Laura Moreno
Michellynne Borromeau
Silvia Alonso
Elena Prada
Verónica Encinas
Soledad Grobas

1. Introducción.

Se nos pide hacer un análisis del dataset de AirBnb.

Nuestro objetivo es averiguar cómo influyen las diferentes características en el valor del inmueble para poder hacer un algoritmo que prediga su precio en función de ciertas características.

1.1. Presentación del Equipo y Elección del Nombre.

Nuestro equipo se llama **Yellow Data Team**. Hemos elegido identificar a nuestro equipo con un color como el amarillo porque se asocia comúnmente con la inteligencia, la felicidad, la energía, el optimismo y la creatividad. Estos atributos sugieren un equipo vibrante y dinámico, que aborda los problemas con una actitud positiva y energética. La asociación con la creatividad implica que el equipo está dispuesto a pensar fuera de lo convencional y encontrar soluciones innovadoras.

Además, el amarillo también puede relacionarse con la claridad y la luz, lo cual es particularmente relevante en un contexto de análisis de datos. Indica que el equipo se enfoca en arrojar luz sobre la información, clarificar datos complejos y proporcionar insights valiosos y transparentes.

A continuación vamos a presentarnos y a contar las tareas que hemos llevado a cabo cada una. Aunque nos gustaría aclarar que cada miembro del equipo ha estado colaborando de manera activa en prácticamente todas las etapas del proyecto.

- **Laura Moreno**: Málaga, especialista en Marketing Digital. Ha colaborado con la organización de tareas, run timming, análisis y visualización de datos y programación.
- **Michellynne Borromeu**:Barakaldo, Diseñadora. Ha colaborado en la visualización de datos en Tableau y excel.
- **Silvia Alonso**: Madrid. Ha colaborado principalmente realizando la maquetación y programación de la web.
- **Elena Prada**: Madrid, especialista en Marketing Digital. Ha colaborado en la visualización de datos en Tableau.
- **Verónica Encina**s: Madrid. Administración de BBDD. He colaborado en el modelado de datos con el diagrama E/R, y en el documento final.
- **Soledad Grobas**: Vigo. He colaborado con la comunicación y organización del equipo, análisis de datos y presentación de documentación final.

Nos gustaría también poner de manifiesto que en nuestro proyecto, hemos empleado **Metodologías Ágiles** para garantizar una gestión eficiente y flexible del trabajo, permitiendo una adaptación rápida a los cambios y mejoras continuas en el proceso. Estas metodologías se centraron en la colaboración, la entrega continua de valor y la retroalimentación constante. Nos hemos organizado con técnicas como estas:

- Hemos tenido Iteraciones Cortas (Sprints): Dividimos el proyecto en ciclos de trabajo cortos por cada apartado. Cada sprint tenía objetivos claros y específicos, lo que nos permitió enfocarnos en pequeñas porciones del trabajo.

- Run Timming (Timeboxing): Desde el principio asignamos un tiempo fijo y máximo a cada apartado del proyecto. Una vez que el tiempo asignado ha transcurrido, la actividad debía encontrarse finalizada.

- Reuniones programadas: Celebramos reuniones programadas breves donde cada miembro del equipo compartía sus progresos, desafíos y planes hasta el próximo encuentro. Estas reuniones fomentaron la comunicación abierta y ayudaron a identificar y resolver problemas rápidamente.

- Trabajo Colaborativo y Multifuncional: Nuestro equipo estaba compuesto por miembros con habilidades complementarias, lo que nos permitió abordar diversas tareas de manera eficiente. La colaboración y la asignación de tareas basada en habilidades específicas aseguraron un flujo de trabajo equilibrado y productivo.

- Documentación Ligera y Comunicación Efectiva: En lugar de una documentación extensa, nos centramos en mantener una comunicación clara y directa a través de herramientas colaborativas (como Google Drive, Google

Docs, Google Collab, Google Sheets, GitHub, etc) y reuniones regulares. Esto permitió una toma de decisiones más rápida y un seguimiento efectivo del progreso.

Teniendo en cuenta el tiempo tan corto para la realización del proyecto, estas metodologías nos han ayudado mucho en la consecución de lo que se conoce como el "producto mínimo viable", que en nuestro contexto, representa el análisis de datos o modelo de datos que incluye solo las características esenciales necesarias para cumplir con los objetivos del proyecto que se nos ha propuesto.

Para cerrar este apartado, queremos compartiros la web que hemos realizado para explicar y dar más detalles sobre nuestro equipo y el proyecto final. Podéis visitarla haciendo clic en este enlace.

1.2. Retos y Dificultades Abordadas.

Durante el **Análisis Exploratorio** nos encontramos con un archivo que necesitaba mucho tratamiento, más allá de lo que puede parecer evidente (como localizar y tratar valores faltantes o nulos), el DataSet original contenía muchísima información que no era relevante para nuestro objetivo, como información de otros países y ciudades o códigos internos de AirBnB. Una vez pudimos comprobar cómo influían las diferentes características de los alojamientos con el precio de estos, se nos rompieron un poco los esquemas, porque a priori, pensábamos que lo que más influencia iba a tener era el barrio. Como veremos posteriormente, no ha resultado del todo así.

En **tableau** nos hemos encontrado con que el programa no hacía lectura de mapas de polígonos sin el código postal. Para solucionar hemos utilizado primero chat gpt con el comando de generar una columna con los códigos postales a partir de la columna de los nombres de barrios (location) de nuestro dataset. Ese proceso fue efectivo, pero al repasar los datos había errores comparando con los datos de Correos (Buscador de códigos postales de Correos | Correos.es). Lo hemos revisado y reemplazado los códigos, teniendo como referencia correos, en excel.

Para agrupar los barrios por distrito hemos tomado como referencia la página de la Comunidad de Madrid(Mapas de Distritos y Barrios - Ayuntamiento de Madrid).

Durante la **elaboración del modelo de regresión lineal** nos enfrentamos a que tras el tratamiento y limpieza de datos, nuestro DataFrame tenía un número de muestras no muy elevado. Este hecho junto con que puede que haya otros modelos más efectivos para la tarea de predecir los precios que el de regresión lineal, así como el de carecer de tiempo para aplicar técnicas de oversampling para contar con más muestras mejor equilibradas, nos ha llevado a entregar un modelo con resultados muy modestos en la predicción del precio de los alojamientos turísticos, en el cual se puede observar que existe underfitting. Aunque como veremos más adelante, hemos conseguido mejorarlo

ligeramente aplicando técnicas para combatirlo (más concretamente la Regularización de Ridge) .

2. Definiendo el Dataset.

Analizamos el dataset y nos familiarizamos con los datos que contiene. (punto 1 del trabajo).

El data set escogido es scrapeado de la web de AirBnB (https://insideairbnb.com/get-the-data/) y fue descargado de la web OpenDataSoft (https://public.opendatasoft.com/explore/dataset/air-bnb-listings/information/?disjunctive.neighbourhood&disjunctive.column_10&disjunctive.city&q=Madrid&location=7,40.5931,-4.422&basemap=jawg.light) .

