

PROYECTO 1: Asignación de horarios

El objetivo de este proyecto es la asignación de horarios a diferentes secciones de clases conformadas, cada una, por un profesor y varios estudiantes.

Descripción general:

La idea del proyecto es realizar un programa en C++ para asignar las horas de clase en un día específico (por ejemplo, lunes), a varias secciones de cursos. Una sección de clase se corresponde con una sección de clases de la UCAB, es decir, una sección está conformada por un profesor y un conjunto de estudiantes. La sección corresponde a una materia, por ejemplo, “Algoritmos y Estructuras de Datos INFO-02002”. Cada sección tiene un conjunto de horas de dictado de clases por semana, que se distribuyen en a lo sumo tres bloques de horas; por ejemplo, “Algoritmos y Estructuras de Datos INFO-02002” tiene tres bloques de horas, a saber, 2 horas de teoría, 2 horas de práctica y 2 horas de laboratorio. Otra sección de otro curso, por ejemplo “Matemáticas Discretas INFO-02000” tiene 2 horas de teoría, otras 2 horas de teoría y 1 hora de práctica.

Una sección de clase cualquiera, tendrá a lo sumo tres bloques de horas. Por ejemplo, “Algoritmos y Estructuras de Datos INFO-02002” tiene los bloques 2, 2, 2; mientras que “Matemáticas Discretas INFO-02000” tiene los bloques 2, 2, 1. Otra sección de algún otro curso tendrá los bloques 2,2. Lo importante es que supondremos que una sección tendrá 1, 2 ó 3 bloques de horas. Las combinaciones de bloques permitidas son 2,2,2 ó 2,2,1 ó 2,1 ó 1,1 ó 3 ó 2,1 ó 1,3 ó 2.

El objetivo final del proyecto es asignar un horario a los bloques de cada sección de clase en un día determinado. Un bloque de horas deberá ser asignado a horas contiguas entre sí, por ejemplo, a “Algoritmos y Estructuras de Datos SECCION 1” que tiene los bloques 2,2,2 (el primero lo llamamos “bloque 1”, al segundo “bloque 2” y al tercero “bloque 3”) le asignamos el horario: bloque 1 de 9:00 a 11:00 el lunes, el bloque 2 de 9:00 a 11:00 los miércoles y el bloque 3 de 9:00 a 12:00 los viernes. Utilizaremos el sistema horario de 24 horas; en lugar de 01:00pm escribimos 13:00.

En este proyecto haremos una asignación parcial de los horarios, pues solo asignaremos los horarios de un solo día.

En la segunda parte del proyecto haremos la asignación completa semanal del horario a los bloques de cada sección.

Toda la información sobre las secciones de clase se tendrá en un archivo de texto que contendrá las diferentes secciones de clase. Este archivo de texto deberá ser leído y almacenado en estructuras de datos adecuadas. Cada sección en el archivo, contendrá la siguiente información:

- a) Nombre del curso.
- b) Bloques de horas.
- c) Nombre, apellido y número de cédula del profesor a cargo de la sección.
- d) Nombre, apellido, número de cédula de cada estudiante inscrito en la sección.

Tendremos las siguientes restricciones en la asignación de horarios a los bloques de una sección específica en un día específico:

- Podemos asignar hasta 4 horas de clase en un mismo día.
- Las horas que se asignen a un bloque deberán ser contiguas, por ejemplo: de 13:00 a 15:00 ó de 9:00 a 11:00
- Si una sección tiene 3 bloques de dos horas cada uno, no podemos asignar más de dos bloques de dos horas un mismo día.
- Si una sección tiene 2 bloques de 2 horas y un bloque de 1 hora, no podemos asignar mas de un bloque de 2 horas el mismo día.

Las horas de un día las denotamos por 1, 2, 3, 4, 5, 6, 7, 8, 9 y 10. Y corresponden a las horas 7:00 a 8:00, 8:00 a 9:00, ... hasta 16:00 a 17:00.

