

# Sprint 5: Consultas con MongoDB

Curso de especialización Data Analytics - IT Academy - Barcelona Activa

Nicolás Aros Marzá

23/01/2026

## Nivel 1

**Crea una base de dades amb MongoDB utilitzant com a col·leccions els arxius adjunts.**

La base de datos fue titulada **Cine** y fue creada en un servidor local. Dentro de esta base de datos se crearon cinco colecciones para contener los documentos guardados en los archivos **.json** que fueron entregados. Estas colecciones tienen por título **comments**, **movies**, **sessions**, **theaters** y **users**.

## Ejercicio 1

Mostra els 2 primers comentaris que hi ha en la base de dades.

Para seleccionar los dos primeros comentarios (o los más antiguos) que hay en la base de datos **Cine**, se realizó una consulta en la colección **comments**. Para esto, los documentos se ordenaron de manera ascendente según los valores de la variable **date**. Luego, se ajustó el límite (**limit**) a dos observaciones. En la imagen se muestran los documentos seleccionados.

Primeros documentos de colección comments.

The screenshot shows the MongoDB Compass interface for the 'comments' collection in the 'Cine' database. The query bar shows a query: `{ field: 'value' }` or `Generate query`. The sort is set to `{date: 1}`. The limit is set to 2. The interface displays two documents in a table view.

	_id ObjectId	name String	email String	movie_id ObjectId	text String	date Date
1	ObjectId('5a9427648b0bee...')	"Mercedes Tyler"	"mercedes_tyler@fakegmai..."	ObjectId('573a1399f29313...')	"Optio totam dolores mag..."	1978-01-01T01:07:09.000+...
2	ObjectId('5a9427658b0bee...')	"Jon Snow"	"kit_harington@gameofthr..."	ObjectId('573a13a4f29313...')	"Dolorem animi tempora u..."	1978-01-01T09:37:49.000+...

## Quants usuaris tenim registrats?

Para mostrar el total de usuarios registrados se utilizó el marco de agregación de MongoDB, el cual permite realizar operaciones de resumen. En este caso, se empleó la etapa `$count` dentro del pipeline. En la imagen se muestra que en la colección `users` se hay 185 usuarios.

Este valor también se puede consultar directamente en la pestaña **Documents** de cada colección.

Para evitar contar valores duplicados en el total, por ejemplo, personas con el mismo correo electrónico, se debe agregar una etapa `$group` antes de `$count` en el pipeline, agrupando por el campo que se desea contar de manera única.

Cantidad de usuarios registrados.

The screenshot shows the MongoDB Compass interface for the `users` collection. The top bar indicates 185 documents. The aggregation pipeline is configured with a single stage named `$count` and the field `n_users`. The output preview shows `n_users : 185`. The interface also includes tabs for Documents, Aggregations, Schema, Indexes, and Validation, as well as buttons for Generate aggregation, Explain, Run, and Options.

## Quants cinemes hi ha en l'estat de Califòrnia?

Para conocer la cantidad de cines que hay en California se realizó una consulta en la colección **theaters** que filtra los documentos cuyo campo **location.address.state** es igual a "CA". Este filtro permite seleccionar únicamente los cines ubicados en ese estado. Luego, en la segunda etapa **\$count** se resume la información en **n\_cinemas\_ca** que muestra que hay 169 cines en el estado de California.

Cantidad de cines en California (CA).

The screenshot shows the MongoDB Aggregations Builder interface. The pipeline consists of two stages:

- Stage 1: \$match** (toggle on). The query is `{ "location.address.state": "CA" }`. The output preview shows a sample of 10 documents, each with `_id`, `theaterId`, and `location` fields.
- Stage 2: \$count** (toggle on). The query is `'n_cinemas_ca'`. The output preview shows a sample of 1 document with the value `n_cinemas_ca : 169`.

At the top, the interface shows the collection `theaters` and the `Aggregations` tab. Buttons for `$match` and `$count` are visible. The top right has a button to `Open MongoDB shell`. The bottom of the interface has buttons for `SAVE`, `CREATE NEW`, `EXPORT DATA`, and `EXPORT CODE`. The bottom right has buttons for `PREVIEW`, `STAGES`, `TEXT`, and `WIZARD`.

## Quin va ser el primer usuari/ària en registrar-se?

La colecció **users** no cuenta con una variable directa que indique el momento en que se creó cada cuenta de usuario, sin embargo, esta información se encuentra disponible en el campo **\_id**. Ante esto, en la primera etapa (**\$addFields**) se crea una nueva variable llamada **date\_account** que guarda el **timestamp** extraído de la variable **\_id** mediante el comando **\$toDate**. Luego, en la etapa **\$sort** se ordenan de manera ascendente los documentos por el campo recién creado **date\_account**, para que los documentos más antiguos aparecen primero. Finalmente mediante la etapa **\$limit** se muestra solo el primero, que sería el registrado más temprano.

### Identificar primer usuario (paso 1)

The screenshot shows the MongoDB Compass interface for the **users** collection. The aggregation pipeline is defined as follows:

```
1 {
2   "date_account": {"$toDate": "$_id"}
3 }
```

The output preview after the **\$addFields** stage (Sample of 10 documents) shows the following documents:

```
{
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),
  "name": "Ned Stark",
  "email": "sean_bean@gameofthron.es",
  "password": "$2b$12$UREFwsRUoyF0CRqGNK0Lz00H...",
  "date_account": 2017-09-13T21:05:56.000+00:00
}
```

The first document is the earliest user, as indicated by the **date\_account** field.

## Identificar primer usuario (paso 2)

localhost:27017 > Cine > users Open MongoDB shell

Documents 185 **Aggregations** Schema Indexes 1 Validation

+ \$addFields \$sort \$limit Generate aggregation ? Explain Run Options

Untitled - modified SAVE + CREATE NEW EXPORT DATA EXPORT CODE PREVIEW STAGES TEXT WIZARD

Stage 2 \$sort ON

```
1 {  
2   "date_account": 1  
3 }
```

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

<code>_id: ObjectId('59b99db4cfa9a34dcd7885b7')</code> <code>name: "Robert Baratheon"</code> <code>email: "mark_addy@gameofthron.es"</code> <code>password: "\$2b\$12\$yGqXLG9LZpXA2xVDhuPnS0Zd...."</code> <code>date_account: 2017-09-13T21:05:56.000+00:00</code>	<code>_id: ObjectId('59b99db4cfa9a34dcd7885b6')</code> <code>name: "Ned Stark"</code> <code>email: "sean_bean@gameofthron.es"</code> <code>password: "\$2b\$12\$UREFwsRUoyF0CRqGNK0L..."</code> <code>date_account: 2017-09-13T21:05:56.000+00:00</code>
--	--

Stage 3 \$limit ON

```
1 1
```

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

<code>_id: ObjectId('59b99db4cfa9a34dcd7885b6')</code> <code>name: "Ned Stark"</code> <code>email: "sean_bean@gameofthron.es"</code> <code>password: "\$2b\$12\$UREFwsRUoyF0CRqGNK0L..."</code> <code>date_account: 2017-09-13T21:05:56.000+00:00</code>
--

## Quantes pel·lícules de comèdia hi ha en la nostra base de dades?

En la colecció `movies` se hace una primera etapa `$match` que filtra a todas las películas de género de comedia (`genre: "Comedy"`). En la segunda etapa `$count` se calcula el total de documentos que cumplen ese criterio. En la imagen se muestra que son 7024 películas.

Cantidad de películas de comedia.

The screenshot shows the MongoDB Aggregations Builder interface. The top bar indicates the database is `localhost:27017 > Cine > movies`. The `Aggregations` tab is selected, showing a pipeline with two stages: `$match` and `$count`. The `$match` stage has a filter `{genres: "Comedy"}`. The `$count` stage has an output field `n_comedy`. The output preview for the `$count` stage shows a single document: `{n_comedy: 7024}`.

```
1
2 {
3   genres: "Comedy"
4 }
```

```
1
2 {
3   n_comedy: 7024
4 }
```

## Ejercicio 2

Mostra'm tots els documents de les pel·lícules produïdes en 1932, però que el gènere sigui drama o estiguin en francès.

Esta consulta requiere filtrar los documentos de la colección `movies` por varios criterios. Para esto, primero se indica el criterio según el año de producción de la película y luego, se incluye el operador lógico `$or` que permite integrar los criterios de género e idioma. Este filtro selecciona los documentos que cumplan con al menos uno de estos dos últimos criterios. En la imagen se observa que hay 18 películas con estas características.

Documentos películas de 1932 que sean dramas o estén en francés.

localhost:27017 > Cine > movies Open MongoDB shell

Documents 24K Aggregations Schema Indexes 1 Validation

`{`  
  `year: 1932,`  
  `$or: [`  
    `{ genres: "Drama" },`  
    `{ languages: "French" }`  
  `]`  
`}`

Generate query Explain Reset Find Options

ADD DATA UPDATE DELETE EXPORT DATA EXPORT CODE

50 1 - 18 of 18

`{`  
  `_id: ObjectId("573a1391f29313caabcd9458")`  
  `plot: "A young artist draws a face at a canvas on his easel. Suddenly the mou..."`  
  `runtime: 50`  
  `rated: "UNRATED"`  
  `cast: Array (4)`  
  `num_reflix_comments: 1`  
  `poster: "https://m.media-amazon.com/images/M/MV5BVMY3ODE5ZWVjLnY1ODk4LTk4Zm..."`  
  `title: "The Blood of a Poet"`  
  `lastupdated: "2015-09-16 13:13:05.537000000"`  
  `languages: Array (1)`  
  `released: 2010-05-20T00:00:00.000+00:00`  
  `directors: Array (1)`  
  `writers: Array (1)`  
  `awards: Object`  
  `year: 1932`  
  `imdb: Object`  
  `countries: Array (1)`  
  `type: "movie"`  
  `tomatoes: Object`  
`}`

`{`  
  `_id: ObjectId("573a1392f29313caabcd99a3")`  
  `plot: "Junta is hated by the people in the village where she lives, especialL..."`  
  `genres: Array (3)`  
  `runtime: 85`  
  `cast: Array (4)`  
  `poster: "https://m.media-amazon.com/images/M/MV5BNQ1NTM2HTQ1ODIyY5S0MTAxLWE1NT..."`  
  `title: "The Blue Light"`  
  `fullplot: "Junta is hated by the people in the village where she lives, especialL..."`  
  `languages: Array (2)`  
  `released: 1934-05-08T00:00:00.000+00:00`  
  `directors: Array (2)`  
  `writers: Array (2)`  
  `awards: Object`  
  `lastupdated: "2015-08-03 01:02:12.350000000"`  
  `year: 1932`  
`}`

### Ejercicio 3

Mostra'm tots els documents de pel·lícules estatunidenques que tinguin entre 5 i 9 premis que van ser produïdes entre 2012 i 2014.

En este ejercicio se incluyen tres criterios para seleccionar documentos de la colección `movies`. A diferencia de los ejercicios anteriores, se establece un rango para las variables numéricas `year` y `awards.wins`. Este rango se define en los operadores `$gte` (límite inferior) y `$lte` (límite superior).

En la imagen se observa que hay 166 películas que cumplen con los criterios establecidos en el filtro.

Documentos películas de entre 2012 y 2014 que tengan entre 5 y 9 premios.

The screenshot shows the MongoDB Compass interface. At the top, the breadcrumb navigation indicates the path: `localhost:27017 > Cine > movies`. Below this, there are tabs for `Documents` (24K), `Aggregations`, `Schema`, `Indexes` (1), and `Validation`. The `Documents` tab is active.

A query filter is applied, shown in the top left of the document list: `{ countries: "USA", awards.wins: { $gte: 5, $lte: 9 }, year: { $gte: 2012, $lte: 2014 } }`. To the right of the filter are buttons for `Generate query`, `Explain`, `Reset`, `Find`, and `Options`.

Below the filter, there are action buttons: `ADD DATA`, `UPDATE`, `DELETE`, `EXPORT DATA`, and `EXPORT CODE`. On the right, a pagination bar shows `50` items per page, `1 - 50 of 166` total items, and navigation arrows.

The document list displays two results. Each document is a JSON object with the following structure:

- `_id`: ObjectId (e.g., `'573a13cf29313caab742df'`)
- `fullplot`: A string describing the film.
- `imdb`: An object containing `year`, `plot`, `genres`, `rated`, `metacritic`, `title`, `lastupdated`, `languages`, `writers`, `type`, `tomatoes`, `poster`, `num_mflix_comments`, `released`, `awards`, `countries`, `cast`, `directors`, and `runtime`.

The first document shown is for the movie "Prometheus" (year 2012, rated R, metacritic 65). The second document shown is for the movie "That's My Boy" (year 2012, rated R, metacritic 31).

## Nivel 2

### Ejercicio 1

Compte quants comentaris escriu un usuari/ària que utilitza “@GAME-OFTHRON.ES” com a domini de correu electrònic.

Para seleccionar los usuarios registrados con correos electrónicos con dominio @gameofthron.es, se utiliza el operador \$regex.

En la expresión utilizada (\$regex: "@gameofthron\\.es\$"), el símbolo \$ se coloca al final para asegurar que la coincidencia ocurra estrictamente al terminar la cadena.

Se debe agregar una doble barra invertida (\\) antes del punto para evitar un error en la sintaxis.

Cantidad de comentarios de usuarios con correos de gameofthron.

The screenshot shows the MongoDB Compass interface with an aggregation pipeline applied to the 'comments' collection. The pipeline consists of two stages: a \$match stage and a \$count stage.

**Stage 1: \$match**

```
1 {
2   email:
3     { $regex: "@gameofthron\\.es$" }
4 }
5
```

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

_id	name	email	movie_id	text	date
ObjectId('5a9427648b0beeb69579d0')	Talisa Maegyr	oona_chaplin@gameofthron.es	ObjectId('573a1396f29313caab041b...')	Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ...	1998-08-22T11:45:03.000+00:00
ObjectId('5a9427648b0beeb69579d5')	Petyr Baelish	aidan_gillien@gameofthron.es	ObjectId('573a1396f29313caab0421...')	Quo deserunt ipsum ipsum. Tenetur eos nemo nam sint praesentium minus ...	2001-07-13T19:25:09.000+00:00
ObjectId('5a9427648b0beeb69579f9')	Ollivander	brenock.o'connor@gameofthron.es	ObjectId('573a1396f29313caab043a...')	Perferendis officia harum ipsum delenti.	2005-01-04T13:49:05.000+00:00

**Stage 2: \$count**

```
1 'n_comments_email_got'
```

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

n_comments_email_got
22841

## Ejercicio 2

Quants cinemes hi ha en cada codi postal situats dins de l'estat Washington D. C. (DC)?

En esta consulta se utiliza la colección `theaters`. La primera etapa es filtrar (`$match`) solamente a los cines que tengan el valor "DC" en la variable `location.address.state`. En la segunda etapa se agrupan por código postal (`location.address.zipcode`) y crea la variable `n_cinema` que muestra la cantidad de cines en cada código postal.

En la imagen se muestra que en cada una de las tres área postales hay un cine.

Cantidad de cines en cada código postal del Estado de Washington D.C.

The screenshot displays the MongoDB Aggregations Builder interface. The top navigation bar shows 'localhost:27017 > Cine > theaters'. The 'Aggregations' tab is active, showing two stages:

- Stage 1: \$match**  
Query: 

```
1 {
2   "location.address.state": "DC"
3 }
```

  
Output preview after \$match stage (Sample of 3 documents):  

<pre>{   "_id": ObjectId("59a47286cfa9a3a73e51e785"),   "theaterId": 1092,   "location": {     "address": {       "state": "DC",       "zipcode": "20002"     }   } }</pre>	<pre>{   "_id": ObjectId("59a47287cfa9a3a73e51ec33"),   "theaterId": 881,   "location": {     "address": {       "state": "DC",       "zipcode": "20010"     }   } }</pre>	<pre>{   "_id": ObjectId("59a47287cfa9a3a73e51ec33"),   "theaterId": 8500,   "location": {     "address": {       "state": "DC",       "zipcode": "20016"     }   } }</pre>
---	--	---
- Stage 2: \$group**  
Query: 

```
1 {
2   "_id": "$location.address.zipcode",
3   "n_cinema": {
4     "$sum": 1
5   }
6 }
```

  
Output preview after \$group stage (Sample of 3 documents):  

<pre>{   "_id": "20002",   "n_cinema": 1 }</pre>	<pre>{   "_id": "20010",   "n_cinema": 1 }</pre>	<pre>{   "_id": "20016",   "n_cinema": 1 }</pre>
--	--	--

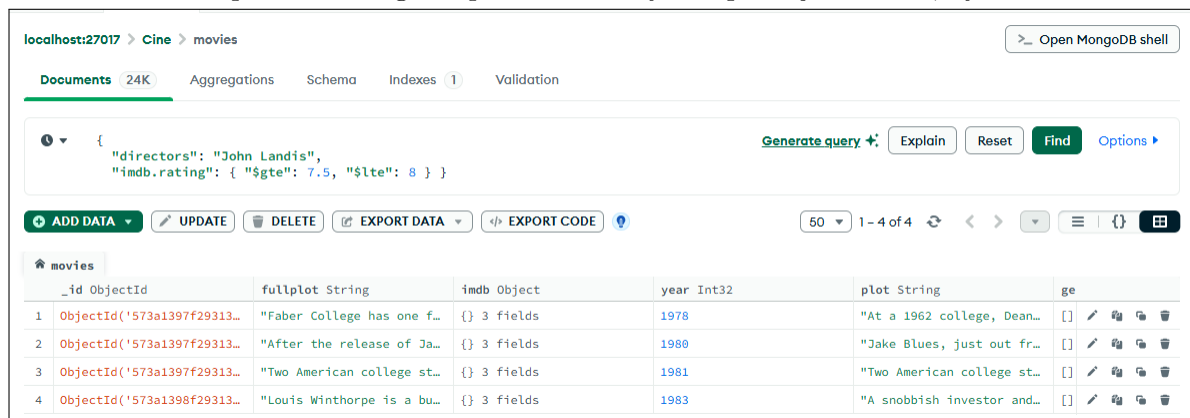
## Nivel 3

### Ejercicio 1

Troba totes les pel·lícules dirigides per John Landis amb una puntuació IMDb (Internet Movie Database) d'entre 7,5 i 8.

Para esta consulta se usa la misma estrategia del ejercicio 3 del nivel 1. Es decir, en la colección `movies` se filtran todas las películas dirigidas por John Landis y luego, se establece un criterio en la variable `imdb.rating` para que se seleccionen los documentos que tengan un valor entre 7.5 y 8. Los límites del rango se definen en los operadores `$gte` (límite inferior) y `$lte` (límite superior).

Documentos películas dirigidas por J. Landis y con puntajes entre 7,5 y 8 en IMDb.



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to `localhost:27017` and the database `Cine`, with the collection `movies` selected. The `Documents` tab is active, showing 24K documents. A query is entered in the query editor:

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

Buttons for `Generate query`, `Explain`, `Reset`, `Find`, and `Options` are visible. Below the query editor, there are buttons for `ADD DATA`, `UPDATE`, `DELETE`, `EXPORT DATA`, and `EXPORT CODE`. The results table shows 4 documents, each with fields: `_id`, `fullplot`, `imdb`, `year`, `plot`, and `ge`.

	_id ObjectId	fullplot String	imdb Object	year Int32	plot String	ge
1	ObjectId('573a1397f29313...')	"Faber College has one f...	{ } 3 fields	1978	"At a 1962 college, Dean...	[ ]
2	ObjectId('573a1397f29313...')	"After the release of Ja...	{ } 3 fields	1980	"Jake Blues, just out fr...	[ ]
3	ObjectId('573a1397f29313...')	"Two American college st...	{ } 3 fields	1981	"Two American college st...	[ ]
4	ObjectId('573a1398f29313...')	"Louis Winthorpe is a bu...	{ } 3 fields	1983	"A snobbish investor and...	[ ]

## Ejercicio 2

Mostra en un mapa la ubicació de tots els teatres de la base de dades.

Para mostrar un mapa la ubicación de los cines que hay en la base de datos se utiliza la pestaña **Schema** y se presiona el botón **Analyze**. Luego, se encuentra una subpestaña con un mapa interactivo que muestra un punto azul para cada uno de los cines. Esta información está guardada en la variable `location.geo.coordinates` en un sistema de coordenadas.

Mapa de localización de los cines.

