

Mongo DB Instalación y Conceptos básicos

Web: <https://www.mongodb.com/>

MongoDB es un sistema de base de datos NoSQL, multiplataforma orientado a documentos con esquema libre, en otras palabras, cada registro puede tener datos diferentes, con atributos que no tienen por qué repetirse de un registro a otro, estos registros se denominan documentos, los cuales se pueden agrupar en colecciones que serían las equivalentes a las tablas de los sistemas relacionales.

Los documentos tienen un formato BSON, que sería un Binary JSON, por lo que trabajaremos sobre nuestros datos en formato JSON.



Características:

Rapidez a la hora de ejecutar sus tareas, gracias a su sencillo sistema de consulta de contenidos.

Fácil de escalar.

Reduce drásticamente los costes de almacenamiento

Incorpora muchos de los tipos de consulta que hemos usado en los sistemas relacionales.

Tipos de datos:

MongoDB trabaja con algunos de los siguientes tipos para tener una referencia más alta visitar la siguiente dirección: <https://docs.mongodb.com/manual/reference/bson-types/>

ObjectId: ObjectId("41c411ba99ed1dc3ez700981"), el cual es referente para los identificadores de los datos que guardamos y solo se usa para la clave principal "_id" pero este es modificable a lo que necesitamos, este identificador se crea automáticamente para todos los documentos que guardemos en nuestra BD y tienen un valor único.

String: Se almacenan entre comillas dobles y en formato UTF-8.

Number: Números de tipo byte, int32, int64 o double.

Boolean: true o false.

Null: null.

Timestamp: Timestamp(1412180887, 1).

Array: Almacenados entre corchetes [] de cualquiera de los otros tipos.

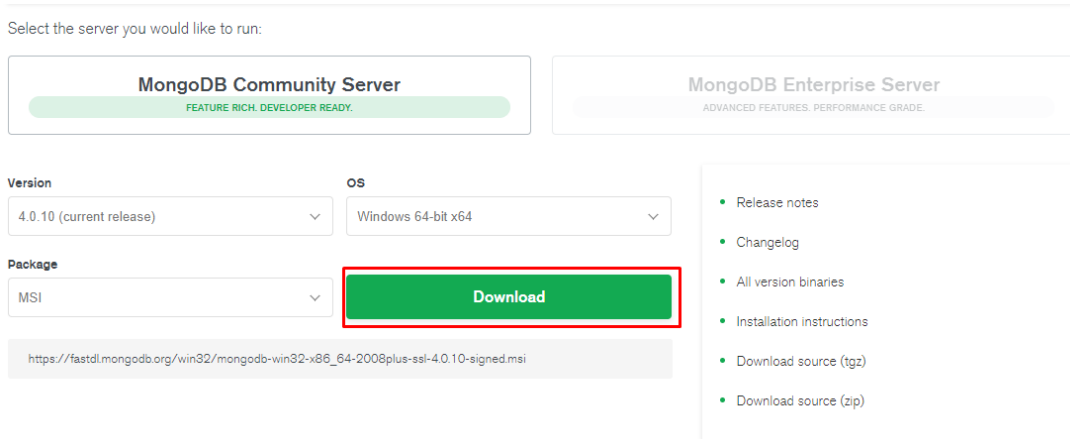
Documentos: Almacenados entre llaves {} o sea en formato JSON de cualquiera de los otros tipos.

Descarga e instalación

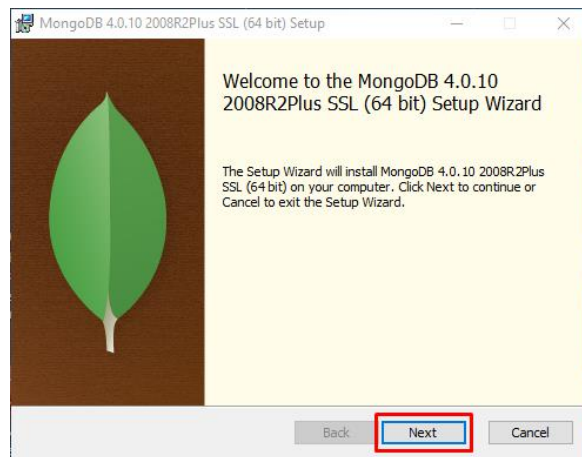
Nos dirigimos a la siguiente dirección:

<https://www.mongodb.com/download-center/community?jmp=docs>

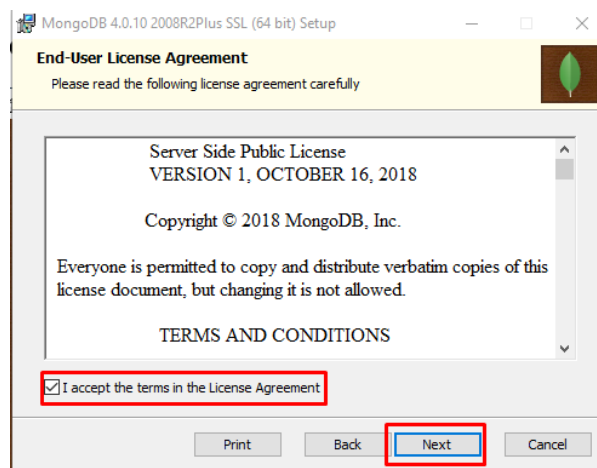
Elegimos nuestro sistema operativo y hacemos click en Download.



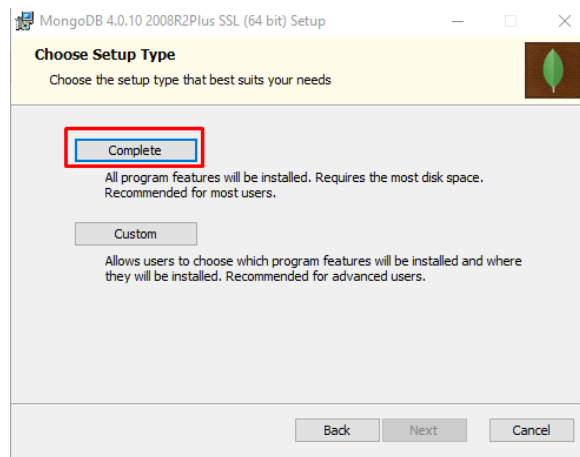
Procedemos a la instalación, hacemos click en Next.



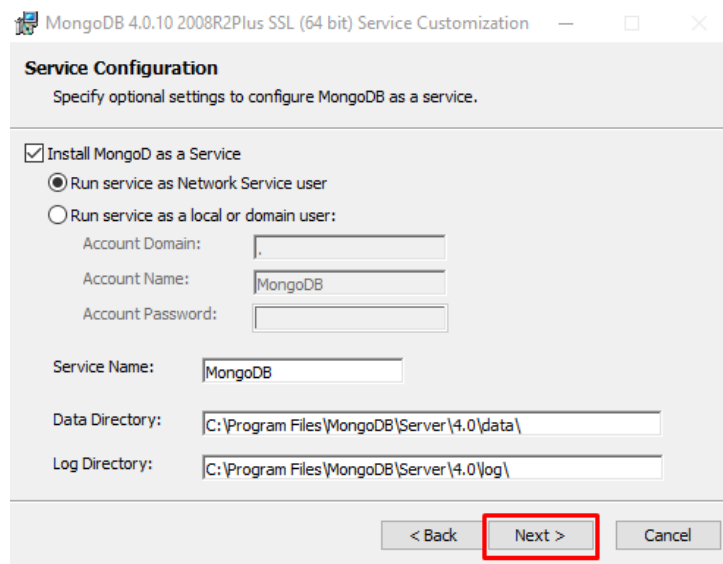
Aceptamos las condiciones y click en Next.