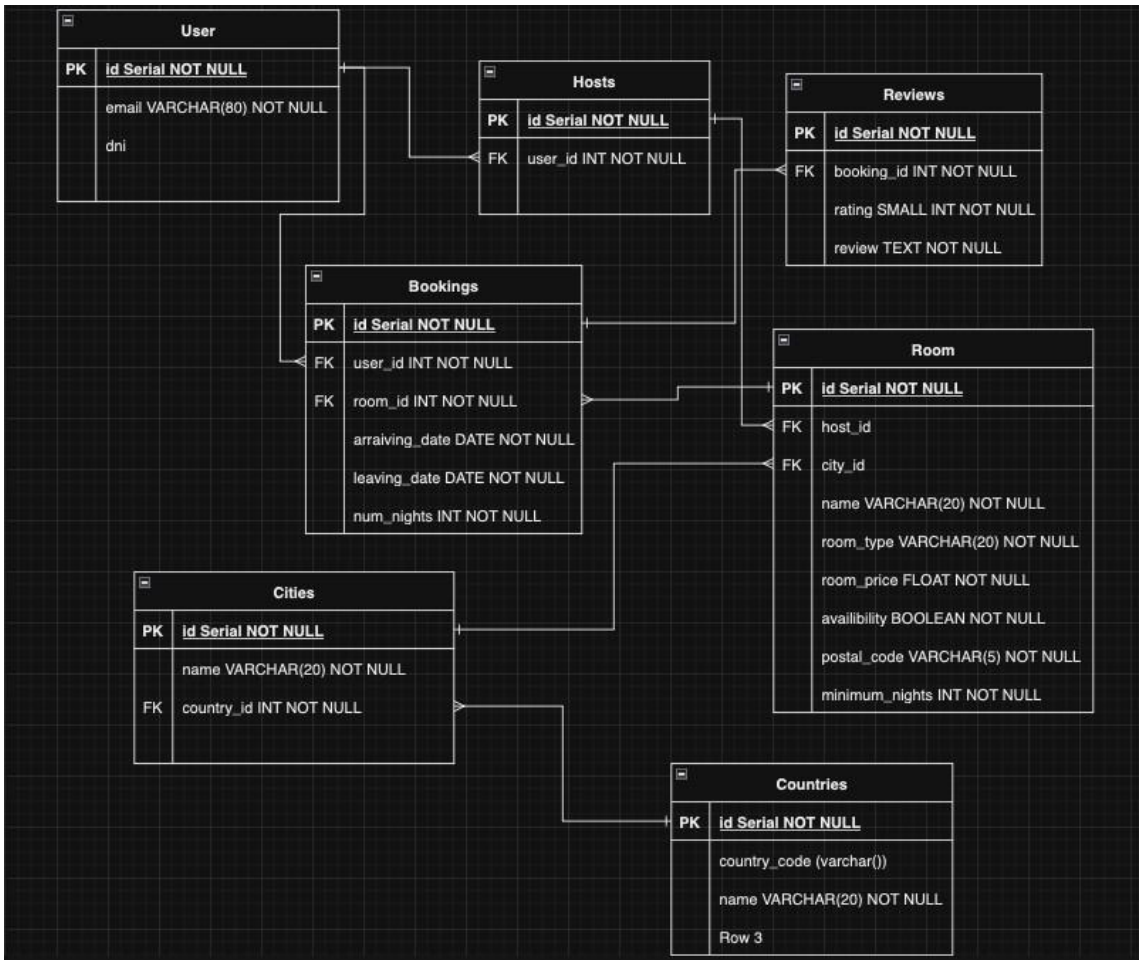
Este conjunto de datos tiene aplicado un filtro y muestra la información principal sobre las habitaciones disponibles en AirBnB en **Madrid** para un período de tiempo determinado.

Las columnas que posee son:

Location	Room type	Availability
Coordinates	Room Price	Rooms rent by the host
Room ID	Number of reviews	Number of reviews per month
Name	Minimum nights	Date last review
Host ID	Updated Date	

3. Arquitectura y Validación de los Datos.

También realizamos el diagrama entidad-relación del dataset mediante la aplicación draw.io (punto 2 del trabajo).



4. Análisis Exploratorio de Datos.

El objetivo del análisis es obtener datos que permitan elaborar un modelo para predecir el precio de alquiler turístico en Madrid.

4.1. Cargar Librerías.

- Utilización de librerías como pandas, numpy, seaborn y matplotlib.

En el análisis de datos, las librerías de Python como `pandas`, `numpy`, `seaborn` y `matplotlib` son herramientas esenciales para la manipulación, análisis y visualización de datos. Cada una de estas librerías ofrece funcionalidades específicas que, en conjunto, permiten llevar a cabo un análisis completo y robusto.

`Pandas` es una librería de código abierto que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y fáciles de usar. Se utiliza principalmente para la creación y manipulación de DataFrames y Series, que son estructuras de datos bidimensionales y unidimensionales respectivamente. `Pandas` facilita la limpieza de datos mediante el manejo de valores nulos, duplicados y transformaciones diversas. Además, permite realizar análisis exploratorios de datos,

como el cálculo de estadísticas resumidas y el filtrado de datos, y es capaz de leer y escribir datos desde y hacia varios formatos como CSV, Excel y SQL.

Por su parte, `numpy` es fundamental para la computación científica en Python. Esta librería proporciona soporte para matrices grandes y multidimensionales, junto con una amplia colección de funciones matemáticas para operar con estas matrices. `Numpy` es ideal para realizar cálculos numéricos de alto rendimiento, manipular arrays multidimensionales (ndarrays) y ejecutar operaciones de álgebra lineal, transformadas de Fourier y generación de números aleatorios.

`Seaborn`, basada en `matplotlib`, es una librería de visualización de datos que proporciona una interfaz de alto nivel para dibujar gráficos estadísticos, atractivos e informativos. Facilita la creación de gráficos estadísticos complejos, como diagramas de dispersión, gráficos de barras, gráficos de violín y mapas de calor. Además, permite explorar las relaciones entre múltiples variables y categorizar datos de manera visual. `Seaborn` también mejora la estética de los gráficos mediante estilos y paletas de colores personalizables.

Finalmente, `matplotlib` es una librería de trazado de gráficos en 2D que produce figuras de calidad publicable en una variedad de formatos impresos y entornos interactivos de Python. Con `matplotlib`, se pueden generar una amplia gama de gráficos, incluyendo gráficos de líneas, gráficos de barras, histogramas, diagramas de dispersión y gráficos de pastel. La librería permite personalizar gráficos ajustando etiquetas, títulos, leyendas y escalas, y soporta la visualización interactiva mediante su integración con interfaces gráficas de usuario y aplicaciones interactivas.

En el contexto del análisis de datos en el notebook, estas librerías se utilizaron de manera integrada para llevar a cabo un análisis exhaustivo. `Pandas` se empleó para cargar datos desde un archivo CSV, limpiar y manipular los datos, incluyendo la normalización de nombres de columnas y el filtrado de filas, y para seleccionar y transformar datos relevantes para el análisis. `Numpy` se utilizó para realizar operaciones numéricas básicas y manipular arrays, manejando valores faltantes o nulos. `Seaborn` se encargó de la creación de gráficos estadísticos para visualizar la distribución de precios y la relación entre variables, generando gráficos de barras y diagramas de dispersión para analizar diferentes tipos de habitaciones y su relación con el precio. Finalmente, `matplotlib` se utilizó para personalizar los gráficos creados, ajustando etiquetas, títulos y estilos para mejorar su presentación.

4.2. Carga de Datos.

- Se carga el archivo `air-bnb-listings.csv` en un DataFrame de pandas.
- Visualización inicial de los datos con `df.head()` para comprobar qué datos cargamos.

4.3. Información Inicial.

- Se utiliza `df.info()` para obtener una visión general del DataFrame.
- Nuestro DataFrame tiene 17 columnas y 1.414.018 filas. A simple vista podemos observar que algunas columnas tienen valores nulos o faltantes (la 1, 8, 9 y 12). Más adelante decidiremos cómo vamos a tratarlos. También revisamos el tipo de datos que manera cada columna y la cantidad de memoria que utiliza el DF.
- Antes de llegar al punto de seleccionar columnas para retirar las que no nos interesan hemos visto oportuno limpiar la cabecera del .CSV para que los nombres de las columnas que vamos a analizar tengan el mismo formato y sean más manejables a la hora de escribirlas..

4.4. Normalización de Columnas.

- Se normalizan los nombres de las columnas para un manejo más sencillo.

4.5. Selección de Columnas.

- Selección de columnas relevantes para el análisis; 'location', 'coordinates', 'name', 'room type', 'room price', 'number of reviews', 'minimum nights', 'availibility', 'rooms rent by the host'.

