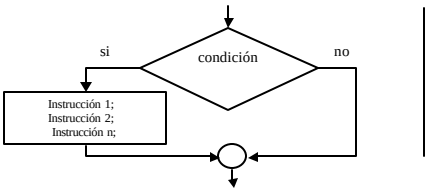
ESTRUCTURAS DE SELECCION

Para decidir cuando la ejecución de un programa cambia de dirección.

Existen diferentes tipos de selección:

1. **Selección Sencilla**.



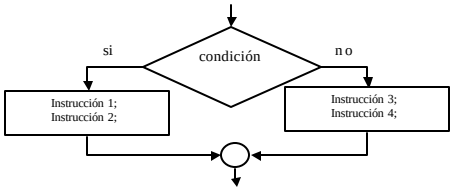
Sintaxis:

```
if (condición)
{
  instrucción 1;
  instrucción 2;
      :
  instrucción n;
}
```

Ejemplo:

```
if (n3<menor)
{
  menor = n3;
  n3 ++;
}
```

2. **Selección Doble.**



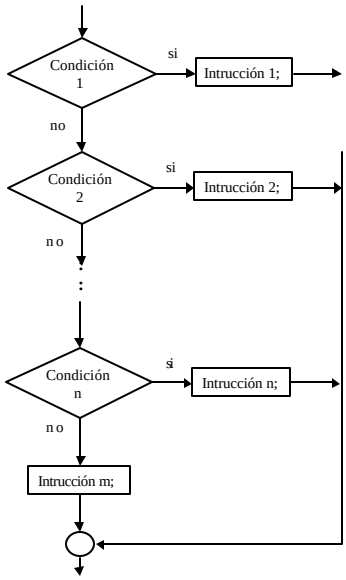
Sintaxis:

```
if (condición)
{
  instrucción 1;
  instrucción 2;
}
else {
  instrucción 3;
  instrucción 4;
}
```

Ejemplo:

```
if (n3<menor)
{
  menor = n3;
  n3 ++;
}
else  n3--;
```

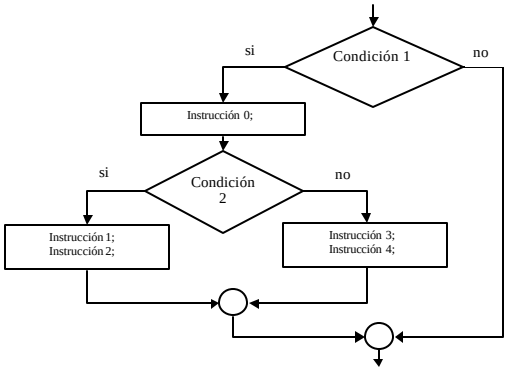
3. **Selección Múltiple.**



Sintaxis:

```
if (condición 1)
   {
   instrucción 1;  .....
   }
else if (condición 2)
    {
     instrucción 2; .....
    }
    else if (condición n)
        {
         instrucción n; .............
        }
        else {
              instrucción m; .............
              }
```

4. Selección Anidada.



Sintaxis:

```
if (condición 1)
{
  instrucción 0;
  if (condición 2)
    {
      instrucción 1;
      instrucción 2;
    }
   else
    {
      instrucción 3;
      instrucción 4;
    }
```

```
#include <iostream.h>
#include <math.h>
void main (void)
{
 double  x = 2, y = 3,  z;
 z = pow(x,y);
 cout << "z = " << z << " \n";
}
```

NOTAS:

- \n ➔ Secuencia de escape, significa "new line".
- \t ➔ Secuencia de escape, significa "tab".

Para darle un formato específico a las entradas y salidas, utilizamos la librería iomanip.h.

Formato de salida usando **cout <<** **:** En el ejemplo anterior si queremos reducir las cifras significativas del resultado en la pantalla.

```
#include <iostream.h>
#include <math.h>
#include <iomanip.h>
void main (void)
{
 double  x = 2, y = 3,  z;
 z = pow(x,y);
 cout << "z = " << setpresicion(2) << z << " \n";
}
```

```
}
```

OXPRESIONES LOGICAS COMPUESTAS

En ocasiones una condición o expresión lógica se compone de dos o más expresiones lógicas sencillas, es cuando se usan estos operadores. Operadores adicionales:

Operador en C	Operador lógico	
&&	AND	La condición es cierta si todas las que la componen son ciertas
||	OR	La condición es cierta si una de las que la componen son ciertas
!	NOT	Negación de la condición, si es cierta **not** lo hace 0, si es falsa **not** lo hace 1

Ejemplos:

if (0 <= x && x<= 100) ➔ si ambas condiciones se cumplen entonces es cierta (valor 1).

if (0 >= x || x >= 4) ➔ si una condición se cumplen entonces es cierta (valor 1).

Función setprecision(n) aumenta o disminuye en N el número de cifras significativas que se muestran en la salida.

Formato de entrada usando **cin >>** **:** En el ejemplo anterior si queremos pedir X y Y en lugar de inicializarlas.

```
#include <iostream.h>
#include <math.h>
#include <iomanip.h>
void main (void)
{
 double  x, y, z;
 cout << "Entre dos números (Oprima <enter> después de cada número)";
 cin >>x;
  cin >>y;
 z = pow(x,y);
 cout << "z = " << setpresicion(2) << z << " \n";
}
```