

SISTEMAS DE INFORMACIÓN EN LAS ORGANIZACIONES

Práctica 1

El poder de los datos: El caso Airbnb

Curso 2024-25



Estudiantes: Iván Soler Parra
Guillem Rodríguez Mitjana

Entrega: 23 / 10 / 2024

Índice

1. Introducción	3
2. Análisis Exploratorio de Datos	4
2.1 ¿De qué manera están distribuidos los alojamientos?	5
2.2 ¿Cómo están distribuidos los precios de los alojamientos?	7
2.3 ¿En qué distritos están los alojamientos más caros y baratos?	8
2.4 ¿Qué ciudad tiene más disponibilidad de alojamientos?	10
2.5 ¿Dónde están los mejores alojamientos?	11
2.6 ¿Qué propietarios tienen más alojamientos?	13
2.7 ¿Qué factores afectan al precio?	15
2.7.1 ¿Los alojamientos más caros tienen más reseñas?	17
2.7.2 ¿El máximo de huéspedes afecta al precio de los alojamientos?	18
2.7.3 ¿Los alojamientos mejor valorados tienen más reseñas?	20
2.8 ¿Los alojamientos más caros ofrecen más servicios?.....	22
3. Visualizaciones geográficas	25
3.1 Mapa de calor de la distribución de precios	29
3.2 Mapa de categorías de los tipos de alojamientos	32
3.3 Mapa de coropletas de los barrios	35
3.4 Mapa de burbujas de la cantidad de alojamientos	40
3.5 Mapa de categorías de los hosts	43

1. Introducción

En los últimos años, el mercado de alquileres temporales ha crecido exponencialmente, impulsado en gran parte por plataformas como Airbnb. Este crecimiento ha transformado tanto la forma en que las personas encuentran alojamiento como el mercado inmobiliario en grandes ciudades. Madrid y Barcelona, dos de las ciudades más emblemáticas de España, se han convertido en puntos clave para el turismo y, por lo tanto, para los alojamientos a corto plazo.

A medida que estas ciudades continúan atrayendo a millones de turistas cada año, el análisis de los datos relacionados con los alojamientos temporales ofrece una oportunidad única para comprender las dinámicas del mercado y descubrir patrones que pueden influir en la oferta y la demanda. En este contexto, el análisis comparativo entre Madrid y Barcelona no solo permite entender las diferencias en los precios, tipos de alojamiento y características, sino también anticipar tendencias y ayudar a los gestores de propiedades a tomar decisiones más informadas.

El **objetivo de este informe** es realizar un análisis exploratorio de los datos de Airbnb de ambas ciudades, comparando aspectos clave como los precios, tipos de propiedades y valoraciones de los alojamientos. Utilizaremos técnicas de análisis estadístico y visualización para identificar diferencias significativas y generar conclusiones que ayuden a interpretar las particularidades de cada mercado.

Mediante este análisis, buscamos responder a preguntas como: **¿Existen diferencias significativas en los precios entre Madrid y Barcelona? ¿Qué tipo de propiedades predominan en cada ciudad? ¿Qué factores influyen en las mejores valoraciones?** Las respuestas a estas preguntas permitirán obtener una visión clara y precisa del comportamiento del mercado de alquileres temporales en ambas ciudades.

Además, **como complemento** a este análisis, en una segunda parte **se ha elaborado una web interactiva** que presenta mapas geográficos con los datos estudiados. Estos mapas permiten visualizar de manera clara las diferencias geoespaciales entre Madrid y Barcelona, facilitando la identificación de patrones de distribución de precios y otros factores relevantes en las distintas zonas de cada ciudad.

La estructura general del proyecto es la siguiente:

```
SI0_Prac1/
├── eda/
├── web/
└── Informe.pdf
```

Donde se encuentran las diferentes partes:

- **eda (Exploratory Data Analysis)**: contiene los estudios de la primera parte.
- **web**: contiene la página web con los mapas geográficos interactivos de la segunda parte.
- **Informe.pdf**: este documento.

2. Análisis Exploratorio de Datos

La primera parte de la práctica consistía en realizar un análisis exploratorio de datos sobre los alojamientos que Airbnb tiene en diferentes ciudades. Un análisis exploratorio es un tratamiento estadístico utilizado para explorar, describir, resumir y entender los datos con los que se trabajan.

En los datos proporcionados por la compañía, pudimos ver que cada alojamiento tenía unas 70 variables. Estas variables pueden ser de diferentes tipos:

Variable			
Cualitativa (categórica)		Cuantitativa (numérica)	
Nominal (sin secuencia lógica)	Ordinal (ordenado)	Discreta (contar)	Continua (medir)
Estado civil, religión, lugar de nacimiento...	Talla de ropa, nivel de satisfacción...	Número de coches, número de hijos...	Peso, velocidad, temperatura...

La estructura de esta parte es la siguiente:

```
eda/
└── 2.1/
    ├── script.py
    ├── grafico1.png
    └── grafico2.png
├── 2.2/
├── 2.3/
├── 2.4/
├── 2.5/
├── 2.6/
└── 2.7/
    ├── 2.7.1/
    ├── 2.7.2/
    ├── 2.7.3/
    └── 2.8/
        ├── listings_barcelona.csv
        └── listings_madrid.csv
```

Básicamente, cada estudio tiene su propio directorio con el nombre de su índice en este documento. Para hacer los estudios, decidimos usar **Python**, ya que cuenta con librerías muy potentes, tanto para trabajar con *dataframes* como para representar y generar gráficos. Por eso, cada estudio tiene su **script.py** que hace todo el trabajo y guarda tanto los gráficos como las tablas en formato *.png* en el mismo directorio.

Las librerías que hemos usado son las siguientes:

- **Pandas**: proporciona estructuras de datos como *dataframes*, que nos facilitan la manipulación, análisis y limpieza de datos tabulares.
- **Matplotlib**: biblioteca de gráficos para crear visualizaciones personalizadas como gráficos de líneas, barras, histogramas, etc.
- **Seaborn**: biblioteca de visualización de datos basada en *matplotlib*, que simplifica la creación de gráficos estadísticos con estilos y colores predefinidos.

2.1 ¿De qué manera están distribuidos los alojamientos?

Lo primero que nos pareció interesante analizar fue la distribución de los alojamientos en las diferentes ciudades. Es decir, observar la cantidad de alojamientos que hay por distrito.

Para ello, realizamos una tabla de distribución de frecuencia sobre la variable cualitativa referente a los distritos: neighbourhood_group_cleansed. Además de la tabla, también hicimos una gráfica de pastel para una mejor visualización de los resultados.

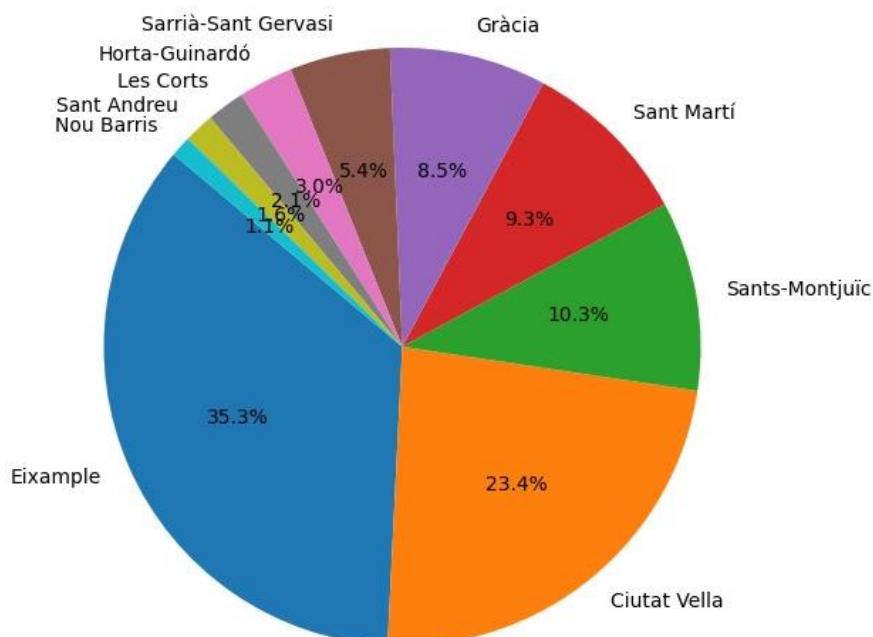
Una vez que tuvimos claras las técnicas que debíamos aplicar, obtuvimos la tabla de la ciudad de Barcelona:

Distrito	Frecuencia	Frecuencia relativa	Porcentaje
Eixample	6882	0.353	35.32
Ciutat Vella	4565	0.234	23.43
Sants-Montjuïc	1997	0.103	10.25
Sant Martí	1808	0.093	9.28
Gràcia	1653	0.085	8.48
Sarrià-Sant Gervasi	1061	0.054	5.45
Horta-Guinardó	575	0.03	2.95
Les Corts	402	0.021	2.06
Sant Andreu	318	0.016	1.63
Nou Barris	221	0.011	1.13
TOTAL	19482	1.0	100.0

En ella, pudimos ver que hay un total de 19482 alojamientos, de los cuales, el 35.32% se encuentran en el distrito del Eixample. Junto a este, también podemos destacar al distrito de Ciutat Vella, ya que contiene un 23.43% de los alojamientos.

Comparado con el resto de los distritos, estos dos son los que más destacan, ya que entre los dos suman un 58.7%.

En la gráfica de pastel podemos ver claramente cómo estos dos distritos albergan más de la mitad de los alojamientos de Barcelona:

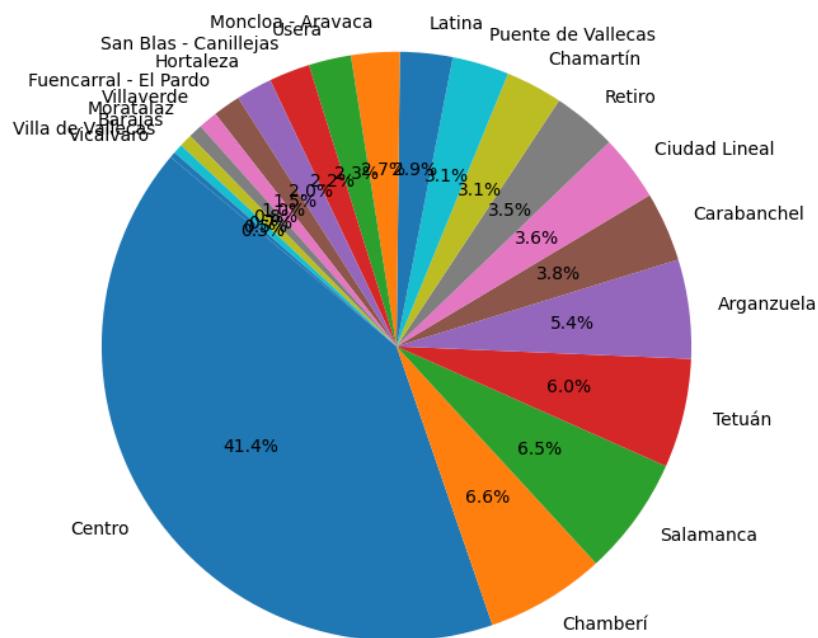


Después de analizar los alojamientos de Barcelona, decidimos analizar también los de Madrid, por lo que realizamos otra tabla de distribución de frecuencia:

Distrito	Frecuencia	Frecuencia relativa	Porcentaje
Centro	11165	0.414	41.38
Chamberí	1768	0.066	6.55
Salamanca	1757	0.065	6.51
Tetuán	1616	0.06	5.99
Arganzuela	1466	0.054	5.43
Carabanchel	1029	0.038	3.81
Ciudad Lineal	973	0.036	3.61
Retiro	952	0.035	3.53
Chamartín	841	0.031	3.12
Puente de Vallecas	840	0.031	3.11
Latina	777	0.029	2.88
Moncloa - Aravaca	718	0.027	2.66
Usera	626	0.023	2.32
San Blas - Canillejas	606	0.022	2.25
Hortaleza	534	0.02	1.98
Fuencarral - El Pardo	406	0.015	1.5
Villaverde	281	0.01	1.04
Moratalaz	207	0.008	0.77
Barajas	190	0.007	0.7
Villa de Vallecas	142	0.005	0.53
Vicálvaro	86	0.003	0.32
TOTAL	26980	1.0	100.0

En esta ocasión, pudimos ver que la capital tiene un total de 26980 alojamientos. Unos 7500 más que la ciudad de Barcelona.

Sin embargo, en la gráfica de pastel de Madrid, podemos apreciar que, comparado con la de Barcelona, la mayoría de los alojamientos se centran en un único distrito. Y es que solo el distrito Centro ya contiene un 41.38% de ellos:

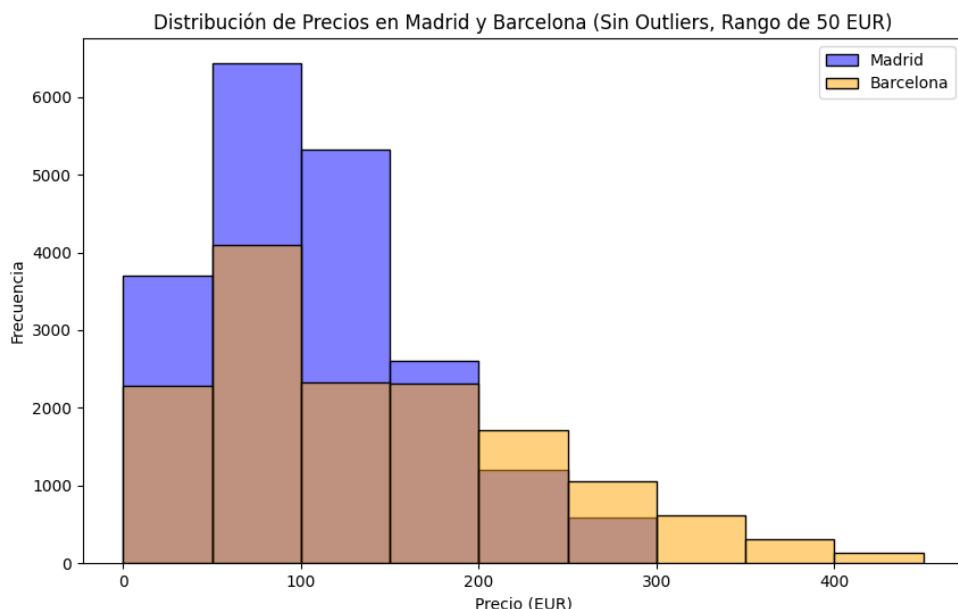


Los alojamientos se centran tanto en este, que en ninguno de los demás distritos se supera el 7%. Algo que no pasaba en Barcelona, donde teníamos hasta cinco distritos (Eixample, Ciutat Vella, Gràcia, Sant Martí y Sants-Montjuïc) con valores superiores al 8%.

2.2 ¿Cómo están distribuidos los precios de los alojamientos?

Otra de las preguntas que se haría cualquier persona interesada en alquilar un alojamiento es: **¿En qué ciudad hay más opciones para un determinado presupuesto?** El objetivo de este análisis es conocer la frecuencia de alojamientos en función del precio.

Para esto, trabajamos únicamente con la variable `price`, eliminando los valores atípicos para mostrar en la gráfica solo los valores más comunes. Entonces, generamos la siguiente gráfica con la distribución de los precios en rangos de 50€, donde podemos ver la comparación entre las opciones de alojamiento en Madrid y Barcelona.



Lo más importante en esta gráfica son los colores azul y naranja que sobresalen en cada barra. Si el color que sobresale es azul, indica una mayor abundancia de alojamientos en ese rango de precios en Madrid. Si el color es naranja, la abundancia es mayor en Barcelona.

Por lo tanto, si estás buscando una opción más económica que no supere los 200€, la mejor opción sería Madrid, ya que la cantidad de alojamientos en este rango es casi del doble. En cambio, si el precio no es una limitación, Barcelona ofrece más opciones a partir de los 200€, como se evidencia por la mayor dispersión en los precios.

Reforzando esto último, en **Madrid**, aproximadamente el **91.04%** de los alojamientos están por debajo de 200€, mientras que, en **Barcelona**, este porcentaje es de un **74.22%**, lo que refleja que es más probable encontrar opciones más costosas en esta última.

Además de observar la distribución de los precios en ambas ciudades, calculamos la mediana y el rango intercuartílico (IQR) para obtener una idea más precisa de los precios típicos. En **Madrid**, la mediana del precio es de 98, con un IQR de 82, lo que sugiere que la mayoría de los precios se encuentran entre **57€** y **139€**. Por su parte, en **Barcelona**, la mediana es de 120 y el IQR es de 135, lo que indica una mayor variabilidad de precios, situándose entre **52.5€** y **187.5€**. Esto refleja que, aunque la mediana de Barcelona es mayor, sus precios están más dispersos, mostrando una oferta más variada.

2.3 ¿En qué distritos están los alojamientos más caros y baratos?

El análisis de los precios medios por distrito en Madrid y Barcelona revela importantes diferencias entre ambos mercados. Este análisis es útil para identificar qué distritos presentan los alojamientos más económicos y en cuáles es necesario disponer de un mayor presupuesto para alquilar.

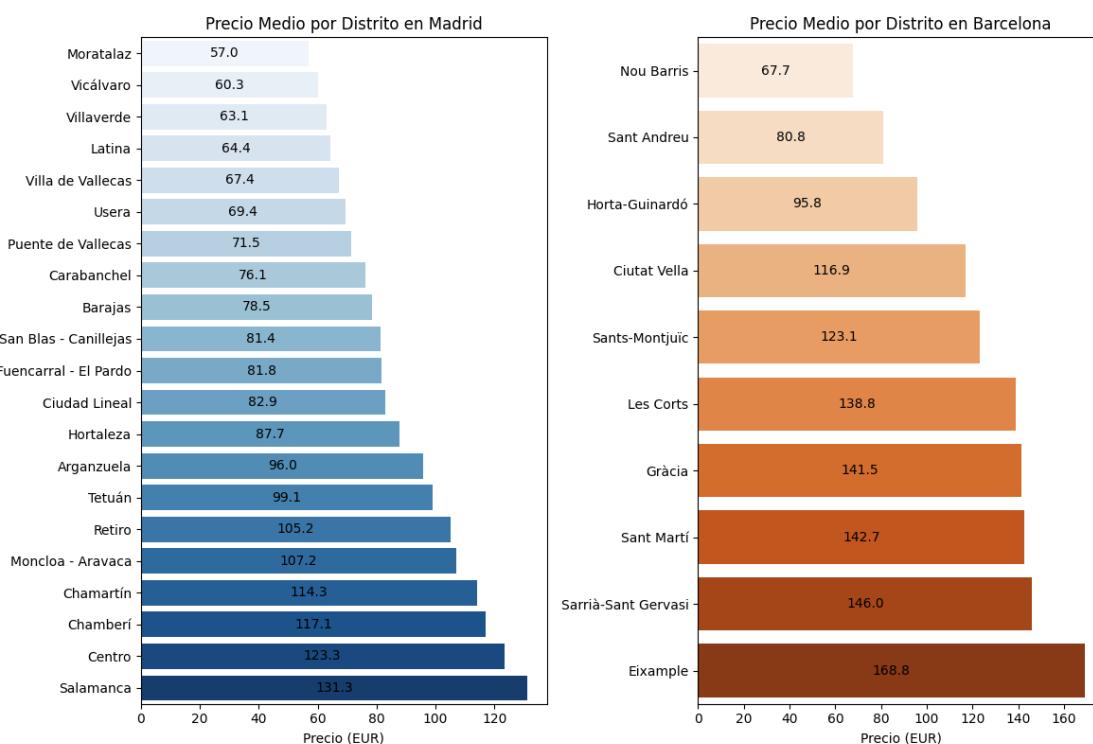
Por este motivo, consideramos separar el precio de los alojamientos en tres rangos diferentes: **económico**, **normal** y **caro**. Para establecer estos rangos, decidimos basarnos en el sueldo medio neto español, el cual se aproxima a los 1800€.

$$\frac{1800\text{€ por mes}}{30 \text{ dias}} = 60 \text{ €/dia}$$

De esta manera obtuvimos que el sueldo diario medio en España es de 60€, por lo que decidimos establecer los rangos entorno a este valor:

- **Económico:** Inferior o igual al sueldo diario (≤ 60).
- **Normal:** Entre uno y dos sueldos diarios (> 60 y ≤ 120).
- **Caro:** Más de dos sueldos diarios (> 120).

