

PROYECTO FINAL MONGODB

Carlos Bustamante Sequeirs

1º ASIR-A 13/11/2021

Índice

Base de Datos	2
Estructura	2
Campos	3
Consultas	6

Base de Datos

Estructura

He creado una base de datos sobre monitores con cada documento con la siguiente estructura:

```
{
  _id:1,
  marca: "acer",
  modelo: "XV272P",
  pulgadas: [24, 27],
  ratioAspecto: {formato: "16:9", curvatura: false},
  pantalla:
    {
      resolucion: "FHD",
      panel: "IPS",
      resfresco: 144,
      bitsProfundidad: 8,
      brillo: 400,
      HDR: 400
    },
  peana:
    {
      ajustable: ["altura", "inclinacion", "rotacion"],
      materiales: ["plastico", "aluminio", "goma"],
      colores: ["negro", "rojo"]
    },
  espacioColor: [{estandar: "sRGB", cobertura: 99},],
  altavoces: true,
  tecnologias:
    ["freesync", "gsync"],
  conectividad:
    {
      USB:
        {
          version:3.0,
          numero: 4
        },
      HDMI:
        {
          version: 2.0,
          numero: 2
        },
      DisplayPort:
        {
          version: 1.4,
          numero: 1
        }
    }
}
```

```

    },
    anunciado: new Date("2019-9-15"),
    puestoEnVenta: new Date ("2020-4-7"),
    comprar: [{tienda: "amazon", precio: 260.90}, {tienda:
"pccomponentes", precio: }, {tienda: "paginaoficial", precio:} ]
  }

```

Campos

A continuación, voy a explicar cada campo que conforman los documentos indexados en la base de datos. Los campos están escritos en minúsculas (exceptuando los que son siglas) y los que son palabras compuesta se ha utilizado el modo de escritura camelcase.

_id: Es el ID que le damos nuestro documento. El ID es único para cada documento dentro de cada colección. Podemos poner nosotros el que queramos o podemos omitirlo y mongo le dará uno automático

marca: Es un campo de tipo string que recoge el nombre de la marca del modelo.

modelo: Es un campo de tipo string que recoge el "ID" del monitor dentro de cada marca.

pulgadas: En un campo de tipo array numerico. Expresa el número de pulgadas disponibles para cada monitor. Sus valores suelen oscilar entre 22-50

ratioAspecto: Es un campo de tipo documento. Este campo recoge el formato de la pantalla y si tiene curvatura o no.

Este campo tipo documento tiene dos campos embebidos:

Formato: De tipo string y define la relación entre el alto y el largo de la pantalla. El formato mas común es el 16:9 aunque también podemos encontrar pantallas ultrawide de 21:9 o más.

Curvatura: Es un campo de tipo booleano e indica si la pantalla es curva o no: **Valores posibles (true/false)**

Pantalla: Es un campo de tipo documento en el que recoge toda la información pertinente a la pantalla:

Campos embebidos:

resolucion: Es de tipo string. Registra la información de la cantidad de pixeles que tiene el monitor. Lo registra mediante el nombre común que se les suele dar a las resoluciones. **Valores posibles:** FHD, QHD, UHD: Pueden llevar delante la letra W que significaría wide(amplio) o D que significaría doble

panel: Es de tipo string. Define el tipo de tecnología del panel. Que pueden ser de tres tipos. IPS, VA y TN. De estas puede haber diferentes variantes como la FIPS o QVA pero estas estarían incluidas en el grupo anterior

refresco: Es de tipo numérico. Registra la cantidad de veces que se refresca la pantalla. El valor está en Herzios. Valores Posibles: Oscilan entre 60 – 360. Aunque los mas comunes suelen ser 60, 144 y 240

bitsProfundidad: Es de tipo numérico. Marca la cantidad de bits de profundidad de color que tiene el monitor. **Valores Posibles:** 6, 8, 10

brillo: Es de tipo numérico. Indica la cantidad de brillo máximo que emite el monitor. Esta medido en nits. **Valores Posibles:** 200~1600

HDR: Tipo numérico. Indica si el monitor tiene la tecnología HDR y cuál de los estándares certificados por VESA. **Valores Posibles:** Este campo puede aparecer o no y puede ser 400, 600, 1000

Peana: Es un campo tipo documento en el que cada campo es un array. Guarda la información pertinente sobre el soporte del monitor.

ajustable: Es un campo de tipo array de strings. Se refiere a las posibilidades de ajuste que tiene el monitor. **Valores Posibles:** El array puede contener todos o ninguno de los valores posibles: “altura”, “inclinacion”, “rotacion”, “orientacion”

materiales: Es un campo de tipo array de strings. Registra con que materiales está construido todo el cuerpo del monitor. **Valores Posibles:** “plastico”...

colores: Es un campo de tipo array de strings. Son los colores del cuerpo del monitor **Valores Posibles:** “negro”...

espacioColor: Es un campo del tipo de arrays de documentos. Registra los espacios de colores compatibles con el monitor y el porcentaje% que cubre de este. Cada documento cuenta con los campos:

estandar: Campo de tipo string. Expresa el estándar certificado que representa. **Valores Posibles:** sRGB, RGB, DCIP3, NTSC

cobertura: Campo de tipo numérico. Expresa en porcentaje cuanto cubre del espacio de color. **Valores Posibles:** Suelen ser valores por encima de 80 y siendo posibles superiores a 100

altavoces: Campo de tipo booleano. Dice si tiene altavoces integrados. **Valores Posibles:** True or False

tecnologias: Es un campo de tipo array de strings. Dice si el monitor tiene alguna tecnología de sincronización adaptativa. **Valores Posibles:** Puede tener 1, ambas o ninguna. "fre sync", "gsync"

conectividad: Es un campo de tipo documento. Guarda los diferentes puertos de conexiones que tiene el monitor. Dentro de cada campo se encuentra otro documento que dice el número de conexiones tiene de ese tipo y la versión que son.

