

## **Proyecto Programado 1**

### **Reservación de salas de estudio**

## **1. Objetivos del proyecto**

### **Objetivo general**

Desarrollar una solución de software para practicar acerca de los conceptos de estructuras de datos lineales.

### **Objetivos específicos**

- Modelar un sistema basado en un conjunto de requerimientos utilizando la técnica divide y vencerás.
- Profundizar el conocimiento en el lenguaje de programación C.
- Desarrollar habilidades de investigación en temas relacionados con:
  - Persistencia de datos mediante manejo de archivos.
  - Interfaces gráficas de usuario.
- Reconocer e implementar buenas prácticas/recomendaciones en el diseño de interfaces gráficas de usuario.
- Fomentar el trabajo en equipo, el desarrollo de habilidades de liderazgo, planeamiento y comunicación efectiva.

## **2. Contexto y las funcionalidades esperadas**

La biblioteca del Tec cuenta con una gran cantidad de salas de estudio a disposición de los estudiantes, pero no existe ningún mecanismo automático que gestione tales recursos. Por lo tanto, se desea construir una solución de software que facilite, la administración del proceso de asignación y monitoreo de salas de estudio a los estudiantes del Tec.

Para que un estudiante pueda utilizar este nuevo servicio, debe estar registrado en el sistema, para registrarse se le solicitará el nombre completo, su carnet, carrera, dirección de correo electrónico, calificación (por omisión con calificación de 100) y un número telefónico.

Cada sala está caracterizada por:

- Identificador de la sala (identificador único, formato "SAL-001")
- Ubicación, que corresponde a una cadena de caracteres.
- Capacidad máxima.
- Recursos (ejemplo: Proyector, Pizarra, A/C, entre muchos otros)
- Estado (Activa, Mantenimiento, Inactiva, Reservada, Por omisión estado Activa)
- Calificación (por omisión con calificación de 100)

Es importante mencionar que cada sala tiene asociada una agenda de servicio, que básicamente contiene horarios, es decir los días de la semana que la sala está disponible, así como las horas de apertura y cierre para cada día.

El sistema debe permitir la creación de reservas de sala, para lo cual se requiere conocer el asunto que será tratado, id de la sala (número consecutivo), estado (por omisión activa, pudiendo pasar a cancelada) así como la sala que se desea reservar, la lista de participantes en la actividad y la fecha, indicando el día y las horas de inicio y finalización. Se puede dar que varios estudiantes reserven la sala con choque de horario, para esto en la funcionalidad de atención de reservaciones, se atenderán dichas reservaciones según la prioridad asignada.

El estudiante no podrá -bajo ninguna circunstancia- reservar salas, si su calificación es 70 o menos, si tiene cinco o más incidencias o si la cantidad de participantes es mayor a la capacidad de la sala seleccionada.

Es del interés del Tec fomentar una cultura que maximice el uso y el cuidado de los recursos disponibles, por lo tanto, el sistema debe permitir el registro de incidentes, como podrían ser: sala reservada no utilizada, ruido excesivo, basura y desorden. Cada uno de ellos implica la reducción de la calificación del estudiante que organiza la reunión en razón de 15, 10 y 7 puntos respectivamente.

La aplicación debe mantener una bitácora de todas las reservas que han sido procesadas exitosamente. El principal interés es poder obtener información relevante al comportamiento de los recursos y poder realizar ciertos procesos de análisis de datos que apoyen la toma de decisiones futuras.

Además, la solución de software debe proveer persistencia de datos mediante archivos, la estructura de cada archivo es decisión del equipo.

## 2.1 Funcionalidades

Se contará con las siguientes funcionalidades:

1. **Agregar o registrar estudiante:** El sistema debe permitir el registro de nuevos estudiantes. Todos los datos del estudiante son requeridos.

Aspectos a considerar:

- El sistema no debe permitir la existencia de estudiantes con un mismo número de carnet.
- La aplicación debe tener pre-cargados al menos 10 estudiantes en el sistema.

2. **Consultar estudiante:** El sistema debe permitir la consulta de la información de un estudiante, así como la lista de reservas que ha realizado y los incidentes que pudieran haberse registrado.

3. **Agregar sala:** El sistema debe permitir el registro de nuevas salas de reunión. Todos los datos de la sala son requeridos. Los horarios disponibles para las salas deben encontrarse registrados. En estos horarios se detalla los días y las fechas de inicio y cierre. Cuando se cree la sala, el usuario decidirá cuales son los horarios de operación que tendrá la sala.

Aspectos a considerar:

- El sistema no debe permitir la existencia de salas con un mismo número de sala.
- La aplicación debe tener pre-cargados al menos 10 salas en el archivo fuente.
- La aplicación debe tener pre-cargados al menos 6 horarios de operación.

4. **Modificar datos de una sala:** El sistema debe permitir la modificación de los siguientes datos de una sala: ubicación, recursos y estado.

5. **Calificar sala:** El sistema debe permitir que se califique el estado general de una sala, para tal fin se solicitará al estudiante el ingreso del código de sala, el número de reservación y una calificación entre 1-100. La calificación de una sala, corresponde al promedio de calificaciones del total de reservas realizadas para la sala donde se registró calificación.

6. **Consultar sala:** El sistema debe permitir la consulta de la información de una sala, incluyendo sus horarios de operación, así como sus reservas dentro los 7 días siguientes a la fecha del sistema.

7. **Crear una reserva:** El sistema debe permitir la creación de reservas indicando la fecha de su solicitud, la hora de inicio y fin, la capacidad mínima requerida y opcionalmente algún recurso que deba tener la sala. El sistema le mostrará aquellas salas que cumplen con la condición de búsqueda y el usuario seleccionará la sala de su elección, seguidamente completará la información del tema y los participantes. El sistema aleatoriamente le asignará una prioridad a esta reservación, las prioridades pueden ser ALTA, MEDIA o BAJA. Este dato será utilizado en el momento de atender las reservaciones.

Aspectos a considerar:

- La aplicación debe dar persistencia a las reservas mediante los archivos.
8. **Consulta de reservas para un estudiante en particular:** El sistema debe permitir que el usuario consulte todo el histórico de reservas que ha realizado un estudiante, para tal fin debe solicitar un número de carnet. Y el historial debe mostrarse de forma cronológica mostrando primero las reservaciones más recientes.
  9. **Cancelar reserva:** El sistema debe permitir que el usuario cancele una reserva que se encuentre al menos a una hora antes de su hora de inicio. En tal caso se debe actualizar el estado de la reserva a “cancelada”. Por cada cancelación de sala se restará 1 a la calificación del estudiante organizador.
  10. **Atención de reservas:** El sistema debe permitir que un usuario atienda las reservas realizadas en el sistema, para esto debe buscar todas las reservas que están con estado “En proceso”, el sistema debe atender dichas reservas por orden de prioridad, en caso que no haya choque de horario de una reserva en la misma sala, debe cambiar el estado de la reserva a “Aprobada” y el estado de la sala correspondiente debe quedar en estado “Reservada”, en caso que haya choque de horario con una reserva que ya está aprobada para la misma sala, se debe cambiar el estado de la reserva a “Rechazada”. Se debe visualizar en el sistema la información de cada reserva que se va atendiendo.
  11. **Registrar incidentes:** El sistema debe permitir registrar los incidentes asociados a una sala, para registrar el incidente se debe indicar el código de la sala, el código de reservación, la descripción del incidente y se debe obtener la fecha del sistema.
  12. **Consultar incidentes:** El sistema debe permitir que el usuario consulte los incidentes, ya sea por un rango de fechas o por el código de sala. Al ejecutar esta consulta se debe mostrar el listado de todos los incidentes y el usuario puede seleccionar uno para ver el detalle del mismo.
  13. **Análisis de datos:** El sistema debe permitir que el usuario realice una serie de consultas sobre los datos, donde la información de estas consultas se desplegará de mediante diferentes formatos de gráficos:

Top 5 de las salas más utilizadas  
Top 5 de los horarios más utilizados  
Top 5 de las carreras que más utilizan las salas  
Top 5 de las salas con mayor puntuación

Debe considerar esta funcionalidad como parte del menú de reportes de la aplicación.

## 2.2 Estructuras de datos

Para el estudio de las estructuras de datos, se deberán implementar las siguientes cuatro estructuras: Lista enlaza simple, lista doblemente enlazada, pila, cola y cola de prioridad. La implementación debe ser con estructuras de datos dinámicas mediante el uso de punteros. Para el proyecto cuando requiera una estructura de datos, solo podrá utilizar sus implementaciones de las estructuras mencionadas. Deben seleccionar la estructura que más se adecúa a la funcionalidad planteada en el sistema.

**Defensa:** En la evaluación existen puntos destinados a la defensa del uso de las estructuras de datos. Deberán defender por qué para una determinada funcionalidad se usó una de las estructuras de datos disponibles. Si otra estructura de datos es más eficiente, se bajarán puntos en el rubro.

## 14. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación C y debe tener una interfaz gráfica.

En caso de requerir librerías externas adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación.

