



UNIVERSIDAD
GERARDO BARRIOS

Líderes en Gestión del Conocimiento

Manual de
Instalación,
configuración
y uso de
MongoDB.



mongoDB

Tabla de contenido

Introducción a MongoDB.....	3
Historia	3
¿Qué es MongoDB?.....	3
Características.....	3
Ventajas y desventajas	4
Instalación de MongoDB.....	4
paso 1 descargar de archivo	5
Paso 2 ejecutar el archivo	5
Paso 3 continuar con el asistente de instalación	6
Interactuando con MongoDB.....	14
Sentencia InsertOne	15
Sentencia InsertMany	15
Diversas maneras de consultar datos:	16
Sentencia DeleteOne.....	17
Sentencia Sort	17
Sentencia Count.....	18
Sentencia DeleteMany.....	19
Sentencia Drop.....	19
Script utilizado en los ejemplos	20
Modificación de Colecciones.....	21
Modificaciones y validaciones de colecciones en MongoDB desde la CMD.....	21
Validando Documentos	31
Script utilizado en los ejemplos	35
Referencias	36

Introducción a MongoDB

Historia

MongoDB fue fundada en 2007 por Dwight Merriman, Eliot Horowitz y Kevin Ryan, el equipo detrás de DoubleClick.

En la empresa de publicidad en Internet DoubleClick (ahora propiedad de Google), el equipo desarrolló y utilizó numerosos almacenes de datos personalizados para solucionar las carencias de las bases de datos existentes. El negocio servía 400 000 anuncios por segundo, pero a menudo tenía problemas de escalabilidad y agilidad. Frustrado, el equipo halló inspiración para crear una base de datos que abordará los desafíos a los que se enfrentaban en DoubleClick. Así fue cómo nació MongoDB. (MongoDB, 2021)



¿Qué es MongoDB?

Es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++, que en lugar de guardar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico. Al ser un proyecto de código abierto, sus binarios están disponibles para los sistemas operativos Windows, GNU/Linux, OS X y Solaris y es usado en múltiples proyectos o implementaciones en empresas como MTV Network, Craigslist, BCI o Foursquare. (OpenWebinars, 2021)

Características

- **Consultas ad hoc.** Con MongoDB podemos realizar todo tipo de consultas. Podemos hacer búsqueda por campos, consultas de rangos y expresiones regulares. Además, estas consultas pueden devolver un campo específico del documento, pero también puede ser una función JavaScript definida por el usuario.
- **Indexación.** El concepto de índices en MongoDB es similar al empleado en bases de datos relacionales, con la diferencia de que cualquier campo documentado puede ser indexado y añadir múltiples índices secundarios.
- **Replicación.** Del mismo modo, la replicación es un proceso básico en la gestión de bases de datos. MongoDB soporta el tipo de replicación primario-secundario. De este modo, mientras podemos realizar consultas con el primario, el secundario actúa como réplica de datos en solo lectura a modo copia de seguridad con la particularidad de que los nodos secundarios tienen la habilidad

de poder elegir un nuevo primario en caso de que el primario actual deje de responder. (Narváez, 2020)

- **Balanceo de carga.** Resulta muy interesante cómo MongoDB puede escalar la carga de trabajo. MongoDB tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balanceo de carga o servicio de replicación de datos, de modo que podemos mantener el sistema funcionando en caso de un fallo del hardware.
- **Almacenamiento de archivos.** Aprovechando la capacidad de MongoDB para el balanceo de carga y la replicación de datos, Mongo puede ser utilizado también como un sistema de archivos. Esta funcionalidad, llamada GridFS e incluida en la distribución oficial, permite manipular archivos y contenido.
- **Ejecución de JavaScript del lado del servidor.** MongoDB tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas. (revistapublicando, 2017)

Ventajas y desventajas

VENTAJAS

- ❖ Validación de documentos
- ❖ Motores de almacenamiento integrado
- ❖ Menor tiempo de recuperación ante fallos

DESVENTAJAS

- ❖ No es una solución adecuada para aplicaciones con transacciones complejas
- ❖ No tiene un reemplazo para las soluciones de herencia
- ❖ Aún es una tecnología joven (Bermejo Corrales, 2015)

Instalación de MongoDB

Requerimientos para la instalación:

- Funciona con cualquier Sistema Operativo de Windows
- No debe ser un Sistema Operativo de 32 Bits
- Mínimo de 512 de memoria RAM

paso 1 descargar de archivo

Para iniciar debemos descargar MongoDB Para ello nos vamos al navegador de nuestra preferencia y entramos al enlace que se sugiere a continuación:

<https://www.mongodb.com/try/download/community>

Descargamos la versión actual:

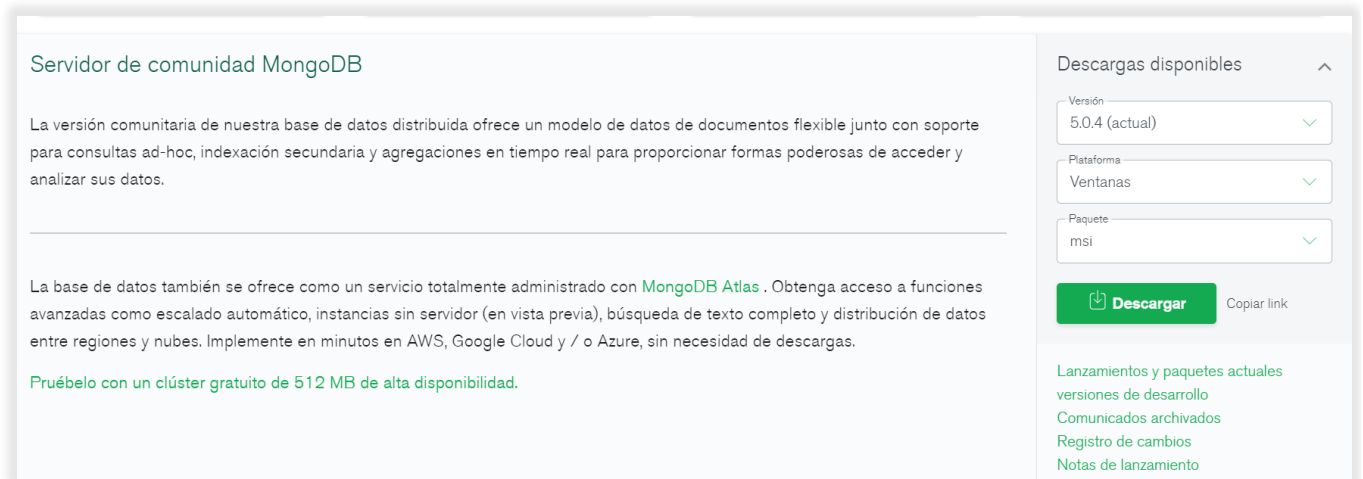


Imagen: Fuente propia

Paso 2 ejecutar el archivo

Se nos descargará algo como esto:

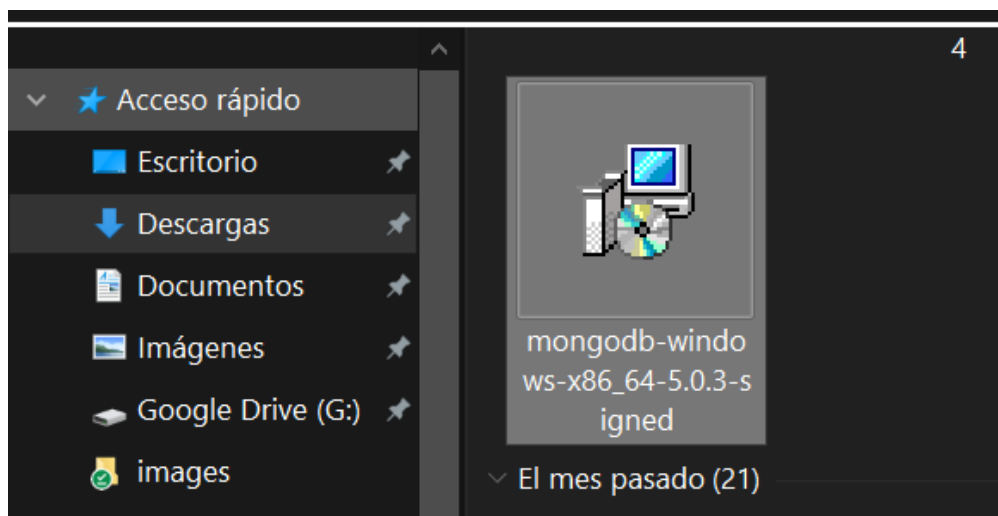


Imagen: Fuente propia

Damos doble clic sobre este instalador para que se ejecute el asistente instalación de MongoDB:

Paso 3 continuar con el asistente de instalación

El asistente nos da la bienvenida, damos clic en siguiente:



Imagen: Fuente propia (instalacion de MongoDB)

Aceptamos los términos y licencia, luego damos clic el siguiente:

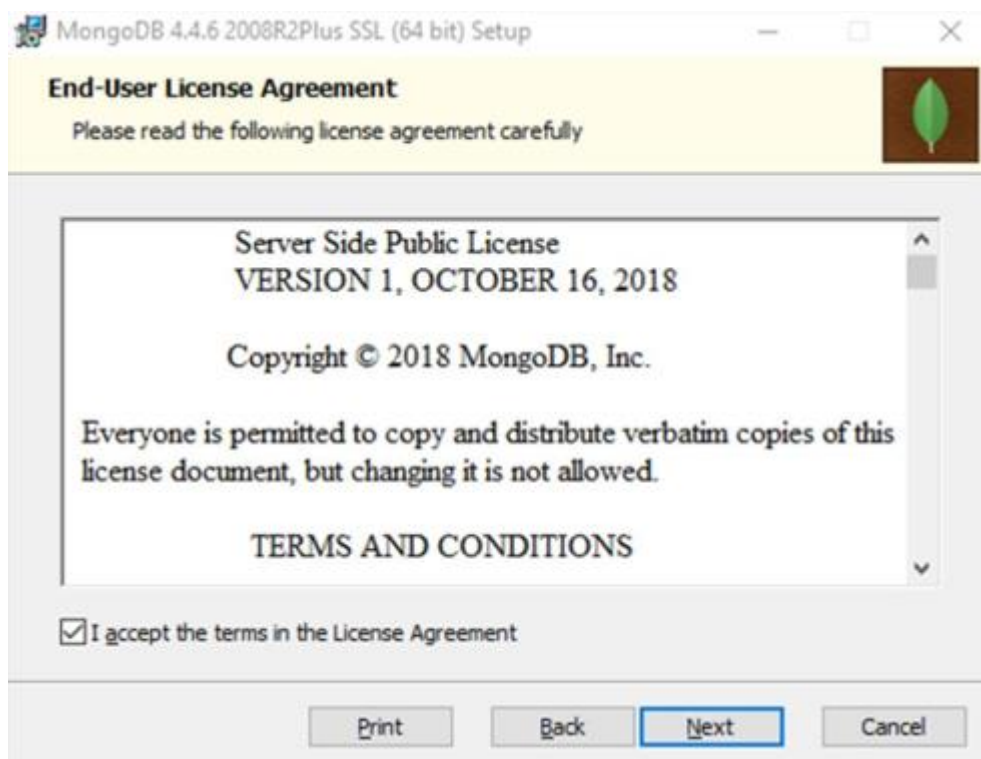


Imagen: Fuente propia (instalacion de MongoDB)

Es recomendable escoger la versión completa para que obtengamos todas características principales, damos clic en "complete".

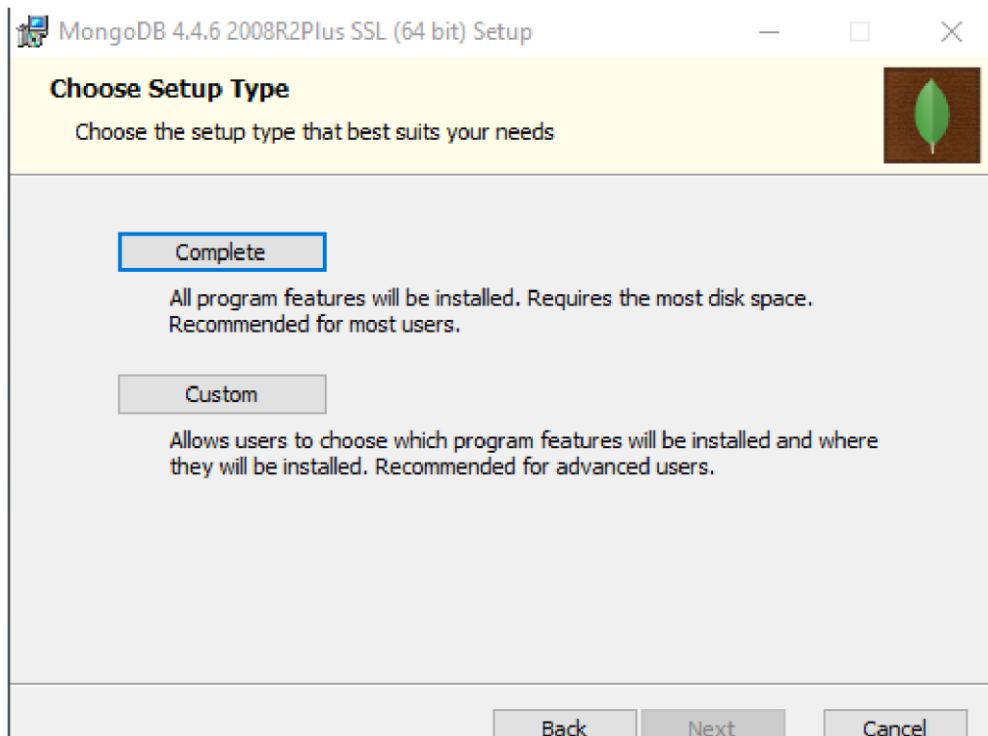


Imagen: Fuente propia (instalacion de MongoDB)

Aquí seleccionamos que mongoDB como servicio, lo que significa que siempre se va a estar ejecutando en segundo plano y no tendremos que iniciarlo de manualmente cuando requiramos del servidor.

