



MANUAL DE PROGRAMACIÓN EN C++

AUTORES:

Ing. Marcelo Sevilla V.

Ing. Valeria Valencia L.

Contenido

INTRODUCCIÓN	4
CONCEPTOS BÁSICOS	5
HISTORIA DE C++	5
ESTRUCTURA BÁSICA DE UN PROGRAMA EN C++	7
1.- Encabezado.	7
2.- Declaración de Librerías.	7
3.- Programa Principal.	7
Tipos de datos primitivos:	8
Tipos de datos definidos por el usuario:	9
SINTAXIS BÁSICA DEL LENGUAJE C++	9
Variable.	10
Constante.	11
PRIMER PROGRAMA EN C++.	11
EJERCICIOS.....	13
ESTRUCTURAS SELECTIVAS.....	16
ESTRUCTURAS DE REPETICIÓN.....	18
Estructura For:.....	19
Estructura While:.....	19
Estructura do – while:	19
EJERCICIOS PROPUESTOS	24
ESTRUCTURA DE UN PROGRAMA EMPLEANDO POO EN C++	61
1.- Encabezado	61
2.- Declaración de Librerías.....	61
3.- Codificación.....	62
4.- Programa Principal.....	62
PRIMER PROGRAMA EN C++.	62
1.- CLASE.....	64
2.-ATRIBUTOS.....	64
3.- MÉTODOS.....	64
4.-OBJETO.....	65
5.- ENCAPSULAMIENTO.....	74

6.- HERENCIA.....	83
7.- POLIMORFISMO.	94
ALGUNOS PROGRAMAS EN C++ UTILIZANDO LA TÉCNICA DE PROGRAMACIÓN ORIENTADA A OBJETOS.	103

SEINSEV

MANUAL DE PROGRAMACIÓN ESTRUCTURADA

INTRODUCCIÓN

El resultado de la investigación, los avances en el ámbito tecnológico se ven reflejados sin lugar a dudas, en las diferentes herramientas hardware y software que existen en la actualidad, son ellas la solución y respuesta a la necesidad de eficiencia y eficacia que los procesos en los diferentes ámbitos del saber demandan actualmente, una de las necesidades primarias es la necesidad de información, el proceso para obtenerla antes del desarrollo tecnológico mencionado, era arduo y engorroso, pues en directa dependencia del área, la obtención de información requería de largos procesos, análisis de datos, aplicación de fórmulas, cálculos que al realizarlos manualmente daban origen a un margen de error muy elevado, todos sabemos que, un cero sin tener valor por sí solo, dependiendo del lugar en el que se encuentre, marca una drástica y definitiva diferencia entre lo correcto, lo incorrecto, lo acertado y lo errado.

La organización y ordenamiento de los diferentes procesos que cada actividad requería, fue un importante paso en lo referente a eficiencia, pero persistían los errores y con ellos, arduas horas empleadas para revisiones que en varias ocasiones concluían con Tener que repetir el trabajo, la persistencia en la búsqueda de herramientas que permitan alcanzar la eficiencia y eficacia requerida, fue dando forma a lo que hoy conocemos como Programación, sin referirnos necesariamente a la programación en el área de informática.

Ahora que tenemos una breve idea del por qué y para que de la programación vale la pena el conocer el significado de la palabra Programación.

Programación: Es la Acción de Programar.

Programar: Es dar las instrucciones a una máquina para que realice su función de manera automática.

Con las dos definiciones se pretende aclarar el objetivo del presente manual de programación, es decir el presente manual permite introducirnos en los conceptos básicos de la Programación Orientada a Objetos P.O.O. empleando el lenguaje de programación C++.

El presente es el primer ejemplar de un conjunto de tres manuales, mismos que se constituyen en una guía de lo que es y se puede hacer empleando C++ y programación estructurada es importante tener en cuenta que los ejercicios que se describen más

adelante tienen infinitas formas de solución y desarrollo, pues la presente muestra solo una de ellas.

La programación no tiene límites, los limitantes los pone la imaginación y razonamiento del programador.

CONCEPTOS BÁSICOS

Lenguaje de Programación: es un conjunto de instrucciones “palabras clave, símbolos, reglas sintácticas” previamente definidas, mismas que permiten la interacción Hombre-Maquina, para escribir órdenes que serán ejecutadas por el computador.

Lenguaje C: es un lenguaje de programación fue creado por Brian Kernigham y Dennis Ritchie quien realizó la primera implementación a mediados de los 70, forma parte de los lenguajes estructurados, no es riguroso en el manejo de diferentes tipos de datos prueba de ellos es, permite conversión entre diferentes tipos de datos.

Lenguaje C++: es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

HISTORIA DE C++

C++ es heredero directo del lenguaje C que a su vez se deriva del lenguaje B [Richards, 1980]. C se mantiene como un subconjunto de C++. Otra fuente de inspiración de C++, como señala su autor Bjar-ne Stroustrup fue Simula 67, del que tomó el concepto de clase (con clases derivadas y funciones virtuales).

Aunque C, es un lenguaje muy potente, tiene dos características que lo hacen inapropiado como una introducción moderna a la programación.

Primero C, requiere un nivel de sofisticación a sus usuarios que les obliga a un difícil aprendizaje para los programadores principiantes, ya que es de comprensión difícil.

Segundo C, fue diseñado al principio de los setenta, y la naturaleza de la programación ha cambiado de modo significativo en la década de los ochenta y noventa.

Para subsanar estas «deficiencias» Bjarne Stroustrup de AT&T Bell Laboratories desarrolló C++ al principio de la década de los ochenta. Stroustrup diseñó C++ **como un mejor C**. En general, C estándar es un subconjunto de C++ y la mayoría de los programas C son también programas C++ (la afirmación inversa no es verdadera). C++, además de añadir propiedades a C, presenta características y propiedades de *programación orientada a objetos*, ASI TAMBIÉN C++ proporciona algunos recursos relativos a:

- **Abstracción de datos.**
- **Verificación estricta de tipos.**
- **Paso de argumentos por referencia.**
- **Limpieza automática de memoria.**
- **Inicialización garantizada**
- **Sobrecarga de operadores y funciones**
- **Tipos genéricos o parametrizados (template).**
- **Tratamiento de excepciones.**
- **Conversión de tipos.**

Se han presentado varias versiones de C++ y su evolución continúa en la actualidad. Las características más notables que han ido incorporándose a C++ son: herencia múltiple, genericidad, plantillas, funciones virtuales, excepciones, etc. C++ ha ido evolucionando año a año y como su autor ha explicado: *evolucionó siempre para resolver problemas encontrados por los usuarios y como consecuencia de conversaciones entre el autor, sus amigos y sus colegas*»⁹.

C++ comenzó su proyecto de estandarización ante el comité ANSI y su primera referencia es *The Annotated C++ Reference Manual*^W. En diciembre de 1989 se reunió el comité X3J16 del ANSI por iniciativa de Hewlett Packard. En junio de 1991 la estandarización de ANSI pasó a formar parte de un esfuerzo de estandarización ISO.

En 1995 se publicó un borrador de estándar para su examen y en 1998 se aprobó el estándar C++ internacional. Stroustrup publicó en 1998 la tercera edición de su libro *The C++ Programming Language* y en el año 2000 una edición especial. Este libro sigue el estándar ANSI/ISO C++.¹

¹ Luis Joyanes Aguilar, “Fundamentos de Programación, Algoritmos, Estructuras de datos y Objetos” Tercera edición 2003, página 31.

ESTRUCTURA BÁSICA DE UN PROGRAMA EN C++

Un programa básico escrito empleando el lenguaje de programación C++ consta de las siguientes partes:

1.- Encabezado.

Se lo emplea para incluir una descripción de lo que realiza el código implementado a continuación, la descripción va entre los sigos /* “Slash y asterisco” de igual manera para finalizar el comentario se emplea los mismos símbolos en orden diferente */ “asterisco y slash”.

2.- Declaración de Librerías.

En esta parte se incluye todas las librerías que se empleara para que el programa se ejecute correctamente, es importante anotar que cada una de las librerías tiene un uso y finalidad, cada librería brinda un conjunto de funcionalidades que facilitan el desarrollo de código es así que entre las librerías básicas y empleadas en este manual se encuentran:

- Iostream.h
- Stdio.h
- Conio.h
- String.h

La declaración de las librerías se la realiza empleando la siguiente estructura:

Símbolo numeral + palabra reservada “include” + símbolo menor que “<” + nombre de la librería + símbolo mayor que “>”

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

3.- Programa Principal.

En esta porción de código se encuentra entre las etiquetas “void main ()” + “{“+ instrucciones+”}”

Al hablar de programación estructurada se hace referencia a un tipo de programación en la que las instrucciones se siguen secuencial y ordenadamente en el mismo orden en que están implementadas, motivo por el que en este caso específico de programación, el

código de todos los programas descritos a continuación se encuentra dentro del programa principal void main ().

Esos son los 3 elementos básicos de un programa en C++.

TIPOS DE DATOS

El lenguaje de programación C++ consta de dos tipos de datos que son los primitivos y los derivados.

Tipos de datos primitivos:

Son los que están previamente definidos en el lenguaje de programación estos son enteros, reales, lógicos o booleanos, carácter, la siguiente tabala describe de mejor manera los tipos ded datos primitivos, teniendo en cuenta que en el presente se describen únicamente los tipos principales no los subtipos, para de esta manera evitar caer en confusiones y ambigüedades.

Tabla 1: Tipos de datos primitivos

	Tipo de Dato	Descripción	Rango
Tipo Entero	Short	Entero corto	-32768 a 32767
	Int	Entero	-2147483648 a 2147483647
	Long	Entero largo	-2147483648 a 2147483647
	char	Carácter ASCII	-128 a 127
Tipo Real	Float	Real con coma flotante	Positivos: 3.4E-38 a 3.4E38 Negativos: -3.4E a -3.4E38
	Double	Real doble, con coma flotante de doble precisión	Positivos: 1.7E-308 a 1.7E308 Negativos: -1.7E-308 a -1.7E308
	Long double	Real doble largo	Positivos: 3.4E-4932 a 1.1E4932 Negativos: -3.4E-4932 a -1.1E4932
Tipo Lógico	booleano	Datos lógicos	0, 1
Tipo Carácter extendido	Wchart_t	Carácter Unicode	0 a 65535
Tipo void	Void	Dato sin tipo	No se puede declarar una variable de este tipo se lo

			emplea para decir que no se retorna ningún valor
	Void *	Puntero a un espacio de memoria	Se puede declarar una variable puntera a void
Tipos de la Biblioteca Estandar	String	Tipo no propio del lenguaje C++;	Tipo definido en la biblioteca estándar std::string para manejo de caenas de caracteres de 8 bits
	wstring	Tipo no propio del lenguaje C++;	Tipo definido en la biblioteca estándar std::wstring para manejo de caenas de caracteres de 16 bits

Tipos de datos definidos por el usuario:

Como su nombre lo indica son tipos de datos que el usuario declara partiendo de otros elementos de tipo clase class, struct, o enum; este tema se verá con más detalle en el Manual de Programación Orientada a Objetos POO.

SINTAXIS BÁSICA DEL LENGUAJE C++

Para el presente manual se emplean un conjunto de instrucciones propias de lenguaje de programación, cada una de ellas tiene un significado o función específica en base a la forma en la que se emplee, se detallan en la siguiente tabla:

Tabla 2: Sintaxis básica del lenguaje C++

Palabra Reservada	Descripción	Observación
asm	Inserta instrucciones de ensamblador en línea	
Auto	Declaración de uan variable para almacenamiento automático	
Bool	Tipo de datos booleano	Tipo de dato
Break	Salir abruptamente de un bucle	
Case	Define un caso de la estructura selectiva switch	
Catch	Captura una excepción	
Char	Tio de datos carácter	Tipo de dato
Class	Permite declarar una clase	
Const	Permite declarar una constante	
Continue	Permite dar un salto para no ejecutar un bucle	

Default	Caso por dedfecto de un conjunto de casos e instrucciones	
Delete	Permite liberar memoria	
Do	Prmite implemenatr un bucle do – while	Estructura ded repeticion
Double	Tipo de datos de coma flotante	Tipo de dato
Else	caso contrario “sino” de una sentencia if “si”	
False	Valor booeano falso	
If	Sentencia de selección simple “si”	Estructura de selección simple
Int	Tipo de dato entero	Tipo ded dato
Long	Tipo de dato entero largo	Tipo de dato
New	Operador que perite crear un objeto	Operador
Private	Permite declara un elemnto privado	
Protected	Permite declara un elemento como protegido	
Public	Declara un elemento como público	
Return	Permite retorna un valor	
Short	Tipo de dato entero corto	Tipo de dato
Signed	Declaración de un tipo entero como signo	
Sizeof	Permite obtener el tamaño de un tipo de dato	
Static	Reserva almacenamiento permanente	
Struct	Permite hacer la definición de una estructura	
Switch	Estructura de selección multiple	
This	Operador que señala el objeto actual de una clase	Operador
True	Valor booleano verdadero	
Typedef	Permite crear alias para datos de un determinado tipo	
Unsigned	Permite declarar una variable sin signo	
Virtual	Permite crear un metodo que puede ser reemplazado por calses derivadas	
Void	Sin tipo, se emplea para declarar funciones que no devuelven nada	Sin tipo
Volatile	Es un factor externoal programa	
Wchar_t	Tipo de datos carácter	
While	Permite declara una estructura while y tabien una estructura do-while	Estructura de repetición

Variable.

Es un espco en memoria que almacena un determinado valor especificado en su declaración “tipo y nombre” ejemplo:

Tabla 3: Ejemplo de declaración de variables

Declaración sin asignación	Declaración con asignación “inicialización de la variable”
Int suma;	Int suma=1;
Char nombre;	Char nombre=” Erick”
Float porcentaje;	Float porcentaje=7,778

Nota: los nombres de las variables no pueden ser palabras reservadas, se recomienda en uso de nombres descriptivos para de esta manera evitar confusiones y ambigüedades al momento de realizar cambios, actualizaciones o dar un mantenimiento al código.

Constante.

Se denomina constante a un valor que permanece sin cambios ni alteraciones durante toda la ejecución del programa, este valor se lo puede emplear en cualquier proceso. Ejemplo.

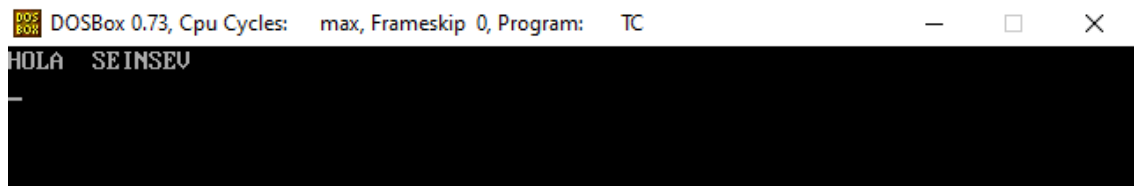
```
const float constgrav = 9.8;
```

PRIMER PROGRAMA EN C++.

Escribir el código de un programa que permita mostrar en la pantalla “HOLA SEINSEV”.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    cout<<”HOLA SEINSEV”<<endl;
    getch();
};
```

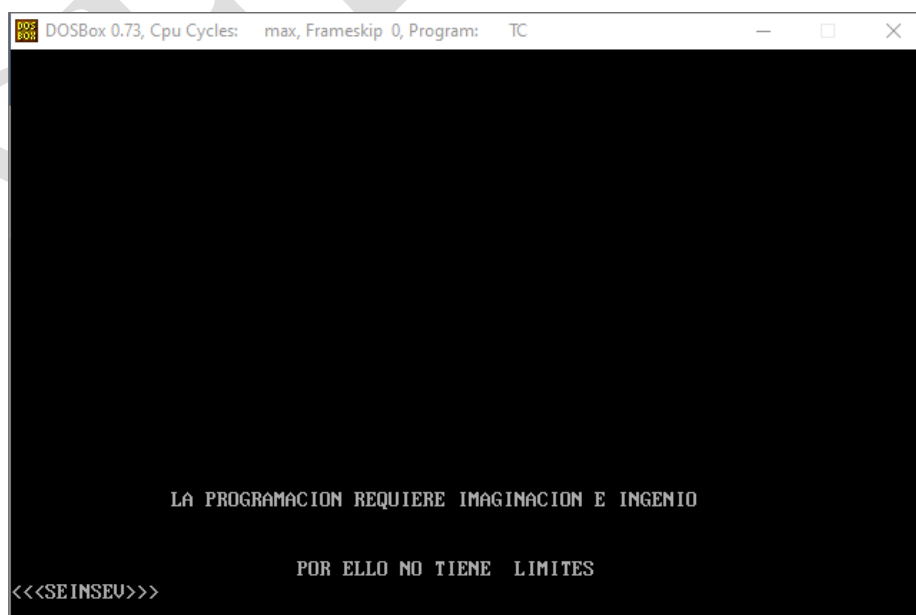
Al ejecutar el código anterior se muestra la salida en pantalla sería la que se muestra en la siguiente imagen:



Escribir el código de un programa que permita mostrar en pantalla
“LA PROGRAMACION REQUIERE IMAGINACIÓN E INGENIO”
“POR ELLO NO TIENE LIMITES”
“SEINSEV”

```
#include <iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
clrscr();
gotoxy(15,20);
cout<<"LA PROGRAMACION REQUIERE IMAGINACION E INGENIO"<<endl;
gotoxy(25,23);
cout<<" POR ELLO NO TIENE LIMITES"<<endl;
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

Al ejecutar el código anterior se muestra la salida en pantalla sería la que se muestra en la siguiente imagen:



EJERCICIOS

1.- Implementar un programa que permita ingresar tres números por teclado y que realice la suma de ellos, mostrando el resultado en pantalla.

/*EL PRESENTE PROGRAMA PERMITE INGRESAR TRES DE NUMEROS Y REALIZAR LA SUMA DE LOS MISMOS*/

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
    clrscr();
    int num1,num2,num3,suma;
    clrscr();
    cout<<"INGRESE EL PRIMER NUMERO"<<endl;
    cin>>num1;
    clrscr();
    cout<<"INGRESE EL SEGUNDO NUMERO"<<endl;
    cin>>num2;
    clrscr();
    cout<<"INGRESE EL TERCER NUMERO"<<endl;
    cin>>num3;
    suma=num1+num2+num3;
    clrscr();
    gotoxy(5,30);
    cout<<"EL RESULTADO DE SUMAR LOS VALORES INGRESADOS ES:
    "<<suma<<endl;
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

2.-Implementar un programa que permita ingresar dos números por teclado minuendo, sustraendo y realice la resta entre ellos, mostrando el resultado en pantalla.

/*EL PRESENTE PROGRAMA PERMITE INGRESAR DOS NUMEROS Y REALIZAR LA RESTA DE LOS MISMOS*/

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
    clrscr();
    int num1,num2,resta;
    clrscr();
```

```
cout<<"INGRESE EL PRIMER NUMERO"<<endl;
cin>>num1;
clrscr();
cout<<"INGRESE EL SEGUNDO NUMERO"<<endl;
cin>>num2;
resta=num1-num2;
clrscr();
gotoxy(20,30);
cout<<"EL RESULTADO DE RESTAR LOS VALORES INGRESADOS ES:
      "<<resta<<endl;
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

3.- Implementar un programa que permita ingresar dos números, multiplicando, multiplicador y realizar la multiplicación de los mismos, mostrando en pantalla el resultado.

```
/*EL PRESENTE PROGRAMA PERMITE INGRESAR DOS NUMEROS Y
   REALIZAR LA MULTIPLICACION DE LOS MISMOS*/
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
  clrscr();
  int num1,num2,mult;
  clrscr();
  cout<<"INGRESE EL PRIMER NUMERO"<<endl;
  cin>>num1;
  clrscr();
  cout<<"INGRESE EL SEGUNDO NUMERO"<<endl;
  cin>>num2;
  mult=num1*num2;
  clrscr();
  gotoxy(20,30);
  cout<<"EL RESULTADO DE MULTIPLICAR LOS VALORES INGRESADOS ES:
        "<<mult<<endl;
  gotoxy(30,50);
  cout<<"<<<SEINSEV>>>"<<endl;
  getch();
};
```

4.- Implementar un programa que permita ingresar dos números dividiendo, divisor y realice la división entre ellos, mostrando en pantalla el resultado.

```
/*EL PRESENTE PROGRAMA PERMITE INGRESAR DOS NUMEROS Y
   REALIZAR LA DIVISION ENTRE LOS MISMOS*/
```

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
    clrscr();
    float num1,num2,divi=0;
    clrscr();
    cout<<"INGRESE EL PRIMER NUMERO"<<endl;
    cin>>num1;
    clrscr();
    cout<<"INGRESE EL SEGUNDO NUMERO"<<endl;
    cin>>num2;
    divi=num1/num2;
    clrscr();
    gotoxy(20,30);
    cout<<"EL RESULTADO DE DIVIDIR LOS VALORES INGRESADOS ES:
        "<<divi<<endl;
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

5.- Implementar un programa que permita ingresar un numero y obtener la raíz cuadrada del mismo, mostrando en pantalla el resultado.
/*EL PRESENTE PROGRAMA PERMITE OBTENER LA RAIZ CUADRADA DE UN NUMERO INGRESADO*/

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
void main()
{
    clrscr();
    float num=0;
    cout<<"INGRESE EL NUMERO"<<endl;
    cin>>num;
    gotoxy(20,30);
    cout<<"La raiz cuadrada del numero ingresado es; "<<sqrt(num)<<endl;
    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    getch();
};
```

ESTRUCTURAS SELECTIVAS

En C++ existen dos estructuras, una de selección simple y una de selección múltiple:

Estructura de selección simple “IF-THEN” (Si **condición** entonces): la presente permite seleccionar una opción ejecutando un conjunto de instrucciones siempre y cuando se cumpla la condición que consta en el if, es importante mencionar que esta estructura tiene además una variante que es “ELSE” (caso contrario), cuando no se cumple la condición que consta en el if pues entonces se ejecuta las condiciones que se encuentran definidas en el else o (caso contrario) ejemplo:

6.- Implementar un programa que permita determinar si un número ingresado es par o Impar.

```
/*EL PRESENTE PROGRAMA PERMITE DECIR SI UN NUMERO INGRESADO
   ES PAR O IMPAR*/
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int num,a=1;
    cout<<"Ingrese el numero"<<endl;
    cin>>num;
    a=(num-((num/2)*2));
    if(a==0)
    {
        cout<<"EL NUMEROINGRESADO ES PAR"<<endl;
    }
    else
    {
        cout<<"EL NUMEROINGRESADO ES IMPAR"<<endl;
    };
    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    getch();
};
```

Estructura de selección múltiple “SWITCH”: La presente estructura de selección permite seleccionar una de un conjunto de opciones presentes, ejemplo:

7.- Implementar un programa que permita ingresar el mes en números y muestre en

pantalla el mes al que corresponde es decir si ingresa 8, debería aparecer el mes es agosto.

/*EL PRESENTE PROGRAMA MUESTRA EL MES INGRESANDO SU CORRESPONDIENTE NÚMERO*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int op=0;
    cout<<"Ingrese el numero del mes"<<endl;
    cin>>op;
    switch(op)
    {
        case 1:
        {
            cout<<"El mes seleccionado es: ENERO"<<endl;
            getch();
        }break;
        case 2:
        {
            cout<<"El mes seleccionado es: FEBRERO"<<endl;
            getch();
        }break;
        case 3:
        {
            cout<<"El mes seleccionado es: MARZO"<<endl;
            getch();
        }break;
        case 4:
        {
            cout<<"El mes seleccionado es: ABRIL"<<endl;
            getch();
        }break;
        case 5:
        {
            cout<<"El mes seleccionado es: MAYO"<<endl;
            getch();
        }break;
        case 6:
        {
            cout<<"El mes seleccionado es: JUNIO"<<endl;
            getch();
        }break;
```

```
case 7:
{
    cout<<"El mes seleccionado es: JULIO"<<endl;
    getch();
}break;
case 8:
{
    cout<<"El mes seleccionado es: AGOSTO"<<endl;
    getch();
}break;
case 9:
{
    cout<<"El mes seleccionado es: SEPTIEMBRE"<<endl;
    getch();
}break;
case 10:
{
    cout<<"El mes seleccionado es: OCTUBRE"<<endl;
    getch();
}break;
case 11:
{
    cout<<"El mes seleccionado es: NOVIEMBRE"<<endl;
    getch();
}break;
case 12:
{
    cout<<"El mes seleccionado es: DICIEMBRE"<<endl;
    getch();
}break;
}
clrscr();
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};
```

ESTRUCTURAS DE REPETICIÓN

En C++ existen un conjunto de estructuras de repetición, cadauna se emplea con un fin específico, las mas utilizadas son:

Estructura For:

Es una estructura de repetición que permite ejecutar un conjunto de instrucciones predefinidas, un numero determinado de veces, para ello define un inicio, un valor en el que se incrementa o decrementa y un valor final hasta el que debe repetirse las instrucciones que se encuentran dentro del ciclo, la estructura del ciclo for en C++ es la siguiente: for (int i=0; i<=10; i++)

Tabla 4: Estructura for

Sentencia for	Declaración de la variable i	Definición del inicio del contador	Condición	Incremento de la variable "Contador"
For	(int i	=0	I<=10	i++)

Estructura While:

Es una estructura de repetición que permite ejecutar un conjunto de instrucciones mientras se cumpla una condición pre establecida, dentro del código de la sentencia while es indispensable el que este implementada la condición de parada es decir la línea que en algún momento dará por finalizado el ciclo de repetición, antes de ejecutar el código que se encuentra dentro del ciclo primero se evalua si se cumple la condicion, la estructura del ciclo de repetición while es el siguiente:

Tabla 5: Estructura While

Sentencia while	Condición	Condición de parada
while	(a<=10)	{ a=a+1 }

Nota: la condición de parada va dentro del código a repetir en el bucle.

