

## OPERADORES LOGICOS

Los operadores lógicos se utilizan para combinar dos o mas condiciones y se basan en el valor que regresa cada condición (true o false) evaluada y se le aplica la tabla de verdad.

Operador	Nombre	Uso	Resultado
&&	AND	op1 && op2	true si op1 y op2 son true, sino es false
	OR	op1    op2	true si op1 o op2 es true, sino es false
!	NOT	!op1	true si op1 es false y false si op1 es true

### OPERADOR && ( AND )

Resultado Condición uno	Resultado Condición dos	Resultado final
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

### Ejemplo 01

Esta tabla vista de otra forma, siendo true u valor de 1 y false un valor de 0. Vamos a evaluar las siguientes variables:

```
int op1 = 10;
int op2 = 10;

// Saber si son iguales y mayor que cero
if( (op1 == op2) && (op2 > 0) ){
    System.out.println ("Son iguales y mayores que cero");
}
```

## Resultado

```
General Output
-----Configuration:
Son iguales y mayores que cero
Process completed.
```

Sustituyendo valores en la expresión y aplicando la tabla de verdad, tenemos:

Condición 1	Condición 2	Expresión	Resultado
Op1 == op2	Op2 > 0		
true	true	true && true	true

## Ejemplo 02

```
int op1 = 10;

// Saber si es positivo y menor que 10

if( ( op1 > 0) && (op1 < 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

## Resultado

```
General Output
-----Config
No cumple la condicion
Process completed.
```

Condición 1	Condición 2	Expresión	Resultado
op1 > 0	Op1 < 10		
true	false	true && false	false

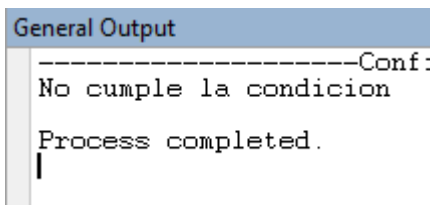
## Ejemplo 03

```
int op1 = 8;

// Saber si esta entre 10 y 20

if( ( op1 >= 10) && (op1 <= 20) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

### Resultado



```
General Output
-----Conf:
No cumple la condicion
Process completed.
|
```

Condición 1	Condición 2	Expresión	Resultado
op1 >= 10	Op1 <= 20		
false	true	false && true	false

## Ejemplo 04

```
int op1 = -8;

// Saber si es positivo e igual a 10

if( ( op1 > 0) && (op1 == 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

## Resultado

```

General Output
-----Configura
No cumple la condicion
Process completed.
    
```

Condición 1	Condición 2	Expresión	Resultado
op1 > 0	op1 == 10		
false	false	false && false	false

*NOTA: Con este operador AND, para que el resultado final sea verdad, todas las condiciones deben ser verdaderas, por eso se le llama una multiplicación lógica ya que false representa un 0 y un true un valor de 1*

## OPERADOR || (OR)

Resultado Condición uno	Resultado Condición dos	Resultado final
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

## Ejemplo 01

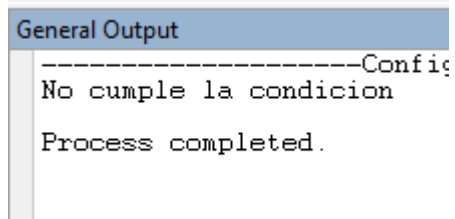
---

```
int op1 = -8;

// Saber si es negativo o igual a 10

if( ( op1 > 0) || (op1 == 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

### Resultado



```
General Output
-----Config
No cumple la condicion
Process completed.
```

Condición 1	Condición 2	Expresión	Resultado
op1 > 0	op1 == 10		
false	false	false && false	false

## Ejemplo 02

---

```
int op1 = -8;

// Saber si es negativo o igual a 10

if( ( op1 < 0) || (op1 == 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

### Resultado

```

General Output
-----Config
Si cumple la condicion
Process completed.
|
    
```

Condición 1	Condición 2	Expresión	Resultado
op1 < 0	op1 == 10		
true	false	true && false	true

### Ejemplo 03

```

int op1 = 10;

// Saber si es negativo o igual a 10

if( ( op1 < 0) || (op1 == 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
    
```

### Resultado

```

General Output
-----Config:
Si cumple la condicion
Process completed.
    
```

Condición 1	Condición 2	Expresión	Resultado
op1 < 0	op1 == 10		
false	true	false && true	true

## Ejemplo 04

```
int op1 = 10;

// Saber si es positivo o igual a 10

if( ( op1 > 0) || (op1 == 10) ){
    System.out.println ("Si cumple la condicion");
} else {
    System.out.println ("No cumple la condicion");
}
```

### Resultado

```
General Output
-----Config
Si cumple la condicion
Process completed.
```

Condición 1	Condición 2	Expresión	Resultado
op1 > 0	op1 == 10		
true	true	true && true	true

*NOTA: A diferencia del AND, el OR con que se cumpla una condición el resultado será verdadero*

## OPERADOR NOT

Su tabla es la siguiente:

Resultado Condición uno	Resultado final
FALSE	TRUE
TRUE	FALSE

El uso de este operador simplemente es para asegurar que algo se procese sobre todo en proceso electrónicos, pero en raras ocasiones se usa en JAVA.

Si por ejemplo, un sensor que supervisa el flujo eléctrico, detecta que no hay el suministro, puede cambiar el estado de una variable para que continúe el proceso sin interrupción hasta que llegue el servicio normal de la energía eléctrica.

### Ejemplo 01

---

```
// Indicador de luz, 0 no hay luz y 1 si hay

int op1 = 0;

// Saber si activar la planta auxiliar

if( !( op1 == 1) ){
    System.out.println ("Activar alarma de suministro de luz");
} else {
    System.out.println ("Sin Alarmas");
}
```

*NOTA: Puede ser usado en un ciclo do while para validar una entrada de un menú.*