

Sesión 4 – Manejo de una base de datos con información multimedia en MongoDB.

Esta sesión la dividiremos en tres actividades. Cada actividad, a su vez, está dividida en varios ejercicios.

Preámbulo.

Implementaremos una base de datos sencilla en MongoDB. Lo que vamos a construir en estas dos sesiones es un directorio de farmacias del País Vasco. En la primera incluiremos algunos datos de tipo tradicional, y en la segunda otros multimedia. Los datos multimedia serán fotografías vinculadas a cada farmacia. La información que tenemos es la siguiente:

1. Una tabla maestra, que hemos descargado de un dataset público del gobierno Vasco, que ya no está disponible en formato Json. La página era: <http://opendata.euskadi.eus/catalogo/-/farmacias-y-botiquines/>. Las obtuvimos en formato Json, ya que es el más amigable para MongoDB. También acepta otros formatos (CSV y TSV).

Algunos de los datos que nos proporciona esa fuente de datos gratuita son:

- "documentName": Propietario de la farmacia
 - "documentDescription": Dirección. Es la concatenación de las columnas "sanidadStreet" (calle), "sanidadPostalCode" (código postal), y "SanidadTown" (municipio), con un espacio en medio de cada dato para separar la información.
 - "sanidadCenterType": Tipo de centro. Siempre es "Farmacia",
 - "sanidadCode": Código de tipo de centro. Siempre es "farmac_far".
 - "sanidadName": Nombre de la farmacia.
 - "sanidadPostalCode": Código postal
 - "sanidadProvince": Provincia
 - "sanidadRegion": Región. Siempre está en vacío.
 - "sanidadStreet": Calle.
 - "sanidadStreetSnap": Nombre corto de la calle.
 - "sanidadTown": Municipio.
2. Un conjunto de fotos de algunas de las farmacias (cada farmacia puede tener ninguna, una o varias fotografías). Las fotos pueden ser del interior o del exterior de la farmacia.

Para el desarrollo de esta práctica asumiremos que la máquina está configurada como una del laboratorio con el sistema operativo Windows, y que tiene MongoDB versión 3.2 instalado, es decir:

C:\mongodb

IMPORTANTE: En esta práctica nos referiremos a la carpeta donde está MongoDB como "Program Files". En algunas versiones de Windows deberéis sustituir esta carpeta por "Archivos de programa".

Si queréis realizar esta práctica en otras máquinas, podéis descargar el servidor de bases de datos gratuitamente en la siguiente dirección web:




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

Nosotros vamos a trabajar en el sistema operativo Windows, sin embargo, existen versiones para distintos sistemas operativos: Linux, MAC OS X, y Solaris. La versión del laboratorio es la 3.2, aunque a fecha de febrero 2019 ya van por la 4.0.

Actividad 1: Configuración preliminar: Acondicionamiento del entorno de trabajo.

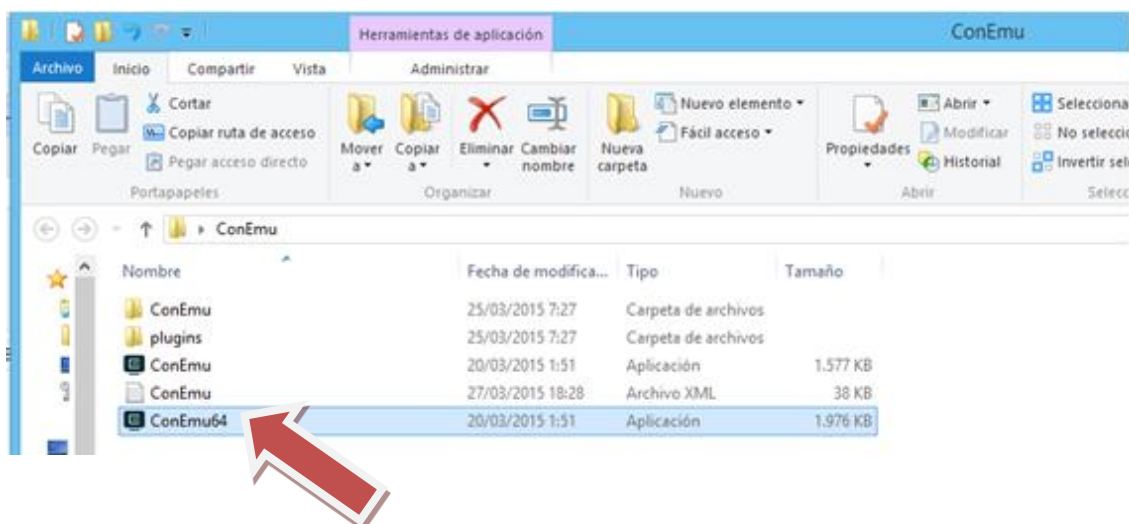
Abstract: Vamos a dejar el entorno del sistema preparado para poder realizar la práctica.

