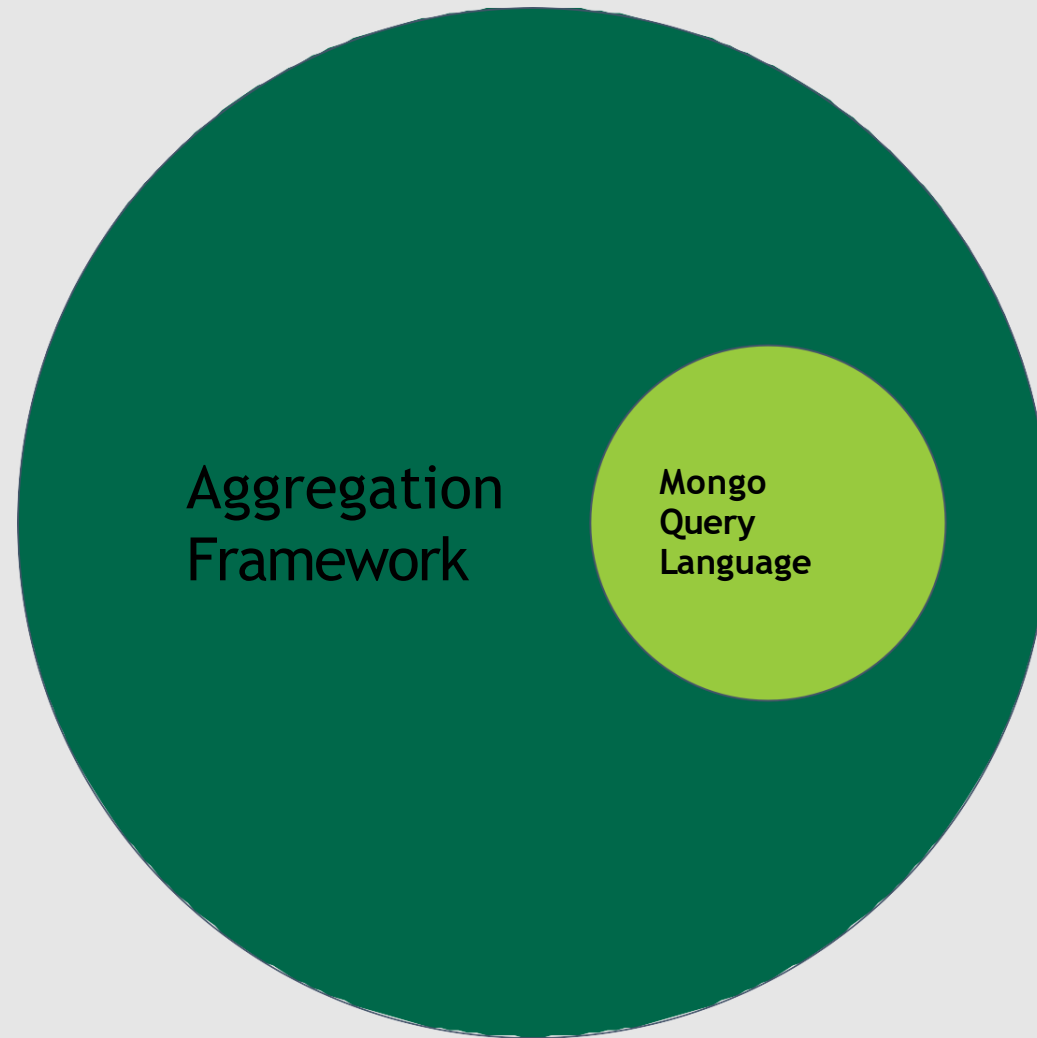
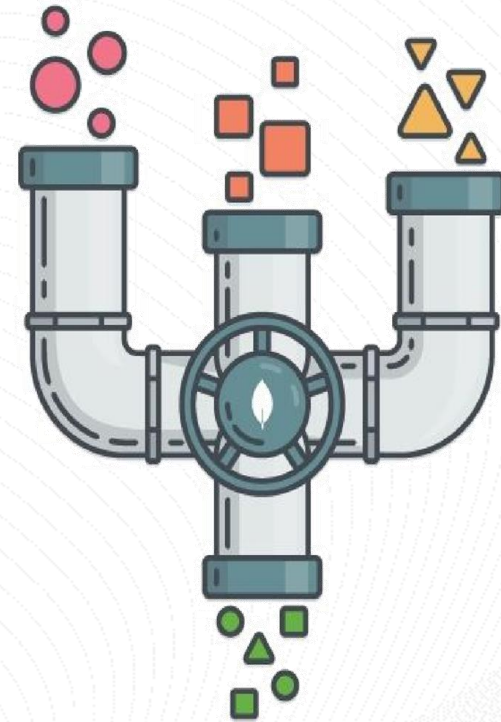




