# Proyecto Ingeniería de Datos: Airbnb SQL Análisis de Datos

**Integrantes:** Daniel Fandiño y Juan Andres Russy

**Enlace del repositorio:** https://github.com/juan-andres-russy/airbnb\_sql\_data\_analysis

El repositorio lo organizamos en carpetas de la siguiente manera:

* En la carpeta Entrega1/ realizamos las correcciones que se nos hicieron con respecto a la primera entrega.
* En la carpeta Entrega2/ realizamos las correcciones que se nos hicieron con respecto a la segunda entrega.
* En la carpeta Proyecto\_final/ se encuentra toda la versión final del proyecto, aquí se encuentra todos los códigos y explicaciones que se mostraran a continuación.

1. **Descripción de los datos**

Hacemos uso de las bases de datos proporcionadas por Airbnb para la ciudad de Ámsterdam el 05 de diciembre, 2021 (Datos disponibles en <http://insideairbnb.com/get-the-data/>).

La forma en que limpiamos estos datos para obtener las tablas que necesitamos para construir el modelo relacional propuesto se encuentra en la carpeta del repositorio que se denomina ‘*Limpieza de Datos/’*. Los datos limpiados que se terminaran cargando en PostgreSQL se encuentran en esta carpeta de [OneDrive](https://uredu-my.sharepoint.com/:f:/g/personal/juan_russy_urosario_edu_co/EivPmiYbTZpBn5b1ogjplFUBJkr31HfQ6evUPhgN75ZuHg?e=zjdaVo).

En total el modelo se compone de 7 tablas:

* City: que contiene el código y el nombre de las ciudades que tienen propiedades listadas en la base. En este caso solo hacemos tenemos datos para la ciudad de Ámsterdam.
* Neighbourhood: Que contiene el nombre y zip code de los barrios donde se encuentran las propiedades listadas y la ciudad a la que pertenecen.
* Host: Datos e identificador de los usuarios que tienen al menos una propiedad listada.
* Listings: Propiedades listadas en la plataforma para alquiler, contiene información básica como lo son el nombre, la descripción, ubicación, disponibilidad y precio.
* Users: Usuarios de la plataforma que demandan el alquiler de las propiedades. Tienen la opción de reservar una propiedad con disponibilidad y realizar comentarios sobre las propiedades.
* Comment: Tabla donde se guardan los comentarios hechos por un usuario sobre una propiedad listada.
* Reserve: Tabla donde se guarda la disponibilidad de las propiedades un año hacia adelante, además se guarda que usuario reservo la propiedad en una fecha específica.

Para más información sobre las entidades, relaciones y atributos ver el archivo [Descripción de las entidades, atributos y relaciones](https://uredu-my.sharepoint.com/:x:/g/personal/juan_russy_urosario_edu_co/EVYrjVE4qBdLuOQBYl8fXycBMvSBoCLWVWPbyvIrWE2zAg?e=S92JO1)

1. **Reglas de negocio**

El problema que estamos interesados en abordar es el monitoreo y análisis de las propiedades listadas en la aplicación. Nuestro principal objetivo es proporcionar a los usuarios y a los oferentes la información necesaria para que puedan negociar acuerdos de alquiler de propiedades. Para esto es necesario diseñar un sistema claro que permita entender las dinámicas de disponibilidad de alquiler y entender las principales características de las oferentes exitosos para mejorar el servicio.

1. La aplicación tiene presencia en varias barrios, que tienen un código postal, de varias ciudades.
2. Cada barrio tiene muchas propiedades listadas las cuales tienen un identificador único.
3. Cada propiedad tiene un Host con un identificador único, un host puede tener muchas propiedades listadas.
4. A la aplicación le interesa tener información suficiente sobre las propiedades listadas: ubicación (coordenadas, barrio, etc.), descripción, numero de comentarios, etc.
5. También le interesa conocer a detalle las fechas de disponibilidad de esas propiedades, además de sus precios.
6. Por regulación debe tener información sobre los Host: nombre, antigüedad, etc.
7. La aplicación sirve como un punto de conexión entre propietarios y huéspedes, por lo que también tiene una base de usuarios que se encuentran interesados en la información que la aplicación ofrece.
8. Los usuarios de la aplicación tienen un identificador único, además de tener la posibilidad de publicar comentarios sobre la calidad del servicio y la propiedad.
9. Una propiedad puede ser reservada por un usuario si tiene disponibilidad en la fecha de interés. Los precios no son fijos y pueden cambiar dependiendo de la temporada.
10. Si una propiedad tiene disponibilidad en una fecha se marca como si tuviera un usuario nulo.
11. **Escenarios de análisis**

