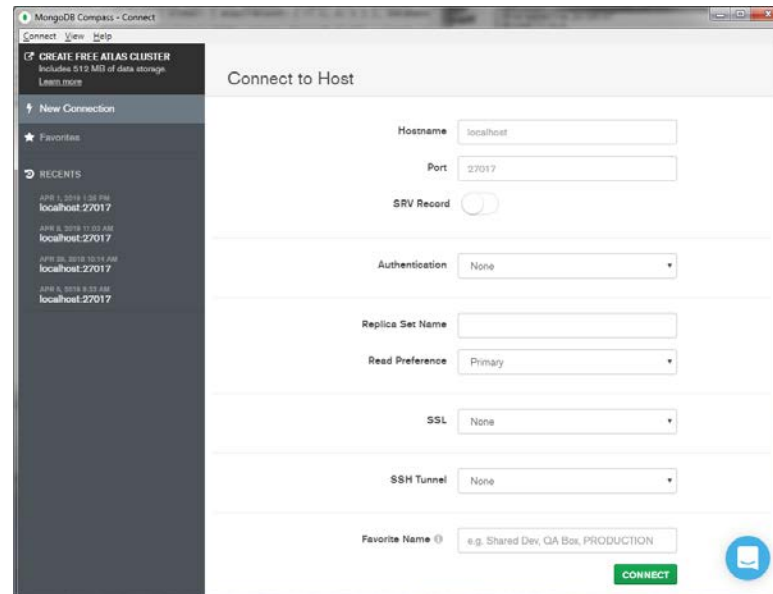


# COMPASS

- **Compass** es un cliente con interfaz gráfica que permite realizar operaciones con MongoDB de forma visual y más amigable que la *shell*
- **Conexión al servidor:** Muestra por defecto la conexión al puerto local 27017



## ***FRAMEWORK* de AGREGACIÓN**

---

El framework de agregación consta de un conjunto de operaciones que procesan los documentos para obtener nuevos resultados, calculados o transformados mediante combinación de sus valores.

Las operaciones agregadas se aplican sobre colecciones de documentos y devuelven los valores procesados en forma de uno o más documentos.

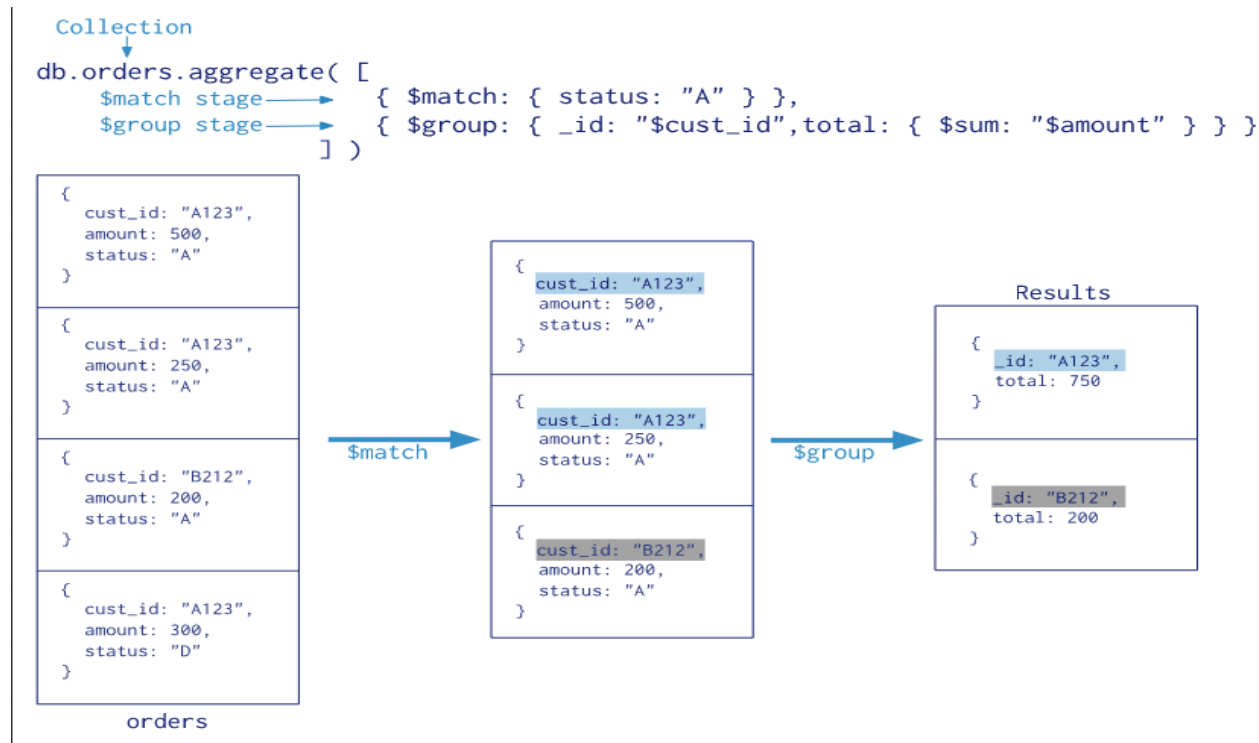
Tiene tres modos de funcionamiento:

- Agregación mediante pipeline (tuberías)
- La función map-reduce. Usa funciones que se deben escribir en JavaScript
- Operaciones de propósito único

# AGREGACIÓN con *Pipeline*

## Agregación con *Pipeline* (tuberías).

Cada operación de agregación o etapa parte de un conjunto de documentos y da como resultado otro conjunto de documentos. Las etapas se van encadenando como *pipes*, de modo que la salida de una etapa es la entrada para la siguiente.



## AGREGACIÓN con *Pipeline*

### Agregación con *Pipeline* (tuberías).

```
db.nombrecolección.aggregate ([{etapa1}, {etapa2}, ...])
```

### EJEMPLO:

```
db.AUTOR.aggregate ([
  {$match: {ANO_NAC: {$lt: 1900}}},
  {$group: {_id: "$NACION", CANTIDAD: {$sum: 1}}} ] )
```

```
> db.AUTOR.aggregate ([
...  {$match: {ANO_NAC: {$lt: 1900}}},
...  {$group: {_id: "$NACION", CANTIDAD: {$sum: 1}}} ] )
{ "_id" : "HUNGRIA", "CANTIDAD" : 1 }
{ "_id" : "SUIZA", "CANTIDAD" : 2 }
{ "_id" : "AUSTRIA", "CANTIDAD" : 1 }
{ "_id" : "ARGENTINA", "CANTIDAD" : 3 }
{ "_id" : "REPUBLICA CHECA", "CANTIDAD" : 1 }
{ "_id" : "PERU", "CANTIDAD" : 1 }
{ "_id" : "ESTADOS UNIDOS", "CANTIDAD" : 4 }
{ "_id" : "RUSIA", "CANTIDAD" : 2 }
{ "_id" : "ALEMANIA", "CANTIDAD" : 3 }
{ "_id" : "ESPANA", "CANTIDAD" : 9 }
{ "_id" : "HOLANDA", "CANTIDAD" : 1 }
{ "_id" : "LIBANO", "CANTIDAD" : 1 }
{ "_id" : "REINO UNIDO", "CANTIDAD" : 8 }
{ "_id" : "NICARAGUA", "CANTIDAD" : 1 }
```

