Ejercicio 4

Modelado de datos en MongoDB

Parte 1. Relaciones 1 a n con documentos integrados

a) Añadir, mediante el comando `update()`, los siguientes comentarios a las noticias

- **Noticia "MARTE"**
  - **Comentario 1**
    autor: Luis
    fecha: 6 de marzo de 2015
    comentario: "¿A dónde vamos a ir a parar?"

```
  var cursor = db.noticias.find({etiqueta_principal: "MARTE"},
{_id:1})
  var vID = cursor.next()

  db.noticias.update({_id: vID._id},
     {$set:{comentarios:[
         {
             autor: "Luis",
             fecha: ISODate("2015-03-06"),
             comentario: "¿A donde vamos a ir a parar?"}]
         }
     });
```

  - **Comentario 2**
    autor: Sandra
    fecha: 7 de marzo de 2015
    comentario: "Yo me iba a vivir a Marte con tal de quitarme de aquí jeje"
    - ⋆ **Respuesta 1:**
      autor: Tomás
      fecha: 7 de marzo de 2015
      comentario:"¿A cuánto sale el billete de ida y vuelta?"

```
  db.noticias.update({_id: vID._id},
     {$push:{comentarios:
         {
             autor: "Sandra",
             fecha: ISODate("2015-03-07"),
```

```
                comentario: "Yo me iba a Marte con tal de quitarme
de aquí jeje",
                respuestas:[
                {
                    autor: "Tomás",
                    fecha: ISODate("2015-03-07"),
                    comentario: "¿A cuánto sale el billete de ida
y vuelta?"
                }]
            }
            }
        });
```

- **Noticia "Whatsapp"**
  - **Comentario 1**
    autor: Carmen
    fecha: 10 de Marzo de 2015
    comentario: "Pero si son todos amarillos!!!!"
  - **Comentario 2**
    autor: Tomás fecha: 11 de Marzo de 2015
    comentario: "¿Para cuándo un emoticono de corte de mangas?"
    - **Respuesta 1:**
      autor: Lucas
      fecha: 11 de Marzo de 2015
      comentarios: "Por lo menos una peineta, jajaja"
    - **Respuesta 2:**
      autor: Pili
      fecha: 12 de Marzo de 2015
      comentario: "Seguro que sería el mas usado"

```
  db.noticias.update({_id: vID._id},
     {$set:{comentarios:[
         {
             autor: "Carmen",
             fecha: ISODate("2015-03-10"),
             comentario: "Pero si son todos amarillos!!!!!"
         },
         {
             autor: "Tomás",
             fecha: ISODate("2015-03-11"),
             comentario: "¿Para cuándo un emoticono de un corte
de mangas?",
             respuestas:[
```

```
                    {
                        autor: "Lucas",
                        fecha: ISODate("2015-03-11"),
                        comentario: "Por lo menos una peineta,
    jajaja"
                    },

                    {
                        autor: "Pili",
                        fecha: ISODate("2015-03-11"),
                        comentario: "Seguro que sería el mas usado"
                    }
                ]
            }
        ]
        }});
```

b) Realizar las siguientes consultas

- Mostrar la noticia y todos los comentarios de la noticia "Marte". Tiene que mostrar, únicamente, el titular de la noticia y los comentarios, **pero sin las respuestas**

```
db.getCollection('noticias').aggregate([
   {$match:{etiqueta_principal:"MARTE"}},
   {$project:{titular:1, comentarios:1}},
   {$project:{"comentarios.respuestas":0}},
   ])
```

- Mostrar, únicamente, las respuestas de los comentarios de la noticia "Whatsapp"

```
db.getCollection('noticias').aggregate([
   {$match:{etiqueta_principal:"WhatsApp"}},
   {$project:{"comentarios.respuestas":1}},
   ])
```

- Mostrar el autor de la primera respuesta del segundo comentario de la noticia "WhatsApp"

```
{
        var cursor = db.noticias.find(
            {etiqueta_principal: "WhatsApp"},
            {_id: 0, comentarios: 1}
            )

        var noticia = cursor.next()
```

```
        print(noticia.comentarios[1].respuestas[0].autor)
}
```

