

UTNIFRGP

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO

Apuntes de clase de la asignatura

INTRODUCCIÓN A LA PROGRAMACIÓN

EXPRESIONES LÓGICAS

CURSO DE NIVELACIÓN – TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN

2024



EXPRESIONES LÓGICAS

Una expresión lógica es una combinación de una o más expresiones que se evalúan para obtener un valor booleano, es decir, Verdadero (*true*) o Falso (*false*).

Estas expresiones pueden incluir números, constantes, variables, y operadores tanto relacionales como lógicos.

Se utilizan comúnmente en estructuras de decisión, también conocidas como selectivas o condicionales, así como en estructuras repetitivas.

Las Expresiones Lógicas se pueden clasificar en:

- Expresiones Lógicas Simples.
- Expresiones Lógicas Complejas.

EXPRESIONES LÓGICAS SIMPLES

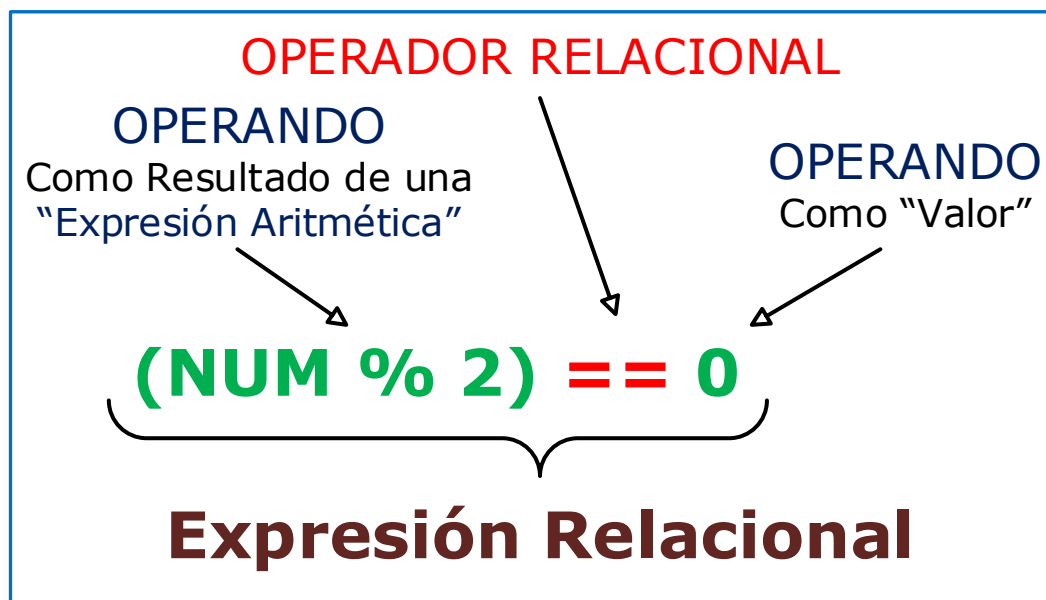
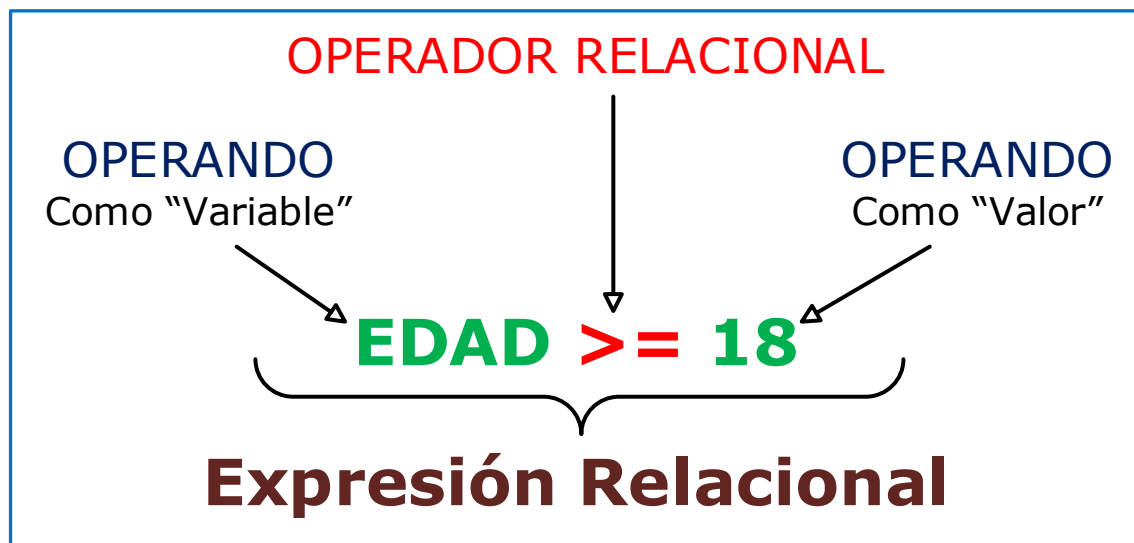
Se considera una Expresión Lógica Simple aquella que contienen una sola condición (*Expresión Lógica*), la cual suele ser una Expresión Relacional.