Paso 1: Descarga los materiales desde el campus virtual, y descomprímelos en una carpeta en el escritorio de Windows. Abre la carpeta de materiales, y verás estos tres ficheros:

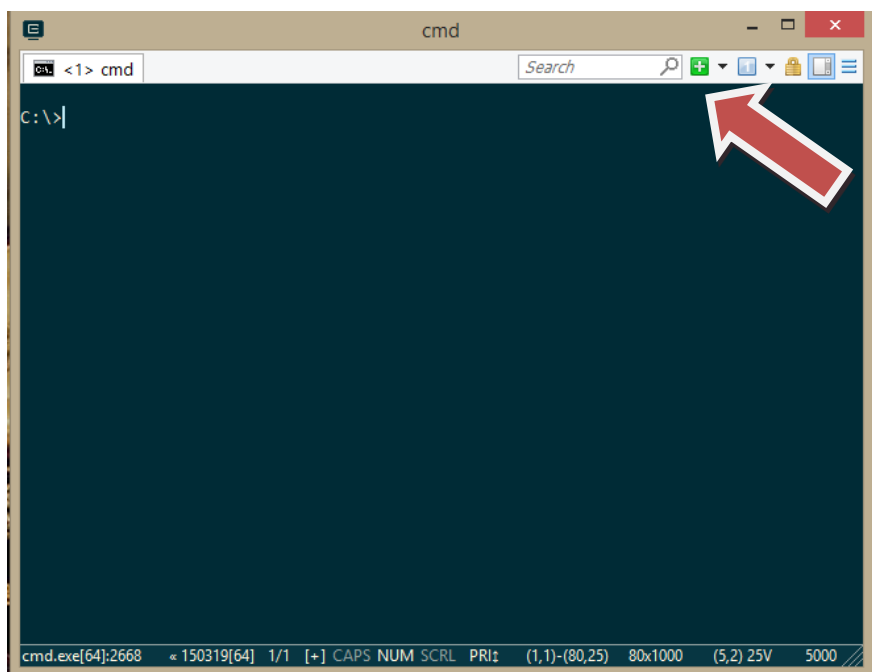
Nombre	Tipo	Tamaño comprimido	
 Exterior1	Archivo JPG	61 KB	I
 farmacias	Archivo JSON	30 KB	I
 Interior1	Archivo JPG	143 KB	I

Copia los archivos de trabajo en la carpeta c:\datosMongo. Comprueba que se han copiado correctamente.

Paso 2: Desarrollaremos la práctica utilizando un emulador gratuito de consola de MS-DOS llamado "ConEmu", que se encuentra también entre los materiales descargables desde el Campus Virtual. Nos permitirá poder interactuar con varias consolas de comandos simultáneamente. Para hacerlo funcionar, tan solo debes ubicar la carpeta en el escritorio de tu máquina. No necesita instalación. Hecho esto, ejecuta esta aplicación:



El resultado debe ser similar a éste:



Crea tres pestañas, que usaremos más adelante:

- La <1> para servir para tener el servidor de bases de datos activo.
- La <2> para ejecutar sentencias de MongoDB.
- La <3> para ejecutar comandos del sistema operativo desde la consola.

