

## Unidad 5. Gestión de bases de datos NoSQL



## Ejercicio MongoDB

23/01/2023

Juan Cebrián Pareja

2º DAM

IES Ramón del Valle-Inclán

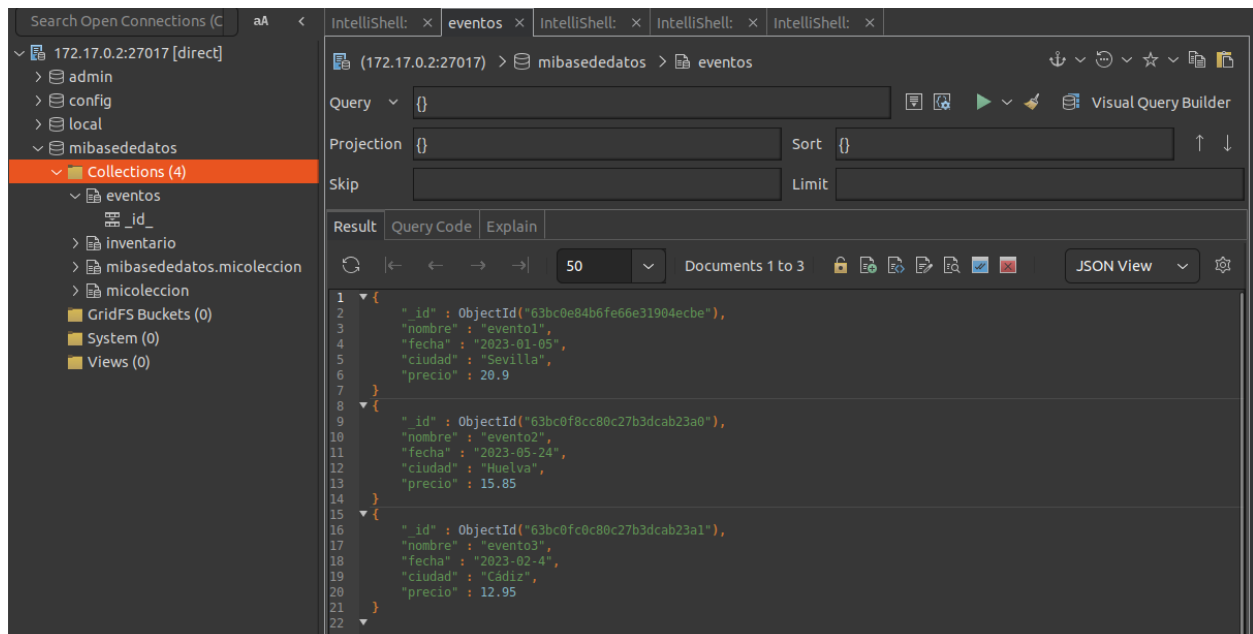
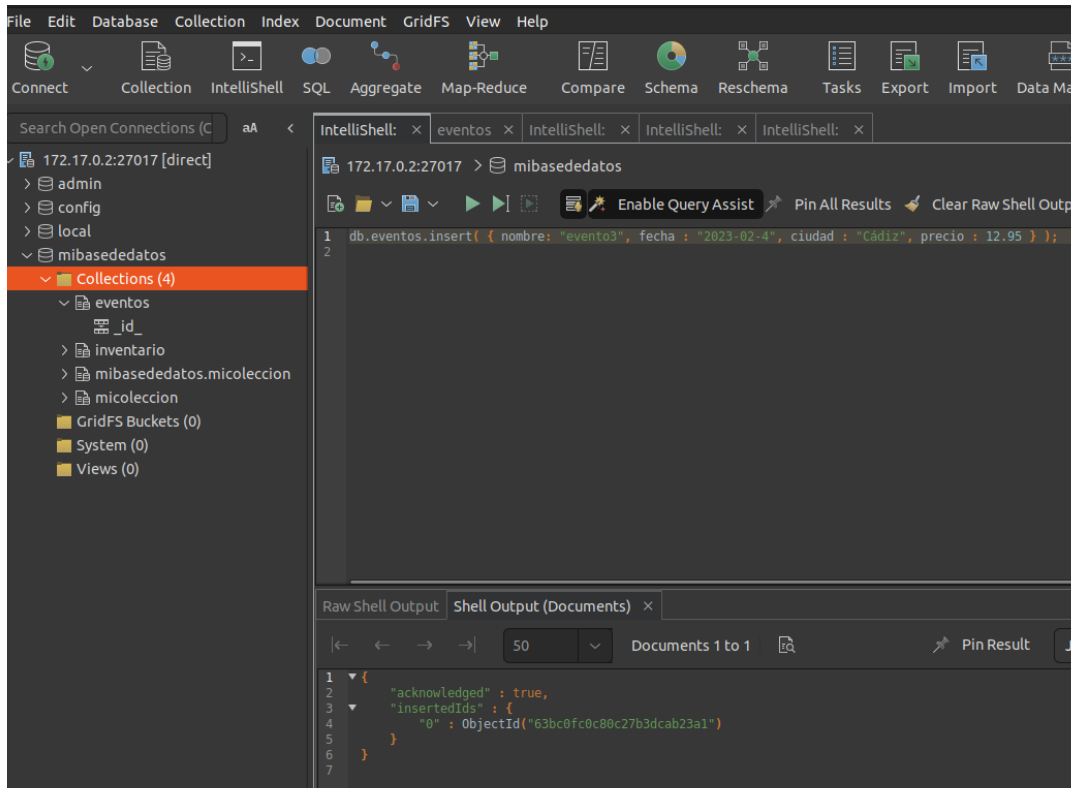
## Índice

