# Reviews and Satisfaction Analysis of AirBnB Brazil and Mexico from June 2010 to February 2021

## Final Submission ofData Analytics

**Comition 49100**

José Chirif

March 2024

## Table of contents

# I. Introduction

## 1. Introduction

Airbnb is an online hotel platform, where customers can book, pay and leave reservations on its online platform. Also, different owners can make their properties available for the stay service.

## 2. Table of versions:

| Review | Submission date |
|---|---|
| 0 | March 10, 2024 |

# II. Dataset y generalidades del proyecto

## 3. Original dataset

For this project, the Airbnb Dataset was selected, which can be found at Original_dataset_link. The dataset contains Airbnb data from Rio de Janeiro (Brazil) and Mexico City (Mexico), covering the period from June 6, 2010, to February 25, 2021. These two cities and the mentioned timeframe will be the focus of the project.

The dataset consists of five tables: Host, Properties, Reviews, Cities, and Countries.

The original dataset is in the following link: https://drive.google.com/drive/folders/1ViJ-dgxlTnecN4JodyvAf2DI6z3HtuJ6

## 4. Scope of the project

The project will cover the analysis of Airbnb in Rio de Janerio (Brazil) and Mexico city (Mexico City) conducted between June 6, 2010 and February 25, 2021.

The project only covers analysis of existing data. Projections or other methods are outside the scope of the project.

## 5. Hypothesis / Objectives

For the present work, 2 hypotheses are proposed:

### 5.1. Hypothesis 1

Between December and February (holidays and summer vacations) there are more reviews due to people's availability.

### 5.2. Hypothesis 2

Percentagewise, higher cost properties have better average customer ratings.

Our objective will be to verify if both Hypotheses are fulfilled.

## 6. Tools

The following software has been used throughout the project:

- Microsoft Excel
- SQL server
- Power BI

# III. Cleaning of original Dataset (Ms. Excel) and relations:

## 7. Original dataset cleaning

### 7.1. Required columns

Based on our Hypothesis / Objectives, we will eliminate the columns that we do not consider necessary.

Therefore, the tables will have the following changes:

*Host:*

| Name | Keep/Delete |
|---|---|
| **host_id** | Keep |
| **host_since** | Keep |
| **host_location** | Delete (the same information is allowed in the properties table) |
| **name** | Keep |
| **host_is_superhost** | Delete (this information will not be used) |
| **host_response_time** | Delete (this information will not be used) |
| **host_response_rate** | Keep |
| **host_identity:verified** | Delete (this information will not be used) |
| **host_total_listings_count** | Delete (this information will not be used) |

*Properties:*

| Name | Keep/Delete |
|---|---|
| **listing_id** | Keep |
| **host_id** | Keep |
| **city** | Keep |
| **apartment_type** | Delete (this information will not be used) |
| **room_type** | Delete (this information will not be used) |
| **bedrooms** | Delete (this information will not be used) |
| **acomodates** | Delete (this information will not be used) |
| **neighborhood** | Delete (this information will not be used) |
| **price** | Keep |

*Reviews:*

| Name | Keep<br>/Delete |
|---|---|
| **date** | Keep |
| **review_id** | Keep |
| **reviewer_id** | Keep |
| **listing_id** | Keep |
| **review_scores_rating** | Keep |
| **review_scores_accuracy** | Keep |
| **review_scores_cleanliness** | Keep |
| **review_scores_value** | Keep |
| **review_scores_communication** | Keep |
| **review_scores_location** | Keep |

*Cities:*

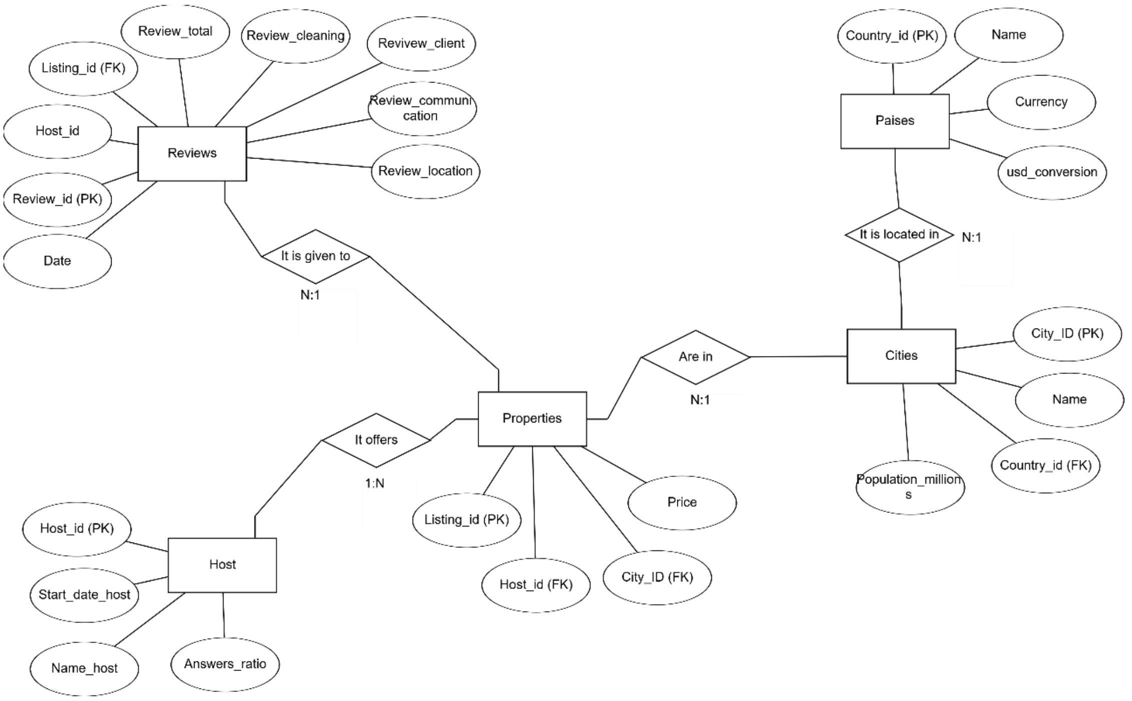
| Name | Keep/Delete |
|---|---|
| **id_city** | Keep |
| **name** | Keep |
| **id_pais** | Keep |
| **population_millons** | Keep |

*Countries:*

| Name | Keep/Delete |
|---|---|
| **id_country** | Keep |
| **name** | Keep |
| **currency** | Keep |
| **conversion_usd** | Keep |
| **population_millons** | Delete (the same information is allowed in the citys table) |

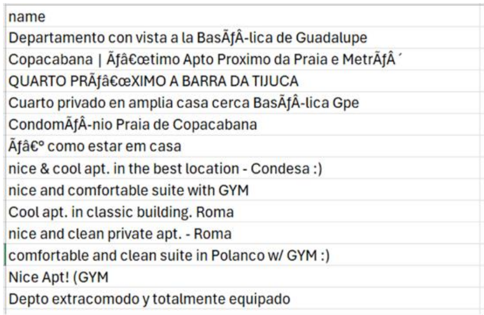
### 7.2. Tables' relation (EDR diagram)

In order to verify that the tables comply with the necessary fields to relate them, the following EDR diagram has been done.

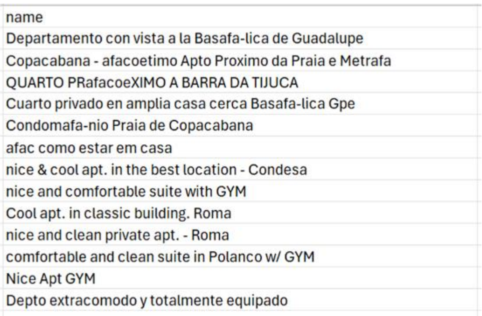


### 7.3. Special characters and other considerations

To ensure the correct loading of the data, we first standardized the characters of the texts. Replacing the characters that do not belong to English or Spanish with the equivalent of our language.



| name |
|---|
| Departamento con vista a la BasÃƒÂ-lica de Guadalupe |
| Copacabana \| Ãƒâ€œtimo Apto Proximo da Praia e MetrÃƒÂ´ |
| QUARTO PRÃƒâ€œXIMO A BARRA DA TIJUCA |
| Cuarto privado en amplia casa cerca BasÃƒÂ-lica Gpe |
| CondomÃƒÂ-nio Praia de Copacabana |
| Ãƒâ€° como estar em casa |
| nice & cool apt. in the best location - Condesa :) |
| nice and comfortable suite with GYM |
| Cool apt. in classic building. Roma |
| nice and clean private apt. - Roma |
| comfortable and clean suite in Polanco w/ GYM :) |
| Nice Apt! (GYM |
| Depto extracomodo y totalmente equipado |



| name |
|---|
| Departamento con vista a la Basafa-lica de Guadalupe |
| Copacabana - afacoetimo Apto Proximo da Praia e Metrafa |
| QUARTO PRafacoeXIMO A BARRA DA TIJUCA |
| Cuarto privado en amplia casa cerca Basafa-lica Gpe |
| Condomafa-nio Praia de Copacabana |
| afac como estar em casa |
| nice & cool apt. in the best location - Condesa |
| nice and comfortable suite with GYM |
| Cool apt. in classic building. Roma |
| nice and clean private apt. - Roma |
| comfortable and clean suite in Polanco w/ GYM |
| Nice Apt GYM |
| Depto extracomodo y totalmente equipado |

Also, every host has been ensured that has a unique ID.

### 7.4. Final tables

Finally, the tables have the following structures:

*Host*

| Name | Data type | Keys |
|---|---|---|
| **Host_id** | Identity | PK |

| Name | Data type | Keys |
|---|---|---|
| **Host_start_date** | DATE | |
| **Name_host** | CHAR(260) | |
| **Host_answers_ratio** | FLOAT | |

*Propiedades*

| Name | Data type | Keys |
|---|---|---|
| **Listing_id** | Identity | PK |
| **Host_id** | INT | FK: Host.Host_id |
| **City_ID** | INT | FK: Cities: City_ID |
| **Price** | INT | |

*Reviews*

| Name | Data type | Keys |
|---|---|---|
| **Date** | **DATE** | |
| **Review_id** | Identity | PK |
| **Guest_id** | INT | |
| **Listing_id** | INT | FK: Properties: listing_id |
| **Total_review** | INT | |
| **Descriptions_review** | INT | |
| **Cleaning_review** | INT | |
| **Client_review** | INT | |
| **Comunication_review** | INT | |
| **Location_review** | INT | |

*Ciudades*

| Name | Data type | Keys |
|---|---|---|
| **ID_ciudad** | Identity | PK |
| **Name** | CHAR(30) | |
| **Country_id** | INT | FK: Countries: Country_id |
| **Population_millions** | FLOAT | |

*Países*

| Name | Data type | Keys |
|---|---|---|
| **Country_ID** | Identity | PK |
| **Name** | CHAR(30) | |
| **Currency** | CHAR(3) | |
| **USD_Convertion** | FLOAT | |

The clean Dataset is in a folder within the same project's repository.

# IV. SQL

A database was created in SQL with the same 5 tables. Then, the data load was performed and it was ensured that the data types are correct with no duplicate IDs.

## 8. SQL Data load

### 8.1. Database creation

The database was created with the following Query:

```
CREATE DATABASE AnalisisAIRBNB;
```

### 8.2. Tables creation

Five tables has been created within same structure as the csv files "7.4 Final tables".

