

Empezando con MongoDB: una BD noSQL de documentos

Profesor: Héctor Gómez Gauchía

Empezando en el Laboratorio

(al final del documento se describe como hacerlo en tu PC o portatil)

0.- IMPORTANTE : tus datos se almacenan en disco local del PC, **que se borra todos los días**

- Recuerda salvar tu BD antes de irte del laboratorio: ver "Para llevar a tu casa la BD entera": botón exportar (mongodump)
- NOTA: Hay una alternativa NO RECOMENDADA: usar un servidor público: pero todo el mundo y la puede modificar o borrar o copiar.

1.- Inicia el servidor ejecutando en el Laboratorio: (cada PC tiene un servidor)

Ejecuta *Todos los Programas -> BBDD -> MongoDB -> Servidor MongoDB*

Se abre una ventana cmd de Windows.

Esperar a que aparezca en los mensajes, una línea para estar preparado:

. . . waiting for connections . . . "port" :27017 y otros mensajes (no hay todavía conexiones)

2.- Inicia el Cliente ejecutando en el Laboratorio: Studio3T

Modo gráfico: ejecuta *Studio3T* en: *Todos los Programas -> BBDD->Studio3T*

- Permite visualizar las BDs y Colecciones en una ventana
- Valida sintaxis. Completa comandos (también la línea de comandos)
- Aunque todavía debes usar comandos de línea para hacer lo que se pide en las Prácticas

3.- Conectar Cliente al servidor ejecutando en el Laboratorio: Studio3T

Puede que Pida Credenciales : Si ha caducado, las pide y luego continúa con la versión "Free"

Dentro del entorno Studio3T:

1.- *Crear una conexión:* botón: Connect se abre en ventana "Connection Manager"

- o "New Connection" + "Manually . ." + "Next": abre ventana con parámetros, dejar los que tiene por defecto: "localhost", puerto 27017 y en "Connection name" poner un Nombre cualquiera >> botón Save >> botón RUN
- o Se añade a la lista de conexiones
- o Escoge la conexión si quiere conectarte: haz click en *connect*

4.- Importar tu BD y exportarla para lleva a casa: en Estudio3T

→ Importa la BD: /pracmongo/ siguiendo la sección *Botón IMPORT (Studio3T)* . Contine la colección *aficiones*.

- Lo hacemos para empezar a trabajar en la práctica y siempre que cambies de PC

_____ Botón IMPORT _____

EJEMPLO: Importar mi BD aficiones

- Botón IMPORT
- Escojo "BSON – mongodump . . ."
- Seleccionar *carpeta raíz o root:* /dump-para-Studio3T/
 - Contiene la carpeta de la BD, /pracmongo/
 - Que contine la colección "aficiones", que está formada por dos ficheros, uno .json y otro .bson
- Crea la BD y la colección:
- Importante: Comprobar en ventana OPERACIONES, que ha importado todos los documentos
- EJ: en "pracmongo done/aficiones done" comprobar que tiene: finished . . . (NN documents, 0 failed)

_____ Botón EXPORT _____

-> comprobar que está seleccionada la BD deseada, EJ: /pracmongo/

- Botón EXPORT
- Escojo opción: "BSON - mongodump"
- Botón: Configure --> abre pestaña: Export
- Dejar el tic en: BSON - mongodump folder
- Quitar el tic de: Enable Compression (porque es una BD muy pequeña)
- Escoger una carpeta para el dump (que será la raíz o root cuando la quiera importar):
Destination folder EJ: /dump-para-Studio3T/
- Botón: Run
- Comprobar en ventana OPERACIONES, que ha exportado todos los documentos que querías, EJ.: " 89 documents done"

5.- Al terminar cada sesión de trabajo: Parar el Servidor

- Siempre, antes de Terminar, Exporta tu BD a otro soporte permanente: tu USB, drive, etc...
- Parar servidor MongoDB:
 - use admin // O bien: abrir una IntelliShell encima de la BD admin del Studio3T
 - db.shutdownServer() // en la shell que has abierto

6.- Trabajando con la IntelliShell

- 1.- Abrir un shell IntelliShell (botón dcho sobre la BD, así se queda asignada a esa ventana):
 - es un intérprete de JavaScript para la BD
 - Completa comandos pulsando TAB
 - Uso de la IntelliShell: Sigo PRAC5enu/empezando. . .
 - escribir sin ";" al final y empieza a escribir en 1ª posición
 - separa las respuestas con una línea
 - Las tres vistas :Json, Tree, Table
 - Cómo ver el contenido de una colección de la bd /local/?
click en startup_log (dialogo: no) db.getCollection("startup_log").find({})
- 2.- Ejecutar un comando después de escribirlo:
 - Si el cursor está pegando al principio o al final del comando: se ejecuta con CTRL + RET
Si está en otro lugar da un error de que no hay comando en esa línea.
 - Ejecutar todos los comandos escritos en el buffer: Icono de la flecha (el de la izq)
- 3.- INSERT un documento
 - a).- B.Dcho encima de la colección >> Add/Edit Validator JSON >> abre ventana
Lo escribo pulsando en boton Validate JSON hasta que no tenga errores (Save NO es para insertar en la colección)
 - b).- (En IntelliShell) Escribo la instrucción Insert y copio el documento validado


```
db.aficiones.insert({
  "Tema": "Libros",
  "Apodo": "XXXX",
  "Nombre": "Stephen wolf",
  "Puntuacion": 1.12,
  "Precio": 111.9,
  "genero": "Novela",
  "autor": "Who Knows",
  "paginas": 1111.0,
  "ISBN": 9.788401339639E12,
  "ValorCalidad": 40.93,
  "Descuento": 1.1
})
```
 - c).- Comprobar la ejecución, ver dos pestañas de resultado:
 - Raw Shell output (mensaje del Intérprete)
 - En los ejemplos: veremos varios avisos de funciones “deprecated” y la que hay que usar
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

```
    acknowledged: true,      ---> Lo ha insertado bien y ---> Se ha inventado el _Id:  
    insertedIds: { '0': ObjectId("64401bc69cf095dac762e823") }  
  }
```

- Shell Output (documents) resultado de la Ejecución (a veces tiene otros nombres)

```
{  
  "acknowledged" : true,  
  "insertedIds" : {  
    "0" : ObjectId("64401bc69cf095dac762e823")  
  }  
}
```

FUNCIONES BASICAS DE MONGO

➔ USAMOS COMMANDOS terminados en ";" -> es obligatorio si están en un script

Consultar y Crear una BD. (-> se ejecutan en la Shell:)

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 Studio3T:

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 : algo que indica si ha 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"]}
  ]
} )
```

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-restaurants.json de la práctica
- ➔ -> antes, ver como importarla a continuación

MAS UTILIDADES PRÁCTICAS

-- Funciones 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" } }
    }
  }
]
```

_____ MODO COMANDO : **mongodump y mongorestore** (NO USAR) _____

- Se necesita instalar un shell como mongosh , para ello hay que instalar MongoTools y se ejecuta así en el LAB:
Todos los programas -> BBDD -> MongoDB -> Entorno MongoDB
Se abre otra ventana cmd de Windows con el prompt: c:\hlocal>

Desde dentro del CMD del Cliente (paso 2.b): c:\hlocal> *mongosh* (SOLO para casos Especiales)

Responde:

MongoDB shell version: N.N.N → es un intérprete de JavaScript para la BD

// en la ventana del servidor dice : accept 1 conexión

// Completa comandos pulsando TAB

// aparece prompt ">". Para ver que funciona, Puedes probar: show dbs;

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 (descrito antes)

- NO actualiza los documentos si coinciden sus _Id únicos

- SI reconstruye los índices

3.- Recupera toda la BD de un 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:\Studio3T\rombodatos\misrestaurantesx.json
```

----- **Empezando en tu PC o Portatil** -----

→ LINK: ----- INFO de DUDAS: <https://community.mongodb.com>

1.- INSTALACIÓN el Servidor de MONGODB

→ INSTALAR ESTA VERSION: MongoDB Community Server

(hay detalles de Instrucciones en:

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

<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://...msi>

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

---- Dos modos de instalar el servidor :

- a) - Como una aplicación para CMD y arrancar a mano: → → Recomiendo ESTA
- Lo arrancas y paras cuando lo necesites (siguiente paso se indica cómo)

- b) - Se puede instalar como un servicio de WIN:
- SOLO si te manejas bien con los servicios, para poder pararlo cuando no lo uses.

---- NO usaremos Compass ahora: decide dejar el check si te interesan tareas de Administración de la BD

2.- Arrancar MongoDB Community Edition DESDE el Interprete CMD

→ HAZ ESTE PASO LA PRIMERA VEZ QUE TRABAJES:

- Crear el Dir de trabajo donde almacena los datos:
cd Disco: (donde va a estar)
md mongoWork

→ he puesto un .bat en la Prac de MongoDB para el arranque, que hace básicamente esto:
(-> Lo puedes ejecutar con doble click encima del .bat directamente en el Navegador de ficheros)

- Abrir CMD.exe como ADMINISTRADOR
- ASIGNA PATH donde está el servidor:
set path=C:\Program Files\MongoDB\Server\TUVERSION\bin\;%path%
- Arranca el servidor mongod.exe --dbpath A:\mongoWork

→ HAZ ESTE PASO CADA VEZ QUE VAYAS A TRABAJAR:

- Arranca el Servidor con el directorio de trabajo que has creado antes: Ejecutar el .bat

→ SIEMPRE -->> ESPERAR HASTA que aparecen, entre los últimos mensajes, estos trozos de texto :

