

Máster Universitario en Big Data y Ciencia de Datos

ASIGNATURA: 01MBID *Fundamentos de la Tecnología Big data*

Actividad 1 – Tareas portafolio

Alumno: **Bru Montes, Israel**

Edición **Octubre 2024 – Grupo A** a 8/11/2024

1. Tareas

1.- Crear una base de datos MongoDB en la nube usando MongoDB Atlas (<https://www.mongodb.com/>)

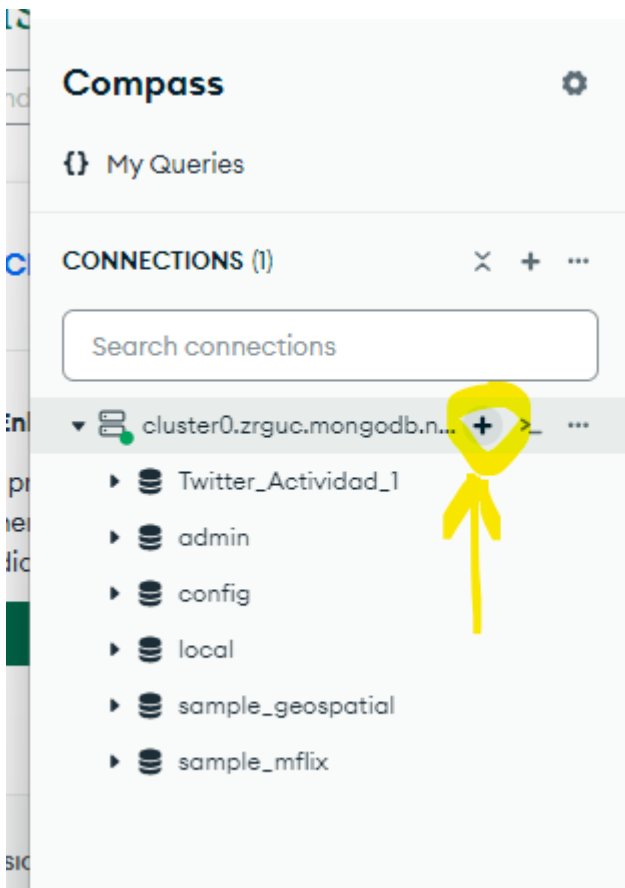
Nos vamos a descargar el GUI compass para poder administrar la base de datos desde una aplicación de escritorio, para ello cogemos la url de conexión y cambiamos el parámetro <password>

The image shows two screenshots. The top screenshot is from the MongoDB Atlas web interface, specifically the 'Clusters' page for 'ISRAEL'S ORG - 2024-10-26 > PROJECT 0'. It shows a 'Cluster0' with a 'Connect' button. A modal window titled 'Connect to Cluster0' is open, showing steps for connecting with MongoDB Compass. It offers two options: 'I don't have MongoDB Compass installed' (highlighted) and 'I have MongoDB Compass installed'. The first step is to select an operating system (macOS arm64 (M1) (11.0+)) and download the Compass (1.44.6) or copy the download URL. The second step is to copy the connection string and open MongoDB Compass. The connection string is shown as 'mongodb+srv://ibru:<db_password>@cluster0.zrguc.mongodb.net/'. The bottom screenshot shows the MongoDB Compass desktop application. It has a sidebar with 'My Queries' and 'CONNECTIONS'. The main area shows a 'Welcome' message and a 'Welcome to MongoDB Compass' section with a 'Add new connection' button. Below this, there is a section titled 'New to Compass and don't have a cluster?' with a 'CREATE FREE CLUSTER' button.

Pulsamos sobre “add new connection”

The screenshot shows the 'New Connection' dialog box in MongoDB Compass. The dialog has a title bar with a close button. Below the title, it says 'Manage your connection settings'. There are two tabs: 'URI' (selected) and 'Edit Connection String'. The 'URI' tab contains a text input field with the value 'mongodb+srv://ibru:<db_password>@cluster0.zrguc.mongodb.net/'. Below this, there are two input fields: 'Name' with the value 'cluster0.zrguc.mongodb.net' and 'Color' with a dropdown menu showing 'No Color'. There is a checkbox labeled 'Favorite this connection' with the text 'Favoriting a connection will pin it to the top of your list of connections' below it. At the bottom left, there is a 'Cancel' button. At the bottom right, there are two buttons: 'Save' and 'Save & Connect'. On the right side of the dialog, there are two informational panels. The first panel is titled 'How do I find my connection string in Atlas?' and contains text about finding the connection string in the Atlas cluster view, with a 'See example' link. The second panel is titled 'How do I format my connection string?' and also contains a 'See example' link.

