**CRUD**

Se presenta a continuación operaciones que se pueden trabajar en MongodB.

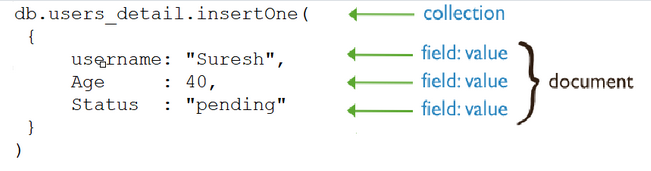
Para iniciar la creación de operaciones en mongodB se debe construir un nuevo documento por medio de los comandos:

* + - db.collection\_name.insertOne()
    - db.collection\_name.insertMany()

En el siguiente ejemplo se construye una colección por nombre “user\_detail” se ingresan algunos datos:

**Figura 1***.*

*Operación de inserción en BD*



*Nota: adaptado de CloudDuggu. (s/f).* <https://www.cloudduggu.com/mongodb/crud-operations/>

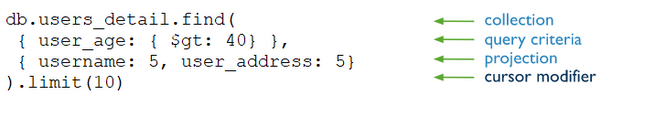
**Operaciones de lectura en MongoDB**

Las diferentes operaciones de lectura en MongoDB son implementadas para leer las colecciones alojadas en la base de datos.

A continuación, la sintaxis para lectura de una colección:

**Figura 2***.*

*Método find () en MongoDB*



*Nota: adaptado de CloudDuggu. (s/f).* <https://www.cloudduggu.com/mongodb/crud-operations/>

**Operaciones de actualización en MongoDB**

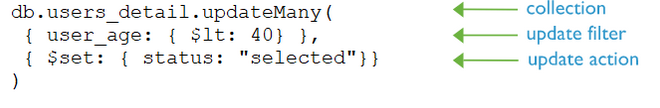
Las operaciones de actualización se implementan para modificar los datos dentro de las colecciones.

Los siguientes métodos son los implementados para realizar las respectivas actualizaciones:

* + - db.collection\_name.updateOne()
    - db.collection\_name.updateMany()
    - db.collection\_name.replaceOne()

**Figura 3***.*

*Operaciones de actualización en MongoDB*



*Nota: adaptado de CloudDuggu. (s/f).* <https://www.cloudduggu.com/mongodb/crud-operations/>

**Operaciones de eliminación en MongoDB**

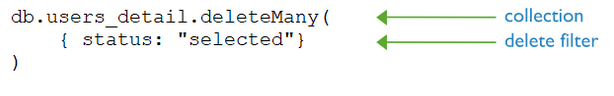
Las operaciones de eliminación se implementan para eliminar los datos dentro de las colecciones.

Los siguientes métodos son los implementados para realizar la eliminación de los datos en las colecciones:

* + - db.collection.deleteOne()
    - db.collection.deleteMany()

**Figura 4***.*

*Operaciones de eliminación en MongoDB*



*Nota: adaptado de* *CloudDuggu. (s/f).* <https://www.cloudduggu.com/mongodb/crud-operations/>

**Guardar datos en MongoDB**

En la base de datos MongoDB dispone de los siguientes métodos para realizar la inserción de datos:

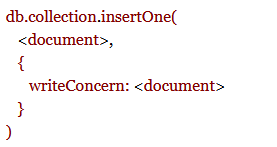
* + - db.collection.insertOne()

Este método es implementado para insertar un solo documento dentro de la colección, en caso de no existir dicha colección el código construirá una.

Sintaxis:

**Figura 5***.*

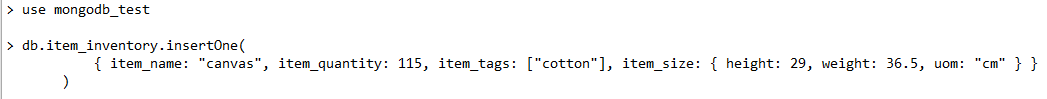
*Guardar datos en MongoDB*



Ejemplo: para el ejemplo nos ubicaremos en la base de datos mongodb\_test, e insertaremos un documento en la colección item\_inventory:

**Figura 6***.*

*Ejemplo guardar datos en MongoDB*



Para encontrar el documento ingresado en la colección, se realiza por medio del siguiente método:

*> db.item\_inventory.find( { item\_name: "canvas" } )*

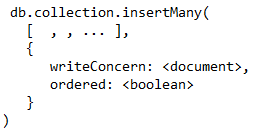
* + - db.collection.insertMany()

Este método es implementado para insertar varios documentos en la colección.

