

MongoDB Fundamentos

Módulo 3

Uso de JavaScript en Mongo Shell

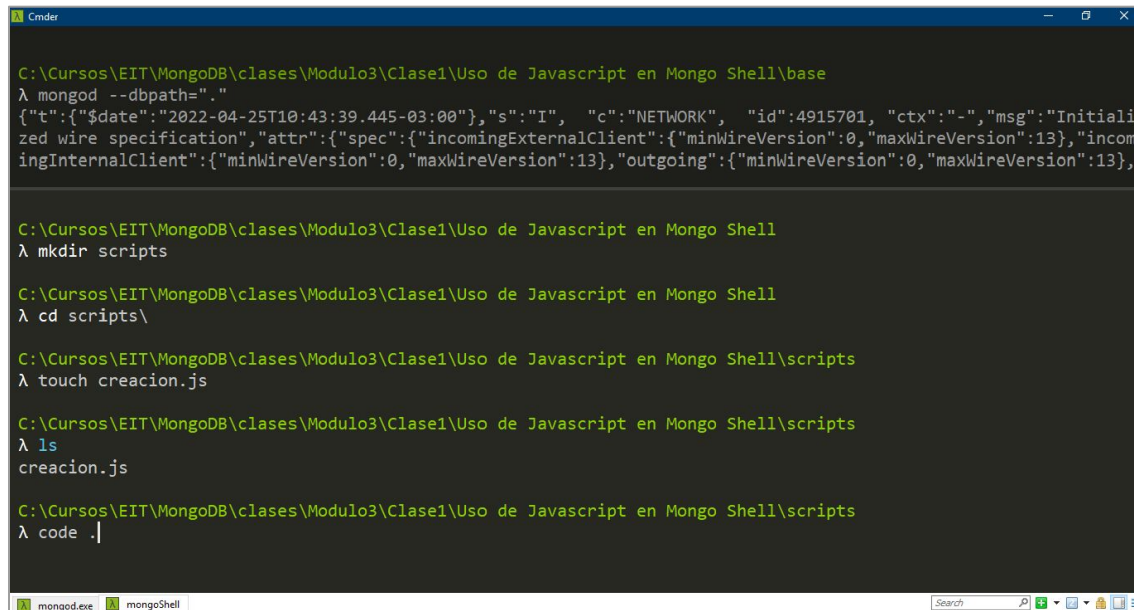
Uso de JavaScript en Mongo Shell

Cargar y ejecutar un archivo *.js

Desde la aplicación de consola de MongoDB (shell), se pueden **ejecutar comandos JavaScript en línea que pueden estar en un archivo *.js** y se pueden cargar, simplemente, mediante un comando. Para implementar un script de mediana o gran complejidad, lo más adecuado es **utilizar un editor de texto** y grabarlo en un archivo *.js

Por ejemplo, se crea la carpeta *scripts* dentro de la carpeta actual de trabajo y dentro de ella un archivo llamado '**creacion.js**'. Luego, con el comando “**code .**”, se lanzará la ejecución del editor de código Visual Studio Code (VSC).





```
C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\base
λ mongod --dbpath="."
{"t":{"$date":"2022-04-25T10:43:39.445-03:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-", "msg":"Initiali
zed wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incom
ingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},

C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell
λ mkdir scripts

C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell
λ cd scripts\

C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts
λ touch creacion.js

C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts
λ ls
creacion.js

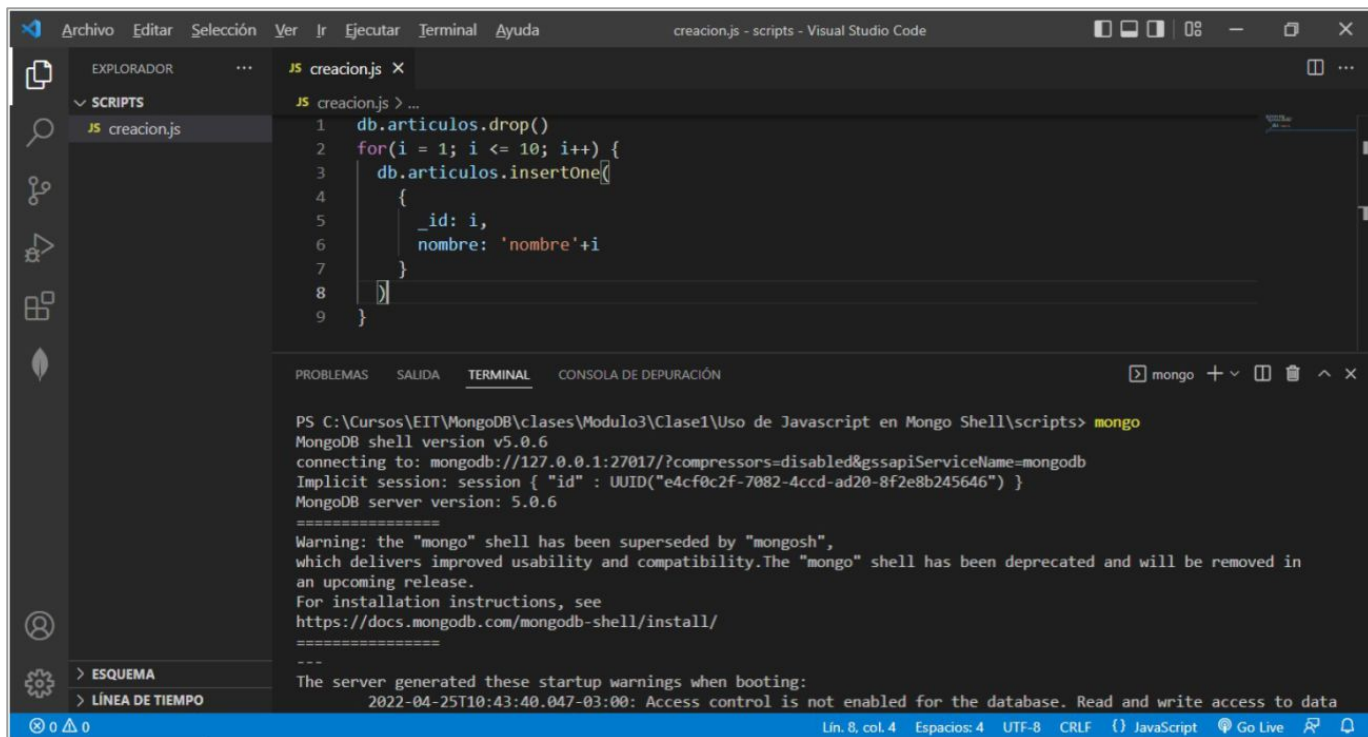
C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts
λ code .|
```

Luego, se codifica el siguiente bloque dentro del archivo 'creacion.js' y se guarda en la carpeta actual.

```
db.articulos.drop()
for(i = 1; i <= 10; i++) {
  db.articulos.insertOne(
    {
      _id: i,
      nombre: 'nombre'+i
    }
  )
}
```

Ahora se puede abrir la consola de VSC y ejecutar Mongo Shell dentro de ella.





The image shows a Visual Studio Code editor window with a file named `creacion.js` open. The file contains a JavaScript script that drops the `articulos` collection and inserts 10 documents into it. The terminal window shows the output of running the script using the `mongo` shell.

```
JS creacion.js X
JS creacion.js > ...
1 db.articulos.drop()
2 for(i = 1; i <= 10; i++) {
3   db.articulos.insertOne(
4     {
5       _id: i,
6       nombre: 'nombre'+i
7     }
8   )
9 }
```

PROBLEMAS SALIDA **TERMINAL** CONSOLA DE DEPURACIÓN

PS C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts> mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e4cf0c2f-7082-4ccd-ad20-8f2e8b245646") }
MongoDB server version: 5.0.6

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.
For installation instructions, see
<https://docs.mongodb.com/mongodb-shell/install/>

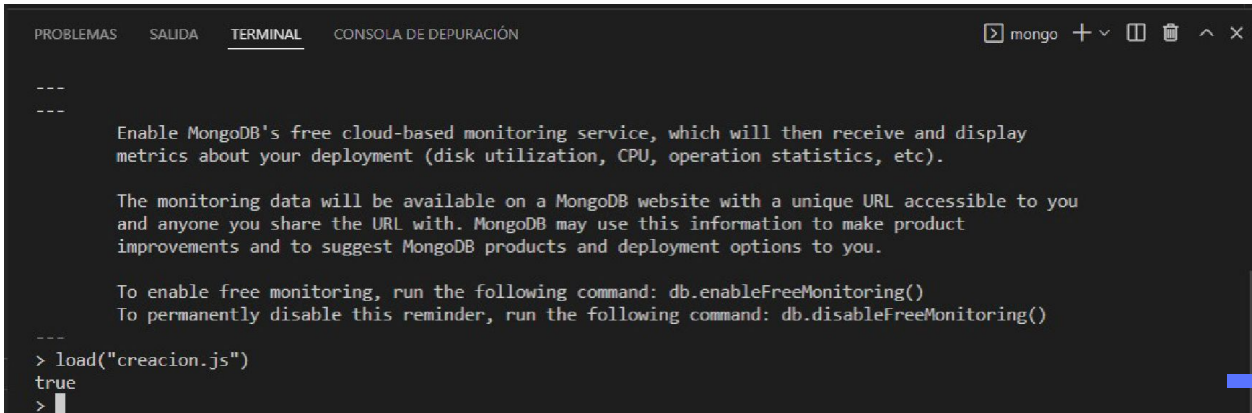
The server generated these startup warnings when booting:
2022-04-25T10:43:40.047-03:00: Access control is not enabled for the database. Read and write access to data

0 0 0 Lin. 8, col. 4 Espacios: 4 UTF-8 CRLF () JavaScript Go Live

Se ejecuta el comando **load** desde el shell de MongoDB:

```
load("creacion.js")
```

Luego de ejecutar la función **load**, aparece la confirmación de si se ejecutó el bloque de comandos en forma exitosa:



```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  mongo + v [icon] [icon] ^ x

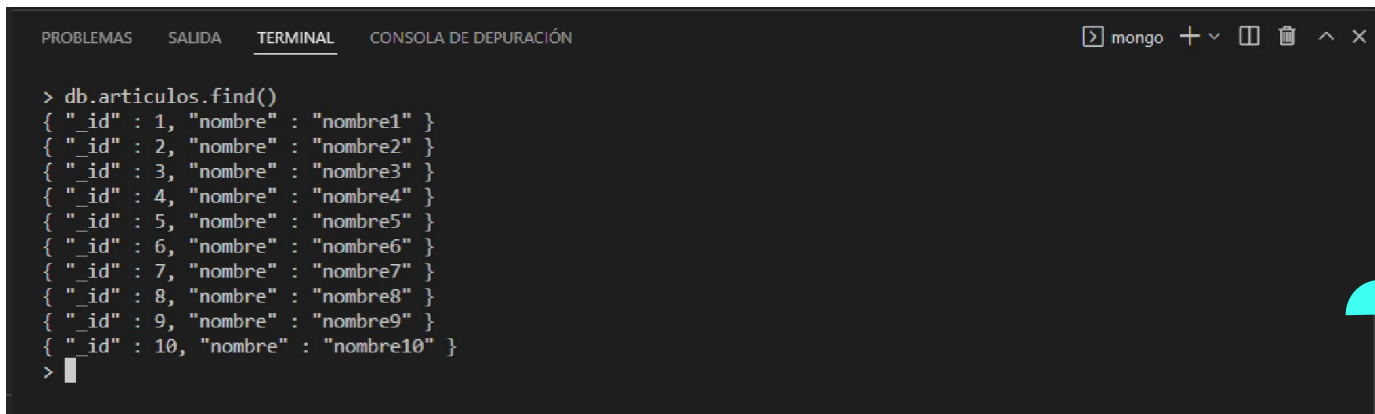
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  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()

---
> load("creacion.js")
true
> 
```

Se puede comprobar que la colección artículos se creó correctamente y se cargaron los 10 documentos:



```
> db.articulos.find()
{ "_id" : 1, "nombre" : "nombre1" }
{ "_id" : 2, "nombre" : "nombre2" }
{ "_id" : 3, "nombre" : "nombre3" }
{ "_id" : 4, "nombre" : "nombre4" }
{ "_id" : 5, "nombre" : "nombre5" }
{ "_id" : 6, "nombre" : "nombre6" }
{ "_id" : 7, "nombre" : "nombre7" }
{ "_id" : 8, "nombre" : "nombre8" }
{ "_id" : 9, "nombre" : "nombre9" }
{ "_id" : 10, "nombre" : "nombre10" }
>
```


Se ha pasado el **path** relativo donde se encuentra el archivo '**creacion.js**'. Si el archivo ***.js** está en otra carpeta desde donde se inició MongoDB Shell, se puede especificar el **path** absoluto al llamar a **load**.

Por ejemplo en este caso:

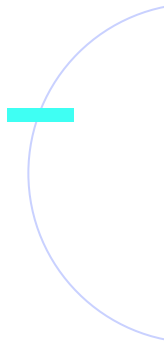
```
load("C:\\Cursos\\EIT\\MongoDB\\clases\\Modulo3\\Clase1\\Uso  
de Javascript en Mongo Shell\\scripts\\creacion.js")
```

Si se usa la barra invertida en el **path** '****' se debe anteponer una barra invertida para “escapar” este carácter de control y que lo tome como una barra literal.

Si se utiliza la barra normal “/” no se necesita escaparla:

```
load("C:/Cursos/EIT/MongoDB/clases/Modulo3/Clase1/Uso de  
Javascript en Mongo Shell/scripts/creacion.js")
```

Ejecutando ambos comandos en MongoDB shell se obtiene el mismo resultado. Veamos la siguiente slide.



Comandos de MongoDB en un archivo *.js

En un script no se pueden utilizar directamente los comandos **show dbs**, **use**, **show collections**, etc. pero se pueden sustituir llamando a métodos:

use base1	db = db.getSiblingDB('base1')
show dbs, show databases	db.adminCommand('listDatabases')
show collections	db.getCollectionNames()
show users	db.getUsers()
show roles	db.getRoles({showBuiltinRoles: true})
show log	db.adminCommand({ 'getLog' : '' })
show logs	db.adminCommand({ 'getLog' : '*' })
it	<pre>cursor = db.collection.find(); while (cursor.hasNext()) { printjson(cursor.next()); }</pre>

En un script, siempre que se necesiten hacer salidas por pantalla debemos utilizar la función **print** y **printjson**, por ejemplo:

```
printjson(db.adminCommand('listDatabases'))
```

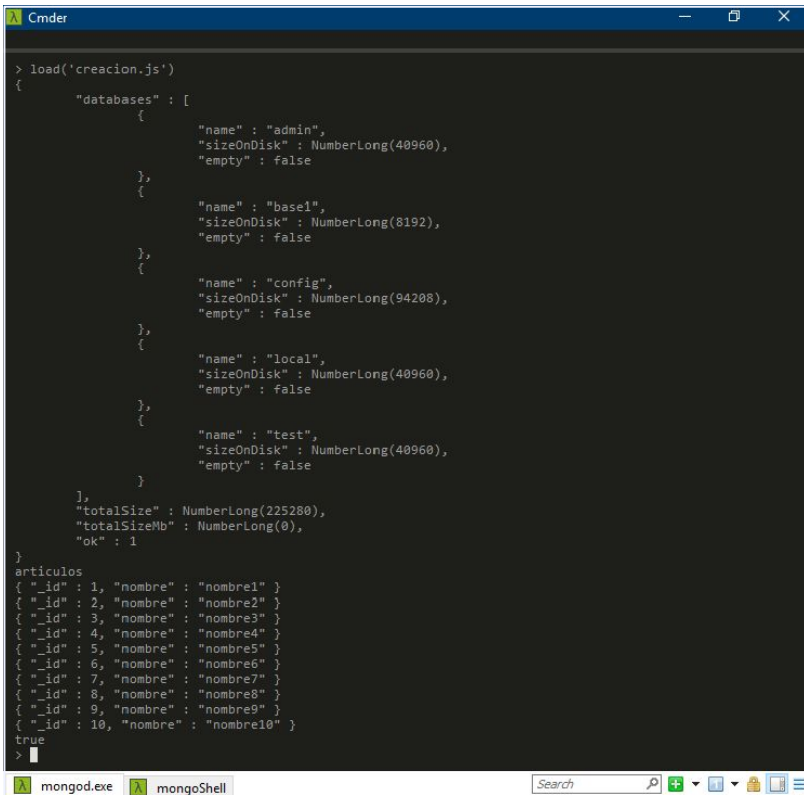
Se modificará el archivo '**creacion.js**' con el siguiente código y luego se volverá a cargar mediante la función 'load':

```
printjson(db.adminCommand('listDatabases'))

db = db.getSiblingDB('base1')
print(db.getCollectionNames())

db.articulos.drop()
for (i = 1; i <= 10; i++) {
    db.articulos.insertOne(
        {
            _id: i,
            nombre: 'nombre' + i
        }
    )
}
cursor = db.articulos.find();
while (cursor.hasNext()) {
    printjson(cursor.next());
}
```

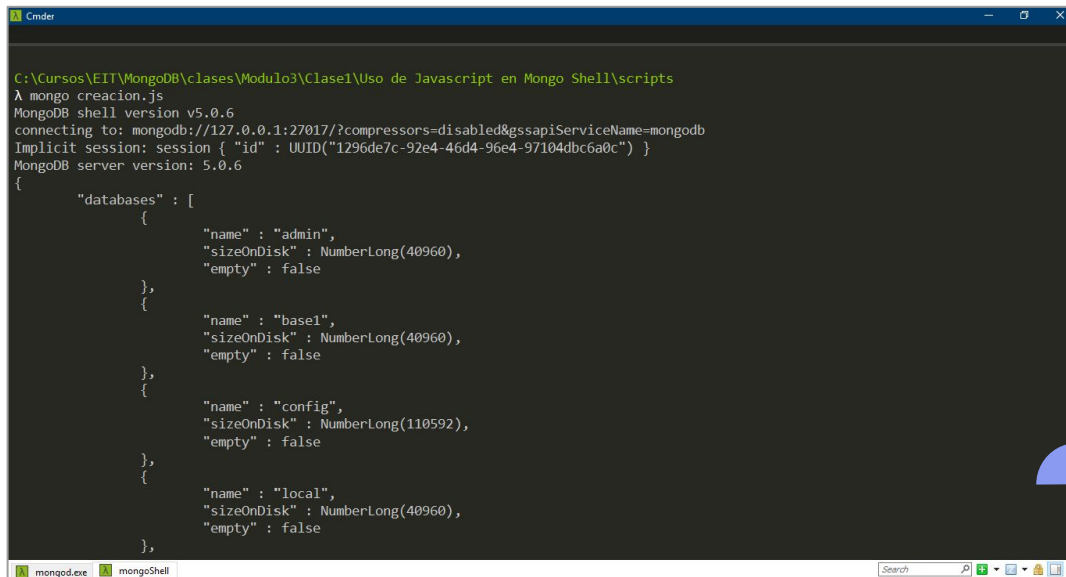
Se puede comprobar que se muestran los datos, en este caso, utilizando Mongo Shell desde una consola externa:



```
> load('creacion.js')
{
  "databases" : [
    {
      "name" : "admin",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    },
    {
      "name" : "base1",
      "sizeOnDisk" : NumberLong(8192),
      "empty" : false
    },
    {
      "name" : "config",
      "sizeOnDisk" : NumberLong(94208),
      "empty" : false
    },
    {
      "name" : "local",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    },
    {
      "name" : "test",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    }
  ],
  "totalSize" : NumberLong(225280),
  "totalSizeMb" : NumberLong(0),
  "ok" : 1
}
articulos
{ "_id" : 1, "nombre" : "nombre1" }
{ "_id" : 2, "nombre" : "nombre2" }
{ "_id" : 3, "nombre" : "nombre3" }
{ "_id" : 4, "nombre" : "nombre4" }
{ "_id" : 5, "nombre" : "nombre5" }
{ "_id" : 6, "nombre" : "nombre6" }
{ "_id" : 7, "nombre" : "nombre7" }
{ "_id" : 8, "nombre" : "nombre8" }
{ "_id" : 9, "nombre" : "nombre9" }
{ "_id" : 10, "nombre" : "nombre10" }
true
>
```

Como opción, se puede ejecutar el script pasándoselo como parámetro a Mongo Shell:

C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts & mongo creacion.js



```
C:\Cursos\EIT\MongoDB\clases\Modulo3\Clase1\Uso de Javascript en Mongo Shell\scripts
λ mongo creacion.js
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("1296de7c-92e4-46d4-96e4-97104dbc6a0c") }
MongoDB server version: 5.0.6
{
  "databases" : [
    {
      "name" : "admin",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    },
    {
      "name" : "base1",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    },
    {
      "name" : "config",
      "sizeOnDisk" : NumberLong(110592),
      "empty" : false
    },
    {
      "name" : "local",
      "sizeOnDisk" : NumberLong(40960),
      "empty" : false
    }
  ]
}
```

**¡Sigamos
trabajando!**

