

## SPRINT 9

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 la 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 que te pide el/la cliente/a, quien está midiendo si serás capaz o no de hacerte cargo de la parte analítica del proyecto vinculado con su base de datos.

## NIVEL 1

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

### Ejercicio 1

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

```
> use Cinema
< switched to db Cinema
> db.comments.find().limit(2)
< {
  _id: ObjectId('5a9427648b0beeb69579cc'),
  name: 'Andrea Le',
  email: 'andrea_le@fakegmail.com',
  movie_id: ObjectId('573a1390f29313caabcd418c'),
  text: 'Rem officiiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',
  date: 2012-03-26T23:20:16.000Z
}
{
  _id: ObjectId('5a9427648b0beeb69579cf'),
  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 vero animi.',
  date: 1987-02-10T00:29:36.000Z
}
```

¿Cuántos usuarios tenemos registrados?

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

¿Cuántos cines hay en el estado de California?

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

¿Quién fue 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/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu'  
}
```

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

```
> db.movies.countDocuments({'type':'movie','genres':'Comedy'})  
< 7002
```

## Ejercicio 2

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

```
> db.movies.find({'type':'movie','year':1932,  
  $or:[{'languages':'French'},{'genres':'Drama'}]})  
< {  
  _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 tries to wipe it away with his hand, but when he',  
  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/MV5BYWY3ODE5ZWVjYmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU0._V1_SV1000_SX677_AL_.jpg',  
  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'  
  ],  
}
```

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

Lo contamos para que quede más claro el número de datos con el que trabajamos.

## Ejercicio 3

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

```
db.movies.find({
  'type':'movie',
  '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 a tedious life, not going anywhere but from t
  imdb: {
    rating: 7.4,
    votes: 211238,
    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 turns into an adventure more extraordinary than
  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'
  ]
}
```

```
> db.movies.find({
  'type':'movie',
  'countries':'USA',
  'awards.wins':{$gte: 5, $lte: 9},
  'year':{ $gte:2012 , $lte: 2014}}).count()
< 162
```

Lo contamos para que quede más claro el número de datos con el que trabajamos.

## NIVEL 2

### Ejercicio 1

Cuenta cuantos comentarios escribe un usuario/a que utiliza “GAMEOFTHRON.ES” como dominio de su correo electrónico.

```
> db.comments.countDocuments({email:/GAMEOFTHRON.ES/i})
< 22841
```

La /i es para que no distinga entre mayúsculas y minúsculas.

## Ejercicio 2

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

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

Se usa “num\_cines: {\$sum: 1 }” para contar el número de códigos postales de cada grupo, en este caso aparecen 3 cines.

## NIVEL 3

### Ejercicio 1

Encuentra todas las películas dirigidas por John Landis con una puntuación IMDb entre 7,5 y 8.

```
> db.movies.find({
  'directors': 'John Landis',
  'type': 'movie',
  '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 who are so sanctimonious no one can stand them",
  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 plans for him.",
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ]
}
```

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

Lo contamos para que quede más claro el número de datos con el que trabajamos.

## Ejercicio 2

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

Para conseguir el mapa tenemos que hacer lo siguiente:

-Seleccionamos la colección “theaters” y marcamos la opción “Schema” del menú superior de MongoDB:

Documents 1.6K Aggregations Schema Indexes 1 Validation

-Nos aparecerá el listado del documento, allí tenemos que seleccionar el apartado “Geo” que es dónde se encuentran las coordenadas:

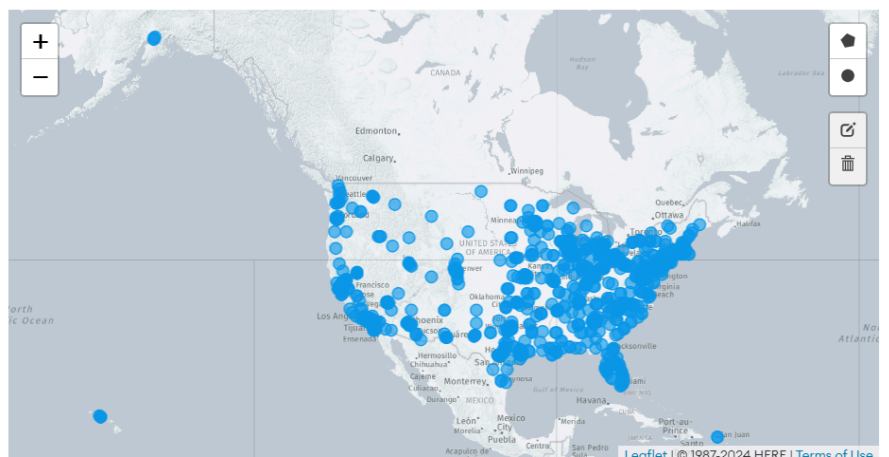
▼ geo  
coordinates

-Finalmente, se despliega el apartado coordiantes, donde aparece el mapa de todos los cines.

This report is based on a sample of 1000 documents. [Learn more](#)

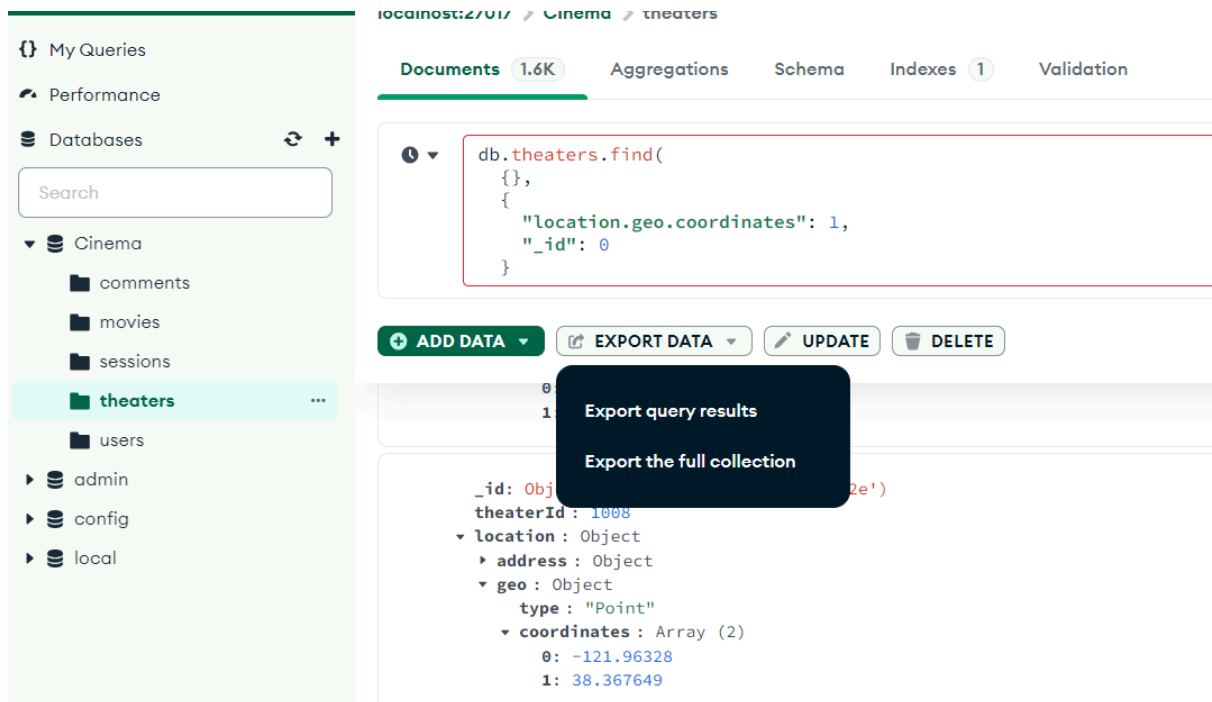
coordinates

coordinates

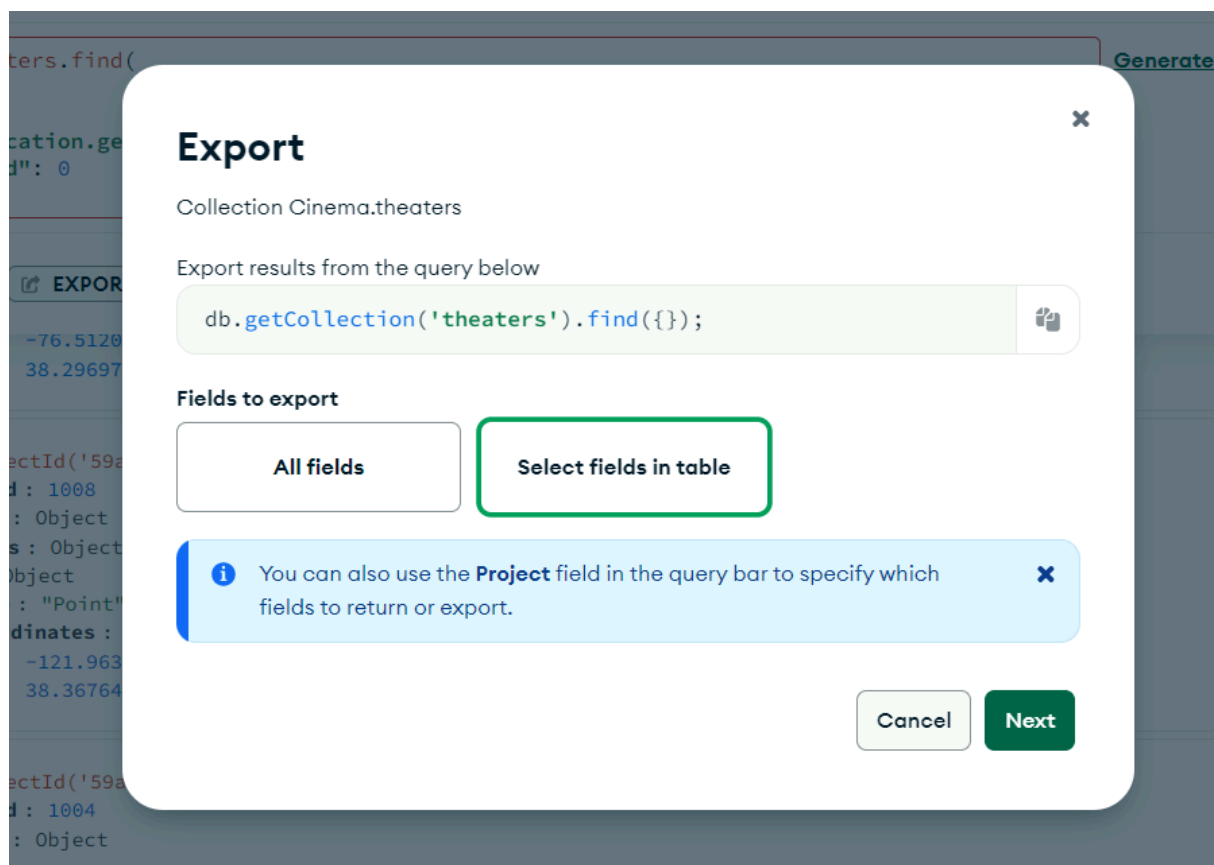


Otra manera de conseguir el mapa es con la ayuda de la herramienta de Google “My Maps”:

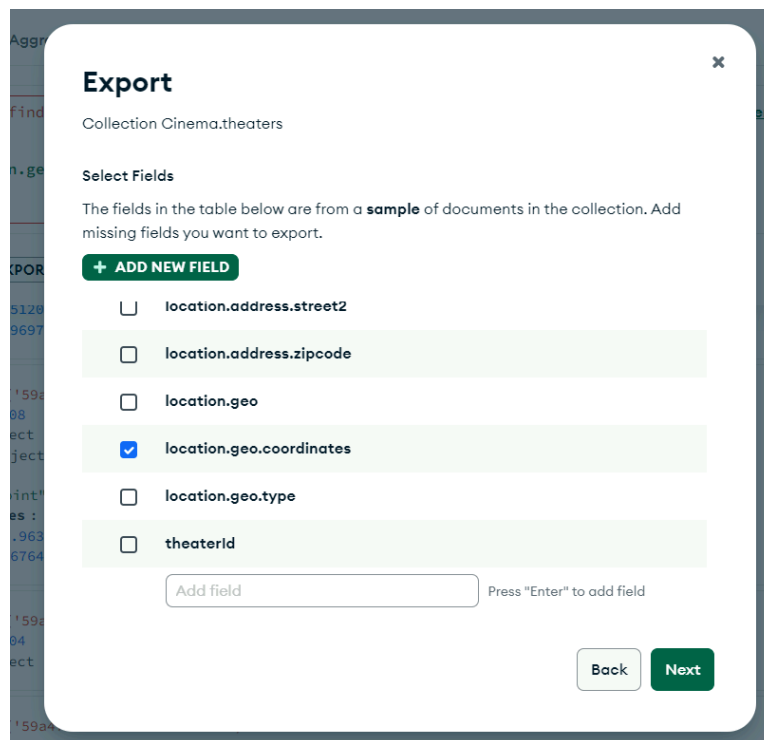
- Seleccionamos nuestra colección “theaters” y en el desplegable “EXPORT DATA”, seleccionamos la opción “Export query results”



-Seguidamente, seleccionamos la opción “Select fields in table” y le damos a Next



-Seleccionamos el apartado dónde se encuentran las coordenadas y le damos a “Next”.



The screenshot shows the 'Export' dialog for the 'Collection Cinema.theaters'. Under 'Select Fields', a list of fields is provided with checkboxes. The field 'location.geo.coordinates' is selected with a blue checkmark. Other fields include 'location.address.street2', 'location.address.zipcode', 'location.geo', 'location.geo.type', and 'theaterId'. At the bottom, there is an 'Add field' input box and 'Back' and 'Next' buttons.

**Export**

Collection Cinema.theaters

Select Fields

The fields in the table below are from a **sample** of documents in the collection. Add missing fields you want to export.

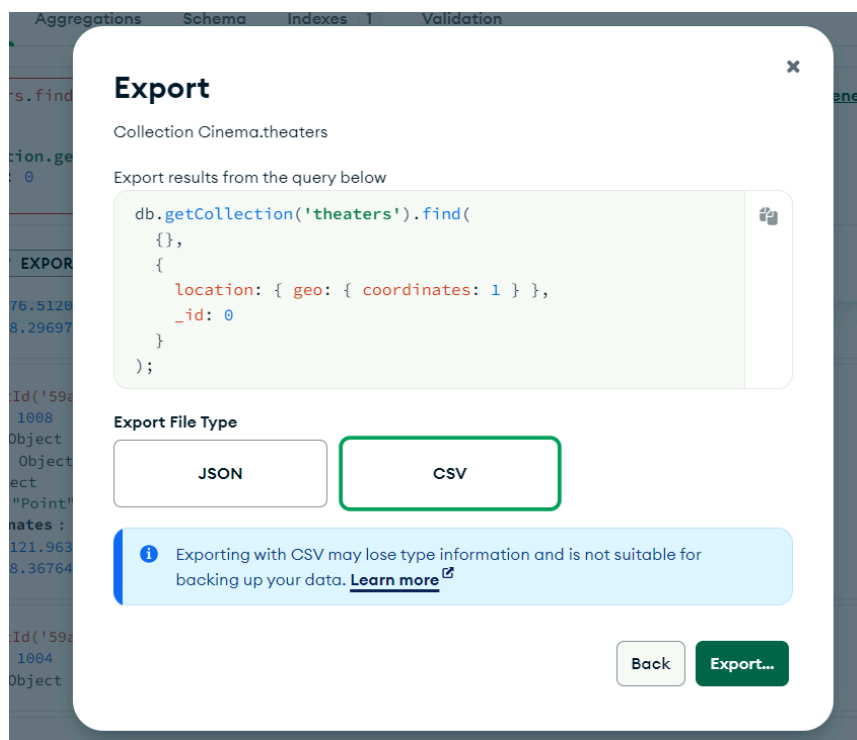
**+ ADD NEW FIELD**

- ☐ location.address.street2
- ☐ location.address.zipcode
- ☐ location.geo
- ☒ location.geo.coordinates
- ☐ location.geo.type
- ☐ theaterId

Add field  Press "Enter" to add field

**Back** **Next**

-Aparecerá la query realizada y simplemente tenemos que exportar en CSV.



The screenshot shows the 'Export' dialog for the 'Collection Cinema.theaters'. It displays the MongoDB query used for the export. Below the query, there are two buttons for 'Export File Type': 'JSON' and 'CSV'. The 'CSV' button is highlighted with a green border. A blue information box at the bottom states: 'Exporting with CSV may lose type information and is not suitable for backing up your data. [Learn more](#)'. At the bottom right, there are 'Back' and 'Export...' buttons.

**Export**

Collection Cinema.theaters

Export results from the query below

```
db.getCollection('theaters').find(
  {},
  {
    location: { geo: { coordinates: 1 } },
    _id: 0
  }
);
```

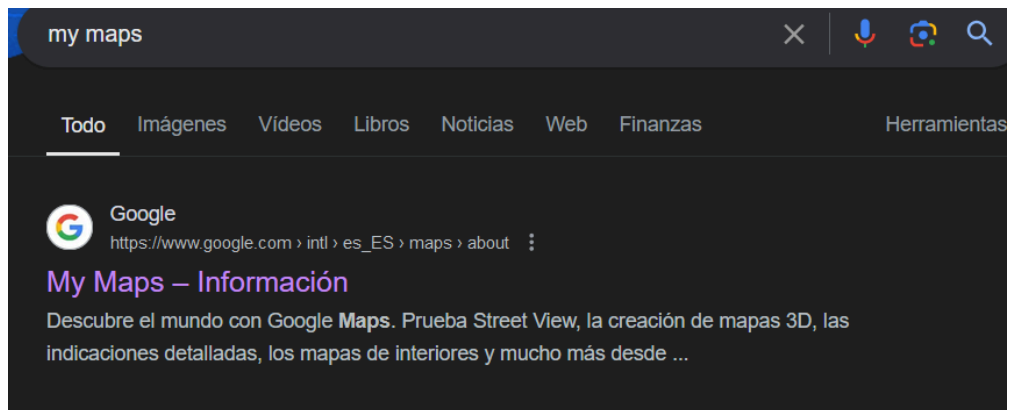
**Export File Type**

**JSON** **CSV**

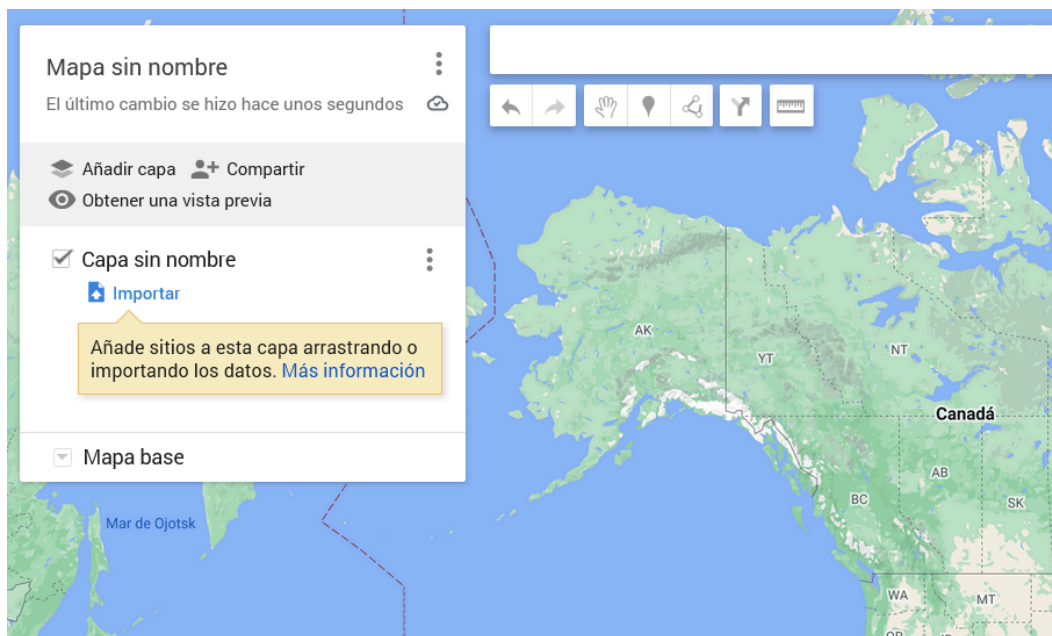
**i** Exporting with CSV may lose type information and is not suitable for backing up your data. [Learn more](#)

**Back** **Export...**

-Vamos a My Maps en google.



-Le damos a importar y arrastramos nuestro CSV.



-Finalmente nos aparece nuestro mapa de cines.

