DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

1) Hallar los tweets del usuario con userid "818839458".

{user_id: ""}
db.tweets.find({ user_id: { $eq: '818839458' } })
db.tweets.find({user_id: '818839458'})
db.tweets.find({"user.id_str": "818839458"})

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

2) Hallar aquellos tweets que tengan más de 500000 retweets.

db.tweets.find({ retweet_count: { $gt: 500000 } })
{retweet_count: {$gt: 500000}}

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

3) Mostrar la cantidad de retweets de los tweets que se hayan hecho desde Argentina o Brasil.

Mirar el campo: place.country (string)

```
{"user.location": {$in: ["Argentina", "Brasil"]}}
```

```
{ $or:[
        {"place.country": "Argentina"},
        {"place.country": "Brasil"}
] }
```

```
{ "place.country": { $in: ["Argentina", "Brasil"] }}
```

```
db.tweets.find(
  {'place.country_code': {$in: [ 'AR', 'BR']}},
  {retweet_count:1}
)
```

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

4) Hallar los usuarios que tengan tweets con 200000 o más retweets y sean en idioma español.

{$and: [{retweet_count: {$gte: 200000}}, {lang: "es"}], {_id: 0, user: 1}}
{$and: [{retweet_count: {$gte: 200000}}, {lang: "es"}]}

db.tweets.find(
  {$and: [{ retweet_count: { $gte: 200000 }},{ lang: 'es'}]},
  {"user.id_str": 1, "user.name": 1}
)

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

5) Mostrar la cantidad de retweets para los tweets que no se hayan hecho en Argentina ni Brasil, pero sí tengan un lugar definido y sean en español.

{ $and: [{"place.country" : {$exists:true}}, {"lang": "es"}, { $nor: [{ "place.country": "Brasil"}, {"place.country": "Argentina"}]}]}
{$and: [{"place.country": {$nin: ["Argentina", "Brasil"]}}, {"place.country": {$ne: null}}, {lang: "es"}], {_id: 0, retweet_count: 1}}
find({"place.country": {$nin: ["Argentina", "Brasil", null]}, lang: "es"}, {retweet_count: 1})

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

6) Mostrar los screen_name de aquellos usuarios que tengan Juan como parte de su nombre.

{"user.screen_name": /Juan/}
{"user.name" : {$regex : "Juan"}} { "user.screen_name": 1, _id: 0}

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

7) Mostrar de los 10 tweets con más retweets, su usuario y la cantidad de retweets.

estas todavia no iban, se puede hacer sin agregacion

find({}, {user: 1, retweet_count: 1}).aggregate({$sort: {retweet_count: -1}}).limit(10) → no sé si está bien, no lo puedo exportar 😐
db.tweets.aggregate([{ $sort: { retweet_count: -1 } }, { $limit: 10 }, { $project: { username: "$user.name", retweet_count: 1 } }])
find({}, {_id: 0, user: 1, retweet_count: 1}).sort({retweet_count: -1}).limit(10)

tipo de ordenamiento
1 ascendente
-1 descendente

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

8) Mostrar de los 10 tweets con más retweets, su usuario y la cantidad de retweets. Ordenar la salida de forma ascendente.

```
db.tweets.aggregate([ { $sort: { retweet_count: -1 } }, { $limit: 10 }, { $sort: { retweet_count: 1 } }, { $project: { username: "$user.name", retweet_count: 1 } } ])
```

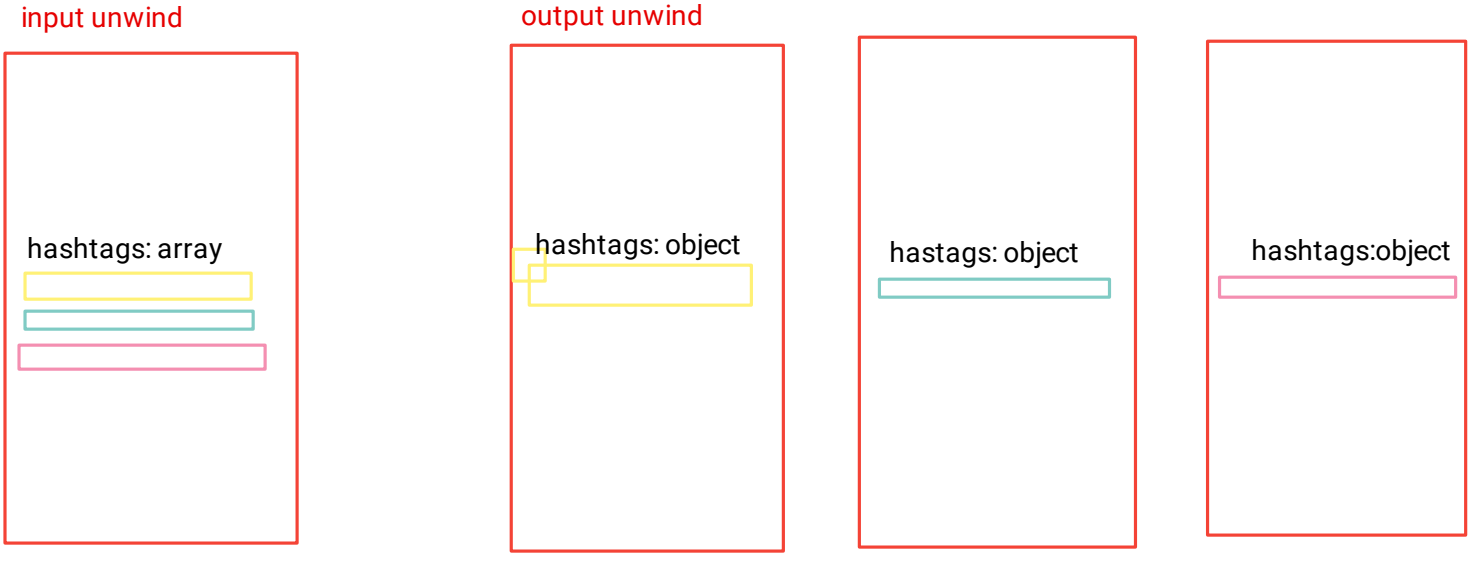
DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

9) Encontrar los 10 hashtags más usados.

```
[
  {
    $unwind:
      {
        path: "$entities.hashtags",
      },
  },
  {
    $group:
      {
        _id: "$entities.hashtags.text",
        counts: {
          $sum: 1,
        },
      },
  },
  {
    $sort:
      {
        counts: -1,
      },
  },
  {
    $limit:
      10,
  },
]
```

reemplaza los arrays en varios objetos de ese dato

por que campo agrupo

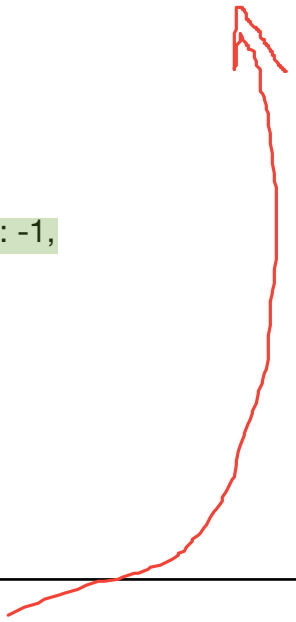


DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

10) Encontrar a los 5 usuarios más mencionados. (les hicieron @)

```
[
  {
    $unwind: {
      path: "$entities.user_mentions",
      preserveNullAndEmptyArrays: false,
    },
  },
  {
    $group: {
      _id: "$entities.user_mentions.id",
      cantidad_menciones: {
        $count: {},
      },
    },
  },
  {
    $sort: {
      cantidad_menciones: -1,
    },
  },
  {
    $limit:
      5,
  },
]
```

es la alternativa de como hacer el de arriba



idem pero con $sum: 1

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

11) Hallar la cantidad de retweets promedio para los tweets que se hayan hecho desde Argentina y aquellos que no.

```
[
  {
    $group: {
      _id: {
        $cond: [
          {
            $eq: ["$place.country", "Argentina"],
          },
          "Argentina",
          "Otros",
        ],
      },
      promedio_retweet: {
        $avg: "$retweet_count",
      },
    },
  },
]
```

output

javascript — Copy code

```
db.tweets.aggregate([
  {
    $group: {
      _id: {
        $cond: [
          { $eq: ["$place.country", "Argentina"] },
          "Argentina",
          "Otros",
        ],
      },
      promedio_retweet: { $avg: "$retweet_count" },
    },
  },
]);
```

El resultado del proceso de agregación será:

json — Copy code

```
[
  { "_id": "Argentina", "promedio_retweet": 12.5 },
  { "_id": "Otros", "promedio_retweet": 6.5 }
]
```

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

12) Por cada usuario obtener una lista de ids de tweets y el largo de la misma.

```
[
  {
    $group: {
      _id: "$user_id",
      nombre_usuario: {
        $first: "$user.name",
      },
      lista_tweets: {
        $push: "$_id",
      }
    },
  },
  {
    $addFields: {
      tamanio_lista: {
        $size: "$lista_tweets",
      },
    },
  },
]
```

1. **`$group`**: Esta es la primera etapa de la consulta. El operador **`$group`** agrupa los documentos de la colección en función de un campo específico, en este caso, el campo "user_id".
   - **`_id`**: Especifica el campo por el cual se realizará la agrupación. En este caso, estamos agrupando por "user_id", lo que significa que todos los tweets del mismo usuario estarán en el mismo grupo.
   - **`nombre_usuario`**: Utiliza el operador de agregación **`$first`** para obtener el nombre de usuario (campo "user.name") del primer tweet encontrado en cada grupo. Suponemos que el nombre de usuario es el mismo para todos los tweets de un usuario en particular, por lo que simplemente tomamos el valor del primer tweet encontrado.
   - **`lista_tweets`**: Utiliza el operador de agregación **`$push`** para crear una lista que contendrá todos los valores del campo "_id" (que representan los IDs de tweets) para cada grupo. La lista se va llenando a medida que se encuentran los tweets del mismo usuario.
2. **`$addFields`**: Esta es la segunda etapa de la consulta y se utiliza para agregar un nuevo campo a cada documento de la colección resultante de la etapa anterior.
   - **`tamanio_lista`**: Utiliza el operador de agregación **`$size`** para calcular el tamaño de la lista de tweets ("lista_tweets") para cada usuario. Esto nos dará el número de tweets que tiene cada usuario.

En resumen, la consulta realiza una operación de agregación que agrupa los tweets por usuario ("user_id"), crea una lista de los IDs de tweets para cada usuario y luego agrega un nuevo campo "tamanio_lista" que representa el número de tweets que tiene cada usuario en su lista. De esta manera, obtenemos la lista de IDs de tweets y el tamaño de la misma para cada usuario.

DB : `mongodb+srv://estudiante:bddfiuba@tweets.ivdga96.mongodb.net/tweets`

13) Hallar la máxima cantidad de retweets *totales* que tuvo algún usuario.

```
[
  { $group: {
      _id: "$user.name",
      retweets_count: {$sum: "$retweet_count"}
  }},
  { $group: {
      _id: null,
      retweets_count: {$max: "$retweets_count"}
  }}
]
```

Supongamos que tenemos los siguientes tweets:

Tweet 1:

```json
{
  "id": 123,
  "user": {
    "name": "user1"
  },
  "retweet_count": 5
}
```

Tweet 2:

1. `$group`: Esta es la primera etapa de la consulta. El operador `$group` agrupa los documentos de la colección en función de un campo específico, en este caso, el campo "user.name".
   - `_id`: Especifica el campo por el cual se realizará la agrupación. En este caso, estamos agrupando por el nombre de usuario ("user.name"), lo que significa que todos los tweets del mismo usuario estarán en el mismo grupo.
   - `retweets_count`: Utiliza el operador de agregación `$sum` para sumar el valor del campo "retweet_count" para cada grupo. Esto calculará la cantidad total de retweets que tuvo cada usuario.

Después de esta etapa, obtendremos los siguientes resultados parciales:

```json
[
  { "_id": "user1", "retweets_count": 13 },
  { "_id": "user2", "retweets_count": 10 }
]
```

1. `$group`: Esta es la segunda etapa de la consulta. Nuevamente, estamos utilizando el operador `$group` para agrupar los resultados obtenidos en la etapa anterior.
   - `_id`: Al establecer `_id` como `null`, estamos combinando todos los documentos resultantes en un solo grupo. Esto nos permitirá realizar una operación de agregación en todos los documentos sin agruparlos por ningún campo en particular.
   - `retweets_count`: Utiliza el operador de agregación `$max` para encontrar el valor máximo de "retweets_count" en todos los documentos agrupados. Esto nos dará la máxima cantidad de retweets totales que tuvo algún usuario.

El resultado final de la consulta será:

```json
[
  { "_id": null, "retweets_count": 13 }
]
```

El resultado indica que la máxima cantidad de retweets totales que tuvo algún usuario es 13. En este caso, el usuario "user1" tuvo 13 retweets en total, que es la cantidad más alta entre todos los usuarios.