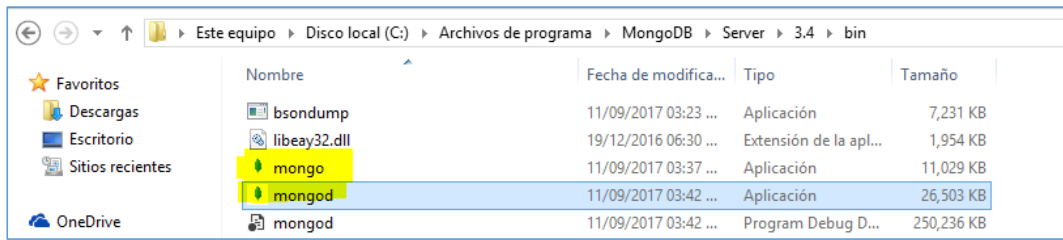


I. Primera Ejecución de Mongo



1. Ubicar el archivo mongo y mongod, por defecto en la dirección:

C:\Program Files\MongoDB\Server\3.4\bin

o la dirección seleccionada al momento de la instalación

2. Ejecutar  (doble click), luego  repetir los mismos pasos del anterior, se levantara la ventana de comandos de mongo así

```
C:\Program Files\MongoDB\Server\3.4\bin\mongo.exe
MongoDB shell version v3.4.9
connecting to: mongod://127.0.0.1:27017
MongoDB server version: 3.4.9
Server has startup warnings:
2017-11-14T17:05:31.450-0500 I CONTROL [initandlisten]
2017-11-14T17:05:31.450-0500 I CONTROL [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2017-11-14T17:05:31.451-0500 I CONTROL [initandlisten] **           Read and wri
te access to data and configuration is unrestricted.
2017-11-14T17:05:31.452-0500 I CONTROL [initandlisten]
```

II. Comandos de Manipulación de Datos

1. **use:** sirve para ubicarse una base de datos existente o si es que no existe la crea, en este caso la base de datos “mitiendita”

```
mongo
>
>
>
> use mitiendita
switched to db mitiendita
>
```

2. **db:** nos muestra en que base de datos nos encontramos

```
mongo
>
>
> db
mitiendita
>
```

3. **show dbs:** Muestra todas las bases de datos creadas, pero no se observa la base de datos "mitiendita", porque no contiene registros ingresados.

```
mongo
>
>
>
> show dbs
DemoDb  0.000GB
admin   0.000GB
local   0.000GB
>
```

4. Creación de Colecciones(tablas)

- a. **Implicitas:** `db.usuarios.insert({ "id" : "1", "name" : "jhon", "age" : "18" })` crea las colecciones de manera implícitas (para el caso de Mongodb son tablas, es decir tabla usuario para este ejemplo) además sus campos y registros van en comillas dobles y separado por comas.

```
mongo
> db.usuario.insert(<<
...   "cedula":"15559",
...   "name":"Juan Perez",
...   "clave":"35741",
...   "pais":"Peru"
... >>)
WriteResult(<< "nInserted" : 1 >>)
>
```

- b. **Explícitas:** `db.createCollection("Productos")`

```
mongo
>
>
>
> db.createCollection("Productos")
{ "ok" : 1 }
>
```

Luego visualizamos lo realizado, y se notará que aparece la colección **"Productos"**, cabe mencionar la ejecución de comandos son sensibles a mayúsculas y minúsculas

```
mongo
>
>
>
> show collections
Productos
usuario
>
```

5. **db.productos.drop()**, elimina la colección al ejecutar este comando, si es que es exitosa la ejecución arrojará **true** como resultado, caso contrario será **false**



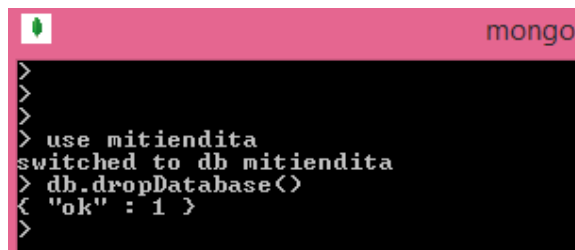
```
mongo
>
>
> db.Productos.drop()
true
>
```

Luego visualizamos lo realizado, y se notará que no aparece la colección **"Productos"**, cabe mencionar la ejecución de comandos son sensibles a mayúsculas y minúsculas



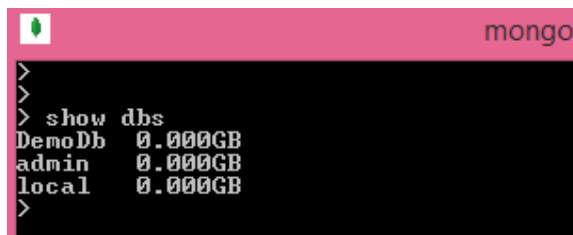
```
mongo
>
>
> show collections
usuario
>
```

6. Eliminar una base de datos
 - a. Ubicarse en la base de datos a eliminar con el comando **"use database"** para el caso del ejemplo sería la base de datos **"mitiendita"**
 - b. Ejecutar el comando para eliminar **db.dropDatabase()**, si la ejecución fue exitosa mostrará **"Ok"** como resultado



```
mongo
>
>
> use mitiendita
switched to db mitiendita
> db.dropDatabase()
{ "ok" : 1 }
>
```

Luego visualizamos lo realizado, y se notará que no aparece la base de datos **"mitiendita"**, cabe mencionar la ejecución de comandos son sensibles a mayúsculas y minúsculas



```
mongo
>
>
> show dbs
DemoDb  0.000GB
admin   0.000GB
local   0.000GB
>
```

7. Listar los registros “**find()**” de manera estructurada mediante el comando “**pretty()**” de la siguiente manera **db.productos.find().pretty()**

```
mongo - Acceso directo
>
> db.productos.find().pretty()
{
  "_id" : ObjectId("5a0bb2be922b4b19971f1fd4"),
  "id" : "1",
  "nombre" : "Camiseta S",
  "valor" : "18.5",
  "stock" : "2"
}
{
  "_id" : ObjectId("5a0bb2f9922b4b19971f1fd5"),
  "id" : "2",
  "nombre" : "Pantalon M",
  "valor" : "20.5",
  "stock" : "6"
}
{
  "_id" : ObjectId("5a0bb317922b4b19971f1fd6"),
  "id" : "3",
  "nombre" : "Polo L",
  "valor" : "10.0",
  "stock" : "3"
}
>
>
```

8. Ingresar un documento o registro en una colección
- a. Ingresar un registro en una nueva tabla o colección que haga referencia a un documento o registro de otra tabla o colección de la siguiente manera

```
db.usuarios.insert({ "cedula": "0999999999", "name": "Pepito", "clave": "123456789", "productos": [1], "n_ordenes": 1 })
```

"**productos**": [1], es el registro o documento de la tabla o colección “**productos**”

```
mongo
>
>
> db.usuarios.insert({
...   "cedula": "0999999999",
...   "name": "Pepito",
...   "clave": "123456789",
...   "productos": [1],
...   "n_ordenes": 1
... })
WriteResult<< "nInserted" : 1 >>
>
```

Luego visualizamos lo realizado, y se notará que aparece la tabla o colección **"productos": [1]** cabe mencionar la ejecución de comandos son sensibles a mayúsculas y minúsculas

```
mongo - Acceso directo
>
>
> db.usuarios.find().pretty()
{
  "_id" : ObjectId<"5a0bb772922b4b19971f1fd7">,
  "cedula" : "0999999999",
  "name" : "Pepito",
  "clave" : "123456789",
  "productos" : [
    1,
  ],
  "n_ordenes" : 1
}
```

9. Editando un registro o documento

db.productos.update({ "id": "1" }, { \$set:{'valor':20.45} })

```
mongo - Acceso directo
>
> db.productos.update<<
...   "id": "1"
... >,
... {
...   $set:{'valor':20.45}
... }>>
WriteResult<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>
```

Luego visualizamos lo realizado, y se notará que ha cambiado el campo valor.

```
mongo - Acceso directo
>
> db.productos.find().pretty()
{
  "_id" : ObjectId<"5a0bb2be922b4b19971f1fd4">,
  "id" : "1",
  "nombre" : "Camiseta S",
  "valor" : "18.5",
  "stock" : "2"
}
```

De **"valor": "18.5"** a **"valor": "20.45"**, cabe mencionar la ejecución de comandos son sensibles a mayúsculas y minúsculas

```
mongo - Acceso directo
>
> db.productos.find().pretty()
{
  "_id" : ObjectId("5a0bb2be922b4b19971f1fd4"),
  "id" : "1",
  "nombre" : "Camiseta S",
  "valor" : 20.45,
  "stock" : "2"
}
```

Nota: Cabe señalar que el comando:

db.productos.update({ "id": "1" }, { \$set:{'valor':20.45} })), sólo actualizará un solo documento.

Si se quiere actualizar más de un documento o registro que cumpla con la condición se debe agregar **"{multi: true}"** al comando de la siguiente manera:

db.productos.update({ "id": "1" }, { \$set:{'valor':20.45}, {multi: true}})

10. Eliminar un registro o documento

db.productos.deleteOne({"id": "1"})

```
mongo - Acceso directo
>
>
> db.productos.deleteOne<{
...  "id": "1"
... }>
{ "acknowledged" : true, "deletedCount" : 1 }
```

Luego visualizamos lo realizado, y se notará que ha cambiado la lista de documentos o registros

```
mongo - Acceso directo
>
> db.productos.find().pretty()
{
  "_id" : ObjectId("5a0bb2be922b4b19971f1fd4"),
  "id" : "1",
  "nombre" : "Camiseta S",
  "valor" : 20.45,
  "stock" : "2"
}
{
  "_id" : ObjectId("5a0bb2f9922b4b19971f1fd5"),
  "id" : "2",
  "nombre" : "Pantalon M",
  "valor" : 20.5,
  "stock" : "6"
}
{
  "_id" : ObjectId("5a0bb317922b4b19971f1fd6"),
  "id" : "3",
  "nombre" : "Polo L",
  "valor" : 10.0,
  "stock" : "3"
}
```

En la cual se nota la ausencia del "id": "1"

```
mongo - Acceso directo
>
> db.productos.find().pretty()
{
  "_id" : ObjectId("5a0bb2f9922b4b19971f1fd5"),
  "id" : "2",
  "nombre" : "Pantalon M",
  "valor" : 20.5,
  "stock" : "6"
}
{
  "_id" : ObjectId("5a0bb317922b4b19971f1fd6"),
  "id" : "3",
  "nombre" : "Polo L",
  "valor" : 10.0,
  "stock" : "3"
}
```

11. Consultas con el comando find()

- a. Si campo "valor" es igual a "20.5", entonces el comando será

```
db.productos.find({ "valor": "20.5" }).pretty(),
```

```
mongo - Acceso directo
>
> db.productos.find(<< "valor": "20.5" >>).pretty()
{
  "_id" : ObjectId("5a0bb2f9922b4b19971f1fd5"),
  "id" : "2",
  "nombre" : "Pantalon M",
  "valor" : 20.5,
  "stock" : "6"
}
```

- b. Si campo “valor” es menor a “40.45”, entonces el comando será, comando de comparación menor que **lt = less than**

```
db.productos.find( { valor: { $lt: 40.45 } } ).pretty()
```



The screenshot shows a terminal window titled "mongo - Acceso directo". The command `db.productos.find({ valor: { $lt: 40.45 } }).pretty()` has been executed. The output is a JSON document representing a product with the following fields: `_id` (ObjectId), `id` (2), `nombre` (Pantalon M), `valor` (20.5), and `stock` (6).

```
>
>
> db.productos.find( { valor: { $lt: 40.45 } } ).pretty()
{
  "_id" : ObjectId("5a0bb2f9922b4b19971f1fd5"),
  "id" : "2",
  "nombre" : "Pantalon M",
  "valor" : 20.5,
  "stock" : "6"
}
```

- c. Si campo “valor” es mayor a “20.5”, entonces el comando será, comando de comparación mayor que **gt = greater than**

```
db.productos.find( { valor: { $gt: 20.5 } } ).pretty()
```



The screenshot shows a terminal window titled "mongo - Acceso directo". The command `db.productos.find({ valor: { $gt: 20.5 } }).pretty()` has been executed. The output is a JSON document representing a product with the following fields: `_id` (ObjectId), `id` (3), `nombre` (Polo L), `valor` (40.45), and `stock` (3).

```
>
>
> db.productos.find( { valor: { $gt: 20.5 } } ).pretty()
{
  "_id" : ObjectId("5a0bb317922b4b19971f1fd6"),
  "id" : "3",
  "nombre" : "Polo L",
  "valor" : 40.45,
  "stock" : "3"
}
```