

Taller de Programación 2

Examen Final 2do. llamado Diciembre (16 de diciembre, 2024)

Enunciado: Sistema de Gestión de Calificaciones

Se busca desarrollar un sistema, con persistencia en la memoria del servidor, que permita registrar las notas obtenidas por alumnos, durante la cursada de una materia. La aplicación backend deberá realizar el listado y el reporte de estadísticas de las notas ingresadas a través de distintas consignas que se detallan a continuación.

Los requerimientos son:

1. Implementar endpoints de ingreso y listado:
 - a. Registrar una nota con los siguientes datos:
 - id (autogenerado por el servidor e irrepitable)
 - nombre alumno
 - apellido alumno
 - nota

Realizar las validaciones necesarias para cada campo. Ej. la nota debe estar entre 0 y 10.

En caso de NO cumplir con los valores estipulados para cada campo se debe retornar un mensaje. Ejemplo : "Los datos ingresados no son válidos", caso contrario retornar la nota ingresada correspondiente con todos sus campos.

- b. Listar los datos de las notas ingresadas.

Ej:

```
[  
  { nombre: 'Juan', apellido: 'Perez', nota: 8 },  
  { nombre: 'Ana', apellido: 'Suarez', nota: 9 },  
  { nombre: 'Juan', apellido: 'Perez', nota: 5 },  
  { nombre: 'Pedro', apellido: 'Gomez', nota: 6 },  
  { nombre: 'Ana', apellido: 'Suarez', nota: 10 },  
]
```

- c. Listar los datos de las notas ingresadas agrupadas por alumno.

Ej:

```
[  
  { nombre: 'Juan', apellido: 'Perez', notas: [8,5] },  
  { nombre: 'Ana', apellido: 'Suarez', notas: [9,10] },  
  { nombre: 'Pedro', apellido: 'Gomez', notas: [6] },  
]
```

2. Crear endpoints para estadísticas:

- a. Listar todos los alumnos y sus promedios, cantidad de notas, nota mínima y nota máxima.

Ej:

```
[  
  { nombre: 'Juan', apellido: 'Perez', promedio: 6.50, cantidad: 2, minima: 5, maxima: 8 },  
  { nombre: 'Ana', apellido: 'Suarez', promedio: 9.50, cantidad: 2, minima: 9, maxima: 10 },  
  { nombre: 'Pedro', apellido: 'Gomez', promedio: 6, cantidad: 1, minima: 6, maxima: 6 },  
]
```

- b. Listar el promedio de todo el curso.

Ej:

```
{ promedio: 7.33 }
```

El servidor recibirá y responderá desde y hacia el frontend con los datos requeridos en formato JSON. En caso de inconvenientes, el servidor responderá con un objeto con un campo **'errorMsg'** informando el motivo de la falla. Todas las respuestas deberán estar correctamente adosadas con su código de estado correspondiente, según el resultado de la operación. En el caso de realizar notificaciones, modularizar adecuadamente. La notificación en sí puede simularse con un mensaje por terminal.

Aclaraciones sobre el desarrollo esperado:

1. El proyecto debe incluir únicamente el backend del sistema, utilizando Node.js + express. El formato del servidor es de tipo REST/RESTful. Tener en cuenta los lineamientos de dicho formato, especialmente a la hora de elegir los nombres de las rutas de acceso al sistema, los verbos que accionan sobre ellas, y cómo pasar datos adicionales a la consulta.
2. El sistema debe estar correctamente separado en capas y componentes, y esta separación debe estar claramente puesta de manifiesto en la estructura de carpetas y archivos. Entre los componentes que esperamos que estén presentes encontramos: router/controlador, casos de uso, modelo/s, DAO/s, repositorios, servicios de terceros, factories, commands (los que correspondan de acuerdo al sistema modelado y se hayan visto en cursada).
3. La *validación de datos* es una parte importante del negocio, por lo tanto, observar cómo y dónde realizarla.
4. *No es necesario utilizar una conexión a base de datos real*; es posible persistir en el DAO/Repositorio usando memoria del servidor.
5. Recordar el rol de las factories, que nos permiten desacoplar las dependencias de nuestros componentes a la hora necesitar una instancia de los mismos. Recordar esto especialmente a la hora de decidir cómo obtener los casos de uso para invocarlos desde la capa de ruteo.
6. Pueden reutilizar código de proyectos realizados durante el cuatrimestre, siempre y cuando el código se utilice y realmente aporte al desarrollo de las funcionalidades pedidas. **La inclusión de código innecesario o fuera de lugar será penalizada.**