

Alumna: Mariela Giménez

Sprint 5: MONGODB

Crea una base de datos con MONGODB utilizando MongoDB utilizando como colecciones los arxius adjunts.

The screenshot shows the MongoDB Compass interface for a database named 'cinematografica' on a local host. The interface displays five collections with their respective statistics:

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
comments	6.93 MB	50 K	284.00 B	1	524.29 kB
movies	21.55 MB	24 K	1.60 kB	1	364.54 kB
sessions	20.48 kB	1	540.00 B	1	36.86 kB
theatres	147.46 kB	1.6 K	223.00 B	1	32.77 kB
users	36.86 kB	185	159.00 B	1	20.48 kB

Como MongoDB trabaja con colecciones, se tiene que ir cargando CREANDO COLLECTION.

EJERCICIO 1

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

```
> db["comments"].find().limit(2) //1.1- Muestra los 2 primeros comentarios que hay en la base de datos.
< {
  _id: ObjectId('5a9427648b9beebe69579cc'),
  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
  date: 2012-03-26T23:20:16.000Z
}
{
  _id: ObjectId('5a9427648b9beebe69579cf'),
  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
  date: 1987-02-10T00:29:36.000Z
}
```

USANDO .sort() para que muestre por orden sort({ date: 1 }): El valor 1 indica orden ascendente y el -1 descendente.

```
> db["comments"].find().sort({date:1}).limit(2) // usando SORT()
< {
  _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.',
  date: 1970-01-01T01:07:09.000Z
}
{
  _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 cupiditate a tempore. Commodi alias expedita dolore explic',
  date: 1970-01-01T09:37:49.000Z
}
```

1.2- Cuantos usuarios tienen registrados?

```
> use cinematografica
< switched to db cinematografica
> db["users"].countDocuments(); // lo mas recomendado
< 185
```

Tambien se puede usar el COUNT() pero no es recomendado.

```
> db["users"].count();
< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
< 185
cinematografica> |
```

1.3- Cuantos cines hay en el estado de California?

```
> db["theatres"].aggregate([
  { $match: { "location.address.state": "CA" } },
  { $group: { _id: "$location.address.state", total: { $sum: 1 } } }
])
< {
  _id: 'CA',
  total: 169
}
```

El \$match es para filtrar. y luego con group agrupo para que me vaya diciendo cuantos cines hay en el estado de california.

1.4- Quien es el primer usuario/a 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'
}
cinematografica>
```

utilizando el OBJECTID() esta descifrado el DATE de cuando fue creado y quien y ordene por asc para saber quien fue el primero y el limit 1 para que salga solo un resultado.

1.5- Cuantas peliculas de comedia hay en nuestra base de datos?

```
> db["movies"].countDocuments({"genres": "Comedy"})
< 7024
```

EJERCICIO 2

2.1- Muestrame todos los documentos de las peliculas producidas en 1932, pero que el genero sea drama o esten en frances.

```
> db.movies.find({
  year: 1932,
  $or: [
    { genres: { $in: ["Drama"] } },
    { languages: { $in: ["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 a
  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/MV5BYWY30DE5ZWETyJlYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTUxXkEYXkFqcGdeQXVyNzI4MDMyNTU@._V1_SY1000
  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'
  ],
}
```

primero busco el año y luego la condicion de si el genero es DRAMA o Frances pero utilizo

el \$in porque podria no haber en algunos documentos.

EJERCICIO 3

3.1- Muestrame todos los documentos de peliculas estatunidenques que tengan entre 5 y 9 premios producidas entre el 2012 y 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 sixteen years for the magazine",
  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",
  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'
  ],
}
```

\$gte: Mayor que y \$lte: menor que porque en mongoDB se usa alfanumerico.

Nivel 2

EJERCICIO 1

1.1- Cuenta cuantos comentarios escribe un usuario/a que utiliza "GAMEOFTHRON.ES" como un dominio de correo electronico.

Este es el resultado de un solo Usuario

```

> // Paso 1: obtener UN usuario
const user = db.users.findOne({
  email: { $regex: /@gameofthron\.es$/i }
})

// Paso 2: contar comentarios de ese usuario
db.comments.countDocuments({
  email: user.email
})
< 273
cinematografica>

```

en caso de que lo que se quisiera es todos los usuarios que este usando el "GAMEOFTHRON.ES", primero se filtra los email distintos y se guarda en la variable EMAIL

```

> const emails = db["users"].distinct("email", { email: { $regex: /@gameofthron\.es$/i } })
> db["users"].distinct("email", { email: { $regex: /@gameofthron\.es$/i } })
< [
  'aidan_gillen@gameofthron.es',
  'alfie_allen@gameofthron.es',
  'amrita_acharia@gameofthron.es',
  'anton_lesser@gameofthron.es',
  'art_parkinson@gameofthron.es',
  'ben_crompton@gameofthron.es',
  'ben_hawkey@gameofthron.es',
  'brenock_o'connor@gameofthron.es',
  'brian_fortune@gameofthron.es',
  'carice_van_houten@gameofthron.es',
  'charles_dance@gameofthron.es',
  'conleth_hill@gameofthron.es',
  'daniel_portman@gameofthron.es',
  'dean-charles_chapman@gameofthron.es',
  'diana_rigg@gameofthron.es',
  'dominic_carter@gameofthron.es',
  'donald_sumpter@gameofthron.es',
  'ellie_kendrick@gameofthron.es',
  'emilia_clarke@gameofthron.es',
  'esme_bianco@gameofthron.es',
  'eugene_simon@gameofthron.es',
  'faye_marsay@gameofthron.es',
  'finn_jones@gameofthron.es',
  'gemma_whelan@gameofthron.es',
  'mwendolene_christie@gameofthron.es'
]

```

y ahora en la tabla de comentarios cuento cuantos email hay.

```

> db.comments.countDocuments({
  email: { $in: emails }
})
< 22841
cinematografica>

```

EJERCICIO 2

2.1.1- Cuantos cines hay en cada codigo postal situados dentro de el estado de Washington D. C. (DC)?

agrupamos para poder filtrar (\$match) y luego agrupamos por codigo postal e ir sumando cada vez que encuentre una coincidencia y al final se ordena por descendente

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

Nivel 3

EJERCICIO 1

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

Filtrar por el director y el imdb entre 7,5 y 8 y seleccionar solo las columnas para poder visualizar

```
> db["movies"].find(  
  {  
    directors: "John Landis",  
    "imdb.rating": { $gte: 7.5, $lte: 8 }  
  },  
  {  
    title: 1,  
    "imdb.rating": 1,  
    directors: 1,  
    _id: 0  
  }  
)
```

Resultados

```

{
  imdb: {
    rating: 7.6
  },
  title: 'Animal House',
  directors: [
    'John Landis'
  ]
}
{
  title: 'The Blues Brothers',
  directors: [
    'John Landis'
  ],
  imdb: {
    rating: 7.9
  }
}
{
  imdb: {
    rating: 7.6
  },
  title: 'An American Werewolf in London',
  directors: [
    'John Landis'
  ]
}
{
  title: 'Trading Places',
  directors: [
    'John Landis'
  ],
  imdb: {
    rating: 7.5
  }
}
inematografica>

```


Exercici 2

3.1.2- Muestra en un mapa la ubicacion de todos los teatros de la base de datos.