Estructura do – while:

Es una estructura de repetición que permite ejecutar un conjunto de instrucciones mientras se cumpla una condición, es importante anotar que este tipo de sentencia permite se ejecute el código del bucle al menos una vez, pues primero ejecuta el conjunto de sentencias dentro del bucle y posteriormente analiza la condición, la estructura del ciclo do -while es:

Tabla 6: Estructura do – while

Sentencia do	Conjunto de instrucciones y condición de parada	Condición
do	{ a=a+1 }	While (a<=10)

Nota: la condición de parada va dentro del código a repetir en el bucle.

8.- Implementar un programa que muestre un menú en el que se encuentren las 4 operaciones fundamentales suma, resta, multiplicación, división y además la opción raíz cuadrada, para de esta manera ingresar los valores requeridos y mostrar el resultado en pantalla, emplear la estructura de selección Switch

```
/*EL PRESENTE PROGRAMA MUESTRA UN MENU QUE PERMITE  
REALIZARLAS 4 OPERACIONES FUNDAMENTALES SUMA, RESTA,  
MULTIMPLICACION, DIVISION Y RAIZ CUADRADA*/
```

```
#include<iostream.h>  
#include<conio.h>  
#include<stdio.h>  
#include<math.h>  
void main()  
{  
clrscr();  
int op1=0;  
do  
{  
int op=0;  
cout<<"ELIJA UNA OPCION"<<endl;  
cout<<"1.- SUMAR VARIOS NUMEROS "<<endl;  
cout<<"2.- RESTAR DOS NUMEROS INGRESADOS"<<endl;  
cout<<"3.- MULTIPLICAR VARIOS NUMEROS "<<endl;  
cout<<"4.- DIVIDIR DOS NUMEROS INGRESADOS"<<endl;  
cout<<"5.- OBTENER LA RAIZ CUADRADA DE UN NUMERO  
INGRESADO"<<endl;  
cin>>op;  
switch(op)  
{  
case 1:  
{  
int numval=0;  
cout<<"Ingresa el numero de valores a ser sumados"<<endl;  
cin>>numval;  
float suma=0;  
for(int i=1;i<=numval;i++)  
{  
float num=0;  
cout<<"Ingresa el valor "<<i<<endl;  
cin>>num;  
suma=suma+num;  
}  
cout<<"El resultado de sumar los valores ingresados es: "<<suma<<endl;  
getch();  
}break;  
case 2:  
{
```

```
float num1,num2,resta;
cout<<"Ingrese el primer numero"<<endl;
cin>>num1;
cout<<"Ingrese el segundo numero"<<endl;
cin>>num2;
resta=num1-num2;
clrscr();
cout<<"EL RESULTADO DE RESTAR LOS VALORES INGRESADOS ES:
"<<resta<<endl;
}break;
case 3:
{
    int numval=0;
    cout<<"Ingrese el numero de valores a ser multiplicados"<<endl;
    cin>>numval;
    float mult=1;
    for(int i=1;i<=numval;i++)
    {
        float num;
        cout<<"Ingrese el valor "<<i<<endl;
        cin>>num;
        mult=mult*num;
    }
    cout<<"El resultado de multiplicar los valores ingresados es: "<<mult<<endl;
    getch();
}break;
case 4:
{
    float num1,num2,divi;
    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    cout<<"Ingrese el primer numero"<<endl;
    cin>>num1;
    cout<<"Ingrese el segundo numero"<<endl;
    cin>>num2;
    divi=num1/num2;
    cout<<"EL RESULTADO DE DIVIDIR LOS VALORES INGRESADOS ES:
"<<divi<<endl;
}break;
case 5:
{
    clrscr();
    float num=0;
    cout<<"INGRESE EL NUMERO"<<endl;
    cin>>num;
    cout<<"La raiz cuadrada del numero ingresado es: "<<sqrt(num)<<endl;
    getch();
```

```

        }break;
    }
    cout<<"SI DESEA CONTINUAR PRESIONE 0, SI DESEA SALIR PRESIONE
        1"<<endl;
    cin>>op1;
    clrscr();
    }while (op1==0);
    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    getch();
};

```

9.- Implementar un programa que muestre un menú en el que se encuentren las 4 operaciones fundamentales suma, resta, multiplicación, división y además la opción raíz cuadrada, para de esta manera ingresar los valores requeridos y mostrar el resultado en pantalla, emplear la estructura de selección IF.

```

/*EL PRESENTE PROGRAMA MUESTRA UN MENU QUE PERMITE
REALIZARLAS 4 OPERACIONES FUNDAMENTALES SUMA, RESTA,
MULTIMPLICACION Y DIVISION*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int op1=0;
    do
    {
        int op=0;
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- SUMAR VARIOS NUMEROS "<<endl;
        cout<<"2.- RESTAR DOS NUMEROS INGRESADOS"<<endl;
        cout<<"3.- MULTIPLICAR VARIOS NUMEROS "<<endl;
        cout<<"4.- DIVIDIR DOS NUMEROS INGRESADOS"<<endl;
        cout<<"5.- OBTENER LA RAIZ CUADRADA DE UN NUMERO
            INGRESADO"<<endl;
        cin>>op;
        if (op==1)
        {
            int numval=0;
            cout<<"Ingrese el numero de valores a ser sumados"<<endl;
            cin>>numval;
            float suma=0;
            for(int i=1;i<=numval;i++)
            {

```

```
float num=0;
cout<<"Ingrese el valor "<<i<<endl;
cin>>num;
suma=suma+num;
}
cout<<"El resultado de sumar los valores ingresados es: "<<suma<<endl;
getch();
};
if (op==2)
{
float num1,num2,resta;
cout<<"Ingrese el primer numero"<<endl;
cin>>num1;
cout<<"Ingrese el segundo numero"<<endl;
cin>>num2;
resta=num1-num2;
clrscr();
cout<<"EL RESULTADO DE RESTAR LOS VALORES INGRESADOS ES:
    "<<resta<<endl;
};
if (op==3)
{
int numval=0;
cout<<"Ingrese el numero de valores a ser multiplicados"<<endl;
cin>>numval;
float mult=1;
for(int i=1;i<=numval;i++)
{
float num;
cout<<"Ingrese el valor "<<i<<endl;
cin>>num;
mult=mult*num;
}
cout<<"El resultado de multiplicar los valores ingresados es: "<<mult<<endl;
getch();
};
if (op==4)
{
float num1,num2,divi;
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
cout<<"Ingrese el primer numero"<<endl;
cin>>num1;
cout<<"Ingrese el segundo numero"<<endl;
cin>>num2;
divi=num1/num2;
```



```
cout<<"EL RESULTADO DE DIVIDIR LOS VALORES INGRESADOS ES:
    "<<divi<<endl;
};
if (op==5)
{
clrscr();
float num=0;
cout<<"INGRESE EL NUMERO"<<endl;
cin>>num;
cout<<"La raiz cuadradadelnumero ingresado es: "<<sqrt(num)<<endl;
getch();
};
cout<<"SI DESEA CONTINUAR PRESIONE 0, SI DESEA SALIR PRESIONE
    1"<<endl;
cin>>op1;
clrscr();
}while (op1==0);
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};
```

EJERCICIOS PROPUESTOS

10.- Implementar un programa que permita sumar dos matrices ingresadas, la matriz resultante será el resultado de sumar los valores de la matriz 1 con su correspondiente en la matriz 2, es decir el elemento de la posición (1,1) de la matriz 1 se sumará con el elemento (1,1) de la matriz 2.

El presente programa permite sumar dos matrices ingresadas y obtener una matriz resultante/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
int a,b,c,d,e,f,g,h,i;
int j,k,l,m,n,o,p,q,r;
int s,t,u,v,a2,x,y,a3,a1;
clrscr();
gotoxy(30,5);
cout<<"SUMA DE MATRICES 3*3 "<<endl;
```

```

gotoxy(2,7);cout<<"INGRESE LA PRIMERA MATRIZ";
gotoxy(10,10);cout<<" ";scanf("%d",&a);
gotoxy(10,12);cout<<" ";scanf("%d",&b);
gotoxy(10,14);cout<<" ";scanf("%d",&c);
gotoxy(14,10);cout<<" ";scanf("%d",&d);
gotoxy(14,12);cout<<" ";scanf("%d",&e);
gotoxy(14,14);cout<<" ";scanf("%d",&f);
gotoxy(18,10);cout<<" ";scanf("%d",&g);
gotoxy(18,12);cout<<" ";scanf("%d",&h);
gotoxy(18,14);cout<<" ";scanf("%d",&i);
gotoxy(36,7);cout<<"INGRESE LA SEGUNDA MATRIZ";
gotoxy(40,10);cout<<" ";scanf("%d",&j);
gotoxy(40,12);cout<<" ";scanf("%d",&k);
gotoxy(40,14);cout<<" ";scanf("%d",&l);
gotoxy(44,10);cout<<" ";scanf("%d",&m);
gotoxy(44,12);cout<<" ";scanf("%d",&n);
gotoxy(44,14);cout<<" ";scanf("%d",&o);
gotoxy(48,10);cout<<" ";scanf("%d",&p);
gotoxy(48,12);cout<<" ";scanf("%d",&q);
gotoxy(48,14);cout<<" ";scanf("%d",&r);
s=a+j;
gotoxy(20,16);cout<<"LA MATRIZ RESULTANTE ES LA SIGUIENTE: ";
gotoxy(29,18);cout<<("%d",s);
t=b+k;
gotoxy(29,20);cout<<("%d",t);
u=c+l;
gotoxy(29,22);cout<<("%d",u);
v=d+m;
gotoxy(33,18);cout<<("%d",v);
a2=e+n;
gotoxy(33,20);cout<<("%d",a2);
x=f+o;
gotoxy(33,22);cout<<("%d",x);
y=g+p;
gotoxy(37,18);cout<<("%d",y);
a3=h+q;
gotoxy(37,20);cout<<("%d",a3);
a1=i+r;
gotoxy(37,22);cout<<("%d",a1);
gotoxy(27,24);cout<<"<<<SEINSEV>>>"<<endl;
getch();
};

```

11.- Implementar un programa que permita obtener la sumatoria y el promedio de un conjunto de números ingresados, el número de valores se pedirá que el usuario ingrese por teclado.

/*ESTE PROGRAMAPERMITE OBTENER LA SUMATORIA Y ELPROMEDIO DE
UN CONJUNTO DE NUMEROS INGRESADOS*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int numval;
    float suma=0;
    cout<<"Ingrese el numero de valores a ingresar"<<endl;
    cin>>numval;
    for (int i=1;i<=numval;i++)
    {
        int num;
        cout<<"Ingreseel valor "<<i<<endl;
        cin>>num;
        suma=suma+num;
    };
    float prom=suma/numval;
    cout<<"LA SUMATORIA DE LOS VALORES INGRESADOS ES: "<<suma<<endl;
    cout<<"EL PROMEDIO DE LOS VALORES INGRESADOS ES: "<<prom<<endl;
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;

    getch();
};
```

12.- Implementar un programa que permita obtener el área y perímetro de un círculo dado su radio.

/*ESTE PROGRAMAPERMITE OBTENER EL AREA EL PERIMETRO DE UN
CIRCULO DADO EL RADIO*/

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    clrscr();
    const pi=3.1416;
    float rad,area=0,perim=0;
    cout<<"Ingrese el radio del circulo "<<endl;
    cin>>rad;
    area=pi*(rad*rad);
    perim=2*pi*rad;
```

```
cout<<"EL AREA ES DE: "<<area<<endl;
cout<<"EL PERIMETRO ES DE: "<<perim<<endl;
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};
```

13.- Implementar un programa que permita obtener el área de un rombo ingresando los valores de la diagonal mayor y diagonal menor.

```
/*ESTE PROGRAMA PERMITE OBTENER EL AREA DE UN ROMBO*/
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    clrscr();
    float d1,d2,area=0;
    cout<<"Ingrese el valor de la diagonal mayor"<<endl;
    cin>>d1;
    cout<<"Ingrese el valor de la diagonal menor"<<endl;
    cin>>d2;
    area=d1*d2;
    cout<<"El area del rombo es: "<<area<<endl;
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

14.- Implementar un programa que permita obtener el área de un triángulo ingresando el valor de la base y altura.

```
/*ESTE PROGRAMA PERMITE OBTENER EL AREA DE UN TRIANGULO*/
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
    clrscr();
    float base,alt,area=0;
    cout<<"Ingrese el valor de la base"<<endl;
    cin>>base;
    cout<<"Ingrese el valor de la altura"<<endl;
    cin>>alt;
    area=(base*alt)/2;
    cout<<"El area del triangulo es: "<<area<<endl;
```

```
gotoxy(30,50);  
cout<<"<<<SEINSEV>>>"<<endl;  
getch();  
};
```

15.- Implementar un programa que permita obtener la fecha y hora actuales del sistema.

```
/*ESTE PROGRAMA PERMITE OBTENER LA HORA ACTUAL DEL SISTEMA*/  
#include<iostream.h>  
#include<conio.h>  
#include<stdio.h>  
#include<math.h>  
#include<time.h>  
void main()  
{  
    time_t hora;  
    time(&hora);  
    clrscr();  
    gotoxy(20,20);  
    cout<<"Hora actual: "<<ctime(&hora)<<endl;  
    gotoxy(30,50);  
    cout<<"<<<SEINSEV>>>"<<endl;  
    getch();  
};
```

16.- Implementar un programa que permita elevar un numero a una potencia.

```
/*ESTE PROGRAMA PERMITE ELEVAR UN NUMERO A UNA POTENCIA*/  
#include<iostream.h>  
#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
void main()  
{  
    clrscr();  
    int base,expo,res=1;  
    cout<<"Ingrese el valor de la base "<<endl;  
    cin>>base;  
    cout<<"Ingrese el valor del exponente "<<endl;  
    cin>>expo;  
    for (int i=1;i<=expo;i++)  
    {  
        res=res*base;  
    };  
    cout<<"El resultado es "<<res<<endl;  
    gotoxy(30,50);  
    cout<<"<<<SEINSEV>>>"<<endl;
```

```
getch();
};
```

17.- Implementar un programa que permita dibujar triángulos, rombo empleando el símbolo asterisco "*", además el usuario deberá ingresar el número de niveles para la figura que se desea mostrar.

/*EL PRESENTE PROGRAMA PERMITE DIBUJAR TRIANGULOS O UN ROMBO
INGRESANDO UN NUMERO DE NIVELES POR TECLADO*/

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int op,nn;
    do
    {
        cout<<"ELIJA UNA OPCION"<<endl<<endl;
        cout<<"1.- ***"<<"          "<<"2.- *"<<endl;;
        cout<<"  ** "<<"          "<<"  **"<<endl;
        cout<<"   *<<"          "<<"   **"<<endl<<endl<<endl<<endl;
        cout<<"3.- ***"<<"          "<<"4.- * "<<endl;
        cout<<"  **"<<"          "<<"  ** "<<endl;
        cout<<"   *<<"          "<<"   **"<<endl<<endl<<endl<<endl;
        cout<<"5.- * "<<"          "<<"6.- SALIR"<<endl;
        cout<<"  ***"<<endl;
        cout<<"   * "<<endl<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
        cin>>op;
        if (op==1)
        {
            cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;
            cin>>nn;
            int nn1=nn;
            while (nn1>=1)
            {
                for (int i=1;i<=nn1;i++)
                {
                    cout<<"*";
                };
                cout<<endl;
                nn1--;
            };
            getch();
            clrscr();
        };
        if (op==2)
```

```
{
    cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;
    cin>>nn;
    int nn1,i,c;
    c=1;
    nn1=nn;
    do
    {
        for(i=1;i<=nn-c;i++)
        {
            cout<<" ";
        }
        for (int j=i;j<=nn;j++)
        {
            cout<<"*";
        }
        cout<<endl;
        nn1--;
        c++;
    }while(nn1>=1);
    getch();
    clrscr();
};
if (op==3)
{
    cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;
    cin>>nn;
    int nn1,c;
    nn1=nn;
    c=1;
    do
    {
        for (int i=1;i<=nn1;i++)
        {
            cout<<"*";
        }
        cout<<endl;
        for(int j=1;j<=c;j++)
        {
            cout<<" ";
        }
        c++;
        nn1--;
    } while(nn1>=1);
    getch();
    clrscr();
};
```

```
if (op==4)
{
    cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;
    cin>>nn;
    int nn1,i,c;
    nn1=nn;
    c=1;
    do
    {
        for (i=1;i<=c;i++)
        {
            cout<<"*";
        }
        for (int j=i;j<=nn;j++)
        {
            cout<<" ";
        }
        cout<<endl;
        c++;
        nn1--;
    }while(nn1>=1);
    getch();
    clrscr();
};
if (op==5)
{
    cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;
    cin>>nn;
    int a,b;
    a=(nn/2);
    b=(nn-(a*2));
    if (b==0)
    {
        cout<<"EL NUMERO DE NIVELES PARA EL ROMBO DEBE SER
            INPAR"<<endl;
    }
    else
    {
        int nn1,aux,aux1,c,i,ct;
        nn1=nn;
        ct=1;
        aux=nn1/2;
        aux1=aux;
        c=1;
        do
        {
            for (i=1;i<=aux;i++)
```



```
{
    cout<<" ";
}
for (int j=i;j<=aux+c;j++)
{
    cout<<"*";
}
for (int k=j;k<=nn1;k++)
{
    cout<<" ";
}
c=c+2;
aux=aux-1;
cout<<endl;
ct=ct+1;
}while(ct<=aux1+1);
int ce,cas,aux2,l,m,ctt;
ce=1;
aux2=nn/2;
cas=nn-2;
ctt=1;
do
{
    for (l=1;l<=ce;l++)
    {
        cout<<" ";
    }
    for (m=1;m<=cas;m++)
    {
        cout<<"*";
    }
    for(int n=m+l;n<=nn;n++)
    {
        cout<<" ";
    }
    cout<<endl;
    ce++;
    cas=cas-2;
    ctt++;
}while(ctt<=aux2);
}
getch();
clrscr();
}
if ((op<0)&&(op>6))
{
    cout<<"ESA OPCION NO EXISTE"<<endl;
```

```
};
}while (op!=6);
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

18.- Implementar un programa que permita resolver una ecuación empleando la fórmula general $x = \frac{-b \pm (b^2 - 4ac)^{1/2}}{2a}$, es decir ingresando los valores de a, b, c

/*EL PRESENTE PROGRAMA PERMITE RESOLVER UNA ECUACION POR MEDIO DE LA FORMULA GENERAL*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
clrscr();
float a,b,c,x1,x2;
cout<<"INGRESE EL VALOR DE a"<<endl;
cin>>a;
cout<<"INGRESE EL VALOR DE b"<<endl;
cin>>b;
cout<<"INGRESE EL VALOR DE c"<<endl;
cin>>c;
x1=(-b+(sqrt((b*b)-(4*a*c))))/2*a;
x2=(-b-(sqrt((b*b)-(4*a*c))))/2*a;
cout<<"La solución 1 es: "<<x1<<endl;
cout<<"La solución 2 es: "<<x2<<endl;

gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

19.- Implementar un programa que permita ingresar notas y determinar cuál es su equivalente respecto a sobresaliente, muy buena, buena, regular, insuficiente, teniendo en cuenta que, el valor de sobresaliente es mayor de 18 y menor igual a 20, muy buena es mayor a 15 y menor o igual a 18, buena es mayor a 13 y menor o igual a 15, regular si es mayor a 10 y menor igual a 13, insuficiente mayor a 0 y menor o igual a 10.

/*EL PRESENTE PROGRAMA PERMITE INGRESAR NOTAS Y DECIDIR CUAL ES SU EQUIVALENCIA*/

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

```
void main ()
{
    clrscr();
    float cant,sum=0,prom=0;
    cout<<"INGRESE EL NUMERO DE VALORES A INGRESAR"<<endl;
    cin>>cant;
    for (int i=1;i<=cant;i++)
    {
        float num=0;
        do{
            cout<<"INGRESE EL VALOR "<<i<<endl;
            cin>>num;
            if ((num<0) || (num>20))
            {
                cout<<"LA NOTA INGRESADA NO ES VALIDA"<<endl;
                i--;
                num=0;
            }
        }while (num<0 &&(num>20));
        if ((num>0) & (num<=10))
        {
            cout<<"LA NOTA INGRESADA ES INSUFICIENTE"<<endl;
        }
        if ((num>10) & (num<=13))
        {
            cout<<"LA NOTA INGRESADA ES REGULAR"<<endl;
        }
        if ((num>13) & (num<=15))
        {
            cout<<"LA NOTA INGRESADA ES BUENA"<<endl;
        }

        if ((num>15) & (num<=18))
        {
            cout<<"LA NOTA INGRESADA ES MUY BUENA"<<endl;
        }
        if ((num>18) & (num<=20))
        {
            cout<<"LA NOTA INGRESADA ES SOBRESALIENTE"<<endl;
        }
        sum=sum+num;
    };
    prom=sum/cant;
    gotoxy(20,40);
    cout<<"EL PROMEDIO DE LAS NOTAS INGRESADAS ES: "<<prom<<" ";
    if ((prom>0) & (prom<=10))
    {
```

```

gotoxy(20,41);
cout<<"Y ES INSUFICIENTE"<<endl;
}
if ((prom>10) & (prom<=13))
{
gotoxy(20,41);
cout<<"Y ES REGULAR"<<endl;
}
if ((prom>13) & (prom<=15))
{
gotoxy(20,41);
cout<<"Y ES BUENA"<<endl;
}

if ((prom>15) & (prom<=18))
{
gotoxy(20,41);
cout<<"Y ES MUY BUENA"<<endl;
}
if ((prom>18) & (prom<=20))
{
gotoxy(20,41);
cout<<"Y ES SOBRESALIENTE"<<endl;
}
gotoxy(20,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};

```

20.- Implementar un programa que permita obtener la serie Fibonacci, ingresando por teclado el número de valores a obtener de la serie.

```

/*EL PRESENTE PROGRAMA PERMITE OBTENR LA SERIE DE FIBONACCI
INGRESADNDO CUANTOS VALORES DE LA SERIE SE DESEA
OBTENER*/
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
int numval,ac1=1,ac2=1,cont=0,aux=0;
cout<<"INGRESE CUANTOS VALORES DE LAS SERIE DESEA
OBTENER"<<endl;
cin>>numval;
if (numval<=0)

```

```
{
    cout<<"INGRESE UN VALOR POSITIVO"<<endl;
}
else
{
    if (numval==1)
    {
        cout<<ac1<<endl;
    }
    else
    {
        cout<<ac1<<" "<<ac2<<" ";
        while (ac1<numval-2)
        {
            aux++;
            cont=ac1+ac2;
            cout<<cont<<" ";
            ac1=ac2;
            ac2=cont;
        }
    }
}
cout<<" "<<endl;
gotoxy(20,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

21.- Implementar un programa que permita obtener la persistencia de un número ingresado, teniendo en cuenta que la persistencia de un número es el número de veces que se deben multiplicar los dígitos de un número hasta que el número queda en un solo dígito ejemplo del número 798, tiene persistencia 2; pues $7*8*9$ es 504, seguidamente $5*0*4$ es 0 por lo que el número tuvo una persistencia de 2.

```
/*el presente programa permite obtener la persistencia de un numero ingresado*/
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include <math.h>
void main()
{
    clrscr();
    int num=0,a=0,b=1,cp=0,aux=0,c=0,d=0,e=0;
    cout<<"ingrese el numero"<<endl;
    cin>>num;
```

```
if ((num<=99)&(num>9))
{
    while (num>=10)
    {
        a=num/10;
        b=num%10;
        aux=a*b;
        num=aux;
        aux=0;
        cp++;
    };
};
if ((num<=999)&(num>99))
{
    while (num>=100)
    {
        a=num/100;
        b=num%100;
        c=b/10;
        b=b%10;
        aux=a*b*c;
        num=aux;
        aux=0;
        cp++;
    };
};
if ((num<=99)&(num>9))
{
    while (num>=10)
    {
        a=num/10;
        b=num%10;
        aux=a*b;
        num=aux;
        aux=0;
        cp++;
    };
};
if ((num<=9999)&(num>999))
{
    while (num>=1000)
    {
        a=num/1000;
        b=num%1000;
        c=b/100;
        b=b%100;
        d=b/10;
```

```

e=b%10;
aux=a*c*d*e;
num=aux;
aux=0;
cp++;
};
if ((num<=999)&(num>99))
{
while (num>=100)
{
a=num/100;
b=num%100;
c=b/10;
b=b%10;
aux=a*b*c;
num=aux;
aux=0;
cp++;
};
};
if ((num<=99)&(num>9))
{
while (num>=10)
{
a=num/10;
b=num%10;
aux=a*b;
num=aux;
aux=0;
cp++;
};
};
cout<<"LA PERSISTENCIA DEL NUMERO ES: "<<cp<<endl;
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();

};

```

22.- Implementar un programa que permita obtener la factorial de un número ingresado.

```

/*EL PRESENTE PROGRAMA PERMITE OBTENER EL FACTORIAL DE UN
NUMERO INGRESADO*/
#include<iostream.h>
#include<conio.h>
#include<string.h>

```

```
void main()
{
    long int f=1,n;
    cout<<"ingrese el numero del que de desea encontra en factorial"<<endl;
    cin>>n;
    for(int i=1; i<=n; i++)
    {
        f=f*i;
    };
    cout<<"El factorial de  "<< n <<"  es  "<<f <<endl;

    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    getch();
};
```

23.- Implementar un programa que permita obtener la suma en columna de números de hasta 3 cifras.

