

Consultas avanzadas en Java con base de datos MongoDB

TAREA PREVIA

Antes de comenzar el desarrollo del programa propuesto en el ejercicio, debes importar los archivos JSON **alumnos.json** y **cursos.json** en las colecciones **alumnos** y **cursos** de una base de datos **formacion**. Esos archivos se encuentran en Moodle.

Podemos copiar los ficheros al contenedor. Para ello utilizaremos el comando **docker cp alumnos.json mongo-server:/root**

Para importar se puede utilizar el comando **mongoimport**:

```
mongoimport --authenticationDatabase admin --db formacion --collection alumnos --file alumnos.json --jsonArray -u root
```

EJERCICIO

Realiza un programa Java que conecta con la base de datos **formacion de MongoDB**, presenta el siguiente menú y, en función de la opción elegida, realiza sobre las colecciones **cursos** y **alumnos** la operación solicitada en las condiciones que se indican a continuación:

- 1.- Añadir notas a alumnos de curso**
- 2.- Añadir tema a curso**
- 3.- Obtener alumnos de curso**
- 4.- Número de alumnos por curso**
- 5.- Modificar nota media de alumno**
- 6.- Modificar horas de curso**
- 7.- Datos de alumno**
- 8.- Nota media en curso**

OPCIÓN 1:

Se pide por teclado el id de un curso y, si existe, se va mostrando el nombre y apellidos de cada uno de sus alumnos y se pide que se dé a cada uno una nota. La nota introducida para cada alumno se añade al array de notas.

OPCIÓN 2:

Se pide por teclado el id de un curso y, si existe, se piden los datos de un tema por teclado y se añade un documento con los datos del tema (título y horas) al array **temas**.

Hoja03_MongoDB_07

OPCIÓN 3:

Se pide por teclado el id de un curso y, si existe, se escribe el título del curso y las horas de duración. A continuación, se escribe un listado ordenado por apellidos, nombre de los alumnos del curso con valor en nota_media mayor o igual que 5 (apellidos, nombre, nota_media).

OPCIÓN 4:

Se escriben en pantalla los títulos de los cursos y cuántos alumnos hay en cada uno de esos cursos.

Debes realizarlo consultando primero los cursos existentes y, para cada curso, obteniendo cuántos alumnos hay. Aplicando el método countDocuments a consultar los alumnos de un curso.

Hay otras alternativas como usar el método **aggregate** que podría obtener cuantos alumnos hay para cada código de curso. Por cada resultado habría que obtener el título del curso. De esta forma no saldrían los cursos en los que no hay alumnos matriculados. Esto es un ejemplo de cómo se obtendría cada id de curso el número de alumnos matriculados.

```
cursor = alumnos.aggregate(  
    Arrays.asList(  
        Aggregates.group("$curso", Accumulators.sum("num", 1))  
    )).iterator();
```

OPCIÓN 5:

Se pide por teclado el id de un alumno y, si existe, se calcula la nota media del array de notas y se modifica con esa nota calculada el valor del atributo **nota_media**.

Debes extraer el contenido del atributo notas en un **ArrayList**.

OPCIÓN 6:

Se pide por teclado el id de un curso y, si existe, se calculan las horas del curso como suma de las horas de los temas. Si no hay temas cargados, las horas serán cero. Con el valor calculado de horas, se modificará el valor del atributo **horas** del curso.

OPCIÓN 7:

Se pide por teclado el id de un alumno y, si existe, se escribe su nombre y apellidos y el título del curso en el que está matriculado.

OPCIÓN 8:

Se pide por teclado el id de un curso y, si existe, se obtiene y escribe la nota media en el curso, es decir, la nota media de las notas medias de los alumnos del curso.