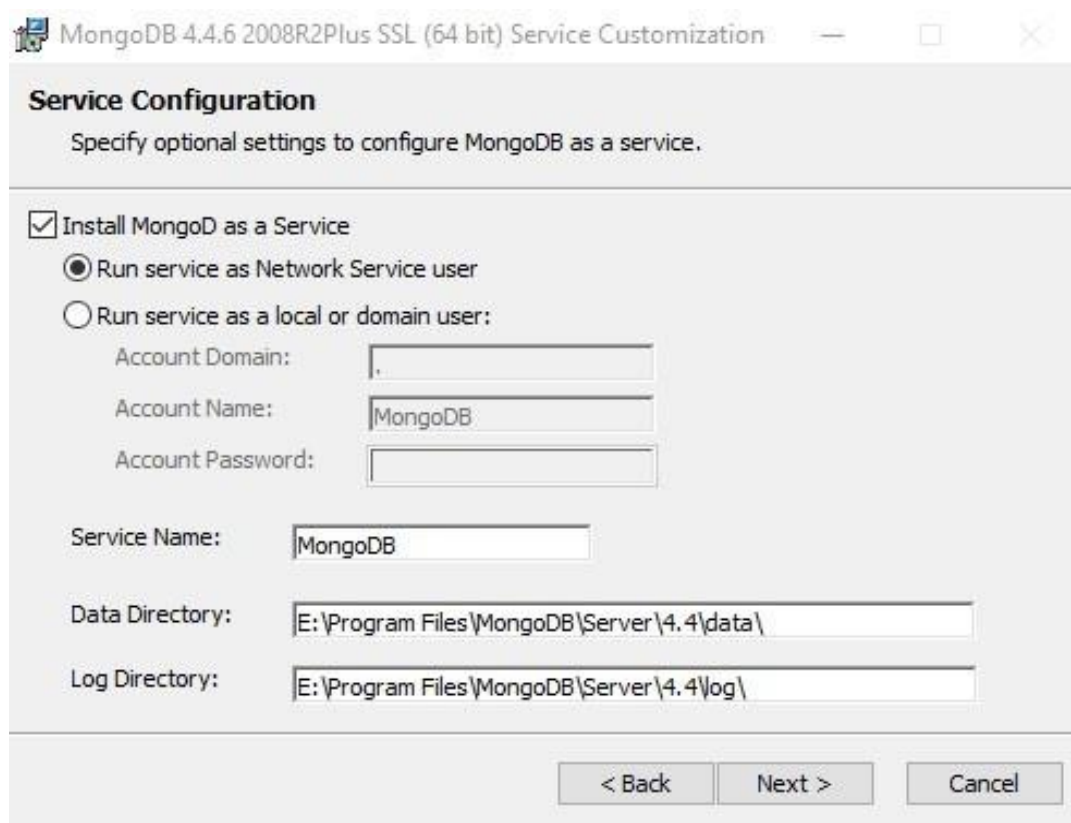


Imagen: Fuente propia (instalacion de MongoDB)

De manera predeterminada, mongo viene con su propia interfaz, para poder hacer uso del gestor, la cual es MongoDB Compass, marcamos la opción en la parte inferior de la ventana y damos clic en siguiente:



Imagen: Fuente propia (instalacion de MongoDB)

Ahora nos muestra tres opciones, regresar para hacer un preview, instalar o cancelar, damos clic en **Install**.

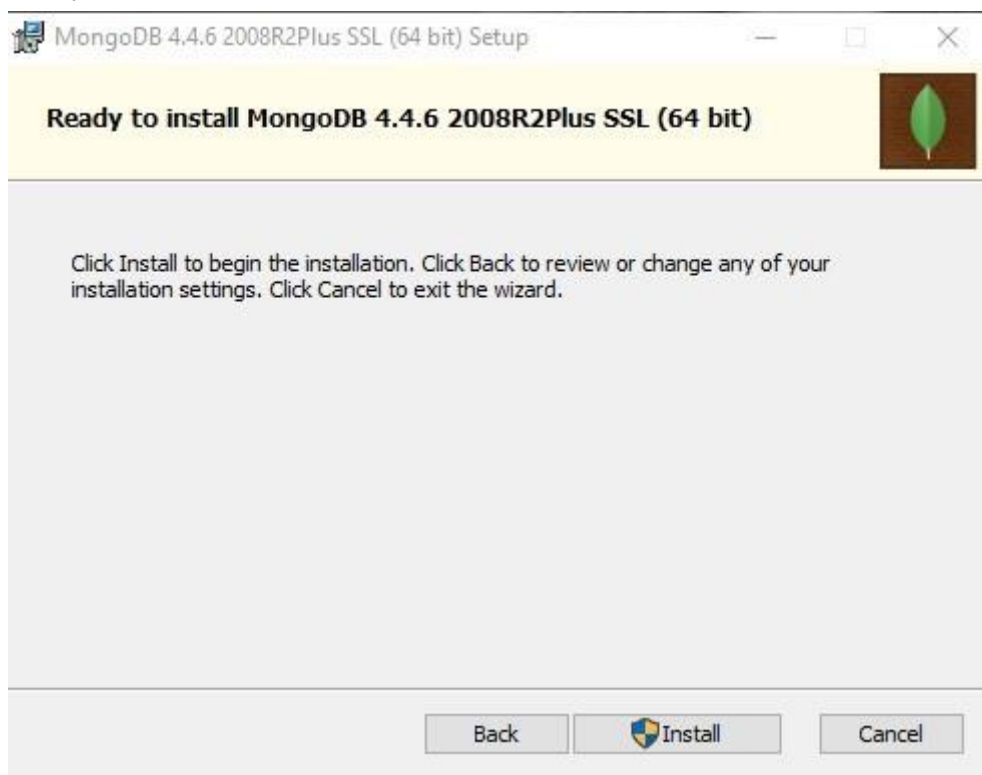


Imagen: Fuente propia (instalacion de MongoDB)

Esperando ...

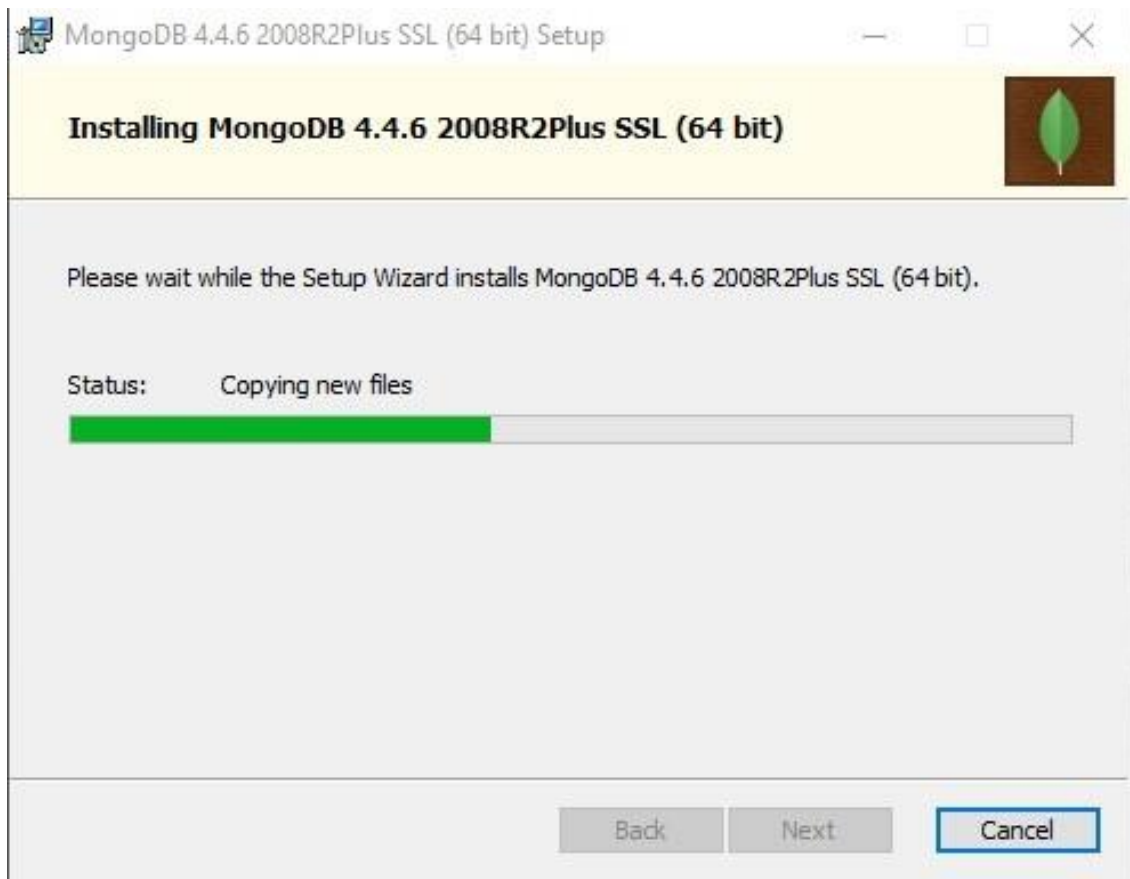


Imagen: Fuente propia (instalación de MongoDB)

Esperando por la instalación del MongoDB Compass:



Imagen: Fuente propia (instalacion de MongoDB)

Nos muestra la siguiente pantalla de finalización de la instalación, damos clic en Finish:

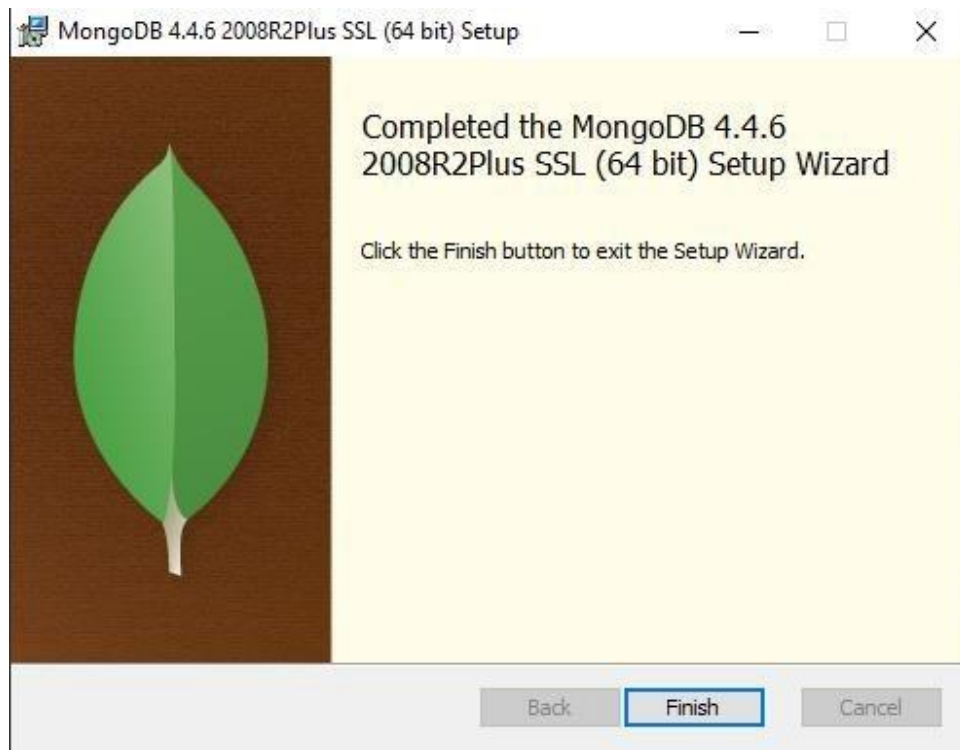


Imagen: Fuente propia (instalacion de MongoDB)

Obtenemos la interfaz de MongoDB Compass, solamente damos clic en **connect**, para poder ingresar al gestor.