Número de  
autores, por  
nacionalidad,  
nacidos antes  
de 1900

# AGREGACIÓN con *Pipeline*

## Agregación con *Pipeline* (tuberías).

Compass, proporciona una interfaz que facilita la definición de agregaciones

Etapas definidas

Muestras de los resultados de aplicar cada etapa

Se pueden seguir añadiendo nuevas etapas

The screenshot shows the MongoDB Compass interface for a database named 'biblio.AUTOR'. The 'Aggregations' tab is selected, and a pipeline named 'pipe1' is defined. The pipeline consists of two stages: '\$match' and '\$group'. The '\$match' stage is currently selected, and its output is displayed on the right. The output shows a sample of 20 documents, including one with a 'LIBROS' array. The '\$group' stage is also visible, and its output shows a sample of 19 documents, including one with a 'CANTIDAD' field. The interface includes a sidebar with a collection list, a central query editor, and a right-hand pane for stage outputs. A red circle highlights the 'Aggregations' tab, and a green arrow points to the 'ADD STAGE' button at the bottom.

Output after \$match stage (Sample of 20 documents)

```
{ "_id": "ObjectID(\"5caf21a9a6e946c57a783d71\")", "NOMBRE": "SAFO", "ANO_NAC": -640, "ANO_FALL": -580, "NACION": "GRECIA", "LIBROS": [ { "_id": "ObjectID(\"5caf21aaa6e946c57a783d71\")", "NOMBRE": "SOFOCLES", "ANO_NAC": -496, "ANO_FALL": -406, "NACION": "GRECIA", "LIBROS": [ ] } ] }
```

Output after \$group stage (Sample of 19 documents)

```
{ "_id": "ARGENTINA", "CANTIDAD": 3 }, { "_id": "REPUBLICA CHECA", "CANTIDAD": 1 }
```

# AGREGACIÓN con *Pipeline*

## Agregación con *Pipeline* en Compass

Exportación del código para distintos lenguajes

The image shows the MongoDB Compass interface with the 'Aggregations' tab selected. A red arrow points to the 'Export To Language' button in the pipeline editor. The pipeline consists of two stages: '\$match' and '\$group'. The '\$match' stage filters documents where 'ANO\_NAC' is 1900. The '\$group' stage groups documents by 'SNACION' and calculates the sum of 'CANTIDAD'.

The 'Export Pipeline To Language' dialog box is open, showing the pipeline code for 'PYTHON 3'. The dialog includes a 'CLOSE' button and an 'Include Import Statements' checkbox.

**My Pipeline:**

```
1 * $match: {
2   * ANO_NAC: {
3     * $lt: 1900
4   }
5 }
6
7 * $group: {
8   * _id: "$SNACION",
9   * CANTIDAD: {
10    * $sum: 1
11  }
12 }
13
14 ]]
```

**Export Pipeline To:** PYTHON 3

**Include Import Statements:** ☐

**CLOSE**

# AGREGACIÓN con *Pipeline*

## Etapas de agregación:

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

Etapas	Descripción	Ejemplo (Usando la colección SOLOAUTOR)
\$project	Cambia la forma del documento: añade, elimina o recalcula campos	<pre>\$project: {NOMBRE: 1,            APELLIDO: 1,            EDAD_FALL: {              \$subtract: [ "\$ANO_FALL", "\$ANO_NAC" ] } }</pre>
\$match	Filtra los documentos para que en la salida sólo estén los que cumplan unos criterios	<pre>\$match: {   ANO_NAC: { \$lt: 1900 } }</pre>
\$group	Agrupa documentos y realiza operaciones de agregación	<pre>\$group: {   _id: "\$NACION",   CANTIDAD: { \$sum: 1 } }</pre>
\$sort	Ordenación de documentos	<pre>\$sort: { APELLIDO: 1, NOMBRE: 1 }</pre>
\$skip	Ignora los primeros N documentos	<pre>\$skip: 10</pre>
\$limit	Obtiene los primeros N documentos	<pre>\$limit: 10</pre>
\$unwind	Descompone un array, desagrupando un documento en varios	<pre>\$unwind: { path: "\$LIBROS" }</pre>
\$lookup	Combina documentos con igual valor en algún campo	<pre>\$lookup: { from: "LIBRO",             localField: "LIBROS",             foreignField: "ISBN",             as: "RESULTADO" }</pre>
\$out	Envía el resultado a una nueva colección	<pre>\$out: "NUEVA_COLECCION"</pre>

# AGREGACIÓN con *Pipeline*

Ejemplo: \$lookup:

The screenshot shows the MongoDB Compass interface for a cluster named 'biblio.SOLOAUTOR'. The left sidebar shows the database structure with collections: admin, biblio, AUTOR, LECTOR, LIBRO, and SOLOAUTOR. The main panel displays the 'Aggregations' tab for the 'SOLOAUTOR' collection, which contains 145 documents. A pipeline stage named '\$lookup' is added and highlighted with a red circle. The stage configuration is as follows:

```
1 /*  
2  * from - The target collection.  
3  * localField - The local join field.  
4  * foreignField - The target join field.  
5  * as - The name for the results.  
6  */  
7 {  
8   from: "LIBRO",  
9   localField: "LIBROS",  
10  foreignField: "ISBN",  
11  as: "RESULTADO"  
12 }
```

The 'Preview of Documents in the Collection' shows two sample documents. The first document has fields: \_id, NOMBRE, ANO\_NAC, ANO\_FALL, NACION, and LIBROS. The second document has fields: \_id, NOMBRE, ANO\_NAC, ANO\_FALL, NACION, and LIBROS. The 'Output after \$lookup stage (Sample of 20 documents)' shows the result of the \$lookup operation, where the 'LIBROS' field is replaced by an array of objects, each containing the fields from the 'LIBRO' collection joined to the original document. The 'RESULTADO' field is highlighted with a red circle.



# AGREGACIÓN con *Pipeline*

**Operaciones de agregación:** Se utilizan para crear las expresiones que se pueden utilizar en las distintas etapas de agregación.

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

Operaciones aritméticas	<code>\$abs</code> <code>\$add</code> <code>\$ceil</code> <code>\$divide</code> <code>\$floor</code> <code>\$mod</code> <code>\$multiply</code> <code>\$pow</code> <code>\$sqrt</code> <code>\$subtract</code> <code>\$trunc</code>
Operaciones de grupo	<code>\$sum</code> <code>\$avg</code> <code>\$first</code> <code>\$last</code> <code>\$max</code> <code>\$min</code>
Operaciones con cadenas	<code>\$concat</code> <code>\$substr</code> <code>\$toLower</code> <code>\$toUpper</code> <code>\$strcasecmp</code>
Operaciones con fechas	<code>\$dayOfYear</code> <code>\$dayOfMonth</code> <code>\$dayOfWeek</code> <code>\$year</code> <code>\$month</code> <code>\$hour</code> <code>\$minute</code> <code>\$second</code> <code>\$dateToString</code>
Operaciones condicionales	<code>\$cond</code> <code>\$ifNull</code>