Parte 2: Relaciones n a n con referencias

a) Para añadir las referencias(en ambos sentidos), debemos añadir un campo de tipo array que contendrá los *object_id* de las noticias relacionadas. Relacionar las siguientes noticias:

- Noticia "MARTE" relacionada con "FÍSICA" e "INTELIGENCIA ARTIFICIAL"

```
{

  var not_marte = db.getCollection('noticias').find({etiqueta_principal:"MARTE"},
{_id:1}).next()
  var id_marte = not_marte._id

  var not_fisica = db.getCollection('noticias').find({etiqueta_principal:
"FÍSICA"}, {_id:1}).next()
  var id_fisica = not_fisica._id

  var not_ia = db.getCollection('noticias').find({etiqueta_principal:
"INTELIGENCIA ARTIFICIAL"}, {_id:1}).next()
  var id_ia = not_ia._id

  db.noticias.update({_id: id_marte},
                     {$set:{noticias_relacionadas:[id_fisica, id_ia]}
                         })

  db.noticias.update({_id: id_fisica},
                     {$set:{noticias_relacionadas:[id_marte]}
                         })
  db.noticias.update({_id: id_ia},
                     {$set:{noticias_relacionadas:[id_marte]}
                         })
}
```

- Noticia "CARTAS AL DIRECTOR" relacionada con "WhatsApp"

```
{

  var not_cartas = db.getCollection('noticias').find({etiqueta_principal:"CARTAS
AL DIRECTOR"}, {_id:1}).next()
  var id_cartas = not_cartas._id

  var not_whatsapp = db.getCollection('noticias').find({etiqueta_principal:"Whats
{_id:1}).next()
```

```
  var id_whatsapp = not_whatsapp._id

  db.noticias.update({_id: id_cartas},
                     {$set:noticias_relacionadas:[id_whatsapp]}
                     )

  db.noticias.update({_id: id_whatsapp},
                     {$set:noticias_relacionadas:[id_cartas]}
                     )

  }
```

- Noticia "FÍSICA" relacionada con "INTELIGENCIA ARTIFICIAL"

```
  {

  var not_fisica = db.getCollection('noticias').find({etiqueta_principal:"FÍSICA"
{_id:1}).next()
  var id_fisica = not_fisica._id

  var not_ia = db.getCollection('noticias').find({etiqueta_principal:"INTELIGENCI
ARTIFICIAL"}, {_id:1}).next()
  var id_ia = not_ia._id

  db.noticias.update({_id: id_fisica},
                     {$push:{noticias_relacionadas:id_ia}}
                     )

  db.noticias.update({_id: id_ia},
                     {$push:{noticias_relacionadas:id_fisica}}
                     )

  }
```

b) Realizar la siguiente consulta

- Mostrar la etiqueta principal de las noticias relacionadas con la noticia FÍSICA

```
  doc_fisica = db.getCollection('noticias').find({etiqueta_principal:"FÍSICA"}).n
  list_relacionadas = doc_fisica.noticias_relacionadas

  list_relacionadas.forEach(
      function(not_id){
          not_rel = db.getCollection('noticias').find({_id: not_id},
{_id: 0}).next()
          print(not_rel.etiqueta_principal)
      }
```

)

Parte 3: Relaciones 1 a n con documentos integrados

Tenemos información sobre equipos de baloncesto y jugadores de baloncesto. Cada equipo puede tener hasta 12 jugadores y, evidentemente, un jugador sólo pertenece a un equipo. Para modelar esta relación se ha optado por incrustar un documento dentro de otro.

- **¿Cuál será la colección principal?, ¿por qué?**

  La colección principal será el campeonato, y estará formada por los equipos, dentro de los cuales se encontrarán los jugadores.