<b>Requisitos previos</b>	<b>2</b>
<b>Consultas</b>	<b>4</b>
1. Mostrar eventos ordenados por precio ascendente.	4
2. Mostrar eventos de Sevilla y Huelva.	5
3. Contar colecciones.	6
<b>Agregaciones</b>	<b>7</b>
4. Obtener nombre y ciudad de los eventos en mayúsculas.	7
5. Obtener nombre de los eventos en mayúsculas y el precio con un descuento del 25%, redondeado con 2 decimales.	8
<b>Actualizar documentos</b>	<b>9</b>
6. Cambiar la fecha del evento cuya ciudad es Cádiz al 2023-02-10.	9
7. Añadir el campo tipo con valor "Concierto" a todos los eventos.	11
<b>Borrar documentos</b>	<b>13</b>
8. Borrar los eventos cuyo precio sea 15€ o menos.	13
9. Borrar todas las colecciones.	15
<b>Bibliografía</b>	<b>16</b>

## Requisitos previos

Inserta los siguientes documentos en una colección llamada eventos:

```
{
  "nombre" : "evento1",
  "fecha" : "2023-01-05",
  "ciudad" : "Sevilla",
  "precio" : 20.9,
}
{
  "nombre" : "evento2",
  "fecha" : "2023-05-24",
  "ciudad" : "Huelva",
  "precio" : 15.85,
}
{
  "nombre" : "evento3",
  "fecha" : "2023-02-04",
  "ciudad" : "Cádiz",
  "precio" : 12.95,
}
```