Sintaxis:

**Figura 7***.*

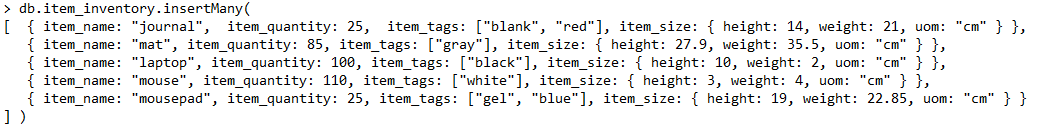
*Insertar varios documentos en MongoDB*



Ejemplo: por medio de los siguientes comandos se insertan varios documentos en la colección item\_inventory de la base de datos mongodb\_test:

**Figura 8***.*

*Ejemplo insertar varios documentos en MongoDB*



Por medio del método find() observamos todos los documentos dentro de la colección:

*> db.item\_inventory.find( {} )*

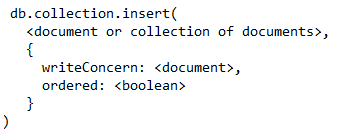
* + - db.collection.insert()

Este método es multifuncional, es usado para insertar un documento o varios documentos dentro de la colección.

Sintaxis:

**Figura 9.**

*Método insert() en MongoDB*



Ejemplo: insertar un documento dentro de la colección:

**Figura 10***.*

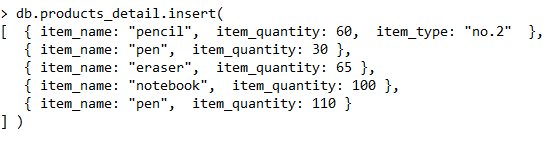
*Ejemplo método insert() en MongoDB*



Insertando varios documentos dentro de la colección:

**Figura 11***.*

*Ejemplo 2 método insert() en MongoDB*



**Modificación de datos en MongoDB**

En la base de datos MongoDB dispone de los siguientes métodos para realizar la modificación de datos:

* + db.collection.updateOne()

Este método es usado para modificar un único documento dentro de la colección.

Sintaxis:

**Figura 12***.*

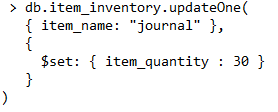
*Modificar un elemento en MongoDB*



Tomando como referencia los valores insertados en el método Insert(), actualizaremos el siguiente item por nombre "item\_name: "journal"" y usaremos el comando $set para asignar el valor de "item\_quantity: 30". Una vez ejecutado el comando, evidenciamos si el valor fue modificado por medio del método find().

**Figura 13***.*

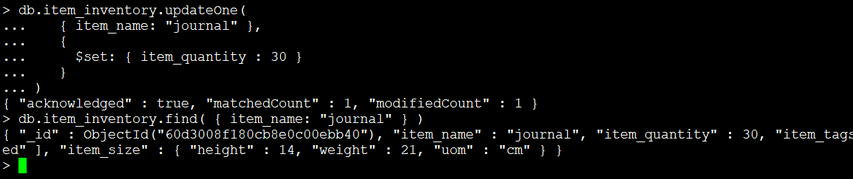
*Ejemplo modificar un elemento en MongoDB*



> db.item\_inventory.find( { item\_name: "journal" } )

**Figura 14***.*

*Ejecución del método updateOne en MongoDB*



* + db.collection.updateMany()

Este método se implementa para modificar todos los documentos que coindicen con el filtro aplicado.

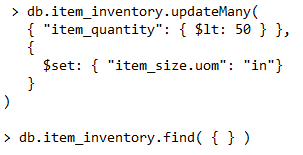
Sintaxis:

db.collection\_name.updateMany(filter, update, options)

Ejemplo:

**Figura 15***.*

*Sintaxis método updateMany en MongoDB*



* + db.collection.replaceOne()

Este método se implementa para modificar un solo documento según si la condición del filtro.

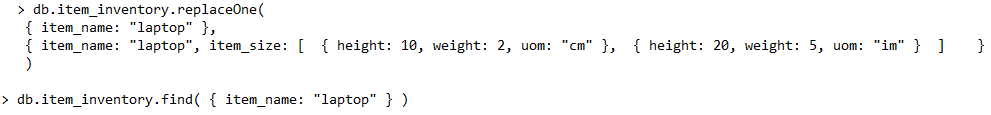
Sintaxis:

db.collection\_name.replaceOne(filter, replacement, options)

Ejemplo:

**Figura 16***.*

*Sintaxis método replaceOne en MongoDB*



**Eliminación de datos en MongoDB**

MongoDb nos brinda dos métodos para la eliminación de datos dentro de una colección, a continuación, se explica cada uno de los métodos:

* + - db.collection\_name.deleteOne()

Este método es implementado para eliminar un solo documento de la colección.

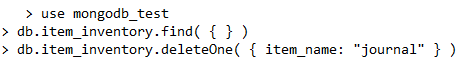
Sintaxis:

*db.collection.deleteOne({ field\_1: value\_1, .. })*

Ejemplo:

**Figura 17***.*

*Ejemplo método deleteOne en MongoDB*



* + - db.collection\_name.deleteMany()

Este método se implementa para la eliminación de todos los documentos dentro de la colección, en caso de aplicar un filtro solo eliminara los documentos que coincidan con la condición.

Sintaxis:

**Figura 18***.*

*Método deleteMany en MongoDB*



Ejemplo:

**Figura 19***.*

*Ejemplo del método deleteMany en MongoDB*



* Consultas.

En MongoDB también podemos realizar una búsqueda de todos los documentos dentro de una colección por medio del método Find(), como en el caso de SQL sería un SELECT \* FROM.

Ejemplo:

> db.inventory\_records.find( { } )

En SQL seria su equivalente:

SELECT \* from inventory\_records;

* + - db.collection.findOne()

Este método se implementa para retornar la búsqueda de un único documento.

Sintaxis:

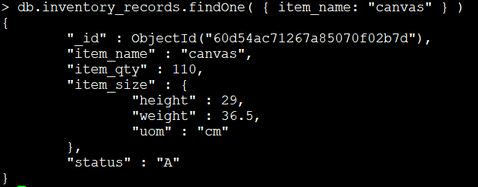
db.collection\_name.findOne()

Ejemplo:



**Figura 20***.*

*Método buscar en MongoDB*



* + - db.collection.find().pretty()

Este método se implementa para que el resultado retorne un formato de fácil lectura.

Sintaxis:

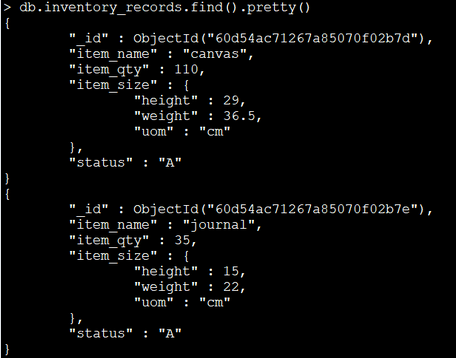
db.collection\_name.find(<query>).pretty()

Ejemplo:



**Figura 21***.*

*Método pretty en MongoDB*



* + - Condición AND en MongoDB

Este operador ***$and*** en MongoDB permite la comparación de dos documentos y retorna el resultado cuando las condiciones coinciden.

Sintaxis:

db.collection\_name.find({ $and: [ {key\_name\_1:value\_detail\_1}, {key\_name\_2:value\_detail\_2} ] })

Ejemplo:

> db.inventory\_records.find({$and:[ {"status":"A"}, {item\_qty: { $lt: 30}} ]})

* + - Condición OR en MongoDB

Este operador $or en MongoDB permite la comparación de dos documentos y retorna el resultado al menos una de las condiciones coincide.

Sintaxis:

db.collection\_name.find({ $or: [ {keyname\_1:valuedetail\_1}, {keyname\_2:valuedetail\_2} ] })

Ejemplo:

> db.inventory\_records.find({$or:[ {"status":"D"}, {item\_qty: { $lt: 30}} ]})

**REFERENCIAS BIBLIOGRÁFICAS**

CloudDuggu. (s/f). *Operaciones CRUD de MongoBD.* CloudDuggu. <https://www.cloudduggu.com/mongodb/crud-operations/>