* **Analizar el índice de ocupación de las propiedades listadas en una fecha y locación especifica.**
  + **Finalidad:** Prever el comportamiento de la demanda por propiedades listadas para brindar información oportuna a las partes involucradas. A los host sobre las fechas y locaciones con mayor cotización y a los usuarios sobre los precios disponibles.
* **Observar la desigualdad en la cantidad de propiedades listadas por los propietarios.**
  + **Finalidad:** Es importante que haya competencia en las ofertas de alquiler para conseguir la mayor eficiencia al momento de atender al usuario. Mirar la concentración o desigualdad en la posesión de estas propiedades podría ayudar a la empresa a decidir si poner un límite de precios y como regular el mercado para que sea más competitivo.
* **Analizar mediante redes de grafos que tan conectados se encuentran los host y los usuarios.**
  + **Finalidad:** El estudio de redes nos puede ayudar a entender si existen comunidades muy fuertes al interior de nuestra plataforma, si existen nodos que están ‘*sobrecargados*’ o tienen una importancia exagerada en las conexiones de la red. De esta manera la empresa puede mejorar las recomendaciones que realiza a los usuarios en su plataforma para conseguir que estos se relacionen con una mayor diversidad de participantes.
* **Analizar mediante redes de grafos que tanta interacción a través de comentarios tiene una propiedad con los usuarios.**
  + **Finalidad:** Podemos estudiar que tan importante son los comentarios al interior de nuestra plataforma relacionando la cantidad de interacciones que tiene una propiedad listada con su índice de ocupación o con su precio.

1. **Explicación script DDL**

El script para construir las tablas del modelo relacional se encuentra en ‘*Proyecto\_final/implementacion\_ddl.sql*’

1. **Explicación carga masiva de datos**

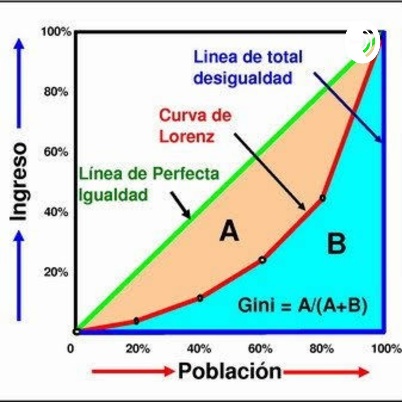
Descargar los archivos que se encuentran en el enlace de [OneDrive](https://uredu-my.sharepoint.com/:f:/g/personal/juan_russy_urosario_edu_co/EivPmiYbTZpBn5b1ogjplFUBJkr31HfQ6evUPhgN75ZuHg?e=zjdaVo) mencionado en la sección anterior.

Correr el archivo ‘*Proyecto\_final/carga\_masiva\_datos.sql’* cambiando la dirección correspondiente a los archivos que se descargaron desde OneDrive.

