

UNIVERSIDAD DE GUADALAJARA

CUCEI

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

PRÁCTICA No. 1

TEMA: Archivos de Acceso Secuencial (Consultas Filtradas)

EQUIPO No. 4

INTEGRANTES:

Padilla Perez Jorge Daray

Munguía Guízar Marlon Uriel

Pérez Palacios Perla Michelle

Pulido Tobias Rafael Agustin

SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE ESTRUCTURAS DE DATOS II | SECCIÓN D19 | 2022B

PROF. Mariscal Lugo Luis Felipe

## **MARCO TEÓRICO**

**¿Qué es un archivo físico?**

Archivo que en realidad existe en el almacenamiento secundario. Es el archivo tal como lo conoce el sistema operativo y que aparece en su directorio de archivos.

**¿Qué es un archivo lógico?**

El archivo, visto por el programa. El uso de archivos lógicos permite a un programa describir las operaciones que van a efectuarse en un archivo sin saber cuál archivo físico se usará. El programa puede entonces usarse para procesar cualquiera de diversos archivos que comparten la misma estructura.

**¿Qué clases se utilizan en el manejo de archivos en C++?**

El archivo de cabecera fstream.h define las clases ifstream, ostream y fstream para operaciones de lectura, escritura y lectura/escritura en archivos respectivamente. Para trabajar con archivos debemos crear objetos de estas clases de acuerdo a las operaciones que deseamos efectuar. Empezamos con las operaciones de escritura, para lo cual básicamente declaramos un objeto de la clase ofstream, después utilizamos la función miembro open para abrir el archivo, escribimos en el archivo los datos que sean necesarios utilizando el operador de inserción y por último cerramos el archivo por medio de la función miembro close.

**¿Qué modos de apertura tiene C++?**

* **ios::app:** Añade toda la salida al final del archivo.
* **ios::ate:** Abre un archivo en modo de salida y se desplaza hasta el final del archivo (por lo general se utiliza para añadir datos a un archivo). Los datos se pueden escribir en cualquier parte del archivo.
* **ios::in:** Abre un archivo en modo de entrada.
* **ios::out:** Abre un archivo en modo de salida.
* **ios::trunc:** Descarta el contenido, si es que existe (también es la acción predeterminada para ios::out).
* **ios::binary:** Abre un archivo en modo de entrada o salida binaria (es decir, que no es texto).

**¿Para qué sirve la función eof()?**

Se usa para verificar si la secuencia del archivo genera algún error, eof (significa End Of File o Fin de Archivo).

Básicamente nos indica si hay o no más información que se pueda obtener del archivo, después de que se intenta extraer datos después del final del flujo.

**¿Para qué sirve la función seekg()?**

reposiciona el apuntador de posición del archivo y lo coloca al principio del archivo (ubicación 0) adjunto a archivoClientesEntrada. El argumento para seekg es comúnmente un entero long.

Se puede especificar un segundo argumento para indicar la dirección de búsqueda, que puede ser ios::beg (la opción predeterminada) para un posicionamiento relativo al inicio de un flujo, ios::cur para un posicionamiento relativo a la posición actual en un flujo, o ios::end para un posicionamiento relativo al final de un flujo.

El apuntador de posición del archivo es un valor entero que especifica la ubicación en el archivo como un número de bytes desde la ubicación inicial del archivo (a ésta también se le conoce como el desplazamiento desde el inicio del archivo). Algunos ejemplos de cómo posicionar el apuntador “obtener” de posición del archivo son:

* // se posiciona en el n-ésimo byte de objetoArchivo (asumiendo ios::beg) objetoArchivo.seekg( n );
* // se posiciona n bytes hacia adelante en objetoArchivo objetoArchivo.seekg( n, ios::cur );
* // se posiciona n bytes hacia atrás desde el final de objetoArchivo objetoArchivo.seekg( n, ios::end );
* // se posiciona al final de objetoArchivo objetoArchivo.seekg( 0, ios::end ); Se pueden realizar las mismas operaciones mediante la función miembro seekp de ostream.

**¿Para qué sirve la función clear()?**

Esta función nos permite restaurar el estado de un flujo a “bueno (Goodbit)” de manera que la entrada/Salida pueda proceder en el flujo.

**¿Para qué sirve la función is\_open()?**

Esta función revisa si un archivo está abierto, Devuelve si es que la secuencia está actualmente asociada a un archivo.

Los flujos se pueden asociar a archivos mediante una llamada exitosa a la apertura de miembros o directamente en la construcción, y se pueden desasociar llamando al cierre o la destrucción.

**Bibliografía:**

* Deitel, P. J., Deitel, P. J., & Elizondo, A. V. R. (2008). C++. Cómo programar (6.a ed.). Pearson Educación.
* C++ Reference. (2022). Cplusplus.com. <https://cplusplus.com/reference/fstream/fstream/is_open/#:~:text=std%3A%3Afstream%3A%3Ais_open&text=Returns%20whether%20the%20stream%20is,calling%20close%20or%20on%20destruction>.

**CÓDIGO FUENTE**

1 /\*

2 Equipo numero 4:

3 MunguÃ­a GuÃ­zar Marlon Uriel

4 PÃ©rez Palacios Perla Michelle

5 Padilla Perez Jorge Daray

6 Pulido Tovias Rafael Agustin

7

8 Seccion D19

9 Calendario 2022B

10 \*/

11

12 #include <iostream>

13 #include <fstream>

14 #include <iomanip> //left, setw, fixed, showpoint

15 #include <cstdlib> //exit

16 #include <conio.h>

17

18 **using namespace std**;

19

20 **void** mostrarLinea(**const char** \* **const**, **int**, **const char** \* **const**); //prototipos

21

22 **int** main()

23 {

24 **int** seleccion, edad;

25 **char** nombre [30];

26 **char** deporte [20];

27

28 **ifstream** archivoDeportistasEntrada; //archivo lÃ³gico

29 archivoDeportistasEntrada.open ("deportistas.txt", ios::in);

30

31 //salir del programa si no se puede crear el archivo

32 **if** (!archivoDeportistasEntrada) //operador ! sobrecargado

33 {

34 **cerr**<<"No se pudo abrir el archivo"<<**endl**;

35 exit(1);

36 }

37

38 **do**

39 {

40

41 system("cls");

42

43 **cout**<<"\n>>>>>>>>>> Menu Asociacion de Deportistas Tapatios <<<<<<<<<<\n"<<**endl**;

44 **cout**<<"1.- Mostrar deportistas menores de edad (5-17). "<<**endl**;

45 **cout**<<"2.- Mostrar deportistas mayores de edad(18-59). "<<**endl**;

46 **cout**<<"3.- Mostrar Mostrar deportistas menores de edad(60-85). "<<**endl**;

47 **cout**<<"0.- Salir. "<<**endl**;

48 **cout**<<"\n>>>>>>>>>>>>>>>>>>>>>> Elige tu opcion <<<<<<<<<<<<<<<<<<<<<<\n"<<**endl**;

49 **cout**<<**endl**;

50

51 **cin**>>seleccion;

52

53 archivoDeportistasEntrada.**clear**();//Limpia cin y establece el bit goodbit para permitir el flujo

54 archivoDeportistasEntrada.seekg(0);//Reposiciona el puntero al inicio del archivo

55

56 **switch**(seleccion)

57 {

58 **case** 0:

59 system("cls");

60 **cout**<<"\n>>>>>>>>>>>>>> Saliendo <<<<<<<<<<<<<<\n"<<**endl**;

61 getch();

62 **break**;

63

64 **case** 1:

65 system("cls");

66 **cout**<<"Deportistas menores de edad:"<<**endl**;

67 **cout**<<left<<**setw**(10)<<"NOMBRE"<<**setw**(13)<<"EDAD"<<"DEPORTE"<<**endl**<<fixed<<showpoint;//imprime la cabecera de la tabla

68

69 **while**(archivoDeportistasEntrada>>nombre>>edad>>deporte)

70 {

71 **if** (edad<18 && edad>4)

72 {

73 mostrarLinea(nombre,edad,deporte);

74 }

75

76 }

77

78 getch();

79 getch();

80 **break**;

81

82 **case** 2:

83 system("cls");

84 **cout**<<"Deportistas mayores de edad:"<<**endl**;

85 **cout**<<left<<**setw**(10)<<"NOMBRE"<<**setw**(13)<<"EDAD"<<"DEPORTE"<<**endl**<<fixed<<showpoint;//imprime la cabecera de la tabla

86

87 **while**(archivoDeportistasEntrada >> nombre >> edad >> deporte)

88 {

89

90 **if** (edad>17 && edad<60)

91 {

92 mostrarLinea(nombre,edad,deporte);

93 }

94

95 }

96

97 getch();

98 getch();

99 **break**;

100

101 **case** 3:

102 system("cls");

103 **cout**<<"Deportistas adultos mayores:"<<**endl**;

104 **cout**<<left<<**setw**(10)<<"NOMBRE"<<**setw**(13)<<"EDAD"<<"DEPORTE"<<**endl**<<fixed<<showpoint;//imprime la cabecera de la tabla

105

106 **while**(archivoDeportistasEntrada >> nombre >> edad >> deporte)

107 {

108

109 **if** (edad>59 && edad<86)

110 {

111 mostrarLinea(nombre,edad,deporte);

112

113 }

114

115 }

116

117 getch();

118 getch();

119 **break**;

120 }

121

122 }**while**(seleccion != 0);

123

124 **return** 0;

125 }//main

126

127

128 // muestra un solo registro del archivo

129 **void** mostrarLinea(**const char** \* **const** nombre, **int** edad, **const char** \* **const** deporte)

130 { **cout**<<left<<**setw**(10)<<nombre<<**setw**(13)<<edad<<**setw**(7)<<deporte<<**endl**;

131 }//mostrar linea

132

**IMAGEN(ES) DEL ARCHIVO(S) FÍSICO(S)**

Texto

Descripción generada automáticamente

**IMÁGENES DE LA CORRIDA DEL PROGRAMA**

Menú del programa.

Texto

Descripción generada automáticamente

Opción 1.

Texto

Descripción generada automáticamente

Opción 2.

Texto

Descripción generada automáticamente

Opción 3.

Texto

Descripción generada automáticamente

Opción salir.

Texto

Descripción generada automáticamente

**Conclusiones:**

* Munguía Guízar Marlon Uriel.

Esta primera práctica me pareció tanto interesante como a la vez educativa, ya que, al poder comenzar a escribir el código desde el aula con ayuda del libro de texto, no fue difícil comprender la forma de resolver la problemática.

* Pérez Palacios Perla Michelle.

Lo anterior nos permite aprender más sobre el manejo de archivos, pudiendo entender de mejor manera la entrada y salida de datos. También el ver los diferentes modos de apertura de un archivo y reforzar los conocimientos de las primeras tareas.

* Padilla Perez Jorge Daray.

Mi conclusión es que la práctica es un tanto sencilla, aunque te obliga e incita a estudiar más y más para mejorar tu programación, además de que me parece super importante el correcto aprendizaje de las funciones anteriormente vistas, me gusto la práctica y la explicación de la clase como apoyo para su correcta resolución.

* Pulido Tobias Rafael Agustin

Los archivos secuenciales tienen muchas aplicaciones desde el punto de guardar registros de información o bien extraer los datos, me quedó claro las limitantes que tiene un archivo de acceso secuencial. Por otro lado el aprender el funcionamiento de funciones como eof() ; o seekg() ; facilitan mucho el recorrer el archivo y manejar correctamente la información de este.