## Consultas

### 1. Mostrar eventos ordenados por precio ascendente.

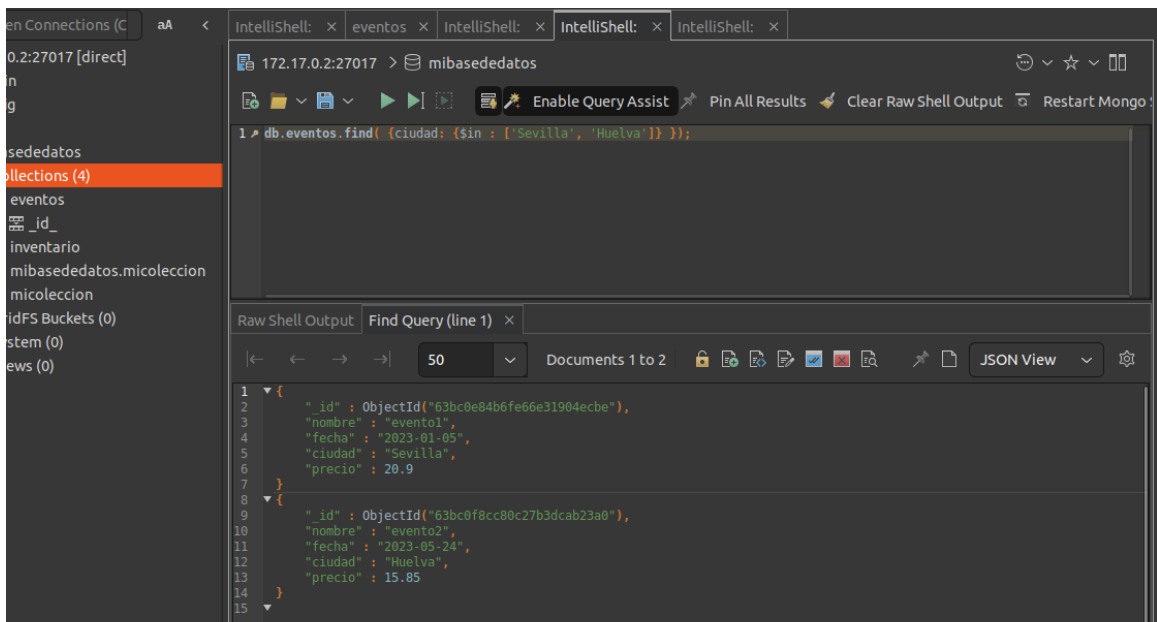
```
db.eventos.find().sort({precio:1});
```

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible, with the 'eventos' collection selected under the 'mibasededatos' database. The main panel displays the query `db.eventos.find().sort({precio:1});` in the IntelliShell. Below the query, the 'Find Query (line 1)' tab shows the results in JSON format, sorted by price in ascending order. The results are as follows:

Index	Document
1	<code>{ "_id" : ObjectId("63bc0fc0c80c27b3dcab23a1"), "nombre" : "evento3", "fecha" : "2023-02-4", "ciudad" : "Cádiz", "precio" : 12.95 }</code>
2	<code>{ "_id" : ObjectId("63bc0f8cc80c27b3dcab23a0"), "nombre" : "evento2", "fecha" : "2023-05-24", "ciudad" : "Huelva", "precio" : 15.85 }</code>
3	<code>{ "_id" : ObjectId("63bc0e84b6fe66e31904ecbe"), "nombre" : "evento1", "fecha" : "2023-01-05", "ciudad" : "Sevilla", "precio" : 20.9 }</code>

