



PRACTICA GESTIÓN DE BIBLIOTECA EN MONGODB

Ing. Maximiliano Carsi Castrejón

DESCRIPCIÓN BREVE

Este documento trata sobre la gestión y manipulación de una base de datos a través de colecciones utilizando la herramienta de MongoDB.

Luis Eduardo Bahena Castillo

8°C IDyGS

INTRODUCCIÓN

Objetivo: Diseñar, crear y manipular una base de datos en MongoDB para gestionar una biblioteca. Los estudiantes deberán resolver problemas relacionados con la gestión de libros, préstamos, usuarios y otras actividades típicas de una biblioteca.

Instrucciones:

1. Creación de la Base de Datos:

- Diseñar el esquema de la base de datos en MongoDB que permita gestionar una biblioteca.
- Crear una nueva base de datos en MongoDB y las colecciones necesarias según el diseño del esquema.

2. Inserción de Datos:

- Insertar datos de ejemplo en las colecciones creadas para simular el funcionamiento de la biblioteca.
- Los datos de ejemplo deben incluir libros, usuarios, préstamos, etc.

3. Consultas:

- Realizar consultas a la base de datos para obtener información relevante sobre la biblioteca.
- Ejemplos de consultas incluyen:
 - Consultar todos los libros disponibles.
 - Encontrar todos los préstamos realizados por un usuario específico.
 - Buscar libros por título, autor, categoría, etc.

4. Registros, Modificaciones y Eliminaciones:

- Realizar registros de nuevos libros, usuarios, préstamos, etc.
- Modificar información de libros, usuarios, etc., según sea necesario.
- Eliminar registros obsoletos o duplicados.

5. Documentación:

- Preparar un informe que incluya:
 - Descripción del esquema de la base de datos diseñada.
 - Detalles de las colecciones creadas y los datos insertados.
 - Ejemplos de consultas realizadas y sus resultados.
 - Detalles de registros, modificaciones y eliminaciones realizadas.

- Conclusiones y reflexiones sobre el proceso y los resultados obtenidos.

Criterios de Evaluación:

- Claridad y coherencia en el diseño de la base de datos para gestionar una biblioteca.
- Correcta utilización de consultas para obtener información relevante sobre los libros, préstamos, usuarios, etc.
- Eficiencia en la manipulación de datos (registros, modificaciones, eliminaciones) en el contexto de una biblioteca.
- Calidad y detalle de la documentación proporcionada en el informe.

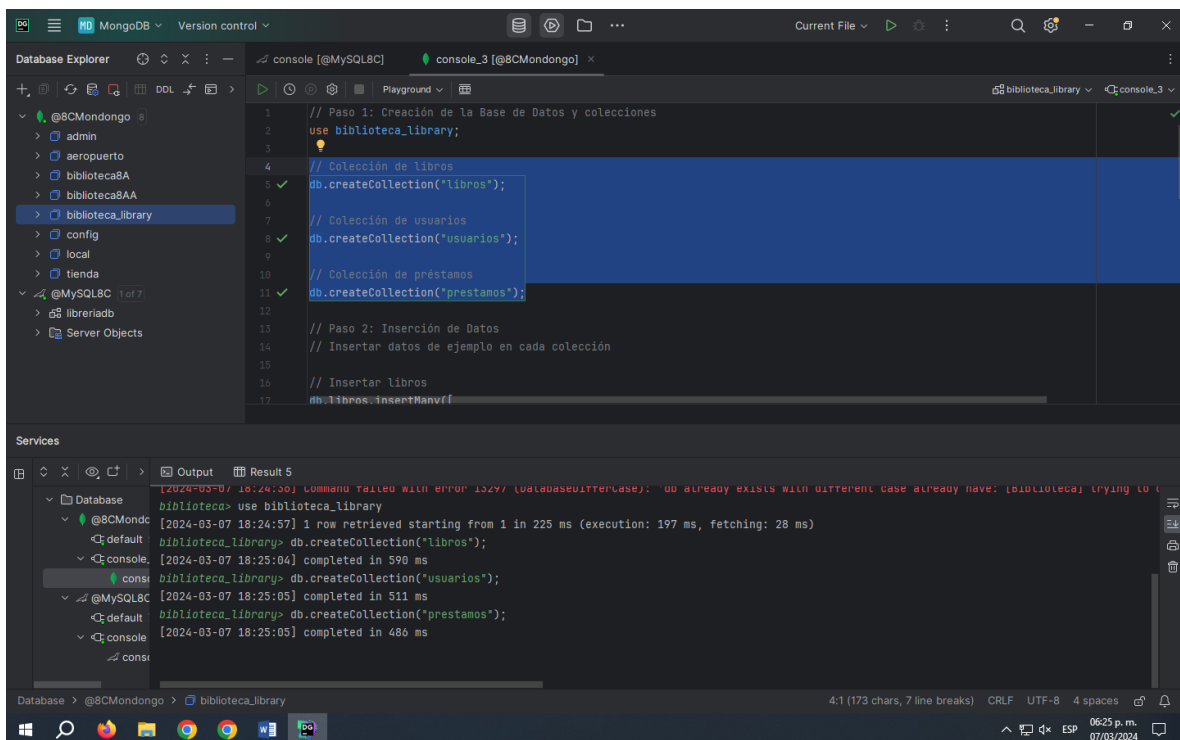
DESARROLLO

¿Qué es una gestión en base de datos en MongoDB? La gestión de base de datos en MongoDB implica organizar, almacenar y manipular datos de manera eficiente y segura utilizando sus características de almacenamiento flexible y consultas poderosas.

