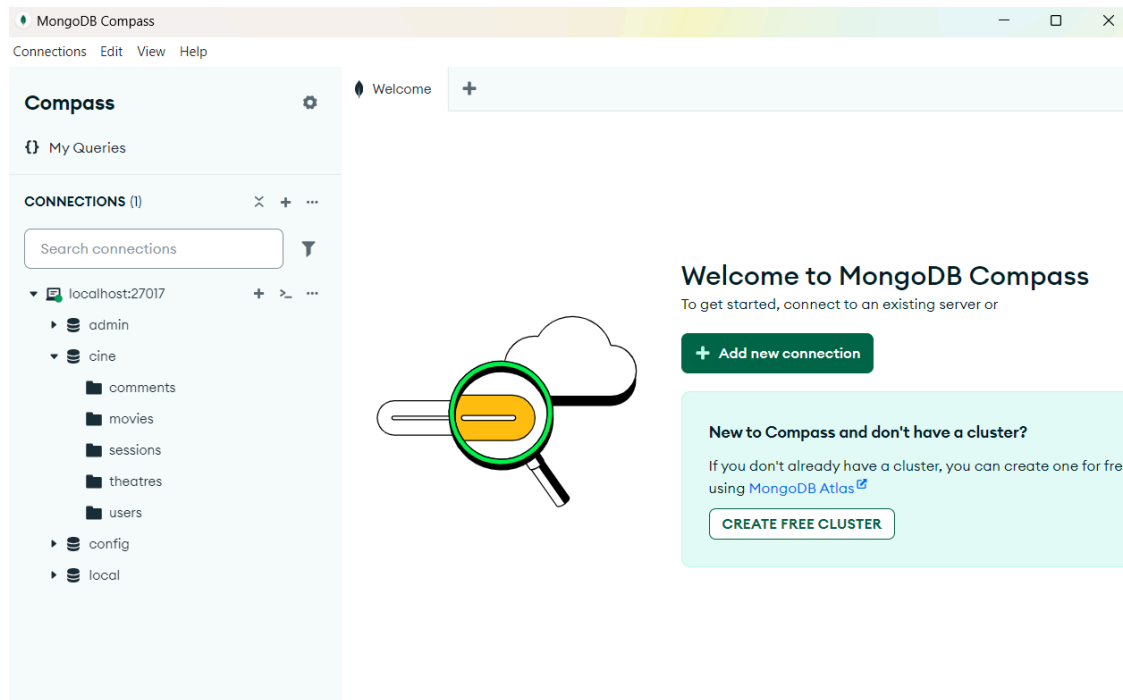


Tasca S9 MongoDB

Instalación de MongoDB

[Download MongoDB Community Server | MongoDB](#)

Luego en MongoDB Compass creamos una nueva base de datos 'cine' y añadimos los archivos .json como colecciones:



Descripción de la tarea

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

- **users:** Almacena información de usuarios/as, incluyendo nombres, correos electrónicos y contraseñas cifradas.
- **theatres:** Contiene datos de cines, como ID, ubicación (dirección y coordenadas geográficas).
- **sessions:** Guarda sesiones de usuario, incluyendo ID de usuario y tokens JWT para 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 solicitadas por el cliente/a, quien evaluará tu capacidad para encargarte del análisis de datos en su base de datos.

Nivel 1

En **MongoDB Shell** (similar a la línea de comandos) se puede escribir consultas especificando en qué **base de datos** estamos trabajando y qué **colección (tabla)** estamos consultando.

Ejercicio 1

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

db.comments.find().limit(2)

```
> 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 voluptates neque culpa molestias. Voluptate unde
  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.
  date: 1987-02-10T00:29:36.000Z
}
```

o dos más antiguos (si se trata de buscar primeros por fecha)

db.comments.find().sort({ "date": 1 }).limit(2)

```
> 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 voluptate esse.
  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 expedita dolore explic
  date: 1970-01-01T09:37:49.000Z
}
```

2. ¿Cuántos usuarios tenemos registrados?

db.users.countDocuments({})

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

3. ¿Cuántos cines hay en el estado de California?

```
db.theatres.countDocuments({ "location.address.state": "CA" })
```

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

4. ¿Quién fue el primer usuario/a en registrarse?

```
db.users.find().sort({ "registeredDate": 1 }).limit(1)
```

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

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

```
db.movies.countDocuments({$and: [ { "type": "movie" }, { "genres": "Comedy" } ]})
```

```
> db.movies.countDocuments({$and: [ { "type": "movie" }, { "genres": "Comedy" } ]})  
< 7002
```

Ejercicio 2

Muestra todos los documentos de las películas producidas en 1932, pero que sean de género drama o estén en francés.

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

```
> db.movies.find({ $and: [ { "type": "movie" }, { "year": 1932 }, { $or: [{ "genres": "Drama" }, { "languages": "French" }] } ] })
< {
  _id: ObjectId('573a1391f29313caabcd9458'),
  plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts talki
  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/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTxxXkEyXkFqcGdeQXVyNzI4MDMyNTU@
  title: 'The Blood of a Poet',
  lastupdated: '2015-09-16 13:13:05.537000000',
  languages: [
    'French'
  ],
  released: 2010-05-20T00:00:00.000Z,
  directors: [
    'Jean Cocteau'
  ],
  writers: [
    'Jean Cocteau'
  ],
  awards: {
```

devuelve 18 registros

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

Ejercicio 3

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

db.movies.find({ "countries": "USA", "awards.wins": { \$gte: 5, \$lte: 9 }, "year": { \$gte: 2012, \$lte: 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 sixte
  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 emb
  genres: [
    'Adventure',
    'Comedy',
    'Drama'
  ],
}
```

devuelve 166 registros

```
> db.movies.countDocuments({ "countries": "USA", "awards.wins": { $gte: 5, $lte: 9 }, "year": { $gte: 2012, $lte: 2014 } })
< 166
```

Nivel 2

Ejercicio 1

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

```
db.comments.countDocuments({"email": { $regex: "@GAMEOFTHRON.ES$", $options: "i" }})
```

```
> db.comments.countDocuments({
  "email": { $regex: "@GAMEOFTHRON.ES$", $options: "i" }
})
< 22841
```

Ejercicio 2

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

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

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

Nivel 3

Ejercicio 1

Encuentra todas las películas dirigidas por John Landis con una puntuación IMDb (Internet Movie Database) entre 7.5 y 8.

db.movies.find({ "type": "movie" , "directors": "John Landis", "imdb.rating": { \$gte: 7.5, \$lte: 8 } })

```
> db.movies.find( { "type": "movie" , "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
  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,
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
```

devuelve 4 registros

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

Ejercicio 2

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

db.theatres.createIndex({ "location.geo": "2dsphere" })

Verificamos la creación de índices

db.theatres.getIndexes()

```
> db.theatres.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { 'location.geo': '2dsphere' },
    name: 'location.geo_2dsphere',
    '2dsphereIndexVersion': 3
  }
]
```

Después de mucha búsqueda, he encontrado la opción de visualizar las ubicaciones de los teatros en la web de MongoDB Atlas. Primero, es necesario registrarse, y una vez hecho, puedes crear una base de datos en línea (supongo que con acceso limitado a un solo usuario en la versión gratuita).

Duplicando el archivo en Atlas, pude generar el mapa en el apartado de Visualización de los datos.

