



Ikaskuntza Birtual eta Digitalizatuen LHII
CIFP de Aprendizajes Virtuales y Digitalizados

Tarea Evaluativa

Víctor Aguinaco

DAM – Acceso a Datos UD06

Curso: 2021/22





ÍNDICE

1 EJERCICIO 1.....	1
2 EJERCICIO 2.....	4
3 EJERCICIO 3.....	5

1 EJERCICIO 1

Realiza las siguientes consultas:

Escribe una consulta que muestre el nombre, género, y la duración (runtime) de las primeras 10 series ordenados por runtime y nombre y muéstralo de forma indentada.

```
db.movies.find({}, {"name" : 1, "genres" : 1, "runtime" : 1}).sort({"runtime":1, "name":1}).pretty().limit(10)
```

Escribe una consulta que muestre el nombre, el género y la puntuación de las series del género "Fantasy" (no exclusivamente) y tienen una puntuación superior a 7.

```
db.movies.find({"genres": "Fantasy", "rating.average":{"$gt : 7}}, {"name":1, "genres":1, "rating.average":1}).pretty()
```

Escribe una consulta que muestre el nombre, el género y la puntuación de las series que tienen una puntuación entre 9 y 9.5 (ambos incluidos).

```
db.movies.find({"rating.average": {$gte : 9, $lte: 9.5}}, {"name":1, "genres":1, "rating.average":1}).pretty()
```



Escribe una consulta que muestre el nombre y el género de las series que tienen como género exclusivamente Drama y Comedy.

```
db.movies.find({"genres":["Drama","Comedy"]},{ "name":1,"genres":1}).pretty()
```

```
root@7b4069347240: /var/lib/mongo
> db.movies.find({"genres":["Drama","Comedy"]},{ "name":1,"genres":1}).pretty()
{
  "_id" : ObjectId("624f1eccf9a4e991323eb75c"),
  "name" : "Californication",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb775"),
  "name" : "Red Band Society",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb7c5"),
  "name" : "Transparent",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb7e0"),
  "name" : "Shameless",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb7f4"),
  "name" : "BoJack Horseman",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb804"),
  "name" : "The Carrie Diaries",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb814"),
  "name" : "Happyland",
  "genres" : [
    "Drama",
    "Comedy"
  ]
}
}
```



Escribe una consulta que muestre el nombre y el género de las series cuyo nombre contiene la cadena "the" y no es del género Comedy

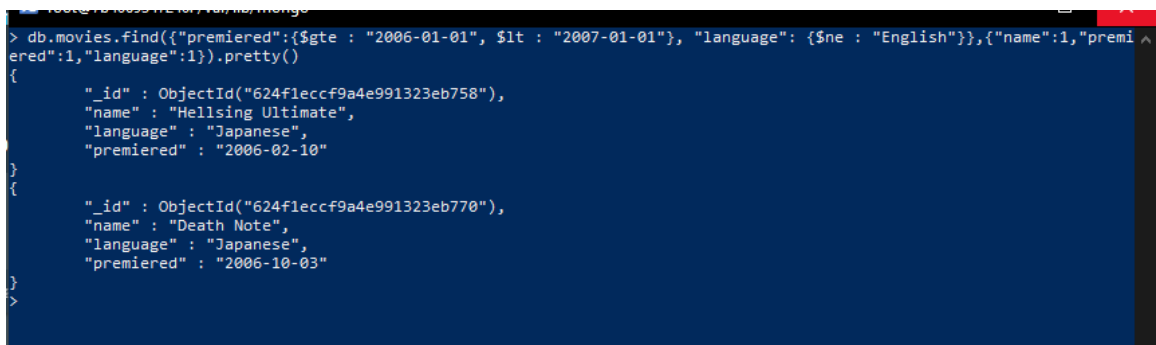
```
db.movies.find({"name": {$regex: /the/}, "genres": {$ne : "Comedy"}}, {"name":1, "genres":1}).pretty()
```

Escribe una consulta que muestre el nombre y la fecha de estreno (premiered) de las series cuya fecha de estreno es posterior a 2014-10-01.

```
db.movies.find({"premiered": {$gt : "2014-10-01"}}, {"name":1, "premiered":1}).pretty()
```

