

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 8

Actividad De Aprendizaje

ALUMNO: LOPEZ RODRIGUEZ CHRISTIAN ADRIAN

Código: 218022125

Ciclo Escolar: 2023-A

Profesor: Juan José Lopez Cisneros

Sección: D06

Horas

Efectivas 7:00

REPORTE DE ACTIVIDAD #A08 - Lectura/Escritura de archivos (el cierre)

Descripción de la Actividad:

A partir de la utilización sugerida de ***"ifstream"*** realiza una función o un método que,

- A. Lea un archivo de texto carácter por carácter e inserte en los registros u objetos del programa.
- B. Lea un archivo de texto línea por línea e inserte en los registros u objetos del programa

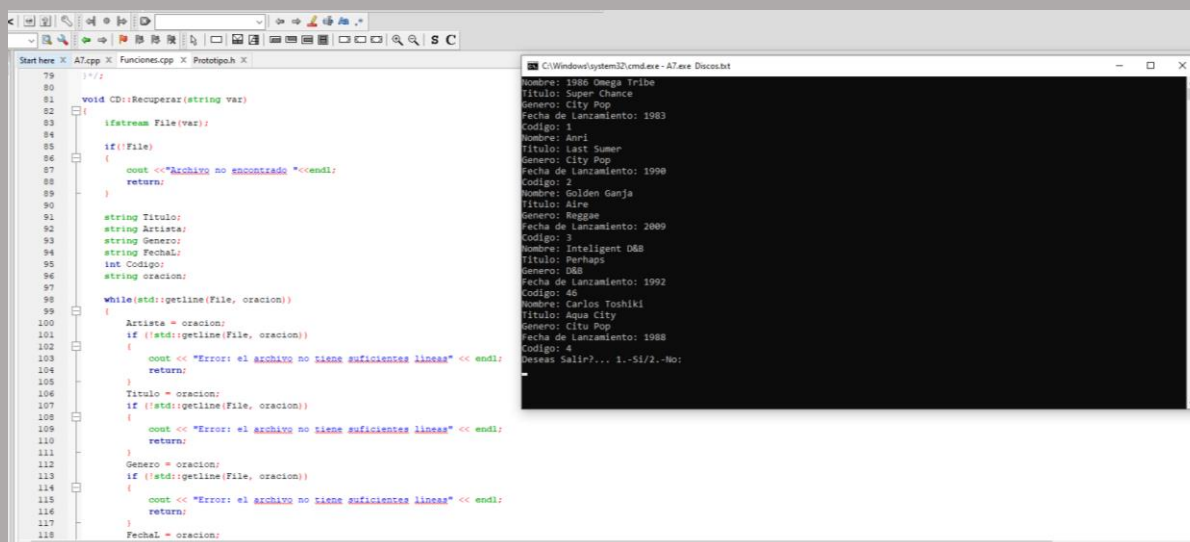
Estudia los modos de lectura en la apertura de un archivo con ***"ifstream"***.

Recuerda que depende del programa realizarás una función o un método que será llamado al principio de su ejecución

Una vez ingresado todos los productos, el programa continúa presentando el menú para que se puedan realizar las demás operaciones previamente programadas.

Objetivos de la actividad: Como objetivo principal tenemos uno el cual es aprender a implementar, saber como funciona los argumentos del main que son conocidos como ***"int argc, char *argv[]"*** que son argumentos de entrada a nuestro programa.

Desarrollo de la actividad:



```
79 //
80
81 void CD::Recuperar(string var)
82 {
83     ifstream File(var);
84
85     if(!File)
86     {
87         cout << "Archivo no encontrado " << endl;
88         return;
89     }
90
91     string Titulo;
92     string Artista;
93     string Genero;
94     string FechaL;
95     int Codigo;
96     string oracion;
97
98     while(!std::getline(File, oracion))
99     {
100         Artista = oracion;
101         if (!std::getline(File, oracion))
102         {
103             cout << "Error: el archivo no tiene suficientes lineas" << endl;
104             return;
105         }
106         Titulo = oracion;
107         if (!std::getline(File, oracion))
108         {
109             cout << "Error: el archivo no tiene suficientes lineas" << endl;
110             return;
111         }
112         Genero = oracion;
113         if (!std::getline(File, oracion))
114         {
115             cout << "Error: el archivo no tiene suficientes lineas" << endl;
116             return;
117         }
118         FechaL = oracion;
```

```
Nombre: 1986 Omega Tribe
Titulo: Super Chance
Genero: City Pop
Fecha de Lanzamiento: 1983
Codigo: 1
Nombre: Anri
Titulo: Last Summer
Genero: City Pop
Fecha de Lanzamiento: 1990
Codigo: 2
Nombre: Golden Ganja
Titulo: Aire
Genero: Ragga
Fecha de Lanzamiento: 2009
Codigo: 3
Nombre: Intelligent D&B
Titulo: Perhaps
Genero: D&B
Fecha de Lanzamiento: 1992
Codigo: 46
Nombre: Carlos Toshiki
Titulo: Aqua City
Genero: City Pop
Fecha de Lanzamiento: 1988
Codigo: 4
Deseas Salir?... 1.-Si/2.-No:
-
```

Imagen 1.1

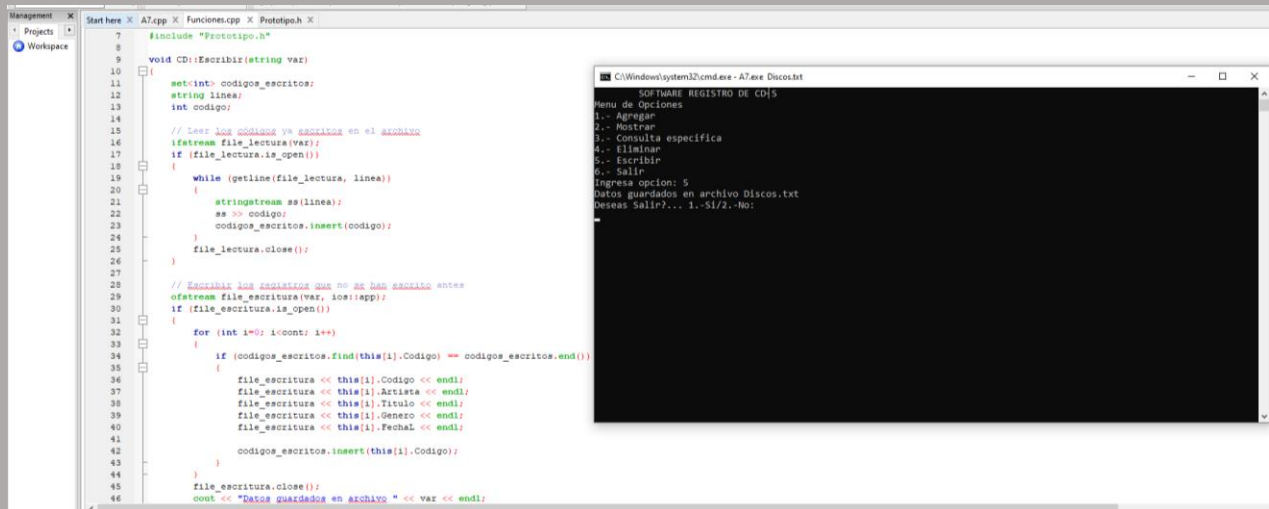


Imagen 1.2.- Véase que una vez se guardaron los archivos desde el cmd se ejecuta el menú del programa como se ve en la imagen podemos seleccionar opciones como aquí lo fue escribiendo en el archivo lo que antes podemos agregar.

Problemas Enfrentados:

Para todo lo anterior los principales problemas enfrentados fue encontrar lo principal que era saber con qué estábamos trabajando, usar argumentos del main y cómo eran guardados en el entero y arreglo, como sacar su uso, así que una vez entendiendo manipulamos como argumento para directorio del archivo. Guardar y escribir archivos fue otro de los problemas, conocer bien mi estructura y saber desde que clase implementar el algoritmo principal para cada correspondiente función.

Conclusión: Teniendo las funciones principales funcionando correctamente fue fácil poder regresar al menú para que el programa siguiera corriendo, también resulto interesante implementar un apartado en la función escribir para que omitiera los registros que están para que sea sobrescrito solo con los nuevos registros que ya en el menú ingresamos, así que para la estructura de la actividad 4 fue aún más fácil de implementar todo lo anterior ya que empecé a implementarlo primero al paradigma de clases.

Imagen 2.1- Creación del ejecutable por medio de la línea de comando.

