**PROYECTO FINAL: SISTEMA DE GESTIÓN DE ACADEMIA DE MÚSICA**

La Academia UQ Música desea modernizar su sistema para administrar estudiantes, profesores, cursos por instrumento (piano, guitarra, violín, canto, etc.), y la programación de clases individuales y grupales. Se requiere un sistema que permita gestionar matrículas, horarios, asistencia y reportes de progreso por instrumento y nivel. La solución debe ofrecer una interfaz clara y segura para cada rol, soportar reglas de negocio (capacidad de grupos, conflictos de horario, prerrequisitos de nivel) y producir reportes útiles para dirección académica.

## **Conceptos de Programación Orientada a Objetos Aplicados**

Para garantizar una arquitectura flexible y reutilizable, el sistema implementará los siguientes conceptos de **Programación Orientada a Objetos (POO):**

* **Herencia:** Se plantea como un mecanismo que favorece la reutilización y organización del código, permitiendo establecer jerarquías entre entidades relacionadas dentro del sistema.
* **Interfaces:** Constituyen una forma de definir contratos de comportamiento comunes, que pueden ser implementados por distintas clases según sus necesidades particulares.
* **Polimorfismo:** Brinda la posibilidad de que un mismo método se adapte a diferentes contextos, facilitando la extensión y la flexibilidad de las operaciones en el sistema.
* **Encapsulamiento:** Actúa como una estrategia de protección y control del acceso a la información, resguardando los datos internos de las entidades y manteniendo la coherencia de los procesos.
* **Abstracción:** Permite modelar las características esenciales de los elementos del sistema, ocultando los detalles innecesarios y proporcionando una visión clara y general de las entidades principales.

## **Roles y funcionalidades**

### **Estudiante**

* Registro y actualización de datos personales.
* Inscripción a cursos (por instrumento y nivel) según cupos y prerrequisitos.
* Consulta de horario y asistencia.
* Descarga de reportes de progreso por curso/nivel.
* Solicitud de clases individuales (según disponibilidad del profesor).

### **Profesor**

* Gestión de horarios disponibles (bloques de tiempo).
* Creación y edición de clases grupales (cupos, aula, instrumento, nivel).
* Registro de asistencia y valoración de progreso (criterios por instrumento/nivel).
* Comentarios formativos y recomendaciones de práctica.

### **Administrador académico**

* Modificación de estudiantes y profesores.
* Creación de cursos (instrumento, nivel, capacidad, prerrequisitos).
* Gestión de aulas y conflictos de horario (no solapar profesor/aula).
* Reportes: asistencia por curso, progreso por instrumento/nivel, ocupación de aulas, carga docente.

## **Reglas de negocio clave**

1. **Capacidad de grupo:** una clase grupal no supera su cupo.
2. **Conflictos de horario:** un profesor y un aula no pueden tener dos clases en el mismo bloque.
3. **Prerrequisitos:** para inscribir un nivel N, el estudiante debe aprobar el nivel N-1 (o validación).
4. **Clases individuales:** creadas bajo demanda; no ocupan disponibilidad del profesor.
5. **Cierre de ciclo:** cada curso debe registrar al menos una evaluación de progreso antes de emitir certificado.

## **Funcionalidad Extendida (Opcional)**

* **Notificaciones por correo**: recordatorios de clase y avisos de cambio de horario.
* **Adjuntos**: subir plan de estudio, partituras, enlaces de práctica.
* **Exportación**: reportes en PDF/CSV.
* **Persistencia**: archivos.

## **Entrega del Proyecto**

Debe incluir:

* **Documento con el análisis del pensamiento computacional.**
* **Diagrama de clases** (en formato PDF o PNG).
* **Enlace al repositorio del proyecto** (control de versiones debe tener 15 commits por integrante del grupo).
* **Uso de herencia, interfaces, polimorfismo y encapsulamiento.**
* **Pruebas unitarias** para todas las clases.
* **Interfaz gráfica de usuario (JavaFX).**

### **Notas a Tener en Cuenta**

* La entrega debe ser realizada por **un solo** integrante del grupo.
* Incluir un comentario con los nombres de todos los integrantes del grupo.
* Todos los integrantes deben estar presentes en la sustentación.
* No se aceptan trabajos después de la hora de cierre.

### **Evaluación**

La nota del proyecto consta de dos partes:

1. **Codificación (0 a 5 puntos).**
2. **Sustentación (factor multiplicador de 0 a 1).**
   * La persona seleccionada para sustentar influirá en la calificación final del grupo.

El resultado final se obtiene multiplicando la nota de codificación por la evaluación de la sustentación.

**Recuerden:** La sustentación es obligatoria para todos los integrantes del grupo y el profesor elegirá al sustentante en el momento de la presentación.