2024 - 2025

ACTIVIDAD1

Máster en Big Data y Data Science

01MBID – Fundamentos de la Tecnología de Big Data

Nombre: Gonzalo Antonio Delgado Rubio

Fecha: 19-05-2024

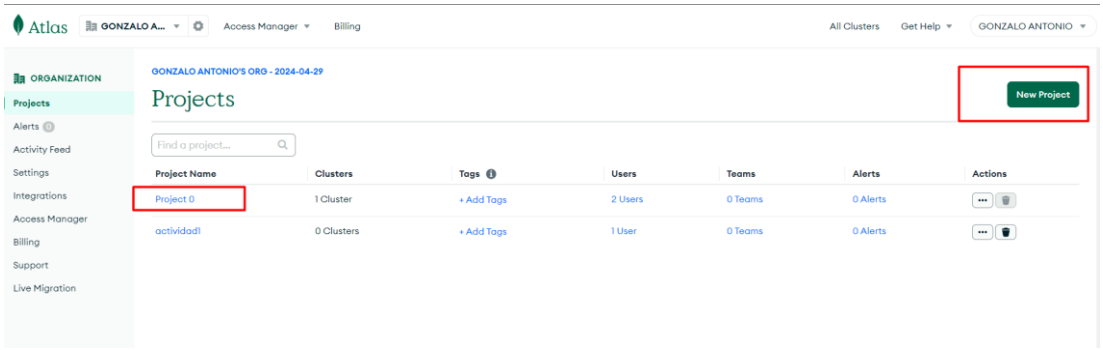
viu | Universidad Internacional de Valencia

Contenido

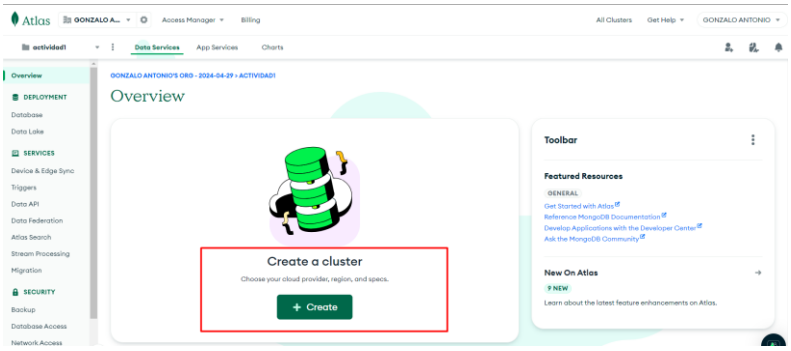
1. Creación de BD MongoDB

Utilizar el servicio cloud MongoDB Atlas

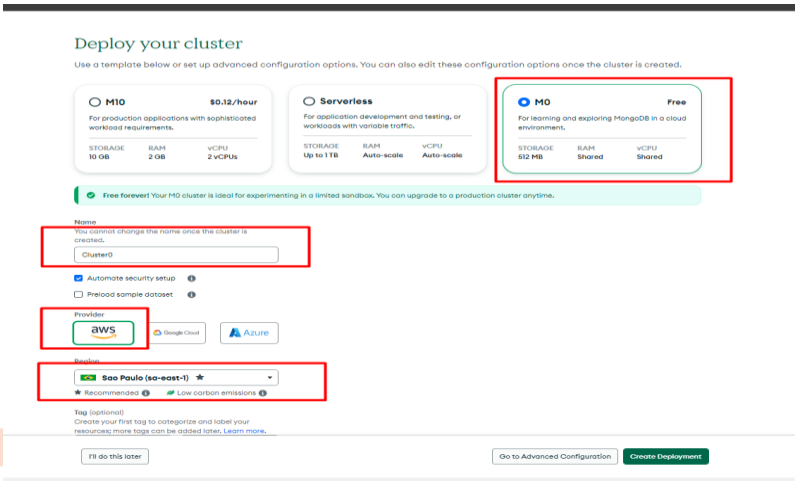
a) Ingresar a mongo DB Atlas y crearse una cuenta (link)
b) Una vez creada la cuenta, procederemos a crear un proyecto.



c) Con nuestro proyecto ya creado, procederemos a crear un cluster.



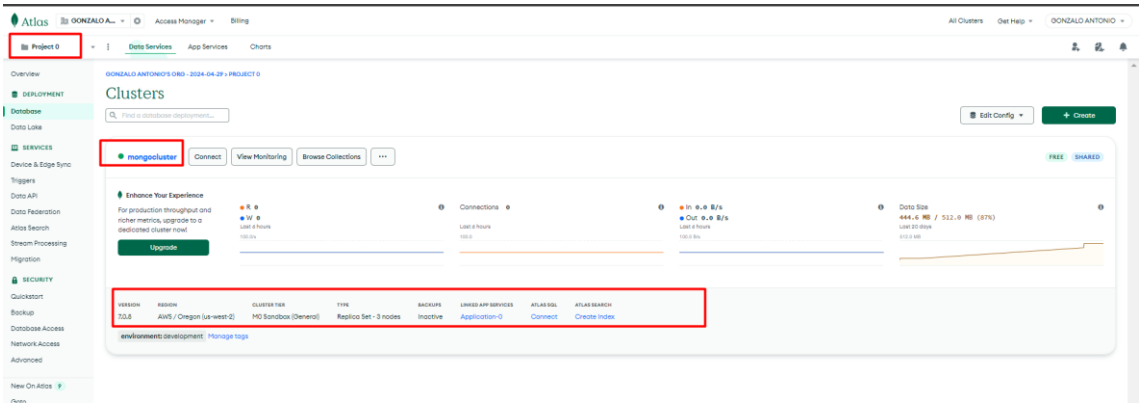
d) Podremos dar las siguientes configuraciones a nuestro cluster



e) Una vez completado, se nos brindará la cadena de conexión a nuestro cluster mongo db sobre la cual estaremos trabajando.