Para realizar las gráficas, utilizamos las variables `neighbourhood_group_cleansed` (distrito) y `price` (distrito). Entonces, lo primero que hicimos fue eliminar los valores atípicos para no obtener unos resultados engañosos y, luego, calculamos la media de precios para cada distrito de ambas ciudades, para así generar los siguientes gráficos:



Al observar el gráfico de **Madrid**, se puede destacar que los distritos **más económicos** son Moratalaz (57€), seguido por Vicálvaro (60.3€) y Villaverde (63.1€). En estos distritos, los precios medios están considerablemente por debajo de la media general, lo que hace de ellos una buena opción para personas con un presupuesto limitado. En el otro extremo, los distritos con los precios **más elevados** son Salamanca (131.3€), seguido por Centro

(123.3€) y por último Chamberí (117.1€). Estos distritos son conocidos por su alta demanda y su proximidad a zonas de interés turístico y comercial, lo que explica el incremento en el precio de los alojamientos.

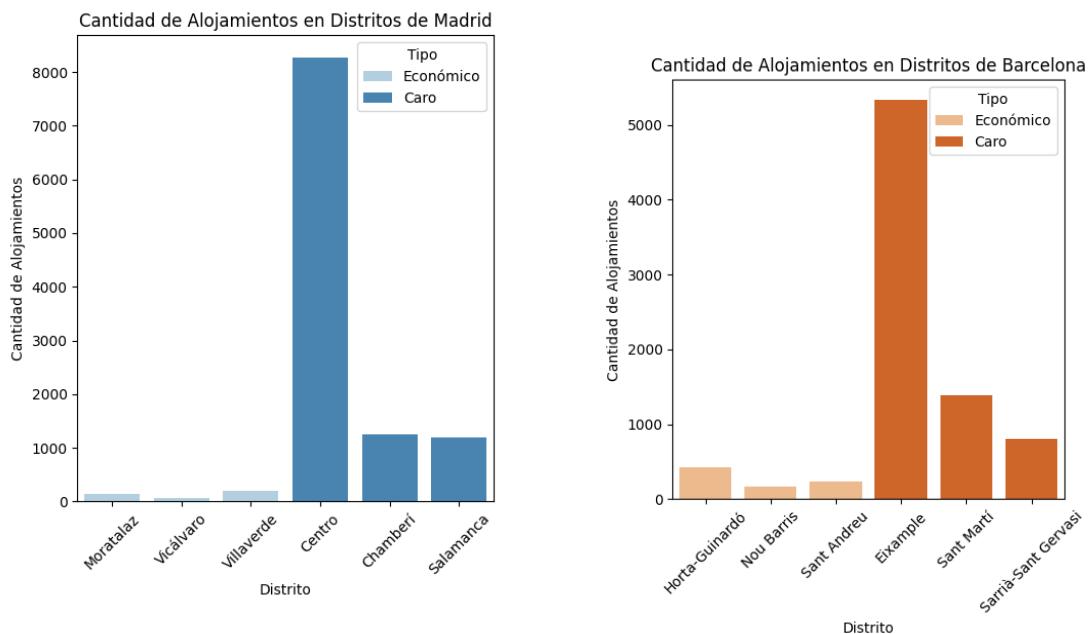
En **Barcelona**, los distritos **más asequibles** son Nou Barris (67.7€), seguido de Sant Andreu (80.8€) y Horta-Guinardó (95.8€). Estos distritos, al igual que en Madrid, están más alejados del núcleo urbano, lo que los convierte en opciones interesantes para aquellos con un presupuesto ajustado. Por otro lado, los distritos **más caros** son Eixample (168.8€), seguido de Sarrià-Sant Gervasi (146€) y Sant Martí (142.7€). Estas áreas, situadas en el centro de la ciudad, son muy demandadas debido a su proximidad a servicios, transporte y lugares de interés.

Sin embargo, también es importante considerar la cantidad de alojamientos en estos distritos, ya que esto puede influir en el precio.

Para reflejar esto, tenemos la cantidad de alojamientos calculados en el 2.1. Sin embargo,

vamos a centrarnos en los 3 alojamientos más baratos y en los 3 más caros de cada ciudad.

Para este estudio, además, sacamos de la ecuación los alojamientos con precios atípicos:



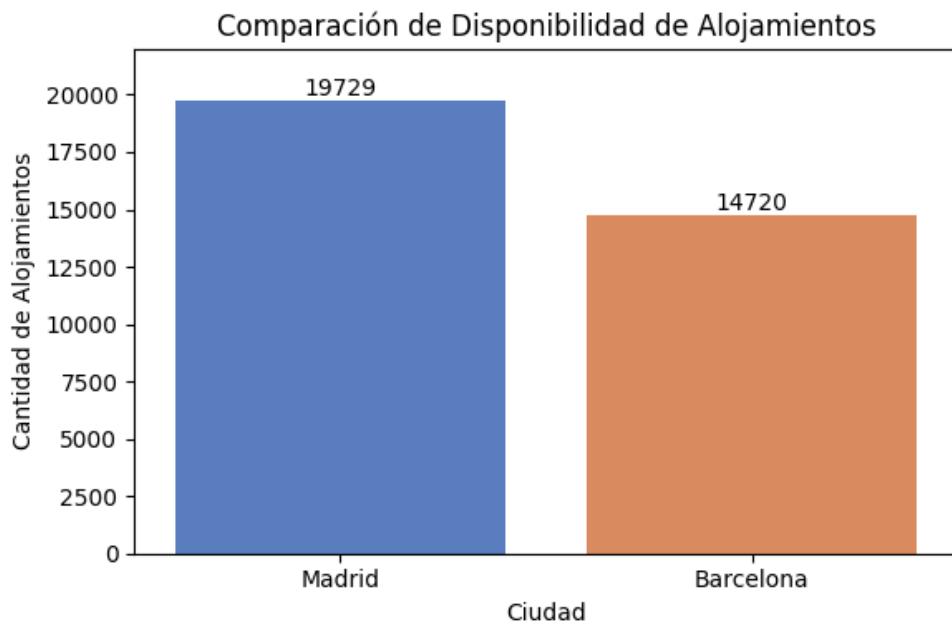
En **Madrid**, el distrito de Centro destaca con más de 8000 alojamientos, lo que refleja su popularidad y accesibilidad. En contraste, los distritos más económicos como Moratalaz y Vicálvaro tienen una cantidad de alojamientos significativamente menor. En **Barcelona**, Eixample alberga más de 5000 alojamientos, mientras que los distritos como Nou Barris presentan una oferta más limitada.

Estos datos ilustran cómo el precio de los alojamientos también varía en función de la cantidad de alojamientos que hay por distrito.

En conclusión, tanto en Madrid como en Barcelona los alojamientos más caros se concentran en las zonas centrales, donde hay más oferta, mientras que las opciones más económicas se encuentran en los distritos periféricos, donde hay menos oferta. Esta tendencia refleja la dinámica del mercado inmobiliario en ambas ciudades, donde la ubicación juega un papel crucial tanto en la determinación de los precios como en la oferta de alojamientos.

2.4 ¿Qué ciudad tiene más disponibilidad de alojamientos?

Otra consideración importante a la hora de comparar los alojamientos en diferentes ciudades es conocer la cantidad disponible de estos en cada una de ellas. Para esto, se ha decidido eliminar los valores atípicos y tener en cuenta solo aquellos alojamientos disponibles mediante la variable `has_availability`. Con esto hecho, generamos el siguiente gráfico:



En este se puede ver la comparación de la disponibilidad de alojamientos entre Madrid y Barcelona, donde **Madrid** cuenta con **casi 20.000** alojamientos, mientras que **Barcelona** presenta una disponibilidad de **casi 15.000** alojamientos.

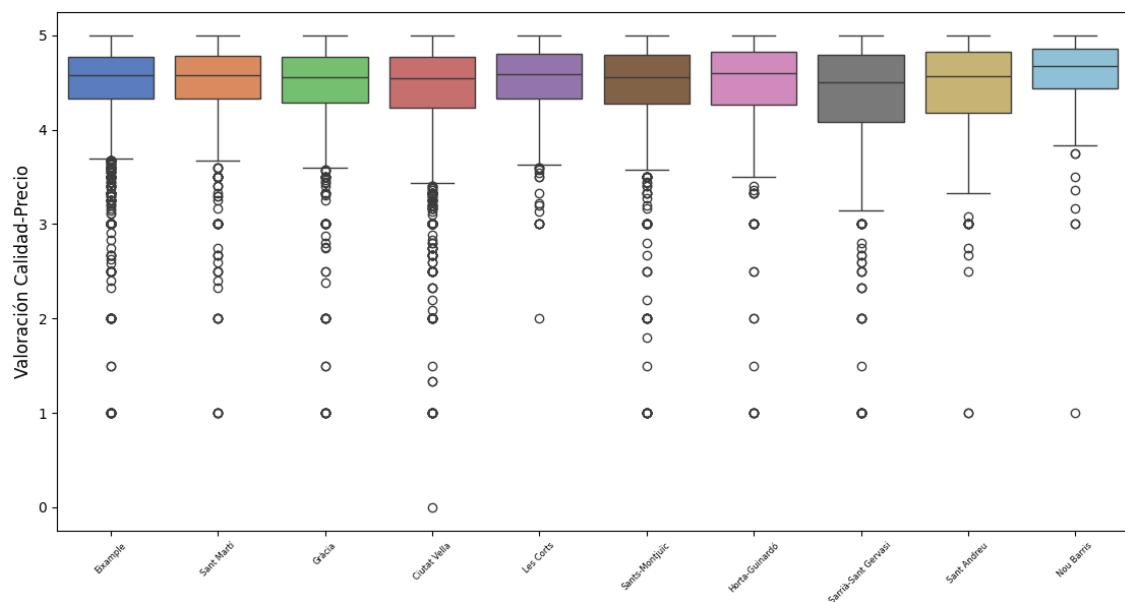
Esto indica que Madrid tiene aproximadamente un 33% más de alojamientos disponibles en comparación con Barcelona, lo que puede deberse a una mayor oferta en el mercado de alquiler o a una demanda más alta en esta ciudad.

2.5 ¿Dónde están los mejores alojamientos?

También nos resultó atractivo conocer qué distritos son los que mejores valoraciones calidad-precio tienen en sus alojamientos. Para analizar y representar este estudio, hicimos uso de los diagramas de caja (*boxplots*), ya que es una herramienta muy útil para visualizar la distribución de los datos y resume de forma sencilla varios estadísticos (mediana, IQR, percentiles, etc.).

Las variables utilizadas fueron la variable cualitativa `neighbourhood_group_cleaned` y la variable cuantitativa `review_scores_value`, que son los distritos y las valoraciones respectivamente.

Con Barcelona, obtuvimos el siguiente diagrama de caja:



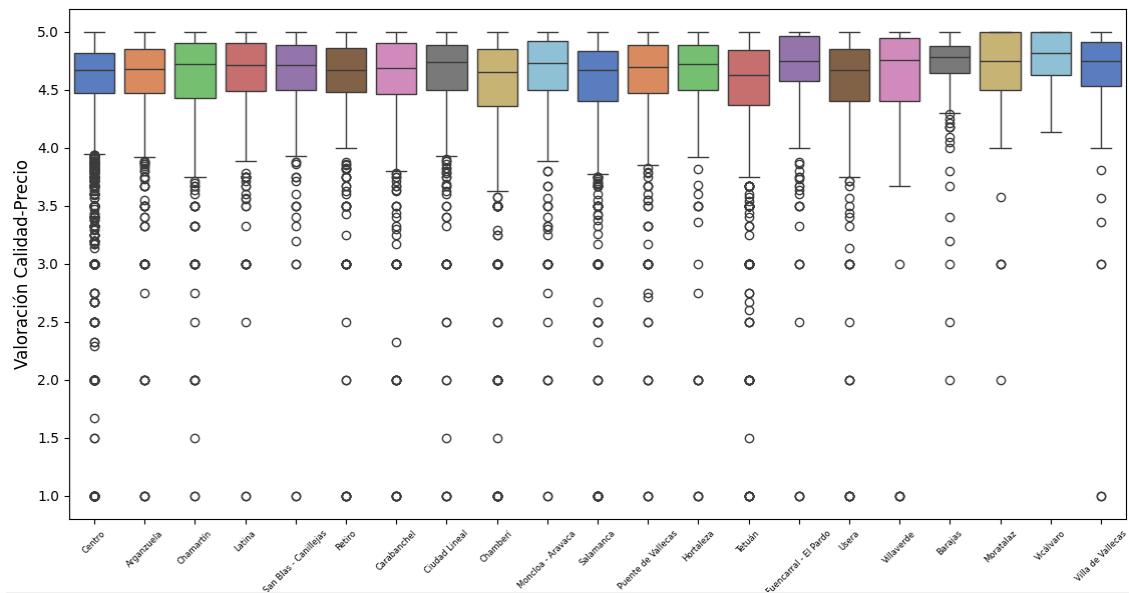
A primera vista y observando la mediana de los distritos, parece ser que el distrito que mejores valoraciones tiene es Nou Barris, y el que peores tiene es Sarrià-Sant Gervasi.

También podemos ver que hemos obtenido una distribución prácticamente simétrica (la línea de la mediana está en medio de la caja) en la mayoría de los distritos. Sin embargo, en distritos como Sarrià-Sant Gervasi, Sant Andreu e incluso en Horta-Guinardó, podemos ver una distribución un poquito más negativa.

También podemos contemplar cómo, en general, los distritos tienen una mayor dispersión de datos hacia abajo (los bigotes son más largos hacia abajo). Es decir, que tienen una mayor cantidad de valoraciones del 4.5 hacia abajo que del 4.5 hacia arriba.

En el diagrama de caja, además, podemos observar cómo la mayoría de los distritos tienen un gran número de *outliers* hacia abajo.

El diagrama de caja de Madrid fue el de a continuación:



A primera vista y observando la mediana de los distritos, parece ser que, en general, los distritos que mejores valoraciones tienen son Vicálvaro y Barajas. Sin embargo, podemos ver cómo distritos como Moratalaz o Vicálvaro tienen algún alojamiento que llega a una valoración de 4.9-5.0.

Por otro lado, los que peores valoraciones tienen parecen ser distritos como Tetuán, Chamberí y Usera.

En la capital también podemos observar que hemos obtenido una distribución prácticamente simétrica (la línea de la mediana está en medio de la caja) en la mayoría de los distritos. Sin embargo, en distritos como Chamartín, Villaverde y Chamberí, podemos ver una distribución un poco más negativa.

En este caso, también podemos observar cómo, en general, los distritos tienen una mayor dispersión de datos hacia abajo (los bigotes son más largos hacia abajo), además de *outliers* hacia abajo.

2.6 ¿Qué propietarios tienen más alojamientos?

También nos resultó interesante saber quiénes eran los 15 propietarios con más alojamientos de cada ciudad.

Para analizar esto, realizamos una tabla de distribución de frecuencia sobre la variable cualitativa `host_name`, que es el nombre de los propietarios. Sin embargo, consideramos que tal vez podrían repetirse nombres, por lo que decidimos también hacer uso de la variable `host_id`, la cual ponemos entre paréntesis después del nombre.

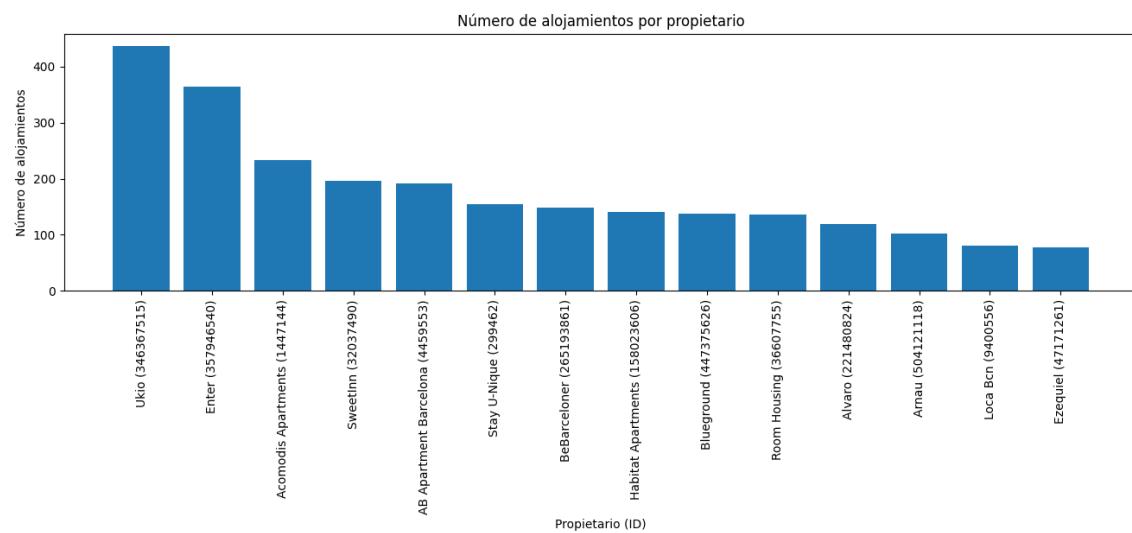
A continuación, podemos ver la tabla de la ciudad de Barcelona:

Propietario	Frecuencia	Frecuencia relativa	Porcentaje
Ukio (346367515)	437	0.168	16.84
Enter (357946540)	365	0.141	14.07
Acomodis Apartments (1447144)	233	0.09	8.98
SweetInn (32037490)	196	0.076	7.55
AB Apartment Barcelona (4459553)	191	0.074	7.36
Stay U-Nique (299462)	155	0.06	5.97
BeBarceloner (265193861)	149	0.057	5.74
Habitat Apartments (158023606)	141	0.054	5.43
Blueground (447375626)	138	0.053	5.32
Room Housing (36607755)	136	0.052	5.24
Alvaro (221480824)	119	0.046	4.59
Arnau (504121118)	102	0.039	3.93
Loca Bcn (9400556)	81	0.031	3.12
Ezequiel (47171261)	78	0.03	3.01
Alba (1391607)	74	0.029	2.85

Al observar los resultados, nos dimos cuenta de que los propietarios con más alojamientos eran empresas que se dedican a los alquileres. Las que más destacan son Ukio y Enter Coliving, con más de un 30% de alojamientos entre las dos.

