

Operadores

Para manejar variables o contantes en C++ es necesario contar con *operadores*. Existen diferentes tipos y son explicados a continuación.

Asignación =

Utilizado para asignar valores a variables.

Ejemplo: `x = 1`

Aritméticos

Operador	Descripción
+	Adición
-	Sustracción
*	Multiplicación
/	División
%	Módulo (resto de una división)

Asignación de compuestos

Operador	Ejemplo	Equivalente a
+=	<code>x += 2</code>	<code>x = x + 2</code>
-=	<code>x -= 2</code>	<code>x = x - 2</code>
*=	<code>x *= 2</code>	<code>x = x * 2</code>
/=	<code>x /= 2</code>	<code>x = x / 2</code>
%=	<code>x %= 2</code>	<code>x = x % 2</code>

Incremento y decremento

Operador	Descripción
<code>++</code>	Incremento en 1
<code>--</code>	Decremento en 1

Ejemplo:

```
int x = 2;  
x++;      // Incrementa en 1 el valor de x  
int y = x; // El valor de y es 3
```

Relacionales y de comparación

Operador	Descripción
<code>==</code>	Igual a
<code>!=</code>	Distinto a
<code><</code>	Menor a
<code>></code>	Mayor a
<code><=</code>	Menor o igual a
<code>>=</code>	Mayor o igual a

Ejemplo:

```
(7 == 5)    // se evalúa como false  
(5 > 4)     // se evalúa como true  
(3 != 2)    // se evalúa como true  
(6 >= 6)    // se evalúa como true  
(5 < 5)     // se evalúa como false
```

No solamente enteros se pueden comparar. También se pueden utilizar variables.

Tomando en cuenta que `a = 2`, `b = 3` y `c = 6`.

```
(a == 5)    // se evalúa como false, ya que a no es 5  
(a * b >= c) // se evalúa como true, ya que (2 * 3 >= 6) es true  
(b + 4 > a*c) // se evalúa como false, ya que (3 + 4 > 2 * 6) es false  
((b = 2) == a) // se evalúa como true
```

Lógicos

Operador `&&`

a	b	a <code>&&</code> b
true	true	true
true	false	false
false	true	false
false	false	false

Operador `||`

a	b	a <code> </code> b
true	true	true
true	false	true
false	true	true
false	false	false

Ejemplo:

```
( ( 5 == 5 ) && ( 3 > 6 ) ) // se evalúa como false ( true && false )
( ( 5 == 5 ) || ( 3 > 6 ) ) // se evalúa como true ( true || false )
```

Operador `!`

Utilizado para negar una condición, cambiando su evaluación de `true` a `false` o viceversa.

Ejemplo:

```
!(5 == 5) // Se evalúa como false.
if(!false) { cout << "¡Es true!" << endl; } // Se evalúa como true
```

Precedencia

```
x = 5 + 7 % 2; // x = 6

x = 5 + (7 % 2); // x = 6 (lo mismo pero sin paréntesis)
x = (5 + 7) % 2; // x = 0
```