Para la actividad, se esta empleando la siguiente cadena de conexión asociada al proyecto1 en mongo db atlas.

```
mongodb+srv://gonzalodelgador:oCepoCGGsI1bVlkU@mongocluster.46ahjqy.mongodb.net/
```

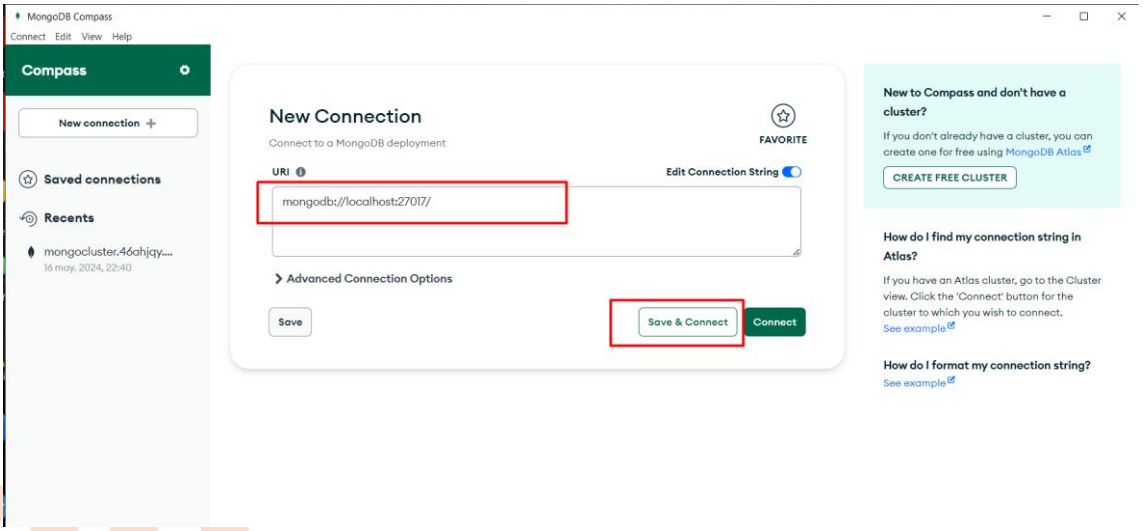


2. Creación de Colecciones

Mediante la herramienta MongoDBCompas realizar la carga de la data fuente.

Para este apartado, descargaremos y emplearemos Mongo db Compas link descarga.

Antes de iniciar, deberemos emplear la cadena de conexión (obtenida en el apartado1) para poder conectarnos a nuestro cluster.

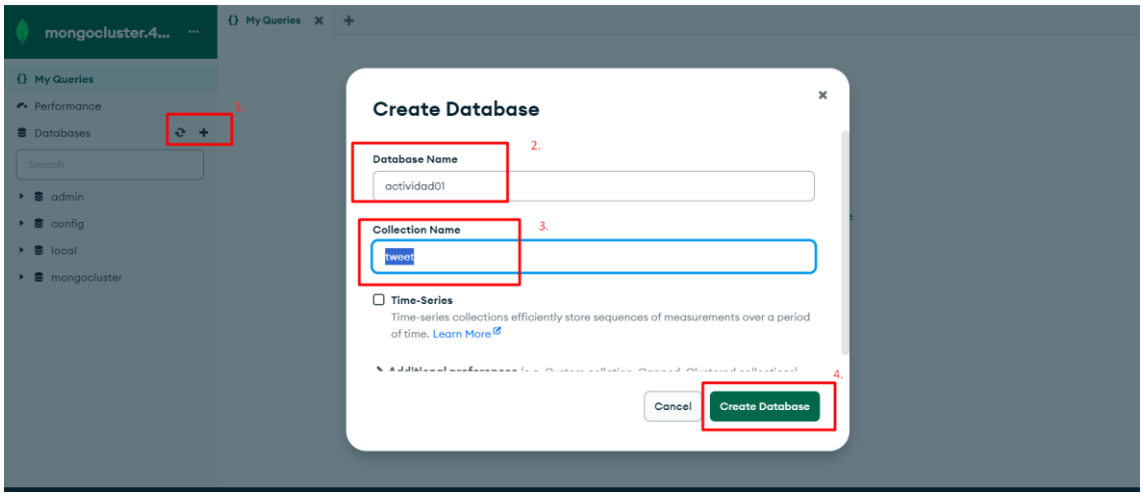


2.1 Carga de Datos Tweeter

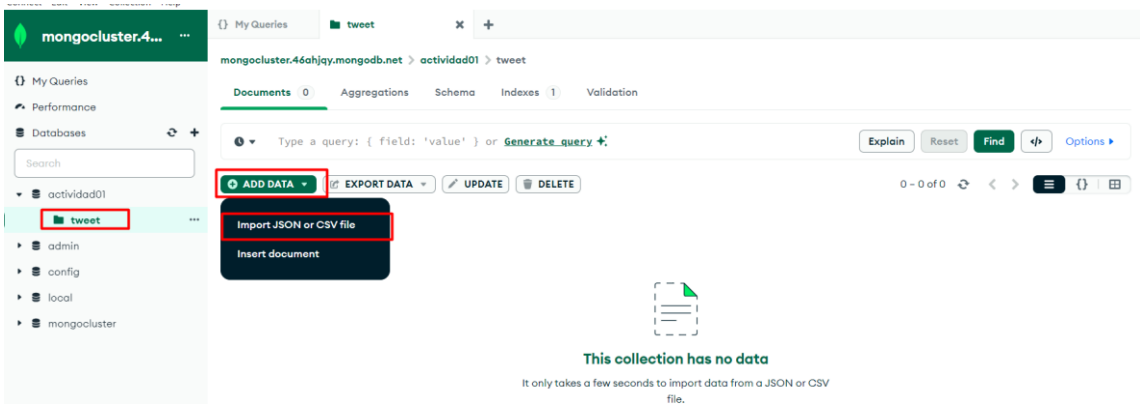
Cargar los datasets proporcionados por el profesor en formato JSON: 1) json de cuentas de twitter y 2) json de tweets.

Para esta sección realizaremos los siguientes pasos:

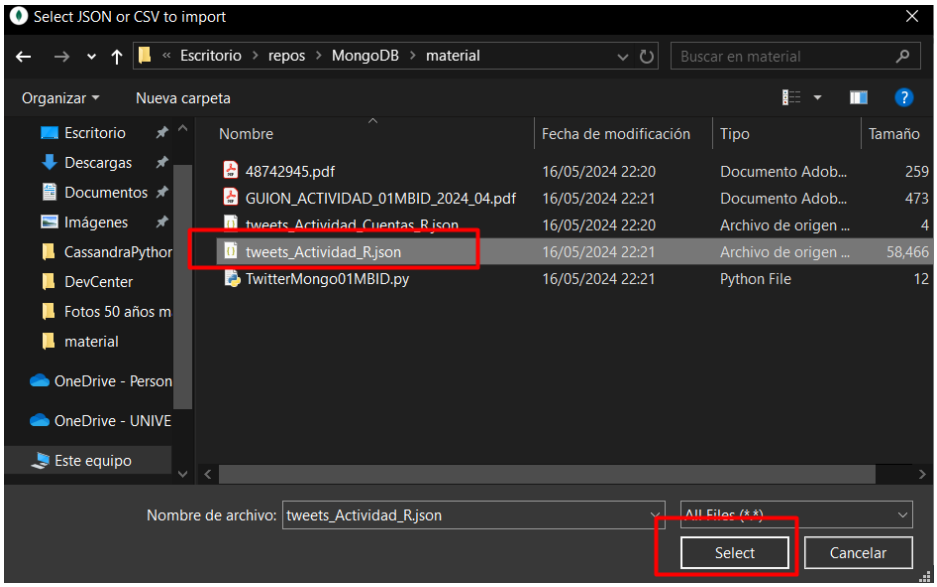
a) Sobre mongoDB Compas, crearemos una base de datos sobre la cual cargaremos nuestra colección tweet.