Las pestañas se crean pulsando el botón verde con la cruz blanca en el centro que se encuentra en la parte superior derecha de la aplicación.

IMPORTANTE: ConEmu permite pegar directamente del portapapeles los comandos a ejecutar. Por ello puedes copiarlos desde este documento, y pegar el texto ahí. En Windows 10 puedes hacer esto mismo con tres ventanas de comandos (cmd), no obstante te podría resultar algo más incómodo.

Actividad 2: Puesta en marcha del sistema.

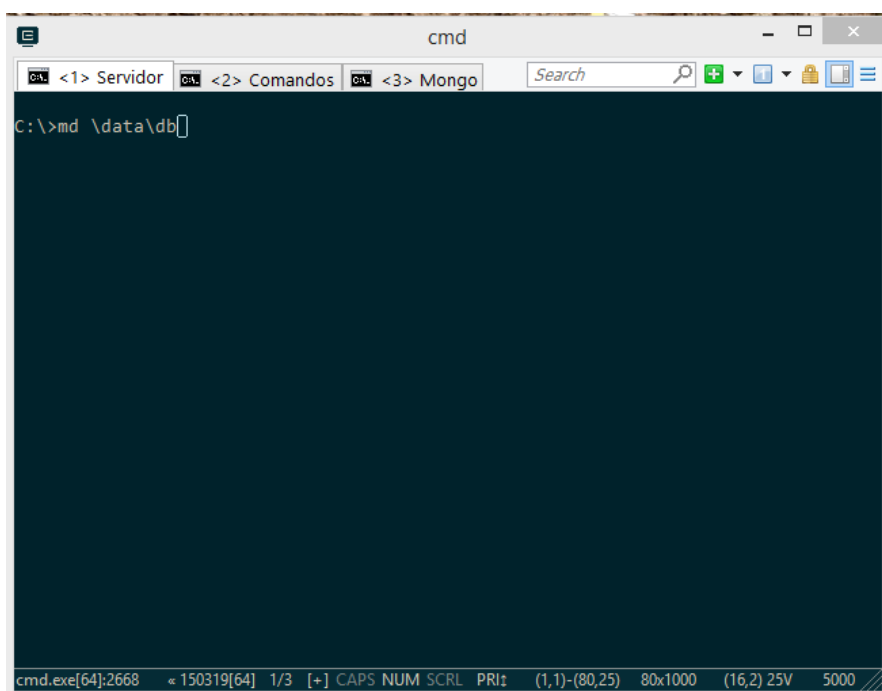
Ejercicio 1: Arranque del servidor de base de datos.

Abstract: Vamos a poner en marcha el servidor de base de datos MongoDB.

MongoDB busca por defecto los archivos de bases de datos en la carpeta `c:\data\db`. Comprueba si existe, y si no crea pues esa carpeta desde la consola <1>. Para ello ejecutaremos el comando:

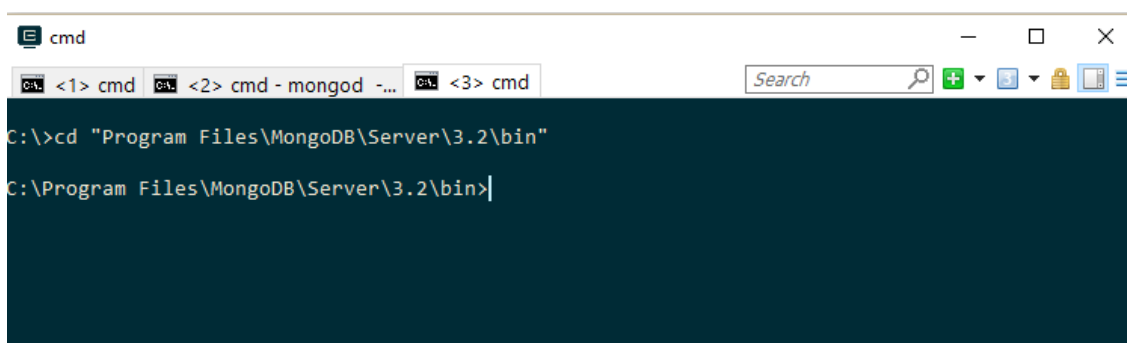
`md c:\data\db`

tal como se muestra en la siguiente imagen:



Hecho esto, movámonos a la carpeta donde se encuentra la aplicación del gestor de MongoDB mediante el comando:

`cd "c:\program files\mongodb\server\3.2\bin"`



Ejecutemos ahora el servidor de la base de datos mediante el siguiente comando:

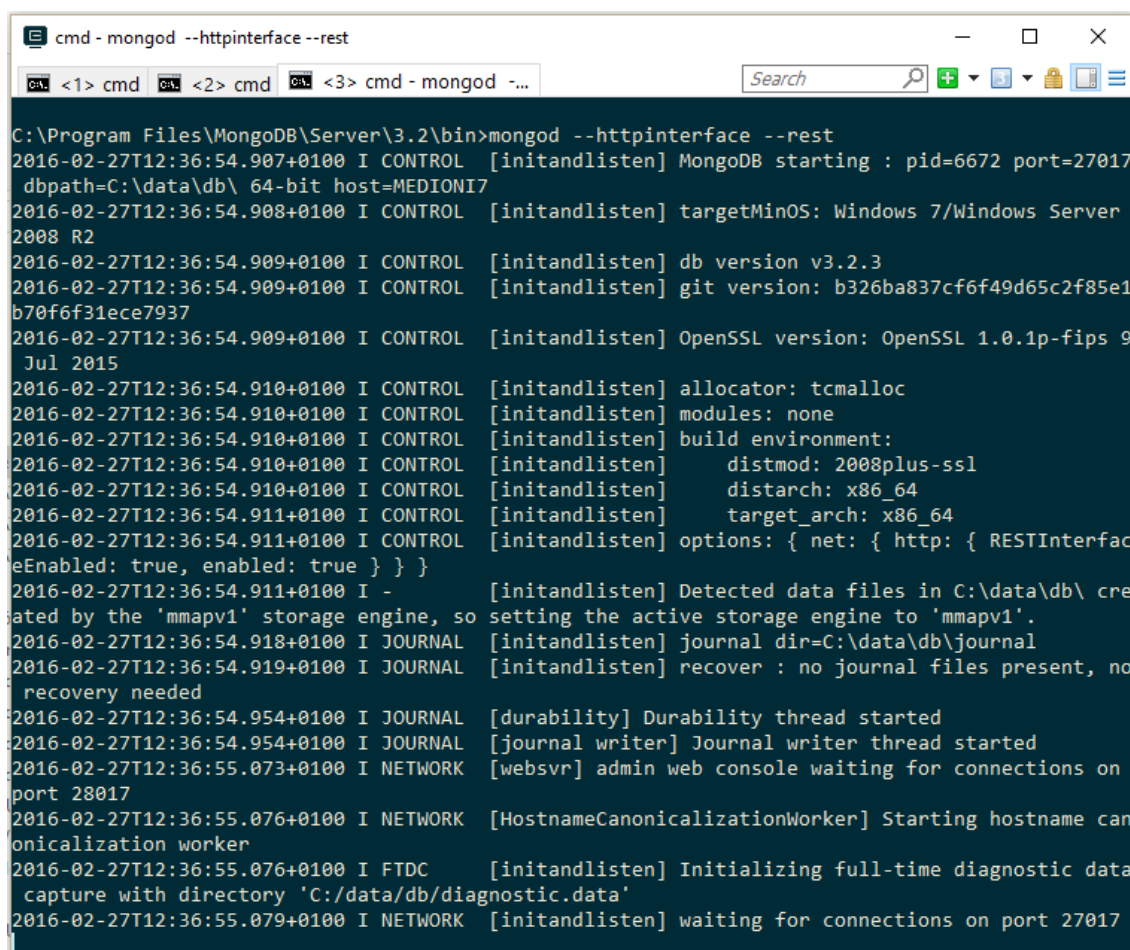
mongod --httpinterface --rest

Con este modo usaremos todos los parámetros del servidor por defecto, excepto las opciones `--httpinterface` y `--rest`. Ellas nos permitirán monitorizar su estado desde un navegador web por medio del protocolo http.



```
cmd
C:\Program Files\MongoDB\Server\3.2\bin>mongod --httpinterface --rest
```

El resultado debe ser algo similar al siguiente:



```
cmd - mongod --httpinterface --rest
C:\Program Files\MongoDB\Server\3.2\bin>mongod --httpinterface --rest
2016-02-27T12:36:54.907+0100 I CONTROL [initandlisten] MongoDB starting : pid=6672 port=27017
dbpath=C:\data\db\ 64-bit host=MEDIONI7
2016-02-27T12:36:54.908+0100 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server
2008 R2
2016-02-27T12:36:54.909+0100 I CONTROL [initandlisten] db version v3.2.3
2016-02-27T12:36:54.909+0100 I CONTROL [initandlisten] git version: b326ba837cf6f49d65c2f85e1
b70f6f31ece7937
2016-02-27T12:36:54.909+0100 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1p-fips 9
Jul 2015
2016-02-27T12:36:54.910+0100 I CONTROL [initandlisten] allocator: tcmalloc
2016-02-27T12:36:54.910+0100 I CONTROL [initandlisten] modules: none
2016-02-27T12:36:54.910+0100 I CONTROL [initandlisten] build environment:
2016-02-27T12:36:54.910+0100 I CONTROL [initandlisten] distmod: 2008plus-ssl
2016-02-27T12:36:54.910+0100 I CONTROL [initandlisten] distarch: x86_64
2016-02-27T12:36:54.911+0100 I CONTROL [initandlisten] target_arch: x86_64
2016-02-27T12:36:54.911+0100 I CONTROL [initandlisten] options: { net: { http: { RESTInterfac
eEnabled: true, enabled: true } } }
2016-02-27T12:36:54.911+0100 I - [initandlisten] Detected data files in C:\data\db\ cre
ated by the 'mmapv1' storage engine, so setting the active storage engine to 'mmapv1'.
2016-02-27T12:36:54.918+0100 I JOURNAL [initandlisten] journal dir=C:\data\db\journal
2016-02-27T12:36:54.919+0100 I JOURNAL [initandlisten] recover : no journal files present, no
recovery needed
2016-02-27T12:36:54.954+0100 I JOURNAL [durability] Durability thread started
2016-02-27T12:36:54.954+0100 I JOURNAL [journal writer] Journal writer thread started
2016-02-27T12:36:55.073+0100 I NETWORK [websvr] admin web console waiting for connections on
port 28017
2016-02-27T12:36:55.076+0100 I NETWORK [HostnameCanonicalizationWorker] Starting hostname can
onicalization worker
2016-02-27T12:36:55.076+0100 I FTDC [initandlisten] Initializing full-time diagnostic data
capture with directory 'C:/data/db/diagnostic.data'
2016-02-27T12:36:55.079+0100 I NETWORK [initandlisten] waiting for connections on port 27017
```

En este punto ya tendremos en marcha el servidor de base de datos con su configuración por defecto (dirección IP localhost, y puerto de escucha 27017), listo para recibir peticiones.

Ejercicio 2. Monitorizar el servidor mediante un navegador Web.

