

Examen UD5. Persistencia con SpringBoot.

Ejercicio único (100% calificación UD5 == RA 6)

Se trata de modificar la aplicación del proyecto GestionaAlumnosUd5 para que, en vez de guardar los datos de alumnos e emails en la variable *repositorioAlumno* implementada con un Map, lo haga en la base de datos mysql configurada en el archivo *application.properties*. Ten en cuenta:

- El proyecto está ya configurado con las dependencias Spring Web, Spring Data JPA, Mysql Driver y Spring Dev Tools, no se necesita ninguna otra dependencia y no se deben cambiar, no hace falta para ninguno de los cometidos. Ya tiene el *application.properties*
- El proyecto tiene creadas las clases Alumno e Email pero tendrás que modificarlas.
- No hay que tocar la clase AlumnoWebController, basta con modificar la clase AlumnoService y sustituir AlumnoMapRepository por repositorios en la base de datos. En AlumnoService debes mantener los nombres de los métodos, pues los invoca AlumnoWebController, pero sí deberás modificar su contenido para usar repositorios de base de datos (te sugiero borrar la variable alumnoMapRepository)
- Alumnos e Emails deben guardarse en tablas separadas.
- Los puntos de acceso son peticiones GET, ejemplos: (más pidiendo /)

<http://localhost:8080/nuevoAlumno?nombre=Maria+Luisa&apellidos=Sola>

<http://localhost:8080/nuevoAlumno?nombre=Lola&apellidos=Trapaza+Gomez>

<http://localhost:8080/alumnoPorId?id=2>

<http://localhost:8080/meteEmailAlumno?id=2&email=aprobando@examen.es>

<http://localhost:8080/buscaPorNombreYApellidos?nombre=Maria+Luisa&apellidos=Sola>

<http://localhost:8080/borraEmailDeAlumno?id=2&email=aprobando@examen.es>

Sobre esta base deberás conseguir que el proyecto funcione igual pero guardando los datos en la base de datos mysql ya definida en el proyecto. Te sugiero organizarte implementando las siguientes funcionalidades paso a paso:

Cometidos a realizar y su puntuación:

1. (1 punto) Correcto guardado de entidades Alumno en base de datos con su id autoincremental gestionado por la base de datos.
2. (0,75 puntos) Correcta modificación de entidades Alumno en base de datos.
3. (0,75 puntos) Correcta eliminación de entidades Alumno de la base de datos por id.
4. (0,75 puntos) Correcta definición de la asociación 1 a N entre Alumno y Email.
5. (1 punto) Correcta inserción de emails en base de datos manteniendo la relación con el alumno y los Emails con id autoincremental gestionado por BD.
6. (0,75 puntos) Correcto borrado de emails de la base de datos.
7. (0,5 puntos) Borrado automático de emails cuando se borra el Alumno propietario.
8. (1 punto) No cree un nuevo alumno si se intenta añadir uno con el mismo nombre y apellidos que otro ya existente. Tampoco modifique un alumno si el resultado de la modificación fuera un alumno ya existente (mismo nombre y apellidos)
9. (1 punto) Búsqueda de alumnos por nombre, por apellidos y por nombre y apellidos (puntos de entrada ya definidos /buscaPorNombre , /buscaPorApellidos y /buscaPorNombreYApellidos) La búsqueda debe hacerla la base de datos, no vale obtener todo y filtrar con código Java (como está ahora)
10. (1 punto) Listado ordenado por nombre y listado por apellidos (puntos de entrada ya definidos /todosPorNombre y /todosPorApellidos) La ordenación debe venir ya de la base de datos, no vale obtener todos y ordenar con código Java (como está ahora)
11. (1,5 puntos) Borra las tablas de la base de datos, importa la base de datos del fichero data.sql con el comando `mysql dwes5 -u daw2 -pLaElipa < data.sql` cambia en `application.properties` `spring.jpa.hibernate.ddl-auto=validate` y modifica el proyecto para que respete la estructura de la base de datos al iniciar la aplicación sin modificar su estructura pero siendo capaz de guardar y recuperar los datos en sus tablas (funcione todo como antes)