

## BDD MULTIDIMENSIONALES

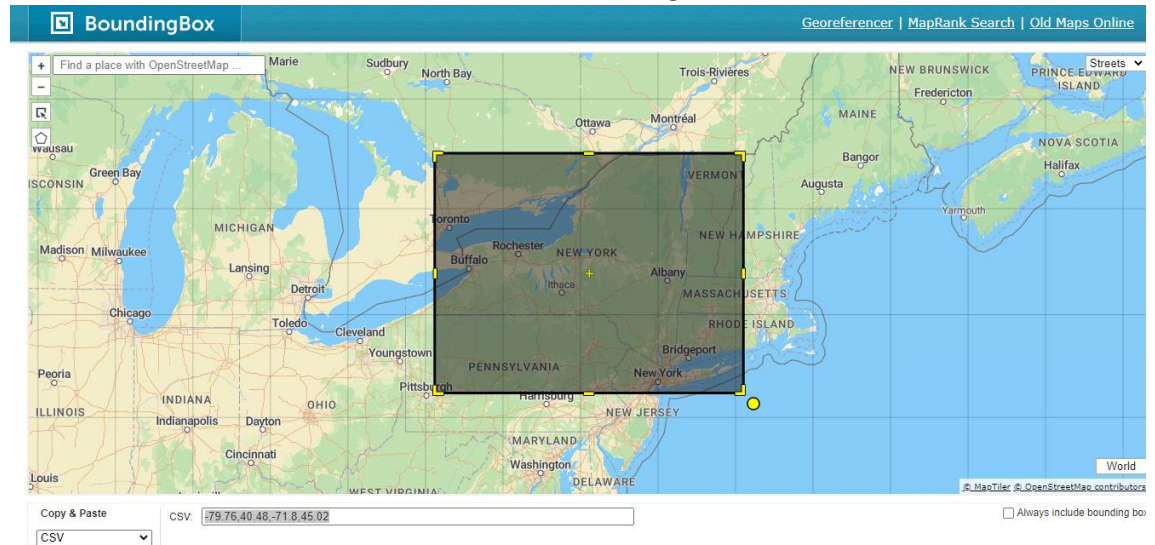
### PRUEBA 2do BIMESTRE

Chantal Morales

2020/08/18

#### 1. Tweets de New York a CouchDB


































Primero buscamos las coordenadas de NYC en Bounding Box



Escribimos el Código de extracción de tweets que es el archivo llamado "nyc.py" y ponemos las coordenadas extraídas para que filtre por geolocalización para obtener la primera base de datos de New York

```
nyc.py
9  #####
10 ckey = "174Qtm7NRcByUC8f1RUpAgLfn"
11 csecret = "Dpbw43cRn8Go2Di0YHYG9ZtCam1nHw2pcgnh670NZn4Mf0mahq"
12 atoken = "1574277428-7ANeKTU7WsNu7Vn7kh2XPigUGIjXuHn043GGRGH"
13 asecret = "K9nZrVDbEAuJ6Beeiubri0w9EAZ019VUrPFJ0drLgG28"
14 #####
15
16 class listener(StreamListener):
17
18     def on_data(self, data):
19         dictTweet = json.loads(data)
20         try:
21
22             dictTweet["_id"] = str(dictTweet['id'])
23             doc = db.save(dictTweet)
24             print ("SAVED" + str(doc) + ">" + str(data))
25         except:
26             print ("Already exists")
27             pass
28         return True
29
30     def on_error(self, status):
31         print (status)
32
33 auth = OAuthHandler(ckey, csecret)
34 auth.set_access_token(atoken, asecret)
35 twitterStream = Stream(auth, listener())
36
37 '''=====couchdb====='''
38 server = couchdb.Server('http://chantal:12345@localhost:5984/') #('http://115.146.93.184:5984/')
39 try:
40     db = server.create('newyork')
41 except:
42     db = server['newyork']
43
44 '''=====LOCATIONS====='''
45
46 twitterStream.filter(locations=[-79.76,40.48,-71.8,45.02])
47 #twitterStream.filter(track=['fifa', 'champions'])
```







[illegible]

Database name					
fifa	1.0 MB	390	No	  	
guayas	0.8 MB	286	No	  	
khe	0.6 MB	242	No	  	
libros	8.2 KB	9	No	  	
manabi	1.0 MB	361	No	  	
nba	1.0 MB	390	No	  	
newyork	275.2 KB	116	No	  	
nyc	0.9 MB	326	No	  	
pichincha	6.8 MB	2529	No	  	
prueba	0.5 KB	1	No	  	
ufc	1.0 MB	390	No	  	

```
'''=====couchdb'====='''
server = couchdb.Server('http://chantal:12345@localhost:5984/') #('http://115.146.93.184:5984/')
try:
    db = server.create('newyork2')
except:
    db = server['newyork2']

'''=====LOCATIONS====='''

#twitterStream.filter(locations=[-79.76,40.48,-71.8,45.02])
twitterStream.filter(track=['NYC'])
```

newyork	275.2 KB	116	No	  
newyork2	125.8 KB	41	No	  

Configuramos la réplica con las BDD (newyork y newyork2) generadas con los tweets. Para ello en Source seleccionamos la BDD desde la cual vamos a mandar los datos en este caso es newyork, tenemos que poner el usuario y password con el que accedemos a couchDB.

Job Configuration

**Source**

Type: Local database

Name: newyork

Authentication: Username and password

chantal

.....

**Target**







Type: Existing local database

Name: newyork2

Authentication: Username and password

http://localhost:5984/newyork http://localhost:5984/newyork2 Aug 18th, 5:23 pm Continuous Running

Réplica hecha podemos ver que aumento el número de documentos en la base de datos objetivo en este caso newyork2.

newyork	275.7 KB	116	No	  
newyork2	399.3 KB	157	No	  

## 2. TikTok

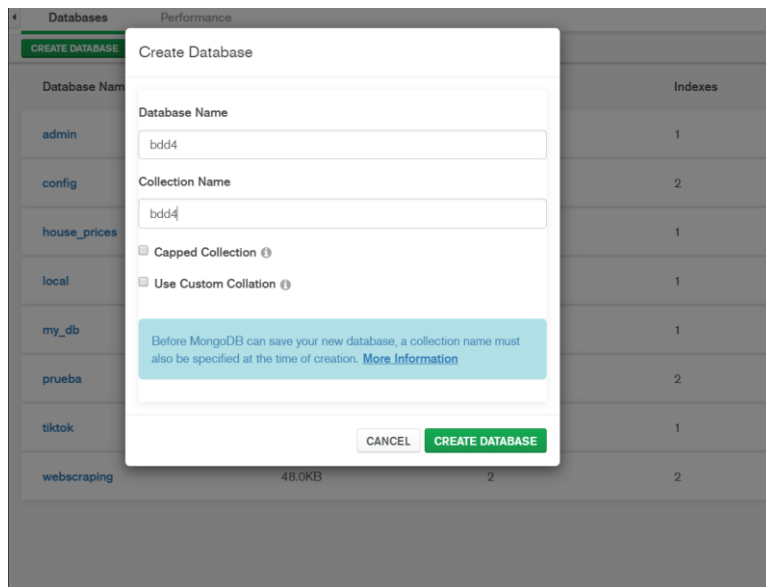
Para obtener datos de un usuario de Tiktok en este caso será de Juapa Zurita escribimos esta línea de código con la cual llamamos la librería de tiktok scraper.

```
MINGW64:/c:/Users/Chanty/Documents/5to/multidimensionales/examen
Chanty@Chanta1M MINGW64 ~/Documents/5to/multidimensionales/examen (dev)
$ tiktok-scraper user juanpazurita -n 300 -t csv
- TikTok Scraper Started
CSV path: C:\Users\Chanty\Documents\5to\multidimensionales\examen\juanpazurita_1597789666653.csv
Chanty@Chanta1M MINGW64 ~/Documents/5to/multidimensionales/examen (dev)
$ |
```

Se nos genera un archivo csv que contiene los datos del usuario, en la carpeta de la dirección en que ejecutamos el código.

