

	UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLOGICOS COORDINACION DE COMPUTACION
Ciclo I	BASE DE DATOS Guía de Laboratorio No. 11 Seguridad, Validaciones y Backups

I. RESULTADOS DE APRENDIZAJE

Que el estudiante:

- Aprenda a realizar validaciones de documentos, para que cada documento sea ingresado con el tipo de dato que corresponde a cada compi.
- Aprenda a crear usuarios con sus respectivos roles.

6. INTRODUCCIÓN

Crear colecciones con validaciones para ir comparando cada documento insertado o actualizando con los criterios especificados en validador.

Db.createCollection("NombreColeccion",{validator:{"condiciones"}})

Antes de crear un usuario debemos crear su respectivo rol para cada tipo de usuario que tendremos en nuestra base de mongo en se crean con la siguiente sintaxis:

Db.comand({createRole:"NombreRole",privileges:[db:"NombreBD",actions:"tipo_CRUD"}],Roles:[{role:"tipoprivilegio",db:"NombreBD"}])

Luego de crear el rol podemos crear nuestro usuario con la siguiente sintaxis:
Db.createUser({user:"NombreUsuario",pwd:"contraseña",roles:["Rol a asignarle a usuario"]})

Para consultar usuarios específico podemos usar: *Db.getUser("NombreUsuario");*

Para consulta todos los usuarios de la BD puede usar: *Db.getUsers();*

Cambiar contraseña: *db.changeUserPassword("NombreUsuario",Contraseña)*

II. MATERIALES Y EQUIPO

No.	Requerimientos	Cantidad
1	Memoria USB	1
2	PC con MongoDB y RoboMongo	1
3	Guía de laboratorio 10	1

III. PROCEDIMIENTO

PARTE 1: Creación de documentos con validación

1. Primero, debemos crear BD “Biblioteca”. Crear colección Productos e ingresar los siguientes documentos a partir de la siguiente tabla:

Nombre	biografía	Fecha de nacimiento	nacionalidad	titulo	paginas	Isbn
George R. R. Martin	American novelist...	1948-09-2	USA	Juego de Tronos	150	0-553-57340-4
George R. R. Martin	American novelist...	1948-09-2	USA	Tormenta de Espadas	620	0-553-57340-7
George R. R. Martin	American novelist...	1948-09-2	USA	Choque de Reyes	340	0-553-57340-5
Jane Austen	novelist...	1775-12-16	Gran Bretaña	Orgullo y Perjuicio	392	978-8497646932
Claudia Lars	poeta...	1899-12-20	Salvadoreña	Tierra de Infancia	120	9789992349199

2. crearemos las colecciones con validaciones de documentos.

Use Biblioteca

```
db.createCollection("Autor",
{validator:{$or:
[
  {nombre:{ $type: "string" }},
  {biografia:{ $type: "string" }},
  {fecha_de_nacimiento:{ $type: "date" }},
  {nacionalidad:{ $type: "string" }}
]
}})
```

```
db.createCollection('Libros',
{validator:
{$or:[
  {titulo:{ $type: 'string' }},
  {paginas:{ $type: 'int' }},
  {isbn:{ $type: 'string' }},
  {autor:[{ $type: 'objectId', $ref: 'Autor' }]}
]}})
```

3. Ingresaremos los dos primeros documentos a las colecciones respectivas, tomando en cuenta que trabajaremos con modelo relacional.

```
db.Autor.insert({
  "id": "AU001",
  "nombre": "George R. R. Martin",
  "biografia": "American novelist...",
  "fecha_de_nacimiento": ISODate("1948-09-02"),
  "nacionalidad": "USA"
})
```

```
db.Libros.insert({
  "titulo": "Juego de Tronos",
  "paginas": 150,
  "isbn": "0-553-57340-4",
  "autor": "AU001"
})
```

4. Termine de ingresar todos los datos que hacen falta a la colección.
5. Realice las respectivas consultas de búsqueda para libros y autores:

```
db.Autor.find({})
db.Libros.find({})
```

