**Parcial 1**

**Juan Felipe Vásquez Bolívar & Jackh Emmanuel Narváez Guerra**

**C.C 1001509367 & C.C 1002752230**

**Augusto Enrique Salazar Jiménez**

**Abril 2023**

**Universidad de Antioquia**

**Facultad de Ingeniería**

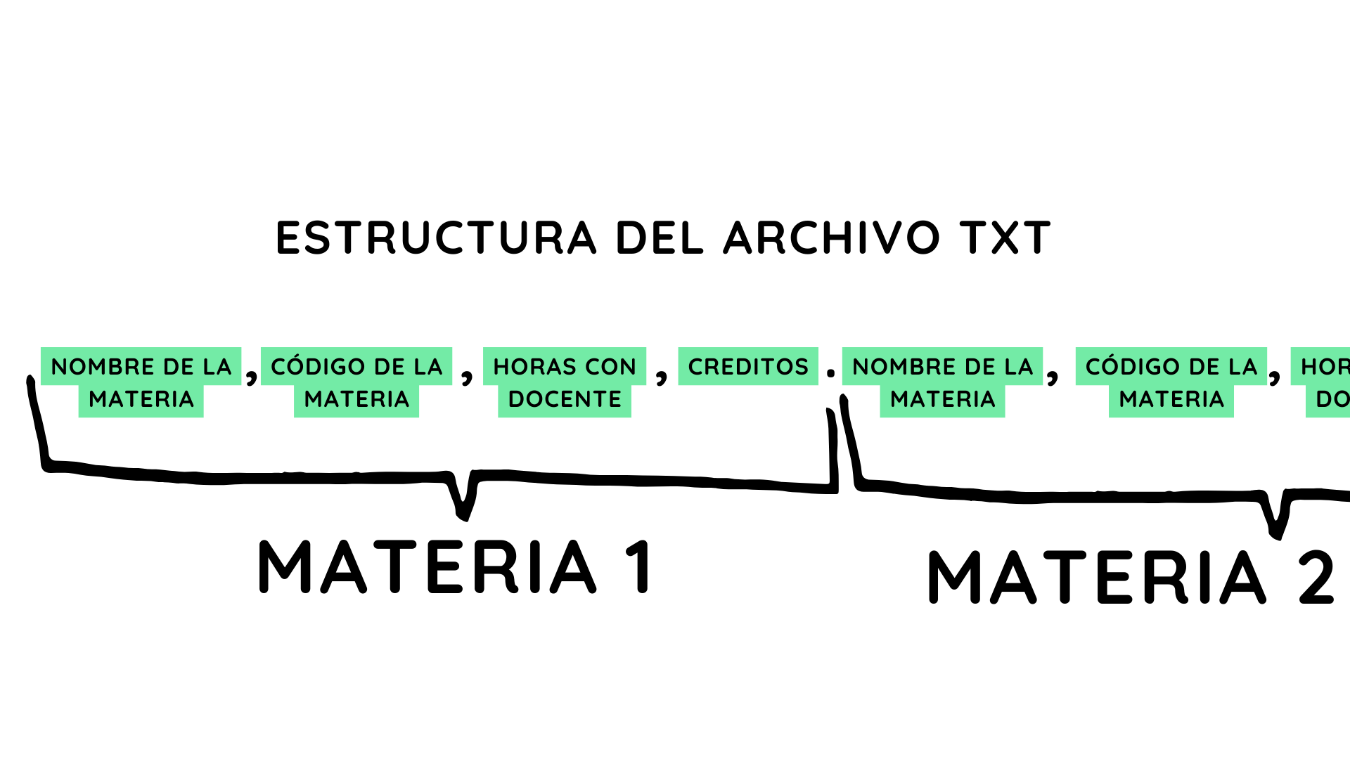
**Informática II**

1. **Gestionar información de cursos**

**1.1 Construcción de Información:** Para almacenar la información con respecto a los cursos de la carrera “Ingeniería de telecomunicaciones” se construirá un archivo de texto plano que contenga las horas con el docente, nombre, código y créditos de la materia, las cuales serán importadas por el programa final.

Esta información será almacenada en el archivo **“Plan de Estudios.txt”.**

**1.2 Estructura de la Información:** Para almacenar la información antes mencionada, es necesario establecer una estructura dentro del archivo de texto plano, que facilite la lectura por parte programa. Por lo tanto, trabajaremos de la siguiente manera:



La anterior estructura se encuentra diseñada para determinar fácilmente el número de materias que tiene la carrera, pues será igual al número de puntos (‘.’) que tenga el texto.

Además, separar los tipos de datos por medio de comas (‘,’) facilita determinar la longitud que tiene cada parámetro, esto con el objetivo de construir arreglos de caracteres más eficientes.

