

TALLER MONGO DB

METODO 1

Crea una base de datos llamada tallerMongoDB.

```
db.createCollection("usuarios")  
  
{ ok: 1 }
```

2. Inserta la siguiente información en una colección llamada usuarios:

```
db.usuarios.insertMany([ {nombre:"juan perez",edad:30,correo:"juan.perez@urosario.com"},  
  {nombre:"ana lopez",edad:25,correo:"ana.lopez@urosario.com"},  
  {nombre:"Luis torres",edad:35,correo:"luis.torres@urosario.com"} ])  
  
{  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId('68225bc8ae913d8a2386135c'),  
    '1': ObjectId('68225bc8ae913d8a2386135d'),  
    '2': ObjectId('68225bc8ae913d8a2386135e')  
  }  
}
```

3. • Encuentra todos los usuarios.

```
db.usuarios.find()  
  
{  
  _id: ObjectId('68225bc8ae913d8a2386135c'),  
  nombre: 'juan perez',  
  edad: 30,  
  correo: 'juan.perez@urosario.com'  
}  
  
{  
  _id: ObjectId('68225bc8ae913d8a2386135d'),  
  nombre: 'ana lopez',  
  edad: 25,  
  correo: 'ana.lopez@urosario.com'  
}  
  
{  
  _id: ObjectId('68225bc8ae913d8a2386135e'),  
  nombre: 'Luis torres',  
  edad: 35,  
  correo: 'luis.torres@urosario.com'  
}
```

Encuentra el usuario con nombre Ana López

```
db.usuarios.find({nombre:{$eq:"ana lopez"}})
{
  _id: ObjectId('68225bc8ae913d8a2386135d'),
  nombre: 'ana lopez',
  edad: 25,
  correo: 'ana.lopez@urosario.com'
}
```

Encuentra todos los usuarios mayores o iguales a 30 años

```
db.usuarios.find({edad:{$gte:30}})
{
  _id: ObjectId('68226eff1fbda6fb5b9cbb79'),
  nombre: 'juan perez',
  edad: 30,
  correo: 'juan.perez@urosario.com'
}
{
  _id: ObjectId('68226eff1fbda6fb5b9cbb7b'),
  nombre: 'Luis torres',
  edad: 35,
  correo: 'luis.torres@urosario.com'
}
```

4. Actualiza los datos

Cambia la edad de Juan Pérez a 31 años.

```
db.usuarios.updateOne({nombre:{$eq:"juan perez"}},{ $set:{edad:31}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Añade el campo activo: true a todos los usuarios con edad mayor o igual a 30 años.

```
db.usuarios.updateMany({edad:{$gt:30}},{ $set:{activo:"true"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

5. Elimina registros: •

Elimina el usuario Luis Torres.

```
db.usuarios.deleteMany({nombre:{$eq:"Luis torres"}})
{
  acknowledged: true,
  deletedCount: 1
}
```

Elimina todos los usuarios menores de 30 años.

```
db.usuarios.deleteMany({edad:{$lt:30}})
{
  acknowledged: true,
  deletedCount: 1
}
```

Parte 2: Gestión de Productos

Crea una nueva colección llamada productos e inserta al menos 5 productos con los siguientes campos:

```
db.productos.insertMany([{nombre:"camisa oversized",precio:20,categoria:"ropa"},
{nombre:"lavadora",precio:700,categoria:"hogar"},
{nombre:"cafetera",precio:30,categoria:"electronica"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68226046ae913d8a2386135f'),
    '1': ObjectId('68226046ae913d8a23861360'),
    '2': ObjectId('68226046ae913d8a23861361')
  }
}
```

```
db.productos.insertMany([{nombre:"ps5",precio:600,categoria:"electronica"},
{nombre:"portatil",precio:700,categoria:"electronica"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68226af1ae913d8a23861367'),
    '1': ObjectId('68226af1ae913d8a23861368')
  }
}
```