Note que si dos secciones, digamos s_1 y s_2 , poseen estudiantes en común o el mismo profesor, entonces las horas asignadas a un bloque cualquiera de s_1 deberán ser todas distintas de las horas asignadas a los bloques de s_2 y viceversa. Por ejemplo: s_1 tiene bloques 2, 2, 2 y s_2 tiene bloques 2, 2. Si comenzamos asignando las horas 1 y 2 al bloque 1 (el primer bloque de 2,2,2) de s_1 entonces ninguno de los dos bloques de s_2 podrá tener las horas 1 ó 2 el mismo día.

Veamos un ejemplo:

Si el archivo de texto que contiene las secciones tiene el siguiente formato:

Seccion: Algoritmos y Estructuras de Datos INFO-02002

Bloques: 2 2 2

Profesor: Carlos Perez ci: 11111111

Estudiantes:

Maria Perez ci: 31000000

Carlos Perez ci: 31000001

Omar Acosta ci: 31000002

Seccion: Algoritmos y Estructuras de Datos INFO-02002

Bloques: 2 2 2

Profesor: Nancy Martinez ci: 11111112

Estudiantes:

Ramon Perez ci: 31000004

Carlos Martinez ci: 31000005

Jose Acosta ci: 31000006

Seccion: Matematicas Discretas INFO-02000

Bloques: 2 2 1

Profesor: Henry Marcano ci: 11111113

Estudiantes:

Maria Perez ci: 31000000

Carlos Perez ci: 31000001

Omar Acosta ci: 31000002

Ramon Perez ci: 31000004

Carlos Martinez ci: 31000005

Jose Acosta ci: 31000006

Tenemos dos secciones de “Algoritmos y Estructuras de Datos” que no tienen profesor común ni estudiantes comunes. Y una sección de “Matemáticas Discretas” que tiene estudiantes comunes con las dos secciones de “Algoritmos y Estructuras de Datos”.

Vamos a proponer el siguiente algoritmo para asignar las horas que podamos, a cada sección, en un solo día:

- Comenzamos asignando el horario del día para la primera sección que aparece en el archivo. La sección de “Algoritmos y Estructuras de Datos” tiene al menos un bloque de 2 horas. Podemos asignar las horas 1 y 2 al primer bloque de dos horas de esta sección y asignar de una vez las horas 3 y 4 al segundo bloque de dos horas, y por las restricciones impuestas NO podemos asignar en este día las otras dos horas del tercer bloque, las cuales quedarán pendiente de asignar en otro día, pero eso no nos compete en este proyecto. Es importante que al asignar un horario a un bloque de una sección también guardemos el día al que corresponde esa asignación, por ejemplo, bloque 1 horas 1 y 2 día X (lunes, martes, etc.); para esta etapa del proyecto siempre será un mismo día, pero para la siguiente etapa no.
- Ahora pasamos a asignar las horas de los bloques de la segunda sección en el archivo (que también es de “Algoritmos y Estructuras de Datos”). Como esta sección no tiene profesores ni estudiantes en común, podemos asignar las mismas horas 1 y 2 al primer bloque de dos horas de esta sección y asignar de una vez las mismas horas 3 y 4 al segundo bloque de dos horas, y por las restricciones impuestas NO podemos asignar en este día las otras dos horas del tercer bloque.
- Ahora pasamos a asignar las horas de la tercera sección en el archivo, que corresponde a “Matemáticas Discretas”. Notemos primero que esta sección tiene estudiantes en común con las dos secciones previas ya asignadas (decimos que las dos secciones previas **están en conflicto** con ésta por tener el mismo profesor o estudiantes en común). Por lo tanto, debemos descartar todas las horas asignadas a las primeras dos secciones, es decir, las horas 1, 2, 3 y 4. Quedan libres las horas de 5 a 10. Como los bloques de esta sección son 2, 2, 1, por las restricciones impuestas NO podemos asignar más de un bloque de 2 horas este día, por lo tanto, asignamos las horas 5 y 6 a un bloque de 2 y la hora 7 al bloque de una hora. Quedaría pendiente asignar las otras 2 horas en otro día, pero eso no nos compete en este proyecto.

