



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS
CIENCIAS DE LA COMPUTACIÓN 1
PROYECTO FINAL 2021-3

Las directivas de una Prestigiosa Universidad, concretamente la decanatura de la facultad de Ingeniería requiere el diseño de un Sistema que administre la información de inscripción de los estudiantes en las asignaturas que se ofrecen semestre a semestre, permitiendo que los estudiantes elaboren sus propios horarios y controlando las medidas de bioseguridad para prevenir contagios de COVID - 19.

La información manejada para una asignatura es la siguiente:

- Programa en que se dicta,
- Código
- Nombre
- Créditos
- Intensidad horaria
- si se dicta o no el próximo semestre
- tipo (pregrado o postgrado)
- semestre al que pertenece

Toda asignatura tiene asociado uno o más grupos. De cada grupo se tiene:

- Identificador
- Nombre del profesor que lo dicta
- el cupo máximo (número máximo de estudiantes que la pueden inscribir)
- los días en que se dicta, las horas y los salones. Vale la pena resaltar que los salones y las horas en las que se dicta una sección pueden variar de acuerdo a los días. Un grupo se puede dictar entre 2 y 4 veces por semana. Por ejemplo, Redes de Computadores puede dictarse el lunes de 2-4 pm en el salón 2-208, el miércoles de 9-10 am en el salón 3-309 y el viernes de 7-8 am en el salón 52-506.
- Capacidad del salón

El programa debe permitir el registro de los grupos de cada asignatura que se vaya a dictar, asociando a cada grupo toda la información de salones y horarios y garantizar que no haya cruce en los horarios de las asignaturas de un profesor.

De un estudiante el Sistema debe manejar la siguiente información:

- código,
- nombre
- edad

- e-mail
- programa al cual pertenece
- tipo (pregrado o postgrado)
- Semestre.
- Fechas de vacunación de cada una de las dosis
- Tipo de vacuna recibida

De los profesores se debe contar con la siguiente información:

- código,
- nombre
- e-mail
- programas en los cuales imparte clase
- Fechas de vacunación de cada una de las dosis
- Tipo de vacuna recibida

Los funcionarios de la administración podrán agregar una asignatura, un grupo, un profesor o un estudiante al Sistema.

Durante el periodo de inscripciones los estudiantes podrán definir su horario de clases, con la posibilidad de adicionar, retirar o cambiar de grupo de cualquier asignatura. El sistema solo muestra las asignaturas que se dictan en el próximo semestre y no debe permitir que el estudiante se inscriba en un grupo que no tiene cupo o que presenta cruce con alguna otra asignatura ya inscrita por el estudiante.

Estructuras de datos:

En memoria secundaria:

La información deberá estar almacenada, de manera permanente, en archivos los cuales deben ser actualizados al final del día o en su defecto al terminar la utilización del programa. Debe tenerse la posibilidad de modificar la información ya sea agregando, eliminando o cambiando registros.

Existen por lo menos los archivos planos para los estudiantes, asignaturas, grupos y profesores.

En memoria Principal:

Con el fin de optimizar las búsquedas deberán utilizarse listas, multilistas, arreglos, pilas, colas o árboles en memoria principal y para la realización de dichas búsquedas, deberá

evitarse la utilización total de la información. Es decir, se espera que apliquen los criterios y conocimientos correspondientes a la eficiencia de los algoritmos. Complejidad en tiempo y memoria. En cuanto a listas se dan ejemplos de la forma como se pueden estructurar:

Nodo asignatura:

Los campos con la información de cada asignatura			
sigPro grama	SigCreditos	SigIntensidadHoraria	SigSemestre

Las cabezas de las listas que se deben crear son:

Cabezas de Lista

Cabeza por programa				
Sistemas	Electrónica	Industrial	Eléctrica	Catastral

Cabeza por semestre									
1	2	3	4	5	6	7	8	9	10

Cabeza por Intensidad Horaria semanal					
2	3	4	5	6	

Cabeza por créditos			
1	2	3	4

Consultas.

El programa debe permitir a las directivas de la universidad, obtener la siguiente información

1. Número total de asignaturas de un programa dado, evidenciando si es de pregrado o posgrado, clasificándolas por semestre al que pertenece y número de créditos.
2. Listado de TODAS las asignaturas que ofrece un programa en el próximo semestre, indicando el código, nombre, horario (días, horas y salones en que se dicta) y cupos disponibles.
3. Listado de estudiantes que están inscritos en un grupo de una asignatura y cuántas vacunas tiene aplicadas.
4. En qué asignaturas está inscrito un estudiante dado, cuáles vacunas tiene aplicadas y en qué fecha se aplicó cada una.
5. Obtener número de estudiantes vacunados en la facultad con una vacuna dada, clasificándolos por programa, tipo de programa, semestre y fecha en que se aplicó la última dosis.
6. Listado de profesores que imparten clase en un programa dado, mostrando las asignaturas que dicta, el tipo de vacuna que se ha aplicado y la fecha de aplicación de cada una de ellas.
7. Listado de profesores y estudiantes que tienen más de 6 meses de su segunda dosis y no han recibido la tercera, indicando su cargo, clasificados por programa.

