

## EVALUACION FINAL – BASE DE DATOS NO RELACIONALES

### VENTA DE ARTÍCULOS USADOS

#### A. Restauración de la base de datos

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

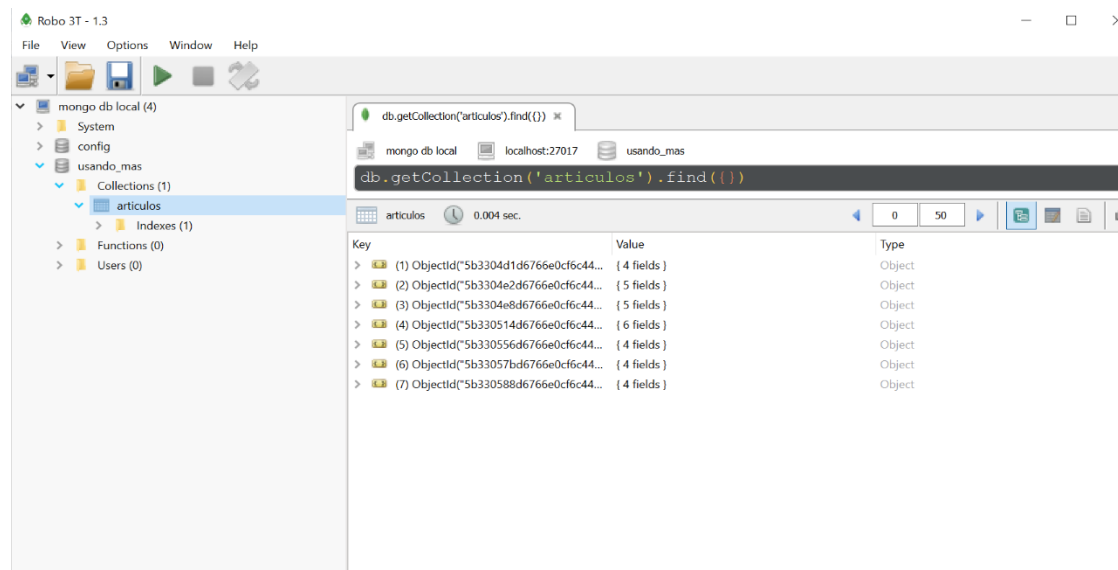
C:\Users\gaby>cd..

C:\Users>cd..

C:\>cd C:\Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongorestore C:\baseDatos\dump
2020-01-26T12:20:52.438-0300    preparing collections to restore from
2020-01-26T12:20:52.506-0300    reading metadata for usando_mas.articulos from C:\baseDatos\dump\usando_mas.articulos.me
tadata.json
2020-01-26T12:20:52.605-0300    restoring usando_mas.articulos from C:\baseDatos\dump\usando_mas.articulos.bson
2020-01-26T12:20:52.609-0300    no indexes to restore
2020-01-26T12:20:52.610-0300    finished restoring usando_mas.articulos (7 documents, 0 failures)
2020-01-26T12:20:52.610-0300    7 document(s) restored successfully. 0 document(s) failed to restore.

