

Clase MongoDB + Node JS

Introducción a MongoDB - Ejecución de comandos desde la consola

1. Iniciar el símbolo del sistema (WIN + r) y digitar mongo

```
C:\Users\Daniel Loaiza>mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("50d063a8-2d6d-4186-b15a-d7a40ebfce93") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2023-02-20T08:03:55.832-05:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

2. Mostrar las bases de datos activas

```
> show dbs
admin  0.000GB
config 0.000GB
local  0.000GB
test   0.000GB
```

3. Retornar el nombre de la base de datos actual

```
> db.getName()
test
```

4. Crear una colección

```
> use cursoMongoDB
switched to db cursoMongoDB
```

5. Mostrar una colección

```
> show collections
```

Nota. No se muestra ninguna colección porque todavía no se ha agregado ningún documento

6. Creación de documentos

```
> var user1 = {  
... name:'Fulano',  
... age:28,  
... email:'fulano@sena.edu.co'  
... }  
> var user2 = {  
... name:'Mengano',  
... age:29,  
... email:'mengano@gmail.com'  
... }  
> var user3 = {  
... name:'Perano',  
... age:30,  
... email:'perano@sena.edu.co'  
... }
```

7. Revisamos que el usuario esté creado

```
> user1  
{ "name" : "Fulano", "age" : 28, "email" : "fulano@sena.edu.co" }
```

8. Inserción de un usuario en la colección

```
> db.users.insertOne(user1)  
{  
  "acknowledged" : true,  
  "insertedId" : ObjectId("63f396e0e9d9e8ec4f1e6165")  
}
```

9. Inserción de muchos usuarios

```
> db.users.insertMany([user2,user3])  
{  
  "acknowledged" : true,  
  "insertedIds" : [  
    ObjectId("63f396f6e9d9e8ec4f1e6166"),  
    ObjectId("63f396f6e9d9e8ec4f1e6167")  
  ]  
}
```

10. Mostrar todos los documentos de la colección

```
> db.users.find()  
{ "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"), "name" : "Fulano", "age" : 28, "email" : "fulano@sena.edu.co" }  
{ "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"), "name" : "Mengano", "age" : 29, "email" : "mengano@gmail.com" }  
{ "_id" : ObjectId("63f396f6e9d9e8ec4f1e6167"), "name" : "Perano", "age" : 30, "email" : "perano@sena.edu.co" }
```

11. Mostrar todos los documentos con pretty()

```
> db.users.find().pretty()  
{  
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),  
  "name" : "Fulano",  
  "age" : 28,  
  "email" : "fulano@sena.edu.co"  
}
```

```

    "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
    "name" : "Mengano",
    "age" : 29,
    "email" : "mengano@gmail.com"
  }
  {
    "_id" : ObjectId("63f396f6e9d9e8ec4f1e6167"),
    "name" : 30,
    "email" : "perano@sena.edu.co"
  }

```

12. Consulta

```

> db.users.find({age:{$ne:29}})
{ "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"), "name" : "Fulano", "age" : 28, "email" : "fulano@sena.edu.co" }
{ "_id" : ObjectId("63f396f6e9d9e8ec4f1e6167"), "name" : 30, "email" : "perano@sena.edu.co" }

> db.users.find({age:{$ne:29}}).pretty()
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6167"),
  "name" : 30,
  "email" : "perano@sena.edu.co"
}

```

13. Consulta com operadores relacionales

```

> //$eq, $ne, $gt, $gte, $lt, $lte
> db.users.findOne({age:{$ne:29}}).pretty()
uncaught exception: TypeError: db.users.findOne(...).pretty is not a function :
@(shell):1:1

> db.users.findOne({age:{$ne:29}})
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}

```

14. Operadores logicos

```

> //$and, $or
> db.users.find({$and:[{age:{$gt:28}},{age:{$lt:30}}]}).pretty()
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
  "name" : "Mengano",
  "age" : 29,
  "email" : "mengano@gmail.com"
}

```

15. Expresiones regulares

```
> // comienza con: /^ /
> // contiene: / /
> // finaliza con: / $/

> db.users.find({name:/^Ful/}).pretty()
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}

> db.users.find({name:/an/}).pretty()
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
  "name" : "Mengano",
  "age" : 29,
  "email" : "mengano@gmail.com"
}

> db.users.find({name:/no$/}).pretty()
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
  "name" : "Mengano",
  "age" : 29,
  "email" : "mengano@gmail.com"
}
```

16. Operador \$in

```
> db.users.find({name:{$in:['Fulano','Mengano']}}).pretty()
{
  "_id" : ObjectId("63f396e0e9d9e8ec4f1e6165"),
  "name" : "Fulano",
  "age" : 28,
  "email" : "fulano@sena.edu.co"
}
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
  "name" : "Mengano",
  "age" : 29,
  "email" : "mengano@gmail.com"
}
```

17. Actualización de elementos si existen \$exists y \$set

```
> db.users.updateMany( {assisted:{ $exists:false}}, { $set:{assisted:false}} )
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }

> db.users.updateOne({name:'Fulano'}, { $set:{assisted:true}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 0 }
```

18. Eliminar elementos si estan creados \$unset

```
> db.users.updateMany({createdAt:{ $exists:true}}, { $unset:{createdAt:true}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
```

19. Incrementar valores \$inc

```
> db.users.updateOne({name:'Mengano'}, { $inc:{age:10}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.users.find({name:'Mengano'}).pretty()
{
  "_id" : ObjectId("63f396f6e9d9e8ec4f1e6166"),
  "name" : "Mengano",
  "age" : 39,
  "email" : "mengano@gmail.com",
  "asistio" : true
}
```

20. Eliminar documentos remove()

```
> db.users.remove({name:'Mengano'})
WriteResult({ "nRemoved" : 1 })
> //db.users.remove({}) <-Ojo! Elimina toda la base de datos
```

21. Renombrar atributos \$rename

```
> db.users.updateMany({}, { $rename:{last_name:'lastName'}})
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 0 }
```