**Instituto Superior de Formación Técnica Nº 151 logo151-trans.gif  
Carrera: Analista de Sistemas  
1 Año. Algoritmos y Estructuras de Datos I**

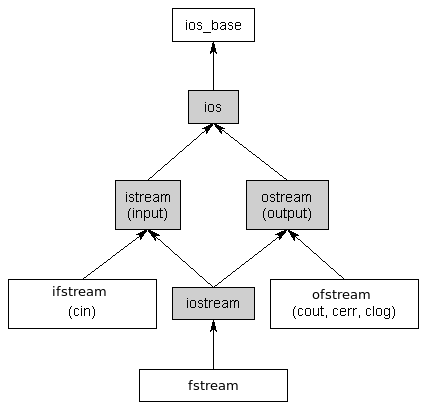
|  |  |
| --- | --- |
| **Trabajo Práctico Nº 3** | **Unidad 3.2** |
| **Modalidad:** Semi-Presencial | **Estratégica Didáctica:** Trabajo Grupal. |
| **Metodología de Desarrollo:** acordar | **Metodología de Corrección:** acordar docente |
| **Carácter de Trabajo:** Obligatorio – con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Marco Teórico:**

1. ¿Por qué necesitaría usar Archivos en una aplicación?
2. Explicar las diferencias entre archivos de texto y binarios (ampliar lo visto en clase)
3. ¿Qué entiende por “Flujos” en la lectura y Escritura?
4. ¿Explicar la librería y los objetos que se utilizan para leer y escribir el Archivo?
5. ¿Cómo deberemos poner el camino o “Path” del archivo a leer o escribir?
6. Explicar en un Diagrama la secuencia de comandos usados en la Escritura
7. Dar ejemplos de código “Descriptivamente Comentado” de Lectura de Archivos.
8. Dar ejemplos de código “Descriptivamente Comentado” de Escritura de Archivos.

**Marco Opcional Teórico:**

**Opcional 1**: Para los que quieren Ampliar les copio el “Modelo de Objetos” que representa la Familia FStream.



**Opcional 2** ¿Cómo puedo leer y analizar archivos CSV en C++?

**Opcional 3:** ¿Qué son los Archivos Binarios?

**Opcional 4 :** ¿Qué tipos de Archivos se pueden leer de forma binaria?

**Opcional 5:** ¿Qué tipos de Librerías de Objetos se deben usar?

**Opcional 6:** Dar ejemplo de Apertura y cierre de Archivos Binarios

**Marco Práctico:**  
Desarrollar en C++ los siguientes programas:  
(Se recomienda utilizar Funciones, Tipos definidos x el usuario y Enumeradores)  
(Leer y Reflexionar sobre las frases de *Bjarne Stroustrup* que se usarán en el archivo de Texto. )

1. Resolver y Realizar en c++   
   1. Crea un programa que vaya leyendo las frases que el usuario teclea y las guarde en un fichero de texto llamado “frasesDeBjarme.txt”. Terminará cuando la frase introducida sea "fin" (esa frase no deberá guardarse en el fichero).
   2. Para el ejercicio anterior, Abrir el archivo y mostrar las frases x pantalla, con cada frase en una línea, con Interlineado.
   3. Para el archivo del ejercicio (a) contar la cantidad de palabras que hay x línea, luego mostrar la cantidad total y el promedio de palabras por línea.
   4. Para el archivo de “frasesDeBjarme.txt” indicar cual es la palabra más Larga.

Lic. Oemig José Luis.

**Marco Teórico:**

1. ¿Por qué necesitaría usar Archivos en una aplicación?

El uso de archivos se hace necesario por varios motivos:

1. La posibilidad de almacenar datos.
2. Almacenar la salida de un programa.
3. Leer datos de manera masiva, sin tener que pedirlos uno a uno por teclado como hemos hecho hasta ahora.
4. Explicar las diferencias entre archivos de texto y binarios (ampliar lo visto en clase)

* **Archivos de Texto**: Son aquellos que pueden contener cualquier clase de datos y de tal manera que son “entendibles” por la gente. Los datos en un archivo de texto se almacenan usando el código ASCII, en el cual cada carácter es representado por un simple byte. Debido a que los archivos de texto utilizan el código ASCII, se pueden desplegar o imprimir. En este tipo de archivos, todos sus datos se almacenan como cadenas de caracteres, es decir, los números se almacenan con su representación ASCII y no su representación numérica, por lo tanto no se pueden realizar operaciones matemáticas directamente con ellos.
* **Archivos Binarios**: Este tipo de archivos almacenan los datos numéricos con su representación binaria. Pueden ser archivos que contienen instrucciones en lenguaje máquina listas para ser ejecutadas. Por ejemplo, cuando escribimos un programa en un lenguaje en particular (como C++, Pascal, Fortran, etc), tenemos las instrucciones almacenadas en un archivo de texto llamado programa fuente, pero una vez que lo sometemos a un proceso de compilación y ejecución nuestro programa lo trasladamos a un programa ejecutable (en lenguaje máquina), que es directamente entendido por la computadora y se crea un archivo binario. En este tipo de archivos también se pueden almacenar diferentes tipos de datos incluyendo datos numéricos; sin embargo, cabe destacar que los datos numéricos se graban con su representación binaria (no con su representación ASCII), por tal razón, cuando se despliegan con un editor de textos o por medio de comandos del sistema operativo, aparecen caracteres raros que no se interpretan. P. ejem. Si se guarda el dato 27 en un archivo binario, se almacena como 00001111 y no como “27”.

1. ¿Qué entiende por “Flujos” en la lectura y Escritura?

La E/S de C++ se da en flujos de bytes. Un flujo es simplemente una secuencia de bytes. En las operaciones de entrada, los bytes fluyen desde un dispositivo (por ejemplo un teclado, una unidad de disco) hacia la memoria principal. En operaciones de salida los bytes fluyen de la memoria principal hacia un dispositivo (por ejemplo una pantalla, una impresora, una unidad de disco).

Se podría pensar en un flujo de datos como una secuencia interminable de caracteres.

1. Explicar la librería y los objetos que se utilizan para leer y escribir el Archivo

El archivo de cabecera *fstream* define las clases *ifstream*, *ostream* y *fstream* para operaciones de lectura, escritura y lectura/escritura en archivos respectivamente.

**Se declara:**

*#include <fstream>*

1. Como deberemos poner el camino o “Path” del archivo a leer o escribir?

Para acceder al archivo *Prueba.txt* ubicado en la carpeta *C:\Test* el path se escribiría:

*C://Test//Prueba.txt*

1. Explicar en un Diagrama la secuencia de comandos usados en la Excritura

|  |  |
| --- | --- |
| *ofstream archivo;* | 1. Declara una variable de tipo ofstream |
| *archivo.open("output.txt");* | 2. Asocia la variable con el archivo de texto (crea el archivo) |
| *archivo << “Hola Mundo” << endl;* | 3. Realiza las escrituras por medio del operador << (insertor) |
| *archivo.close();* | 4. Desliga la variable del archivo de texto (cierra el archivo) |

1. Dar ejemplos de código “Descriptivamente Comentado” de Lectura de Archivos.

1. Dar ejemplos de código “Descriptivamente Comentado” de Escritura de Archivos.

**Opcional 3:** Que son los Archivos Binarios

Respondido en el punto 2.

**Opcional 4 :** ¿Qué tipos de Archivos se pueden leer de forma binaria?

Se puede leer de forma binaria todo tipo de archivos, desde texto, imágenes, audio, etc.

Estos archivos poseen una cabecera que describe el tipo de archivo (esto está directamente asociado con la extensión del archivo). Por ejemplo:

Cabeceras (headers) más utilizadas:

**Tipo de Archivo Cabecera En ASCII**

.ZIP 50 4B 03 04 PK

.RAR 52 61 72 21 Rar!

.TAR 1F 8B 08 00

.TGZ 1F 9D 90 70

.DOC D0 CF 11 E0 ÐÏ.à

.XLS D0 CF 11 E0

.PDF 25 50 44 46 %PDF

.WMV 30 26 B2 75

.FLV 46 4C 56 01 FLV

.BMP 42 4D F8 A9/ 62 25 / 76 03 BM, BMp% , BMv

.GIF 47 49 46 38 39 61 / 37 61 GIF89a GIF87a

.ICO 00 00 01 00

.JPEG FF D8 FF E0 / FE JFIF

.PNG 89 50 4E 47 PNG

.SFW 43 57 53 06 / 08 Cws

.MP3 49 44 33 2E /03 ID3

.EXE 4D 5A 50 00 /90 00 MZP / MZ

.DLL 4D 5A 90 00 MZ

Linux bin 7F 45 4C 46 ELF

**Opcional 5:** ¿Qué tipos de Librerías de Objetos se deben usar?

Los archivos binarios se tratan con la clase fstream.

**Opcional 6:** Dar ejemplo de Apertura y cierre de Archivos Binarios

Se adjunta un archivo llamado **ArchivosBIN.cpp** que, basándose en la información provista en la tabla anterior, lee la cabecera de una archivo binario y responde si el archivo en cuestión es efectivamente un archivo .bmp o no.