

ÍNDICE:

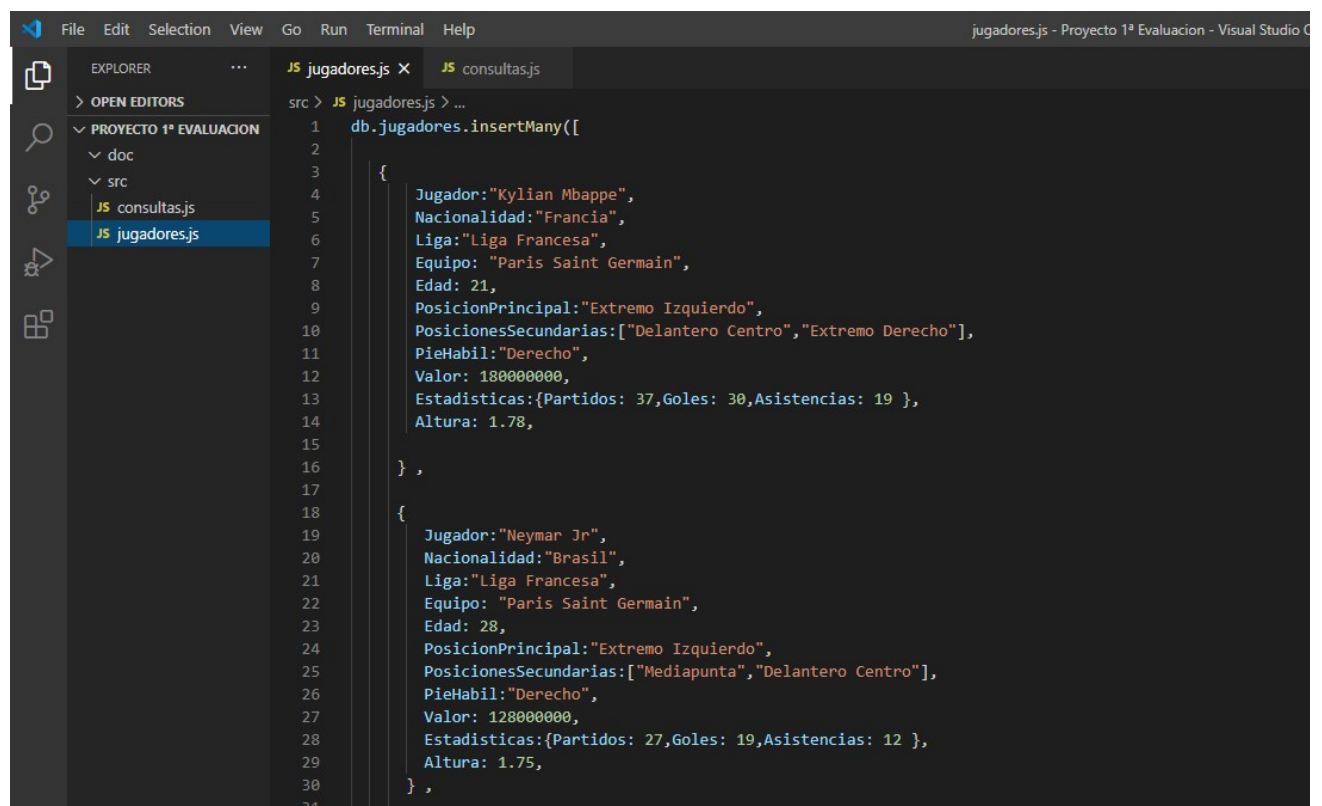
- Introducción.
- Estructura de los Documentos.
- Operadores Utilizados.

Introducción.

El proyecto está basado en una colección donde he introducido unos 42 futbolistas en total con valores diferentes estructurados en distintos tipos de campos.(cadenas de caracteres, números, tipo documento, tipo array...).

Luego he realizado una serie de consultas con distintos operadores para encontrar los distintos jugadores que busquemos.

Esto puede ser útil , por ejemplo para un ojeador de un equipo , que pretende encontrar una serie de jugadores con unas características específicas de manera rapida.



```
1 db.jugadores.insertMany([
2
3   {
4     Jugador:"Kylian Mbappe",
5     Nacionalidad:"Francia",
6     Liga:"Liga Francesa",
7     Equipo: "Paris Saint Germain",
8     Edad: 21,
9     PosicionPrincipal:"Extremo Izquierdo",
10    PosicionesSecundarias:["Delantero Centro","Extremo Derecho"],
11    PieHabil:"Derecho",
12    Valor: 180000000,
13    Estadisticas:{Partidos: 37,Goles: 30,Asistencias: 19 },
14    Altura: 1.78,
15  },
16
17
18  {
19    Jugador:"Neymar Jr",
20    Nacionalidad:"Brasil",
21    Liga:"Liga Francesa",
22    Equipo: "Paris Saint Germain",
23    Edad: 28,
24    PosicionPrincipal:"Extremo Izquierdo",
25    PosicionesSecundarias:["Mediapunta","Delantero Centro"],
26    PieHabil:"Derecho",
27    Valor: 128000000,
28    Estadisticas:{Partidos: 27,Goles: 19,Asistencias: 12 },
29    Altura: 1.75,
30  },
31 ])
```

Estructura de los documentos.

