



**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACIÓN**

**Ciclo
II**

**Programación de Algoritmos
Guía de Laboratorio No. 5
Estructura condicional (if)**

I. RESULTADOS DE APRENDIZAJE

Que el estudiante:

- Aplique correctamente las estructuras de decisión (**if** simple y doble) dentro del código fuente de un código fuente de C++
- Diseñe menús de varias opciones en un programa de C++, utilizando instrucciones **if-else**

II. INTRODUCCIÓN

La toma de decisiones en C++

Con el lenguaje C++ puede tomar decisiones gracias a 2 estructuras:

A. Estructura **if**

B. Estructura **switch**

Este tipo de sentencias le permiten alterar la secuencia normal de pasos de su programa en base a la redacción de condiciones específicas.

A continuación veremos sintaxis y uso de la estructura if.

Estructura if

A la estructura **if** se le define una **condición** para así bifurcar la secuencia del programa en una de dos alternativas posibles, de la siguiente manera:

- a) Si se cumple la condición ejecutará las **Sentencias1**.
- b) Si no se cumplen la condición, ejecutará las **Sentencias2**. Esta sección es opcional.

La sintaxis de la estructura if se define así:

```
if(condición) {  
    //Sentencias si cumple condición  
}  
else {  
    //Sentencias si no cumple condición  
}
```

Cada uno de los dos bloques de sentencias: el bloque `if` (si se cumple condición) y el bloque `else` (si no se cumple condición), tiene una pareja de llaves de apertura-cierre `{ }` para indicar el inicio y final de las sentencias a realizar en cada caso dentro de la estructura condicional `if`.

Los valores lógicos que utiliza C son los valores de tipo **bool**: `true` (que es verdadero y C++ lo maneja como el número entero 1) y `false` (significa falso o un valor entero de 0). Al evaluar a una Condición en un `if`:

- a) si esta devuelve `true`, se ejecuta el bloque de sentencias 1.
- b) Si devuelve `false`, se ejecuta el bloque de sentencias 2. Este bloque `else { }` es opcional, puede ser ignorado por el programador si lo cree así conveniente.

Las condiciones a evaluar por una estructura **if** pueden ser de 2 tipos:

- a) *Simple* y b) *Compleja*.

En una **condición simple**, se compara una variable con un valor o sino con otra variable, gracias a los **operadores relacionales** de C++, los cuales son los siguientes:

<code>></code>	mayor que	<code><=</code>	menor o igual que
<code><</code>	menor que	<code>==</code>	igual que
<code>>=</code>	mayor o igual que	<code>!=</code>	Diferente a, No igual que

Algunos ejemplos de condiciones simples utilizadas dentro de un IF son los siguientes:

`if (a>5)` `if (Z<=12)` `if (q != 0)`

Una **condición compleja** permite evaluar 2 o más condiciones simples como sola una evaluación lógica, gracias al uso de los **operadores lógicos**. Los operadores lógicos de C son los siguientes:

Operador lógico	Descripción de funcionamiento
<code>&&</code>	(Y lógico: AND/Y) El operador <code>&&</code> equivale al "AND" o "Y"; devuelve 1 (true) sólo si las 2 expresiones evaluadas son verdaderas (true) o distintas de cero. En caso contrario devuelve cero (falso). Si la primera expresión evaluada es falsa (false), C no evalúa a la segunda y devuelve false.
<code> </code>	(O lógico: OR/O) El operador <code> </code> equivale al "OR" u "O inclusivo"; devuelve 1 (true) si cualquiera de las expresiones evaluadas es verdadera (true) o distinta de cero, en caso contrario devuelve 0 (false). Si la primera expresión evaluada es verdadera 1, la segunda no se evalúa.
<code>!</code>	No lógico o negación: (signo de admiración NOT / negación de). El operador <code>!</code> es equivalente al "NOT", o "NO", y devuelve "true" sólo si la expresión evaluada es falsa (false) o cero, en caso contrario devuelve "true".