The queries are:

```
USE AnalisisAIRBNB;

--Host table:
CREATE TABLE Host(
Host_ID INT PRIMARY KEY NOT NULL,
Fecha_inicio_host DATE,
Ubicacion_del_host CHAR(50),
Nombre_host CHAR(30),
Ratio_respuestas_host FLOAT
);
ALTER TABLE Host
DROP COLUMN Ubicacion_del_host;

ALTER TABLE Host
ALTER COLUMN Nombre_host CHAR(260);
```

```
--Reviews table:
CREATE TABLE Reviews(
Fecha DATE,
Review_id INT PRIMARY KEY NOT NULL,
Huesped_id INT NOT NULL,
Listing_id INT NOT NULL,
Review_puntaje_total INT,
Review_puntaje_exactituddescripcion INT,
Review_puntaje_limpieza INT,
Review_puntaje_valorcliente INT,
Review_puntaje_comunicacion INT,
Review_puntaje_ubicacion INT
);



--Cities table:
CREATE TABLE Ciudades(
ID_ciudad INT PRIMARY KEY NOT NULL,
Nombre CHAR(30) NOT NULL,
ID_Pais INT NOT NULL,
Poblacion_millones FLOAT
);


--Countries table:
CREATE TABLE Paises(
ID_Pais INT PRIMARY KEY NOT NULL,
Nombre CHAR(30) NOT NULL,
Moneda CHAR(3) NOT NULL,
Conversion_usd FLOAT
);
```

### 8.3. Data loading to database

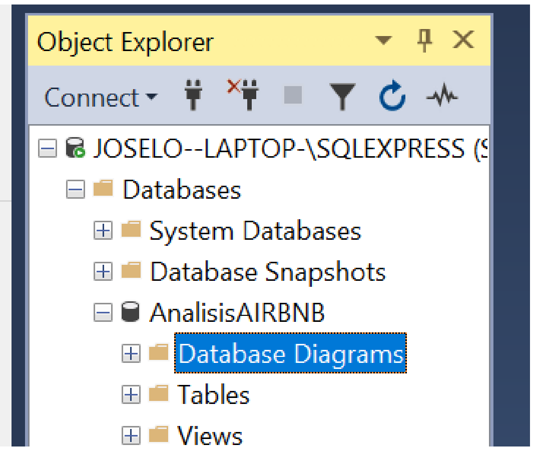
From the csv files, queries were created with Excel to load the tables. We made sure that all IDs are unique values and dates were replaced by empty cells.

The queries and SQL database are in a folder within the same project's repository.

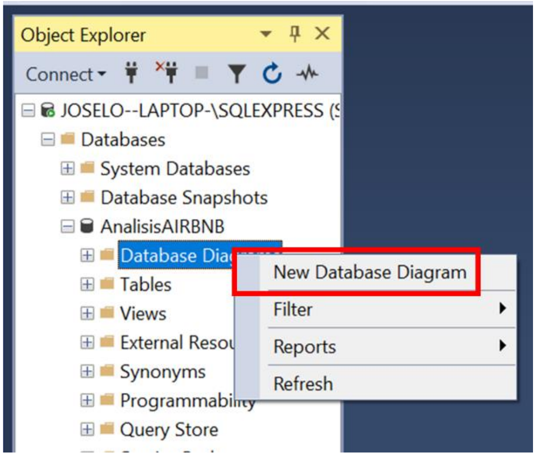
## 9. SQL tables relation

According to the Tables' relation (EDR diagram) detailed in chapter 7.2 above the tables created in SQL were linked

a. Click on "Object Explorer" -> "Databases" -> "analisisAIRBNB".

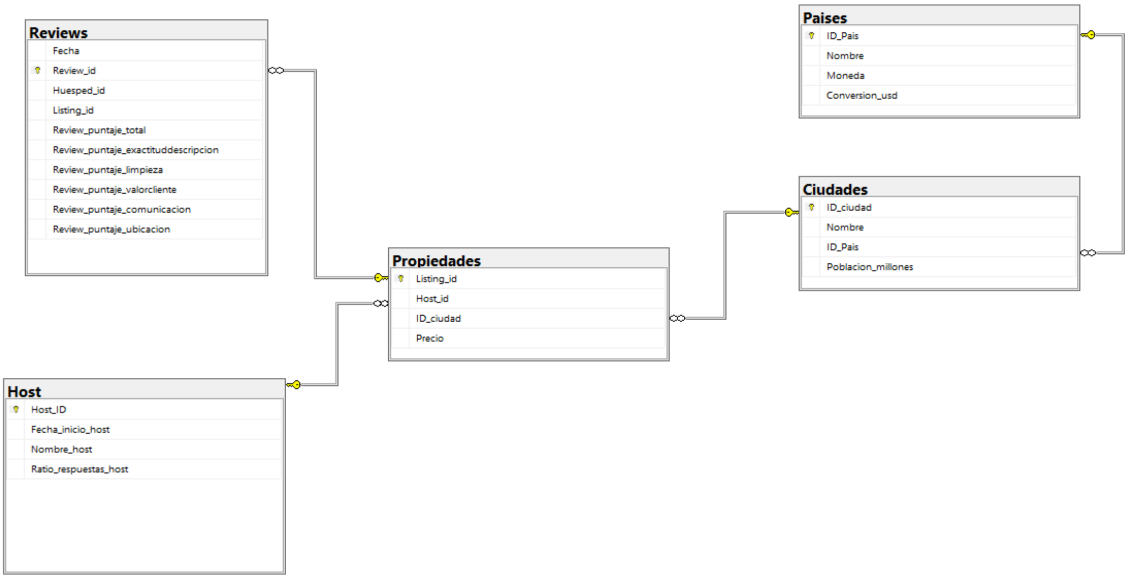


b. Right click on “Databases Diagrams” and select “New Database Diagram”



c. Selecta all the tables and click on “Add”.

d. In each table select the PK and drag to the FK of the other tables, leaving the relations in SQL as shown in the following figure:

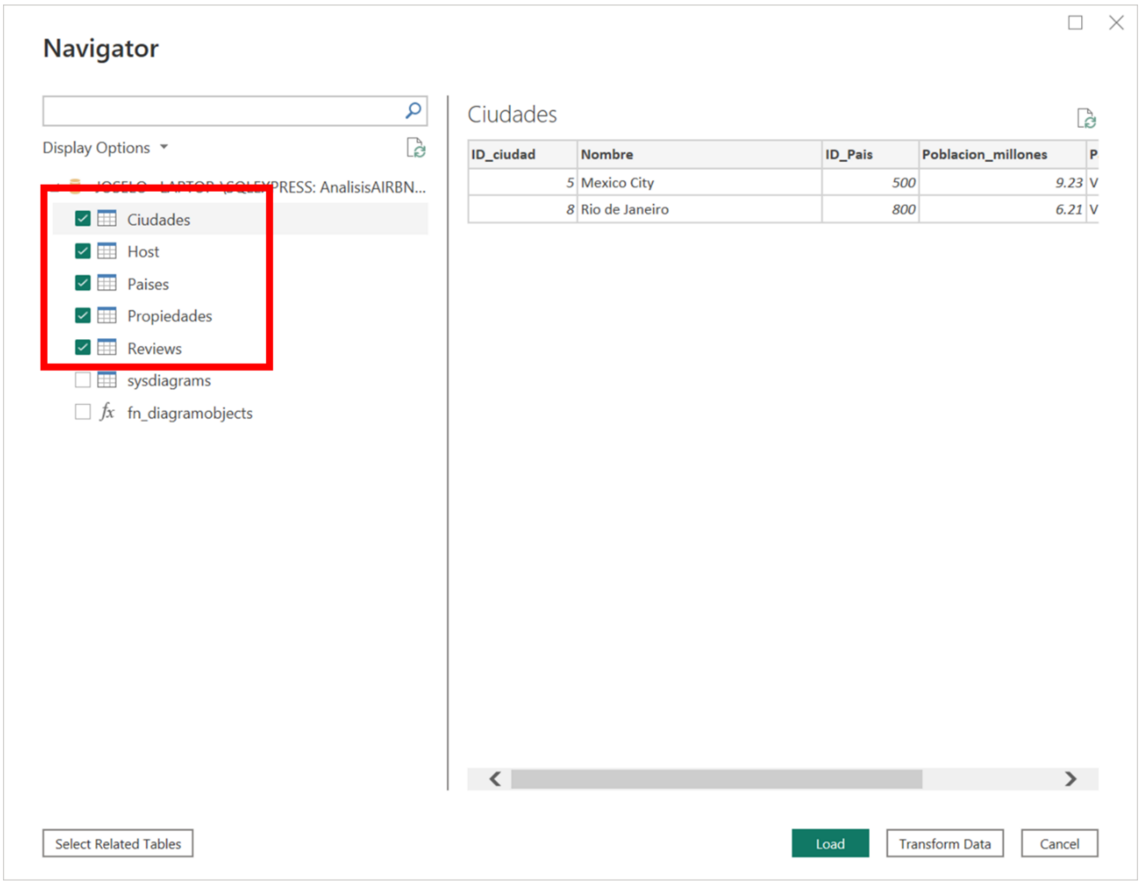


# V. Data analysis (Power BI)

## 10. Data loading in Power BI

a. The load is performed from SQL Server (with SQL Management credentials).

b. The tables are selected and loaded. Diagrams will be import and new relations and tables will be made.



JOSELO-LAPTOP\SQLEXPRESS: AnalisisAIRBN...

- ☑ Ciudades
- ☑ Host
- ☑ Paises
- ☑ Propiedades
- ☑ Reviews
- □ sysdiagrams
- □ fn_diagramobjects

Ciudades

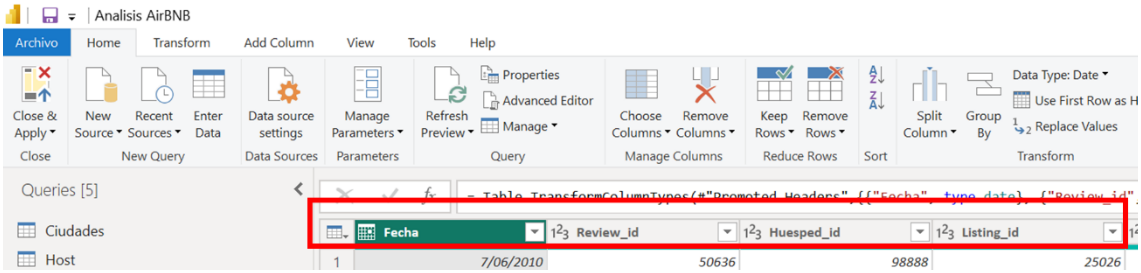
| ID_ciudad | Nombre | ID_Pais | Poblacion_millones | P |
|---|---|---|---|---|
| 5 | Mexico City | 500 | 9.23 | V |
| 8 | Rio de Janeiro | 800 | 6.21 | V |

## 11. ETL Process

### 11.1. ETL in Power Query

a. Open Power Query.

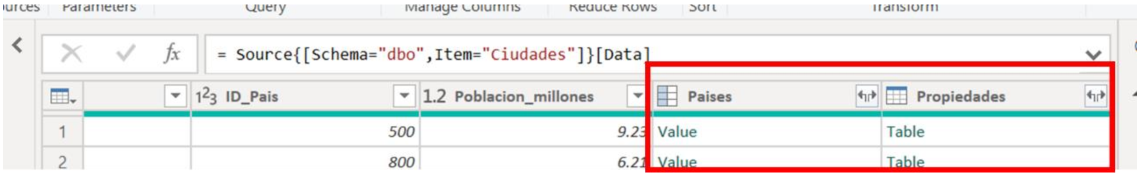
b. Check that the data types are correct (in all tables).