Los documentos están estructurados de la siguiente manera:

```
31
32 {
33   Jugador:"Lionel Messi",
34   Nacionalidad:"Argentina",
35   Liga:"Liga Española",
36   Equipo: "FC Barcelona",
37   Edad: 33,
38   PosicionPrincipal:"Extremo Derecho",
39   PosicionesSecundarias:["Delantero Centro","Mediapunta"],
40   PieHabil:"Izquierdo",
41   Valor: 100000000,
42   Estadisticas:{Partidos: 44,Goles: 31,Asistencias: 27 },
43   Altura: 1.70,
44 } ,
45
```

Hay un total de 11 campos de distinto tipo:

Cadena de Caracteres:

Jugador, Nacionalidad, Liga, Equipo, PosicionPrincipal, PieHabil.

Números:

Edad, Valor, Altura.

Tipo Array:

PosicionesSecundarias.

Tipo Documentos:

Estadísticas.

Operadores Utilizados.

\$and Devuelve los documentos que cumplan ambos campos impuestos.

Ejemplo:

```
db.jugadores.find({$and: [ {PosicionPrincipal: {$eq: "Portero"}},  
{Valor: {$lte: 100000000}}, {Valor: {$gte: 60000000}}, {Altura: {$gte: 1.90}} ] })  
.pretty()
```

\$gt Muestra los documentos con un campos mayores que el indicado.

Ejemplo:

```
db.jugadores.find({$and: [{Equipo: {$in: ["Manchester City", "Liverpool FC", ]}},  
{Valor: {$gt: 100000000}}, {Jugador: {$not: {$regex: /Salah$/}}}] }).pretty()
```

\$lt Muestra los documentos con campos menores que el indicado.

Ejemplo:

```
db.jugadores.find({$and: [{Nacionalidad: {$in: ["Portugal", "Noruega", "España",  
"Francia", ]}}, {Edad: {$lt: 22}} ] }).pretty()
```

\$gte Muestra los documentos con campos mayores o iguales al indicado.

Ejemplo:

```
db.jugadores.find({$and: [ {PosicionPrincipal: {$eq: "Lateral Derecho"}},  
{"Estadisticas.Partidos": {$gte: 40}} ] }).pretty()
```

\$lte Muestra los documentos con campos menores o iguales al indicado.

Ejemplo:

```
db.jugadores.find({$and: [ {PosicionPrincipal: {$eq: "Portero"}},  
{Valor: {$lte: 100000000}}, {Valor: {$gte: 60000000}}, {Altura: {$gte: 1.90}} ] })  
.pretty()
```

\$ne Muestra los documentos con campos que no son el indicado.

Ejemplo:

```
db.jugadores.find({$and: [ {PosicionPrincipal: {$eq: "Defensa Central"}},  
{ Liga: { $ne: "Liga Inglesa" }},{ $nor: [{ Edad: {$gt: 30} }] } ]}).pretty()
```

\$in Coincide con cualquiera de los valores especificados en un array.

Ejemplo:

```
db.jugadores.find({$and: [{Nacionalidad: { $in: [ "Portugal", "Noruega", "España",  
"Francia", ] }},{Edad:{$lt:22}} ] }).pretty()
```

\$nin No coincide con ninguno de los valores especificados en un array.

Ejemplo:

```
db.jugadores.find({$and: [{PosicionPrincipal: {$eq: "Delantero Centro"}},  
{PosicionesSecundarias: { $nin: ["Extremo Derecho", "Extremo Izquier", "Mediapunta"] }},  
{Altura:{$gte: 1.95}},{PieHabil: { $ne: "Izquierdo" } } ]}).pretty()
```

\$not Invierte el efecto de una expresión de consulta y devuelve documentos que no coinciden con la expresión de consulta.

Ejemplo:

```
db.jugadores.find({$and: [{Equipo: { $in: [ "Manchester City", "Liverpool FC", ] }},  
{Valor:{$gt:100000000}},{Jugador: { $not:{$regex: /Salah$/}}} ]}).pretty()
```

\$or Devuelve los documentos que cumplan alguna de las condiciones impuestas.

Ejemplo:

```
db.jugadores.find({ $or: [ {Jugador: {$regex: /^K/}},{Altura: {$gte: 1.95}},  
{Altura: {$lt: 1.70}},{ "Estadisticas.Partidos": {$lt: 10}} ] }).pretty()
```

\$nor Devuelve los documentos que no cumplan alguna de las condiciones impuestas.

Ejemplo:

```
db.jugadores.find({$and: [ {PosicionPrincipal: {$eq: "Defensa Central"}},  
{ Liga: { $ne: "Liga Inglesa" }},{ $nor: [{ Edad: {$gt: 30} }] } ]}).pretty()
```

\$regex Devuelve documentos donde los valores coinciden con una expresión regular especificada.

Ejemplo:

```
db.jugadores.find( { Jugador: { $regex: /vinga$/ } } ).pretty()
```

\$size Devuelve documentos si el campo de la matriz tiene un tamaño específico.

Ejemplo:

```
db.jugadores.find( { PosicionesSecundarias: { $size: 3 } } ).pretty()
```