También podemos ver a algunos particulares, pero ya con valores por debajo del 5%.

Para visualizar mejor los resultados, decidimos hacer una gráfica de barras:



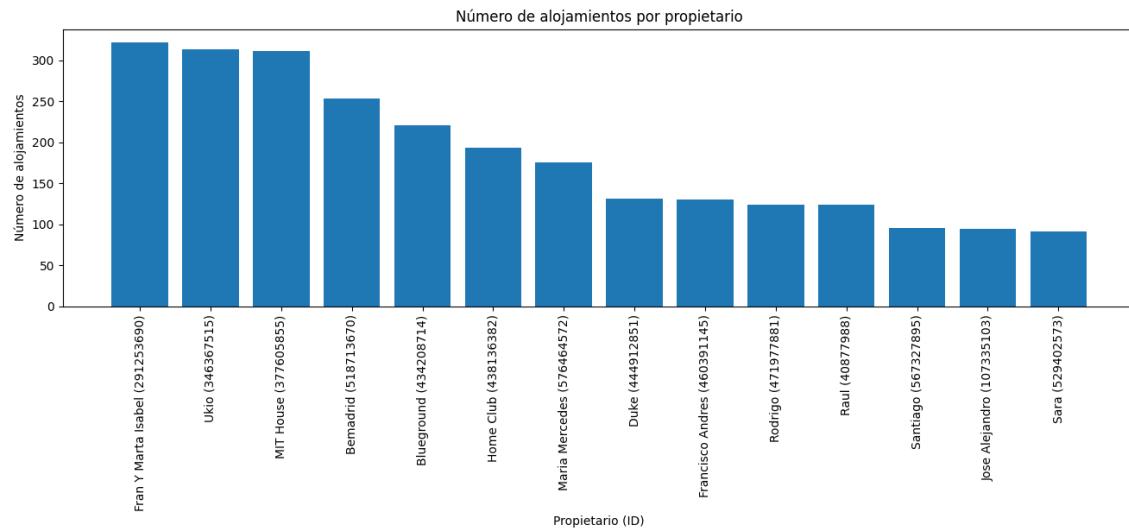
Después de analizar Barcelona, analizamos también los 15 propietarios de Madrid con más alojamientos, por lo que realizamos otra tabla de distribución de frecuencia:

Propietario	Frecuencia	Frecuencia relativa	Porcentaje
Fran Y Marta Isabel (291253690)	322	0.121	12.07
Ukio (346367515)	314	0.118	11.77
MIT House (377605855)	312	0.117	11.7
Bemadrid (518713670)	254	0.095	9.52
Blueground (434208714)	221	0.083	8.29
Home Club (438136382)	193	0.072	7.24
Maria Mercedes (576464572)	176	0.066	6.6
Duke (444912851)	131	0.049	4.91
Francisco Andres (460391145)	130	0.049	4.87
Rodrigo (471977881)	124	0.046	4.65
Raul (40877988)	124	0.046	4.65
Santiago (567327895)	95	0.036	3.56
Jose Alejandro (107335103)	94	0.035	3.52
Sara (529402573)	91	0.034	3.41
Nacho (391810099)	86	0.032	3.22

En los resultados de la capital, también podemos observar cómo empresas de alquiler reconocidas tienen un gran porcentaje de los alojamientos. De hecho, Ukio era la empresa con más alojamientos de Barcelona.

También podemos ver cómo en el primer puesto tenemos lo que parece ser una pareja de particulares que alquila su piso, pero investigando un poco por Internet, te das cuenta de que no es más que una empresa encubierta.

Al ver esto, no nos queda otra que dudar también sobre el resto de los nombres de propietarios que parecen particulares y tienen un gran número de alojamientos.



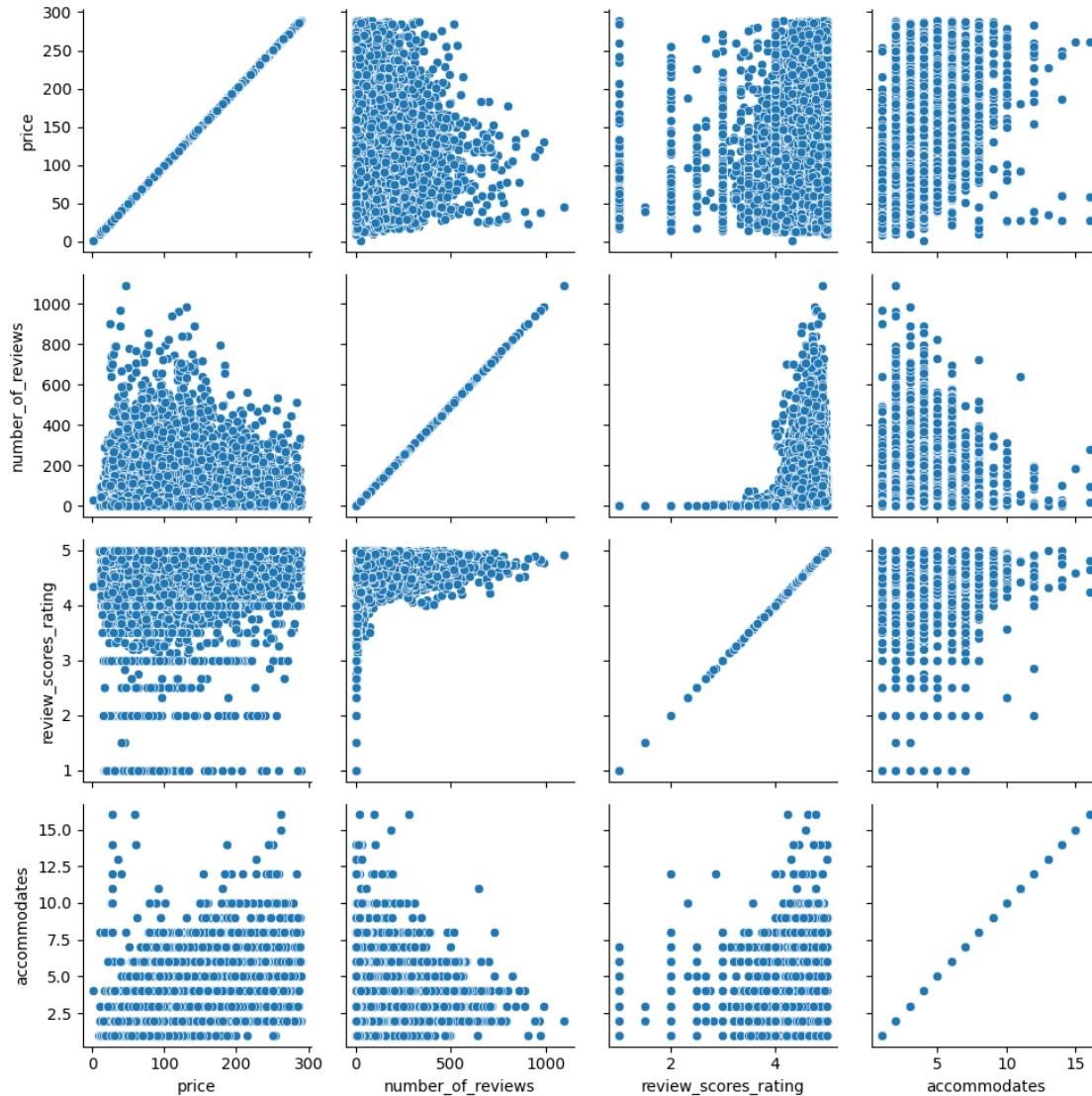
2.7 ¿Qué factores afectan al precio?

Este análisis tiene como objetivo explorar y comparar varias variables cuantitativas que influyen en el precio de los alojamientos en ambas ciudades. Se han seleccionado cuatro variables clave:

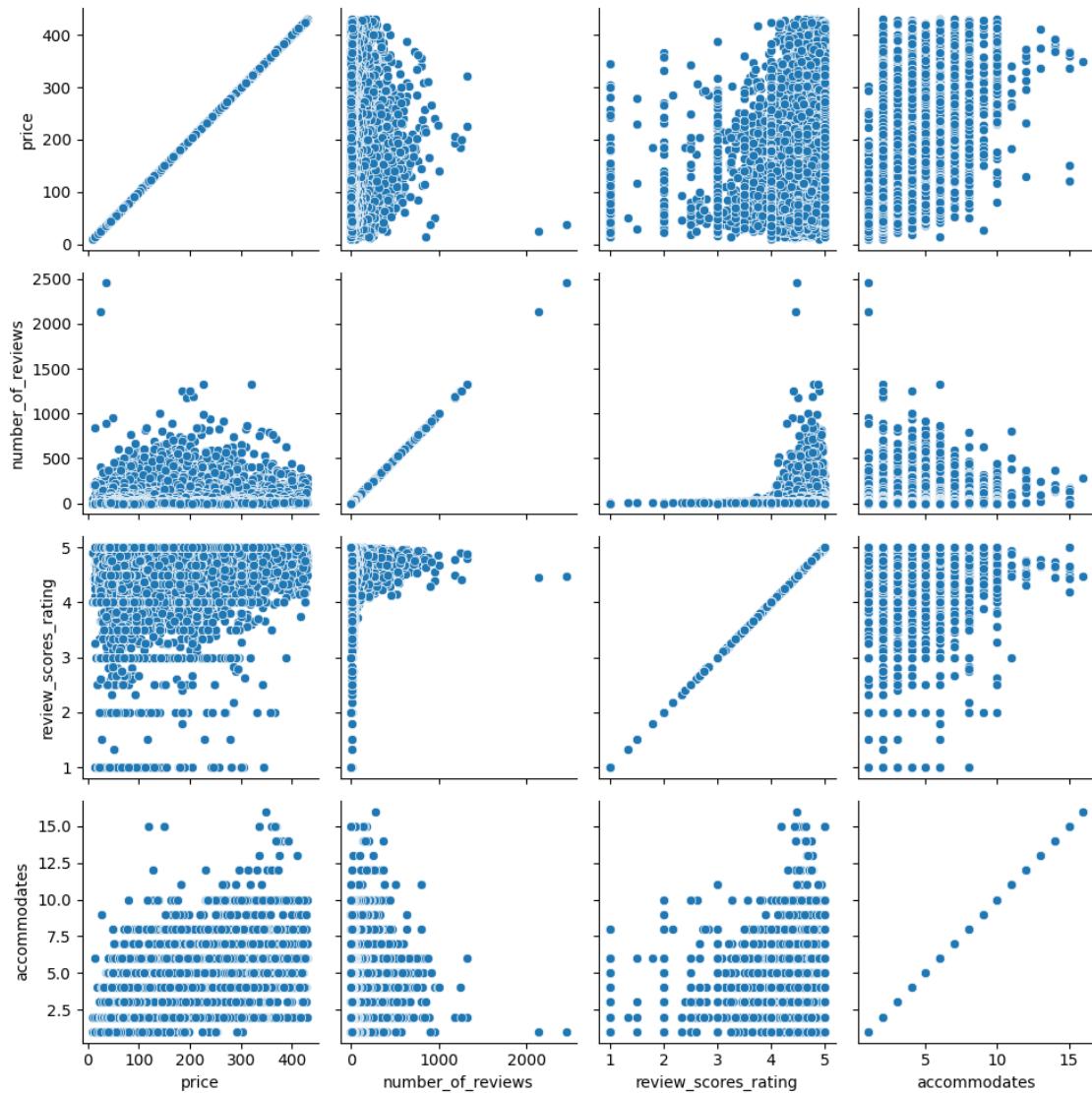
- price: precio del alojamiento.
- number_of_reviews: número de reseñas.
- review_scores_rating: valoración general de las reseñas.
- accommodates: capacidad máxima de inquilinos.

A través de la creación de matrices de correlación, se busca comprender cómo estas variables interactúan y afectan el valor de los alojamientos. Eso sí, eliminando aquellos alojamientos con precios atípicos.

Entonces, para **Madrid** obtuvimos la siguiente matriz de correlación:



Por otro lado, en **Barcelona** se puede observar la siguiente matriz de correlación:



Los coeficientes de correlación de Pearson en **Madrid** fueron:

	price	number_of_reviews	review_scores_rating	accommodates
price	1.0	0.03	0.03	0.61
number_of_reviews	0.03	1.0	0.1	0.09
review_scores_rating	0.03	0.1	1.0	-0.04
accommodates	0.61	0.09	-0.04	1.0

En **Barcelona** en cambio, fueron:

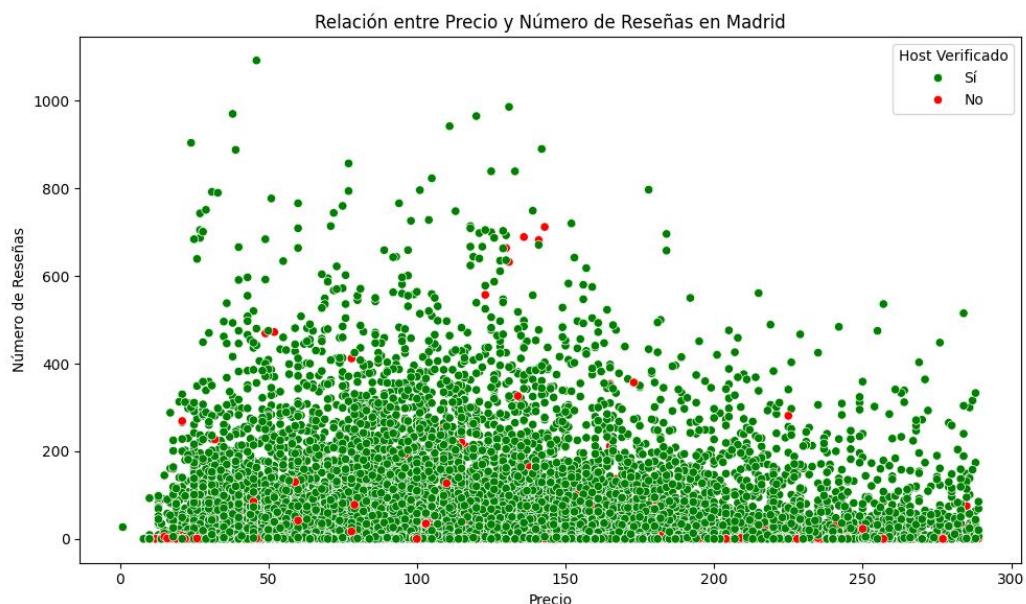
	price	number_of_reviews	review_scores_rating	accommodates
price	1.0	0.23	0.06	0.62
number_of_reviews	0.23	1.0	0.12	0.16
review_scores_rating	0.06	0.12	1.0	-0.04
accommodates	0.62	0.16	-0.04	1.0

En Barcelona se puede observar el aumento de la correlación en casi todos los pares de variables, aunque en ambas ciudades solo se ve una fuerte correlación entre **el precio y el máximo de huéspedes**.

Debido a que en estos resultados apenas observamos fuertes correlaciones entre las variables, decidimos entrar más en detalle e intentar buscar más variedades en los estudios en los siguientes puntos.

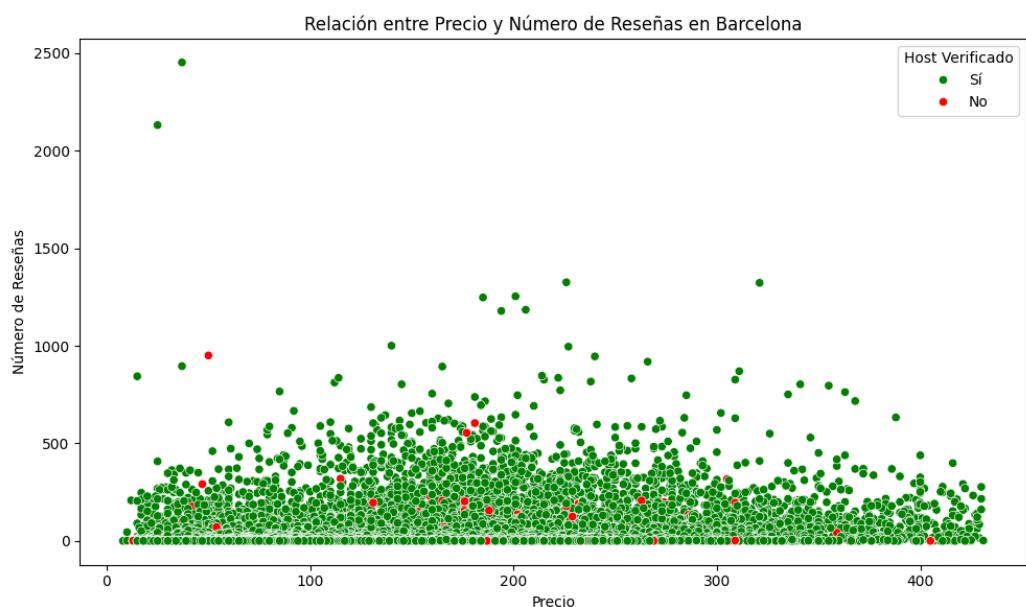
2.7.1 ¿Los alojamientos más caros tienen más reseñas?

La primera relación que analizamos más en detalle fue entre el precio y la cantidad de reseñas. Para ir un poco más allá, introducimos también si el propietario estaba o no verificado en la aplicación (`host_identity_verified`), para ver si esto tenía impacto en la demanda de alojamientos. Luego, generamos el siguiente gráfico para **Madrid**:



Lo primero que se puede observar es que el hecho de estar o no verificado, no tiene gran importancia, ya que, aunque haya muy pocos no verificados, estos están repartidos por todo el gráfico. Por otro lado, se puede ver cómo los alojamientos más baratos tienden a tener un mayor número de reseñas y, a medida que el precio va subiendo, la cantidad de reseñas es cada vez menor y más dispersa.