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"Fecha", type date}, {"Review_id"
```

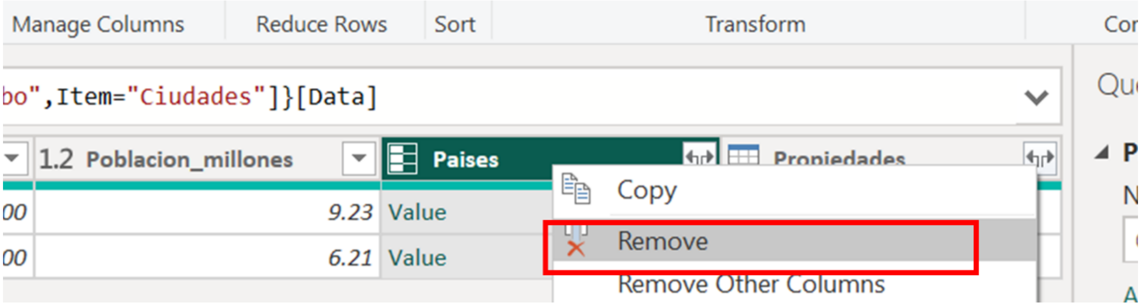
|  | Fecha | Review_id | Huesped_id | Listing_id |
|---|---|---|---|---|
| 1 | 7/06/2010 | 50636 | 98888 | 25026 |

c. Related tables in SQL are compressed at the end of each table (similar to merging tables). We will proceed to remove those columns.

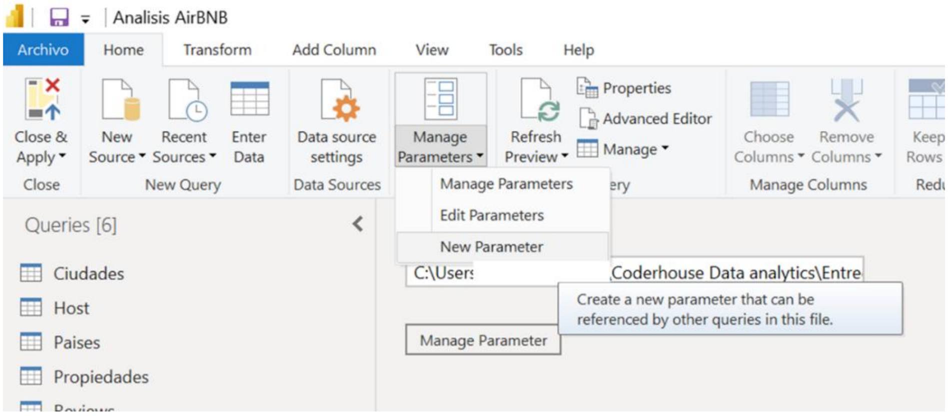


```
= Source{[Schema="dbo",Item="Ciudades"]}[Data]
```

| | ID_Pais | Poblacion_millones | Paises | Propiedades |
|---|---|---|---|---|
| 1 | 500 | 9.23 | Value | Table |
| 2 | 800 | 6.21 | Value | Table |

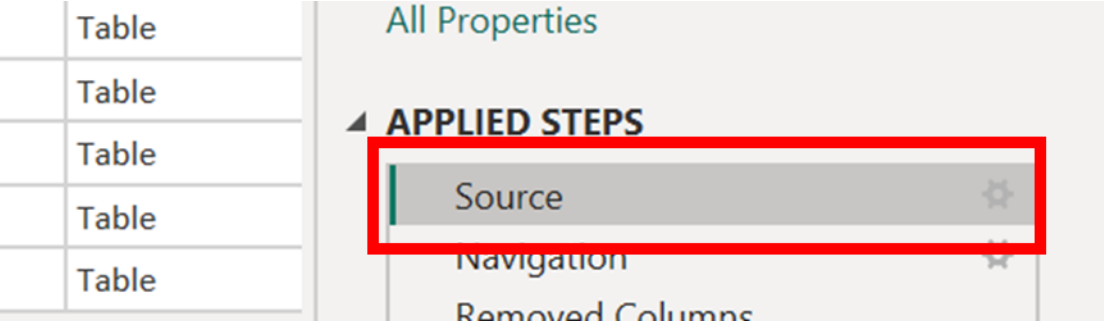


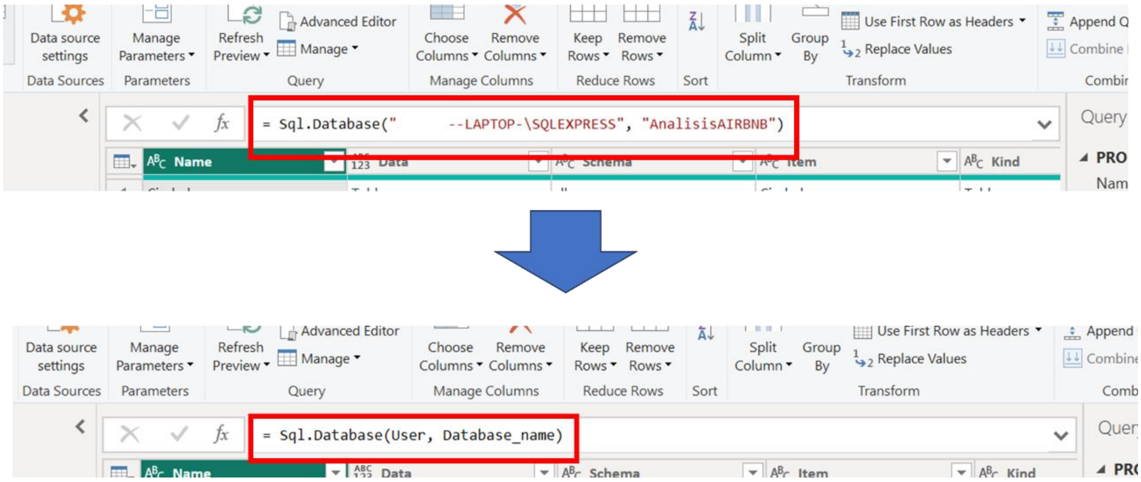
d. The parameter "user" and "database_name" where we will put the SQL credential and the database Name respectively will be created.



Create a new parameter that can be referenced by other queries in this file.

e. In each table, we go to the first step (Source) and modify the credentials with the created parameters.





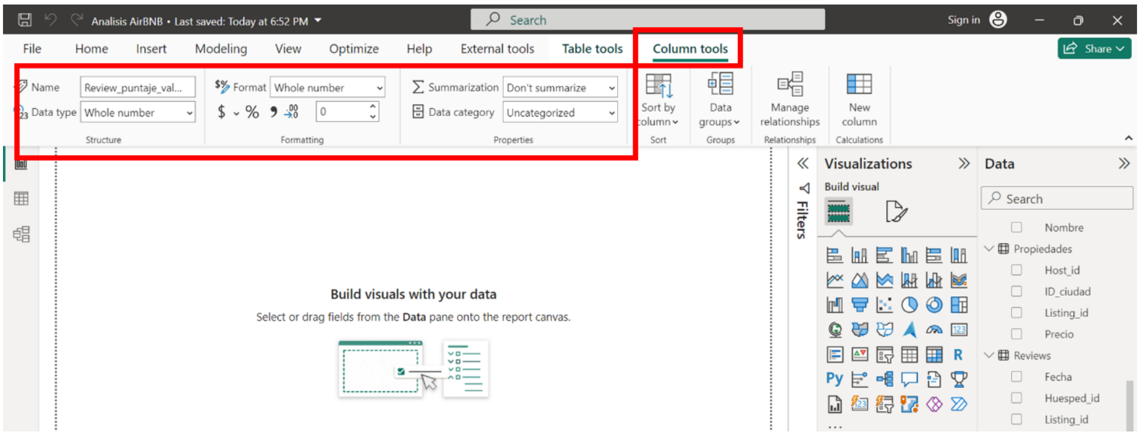
```
= Sql.Database("        --LAPTOP-\SQLEXPRESS", "AnalisisAIRBNB")
```

```
= Sql.Database(User, Database_name)
```

f. If the SQL credential is modified (or the PC is changed) only the "user" parameter must be modified with the new credential. In the same way, if the database Name is modified, only the new Name must be entered to the "Database_name" parameter.

### 11.2. ETL in Power BI

g. After ETL process in Power Query has been done, all fields in the Power BI dashboard should be checked.



Analisis AirBNB • Last saved: Today at 6:52 PM

Name: Review_puntaje_val...
Data type: Whole number
Format: Whole number
Summarization: Don't summarize
Data category: Uncategorized

Build visuals with your data

Select or drag fields from the Data pane onto the report canvas.

## 12. Power BI Dashboard template

While the dashboard template was being made, the testing of the hypotheses has been prioritized. Other elements of interest that arise during the analysis will then be covered.

The Hypothesis has been made in chapter 5 Hypothesis / Objectives.

Hypothesis recapt:

a. "Between December and February (holidays and summer vacations) there are more reviews due to people's availability."

b. "Percentagewise, higher cost properties have better average customer ratings."

So, in order to test the hypothesis, the reviews (including quantities per month), and relate price to customer satisfaction must been analyzed.

Finally, the report Will divide in:

a. Home page.

b. Reviews analysis.

c. Satisfaction analysis.

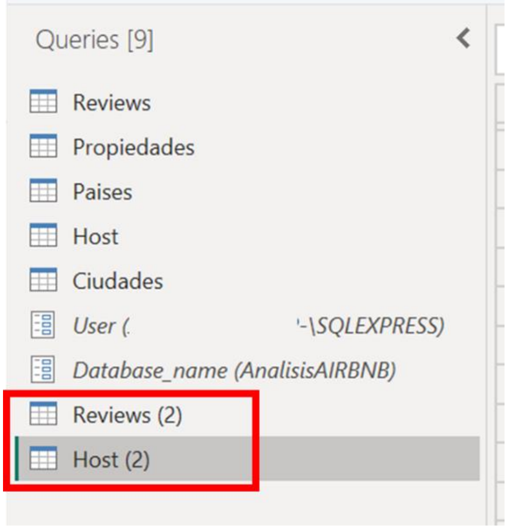
d. Price analysis.

# 13. Calendar table

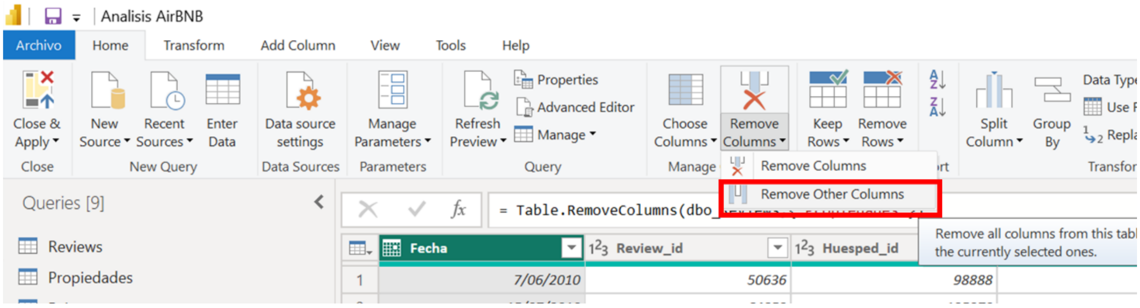
## 13.1. Calendar table from Power Query

“Reviews” and “Host” are the only tables with dates.

a. First duplicate the tables that have dates.



b. On the duplicated tables, removes other columns (that aren´t dates).

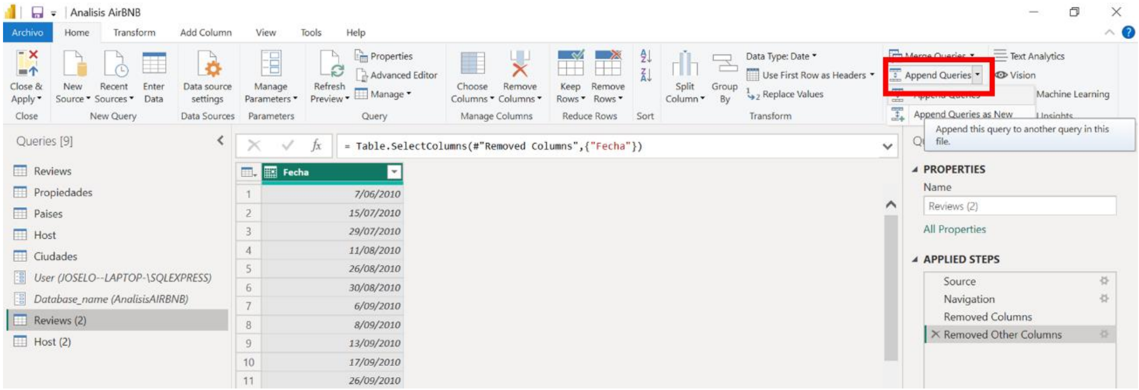


