



Evaluación

Evaluación

Sobre un Dataset de datos dado -> [Movies](#), realizar:

- 1 Ejercicio de Cargar / Importar dataset
- 23 Ejercicios sobre inserción, actualización, proyección, filtrado y el pipeline de Agregación
- 3 Ejercicios Libres sobre el pipeline de Agregación
- Entregar fichero PDF con las queries de TODOS los ejercicios mostrando una captura de pantalla con los resultados



Ejercicios

- 0. Realizar la importación del json en una colección llamada "movies".
- 1. Analizar con find la colección. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : ObjectId("5dc41d848cc90c00a2e25b"),  
  "title" : "XXXX",  
  "year" : 1900,  
  "cast" : [],  
  "genres" : []  
}
```

- 2. Contar cuántos documentos (películas) tiene cargado. Se muestra una salida de ejemplo, el resultado es erróneo.

```
100000
```

- 3. Insertar una película. Se muestra una salida de ejemplo, el resultado es erróneo.

```
WriteResult({ "nInserted" : 1 })
```

- 4. Borrar la película insertada en el punto anterior (en el 3). Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "acknowledged" : true,  
  "deletedCount" : 1  
}
```



Ejercicios

- 5. Contar cuantas películas tienen actores (cast) que se llaman "and". Estos nombres de actores están por ERROR. Se muestra una salida de ejemplo, el resultado es erróneo.

500

- 6. Actualizar los documentos cuyo actor (cast) tenga por error el valor "and" como si realmente fuera un actor. Para ello, se debe sacar únicamente ese valor del array cast. Por lo tanto, no se debe eliminar ni el documento (película) ni su array cast con el resto de actores. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{
```

```
  "acknowledged" : true,
```

```
  "matchedCount" : 500,
```

```
  "modifiedCount" : 500
```

```
}
```

Antes

Después

cast: ["Tom", "and", "Jerry"] -> cast: ["Tom", "Jerry"]

- 7. Contar cuantos documentos (películas) tienen el array 'cast' vacío. Se muestra una salida de ejemplo, el resultado es erróneo.

1000



Ejercicios

- 8. Actualizar TODOS los documentos (películas) que tengan el array cast vacío, añadiendo un nuevo elemento dentro del array con valor Undefined. Cuidado! El tipo de cast debe seguir siendo un array. El array debe ser así -> ["Undefined"]. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "acknowledged" : true,  
  "matchedCount" : 1000,  
  "modifiedCount" : 1000  
}
```

Ejemplo de cómo debería quedar el campo "cast"

```
{  
  "_id" : ObjectId("5dc41d848cc90c0c00a2e25b"),  
  "title" : "XXXX",  
  "year" : 1900,  
  "cast" : ["Undefined"],  
  "genres" : []  
}
```



Ejercicios

- 9. Contar cuantos documentos (películas) tienen el array genres vacío. Se muestra una salida de ejemplo, el resultado es erróneo.

```
1000
```

- 10. Actualizar TODOS los documentos (películas) que tengan el array genres vacío, añadiendo un nuevo elemento dentro del array con valor Undefined. Cuidado! El tipo de genres debe seguir siendo un array. El array debe ser así -> ["Undefined"]. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "acknowledged" : true,  
  "matchedCount" : 1000,  
  "modifiedCount" : 1000  
}
```

Ejemplo de cómo debería quedar el campo "genres"

```
{  
  "_id" : ObjectId("5dc41d848cc90c0c00a2e25b"),  
  "title" : "XXXX",  
  "year" : 1900,  
  "cast" : ["Undefined"],  
  "genres" : ["Undefined"]  
}
```



Ejercicios

- 11. Mostrar el año más reciente / actual que tenemos sobre todas las películas. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "year" : 3333  
}
```

- 12. Contar cuántas películas han salido en los últimos 20 años. Debe hacerse desde el último año que se tienen registradas películas en la colección, mostrando el resultado total de esos años. Se debe hacer con el Framework de Agregación. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : null,  
  "total" : 9999  
}
```

- 13. Contar cuántas películas han salido en la década de los 60 (del 60 al 69 incluidos). Se debe hacer con el Framework de Agregación. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : null,  
  "total" : 9999  
}
```



Ejercicios

- 14. Mostrar el año u años con más películas mostrando el número de películas de ese año. Revisar si varios años pueden compartir tener el mayor número de películas. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : 2015,  
  "pelis" : 1000  
}
```

- 15. Mostrar el año u años con menos películas mostrando el número de películas de ese año. Revisar si varios años pueden compartir tener el menor número de películas. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : 2015,  
  "pelis" : 1  
}
```

- 16. Guardar en nueva colección llamada "actors" realizando la fase \$unwind por actor. Después, contar cuantos documentos existen en la nueva colección. Se muestra una salida de ejemplo, el resultado es erróneo.

```
100000
```



Ejercicios

- 17. Sobre actors (nueva colección), mostrar la lista con los 5 actores que han participado en más películas mostrando el número de películas en las que ha participado. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : "Pepito Palotes",  
  "cuenta" : 500  
}
```

- 18. Sobre actors (nueva colección), agrupar por película y año mostrando las 5 en las que más actores hayan participado, mostrando el número total de actores. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : {  
    "title" : "The Four Musketeers",  
    "year" : 2000  
  },  
  "cuenta" : 50  
}
```



Ejercicios

- 19. Sobre actors (nueva colección), mostrar los 5 actores cuya carrera haya sido la más larga. Para ello, se debe mostrar cuándo comenzó su carrera, cuándo finalizó y cuántos años ha trabajado. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{  
  "_id" : "Ford Scott",  
  "comienza" : 1939,  
  "termina" : 2007,  
  "anos" : 68  
}
```

- 20. Sobre actors (nueva colección), Guardar en nueva colección llamada "genres" realizando la fase \$unwind por genres. Después, contar cuantos documentos existen en la nueva colección. Se muestra una salida de ejemplo, el resultado es erróneo.

```
200000
```



Ejercicios

- 21. Sobre genres (nueva colección), mostrar los 5 documentos agrupados por "Año y Género" que más número de películas **diferentes** tienen mostrando el número total de películas. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{
  "_id" : {
    "year" : 1999,
    "genre" : "Unico"
  },
  "pelis" : 400
}
```



Ejercicios

- 22. Sobre genres (nueva colección), mostrar los 5 actores y los géneros en los que han participado con más número de géneros **diferentes**, se debe mostrar el número de géneros diferentes que ha interpretado. Importante! Se necesita previamente filtrar para descartar aquellos actores llamados "Undefined". Aclarar que no se eliminan de la colección, sólo que filtramos para que no aparezcan. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{
  "_id" : "Camilo Jose",
  "numgeneros" : 5,
  "generos" : [
    "Science Fiction",
    "Drama",
    "Crime",
    "Horror",
    "Dance"
  ]
}
```



Ejercicios

- 23. Sobre genres (nueva colección), mostrar las 5 películas y su año correspondiente en los que más géneros **diferentes** han sido catalogados, mostrando esos géneros y el número de géneros que contiene. Se muestra una salida de ejemplo, el resultado es erróneo.

```
{
  "_id" : {
    "title" : "American Humans",
    "year" : 2001
  },
  "numgeneros" : 3,
  "generos" : [
    "Action",
    "Biography",
    "Crime"
  ]
}
```

- 24. Query libre sobre el pipeline de agregación.
- 25. Query libre sobre el pipeline de agregación.
- 26. Query libre sobre el pipeline de agregación