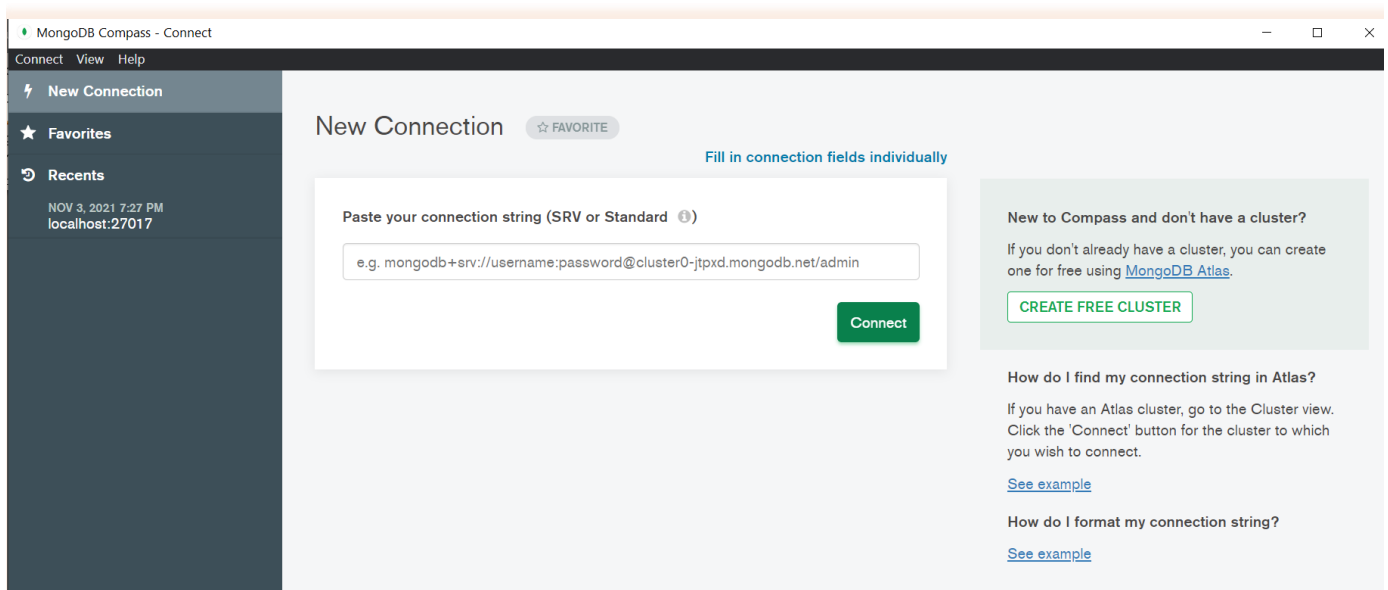


Imagen: Fuente propia (instalacion de MongoDB)

Para poder ejecutar mongo en la CMD de Windows es necesario ubicar la carpeta donde se instaló MongoDB, en la mayoría de los casos, si no se modificó ruta al momento de instalar, suele ser la siguiente:

C:\Program Files\MongoDB\Server\5.0\bin (copiamos esta ruta)

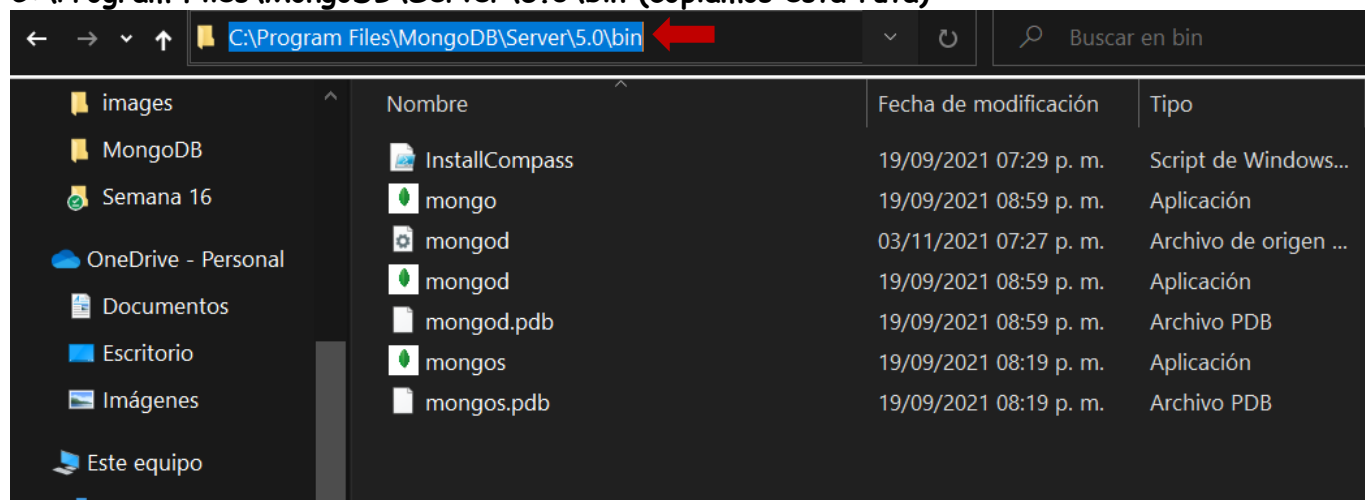


Imagen: Fuente propia (instalacion de MongoDB)

Digitamos en nuestro pc **"Editar las Variables de entorno del sistema"**, se nos despliega una ventana de propiedades del sistema.

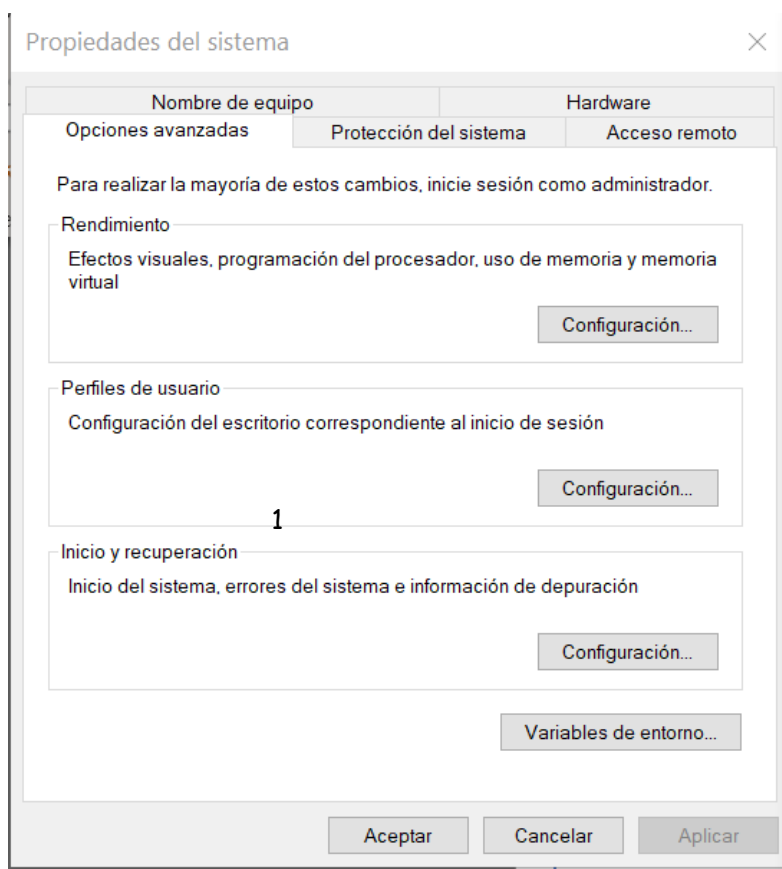


Imagen: Fuente propia (instalacion de MongoDB)

Damos clic a una de las últimas opciones que dice **"Variables de entorno"** y ubicamos las del sistema,

En ese listado debemos buscar una que se llame: **path**,

Editamos el **path**

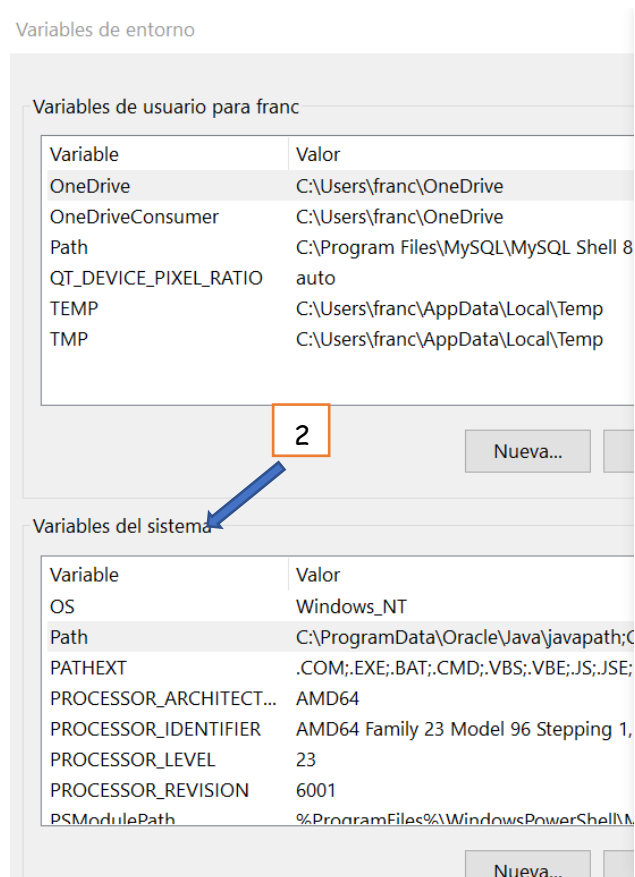


Imagen: Fuente propia (instalación de MongoDB)

Al final agregamos la Ruta que habíamos copiado con anterioridad:

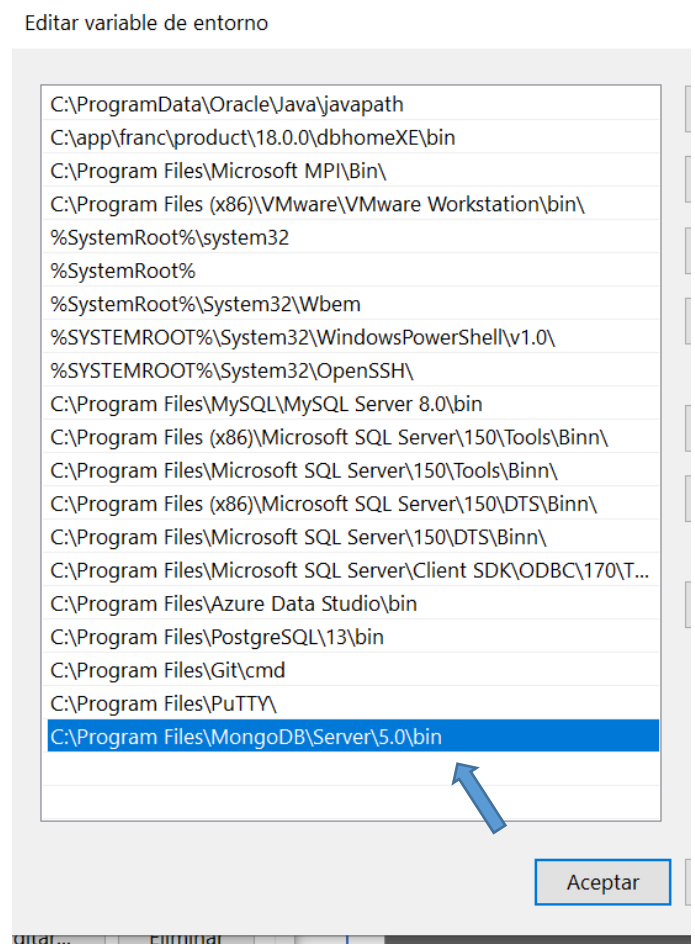
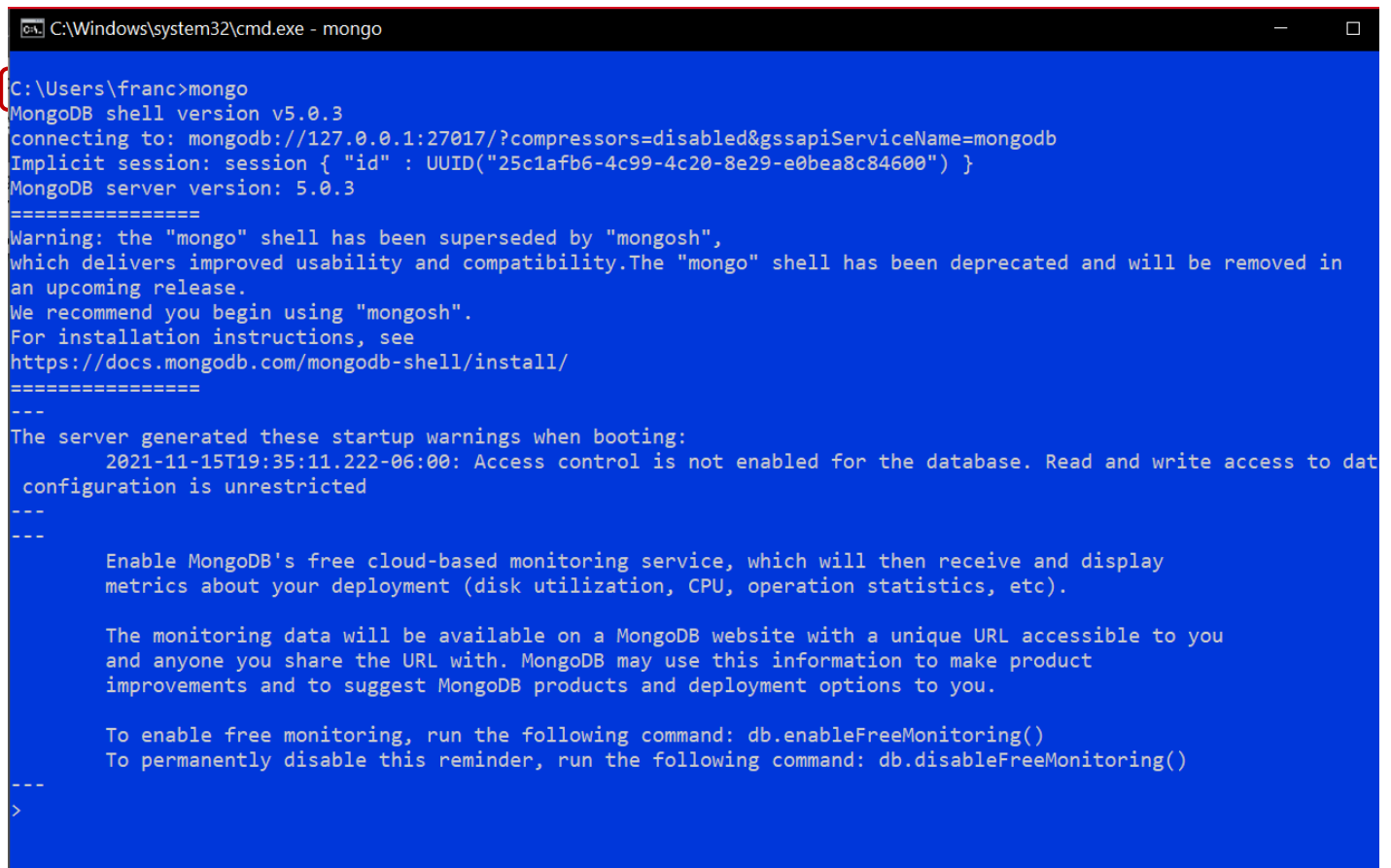