A través del botón “+”



Create Database

Database Name

New_Tweeters

Collection Name

tweets

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences

(e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

2.- Crear las colecciones



Create Collection

Collection Name

new-tweeters_account

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Collection

Import

To collection New_Tweeters.tweets

Import file: tweets_Actividad_R.json

Options

☐ Stop on errors

Cancel

Import

Create Collection

Collection Name

new-tweeters_account

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)

CancelCreate Collection

Collection Name

1=

View

Create Database

Database Name

sample_geospatial

Collection Name

geospatial_data

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)

CancelCreate Database

ets

ge size

MB

etsA

ge size

0 kB

s:

Total index size:

442.37 kB

s:

Total index size:

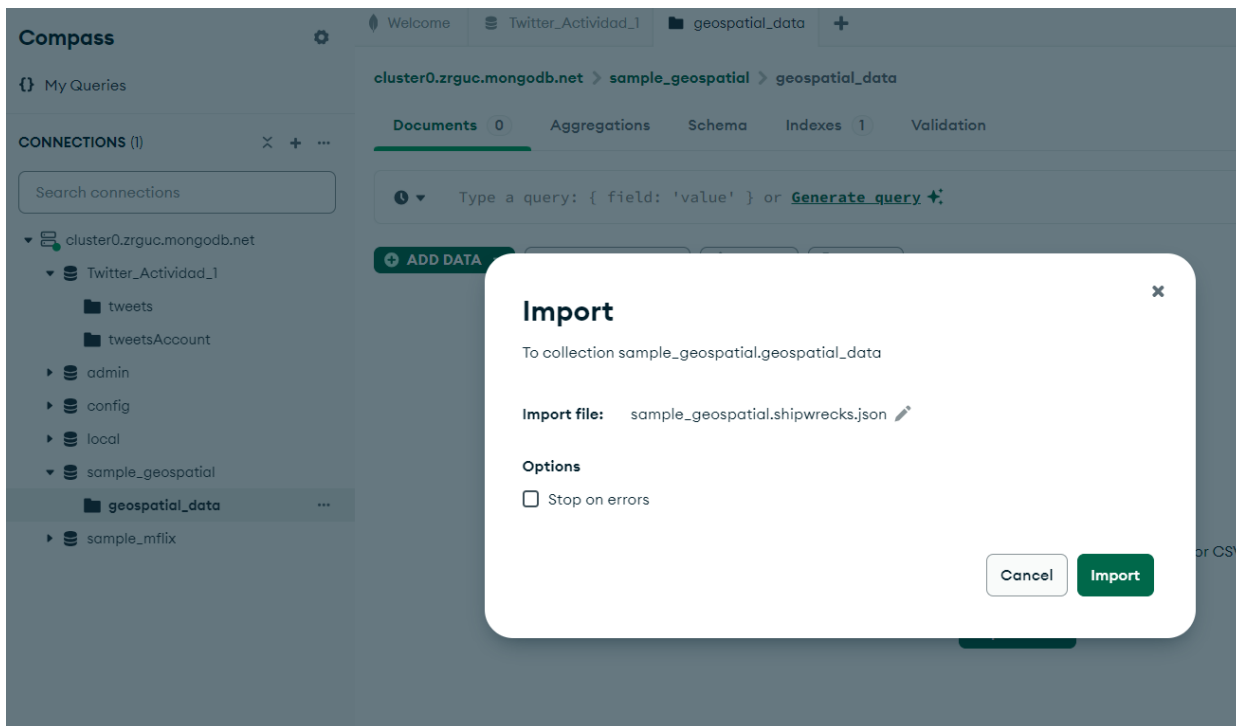
20.48 kB

2.1.- Cargar los datasets de cuentas de twitter y tweets proporcionados por el profesor.

The screenshot shows the MongoDB Atlas web interface. On the left, the 'CONNECTIONS (1)' sidebar lists a cluster named 'cluster0.zrguc.mongodb.net'. Under this cluster, there is a database 'New_Tweeters' and a collection 'new-tweeters_account'. The 'new-tweeters_account' collection is selected, and its contents are displayed in the main panel. The main panel shows the 'Documents' tab with 0 documents. A yellow circle highlights the 'ADD DATA' button, which has a dropdown menu with options 'Import JSON or CSV file' and 'Insert document'. Below the main panel, a message states 'This collection has no data' and 'It only takes a few seconds to import data from a JSON or CSV file.'