Una expresión relacional es una combinación de valores, variables, constantes y operadores relacionales que se evalúan para obtener un valor booleano, es decir, Verdadero (*true*) o Falso (*false*), se puede decir que toda Expresión Relacional es una Expresión Lógica porque se evalúa a Verdadero o Falso.

OPERADORES RELACIONALES

Los operadores de relación se utilizan para describir una relación entre dos operandos (*valores*). Estos operandos pueden ser valores, variables, constantes, o resultados de una expresión aritmética, pero deben ser del mismo tipo de dato.

OPERADORES RELACIONALES	
Operador	Descripción
==	IGUAL A ...
!=	DISTINTO DE ...
>	MAYOR QUE ...
<	MENOR QUE ...
>=	MAYOR O IGUAL QUE ...
<=	MENOR O IGUAL QUE ...



Ejemplos de uso Operadores Relacionales:

Mayor que (>):

Evalúa si el valor de la izquierda es Mayor que el de la derecha.

10 > 8 Devuelve -> Verdadero

Menor que (<):

Evalúa si el contenido de la variable Nota es Menor que 7.

Nota=9 (Asignamos previamente un valor a la variable Nota)

Nota < 7 Devuelve -> Falso



Mayor o Igual que (\geq):

Evalúa si el valor de la variable Nota es Mayor o Igual que el valor de la variable Ap.

Nota=6 (*Asignamos previamente un valor a la variable Nota*)

Ap.=6 (*Asignamos previamente un valor a la variable Ap*)

Nota \geq Ap Devuelve -> Verdadero

Menor o Igual que (\leq):

Evalúa si el valor devuelto por la expresión aritmética (Nota1 + Nota2 + Nota3) / 3 es menor o igual que el valor de la variable Ap.

Nota1=6 (*Asignamos previamente un valor a la variable Nota*)

Nota2=8 (*Asignamos previamente un valor a la variable Nota*)

Nota3=7 (*Asignamos previamente un valor a la variable Nota*)

Ap=6 (*Asignamos previamente un valor a la variable Ap*)

(Nota1+Nota2+Nota3)/3 \leq Ap Devuelve -> Falso

Igual que ($=$):

Evalúa si el valor de la variable Nota es Igual que 10.

Nota=10 (*Asignamos previamente un valor a la variable Nota*)

Nota $=$ 10 Devuelve -> Verdadero

Distinto que (\neq):

Evalúa si el valor de la variable Nota es Distinto que el valor de la variable Ap.

Nota=6 (*Asignamos previamente un valor a la variable Nota*)

