

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DIVISIÓN DE INGENIERÍAS DEPARTAMENTO DE INFORMÁTICA



Seminario de solución de problemas de estructuras de datos II

Martes y jueves

Actividad 10

Actividad De Aprendizaje

ALUMNO: LOPEZ RODRIGUEZ CHRISTIAN ADRIAN

Código: 218022125

Ciclo Escolar: 2023-A

Profesor: Juan José Lopez Cisneros

Sección: D06

Horas

Efectivas 10:00

REPORTE DE ACTIVIDAD #A10 - Diversas técnicas de lectura/escritura de archivos... (Campos fijos)

A partir de la utilización sugerida de “*ifstream*” y “*ofstream*” realiza una función o un método que dado un *archivo binario de campos fijos* (Fig. 1.) realice lo siguiente:

1	Geometría Analítica	libro de texto	650
2	Programación básica	libro de texto	450.34
3	Álgebra lineal	libro de texto	620.10

Fig. 1. Ejemplo de archivo binario de campos fijos

- Lea carácter por carácter e inserte en los registros u objetos del programa.
- Lea línea por línea e inserte en los registros u objetos del programa

Desarrollo de la actividad:

```

4 #include <iostream>
5 #include <string.h>
6 #include "Prototipo.h"
7
8 void CD::Escribir(string var)
9 {
10     system("cls");
11     ofstream File(var, ios::binary);
12     char contReg[20];
13     if (!File)
14     {
15         cout << "Error al abrir el archivo" << endl;
16         return;
17     }
18
19     //Escribir encabezado de registros
20     sprintf(contReg, "%04d", cont);
21     File.write(contReg, 4);
22     // Escribir los datos en el archivo
23     for (int i = 0; i < cont; i++)
24     {
25         char artista[20], titulo[20], genero[20], fecha[20], codigo[20];
26
27         // Copiar los datos a los char arrays
28         strcpy(artista, this[i].Artista.c_str(), 20);
29         strcpy(titulo, this[i].Titulo.c_str(), 20);
30         strcpy(genero, this[i].Genero.c_str(), 20);
31         strcpy(fecha, this[i].Fecha.c_str(), 20);
32         sprintf(codigo, "%04d", this[i].Codigo);
33
34         // Escribir los datos en el archivo
35         File.write(artista, 20);
36         File.write(titulo, 20);
37         File.write(genero, 20);
38         File.write(fecha, 20);
39         File.write(codigo, 4);
40     }
41 }
42
43

```

```

ENCABEZADO 4
Tamaño del Archivo: 342
Número de bytes por registro: 84
Ginger Root
Sue
City Pop
2018
Codigo 1
Golden Ganja
Musica Libre
Reggae
2009
Codigo 2
1983 Omega Tribe
Super Chance
City Pop
1986
Codigo 3
Anri
Last Summer
City Pop
1990
Codigo 4
argv2 :Discos bin
Puede Salir?... 1.-SI/2.-No:

```

Imagen 1.1 Aquí observamos que se guarda un índice o encabezado de cuantos registros se tienen registrados esto una vez sabiendo cuantos registros se tienen que leer se pueden saber los bytes en donde se leerá el archivo y cuanto mide cada uno.

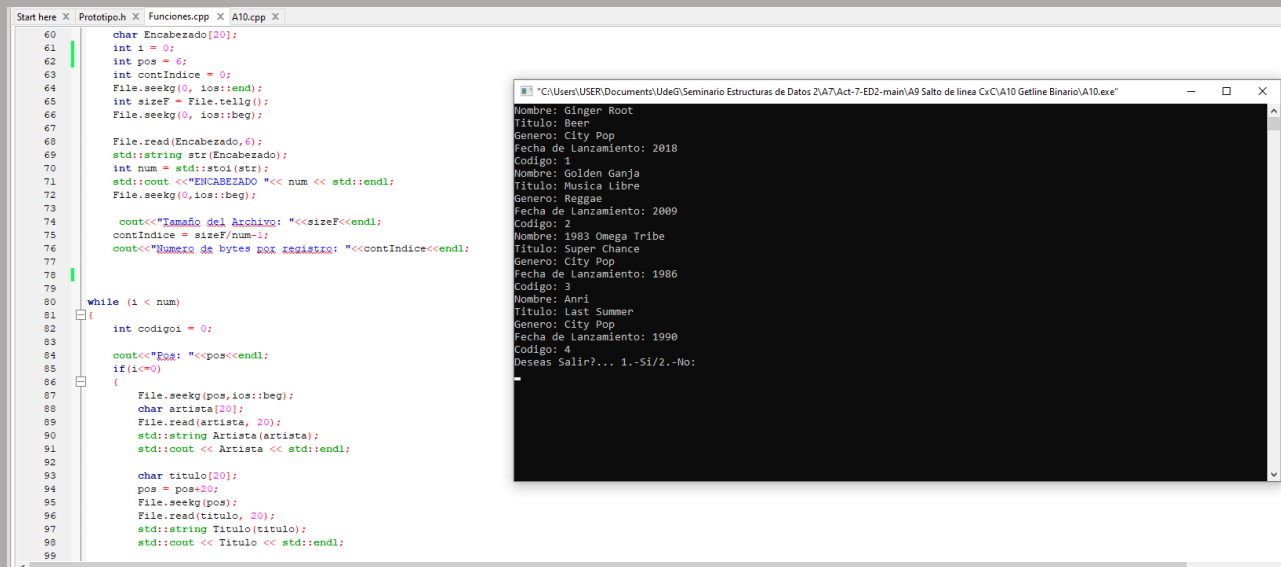


Imagen 1.2 Para ello vemos que se guarda en nuestra estructura en la que nosotros damos por hecho el formato para el que está hecho nuestra estructura así que nos saltamos los 6 primeros caracteres porque estos solamente contiene el índice de registros así que empezara a leer desde el byte numero 6 dándole cierto tamaño para que lea el cual nosotros sabemos de antemano iterando la posición y sumándola para así terminar de leer el archivo con la ayuda del contador de posiciones.

Problemas Enfrentados:

Realmente hubieron problemas pero nada que no se pueda resolver una vez teniendo en cuenta la estructura de un archivo binario el cual no es nada convencional pues uno de los problemas para la lectura es que leyera cantidad fija y no hasta que encontrara cierto carácter ya que aquí está leyendo caracteres hexadecimales y por ende el archivo cuenta con posiciones para que sea más fácil el posicionar cada registro de otra manera sería complicado. Así que la ayuda visual de los codificadores de texto fue obligatoria. Por lo que otro problema sería como guardar texto en nuestro archivo así que fue más simple que leerlos.

Conclusión:

Es una manera en la que se desperdicia mucha memoria en el archivo pero óptima para un formato preestablecido si es que la memoria para base de datos gigantes no es un problema, pero es importante mencionar que esto resultó ser una buena práctica para mi programación adaptando mi estilo a los problemas enfrentados ya que con una manera para guardar un string en la estructura solo quedaría por pensar una manera propia de implementar los algoritmos para la conversiones de los archivos y adaptarlos a la escritura y lectura para que este no involucre contenido basura.