

SPRINT 5 - MONGO DB - TAREAS

MARIANA P

Descripción

Trabajaremos con una base de datos que contiene colecciones relacionadas con una aplicación de entretenimiento cinematográfico:

- users: Almacena información de usuarios/as, incluyendo nombres, emails y contraseñas cifradas.
- theatres: Contiene datos de cines, como ID, ubicación (dirección y coordenadas geográficas).
- sesiones: Guarda sesiones de usuario, incluyendo ID de usuario y tokens JWT para la autenticación.
- movies: Incluye detalles de películas, como trama, géneros, duración, elenco, comentarios, año de lanzamiento, directores, clasificación y premios.
- comments: Almacena comentarios de usuarios/as sobre películas, con información del autor/a del comentario, ID de la película, texto del comentario y la fecha.
- Llevarás a cabo algunas consultas que te pide el cliente/a, el cual está midiendo si serás capaz o no de hacerte cargo de la parte analítica del proyecto vinculado con su base de datos.

Nivel 1

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



The screenshot shows a terminal window with two panes. The left pane displays the file structure of a MongoDB database named 'sprint_5_server'. The right pane shows the MongoDB shell commands used to create the database and its collections.

```
[0m[31m[1m> [22m[39m[90m 1 |[39m create database [90m [31m[1m|[22m[39m[0m
>
>
>
>
>
> use sprint_5
< switched to db sprint_5
> db.createCollection ("users")
< { ok: 1 }
> db.createCollection ("comments")
< { ok: 1 }
> db.createCollection ("movies")
< { ok: 1 }
> db.createCollection ("sessions")
< { ok: 1 }
sprint_5>
```

- COMENTARIOS:

Aquí tuve que crear la nueva base de datos sin comandos y aproveché para crear la primera colección “theaters”. Por eso no aparece junto a los demás comandos.

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible under the 'sprint_5_server' database, including the 'comments' collection which is currently selected. The main pane displays the results of a query on the 'comments' collection, showing three document snippets. The first snippet contains fields like _id, name, email, movie_id, text, and date. The second snippet contains similar fields. The third snippet is partially visible. At the bottom left, a modal window indicates that the import process is completed, having imported 50304 documents.

```

_id: ObjectId('5a9427648b0beebeb69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd418c')
text: "Rem officiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00

_id: ObjectId('5a9427648b0beebeb69579cf')
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd41b1')
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00

_id: ObjectId('5a9427648b0beebeb69579d0')
name: "Talisa Maegyr"
email: "oona_chaplin@gameofthron.es"
movie_id: ObjectId('573a1390f29313caabcd41b1')
text: "Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..."
date: 1998-08-22T11:45:03.000+00:00

```

Import completed.
50304 documents imported.

- COMENTARIOS: luego añadí los datos de todas las colecciones una a una a través de “import data”.

Ejercicio 1

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

```
> db.createCollection ("sessions")
< { ok: 1 }

> db.comments.find().limit(2)
< [
    {
        _id: ObjectId('5a9427648b0beebeb69579cc'),
        name: 'Andrea Le',
        email: 'andrea_le@fakegmail.com',
        movie_id: ObjectId('573a1390f29313caabcd418c'),
        text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum volup',
        date: 2012-03-26T23:20:16.000Z
    }
    {
        _id: ObjectId('5a9427648b0beebeb69579cf'),
        name: 'Greg Powell',
        email: 'greg_powell@fakegmail.com',
        movie_id: ObjectId('573a1390f29313caabcd41b1'),
        text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Commodo nis',
        date: 1987-02-10T00:29:36.000Z
    }
]
```

- COMENTARIOS: la función “find” para buscar sumada a la función “limit 2” para encontrar los 2 primeros registros.

¿Cuántos usuarios tenemos registrados?

185

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

¿Cuántos cines hay en el estado de California?

169

```
_id: ObjectId('59a47286cfa9a3a73e51e7f4')
theaterId : 1199
location : Object
  address : Object
    street1 : "9652 Main Street"
    city : "Fairfax"
    state : "VA"
    zipcode : "22031"
geo : Object
```

```
_id: ObjectId('59a47286cfa9a3a73e51e7f5')
theaterId : 133
location : Object
  address : Object
    street1 : "1901 Arden Way"
    city : "Sacramento"
    state : "CA"
    zipcode : "95815"
geo : Object
```

```
> db.theathers.find({"location.address.state":"CA"}).count()
< 169
sprint_5>
```

- COMENTARIOS: para segmentar sólo los de CA tenemos que definir el estado (state) que está dentro de “address” que a la vez se encuentra dentro de “location”.