Nombre	Fecha de modificación	Tipo	Id
 juanpazurita_1597789666653.csv	18/8/2020 17:27	Archivo de valores...	

Una vez hecho esto vamos a MongoDB y creamos una base de datos y una colección donde vamos a guardar estos datos. En este caso será bdd4 y la colección bdd4.



Ingresamos a BDD4 y la colección creada e importamos el documento csv que se genero de nuestro usuario de tiktok y hacemos clic en importar.

Import To Collection bdd4.bdd4

**Select File**

C:\Users\Chanty\Documents\5to\multidimensionales\examen\juanpazurita\_ BROWSE

**Select Input File Type**

JSON CSV

**Options**

Select delimiter: COMMA

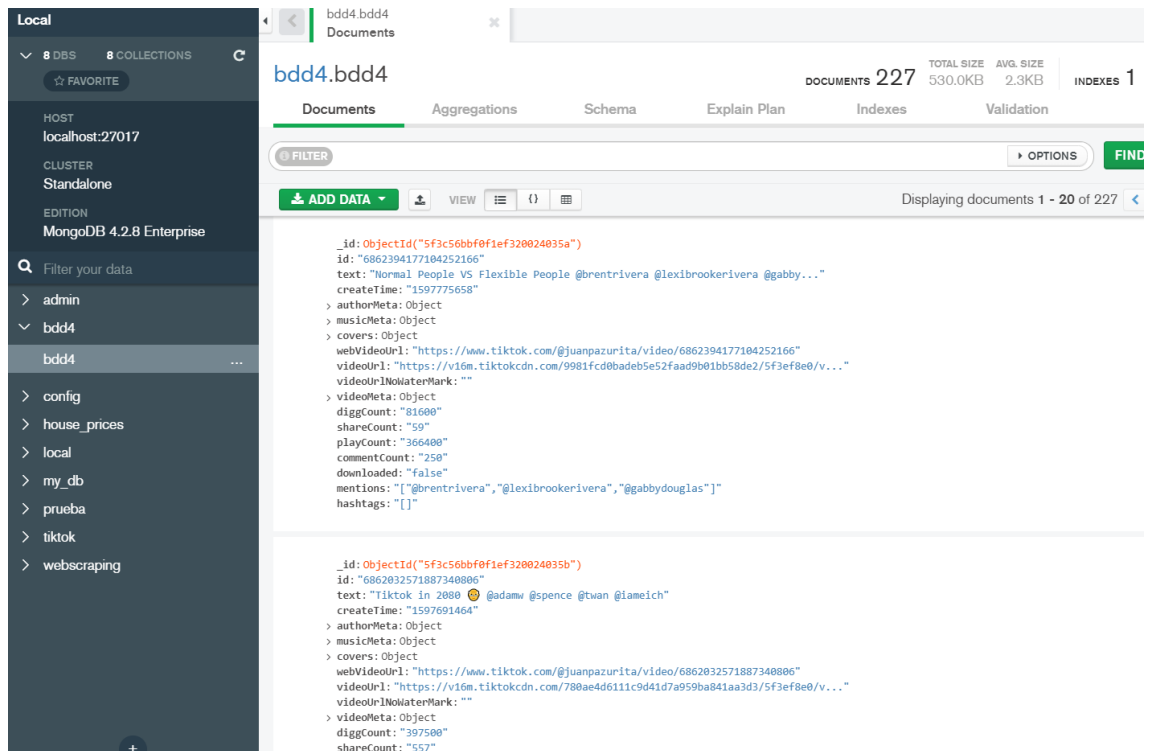
☒ Ignore empty strings

☐ Stop on errors

authorMeta.secUid String	<input checked="" type="checkbox"/> authorMeta.name String
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita
«LjABAAAAorp-3LMUWuDDnos0fg3AoZhJ12YBAFNlWuWzhB8I8Zv1dq0aux7sio6xrmsbP_d	juanpazurita

CANCEL IMPORT

Verificamos que se hayan insertado correctamente los documentos del archivo csv.



### 3. WebScraping

Para esto es necesario este script en el que utilizamos la librería de BeautifulSoup para web Scraping, especificamos la página de la que queremos obtener datos, la etiqueta y la clase y también definimos la BDD y la colección donde se guardaran estos datos en este caso bdd3 y la colección rt. (script en el archivo llamado webscraping.py)

```
webscraping.py
import requests
from bs4 import BeautifulSoup
# Import MongoClient from pymongo so we can connect to the database
from pymongo import MongoClient

if __name__ == '__main__':
    # Instantiate a client to our MongoDB instance
    db_client = MongoClient('mongodb://localhost:27017')
    bdd3 = db_client.bdd3
    rt = bdd3.rt

    response = requests.get("https://actualidad.rt.com")
    soup = BeautifulSoup(response.content, "lxml")

    post_titles = soup.find_all("a", class_="Link-root Link-isFullCard")

    extracted = []
    for post_title in post_titles:
        extracted.append({
            'title' : post_title.text,
            'link' : "https://actualidad.rt.com" + post_title['href']
        })

    # Iterate over each post. If the link does not exist in the database, it's new! Add it.
    for post in extracted:
        if db_client.bdd3.rt.find_one({'link': post['link']}) is None:
            # Let's print it out to verify that we added the new post
            print("Found a new listing at the following url: ", post['link'])
            db_client.bdd3.rt.insert(post)
```

Verificamos que se hayan insertado los datos de la página en este caso de RT news.

The screenshot shows the MongoDB Compass interface for a database named 'bdd3'. The collection 'bdd3.rt' is selected, showing 37 documents with a total size of 8.9KB and an average size of 246B. The 'Documents' tab is active, displaying a list of documents. Each document contains fields: '\_id' (ObjectId), 'title' (string), and 'link' (string). The documents are listed as follows:

- Document 1:   
\_id: ObjectId("5f3c57a89e93f5b1a7e7b316")  
title: "OMS: "La población mundial no está cerca para nada de..."  
link: "https://actualidad.rt.com/actualidad/363608-oms-poblacion-mundial-inmu..."
- Document 2:   
\_id: ObjectId("5f3c57a89e93f5b1a7e7b317")  
title: Una cepa más infecciosa del coronavirus se propaga po..."  
link: "https://actualidad.rt.com/actualidad/363549-cepa-mutacion-infecciosa-c..."
- Document 3:   
\_id: ObjectId("5f3c57a89e93f5b1a7e7b318")  
title: Cómo los 106 días de huelga de hambre de un líder map..."  
link: "https://actualidad.rt.com/actualidad/363595-huelga-hambre-mapuche-cris..."
- Document 4:   
\_id: ObjectId("5f3c57a89e93f5b1a7e7b319")  
title: Rafael Correa, sobre su candidatura a la Vicepresiden..."  
link: "https://actualidad.rt.com/actualidad/363606-correa-candidatura-vicepre..."
- Document 5:   
\_id: ObjectId("5f3c57a89e93f5b1a7e7b31a")  
title: ¿Militarización o cumplimiento del Acuerdo de Paz? El..."  
link: "https://actualidad.rt.com/actualidad/363456-militarizacion-paz-dilema-..."

#### 4. Tweet Mongo

Para esto es necesario este script en el que utilizamos la librería de tweepy para obtener datos de twitter especificamos las palabras en el filtro de búsqueda de los que queremos obtener datos la BDD y la colección donde se guardaran estos datos en este caso bdd3 y la colección trump. (script en el archivo llamado tweetTrump.py). Es necesario para esto tener credenciales de twitter developer.

```

tweetTrump.py x
1 import pymongo
2 from tweepy import Stream
3 from tweepy import OAuthHandler
4 from tweepy.streaming import StreamListener
5 import json
6
7
8 ckey = "l74Qtm7NRcByUC8f1RUUpAgLfFn"
9 csecret = "Dpbw43cRn8Go2Di0VHYG9ZtCam1nHw2pcgnh670NZn4Mf0mahq"
10 atoken = "1574277428-7ANeKTU7WsNu7Vn7kh2XPigUGIjXuHn043GGRGH"
11 asecret = "K9nZrVDbEAuJ6Beeiubri0w9EAZ019VUrPFJ0drLg28"
12
13
14 class listener(StreamListener):
15
16     def on_data(self, data):
17         dictTweet = json.loads(data)
18         try:
19             dictTweet["_id"] = str(dictTweet['id'])
20             doc = mycol.insert_one(dictTweet)
21             print("SAVED" + str(doc) + "=>" + str(data))
22         except:
23             print("Already exists")
24             pass
25         return True
26
27     def on_error(self, status):
28         print(status)
29
30
31 auth = OAuthHandler(ckey, csecret)
32 auth.set_access_token(atoken, asecret)
33 twitterStream = Stream(auth, listener())
34
35 '''Mmongo'''
36
37 myclient = pymongo.MongoClient("mongodb://localhost:27017")
38 mydb=myclient["bdd3"]
39 mycol = mydb["trump"]
40

```

Ejecutamos el script.

```

C:\Windows\py.exe
es/themes\theme1/bg.png", "profile_background_tile": false, "profile_link_color": "1DA1F2", "profile_sidebar_border_color":
":C0DEED", "profile_sidebar_fill_color": "DDEEFF", "profile_text_color": "333333", "profile_use_background_image": true, "prof
le_image_url": "http://pbs.twimg.com/profile_images/686281894502207488/SDPM2-AS_normal.jpg", "profile_image_url_https":
":https://pbs.twimg.com/profile_images/686281894502207488/SDPM2-AS_normal.jpg", "profile_banner_url": "https://pbs.
.twimg.com/profile_banners/60202093/1452457254", "default_profile": true, "default_profile_image": false, "following": nul
l, "follow_request_sent": null, "notifications": null, "geo": null, "coordinates": null, "place": null, "contributors": null, "retwe
ted_status": {"created_at": "Tue Aug 18 16:42:48 +0000 2020", "id": "1295763087996473347", "id_str": "1295763087996473347", "tex
t": "Donald Trump is one of God\u2019s finest warriors.", "source": "\u003ca href='\"http://twitter.com/download/iphone\"' rel='\"nofollow\"'\u003eTwitter for iPhone\u003c/a\u003e", "truncated": false, "in_reply_to_status_id": null, "in_reply_to_s
tatus_id_str": null, "in_reply_to_user_id": null, "in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "user": {"id":
91583544, "id_str": "91583544", "name": "Nick Adams", "screen_name": "NickAdamsinUSA", "location": "Dallas, TX", "url": "https://
/www.nickadamsusa.com/", "description": "Best selling author endorsed by @POTUS. Founder & President of @IFLAGUSA. Austr
lian by birth; American by choice.", "translator_type": "none", "protected": false, "verified": true, "followers_count": 129853,
"friends_count": 121297, "listed_count": 459, "favourites_count": 2746, "statuses_count": 4299, "created_at": "Sat Nov 21 14:39:
9 +0000 2009", "utc_offset": null, "time_zone": null, "geo_enabled": false, "lang": null, "contributors_enabled": false, "is_trans
lator": false, "profile_background_color": "C0DEED", "profile_background_image_url": "http://abs.twimg.com/images/themes\
theme1/bg.png", "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "profile
background_tile": false, "profile_link_color": "0084B4", "profile_sidebar_border_color": "C0DEED", "profile_sidebar_fill_color":
":DDEEFF", "profile_text_color": "333333", "profile_use_background_image": true, "profile_image_url": "http://pbs.twimg.co
m/profile_images/1270107259230707717/_af0YsM_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile
images/1270107259230707717/_af0YsM_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/91583
544/1591900473", "default_profile": false, "default_profile_image": false, "following": null, "follow_request_sent": null, "noti
fications": null, "geo": null, "coordinates": null, "place": null, "contributors": null, "is_quote_status": false, "quote_count": 30,
"reply_count": 51, "retweet_count": 712, "favorite_count": 3289, "entities": {"hashtags": [], "urls": [], "user_mentions": [], "symbo
ls": []}, "favorited": false, "retweeted": false, "filter_level": "low", "lang": "en", "is_quote_status": false, "quote_count": 0,
"reply_count": 0, "retweet_count": 0, "favorite_count": 0, "entities": {"hashtags": [], "urls": [], "user_mentions": [{"screen_name":
"NickAdamsinUSA", "name": "Nick Adams", "id": "91583544", "id_str": "91583544", "indices": [3, 18]}, {"screen_name":
"realDonaldTrump", "name": "Donald Trump", "id": "80759749", "id_str": "80759749", "indices": [19, 38]}], "symbols": [], "favorited": fals
e, "retweeted": false, "filter_level": "low", "lang": "en", "timestamp_ms": "1597790703698"}

```

Verificamos que se haya creado la colección con los respectivos documentos que se recolectaron mientras estaba corriendo el proceso.

Local Collections

8 DBS 8 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.2.8 Enterprise

Filter your data

admin

bdd3

rt

trump

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size
rt	37	246.3 B	8.9 KB	1	20.0 KB
trump	206	7.4 KB	1.5 MB	1	20.0 KB

bdd3.trump

DOCUMENTS 206 TOTAL SIZE 1.5MB AVG. SIZE 7.4KB INDEXES 1

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER OPTIONS FIND

ADD DATA VIEW

Displaying documents 1 - 20 of 206

```

_id: "1295854201441062912"
created_at: "Tue Aug 18 22:44:52 +0000 2020"
id: 1295854201441062912
id_str: "1295854201441062912"
text: "RT @DanWheatley4: @realDonaldTrump https://t.co/2r8WvYVkou"
source: "<a href='\"https://mobile.twitter.com\"' rel='\"nofollow\"'>Twitter Web App</a...\"
truncated: false
in_reply_to_status_id: null
in_reply_to_status_id_str: null
in_reply_to_user_id: null
in_reply_to_user_id_str: null
in_reply_to_screen_name: null
user: Object
  geo: null
  coordinates: null
  place: null
  contributors: null
  retweeted_status: Object
    quoted_status_id: 1295829572873850881
    quoted_status_id_str: "1295829572873850881"
  quoted_status: Object
    quoted_status_permalink: Object
      is_quote_status: true
      quote_count: 0
      reply_count: 0

```

SHOW 9 MORE FIELDS

```

_id: "1295854201994727430"
created_at: "Tue Aug 18 22:44:52 +0000 2020"
id: 1295854201994727430
id_str: "1295854201994727430"
text: "RT @RichardGrenell: Is @Twitter going to suspend all these accounts pu..."

```

## 5. De la BDD3 a la BDD4

Para esto exportamos la colección en formato json, hacemos clic en exportar, seleccionamos la ruta de destino y el nombre con el cual se exportará el documento.

Export Collection bdd3.rt

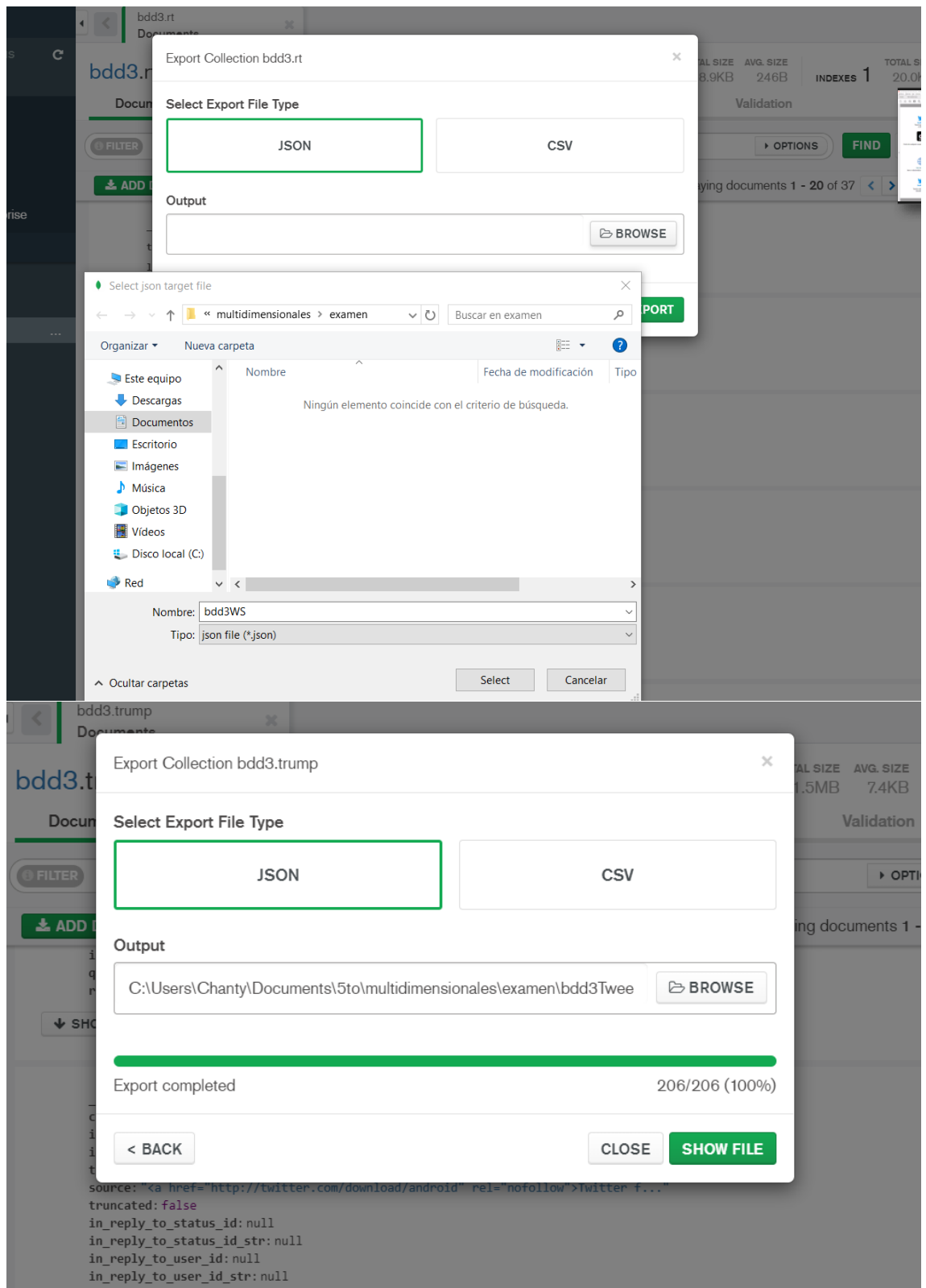
Export query with filters — 37 results (Recommended)

db.rt.find({})

Export Full Collection

CANCEL SELECT FIELDS







Hacemos clic en exportar, y verificamos que se haya generado el archivo .json de cada colección.

Documentos > 5to > multidimensionales > examen

Buscar en examen

Photo Print

Nombre	Fecha de modificación	Tipo	Ta
 bdd3Tweets.json	18/8/2020 17:50	JetBrains WebStorm	
 bdd3WS.json	18/8/2020 17:49	JetBrains WebStorm	

Una vez que obtenemos estos archivos importamos a la otra base de datos en este caso la bdd4. Para esto seleccionamos el archivo json que queremos importar en este caso se importará en la misma colección.

bdd4.

Docu

FILTER

ADD

Import To Collection bdd4.bdd4

Select File

C:\Users\Chanty\Documents\5to\multidimensionales\examen\bdd3WS.json

BROWSE

Select Input File Type

JSON

CSV

Options

☒ Ignore empty strings
 ☐ Stop on errors

Specify Fields and Types

	<input checked="" type="checkbox"/> <u>i</u> d ObjectID	<input checked="" type="checkbox"/> title String
1	5f3c57a89e93f5b1a7e7b316	OMS: "La población mundial no está cerca para nada de los nivel
2	5f3c57a89e93f5b1a7e7b317	Una cepa más infecciosa del coronavirus se propaga por Asia: ¿e
3	5f3c57a89e93f5b1a7e7b318	Cómo los 106 días de huelga de hambre de un líder mapuche han a
4	5f3c57a89e93f5b1a7e7b319	Rafael Correa, sobre su candidatura a la Vicepresidencia de Ecu
5	5f3c57a89e93f5b1a7e7b31a	¿Militarización o cumplimiento del Acuerdo de Paz? El dilema de
6	5f3c57a89e93f5b1a7e7b31b	Álvaro Uribe renuncia a su curul en el Senado colombiano en med
7	5f3c57a89e93f5b1a7e7b31c	Insurgentes en Malí arrestan al presidente y al primer ministro
8	5f3c57a89e93f5b1a7e7b31d	Un asteroide 'roza' la Tierra y la NASA no logra detectarlo a t
9	5f3c57a89e93f5b1a7e7b31e	El acuerdo entre Israel y EAU, el último incendio de Donald Tru
10	5f3c57a89e93f5b1a7e7b31f	Sharon Stone llama "asesino" a Trump y revela la desgarradora h

CANCEL

IMPORT

Import To Collection bdd4.bdd4

Select File

C:\Users\Chanty\Documents\5to\multidimensionales\examen\bdd3WS.jsor [BROWSE](#)

Select Input File Type

**JSON** CSV

Options

☒ Ignore empty strings

☐ Stop on errors

Specify Fields and Types

	<input checked="" type="checkbox"/> _id ObjectID	<input checked="" type="checkbox"/> title String
1	Sf3c57a89e93f5b1a7e7b316	OMS: "La población mundial no está cerca para nada de los nivel
2	Sf3c57a89e93f5b1a7e7b317	Una cepa más infecciosa del coronavirus se propaga por Asia: ¿e
3	Sf3c57a89e93f5b1a7e7b318	Cómo los 106 días de huelga de hambre de un líder mapuche han a
4	Sf3c57a89e93f5b1a7e7b319	Rafael Correa, sobre su candidatura a la Vicepresidencia de Ecu
5	Sf3c57a89e93f5b1a7e7b31a	¿Militarización o cumplimiento del Acuerdo de Paz? El dilema de
6	Sf3c57a89e93f5b1a7e7b31b	Álvaro Uribe renuncia a su curul en el Senado colombiano en med
7	Sf3c57a89e93f5b1a7e7b31c	Insurgentes en Malí arrestan al presidente y al primer ministro
8	Sf3c57a89e93f5b1a7e7b31d	Un asteroide 'roza' la Tierra y la NASA no logra detectarlo a t
9	Sf3c57a89e93f5b1a7e7b31e	El acuerdo entre Israel y EAU, el último incendio de Donald Tru
10	Sf3c57a89e93f5b1a7e7b31f	Sharon Stone llama "asesino" a Trump y revela la desgarradora h

Import completed 37 (100%)

Verificamos que se hayan insertado correctamente los documentos de la bdd3.

bdd4.bdd4

DOCUMENTS 470 TOTAL SIZE 1.9MB AVG. SIZE 4.1KB INDEXES 1

Documents Aggregations Schema Explain Plan Indexes Validation

[FILTER](#) [OPTIONS](#) [FIN](#)

[ADD DATA](#) [VIEW](#) [LIST](#) [JSON](#) [GRID](#)

Displaying documents 1 - 20 of 470

```

_id: "1295854201441062912"
created_at: "Tue Aug 18 22:44:52 +0000 2020"
id: 1295854201441063000
id_str: "1295854201441062912"
text: "RT @Dankwheatley4: @realDonaldTrump https://t.co/2r8VwYVkou"
source: "<a href='\"https://mobile.twitter.com\"' rel='\"nofollow\"'>Twitter Web App</a...\"
truncated: false
in_reply_to_status_id: null
in_reply_to_status_id_str: null
in_reply_to_user_id: null
in_reply_to_user_id_str: null
in_reply_to_screen_name: null
user: Object
geo: null
coordinates: null
place: null
contributors: null
retweeted_status: Object
quoted_status_id: 1295829572873851000
quoted_status_id_str: "1295829572873850881"
quoted_status: Object
quoted_status_permalink: Object
is_quote_status: true
quote_count: 0
reply_count: 0

```

[SHOW 9 MORE FIELDS](#)

```

_id: "1295854201994727430"
created_at: "Tue Aug 18 22:44:52 +0000 2020"
id: 1295854201994727400
id_str: "1295854201994727430"
text: "RT @RichardGreenall: Is @Twitter going to suspend all these accounts nu

```

## 6. Couch a Mongo

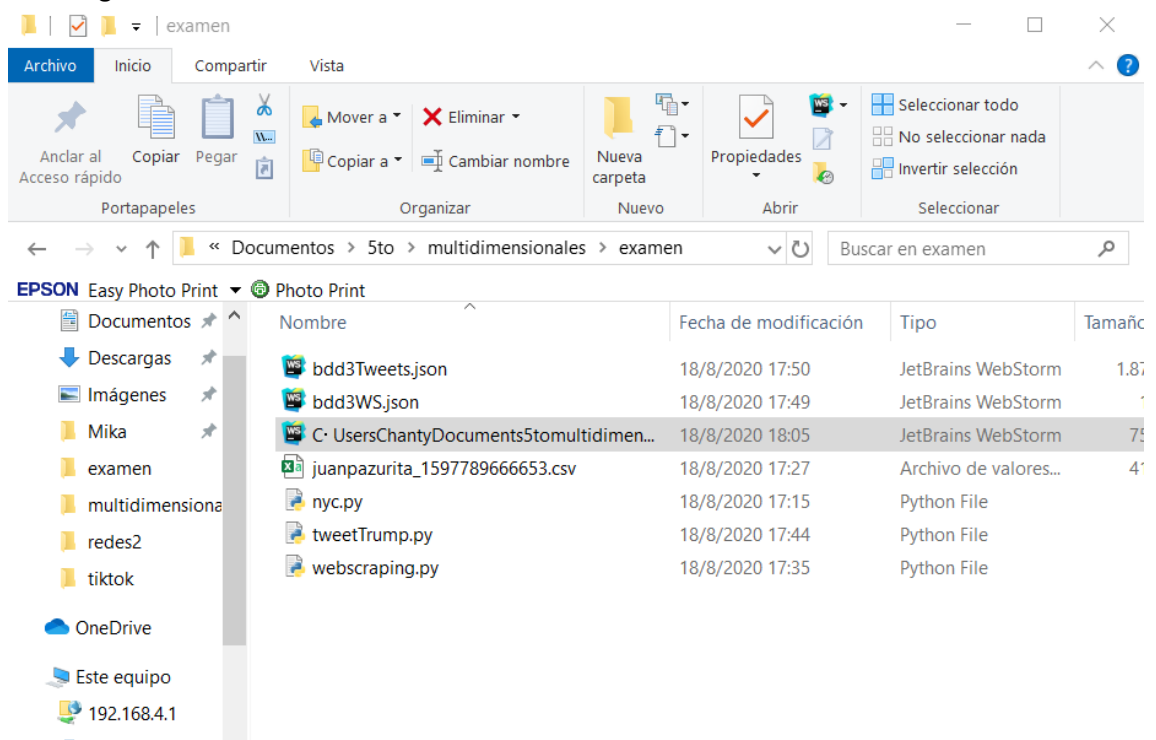
Para exportar un archivo tipo json desde CouchDB utilizamos este comando en el cual especificamos el nombre de la base de datos que queremos exportar, el usuario y password del usuario en couchDB.

```
MINGW64:/c/Users/Chanty/Documents/5to/multidimensionales/examen

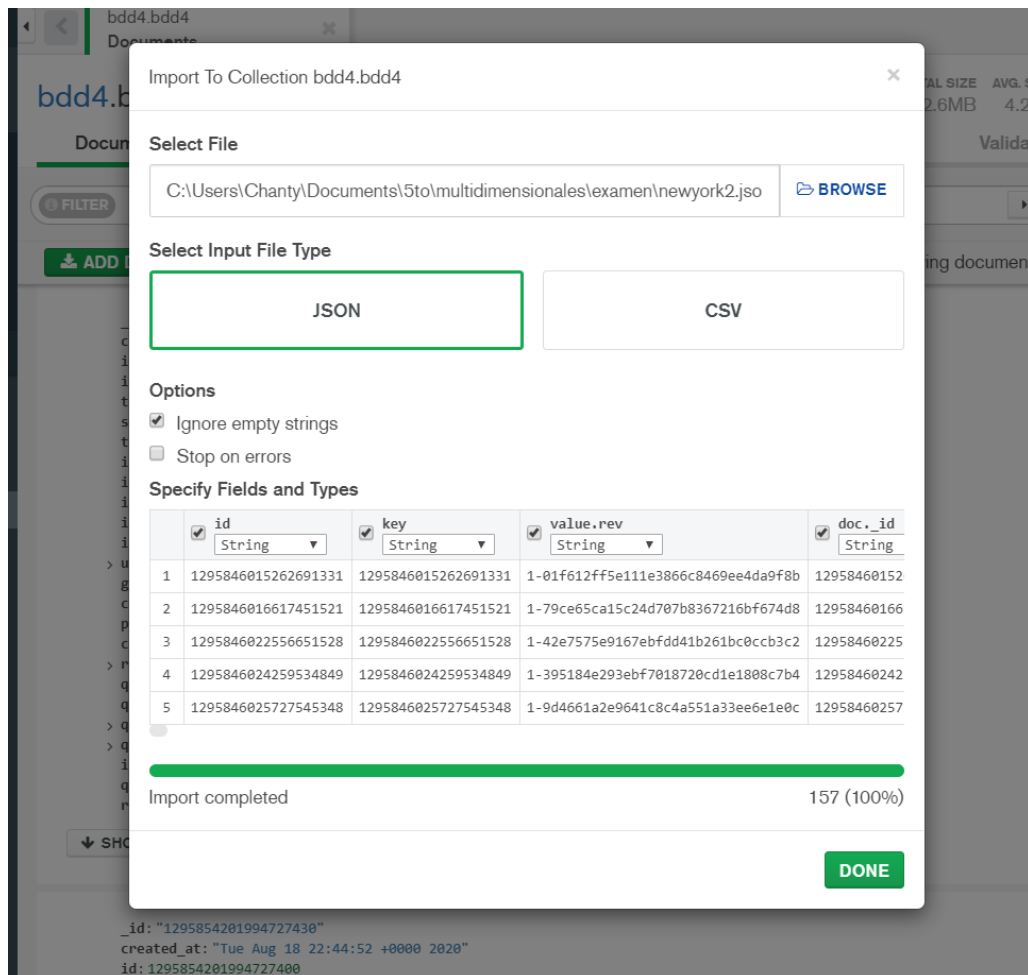
Chanty@ChantalM MINGW64 ~/Documents/5to/multidimensionales/examen (dev)
$ curl -X GET http://chantal:12345@127.0.0.1:5984/newyork2/_all_docs?include_docs=true > C:\Users\Chanty\Documents\5to\multidimensionales\examen\newyork2.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  752k    0  752k    0     0  11.1M      0  --:--:-- --:--:-- --:--:-- 11.1M

Chanty@ChantalM MINGW64 ~/Documents/5to/multidimensionales/examen (dev)
$ |
```

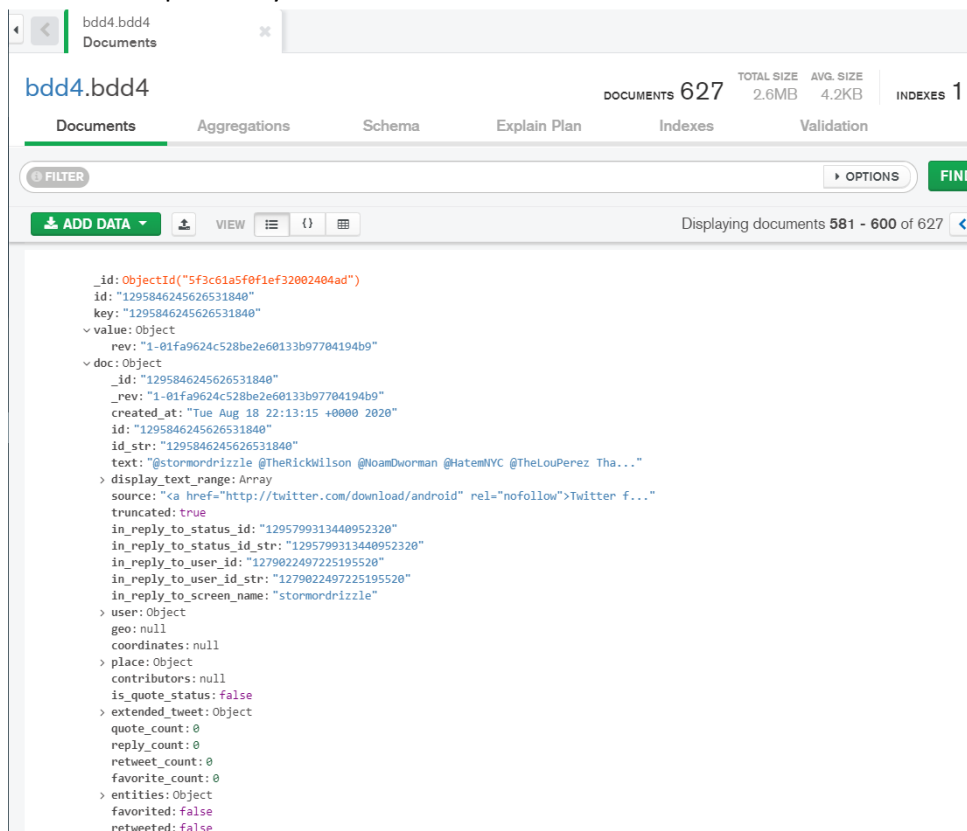
Verificamos que se nos haya creado el documento json en la carpeta destino  
Es necesario modificar el archivo json de couchdb y lo adaptemos al formato que coge en mongoDB



Importamos este archivo json a la bdd4 y la colección con los datos en común. (archivo json de couchdb llamado newyork2.json)

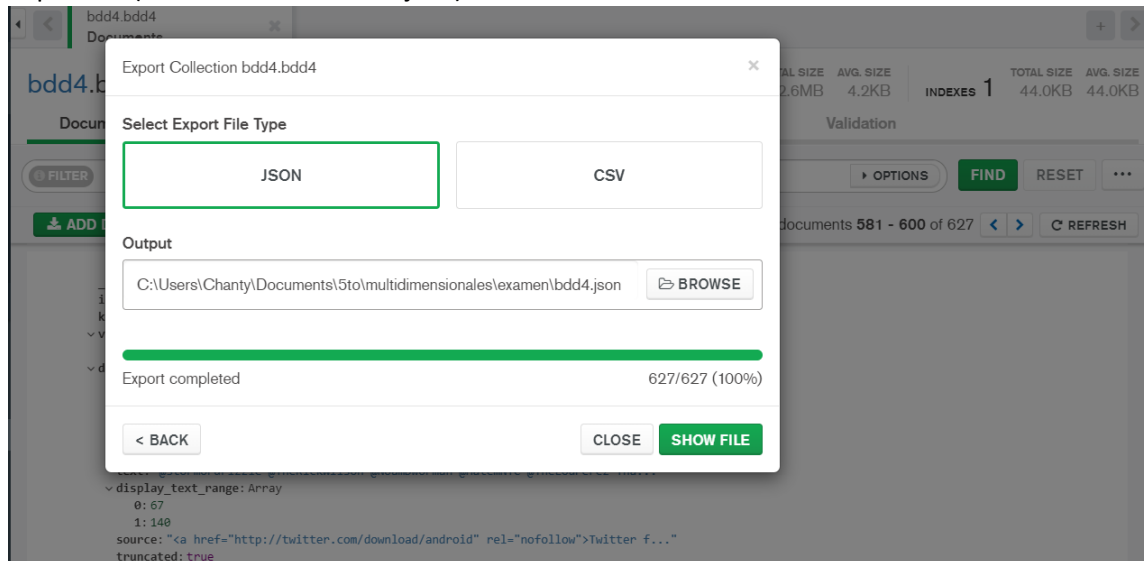


Verificamos que se hayan insertado correctamente los documentos.



## 7. Mongo a Atlas

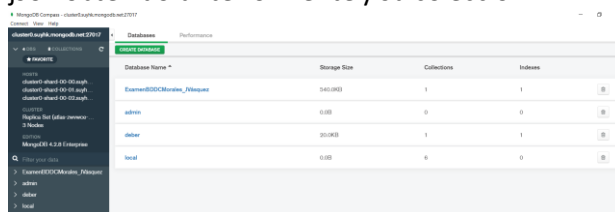
Para subir los datos al cluster es necesario primero descargar un archivo tipo json que contenga los datos finales de los 4 ejercicios realizados, en este caso seleccionamos exportar en la bdd4 en tipo json, seleccionamos la carpeta destino y hacemos clic en exportar. (archivo llamado bdd4.json)



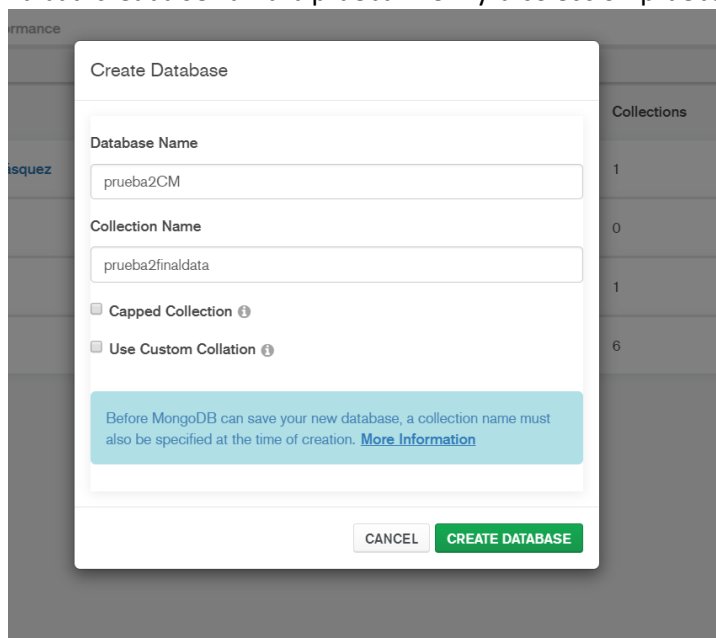
Ingresamos a nuestro cluster para esto tenemos que poner la cadena de conexión del mismo en este caso es:

mongodb+srv://cmorales:12345@cluster0.suyhk.mongodb.net/test

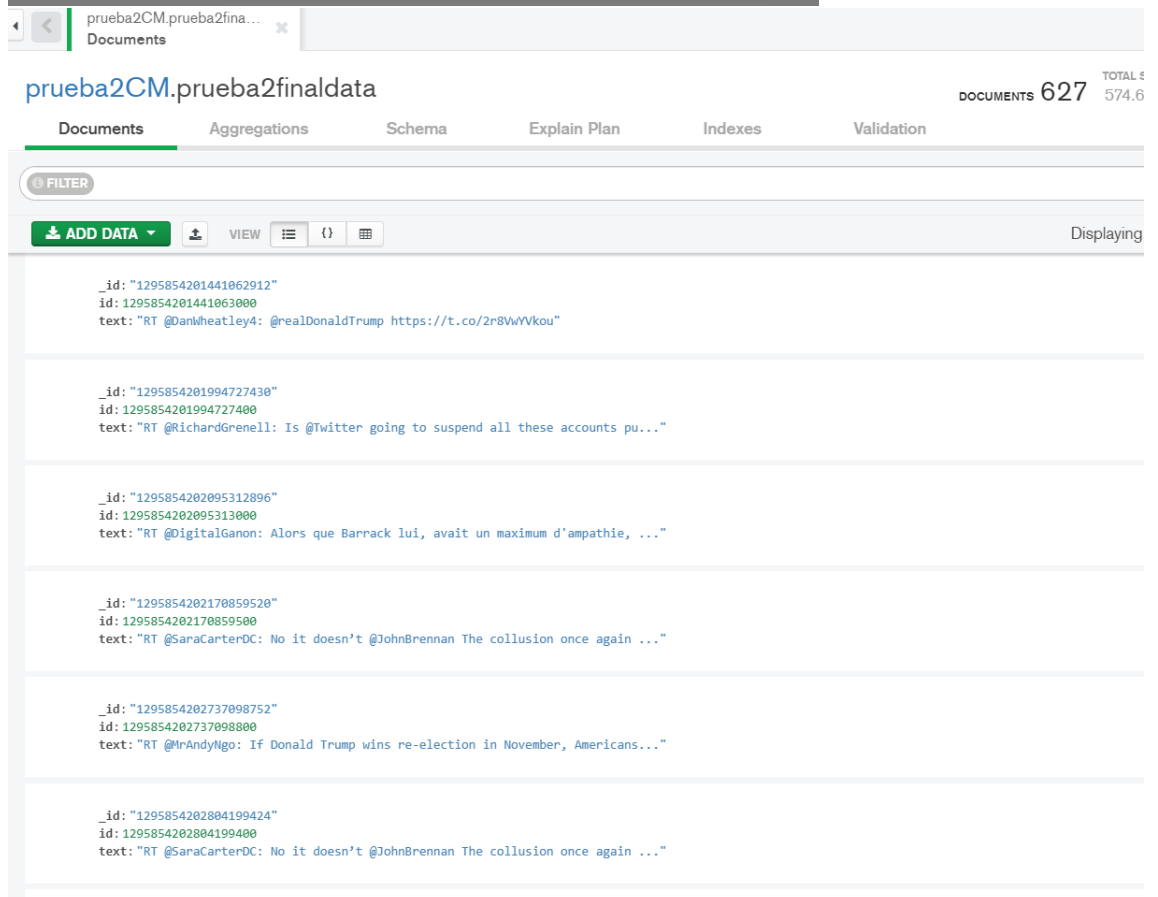
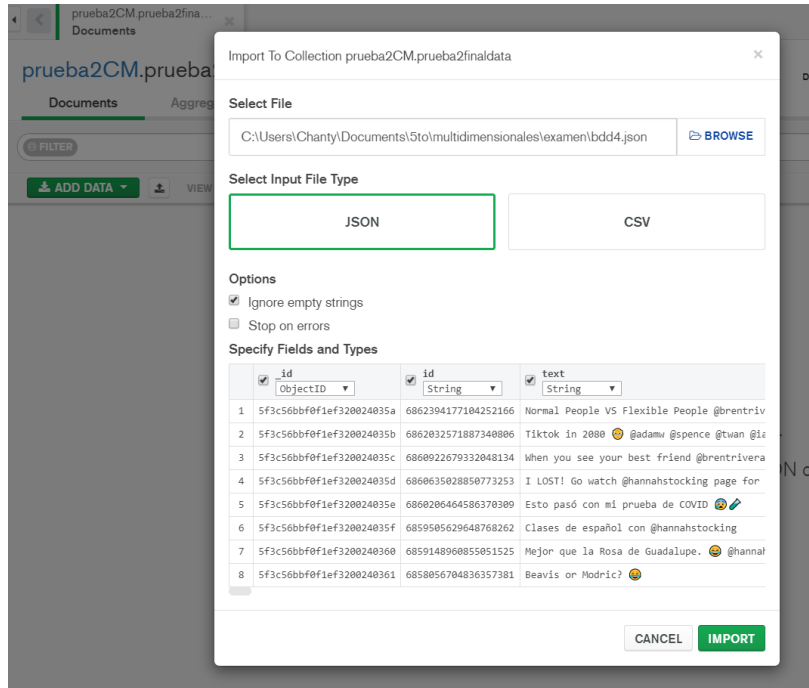
Una vez que accedemos al mismo creamos una base de datos donde importaremos el json obtenido anteriormente y su colección.



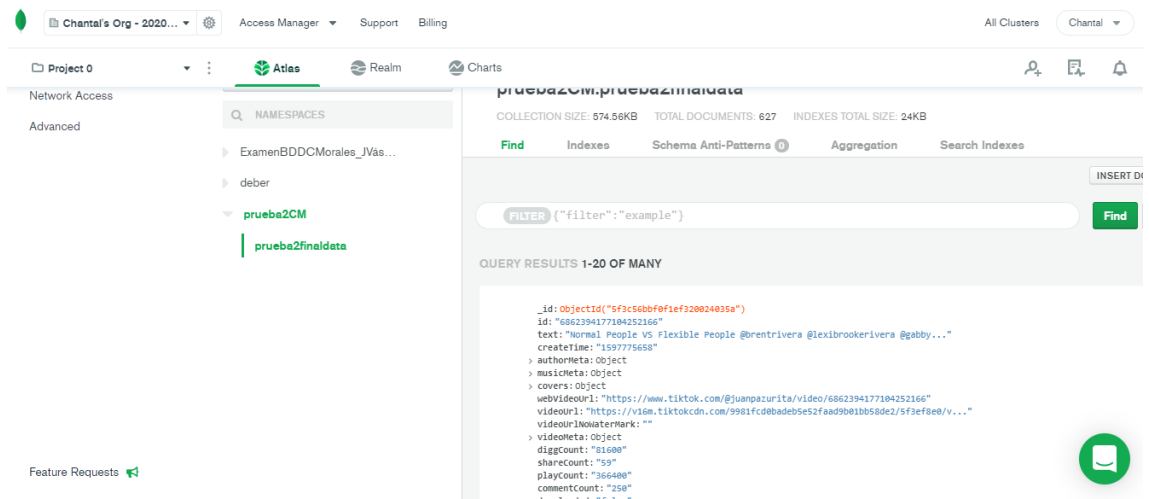
La bdd creada se llamará prueba2BCM y la colección pruebafinaldata



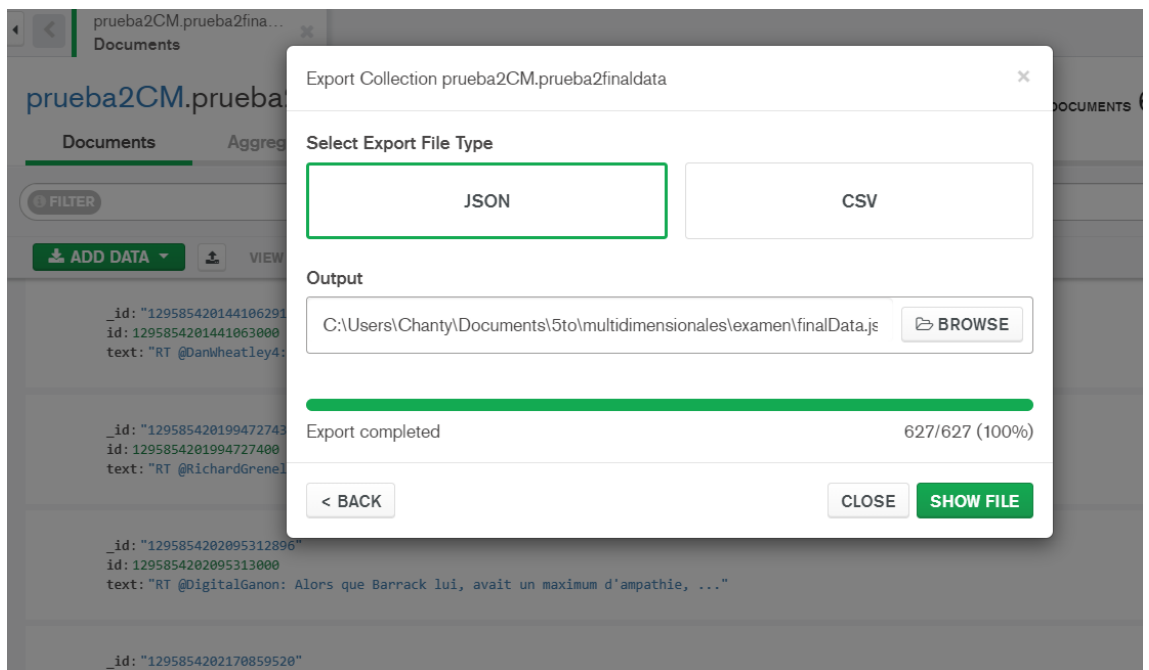
Importamos el json obtenido de la bdd4 como lo he realizado en los anteriores pasos.



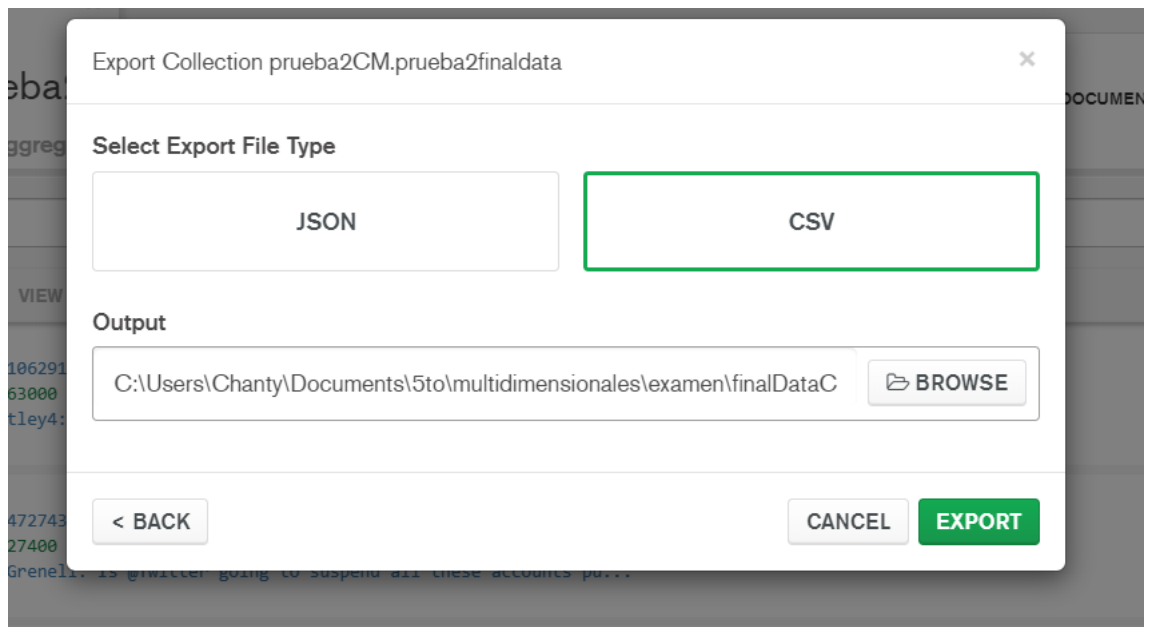
En la imagen a continuación, podemos ver la información desde el cluster en Atlas.





8. Finalmente, exportamos en formato json y formato csv la colección con los datos agrupado de las demás bdd que están en el cluster, para esto solo es necesario primero escoger formato json y poner el nombre del archivo, y de igual manera para el formato csv.







Verificamos la creación de los documentos.

 finalData.json	18/8/2020 19:05	JetBrains WebStorm
 finalDataCM.csv	18/8/2020 19:07	Archivo de valores...

#### CADENA DE CONEXIÓN:

mongodb+srv://cmorales:12345@cluster0.suyhk.mongodb.net/test