Antes de hacer la práctica ... ¡¡¡¡¡ IMPORTANTE !!!!!

Ya no será necesario detallar cada paso básico, puesto que hasta este momento ya se sabe todo lo que se va a realizar (Ejemplo como abrir aplicaciones, descargar o crear repositorios, descargar documentos, crear variables, conectar bases de datos entre otros).

Paso 1: Abrir la aplicación DataGrip para acceder a la base de datos en **MongoDB** y con los siguientes comandos en pantalla, usar la base de datos **biblioteca_library** y crear 3 colecciones.



The screenshot shows the MongoDB DataGrip interface. The left sidebar displays the 'Database Explorer' with a tree view of databases and collections. The 'biblioteca_library' database is selected. The main editor area shows a SQL script with the following commands:

```
// Paso 1: Creación de la Base de Datos y colecciones
use biblioteca_library;

// Colección de libros
db.createCollection("libros");

// Colección de usuarios
db.createCollection("usuarios");

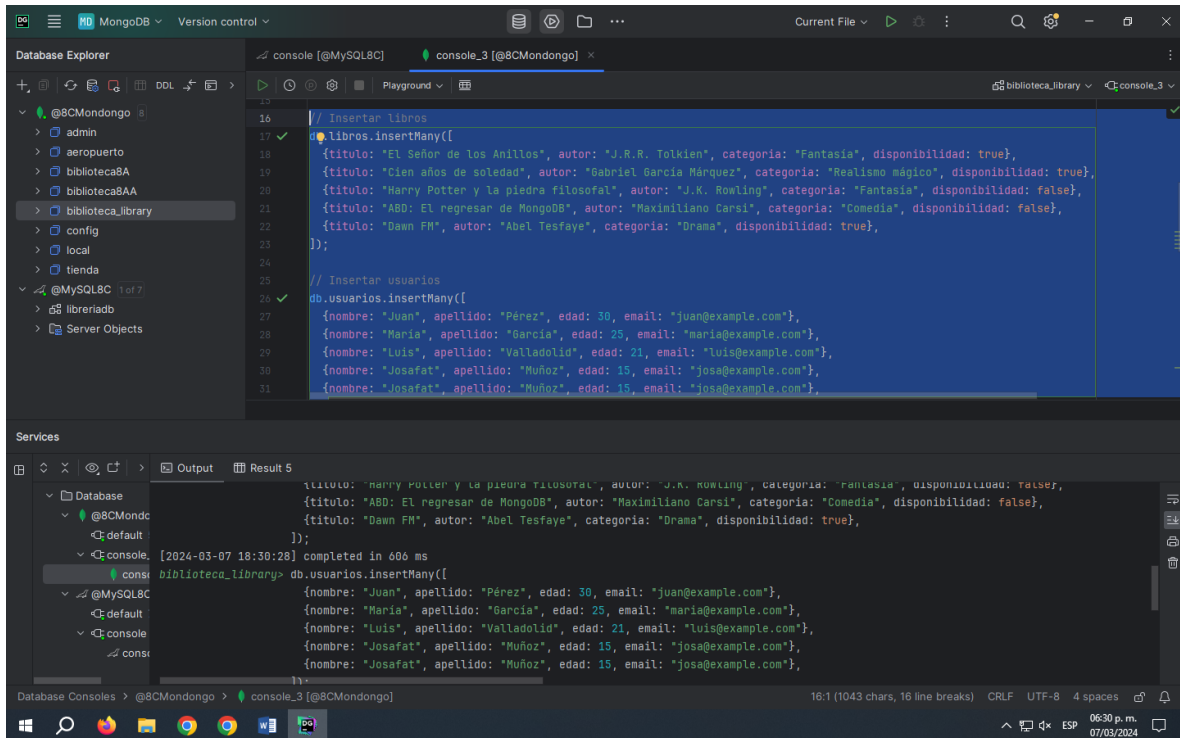
// Colección de préstamos
db.createCollection("prestamos");

// Paso 2: Inserción de Datos
// Insertar datos de ejemplo en cada colección
// Insertar libros
db.libros.insertMany([
```

The bottom panel shows the 'Output' window with the following log entries:

```
[2024-03-07 18:24:30] Command failed with error 1327 (DatabaseAlreadyExist): db already exists with different case already have: [biblioteca] trying to
biblioteca> use biblioteca_library
[2024-03-07 18:24:57] 1 row retrieved starting from 1 in 225 ms (execution: 197 ms, fetching: 28 ms)
biblioteca_library> db.createCollection("libros");
[2024-03-07 18:25:04] completed in 590 ms
const biblioteca_library> db.createCollection("usuarios");
[2024-03-07 18:25:05] completed in 511 ms
biblioteca_library> db.createCollection("prestamos");
[2024-03-07 18:25:05] completed in 486 ms
```

Paso 2: Ejecutar los inserts de cada colección para verificar y manipular las conexiones de datos de las colecciones **libros** y **autores**.



```

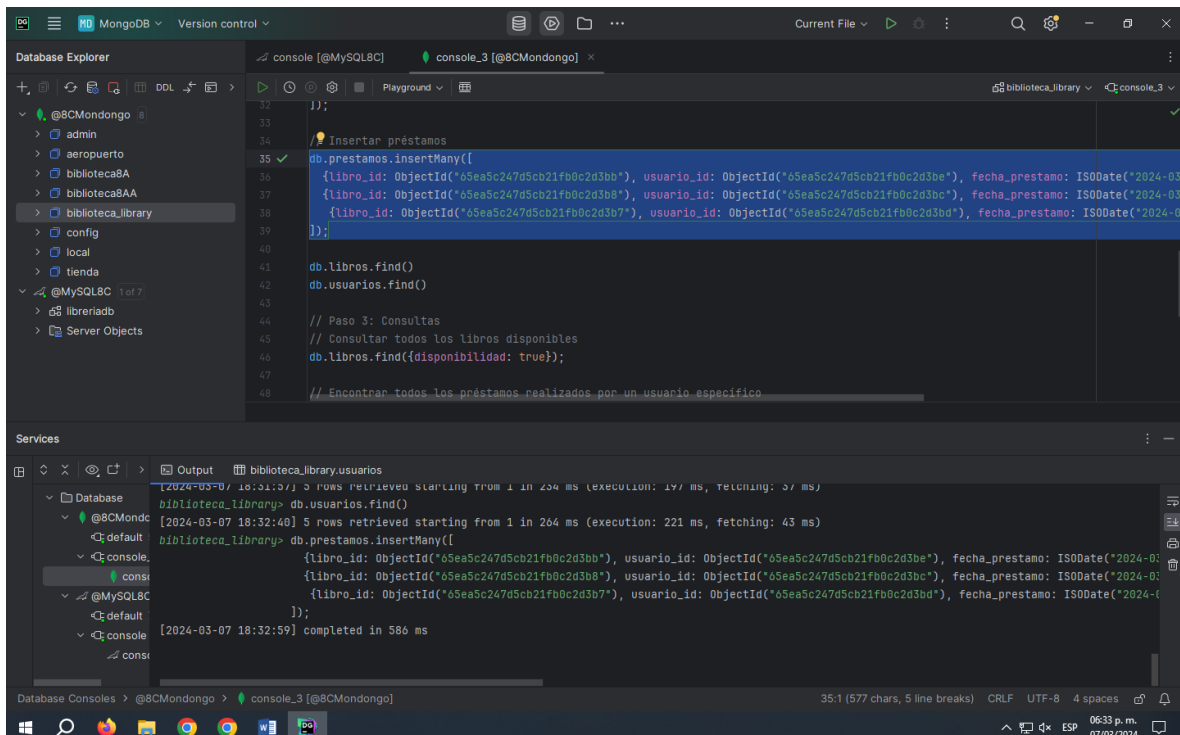
16 // Insertar libros
17 db.libros.insertMany([
18   {titulo: "El Señor de los Anillos", autor: "J.R.R. Tolkien", categoria: "Fantasia", disponibilidad: true},
19   {titulo: "Cien años de soledad", autor: "Gabriel García Márquez", categoria: "Realismo mágico", disponibilidad: true},
20   {titulo: "Harry Potter y la piedra filosofal", autor: "J.K. Rowling", categoria: "Fantasia", disponibilidad: false},
21   {titulo: "ABD: El regresar de MongoDB", autor: "Maximiliano Carisi", categoria: "Comedia", disponibilidad: false},
22   {titulo: "Dawn FM", autor: "Abel Tesfaye", categoria: "Drama", disponibilidad: true},
23 ]);
24
25 // Insertar usuarios
26 db.usuarios.insertMany([
27   {nombre: "Juan", apellido: "Pérez", edad: 30, email: "juan@example.com"},
28   {nombre: "Maria", apellido: "García", edad: 25, email: "maria@example.com"},
29   {nombre: "Luis", apellido: "Valladolid", edad: 21, email: "luis@example.com"},
30   {nombre: "Josafat", apellido: "Muñoz", edad: 15, email: "josa@example.com"},
31   {nombre: "Josafat", apellido: "Muñoz", edad: 15, email: "josa@example.com"},
32 ]);
  
```

Output:

```

{
  "titulo": "Harry Potter y la piedra filosofal", "autor": "J.K. Rowling", "categoria": "Fantasia", "disponibilidad": false,
  "titulo": "ABD: El regresar de MongoDB", "autor": "Maximiliano Carisi", "categoria": "Comedia", "disponibilidad": false,
  "titulo": "Dawn FM", "autor": "Abel Tesfaye", "categoria": "Drama", "disponibilidad": true,
}
  
