Proyecto MongoDB

Francisco Javier Doblado Alonso
1º ASIR
Base de Datos

Sumario

1ª Parte

1. Debes seleccionar un fichero json que incluya todos los tipos de datos soportados en MongoDB.

Ya lo subí al redmine el mongo utilizado.

2. Con la utilidad mongoimport introduce los documentos correspondientes a esa colección.

```
usuario@debian:~$ mongoimport --db movies --collection movies --file /home/usuario/movies.json --jsonArray
2024-05-05T13:26:48.171+0200    connected to: mongodb://localhost/
2024-05-05T13:26:48.188+0200    13 document(s) imported successfully. 0 document(s) failed to import.
```

3. Inserta varios documentos utilizando los dos métodos de inserción de MongoDB

```
movies> db.movies.insertMany([
...     {
...         "title": "The Dark Knight",
...         "year": 2008,
...         "director": "Christopher Nolan",
...         "genre": ["Action", "Crime", "Drama"],
...         "rating": 9.0
...     },
...     {
...         "title": "Interstellar",
...         "year": 2014,
...         "director": "Christopher Nolan",
...         "genre": ["Adventure", "Drama", "Sci-Fi"],
...         "rating": 8.6
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66376f821bd18a70b4ef7cea'),
    '1': ObjectId('66376f821bd18a70b4ef7ceb')
  }
}
```

```
movies> db.movies.insertOne({
...     "title": "Inception",
...     "year": 2010,
...     "director": "Christopher Nolan",
...     "genre": ["Action", "Adventure", "Sci-Fi"],
...     "rating": 8.8
... })
{
  acknowledged: true,
  insertedId: ObjectId('66376f771bd18a70b4ef7ce9')
}
```

4. Elimina varios documentos utilizando los dos métodos de eliminación de MongoDB

```
movies> db.movies.deleteOne({ "title": "Inception" })
{ acknowledged: true, deletedCount: 1 }
movies> db.movies.deleteMany({ "director": "Christopher Nolan" })
{ acknowledged: true, deletedCount: 2 }
```

5. Actualiza varios documentos utilizando los tres métodos de eliminación de MongoDB

```
movies> db.movies.updateOne( { "title": "The Dark Knight" }, /* Filtro*/ { $set: { "rating": 9.2 } } /* Actualización*/ )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

```
movies> db.movies.updateMany(
...     { "director": "Christopher Nolan" }, // Filtro
...     { $set: { "rating": 8.5 } } // Actualización
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

```
movies> db.movies.replaceOne(
...     { "title": "The Dark Knight" }, // Filtro
...     {
...         "title": "The Dark Knight Rises", // Nuevos datos
...         "year": 2012,
...         "director": "Christopher Nolan",
...         "genre": ["Action", "Crime", "Drama"],
...         "rating": 8.9
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

6. Consultas:

1. Al menos incluye 5 consultas de datos simples

Consultar todos los datos

```
movies> db.movies.find({})
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
    duration: 'PT120M',
    releaseDate: '1959-09-28',
    originalTitle: 'Some Like It Hot',
    storyline: "When two Chicago musicians, Joe and Jerry, witness th
to get a gig out of town but the only job they know of is in an all-g
inly enjoy being around the girls, especially Sugar Kane Kowalczyk wh
 ensues as the two men try to keep their true identities hidden and S
      'garykmcd',
    actors: [ 'Marilyn Monroe', 'Tony Curtis', 'Jack Lemmon' ],
    recorded: { country: 'EEUU', city: 'Chicago' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV5B
    awards: [
      { name: 'Oscar', category: 'Best Picture', year: 1960 },
      { name: 'Golden Globe', category: 'Best Actor', year: 1960 }
    ]
  },
```

Buscar películas ganadoras del oscar

```
movies> db.movies.find({ "awards.name": "Oscar" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
    duration: 'PT120M',
    releaseDate: '1959-09-28',
    originalTitle: 'Some Like It Hot',
    storyline: "When two Chicago musicians, Joe and Jerry, witness
to get a gig out of town but the only job they know of is in an all
inly enjoy being around the girls, especially Sugar Kane Kowalczyk
 ensues as the two men try to keep their true identities hidden and
      'garykmcd',
    actors: [ 'Marilyn Monroe', 'Tony Curtis', 'Jack Lemmon' ],
    recorded: { country: 'EEUU', city: 'Chicago' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV
    awards: [
      { name: 'Oscar', category: 'Best Picture', year: 1960 },
      { name: 'Golden Globe', category: 'Best Actor', year: 1960 }
```

Buscar peliculas con una duración superior a 2 horas