Imagen: Fuente propia (instalación de MongoDB)

Luego de haber realizado esos pasos, entramos al CMD de Windows y simplemente digitamos mongo, y como podemos observar ya estamos dentro del gestor.

A screenshot of a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe - mongo". The window has a blue background. The text inside shows the user typing "mongo" at the prompt "C:\Users\franc>". The output displays "MongoDB shell version v5.0.3", the connection details "connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb", the session ID "Implicit session: session { 'id' : UUID('25c1afb6-4c99-4c20-8e29-e0bea8c84600') }", and the server version "MongoDB server version: 5.0.3". A warning message follows, stating that the "mongo" shell is superseded by "mongosh" and will be deprecated. It provides a link to the MongoDB documentation for installation. Below this, it shows startup warnings generated by the server, including a message about unrestricted access to database configuration and a reminder to enable free cloud-based monitoring. The prompt returns to ">".

```
C:\Windows\system32\cmd.exe - mongo
C:\Users\franc>mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("25c1afb6-4c99-4c20-8e29-e0bea8c84600") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-11-15T19:35:11.222-06:00: Access control is not enabled for the database. Read and write access to dat
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

Imagen: Fuente propia (instalacion de MongoDB)

Hasta este punto hemos logrado realizar una instalación completa de MongoDB y la interfaz Compass, adicional a eso se logró configurar mongo para poder trabajar directamente desde la consola de Windows CMD.

De esta manera podemos entrar al Compass para trabajar con Mongo o lo hacemos también desde nuestra consola.

Interactuando con MongoDB

Show dbs: Nos muestra las bases de datos que tenemos creadas, por defecto solo tenemos tres, **admin**, **config** y **local**.

Para crear una base de datos necesitamos utilizar el comando **USE**, este comando nos permite crear la base de datos si no existe, y si ya existe nos permite cambiarnos a esa base de datos. Crearemos una base de datos llamada **baseJunior**.

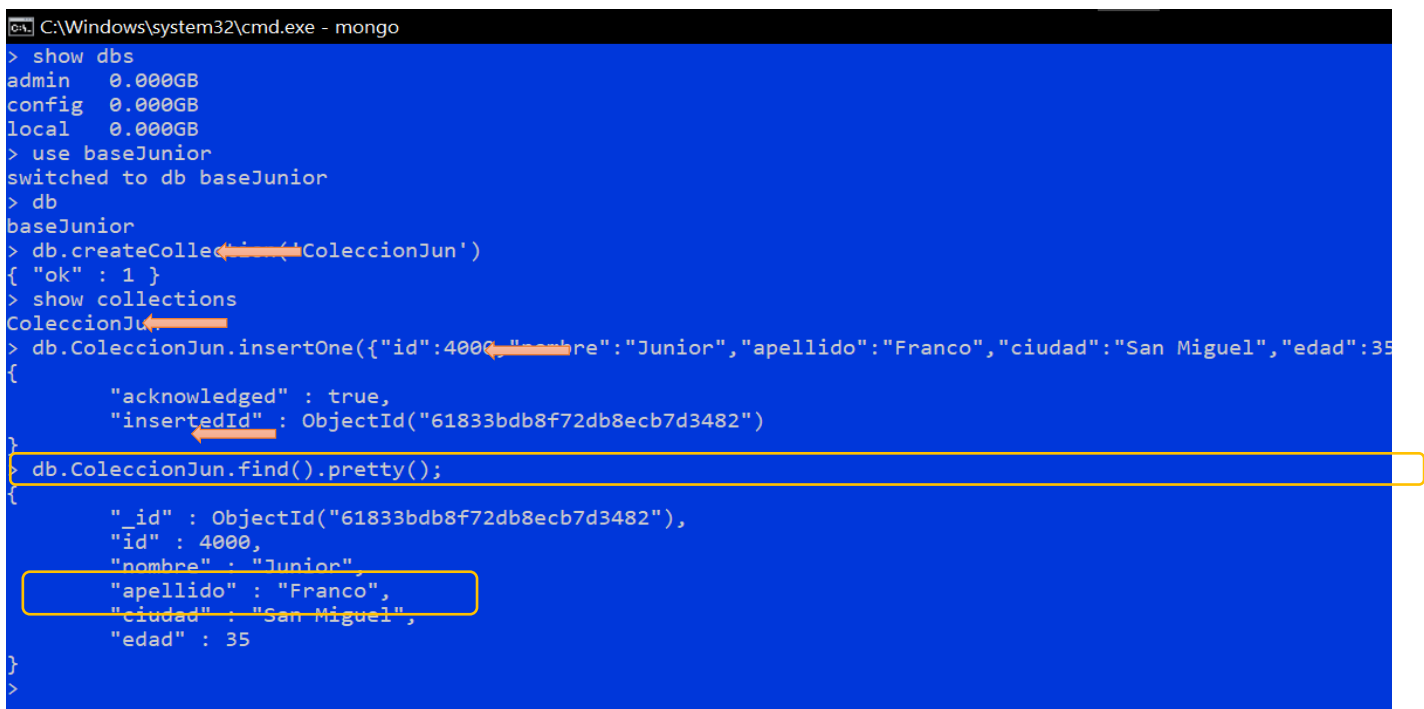
Para crear una colección, hacemos uso de la palabra reservada **"db.createCollection('ColeccionJun')"** Para mostrar las colecciones de la base de datos utilizamos el comando **"show collections"**.

A continuación le insertamos documentos a esta colección, para hacer esto podemos hacerlo de diversas maneras: (insertOne, insertMany e incluso con un **BULK**).

Insertamos un solo documento con la función **db.ColeccionJun.insertOne()** y especificamos dentro los valores a almacenar dentro de este documento en formato JSON

Luego de haber insertado un documento, consultamos con **db.ColeccionJun.find().pretty()** con esto, sin necesidad del **.pretty()** podemos obtener resultados, sin embargo la función **.pretty()** nos devuelve el resultado de una manera más legible y presentable.

Observamos que nos devuelve el contenido de la colección únicamente el documento que acabamos de insertar.



```

C:\Windows\system32\cmd.exe - mongo
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> use baseJunior
switched to db baseJunior
> db
baseJunior
> db.createCollection('ColeccionJun')
{ "ok" : 1 }
> show collections
ColeccionJun
> db.ColeccionJun.insertOne({ "id": 4000, "nombre": "Junior", "apellido": "Franco", "ciudad": "San Miguel", "edad": 35 })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61833bdb8f72db8ecb7d3482")
}
> db.ColeccionJun.find().pretty();
{
  "_id" : ObjectId("61833bdb8f72db8ecb7d3482"),
  "id" : 4000,
  "nombre" : "Junior",
  "apellido" : "Franco",
  "ciudad" : "San Miguel",
  "edad" : 35
}
>
  
```

Imagen: Fuente propia (Interactuando con MongoDB)

Sentencia InsertOne

Usamos la sentencia **insertOne** para insertar un segundo documento. Luego consultamos los documentos y ya tenemos dos.

```

C:\Windows\system32\cmd.exe - mongo
> db.ColeccionJun.insertOne({"id":4001,"nombre":"Maria","apellido":"Granados","ciudad":"San Jacinto","edad":22})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61833c5e8f72db8ecb7d3483")
}
> db.ColeccionJun.find().pretty();
{
  "_id" : ObjectId("61833bdb8f72db8ecb7d3482"),
  "id" : 4000,
  "nombre" : "Junior",
  "apellido" : "Franco",
  "ciudad" : "San Miguel",
  "edad" : 35
}
{
  "_id" : ObjectId("61833c5e8f72db8ecb7d3483"),
  "id" : 4001,
  "nombre" : "Maria",
  "apellido" : "Granados",
  "ciudad" : "San Jacinto",
  "edad" : 22
}

```

Imagen: Fuente propia (Sentencia InsertOne MongoDB)

Sentencia InsertMany

Para hacer uso de la sentencia **insertMany** creamos una nueva colección llamada **ColeccionJunProv** y para esto, los documentos deben ir entre corchetes así como en el **insertOne** pero separados por comas y dentro de corchetes que a su vez están dentro de los parentesis del **insertMany**, insertamos 6 documentos:

```

C:\Windows\system32\cmd.exe - mongo
> db.createCollection('ColeccionJunProv')
{ "ok" : 1 }
> show collections
ColeccionJun
ColeccionJunProv
> db.ColeccionJunProv.insertMany([
... {"id_pro":"prod1","nombre":"Sopa","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789},
... {"id_pro":"prod2","nombre":"Fideos","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700},
... {"id_pro":"prod3","nombre":"Chicles","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789},
... {"id_pro":"prod1","nombre":"Manzanas","proveedor":"El Rodeo","ciudad":"Ciudad Barrios","telefono":25569890},
... {"id_pro":"prod2","nombre":"Pepino","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895},
... {"id_pro":"prod3","nombre":"Ketchup","proveedor":"Super Josué","ciudad":"La Libertad","telefono":21345678}
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("618341288f72db8ecb7d3484"),
    ObjectId("618341288f72db8ecb7d3485"),
    ObjectId("618341288f72db8ecb7d3486"),
    ObjectId("618341288f72db8ecb7d3487"),
    ObjectId("618341288f72db8ecb7d3488"),
    ObjectId("618341288f72db8ecb7d3489")
  ]
}

```

Imagen: Fuente propia (Sentencia InsertMany MongoDB)

Consultamos los documentos insertados sin hacer uso del pretty() y observamos como nos devuelve los 6 resultados:

```
C:\Windows\system32\cmd.exe - mongo
> db.ColeccionJunProv.find()
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3486"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618341288f72db8ecb7d3488"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
>
```

Imagen: Fuente propia (Utilizando la sentencia Pretty MongoDB)

Diversas maneras de consultar datos:

1. Consultamos los documentos donde el **id_pro**, sea igual a **prod1**, obtenemos dos resultados.
2. Filtro para obtener los documentos que contengan como proveedor: "Franco". Obtenemos dos resultados.
3. Filtro de búsqueda para obtener todos los documentos cuya ciudad empiece por la letra "S", obtenemos 3 resultados.
4. Realizamos consulta para obtener la cantidad de documentos cuya ciudad empiece por la letra "S", tal como en la consulta anterior, obtenemos 3 pero esta vez un conteo.

```
C:\Windows\system32\cmd.exe - mongo
> db.ColeccionJunProv.find({"id_pro":"prod1"})
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
> db.ColeccionJunProv.find({"proveedor":"Franco"})
{ "_id" : ObjectId("618341288f72db8ecb7d3486"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3488"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
> db.ColeccionJunProv.find({"ciudad":/^S/})
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3486"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3488"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
> db.ColeccionJunProv.find({"ciudad":/^S/}).count()
3
>
```

Imagen: Fuente propia (Consultando datos MongoDB)

Sentencia DeleteOne

Para borrar documentos de una colección, lo podemos hacer igual que cuando insertamos un documento ó insertamos varios de un a vez.

A continuacion borramos los documentos que Tienen como proveedor a **"Franco"**

Luego de haberlos borrado uno por uno consultamos nuevamente los documentos y obtenemos 4 de los 6 que teníamos al principio.

