

Proyecto Final

Índice

1. Base de datos de videojuegos
2. Consultas
3. Operadores

Base de datos de videojuegos

En este trabajo he creado una base de datos con 20 juegos distintos incluyendo campos como las desarrolladoras, los géneros, el número de jugadores que pueden participar en una partida...

La estructura de cada documento es la siguiente:

```
db.juegos.insertMany([
  {
    _id: 1,
    titulo: "Destiny 2",
    precio: 0,
    desarrolladora: ["Bungie", "Activision"],
    pegi: 16,
    jugabilidad_mando: true,
    fechasalida: new Date("2017-09-06"),
    genero: ["FPS", "MMORPG", "3D", "Acción", "Looter Shooter"],
    njugadores: [1,2,3,4,5,6,7,8,9,10,11,12],
    especificaciones:[
      {
        minimas: true,
        procesador: "Intel Core i3",
        grafica: "GTX 660",
        memoria:{
          ram: 6,
          almacenamiento: 105}},
      {
        minimas: false,
        procesador: "Intel Core i5",
        grafica: "GTX 970",
        memoria:{
          ram: 8,
          almacenamiento: 105}}
    ],
    tienedlc: true,
    dlc:[
      {nombre: "Los Renegados",
        preciodlc: 25},
      {nombre: "Bastión de sombras",
        preciodlc: 25},
      {nombre: "Más alla de la luz",
        preciodlc: 40},
      {nombre: "La Reina bruja",
        preciodlc: 40}
    ],
  },
],
```

En cada documento los campos: título y género son tipo string, precio y pegi son de números, jugabilidad_mando y tienedlc son booleanos, fechasalida es tipo date, njugadores es un array de números y especificaciones junto con dlc son arrays de documentos.

Consultas

```
db.juegos.find( {$and: [{ jugadores: { $gte: 12 }}, {genero: { $in: ["FPS","Primera Persona","Acción"]}}], {_id: 0, titulo: 1, jugadores: 1, genero: 1}})
```

Con este operador buscaba un juego de acción en primera persona que permitiese un mínimo de 12 personas

```
db.juegos.find(
  {$and: [
    {pegi: { $lt: 14 }},
    {precio: { $lte: 40 }},
    { $or: [
      { "especificaciones.procesador": { $regex: /^i3/ }},
      { "especificaciones.procesador": { $regex: /^i2/ }},
      { "especificaciones.grafica": { $regex: /^GTX 450/ }},
      { "especificaciones.grafica": { $regex: /^GTX 400/ }}} ] },
  {_id: 0, titulo: 1, precio: 1, especificaciones: 1}
)
```

Con este operador buscaba un juego para menores de 14 años, que costase menos de 40 euros y que un ordenador con esas especificaciones pudiese procesarlo

```
db.juegos.find( {$and: [{jugabilidad_mando: true},{genero: "Acción"},{desarrolladora: "DICE"}], {_id: 0, titulo: 1, jugabilidad_mando: 1, desarrolladora: 1, genero: 1}})
```

Con este operador buscamos un juego que se pueda jugar con mando, que sea de acción y que fue desarrollado por DICE

```
db.juegos.find({desarrolladora: { $regex: /Ele.troni. Arts/ }},{titulo: 1, desarrolladora: 1})
```

Con este buscamos un juego desarrollado por una desarrolladora de la cual no sabemos el nombre exacto

```
db.juegos.find( {$and: [{fechasalida: { $gte: new Date("2015-01-01")}}, {genero: { $ne: "Acción"} }], {titulo: 1, fechasalida: 1, genero: 1}})
```

Con este buscamos los juegos que salieron a partir de 2015 que no sean de acción

```
db.juegos.find( {$nor: [{genero: "2D"}, {tienedlc: false}], {_id: 0, titulo: 1}})
```

Con este buscamos todos los juegos en 3D con dlc

```
db.juegos.find( {$and: [{tienedlc: true},{ $expr: { $gte: [ "$precio" , "$$ROOT.preciodlc" ] } } ] }, {_id:0,titulo:1,precio:1,dlc:1})
```

Con este buscamos todos los juegos en los que el precio base supera al precio de sus dlc

```
db.juegos.find({dlc: { $exists: true }}, {titulo:1,dlc:1})
```

Con este buscamos todos los juegos que tienen el campo dlc

```
db.juegos.aggregate([{$group: {_id : "$precio"}}])
```

Con este se nos muestra una lista de los distintos valores que han tomado los precios de los juegos en nuestra base

```
db.juegos.find( { precio: { $mod: [ 4, 0 ] } }, {titulo:1,precio:1})
```

Con este buscamos los juegos en los que el precio dividido entre 4 da 0 de resto

Operadores

- **\$and**
Muestra todos los documentos que cumplen todas las condiciones
- **\$gte**
Muestra los documentos en los que el valor del campo seleccionado supera al valor seleccionado
- **\$in**
Muestra los documentos en los que el campo seleccionado tiene uno de los valores especificados
- **\$or**
Muestra los documentos que cumplen una de las condiciones especificadas
- **\$regex**
Permite buscar y utilizar patrones en queries de tipo string
- **\$nor**
Muestra los documentos que no cumplen ninguna de las condiciones
- **\$ne**
Muestra todos los documentos en los que los valores de los campos no son iguales a los especificados
- **\$expr**
Permite utilizar expresiones de agregación en los queries
- **\$exists**
Muestra los documentos en los que existen el campo seleccionado
- **\$group**
Permite agrupar los valores de distintos campos y mostrarlos
- **\$mod**
Muestra los documentos en los que los valores de un elemento divididos por un número especificado da como resto el valor indicado