

Introducción a las Variables y Operadores en C++

Victor Armando Jaramillo Moreno

8 Febrero 2025

1 ¿Qué son las variables?

Las variables son espacios en la memoria de la computadora que guardan información. Puedes pensar en ellas como cajas donde guardas cosas para usarlas más tarde.

Por ejemplo, si quieres que un programa recuerde tu edad, puedes guardar ese número en una variable.

2 Declarando variables

Antes de usar una variable, debes decirle a la computadora qué tipo de dato guardará.

Por ejemplo, si quieres guardar un número entero (sin decimales), usas:

```
int edad;
```

Aquí, `int` significa "número entero" y `edad` es el nombre de la variable.

3 Asignando valores a una variable

Después de declarar una variable, puedes darle un valor.

Por ejemplo, si quieres que la variable `edad` guarde el número 16:

```
edad = 16;
```

También puedes hacer ambas cosas en una sola línea:

```
int edad = 16;
```

4 Tipos de variables en C++

Existen diferentes tipos de variables según lo que necesites guardar.

4.1 Booleanos (bool)

Los booleanos solo pueden guardar dos valores: `true` (verdadero) o `false` (falso). Se usan para responder preguntas de "sí" o "no".

```
bool aprobado = true;
```

4.2 Números enteros (int)

Los números enteros (`int`) son aquellos sin decimales, como 5, -3 o 1000.

```
int cantidad = 10;
```

4.3 Números con decimales (float y double)

Si necesitas guardar números con decimales, puedes usar `float` o `double`.

```
float precio = 99.99;  
double temperatura = 36.6;
```

4.4 Letras o caracteres (char)

Si quieres guardar una sola letra o símbolo, usa `char`.

```
char inicial = 'A';
```

Los caracteres deben ir entre comillas simples (' ').

5 Uso de Operadores en C++

Los operadores en C++ nos permiten hacer operaciones matemáticas con variables, comparaciones y más.

5.1 Operadores Aritméticos

Estos operadores se usan para hacer cálculos matemáticos:

- + (suma): `a + b`
- - (resta): `a - b`
- * (multiplicación): `a * b`
- / (división): `a / b`
- % (módulo, resto de la división): `a % b`

Ejemplo:

```
int a = 10, b = 3;
int suma = a + b; // suma = 13
int resta = a - b; // resta = 7
int multiplicacion = a * b; // multiplicacion = 30
int division = a / b; // division = 3 (sin decimales)
int modulo = a % b; // modulo = 1 (resto de la división)
```

5.2 Operadores de Comparación

Estos operadores comparan dos valores y devuelven `true` o `false`.

- `==` (igual a): `a == b`
- `!=` (diferente de): `a != b`
- `>` (mayor que): `a > b`
- `<` (menor que): `a < b`
- `>=` (mayor o igual que): `a >= b`
- `<=` (menor o igual que): `a <= b`

Ejemplo:

```
int x = 5, y = 10;
bool resultado = (x < y); // resultado = true
```

5.3 Operadores Lógicos

Se usan para combinar condiciones.

- `&&` (AND): Devuelve `true` si ambas condiciones son verdaderas.
- `||` (OR): Devuelve `true` si al menos una condición es verdadera.
- `!` (NOT): Invierte el valor de la condición.

Ejemplo:

```
bool condicion1 = (5 > 3); // true
bool condicion2 = (10 < 2); // false

bool resultado1 = condicion1 && condicion2; // false
bool resultado2 = condicion1 || condicion2; // true
bool resultado3 = !condicion1; // false
```

6 Entrada de datos del usuario

Si quieres que el usuario escriba un valor para la variable, puedes usar `cin`.

Por ejemplo, si quieres que alguien escriba su edad:

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

int main() {
    int edad;
    cout << "Ingresa tu edad: ";
    cin >> edad;
    cout << "Tienes " << edad << " años." << endl;
    return 0;
}
```

7 Conclusión

Las variables y los operadores son esenciales en la programación. Gracias a ellos, podemos hacer cálculos, comparar valores y tomar decisiones en nuestros programas.