```
> db.ColeccionJunProv.deleteOne({"proveedor":"Franco"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.ColeccionJunProv.deleteOne({"proveedor":"Franco"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.ColeccionJunProv.find({"proveedor":"Franco"})
[]
> db.ColeccionJunProv.find()
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
```

Imagen: Fuente propia (Sentencia DeleteOne MongoDB)

Sentencia Sort

Volvemos a insertar los documentos con proveedor **"Franco"** y las consultas que hacemos ahora son para ordenar de manera ascendente los documentos obtenidos, mediante el campo **"id_pro"** y es por eso que nos muestra los documentos con valores de prod1, prod1, prod2, prod2, prod3, prod3.

De igual manera consultamos los documentos y ordenamos alfabéticamente por el nombre del proveedor.

```
C:\Windows\system32\cmd.exe - mongo
> db.ColeccionJunProv.find()
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
{ "_id" : ObjectId("618344818f72db8ecb7d348a"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618344818f72db8ecb7d348b"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
> db.ColeccionJunProv.find().sort( { "id_pro": 1 } )
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618344818f72db8ecb7d348b"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
{ "_id" : ObjectId("618344818f72db8ecb7d348a"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
> db.ColeccionJunProv.find().sort( { "nombre": 1 } )
{ "_id" : ObjectId("618344818f72db8ecb7d348a"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618344818f72db8ecb7d348b"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
```

Imagen: Fuente propia (Sentencia Short MongoDB)

Sentencia Count

Ingresamos a la consola, mostramos las bases de datos, seleccionamos la **baseJunior**, mostramos sus colecciones y contamos los documentos de esa colección.

- ✓ ColeccionJun contiene 2 documentos.
- ✓ ColeccionJunProv, contiene 6 documentos.

```

C:\Windows\system32\cmd.exe - mongo
> show dbs
admin          0.000GB
baseJunior     0.000GB
config         0.000GB
local          0.000GB
> use baseJunior
switched to db baseJunior
> show collections
ColeccionJun
ColeccionJunProv
> db.ColeccionJun.count()
2
> db.ColeccionJunProv.count()
6

```

Imagen: Fuente propia (Sentencia Count MongoDB)

Verificamos desde MongoCompass las colecciones y documentos insertados en ColeccionJunProv.

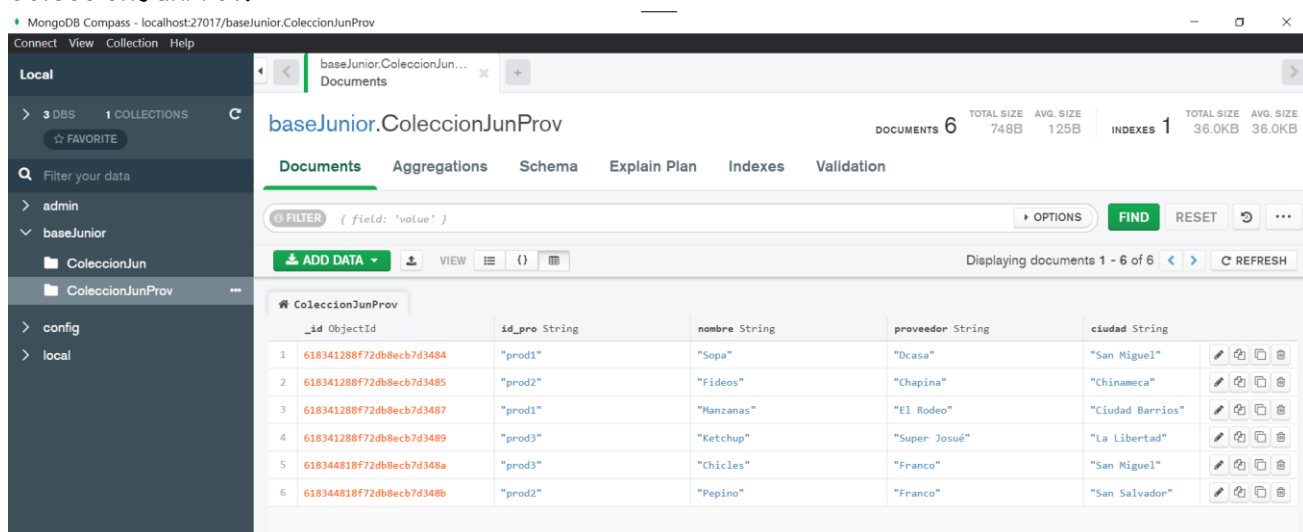


Imagen: Fuente propia (MongoCompass)

Sentencia DeleteMany

Procedemos a hacer uso de la segunda función para borrar documentos de una colección, primeramente mostramos los documentos completos con un find() luego procedemos a borrar todos los que coincidan con la condición que el **id_pro** sea igual a **prod1**.

Volvemos a consultar los documentos y esta vez hemos eliminado más de un solo documento al mismo tiempo.



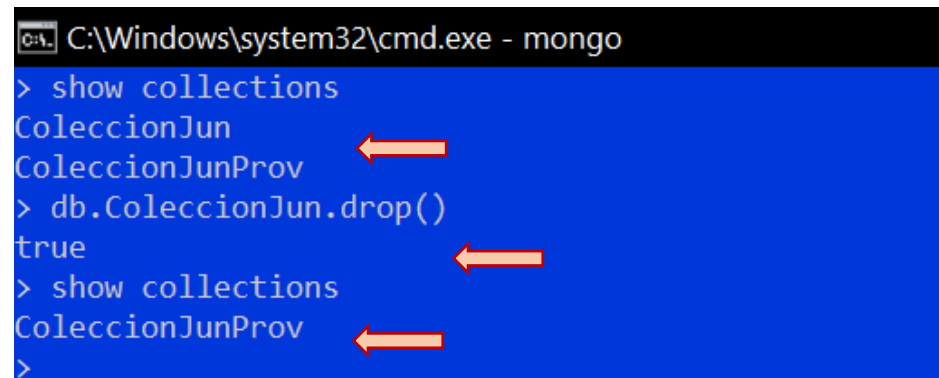
```

C:\Windows\system32\cmd.exe - mongo
> db.ColeccionJunProv.find()
{ "_id" : ObjectId("618341288f72db8ecb7d3484"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789 }
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3487"), "id_pro" : "prod1", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
{ "_id" : ObjectId("618344818f72db8ecb7d348a"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618344818f72db8ecb7d348b"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
> db.ColeccionJunProv.deleteMany({"id_pro":"prod1"})
{ "acknowledged" : true, "deletedCount" : 2 }
> db.ColeccionJunProv.find()
{ "_id" : ObjectId("618341288f72db8ecb7d3485"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700 }
{ "_id" : ObjectId("618341288f72db8ecb7d3489"), "id_pro" : "prod3", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678 }
{ "_id" : ObjectId("618344818f72db8ecb7d348a"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789 }
{ "_id" : ObjectId("618344818f72db8ecb7d348b"), "id_pro" : "prod2", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895 }
>
  
```

Imagen: Fuente propia (Sentencia DeleteMany MongoDB)

Sentencia Drop

Para eliminar por completo una colección, debemos escribir la sentencia **drop()**, luego de haber escrito el nombre de la colección, como observamos a continuación, nos devuelve el valor booleano "true" indicando que si se ha borrado exitosamente, por lo tanto consultamos nuevamente las colecciones y tenemos solo una.



```

C:\Windows\system32\cmd.exe - mongo
> show collections
ColeccionJun
ColeccionJunProv
> db.ColeccionJun.drop()
true
> show collections
ColeccionJunProv
>
  
```

Imagen: Fuente propia (Sentencia Drop MongoDB)

Script utilizado en los ejemplos**---InsertOne----**

```
db.ColeccionJun.insertOne({"id":4000,"nombre":"Junior","apellido":"Franco","ciudad":"San Miguel","edad":35})
db.ColeccionJun.insertOne({"id":4001,"nombre":"Maria","apellido":"Granados","ciudad":"San Jacinto","edad":22})
```

---InsertMany----

```
db.ColeccionJunProv.insertMany([
{"id_pro":"prod1","nombre":"Sopa","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789},
{"id_pro":"prod2","nombre":"Fideos","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700},
{"id_pro":"prod3","nombre":"Chicles","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789},
{"id_pro":"prod1","nombre":"Manzanas","proveedor":"ElRodeo","ciudad":"Ciudad Barrios","telefono":25569890},
{"id_pro":"prod2","nombre":"Pepino","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895},
{"id_pro":"prod3","nombre":"Ketchup","proveedor":"SuperJosué","ciudad":"La Libertad","telefono":21345678}])
```

----Consultar en orden---

```
db.ColeccionJunProv.find().sort( { "id_pro": 1 } )
db.ColeccionJunProv.find().sort( { "nombre": 1 } )
```

---Consultas diversas---

```
db.ColeccionJunProv.find({"id_pro":"prod1"})
db.ColeccionJunProv.find({"proveedor":"Franco"})
db.ColeccionJunProv.find({"ciudad":"/^S/})
db.ColeccionJunProv.find({"ciudad":"/^S/}).count()
```

--- Contar registros de una colección ---

```
db.ColeccionJun.count()
db.ColeccionJunProv.count()
```

--- Borrando datos de una colección ---

```
db.ColeccionJunProv.find()
db.ColeccionJunProv.deleteMany({"id_pro":"prod1"})
```

---InsertMany----

```
db.ColeccionJunProv.insertMany([
{"id_pro":"prod3","nombre":"Chicles","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789},
{"id_pro":"prod2","nombre":"Pepino","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895}])
```

****Borra todos los registros que tengan como id_pro el valor: prod1****

```
db.ColeccionJunProv.deleteMany({"id_pro":"prod1"})
db.ColeccionJunProv.find()
```

--- Borrando una colección ---

```
show collections
db.ColeccionJun.drop()
show collections
```


Modificaciones y validaciones de colecciones en MongoDB desde la CMD

Modificación de Colecciones

Para dar inicio con esta práctica empezaremos creando una colección que llamaremos "producto1", después de eso, hacemos insercciones dentro de esta colección, observemos que algunos datos tienen similitudes en sus datos, lo cual nos facilitará la tarea de realizar actualizaciones en dichos documentos.