C:\Program Files\MongoDB\Server\4.2\bin>
```



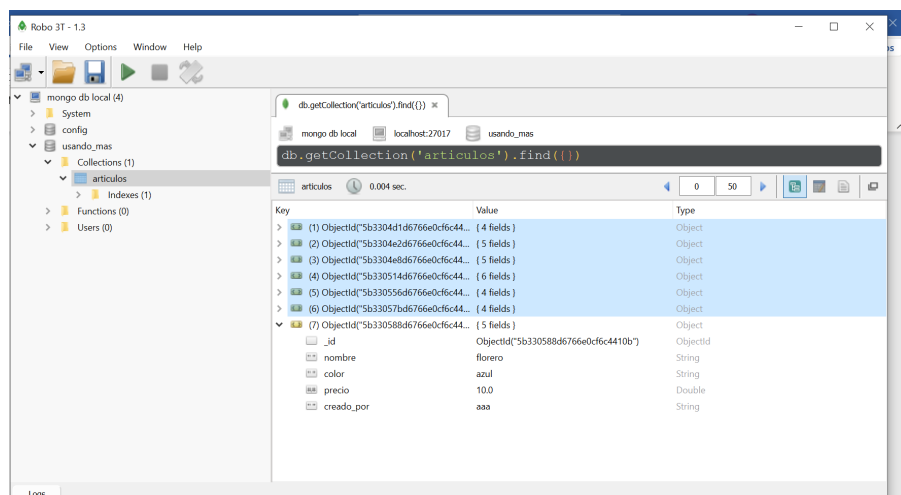
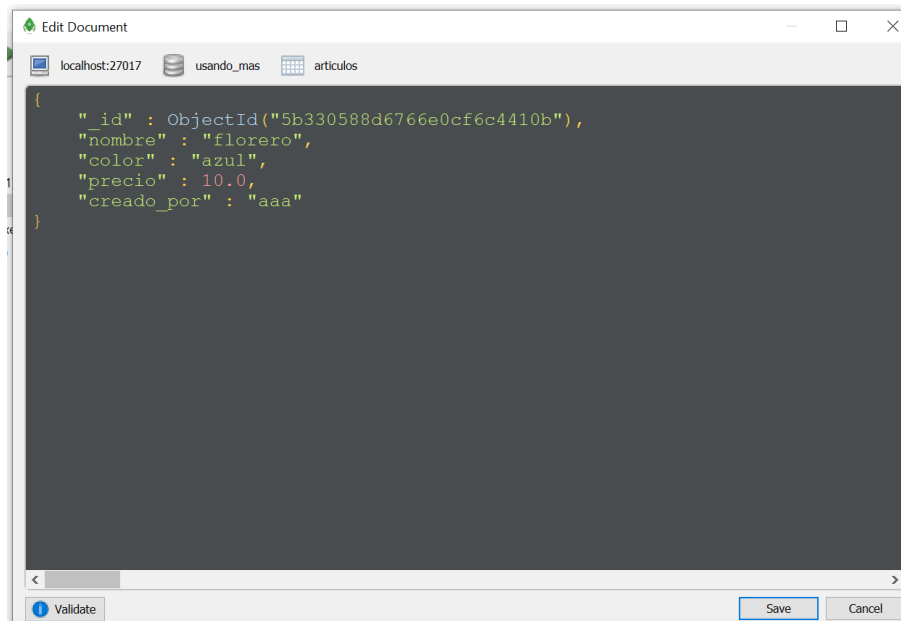
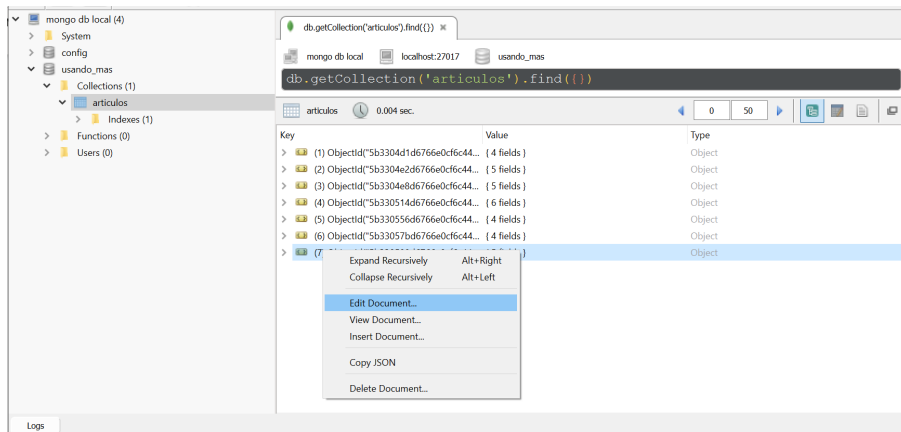
#### B. Actualización de los siguientes productos de la Base de Datos

Florero

