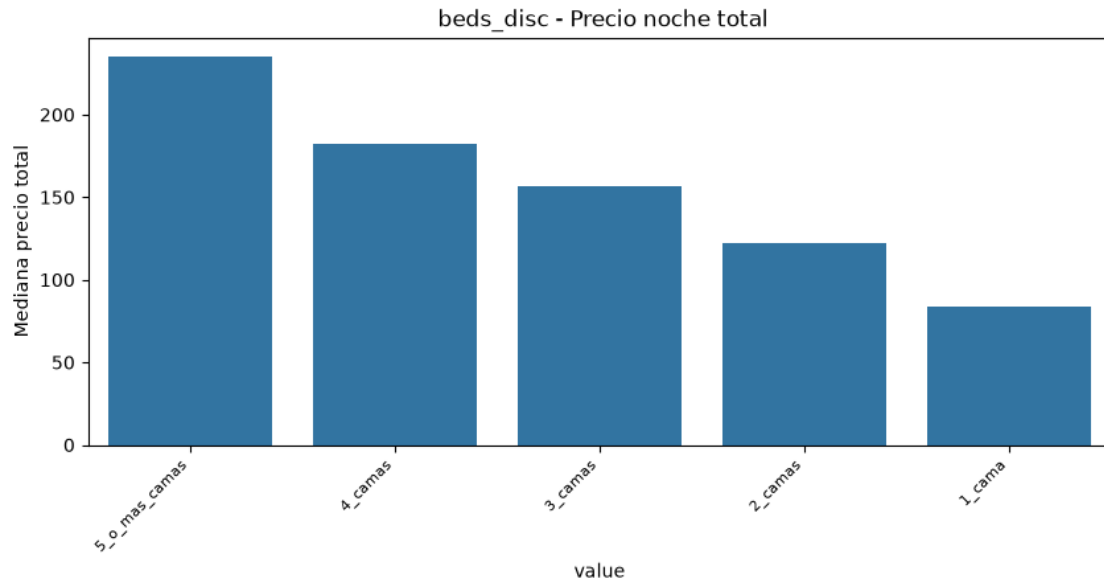
12	Localizacion	Puente de Vallecas	84.0
13	Localizacion	Retiro	120.0
14	Localizacion	Salamanca	139.0
15	Localizacion	San Blas - Canillejas	76.0
16	Localizacion	Tetuán	103.0
17	Localizacion	Usera	66.5
18	Localizacion	Vicálvaro	72.0
19	Localizacion	Villa de Vallecas	82.0
20	Localizacion	Villaverde	80.5

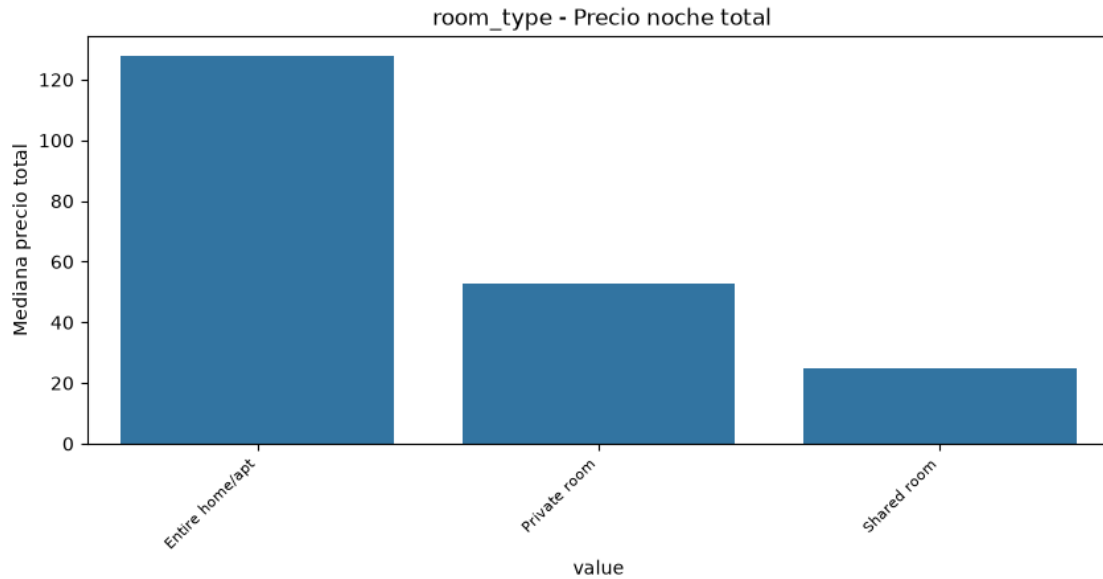
	estimated_occupancy_l365d	ingreso_anual	coste_adquisicion	margen_bruto
0	114.0	11280.0	237975.0	4.360
1	198.0	10736.0	193912.5	5.400
2	79.0	4788.0	144075.0	3.070
3	129.0	16188.0	290437.5	5.055
4	78.0	9100.0	308325.0	2.510
5	90.0	10488.0	345337.5	2.660
6	100.0	7344.0	195037.5	3.620
7	62.0	4815.0	210375.0	2.140
8	60.0	5280.0	205612.5	2.340
9	90.0	5130.0	151650.0	3.170
10	111.0	9870.0	237450.0	3.660
11	109.0	6820.0	171112.5	3.730
12	79.0	5580.0	130500.0	3.870
13	84.0	9964.5	316650.0	2.755
14	96.0	14025.0	406537.5	2.890
15	66.0	4602.0	154387.5	2.950
16	64.0	6087.0	232125.0	2.370
17	78.0	5844.0	135975.0	3.340
18	61.0	4080.0	149250.0	2.440
19	105.0	6375.0	144187.5	4.305
20	50.0	3723.0	123825.0	2.985

```
[17]: for dimension in minicubo["variable"].unique():
        data = minicubo[minicubo["variable"] == dimension].
        ↪sort_values("precio_noche_total", ascending = False)
        plt.figure(figsize=(10,4))
        sns.barplot(x=data["value"], y=data["precio_noche_total"])
        plt.title(f" {dimension} - Precio noche total")
        plt.ylabel("Mediana precio total")
        plt.xticks(rotation=45, ha="right", fontsize=8)
        plt.show()
```

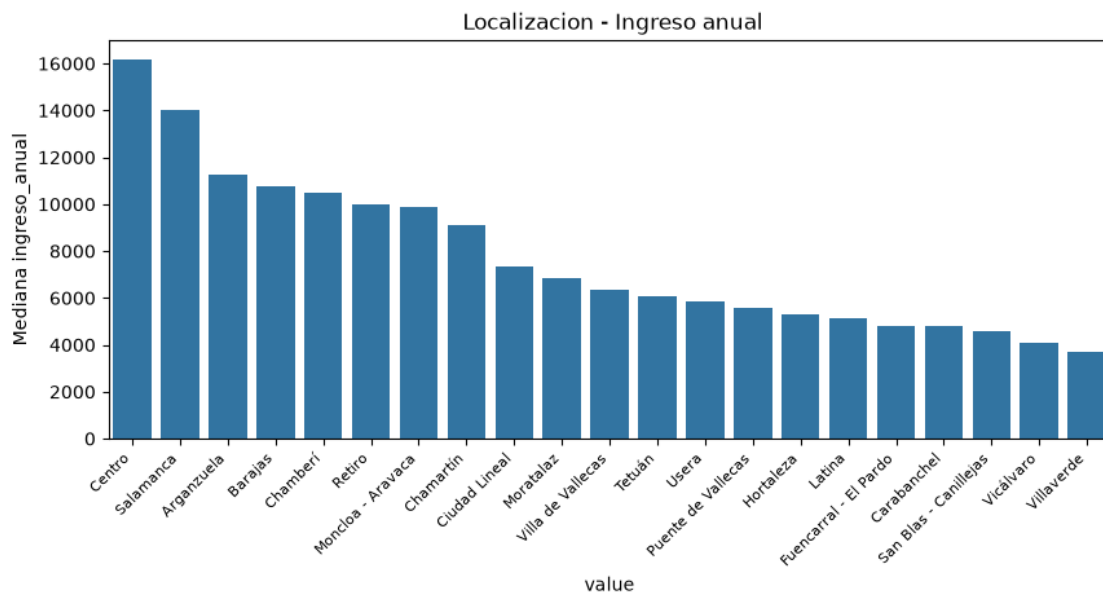


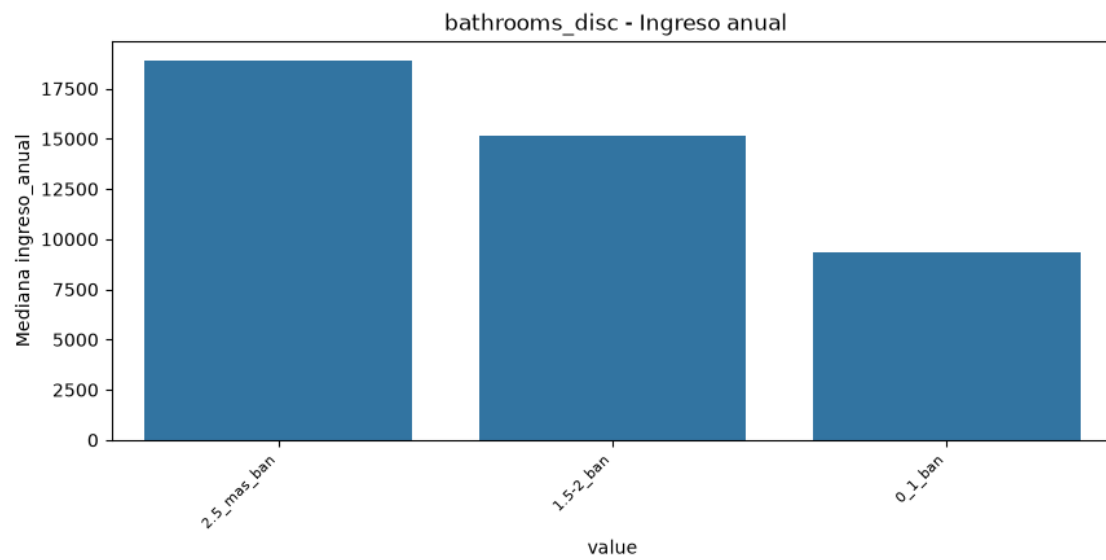
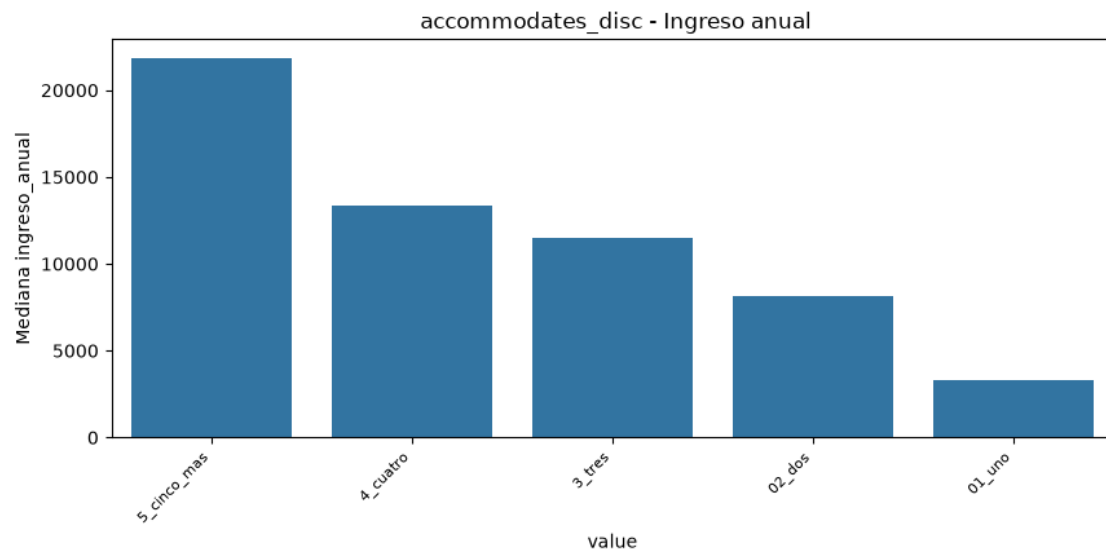