¿Cuál fue el primer usuario/a en registrarse?

```
> db.users.find().sort({_id: 1})
< [
  {
    _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
    name: 'Ned Stark',
    email: 'sean_bean@gomeofthron.es',
    password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIj9RJAqMUQ74crlJ1Vu'
  }
]
```

- COMENTARIOS: buscar (find) por el id = 1 (sort) para encontrar el primero.

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

7024

```
> db.movies.distinct("genres")
< [
    'Action',      'Adventure', 'Animation',
    'Biography',   'Comedy',     'Crime',
    'Documentary', 'Drama',     'Family',
    'Fantasy',     'Film-Noir',  'History',
    'Horror',      'Music',     'Musical',
    'Mystery',     'News',      'Romance',
    'Sci-Fi',       'Short',    'Sport',
    'Talk-Show',    'Thriller',  'War',
    'Western'
]
```

```
> db.movies.find({"genres":"Comedy"}).count()
< 7024
```

- **COMENTARIOS:** Primero tengo que conocer cómo se escribe exactamente el nombre de los géneros, para entonces copiar y pegar en el comando y así filtrar sólo los de comedia (Comedy).

Ejercicio 2

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

- **COMENTARIOS:** aquí tengo que usar las condiciones “or” y “and” para encontrar las dos casuísticas de género (Drama) y idioma (French).

```
> db.movies.find({$and: [ {$or: [{genres: "Drama"}, {languages: "French"}]}, {year: 1932}]})  
< [  
  {_id: ObjectId('573a1391f29313caabcd9458'),  
   plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing  
   runtime: 55,  
   rated: 'UNRATED',  
   cast: [  
     'Enrique Rivero',  
     'Elizabeth Lee Miller',  
     'Pauline Carton',  
     'Odette Talazac'  
   ],  
   num_mflix_comments: 1,  
   poster: 'https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWETYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0Y1  
   title: 'The Blood of a Poet',  
   lastupdated: '2015-09-16 13:13:05.537000000',  
   languages: [  
     'French'  
   ],  
   _id: ObjectId('573a1392f29313caabcd99a3'),  
   plot: 'Junta is hated by the people in the village where she lives, especially by the women, w  
   genres: [  
     'Drama',  
     'Fantasy',  
     'Mystery'  
   ],  
   runtime: 85,  
   cast: [  
     'Leni Riefenstahl',  
     'Mathias Wieman',  
     'Beni Fèhrer',  
     'Max Holzboer'  
   ],  
   poster: 'https://m.media-amazon.com/images/M/MV5BNTQ1NTMzMzMTQtODIyYS00MTAxLWE1NTgtZGFkZTE2YmZkZn  
   title: 'The Blue Light',  
   fullplot: 'Junta is hated by the people in the village where she lives, especially by the women  
   languages: [  
     'German',  
     'Italian'  
   ]  
>
```

```
_id: ObjectId('573a1392f29313caabcd99a3'),  
plot: 'Junta is hated by the people in the village where she lives, especially by the women, w  
genres: [  
  'Drama',  
  'Fantasy',  
  'Mystery'  
],  
runtime: 85,  
cast: [  
  'Leni Riefenstahl',  
  'Mathias Wieman',  
  'Beni Fèhrer',  
  'Max Holzboer'  
],  
poster: 'https://m.media-amazon.com/images/M/MV5BNTQ1NTMzMzMTQtODIyYS00MTAxLWE1NTgtZGFkZTE2YmZkZn  
title: 'The Blue Light',  
fullplot: 'Junta is hated by the people in the village where she lives, especially by the women  
languages: [  
  'German',  
  'Italian'  
]
```

Ejercicio 3

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

```
> directors : Array (1)
  rated : "UNRATED"
  awards : Object
    wins : 1
    nominations : 0
    text : "1 win."
  lastupdated : "2015-08-26 00:03:50.133000000"
  year : 1893
> imdb : Object
  type : "movie"
> tomatoes : Object
```

```
> 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 the
  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 global

  poster: 'https://m.media-amazon.com/images/M/MV5BODYwNDYxNDk1Nl5BMl5BanBnXkFtZTgwOTAwMTk2MDE@._V1_SY1000_SX677_AL_.jpg',
  num_mflix_comments: 1,
  released: 2013-12-25T00:00:00.000Z,
  awards: {
    wins: 6,
    nominations: 13,
    text: '6 wins & 13 nominations.'
  },
  countries: [
    'USA',
    'Canada'
  ],
  cast: [
    'Ben Stiller',
    'Kristen Wiig',
    'Jon Daly',
    'Kathryn Hahn'
```

Nivel 2

Ejercicio 1

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

```
> db.comments.find({ email: /GAMEOFTHRON\.ES$/i}).count()
< 22841
```

- COMENTARIOS: aquí tengo que buscar solamente correos con [GAMEOFTHRON.ES](#). Uso el / para separar la palabra y el “\$” para limitar la búsqueda a ES y el “i” para no hacer distinción de caracteres con mayúsculas o minúsculas.

