

## SPRINT 9: CONSULTAS CON MONGODB

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.

```
> db.comments.find().limit(2)
< {
  _id: ObjectId('5a9427648b0beebeb69579cc'),
  name: 'Andrea Le',
  email: 'andrea_le@fakegmail.com',
  movie_id: ObjectId('573a1390f29313caabcd418c'),
  text: 'Rem officis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',
  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. Commodi nisi sit placeat rerum vero cupiditate neque. Dolorum nihil vero animi.',
  date: 1987-02-10T00:29:36.000Z
}
```

¿Cuántos usuarios tenemos registrados?

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

¿Cuántos cines existen en el estado de California?

```
> db.theaters.aggregate([
  {$match: {'location.address.state': 'CA'}},
  {$count: 'total_cines'}
])
< {
  total_cines: 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'
}
```

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

En los documentos de la colección **'movies'** hay un campo llamado *type* que distingue si el documento guardado es de una película o una serie (véase la captura a continuación).

```
db.movies.distinct('type')
[ 'movie', 'series' ]
```

Si no tenemos en cuenta la distinción del tipo, aparecen 7024 documentos como resultado. Sin embargo, si filtramos únicamente por películas, tal y como piden en el ejercicio, aparecen 7002 documentos.

```
> db.movies.aggregate([
  {$match: {'genres': 'Comedy'}},
  {$count: 'total_peliculas_series'}
])
< {
  total_peliculas_series: 7024
}
```

```
> db.movies.aggregate([
  {$match:
    {$and: [{'type': 'movie'}, {'genres': 'Comedy'}]}
  },
  {$count: 'total_peliculas'})
< {
  total_peliculas: 7002
}
```

### **Ejercicio 2.** Muéstrame todos los documentos de las películas producidas en 1932, pero que el género sea drama o estén en francés.

Tengo en cuenta que nos piden únicamente documentos de películas.

Consulta realizada: db.movies.find

```
{
  $and: [{ 'year': 1932 }, { 'type': 'movie' }],
  $or: [{ 'genres': 'Drama' }, { 'languages': 'French' } ]
}
```

En la siguiente captura aparece el extracto del primer documento que cumple con las condiciones. Cabe destacar que ese documento no tiene género definido, pero sí idioma, año y tipo.

```
db.movies.find({
  $and:[{'year':1932}, {'type':'movie'}]],
  $or:[{'genres':'Drama'}, {'languages':'French'}]
})
{
  "_id": ObjectId('573a1391f29313caabed9458'),
  "plot": 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts talking. The artist tries to wipe it away with his hand, but when he',
  "runtime": 55,
  "rated": 'UNRATED',
  "cast": [
    'Enrique Rivero',
    'Elizabeth Lee Miller',
    'Pauline Carton',
    'Odette Talazac'
  ],
  "num_oflix_comments": 1,
  "poster": 'https://s.media-amazon.com/images/H/WV5BYMY300ESZWEtYjlmY108NjA4LTk4ZWYtMzBhZDZ5MjV0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU0._V1_SY1000_SX677_AL_.jpg',
  "title": 'The Blood of a Poet',
  "lastupdated": '2015-09-16 13:13:05.537000000',
  "languages": [
    'French'
  ],
}
```

En resumen, hay 18 documentos que cumplen con las condiciones establecidas.

```
> db.movies.find({
  $and:[{'year':1932}, {'type':'movie'}],
  $or:[{'genres':'Drama'}, {'languages':'French'}]
}).count()
< 18
```

A continuación, muestro otro ejemplo visual de que los filtros aplicados devuelven la información deseada: el documento cumple con el año de producción, tipo y género, pero no con el idioma.

```
_id: ObjectId('573a1392f29313caabcd9d4a')
plot: "A condemned murderer, in the process of being executed, relives the ev..."
▼ genres: Array (2)
  0: "Drama"
  1: "Thriller"
runtime: 67
► cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BZWJmM2JmYjQtODY3YS00M2Q1LTlinj..."
title: "Two Seconds"
fullplot: "Allen claims he his being executed for the wrong murder. Flashbacks sh..."
▼ languages: Array (1)
  0: "English"
released: 1932-05-28T00:00:00.000+00:00
► directors: Array (1)
► writers: Array (2)
► awards: Object
lastupdated: "2015-09-04 00:30:22.147000000"
year: 1932
► imdb: Object
► countries: Array (1)
  type: "movie"
► tomatoes: Object
```

**NOTA:** haciendo comprobaciones extra, he visto que no hay ninguna serie que cumpla con los requisitos deseados, así que el filtro de `{'type': 'movie'}` no sería estrictamente necesario en este caso.

```
> db.movies.find({
  'year':1932,
  $or:[{'genres':'Drama'}, {'languages':'French'}]
}).count()
< 18
```

**Ejercicio 3. Muéstrame todos los documentos de películas estadounidenses que tengan entre 5 y 9 premios que fueron producidas entre 2012 y 2014.**

Tengo en cuenta que los extremos de los rangos de premios y año de producción son inclusivos, y que nos piden únicamente documentos de películas.

Consulta realizada: db.movies.find

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













A continuación, muestro extracto del primer documento del resultado:

```
_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/MV5BODYwNDYxNDk1Nl5BML5BanBnXkFtZT..."
    num_mflix_comments: 1
    released: 2013-12-25T00:00:00.000+00:00
  awards: Object
    wins: 6
    nominations: 13
    text: "6 wins & 13 nominations."
  countries: Array (2)
    0: "USA"
    1: "Canada"
  cast: Array (4)
```

En total obtenemos 162 documentos de películas que cumplen con los criterios solicitados.

```
> db.movies.find({
  'type':'movie',
  'countries':'USA',
  'awards.wins':{$gte:5,$lte:9},
  'year':{$gte:2012,$lte:2014}
}).count()
< 162
```

**NOTA:** en este caso sí que es importante el filtro *'type': 'movie'*, ya que hay 4 series que también cumplen con los filtros aplicados.

🌟 movies						
title String	year Int32	type String	_id ObjectId	plot String	genres Array	
"The Men Who Built Americ...	2012	"series"	1 ObjectId('573a13d7f29313c...	"Miniseries shines a spot...	[ ] 3 elements	  
"The Bible"	2013	"series"	2 ObjectId('573a13d9f29313c...	"The story of God's creat...	[ ] 2 elements	  
"Halo 4: Forward Unto Daw...	2012	"series"	3 ObjectId('573a13d9f29313c...	"UNSC Cadet Thomas Lasky ...	[ ] 2 elements	  
"The Honourable Woman"	2014	"series"	4 ObjectId('573a13e6f29313c...	"Centers on Nessa Stein, ...	[ ] 2 elements	  

## NIVEL 2

**Ejercicio 1.** Cuenta cuántos comentarios escribe un usuario que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico.

En la consulta utilizo \$regex, que es equivalente al LIKE en lenguaje SQL.

- \$regex: /@gameofthron.es\$/i
  - \$: para indicar que email tiene que terminar en '@gameofthron.es'
  - i: para indicar que no haga distinción entre mayúsculas y minúsculas

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

**Ejercicio 2.** ¿Cuántos cines existen en cada código postal situados dentro del estado Washington DC (DC)?

En la consulta realizada utilizo:

- \$match para filtrar por estado DC
- \$group para agrupar por código postal
  - \$count para hacer el recuento de cines por cada código postal

Como resultado, obtenemos tres códigos postales diferentes con un cine en cada uno.

```
> db.theaters.aggregate([
  {$match: {'location.address.state': 'DC'}},
  {$group: {_id: '$location.address.zipcode',
    cantidad_cines: {$count: {}}}}
])
< [
  {
    _id: '20016',
    cantidad_cines: 1
  },
  {
    _id: '20002',
    cantidad_cines: 1
  },
  {
    _id: '20010',
    cantidad_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.

```
Consulta realizada: db.movies.find({  
    'type':'movie',  
    'directors':'John Landis',  
    'imdb.rating':{'$gte':7.5,'$lte':8}  
})
```

Hay un total de 4 películas dirigidas por John Landis que cumplen con el rating imdb.

```
> db.movies.find({  
  'type':'movie',  
  'directors':'John Landis',  
  'imdb.rating':{'$gte':7.5,'$lte':8}  
}).count()  
< 4
```

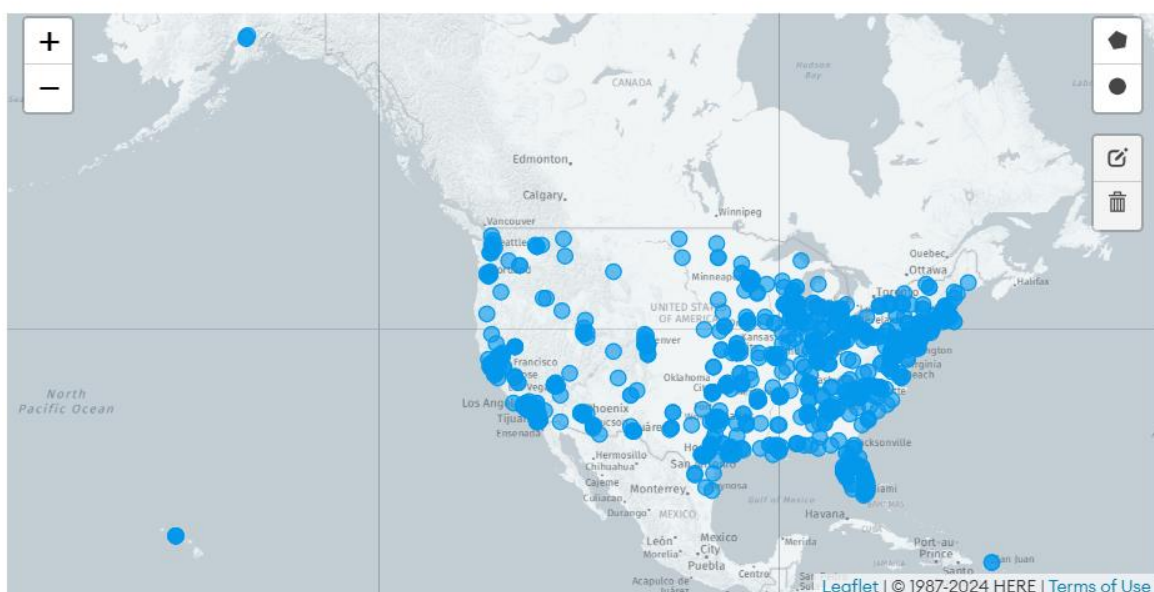
Los detalles de estas películas se muestran a continuación:

	_id ObjectId	fullplot String	year Int32	rating Double	0 String
1	ObjectId('573a1397f29313c...	"Faber College has one fr...	1978	7.6	"John Landis"
2	ObjectId('573a1397f29313c...	"After the release of Jak...	1980	7.9	"John Landis"
3	ObjectId('573a1397f29313c...	"Two American college stu...	1981	7.6	"John Landis"
4	ObjectId('573a1398f29313c...	"Louis Winthorpe is a bus...	1983	7.5	"John Landis"

**Ejercicio 2.** Muestra en un mapa la ubicación de todos los teatros de la base de datos.

Para mostrar las ubicaciones de los teatros en un mapa, he analizado 'schema' de la colección 'theaters'. Al contener coordenadas de las localizaciones, aparece la opción de geo-coordinates.

This report is based on a sample of 1000 documents.



**NOTA:** en la captura del mapa únicamente aparecen 1000 documentos (teatros) representados, en lugar de los casi 1600 teatros registrados.

Esto se debe a que en MongoDBCompass se selecciona un subconjunto de documentos de la colección para analizar para mejorar la eficiencia.