Color: azul

Precio: 10

Creado\_por: aaa



## C. Insertando nuevos documentos

Mesa

Color: gris

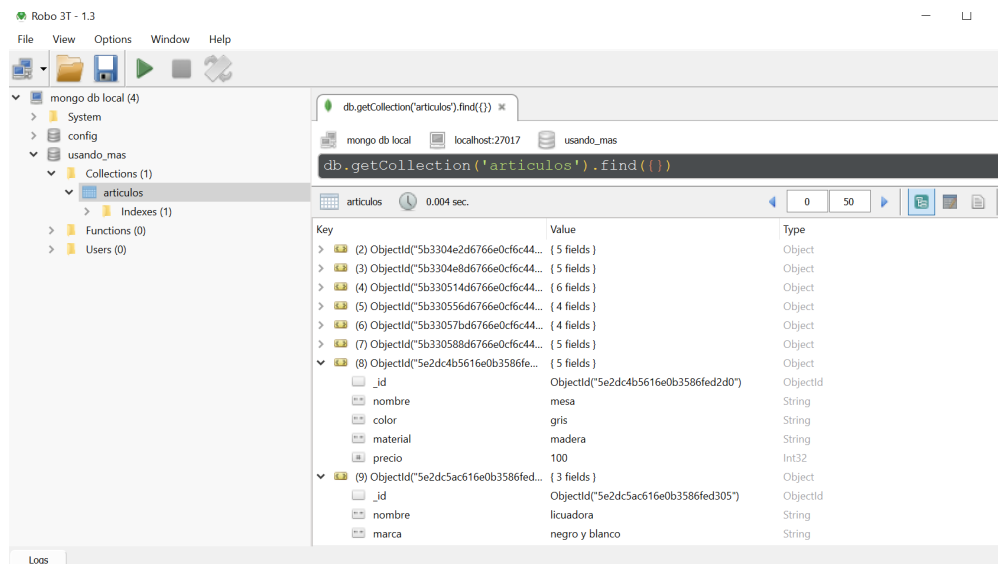
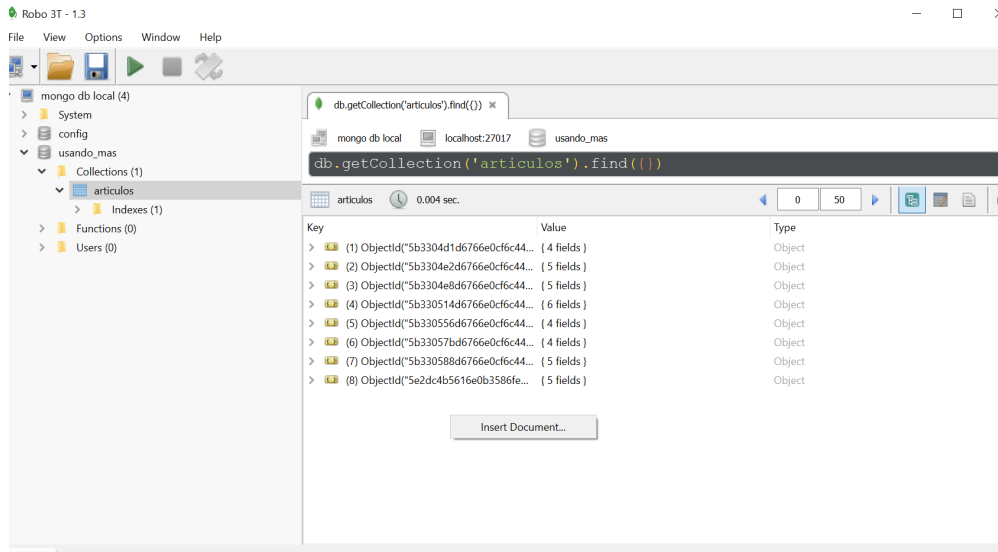
Material: madera