Por último, no se decide utilizar saltos de línea con el objetivo de facilitar la lectura del archivo.

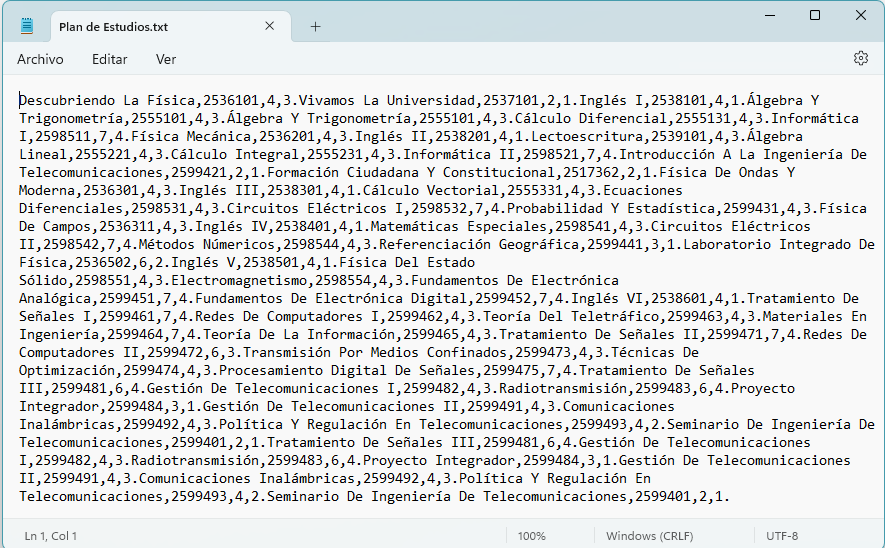
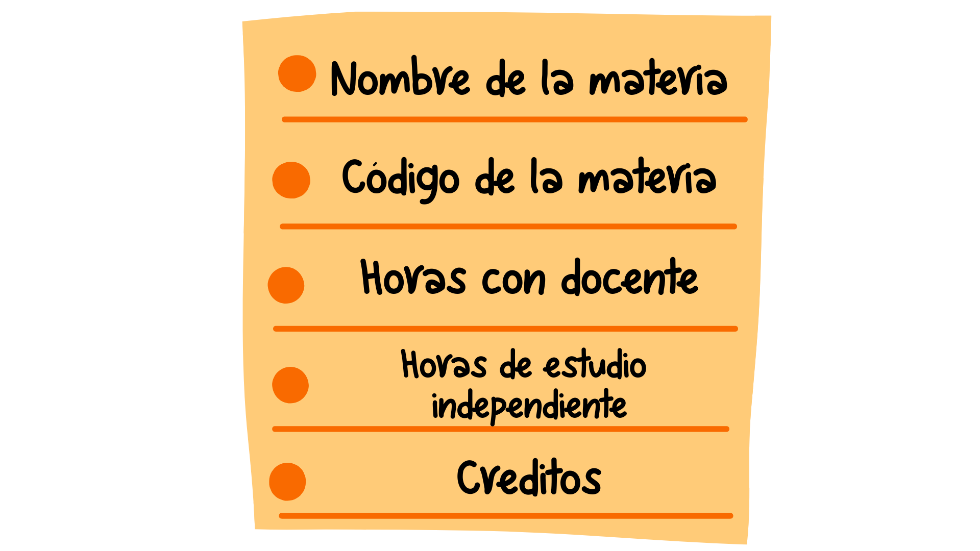


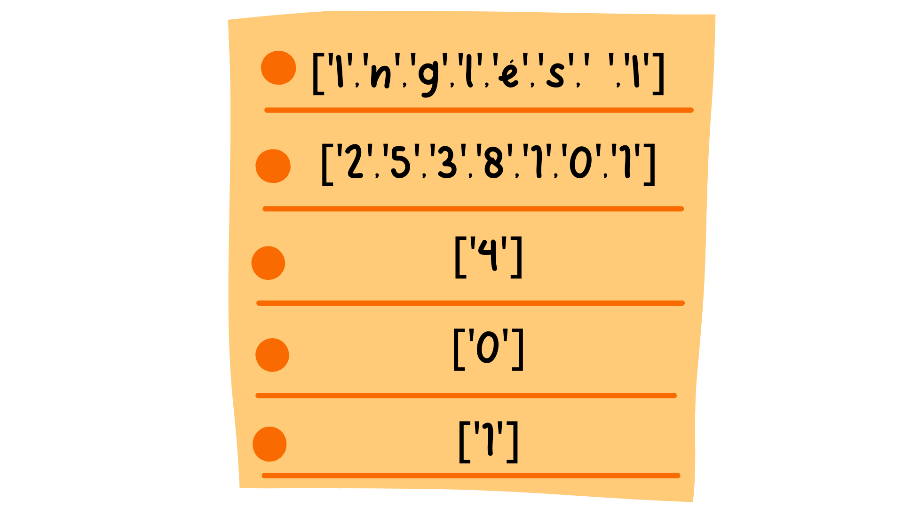
Ilustración : Ejemplo de cómo estará almacenada la información.

**NOTA:** Durante esta primera fase del programa, no se tendrá en cuenta las electivas profesionales, electivas prácticas académicas y las electivas complementarias. No obstante, en caso de querer incluirlas, será necesario incluirlas en el archivo “Plan de Estudios.txt” con la estructura antes mencionada.

**1.3 Idea general:** El objetivo de esta parte del programa, será importar la información contenida en “Plan de Estudios.txt” y organizar esta información en un conjunto de matrices (Puntero triple), donde cada matriz contará con 5 filas con la siguiente estructura:



Cada fila contará con una cantidad variable de columnas, ya que dependerá de la cantidad de caracteres que contenga, por ejemplo:

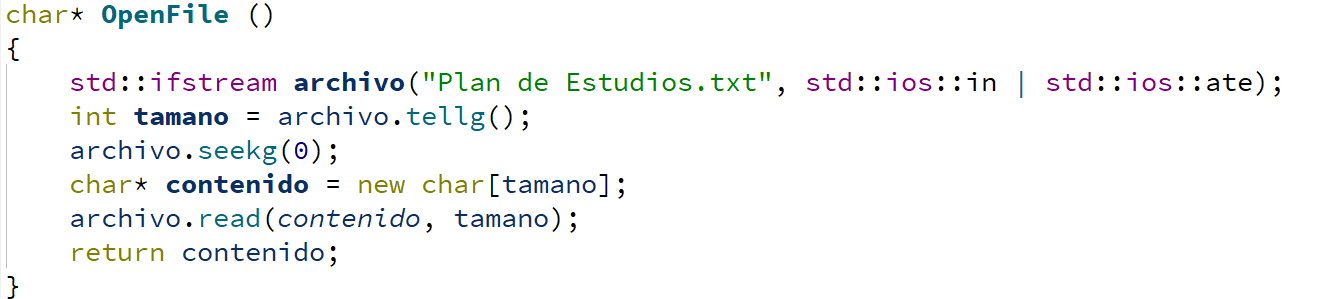


Es importante mencionar, que el valor ingresado a la fila 4, correspondiente a “Horas de estudio independiente”, se calculará por medio de la ecuación estipulada.

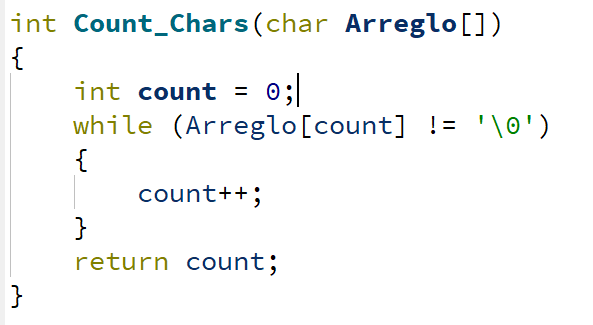
**Funciones**

Para el desarrollo de esta parte del programa y cumplir con el objetivo antes mencionado, se crean las siguientes funciones, las cuales estarán contenidas dentro de **“GestionInformacion.h”**:

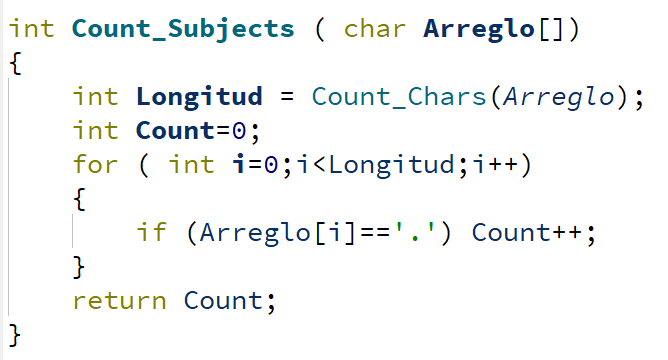
**1. Open\_File :** Función que no recibe ninguna entrada, y retorna un puntero tipo Char, que apunta a un arreglo de caracteres que contiene la información del archivo “Plan de Estudios.txt”.



**2. Count\_Chars :** Función que recibe como entrada un arreglo de caracteres, y retorna un numero entero con la cantidad de caracteres de dicho arreglo.

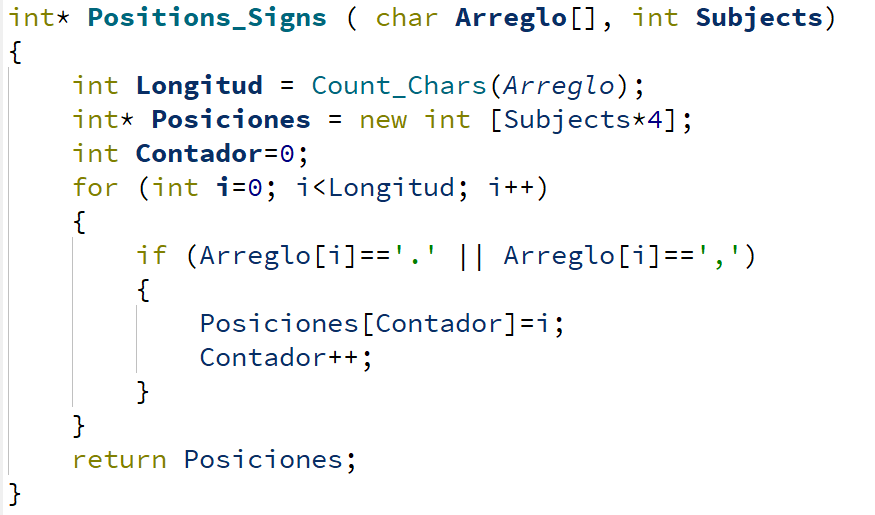


**3. Count\_Subjects :** Función que recibe como entrada un arreglo de caracteres, y retorna un numero entero con la cantidad de asignaturas registradas. Teniendo en cuenta que cada materia se encuentra separada por medio de un punto (‘.’), bastará con contar la cantidad de veces que se repite el carácter ‘.’, para determinar el número materias.



**4. Positions\_Signs:** Función que recibe como entrada un arreglo de caracteres, y un numero entero, y retorna un arreglo de enteros con la posición de los caracteres ‘,’ y ‘.’, dentro del arreglo.

El número entero corresponde a la cantidad de materias (Puntos ‘.’) que contiene el arreglo y teniendo en cuenta que en la estructura diseñada, cada materia cuenta con 4 signos ( 3 comas y 1 punto ), se crea el arreglo de enteros a retornar en la memoria dinámica.



**5. Reserve\_Matix\_Group:** Función que recibe como entrada un arreglo de números enteros y un número entero, y retorna un puntero triple.

Esta función realiza las reservas de memoria dinámica con base en la cantidad de caracteres que contenga cada subcategoría de cada materia, con el objetivo de realizar un uso optimo de la memoria dinámica.

1. **Ingresar información de los cursos matriculados**

**2.1 Información solicitada:** Para esta parte del programa se solicitará inicialmente el usuario del estudiante, el cual corresponderá a su correo institucional, esto con el objetivo de crear un archivo único por estudiante con un horario de acuerdo a las materias matriculadas. Así mismo, será necesario suministrar el código de cada materia matriculada y los días y horas en que recibirá cada una de las clases.

**2.2 Construcción de la información:** Con base en la información anteriormente suministrada, se construirá un conjunto de matrices (Puntero triple) que cumplirá el papel de diagrama de horario, por lo tanto, en total se crearán 7 matrices (Punteros dobles), correspondientes a los días de la semana, donde la matriz cero corresponderá al día lunes, la matriz uno corresponderá al día martes y así sucesivamente.

Cada una de estas matrices contará con una cantidad de 24 filas, y una cantidad de 7 columnas, es decir, se creará un arreglo de 24 elementos, donde cada elemento será un arreglo de 7 caracteres.

Cada uno de estos 24 elementos representará una hora del día, donde el elemento cero, corresponde a la primera hora del día, es decir, de las 00 a las 01, el elemento uno corresponde a la segunda hora del día, es decir, de las 01 a las 02, y así sucesivamente. Es importante aclarar que, para mayor facilidad a la hora de crear y leer la información, se utilizará un formato de 24 horas.

**2.3 Convención para el registro de la información:** Una vez creada la estructura antes mencionada, se establecen las siguientes convenciones para programar la actividad que se realizará durante esa hora.

* **“0000000”:** Este arreglo se utilizará para determinar que es una hora programada para dormir.
* **“1111111”:** Este arreglo se utilizará para determinar que es una hora libre, en la cual no tiene programada ninguna actividad de estudio.
* **“2222222”:** Este arreglo se utilizará para determinar que es una hora programada para estudiar.
* **[Código de la materia]:** Este arreglo se utilizará para determinar que el estudiante tiene clase de la materia a la que corresponda dicho código.

Estas convenciones facilitaran la lectura de la información por parte del programa.

A continuación, se relaciona la representación de cómo se vería una matriz de la estructura planteada, utilizando las convenciones establecidas:

