

MongoDB

1. La captura de tweets es toda en un mismo día: hallar el hashtag más utilizado en cada hora del día.

```
db.tweets.aggregate([
  {$unwind: '$entities.hashtags'},
  // agrupo por hora y hashtag
  {$group:
    {_id: {'hora':{$hour: '$created_at'},
    'texto': '$entities.hashtags.text'},
    cantidad: {$sum: 1}}},
  // ordeno por hora y sort descendiente
  {$sort: {'_id.hora': 1, 'cantidad': -1}},
  // agrupo por hora y selecciono el de mayor conteo
  {$group: {_id: '$_id.hora',
    hashtag: {$first: '$_id.texto'},
    count: {$first: '$cantidad'}}},
  {$sort: {_id: 1}}
])
```

```
< {
  _id: 0,
  hashtag: 'JuicioALaJusticia',
  count: 327
}
{
  _id: 1,
  hashtag: 'MacriOSaqueoK',
  count: 322
}
{
  _id: 2,
  hashtag: 'MiVidaEresTu',
  count: 384
}
{
  _id: 3,
  hashtag: 'River',
  count: 237
}
{
  _id: 4,
  hashtag: 'UnMundoDe10',
  count: 133
}
{
  _id: 5,
  hashtag: 'MacriOSaqueoK',
  count: 87
}
```

```
{
  _id: 6,
  hashtag: 'FelizAniBersarioRiBer',
  count: 41
}
{
  _id: 7,
  hashtag: 'MacriOSaqueoK',
  count: 23
}
{
  _id: 8,
  hashtag: 'FelizAniBersarioRiBer',
  count: 32
}
{
  _id: 9,
  hashtag: 'HabraConsecuencias',
  count: 72
}
{
  _id: 10,
  hashtag: 'HabraConsecuencias',
  count: 282
}
{
  _id: 11,
  hashtag: 'HabraConsecuencias',
  count: 339
}
```

```
{
  _id: 12,
  hashtag: 'HabraConsecuencias',
  count: 272
}
{
  _id: 13,
  hashtag: 'FelizAniBersarioRiBer',
  count: 359
}
{
  _id: 14,
  hashtag: 'FelizAniBersarioRiBer',
  count: 342
}
{
  _id: 15,
  hashtag: 'NoAGraham',
  count: 322
}
{
  _id: 16,
  hashtag: 'FelizAniBersarioRiBer',
  count: 349
}
{
  _id: 17,
  hashtag: 'SiniestraVidal',
  count: 345
}
```

```
{
  _id: 18,
  hashtag: 'FelizAniBersarioRiBer',
  count: 247
}
{
  _id: 19,
  hashtag: 'SiniestraVidal',
  count: 297
}
```

En caso de haber dos hashtags que se usaron la misma cantidad de veces en una misma hora, se elige el primero según el orden en la bdd por el uso del \$first

2. No distingo entre mayuscula y minúscula. Si quisiera distinguir se tendría que sacar el \$toLowerCase

```
db.tweets.aggregate([
  { $unwind: "$entities.hashtags" },
  // Genera un doc con un arreglo de letras en minúscula del hashtag y mantiene el
  hashtag para referencia
  { $project: {
    letra: {
      $map: {
        input: { $range: [0, { $strLenCP: "$entities.hashtags.text" }] },
        as: "i",
        in: { $toLowerCase: { $substrCP: ["$entities.hashtags.text", "$$i", 1] }}
      }
    },
    hashtag: "$entities.hashtags.text"
  }},

  { $unwind: "$letra" },

  // agrupa por letra, cuenta la cant de veces que aparece la letra y guarda en un
  set cada hashtag con esa letra
  { $group: {
    _id: "$letra",
    count: { $sum: 1 },
    hashtags: { $addToSet: "$hashtag" }
  }}
])
```

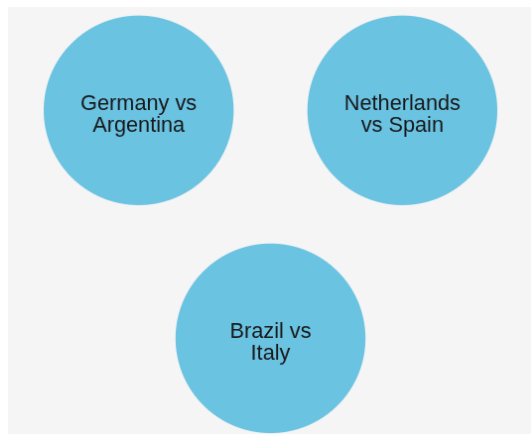
```
< {
  _id: 'a',
  count: 295667,
  hashtags: [
    'JuntosSomosJuan',
    'musical',
    '82AñosSSN',
    'PASALO',
    'Argentores',
    'HolaGrupo',
    'sicilia',
    'LWA_jp',
    'Cuidatufertilidad',
    'NiMacri',
    'OEANoAlAborto',
    'MalNacidos',
    'MalvinasNosUne',
    'MOJIFRASES',
    'SanMartíndeLosAndes',
```

```
    'BateaMahuida',
    'LoVisteEnADN',
    'JuvenilesAPleno',
    'Patético',
    'Reserva',
    'IncubadoraRUS',
    'MovistarF1',
    'SolitaySola',
    ... 30706 more items
  ]
}
```

Muestro solo un pedazo de la respuesta (solo letra **aA**) porque es muy larga

3. Considero finales donde no hubieron goles (home_score = 0, away_score = 0) o donde minute_regulation > 90 (o sea finales sin goles ni en el primer ni en el segundo tiempo)

```
MATCH (m:Match)
WHERE m.stage = "final"
  AND NOT EXISTS {
    MATCH (:Player)-[s:SCORED_IN]->(m)
    WHERE s.minute_regulation <= 90
  }
RETURN m;
```



4. De la documentacion: COLLECT does not handle null values in the same way that the aggregating function collect() does. The collect() function automatically removes null values. COLLECT will not remove null values automatically.
Solo se elige uno si hay empate de goleadores

```
MATCH (p:Player)-[s:SCORED_IN]->(m:Match)
WHERE s.own_goal = false
WITH date(m.date).year AS year, p, count(s) AS goles
ORDER BY year, goles DESC
WITH year, collect({jugador: p, goles: goles}) AS lista
RETURN year, lista[0].jugador.given_name as Nombre,
lista[0].jugador.family_name AS Apellido, lista[0].goles AS Goles;
```