```
movies> db.movies.find({ duration: { $gt: 'PT120M' } })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e1'),
    Object_ID: 'Sierra_Madres_skatt_1948',
    title: 'Sierra Madres skatt',
    year: '1948',
    genres: [ 'Adventure', 'Drama', 'Western' ],
    ratings: [
      8, 9,  7, 6,  9, 3, 8, 6, 9,
      5, 5,  6, 7, 10, 5, 1, 3, 1,
      3, 9, 10, 8,  3, 7, 9, 7, 3,
      7, 5,  9
    ],
    duration: 'PT126M',
    releaseDate: '1948-04-29',
    originalTitle: 'The Treasure of the Sierra Madre',
    storyline: 'Fred C. Dobbs and Bob Curtin, both down on their luck in Tampico,
tral Mexico. Through enormous difficulties, they eventually succeed in finding go
' +
      'Jim Beaver <jumblejim@prodigy.net>',
    actors: [ 'Humphrey Bogart', 'Walter Huston', 'Tim Holt' ],
    recorded: { country: 'EEUU', city: 'San Francisco' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV5BMTQ4MzUzOTYw
    awards: [ { name: 'Academy Award', category: 'Best Picture', year: 1949 } ]
```

Buscar películas estrenadas después del 1990

```
movies> db.movies.find({ releaseDate: { $gt: '1990-01-01' } })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e2'),
    Object_ID: 'De_skoningslösa_1992',
    title: 'De skoningslösa',
    year: '1992',
    genres: [ 'Drama', 'Western' ],
    ratings: [
       6, 7, 2, 9, 10, 4, 4, 10,  5,
       9, 9, 2, 3,  3, 1, 8,  3, 10,
       2, 5, 1, 7,  8, 7, 3, 10,  4,
      10, 9, 3
    ],
    duration: 'PT131M',
    releaseDate: '1992-09-11',
    originalTitle: 'Unforgiven',
    storyline: "The town of Big Whisky is full of normal people trying t
es just try to get by.Then a couple of cowboys cut up a whore. Dissatisf
', and aging killer William Munny. Munny reformed for his young wife, an
d partner Ned, saddles his ornery nag, and rides off to kill one more ti
      'Charlie Ness',
    actors: [ 'Clint Eastwood', 'Gene Hackman', 'Morgan Freeman' ],
    recorded: { country: 'EEUU', city: '' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV5BODM
    awards: [ { name: 'BAFTA', category: 'Best Film', year: 1993 } ]
  }
```

Buscar películas con el genero Drama

```
movies> db.movies.find({ genres: "Drama" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e1'),
    Object_ID: 'Sierra_Madres_skatt_1948',
    title: 'Sierra Madres skatt',
    year: '1948',
    genres: [ 'Adventure', 'Drama', 'Western' ],
    ratings: [
      8, 9,  7, 6,  9, 3, 8, 6, 9,
      5, 5,  6, 7, 10, 5, 1, 3, 1,
      3, 9, 10, 8,  3, 7, 9, 7, 3,
      7, 5,  9
    ],
```

2. Al menos 3 consultas con arrays

Buscar películas con una calificación de al menos 8

```
movies> db.movies.find({ ratings: { $elemMatch: { $gte: 8 } } })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
```

Buscar películas protagonizadas por un actor específico

```
movies> db.movies.find({ actors: "Robert De Niro" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e4'),
    Object_ID: 'Tjuren_från_Bronx_1980',
    title: 'Tjuren från Bronx',
    year: '1980',
    genres: [ 'Biography', 'Drama', 'Sport' ],
    ratings: [
      3, 10, 8, 1, 5,  4, 9,  5,  9,
      2,  8, 3, 2, 6,  3, 5, 10,  2,
      1,  9, 3, 9, 4, 10, 3,  3, 10,
      6,  9, 3
    ],
    duration: 'PT129M',
    releaseDate: '1981-03-20',
    originalTitle: 'Raging Bull',
    storyline: "When Jake LaMotta steps into a boxing ring and ob
any moment. Though LaMotta wants his family's love, something alw
fe, he winds up in the ring alone.                Written by\n" +
      'alfiehitchie',
    actors: [ 'Robert De Niro', 'Cathy Moriarty', 'Joe Pesci' ],
```

Buscar películas que hayan ganado mas de un premio

```
movies> db.movies.find({ "awards": { $exists: true, $not: { $size: 1 } } })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
    duration: 'PT120M',
    releaseDate: '1959-09-28',
    originalTitle: 'Some Like It Hot',
    storyline: "When two Chicago musicians, Joe and Jerry, witness the the
to get a gig out of town but the only job they know of is in an all-girl b
inly enjoy being around the girls, especially Sugar Kane Kowalczyk who sin
 ensues as the two men try to keep their true identities hidden and Spats
      'garykmcd',
    actors: [ 'Marilyn Monroe', 'Tony Curtis', 'Jack Lemmon' ],
    recorded: { country: 'EEUU', city: 'Chicago' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV5BNzAy(
    awards: [
      { name: 'Oscar', category: 'Best Picture', year: 1960 },
      { name: 'Golden Globe', category: 'Best Actor', year: 1960 }
    ],
```

3. Al menos 3 consultas con documentos embebidos

Buscar películas con actores que hayan ganado premios

