

## Tarea S9.01. Consultas con MongoDB

Entrega:

Almacena en un repositorio de tu GitHub una carpeta que contenga:

Un PDF que contenga una captura de pantalla de MongoDB Compass donde se pueda observar cada script de las consultas que has hecho y el resultado obtenido para cada ejercicio.

En la entrega, coloca el link a este repositorio.

### Descripción

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

Permite el uso del texto de forma anónima para mejorar el servicio de traducción.

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.

Realizarás algunas consultas que te pide el cliente/a, quien 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.

### - Ejercicio 1

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

Primera manera, en las opciones de la colección poner {"date": 1} en Sort y Limit 2

The screenshot shows the MongoDB Compass interface. At the top, there's a query builder with the following settings:

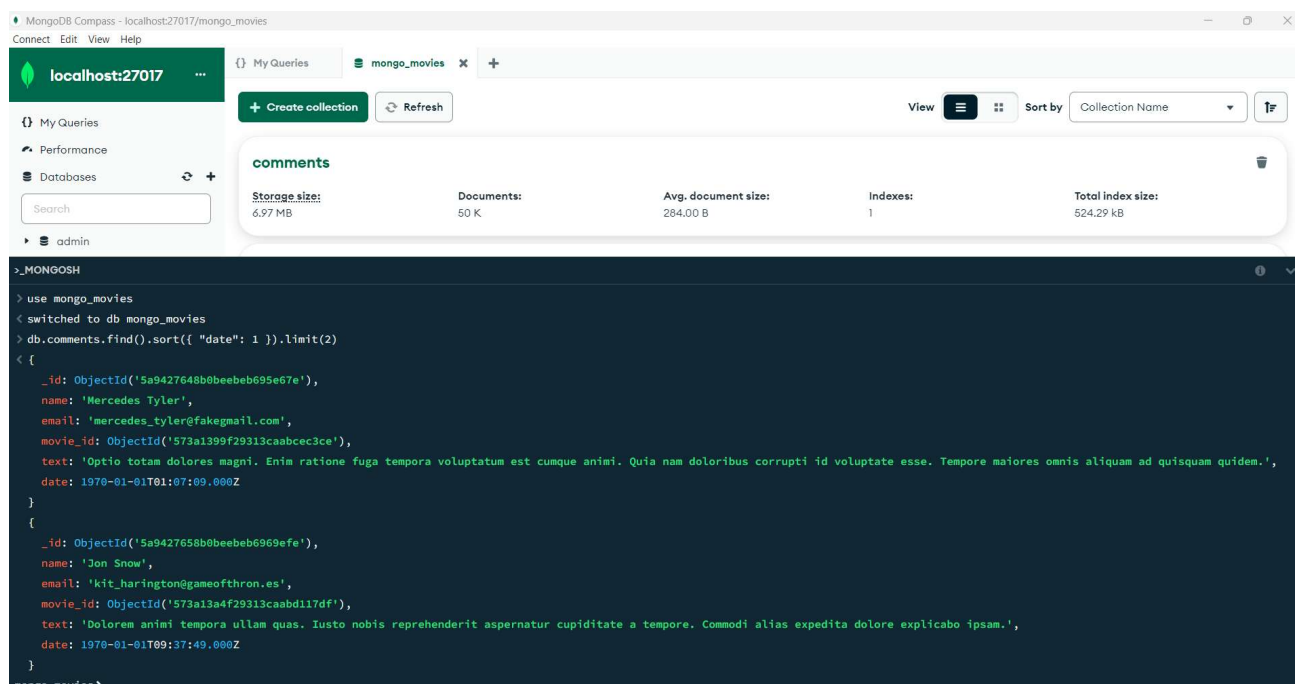
- Project:** { field: 0 }
- Sort:** { "date": 1 }
- Collation:** { locale: 'simple' }
- MaxTimeMS:** 60000
- Skip:** 0
- Limit:** 2