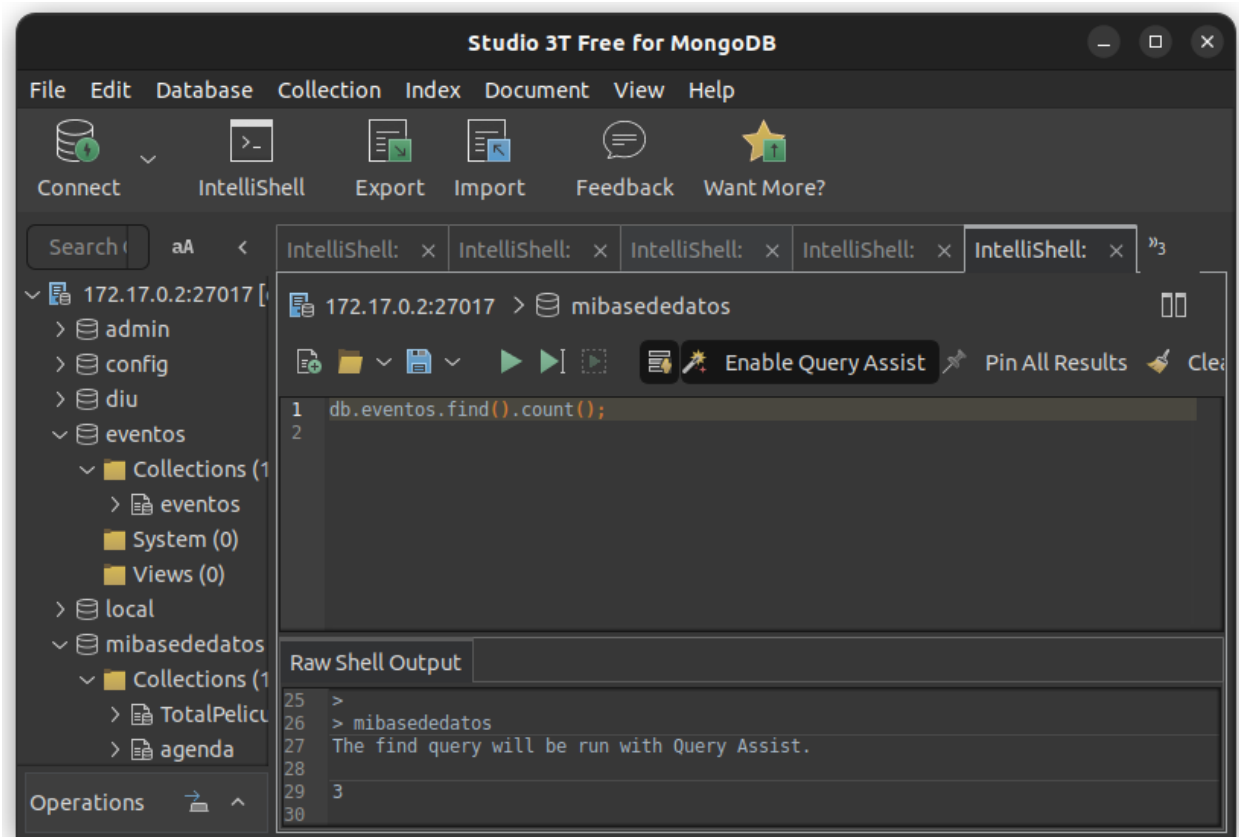
## 2. Mostrar eventos de Sevilla y Huelva.

```
db.eventos.find( {ciudad: ["Sevilla", "Huelva"]} );
```



### 3. Contar colecciones.

```
db.eventos.find().count();
```



## Agregaciones

### 4. Obtener nombre y ciudad de los eventos en mayúsculas.

```
db.eventos.aggregate([ {$project: { nombre: { $toUpper: "$nombre" },
                                ciudad: { $toUpper: "$ciudad" }
                              }
                        ] )
```

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible, with the 'eventos' collection selected under the 'mibasededatos' database. The main panel displays the following aggregate query in the IntelliShell:

```
1 db.eventos.aggregate([ {$project: { nombre: { $toUpper: "$nombre" },
2                               ciudad: { $toUpper: "$ciudad" }
3                               }
4                               }
5                               ] )
6
```

Below the query editor, the 'Raw Shell Output' tab is active, showing the results of the query. The results are displayed as a JSON array of three documents, each representing an event with its ID, name (in uppercase), and city (in uppercase):

```
1 {
2   "_id" : ObjectId("63bc0e84b6fe66e31904ecbe"),
3   "nombre" : "EVENT01",
4   "ciudad" : "SEVILLA"
5 }
6 {
7   "_id" : ObjectId("63bc0f8cc80c27b3dcab23a0"),
8   "nombre" : "EVENT02",
9   "ciudad" : "HUELVA"
10 }
11 {
12   "_id" : ObjectId("63dcf7495187409182e6863c"),
13   "nombre" : "EVENT03",
14   "ciudad" : "CÁDIZ"
15 }
16
```



5. Obtener nombre de los eventos en mayúsculas y el precio con un descuento del 25%, redondeado con 2 decimales.

```
db.eventos.aggregate([ {$project:
    { nombre: { $toUpper: "$nombre" },
      descuento: { $round:
        [{$multiply : ["$precio", 0.75]}, 2] }
    }
  } ] )
```

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible, with the 'eventos' collection selected under the 'mibasededatos' database. The main panel displays the following aggregate query in the IntelliShell:

```
1 db.eventos.aggregate([ {$project:
2   { nombre: { $toUpper: "$nombre" },
3     descuento: { $round:
4       [{$multiply : ["$precio", 0.75]}, 2] }
5   }
6 ] )
```

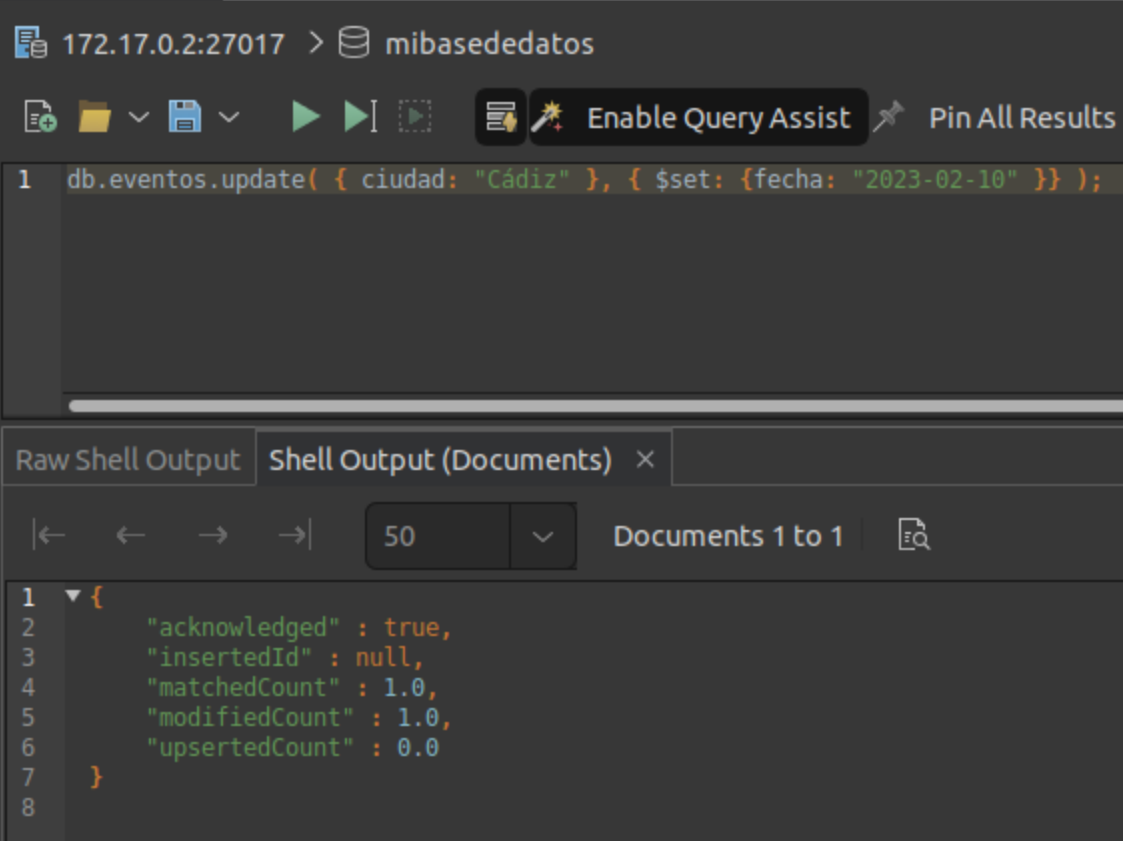
Below the query, the 'Aggregate Query (line 1)' tab shows the results in JSON View. The results are three documents, each containing an '\_id', a 'nombre' in uppercase, and a 'descuento' value rounded to two decimal places.