En **Barcelona**, obtuvimos los siguientes resultados:



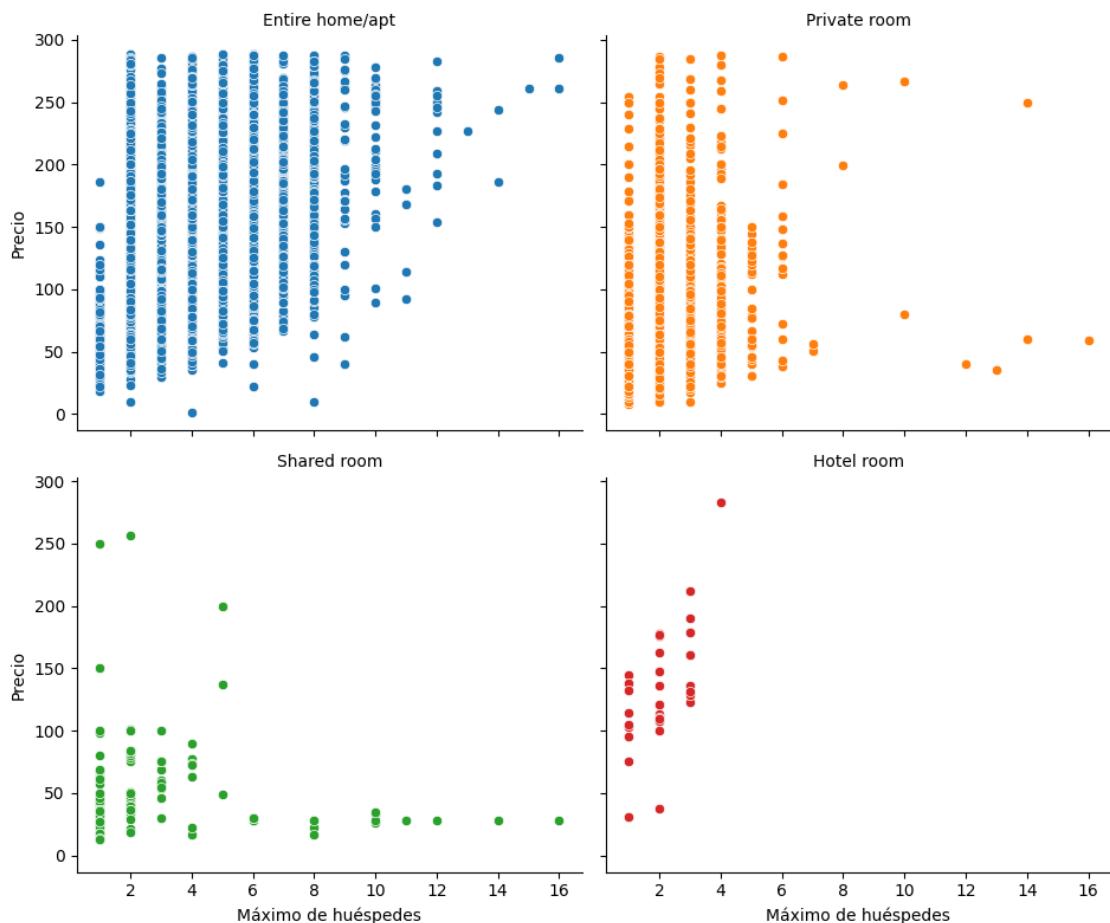
Al igual que en Madrid, el estar o no verificado no es un problema a la hora de alquilar un alojamiento, ya que los pocos que hay también están distribuidos por todo el gráfico. En cambio, se puede ver como la relación entre el precio y el número de reseñas es uniforme. Aunque el precio vaya aumentando, la cantidad y la dispersión del número de reseñas son prácticamente iguales.

2.7.2 ¿El máximo de huéspedes afecta al precio de los alojamientos?

Otro análisis importante fue determinar si la capacidad máxima de huéspedes tiene un impacto directo en el precio del alojamiento. Asimismo, decidimos separar las relaciones en función del tipo de habitación (room_type), ya que no es lo mismo un apartamento entero para 4 personas que una sola habitación también para 4 personas. Esta variable se encuentra en todos los alojamientos y permite 4 opciones:

- **Entire home/apt:** Apartamento entero.
- **Private room:** Habitación privada (alquilas la habitación).
- **Shared room:** Habitación compartida (alquilas una cama de la habitación).
- **Hotel room:** Habitación de hotel.

Entonces, obtuvimos las siguientes comparaciones en **Madrid**:

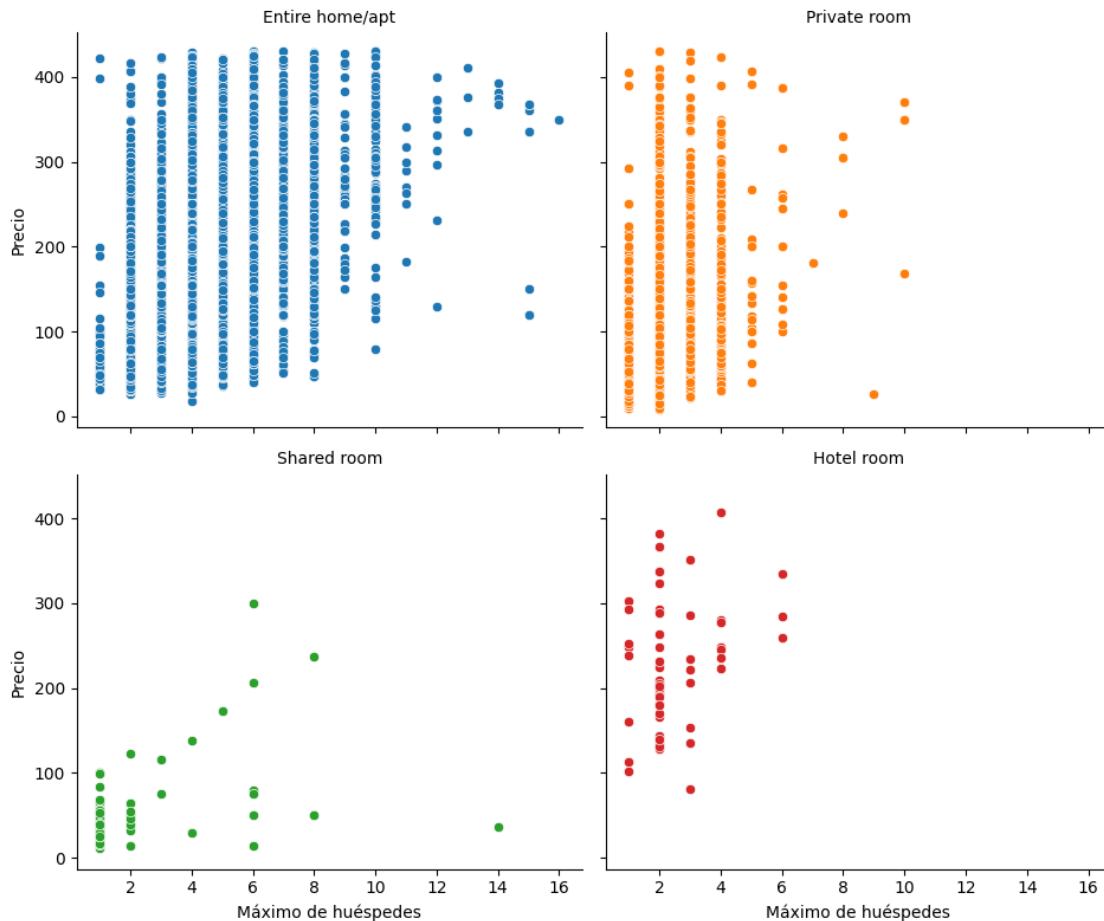


En **los apartamentos enteros** se puede ver cómo llegado a un punto, el precio empieza a subir a medida que la capacidad máxima también va aumentando. Aproximadamente hasta los 8 huéspedes, existe una gran variedad de alojamientos de todos los precios y, pasados esos 8, las opciones de alojamientos baratos van desapareciendo.

En las **habitaciones privadas**, a partir de los 6 huéspedes ya casi no hay alojamientos, mientras que, en las **habitaciones compartidas**, los precios van disminuyendo a medida que la capacidad máxima va aumentando. Esto último es comprensible, ya que con cuantos más huéspedes haya, con más personas vas a tener que estar compartiendo la habitación.

Por último, en **las habitaciones de hotel**, se puede ver como el máximo es solamente de 4 personas y el precio va de la mano con esta capacidad máxima.

Por otro lado, en **Barcelona** obtuvimos los siguientes resultados:



En **los apartamentos enteros**, se observa el mismo comportamiento que en Madrid, donde llegados a los 8 huéspedes, la oferta de alojamientos baratos desaparece.

Lo mismo pasa para **las habitaciones privadas**. Al igual que en Madrid, a partir de los 6 huéspedes, la oferta de alojamientos es casi inexistente. En cambio, en **las habitaciones compartidas**, se puede ver que no hay tanta oferta y que los precios van incrementando a medida que la capacidad máxima lo va haciendo también.

Por último, en **las habitaciones de hotel**, se ve una mayor oferta a partir de los 4 huéspedes, la cual llega hasta los 6. Aun así, el precio también va un poco de la mano con esta capacidad máxima, aunque no tanto como en Madrid.

Estas conclusiones se pueden observar también con los coeficientes de correlación de Pearson entre el precio y el máximo de huéspedes, separando los datos en función del tipo de habitación:

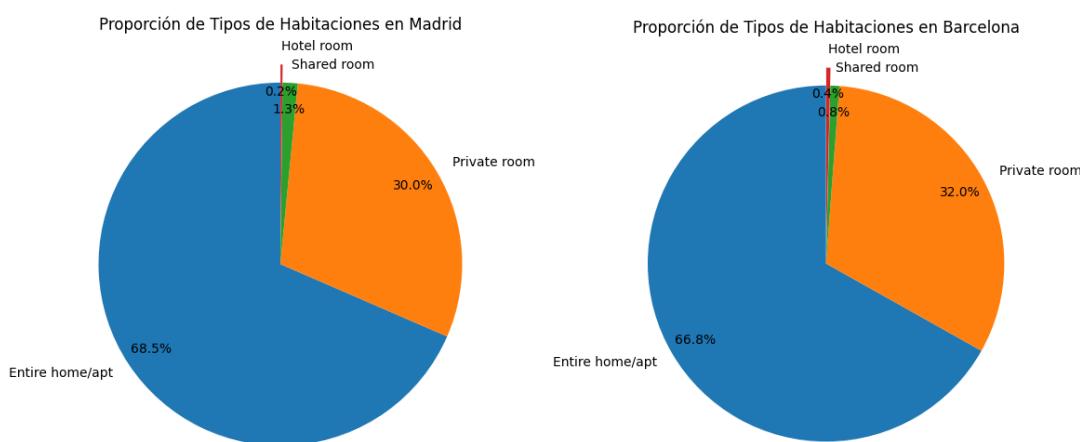
Madrid

Coeficiente de Correlación	
Entire home/apt	0.46
Hotel room	0.64
Private room	0.35
Shared room	0.02

Barcelona

Coeficiente de Correlación	
Entire home/apt	0.49
Hotel room	0.27
Private room	0.49
Shared room	0.42

Por último, observando la densidad de puntos en los gráficos de cada ciudad, se puede intuir que ambas presentan una distribución similar de los alojamientos en función del tipo de habitación. Esto último se puede confirmar con los siguientes gráficos:



En esta comparación, se puede observar cómo la distribución de las dos ciudades es casi idéntica, con proporciones comparables de alojamientos en cada categoría.

Esta similitud sugiere que, a pesar de las diferencias culturales y económicas entre las dos ciudades, los propietarios ofrecen opciones de alojamiento de manera uniforme, lo que podría reflejar una tendencia en el mercado que favorece configuraciones de habitación similares en contextos de grandes ciudades.

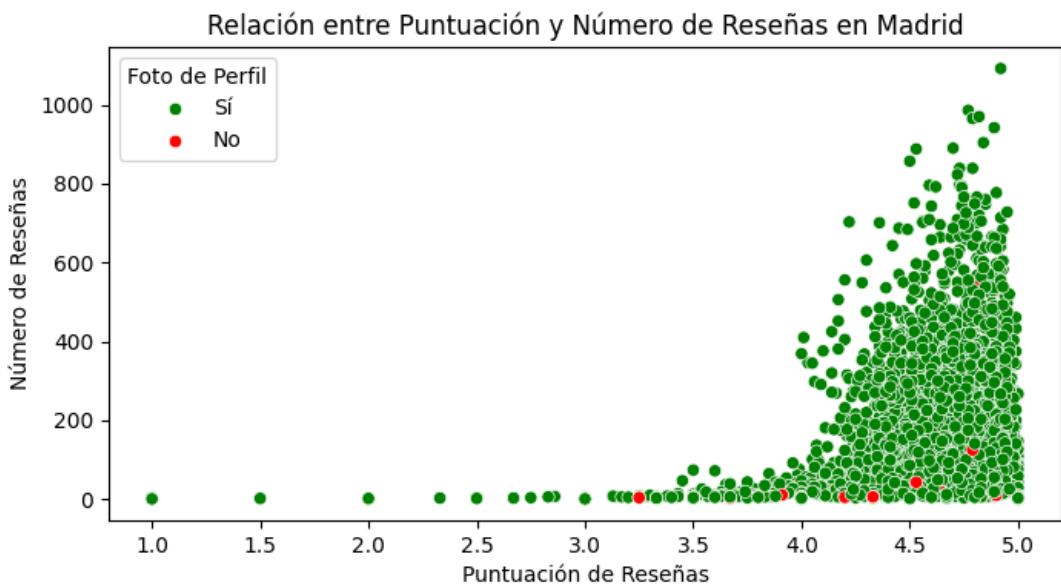
2.7.3 ¿Los alojamientos mejor valorados tienen más reseñas?

Otra relación que nos pareció interesante analizar fue la que podría haber entre la puntuación general y el número de reseñas.

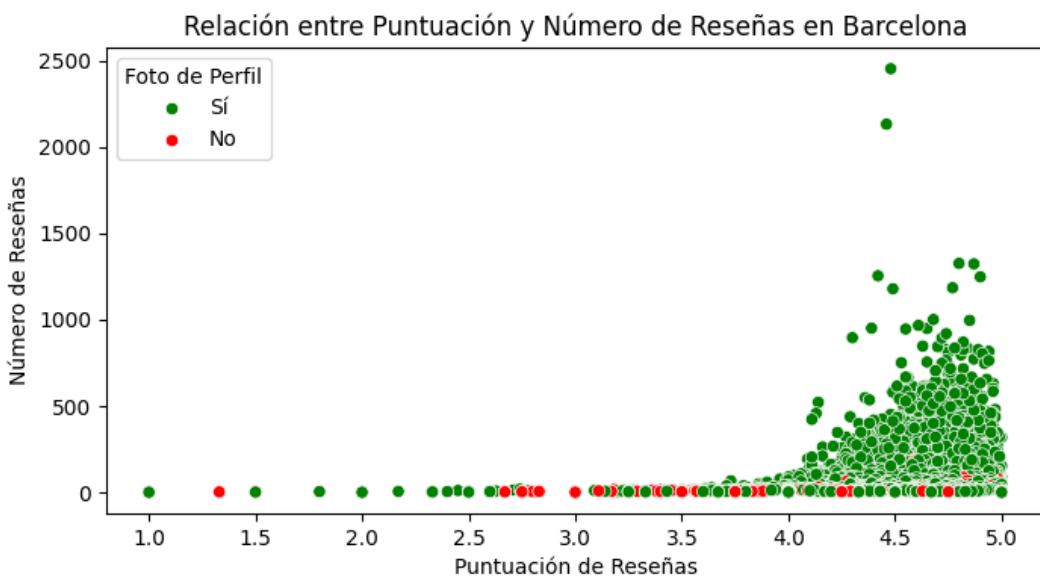
Este estudio pretende revelar si los alojamientos con mejores puntuaciones son también los que tienen más reseñas o, al contrario, aquellos con menos puntuación son los que tienen menos.

Además, añadimos la variable `host_has_profile_pic`, para ver si el hecho de que el propietario tenga foto de perfil es algo para tener en cuenta a la hora de alquilar un alojamiento.

Entonces, obtuvimos el siguiente gráfico para la ciudad de **Madrid**:



Y para **Barcelona**, los resultados fueron los siguientes:



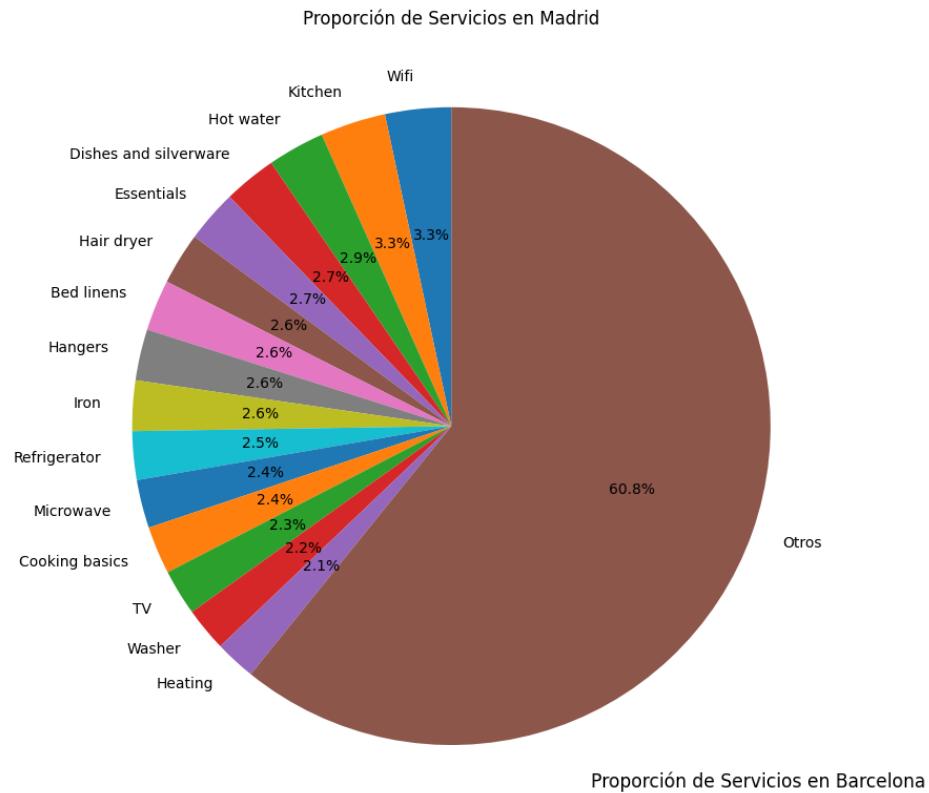
En ambas ciudades se puede observar el mismo patrón, en el que los alojamientos con puntuaciones superiores a 4 son los que tienen una mayor cantidad de reseñas. A partir de esta puntuación, se puede ver también cómo el número de reseñas va incrementando a la vez que lo va haciendo la puntuación.

Se puede ver también que los propietarios sin foto de perfil son una minoría en ambas ciudades, aunque en Barcelona frecuentan más que en Madrid.

Además, el hecho de tener o no foto, no es un indicativo para desconfiar de que vaya a ser un mal alojamiento, ya que estos tienen puntuaciones altas también. Asimismo, en lo que sí se ve afectado es en el número de reseñas, ya que la gran mayoría de ellos tienen un número ínfimo de estas.

