

# SPRINT 5



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

He creado la base de datos mediante la interfaz **MongoDB Compass** y he tenido que insertar una primera colección en el momento de la creación de la base de datos.(Hay el símbolo “+” al lado de “localhost:27017” a la izq. de la pantalla)

A screenshot of the MongoDB Compass interface. On the left, there's a sidebar with 'My Queries' and 'Data Modeling' sections, and a 'CONNECTIONS' section showing a connection to 'localhost:27017' with sub-options for 'admin', 'config', and 'local'. The main area shows a table with three databases: 'admin' (Storage size 20.48 kB, 1 collection, 1 index), 'config' (Storage size 1 kB, 1 collection, 2 indexes), and 'local' (Storage size 1 kB, 1 collection, 1 index). A modal window titled 'Create Database' is open in the center. It has fields for 'Database Name' (set to 'Cinema') and 'Collection Name' (set to 'comments'). There's a checkbox for 'Time-Series' which is unchecked. Below the checkbox is a link 'Time-series collections efficiently store sequences of measurements over a period of time. Learn More'. At the bottom of the modal are 'Cancel' and 'Create Database' buttons. Above the modal, there are buttons for 'Open MongoDB shell', '+ Create database', and 'Refresh'.

Después he ido **creando**, una por una, las distintas **colecciones** de la base de datos, mediante la interfaz a través de **Create Collection**  
(Hay el símbolo “+” al lado de “Cinema” a la izq. de la pantalla)

A screenshot of the MongoDB Compass interface. The left sidebar is identical to the previous screenshot. The main area shows the 'Cinema' database selected. Inside the 'Cinema' folder, there's a 'comments' collection. A modal window titled 'Create Collection' is open. It has a field for 'Collection Name' which is empty. Below it is a checkbox for 'Time-Series' which is unchecked. There's also a link 'Time-series collections efficiently store sequences of measurements over a period of time. Learn More'. At the bottom of the modal are 'Cancel' and 'Create Collection' buttons. The background shows the 'Documents' tab of the 'comments' collection, with a search bar at the top and various data manipulation buttons below.

Ahora que tengo las varias colecciones puedo cargar los ficheros con la opción **Add data**:

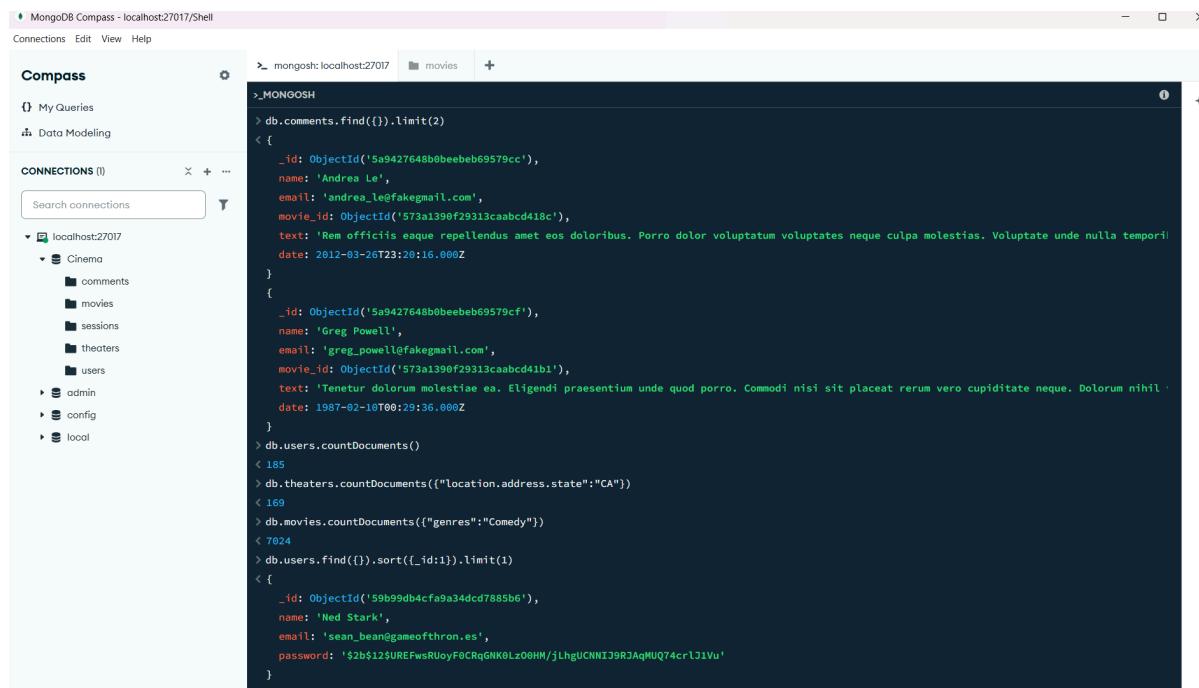
The screenshot shows the Compass MongoDB interface. On the left, the 'CONNECTIONS' sidebar lists 'localhost:27017' with its collections: 'Cinema' (containing 'comments', 'movies', 'sessions', 'theatres', 'users'), 'admin', 'config', and 'local'. The main area shows the 'localhost:27017 > Cinema > comments' collection. At the top, there are tabs for 'Documents' (0), 'Aggregations', 'Schema', and 'Indexe'. Below the tabs is a search bar with placeholder text 'Type a query: { field: 'value' } or [Generate](#)'. Underneath the search bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and a trash can icon. A large dark overlay box covers the middle section, containing the text 'Import JSON or CSV file' and 'Insert document'.

Después de cargar todos los fichero tengo la base de datos “Cinema” completa y lista para las consultas :

This screenshot shows the 'users' collection in the 'Cinema' database. The 'Documents' tab is selected, showing 185 documents. The interface includes a search bar, a toolbar with 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE' buttons, and a results grid showing five sample document cards. Each card displays a document's \_id, name, email, and password. The cards are scrollable, and the bottom right corner of the interface shows a small '100' and '1 - 100 of 185' indicator.

## Exercici 1

- Mostra els 2 primers comentaris que hi ha en la base de dades.
- Quants usuaris tenim registrats?
- Quants cinemes hi ha en l'estat de Califòrnia?
- Quin va ser el primer usuari/ària en registrar-se?
- Quantes pel·lícules de comèdia hi ha en la nostra base de dades?

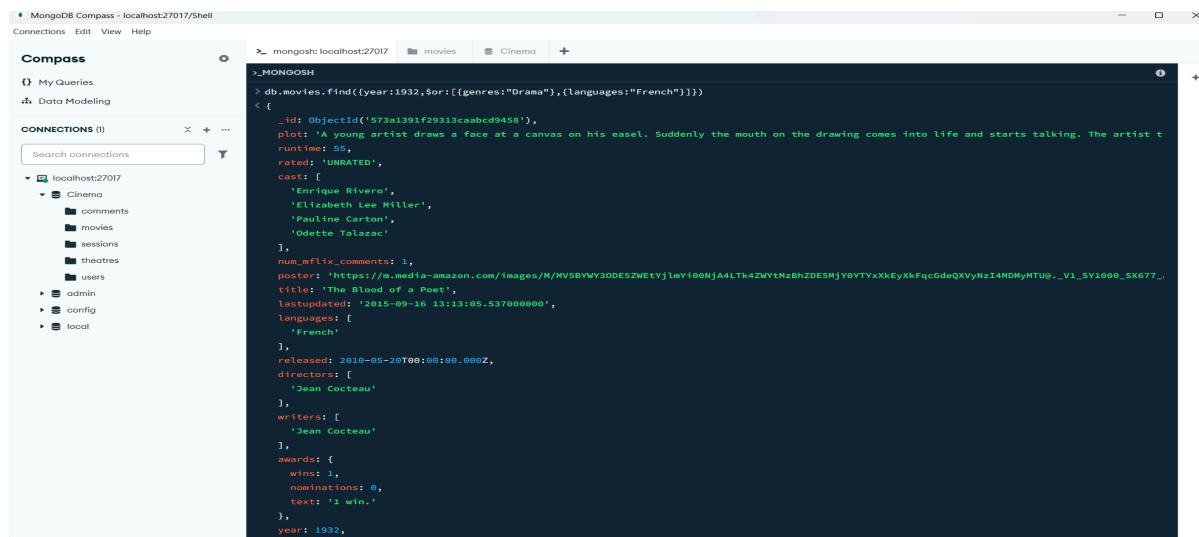


The screenshot shows the MongoDB Compass interface with a shell session titled 'mongosh: localhost:27017'. The session contains the following MongoDB code:

```
>_MONGOSH
> db.comments.find({}).limit(2)
< [
  {
    _id: ObjectId('5a9427648b0beebe69579cc'),
    name: 'Andrea Lee',
    email: 'andrea_leefake@gmail.com',
    movie_id: ObjectId('573a1390f29313caabcd418c'),
    text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporil',
    date: 2012-03-26T23:20:16.000Z
  },
  {
    _id: ObjectId('5a9427648b0beebe69579cf'),
    name: 'Greg Powell',
    email: 'greg_powellfake@gmail.com',
    movie_id: ObjectId('573a1390f29313caabcd41b1'),
    text: 'Tenetur dolorum molestiae ea. Eligiendi praesentium unde quod porro. Commodi nisi sit placeat rerum vero cupiditate neque. Dolorum nihil .',
    date: 1987-02-10T00:29:36.000Z
  }
]
> db.users.countDocuments()
< 185
> db.theaters.countDocuments({"location.address.state":"CA"})
< 169
> db.movies.countDocuments({"genres":"Comedy"})
< 7024
> db.users.find({}).sort({_id:1}).limit(1)
< [
  {
    _id: ObjectId('59b99db4cfa9a34cd7885b6'),
    name: 'Ned Stark',
    email: 'sean_bean@gameofthrones.es',
    password: '$2b$12$UREFwsR0yF0CrqGNK0Lz00H/jLhgUCNNIj9RJAqHUQ74crlJ1Vu'
  }
]
```

## Exercici 2

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

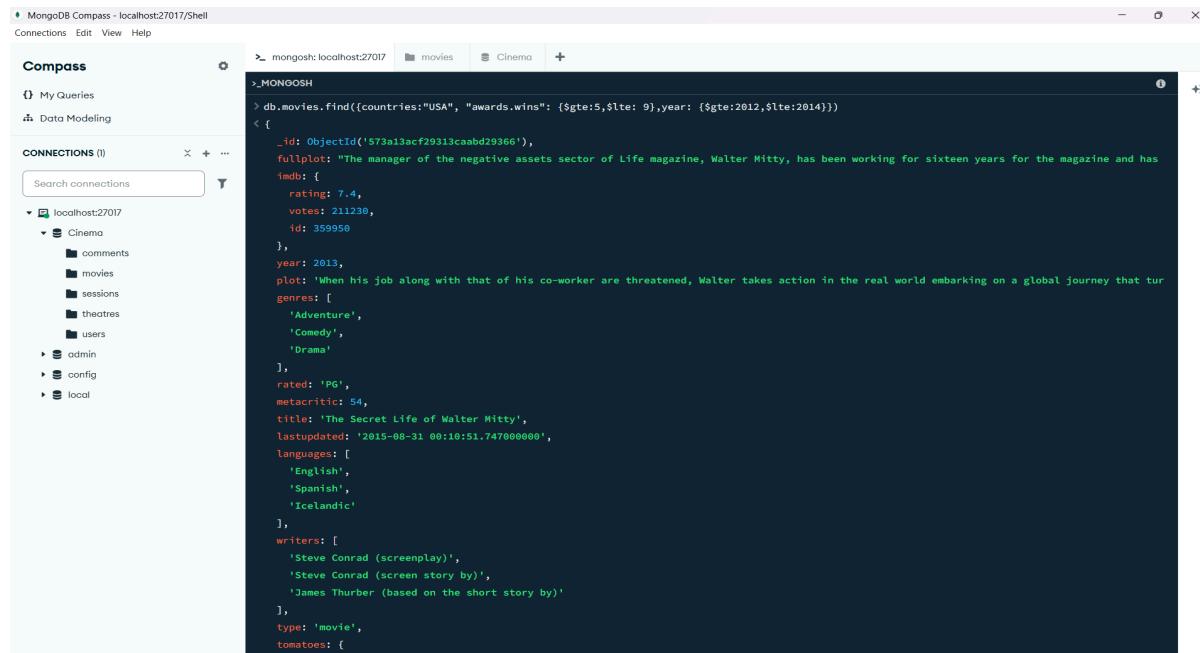


The screenshot shows the MongoDB Compass interface with a shell session titled 'mongosh: localhost:27017'. The session contains the following MongoDB code:

```
>_MONGOSH
> db.movies.find({year:1932,sort:[{"genres":"Drama"}, {"languages:"French"}]})
< [
  {
    _id: ObjectId('573a1391f29313caabcd458'),
    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 t',
    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/MV5BYWY30DE52WETyjlaYt00NjA4LTk4ZWVtMzBhZDE5MjY0YTtxXkEyXkFqcGdeQXVxNzI4MDMyMTU0._V1_SV1000.SX677_.',
    title: 'The Blood of a Poet',
    lastupdated: '2015-09-16 13:13:05.537000000',
    languages: [
      'French'
    ],
    released: 2016-05-26T00:00:00.000Z,
    directors: [
      'Jean Cocteau'
    ],
    writers: [
      'Jean Cocteau'
    ],
    awards: {
      wins: 1,
      nominations: 0,
      text: '1 win.'
    },
    year: 1932
  }
]
```

## Exercici 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.



The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'localhost:27017' with its databases: 'Cinema', 'comments', 'movies', 'sessions', 'theatres', and 'users'. The 'admin', 'config', and 'local' databases are also listed. The main panel displays a mongo shell command and its execution results:

```
>_MONGOOSH
> db.movies.find({countries:"USA", "awards.wins": {$gte:5,$lte: 9}},year: {$gte:2012,$lte:2014})
< [
  {
    _id: ObjectId('573a13acf29313caabdd29366'),
    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: 359958
    },
    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)',
      'Steve Conrad (Screen story by)',
      'James Thurber (based on the short story by)'
    ],
    type: 'movie',
    tomatoes: [

```

## Commentario:

En el ejercicio 1 he utilizado la función base **find()** para consultar documentos en las colecciones.

He combinado esta función con **limit()** para restringir el número de resultados, con **countDocuments()** para contar registros y con **sort()** para ordenarlos.

Después en el ejercicio 2 he utilizado el operador lógico **\$or**.

En el ejercicio 3 he visto cómo acceder a campos anidados a través de la sintaxis compuesta así : “**award.wins**”.



## Nivell 2

### Exercici 1

Compte quants comentaris escriu un usuari/ària que utilitza "GAMEOFTHRON.ES" com a domini de correu electrònic.

```
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
localhost:27017
Cinema
mongosh: localhost:27017 | comments | +
>_MONGOSH
> db.comments.countDocuments({email:{$regex:/GAMEOFTHRON\.ES$/i}})
< 22841
Cinema>
```

Aquí he utilizado la función **\$regex** para buscar un texto exacto.

Es importante el uso de los delimitadores “/...../”, el símbolo \$ para indicar donde acaba el texto y la barra invertida “\” para escapar caracteres especiales como el punto. Además, la letra “i” al final indica que la búsqueda es insensible a mayúsculas y minúsculas.

### Exercici 2

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

```
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
localhost:27017
Cinema
comments
movies
sessions
theatres
users
admin
config
mongosh: localhost:27017 | theatres | +
>_MONGOSH
> db.theatres.aggregate([
  {$match:{'location.address.state' : "DC"}},
  {$group:{'_id':'$location.address.zipcode", total_cinemas:$sum:1}},
  {$sort:{total_cinemas:-1}}])
< [
  {
    "_id: '20002',
    total_cinemas: 1
  },
  {
    "_id: '20010',
    total_cinemas: 1
  },
  {
    "_id: '20016',
    total_cinemas: 1
  }
]
```

**Aggregate**: es una función avanzada que nos permite insertar una lista de comandos que se ejecutan uno por uno en orden:

**Match** filtra la colección “theatres” que tengan state = “DC”.

**Group** agrupa los documentos para el campo zipcode.

(“**\_id**” en este caso funciona como palabra clave)

**Sum** dentro de cada grupo cuenta cuántos documentos hay

**Sort** ordena la salida del resultado,-1 da mayor a menor



## Nivell 3

### Exercici 1

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

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with 'Compass' and sections for 'My Queries' and 'Data Modeling'. Under 'CONNECTIONS', it lists 'localhost:27017' with a 'Cinema' database selected, containing collections for 'comments', 'movies', 'sessions', 'theatres', and 'users'. The main area is a code editor with the title 'mongosh: localhost:27017/movies'. It contains a MongoDB shell command to find movies directed by John Landis with an IMDb rating between 7.5 and 8.0. The output shows one document from the 'movies' collection, which is a movie titled 'The Blues Brothers'.

```
>_MONGOSH
> db.movies.aggregate([
  { $match:{directors:"John Landis",
            "imdb.rating":{$gte:7.5,$lte:8}}},
  {$sort:{"imdb.rating":-1}}]
< [
  {
    _id: ObjectId('573a1397f29313caabce76f7'),
    plot: 'Jake Blues, just out from prison, puts together his old band to save the Catholic home where he and brother Elwood were raised.',
    genres: [
      'Action',
      'Comedy',
      'Crime'
    ],
    runtime: 133,
    rated: 'R',
    cast: [
      'John Belushi',
      'Dan Aykroyd',
      'James Brown',
      'Cab Calloway'
    ],
    num_aflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BYTdlMDExOGUtN2I3MS00MjY5LWE1NTAtYzc3MzIxN2M3OWY1XKEyXkFqcGdeQXVyNzkwNjQ5NzN0LVI_SV1000_SK677_AL',
    title: 'The Blues Brothers',
    fullplot: 'After the release of Jake Blues from prison, he and brother Elwood go to visit "The Penguin", the last of the nuns who raised them in',
    languages: [
      'English'
    ],
    released: 1980-06-20T00:00:00Z,
    directors: [
      'John Landis'
    ],
    writers: [
      'Dan Aykroyd'
    ]
  }
]
```

En este ejercicio la clave fue añadir los operadores **\$gte** ( $\geq$ ) y **\$lte** ( $\leq$ ), qué son operadores de comparación numérica.

### Exercici 2

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

Para ver un **mapa**, es recomendable crear un índice **geoespacial**, ya que permite buscar coordenadas de manera rápida incluso con muchos documentos y mejora la eficiencia. Además, garantiza que las consultas geoespaciales como **\$near** o **\$geoWithin** funcionen correctamente.

## 1.Creación de index:

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar is open, showing a connection to 'localhost:27017' with a database named 'Cinema'. Inside 'Cinema', there are collections: 'comments', 'movies', 'sessions', 'theaters', 'users', and system collections like 'admin', 'config', 'local', and 'test'. The main area is focused on the 'theaters' collection. At the top, tabs for 'Documents', 'Aggregations', 'Schema', 'Indexes', and 'Validation' are present. The 'Indexes' tab is selected, showing one existing index: '\_id\_ (REGULAR, 32.6 kB, 2 since Wed Dec 24 2025)'. Below this, a 'Create Index' dialog box is open, titled 'Create Index' for 'Cinema.theaters'. It contains two dropdowns under 'Index fields': 'location.geo' and '2dsphere'. There is also an 'Options' section. At the bottom of the dialog are 'Cancel' and 'Create Index' buttons.

## 2.Mapa de los theatres:

This screenshot shows the MongoDB Compass interface again, but this time the 'Schema' tab is selected for the 'theaters' collection. The left sidebar remains the same. The main area has a search bar at the top with the placeholder 'Type a query: { field: 'value' } or Generate query'. Below it, there's an 'EXPORT SCHEMA' button and a note stating 'This report is based on a sample of 1000 documents. Learn more'. The schema is displayed in three sections: 'document', 'geo' (with 'coordinates'), and 'theaterId'. The 'geo' section includes a map of North America with numerous blue dots representing theater locations. A legend at the bottom of the map indicates the size of the dots corresponds to the value in the 'theaterId' field. The 'theaterId' section shows a list of values: 1403, 196, 8022, 1526, 473, 1903, 892, 1437, 8809, 231, 948, 1477, 835, 1886, with their respective counts: 339, 617, 8162, 201, 6047, 444.

## NOTA:

He guardado todos los scripts de las consultas en un fichero llamado “**Script\_Sprint5.js**”, que voy a subir al repositorio de GitHub para tener una traza del trabajo realizado.

## TODOS LOS PASAJE DE CREACIÓN Y IMPORTACIÓN PUEDEN HACERSE TAMBIÉN CON CÓDIGO, A TRAVÉS DE POWERSHELL.

**He tenido que descargar “MongoDB Tools” y añadir el PATH para que me funcionen los comandos MongoDB en el PowerShell:**

```
$sysPath = [Environment]::GetEnvironmentVariable("Path", "Machine")
[Environment]::SetEnvironmentVariable("Path", $sysPath + ";C:\Program
Files\MongoDB\Tools\100\bin", "Machine")
```

**Ejemplo para importar collection con “mongoimport”:**

**1.paso, seleccionamos la ruta donde se encuentran los ficheros**

```
cd "C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5"
```

**2.paso, comando de importación de la colección “users”:**

```
mongoimport --db Cinema --collection users --file "users.json"
```

```
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> cd "C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5"
>> mongoimport --db Cinema --collection users --file "users.json"
2025-12-22T19:44:23.605+0100      connected to: mongodb://localhost/
2025-12-22T19:44:23.709+0100      185 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5>
```

**Nota:**

Si la base de datos “**Cinema**” y la colección “**users**” no existen se crean al momento de la importación, MongoDB es muy flexible en estas cosas.

**Script de todas las importación de colecciones completadas:**

```
Administrator:Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows
PS C:\Windows\system32> cd "C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5"
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> mongoimport --db Cinema --collection users --file "users.json"
2025-12-24T17:05:48.857+0100      connected to: mongodb://localhost/
2025-12-24T17:05:49.030+0100      185 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> mongoimport --db Cinema --collection comments --file "comments.json"
2025-12-24T17:07:19.656+0100      50304 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> mongoimport --db Cinema --collection sessions --file "sessions.json"
2025-12-24T17:08:21.487+0100      connected to: mongodb://localhost/
2025-12-24T17:08:21.574+0100      1 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> mongoimport --db Cinema --collection movies --file "movies.json"
2025-12-24T17:08:59.990+0100      connected to: mongodb://localhost/
2025-12-24T17:08:59.990+0100      23539 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5> mongoimport --db Cinema --collection theaters --file "theaters.json"
2025-12-24T17:19:27.820+0100      connected to: mongodb://localhost/
2025-12-24T17:19:27.912+0100      1564 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\eirad\Desktop\IT ACADEMY\NoSql\BaseDeDatos\SPRINT5>
```

## TENEMOS VARIOS FORMATO DE FICHEROS QUE PODEMOS IMPORTAR EN MONGODB.

**Lista de los más comunes y sus sintaxis básicas:**

### 1. NDJSON (Newline Delimited JSON)

```
mongoimport --db --collection --file datos.ndjson
```

### 2. JSON Array

```
mongoimport --db --collection --file datos.json --jsonArray
```

### 3. CSV (Comma Separated Values)

```
mongoimport --db --collection --type csv --file datos.csv --headerline
```

### 4. TSV (Tab Separated Values)

```
mongoimport --db --collection --type tsv --file datos.tsv --headerline
```

### Commentario:

Aunque mis archivos tienen extensión “.json”, en realidad están en formato **NDJSON** (un documento por línea). Por eso puedo importarlos sin usar “--jsonArray”.

## TENEMOS OTRA OPCIÓN PARA CREAR INDEX , ESTA VEZ A TRAVÉS DE MONGOSH.

**Código para crear index:**



```
>_MONGOOSH
> use Cinema
< switched to db Cinema
> db.theaters.createIndex({"location.geo":"2dsphere"})
< location.geo_2dsphere
Cinema >
```