4.6. Tratamiento de los Datos.

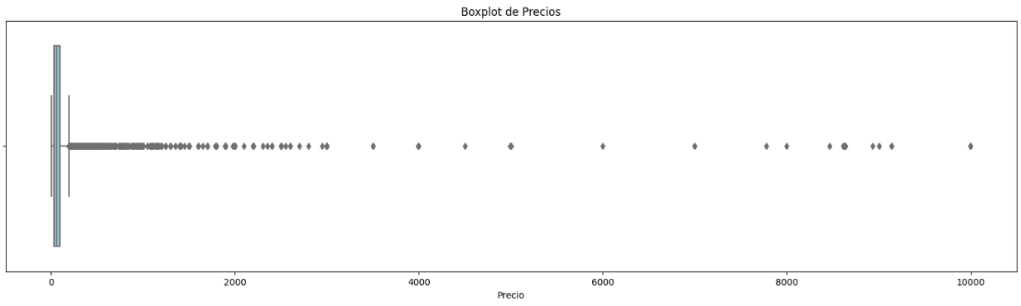
- Filtrado de filas para mantener sólo aquellas que contienen "Madrid" en la columna `location`.
- Eliminación de filas con valores nulos y limpieza adicional del DataFrame.
- Nuestro nuevo DF tiene un número de filas y columnas más reducido, así que ahora ocupa menos. La cabecera del DF ahora empieza en la fila 11.347 en lugar de en la 0. También nos encontramos con que por ahora solo manejaremos dos tipos de datos y con que solo la columna 'name' tiene valores NaN.
- Como los datos faltantes son cadenas de texto, en principio no debe de haber problemas, porque no afectarían a las operaciones matemáticas (como sí sucedería con enteros o flotantes), aún así, para que no queden vacíos los rellenaremos con un nombre genérico; ‘Nombre no Disponible’.
- Comprobamos si hay nombres que sean números o que tengan irregularidades y lo solucionamos asignándole ‘Desconocido’.
- Tratamos la columna ‘coordinates’ para separarla en ‘latitude’ y’ longitude’.
- Tratamos la columna 'location' para quedarnos solo con el barrio, ya que solo trabajaremos con alojamientos en Madrid.
- Comprobamos que seguimos contando con el mismo número de filas, así que estos últimos tratamientos no han supuesto pérdida de información.

- Llegando a este punto nos planteamos, ¿cuántas de las filas de nuestro DF tienen valor 0? Creemos que es importante comprobarlo para intentar comprender si esto puede afectar al modelo que intentamos construir.
- Ahora que sabemos qué columnas contienen ceros, podemos sacar conclusiones. En las columnas de disponibilidad y número de reseñas, el valor 0 no debe afectarnos al modelo, puesto que 0 indica que no está disponible el alojamiento y un alojamiento puede no tener reseñas. El 0 que sí es importante que sustituyamos es el de la columna precio. Un solo 0 no va a afectar mucho a los cálculos y resultados que estamos llevando a cabo, pero lo más correcto es referenciarlo y tratarlo, ya que si estuviésemos ante un DF con un número elevado de ceros en la columna precio, estaríamos haciendo cálculos sesgados que inducirían a errores en nuestro modelo.
- Como un Alojamiento Turístico no puede ser alquilado gratis a precio 0, vamos a imputarle a esta fila la mediana de los precios. Preferimos usar la mediana en lugar de la media porque es menos sensible a valores atípicos. Sabemos que el rango de precios es muy amplio, así que la mejor opción sin duda es la mediana.

4.7. Visualización Gráfica de Datos - Buscando Outliers y Relaciones.

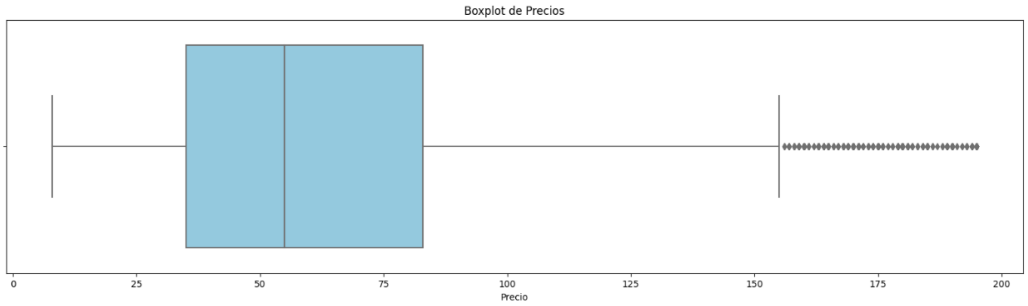
- Como posteriormente vamos a construir un modelo que prediga el precio, vamos a prestarle bastante atención a esta variable.
- Vemos que tenemos 21.255 registros, y que los precios fluctúan entre 8 y 9.999 euros, una fluctuación muy grande. En esta tabla leemos que:
    - mean: La media (promedio) de los valores en la columna "room price". En promedio, el precio de las habitaciones es de aproximadamente 163.99.
    - std: La desviación estándar de los valores en la columna "room price". Esto mide la dispersión de los valores con respecto a la media. En este caso, la desviación estándar es de aproximadamente 559.47.
    - min: El valor mínimo en la columna "room price". El precio mínimo de una habitación es de 8.
    - 25%: El percentil 25, también conocido como el primer cuartil o el cuartil inferior. Esto indica que el 25% de los precios de las habitaciones son iguales o inferiores a 36.
    - 50%: El percentil 50, que es la mediana. Esto indica que el 50% de los precios de las habitaciones son iguales o inferiores a 60.
    - 75%: El percentil 75, también conocido como el tercer cuartil o el cuartil superior. Esto indica que el 75% de los precios de las habitaciones son iguales o inferiores a 100.
    - max: El valor máximo en la columna "room price". El precio máximo de una habitación es de 9999.
- Empezaremos por localizar y eliminar los outliers o valores atípicos en el precio.

In [32]:
```
# Bloxplot solo del precio para ver outliers
plt.figure(figsize=(20, 5))
sns.boxplot(x=df_madrid["room price"], color='skyblue')
plt.title('Boxplot de Precios')
plt.xlabel('Precio')

plt.show()
```



- Vemos que hay muchos valores fuera de rango, tantos que la visualización de estos resulta compleja, así que emplearemos el método del rango intercuartílico para eliminarlos. Para ello calcularemos el rango intercuartílico (IQR), que es la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1). Los outliers se definen como aquellos valores que están por encima de Q3 + 1.5 * IQR o por debajo de Q1 - 1.5 * IQR.

