Trabajo Tema 4 - Otros modelos de datos para SI

Grupo MATARRATAS (A2):
Francisco Javier Jiménez Legaza
Ángel Gómez Ferrer
Cristian Fernández Jiménez
Santiago Muñoz Castro
Raúl Rodríguez Pérez
Raúl Castro Moreno

ÍNDICE

1. Descarga e instalación del SGBD

El desarrollo de este trabajo ha sido realizado en Ubuntu, eligiendo el SGBD NOSQL y Open Source **mongoDB** e instalándolo siguiendo el tutorial: link

Para empezar abrimos el terminal y ejecutamos los siguientes comandos:

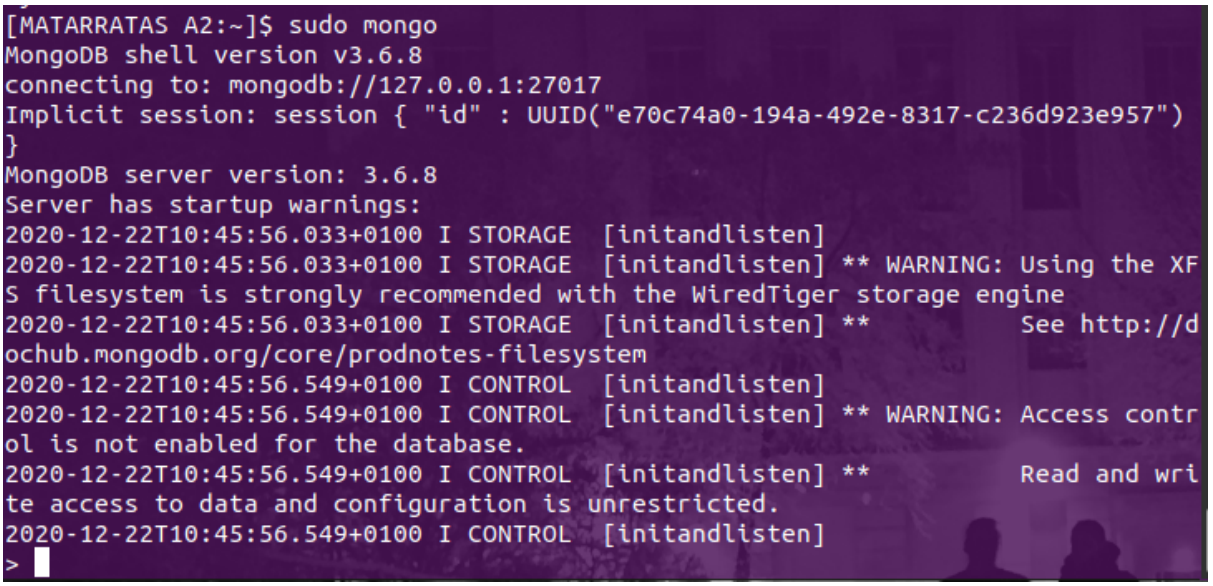
1 - sudo apt update (Actualizamos repositorios)

2 - sudo apt upgrade (Nos va a actualizar todos los paquetes instalados que tengan alguna nueva versión)

3 - sudo apt install mongodb (Descarga e instala el repositorio MongoDB).

4 - sudo systemctl start mongodb (Inicia el proceso de MongoDB).

5 - sudo mongo (Lanza nuestro nuevo SGBD MongoDB).



2. Descripción DDL y DML utilizado

Ahora procederemos a explicar las sentencias básicas del DDL y DML de nuestra SGBD MongoDB.

Para crear una base de datos utilizamos la siguiente sentencia:

***> use <Nombre de la BD>*.**

(Este comando se usa también para moverse de base de datos, pero en caso de que no esté creada, la crea)

Para borrar una base de datos que estamos utilizando usamos la siguiente sentencia:

> db.dropDatabase().

DDL:

Para crear una colección (lo más parecido a una tabla de una base de datos relacional) utilizamos el siguiente comando con el nombre de la colección:

> db.createCollection(“Nombre de la coleccion”)

MongoDB no requiere de una especificación sobre las variables y el tipo de atributos que tiene dicha tabla, este los conocerá cuando se inserten directamente los valores de esta.

Para borrar una colección de la BD usamos el siguiente comando:

> db.nombreDeLaColeccion.drop()

DML:

Para insertar datos en las tablas utilizamos el comando:

> db.<nombreColeccion>.insert(documento) por ejemplo:

> db.<nombreColeccion>.insert({
“<nombreAtributo>”: “valor”,
“<nombreAtributo2>”: “valor”,
...
...
})

Valor irá entre comillas si es un string, también podríamos asignar un valor booleano o numérico (sin “ ”);

Para actualizar el valor de un atributo de nuestra colección (lo que equivale a una sentencia UPDATE en SQL) utilizamos la siguiente sentencia.

> db.<nombreColeccion>.update(
{“<AtributoIdentificativo>”:”<valorAEncontrar>”},
{$set: {“<AtributoACambiar>”:”<nuevoValor>”} })

Con ‘AtributoIdentificativo’ nos referimos a la condición identificativa para saber qué colecciones queremos modificar y una vez encontradas le decimos que modifique ($set) los atributos que queramos con el ‘nuevoValor’ que le digamos, también podemos utilizar $unset para eliminar un valor (ponerlo en null) de algún campo existente (de lo contrario no hace nada).

Para consultar nuestra tabla bastaría con el comando:

> db.<nombreColeccion>.find()

Esta sentencia nos buscaría y mostraría la colección que buscamos pero lo haría de forma poco representable, pero si le añadimos **.pretty()** al final de la sentencia, nos aparecerá con sangrado y de forma bonita y entendible.

> db.<nombreColeccion>.find().pretty()

Para filtrar la tabla elegida a mostrar únicamente los campos que queramos podemos utilizar una condición de la siguiente manera:

> db.<nombreColeccion>.find({ <nombreAtributo>:<Condición>})

Operacion	Ejemplo
Igualdad	{ "stock": 0 }
Menor Que	{ "valor": {$lt: 15.0} }
Menor o Igual Que	{ "valor": {$lte: 16.0} }
Mayor Que	{ "valor": {$gt: 18.0} }
Mayor o Igual Que	{ "valor": {$gte: 16.0} }
No es Igual	{ "valor": {$ne: 0} }
AND	{ {key1: value1, key2:value2} }

Estas son algunas de las condiciones que podemos utilizar

Para borrar una fila que se identifique con un valor en específico usamos el siguiente comando:

> db.<nombreColeccion>.deleteOne(
{ <nombreAtributo>:<Condición>})

3. Sentencias de prueba de MongoDB

Primero creamos una base de datos ‘trabajoT4’ mediante la siguiente sentencia:

> use trabajoT4 (Crea la BD)

Vamos a crear la colección de *jugadores*:

> db.createCollection(“jugadores”)

Como queremos crear dos colecciones repetimos el comando pero para crear la colección *equipos*

> db.createCollection(“equipos”)

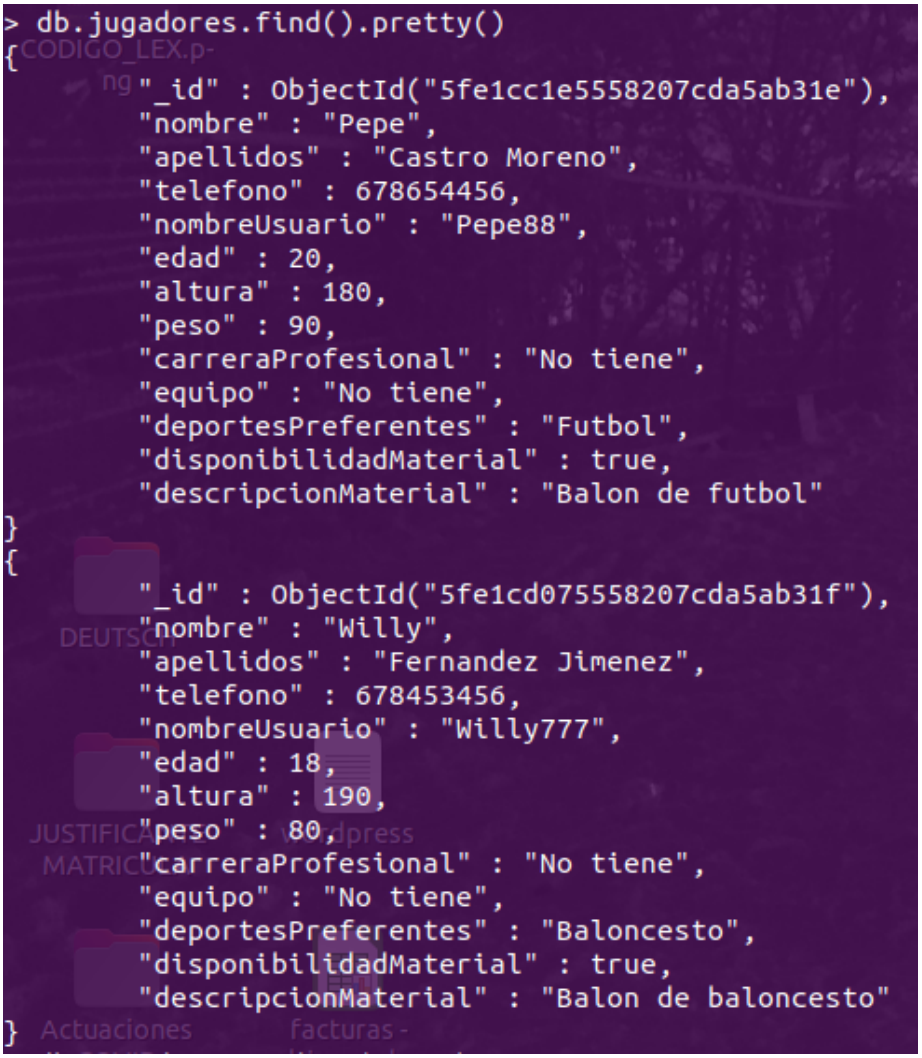
Ahora vamos a realizar unas inserciones en dichas colecciones.
Empezaremos con la inserción de unos jugadores en la colección de jugadores:

> db.jugadores.insert({"nombre": "Pepe", "apellidos": "Castro Moreno", "telefono": 678654456, "nombreUsuario": "Pepe88", "edad": 20, "altura": 180, "peso": 90, "carreraProfesional": "No tiene", "equipo": "No tiene", "deportesPreferentes": "Futbol", "disponibilidadMaterial": true, "descripcionMaterial": "Balon de futbol"})

> db.jugadores.insert({"nombre": "Willy", "apellidos": "Fernandez Jimenez", "telefono": 678453456, "nombreUsuario": "Willy777", "edad": 18, "altura": 190, "peso": 80, "carreraProfesional": "No tiene", "equipo": "No tiene", "deportesPreferentes": "Baloncesto", "disponibilidadMaterial": true, "descripcionMaterial": "Balon de baloncesto"})

Ahora con el siguiente comando, podremos visualizar lo que tiene almacenado la colección, comprobando así que la inserción se ha realizado correctamente.

> db.jugadores.find().pretty()



Ahora vamos a realizar 2 inserciones en la colección de equipos:

> db.equipos.insert({"nombreEquipo": "betis", "tag": "bts", "deporte": "Futbol", "lider":"Pepe88"})
WriteResult({ "nInserted" : 1 })

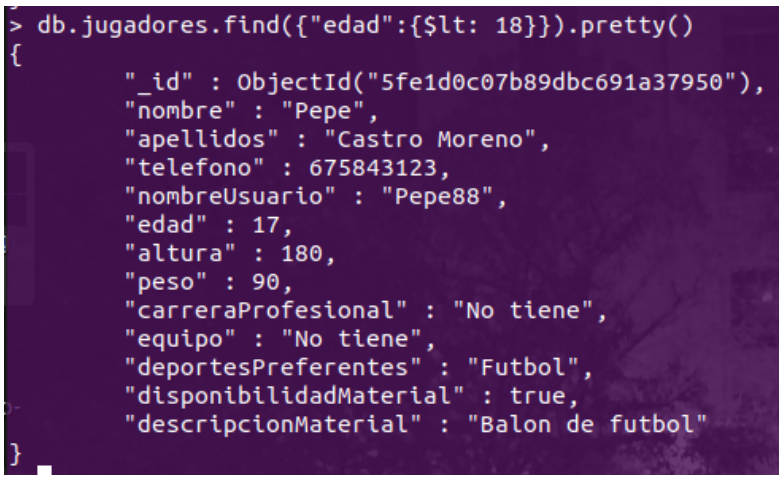
> db.equipos.insert({"nombreEquipo": "numanciafc", "tag": "nmc", "deporte": "Baloncesto", "lider":"Willy777"})
WriteResult({ "nInserted" : 1 })

Y comprobamos que se haya realizado correctamente

Ahora vamos a realizar una modificación de un jugador de la colección jugadores, de la siguiente forma:

>db.jugadores.update({"nombre":"Pepe"},{$set:{"telefono":675843123, "edad":17}})

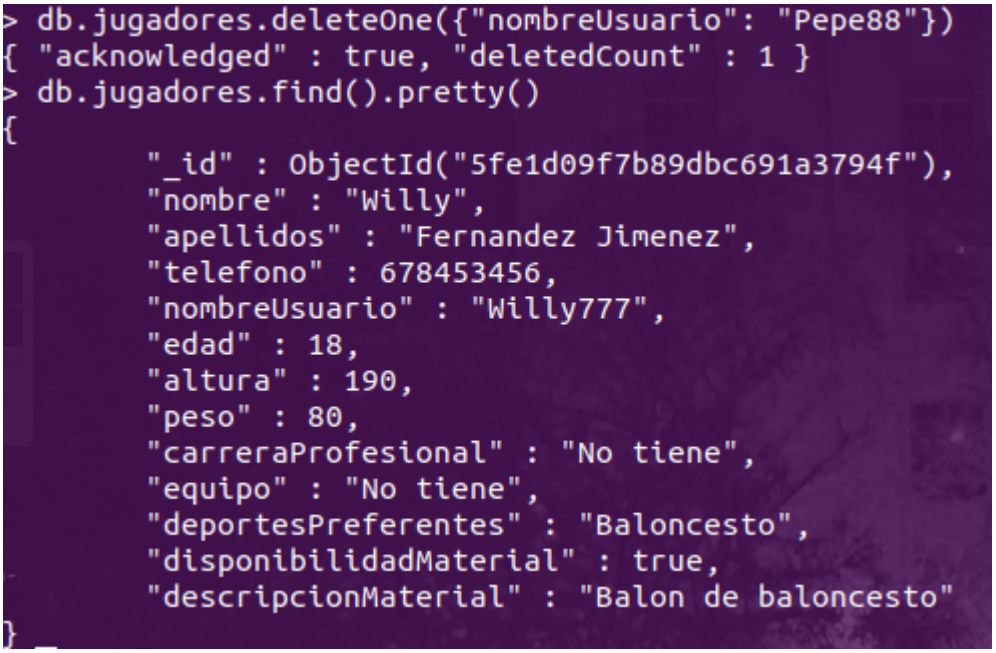
Realizamos una consulta según una condición la cual es, los jugadores con una edad menor de 18 años. Como los 2 jugadores tenían más de 18, no deberían de aparecer, pero puesto que hemos modificado la edad de uno de ellos a 17, nos aparece lo siguiente al realizar la consulta.
> db.jugadores.find({"edad":{$lt: 18}}).pretty()



Ahora borramos un objeto de la colección

> db.jugadores.deleteOne({"nombreUsuario": "Pepe88"})

Y comprobamos si se ha realizado



4. Mecanismo de conexión al SGBD desde una aplicación

Para explicar el mecanismo de conexión desde una aplicación vamos a utilizar el lenguaje Java.

Tras instalar el Driver de MongoDB para Java, tendremos que resolver algunas de las dependencias lo cual se puede hacer con Gradle o Maven (explicar esto se haría muy denso). Una vez tenemos todo instalado a la hora de programar conectamos a la base de datos de esta forma:

```
MongoClient mongoClient = new MongoClient(new MongoClientURI("mongodb://localhost:27017"));
```

Donde *mongodb://localhost:27017* será la dirección donde está instalado MongoDB.

Si utilizamos una instancia local en el puerto por defecto utilizaremos:

```
MongoClient mongoClient = new MongoClient();
```

De esta forma seleccionamos las bases de datos:

```
DB database = mongoClient.getDB("TheDatabaseName");
```

Y así podemos utilizar una de las colecciones:

```
DBCollection collection = database.getCollection("TheCollectionName");
```

Y ya podremos empezar a realizar operaciones.

5. Discusión sobre adecuación para implementar el SI de la práctica

MongoDB es un SGBD NoSQL orientado a documentos, almacenando colecciones utilizando JSON. Esto es útil cuando se quieren realizar proyectos apoyados en grandes conjuntos de datos sin estructurar, puesto que no están atados a ningún esquema aunque sí garantizan y procesamiento muy rápido.

Sin embargo, no es adecuado para nuestra práctica, pues requiere una buena estructuración de los datos entre sí, a la hora de almacenar eventos y equipos relacionados entre sí muy fuertemente.

6. Bibliografía

https://docs.mongodb.com/manual/reference/mongo-shell/

https://docs.mongodb.com/manual/reference/method/

https://www.mongodb.com/blog/post/getting-started-with-mongodb-and-java-part-i

https://linuxhint.com/install_mongodb_ubuntu_20_04/

https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/mongodb-presentacion-y-comparacion-con-mysql/