6. Como puedes observar la información se muestra solo por colección separada, si queremos mostrar todas la información como que si fuera un solo documento se deberá realizar la siguiente consulta:

```
db.Libros.aggregate([
  {
    $lookup:
    {from: "Autor",
     localField: "autor",
     foreignField: "id",
     as: "informacion_autor"
    }
  }
])
```

7. Mostrar libros por nombre de autor ordenado en forma ascendente.

```
db.Libros.aggregate([
  {
    $lookup:
    {from: "Autor",
```

```

        localField:"autor",
        foreignField:"id",
        as: "informacion_autor"
    }
},
{
    $sort:{"informacion_autor.nombre":1}
})

```

8. Cantidad de libros registrados por autor:

```

db.Libros.aggregate([
{
    $lookup:
        {from:"Autor",
        localField:"autor",
        foreignField:"id",
        as: "informacion_autor"
        }
},
{
    $group:{
        _id : "informacion_autor.name" ,
        count: { $sum: 1 }
    }
})

```

9. Nombre de libros por autor:

```

db.Libros.aggregate([
{
    $lookup:
        {
            from:"Autor",
            localField:"autor",
            foreignField:"_id",
            as:"información_autor"
        }
},
{
    $group:{_id:"información_autor.nombre",
    title:{$push:"$titulo"}}
})

```

10. Mostrar libro con mayor número de páginas.

```
db.Libros.aggregate([
{
$lookup:
{
from:"Autor",
localField:"autor",
foreignField:"id",
as:"informacion_autor"
}},
{
$group:{_id:{id:"informacion_autor.nombre",
Titulo_Libro:"$titulo"},
mayor:{$max:"$paginas"}},
{$sort:{"mayor":1}},
{$limit:1}
})
```

PARTE 2: Creación de usuarios en MongoDB

11. Abrir una nueva consola, pero usando la base de datos **Admin**.

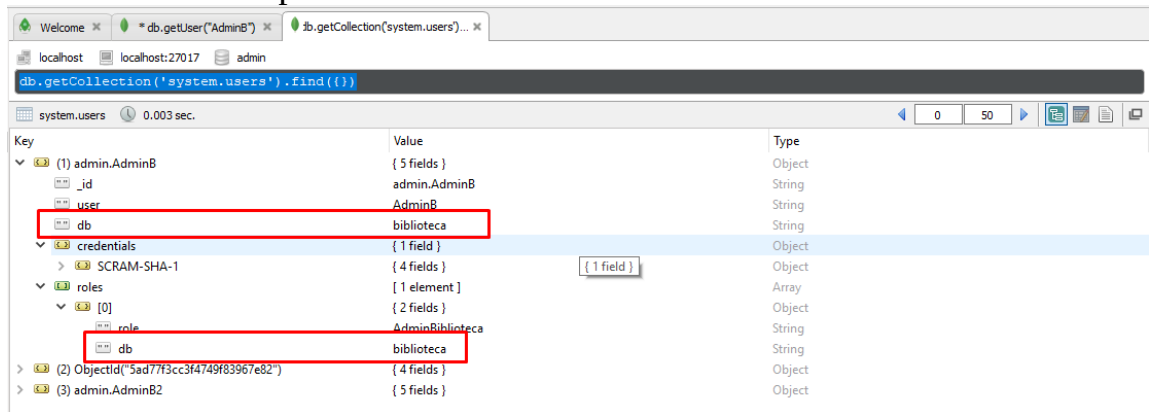
12. En esa consola crearemos el Rol para usuario “AdminBiblioteca”, y luego crearemos el usuario AdminB

```
Db.runCommand({createRole:"AdminBiblioteca",
privileges:[
{resource:{db:"Biblioteca",collection:""},actions:["find","update","insert","remove"]}],
roles:[
{role:"readWrite",db:"Biblioteca"}
]});
db.createUser({
user:"AdminB",
pwd:"123456",
roles:["AdminBiblioteca"]
})
```

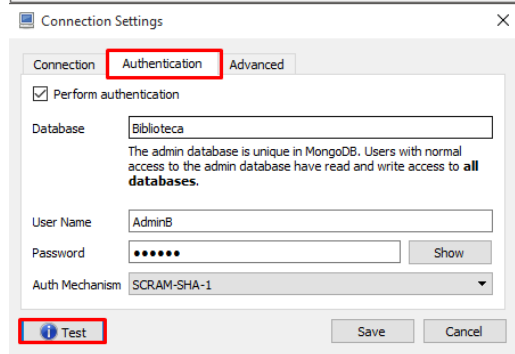
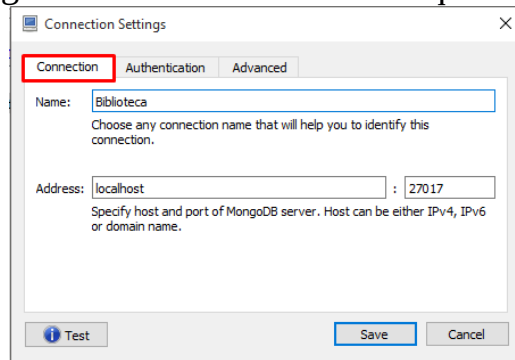
13. Para consultar si se nos creó el usuario, lo hacemos con la siguiente instrucción:

```
db.getCollection('system.users').find({})
```

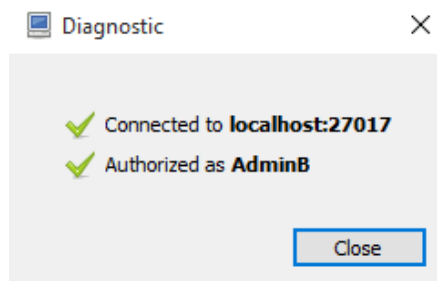
14. Debemos verificar que todos los datos estén correctos, si los datos están correctos proceder a hacer el paso 15, si no entonces debemos cambiar los datos por nuestra base biblioteca.



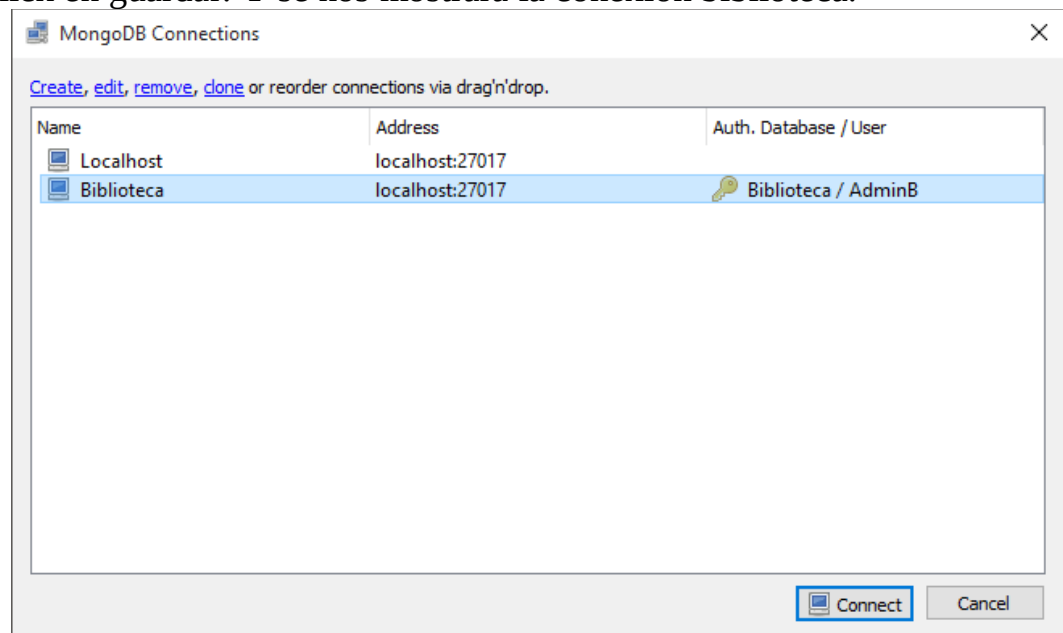
15. Deberemos cerrar conexión y crearemos una nueva conexión, para logearnos con nuestro usuario que hemos creado.



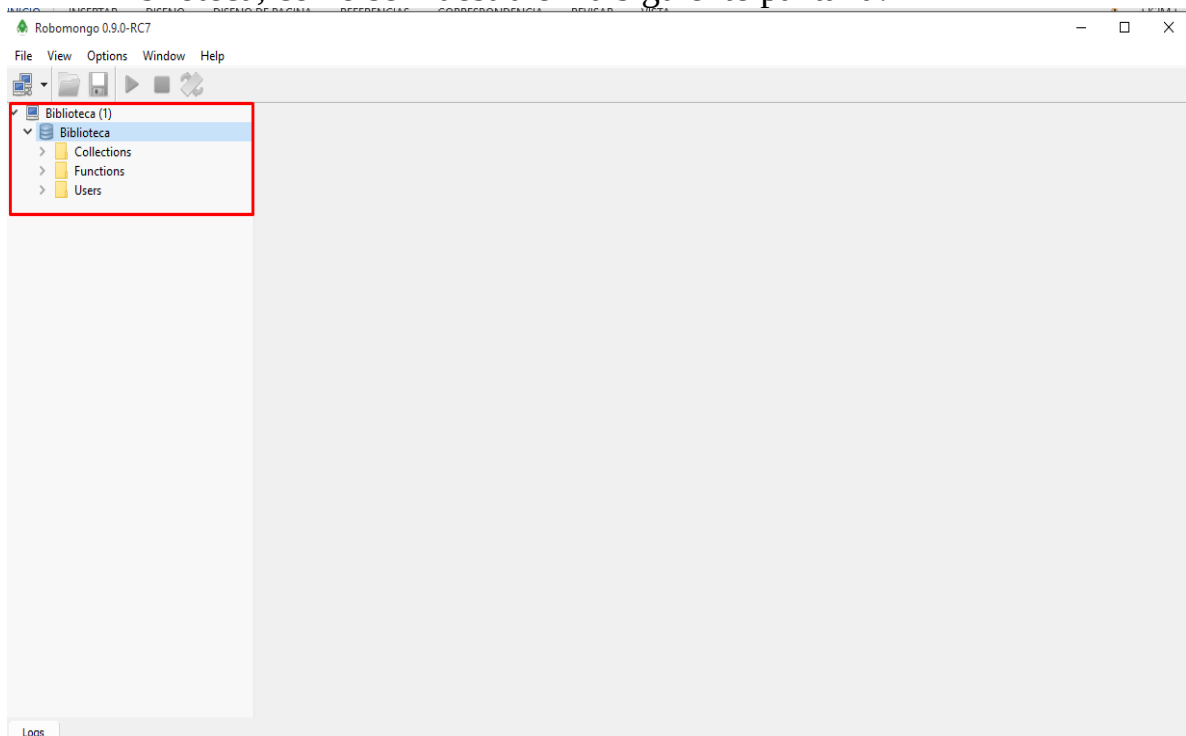
Nota: Debemos dar clic en Test para verificar que el usuario tiene permisos para ingresar a nuestra base de datos.



16. Click en guardar. Y se nos mostrara la conexión biblioteca.



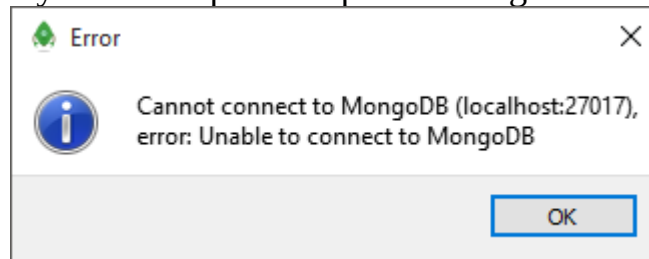
17. Con este usuario solo podremos ver, modificar y leer la base de datos Biblioteca, como se muestra en la siguiente pantalla:



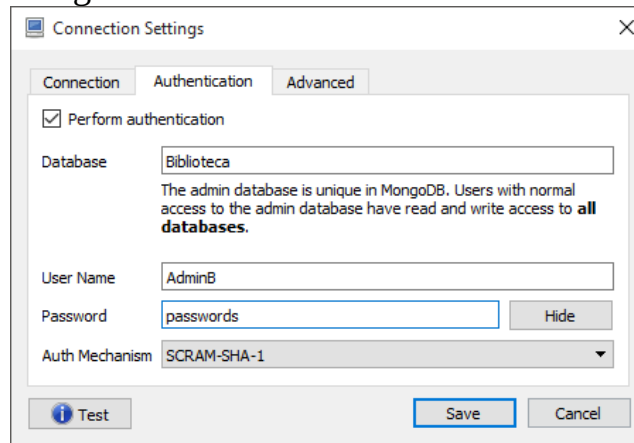
18. Si deseamos cambiar el password del usuario lo podemos hacer con el siguiente query:

```
db.changeUserPassword("AdminB", "password")
```

19. Cerremos la sesión y volvamos a intentar ingresar con la conexión que acabamos crear y veremos que nos aparece el siguiente error:



20. Necesitamos editar la conexión con el nuevo password que ingresamos y al usuario y luego volvemos a intentar conectarnos y hoy si podremos ingresar.



21. Si queremos Eliminar un usuario, lo haremos de la siguiente forma.

```
db.removeUser("AdminB")
```

22. Consultamos si nuevamente si con la función getUser. Si el usuario se eliminó.

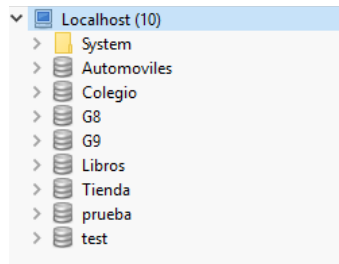
```
db.getUser("AdminB")
```

PARTE 3: Respaldo y Restauración en MongoDB

1. Abrir otra ventana de consola de Windows y escribir en ella el documento **mongodump**, haremos una Respaldo de todas las collecciones

NOTA: Si queremos hacer un respaldo solo de una BD se escribe el siguiente comando: **mongodump --db Nombre_Base_Datos**

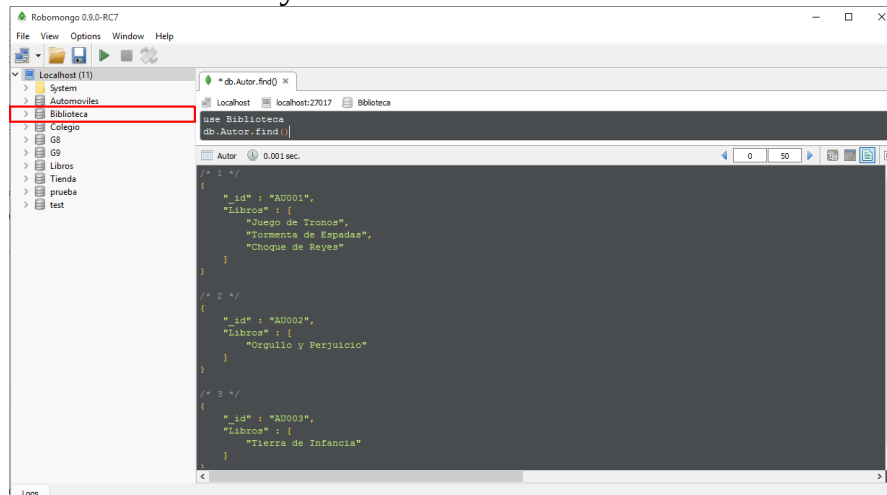
2. Eliminar la base de datos Biblioteca:



3. Restauraremos nuestra base de datos Biblioteca con el siguiente comando:

mongorestore - -db Biblioteca dump/Biblioteca

4. Ahora volvamos a nuestro cliente RoboMongo y podemos comprobar que nuestra base de datos Biblioteca se restauró con todas sus colecciones y Clientes.



7. EJERCICIO COMPLEMENTARIO

NOMBRE	APELLIDO	EMAIL	EDAD	FECHA NACIMIENTO	ESTADO CIVIL	NACIONALIDAD	DEPARTAMENTO	JEFE
Karina	Rivera	Karina.rivera@udb.sv	35	1978-12-10	Soltera	Hondureña	Área Administrativa	Juan Rosales
Sonia	Gutierrez	Sonia.Gutierrez@udb.sv	42	1960-10-10	Casada	Salvadoreña	Área Administrativa	Juan Rosales
Juan	Romero	Juan.romero@udb.sv	25	1992-11-24	Divorciado	Guatemalteco	Área Administrativa	Juan Rosales

Juan	Perez		55	1954-02-28	Viudo	Nicaragüense	Área de Servicios	Pedro Luna
Andrea	Núñez	Andrea.nunez@udb.sv	55	1954-03-30	Casada	Costarricense	Área de mantenimiento	Margarita Flores
Raúl	Perez	Raul.perez@udb.sv	25	1992-02-10	Soltero	Panameño	Área de mantenimiento	Margarita Flores

Consideración para validaciones:

- Nombre, apellido, nombre de jefe solo debe aceptar letras (\$type).
- Todo correo ingresado debe ser de dominio udb.sv (\$regex).
- Edad debe ser un numero entero (\$type).
- Soltero solo debe aceptar los siguientes estados civiles: Soltero, casado, viudo, divorciado (\$in).
- Nacionalidad solo debe aceptar nacionalidad centro americana (\$in).
- Departamento solo pueden ser: Área de mantenimiento, servicios, administrativa, informática, recursos humanos.

Ejercicio 1:

1. Mostrar cantidad de empleados por departamento.
2. Mostrar el empleado con mayor edad.
3. Mostrar empleados con edades de entre 20 y 45 años.
4. Mostrar empleados cuyo nombre es Juan.
5. Sacar promedio de edad por departamento.
6. Mostrar empleados cuyos apellidos sea diferente de Pérez.
7. Mostrar empleados ordenados de forma ascendente por estado civil.

8. BIBLIOGRAFIA

Usaola, M. P. (s.f.). MongoDB: gestión, administración y desarrollo de aplicaciones. Universidad de Catilla.

HOJA DE EVALUACIÓN

Alumno:	Carnet:
Docente:	Fecha:
Título de la guía: Seguridad, Validaciones y Backups	No.: Guía 11

Actividad a evaluar	Criterio a evaluar	Cumplió		Puntaje
		SI	NO	
Procedimiento de guías	Realizo los ejercicios de ejemplos durante la práctica. (40%)			
	Presento todos los problemas resueltos. (20%)			
	Funcionan todos correctamente y sin errores. (30%)			
	Envío los ejercicios en una carpeta comprimida de acuerdo en el tiempo estipulado por el docente. (10%)			
	PROMEDIO:			
Ejercicio Complementario	Realizó el análisis de resultados durante la guía de práctica (40%)			
	Presentó todos los problemas resueltos (20%)			
	Funcionan todos correctamente y sin errores (30%)			
	Envío los ejercicios en una carpeta comprimida de acuerdo en el tiempo estipulado por el docente (10%)			
	PROMEDIO:			