Por lo tanto, como ya asignamos los horarios de todas las secciones para un día, podemos guardar en un archivo la asignación de horarios resultante. Por ejemplo, el archivo con la solución sería:

Asignación horaria de este día (llamémoslo X):

Día X

Curso: Algoritmos y Estructuras de Datos INFO-02002

Profesor: Carlos Perez ci: 11111111

Horario: 7:00 a 9:00, 9:00 a 11:00

Estudiantes:

Maria Perez ci: 31000000

Carlos Perez ci: 31000001

Omar Acosta ci: 31000002

Curso: Algoritmos y Estructuras de Datos INFO-02002

Profesor: Nancy Martinez ci: 11111112

Horario: 7:00 a 9:00, 9:00 a 11:00

Estudiantes:

Ramon Perez ci: 31000004

Carlos Martinez ci: 31000005

Jose Acosta ci: 31000006

Curso: Matematicas Discretas INFO-02000

Profesor: Henry Marcano ci: 11111113

Horario: 11:00 a 13:00, 13:00 a 14:00

Estudiantes:

Maria Perez ci: 31000000

Carlos Perez ci: 31000001

Omar Acosta ci: 31000002

Ramon Perez ci: 31000004

Carlos Martinez ci: 31000005

Jose Acosta ci: 31000006

Note que el algoritmo anterior es sensible (puede dar resultados distintos) al orden en que colocamos las secciones en el archivo, porque dará prioridad en la asignación de horarios, teniendo mayor prioridad las primeras secciones de clase en el archivo. Si consideramos las secciones en orden inverso a como aparecen en el archivo entonces podríamos obtener una asignación de horarios distinta.

Hay otra estrategia para ir asignando horarios a secciones en un día, que llamaremos **estrategia de mayor conflicto** y consiste en lo siguiente:

- a) Al momento en que vamos a asignar el horario a una sección, la escogeremos entre las que posean la mayor cantidad de conflictos con ella y que ya hayan sido asignados (recuerde que dos secciones están en conflicto si poseen el mismo profesor o tienen estudiantes en común), y desempataremos por la que tenga mayor cantidad de conflictos con ella (ya asignados o no).
- b) Note que para determinar la sección propuesta en (a), podemos aplicar el algoritmo siguiente: primero creamos un vector (o lista) con cada sección no asignada, su número de conflictos con ya asignadas y su número de conflictos con asignadas o no. Luego se determina en este vector (o lista) la sección buscada.

Requerimientos:

La entrada de datos se hará mediante un archivo con el formato antes descrito. Se les dará un archivo de prueba, que podrán modificar a su antojo para hacer distintas pruebas.

La interfaz del usuario será como sigue:

Deberá existir un menú de opciones por pantalla donde se solicite al usuario el nombre del archivo que contiene las secciones.

Luego se le presenta otro menú que ciclará hasta que el usuario decida salir de él, y que consiste en una de estas tres opciones:

- si desea la asignación de horarios considerando las secciones en el orden en que aparecen en el archivo.
- si desea la asignación de horarios considerando el orden inverso en el archivo, es decir, empezar a asignar la última sección que aparece en el archivo, luego la penúltima y así sucesivamente.

- si desea la asignación de horarios considerando **la estrategia de mayor conflicto**.

Luego de hacer la asignación según la estrategia escogida, se pregunta al usuario el nombre del archivo donde se almacenará el resultado de la asignación y también se le preguntará si desea que se muestre el archivo resultante por pantalla.

En caso de que el usuario solicite mostrar por pantalla el resultado de la asignación de horarios, se le debe preguntar si desea la versión corta o la versión larga. La versión larga es la mostrada anteriormente, donde se muestra la lista de estudiantes. La versión corta es parecida en formato a la larga, pero no se muestran los estudiantes.

