**Pensamiento Computacional**

**¿Qué se solicita finalmente (problema)?** La Academia UQ Música necesita modernizar su sistema de gestión académica, para gestionar estudiantes, profesores, cursos por instrumento y nivel, junto con la programación de clases individuales y grupales. El problema central es la falta de un sistema unificado que permita gestionar matrículas, horarios, asistencia, evaluaciones, informes y control de reglas de negocio, de manera segura y accesible para cada rol (estudiante, profesor, administrador académico).

**¿Qué información es relevante dado el problema anterior?**

* **Estudiantes:** datos personales, cursos inscritos, asistencia, progreso, solicitudes de clases individuales.
* **Profesores:** disponibilidad de horario, clases asignadas, registro de asistencia, evaluaciones, comentarios.
* **Cursos:** instrumento, nivel, cupo, prerrequisitos, aulas asignadas, horarios y lista de estudiantes.
* **Aulas:** identificador, capacidad, disponibilidad.
* **Clases:** tipo (individual o grupal), horario, profesor, aula, lista de inscritos.
* **Reportes:** asistencia, progreso académico, carga docente, ocupación de aulas.
* **Reglas:** capacidad de grupo, conflictos de horario, cumplimiento de prerrequisitos, registro mínimo de evaluaciones antes del cierre del ciclo.

**¿Cómo se agrupa la información relevante?** La información se agrupa de acuerdo con las entidades principales del sistema y sus relaciones:

* **Persona (clase base):** atributos comunes como nombre, identificación, correo, etc.
* **Estudiante (hereda de Persona):** cursos inscritos, historial académico, progreso.
* **Profesor (hereda de Persona):** horarios disponibles, clases a cargo, evaluaciones.
* **Curso:** instrumento, nivel, cupo, prerrequisitos.
* **Clase:** instancia de un curso con profesor, aula, horario y lista de estudiantes.
* **Administrador Académico:** gestiona cursos, profesores, aulas y reportes.
* **Aula:** nombre, capacidad, disponibilidad.
* **Reporte:** genera estadísticas de asistencia, progreso y ocupación.

**¿Qué funcionalidades se solicitan?**

* **Estudiante:** registro/actualización, inscripción a cursos (según cupo y prerrequisitos), consulta de horario/asistencia, descarga de informes de progreso, solicitud de clases individuales.
* **Profesor:** gestión de disponibilidad, creación/edición de clases grupales, registro de asistencia y notas, comentarios de seguimiento.
* **Administrador académico:** creación de cursos, asignación de aulas y profesores, resolución de conflictos de horario, modificación de datos, generación de informes institucionales.
* **Sistema (general):** control de capacidad, verificación de prerrequisitos, prevención de solapamiento de horarios, cierre de ciclo con evaluación registrada.
* **Extras opcionales:** notificaciones por correo, carga de archivos (plan de estudio, partituras), exportación de informes en PDF/CSV, persistencia en archivos.

**¿Cómo se distribuyen las funcionalidades?** Cada rol tiene una interfaz propia y funcionalidades específicas según sus permisos:

* Los **estudiantes** interactúan con el sistema desde un panel académico donde pueden inscribirse, ver su progreso y asistir a clases.
* Los **profesores** gestionan su agenda, crean clases grupales y evalúan a los estudiantes.
* El **administrador académico** controla todo el sistema: crea cursos, asigna profesores, válida reglas y genera informes.
* El **sistema base** se encarga de aplicar las **reglas de negocio** , verificando cupos, horarios y prerrequisitos automáticamente.

**¿Qué debes hacer para probar las funcionalidades?**

1. **Casos de prueba funcional:**

* Registrar un estudiante y verificar que sus datos se guardan correctamente.
* Intentar inscribir a un curso sin cumplir prerrequisitos → el sistema debe rechazarlo.
* Crear una clase grupal con más estudiantes que su cupo → debe mostrar error.
* Asignar un profesor a dos clases en el mismo horario → el sistema debe impedirlo.
* Registrar asistencia y progreso, luego generar un reporte → debe reflejar los datos reales.

1. **Pruebas de integración:**

* Validar que los cambios de horario de una clase actualicen el reporte del aula y del profesor.
* Confirmar que la creación de un curso actualiza la lista de inscripción disponible para los estudiantes.

1. **Pruebas de interfaz (UI):**

* Comprobar que cada rol solo vea sus funciones.
* Probar accesos inválidos o intentos de edición de datos no autorizados.

1. **Pruebas de persistencia:**

* Guardar y recuperar información desde archivos (matrículas, informes, evaluaciones).