

## Material de apoyo 6to año para la guía 5 II lapso del 22 02 al 12 03 21

Estructura de datos:

### Estructuras Condicionales en C++

✓□ Sentencia If-Else: La sentencia if permite a un programador tomar una decisión para ejecutar una acción u otra cosa, basándose en el resultado verdadero o falso de una expresión.

Ejemplo de su declaración dentro del cuerpo del programa:

```
if (condición)
{
    sentencia 1;      //hacer algo
    sentencia 2;
}
else
{
    sentencia 3;      //hacer otra cosa
    sentencia 4;
}
```

Se tomará este ejercicio de ejemplo para su explicación: *se enumeran las líneas de código para hacer referencia a ellas en las explicación para efectos de desarrollo no debe enumerar se.*

```
1#include <stdio.h>
2#include <stdlib.h>
3 intmain ()
4 {
5 inti=1, j=2;
6if (i<j){
7 printf("i es menor que j") ;}
```

```

8 else{
9 printf ("i es mayor que j");}
10 return 0;
11 }

```

La ejecución de una sentencia if consiste en:

1. Evaluar la condición entre paréntesis.

✓  Esta condición puede ser una **expresión numérica**.

Ejemplo: La variable X debe tener un valor diferente de cero, lo cual se escribiría como:  
 $x \neq 0$  véase la tabla de operadores lógicos de la guía anterior.

✓  **De relación** ejemplo: el valor de la variable i debe ser menor que el de la variable j.  
 Se escribiría `if (i<j)`

✓  **De lógica** ejemplo: el valor de la variable I y el de j debe ser diferente del cero, el cual se escribiría: `i != 0 && j != 0`, o bien simplificada `I && j`.

2. Si el resultado de evaluar la condición es verdadero, es decir que es distinto de cero se ejecutan la sentencia encerrada entre llaves después de if, si la condición es falsa se ejecuta la línea de código número 9, por lo que se usa else.

✓  Anidamiento If-Else: se dice que una sentencia if-else esta anidada cuando dentro de su formato general se encuentran otras sentencias if. Esto es:

```

if (condición 1)
{
    if (condición 2) sentencia 1;
}
else
sentencia 2;

```

Las llaves del código anterior indican que la cláusula else está emparejada con el primer if.

Cuando en el código de un programa aparecen sentencias if-else anidadas y no existen llaves que definan claramente cómo emparejar los else y los if, la regla que se debe utilizar es que cada else corresponde con el if más próximo que no haya sido emparejado. Ejemplo:

```
if (condición 1)
  if (condición 2)
    sentencia 1;
  else
    sentencia 2;
```

En este código la cláusula else corresponde al segundo if.

Los anidamientos pueden llegar a confundir en cuanto a su organización, la manera de hacerlo más legible es que el programador utilice tabulaciones para señalar cada bloque. Ejemplo

	<pre>if (condición 1) {   if (condición 2)     {sentencia 1;}   else     {sentencia 2;} }</pre>	<pre>if (condición 1) {   if (condición 2)     {sentencia 1;} } else {sentencia 2;}</pre>
--	---	---

La sentencia switch

Esta sentencia nos permite seleccionar en función de condiciones múltiples. Su sintaxis es:

```
switch (expresión) {
  case valor_1: sentencia 11;
  sentencia 12;
  ...
  sentencia 1n;
  break;
  case valor_2: sentencia 21;
  sentencia 22;
  ...
  sentencia 2m;
  break;
  ...
}
```

```
default: sentencia d1;  
sentencia d2;  
...  
sentencia dp  
}
```

El paréntesis en la expresión es obligatorio. El funcionamiento es el siguiente, si al evaluar la expresión se obtiene uno de los valores indicados por `case valor_i` se ejecutan todas las sentencias que encontremos hasta llegar a un `break` (o al cierre de las llaves). Si no se verifica ningún `case` pasamos a las sentencias `default`, si existe (`default` es opcional) y si no existe no hacemos nada.

Indicaremos que si queremos hacer lo mismo para distintos valores podemos escribir los `case` seguidos sin poner `break` en ninguno de ellos y se ejecutará lo mismo para todos ellos.