Below the query builder, there are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". The results section shows two documents:

```
{
  "_id": ObjectId('5a9427648b0beebe695e67e'),
  "name": "Mercedes Tyler",
  "email": "mercedes_tyler@fakegmail.com",
  "movie_id": ObjectId('573a1399f29313caabcec3ce'),
  "text": "Optio totam dolores magni. Enim ratione fuga temporam voluptatum est cu...",
  "date": 1978-01-01T01:07:09.000+00:00
}
```

```
{
  "_id": ObjectId('5a9427658b0beebe6969efe'),
  "name": "Jon Snow",
  "email": "kit_harrington@gameofthron.es",
  "movie_id": ObjectId('573a13a4f29313caabd117df'),
  "text": "Dolorem animi tempora ullam quas. Iusto nobis reprehenderit aspernatur...",
  "date": 1978-01-01T09:37:49.000+00:00
}
```

Segunda manera: en mongodb shell poner `db.comments.find().sort({ "date": 1 }).limit(2)` tras seleccionar la base de datos con `use`:

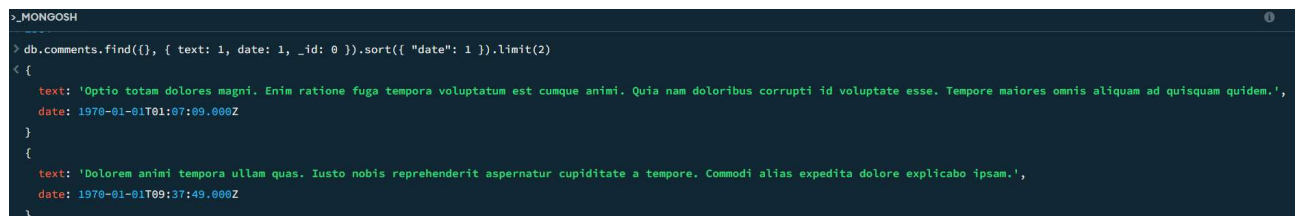


The screenshot shows the MongoDB Compass interface for the `mongo_movies` database. The `comments` collection is selected, showing a storage size of 6.97 MB, 50 K documents, and 1 index. Below this, the MongoDB Shell displays the results of the query `db.comments.find().sort({ "date": 1 }).limit(2)`, showing two documents sorted by date. The first document is from Mercedes Tyler (1970-01-01T01:07:09.000Z) and the second is from Jon Snow (1970-01-01T09:37:49.000Z).

```
> use mongo_movies
switched to db mongo_movies
> db.comments.find().sort({ "date": 1 }).limit(2)
{
  "_id": ObjectId("5a9427648b0beebe695e67e"),
  "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 voluptate esse. Tempore maiores omnis aliquam ad quisquam quidem.",
  "date": 1970-01-01T01:07:09.000Z
}
{
  "_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 cupiditate a tempore. Commodi alias expedita dolore explicabo ipsam.",
  "date": 1970-01-01T09:37:49.000Z
}
```

Para ver solo los campos de texto y la fecha (que evidencia la antigüedad de los comentarios) se modifica la consulta de la siguiente manera:

`db.comments.find({}, { text: 1, date: 1, _id: 0 }).sort({ "date": 1 }).limit(2)`



The screenshot shows the MongoDB Shell displaying the results of the query `db.comments.find({}, { text: 1, date: 1, _id: 0 }).sort({ "date": 1 }).limit(2)`. The output shows two documents with only the `text` and `date` fields displayed, sorted by date.

```
> db.comments.find({}, { text: 1, date: 1, _id: 0 }).sort({ "date": 1 }).limit(2)
{
  "text": "Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cumque animi. Quia nam doloribus corrupti id voluptate esse. Tempore maiores omnis aliquam ad quisquam quidem.",
  "date": 1970-01-01T01:07:09.000Z
}
{
  "text": "Dolorem animi tempora ullam quas. Iusto nobis reprehenderit aspernatur cupiditate a tempore. Commodi alias expedita dolore explicabo ipsam.",
  "date": 1970-01-01T09:37:49.000Z
}
```

¿Cuántos usuarios tenemos registrados?