```
movies> db.movies.find({ "actors": { $in: ["Robert De Niro", "Morgan Freeman"] } })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e2'),
    Object_ID: 'De_skoningslösa_1992',
    title: 'De skoningslösa',
    year: '1992',
    genres: [ 'Drama', 'Western' ],
    ratings: [
       6, 7, 2, 9, 10, 4, 4, 10,  5,
       9, 9, 2, 3,  3, 1, 8,  3, 10,
       2, 5, 1, 7,  8, 7, 3, 10,  4,
      10, 9, 3
    ],
    duration: 'PT131M',
    releaseDate: '1992-09-11',
    originalTitle: 'Unforgiven',
    storyline: "The town of Big Whisky is full of normal people trying to lead quie
es just try to get by.Then a couple of cowboys cut up a whore. Dissatisfied with Bi
', and aging killer William Munny. Munny reformed for his young wife, and has been
d partner Ned, saddles his ornery nag, and rides off to kill one more time, blurrin
      'Charlie Ness',
    actors: [ 'Clint Eastwood', 'Gene Hackman', 'Morgan Freeman' ],
    recorded: { country: 'EEUU', city: '' }
```

Buscar película con premios oscar

```
movies> db.movies.find({ "awards.name": "Oscar" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
    duration: 'PT120M',
    releaseDate: '1959-09-28',
    originalTitle: 'Some Like It Hot',
    storyline: "When two Chicago musicians, Joe and
to get a gig out of town but the only job they know
inly enjoy being around the girls, especially Sugar
 ensues as the two men try to keep their true ident
      'garykmcd',
    actors: [ 'Marilyn Monroe', 'Tony Curtis', 'Jac
    recorded: { country: 'EEUU', city: 'Chicago' },
    posterurl: 'https://images-na.ssl-images-amazon
    awards: [
      { name: 'Oscar', category: 'Best Picture', ye
```

Buscar películas con premios de Mejor Película"

```
movies> db.movies.find({ "awards.category": "Best Picture" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7e0'),
    Object_ID: 'I_hetaste_laget_1959',
    title: 'I hetaste laget',
    year: '1959',
    genres: [ 'Comedy', 'Romance' ],
    ratings: [
      5, 5, 7, 5, 2, 7, 2, 8,  4,
      6, 2, 1, 4, 5, 2, 8, 2,  1,
      1, 8, 5, 8, 9, 8, 4, 4, 10,
      7, 6, 2
    ],
    duration: 'PT120M',
    releaseDate: '1959-09-28',
    originalTitle: 'Some Like It Hot',
    storyline: "When two Chicago musicians, Joe and Jerry, witne
to get a gig out of town but the only job they know of is in an
inly enjoy being around the girls, especially Sugar Kane Kowalcz
 ensues as the two men try to keep their true identities hidden
      'garykmcd',
    actors: [ 'Marilyn Monroe', 'Tony Curtis', 'Jack Lemmon' ],
    recorded: { country: 'EEUU', city: 'Chicago' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M
    awards: [
      { name: 'Oscar', category: 'Best Picture', year: 1960 },
```

4. Al menos 1 consulta de agrupación

Consulta de películas agrupadas por año de lanzamiento

```
movies> db.movies.aggregate([
...     {
...         $group: {
...             _id: "$year",
...             count: { $sum: 1 }
...         }
...     },
...     { $sort: { _id: 1 } }
... ])
[
  { _id: '1948', count: 1 },
  { _id: '1949', count: 2 },
  { _id: '1950', count: 1 },
  { _id: '1959', count: 1 },
  { _id: '1961', count: 1 },
  { _id: '1980', count: 2 },
  { _id: '1988', count: 1 },
  { _id: '1992', count: 1 },
  { _id: '2004', count: 2 },
  { _id: '2005', count: 1 }
]
```

Recuento de películas por género

```
movies> db.movies.aggregate([
...     {
...         $unwind: "$genres" // Separa los documentos por cada género
...     },
...     {
...         $group: {
...             _id: "$genres",
...             count: { $sum: 1 } // Calcula el recuento de películas por género
...         }
...     },
...     {
...         $sort: { count: -1 } // Ordena los resultados por el recuento en orden descendente
... }
... ])
[
  { _id: 'Drama', count: 8 },
  { _id: 'Thriller', count: 4 },
  { _id: 'Biography', count: 4 },
  { _id: 'Action', count: 3 },
  { _id: 'Film-Noir', count: 2 },
  { _id: 'Adventure', count: 2 },
  { _id: 'Western', count: 2 },
  { _id: 'Mystery', count: 2 },
  { _id: 'History', count: 2 },
  { _id: 'Sport', count: 2 },
  { _id: 'Romance', count: 1 },
  { _id: 'Comedy', count: 1 }
]
```

2ª Parte

Debes realizar un programa python que se conecte a la colección creada en la 1ª parte y mediante un menú permita insercción, eliminación, modificación y al menos una consulta de cada una de los distintos tipos.

Primero debemos de instalar la librería pymongo con un pip install pymongo

movies.py > update_movie

```python
from pymongo import MongoClient

client = MongoClient('localhost', 27017)
db = client.movies

def insert_movie():
    title = input("Ingrese el titulo de la pelicula: ")
    year = input("Ingrese el año de lanzamiento: ")
    genre = input("Ingrese el género de la pelicula: ")
    db.movies.insert_one({"title": title, "year": year, "genre": genre})
    print("Película insertada correctamente.")

def delete_movie():
    title = input("Ingrese el titulo de la película que desea eliminar: ")
    result = db.movies.delete_one({"title": title})
    if result.deleted_count == 1:
        print("Película eliminada correctamente.")
    else:
        print("No se encontró ninguna película con ese titulo.")

def update_movie():
    title = input("Ingrese el titulo de la película que desea actualizar: ")
    new_title = input("Ingrese el nuevo titulo de la pelicula: ")
    new_year = input("Ingrese el nuevo año de lanzamiento: ")
    new_genre = input("Ingrese el nuevo género de la pelicula: ")
    result = db.movies.update_one({"title": title}, {"$set": {"title": new_title, "year": new_year, "genre":
    if result.modified_count == 1:
        print("Película actualizada correctamente.")
    else:
        print("No se encontró ninguna película con ese titulo.")

def consultar_por_año():
    años = db.movies.distinct("year")
    print("Años disponibles:")
    for año in años:
        print(año)

    año = input("Introduce el año de la película que deseas consultar: ")
    peliculas = list(db.movies.find({"year": año}, {"_id": 0, "title": 1}))
    if not peliculas:
        print("No se encontraron películas para el año especificado.")
    else:
        print(f"Películas lanzadas en {año}:")
        for pelicula in peliculas:
            print(pelicula['title'])

def consulta_por_actor():
    actores = db.movies.distinct("actors")
    print("Actores disponibles:")
    for actor in actores:
        print(actor)
```

```python
def consulta_por_premio():
    premios = db.movies.distinct("awards.name")
    print("Premios disponibles:")
    for premio in premios:
        print(premio)

    premio = input("Introduce el nombre del premio que deseas buscar: ")
    peliculas = list(db.movies.find({"awards.name": premio}, {"_id": 0, "title": 1}))
    if not peliculas:
        print("No se encontraron películas con ese premio.")
    else:
        print(f"Películas que han ganado el premio {premio}:")
        for pelicula in peliculas:
            print(pelicula['title'])

def consulta_agrupacion():
    pipeline = [
        {"$unwind": "$genres"},
        {"$group": {"_id": "$genres", "count": {"$sum": 1}}}
    ]
    resultados = db.movies.aggregate(pipeline)
    print("Número de películas por género:")
    for resultado in resultados:
        print(f"{resultado['_id']}: {resultado['count']}")

while True:
    print("\nMenú:")
    print("1. Insertar película")
    print("2. Eliminar película")
    print("3. Actualizar película")
    print("4. Buscar película por año")
    print("5. Consultar películas por actor")
    print("6. Consultar películas por premio")
    print("7. Consulta de agrupación por género")
    print("8. Salir")

    choice = input("Seleccione una opción: ")

    if choice == "1":
        insert_movie()
    elif choice == "2":
        delete_movie()
    elif choice == "3":
        update_movie()
    elif choice == "4":
        consultar_por_año()
    elif choice == "5":
        consulta_por_actor()
    elif choice == "6":
        consulta_por_premio()
    elif choice == "7":
```

insertar película

```
(gns3env) usuario@debian:~$ /home/usuario/gns3env/bin/python /home/usuario/movies.py

Menú:
1. Insertar pelicula
2. Eliminar pelicula
3. Actualizar pelicula
4. Buscar pelicula por año
5. Consultar peliculas por actor
6. Consultar peliculas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 1
Ingrese el titulo de la pelicula: gray zone
Ingrese el año de lanzamiento: 2024
Ingrese el género de la película: Drama, Terror
Pelicula insertada correctamente.
```

```
movies> db.movies.find({"title": "gray zone"})
[
  {
    _id: ObjectId('663bc9ad83dd94d78737700f'),
    title: 'gray zone',
    year: '2024',
    genre: 'Drama, Terror'
  }
]
```

Eliminar película

```
Menú:
1. Insertar pelicula
2. Eliminar pelicula
3. Actualizar pelicula
4. Buscar pelicula por año
5. Consultar peliculas por actor
6. Consultar peliculas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 2
Ingrese el titulo de la pelicula que desea eliminar: gray zone
Pelicula eliminada correctamente.
```

```
movies> db.movies.find({"title": "gray zone"})

movies>
```

Actualizar película por año

Ahora creare otra vez la película pepe para actualizarla.

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 3
Ingrese el título de la película que desea actualizar: gray zone
Ingrese el nuevo título de la película: pink zone
Ingrese el nuevo año de lanzamiento: 2025
Ingrese el nuevo género de la película: Drama
Película actualizada correctamente.
```

```
movies> db.movies.find({"title": "gray zone "})
[
  {
    _id: ObjectId('663bcad583dd94d787377010'),
    title: 'gray zone ',
    year: '2024',
    genre: 'Action'
  }
]
movies> db.movies.find({"title": "pink zone"})
[
  {
    _id: ObjectId('663bcad583dd94d787377010'),
    title: 'pink zone',
    year: '2025',
    genre: 'Drama'
  }
]
```

Buscar película por año

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 4
Años disponibles:
1948
1949
1950
1959
1961
1980
1988
1992
2004
2005
2006
2025
2035
3045
Introduce el año de la película que deseas consultar: 1980
Películas lanzadas en 1980:
Tjuren från Bronx
```

Consultar películas por actor

```
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 5
Actores disponibles:
Alan Rickman
Alexandra Maria Lara
Alida Valli
Anne Baxter
Bette Davis
Bonnie Bedelia
Bruce Willis
Bruno Ganz
Cathy Moriarty
Christian Bale
Clint Eastwood
Eijirô Tôno
Gene Hackman
George Sanders
Humphrey Bogart
Jack Lemmon
Joe Pesci
Joseph Cotten
Ken Watanabe
Marilyn Monroe
Michael Caine
Morgan Freeman
Orson Welles
Robert De Niro
Tatsuya Nakadai
Tim Holt
Tony Curtis
Toshirô Mifune
Ulrich Matthes
Walter Huston
Introduce el nombre del actor que deseas buscar: Tim Holt
Películas con el actor Tim Holt:
Sierra Madres skatt
```

Consultar películas por premio

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción:
Opción inválida. Por favor, seleccione una opción válida.

Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 6
Premios disponibles:
Academy Award
BAFTA
Cannes Film Festival
Golden Globe
Goya
Oscar
Introduce el nombre del premio que deseas buscar: Goya
Películas que han ganado el premio Goya:
Batman Begins
```

Consulta de agrupación por género

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Consultar películas por actor
6. Consultar películas por premio
7. Consulta de agrupación por género
8. Salir
Seleccione una opción: 7
Número de películas por género:
Romance: 1
Drama: 8
Adventure: 2
Sport: 2
Comedy: 1
Western: 2
Thriller: 4
Action: 3
Film-Noir: 2
Mystery: 2
History: 2
Biography: 4
```

3ª Parte

1. Debes seleccionar al menos cuatro tablas de tu proyecto que incluyan una relación N:M yuna 1:N con el mayor número de tipos de datos soportados en MongoDB.

Jugadores: Representa a los jugadores de fútbol. Hay una relación 1:N entre los jugadores y los clubes a los que pertenecen, y una relación N:M entre los jugadores y los contratos.

Club: Representa a los clubes de fútbol. Existe una relación 1:N entre los clubes y los entrenadores que los dirigen, y una relación N:M entre los clubes y los jugadores.

Entrenador: Representa a los entrenadores de fútbol. Hay una relación 1:N entre los entrenadores y los clubes a los que pertenecen, y una relación N:M entre los entrenadores y los clubes.

Contrato: Representa los contratos entre jugadores y clubes. Existe una relación 1:N entre los contratos y los jugadores, y una relación N:M entre los contratos y los clubes.

Tabla Jugadores:

DNI	Nombre	Apellidos	Fecha_Nacimiento	Direccion	Telefono	Nacionalidad	Talla	Peso	Clubes
12334978A	Lionel	Messi	1987-06-24	Barcelona	1234567890	Argentina	M	68	FC Barcelo PSG
87654321B	Cristiano	Ronaldo	1985-02-05	Turin	9876543210	Portugal	L	83	Juventus
45678901C	Neymar	Jr.	1992-02-05	Paris	5678901234	Brasil	L	68	Paris Saint-Germain

Tabla Club:

Nombre	Palmares	Ciudad	Division	Internacionalidad	DNI_Entrenador
FC Barcelona	25	Barcelona	1	1	12345678A
Juventus	37	Turin	1	1	87654321B
Paris Saint-Germain	11	Paris	1	1	45678901C

Tabla Entrenador:

DNI	Nombre
12315278A	Pep Guardiola
87654321B	Zinedine Zidane
45678901C	Jurgen Klopp

Tabla Contrato:

Fecha	Duracion	Salario	Clausula	DNI_Jugador	Nombre_Club
2024-01-15	2029-01-15	1200000	1000	12334978A	FC Barcelona
2024-01-15	2028-01-15	800000	8000	87654321B	Juventus

2. Debes pasar de SQL a MongoDB.

Para pasarlo de sql a MongoDB aprovechando los conocimiento del ejercicio anterior lo que hice fue con un programa de python pasarlo directamente a la base de dato de mi mongo llamada fut1;

crearfutbol.py > ...

```python
from datetime import datetime
from pymongo import MongoClient

client = MongoClient('localhost', 27017)
db = client.fut1


jugadores = [
    {"DNI": "12334978A", "Nombre": "Lionel", "Apellidos": "Messi", "Fecha_Nacimiento": datetime(1987, 6, 24), "Direccion": "Barce
    {"DNI": "87654321B", "Nombre": "Cristiano", "Apellidos": "Ronaldo", "Fecha_Nacimiento": datetime(1985, 2, 5), "Direccion": "T
    {"DNI": "45678901C", "Nombre": "Neymar", "Apellidos": "Jr.", "Fecha_Nacimiento": datetime(1992, 2, 5), "Direccion": "Paris",
]
db.Jugadores.insert_many(jugadores)


clubes = [
    {"Nombre": "FC Barcelona", "Palmares": "25", "Ciudad": "Barcelona", "Division": 1, "Internacionalidad": 1, "Entrenadores": ["
    {"Nombre": "Juventus", "Palmares": "37", "Ciudad": "Turin", "Division": 1, "Internacionalidad": 1, "Entrenadores": ["8765432
]


entrenadores = [
    {"DNI": "12315278A", "Nombre": "Pep Guardiola"},
    {"DNI": "87654321B", "Nombre": "Zinedine Zidane"},
    {"DNI": "45678901C", "Nombre": "Jurgen Klopp"}
]
db.Entrenador.insert_many(entrenadores)


contratos = [
    {"Fecha": datetime(2024, 1, 15), "Duracion": datetime(2029, 1, 15), "Salario": 1200000, "Clausula": 1000, "DNI_Jugador": "123
    {"Fecha": datetime(2024, 1, 15), "Duracion": datetime(2028, 1, 15), "Salario": 800000, "Clausula": 8000, "DNI_Jugador": "876
]
db.Contrato.insert_many(contratos)
```

```
(gns3env) usuario@debian:~$ /home/usuario/gns3env/bin/python /home/usuario/crearfutbol.py
(gns3env) usuario@debian:~$
```

Luego reviso que se halla realizado desde mongo

```
test> use fut1
switched to db fut1
fut1> show tables
Club
Contrato
Entrenador
Jugadores
fut1>
```

3. Inserta varios documentos utilizando los dos métodos de inserción de MongoDB

```
fut1> db.Club.insertMany([
...     {
...         "Nombre": "Real Madrid",
...         "Palmares": "34",
...         "Ciudad": "Madrid",
...         "Division": 1,
...         "Internacionalidad": 1,
...         "DNI_Entrenador": "87654321D"
...     },
...     {
...         "Nombre": "Manchester United",
...         "Palmares": "66",
...         "Ciudad": "Manchester",
...         "Division": 1,
...         "Internacionalidad": 1,
...         "DNI_Entrenador": "98765432E"
...     }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('663bd953602ef17c90184186'),
    '1': ObjectId('663bd953602ef17c90184187')
  }
}
```

```
fut1> db.Entrenador.insertMany([
...     {
...         "DNI": "87654321D",
...         "Nombre": "Carlo Ancelotti"
...     },
...     {
...         "DNI": "98765432E",
...         "Nombre": "Sir Alex Ferguson"
...     }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('663bd980602ef17c90184188'),
    '1': ObjectId('663bd980602ef17c90184189')
  }
}
```

4. Elimina varios documentos utilizando los dos métodos de eliminación de MongoDB

```
fut1> db.Club.deleteMany({
...     "Nombre": { $in: ["Real Madrid", "Manchester United"] }
... });
{ acknowledged: true, deletedCount: 4 }
fut1>
```

```
fut1> db.Entrenador.remove({
...     "Nombre": { $in: ["Carlos", "Sir Alex Ferguson"] }
... });
{ acknowledged: true, deletedCount: 1 }
```

5. Actualiza varios documentos utilizando los tres métodos de actualización de MongoDB

```
fut1> db.Entrenador.updateMany(
...     { "Nombre": "Pep Guardiola" },
...     { $set: { "Nombre": "Pep Guardiola Jr" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
fut1> db.Entrenador.update(
...     { "Nombre": "Zinedine Zidane" },
...     { $set: { "Nombre": "Zizou" } },
...     { multi: true }
... );
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
fut1> const operations = [
...     {
...         updateMany: {
...             filter: { "Nombre": "Jurgen Klopp" },
...             update: { $set: { "Nombre": "Jurgen Klopp Sr" } }
...         }
...     }
... ];

fut1>

fut1> db.Entrenador.bulkWrite(operations);
{
  acknowledged: true,
  insertedCount: 0,
  insertedIds: {},
  matchedCount: 1,
  modifiedCount: 1,
  deletedCount: 0,
  upsertedCount: 0,
  upsertedIds: {}
}
```

6. Consultas:

1. Al menos incluye 5 consultas de datos simples

Consulta para encontrar todos los documentos en una colección

```
fut1> db.Entrenador.find({});
[
  {
    _id: ObjectId('663bd255f8f67c732a494574'),
    DNI: '12315278A',
    Nombre: 'Pep Guardiola Jr'
  },
  {
    _id: ObjectId('663bd255f8f67c732a494575'),
    DNI: '87654321B',
    Nombre: 'Zizou'
  },
  {
    _id: ObjectId('663bd255f8f67c732a494576'),
    DNI: '45678901C',
    Nombre: 'Jurgen Klopp Sr'
  },
  {
    _id: ObjectId('663bd980602ef17c90184188'),
    DNI: '87654321D',
    Nombre: 'Carlo Ancelotti'
  }
]
```

Consulta para encontrar documentos con un campo específico igual a un valor dado

```
fut1> db.Jugadores.find({ Nacionalidad: 'Argentina' });
[
  {
    _id: ObjectId('663bd255f8f67c732a49456f'),
    DNI: '12334978A',
    Nombre: 'Lionel',
    Apellidos: 'Messi',
    Fecha_Nacimiento: ISODate('1987-06-24T00:00:00.000Z'),
    Direccion: 'Barcelona',
    Telefono: '1234567890',
    Nacionalidad: 'Argentina',
    Talla: 'M',
    Peso: '68',
    Clubes: [ 'FC Barcelona', 'Paris Saint-Germain' ]
  }
]
```

Consulta para encontrar documentos con un campo que contenga una cadena específica

```
fut1> db.Club.find({ Nombre: /Barcelona/ });
[
  {
    _id: ObjectId('663bd255f8f67c732a494572'),
    Nombre: 'FC Barcelona',
    Palmares: '25',
    Ciudad: 'Barcelona',
    Division: 1,
    Internacionalidad: 1,
    Entrenadores: [ '12315278A', '45678901C' ]
  }
]
```

Consulta para encontrar documentos con un campo que sea mayor que un valor dado

```
fut1> db.Jugadores.find({ Peso: { $gt: '70' } });
[
  {
    _id: ObjectId('663bd255f8f67c732a494570'),
    DNI: '87654321B',
    Nombre: 'Cristiano',
    Apellidos: 'Ronaldo',
    Fecha_Nacimiento: ISODate('1985-02-05T00:00:00.000Z'),
    Direccion: 'Turin',
    Telefono: '9876543210',
    Nacionalidad: 'Portugal',
    Talla: 'L',
    Peso: '83',
    Clubes: [ 'Juventus' ]
  }
]
```

Consulta para encontrar documentos que cumplan múltiples condiciones

```
fut1> db.Jugadores.find({ Nacionalidad: 'Brasil', Talla: 'L' });
[
  {
    _id: ObjectId('663bd255f8f67c732a494571'),
    DNI: '45678901C',
    Nombre: 'Neymar',
    Apellidos: 'Jr.',
    Fecha_Nacimiento: ISODate('1992-02-05T00:00:00.000Z'),
    Direccion: 'Paris',
    Telefono: '5678901234',
    Nacionalidad: 'Brasil',
    Talla: 'L',
    Peso: '68',
    Clubes: [ 'FC Barcelona', 'Paris Saint-Germain' ]
  }
]
```

2. Al menos 3 consultas con arrays

Consulta para encontrar todos los clubes que tienen más de un entrenador

```
fut1> db.Club.find({ "Entrenadores.1": { $exists: true } });
[
  {
    _id: ObjectId('663bd255f8f67c732a494572'),
    Nombre: 'FC Barcelona',
    Palmares: '25',
    Ciudad: 'Barcelona',
    Division: 1,
    Internacionalidad: 1,
    Entrenadores: [ '12315278A', '45678901C' ],
    jugadores: [ 'Lionel Messi', 'Neymar' ]
  }
]
```

Consulta para encontrar todos los clubes que tienen a Lionel Messi como jugador

```
fut1> db.Club.find({ jugadores: 'Lionel Messi' });
[
  {
    _id: ObjectId('663bd255f8f67c732a494572'),
    Nombre: 'FC Barcelona',
    Palmares: '25',
    Ciudad: 'Barcelona',
    Division: 1,
    Internacionalidad: 1,
    Entrenadores: [ '12315278A', '45678901C' ],
    jugadores: [ 'Lionel Messi', 'Neymar' ]
  }
]
```

Consulta para encontrar todos los contratos que tienen un salario mayor a 1 millón

```
fut1> db.Contrato.find({ Salario: { $gt: 1000000 } });
[
  {
    _id: ObjectId('663bd255f8f67c732a494577'),
    Fecha: ISODate('2024-01-15T00:00:00.000Z'),
    Duracion: ISODate('2029-01-15T00:00:00.000Z'),
    Salario: 1200000,
    Clausula: 1000,
    DNI_Jugador: '12334978A',
    Nombre_Club: 'FC Barcelona'
  }
]
```

3. Al menos 3 consultas con documentos embebido

Consulta para encontrar todos los clubes que tienen más de un entrenador

```
fut1> db.Jugadores.find({ Contrato: { $exists: true } });
[
  {
    _id: ObjectId('663be407e4a7cbd01165cee8'),
    DNI: '12334978A',
    Nombre: 'Lionel',
    Apellidos: 'Messi',
    Contrato: {
      Fecha: ISODate('2024-01-15T00:00:00.000Z'),
      Duracion: ISODate('2029-01-15T00:00:00.000Z'),
      Salario: 1200000,
      Clausula: 1000,
      Nombre_Club: 'FC Barcelona'
    }
  },
  {
    _id: ObjectId('663be407e4a7cbd01165cee9'),
    DNI: '87654321B',
    Nombre: 'Cristiano',
    Apellidos: 'Ronaldo',
    Contrato: {
      Fecha: ISODate('2024-01-15T00:00:00.000Z'),
      Duracion: ISODate('2028-01-15T00:00:00.000Z'),
      Salario: 800000,
      Clausula: 8000,
      Nombre_Club: 'Juventus'
    }
  }
]
```

Consulta para encontrar todos los entrenadores con información de club embebida:

```
fut1> db.Entrenador.find({ "Club": { $exists: true } });
[
  {
    _id: ObjectId('663bd255f8f67c732a494574'),
    DNI: '12315278A',
    Nombre: 'Pep Guardiola Jr',
    Club: { Nombre: 'FC Barcelona', Palmares: '25', Ciudad: 'Barcelona' }
  },
  {
    _id: ObjectId('663bd255f8f67c732a494575'),
    DNI: '87654321B',
    Nombre: 'Zizou',
    Club: { Nombre: 'FC Barcelona', Palmares: '25', Ciudad: 'Barcelona' }
  },
  {
    _id: ObjectId('663bd255f8f67c732a494576'),
    DNI: '45678901C',
    Nombre: 'Jurgen Klopp Sr',
    Club: { Nombre: 'FC Barcelona', Palmares: '25', Ciudad: 'Barcelona' }
  },
  {
    _id: ObjectId('663bd980602ef17c90184188'),
    DNI: '87654321D',
    Nombre: 'Carlo Ancelotti',
    Club: { Nombre: 'FC Barcelona', Palmares: '25', Ciudad: 'Barcelona' }
  }
]
```

Consulta para encontrar todos los clubes con información de entrenadores embebida

```
fut1> db.Club.find({ "Entrenadores": { $exists: true } });
[
  {
    _id: ObjectId('663bd255f8f67c732a494572'),
    Nombre: 'FC Barcelona',
    Palmares: '25',
    Ciudad: 'Barcelona',
    Division: 1,
    Internacionalidad: 1,
    Entrenadores: [ '12315278A', '45678901C' ],
    jugadores: [ 'Lionel Messi', 'Neymar' ]
  },
  {
    _id: ObjectId('663bd255f8f67c732a494573'),
    Nombre: 'Juventus',
    Palmares: '37',
    Ciudad: 'Turin',
    Division: 1,
    Internacionalidad: 1,
    Entrenadores: [ '87654321B' ],
    jugadores: [ 'Cristiano Ronaldo' ]
  }
]
```

4. Al menos 1 consulta de agrupación

Consulta para agrupar los jugadores por nacionalidad y calcular el promedio de sus pesos:

```
fut1> db.Jugadores.aggregate([
...     {
...         $group: {
...             _id: "$Nacionalidad",
...             promedioPeso: { $avg: { $toDouble: "$Peso" } }
...         }
...     }
... ]);
[
  { _id: 'Brasil', promedioPeso: 68 },
  { _id: null, promedioPeso: null },
  { _id: 'Argentina', promedioPeso: 68 },
  { _id: 'Portugal', promedioPeso: 83 }
]
```