_id	nombre	descuento
ObjectId("63bc0e84b6fe66e31904ecbe")	EVENT01	15.67
ObjectId("63bc0f8cc80c27b3dcab23a0")	EVENT02	11.89
ObjectId("63dcf7495187409182e6863c")	EVENT03	9.71

## Actualizar documentos

6. Cambiar la fecha del evento cuya ciudad es Cádiz al 2023-02-10.

```
db.meventos.update( {ciudad: "Cádiz" }, { $set: {fecha: "2023-02-10" }} );
```



The screenshot shows a MongoDB Shell interface with the following components:

- Header:** Connection string `172.17.0.2:27017` and database name `mibasededatos`.
- Toolbar:** Includes icons for file operations, a play button, and buttons for `Enable Query Assist` and `Pin All Results`.
- Command Editor:** Line 1 contains the command `db.eventos.update( { ciudad: "Cádiz" }, { $set: { fecha: "2023-02-10" } } );`.
- Output Panel:** Switched to `Shell Output (Documents)`. It shows a single document with the following fields:

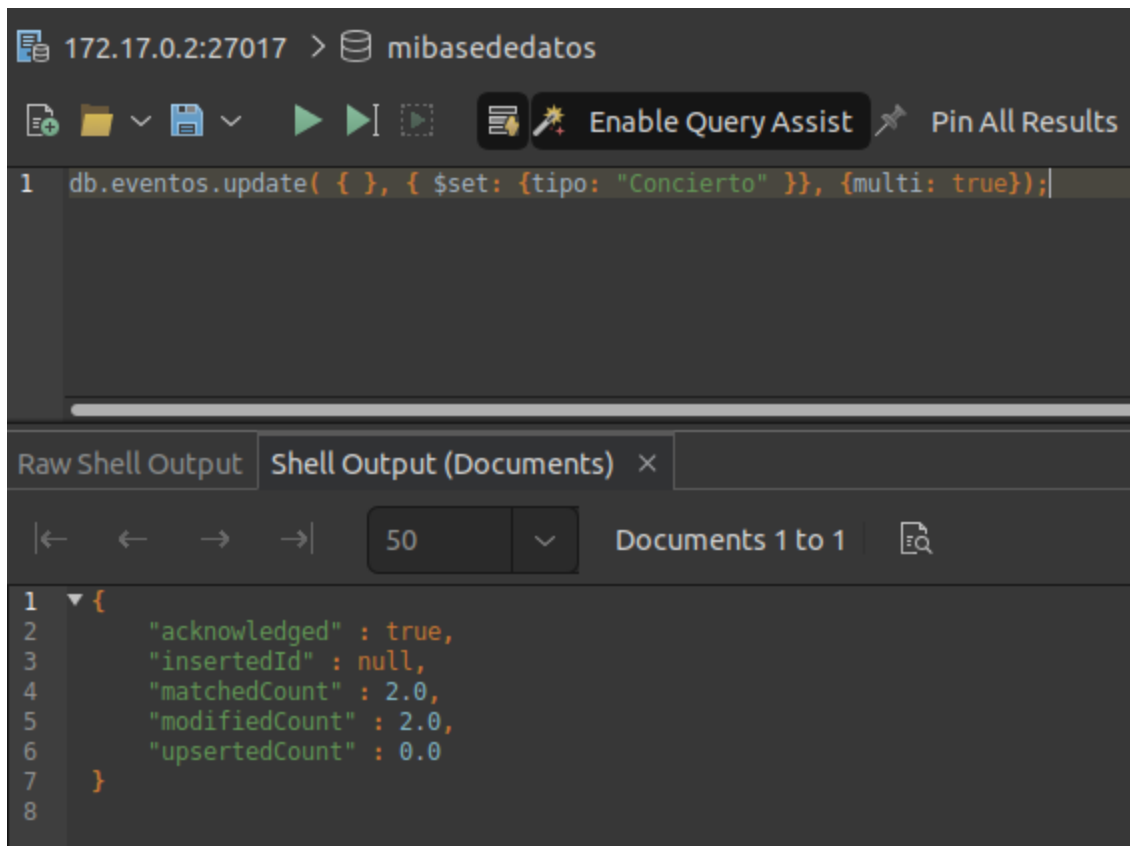
```
{
  "acknowledged" : true,
  "insertedId" : null,
  "matchedCount" : 1.0,
  "modifiedCount" : 1.0,
  "upsertedCount" : 0.0
}
```

Colección modificada:

```
14 }  
15 {  
16   "_id" : ObjectId("63bc0fc0c80c27b3dcab23a1"),  
17   "nombre" : "evento3",  
18   "fecha" : "2023-02-10",  
19   "ciudad" : "Cádiz",  
20   "precio" : 12.95  
21 }  
22
```

## 7. Añadir el campo tipo con valor "Concierto" a todos los eventos.

```
db.eventos.update( {}, { $set: {tipo: "Concierto" }}, {multi:true});
```



The screenshot shows a MongoDB Shell interface with the following components:

- Header:** 172.17.0.2:27017 > mibasededatos
- Toolbar:** Includes icons for file operations, a play button, and buttons for "Enable Query Assist" and "Pin All Results".
- Command Line:** The command `db.eventos.update( {}, { $set: {tipo: "Concierto" }}, {multi: true});` is entered on line 1.
- Output Tabs:** Two tabs are visible: "Raw Shell Output" and "Shell Output (Documents)". The "Shell Output (Documents)" tab is active.
- Output Content:** The output shows a single document with the following fields:

```
{
  "acknowledged" : true,
  "insertedId" : null,
  "matchedCount" : 2.0,
  "modifiedCount" : 2.0,
  "upsertedCount" : 0.0
}
```
- Navigation:** Below the output, there are navigation arrows, a page size selector set to 50, and a document range indicator showing "Documents 1 to 1".

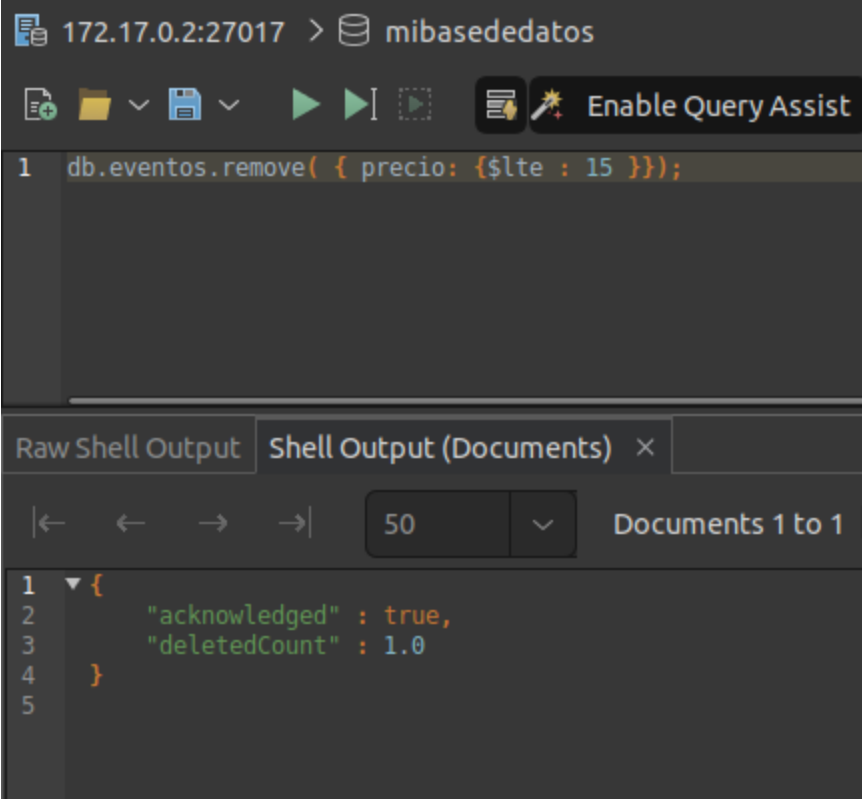
Eventos modificados:

```
1  ▼ {
2    "_id" : ObjectId("63bc0e84b6fe66e31904ecbe"),
3    "nombre" : "evento1",
4    "fecha" : "2023-01-05",
5    "ciudad" : "Sevilla",
6    "precio" : 20.9,
7    "tipo" : "Concierto"
8  }
9  ▼ {
10   "_id" : ObjectId("63bc0f8cc80c27b3dcab23a0"),
11   "nombre" : "evento2",
12   "fecha" : "2023-05-24",
13   "ciudad" : "Huelva",
14   "precio" : 15.85,
15   "tipo" : "Concierto"
16 }
17 ▼
```