La sintaxis de uso cada uno de estos operadores lógicos es:

`(<expresión1>&&<expresión2>)` `(<expresión1>||<expresión2>)` `!<Expresión>`

Ejemplos de condiciones complejas utilizando los operadores lógicos dentro de un if:

```
if ((a>5)&&(Z<=12))
if ((Z>5)|| (w<1))
if ((renta<=120)&&(AFP>45))
if ((a>0)|| (b<=0)&&(n != a))
```

Anidamiento de estructuras de decisión

Se anidan colocando un if en el interior de otro if, de manera jerarquica

DiseñoPseudocódigo	Código en C
Si (condición1) entonces Si (condición2) entonces fin si fin si	if (condicion1){ ... if (condicion2){ } ... }

III. MATERIALES Y EQUIPO

No.	Requerimiento	Cantidad
1	Compilador de C++	1
2	Practica #5 de Programacion de Algoritmos	1
3	Memoria USB	1

IV. PROCEDIMIENTO

1. Cree una carpeta llamada **PAL_Guia5deCARNET**, para que guarde los archivos cpp del procedimiento a continuación y también de los ejercicios solicitados en el análisis de resultados.

Recuerde reemplazar CARNET por su respectivo número de carnet.

2. Ahora desarrolle cada uno de los programas de C++ a continuación, teniendo cuidado de aplicar las reglas siguientes en cada código fuente:

- Cuide el uso de mayúsculas y minúsculas en al redactar palabras reservadas y los nombres de variables y/o funciones, **ya que lenguaje C es sensible a mayúsculas y minúsculas.**
- Realice la compilación respectiva, con el fin de generar el archivo .exe de aplicación.

- Desarrolle varias pruebas de ejecución, ingresando diversos datos en cada prueba, para construir una idea general del funcionamiento del programa
- Cuando encuentre una nota de **IMPORTANTE**, al final de un código, realice sin falta las pruebas solicitadas ahí, para deducir las respuestas apropiadas

1. Código cpp del Programa: Ejemplo1.cpp

¿Cómo saber si un número ingresado por el usuario es positivo, negativo o cero?

```
#include<iostream>
using namespace std;
#include<conio.h>
//programa para saber si un numero dado por usuario es
//positivo, negativo o cero
main(){
    //declaracion de variables
    int num; //crea variable entera
    cout<<"Programa para evaluar a un numero"<<endl<<endl;
    cout<<"Ingrese un numero entero cualquiera: "; cin>>num;
    if(num>0)
    cout<<"\n Numero " <<num<<" es Positivo";
    else
        if(num<0)
            cout<<"\n Numero " <<num<<" es Negativo";
        else
            cout<<"\n Numero " <<num<<" es Cero";

    getch();
} //fin main
```

2. Código cpp del Programa: Ejemplo2.cpp

Pida al usuario la edad y el sexo, para que la computadora le indique si ya puede jubilarse. Tome en cuenta que un hombre se puede jubilar cuando tenga 60 años o más, en cambio, una mujer mayor se jubilará si tiene más de 54 años.

```
#include<iostream>
using namespace std;
#include<conio.h>
```

```

//program. para saber si una persona puede o no jubilarse
main(){
//declaracion de variables
int edad;
char sexo;

cout<<"Ingrese su edad (en años): "; cin>>edad;
cout<<"Presione letra f(si es mujer) o m (Si es hombre)";
sexo=getche();

if(!(sexo=='m' || sexo=='f'))
cout<<"\n\nSexo indeterminado, fin del programa";
else
if(sexo=='f' && edad>54)
cout<<"\n\nUsted puede jubilarse"<<endl;
else
if(sexo=='m' && edad>=60)
cout<<"\n\nUsted puede jubilarse"<<endl;
else
cout<<"\n\nUsted aun puede jubilarse porque tiene menos de 54 años"
(char) 164<< "os"<<endl;

getch();
} //fin main

```

IMPORTANTE:

- El texto `\xa4` resaltado en el `cout` que pide la edad, muestra el código ASCII de la letra `ñ`, el cual es `164`. Pero este valor se brinda en su equivalente hexadecimal (`A4`).
- También puede realizarse una conversión explícita a `char` del código `ascii 164` del carácter `ñ`. Con la sentencia `(char) 164` se retorna el carácter imprimible del código `ascii 164`.

3. Nombre del código fuente: Ejemplo3.cpp

Cálculo del área de una figura geométrica

Pregunte al usuario cuál de los siguientes tipos de área desea calcular:

1. rectángulo 2. triangulo 3. circulo,

Luego se pide los datos apropiados de la figura seleccionada y muestra el valor del área final.

Debe avisar a usuario cuando no seleccione alguna de las opciones proporcionadas

```

/*programa de seleccion de area (rectangulo, triangulo y circulo)
y su calculo respectivo */
#include<iostream>
using namespace std;
#include<conio.h> //porque usara funcion getch
#include<math.h> //usa funcion pow
#include<stdlib.h> //usa funcion system

#define PI 3.141592 //declara valor de constante PI

main(){
//declaracion de variables a utilizar
//>>> entradas
char TipArea;//tipo de area a calcular
float lar,anc; //Largo y ancho de rectangulo
float bas,alt; //base y altura de triangulo
float radio; //radio de circulo
//>>>salidas
float area=0; //resultado calculo area seleccionada

//define formato de impresion para valores decimales a mostrar con cout
cout.setf(ios::fixed);//formateo numero coma punto fijo
cout.setf(ios::showpoint); //muestre punto decimal
cout.precision(2);//define solo 2 cifras decimales a mostrar
cout<<"*****"<<endl;
cout<<"* PROGRAMA PARA CALCULOS DE AREAS *"<<endl;
cout<<"*****";

cout<<"\n\nSelecione letra de opcion de una de "<<endl;
cout<<"las siguientes figuras para mostrarle su Area:"<<endl;
cout<<endl<<"\t(a). Rectangulo\t(b). Triangulo\t(c). Circulo"<<endl;

TipArea=getch();//captura UNA tecla presionada por usuario

if(TipArea=='a'){
    system("cls"); //Limpia contenido pantalla de ejecucion

    //pide datos necesarios para determinar area del rectangulo

```

```

    cout<<"Escribe largo del rectangulo: "; cin>>lar;
    cout<<endl<<"y ahora digita ancho del mismo: "; cin>>anc;
    area=lar*anc;
cout<<"El Area del rectangulo es de "<<area<<" metros cuadrados"<<endl;
}
else{
    if(TipArea=='b'){
        system("cls");
        //pide datos necesarios para calcular area do triangulo
        cout<<"\tEscribe base y luego largo del triangulo:"<<endl;
        cin>>bas>>alt;
        area=(bas*alt)/2;
        cout<<"Area del triangulo es "<<area<<" metros cuadrados"<<endl;
    }else{
        if (TipArea=='c'){
            system("cls") ;
            //pide los datos necesarios para calcular area del circulo
            cout<<endl<<"Escribe el radio del circulo:\t";
            cin>>radio;
            //La funcion pow calcula la potencia de un numero
            area=PI*pow(radio,2); //invoca funcion pow
            cout<<"\nArea del Circulo es de "<<area<<" metros cuadrados"<<endl;
        }else{
            cout<<endl<<endl<<"\t>>ERROR: No me diste una opcion valida<<";
        }//fin tercer if-else
    }//fin segundo if-else
}//fin primer if-else
getch(); //genera pausa en ejecucion de la aplicacion

}//fin funcion main

```

IMPORTANTE:

- Para cualquiera de las figuras geométricas, ingrese valores tanto enteros como decimales (con 2 o 3 cifras decimales) en cualquiera de las pruebas y evalúe la presentación del resultado de salida (cifras decimales)

4. Nombre de código fuente Ejemplo4.cpp

Determine el sueldo mensual normal y neto de un Vendedor, tomando en cuenta que este tiene un sueldo base más una comisión por sus ventas realizadas en el mes. Y del sueldo total anterior, se le descuentan la renta (10%) y el ISSS (3.07%).

El porcentaje de *comisión mensual* de la venta mensual se indica en la siguiente tabla:

<i>Ventas Mensuales(\$)</i>	<i>Comisión</i>
Menores de \$60.5	5%
Entre \$60.5 y \$80	10%
Más de \$80 y menos de \$200	18%
De \$200 en adelante	25%

```
#include<iostream>
using namespace std;

#include<stdlib.h> //usa funcion system

main(){
//entradas
float sueldobase=0;
float venta=0;
//proceso
float comision=0;
//salidas
float sueldonormal, sueldoneto;

system("color 4a");

cout<<"\t\tPrograma calculo sueldo neto de un Vendedor";
cout<<"\n\nDigite sueldo base del empleado: $ ";
cin>>sueldobase;
cout<<"\nIngrese total de ventas: $ ";
cin>>venta;

//determina comision por ventas realizadas
if(venta<60.5) comision=0.05*venta;
else if(venta>=60.5 && venta<=80) comision=0.10*venta;
    else if(venta>80 && venta<200) comision=0.18*venta;
        else comision=0.25*venta;
```

```

system("color f2");
//calculo sueldo mensual
sueldonormal=sueldobase+comision;
cout<<"\n\nSueldo normal $ "<<sueldonormal;

//aplica descuentos de ley (renta 10%, ISSS 3.07%)
sueldoneto=(1-0.10-0.0307)*sueldonormal;
cout<<"y Sueldo neto $"<<sueldoneto<<"\n\n";

system("pause"); //genera pausa en ejecucion de la aplicacion

} //fin funcion main

```

IMPORTANTE:

- La cadena de texto "4a" usada en la llamada de `system("color 4a")` representa un Byte escrito en hexadecimal (de 2 dígitos). En las 2 llamadas de `system` que cambian el color, modifique valores "4a" y "f2" por otra pareja de dígitos hexadecimales diferentes, incluso pruebe después con dígitos iguales (Ej.: "bb" o "55") ¿Qué sucede con los resultados del programa?

V. ANÁLISIS DE RESULTADOS

Lea cuidadosamente a cada uno de los problemas a continuación, para luego, elaborar la solución apropiada en un código fuente de C++. En cada solución, debe utilizar la menor cantidad de decisiones.

PROBLEMA 1:

Solicite a usuario el nombre de una materia que ha cursado y la nota final obtenida en la misma.

Luego, se mostrara el equivalente de esta nota ingresada en el Standard Grading Scale System, utilizado en el sistema educativo de EEUU.

PROBLEMA 2:

Permita a un usuario retornarle en pantalla al resultado de elevar cualquier valor entero a otro valor entero. El valor entero de la base y del exponente pueden ser positivos, negativos o hasta ceros.

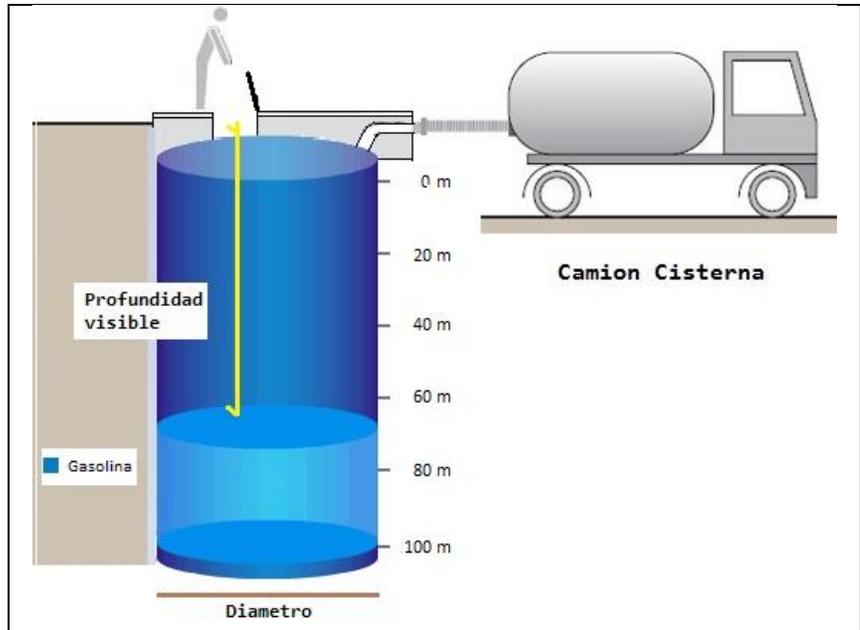
PROBLEMA 3:

Tal como se muestra en el diagrama, en una estación de servicio, la medida del volumen de gasolina almacenado en su tanque cilíndrico subterráneo (de 100 metros de profundidad) se obtiene al medir la distancia desde la parte superior del tanque a la superficie de la gasolina (la profundidad visible).

El transporte del combustible almacenado en el tanque se hace por medio de camiones cisterna, cada uno con capacidad de 100 pies cúbicos.

Solicite los datos necesarios al encargado de la gasolinera, para determinar:

- a) El volumen **actual** (en metros cúbicos) de combustible dentro del tanque.
- b) Cantidad de camiones cisterna que serian necesarios para lograr transportar el volumen calculado en el literal anterior hacia otra gasolinera.



Recuerde que debe validar el ingreso de cada dato solicitado.

En caso de un error en el ingreso de alguno de los datos solicitados, el programa muestra el aviso de error apropiado y finaliza su ejecución, sin mostrar resultados.

PROBLEMA 4:

Ayude a un estudiante de electrónica básica a determinar el resultado de la compuerta lógica EXOR y de una NAND.

Los valores lógicos que requiere ingresarse o mostrarse son las letras (v, f) en minúsculas.

PROBLEMA 5:

Solicite a usuario el valor de la hora actual, en horas, minutos y segundos; así como, si es antes mediano (am) o después mediano (pm). Luego, mostrarle la hora que sería después de transcurrir 5 segundos exactamente.

Observe algunos ejemplos de horas ingresadas y su correspondiente valor luego de 10 segundos.

Hora ingresada	Hora (luego de 10 segundos)
4 am.: 8 min.: 25 seg.	4 am.: 8 min.: 30 seg.
5 pm.: 45 min.: 57 seg.	5 pm.: 46 min.: 2 seg.
11 pm.: 59 min.: 56 seg.	12 am.: 0 min.: 1 seg.
7 am.: 59 min.: 54 seg.	8 am.: 0 min.: 4 seg.

PROBLEMA 6:

Permita recibir del usuario a los 3 componentes numéricos que definen a una fecha (día, mes y año), para que finalmente le muestre en pantalla, pero bajo el siguiente formato:

(día) de (nombre de mes) del (año)

Tome en cuenta las siguientes aclaraciones:

- Limitar el año valido solamente al rango entre 1900 a 2050
- Ignorar la regla de los años bisiestos.

Ejemplos:

- + Si usuario ingresa día, mes y año: 26, 12, 2013, respectivamente, el programa le mostrará la fecha así:
26 de Diciembre del 2013
- + Si usuario ingresa día, mes y año: 31, 2, 2007, respectivamente, el programa mostrará el mensaje:
Error en la fecha ingresada, día no existe en el mes ingresado.
- + Si usuario ingresa día, mes y año: 18, 56, 1978, respectivamente, el programa mostrará el mensaje:
Error en la fecha ingresada, numero de mes valido solamente entre 1 a 12
- + Si usuario ingresa día, mes y año: 29, 2, 2052, respectivamente, el programa mostrará el mensaje:
Error: años validos solamente entre 1900 a 2050

PROBLEMA 7:

Cree un programa que lee los tres lados de un triángulo, para que luego evalué a que tipo pertenece, según estas normas:

Si los tres lados son iguales, se trata de un triángulo equilátero

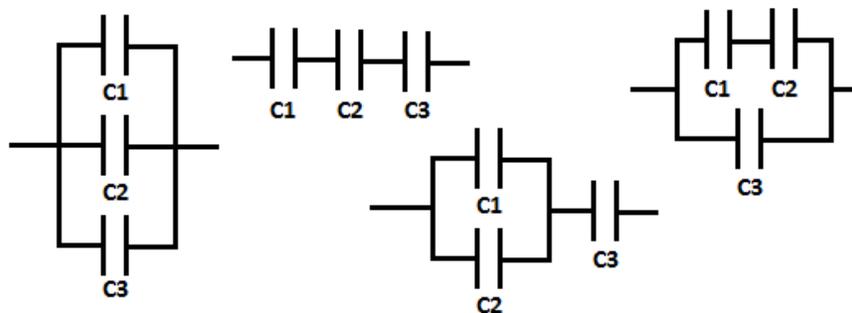
Si los tres lados no son iguales entre sí, es un triángulo escaleno

Si dos de sus lados son iguales es un triángulo isósceles

PROBLEMA 8:

Ayude a un estudiante de electrónica a calcular el capacitor equivalente de una serie de 3 capacitores diferentes (medidos en picofaradios).

Considere que todos los dispositivos pueden estar conectados en serie o en paralelo o puede haber mezclas de ambos tipos de conexiones, tal como se muestra en el siguiente diagrama:



Guía de Laboratorio No. 5: RÚBRICA DE EVALUACIÓN

Actividad a evaluar: ANÁLISIS DE RESULTADOS

Formar grupos entre 3 a 5 estudiantes, llenar esta hoja de evaluación y entregarla a su docente.

Su instructor seleccionará 3 problemas de la DISCUSION DE RESULTADOS, para ser resueltos apropiadamente por el grupo.

Lista de Integrantes:

CARNET 1	CARNET 2	CARNET 3	CARNET 4	CARNET 5

Problemas a resolver:

Criterio a evaluar	¿Prob 1?	¿Prob 2?	¿Prob 3?	PROM	Puntaje
(15%) Define las variables de entrada y salida esperadas					
(20%) Se redactan las expresiones lógicas apropiadas en las sentencias (if-else) Se usa la menor cantidad de estructuras de control (if)					
(25%) Se cumplen cada una de las restricciones descritas en el problema Dialogo claro y explicito con usuario final.					
(20%) Código fuente se compila y obtiene a c/u de los resultados solicitados.					
(20%) Documentacion interna apropiada, por cada línea y bloque de líneas					
Nota:					

Importante: Se descontara **50%** en cualquier solución que use switch, ciclos, creación de funciones u otros contenidos aun no vistos en planificación del curso.