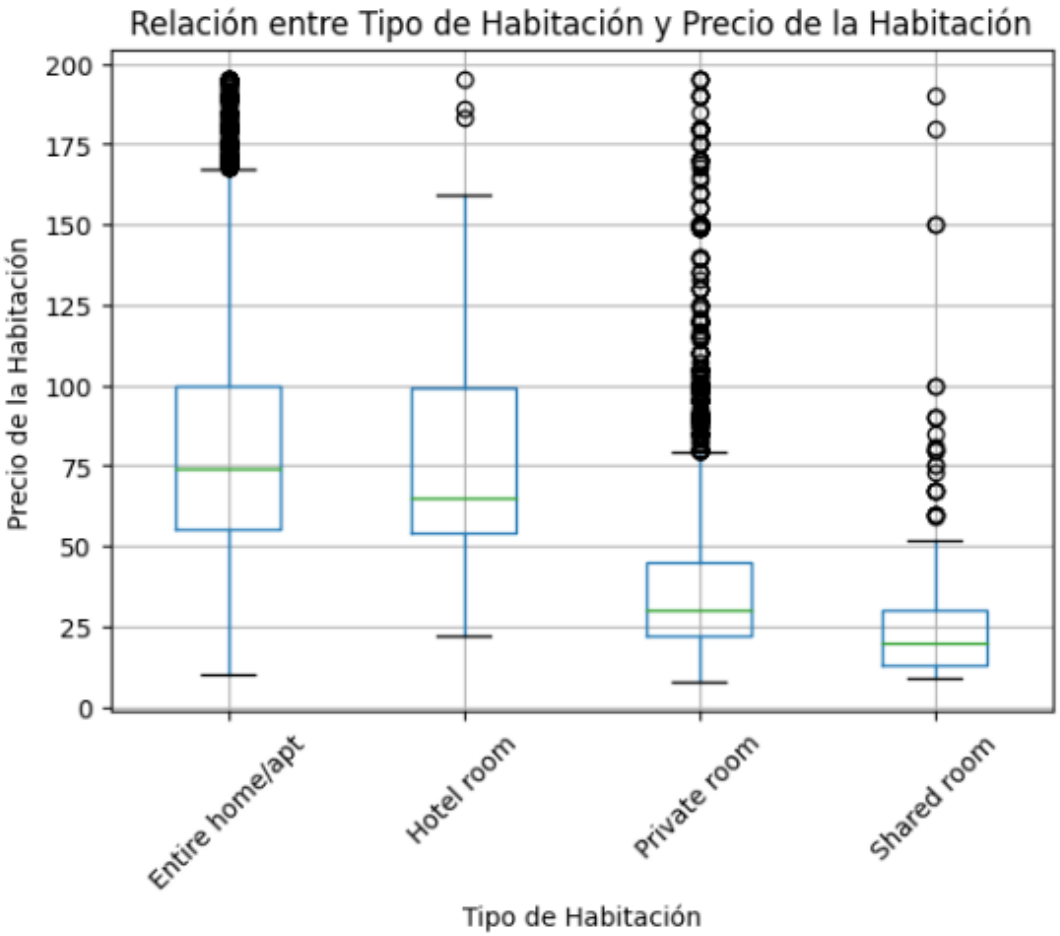
In [34]:
```
# Bloxplot de precios sin valores atípicos
plt.figure(figsize=(20, 5))
sns.boxplot(x=df_madrid_clean["room price"], color='skyblue')
plt.title('Boxplot de Precios')
plt.xlabel('Precio')

plt.show()
```



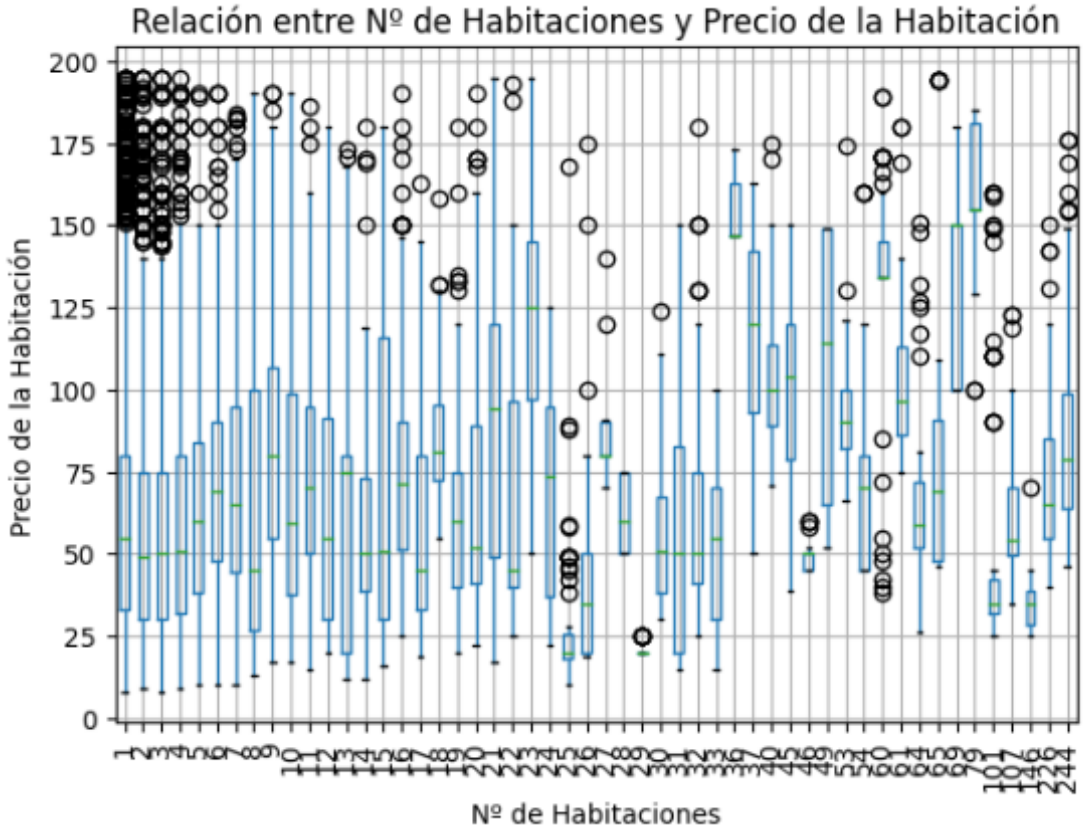
- Ahora vamos a intentar comprender cómo afecta cada tipo de variable al precio del alquiler turístico. Usaremos la matriz de correlación , que es una tabla que muestra las correlaciones entre todas las variables en un conjunto de datos. Cada celda en la tabla representa el coeficiente de correlación entre dos variables. Cuanto más alto sea el número más relacionadas están las variables. nos fijaremos cuáles afectan más al precio. Pero antes, necesitamos que las variables 'location' que contiene los barrios y 'room type', tenga un valor numérico para poder correlacionarla con el precio.

- Tras esta transformación ya podemos realizar la Matriz de Correlación.

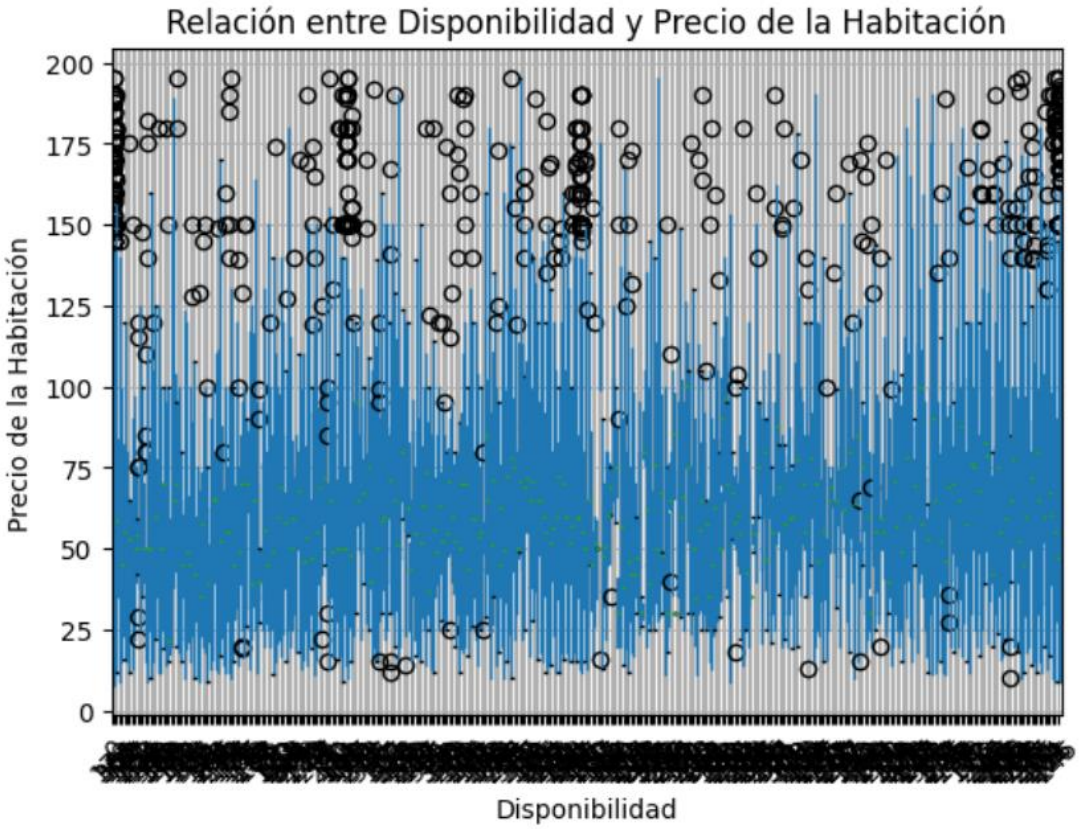
- Con esta gráfica ya podemos obtener conclusiones. Lo que más influye en el precio es el 'room type' (tipo de habitación), seguido bastante de lejos por 'rooms rent by the host' (habitaciones alquiladas por el anfitrión) y 'availibility' (disponibilidad).
- Visualizamos los datos por tipo de habitación y vemos que lo que más influye en el precio es si se trata de un apartamento entero o una habitación privada, ya que habitación compartida y habitación de hotel no tienen apenas representación en el DataFrame. Así que debemos eliminarlos para simplificar el modelo de cara a que tenga una mayor precisión. Así que empezaremos a filtrar los datos.



- Ahora visualizamos la relación del número de habitaciones con el precio.



- El número de habitaciones afecta ya que cuantas más habitaciones tiene el alojamiento más caro es su precio, seguramente esté también relacionado con el tamaño de las instalaciones y los m^2 totales que se están alquilando. Por eso también vamos a simplificar un poco. Aunque no dudamos de que puedan existir propiedades con 244 habitaciones, estamos casi seguras de que seguramente se trate de un error. Así que visualizando los datos, para nuestro modelo vamos a quedarnos con los datos de precio para los que tenemos más de 300 muestras. Esto supone cortar de DF quedándonos únicamente con las filas que se encuentren entre 1 a 7 habitaciones.
- Analizamos la disponibilidad y su relación con el precio.



- Aunque el gráfico no se distingue muy bien, a pesar de nuestros esfuerzos por hacer el eje X más largo y poder separar más las líneas, teniendo en cuenta que buscamos reducir la complejidad de los datos y su distribución, vamos a eliminar todas las disponibilidades que tengan una prevalencia menor a 300.

4.8. Guardado de Datos.

- El DataFrame tratado se guarda en un archivo CSV `CLEAN-air-bnb-listings.csv` para análisis futuros y para emplearlo en la elaboración de nuestro modelo.

4.10. Conclusiones.

A pesar de que en primera instancia podría parecer que lo que más influye en el precio es la localización de un alojamiento, de los datos que hemos trabajado se pueden extraer una conclusiones diferentes. Lo que más influye en el precio es el 'room type' (tipo de habitación), seguido bastante de lejos por 'rooms rent by the host' (habitaciones alquiladas por el anfitrión) y 'availability' (disponibilidad).

Hemos trabajado el DataSet original para simplificarlo y optimizarlo en función a lo que necesitaremos posteriormente para crear nuestro modelo con Regresión Lineal. Por eso hemos buscado limpiarlo de los valores atípicos de las variables que más influyen en el precio.

Si comparamos en DataSet inicial con el final guardado tras el Análisis Exploratorio de Datos, nos encontramos con:

4.10.1. DataFrame Inicial.

- Dimensiones: El DataFrame inicial tiene 1,414,018 filas y 17 columnas.
- Columnas:
    - `Room ID`: Identificador de la habitación.
    - `Name`: Nombre del alojamiento.
    - `Host ID`: Identificador del anfitrión.
    - `Neighbourhood`: Vecindario donde se encuentra el alojamiento.
    - `Room type`: Tipo de habitación.
    - `Room Price`: Precio de la habitación.
    - `Minimum nights`: Número mínimo de noches.
    - `Number of reviews`: Número de reseñas.
    - `Date last review`: Fecha de la última reseña.
    - `Number of reviews per month`: Número de reseñas por mes.
    - `Rooms rent by the host`: Número de habitaciones alquiladas por el anfitrión.
    - `Availibility`: Disponibilidad.
    - `Updated Date`: Fecha de actualización.
    - `City`: Ciudad.
    - `Country`: País.
    - `Coordinates`: Coordenadas.
    - `Location`: Ubicación.
- Valores nulos: El DataFrame inicial contiene 801,413 valores nulos distribuidos en varias columnas.

4.10.2. DataFrame Final.

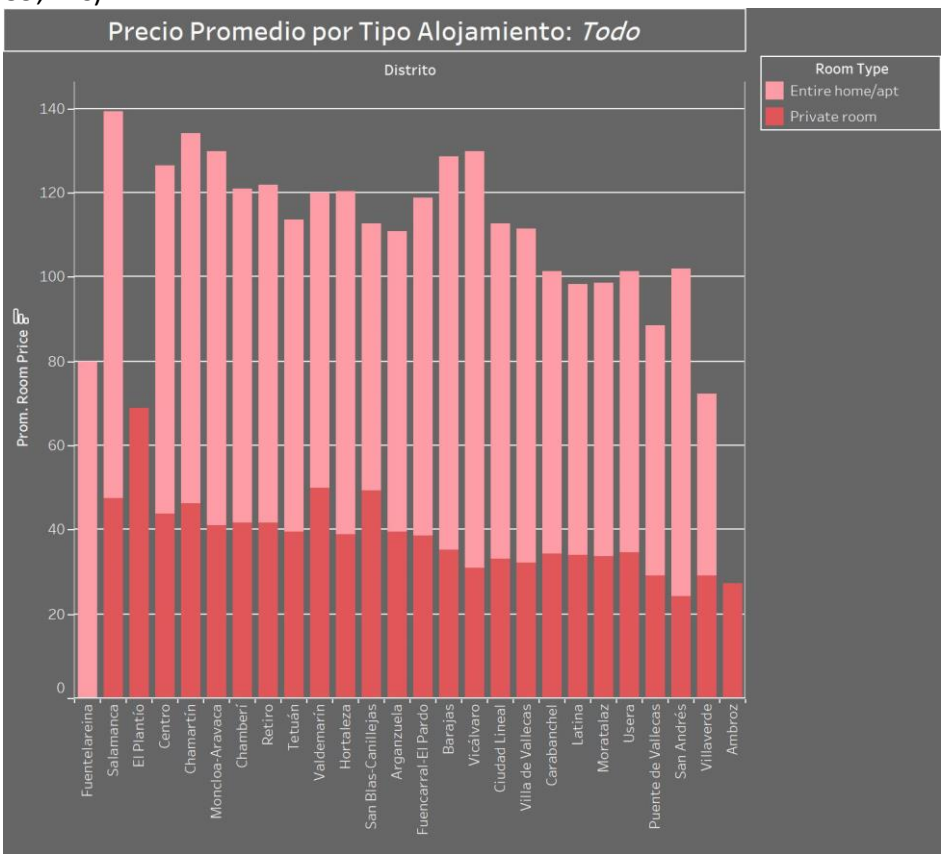
- Dimensiones: El DataFrame final tiene 7,830 filas y 9 columnas.
- Columnas:
    - `location`: Ubicación.
    - `coordinates`: Coordenadas.
    - `name`: Nombre del alojamiento.
    - `room type`: Tipo de habitación.
    - `room price`: Precio de la habitación.
    - `number of reviews`: Número de reseñas.
    - `minimum nights`: Número mínimo de noches.
    - `availibility`: Disponibilidad.
    - `rooms rent by the host`: Número de habitaciones alquiladas por el anfitrión.
- Valores nulos: El DataFrame final no contiene valores nulos.

5. Visualización de las métricas.

Realización de dashboard en Tableau.