Precio: 100

Licuada

Precio:50

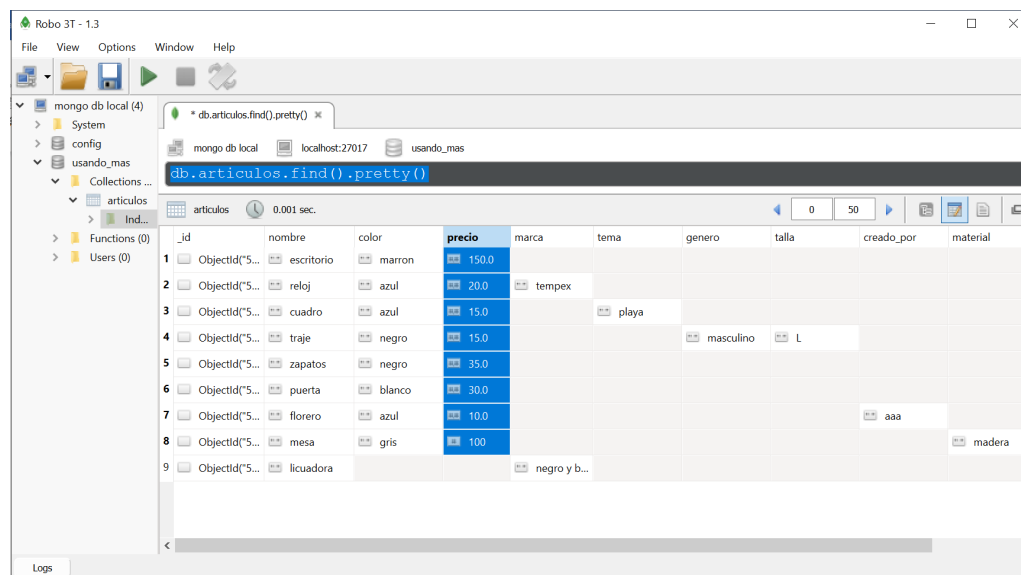
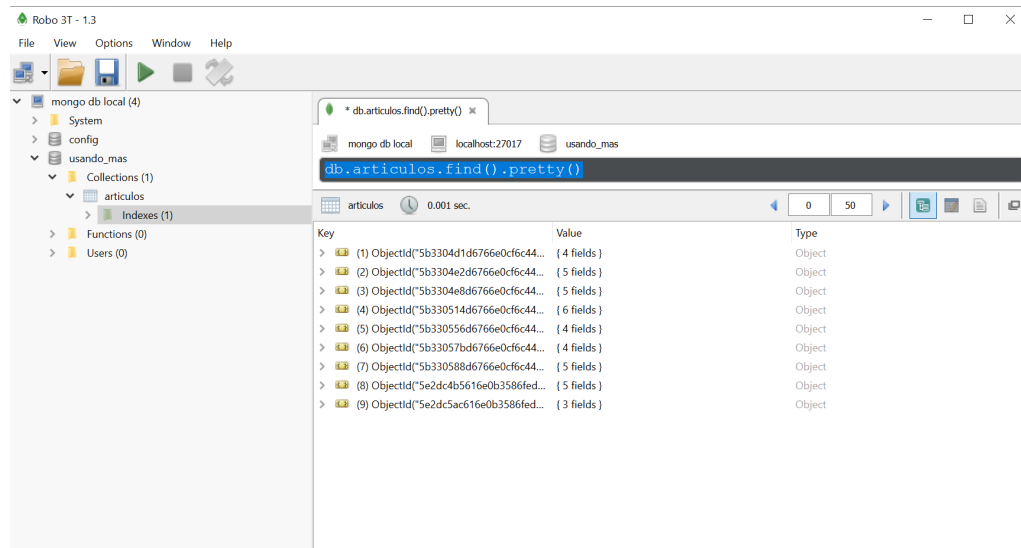
Marca: negro y blanco



## D. Información para el CEO:

### 1. Todos los artículos con todos los atributos

