# Índice

[**Índice 1**](#_qgj57awqcchu)

[**Pipelines de agregación 2**](#_bt737br1ttdk)

[Ejemplo 2](#_e0d7g5sajdzn)

[Datos de muestra 2](#_4ra8s75f01k8)

[**Funciones 2**](#_bt737br1ttdk)

[Aggregation $group 2](#_cd6jsn1inago)

[Ejemplo 2](#_fyp9y125cllk)

[Aggregation $limit 2](#_4wmuei3ij8zi)

[Ejemplo 3](#_2uftzvsc68ax)

[Aggregation $project 3](#_4mhvoysdfft5)

[Ejemplo 3](#_6d4trx6ynbg3)

[Aggregation $sort 3](#_7fcty08nmsz0)

[Ejemplo 3](#_ih2ud97ixkn4)

[Aggregation $match 4](#_6pz3xp56ykk0)

[Ejemplo 4](#_tkrh9ff493bu)

[Aggregation $addFields 4](#_rzxvxmtxff97)

[Ejemplo 4](#_qq8w87x98ikb)

[Aggregation $count 5](#_5h8zx9sxb27w)

[Ejemplo 5](#_tbsltyju3x2a)

[Aggregation $lookup 5](#_v44e3fg2qdm8)

[Ejemplo 5](#_yqidrf2p145w)

[Aggregation $out 6](#_4uwy4zpe3skr)

[Ejemplo 6](#_mh098wpvfcat)

[**Bibliografía 7**](#_c7j00pcyt0sk)

# 

# Pipelines de agregación

Las operaciones de agregación le permiten agrupar, ordenar, realizar cálculos, analizar datos y mucho más.

Los canales o pipelines de agregación pueden tener una o más "etapas". El orden de estas etapas es importante. Cada etapa actúa sobre los resultados de la etapa anterior.

### Ejemplo

*db.posts.aggregate([*

*// Etapa 1: solo busque documentos que tengan más de 1 me gusta*

*{*

*$match: { me gusta: { $gt: 1 } }*

*},*

*// Etapa 2: Agrupar documentos por categoría y sumar los gustos de cada categoría*

*{*

*$grupo: { \_id: "$categoría", totalMe gusta: { $suma: "$me gusta" } }*

*}*

*])*

## Datos de muestra

Para demostrar el uso de etapas en un proceso de agregación, cargaremos datos de muestra en nuestra base de datos.

Desde el panel de [MongoDB Atlas](https://www.mongodb.com/docs/), vaya a Bases de datos. Haga clic en los puntos suspensivos y seleccione "Cargar conjunto de datos de muestra". Esto cargará varios conjuntos de datos de muestra en su base de datos.

En las siguientes secciones explicaremos varias etapas del proceso de agregación con más detalle utilizando estos datos de muestra.

# Funciones

## Aggregation $group

Esta etapa de agregación agrupa documentos por la expresión \_id única proporcionada.

No confundas esta expresión \_id con el ObjectId \_id proporcionado a cada documento.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_airbnb" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.listingsAndReviews.aggregate(*

*[ { $group : { \_id : "$property\_type" } } ]*

## Aggregation $limit

*Esta etapa de agregación limita la cantidad de documentos pasados a la siguiente etapa.*

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_mflix" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.movies.aggregate([ { $limit: 1 } ])*

## Aggregation $project

Esta etapa de agregación pasa solo los campos especificados a la siguiente etapa de agregación. Esta es la misma proyección que se usa con el método find().

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_restaurants" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.restaurants.aggregate([*

*{*

*$project: {*

*"name": 1,*

*"cuisine": 1,*

*"address": 1*

*}*

*},*

*{*

*$limit: 5*

*}*

*])*

## Aggregation $sort

Esta etapa de agregación ordena todos los documentos en el orden especificado.

Recuerda que el orden de tus etapas importa. Cada etapa solo actúa sobre los documentos que las etapas anteriores proporcionan.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_airbnb" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.listingsAndReviews.aggregate([*

*{*

*$sort: { "accommodates": -1 }*

*},*

*{*

*$project: {*

*"name": 1,*

*"accommodates": 1*

*}*

*},*

*{*

*$limit: 5*

*}*

*])*

## Aggregation $match

Esta etapa de agregación se comporta como un find. Filtra los documentos que coinciden con la consulta proporcionada.

Usar $match al inicio de la tubería puede mejorar el rendimiento ya que limita la cantidad de documentos que las siguientes etapas deben procesar.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_airbnb" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.listingsAndReviews.aggregate([*

*{ $match : { property\_type : "House" } },*

*{ $limit: 2 },*

*{ $project: {*

*"name": 1,*

*"bedrooms": 1,*

*"price": 1*

*}}*

*])*

## Aggregation $addFields

Esta etapa de agregación añade nuevos campos a los documentos.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_restaurants" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.restaurants.aggregate([*

*{*

*$addFields: {*

*avgGrade: { $avg: "$grades.score" }*

*}*

*},*

*{*

*$project: {*

*"name": 1,*

*"avgGrade": 1*

*}*

*},*

*{*

*$limit: 5*

*}*

*])*

## Aggregation $count

Esta etapa de agregación cuenta la cantidad total de documentos pasados desde la etapa anterior.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_restaurants" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.restaurants.aggregate([*

*{*

*$match: { "cuisine": "Chinese" }*

*},*

*{*

*$count: "totalChinese"*

*}*

*])*

## Aggregation $lookup

Esta etapa de agregación realiza un join externo izquierdo a una colección en la misma base de datos.

Hay cuatro campos obligatorios:

1. **from**: La colección a usar para la búsqueda en la misma base de datos.
2. **localField**: El campo en la colección principal que puede usarse como identificador único en la colección from.
3. **foreignField**: El campo en la colección from que puede usarse como identificador único en la colección principal.
4. **as**: El nombre del nuevo campo que contendrá los documentos coincidentes de la colección from.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_mflix" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

db.comments.aggregate([

{

$lookup: {

from: "movies",

localField: "movie\_id",

foreignField: "\_id",

as: "movie\_details",

},

},

{

$limit: 1

}

])

## Aggregation $out

Esta etapa de agregación escribe los documentos devueltos desde la tubería de agregación a una colección.

La etapa $out debe ser la última etapa de la tubería de agregación.

### Ejemplo

En este ejemplo, estamos usando la base de datos "sample\_airbnb" cargada de nuestros datos de muestra en la sección de Introducción a Agregaciones.

*db.listingsAndReviews.aggregate([*

*{*

*$group: {*

*\_id: "$property\_type",*

*properties: {*

*$push: {*

*name: "$name",*

*accommodates: "$accommodates",*

*price: "$price",*

*},*

*},*

*},*

*},*

*{ $out: "properties\_by\_type" },*

*])*

# Bibliografía

<https://www.w3schools.com/mongodb/index.php>