

## DOCUMENTACIÓN PROYECTO PRIMERA EVALUACIÓN

Este proyecto está orientado hacia la creación de una base de datos que contiene datos sobre libros que han sido publicados recientemente.

En ella podemos observar que tenemos una colección llamada Libros, que está en la base de datos CasaDelLibro dentro de MongoDB.

La colección tiene varios campos con diferentes tipos de datos:

- Cadena de caracteres: título, editorial, autor.
- Número: páginas del libro
- Fecha: fecha de publicación
- Tipo documento: sinopsis
- Tipo array: dimensiones del libro en milímetros y géneros.

Para realizar las consultas se han utilizado los siguientes operadores:

COMPARADORES	
\$eq	Coincide con valores que son iguales a un valor especificado.
\$gt	Coincide con valores que son mayores que un valor especificado.
\$gte	Coincide con valores que son mayores o iguales a un valor especificado.
\$in	Coincide con cualquiera de los valores especificados en una matriz.
\$lt	Coincide con valores que son menores que un valor especificado.
\$lte	Coincide con valores que son menores o iguales a un valor especificado.
\$ne	Coincide con todos los valores que no son iguales a un valor especificado.
\$nin	No coincide con ninguno de los valores especificados en una matriz.

LÓGICOS	
\$and	Devuelve todos los documentos que cumplen las condiciones de ambas cláusulas.
\$not	Devuelve documentos que no coinciden con la expresión de consulta.
\$nor	Devuelve todos los documentos que no coinciden con ambas cláusulas.
\$or	Devuelve todos los documentos que cumplen las condiciones de cualquiera de las cláusulas.

ELEMENTOS EXTRAS	
\$regex	Selecciona documentos donde los valores coinciden con una expresión regular especificada.
\$elemMatch	Selecciona documentos si el elemento del campo de matriz coincide con todas las condiciones especificadas.
\$text	Realiza búsqueda de texto.
\$mod	Realiza una operación de módulo sobre el valor de un campo y selecciona documentos con un resultado específico.

Para completar la base de datos se podrían crear varias colecciones para posteriormente relacionarlas, por ejemplo, se podría crear:

Colección llamada AUTORES, donde aparecerían, el idautor, el nombre, apellidos, edad, fecha nacimiento, paisnacimiento.

Colección géneros donde aparecería el idGenero y el nombre del género.

## EXPLICACIÓN OPERADOR \$MOD

Tenemos un campo numérico, en este caso las dimensiones de los libros.

La sintaxis que se debe usar es:

```
db.coleccion.find( campo: { $mod: [ divisor, resto ] } }
```

En el caso de las dimensiones y de la colección libros:

```
db.libros.find( { Dimensiones_mm: { $mod: [4, 0] } } )
```

¿Qué quiere decir esto?

Nos devuelve aquellos libros en los que aparecen números dentro del array que dividiéndolos por el divisor en este caso 4 den 0 de resto.

Por ejemplo:

"Dimensiones\_mm": [ 210, 140] <- el número 210 dividido entre 4 no da 0 de resto, pero si lo hace el número 140.

$210 / 4 = 52.5$  y de resto 1

$140 / 4 = 35$  y de resto 0

"Dimensiones\_mm": [ 240, 154] <- el número 154 dividido entre 4 no da 0 de resto, pero si lo hace el número 240.

$240 / 4 = 60$  y de resto 0

$154 / 4 = 38.5$  y de resto 1

"Dimensiones\_mm": [ 237, 164] <- el número 237 dividido entre 4 no da 0 de resto, pero si lo hace el número 164.

$237 / 4 = 59.25$  y de resto 1

$164 / 4 = 41$  y de resto 0