Trabajo Práctico Integrador

2023 – Segundo Cuatrimestre

# Enunciado

Se debe desarrollar un sistema de gestión académica. Este sistema permitirá gestionar información relacionada con materias, alumnos, profesores, ciclos lectivos y comisiones académicas. El objetivo principal es garantizar un proceso eficiente de inscripción de alumnos en comisiones, el registro de notas y la administración general de la institución.

# Descripción del problema

La institución educativa necesita una aplicación que le permita gestionar su oferta académica. Para ello, se han definido las siguientes clases y funcionalidades clave:

# Clases

**Materia**: Contiene información sobre las materias ofrecidas, incluyendo su identificación única (ID) y nombre.

**Alumno**: Representa a los estudiantes de la institución y almacena datos personales como nombre, apellido, fecha de nacimiento, fecha de ingreso, entre otros.

**Profesor**: Similar a la clase Alumno, pero para docentes.

**CicloLectivo**: Describe un período académico, con fechas de inicio y finalización, así como fechas para inscripciones.

**Comision**: Representa grupos de estudiantes que cursarán una materia en un ciclo lectivo específico. Tiene asignados profesores, alumnos, y registros de exámenes.

**Aula**: Define las aulas disponibles, con una capacidad máxima de alumnos.

# Funcionalidades

**agregarMateria**

//No se puede agregar 2 materias con mismo Id

**agregarAlumno**

//No se puede agregar 2 alumnos con mismo Dni

**agregarCicloLectivo**

**//**No se puede asignar 2 ciclos Lectivo con mismo Id y no se pueden superponer los rangos de fechas entre 2 ciclos distintos

**agregarComision**

//No se Pueden generar 2 Comisiones para la misma materia, el mismo cicloLectivo y el mismo turno

**agregarDocentes**

//No se puede agregar 2 docentes con el mismo dni

**asignarDocentesAComision**

//El mismo docente no puede ser profesor de la misma comisión 2 veces.

**agregarCorrelatividad(idMateria, idCorrelativa)**

//Se debe validar que ambos códigos existan en una materia

**eliminarCorrelatividad(idMateria, idCorrelativaAELiminar)**

//Se debe validar que esa correlatividad exista

**inscribirAlumnoAComision (dni, idComision)**

// Verificar que el alumno y la comisión estén dados de alta

//No se puede inscribir Alumnos si este no tiene aprobadas todas las correlativas. Se aprueba con 4 o más.

//La inscripción no se puede realizar si esta fuera de fecha Inscripción

//No se puede inscribir el alumno si excede la cantidad de alumnos permitidos en el aula

//No se puede inscribir el Alumno si ya está inscripto a otra comisión el mismo día y Turno

//No se puede inscribir a una materia que haya aprobado previamente

**asignarProfesorAlaComision(idComision, dniDocente)**

// verificar que exista la comisión y el docente

//Cada 20 alumnos se debe agregar un docente ejemplo de 1 a 20 alumnos solo se puede asignar un docente, de 21 a 40 2 docentes

**asignarAulaAlaComision(idComision, dniDocente)**

// verificar que exista la comisión y el docente

//Cada 20 alumnos se debe agregar un docente ejemplo de 1 a 20 alumnos solo se puede asignar un docente, de 21 a 40 2 docentes

**registrarNota (idComision, idAlumno, nota)**

// La nota debe estar entre 1 y 10

// No se puede asignar Una nota Mayor o igual a 7 si no están todas las correlativas aprobadas (Mayor o igual a 7)

//Las notas pueden ser de tipo 1erParc, 2doParc, Rec1Parcial, Rec2Parcial, Final

//no puede rendir 2 recuperatorios, es solo 1.

//para cargar la nota final, debe tener aprobadas las parciales

**obtenerMateriasAprobadasParaUnAlumno(idAlumno)**

Debe devolver un arreglo de materias

**obtenerNota(idAlumno, idMateria)**

**obtenerMateriasQueFaltanCursarParaUnAlumno(idAlumno)**

Debe devolver un arreglo de materias

**calcularPromedio(idAlumno)**

# Instrucciones

* Diseñar e implementar este sistema de gestión académica en el lenguaje Java
* Cumplir con los requerimientos funcionales
* Agregar las piezas que consideres necesarias para el correcto funcionamiento del sistema
* Realizar las validaciones necesarias para garantizar la integridad de los datos.
* Crear un programa de prueba que demuestre el funcionamiento, asegurando que los casos de uso del flujo normal se cumplan, y algunos flujos alternativos.

# Entrega

La entrega del trabajo debe incluir el código fuente en github, cada integrante debe crear su propia rama para trabajar, y la entrega final se hace en la rama master/main. Se debe adjuntar un link del repositorio por MiEL

La fecha de entrega para este trabajo práctico es 27/09/2023