8. Dado un día y una hora, mostrar el listado de los salones que se encuentran ocupados, mostrando el nombre de la asignatura, el programa al que pertenece, el nombre del profesor, la capacidad máxima del salón y el número de estudiantes inscritos en la asignatura.
9. Dada una vacuna, mostrar el listado de estudiantes y profesores que han sido inoculados con ella, indicando la fecha y en cuál o cuáles dosis fue recibida.

Simulación de Inscripción

- Creará las estructuras en memoria principal a partir de los archivos planos.
- Permitirá modificar los datos en memoria principal. Retirar un registro, agregar uno nuevo, modificar uno existente.
- Simulará el proceso de inscripción de materias, generando aleatoriamente la cantidad de estudiantes que son asignadas a cada grupo. La suma total no puede superar el número máximo de estudiantes por grupo ni el aforo máximo de cada salón. El proceso debe permitir que sea posible agregar más grupos, cuando la decanatura así lo determine.
- Simulará el proceso de inscripción, generando aleatoriamente los estudiantes inscritos, la asignatura y el grupo (dependiendo del semestre En el que se encuentre) así como el tipo de vacuna que le fue aplicada en cada dosis, teniendo en cuenta que un estudiante que no haya completado su esquema de vacunación, no podrá asistir a clases.
- Para cada EPS mostrará los siguientes resultados:

Por cada programa: número de vacunados con cada tipo de vacuna y su porcentaje. Número de estudiantes y/o profesores que no han completado su esquema de vacunación.

Por cada grupo: número de estudiantes vacunados, clasificados por Tipo de vacuna, rango de edad y fecha de cada dosis.

- **Estadísticas:**

- ✓ Reporte por programa: total de estudiantes vacunados clasificados por semestre, tipo de vacuna y fecha de la segunda dosis.
 - ✓ Reporte por asignatura: Totales por cada grupo clasificados por ciudad y sexo.
 - ✓ Total, por facultad: Totales por tipo de vacuna, clasificados por programa, rango de edad y semestre.
- Finalizada la simulación actualizará los archivos planos con los cambios realizados en memoria principal. Lo cual permitirá nuevos procesos de simulación. Tenga en cuenta que una vez iniciada la simulación de la inscripción de materias, no podrá eliminarse ni agregarse ninguna materia, grupo, programa o estudiante.

Requisitos funcionales:

- Con el fin de optimizar las búsquedas deberán utilizarse listas y la multilista definida. Las cuales se construyen al iniciar el programa y a partir de la información almacenada en los archivos.

- Debe garantizar la eficiencia en las búsquedas aprovechando la estructura de datos definida.
- El programa contará con los menús y submenús que faciliten la realización de cada una de las consultas y la simulación.
- Adicionalmente, las listas deben crearse de manera genérica, de tal forma que puedan agregarse fácilmente otras consultas adicionales por parte de la decanatura.
- Para la entrega y sustentación final deben incluir los archivos planos indispensables con no menos de 20 registros en el archivo de estudiantes en cada grupo, las carreras definidas 3 asignaturas por cada semestre y 2 grupos por asignatura.
- Debe aplicar programación orientada a objetos y el principio de alta cohesión y bajo acoplamiento.
- Debe aplicar los conceptos de optimización estudiados en el curso. No limitarse a garantizar el funcionamiento del programa.
- Al finalizar el programa se actualizan los archivos incluyendo los cambios realizados durante la ejecución.

Condiciones de entrega

1. Se puede elaborar en grupos de por lo menos dos y máximo tres personas (Es un trabajo en grupo). Se sustentará individualmente.
2. Se entrega a través de aula virtual una carpeta comprimida con: código fuente, ejecutable, archivos planos y documento soporte. Sencillo manual para ejecución.
3. El programa debe correr en DevC++, sin requerir ningún tipo de ajuste.
4. Documento soporte incluye: Diseño de la solución, Contenido y estructura de archivos planos, Estructuras de datos en memoria principal definidas, incluyendo diagrama.
5. Archivo fuente debidamente autodocumentado.

Docente: Deicy Alvarado