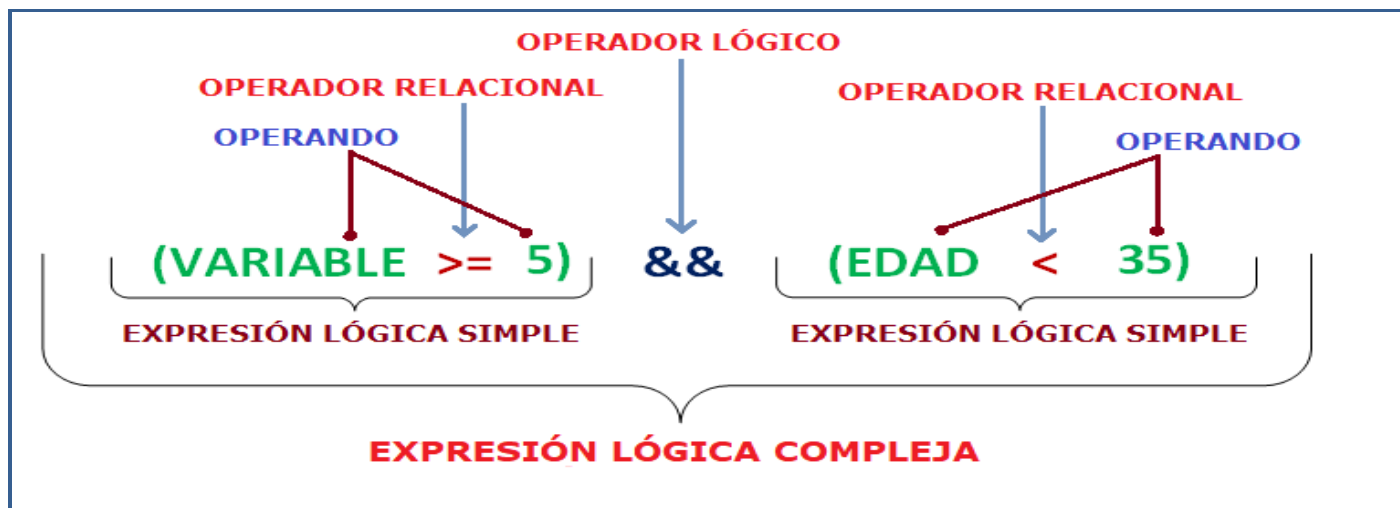
Ap=6 (*Asignamos previamente un valor a la variable Ap*)

Nota \neq Ap Devuelve -> Falso



EXPRESIONES LÓGICAS COMPLEJAS

Son expresiones que combinan dos o más expresiones lógicas simples utilizando operadores lógicos, lo que permite evaluar múltiples expresiones lógicas a la vez. Estas expresiones se evalúan para obtener un valor booleano, es decir, Verdadero (*true*) o Falso (*false*).



OPERADORES LÓGICOS

Los operadores lógicos permiten combinar expresiones lógicas simples para crear expresiones lógicas complejas. Los operandos pueden ser una combinación de expresiones lógicas simples o valores lógicos (*verdadero/falso*).

OPERADOR	REPRESENTACIÓN	DESCRIPCIÓN
AND	$\&\&$	Y- CONJUNCIÓN
OR	$ $	O - DISYUNCIÓN
NOT	!	NEGACIÓN

PRIORIDAD	OPERADOR	DESCRIPCIÓN
1	!	NEGACIÓN
2	$\&\&$	Y- CONJUNCIÓN
3	$ $	O - DISYUNCIÓN



TABLA DE VERDAD

Una tabla de verdad es una herramienta utilizada en lógica y en ciencias de la computación para mostrar de manera sistemática cómo se comportan las expresiones lógicas bajo diferentes combinaciones de valores de verdad.

En otras palabras, una tabla de verdad lista todas las posibles combinaciones de valores (*verdadero* y *falso*) para los operandos involucrados en una expresión lógica, y muestra el resultado de la expresión para cada combinación.

Componentes de una Tabla de Verdad

Cada operando o expresión lógica individual tiene su propia columna en la tabla. Después de las columnas de operandos, se incluyen las columnas que muestran los resultados de las combinaciones de esos operandos cuando se aplican operadores lógicos como **AND**, **OR**, y **NOT**. Una tabla de verdad es una representación que facilita la comprensión y el análisis del comportamiento de las expresiones lógicas bajo todas las combinaciones posibles de valores de verdad. Es una herramienta fundamental para quienes trabajan en lógica y programación.

OPERADOR LÓGICO AND (&&)

El operador lógico **AND**, representado comúnmente como **&&**, se utiliza para combinar dos o más expresiones lógicas. Devuelve Verdadero (*true*) solo si todas las expresiones que combina son verdaderas. Si alguna de las expresiones es Falso (*false*), el resultado de la expresión completa será Falso. La función principal de este operador es evaluar que todas las expresiones lógicas dadas sean verdaderas.

El término "**Y**" expresa la conjunción que indica que dos o más expresiones lógicas deben cumplirse simultáneamente.