```
... "listener", "msg": "Listening on", "attr": {"address": "127.0.0.1"}  
... , "msg": "Waiting for connections", "attr": {"port": 27017, "ssl": "off"}}
```

3.- INSTALACIÓN y USO del Editor Studio3T (un cliente de MongoDB)

→ Tanto en el laboratorio como en casa usaremos el mismo entorno: Studio3T

- Crea un directorio / carpeta para datos de trabajo del Studio3T, ejemplo D:/mongowork

-- **Descarga e Instalar Studio3T** : <https://studio3t.com/download-studio3t-free/>

- Después de 30 días caduca la versión completa: Entonces te pide credenciales :
- En la misma versión, "Switch to the free edition"
- (quita algunas funciones Free-forever version, que no necesitamos)

----- **Abre el editor Studio3T** : hay un icono en el escritorio

- Antes Arranco servidor

- Arranco Studio3T, hago login con EMAIL, pass y un Nombre

-- Creo conexión: Botón Connect + parámetros por defecto:

(para comunicarte con tu servidor local de MongoDB, que debe estar funcionando)

2.-BIS INSTALACIÓN y USO de las “utilidades”: MongoDB Tools 100 (y Restaurar un BD)

→ (poco uso actualmente) Las necesitamos para hacer operaciones de gestión que no están en Studio3T

→ LINK: ----- MANUAL de MongoDB: <https://www.mongodb.com/docs/database-tools/>

PASOS

- DESDE: <https://www.mongodb.com/try/download/database-tools> ,
Dentro de <https://docs.mongodb.com/database-tools/installation/installation/>
ejecuta el enlace del MongoDB Download Center :

→ debes escoger tu plataforma donde vas a ejecutarlas, ejemplo:

- versión 100.XXXXX
- platform Window x86_64
- Package : msi

→ bajas el mongodb-database-tools-windows-x86_64-100.xxxxxxxx.ZIP

→ (en CMD de WIN)

Descomprimirlo en un USB para poder traerlo todos los días al laboratorio (son ejecutables),

Pongo en un directorio mío EJ: D:\Mongo-TOOLS\

verás: \mongodb-database-tools-windows-x86_64-100.xxxxxxxx\bin

→ Listo para RESTAURAR UNA COLECCION **aficiones.bson**

- abro el CMD

- cd /dir donde están el fichero .bson y metadatos.bson. EJ:

> cd D:\prac5-API-2-3-4--MongoDB-CV\dump-para-empezar-prac

- posicionas en el dico donde está, EJ :> D:

- Ejecutar el mongorestore:

D:\Mongo-TOOLS\mongodb-database-tools-windows-x86_64-100.5.2\bin\mongorestore --db pracmongo aficiones.bson

así crea una BD (si no existe) *pracmongo*. Y una colección *aficiones*:

RESPONDE varios mensajes, al terminar no debe rechazar ningún documento:

finished restoring pracmongo.aficiones (89 documents, 0 failures)

89 document(s) restored successfully. 0 document(s) failed to restore.

- Para poder ver la nueva BD en Studio3T : debes refrescar con botón Dcho sobre la conexión

→ ----- NOTA: si conoces bien el uso de PATH puedes hacer esto:

→ Para poder ejecutar desde cualquier directorio en tu PC personal

en el CMD pones: `set path=C:\Program Files\MongoDB\Tools\100\bin;%path%`

(suponiendo que tienes el descomprimido en el path: `C:\Program Files\MongoDB\Tools\100\bin`)
→ `mongorestore --db pracmongo aficiones.bson`

→ **IMPORTAR BD o una Colección (usando la utilidades).**

Si quieres importar una base de datos. Si no existe *pracmongo*, la 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\TUVERSION\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 Studio3T y Mongoddb antes de recuperar la BD, para ver esta nueva BD, tienes que hacer refresh con botón dcho encima de la conexión en Studio3T

4.- OTRAS OPERACIONES HABITUALES

*** Otro cliente, es una Shell en modo línea de comandos: **mongo**

- Ejecutar desde otra ventana del CMD : **mongo**

Antes: debes haber asignado el *path* correcto como arriba

→ la nueva es **mongosh** , que se descarga aparte del Studio3T

del <https://www.mongodb.com/docs/mongodb-shell/>

→ "mongo shell" es deprecate a partir de la V.5.0 de MongoDB