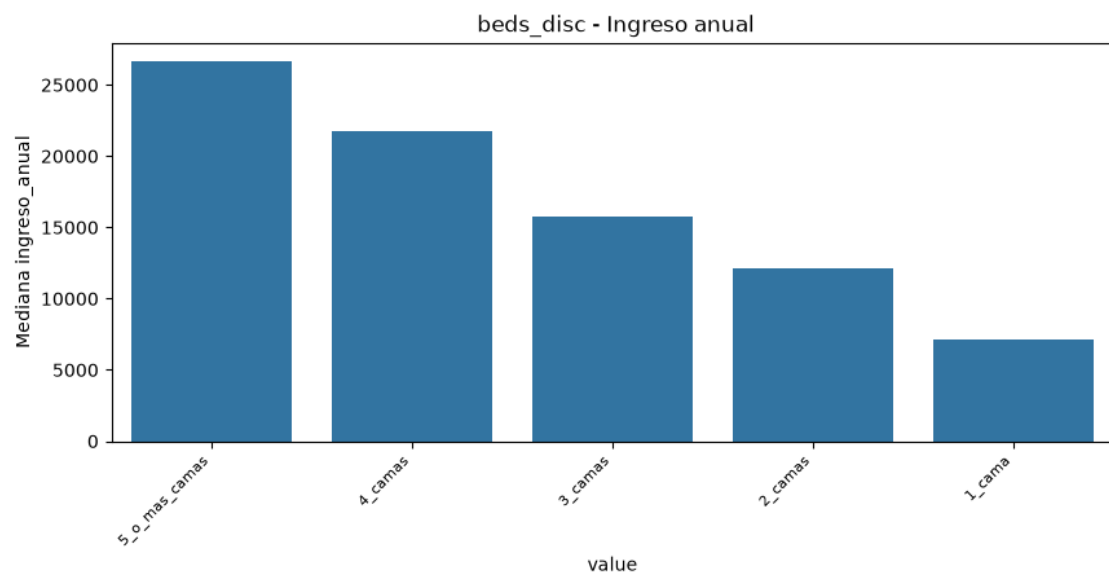
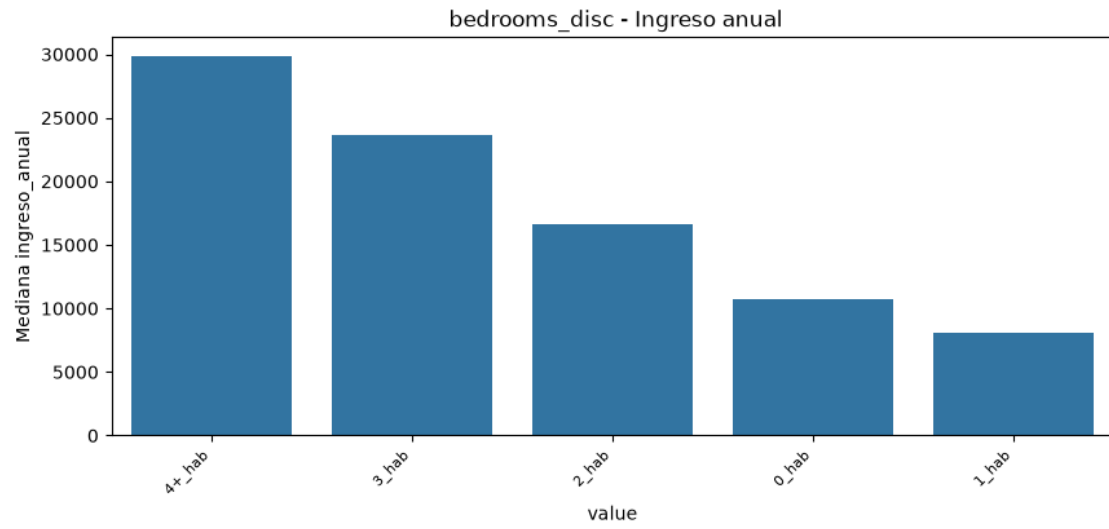


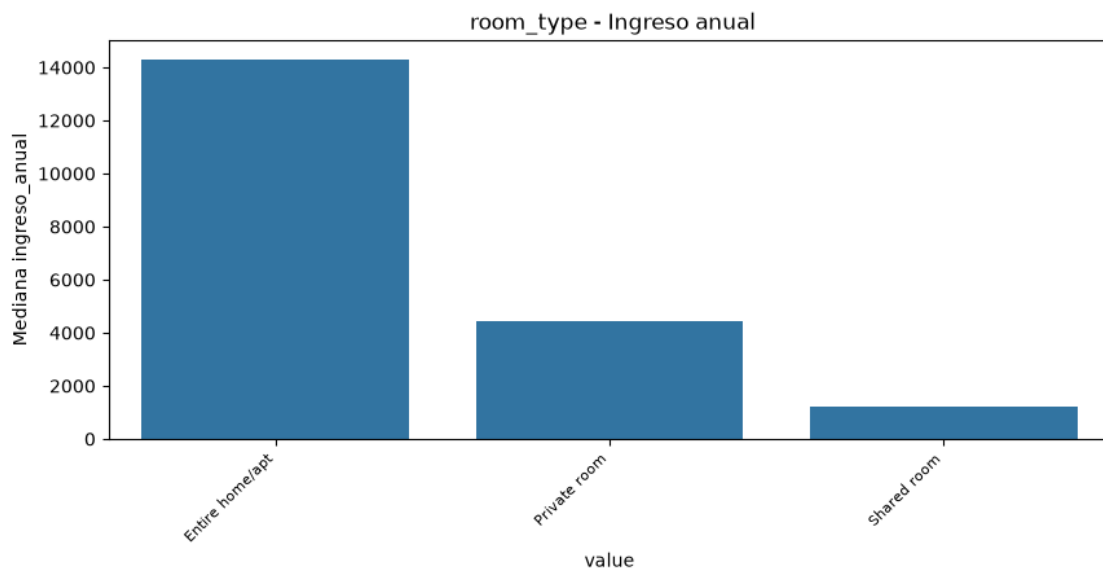
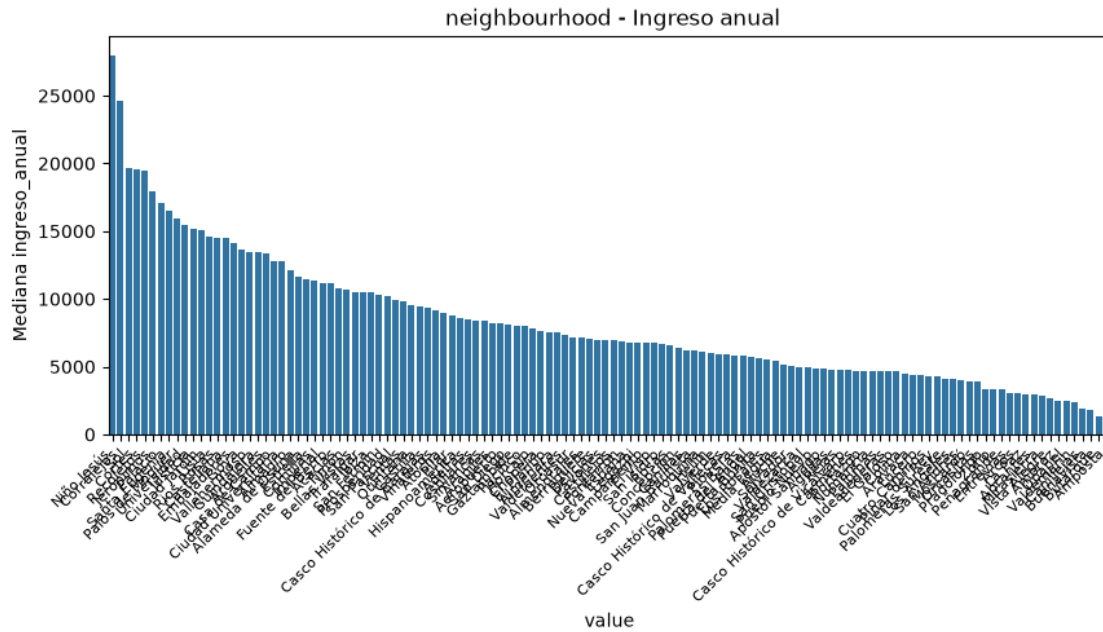


```
[18]: for dimension in minicubo["variable"].unique():
      data = minicubo[minicubo["variable"] == dimension].
      ↪sort_values("ingreso_anual", ascending = False)
      plt.figure(figsize=(10,4))
      sns.barplot(x=data["value"], y=data["ingreso_anual"])
      plt.title(f" {dimension} - Ingreso anual")
      plt.ylabel("Mediana ingreso_anual")
      plt.xticks(rotation=45, ha="right", fontsize=8)
      plt.show()
```



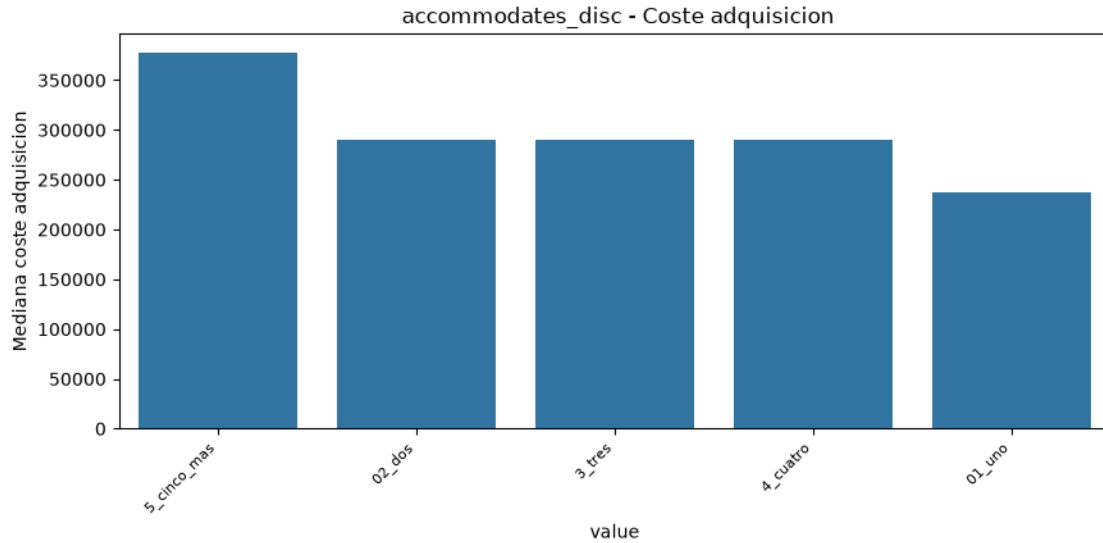
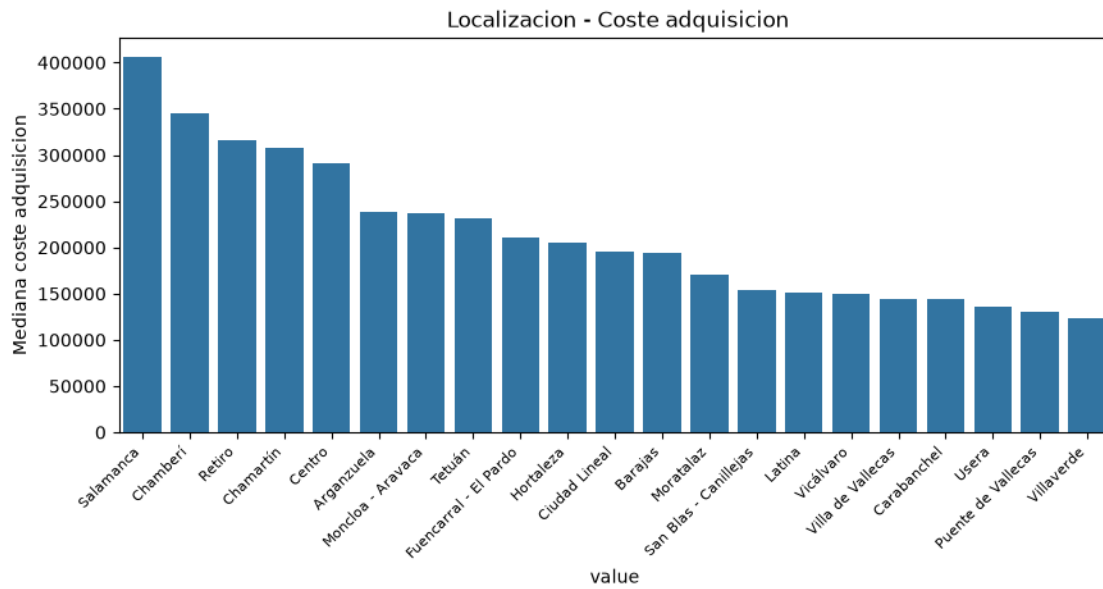