```
C:\Windows\system32\cmd.exe - mongo
> db.createCollection('productos1')
{ "ok" : 1 }
> db.productos1.insertMany([
... {"id_pro":"prod1","nombre":"Sopa","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789,"cantidad":12},
... {"id_pro":"prod2","nombre":"Fideos","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700,"cantidad":12},
... {"id_pro":"prod3","nombre":"Chicles","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":12},
... {"id_pro":"prod4","nombre":"Manzanas","proveedor":"El Rodeo","ciudad":"Ciudad Barrios","telefono":25569890,"cantidad":12},
... {"id_pro":"prod5","nombre":"Pepino","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895,"cantidad":12},
... {"id_pro":"prod6","nombre":"Ketchup","proveedor":"Super Josué","ciudad":"La Libertad","telefono":21345678,"cantidad":12},
... {"id_pro":"prod7","nombre":"Ketchup","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789,"cantidad":25},
... {"id_pro":"prod8","nombre":"Pepino","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700,"cantidad":25},
... {"id_pro":"prod9","nombre":"Manzanas","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":25},
... {"id_pro":"prod10","nombre":"Peras","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":25},
... {"id_pro":"prod11","nombre":"Chicles","proveedor":"El Rodeo","ciudad":"Ciudad Barrios","telefono":25569890,"cantidad":25},
... {"id_pro":"prod12","nombre":"Fideos","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895,"cantidad":19},
... {"id_pro":"prod13","nombre":"Sopa","proveedor":"Super Josué","ciudad":"La Libertad","telefono":21345678,"cantidad":21}
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61918cd89b2a01d638d197c4"),
    ObjectId("61918cd89b2a01d638d197c5"),
    ObjectId("61918cd89b2a01d638d197c6"),
    ObjectId("61918cd89b2a01d638d197c7"),
    ObjectId("61918cd89b2a01d638d197c8"),
    ObjectId("61918cd89b2a01d638d197c9"),
    ObjectId("61918cd89b2a01d638d197ca"),
    ObjectId("61918cd89b2a01d638d197cb"),
    ObjectId("61918cd89b2a01d638d197cc"),
    ObjectId("61918cd89b2a01d638d197cd"),
    ObjectId("61918cd89b2a01d638d197ce"),
    ObjectId("61918cd89b2a01d638d197cf"),
    ObjectId("61918cd89b2a01d638d197d0")
  ]
}
```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Consultamos los datos insertados previamente en la colección "producto1", aplicamos un filtro de orden ascendente por el nombre del producto:



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Sopa", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 21 }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Para iniciar, vamos a utilizar la sentencia `update`, complementada con `upsert:true`, el cual nos permite realizar una insercción en caso de no encontrar un documento, pero realiza un `update` si encuentra algún documento con las condiciones especificadas.

Nosotros, buscamos el documento que tenga por `id_pro`, el **prod13**, y modifica (o inserta en caso de no encontrarlo), observamos que el resultado nos dice que se ha encontrado un documento, por lo cual no se ha insertado ninguno, y nos confirma que se ha hecho la modificación, ahora nuestro documento con el `prod13`, tiene modificaciones de todos los campos que se especificaron.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.update({"id_pro":"prod13"},{ $set: {"nombre":"Mirasol","proveedor":"DCASA","ciudad":"Sonsonate","Telefono":23456789,"cantidad":99},{ upsert: true}
)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "ca
ntidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 255698
90, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "can
tidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "
cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 2134567
8, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "can
tidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 255698
90, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "c
antidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "proveedor" : "DCASA", "ciudad" : "Sonsonate", "telefono" : 21345678, "can
tidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "c
antidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "can
tidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "can
tidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantid
ad" : 12 }
(i-search)`':

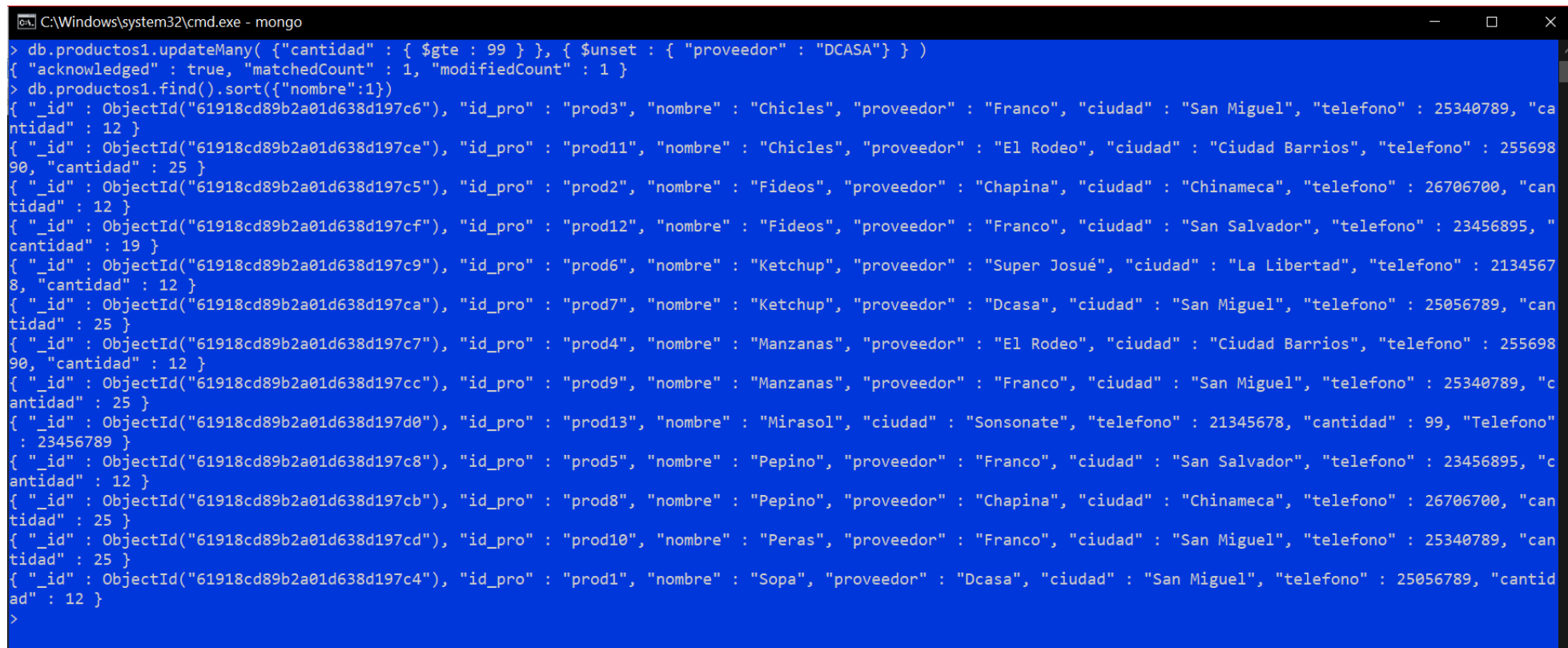
```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

A continuación tomando como referencia siempre el prod13, vamos a realizar una modificación a partir del campo: cantidad, por eso especificamos con un \$gte(mayor o igual que) en este caso la cantidad que es 99, y va eliminar un campo en todos los documentos que cumplan con esta condición, como solo tenemos un campo unicamente nos devuelve como resultado que ha encontrado una coincidencia, y que ha modificado un documento.

Cuando regresamos a consultar nuestra información podemos apreciar que se ha borrado el campo de proveedor:DCASA en el documento donde la cantidad es igual o mayor a 99.

De esta forma funciona el updateMany, buscando todos los documentos que coincidan con la condición para luego realizar las modificaciones que nosotros le especifiquemos.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.updateMany( {"cantidad" : { $gte : 99 } }, { $unset : { "proveedor" : "DCASA" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12 }
>

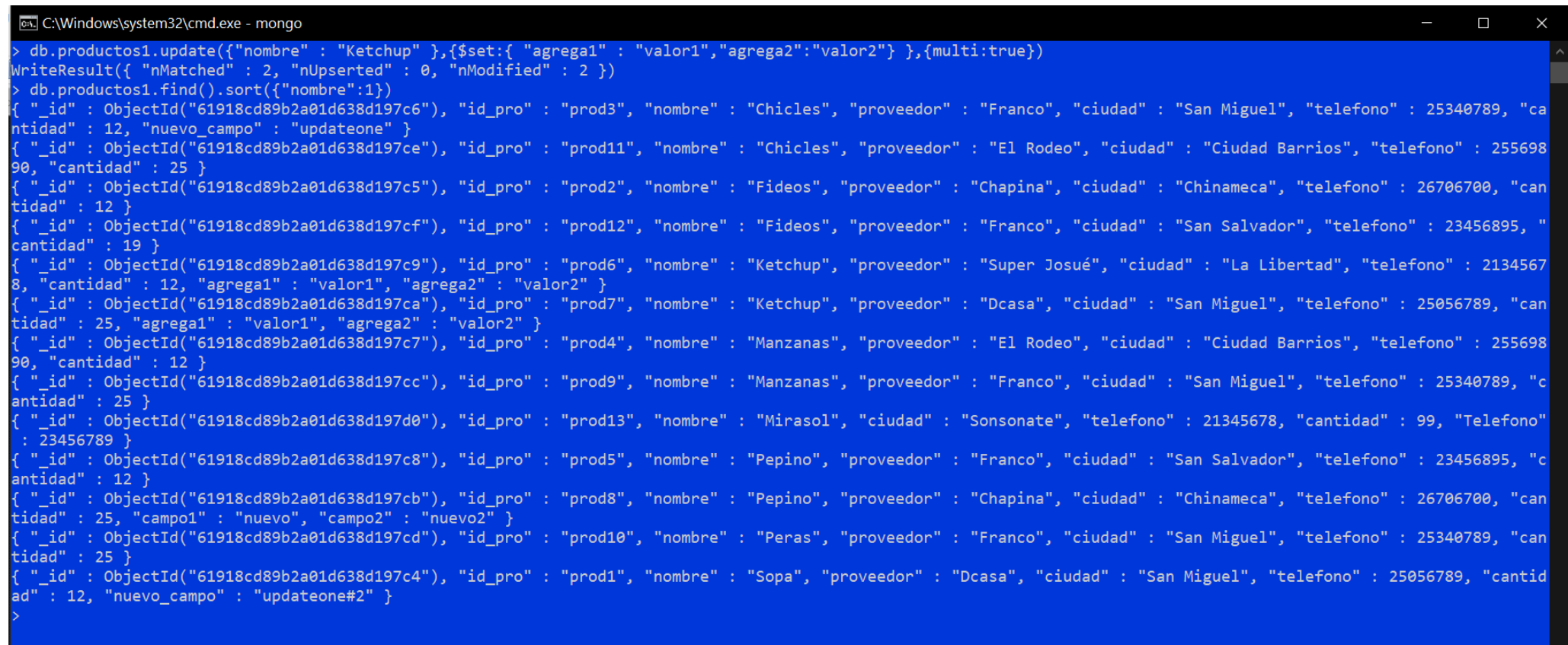
```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Seguidamente, haremos uso de la sentencia **update**, (similar a `updateMany`) puesto que esta funciona buscando los documentos que coinciden y modificando los documentos, con la palabra reservada **multi:true**.

Buscamos los documentos donde el nombre sea igual **Ketchup** y lo que vamos a hacer es agregar en todas las coincidencias dos campos nuevos. **agrega1** y **agrega2** respectivamente.

Al ejecutar esta sentencia, nos devuelve como resultado que ha encontrado dos coincidencias en la búsqueda y que ha realizado la modificación en ambos documentos, y como podemos verificar, ya tenemos en ambos, las modificaciones de los campos nuevos agregados.



```

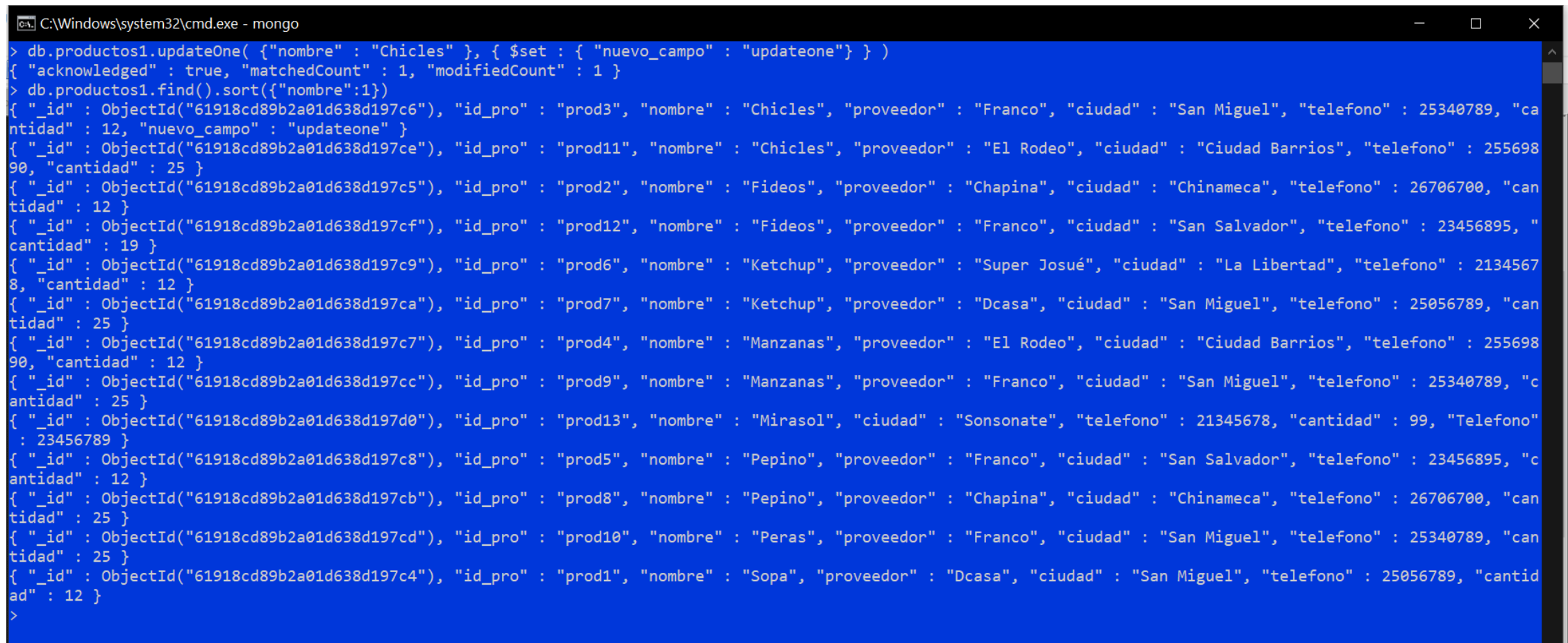
C:\Windows\system32\cmd.exe - mongo
> db.productos1.update({"nombre" : "Ketchup" },{$set:{ "agrega1" : "valor1","agrega2":"valor2"} },{multi:true})
WriteResult({ "nMatched" : 2, "nUpserted" : 0, "nModified" : 2 })
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12, "agrega1" : "valor1", "agrega2" : "valor2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25, "agrega1" : "valor1", "agrega2" : "valor2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25, "campo1" : "nuevo", "campo2" : "nuevo2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12, "nuevo_campo" : "updateone#2" }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Ahora haremos uso del **updateOne** para modificar un documento aunque existan varias coincidencias que cumplan con la condiciones para poder modificar.

Vamos a buscar todos los documentos que tengan como nombre el producto "chicle" (tenemos 2), y al encontrar la primer coincidencia, la modificará y terminará el proceso, es por eso que nos dice que ha encontrado una coincidencia y la ha modificado, al consultar los datos, observamos que el prod3 con nombre chicle, se le ha agregado el nuevo campo con el valor "updateone".



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.updateOne( {"nombre" : "Chicles" }, { $set : { "nuevo_campo" : "updateone"} } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12 }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Realizamos un segundo ejemplo con el update one, y esta vez, modificaremos el documento que tenga en cantidad un 12, como podemos observar tenemos varios documentos con cantidad 12, pero como el update one solo modifica uno, en el resultado obtenemos que ha encontrado uno y lo modificado, dicho documento modificado corresponde al prod1.