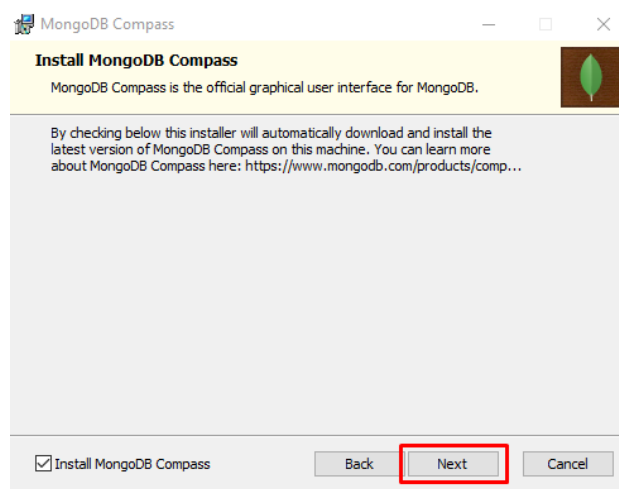
Hacemos click en Complete.



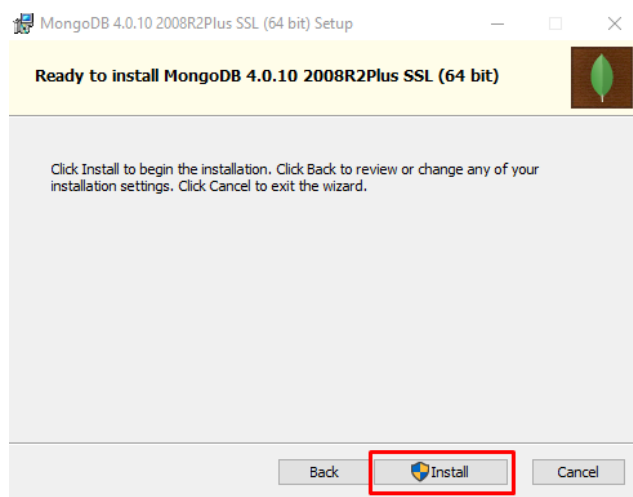
Dejamos las opciones por defecto y hacemos click en Next.



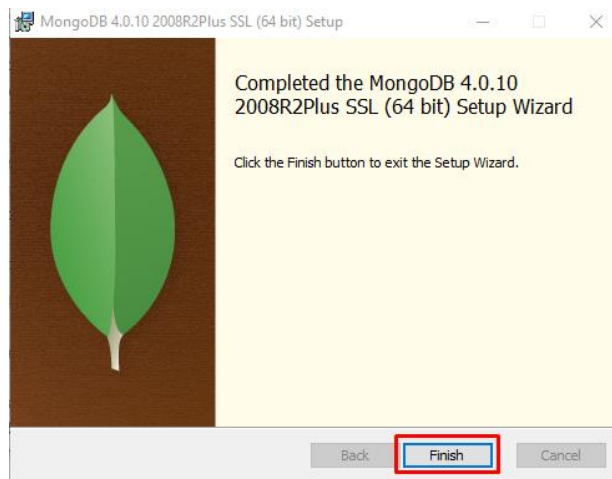
Igualmente hacemos click en Next, desmarcamos la opción de Instalar MongoDB Compass si no deseamos instalarlo.



