



## Hoja de ejercicios 3-06

---

### Programación Java de BD MongoDB

#### Objetivos:

- Consultar datos de un documento
- Añadir elementos a un array de un documento
- Realizar funciones de agrupamiento
- Controlar relaciones entre documentos de dos colecciones
- Obtener datos de un array de un documento

#### TAREA PREVIA

Antes de comenzar el desarrollo del programa propuesto en el ejercicio 1, 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 dentro de la tarea de Moodle.

#### EJERCICIO 1

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 y media
- 5.- Modificar nota media de alumno
- 6.- Modificar horas de curso
- 7.- Datos de alumno
- 8.- Nota media en curso
- 9.- Subir nota media

#### **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:**



## Hoja de ejercicios 3-06

---

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 al array **temas**.

### 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, cuantos alumnos hay en cada uno de esos cursos y la media de las notas medias de los alumnos de cada curso.

Debes realizarlo usando el método **aggregate** con el que podrías 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 para cada curso el número de alumnos matriculados.

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

Otro modo de realizarlo sería consultando primero los cursos existentes y, para cada curso, obteniendo cuantos alumnos hay. Aplicando el método count a consultar los alumnos de un curso.

### 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:



## Hoja de ejercicios 3-06

---

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.

### **OPCIÓN 9:**

A todos los alumnos del curso que se indique por teclado se les sube la nota media 0.1 puntos.



## **Hoja de ejercicios 3-06**

---