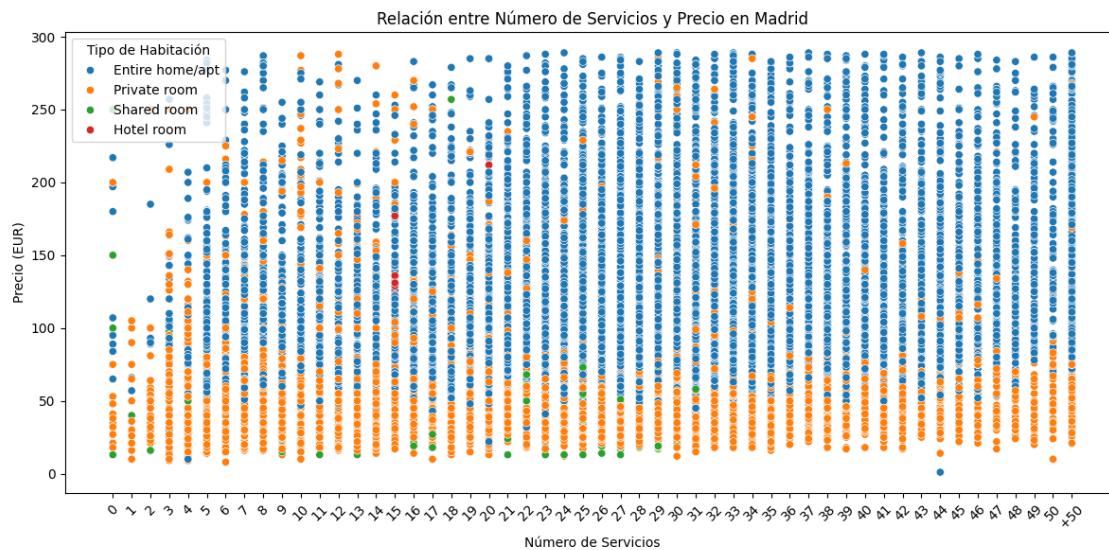
2.8 ¿Los alojamientos más caros ofrecen más servicios?

Otra variable que decidimos analizar fue la de **amenities**, la cual indica los servicios que ofrecen los alojamientos. Esta variable fue un poco más difícil de tratar, ya que estaba en formato JSON, por lo que había que separarlo para obtener los diferentes servicios. Entonces, lo primero que hicimos fue obtener la distribución de los servicios de ambas ciudades, enfocándonos en los 15 más frecuentados.



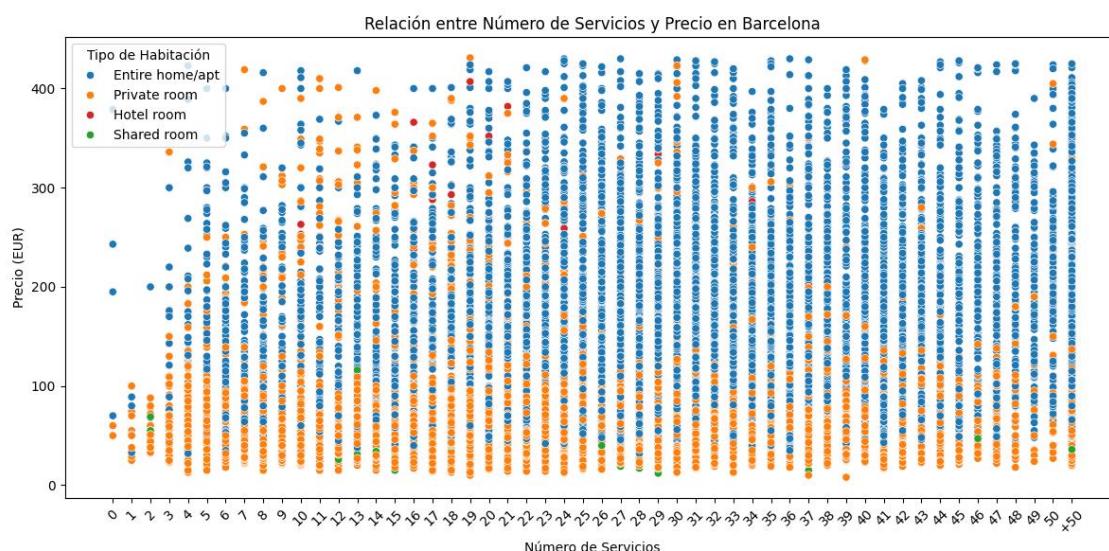
Como se puede observar, en ambas ciudades las distribuciones son prácticamente idénticas, con exactamente los mismos 15 servicios más frecuentados. Vistos un poco por encima los servicios que más se pueden encontrar en ambas ciudades, decidimos hacer otro gráfico para cada ciudad para poder ver si existe una relación entre el precio de los alojamientos y el número de servicio que incluyen.

Además, también añadimos el tipo de habitación (room_type) de cada alojamiento, para ver si esto también tenía relación con el número de servicios del alojamiento. Entonces, los resultados de **Madrid** fueron los siguientes:



En este gráfico se puede observar que el número de servicios no tiene una implicación directa con el precio, ya que, tanto en los apartamentos enteros como en las habitaciones privadas, que son los más representados, el precio se mantiene a medida que se van añadiendo servicios. Lo mismo pasa con las habitaciones compartidas y las de hotel, aunque hay muy pocos datos como para poder sacar conclusiones.

Por otro lado, los resultados de **Barcelona** fueron prácticamente los mismos:



Esta poca relación entre el precio y el número de servicios de ambas ciudades también se puede observar con los coeficientes de correlación de Pearson, donde ninguno llega a superar el 25%:

Madrid

room_type	Coeficiente de Correlación
Entire home/apt	0.15
Hotel room	0.24
Private room	0.05
Shared room	-0.12

Barcelona

room_type	Coeficiente de Correlación
Entire home/apt	0.2
Hotel room	-0.18
Private room	0.04
Shared room	-0.0

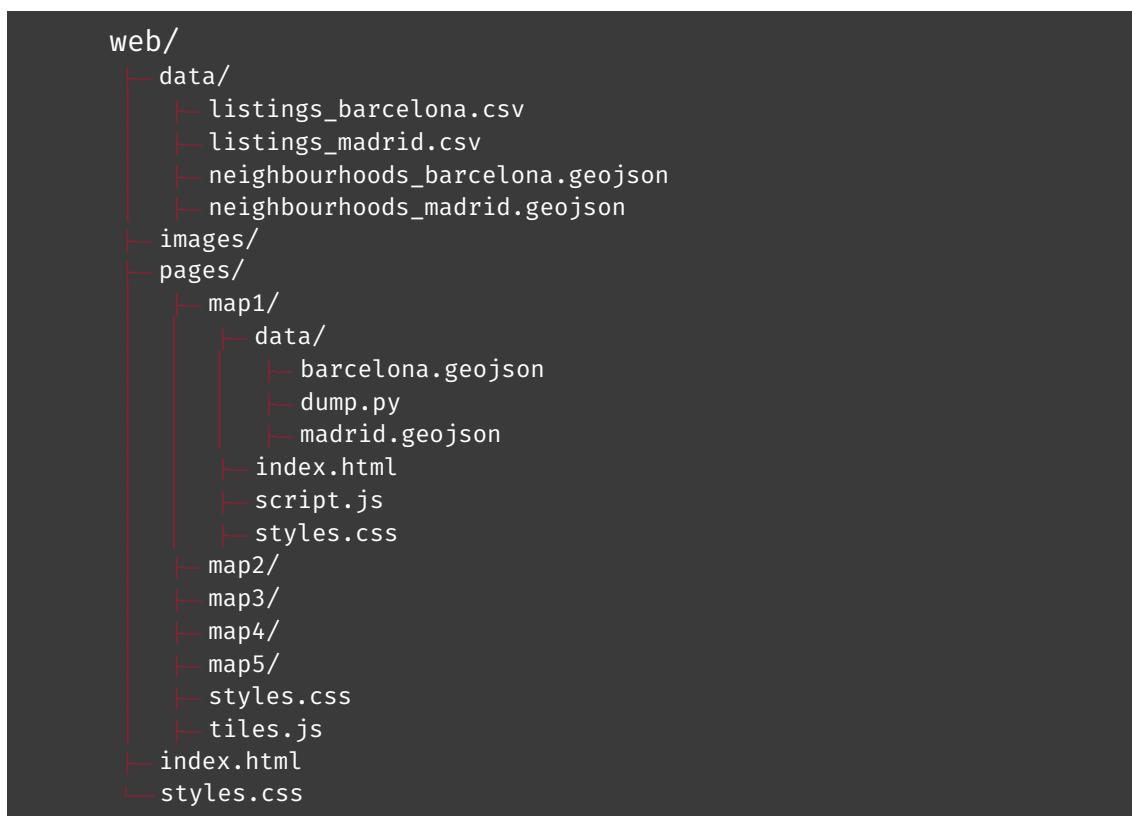
Lo curioso de estos coeficientes es la contrariedad que existe entre las habitaciones de hotel de ambas ciudades. En **Madrid**, el valor es de 0.24, lo que indica una relación directa, donde los precios aumentan juntamente con la cantidad de servicios, mientras que en **Barcelona** este valor es de -0.18, indicando el efecto contrario.

3. Visualizaciones geográficas

La segunda parte de la práctica consistía en realizar un análisis geográfico con los datos de los alojamientos de Airbnb.

Un análisis geográfico es un proceso que utiliza datos relacionados con la ubicación para explorar, comprender y visualizar la distribución espacial de un fenómeno. En este tipo de análisis, se busca identificar patrones, tendencias y relaciones en función del lugar donde ocurren, lo que permite interpretar cómo la geografía influye en aspectos específicos, como el comportamiento de los usuarios, los precios o la demanda de servicios en diferentes zonas.

Para hacer este análisis decidimos hacer una web estática usando los lenguajes básicos **html, css y javascript**. Para entender un poco el proyecto, esta sería su estructura general:



El primer nivel de directorio contiene lo siguiente:

- **data**: contiene los ficheros con datos base descargados de Airbnb. Tanto los tabulares como los geográficos de los barrios.
- **images**: contiene todas las imágenes usadas en la web.
- **pages**: contiene cada uno de los mapas, donde cada mapa es una página.
- **index.html**: contenido de la página principal.
- **styles.css**: estilos de la página central.

Entonces, dentro de la carpeta de mapas (**pages**), encontramos un directorio para cada mapa y dos ficheros más: **styles.css**, para tener los mismos estilos en toda la web, y **tiles.js**, que es el código que se utiliza en todos los mapas (TMS, zoom, capas, etc.) y que se explicará más adelante.

Dentro de los directorios de mapas, se encuentra la misma estructura en todos:

- **data**: contiene el script *Python* para hacer el volcado de los datos que queremos de los *.csv* hacia un nuevo *.geojson* para cada ciudad. Por lo que en este directorio es donde se encuentran los archivos que se usarán para visualizar los datos en los mapas.
- **index.html**: contenido de la página.
- **script.js**: código extra que solo funciona en esa página.
- **styles.css**: estilos extra de la página.

Tile Map Service (TMS)

Lo primero que hace el script **tiles.js**, es cargar todos los TMSs en formato de capas de **Leaflet**, que es la librería que hemos decidido usar para generar los mapas. Entonces, lo que hace es asignar una variable a cada una de estas capas. Por ejemplo, con la de *OpenStreetMaps* sería de la siguiente forma:

```
const openstreetmapsLayer = L.tileLayer(  
    "https://s.tile.openstreetmap.org/{z}/{x}/{y}.png",  
    {  
        minZoom: 2,  
        maxZoom: 20,  
    }  
)
```

Además del TMS, también le asignamos el rango de *zoom* permitido. Para obtener todos estos TMSs, usamos la siguiente web:

<https://leaflet-extras.github.io/leaflet-providers/preview/>

Una vez cargados todos los servicios, inicializa el mapa con una vista centrada en España y establece un TMS por defecto.

```
// Inicializar el mapa centrado en España  
const map = L.map("map").setView([40.4168, -3.7038], 6);  
  
// Añadir TMS por defecto  
currentLayerName = "alidade-satellite";  
currentLayer = StadiaAlidadeSatelliteLayer;  
currentLayer.addTo(map);
```

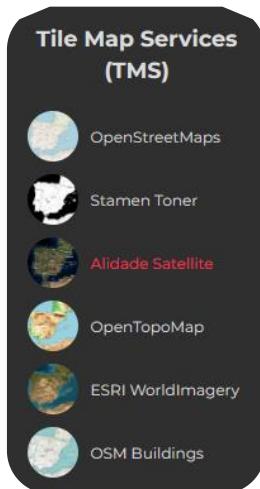
Este script cuenta también con una función para cambiar el TMS, lo que nos permite poder cambiarlo desde otro script diferente, que principalmente será el propio de cada mapa. Esta función se llama **changeMapLayer()** y tiene como único argumento el nombre del TMS al que se quiere cambiar. El funcionamiento de esta es muy simple:

1. Se elimina la capa con el TMS actual.
2. Se obtiene el nuevo TMS en función del nombre pasado a la función. Hay que recordar que están todos guardados en variables por lo que solo se obtiene la variable con la capa ya inicializada.
3. Se carga la nueva capa al mapa.

La única complicación es cuando se pone o se quita la de *OSM Buildings*, ya que esta está formada por dos capas, la del TMS y la de los edificios 3D. Si se quita, lo único que hay que hacer es eliminar también la de los edificios. En cambio, si lo que se quiere es ponerla, el flujo es el siguiente:

1. Se guardan y eliminan todas las capas diferentes del TMS.
2. Se elimina la capa con el TMS actual.
3. Se añaden tanto la capa con el TMS de *OSM Buildings* como la capa con los edificios en 3D.
4. Se añaden las capas eliminadas en el primer paso.

Esto se hace para evitar que la capa con los edificios quede por encima de otras capas como podría ser la de calor. Para poder cambiar el TMS en los mapas, se utiliza el siguiente menú que aparece al lado derecho de los mapas.



Centrar el mapa

Otra funcionalidad que añade el script ***tiles.js*** es la posibilidad de centrar el mapa en una posición específica. Al igual que antes, de esto se encarga una función que se puede llamar desde cualquier otro fichero que tenga este importado. La función se llama ***centerMap()*** y acepta tres argumentos:

- ***latitude***: posición en grados respecto el ecuador.
- ***longitude***: posición en grados respecto el meridiano de Greenwich.
- ***zoom***: valor del *zoom* en el mapa.

Lo que decidimos hacer para este efecto es eliminar todas las capas diferentes del TMS durante el movimiento, por lo que, antes que nada, guarda y elimina todas las capas superiores a la del TMS. También decidimos usar la función ***flyTo()*** en lugar de ***setView()***, ya que la primera hace un desplazamiento suave, mientras que la segunda solo carga las nuevas coordenadas sin ningún efecto de desplazamiento.

Una vez terminado el desplazamiento, nos dimos cuenta de un pequeño *bug*, el cual era que, si hacías *zoom* después de haber centrado el mapa, este se teletransportaba a unas coordenadas aleatorias. Para arreglarlo, decidimos hacer un ***setView()*** al terminar el ***flyTo()*** para asegurarnos de que el mapa estaba en las coordenadas a las que queríamos ir. A parte de eso, también cargamos las capas eliminadas al principio.

Para poder usar esta funcionalidad en los mapas está este menú encima de ellos:



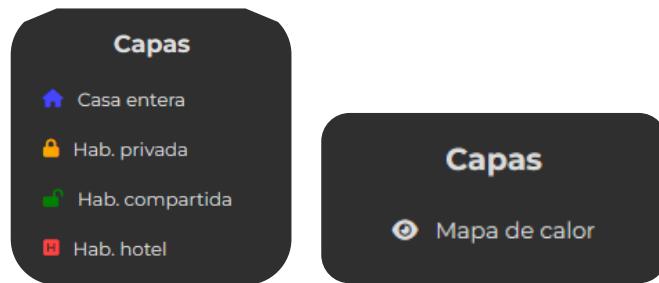
Tienes dos opciones, o ir a puntos ya predefinidos (Madrid, Barcelona, España) o ir a un punto personalizado indicando sus coordenadas. En el segundo caso, tanto la latitud como la longitud son obligatorias, mientras que el zoom se mantendrá igual en caso de que no lo indiques.

Visibilidad de capas

La última funcionalidad que proporciona el script **tiles.js** es la de cambiar la visibilidad de las capas diferentes a la del TMS. Esta funcionalidad se puede conseguir llamando a la función **toggleLayerView()**, pasándole como argumento la capa a la que quieras cambiar la visibilidad.

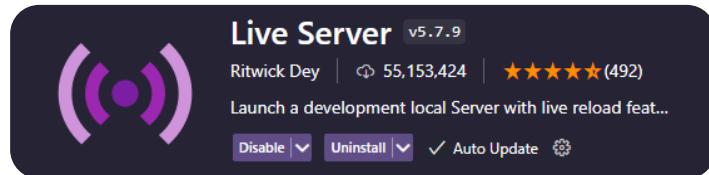
Su funcionamiento es muy básico, únicamente mantiene una constancia sobre las capas que no están visibles usando un *array*. Entonces, cuando el usuario desactiva una capa, esta se guarda en la variable y se elimina del mapa. Si, por el contrario, el usuario activa una capa, esta se quita de la variable y se añade al mapa.

Esta funcionalidad se puede usar con los paneles llamados “Capas”, que aparecen al lado derecho de los mapas y encima de los TMSs.



Web

Por último, la forma que recomendamos y hemos estado usando nosotros para ver y probar la página, usando VSCode juntamente con la extensión **Live Server**.



Lo que hace esta extensión es crear un servidor local para alojar archivos web, además de permitir una recarga en tiempo real, donde los cambios hechos en el código se actualizan al instante en la página web. Para hacerlo, solo hay que tener el proyecto abierto en el VSCode y darle al botón de la parte inferior derecha que dice “Go Live”.



3.1 Mapa de calor de la distribución de precios

El primer análisis geográfico que hicimos fue el de la distribución de precios de los alojamientos. Para ello, utilizamos un mapa de calor, donde las áreas más intensas indican zonas con mayores concentraciones de precios elevados, mientras que las áreas más suaves reflejan precios más bajos o menos densidad de alojamientos.

Generación de los GeoJSON

Para este estudio únicamente usamos las variables `latitude` y `longitude` para situar los alojamientos en el mapa, y la variable `price` para identificar las zonas más caras y baratas. Para hacer esto usamos el script `dump.py`, el cual obtiene los datos de los `.csv` y limpia la columna del precio para convertirlos a decimales.

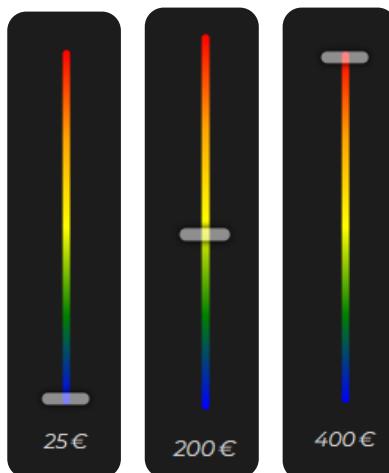
El script crea los GeoJSON desde cero para cada ciudad con una colección de `Features`, donde cada entidad es de tipo `Point` con la longitud y la latitud como coordenadas. Además, añadimos el precio a las propiedades para poder luego en el *frontend* generar el mapa de calor.