El sistema debe tener persistencia de datos por medio de archivos.

## 15. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, y restricciones.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: instrucciones de compilación, ejecución y uso de todas sus funcionalidades.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema: ¿Qué se tiene que resolver y cómo se va hacer?
5. Diseño del programa: decisiones de diseño, uso de estructuras de datos, uso de librerías. Justifique todas sus decisiones.
6. Librerías usadas.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).

## 16.Evaluación

1. Documentación interna 3%.
2. Documentación externa 12%.
3. Implementación, manejo y defensa de estructuras de datos 15%.
4. Agregar o registrar estudiante 5%
5. Consultar estudiante 3%
6. Agregar sala 5%
7. Modificar datos de una sala 4%
8. Calificar sala 4%
9. Consultar sala 3%
10. Crear una reserva 7%.
11. Consulta de reservas para un estudiante en particular 5%
12. Cancelar reserva 4%
13. Atención de reservas 10%.
14. Registrar incidentes 5%.
15. Consultar incidentes 3%.
16. Análisis de datos 12%.

## 17.Aspectos administrativos

1. Debe crear un archivo .zip ("PP1.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **solucion\_computacional**, en la primera deberá incluir el documento de *word* o *open office* (no pdf) solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo **info.txt** debe contener la siguiente información (cualidades):
  - a. Nombre del curso
  - b. Número de semestre y año lectivo
  - c. Nombre del Estudiante x2
  - d. Número de carnet x2
  - e. Número de proyecto programado
  - f. Fecha de entrega
  - g. Estatus de la entrega (definido por el responsable de la implementación de la tarea, debe ser **CONGRUENTE** con la solución entregada):  
[Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]
2. Deberá subir el archivo antes mencionado al TEC Digital en el curso de ALGORITMOS Y ESTRUCTURAS DE DATOS GR 01, en la asignación llamada "Proyecto 1" debajo del rubro de "Proyectos". El proyecto debe entregarse a más tardar el **jueves 25 de junio a las 11:55pm**

3. Dentro de la carpeta de **solucion\_computacional**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2.txt**) que contenga todo el texto de la solución del o de los archivos .c presentados (las implementaciones). Este archivo puede ser revisado en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original, y no se va a tolerar el plagio.** Este archivo también deberá ser subido al TEC Digital en el curso de ALGORITMOS Y ESTRUCTURAS DE DATOS GR 01, en la asignación llamada “Proyecto 1 (archivo TXT)” debajo del rubro de “Proyectos”. Se deberá subir el archivo siguiendo los mismos rangos de tiempo que los descritos en el punto anterior.
4. Las tareas deberán ser revisadas con el profesor o el asistente. Todos los miembros del grupo deberán participar de la revisión, de lo contrario a la nota obtenida grupalmente se le descontará 15%. Además, si durante la revisión el estudiante no puede demostrar conocimiento de la solución presentada, se le descontará hasta un 15% de la nota grupal.
5. La tarea se hará en grupos de 2 personas.
6. La tarea vale un 20% de la nota del curso.
7. Antes de la entrega del proyecto deben enviar un correo electrónico con el asunto **“[Algoritmos 220] PP1:GRUPO”** e indicar los nombres de las personas que conforman el grupo de trabajo para este proyecto programado. Esta información es necesaria para crear los grupos en el TEC Digital. Si no envían el correo, se calificará el proyecto de forma individual al miembro del equipo que la suba.

## 18. Condiciones

Este proyecto programado se rige por las siguientes condiciones:

1. El desarrollo del proyecto es estrictamente en grupos de 2.
2. Debe cumplir con todo lo indicado en las secciones de “Funcionalidad”, “Estructuras de Datos”, “Aspectos técnicos” y “Documentación”.
3. Deberá entregarse en tiempo y forma según el plazo establecido por el profesor al momento la lectura de este documento.
4. El lenguaje de programación a utilizar es C.

Nota: El incumplimiento de alguna condición implicará una calificación de cero.

**IMPORTANTE: CONOCIMIENTO DE LA SOLUCIÓN PRESENTADA.** En la revisión del trabajo, los estudiantes deben demostrar un completo dominio de la solución implementada, tanto desde el punto de vista técnico (uso de herramientas) como de la funcionalidad del proyecto. La revisión se puede hacer individualmente o en grupos, examinando la solución o temas específicos aplicados en el proyecto. Todos los integrantes del grupo deben tener el mismo conocimiento de la solución presentada. Recuerde que de no contestar las preguntas durante la revisión, se puede descontar hasta un 15% de la nota obtenida de forma grupal.