```
= Table.RemoveColumns(dbo_
```

Remove all columns from this tabl the currently selected ones.

| | Fecha | Review_id | Huesped_id |
|---|---|---|---|
| 1 | 7/06/2010 | 50636 | 98888 |

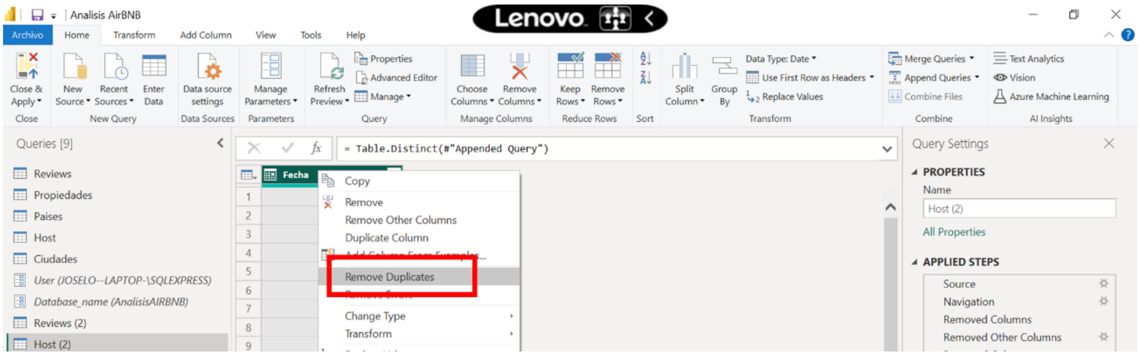
c. Append both tables (the date title must be the same).



```
= Table.SelectColumns(#"Removed Columns",{"Fecha"})
```

| | Fecha |
|---|---|
| 1 | 7/06/2010 |
| 2 | 15/07/2010 |
| 3 | 29/07/2010 |
| 4 | 11/08/2010 |
| 5 | 26/08/2010 |
| 6 | 30/08/2010 |
| 7 | 6/09/2010 |
| 8 | 8/09/2010 |
| 9 | 13/09/2010 |
| 10 | 17/09/2010 |
| 11 | 26/09/2010 |

d. Remove duplicated dates.



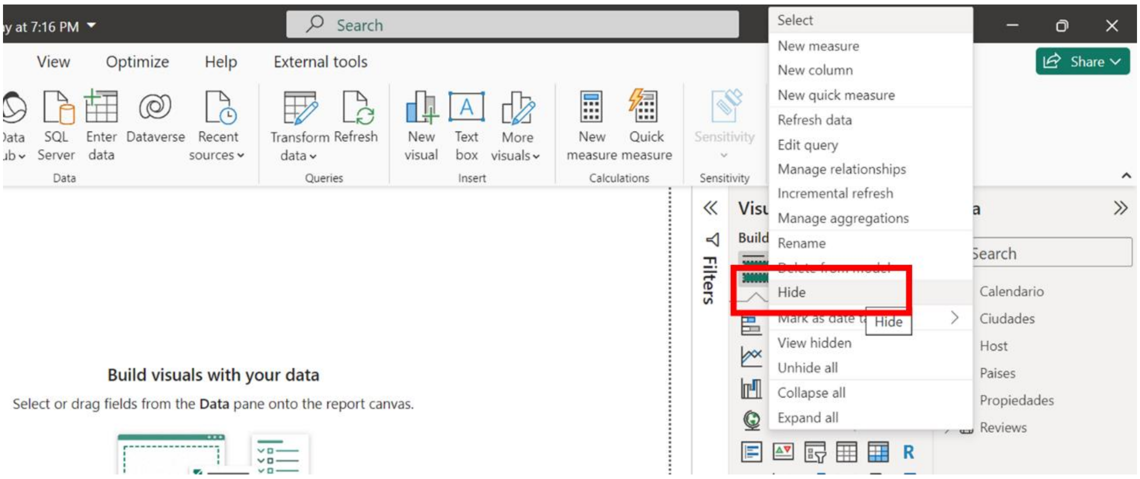
```
= Table.Distinct(#"Appended Query")
```

e. Order the dates.

f. Rename the table as Calendar.

## 13.2. Calendar (on Power BI)

g. First remove the duplicated tables to avoid mistakes.



h. Finally, add the following DAX columns on the Calendar Table:

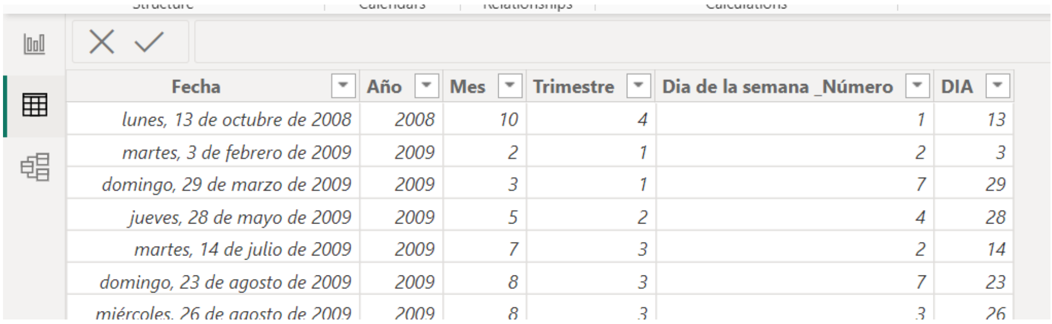
```
Year = YEAR(Calendar[Date])
```