Funcionalidades

Este mapa cuenta con dos funcionalidades extra de interactividad. La primera es un botón que llama a la función `changeLayerView()` para habilitar o deshabilitar la capa de calor. Esta función solo funciona si el mapa no se está moviendo, es decir, si no se está haciendo la animación de movimiento al centrarla en un lugar específico. Entonces, cambia los estilos del botón y llama a la función `toggleLayerView()` explicada en la parte de [visibilidad de capas](#).



La segunda funcionalidad consiste en poder regular el precio a partir del cual se calcula la intensidad de las zonas de calor. Para hacer esto, hemos añadido un *slider* que va desde los 25€ hasta los 400€, donde cuanto más alto sea el precio, menos zonas rojas habrá en el mapa. Entonces, cuando se desplaza el *slider*, lo que se hace es eliminar la capa de calor actual y generarla de nuevo con la intensidad especificada.



Resultado

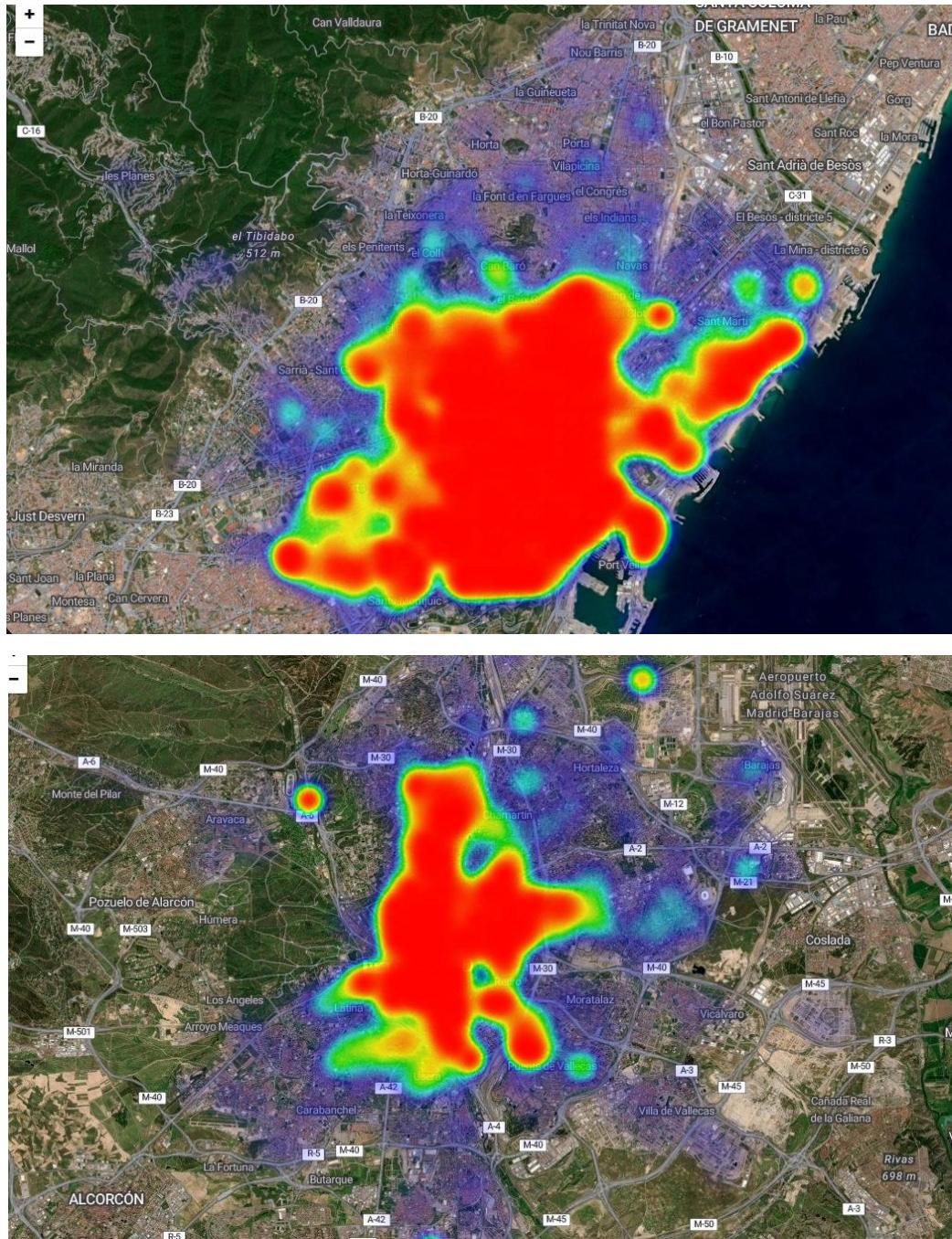
A continuación podemos observar unas capturas de cómo quedan los mapas de calor en Barcelona y en Madrid respectivamente:



En ambas ciudades se puede ver como las zonas rojas, es decir, las zonas en las que se encuentran la mayoría de los alojamientos más caros se extienden más por lo que viene siendo el centro de las ciudades. Y, a medida que nos vamos acercando a la periferia, el color se va convirtiendo en azul y se va difuminando, indicando no solo que los precios son más bajos, sino que también hay menos alojamientos disponibles.

Hay que tener en cuenta que estos resultados se han obtenido con una intensidad basada en los 200€ por noche, por lo que es normal que una gran parte de las ciudades esté en tonos azules indicando unos precios económicos.

Por ejemplo, si bajamos la intensidad hasta los 50€, el cual sería un precio dentro de un rango económico según nuestros estudios realizados en la primera parte, podemos ver como la oferta de los alojamientos asequibles disminuye considerablemente.



3.2 Mapa de categorías de los tipos de alojamientos

El segundo análisis que realizamos fue en base a los tipos de alojamientos. En este estudio hemos querido ver en el mapa tanto la localización como el tipo de los alojamientos. Para llevarlo a cabo hicimos uso de un mapa de categorías, donde cada tipo de alojamiento fue representado por un color.

Generación de los GeoJSON

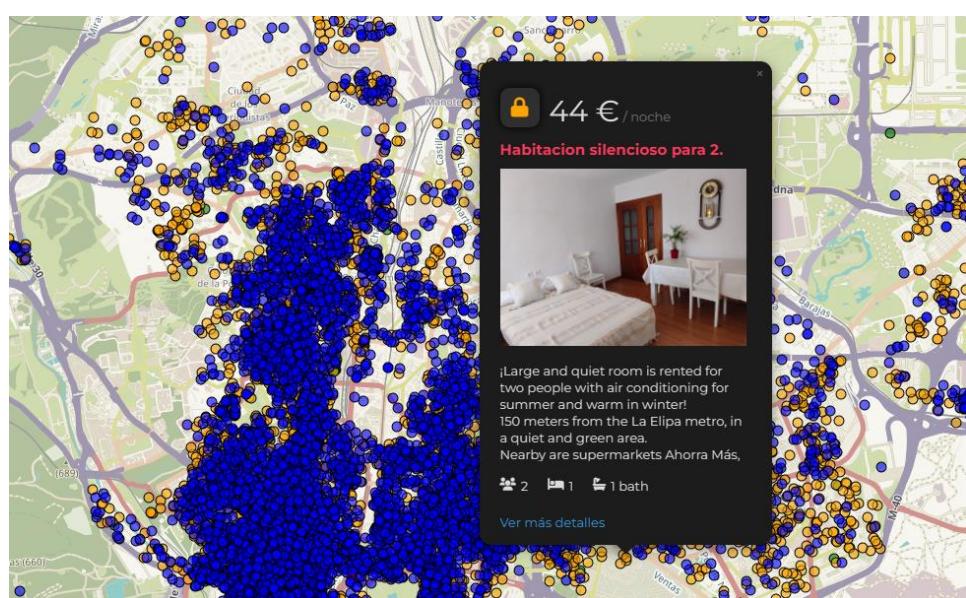
En este estudio utilizamos tanto las variables *latitude* y *longitude* para la localización de los alojamientos como la variable *room_type* para identificar los tipos. Estos tipos pueden ser: habitación compartida, habitación privada, habitación de hotel y casa o apartamento entero.

Al igual que en el anterior, el **dump.py** carga todos los datos tabulares, limpia los precios y crea los GeoJSON con colecciones de tipo *Point* junto a sus coordenadas. Asimismo, como quisimos mostrar también los datos de los alojamientos, añadimos a las propiedades de cada entidad algunos de los datos más importantes:

- *name*: el nombre del alojamiento.
- *description*: una descripción sobre el alojamiento.
- *picture_url*: una foto del alojamiento.
- *accommodates*: capacidad máxima de inquilinos.
- *bathrooms_text*: cantidad de baños.
- *bedrooms*: cantidad de habitaciones.
- *price*: el precio por noche.
- *listing_url*: un enlace hacia la página del alojamiento en Airbnb para más información.

Funcionalidades

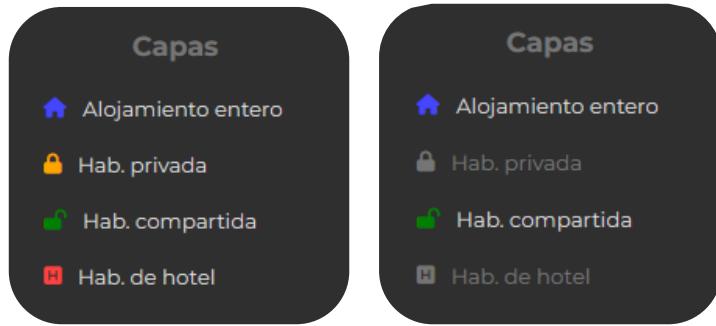
Este mapa cuenta también con dos funcionalidades extra de interactividad. La primera es que, al hacer clic sobre cualquier alojamiento del mapa, se nos abre la información más importante de él. Para hacer esto, lo único que hicimos fue vincular un *Popup* a cada punto del mapa, al cual le dimos un *html* estructurado para luego poder editar sus estilos. Con esto conseguimos un resultado como el siguiente.



La otra funcionalidad sirve para poder ocultar los alojamientos en función del tipo. Para hacer esto, usamos el mismo método que en mapa de calor, es decir, llamar a la función `toggleLayerView()`.

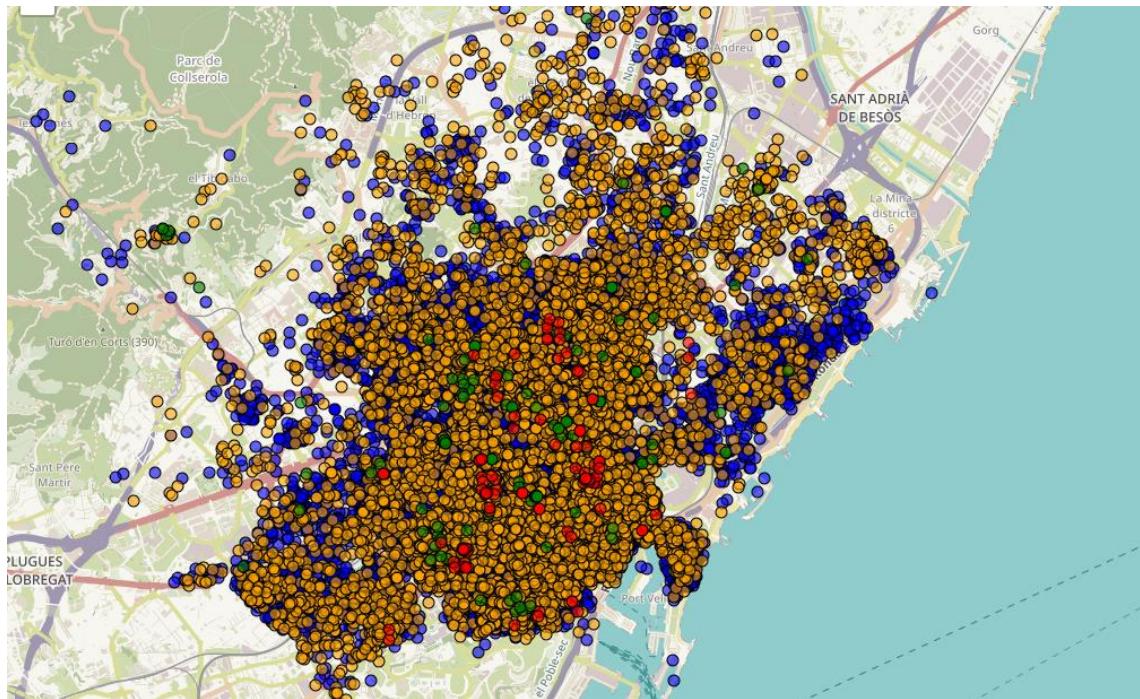
La única diferencia fue que tuvimos que crear una capa diferente para cada tipo de alojamiento, de esta forma solo necesitábamos crear una función intermedia para seleccionar la capa correcta que pasarse a la función mencionada. Esta función intermedia se llama `changeLayerView()` y usa un `switch` para escoger la capa específica.

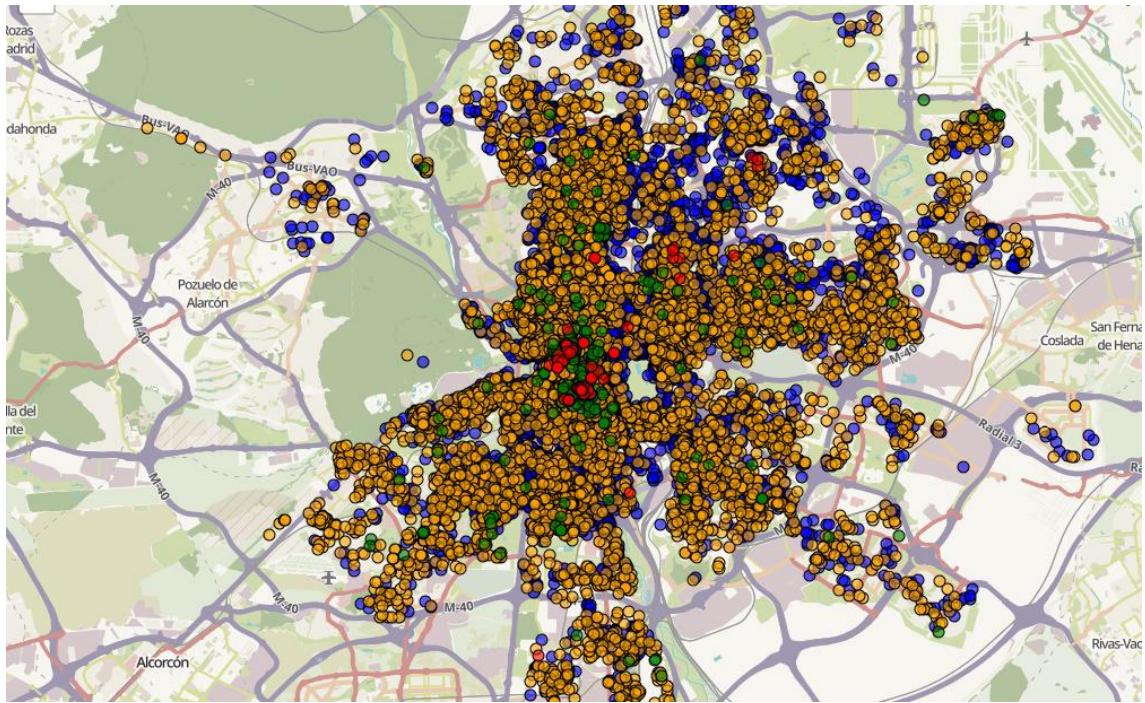
Para interactuar con esta funcionalidad, hay un botón para cada tipo de alojamiento en el panel derecho llamado Capas:



Resultados

A continuación, podemos observar un par de capturas del mapa de Barcelona y de Madrid:





Lo primero que hemos podido ver en los mapas es que de lo que más hay son casas o apartamentos enteros y habitaciones privadas. Por otro lado, hemos visto que hay pocas habitaciones compartidas y habitaciones de hotel. Esto se confirma con el estudio que hicimos relacionando el precio con el máximo de huéspedes, donde añadimos la variable de tipo de alojamiento y generamos gráficos de queso para ver las distribuciones.

También se puede apreciar que donde más cantidad de alojamientos se acumulan es en las partes centrales de las ciudades.

3.3 Mapa de coroletas de los barrios

El siguiente estudio que hicimos fue un mapa coroplético de los barrios para representar datos cuantitativos y estadísticos mediante una técnica cartográfica de escala de colores.

Los diferentes datos que analizamos fueron los siguientes:

-  El precio medio por noche.
-  La cantidad de alojamientos.
-  La media de reseñas.
-  La media de valoración general.
-  La media de valoración del barrio.

Al ser un mapa coroplético, el color de cada zona indica una estadística, por lo que estos colores tienen que seguir una escala que permita identificar a simple vista las zonas con valores más altos o bajos. Por esta razón, decidimos usar una escala de 6 colores que fuera desde el verde (valores bajos) hasta el rojo oscuro (valores altos), pasando por amarillos y naranjas.



Para determinar qué valor pertenecía a cada color, calculamos 5 límites que harían de barrera entre los diferentes colores. Para ser más precisos, obtuvimos estos límites mediante el conjunto entero (Madrid y Barcelona) de los datos de la variable a estudiar. Entonces, calculamos tanto la media como la desviación estándar:

```
// Definir los límites de los rangos
rango1 = media - 2 * desviacion;
rango2 = media - desviacion;
rango3 = media;
rango4 = media + desviacion;
rango5 = media + 2 * desviacion;
```

Todo esto lo hace el **script.js** asociado al mapa con la función **obtenerRangos()**, la cual recibe como parámetro el conjunto de datos. Luego, la función **getColor()** es la encargada de asociar el color a cada uno de los valores que se le pasan por parámetro.

Generación de los GeoJSON

Esta generación de los GeoJSON fue algo más compleja que las anteriores, ya que esta vez tuvimos que añadir los barrios de las ciudades, por lo que tuvimos que basarnos en los **.geojson** que nos proporcionaba la web de Airbnb. Asimismo, esta vez no necesitábamos pasar todo el conjunto de datos, ya que solo queríamos estadísticas como la media o el conteo. Por esta razón, creamos la función **calcular_estadisticas_por_distrito()**, la cual es la encargada de hacer estos cálculos en base al barrio y devolverlos en un nuevo **DataFrame**.

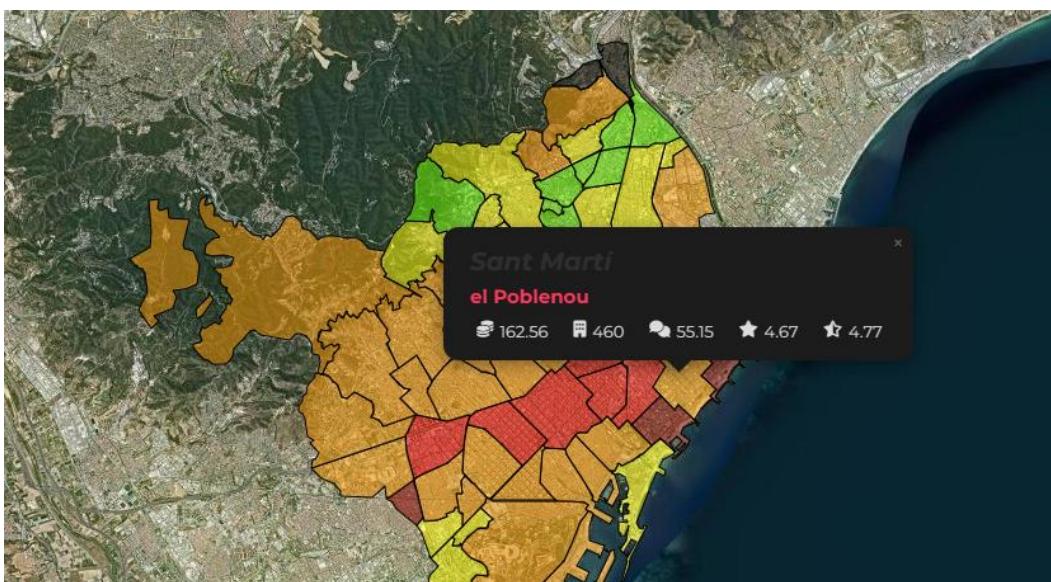
Entonces, para crear los GeoJSON recorrimos las **features** de los **.geojson** con los barrios (**neighbourhood_ciudad.geojson**) y añadimos los valores calculados a las propiedades pertinentes, basándonos en el campo **neighbourhood** que contenía los nombres de los barrios para identificar cada **feature**.

Para hacer todo esto usamos las siguientes variables:

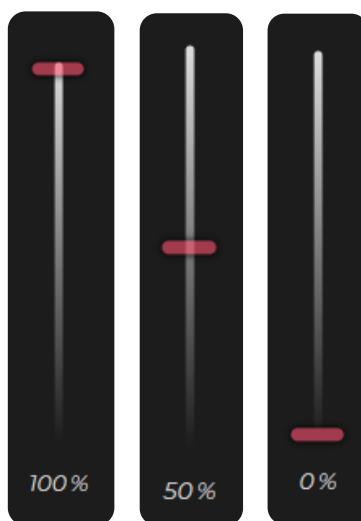
- `price`: para calcular el precio medio.
- `id`: para contar la cantidad de alojamientos.
- `number_of_reviews`: para calcular la cantidad media de reseñas.
- `review_scores_rating`: para calcular la valoración general media.
- `review_scores_location`: para calcular la valoración del barrio media.

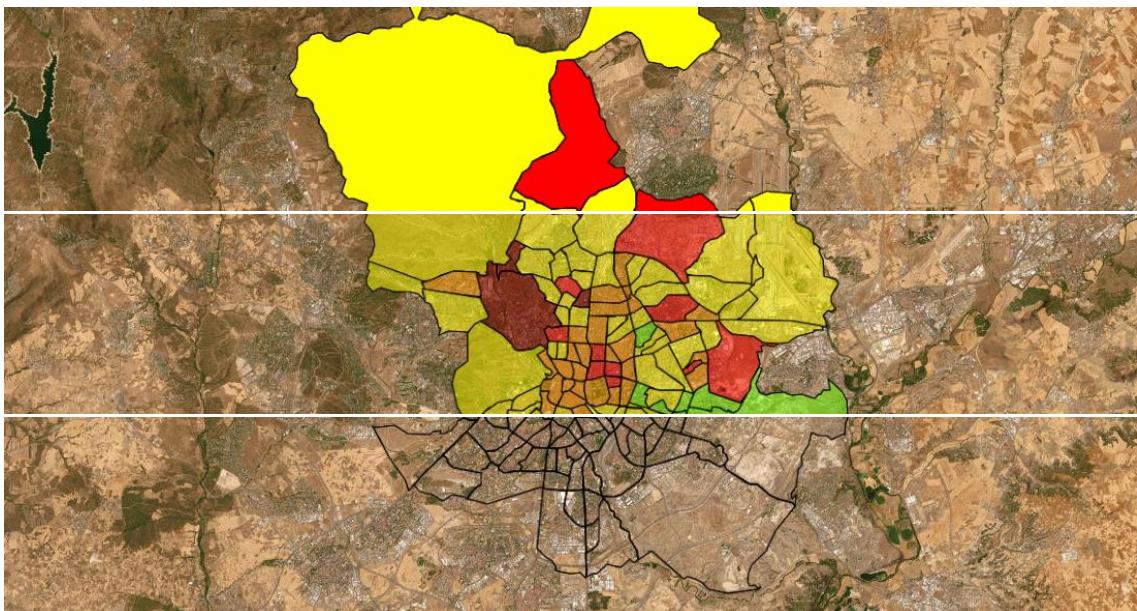
Funcionalidades

Este mapa cuenta con tres funcionalidades extra. La primera es simplemente mostrar información de cada barrio al hacer clic encima, lo cual hicimos asociando un *Popup* a cada barrio a la hora de crear la capa. Al igual que en el anterior estudio, añadimos un *html* estructurado para editar sus estilos y obtener el siguiente resultado.



La segunda funcionalidad que añadimos fue un *Slider* para poder cambiar la opacidad de la capa, para así poder ver tanto las particiones de los barrios como el TMS de debajo. Para conseguir esto lo que hicimos fue añadir un *listener* al *Slider* para que creara de nuevo el mapa cada vez que se cambiara su valor, pero esta vez modificando su opacidad.





Por último, la última funcionalidad que agregamos fue la de poder seleccionar la variable a mostrar en el mapa. Como hemos dicho en la introducción de este estudio, quisimos poder representar diferentes variables cuantitativas, por lo que decidimos que cada una de estas tuviera su propia capa en el mapa y que fuera el usuario el que pudiera ir intercambiando la vista de ellas.

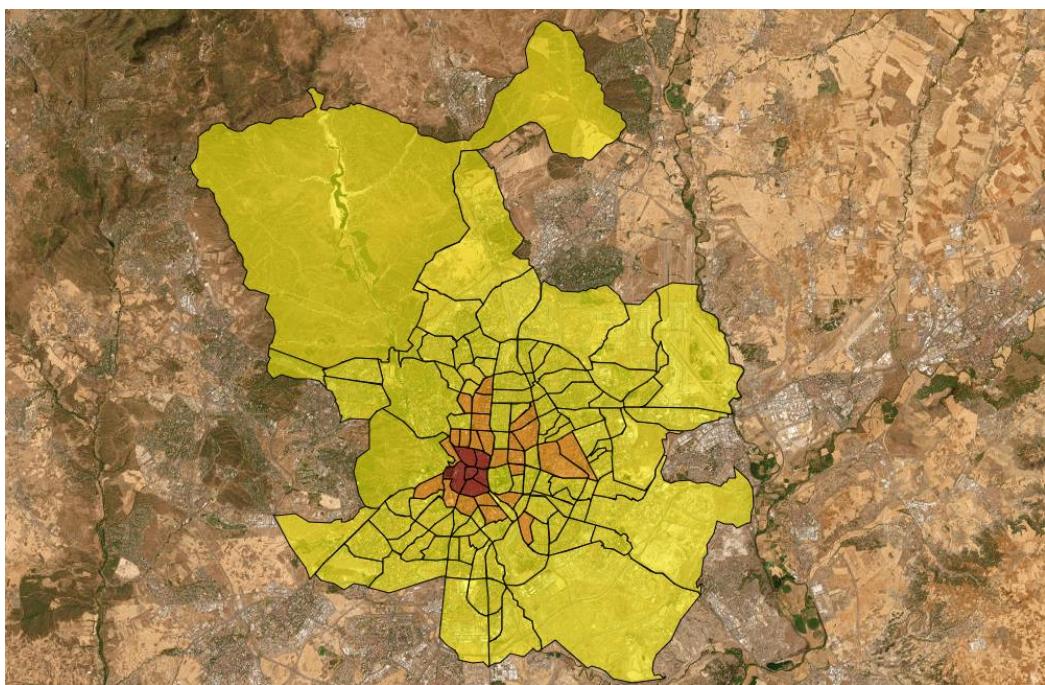
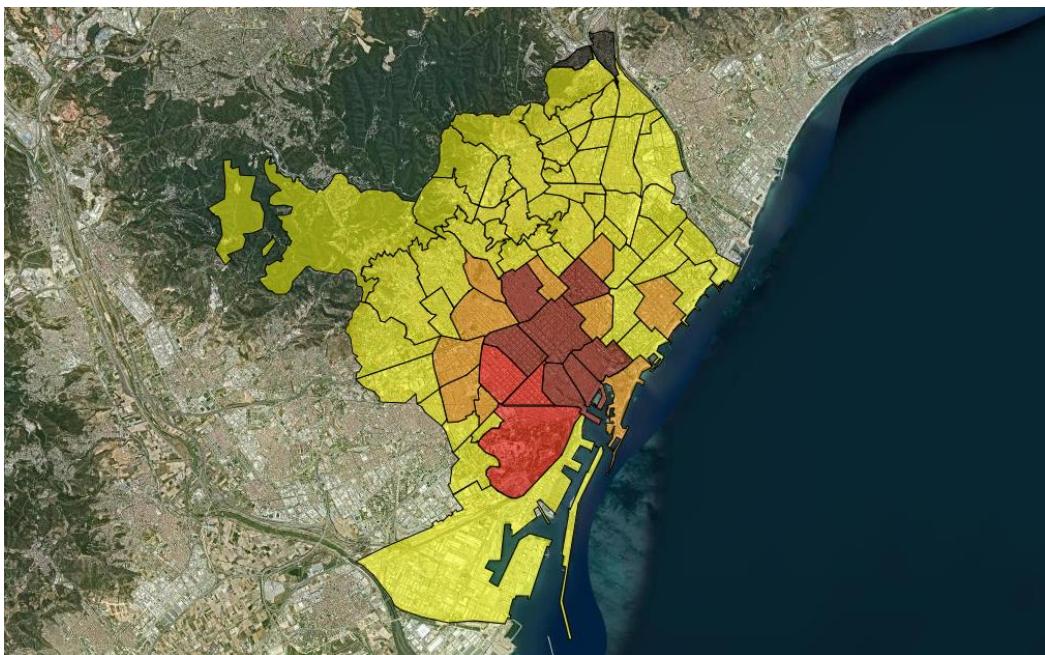
Para hacer esto, creamos la función ***changeFuncionality()***, la cual recibe el nombre de la variable por parámetro y crea de nuevo la capa de coropletas usando solamente los datos de esa variable. Para que el usuario pudiera hacer estos cambios, añadimos el panel llamado *En función de*, el cual tiene la lista de las variables que se pueden representar estadísticamente con un mapa de coropletas.



Con esta funcionalidad hay que tener en cuenta que los colores rojos siempre indican los valores más altos, independientemente de si esto es mejor o peor. Por ejemplo, en el precio ya está bien porque que la vista asocia inconscientemente el color rojo con precios altos, pero en las valoraciones en cambio, el color rojo parecería que ese barrio tiene malas valoraciones cuando en realidad lo que quiere decir es que la valoración media es alta.

Resultados

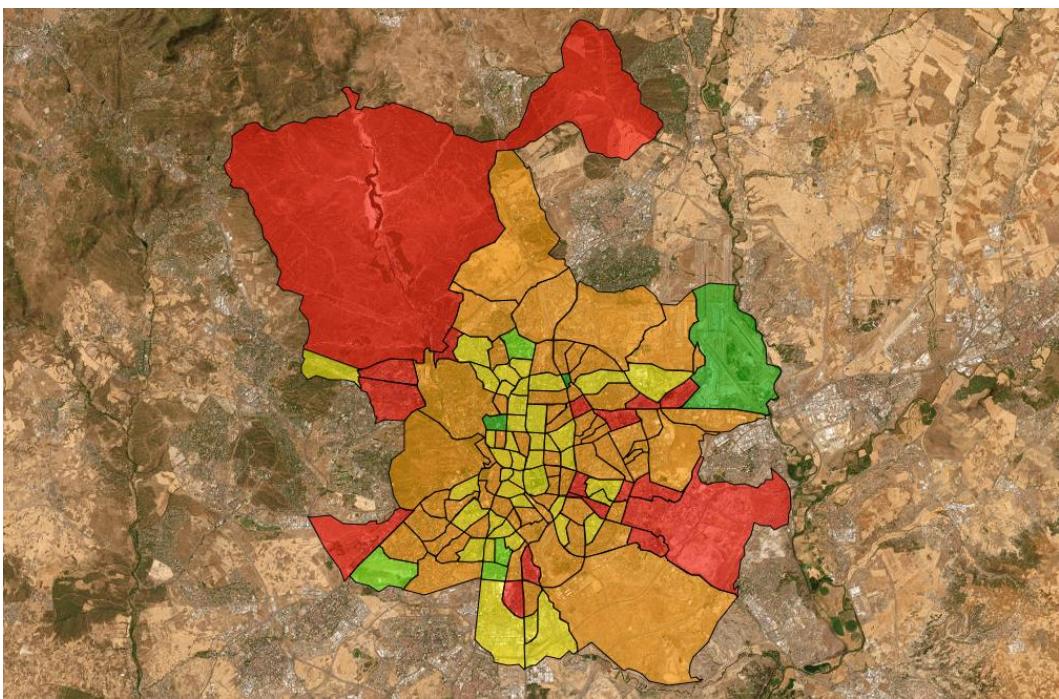
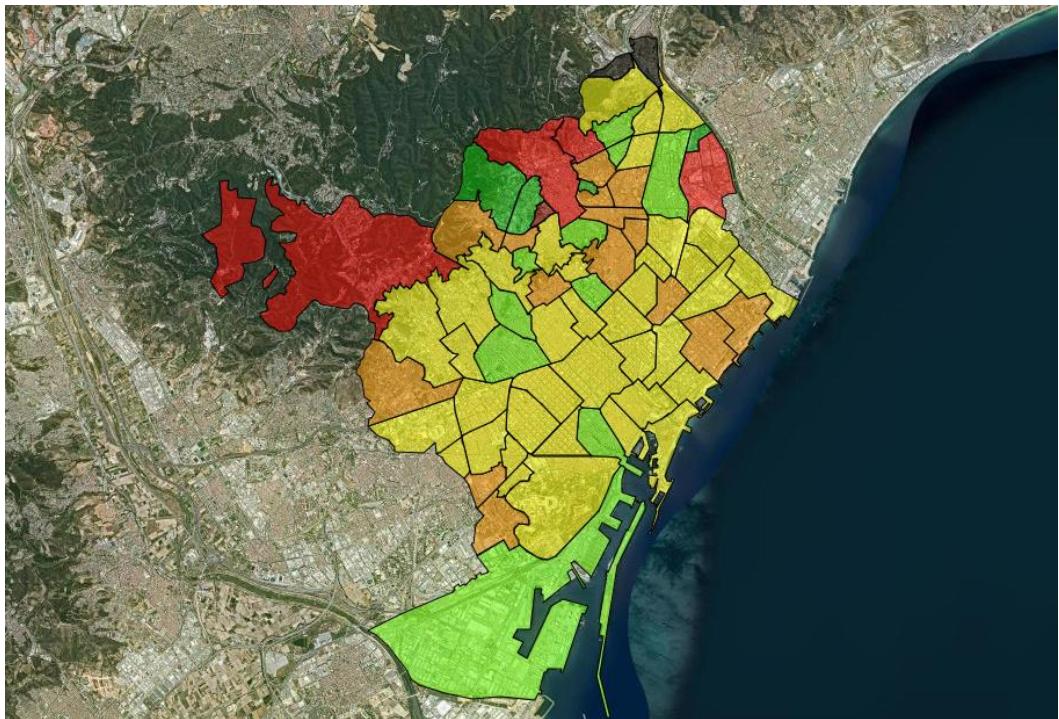
Para comparar algunas variables en ambas ciudades, tendríamos por ejemplo la cantidad de alojamientos en Barcelona y Madrid respectivamente:



Lo primero que llama la atención en estos resultados es la poca presencia de colores verdes, y esto tiene una explicación simple, la cual es que los que aparecen en rojo oscuro tienen cantidades de alojamientos extremas, perjudicando los límites que obtenemos para calcular los rangos.

En ambas ciudades, los barrios rojos oscuro se mueven entre los 1000 y 3000 alojamientos, mientras que los amarillos difícilmente llegan a los 200.

Otra comparación interesante sería la de la valoración general media:



En esta comparación se puede ver como los distritos con una media de valoración general son los que se encuentran más a las afueras de las ciudades, ya que no hay que olvidar que el color rojo representa los valores más altos. Aún así, las diferencias son mínimas, ya que los rojos rondan valoraciones de 4.9 mientras que los verdes se mueven cerca de los 4.5.

3.4 Mapa de burbujas de la cantidad de alojamientos

Otro análisis que realizamos fue sobre la cantidad de alojamientos que hay por distrito. Para realizar este estudio, utilizamos un mapa de burbujas, el cual representa datos cuantitativos y estadísticos basándose en el tamaño de las burbujas. Es decir, cuanto más grande sea la burbuja, más alojamientos tiene el distrito.

Generación de los GeoJSON

En este caso, únicamente utilizamos la variable `neighbourhood_group_cleansed` tanto para saber el nombre de los distritos como para contabilizar la cantidad de alojamientos que hay en cada uno de ellos.

Para ubicar las burbujas en los centros de cada distrito, decidimos poner las coordenadas (latitud y longitud) a mano en el script `dump.py`, ya que de esta manera conseguíamos el resultado que más nos convenía. Para conseguir las coordenadas, utilizamos la página de *OpenStreetMap*:

<https://www.openstreetmap.org/>.

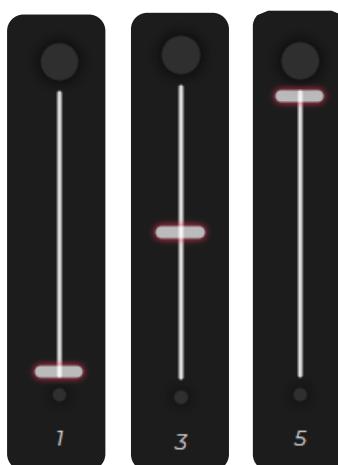
Para crear los GeoJSON, los hicimos de cero en el mismo script usando colecciones de *Features* donde cada una es de tipo *Point* con las coordenadas de cada uno de los distritos, juntamente con su nombre y la cantidad de alojamientos en las propiedades.

