**TP6 – COMANDOS PARA LA ADMINISTRACIÓN DE LAS BASES DE DATOS**

**1. APLICAR LOS SIGUIENTES COMANDOS**

**CREAR UN USUARIO**

use admin

db.createUser({"user":"admin","pwd":"pass12345","roles":[{"role":"root","db":"admin"}]}) -->

Ha devuelto el siguiente objeto JSON: { ok: 1 }

db.getUsers() → Devuelve un JSON con toda la información del usuario.

db.auth("admin","pass12345") → Devuelve el siguiente objeto JSON: { ok: 1 }

**CONFIGURAR LA INTERFAZ PARA INICIAR LA AUTENTICACIÓN**

Para la configuración debemos editar el mencionado archivo y buscar la sección denominada: #security y luego escribir el siguiente texto:

security:

autorization: “enabled”

**LUEGO**, REINICIAMOS MONGODB, en Windows:

Abrir CMD:

net stop MongoDB

net start MongoDB

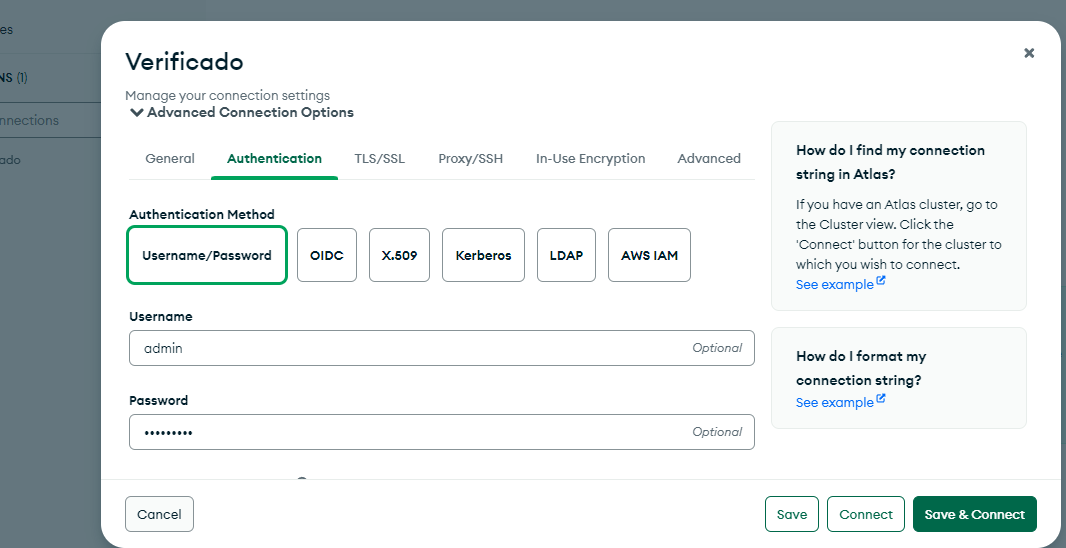
LUEGO DE CONFIGURAR, EN LA TERMINAL ESCRIBIMOS

use admin

db.auth(“admin”, “pass12345”)

**Y ESTAMOS AUTENTICADOS, ES DECIR, PODEMOS USAR COMANDOS MONGODB CON NORMALIDAD**

**LO MISMO SUCEDE EN LA INTERFAZ, DEBEMOS CREAR UNA NUEVA CONEXION CON LOS DATOS VERIFICADOS:**



**\*\*He notado que al borrar la conexión y crear una nueva, con metodos de autenticacion, no me eliminó los datos de la base de datos**

**Ejercicio Usuarios Pasos**

Crear dos usuarios con diferentes roles y verificar que cada uno tiene los permisos que le hemos otorgado.