Aggregation Pipelines



# Pipelines: Tuberías de datos





# Pipelines: Tuberías de datos



## Documentación

<https://www.mongodb.com/docs/manual/aggregation/>

# Ejemplo: find() = aggregate()



Currently connected to Mongo Atlas. Click here to change connection.

```
use("sample_airbnb")
```

```
// db.listingsAndReviews.find({  
//   amenities: "Wifi"  
// }, {  
//   price: 1,  
//   amenities: 1  
// })
```

```
// [], [], []
```

```
db.listingsAndReviews.aggregate([  
  { $match: { amenities: "Wifi" } }, // find  
  { $project: { price: 1, amenities: 1 } }, // project  
)
```

# Ejemplo: find() = aggregate()



Currently connected to Mongo Atlas. Click here to change connection.

```
use("sample_airbnb")
```

```
// db.listingsAndReviews.find({  
//   amenities: "Wifi"  
// }, {  
//   price: 1,  
//   amenities: 1  
// })
```

```
// [], [], []
```

```
db.listingsAndReviews.aggregate([  
  { $match: { amenities: "Wifi" } }, // find  
  { $project: { address: 1 } }, // project  
  { $group: { _id: "$address.country", count: { $sum: 1 } } }  
])
```





Aggregation \$match

# \$match



- Esta etapa de agregación se comporta como un find(). Filtrará los documentos que coincidan con la consulta proporcionada.
- El uso temprano \$match en el pipeline puede mejorar el rendimiento, ya que limita la cantidad de documentos que deben procesar las siguientes etapas.

```
db.listingsAndReviews.aggregate([  
  {$match:{amenities:"Wifi"}},//find  
  ⚡ {$project:{price: true, amenities:true }}, //$projection  
])
```





Aggregation \$project

# \$project



- Pasa solo los campos especificados a la siguiente etapa de agregación.

```
//Concatenar
db.restaurants.aggregate(
[
  {
    $match:{borough:"Brooklyn"}
  },
  {
    $match:{cuisine:"Mexican"}
  },
  {
    $project:{
      _id:0,name:1,grades:1,borough:1
    }
  }
]
)
```



Aggregation \$group

# \$group



- Esta etapa de agregación agrupa los documentos según el `_id` proporcionado.
- **Ojo!** No confunda este `_id` con el `_id` de `ObjectId` proporcionado a cada documento.

```
// 4. Agrupar por atributos
db.listado.aggregate([
  {
    $group:{
      _id:"$estado",
      nroAprendices:{$sum:1},
    }
  }
])
```





Aggregation \$limit

# \$limit



- Esta etapa de agregación limita el número de documentos que pasan a la siguiente etapa.
- Es distinto del \$project porque limita la cantidad de documentos que se muestran, pero no la cantidad de campos.

```
// 9.Muestra una pelicula  
use("sample_mflix")  
db.movies.aggregate([ { $limit: 1 } ])
```



Aggregation \$sort

# \$sort



- Esta etapa de agregación agrupa todos los documentos en el orden de clasificación especificado.
- Ojo! El orden de tus etapas importa: Cada etapa sólo actúa sobre los documentos que le proporcionan las etapas anteriores.

```
use("sample_restaurants")
db.restaurants.aggregate([
  {
    $match:{borough:"Brooklyn"}
  },
  {
    $match:{cuisine:"Mexican"}
  },
  {
    $project:{_id:0,name:1,"address.zipcode":1}
  },
  {
    $sort:{name:1}
  }
])
```





Aggregation \$count

# \$count



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

```
//2. Cuenta el total de las propiedades de tipo casa
db.listingsAndReviews.aggregate([
  {
    $match: { property_type : "House" }
  },
  {
    $count: "totalHouse"
  }
])
```



Aggregation \$lookup

# Ejemplo \$lookup



- <https://www.mongodb.com/docs/manual/reference/operator/aggregation/lookup/>

```
//10. Une los comentarios con las peliculas
// muestra solo un comentario
use("sample_mflix")
db.comments.aggregate([
  {
    $lookup: {
      from: "movies",
      localField: "movie_id",
      foreignField: "_id",
      as: "movie_details",
    },
  },
  {
    $limit: 1
  }
])
```



# \$lookup



- Esta etapa de agregación realiza una unión izquierda (left join) a una colección en la misma base de datos.
- Hay cuatro campos obligatorios:
  1. **from:** La colección que se usará para la búsqueda en la misma base de datos
  2. **localField:** el campo de la colección principal que se puede utilizar como identificador único en la colección from.
  3. **foreignField:** el campo de la colección from que se puede utilizar como identificador único en la colección principal.
  4. **as:** El nombre del nuevo campo que contendrá los documentos coincidentes de la colección from.



Aggregation \$addfields

# \$addfields



- Esta etapa de agregación agrega nuevos campos a los documentos.

```
use("sample_mflix")
db.comments.aggregate([
  {
    $lookup: {
      from: "movies",
      localField: "movie_id",
      foreignField: "_id",
      as: "movie_details",
    },
  },
  {
    $limit: 1
  }
])
```



Aggregation \$unwind



# Ejemplo \$unwind



- <https://www.mongodb.com/docs/manual/reference/operator/aggregation/unwind/>

```
db.restaurants.aggregate( [  
  {  
    $unwind: { path: "$grades" }  
  },  
  {  
    $limit: 4  
  }  
])
```



Aggregation \$map

# Ejemplo \$map



- <https://www.mongodb.com/docs/manual/reference/operator/aggregation/map/>

## Definition

### \$map

Applies an [expression](#) to each item in an array and returns an array with the applied results.

The [\\$map](#) expression has the following syntax:

```
{ $map: { input: <expression>, as: <string>, in: <expression> } }
```



# Ejemplo \$map



- <https://www.mongodb.com/docs/manual/reference/operator/aggregation/map/>

```
// $map
db.restaurants.aggregate([
  {
    $match: {borough: "Brooklyn"}
  },
  {
    $project: {
      adjustedGrades: {
        $map: {
          input: "$grades.score",
          as: "grade",
          in: { $divide: [ "$$grade", 5 ] }
        }
      }
    }
  }
])
```



Aggregation \$out

# Aggregation \$out



- Toma los documentos retornados por el pipeline de agregación y los envía a una colección específica.
- El operador **\$out** debe ser utilizado en la última etapa del pipeline.
- El operador **\$out** permite que el framework de agregación devuelva conjuntos de resultados de cualquier tamaño.
- \$out puede tomar especificar tanto la **base de datos** como la **colección** de salida:

```
{ $out: { db: "<output-db>", coll: "<output-collection>" } }
```

# Ejercicio resuelto en clase



- Utilizando la colección **listingsAndReviews** de la base de datos de **sample\_airbnb** ¿Cuáles son los 10 barrios de la ciudad de Sydney que tienen, en promedio, las locaciones más baratas de una habitación?





# GRACIAS

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)