

## Marco de trabajo de agregación

MongoDB es una base de datos NoSQL que proporciona una gran flexibilidad para trabajar con datos en formatos variados. Uno de los aspectos más poderosos de MongoDB es su marco de trabajo de agregación, que permite realizar consultas complejas y transformaciones de datos. **El marco de trabajo de agregación en MongoDB se basa en el concepto de "pipelines" y "etapas".**

### Pipeline de Agregación

Un pipeline de agregación es una secuencia de etapas de procesamiento de datos. Cada etapa transforma los documentos de alguna manera (filtrado, agrupación, proyección, etc.) y pasa su salida a la siguiente etapa en la secuencia.

### Etapas Comunes de Agregación

**\$match:** Filtra los documentos para pasar solo aquellos que coinciden con las condiciones dadas.

1. **\$group:** Agrupa los documentos por alguna clave especificada y permite realizar operaciones en los datos agrupados.
2. **\$sort:** Ordena los documentos.
3. **\$project:** Se utiliza para seleccionar o excluir campos específicos de los documentos.
4. **\$limit:** Limita el número de documentos a pasar a la siguiente etapa.
5. **\$skip:** Salta un número específico de documentos.
6. **\$unwind:** Descompone un campo de tipo array de los documentos para que cada elemento del array se trate como un documento separado.
7. **\$addFields / \$set:** Agrega nuevos campos a los documentos.
8. **\$lookup:** Realiza una especie de "join" con otra colección.

## Ejemplos de Pipeline de Agregación

### Ejemplo 1: Pipeline con 5 Etapas

Supongamos que queremos analizar las películas por género, calificación y país, con el siguiente pipeline:

- **\$match:** Seleccionamos películas después del año 2000.
- **\$unwind:** Descompone el campo genres.
- **\$group:** Agrupa por género y calcula la calificación media.
- **\$sort:** Ordena los géneros por calificación media descendente.
- **\$limit:** Limitamos a 10 los géneros a mostrar.

```
db.movieDetails.aggregate([
  { $match: { year: { $gt: 2000 } } },
  { $unwind: "$genres" },
  { $group: { _id: "$genres", averageRating: { $avg: "$imdb.rating" } } },
],
  { $sort: { averageRating: -1 } },
  { $limit: 10 }
]);
```

### Ejemplo 2: Pipeline con 3 Etapas

Ahora, digamos que queremos encontrar películas con cierto número de premios y que sean de un país específico:

- **\$match:** Seleccionamos películas con más de 5 premios.
- **\$unwind:** Descompone el campo countries.
- **\$match:** Seleccionamos películas del país 'USA'.

```
db.movieDetails.aggregate([
  { $match: { "awards.wins": { $gt: 5 } } },
  { $unwind: "$countries" },
  { $match: { countries: "USA" } }
]);
```

### Ejemplo 3: Análisis de Directores y Géneros

Encontrar los directores más populares en géneros específicos basándonos en la cantidad de películas dirigidas y la calificación promedio.

- **\$match:** Filtrar películas de ciertos géneros.
- **\$group:** Agrupar por director, contar películas y calcular calificación promedio.
- **\$sort:** Ordenar por número de películas dirigidas.
- **\$limit:** Limitar a los 5 directores principales.

```
db.movieDetails.aggregate([
  { $match: { genres: { $in: ["Action", "Drama"] } } },
  { $group: { _id: "$director", totalMovies: { $sum: 1 },
averageRating: { $avg: "$imdb.rating" } } },
  { $sort: { totalMovies: -1 } },
  { $limit: 5 }
]);
```

### Ejemplo 4: Tendencias en Duración de Películas

Analizar la tendencia en la duración de las películas a lo largo de los años.

- **\$project:** Seleccionar solo los campos necesarios (año y duración).
- **\$group:** Agrupar por año y calcular duración promedio.
- **\$sort:** Ordenar por año

```
db.movieDetails.aggregate([
  { $project: { year: 1, runtime: 1 } },
  { $group: { _id: "$year", averageRuntime: { $avg: "$runtime" } } },
  { $sort: { _id: 1 } }
]);
db.movieDetails.aggregate([
  { $project: { year: 1, runtime: 1 } },
  { $group: { _id: "$year", averageRuntime: { $avg: "$runtime" } } },
  { $sort: { _id: 1 } },
  { $project: { year: "$_id", averageRuntime: { $trunc:
["$averageRuntime", 2] } } }
]);
```

### Ejemplo 5: Clasificación de Películas por País y Género

Contar el número de películas por país y género.

- **\$unwind:** Descomponer el campo countries.
- **\$unwind:** Descomponer el campo genres.
- **\$group:** Agrupar por país y género, contar películas.
- **\$sort:** Ordenar por país y género.

```
db.movies.aggregate([
  { $unwind: "$countries" },
  { $unwind: "$genres" },
  { $group: { _id: { country: "$countries", genre: "$genres" }, count:
{ $sum: 1 } } },
  { $sort: { "_id.country": 1, "_id.genre": 1 } }
]);
```

### Ejemplo 6: Análisis de Actores y su Participación en Géneros

Identificar actores que han participado en una diversidad de géneros.

- **\$unwind:** Descomponer el campo actors.
- **\$unwind:** Descomponer el campo genres.
- **\$group:** Agrupar por actor y contar los géneros únicos.
- **\$match:** Seleccionar actores que han estado en más de 3 géneros diferentes.
- **\$sort:** Ordenar por número de géneros.

```
db.movies.aggregate([
  { $unwind: "$actors" },
  { $unwind: "$genres" },
  { $group: { _id: "$actors", uniqueGenres: { $addToSet: "$genres" } }
},
  { $project: { actor: "$_id", genresCount: { $size: "$uniqueGenres" }
} },
  { $match: { genresCount: { $gt: 3 } } },
  { $sort: { genresCount: -1 } }
]);
```