```
db.articulos.find().pretty()
```



```
/* 1 */
{
  "_id" : ObjectId("5b3304d1d6766e0cf6c44105"),
  "nombre" : "escritorio",
  "color" : "marron",
  "precio" : 150.0
}
```

```
/* 2 */
{
  "_id" : ObjectId("5b3304e2d6766e0cf6c44106"),
  "nombre" : "reloj",
  "color" : "azul",
  "marca" : "tempex",
  "precio" : 20.0
}
```

```
/* 3 */
{
  "_id" : ObjectId("5b3304e8d6766e0cf6c44107"),
  "nombre" : "cuadro",
  "color" : "azul",
  "tema" : "playa",
  "precio" : 15.0
}
```

```
/* 4 */
{
  "_id" : ObjectId("5b330514d6766e0cf6c44108"),
  "nombre" : "traje",
  "color" : "negro",
  "genero" : "masculino",
  "talla" : "L",
  "precio" : 15.0
}
```

```
/* 5 */
{
  "_id" : ObjectId("5b330556d6766e0cf6c44109"),
  "nombre" : "zapatos",
  "color" : "negro",
  "precio" : 35.0
}
```

```
/* 6 */
{
  "_id" : ObjectId("5b33057bd6766e0cf6c4410a"),
```

```

    "nombre" : "puerta",
    "color" : "blanco",
    "precio" : 30.0
  }

/* 7 */
{
  "_id" : ObjectId("5b330588d6766e0cf6c4410b"),
  "nombre" : "florero",
  "color" : "azul",
  "precio" : 10.0,
  "creado_por" : "aaa"
}

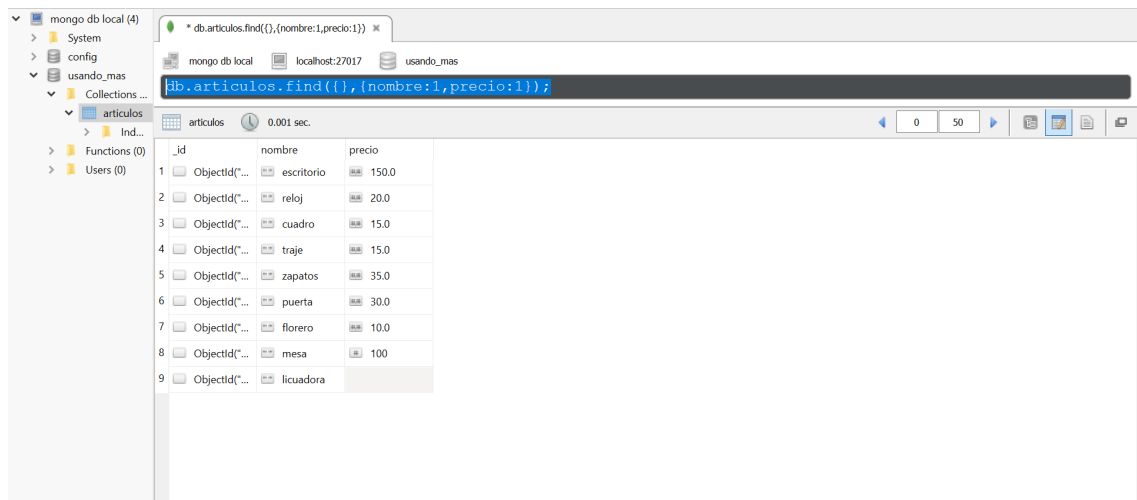
/* 8 */
{
  "_id" : ObjectId("5e2dc4b5616e0b3586fed2d0"),
  "nombre" : "mesa",
  "color" : "gris",
  "material" : "madera",
  "precio" : 100
}

/* 9 */
{
  "_id" : ObjectId("5e2dc5ac616e0b3586fed305"),
  "nombre" : "licuadora",
  "marca" : "negro y blanco"
}

```

## 2. Todos los artículos con solo nombre y costo del producto

```
db.articulos.find({}, {nombre:1, precio:1});
```



The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'mongo db local (4)' containing 'System', 'config', 'usando\_mas', and 'articulos'. The 'articulos' collection is selected. The main window shows the query `db.articulos.find({}, {nombre:1, precio:1});` and its results. The results are displayed in a table with 9 documents, each containing an `_id`, `nombre`, and `precio`.

	_id	nombre	precio
1	ObjectId("5b330588d6766e0cf6c4410b")	escritorio	150.0
2	ObjectId("5b330588d6766e0cf6c4410b")	reloj	20.0
3	ObjectId("5b330588d6766e0cf6c4410b")	cuadro	15.0
4	ObjectId("5b330588d6766e0cf6c4410b")	traje	15.0
5	ObjectId("5b330588d6766e0cf6c4410b")	zapatos	35.0
6	ObjectId("5b330588d6766e0cf6c4410b")	puerta	30.0
7	ObjectId("5b330588d6766e0cf6c4410b")	florero	10.0
8	ObjectId("5b330588d6766e0cf6c4410b")	mesa	100
9	ObjectId("5e2dc5ac616e0b3586fed305")	licuadora	

mongo db local (4)

System

config

usando\_mas

Collections...

articulos

Ind...

Functions (0)

Users (0)

articulos 0.001 sec.

0 50

	nombre	precio
1	escritorio	150.0
2	reloj	20.0
3	cuadro	15.0
4	traje	15.0
5	zapatos	35.0
6	puerta	30.0
7	florero	10.0
8	mesa	100
9	licuadora	

Logs

### 3. Todos los artículos con solo nombre y costo del producto ordenados por costo

