

Ejercicio integrador 3: Base de datos

Alumno: Alancay, Matias

Sistema para administrar un CRUD de alumnos, ingresando su nombre, apellido y nota. Para el desarrollo del mismo, usamos java y base de datos mySql. Además del CRUD, se agregaron dos funcionalidades para obtener el total de alumnos registrados y el promedio general de los alumnos registrados.

- Implementacion de las diferentes funcionalidades:

id	nombre	apellido	nota
13	Matias	Alancay	10.0
14	Jose	Velez	7.0
15	Alicia	Maravilla	9.0

Total Alumnos: 3 Promedio General 8.67

Agregar Alumno: Para agregar un nuevo alumno, debo primero completar los siguientes datos en los inputs correspondientes: nombre, apellido y nota. Luego de completar los datos, tocar en el botón de crear registro.

Actualizar alumno: Para actualizar un alumno existente, primero debo seleccionar un alumno de los que figuran en la tabla haciendo click sobre la fila. De esta manera, se cargaran los datos en los inputs automáticamente. Luego podre editar su información y guardarla tocando en actualizar registro.

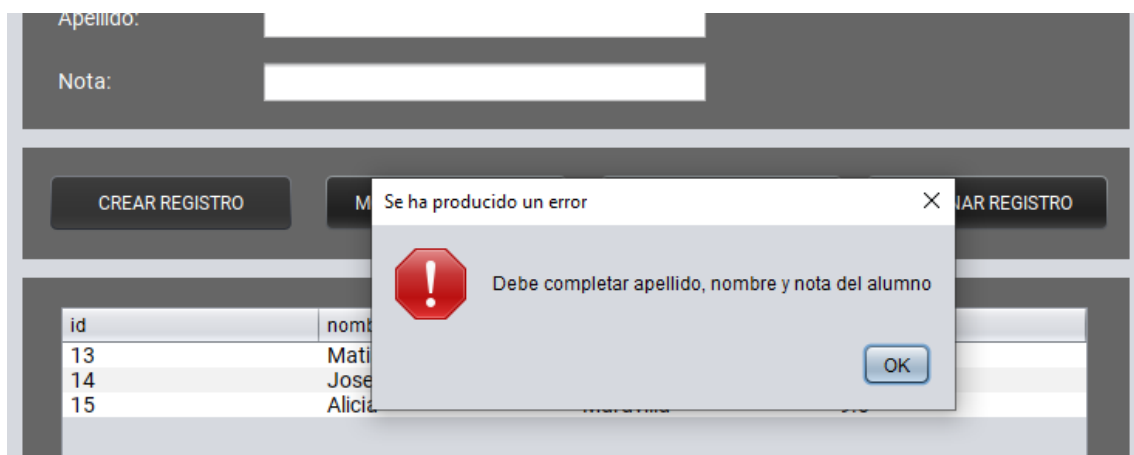
Mostrar alumnos: Para poder ver los alumnos registrados, solo debo tocar en el botón mostrar registros. Aunque al iniciar el sistema, ya se cargan los alumnos en la tabla y luego de cada operación CRUD realizada en la base de datos se vuelven a cargar para actualizar la tabla.

Eliminar alumno: Para eliminar un alumno existente, primero debo seleccionar un alumno de los que figuran en la tabla haciendo click sobre la fila. Una vez seleccionado el alumno, tocar en el botón eliminar registro.

- Lógica utilizada para validar los datos:

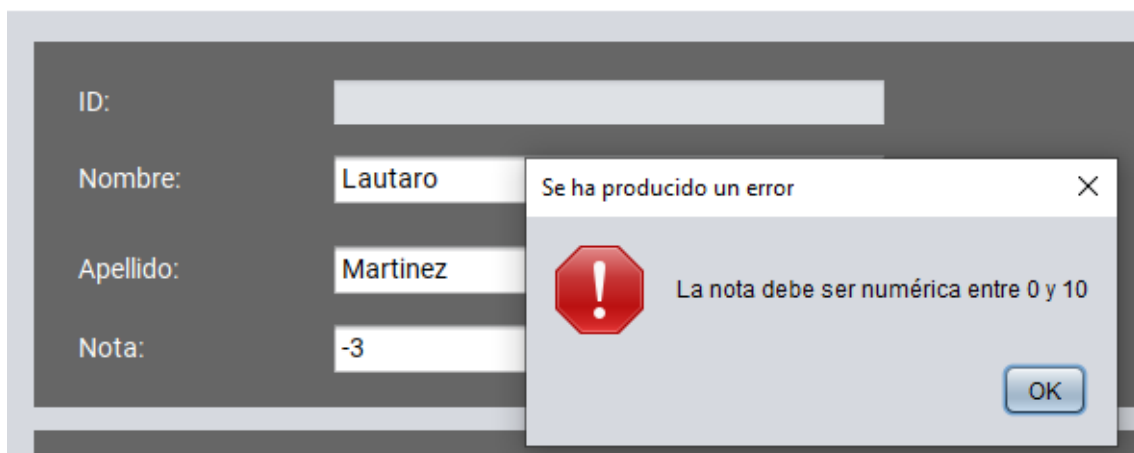
Tanto para agregar o modificar un alumno, validamos que se hayan ingresado todos los datos del mismo. En caso contrario, lanzamos una excepción propia que no se completaron los datos (`DatosIncompletosException`).

La forma de validar que se ingresaron todos los datos, es comprobar en código que los inputs de texto para ingresar los datos no están vacíos.



Además, validamos que la nota ingresada sea un valor correcto entre 0 y 10. Esto lo validamos mediante un condicional y en caso que no se cumpla lanzamos nuestra excepción `NotaIncorrectaException`.

Ejercicio 3



También validamos que en el caso de eliminar o modificar un alumno, debe seleccionar de los alumnos que figuran en la tabla. Para eso usamos un condicional en el que evaluamos que haya seleccionado una fila de la cual obtenemos los datos del alumno.

-Informe detallado de cualquier problema que hayan enfrentado durante el desarrollo de la aplicación y cómo los resolvieron.

Investigar sobre como implementar algunas consultas a parte del CRUD. Como usar count y avg para obtener el total de alumnos o el promedio general.

Terminar de implementar las demás funcionalidades CRUD en la capa DAO.

Investigar sobre propiedades adicionales para poder modificar los distintos componentes utilizados.