Funcionalidades

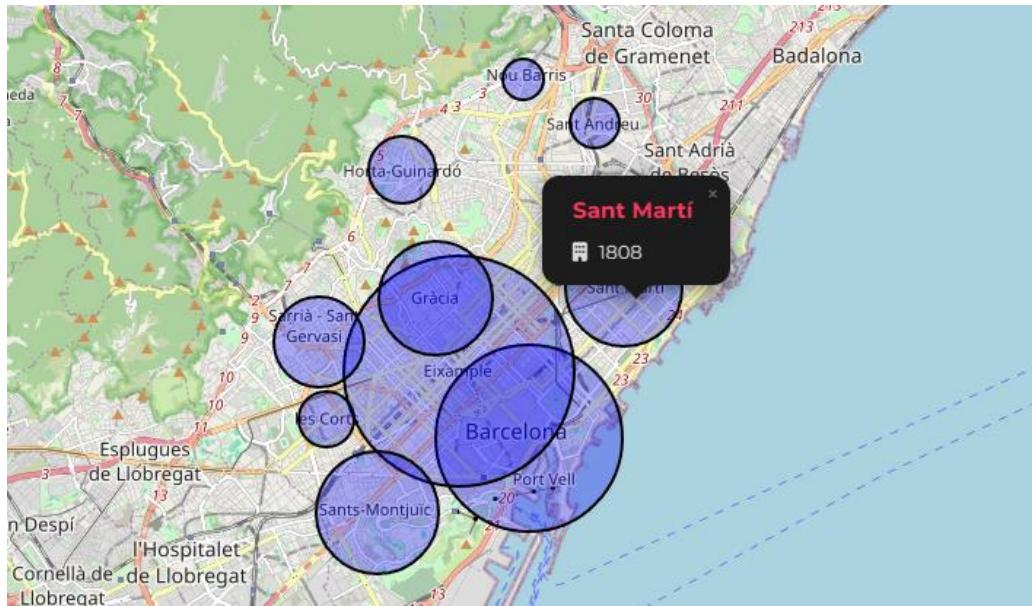
A este mapa le añadimos dos funcionalidades muy parecidas a las del [mapa de calor](#). La primera fue para habilitar o deshabilitar la capa de burbujas. Su funcionamiento es el mismo, llamar a la función `toggleLayerView()` con la capa de burbujas como parámetro. Para hacerlo, también se encuentra el botón en el panel de *Capas* de la derecha del mapa.



La segunda funcionalidad es un *Slider* también, pero esta vez para poder modificar la escala de las burbujas. Eso sí, el funcionamiento es el mismo, el cual es crear de nuevo la capa de burbujas con el valor del *Slider*. Por defecto, la escala de las burbujas es de 3, pero con esto se puede ir desde el 1 hasta el 5.

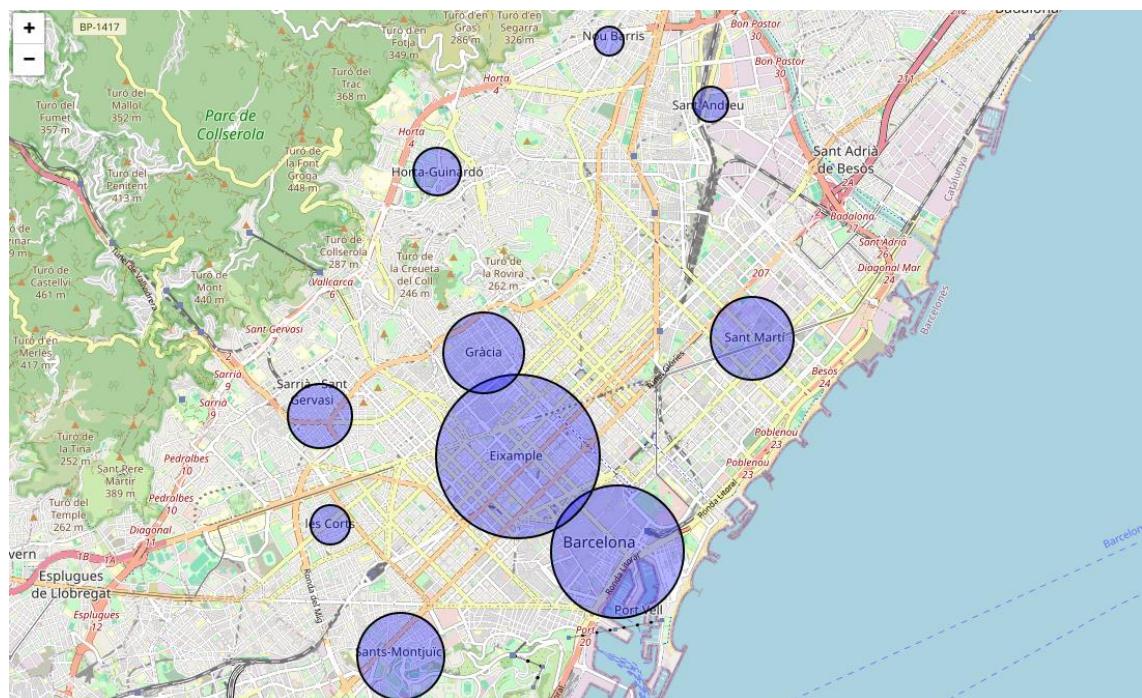


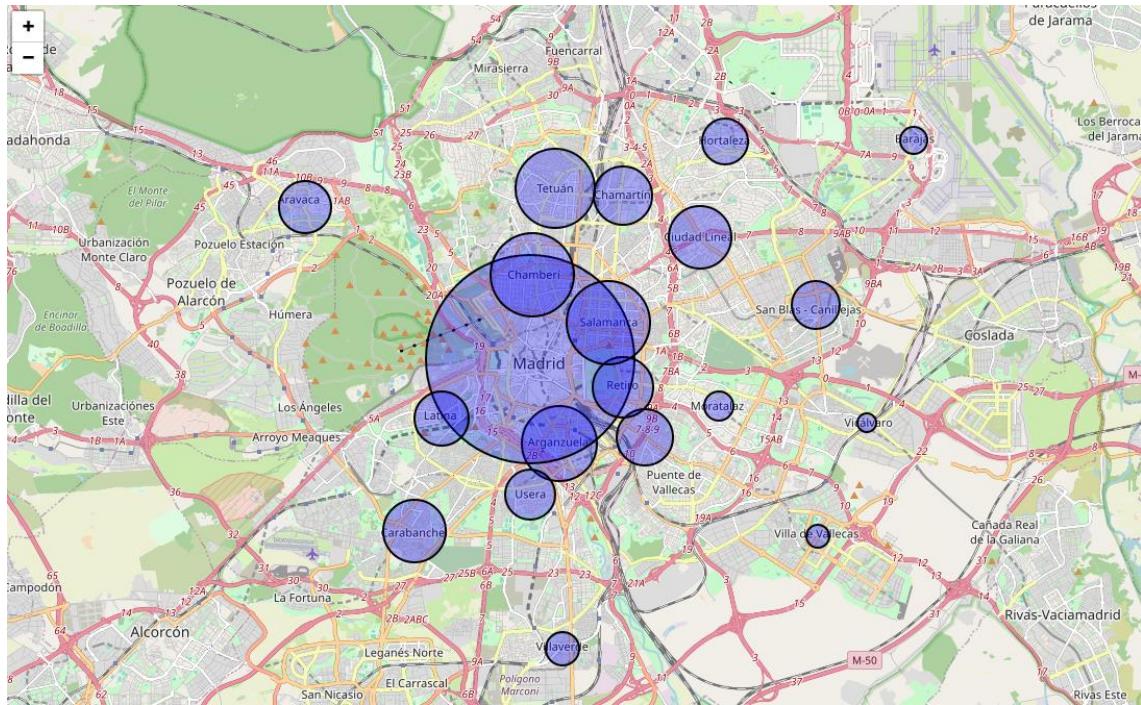
Una última funcionalidad sería la cantidad de alojamientos que se muestra en cada distrito al hacer clic en su burbuja. Como en todos los estudios anteriores, esto lo conseguimos asociando *Popups* a cada burbuja.



Resultados

A continuación, podemos observar un par de capturas de los mapas de Barcelona y Madrid respectivamente:





En Barcelona, podemos ver que los distritos con burbujas más grandes son los del Eixample y Ciutat Vella. Después de estos, podríamos decir que las siguientes más grandes son las de los distritos de Gràcia, Sant Martí y Sants-Montjuïc.

Por otro lado, en Madrid, podemos observar cómo la burbuja del distrito Centro es la más grande, mucho más que las demás. Asimismo, podemos ver cómo otras burbujas con tamaños considerables, como las de Chamberí, Salamanca y Arganzuela, rodean el centro. Con esto podemos afirmar que, en general, hay una gran cantidad de alojamientos centralizados (no solo por el distrito Centro).

Estas diferencias en la cantidad de alojamientos de los distritos se pueden corroborar en el [primer estudio de la fase 1](#), pero ahora podemos verlo muchísimo mejor representado en un mapa real.

3.5 Mapa de categorías de los hosts

El último análisis geográfico que quisimos realizar fue sobre los 5 propietarios con más alojamientos de cada ciudad. Más concretamente, quisimos ver representados en el mapa todos los alojamientos de los que disponen. De esta manera, podríamos analizar si los tiene todos juntos o en zonas en concreto. Para representar esto, utilizamos otra vez el mapa de categorías, donde cada propietario tendrá un color asignado independientemente de la ciudad.

Generación de los GeoJSON

Para este estudio, lo primero que hace el script **dump.py** es filtrar los .csv para obtener solo los alojamientos de los 5 propietarios que más alojamientos tienen en cada ciudad. Entonces, al igual que en el estudio de categorías anterior, crea un nuevo GeoJSON para cada ciudad con una *Feature* de tipo *Point* para cada alojamiento.

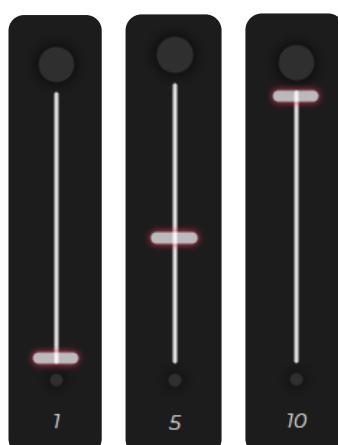
Además, a cada *Feature* añade los siguientes datos en el campo de propiedades:

- `host_id`: el identificador del propietario.
- `host_name`: el nombre del propietario.
- `host_since`: la fecha desde la cual son propietarios.
- `host_url`: un enlace hacia el perfil del propietario.
- `host_response_time`: el tiempo de respuesta del propietario.
- `host_response_rate`: la tasa de respuesta del propietario.
- `host_acceptance_rate`: la tasa de aceptación del propietario.
- `host_picture_url`: una foto del propietario.
- `host_identity_verified`: si el propietario está verificado o no.
- `listing_url`: un enlace hacia la página del alojamiento en Airbnb.

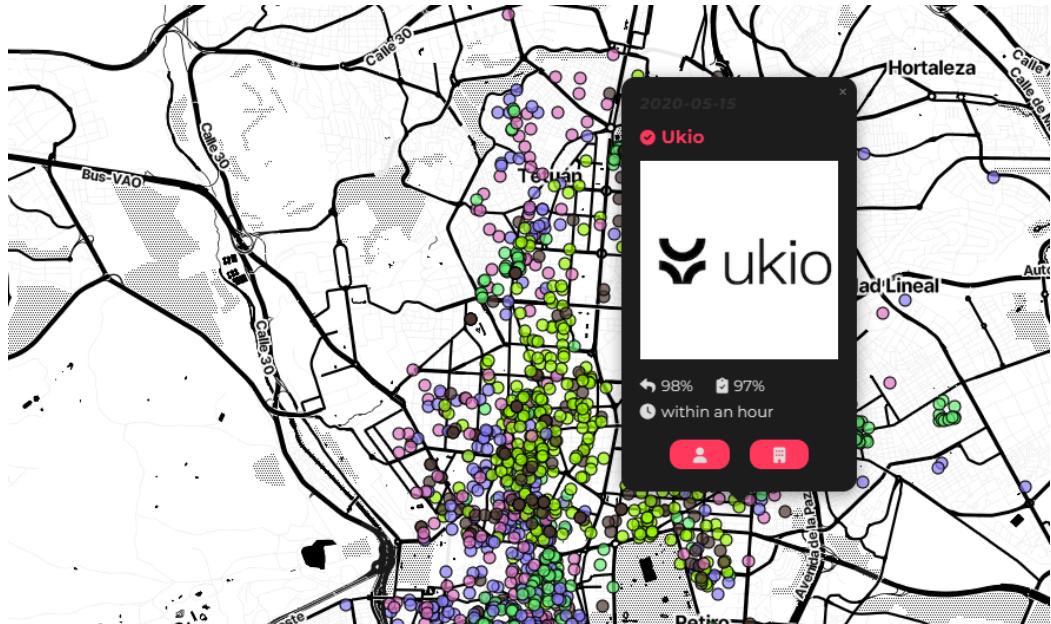
Funcionalidades

Este último estudio cuenta con cuatro funcionalidades muy parecidas a otros estudios anteriores. La primera es más una curiosidad que una funcionalidad, y es la forma en la que se asignan los colores a los propietarios, la cual es totalmente aleatoria, por eso los colores cambian cada vez que se recarga la página. Para esto hicimos dos funciones diferentes: **getRandomColor()** para generar un color hexadecimal aleatorio y **getColor()** para asignar un color nuevo a cada propietario en función de su identificador.

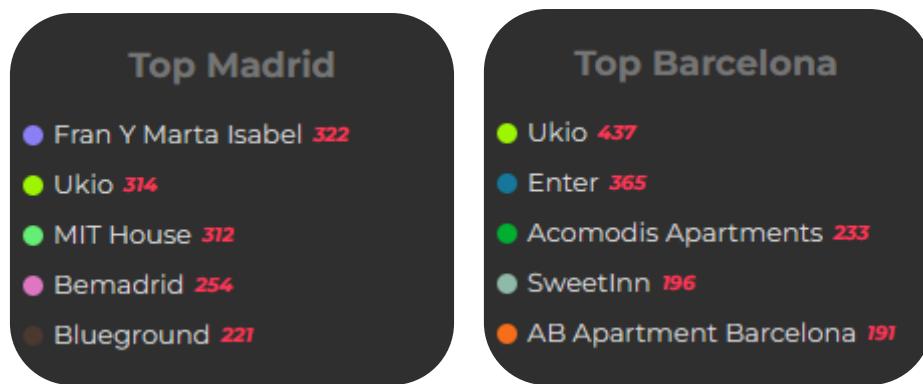
La segunda funcionalidad es un *Slider* como el del estudio anterior, pero esta vez para modificar el radio de los círculos.



La tercera funcionalidad es la de representar información de los propietarios al hacer clic encima de los alojamientos. El problema es que no era muy relevante ya que todas eran iguales para un mismo propietario. Por eso, decidimos añadir enlaces hacia la web de Airbnb tanto para el propietario como para el alojamiento en cuestión.



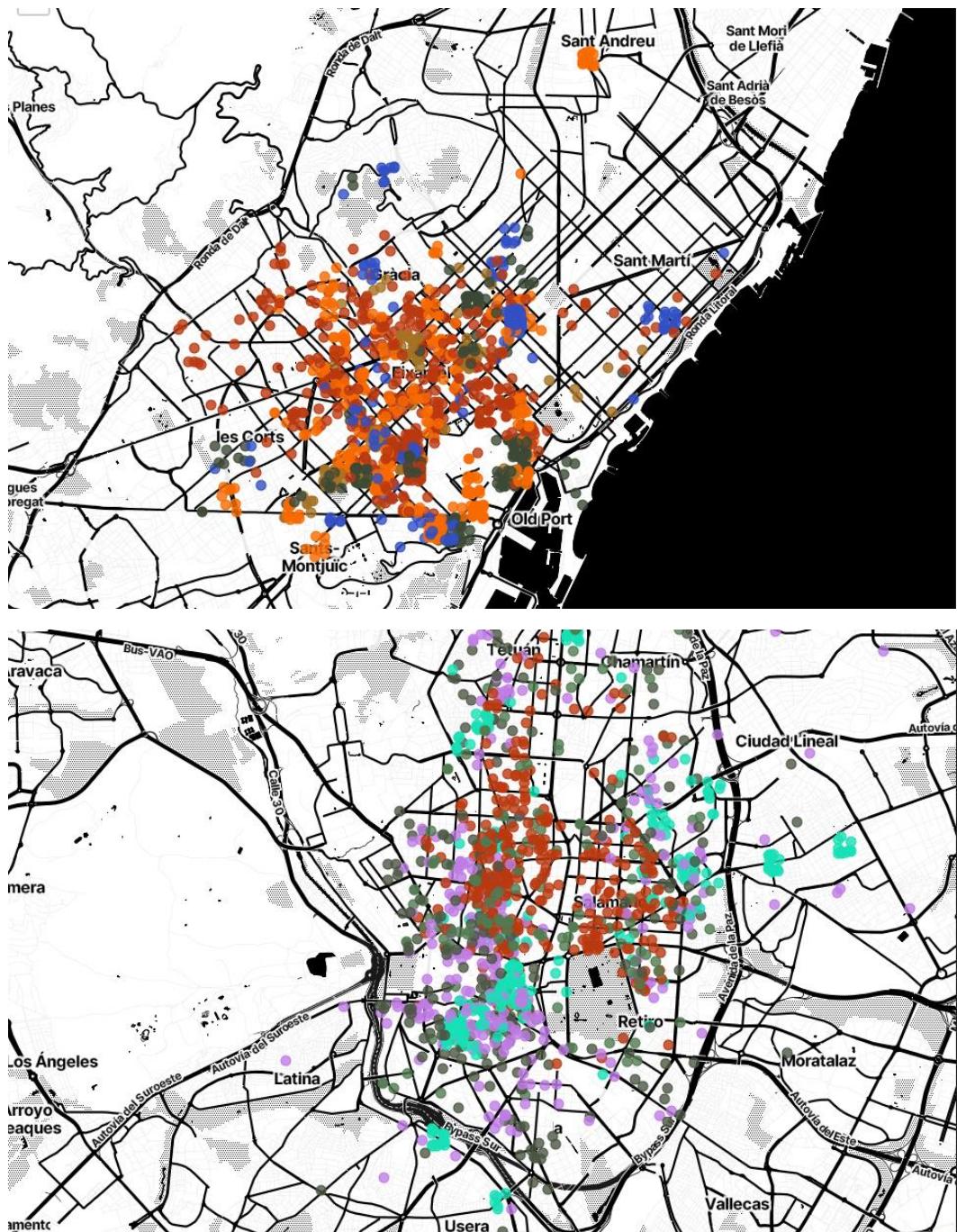
La última funcionalidad que añadimos fue un podio para cada ciudad, donde aparecen los 5 propietarios junto a la cantidad de sus alojamientos y ordenados de mayor a menor. Para hacer esto de forma dinámica y sin *hardcodear*, decidimos que se añadiera el contenido desde el *javascript* en el momento en el que se crea la capa de categorías. Con esto conseguimos que siempre se muestre el top 5 sin importar los datos que carguemos. Estos tops se pueden ver en la parte derecha del mapa, encima de los TMS.



Viendo esto, los resultados concuerdan con los del estudio de la primera fase donde analizamos [los propietarios con más alojamientos](#) de ambas ciudades.

Resultados

A continuación, podemos observar un par de capturas tanto de Barcelona como de Madrid:



En estos resultados no se observan tendencias ni zonas específicas ocupadas por propietarios en concreto, sino que más bien vemos como todos ellos tienen alojamientos repartidos por todas las zonas. Aun así, al estar hablando de propietarios con más de 200 alojamientos aproximadamente, es normal que estén repartidos y no solamente en zonas específicas.