

Operadores Relacionales y Operadores Lógicos

Operadores Relacionales o de Comparación

Los operadores relacionales son aquellos que se utilizan para comparar valores y devolver un resultado booleano (verdadero o falso). En C, estos operadores son fundamentales para tomar decisiones en estructuras de control como if, while, for, etc.

En **C estándar (C89/C90 y C99)** **no existe un tipo de dato booleano nativo** como en otros lenguajes (por ejemplo, `bool` en C++ o `boolean` en Java).

Sin embargo, desde **C99** se introdujo el tipo `bool` en la librería estándar `<stdbool.h>`, lo que permite trabajar con valores booleanos de forma más clara.



Igual a ==

1. Igual a ==

- Comprueba si dos valores son exactamente iguales
- Retorna 1 (verdadero) si los valores son iguales
- Retorna 0 (falso) si los valores son diferentes

Es importante distinguirlo del operador de asignación =



Diferente de !=

2. Diferente de !=

- Comprueba si dos valores son diferentes
- Retorna 1 (verdadero) si los valores son distintos
- Retorna 0 (falso) si los valores son iguales



Mayor que >

3. Mayor que >

- Comprueba si el valor de la izquierda es mayor que el de la derecha
- Retorna 1 (verdadero) si se cumple la condición
- Retorna 0 (falso) si no se cumple



Menor que <

4. Menor que <

- Comprueba si el valor de la izquierda es menor que el de la derecha
- Retorna 1 (verdadero) si se cumple la condición
- Retorna 0 (falso) si no se cumple



Mayor o igual que \geq

5. Mayor o igual que \geq

- Comprueba si el valor de la izquierda es mayor o igual que el de la derecha
- Retorna 1 (verdadero) si se cumple la condición
- Retorna 0 (falso) si no se cumple



Menor o igual que <=

5. Menor o igual que <=

- Comprueba si el valor de la izquierda es menor o igual que el de la derecha
- Retorna 1 (verdadero) si se cumple la condición
- Retorna 0 (falso) si no se cumple



Consideraciones

Los operadores relacionales pueden comparar valores de diferentes tipos, pero se realizará una conversión implícita (por ejemplo int y float)

Se usan en estructuras de control, por ejemplo con el if, while etc

Se pueden combinar con operadores lógicos para crear condiciones más complejas:

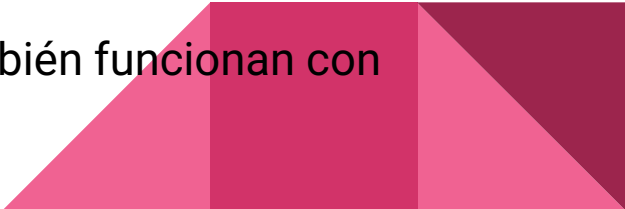
```
int a = 10, b = 20, c = 15;
```

```
if (a < b && b > c) {
```

```
    printf("Condición verdadera");
```

```
}
```

Comparación de caracteres Los operadores relacionales también funcionan con caracteres, comparando sus valores ASCII



Errores Comunes

- Confundir `==` (comparación) con `=` (asignación)
- Usar `==` para comparar números de punto flotante (por problemas de precisión)
- No usar paréntesis en condiciones complejas




Operadores lógicos

Los operadores lógicos permiten combinar múltiples condiciones en expresiones booleanas. Se utilizan comúnmente en estructuras de control como `if`, `while` y `for`.

Los operadores lógicos evalúan la expresión lógica y devuelven un resultado.

El resultado siempre es un valor booleano. Un valor booleano determina si la expresión es verdadera `true` o falsa `false`.

Hay tres operadores lógicos en la programación en C: el AND lógico (`&&`), el OR lógico (`||`) y el NOT lógico (`!`).



Operador AND &&

El operador lógico AND (&&) verifica si todos los operandos son verdaderos: el resultado es verdadero sólo cuando todos los operandos son verdaderos.

Primer operando	Segundo Operando	Resultado
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE



Operador AND &&

```
#include <stdio.h>
```

```
int main( ) {
```

```
    int a = 20;
```

```
    int b = 30;
```

```
    if (a > 10 && b > 10){
```

```
        printf("Ambos números son mayores que 10\n");
```

```
    }
```



Operador OR ||

La operación lógica OR (||) verifica si al menos uno de los operandos es verdadero; el resultado es verdadero si al menos uno de los operandos es verdadero.

Primer operando	Segundo Operando	Resultado
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Con el operador OR (||), si el primer operando es verdadero, entonces el segundo operando no se evalúa.



Operador OR ||

```
#include <stdio.h>
```

```
int main(void) {
```

```
    int a = 20;
```

```
    int b = 5;
```

```
    if (a > 10 || b > 10){
```

```
        printf("Al menos uno de los números es mayor que 10");
```

```
    }
```



Operador NOT !

El operador lógico NOT (!) niega el operando – es decir, devuelve lo opuesto al operando.

Si el operando es `true`, devuelve `false`.

Y si es `false`, devuelve `true`.

Aquí está la tabla de verdad para el operador NOT (!):



Operador NOT !

```
int estaLloviendo = 1; // 1 significa verdadero
```

```
// Usamos el operador ! para negar el valor
```

```
if (!estaLloviendo) {
```

```
    printf("No está lloviendo, es un buen día para salir.\n");
```

```
} else {
```

```
    printf("Está lloviendo, mejor quédate en casa.\n");
```

```
}
```