. Realiza las siguientes consultas: •

Encuentra todos los productos con precio mayor a \$100.

```
db.productos.find({precio:{$gt:100}})
{
  _id: ObjectId('68226aeca913d8a23861365'),
  nombre: 'lavadora',
  precio: 700,
  categoria: 'hogar'
}
{
  _id: ObjectId('68226af1ae913d8a23861367'),
  nombre: 'ps5',
  precio: 600,
  categoria: 'electronica'
}
{
  _id: ObjectId('68226af1ae913d8a23861368'),
  nombre: 'portatil',
  precio: 700,
  categoria: 'electronica'
}
```

Ordena los productos por precio de manera descendente

```
db.productos.find().sort({precio:-1})
{
  _id: ObjectId('68226aeca913d8a23861365'),
  nombre: 'lavadora',
  precio: 700,
  categoria: 'hogar'
}
{
  _id: ObjectId('68226af1ae913d8a23861368'),
  nombre: 'portatil',
  precio: 700,
  categoria: 'electronica'
}
{
  _id: ObjectId('68226af1ae913d8a23861367'),
  nombre: 'ps5',
  precio: 600,
  categoria: 'electronica'
}
{
  _id: ObjectId('68226aeca913d8a23861366'),
  nombre: 'cafetera',
  precio: 30,
  categoria: 'electronica'
}
{
  _id: ObjectId('68226aeca913d8a23861364'),
  nombre: 'camisa oversized',
```

```
precio: 20,
categoria: 'ropa'
```

Actualiza los datos: •

Añade un campo en stock con valor true a todos los productos

```
db.productos.updateMany({},{$set:{"en stock":"true"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
```

- Cambia el valor de en stock a false para los productos cuyo precio sea mayor a \$500.

```
db.productos.updateMany({precio:{$gt:500}},{$set:{"en stock":"false"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

- Elimina registros: • Elimina todos los productos cuyo precio sea menor a \$50

```
db.productos.deleteMany({precio:{$lt:50}})
{
  acknowledged: true,
  deletedCount: 2
}
```

Parte 3: Agregaciones y Análisis 1. Realiza una agregación para calcular: • El precio promedio de los productos agrupados por categoría.

```
db.productos.aggregate([{$group: {_id: "$categoria",precioPromedio: { $avg: "$precio" }}}])
{
  _id: 'hogar',
  precioPromedio: 700
}
{
  _id: 'electronica',
  precioPromedio: 650
}
```

// para este punto busque documentación:

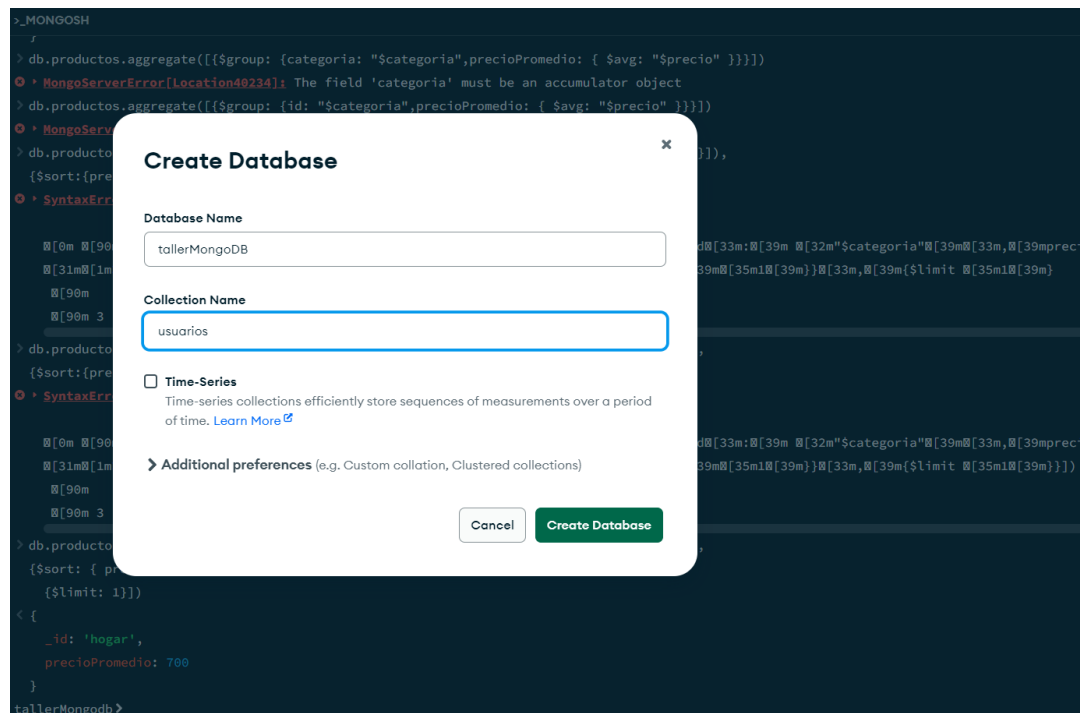
<https://www.mongodb.com/docs/manual/aggregation/>

Crea una consulta que devuelva: • La categoría con el mayor precio promedio