- Para trabajar con datos geográficos de Madrid hemos implementado dos nuevas columnas referentes a código postal y distritos de Madrid.

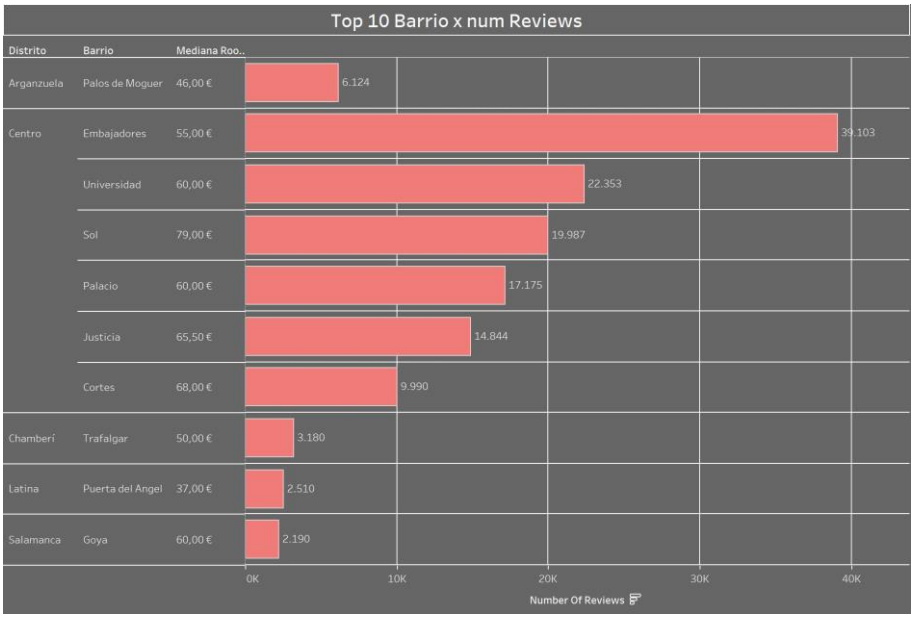
- Se observa que el **precio por habitación varía según el tipo de habitación** (Room Type), siendo el precio medio mayor en los apartamentos enteros (entire home/apt) que por habitación privada (Private Room) (80,47€ vs 39,72€).



- Se observa que las localizaciones donde los precios son más altos no son las mismas para el alquiler de todo el apartamento que el de una habitación, lo cual también concuerda con lo confirmado en el punto anterior. (precio x localización privada y apartamento entero)

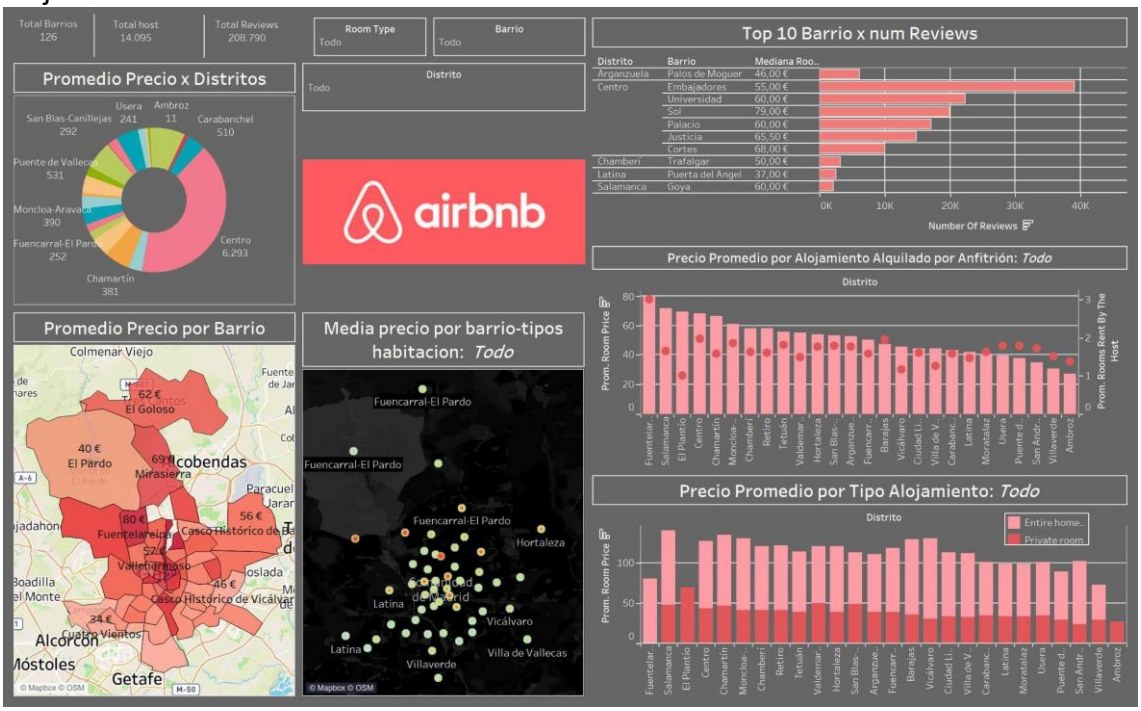
- El **número mínimo de noches** no tiene relación con el precio, hay algunas localizaciones como el Viso que tiene un mínimo de 12,97 noches y el precio es más bajo que por ejemplo en Recoletos, donde el número de noches mínimas es de 7,75 (gráfico precio-localización.reviews) gráfico precio por noches

- La cantidad de **reviews** está directamente vinculada al distrito con mayor número de hospedajes, pero esto no afecta al precio, no por tener más reviews el alojamiento es más caro.



- Podemos observar también que la **disponibilidad del alojamiento** afecta al precio.

- El número medio de **habitaciones alquiladas por el host** se encuentra en 2 habitaciones y vemos que es determinante en el precio de alquiler del alojamiento.



-

6. Pre-procesamiento y Modelado.

Cargamos el EDA preprocesado que realizamos en el apartado ‘4. Análisis Exploratorio de Datos’ para elaborar un modelo que prediga el precio de un alquiler turístico en Madrid.

6.1. Cargamos las librerías necesarias.

- Empleamos las librerías pandas, numpy, seaborn, matplotlib (ya instroducidas anteriormente en este informe) y añadimos el uso de scikitlearn.
- Scikit-learn es una librería de Python para el aprendizaje automático que incluye herramientas simples y eficientes para la minería de datos y el análisis de datos. Se utiliza en varios aspectos del análisis de datos y la construcción de modelos. En cuanto a modelos de regresión, se incluye LinearRegression para la construcción de modelos de regresión lineal, Ridge para la construcción de modelos de regresión Ridge (regresión lineal con regularización L2) y Lasso para la construcción de modelos de regresión Lasso (regresión lineal con regularización L1). Para el preprocesamiento de datos, StandardScaler se utiliza para el escalado y normalización de características. La división de datos se realiza con train_test_split, que divide los datos en conjuntos de entrenamiento y prueba. En la búsqueda de parámetros, GridSearchCV se emplea para encontrar los hiperparámetros óptimos mediante validación cruzada. Para la evaluación de modelos, se utilizan mean_squared_error para calcular el error cuadrático medio y evaluar la precisión del modelo, y r2_score para calcular el coeficiente de determinación (R^2) y evaluar el rendimiento del modelo.

6.2. Carga de Datos.

- Se carga el archivo `CLEAN-air-bnb-listings2.csv` en un DataFrame de pandas.
- Visualización inicial de los datos con `df.head()` para comprobar qué datos cargamos.