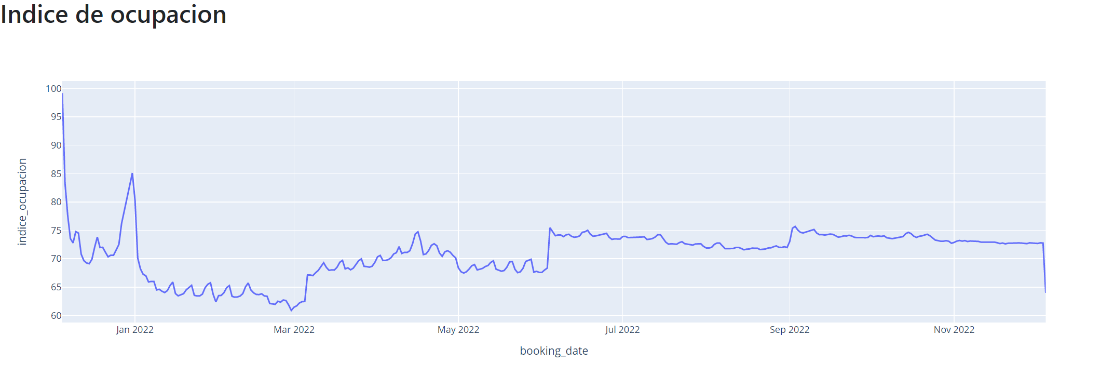
1. **Explicación verificación módulos de Python y Dash**

En la carpeta ‘*Proyecto\_final/Python y Dash/*’ se encuentran 3 archivos:

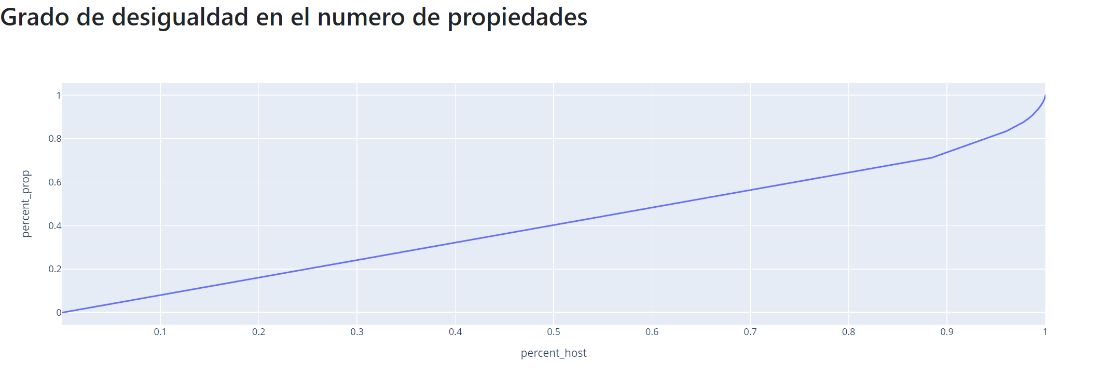
* El primero que se llama “*Connection.py*” realiza la conexión con la base de datos. Esta conexión esta echa para conectar con nuestro postgreSQL, modificar en caso de usar en otro dispositivo.
* El segundo que se llama “Funciones.py” se encarga de definir las consultas a la base de datos para crear los diagramas, estas son:
  + **Indice\_ocupacion:** Se encarga de mostrar el porcentaje de ocupación de todos los listings a lo largo de un periodo de tiempo
  + **Gini:** Es una medida de desigualdad que normalmente se utiliza para medir la desigualdad de ingresos en un país, pero que en nuestro caso, lo utilizamos para mostrar la desigualdad en la cantidad de propiedades listadas por un solo host. La imagen que diseñamos en Dash representa la Curva de Lorentz que se puede visualizar en la siguiente gráfica:



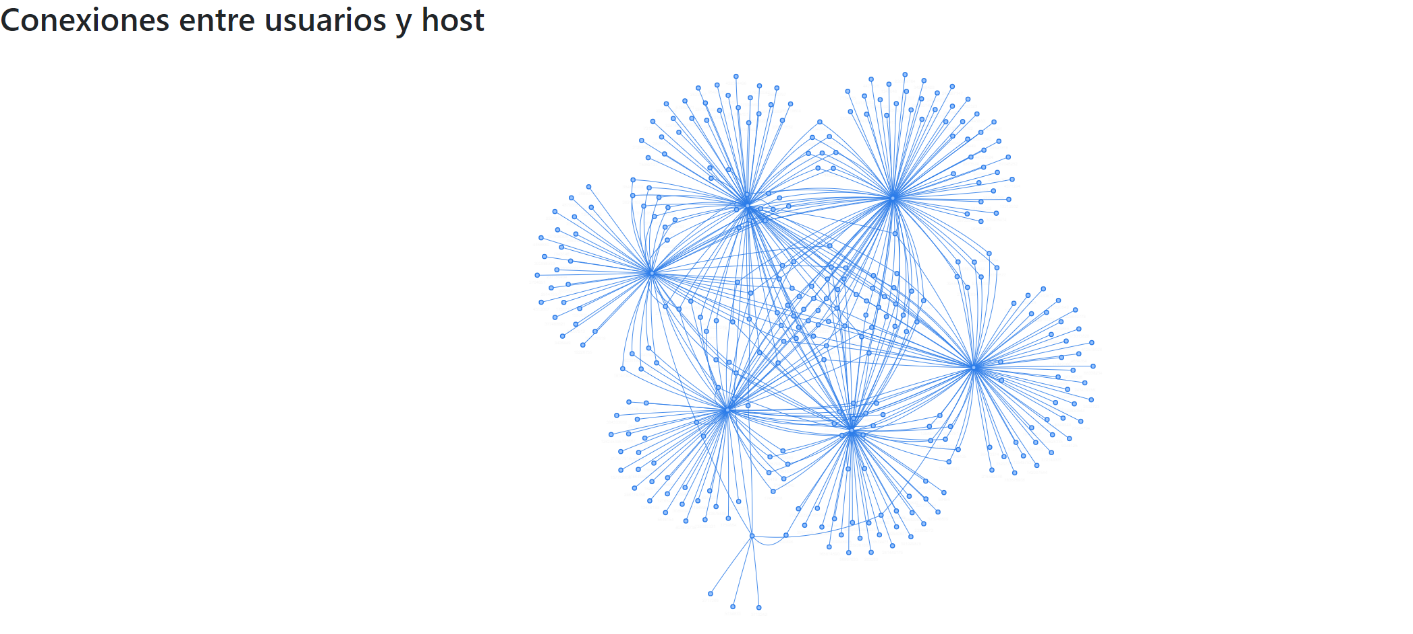
* + **Net:** Se encarga de mostrar todas las conexiones que hay entre los users y los host, mostrando si un user alguna vez con algún host
  + **Net2:** Se encarga de mostrar la relación que tienen las propiedades listadas con los usuarios a través de los comentarios que tienen en la plataforma.
* El tercero que se llama “airbnb.py” se encarga de llamar estas funciones, inicializar el dash en el localhost y crear los siguientes gráficos:
  + **Índice de ocupación:** Un gráfico lineal que compara el % de ocupación total en el tiempo