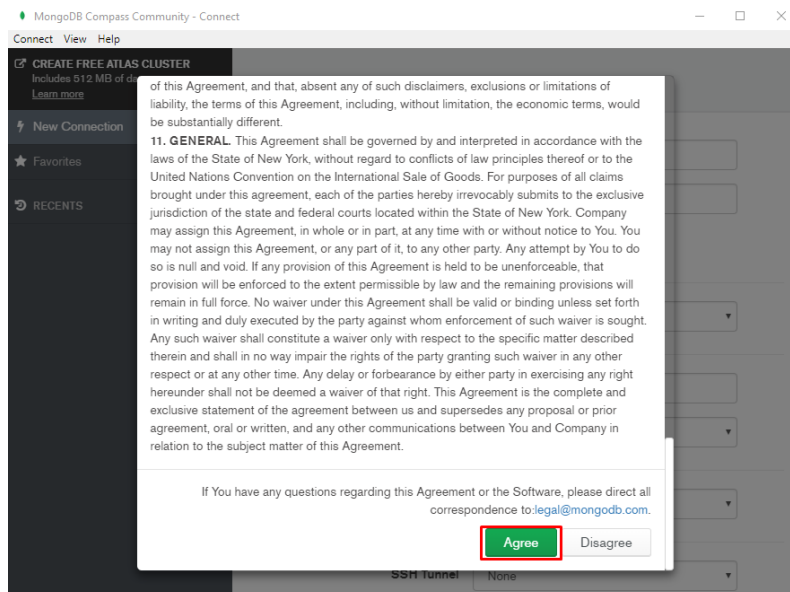
Y hacemos click en Install.



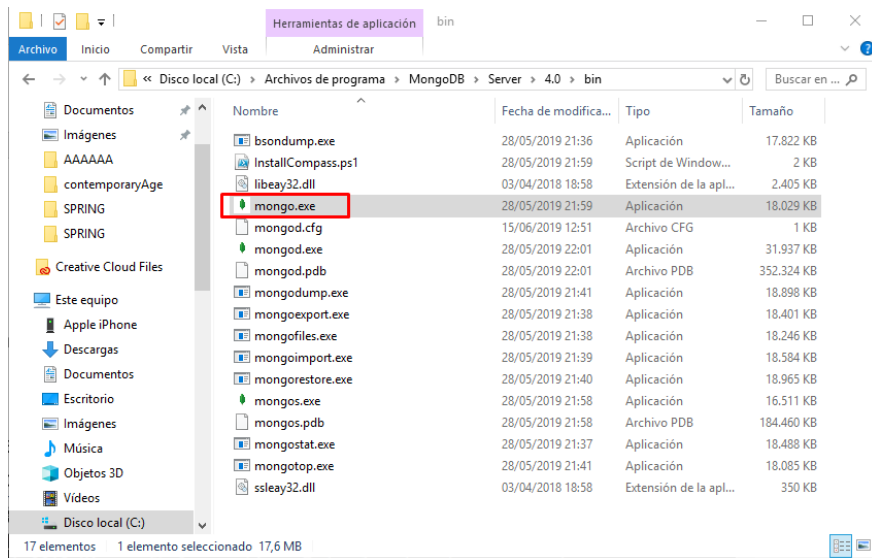
Y click en Finish.



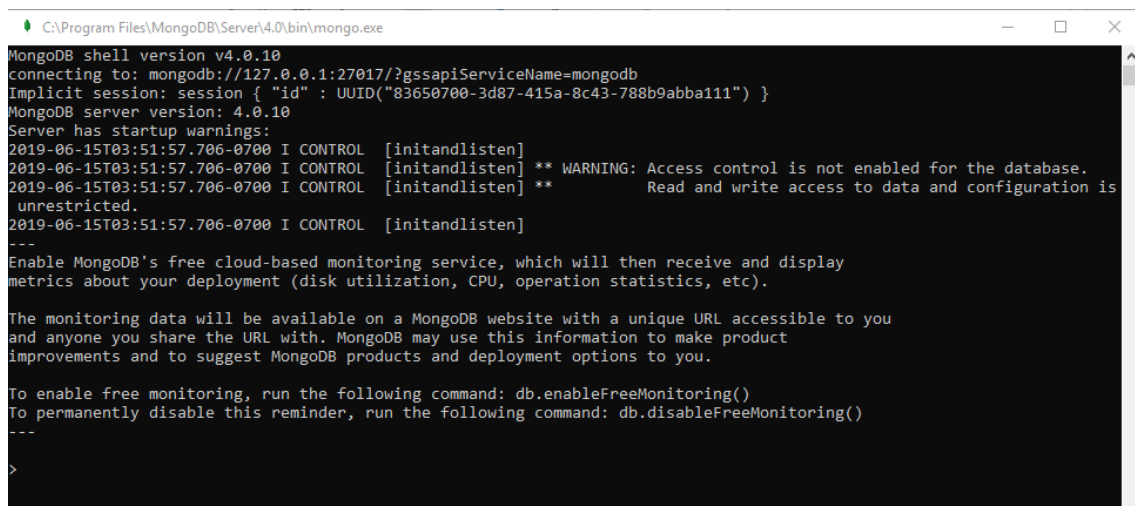
Hacemos click en Agree y cerramos MongoDB Compass si hemos elegido anteriormente instalarlo.



Inicializamos MongoDB haciendo doble click sobre mongo.exe que lo podemos encontrar en la siguiente ruta: C:\Program Files\MongoDB\Server\4.0\bin



Y como vemos arranca correctamente y ya está listo para ejecutar nuestros comandos.



Comandos básicos

Una vez arrancado MongoDB podemos ver las bases de datos que tenemos, por defecto vienen creadas tres.

show dbs

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo.exe
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB
>
```

Para crear una nueva escribimos el comando use nombre_bd que también nos sirve para usar una base de datos existente o sea se cambiar entre bases de datos.

use mongodb_rest

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo.exe
> use mongodb_rest
switched to db mongodb_rest
>
```

Creamos una colección dentro de la base de datos mongodb_rest.

db.createCollection("personas")

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo.exe
> db.createCollection("personas")
{ "ok" : 1 }
>
```

Mostrar las colecciones de la base de datos en la que estamos.

show collections

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo.exe
> show collections
personas
>
```

Insertar un documento.

db.personas.insert({"_id":"1","nombre":"javi"})

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo.exe
> db.personas.insert({"_id":"1","nombre":"javi"})
WriteResult({ "nInserted" : 1 })
>
```

Insertar varios documentos.

```
db.personas.insert([{"_id":"2","nombre":"jose"}, {"_id":"3","nombre":"dani"}])
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.insert([{"_id":"2","nombre":"jose"}, {"_id":"3","nombre":"dani"}])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

Una vez añadidos los documentos podemos que se ha creado nuestra BD usando de nuevo

show dbs

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> show dbs
admin          0.000GB
config         0.000GB
local          0.000GB
mongodb_rest  0.000GB
```

Ver los documentos de una colección.

```
db.personas.find()
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find()
{ "_id" : "1", "nombre" : "javi" }
{ "_id" : "2", "nombre" : "jose" }
{ "_id" : "3", "nombre" : "dani" }
>
```

Ver un documento de una colección con un valor.

```
db.personas.find({"nombre":"javi"})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"nombre":"javi"})
{ "_id" : "1", "nombre" : "javi" }
>
```

Ver un documento de una colección con más de un valor.

```
db.personas.find({"_id":"1","nombre":"javi"})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"nombre":"javi"})
{ "_id" : "1", "nombre" : "javi" }
>
```

Borrar un registro.

```
db.personas.remove({_id:"3"})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.remove({_id:"3"})
WriteResult({ "nRemoved" : 1 })
>
```

Borrar una colección.

```
db.personas.drop()
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.drop()
true
>
```

Buscar y mostrar solamente un campo en este caso "nombre", el 1 muestra el campo y el 0 no muestra el campo.

```
db.personas.find({"nombre":"javi"},{"nombre":1,"_id":0})
```

findOne() Solo muestra el primero aunque obtenga más de un resultado

```
db.personas.findOne({"nombre":"javi"},{"nombre":1,"_id":0})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"nombre":"javi"},{"nombre":1,"_id":0})
{ "nombre" : "javi" }
>
```

Contar los documentos de una colección.