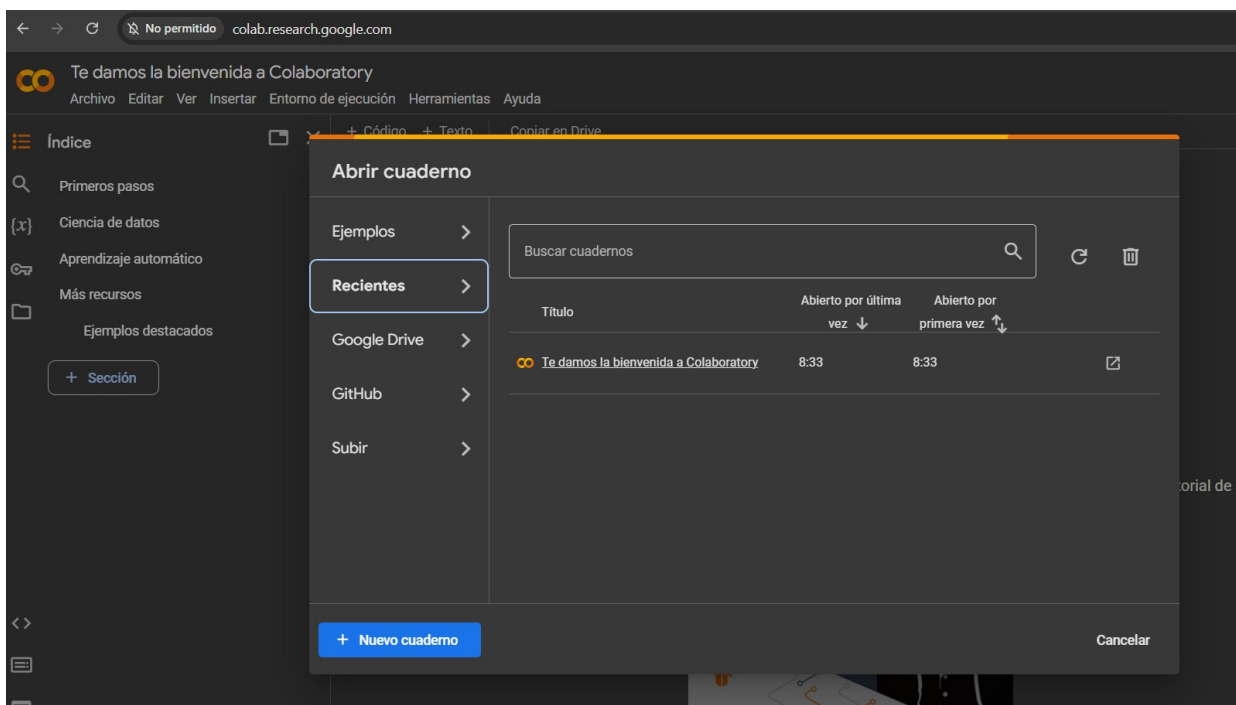
2.2.- Cargar la colección de ejemplo geolocalizada de MongoDB Atlas (sample_geospatial -> shipwrecks).

This screenshot is identical to the one above, showing the MongoDB Atlas interface with the 'new-tweeters_account' collection selected. It highlights the 'ADD DATA' button and the message 'This collection has no data'.

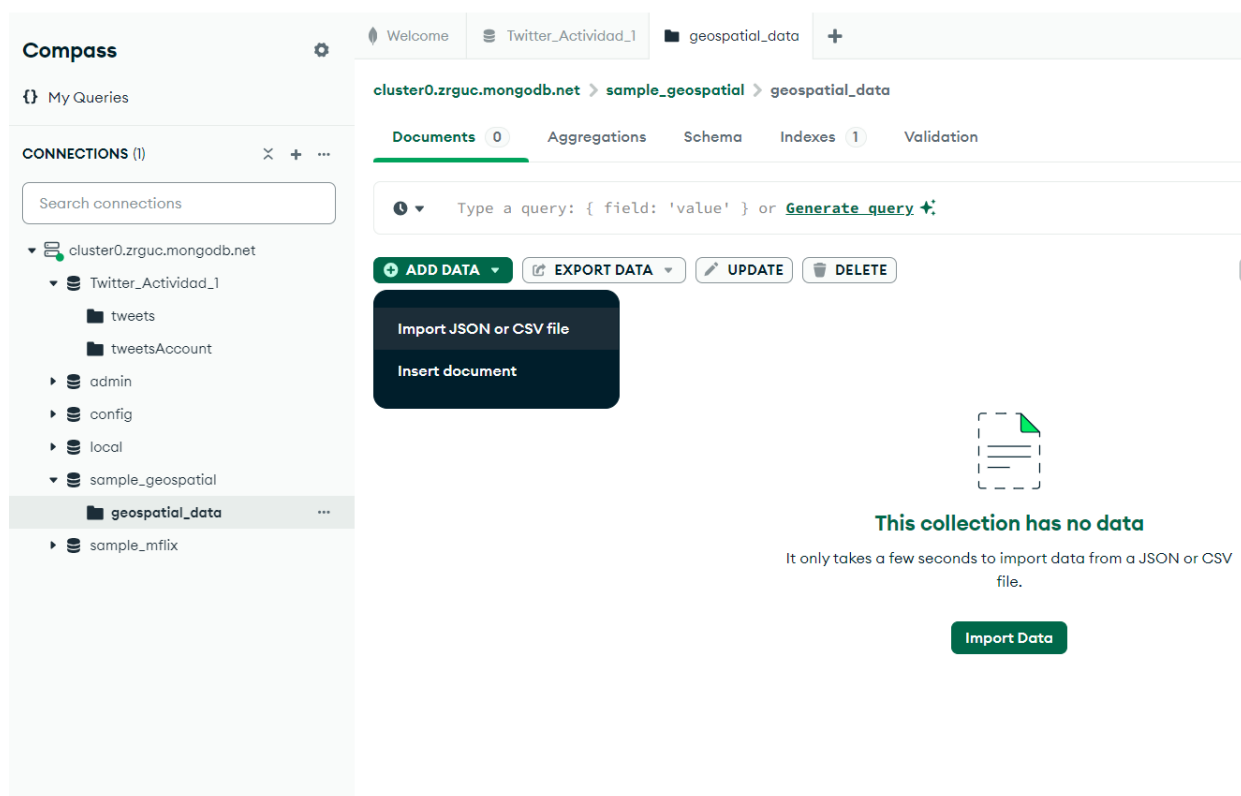


3.- Crear una cuenta en Google Colab.