## Borrar documentos

8. Borrar los eventos cuyo precio sea 15€ o menos.

```
db.eventos.remove( { precio: { $lte : 15 } } );
```



The screenshot shows a MongoDB Shell interface. The top bar indicates the connection to 172.17.0.2:27017 and the database mibasededatos. The command `db.eventos.remove( { precio: { $lte : 15 } } );` is entered in the command line. Below the command line, the 'Shell Output (Documents)' tab is active, displaying the result of the operation: `{ "acknowledged" : true, "deletedCount" : 1.0 }`. The interface includes navigation buttons and a document count of 1.

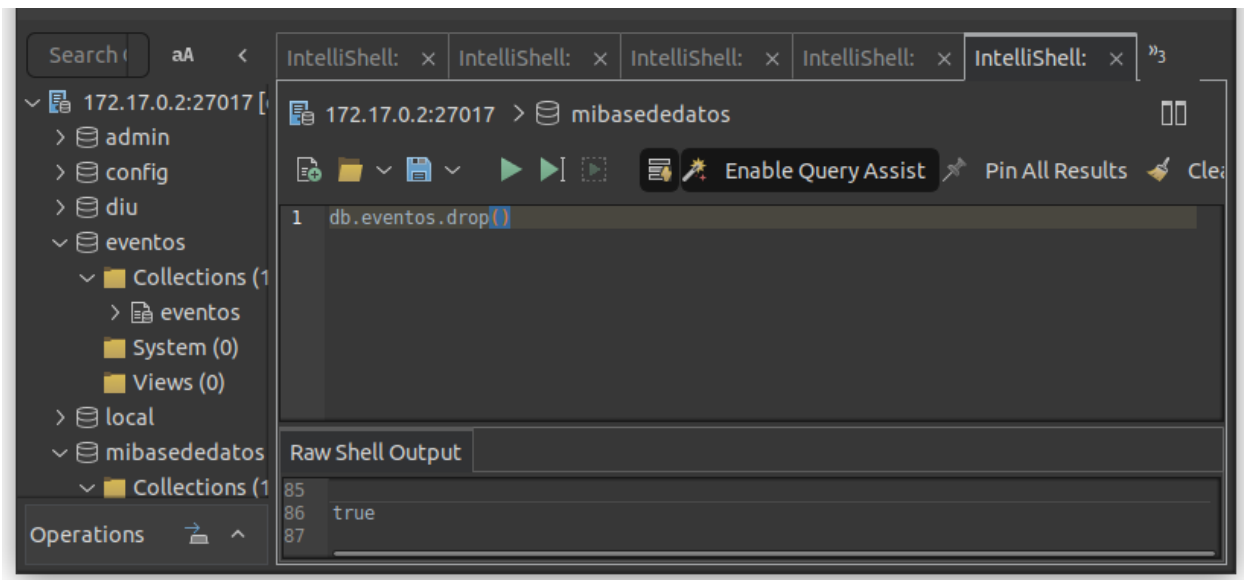
```
172.17.0.2:27017 > mibasededatos  
1 db.eventos.remove( { precio: { $lte : 15 } } );  
  
Raw Shell Output | Shell Output (Documents) ×  
|← ← → →| 50 Documents 1 to 1  
1 {  
2   "acknowledged" : true,  
3   "deletedCount" : 1.0  
4 }  
5
```

Ya no aparecen eventos cuyo precio sea 15€ o menos:

```
1  {
2    "nombre" : "evento1",
3    "fecha" : "2023-01-05",
4    "ciudad" : "Sevilla",
5    "precio" : 20.9
6  }
7
8  {
9    "_id" : ObjectId("63bc0f8cc80c27b3dcab23a0"),
10   "nombre" : "evento2",
11   "fecha" : "2023-05-24",
12   "ciudad" : "Huelva",
13   "precio" : 15.85
14 }
15
```

## 9. Borrar todas las colecciones.

```
db.eventos.drop();
```





### Bibliografía

- [Apuntes de Clase \(Moodle\)](#)
- [Web Tutorial MongoDB](#)