

Instalacion en MongoDB

Instalación en Windows:

Existen tres versiones de MongoDB para Windows:

- MongoDB para Windows Server 2008 R2 es el único que corre en Windows Server 2008 R2, Windows 7 de 64-bit, y nuevas versiones de Windows.
- MongoDB para Windows 64-bit corre en cualquier versión de Windows (desde Windows XP) incluyendo Windows Server 2008 R2 y Windows 7.
- MongoDB para Windows 32-bit corre en cualquier versión de Windows (desde Windows XP) de 32 bits. NOTA: SOLO SOPORTA BASE DE DATOS MENORES A 2GB).

Para saber que versión de Windows tiene en su computador en la línea de comandos que este corriendo como administrador digite la siguiente instrucción:

- `wmic os get osarchitecture`

Descargar la versión en .zip deseablemente que se acople al sistema operativo que usted disponga en su equipo.

- <https://www.mongodb.org/downloads>

Después de descargar el archivo y descomprimirlo ubicar la carpeta preferiblemente en C:, luego para tener una mejor organización abrimos la línea de comando nos ubicamos en C: y ejecutamos el siguiente código:

- `Move c:\mongodb-win32-* C:\mongodb`

Mongodb requiere un directorio de archivos donde se almacenan todos los datos que por default esta en C:\data\db para crearlo desde la línea de comando ejecutamos el codigo:

- `md C:\data\db`

Ahora se debe proceder a ejecutar el servidor con el siguiente comando:

- `C:\mongodb\bin\mongod.exe`

En el caso que se quiera especificar otro directorio de archivos ejecutamos mongod.exe con la opción `-dbpath`. Por ejemplo:

- `C:\mongodb\bin\mongod.exe -dbpath d:\test\mongodb\data`

Luego procedemos a abrir otra línea de commando como administrador para conectarnos a mongo db a través de mongo.exe sino se ha modificado el directorio de archivos no debemos usar la opción `–dbpath` en caso contrario utilizarlo especificando la misma ruta que mongod.exe:

- `C:\mongodb\bin\mongo.exe`

Después de ejecutar mongo.exe podemos empezar a trabajar en las bases de datos.

Comandos Básicos en Mogodb

Primero se debe seleccionar la base de datos donde deseamos trabajar.

Para saber en qué base de datos nos encontramos utilizamos el siguiente comando:

- `db`

Para crear una base de datos o seleccionar una base de datos utilizamos el siguientes comando:

- `use nombre`

Para ver las bases de datos existentes utilizamos el comando:

- `show dbs`

En caso de necesitar ayudar y ver la documentación de un comando utilizamos el comando:

- `help`

En mogodb no se guardan los datos en una tabla sino en un colección el cual agrupa un número indeterminado de documentos que vienen a ser los registros si lo comparamos con las bases de datos relaciones.

Los documentos tienen el formato JSON el cual cumple el siguiente formato:

```
{
  atributo1: "valor",
  atributo2: numero,
  atributo3: {
    atributo 3.1: "valor",
    atributo 3.2: numero
  }
}
```

Para crear documentos en mongodb lo hacemos de la siguiente forma:

- `j = { name : "mongo" }`
- `k = { x : 3 }`

Luego de crear los documentos los insertamos a una colección la cual si no existe es creada automáticamente:

- `db.testData.insert(j)`
- `db.testData.insert(k)`

Para observar las colecciones que existen en la base de datos lo hacemos con el comando:

- `show collections`

Para confirmar que documentos tenemos dentro de la colección lo hacemos mediante el comando:

- `db.testData.find()`

Podemos crear varios documentos mediante un for loop de la siguiente manera:

- `for (var i = 1; i <= 25; i++) db.testData.insert({ x : i })`

Para observar los documentos que acabos de ingresar es:

- `db.testData.find()`

Podemos crear varios documentos mediante una función en el Shell:

```
function insertData(dbName, colName, num) {  
  
    var col = db.getSiblingDB(dbName).getCollection(colName);  
  
    for (i = 0; i < num; i++) {  
        col.insert({x:i});  
    }  
  
    print(col.count());  
  
}
```

Para llamar la función lo hacemos de la siguiente forma:

- `insertData("test", "testData", 400)`

Para poder iterar dentro de una colección lo podemos hacer de la siguiente manera:

- `var c = db.testData.find()`
- `while (c.hasNext()) printjson(c.next())`

También podremos ver un documento mediante la posición en el vector:

- `printjson(c [4])`

Consultar un documento específico lo logramos con el siguiente comando:

- `db.testData.find({ x : 18 })`