El número de usuarios aparece en dos sitios al abrir la colección `users` y es 185:

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'config', 'local', 'mongo\_movies', and 'users'. The 'users' database is selected, and the 'users' collection is highlighted. The main panel shows the 'Documents' tab with a count of 185. A query bar at the top contains the text 'Type a query: { field: 'value' } or Generate query'. Below the query bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The document list shows three sample documents with fields like '\_id', 'name', 'email', and 'password'. The document count '185' is highlighted with a red box.

Otra manera de obtenerlo es con la consulta siguiente:

`db.users.countDocuments()`

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

¿Cuántos cines existen en el estado de California?

De la misma manera que en el ejercicio anterior, se obtiene el número 1564:

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'config', 'local', 'mongo\_movies', and 'theaters'. The 'theaters' database is selected, and the 'theaters' collection is highlighted. The main panel shows the 'Documents' tab with a count of 1564. A query bar at the top contains the text 'Type a query: { field: 'value' } or Generate query'. Below the query bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The document list shows three sample documents with fields like '\_id', 'theaterId', and 'location'. The document count '1564' is highlighted with a red box.

o bien:

```
>_MONGOSH
> db.theaters.countDocuments()
< 1564
```

¿Cuál fue el primer usuario en registrarse?

No hay campo de timestamp pero se puede deducir que el primer usuario es el primero listado en la colección que está ordenada por defecto por orden alfabético del campo `_id`:

ADD DATA EXPORT DATA UPDATE DELETE

```
{
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu"
```

Si, es necesario se puede especificar el orden (Sort) y el número de registros = 1 (limit) en las opciones:

Documents 185 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find </> Options

Project { field: 0 }

Sort { \_id: 1 } MaxTimeMS 60000

Collation { locale: 'simple' } Skip 0 Limit 1

ADD DATA EXPORT DATA UPDATE DELETE 1 - 1 of 1

```
{
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu"
```

La consulta (con y sin aislar el campo del nombre) sería:

```
> _MONGOSH

> db.users.find().sort({ "_id": 1 }).limit(1)
< {
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": 'Ned Stark',
  "email": 'sean_bean@gameofthron.es',
  "password": '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu'
}

> db.users.find({}, { name: 1, _id: 0 }).sort({ "_id": 1 }).limit(1)
< {
  "name": 'Ned Stark'
}
```

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

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

## - Ejercicio 2

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

Documents 23.5K

Aggregations

Schema

Indexes 1

Validation

🔍

{ year: 1932, \$or: [{ genres: "Drama" }, { languages: "French" }] }

💡 Generate query ⚡

Explain

Reset

➕ ADD DATA

📄 EXPORT DATA

✎ UPDATE

🗑 DELETE

1 - 18 of 18 ↻

```
_id: ObjectId('573a1391f29313caabcd9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mou..."
runtime: 55
rated: "UNRATED"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWYjLmYi00NjA4LTk4ZW..."
title: "The Blood of a Poet"
lastupdated: "2015-09-16 13:13:05.537000000"
languages: Array (1)
released: 2010-05-20T00:00:00.000+00:00
directors: Array (1)
writers: Array (1)
awards: Object
year: 1932
imdb: Object
countries: Array (1)
type: "movie"
tomatoes: Object
```

```
_id: ObjectId('573a1392f29313caabcd99a3')
plot: "Junta is hated by the people in the villages where she lives. especiall "
```

La consulta mostrando solo los títulos sería:

```
db.movies.find({ year: 1932, $or: [{ genres: "Drama" }, { languages: "French" }] }, { title: 1, _id: 0 })
```