```
Month = MONTH(Calendar[Date])
```

```
Quarter = QUARTER(Calendar[Date])
```

```
Day_number = WEEKDAY(Calendar[Date],2
```

```
Day = DAY(Calendar[Date])
```

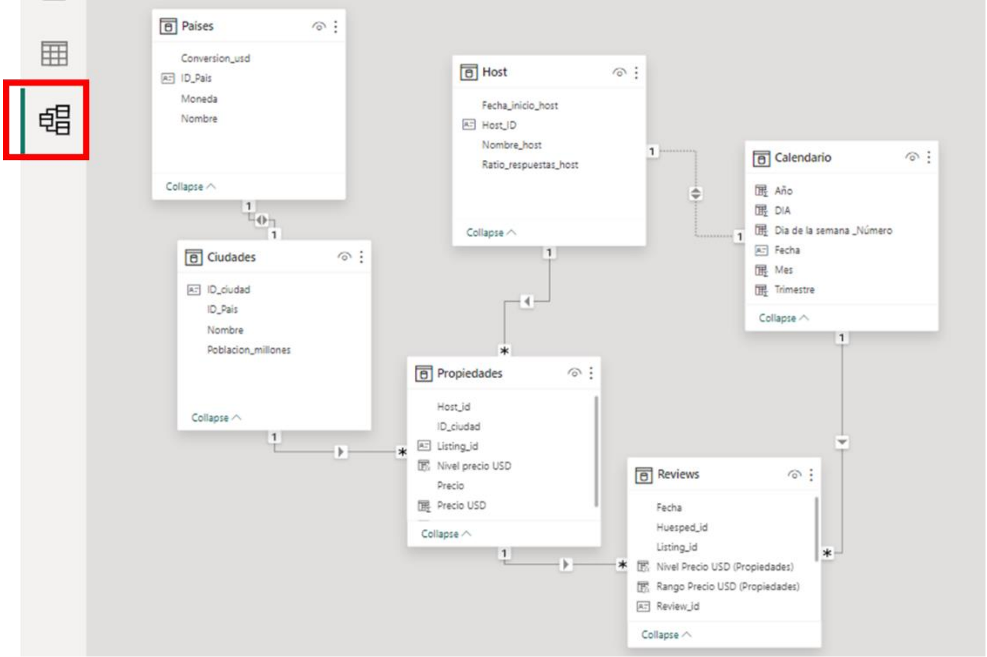


| Fecha | Año | Mes | Trimestre | Dia de la semana _Número | DIA |
|---|---|---|---|---|---|
| lunes, 13 de octubre de 2008 | 2008 | 10 | 4 | 1 | 13 |
| martes, 3 de febrero de 2009 | 2009 | 2 | 1 | 2 | 3 |
| domingo, 29 de marzo de 2009 | 2009 | 3 | 1 | 7 | 29 |
| jueves, 28 de mayo de 2009 | 2009 | 5 | 2 | 4 | 28 |
| martes, 14 de julio de 2009 | 2009 | 7 | 3 | 2 | 14 |
| domingo, 23 de agosto de 2009 | 2009 | 8 | 3 | 7 | 23 |
| miércoles, 26 de agosto de 2009 | 2009 | 8 | 3 | 3 | 26 |

## 14. Relations and connections tables

In the table relations view, the PKs are dragged to the FKs of the other tables, respecting the EDR detailed in the chapter “7.2 Tables’ relation (EDR diagram)”.

The calendar table created for the analysis should be added to this relationship, as follows:

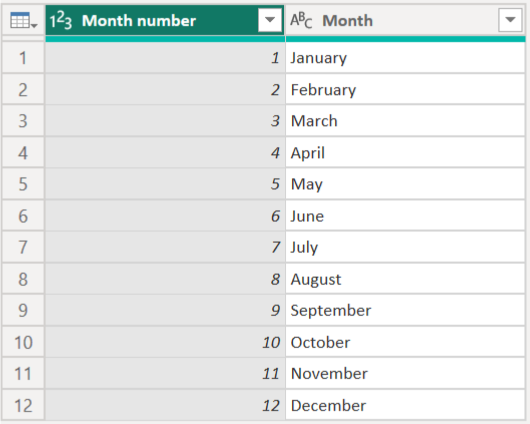


**Note:** The relation between Calendar and Host tables is as an inactive relation. The main one is Calendar to Reviews, since it is one of the objectives to be analyzed.

More complementary tables will be created, which will also be shown in the diagram.

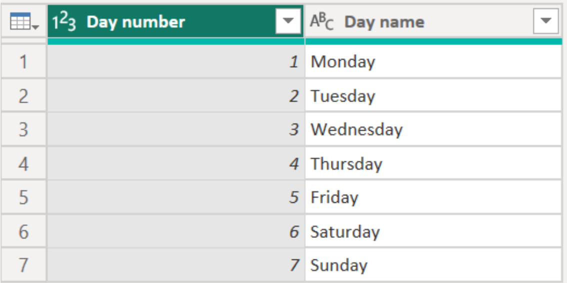
## 14.1. Supplementary Calendar Tables

On Power Query, create the following tables:



| | Month number | Month |
|---|---|---|
| 1 | 1 | January |
| 2 | 2 | February |
| 3 | 3 | March |
| 4 | 4 | April |
| 5 | 5 | May |
| 6 | 6 | June |
| 7 | 7 | July |
| 8 | 8 | August |
| 9 | 9 | September |
| 10 | 10 | October |
| 11 | 11 | November |
| 12 | 12 | December |

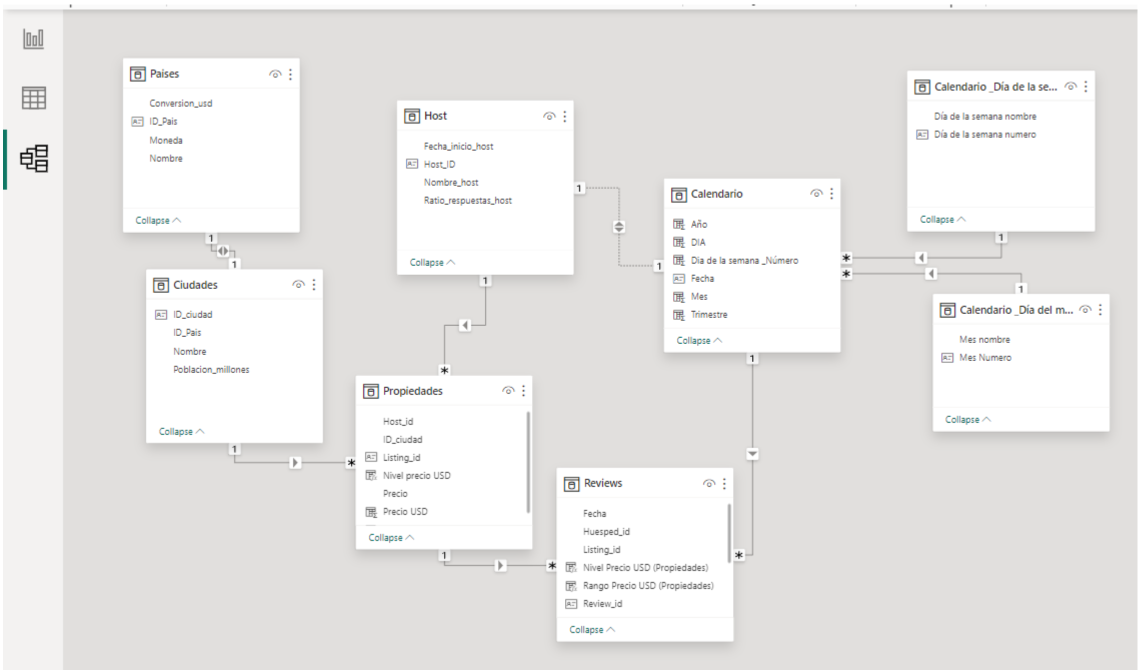
*Calendar _month name*



| | Day number | Day name |
|---|---|---|
| 1 | 1 | Monday |
| 2 | 2 | Tuesday |
| 3 | 3 | Wednesday |
| 4 | 4 | Thursday |
| 5 | 5 | Friday |
| 6 | 6 | Saturday |
| 7 | 7 | Sunday |

*Calendar _Day Name*

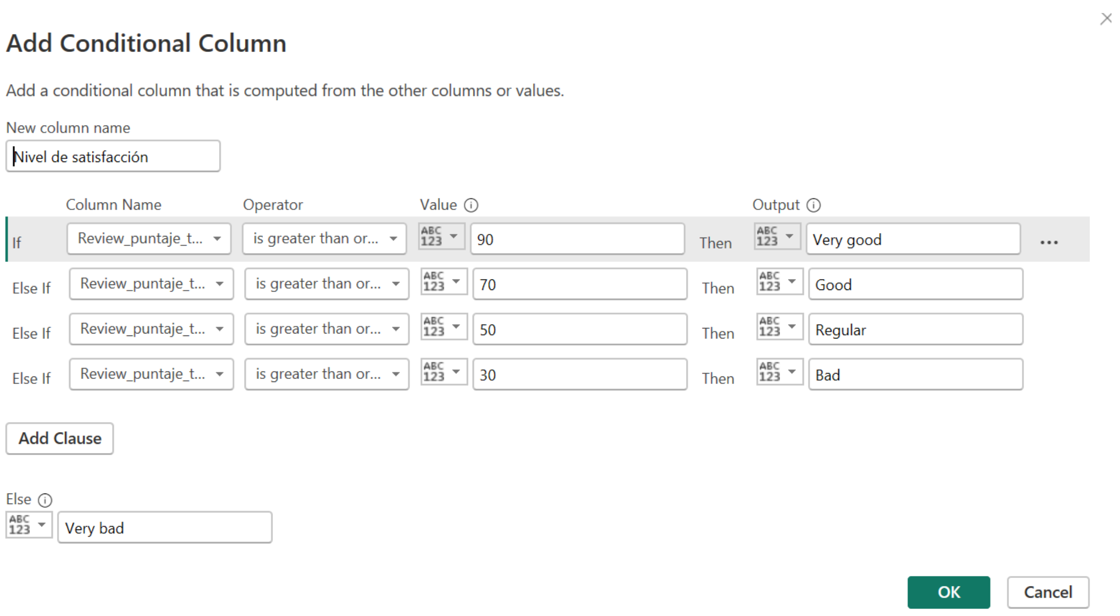
These tables are related to Calendar (month number and day of the week number) to provide more detail.



## 14.2. Review Score to Categories

*Total scores categories*

The total score goes from 0 to 100. Then, on Power Query, duplicate the reviews table, delete all columns except total score, and create the following conditional column:



Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name

Nivel de satisfacción

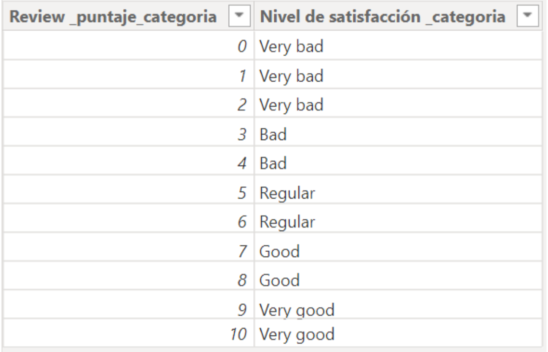
| | Column Name | Operator | Value | | | Output |
|---|---|---|---|---|---|---|
| If | Review_puntaje_t... | is greater than or... | ABC 123 | 90 | Then | ABC 123 Very good |
| Else If | Review_puntaje_t... | is greater than or... | ABC 123 | 70 | Then | ABC 123 Good |
| Else If | Review_puntaje_t... | is greater than or... | ABC 123 | 50 | Then | ABC 123 Regular |
| Else If | Review_puntaje_t... | is greater than or... | ABC 123 | 30 | Then | ABC 123 Bad |

Else

ABC 123 Very bad

### *Other categorie's scores*

All other scores range are from 0 to 10. Then following the satisfaction levels created in total score, a table will be created in Power Query with the same categories as follows:

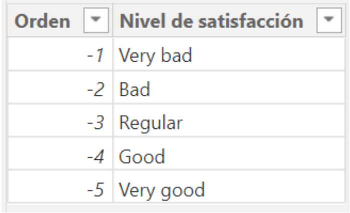


| Review _puntaje_categoria | Nivel de satisfacción _categoria |
|---|---|
| 0 | Very bad |
| 1 | Very bad |
| 2 | Very bad |
| 3 | Bad |
| 4 | Bad |
| 5 | Regular |
| 6 | Regular |
| 7 | Good |
| 8 | Good |
| 9 | Very good |
| 10 | Very good |

If the score is multiplied by 10, it will follow the same Total Score category.

### *Ordering categories by scores*

The following table has been created so that the scoring categories have a logical structure:



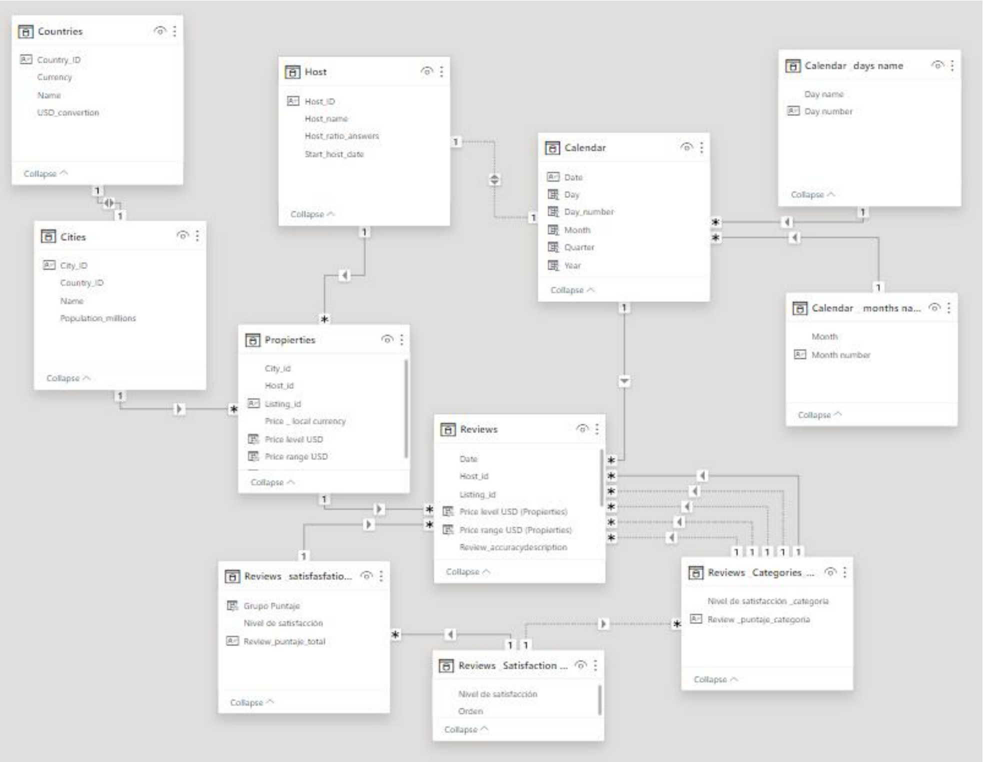
| Orden | Nivel de satisfacción |
|---|---|
| -1 | Very bad |
| -2 | Bad |
| -3 | Regular |
| -4 | Good |
| -5 | Very good |

The Satisfaction Category  is detailed in chapter 17.2.

Finally, the tables "Reviews _Score_categories" and "Reviews _total satisfaction" has been connected. All Primary keys are "Satisfaction level".

### 14.3. Complementary tables ralations

Finally, all relations are as follows:



## 15. DAX new columns

### 15.1. Propierties table

*Price USD*

Standar all prices to USD currency:

a. Price: With the relations copy the conversion value from the table Countries to Properties and multiply it by the Local Price.

b. Condition: If the Price is greater than 40000, it is considered 0. In this way we eliminate the anomalous data that do not allow a good analysis.

c. Blanks: Convert the values equal to zero into blank cells. In this way statistics are made with those that only have numeric values.

d. Return = blanks.

```
Price USD =
VAR Price = RELATED(Countries[USD_convertion]) * Propierties[Price _ local
currency]
VAR _Condition = IF(Price >40000, 0, Price)
VAR Blanks = IF(_Condition=0,BLANK(),_Condition)
RETURN Blanks
```

### *Price Range USD*

Create ranges from the price in USD to be able to make a correct analysis.

a. Range_distance: Determine the range (number) of each interval.

b. Multiplo_range: R Round the Price USD in Distance_range.

c. Lower_range:

   a. If Multiplo_range is less than Price USD, then that multiplo + 1 cent will be taken.

   b. Otherwise, that multiplo will be subtracted Distance_range to go to the lower range and the cent on the dollar will be added.

d. Upper_range:

   a. If Multiplo_range is greater than or equal to Price USD, that multiplo will be taken as the upper range.

   b. Otherwise, the range distance will be added.

e. Concatenate: Concatenates the lower rank, a hyphen and the upper rank.

f. Result = concatenate.

```
Price range USD =
VAR Range_distance = 40
VAR Multiplo_range = MROUND(Propierties[Price USD], Range_distance)
VAR Lower_range = IF(Multiplo_range<Propierties[Price USD], Multiplo_range+0.01,
Multiplo_range - Range_distance + 0.01)
VAR Upper_range = IF (Multiplo_range>=Propierties[Price USD], Multiplo_range,
Multiplo_range + Range_distance)
VAR Concatenate = Lower_range & " - " & Upper_range
VAR Result = IF(Propierties[Price USD]=BLANK(),BLANK(),Concatenate)
RETURN Result
```

### *Price level  USD*

Divide the price into 5 categories.

a. USD_Price: Price in USD.

b. _max_price: The highest price in the entire table in USD.

c. Range1: These are the prices that reach a maximum of the 5th part of the _max_price.

d. Range2: These are the prices that reach a maximum of the 2/5th part of the _max_price.

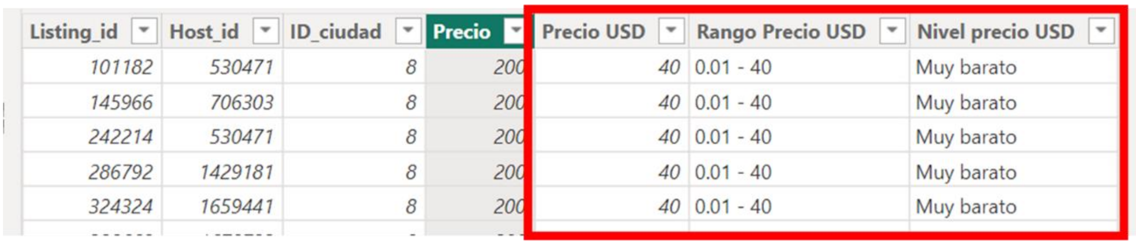
e. Range3: These are the prices that reach a maximum of the 3/5th part of the _max_price.

f. Range4: These are the prices that reach a maximum of the 4/5th part of the _max_price.

g. _Level: is the conditional where range1 is equivalent to “Very cheap”, otherwise, if it does not exceed range2 it is equivalent to “Cheap”, following the logic, if it does not exceed range3 it is “Regular price”, if it does not exceed range4 it is “High price”, and the rest are “Very high price”

h. Result = _Level.

```
Price level USD =
VAR USD_price = Propierties[Price USD]
VAR _max_price = MAX(Propierties[Price USD])
VAR Range1 = _max_price/5
VAR Range2 = _max_price/5*2
VAR Range3 = _max_price/5*3
VAR Range4 = _max_price/5*4
VAR _Level = IF ( USD_price <= Range1 , "Very cheap",
                IF ( USD_price <= Range2 , "Cheap",
                 IF ( USD_price <= Range3 , "Regular price",
                 IF ( USD_price <= Range4 , "Expensive",
                 "Very expensive"))))
RETURN _Level
```



| Listing_id | Host_id | ID_ciudad | Precio | Precio USD | Rango Precio USD | Nivel precio USD |
|---|---|---|---|---|---|---|
| 101182 | 530471 | 8 | 200 | 40 | 0.01 - 40 | Muy barato |
| 145966 | 706303 | 8 | 200 | 40 | 0.01 - 40 | Muy barato |
| 242214 | 530471 | 8 | 200 | 40 | 0.01 - 40 | Muy barato |
| 286792 | 1429181 | 8 | 200 | 40 | 0.01 - 40 | Muy barato |
| 324324 | 1659441 | 8 | 200 | 40 | 0.01 - 40 | Muy barato |

## 15.2. Reviews table

*Price range USD*

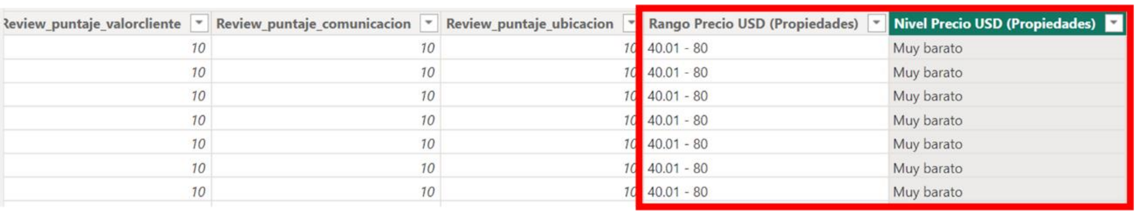
Copies the Price Ranges in USD from the Property Table to the Reviews table.

```
Price range USD (Propierties) =
LOOKUPVALUE(
    Propierties[Price range USD],
    Propierties[Listing_id],
    Reviews[Listing_id])
```

*Price level USD*

Copies the Price Levels (categories) in USD from the Property Table to the Reviews table.

```
Price level USD (Propierties) =
LOOKUPVALUE(
    Propierties[Price level USD],
    Propierties[Listing_id],
    Reviews[Listing_id])
```



| Review_puntaje_valorcliente | Review_puntaje_comunicacion | Review_puntaje_ubicacion | Rango Precio USD (Propiedades) | Nivel Precio USD (Propiedades) |
|---|---|---|---|---|
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |
| 10 | 10 | 10 | 40.01 - 80 | Muy barato |

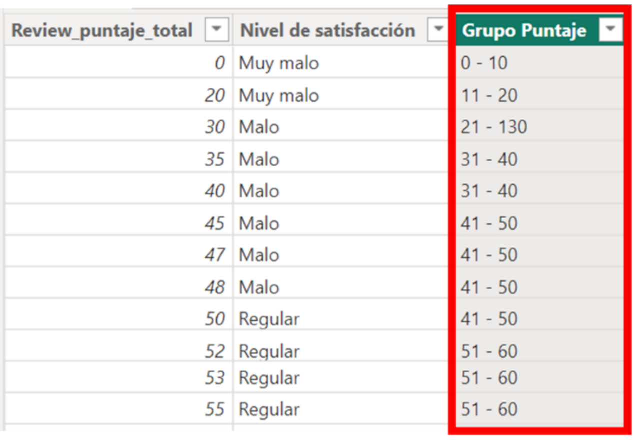
### 15.3. Reviews _satisfaction total table

New columns on the table created in the chapter 14.2 Review Score to Categories.

*Score group*

Group the total scores in groups of 10.

```
Score group =
VAR Points = 'Reviews _satisfasfation total'[Review_puntaje_total]
VAR _Group =
    IF(Points <= 10, "0 - 10",
    IF(Points <= 20, "11 - 20",
    IF(Points <= 30, "21 - 130",
    IF(Points <= 40, "31 - 40",
    IF(Points <= 50, "41 - 50",
    IF(Points <= 60, "51 - 60",
    IF(Points <= 70, "61 - 70",
    IF(Points <= 80, "71 - 80",
    IF(Points <= 90, "81 - 90",
    "91 - 100"
    )))))))))
RETURN _Group
```



| Review_puntaje_total | Nivel de satisfacción | Grupo Puntaje |
|---|---|---|
| 0 | Muy malo | 0 - 10 |
| 20 | Muy malo | 11 - 20 |
| 30 | Malo | 21 - 130 |
| 35 | Malo | 31 - 40 |
| 40 | Malo | 31 - 40 |
| 45 | Malo | 41 - 50 |
| 47 | Malo | 41 - 50 |
| 48 | Malo | 41 - 50 |
| 50 | Regular | 41 - 50 |
| 52 | Regular | 51 - 60 |
| 53 | Regular | 51 - 60 |
| 55 | Regular | 51 - 60 |

## 16. DAX Measures

### 16.1. Measures table

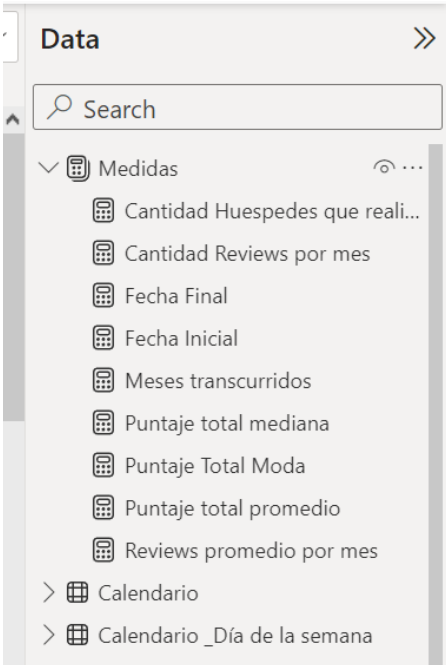
The “Measures” table was created, where all the measures we create will be ordered. To do so, the following steps were performed:

a. Create an empty table called “_Measures””.

b. Create the first measure in this table.

c. Delete the empty column leaving only the measurement.

This steps will place the table at the top of our dashboard. The other measures must be created in this table, if it is created in a separate table, it can be dragged from the Model view.


Data

Search

Medidas
- Cantidad Huespedes que reali...
- Cantidad Reviews por mes
- Fecha Final
- Fecha Inicial
- Meses transcurridos
- Puntaje total mediana
- Puntaje Total Moda
- Puntaje total promedio
- Reviews promedio por mes

Calendario

Calendario _Día de la semana

### 16.2. Measures

*Date start*

It is the first date in the calendar, it is used to verify the filters and to be able to create new measures more easily.

```
Date start = MIN(Calendar[Date])
```

*Date final*

This is the last date in the calendar, used to check the filters and to be able to create new measures more easily.

```
Date final = MAX(Calendar[Date])
```

*Host that make reviews per month*

Calculate how many guests (with different IDs) performed Reviews per month.

```
Host that make reviews per month =
CALCULATE(DISTINCTCOUNT(Reviews[Host_id]),Calendar[Month])
```

*Reviews average per month*

Calculate how many Reviews were performed per month.

```
Reviews average per month =
VAR number_reviews = DISTINCTCOUNT(Reviews[Review_id])
VAR division = DIVIDE(number_reviews, [Months])
VAR result = IF(division=BLANK(),number_reviews,division)
RETURN result
```

### *Months*

Calculates the number of months elapsed from the start date to the end date.

```
Months = DATEDIFF([Date start], [Date final], MONTH)
```

### *Total score - median*

Median of total score.

```
Total score - median = MEDIAN(Reviews[Review_total])
```

### *Total score - average*

Average of total score.

```
Total score - average = AVERAGE(Reviews[Review_total])
```

### *Total score - mode*

Mode of the total score. To obtain this statistic, the following steps were taken:

a. filter_no_blanks: Filter the review table with all scores that have some value. Scores that are blank (null) are ignored.

b. _count: Create internally a summary table of Filtro_no_blanks, with the total score columns and the count of each score.

c. _max: Is the maximum count value of the table _count.

d. mode: Calculates the average of the modes (scores with the highest number of counts).

e. Result = mode.

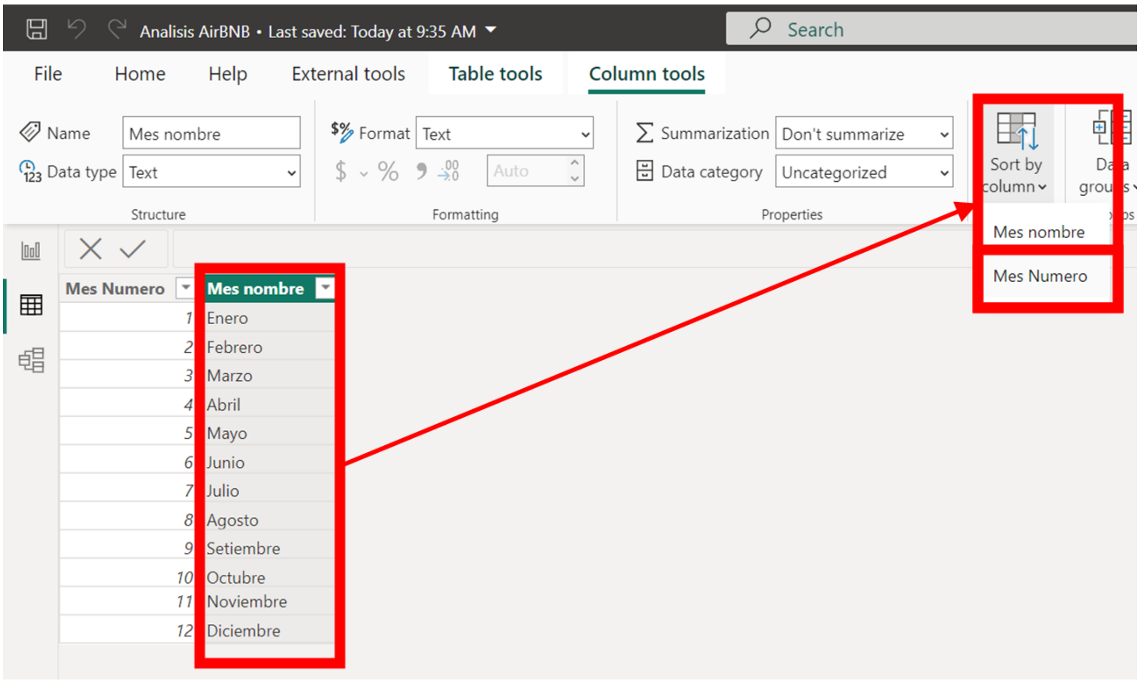
```
Total score - mode =
VAR filter_no_blanks = FILTER(Reviews, Reviews[Review_total]<>BLANK())
VAR _count = SUMMARIZE(
    filter_no_blanks,
    [Review_total],
    "total",
    COUNT(Reviews[Review_total])
    )
VAR _max = MAXX(_count, [total])