```
db.articulos.find({}, {nombre:1, precio:1, _id:0}).sort({precio:1});
```

mongo db local (4)

System

config

usando\_mas

Collections...

articulos

Ind...

Functions (0)

Users (0)

articulos 0.005 sec.

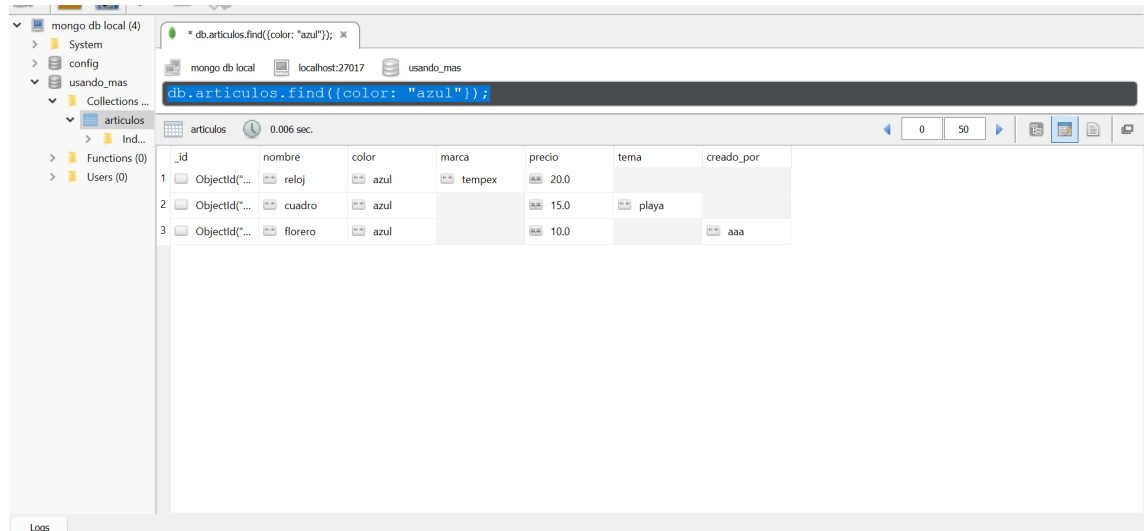
0 50

	nombre	precio
1	licuadora	
2	florero	10.0
3	cuadro	15.0
4	traje	15.0
5	reloj	20.0
6	puerta	30.0
7	zapatos	35.0
8	mesa	100
9	escritorio	150.0

Logs

#### 4. Todos los productos de color azul

```
db.articulos.find({color: "azul"});
```

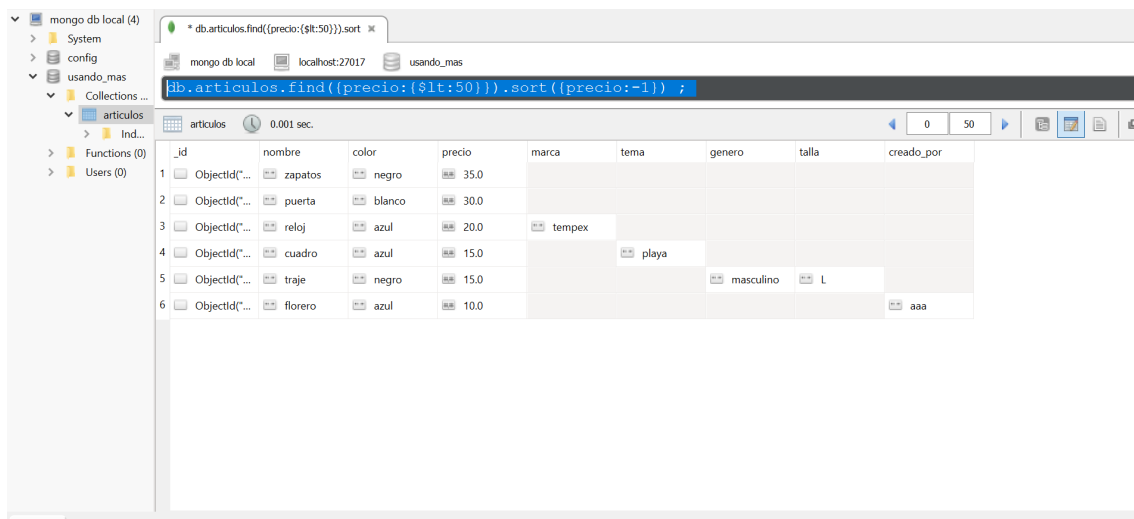


The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'mongo db local (4)' containing 'System', 'config', 'usando\_mas', and 'articulos'. The 'articulos' collection is selected. The main window shows the query `db.articulos.find({color: "azul"});` in the command bar. Below the query, a table displays the results of the query, showing 3 items. The table has columns: `_id`, `nombre`, `color`, `marca`, `precio`, `tema`, and `creado_por`. The results are:

_id	nombre	color	marca	precio	tema	creado_por
1	Objectid("...")	reloj	azul	tempex	20.0	
2	Objectid("...")	cuadro	azul		15.0	playa
3	Objectid("...")	florero	azul		10.0	aaa

#### 5. Todos los productos con precio menor a 50, ordenados descendentemente por precio.