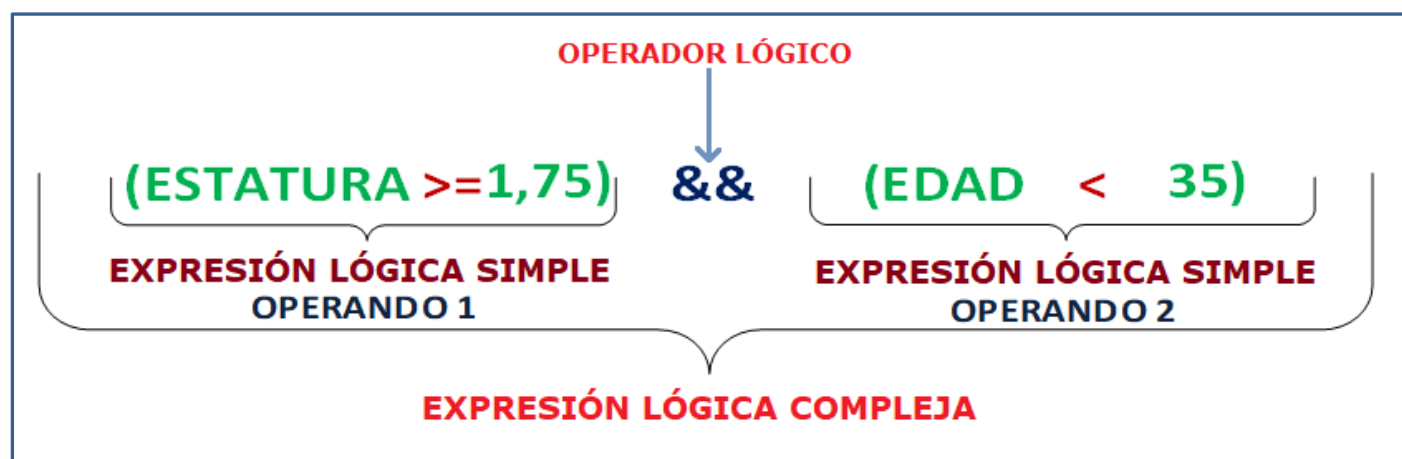




TABLA DE VERDAD – OPERADOR LÓGICO AND (&&)		
OPERANDO 1	OPERANDO 2	OPERANDO 1 && OPERANDO 2 RESULTADO
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	FALSO
FALSO	VERDADERO	FALSO
FALSO	FALSO	FALSO

Una expresión lógica compleja con el operador lógico **AND** solo es Verdadera cuando ambos operandos son verdaderos. En cualquier otro caso, el resultado será Falso.

Ejemplos de uso Operador Lógico AND (&&):



El operador lógico **AND** es esencial cuando se necesita verificar que todas las condiciones (*expresiones lógicas*) sean verdaderas antes de ejecutar una acción específica. Este operador es clave en la construcción de expresiones lógicas complejas y en la toma de decisiones dentro de un algoritmo.

OPERADOR LÓGICO OR (||)

El operador lógico **OR**, comúnmente representado como **||**, se utiliza para combinar dos o más expresiones lógicas. Devuelve verdadero si al menos uno de los operandos (*expresiones lógicas*) es Verdadero. Si todas las expresiones lógicas combinadas son Falsos (*false*), el resultado de la expresión lógica completa será Falso.

El término "**O**" expresa la disyunción, que indica situaciones alternativas donde una, algunas o todas las opciones pueden ser Verdaderas.