Ejercicio 2

¿Cuántos cines hay en cada código postal situados dentro del estado Washington D. C. (DC)?

- COMENTARIOS: aquí tengo que organizar los datos basado otra vez en “location.address.state” dentro de la colección cines “Theaters” pero ahora en DC y agruparlos por su código postal (zipcode).
- 1. Filtro todos los documentos donde el estado (state) es "DC".
- 2. Agrupo los documentos por el código postal (zipcode).
- 3. Sumo cuántos teatros hay en cada código postal.
- 4. Organizo en orden decreciente

```
db.theaters.aggregate([
  {$match: {"location.address.state": "DC"}},
  {$group: {"_id": "$location.address.zipcode",
            count: {$sum: 1}}},
  { $sort: { count: -1 } }
])
```

*He probado muchas veces y no me da errores pero tampoco enseña los resultados:

```
> db.theaters.aggregate([{$match:{"location.address.state":"DC"}},  
  {$group:{_id:"$location.address.zipcode",  
    count:{$sum:1}}},  
  { $sort: { count: -1 } }  
)  
<  
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  
   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  
   genres: [  
     'Comedy'  
   ],  
   rated: 'R',  
   metacritic: 82,  
   title: 'Animal House',  
   lastupdated: '2015-09-13 00:02:47.803000000',  
   languages: [  
     'English',  
     'Italian'  
   ],  
   ...}
```

- COMENTARIOS: Importante recordar usar `.` en lugar de `,` para los decimales.

Ejercicio 2

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

The screenshot shows two instances of the Compass MongoDB interface. The top instance displays the 'Schema' tab for the 'theathers' collection, with three document snippets listed:

- `_id: ObjectId('59a47286cfa9a3a73e51e72c')`
`theaterId : 1000`
► `location : Object`
- `_id: ObjectId('59a47286cfa9a3a73e51e72d')`
`theaterId : 1003`
► `location : Object`
- `_id: ObjectId('59a47286cfa9a3a73e51e72e')`
`theaterId : 1008`
► `location : Object`

The bottom instance shows the 'Schema' tab for the 'theathers' collection, with an 'Analyze' button highlighted by a yellow circle. Below it, the 'EXPORT SCHEMA' section displays the following fields and their statistics:

- _id**: objectid
Histogram: S M T W T F S 0:00 6:00 12:00 18:00 23:00
first: 2017-08-28 19:44:04 last: 2017-08-28 19:44:07
- location**: Document with 2 nested fields.
Histogram: S M T W T F S 0:00 6:00 12:00 18:00 23:00
- theaterId**: int32
Histogram: 1095 2505 899 308 1380 1149 2910 246 2864 156 195
264 2884 572 792 8143 1519 489 8177 602



Type a query: { field: 'value' }

Reset

Analyze



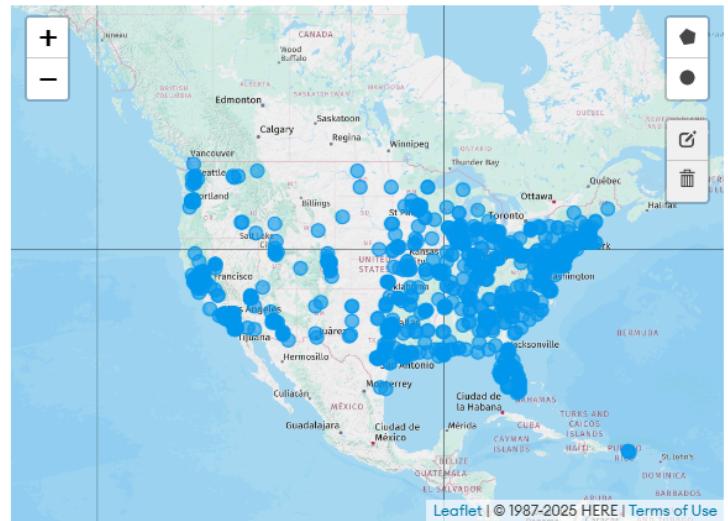
Options ►

EXPORT SCHEMA

This report is based on a sample of 1000 documents. [Learn more](#)

▶ geo

coordinates



Leaflet | © 1987-2025 HERE | [Terms of Use](#)