  Debe ser así para poder asociar correctamente los jugadores a cada equipo, y asegurar que estos no estén en mas de un equipo.

a) Crear la colección "equipos" con la siguiente información

Equipo	**Pabellón (espectadores)**	**Fundación**	**Sede**
F.C. Barcelona	Palau Blaugrana (7585 espectadores)	1926	Barcelona
FIATC Joventut	Olímpico de Barcelona (12500 espectadores)	1930	Badalona
Real Madrid C.F.	Palacio de los Deportes (12500 espectadores)	1932	Madrid

Además, cada documento del equipo debe tener la información de los jugadores

Número	Nombre	Nacionalidad	Altura	Edad
5	Doellman, Justin	USA	2.04	30
9	Huertas, Marcelinho	ITA	1.91	31
10	Abrines, Álex	ESP	1.98	21

Número	Nombre	Nacionalidad	Altura	Edad
0	Mallet, Demond	USA	1.82	37
6	Vidal, Sergi	ESP	1.98	33
9	Llovet, Nacho	ESP	2.01	23

Número	Nombre	Nacionalidad	Altura	Edad
4	Rivers, K.C.	GNB	1.96	28
14	Ayón, Gustavo	MEX	2.06	29
7	Campazzo, Facundo	ARG	1.79	24

- **F.C. Barcelona**

```
{
    nombre_equipo: "F.C. Barcelona",
    pabellon: {
        nombre: "Palau Blaugrana",
        espectadores: 7585
    },
    fundacion: 1926,
    sede: "Barcelona",
    jugadores: [
        {
            numero: 5,
            nombre: "Doellman, Justin",
            nacionalidad: "USA",
            altura: 2.04,
            edad: 30
        },
        {
            numero: 9,
            nombre: "Huertas, Marcelinho",
            nacionalidad: "ITA",
            altura: 1.91,
            edad: 31
        },
        {
            numero: 10,
            nombre: "Abrines, Alex",
            nacionalidad: "ESP",
            altura: 1.98,
            edad: 21
        }
    ]
}
```

- **FIATC Joventut**

```
{
    "_id" : ObjectId("5fad4a77a0eb19a9b511b34f"),
    "nombre_equipo" : "FIATC Joventut",
    "pabellon" : {
        "nombre" : "Olímpico de Barcelona",
        "espectadores" : 12500
    },
    "fundacion" : 1930,
    "sede" : "Badalona",
    "jugadores" : [
```

```
            {
                "numero" : 0,
                "nombre" : "Mallet, Demond",
                "nacionalidad" : "USA",
                "altura" : 1.82,
                "edad" : 37
            },
            {
                "numero" : 6,
                "nombre" : "Vidal, Sergi",
                "nacionalidad" : "ESP",
                "altura" : 1.98,
                "edad" : 33
            },
            {
                "numero" : 9,
                "nombre" : "Llovet, Nacho",
                "nacionalidad" : "ESP",
                "altura" : 2.01,
                "edad" : 23
            }
        ]
    }
```

- **Real Madrid C.F.**

```
    {
        "_id" : ObjectId("5fad60b9a0eb19a9b511b51b"),
        "nombre_equipo" : "Real Madrid C.F.",
        "pabellon" : {
            "nombre" : "Palacio de los Deportes",
            "espectadores" : 12500
        },
        "fundacion" : 1932,
        "sede" : "Madrid",
        "jugadores" : [
            {
                "numero" : 4,
                "nombre" : "Rivers K.C.",
                "nacionalidad" : "GNB",
                "altura" : 1.96,
                "edad" : 28
            },
            {
                "numero" : 14,
```

```
            "nombre" : "Ayón, Gustavo",
            "nacionalidad" : "MEX",
            "altura" : 2.06,
            "edad" : 29
        },
        {
            "numero" : 7,
            "nombre" : "Campazzo, Facundo",
            "nacionalidad" : "ARG",
            "altura" : 1.79,
            "edad" : 24
        }
    ]
}
```

b) Realizar las siguientes consultas (estas consultas utilizan condiciones de búsqueda o proyectan sobre campos de los documentos integrados):

- Obtener toda la información de los jugadores del Joventut

```
db.getCollection('Equipos').aggregate([
    {$match: {nombre_equipo: "FIATC Joventut"}},
    {$project: {_id: 0, jugadores:1}}
    ])
```

- Obtener el nombre del equipo que tenga algún jugador argentino

```
db.getCollection('Equipos').aggregate([
    {$match: {"jugadores.nacionalidad": "ARG"}},
    {$project: {nombre_equipo:1, _id:0}}
    ])
```

