



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



FUNDAMENTOS DE SOFTWARE

ASIGNATURA:

Programación Orientada a Objetos

PROFESOR:

Ing. Juan Pablo Zaldumbide

PERÍODO ACADÉMICO:

2020-B

INFORME DE LABORATORIO

TÍTULO:

Corrección del examen



mongoDB

NOMBRE DE ESTUDIANTE: Iván Fraga

Denisse Cumbal

David Cacuango

Erick Andrade

Luis Catota

Marlon Tuquerrez

FECHA MÁXIMA DE ENTREGA: 31/08//2021

OBSERVACIONES:

1.py

Tres localizaciones

Mediante el código y el Api podemos obtener los datos

Resumen:

Trabajamos en python, pues importamos las librerías : pip install tweepy para que se guarden en couchdb donde se pone nuestras credenciales :

```
'''=====couchdb=====''' server =
couchdb.Server('http://CUMBAL:12345@127.0.0.1:5984') #('http://115.146.93.184:5984/')
try: db = server.create('exam31py') except: db = server['M']
```

```
'''=====LOCATIONS====='''
```

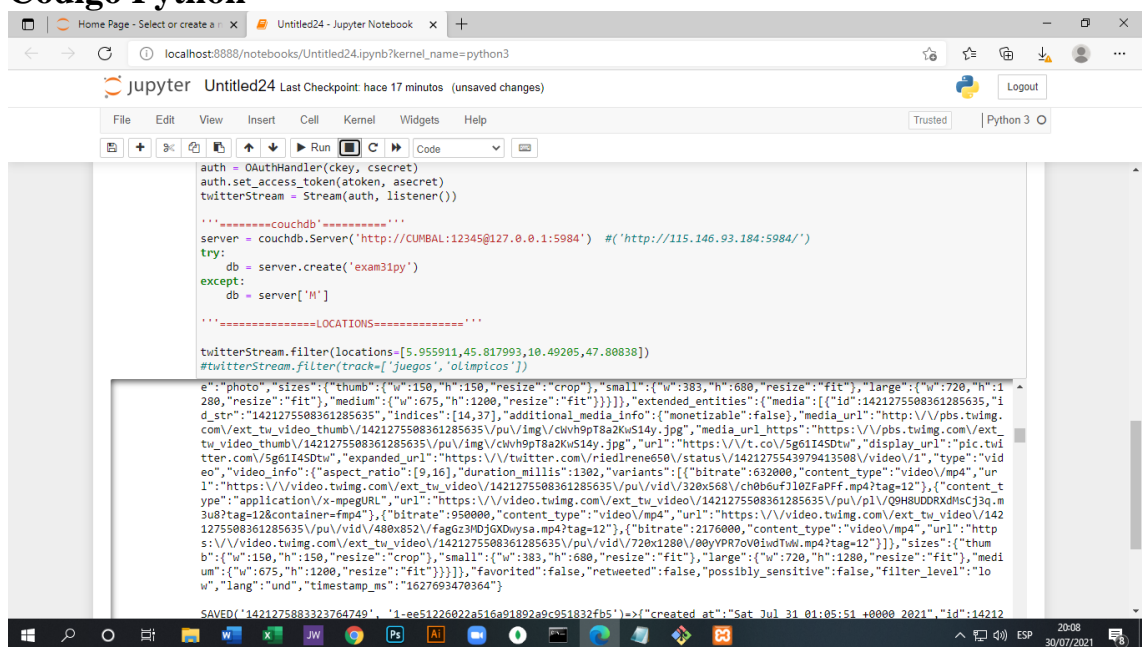
```
twitterStream.filter(locations=[5.955911,45.817993,10.49205,47.80838])
```

mediante este codigo nos podemos ubicar en el mapa y generar las coordenadas almacenar en la base de datos que deseamos en este caso fue couchdb por eso las credenciales son de esta base de datos y

```
twitterStream.filter(locations=[5.955911,45.817993,10.49205,47.80838])
```

ponemos la ubicacion que deseamos extraer los datos

Código Python



```

}
'''=====couchdb====='''
server = couchdb.Server('http://CUMBAL:12345@127.0.0.1:5984') #('http://115.146.93.184:5984/')
try:
    db = server.create('exam31py')
except:
    db = server['M']

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

twitterStream.filter(locations=[5.955911,45.817993,10.49205,47.80838])
#twitterStream.filter(track=['juegos', 'olimpicos'])

e="photo", "sizes": {"thumb": {"w": 150, "h": 150, "resize": "crop"}, "small": {"w": 383, "h": 680, "resize": "fit"}, "large": {"w": 720, "h": 1280, "resize": "fit"}, "medium": {"w": 675, "h": 1200, "resize": "fit"}}, "extended_entities": {"media": [{"id": "1421275508361285635", "id_str": "1421275508361285635", "indices": [14, 37], "additional_media_info": {"monetizable": false}, "media_url": "http://pbs.twimg.com/ext_tw_video_thumb/1421275508361285635/pu/img/cwvh9p78a2KwS14y.jpg", "media_url_https": "https://pbs.twimg.com/ext_tw_video_thumb/1421275508361285635/pu/img/cwvh9p78a2KwS14y.jpg", "url": "https://t.co/V5g61I45Dtw", "display_url": "pic.twitter.com/V5g61I45Dtw", "expanded_url": "https://twitter.com/riedlrene50/status/1421275543979413508/video/1", "type": "video", "video_info": {"aspect_ratio": [9, 16], "duration_millis": 1302, "variants": [{"bitrate": 632000, "content_type": "video/mp4", "url": "https://video.twimg.com/ext_tw_video/1421275508361285635/pu/vid/320x568/ch0b0ufj102FaPff.mp4?tag=12"}, {"content_type": "application/x-mpegurl", "url": "https://video.twimg.com/ext_tw_video/1421275508361285635/pu/pl/Q9H8UDRXdmScj3q.m3u8?tag=12&container=fmp4"}, {"bitrate": 958000, "content_type": "video/mp4", "url": "https://video.twimg.com/ext_tw_video/1421275508361285635/pu/vid/480x852/fag6z3MDjGxDwysa.mp4?tag=12"}, {"bitrate": 2176000, "content_type": "video/mp4", "url": "https://video.twimg.com/ext_tw_video/1421275508361285635/pu/vid/720x1280/00yVP70v0lwdTww.mp4?tag=12"}]}, "sizes": {"thumb": {"w": 150, "h": 150, "resize": "crop"}, "small": {"w": 383, "h": 680, "resize": "fit"}, "large": {"w": 720, "h": 1280, "resize": "fit"}, "medium": {"w": 675, "h": 1200, "resize": "fit"}}, "favorited": false, "retweeted": false, "possibly_sensitive": false, "filter_level": "low", "lang": "und", "timestamp_ms": "1627693470364"}
  