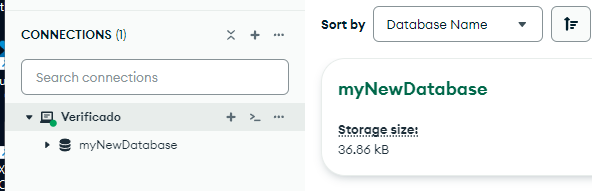
1. primero : use admin

2. creamos primer usuario lector: db.createUser({"user":"lector","pwd":"lector","roles":[{"role":"read","db":"myNewDatabase"}]}) → nos da {ok:1}

3. creamos segundo usuario con permisos de lectura escritura: db.createUser({"user":"editor","pwd":"editor","roles":[{"role":"readWrite","db":"myNewDatabase"}]})

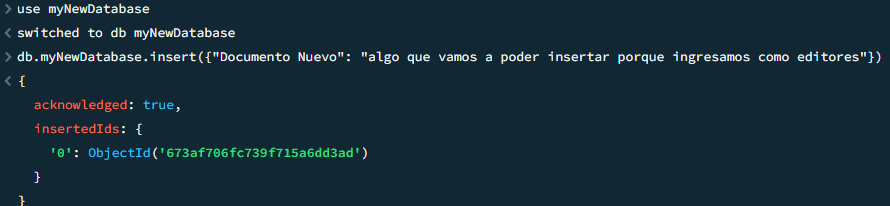
4. autenticar como lector: db.auth("lector","lector")

5. verificamos que EL LECTOR solo tiene permisos de lectura:



6. autenticar como editor: db.auth(“editor”,”editor”)

7. verificar que EL EDITOR tiene permisos de RW:



8. Listo!!!



Para eliminar un usuario, se debe utilizar el siguiente comando:

db.dropUser("lector")

**VEMOS QUE EL USUARIO LECTOR NO EXISTE MAS**

**CREAR ÍNDICES DE BÚSQUEDA**

db.getCollection("libros").createIndex({"Title":1}) → para mejorar los tiempos de respuesta



**db.getCollection("myCollection").createIndex(**{"ISBN":1},{unique:true})



**AHORA TODOS LOS LIBROS TIENEN UN ISBN UNICO, EXCEPTO UNO QUE ES NULO, LO QUE SIGNIFICA QUE SERA EL UNICO ISBN NULO**

**AHORA ELIMINAMOS INDICE, RECORDEMOS QUE SU NOMBRE ES ISBN\_1**



**C**ómo crear un índice compuesto por dos campos:

**db.getCollection("myCollection").createIndex({"Title":1,"Autor":1})**



**2. Estudiar y aplicar comandos de MongoDB para las siguientes actividades:**

* Carga masiva de documentos, comando bulk
* Realizar copias de seguridad
* Realizar restauraciones a partir de un backup
* Exportación e Importación de colecciones
* Propiedades de Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID) para
* transacciones.

ENTREGABLE

Los grupos deben entregar un documento con la descripción de los comandos investigados,

describiendo la sintaxis y los resultados obtenidos.

### **Informe: Comandos de MongoDB para Actividades Específicas**

**CARGA MASIVA DE DOCUMENTOS, COMANDO BULK:**

Provee operaciones para escribir documentos en una misma coleccion. db.collection.bulkWrite() provee operaciones en bloque para insertar, actualizar y eliminar.

**Operaciones desordenadas vs ordenadas**

Pueden ser operaciones ordenadas o desordenadas. Mongo ejecuta las operaciones en serie por defecto, y si hay un error en alguna operación, no ejecuta las siguientes operaciones. Cuando son desordenadas, Mongo puede ejecutar las operaciones en paralelo y continua aunque alguna tenga un error. Por defecto, las operaciones son ordenadas. Para especificar el tipo de orden, se escribe ‘ordered : false’

bulkWrite() metodos → insertOne, updateOne, updateMany, replaceOne, deleteOne, deleteMany

#### **Sintaxis**

db.<collection>.bulkWrite([

{ insertOne: { document: { key1: "value1", key2: "value2" } } },

{ updateOne: { filter: { key1: "value1" }, update: { $set: { key2: "newValue" } } } },

{ deleteOne: { filter: { key1: "value1" } } }

])

Este comando se divide en 4 partes, la llamada a bulkWrite, la operación de inserción, la operación de actualización y la operación de eliminación.

#### **Resultados**



* Inserta múltiples documentos, actualiza o elimina según las operaciones definidas.
* Retorna un resumen indicando el número de operaciones exitosas y fallidas.

