



# Diseño y desarrollo de un sistema de gestión del tiempo de estudio para estudiantes universitarios en C++

Cristian David Ibarra Cortés

Facultad de Ingeniería

Departamento de Ingeniería y Telecomunicaciones

## Tabla de contenido

<b>1. Introducción</b> .....	<b>1</b>
<b>2. Planteamiento del problema</b> .....	<b>2</b>
2.1 ¿Qué debe hacer el sistema? .....	2
2.2 Herramientas permitidas y no permitidas.....	2
<b>3. Análisis y propuesta de solución</b> .....	<b>4</b>
3.1 Gestión de la información.....	5
3.2 Procesamiento de la información.....	5

## 1. Introducción

Este documento pretende plasmar de forma detallada el proceso de análisis llevado a cabo durante el proceso de desarrollo y construcción de una posible solución al problema planteado para el parcial 1 de la asignatura Informática II. El ejercicio consiste en desarrollar un sistema que nos permita a nosotros como estudiantes realizar una gestión adecuada del tiempo de estudio, utilizando como información principal, el pensum del correspondiente pregrado y las materias que se encuentre cursando cada alumno; se debe realizar una gestión de la información de tal manera que el sistema le permita al usuario conocer cuáles son los espacios de tiempo apropiados para estudiar. Para comenzar; haremos una descripción detallada del problema, que es lo que se debe hacer, cuáles son las herramientas que podemos utilizar, un análisis de cómo podríamos solucionarlo y finalmente una propuesta con la posible solución al problema valiéndonos de conceptos aprendidos en clase, tales como, manejo de archivos, gestión de memoria, funciones, entre otros.

## 2. Planteamiento del problema

El objetivo principal de este trabajo es desarrollar de un sistema de gestión del tiempo para estudiantes que permita ayudar a mitigar uno de los problemas que más se presenta dentro de la comunidad estudiantil; el inadecuado manejo del tiempo de estudio. Para lograr este objetivo, debemos hacer uso de las herramientas que nos han sido otorgadas durante el curso, tanto en la teoría como en la práctica, haciendo un énfasis especial en el manejo de archivos y la gestión de memoria.

### 2.1 ¿Qué debe hacer el sistema?

El manejo de archivos es fundamental para el desarrollo de este sistema ya que este debe hacer la gestión de la información utilizando dos archivos. El primero, al que llamaremos *Archivo del pensum*, contiene toda la información concerniente a las materias en el pensum del pregrado de Ingeniería Electrónica; código de la materia, cantidad de créditos, cantidad de horas de teoría, cantidad de horas de práctica y cuánto tiempo se debe dedicar a cada materia por fuera de las clases según el número de créditos. El segundo archivo, al que llamaremos "*Archivo del estudiante*" contiene toda la información del estudiante; las materias que está cursando y los horarios en los que debe asistir a las clases asistidas por maestros, por lo cual, el objetivo principal del sistema es hacer el computo de esta información y presentar al estudiante un horario donde le indica cuales son los momentos más apropiados para estudiar y poder cumplir con el número de horas de estudio necesarias para cumplir a cabalidad con el pensum.

### 2.2 Herramientas permitidas y no permitidas

Para la implementación de este sistema está prohibido el uso de objetos, especialmente objetos de la clase *string*. Tampoco está permitido el uso de cualquier librería distinta a la *fstream* para el manejo de archivos. Para el manejo de la información, se deberán emplear punteros del tipo *char* y por supuesto, está permitido el uso de todos los conceptos relacionados con el manejo de archivos y el uso de memoria.

### 3. Análisis y propuesta de solución

A continuación, se hace la descripción de los pasos que se seguirán para conseguir el objetivo propuesto.

#### 3.1 Gestión de la Información

La información tanto de los cursos, como de los estudiantes estará almacenada en los archivos "cursos.txt" y "estudiantes.txt", para facilitar el manejo. He decidido almacenar la información con un formato muy específico que me facilitará su manipulación más adelante, en el caso del archivo que contiene los cursos, la información se almacenará de la siguiente manera:

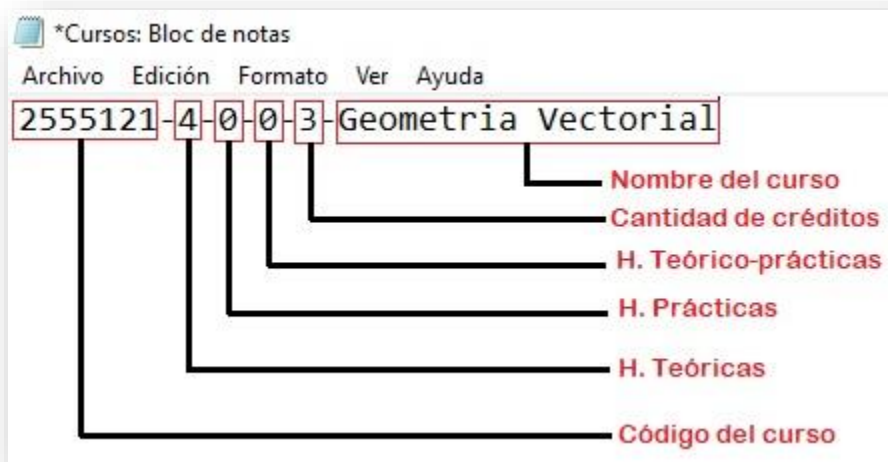


Ilustración 1

Cada línea del archivo contendrá la información completa de un curso, como se observa en la *ilustración 1*, considero que esta es una forma organizada para almacenar la información que se requiere a cerca de los cursos y muy similar a la manera como se nos presenta la información de un curso en el plan de estudios:

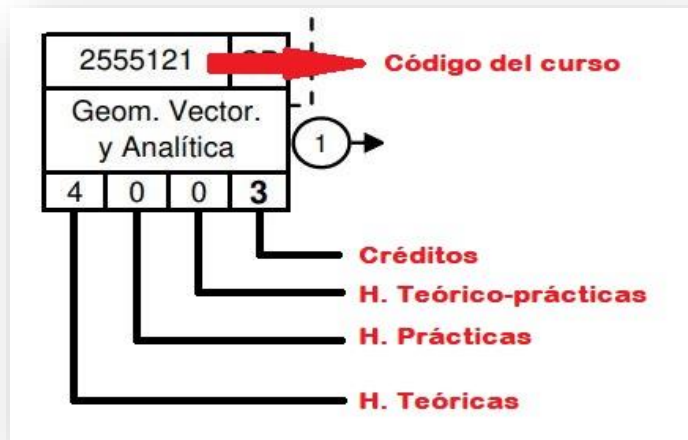


Ilustración 2

El archivo "estudiantes.txt", almacenará la información completa de los estudiantes registrados en el sistema de la siguiente manera:

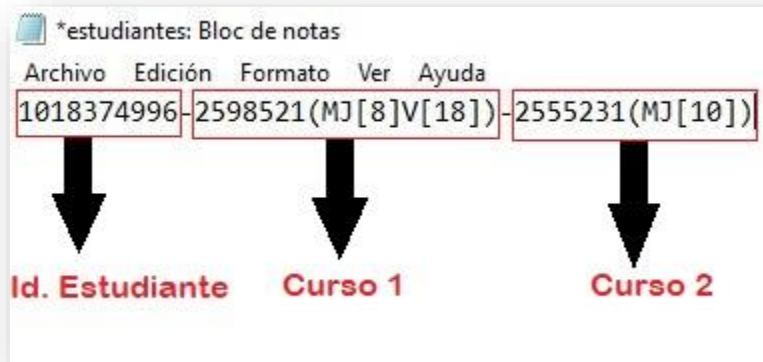


Ilustración 3

Como podemos apreciar en la ilustración, cada línea contiene la información completa del estudiante y sus cursos matriculados. En primer lugar, tendremos el número de identificación del estudiante registrado, luego, la información de cada curso separada por guiones. He seleccionado también este formato porque considero que me permitirá acceder de una manera más fácil y ordenada a la

información de los cursos matriculados por cada estudiante, en este formato, podremos diferenciar la información de los horarios de cada curso de la siguiente manera. Como bien sabemos, generalmente los cursos que tienen 4 créditos emplean 7 horas de clase semanales asistidas por el profesor donde 4 son de teoría y 3 son de práctica, además, las horas de teoría se dividen en 2 días de a 2 horas por jornada, es así entonces como las materias con 4 créditos tendrán el formato mostrado a continuación:

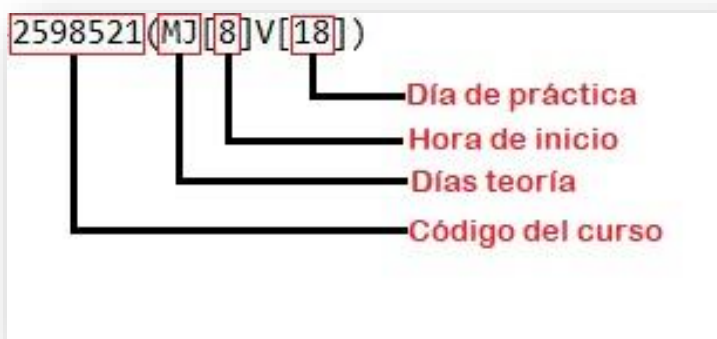


Ilustración 4

También es bien sabido que en su mayoría las materias de 3 créditos requieren de 4 horas de estudio semanales asistidas por el maestro y estas se dividen en 2 bloques de 2 horas en 2 días de la semana, por lo tanto, el formato para las materias de 3 créditos quedaría así:

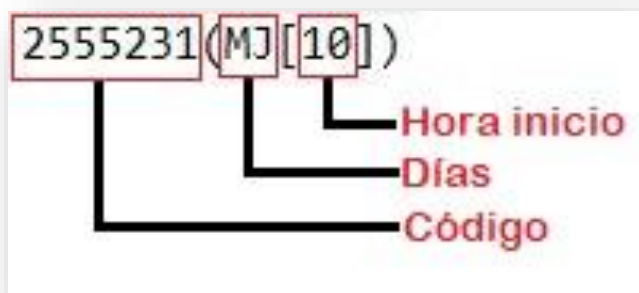


Ilustración 5

Existen algunos casos particulares, como es el caso de las materias electivas o de laboratorio, pero estos casos los evaluaremos individualmente a la hora de recolectar la información.

### 3.2 Procesamiento de la información

Para empezar, cuando un usuario ejecute el programa con el ánimo de aprovechar su función principal, es decir, planear su semana de estudio, primero deberá identificarse o registrarse si es que aun no se encuentra en la base de datos; durante el registro, el programa le pide al usuario que ingrese su numero de identificación, luego el usuario debe registrar su primera materia ya que debe tener registrada por lo menos una. El sistema inicia solicitando el código del curso, esto nos servirá para comparar este valor con la información que se encuentra registrada en el archivo "*cursos.txt*" y determinar rápidamente de que curso se trata, de esta forma podremos determinar cuántas horas de teoría o práctica requiere el curso y determinar cual es el formato que debemos usar para almacenar esta información, como ya se había visto, los cursos de 3 y 4 créditos tienen un formato específico. Después de solicitar el código, el sistema pide al usuario que ingrese los días en los que se ve la materia; en el caso de las materias de 3 créditos el sistema pide 2 días (Que son de teoría por lo visto anteriormente) y en el caso de las materias de 4 créditos, el sistema pedirá 3 días, los 2 de teoría y el día en que se desarrolla la práctica, obviamente también se analizará de manera particular los casos en que esta regla no aplique. Luego de todo esto la información será almacenada como se mostró en las anteriores ilustraciones según corresponda, utilizando por supuesto todas las herramientas que nos brinda la librería *fstream*.

Luego de que la información ya se encuentra almacenada correctamente, procedemos a realizar el computo de los datos con el fin de dar a conocer al usuario los mejores resultados posibles. Cuando el usuario haya ingresado su identificación (En caso de estar registrado), el sistema procederá a buscar su información almacenada, para esto, el documento se leerá línea por línea con la ayuda de las funciones de *fstream*. La información de cada línea se almacena en un arreglo y posteriormente se aplica una función para extraer el número del documento, si ambos coinciden, significa que estamos en la línea correcta y utilizaremos el arreglo principal para operar, un bosquejo de este programa podría ser:



```

char* documentoentrada[];
char* documento[];
char* arreglo[];

funcion(arreglo[])=documento;

if(documento==documentoentrada){
|  //Trabajar sobre esta linea
}

```

*Ilustración 6*

La planificación del estudio se limitará a un horario específico. El sistema solicitará al usuario que indique la hora más temprana y la hora más tarde en que estaría dispuesto a comenzar y finalizar el estudio, respectivamente. De esta manera se establecerá una franja horaria para las actividades académicas, esta franja horaria estará representada por un arreglo de char cuyo valor en cada posición indica si el espacio se encuentra ocupado o no, la cantidad de elementos es el número de horas en la franja seleccionada, en este arreglo, si el valor del arreglo es diferente de "", entonces el espacio se puede ocupar, por lo tanto se le indica al usuario que el espacio está disponible y si desea ocuparlo, si la respuesta del usuario es afirmativa, se le solicita que indique que materia desea estudiar en ese espacio, de acuerdo a lo que el usuario indique, se llenará esta casilla con el código que corresponde a la materia que el alumno quiere estudiar. El sistema irá mostrando por días los espacios disponibles para dedicar a las horas de estudio no asistidas por docente, al mismo tiempo irá mostrando cuantas horas le faltan para poder cumplir con el requerimiento mensual y semanal necesario según el número de créditos del curso, el estudiante debe seleccionar cuales son los horarios de su preferencia para cumplir con estas horas, el sistema irá llenando los espacios vacíos según las preferencias del usuario, si en determinado momento el sistema establece que los espacios disponibles que quedan no son suficientes para cumplir con el tiempo que se debe estudiar semanalmente, entonces asignará al tiempo que se requiere a los días faltantes en los espacios vacíos automáticamente. La siguiente ilustración pretende darnos una idea de como el sistema va organizando la información:

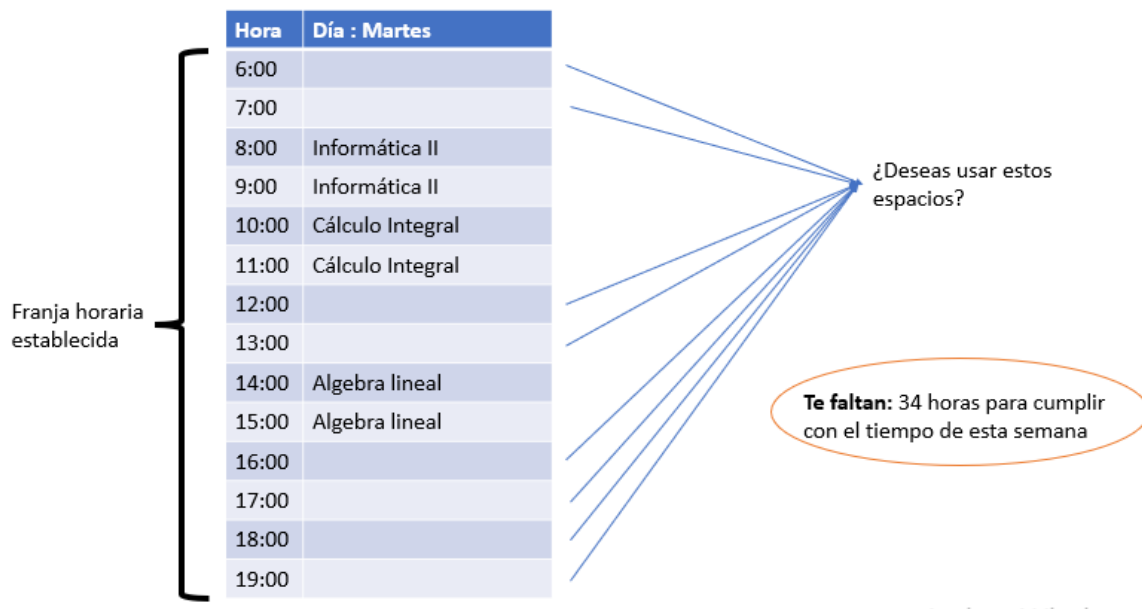


Ilustración 7

Este puntero de char, hará parte de un doble puntero que contendrá 7 punteros de tipo char que representan los 7 días de la semana, la longitud de estos punteros está determinada por el número de horas que esté dispuesto a estudiar el alumno cómo ya habíamos dicho; primero llevo a este puntero toda la información de los cursos del estudiante, para lograrlo debo volver al arreglo que contiene la line que estoy operando y crear una función que me permita determinar en que día y que hora se tienen clases asistidas por docente con la información que me brinda el documento “*estudiantes.txt*”, cuando haya obtenido el dato, me dirijo al arreglo en la posición que representa ese día y esa hora y convierto su valor en el código del curso, además añado un marcador (puede ser un numero adicional o algún carácter que no haya usado) para indicar que esa clase es asistida por maestro; de esta manera, cuando pasemos a consultar al estudiante, el arreglo ya contendrá la información de las clases presenciales, con lo que podremos proceder a ingresar las horas no asistidas por profesor. El siguiente esquema muestra un ejemplo del arreglo antes de pedirle al estudiante que llene los espacios con sus preferencias:

Días de la semana								
Posiciones de la franja horaria:	Hora	Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
	6:00 [0]							
	7:00 [1]							
	8:00 [2]			2598521		2598521		
	9:00 [3]			2598521		2598521		
	10:00 [4]			2555231		2555231		
	11:00 [5]			2555231		2555231		
	12:00 [6]							
	13:00 [7]							
	14:00 [8]			2555221	2536201	2555221	2536201	
	15:00 [9]			2555221	2536201	2555221	2536201	
	16:00 [10]							
	17:00 [11]							
	18:00 [12]						2598521	
	19:00 [13]						2598521	
	20:00 [14]						2598521	
	21:00 [15]							

Activar Windows

Ilustración 8

El arreglo se recorre en sus 2 dimensiones y si el espacio está vacío, el sistema pregunta si se quiere usar, de lo contrario avanza, este ciclo se repite hasta que se den una de las condiciones anteriormente mencionadas.

Ahora, luego de que ya han sido asignadas todas las horas necesarias en las franjas horarias establecidas, procederemos a presentar al usuario la información con el horario que ha salido como resultado, para esto solo deberemos recorrer el doble puntero en sus 2 dimensiones y recordar que sus posiciones representan los días de la semana y las horas en la franja horaria respectivamente.