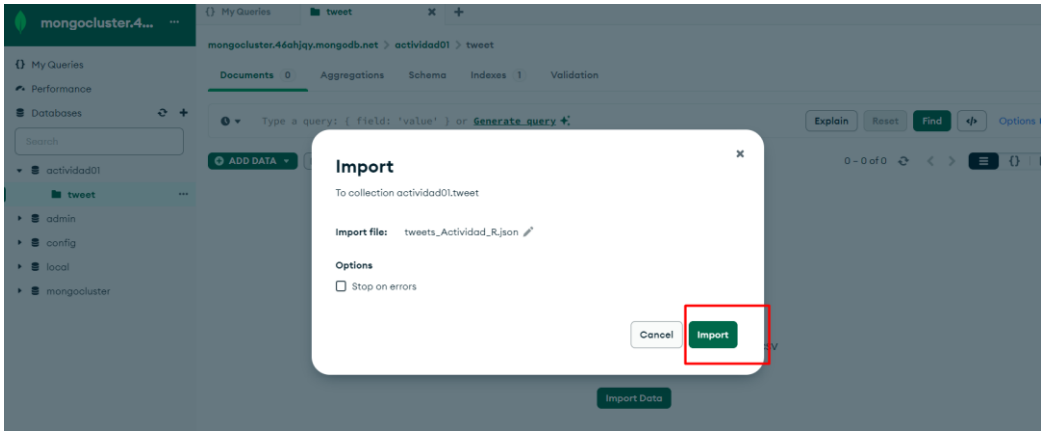
b) Ahora comenzaremos importando la data “tweets_Actividad_R.json”. Para esto debemos dar click en importar JSON.

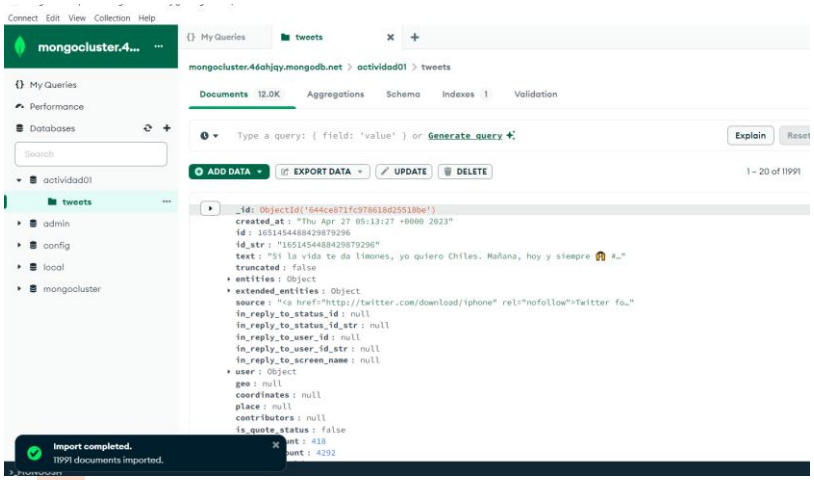


c) Ahora nos dirigiremos a donde tenemos nuestro archivo almacenado.

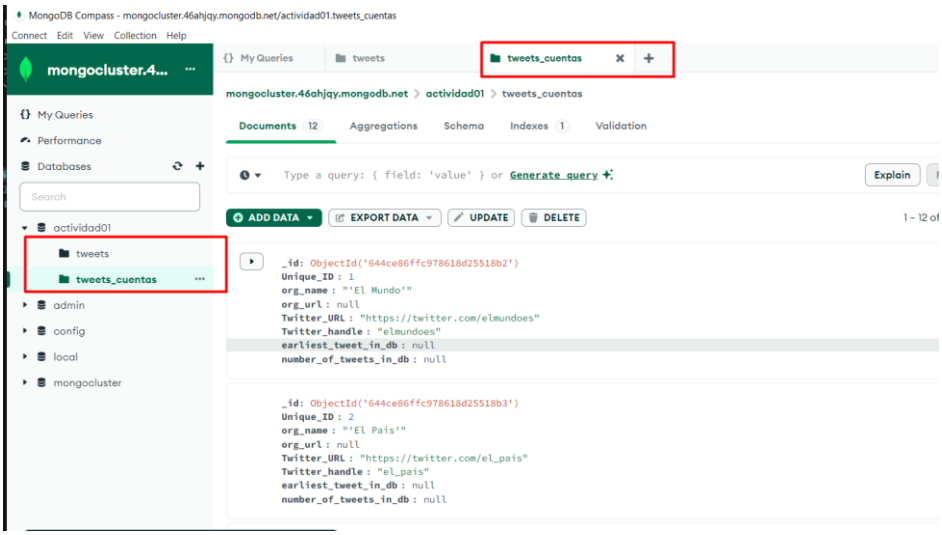


d) Daremos click a importar y esperamos a que se termine de cargar los datos





e) Repetir los mismos pasos para el archivo “tweets_Actividad_Cuentas_R.json” para lo cual hemos creado una colección llamada tweets_cuentas.

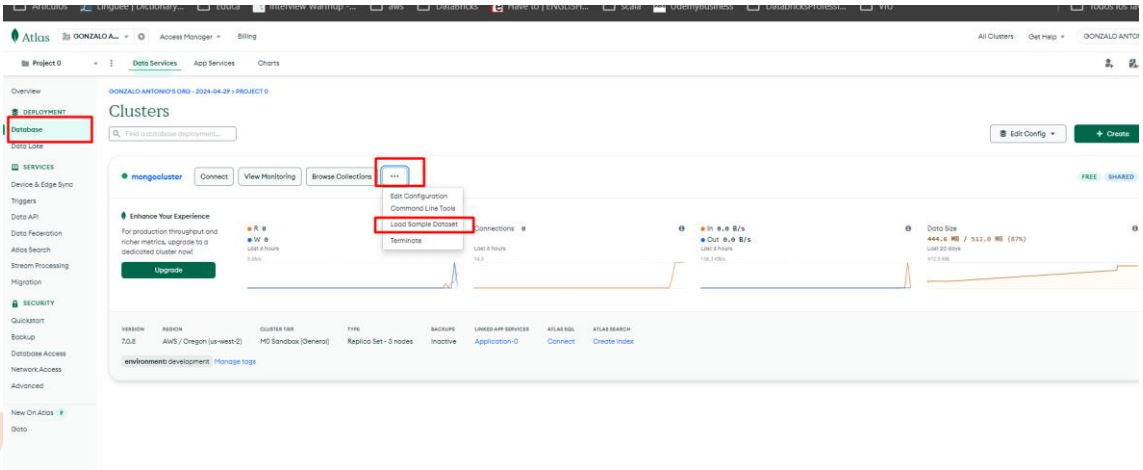


2.2 Carga de Datos Geoespecial

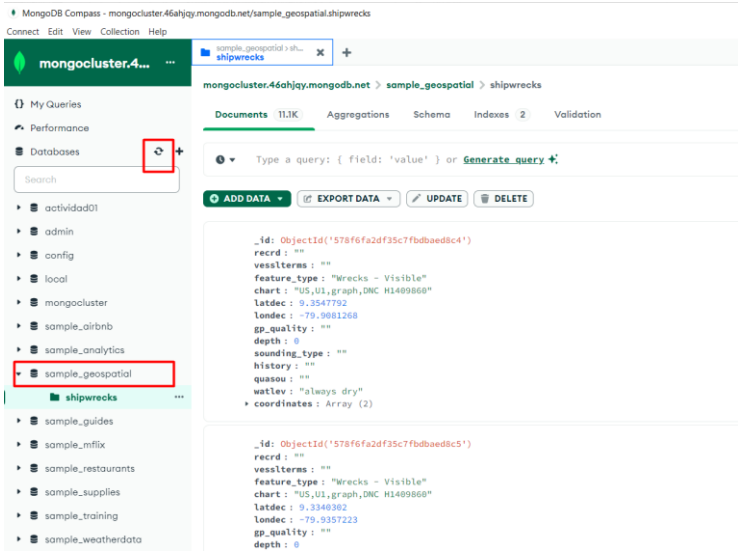
Cargar la colección de ejemplo de MongoDB Atlas cond datos de geolocalización (sample_geospatial >> shipwrecks).

Para esta parte, debemos desplegar las bases de datos ejemplo sobre nuestro cluster. Para lo cual, realizaremos el siguiente paso.

a) Dirigirse a mongo DB Atlas
b) Sobre la web dirigirnos a nuestro proyecto, apartado Database y daremos click en Load Sample Databases.



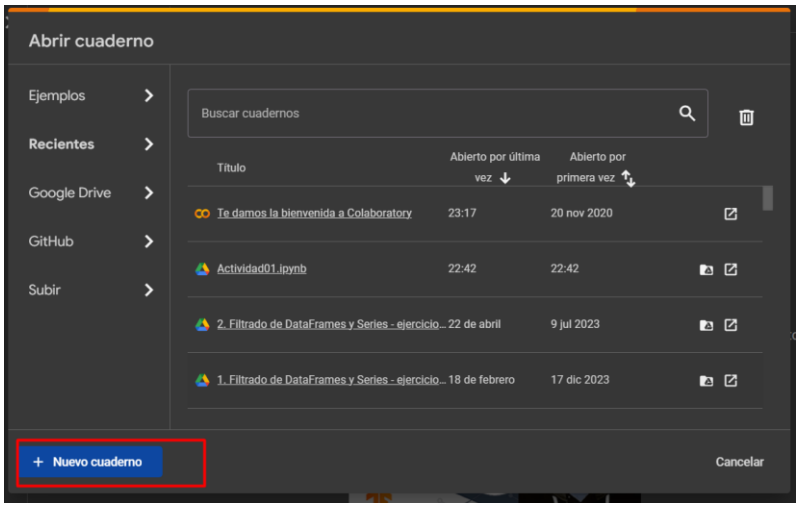
c) Luego de unos minutos, podremos ver las nuevas bases de datos cargadas sobre la herramienta MongoDB Compass.



3. Creación de Cuenta Google Colab

Para este apartado, emplearemos una cuenta de Google ya establecida y generaremos un archivo Google colab.

a) Podemos ingresar al siguiente link https://colab.research.google.com/?hl=es
b) Sobre el link crearemos un nuevo cuaderno de Google donde empezaremos a desarrollar las tareas solicitadas.



c) Debemos tener algo similar a esto. Aquí empezaremos a codear.

4. Consulta y Manipulación de Datos

Realizar los siguientes cálculos mediante consultas Python + MongoDb.

Para esto primero debemos conectarnos a la bd mediante el siguiente código

Link Consulta: https://drive.google.com/file/d/1r0xV-kRhBno_PPHY8p7V02FRA1CczBa0/view?usp=sharing

```python
from pymongo import MongoClient

def conexion_mongo_db(cadena_conexion:str, db_name:str):
    client = MongoClient(cadena_conexion)
    db = client[db_name]
    return db

# Usar la función para crear una conexión
db =
conexion_mongo_db(cadena_conexion='mongodb+srv://gonzalodelgador:oCepoCGGsI1bVlkU@mongocluster.46ahjqy.mongodb.net/', db_name='actividad01')
```

4.1 Añadir Campos Amigos y Tweets enviados

En la colección de twitters, añadir los campos amigos y tweets enviados: cargar los datos correspondientes mediante consulta mongodb + código python.

El siguiente método se encarga de añadir la cuenta de tweets enviados y amigos.

```python
def actualizar_tweets(db):
    # Seleccionar la colección
    collection = db['tweets']

    # Ejecutar la consulta de agregación
    results = list(collection.aggregate([
        {
            "$group": {
                "_id": "$user.screen_name",  # Agrupamos según el nombre de usuario
                "friends_count": {"$first": "$user.friends_count"},  # Tomamos el valor de friends_count del primer documento de cada usuario
                "statuses_count": {"$first": "$user.statuses_count"}  # Tomamos el valor de statuses_count del primer documento de cada usuario
            }
        }
    ]))

    # Actualizamos documento
    for result in results:
        screen_name = result["_id"]
        friends_count = result["friends_count"]
        statuses_count = result["statuses_count"]

        # Actualizar en la coleccion cuentas_tweet
        db.tweets_cuentas.update_one(
            {"Twitter_handle": screen_name},
              {
                "$set": {
                    "friends_count": friends_count,
                    "statuses_count": statuses_count
                }})
    pass
actualizar_tweets(db)
```

4.2 Calcular Nivel Frescura del tweet

En la colección de tweets, calcular la antigüedad del tweet en función de la fecha actual considerando antigüedad 0 el día de hoy y sumando +1 por cada día transcurrido. Este nuevo campo se llamará Frescura.

El siguiente código emplea la librería datetime para hacer la diferencia entre la fecha de creación del tweet y la fecha actual y poder realizar el cálculo del nivel de frescura solicitado.

```
from datetime import datetime


def calcular_nivel_frescura_tweet(db):
    """Funcion encargada de calcular el nivel de frescura para la coleccion tweets"""


    collection = db['tweets']


    # obtenemos datos de fecha sistema
    now = datetime.now()
    now_str = now.strftime('%Y-%m-%dT%H:%M:%S')


    # Definir el pipeline de agregación
    pipeline = [
        {
            '$addFields': {
                'created_at_date': {
                    '$toDate': '$created_at'  # Convertir la cadena 'created_at' a una fecha
                },
                'to_date': now_str  # Usar la fecha actual
            }
        },
        {
            '$addFields': {
                'Frescura': {
                    '$floor': [  # Redondear hacia abajo al entero más cercano
                        {
                            '$divide': [
                                {
                                    '$subtract': [
                                        {'$toDate': '$to_date'},  # Convertir la cadena 'to_date' a una fecha
                                        '$created_at_date'  # Restar la fecha de creación del tweet
                                    ]
                                },
                                1000 * 60 * 60 * 24  # Convertir la diferencia de milisegundos a días
                            ]
                        }
                    ]
                }
```

```
        }
    },
    {
        '$merge': 'tweets'  # Actualizar los documentos existentes
    }
]

# Ejecutar la operación de agregación
collection.aggregate(pipeline)
```

4.3 Calcular Nivel Madurez tweet

En la colección de tweets, calcular la antigüedad del tweet relativa con la fecha de creación de la cuenta. Considerando antigüedad 0 si fue enviado el mismo día de creación de la cuenta y sumando +1 por cada día transcurrido desde entonces en función de la fecha del tweet. Este nuevo campo se llamará Madurez.

De igual forma, el código muestra los días de diferencia entre la fecha de creación de la cuenta y la fecha de creación del tweet para obtener el nivel de madurez solicitado.

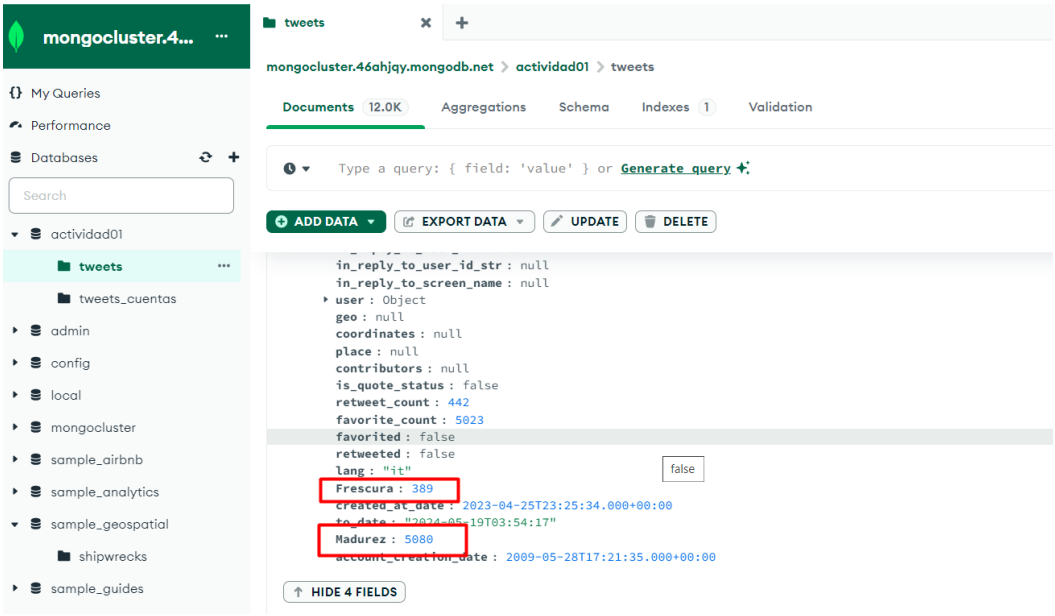
```python
def calcular_nivel_madurez_tweet(db):
    """Funcion que se encarga de calcular el nivel de madurez de los tweets"""

    # 1. seleccionamos colección
    collection = db['tweets']

    # Creamos pipeline de agregación
    pipeline = [
        {
            '$addFields': {
                'created_at_date': {
                    '$toDate': '$created_at'  # Convertir la cadena 'created_at' a una fecha
                },
                'account_creation_date': {
                    '$toDate': '$user.created_at'  # Convertir la cadena 'user.created_at' a una fecha
                }
            }
        },
        {
            '$addFields': {
                'Madurez': {
                    '$floor': [  # Redondear hacia abajo al entero más cercano
                        {
                            '$divide': [
                                {
                                    '$subtract': [
                                        '$created_at_date',  # Fecha de creación del tweet
                                        '$account_creation_date'  # Fecha de creación de la cuenta
                                    ]
                                },
                                1000 * 60 * 60 * 24  # Convertir la diferencia de milisegundos a días
                            ]
                        }
                    ]
                }
            }
        },
        {
            '$merge': 'tweets'  # Actualizar los documentos existentes
        }
    ]

    # Ejecutar la operación de agregación
    collection.aggregate(pipeline)
```

Para hacer las validaciones de los cambios realizados se desplego código sobre pandas o también se puede visualizar sobre mongo DB Compass.

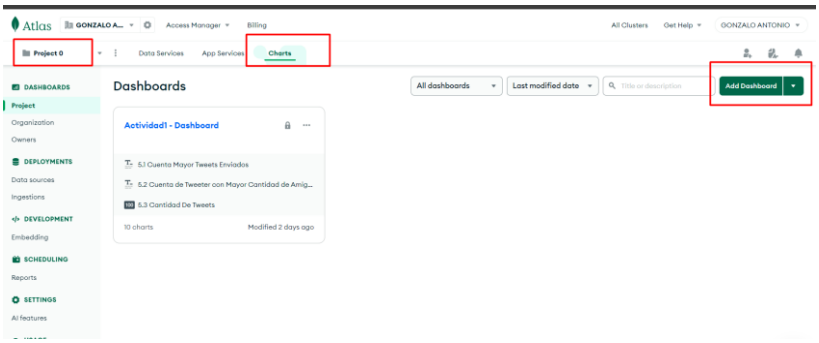


5. visualización de Datos

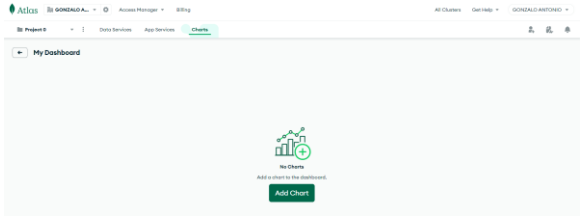
Para la siguiente sección haremos uso de MongoDB Atlas para poder realizar los gráficos solicitados. Para ellos, sobre nuestro proyecto, nos dirigiremos a la sección de charts y crearemos un nuevo dashboard.

Los datos podrán ser visualizados según el link

https://charts.mongodb.com/charts-project-0-fqvitlc/public/dashboards/6646e2b7-09fa-49f9-865c-06478ee4ff31



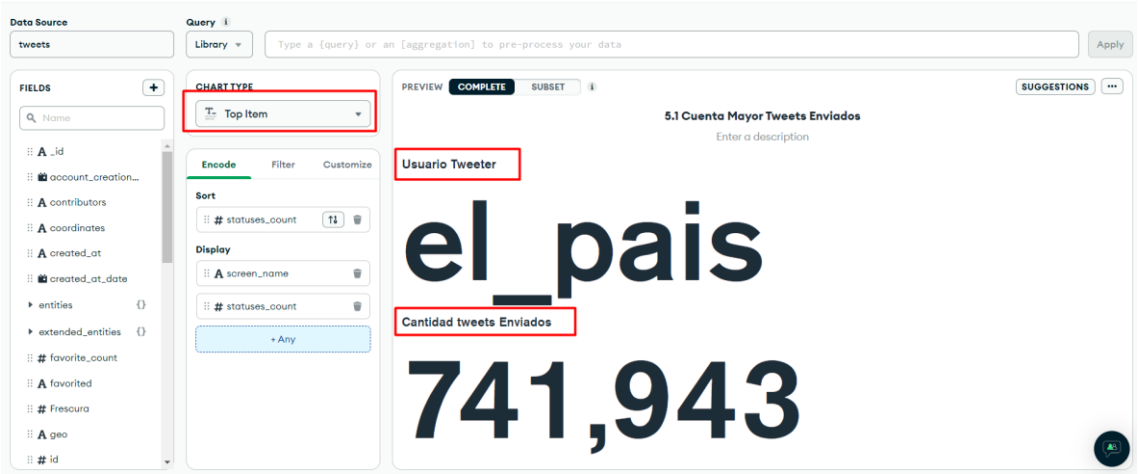
Deberemos obtener un espacio de trabajo como el siguiente



5.1 Cuenta de Tweet con Mayor cantidad de tweets enviados

Cuenta de Twitter con mayor cantidad de tweets enviados. Es un dato disponible dentro de los datos de cada tweet (Developer Plataform – Docs – Object Model).

Se emplea el tipo de gráfico top item para señalar el item con mayor cantidad de tweets enviados de acuerdo al dato statususes_count

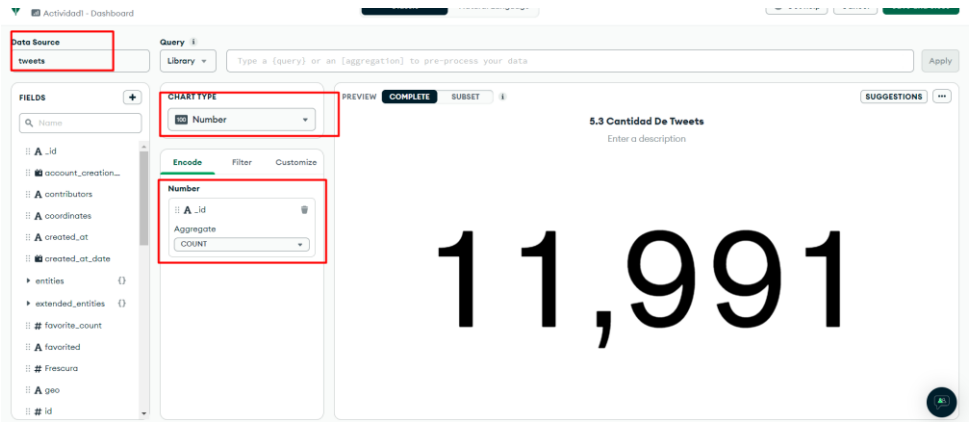


5.2 Cuenta de Tweet con Mayor cantidad de Amigos

Algo similar se emplea para la cuenta de Twitter con la mayor cantidad de amigos. Empleamos la colección tweets_cuentas.

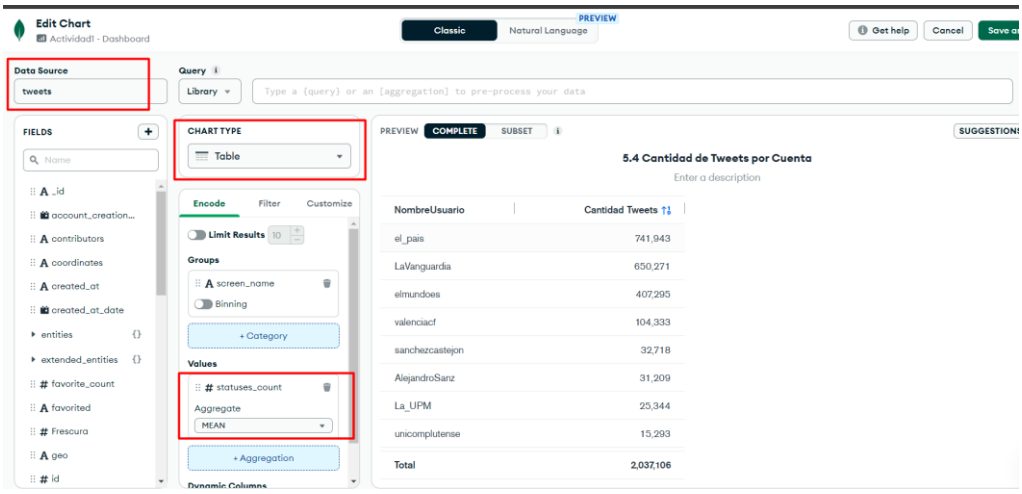
5.3 Cantidad Total de Tweets

EL numero total de tweets lo obtenemos como la cuenta del campo id de la colección tweets.

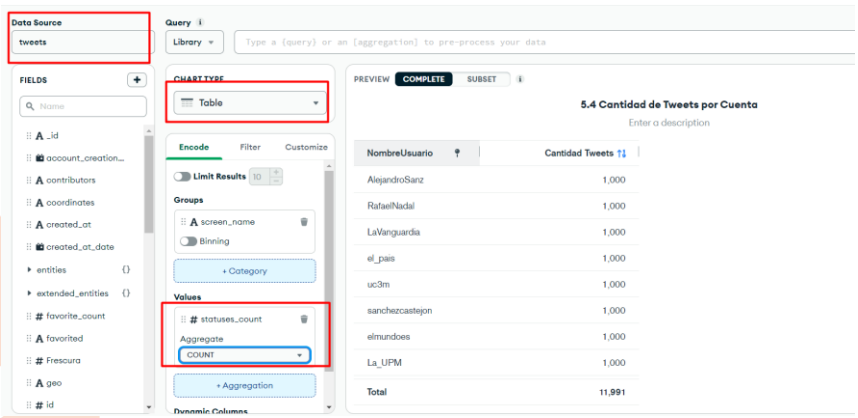


5.4 Cantidad de Tweets por cuenta

Genero una tabla a partir de la colección tweets mostrando el nombre de usuario y el promedio de la llave statususes_count que lleva la cuenta de tweets de la cuenta.

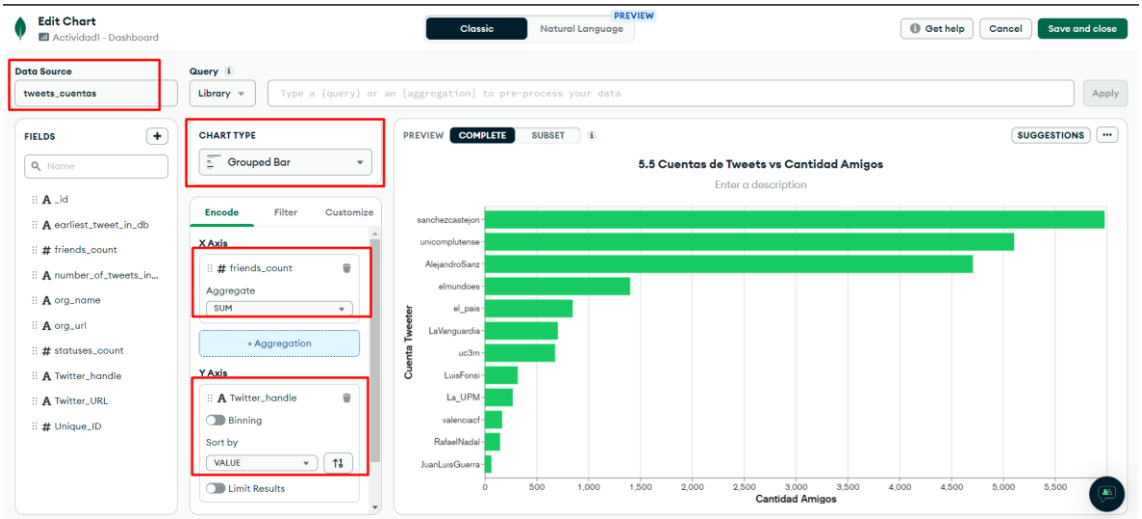


Otra forma de hacerlo sería tomar la cantidad de tweets en la colección por usuario



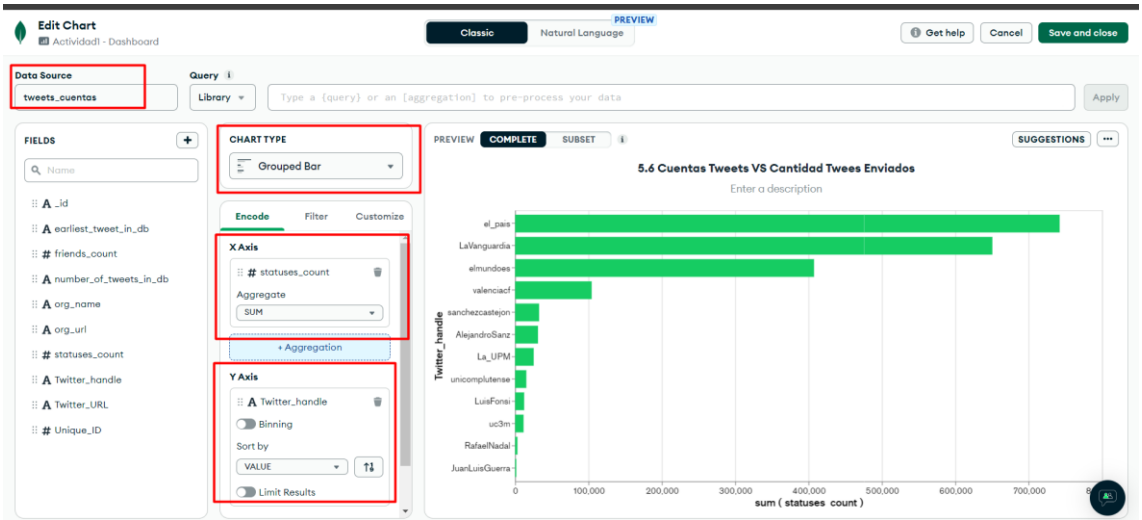
5.5 Cuenta de tweet vs cantidad de amigos

Realizaremos el grafico agrupado de cuenta de tweeter según la cantidad de amigos, dato obtenido por medio de Python.



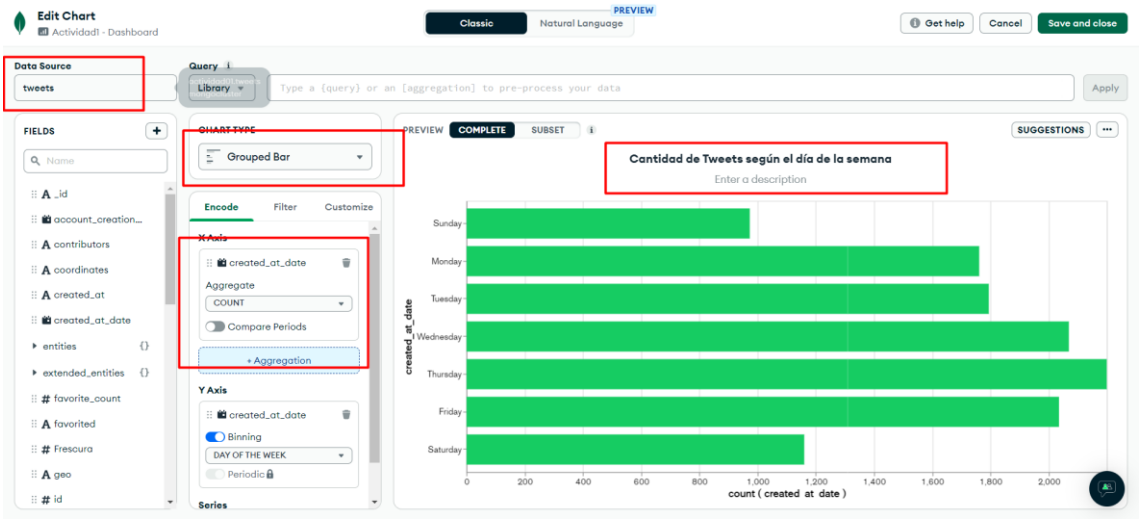
5.6 Cuenta de tweet vs tweets enviados

De forma similar hacemos el grafico agrupado de cantidad de tweets enviados agrupando el twiteer_handle y statuses_count.



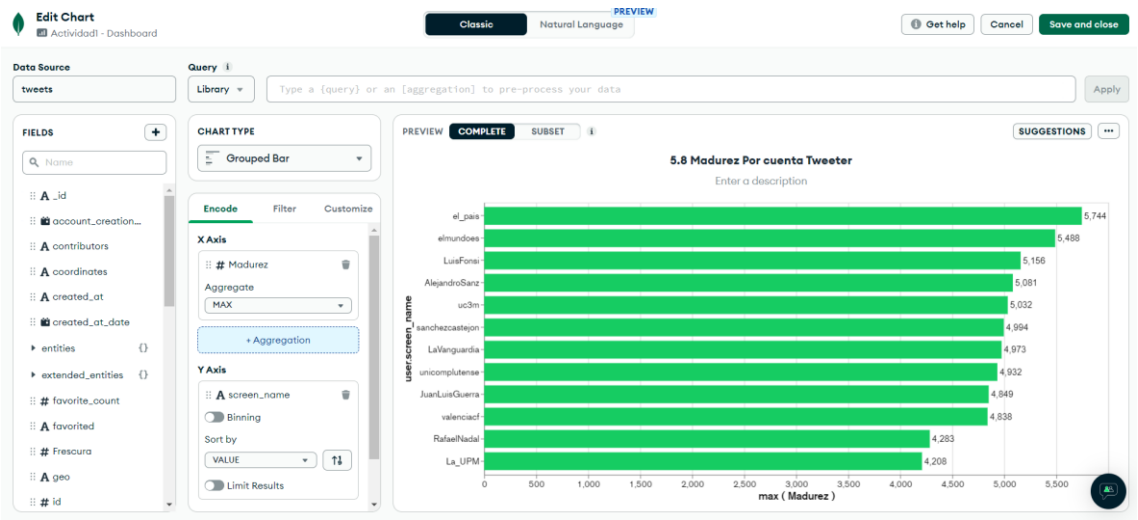
5.7 Tweets vs Días de la semana

Para obtener la cantidad de tweet según el día de semana, se debe agrupar la información según el campo created_at_date originado mediante Python.



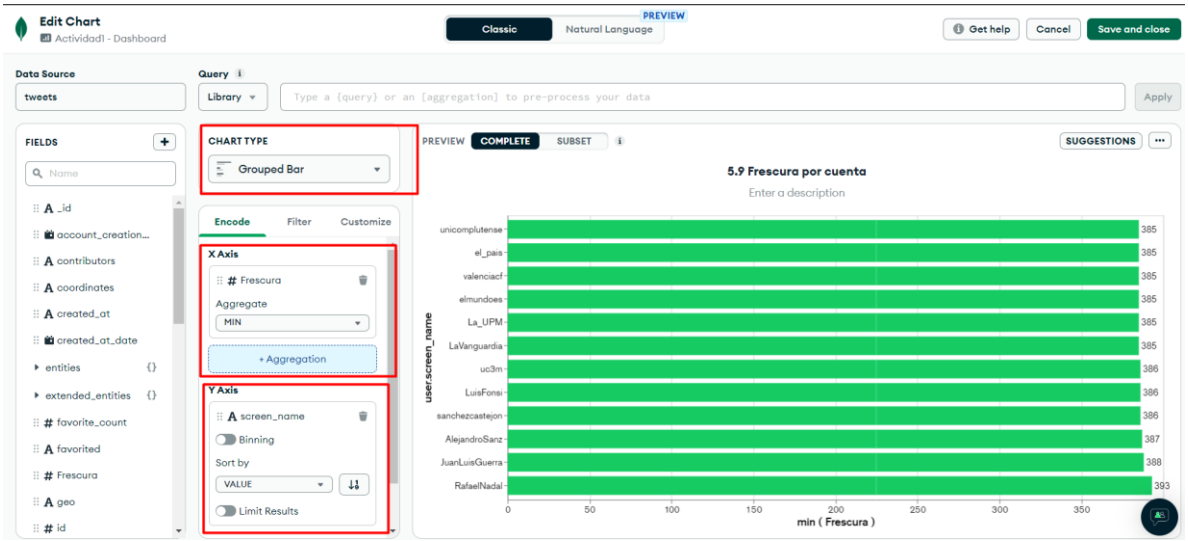
5.8 Ordenar tweets por nivel de madurez. Cuenta de tweet más madura

Obtengo la cuenta de tweet más madura agrupando por el valor máximo de nivel de madurez del tweet.



5.9 Ordenar los Tweets según su Frescura. Indicar también que cuenta de Twitter es la más Fresca.

Realizo un gráfico de barras donde agrupo la cuenta de tweet según el mínimo valor del nivel de frescura.



5.10 Mapa mundial según precios hundidos

Visualizar en el mapa mundial todos los pecios hundidos hasta 20 metros de profundidad. Pintando en verde los hundidos hasta 10 metros, amarillo hasta 15 metros y en rojo hasta 20 metros.

Se genera un gráfico GeoStatte donde mostraremos los valores de Depth según las coordenadas en el mapa y filtramos los valores entre 5 y 20 con un bin de 5.