Para mostrar cada película en una sola fila, se añade `.forEach(function(movie){print(movie.title);})`

```
db.movies.find({year: 1932, $or: [{ genres: "Drama" }, {languages: "French"}] },
{ title: 1, _id: 0 } ).forEach(function(movie) {print(movie.title);});
```

```

>_MONGOOSH

> db.movies.find({year: 1932, $or: [{ genres: "Drama" },{languages: "French"}] },{ title: 1, _id: 0 }).forEach(function(movie) {print(movie.title)});
< The Blood of a Poet
< The Blue Light
< Broken Lullaby
< The Crowd Roars
< A Farewell to Arms
< Forbidden
< Freaks
< Grand Hotel
< I Am a Fugitive from a Chain Gang
< The Mummy
< Payment Deferred
< The Red Head
< Red Dust
< Scarface
< Shanghai Express
< Smilin' Through
< Two Seconds
< I Was Born, But...
mongo_movies>

```

### - Ejercicio 3

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

Primera manera:

mongo\_movies > movies

Documents 23.5K Aggregations Schema Indexes 1 Validation

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

Generate query Explain Reset

ADD DATA EXPORT DATA UPDATE DELETE 1 - 20 of 166

```

  _id: ObjectId('573a13acf29313caabd29366')
  fullplot: "The manager of the negative assets sector of Life magazine, Walter Mit..."
  ▶ imdb: Object
  year: 2013
  plot: "When his job along with that of his co-worker are threatened, Walter t..."
  ▶ 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/MV5BODYwNDYxNDk1N15BMl5BanBnXkFtZT..."
  num_mflix_comments: 1
  released: 2013-12-25T00:00:00.000+00:00
  ▶ awards: Object
  ▶ countries: Array (2)
  ▶ cast: Array (4)
  ▶ directors: Array (1)
  runtime: 114

```

Segunda manera:

```
> db.movies.find({
  countries: "USA",
  "awards.wins": { $gte: 5, $lte: 9 },
  year: { $gte: 2012, $lte: 2014 }
}, { title: 1, _id: 0 }).forEach(function(movie) {
  print(movie.title);
});
< The Secret Life of Walter Mitty
< The Croods
< The Book Thief
< World War Z
< Godzilla
< Mr Hockey: The Gordie Howe Story
< Hitchcock
< Lone Survivor
< One Chance
< The Lone Ranger
< The World's End
```

## Nivel 2

### - Ejercicio 1

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

Hay muchos usuarios con emails dentro de este dominio. Los mensajes de todos ellos suman la cifra 22841:



localhost:27017

My Queries Performance Databases Search

admin config local mongo\_movies

comments

movies sessions test theaters users test

mongo\_movies > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

{email: /@gameofthron.es/ }

Project {}

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

Collation { locale: 'simple' }

Generate query Explain Reset

MaxTimeM

Skip 0 Lim

ADD DATA EXPORT DATA UPDATE DELETE

1 - 20 of 22841

```
{ "_id": ObjectId("5a9427648b0beeb69579d5"),  
  "name": "Petyr Baelish",  
  "email": "aidan_gillen@gameofthron.es",  
  "movie_id": ObjectId("573a1390f29313caabcd4218"),  
  "text": "Quo deserunt ipsam ipsum. Tenetur eos nemo nam sint praesentium minus ...",  
  "date": 2001-07-13T19:25:09.000+00:00 }
```

```
{ "_id": ObjectId("5a9427648b0beeb69579db"),  
  "name": "Olly",  
  "email": "brenock.o.connor@gameofthron.es",  
  "movie_id": ObjectId("573a1390f29313caabcd413b"),  
  "text": "Perspiciatis sit pariatur quas. Perferendis officia harum ipsum delenit...",  
  "date": 2005-01-04T13:49:05.000+00:00 }
```

```
>_MONGOOSH  
> db.comments.find({email: /@gameofthron.es/ }).count()  
< 22841
```