6.3. Pasar Datos Categóricos a Numéricos en 'room type'.

- En el análisis y modelado de datos, a menudo encontramos variables categóricas, como 'room type' en el caso de alquileres turísticos. Los modelos de aprendizaje automático y estadísticos generalmente requieren datos numéricos como entrada, ya que las operaciones matemáticas subyacentes no pueden manejar directamente las etiquetas categóricas. Por lo tanto, es crucial convertir las variables categóricas en un formato numérico antes de entrenar el modelo.
- Para ello empleamos la técnica de Codificación de Etiquetas (Label Encoding). Asigna un número entero único a cada categoría. Por ejemplo, para la variable

'room type', las categorías 'Entire home/apt', 'Private room' han sido codificadas como ‘Private room': 0, 'Entire home/apt': 1.

6.4. Estandarización de datos y modelo.

- Comprobamos si los datos tienen escalas diferentes empleando la desviación estándar. Así sabremos si es de utilidad normalizarlos antes de construir el modelo.

In [5]:
```
# Desviación estándar de las características más amplias
std_deviation = df[['rooms rent by the host', 'availibility']].std()

print("Desviación estándar de características:")
print(std_deviation)
```

```
Desviación estándar de características:
rooms rent by the host      1.376601
availibility              150.310732
dtype: float64
```

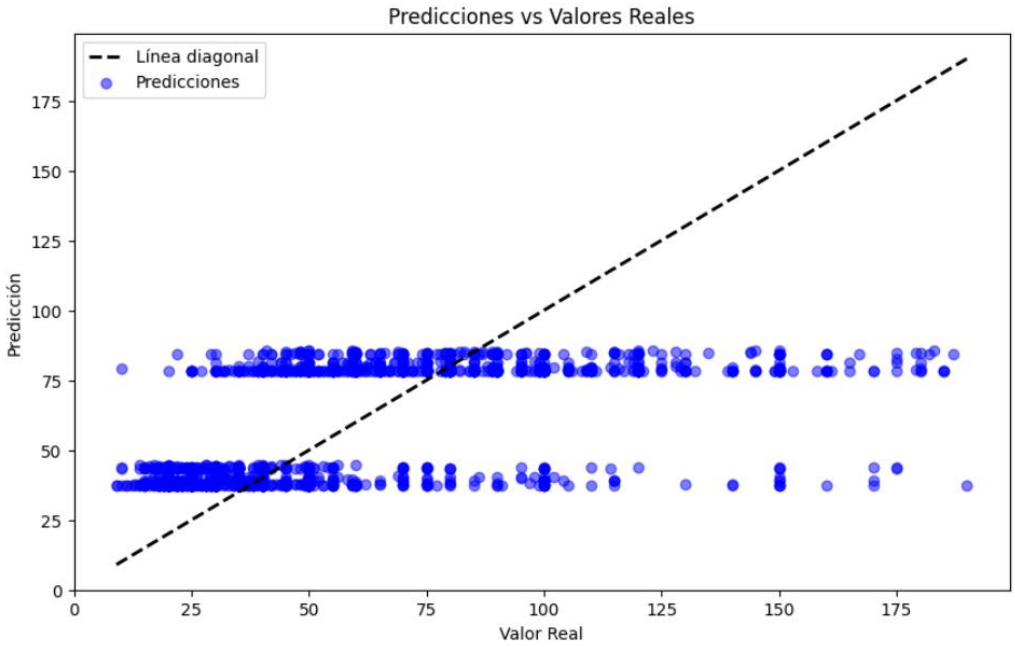
- La característica 'rooms rent by the host' tiene una desviación estándar relativamente pequeña (1.376601), mientras que 'availability' tiene una desviación estándar significativamente mayor (150.310732). Por ello 'availability' puede tener valores que varían en un rango mucho más amplio en comparación con 'rooms rent by the host'. La normalización puede ser útil para garantizar que todas las características estén en una escala similar, lo que puede ayudar al algoritmo de regresión lineal a converger más rápido y a evitar que características con magnitudes más grandes dominen el modelo.
- Por ello emplearemos StandardScaler en la construcción del modelo.

```
Media de las características en X después de la estandarización:
[6.12536841e-17 4.53730993e-18]

Desviación estándar de las características en X después de la estandarización:
[1. 1.]
```

- Dividimos los datos, entrenamos el modelo de regresión lineal y evaluamos los resultados empleando el Error Cuadrático Medio y el Coeficiente de Determinación. Obtenemos estos resultados:

```
Mean Squared Error: 984.657556699563
R^2 Score: 0.3104253913398999
```

- Los resultados del modelo de regresión indican un Error Cuadrático Medio (ECM) de 984.66, lo que sugiere que las predicciones del modelo tienen, en promedio, un error significativo en comparación con los valores reales, indicando una precisión moderada. El Coeficiente de Determinación (R^2) es 0.31, lo que significa que solo el 31% de la variabilidad en los precios de alquiler es explicada por las variables del modelo, señalando que el modelo tiene un poder explicativo limitado y que hay otros factores importantes que no están siendo considerados en el análisis. No son resultados muy buenos, así que intentamos mejorarlos.
- Graficamos para obtener más conclusiones:



- Los resultados para este modelo de regresión lineal son muy modestos en la predicción del precio de los alojamientos turísticos, podemos observar que existe underfitting. Tenemos un MSE relativamente alto y un R^2 que explica sólo una parte moderada de la variabilidad en los datos. Así que vamos a intentar mejorarlo optimizando hiperparámetros mediante técnicas de regularización.

6.5. Optimización de hiperparámetros con GridSearchCV para Ridge y Lasso.

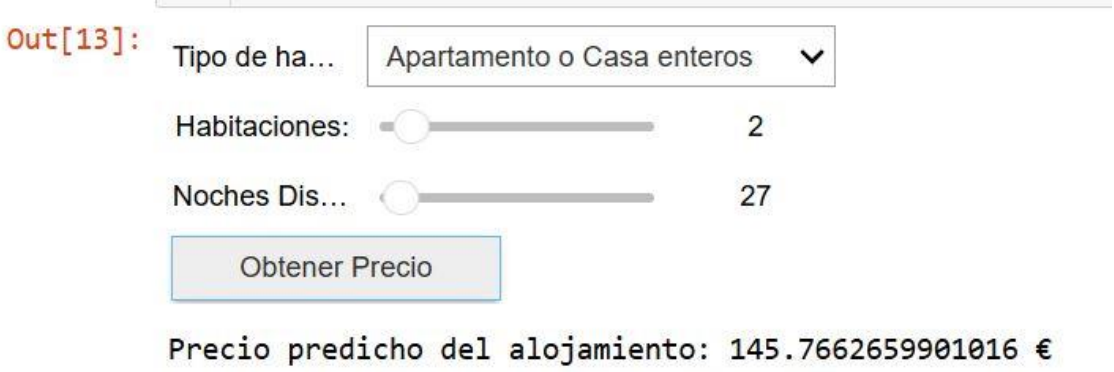
- La optimización de hiperparámetros con GridSearchCV para los modelos Ridge y Lasso se utiliza para identificar los valores óptimos de regularización que mejoran el rendimiento del modelo, evitando el sobreajuste. Ridge introduce una penalización L2 y Lasso una penalización L1 en la función de error, ayudando a manejar la multicolinealidad y a seleccionar características relevantes, respectivamente. GridSearchCV evalúa de manera sistemática diferentes combinaciones de hiperparámetros para determinar los que ofrecen el mejor desempeño en términos de precisión y generalización del modelo.

```
Ridge Regression:
Best Alpha: 1
Mean Squared Error: 984.5711528365132
R^2 Score: 0.3104859016255763

LASSO Regression:
Best Alpha: 0.1
Mean Squared Error: 985.0372084545874
R^2 Score: 0.3101595139202663
```

- Conseguimos una leve mejora de los resultados con Ridge Regression. Hay que dejar claro que podríamos llevar a cabo otras acciones para seguir mejorando el modelo.

6.6. Guardamos el modelo, lo cargamos y lo probamos.

- Una vez hemos guardado nuestro modelo volvemos a cargarlo para hacer unas pruebas de predicción empleando la librería ipywidgets.
- Ipywidgets es una biblioteca de Python que permite crear controles interactivos y widgets en Jupyter Notebooks, facilitando la interacción y visualización de datos. Con ella podemos utilizar controles como deslizadores, menús desplegables, botones y casillas de verificación para manipular y explorar datos de manera dinámica.