```

Paso 3: Hacer lo mismo del paso anterior pero ahora con la colección **préstamos**.



```

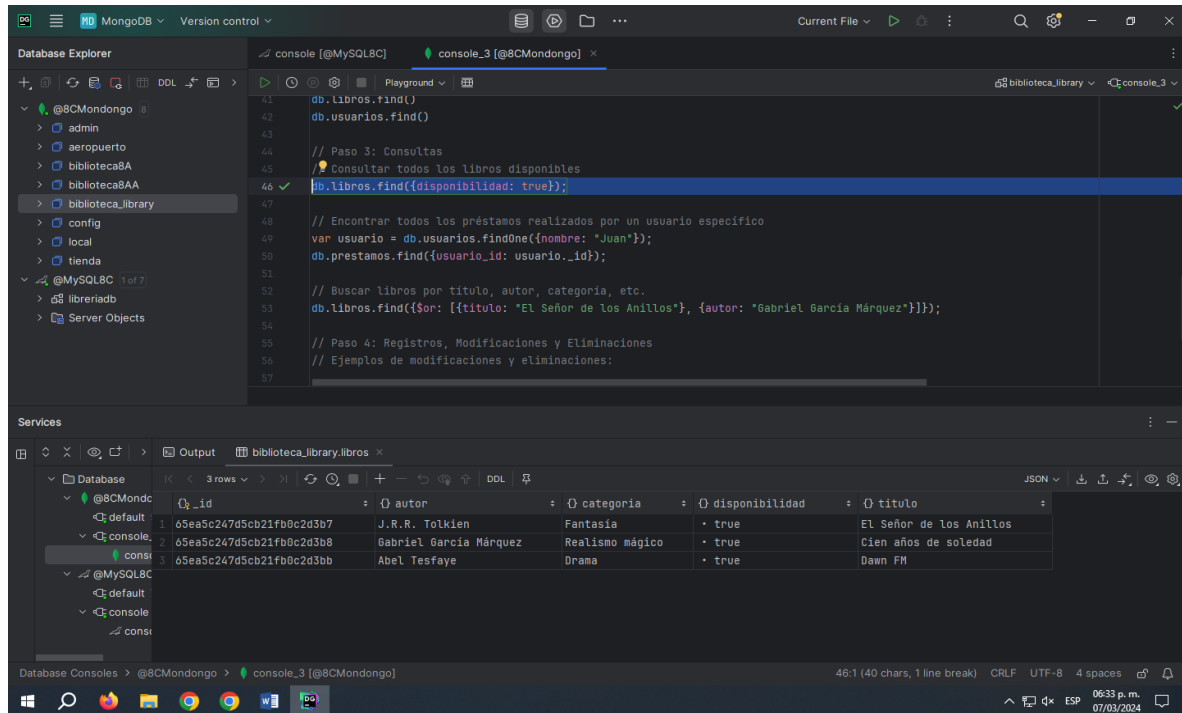
32
33
34 // Insertar préstamos
35 db.prestamos.insertMany([
36   {libro_id: ObjectId("65ea5c247d5cb21fb0c2d3bb"), usuario_id: ObjectId("65ea5c247d5cb21fb0c2d3be"), fecha_prestamo: ISODate("2024-03-07T18:32:40Z")},
37   {libro_id: ObjectId("65ea5c247d5cb21fb0c2d3bb"), usuario_id: ObjectId("65ea5c247d5cb21fb0c2d3bc"), fecha_prestamo: ISODate("2024-03-07T18:32:40Z")},
38   {libro_id: ObjectId("65ea5c247d5cb21fb0c2d3b7"), usuario_id: ObjectId("65ea5c247d5cb21fb0c2d3bd"), fecha_prestamo: ISODate("2024-03-07T18:32:40Z")},
39 ]);
40
41 db.libros.find()
42 db.usuarios.find()
43
44 // Paso 3: Consultas
45 // Consultar todos los libros disponibles
46 db.libros.find({disponibilidad: true});
47
48 // Encontrar todos los préstamos realizados por un usuario específico
  
```

Output:

```

biblioteca_library.usuarios
{ "_id": "65ea5c247d5cb21fb0c2d3bb", "nombre": "Juan", "apellido": "Pérez", "edad": 30, "email": "juan@example.com" }
{ "_id": "65ea5c247d5cb21fb0c2d3be", "nombre": "Maria", "apellido": "García", "edad": 25, "email": "maria@example.com" }
{ "_id": "65ea5c247d5cb21fb0c2d3bc", "nombre": "Luis", "apellido": "Valladolid", "edad": 21, "email": "luis@example.com" }
{ "_id": "65ea5c247d5cb21fb0c2d3bd", "nombre": "Josafat", "apellido": "Muñoz", "edad": 15, "email": "josa@example.com" }
{ "_id": "65ea5c247d5cb21fb0c2d3bd", "nombre": "Josafat", "apellido": "Muñoz", "edad": 15, "email": "josa@example.com" }
  
