

Estructuras de Control Condicional

Rodolfo Christian Catunta Uturunco (Elemental Bolivia)



21 de febrero de 2025

- 1 Sentencia de Decisión: if
- 2 Sentencia de Decisión: switch
- 3 Operador Ternario

- 1 Sentencia de Decisión: if
- 2 Sentencia de Decisión: switch
- 3 Operador Ternario

Sentencia de Decisión if

La sentencia de decisión if permite evaluar una expresión **booleana** para de esa forma seleccionar el set de instrucciones a ejecutar.

```
1 if(expresion booleana){  
2     // Parte VERDADERA  
3     // Set de instrucciones si la exp. booleana es VERDADERA  
4 }  
5 else{  
6     // Parte FALSA  
7     // Set de Instrucciones si la exp. bool. es FALSA  
8 }
```



Expresiones Booleanas

Las expresiones booleanas en C++ usualmente se construyen en base a **operadores de comparación** y a **conectivos lógicos**.

OPERACIONES DE COMPARACIÓN

| | | | |
|------|---------------|------------------|------------------|
| $<$ | MENOR | $3 < 7$ (V) | $5 < 5$ (F) |
| $<=$ | MENOR O IGUAL | $5 <= 5$ (V) | $9 <= 8$ (F) |
| $>$ | MAYOR | $7 > 2$ (V) | $3 > 5$ (F) |
| $>=$ | MAYOR O IGUAL | $9 >= 5$ (V) | $5 >= 9$ (F) |
| $==$ | IGUAL | $20 == 10$ (V) | $'a' == 'b'$ (F) |
| $!=$ | DIFERENTE | $'a' != 'b'$ (V) | $10 != 10$ (F) |

char →

Conectores Lógicos

AND (Y) and &&

| P | q | P and q |
|---|---|---------|
| V | V | V |
| V | F | F |
| F | V | F |
| F | F | F |

$$\begin{array}{ccc} V & & F \\ 7 > 3 & \text{and} & 3 < 4 \end{array}$$

F

OR (O) or ||

| P | q | P or q |
|---|---|--------|
| V | V | V |
| V | F | V |
| F | V | V |
| F | F | F |

$$\begin{array}{ccc} F & & V \\ 3 > 9 & \text{or} & 3 < 5 \end{array}$$

V

NOT (NO) not ~

| P | not P |
|---|-------|
| V | F |
| F | V |

$$\text{not}(7 > 9) \rightarrow V$$

Ejemplos

① $3 > 1$ and $7 < 10$

② $4 < 10$ or $7 \neq 7$

③ $\text{not}(3 < 1 \text{ or } 4 > 4)$

④ Para que valores ^{enteros} de x la expresión es falsa
 $x < 3$ or $x > 7$

Ejemplo if: anidado

```
1 if (numero>0){  
2     cout<<"POSITIVO"<<endl;  
3 }  
4 else if (numero<0){  
5     cout<<"NEGATIVO"<<endl;  
6 }  
7 else {  
8     cout<<"CERO"<<endl;  
9 }
```


- 1 Sentencia de Decisión: if
- 2 Sentencia de Decisión: switch
- 3 Operador Ternario

Sentencia de Decisión switch

La sentencia de decisión **switch**, a diferencia de **if** permite la creación de múltiples caminos de decisión. Es muy útil en la creación de menús.

```
1 switch(variable a evaluar){  
2     case primer_caso:  
3         // Instrucciones primer_caso  
4         break;  
5     case segundo_caso:  
6         // Instrucciones segundo_caso  
7         break;  
8     case tercer_caso:  
9         // Instrucciones tercer_caso  
10        break;  
11    default:  
12        // Instrucciones por defecto  
13        // Cuando no se ingresa por ningun caso  
14 }
```

Ejemplo switch

```
1 switch (opcion){  
2     case 'A':  
3         cout<<" Selecciono la opcion A"<<endl;  
4         break;  
5     case 'B':  
6         cout<<" Selecciono la opcion B"<<endl;  
7         break;  
8     case 'C':  
9         cout<<" Selecciono la opcion C"<<endl;  
10        break;  
11    default:  
12        cout<<"Su opcion no es valida"<<endl;  
13 }
```

- 1 Sentencia de Decisión: if
- 2 Sentencia de Decisión: switch
- 3 Operador Ternario

Operador ternario

El operador ternario sirve para abreviar una sentencia condicional. Es usado principalmente en la asignacion de variables o en la impresión de resultados.

```
1 //Forma
2 (condicion) ? proceso1 : proceso2;
3
4 //Asignacion de variables
5 int x = (edad>18) ? edad : 18;
6
7 //En la impresion
8 cout<< ((edad>=18) ? "Mayor de edad" : "Menor de edad") <<
    endl;
```