



Estructuras de control para la programación: Estructuras de control selectiva (Simple- Compuesta-Múltiple)

FUNDAMENTOS DE PROGRAMACIÓN



Teoría

Reconoce la sintaxis de la estructura de control selectivas: Simple, Compuesta y Múltiple, en el lenguaje de programación



RECUERDA: Estructura de Control Selectiva Simple

Programa en C++: Semana3Programa1.cpp

Ejemplo:

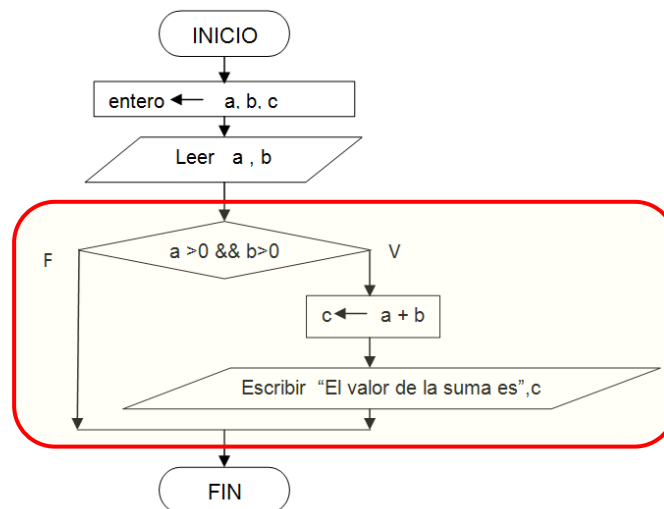
Sumar dos números enteros **si** son positivos (mayor a cero)

Solución:

Variables:

a: valor de primer numero
b: valor de segundo numero
c: valor de la suma

Diagrama de Flujo:



```
1 #include<iostream>
2 using namespace std;
3
4 int main( )
5 {
6     setlocale(LC_CTYPE,"Spanish");
7
8     int a, b;
9     int c;
10
11     cout<<"Ingrese valor de primer número:\t\t";
12     cin>>a;
13     cout<<"\n";
14
15     cout<<"Ingrese valor de segundo número:\t";
16     cin>>b;
17     cout<<"\n";
18
19     if( a>0 && b>0)
20     {
21         c = a + b;
22         cout<<"La suma es:\t\t\t" << c << "\n";
23     }
24
25     return 0;
26 }
27
```

CASO I

3

7

if(3>0 && 7>0)

c=3+7=10

10

CASO II

-3

7

if(-3>0 && 7>0)



RECUERDA: Estructura de Control Selectiva Compuesta

Ejemplo:

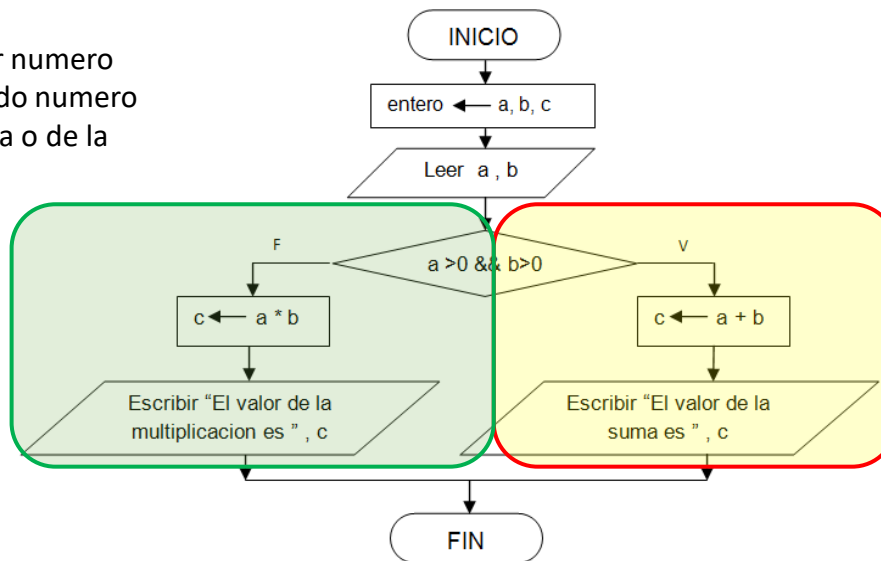
Sumar dos números enteros **si** son positivos (mayor a cero) **sino** multiplicarlos.

Solución:

Variables

a: valor de primer numero
b: valor de segundo numero
c: valor de la suma o de la multiplicación

Diagrama de Flujo



Programa en C++: Semana3Programa2.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main( )
5 {
6     setlocale(LC_CTYPE, "Spanish");
7
8     int a, b;
9     int c;
10
11     cout<<"Ingrese valor de primer número:\t\t";
12     cin>>a;
13     cout<<"\n";
14
15     cout<<"Ingrese valor de segundo número:\t";
16     cin>>b;
17     cout<<"\n";
18
19     if( a>0 && b>0)
20     {
21         c = a + b;
22         cout<<"La suma es:\t\t\t" << c << "\n";
23     }
24     else
25     {
26         c = a * b;
27         cout<<"La multiplicación es:\t\t\t" << c << "\n";
28     }
29
30     return 0;
31 }
```

CASO I

4

5

if(4>0 && 5>0)

c=4+5=9

9

CASO II

4

-5

if(4>0 && -5>0)

else

c=4* -5= -20

-20



RECUERDA: Estructura de Control Selectiva Múltiple

Ejemplo:

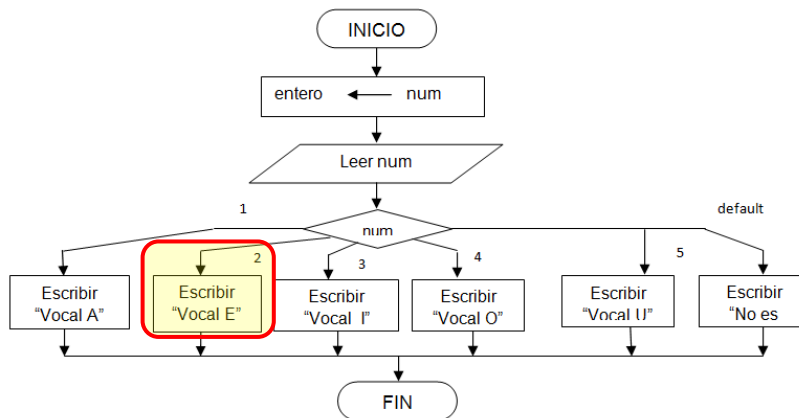
Ingresa un número y mostrar que vocal es.

Solución:

Variables

num: valor de numero que representa a una vocal

Diagrama de Flujo



Programa en C++: Semana3Programa3.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main( )
5 {
6     setlocale(LC_CTYPE,"Spanish");
7
8     int num;
9
10    cout<<"Ingrese número:\t\t";
11    cin>>num;
12
13    cout<<"\n";
14
15    switch(num)
16    {
17        case 1: cout<<"Vocal A"; break;
18        case 2: cout<<"Vocal E"; break;
19        case 3: cout<<"Vocal I"; break;
20        case 4: cout<<"Vocal O"; break;
21        case 5: cout<<"Vocal U"; break;
22        default: cout<<"No es numero para vocal";
23    }
24
25    cout<<"\n";
26
27    return 0;
28 }
```

Caso I

Caso II

2

8

switch(2)

switch(8)

Vocal E

No es numero para vocal

***fin de programa



RECUERDA:

¿Cuándo usar
Selectiva Simple
o Selectiva
Compuesta?:

```
1
2
3  if (cantidad < 5)
4      dcto = 0.1;
5
6  if (cantidad >= 5 && cantidad < 10)
7      dcto = 0.2;
8
9  if (cantidad >= 10)
10     dcto = 0.4;
11
```

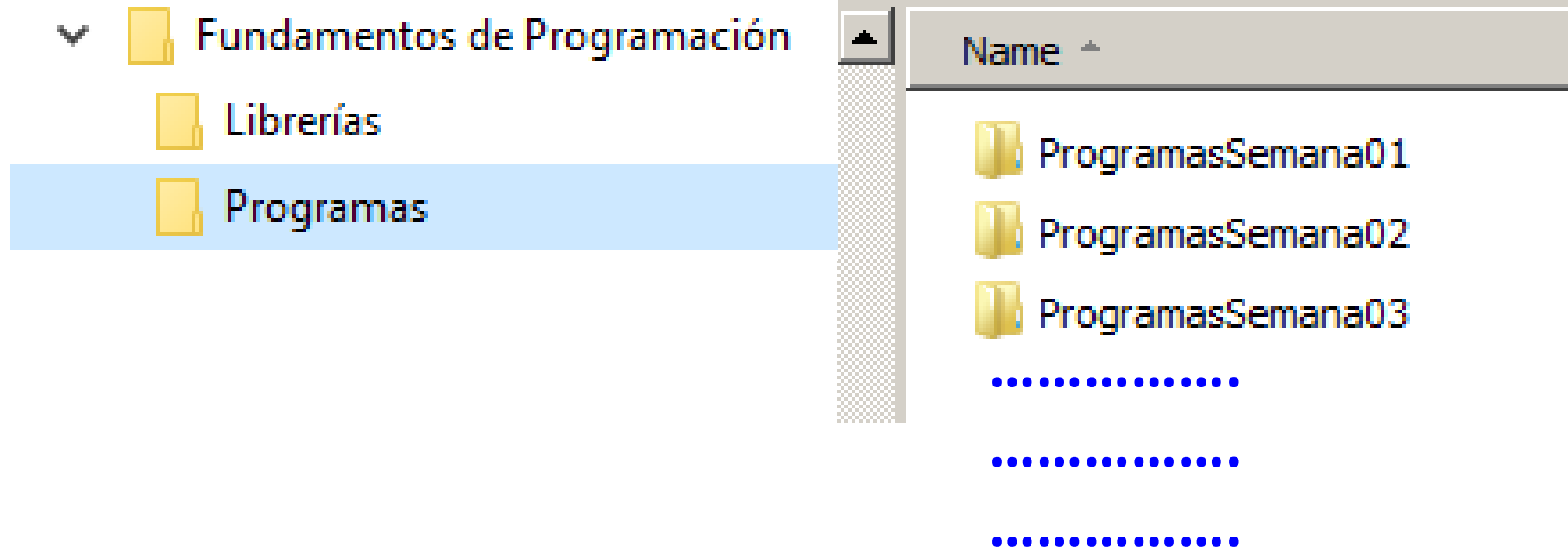
Si la cantidad fuese < 5
Que pasaría con el programa:
Sigue leyendo los siguientes if() ??
Es adecuado seguir ejecutando los demás if()??
Hay uso de recursos y tiempo innecesarios??

```
25
26  if (cantidad < 5)
27      dcto = 0.1;
28  else
29      if (cantidad >= 5 && cantidad < 10)
30          dcto = 0.2;
31      else
32          if (cantidad >= 10)
33              dcto = 0.4;
34
```

Si la cantidad fuese < 5
Que pasaría con el programa:
Sigue leyendo los siguientes if() ??
Es adecuado **no** seguir ejecutando los demás if()??
Hay uso de recursos y tiempo innecesarios??



Recomendación para crear los programas:



Ahora, elaboremos los programas ejemplos..!



Nombre Programa: Semana3Ventas.cpp

Dato que ingresas, ejemplo:

`cin>>Cant;`

Es dato que validas, ejemplo:

`if(Cant < 0)`

...

`else`

`cout<<"ERROR";`

Si el dato es correcto, recién puedes ingresar otro dato, ejemplo Precio.

```
1  #include<iostream>
2  using namespace std;
3
4  int main( )
5  {  double Cant, Prec, dcto, MBruto, MDcto, MFinal;
6
7      cout<<"Ingrese cantidad: ";
8      cin>>Cant;
9
10     if(Cant > 0 )  VALIDA
11     {
12         cout<<"Ingrese precio: ";
13         cin>>Prec;
14
15         if(Prec > 0 )  VALIDA
16         {
17             if(Cant > 0 && Cant <= 2 )
18                 dcto = 0;
19             else
20                 if(Cant > 2 && Cant <= 5 )
21                     dcto = 0.1;
22                 else
23                     if(Cant > 5 && Cant <= 10 )
24                         dcto = 0.15;
25                     else
26                         if(Cant > 10 )
27                             dcto = 0.2;
28
29             MBruto= Prec * Cant;
30             MDcto = MBruto * dcto;
31             MFinal = MBruto - MDcto;
32
33             cout<<"El monto Final es: " << MFinal<<endl;
34         }
35         else
36             cout<<"ERROR. Debe ser >0 " <<endl;  VALIDA
37     }
38     else
39         cout<<"ERROR. Debe ser >0 " <<endl;  VALIDA
40
41     return 0;
42 }
```

El if()-else **se usa para validar** el ingreso de datos.
Ejemplo: Valida la cantidad y valida el precio.

El if()-else **se usa para verificar** datos para el proceso de solución.

**COMPARA
(VERIFICA)**

Calcular y Mostrar

VALIDA

VALIDA





Nombre Programa: Programa2_Barcos

¿Y si queremos validar mayúsculas y minúsculas?

Elaborar el programa para: Ingrese la inicial (mayúscula o minúscula) de un tipo de barco y reporte el nombre del mismo, y si no existe, reportar "TIPO DE BARCO NO ENCONTRADO".

Guardar en carpeta ProgramaSemana3: Proyecto3Selectiva y el archivo Barcos.cpp

Inicial	Nombre de Barco
'B' o 'b'	Buque
'F' o 'f'	Fragata
'C' o 'c'	Crucero
otra letra	"TIPO DE BARCO NO ENCONTRADO".

Ejemplo:
VERIFICAR con fragata en mayúscula y minúscula

Los mensajes son los mismos...Eliminemos una línea de código redundante.

El case f no tiene código a ejecutar, y tampoco break, por eso ejecuta el siguiente case F (que tiene el mensaje esperado y el break).

Lo mismo ocurre con los demás casos.

Se elimina la sentencia inmediata anterior al mensaje

Y así valida mayúsculas y minúsculas, usando solo switch().

```
1  #include<iostream>
2  using namespace std;
3
4  int main( )
5  {
6      char letra;
7
8      cout<<"Ingrese letra: ";
9      cin>> letra;
10
11     if( letra=='b' || letra=='B' || letra=='c' || letra=='C' || letra=='f' || letra=='F' )
12     {
13         switch(letra)
14         {
15             case 'b': {cout<<"Es un Buque."; } break;
16             case 'B': {cout<<"Es un Buque."; } break;
17             case 'f': {cout<<"Es una Fragata."; } break;
18             case 'F': {cout<<"Es una Fragata."; } break;
19             case 'c': {cout<<"Es un Crucero."; } break;
20             case 'C': {cout<<"Es un Crucero."; } break;
21         }
22     }
23     else
24     {
25         cout<<"ERROR. Debe ser b o B o c o C o f o F "<<endl;
26     }
27
28     return 0;
29 }
```

03 Sentencias Secuenciales
01 Sentencia Selectiva Compuesta

Valida Datos

01 Sentencia Selectiva Múltiple
En cada caso, existe una sentencia Secuencial, pero pueden existir mas sentencias de diferentes tipos.



Conclusiones

Estructura de Control Selectiva

Puede ser:

- Simple
- Compuesta
- Múltiple

Se usa para validar el ingreso de datos

Se usa para comparar (verificar) datos para un cálculo

Programa

```
2 using namespace std;
3
4 int main()
5 {
6     double precioProd, MBruto, MBono, MDcto, MTotal, bono, dcto;
7     int cantProd;
8     char genero;
9
10    cout << "Ingrese cantidad: ";
11    cin >> cantProd;
12
13    if(cantProd > 0)
14    {
15        cout << "Ingrese precio: ";
16        cin >> precioProd;
17
18        if (precioProd > 0)
19        {
20            cout << "Ingrese inicial de genero: ";
21            cin >> genero;
22
23            if (genero == 'F' || genero == 'f' || genero == 'M' || genero == 'm')
24            {
25                switch (genero)
26                {
27                    case 'M':
28                    case 'm': { bono = 0.18;
29                             if(cantProd <= 10)
30                                 dcto = 0.2;
31                             else if(cantProd > 10)
32                                 dcto = 0.5;
33                             } break;
34
35                    case 'F':
36                    case 'f': { bono = 0.25;
37                             if(cantProd <= 10)
38                                 dcto = 0.3;
39                             else if(cantProd > 10)
40                                 dcto = 0.4;
41                             } break;
42                }
43
44                MBruto = cantProd * precioProd;
45                MBono = MBruto * bono;
46                MDcto = MBruto * dcto;
47                MTotal = MBruto + MBono - MDcto;
48                cout << "El monto bruto es:" << MBruto << endl;
49                cout << "El monto de bonificacion es:" << MBono << endl;
50                cout << "El monto de descuento es:" << MDcto << endl;
51                cout << "El monto total de pago es:" << MTotal << endl;
52            }
53            else
54                cout << "Error en ingreso de genero." << endl;
55        }
56        else
57            cout << "Error en ingreso de precio producto. " << endl;
58    }
59    else
60        cout << "Error en ingreso de cantidad producto. " << endl;
61    return 0;
62 }
```





Práctica

Crea programas con estructuras de control para la programación selectivas: Simple, Compuesta y Múltiple en el lenguaje de programación C/ C++.



Desarrollamos la Guía Práctica 03



Universidad
Continental

Guía práctica N°03

Fundamentos de Programación

Primera Unidad: Estructuras de control para la programación: secuencial y selectiva

Sección :	Apellidos :
Docente :	Nombres :
	Fecha : / / Duración: 100 min
	Tipo de práctica: Individual (x) Grupal ()

Instrucciones: Lea detenidamente cada enunciado y desarrolle lo solicitado.
Utilizar el Dev C++ para el desarrollo de los siguientes programas

- Elaborar un programa para ingresar el sueldo de una persona (mayor a cero), si el sueldo es igual a 930, mostrar un mensaje "SUELDO BASICO", en caso contrario mostrar un mensaje "SUELDO NO ES BASICO".
- Elaborar un programa para ingresar la edad de una persona (mayor igual a cero y menor igual a 120), si la edad es mayor igual a 18, mostrar un mensaje "ES MAYOR DE EDAD", en caso contrario mostrar un mensaje "ES MENOR DE EDAD".
- Elaborar un programa que determine el mayor, menor y el intermedio de tres números ingresados.
- Elaborar un programa que muestre el nivel de avance de un estudiante dentro de una universidad, según el número de créditos acumulados hasta la fecha (ver tabla). La cantidad acumulada de créditos debe ser mayor igual a cero y como máximo 160.

Créditos acumulados	Año académico
Menos de 32	Primer año
De 32 a 43	Segundo año
De 44 a 95	Tercer año
De 96 a 127	Cuarto año
De 128 a más	Quinto año

Ejemplo:
Ingrese la cantidad de créditos acumulados: 76 y se muestra el año académico: Tercer año

- Elaborar un programa para un restaurant que ofrece un descuento del 10% para consumos de hasta \$/30.00, un descuento de 20% para consumos mayores y para ambos casos aplica un impuesto del 18%. Determinar el importe a pagar por lo consumido, mostrando todos los importes. Además, tenga en cuenta que el consumo a ingresar, es mayor a cero.
- Elaborar un programa para un alumno que tiene 4 notas, y se le da la opción de presentar una tarea adicional para eliminar la nota más baja se pide:
 - El promedio del alumno con las 4 notas iniciales.
 - Mostrar la nota más baja.
 - Mostrar su nuevo promedio
- Elaborar un programa que calcule el monto total de pago, luego de ingresar el género de un cliente, además de la cantidad de productos a adquirir, con su respectivo precio. Para calcular el monto total a pagar, se asignan descuentos según el género y la cantidad adquirida.

	Genero	
	M	F
Cantidad	<=10	>10
Porcentaje Descuento	0.2	0.5

	Genero	
	M	F
Bono Descuento	0.18	0.25

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto3_Sueldo
Nombre Programa: Programa3_Sueldo

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto4_Edad
Nombre Programa: Programa4_Edad

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto5_MayorMenor
Nombre Programa: Programa5_MayorMenor

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto6_Creditos
Nombre Programa: Programa6_Creditos

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto7_Descuento
Nombre Programa: Programa7_Descuento

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto8_Notas
Nombre Programa: Programa8_Notas

Carpeta: Programas Semana 03
Nombre Proyecto: Proyecto9_VentasGenero
Nombre Programa: Programa9_VentasGenero

Referencias bibliográficas consultadas y/o enlaces recomendados

- JOYANES, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos, 4ª Edición. Madrid: McGraw-Hill.



NombrePrograma: Programa9_VentasGenero

Se requiere calcular el monto total de pago, luego de ingresar el genero de un cliente, además de la cantidad de productos a adquirir, con su respectivo precio. Para calcular el monto total a pagar, se asignan descuentos según el genero y la cantidad adquirida.

	Genero	
	M	F
Bono Descuento	0.18	0.25

	Genero			
	M		F	
Cantidad	<=10	>10	<=10	>10
Porcentaje Descuento	0.2	0.5	0.3	0.4

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      double precioProd, MBruto, MBono, MDcto, MTotal, bono, dcto;
7      int cantProd;
8      char genero;
9
10     cout << "Ingrese cantidad: ";
11     cin >> cantProd;
12
13     if(cantProd > 0)
14     {
15         cout << "Ingrese precio: ";
16         cin >> precioProd;
17
18         if (precioProd > 0)
19         {
20             cout << "Ingrese inicial de genero: ";
21             cin >> genero;
22
23             if (genero == 'F' || genero == 'f' || genero == 'M' || genero == 'm')
24             {
25                 switch (genero)
26                 {
27                     case 'M':
28                     case 'm':
29                         {
30                             bono = 0.18;
31                             if(cantProd <= 10)
32                                 dcto = 0.2;
33                             else
34                                 if(cantProd > 10)
35                                     dcto = 0.5;
36                             } break;
37
38                     case 'F':
39                     case 'f':
40                         {
41                             bono = 0.25;
42                             if(cantProd <= 10)
43                                 dcto = 0.3;
44                             else
45                                 if(cantProd > 10)
46                                     dcto = 0.4;
47                             } break;
48                 }
49
50                 MBruto = cantProd * precioProd;
51                 MBono = MBruto * bono;
52                 MDcto = MBruto * dcto;
53                 MTotal = MBruto + MBono - MDcto;
54                 cout << "El monto bruto es:" << MBruto << endl;
55                 cout << "El monto de bonificacion es:" << MBono << endl;
56                 cout << "El monto de descuento es:" << MDcto << endl;
57                 cout << "El monto total de pago es:" << MTotal << endl;
58             }
59             else
60                 cout << "Error en ingreso de genero." << endl;
61         }
62         else
63             cout << "Error en ingreso de precio producto. " << endl;
64     }
65     else
66         cout << "Error en ingreso de cantidad producto. " << endl;
67     return 0;
68 }
```

Valida Datos

Valida Datos

Verifica Datos

Verifica Datos

Verifica Datos

Verifica Datos



Conclusiones

Estructura de Control Selectiva

Puede ser:

- Simple
- Compuesta
- Múltiple

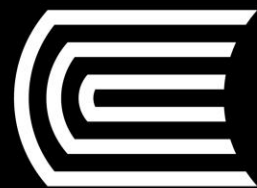
Se usa para validar el ingreso de datos

Se usa para comparar (verificar) datos para un cálculo

Programa

```
2 using namespace std;
3
4 int main()
5 {
6     double precioProd, MBruto, MBono, MDcto, MTtotal, bono, dcto;
7     int cantProd;
8     char genero;
9
10    cout << "Ingrese cantidad: ";
11    cin >> cantProd;
12
13    if(cantProd > 0)
14    {
15        cout << "Ingrese precio: ";
16        cin >> precioProd;
17
18        if (precioProd > 0)
19        {
20            cout << "Ingrese inicial de genero: ";
21            cin >> genero;
22
23            if (genero == 'F' || genero == 'f' || genero == 'M' || genero == 'm')
24            {
25                switch (genero)
26                {
27                    case 'M':
28                    case 'm': { bono = 0.18;
29                             if(cantProd <= 10)
30                                 dcto = 0.2;
31                             else
32                                 if(cantProd > 10)
33                                     dcto = 0.5;
34                             } break;
35
36                    case 'F':
37                    case 'f': { bono = 0.25;
38                             if(cantProd <= 10)
39                                 dcto = 0.3;
40                             else
41                                 if(cantProd > 10)
42                                     dcto = 0.4;
43                             } break;
44                }
45
46                MBruto = cantProd * precioProd;
47                MBono = MBruto * bono;
48                MDcto = MBruto * dcto;
49                MTtotal = MBruto + MBono - MDcto;
50                cout << "El monto bruto es:" << MBruto << endl;
51                cout << "El monto de bonificacion es:" << MBono << endl;
52                cout << "El monto de descuento es:" << MDcto << endl;
53                cout << "El monto total de pago es:" << MTtotal << endl;
54            }
55            else
56                cout << "Error en ingreso de genero." << endl;
57        }
58        else
59            cout << "Error en ingreso de precio producto. " << endl;
60    }
61    else
62        cout << "Error en ingreso de cantidad producto. " << endl;
63
64    return 0;
65 }
```





ucontinental.edu.pe