```

Paso 4: Ahora empezaremos a dar gestión a las colecciones, iniciaremos buscando los libros disponibles con comando **find**.



```

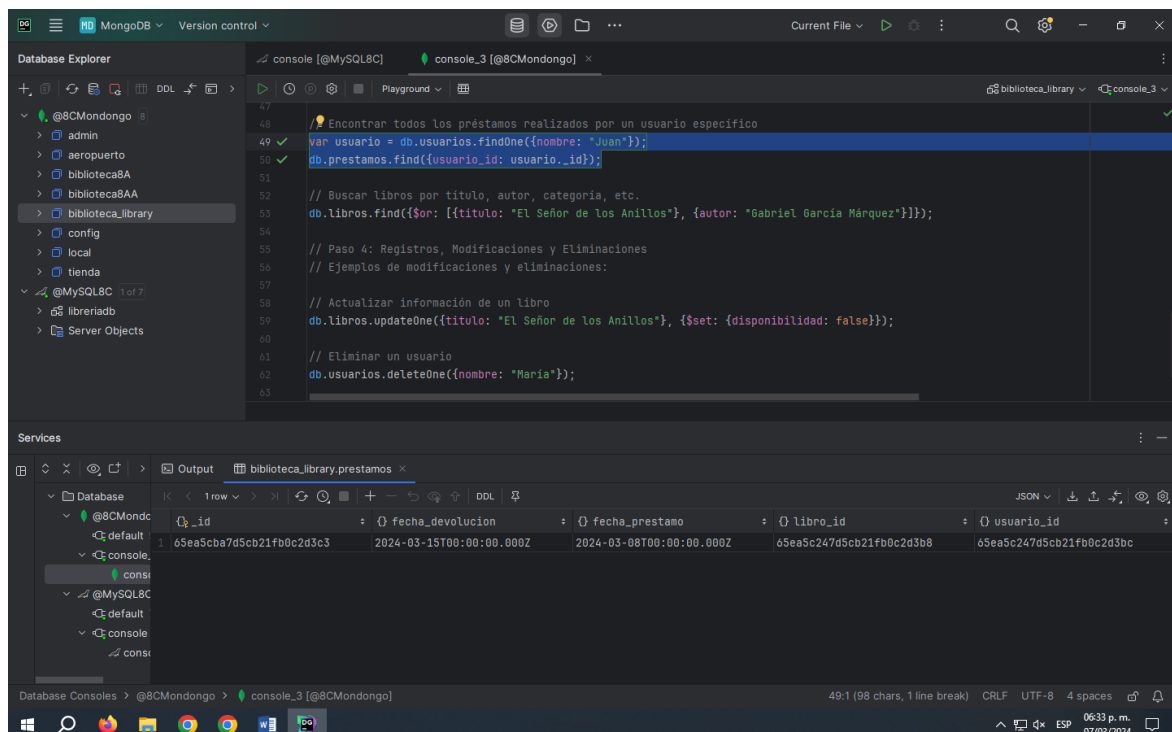
41 db.libros.find()
42 db.usuarios.find()
43
44 // Paso 3: Consultas
45 // Consultar todos los libros disponibles
46 db.libros.find({disponibilidad: true});
47
48 // Encontrar todos los préstamos realizados por un usuario específico
49 var usuario = db.usuarios.findOne({nombre: "Juan"});
50 db.prestamos.find({usuario_id: usuario._id});
51
52 // Buscar libros por título, autor, categoría, etc.
53 db.libros.find({$or: [{titulo: "El Señor de los Anillos"}, {autor: "Gabriel García Márquez"}]});
54
55 // Paso 4: Registros, Modificaciones y Eliminaciones
56 // Ejemplos de modificaciones y eliminaciones:
57

```

Output: biblioteca_library.libros

_id	autor	categoría	disponibilidad	título
65ea5c247d5cb21fb0c2d3b7	J.R.R. Tolkien	Fantasia	true	El Señor de los Anillos
65ea5c247d5cb21fb0c2d3b8	Gabriel García Márquez	Realismo mágico	true	Cien años de soledad
65ea5c247d5cb21fb0c2d3bb	Abel Tesfaye	Drama	true	Dawn FM

Paso 5: Después vamos a declarar una variable usuario para poder buscar por el nombre, de acuerdo al siguiente comando en la colección de **préstamos**.



```

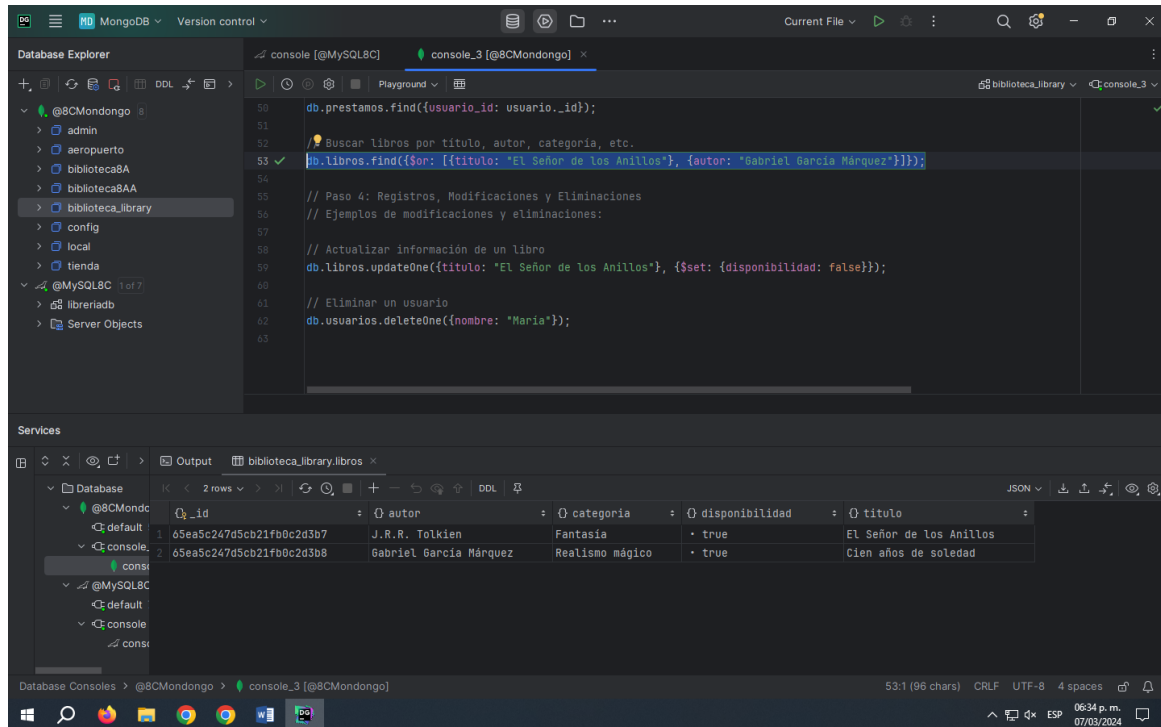
47
48 // Encontrar todos los préstamos realizados por un usuario específico
49 var usuario = db.usuarios.findOne({nombre: "Juan"});
50 db.prestamos.find({usuario_id: usuario._id});
51
52 // Buscar libros por título, autor, categoría, etc.
53 db.libros.find({$or: [{titulo: "El Señor de los Anillos"}, {autor: "Gabriel García Márquez"}]});
54
55 // Paso 4: Registros, Modificaciones y Eliminaciones
56 // Ejemplos de modificaciones y eliminaciones:
57
58 // Actualizar información de un libro
59 db.libros.updateOne({titulo: "El Señor de los Anillos"}, {$set: {disponibilidad: false}});
60
61 // Eliminar un usuario
62 db.usuarios.deleteOne({nombre: "Maria"});
63