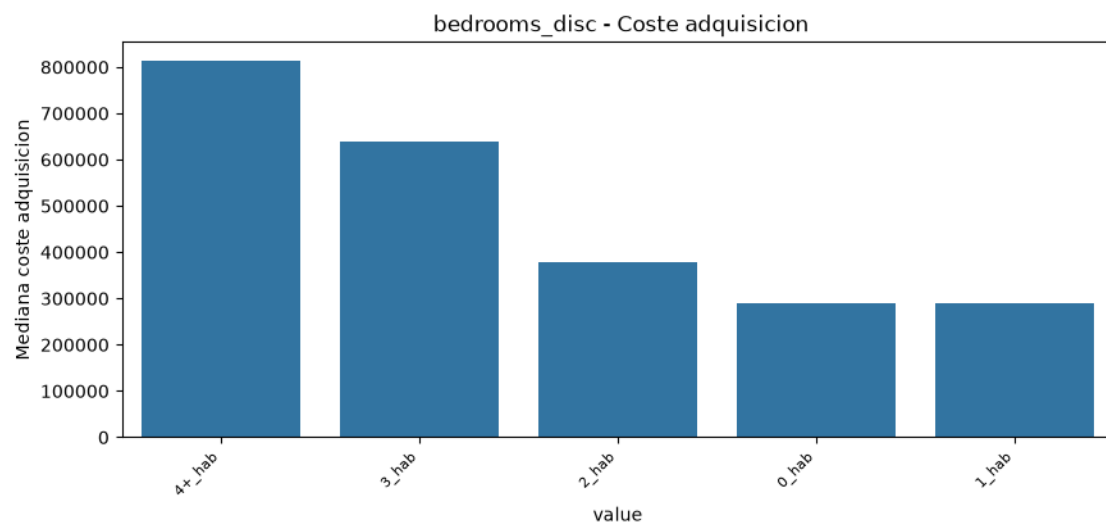
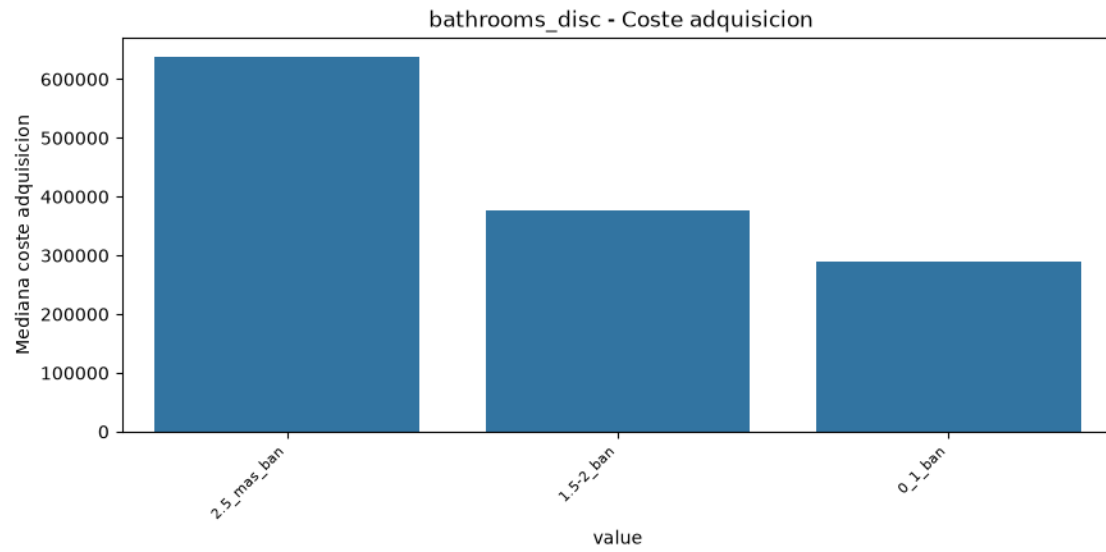


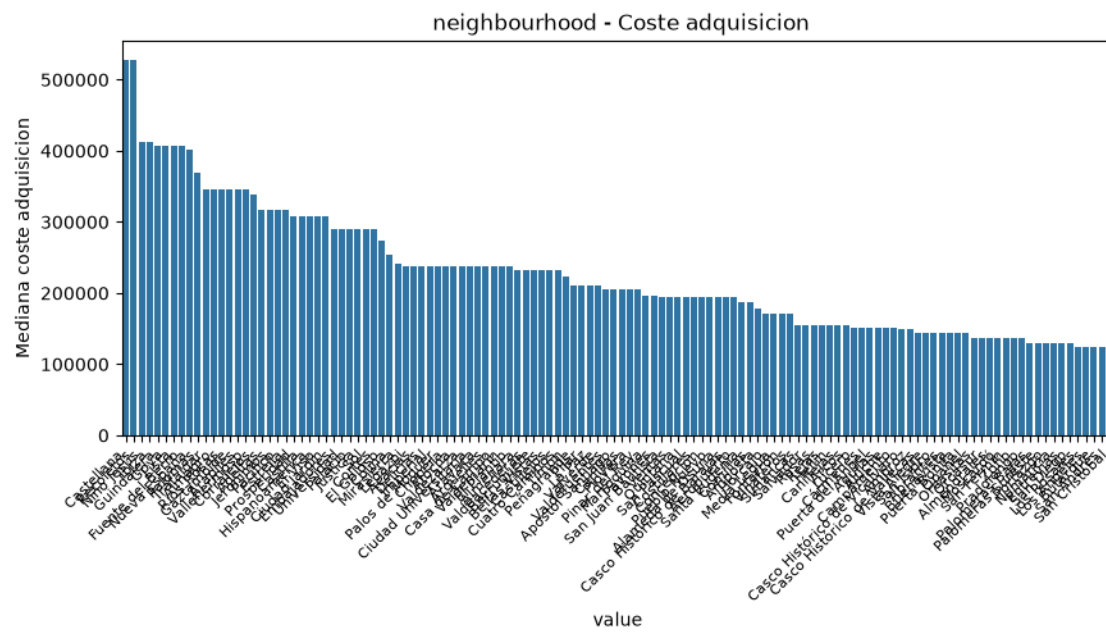
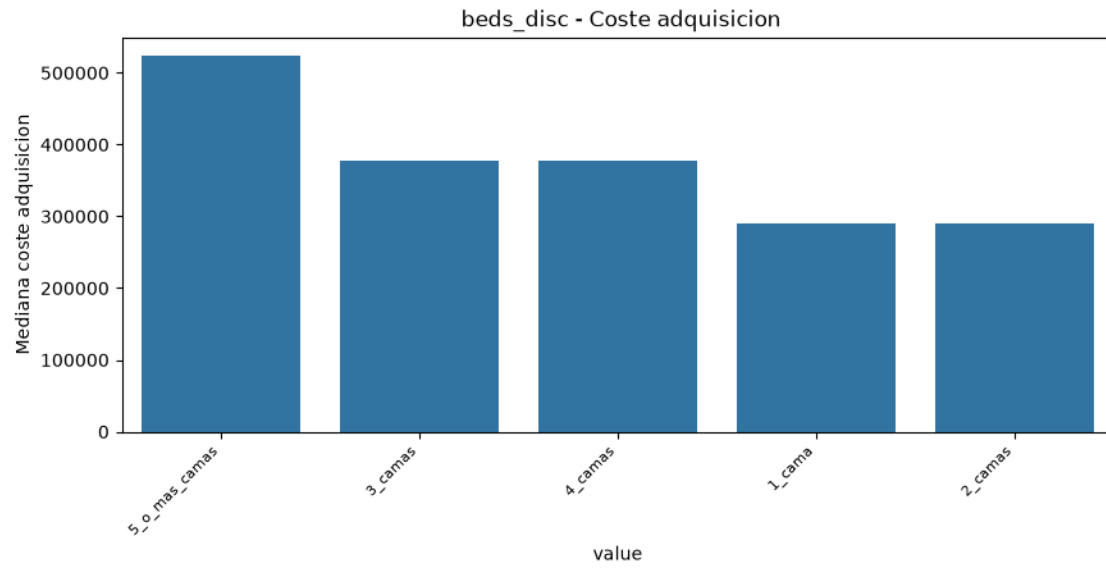


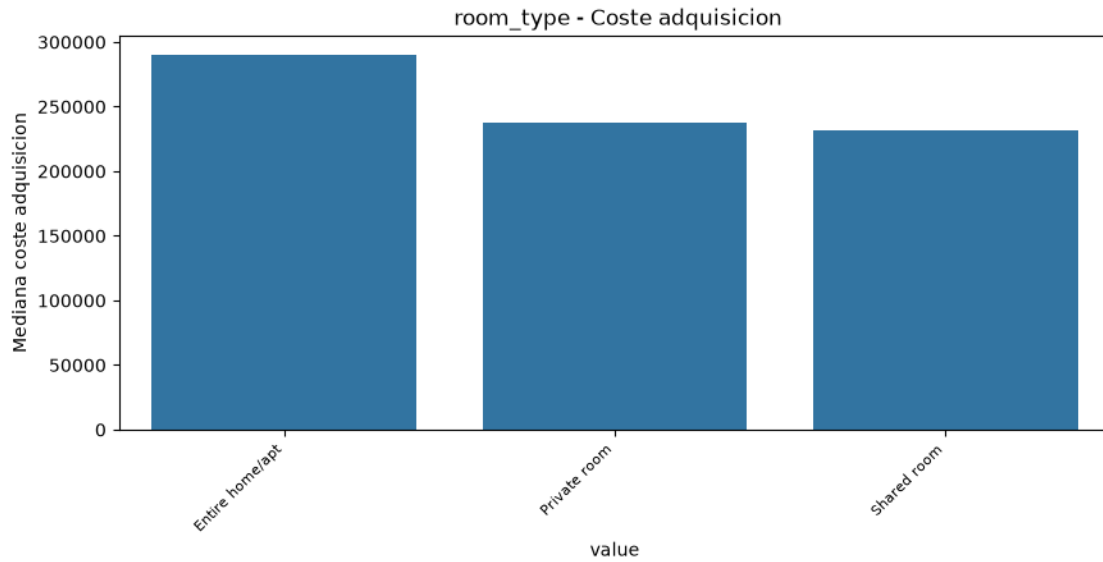
```
[19]: for dimension in minicubo["variable"].unique():
      data = minicubo[minicubo["variable"] == dimension].
      ↪sort_values("coste_adquisicion", ascending = False)
      plt.figure(figsize=(10,4))
      sns.barplot(x=data["value"], y=data["coste_adquisicion"])
      plt.title(f" {dimension} - Coste adquisicion")
```

```
plt.ylabel("Mediana coste adquisicion")
plt.xticks(rotation=45, ha="right", fontsize=8)
plt.show()
```

