

CICLO 4a

[FORMACIÓN POR CICLOS]

Desarrollo de APLICACIONES WEB

Creación y consulta datos en MongoDB





Cargando datos en la consola

Desde la consola de mongoDB digitamos:

```
$> use nuevaDB
$> documento = {id: 1, nombre: "MongoDB",
url: "http://www.mongodb.org", tipo: "Documentacion"}
```

- Para insertar el documento anterior usamos: \$> db.nueva_coleccion.insert(documento)
- De esta manera tenemos nuestra <u>primera base de datos, colección y documento.</u>

Nota: si se va a copiar y pegar en la consola, es necesario copiar el comando después del símbolo ">". Si se va a pegar sobre el PowerShell, se deben usar las teclas ctrl + v y luego hacer clic derecho.



Cargando ficheros en la consola



Para confirmar los cambios anteriores usamos el comando:

```
$>show collections
nueva_colección
system.indexes
```

 Para verificar que se almacenó un documento en la nueva colección usamos el comando:

Operaciones de lectura

Para hacer una consulta usamos el comando:

```
$>db.collection.find(query, projection)
  query: especifica condición o filtro
  projection: especifica los campos a proyectar
```

• Comparación entre la consulta tradicional y las consultas en MongoDB:

```
SELECT *
FROM inventory

SELECT id,
item {},
FROM inventory {item:1})

SELECT item
FROM inventory

SELECT item
FROM inventory

db.inventory.find(
{},
{item:1})
```



MongoDB: consultas básicas

• Comparación entre la consulta tradicional y las consultas en MongoDB:

```
mongoDB
```

```
SELECT item
                          db.inventory.find(
                            {status: "A"},
FROM inventory
WHERE status="A"
                            {item:1, id:0})
SELECT item
                         db.inventory.find(
                           {status:{$ne:"A"}},
FROM inventory
WHERE status <>"A"
                           {item:1, id:0})
SELECT *
                         db.inventory.find(
                           {status: "A"},
FROM inventory
WHERE status="A" or
                                {item:"paper"}]})
      item = "paper"
```

Crear documentos

Una colección en MongoDB es lo mismo que una tabla en MySQL.

Para crear una colección en MongoDB se usa el método createCollection ():

Ejemplo: Crear una colección llamada "clientes".

MongoDB espera hasta que se haya insertado un documento antes de que realmente cree la colección.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";
MongoClient.connect(url, function(err, db) {
if (err) throw err;
 console.log("Database created!");
 db.close();
});
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
 if (err) throw err;
 var dbo = db.db("mydb");
 dbo.createCollection("customers", function(err, res) {
 if (err) throw err;
 console.log("Collection created!");
 db.close();
                                                           mongoDB
});
```



Importante: en MongoDB, una colección no se crea hasta que obtiene contenido.

Crear documentos

Para insertar un registro, o documento como se llama en MongoDB, en una colección, se usa el método insertOne ().

Un documento en MongoDB es lo mismo que un registro en MySQL.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
if (err) throw err;
 var dbo = db.db("mydb");
var myobj = { name: "Company Inc", address: "Highway 37"
 dbo.collection("customers").insertOne(myobj,
function(err, res) {
 if (err) throw err;
 console.log("1 document inserted");
 db.close();
 });
});
                                                     ♥ mongoDB
```



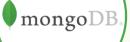
Importante: en MongoDB, una colección no se crea hasta que obtiene contenido.

Recuperar documentos

En MongoDB se usan los métodos find y findOne para buscar datos en una colección.

Al igual que la instrucción SELECT se usa para buscar datos en una tabla en una base de datos MySQL.

```
//Encuentre el primer documento en la colección de
clientes
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
 if (err) throw err;
 var dbo = db.db("mydb");
 dbo.collection("customers").findOne({},
function(err, result) {
 if (err) throw err;
 console.log(result.name);
 db.close();
 });
});
```



Eliminar documentos

Para eliminar un registro o documento, como se llama en MongoDB, se usa el método deleteOne (). El primer parámetro del método deleteOne () es un objeto de consulta que define qué documento eliminar.

```
//Borre el documento con la dirección "Montaña 21":
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
if (err) throw err;
var dbo = db.db("mydb");
 var myquery = { address: 'Mountain 21' };
 dbo.collection("customers").deleteOne(myquery,
function(err, obj) {
if (err) throw err;
 console.log("1 document deleted");
 db.close();
});
                                               mongoDF
```



Nota: si la consulta encuentra más de un documento, solo se elimina la primera aparición.