

# Comandos

## Básicos

### Mongo DB

Autor: Ildelfonso Albares García

Fecha: 19/12/2024

## Tabla de contenido

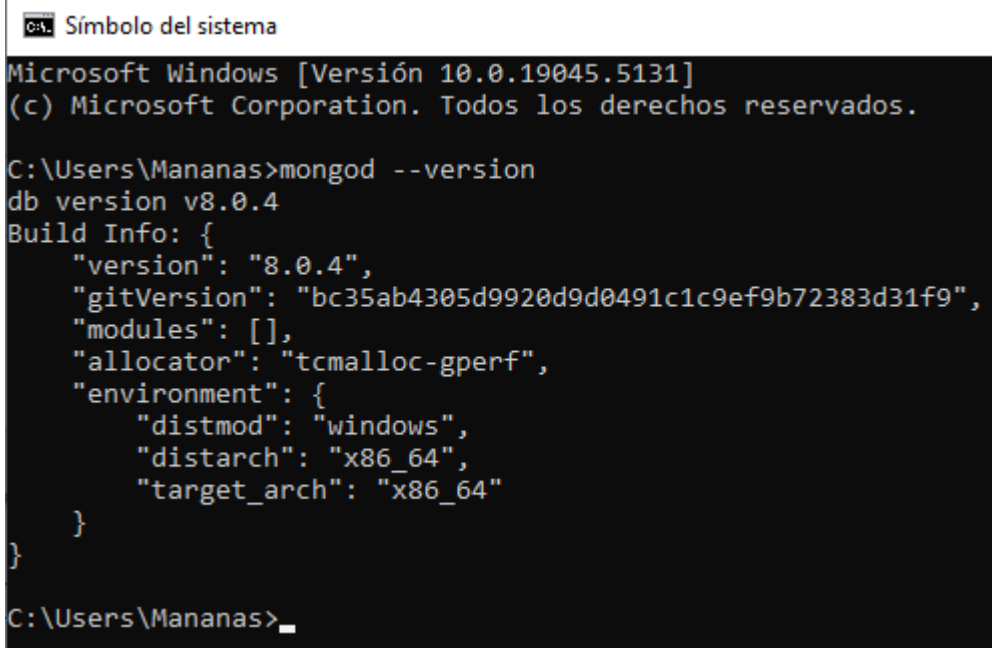
<b>1.</b>	<b><i>Conocer la versión de mongo instalada: .....</i></b>	<b><i>1</i></b>
<b>2.</b>	<b><i>Ver la ayuda: .....</i></b>	<b><i>1</i></b>
<b>3.</b>	<b><i>Conocer el estado:.....</i></b>	<b><i>1</i></b>
<b>4.</b>	<b><i>Muestra las bases de datos actuales del sistema: .....</i></b>	<b><i>2</i></b>
<b>5.</b>	<b><i>Crear una base de datos:.....</i></b>	<b><i>2</i></b>
<b>6.</b>	<b><i>Borrar una base de datos: .....</i></b>	<b><i>2</i></b>
<b>7.</b>	<b><i>Crear una colección:.....</i></b>	<b><i>2</i></b>
<b>8.</b>	<b><i>Eliminar una colección: .....</i></b>	<b><i>2</i></b>
<b>9.</b>	<b><i>Indica el número de documentos de una colección:.....</i></b>	<b><i>2</i></b>
<b>10.</b>	<b><i>Crear un documento en una colección:.....</i></b>	<b><i>2</i></b>
<b>11.</b>	<b><i>Crear varios documentos en una colección:.....</i></b>	<b><i>2</i></b>
<b>12.</b>	<b><i>Muestra todos los documentos de una colección:.....</i></b>	<b><i>3</i></b>
<b>13.</b>	<b><i>Muestra todos los documentos que cumplan una condición: .....</i></b>	<b><i>3</i></b>
<b>14.</b>	<b><i>Modifica los campos del documento: .....</i></b>	<b><i>3</i></b>
<b>15.</b>	<b><i>Modifica los campos de varios documentos: .....</i></b>	<b><i>4</i></b>
<b>16.</b>	<b><i>Elimina un documento:.....</i></b>	<b><i>4</i></b>
<b>17.</b>	<b><i>Elimina varios documentos: .....</i></b>	<b><i>4</i></b>

## Comandos Básicos Mongo DB

### 1. Conocer la versión de mongo instalada:

Comando:

`mongod --version`



```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.5131]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Mananas>mongod --version
db version v8.0.4
Build Info: {
  "version": "8.0.4",
  "gitVersion": "bc35ab4305d9920d9d0491c1c9ef9b72383d31f9",
  "modules": [],
  "allocator": "tcmalloc-gperf",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}

C:\Users\Mananas>
```

### 2. Ver la ayuda:

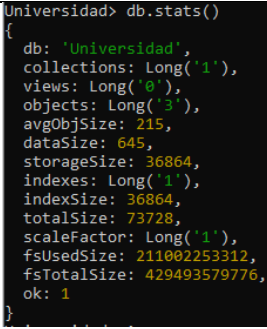
Comando:

`db.help()`

### 3. Resumen estadístico detallado de la base de datos con la que estás trabajando :

Comando:

`db.stats()`



Comando:

```
Universidad> show dbs
Hospital      224.00 KiB
Universidad   72.00 KiB
admin         40.00 KiB
config        108.00 KiB
local         72.00 KiB
```

Comando:

```
2024-12-20T08:48:47.390+01:00: Access control is not enabled for the databat
test> use Universidad
switched to db Universidad
Universidad>
```

Comando:

```
db.nombre_base_datos.drop()
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Universidad> use Prueba
switched to db Prueba
Prueba> db.Prueba.drop()
true
Prueba>
```

## 7. Crear una colección:

Comando:

```
db.createCollection(Nombre,{opciones})
```

Parámetro opciones (no obligatorio):

- capped: Indica si la colección está capada (tiene un tamaño máximo).
- size: Especifica el tamaño máximo de una colección capada en bytes.
- max: Especifica el número máximo de documentos en una colección capada.
- autoIndexId: Determina si se creará automáticamente un índice en el campo `_id`.
- storageEngine: Especifica el motor de almacenamiento a utilizar.

Ejemplo:

```
db.createCollection(Nombre,{capped : true, size : sizeLimit , max : documentLimit })
```

## 8. Eliminar una colección:

Comando:

```
db.collectionName.drop()
```

## 9. Indica el número de documentos de una colección:

Comando:

```
db.usuarios.countDocuments()
```

```
Universidad> db.alumno.find()
[
  {
    _id: ObjectId('6765272311ee83f03ae24c3f'),
    edad: 30,
    nombre: 'Ana',
    aficiones: [ 'futbol', 'pesca', 'escalada' ],
    email: 'ana@mail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Castellano', zonaHoraria: 2 }
  },
  {
    _id: ObjectId('6765276311ee83f03ae24c41'),
    edad: 28,
    nombre: 'Benito',
    aficiones: [ 'futbol', 'baloncesto', 'tenis' ],
    email: 'BenitoFloro@gmail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Ingles', zonaHoraria: 1 }
  },
  {
    _id: ObjectId('6765277211ee83f03ae24c43'),
    edad: 24,
    nombre: 'Andrea',
    aficiones: [ 'boxeo', 'baile', 'lectura' ],
    email: 'Andrea@gmail.com',
    preferencias: { seguimientoEmails: false, idioma: 'Castellano', zonaHoraria: 2 }
  }
]
Universidad> db.alumno.countDocuments()
3
Universidad>
```

## 10. Crear un documento en una colección:

Comando:

```
db.collectionName.insertOne()
```

## 11. Crear varios documentos en una colección:

Comando:

```
db.collectionName.insertMany()
```

Ejemplo:

```
db.Doctor.insertMany([
  {hospital_cod:"13",doctor_no:"435",apellido:"Lopez A.",especialidad:"Cardiologia",salario:350000},
  {hospital_cod:"22",doctor_no:"777",apellido:"Galindo J.",especialidad:"Neurologia",salario:250000},
  {hospital_cod:"18",doctor_no:"982",apellido:"Cajal R.",especialidad:"Cardiologia",salario:290000},
  {hospital_cod:"22",doctor_no:"453",apellido:"Galo C.",especialidad:"Pediatria",salario:250000},
  {hospital_cod:"22",doctor_no:"398",apellido:"Best K.",especialidad:"Psiquiatria",salario:150000},
  {hospital_cod:"22",doctor_no:"385",apellido:"Cabeza D.",especialidad:"Psiquiatria",salario:125000},
  {hospital_cod:"45",doctor_no:"522",apellido:"Adams C.",especialidad:"Neurologia",salario:450000},
  {hospital_cod:"18",doctor_no:"585",apellido:"Miller G.",especialidad:"Ginecologia",salario:450000}
])
```

## 12. Muestra todos los documentos de una colección:

Comando:

```
db.collectionName.find()
```

```
Universidad> db.alumno.find()
[
  {
    _id: ObjectId('6765272311ee83f03ae24c3f'),
    edad: 30,
    nombre: 'Ana',
    aficiones: [ 'futbol', 'pesca', 'escalada' ],
    email: 'ana@mail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Castellano', zonaHoraria: 2 }
  },
  {
    _id: ObjectId('6765276311ee83f03ae24c41'),
    edad: 28,
    nombre: 'Benito',
    aficiones: [ 'futbol', 'baloncesto', 'tenis' ],
    email: 'BenitoFloro@gmail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Ingles', zonaHoraria: 1 }
  },
  {
    _id: ObjectId('6765277211ee83f03ae24c43'),
    edad: 24,
    nombre: 'Andrea',
    aficiones: [ 'boxeo', 'baile', 'lectura' ],
    email: 'Andrea@gmail.com',
    preferencias: { seguimientoEmails: false, idioma: 'Castellano', zonaHoraria: 2 }
  }
]
Universidad>
```

### 13. Muestra todos los documentos que cumplan una condición:

Comando:

```
db.collectionName.find({condición })
```

Opciones de condición:

- \$eq: Igual a
- \$gt: Mayor que
- \$gte: Mayor o igual que
- \$lt: Menor que
- \$lte: Menor o igual que
- \$ne: Diferente de
- \$in: Incluido en un conjunto
- \$nin: No incluido en un conjunto
- \$and: Y lógico
- \$or: O lógico
- \$not: Negación
- \$regex: Expresiones regulares

Opciones de limite y orden (no obligatorio):

- limit(): Limita el número de resultados.
- sort(): Ordena los resultados.

Ejemplo1:

```
db.alumno.find()
```

```
Universidad> db.alumno.find()
[
  {
    _id: ObjectId('6765272311ee83f03ae24c3f'),
    edad: 30,
    nombre: 'Ana',
    aficiones: [ 'fútbol', 'pesca', 'escalada' ],
    email: 'ana@mail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Castellano', zonaHoraria: 2 }
  },
  {
    _id: ObjectId('6765276311ee83f03ae24c41'),
    edad: 28,
    nombre: 'Benito',
    aficiones: [ 'fútbol', 'baloncesto', 'tenis' ],
    email: 'BenitoFloro@gmail.com',
    preferencias: { seguimientoEmails: true, idioma: 'Ingles', zonaHoraria: 1 }
  },
  {
    _id: ObjectId('6765277211ee83f03ae24c43'),
    edad: 24,
    nombre: 'Andrea',
    aficiones: [ 'boxeo', 'baile', 'lectura' ],
    email: 'Andrea@gmail.com',
    preferencias: { seguimientoEmails: false, idioma: 'Castellano', zonaHoraria: 2 }
  }
]
```

Ejemplo2:

```
db.usuarios.find({ edad: { $gt: 30 } }, { nombre: 1, _id: 0 })
```

Ejemplo 3:

```
db.Doctor.find().sort({salario:1})
```



```
Hospital> db.Doctor.find().sort({salario:1})
[
  {
    _id: ObjectId('6764059fd44b851ab14eeb8f'),
    hospital_cod: '22',
    doctor_no: '385',
    apellido: 'Cabeza D.',
    especialidad: 'Psiquiatria',
    salario: 125000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb8e'),
    hospital_cod: '22',
    doctor_no: '398',
    apellido: 'Best K.',
    especialidad: 'Psiquiatria',
    salario: 150000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb8b'),
    hospital_cod: '22',
    doctor_no: '777',
    apellido: 'Galindo J.',
    especialidad: 'Neurologia',
    salario: 250000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb8d'),
    hospital_cod: '22',
    doctor_no: '453',
    apellido: 'Galo C.',
    especialidad: 'Pediatria',
    salario: 250000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb8c'),
    hospital_cod: '18',
    doctor_no: '982',
    apellido: 'Cajal R.',
    especialidad: 'Cardiologia',
    salario: 290000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb8a'),
    hospital_cod: '13',
    doctor_no: '435',
    Hospital>
    especialidad: 'Cardiologia',
    salario: 350000
  },
  {
    _id: ObjectId('6764059fd44b851ab14eeb90'),
    hospital_cod: '45',
    doctor_no: '522',
    apellido: 'Adams C.',
    Hospital>
  }
]
```

## 14. Modifica los campos del documento:

Comando:

```
db.nombre_de_la_coleccion.updateOne( { criterio_de_busqueda }, { operador: { campo_a_actualizar: nuevo_valor } }, { opciones } )
```

Opciones de {operador}:

- \$set: Un operador de actualización que establece nuevos valores para los campos especificados.
- \$inc: Incrementa un número.

- `$unset`: Elimina un campo.
- `$push`: Agrega un elemento a una matriz.
- `$pull`: Elimina un elemento de una matriz.
- `$addToSet`: Agrega un elemento a una matriz si no existe.
- `$rename`: Renombra un campo.
- `$currentDate`: Establece el valor de un campo a la fecha y hora actual.

Opciones de {opciones} (no obligatorio)

- `upsert`: Si el documento no existe, lo crea.
- `multi`: Actualiza todos los documentos que coincidan con el criterio (no recomendado, ya que puede tener consecuencias inesperadas).

Ejemplo:

```
db.usuarios.updateOne({ nombre: "Beniro" }, { $set: { edad: 26 } })
```

```
Universidad> db.alumno.updateOne({"nombre":"Benito"}, {$set: {"edad":26}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

## 15. Modifica los campos de varios documentos:

Comando:

```
db.collectionName.updateMany(query, update, options)
```

Ejemplo:

```
Universidad> db.alumno.updateMany({"preferencias.zonaHoraria":1},{$set: {"preferencias.zonaHoraria":4}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

## 16. Elimina un documento:

Comando:

```
db.collectionName.deleteOne()
```

## 17. Elimina varios documentos:

Comando:

```
db.collectionName.deleteMany()
```

## 18. Búsquedas complejas con documentos que incluyen arrays y objetos:

Comando:

```
db.alumno.find({"preferencias.seguimientoEmails":false,"aficiones":{"$ne":"senderismo"}})
```

Resultado:

```
Universidad> db.alumno.find({"preferencias.seguimientoEmails":false,"aficiones":{"$ne":"senderismo"}})
[
  {
    _id: ObjectId('6765277211ee83f03ae24c43'),
    edad: 24,
    nombre: 'Andrea',
    aficiones: [ 'boxeo', 'baile', 'lectura' ],
    email: 'Andrea@gmail.com',
    preferencias: { seguimientoEmails: false, idioma: 'Castellano', zonaHoraria: 2 }
  }
]
```