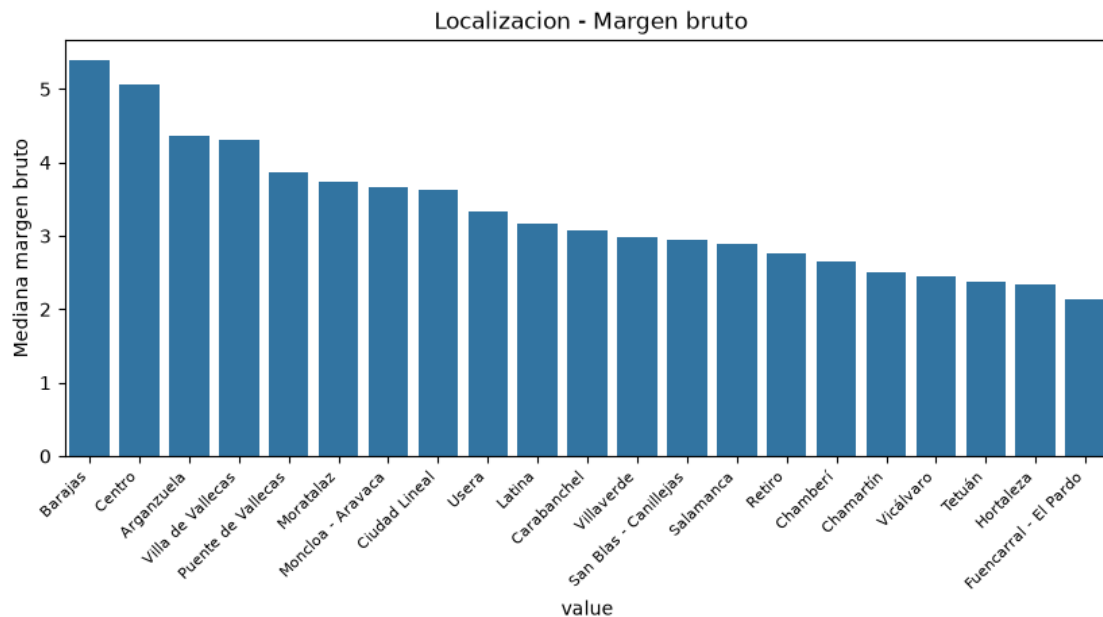


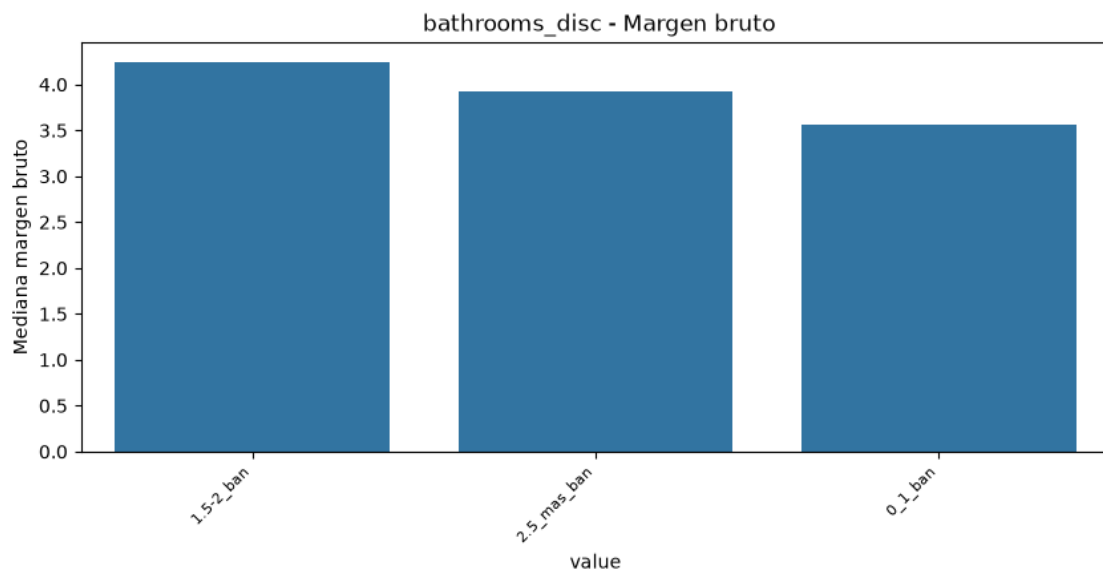
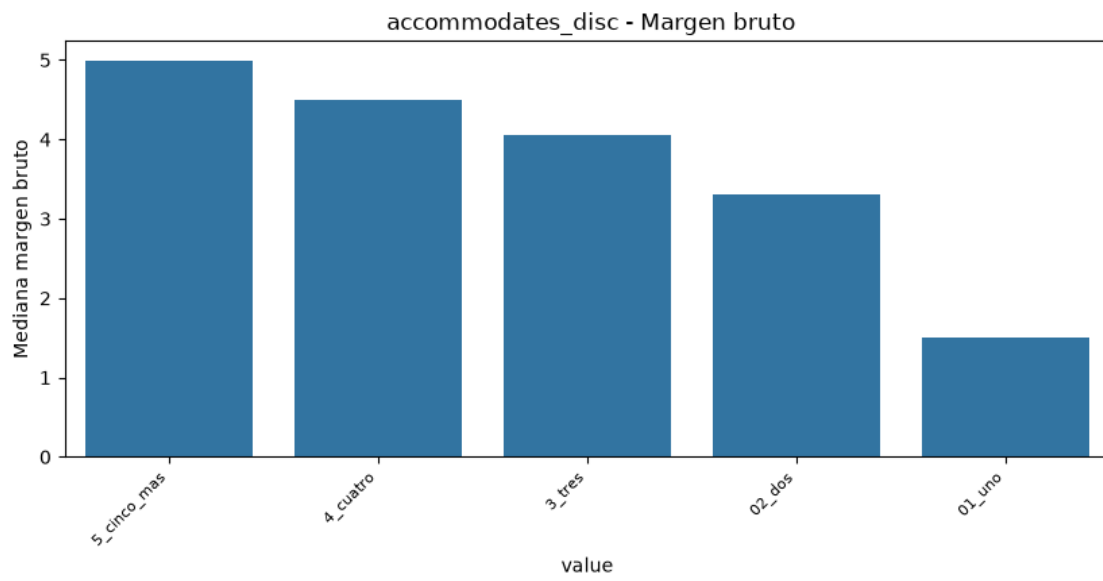


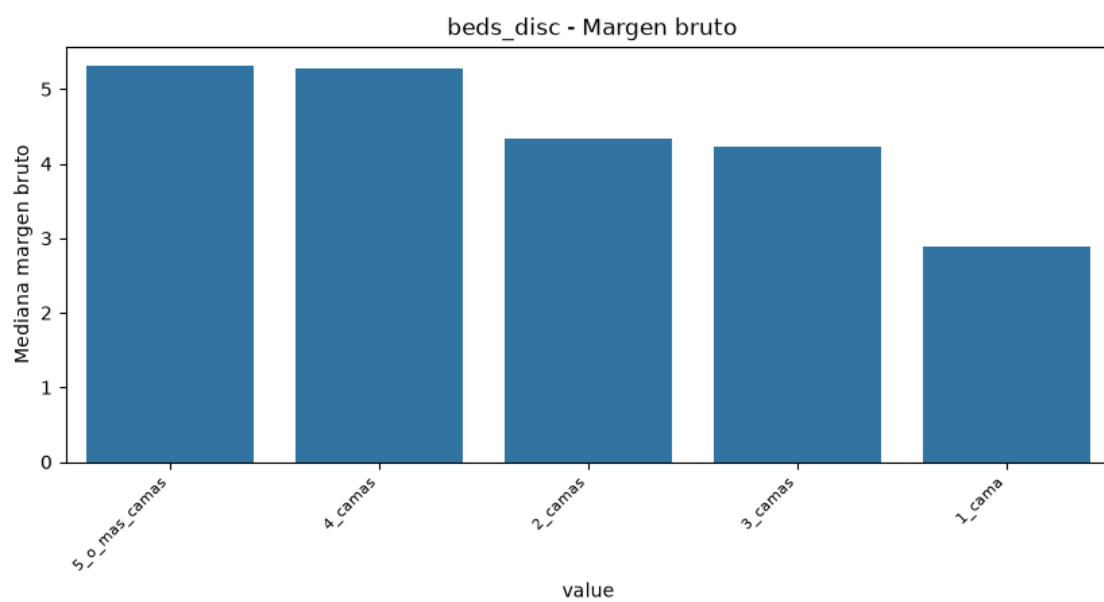
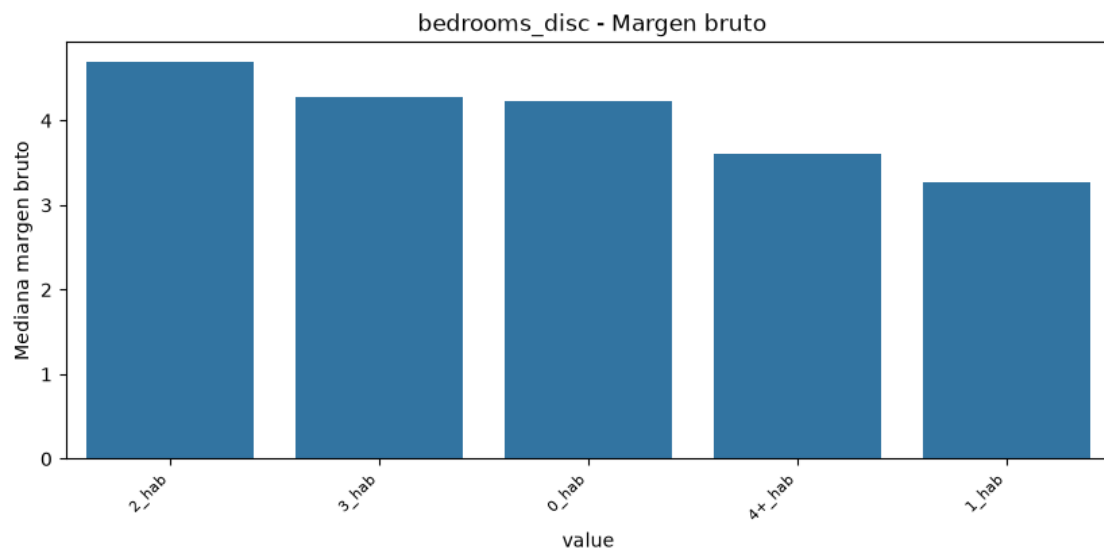