Con esto confirmamos que los demás documentos no se han modificado

```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.updateOne( {"cantidad" : { $eq : 12 } }, { $set : { "nuevo_campo" : "updateone#2" } } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12, "nuevo_campo" : "updateone#2" }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Haciendo uso del update en mongo, buscamos los documentos que tiene por nombre Ketchup, y a estos dos, con un multi:true, vamos a eliminar los campos que agregamos en el paso anterior, para esto ahcemos uso de la palabra reservada \$unset, y los datos que habíamos agregado anteriormente para Ketchup, los hemos eliminado.

Observamos que para eliminar estos campos es necesario mencionar los campos y el contenido.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.update({"nombre" : "Ketchup" }, { $unset : { "agrega1" : "valor1","agrega2":"valor2"} },{multi:true})
WriteResult({ "nMatched" : 2, "nUpserted" : 0, "nModified" : 2 })
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25, "campo1" : "nuevo", "campo2" : "nuevo2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12, "nuevo_campo" : "updateone#2" }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Nuevamente hacemos uso del **upsert** pero esta vez le diremos que modifique un valor que no existe, verificamos que no ha encontrado ningún documento, pero ha realizado una inserción y nos devuelve el id que le asignó al nuevo documento agregado al no encontrar coincidencias.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.update({"id":4002 },{ $set: {"nombre":"Caramelos","proveedor":"Colombina","ciudad":"Montecarlos","cantidad":29}},{ upsert: true})
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("61919666c63de5378475bb30")
})
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61919666c63de5378475bb30"), "id" : 4002, "cantidad" : 29, "ciudad" : "Montecarlos", "nombre" : "Caramelos", "proveedor" : "Colombina" }
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12 }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25, "campo1" : "nuevo", "campo2" : "nuevo2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12, "nuevo_campo" : "updateone#2" }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Regresamos con otro ejemplo de Update Many, ahora buscaremos todos los documentos que tengan en cantidad 12 y ahora a todos ellos les agregaremos un nuevo campo que llamaremos **ubicados**, y el valor llamado **"tu tienes cantidad 12"**

Verificamos la respuesta del gestor y obtenemos 6 coincidencias, y por ende 6 modificaciones.

Consultamos los datos y encontramos los 6 documentos modificados con la sentencia UpdateMany.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos1.updateMany( {"cantidad" : { $eq : 12 } }, { $set : { "Ubicados" : "Tu tienes cantidad 12"} } )
{ "acknowledged" : true, "matchedCount" : 6, "modifiedCount" : 6 }
> db.productos1.find().sort({"nombre":1})
{ "_id" : ObjectId("61919666c63de5378475bb30"), "id" : 4002, "cantidad" : 29, "ciudad" : "Montecarlos", "nombre" : "Caramelos", "proveedor" : "Colombina" }
{ "_id" : ObjectId("61918cd89b2a01d638d197c6"), "id_pro" : "prod3", "nombre" : "Chicles", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 12, "nuevo_campo" : "updateone", "Ubicados" : "Tu tienes cantidad 12" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ce"), "id_pro" : "prod11", "nombre" : "Chicles", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c5"), "id_pro" : "prod2", "nombre" : "Fideos", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 12, "Ubicados" : "Tu tienes cantidad 12" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cf"), "id_pro" : "prod12", "nombre" : "Fideos", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 19 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c9"), "id_pro" : "prod6", "nombre" : "Ketchup", "proveedor" : "Super Josué", "ciudad" : "La Libertad", "telefono" : 21345678, "cantidad" : 12, "Ubicados" : "Tu tienes cantidad 12" }
{ "_id" : ObjectId("61918cd89b2a01d638d197ca"), "id_pro" : "prod7", "nombre" : "Ketchup", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c7"), "id_pro" : "prod4", "nombre" : "Manzanas", "proveedor" : "El Rodeo", "ciudad" : "Ciudad Barrios", "telefono" : 25569890, "cantidad" : 12, "Ubicados" : "Tu tienes cantidad 12" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cc"), "id_pro" : "prod9", "nombre" : "Manzanas", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197d0"), "id_pro" : "prod13", "nombre" : "Mirasol", "ciudad" : "Sonsonate", "telefono" : 21345678, "cantidad" : 99, "Telefono" : 23456789 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c8"), "id_pro" : "prod5", "nombre" : "Pepino", "proveedor" : "Franco", "ciudad" : "San Salvador", "telefono" : 23456895, "cantidad" : 12, "Ubicados" : "Tu tienes cantidad 12" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cb"), "id_pro" : "prod8", "nombre" : "Pepino", "proveedor" : "Chapina", "ciudad" : "Chinameca", "telefono" : 26706700, "cantidad" : 25, "campo1" : "nuevo", "campo2" : "nuevo2" }
{ "_id" : ObjectId("61918cd89b2a01d638d197cd"), "id_pro" : "prod10", "nombre" : "Peras", "proveedor" : "Franco", "ciudad" : "San Miguel", "telefono" : 25340789, "cantidad" : 25 }
{ "_id" : ObjectId("61918cd89b2a01d638d197c4"), "id_pro" : "prod1", "nombre" : "Sopa", "proveedor" : "Dcasa", "ciudad" : "San Miguel", "telefono" : 25056789, "cantidad" : 12, "nuevo_campo" : "updateone#2", "Ubicados" : "Tu tienes cantidad 12" }
>

```

Imagen: Fuente propia (Modificación de Colecciones MongoDB)

Validando Documentos

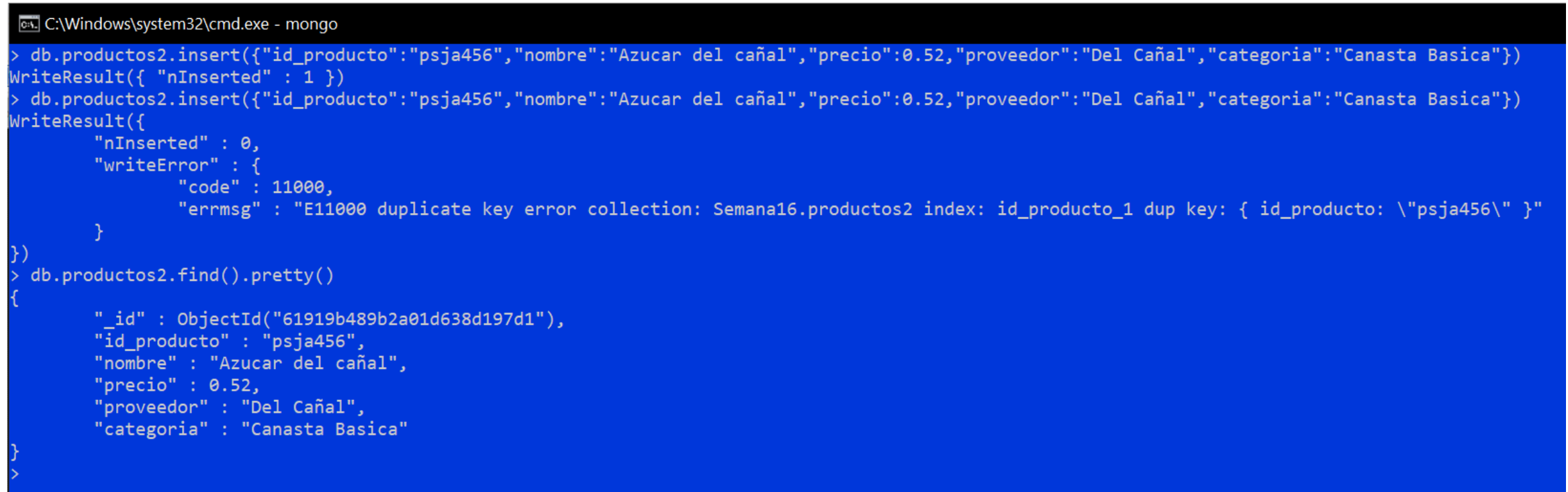
- ✓ Creamos una colección llamada productos2 y usamos en el validador los campos a almacenar.
- ✓ Id_producto (unicamente podra aceptar valores de tipo string por ejemplo un producto "PMJA345").
- ✓ Nombre: acepta valores de tipo string ""; precio: (acepta valores de tipo decimal); proveedor (calores string);
- ✓ Para la categoría hemos definido un listado en el cual, si se intenta agregar uno que no este en este listado, nos devolverá un error de validación.

Luego de eso creamos un indice que agiliza las consultas en nuestra colección y además de eso le decimos que es un campo **unique**, para que el código del producto no pueda repetirse en ninguna insercción que se realice. Cómo dato importante al final nos dice que tenemos dos indices creados, y por tanto estos dos indices corresponden a el que genera mongo por defecto y el que hemos creado nosotros a partir del campo **id_producto**.

```
C:\Windows\system32\cmd.exe - mongo
> show collections
productos1
> db.createCollection( "productos2",
... { validator: { $and:
... [
... { id_producto: { $type: "string" } },
... { nombre: { $type: "string" } },
... { precio: { $type: "double" } },
... { proveedor: { $type: "string" } },
... { categoria: { $in: [ "Lacteos", "Comestibles", "Golosinas", "Canasta Basica", "Enlatados","Bebidas" ] } }
... ]
... }
... } )
{ "ok" : 1 }
> db.productos2.createIndex( { "id_producto": 1 }, { unique: true } )
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
>
```

Imagen: Fuente propia (Validando Colecciones MongoDB)

Ahora probamos el campo unico que creamos, insertamos un documento con codigo "psja456" y obtenemos correcto, ahora intentamos volver a registrar el mismo producto y nos devuelve el error en el que nos aclara que tenemos la llave id_producto **duplicada**.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos2.insert({"id_producto":"psja456","nombre":"Azucar del cañal","precio":0.52,"proveedor":"Del Cañal","categoria":"Canasta Basica"})
WriteResult({ "nInserted" : 1 })
> db.productos2.insert({"id_producto":"psja456","nombre":"Azucar del cañal","precio":0.52,"proveedor":"Del Cañal","categoria":"Canasta Basica"})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: Semana16.productos2 index: id_producto_1 dup key: { id_producto: \"psja456\" }"
  }
})
> db.productos2.find().pretty()
{
  "_id" : ObjectId("61919b489b2a01d638d197d1"),
  "id_producto" : "psja456",
  "nombre" : "Azucar del cañal",
  "precio" : 0.52,
  "proveedor" : "Del Cañal",
  "categoria" : "Canasta Basica"
}
>

```

Imagen: Fuente propia (Validando Colecciones MongoDB)

Ahora el momento de probar las validaciones de la colección que creamos, intentamos insertar un documento que tiene en el campo precio un valor de tipo string con \$7.00 pero nos devuelve un error y el error radica en que como se nos muestra el campo precio es de tipo double...

Confirmamos que la validación para el campo precio, está correcta.



```

C:\Windows\system32\cmd.exe - mongo
> db.productos2.insert({"id_producto":"psja457","nombre":"Sopa Laky","precio":"$7.00","proveedor":"La Chapina","categoria":"Canasta Basica"})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 121,
    "errmsg" : "Document failed validation",
    "errInfo" : {
      "failingDocumentId" : ObjectId("61919d7e9b2a01d638d197d3"),
      "details" : {
        "operatorName" : "$and",
        "clausesNotSatisfied" : [
          {
            "index" : 2,
            "details" : {
              "operatorName" : "$type",
              "specifiedAs" : {
                "precio" : {
                  "$type" : "double"
                }
              },
              "reason" : "type did not match",
              "consideredValue" : "$7.00",
              "consideredType" : "string"
            }
          }
        ]
      }
    }
  }
})

```

Imagen: Fuente propia (Validando Colecciones MongoDB)

Procedemos a corregir el campo precio y nos devuelve correcta la insercción consultamos la coleccion y tenemos dos documentos, el primero que ocupamos para probar el campo **unique** y el segundo que acabamos de corregir del precio.

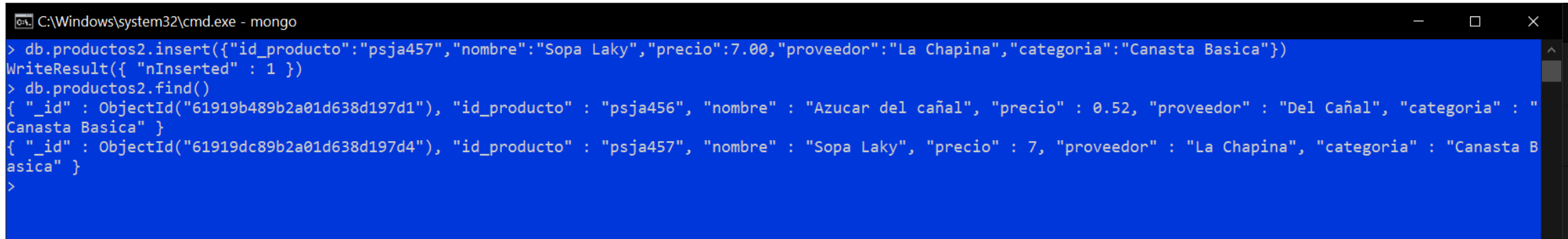


Imagen: Fuente propia (Validando Colecciones MongoDB)

Ahora vamos a probar la validación en el campo de las categorías, observemos que se intenta agregar un documento con la categoria que no existe (Navideños) y precisamente eso es lo que nos dice mongo, que la categoria navideña no existe dentro las validaciones que le han prdenado hacer, por tanto no permite crear el documento.

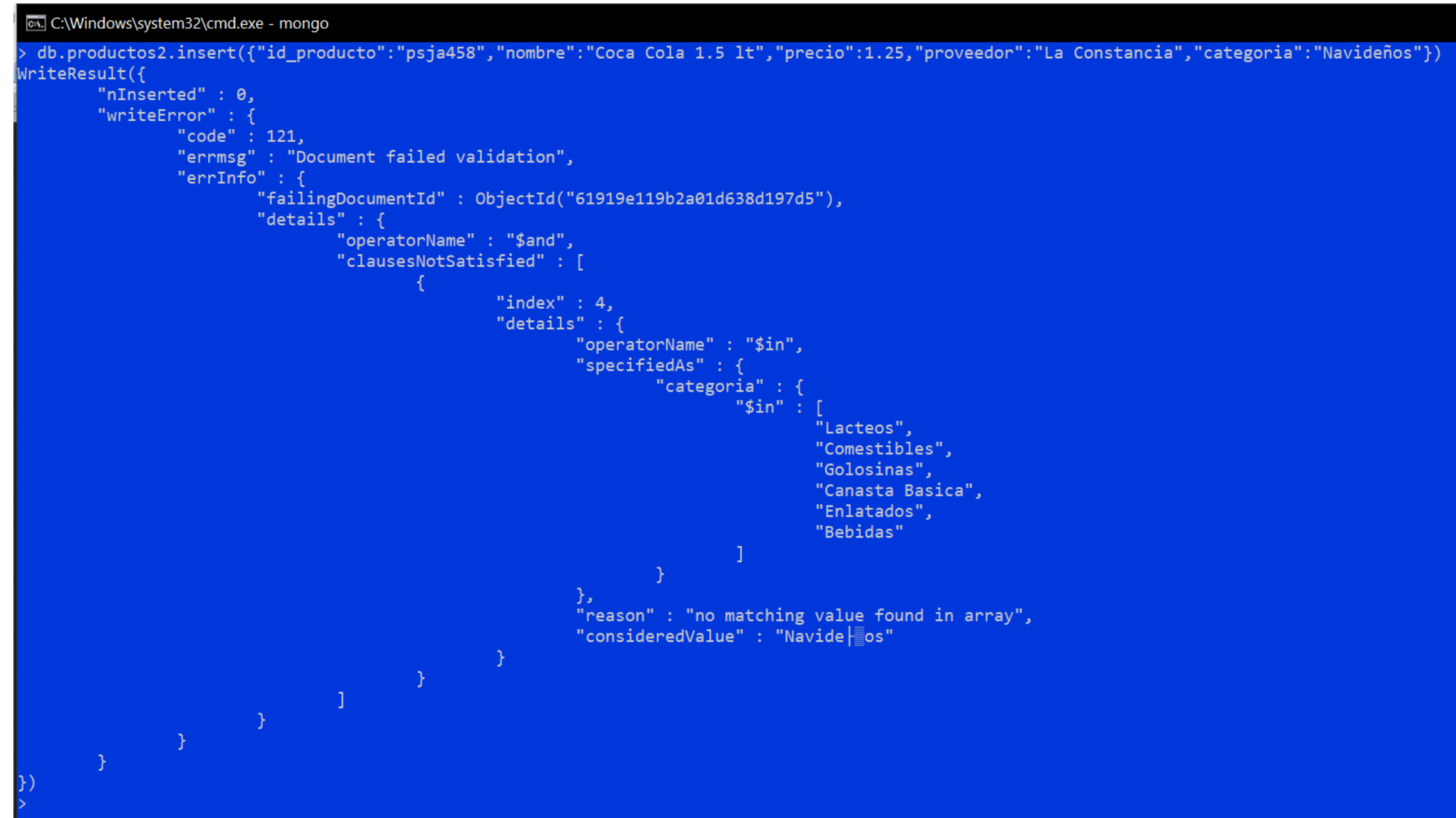


Imagen: Fuente propia (Validando Colecciones MongoDB)

Realizamos las respectivas correcciones, agregando una categoría existente y ahora el documento se inserta sin problema alguno.

```
C:\Windows\system32\cmd.exe - mongo
> db.productos2.insert({"id_producto":"psja458","nombre":"Coca Cola 1.5 lt","precio":1.25,"proveedor":"La Constancia","categoria":"Bebidas"})
WriteResult({ "nInserted" : 1 })
> db.productos2.find()
{ "_id" : ObjectId("61919b489b2a01d638d197d1"), "id_producto" : "psja456", "nombre" : "Azucar del cañal", "precio" : 0.52, "proveedor" : "Del Cañal", "categoria" : "Canasta Basica" }
{ "_id" : ObjectId("61919dc89b2a01d638d197d4"), "id_producto" : "psja457", "nombre" : "Sopa Laky", "precio" : 7, "proveedor" : "La Chapina", "categoria" : "Canasta Basica" }
{ "_id" : ObjectId("61919e3a9b2a01d638d197d6"), "id_producto" : "psja458", "nombre" : "Coca Cola 1.5 lt", "precio" : 1.25, "proveedor" : "La Constancia", "categoria" : "Bebidas" }
>
```

Imagen: Fuente propia (Validando Colecciones MongoDB)

Consultamos la información de nuestra colección con respecto a las validaciones y se nos muestra el listado de todo lo que la colección necesita validar antes de insertar un documento. Y adicional a eso nos muestra información acerca de los índices disponibles en nuestra colección:

```
C:\Windows\system32\cmd.exe - mongo
> db.getCollectionInfos({name: "productos2"})
[
  {
    "name" : "productos2",
    "type" : "collection",
    "options" : {
      "validator" : {
        "$and" : [
          {
            "id_producto" : {
              "$type" : "string"
            }
          },
          {
            "nombre" : {
              "$type" : "string"
            }
          },
          {
            "precio" : {
              "$type" : "double"
            }
          },
          {
            "proveedor" : {
              "$type" : "string"
            }
          },
          {
            "categoria" : {
              "$in" : [
                "Lacteos",
                "Comestibles",
                "Golosinas",
                "Canasta Basica",
                "Enlatados",
                "Bebidas"
              ]
            }
          }
        ]
      }
    },
    "info" : {
      "readOnly" : false,
      "uuid" : UUID("fc67e47f-df71-4c65-96c5-cc86e4c2be5f")
    },
    "idIndex" : {
      "v" : 2,
      "key" : {
        "_id" : 1
      },
      "name" : "_id_"
    }
  }
]
(i-search)`:`:
```

Imagen: Fuente propia (Validando Colecciones MongoDB)

Script utilizado en los ejemplos

```
db.createCollection('productos1')
```

```
db.productos1.insertMany([
{"id_pro":"prod1","nombre":"Sopa","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789,"cantidad":12},
{"id_pro":"prod2","nombre":"Fideos","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700,"cantidad":12},
{"id_pro":"prod3","nombre":"Chicles","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":12},
{"id_pro":"prod4","nombre":"Manzanas","proveedor":"El Rodeo","ciudad":"Ciudad Barrios","telefono":25569890,"cantidad":12},
{"id_pro":"prod5","nombre":"Pepino","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895,"cantidad":12},
{"id_pro":"prod6","nombre":"Ketchup","proveedor":"Super Josué","ciudad":"La Libertad","telefono":21345678,"cantidad":12},
{"id_pro":"prod7","nombre":"Ketchup","proveedor":"Dcasa","ciudad":"San Miguel","telefono":25056789,"cantidad":25},
{"id_pro":"prod8","nombre":"Pepino","proveedor":"Chapina","ciudad":"Chinameca","telefono":26706700,"cantidad":25},
{"id_pro":"prod9","nombre":"Manzanas","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":25},
{"id_pro":"prod10","nombre":"Peras","proveedor":"Franco","ciudad":"San Miguel","telefono":25340789,"cantidad":25},
{"id_pro":"prod11","nombre":"Chicles","proveedor":"El Rodeo","ciudad":"Ciudad Barrios","telefono":25569890,"cantidad":25},
{"id_pro":"prod12","nombre":"Fideos","proveedor":"Franco","ciudad":"San Salvador","telefono":23456895,"cantidad":19},
{"id_pro":"prod13","nombre":"Sopa","proveedor":"Super Josué","ciudad":"La Libertad","telefono":21345678,"cantidad":21}
])
```

En caso de modificar uno o varios de los campos de un documento procedemos a realizar la siguiente consulta solo que existe y solo va a actualizar los datos nuevos que identifique dentro de la consulta

```
db.productos1.update({"id_pro":"prod13" },{ $set: {"nombre":"Mirasol","proveedor":"DCASA","ciudad":"Sonsonate","Telefono":23456789,"cantidad":99} },{
upsert: true})
```

para borrar el campo agregado

```
db.productos1.updateMany( {"cantidad" : { $gte : 99 } }, { $unset : { "proveedor" : "DCASA" } } )
```

para agregar un campo

```
db.productos1.updateOne( {"nombre" : "Chicles" }, { $set : { "nuevo_campo" : "updateone" } } )
db.productos1.updateOne( {"cantidad" : { $eq : 12 } }, { $set : { "nuevo_campo" : "updateone#2" } } )
```

para agregar un campo con la sentencia update y multi para actualizar todos los documentos

```
db.productos1.update({"nombre" : "Ketchup" },{ $set:{ "agrega1" : "valor1","agrega2":"valor2" }},{multi:true})
```

para borrar

```
db.productos1.update({"nombre" : "Ketchup" }, { $unset : { "agrega1" : "valor1","agrega2":"valor2" } },{multi:true})
```

****actualizar o crear un documento si existiera o no en la coleccion (UPSERT)****

```
db.productos1.update({"id":4002 },{ $set: {"nombre":"Caramelos","proveedor":"Colombina","ciudad":"Montecarlos","cantidad":29} },{ upsert: true})
```

para modificar todos los documentos

```
db.productos1.updateMany( {"cantidad" : { $eq : 12 } }, { $set : { "Ubicados" : "Tu tienes cantidad 12" } } )
```

**** validación de colección****

```
db.createCollection( "productos2",
{ validator: { $and:
[
{ id_producto: { $type: "string" } },
{ nombre: { $type: "string" } },
{ precio: { $type: "double" } },
{ proveedor: { $type: "string" } },
{ categoria: { $in: [ "Lacteos", "Comestibles", "Golosinas", "Canasta Basica", "Enlatados", "Bebidas" ] } }
]
}
} )
```

++Creando un índice y dejándolo como único++

```
db.productos2.createIndex( { "id_producto": 1 }, { unique: true } )
```

++insertamos un documento++

```
db.productos2.insert({"id_producto":"psja456","nombre":"Azucar del cañal","precio":0.52,"proveedor":"Del Cañal","categoria":"Canasta Basica"})
```

++probando la validación++

```
db.productos2.insert({"id_producto":"psja457","nombre":"Sopa Laky","precio":"$7.00","proveedor":"La Chapina","categoria":"Canasta Basica"})
db.productos2.insert({"id_producto":"psja458","nombre":"Coca Cola 1.5 lt","precio":1.25,"proveedor":"La Constancia","categoria":"Comestibles"})
db.productos2.insert({"id_producto":"psja459","nombre":"Sprite","precio":1.50,"proveedor":"Coca Cola","categoria":"Bebidas"})
db.productos2.insert({"id_producto":"psja460","nombre":"Arroz San Francisco","precio":0.76,"proveedor":"Arrozera","categoria":"Canasta Basica"})
db.productos2.insert({"id_producto":"psja461","nombre":"Leche Anchor","precio":1.10,"proveedor":"Mazel","categoria":"Canasta Basica"})
db.productos2.insert({"id_producto":"psja462","nombre":"Leche Anchor","precio":1.10,"proveedor":"Carmela","categoria":"Golosinas"})
db.productos2.insert({"id_producto":"psja463","nombre":"Sprite","precio":1.50,"proveedor":"La Constancia","categoria":"Bebidas"})
```

```
db.getCollectionInfos({name: "productos2"})
```

Referencias

- Bermejo Corrales, M. (Julio de 2015). *Desarrollo de un sistema de autoescalado dinámico de base de datos distribuida MongoDB sobre una plataforma cloud OpenStack*. Obtenido de Desarrollo de un sistema de autoescalado dinámico de base de datos distribuida MongoDB sobre una plataforma cloud OpenStack:
<https://dehesa.unex.es/handle/10662/3759>
- MongoDB. (2021). *MongoDB*. Obtenido de MongoDB:
<https://www.mongodb.com/es/company>
- Narváez, M. E. (2020). *Análisis de Desempeño entre MONGODB y COUCHDB*. Obtenido de Análisis de Desempeño entre MONGODB y COUCHDB:
<https://doi.org/10.47187/perspectivas.vol2iss2.pp13-20.2020>
- OpenWebinars. (2021). *que es MongoDB*. Obtenido de que es MongoDB:
<https://openwebinars.net/blog/que-es-mongodb/>
- revistapublicando. (2017). *Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD*. Obtenido de Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD :
https://revistapublicando.org/revista/index.php/crv/article/view/398/pdf_342