**Ejemplo: queremos hacer lo siguiente:**

1. Agregar dos documentos usando insertOne (insertar dos libros)

2. Actualizar un documento usando updateOne (modificar un libro)

3. Eliminar un documento usando deleteOne() (borrar un libro)

4. Reemplazar un documento usando replaceOne (reemplazar un libro)

try{db.myCollection.bulkWrite([

{insertOne:{document: { "Title": "The Picture of Dorian Gray", "Author": "Oscar Wilde", "Categories": ["Fiction", "Classics"], "Stock": 12, "Edition": { "format": "Paperback", "Page": 352 }, "Language": "English", "PublishedDate": "2015-05-15", "ISBN": 9780141439570 } }},

{insertOne:{ document: { "Title": "The Adventures of Sherlock Holmes", "Author": "Arthur Conan Doyle", "Categories": ["Mystery", "Detective Fiction", "Classics"], "Stock": 20, "Edition": { "format": "Hardcover", "Page": 340 }, "Language": "English", "PublishedDate": "2018-03-10", "ISBN": 9780192834923 } }},

{updateOne:{

filter: {“ISBN”: 141185139},

update: {$set:{“Stock”:32}},

upsert: false

}},

{deleteOne:{filter: { “ISBN”: 141182679 } }},

{replaceOne:{

filter: {"\_id": ObjectId('673bb74015550d1e879d590c')},

replacement: { "Title": "The Hound of the Baskervilles", "Author": "Arthur Conan Doyle", "Categories": [ "Mystery", "Crime", "Detective Fiction", "Classics" ], "Stock": 15, "Edition": { "format": "Paperback", "Page": 256 }, "Language": "English", "PublishedDate": "2020-06-10", "ISBN": 9781853260332 },

{upsert: false }

}}

]) }catch(error){

print(error)}

**OPERACION EXITOSA**: agregados ‘The Picture of Dorian Gray’ y ‘The Adventures of Sherlock Holmes’, actualizado ‘The Plague’ a Stock: 32, eliminado ‘On The Road’ y reemplazado un libro por ‘The Hound of the Baskervilles’



Para ejecutar los comandos a continuación, es necesario realizar la instalación de Mongo Tools y agregarlo a las variables de entorno del sistema.

**Pasos para utilizar MONGODB TOOLS:**

* **Descargar MongoDB Tools (Windows):**

Dirígete a la página de descargas oficial de MongoDB Tools:

<https://www.mongodb.com/try/download/database-tools>

Asegúrate de seleccionar la versión adecuada según tu sistema operativo y la arquitectura.

Descarga el archivo .msi (installer).

* **Instalar MongoDB Tools:**

Ejecuta el archivo .msi descargado.

Sigue las instrucciones del asistente de instalación.

Una vez instalado, abre la carpeta “bin” en la ruta donde se instaló MongoDB Tools.

Copia la ruta (ej. C:\Program Files\MongoDB\Tools\100\bin”)

* **Agrega MongoDB Tools al PATH (variables de entorno del sistema)**

Haz click derecho en “Mi equipo”.

Ve a propiedades.

Haz click en Configuración avanzada del Sistema.

Haz click en Variables de Entorno.

Haz doble click en PATH.

Haz click en nuevo y pega la dirección de MongoDB Tools.

Haz click en aceptar, luego aceptar y luego reinicia tu computadora.

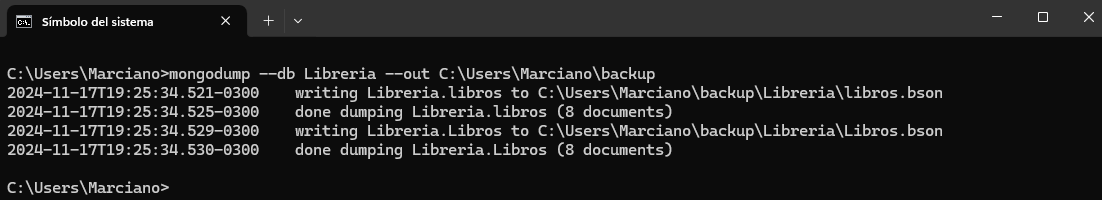
**2. Realizar Copias de Seguridad**

#### **Descripción**

MongoDB permite crear backups mediante el comando **mongodump**, que exporta una base de datos o colección en formato BSON.

#### **Sintaxis**

mongodump --db Libreria --out C:\Users\<TuUsuario>\Desktop\backup



este comando se ejecuta en la consola de Windows o el Bash de Linux, no dentro de compass

#### **Resultados**

* Genera un conjunto de archivos BSON y JSON para la base de datos especificada.
* Estos archivos se almacenan en la ubicación de destino definida por **backup\_directory**.

### **3. Restauración a Partir de un Backup**

#### **Descripción**

Para restaurar una base de datos desde un backup, se utiliza el comando **mongorestore**.

#### **Sintaxis**

mongorestore --db Libreria C:\Users\Marciano\backup/Libreria

#### 

#### **Resultados**

* Restaura los datos desde el archivo BSON al servidor MongoDB.
* Sobrescribe los datos existentes en la base de datos si hay conflicto de nombres.

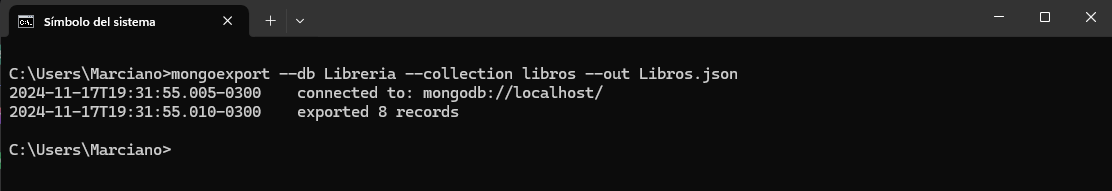
### **4. Exportación e Importación de Colecciones**

#### **Exportación**

Se utiliza el comando **mongoexport** para exportar datos en formato JSON o CSV.

**Sintaxis**

mongoexport --db Libreria --collection libros --out Libros.json



(en Windows) mongoexport exporta el json en la carpeta en la que se ejecuta el comando

**Resultados**

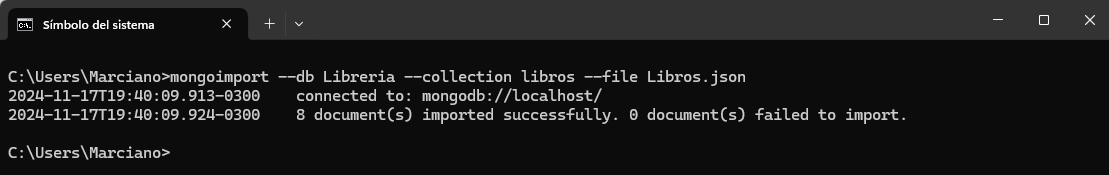
* Exporta los documentos de la colección en el formato especificado.

#### **Importación**

El comando **mongoimport** permite cargar los datos exportados a una base de datos.

**Sintaxis**

mongoimport --db Libreria --collection libros --file Libros.json



(en Windows) mongoimport también lee el archivo json desde la ruta donde se ejecuta el comando

**Resultados**

* Importa los documentos desde el archivo JSON a la colección especificada.

**5. Propiedades ACID para Transacciones**

#### **Descripción**

MongoDB soporta transacciones que cumplen con las **propiedades ACID**, asegurando confiabilidad en las operaciones. Estas se ejecutan con el cliente de MongoDB mediante sesiones.

#### **Sintaxis**

------

* Conexión: El código se conecta a MongoDB usando MongoClient.connect.
* Inicio de sesión y transacción: Se inicia una sesión con startSession y luego se inicia una transacción.
* Operaciones dentro de la transacción: Se insertan y actualizan documentos dentro de la transacción.
* Confirmación o reversión: Si todo va bien, se confirma la transacción con commitTransaction. Si hay un error, se revierte con abortTransaction.
* Cierre de la sesión: Finalmente, se cierra la sesión con endSession.

#### **Resultados**

* Garantiza la atomicidad de las operaciones: todas las acciones se completan o ninguna lo hace.
* Ofrece consistencia asegurando que la base de datos permanezca en un estado válido.
* Proporciona aislamiento para que las transacciones no interfieran entre sí.
* Los datos confirmados son duraderos, incluso ante fallos del sistema.