7. Conclusiones del Informe.

7.1. Suposiciones Iniciales.

Nuestra suposición inicial era que la ubicación de los alojamientos en diferentes barrios tendría la mayor influencia en el precio. En general, se espera que los alojamientos situados en los distritos más céntricos y populares sean los más caros debido a su proximidad a las principales atracciones, servicios y transporte público. Además, pensábamos que el número de habitaciones de un alojamiento estaría estrechamente relacionado con el precio, ya que una mayor cantidad de habitaciones generalmente implica una mayor superficie y, por lo tanto, un precio más alto.

En teoría, estos factores parecían ser los determinantes más obvios y lógicos del precio. La ubicación, por su atractivo y conveniencia, y el tamaño del alojamiento, reflejado en el número de habitaciones, parecían ser las variables más influyentes. Sin embargo, tras realizar un análisis exhaustivo de los datos, nuestros resultados revelaron una realidad diferente.

El análisis de datos nos mostró que, contrario a nuestra suposición inicial, la ubicación no es el factor predominante que determina el precio de los alojamientos. Como podemos extraer del Análisis de Datos con Python, la correlación entre ubicación y precio es de -0.17540061422614808, lo que indica una relación negativa y débil entre estas dos variables en el contexto de los alojamientos turísticos.

Aunque la ubicación tiene algún impacto, no es tan significativo como esperábamos. La variable que realmente sobresale en su influencia sobre el precio es el 'room type' (tipo de habitación). Este hallazgo sugiere que los huéspedes valoran más el tipo de espacio que alquilan, como si es una habitación privada o un apartamento completo, que la localización específica del alojamiento.

Además del tipo de habitación, otras variables que demostraron tener una influencia considerable, aunque menor en comparación, fueron 'rooms rent by the host' (habitaciones alquiladas por el anfitrión) y 'availibility' (disponibilidad). La cantidad de habitaciones que el anfitrión alquila puede indicar que se trata de un espacio más amplio a alquilar. La disponibilidad del alojamiento también juega un papel importante, ya que refleja la demanda y la flexibilidad del espacio para los potenciales huéspedes.

7.2. Métricas Seleccionadas.

Las métricas seleccionadas para nuestro análisis de precios han sido 'location', 'coordinates', 'name', 'room type', 'room price', 'number of reviews', 'minimum nights', 'availability', 'rooms rent by the host'. Son las que hemos relacionado con el precio de los alojamientos, empleando para ello la correlación entre las variables.

La correlación es una medida estadística que varía entre -1 y 1. Un valor de 1 indica una correlación positiva perfecta: a medida que una variable aumenta, la otra también lo hace. Un valor de -1 indica una correlación negativa perfecta: a medida que una variable aumenta, la otra disminuye.

De esta manera hemos podido contrastar hipótesis y evaluar las suposiciones iniciales.

Una vez hemos elaborado el modelo de predicción de precios con regresión lineal, hemos trabajado con métricas de rendimiento para evaluar la eficacia de este. En concreto, hemos empleado el Error Cuadrático Medio (MSE) y el Coeficiente de Determinación (R^2). Son métricas esenciales para evaluar el rendimiento de un modelo de regresión. El MSE mide la precisión del modelo en términos de la magnitud del error, mientras que el R^2 mide qué tan bien las variables independientes explican la variabilidad de la variable dependiente.

7.3. Teniendo en Cuenta lo Aprendido.

Teniendo en cuenta lo aprendido, en este proyecto lo haríamos casi todo igual, a excepción de la fase de prototipado del modelo, ya que habríamos querido conseguir que tuviera una mayor precisión al hacer predicciones. Para ello podríamos haber realizado acciones como estas:

- Podríamos haber intentado buscar más datos.
- Emplear técnicas de oversampling y equilibrado de muestras.
- Haber probado con otros modelos como Árboles de Decisión, Random Forest o Redes Neuronales.

7.4. Conclusiones.

- **Influencia del Tipo de Habitación**:

Nuestro análisis exploratorio de datos (EDA) y el modelado de regresión lineal mostraron que el tipo de habitación ('room type') es el factor más determinante en la predicción del precio de los alquileres en Airbnb. Este hallazgo sugiere que los huéspedes valoran significativamente si están alquilando una habitación compartida, privada o un apartamento entero.

- **Impacto de la Disponibilidad y el Número de Habitaciones**:

La disponibilidad ('availability') y el número de habitaciones alquiladas por el anfitrión ('rooms rent by the host') también demostraron tener una influencia notable, aunque menor que el tipo de habitación. Esto indica que la flexibilidad y el tamaño del alojamiento son factores importantes, pero no tan críticos como el tipo de habitación.

- **Relación Débil con la Ubicación**:

Contrario a nuestras suposiciones iniciales, la ubicación mostró una correlación débil con el precio (-0.1754). Esto sugiere que, aunque la ubicación es tradicionalmente vista como un factor clave, otros aspectos del alojamiento pueden tener un impacto más significativo en los precios de alquiler en el contexto de Airbnb.

- **Precisión del Modelo**:

El modelo de regresión lineal resultó en un Error Cuadrático Medio (ECM) de 984.66 y un Coeficiente de Determinación (R^2) de 0.31. Estos resultados indican que el modelo tiene una precisión moderada y puede explicar aproximadamente el 31% de la variabilidad en los precios, lo que propone la necesidad de considerar modelos más complejos o variables adicionales para mejorar la predicción.

7.5. Lecciones Aprendidas.

- **Importancia del EDA**:

El análisis exploratorio de datos es crucial para entender la estructura y las relaciones dentro del dataset. Herramientas como histogramas, diagramas de dispersión y matrices de correlación nos permitieron identificar patrones y relaciones que no eran evidentes inicialmente.

- **Limitaciones de la Regresión Lineal**:

La regresión lineal, aunque simple y fácil de interpretar, puede no capturar todas las complejidades de los datos, especialmente cuando hay relaciones no lineales o interacciones entre variables. Considerar modelos más avanzados como Random Forest o Gradient Boosting podría mejorar la precisión de las predicciones.

- **Importancia de las Variables**:

La elección de las variables predictoras es crítica. Nuestro modelo mostró que variables como el tipo de habitación tienen un mayor impacto en los precios de alquiler que la ubicación. Esta lección subraya la importancia de seleccionar cuidadosamente las características basadas en el contexto del problema.

- **Preprocesamiento de Datos**:

La limpieza y preparación de los datos son pasos fundamentales. Manejar los valores nulos, normalizar las variables y transformar las variables categóricas en numéricas (por ejemplo, con codificación one-hot) son prácticas esenciales que pueden influir significativamente en el rendimiento del modelo.

- **Validación y Evaluación del Modelo**:

La validación cruzada y el ajuste de hiperparámetros son vitales para evaluar el rendimiento del modelo de manera robusta y evitar errores de ajuste. Además, es importante utilizar múltiples métricas de evaluación para obtener una visión más completa del rendimiento del modelo.

- **Iteración y Mejora Continua**:

El desarrollo de modelos es un proceso iterativo. Los insights obtenidos de las evaluaciones iniciales deben usarse para refinar y mejorar el modelo continuamente. La incorporación de nuevos datos y características, así como la experimentación con diferentes algoritmos, puede llevar a mejoras significativas en las predicciones.

- Como conclusión final nos gustaría agregar, que cuando se trata de datos ser debe ser prudente y escéptico, ya que las suposiciones que resultan más obvias pueden no ser ciertas, aunque parezcan muy evidentes.

8. Recursos Adicionales.

- Repositorio en GitHub: https://github.com/SoleGrobas/PracticaFinalkeepcoding
- Web del Equipo y Proyecto: https://datasilvia.github.io/YellowDataTeam/