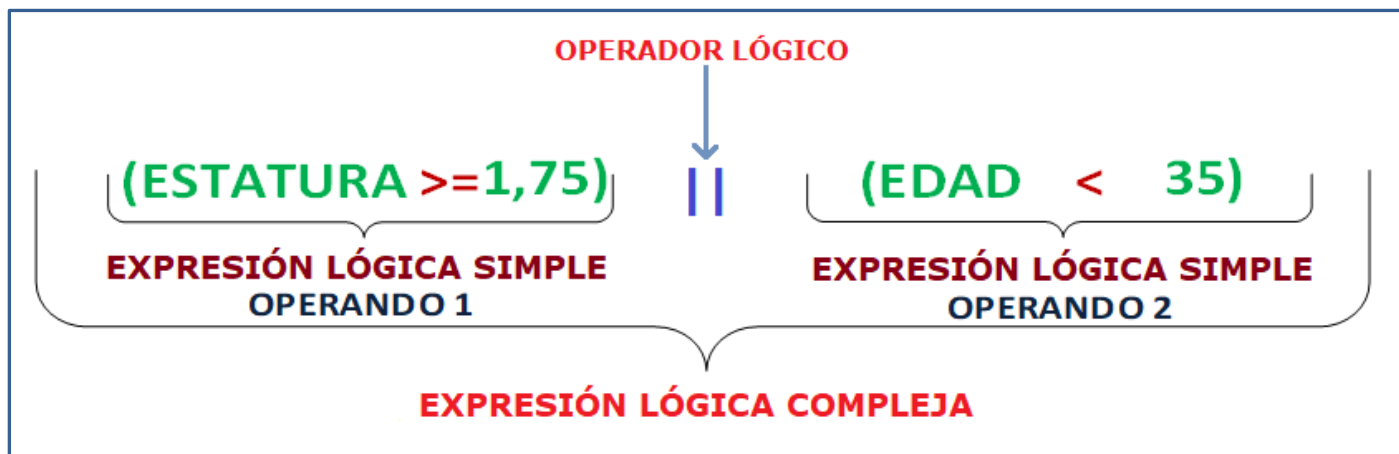
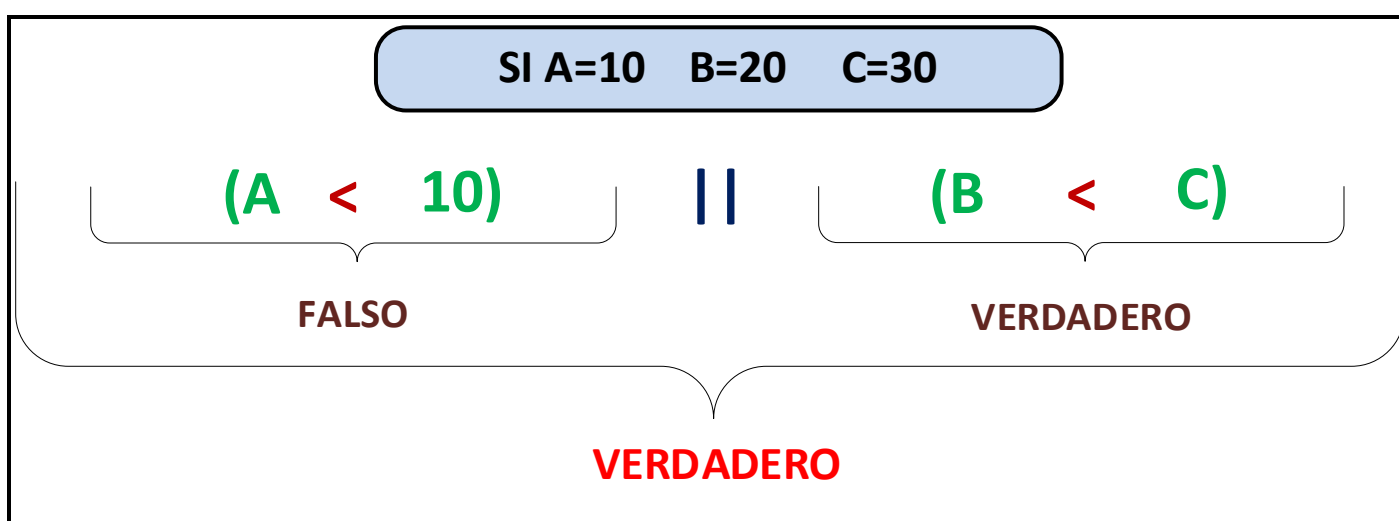


TABLA DE VERDAD – OPERADOR LÓGICO OR (||)

OPERANDO 1	OPERANDO 2	OPERANDO 1 OPERANDO 2 RESULTADO
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO
FALSO	VERDADERO	VERDADERO
FALSO	FALSO	FALSO

Una expresión lógica compleja con el operador lógico **OR** solo es Falso cuando ambos operandos son Falsos. En cualquier otro caso, el resultado será Verdadero.

Ejemplo de uso Operador Lógico OR (||):



El operador lógico **OR** es fundamental en situaciones donde se requiere que al menos una condición (*expresión lógica*) sea verdadera para que una acción se realice.



Este operador es clave para manejar condiciones alternativas y definir criterios más flexibles, permitiendo que una acción se ejecute dentro de un algoritmo cuando existen múltiples caminos posibles.

OPERADOR LÓGICO NOT (!)