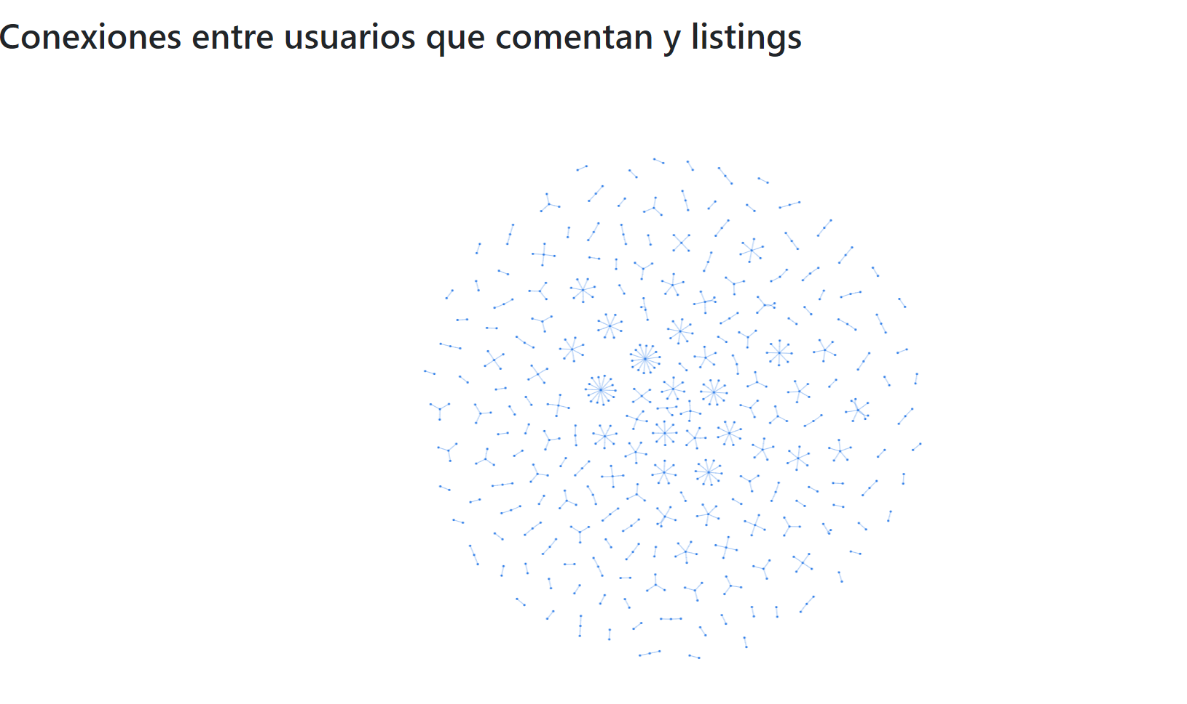
* **Grado de desigualdad en el número de propiedades:** Un gráfico lineal que representa la curva de un diagrama de Gini en el grado de desigualdad del número de propiedades de cada host



* **Conexiones entre usuarios y host:** Un gráfico de conexiones que hicimos con el módulo de “*visdcc*” que representa algunas de las conexiones de los usuarios que han reservado alguna vez con algún host.



* **Conexiones entre usuarios que comentan y listings:** Un gráfico de conexiones que también hicimos con el módulo de “visdcc” que representa algunas de las conexiones entre el id de los usuarios que han comentado en algún listing.