```
db.articulos.find({precio:{$lt:50}}).sort({precio:-1}) ;
```



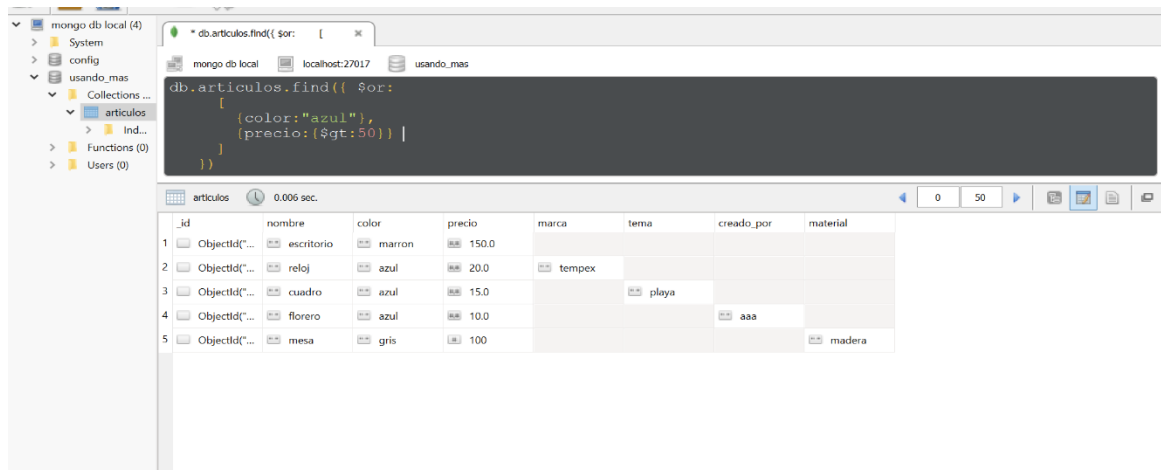
The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'mongo db local (4)' containing 'System', 'config', 'usando\_mas', and 'articulos'. The 'articulos' collection is selected. The main window shows the query `db.articulos.find({precio:{$lt:50}}).sort({precio:-1}) ;` in the command bar. Below the query, a table displays the results of the query, showing 6 items. The table has columns: `_id`, `nombre`, `color`, `precio`, `marca`, `tema`, `genero`, `talla`, and `creado_por`. The results are:

_id	nombre	color	precio	marca	tema	genero	talla	creado_por
1	Objectid("...")	zapatos	negro	35.0				
2	Objectid("...")	puerta	blanco	30.0				
3	Objectid("...")	reloj	azul	20.0	tempex			
4	Objectid("...")	cuadro	azul	15.0		playa		
5	Objectid("...")	traje	negro	15.0		masculino	L	
6	Objectid("...")	florero	azul	10.0				aaa



## 6. Todos los productos que son azules o el precio es mayor que 50

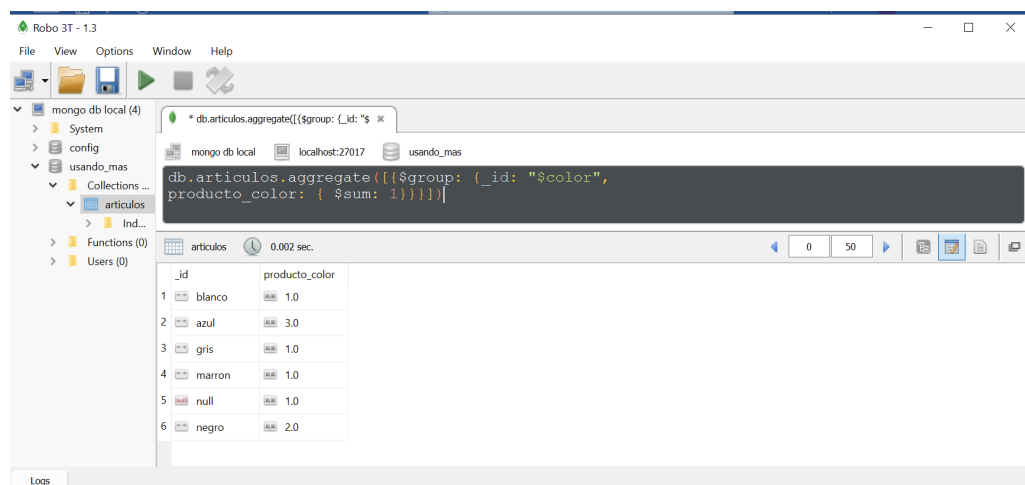