```

Output: biblioteca_library.prestamos

_id	fecha_devolucion	fecha_prestamo	libro_id	usuario_id
65ea5c247d5cb21fb0c2d3c3	2024-03-15T00:00:00.000Z	2024-03-08T00:00:00.000Z	65ea5c247d5cb21fb0c2d3b8	65ea5c247d5cb21fb0c2d3bc

Paso 6: Siguiendo con el comando anterior **find**, ahora buscar a partir de títulos en la colección de **libros** (como se muestra en la siguiente pantalla):



The screenshot shows the MongoDB console with the following code in the console_3 window:

```

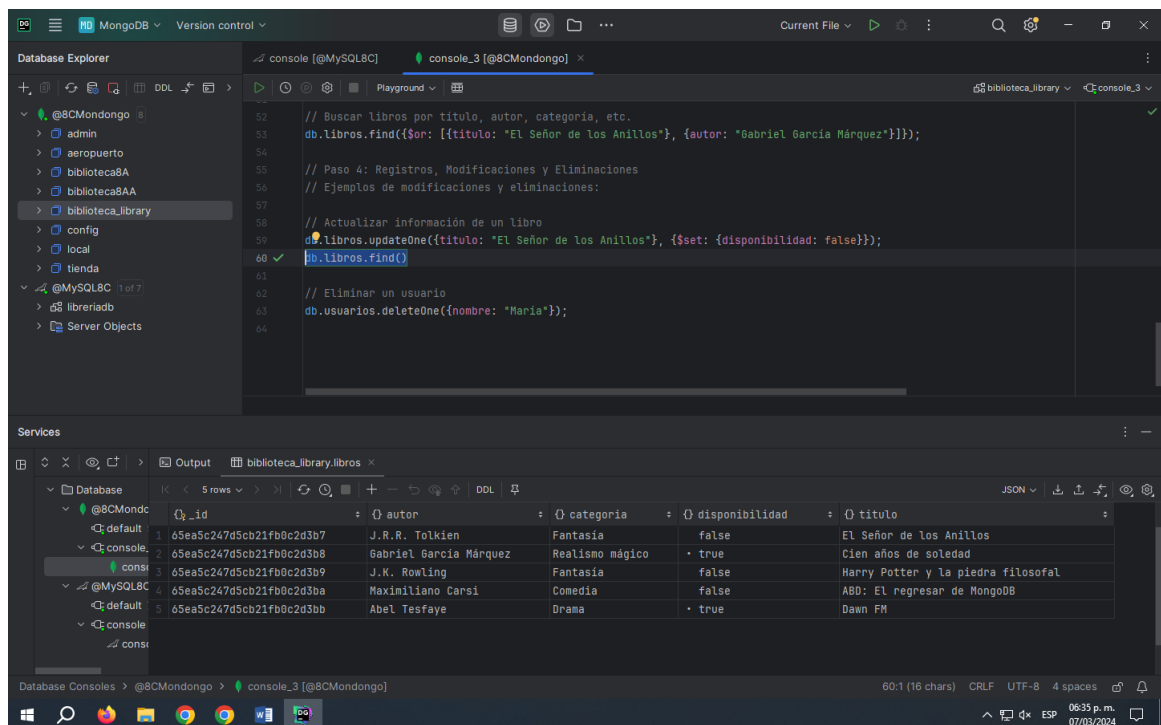
50 db.prestamos.find({usuario_id: usuario_id});
51
52 // Buscar libros por título, autor, categoría, etc.
53 db.libros.find({$or: [{titulo: "El Señor de los Anillos"}, {autor: "Gabriel García Márquez"}]});
54
55 // Paso 4: Registros, Modificaciones y Eliminaciones
56 // Ejemplos de modificaciones y eliminaciones:
57
58 // Actualizar información de un libro
59 db.libros.updateOne({titulo: "El Señor de los Anillos"}, {$set: {disponibilidad: false}});
60
61 // Eliminar un usuario
62 db.usuarios.deleteOne({nombre: "Maria"});
63