VAR mode =
    CALCULATE(
        AVERAGE(Reviews[Review_total]),
        FILTER(_count, [total] = _max)
    )

RETURN mode
```

# 17. Other preparations

## 17.1. Ordering of months and days in complementary calendaring tables

a. Select the column to be sorted (month Name or weekday Name, depending on the table).

b. In the column tool tab select "sort by" and the numerical category of the month or day of the week.

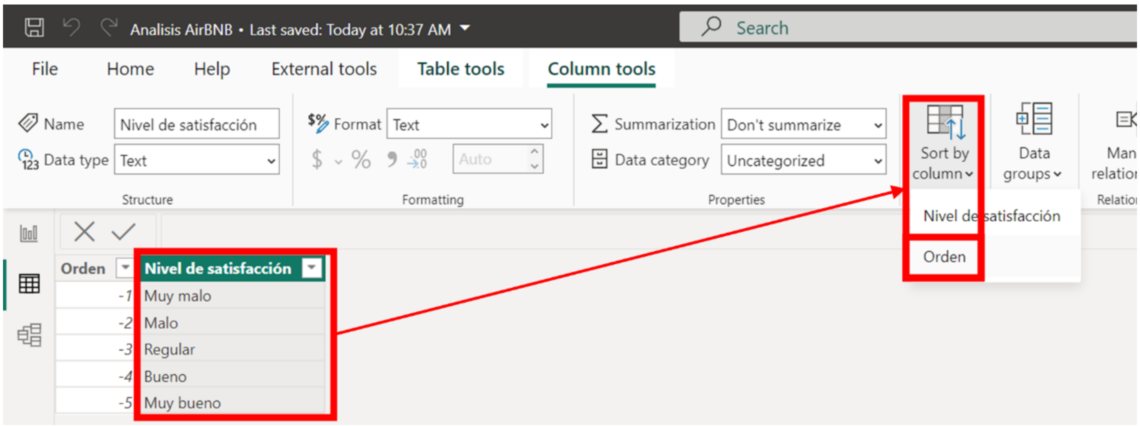


Name: Mes nombre
Data type: Text
Format: Text
Auto
Summarization: Don't summarize
Data category: Uncategorized

| Mes Numero | Mes nombre |
|---|---|
| 1 | Enero |
| 2 | Febrero |
| 3 | Marzo |
| 4 | Abril |
| 5 | Mayo |
| 6 | Junio |
| 7 | Julio |
| 8 | Agosto |
| 9 | Setiembre |
| 10 | Octubre |
| 11 | Noviembre |
| 12 | Diciembre |

## 17.2. Satisfaction Category scores ordering

The same process is followed as "Ordering of months and days in complementary calendaring tables" with the table "Reviews _Satisfaction scores ordered".



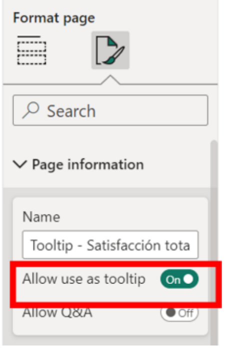
Name: Nivel de satisfacción
Data type: Text
Format: Text
Auto
Summarization: Don't summarize
Data category: Uncategorized

| Orden | Nivel de satisfacción |
|---|---|
| -1 | Muy malo |
| -2 | Malo |
| -3 | Regular |
| -4 | Bueno |
| -5 | Muy bueno |

## 17.3. Tooltips

a. The Tooltip page is created.

b. In Page Setup, Page Info, "Allow to use as tooltip" is enabled.



Format page

Page information

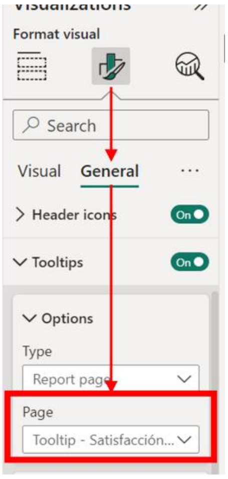
Name: Tooltip - Satisfacción tota

Allow use as tooltip: On

Allow Q&A: Off

c. Select the visual object to which you want to add this Tooltip.

d. In Visual Object Format, General, Tooltips, select the page that was created in step a.



Format visual

Visual General

Header icons: On

Tooltips: On

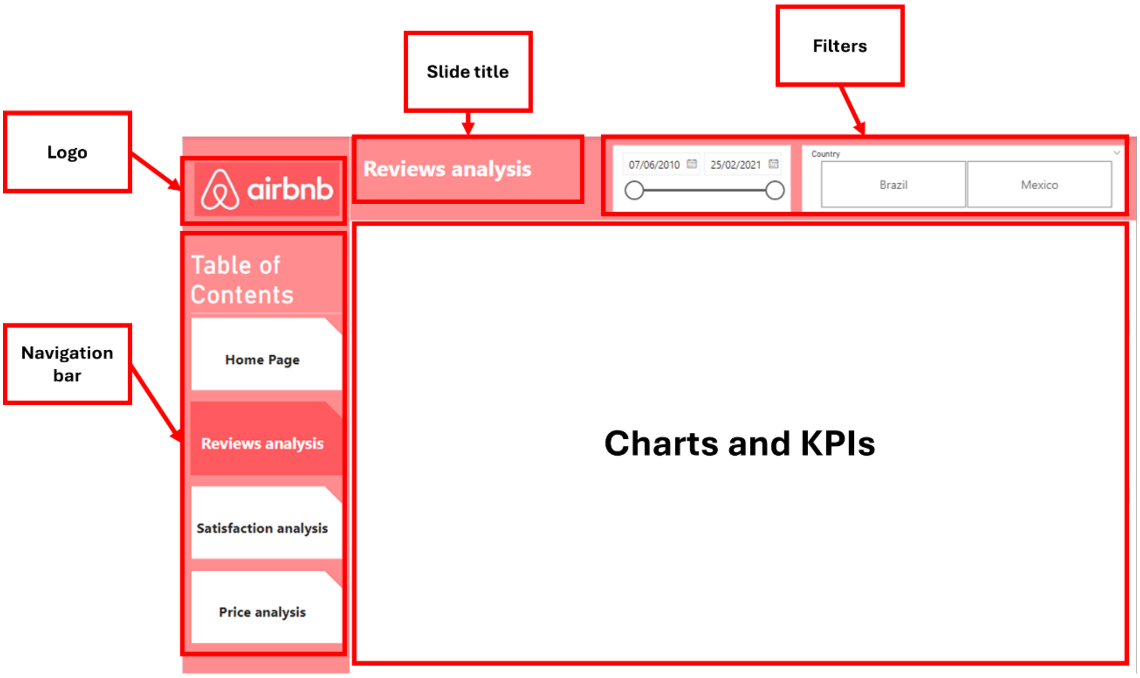
Options

Type: Report page

Page: Tooltip - Satisfacción...

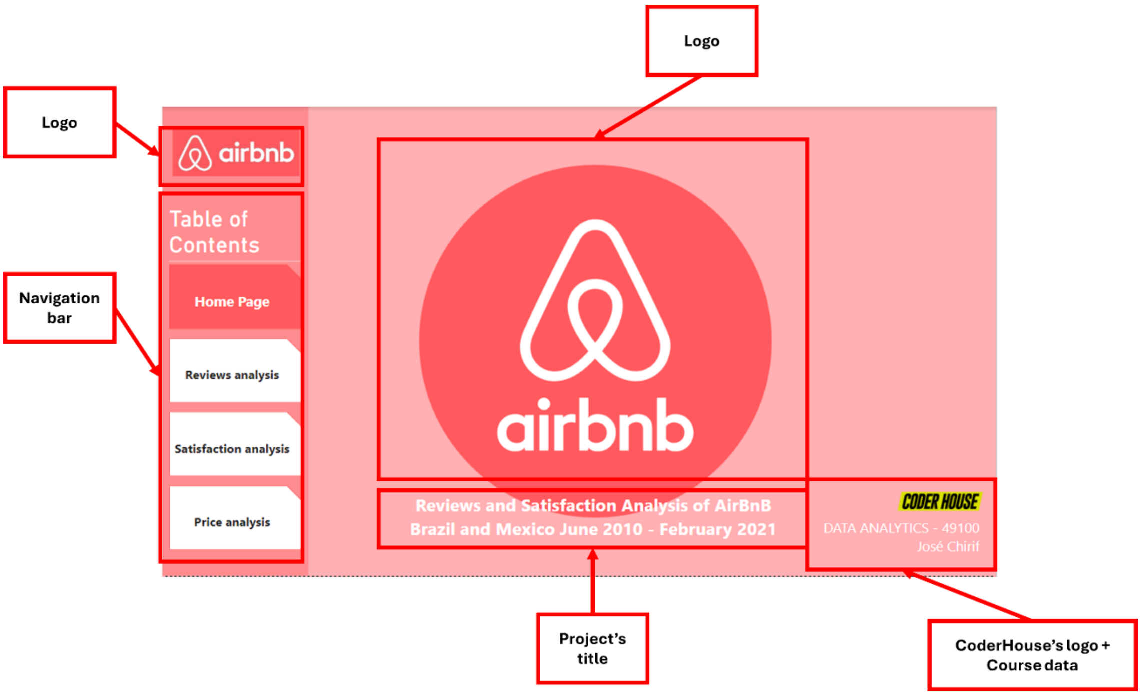
## 18. Structure of the slides

The slides have the following structure:



# 19. Dashboard

## 19.1. Home page



Logo

Logo

Navigation bar

airbnb

Table of Contents

Home Page

Reviews analysis

Satisfaction analysis

Price analysis

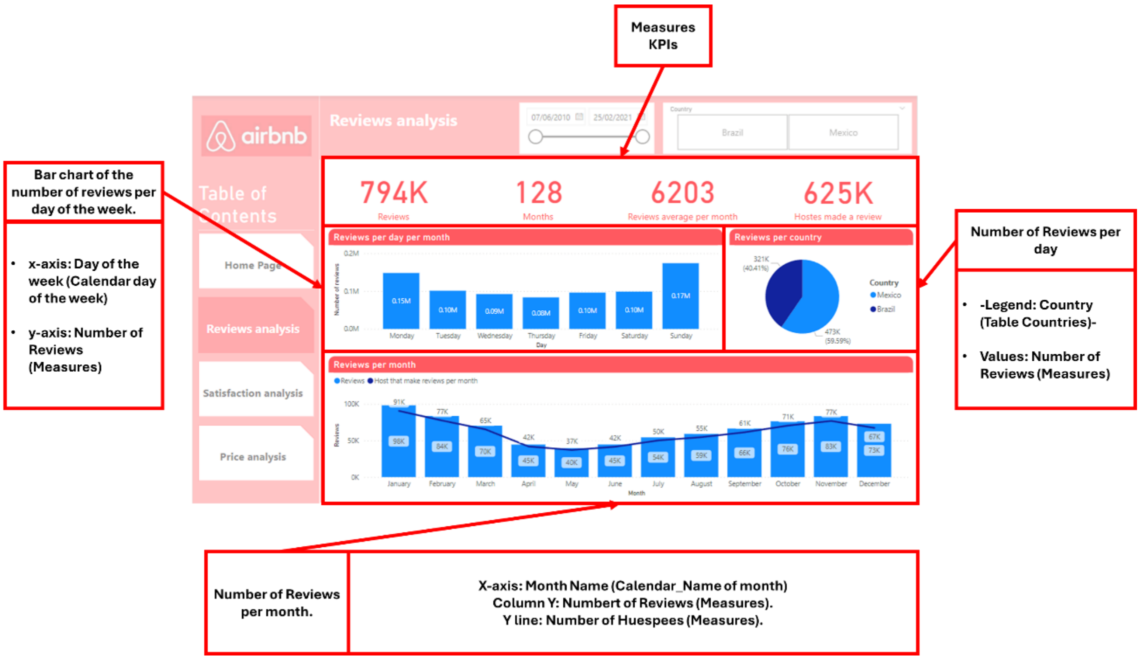
Reviews and Satisfaction Analysis of AirBnB
Brazil and Mexico June 2010 - February 2021

CODER HOUSE
DATA ANALYTICS - 49100
José Chirif

Project's title

CoderHouse's logo + Course data

## 19.2. Reviews analysis



Measures KPIs

Bar chart of the number of reviews per day of the week.

- x-axis: Day of the week (Calendar day of the week)
- y-axis: Number of Reviews (Measures)

Number of Reviews per day

- -Legend: Country (Table Countries)-
- Values: Number of Reviews (Measures)

airbnb

Reviews analysis

07/06/2010 25/02/2021

Country: Brazil, Mexico

Table of Contents: Home Page, Reviews analysis, Satisfaction analysis, Price analysis

794K Reviews

128 Months

6203 Reviews average per month

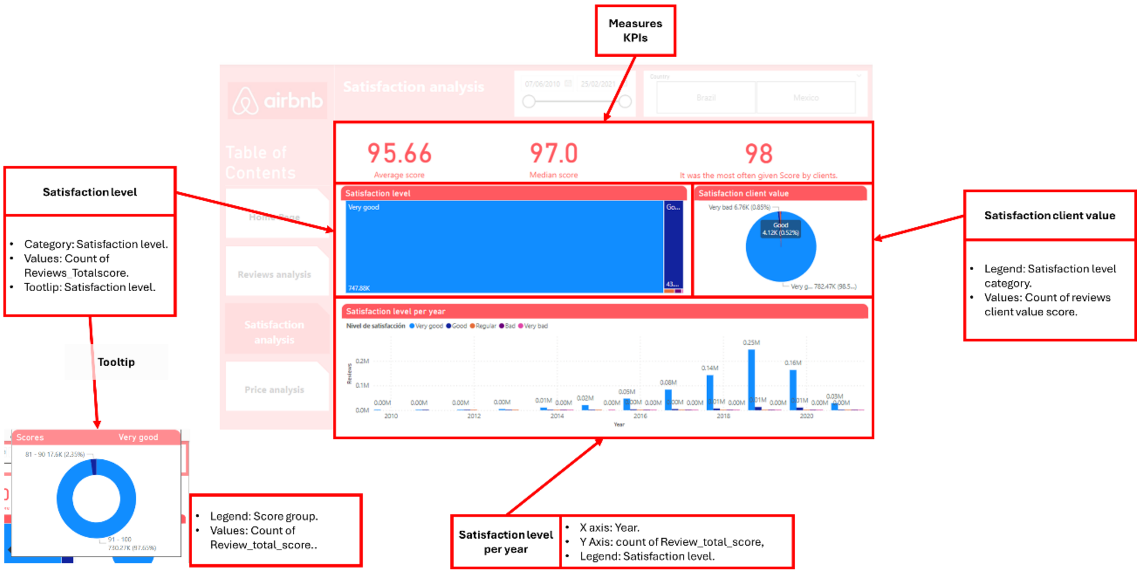
625K Hostes made a review

Number of Reviews per month.

X-axis: Month Name (Calendar_Name of month)
Column Y: Numbert of Reviews (Measures).
Y line: Number of Huespees (Measures).

**Note:** To this slide the filter Date has been applied after June 6, 2010, because there are no Reviews dated prior to that date.

## 19.3. Satisfaction analysis



Measures KPIs

**Satisfaction level**

- Category: Satisfaction level.
- Values: Count of Reviews_Totalscore.
- Tooltip: Satisfaction level.

Tooltip

- Legend: Score group.
- Values: Count of Review_total_score..

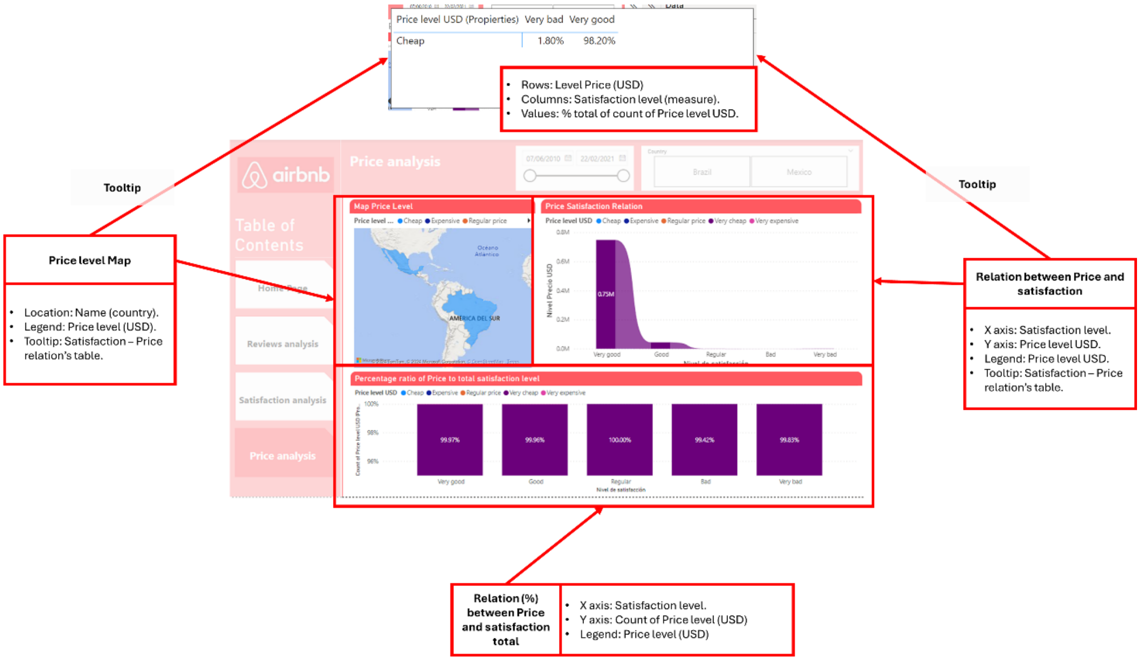
**Satisfaction client value**

- Legend: Satisfaction level category.
- Values: Count of reviews client value score.

**Satisfaction level per year**

- X axis: Year.
- Y Axis: count of Review_total_score,
- Legend: Satisfaction level.

## 19.4. Price analysis



| Price level USD (Propierties) | Very bad | Very good |
|---|---|---|
| Cheap | 1.80% | 98.20% |

- Rows: Level Price (USD)
- Columns: Satisfaction level (measure).
- Values: % total of count of Price level USD.

Tooltip

Tooltip

**Price level Map**

- Location: Name (country).
- Legend: Price level (USD).
- Tooltip: Satisfaction – Price relation's table.

**Relation between Price and satisfaction**

- X axis: Satisfaction level.
- Y axis: Price level USD.
- Legend: Price level USD.
- Tooltip: Satisfaction – Price relation's table.

**Relation (%) between Price and satisfaction total**

- X axis: Satisfaction level.
- Y axis: Count of Price level (USD)
- Legend: Price level (USD)

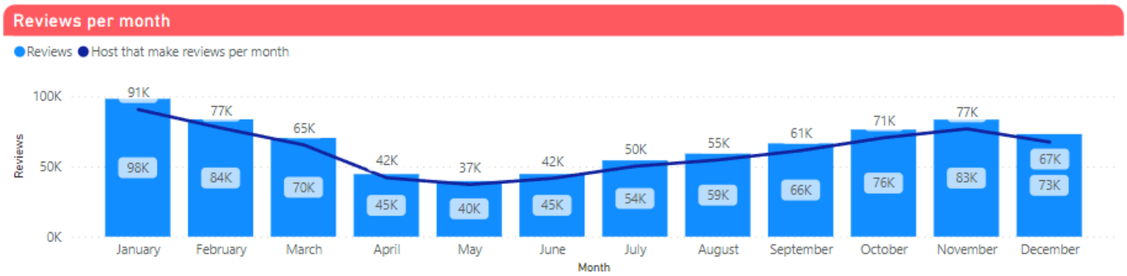
# VI. Conclusions and recommendations

## 20. Hypothesis corroboration

### 20.1. Hypothesis 1

Hypothesis: “Between December and February (holidays and summer vacations) there are more reviews due to people's availability”.

In the “Reviews Analysis” slide we have the following chart.



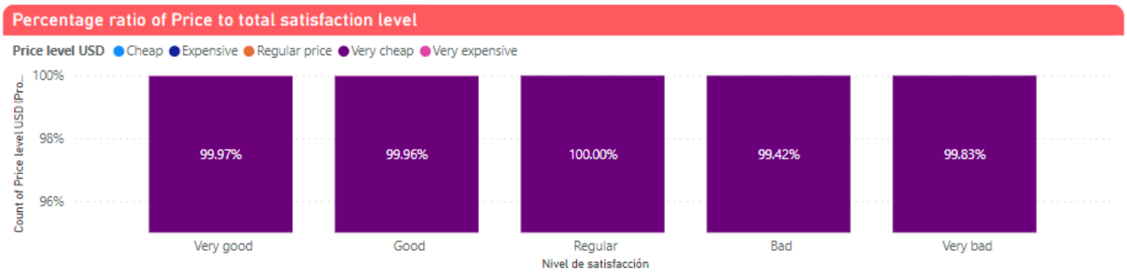
This graph shows that in January the number of reviews starts to decline and in May it starts to grow gradually. However, from November to December there is a small drop in the number of reviews.

Therefore, this is a **false hypothesis.** The months with the highest number of reviews are January, February and November respectively.

### 20.2. Hypothesis 2

Hypothesis: “Percentagewise, higher cost properties have better average customer ratings”.

In the “Price Analysis” slide we have the following chart.



In this slide, we can observe that regardless of the price (most of the rentals are “very cheap”), users have mixed ratings of the properties. Therefore, it is **false Hypothesis.**

## 21. Conclusions and recommendations

a. There is a crowded season (of number of reviews) at the beginning of the year the highest peak, from there it decreases until May and starts to grow gradually until November where there is a small decrease to December. The number of reviews may be subject to the number of bookings.

   **Recommendation:** Align marketing strategies with the seasonality of bookings. I recommend making projections, which would be the subject of another more in-depth study, since it is not within the scope of the project.

b. Although it is true that customer satisfaction levels are distributed at any price, the relationship between "High price" and "Poor satisfaction" was noticeable.

   **Recommendation:** Verify whether the prices of this group of properties are in line with what they offer, and then ensure that the end user experience is improved regardless of the price.