```
/*El presnete programa permite realizar la suma en columna de numeros de hasta tres
cifras*/
#include<conio.h>
#include<iostream.h>
#include<stdio.h>
#include<math.h>
void main ()
{
    clrscr();
    int
        a,b,a1=0,a2=0,a3=0,b1=0,b2=0,b3=0,res1=0,res2=0,r1=0,r2=0,r3=0,llev1=0,llev
        2=0;
    cout<<"Ingrese el Primer Valor "<<endl;
    cin>>a;
    cout<<"Ingrese el Segundo Valor "<<endl;
    cin>>b;
    a1=a/100;
    res1=a%100;
    a2=res1/10;
    a3=res1%10;
    b1=b/100;
    res2=b%100;
    b2=res2/10;
    b3=res2%10;
    cout<<"el valor a1 es "<<a1<<endl;
    cout<<"el valor a2 es "<<a2<<endl;
    cout<<"el valor a3 es "<<a3<<endl;
    cout<<"el valor b1 es "<<b1<<endl;
```



```
cout<<"el valor b2 es "<<b2<<endl;
cout<<"el valor b3 es "<<b3<<endl;
r3=(a3+b3)%10;
llev1=(a3+b3)/10;
r2=(a2+b2+llev1)%10;
llev2=(a2+b2+llev1)/10;
r1=(a1+b1+llev2);
cout<<"EL resultado es "<<r1*100+r2*10+r3<<endl;
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

24.- Implementar un programa que permita imprimir un listado de números perfectos que se encuentren en un rango de valores dado e ingresado por teclado.

```
/*El presente programa imprime una lista de numeros perfectos*/
/*siendo perfectos aquellos cuya suma de sus divisores ecepto el numero dan el mismo numero*/
#include<math.h>
#include<conio.h>
#include<stdio.h>
#include<iostream.h>
void main()
{
clrscr();
int li,ls;
cout<<"Ingrese el rango de valores a buscar"<<endl;
cout<<"Ingrese el limite inferior "<<endl;
cin>>li;
cout<<"Ingrese el limite superior "<<endl;
cin>>ls;
for (int i=li;i<=ls;i++)
{
int sumdiv=0;
for (int j=1;j<i;j++)
{
if (i%j==0)
{
sumdiv=sumdiv+j;
}
}
if (sumdiv==i)
{
cout<<i<<" Es numero perfecto"<<endl;
}
}
}
```

```
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};
```

25.- Implementar un programa que permita invertir un numero de 8cifras es decir si el numero ingresado es 12345678 obtener como resultado el numero 87654321.

```
/*El presente programa permite invertir un numero de 8 cifras ingresado*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
    clrscr();
    long inv,num,prim,seg,ter,cuar,quin,sex,sep,oct,res1,res2,res3,res4,res5,res6;
    cout<<"Ingrese el numero que desea invertir, asegurece que tenga 8 cifras"<<endl;
    cin>>num;
    oct=num/10000000;
    res1=num%10000000;
    sep=res1/1000000;
    res2=res1%1000000;
    sex=res2/100000;
    res3=res2%100000;
    quin=res3/10000;
    res4=res3%10000;
    cuar=res4/1000;
    res5=res4%1000;
    ter=res5/100;
    res6=res5%100;
    seg=res6/10;
    prim=res6%10;

    inv=(prim*10000000)+(seg*1000000)+(ter*100000)+(cuar*10000)+(quin*1000
    )+(sex*100)+(sep*10)+(oct);
    cout<<"El numero invertido es "<<inv<<endl;
    gotoxy(30,50);
    cout<<"<<<<SEINSEV>>>>"<<endl;
    getch();
};
```

26.- Implementarun programa que eprmita obtener lo n primeros números de la siguiente serie $2/3+4/6-6/9+...n^*/$; teniendo en cuenta que el valor de n ingresa el usuario por teclado.

/*El presente programa permite imprimir los n primeros terminos de la siguiente serie: - $2/3+4/6-6/9+...n^*/$

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
    clrscr();
    int cont=1,nval,nume=0,den=0;
    cout<<"Ingrese el numero de valores que desea obtener de la serie "<<endl;
    cin>>nval;
    while (cont<nval)
    {
        nume=nume+2;
        den=den+3;
        cont++;
        if (cont%2==0)
        {
            cout<<"+"<<nume<<"/"<<den<<" ";
        }else
        {
            cout<<"-<<nume<<"/"<<den<<" ";
        }
    }
    gotoxy(30,50);
    cout<<"SEINSEV"<<endl;
    getch();
};
```

27.- Implementar un programa que permita obtener los n primeros elementos de la siguiente serie $2/3+4/6-6/9+...n*/$; teniendo en cuenta que el valor de n se ingresará por teclado.

/*El presente programa permite imprimir los n primeros terminos de la siguiente serie: -

$2/3+4/6-6/9+...n*/$

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
    clrscr();
    long cont=0,nval,nume=0,den=0,nus=1,des=1;
    cout<<"Ingrese el numero de valores que desea obtener de la serie "<<endl;
    cin>>nval;
    while (cont<nval)
    {
        nus=1;
        des=1;
        nume=nume+2;
```

```

den=den+1;
for (int i=1;i<=nume;i++)
{
    nus=nus*i;
}
for (int j=1;j<=den;j++)
{
    des=des*den;
}
if (cont%2==0)
{
    cout<<" "<<nus<<"/"<<des<<" ";
}
else
{
    cout<<"-"<<nus<<"/"<<des<<" ";
}
cont++;
}
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};

```

28.- Implementar un programa que permita obtener el dígito verificador de la cédula de identidad ingresando los 9 primeros números.

```

/*ESTE PROGRAMA PERMITE OBTENER EL DIGITO VERIFICADOR*/
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    clrscr();
    int ced[9];
    int pospa,pim,pim1,pim2,pim3,pim4,pim5,sum,ds,dverif;
    for (int i=0;i<=8;i++)
    {
        cout<<"Ingrese el digito "<<i<<endl;
        cin>>ced[i];
    }
    pospa=ced[1]+ced[3]+ced[5]+ced[7];
    pim1=ced[0]*2;
    if (pim1>9)
    {
        pim1=pim1-9;
    }
}

```

```
pim2=ced[2]*2;
if (pim2>9)
{
    pim2=pim2-9;
}
pim3=ced[4]*2;
if (pim3>9)
{
    pim3=pim3-9;
}
pim4=ced[6]*2;
if (pim4>9)
{
    pim4=pim4-9;
}
pim5=ced[8]*2;
if (pim5>9)
{
    pim5=pim5-9;
}
pim=pim1+pim2+pim3+pim4+pim5;
sum=pim+pospa;
ds=((sum/10)*10)+10;
dverif=sum-ds;
if (dverif==10)
{
    dverif=0;
}
cout<<"El digito verificador es "<<dverif<<endl;
gotoxy(30,50);
cout<<"SEINSEV"<<endl;
getch();
};
```

29.- Implementar un programa que permita ingresar 2 vectores e intercambiar sus elementos.

/*El presente programa permite ingresar 2 vectores e intercambiar sus valores*/

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

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

```
#include<stdio.h>
```

```
#include<math.h>
```

```
void main()
```

```
{
```

```
    clrscr();
```

```
    int n,aux;
```

```
    cout<<"Ingrese el numero de elementos de los vectores"<<endl;
```

```
cin>>n;
int vec1[30],vec2[30];
for (int i=0;i<=n;i++)
{
    cout<<"Ingrese el elemento "<<i<<endl;
    cin>>vec1[i];
}
for (int j=0;j<=n;j++)
{
    cout<<"Ingrese el elemento "<<j<<endl;
    cin>>vec2[j];
}
clrscr();
cout<<"Los elementos del primer vector son> "<<endl;
for (int k=0;k<=n;k++)
{
    cout<<"Elemento "<<k<<" "<<vec1[k]<<endl;
}
cout<<endl<<"Los elementos del segundo vector son> "<<endl;
for (int l=0;l<=n;l++)
{
    cout<<"Elemento "<<l<<" "<<vec2[l]<<endl;
}
for (int m=0;m<=n;m++)
{
    aux=vec1[m];
    vec1[m]=vec2[m];
    vec2[m]=aux;
}
cout<<"Los elementos del primer vector son> "<<endl;
for (int o=0;o<=n;o++)
{
    cout<<"Elemento "<<o<<" "<<vec1[o]<<endl;
}
cout<<endl<<"Los elementos del segundo vector son> "<<endl;
for (int p=0;p<=n;p++)
{
    cout<<"Elemento "<<p<<" "<<vec2[p]<<endl;
}
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

30.- Implementar un programa que permita ordenar descendientemente los elementos de un vector, empleando el método de la burbuja.

/*El presente programa permite ordenar un vector descendientemente por el metodo de la burbuja*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int n,aux;
    cout<<"Ingrese el numero de elementos de los vectores"<<endl;
    cin>>n;
    int vec1[30],vec2[30];
    for (int i=0;i<=n;i++)
    {
        cout<<"Ingrese el elemento "<<i<<endl;
        cin>>vec1[i];
    }
    for (int z=0;z<=n;z++)
    {
        for (int j=0;j<=n;j++)
        {
            if (vec1[z]>vec1[j])
            {
                aux=vec1[z];
                vec1[z]=vec1[j];
                vec1[j]=aux;
            }
        }
    }
    clrscr();
    cout<<"Los elementos del vector ordenados ascendentemente son> "<<endl;
    for (int o=0;o<=n;o++)
    {
        cout<<"Elemento "<<o<<" "<<vec1[o]<<endl;
    }
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

31.- Implementar un programa que permita ordenar ascendentemente los elementos de un vector, empleando el método de la burbuja.

/*El presente programa permite ordenar un vector ascendentemente por el metodo de la burbuja*/

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

```
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int n,aux;
    cout<<"Ingrese el numero de elementos de los vectores"<<endl;
    cin>>n;
    int vec1[30],vec2[30];
    for (int i=0;i<=n;i++)
    {
        cout<<"Ingrese el elemento "<<i<<endl;
        cin>>vec1[i];
    }
    for (int z=0;z<=n;z++)
    {
        for (int j=0;j<=n;j++)
        {
            if (vec1[z]<vec1[j])
            {
                aux=vec1[z];
                vec1[z]=vec1[j];
                vec1[j]=aux;
            }
        }
    }
    clrscr();
    cout<<"Los elementos del vector ordenados ascendentemente son "<<endl;
    for (int o=0;o<=n;o++)
    {
        cout<<"Elemento "<<o<<" "<<vec1[o]<<endl;
    }
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

32.- Implementar u programa que permita ingresar un vector y posteriormente buscar un elemento del mismo, en el caso de existir el elemento a buscar mostrar la posición en la que se encuentra el elemento.

```
/*El presente programa permite buscar un elemento en un vecto ingresado y da su
posicion en el vector*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
```



```
#include<math.h>
void main()
{
    clrscr();
    int n,nbus,ban;
    cout<<"Ingrese el numero de elementos de los vectores"<<endl;
    cin>>n;
    int vec1[30],vec2[30];
    for (int i=0;i<=n;i++)
    {
        cout<<"Ingrese el elemento "<<i<<endl;
        cin>>vec1[i];
    }
    cout<<"Ingrese el elemento a buscar "<<endl;
    cin>>nbus;
    for (int z=0;z<=n;z++)
    {
        if (nbus==vec1[z])
        {
            ban=1;
            cout<<"El valor "<<nbus<<" existe en el vector y esta en la posicion: "<<z<<endl;
        }
    }
    if (ban==0)
    {
        cout<<"El valor ingresado no existe en el vector"<<endl;
    }
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
    getch();
};
```

33.- Implementar un programa que permita insertar un elemento dentro de un vector y ordenarlo ascendentemente.

```
/*El presente programa permite insertar un elemento dentro de un vector y ordenarlo
ascendentemente*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    int n=0,aux=0,ins=0,vec[30];
    cout<<"Ingrese el numero de elementos del vector"<<endl;
    cin>>n;
```

```

cout<<"Ingrese los elementos del vector"<<endl;
/*ingreso de los elementos del vector*/
for (int a=0; a<=n;a++)
{
    cout<<"Ingrese el elemento "<<a<<endl;
    cin>>vec[a];
};
cout<<"Ingrese el elemento a insertar en el vector"<<endl;
cin>>ins;
int m=n+1;
vec[n+1]=ins;
/*ordenacion ascendente del vector*/
for (int i=0;i<=m;i++)
{
    for (int j=0;j<=m-1;j++)
    {
        if (vec[i]<vec[j])
        {
            aux=vec[i];
            vec[i]=vec[j];
            vec[j]=aux;
        }
    }
}
clrscr();
/*escritura del vector ordenado e insertado nel valor*/
cout<<"LOS ELEMENTOS DEL NUEVO VECTOR SON: "<<endl;
for(int o=0;o<=n+1;o++)
{
    cout<<"El elemento " <<o<<" del vector es: "<<vec[o]<<endl;
}
getch();
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};

```

34.- Implemenatr un programa que permita mostrar el funcionamiento de los principales operadores de cadenas de caracteres que se emplea en C++.

```

/*El presente programa presenta un menu para interactuar con los principales
operadores para cadenas de caracteres*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()

```

```
{
int op=0;
cout<<"ELIJA UNA OPCION "<<endl;
while (op==0)
{
clrscr();
cout<<"1.- Obtener la longitud de una cadena strlen"<<endl;
cout<<"2.- Unir dos cadenas de caracteres"<<endl;
cout<<"3.- Copiar una cadena a otra cadena"<<endl;
cout<<"4.- Comparar dos cadenas de caracteres"<<endl;
cin>>op;
}
if (op==1)
{
clrscr();
char cad1[40];
cout<<"INGRESE LA CADENA PARA OBTENER LA LONGITUD"<<endl;
gets(cad1);
cout<<" La longitud de la Cadena ingresada es : "<<strlen (cad1)<<endl;
getch();
}
if (op==2)
{
clrscr();
char cad1[40], cad2[40];
cout<<"INGRESE LA PRIMERA CADENA"<<endl;
gets(cad1);
cout<<"INGRESE LA SEGUNDA CADENA"<<endl;
gets(cad2);
cout<<" La nueva cadena es : "<<strcat (cad1, cad2)<<endl;
getch();
}
if (op==3)
{
clrscr();
char cad1[40], cad2[40];
cout<<"INGRESE LA PRIMERA CADENA"<<endl;
gets(cad1);
cout<<"INGRESE LA SEGUNDA CADENA"<<endl;
gets(cad2);
clrscr();
cout<<" La cadena1 es : "<<cad1<<endl;
cout<<" La cadena2 es : "<<cad2<<endl;
strcpy (cad1,cad2);
}
```

```
cout<<"AL COPIAR LA CADENA 2 A LA CADENA 1 EL RESULTADO ES:
    "<<endl;
cout<<" La cadena1 es : "<<cad1<<endl;
cout<<" La cadena2 es : "<<cad2<<endl;
getch();

}
if (op==4)
{
    clrscr();
    char cad1[40], cad2[40];
    cout<<"INGRESE LA PRIMERA CADENA"<<endl;
    gets(cad1);
    cout<<"INGRESE LA SEGUNDA CADENA"<<endl;
    gets(cad2);
    cout<<" El resultado de aplicar el operador strcmp es : "<<strcmp (cad1,cad2)<<endl;
    getch();
}
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

35.- Implemenatr un programa que permita determinar si una cadena ingresada en palindrome, teneindo en cuenta que una palabra es palindrome cuando se lee igual, de izquierda a derecha que de derecha a izquierda ejemplo: ana.

```
/*PROGRAMA PARA DETERMINAR SI UNA PALABRA ES PALINDROME*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
    clrscr();
    char cad1[40];
    int a,b,cont=0;
    cout<<"Ingrese la Cadena"<<endl;
    gets(cad1);
    a=(strlen(cad1))-1;
    b=a;
    for (int i=0;i<(a/2);i++)
    {
        if (cad1[i]==cad1[b])
        {
            cout<<"La Posicion "<<i<<"Coincide con la posicion "<<b<<endl;
            b--;
        }
    }
}
```

```

    cont++;
}
else
{
    cout<<"La Posicion "<<i<<"No Coincide con la posicion "<<b<<endl;
    b--;
}
}
if (cont==(a/2))
{
    cout<<"Por las Coincidencias anteriores La Palabra Es Palindrome"<<endl;
}
getch();
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};

```

36.- Implementar un programa que permita determinar si un número es capicúa o no, teniendo en cuenta que un número capicúa es aquel que se lee igual de izquierda a derecha que de derecha a izquierda ejemplo: 131

/*El presente programa permite determinar si un número ingresado no mayor de 99999 es Capicua o no*/

```

#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
    clrscr();
    char op='s';
    while ((op=='S')||(op=='s'))
    {
        long num,num1,num2,num3,a,b,c,d,e;
        cout<<"Ingrese el Numero"<<endl;
        cin>>num;
        if (num<10)
        {
            cout<<"El Numero es de una sola cifra"<<endl;
        };
        if ((num>10) && (num<100))
        {
            a=num/10;
            b=num-((num/10)*10);
            if (a==b)
            {
                cout<<"El Numero es Capicua"<<endl;
            }
        }
    }
}

```

```
}  
else  
{  
    cout<<"El Numero No es Capicua"<<endl;  
}  
};  
if ((num>99) && (num<999))  
{  
    a=num/100;  
    num1=num-(a*100);  
    b=num1/10;  
    c=num1-(b*10);  
    if (a==c)  
    {  
        cout<<"El Numero es Capicua"<<endl;  
    }  
    else  
    {  
        cout<<"El Numero No es Capicua"<<endl;  
    }  
}  
if ((num>999) & (num<9999))  
{  
    a=num/1000;  
    num1=num-(a*1000);  
    b=num1/100;  
    num2=num1-(b*100);  
    c=num2/10;  
    d=num2-(c*10);  
    if ((a==d)&&(b==c))  
    {  
        cout<<"El Numero es Capicua"<<endl;  
    }  
    else  
    {  
        cout<<"El Numero No es Capicua"<<endl;  
    }  
};  
if ((num>9999) && (num<99999))  
{  
    a=num/10000;  
    num1=num-(a*10000);  
    b=num1/1000;  
    num2=num1-(b*1000);  
    c=num2/100;  
    num3=num2-(c*100);  
    d=num3/10;
```

```
e=num3-(d*10);
if ((a==e)&&(b==d))
{
    cout<<"El Numero es Capicua"<<endl;
}
else
{
    cout<<"El Numero No es Capicua"<<endl;
}
}
cout<<"Desea continuar S/N"<<endl;
cin>>op;
};
getch();
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

37.- Impleemnatr un programa que permita determinar el signo de una persona, ingresando la fecha de nacimeinto.

/*ESTE PROGRAMA PERMITE DETERMINAR EL SIGNO DEL ZODIACO DE UNA PERSONA DADA SU FECHA DE NACIMIENTO*/

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char op='S';
    while ((op=='S')||(op=='s'))
    {
        int dia,mes;
        cout<<"Ingrese la fecha de nacimiento DIA"<<endl;
        cin>>dia;
        cout<<"Ingrese el mes de nacimiento en numeros"<<endl;
        cin>>mes;
        if (((mes==3)&&(dia>=21))||((mes==4)&&(dia<=20)))
        {
            cout<<"La fecha ingresada corresponde a: ARIES"<<endl;
        }
        if (((mes==4)&&(dia>=21))||((mes==5)&&(dia<=21)))
        {
            cout<<"La fecha ingresada corresponde a: TAURO"<<endl;
        }
    }
```

```

if (((mes==5)&&(dia>=22))||((mes==6)&&(dia<=21)))
{
    cout<<"La fecha ingresada corresponde a: GEMINIS"<<endl;
}
if (((mes==6)&&(dia>=22))||((mes==7)&&(dia<=23)))
{
    cout<<"La fecha ingresada corresponde a: CANCER"<<endl;
}
if (((mes==7)&&(dia>=24))||((mes==8)&&(dia<=23)))
{
    cout<<"La fecha ingresada corresponde a: LEO"<<endl;
}
if (((mes==8)&&(dia>=24))||((mes==9)&&(dia<=23)))
{
    cout<<"La fecha ingresada corresponde a: VIRGO"<<endl;
}
if (((mes==9)&&(dia>=24))||((mes==10)&&(dia<=23)))
{
    cout<<"La fecha ingresada corresponde a: LIBRA"<<endl;
}
if (((mes==10)&&(dia>=24))||((mes==11)&&(dia<=22)))
{
    cout<<"La fecha ingresada corresponde a: ESCORPION"<<endl;
}
if (((mes==11)&&(dia>=23))||((mes==12)&&(dia<=21)))
{
    cout<<"La fecha ingresada corresponde a: SAGITARIO"<<endl;
}
if (((mes==12)&&(dia>=22))||((mes==1)&&(dia<=20)))
{
    cout<<"La fecha ingresada corresponde a: CAPRICORNIO"<<endl;
}
if (((mes==1)&&(dia>=21))||((mes==2)&&(dia<=19)))
{
    cout<<"La fecha ingresada corresponde a: ACUARIO"<<endl;
}
if (((mes==2)&&(dia>=20))||((mes==3)&&(dia<=20)))
{
    cout<<"La fecha ingresada corresponde a: PICIS"<<endl;
}
cout<<"Desea continuar S/N"<<endl;
cin>>op;
};
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};

```


38.- Implementar un programa que permita ingresar un valor de una determinada unidad de medida y posteriormente obtener ese valor convertido a sus múltiplos y submúltiplos es decir si ingreso 1000m deberá aparecer en los múltiplos 1km, 10 Hm, 100Dm.

/*EL PRESENTE PROGRAMA PERMITE CONVERTIR UNIDADES A SUS MULTIPLOS Y SUBMULTIPLOS*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
    clrscr();
    char op='s';
    int op2=0,val;
    while ((op=='s')||(op=='S'))
    {
        longint
        valkilom=0,valhecto=0,valdecam=0,valmetro=0,valdecim=0,valcenti=0,valmili
        m=0;
        cout<<"Seleccione una de las unidades en las que ingresara el valor a
        Convertir"<<endl;
        cout<<"Opcion 1.-Kilometros"<<endl;
        cout<<"Opcion 2.-Hectometros"<<endl;
        cout<<"Opcion 3.-Decametros"<<endl;
        cout<<"Opcion 4.-metros"<<endl;
        cout<<"Opcion 5.-decimetros"<<endl;
        cout<<"Opcion 6.-centimetros"<<endl;
        cout<<"Opcion 7.-milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
        cin>>op2;
        if (op2==1)
        {
            clrscr();
            val=0;
            cout<<"Ingrese el valor a convertir"<<endl;
            cin>>val;
            valhecto=val*10;
            valdecam=val*100;
            valmetro=(val*1000);
            valdecim=(val*1000)*10;
            valcenti=(val*1000)*100;
            valmilim=(val*1000)*1000;
            cout<<val <<"Kilometros es el equivalente a: "<<valhecto<<"
            Hectometros"<<endl;
```

```

        cout<<val <<"Kilometros es el equivalente a: "<<valdecam<<"
Decametros"<<endl;
        cout<<val <<"Kilometros es el equivalente a: "<<valmetro<<"
metros"<<endl;
        cout<<val <<"Kilometros es el equivalente a: "<<valdecim<<"
decimetros"<<endl;
        cout<<val <<"Kilometros es el equivalente a: "<<valcenti<<"
centimetros"<<endl;
        cout<<val <<"Kilometros es el equivalente a: "<<valmilim<<"
milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }
    if (op2==2)
    {
        clrscr();
        val=0;
        cout<<"Ingrese el valor a convertir"<<endl;
        cin>>val;
        valkilom=val/10;
        valdecam=(val*100)/10;
        valmetro=(val*100);
        valdecim=(val*100)*10;
        valcenti=(val*100)*100;
        valmilim=(val*100)*1000;
        cout<<val <<"Hectometros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
        cout<<val <<"Hectometros es el equivalente a: "<<valdecam<<"
Decametros"<<endl;
        cout<<val <<"Hectometros es el equivalente a: "<<valmetro<<"
metros"<<endl;
        cout<<val <<"Hectometros es el equivalente a: "<<valdecim<<"
edcmetros"<<endl;
        cout<<val <<"Hectometros es el equivalente a: "<<valcenti<<"
centimetros"<<endl;
        cout<<val <<"Hectometros es el equivalente a: "<<valmilim<<"
milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }
    if (op2==3)
    {
        clrscr();
        val=0;
        cout<<"Ingrese el valor a convertir"<<endl;
        cin>>val;
        valkilom=(val*10)/1000;

```

```

        valhecto=(val*10)/100;
        valmetro=(val*10);
        valdecim=(val*10)*10;
        valcenti=(val*10)*100;
        valmilim=(val*10)*1000;
        cout<<val <<"Decametros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
        cout<<val <<"Decametros es el equivalente a: "<<valhecto<<"
Hectometros"<<endl;
        cout<<val <<"Decametros es el equivalente a: "<<valmetro<<"
metros"<<endl;
        cout<<val <<"Decametros es el equivalente a: "<<valdecim<<"
edcimetros"<<endl;
        cout<<val <<"Decametros es el equivalente a: "<<valcenti<<"
centimetros"<<endl;
        cout<<val <<"Decametros es el equivalente a: "<<valmilim<<"
milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }
    if (op2==4)
    {
        clrscr();
        val=0;
        cout<<"Ingrese el valor a convertir"<<endl;
        cin>>val;
        valkilom=val/1000;
        valhecto=val/100;
        valdecam=val/10;
        valdecim=val*10;
        valcenti=val*100;
        valmilim=val*1000;
        cout<<val <<"metros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
        cout<<val <<"metros es el equivalente a: "<<valhecto<<"
Hectometros"<<endl;
        cout<<val <<"metros es el equivalente a: "<<valdecam<<"
Decametros"<<endl;
        cout<<val <<"metros es el equivalente a: "<<val*10<<"
decimetros"<<endl;
        cout<<val <<"metros es el equivalente a: "<<val*100<<"
centimetros"<<endl;
        cout<<val <<"metros es el equivalente a: "<<val*1000<<"
milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }

```

```

if (op2==5)
{
    clrscr();
    val=0;
    cout<<"Ingrese el valor a convertir"<<endl;
    cin>>val;
    valkilom=(val*0,1)/1000;
    valhecto=(val*0,1)/100;
    valdecam=(val*0,1)/10;
    valmetro=(val*0,1);
    valcenti=(val*0,1)*100;
    valmilim=(val*0,1)*1000;
    cout<<val <<"decimetros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
    cout<<val <<"decimetros es el equivalente a: "<<valhecto<<"
Hectometros"<<endl;
    cout<<val <<"decimetros es el equivalente a: "<<valdecam<<"
Decametros"<<endl;
    cout<<val <<"decimetros es el equivalente a: "<<valmetro<<"
metros"<<endl;
    cout<<val <<"decimetros es el equivalente a: "<<valcenti<<"
centimetros"<<endl;
    cout<<val <<"decimetros es el equivalente a: "<<valmilim<<"
milimetros"<<endl;
    gotoxy(30,50);
    cout<<"<<<SEINSEV>>>"<<endl;
}
if (op2==6)
{
    clrscr();
    val=0;
    cout<<"Ingrese el valor a convertir"<<endl;
    cin>>val;
    valkilom=val/(100*1000);
    valhecto=val/(100*100);
    valdecam=val/(100*10);
    valmetro=(val/100);
    valdecim=val*10;
    valmilim=val*100;
    cout<<val <<"centimetros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
    cout<<val <<"centimetros es el equivalente a: "<<valhecto<<"
Hectometros"<<endl;
    cout<<val <<"centimetros es el equivalente a: "<<valdecam<<"
Decametros"<<endl;
    cout<<val <<"centimetros es el equivalente a: "<<valmetro<<"
metros"<<endl;

```

```

        cout<<val <<"centimetros es el equivalente a: "<<valdecim<<"
decimetros"<<endl;
        cout<<val <<"centimetros es el equivalente a: "<<valmilim<<"
milimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }
    if (op2==7)
    {
        clrscr();
        val=0;
        cout<<"Ingrese el valor a convertir"<<endl;
        cin>>val;
        valkilom=(val/1000)/1000;
        valhecto=(val/1000)/100;
        valdecam=(val/1000)/10;
        valmetro=(val/1000);
        valdecim=(val/1000)*10;
        valcenti=(val/1000)*100;
        cout<<val <<"milimetros es el equivalente a: "<<valkilom<<"
Kilometros"<<endl;
        cout<<val <<"milimetros es el equivalente a: "<<valhecto<<"
Hectometros"<<endl;
        cout<<val <<"milimetros es el equivalente a: "<<valmetro<<"
Decametros"<<endl;
        cout<<val <<"milimetros es el equivalente a: "<<valdecim<<"
metros"<<endl;
        cout<<val <<"milimetros es el equivalente a: "<<valcenti<<"
decimetros"<<endl;
        cout<<val <<"milimetros es el equivalente a: "<<valmilim<<"
centimetros"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
        gotoxy(30,50);
        cout<<"<<<SEINSEV>>>"<<endl;
    }
    cout<<"Desea continuar S/N"<<endl;
    cin>>op;
}
getch();
};

```

MANUAL DE PROGRAMACIÓN ORIENTADA A OBJETOS

ESTRUCTURA DE UN PROGRAMA EMPLEANDO POO EN C++

Un programa básico escrito empleando el lenguaje de programación C++ consta de las siguientes partes:

1.- Encabezado

Se lo emplea para incluir una descripción de lo que realiza el código implementado a continuación, la descripción va entre los signos /* “Slash y asterisco” de igual manera para finalizar el comentario se emplea los mismos símbolos en orden diferente */ “asterisco y slash”.