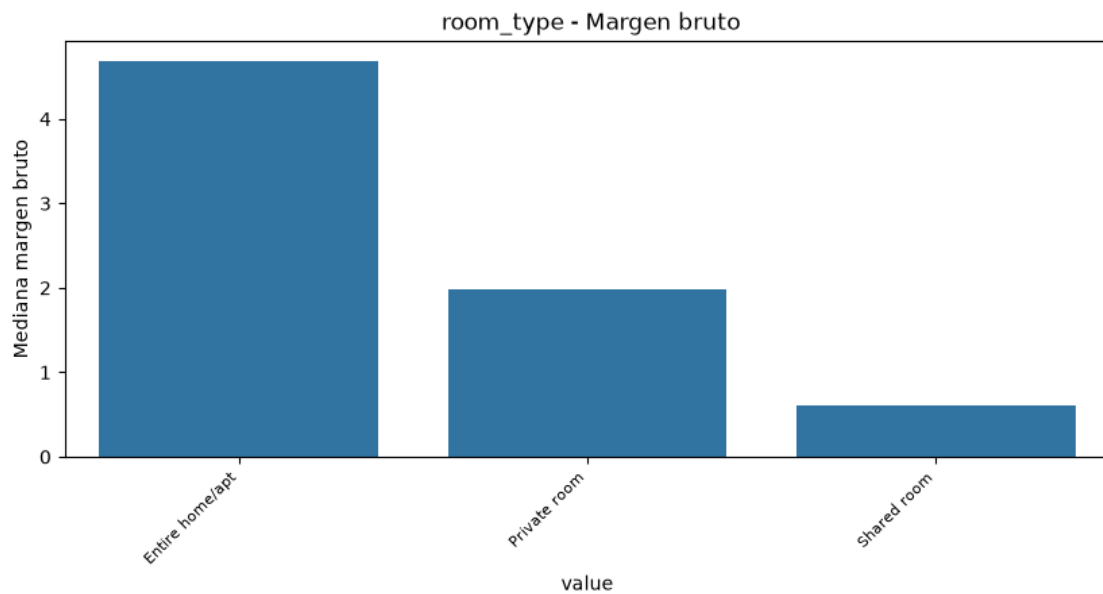
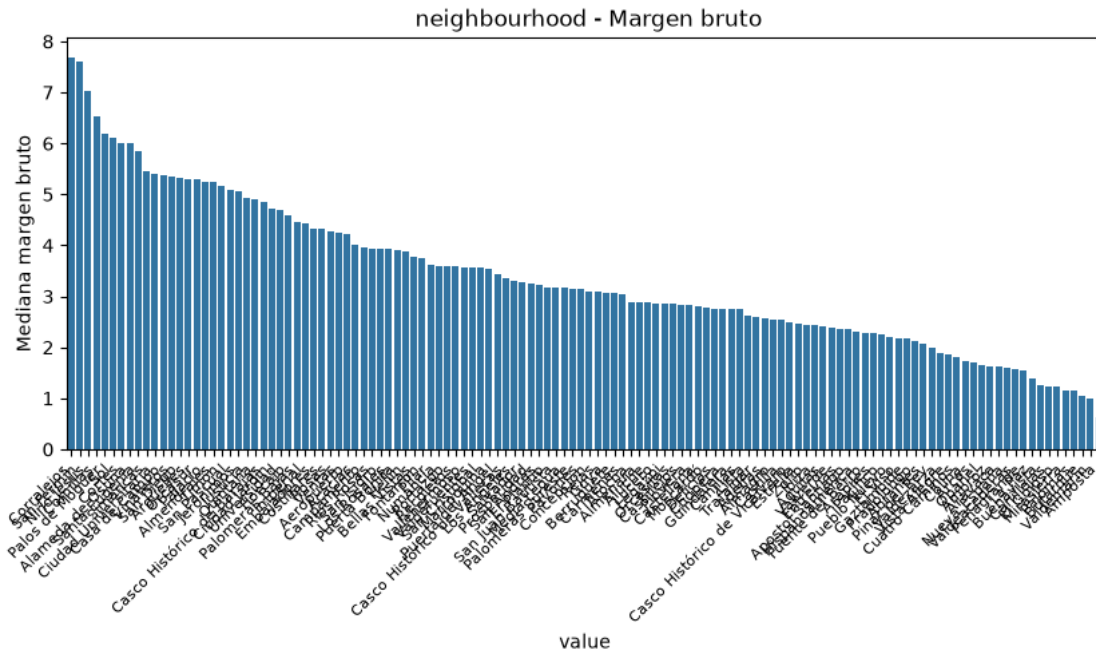


```
[20]: for dimension in minicubo["variable"].unique():
    data = minicubo[minicubo["variable"] == dimension].
    ↪sort_values("margen_bruto", ascending = False)
    plt.figure(figsize=(10,4))
    sns.barplot(x=data["value"], y=data["margen_bruto"])
    plt.title(f" {dimension} - Margen bruto")
    plt.ylabel("Mediana margen bruto")
    plt.xticks(rotation=45, ha="right", fontsize=8)
    plt.show()
```









1 Insights de rentabilidad

Los distritos con mayor rentabilidad son Barajas y Centro.

Los inmuebles que permiten 5 o mas personas tienen mas rentabilidad posteriormente de 4, pero se aprecia un salto considerable en el margen cuando es de 2 personas.

Los inmuebles con 1.5 a 2 baños son los que muestran mas rentabilidad.

Los inmuebles con 2 habitaciones muestran mas rentabilidad, mientras que 3 y 0 habitaciones siguen despues casi empatados.

Los inmuebles que tiene 4 y 5 camas parecen tener mayor rentabilidad, se aprecia que tienen la misma, lo mismo sucede con 2 y 3 camas.

El barrio mas rentable parece ser Corralejos.

El tipo de inmuebles con mayor rentabilidad es una casa/apartamento, aparentemente superior a cuartos privados/compartidos.

```
[21]: #Utilizemos los insights anteriores
```

```
df_prop = df.query("Localizacion in ['Barajas', 'Centro', 'Arganzuela', 'Villa_\n↳de Vallecas'] and accommodates_disc in ['5_cinco_mas','4_cuatro'] and_\n↳bedrooms_disc in ['2_hab','3_hab'] and room_type == 'Entire home/apt')\ndf_prop
```

```
[21]:
```

	name	neighbourhood	\
15	2 bedroom apartment in Chueca-Fuen	Justicia	
19	Ático Celestial en Barrio La Latina	Palacio	
22	Malasaña: Amazing&extremely central 2BD flat	Universidad	
25	Sol. Your home in Madrid's Heart	Sol	
39	Heima Plaza Mayor - Latina 32	Palacio	
...	...	...	
8261	Fuencarral XXIII	Justicia	
8264	Dúplex frente Palacio Real con hidromasaje	Palacio	
8265	Apartamento Premium - 2BR - Centro histórico -...	Sol	
8266	Piso elegante en Atocha 2 hab. 2 baños	Embajadores	
8267	Piso Amplio cerca Plaza de España 2 hab 2 baños	Palacio	

	latitude	longitude	room_type	precio_noche	minimum_nights	\
15	40.422420	-3.695620	Entire home/apt	84.0	5	
19	40.408690	-3.710730	Entire home/apt	178.0	3	
22	40.423230	-3.704340	Entire home/apt	114.0	15	
25	40.416040	-3.708340	Entire home/apt	127.0	2	
39	40.410270	-3.712200	Entire home/apt	365.0	3	
...	...	...	...	...	...	
8261	40.427935	-3.701677	Entire home/apt	455.0	1	
8264	40.417539	-3.711291	Entire home/apt	162.0	1	
8265	40.414660	-3.704010	Entire home/apt	162.0	2	
8266	40.410140	-3.697220	Entire home/apt	246.0	2	
8267	40.419857	-3.708920	Entire home/apt	253.0	2	

	availability_365	description	\
15	199	Comfy and cozy Two bedrooms apartment in Down...	
19	259	Spectacular penthouse with two bedrooms and on...	

22	271	Fantastic apartment located in the vibrant Mal...
25	273	DESIGN apartment in the HEART of Madrid. ...
39	306	A unique apartment, located in an old, early 1...
...	...	...
8261	345	This spacious, renovated apartment with modern...
8264	329	Experience Madrid from an exclusive duplex i...
8265	337	Enjoy the comfort of this magnificent accommod...
8266	288	It has a privileged location, which will make ...
8267	308	It has a privileged location, which will make ...

	accommodates	...	precio_noche_total	ingreso_anual	m2_estimado	\
15	5	...	84.0	9240.0	65.0	
19	6	...	178.0	44856.0	65.0	
22	4	...	114.0	29070.0	65.0	
25	5	...	127.0	32385.0	65.0	
39	6	...	365.0	72270.0	110.0	
...	...	...	...	...	...	
8261	6	...	455.0	2730.0	110.0	
8264	4	...	162.0	1944.0	65.0	
8265	5	...	162.0	1944.0	65.0	
8266	4	...	246.0	1476.0	65.0	
8267	4	...	253.0	1518.0	65.0	

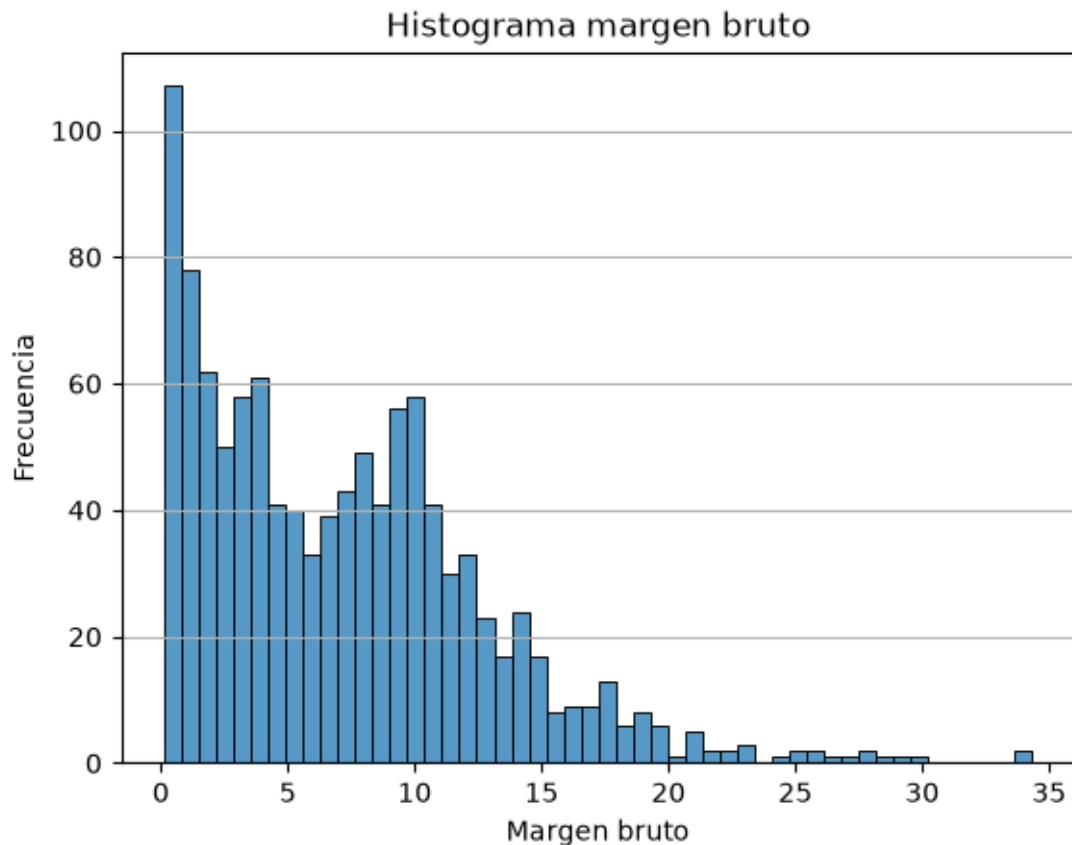
	coste_adquisicion	atractivo_turistico	bedrooms_disc	bathrooms_disc	\
15	377568.75	10.892490	2_hab	0_1_ban	
19	377568.75	5.192185	2_hab	0_1_ban	
22	377568.75	10.960249	2_hab	1.5-2_ban	
25	377568.75	18.119982	2_hab	0_1_ban	
39	638962.50	6.705295	3_hab	2.5_mas_ban	
...	...	...	...	...	
8261	638962.50	3.145864	3_hab	1.5-2_ban	
8264	377568.75	17.108644	2_hab	1.5-2_ban	
8265	377568.75	15.174231	2_hab	1.5-2_ban	
8266	377568.75	9.662741	2_hab	1.5-2_ban	
8267	377568.75	17.017958	2_hab	1.5-2_ban	