Aunque con una cuenta de gmail es supersencillo, te das de alta directamente y cargas el notebook.



4.- Utilizando como base el script en Python profesor, realizar los cambios necesarios para:



4.1.- En la colección de cuentas de twitter, tener los campos amigos y tweets enviados, cargar los datos correspondientes mediante consulta mongodb + código python.

En el archivo phyton.

4.2.- En la colección de tweets, calcular la antigüedad para cada tweet en función de la fecha actual considerando antigüedad 0 el día de hoy y sumando +1 por cada día transcurrido. Incluir en el mismo documento del tweet un nuevo campo que se llamará antigüedad_dias con esa antigüedad calculada.

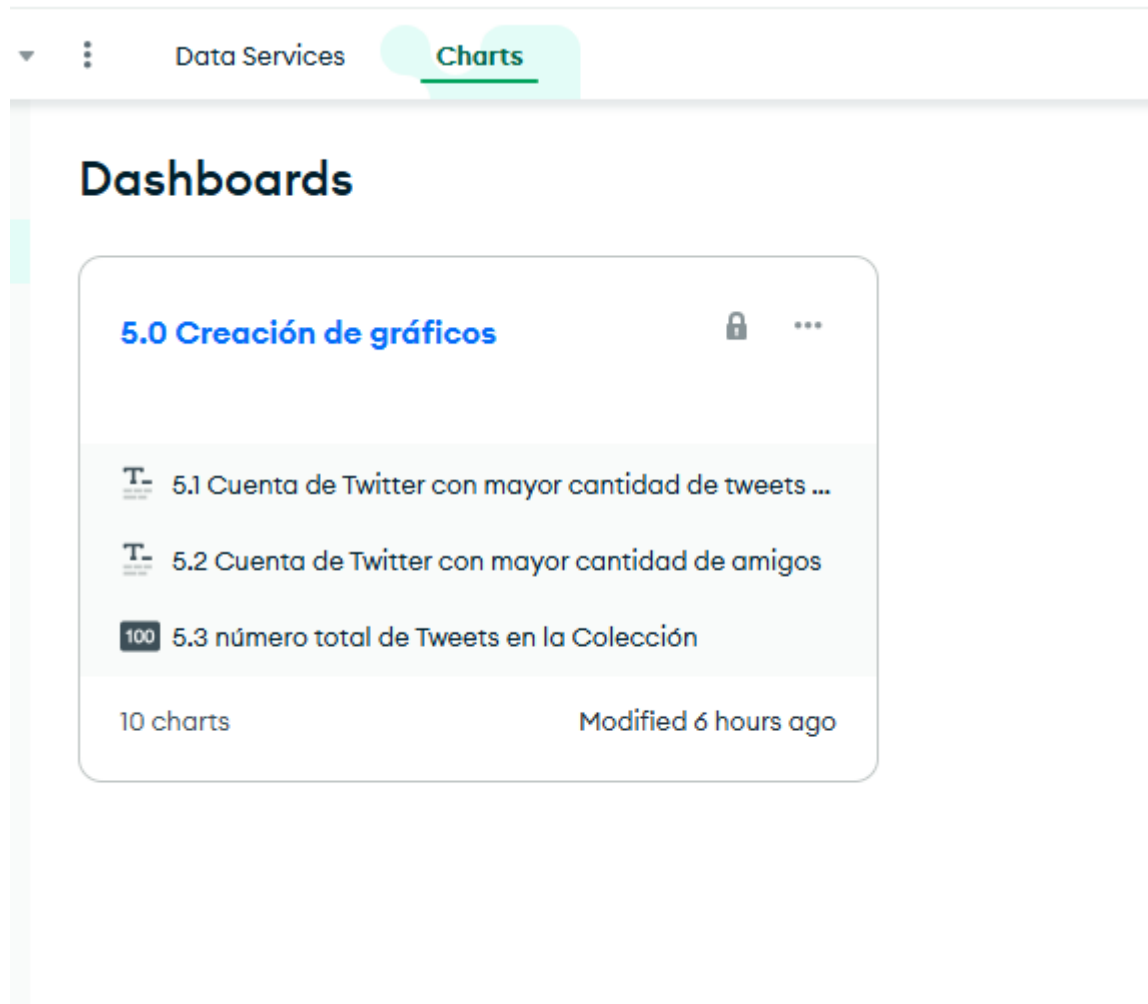
En el archivo phyton.

4.3.- En la colección de tweets, calcular la antigüedad de cada tweet relativa con la fecha de creación de la cuenta más antigua de la colección actual de datos. Considerando antigüedad 0 si fue enviado el mismo día de creación y sumando +1 por cada día transcurrido desde entonces en función de la fecha

del tweet. Incluir en el mismo documento del tweet un nuevo campo se llamará `frescura_relativa_dias`.

En el archivo `phyton`.

5.- Cuadro de mandos (MongoDB Charts)



5.1.- Crear un chart que muestre la cuenta de Twitter con mayor cantidad de tweets enviados.



He elegido “top item” porque permite mediante el campo “sort” ordenar el campus agregado “statuses_count” y mostrarlo.

