**LAB 4**

**Analisis y diseño**

**1. ¿Qué debe hacer el programa? Haga una lista de las opciones que brindará al usuario en la interacción con su sistema:**

**Cargar Archivo de Salones:**

* Permite al usuario cargar un archivo CSV que contiene información sobre salones, como ID de sede, edificio, nivel, ID de salón y capacidad.

**Cargar Archivo de Cursos:**

* Permite al usuario cargar un archivo CSV que contiene información sobre cursos, incluyendo ID de curso, ID de sede, nombre del curso, horario, duración, días y cantidad de estudiantes.

**Consultar Salón:**

* Permite al usuario buscar y mostrar detalles específicos de un salón en función de la sede, edificio, nivel y ID del salón proporcionados.

**Mostrar Detalles del Curso:**

* Permite al usuario buscar y mostrar detalles específicos de un curso en función de su ID de curso.

**Asignar Salones:**

* Realiza la asignación automática de salones a cursos según ciertos criterios, como capacidad y sede.

**Generar Informe:**

* Muestra un informe que lista los cursos asignados y los cursos no asignados, junto con sus detalles.

**Exportar Resultados:**

* Permite al usuario exportar los resultados (información sobre cursos asignados y no asignados) a un archivo CSV.

**Clases Necesarias:**

AsignadorSalones: Esta clase principal gestiona todo el funcionamiento del programa. Sus responsabilidades incluyen cargar archivos de salones y cursos, consultar detalles de un salón específico, consultar detalles de un curso específico, asignar salones a cursos según ciertos criterios, generar un informe y exportar resultados. Veamos sus propiedades y métodos:

**Propiedades:**

* salones (List<Salon>): Almacena la lista de objetos de salones cargados desde el archivo.
* cursos (List<Curso>): Almacena la lista de objetos de cursos cargados desde el archivo.
* cursosAsignados (List<Curso>): Almacena la lista de cursos a los que se les ha asignado un salón.
* cursosNoAsignados (List<Curso>): Almacena la lista de cursos que no han sido asignados a ningún salón.

**Métodos:**

* cargarArchivoSalones(String nombreArchivo): Carga los datos de salones desde un archivo CSV.
* cargarArchivoCursos(String nombreArchivo): Carga los datos de cursos desde un archivo CSV.
* consultarSalon(int idSede, int idSalon, char edificio, int nivel): Consulta y muestra los detalles de un salón específico.
* mostrarDetallesCurso(int idCurso): Muestra los detalles de un curso específico.
* asignarSalones(): Asigna salones a cursos según ciertos criterios y actualiza las listas cursosAsignados y cursosNoAsignados.
* generarInforme(): Genera un informe mostrando los cursos asignados y no asignados.
* exportarResultado(String nombreArchivo): Exporta el informe a un archivo.
* Salon: Esta clase representa un salón en la institución educativa. Sus propiedades y métodos son:

**Propiedades:**

* idSede (int): Identificador de la sede a la que pertenece el salón.
* edificio (char): Letra que representa el edificio en el que se encuentra el salón.
* nivel (int): Nivel o piso en el que se encuentra el salón.
* idSalon (int): Identificador único del salón.
* capacidad (int): Capacidad máxima de estudiantes en el salón.
* horarios (Map<String, String[]>): Un mapa que almacena los horarios disponibles del salón por día de la semana.
* Curso: Esta clase representa un curso en la institución educativa. Sus propiedades y métodos son:

**Propiedades:**

* idCurso (int): Identificador único del curso.
* idSede (int): Identificador de la sede a la que pertenece el curso.
* nombreCurso (String): Nombre del curso.
* horario (int): Horario del curso.
* duracion (int): Duración del curso en semanas.
* dias (String): Días de la semana en los que se imparte el curso.
* cantidadEstudiantes (int): Cantidad de estudiantes inscritos en el curso.
* salonAsignado (Salon): Referencia al salón asignado al curso (puede ser nulo si no se ha asignado).

UML