2.- Declaración de Librerías

En esta parte se incluye todas las librerías que se empleará para que el programa se ejecute correctamente, es importante anotar que cada una de las librerías tiene un uso y finalidad, cada librería brinda un conjunto de funcionalidades que facilitan el desarrollo de código es así que entre las librerías básicas y empleadas en este manual se encuentran:

- Iostream.h
- Stdio.h
- Conio.h
- String.h

La declaración de las librerías se la realiza empleando la siguiente estructura:

Símbolo numeral + palabra reservada “include” + símbolo menor que “<” + nombre de la librería + símbolo mayor que “>”

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

3.- Codificación

En esta área se encuentra el desarrollo del código, declaración, definición de clases, métodos, objetos y demás código que forma parte de un programa empleando POO.

4.- Programa Principal

En esta porción de código se encuentra entre las etiquetas “void main ()” + “{“+ instrucciones+”}”

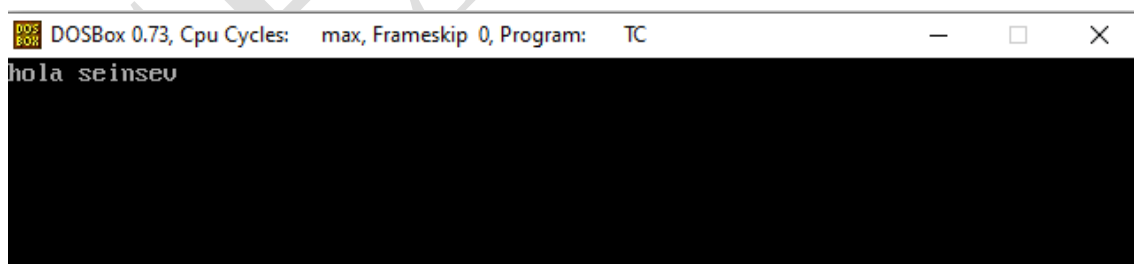
Esos son los 4 elementos que forman parte de un programa en C++.

PRIMER PROGRAMA EN C++.

Escribir el código de un programa que permita mostrar en la pantalla “HOLA SEINSEV”.

```
#include <iostream.h>
#include<conio.h>
Void main()
{
    cout<<”HOLA SEINSEV”<<endl;
    getch();
};
```

Al ejecutar el código anterior se muestra la salida en pantalla sería la que se muestra en la siguiente imagen:

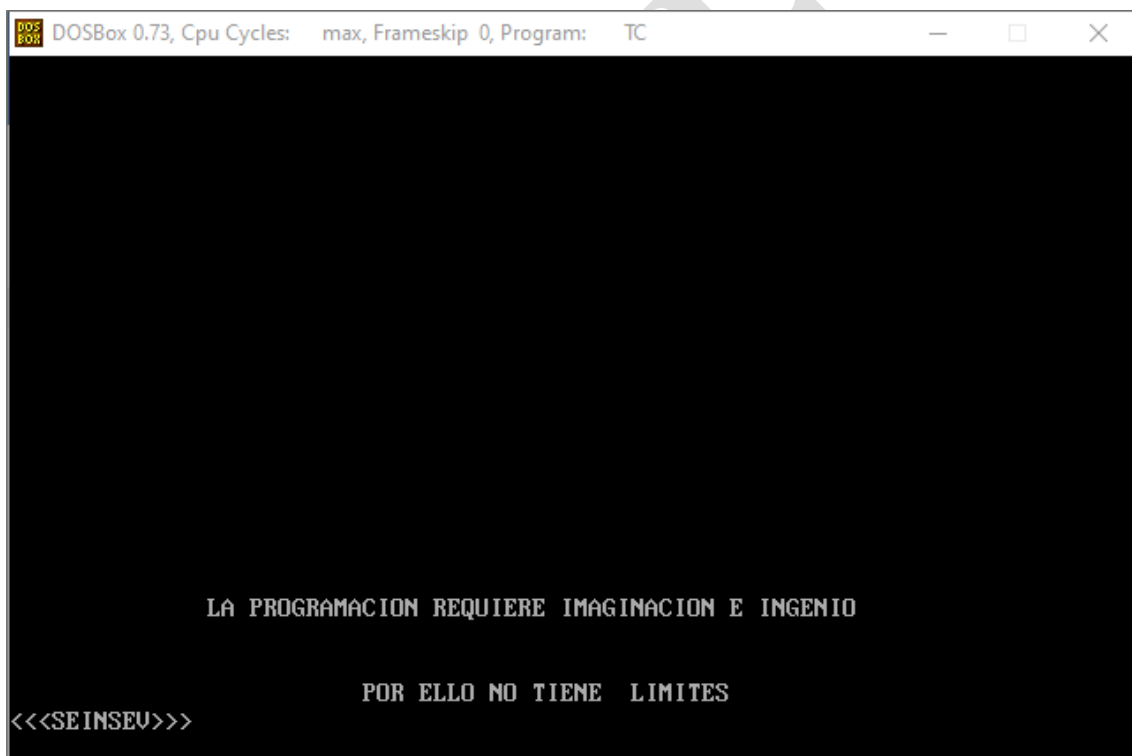


Escribir el código de un programa que permita mostrar en pantalla:
“LA PROGRAMACIÓN REQUIERE IMAGINACIÓN E INGENIO”
“POR ELLO NO TIENE LÍMITES”
“SEINSEV”

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

```
#include<stdio.h>
void main()
{
clrscr();
gotoxy(15,20);
cout<<"LA PROGRAMACION REQUIERE IMAGINACION E INGENIO"<<endl;
gotoxy(25,23);
cout<<" POR ELLO NO TIENE LIMITES"<<endl;
gotoxy(30,50);
cout<<"<<<SEINSEV>>>"<<endl;
getch();
};
```

Al ejecutar el código anterior se muestra la salida en pantalla, se puede observar en la siguiente imagen:



La programación orientada a objetos se basa en los principios de Encapsulamiento, Herencia, Polimorfismo, y emplea los conceptos de:

1. CLASE.
2. ATRIBUTO.
3. MÉTODO.
4. OBJETO.
5. ENCAPSULAMIENTO.

6. HERENCIA.
 - HERENCIA SIMPLE.
 - HERENCIA MÚLTIPLE.
7. POLIMORFISMO.

1.- CLASE.

Tipo de dato definido por el usuario en la que se determina las características y comportamiento de los objetos ejemplo:

```
Class curso
{
Char paralelo[50];
Int numerotestudiantes;
};
```

2.-ATRIBUTOS.

Conjunto de elementos con su tipo que determinan las características de los objetos ejemplo:

```
Int anio;
Flota precio;
Char nombre [25];
```

3.- MÉTODOS.

Son los que determinan el comportamiento de los objetos ante un determinado mensaje, los métodos también pueden enviar mensajes a otros métodos solicitando información o una determinada acción ejemplos:

Método leer un número.

```
float leer ()
{
float num;
Cout<<" INGRESE EL NUMERO" <<endl;
Cin>>num;
Return(num);
};
```

Método para sumar dos números.

```
float suma (float a, float b)
{
```

```
a=leer()  
b=leer();  
c=a+b;  
return(c);  
};
```

4.-OBJETO.

Es una variable del tipo definido por una clase, es decir es una entidad que tiene atributos y características particulares ejemplo:

ohombre; olapiz; omochila;

Con estas cuatro definiciones “Clase, Atributo, Método, Objeto” realizar un programa orientado a objetos que permita realizar las cuatro operaciones fundamentales dado dos números.

```
#include<iostream.h>  
#include<conio.h>  
class oper  
{  
public:  
float a,b;  
void menu();  
void leer();  
float suma(float a, float b);  
float resta(float a, float b);  
float divi(float a, float b);  
float multi(float a, float b);  
void mostrar(float a, float b);  
};  
void oper::menu()  
{  
int op=0;  
do  
{  
cout<<"ELIJA UNA OPCION"<<endl;  
cout<<"1.- SUMAR DOS NUMEROS"<<endl;  
cout<<"2.- RESTAR DOS NUMEROS"<<endl;  
cout<<"3.- MULTIPLICAR DOS NUMEROS"<<endl;  
cout<<"4.- DIVIDIR DOS NUMEROS"<<endl;  
cout<<"5.- SALIR"<<endl;  
cin>>op;  
if (op==1)  
{
```

```
leer();
cout<<"EL RESULTADO ES: "<<suma(a,b)<<endl;
getch();
clrscr();
};
if (op==2)
{
leer();
cout<<"EL RESULTADO ES: "<<resta(a,b)<<endl;
getch();
clrscr();
};
if (op==3)
{
leer();
cout<<"EL RESULTADO ES: "<<multi(a,b)<<endl;
getch();
clrscr();
};
if (op==4)
{
leer();
cout<<"EL RESULTADO ES: "<<divi(a,b)<<endl;
getch();
clrscr();
};
}while (op!=5);
};
void oper::leer()
{
cout<<"INGRESE EL PRIMER NUMERO"<<endl;
cin>>a;
cout<<"INGRESE EL SEGUNDO NUMERO"<<endl;
cin>>b;
};
float oper::suma(float a, float b)
{
return(a+b);
};
float oper::resta(float a, float b)
{
return(a-b);
};
float oper::divi(float a, float b)
{
return(a/b);
};
```

```
float oper::multi(float a, float b)
{
    return(a*b);
};
void main()
{
    oper ocalc;
    ocalc.menu();
};
```

REALIZAR UN PROGRAMA QUE PERMITA VISUALIZAR LAS SIGUIENTES FIGURAS INGRESANDO EL NUMERO DE NIVELES UTILIZANDO ASTERISCOS.

FIGURAS A VIZUALIZAR:

1.- *	2.- *	3.- **	4.- **	5.- *
**	**	*	*	***
				*

/* programa básico dibuja los diferentes triángulos y el rombo dado tamaño*/

```
#include<iostream.h>
#include<conio.h>
class graf
{
public:
    int nn;
    void menu();
    void leer();
    void triaa();
    void triab();
    void triac();
    void triad();
    void rombo();
};
void graf::menu()
{
    int op;
    do
    {
        cout<<"ELIJA UNA OPCION"<<endl<<endl;
        cout<<"1.- ***"<<"2.- *"<<endl;
        cout<<"  **"<<"  **"<<endl;
        cout<<"  *"<<"  ***"<<endl<<endl<<endl<<endl;
        cout<<"3.- ***"<<"4.- * "<<endl;
        cout<<"  **"<<"  **"<<endl;
        cout<<"  *"<<"  ***"<<endl<<endl<<endl<<endl;
        cout<<"5.- * "<<"6.- SALIR"<<endl;
        cout<<"  ***"<<endl;
```

```
cout<<"    * "<<endl<<endl;
cin>>op;
if (op==1)
{
    leer();
    triaa();
    getch();
    clrscr();
};
if (op==2)
{
    leer();
    triab();
    getch();
    clrscr();
};
if (op==3)
{
    leer();
    triac();
    getch();
    clrscr();
};
if (op==4)
{
    leer();
    triad();
    getch();
    clrscr();
};
if (op==5)
{
    leer();
    int a,b;
    a=(nn/2);
    b=(nn-(a*2));
    if (b==0)
    {
        cout<<"EL NUMERO DE NIVELES PARA EL ROMBO DEBE SER
INPAR"<<endl;
    }
    else
    {
        rombo();
    }
    getch();
    clrscr();
}
```

```
}  
if ((op<0)&&(op>6))  
{  
    cout<<"ESA OPCION NO EXISTE"<<endl;  
};  
}while (op!=6);  
};  
void graf::leer()  
{  
    cout<<"INGRESE EL NUMERO DE NIVELES DE LA FIGURA"<<endl;  
    cin>>nn;  
};  
void graf::triaa()  
{  
    int nn1=nn;  
    while (nn1>=1)  
    {  
        for (int i=1;i<=nn1;i++)  
        {  
            cout<<"*";  
        };  
        cout<<endl;  
        nn1--;  
    };  
};  
void graf::triab()  
{  
    int nn1,i,c;  
    c=1;  
    nn1=nn;  
    do  
    {  
        for(i=1;i<=nn-c;i++)  
        {  
            cout<<" ";  
        }  
        for (int j=i;j<=nn;j++)  
        {  
            cout<<"*";  
        }  
        cout<<endl;  
        nn1--;  
        c++;  
    }while(nn1>=1);  
};  
void graf::triac()  
{
```

```
int nn1,c;
nn1=nn;
c=1;
do
{
for (int i=1;i<=nn1;i++)
{
cout<<"*";
}
cout<<endl;
for(int j=1;j<=c;j++)
{
cout<<" ";
}
c++;
nn1--;
} while(nn1>=1);
};

void graf::triad()
{
int nn1,i,c;
nn1=nn;
c=1;
do
{
for (i=1;i<=c;i++)
{
cout<<"*";
}
for (int j=i;j<=nn;j++)
{
cout<<" ";
}
cout<<endl;
c++;
nn1--;
}while(nn1>=1);
};

void graf::rombo()
{
int nn1,aux,aux1,c,i,ct;
nn1=nn;
ct=1;
aux=nn1/2;
aux1=aux;
c=1;
do
```

```
{
for (i=1;i<=aux;i++)
{
cout<<" ";
}
for (int j=i;j<=aux+c;j++)
{
cout<<"*";
}
for (int k=j;k<=nn1;k++)
{
cout<<" ";
}
c=c+2;
aux=aux-1;
cout<<endl;
ct=ct+1;
}while(ct<=aux1+1);
int ce,cas,aux2,l,m,ctt;
ce=1;
aux2=nn/2;
cas=nn-2;
ctt=1;
do
{
for (l=1;l<=ce;l++)
{
cout<<" ";
}
for (m=1;m<=cas;m++)
{
cout<<"*";
}
for(int n=m+1;n<=nn;n++)
{
cout<<" ";
}
cout<<endl;
ce++;
cas=cas-2;
ctt++;
}while(ctt<=aux2);
};
void main()
{
clrscr();
```



```
graf yo;  
yo.menu();  
};
```

En el código anterior se muestra en negrilla la sintaxis de las estructuras de repetición if, else, do while, for.

Realizar un programa que ingrese la marca, modelo, precio, imei y frecuencia de un celular además de los datos del almacén y permita ingresar un nuevo celular, buscar un celular dado su imei, mostrar todos los celulares existentes.

```
#include<iostream.h>  
#include<conio.h>  
#include<string.h>  
#include<stdio.h>  
class celu  
{  
public:  
char marc[15];  
char model[10];  
char frec[15];  
char imei[15];  
float precio;  
void leer();  
void mostrar();  
};  
void celu::leer()  
{  
cout<<"INGRESE LA MARCA DEL CELULAR"<<endl;  
gets(marc);  
cout<<"INGRESE EL MODELO DEL CELULAR"<<endl;  
gets(model);  
cout<<"INGRESE EL IMEI DEL CELULAR"<<endl;  
gets(imei);  
cout<<"INGRESE LA FRECUENCIA DEL CELULAR"<<endl;  
gets(frec);  
cout<<"INGRESE EL PRECIO DEL CELULAR"<<endl;  
cin>>precio;  
}  
void celu::mostrar()  
{  
cout<<"LA MARCA DEL CELULAR ES:"<<marc<<endl;  
cout<<"EL MODELO DEL CELULAR ES:"<<model<<endl;  
cout<<"LA FRECUENCIA DEL CELULAR ES:"<<frec<<endl;  
cout<<"EL IMEI DEL CELULAR ES: "<<imei<<endl;  
cout<<"EL PRECIO DEL CELULAR ES:"<<precio<<endl;
```

```
}  
class alm  
{  
public:  
int ncel;  
char nom[15];  
char ruc[12];  
char nompro[20];  
celu vcel[100];  
void leer();  
void mostrar();  
};  
void alm::leer()  
{  
char op;  
int i;  
i=0;  
cout<<"INGRESE EL NOMBRE DEL ALMACEN "<<endl;  
gets(nom);  
cout<<"INGRESE EL RUC DEL PROPIETARIO "<<endl;  
gets(ruc);  
cout<<"INGRESE EL NOMBRE DEL PROPIETARIO "<<endl;  
gets(nompro);  
do  
{  
vcel[i].leer();  
i++;  
cout<<"DESEA INGRESAR OTRO CELULAR S/N"<<endl;  
cin>>op;  
}while ((op=='s')||(op=='S'));  
ncel=i;  
};  
void alm::mostrar()  
{  
char cla[15];  
cout<<"INGRESE LA CLAVE DEL CELULAR A BUSCAR"<<endl;  
gets(cla);  
for (int i=0;i<=ncel;i++)  
{  
if ((strcmp(cla,vcel[i].imei))==0)  
{  
vcel[i].mostrar();  
};  
};  
};  
void main()  
{
```

```
alm yo;  
yo.leer();  
yo.mostrar();  
getch();  
}
```

5.- ENCAPSULAMIENTO.

Es la propiedad que permite proteger a los datos de manipulaciones no autorizadas, el encapsulamiento se lo realiza a través de los métodos **SET** y **GET**.

El siguiente código realiza las mismas tareas que el código anterior, pero en este caso se utiliza encapsulamiento.

Realizar un programa que ingrese la marca, modelo, precio, imei y frecuencia de un celular además de los datos del almacén y permita ingresar un nuevo celular, buscar un celular dado su imei, mostrar todos los celulares existentes, utilizando la técnica de encapsulamiento.

```
#include<iostream.h>  
#include<conio.h>  
#include<string.h>  
#include<stdio.h>  
class celu  
{  
private:  
char marc[15];  
char model[10];  
char frec[15];  
char imei[15];  
float precio;  
public:  
void setmarc(char *nmarc);  
char *getmarc();  
void setmodel(char *nmodel);  
char *getmodel();  
void setfrec(char *nfrec);  
char *getfrec();  
void setimei(char *nimei);  
char *getimei();  
void setprecio(float nprecio);  
float getprecio();  
void leer();  
void mostrar();  
};
```

```
void celu::setmarc(char *nmarc)
{
    strcpy(marc,nmarc);
}
char *celu::getmarc()
{
    return(marc);
}
void celu::setmodel(char *nmodel)
{
    strcpy(model,nmodel);
}

char *celu::getmodel()
{
    return(model);
}
void celu::setfrec(char *nfrec)
{
    strcpy(frec,nfrec);
}
char *celu::getfrec()
{
    return(frec);
}

void celu::setimei(char *nimei)
{
    strcpy(imei,nimei);
}
char *celu::getimei()
{
    return(imei);
}
void celu::setprecio(float nprecio)
{
    precio=nprecio;
}

float celu::getprecio()
{
    return(precio);
}
void celu::leer()
{
    cout<<"INGRESE LA MARCA DEL CELULAR"<<endl;
    gets(marc);
}
```

```
cout<<"INGRESE EL MODELO DEL CELULAR"<<endl;
gets(model);
cout<<"INGRESE EL IMEI DEL CELULAR"<<endl;
gets(imei);
cout<<"INGRESE LA FRECUENCIA DEL CELULAR"<<endl;
gets(frec);
cout<<"INGRESE EL PRECIO DEL CELULAR"<<endl;
cin>>precio;
}
void celu::mostrar()
{
cout<<"LA MARCA DEL CELULAR ES:"<<marc<<endl;
cout<<"EL MODELO DEL CELULAR ES:"<<model<<endl;
cout<<"LA FRECUENCIA DEL CELULAR ES:"<<frec<<endl;
cout<<"EL IMEI DEL CELULAR ES: "<<imei<<endl;
cout<<"EL PRECIO DEL CELULAR ES:"<<precio<<endl;
}
class alm
{
private:
int ncel;
char nom[15];
char ruc[12];
char nompro[20];
celu vcel[100];
public:
void setnecl(int nncel);
int getncel();
void setnom(char *nnom);
char *getnom();
void setruc(char *nruc);
char *getruc();
void setnompro(char *nnompro);
char *getnompro();
void leer();
void mostrar();
};
void alm::setnecl(int nncel)
{
ncel=nncel;
}
int alm::getncel()
{
return(ncel);
}
void alm::setnom(char *nnom)
{
```

```
strcpy(nom,nnom);
}
char *alm::getnom()
{
    return(nom);
}
void alm::setruc(char *nruc)
{
    strcpy(ruc,nruc);
}
char *alm::getruc()
{
    return(ruc);
}
void alm::setnompro(char *nnompro)
{
    strcpy(nompro,nnompro);
}
char *alm::getnompro()
{
    return(nompro);
}
void alm::leer()
{
    char op;
    int i;
    i=0;
    cout<<"INGRESE EL NOMBRE DEL ALMACEN "<<endl;
    gets(nom);
    cout<<"INGRESE EL RUC DEL PROPIETARIO "<<endl;
    gets(ruc);
    cout<<"INGRESE EL NOMBRE DEL PROPIETARIO "<<endl;
    gets(nompro);
    do
    {
        vcel[i].leer();
        i++;
        cout<<"DESEA INGRESAR OTRO CELULAR S/N"<<endl;
        cin>>op;
    }while ((op=='s')||(op=='S'));
    ncel=i;
}

void alm::mostrar()
{
    char cla[15];
    cout<<"INGRESE LA CLAVE DEL CELULAR A BUSCAR"<<endl;
```

```
gets(cla);
for (int i=0;i<=ncel;i++)
{
    if ((strcmp(cla,vcel[i].getimei()))==0)
    {
        vcel[i].mostrar();
    }
}
}
```

```
void main()
{
    clrscr();
    alm yo;
    yo.leer();
    clrscr();
    yo.mostrar();
    getch();
}
```

Antes de pasar al siguiente tema revisaremos lo que son los constructores y destructores de una clase determinada.

Constructor.- es el primer método que se ejecuta de una determinada clase, este método es el encargado de construir, reservar espacio en memoria e inicializar las variables de una clase determinada.

Destructor.- este método es el último que se invoca pues hace todo lo contrario que el constructor y esto es, libera el espacio en memoria y destruye los objetos creados por el constructor.

Si el programador no define un constructor y un destructor en la clase C++ se encarga de crearlo automáticamente.

En el siguiente código se muestra la implementación de constructores y destructores en el código del programa de celulares que ya lo revisamos anteriormente.

/*programa con encapsulamiento constructores y destructores ingresa datos del almacén y de celulares y muestra un celular dado su imei*/

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

```
class cel
{
private:
```

```
char marca[20];
char model[10];
char frec[20];
char imei[15];
float precio;
public:
cel();
void setmarca(char *nmarca);
char *getmarca();
void setmodel(char *nmodel);
char *getmodel();
void setfrec(char *nfrec);
char *getfrec();
void setimei(char *nimei);
char *getimei();
void setprecio(float nprecio);
float getprecio();
void leer();
void mostrar();
~cel();
}
cel::cel()
{
cout<<"CONSTRUYENDO CEL"<<endl;
};
void cel::setmarca(char *nmarca)
{
    strcpy(marca,nmarca);
}
char *cel::getmarca()
{
    return(marca);
}
void cel::setmodel(char *nmodel)
{
    strcpy(model,nmodel);
}
char *cel::getmodel()
{
    return(model);
}
void cel::setfrec(char *nfrec)
{
    strcpy(frec,nfrec);
}
char *cel::getfrec()
{
```



```

return(frec);
}
void cel::setimei(char *nimei)
{
    strcpy(imei,nimei);
}
char *cel::getimei()
{
    return(imei);
}
void cel::setprecio(float nprecio)
{
    precio=nprecio;
}
float cel::getprecio()
{
    return(precio);
}
void cel::leer()
{
    cout<<"INGRESE LA MARCA DEL CELULAR"<<endl;
    gets(marca);
    cout<<"INGRESE EL MODELO DEL CELULAR"<<endl;
    gets(model);
    cout<<"INGRES LA FRECUENCIA DEL CELULAR"<<endl;
    gets(frec);
    cout<<"INGRES EL IMEI DEL CELULAR"<<endl;
    gets(imei);
    cout<<"INGRESE EL PRECIO DEL CELULAR"<<endl;
    cin>>precio;
}
void cel::mostrar()
{
    cout<<"LA MARCA DEL CELULAR ES: "<<marca<<endl;
    cout<<"LA FRECUENCIA DEL CELULAR ES: "<<frec<<endl;
    cout<<"EL MODELO DEL CELULAR ES: "<<model<<endl;
    cout<<"EL IMEI DEL CELULAR ES: "<<imei<<endl;
    cout<<"EL PRECIO DEL CELULAR ES: "<<precio<<endl;
}
cel::~~cel()
{
    cout<<"destruyendo cel"<<endl;
}
class alm
{
private:
    char nomal[20];

```

```
char nompro[30];
char ruc[14];
int ncel;
cel vcel[100];
public:
alm();
void setnomal(char *nnomal);
char *getnomal();
void setnompro(char *nnompro);
char *getnompro();
void setruc(char *nruc);
char *getruc();
void setncel(int nncel);
float getncel();
void llenar();
void busc();
~alm();
}
alm::alm()
{
cout<<"CONSTRUYENDO ALM"<<endl;
}
void alm::setnomal(char *nnomal)
{
    strcpy(nomal,nnomal);
}
char *alm::getnomal()
{
    return(nomal);
}
void alm::setnompro(char *nnompro)
{
    strcpy(nompro,nnompro);
}
char *alm::getnompro()
{
    return(nompro);
}
void alm::setruc(char *nruc)
{
    strcpy(ruc,nruc);
}
char *alm::getruc()
{
    return(ruc);
}
void alm::setncel(int nncel)
```

```
{
    ncel=nncel;
}
float alm::getncel()
{
    return(ncel);
}
void alm::llenar()
{
    cout<<"INGRESE EL NOMBRE DEL ALMACEN"<<endl;
    gets(nomal);
    cout<<"INGRESE EL NOMBRE DEL PROPIETARIO"<<endl;
    gets(nompro);
    cout<<"INGRESE EL RUC DEL PROPIETARIO"<<endl;
    gets(ruc);
    int i=0;
    char op;
    do
    {
        vcel[i].leer();
        i++;
        cout<<"DESEA INGRESAR OTRO CELULAR S/N"<<endl;
        cin>>op;

    }while(op=='s');
    setncel(i);
}
void alm::busc()
{
    char busim[15];
    cout<<"INGRESE EL IMEI DEL CELULAR A BUSCAR"<<endl;
    gets(busim);
    for(int j=0;j<=ncel;j++)
    {
        if ((strcmp(vcel[j].getimei(),busim))==0)
        {
            vcel[j].mostrar();
        }
    }
}
alm::~~alm()
{
    cout<<"destruyendo alm"<<endl;
}
void main()
{
    clrscr();
```

```
alm yo;  
yo.llenar();  
yo.busc();  
getch();  
}
```

6.- HERENCIA.

Es el mecanismo que permite crear una clase a partir de otra clase pre existente es decir comparte automáticamente métodos y datos entre clases, subclases y objetos, existen dos tipos que son:

- Herencia Simple. – en este tipo de herencia se pueden derivar cuantas clases sean necesarias a partir de una clase existente “CLASE BASE” incluso cuando esta ultima sea una clase derivada, es decir una clase derivada puede ser clase base de otra clase y así sucesivamente.
- Herencia Múltiple. – este tipo de herencia se la tiene cuando una clase derivada puede tener varias clases base, es decir cuando una clase derivada hereda los atributos de más de una clase base.

Realizar un programa que utilice herencia para que dadas las coordenadas de un punto “centro”, calcule el área y volumen de un círculo y de un cilindro con los datos de coordenadas ingresados, además que muestre las coordenadas, áreas, volúmenes de acuerdo a la figura mostrada.

/*Programa con encapsulamiento y herencia desde el punto al círculo y cilindro muestra las coordenadas del punto de origen así como el radio la altura el área y el volumen*/

```
#include<iostream.h>  
#include<conio.h>  
#include<stdio.h>  
class punto  
{  
    float x,y;  
public:  
    void setx(float nx);  
    float getx();  
    void sety(float ny);  
    float gety();  
    void leer();  
    void mostrar();  
    punto();  
    ~punto();  
};
```

```
void punto::setx(float nx)
{
    x=nx;
};
float punto::getx()
{
    return(x);
};
void punto::sety(float ny)
{
    y=ny;
}
float punto::gety()
{
    return(y);
};
void punto::leer()
{
    cout<<"INGRESE LA COORDENADA X "<<endl;
    cin>>x;
    cout<<"INGRESE LA COORDENADA Y "<<endl;
    cin>>y;
};
void punto::mostrar()
{
    cout<<"LA COORDENADA X ES: "<<x<<endl;
    cout<<"LA COORDENADA Y ES: "<<y<<endl;
};
punto::punto()
{
    cout<<"CONSTRUYENDO PUNTO"<<endl;
    setx(0);
    sety(0);
};
punto::~~punto()
{
    cout<<"destruyendo punto"<<endl;
};
class cir:public punto
{
    float r;
public:
    void setr(float nr);
    float getr();
    void leer();
    void mostrar();
    cir();
```

```

~cir();
};
void cir::setr(float nr)
{
    r=nr;
};
float cir::getr()
{
    return(r);
};
void cir::leer()
{
    float cx,cy;
    cout<<"INGRESE LA COORDENADA X"<<endl;
    cin>>cx;
    setx(cx);
    cout<<"INGRESE LA COORDENADA Y"<<endl;
    cin>>cy;
    sety(cy);
    cout<<"INGRESE EL RADIO"<<endl;
    cin>>r;
};
void cir::mostrar()
{
    float ar;
    ar=3.1416*r*r;
    cout<<"LA COORDENADA X ES: "<<getx()<<endl;
    cout<<"LA COORDENADA Y ES: "<<gety()<<endl;
    cout<<"EL RADIO ES: "<<r<<endl;
    cout<<"EL AREA ES: "<<ar<<endl;
};
cir::cir()
{
    cout<<"CONSTRUYENDO CIR"<<endl;
    setr(0);
};
cir::~~cir()
{
    cout<<"destruyendo cir"<<endl;
};
class cil:public cir
{
    float h;
public:
    void seth(float nh);
    float geth();
    void leer();
};

```

```
void mostrar();
cil();
~cil();
};
void cil::seth(float nh)
{
    h=nh;
};
float cil::geth()
{
    return(h);
}
void cil::leer()
{
    float cx,cy,nr;
    cout<<"INGRESE LA COORDENADA X"<<endl;
    cin>>cx;
    setx(cx);
    cout<<"INGRESE LA COORDENADA Y"<<endl;
    cin>>cy;
    sety(cy);
    cout<<"INGRESE EL RADIO"<<endl;
    cin>>nr;
    setr(nr);
    cout<<"INGRESE LA ALTURA DEL CILINDRO"<<endl;
    cin>>h;
};
void cil::mostrar()
{
    float ar,v;
    ar=3.1416*getr()*getr();
    cout<<"LA COORDENADA X ES: "<<getx()<<endl;
    cout<<"LA COORDENADA Y ES: "<<gety()<<endl;
    cout<<"EL RADIO ES: "<<getr()<<endl;
    cout<<"EL AREA ES: "<<ar<<endl;
    v=(3.1416*getr()*getr()*geth());
    cout<<"EL VOLUMEN DEL CILINDRO ES: "<<v<<endl;
    cout<<"LA ALTURA DEL CILINDRO ES:"<<h<<endl;
};
cil::cil()
{
    cout<<"CONSTRUYENDO CIL"<<endl;
    seth(0);
};

cil::~~cil()
{

```

```
cout<<"destruyendo cil"<<endl;
};
void main()
{
    punto yo;
    yo.leer();
    yo.mostrar();
    cir yo1;
    yo1.leer();
    yo1.mostrar();
    cil yo2;
    yo2.leer();
    yo2.mostrar();
    getch();
};
```

Realizar un programa que cree la nomina de trabajadores por horas y por nombramiento, muestre todos los datos de los trabajadores además de buscar a un trabajador dado su numero de cedula, utilizando las técnicas de poo aprendidas.

```
#include<conio.h>
#include<iostream.h>
#include<string.h>
#include<stdio.h>
class trab
{
    char nom[50];
    char ci[12];
public:
    trab();
    void setnom(char *nnom);
    char *getnom();
    void setci(char *nci);
    char *getci();
    void leer();
    void mostrar();
    ~trab();
};
trab::trab()
{
    cout<<"CONSTRUYENDO TRAB"<<endl;
    setnom("");
    setci("");
};
void trab::setnom(char *nnom)
{
    strcpy(nom,nnom);
```



```
};  
char *trab::getnom()  
{  
    return(nom);  
};  
void trab::setci(char *nci)  
{  
    strcpy(ci,nci);  
};  
char *trab::getci()  
{  
    return(ci);  
};  
void trab::leer()  
{  
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;  
    gets(nom);  
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;  
    cin>>ci;  
};  
void trab::mostrar()  
{  
    cout<<"NOMBRE: "<<nom<<endl;  
    cout<<"CEDULA: "<<ci<<endl;  
};  
trab::~~trab()  
{  
    cout<<"destruyendo trab"<<endl;  
};  
class trabh:public trab  
{  
    float nh,vh;  
public:  
    trabh();  
    void setnh(float nnh);  
    float getnh();  
    void setvh(float nvh);  
    float getvh();  
    void leer();  
    void mostrar();  
    ~trabh();  
};  
trabh::trabh()  
{  
    cout<<"CONSTRUYENDO TRABH"<<endl;  
    setnh(0);  
    setvh(0);
```

```
};  
void trabh::setnh(float nnh)  
{  
    nh=nnh;  
};  
float trabh::getnh()  
{  
    return(nh);  
};  
void trabh::setvh(float nvh)  
{  
    vh=nvh;  
};  
float trabh::getvh()  
{  
    return(vh);  
};  
void trabh::leer()  
{  
    char nom1[25];  
    char ci1[12];  
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;  
    gets(nom1);  
    setnom(nom1);  
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;  
    cin>>ci1;  
    setci(ci1);  
    cout<<"INGRESE EL NUMERO DE HORAS LABORADAS"<<endl;  
    cin>>nh;  
    cout<<"INGRESE EL VALOR DE LA HORA LABORADA"<<endl;  
    cin>>vh;  
};  
void trabh::mostrar()  
{  
    cout<<"NOMBRE: "<<getnom()<<endl;  
    cout<<"CEDULA NUMERO: "<<getci()<<endl;  
    cout<<"EL NUMERO DE HORAS LABORADAS ES: "<<nh<<endl;  
    cout<<"EL VALOR DE LA HORA LABORADA ES: "<<vh<<endl;  
};  
trabh::~~trabh()  
{  
    cout<<"destruyendo trabh"<<endl;  
};  
class trabh::public trab  
{  
    float suel,des;  
public:
```

```
trabn();
void setsuel(float nsuel);
float getsuel();
void setdes(float ndes);
float getdes();
void leer();
void mostrar();
~trabn();
};
trabn::trabn()
{
    cout<<"CONSTRUYENDO TRABN"<<endl;
    setsuel(0);
    setdes(0);
};
void trabn::setsuel(float nsuel)
{
    suel=nsuel;
};
float trabn::getsuel()
{
    return(suel);
};
void trabn::setdes(float ndes)
{
    des=ndes;
};
float trabn::getdes()
{
    return(des);
};
void trabn::leer()
{
    char nom1[25];
    char ci1[12];
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;
    gets(nom1);
    setnom(nom1);
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;
    cin>>ci1;
    setci(ci1);
    cout<<"INGRESE EL VALOR DEL SALARIO"<<endl;
    cin>>suel;
    cout<<"INGRESE EL VALOR DEL DESCUENTO"<<endl;
    cin>>des;
};
void trabn::mostrar()
```

```
{
    cout<<"NOMBRE: "<<getnom()<<endl;
    cout<<"CEDULA NUMERO: "<<getci()<<endl;
    cout<<"SALARIO: "<<suel<<endl;
    cout<<"DESCUENTO: "<<des<<endl;
};
trabn::~trabn()
{
    cout<<"destruyendo trabn"<<endl;
};
class nom
{
private:
    int cth;
    int ctn;
    char nom_emp[50];
    trabh vth[100];
    trabn vtn[100];
public:
    nom();
    void setcth(int nth);
    int getcth();
    void setctn(int nctn);
    int getctn();
    void setnom_emp(char *nnom_emp);
    char *getnom_emp();
    void leer();
    void mostrar();
    void busced();
    ~nom();
};
nom::nom()
{
    cout<<"CONSTRUYENDO NOM"<<endl;
    setnom_emp("");
    setcth(0);
    setctn(0);
};
void nom::setcth(int nth)
{
    cth=nth;
};
int nom::getcth()
{
    return(cth);
};
void nom::setctn(int nctn)
```

```
{
    ctn=nctn;
};
int nom::getctn()
{
    return(ctn);
};
void nom::setnom_emp(char *nnom_emp)
{
    strcpy(nom_emp,nnom_emp);
};
char *nom::getnom_emp()
{
    return(nom_emp);
};
void nom::leer()
{
    int op,th,tn;
    char op1;
    th=0;
    tn=0;
    cout<<"INGRESE EL NOMBRE DE LA EMPRESA"<<endl;
    gets(nom_emp);
    do
    {
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- SI EL TRABAJADOR A INGRESAR ES POR HORAS"<<endl;
        cout<<"2.- SI EL TRABAJADOR A INGRESAR ES POR HORAS"<<endl;
        cin>>op;
        if (op==1)
        {
            vth[th].leer();
            th++;
        };
        if (op==2)
        {
            vtn[tn].leer();
            tn++;
        };
        cout<<"DESEA INGRESAR OTRO TRABAJADOR....?? S/N"<<endl;
        cin>>op1;
    }while ((op1=='s')||(op1=='S'));
    cth=th;
    ctn=tn;
};
void nom::mostrar()
{
```

```
cout<<"EMPRESA: "<<nom_emp<<endl<<endl<<endl;
cout<<"TRABAJADORES POR HORAS"<<endl;
for (int i=0;i<cth;i++)
{
    vth[i].mostrar();
};
cout<<endl<<endl<<endl;
cout<<"TRABAJADORES POR NOMBRAMIENTO"<<endl;
for (int j=0;j<ctn;j++)
{
    vtn[j].mostrar();
};
cout<<endl<<endl<<endl;
};
void nom::busced()
{
    int th,tn,aux,tt,ct;
    th=0;
    tn=0;
    aux=0;
    char busci[12];
    cout<<"INGRESE LA CEDULA DEL TRABAJADOR A BUSCAR"<<endl;
    cin>>busci;
    do
    {
        if ((strcmp(vth[th].getci(),busci))==0)
        {
            vth[th].mostrar();
            aux=1;
        }
        else
        {
            th++;
        };
        if (aux==0)
        {
            if ((strcmp(vtn[tn].getci(),busci))==0)
            {
                vtn[tn].mostrar();
                aux=1;
            }
            else
            {
                tn++;
            };
        };
    }while(aux==0);
```

```
};  
  
nom::~~nom()  
{  
    cout<<"destruyendo nom"<<endl;  
};  
void main()  
{  
    clrscr();  
    nom yo;  
    yo.leer();  
    yo.mostrar();  
    yo.buscado();  
    getch();  
};
```

7.- POLIMORFISMO.

Es la facultad de asumir muchas formas y llegar a un mismo fin, objetos de clases distintas responden de diferente manera ante un mismo mensaje.

Realizar un programa que ingrese los datos de estudiantes, estudiantes artistas y estudiantes deportistas, muestre los estudiantes con su calificación y si ingresan a examen se exoneran o reprueban muestra los artistas que tocan guitarra y los deportistas de un deporte dado y que tienen un acumulado mayor a 27.

```
#include<iostream.h>  
#include<string.h>  
#include<stdio.h>  
#include<conio.h>  
class est  
{  
    char nom[25];  
    char cod[5];  
    float nota;  
public:  
    est();  
    void setnom(char *nnom);  
    char *getnom();  
    void setcod(char *ncod);  
    char *getcod();  
    void setnota(float nnota);  
    float getnota();  
    void leer();  
    void mostrar();  
    ~est();
```

```
};
est::est()
{
    cout<<"CONSTRUYENDO EST"<<endl;
    setnom("");
    setcod("");
    setnota(0);
};
void est::setnom(char *nnom)
{
    strcpy(nom,nnom);
};
char *est::getnom()
{
    return(nom);
};
void est::setcod(char *ncod)
{
    strcpy(cod,ncod);
};
char *est::getcod()
{
    return(cod);
};
void est::setnota(float nnota)
{
    nota=nnota;
};
float est::getnota()
{
    return(nota);
};
void est::leer()
{
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
    gets(nom);
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
    cin>>cod;
    cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
    cin>>nota;
};
void est::mostrar()
{
    cout<<"NOMBRE: "<<nom<<endl;
    cout<<"CODIGO: "<<cod<<endl;
    cout<<"NOTA: "<<nota<<endl;
};
```



```
est::~~est()
{
    cout<<"destruyendo est"<<endl;
};
class estart:public est
{
    char act[15];
public:
    estart();
    void setact(char *nact);
    char *getact();
    void leer();
    void mostrar();
    ~estart();
};
estart::estart()
{
    cout<<"CONSTRUYENDO ESTART"<<endl;
    setact("");
};
void estart::setact(char *nact)
{
    strcpy(act,nact);
};
char *estart::getact()
{
    return(act);
};
void estart::leer()
{ char nom1[15];
  char cod1[5];
  float nota1;
  cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
  gets(nom1);
  setnom(nom1);
  cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
  cin>>cod1;
  setcod(cod1);
  cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
  cin>>nota1;
  setnota(nota1);
  cout<<"INGRESE LA ACTIVIDAD QUE REALIZA"<<endl;
  cin>>act;
};
void estart::mostrar()
{
    cout<<"NOMBRE: "<<getnom()<<endl;
```

```
cout<<"CODIGO: "<<getcod()<<endl;
cout<<"NOTA: "<<getnota()<<endl;
cout<<"PRACTICA: "<<act<<endl;
};
estart::~estart()
{
    cout<<"destruyendo art"<<endl;
};
class estdep:public est
{
    char dep[15];
    char lugar[];
public:
    estdep();
    void setdep(char *ndep);
    char *getdep();
    void setlugar(char *nlugar);
    char *getlugar();
    void leer();
    void mostrar();
    ~estdep();
};
estdep::~estdep()
{
    cout<<"CONSTRUYENDO ESTDEP"<<endl;
    setdep("");
    setlugar("");
};
void estdep::setdep(char *ndep)
{
    strcpy(dep,ndep);
};
char *estdep::getdep()
{
    return(dep);
};
void estdep::setlugar(char *nlugar)
{
    strcpy(lugar,nlugar);
};
char *estdep::getlugar()
{
    return(lugar);
};
void estdep::leer()
{ char nom1[15];
  char cod1[5];
```

```
float nota1;
cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
gets(nom1);
setnom(nom1);
cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
cin>>cod1;
setcod(cod1);
cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
cin>>nota1;
setnota(nota1);
cout<<"INGRESE EL DEPORTE QUE PARCTICA"<<endl;
cin>>dep;
cout<<"INGRESE EL LUGAR OBTENIDO EN LOS ULTIMOS JUEGOS"<<endl;
cin>>lugar;
};
void estdep::mostrar()
{
cout<<"NOMBRE: "<<getnom()<<endl;
cout<<"CODIGO: "<<getcod()<<endl;
cout<<"NOTA: "<<getnota()<<endl;
cout<<"PRACTICA: "<<dep<<endl;
cout<<"OBTUVIERON EL "<<lugar<<" EN LOS ULTIMOS JUEGOS"<<endl;
};
estdep::~~estdep()
{
cout<<"destruyendo dep"<<endl;
};
class lista
{
char mat[15];
estart vea[30];
estdep ved[30];
est ves[30];
int ced,cea,ce;
public:
void setmat(char *mat);
char *getmat();
void setcea(int ncea);
int getcea();
void setced(int nced);
int getced();
void setce(int nce);
int getce();
void leer();
void mostrar();
void busdep();
void busart();
```

```
~lista();
};
void lista::setmat(char *nmat)
{
    strcpy(mat,nmat);
};
char *lista::getmat()
{
    return(mat);
};
void lista::setcea(int ncea)
{
    cea=ncea;
};
int lista::getcea()
{
    return(cea);
};
void lista::setced(int nced)
{
    ced=nced;
};
int lista::getced()
{
    return(ced);
};
void lista::setce(int nce)
{
    ce=nce;
};
int lista::getce()
{
    return(ce);
};
void lista::leer()
{
    int op1;
    char op;
    op='s';
    cout<<"INGRESE LA MATERIA"<<endl;
    cin>>mat;
    while (op=='s')
    {
        cout<<"ELIJA UNBA OPCION "<<endl;
        cout<<"1.- INGRESAR UN ESTUDIANTE "<<endl;
        cout<<"2.- INGRESAR UN ESTUDIANTE DEPORTISTA"<<endl;
        cout<<"3.- INGRESAR UN ESTUDIANTE ARTISTA"<<endl;
```

```
cin>>op1;
if (op1==1)
{
    ves[ce].leer();
    ce++;
};
if (op1==2)
{
    ved[ced].leer();
    ced++;
};
if (op1==3)
{
    vea[cea].leer();
    cea++;
};
cout<<"DESEA INGRESAR OTRO ESTUDIANTE S/N"<<endl;
cin>>op;
};
};
void lista::mostrar()
{
    int op;
    cout<<"ELIJA UNA OPCION"<<endl;
    cout<<"1.- MOSTRAR LOS ESTUDIANTES EXONERADOS"<<endl;
    cout<<"2.- MOSTRAR LOS ESTUDIANTES CON DERECHO A EXAMEN
PRINCIPAL"<<endl;
    cout<<"3.- MOSTRAR LOS ESTUDIANTES REPROBADOS"<<endl;
    cin>>op;
    if (op==1)
    {
        cout<<"MATERIA: "<<mat<<endl;
        cout<<"LOS ESTUDIANTES EXONERADOS SON:"<<endl<<endl;
        for (int i=0;i<cea;i++)
        {
            if((vea[i].getnota()+1)>=25)
            {
                vea[i].mostrar();
            };
        };
        for (int l=0;l<ced;l++)
        {
            if((ved[l].getnota()+2)>=25)
            {
                ved[l].mostrar();
            };
        };
    };
};
```

```
for (int h=0;h<ce;h++)
{
    if(ves[i].getnota()>=25)
    {
        ves[i].mostrar();
    };
};

if (op==2)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES CON DERECHO A EXAMEN PRINCIPAL
SON:"<<endl<<endl;
    for (int i=0;i<cea;i++)
    {
        if((vea[i].getnota()+1>=4)&&(vea[i].getnota()+1<=23))
        {
            vea[i].mostrar();
        };
    };
    for (int j=0;j<ced;j++)
    {
        if((ved[i].getnota()+2>=4)&&(vea[i].getnota()+2<=23))
        {
            ved[i].mostrar();
        };
    };
    for (int k=0;k<ce;k++)
    {
        if(ves[i].getnota()<=23)
        {
            ves[i].mostrar();
        };
    }
};

if (op==3)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES REPROBADOS SON:"<<endl<<endl;
    for (int i=0;i<cea;i++)
    {
        if((vea[i].getnota()+1<=3))
        {
            vea[i].mostrar();
        };
    };
};
```

```
for (int m=0;m<ced;m++)
{
    if((ved[i].getnota()+2<=3))
    {
        ved[i].mostrar();
    };
};
for (int p=0;p<ce;p++)
{
    if(ves[i].getnota()<=3)
    {
        ves[i].mostrar();
    };
};
};
void lista::busdep()
{
    char dep1[15];
    cout<<"INGRESE EL DEPORTE A BUSCAR"<<endl;
    cin>>dep1;
    clrscr();
    cout<<"LOS ESTUDIANTES DEPORTIATAS QUE PRACTICAN EL DEPORTE
INGRESADO Y QUE TIENEN UN ACUNULADO MAYOR A 27 SON: "<<endl;
    for (int i=0;i<getced();i++)
    {
        if (((strcmp(dep1,ved[i].getdep()))==0)&&(ved[i].getnota()+2>=27))
        {
            ved[i].mostrar();
        }
    }
};
void lista::busart()
{
    cout<<"LOS ESTUDIANTES QUE ENTONAN GUITARRA SON: "<<endl;
    for (int i=0;i<=getcea();i++)
    {
        if ((strcmp(vea[i].getact(),"guitarra")==0)
        {
            vea[i].mostrar();
        };
    };
};
lista::~~lista()
{
    cout<<"destruyendo lista"<<endl;
};
```

```
void main ()  
{  
  clrscr();  
  lista yo;  
  yo.leer();  
  yo.mostrar();  
  yo.busdep();  
  yo.busart();  
  getch(); };
```

*ALGUNOS PROGRAMAS EN C++ UTILIZANDO LA TÉCNICA DE
PROGRAMACIÓN ORIENTADA A OBJETOS.*

1.- ESTE PROGRAMA ES LA NOMINA DE TRABAJADORES POR HORAS Y POR NOMBRAMIENTO, MUESTRA TODOS LOS DATOS DE LOS TRABAJADORES SDEMAS DE BUSCAR A UN TRABAJADOR DADO SU NUMERO DE CEDULA SE UTILIZA POLIMORFISMO

```
#include<conio.h>  
#include<iostream.h>  
#include<string.h>  
#include<stdio.h>  
class trab  
{  
  char nom[50];  
  char ci[12];  
  public:  
  trab();  
  void setnom(char *nnom);  
  char *getnom();  
  void setci(char *nci);  
  char *getci();  
  void leer();  
  void mostrar();  
  ~trab();  
};  
trab::trab()  
{  
  cout<<"CONSTRUYENDO TRAB"<<endl;  
  setnom("");  
  setci("");  
};  
void trab::setnom(char *nnom)  
{
```



```
strcpy(nom,nnom);
};
char *trab::getnom()
{
    return(nom);
};
void trab::setci(char *nci)
{
    strcpy(ci,nci);
};
char *trab::getci()
{
    return(ci);
};
void trab::leer()
{
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;
    gets(nom);
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;
    cin>>ci;
};
void trab::mostrar()
{
    cout<<"NOMBRE: "<<nom<<endl;
    cout<<"CEDULA: "<<ci<<endl;
};
trab::~~trab()
{
    cout<<"destruyendo trab"<<endl;
};
class trabh:public trab
{
    float nh,vh;
public:
    trabh();
    void setnh(float nnh);
    float getnh();
    void setvh(float nvh);
    float getvh();
    void leer();
    void mostrar();
    ~trabh();
};
trabh::trabh()
{
    cout<<"CONSTRUYENDO TRABH"<<endl;
    setnh(0);
```

```
setvh(0);
};
void trabh::setnh(float nnh)
{
    nh=nnh;
};
float trabh::getnh()
{
    return(nh);
};
void trabh::setvh(float nvh)
{
    vh=nvh;
};
float trabh::getvh()
{
    return(vh);
};
void trabh::leer()
{
    /*char nom1[25];
    char ci1[12];
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;
    gets(nom1);
    setnom(nom1);
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;
    cin>>ci1;
    setci(ci1);*/
    trab::leer();
    cout<<"INGRESE EL NUMERO DE HORAS LABORADAS"<<endl;
    cin>>nh;
    cout<<"INGRESE EL VALOR DE LA HORA LABORADA"<<endl;
    cin>>vh;
};
void trabh::mostrar()
{
    float tot;
    /*cout<<"NOMBRE: "<<getnom()<<endl;
    cout<<"CEDULA NUMERO: "<<getci()<<endl; */
    trab::mostrar();
    cout<<"EL NUMERO DE HORAS LABORADAS ES: "<<nh<<endl;
    cout<<"EL VALOR DE LA HORA LABORADA ES: "<<vh<<endl;
    tot=nh*vh;
    cout<<"RECIBE UN SALARIO DE: "<<tot<<" DOLARES"<<endl;
};
trabh::~~trabh()
{
```

```
cout<<"destruyendo trabh"<<endl;
};
class trahn:public trab
{
    float suel,des;
public:
    trahn();
    void setsuel(float nsuel);
    float getsuel();
    void setdes(float ndes);
    float getdes();
    void leer();
    void mostrar();
    ~trahn();
};
trahn::trahn()
{
    cout<<"CONSTRUYENDO TRABN"<<endl;
    setsuel(0);
    setdes(0);
};
void trahn::setsuel(float nsuel)
{
    suel=nsuel;
};
float trahn::getsuel()
{
    return(suel);
};
void trahn::setdes(float ndes)
{
    des=ndes;
};
float trahn::getdes()
{
    return(des);
};
void trahn::leer()
{
    /*char nom1[25];
    char ci1[12];
    cout<<"INGRESE EL NOMBRE DEL TRABAJADOR"<<endl;
    gets(nom1);
    setnom(nom1);
    cout<<"INGRESE EL NUMERO DE CEDULA"<<endl;
    cin>>ci1;
    setci(ci1);*/
```

```
trab::leer();
cout<<"INGRESE EL VALOR DEL SALARIO"<<endl;
cin>>suel;
cout<<"INGRESE EL VALOR DEL DESCUENTO"<<endl;
cin>>des;
};
void trabn::mostrar()
{
    float tot;
    /*cout<<"NOMBRE: "<<getnom()<<endl;
    cout<<"CEDULA NUMERO: "<<getci()<<endl;*/
    trab::mostrar();
    cout<<"SALARIO: "<<suel<<endl;
    cout<<"DESCUENTO: "<<des<<endl;
    tot=suel-des;
    cout<<"RECIBE UN SALARIO DE: "<<tot<<" DOLARES"<<endl;
};
trabn::~~trabn()
{
    cout<<"destruyendo trabn"<<endl;
};
class nom
{
private:
    int cth;
    int ctn;
    char nom_emp[50];
    trabh vth[100];
    trabn vtn[100];
public:
    nom();
    void setcth(int ncth);
    int getcth();
    void setctn(int nctn);
    int getctn();
    void setnom_emp(char *nnom_emp);
    char *getnom_emp();
    void leer();
    void mostrar();
    void busced();
    ~nom();
};
nom::nom()
{
    cout<<"CONSTRUYENDO NOM"<<endl;
    setnom_emp("");
    setcth(0);
```

```
    setctn(0);
};
void nom::setcth(int nth)
{
    cth=nth;
};
int nom::getcth()
{
    return(cth);
};
void nom::setctn(int nctn)
{
    ctn=nctn;
};
int nom::getctn()
{
    return(ctn);
};
void nom::setnom_emp(char *nnom_emp)
{
    strcpy(nom_emp,nnom_emp);
};
char *nom::getnom_emp()
{
    return(nom_emp);
};
void nom::leer()
{
    int op,th,tn;
    char op1;
    th=0;
    tn=0;
    cout<<"INGRESE EL NOMBRE DE LA EMPRESA"<<endl;
    gets(nom_emp);
    do
    {
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- SI EL TRABAJADOR A INGRESAR ES POR HORAS"<<endl;
        cout<<"2.- SI EL TRABAJADOR A INGRESAR ES POR HORAS"<<endl;
        cin>>op;
        if (op==1)
        {
            vth[th].leer();
            th++;
        };
        if (op==2)
        {
```

```

    vtn[tn].leer();
    tn++;
};
cout<<"DESEA INGRESAR OTRO TRABAJADOR....?? S/N"<<endl;
cin>>op1;
}while ((op1=='s')||(op1=='S'));
cth=th;
ctn=tn;
};
void nom::mostrar()
{
    cout<<"EMPRESA: "<<nom_emp<<endl<<endl<<endl;
    cout<<"TRABAJADORES POR HORAS"<<endl;
    for (int i=0;i<cth;i++)
    {
        vth[i].mostrar();
    };
    cout<<endl<<endl<<endl;
    cout<<"TRABAJADORES POR NOMBRAMIENTO"<<endl;
    for (int j=0;j<ctn;j++)
    {
        vtn[j].mostrar();
    };
    cout<<endl<<endl<<endl;
};
void nom::busced()
{
    int th,tn,aux,tt,ct;
    char busci[12];
    th=0;
    tn=0;
    aux=0;
    cout<<"INGRESE LA CEDULA DEL TRABAJADOR A BUSCAR "<<endl;
    gets(busci);
    while ((aux==0)&&(th<cth))
    {
        if ((strcmp(vth[th].getci(),busci))==0)
        {
            vth[th].mostrar();
            th++;
            aux=1;
        }
    };
    while ((aux==0)&&(tn<ctn))
    {
        if ((strcmp(vtn[tn].getci(),busci))==0)
        {

```

```
vtn[tn].mostrar();
tn++;
aux=1;
}
};
};
nom::~~nom()
{
    cout<<"destruyendo nom"<<endl;
};
void main()
{
    clrscr();
    nom yo;
    yo.leer();
    yo.mostrar();
    yo.buscado();
    getch();
};
```

2.- ESTE PROGRAMA INGRESA LOS DATOS DE ESTUDIANTES ,EST ARTISTAS Y EST DEPORTISTAS MUESTRA LOS EST CON SU CALIF Y SI INGRESAN A EXAMEN SE EXONERAN O REPRUEBAN MUESTRA LOS ART QUE TOCAN GUITARRA Y LOS DEP DE UN DEP DADO Y QUE TIENNE UN ACUMULADO MAYOR A 27 CON POLIMORFISMO.

```
#include<iostream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>
class est
{
    char nom[25];
    char cod[5];
    float nota;
public:
    est();
    void setnom(char *nnom);
    char *getnom();
    void setcod(char *ncod);
    char *getcod();
    void setnota(float nnota);
    float getnota();
    void leer();
    void mostrar();
    ~est();
```

```
};  
est::est()  
{  
    cout<<"COSTRUYENDO EST"<<endl;  
    setnom("");  
    setcod("");  
    setnota(0);  
};  
void est::setnom(char *nnom)  
{  
    strcpy(nom,nnom);  
};  
char *est::getnom()  
{  
    return(nom);  
};  
void est::setcod(char *ncod)  
{  
    strcpy(cod,ncod);  
};  
char *est::getcod()  
{  
    return(cod);  
};  
void est::setnota(float nnota)  
{  
    nota=nnota;  
};  
float est::getnota()  
{  
    return(nota);  
};  
void est::leer()  
{  
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;  
    gets(nom);  
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;  
    cin>>cod;  
    cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;  
    cin>>nota;  
};  
void est::mostrar()  
{  
    cout<<"NOMBRE: "<<nom<<endl;  
    cout<<"CODIGO: "<<cod<<endl;  
    cout<<"NOTA: "<<nota<<endl;  
};
```



```
est::~~est()
{
    cout<<"destruyendo est"<<endl;
};
class estart:public est
{
    char act[15];
public:
    estart();
    void setact(char *nact);
    char *getact();
    void leer();
    void mostrar();
    ~estart();
};
estart::estart()
{
    cout<<"CONSTRUYENDO ESTART"<<endl;
    setact("");
};
void estart::setact(char *nact)
{
    strcpy(act,nact);
};
char *estart::getact()
{
    return(act);
};
void estart::leer()
{
    /*char nom1[15];
    char cod1[5];
    float nota1;
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
    gets(nom1);
    setnom(nom1);
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
    cin>>cod1;
    setcod(cod1);
    cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
    cin>>nota1;
    setnota(nota1);*/
    est::leer();
    cout<<"INGRESE LA ACTIVIDAD QUE REALIZA"<<endl;
    cin>>act;
};
void estart::mostrar()
```

```
{
/* cout<<"NOMBRE: "<<getnom()<<endl;
cout<<"CODIGO: "<<getcod()<<endl;
cout<<"NOTA: "<<getnota()<<endl;*/
est::mostrar();
cout<<"PRACTICA: "<<act<<endl;
};
estart::~estart()
{
cout<<"destruyendo art"<<endl;
};
class estdep:public est
{
char dep[15];
char lugar[];
public:
estdep();
void setdep(char *ndep);
char *getdep();
void setlugar(char *nlugar);
char *getlugar();
void leer();
void mostrar();
~estdep();
};
estdep::estdep()
{
cout<<"CONSTRUYENDO ESTDEP"<<endl;
setdep("");
setlugar("");
};
void estdep::setdep(char *ndep)
{
strcpy(dep,ndep);
};
char *estdep::getdep()
{
return(dep);
};
void estdep::setlugar(char *nlugar)
{
strcpy(lugar,nlugar);
};
char *estdep::getlugar()
{
return(lugar);
};
};
```

```
void estdep::leer()
{
/*char nom1[15];
char cod1[5];
float nota1;
cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
gets(nom1);
setnom(nom1);
cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
cin>>cod1;
setcod(cod1);
cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
cin>>nota1;
setnota(nota1);*/
est::leer();
cout<<"INGRESE EL DEPORTE QUE PARCTICA"<<endl;
cin>>dep;
cout<<"INGRESE EL LUGAR OBTENIDO EN LOS ULTIMOS JUEGOS"<<endl;
cin>>lugar;
};
void estdep::mostrar()
{
/*cout<<"NOMBRE: "<<getnom()<<endl;
cout<<"CODIGO: "<<getcod()<<endl;
cout<<"NOTA: "<<getnota()<<endl;*/
est::mostrar();
cout<<"PRACTICA: "<<dep<<endl;
cout<<"OBTUVIERON EL "<<lugar<<" EN LOS ULTIMOS JUEGOS"<<endl;
};
estdep::~~estdep()
{
cout<<"destruyendo dep"<<endl;
};
class lista
{
char mat[15];
estart vea[30];
estdep ved[30];
est ves[30];
int ced,cea,ce;
public:
void setmat(char *mat);
char *getmat();
void setcea(int ncea);
int getcea();
void setced(int nced);
int getced();
```

```
void setce(int nce);
int getce();
void leer();
void mostrar();
void busdep();
void busart();
~lista();
};
void lista::setmat(char *nmat)
{
    strcpy(mat,nmat);
};
char *lista::getmat()
{
    return(mat);
};
void lista::setcea(int ncea)
{
    cea=ncea;
};
int lista::getcea()
{
    return(cea);
};
void lista::setced(int nced)
{
    ced=nced;
};
int lista::getced()
{
    return(ced);
};
void lista::setce(int nce)
{
    ce=nce;
};
int lista::getce()
{
    return(ce);
};
void lista::leer()
{
    int op1;
    char op;
    op='s';
    cout<<"INGRESE LA MATERIA"<<endl;
    cin>>mat;
```

```
while (op=='s')
{
    cout<<"ELIJA UNBA OPCION "<<endl;
    cout<<"1.- INGRESAR UN ESTUDIANTE "<<endl;
    cout<<"2.- INGRESAR UN ESTUDIANTE DEPORTISTA"<<endl;
    cout<<"3.- INGRESAR UN ESTUDIANTE ARTISTA"<<endl;
    cin>>op1;
    if (op1==1)
    {
        ves[ce].leer();
        ce++;
    };
    if (op1==2)
    {
        ved[ced].leer();
        ced++;
    };
    if (op1==3)
    {
        vea[cea].leer();
        cea++;
    };
    cout<<"DESEA INGRESAR OTRO ESTUDIANTE S/N"<<endl;
    cin>>op;
};
void lista::mostrar()
{
    int op;
    cout<<"ELIJA UNA OPCION"<<endl;
    cout<<"1.- MOSTRAR LOS ESTUDIANTES EXONERADOS"<<endl;
    cout<<"2.- MOSTRAR LOS ESTUDIANTES CON DERECHO A EXAMEN
PRINCIPAL"<<endl;
    cout<<"3.- MOSTRAR LOS ESTUDIANTES REPROBADOS"<<endl;
    cin>>op;
    if (op==1)
    {
        cout<<"MATERIA: "<<mat<<endl;
        cout<<"LOS ESTUDIANTES EXONERADOS SON:"<<endl<<endl;
        for (int i=0;i<cea;i++)
        {
            if((vea[i].getnota()+1)>=25)
            {
                vea[i].mostrar();
            };
        };
        for (int l=0;l<ced;l++)
```

```
{
    if((ved[i].getnota()+2)>=25)
    {
        ved[i].mostrar();
    };
};
for (int h=0;h<ce;h++)
{
    if(ves[i].getnota()>=25)
    {
        ves[i].mostrar();
    };
};
};

if (op==2)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES CON DERECHO A EXAMEN PRINCIPAL
SON:"<<endl<<endl;
    for (int i=0;i<cea;i++)
    {
        if((vea[i].getnota()+1>=4)&&(vea[i].getnota()+1<=23))
        {
            vea[i].mostrar();
        };
    };
    for (int j=01;j<ced;j++)
    {
        if((ved[i].getnota()+2>=4)&&(vea[i].getnota()+2<=24))
        {
            ved[i].mostrar();
        };
    };
    for (int k=0;k<ce;k++)
    {
        if(ves[i].getnota()<=23)
        {
            ves[i].mostrar();
        };
    }
};

if (op==3)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES REPROBADOS SON:"<<endl<<endl;
```

```
for (int i=0;i<cea;i++)
{
if((vea[i].getnota()+1<=3))
{
vea[i].mostrar();
};
};
for (int m=0;m<ced;m++)
{
if((ved[i].getnota()+2<=3))
{
ved[i].mostrar();
};
};
for (int p=0;p<ce;p++)
{
if(ves[i].getnota()<=3)
{
ves[i].mostrar();
};
};
};
void lista::busdep()
{
char dep1[15];
cout<<"INGRESE EL DEPORTE A BUSCAR"<<endl;
cin>>dep1;
clrscr();
cout<<"LOS ESTUDIANTES DEPORTIATAS QUE PRACTICAN EL DEPORTE
INGRESADO Y QUE TIENEN UN ACUNULADO MAYOR A 27 SON: "<<endl;
for (int i=0;i<getced();i++)
{
if (((strcmp(dep1,ved[i].getdep()))==0)&&(ved[i].getnota()+2>=27))
{
ved[i].mostrar();
}
}
};
void lista::busart()
{
cout<<"LOS ESTUDIANTES QUE ENTONAN GUITARRA SON: "<<endl;
for (int i=0;i<=getcea();i++)
{
if ((strcmp(vea[i].getact(),"guitarra")==0))
{
vea[i].mostrar();
}
```

```

};
};
};
lista::~lista()
{
    cout<<"destruyendo lista"<<endl;
};
void main ()
{
    clrscr();
    lista yo;
    yo.leer();
    yo.mostrar();
    yo.busdep();
    yo.busart();
    getch();
};

```

3.- ESTE PROGRAMA INGRESA LOS DATOS DE ESTUDIANTES, EST ARTISTAS Y EST DEPORTISTAS MUESTRA LOS EST CON SU CALIF Y SI INGRESAN A EXAMEN SE EXONERAN O REPRUEBAN MUESTRA LOS ART QUE TOCAN GUITARRA Y LOS DEP DE UN DEP DADO Y QUE TIENNE UN ACUMULADO MAYOR A 27 CON POLIMORFISMO DINAMICO

```

#include<iostream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>
class est
{
    char nom[25];
    char cod[5];
    float nota;
public:
    est();
    void setnom(char *nnom);
    char *getnom();
    void setcod(char *ncod);
    char *getcod();
    void setnota(float nnota);
    float getnota();
    void leer();
    void mostrar();
    ~est();
};
est::est()
{

```



```
cout<<"COSTRUYENDO EST"<<endl;
setnom("");
setcod("");
setnota(0);
};
void est::setnom(char *nnom)
{
    strcpy(nom,nnom);
};
char *est::getnom()
{
    return(nom);
};
void est::setcod(char *ncod)
{
    strcpy(cod,ncod);
};
char *est::getcod()
{
    return(cod);
};
void est::setnota(float nnota)
{
    nota=nnota;
};
float est::getnota()
{
    return(nota);
};
void est::leer()
{
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
    gets(nom);
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
    cin>>cod;
    cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
    cin>>nota;
};
void est::mostrar()
{
    cout<<"NOMBRE: "<<nom<<endl;
    cout<<"CODIGO: "<<cod<<endl;
    cout<<"NOTA: "<<nota<<endl;
};
est::~~est()
{
    cout<<"destruyendo est"<<endl;
```

```
};  
class estart:public est  
{  
    char act[15];  
public:  
    estart();  
    void setact(char *nact);  
    char *getact();  
    void leer();  
    void mostrar();  
    ~estart();  
};  
estart::estart()  
{  
    cout<<"CONSTRUYENDO ESTART"<<endl;  
    setact("");  
};  
void estart::setact(char *nact)  
{  
    strcpy(act,nact);  
};  
char *estart::getact()  
{  
    return(act);  
};  
void estart::leer()  
{  
    /*char nom1[15];  
    char cod1[5];  
    float nota1;  
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;  
    gets(nom1);  
    setnom(nom1);  
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;  
    cin>>cod1;  
    setcod(cod1);  
    cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;  
    cin>>nota1;  
    setnota(nota1);*/  
    est::leer();  
    cout<<"INGRESE LA ACTIVIDAD QUE REALIZA"<<endl;  
    cin>>act;  
};  
void estart::mostrar()  
{  
    /* cout<<"NOMBRE: "<<getnom()<<endl;  
    cout<<"CODIGO: "<<getcod()<<endl;
```

```
cout<<"NOTA: "<<getnota()<<endl;*/
est::mostrar();
cout<<"PRACTICA: "<<act<<endl;
};
estart::~estart()
{
    cout<<"destruyendo art"<<endl;
};
class estdep:public est
{
    char dep[15];
    char lugar[];
public:
    estdep();
    void setdep(char *ndep);
    char *getdep();
    void setlugar(char *nlugar);
    char *getlugar();
    void leer();
    void mostrar();
    ~estdep();
};
estdep::estdep()
{
    cout<<"CONSTRUYENDO ESTDEP"<<endl;
    setdep("");
    setlugar("");
};
void estdep::setdep(char *ndep)
{
    strcpy(dep,ndep);
};
char *estdep::getdep()
{
    return(dep);
};
void estdep::setlugar(char *nlugar)
{
    strcpy(lugar,nlugar);
};
char *estdep::getlugar()
{
    return(lugar);
};
void estdep::leer()
{
    /*char nom1[15];
```

```

char cod1[5];
float nota1;
cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
gets(nom1);
setnom(nom1);
cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
cin>>cod1;
setcod(cod1);
cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
cin>>nota1;
setnota(nota1);*/
est::leer();
cout<<"INGRESE EL DEPORTE QUE PARCTICA"<<endl;
cin>>dep;
cout<<"INGRESE EL LUGAR OBTENIDO EN LOS ULTIMOS JUEGOS"<<endl;
cin>>lugar;
};
void estdep::mostrar()
{
/*cout<<"NOMBRE: "<<getnom()<<endl;
cout<<"CODIGO: "<<getcod()<<endl;
cout<<"NOTA: "<<getnota()<<endl;*/
est::mostrar();
cout<<"PRACTICA: "<<dep<<endl;
cout<<"OBTUVIERON EL "<<lugar<<" EN LOS ULTIMOS JUEGOS"<<endl;
};
estdep::~~estdep()
{
cout<<"destruyendo dep"<<endl;
};
class lista
{
char mat[15];

estart *vea[30];
estdep *ved[30];
est *ves[30];

int ced,cea,ce;
public:
void setmat(char *mat);
char *getmat();
void setcea(int ncea);
int getcea();
void setced(int nced);
int getced();
void setce(int nce);

```

```
int getce();
void leer();
void mostrar();
void busdep();
void busart();
~lista();
};
void lista::setmat(char *nmat)
{
    strcpy(mat,nmat);
};
char *lista::getmat()
{
    return(mat);
};
void lista::setcea(int ncea)
{
    cea=ncea;
};
int lista::getcea()
{
    return(cea);
};
void lista::setced(int nced)
{
    ced=nced;
};
int lista::getced()
{
    return(ced);
};
void lista::setce(int nce)
{
    ce=nce;
};
int lista::getce()
{
    return(ce);
};
void lista::leer()
{
    int op1;
    char op;
    op='s';
    ce=0;
    cea=0;
    ced=0;
```

```
cout<<"INGRESE LA MATERIA"<<endl;
cin>>mat;
while ((op=='s')||(op=='S'))
{
    cout<<"ELIJA UNBA OPCION "<<endl;
    cout<<"1.- INGRESAR UN ESTUDIANTE "<<endl;
    cout<<"2.- INGRESAR UN ESTUDIANTE DEPORTISTA"<<endl;
    cout<<"3.- INGRESAR UN ESTUDIANTE ARTISTA"<<endl;
    cin>>op1;
    if (op1==1)
    {
        ves[ce]=new est;
        ves[ce]->leer();
        ce++;
    };
    if (op1==2)
    {
        ved[ced]=new estdep;
        ved[ced]->leer();
        ced++;
    };
    if (op1==3)
    {
        vea[cea]=new estart;
        vea[cea]->leer();
        cea++;
    };
    cout<<"DESEA INGRESAR OTRO ESTUDIANTE S/N"<<endl;
    cin>>op;
};
};
void lista::mostrar()
{
    /*aux me dice si existieron alumnos exonerados o no*/
    int op,aux,aux1,aux2;
    aux=0;
    aux1=0;
    aux2=0;
    cout<<"ELIJA UNA OPCION"<<endl;
    cout<<"1.- MOSTRAR LOS ESTUDIANTES EXONERADOS"<<endl;
    cout<<"2.- MOSTRAR LOS ESTUDIANTES CON DERECHO A EXAMEN
PRINCIPAL"<<endl;
    cout<<"3.- MOSTRAR LOS ESTUDIANTES REPROBADOS"<<endl;
    cin>>op;
    if (op==1)
    {
        cout<<"MATERIA: "<<mat<<endl;
```

```
cout<<"LOS ESTUDIANTES EXONERADOS SON:"<<endl<<endl;
for (int i=0;i<cea;i++)
{
    if((vea[i]->getnota()+1)>=25)
    {
        vea[i]->mostrar();
        aux=1;
    };
};
for (int l=0;l<ced;l++)
{
    if((ved[i]->getnota()+2)>=25)
    {
        ved[i]->mostrar();
        aux=1;
    };
};
for (int h=0;h<ce;h++)
{
    if(ves[i]->getnota())>=25)
    {
        ves[i]->mostrar();
        aux=1;
    };
};
if(aux==0)
{
    cout<<"NO HAY ESTUDIANTES EXONERADOS"<<endl<<endl<<endl<<endl;
};
};

/*aux1 me dice si existieron alumnos exonerados o no*/
if (op==2)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES CON DERECHO A EXAMEN PRINCIPAL
SON:"<<endl<<endl;
    for (int i=0;i<cea;i++)
    {
        if((vea[i]->getnota()+1>=4)&&(vea[i]->getnota()+1<=24))
        {
            vea[i]->mostrar();
            aux1=1;
        };
    };
    for (int j=01;j<ced;j++)
    {
```

```
if(((ved[j]->getnota()+2)>=4)&&((ved[j]->getnota()+2)<=24))
{
    ved[j]->mostrar();
    aux1=1;
};
};
for (int k=0;k<ce;k++)
{
    if((ves[i]->getnota())>=4)&&(ves[i]->getnota())<=24))
    {
        ves[i]->mostrar();
        aux1=1;
    };
}
if(aux1==0)
{
    cout<<"NO HAY ESTUDIANTES CON DERECHO A EXAMEN
PRINCIPAL"<<endl<<endl<<endl<<endl;
};

};
if (op==3)
{
    cout<<"MATERIA: "<<mat<<endl;
    cout<<"LOS ESTUDIANTES REPROBADOS SON:"<<endl<<endl;
    for (int i=0;i<cea;i++)
    {
        if((vea[i]->getnota()+1<=3))
        {
            vea[i]->mostrar();
            aux2=1;
        };
    };
    for (int m=0;m<ced;m++)
    {
        if((ved[i]->getnota()+2<=3))
        {
            ved[i]->mostrar();
            aux2=1;
        };
    };
    for (int p=0;p<ce;p++)
    {
        if(ves[i]->getnota())<=3)
        {
            ves[i]->mostrar();
            aux2=1;
        }
    }
}
```



```
};
};
if(aux2==0)
{
    cout<<"NO HAY ESTUDIANTES REPROBADOS"<<endl<<endl<<endl<<endl;
};
};
};
void lista::busdep()
{
    int aux;
    aux=0;
    char dep1[15];
    cout<<"INGRESE EL DEPORTE A BUSCAR"<<endl;
    cin>>dep1;
    clrscr();
    cout<<"LOS ESTUDIANTES DEPORTISTAS QUE PRACTICAN EL DEPORTE
INGRESADO Y QUE TIENEN UN ACUNULADO MAYOR IGUAL A 27 SON:
"<<endl;
    for (int i=0;i<getced();i++)
    {
        if (((strcmp(dep1,ved[i]->getdep()))==0)&&(ved[i]->getnota()+2>=27))
        {
            ved[i]->mostrar();
            aux=1;
        }
    }
    if (aux==0)
    {
        cout<<"NO EXISTEN ESTUDUANTES QUE CUMPLAN CON      TENER UN
ACUNULADO MAYOR IGUAL A 27 Y QUE PRACTIQUEN EL DEPORTE
INDICADO"<<
endl;
    };
};

};
void lista::busart()
{
    int l;
    l=0;
    cout<<"LOS ESTUDIANTES QUE ENTONAN GUITARRA SON: "<<endl;
    for (int i=0;i<=getcea();i++)
    {
        if ((strcmp(vea[i]->getact(),"guitarra")==0))
        {
            vea[i]->mostrar();
        }
    }
}
```

```

l=1;
};
}
if(l==0)
{
cout<<"NO HAY ESTUDIANTES QUE ENTONEN GUITARRA"<<endl;
};
};
lista::~~lista()
{
cout<<"destruyendo lista"<<endl;
for (int i=0;i<cea;i++)
{
delete(vea[i]);
};
for (int j=0;j<ced;j++)
{
delete(ved[j]);
};
for (int k=0;k<ce;k++)
{
delete(ves[k]);
};
};
void main ()
{
clrscr();
lista yo;
yo.leer();
yo.mostrar();
yo.busdep();
yo.busart();
getch();
};

```

4.- ESTE PROGRAMA MUESTRA LOS ESTUDIANTES INGRESADOS PUEDE CREAR VARIOS CURSOS E INTERCAMBIAR A LOS ESTUDIANTES DE UN CURSO A OTRO ADEMÁS DE ELIMINAR UN ESTUDIANTE DE UN CURSO ASÍ COMO ELIMINAR UN CURSO.

```

#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
class est
{
char nomest[30];

```

```
char cod[7];
float nota;
char act[20];
char dep[20];
int lug;
int tip;
public:
    est();
    void setnomest(char *nnomest);
    char *getnomest();
    void setcod(char *ncod);
    char *getcod();
    void setnota(float nnota);
    float getnota();
    virtual void setact(char *nact);
    virtual char *getact();
    virtual void setdep(char *ndep);
    virtual char *getdep();
    virtual void setlug(int nlug);
    virtual int getlug();
    void settip(int ntip);
    int gettip();
    virtual void leer();
    virtual void mostrar();
    ~est();
};
est::est()
{
    setnomest("");
    setcod("");
    setnota(0);
    settip(0);
};
void est::setnomest(char *nnomest)
{
    strcpy(nomest,nnomest);
};
char *est::getnomest()
{
    return(nomest);
};
void est::setcod(char *ncod)
{
    strcpy(cod,ncod);
};
char *est::getcod()
{
```

```
return(cod);
};
void est::setnota(float nnota)
{
    nota=nnota;
};
float est::getnota()
{
    return(nota);
};
void est::setact(char *nact)
{
    strcpy(act,nact);
};
char *est::getact()
{
    return(act);
};
void est::setdep(char *ndep)
{
    strcpy(dep,ndep);
};
char *est::getdep()
{
    return(dep);
};
void est::setlug(int nlug)
{
    lug=nlug;
};
int est::getlug()
{
    return(lug);
};
void est::settip(int ntip)
{
    tip=ntip;
};
int est::gettip()
{
    return(tip);
};
void est::leer()
{
    cout<<"INGRESE EL NOMBRE DEL ESTUDIANTE"<<endl;
    gets(nomest);
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE"<<endl;
```

```
gets(cod);
cout<<"INGRESE LA NOTA DEL ESTUDIANTE"<<endl;
cin>>nota;
};
void est::mostrar()
{
    cout<<"Nombre: "<<nomest<<endl;
    cout<<"Codigo: "<<cod<<endl;
    cout<<"Nota: "<<nota<<endl;
};
est::~est()
{

};
class art:public est
{
    char act[15];
public:
    art();
    void leer();
    void mostrar();
    ~art();
};
art::art()
{
    settip(1);
    setact("");
};
void art::leer()
{
    est::leer();
    cout<<"INGRESE EL ARTE QUE PRACTICA"<<endl;
    cin>>act;
    setact(act);
};
void art::mostrar()
{
    est::mostrar();
    cout<<"Practica: "<<getact()<<endl;
};
art::~~art()
{
};
class depo:public est
{
    char dep[15];
    int lug;
```

```
public:
depo();
void leer();
void mostrar();
~depo();
};
depo::depo()
{
    setdep("");
    setlug(0);
    settip(2);
};
void depo::leer()
{
    est::leer();
    cout<<"INGRESE EL DEPORTE QUE PRACTICA"<<endl;
    cin>>dep;
    setdep(dep);
    cout<<"INGRESE EL LUGAR OBTENIDO EN LOS ULTIMOS JUEGOS"<<endl;
    cin>>lug;
    setlug(lug);
};
void depo::mostrar()
{
    est::mostrar();
    cout<<"Practica: "<<getdep()<<endl;
    cout<<"Obtuvo el "<<getlug()<<" Lugar en los ultimos juegos"<<endl;
};
depo::~depo()
{

};
class curso
{
    char para[3];
    char mate[15];
    int nea,ned,nes,ntes;
    est *vest[100];
public:
    curso();
    void setpara(char *npara);
    char *getpara();
    void setmate(char *nmate);
    char *getmate();
    void setnea(int nnea);
    int getnea();
    void setned(int nned);
```

```

int getned();
void setnes(int nnes);
int getnes();
void setntes(int nntes);
int getntes();
void leer();/*lee los datos de un curso y llama a ingresar estudiantes si el usuario lo
requiere*/
void leerest();/*ingresa un listado de estudiantes*/
void mostrar();/*muestra los estudiantes ingresados*/
void estex();/*muestra los estudiantes exonerados*/
void estre();/*muestra los estudiantes reprobados*/
void estprin();/*muestra los estudiantes que ingresan a examen principal*/
virtual void mencurso();/*muestra el menu para mostrar estudiantes exonerados a
examen y reprobados*/
void estgui();/*muestra los estudiantes artistan que entonen guitarra*/
void estdep();/*muestra los estudiantes deportistas*/
void moscod();/*muestra un estudiante dado su codigo*/
void elimest();/*elimina un estudiante del curso*/
est *buscod(char cod[7]);/*busca un estudiante por codigo*/
void recove(int i);/*rrecorre el vector de estudiantes*/
void ingest();/*ingresa un nuevo estudiante al curso*/
est *buscod();/*busca el codigo de un estudiante y devuelve la posicion de memoria*/
void cambiar(est *ptr);/*asigna la direccion de memoria al otro curso*/
~curso();
};
curso::curso()
{
    setpara("");
    setmate("");
    setnea(0);
    setned(0);
    setnes(0);
    setntes(0);
};
void curso::setpara(char *npara)
{
    strcpy(para,npara);
};
char *curso::getpara()
{
    return(para);
};
void curso::setmate(char *nmate)
{
    strcpy(mate,nmate);
};
char *curso::getmate()

```

```
{
    return(mate);
};
void curso::setnea(int nnea)
{
    nea=nnea;
};
int curso::getnea()
{
    return(nea);
};
void curso::setned(int nned)
{
    ned=nned;
};
int curso::getned()
{
    return(ned);
};
void curso::setnes(int nnes)
{
    nes=nnes;
};
int curso::getnes()
{
    return(nes);
};
void curso::setntes(int nntes)
{
    ntes=nntes;
};
int curso::getntes()
{
    return(ntes);
};
void curso::leer()
{
    char op[2];
    cout<<"INGRESE EL PARALELO"<<endl;
    gets(para);
    cout<<"INGRESE LA MATERIA"<<endl;
    cin>>mate;
    cout<<"DESEA INGRESAR UN LISTADO DE ESTUDIANTES S/N"<<endl;
    cin>>op;
    if ((op[0]=='s')||(op[0]=='S'))
    {
        leorest();
    }
}
```



```
};  
};  
void curso::leerest()  
{  
    char op[2];  
    int op1;  
    do  
    {  
        cout<<"ELIJA UNA OPCION"<<endl;  
        cout<<"1.- INGRESAR UN ESTUDIANTE "<<endl;  
        cout<<"2.- INGRESAR UN ESTUDIANTE ARTISTA "<<endl;  
        cout<<"3.- INGRESAR UN ESTUDIANTE DEPORTISTA"<<endl;  
        cin>>op1;  
        if (op1==1)  
        {  
            vest[ntes]= new est();  
            vest[ntes]->leer();  
            ntes++;  
        };  
        if (op1==2)  
        {  
            vest[ntes]= new art();  
            vest[ntes]->leer();  
            ntes++;  
        };  
        if (op1==3)  
        {  
            vest[ntes]= new depo();  
            vest[ntes]->leer();  
            ntes++;  
        };  
        cout<<"DESEA INGRESAR OTRO ESTUDIANTE S/N"<<endl;  
        cin>>op;  
    }while ((op[0]!='s')||(op[0]!='S'));  
};  
void curso::mostrar()  
{  
    cout<<"Materia: "<<mate<<endl;  
    cout<<"Paralelo: "<<para<<endl;  
    for (int i=0;i<ntes;i++)  
    {  
        vest[i]->mostrar();  
    };  
};  
void curso::estex()  
{  
    cout <<"LOS ESTUDIANTES EXONERADOS SON: "<<endl;
```

```
for (int i=0;i<ntes;i++)
{
    if((vest[i]->gettip()==0)&&(vest[i]->getnota())>=24))
    {
        vest[i]->mostrar();
    };
    if((vest[i]->gettip()==1)&&(vest[i]->getnota()+1>=24))
    {
        vest[i]->mostrar();
    };
    if((vest[i]->gettip()==2)&&(vest[i]->getnota()+2>=24))
    {
        vest[i]->mostrar();
    };
};
void curso::estre()
{
    cout <<"LOS ESTUDIANTES REPROBADOS SON: "<<endl;
    for (int i=0;i<ntes;i++)
    {
        if((vest[i]->gettip()==0)&&(vest[i]->getnota())<4))
        {
            vest[i]->mostrar();
        };
        if((vest[i]->gettip()==1)&&(vest[i]->getnota()+1<4))
        {
            vest[i]->mostrar();
        };
        if((vest[i]->gettip()==2)&&(vest[i]->getnota()+2<4))
        {
            vest[i]->mostrar();
        };
    };
};
void curso::estprin()
{
    cout <<"LOS ESTUDIANTES CON DERECHO A EXAMEN SON: "<<endl;
    for (int i=0;i<ntes;i++)
    {
        if((vest[i]->gettip()==0)&&(vest[i]->getnota())>=4)&&(vest[i]->getnota())<=23))
        {
            vest[i]->mostrar();
        };
        if((vest[i]->gettip()==1)&&(vest[i]->getnota()+1>=4)&&(vest[i]->getnota()+1<=23))
        {
            vest[i]->mostrar();
        };
    };
};
```

```
};  
if((vest[i]->gettip()==2)&&(vest[i]->getnota()+2>=4)&&(vest[i]->getnota()+2<=23))  
{  
    vest[i]->mostrar();  
};  
};  
};  
void curso::estgui()  
{  
    for (int i=0;i<ntes;i++)  
    {  
        if ((vest[i]->gettip()==1)&&((strcmp(vest[i]-  
>getact(),"guitarra")==0)||((strcmp(vest[i]->getact(),"GUITARRA")==0))  
        {  
            vest[i]->mostrar();  
        };  
    };  
};  
void curso::estdep()  
{  
    for (int j=0;j<=ntes;j++)  
    {  
        if (vest[j]->gettip()==2)  
        {  
            vest[j]->mostrar();  
        };  
    };  
};  
void curso::moscod()  
{  
    int a=0;  
    int i=0;  
    char cod[7];  
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE A BUSCAR"<<endl;  
    cin>>cod;  
    while((a==0)&&(i<ntes))  
    {  
        if ((strcmp(vest[i]->getcod(),cod))==0)  
        {  
            vest[i]->mostrar();  
            a=1;  
        };  
        i++;  
    };  
    if (a==0)  
    {  
        cout<<"EL CODIGO INGRESADO NO EXISTE"<<endl;
```

```
};  
};  
void curso::mencurso()  
{  
    int op;  
    while(op!=10)  
    {  
        cout<<"ELIJA UNA OPCION "<<endl;  
        cout<<"1.- MOSTRAR ESTUDIANTES EXONERADOS "<<endl;  
        cout<<"2.- MOSTRAR ESTUDIANTES REPROBADOS "<<endl;  
        cout<<"3.- MOSTRAR ESTUDIANTES CON DERECHO A EXAMEN "<<endl;  
        cout<<"4.- MOSTRAR LOS ESTUDIANTES DEPORTISTAS"<<endl;  
        cout<<"5.- MOSTRAR LOS ESTUDIANTES QUE ENTONEN GUITARRA"<<endl;  
        cout<<"6.- MOSTRAR UN ESTUDIANTE DADO SU CODIGO"<<endl;  
        cout<<"7.- ELIMINAR UN ESTUDIANTE DADO SU CODIGO"<<endl;  
        cout<<"8.- INGRESAR UN NUEVO ESTUDIANTE "<<endl;  
        cout<<"9.- MOSTRAR LOS ESTUDIANTES DEL CURSO"<<endl;  
        cout<<"10.- SALIR DE ESTE MENU "<<endl;  
        cin>>op;  
        if (op==1)  
        {  
            estex();  
        };  
        if (op==2)  
        {  
            estre();  
        };  
        if (op==3)  
        {  
            estprin();  
        };  
        if (op==4)  
        {  
            estdep();  
        };  
        if (op==5)  
        {  
            estgui();  
        };  
        if (op==6)  
        {  
            moscod();  
        };  
        if (op==7)  
        {  
            elimest();  
            mostrar();  
        }  
    }  
}
```

```
};  
if (op==8)  
{  
    ingest();  
    mostrar();  
};  
if (op==9)  
{  
    mostrar();  
};  
};  
/*PREGUNTA como hacer para retornar al menu de escuela*/  
};  
est *curso::buscod()  
{  
    est *ptr;  
    int a=0;  
    int i=0;  
    char cod[7];  
    cout<<"INGRESE EL CODIGO DEL ESTUDIANTE A CAMBIAR"<<endl;  
    cin>>cod;  
    while((a==0)&&(i<ntes))  
    {  
        if ((strcmp(vest[i]->getcod(),cod))==0)  
        {  
            ptr=vest[i];  
            a=1;  
            recove(i);  
        };  
        i++;  
        if (a==0)  
        {  
            ptr=NULL;  
        };  
    };  
    return(ptr);  
};  
void curso::cambiar(est *ptr)  
{  
    vest[ntes]=ptr;  
    ntes++;  
};  
void curso::elimet()  
{  
    char cod[7];  
    int a=0;  
    int i=0;
```

```
cout<<"INGRESE EL CODIGO DEL ESTUDIANTE A ELIMINAR"<<endl;
gets(cod);
while ((a==0)&&(i<ntes))
{
    if ((strcmp(vest[i]->getcod(),cod))==0)
    {
        a=1;
        delete(vest[i]);
        recove(i);
    };
    i++;
};
if (a==0)
{
    cout<<"EL ESTUDIANTE NO SE ELIMINO"<<endl;
};
};
void curso::recove(int i)
{
    for (int j=i;j<ntes;j++)
    {
        vest[j]=vest[j+1];
    };
    j--;
    setntes(j);
};
void curso::ingest()
{
    if (ntes<100)
    {
        leerest();
    }
    else
    cout<<"EL CURSO ESTA LLENO"<<endl;
};
curso::~~curso()
{
    for(int i=0;i<ntes;i++)
    {
        delete(vest[i]);
    };
};
class esc
{
    int ncur;
    char nomesc[20];
    curso *vcur[20];
};
```

```

public:
    esc();
    void setnomesc(char *nnomesc);
    char *getnomesc();
    void setncur(int n_cur);
    int getncur();
    void leer();/*ingresa los datos de la escuela e ingresa curso si se requiere*/
    void leercur();/*ingresa cursos*/
    void mostrar();/*muestra los datos de la escuela y cursos*/
    void menuesc();/*muestra el menu de los metodos de escuela*/
    int buspar(char par[2]);/*busca un paralelo y devuelve su posicion*/
    void recovcu(int a);/*recorre vector de cursos*/
    void elimcur();/*elimina un curso*/
    void metcur();/*permite interactuar con los metodos de curso*/
    void ingunest();/*ingresa un estdianete a un curso*/
    void eliunest();/*elimina un estudiante e un curso*/
    void camcur();/*ingresa los cursos que intervienen en el cambio*/
    ~esc();
};

esc::esc()
{
    setnomesc("");
    setncur(0);
};

void esc::setnomesc(char *nnomesc)
{
    strcpy(nomesc,nnomesc);
};

char *esc::getnomesc()
{
    return(nomesc);
};

void esc::setncur(int n_cur)
{
    ncur=n_cur;
};

int esc::getncur()
{
    return(ncur);
};

void esc::leer()
{
    char op[2];
    cout<<"INGRESE EL NOMBRE DE LA ESCUELA" <<endl;
    gets(nomesc);
    cout<<"DESEA INGRESAR UN CURSO O LISTADO DE CURSOS S/N"<<endl;
    cin>>op;
}

```

```

if ((op[0]=='s')||(op[0]=='S'))
{
    leercur();
};
};
void esc::leercur()
{
    char op[2];
    do
    {
        vcur[ncur]=new curso();
        vcur[ncur]->leer();
        ncur++;
        cout<<"DESEA INGRESAR OTRO CURSO S/N"<<endl;
        cin>>op;
    }while ((op[0]=='s')||(op[0]=='S')&&(ncur<20));
};
void esc::mostrar()
{
    cout<<"Escuela: "<<nomesc<<endl;
    for (int i=0;i<ncur;i++)
    {
        vcur[i]->mostrar();
    };
};
int esc::buscpar(char par[2])
{
    int pos,a,i;
    a=0;
    i=0;
    while ((a==0)&&(i<ncur))
    {
        if ((strcmp(vcur[i]->getpara(),par))==0)
        {
            pos=i;
            a=1;
        };
        i++;
    };
    if (a==0)
    {
        pos=-1;
    };
    return(pos);
};
void esc::recovcu(int a)
{

```



```
for(int i=a;i<ncur;i++)
{
    vcur[i]=vcur[i+1];
};
ncur--;
};
void esc::elimcur()
{
    int a;
    char para[2];
    cout<<"INGRESE EL PARALELO A ELIMINAR"<<endl;
    cin>>para;
    a=buscpar(para);
    if(a>=0)
    {
        delete(vcur[a]);
        recovcu(a);
    }
    else
    {
        cout<<"EL CURSO INGRESADO NO EXISTE"<<endl;
    };
};
void esc::metcur()
{
    int a;
    char para[2];
    cout<<"INGRESE EL CURSO CON EL QUE DESEA TRABAJAR"<<endl;
    gets(para);
    a=buscpar(para);
    if (a>=0)
    {
        vcur[a]->mencurso();
    };
    if (a<0)
    {
        cout<<"EL CURSO INGRESASO NO EXISTE"<<endl;
    };
};
void esc::eliunest()
{
    char para[2];
    int a;
    cout<<"INGRESE EL CURSO EN EL QUE SE ENCUENTRA EL ESTUDIANTE A ELIMINAR"<<endl;
    cin>>para;
    a=buscpar(para);
```

```
if (a>=0)
{
    vcur[a]->elimest();
};
};
void esc::ingunest()
{
    char para[2];
    int a;
    cout<<"INGRESE EL CURSO EN EL QUE DESEA INGRESAR AL
ESTUDIANTE"<<endl;
    cin>>para;
    a=buscpar(para);
    if (a>=0)
    {
        vcur[a]->ingest();
    };
};
void esc::menuesc()
{
    int op;
    while(op!=10)
    {
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- INGRESAR UNLISTADO DE CURSOS"<<endl;
        cout<<"2.- INGRESAR UN NUEVO CURSO"<<endl;
        cout<<"3.- ELIMINAR UN CURSO"<<endl;
        cout<<"4.- INGRESAR UN NUEVO ESTUDIANTE A UN CURSO"<<endl;
        cout<<"5.- ELIMINAR UN ESTUDIANTE"<<endl;
        cout<<"6.- CAMBIAR UN ESTUDIANTE DE PARALELO"<<endl;
        cout<<"7.- TRABAJAR CON UN CURSO"<<endl;
        cout<<"8.- MOSTRAR TODOS LOS CURSOS"<<endl;
        cout<<"10.- SALIR DE ESTE MENU"<<endl;
        cin>>op;
        if (op==1)
        {
            leer();
        };
        if (op==2)
        {
            leercur();
        };
        if (op==3)
        {
            elimcur();
        };
        if (op==4)
```

```
{
    ingunest();
};
if (op==5)
{
    eliunest();
};
if (op==6)
{
    camcur();
};
if (op==7)
{
    metcur();
};
if (op==8)
{
    mostrar();
};
};
};
void esc::camcur()
{
    char para1[2],para2[2];
    int a,b;
    est *ptr;
    cout<<"INGRESE EL CURSO DE ORIGEN DEL ESTUDIANTE"<<endl;
    cin>>para1;
    a=buscpar(para1);
    if (a>=0)
    {
        cout<<"INGRESE EL CURSO DE DESTINO DEL ESTUDIANTE"<<endl;
        cin>>para2;
        b=buscpar(para2);
        if (b>=0)
        {
            ptr=vcur[a]->buscod();
            if (ptr!=NULL )
            {
                vcur[b]->cambiar(ptr);
            };
        };
    };
};
esc::~~esc()
{
    for (int i=0;i<ncur;i++)
```

```
{
    delete(vcur[i]);
};
};
void main()
{
    /*clrscr(); */
    esc yo;
    yo.menuesc();
    getch();
};
```

5.- ESTE PROGRAMA INGRESA LOS DATOS DE UN AEROPUERTO ASI COMO EUN LISTADO DE AVIONES EN FUNCIONAMIENTO PARA LUEGO MEDIANTE ALGUNAS OPCIONES PERMITIR AL USUARIO CREAR UN LISTADO DE AVIONES EN MANTENIMIENTO, HABILITAR UN AVION, MOSTRAR LOS AVIONES CON CAPACIDAD DE CARGA MAYOR A 100 METROS CUBICOS, MOSTRAR LOS DATOS DE AVIONES EN FUNCIONAMIENTO Y EN MANTENIMIENTO.

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
#include<conio.h>
class avion
{
    char marca[15],modelo[15],iden[7];
    float hv,hr;
    int tip;
public:
    avion();
    void setmarca(char *nmarca);
    char *getmarca();
    void setmodelo(char *nmodelo);
    char *getmodelo();
    void setiden(char *niden);
    char *getiden();
    void sethv(float nhv);
    float gethv();
    void sethr(float nhr);
    float gethr();
    void settip(int ntip);
    int gettip();
    virtual void leer();
    virtual void mostrar();
    virtual float getcar2();
    ~avion();
```

```
};  
avion::avion()  
{  
    setmarca("");  
    setmodelo("");  
    setiden("");  
};  
void avion::setmarca(char *nmarca)  
{  
    strcpy(marca,nmarca);  
};  
char *avion::getmarca()  
{  
    return(marca);  
};  
void avion::setmodelo(char *nmodelo)  
{  
    strcpy(modelo,nmodelo);  
};  
char *avion::getmodelo()  
{  
    return(modelo);  
};  
void avion::setiden(char *niden)  
{  
    strcpy(iden,niden);  
};  
char *avion::getiden()  
{  
    return(iden);  
};  
void avion::sethv(float nhv)  
{  
    hv=nhv;  
};  
float avion::gethv()  
{  
    return(hv);  
};  
void avion::sethr(float nhr)  
{  
    hr=nhr;  
};  
float avion::gethr()  
{  
    return(hr);  
};
```

```

void avion::settip(int ntip)
{
    tip=ntip;
};
int avion::gettip()
{
    return(tip);
};
void avion::leer()
{
    cout<<"INGRESE LA MARCA DEL AVION"<<endl;
    gets(marca);
    cout<<"INGRESE EL MODELO DEL AVION"<<endl;
    gets(modelo);
    cout<<"INGRESE EL IDENTIFICADOR DEL AVION"<<endl;
    gets(iden);
    cout<<"INGRESE LAS HORAS DE VUELO"<<endl;
    cin>>hv;
    cout<<"INGRESE LAS HORAS DE VUELO ANTES DEL
MANTENIMIENTO"<<endl;
    cin>>hr;
};
void avion::mostrar()
{
    cout<<"Identificador: "<<iden<<endl;
    cout<<"Marca: "<<marca<<endl;
    cout<<"Modelo: "<<modelo<<endl;
    cout<<"Horas de Vuelo: "<<hv<<endl;
    cout<<"Horas Recomendadas: "<<hr<<endl;
};
float avion::getcar2()
{
    return(0);
};
avion::~~avion()
{
};
class acar:public avion
{
    float car1,car2;
public:
    acar();
    void setcar1(float ncar1);
    float getcar1();
    float getcar2();
    void leer();
};

```

```
void mostrar();
~acar();
};
acar::acar()
{
    setcar1(0);

    settip(1);
};
float acar::getcar2()
{
    return(car2);
};
void acar::setcar1(float ncar1)
{
    car1=ncar1;
};
float acar::getcar1()
{
    return(car1);
};

void acar::leer()
{
    avion::leer();
    cout<<"INGRESE LA CAPACIDAD DE CARGA EN TONELADAS"<<endl;
    cin>>car1;
    cout<<"INGRESE LA CAPACIDAD DE CARGA EN METROS CUBICOS"<<endl;
    cin>>car2;

};
void acar::mostrar()
{
    avion::mostrar();
    cout<<"CAPACIDAD DE CARGA EN TONELADAS: "<<car1<<endl;
    cout<<"CAPACIDAD DE CARGA EN METROS CUBICOS: "<<car2<<endl<<endl;
};
acar::~acar()
{

};

class apas:public avion
{
    int npas;
public:
    apas();
    void setnpas(int nnpas);
```

```
int getnpas();
void leer();
void mostrar();
~apas();
};
apas::apas()
{
    setnpas(0);
    settip(2);
};
void apas::setnpas(int nnpas)
{
    npas=nnpas;
};
int apas::getnpas()
{
    return(npas);
};
void apas::leer()
{
    avion::leer();
    cout<<"INGRESE EL NUMERO DE PASAJEROS "<<endl;
    cin>>npas;
};
void apas::mostrar()
{
    avion::mostrar();
    cout<<"CAPACIDAD DE PASAJEROS: "<<npas<<endl<<endl;
};
apas::~~apas()
{
};
class aero
{
    int naf,nam;
    char nom[20];
    avion *vaf[100];
    avion *vam[100];
public:
    aero();
    void setnaf(int nnaf);
    int getnaf();
    void setnam(int nnam);
    int getnam();
    void setnom(char *nnom);
    char *getnom();
};
```



```

void leer();/*ingresa los datos del aeropuerto y pregunta si desea ingresar un listado de
aviones*/
void mostrar();/*muestra los datos del aeropuerto y aviones*/
void mostraraf();/*muestra los aviones del listado de funcionamiento*/
void mostraram();/*muestra los aviones del listado de mantenimiento*/
void leeravi();/*ingresa aviones en funcionamiento*/
void recoaf(int a);/*rrecorre un vector de aviones en funcionamiento*/
void recoam(int a);/*rrecorre un vector de aviones en mantenimiento*/
void avman();/*crea un listado de aviones en mantenimiento*/
void carmay();/*muestra los aviones de carga con capacidad mayor a 100*/
void abili();/*pasa un avion de mantenimiento a funcionamiento*/
avion *buscav(char iden[7]);/*busca el avion si lo encuentra devuelve un puntero*/
void elim();/*elimina un avion definitivamente de la lista*/
void inguna();/*ingresa un nuevo avion al listado*/
void menu();/*muestra un menu de opciones */
~aero();
};
aero::aero()
{
    setnom("");
    setnaf(0);
    setnam(0);
};
void aero::setnaf(int nnaf)
{
    naf=nnaf;
};
int aero::getnaf()
{
    return(naf);
};
void aero::setnam(int nnam)
{
    nam=nnam;
};
int aero::getnam()
{
    return(nam);
};
void aero::setnom(char *nnom)
{
    strcpy(nom,nnom);
};
char *aero::getnom()
{
    return(nom);
};

```

```
void aero::mostraraf()
{
    cout<<"LOS AVIONES EN FUNCIONAMIENTO SON:"<<endl<<endl;
    for (int i=0;i<getnaf();i++)
    {
        vaf[i]->mostrar();
    };
};
void aero::mostraram()
{
    cout<<"LOS AVIONES EN MANTENIMIENTO SON:"<<endl<<endl;
    for (int i=0;i<getnam();i++)
    {
        vam[i]->mostrar();
    };
};
void aero::leer()
{
    char op[2];
    cout<<"INGRESE EL NOMBRE DEL AEROPUERTO"<<endl;
    gets(nom);
    cout<<"DESEA INGRESAR UN LISTADO DE AVIONES S/N"<<endl;
    cin>>op;
    if ((op[0]=='s')||(op[0]=='S'))
    {
        leeravi();
    };
};
void aero::leeravi()
{
    char op[2];
    int op1;
    do
    {
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- INGRESAR UN AVION DE CARGA"<<endl;
        cout<<"2.- INGRESAR UN AVION DE PASAJEROS"<<endl;
        cin>>op1;
        if (op1==1)
        {
            vaf[naf]=new acar();
            vaf[naf]->leer();
            naf++;
        };
        if (op1==2)
        {
            vaf[naf]=new apas();
        }
    }
    while (op1!=1 && op1!=2);
};
```

```
vaf[naf]->leer();
naf++;
};
setnaf(naf);
cout<<"DESEA INGRESAR OTRO AVION S/N"<<endl;
cin>>op;
}while ((op[0]!='S')||(op[0]!='s'));
};
void aero::mostrar()
{
cout<<"Aeropuerto: "<<nom<<endl;
if (naf>=0)
{
mostraraf();
};
if (nam>0)
{
mostraram();
};
};
void aero::recoaf(int a)
{
for (int i=a;i<naf-1;i++)
{
vaf[i]=vaf[i+1];
};
naf--;
};
void aero::recoam(int a)
{
for (int i=a;i<nam-1;i++)
{
vam[i]=vam[i+1];
};
nam--;
};
avion *aero::buscav(char *iden)
{
int i=0;
int a=0;
avion *ptr1;
while((a==0)&&(i<nam))
{
if ((strcmp(vam[i]->getiden(),iden))==0)
{
a=1;
ptr1=(vam[i]);
}
```

```

    recoam(i);
};
i++;
};
if (a==0)
{
    cout<<"EL IDENTIFICADOR INGRESADO NO CONSTA EN
MANTENIMIENTO"<<endl;
    ptr1=NULL;
};
return(ptr1);
};
void aero::avman()
{
    char op[2];
    int i,a,j;
    i=0;
    cout<<"DESEA CREAR UN LISTADO DE LOS AVIONES QUE NECESITAN
MANTENIMIENTO S/N"<<endl;
    cin>>op;
    if ((op[0]=='S')||(op[0]=='s'))
    {
        while(i<naf)
        {
            if ((vaf[i]->gethv())>=(vaf[i]->gethr()))
            {
                vam[nam]=vaf[i];
                nam++;
                recoaf(i);
            };
            i++;
        };
    };
};
void aero::carmay()
{
    cout<<"LOS AVIONES DE CARGA CON CAPACIDAD MAYOR A 100 SON:
"<<endl;
    for (int i=0;i<naf;i++)
    {
        if ((vaf[i]->gettip()==1)&&(vaf[i]->getcar2())>=100))
        {
            vaf[i]->mostrar();
        };
    };
};
};

```

```
void aero::abili()
{
    char op[2],iden[7];
    avion *ptr;
    cout<<"DESEA HABILITAR UN AVION S/N"<<endl;
    cin>>op;
    if ((op[0]=='s')||(op[0]=='S'))
    {
        cout<<"INGRESE EL IDENTIFICADOR DEL AVION A HABILITAR"<<endl;
        cin>>iden;
        ptr=buscav(iden);
        if (ptr!=NULL)
        {
            vaf[naf]=ptr;
            naf++;
        };
    };
};
void aero::elim()
{
    int a=0;
    int i=0;
    char op[2],iden[7];
    cout<<"DESEA ELIMINAR UN AVION DEL LISTADO S/N"<<endl;
    cin>>op;
    if ((op[0]=='S')||(op[0]=='s'))
    {
        cout<<"INGRESE EL IDENTIFICADOR DEL AVION A ELIMINAR"<<endl;
        gets(iden);
        while((a==0)&&(i<naf))
        {
            if ((strcmp(vaf[i]->getiden(),iden))==0)
            {
                a=1;
                delete(vaf[i]);
                recoaf(i);
            };
            i++;
        };
        if (a==0)
        {
            cout<<"EL IDENTIFICADOR INGRESADO NO ESTA EN  
FUNCIONAMIENTO"<<endl;
        };
    };
};
void aero::inguna()
```

```
{
char op[2];
cout<<"DESEA INGRESAR UN NUEVO AVION A LA LISTA S/N"<<endl;
cin>>op;
if ((op[0]=='S')||(op[0]=='s')&&(naf<100))
{
    leeravi();
};
};
void aero::menu()
{
    int op;
    while(op!=10)
    {
        cout<<"ELIJA UNA OPCION"<<endl;
        cout<<"1.- INGRESAR DATOS DEL AEROPUERTO Y LISTADO DE AVIONES"<<endl;
        cout<<"2.- INGRESAR UN AVION O UN LISTADO DE AVIONES"<<endl;
        cout<<"3.- CREAR UN LISTADO DE AVIONES QUE NECESITAN MANTENIMIENTO"<<endl;
        cout<<"4.- MOSTRAR DATOS DE AVIONES Y DEL AEROPUERTO"<<endl;
        cout<<"5.- HABILITAR UN AVION"<<endl;
        cout<<"6.- ELIMINAR UN AVION"<<endl;
        cout<<"7.- MOSTRAR DATOS DE AVIONES EN FUNCIONAMIENTO"<<endl;
        cout<<"8.- MOSTRAR DATOS DE AVIONES EN MANTENIMIENTO"<<endl;
        cout<<"9.- AVIONES DE CARGA CON CAPACIDAD > O = ALOS 100 METROS CUBICOS"<<endl;
        cin>>op;
        if (op==1)
        {
            leer();
        };
        if (op==2)
        {
            inguna();
        };
        if (op==3)
        {
            avman();
        };
        if (op==4)
        {
            mostrar();
        };
        if (op==5)
        {
            abili();
        };
    }
}
```

```

};
if (op==6)
{
    elim();
};
if (op==7)
{
    mostraraf();
};
if (op==8)
{
    mostraram();
};
if (op==9)
{
    carmay();
};
};
};
void aero::~aero()
{
    for (int i=0;i<naf;i++)
    {
        delete(vaf[i]);
    };
    for (int j=0;j<nam;j++)
    {
        delete(vam[i]);
    };
};
void main()
{
    clrscr();
    aero yo;
    yo.menu();
    getch();
};

```

6.- ESTE PROGRAMA INGRESA UN LISTADO DE REFRIGERADORES, LICUADORAS, DE LOS REFRIGERADORES SE TIENE NUMERO DE PUERTAS Y CAPACIDAD EN METROS CUBICOS, Y DE LAS LICUADORAS SE TIENE EL NUMERO DE VELOCIDADES, EL PROGRAMA DEBE PERMITIR: 1.-INGRESAR LOS DATOS DEL ALMACEN Y LISTADO DE PRODUCTOS. 2.- INGRESAR UN NUEVO A LOS EXISTENTES. 3.- PASAR UN PRODUCTO A LA LISTA DE VENDIDOS DADO EL CODIGO. 4.- MOSTRAR TODOS LOS PRODUCTOS EXISTENTES Y VENDIDOS. 5.- MOSTRAR PRODUCTOS VENDIDOS LICUADORAS O REFRIGERADORAS. 6.- REINGRESAR UN PRODUCTO VENDIDO AL ALMACEN. 7.- ELIMINAR UN ELECTRODOMESTICO DADO SU

IDENTIFICADOR. 8.- MOSTRAR REFRIGERADORES DE 2 PUERTAS COLOR BLANCO.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class elec
{
    char marca[20];
    char model[20];
    char color[15];
    char cod[7];
    int af,tip,np;
public:
    elec();
    virtual void setnp(int nnp);
    virtual int getnp();
    void setmarca(char *nmarca);
    char *getmarca();
    void setmodel(char *nmodel);
    char *getmodel();
    void setcolor(char *ncolor);
    char *getcolor();
    void setcod(char *ncod);
    char *getcod();
    virtual void settip(int ntip);
    virtual int gettip();
    void setaf(int naf);
    int getaf();
    virtual void leer();
    virtual void mostrar();
    ~elec();
};
elec::elec()
{
    setmarca("");
    setmodel("");
    setcolor("");
    setcod("");
    setaf(0);
};
void elec::setnp(int nnp)
{
    np=nnp;
};
int elec::getnp()
```



```
{
    return(np);
};
void elec::setmarca(char *nmarca)
{
    strcpy(marca,nmarca);
};
char *elec::getmarca()
{
    return(marca);
};
void elec::setmodel(char *nmodel)
{
    strcpy(model,nmodel);
};
char *elec::getmodel()
{
    return(model);
};
void elec::setcolor(char *ncolor)
{
    strcpy(color,ncolor);
};
char *elec::getcolor()
{
    return(color);
};
void elec::settip(int ntip)
{
    tip=ntip;
};
int elec::gettip()
{
    return(tip);
};
void elec::setaf(int naf)
{
    af=naf;
};
int elec::getaf()
{
    return(af);
};
void elec::setcod(char *ncod)
{
    strcpy(cod,ncod);
};
};
```

```

char *elec::getcod()
{
    return(cod);
};
void elec::leer()
{
    cout<<"INGRESE EL CODIGO DEL ELECTRODOMESTICO"<<endl;
    gets(cod);
    cout<<"INGRESE LA MARCA DEL ELECTRODOMESTICO"<<endl;
    gets(marca);
    cout<<"INGRESE EL MODELO DEL ELECTRODOMESTICO"<<endl;
    gets(model);
    cout<<"INGRESE EL COLOR DEL ELECTRODOMESTICO"<<endl;
    gets(color);
    cout<<"INGRESE EL AÑO DE FABRICACION DEL
ELECTRODOMESTICO"<<endl;
    cin>>af;
};
void elec::mostrar()
{
    cout<<"Marca: "<<marca<<endl;
    cout<<"Modelo: "<<model<<endl;
    cout<<"Color: "<<color<<endl;
    cout<<"Año de Fabricación: "<<af<<endl;
};
elec::~elec()
{

};
class ref:public elec
{
    float cap;
    int np;
public:
    ref();
    void setcap(int ncap);
    int getcap();
    void leer();
    void mostrar();
    ~ref();
};
ref::ref()
{
    setcap(0);
    setnp(0);
    settip(1);
};

```

```

void ref::setcap(int ncap)
{
    cap=ncap;
};
int ref::getcap()
{
    return(cap);
};
void ref::leer()
{
    elec::leer();
    cout<<"INGRESE LA CAPACIDAD DEL REFRIGERADOR EN METROS
CUBICOS"<<endl;
    cin>>cap;
    cout<<"INGRESE EL NUMERO DE PUERTAS DEL REFRIGERADOR"<<endl;
    cin>>np;
    setnp(np);
};
void ref::mostrar()
{
    elec::mostrar();
    cout<<"Capacidad: "<<cap<<" Metros C bicos"<<endl;
    cout<<"Puertas: "<<np<<endl;
};
ref::~ref()
{

};
class licua:public elec
{
    int nv;
public:
    licua();
    void setnv(int nnv);
    int getnv();
    void leer();
    void mostrar();
    ~licua();
};
licua::licua()
{
    setnv(0);
    settip(2);
};
void licua::setnv(int nnv)
{
    nv=nnv;

```

```

};
int licua::getnv()
{
    return(nv);
};
void licua::leer()
{
    elec::leer();
    cout<<"INGRESE EL NUMERO DE VELOCIDADES"<<endl;
    cin>>nv;
};
void licua::mostrar()
{
    elec::mostrar();
    cout<<"Numero de Velocidades"<<nv<<endl;
};
licua::~licua()
{

};
class almacen
{
    char nom[25];
    char direc[25];
    int npr,npv;
    elec *vpr[100];
    elec *vpv[100];
public:
    almacen();
    void setnom(char *nnom);
    char *getnom();
    void setdirec(char *ndirec);
    char *getdirec();
    void setnpr(int nnpr);
    int getnpr();
    void setnpv(int nnpv);
    int getnpv();
    void leer();
    void mostrar();/*muestra todos los productos exixtentes y vendidos*/
    void leerpro();/*ingresa un listado de productos*/
    void prven();/*muestra los productos vendidos dado su codigo*/
    void mosref();/*muestra las refrigeradoras vendidas*/
    void moslic();/*muestra las licuadoras vendidas*/
    void mostip();/*muestra los productos vendidos licuadoras y refrigeradoras*/
    void proven();/*pregunta al usuario si desea ingresar un producto vendido*/
    void recovepr(int i);/*recorre el vector de productos exixtentes*/
    void devol();/*reingresa un producto vendido*/

```

```
void recovpv(int i);/*recorre el vector de productos vendidos*/
void menu();/*muestra el menu con las diferentes opciones del sistema*/
int buscodex(char cod[7]);/*busca el codigo y devuelve la posicion en el vector
existentes*/
int buscodpv(char cod[7]);/*busca el codigo y devuelve la posicion en el vector
vendidos*/
void eliminar();/*muestra el menu para eliminar existentes o vendidos*/
void elimven();/*elimina un producto de los vendidos*/
void elimex();/*elimina un producto de los existentes*/
void refblan();/*muestra refrigeradores de 2 puertas color blanco*/
~almacen();
};
almacen::almacen()
{
    setnom("");
    setdirec("");
    setnpr(0);
    setnpv(0);
};
void almacen::setnom(char *nnom)
{
    strcpy(nom,nnom);
};
char *almacen::getnom()
{
    return(nom);
};
void almacen::setdirec(char *ndirec)
{
    strcpy(direc,ndirec);
};
char *almacen::getdirec()
{
    return(direc);
};
void almacen::setnpr(int nnpr)
{
    npr=nnpr;
};
int almacen::getnpr()
{
    return(npr);
};
void almacen::setnpv(int npv)
{
    npv=npv;
};
```

```
int almacen::getnpv()
{
    return(npv);
};
void almacen::leer()
{
    char op[2];
    cout<<"INGRESE EL NOMBRE DEL ALMACEN"<<endl;
    gets (nom);
    cout<<"INGRES LA DIRECCION"<<endl;
    gets(direc);
    cout<<"Desea ingresar un listado de productos S/N"<<endl;
    cin>>op;
    if ((op[0]=='s')||(op[0]=='S'))
    {
        leerpro();
    };
};
void almacen::mostrar()
{
    cout<<"PRODUCTOS EXIXTENTES"<<endl;
    for (int i=0;i<npr;i++)
    {
        vpr[i]->mostrar();
    };
    cout<<"PRODUCTOS VENDIDOS"<<endl;
    for (int j=0;j<npv;j++)
    {
        vpv[j]->mostrar();
    };
};
void almacen::leerpro()
{
    char op[2];
    int op1;
    do
    {
        cout<<"1.- INGRESAR UN REFRIGERADOR"<<endl;
        cout<<"2.- INGRESAR UNA LICUADORA"<<endl;
        cin>>op1;
        if (op1==1)
        {
            vpr[npr]=new ref();
            vpr[npr]->leer();
            npr++;
        };
        if (op1==2)
```

```
{
    vpr[npr]=new licua();
    vpr[npr]->leer();
    npr++;
};
cout<<"DESEA INGRESAR OTRO ELECTRODOMESTICO S/N"<<endl;
cin>>op;
}while((op[0]=='S')||(op[0]=='s'));
};
void almacen::proven()
{
    char op[2];
    cout<<"DESEA INGRESAR UN PRODUCTO VENDIDO S/N"<<endl;
    cin>>op;
    if ((op[0]=='s')||(op[0]=='S')&&(npv<100))
    {
        prven();
    };
};
void almacen::eliminar()
{
    int op;
    cout<<"ELIJA UNA OPCION"<<endl;
    cout<<"1.- ELIMINAR UN PRODUCTO DE LOS EXISTENTES"<<endl;
    cout<<"2.- ELIMINAR UN PRODUCTO DE LOS VENDIDOS"<<endl;
    cin>>op;
    if (op==1)
    {
        elimex();
    };
    if (op==2)
    {
        elimven();
    };
};
void almacen::elimven()
{
    int b;
    char cod[7];
    cout<<"INGRESE EL CODIGO DEL PRODUCTO A ELIMINAR"<<endl;
    gets(cod);
    b=(buscodpv(cod));
    if(b>=0)
    {
        delete(vpv[b]);
        recovpv(b);
    }
}
```

```
else
    cout<<"NO SE ELIMINO EL PRODUCTO"<<endl;
};
void almacen::elimex()
{
    int b;
    char cod[7];
    cout<<"INGRESE EL CODIGO DEL PRODUCTO A ELIMINAR"<<endl;
    gets(cod);
    b=buscodex(cod);
    if(b>=0)
    {
        delete(vpr[b]);
        recovepr(b);
    }
    else
        cout<<"NO SE ELIMINO EL PRODUCTO"<<endl;
};
int almacen::buscodpv(char cod[7])
{
    int a,i,b;
    a=0;
    i=0;
    b=0;
    while((a==0)&&(i<npv))
    {
        if((strcmp(vpv[i]->getcod(),cod))==0)
        {
            a=1;
            b=i;
        }
        else
            i++;
    };
    if (a==0)
    {
        b=-1;
    };
    return(b);
};
int almacen::buscodex(char cod[7])
{
    int a,i,b;
    a=0;
    i=0;
    b=0;
    while((a==0)&&(i<npr))
```



```
{
if((strcmp(vpr[i]->getcod(),cod))==0)
{
a=1;
b=i;
}
else
i++;
};
if (a==0)
{
b=-1;
};
return(b);
};
void almacen::prven()
{
char cod[7];
int i=0;
int a=0;
cout<<"INGRESE EL CODIGO DEL PRODUCTO VENDIDO"<<endl;
gets(cod);
while((a==0)&&(i<npr))
{
if ((strcmp(vpr[i]->getcod(),cod))==0)
{
a=1;
vpv[npv]=vpr[i];
recovepr(i);
npv++;
}
else
i++;
};
if (a==0)
{
cout<<"EL CODIGO INGRESADO NO CONSTA EN LA LISTA DEL
ALMACEN"<<endl;
};
};
void almacen::recovpv(int i)
{
for (int j=i;j<npv;j++)
{
vpv[j]=vpv[j+1];
};
npv--;
```

```
};  
void almacen::recovepr(int i)  
{  
    for (int j=i;j<npr;j++)  
    {  
        vpr[j]=vpr[j+1];  
    };  
    npr--;  
};  
void almacen::mosref()  
{  
    cout<<"LAS REFRIGERADORAS VENDIDAS SON"<<endl;  
    for (int i=0;i<npr;i++)  
    {  
        if (vpv[i]->gettip()==1)  
        {  
            vpv[i]->mostrar();  
        };  
    };  
};  
void almacen::moslic()  
{  
    cout<<"LAS LICUADORAS VENDIDAS SON"<<endl;  
    for (int i=0;i<npr;i++)  
    {  
        if (vpv[i]->gettip()==2)  
        {  
            vpv[i]->mostrar();  
        };  
    };  
};  
void almacen::mostip()  
{  
    int op;  
    cout<<"1.- MOSTRAR LICUADORAS VENDIDAS"<<endl;  
    cout<<"1.- MOSTRAR REFRIGERADORAS VENDIDAS"<<endl;  
    cin>>op;  
    if (op==1)  
    {  
        moslic();  
    };  
    if (op==2)  
    {  
        mosref();  
    };  
};  
void almacen::devol()
```

```

{
char cod[7];
int i,a;
i=0;
a=0;
cout<<"INGRESE EL CODIGO DEL PRODUCTO A REINGRESAR"<<endl;
gets(cod);
while((a==0)&&(i<npv))
{
if ((strcmp(vpv[i]->getcod(),cod))==0)
{
a=1;
vpr[npr]=vpv[i];
recovpv(i);
npr++;
}
else
i++;
};
if (a==0)
{
cout<<"El CODIGO INGRESADO NO CONSTA EN LA LISTA DE PRODUCTOS
VENDIDOS"<<endl;
};
};
void almacen::refblan()
{
cout<<"LOS REFRIGERADORES BLANCOS DE DOS PUERTAS EXISTENTES
SON: "<<endl;
for(int i=0;i<npr;i++)
{
if((vpr[i]->gettip()==1)&&(vpr[i]->getnp()==2)&&((strcmp(vpr[i]-
>getcolor(),"blanco"))==0)||((strcmp(vpr[i]->getcolor(),"BLANCO"))==0))
{
vpr[i]->mostrar();
};
};
};
void almacen::menu()
{
int op;
while(op!=9)
{
cout<<"ELIJA UNA OPCION"<<endl;
cout<<"1.- INGRESAR LOS DATOS DEL ALMACEN Y UN LISTADO DE
PRODUCTOS"<<endl;
cout<<"2.- INGRESAR UN NUEVO A LOS EXIXTENTES"<<endl;

```

```
cout<<"3.- PASAR UN PRODUCTO A LA LISTA DE VENDIDOS DADO EL
CODIGO"<<endl;
cout<<"4.- MOSTRAR TODOS LOS PRODUCTOS EXISTENTES Y
VENDIDOS"<<endl;
cout<<"5.- MOSTRAR PRODUCTOS VENDIDOS LICUADORAS O
REFRIGERADORAS"<<endl;
cout<<"6.- REINGRESAR UN PRODUCTO VENDIDO AL ALMACEN"<<endl;
cout<<"7.- ELIMINAR UN ELECTRODOMESTICO DADO SU
IDENTIFICADOR"<<endl;
cout<<"8.- MOSTRAR REFRIGERADORES DE 2 PUERTAS COLOR
BLANCO"<<endl;
cin>>op;
if(op==1)
{
    leer();
};
if (op==2)
{
    leerpro();
};
if(op==3)
{
    prven();
};
if (op==4)
{
    mostrar();
};
if (op==5)
{
    mostip();
};
if (op==6)
{
    devol();
};
if (op==7)
{
    eliminar();
};
if(op==8)
{
    refblan();
};
};
almacen::~~almacen()
```

```
{  
for (int i=0;i<npr;i++)  
{  
    delete(vpr[i]);  
};  
for (int j=0;j<npr;j++)  
{  
    delete(vpr[j]);  
};  
};  
void main()  
{  
    clrscr();  
    almacen yo;  
    yo.menu();  
    getch();  
};
```