Luego se volverá a solicitar otro archivo de secciones para hacer la asignación, en caso de que el usuario desee calcular la asignación para un nuevo archivo. Hasta que el usuario indique que no desea calcular nuevos horarios.

Grupos de proyecto:

- Se constituirán en las primeras grupos de 3 estudiantes para hacer los dos proyectos del curso.

Características técnicas:

- El programa debe ser implementado en Lenguaje C++. Solo se permite el uso de tipos primitivos del lenguaje, cadena de caracteres (tipo string), tipo vector y listas enlazadas simples como estructuras de datos. Para almacenar las secciones deberá utilizar listas enlazadas, también deberá utilizar listas enlazadas para almacenar la lista de los estudiantes por cada sección.
- Debe estructurar en .h y .cpp los elementos necesarios para la gestión de los requerimientos, tratando de encapsular bien cada funcionalidad, evitando el desorden en el código. Por ejemplo encapsular el tipo SECCION, el tipo lista de SECCIONES con sus operaciones; por ejemplo, la operación de asignar horarios en un día específico, determinar la siguiente sección a asignar en un día específico (la primera no asignada, la última no asignada, o la con mayores conflictos. Pueden ser tres funciones diferentes).
- Se supone que el archivo con las secciones está correcto, es decir, se rige por el formato dado anteriormente.

Fecha Entrega: 12 de Mayo

Características de la evaluación:

- Cada grupo deberá subir por drive a la carpeta compartida Proyecto 1:

https://drive.google.com/drive/folders/19dgOheRezaLapJLaREdF_uAJMkJ-On?usp=drive_link

un archivo .zip con los programas fuente (solo los .h y .cpp) en C++ y el pdf describiendo y explicando en forma general y breve el programa (la solución propuesta, tipos de datos, estructuras de datos y modularización); es decir, la arquitectura del programa. Luego deberán explicar cómo compilar y ejecutar el programa y comentarios adicionales. El nombre del archivo .zip deberá ser: **PROYECTO1_GRUPO_X.zip** (o **rar**) donde X es el número del grupo.

- Los tres integrantes deben conocer la solución de todo el proyecto.
- En clase de laboratorio se pedirá hacer a todos los estudiantes una pequeña modificación al proyecto que debe ser implementada individualmente (calificación individual) y subir por drive (a la carpeta compartida “Solución individual Proyecto 1”) un .zip con todo el proyecto e indicando con un comentario (//nueva linea) las nuevas líneas de código. El nombre del archivo .zip deberá ser: **PROYECTO1_GRUPO_X_APELLIDO_NOMBRE.zip** donde X es el número del grupo. Para ese día deberá llevar en su pendrive o en la nube la solución (programa en C++) hecha por su grupo del proyecto 1, el cual deberá modificar. Tiene una valoración de 4 puntos sobre los 20 puntos.

Recomendaciones generales:

- Definir un jefe de proyecto en el grupo que supervise las actividades (por supuesto, este integrante debe desarrollar también).
- Discutir en conjunto sobre los tipos y las estructuras de datos a utilizar y su modularización, y las actividades que va a realizar cada integrante. Analicen primero, hagan un esquema (o arquitectura) del código por tipos de datos y funcionalidades en general (.h y .cpp). Definan tareas a realizar, cuales tareas son prioritarias).
- La implementación deberá ser lo más modular posible. Por ejemplo, definir los tipos de datos (ejemplo: seccion, secciones) y en consecuencia los algoritmos a desarrollar, todas las operaciones necesarias sobre secciones, etc., agruparlas en .h y .cpp aparte.

- Imprescindible: Documentar todas las funciones y el programa en general como se indicó en clases. Y es recomendable colocar comentarios entre líneas de código para ir explicando lo que realiza un fragmento de código dentro de una función.