```

The output window shows the results of the find query:

_id	autor	categoría	disponibilidad	título
65ea5c247d5cb21fb0c2d3b7	J.R.R. Tolkien	Fantasia	true	El Señor de los Anillos
65ea5c247d5cb21fb0c2d3b8	Gabriel García Márquez	Realismo mágico	true	Cien años de soledad

Paso 7: Ahora con comando **updateOne**, cambiar el status de disponibilidad **false** a un registro de la colección **libros** (como se muestra en la siguiente pantalla):



The screenshot shows the MongoDB console with the following code in the console_3 window:

```

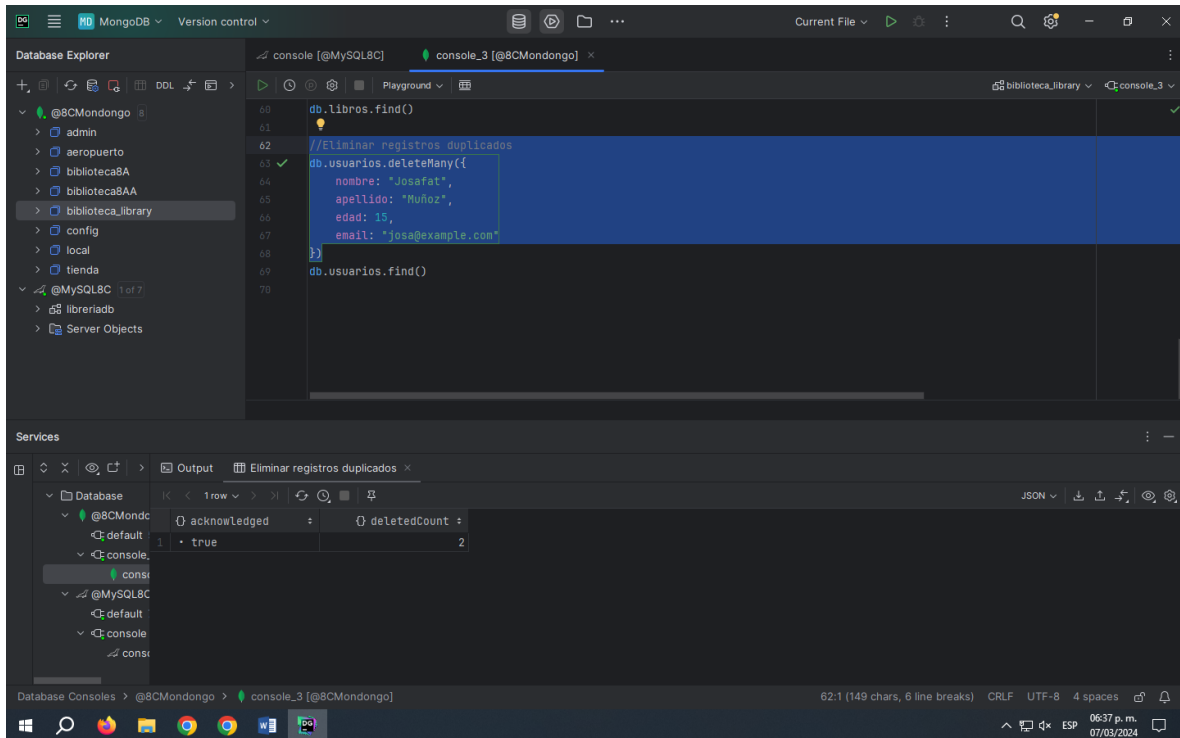
52 // Buscar libros por título, autor, categoría, etc.
53 db.libros.find({$or: [{titulo: "El Señor de los Anillos"}, {autor: "Gabriel García Márquez"}]});
54
55 // Paso 4: Registros, Modificaciones y Eliminaciones
56 // Ejemplos de modificaciones y eliminaciones:
57
58 // Actualizar información de un libro
59 db.libros.updateOne({titulo: "El Señor de los Anillos"}, {$set: {disponibilidad: false}});
60 db.libros.find()
61
62 // Eliminar un usuario
63 db.usuarios.deleteOne({nombre: "Maria"});
64

```

The output window shows the results of the find query after the update:

_id	autor	categoría	disponibilidad	título
65ea5c247d5cb21fb0c2d3b7	J.R.R. Tolkien	Fantasia	false	El Señor de los Anillos
65ea5c247d5cb21fb0c2d3b8	Gabriel García Márquez	Realismo mágico	true	Cien años de soledad
65ea5c247d5cb21fb0c2d3b9	J.K. Rowling	Fantasia	false	Harry Potten y la piedra filosofal
65ea5c247d5cb21fb0c2d3ba	Maximiliano Carsi	Comedia	false	ABD: EL regresar de MongoDB
65ea5c247d5cb21fb0c2d3bb	Abel Tesfaye	Drama	true	Dawn FM

Paso 8: Por último, para eliminar los registros duplicados, con comando **deleteMany** especificar el o los campos a eliminar y realizará la acción en la colección de **usuarios**.



The screenshot shows the MongoDB Studio interface. In the left sidebar, the 'Database Explorer' shows the 'biblioteca_library' database selected. The main editor shows a JavaScript console with the following code:

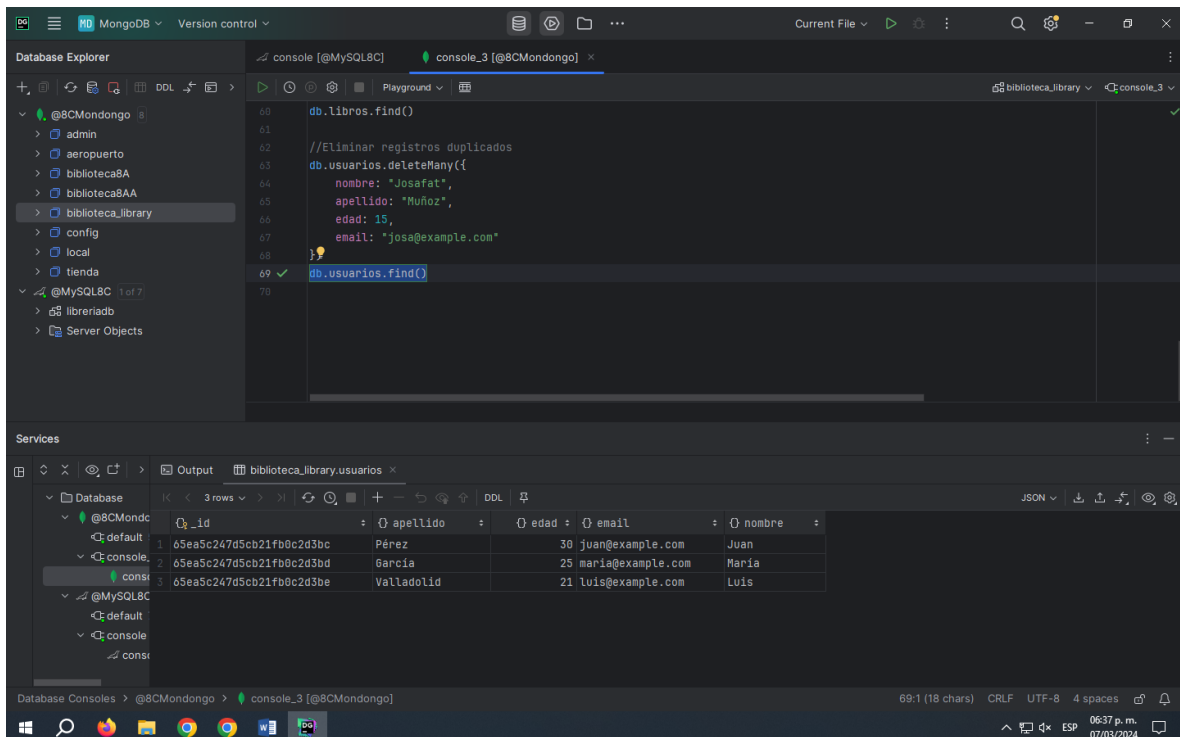
```

60 db.libros.find()
61
62 //Eliminar registros duplicados
63 db.usuarios.deleteMany({
64   nombre: "Josafat",
65   apellido: "Muñoz",
66   edad: 15,
67   email: "josa@example.com"
68 })
69 db.usuarios.find()
70
  
```

The 'Output' pane at the bottom shows the result of the command:

acknowledged	deletedCount
true	2

Paso 9: Podemos ver como se eliminaron los registros duplicados del usuario Josafat.



The screenshot shows the MongoDB Studio interface. In the left sidebar, the 'Database Explorer' shows the 'biblioteca_library' database selected. The main editor shows the same JavaScript console as in the previous step, but the 'Output' pane now shows the result of the 'db.usuarios.find()' command:

_id	apellido	edad	email	nombre
65ea5c247d5cb21fb0c2d3bc	Pérez	30	juan@example.com	Juan
65ea5c247d5cb21fb0c2d3bd	García	25	maria@example.com	Maria
65ea5c247d5cb21fb0c2d3be	Valladolid	21	luis@example.com	Luis

CONCLUSIÓN

La gestión de base de datos en MongoDB abarca una serie de prácticas y procesos clave para garantizar un almacenamiento eficiente, seguro y escalable de datos. Al aprovechar su modelo de datos flexible basado en documentos, MongoDB ofrece una gran versatilidad en la representación y manipulación de la información. Su capacidad para manejar grandes volúmenes de datos y distribuirlos en clústeres facilita el crecimiento y la escalabilidad de las aplicaciones.

La gestión de bases de datos en MongoDB incluye la creación y administración de colecciones, índices y usuarios, así como la implementación de estrategias de respaldo y recuperación de datos para garantizar la integridad y disponibilidad de la información. La optimización de consultas y la afinación del rendimiento son aspectos importantes de la gestión de bases de datos en MongoDB para garantizar un acceso rápido y eficiente a los datos.

Además, la gestión de bases de datos en MongoDB implica la implementación de medidas de seguridad robustas, como la autenticación y la autorización, para proteger los datos sensibles contra accesos no autorizados y amenazas cibernéticas. En resumen, una gestión eficaz de la base de datos en MongoDB es esencial para mantener el buen funcionamiento y la seguridad de las aplicaciones que dependen de ella.

