

PROYECTO 2º EVALUACIÓN

HECHO POR LUCÍA RAMÍREZ MONJE

Este proyecto está enfocado en la base de datos de una [Biblioteca](#).

En la biblioteca tenemos almacenadas las siguientes colecciones de datos:

COLECCIÓN DE LIBROS:

Esta colección contiene los datos de los libros que han pasado por la biblioteca, o que están expuestos ahora mismos. No son libros físicos sino lógicos, la componen los siguientes campos:

Título del libro: "string" o cadena de caracteres.

Editorial o editoriales que lo distribuyen: "string" o cadena de caracteres.

El autor o los autores del libro: "string" o cadena de caracteres.

Fecha en la que se publicó: "date" o fecha.

Numero de páginas del libro: numérico

Código identificador del libro: "string" o cadena de caracteres.

Géneros a los que pertenece: array de strings

COLECCIÓN DE AUTORES:

Esta colección contiene los datos personales de todos los autores que hayan escrito un libro que haya estado en posesión de la biblioteca. La componen los siguientes campos:

DNI del autor: "string" o cadena de caracteres.

Nombre real del autor: "string" o cadena de caracteres.

Fecha de nacimiento: numérico

Fecha de fallecimiento: numérico (no disponible en los autores que aun viven)

Nacionalidad: "string" o cadena de caracteres.

COLECCIÓN DE EJEMPLARES:

Esta colección contiene los volúmenes que están actualmente expuestos en la biblioteca y que pueden ser prestados. La componen los siguientes campos:

Nº identificador del ejemplar (no es el mismo que identifica el libro): numérico

El deterioro para poder llevar una idea de cuando deben ser reemplazados:
booleano

ISBN Nº identificador del libro: "string" o cadena de caracteres.

PROYECTO 2º EVALUACIÓN

HECHO POR LUCÍA RAMÍREZ MONJE

COLECCIÓN DE PRESTAMOS:

Esta colección contiene los datos de los prestamos que se han hecho en la biblioteca, para un caso real harían falta los campos "Fecha de entrega" y "Fecha de Devolución" del libro adquirido. La componen los siguientes campos:

Código del usuario que está vinculado a otra colección donde aparecen sus datos: numérico

NºIdentificador del ejemplar que se ha llevado: numérico

COLECCIÓN DE USUARIOS:

Esta colección contiene los datos de los usuarios que están inscritos en la biblioteca. La componen los siguientes campos:

NºIdentificador único para cada usuario: numérico

Nombre del usuario: "string" o cadena de caracteres.

Contacto, formado por el email y el teléfono del usuario, puede tener no aparecer alguno de los dos, pero nunca los dos a la vez: documento formado por dos cadenas de caracteres.

PROYECTO 2º EVALUACIÓN

HECHO POR LUCÍA RAMÍREZ MONJE

OPERADORES UTILIZADOS

Las operaciones de agregación procesan los registros de datos y devuelven los resultados calculados. Las operaciones de agregación agrupan valores de varios documentos y pueden realizar una variedad de operaciones en los datos agrupados para devolver un único resultado.

Para realizar este proyecto se han utilizado los siguientes operadores:

OPERADOR	DESCRIPCIÓN	SINTAXIS
\$unwind	Deconstruye un campo de matriz a partir de los documentos de entrada para generar un documento para cada elemento. Cada documento de salida es el documento de entrada con el valor del campo de matriz reemplazado por el elemento.	<pre>{ \$unwind: { path: <field path>, includeArrayIndex: <string>, preserveNullAndEmptyArrays: <boolean> } }</pre>
\$unset	Elimina / excluye campos de los documentos.	<pre>{ \$unset: "<field>" }</pre>
\$group	Agrupar los documentos de entrada por la <code>_id</code> especificada y para cada agrupación distinta, genera un documento.	<pre>{ \$group: { _id: <expression>, // Group By Expression <field1>: { <accumulator1> : <expression1> }, ... } }</pre>
\$bucket	Categoriza los documentos entrantes en grupos, denominados depósitos, según una expresión específica y los límites del depósito, y genera un documento por cada depósito.	<pre>{ \$bucket: { groupBy: <expression>, boundaries: [<lowerbound1>, <lowerbound2>, ...], default: <literal>, output: { <output1>: { <\$accumulator expression> }, ... <outputN>: { <\$accumulator expression> } } } }</pre>
\$match	Filtra los documentos para pasar solo los documentos que coinciden con las condiciones especificadas a la siguiente etapa de canalización.	<pre>{ \$match: { <query> } }</pre>

PROYECTO 2º EVALUACIÓN

HECHO POR LUCÍA RAMÍREZ MONJE

\$out	Toma los documentos devueltos por la canalización de agregación y los escribe en una colección específica.	<pre>{ \$out: { db: "<output-db>", coll: "<output-collection>" } }</pre>
\$project	Pasa los documentos con los campos solicitados a la siguiente etapa del proceso. Los campos especificados pueden ser campos existentes de los documentos de entrada o campos recién calculados.	<pre>{ \$project: { <specification(s)> } }</pre>
\$merge	Escribe los resultados de la canalización de agregación en una colección especificada.	<pre>{ \$merge: <collection> }</pre>
\$lookup	Realiza una combinación externa izquierda a una colección no fragmentada en la misma base de datos para filtrar los documentos de la colección "unida" para su procesamiento.	<pre>{ \$lookup: { from: <collection to join>, localField: <field from the input documents>, foreignField: <field from the documents of the "from" collection>, as: <output array field> } }</pre>

PROYECTO 2º EVALUACIÓN

HECHO POR LUCÍA RAMÍREZ MONJE

GRÁFICA

Se pide una consulta individualizada de cuantos libros ha escrito cada autor.

La sintaxis del ejercicio es la siguiente:

```
db.autores.aggregate([
  {
    "$lookup": {
      "from": "libros",
      "localField": "Nombre",
      "foreignField": "Autor",
      "as": "Libros escritos"
    }
  },
  {
    "$unset": [
      "Nacionalidad",
      "DNI",
      "FechaFall",
      "FechaNac",
      "_id"
    ]
  },
  {
    "$project": {
      "Nombre": 1,
      "Libros escritos": { "$size": "$Libros escritos" }
    }
  },
  { "$merge": "Grafico" }
]).pretty()
```

Solución y GRÁFICO:

```
db.Grafico.find({}).pretty()
{ "_id" : ObjectId("60422ece2ae72858e1f378bc"), "Libros escritos" : 1, "Nombre" : "Robert Louis Stevenson" }
{ "_id" : ObjectId("60422ece2ae72858e1f378bd"), "Libros escritos" : 2, "Nombre" : "Arturo Pérez Reverte" }
{ "_id" : ObjectId("60422ece2ae72858e1f378be"), "Libros escritos" : 1, "Nombre" : "Ken Follett" }
{ "_id" : ObjectId("60422ece2ae72858e1f378bf"), "Libros escritos" : 1, "Nombre" : "Juan José Millas" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c0"), "Libros escritos" : 1, "Nombre" : "Blanca García" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c1"), "Libros escritos" : 1, "Nombre" : "Megan Maxwell" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c2"), "Libros escritos" : 4, "Nombre" : "Rosa Montero" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c3"), "Libros escritos" : 4, "Nombre" : "Javier Castillo" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c4"), "Libros escritos" : 1, "Nombre" : "Anne Frank" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c5"), "Libros escritos" : 1, "Nombre" : "Eva García Sáenz de Urturi" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c6"), "Libros escritos" : 1, "Nombre" : "Benjamín Labatut" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c7"), "Libros escritos" : 1, "Nombre" : "Elena Ferrante" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c8"), "Libros escritos" : 1, "Nombre" : "María Reig" }
{ "_id" : ObjectId("60422ece2ae72858e1f378c9"), "Libros escritos" : 1, "Nombre" : "Alice Kellen" }
```

