

**UNIVERSIADA PRIVADA DE TACNA
FACULTA DE INGENIERIA
ESCUOLA PORIFESIONAL DE INGENIERIA DE
SISITEMAS**



**EJERCICIOS_FICHEROS
INTREGRANTES**

- **EDUARDO MAMANI LAURA_2022074281**
- **OSCAR SALAS CORDERO_2023078695**
- **GOMEZ M. MIGUEL ANGEL_2018060901**

DOCENTE: MSc. Haydee Raquel Sisa Yataco

CURSO: ESTRUCTURA DE DATOS

Ejercicio 1:

ENUNCIADO

Escribe un programa en C++ que cree un archivo de texto llamado `datos.txt`. Luego, escribe tres líneas de texto en el archivo, donde cada línea contenga un mensaje diferente. Asegúrate de cerrar el archivo después de escribir en él.

PROGRAMA:

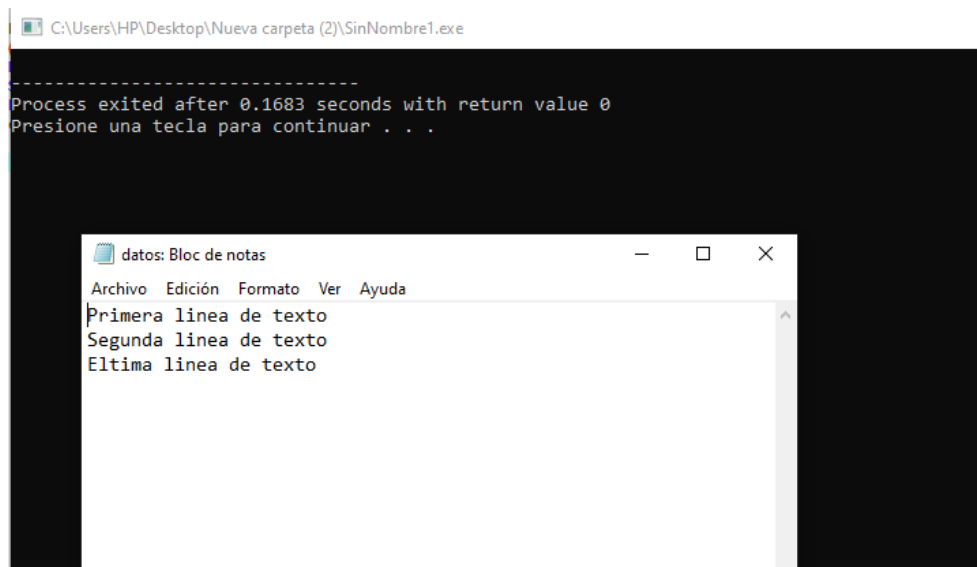
```
#include <fstream>
#include <iostream>
using namespace std;
int main()
{
    ofstream archivo; // objeto de la clase ofstream

    archivo.open("datos.txt");

    archivo << "Primera linea de texto" << endl;
    archivo << "Segunda linea de texto" << endl;
    archivo << "Última linea de texto" << endl;

    archivo.close();
    return 0;
}
```

EJECUCION:



EJERCICIO 2:

Este programa en C++ tiene como objetivo crear un archivo de texto llamado EJEMPLOS.TXT y escribir en él algunas líneas de texto con números, valores decimales y una calificación asociada. El programa utiliza la clase ofstream para manejar el flujo de salida hacia un archivo.

El programa realiza las siguientes operaciones:

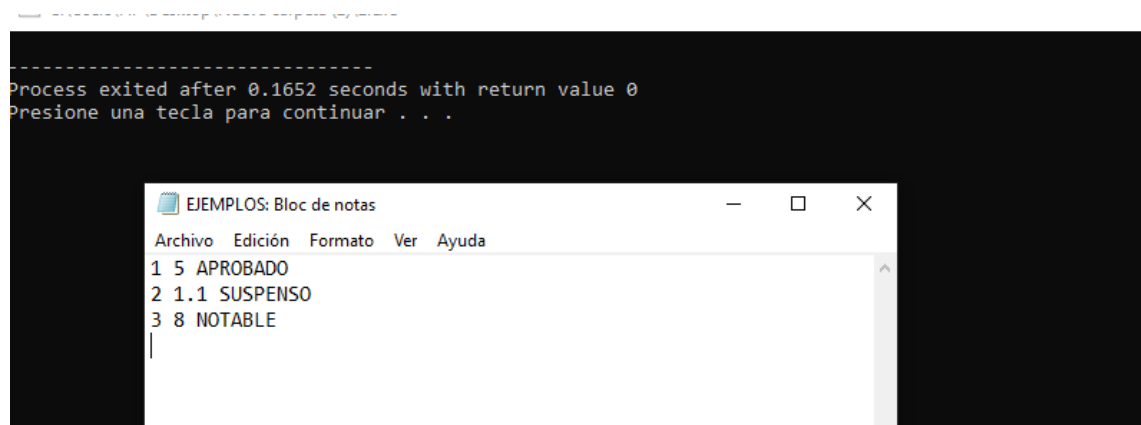
- Abre un archivo de texto llamado EJEMPLOS.TXT en modo de escritura.
- Si el archivo no se puede abrir, muestra un mensaje de error en la consola.
- Si el archivo se abre correctamente, escribe tres líneas de texto en el archivo. Cada línea contiene:
 - ✓ Un número entero.
 - ✓ Un valor decimal.
 - ✓ Una calificación en formato texto (APROBADO, SUSPENSO, NOTABLE).

Finalmente, cierra el archivo para guardar los cambios.

PROGRAMA:

```
#include <fstream> // Biblioteca para el manejo de ficheros
#include <iostream>
using namespace std; // Biblioteca para la entrada-salida estándar
int main(){
    ofstream fichout("EJEMPLOS.TXT",ios::out);
    if (!fichout)
        cout << "\n Incapaz de crear este o abrir el fichero \n";
    else {
        fichout << 1 << " " << 5.0 << " APROBADO" << endl; // Escritura en el fichero
        fichout << 2 << " " << 1.1 << " SUSPENSO" << endl;
        fichout << 3 << " " << 8.0 << " NOTABLE " << endl;
        fichout.close();
    }
} // Fin del main
```

EJECUCION:



EJERCICIO 3

ENUNCIADO

Este programa en C++ tiene como objetivo leer un archivo de texto llamado EJEMPLOS.TXT, procesar su contenido y mostrarlo en la consola. El archivo debe contener líneas con un número entero, un valor decimal y una cadena de texto (como una calificación). El programa utiliza la clase ifstream para manejar el flujo de entrada desde un archivo.


El programa realiza las siguientes operaciones:

- ✓ Abre el archivo EJEMPLOS.TXT en modo de lectura.
- ✓ Si no se puede abrir el archivo, muestra un mensaje de error indicando que no se pudo abrir el archivo.
- ✓ Si el archivo se abre correctamente, lee los datos del archivo línea por línea, procesando un número entero, un valor decimal y una cadena de texto por cada línea.
- ✓ Los datos leídos se muestran en la consola.
- ✓ El archivo se cierra después de haber sido procesado.

PROGRAMA

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  typedef char TCadena[30];
5
6  int main() {
7      int i;
8      float r;
9      TCadena cad;
10     ifstream fichin("EJEMPLOS.TXT");
11
12     if (!fichin) {
13         cout << "\n Incapaz de crear o abrir el fichero ";
14     } else {
15         fichin >> i;
16
17         while (!fichin.eof()) {
18             cout << i << " ";
19             fichin >> r;
20             cout << r << " ";
21             fichin >> cad;
22             cout << cad << "\n";
23             fichin >> i;
24         }
25         fichin.close();
26     }
27     return 0;
28 }
29
30
```

EJECUCION:

 C:\Users\HP\Desktop\Nueva carpeta (2)\3.exe

```
1 5 APROBADO
2 1.1 SUSPENSO
3 8 NOTABLE
```

```
-----
Process exited after 0.1739 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 04

- **Enunciado del ejercicio**

Crear un Programa en lenguaje C que permita al usuario:

1. Ingresar frases que se almacenarán en un archivo de texto llamado `frases.txt`.
2. Mostrar en pantalla las frases almacenadas en el archivo.
3. Salir del programa al presionar `Enter` en una línea vacía.

- **Código del programa.**

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     FILE* ficheroU;
6     char frase[61];
7     int i = 0;
8
9     ficheroU = fopen("frases.txt", "wt");
10    printf(" PROGRAMA para ESCRIBIR y almacenar FRASES.\n");
11    printf(" Cuando quiera salir, simplemente pulse \"Intro\".\n\n");
12
13    do {
14        if (i == 0)
15            puts("\nEscriba una FRASE:\n(o pulse \"Intro\").\n");
16        else
17            puts("\nEscriba otra FRASE:\n(o pulse \"Intro\").\n");
18        gets(frase);
19        fprintf(ficheroU, "%s\n", frase);
20        i++;
21    } while (strcmp(frase, "") != 0);
22
23    printf("He aqui lo que escribio:\n\n");
24    fclose(ficheroU);
25
26    ficheroU = fopen("frases.txt", "rt");
27    do {
28        fgets(frase, 60, ficheroU);
29        puts(frase);
30    } while (!feof(ficheroU));
31
32    getchar();
33    printf("...Hasta luego!");
34    getchar();
35    fclose(ficheroU);
36    return 0;
37 }
```

- **Ventana de ejecución.**

```
C:\Users\HP\Desktop\Ejercicio_04.exe

PROGRAMA para ESCRIBIR y almacenar FRASES.
Cuando quiera salir, simplemente pulse "Intro".

Escriba una FRASE:
(o pulse "Intro").

,e gusta aprender programACION

Escriba otra FRASE:
(o pulse "Intro").

...Hasta luego

Escriba otra FRASE:
(o pulse "Intro").
```

EJERCICIO_05:

- **Enunciado del ejercicio.**

El objetivo de este programa es permitir al usuario escribir frases que serán almacenadas en un archivo de texto llamado 'registroDeUsuario.txt'. El programa continuará pidiendo frases hasta que el usuario escriba la palabra 'fin', momento en el cual se detendrá y cerrará el archivo.

- **Código del programa.**

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      FILE* ptFichero;
6      char fin[] = "fin";
7      char frase[60];
8
9      ptFichero = fopen("registroDeUsuario.txt", "wt");
10     printf("PROGRAMA para ESCRIBIR FRASES.\nCuando quiera salir, escriba la palabra fin.\n\n");
11     do {
12         puts("\nEscriba una FRASE:\n(o fin). \n");
13         gets(frase);
14         if (strcmp(frase, fin) == 0)
15             break;
16         fprintf(ptFichero, "%s\n", frase);
17     } while (strcmp(frase, fin) != 0);
18
19     fclose(ptFichero);
20     return 0;
21 }
```

- **Ventana de ejecución.**

C:\Users\HP\Desktop\PROTECTO FINAL\Ejercicio_03.exe

```
PROGRAMA para ESCRIBIR FRASES.  
Cuando quiera salir, escriba la palabra fin.
```

```
Escriba una FRASE:  
(o fin).
```

*registroDeUsuario.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Al ejecutar el programa, la consola solicitará al usuario que introduzca frases. Por ejemplo:

```
Entrada:  
Hola mundo  
Este es un ejemplo  
fin  
|
```


ARBOL

Crear un programa en C++ que construya un Árbol Binario de Búsqueda (ABB), permita insertar nodos con valores enteros y realice los recorridos en orden, pre orden y post orden, mostrando el árbol de manera jerárquica.

Código C++ :

```
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4  struct nodo{
5      int nro;
6      struct nodo *izq, *der;
7  };
8  typedef struct nodo *ABB;
9  /* es un puntero de tipo nodo que hemos llamado ABB, que utilizaremos
10 para mayor facilidad de creacion de variables */
11 ABB crearNodo(int x)
12 {
13     ABB nuevoNodo = new(struct nodo);
14     nuevoNodo->nro = x;
15     nuevoNodo->izq = NULL;
16     nuevoNodo->der = NULL;
17     return nuevoNodo;
18 }
19 void insertar(ABB &arbol, int x)
20 {
21     if(arbol==NULL)
22     {
23         arbol = crearNodo(x);
24     }
25     else if(x < arbol->nro)
26         insertar(arbol->izq, x);
27     else if(x > arbol->nro)
28         insertar(arbol->der, x);
29 }
30 void preOrden(ABB arbol)
31 {
32     if(arbol!=NULL)
33     {
34         cout << arbol->nro <<" ";
35         preOrden(arbol->izq);
36         preOrden(arbol->der);
37     }
38 }
39 void enOrden(ABB arbol)
40 {
41     if(arbol!=NULL)
42     {
43         enOrden(arbol->izq);
44         cout << arbol->nro <<" ";
45         enOrden(arbol->der);
46     }
47 }
48 void postOrden(ABB arbol)
```

```

48 void postOrden(ABB arbol)
49 {
50     if(arbol!=NULL)
51     {
52         postOrden(arbol->izq);
53         postOrden(arbol->der);
54         cout << arbol->nro << " ";
55     }
56 }
57 void verArbol(ABB arbol, int n)
58 {
59     if(arbol==NULL)
60         return;
61     verArbol(arbol->der, n+1);
62     for(int i=0; i<n; i++)
63         cout<<" ";
64     cout<< arbol->nro <<endl;
65     verArbol(arbol->izq, n+1);
66 }
67 int main()
68 {
69     ABB arbol = NULL; // creado Arbol
70     int n; // numero de nodos del arbol
71     int x; // elemento a insertar en cada nodo
72     cout << "\n\t\t ..[ ARBOL BINARIO DE BUSQUEDA ].. \n\n";
73     cout << " Numero de nodos del arbol: ";
74     cin >> n;
75     cout << endl;
76     for(int i=0; i<n; i++)
77     {
78         cout << " Numero del nodo " << i+1 << ": ";
79         cin >> x;
80         insertar( arbol, x);
81     }
82     cout << "\n Mostrando ABB \n\n";
83     verArbol( arbol, 0);
84     cout << "\n Recorridos del ABB";
85     cout << "\n\n En orden : "; enOrden(arbol);
86     cout << "\n\n Pre Orden : "; preOrden(arbol);
87     cout << "\n\n Post Orden : "; postOrden(arbol);
88     cout << endl << endl;
89     system("pause");
90     return 0;
91 }

```

Compilación:

```

C:\Users\HP\Desktop\Ejercicio_Arbol.exe

..\[ ARBOL BINARIO DE BUSQUEDA ]..

Numero de nodos del arbol: 5

Numero del nodo 1: 2
Numero del nodo 2: 3
Numero del nodo 3: 6
Numero del nodo 4: 1
Numero del nodo 5: 9

Mostrando ABB

  9
 6
 3
2
1

Recorridos del ABB

En orden : 1 2 3 6 9

Pre Orden : 2 1 3 6 9

Post Orden : 1 9 6 3 2

Presione una tecla para continuar . . .

```