- Obtener el nombre del equipo que tenga algún jugador español de 33 años

```
db.getCollection('Equipos').aggregate([
    {$match: {"jugadores.edad": 33, "jugadores.nacionalidad": "ESP"}},
    {$project: {nombre_equipo:1, _id:0}}
    ])
```

- En MongoDB, una consulta devuelve un documento completo, es decir, no se pueden devolver sólo los elementos de un subdocumento (que a su vez están dentro de un array) que cumplan una condición. Por ejemplo, para obtener el nombre y la altura de los jugadores que miden más de 2 metros se puede utilizar la sentencia:

```
db.equipos.find({"jugadores.altura": {$gt:2}}, {"jugadores.nombre":1,
"jugadores.altura":1, _id:0})
```

Sin embargo, esta consulta devuelve todos los jugadores de un equipo que, al menos, tenga un jugador de más de 2 metros ya que los jugadores (subdocumentos) están dentro de un array.

La solución debe hacerse a nivel de código. Se debe recoger la consulta en un cursor y recorrerlo para mostrar, únicamente, los elementos del array que cumplen la condición.

```
  jugadores_altos = db.getCollection('Equipos').find({"jugadores.altura":
{$gt:2}},
                                        {"jugadores.nombre":1, "jugadores.altura":1,
_id:0}).next()
  array_jug = jugadores_altos.jugadores

  array_jug.forEach(
      function(jugador){
          if(jugador.altura >= 2){
              print("nombre: " + jugador.nombre + "\naltura: " +
jugador.altura)
          }
      }
  )
```

Parte 4. Relaciones 1 a n con referencias

Ahora queremos mantener los documentos de los equipos y los documentos de los jugadores en colecciones separadas y relacionarlos mediante referencias.

- **¿Qué es más eficiente?, ¿poner referencias de los jugadores en el equipo o poner la referencia del equipo en los jugadores?**

  Creemos sería mas eficiente poner las referencias de los jugadores dentro del equipo, puesto que nos permitiría acceder a todos los jugadores del equipo en una sola consulta.

a) Crear las colecciones "equipos" y "jugadores" y relacionarlas mediante referencias

Empezamos creando la colección de jugadores:

```
/* 1 */
{
    "_id" : ObjectId("5fad87afa0eb19a9b511b8c4"),
    "numero" : 10,
    "nombre_jugador" : "Abrines, Alex",
    "nacionalidad" : "ESP",
    "altura" : 1.98,
    "edad" : 21
```

```
}

/* 2 */
{
    "_id" : ObjectId("5fad87f3a0eb19a9b511b8d0"),
    "numero" : 5,
    "nombre_jugador" : "Doellman, Justin",
    "nacionalidad" : "USA",
    "altura" : 2.04,
    "edad" : 30
}

/* 3 */
{
    "_id" : ObjectId("5fad8807a0eb19a9b511b8d5"),
    "numero" : 9,
    "nombre_jugador" : "Huertas, Marcelino",
    "nacionalidad" : "ITA",
    "altura" : 1.91,
    "edad" : 31
}

/* 4 */
{
    "_id" : ObjectId("5fad8942a0eb19a9b511b90d"),
    "numero" : 0,
    "nombre_jugador" : "Mallet, Demond",
    "nacionalidad" : "USA",
    "altura" : 1.82,
    "edad" : 37
}

/* 5 */
{
    "_id" : ObjectId("5fad8942a0eb19a9b511b910"),
    "numero" : 6,
    "nombre_jugador" : "Vidal, Sergi",
    "nacionalidad" : "ESP",
    "altura" : 1.98,
    "edad" : 33
}

/* 6 */
{
    "_id" : ObjectId("5fad8942a0eb19a9b511b913"),
```

```
    "numero" : 9,
    "nombre_jugador" : "Llovet, Nacho",
    "nacionalidad" : "ESP",
    "altura" : 2.01,
    "edad" : 23
}

/* 7 */
{
    "_id" : ObjectId("5fad89fea0eb19a9b511b92f"),
    "numero" : 4,
    "nombre_jugador" : "Rivers, K.C.",
    "nacionalidad" : "GNB",
    "altura" : 1.96,
    "edad" : 28
}

/* 8 */
{
    "_id" : ObjectId("5fad89fea0eb19a9b511b932"),
    "numero" : 14,
    "nombre_jugador" : "Ayón, Gustavo",
    "nacionalidad" : "MEX",
    "altura" : 2.06,
    "edad" : 29
}

/* 9 */
{
    "_id" : ObjectId("5fad89fea0eb19a9b511b935"),
    "numero" : 7,
    "nombre_jugador" : "Campazzo, Facundo",
    "nacionalidad" : "ARG",
    "altura" : 1.79,
    "edad" : 24
}
```