Abstract: Vamos a ver en un navegador web el estado del servidor MongoDB que acabamos de poner en marcha.

Ejecuta un navegador web (el que quieras), y accede a la url:

http://localhost:28017

MongoDB escucha el protocolo http por el puerto sumadas 1000 unidades al que escucha peticiones de base de datos. En nuestro caso, el 28017, ya que el de base de datos es el 27017.

[illegible]

En este punto podremos ver el estado del servidor de base de datos.

Investiga por las pestañas para ver lo que nos muestra sobre el estado del servidor. Observa que mucha información nos la muestra en formato Json, el que utiliza MongoDB para almacenar los datos.

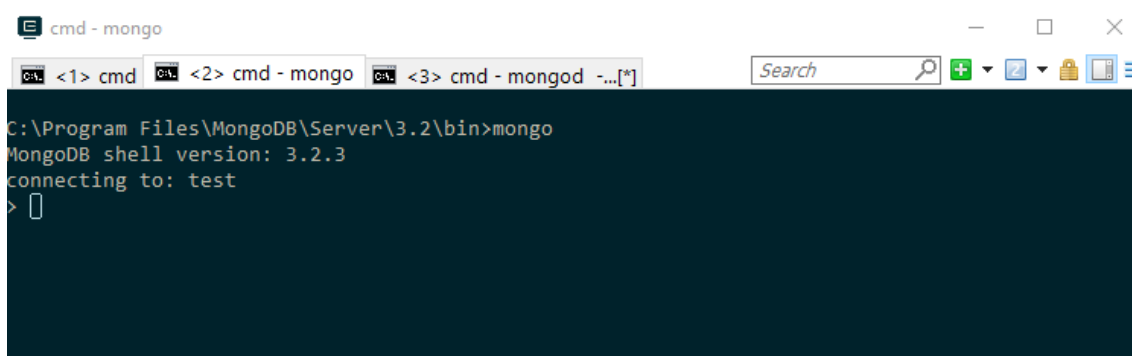
Ejercicio 3. Poner en marcha la aplicación cliente de MongoDB que permite ejecutar comandos de base de datos.

Abstract: Vamos a ejecutar la aplicación Mongo.exe, que nos permitirá interactuar con el servidor de bases de datos MongoDB.

Activa la pantalla de comandos <2> pulsando sobre ella. Movámonos a la carpeta donde se encuentra la aplicación de interacción con el servidor MongoDB mediante el comando:

cd "c:\program files\mongodb\server\3.2\bin"

Ejecuta a continuación el programa "mongo.exe". El resultado es el siguiente:



```
cmd - mongo
C:\Program Files\MongoDB\Server\3.2\bin>mongo
MongoDB shell version: 3.2.3
connecting to: test
>
```

En este punto tenemos ya el servidor de base de datos aceptando peticiones, y el cliente listo para realizárselas.

Actividad 3: Sentencias CRUD en MongoDB.

A partir de ahora, los ejercicios los realizarás tú. Consulta la documentación de la sesión para conocer los comandos que necesitarás para solucionar cada uno.

Nota importante: MongoDB es sensible al uso de mayúsculas y minúsculas. Cuando crees colecciones y bases de datos en él, asegúrate que usas los nombres tal como los has definido en su creación. Por ejemplo, la colección "Farmacias" será distinta de la colección "farmacias".

Ejercicio 4: Incorporar a tu servidor la información de la base de datos de ejemplo (CREATE).

Abstract: Vamos a crear a una base de datos que llamaremos "dbdm". Le añadiremos una colección de documentos con información sobre las farmacias del País Vasco. Esta información la tenemos en un archivo de texto, con formato Json.

Ejecuta en la pantalla <3> el comando para importar la colección a una base de datos en tu servidor MongoDB. Los parámetros del comando son:

BASEDATOS = dbdm

COLECCIÓN = Farmacias

FICHERO = "c:\datosMongo\Farmacias.json" (debes incluir las comillas dobles)

Lo que queremos hacer es cargar en la base de datos "dbdm" una colección llamada "Farmacias" en el servidor activo en nuestra máquina (el que hemos puesto en marcha en la actividad anterior).