```
db.productos.aggregate([{$group: {_id: "$categoria", precioPromedio: { $avg: "$precio" } }},
{$sort: { precioPromedio: -1 }},
{$limit: 1}])
{
  _id: 'hogar',
  precioPromedio: 700
}
```

METODO 2

Crea una base de datos llamada tallerMongoDB. 2. Inserta la siguiente información en una colección llamada usuarios:



Insert Document



To collection tallerMongoDB.usuarios

VIEW



```
1  [
2  {
3    "nombre": "juan perez", "edad": 30, "correo": "juan.perez@uros:
4  },
5  {
6    "nombre": "ana lopez", "edad": 25, "correo": "ana.lopez@urosar:
7  },
8  {
9    "nombre": "Luis torres", "edad": 35, "correo": "luis.torres@
10 }
11 ]
```

Cancel

Insert



```
_id: ObjectId('68226eff1fbda6fb5b9cbb79')
nombre : "juan perez"
edad : 30
correo : "juan.perez@urosario.com"
```

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7a')
nombre : "ana lopez"
edad : 25
correo : "ana.lopez@urosario.com"
```

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7b')
nombre : "Luis torres"
edad : 35
correo : "luis.torres@urosario.com"
```

3. Realiza las siguientes consultas: • Encuentra todos los usuarios

🔍

{}

Generate query

Explain

Reset

Find

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 3 of 3

↺

↻

↷

▶

```
_id: ObjectId('68226eff1fbda6fb5b9cbb79')
nombre: "juan perez"
edad: 30
correo: "juan.perez@urosario.com"
```

⌵

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7a')
nombre: "ana lopez"
edad: 25
correo: "ana.lopez@urosario.com"
```

⌵

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7b')
nombre: "Luis torres"
edad: 35
correo: "luis.torres@urosario.com"
```

⌵

Encuentra el usuario con nombre Ana López.

🔍

```
{"nombre":{"$eq:"ana lopez"}}
```

Generate query

Explain

Reset

Find

</>

ADD DATA

EXPORT DATA

UPDATE

DELETE

INSIGHT

25

1 - 1 of 1

↺

↻

↷

☰

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7a')
nombre: "ana lopez"
edad: 25
correo: "ana.lopez@urosario.com"
```

⌵

Encuentra todos los usuarios mayores o iguales a 30 años.

🕒

{ "edad": { "\$gte": 30 } }

Generate query 🚀

+ ADD DATA ▾

📄 EXPORT DATA ▾

✎ UPDATE

🗑 DELETE

💡 INSIGHT

25

_id: ObjectId('68226eff1fbda6fb5b9cbb79')

nombre : "juan perez"

edad : 30

correo : "juan.perez@urosario.com"

_id: ObjectId('68226eff1fbda6fb5b9cbb7b')

nombre : "Luis torres"

edad : 35

correo : "luis.torres@urosario.com"

Cambia la edad de Juan Pérez a 31 años

Update 1 document

tallerMongoDB.usuarios

Filter ⓘ

```
{ nombre: { $eq: 'juan perez' } }
```

Update

[Learn more about Update syntax](#)

```
1 {
2   $set: {
3     edad: 31
4   },
5 }
```

★ Save

Cancel

Update 1 document

```
_id: ObjectId('68226eff1fbda6fb5b9cbb79')
nombre : "juan perez"
edad : 31
correo : "juan.perez@urosario.com"
```

Añade el campo activo: true a todos los usuarios con edad mayor o igual a 30 años.