Escribe una consulta que muestre el nombre, la fecha de estreno y el idioma (language) de las series cuya fecha de estreno se es el año 2006 y que no sean de lengua inglesa.

```
db.movies.find({"premiered": {$gte : "2006-01-01", $lt : "2007-01-01"}, "language": {$ne : "English"}}, {"name":1, "premiered":1, "language":1}).pretty()
```



```
> db.movies.find({"premiered": {$gte : "2006-01-01", $lt : "2007-01-01"}, "language": {$ne : "English"}}, {"name":1, "premiered":1, "language":1}).pretty()
{
  "_id" : ObjectId("624f1eccf9a4e991323eb758"),
  "name" : "Hellsing Ultimate",
  "language" : "Japanese",
  "premiered" : "2006-02-10"
}
{
  "_id" : ObjectId("624f1eccf9a4e991323eb770"),
  "name" : "Death Note",
  "language" : "Japanese",
  "premiered" : "2006-10-03"
}
>
```

Escribe una consulta que muestre el nombre y el idioma (language) de las series cuya genero no está definido, ordenado por nombre.

```
db.movies.find({"genres": {$size : 0}}, {"name":1, "language":1}).sort({"name":1}).pretty()
```

Escribe una consulta que muestre el nombre, y la duración (runtime) de las series que duran 30 minutos o múltiplos de 30, ordenados de menor a mayor duración y por nombre.

```
db.movies.find({"runtime": {$mod : [30,0]}}, {"name":1, "runtime":1}).sort({"runtime": 1, "name":1}).pretty()
```

Realiza la consulta e en Java

***Ver archivo zip adjunto**



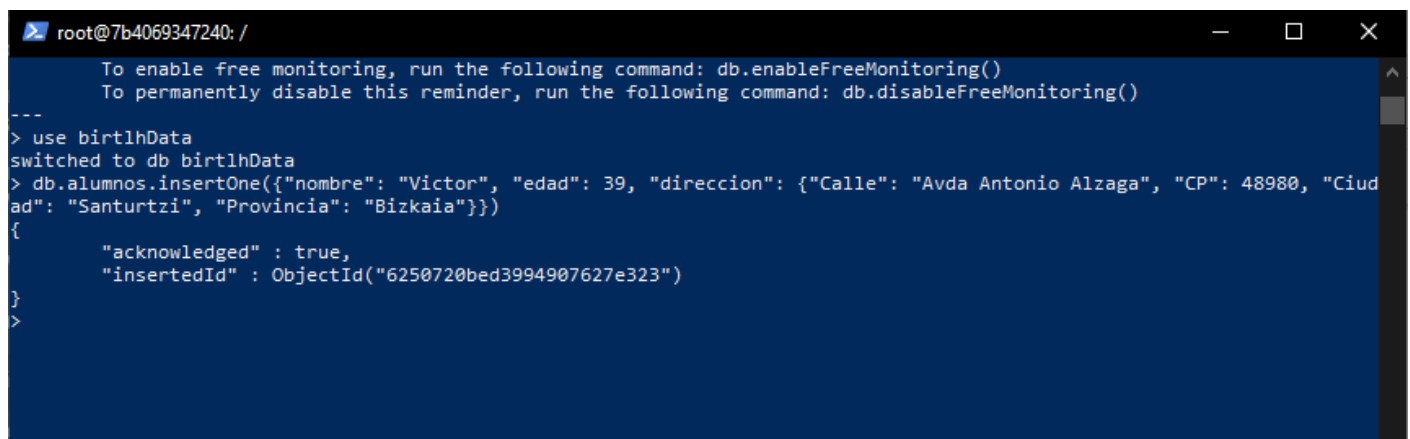
2 EJERCICIO 2

Realiza las siguientes operaciones de insertar, actualizar y borrar en una BD nueva llamada birthData:

CREACIÓN → insertOne(), insertMany().

Introduce en la colección alumnos un nuevo alumno con los campos nombre, edad, dirección. Dirección será a su vez un nuevo documento (nested document) que contendrá los siguientes datos: calle, CP, ciudad, provincia.

```
db.alumnos.insertOne({"nombre": "Victor", "edad": 39, "direccion": {"Calle": "Avda Antonio Alzaga", "CP": 48980, "Ciudad": "Santurtzi", "Provincia": "Bizkaia"}})
```



```
root@7b4069347240: /
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use birthData
switched to db birthData
> db.alumnos.insertOne({"nombre": "Victor", "edad": 39, "direccion": {"Calle": "Avda Antonio Alzaga", "CP": 48980, "Ciudad": "Santurtzi", "Provincia": "Bizkaia"}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("6250720bed3994907627e323")
}
```

Introduce en la colección alumnos dos nuevos alumnos a la vez con los campos nombre, edad, dirección. Dirección será a su vez un nuevo documento (nested document) que contendrá los siguientes datos: calle completa, CP, ciudad, provincia. De estos dos nuevos alumnos guardaremos también sus hobbies en un array con el mismo nombre.

```
db.alumnos.insertMany([{"nombre": "Pablo", "edad": 26, "direccion": {"Calle": "Plaza Nueva", "CP": 48001, "Ciudad": "Bilbao", "Provincia": "Bizkaia"}, "hobbies": ["Surf", "Videojuegos"]}, {"nombre": "Elena", "edad": 32, "direccion": {"Calle": "Landaberde", "CP": 01010, "Ciudad": "Gasteiz", "Provincia": "Araba"}, "hobbies": ["Musica", "Cine", "Videojuegos"]}]
```

ACTUALIZACIÓN

Actualiza e introduce tres hobbies también al primer alumno insertado

```
db.alumnos.updateOne({_id: ObjectId("6250720bed3994907627e323")}, {$set: {"hobbies": ["Cine", "Videojuegos", "Musica"]}})
```

BORRADO

Borra los alumnos que tenga un determinado hobby

```
db.alumnos.deleteOne({"hobbies": "Surf"})
```

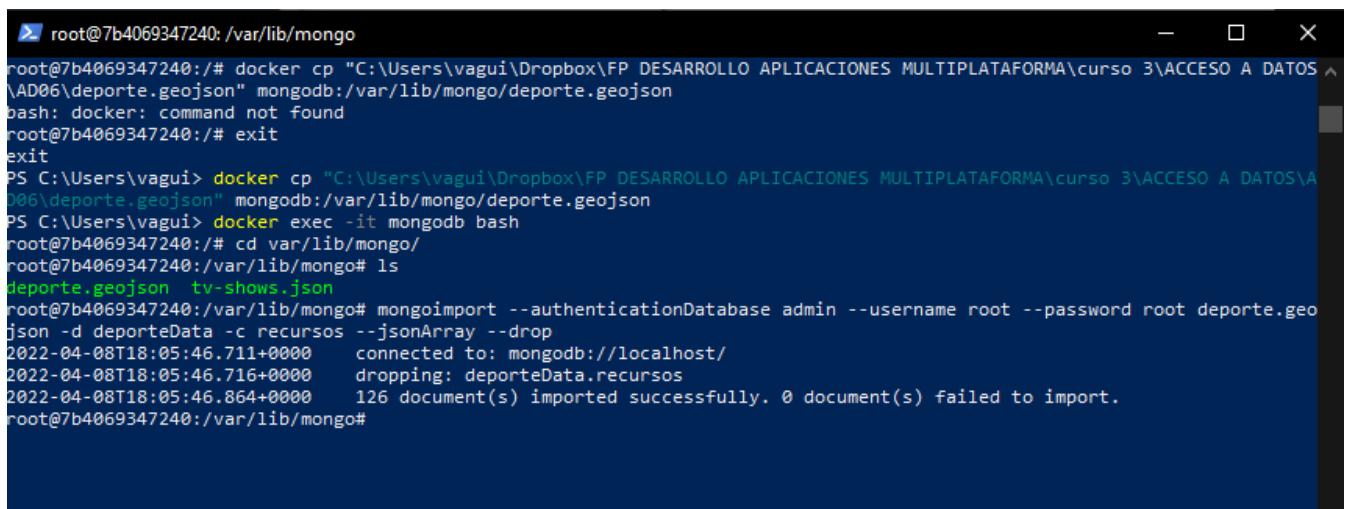


3 EJERCICIO 3

Ejercicio libre: Elige una BD geojson de la web Open Data Euskadi u otra similar y realiza 3 consultas geoespaciales diferentes utilizando al menos los operadores que hemos visto en la tarea de aprendizaje 3. (\$near, \$geoWithin...)

Para este ejercicio he utilizado el archivo deporte.geojson que contiene todos los recursos deportivos de Euskadi listados por OpenDataEuskadi. Tras copiarlo al container de mongodb lo he importado a la base de datos con el siguiente comando:

```
mongoimport --authenticationDatabase admin --username root --password root deporte.geojson -d deporteData -c recursos --jsonArray --drop
```



```
root@7b4069347240: /var/lib/mongo
root@7b4069347240:/# docker cp "C:\Users\vagui\Dropbox\FP DESARROLLO APLICACIONES MULTIPLATAFORMA\curso 3\ACCESO A DATOS\AD06\deporte.geojson" mongodb:/var/lib/mongo/deporte.geojson
bash: docker: command not found
root@7b4069347240:/# exit
exit
PS C:\Users\vagui> docker cp "C:\Users\vagui\Dropbox\FP DESARROLLO APLICACIONES MULTIPLATAFORMA\curso 3\ACCESO A DATOS\AD06\deporte.geojson" mongodb:/var/lib/mongo/deporte.geojson
PS C:\Users\vagui> docker exec -it mongodb bash
root@7b4069347240:/# cd var/lib/mongo/
root@7b4069347240:/var/lib/mongo# ls
deporte.geojson  tv-shows.json
root@7b4069347240:/var/lib/mongo# mongoimport --authenticationDatabase admin --username root --password root deporte.geojson -d deporteData -c recursos --jsonArray --drop
2022-04-08T18:05:46.711+0000    connected to: mongodb://localhost/
2022-04-08T18:05:46.716+0000    dropping: deporteData.recursos
2022-04-08T18:05:46.864+0000    126 document(s) imported successfully. 0 document(s) failed to import.
root@7b4069347240:/var/lib/mongo#
```



Consulta 1:

Recursos a menos de 5 km de mi vivienda habitual (43.3233198, -3.0326453)

```
db.recursos.createIndex({geometry: "2dsphere"})
```

```
db.recursos.find({geometry: {$near: {$geometry: {type: "Point", coordinates: [-3.0326453, 43.3233198]}}, $maxDistance: 5000}}, {"properties.documentname": 1, "properties.templateType": 1, "properties.municipality": 1, "properties.sporttype": 1}).pretty()
```

```
root@7b4069347240: /var/lib/mongo
> db.recursos.find({geometry: {$near: {$geometry: {type: "Point", coordinates: [-3.0326453, 43.3233198]}}, $maxDistance: 5000}}, {"properties.documentname": 1, "properties.templateType": 1, "properties.municipality": 1, "properties.sporttype": 1}).pretty()
pretty()
  "_id" : ObjectId("6250797af0ae5127ff07db1b"),
  "properties" : {
    "documentname" : "Puerto Deportivo de Santurtzi",
    "templateType" : "Instalaciones deportivas",
    "sporttype" : "Puerto deportivo",
    "municipality" : "Santurtzi"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07dad3"),
  "properties" : {
    "documentname" : "Escuela Náutica Sotavento",
    "templateType" : "Empresas de turismo activo y deporte aventura",
    "sporttype" : "Naturaleza",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07db21"),
  "properties" : {
    "documentname" : "Real Club Marítimo del Abra Real Sporting Club",
    "templateType" : "Instalaciones deportivas",
    "sporttype" : "Club náuticos",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07daea"),
  "properties" : {
    "documentname" : "Getxo - Puerto deportivo del Real Club Marítimo del Abra",
    "templateType" : "Instalaciones deportivas",
    "sporttype" : "Puerto deportivo",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07dada"),
  "properties" : {
    "documentname" : "Dive TARPOON",
    "templateType" : "Empresas de turismo activo y deporte aventura",
    "sporttype" : "Naturaleza",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07db14"),
  "properties" : {
    "documentname" : "Puerto Deportivo El Abra-Getxo",
    "templateType" : "Instalaciones deportivas",
    "sporttype" : "Puerto deportivo",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07dabb"),
  "properties" : {
    "documentname" : "As de Guía",
    "templateType" : "Empresas de turismo activo y deporte aventura",
    "sporttype" : "Naturaleza",
    "municipality" : "Getxo"
  }
}

  "_id" : ObjectId("6250797af0ae5127ff07dad9"),
  "properties" : {
    "documentname" : "Escuela Vasca de Parapente y Paramotor (EVP-P)",
    "templateType" : "Empresas de turismo activo y deporte aventura",
    "sporttype" : "Naturaleza",
    "municipality" : "Getxo"
  }
}
```


Consulta 2:

Nombres de los recursos que se encuentran dentro del Gran Bilbao:

punto 1: 43.332796863600436, -3.125763829433292
punto 2: 43.37844321062219, -3.0064409724419474
punto 3: 43.23977944012737, -2.819887699757718
punto 4: 43.19792388375848, -2.9441081367002004

```
const p1 = [-3.125763829433292, 43.332796863600436]  
const p2 = [-3.0064409724419474, 43.37844321062219]  
const p3 = [-2.819887699757718, 43.23977944012737]  
const p4 = [-2.9441081367002004, 43.19792388375848]
```

```
db.recursos.find({geometry:{$geoWithin:{$geometry:{type: "Polygon",coordinates:[[p1,p2,p3,p4,p1]]}}},  
{"properties.documentname":1}).pretty()
```

Consulta 3:

Cuantos recursos hay a 10 km a la redonda de mi vivienda habitual.

```
db.recursos.find({geometry:{$geoWithin: {$centerSphere: [[-3.0326453,43.3233198], 10/6378.1]}}}).count()
```

