

# Nivel 1

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

Para realizar consultas debemos estar dentro de la base de datos (db.) y seleccionar la colección con la que queremos trabajar.

## Ejercicio 1

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

Utilizo find() (equivalente a SELECT en SQL) y aplico .limit(2) para obtener únicamente los dos primeros documentos de la colección comments.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> 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 nulla temporibu
    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 ve
    date: 1987-02-10T00:29:36.000Z
  }
]
Cine>
```

- ¿Cuántos usuarios tenemos registrados?

Uso countDocuments() (equivalente a COUNT en SQL) para contar el total de documentos en la colección users.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.users.countDocuments()
< 185
```

- ¿Cuántos cines existen en el estado de California?

Filtro por el valor "CA" dentro del campo anidado location.address.state utilizando {} para definir el criterio.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.theaters.countDocuments({"location.address.state":"CA"})
< 169
Cine>
```

- ¿Cuál fue el primer usuario en registrarse?

Como no existe un campo de fecha, ordeno por \_id en orden ascendente (sort({\_id: 1})), porque los ObjectId más antiguos tienen un timestamp interno menor.

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

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

Filtro por el campo genres usando {genres: "Comedy"} y cuento los documentos.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.movies.countDocuments({ genres: "Comedy" })
< 7024
Cine>
```

## Ejercicio 2

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

Aplico tres criterios:

- año: {year: 1932}
- género drama
- idioma francés

Uso el operador \$or para combinar género e idioma:

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

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.movies.find({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 talking. The artist tri
  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/MV5BYWY3ODE5ZWEtYjlmVi00NjA4LTk4ZWYtMzBhZDE5MjY0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_SY1000_SX677_AL
  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: {
```

## Ejercicio 3

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

Aplico tres filtros:

- país: {countries: "USA"}
- premios ganados: awards.wins entre 5 y 9 usando \$gte y \$lte
- año entre 2012 y 2014, también con \$gte y \$lte.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> 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 magazine and has
  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 journey that tur
  genres: [
    'Adventure',
    'Comedy',
    'Drama'
  ],
  rated: 'PG',
  metacritic: 54,
  title: 'The Secret Life of Walter Mitty',
  lastupdated: '2015-08-31 00:10:51.747000000',
  languages: [
    'English',
    'Spanish',
    'Icelandic'
  ],
  writers: [
    'Steve Conrad (screenplay)',
```

## Nivel 2

### Ejercicio 1

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

Uso el operador `$regex` para buscar todas las direcciones de correo que contienen ese dominio.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.comments.countDocuments({"email":{"$regex" : "gameofthron.es"}})
< 22841
Cine>
```

### Ejercicio 2

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

Uso una agregación:

1. `$match` para filtrar por "location.address.state": "DC".
2. `$group` para agrupar por código postal.
3. `$sum` para contar cuántos cines hay en cada grupo.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> db.theaters.aggregate([{"$match": {"location.address.state": "DC" } },
                        {"$group": {"_id": "$location.address.zipcode", count: {"$sum": 1 } } }])
< {
  _id: '20002',
  count: 1
}
{
  _id: '20010',
  count: 1
}
{
  _id: '20016',
  count: 1
}
Cine> |
```

# 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.

Filtro por director y aplico un rango al campo imdb.rating utilizando \$gte y \$lte.

```
>_MONGOSH
> use Cine;
< switched to db Cine
> 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, anglo-saxon, rich young men w
  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 trouble-makers have other plan
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ],
  writers: [
    'Harold Ramis',
    'Douglas Kenney',
    'Chris Miller'
  ],
}
```

## Ejercicio 2

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

### 1. Crear el índice geoespacial

- En la colección theaters, pestaña Indexes.
- Crear un índice tipo 2dsphere sobre location.geo.

Este índice permite interpretar correctamente los datos de tipo Point y habilita la visualización geográfica.

### 2. Explorar el esquema

- Pestaña Schema → Analyze Schema.
- Identifico el campo location.geo.
- Compass detecta automáticamente los puntos gracias al índice 2dsphere.

### 3. Visualizar el mapa

- Desde el bloque location.geo en la vista del esquema, abro el mapa interactivo que muestra todos los teatros.