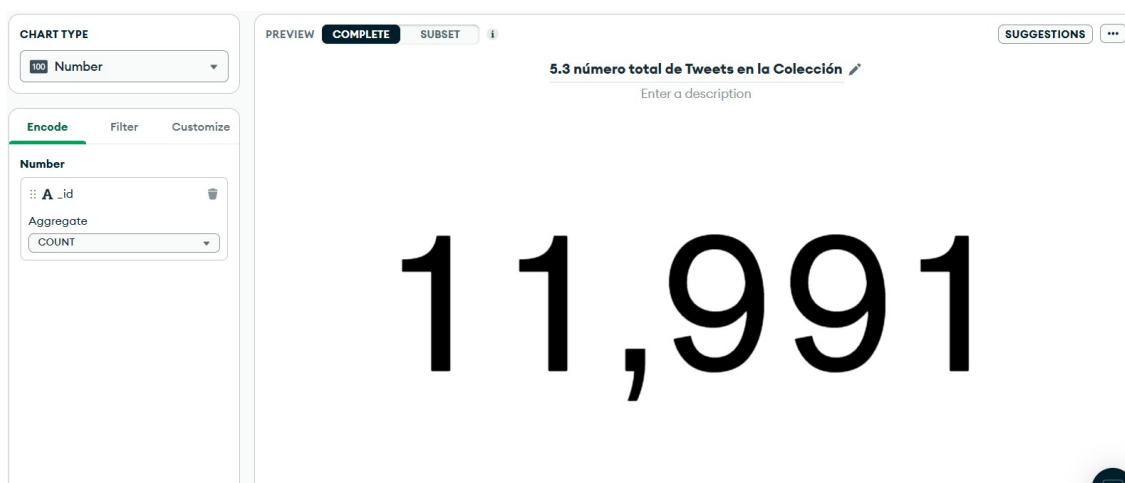
5.2.- Crear un chart que muestre la cuenta de Twitter con mayor cantidad de amigos (cuentas a las que sigue el usuario).



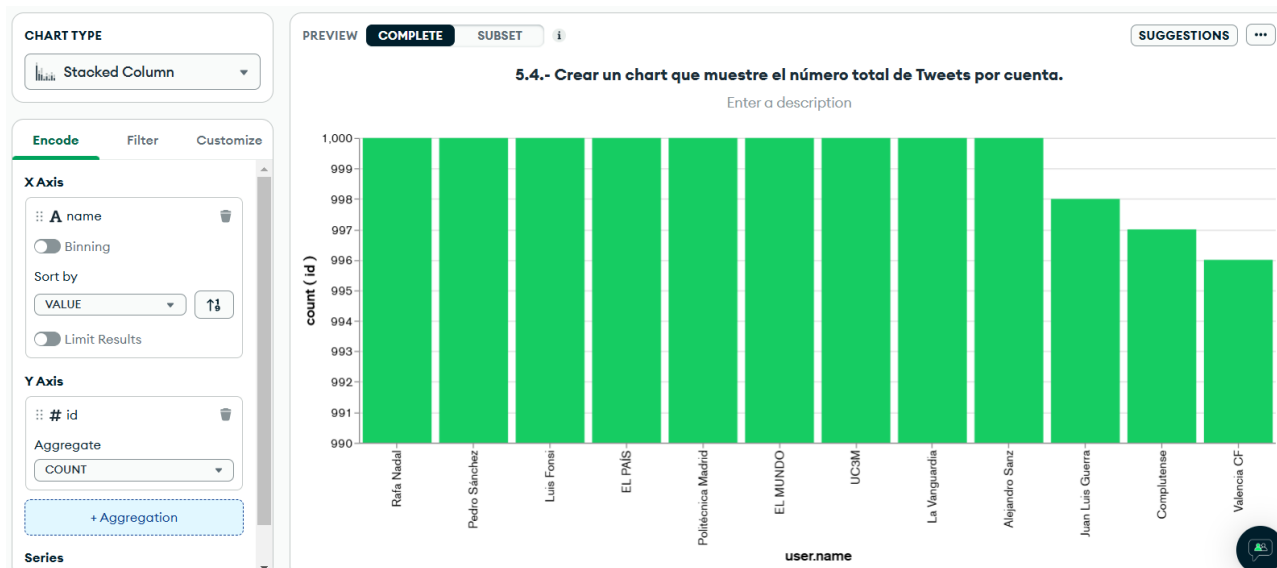
Como el anterior, con el tipo de gráfico “top item” nos permite crear una tarjeta con un valor agregado y seleccionar tanto el máximo como el mínimo, para nuestro caso seleccionamos el máximo.

5.3.- Crear un chart que muestre el número total de Tweets en la Colección.

Con el gráfico “Number” podemos obtener de forma agregada, para nuestro caso hacemos un count, todos los tweets que existen en nuestra colección.



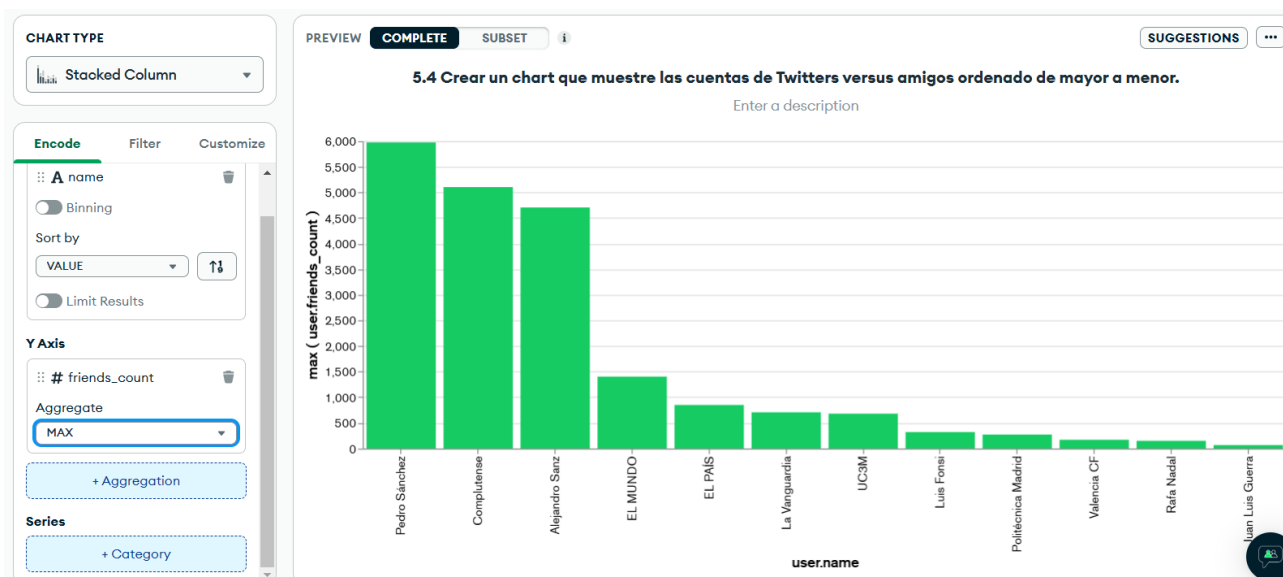
5.4.- Crear un chart que muestre el número total de Tweets por cuenta.



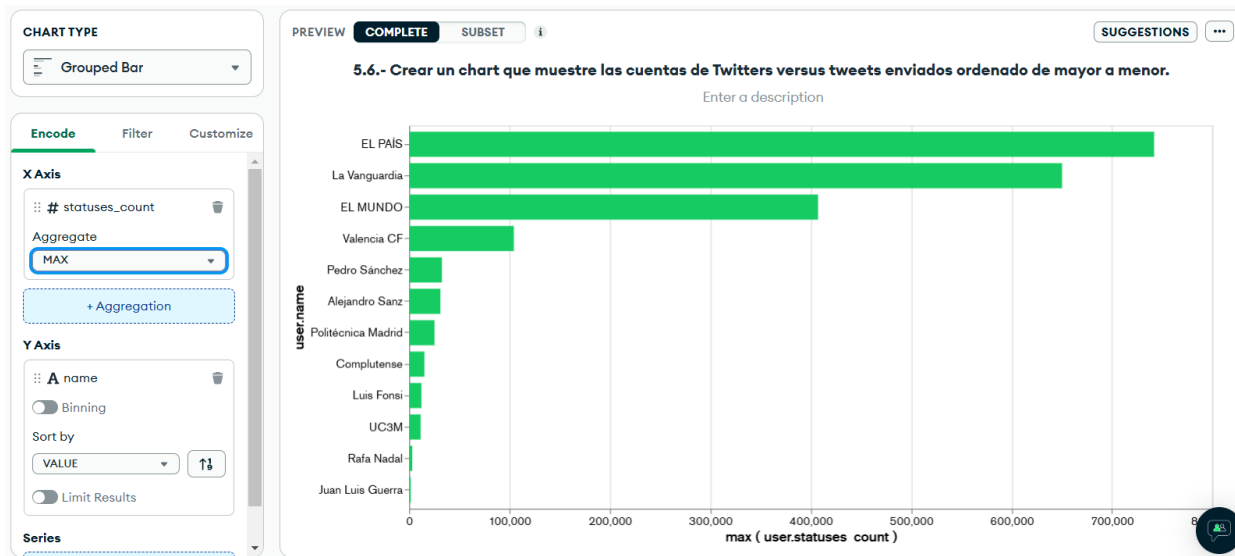
Utilizando el gráfico de barras por columnas “Stacked column”, hemos ordenado en modo descendente el número de tweets (count(id)) de cada cuenta de Tweet (name); hemos tenido que utilizar la “customización” para poder escalar el mínimo a 990, ya que no se apreciaba bien el gráfico.

5.5.- Crear un chart que muestre las cuentas de Twitters versus amigos ordenado de mayor a menor.

Elegimos el tipo de gráfico de barras “Stacked Column”, para mostrar la cuenta de Twitter haciendo un “max(friends.count)” y ordenarlo de forma descendente.

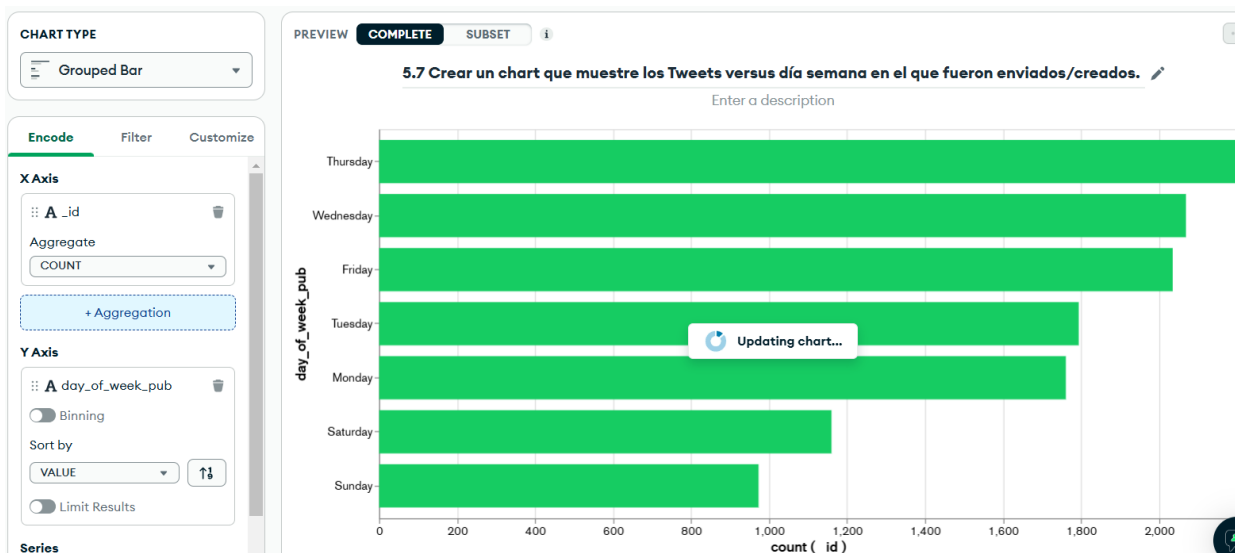


5.6.- Crear un chart que muestre las cuentas de Twitters versus tweets enviados ordenado de mayor a menor.



Aquí hemos utilizado el gráfico “grouped Bar” y hemos obtenido el máximo de los tweets enviados por cuenta, ordenandolo de forma descendente (el que tiene mayor número el primero).

5.7.- Crear un chart que muestre los Tweets versus día semana en el que fueron enviados/creados.

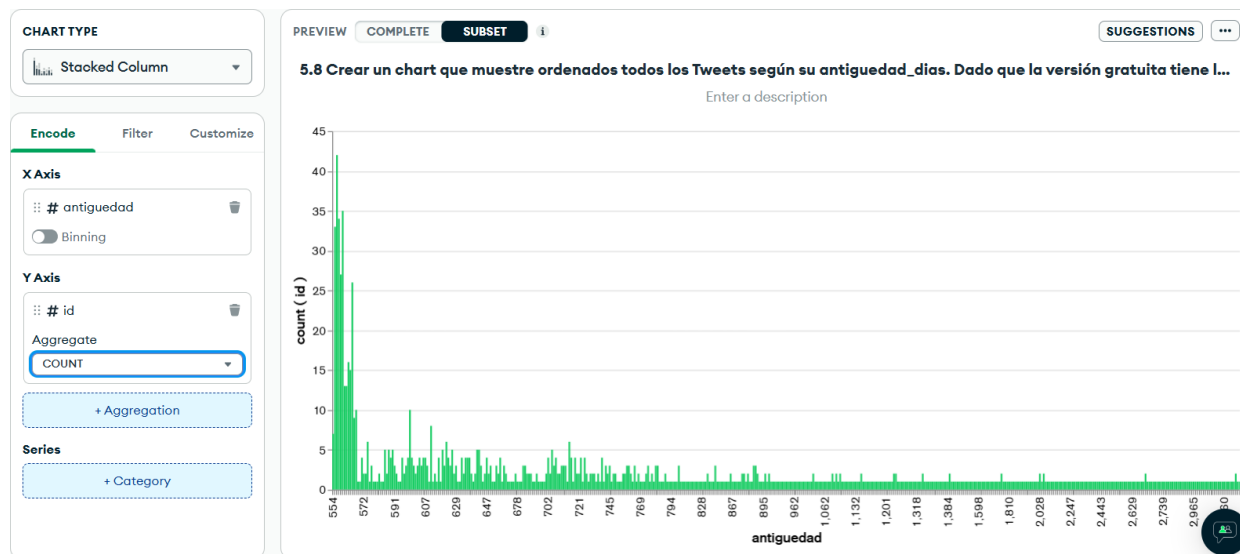


He creado desde la actividad 4.2 la columna fecha_pub y day_of_week_pub para indicar el día de la semana que se creo

5.8.- Crear un chart que muestre ordenados todos los Tweets según su antigüedad_dias. Dado que la versión gratuita tiene limitaciones de la cantidad