Campos embebidos.

USB: version, numero

HDMI: version, numero

DisplayPort: version, numero

anunciado y puestoEnVenta: Son dos campos de tipo fecha de cuando se anuncio el producto y cuando se puso en venta.

Comprar: Es un campo de tipo array de documentos que guarda las tiendas y sus respectivos precios en las diferentes tiendas.

Consultas

Un jugador de videojuegos profesional que necesita un monitor de un tamaño comprendida entre 24 y 27 con un refresco mayor o igual de 144hz, que sea ajustable en altura e inclinación que tenga alguna tecnologia ya sea freesync o gsync tenga al menos 2 usb y un displayport*/

```
db.monitores.find(
  {
    $and:
    [
      {pulgadas: {$all: [24, 27]}},
      {"pantalla.refresco": {$gte: 144}},
      {"peana.ajustabilidad": {$all: ["altura", "inclinacion"]}},
      {$or:[{tecnologias: {$eq: "freesync"}}, {tecnologias: {$eq:
"gsync" } } ]},
      {"conectividad.USB.number": {$gte: 2}},
      {"conectividad.DisplayPort": {$exists: true}}, {pulgadas:1,
"pantalla.refresco": 1, "peana.ajustabilidad":1, tecnologias:1,
conectividad:1}
    ]
  }, {pulgadas:1, "pantalla.refresco":1, "peana.ajustabilidad":1,
tecnologias:1, conectividad:1}
)
```

En esta consulta utilizo los operadores \$and, \$all, \$gte, \$or, \$eq, \$exists y también he utilizado las projections para que solo me muestre los campos que me interesa

```

/*Un fotografo necesita un monitor para poder editar sus fotos. Necesita
que el monitor tenga una resolución
de QHD, UHD o WQHD una profundidad de bits de 10. Tenga HDR. Que tenga
espacio de color "sRGB" mayor al 97
y que tambien tenga el espacio de color DCIP3 sin importar el porcentaje.
Ademas los quiere ordenados por su
fecha de salida al mercado para*/

db.monitores.find(
  {
    $and:
    [
      {$or: [{"pantalla.resolucion": {$regex:
/Q/}}, {"pantalla.resolucion": {$regex: /^U/}}]},
      {"pantalla.bitsProfundidad": 10},
      {"pantalla.HDR": {$exists: true}},
      {$and: [
        {espacioColor: {$elemMatch: {estandar: "sRGB", cobertura:
{$gte: 97}}}},
        {"espacioColor.estandar": "DCIP3"}
      ]}
    ]
  }, {"pantalla.resolucion":1, "pantalla.bitsProfundidad":1,
"pantalla.HDR":1, "espacioColor":1, puestoEnVenta:1}
).sort({puestoEnVenta:-1})

```

Utilizo los operadores \$and, \$or, \$regex, \$exists, \$elemMatch,

He utilizado las projections para que me muestre los campos que me interesa.

Y he utilizado el método .sort para que me ordene los resultados según su fecha en la que salió al mercado


```
/*Un editor de video necesita un monitor con las siguientes
caracteristicas: Que tenga un Ratio de aspecto mayor de
16:9, que tenga un panel IPS de cualquier tipo, que sea ajustable en
altura e inclinacion, que tenga altavoces
y que se pueda comprar por amazon*/
```

```
db.monitores.find({
  "ratioAspecto.formato": {$ne: "16:9"},
  "pantalla.panel": {$regex: /IPS$/},
  "peana.ajustabilidad": {$all: ["altura", "inclinacion"]},
  altavoces: true,
  "comprar.tienda": "amazon"
}, {"ratioAspecto.formato":1, "pantalla.Panel":1, "altavoces": 1,
"comprar":1})
```

He utilizado los operadores \$ne, \$regex, \$all, las projections

```
/*Quiero conocer todos los monitores que tienen mas de un tamaño de
pulgadas, tienen mas de 60Hz de refresco,
pero menos de 300, que tienen partes hechas de aluminio, fueron
anunciados en 2020*/
```

```
db.monitores.find({
  pulgadas: {$not:{$size: 1}},
  "pantalla.refresco": {$gt: 60, $lt: 300},
  "peana.materiales": "aluminio",
  anunciado: {$gte: new Date("2020-1-1"), $lt: new Date("2021-1-1")},
  "comprar.precio": {$lt: 300}
},
{pulgadas:1, "pantalla.refresco":1, "peana.materiales":1,
"anunciado":1, comprar:1}
)
```

He utilizado los operadores \$not, \$size, \$gt, \$lt, y las projections

```

/*Quiero conocer el número de monitores que tienen panel IPS, 10bits de
profundidad, HDR superior a 400, que no tenga el color blanco ni
plateado,
en el espacio de color tienen un sRGB superior a 100 o tienen mas de dos
estandar de espacio de color*/

db.monitores.find({
  $and:
  [
    {"pantalla.panel": {$regex: /IPS$/}},
    {"pantalla.bitsProfundidad": {$eq: 10}},
    {"pantalla.HDR": {$gt: 400}},
    {"peana.colores": {$nin: ["blanco", "plateado"]}},
    {$or: [
      {espacioColor: {$elemMatch: {estandar: "sRGB", cobertura:
{$gt: 100}}}},
      {espacioColor: {$not: {$size: 2}}}
    ]}
  ]
}).count()

```

He utilizado los operadores \$regex, \$nin, \$or, \$elemMatch, \$not

Además he utilizado el método .count() para que me muestre el número de documentos que coinciden con el find