# Arquitectura y validación de los datos

*Lenguaje de programación aplicado: Python.*

## **Primeros pasos**

Para empezar el proyecto, lo primero que se debe hacer es analizar el dataset proporcionado, revisando el tipo de datos que hay y la calidad de estos mismos. Es importante ver también qué columnas del dataset son necesarias y relevantes, para así poder focalizar los datos y crear la base de datos necesaria para conseguir el objetivo que es determinar a qué anfitriones registrados hay que enfocar la campaña de buenas prácticas para que alcancen el nivel de superhost antes del 31 de diciembre de 2017, teniendo en cuenta que los nuevos anfitriones adheridos a través de la campaña conseguirán beneficios exclusivos durante el año 2018 y los superhost que mantengan su status durante todo el año 2017 también recibirán beneficios exclusivos durante el primer trimestre de 2018.

Una vez se tenga claro qué columnas son las necesarias, el siguiente paso es hacer el modelo de la base de datos. Inicialmente se hace una sola tabla con todas las columnas que se vayan a usar y el tipo de dato va a ser el dato inicial que viene ya con el dataset.

La base de datos ha sido creada en Postgres con ayuda de Jupyter Notebook para hacer las conexiones con las bases de datos, para que no exista posibilidad de que ningún dato se quede perdido y poder cargarlos sin problema para su correcta visualización; apoyado por Talend para la una mejor integración de todos los datos y por tanto el uso de ETL para la extracción, transformación y carga de los datos en un solo repositorio; posteriormente esta herramienta ha facilitado a su vez, la creación del proceso Data Warehouse.

En esta fase, como se comenta anteriormente, se introduce la librería que se va a usar para la conexión con la base de datos, se importa la librería de Pandas, se introduce la base de datos, se eliminan las columnas que no se vayan a utilizar en el desarrollo del trabajo y se establece la conexión con la base de datos.

Después de traer la información del dataset y agregarla a la base de datos, para verificar que esa información sí está en la base de datos, se hace una desconexión para que no se generen errores. En Postgres, se puede verificar que el proceso de mover todos los datos se da sin ningún problema. Todo esto que se explica anteriormente se emplea para tomar dos caminos de desarrollo: por una parte se emplea para realizar el Data Warehouse y por otra parte para realizar el análisis de los datos.

Un paso para organizar, comprender mejor y acudir a la información seleccionada de manera sencilla, es crear un diccionario con la explicación de las columnas y comprobar cuál es el tipo de dato contiene cada una de estas.

## **Limpieza del dataset y base de datos**

Para realizar el análisis y la exploración principal de los datos, se usa en este proyecto un cuaderno de Colab; para ello, se va a introducir la información del dataset inicial y las columnas que ya han sido elegidas en el paso anterior. A continuación se instalan todas las librerías que se requieran (mapas, geolocalización, Pandas, NumPy…).

En este paso se puede visualizar cómo son los datos, si son o no nulos, en el caso de que sean nulos, muchos de ellos se pueden eliminar puesto que podrían llevar a confusión para el posterior análisis. También existe la opción de imputar los datos, sobreescribiendo el dato por un valor que se considere probable para esa variable; para este paso se usa el dato que sea probabilísticamente mayor, en este como es el caso de Monthly Price, Weekly Price, Review Scores Value, Host, Zip Code, etc.

De las 88 columnas que eran, se eligen 36 columnas. Una vez que ya se ha seleccionado y acotado la información de las columnas, se revisa y se vuelve a comprobar para poder seguir editando pequeños matices, como por ejemplo, pasar todas las variables categóricas a letras minúsculas. Aún se pueden hacer más acotaciones, en este caso la ciudad objeto de estudio es Madrid, por lo tanto se filtran los datos que sean sólo de Madrid, ya que el dataset tiene información de otras ciudades nacionales e internacionales.

### **Análisis de las variables categóricas**

En el análisis de los datos categóricos se le da especial importancia a las columnas *host* y *superhost*, en donde *host* se convierte en *false* y *superhost* en *true*, a continuación se revierte el proceso: se reemplaza *true* por *superhost* y *false* por *host*. Este paso se realiza para evitar confusiones y para que se pueda hacer el cambio automático cuando busquemos los datos para el análisis posterior.

Se analiza la columna *Neighbourhood Group Cleansed* con el fin de crear después un diccionario, para que los valores sean únicos y tengan su código postal correspondiente. Se agrupan *Neighbourhood Cleansed* con *Zipcode* y se trabaja con esos valores. Luego, se van filtrando los errores y valores nulos hasta que ya queden limpias las columnas y los datos de estas sean claros y correctos, para poder trabajar de una manera más cómoda y ordenada con ellos.

### **Análisis de las variables numéricas**

Se lleva a cabo un procedimiento parecido al de las variables categóricas, en dónde se va acotando y filtrando los valores erróneos y nulos hasta conseguir la limpieza total de los datos. Se da especial importancia al precio de las rentas de los alojamientos.

La detección de los datos atípicos juega un papel muy importante en la toma de decisiones, puesto que distorsionan los resultados al cambiar el comportamiento del resto de los casos, afectando las medidas estadísticas utilizadas para representar a la muestra elegida (p.ej. el promedio) y porque afectan considerablemente a otras técnicas de análisis de datos, cuya base es la presencia de normalidad.

En ambos casos, en el análisis de las variables, aunque es una práctica habitual el tomar datos similares calculados con los métodos estadísticos robustos, para este proyecto en particular ha resultado más lógico prescindir de datos que no fueran a resultar importantes así como limpiar con exhaustividad el dataset proporcionado, ya que podría dar margen a errores de veracidad y precisión de los resultados finales.

También se ha limpiado rigurosamente el dataset y la información de sus columnas para poder visualizar los diagramas de barras generados con mayor claridad, ya que por el aspecto de estos en una fase inicial, habría sido complicado analizar la información; puesto que es prácticamente imposible ver los datos que reflejan, como es el caso del diagrama de geolocalización y/o funciones.

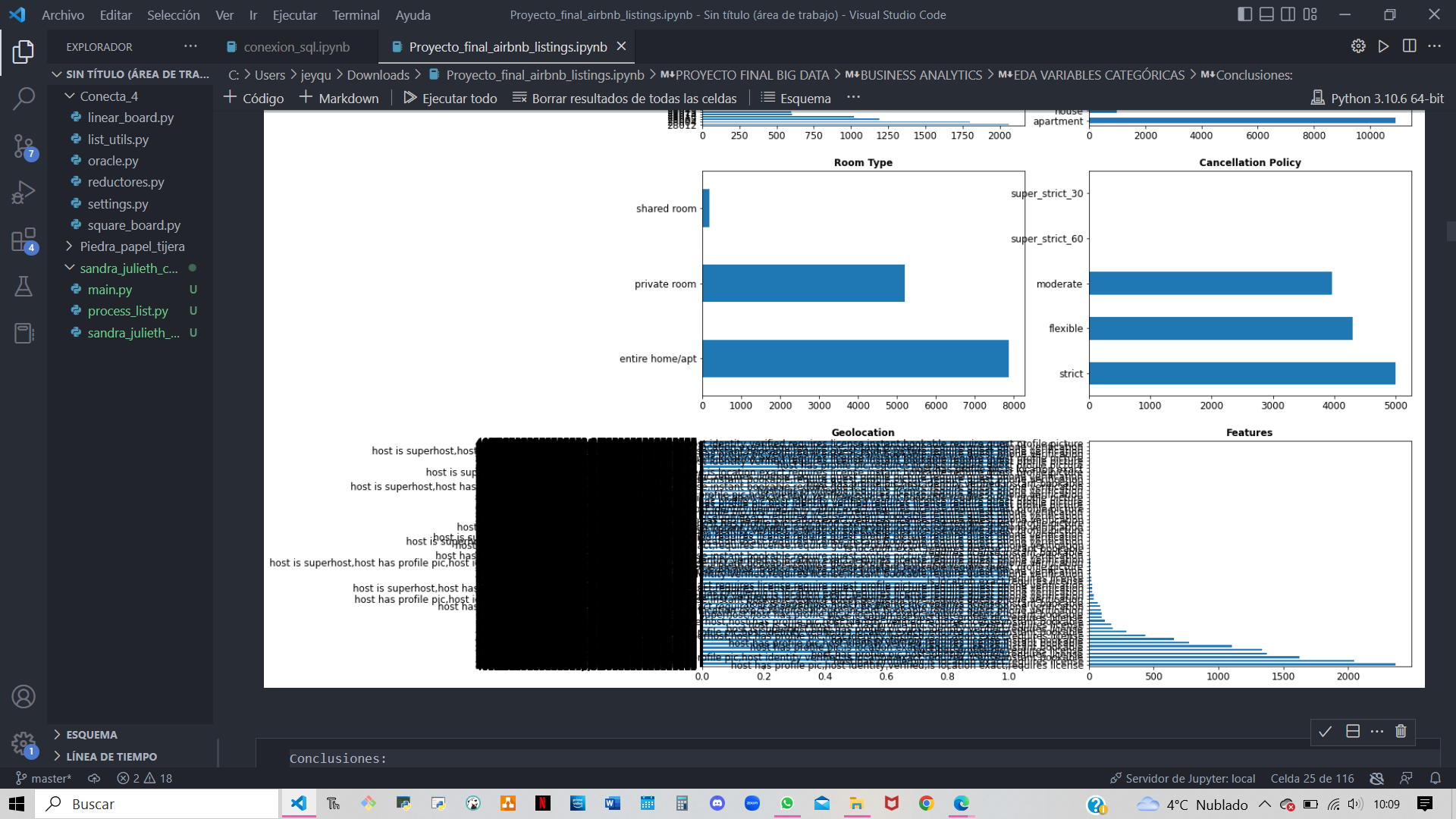


Diagrama de barras generado con Colab a partir del dataset de Airbnb. Diagrama\_1\_Geolocalización. Diagrama\_2\_Funciones.

## **Conclusiones**

En todo este proceso han ido surgiendo más preguntas, además de las planteadas a raíz del análisis de datos, habiendo marcado un objetivo principal, preguntas que se han ido contestando a lo largo de la realización del proyecto, para mayor calidad y satisfacción del cliente que ha requerido nuestros servicios.

Las conclusiones que podemos obtener de esta limpieza y corrección de los datos son:

* Podemos encontrar una diferencia notable entre la cantidad de *superhost* y *host* que existen, registrándose así 1554 tipos de usuario *superhost* con el 11.72% de los alojamientos y 11710 tipos de usuario tipo *host* con el 88.28% los alojamientos. Los Host tienen ratios de respuesta más bajos que los Superhosts.



Diagrama de barras generado a partir de la limpieza del dataset de Airbnb acerca de la cantidad de *host* y *superhost* que hay en Madrid.

* Según este análisis de los datos, el precio del alquiler depende de la zona donde se encuentre el *host*; la media del costo por alquiler tiende a ser más alta en Fuentelarreina y más baja en Pueblo Nuevo.