```

Verificamos que se halla almacenado en Couchdb

id	key	value
1421275337485344771	1421275337485344771	{ "rev": "1-58d5d719b835167693a8bc..." }
1421275543979413508	1421275543979413508	{ "rev": "1-f4261e1cc9c14a0f5944bb3..." }
1421275883323764749	1421275883323764749	{ "rev": "1-ee51226022a516a91892a9..." }
1421276076014198787	1421276076014198787	{ "rev": "1-00d44eb0d7cc7d2b7468e7..." }
1421276202929688578	1421276202929688578	{ "rev": "1-28c0a926db4c8ebd4f0f6a0..." }
1421276234122674181	1421276234122674181	{ "rev": "1-f402ff0176d38ce453f3745f2..." }

exam1_1py	3.7 KB	2	No			
exam31py	16.1 KB	6	No			
exam4_1py	6.2 MB	2253	No			

Buscamos lo lugares cerca a TOKIO

Find a place with OpenStreetMap ...

Copy & Paste

CSV: 5 9559,45.818,10.4921,47.8084

2.py track

Usamos el mismo código para la localización de twitter con la diferencia de la línea que nos permite extraer los datos que comentamos, y ubicamos las palabras que deseamos buscar y extraer los datos.

Resumen:

El código es similar y la lógica también en este caso solo cambia una línea de código `#twitterStream.filter(track=['juegos','olimpicos'])` aquí le decimos que extraiga de twitter las palabras "juegos", "olimpicos", y las guarde en nuestra base de datos.

```

    return True

    def on_error(self, status):
        print (status)

    auth = OAuthHandler(ckey, csecret)
    auth.set_access_token(accessToken, asecret)
    twitterStream = Stream(auth, listener())

    '''=====Couchdb====='''
    server = couchdb.Server('http://CUMBAL:12345@127.0.0.1:5984') #('http://115.146.93.184:5984/')
    try:
        db = server.create('noemi')
    except:
        db = server['H']

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

    #twitterStream.filter(locations=[5.9559,45.818,10.4921,47.8084])
    twitterStream.filter(track=['juegos','olimpicos'])

    6be\u00e0f \u00e0f Ig: @galejandro_tecu @sotoshop", "translator_type": "none", "protected": false, "verified": false, "follower
    s_count": 97, "friends_count": 114, "listed_count": 0, "favourites_count": 4419, "statuses_count": 1967, "created_at": "Thu Nov 11 20:0
    1:35 +0000 2010", "utc_offset": null, "time_zone": null, "geo_enabled": true, "lang": null, "contributors_enabled": false, "is_translato
    r": false, "profile_background_color": "14181f", "profile_background_image_url": "http://abs.twimg.com/images/themes/theme9/bg.gif",
    "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme9/bg.gif", "profile_background_tile": false, "profile
    link_color": "2fc2ef", "profile_sidebar_border_color": "181a1e", "profile_sidebar_fill_color": "252429", "profil
    e_text_color": "666666", "profile_use_background_image": true, "profile_image_url": "http://pbs.twimg.com/profile_images/13890
    28739913236482/3yPGLRKR_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile_images/1389028739913236482
    /3yPGLRKR_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/214601400/1376104218", "default_profile
    e": false, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null, "withheld_in_countrie
    s": [], "geo": null, "coordinates": null, "place": null, "contributors": null, "retweeted_status": {"created_at": "Sat Jul 31 00:46:20 +

```

Verificamos que se halla almacenado en la base de datos Couchdb

_id	contributors	coordinates	created_at	entities
1421272467381833...	null	null	Sat Jul 31 00:52:16 ...	{"hashtags": [], "urls...
1421272471416840...	null	null	Sat Jul 31 00:52:17 ...	{"hashtags": [], "urls...
1421272473702649...	null	null	Sat Jul 31 00:52:18 ...	{"hashtags": [{"text...
1421272474659037...	null	null	Sat Jul 31 00:52:18 ...	{"hashtags": [], "urls...
1421272476718354...	null	null	Sat Jul 31 00:52:19 ...	{"hashtags": [], "urls...
1421272478186446...	null	null	Sat Jul 31 00:52:19 ...	{"hashtags": [{"text...
1421272478492594...	null	null	Sat Jul 31 00:52:19 ...	{"hashtags": [], "urls...
1421272480703000...	null	null	Sat Jul 31 00:52:20 ...	{"hashtags": [{"text...
1421272481059545...	null	null	Sat Jul 31 00:52:20 ...	{"hashtags": [], "urls...
1421272486365339...	null	null	Sat Jul 31 00:52:21 ...	{"hashtags": [{"text...

3.PY.- WEB_SCRAPING

Es una forma de limpieza de información, para esta práctica escogimos una página referente a las olimpiadas e limpiamos los datos a extraer e insertamos en la base de Datos MongoDB.

Resumen:

En esta forma trabajamos en python
buscamos una pagina en Internet donde y revisamos el html

```
response =
```

```
requests.get("https://resultados.as.com/resultados/juegos_olimpicos/resultados/") soup
= BeautifulSoup(response.content, "lxml") ''' aqui ponemos la clase si es o o y la clase '''
```

```
equi =soup.find_all('span',class_='deporte-nombre')
```

```
''' aqui todo guardamos en una lista ''' equipos =list() for i in equi:
equipos.append(i.text) print(equipos)
```

''' aqui la guardamos en json a mongodb mediante un diccionario ''' for i in range(len(equipos)):

```
a={
    '3.py':equipos[i]}
```

```
my_posts.insert_one(a)
```

```
a={}
```

```
print(equipos)
```

Código Python.

```
In [5]: import requests
        from bs4 import BeautifulSoup
        from pymongo import MongoClient
        import pandas as pd
        import bson
        from bson.raw_bson import RawBSONDocument

        db_client = MongoClient("localhost")
        my_db = db_client.examen_olimpiadas
        my_posts = my_db.tokio

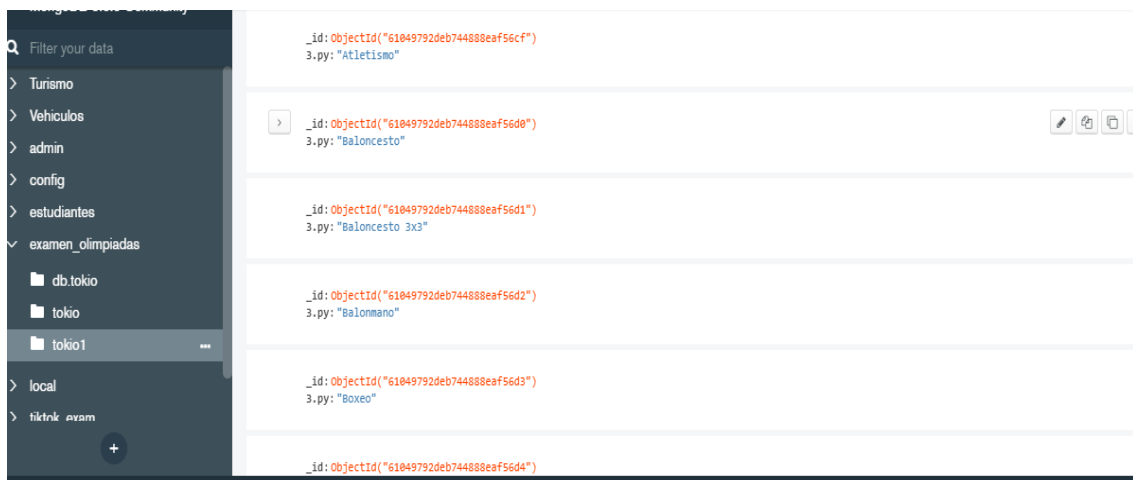
        def find_2nd(string, substring):
            return string.find(substring, string.find(substring) + 1)
        def find_1st(string, substring):
            return string.find(substring, string.find(substring))

In [6]: response = requests.get("https://resultados.as.com/resultados/juegos_olimpicos/resultados/")
        soup = BeautifulSoup(response.content, "lxml")

        Course=[]
        Provider=[]
        Duration=[]
        Start_Date=[]
        Offered_By=[]
        No_Of_Reviews=[]
        Rating=[]

        ...
        aqui guardamos con el nombre n.py el scrapping de html
```

Verificamos que se almacene correctamente la información solicitada.



4.PY FACEBOOK

Para este procedimiento es importante importar una librería “from facebook_scraper import get_posts”, para poder extraer los datos de la red Social

Resumen:

```
from bs4 import BeautifulSoup
```

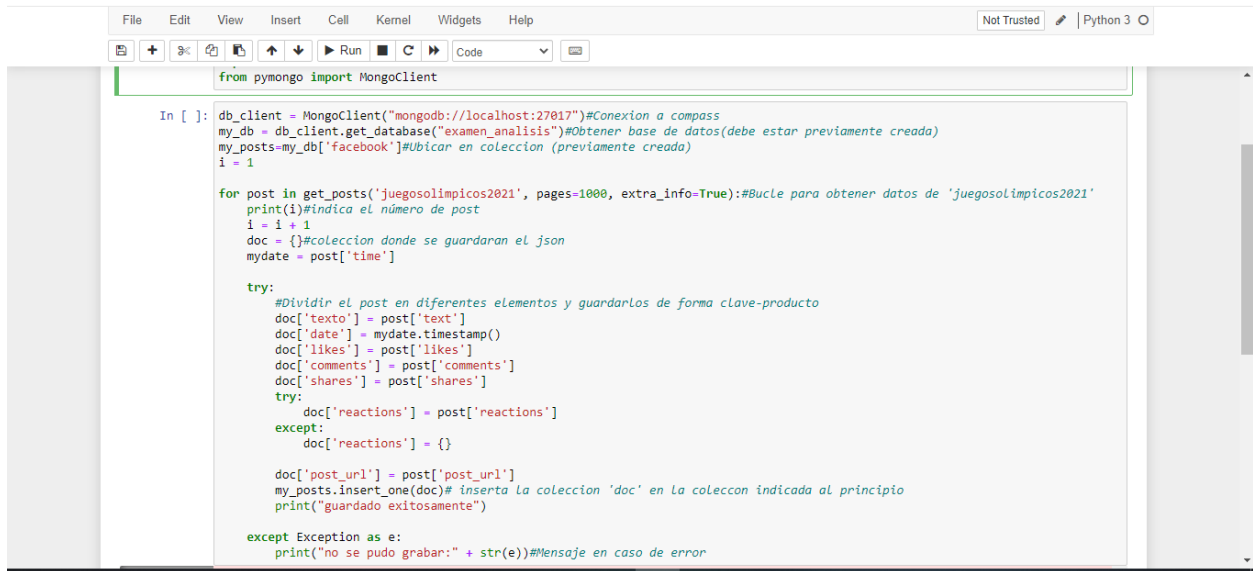
```
import pandas as pd
```

```
import requests
```

```
from facebook_scraper import get_posts
```

```
import couchdb
import json
import time
from pymongo import MongoClient
Se importo las librerías necesarias.
```

Codigo Pyhton:



```
from pymongo import MongoClient

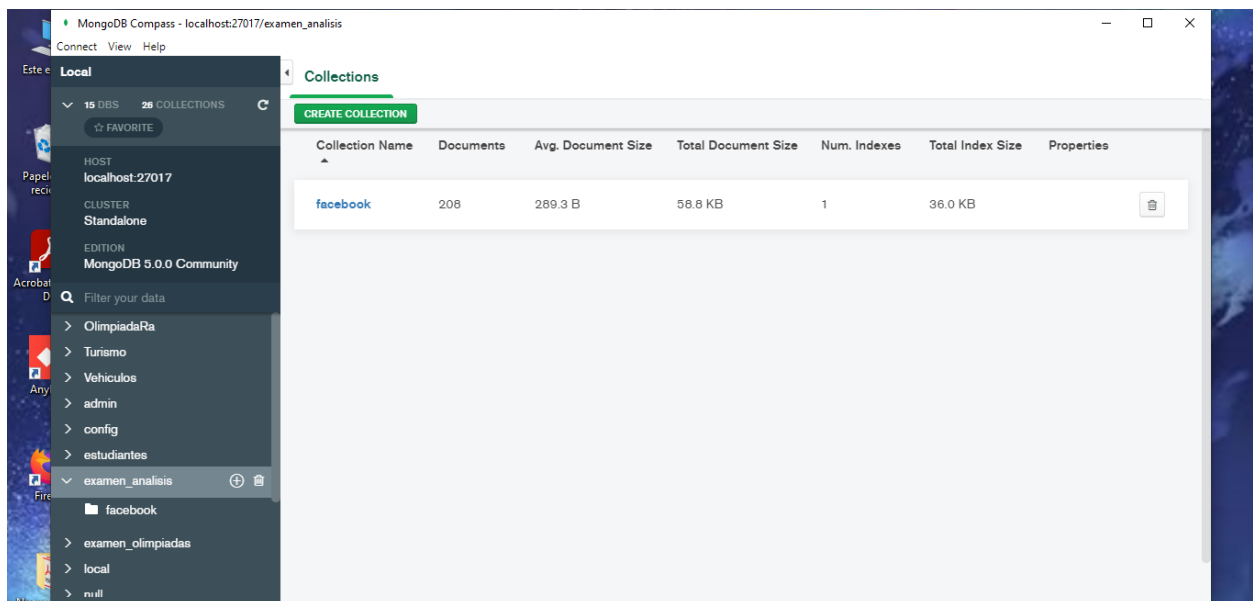
In [ ]: db_client = MongoClient("mongodb://localhost:27017")#Conexion a compass
my_db = db_client.get_database("examen_analisis")#Obtener base de datos(debe estar previamente creada)
my_posts=my_db['facebook']#Ubicar en coleccion (previamente creada)
i = 1

for post in get_posts('juegosolimpicos2021', pages=1000, extra_info=True):#Bucle para obtener datos de 'juegosolimpicos2021'
    print(i)#indica el número de post
    i = i + 1
    doc = {}#coleccion donde se guardaran el json
    mydate = post['time']

    try:
        #Dividir el post en diferentes elementos y guardarlos de forma clave-valor
        doc['texto'] = post['text']
        doc['date'] = mydate.timestamp()
        doc['likes'] = post['likes']
        doc['comments'] = post['comments']
        doc['shares'] = post['shares']
        try:
            doc['reactions'] = post['reactions']
        except:
            doc['reactions'] = {}

        doc['post_url'] = post['post_url']
        my_posts.insert_one(doc)# inserta la coleccion 'doc' en la coleccion indicada al principio
        print("guardado exitosamente")
    except Exception as e:
        print("no se pudo grabar:" + str(e))#Mensaje en caso de error
```

Verificamos que se almacene en la Base de Datos MongoDB



5.PY Tiktok

Mediante los comando ‘ tictock-scraper user ***** -t csv’ extraimos datos de la red social tictock

Resumen: En esta trabajamos con tiktok para esto necesitamos : tiktok-scraper user tokyo2020_brasil -t csv en mi caso use git bash para trabajar y todo guarde en una carpeta , busque los usuarios en internet ,despues es importante subir nuestros csv a mySQL, en mi cso trabaje con XAMP en my localshot cree una base de datos donde inserte un csv que se me genero de lo csv de tiktok, para esto debemos de antemano debemos saber como este estructurado si es co comas puntos y eso despues q especificamos todos los subimos vemos que no hay error y los mismo hacemos con todos los csv que tenemos ahora

```

MINGW64/c:/Users/CUMBAL/Desktop/base/analisis/examen_tiktok
CUMBAL@CUMBAL MINGW64 ~/Desktop/base/analisis/examen_tiktok
$ tiktok-scraper user Bellapoarch -t csv
- TikTok Scraper Started
CSV path: C:\Users\CUMBAL\Desktop\base\analisis\examen_tiktok\Bellapoarch_1627686836049.csv

CUMBAL@CUMBAL MINGW64 ~/Desktop/base/analisis/examen_tiktok
$ tiktok-scraper user oechileoficial -t csv
- TikTok Scraper Started
CSV path: C:\Users\CUMBAL\Desktop\base\analisis\examen_tiktok\oechileoficial_1627686909484.csv

CUMBAL@CUMBAL MINGW64 ~/Desktop/base/analisis/examen_tiktok
$ tiktok-scraper user -t csv
- TikTok Scraper Started
Missing input

CUMBAL@CUMBAL MINGW64 ~/Desktop/base/analisis/examen_tiktok
$ tiktok-scraper user -t csv
- TikTok Scraper Started
Missing input

CUMBAL@CUMBAL MINGW64 ~/Desktop/base/analisis/examen_tiktok

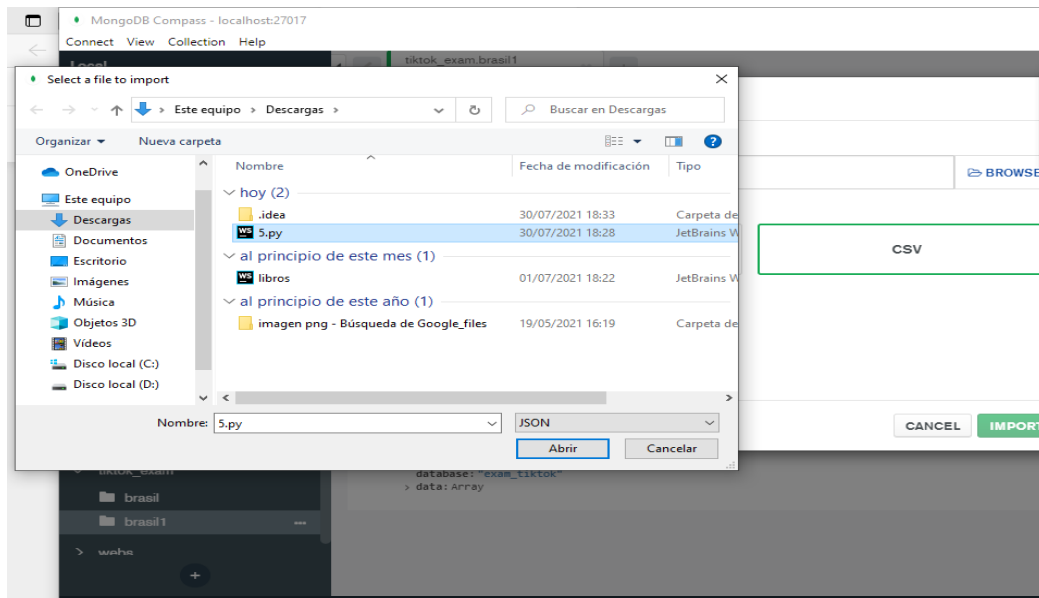
```

Subimos exitosamente los csv a phpMyAdmin

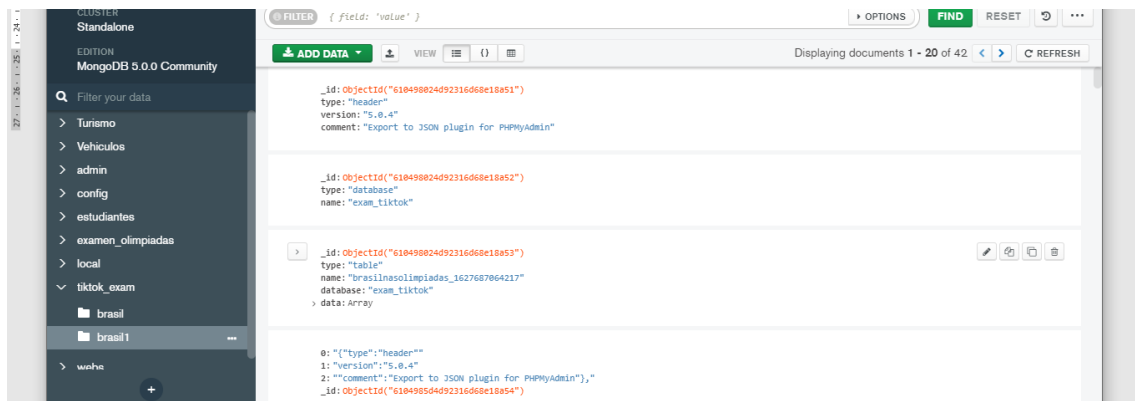
id	secretID	text	createTime	authorMeta.id	authorMeta.sectid
6990578879505206534	6990578879505206534	UM JOGÃO ???? nossos merlinos de ouro brilharam?? B.	1627620983	6954521267132056582	MS4wLjABAAAAnwPExYUDQqFPR0kFOHmdFTth_ICKeFq
6990102350614531333	6990102350614531333	E vamos de TODO APOIO para os merlinos do Vôlei, nã.	1627510030	6954521267132056582	MS4wLjABAAAAnwPExYUDQqFPR0kFOHmdFTth_ICKeFq
6990095519561436421	6990095519561436421	Eu confio no potencial delas ?? #fy #foryou #vôlei.	1627508440	6954521267132056582	MS4wLjABAAAAnwPExYUDQqFPR0kFOHmdFTth_ICKeFq
6990064252530822406	6990064252530822406	O voo perfeito da Butterfly @douglinhas ?	1627501160	6954521267132056582	MS4wLjABAAAAnwPExYUDQqFPR0kFOHmdFTth_ICKeFq
6989998437785750790	6989998437785750790	QUE ORGULHO ?? Em busca do lugar mais alto do pód.	1627485836	6954521267132056582	MS4wLjABAAAAnwPExYUDQqFPR0kFOHmdFTth_ICKeFq

6.PY SUBIR A MONGODB LOCAL

En nuestro phpMyAdmin , buscamos la opción de exportar y escogemos la opción de CSV ,posterior, abrimos een Mongoddb creamos una base de datos luego ponemos en insertar escogemos si es formato csv y listo ya lo importamos .

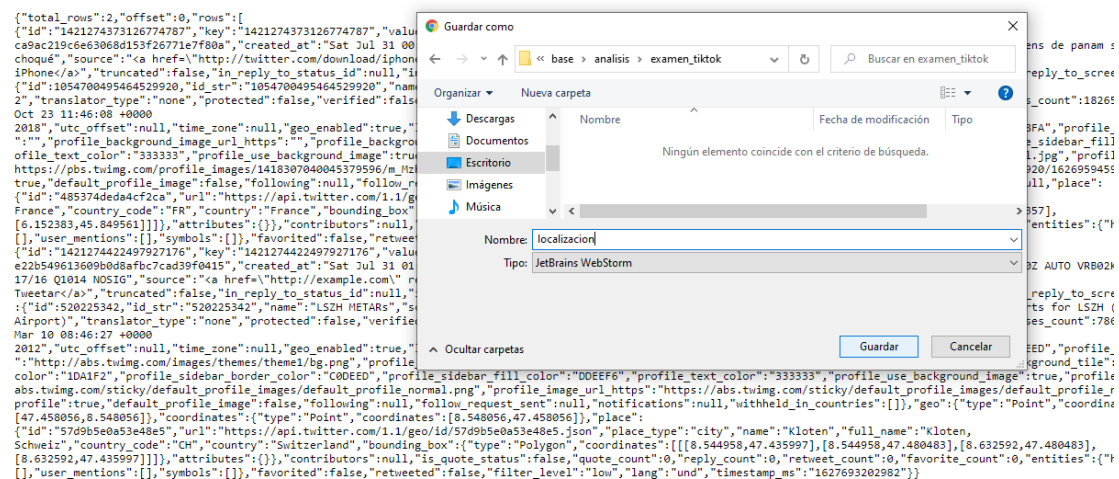


Comprobamos que esta correctamente nuestro csv

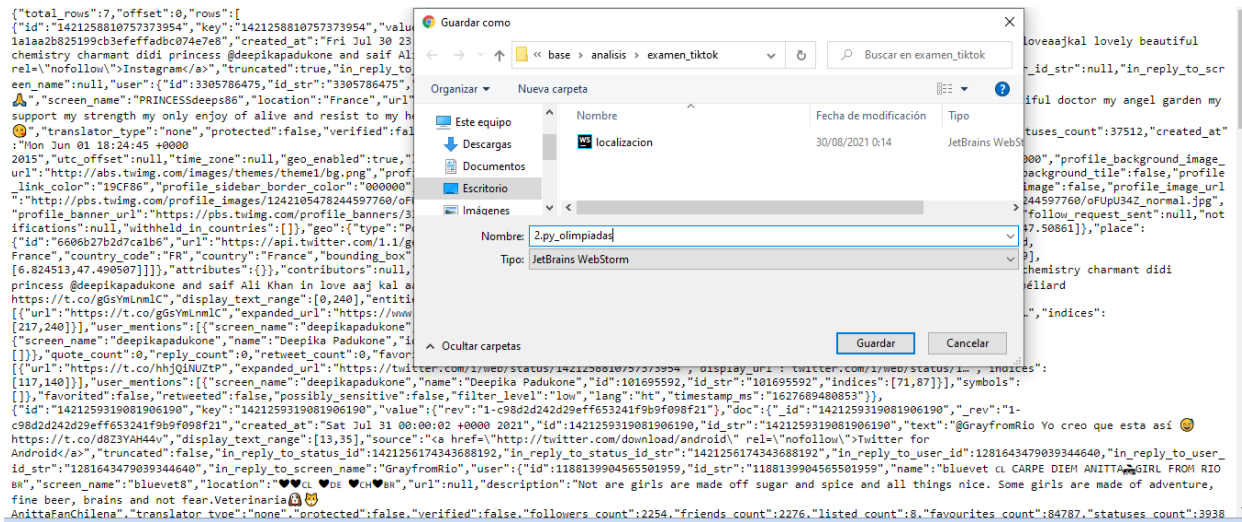


7.PY Almacenar los datos de Localizaciones de Couchdn en Mongodb

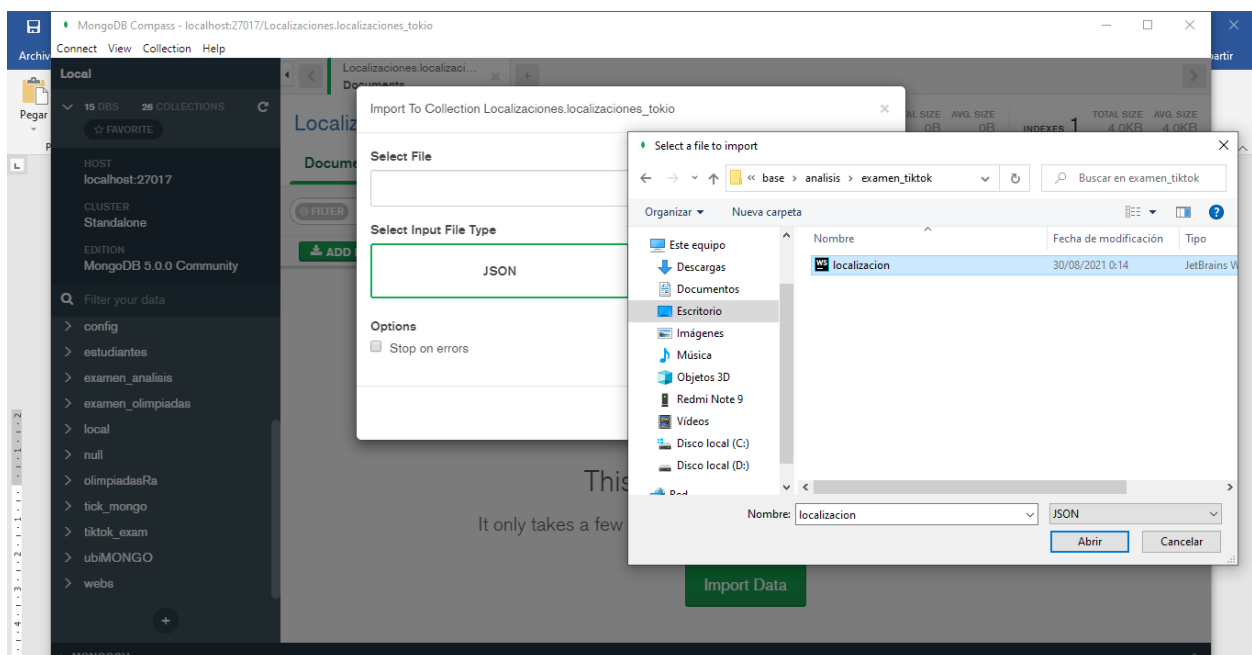
Para ello vamos a la base de datos Couch ponemos en json y podemos ver nuestro Json, lo guardamos como json y lo almacenamos, luego lo subimos a Mongo como json.



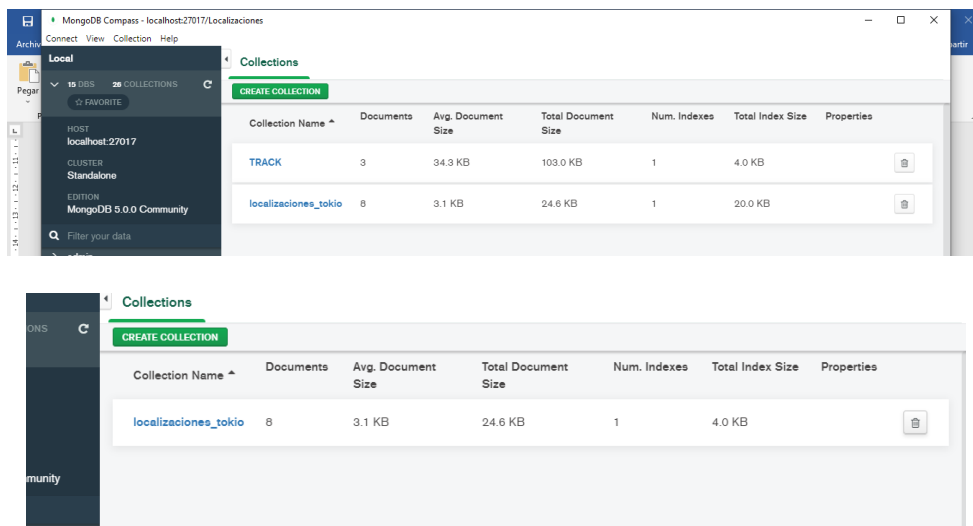
Lo mismo hacemos con TRACK



Importamos a Mongo el json de las Localizaciones de Couchdb

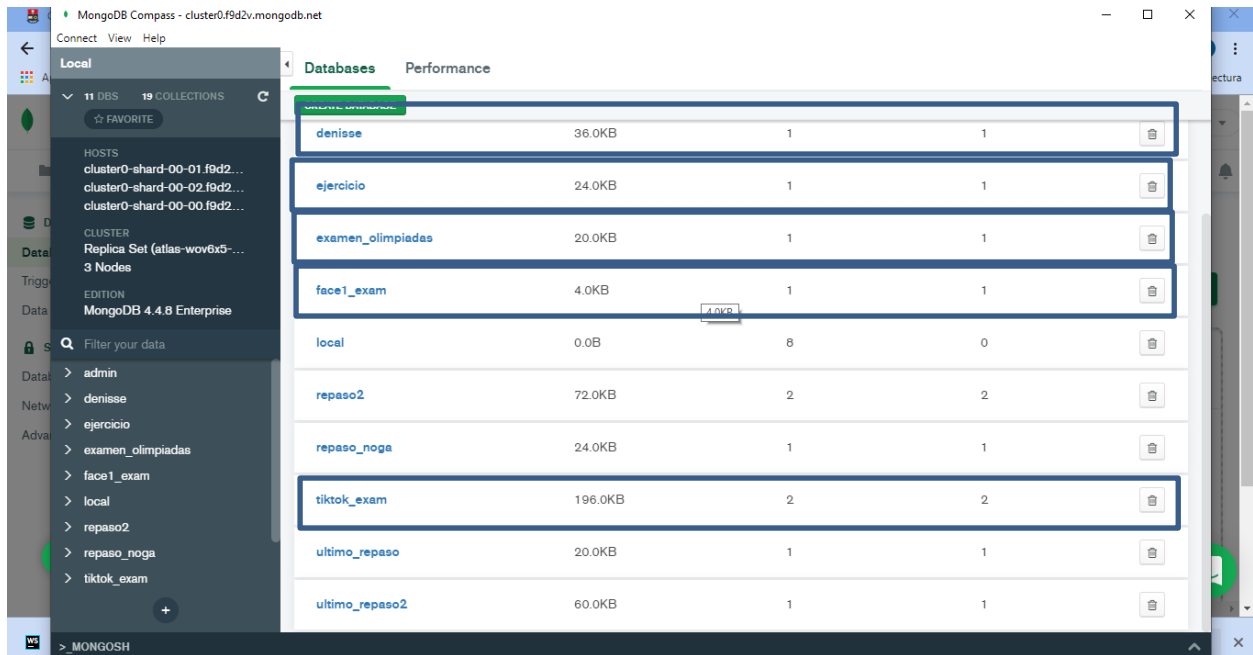


Verificamos que se insertó correctamente



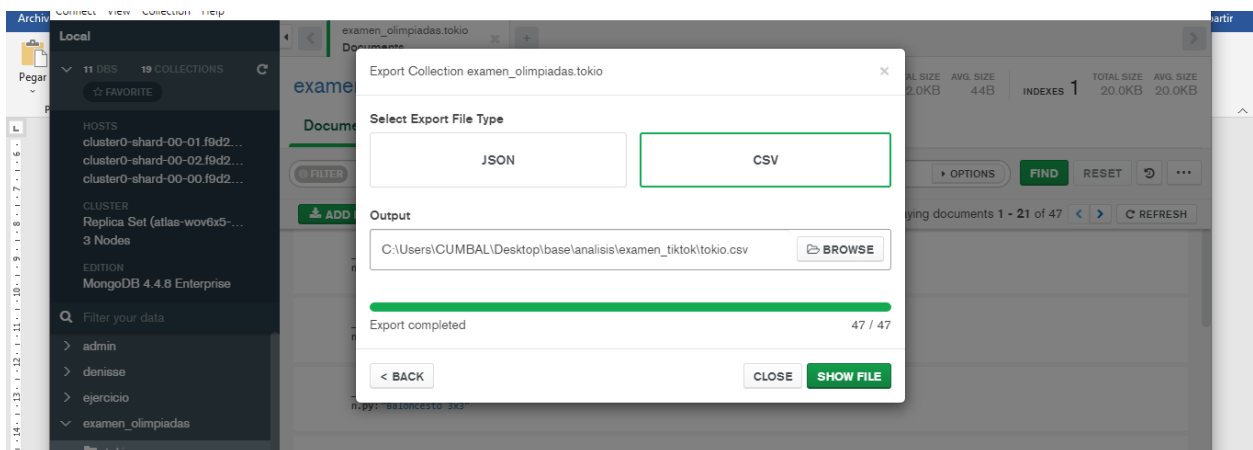
8.PY Subimos todo a Mongo en la NUBE

Lo podemos hacer desde Python o importando los CSV O Json de MongoDB local a Mongo en la Nube

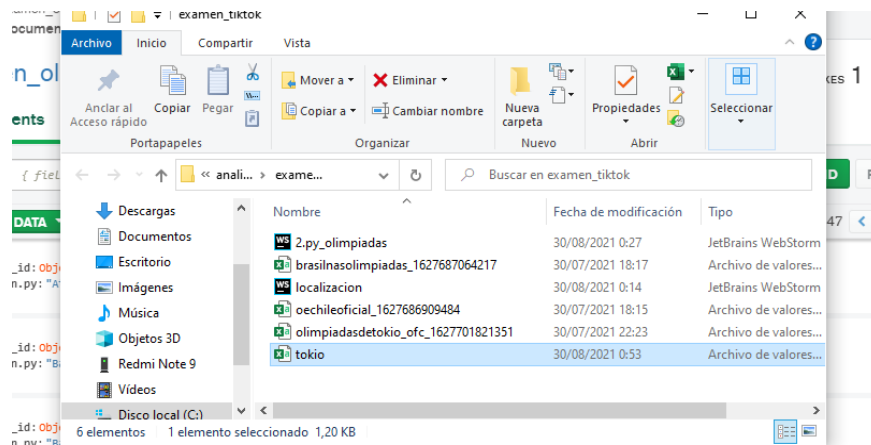


9.py

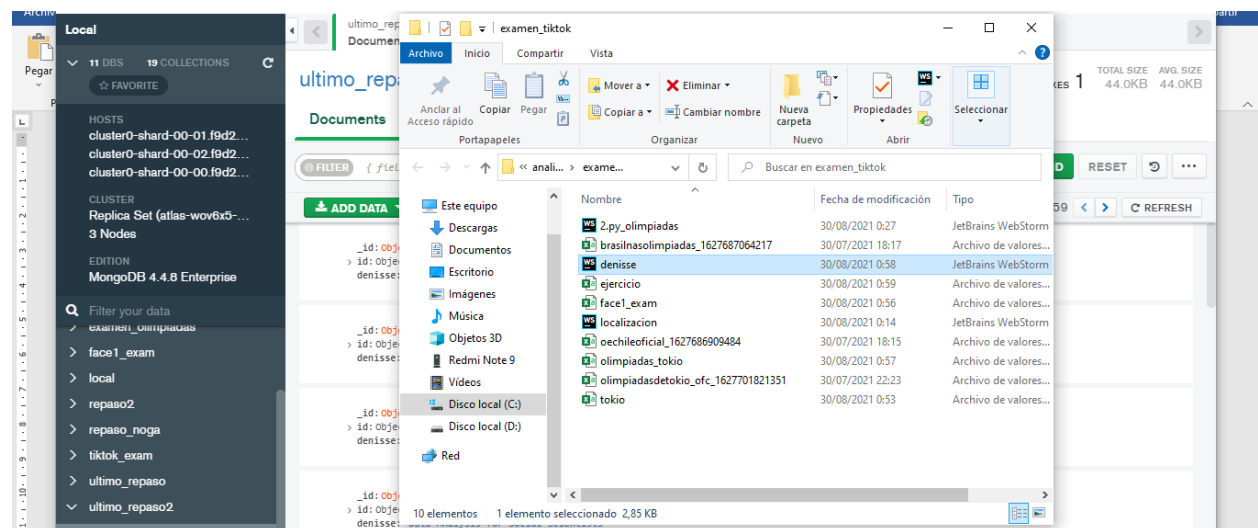
Después que hemos subido todo a la nube procedemos a descargarnos en formato csv



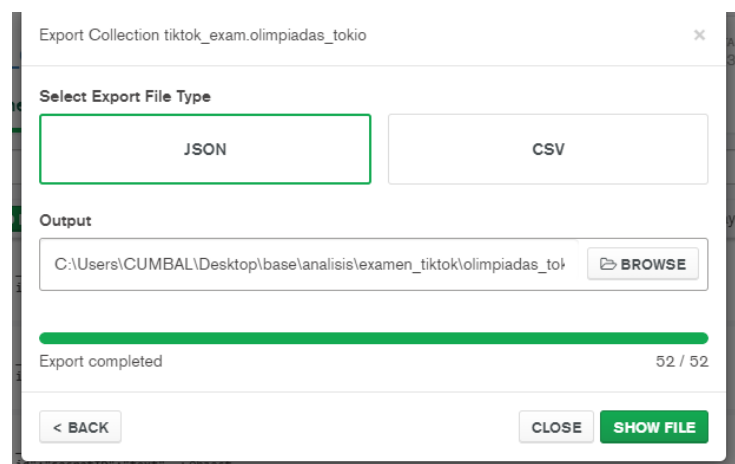
Comprobamos que es el formato que nos hemos exportado















Luego de exportar todos los CSV



10.py Exportamos la información de la base de Datos de Mongolddb y la bajamos con formato json, con la misma lógica que un csv



Verificamos que tenemos nuestro Json

	Nombre	Fecha de modificación	Tipo
0	 2.py_olimpiadas	30/08/2021 0:27	JetBrains WebStorm
	 brasilnasolimpiadas_1627687064217	30/07/2021 18:17	Archivo de valores...
	 denisse	30/08/2021 0:58	JetBrains WebStorm
	 ejercicio	30/08/2021 0:59	Archivo de valores...
	 ejercicio	30/08/2021 1:04	JetBrains WebStorm
15	 face1_exam	30/08/2021 0:56	Archivo de valores...
	 localizacion	30/08/2021 0:14	JetBrains WebStorm
	 oechileoficial_1627686909484	30/07/2021 18:15	Archivo de valores...
	 olimpiadas_tokio	30/08/2021 0:57	Archivo de valores...
	 olimpiadas_tokio	30/08/2021 1:02	JetBrains WebStorm
19	 olimpiadasdetokio_ofc_1627701821351	30/07/2021 22:23	Archivo de valores...
	 tokiio	30/08/2021 0:53	Archivo de valores...
(C:)			