

## Práctica 4: Agregación

0. Iniciar el servidor (mongod -dbpath datos)
1. Bajar del campus los ficheros minitweet.json y miniuser.json
2. Desde un terminal de linux teclear  
mongoimport --db test --collection user --file user.json  
mongoimport --db test --collection tweet --file tweet.json

### Nota:

- No olvidar poner los nombres
- Copiar en solucion.txt solo consultas.

En esta práctica vamos a utilizar las operaciones de agregación `$group` y `$sort`.  
En particular, queremos consultas para:

- 1) Usando la colección tweet, para cada usuario indicar el número total de tweets. El usuario corresponde al campo user.screen. La salida será de la forma:

```
....  
{ "_id" : "FoxNews", "total" : 471 }  
{ "_id" : "DanScavino", "total" : 51 }  
{ "_id" : "KellyannePolls", "total" : 18 }  
{ "_id" : "LouDobbs", "total" : 34 }  
....
```

- 2) La misma consulta pero de forma que muestre en primer lugar el usuario con más tweets. La salida será de la forma:

```
{ "_id" : "TrumpDynastyUSA", "total" : 2000 }  
{ "_id" : "USANEEDSTRUMP", "total" : 1998 }  
{ "_id" : "kupajo322", "total" : 1198 }  
{ "_id" : "HeroAssange", "total" : 794 }  
....
```

- 3) En user, queremos saber el número medio de seguidores de los usuarios.
- 4) En tweet, queremos saber cuantos tweets de media de los usuarios. Sugerencia: enlazar varias agrupaciones
- 5) En user queremos saber cuántos usuario tienen geo\_enabled a true y cuántos geo\_enabled a false
- 6) En user, queremos obtener un listado de las zonas horarias (time\_zone) en un array sin repeticiones. La salida será del estilo

```
["Tehran", "Auckland", "America/New_York", "Karachi", "New Delhi", "London", "Central Time  
(US & Canada)", "America/Chicago", null, "Eastern Time (US & Canada)", "Quito", "Mexico  
City", "Atlantic Time (Canada)", "Arizona", "Indiana (East)", "Pacific Time (US & Canada)",  
"Mountain Time (US & Canada)" ]
```

7) En tweet, para cada usuario (user.screen), queremos solo su primer tweet (primero en orden cronológico, es decir, más antiguo según created\_at). Además, los usuarios también deben mostrarse por ordena alfabético. Las primeras líneas:

```
{ "_id" : "0HOUR1____", "fecha" : ISODate("2016-11-01T16:21:33Z") }
{ "_id" : "2ALAW", "fecha" : ISODate("2016-11-01T23:02:38Z") }
{ "_id" : "ABCPolitics", "fecha" : ISODate("2016-11-01T14:56:31Z") }
...
```

8) En user, queremos saber por cada pareja tweets.RT,tweets.original el número de usuarios con estos datos. La salida será del estilo:

```
{ "_id" : { "RT" : 130, "original" : 172 }, "total" : 1 }
{ "_id" : { "RT" : 7, "original" : 59 }, "total" : 1 }
{ "_id" : { "RT" : 2, "original" : 16 }, "total" : 1 }
```

9) En user queremos obtener ,para cada zona horaria (time\_zone), la lista de todos los usuarios (screen\_name) que tienen esa zona en su perfil.

Por ejemplo para la zona hora "Mountain Time (US & Canada)" obtendremos la lista  
[ "DavidKWilliams", "syeddoha" ]

10) Difícil: En tweet, queremos saber el número total de tweets agrupados por día, hora . Por ejemplo, el resultado nos dirá que el día 13 a la hora 0 hubo 7 tweets.