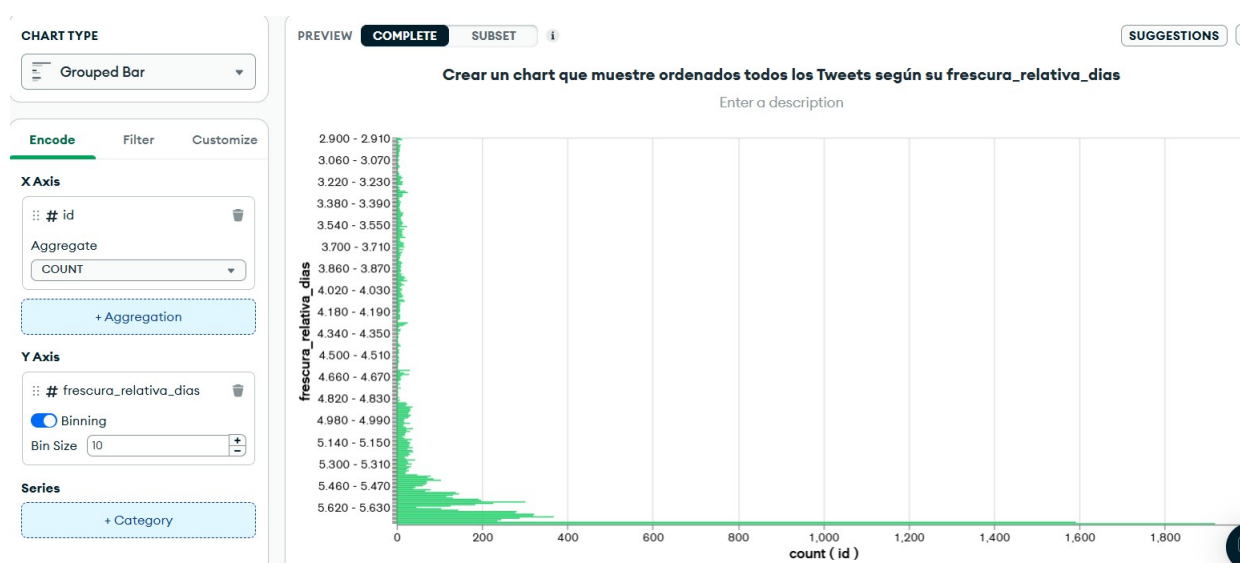
de datos a mostrar, mostrar dos charts: uno ordenado de mayor a menor y otro de menor a mayor.

Mostrando el count de los tweets agrupados sobre la antigüedad de estos.



5.9.- Crear un chart que muestre ordenados todos los Tweets según su frescura_relativa_días. Dado que la versión gratuita tiene limitaciones de la cantidad de datos a mostrar, mostrar dos charts: uno ordenado de mayor a menor y otro de menor a mayor.

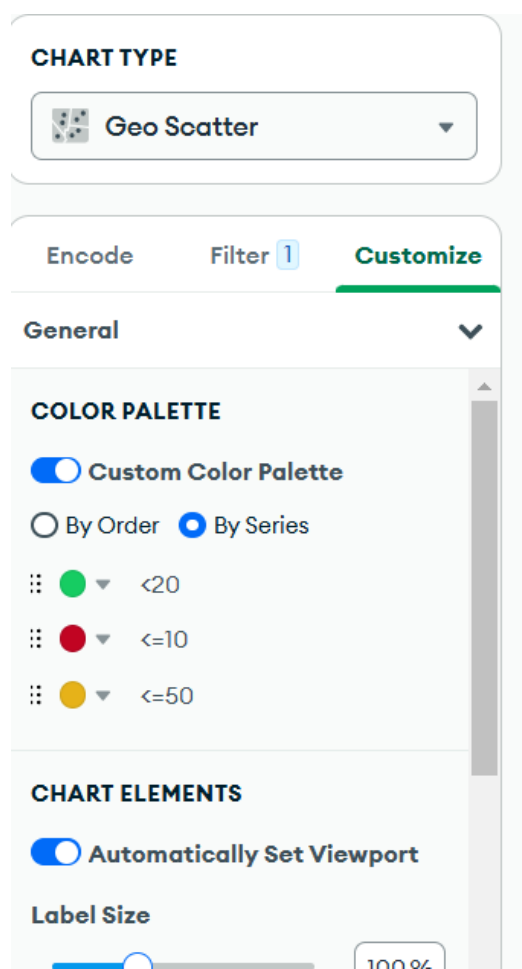
Con el group Bar mostramos los datos de la frescura relativa de días respecto de los tweets.



5.10.- Crear un chart para visualizar en el mapa mundial todos los pecios hundidos hasta 50 metros de profundidad. Pintando en verde los hundidos hasta los 10 metros, amarillo hasta los 20 metros y en rojo hasta los 50 metros.

Creando una nueva columna “depth_group” que agrupamos los resultados y después filtramos aquellos que son ‘+50’ para quitarlos:

```
{
  $cond: { if: { $lte: ['$depth', 10]}, then: '<=10',
    else: { $cond: { if: { $gte: ['$depth', 50]}, then: '+50',
      else: { $cond: { if: { $lte: ['$depth', 20]}, then: '<20',
        else: '<=50' }}}
    }
  }
}
```



podemos crear una serie nueva:

