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 quiet
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 r
d partner Ned, saddles his ornery nag, and rides off to kill one more time, blurring
      '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

```python
from pymongo import MongoClient

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

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

def delete_movie():
    title = input("Ingrese el título 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 título.")

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

def find_movie():
    year = input("Ingrese el año de lanzamiento de la película que desea buscar: ")
    movies = db.movies.find({"year": year})
    print("Películas encontradas:")
    for movie in movies:
        print(movie)
```

```python
while True:
    print("\nMenú:")
    print("1. Insertar película")
    print("2. Eliminar película")
    print("3. Actualizar película")
    print("4. Buscar película por año")
    print("5. Salir")

    choice = input("Seleccione una opción: ")

    if choice == "1":
        insert_movie()
    elif choice == "2":
        delete_movie()
    elif choice == "3":
        update_movie()
    elif choice == "4":
        find_movie()
    elif choice == "5":
        print("¡Hasta luego!")
        break
    else:
        print("Opción inválida. Por favor, seleccione una opción válida.")
```

insertar película

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

Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Salir
Seleccione una opción: 1
Ingrese el título de la película: pepe
Ingrese el año de lanzamiento: 2004
Ingrese el género de la película: Drama
Película insertada correctamente.
```

```
movies> db.movies.find({ "title": "pepe" })
[
  {
    _id: ObjectId('66377ed386579e22c1b48ef9'),
    title: 'pepe',
    year: '2004',
    genre: 'drama'
  },
```

Eliminar película

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Salir
Seleccione una opción: 2
Ingrese el título de la película que desea eliminar: pepe
Película eliminada correctamente.
```

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Salir
Seleccione una opción: 2
Ingrese el título de la película que desea eliminar: pepe
No se encontró ninguna película con ese título.
```

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. Salir
Seleccione una opción: 1
Ingrese el título de la película: Pepe
Ingrese el año de lanzamiento: 2004
Ingrese el género de la película: Drama
Película insertada correctamente.
```

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Salir
Seleccione una opción: 3
Ingrese el título de la película que desea actualizar: Pepe
Ingrese el nuevo título de la película: pepe y pepa
Ingrese el nuevo año de lanzamiento: 2008
Ingrese el nuevo género de la película: Terror
Película actualizada correctamente.
```

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

movies> db.movies.find({ "title": "pepe y pepa" })
[
  {
    _id: ObjectId('66378fb1aef61f1062abb3dc'),
    title: 'pepe y pepa',
    year: '2008',
    genre: 'Terror'
  }
]
```

Buscar película por año

```
Menú:
1. Insertar película
2. Eliminar película
3. Actualizar película
4. Buscar película por año
5. Salir
Seleccione una opción: 4
Introduce el año de la película que deseas consultar: 2005
Películas lanzadas en 2005:
Batman Begins
```

```
movies> db.movies.find({ "year": "2005" })
[
  {
    _id: ObjectId('66376cf862c647b27e2ae7ec'),
    Object_ID: 'Batman Begins_2005',
    title: 'Batman Begins',
    year: '2005',
    genres: [ 'Action', 'Adventure' ],
    ratings: [
      2, 5, 2, 8, 1, 2,  9,  5, 10,
      3, 7, 7, 7, 4, 8,  2, 10,  6,
      6, 3, 7, 8, 3, 7, 10,  5,  8,
      2, 9, 4
    ],
    duration: 'PT140M',
    releaseDate: '2005-07-27',
    originalTitle: '',
    storyline: "When his parents are killed, billionaire playboy Bruce Wa
 evil in Gotham City by Ducard, Bruce prevents this plan from getting any
and the corrupt as the icon known as 'Batman'. But it doesn't stay quiet
      'konstantinwe',
    actors: [ 'Christian Bale', 'Michael Caine', 'Ken Watanabe' ],
    recorded: { country: 'EEUU', city: 'Ohio' },
    posterurl: 'https://images-na.ssl-images-amazon.com/images/M/MV5BNTM3
    awards: [ { name: 'Goya', category: 'Best Picture', year: 2004 } ]
  }
]
```

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.

E utilizado las siguientes tablas;

Jugadores: Contiene información sobre los jugadores.

Contrato: Almacena los contratos de los jugadores con los clubes.

Partido: Registra los partidos.

Jugadores_partido: Asocia los jugadores con los partidos y registra su desempeño en cada partido.

2. Debes pasar de SQL a MongoDB.

Insertar jugadores

```
test> db.jugadores.insertMany([
...   {
...     "_id": ObjectId("663796e0f0f6a6f905925bb1"),
...     "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",
...     "Actuaciones_Seleccion": 0,
...     "club": {
...       "$ref": "club",
...       "$id": ObjectId("66379693f0f6a6f905925bae")
...     }
...   },
...   {
...     "_id": ObjectId("663796e0f0f6a6f905925bb2"),
...     "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",
...     "Actuaciones_Seleccion": 0,
...     "club": {
...       "$ref": "club",
...       "$id": ObjectId("66379693f0f6a6f905925baf")
...     }
```

insertar contrato

```
test> db.contrato.insertMany([
...   {
...     "_id": ObjectId("663796e0f0f6a6f905925bba"),
...     "Fecha": ISODate("2024-01-15T00:00:00.000Z"),
...     "Duracion": ISODate("2029-01-15T00:00:00.000Z"),
...     "Salario": 1200000,
...     "Clausula": 1000,
...     "jugador": {
...       "$ref": "jugadores",
...       "$id": ObjectId("663796e0f0f6a6f905925bb1")
...     },
...     "club": {
...       "$ref": "club",
...       "$id": ObjectId("66379693f0f6a6f905925bae")
...     }
...   },
...   // Aquí podrías añadir más documentos de contrato si fuera necesario
... ]);
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('663796e0f0f6a6f905925bba') }
}
```

insertar los partidos

```
test> db.partidos.insertMany([
...   {
...     "_id": 1003,
...     "Fecha": ISODate("2024-01-15T00:00:00.000Z"),
...     "Alineacion_inicial": "Alineacion inicial para el partido 1001"
...   },
...   // Aquí podrías añadir más documentos de partidos si fuera necesario
... ]);
{ acknowledged: true, insertedIds: { '0': 1003 } }
test>
```

insertar los jugadores_partidos

```
test> db.jugadores_partidos.insertMany([
...   {
...     "ID_Partido": 1003,
...     "Posicion": "Delantero",
...     "FaltasR": 2,
...     "FaltasC": 1,
...     "BalonesP": 20,
...     "BalonesR": 5,
...     "Amolestaciones": 1,
...     "Minuto_jugados": 90,
...     "jugador": {
...       "$ref": "jugadores",
...       "$id": ObjectId("663796e0f0f6a6f905925bb1")
...     }
...   },
...   // Aquí podrías añadir más documentos de jugadores_partidos si fuera necesario
... ]);
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6637a766e9c50a354579f151') }
}
```

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

```
test> db.jugadores.insertOne({
...   "DNI": "09876543X",
...   "Nombre": "Neymar",
...   "Apellidos": "da Silva Santos",
...   "Fecha_Nacimiento": ISODate("1992-02-05T00:00:00.000Z"),
...   "Direccion": "Paris",
...   "Telefono": "123456789",
...   "Nacionalidad": "Brasil",
...   "Talla": "M",
...   "Peso": "68",
...   "Actuaciones_Seleccion": 0,
...   "club": {
...     "$ref": "club",
...     "$id": ObjectId("66379693f0f6a6f905925bb0")
...   }
... });
{
  acknowledged: true,
  insertedId: ObjectId('6637a9e1e9c50a354579f152')
}
```

```
test> db.jugadores.insertMany([
...   {
...     "DNI": "01234567Y",
...     "Nombre": "Luis",
...     "Apellidos": "Suárez",
...     "Fecha_Nacimiento": ISODate("1987-01-24T00:00:00.000Z"),
...     "Direccion": "Madrid",
...     "Telefono": "987654321",
...     "Nacionalidad": "Uruguay",
...     "Talla": "L",
...     "Peso": "80",
...     "Actuaciones_Seleccion": 0,
...     "club": {
...       "$ref": "club",
...       "$id": ObjectId("66379693f0f6a6f905925bb0")
...     }
...   },
...   {
...     "DNI": "23456789Z",
...     "Nombre": "Gareth",
...     "Apellidos": "Bale",
...     "Fecha_Nacimiento": ISODate("1989-07-16T00:00:00.000Z"),
...     "Direccion": "Madrid",
...     "Telefono": "987654321",
...     "Nacionalidad": "Gales",
...     "Talla": "L",
...     "Peso": "84",
...     "Actuaciones_Seleccion": 0,
...     "club": {
...       "$ref": "club",
...       "$id": ObjectId("66379693f0f6a6f905925baf")
...     }
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6637a9f6e9c50a354579f153'),
    '1': ObjectId('6637a9f6e9c50a354579f154')
```

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

```
test> db.jugadores.deleteOne({ "DNI": "09876543X" });
{ acknowledged: true, deletedCount: 1 }
test>
```

```
test> db.jugadores.deleteMany({ "Nacionalidad": "Uruguay" });
{ acknowledged: true, deletedCount: 1 }
test>
```

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

```
test> db.jugadores.updateOne(
...   { "DNI": "09876543X" },
...   { $set: { "Telefono": "555555555" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test>
```

```
test> db.jugadores.updateMany(
...   { "Nacionalidad": "Uruguay" },
...   { $set: { "Actuaciones_Seleccion": 10 } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test>
```

```
test> db.jugadores.findOneAndUpdate(
...   { "Nombre": "Luis" },
...   { $set: { "Peso": "75" } },
...   { returnDocument: "after" }
... );
null
test>
```

6. Consultas:

1. Al menos incluye 5 consultas de datos simples

Consulta para obtener todos los documentos de una colección:

```
test> db.jugadores.find();
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 0,
    club: DBRef('club', ObjectId('66379693f0f6a6f905925bae'))
  },
  {
    _id: ObjectId('663796e0f0f6a6f905925bb2'),
    DNI: '87654321B',
    Nombre: 'Cristiano',
    Apellidos: 'Ronaldo',
    Fecha_Nacimiento: ISODate('1985-02-05T00:00:00.000Z'),
    Direccion: 'Turin',
    Telefono: '9876543210',
    Nacionalidad: 'Portugal',
```

Consulta para obtener un documento específico por su campo único

```
test> db.jugadores.findOne({ "DNI": "12334978A" });
{
  _id: ObjectId('663796e0f0f6a6f905925bb1'),
  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',
  Actuaciones_Seleccion: 0,
  club: DBRef('club', ObjectId('66379693f0f6a6f905925bae'))
}
test>
```

Consulta para contar el número total de documentos en una colección:

```
test> db.jugadores.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
4
```

Consulta para obtener documentos que coincidan con un filtro específico:

```
test> db.jugadores.find({ "Nacionalidad": "Argentina" });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 0,
    club: DBRef('club', ObjectId('66379693f0f6a6f905925bae'))
  }
]
test>
```

Consulta para ordenar documentos por un campo específico:

```
test> db.jugadores.find().sort({ "Nombre": 1 });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb2'),
    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',
    Actuaciones_Seleccion: 0,
    club: DBRef('club', ObjectId('66379693f0f6a6f905925baf'))
  },
  {
    _id: ObjectId('6637a9f6e9c50a354579f154'),
    DNI: '23456789Z',
    Nombre: 'Gareth',
    Apellidos: 'Bale'
```

2. Al menos 3 consultas con arrays

consultar los jugadores que pertenecen al barcelona

```
test> db.jugadores.find({ "club": { $in: ["FC Barcelona"] } });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 0,
    club: 'FC Barcelona'
  }
]
```

consultar los jugadores con mas de 20 actuaciones en la seleccion

```
test> db.jugadores.find({ "Actuaciones_Seleccion": { $gt: 20 } });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 30,
    club: 'FC Barcelona'
  }
]
```

Buscar jugadores que tengan una altura superior a 180 cm y un peso inferior a 70 kg:

```
test> db.jugadores.find({ "Talla": { $gt: "180" }, "Peso": { $lt: "70" } })
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 30,
    club: 'FC Barcelona'
  },
  {
    _id: ObjectId('663796e0f0f6a6f905925bb3'),
    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',
    Actuaciones_Seleccion: 0,
    club: DBRef('club', ObjectId('66379693f0f6a6f905925bb0'))
  }
]
```

3. Al menos 3 consultas con documentos embebido

Buscar jugadores que pertenezcan a un club con un palmarés superior a 20 títulos

```
test> db.jugadores.find({ "club.Palmares": { $gt: "20" } });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 30,
    club: [
      {
        _id: ObjectId('66379693f0f6a6f905925bae'),
        Nombre: 'FC Barcelona',
        Palmares: '25',
        Ciudad: 'Barcelona',
        Division: 1,
        Internacionalidad: 1,
        DNI_Entrenador: '12345678A'
      }
    ],
    Nombre_Club: 'FC Barcelona'
  }
]
```

Consulta para encontrar jugadores que hayan jugado en un estadio con una capacidad superior a 50000 personas

```
test> db.jugadores.find({ "club.Estadio.Capacidad": { $gt: 50000 } });
[
  {
    _id: ObjectId('663796e0f0f6a6f905925bb1'),
    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',
    Actuaciones_Seleccion: 30,
    club: {
      _id: ObjectId('66379693f0f6a6f905925bae'),
      Nombre: 'FC Barcelona',
      Palmares: '25',
      Ciudad: 'Barcelona',
      Division: 1,
      Internacionalidad: 1,
      DNI_Entrenador: '12345678A',
      Estadio: {
        Nombre_Estadio: 'Camp Nou',
        Capacidad: 99000,
        Ciudad: 'Barcelona'
      }
    },
    Nombre_Club: 'FC Barcelona'
  }
]
```

4. Al menos 1 consulta de agrupación

Total de Jugadores por Club

```
test> db.jugadores.aggregate([
...   {
...     $group: {
...       _id: "$club.Nombre",
...       totalJugadores: { $sum: 1 }
...     }
...   }
... ]);
[
  { _id: null, totalJugadores: 3 },
  { _id: 'FC Barcelona', totalJugadores: 1 }
]
```

Salario Promedio de Jugadores por Nacionalidad

```
test> db.jugadores.aggregate([
...   {
...     $group: {
...       _id: "$Nacionalidad",
...       salarioPromedio: { $avg: "$Contrato.Salario" }
...     }
...   }
... ]);
[
  { _id: 'Portugal', salarioPromedio: null },
  { _id: 'Brasil', salarioPromedio: null },
  { _id: 'Argentina', salarioPromedio: null },
  { _id: 'Gales', salarioPromedio: null }
]
```