Update 2 documents



tallerMongoDB.usuarios

Filter ⓘ

```
{ edad: { $gte: 30 } }
```

Update

[Learn more about Update syntax](#)

```
1 {
2   $set: {
3     activo: "true"
4   },
5 }
```



★ Save



Cancel

Update 2 documents


```
_id: ObjectId('68226eff1fbda6fb5b9cbb79')
nombre : "juan perez"
edad : 31
correo : "juan.perez@urosario.com"
activo : "true"
```

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7b')
nombre : "Luis torres"
edad : 35
correo : "luis.torres@urosario.com"
activo : "true"
```

. Elimina registros: • Elimina el usuario Luis Torres.

 **Delete 1 document** 

tallerMongoDB.usuarios

Filter ⓘ { nombre: { \$eq: 'Luis torres' } }  Export

Preview (sample of 1 document)

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7b')
nombre: "Luis torres"
edad: 35
correo: "luis.torres@urosario.com"
activo: "true"
```

Cancel

Delete 1 document

- Elimina todos los usuarios menores de 30 años.

 **Delete 1 document** 

tallerMongoDB.usuarios

Filter ⓘ { edad: { \$lt: 30 } }  Export

Preview (sample of 1 document)

```
_id: ObjectId('68226eff1fbda6fb5b9cbb7a')
nombre: "ana lopez"
edad: 25
correo: "ana.lopez@urosario.com"
```

Cancel

Delete 1 document

Parte 2: Gestión de Productos 1. Crea una nueva colección llamada productos e inserta al menos 5 productos con los siguientes campos:

Create Collection



Collection Name

productos

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ Additional preferences (e.g. Custom collation, Clustered collections)

Cancel

Create Collection

VIEW

```
1 [
2   { "nombre": "camisa oversized", "precio": 20, "categoria": "ropa" },
3   { "nombre": "lavadora", "precio": 700, "categoria": "hogar" },
4   { "nombre": "cafetera", "precio": 30, "categoria": "electronica" },
5   { "nombre": "ps5", "precio": 600, "categoria": "electronica" },
6   { "nombre": "portatil", "precio": 700, "categoria": "electronica" }
7 ]
```

Cancel

Insert

<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb80') nombre : "camisa oversized" precio : 20 categoria : "ropa" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb81') nombre : "lavadora" precio : 700 categoria : "hogar" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb82') nombre : "cafetera" precio : 30 categoria : "electronica" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb83') nombre : "ps5" precio : 600 categoria : "electronica" </pre>
<div>REFRESH</div> <pre> _id: ObjectId('6822738a1fbda6fb5b9cbb84') nombre : "portatil" precio : 700 categoria : "electronica" </pre>

2. Realiza las siguientes consultas: • Encuentra todos los productos con precio mayor a \$100.

<div>🕒</div> <div>{precio:{\$gt:30}}</div> <div>Geni</div>
<div>+ ADD DATA</div> <div>EXPORT DATA</div> <div>UPDATE</div> <div>DELETE</div> <div>INSIGHT</div>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb81') nombre : "lavadora" precio : 700 categoria : "hogar" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb83') nombre : "ps5" precio : 600 categoria : "electronica" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb84') nombre : "portatil" precio : 700 categoria : "electronica" </pre>

• Ordena los productos por precio de manera descendente

Type a query: { field: 'value' } or [Generate query](#)

[Explain](#) [Reset](#) [Find](#) [</>](#)