Date cuenta de lo siguiente:

1. No hemos tenido que crear la base de datos. El hecho de importar una colección en ella la crea con la configuración por defecto.
2. No le hemos indicado la estructura de los documentos de la colección. La ha tomado directamente del archivo Json. Haciendo un símil con sistemas relacionales, no hemos tenido que ejecutar la instrucción "CREATE TABLE".

Ejercicio 5: Ejecutando consultas sencillas (READ).

Abstract: Vamos a realizar consultas a la colección Farmacias en la base de datos dbdm.

Lanzaremos un par de consultas sencillas a la base de datos. Aunque nuestro servidor ahora tan solo tiene una base de datos (dbdm), podría albergar muchas otras.

1. Ejecuta el comando correspondiente para seleccionar la base de datos dbdm.
2. Vamos a obtener todos los datos de los documentos de la colección Farmacias que estén en la ciudad de Bilbao. Ejecuta el comando correspondiente en la pantalla <2>. Los parámetros de la consulta son:

COLECCIÓN = Farmacias
FILTRO = sanidadTown: "Bilbao"

El resultado debe ser similar al que muestra la imagen siguiente:

```
cmd - mongo
> use dbdm
switched to db dbdm
> find('Farmacias', {'sanidadTown': 'Bilbao'})
{
  "sanidadCode" : "farmac_far",
  "sanidadName" : "Farmacia: MARTINEZ ECHAVE, Javier",
  "sanidadPostalCode" : "48013",
  "sanidadProvince" : "Bizkaia",
  "sanidadRegion" : "",
  "sanidadStreet" : "María Díaz de Haro, 9",
  "sanidadStreetSnap" : "",
  "sanidadTown" : "Bilbao"
}
{
  "_id" : ObjectId("5515d0e227adfcfd4d5ca4fe"),
  "documentName" : "GARCIA CERVERA, Fernando",
  "documentDescription" : "Zamacola, 57 48003 Bilbao",
  "sanidadCenterType" : "Farmacia",
  "sanidadCode" : "farmac_far",
  "sanidadName" : "Farmacia: GARCIA CERVERA, Fernando",
  "sanidadPostalCode" : "48003",
  "sanidadProvince" : "Bizkaia",
  "sanidadRegion" : "",
  "sanidadStreet" : "Zamacola, 57",
  "sanidadStreetSnap" : "",
  "sanidadTown" : "Bilbao"
}
Type "it" for more
>
```

La función "find" en MongoDB es equivalente a la sentencia SELECT en SQL. Estudiemos un poco el ejemplo:

1. La sentencia equivalente en SQL sería: `Select * from farmacias where sanidadTown = 'Bilbao'`
2. La función `pretty()` nos muestra el resultado formateado. Podemos ejecutar la misma sentencia quitando esa función, y nos mostrará todos los caracteres seguidos, siendo muy complicada la interpretación de la información.
3. Los filtros se incluyen con formato JSON dentro del primer parámetro de la función `find`.
4. Esta sintaxis es muy similar a la de Java o JavaScript.
5. Find devuelve los 20 primeros resultados. Si queremos ver más, ejecutaremos el comando "it";

Vamos a complicar un poco la sentencia de búsqueda, y vamos a seleccionar únicamente los nombres de los titulares (columna `documentName`), y la ciudad de las farmacias de Bilbao. Antes hemos seleccionado todas las columnas. Para ello usaremos la misma

función, pero indicándole qué columnas queremos mostrar. Los parámetros de la consulta son:

COLECCIÓN = Farmacias
FILTRO = sanidadTown: "Bilbao"
MOSTRAR = sanidadTown:1, documentName:1