## - Ejercicio 2

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

Respuesta: Sólo 3:

localhost:27017

My Queries Performance Databases Search

admin config local mongo\_movies

comments movies sessions test theaters users test

mongo\_movies > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

{ "location.address.state": 'DC' }

Generate query Explain Reset

ADD DATA EXPORT DATA UPDATE DELETE

1 - 3 of 3

```
{ "_id": ObjectId("59a47286cfa9a3a73e51e785"),  
  "theaterId": 1092,  
  "location": {  
    "address": {  
      "street1": "3100 14th St Nw",  
      "city": "Washington",  
      "state": "DC",  
      "zipcode": "20010"  
    },  
    "geo": {}  
  }  
}
```

```
{ "_id": ObjectId("59a47287cfa9a3a73e51ec33"),  
  "theaterId": 801,  
  "location": {}  
}
```

```
{ "_id": ObjectId("59a47287cfa9a3a73e51ecd8"),  
  "theaterId": 8500,  
  "location": {}  
}
```

```
>_MONGOOSH  
> db.theaters.find({ "location.address.state": 'DC' }).count()  
< 3
```

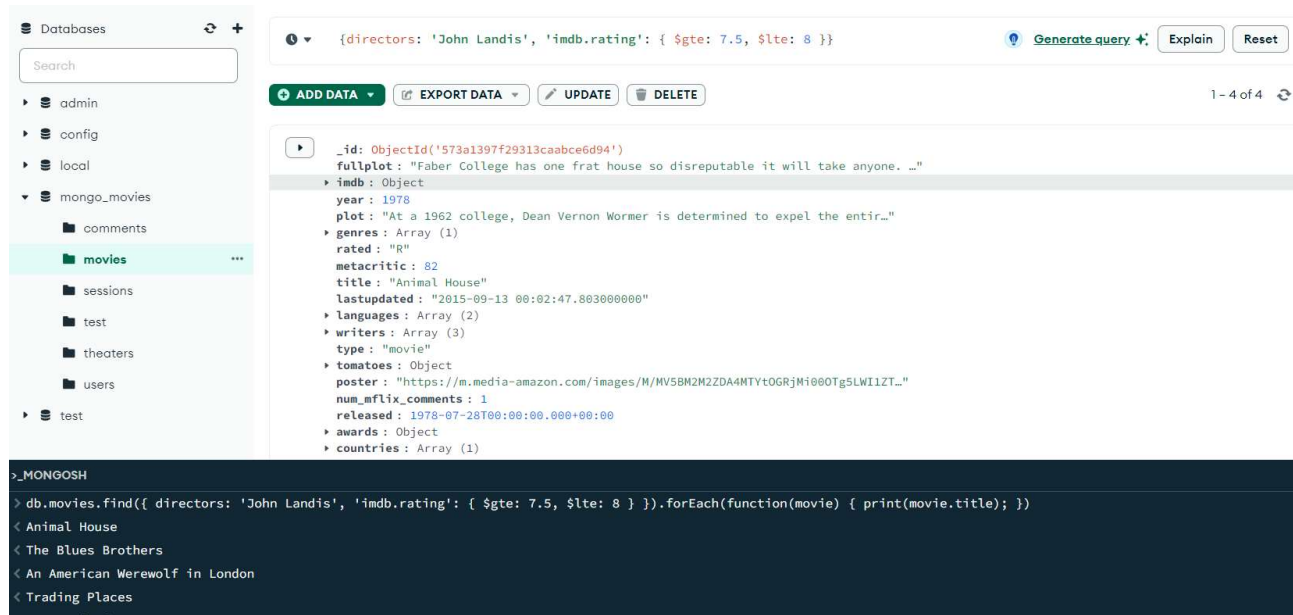


### Nivel 3

#### - Ejercicio 1

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

Hay 4 películas que responden a estas condiciones:

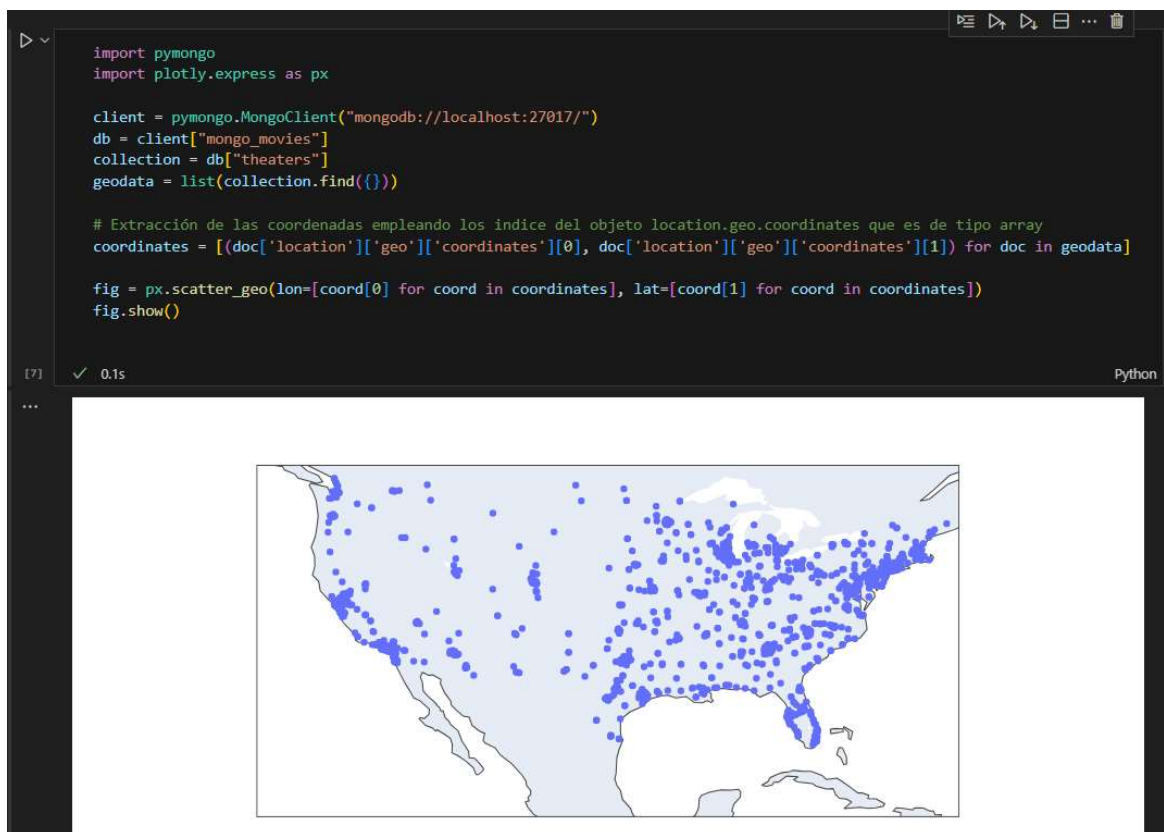


The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible, with 'mongo\_movies' selected. The main panel displays a query: `{directors: 'John Landis', 'imdb.rating': { $gte: 7.5, $lte: 8 }}`. Below the query, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The results section shows a single document for the movie 'Animal House' with details like year (1978), plot, genres, and ratings. At the bottom, a terminal window shows the MongoDB shell command: `db.movies.find({ directors: 'John Landis', 'imdb.rating': { $gte: 7.5, $lte: 8 }}).forEach(function(movie) { print(movie.title); })`, which outputs the titles of the four movies: 'Animal House', 'The Blues Brothers', 'An American Werewolf in London', and 'Trading Places'.

#### - Ejercicio 2

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

Esto se puede lograr de muchas maneras. El conector de mongoDB para Power BI requiere acceso al servicio en la nube de mongoDB llamado Atlas. Una manera más simple es conectarse a mongoDB mediante pymongo y visualizar los datos empleando una librería como plotly.express:



Si es necesario se puede modificar el código para poder mostrar las direcciones de los cines como etiquetas popup.