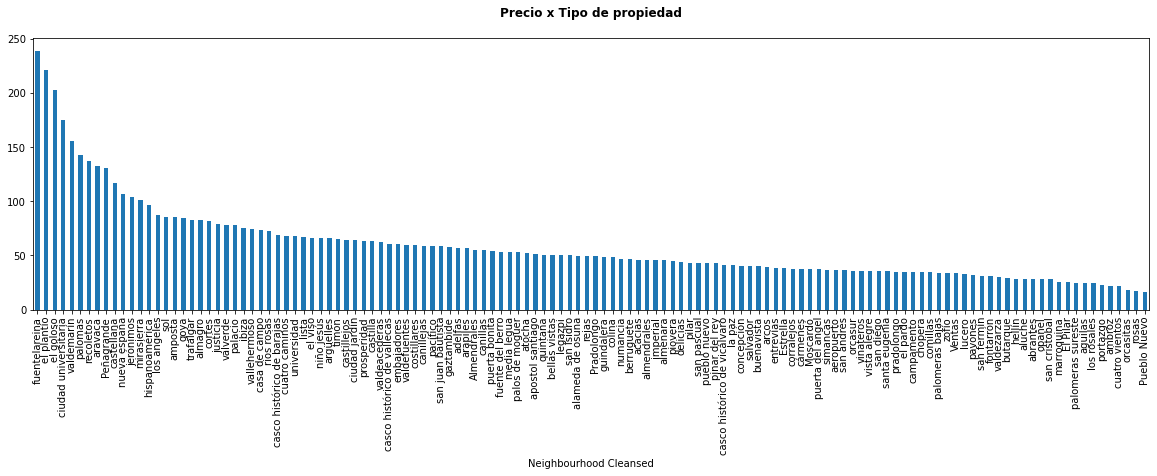
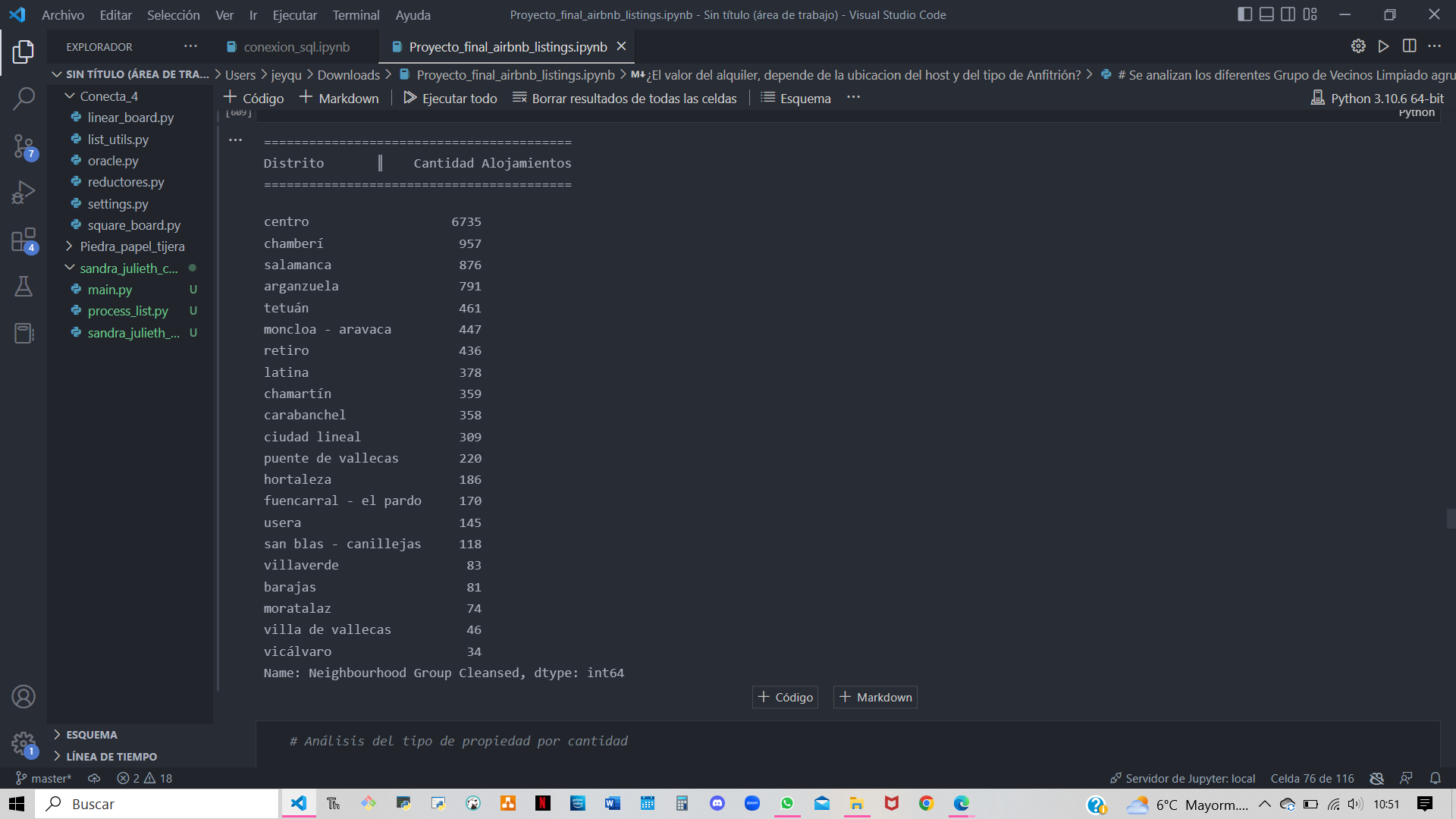