```
db.personas.find({}).count()
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({}).count()
2
>
```

Contar todos los que tienen un mismo nombre.

```
db.personas.find({"nombre":"javi"}).count()
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"nombre":"javi"}).count()
1
>
```

Modificar un valor.

```
db.personas.update({_id:"1"}, {$set: {"nombre":"jruano"}})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.update({_id:"1"}, {$set: {"nombre":"jruano"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```


Modificar varios valores a la vez.

```
db.personas.update({"nombre":"jruano"}, {$set: {"nombre":"javi"},"multi":true})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.update({"nombre":"jruano"}, {$set: {"nombre":"javi"},"multi":true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

Operadores:

\$gt = >

\$gte = >=

\$lt = <

\$lte = <=

\$ne = !

Mostrar los mayores de 1

```
db.personas.find({"_id":{"$gt":"1"},"nombre":1,"_id":0})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"_id":{"$gt":"1"},"nombre":1,"_id":0})
{ "nombre" : "jose" }
>
```

\$in = igual a [valor1,valor2,...] (No funciona con un solo valor).

```
db.personas.find({"_id":{"$in:["1","2"]},"nombre":1,"_id":0})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"_id":{"$in:["1","2"]},"nombre":1,"_id":0})
{ "nombre" : "javi" }
{ "nombre" : "jose" }
>
```

Ver los documentos en los que existe un campo.

```
db.personas.find({"nombre":{"$exists:true"},"nombre":1,"_id":0})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({"nombre":{"$exists:true"},"nombre":1,"_id":0})
{ "nombre" : "javi" }
{ "nombre" : "jose" }
>
```

Consulta con or

```
db.personas.find({$or:{"nombre":"javi"},"nombre":"jose"},"nombre":1,"_id":1})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({$or:{"nombre":"javi"},"nombre":"jose"},"nombre":1,"_id":1})
{ "_id" : "1", "nombre" : "javi" }
{ "_id" : "2", "nombre" : "jose" }
>
```

Consulta con and

```
db.personas.find({'$and':[{"nombre":"jose"}, {"_id":{"$gte":"1"}}]}, {"nombre":1, "_id":1})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({'$and':[{"nombre":"jose"}, {"_id":{"$gte":"1"}}]}, {"nombre":1, "_id":1})
{ "_id" : "2", "nombre" : "jose" }
>
```

Consulta combinada con or y and

```
db.personas.find({'$or':[{"nombre":"javi"}, {"$and':[{"nombre":"jose"}, {"_id":{"$gte":"1"}}]}]}, {"nombre":1, "_id":1})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({'$or':[{"nombre":"javi"}, {"$and':[{"nombre":"jose"}, {"_id":{"$gte":"1"}}]}]}, {"nombre":1, "_id":1})
{ "_id" : "1", "nombre" : "javi" }
{ "_id" : "2", "nombre" : "jose" }
>
```

Buscar los documentos que no cumplan una condición.

\$not -> solo puede usarse con otros operadores como \$gt o \$lt

```
db.personas.find({'_id':{'$not':{'$gte":"4"}}})
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.personas.find({'_id':{'$not':{'$gte":"4"}}})
{ "_id" : "1", "nombre" : "javi" }
{ "_id" : "2", "nombre" : "jose" }
>
```

\$nor -> acepta dos o más valores

Devuelve también los documentos si los campos no existen. Para evitar esto, si es algo que no deseamos, podemos añadir el operador \$exists.

Borrar una BD cambiamos a usar la BD que deseamos con use y borramos la BD.

```
db.dropDatabase()
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> use mongodb_rest
switched to db mongodb_rest
> db.dropDatabase()
{ "dropped" : "mongodb_rest", "ok" : 1 }
>
```