

## Base de datos II

# Tipos de índices en MongoDB

Los diferentes tipos de índices que existen en MongoDB son :

- **Índices simples o de campo único.** Estos índices se aplican a un solo campo en nuestra colección. Para declarar dicho índice, debemos usar el siguiente comando:

```
db.peliculas.ensureIndex({ "_id": 1 })
```

El número indica que queremos que el índice se ordene de forma ascendente. Si queremos un orden descendente, el parámetro será -1.

- **Índices compuestos.** En este caso, el índice se generará en varios campos.

```
db.peliculas.ensureIndex( { título : 1, género :-1 } )
```

El índice que se generará con esta instrucción agrupará los datos primero por el campo de título y luego por el campo de género. Los índices compuestos nos permiten consultar uno o más campos, sin tener que incluirlos todos. En el ejemplo anterior, podemos consultar tanto por título de película como por género.



- **Índices únicos.** Se requieren índices únicos y múltiples para contener valores únicos. Logramos esto agregando el parámetro único al crearlos.

```
db.peliculas.ensureIndex( { _id : 1 }, {"único":verdadero} )
```

Si intenta insertar valores repetidos en un campo con un índice único, MongoDB devolverá un error.

- **Índices escasos.** Los índices dispersos incluyen todos los documentos, incluso aquellos que no contienen el campo indexado. MongoDB no nos obliga a usar un esquema. Por lo tanto, no hay ningún requisito de que los campos existan en el documento. En este caso, el campo de user\_name puede o no estar en el documento. Para crear índices que solo incluyan documentos cuyo campo indexado existe, usaremos la opción sparse.

```
db.users.ensureIndex( { "user_name" : 1 }, {"sparse":true} )
```

- **Índices de texto.** Al igual que MySQL, MongoDB admite consultas de búsqueda de texto, específicamente en contenido de cadena. Los índices de texto deben ser una cadena o una matriz de elementos de cadena. Podemos crear índices de texto usando el método db.collectionName.createIndex().

```
db.peliculas.createIndex({titulo: "text"})  
'titulo_text'
```



Para buscar con el índice de texto usamos:

```
db.peliculas.find({$text:{$search: "Harry"}}).pretty()
{ _id: ObjectId("61f75b04cae35901929407b2"),
  id: 8,
  título: 'Harry Potter y la piedra filosofal',
  puntaje: Decimal128("10.00"),
  premios: 1,
  fecha_lanzamiento: 2008-04-04T03:00:00.000Z,
  duración: 120,
  género: 'Aventuras' }
{ _id: ObjectId("61f75b04cae35901929407b3"),
  id: 9,
  título: 'Harry Potter y la cámara secreta',
  puntaje: Decimal128("3.50"),
  premios: 2,
  fecha_lanzamiento: 2009-04-08T03:00:00.000Z,
  duración: 200,
  género: 'Aventuras' }
{ _id: ObjectId("61f75b04cae35901929407b0"),
  id: 6,
  título: 'Harry Potter y las Reliquias de la Muerte - Parte 2',
  puntaje: Decimal128("9.00"),
  premios: 2,
  fecha_lanzamiento: 2008-04-07T03:00:00.000Z,
  duración: 190,
  género: 'Suspenso' }
```

Se encontraron 3 películas cuyo título tenía el término "Harry".



Para ver el índice de texto creado, usamos el comando:

```
db.peliculas.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { titulo: 1, genero: -1 }, name: 'titulo_1_genero_-1' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'titulo_text',
    weights: { titulo: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
```

Para eliminar el índice de texto, usamos el método `dropIndex()`.

```
db.peliculas.dropIndex("titulo_1_genero_-1")
{ nIndexesWas: 3, ok: 1 }
```

- **Índice comodín.** Usando el especificador comodín (`$**`), puede crear múltiples campos de índice de texto. Debido al índice de texto comodín, MongoDB indexa todos y cada uno de los campos que contienen datos de cadena en todos los documentos presentes en la colección dada. El índice de texto comodín es útil para datos no estructurados en los que no sabemos qué campo contiene datos de cadena o para consultas ad-hoc. Le permite buscar texto en todos los campos que contienen datos de cadena. El índice de texto comodín puede formar parte del índice compuesto.

```
sdb.peliculas.createIndex( { "$**": "text" } )
```



### → Especificar los pesos

Para un índice de texto, el peso de un campo indexado es la importancia del campo. Para cada campo de índice en el documento, MongoDB suma los resultados, multiplicando el número de coincidencias por peso. Ahora, utilizando esta suma, MongoDB calcula la puntuación del documento. El peso predeterminado del campo de índice es 1 y puede ajustar el peso del índice utilizando el método `createIndex()`.

Ejemplo:

```
db.peliculas.createIndex({titulo:"text", genero:"text"},  
                        {weights:{titulo:10, genero:5},  
                        name:"TextIndex"})
```

Aquí, el peso de los campos de título y género son 10 y 5.