El resultado de la consulta debe ser similar al que se muestra en la siguiente imagen:

```
cmd - mongo
<1> Servidor[*] <2> Mongo <3> Comandos
"documentName" : "SOLAY ISPIZUA, Jon Mikel",
"sanidadTown" : "Bilbao"
}
{
  "_id" : ObjectId("5515d0e227adfcfd4d5ca4f5"),
  "documentName" : "RUIZ GOLVANO, Joseba Andoni",
  "sanidadTown" : "Bilbao"
}
{
  "_id" : ObjectId("5515d0e227adfcfd4d5ca4f6"),
  "documentName" : "BLANCO ARCE, Amaya",
  "sanidadTown" : "Bilbao"
}
{
  "_id" : ObjectId("5515d0e227adfcfd4d5ca4f8"),
  "documentName" : "MARTINEZ ECHAVE, Javier",
  "sanidadTown" : "Bilbao"
}
{
  "_id" : ObjectId("5515d0e227adfcfd4d5ca4fe"),
  "documentName" : "GARCIA CERVERA, Fernando",
  "sanidadTown" : "Bilbao"
}
Type "it" for more
>
mongo.exe[64]:3836  150319[64] 2/3  [+ ] CAPS NUM SCRL PRI: (1,381)-(87,405) 87x1000 (3,405) 25V 1004 100%
```

Ejercicio 6: Incorporación de información a la base de datos (INSERT).

Abstract: Vamos a incorporar nuevos documentos a la colección farmacias.

Selecciona la pestaña <2> para interactuar con la base de datos.

Inserta en la colección farmacias el siguiente documento:

DATOS = _id: 2, documentName: "Farmacia DBDM", sanidadTown: "Alicante"

Consulta la base de datos para asegurarte que se ha insertado ese documento correctamente.

Nota importante: el campo "_id" es una columna especial en MongoDB que hace las funciones de clave primaria automática. Todas las colecciones la tienen de manera obligatoria, y sus valores no se pueden repetir en la misma colección. Si no le damos valor, el gestor de la base de datos le asigna un valor único automático.

Ejercicio 7: Modificación de información de la base de datos (UPDATE).

Abstract: Vamos a modificar información ya existente en la base de datos de farmacias.

Selecciona la pestaña <2> para interactuar con la base de datos de farmacias.

Modifica el documento que hemos insertado antes en colección farmacias con estos nuevos datos:

CONDICIONES = _id:2

DATOS = _id: 2, documentName: "Farmacia DBDM (Modificado)"

Nota importante: MongoDB sustituirá el primer documento que encuentre que cumpla las CONDICIONES indicadas por lo que pongamos en DATOS, no únicamente esas columnas. Las columnas que no indiquemos se perderán. En resumen: se sustituye un documento completo por el otro. Para modificar parcialmente el documento, o modificar muchos documentos que cumplan esa condición, tendríamos que hacerlo usando el operador \$set, que no veremos en estas sesiones por ser algo más avanzado.

Consulta la base de datos para asegurarte que se ha cambiado ese documento. Observa si se sigue manteniendo el valor de la columna sanidadTown que hemos insertado en el ejercicio anterior.

Ejercicio 8: Borrado de información de la base de datos (DELETE).

Abstract: Vamos a borrar un documento de la colección Farmacias.

Selecciona la pestaña <2> para interactuar con la base de datos de farmacias.

Borra el documento que hemos insertado y modificado antes en la colección farmacias:

CONDICIONES = _id:2

Consulta la base de datos para asegurarte que ya no existe ese documento.

Fin de la sesión.

Una vez terminados los ejercicios, deberemos cerrar adecuadamente el servidor. Para ello, ve a la pestaña <2> y ejecuta este comando:

db.getSiblingDB("admin").shutdownServer()

Los servidores de bases de datos deben ser desconectados de manera controlada, para que puedan terminar todas las transacciones pendientes, y guardar las cachés.

Hecho esto, cierra las tres sesiones de ConEmu.