

Comandos básicos en MongoDB	
Comando	Descripción
<code>show dbs</code>	Muestra las bases de datos disponibles en el servidor, es decir, bds no vacías.
<code>use [nombre BD]</code>	Permite conectarse a una base de datos; en el caso de que esta no exista, la crea.
<code>db</code>	Muestra la base de datos a la que actualmente se encuentra conectada.
<code>show collections</code>	Muestra las colecciones existentes en la bd actual
<code>db.createCollection</code> ( <i>"nombreColeccion"</i> )	Forma explícita para crear una colección con el nombre específico dentro de la base de datos.
<code>db.nombreColeccion.insertOne(</code>  <code>{ "clave1": "valor1",</code>  <code>"clave2": "valor2" ...,</code>  <code>"claveN": "valorN" } )</code>	Inserta un nuevo documento en la colección. El documento tiene la estructura de pares clave-valor Si la colección nombrada no existe, la crea e inserta el documento en su interior. Si desea hacer varias inserciones en una sola transacción, debe usar el comando <code>insertMany</code> .
<code>db.dropDatabase()</code>	Elimina la base de datos actual.
<code>db.nombreColeccion.find()</code>	Muestra todos los documentos de la colección en formato estándar
<code>db.nombreColeccion.find().pretty()</code>	Muestra todos los documentos de la colección en formato json ordenado y legible.
<code>db.nombreColeccion.find({"clave":</code>  <code>"valor"}).pretty()</code>	Muestra todos los elementos de la colección que cumplan con la condición (igualdad) de que la clave tenga el valor específico dado.
<code>db.nombreColeccion.find().limit(1)</code>	El método <code>limit(N)</code> limita la cantidad de registros a mostrar en la consulta
<code>db.nombreColeccion.find().sort({</code> <code>clave:1})</code>	El método <code>sort</code> permite ordenar la consulta a partir del campo específico. El parámetro 1 indica ordenamiento ascendente. El parámetro -1 indica ordenamiento descendente. Se pueden agregar varios campos de ordenamiento separándolos por coma.
<code>db.nombreColeccion.updateOne(</code> <code>{ "clave": "valo</code>  <code>r"}, { \$set: { "claveN": valorN } }</code>	Actualiza el primer documento de la colección dado que cumpla con la condición <code>"clave": "valor"</code> y la modificación a realizar sería asignar a la <code>"claveN": valorN</code> . Si el campo a asignar no existe, lo agrega. Si desea actualizar todos los documentos que cumplan con el criterio, debe usar el comando <code>updateMany</code> .
<code>db.nombreColeccion.updateOne(</code> <code>{ "clave": "valor</code>  <code>"}, { \$unset: { "claveN": "" } }</code>	Actualiza el documento de la colección si cumple con la condición <code>"clave": "valor"</code> y la modificación a realizar sería la eliminación del campo <code>"claveN"</code> .
<code>db.nombreColeccion.deleteOne({</code> <code>"clave" :</code>  <code>"valor"});</code>	Elimina el primer documento de la colección que cumpla con la condición de igualdad de la clave y el valor provistos en el comando. Si desea eliminar todos los documentos que cumplan con el criterio, deberá usar el comando <code>deleteMany</code> .
<code>db.colenombreColeccion.drop();</code>	Elimina la colección nombrada en el comando junto con todos los documentos que tuviera dicha colección. Devuelve el valor de <code>true</code> o <code>false</code> dependiendo de si logró o no realizar la eliminación de la colección. En el entorno de producción, este comando es de mucho cuidado, ya que una vez se ejecuta no hay forma de deshacerlo