

ESCUELA POLITÉCNICA NACIONAL  
ESCUELA DE FORMACIÓN DE TECNÓLOGOS

NOMBRE: ISRAEL VÁSQUEZ

FECHA: 18/08/2020

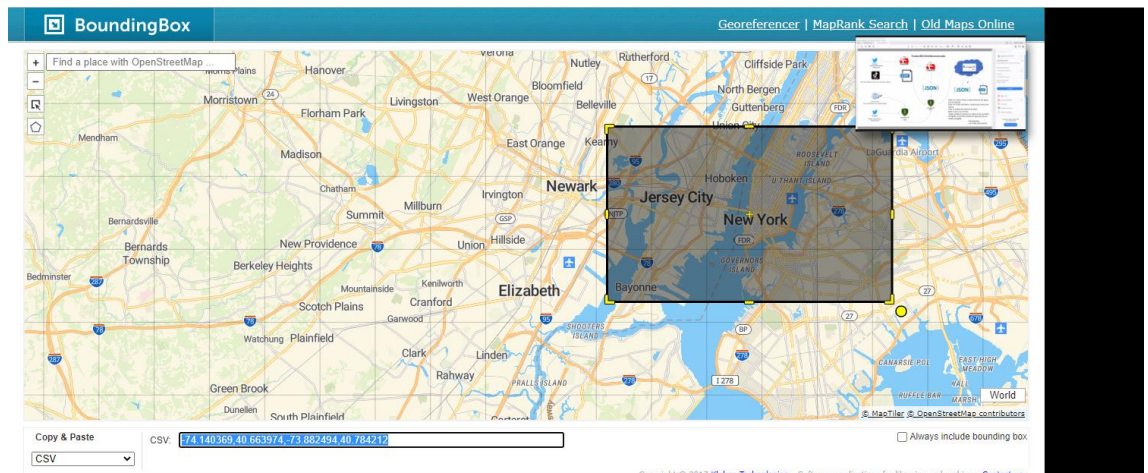
MATERIA: BASES DE DATOS MULTIDIMENSIONALES

TEMA: PRUEBA

## DESARROLLO

Captura de Datos de Tweets por coordenadas.

1. Conseguir las coordenadas para nueva york. Se las puede conseguir en bounding box



2. Añadimos las coordenadas al script

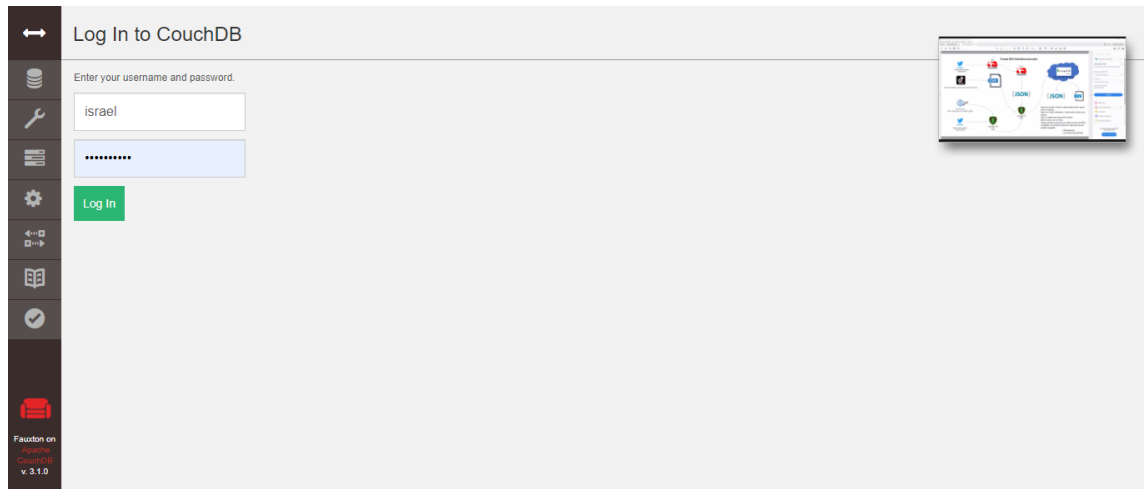
```
19 dictTweet = json.loads(data)
20 try:
21     dictTweet["_id"] = str(dictTweet['id'])
22     doc = db.save(dictTweet)
23     print ("SAVED" + str(doc) + ">" + str(data))
24 except:
25     print ("Already exists")
26     pass
27 return True
28
29 def on_error(self, status):
30     print (status)
31
32
33 auth = OAuthHandler(ckey, csecret)
34 auth.set_access_token(accessToken, accessTokenSecret)
35 twitterStream = Stream(auth, listener())
36
37 '''=====couchdb====='''
38 server = couchdb.Server('http://israel:jypam.1995@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=[-74.140369,40.663974,-73.882494,40.784212])
47 # twitterStream.filter(track=['macbook', 'macbook pro'])
```

### 3. Corremos el script desde la terminal

```
C:\Windows\System32\cmd.exe - python tweets.py

source": "\u003c href=\\"http://twitter.com/download/iphone\\" rel=\\"nofollow\\" \u003eTwitter for iPhone\u003c/a\u003e",
"truncated": true, "in_reply_to_status_id": 1295774348985409537, "in_reply_to_status_id_str": "1295774348985409537", "in_reply_to_user_id": 27934142, "in_reply_to_user_id_str": "27934142", "in_reply_to_screen_name": "slmandel", "user": {"id": 885719628, "id_str": "885719628", "name": "Phyllis", "screen_name": "PCalistro", "location": "Manhattan, NY", "url": null, "description": "NY C Attorney; Former Public Defender \u2696\ufe0f\u2696\ufe0f Married with Kids; \u2696\ufe0f Alum of SUNY Binghamton & Boston U Law #guncontrol #Votebluenomatterwho \u2696\ufe0f", "translator_type": "none", "protected": false, "verified": false, "followers_count": 2144, "friends_count": 3310, "listed_count": 9, "favourites_count": 64960, "statuses_count": 14047, "created_at": "Wed Oct 17 01:16:57 +0000 2012", "utc_offset": null, "time_zone": null, "geo_enabled": true, "lang": null, "contributors_enabled": false, "is_translator": 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": "1DA1F2", "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.com/profile_images/1267685334835494913/EFNdVbjV_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile_images/1267685334835494913/EFNdVbjV_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/885719628/1577890183", "default_profile": true, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null}, "geo": null, "coordinates": null, "place": {"id": "01a9a39529b27f36", "url": "https://api.twitter.com/1.1/geo/id/01a9a39529b27f36.json", "place_type": "city", "name": "Manhattan", "full_name": "Manhattan, NY", "country_code": "US", "country": "United States", "bounding_box": {"type": "Polygon", "coordinates": [[[-74.026675, 40.683935], [-74.026675, 40.877483], [-73.910408, 40.877483], [-73.910408, 40.683935]]]}, "attributes": {}, "contributors": null, "is_quote_status": false, "extended_tweet": {"full_text": "@slmandel Correct it was actually 2 parties, but still.... I just moved my son back to ND 10 days ago. If ND cannot control this, doesn't u2019t bode well for any college. #NotreDame \u2696\ufe0f", "display_text_range": [10, 180], "entities": {"hashtags": [{"text": "NotreDame", "indices": [168, 178]}], "urls": [], "user_mentions": [{"screen_name": "slmandel", "name": "Stewart Mandel", "id": 27934142, "id_str": "27934142", "indices": [0, 9]}, {"screen_name": "muehxrDBKw", "expanded_url": "https://twitter.com/i/web/status/1295846190195574274", "display_url": "twitter.com/i/web/status/1\u2026", "indices": [117, 140]}], "user_mentions": [{"screen_name": "slmandel", "name": "Stewart Mandel", "id": 27934142, "id_str": "27934142", "indices": [0, 9]}, {"screen_name": "muehxrDBKw", "expanded_url": "https://twitter.com/i/web/status/1295846190195574274", "display_url": "twitter.com/i/web/status/1\u2026", "indices": [117, 140]}], "symbols": [], "favorited": false, "retweeted": false, "filter_level": "low", "lang": "en", "timestamp_ms": "1597788784203"}
```

### 4. Accedemos a CouchDB



### 5. Corremos un nuevo script para crear otra base de datos para ejecutar la replica

```
16 class listener(StreamListener):
17
18     def on_data(self, data):
19         dictTweet = json.loads(data)
20         try:
21             dictTweet["id"] = str(dictTweet['id'])
22             doc = db.save(dictTweet)
23             print ("SAVED" + str(doc) + "<=> " + str(data))
24         except:
25             print ("Already exists")
26             pass
27         return True
28
29     def on_error(self, status):
30         print (status)
31
32
33 auth = OAuthHandler(ckey, csecret)
34 auth.set_access_token(accessToken, asecret)
35 twitterStream = Stream(auth, listener())
36
37 '''=====couchdb====='''
38 server = couchdb.Server('http://israel:jypam.1995@localhost:5984/') #('http://115.146.93.184:5984/')
39 try:
40     db = server.create('macbook_pro')
41 except:
42     db = server['macbook_pro']
43
44 '''=====LOCATIONS====='''
45
46 # twitterStream.filter(locations=[-74.140369,40.663974,-73.882494,40.784212])
47 twitterStream.filter(track=['macbook', 'macbook pro'])
```

```

C:\Users\jon20>cd downloads

C:\Users\jon20\Downloads>python tweets.py
SAVED('1294810626854879233', '1-a5ce18b824963a72cc40395d01d97cb0')=>{"created_at": "Sun Aug 16 01:38:04 +0000 2020", "id":
1294810626854879233, "id_str": "1294810626854879233", "text": "I hate that somehow my last messages to my aunt Rachael never
got backed up to the cloud. I\u2019ve searched my MacBook\u2026 https://t.co/oJzNvgJtSj", "source": "\u003ca href=\\"ht
tp://twitter.com/download/iphone\\" rel=\\"nofollow\\" \u003eTwitter for iPhone\u003c/\u003e", "truncated": true, "in_re
ply_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_re
ply_to_screen_name": null, "user": {"id": 911341926292557824, "id_str": "911341926292557824", "name": "Niggadamus", "screen_name":
"yruarguing", "location": null, "url": null, "description": "In my bag and out of pocket she/her | Anti-coon | black as fuck|
The Boondocks Stan", "translator_type": "none", "protected": false, "verified": false, "followers_count": 127, "friends_count": 11
5, "listed_count": 1, "favourites_count": 100893, "statuses_count": 9289, "created_at": "Fri Sep 22 21:30:21 +0000 2017", "utc_o
fset": null, "time_zone": null, "geo_enabled": true, "lang": null, "contributors_enabled": false, "is_translator": false, "profile_b
ackground_color": "F5F8FA", "profile_background_image_url": "", "profile_background_image_url_https": "", "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, "profile_image_url": "http://pbs.twimg.com/profile_i
mages/1289051129247080449/hzRSEpE3_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile_images/128
9051129247080449/hzRSEpE3_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/91134192629255782
4/1595922540", "default_profile": true, "default_profile_image": false, "following": null, "follow_request_sent": null, "notific
ations": null, "geo": null, "coordinates": null, "place": null, "contributors": null, "is_quote_status": false, "extended_tweet": {"
full_text": "I hate that somehow my last messages to my aunt Rachael never got backed up to the cloud. I\u2019ve searched
my MacBook for the old messages and I\u2019ve come up with everything except the text where she says she loves me", "dis
play_text_range": [0, 212], "entities": {"hashtags": [], "urls": [], "user_mentions": [], "symbols": []}}, "quote_count": 0, "reply_co
unt": 0, "retweet_count": 0, "favorite_count": 0, "entities": {"hashtags": [], "urls": [{"url": "https://t.co/oJzNvgJtSj", "expan
ded_url": "https://twitter.com/i/web/status/1294810626854879233", "display_url": "twitter.com/i/web/status/1\u202
6", "indices": [116, 139]}], "user_mentions": [], "symbols": []}, "favorited": false, "retweeted": false, "filter_level": "low", "lang
": "en", "timestamp_ms": "1597541884517"}

```

## 6. En CouchDB configuramos las replicas entre las bases de datos

Replication

Replicator DB Activity | \_replicate Activity

Replications must have a replication document to display in the following table.

Filter replications

Source	Target	Start Time	Type	State	Actions
http://localhost:5984/macbook_pro	http://localhost:5984/newyork	Aug 18th, 5:18 pm	Continuous	Running	[Icons]
http://localhost:5984/newyork	http://localhost:5984/macbook_pro	Aug 18th, 5:19 pm	Continuous	Running	[Icons]

New Replication

## 7. Comprobamos que contengan los mismos documentos.

Databases

Database name

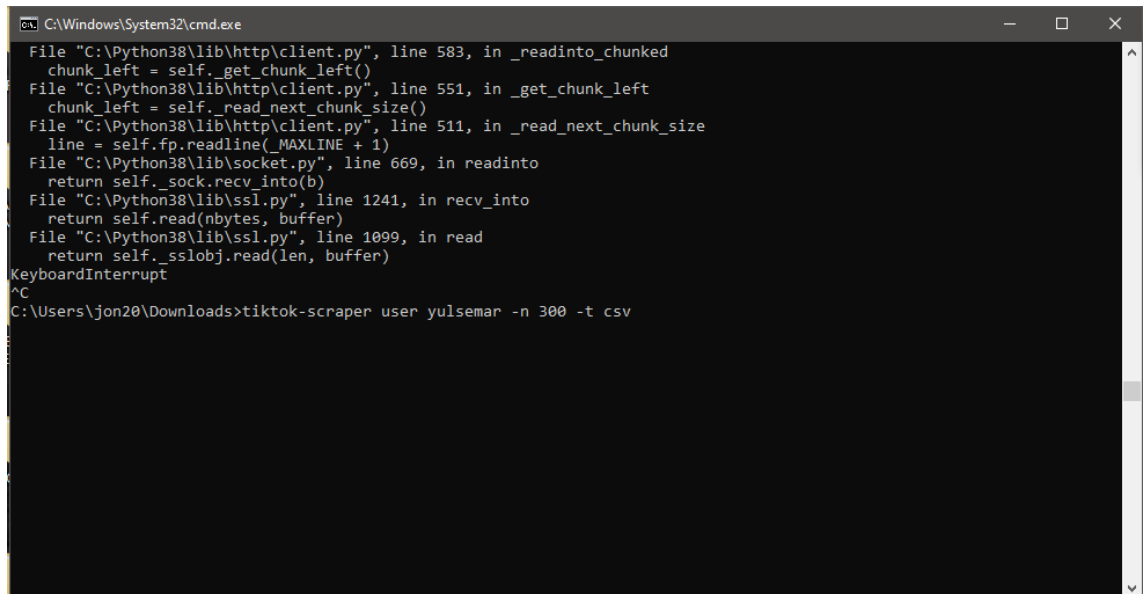
Create Database

Name	Size	# of Docs	Partitioned	Actions
_replicator	13.5 KB	3	No	[Icons]
macbook_pro	4.5 MB	1872	No	[Icons]
newyork	4.6 MB	1872	No	[Icons]

Showing 1-3 of 3 databases. Databases per page 20

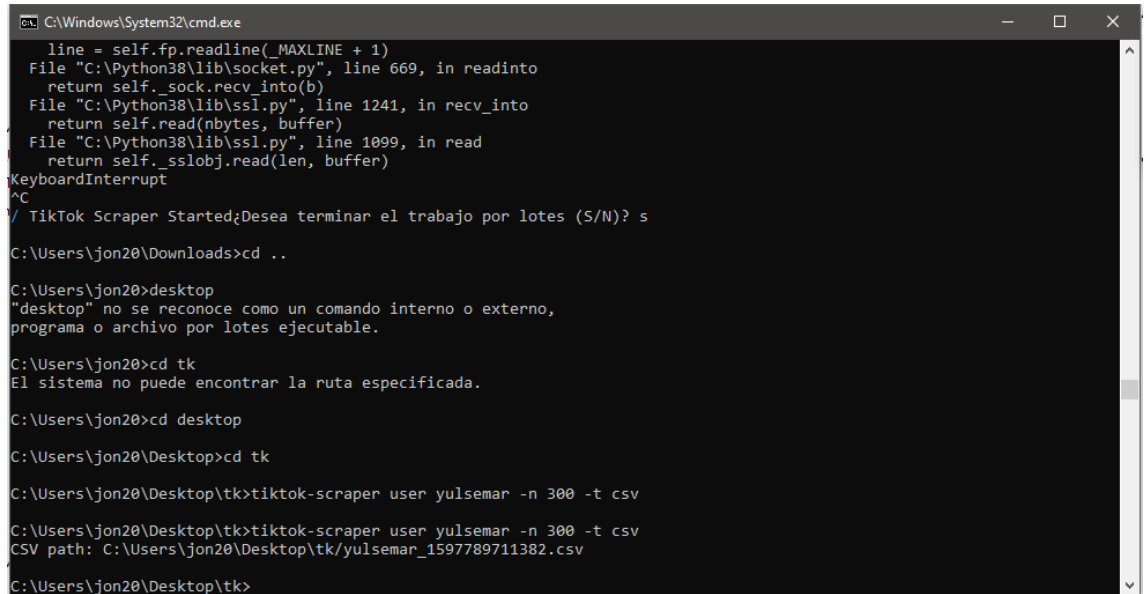
## Datos Tik Tok a MongoDB

1. Abrimos la terminal para ejecutar los comandos, así obtendremos un archivo csv de los datos del usuario yulsemear.



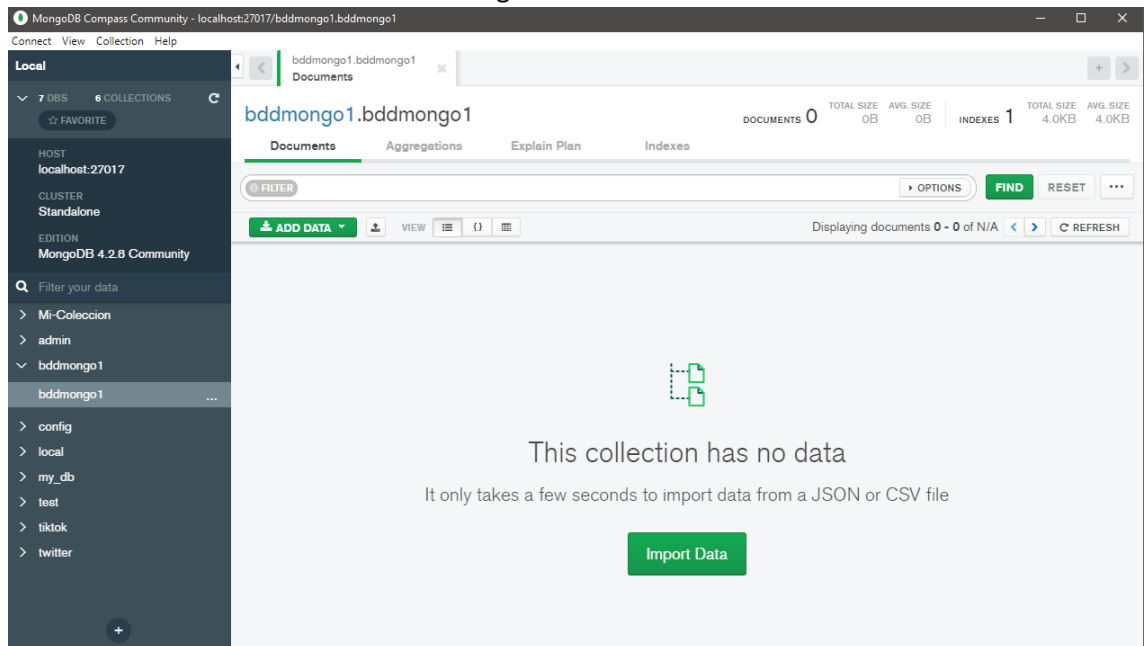
```
C:\Windows\System32\cmd.exe
File "C:\Python38\lib\http\client.py", line 583, in _readinto_chunked
  chunk_left = self._get_chunk_left()
File "C:\Python38\lib\http\client.py", line 551, in _get_chunk_left
  chunk_left = self._read_next_chunk_size()
File "C:\Python38\lib\http\client.py", line 511, in _read_next_chunk_size
  line = self.fp.readline(_MAXLINE + 1)
File "C:\Python38\lib\socket.py", line 669, in readinto
  return self._sock.recv_into(b)
File "C:\Python38\lib\ssl.py", line 1241, in recv_into
  return self.read(nbytes, buffer)
File "C:\Python38\lib\ssl.py", line 1099, in read
  return self._sslobj.read(len, buffer)
KeyboardInterrupt
^C
C:\Users\jon20\Downloads>tiktok-scraper user yulsemear -n 300 -t csv
```

2. Esto generara nuestro archivo csv.

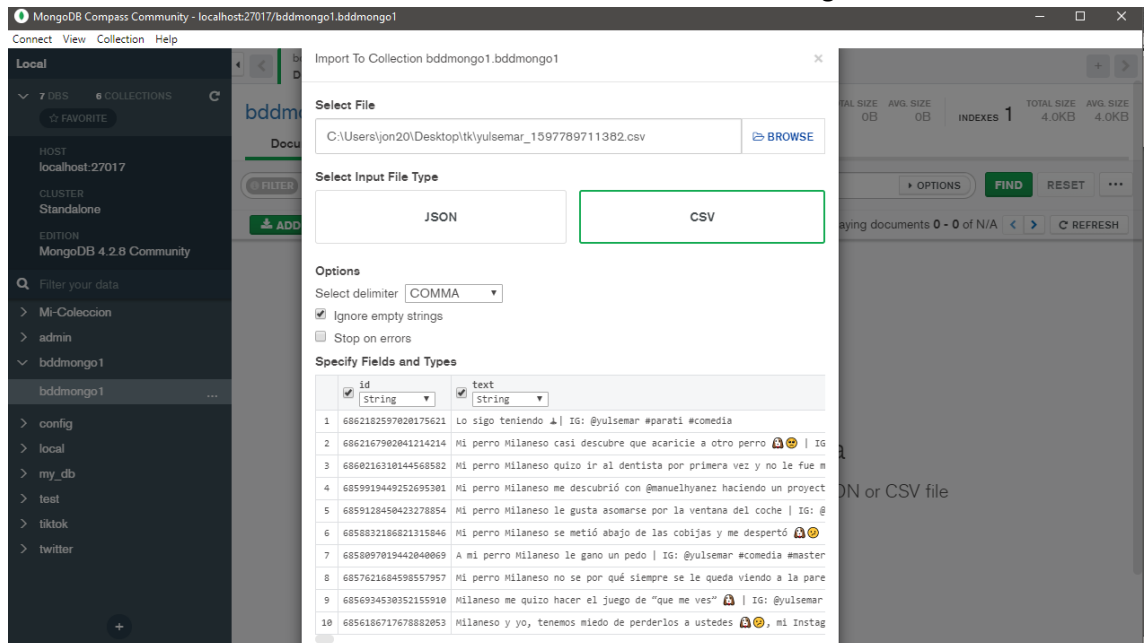


```
C:\Windows\System32\cmd.exe
File "C:\Python38\lib\socket.py", line 669, in readinto
  return self._sock.recv_into(b)
File "C:\Python38\lib\ssl.py", line 1241, in recv_into
  return self.read(nbytes, buffer)
File "C:\Python38\lib\ssl.py", line 1099, in read
  return self._sslobj.read(len, buffer)
KeyboardInterrupt
^C
/ Tiktok Scraper Started¿Desea terminar el trabajo por lotes (S/N)? s
C:\Users\jon20\Downloads>cd ..
C:\Users\jon20>desktop
"desktop" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\jon20>cd tk
El sistema no puede encontrar la ruta especificada.
C:\Users\jon20>cd desktop
C:\Users\jon20\Desktop>cd tk
C:\Users\jon20\Desktop\tk>tiktok-scraper user yulsemear -n 300 -t csv
C:\Users\jon20\Desktop\tk>tiktok-scraper user yulsemear -n 300 -t csv
CSV path: C:\Users\jon20\Desktop\tk\yulsemear_1597789711382.csv
C:\Users\jon20\Desktop\tk>
```

3. Ahora conectaremos con MongoDB, creando la base de datos para alojar el archivo csv. La base de datos se llamará bddMongo1.



4. Añadimos datos a nuestra colección. Subiremos nuestro archivo csv generado.



5. Así quedará finalmente nuestra base de datos con todos los documentos.

## Tweets relacionados a Donald Trump

1. Para esto debemos configurar nuestro script para que pueda hacer una captura de los tweets, distinguiéndolos por el id para que estos no se repitan, luego podremos especificar nuestra dirección de mongodb para guardar los datos. Finalmente, especificamos el tema del que haremos la búsqueda.

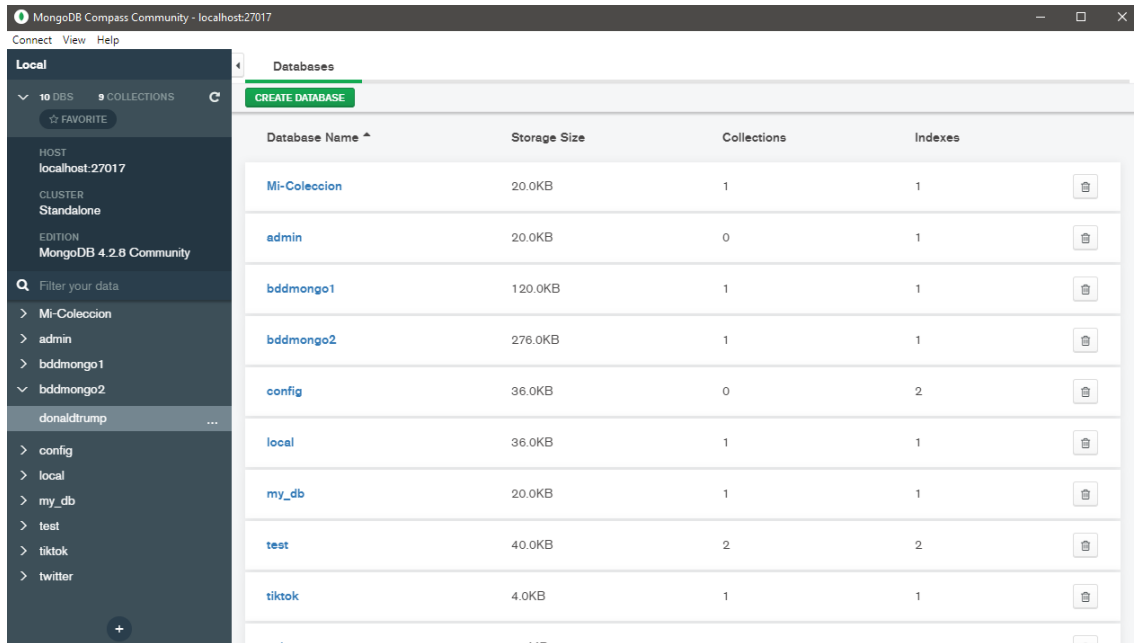
```
16 def on_data(self, data):
17     dictTweet = json.loads(data)
18     try:
19         dictTweet["id"] = str(dictTweet['id'])
20         doc = mycol.insert_one(dictTweet)
21         print("SAVED" + str(doc) + ">" + str(data))
22     except:
23         print("Already exists")
24         pass
25     return True
26
27 def on_error(self, status):
28     print(status)
29
30
31 auth = OAuthHandler(ckey, csecret)
32 auth.set_access_token(accessToken, asecret)
33 twitterStream = Stream(auth, listener())
34
35 '''Mongo'''
36
37 myclient = pymongo.MongoClient("mongodb://localhost:27017/?readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false")
38 mydb = myclient["bddmongo2"]
39 mycol = mydb["donalddump"]
40
41 '''-----LOCATIONS-----'''
42
43
44 twitterStream.filter(track=['Donald Trump'])
45
46
```

2. A continuación, ejecutaremos el script para que empiece con la captura y guardado de los datos.

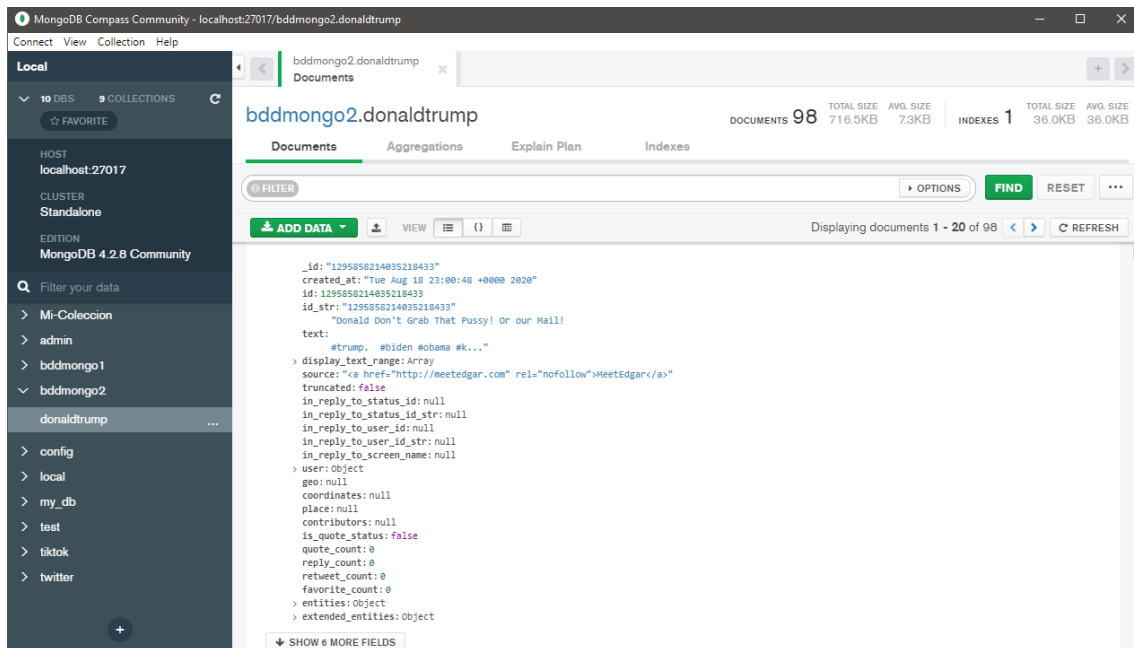
```
C:\Windows\System32\cmd.exe - python script3.py
'id':1295820645369077760,"source_status_id_str":"1295820645369077760","source_user_id":767788995962998784,"source_user_i
d_str":"767788995962998784"}],"favorited":false,"retweeted":false,"possibly_sensitive":false,"filter_level":"low","lang
":"en","is_quote_status":false,"quote_count":0,"reply_count":0,"retweet_count":0,"favorite_count":0,"entities":{"hashta
gs":[{"text":"RepublicanConfessions","indices":[73,95]}],"urls":[],"user_mentions":[{"screen_name":"MeidasTouch","name":
"MeidasTouch.com","id":1243560408025198593,"id_str":"1243560408025198593","indices":[3,15]}],"symbols":[],"media":[{"id
":1295819786052677639,"id_str":"1295819786052677639","indices":[96,119],"additional_media_info":{"title":"#RepublicanCo
nfessions","description":"Click the link in the video and help us air these in battleground states.", "embeddable":true,"mo
netizable":false,"media_url":"http://pbs.twimg.com/media/EfutuSsXkAA3h4h.jpg","media_url_https":"https://pbs.twi
ng.com/media/EfutuSsXkAA3h4h.jpg","url":"https://t.co/VJAmb20oiIs","display_url":"pic.twitter.com/VJAmb20oiIs","expa
nded_url":"https://twitter.com/ReallyAmerican1/status/1295820645369077760/video/1","type":"photo","sizes":{"medium
":{"w":1024,"h":576,"resize":"fit"},"thumb":{"w":150,"h":150,"resize":"crop"},"large":{"w":1024,"h":576,"resize":"fit"},
"small":{"w":680,"h":383,"resize":"fit"},"source_status_id":1295820645369077760,"source_status_id_str":"129582064536907
7760","source_user_id":767788995962998784,"source_user_id_str":"767788995962998784"}],"extended_entities":{"media":[{"i
d":1295819786052677639,"id_str":"1295819786052677639","indices":[96,119],"additional_media_info":{"title":"#RepublicanCo
nfessions","description":"Click the link in the video and help us air these in battleground states.", "embeddable":true,"mo
netizable":false,"media_url":"http://pbs.twimg.com/media/EfutuSsXkAA3h4h.jpg","media_url_https":"https://pbs.twi
ng.com/media/EfutuSsXkAA3h4h.jpg","url":"https://t.co/VJAmb20oiIs","display_url":"pic.twitter.com/VJAmb20oiIs","expa
nded_url":"https://twitter.com/ReallyAmerican1/status/1295820645369077760/video/1","type":"video","video_info":{"
aspect_ratio":[16,9],"duration_millis":84351,"variants":[{"bitrate":832000,"content_type":"video/mp4","url":"https://
/video.twimg.com/amplify_video/1295819786052677639/vid/640x360/1VZUBIzX5XnSolRo.mp4?tag=13"}, {"bitrate":288000,"con
tent_type":"video/mp4","url":"https://video.twimg.com/amplify_video/1295819786052677639/vid/480x270/pP9vK7gXuyyl
ZMkN.mp4?tag=13"}, {"content_type":"application/x-mpegURL","url":"https://video.twimg.com/amplify_video/129581978605
2677639/pl/W872a8HIZs0BnT2u.m3u8?tag=13"}, {"bitrate":2176000,"content_type":"video/mp4","url":"https://video.twimg
.com/amplify_video/1295819786052677639/vid/1280x720/26AF-w05aD75EUnB.mp4?tag=13"}],"sizes":{"medium":{"w":1024,"h":
576,"resize":"fit"},"thumb":{"w":150,"h":150,"resize":"crop"},"large":{"w":1024,"h":576,"resize":"fit"},"small":{"w":680
,"h":383,"resize":"fit"},"source_status_id":1295820645369077760,"source_status_id_str":"1295820645369077760","source_us
er_id":767788995962998784,"source_user_id_str":"767788995962998784"}],"favorited":false,"retweeted":false,"possibly_sen
sitive":false,"filter_level":"low","lang":"en","timestamp_ms":"1597791653239"}
```



3. En MongoDB podemos ir chequeando como se llena de documentos mientras el script se siga ejecutando. En este caso todo estará guardado en la base bddmongo2.



4. Al acceder a la base, previsualizaremos los datos en forma de documentos.



## WebScraping

### 1. Seleccionamos una página para poder obtener los datos



2. En el script para poder hacer la captura de datos mediante webscraping, en este deberá especificarse la dirección del cliente de la base de datos, el nombre de la base y el nombre de la colección, en caso de que no existan, se crearán. Luego deberemos especificar la dirección de la página que vamos a recorrer. Colocaremos el nombre de la etiqueta que va a capturar y se va a ir guardando en la base. El script realiza un recorrido de todos los datos y va separando los datos que necesitamos, en este caso es el título de las publicaciones en la etiqueta a y con su respectiva clase.

```
1 import requests
2 from bs4 import BeautifulSoup
3 # Import MongoClient from pymongo so we can connect to the database
4 from pymongo import MongoClient
5
6
7
8
9 if __name__ == '__main__':
10     # Instantiate a client to our MongoDB instance
11     db_client = MongoClient("mongodb://localhost:27017/?readPreference=primary&appName=MongoDB%20Compass%20Community&ssl=false")
12     bddmongo2 = db_client.bddmongo2
13     posts = bddmongo2.posts
14
15
16     response = requests.get("https://www.wattpad.com/stories/aventura")
17     soup = BeautifulSoup(response.content, "lxml")
18
19     post_titles = soup.find_all("a", class_="title meta on-story-preview")
20
21     extracted = []
22     for post_title in post_titles:
23         extracted.append({
24             'title': post_title.text,
25             'link': post_title["/story/130579579-bad-boy-peligro-1-completa"]
26         })
27
28     # Iterate over each post. If the link does not exist in the database, it's new! Add it.
29     for post in extracted:
30         if db_client.bddmongo2.posts.find_one({'link': post['link']}) is None:
31             # let's print it out to verify that we added the new post
32             print("Found a new listing at the following url: ", post['link'])
33             db_client.bddmongo2.posts.insert(post)
34
35
```



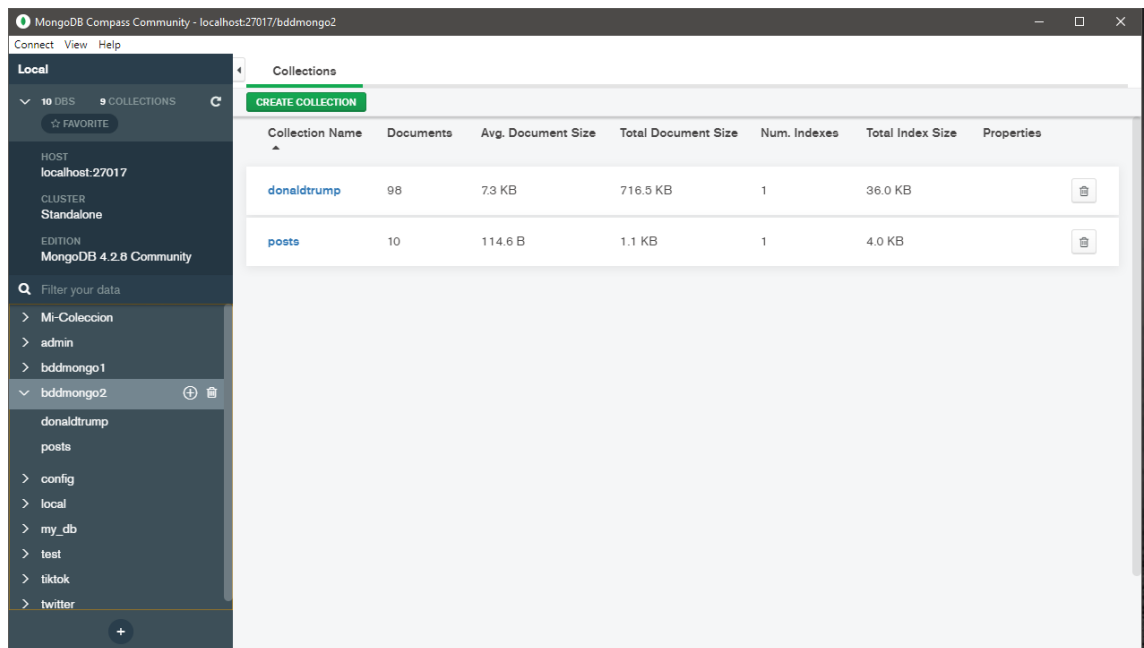
3. A continuación, ejecutaremos nuestro script para que comience con la captura de los datos.

```
C:\Windows\System32\cmd.exe

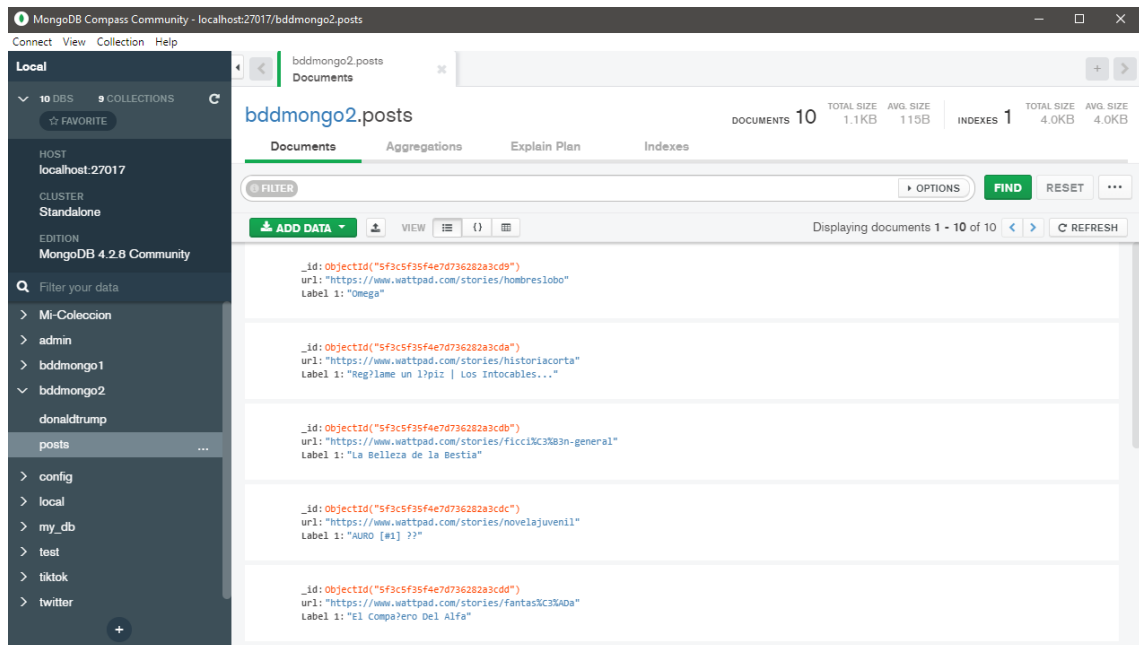
Traceback (most recent call last):
  File "script3.py", line 44, in <module>
    twitterStream.filter(track=['Donald Trump'])
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 474, in filter
    self._start(is_async)
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 389, in _start
    self._run()
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 320, in _run
    six.reraise(*exc_info)
  File "C:\Python38\lib\site-packages\six.py", line 703, in reraise
    raise value
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 289, in _run
    self._read_loop(resp)
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 339, in _read_loop
    line = buf.readline()
  File "C:\Python38\lib\site-packages\tweepy\streaming.py", line 200, in readline
    self._buffer += self._stream.read(self._chunk_size)
  File "C:\Python38\lib\site-packages\urllib3\response.py", line 540, in read
    raise IncompleteRead(self._fp_bytes_read, self.length_remaining)
  File "C:\Python38\lib\contextlib.py", line 131, in __exit__
    self.gen.throw(type, value, traceback)
  File "C:\Python38\lib\site-packages\urllib3\response.py", line 454, in _error_catcher
    raise ProtocolError("Connection broken: %r" % e, e)
urllib3.exceptions.ProtocolError: ('Connection broken: IncompleteRead(0 bytes read)', IncompleteRead(0 bytes read))

C:\Users\jon20\Downloads>python webscrapping.py
```

4. Finalmente, en nuestra base de datos en MongoDB, podremos verificar la creación de la colección posts que contendrá los datos.

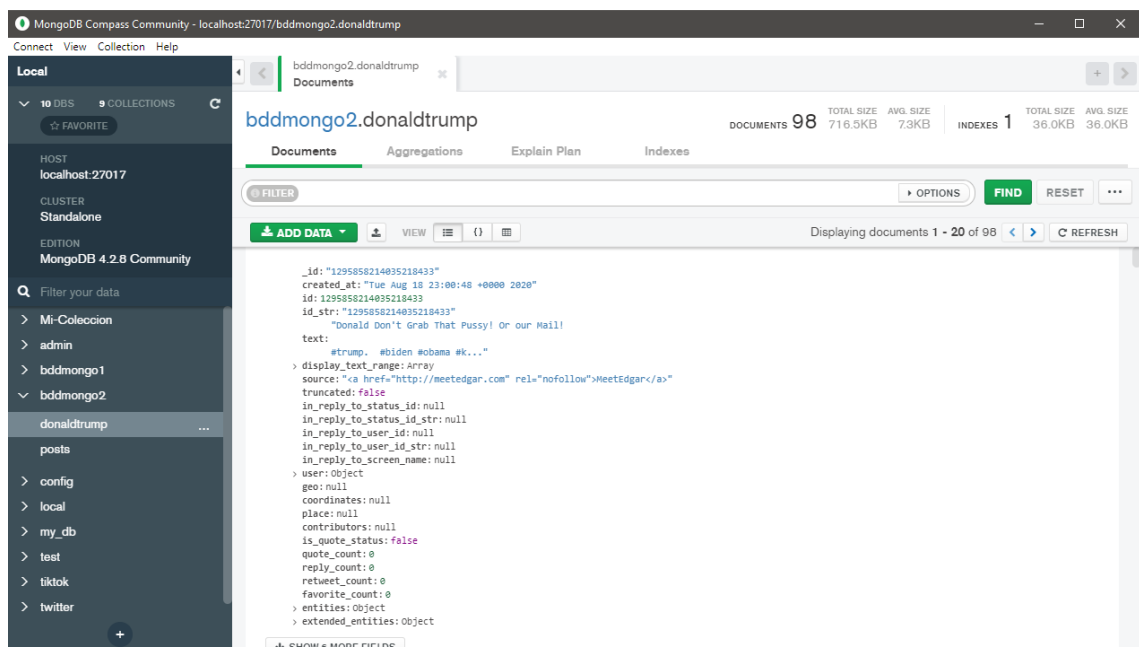


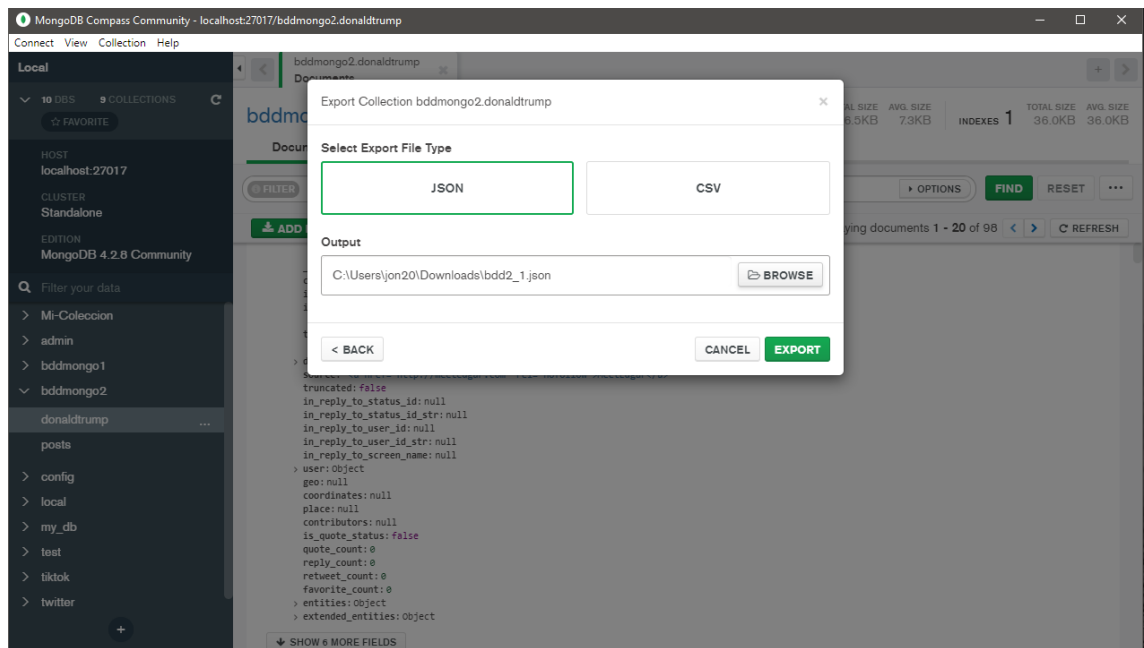
5. Los datos se presentan de la siguiente manera, en donde se puede visualizar la etiqueta que escogimos junto con su id.



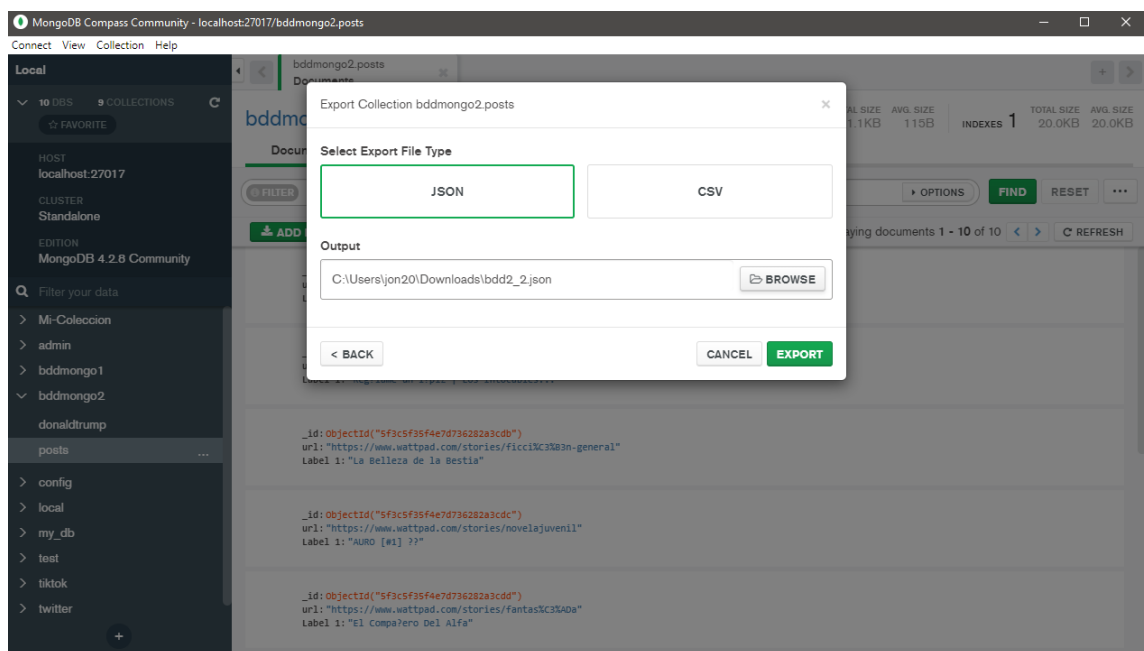
Exportar datos de una base de mongo db a otra

1. Para poder exportar datos de una base de mongo a otra, debemos darle clic en exportar archivo, cuando estemos dentro de alguna colección específica. Se hará el mismo procedimiento para la colección donaldtrump y posts.

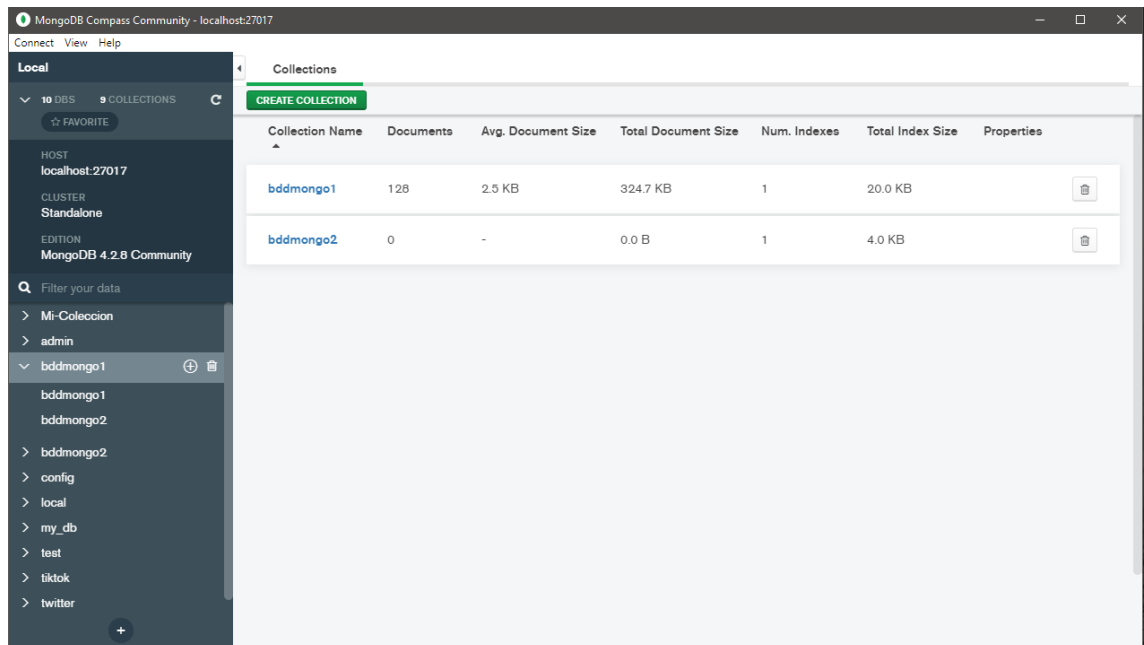




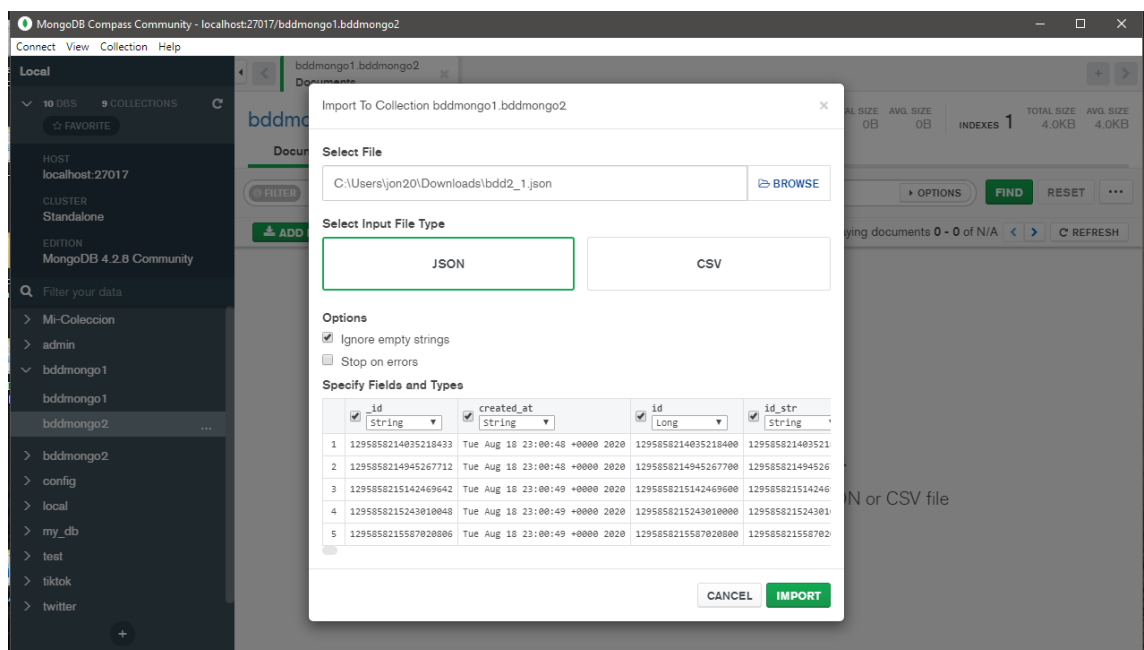
2. A los archivos exportado se los nombró bdd2\_1 y bdd2\_2, respectivamente.



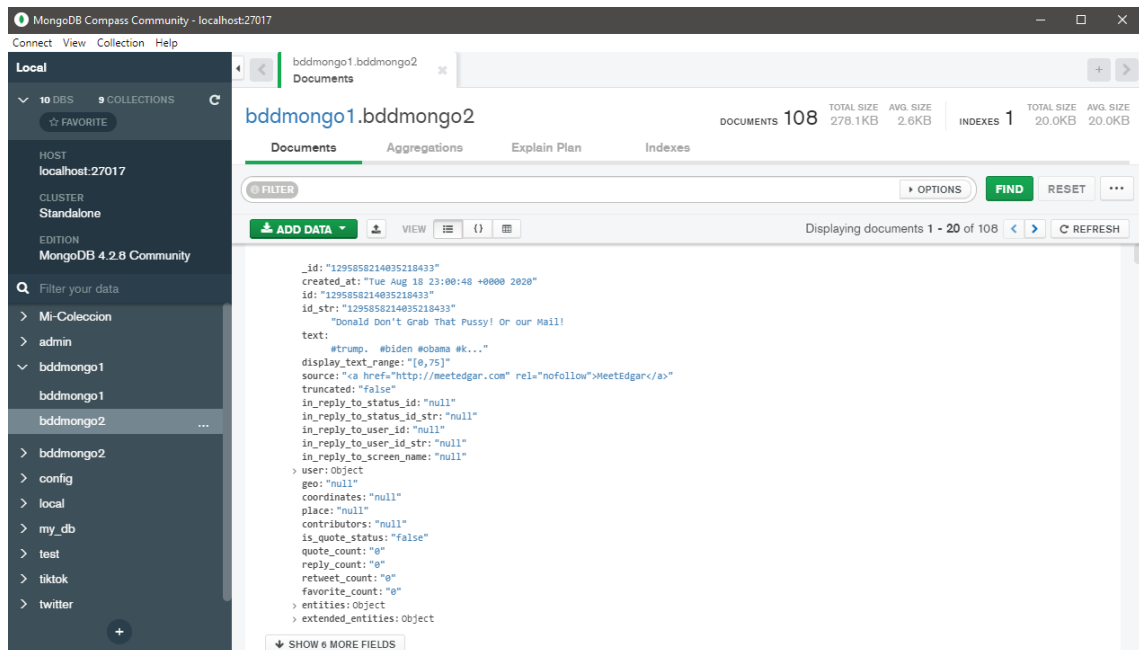
3. Con los archivos ya exportados, procedemos a importarlos en la nueva base de datos, en la cual se podrá añadir los dos archivos json dentro de una misma colección.



4. Solo será necesario darle clic en añadir archivo y especificar la ruta y formato del archivo, al final le damos en import.



5. Los datos se podrán visualizar en nuestra nueva base que contendrá la suma de los dos archivos, acumulando 108 documentos.



## Exportar datos de CouchDB a MongoDB

1. Para la exportación de los archivos de nuestra base de datos en CouchDB, deberemos ejecutar el comando curl -x GET, al que deberemos añadir nuestra dirección local donde está alojada nuestra base de datos junto con nuestras credenciales. Además, indicaremos la dirección en nuestro equipo donde vamos a guardar nuestro archivo json generado.

```
C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@127.0.0.1:5984/mls/_all_docs?include_docs=true > C:\Users\jon20\Desktop\bddcouch.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100    58    100    58      0     58      0:00:01 --:--:--  0:00:01 1234

C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@127.0.0.1:5984/newyork/_all_docs?include_docs=true > C:\Users\jon20\Desktop\bddcouch.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100    41    100    41      0     41      0:00:01 --:--:--  0:00:01 872

C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@localhost:5984/#database/newyork/_all_docs > C:\Users\jon20\Desktop\bddcouch.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   247    100   247      0     247      0:00:01 --:--:--  0:00:01 988

C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@localhost:5984/#database/newyork/_all_docs > C:\Users\jon20\Desktop\bddcouch.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   247    100   247      0     247      0:00:01 --:--:--  0:00:01 988

C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@localhost:5984/#database/newyork/_all_docs?include_docs=true > C:\Users\jon20\Desktop\bddcouch.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   247    100   247      0     247      0:00:01 --:--:--  0:00:01 988

C:\Users\jon20\Desktop>curl -X GET http://127.0.0.1:5984/newyork/_all_docs?include_docs=true > C:\Users\jon20\Desktop\db.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100    78    100    78      0     78      0:00:01 --:--:--  0:00:01 1258

C:\Users\jon20\Desktop>curl -X GET http://israel:jypam.1995@127.0.0.1:5984/newyork/_all_docs?include_docs=true > C:\Users\jon20\Desktop\db.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 9871k   100 9871k      0    9871k      0:00:01 --:--:--  0:00:01 15.0M

C:\Users\jon20\Desktop>
```

2. El archivo json se mostrará de la siguiente manera, será tan extenso como datos tenga en su estructura.

```
4 {"id":"1294810783444897792","key":"1294810783444897792","value":{"rev":"1-8f35d0750f9fdbce65d08493c9162e8f"},"doc":{"_id":"1294810783444897
5 {"id":"1294810888155799552","key":"1294810888155799552","value":{"rev":"1-0a2802ebd33a3314da8a410ae6fa79fd"},"doc":{"_id":"1294810888155799
6 {"id":"1294810915339026432","key":"1294810915339026432","value":{"rev":"1-a96196c15969ad2d15a2a4496e40a9e7"},"doc":{"_id":"1294810915339026
7 {"id":"1294810937174761472","key":"1294810937174761472","value":{"rev":"1-aa6d2cfc4fd6ceca7fc683305bf9e2"},"doc":{"_id":"1294810937174761
8 {"id":"1294810963217014787","key":"1294810963217014787","value":{"rev":"1-564cf60277fb58c9eb841155f137468b"},"doc":{"_id":"1294810963217014
9 {"id":"1294810989465153536","key":"1294810989465153536","value":{"rev":"1-2afc0d67f87ae737c3f1f1c3ea4885a5"},"doc":{"_id":"1294810989465153
10 {"id":"1294810994078867456","key":"1294810994078867456","value":{"rev":"1-7284a7b7308e0729d636a8bf31e8c754"},"doc":{"_id":"1294810994078867
11 {"id":"1294811016866529281","key":"1294811016866529281","value":{"rev":"1-3eb825ef0ec90a2ef9abdcac7594c845"},"doc":{"_id":"1294811016866529
12 {"id":"1294811032804720643","key":"1294811032804720643","value":{"rev":"1-3511314cd7388b457ea334f3c6f348d"},"doc":{"_id":"1294811032804720
13 {"id":"1294811050135576576","key":"1294811050135576576","value":{"rev":"1-62cae671b648ec929bac778eb80cfb7e"},"doc":{"_id":"1294811050135576
14 {"id":"1294811118230216704","key":"1294811118230216704","value":{"rev":"1-d9ee7183281ded0a65b71b0707aea7c4"},"doc":{"_id":"1294811118230216
15 {"id":"1294811122856468480","key":"1294811122856468480","value":{"rev":"1-ee194d5659cbfa4af6912f6e414fc1c6"},"doc":{"_id":"1294811122856468
16 {"id":"1294811203437563906","key":"1294811203437563906","value":{"rev":"1-96ba0fd6f0e87783f9fdc759c48b9d8"},"doc":{"_id":"1294811203437563
17 {"id":"129481125175588641","key":"129481125175588641","value":{"rev":"1-643eeef70b92028265a9f2da6684927d"},"doc":{"_id":"1294811251755886
18 {"id":"1294811305468018688","key":"1294811305468018688","value":{"rev":"1-16e7c887f654715fb7b390aeb9d966af"},"doc":{"_id":"1294811305468018
19 {"id":"1294811347549478914","key":"1294811347549478914","value":{"rev":"1-d7c9abbf3e0437a436d487d1bcf7723"},"doc":{"_id":"1294811347549478
20 {"id":"1294811384836845568","key":"1294811384836845568","value":{"rev":"1-bf4aaade173aa62fb29073bc771efb88"},"doc":{"_id":"1294811384836845
21 {"id":"1294811387655585792","key":"1294811387655585792","value":{"rev":"1-25507f24aa685108c5baaf05b62cc45ad"},"doc":{"_id":"1294811387655585
22 {"id":"1294811396505509888","key":"1294811396505509888","value":{"rev":"1-58336adfe052586a2439a6e6f16e3a16"},"doc":{"_id":"1294811396505509
23 {"id":"1294811445020905472","key":"1294811445020905472","value":{"rev":"1-117152b331523abb9c8d37c1eb21f9cc"},"doc":{"_id":"1294811445020905
24 {"id":"1294811591171387398","key":"1294811591171387398","value":{"rev":"1-48eecd2c1a1a454e49cda2c53f5935"},"doc":{"_id":"1294811591171387
25 {"id":"1294811593834864640","key":"1294811593834864640","value":{"rev":"1-6703a4b6cbd175dc8ee014cb795ca458"},"doc":{"_id":"1294811593834864
26 {"id":"1294811594765918209","key":"1294811594765918209","value":{"rev":"1-87a725cb186708d01bb713b96d94d6a2"},"doc":{"_id":"1294811594765918
27 {"id":"1294811595034333186","key":"1294811595034333186","value":{"rev":"1-53570a1fe26d59c95d5ff547aa31760"},"doc":{"_id":"1294811595034333
28 {"id":"1294811636637761536","key":"1294811636637761536","value":{"rev":"1-0fd44265d7a9002860bb83c9cfb74e50"},"doc":{"_id":"1294811636637761
29 {"id":"1294811705080451074","key":"1294811705080451074","value":{"rev":"1-737d6886469c1856b5147f7551b37397"},"doc":{"_id":"1294811705080451
30 {"id":"1294811756674465794","key":"1294811756674465794","value":{"rev":"1-940bf05bd0da935460f293cc4a237c32"},"doc":{"_id":"1294811756674465
31 {"id":"129481176076511745","key":"129481176076511745","value":{"rev":"1-5563af8f3d1fc243fc7eedf82f4d65a"},"doc":{"_id":"129481176076511
32 {"id":"1294811769978798080","key":"1294811769978798080","value":{"rev":"1-d09ba25fb77d5802a0fddbd88103cd0f"},"doc":{"_id":"1294811769978798
33 {"id":"1294811781907410944","key":"1294811781907410944","value":{"rev":"1-c5e596bae6ad50a44f3d2ec48ed31b7e"},"doc":{"_id":"1294811781907410
34 {"id":"1294811782591270912","key":"1294811782591270912","value":{"rev":"1-127085191cbf5e3d810dc36bf54f338d"},"doc":{"_id":"1294811782591270
35 {"id":"1294811785527046144","key":"1294811785527046144","value":{"rev":"1-d94e1800b7bed0c6b8aa43633f8762c5"},"doc":{"_id":"1294811785527046
36 {"id":"1294811787154595841","key":"1294811787154595841","value":{"rev":"1-97550bb8e3cccc1fc1fd81cad8f4b872"},"doc":{"_id":"1294811787154595
37 {"id":"1294811787184013312","key":"1294811787184013312","value":{"rev":"1-f245038de7ade1f0fc13d3589f42a0e3"},"doc":{"_id":"1294811787184013
```

3. El siguiente paso será importar este archivo json desde nuestro mongodb, para ello, simplemente crearemos una nueva colección donde guardaremos estos datos. El nombre de esta colección es couch.

Import To Collection bddmongo1.couch

Select File

C:\Users\jon20\Desktop\db.json

BROWSE

Select Input File Type

JSON

CSV

Options

☒ Ignore empty strings

☐ Stop on errors

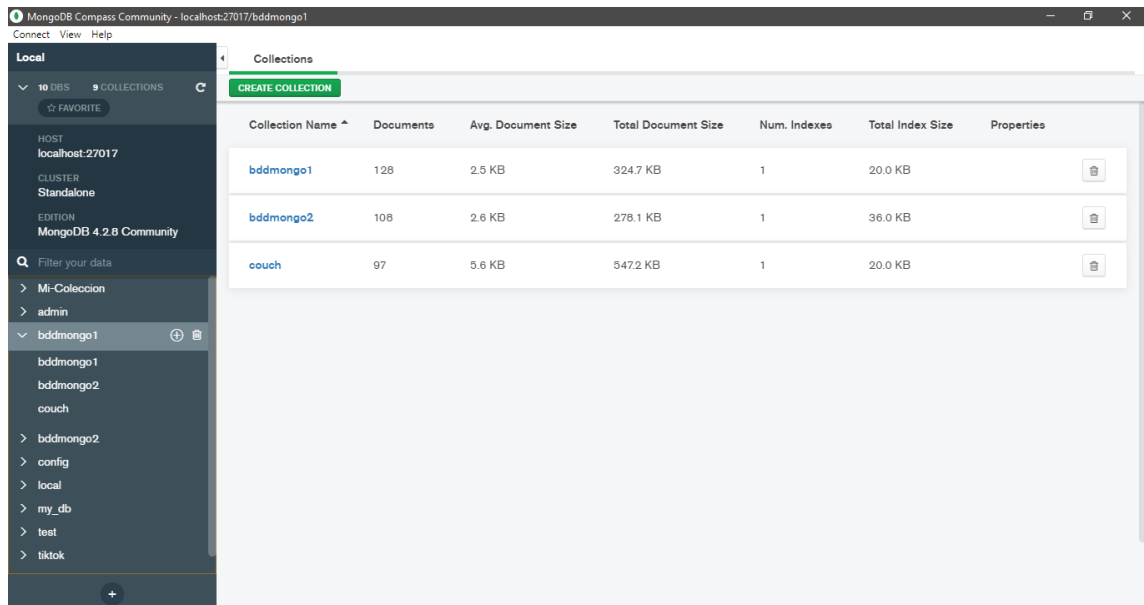
Specify Fields and Types

	<input checked="" type="checkbox"/> id String	<input checked="" type="checkbox"/> key String	<input checked="" type="checkbox"/> value.rev String	<input checked="" type="checkbox"/> doc_id String
1	1294810626854879233	1294810626854879233	1-a5ce18b824963a72cc40395d01d97cb0	129481062685
2	1294810750213529601	1294810750213529601	1-523e299768ead0785d93b84ab66eb111	129481075021
3	1294810783444897792	1294810783444897792	1-8f35d0750f9fdbce65d08493c9162e8f	129481078344
4	1294810888155799552	1294810888155799552	1-0a2802ebd33a3314da8a410ae6fa79fd	129481088815
5	1294810915339026432	1294810915339026432	1-a96196c15969ad2d15a2a4496e40a9e7	129481091533

CANCEL

IMPORT

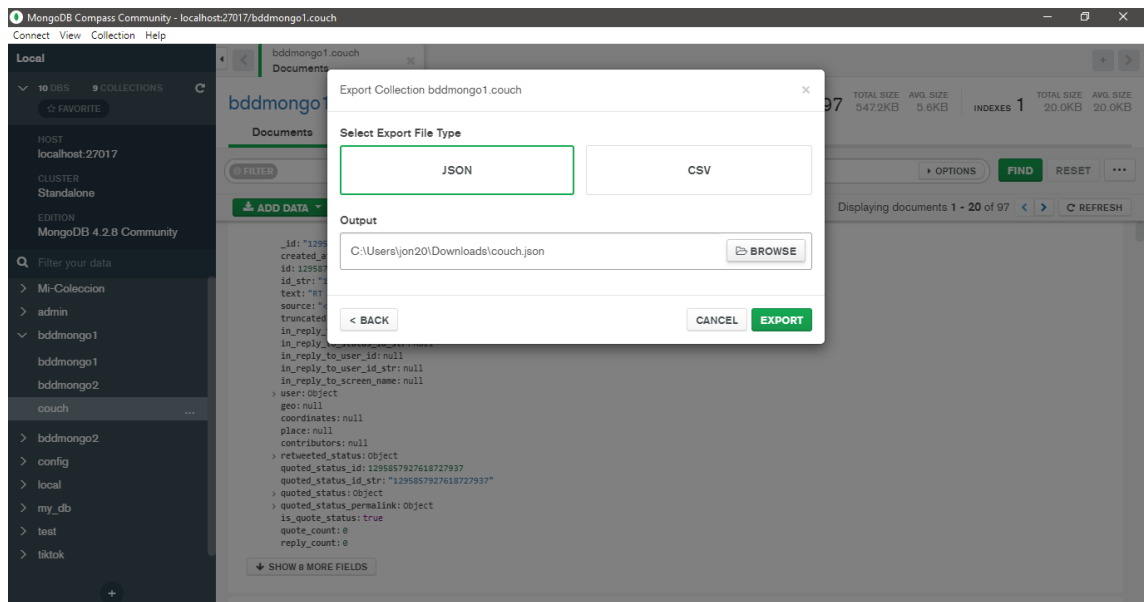
- Podremos comprobar que se han importado los datos correctamente a nuestra colección, con el número de documentos que contiene.



Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
<a href="#">bddmongo1</a>	128	2.5 KB	324.7 KB	1	20.0 KB	
<a href="#">bddmongo2</a>	108	2.6 KB	278.1 KB	1	36.0 KB	
<a href="#">couch</a>	97	5.6 KB	547.2 KB	1	20.0 KB	

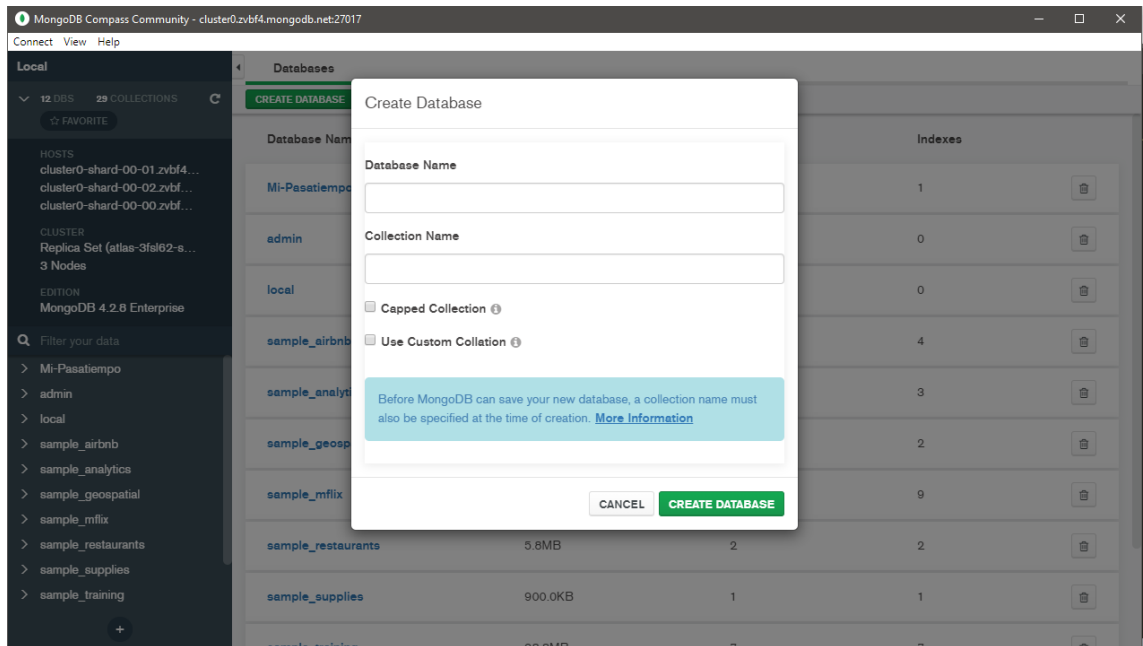
Enviar base de datos local a la cuenta de atlas

- Para poder trasladar nuestras colecciones a un cluster en el servidor con nuestra cuenta de atlas, primero debemos exportar en formato json, los datos de nuestras colecciones, bdd2\_1, bdd2\_2 y couch, que ya habíamos obtenido anteriormente.

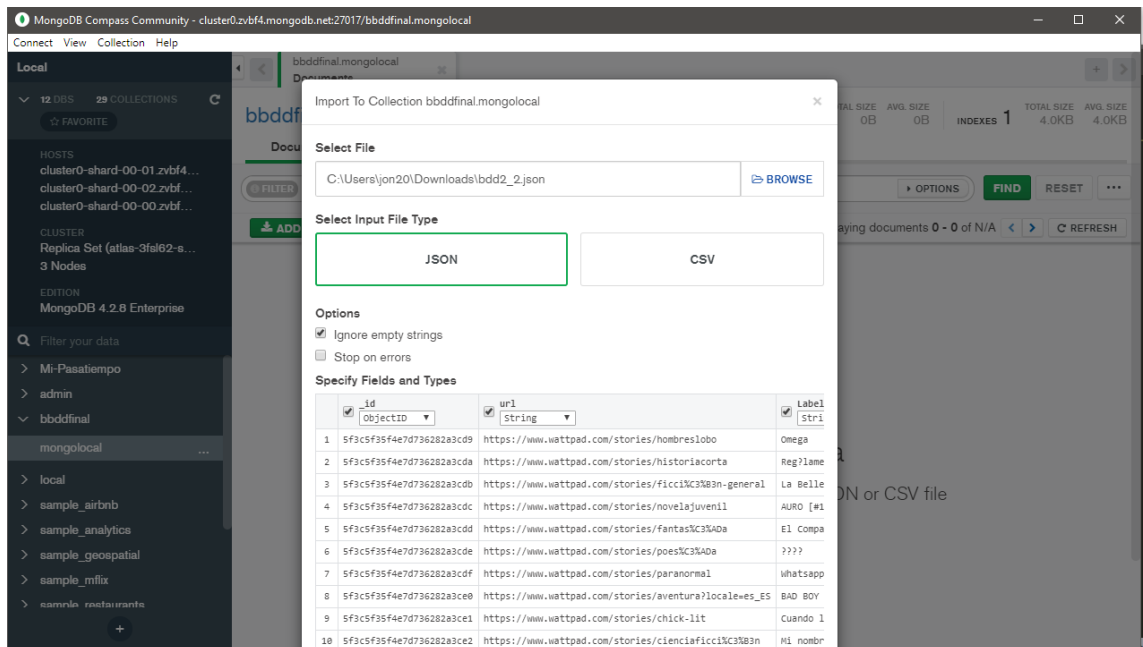




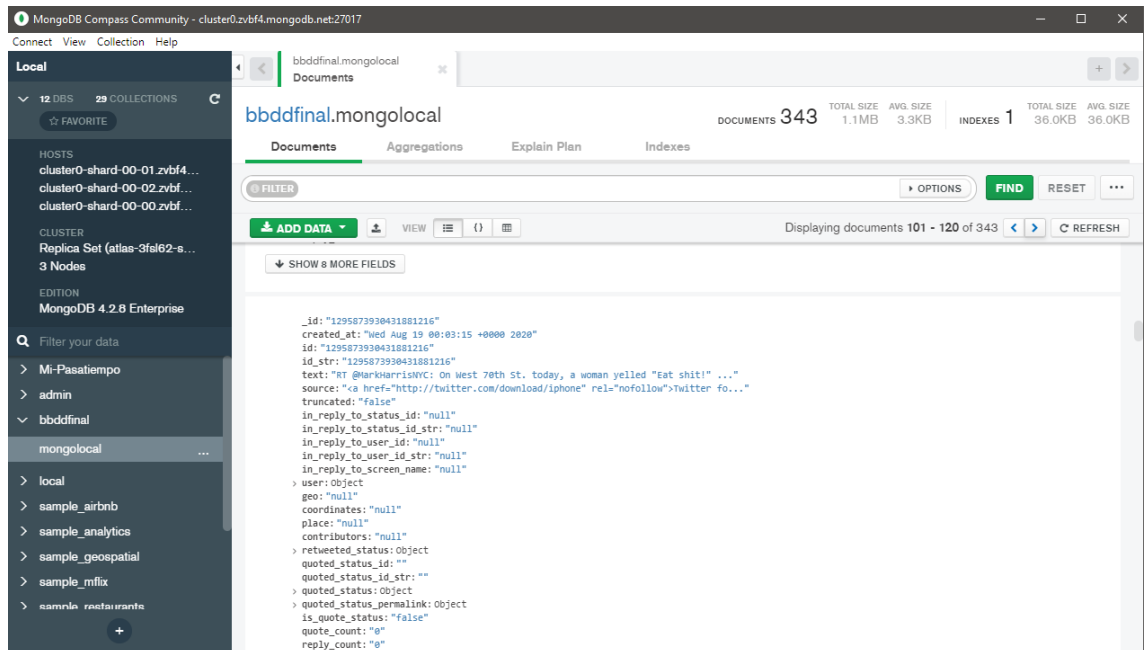
2. Deberemos acceder a nuestro cluster, especificando la dirección del mismo y sus credenciales de acceso. Ya dentro de éste, podremos crear una colección para guardar todos nuestros datos.



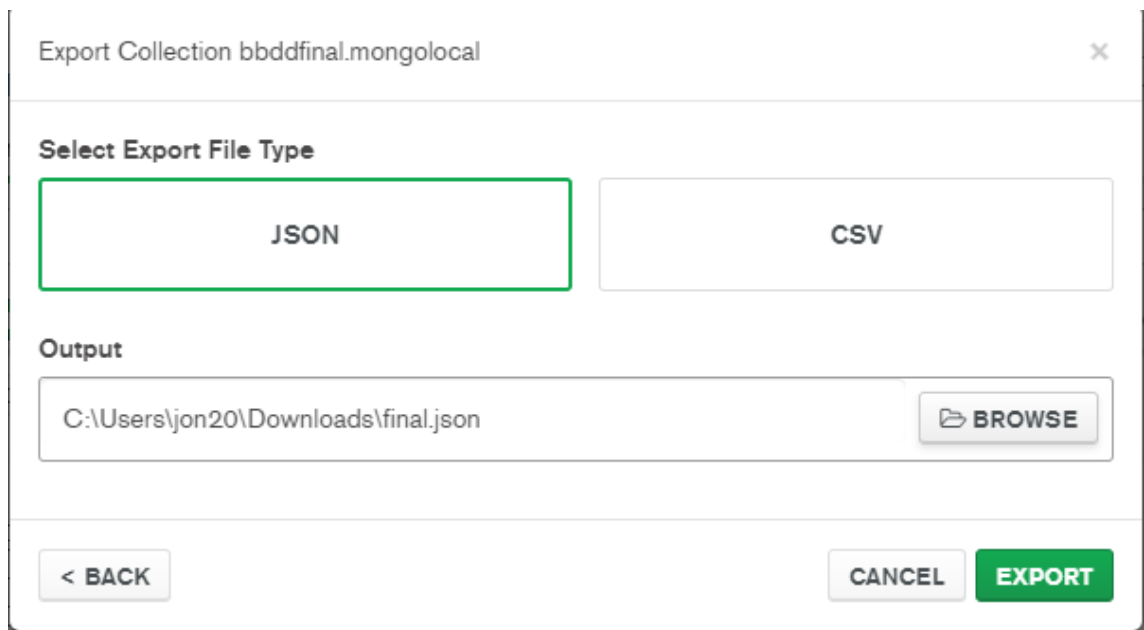
3. Nuestra base de datos se llama bbddfial y nuestra colección mongolocal. Aquí, vamos a importar los archivos json que obtuvimos de nuestra base de datos local de mongodb.



4. De esta forma habremos recolectado los datos de nuestras tres colecciones, las cuales estarán en nuestro cluster en el servidor. Entre todos los documentos hemos recolectado 343 documentos.



Finalmente, exportaremos nuestros datos en formato json y csv para cumplir el propósito de la práctica.



Export Collection bddfinal.mongolocal

Select Export File Type

JSON

CSV

Output

C:\Users\jon20\Downloads\final.csv

BROWSE

< BACK

CANCEL

EXPORT

Cadena de conexión MongoDB:

mongodb+srv://Israel:jypam.1995@cluster0.zvbf4.mongodb.net/test?authSource=admin&replicaSet=atlas-3fsl62-shard-0&readPreference=primary&appname=MongoDB%20Compass%20Community&ssl=true