Luego creamos los equipos, añadiendo las referencias a los jugadores:

```
{
    "_id" : ObjectId("5fad8c0ea0eb19a9b511b988"),
    "nombre_equipo" : "F.C. Barcelona",
    "pabellon" : {
        "nombre" : "Palau Blaugrana",
        "espectadores" : 7585
    },
```

```
    "fundacion" : 1926,
    "sede" : "Barcelona",
    "jugadores" : [
        ObjectId("5fad87f3a0eb19a9b511b8d0"),
        ObjectId("5fad8807a0eb19a9b511b8d5"),
        ObjectId("5fad87afa0eb19a9b511b8c4")
    ]
}

{
    "_id" : ObjectId("5fad8d50a0eb19a9b511b9c1"),
    "nombre_equipo" : "FIATC Joventut",
    "pabellon" : {
        "nombre" : "Olímpico de Barcelona",
        "espectadores" : 12500
    },
    "fundacion" : 1930,
    "sede" : "Badalona",
    "jugadores" : [
        ObjectId("5fad8942a0eb19a9b511b90d"),
        ObjectId("5fad8942a0eb19a9b511b910"),
        ObjectId("5fad8942a0eb19a9b511b913")
    ]
}
{
    "_id" : ObjectId("5fad8e55a0eb19a9b511b9eb"),
    "nombre_equipo" : "Real Madrid C.F.",
    "pabellon" : {
        "nombre" : "Palacio de los deportes",
        "espectadores" : 12500
    },
    "fundacion" : 1932,
    "sede" : "Madrid",
    "jugadores" : [
        ObjectId("5fad89fea0eb19a9b511b92f"),
        ObjectId("5fad89fea0eb19a9b511b932"),
        ObjectId("5fad89fea0eb19a9b511b935")
    ]
}
```

b) Realizar las mismas consultas del ejercicio anterior

- Obtener toda la información de los jugadores del Joventut

```
  equipo = db.getCollection('EquiposRef').find({nombre_equipo: "FIATC
Joventut"}).next()
```

```
  list_jugadores = equipo.jugadores

  list_jugadores.forEach(
      function(id_jugador){
          jugador = db.getCollection('JugadoresRef').find({_id: id_jugador},
{_id: 0}).next()
          print(jugador)
      }
  )
```

- Obtener el nombre del equipo que tenga algún jugador argentino

```
  cur_jugadores = db.getCollection('JugadoresRef').find({nacionalidad:
"ARG"}, {_id:1}).next()
  id_jugador = cur_jugadores._id

  equipo = db.getCollection('EquiposRef').find({jugadores:id_jugador},
{nombre_equipo:1, _id: 0}).next()
  print(equipo.nombre_equipo)
```

- Obtener el nombre del equipo que tenga algún jugador español de 33 años

```
  cur_jugadores = db.getCollection('JugadoresRef').find({nacionalidad:
"ESP", edad:33}, {_id:1}).next()
  id_jugador = cur_jugadores._id

  equipo = db.getCollection('EquiposRef').find({jugadores:id_jugador},
{nombre_equipo:1, _id: 0}).next()
  print(equipo.nombre_equipo)
```

Conclusión

Tras realizar las consultas, hemos comprobado que la decisión de diseño de referenciar a los jugadores dentro del equipo ha resultado menos eficiente que haber referenciado el equipo dentro de los jugadores.

Aunque el diseño actual permite acceder a todos los jugadores de un equipo, el buscar el equipo de un jugador individual requiere un esfuerzo adicional, puesto que hay que buscar primero el jugador que cumpla las características, y luego buscar su referencia dentro de los equipos.