

ENTREGABLE: sprint 05

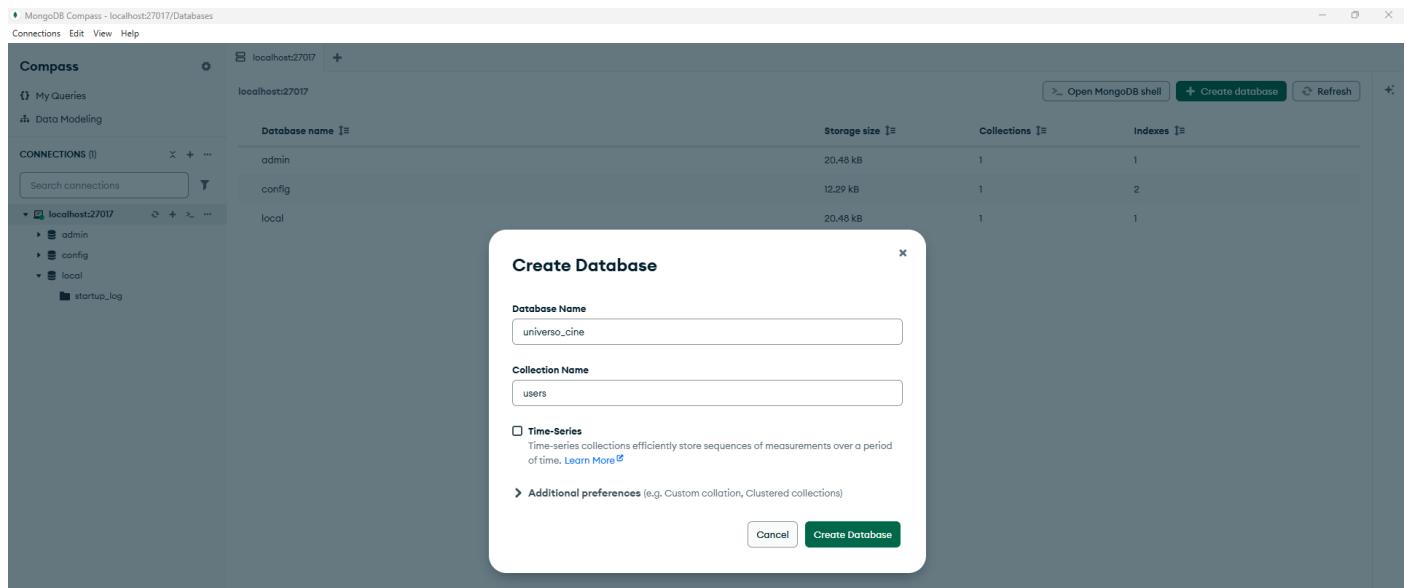
ALUMNO: Ignacio Soldevilla Royo

FECHA: 10-11-2025

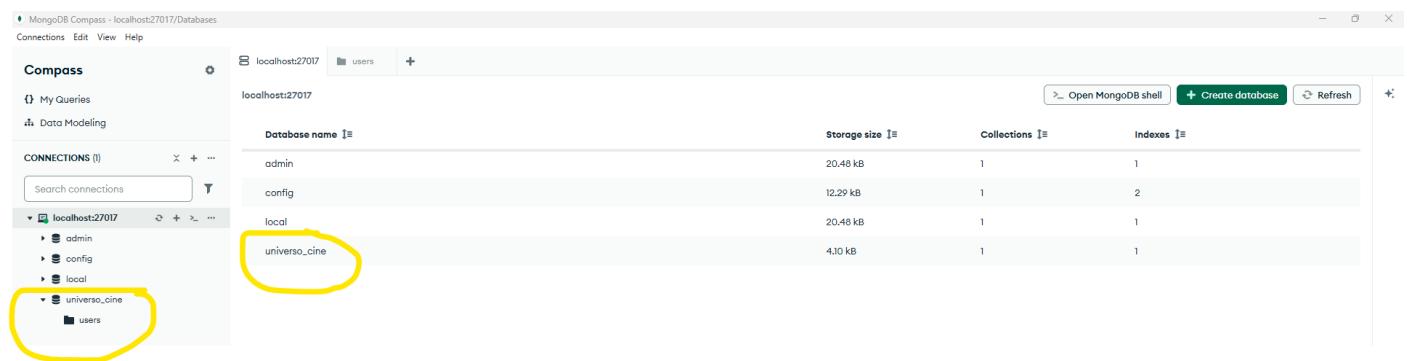
NIVEL 1

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

1. La base de datos la voy a llamar “universo_cine” y la voy a crear desde compass junto con la primera colección “users”, ya que siempre al crear una BD hay que crearla con 1 colección



2. Compruebo que se ha creado tanto la base de datos como la colección:



3. Creo el resto de colecciones: “theatres”, “sesiones”, “movies” y “comments”, entrando en la BD y con el botón de “+ Create collection”. Hay que repetir esta acción para cada nueva colección:

MongoDB Compass - localhost:27017/universo_cine

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS ()

Search connections

localhost:27017 admin config local universo_cine users

localhost:27017 > universo_cine

Collection name: users

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

> Open MongoDB shell + Create collection Refresh

Create Collection

Collection Name: theatres

Time-Series: Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Additional preferences (e.g. Custom collation, Clustered collections)

Cancel Create Collection

This screenshot shows the MongoDB Compass interface. In the main pane, there is a table with one row for the 'users' collection. The 'Collection name' column shows 'users', 'Properties' is '-', 'Storage size' is '4.10 kB', 'Documents' is '0', 'Avg. document size' is '0 B', 'Indexes' is '1', and 'Total index size' is '4.10 kB'. At the top right of the main pane, there is a green button labeled '+ Create collection' which is circled in red. A modal window titled 'Create Collection' is open in the center. It has a text input field for 'Collection Name' containing 'theatres'. Below it is a checkbox for 'Time-Series' which is unchecked. There is also a section for 'Additional preferences' with a link to 'Learn More'. At the bottom of the modal are two buttons: 'Cancel' and 'Create Collection'.

MongoDB Compass - localhost:27017/universo_cine

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS ()

Search connections

localhost:27017 admin config local universo_cine theatres users

localhost:27017 > universo_cine

Collection name: theatres

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

Collection name: users

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

> Open MongoDB shell + Create collection Refresh

Create Collection

Collection Name: sesiones

Time-Series: Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Additional preferences (e.g. Custom collation, Clustered collections)

Cancel Create Collection

This screenshot shows the MongoDB Compass interface after creating the 'theatres' collection. Now, there are two rows in the main table: 'theatres' and 'users'. Both have '-' in the 'Properties' column and '4.10 kB' in the 'Storage size' column. The 'Create Collection' dialog is still open, showing 'sesiones' as the collection name. The 'Time-Series' checkbox is unchecked. At the bottom are 'Cancel' and 'Create Collection' buttons.

MongoDB Compass - localhost:27017/universo_cine

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS ()

Search connections

localhost:27017 admin config local universo_cine sesiones theatres users

localhost:27017 > universo_cine

Collection name: sesiones

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

Collection name: theatres

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

Collection name: users

Properties: -

Storage size: 4.10 kB

Documents: 0

Avg. document size: 0 B

Indexes: 1

Total index size: 4.10 kB

> Open MongoDB shell + Create collection Refresh

Create Collection

Collection Name: movies

Time-Series: Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Additional preferences (e.g. Custom collation, Clustered collections)

Cancel Create Collection

This screenshot shows the MongoDB Compass interface after creating the 'sesiones' collection. Now, there are three rows in the main table: 'sesiones', 'theatres', and 'users'. The 'Create Collection' dialog is still open, showing 'movies' as the collection name. The 'Time-Series' checkbox is unchecked. At the bottom are 'Cancel' and 'Create Collection' buttons.

MongoDB Compass - localhost:27017/universo_cine

localhost:27017 > universo_cine

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
movies	-	4.10 kB	0	0 B	1	4.10 kB
sesiones	-	4.10 kB	0	0 B	1	4.10 kB
theatres	-	4.10 kB	0	0 B	1	4.10 kB
users	-	4.10 kB	0	0 B	1	4.10 kB

Create Collection

Collection Name: comments

Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

[Additional preferences](#) (e.g. Custom collation, Clustered collections)

[Cancel](#) [Create Collection](#)

4. Compruebo que ya tengo las 5 colecciones creadas en la base de datos o contenedor llamado "universo_cine":

MongoDB Compass - localhost:27017/universo_cine

localhost:27017 > universo_cine

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
comments	-	4.10 kB	0	0 B	1	4.10 kB
movies	-	4.10 kB	0	0 B	1	4.10 kB
sesiones	-	4.10 kB	0	0 B	1	4.10 kB
theatres	-	4.10 kB	0	0 B	1	4.10 kB
users	-	4.10 kB	0	0 B	1	4.10 kB

Compass

My Queries Data Modeling

Connections (localhost:27017) Search connections

- admin
- config
- local
- universo_cine
 - comments
 - movies
 - sesiones
 - theatres
 - users

Comments

Comments

Comments

Comments

Comments

5. Una vez creadas las colecciones, hay que importar los documentos de los ficheros json que me da el ejercicio. Para ello empiezo por la colección "users". Entro en la colección y en el botón "+ ADD DATA" cojo la opción "Import JSON or CSV file":

MongoDB Compass - localhost:27017/universo_cine.users

localhost:27017 > universo_cine > users

Documents	Aggregations	Schema	Indexes	Validation
0			1	

Type a query: { field: 'value' } or [Generate query](#)

[EXPLAIN](#) [RESET](#) [FIND](#) [OPTIONS](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

[Import JSON or CSV file](#) [Insert document](#)

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

[Import data](#)

Compass

My Queries Data Modeling

Connections (localhost:27017) Search connections

- admin
- config
- local
- universo_cine
 - comments
 - movies
 - sesiones
 - theatres
 - users

Voy a marcar la opción "Stop on errors" para tener 100% claro si hay errores o no al importar los documentos:

Y una vez pulsado el botón “Import” veo que se han importado todos los datos correctamente porque no hay ningún mensaje de error y por el mensaje en pantalla y también porque son 185 documentos que coincide con el nº de líneas del fichero users.json que se ve si se edita con un text.edit o con Visual Studio Code:

Import completed.
185 documents imported.

6. Repito los pasos del punto 5 anterior ahora para la colección “theatres”:

The screenshot shows the MongoDB Compass interface. The left sidebar displays connections and databases, with 'localhost:27017' and 'universo_cine' selected. The main area shows the 'theatres' collection under 'localhost:27017 > universo_cine > theatres'. A modal window titled 'Import' is open, showing the path 'To collection universo_cine.theatres' and the file 'Import file: theatres.json'. Below the file path, there are 'Options' and a checked 'Stop on errors' checkbox. At the bottom of the modal are 'Cancel' and 'Import' buttons, with 'Import' being highlighted.

Y una vez pulsado el botón “Import” veo que se han importado todos los datos correctamente porque no hay ningún mensaje de error y por el mensaje en pantalla y también porque son 1564 documentos que coincide con el nº de líneas del fichero theatres.json que se ve si se edita con un text.edit o con Visual Studio Code:

The screenshot shows the MongoDB Compass interface with the 'theatres' collection selected. The main area lists 1,564 documents. A yellow arrow points from the bottom left towards the list. A message at the bottom left of the interface area says 'Import completed. 1564 documents imported.'

```

_id: ObjectId("59a47286cfa9a3a73e51e72c")
theaterId: 1000
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e72d")
theaterId: 1003
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e72e")
theaterId: 1008
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e72f")
theaterId: 1004
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e730")
theaterId: 1002
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e731")
theaterId: 1010
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e732")
theaterId: 1014
location: Object

_id: ObjectId("59a47286cfa9a3a73e51e733")
theaterId: 1012
location: Object

```

7. Repito los pasos del punto 5 anterior ahora para la colección “sesiones”:

The screenshot shows the MongoDB Compass interface. On the left, the connection tree shows 'localhost:27017' and the 'universo_cine' database with its collections: comments, movies, sesiones, theatres, and users. The 'sesiones' collection is selected. In the main area, a search bar and buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE are visible. A modal window titled 'Import' is open, showing the path 'localhost:27017 > universo_cine > sesiones'. The 'Import file' field contains 'sessions.json' and has a yellow circle around it. Below it are 'Options' and a checked 'Stop on errors' checkbox. At the bottom of the modal are 'Cancel' and 'Import' buttons.

Y una vez pulsado el botón “Import” veo que se han importado todos los datos correctamente porque no hay ningún mensaje de error y por el mensaje en pantalla y también porque son 1 documentos que coincide con el nº de líneas del fichero sesiones.json que se ve si se edita con un text.edit o con Visual Studio Code:

The screenshot shows the MongoDB Compass interface after the import. The connection tree and collection selection are the same as the previous screenshot. The main pane now displays the imported document: '_id: ObjectId('5a97f9c91c807bb9c6eb5fb4')', 'user_id: "t3qulfem@kwi5.fur"', and 'jwt: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MTk5MDkzMjEsIm5iZlGM..."'. A yellow arrow points from the previous 'Import' dialog to this document view. A notification at the bottom left of the main pane says 'Import completed. 1 document imported.'

8. Repito los pasos del punto 5 anterior ahora para la colección “movies”:

The screenshot shows the MongoDB Compass interface. In the center, a modal window titled 'Import' is open. Inside, it says 'Import collection universo_cine.movies' and 'Import file: movies.json'. At the bottom right of the modal is a green 'Import' button. The entire 'Import' dialog is circled in yellow.

Y una vez pulsado el botón “Import” veo que se han importado todos los datos correctamente porque no hay ningún mensaje de error y por el mensaje en pantalla y también porque son 23.539 documentos que coincide con el nº de líneas del fichero movies.json que se ve si se edita con un text.edit o con Visual Studio Code:

The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The main area displays a large amount of JSON data for a movie document, including fields like _id, plot, genres, runtime, cast, num_afix_comments, title, fullplot, countries, released, directors, rated, awards, lastupdated, year, imdb, type, and tomatoes. A yellow arrow points downwards from the top of the document list to a dark blue banner at the bottom that reads 'Import completed. 23539 documents imported.' with a checkmark icon.

9. Repito los pasos del punto 5 anterior ahora para la colección “comments”:

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar is open, showing a connection to 'localhost:27017' with several databases listed: 'admin', 'config', 'local', and 'universo_cine'. Under 'universo_cine', there are collections: 'comments', 'movies', 'sesiones', 'theatres', and 'users'. In the main area, the 'localhost:27017 > universo_cine > comments' collection is selected. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. To the right, there are buttons for 'Explain', 'Reset', 'Find', 'Options', and a dropdown menu. A modal window titled 'Import' is open, showing the path 'To collection universo_cine.comments'. It has a field 'Import file:' with the value 'comments.json' highlighted by a yellow circle. There are also 'Options' and 'Stop on errors' checkboxes. At the bottom of the modal are 'Cancel' and 'Import' buttons, with 'Import' highlighted by a yellow circle.

Y una vez pulsado el botón “Import” veo que se han importado todos los datos correctamente porque no hay ningún mensaje de error y por el mensaje en pantalla y también porque son 50.304 documentos que coincide con el nº de líneas del fichero comments.json que se ve si se edita con un text.edit o con Visual Studio Code:

The screenshot shows the same MongoDB Compass interface as before, but now the 'Comments' collection is populated with data. The data is shown in a table format with columns: '_id', 'name', 'email', 'movie_id', 'text', and 'date'. There are five rows of data, each representing a comment document. A yellow arrow points from the previous screenshot to this one. At the bottom of the screen, a dark bar displays a green checkmark icon and the text 'Import completed. 50304 documents imported.'

10. Por último, visualizo el resumen de la base de datos con sus colecciones y sus informaciones:

The screenshot shows the MongoDB Compass interface with the 'localhost:27017 > universo_cine' collection selected. On the left, the 'Connections' sidebar is open, showing the same database structure as before. In the main area, a table provides a summary of the database's collections. The columns are: 'Collection name', 'Properties', 'Storage size', 'Documents', 'Avg. document size', 'Indexes', and 'Total index size'. The data is as follows:

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
comments	-	4.10 kB	50K	284.00 B	1	4.10 kB
movies	-	21.55 MB	24K	1.60 kB	1	368.64 kB
sesiones	-	4.10 kB	1	540.00 B	1	4.10 kB
theatres	-	4.10 kB	1.6K	223.00 B	1	4.10 kB
users	-	4.10 kB	185	159.00 B	1	4.10 kB

At the top right of the main area, there are buttons for 'Open MongoDB shell', 'Create collection', and 'Refresh'. A yellow circle highlights the 'Create collection' button.

Ejercicio 1

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

Entro en la colección “comments”, amplio los campos para hacer el filtro pulsando “Options” y pongo:

- Sort {"date": 1} para que me ordene los documentos por fecha ascendente
- Limite a 2 para que me visualice solo los 2 primeros comentarios y le doy a Find

MongoDB Compass - localhost:27017/universo_cine.comments

Connections Edit View Collection Help

Compass

My Queries

Data Modeling

CONNECTIONS ()

localhost:27017

- admin
- config
- local
- universo_cine
 - comments
 - movies
 - sesiones
 - theatres
 - users

localhost:27017 > universo_cine > comments

Documents 50K Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) +:

Project { field: 0 }

Sort {"date": 1}

Collection { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 40000

Find Options ▾

Skip 0 Limit 2

ADD DATA EXPORT DATA UPDATE DELETE

1–2 of 2

`_id: ObjectId('5a9427648b0beeb695e67e')`
`name: "Mercedes_Tyler"`
`email: "Mercedes_tyler@fakegmail.com"`
`movie_id: ObjectId('573a139f29313cabec3ce')`
`text: "Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cu."`
`date: 1970-01-01T01:07:09.000+00:00`

`_id: ObjectId('5a9427658b0beeb6969efc')`
`name: "Don_Snow"`
`email: "kit_harington@gameofthron.es"`
`movie_id: ObjectId('573a13a4f29313cabd117df')`
`text: "Dolorum animi tempora ullam quas. Iusto nobis reprehenderit aspernatur.."`
`date: 1970-01-01T09:37:49.000+00:00`

1.2 ¿Cuántos usuarios tenemos registrados?

Entro en la colección “users” y filtro por la query {} que me visualiza todos los documentos (hay unos por cada usuario) y arriba a la derecha, al lado de la visualización de documentos por página, puedo ver el número total de documentos que son 185

MongoDB Compass - localhost:27017/universo_cine.users

Connections Edit View Collection Help

Compass

My Queries

Data Modeling

CONNECTIONS ()

localhost:27017

- admin
- config
- local
- universo_cine
 - comments
 - movies
 - sesiones
 - theatres
 - users

localhost:27017 > universo_cine > users

Documents 185 Aggregations Schema Indexes 1 Validation

{}

Generate query + Explain Reset Find Options ▾

26 1–25 of 185

ADD DATA EXPORT DATA UPDATE DELETE

`_id: ObjectId('59b99db4cf9a9a34cd7885b6')`
`name: "Ned_Stark"`
`email: "sean_beans@gameofthron.es"`
`password: "52b5125UREwsRUoyF8CQGNK0LzO0HM/jLhgUCNNI39RJAqMUQ74crJ1Vu"`

`_id: ObjectId('59b99db4cf9a9a34cd7885b7')`
`name: "Robert_Baratheon"`
`email: "mark_andy@gameofthron.es"`
`password: "52b5123yGqxLG9L2pXAx2xV0huPns0Zd.VURVkz7wgOLY3pn00sTu2512032y"`

`_id: ObjectId('59b99db4cf9a9a34cd7885b8')`
`name: "Jaime_Lannister"`
`email: "nikolaj_coster-waldau@gameofthron.es"`

También lo puedo hacer en la pestaña “Aggregations”, con un nuevo stage usando \$count, así:

MongoDB Compass - localhost:27017/universo_cine.users

Connections Edit View Collection Help

Compass

My Queries Data Modeling

CONNECTIONS ()

localhost:27017 admin config local universo_cine comments movies sesiones theatres users

localhost:27017 > universo_cine > users

Documents 185 Aggregations Schema Indexes 1 Validation

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

185 Documents in the collection

Preview of documents

_id: ObjectId('59b99db4cfa9a34cd7885b6') name: "Ned Stark" email: "sean.bean@gameofthron.es" password: "\$2b\$12\$UREFwsRUoyF0CRqGNK0Lz0H..."

_id: ObjectId('59b99db4cfa9a34cd7885b7') name: "Robert Baratheon" email: "mark.addy@gameofthron.es" password: "\$2b\$12\$GqxLG9LZpXA2xVDhuPnSOZd..."

_id: ObjectId('59b99db5cf9a34cd7885b8') name: "Jaime Lannister" email: "nikolaj.coster-waldau@gameofthron.es" password: "\$2b\$12\$6vz7wiw0.EISRilvqlzUc."

Stage 1 \$count

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

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

total_usuarios : 185

+ Add stage Learn more about creating pipeline stages

1.3 ¿Cuántos cines existen en el estado de California?

Entro en la colección “theatres” y construyo el filtro usando la notación de puntos para ir entrando en cada documento hasta llegar al campo city, que buscare los documentos en que la city coincida con “California”. Arriba a la derecha, al lado de la visualización de documentos por página, puedo ver el número total de documentos que son 1

MongoDB Compass - localhost:27017/universo_cine.theatres

Connections Edit View Collection Help

Compass

My Queries Data Modeling

CONNECTIONS ()

localhost:27017 admin config local universo_cine comments movies sesiones theatres users

localhost:27017 > universo_cine > theatres

Documents 1.6K Aggregations Schema Indexes 1 Validation

{location.address.city: "California"} Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - of 1

_id: ObjectId('59a47286cfa9a3a73e51e72d') theaterId: 1003 location: Object address: Object street: "45235 Worth Ave." city: "California" state: "MD" zipcode: "20619" geo: Object type: "Point" coordinates: Array (2) 0: -76.512016 1: 38.29697

1.4 ¿Cuál fue el primer usuario en registrarse?

Entro en la colección “users” y veo que en los documentos, no hay un campo de fecha de creación o registro, así que voy a asumir que el campo “_id” se creó automáticamente y ascendente, con lo que, en el filtro, ordenaré ascendentemente todos los documentos por este campo y limitare la visualización a 1 para ver el primero

localhost:27017 > universo_cine > users

Documents 185 Aggregations Schema Indexes 1 Validation

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

**_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name : "Ned Stark"
email : "sean_beansgameofthron.es"
password : "\$2b\$12\$UREFwsRuoyF8CRqGhNK0Lz0HHM/jLhgUCNNIj9RJAqMUQ74crLJ1Vu"**

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

Entro en la colección “movies” y veo que el campo donde esta el género es el campo “genres”, que es del tipo vector o array y que una misma película puede tener varios géneros, con lo cual “Comedy” puede estar en posiciones del vector distintas en documentos diferentes. Como nos dice que este, independientemente de la posición, el filtro será simple, así { “genres”: “Comedy” } y veo, arriba a la derecha, al lado de la visualización de documentos por página, que el número total de documentos o películas que tienen género “Comedy” es de 7.024

localhost:27017 > universo_cine > movies

Documents 24K Aggregations Schema Indexes 1 Validation

{ "genres": "Comedy" }

ADD DATA EXPORT DATA UPDATE DELETE

**_id: ObjectId('573a1390f29313caabcd50e5')
plot : "The cartoonist, Winsor McCay, brings the Dinosaur back to life in th..."
genres : Array (3)
0: "Animation"
1: "Short"
2: "Comedy"
runtime : 12**

**_id: ObjectId('573a1390f29313caabcd60e4')
plot : "Charlie is an immigrant who endures a challenging voyage and gets into..."
genres : Array (3)
0: "Short"
1: "Comedy"
2: "Drama"
runtime : 30**

1 - 25 of 7024

Ejercicio 2

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

Entro en la colección "movies" en documents. Si pongo varias condiciones en el filtro son por defecto AND, pero en este caso tengo que usar también una OR al pedirme que sea Drama o en Frances, así que voy a tener que usar el operador \$or para esta segunda condición y añadirlo a la primera condición. Esto lo logro con el filtro: {"year": 1932, \$or:[{"genres": "Drama"}, {"languages": "French"}]} para obtener que son 18 películas las que cumplen las condiciones del ejercicio. En el pantallazo se muestran 2 de los 18 que se verían con scroll

```

{
  "_id": ObjectId('573a1391f29313caabcd9458'),
  "plot": "A young artist draws a face at a canvas on his easel. Suddenly the mou...",
  "runtime": 55,
  "rated": "UNRATED",
  "cast": [
    ...
  ],
  "num_mflix_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4Z...",
  "title": "The Blood of a Poet",
  "lastupdated": "2015-09-16 13:13:05.537000000",
  "languages": [
    ...
  ],
  "released": "2010-05-26T00:00:00.000+00:00",
  "directors": [
    ...
  ],
  "writers": [
    ...
  ],
  "awards": {
    ...
  },
  "year": 1932,
  "imdb": {
    ...
  },
  "countries": [
    ...
  ],
  "type": "movie",
  "tomatoes": {
    ...
  }
}

{
  "_id": ObjectId('573a1392f29313caabcd99a3'),
  "plot": "Junta is hated by the people in the village where she lives, especially ...",
  "genres": [
    ...
  ],
  "runtime": 85,
  "cast": [
    ...
  ],
  "poster": "https://m.media-amazon.com/images/M/MV5BNTQ1NTMzMTQtODIyYS00MTAxLWE1NT...",
  "title": "The Blue Light",
  "fullplot": "Junta is hated by the people in the village where she lives, especially ...",
  "languages": [
    ...
  ],
  "released": "1934-05-08T00:00:00.000+00:00",
  "directors": [
    ...
  ],
  "writers": [
    ...
  ],
  "awards": {
    ...
  }
}

```

También lo podría hacer creando un pipeline en Aggregations paso a paso de la siguiente manera (separo en 2 etapas distintas el and y el or para claridad, aunque se podría poner en una misma etapa):

localhost:27017 > universo_cine > movies

Open MongoDB shell

Documents (24K) Aggregations Schema Indexes (1) Validation

\$match \$match \$count

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW {} STAGES TEXT WIZARD Run Options

23539 Documents in the collection

Preview of documents

```
_id: ObjectId('573a1390f29313caabcd4135')
plot: "Three men hammer on an anvil and pass a bottle of beer around."
genres: Array (1)
runtime: 1
cast: Array (2)
num_mflix_comments: 1
title: "Blacksmith Scene"
```

```
_id: ObjectId('573a1390f29313caabcd42e8')
plot: "A group of bandits stage a brazen train hold-up, only to find a determined..."
```

```
_id: ObjectId('573a1390f29313caabcd4323')
plot: "A young boy, oppressed by his mother, goes on an outing in the country..."
```

Stage 1 \$match

```
1 /**
2 * query: The query in MQL.
3 */
4 {
5   "year": 1932
6 }
```

Output preview after \$match stage (Sample of 10 documents)

```
_id: ObjectId('573a1391f29313caabcd9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mouse..."
```

```
_id: ObjectId('573a1392f29313caabcd99a')
plot: "Junta is hated by the people in the village where she lives, especially..."
```

Stage 2 \$match

```
1 /**
2 * query: The query in MQL.
3 */
4 {
5   $or:[{"genres": "Drama"}, {"languages": "French"}]
6 }
```

Output preview after \$match stage (Sample of 10 documents)

```
_id: ObjectId('573a1391f29313caabcd9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mouse..."
```

```
_id: ObjectId('573a1392f29313caabcd99a')
plot: "Junta is hated by the people in the village where she lives, especially..."
```

Stage 3 \$count

```
1 /**
2 * Provide the field name for the count.
3 */
4 'Numero peliculas resultante'
```

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

Numero peliculas resultante : 18

Add stage

Learn more about aggregation pipeline stages

Ejercicio 3

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

Entro en la colección “movies” en documents. Como hay 2 condiciones que ponen límite mínimo y máximo, voy a utilizar los operadores mayor o igual (\$gte) y menor o igual (\$lte) y voy encadenando las 3 condiciones:

1. {"countries": "USA",}
2. "awards.wins": {\$gte: 5, \$lte:9},
3. "year": {\$gte: 2012, \$lte: 2014}}

Quedando el filtro final: {"countries": "USA", "awards.wins": {\$gte: 5, \$lte:9}, "year": {\$gte: 2012, \$lte: 2014}}

Para obtener un total de 166 documentos. En el pantallazo se ven los 2 primeros documentos, para el resto hay que hacer scroll

The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The top navigation bar includes tabs for 'comments', 'users', 'sesiones', 'theatres', 'movies', and a '+' button. Below the navigation is a search bar with the URL 'localhost:27017 > universo_cine > movies'. To the right of the search bar is a 'Open MongoDB shell' button. The main area is titled 'Documents' with a count of '24K'. Below this are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The document list starts with two highlighted entries:

```
es": "USA", "awards.wins": {$gte: 5, $lte:9}, "year": {$gte: 2012, $lte: 2014]} Generate query + Explain Reset Find Options
```

The first document has a yellow circle around its fields:

```
genres: Array (3)
runtime: 116
metacritic: 63
rated: "PG-13"
cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BNDQ4YzFmNzktMmM5ZC00MDZjLTK1OT...
title: "World War Z"
fullplot: "Life for former United Nations investigator Gerry Lane and his family ...
languages: Array (4)
released: 2013-06-21T00:00:00.000+00:00
directors: Array (1)
writers: Array (6)
awards: Object
lastupdated: "2015-09-10 17:37:02.647000000"
year: 2013
imdb: Object
countries: Array (2)
type: "movie"
tomatoes: Object
```

The second document also has a yellow circle around its fields:

```
_id: ObjectId('573a13b9f29313caab4df21f')
fullplot: "In 1999, the Janjira nuclear plant was mysteriously destroyed with mos...
imdb: Object
year: 2014
plot: "The world is beset by the appearance of monstrous creatures, but one o...
genres: Array (3)
rated: "PG-13"
metacritic: 62
title: "Godzilla"
```

También lo podría hacer creando un pipeline en Aggregations paso a paso de la siguiente manera

localhost:27017 > universo_cine > movies

Open MongoDB shell

Documents 24K Aggregations Schema Indexes 1 Validation

\$match \$match \$match \$count

Generate aggregation Explain Export Run Options

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

23539 Documents in the collection

Preview of documents

`_id: ObjectId('573a1390f29313caabcd4135')
plot: "Three men hammer on an anvil and pass a bottle of beer around."
genres: Array (1)
runtime: 1
cast: Array (2)
num_mflix_comments: 1
title: "Blacksmith Scene"`

`_id: ObjectId('573a1390f29313caabcd42e8')
plot: "A group of bandits stage a brazen train hold-up, only to find a determined..."
genres: Array (2)
runtime: 11
cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BMTU3Nj...`

`_id: ObjectId('573a1390f29313caabcd4323')
plot: "A young boy, oppressed by his mother, goes on an outing in the country."
genres: Array (3)
runtime: 14
rated: "UNRATED"
cast: Array (4)
num_mflix_comments: 2`

Stage 1 \$match

Output preview after \$match stage (Sample of 10 documents)

`_id: ObjectId('573a1390f29313caabcd4135')
plot: "Three men hammer on an anvil and pass a bottle of beer around."
genres: Array (1)
runtime: 1
cast: Array (2)
num_mflix_comments: 1
title: "Blacksmith Scene"
fullplot: "A stationary camera looks at a..."`

`_id: ObjectId('573a1390f29313caabcd42e8')
plot: "A group of bandits stage a brazen train hold-up, only to find a determined..."
genres: Array (2)
runtime: 11
cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BMTU3Nj...`

Stage 2 \$match

Output preview after \$match stage (Sample of 10 documents)

`_id: ObjectId('573a1391f29313caabcd7f27')
plot: "A prospector goes to the Klondike in search of gold and finds it and more."
genres: Array (3)
runtime: 95
rated: "NOT RATED"
title: "The Gold Rush"
num_mflix_comments: 2
poster: "https://m.media-amazon.com/images/M/MV5BMTU3Nj...`

`_id: ObjectId('573a1391f29313caabcd88d8')
plot: "A street cleaner saves a young woman's life, and the pair slowly fall in love."
genres: Array (2)
runtime: 110
rated: "NOT RATED"
title: "7th Heaven"
num_mflix_comments: 1`

Stage 3 \$match

Output preview after \$match stage (Sample of 10 documents)

`_id: ObjectId('573a13acf29313caabd29366')
fullplot: "The manager of the negative assets sector of Life magazine, Walter Miller,..."
imdb: Object
year: 2013
plot: "When his job along with that of his co-worker are threatened, Walter Miller..."
genres: Array (3)`

`_id: ObjectId('573a13b5f29313caabd4577')
plot: "After their cave is destroyed, a caveman family must trek through the..."
genres: Array (3)
runtime: 98
metacritic: 55
rated: "PG"
cast: Array (4)`

The screenshot shows the MongoDB aggregation pipeline interface. On the left, Stage 4 is selected, with the stage name '\$count' and a status indicator showing it is active. The code for the stage is:

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

On the right, the 'Output preview after \$count stage (Sample of 1 document)' is shown, displaying the result: 'Total_documentos : 166'. This result is highlighted with a yellow oval.

At the bottom center is a green button labeled '+ Add stage'. Below it, a link reads 'Learn more about aggregation pipeline stages'.

NIVEL 2

Ejercicio 1

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

Para este ejercicio voy a utilizar la coincidencia por subcadena utilizando el operador \$regex para buscar todos los documentos en los que el valor de incluya el término “gameofthron.es” y como me pide los comentarios de unos de ellos, luego los voy a agrupar por el campo “email” que es único para cada usuario y sumando 1 con el operador \$sum:1 por cada vez que aparecen. Las agrupaciones y conteos por grupo solo se pueden hacer en aggregations.

Primero con el Stage 1 filtro todos los que tienen el mismo dominio "@gameofthron.es" :

localhost:27017 > universo_cine > comments

Open MongoDB shell

Documents 50K 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 ▾

50304 Documents in the collection

Preview of documents

_id: ObjectId('5a9427648b0beebeb69579cc')
name : "Andrea Le"
email : "andrea_le@fakegmail.com"
movie_id : ObjectId('573a1390f29313caabcd41...
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('573a1390f29313caabcd41...
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@gameofthrone.es"
movie_id : ObjectId('573a1390f29313caabcd41b...
text : "Rem itaque ad sit rem voluptatibus.
Ad fugiat maxime illum optio iure ..."
date : 1998-08-22T11:45:03.000+00:00

Stage 1 \$match

```
1 /* * query: The query in MQL. * */ 4 { 5 "email": {$regex: "@gameofthrone.es"} 6 }
```

Output preview after \$match stage (Sample of 10 documents)

_id: ObjectId('5a9427648b0beebeb69579d0')
name : "Talisa Maegyr"
email : "oona_chaplin@gameofthrone.es"
movie_id : ObjectId('573a1390f29313caabcd41b...
text : "Rem itaque ad sit rem voluptatibus.
Ad fugiat maxime illum optio iure ..."
date : 1998-08-22T11:45:03.000+00:00

_id: ObjectId('5a9427648b0beebeb69579d')
name : "Petyr Baelish"
email : "aidan_gillen@gameofthrone.es"
movie_id : ObjectId('573a1390f29313caab...
text : "Quo deserunt ipsam ipsum. Net
nemo nam sint praesentium minus
date : 2001-07-13T19:25:09.000+00:00

Y con el stage 2 los agrupo por el campo “email” sumando 1 cada vez que aparecen. Aquí ya veo por ejemplo que el usuario con email conleth_hill@gameofthron.es ha hecho un total de 271 comentarios:

The screenshot shows the MongoDB aggregation pipeline interface. On the left, the pipeline stages are defined:

```
1  /***
2   * _id: The id of the group.
3   * fieldN: The first field name.
4   */
5  {
6    _id: "$email",
7    "Total_Comentarios": {
8      $sum: 1
9    }
10 }
```

The second stage is a `$group` stage, indicated by the dropdown menu and the blue link in the stage list.

The output preview shows two documents from a sample of 10:

- `_id: "conleth_hill@gameofthron.es"`
- `Total_Comentarios : 271`

Another document is partially visible on the right:

- `_id: "hafþór_július_björnsson@gameoftl`
- `Total_Comentarios : 285`

A yellow oval highlights the first document in the preview, and another yellow oval highlights the second document.

Ejercicio 2

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

Como hay que contar por la segmentación de código postal, lo tengo que hacer por aggregations.

Con el Stage 1 filtro los documentos que tienen "state" igual a "DC" que son 3

The screenshot shows the MongoDB aggregation interface. At the top, there are tabs for 'Documents' (1.6K), 'Aggregations' (selected), 'Schema', 'Indexes' (1), and 'Validation'. Below the tabs, there are buttons for '\$match' and '\$group'. On the right, there are buttons for 'Generate aggregation', 'Explain', 'Export', 'Run', and 'Options'. Underneath these are buttons for 'Untitled - modified', 'SAVE', 'CREATE NEW', and 'EXPORT TO LANGUAGE'. To the right of these are buttons for 'PREVIEW', 'STAGES', 'TEXT', 'WIZARD', and a gear icon. A yellow oval highlights the 'Stage 1 \$match' section. The output preview shows three documents matching the query: one with theaterId 1000, one with 1003, and one with 1008.

Y con el stage 2 los agrupo por su “zipcode” y sumo 1 por cada vez que aparecen:

The screenshot shows the MongoDB aggregation interface with the '\$group' stage expanded. The stage definition is as follows:

```

1  /**
2   * _id: The id of the group.
3   * fieldN: The first field name.
4   */
5  {
6    _id: "$location.address.zipcode",
7    numero_cines: {
8      $sum: 1
9    }
10 }

```

The output preview shows two groups: one for zipcode '20002' with a count of 1, and another for '20010' with a count of 1.

Y si expando el resultado se visualiza mejor que hay 1 cine en cada uno de los 3 códigos postales:

STAGE INPUT
Sample of 3 documents

```

1  /**
2   * _id: The id of the group.
3   * fieldN: The first field name.
4   */
5  {
6    _id: "$location.address.zipcode",
7    numero_cines: {
8      $sum: 1
9    }
10 }

```

STAGE OUTPUT
Sample of 3 documents

```

_id: "20002"
numero_cines : 1

_id: "20010"
numero_cines : 1

_id: "20016"
numero_cines : 1

```

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.

Entro en la colección “movies” en documents. Como hay 2 condiciones que ponen en el rating, límite mínimo (7.5) y máximo (8), voy a utilizar los operadores mayor o igual (\$gte) y menor o igual (\$lte), ya que entiendo que si una película tiene el valor de los límites también hay que contarla (el 7.5 y el 8)

El filtro sería: {"directors": "John Landis", "imdb.rating": {\$gte: 7.5, \$lte:8}}

Y voy a visualizar solo el id, el título y el rating, para eso en el campo de filtrado Project pongo: {"_id": 1, "title": 1, "imdb.rating": 1 }

localhost:27017 > universo.cine > movies

Documents 24K Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options ▾

Max Time MS 60000
Skip 0 Limit 0

EXPORT DATA ▾ 26 ▾ 1 - 4 of 4 > < > ▾

```

① { "directors": "John Landis", "imdb.rating": {$gte: 7.5, $lte:8} }

Project { "_id": 1, "title": 1, "imdb.rating": 1, "directors": 1 }

Sort { field: -1 } or [[{"field": -1}]

Collation { locale: 'simple' }

Index Hint { field: -1 }

```

Document 1:

```

_id: ObjectId('573a1397f29313caabce6d94')
  + Imdb: Object
    rating: 7.6
  title: "Animal House"
  + directors: Array (1)
    0: "John Landis"

```

Document 2:

```

_id: ObjectId('573a1397f29313caabce76f7')
  title: "The Blues Brothers"
  + directors: Array (1)
    0: "John Landis"
  + Imdb: Object
    rating: 7.9

```

Document 3:

```

_id: ObjectId('573a1397f29313caabce7d06')
  + Imdb: Object
    rating: 7.6
  title: "An American Werewolf in London"
  + directors: Array (1)
    0: "John Landis"

```

Document 4:

```

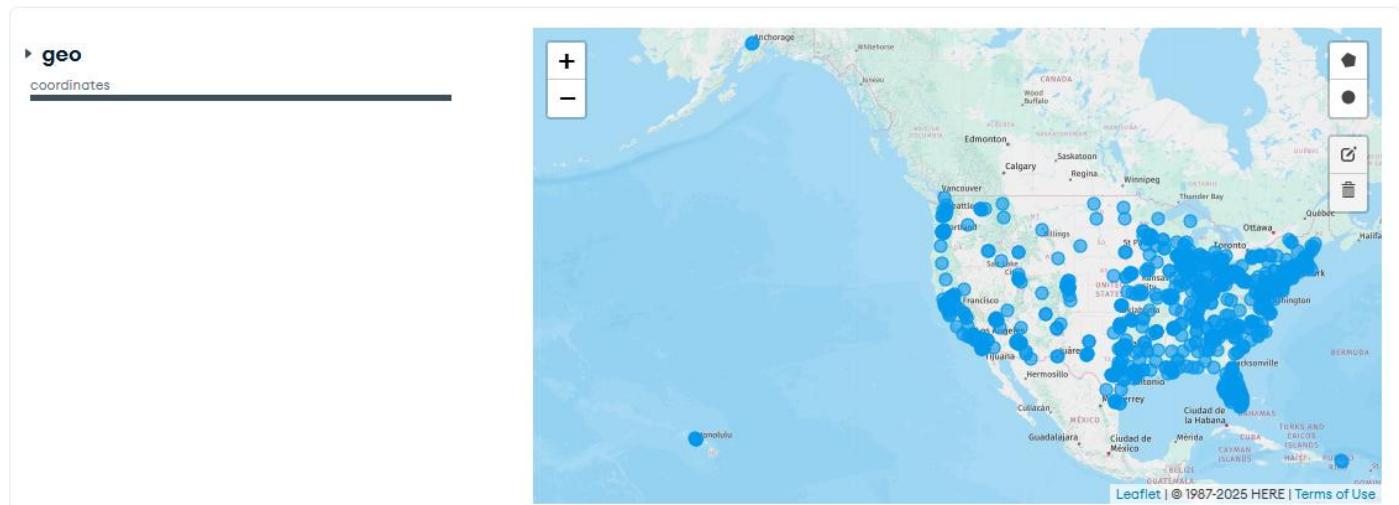
_id: ObjectId('573a1398f29313caabce8deb')
  title: "Trading Places"
  + directors: Array (1)
    0: "John Landis"
  + Imdb: Object
    rating: 7.5

```

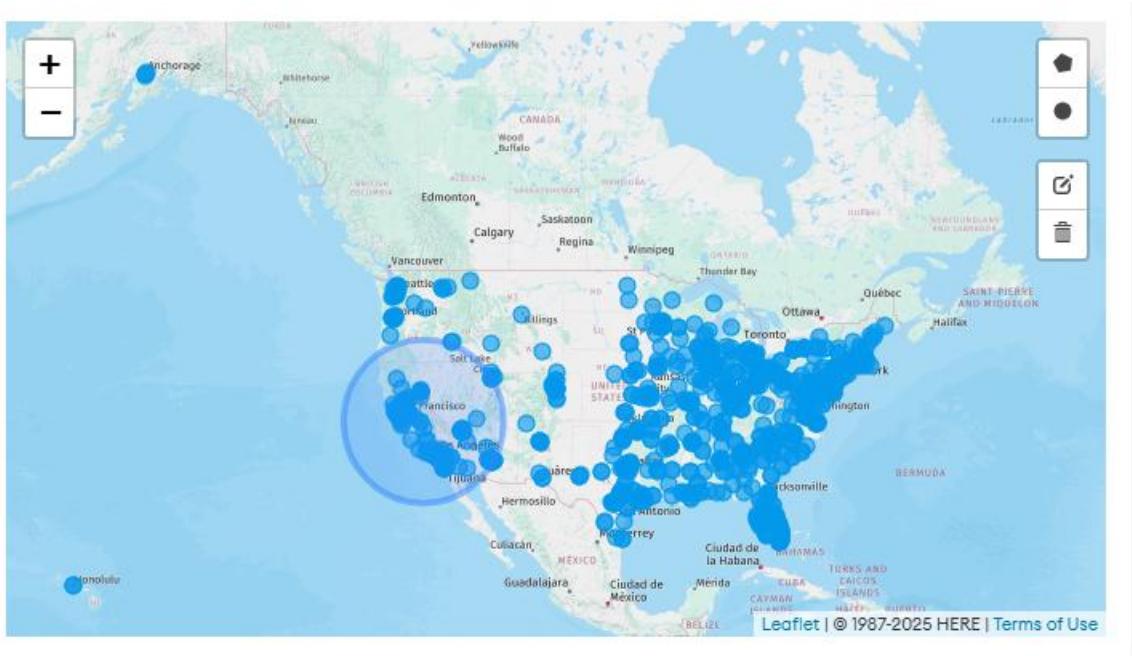
Ejercicio 2

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

Entro en la colección “movies” en la pestaña “Schema” y le doy al botón de analizar. Una vez termina, en los resultados expando el apartado geo y ahí ya tengo el mapa:



Y si quisiera buscar una zona mas concreta dentro del mapa, hay que usar la opción del punto del mapa para determinar la zona que me gustaría expandir haciendo un circulo y volviendo a dar a el botón de “Analyze”, por ejemplo:



Y después de dar al botón Analyze obtengo:

