

## Entrada/ salida y compilación de programas en C++

- Escriba un programa que lea el precio de un producto y muestre el precio del producto al aplicarle el IVA. Defina una función que calcule el IVA y retorne el valor.
- Realice un programa que lea los siguientes datos de una persona: Edad: dato de tipo entero. Sexo: dato de tipo carácter. Altura en metros: dato de tipo flotante. Muestre los datos leídos. Desarrolle su programa usando funciones.
- Escriba un programa para guardar números telefónicos que muestre un menú con las siguientes opciones. Investigue como usar archivos de texto de tal manera que los contactos queden creados en un archivo de texto:
  1. Crear(nombre,apellidos,telefono)
  2. Agregar más contactos(nombre,apellidos,telefono)
  3. Visualizar contactos

## INFORME CODIGO:

### EJERCICIO 1:

```
#include <iostream>
#include <conio.h>

using namespace std;

// Definicion de funcion.

float aplicarIva(float valorProducto, float iva){

    float ivaFinal = iva / 100 ;
    float ivaAplicado = (valorProducto * ivaFinal) + valorProducto;

}

int main(){

    float valorProducto;
    float iva;

    cout << " Introduzca el precio del producto al que le aplicara el IVA: ";
    cin >> valorProducto;

    cout << " Introduzca el valor del iva a aplicar [Entero]: ";
    cin >> iva;

    cout << " El resultado del producto con iva incluido es " << aplicarIva(valorProducto, iva) << endl;
    // resultado = aplicarIva(iva, valorProducto);

    getch();
    return 0;
}
```

### Especificaciones:

- Realicé una función para calcular el IVA, donde recibía 2 variables [Valor del producto y el IVA]
- Se le pide al usuario la información, tanto del IVA como del valor del producto para que no se evalúe el IVA de 19 de Colombia, si no cualquier tipo que desee calcular.
- Incluí el getch al final para darle una pausa al programa.

## EJERCICIO 1:

```
13 using namespace std;
14
15 struct Datos{
16     int edad;
17     string sexo;
18     float altura;
19 }
20 }datos;
21
22 void agregarInformacion(){
23     int respuesta;
24
25     cout << "Digite su edad: ";
26     cin >> datos.edad;
27
28     cout << "Digite su sexo: ";
29     cin >> datos.sexo;
30
31     cout << "Digite su altura: ";
32     cin >> datos.altura;
33
34     cout << "\n";
35
36     cout << "Edad: " << datos.edad << endl;
37     cout << "Sexo: " << datos.sexo << endl;
38     cout << "Altura: " << datos.altura << endl << endl;
39 }
40
41 int main(){
42     agregarInformacion();
43
44     system("pause");
45     return 0;
46 }
```

### Especificaciones:

- Cree una struct donde guarda los valores de edad, sexo y altura de una persona.
- Realicé una función para pedir la información llamada "AgregarInformación".
- En la función creada pido la información a agregar en cada una de las variables del struct.
- Para finalizar la muestro en pantalla.

### EJERCICIO 3:

#### Información encontrada sobre Ficheros/Archivos:

(A) Como fichero de entrada: Para ello empleamos la sentencia

```
ifstream descriptor ("nombre.extensión");
```

(B) Como fichero de salida: Para ello empleamos la sentencia

```
ofstream descriptor ("nombre.extensión");
```

```
ifstream descriptor;           // Para ficheros de  
entrada
```

```
ofstream descriptor;          // Para ficheros de salida
```

*SEGUNDO: Asociamos el descriptor de ficheros al fichero en sí.*

```
descriptor.open("nombre.extensión",int modo);
```

Donde modo puede ser una combinación (empleando |) de lo siguiente:

```
ios:: in    // Modo entrada  
ios:: out   // Modo salida (por defecto)  
ios:: app   // Modo añadir, o sea, posicionar el cursor del fichero (ver  
abajo)  
            // al final del fichero antes de cada escritura  
ios:: binary // El archivo se abre en modo binario  
ios:: ate    // El archivo se abre y el cursor se posiciona al final  
ios:: nocreate // Genera un error si el fichero no existe  
ios:: noreplace // Genera un error si el fichero existe ya
```

### Detección de fin de fichero y otras funciones.

```
descriptor.función();
```

Donde *función* es una de las siguientes:

- La función *eof()* que devuelve un valor distinto de cero si se ha alcanzado el final del fichero.

#### **LECTURA ADELANTADA!!**

- La función *fail()*.
- La función *good()*

```

using namespace std;

struct Info{
    string nombre;
    string apellido;
    string telefono;
}info;

void crearContacto(){
    ofstream archivo;
    int respuesta;

    archivo.open("InformacionUsuarios.txt", ios::out);
    if(archivo.fail()){
        cout << "El archivo no se pudo crear" ;
        exit(1);
    }

    archivo << "\t Informacion Personal" << endl;

    do{
        fflush(stdin);
        cout << "Digite el nombre: ";
        getline(cin,info.nombre);

        cout << "Digite el apellido: ";
        getline(cin,info.apellido);

        cout << "Digite el telefono: ";
        getline(cin,info.telefono);

        cout << "\n";
    }
}

```

#### Especificaciones 1 Struct + Función “crearContacto”:

- Cree una struct donde guarda la información (nombre, apellido y teléfono).
- Realicé una función para pedir la información llamada crear contacto:
  - **ofstream archivo** -> Que es el que determina el archivo.
  - **archivo.open(“InformacionUsuarios.txt”, ios::out);** -> Abre el archivo en modo salida.
  - **archivo.fail** -> Lo que hace es verificar que el archivo se abra bien y en caso tal que no muestre el mensaje.
  - **Fflush(stdin)** -> Realiza la limpieza del buffer de entrada.

- Use un do while para pedir la información que desee agregar de un usuario nuevo en el archivo y en caso tal de querer agregar otro más que pueda hacerlo.

## Especificaciones 2 Función “agregarContacto”:

```
void agregarContacto(){
    ofstream archivo;
    int respuesta;

    archivo.open("InformacionUsuarios.txt", ios::app);

    if(archivo.fail()){
        cout << "El archivo no se pudo abrir correctamente";
        exit(1);
    }

    do{
        fflush(stdin);
        cout << "Digite el nombre: ";
        getline(cin,info.nombre);

        cout << "Digite el apellido: ";
        getline(cin,info.apellido);

        cout << "Digite el telefono: ";
        getline(cin,info.telefono);

        cout << "\n";

        archivo << "Nombre: " << info.nombre << endl;
        archivo << "Apellido: " << info.apellido << endl;
        archivo << "Telefono: " << info.telefono << endl << endl;

        cout << " Desea agregar otro contacto nuevo (1 = Si | 2 = No): " << endl;
        cin >> respuesta;
    }while(respuesta == 1);

    archivo.close();
}
```

- Archivo.open("InformacionUsuarios.txt, ios::app); -> Abre el archivo para que se pueda agregar información a este.
- Use un do while para pedir la información que desee agregar en el archivo y en caso tal de querer agregar alguno más que pueda hacerlo.
- Pedí la información a agregar en el do while, y luego la agregué en el archivo que ya hay.

### Especificaciones 3 Función "verInformacion":

```
void verInformacion(){  
    ifstream archivo;  
    string texto;  
  
    archivo.open("InformacionUsuarios.txt", ios::in);  
  
    if(archivo.fail()){  
        cout << "El archivo no se pudo abrir correctamente";  
        exit(1);  
    }  
  
    while(!archivo.eof()){  
        getline(archivo,texto);  
        cout << texto << endl;  
    }  
    archivo.close();  
}
```

- **Archivo.open("InformacionUsuarios.txt, ios::in);** -> Abre el archivo para que se pueda leer.
- Hice la condición para verificar que el archivo se abre correctamente.
- El ciclo while lo que hace es leer línea a línea, cuando llegue a una línea sin texto parará.  
**Eof -> End of line.**



### Especificaciones 3 Función "menu":

```
void menu(){
    int op;
    do{
        cout << "\t -----MENU----- " << endl;
        cout << " 1. Crear(nombre,apellidos,telefono) " << endl;
        cout << " 2. Agregar mas contactos(nombre,apellidos,telefono) " << endl;
        cout << " 3. Visualizar contactos existentes " << endl;
        cout << " 4. Salir" << endl;
        cout << "Opcion: ";
        cin >> op;

        switch(op){
            case 1: crearContacto();
                cout << "\n";
                system("Pause");
                break;

            case 2: agregarContacto();
                cout << "\n";
                system("Pause");
                break;

            case 3: verInformacion();
                cout << "\n";
                system("Pause");
                break;
        }
    }while(op != 4);
}
```

- Cree una función para el menu, donde imprime las opciones que hay y la opción que decida el usuario se almacenará en op.
- Uso un switch para hacer uso de las funciones que cree anteriormente y realizar lo que pida el usuario.
- Por último, la opción debe ser diferente a 4 para que no finalicé el programa.

Especificaciones 4 Función “verInformacion”:

```
int main(){  
    menu();  
  
    system("pause");  
    return 0;  
}
```

- Función main principal para ejecutar el menu y de ahí se despliega lo demás.