**Ejemplo:** `$project: { NOMBRE: { $toUpper: "$NOMBRE" },  
FECHA_NAC: 1,  
DIA: { $dayOfWeek: "$FECHA_NAC" } }`

**NOTA:** Para usar el contenido de un campo en una operación se escribe `"$nombrecampo"`

# ÍNDICES

## CREACIÓN DE ÍNDICES

`db.nombrecoleccion.createIndex(campos, opciones)`

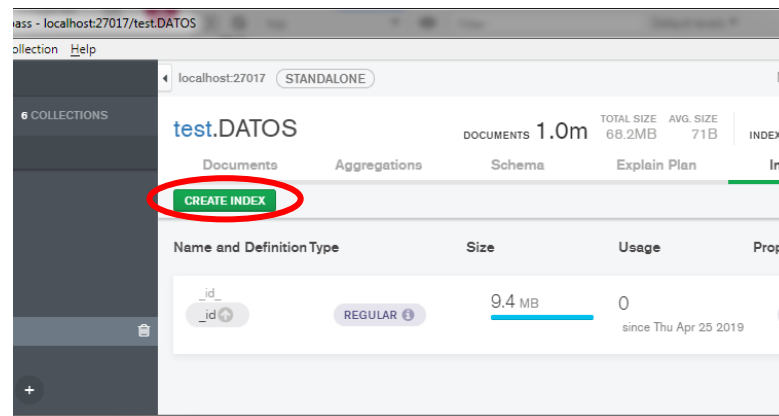
**campos:** indica los campos que componen el índice; a cada campo se le asigna un valor que indica si la información se almacena ordenada de modo ascendente (valor 1) o descendente (valor -1).

**opciones:** documento opcional que define opciones mediante los siguientes campos que pueden tomar los valores true o false

background: la creación del índice se hará en segundo plano permitiendo que se sigan ejecutando comandos en paralelo

unique: todas las claves del índice tendrán un valor único

En Compass



# ÍNDICES

---

## BORRADO DE ÍNDICES

Borrado de un índice:

```
db.nombrecoleccion.dropIndex(campos)
```

**campos:** indica los campos que componen el índice a borrar

Borrado de todos los índices de una colección:

```
db.nombrecoleccion.dropIndexes()
```

## CONSULTA DE ÍNDICES

Obtener información de todos los índices de una colección:

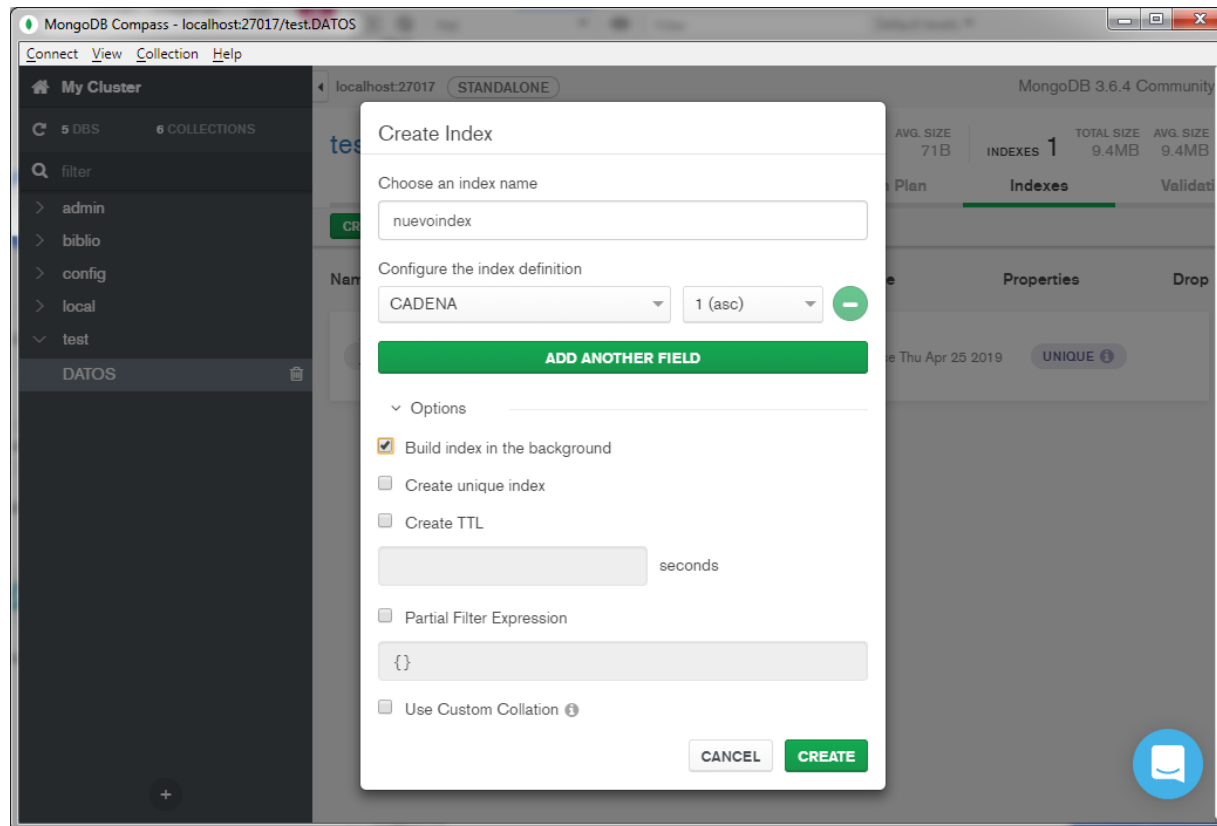
```
db.nombrecoleccion.getIndexes()
```

# ÍNDICES

## Ejemplo

```
db.DATOS.createIndex( { "CADENA":1} , {background:true} )
```

## En Compass



# Explain plan

## Explain plan

Ejemplo de plan de ejecución de una consulta **sin haber definido un índice** sobre el campo "CADENA"

MongoDB Compass - localhost:27017/test.DATOS

localhost:27017 (STANDALONE) MongoDB 3.6.4 Community

test.DATOS DOCUMENTS 1.0m TOTAL SIZE 68.2MB AVG. SIZE 71B INDEXES 1 TOTAL SIZE 9.4MB AVG. SIZE 9.4MB

Documents Aggregations Schema Explain Plan Indexes

FILTER {"CADENA": "mmx1"} OPTIONS EXPLAIN RESET

VIEW DETAILS AS VISUAL TREE RAW JSON

Query Performance Summary

- Documents Returned: 1
- Index Keys Examined: 0
- Documents Examined: 1000000
- Actual Query Execution Time (ms): 708
- Sorted in Memory: no
- No index available for this query.

COLLSCAN

nReturned: 1 Execution Time: 550 ms

Documents Examined: 1000000

DETAILS

```
{
  "stage": "COLLSCAN",
  "filter": {
    "CADENA": {
      "seq": "mmx1"
    }
  },
  "nReturned": 1,
  "executionTimeMillisEstimate": 550,
  "works": 1000002,
  "advanced": 1,
  "needTime": 1000000,
  "needYield": 0,
  "saveState": 7812,
  "restoreState": 7812,
  "isEOF": 1,
  "invalidates": 0,
  "direction": "forward",
  "docsExamined": 1000000
}
```

# Explain plan

**Explain plan:** Muestra el plan de ejecución de una consulta

Ejemplo de plan de ejecución de una consulta **con un índice** sobre el campo "CADENA"

MongoDB Compass - localhost:27017/test.DATOS

Connect View Collection Help

My Cluster

localhost:27017 STANDALONE MongoDB 3.6.4 Community

test.DATOS

DOCUMENTS 1.0m TOTAL SIZE 68.2MB AVG. SIZE 71B INDEXES 2 TOTAL SIZE 33.0MB AVG. SIZE 16.5MB

Documents Aggregations Schema Explain Plan Indexes

FILTER {"CADENA": "mmx1"} OPTIONS EXPLAIN RESET

VIEW DETAILS AS VISUAL TREE RAW JSON

Query Performance Summary

- Documents Returned: 1
- Index Keys Examined: 1
- Documents Examined: 1
- Actual Query Execution Time (ms): 152
- Sorted in Memory: no
- Query used the following index: CADENA

FETCH

nReturned: 1 Execution Time: 0 ms

DETAILS

IXSCAN

nReturned: 1 Execution Time: 0 ms

Index Name: CADENA\_1

Multi Key Index: no

DETAILS

# Documentación MongoDB

---

Para ampliar información:

<https://docs.mongodb.com/>

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

<https://university.mongodb.com>