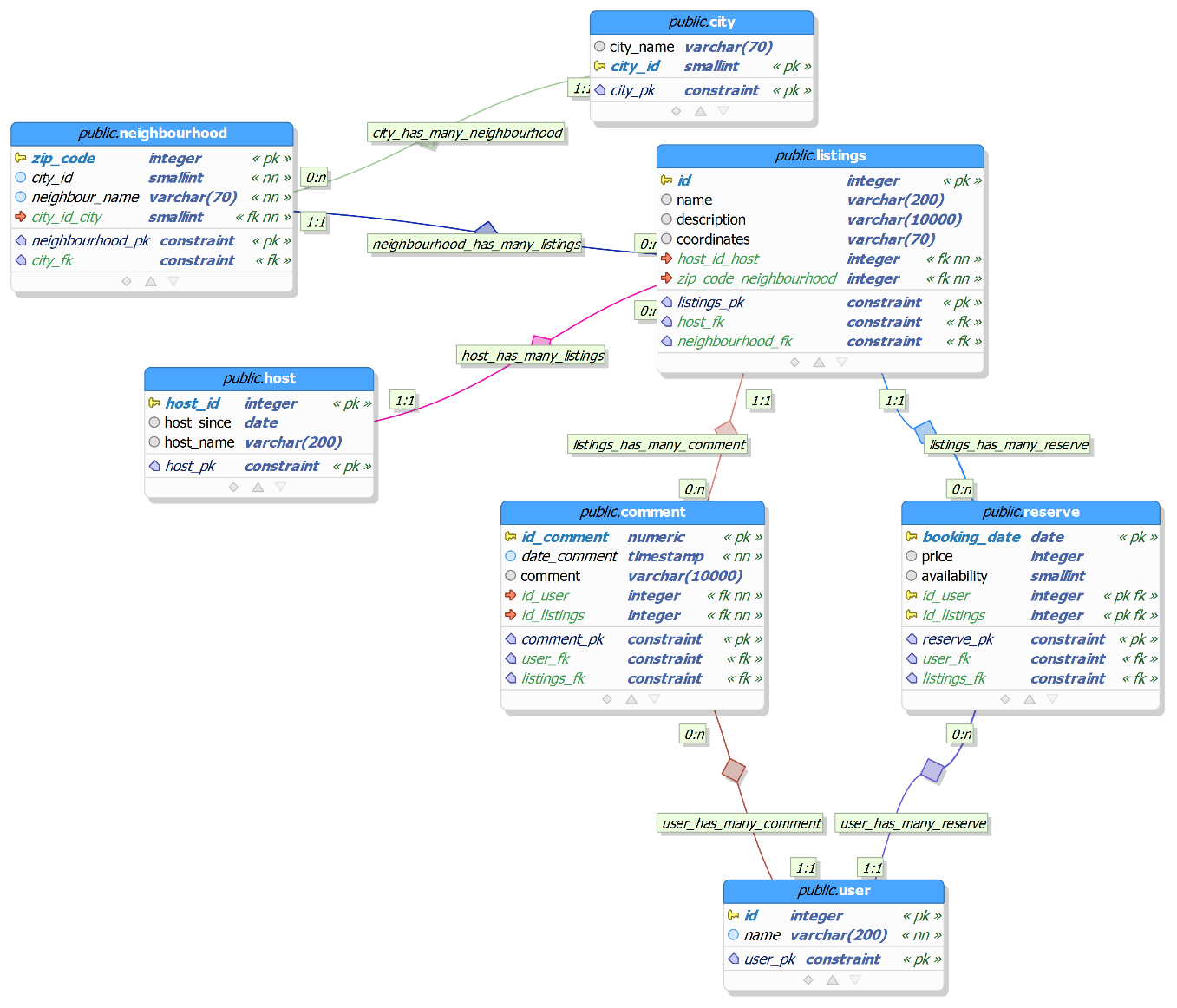


1. **Modelo Entidad Relación**

**Diagrama

Descripción generada automáticamente**

1. **Modelo Relacional en Tercera Forma Normal**



1. **Conclusión del proyecto**

Aunque el proyecto sin duda fue un proceso divertido y lleno de aprendizaje. También estuvo lleno de obstáculos y desafíos que tuvimos que sobrepasar. Los mayores ejemplos de esto fueron:

* En la carga de datos, ya que en el proyecto manejábamos gran variedad de datos, en grandes cantidades. Un error en el asignar correctamente un tipo nos costaba bastante tiempo para volver a cargar los modelos. No contamos con que habían id’s muy largos que necesitaban un *bigint* o que los comentarios tenían más de *500 caracteres*
* En la limpieza de datos ya que en muchos casos se repetían o estaban en formatos diferentes a los que necesitábamos en nuestro modelo relacional. Además de que creamos relaciones que antes no existían como por ejemplo los que se encuentran en la carpeta reservas.
* La creación de las gráficas, ya que quisimos hacer algo diferente y queríamos algo que destacara, decidimos hacer una curva de Lorentz para visualizar el índice de Gini y 2 diagramas de Network. El diagrama de Gini fue complicado ya que la función no solo debía traer los datos si no organizar los host de menor a mayor cantidad de propiedades y transformarlos en porcentajes comparables. Los diagramas de Network fue aún más complicado ya que teníamos que usar un nuevo módulo con el que no teníamos nada de experiencia (*visdcc*) para crear las conexiones en los diagramas.
* Sin importar de estos obstáculos, el proyecto se pudo realizar y corregir en su totalidad, brindando como resultado 4 escenarios de análisis muy interesantes que nos permiten entender de forma mucho más sencilla como funciona la plataforma de Airbnb y que clase de datos y procesos tienen.

1. **Otros:**

**Imágenes de prueba que las consultas en Python funcionan correctamente**

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

**Imagines de prueba de que la página en Dash funciono correctamente**