El operador lógico **NOT**, comúnmente representado como "!", se utiliza para negar el valor de una expresión lógica. Si la expresión es Verdadera (*true*), el operador **NOT** la convierte en Falso (*false*), y viceversa.

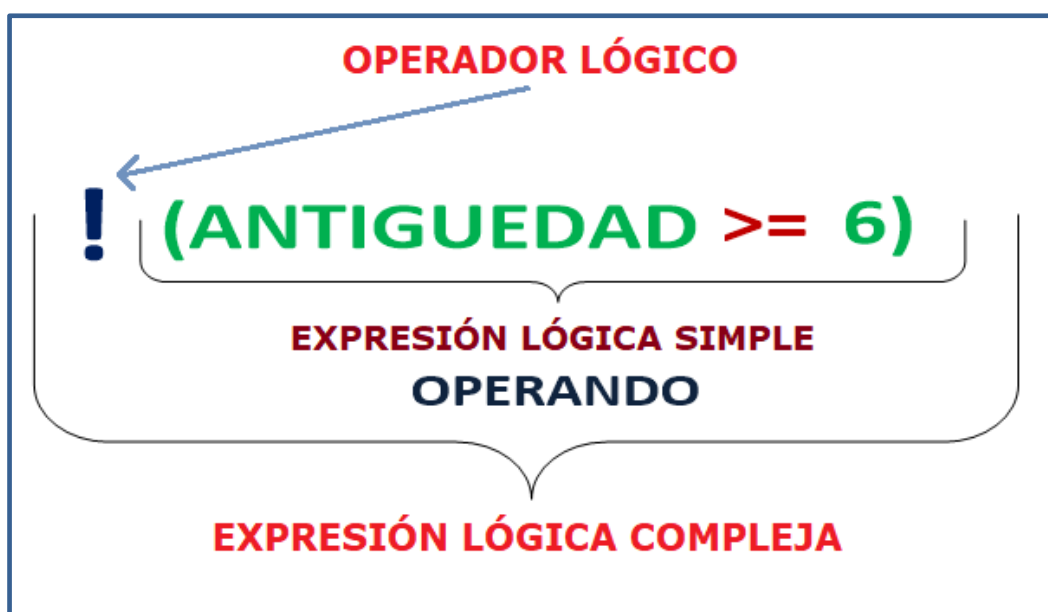
El operador **NOT** invierte el valor de la expresión lógica a la que se aplica:

- Si la expresión lógica es Verdadera, el operador **NOT** devuelve Falso.
- Si la expresión lógica es Falso, el operador **NOT** devuelve Verdadero.

TABLA DE VERDAD – OPERADOR LÓGICO NOT (!)

OPERANDO	! OPERANDO RESULTADO
VERDADERO	FALSO
FALSO	VERDADERO

Ejemplo de uso Operador Lógico NOT (!):



El operador lógico **NOT** es esencial para invertir el valor de verdad de una expresión lógica, permitiendo la construcción de condiciones que manejan casos donde la ex-



presión original no es verdadera. Es un operador fundamental en la lógica de programación.

PRIORIDAD (PRECEDENCIA) DE LOS OPERADORES EN GENERAL

Tabla que muestra la prioridad (precedencia) de los operadores aritméticos, relacionales y lógicos en general, desde la mayor prioridad hasta la menor:

Nivel	Operador	Tipo	Asociatividad	Descripción
1	()	Agrupación	N/A	Agrupar expresiones, evaluar primero
2	**	Aritmético	Derecha a izquierda	Exponenciación (en algunos lenguajes)
3	*, /, %	Aritmético	Izquierda a derecha	Multiplicación, división, Resto
4	+, -	Aritmético	Izquierda a derecha	Suma, resta
5	<, <=, >, >=	Relacional	Izquierda a derecha	Comparaciones de menor, mayor, etc.
6	==, !=	Relacional	Izquierda a derecha	Igualdad y desigualdad
7	! (NOT)	Lógico	Derecha a izquierda	Negación lógica (NOT)
8	&& (AND)	Lógico	Izquierda a derecha	Y lógico (AND)
9	(OR)	Lógico	Izquierda a derecha	O lógico (OR)

- Los paréntesis () son utilizados para cambiar el orden natural de la evaluación de las expresiones.
- ** para la exponenciación no es universal y depende del lenguaje. En muchos lenguajes, la exponenciación se realiza mediante funciones como pow().