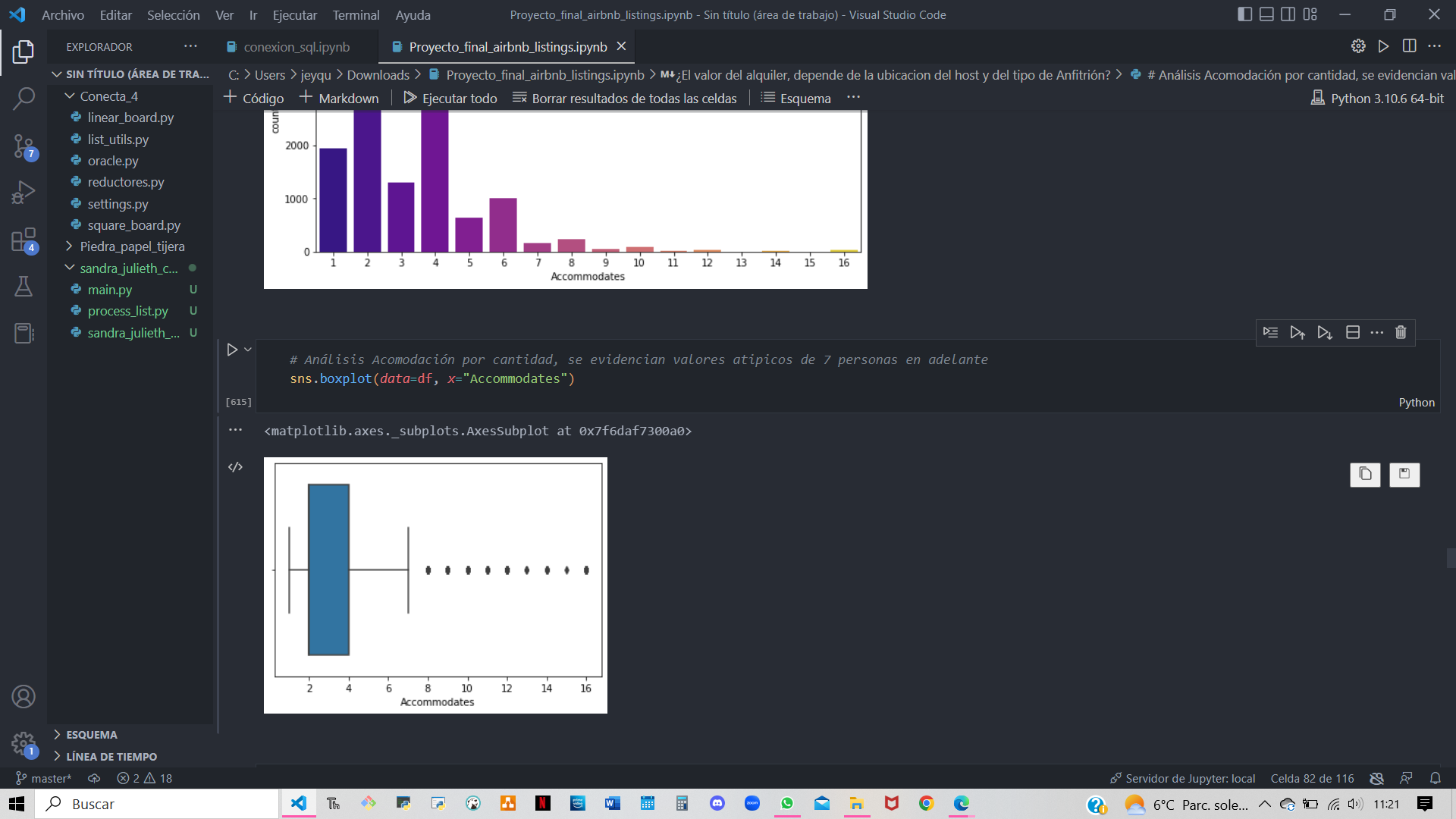
Diagrama de barras generado a partir de la limpieza del dataset de Airbnb acerca del tipo de propiedad que hay en Madrid.

* La cantidad de alojamientos es más alta en el distrito Centro de Madrid con 6735 *Neighbourhood Group Cleansed* y la más baja es en el Distrito Vicálvaro 34 *Neighbourhood Group Cleansed.*

**

Cuadro extraído de Colab del dataset de Airbnb acerca la cantidad de alojamientos que hay en los distintos distritos de Madrid

* El tipo de alojamiento más común es el apartamento con 10910. Se evidencia una gran diferencia con el resto de tipos de alojamiento siendo el segundo mayor requerido con un número de 999, las casas.
* En la variable *Accommodates*, se evidencian valores atípicos para las acomodaciones de más de 6 personas en adelante, las habitaciones más comunes son *entire home/apt* con una cantidad de 7875, la cantidad de habitaciones más común es una por vivienda y la menos común es 8, presentándose valores atípicos.



Cuadro extraído de Colab del dataset de Airbnb acerca de los valores atípicos de la variable Accommodates.

* Las habitaciones menos comunes son *shared room* con una cantidad de 192; la cantidad de camas más común es 1 por vivienda y la menos común es de 15 por vivienda. La cantidad de baños más común es 1 y la menos común es 7 baños, la cantidad de huéspedes más común que se presenta en esta limpieza de datos es 1 y la menos común es 14, presentándose valores atípicos en estas últimas tres variables.
* La mediana que se presenta en la cantidad de personas extra es de 7; el mínimo de noches tiene una mediana de 2 y el máximo tiene una mediana de 1125, presentándo en los dos casos, valores atípicos.
* La política de cancelación más común es *Strict* con 5004 viviendas con esta política, la menos común es *super\_strict\_30*, con 2 viviendas que usan esta política
* El *Host Id* nos indica que hay varios anfitriones con varios hospedajes a su nombre y con una mediana de 2.75; es decir, los anfitriones tienden a tener dos alojamientos registrados.

A este punto, se deja todo preparado para la siguiente parte del proyecto y así poder implementar las distintas herramientas que se han usado en la formación, en este caso R para el análisis exploratorio y el modelado de la regresión lineal y para Tableau en la que se hace un mapa ubicando los superanfitriones, número, y nombre del distrito.

DATA WAREHOUSE

En el almacén de los datos hubo un inconveniente, se intentó hacer el almacén de los datos con el dataset inicial, pero en cuanto se hacía la prueba de quitar las columnas que no habían sido seleccionadas, el dataset generaba error. Por tanto, finalmente se tomó la decisión de recurrir a SQL,en donde se hizo un job sencillo donde cargamos la tabla inicial donde estan los datos, esa tabla inicial la alimentamos desde el Jupyter Notebook, los datos vienen en lectura de datos, pasan por el team app y llegan a la tabla ODS. Se invierte mucho tiempo en sacar los datos del dataset al Data Warehouse. Se pasan los 147 y pico de datos y es así como los datos quedan grabados finalmente en SQL.

A la hora de hacer el almacenamiento de los datos surgió un problema, la idea era usar el Data Warehouse para esta parte como se menciona anteriormente en los primeros pasos; sin embargo, no ha sido posible, ya que se intentó hacer el almacén de los datos con el dataset inicial, pero en cuanto se hacía la prueba de quitar las columnas que no habían sido seleccionadas, el dataset generaba error. Por tanto, finalmente se tomó la decisión de recurrir a SQL, en donde se hizo un job sencillo donde cargamos la tabla inicial donde estan los datos, esa tabla inicial la alimentamos desde el Jupyter Notebook, los datos vienen en lectura de datos, pasan por el tMap y llegan a la tabla ODS. Se pasan los 14 780 datos y es así como se puede ver que los datos - tanto en la base de datos original como en el respaldo - quedan grabados finalmente en SQL.