

# Bases de Datos 2025

## Recuperatorio II: NoSQL

Sergio Canchi, Agustín Curto, Ramiro Demasi,  
Diego Gimenez, Guido Ivetta, Iván Nieva

### Contexto

La base de datos **mflix** contiene datos sobre películas y salas de cine. Adicionalmente incluye datos de usuarios y comentarios sobre las películas. A continuación se describen todas las colecciones de la base de datos **mflix**.

Colección	Descripción
movies	Contiene información de las películas, incluyendo título, año de estreno, y director.
theaters	Contiene ubicaciones de salas de cine.
users	Contiene información del usuario.
comments	Contiene comentarios asociados a películas específicas
sessions	Campo de metadatos. Contiene JSON Web Token de los usuarios.

Para cargar la base de datos **mflix** ejecutar los siguientes comandos:

**Instructivo para los que usan sus computadoras personales:**

```
$ tar xzvf recuperatorio2.tar.gz  
$ cd recuperatorio2/  
$ mongorestore --host <host> --drop --gzip --db mflix mflix/
```

**Instructivo para los que usan las computadoras del laboratorio:**

El primer paso es obtener la imagen de mongo 8. Para ello, correr el siguiente comando:

```
$ docker pull mongo:8
```

**En caso de tener problemas**, descargar el archivo mongo8.tar.gz y usar el comando:

```
$ docker load -i mongo8.tar.gz
```

Ahora que ya tenemos la imagen, podemos levantar el contenedor y cargar los datos:

```
$ tar xzvf recuperatorio2.tar.gz
$ cd recuperatorio2/
$ docker run --name mongo-recuperatorio2 \
    -p 27017:27017 -v mongod.conf:/etc/mongod.conf \
    -v ./mflix:/data/mflix -d mongo:8
$ docker exec -it mongo-recuperatorio2 mongorestore \
    --drop --gzip --db mflix /data/mflix/
// (Opcional) para ejecutar el cliente
$ docker exec -it mongo-recuperatorio2 mongosh
```

## Consignas

1. Estás desarrollando la funcionalidad de búsqueda para una aplicación de películas. Un usuario escribe "Madagascar" en la barra de búsqueda y la aplicación debe mostrar todos los comentarios asociados a esa película específica. Escribe una consulta de agregación (un pipeline) que muestre todos los comentarios para esta película. El resultado debe mostrar solamente el título de la película y el texto del comentario.

Estructura de cada documento del output:

```
{ "tituloPelicula": "Madagascar", "textoComentario": "Wow, great movie!" }
{ "tituloPelicula": "Madagascar", "textoComentario": "It was amazing." }
...
```

2. El equipo de curación de contenido ha decidido revalorizar las películas clásicas. En la colección movies, todas las películas estrenadas antes de 1975 deben ser actualizadas de dos maneras:
  - a. Se debe añadir el string "Classic" a su arreglo genres (sin duplicados).
  - b. Para reflejar su valor histórico, su calificación imdb.rating debe ser "inflada" en un 10% (es decir, multiplicada por 1.1).

**NOTA:** no todas las películas tienen un imdb.rating válido. El campo puede no existir, ser null o, en algunos casos, ser un string vacío (""). La consulta debe filtrar los documentos para asegurar que la actualización se aplique únicamente a las películas que tengan un imdb.rating que sea un número válido para el cálculo.

**NOTA 2:** no importa si al "inflar" la nota queda en más de 10

3. Crear una vista llamada vista\_directores\_prolificos. El propósito de esta vista es identificar a los 10 directores con la mayor cantidad de películas registradas en la

colección y analizar su performance promedio. Cada documento de salida debe contener:

- a. El nombre del director
- b. La cantidad total de películas que ha dirigido (campo: `cantidad_peliculas`).
- c. El rating promedio de todas sus películas, basado en el campo `imdb.rating` (campo: `rating_promedio`).

Los resultados deben estar ordenados de forma descendente según `cantidad_peliculas`.

4. Especificar reglas de validación en la colección `movies`. Los requerimientos a validar son:
  - a. El `runtime` debe ser mayor que cero.
  - b. El `type` debe ser sólo de alguno de estos dos tipos: [ 'movie', 'series' ]
  - c. Dos reglas que ustedes propongan, explicar por qué lo eligieron.

## Puntos a tener en cuenta

- Algunas consultas pueden resolverse con `find` y `aggregate`. En estos casos se puede resolver eligiendo alguno de estos métodos.
- Resolver las consultas sin vistas salvo que se lo pida explícitamente.
- Mostrar únicamente los campos pedidos en la consigna.
- Se piden que los campos que se devuelven sean valores escalares a menos que se pida los valores de los campos devueltos podrán ser documentos anidados, arreglos de escalares o arreglos de documentos.
- Buscar hacer la consulta de la forma más sencilla y eficiente posible.
- Se evaluará el correcto formato de las soluciones:
  - El código entregado debe ser legible.
  - Utilizar indentación de espacios de manera uniforme.

### Detalle Importante

El alumno puede usar sus notas personales e internet, pero queda prohibido utilizar modelos generativos (Chat-GPT, LLAMA, etc) y comunicarse con otras personas oral o digitalmente.

Romper alguna de estas dos restricciones invalida el examen

## Entrega

- Se entregará un archivo comprimido `soluciones.tar.gz` o `soluciones.zip` (con `soluciones.js` adentro) con las soluciones de los 5 ejercicios. Separar las soluciones mediante comentarios de Javascript /\* \*/ o //.
- La entrega se hará mediante el [Aula Virtual](#) en el [correspondiente apartado](#).

- Tendrán hasta las 17:00 para que se considere una entrega completa. La recomendación es empezar a subir el archivo a las 16:50 para evitar cualquier eventualidad.