```
db.articulos.find({ $or:
  [
    {color:"azul"},
    {precio:{$gt:50}}
  ]
})
```



Listados solicitados por el director para el análisis de ventas:

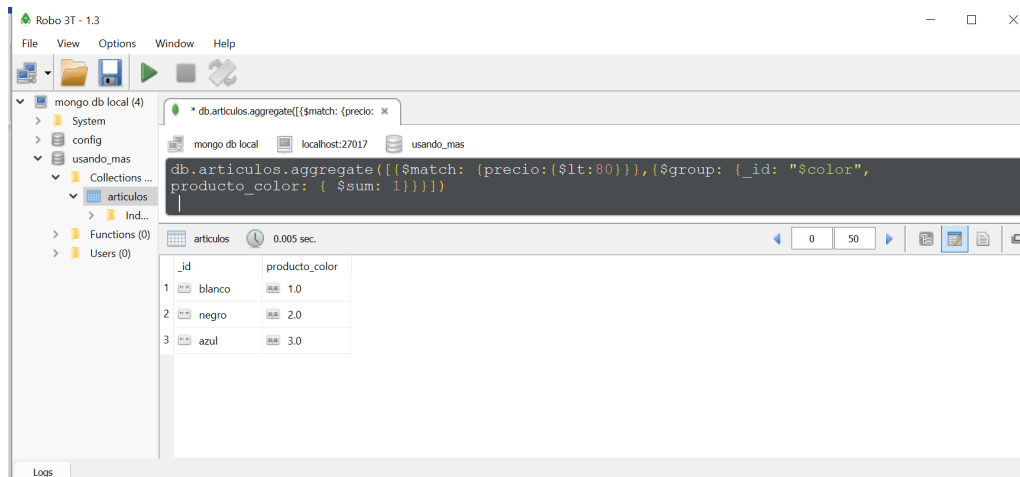
## 1. Cantidad de productos agrupados por color

```
db.articulos.aggregate([{$group: {_id: "$color", producto_color: { $sum: 1}}}]
```



## 2. Cantidad de productos agrupados por color con precio menor a 80

```
db.articulos.aggregate([{$match: {precio:{$lt:80}}},{ $group: { _id: "$color",  
producto_color: { $sum: 1}}}]])
```



The screenshot shows the Robo 3T 1.3 application window. The left sidebar displays the database structure: 'mongo db local (4)' containing 'System', 'config', 'usando\_mas', and 'articulos' collections. The main window shows the aggregation query: `db.articulos.aggregate([{$match: {precio:{$lt:80}}},{ $group: { _id: "$color", producto_color: { $sum: 1}}}]])`. Below the query, the results are displayed in a table with columns '\_id' and 'producto\_color'.

	_id	producto_color
1	blanco	1.0
2	negro	2.0
3	azul	3.0