Project {}

Sort {precio:-1} **Max Time MS** 60000

Collation { locale: 'simple' } **Skip** 0 **Limit** 0

Index Hint { field: -1 }

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#) [INSIGHT](#) 25 1 - 5 of 5

```
{
  "_id": ObjectId('6822738a1fbda6fb5b9cbb81'),
  "nombre": "lavadora",
  "precio": 700,
  "categoria": "hogar"
}
```

```
{
  "_id": ObjectId('6822738a1fbda6fb5b9cbb84'),
  "nombre": "portatil",
  "precio": 700,
  "categoria": "electronica"
}
```

```
{
  "_id": ObjectId('6822738a1fbda6fb5b9cbb83'),
  "nombre": "ps5",
  "precio": 600,
  "categoria": "electronica"
}
```

3. Actualiza los datos: • Añade un campo en stock con valor true a todos los productos.

```
1 {
2   $set: {
3     "en stock": "true"
4   },
5 }
```


<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb80') nombre: "camisa oversized" precio: 20 categoria: "ropa" en stock: "true" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb81') nombre: "lavadora" precio: 700 categoria: "hogar" en stock: "true" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb82') nombre: "cafetera" precio: 30 categoria: "electronica" en stock: "true" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb83') nombre: "ps5" precio: 600 categoria: "electronica" en stock: "true" </pre>

- Cambia el valor de en stock a false para los productos cuyo precio sea mayor a \$500

Update 3 documents

tallerMongoDB.productos

Filter ⓘ

```
{ precio: { $gt: 500 } }
```

Update

[Learn more about Update syntax](#)

```

1 {
2   $set: {
3     "en stock": "false"
4   },
5 }

```

★ Save

Cancel

Update 3 documents

{precio:{\$gt:500}}

Get

ADD DATA

EXPORT DATA

UPDATE

DELETE

INSIGHT

<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb81') nombre: "lavadora" precio: 700 categoria: "hogar" en stock: "false" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb83') nombre: "ps5" precio: 600 categoria: "electronica" en stock: "false" </pre>
<pre> _id: ObjectId('6822738a1fbda6fb5b9cbb84') nombre: "portatil" precio: 700 categoria: "electronica" en stock: "false" </pre>

- Elimina todos los productos cuyo precio sea menor a \$50.

Delete 2 documents

tallerMongoDB.productos

Filter

{ precio: { \$lt: 50 } }

Export

Preview (sample of 2 documents)

```

_id: ObjectId('6822738a1fbda6fb5b9cbb80')
nombre: "camisa oversized"
precio: 20
categoria: "ropa"
en stock: "true"

```

```

_id: ObjectId('6822738a1fbda6fb5b9cbb82')
nombre: "cafetera"
precio: 30
categoria: "electronica"
en stock: "true"

```

Cancel

Delete 2 documents

Parte 3: Agregaciones y Análisis 1. Realiza una agregación para calcular:

- El precio promedio de los productos agrupados por categoría.

▼ Stage 1 \$group 


```
1  /**
2   * _id: The id of the group.
3   * fieldN: The first field name.
4   */
5  {
6    _id: "$categoria",
7    precioPromedio: {
8      $avg: "$precio"
9    }
10 }
```

Output after \$group stage (Sample of 2 documents)

<code>_id: "hogar"</code> <code>precioPromedio : 700</code>	<code>_id: "electronica"</code> <code>precioPromedio : 650</code>
--	--

La categoría con el mayor precio promedio


//me costo bastante, lo que hice fue usar la parte de compass “agregations” y dividi la consulta de mayor promedio en 3, primero promedios luego orden luego limite


▼ Stage 2 \$sort 

```
1  /**
2   * Provide the number of documents to limit
3   */
4  {precio:-1}
```

Output after \$sort stage (Sample of 2 documents)

<code>_id: "electronica"</code> <code>precioPromedio : 650</code>	<code>_id: "hogar"</code> <code>precioPromedio : 700</code>
--	--



▼ Stage 3 \$limit 

```
1  /**
2   * Provide the number of documents to limit
3   */
4  1
```

Output after \$limit stage (Sample of 1 document)

<code>_id: "electronica"</code> <code>precioPromedio : 650</code>
--