	accommodates_disc	beds_disc	margen_bruto
15	5_cinco_mas	4_cammas	2.45
19	5_cinco_mas	3_cammas	11.88
22	4_cuatro	2_cammas	7.70
25	5_cinco_mas	4_cammas	8.58
39	5_cinco_mas	5_o_mas_cammas	11.31
...	...	...	...
8261	5_cinco_mas	4_cammas	0.43
8264	4_cuatro	1_cama	0.51
8265	5_cinco_mas	1_cama	0.51
8266	4_cuatro	2_cammas	0.39

8267 4_cuatro 2_camás 0.40

[1087 rows x 27 columns]

```
[22]: sns.histplot(df_prop.margen_bruto, bins=50)
plt.title("Histograma margen bruto")
plt.ylabel("Frecuencia")
plt.xlabel("Margen bruto")
plt.grid(axis='y');
```

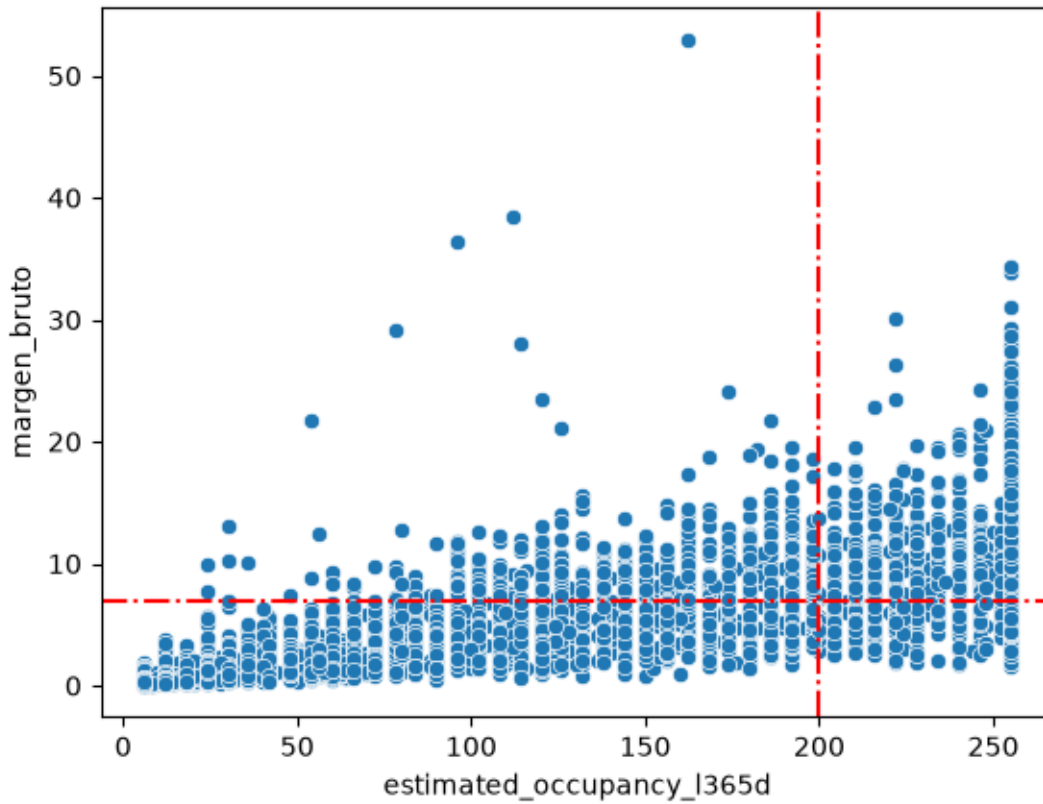


```
[23]: # Podemos hacer una grafico para estudiar la ocupacion estimada buscando un
      ↪ margen bruto específico

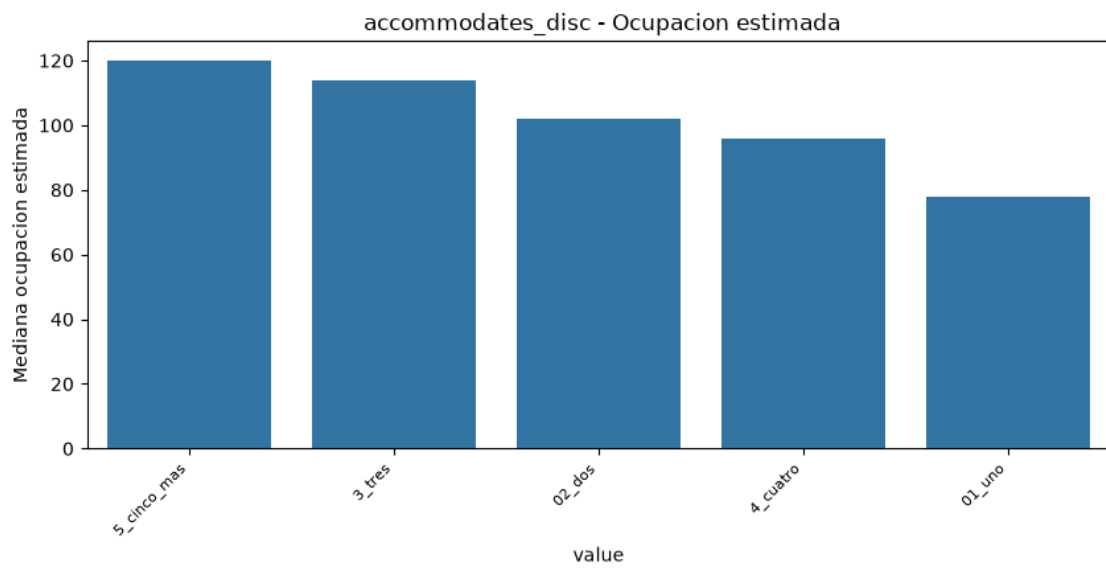
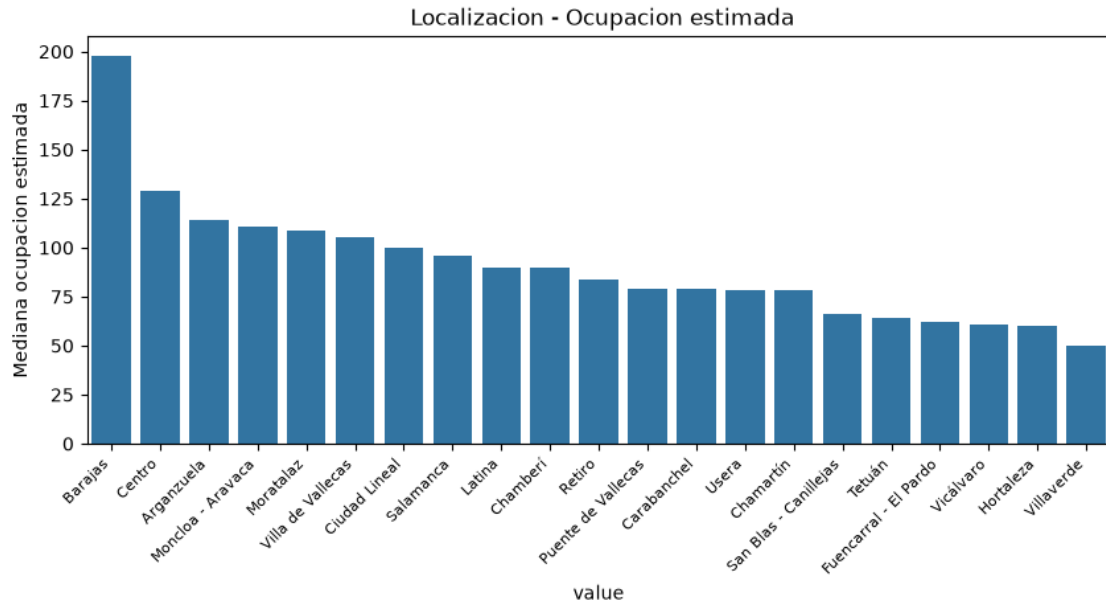
margen_bruto = 7
estimated_d = 200

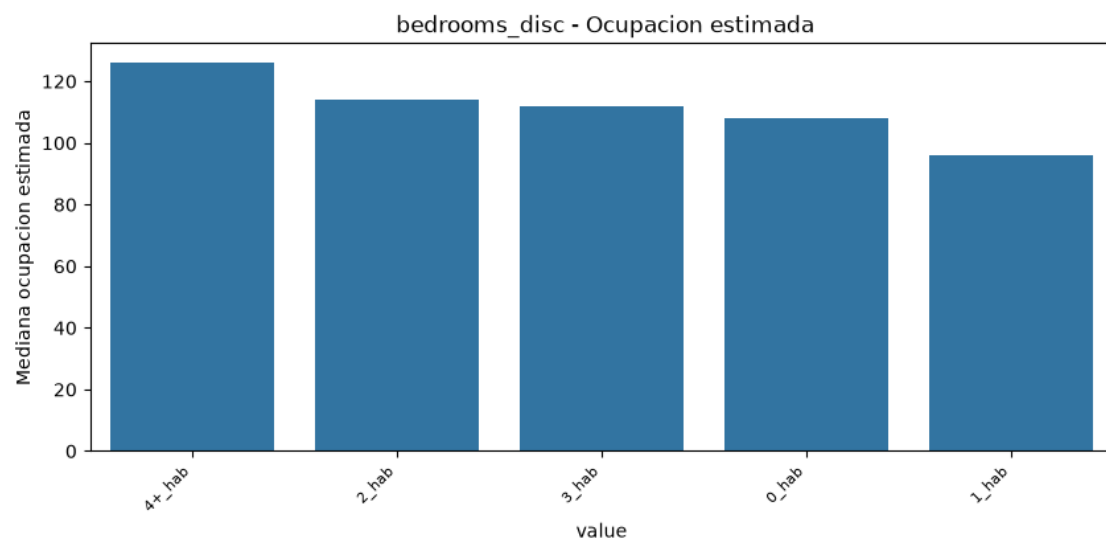
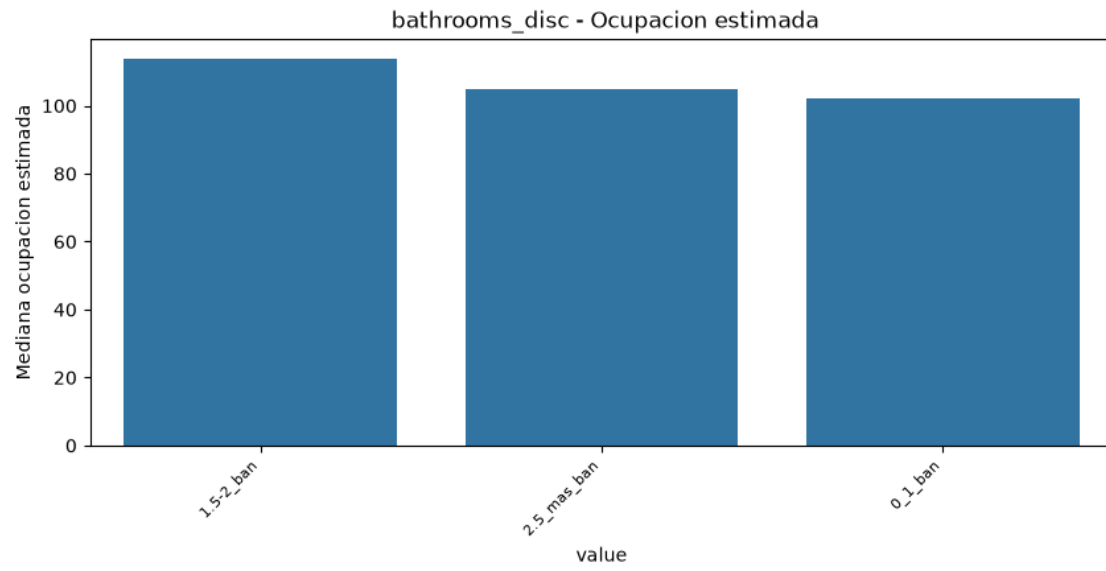
sns.scatterplot(data = df, x="estimated_occupancy_l365d", y="margen_bruto")
plt.axhline(margen_bruto, color="red", linestyle="-.")
plt.axvline(estimated_d, color="red", ls="-.")
```

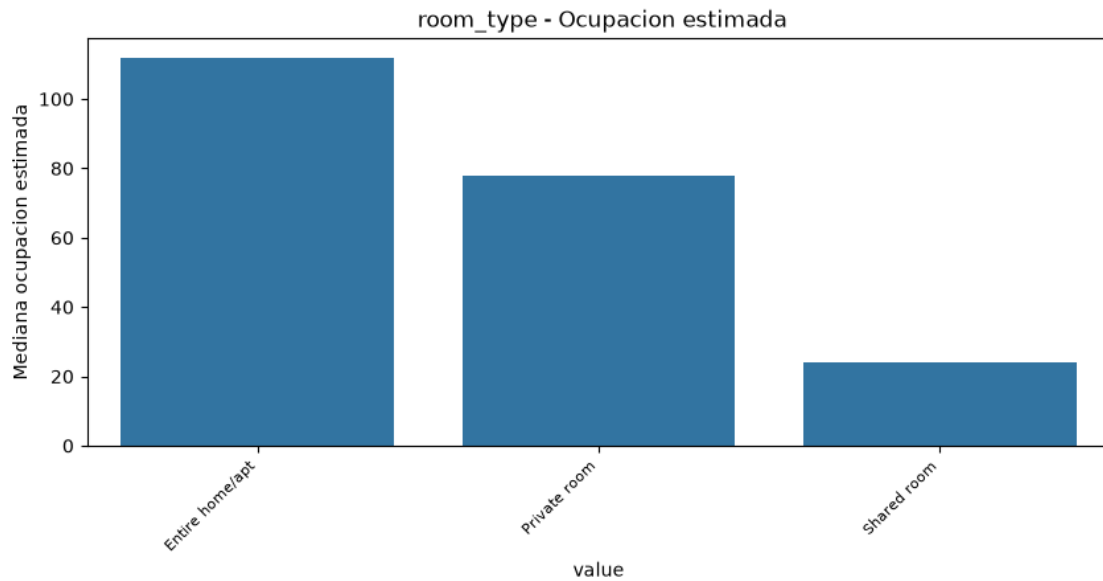
[23]: <matplotlib.lines.Line2D at 0x7ffeb95aad50>



```
[24]: for dimension in minicubo["variable"].unique():
        data = minicubo[minicubo["variable"] == dimension].
        ↪sort_values("estimated_occupancy_l365d", ascending = False)
        plt.figure(figsize=(10,4))
        sns.barplot(x=data["value"], y=data["estimated_occupancy_l365d"])
        plt.title(f" {dimension} - Ocupacion estimada")
        plt.ylabel("Mediana ocupacion estimada")
        plt.xticks(rotation=45, ha="right", fontsize=8)
        plt.show()
```







2 Insights ocupación

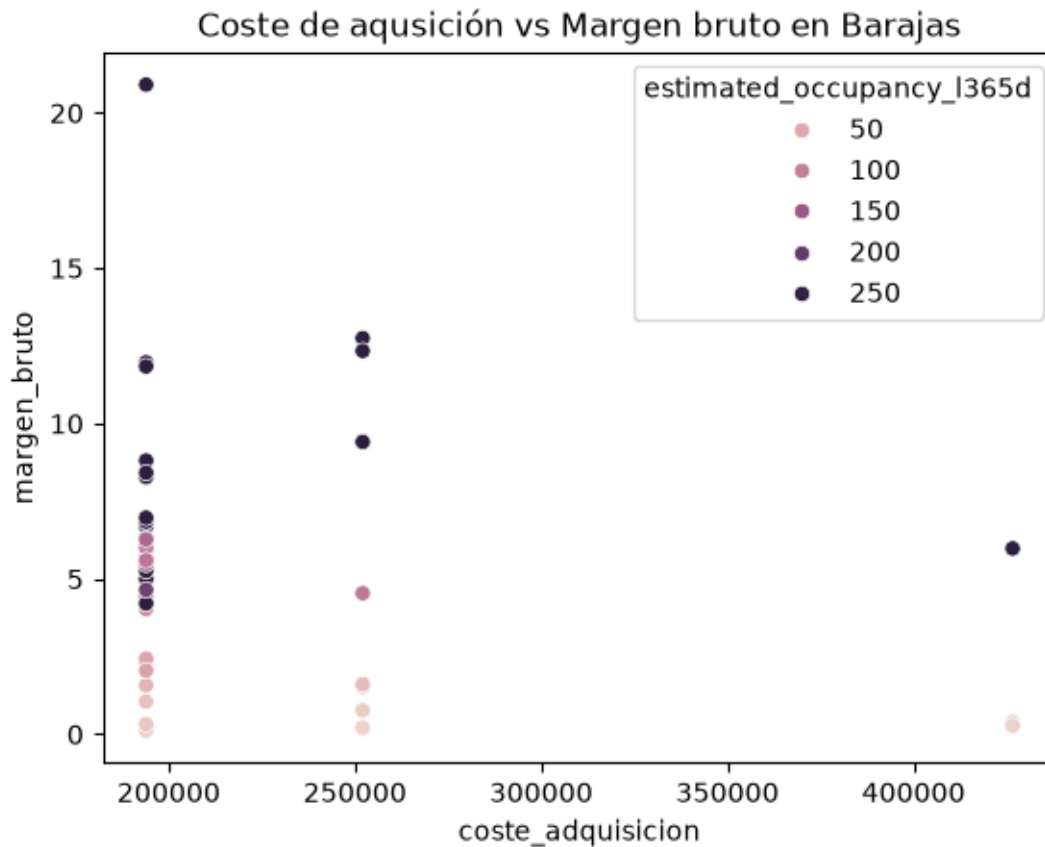
Parece ser que Barajas es el distrito con mas ocupación, seguido de Centro y Arganzuela.

Los inmuebles con 5 o mas espacio para personas son los que espera mas ocupacion, seguido de los de 3 personas.

4 o mas habitaciones son las que mas ocupación esperan.

```
[25]: sns.scatterplot(data = df.query("Localizacion == 'Barajas'"),
    ↪x="coste_adquisicion", y="margen_bruto", hue="estimated_occupancy_l365d")
plt.title("Coste de adquisición vs Margen bruto en Barajas")
```

```
[25]: Text(0.5, 1.0, 'Coste de adquisición vs Margen bruto en Barajas')
```



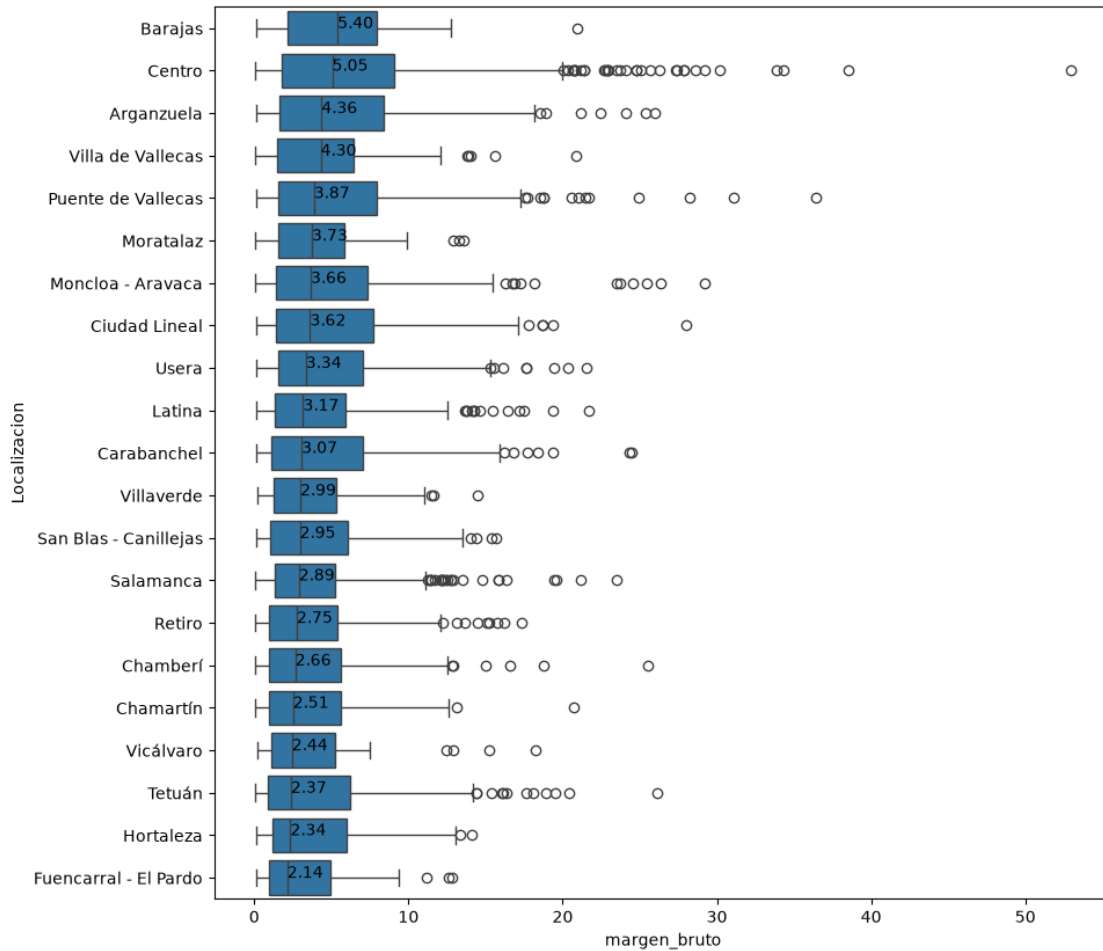
3 Donde comprar

Los 3 distritos a explorar son Barajas, Centro y Arganzuela.

```
[26]: plt.figure(figsize=(10,10));
ax = sns.boxplot(data=df, y = "Localizacion", x= "margen_bruto", order=df.
    ↳groupby("Localizacion")["margen_bruto"].median().sort_values(ascending=False).
    ↳index);

medianas = df.groupby("Localizacion")["margen_bruto"].median()

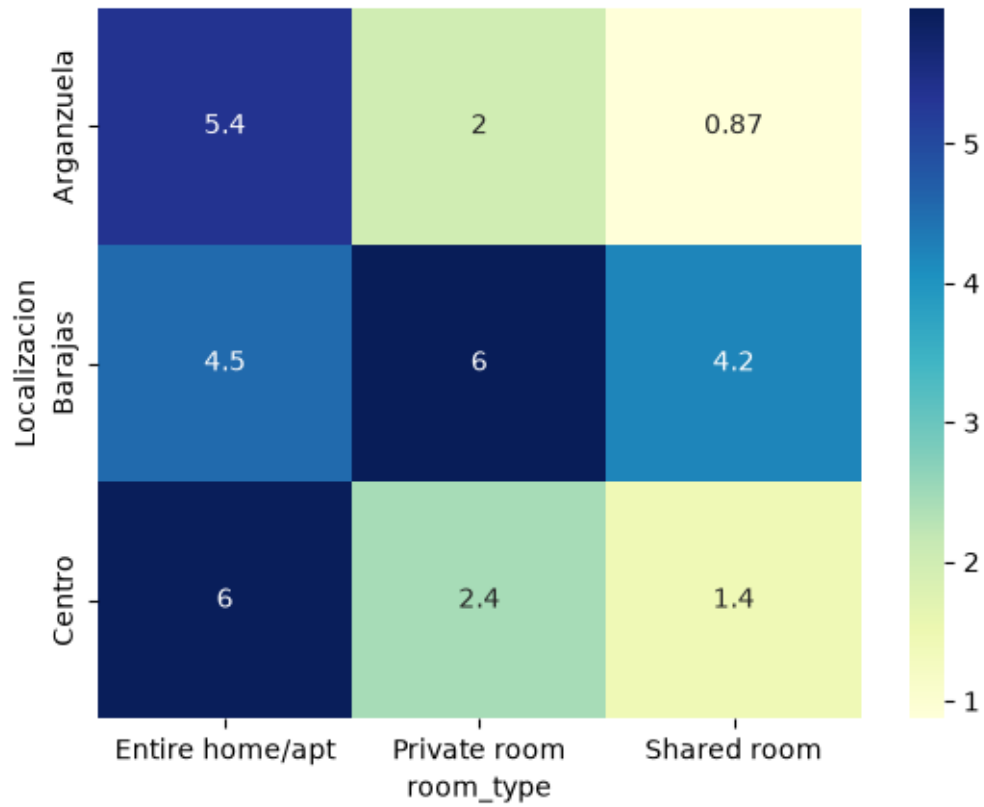
for i, label in enumerate(ax.get_yticklabels()):
    valor_med = medianas[label.get_text()]
    ax.text(valor_med, i, f"{valor_med:.2f}")
```



```
[27]: dist_int = ["Barajas", "Centro", "Arganzuela"]
      df_dist = df[df.Localizacion.isin(dist_int)]
```

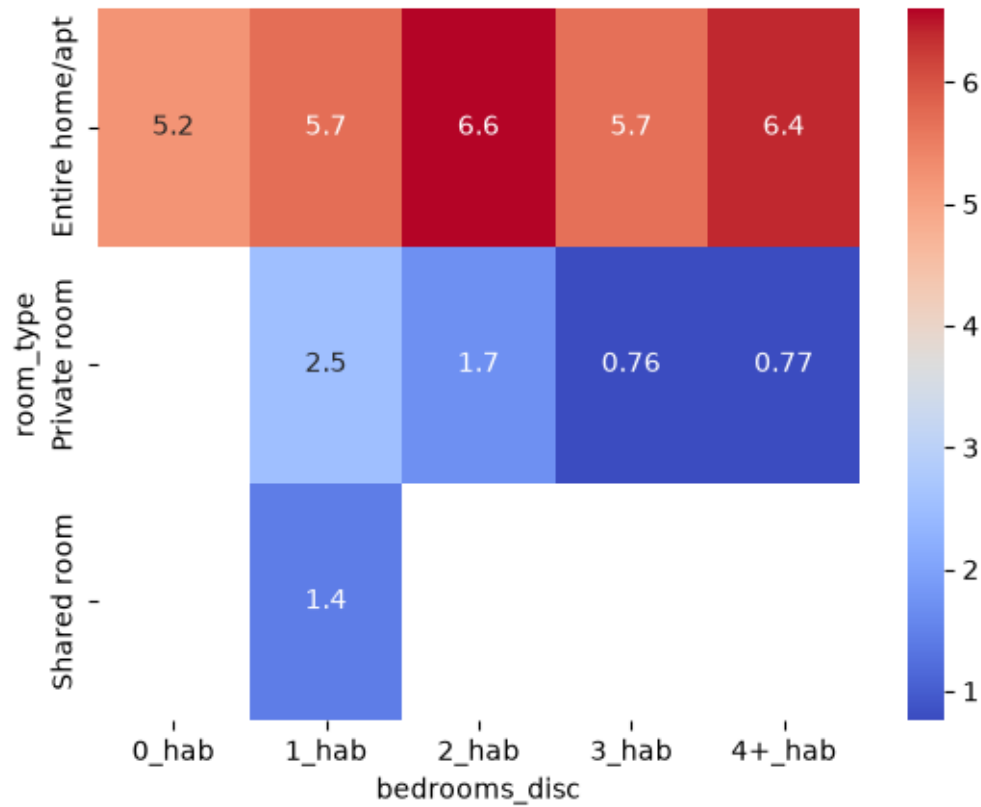
```
[28]: pt = df_dist.pivot_table(values="margen_bruto", index="Localizacion",
      ↪ columns="room_type", aggfunc="median")
      sns.heatmap(pt, annot=True, cmap="YlGnBu")
```

```
[28]: <Axes: xlabel='room_type', ylabel='Localizacion'>
```



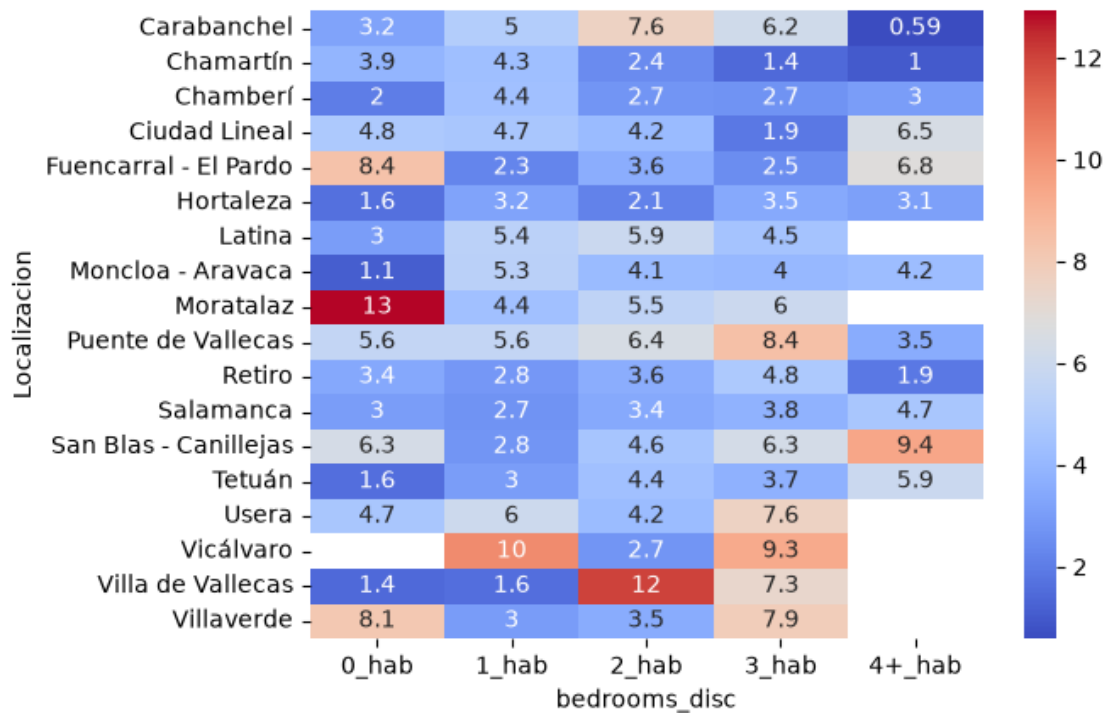
```
[29]: pt = df_dist.pivot_table(values="margen_bruto", index="room_type",
    ↪columns="bedrooms_disc", aggfunc="median")
sns.heatmap(pt, annot=True, cmap="coolwarm")
```

```
[29]: <Axes: xlabel='bedrooms_disc', ylabel='room_type'>
```



```
[33]: df_temp = df.query("room_type == 'Entire home/apt' and Localizacion not in_
↳ ['Barajas', 'Centro', 'Arganzuela']")
pt = df_temp.pivot_table(values="margen_bruto", index="Localizacion",
↳ columns="bedrooms_disc", aggfunc="median")
sns.heatmap(pt, annot=True, cmap="coolwarm")
```

```
[33]: <Axes: xlabel='bedrooms_disc', ylabel='Localizacion'>
```



4 Análisis geográfico

```
[35]: import folium
from folium.plugins import HeatMap

df_map = df.dropna(subset=["latitude", "longitude", "margen_bruto"])
madrid_coords = [40.4168, -3.7038]

m = folium.Map(location=madrid_coords, zoom_start=11)

heat_data = [[row["latitude"], row["longitude"], row["margen_bruto"]] for idx, row in df_map.iterrows()]
HeatMap(heat_data, radius=8, blur=15, max_zoom=1).add_to(m)

from IPython.display import display
display(m)

m.save("../entregables/graficos/mapa_rentabilidad.html")
```

<folium.folium.Map at 0x7ffeb83af890>

```
[36]: df_map = df.dropna(subset=["latitude", "longitude", "margen_bruto"])
madrid_coords = [40.4168, -3.7038]
```

```

m2 = folium.Map(location=madrid_coords, zoom_start=11)

heat_data = [[row["latitude"], row["longitude"],
    ↳row["estimated_occupancy_l365d"]] for idx, row in df_map.iterrows()]
HeatMap(heat_data, radius=8, blur=15, max_zoom=1).add_to(m2)

from IPython.display import display
display(m2)

m2.save("../entregables/graficos/mapa_ocupacion.html")

```

<folium.folium.Map at 0x7ffeb9da02d0>

```

[38]: df_oportunidades = df_map.query("margen_bruto > 20 and estimated_occupancy_l365d_
    ↳> 200")
m3 = folium.Map(location=madrid_coords, zoom_start=11)

for _, row in df_oportunidades.iterrows():
    folium.CircleMarker(
        location=[row["latitude"], row["longitude"]],
        radius=6,
        color="blue",
        fill=True,
        popup=f"Barrio: {row["neighbourhood"]} <br> Rentabilidad:
    ↳ {row["margen_bruto"]:.2f} <br> Ocupación: {row["estimated_occupancy_l365d"]}"
    ).add_to(m3)

from IPython.display import display
display(m3)

m3.save("../entregables/graficos/mapa_ocupacion.html")

```

<folium.folium.Map at 0x7ffeb9992060>

[]: