



DATA ANALYTICS – MONGODB – SPRINT 5



Ana Claudia da Costa

Índice

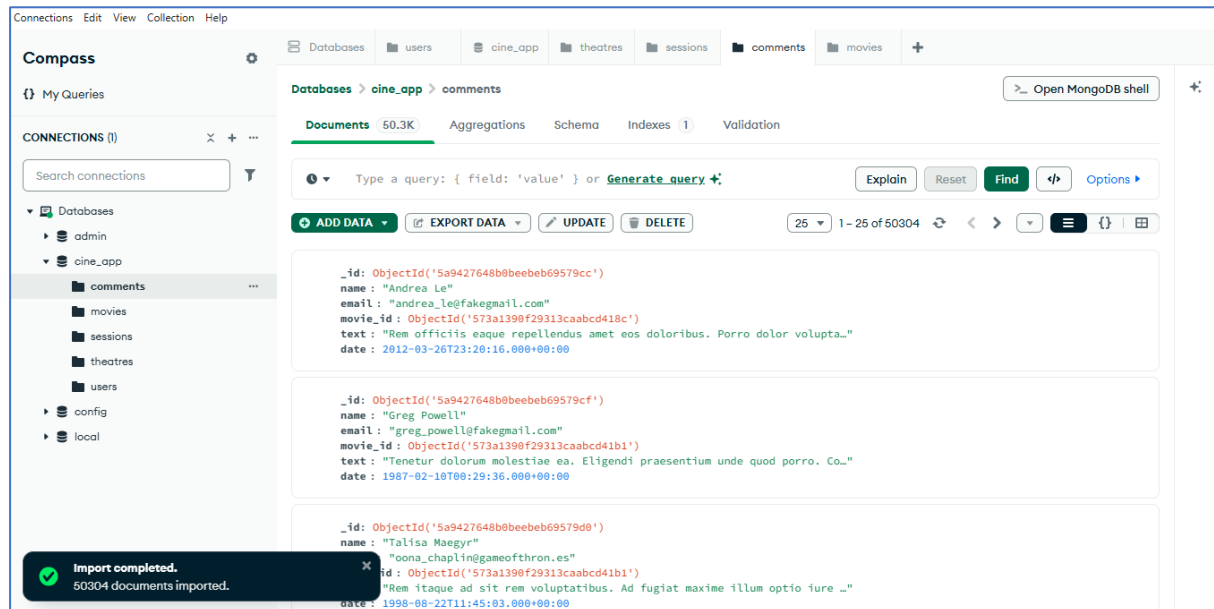
Sprint 5.....	2
Nivel 1	2
Ejercicio 1	2
Ejercicio 2	6
Ejercicio 3	7
Nivel 2	8
Ejercicio 1	8
Ejercicio 2	8
Nivel 3	10
Ejercicio 1	10
Ejercicio 2	11
Revisión.....	11

Data Analytics – MongoDB – Sprint 5

Sprint 5

Nivel 1

Crea una base de datos con MongoDB utilizando como colecciones los archivos adjuntos.



Ejercicio 1

Muestra los 2 primeros comentarios que aparecen en la base de datos.

```
> use cine_app
switched to db cine_app
> db.comments.find().sort({ date: 1 }).limit(2)
< {
  _id: ObjectId('5a9427648b0beebeb695e67e'),
  name: 'Mercedes Tyler',
  email: 'mercedes_tyler@fakegmail.com',
  movie_id: ObjectId('573a1399f29313caabcec3ce'),
  text: 'Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cumque animi. Quia nam doloribus corrupti id v',
  date: 1970-01-01T01:07:09.000Z
}
{
  _id: ObjectId('5a9427658b0beebeb6969efe'),
  name: 'Jon Snow',
  email: 'kit_harington@gameofthron.es',
  movie_id: ObjectId('573a13a4f29313caabd117df'),
  text: 'Dolorem animi tempora ullam quas. Iusto nobis reprehenderit aspernatur cupiditate a tempore. Commodi alias expedit',
  date: 1970-01-01T09:37:49.000Z
}
cine_app>
```

Data Analytics – MongoDB – Sprint 5

Databases > cine_app > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

Generate query + Explain Reset Find </> Options ▾

Project { field: 0 }

Sort { date: 1 } Max Time MS 60000

Collation { locale: 'simple' } Skip 0 Limit 2

Index Hint { field: -1 }

ADD DATA EXPORT DATA UPDATE DELETE ? 25 1 - 2 of 2 < > {}

```
{
  "_id": ObjectId("5a9427648b0beebe695e67e"),
  "name": "Mercedes Tyler",
  "email": "mercedes_tyler@fakegmail.com",
  "movie_id": ObjectId("573a1399f29313caabce3ce"),
  "text": "Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cu...",
  "date": 1970-01-01T01:07:09.000+00:00
}
```

```
{
  "_id": ObjectId("5a9427658b0beebe6969efe"),
  "name": "Jon Snow",
  "email": "kit_harington@gameofthron.es",
  "movie_id": ObjectId("573a13a4f29313caabd117df"),
  "text": "Dolorem animi tempora ullam quas. Iusto nobis reprehenderit aspernatur...",
  "date": 1970-01-01T09:37:49.000+00:00
}
```

¿Cuántos usuarios tenemos registrados?

```
> db.users.countDocuments()
< 185
cine_app>
```

Databases > cine_app > users

Documents 185 Aggregations Schema Indexes 1 Validation

\$count Generate aggregation + ? Explain Export Run Options ▸

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 \$count

```
1 //**
2 * Provide the field name for the count.
3 */
4 'Users'
```

Output preview after \$count stage (Sample of 1 document)

Users : 185

¿Cuántos cines existen en el estado de California?

```
> db.theatres.countDocuments({"location.address.state": "CA"})
< 169
cine_app>
```

Author: Ana Cláudia da Costa

Data Analytics – MongoDB – Sprint 5

Databases > cine_app > theatres

Documents 1.6K Aggregations Schema Indexes 1 Validation

Generate aggregation Explain Export Run Options

Untitled – modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7     {
8       "location.address.state": "CA"
9     }
10  },
11  {
12    $count:
13      /**
14       * Provide the field name for the count.
15       */
16    "California State Cinemas"
17  }
18 ]
```

PIPELINE OUTPUT PREVIEW

Sample of 1 document

California State Cinemas : 169

¿Cuál fue el primer usuario en registrarse?

```
> db.users.find().sort({_id: 1}).limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crLJ1Vu'
}
cine_app>
```

Databases > cine_app > users

Documents 185 Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options

Project { field: 0 }

Sort { _id: 1 }

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 1

ADD DATA EXPORT DATA UPDATE DELETE

25 1 – 1 of 1

```
_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crLJ1Vu"
```

Author: Ana Cláudia da Costa

Data Analytics – MongoDB – Sprint 5

¿Cuántas películas de comedia existen en nuestra base de datos?

```
> db.movies.countDocuments({genres: "Comedy"})
< 7024
cine_app>
```

The screenshot shows the MongoDB Atlas interface for the 'cine_app' database, specifically the 'movies' collection. The 'Aggregations' tab is selected, showing a pipeline with two stages: '\$match' and '\$count'. The '\$match' stage has a query filter for 'genres: "Comedy"'. The '\$count' stage is configured to count the number of documents. The pipeline output preview shows a single document: 'Comedy Movies : 7024'.

Databases > cine_app > movies Open MongoDB shell

Documents 23.5K **Aggregations** Schema Indexes 1 Validation

\$match \$count Generate aggregation Explain Export Run Options

Untitled – modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7     {
8       genres: "Comedy"
9     }
10  },
11  {
12    $count:
13      /**
14       * Provide the field name for the count.
15       */
16      "Comedy Movies"
17  }
18 ]
```

PIPELINE OUTPUT PREVIEW OUTPUT OPTIONS

Sample of 1 document

Comedy Movies : 7024

Data Analytics – MongoDB – Sprint 5

Nivel 1

Ejercicio 2

Muéstrame todos los documentos de las películas producidas en 1932, pero que el género sea drama o estén en francés.

```
> db.movies.find({year: 1932, $or: [{ genres: "Drama", languages: "French"}]})
< {
  _id: ObjectId('573a1392f29313caabcd9a76'),
  plot: 'In this romance a librarian takes a cruise and falls for an unobtainable man, a district attorney married to a cripp
  genres: [
    'Drama',
    'Romance'
  ],
  runtime: 85,
  rated: 'APPROVED',
  cast: [
    'Barbara Stanwyck',
    'Adolphe Menjou',
    'Ralph Bellamy',
    'Dorothy Peterson'
  ],
  num_mflix_comments: 3,
  poster: 'https://m.media-amazon.com/images/M/MV5BMjEzNTU5NTk2N15BMl5BanBnXkFtZTgwMzA2NTAwMjE@._V1_SY1000_SX677_AL_.jpg',
  title: 'Forbidden',
  fullplot: "On a cruise to Cuba, Lulu Smith falls in love with Bob Grover. Back home, she breaks off the romance when he tel
```

The screenshot shows the MongoDB Atlas web interface. At the top, the breadcrumb navigation is 'Databases > cine_app > movies'. There's a button 'Open MongoDB shell'. Below this, there are tabs for 'Documents' (23.5K), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'. The 'Documents' tab is active. In the query editor, the query is: `{year: 1932, $or: [{ genres: "Drama", languages: "French"}]}`. There are buttons for 'Generate query', 'Explain', 'Reset', 'Find', and 'Options'. Below the query, there are fields for 'Project', 'Sort', 'Collation', 'Index Hint', 'Skip', 'Limit', and 'Max Time MS'. The 'Find' button is highlighted. Below the query editor, there's a section for 'EXPORT DATA' with a dropdown menu. Below this, there's a list of documents. The first document is: `{_id: ObjectId('573a1392f29313caabcd9a76'), genres: ['Drama', 'Romance'], languages: ['English', 'French'], year: 1932}`. The second document is: `{_id: ObjectId('573a1392f29313caabcd9a7e'), genres: ['Drama', 'Horror'], languages: ['English', 'French', 'German'], year: 1932}`.

Data Analytics – MongoDB – Sprint 5

Ejercicio 3

Muéstrame todos los documentos de películas estadounidenses que tengan entre 5 y 9 premios que fueron producidas entre 2012 y 2014

```
> db.movies.find({ countries: "USA", "awards.wins": { $gte: 5, $lte: 9 }, year: { $gte: 2012, $lte: 2014 } })
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for
  imdb: {
    rating: 7.4,
    votes: 211230,
    id: 359950
  },
  year: 2013,
  plot: "When his job along with that of his co-worker are threatened, Walter takes action in the real world embarking on a g
  genres: [
    'Adventure',
    'Comedy',
    'Drama'
  ],
  rated: 'PG',
  metacritic: 54,
  title: 'The Secret Life of Walter Mitty',
  lastupdated: '2015-08-31 00:10:51.747000000',
  languages: [
    'English',
    'Spanish',
    'Icelandic'
  ]
}
```

Databases > cine_app > movies

Documents 23.5K Aggregations Schema Indexes 1 Validation

Generate query ⚡ Explain Reset Find </> Options ▾

Query:

```
{
  countries: "USA",
  "awards.wins": { $gte: 5, $lte: 9 },
  year: { $gte: 2012, $lte: 2014 }
}
```

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]]

Collation: { locale: 'simple' }

Index Hint: { field: -1 }

Max Time MS: 60000

Skip: 0 Limit: 0

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 25 of 166

Results:

```
{
  genres: Array (3)
  rated: "PG"
  metacritic: 54
  title: "The Secret Life of Walter Mitty"
  lastupdated: "2015-08-31 00:10:51.747000000"
  languages: Array (3)
  writers: Array (3)
  type: "movie"
  tomatoes: Object
  poster: "https://m.media-amazon.com/images/M/MV5BODYwNDYxNDk1Nl5BMl5BanBnXkFtZT..."
  num_mflix_comments: 1
  released: 2013-12-25T00:00:00.000+00:00
}
```


Data Analytics – MongoDB – Sprint 5

Nivel 2

Ejercicio 1

Cuenta cuántos comentarios escribe un usuario que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico.

```
> db.comments.countDocuments({ email: { $regex: /gameofthron\.es$/i}})
< 22841
cine_app>
```

The screenshot shows the MongoDB Atlas Aggregations Builder interface. The breadcrumb navigation indicates the database is 'cine_app' and the collection is 'comments'. The 'Aggregations' tab is selected, showing a pipeline with two stages: '\$match' and '\$count'. The '\$match' stage has a regex filter for the email domain. The '\$count' stage is configured to count documents where the field 'Domain_GameOfThron' matches the value '22841'. The 'PIPELINE OUTPUT PREVIEW' on the right shows a sample document with 'Domain_GameOfThron : 22841'.

```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7     {
8       email: {
9         $regex: /gameofthron\.es$/i
10      }
11    },
12  },
13  {
14    $count:
15      /**
16       * Provide the field name for the count.
17       */
18      "Domain_GameOfThron"
19  }
20 ]
```

PIPELINE OUTPUT PREVIEW
Sample of 1 document

Domain_GameOfThron : 22841

Ejercicio 2

¿Cuántos cines existen en cada código postal situados dentro del estado Washington DC (DC)?

```
> db.theatres.aggregate([{$match: {"location.address.state": "DC"}}, {$group: {_id: "$location.address.zipcode", Total_zipcode:
< {
  _id: '20002',
  Total_zipcode: 1
}
{
  _id: '20010',
  Total_zipcode: 1
}
{
  _id: '20016',
  Total_zipcode: 1
}
]
cine_app>
```

Data Analytics – MongoDB – Sprint 5

Databases > cine_app > theatres Open MongoDB shell

Documents 1.6K Aggregations Schema Indexes 1 Validation

\$match \$group Generate aggregation Explain Export Run Options

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7     {
8       "location.address.state": "DC"
9     }
10  },
11  {
12    $group:
13      /**
14       * _id: The id of the group.
15       * fieldN: The first field name.
16       */
17    {
18      _id: "$location.address.zipcode",
19      Total_zipcode: {
20        $sum: 1
21      }
22    }
23  }
24 ]
```

PIPELINE OUTPUT PREVIEW OUTPUT OPTIONS

Sample of 3 documents

_id: "20002"	Total_zipcode: 1
_id: "20010"	Total_zipcode: 1
_id: "20016"	Total_zipcode: 1

Data Analytics – MongoDB – Sprint 5

Nivel 3

Ejercicio 1

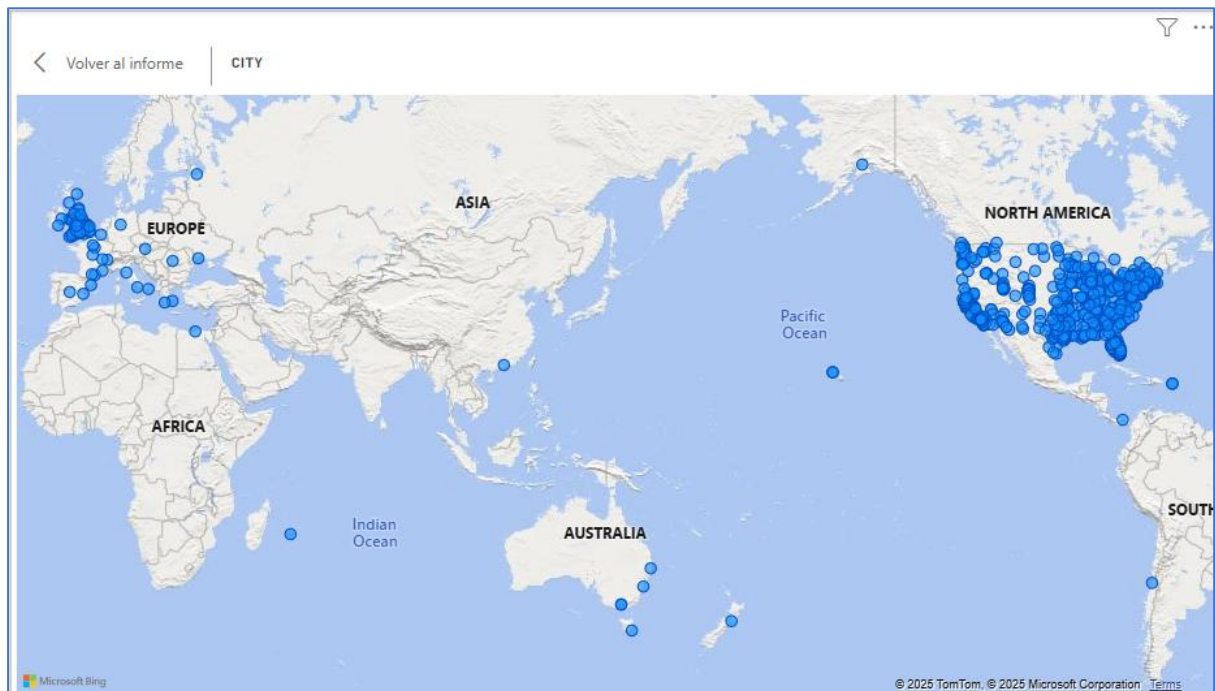
Encuentra todas las películas dirigidas por John Landis con una puntuación IMDb (Internet Movie Database) de entre 7,5 y 8.

```
> db.movies.find({ directors: "John Landis", "imdb.rating": {$gte: 7.5, $lte: 8}})
< {
  _id: ObjectId('573a1397f29313caabce6d94'),
  fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of white, anglo
  imdb: {
    rating: 7.6,
    votes: 84834,
    id: 77975
  },
  year: 1978,
  plot: 'At a 1962 college, Dean Vernon Wormer is determined to expel the entire Delta Tau Chi Fraternity, but those troubl
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ],
  writers: [
```

The screenshot shows the MongoDB Atlas web interface. The top navigation bar indicates the database is 'cine_app' and the collection is 'movies'. The 'Aggregations' tab is selected. The query editor shows an aggregation pipeline with two stages: a '\$match' stage with a query for movies directed by 'John Landis', and another '\$match' stage with a query for movies with an IMDb rating between 7.5 and 8. The 'PIPELINE OUTPUT PREVIEW' section shows a sample of 4 documents, with the first document displayed, matching the JSON output shown in the first block. The interface includes various toolbars for saving, creating new queries, exporting to different languages, and running the aggregation.

Ejercicio 2

Muestra en un mapa la ubicación de todos los teatros de la base de datos.



Revisión

Revisado por: