

# Empezando con MongoDB en tu PC: una BD noSQL de documentos

Profesor: Héctor Gómez Gauchía

## PASOS para INSTALAR MONGODB y ROBOMONGO (ROBO 3T) en WIN

### RESUMEN:

1-) Descargamos MongoDB de la página oficial, asegurarnos de bajar la versión para nuestra versión de Windows. Es el servidor de BD.

----- Dos ejecutables importantes:

Servidor: mongod.exe y Cliente modo línea comando: mongo.exe

2-) Descargamos el editor Robomongo de la página oficial de Robomongo (Actualmente ROBO 3T), es el entorno desarrollo que usamos. Es un cliente del servidor de MongoDB

3-) Instalamos MongoDB y Robomongo. Acordarnos de la ruta donde instalamos MongoDB.

4-) Arrancar el servidor. Arrancar el editor. Hacer una conexión con el servidor. Restaurar una BD.

### DESCRIPCION DETALLADA de la INSTALACION

→ INSTALAR ESTA VERSION: MongoDB Community Server

- Ver Instrucciones en:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition>

----- ENTRA en <https://www.mongodb.com/download-center/community>

----- Descarga el Paquete MSI: con installer

[https://fastdl.mongodb.org/win32/mongodb-win32-x86\\_64-2012plus-4.2.6-signed.msi](https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2012plus-4.2.6-signed.msi)

----- en WIN: Ejecuta el .msi: ---- Setup :

---- Dos modos de instalar el servidor :

- a) - Se puede instalar como un servicio de WIN:

- Si te manejas bien para poder pararlo cuando no lo uses.

- b) - Como una aplicación para CMD y arrancar a mano:

- Lo arrancas y paras cuando lo necesites

--- Recomiendo b)

---- Puedes instalar Compass ahora (no lo usaremos para empezar) : deja el check

- Tarda en prepararse

- Te dice que tiene que cerrar el firefox

- Lo tengo que cerrar a mano.

- Termina y se abre el Compass : Cierrálo.

## PASOS del USO HABITUAL

### \*\*\* ---- Arrancar MongoDB Community Edition DESDE el Interprete CMD

- Abrir CMD.exe como ADMINISTRADOR
- Crear el Dir de trabajo donde almacena los datos:  
cd Disco: (donde va a estar)  
md mongoWork
- ASIGNA PATH donde está el servidor:  
set path=C:\Program Files\MongoDB\Server\4.2\bin\;%path%
- Arranca el Servidor con el directorio de trabajo que quieras: Ejecutar  
> mongod.exe --dbpath A:\mongoWork  
  
-->> HASTA que aparece entre los últimos mensajes:  
I NETWORK [listener] Listening on 127.0.0.1

### \*\*\* ---- Arrancar el Editor Robomongo (Robo3T) y crea una conexión : todas las BDs dentro

#### \*\*\* Otro cliente en modo línea de comandos: mongo

- Ejecutar desde otra ventana del CMD : mongo  
Antes: debes haber asignado el *path* correcto como arriba

#### \*\*\* -- Parar MongoDB desde un cliente como Robo3T:

- use admin
- db.shutdownServer()

### \*\*\* IMPORTAR BD.- Si quieres importar alguna base de datos y crear automáticamente esa BD:

#### 1.- Abres una ventana CMD:

- Te sitúas en el directorio raíz donde tienes el dir. con tu BD , ejemplo:  
Si tienes la práctica en a: \ . . . \prac-MongoDB-para-CV\dump-para-empezar-prac\ HACES:  
X: > a:  
A:> cd . . . \prac-MongoDB-para-CV

#### 2.- asignas el path donde esté instalado el servidor:

A:> set path=C:\Program Files\MongoDB\Server\4.2\bin\;%path%

#### 3.- luego ejecutas la recuperación del dump: (aparece en la BD *pracmongo*)

A:> mongorestore --db pracmongo dump-para-empezar-prac

4.- Si tenías funcionando Robo3T y Mongoddb antes de recuperar la BD, para ver esta nueva BD, tienes que hacer refresh con botón dcho encima de la conexión en Robo3T

## FUNCIONES BASICAS DE MONGO

### Consultar y Crear una BD.

En la variable **db** se almacena la BD actual, es un objeto *database*  
MongoDB siempre se inicia con la BD "test" por defecto

```
> show dbs           // da los nombres de todas las BDs que hay
show collections     // da los nombres de todas las colecciones en la BD actual (en db) SOLO si tienen docs.
➔ LINK interesante: documentación http://docs.mongodb.org/manual/

> db.getName()       // devuelve el nombre de la BD actual sobre la que ejecuta los comandos
> db.help(); <<<     // Ayuda de todos los métodos aplicables al objeto en var. db
> db.test.help(); <<< // Ayuda de todos los métodos aplicables al objeto colección "test"

> db.getMongo();     // devuelve el objeto connection: tiene métodos útiles asociados a él

> use midb           // crear una BD con ese nombre y la asigna como "actual"
> miconn= db.getMongo(); // obtener la conexión en una variable para poder llamar a su métodos
                        // también se puede usar como objeto "db"
> nuevaDB=miconn.getDB("midb"); // asignamos la Base de Datos "midb" a la variable: es un objeto database
                        // También dispone de métodos útiles

> nuevaDB.getName(); // nos da el nombre de la BD que tiene en la var nuevaDB
```

### Crear un objeto *collection* (dentro de una BD)

Para almacenar datos dentro de la BD que apunta la variable *db*

```
> use testData      // si ya existe la coloca como actual. Si no existe, la crea y asigna como actual
> db.createCollection("palabras"); // crea una colección (las comillas deben ser verticales ", no de abrir o cerrar: " ")
> coll = db.getCollection("palabras"); // almacena la referencia a la colección en la var. coll
test.palabras

> miscoles = db.getCollectionNames(); // almacena un objeto con los nombres de colecciones
> printjson(miscoles); // imprime los nombre
[ "palabras" ]

> db.createCollection("nuevaColeA"); // crea otra colección
{ "ok" : 1 }
> db.createCollection("nuevaColeB");
{ "ok" : 1 }
> print("Después de crear colección:"); // imprime texto
Después de crear colección

> collections = db.getCollectionNames(); // responde con una lista de objetos
[ "nuevaColeA", "nuevaColeB", "palabras", "system.indexes" ]

> printjson(collections); // imprime json de los objetos
[ "nuevaColeA", "nuevaColeB", "palabras", "system.indexes" ]

> db.createCollection("nuevaColeA"); // intenta crear una col. que ya existe: da un obj con mensaje
{ "ok" : 0, "errmsg" : "collection already exists" }

> printjson(db.createCollection("nuevaColeB")); // imprime json del contenido del objeto de error
{ "ok" : 0, "errmsg" : "collection already exists" }

> print("Después de crear últimas colecciones");
Después de crear últimas colecciones

> miscoles = db.getCollectionNames(); // devuelve un objeto con los nombres
```

```
[ "nuevaColeA", "nuevaColeB", "palabras", "system.indexes" ]
```

```
> printjson(miscoles); // imprime el contenido del objeto
```

```
[ "nuevaColeA", "nuevaColeB", "palabras", "system.indexes" ]
```

## **Crear documentos en la colección “palabras” de la db**

Ayuda para escribir bien la sintaxis es validar json en Robomongo:

botón dcho sobre una colección + “Insert Document”

Se abre una ventana editor de texto. Escribes el documento { . . . } y botón “validate” (antes de salvar “save”)

```
> db.palabras.insert( // creamos un documento con la palabra “tomate”
{ "_id":{"str":"52d87454483398c8f24222"}, // si no das tú el valor, genera un valor único (índice y clave únicos)
  "palabra":"tomate",
  "primera":"t",
  "ultima":"e",
  "tamaño":6,
  "letras":["t","o","m","a","t","e"],
  "estadis": {"vocales":3, "consonantes":3},
  "caractsets":[
    {"tipo":"consonantes","caracts":["t","m"]},
    {"tipo":"vocales","caracts":["e","a"]}
  ]
} )
```

// responde : insertado un registro

```
> // SI REPETIMOS el insert
```

```
E11000 duplicate key error collection: midb.palabras index: _id_ dup key: { : { str: "52d87454483398c8f24222" } }
```

```
> db.palabras.count(); // consulta cuántos docs hay en la colección “palabras”
```

```
1
```

```
db.palabras.insert(
{ "_id":{"str":"52d87454483398c8f2429277"},
  "palabra":"the",
  "primera":"t",
  "ultima":"e",
  "tamaño":3,
  "letras":["t","h","e"],
  "estadis": {"vocales":1, "consonantes":2},
  "caractsets":[
    {"tipo":"consonantes","caracts":["t","h"]},
    {"tipo":"vocales","caracts":["e"]}
  ]
} )
```

## **→ Cómo puedo validar la sintaxis JSON de un documento que he escrito a mano?**

En el Robo3T, encima de la colección que quieras:

Hacer “insert document”: se abre una ventana par escribir el documento

Abajo tiene el botón de “validate”

## **Buscar documentos**

ANTES hemos ejecutado esta instrucción para almacenar la referencia a la colección en la var. coll  
coll = db.getCollection("palabras");

```
> coll.find() // da todos los objetos en la colección apuntada por coll, en este caso "palabras"

> sale= coll.find ( { "tamaño" : 3, "ultima" : "e" }, {}); // "the" tiene de tamaño 3

> sale= coll.find ( { "tamaño" : 6, "ultima" : "e" }, {}); // "tomate" tiene de tamaño 6

> sale= coll.find ( { "tamaño" : { $gt: 0, $lt: 4 }, "ultima" : 'e' }, {});

> sale= coll.find ( { "tamaño" : { $gt: 0, $lt: 4 }, "ultima" : 'e' }, {palabra: 1}); // da solo el campo "palabra"

> if (sale) {print("fue bien")} ; // compruebo si ha ido bien la instrucción anterior
```

- ➔ **Mas ejercicios: Puedes probar búsquedas con la BD aficion-restaurantes.json de la práctica antes, ver como importarla a continuación**

## **Para llevar a tu casa la BD entera: mongodump y mongorestore**

### **1.- Salvar Toda la BD "test" al directorio A:\mongoWork**

```
mongodump --db test --out A:\mongoWork
```

----- crea un directorio con dos ficheros, uno es el de metadatos

```
C:\Users\misuario> mongodump --db test --out A:\mongoWork
```

```
2016-05-06T17:52:14.384+0200  writing test.esp-restaurantes to
2016-05-06T17:52:14.387+0200  done dumping test.esp-restaurantes (15 documents)
```

➔ *Algunos parámetros útiles:*

si queremos que cree un archivo solo --archive <nombreFichero>

Para comprimir --gzip

### **2.- mongorestore : para incluir en MongoDB un BD o colección generada con mongodump**

- mongorestore crea una BD o añade a la que ponemos de destino

- NO actualiza los documentos si coinciden sus \_Id únicos

- SI reconstruye los índices

### **3.- Recupera toda la BD de dump del path indicado en la bd indicada**

```
mongorestore --db midb A:\mongoWork\test (la BD test puede tener varias colecciones)
```

```
C:\Users\misuario> mongorestore --db midb A:\mongoWork\test
```

```
2016-05-06T18:10:20.105+0200  building a list of collections to restore from A:\mongoWork\test dir
2016-05-06T18:10:20.108+0200      reading metadata for midb.esp-restaurantes from A:\mongoWork\test\esp-
restaurantes.metadata.json
2016-05-06T18:10:20.191+0200  restoring midb.esp-restaurantes from A:\mongoWork\test\esp-restaurantes.bson
2016-05-06T18:10:20.195+0200  restoring indexes for collection midb.esp-restaurantes from metadata
2016-05-06T18:10:20.196+0200  finished restoring midb.esp-restaurantes (15 documents)
2016-05-06T18:10:20.198+0200  done
```

#### 4.- Recupera una sola colección en la bd indicada

```
C:\Users\misuario> mongorestore --collection dump-restaurantes --db nada A:\mongoWork\test\esp-restaurantes.bson
```

```
2016-05-06T18:06:24.548+0200 checking for collection data in A:\mongoWork\test\esp-restaurantes.bson
2016-05-06T18:06:24.551+0200 reading metadata for nada.dump-restaurantes from A:\mongoWork\test\esp-
restaurantes.metadata.json
2016-05-06T18:06:24.637+0200 restoring nada.dump-restaurantes from A:\mongoWork\test\esp-restaurantes.bson
2016-05-06T18:06:24.699+0200 restoring indexes for collection nada.dump-restaurantes from metadata
2016-05-06T18:06:24.727+0200 finished restoring nada.dump-restaurantes (15 documents)
2016-05-06T18:06:24.728+0200 done
```

#### 5.- Importar y exportar en la BD una colección restaurantes (mejor usar el dump y restore)

- **Exportar desde el CMD, sin cliente:** (NO desde dentro del intérprete MongoDB)

```
mongoexport --db test --collection esp-restaurantes --out misrestaurantesx.json
```

- **Importar, En el dir donde están tus datos de la BD**

```
mongoimport --db test --collection restaurantes --drop --file E:\ABD-campusCV\misrestaurantes.json
```

```
mongoimport --db otraBD --collection restaurantes --drop --file D:\robomongo\rombodatos\misrestaurantesx.json
```

### **MAS UTILIDADES PRÁCTICAS**

----- Constructores: Comandos y funciones útiles (DENTRO DE LA SHELL)

#### -- Nativos del MS-DOS

- cd(): cambia el working directory. EJ: Cambio al dir local donde tengo mis datos  
cd("H:/robomongo/rombodatos")
- pwd() para confirmar
- load() carga y ejecuta un fich JavaScript
- mkdir()
- pwd() muestr el work. dir.
- removeFile borra fichero

#### Fuciones Propias de MongoDB

- Date() crea el obj, date con fecha de hoy
  - db.traffic.insert( { \_id: 1, volume: NumberLong('2980000'), date: new Date() } )
- ObjectId.toString()
- \$dateToString

**Uso de un aggregate para devolver el campo fecha en forma de strings formateadas usando \$dateToString**

Si tengo este documento en db.micole

```
{
  "_id" : 1,
  "nombre" : "abc",
  "precio" : 10,
  "cantidad" : 2,
  "miFecha" : ISODate("2014-01-01T08:15:39.736Z")
}
```

Puedo formatear partes de la fecha:

```
db.micole.aggregate(
[
  {
    $project: {
      AñoMesDía: { $dateToString: { format: "%Y-%m-%d", miFecha: "$date" } },
      hora: { $dateToString: { format: "%H:%M:%S:%L", miFecha: "$date" } }
    }
  }
]
```

