

# Bases de Datos 2023

## Práctico VIII: Pipeline de Agregación en MongoDB

Juan Cabral, Tadeo Cocucci,  
Ramiro Demasi, Guido Ivetta

Continuando con la bases de datos **mflix** y **restaurantdb** realizar las siguientes consultas usando el pipeline de agregación:

1. Cantidad de cines (*theaters*) por estado.
2. Cantidad de estados con al menos dos cines (*theaters*) registrados.
3. Cantidad de películas dirigidas por "Louis Lumière". Se puede responder sin pipeline de agregación, realizar ambas queries.
4. Cantidad de películas estrenadas en los años 50 (desde 1950 hasta 1959). Se puede responder sin pipeline de agregación, realizar ambas queries.
5. Listar los 10 géneros con mayor cantidad de películas (tener en cuenta que las películas pueden tener más de un género). Devolver el género y la cantidad de películas. Hint: [unwind](#) puede ser de utilidad
6. Top 10 de usuarios con mayor cantidad de comentarios, mostrando Nombre, Email y Cantidad de Comentarios.
7. Ratings de IMDB promedio, mínimo y máximo por año de las películas estrenadas en los años 80 (desde 1980 hasta 1989), ordenados de mayor a menor por promedio del año.
8. Título, año y cantidad de comentarios de las 10 películas con más comentarios.
9. Crear una vista con los 5 géneros con mayor cantidad de comentarios, junto con la cantidad de comentarios.
10. Listar los actores (*cast*) que trabajaron en 2 o más películas dirigidas por "Jules Bass". Devolver el nombre de estos actores junto con la lista de películas (solo título y año) dirigidas por "Jules Bass" en las que trabajaron.
  - a. *Hint1*: [addToSet](#)
  - b. *Hint2*: `{'name.2': {$exists: true}}` permite filtrar arrays con al menos 2 elementos, entender por qué.
  - c. *Hint3*: Puede que tu solución no use *Hint1* ni *Hint2* e igualmente sea correcta
11. Listar los usuarios que realizaron comentarios durante el mismo mes de lanzamiento de la película comentada, mostrando Nombre, Email, fecha del comentario, título de la película, fecha de lanzamiento. HINT: usar \$lookup con multiple condiciones
12. Listar el id y nombre de los restaurantes junto con su puntuación máxima, mínima y la suma total. Se puede asumir que el restaurant\_id es único.
  - a. Resolver con \$group y accumulators.
  - b. Resolver con expresiones sobre arreglos (por ejemplo, \$sum) pero sin \$group.

- c. Resolver como en el punto b) pero usar \$reduce para calcular la puntuación total.
  - d. Resolver con find.
13. Actualizar los datos de los restaurantes añadiendo dos campos nuevos.
- a. "average\_score": con la puntuación promedio
  - b. "grade": con "A" si "average\_score" está entre 0 y 13,  
con "B" si "average\_score" está entre 14 y 27  
con "C" si "average\_score" es mayor o igual a 28
- Se debe actualizar con una sola query.
- a. HINT1. Se puede usar pipeline de agregación con la operación update
  - b. HINT2. El operador \$switch o \$cond pueden ser de ayuda.