

Algoritmos y Estructuras de Datos

Variables y Expresiones.



Contenidos del documento:

- ¿Qué son las variables y de qué tipo disponemos?
- ¿Qué son las expresiones lógicas ?
- ¿Qué son las expresiones aritméticas?
- ¿Qué son las expresiones combinadas?

Variables:

Una variable es como una etiqueta o un nombre que le damos a una ubicación en la memoria de la computadora para almacenar y manipular datos. Imaginá que tenés varias cajas donde podés guardar cosas. Cada caja tiene una etiqueta para que sepas qué hay dentro. En programación, las variables son esas etiquetas, y el contenido de la caja son los datos que almacenamos.

Características clave de las variables:

- I. **Nombre Único:** Cada variable tiene un nombre que la distingue de las demás. Este nombre debe seguir ciertas reglas (por ejemplo, no empezar con un número, no contener espacios en algunos lenguajes, etc.).
- II. **Tipo de Dato:** Una variable está asociada a un tipo de dato específico, que define qué tipo de información puede almacenar (por ejemplo, números enteros, números con decimales, texto, valores booleanos - verdadero/falso, etc.) y qué operaciones se pueden realizar con ella.
- III. **Almacenamiento de un Valor:** En un momento dado, una variable contiene un único valor de acuerdo con su tipo de dato. Sin embargo, a diferencia de las constantes*, este valor puede cambiar durante la ejecución del programa.
- IV. **Ubicación en Memoria:** Internamente, la computadora asigna un espacio en su memoria para cada variable. El nombre de la variable nos permite acceder a ese espacio sin tener que recordar la dirección de memoria exacta.

* Una constante es un valor fijo que se utiliza en un programa y que no puede ser modificado una vez que se le ha asignado un valor inicial. Pensá en ella como un dato que permanece inalterable durante toda la ejecución del programa.

Variables Aritméticas:

Almacenan valores numéricos que se utilizan en cálculos matemáticos.

Tipos de Datos Comunes:

- Entero (**int, integer**): Representa números completos, positivos o negativos, sin parte decimal (ej: -5, 0, 10).
- De Punto Flotante (**float, double, real**): Representa números que pueden tener una parte decimal (ej: 3.14, -0.5, 2.0).

Variables Lógicas (o Booleanas):

Almacenan valores de verdad que representan condiciones o estados. Sólo pueden tomar dos valores: **Verdadero** (True) o **Falso** (False). Su tipo de dato es Booleano.

Son fundamentales en la toma de decisiones dentro de un programa, controlando el flujo de ejecución mediante operadores lógicos (**AND, OR, NOT**) y relaciones (**>, <, ==, !=**).

Variables de Carácter:

Almacenan un único carácter alfanumérico y su tipo de dato es **char**. Puede ser cualquier letra (mayúscula o minúscula), dígito (0-9) o símbolo especial (ej: 'a', '7', '\$').

Variables de Cadena:

Almacenan una secuencia de cero o más caracteres. Su tipo de Dato es **string**. Puede ser cualquier combinación de letras, números y símbolos, formando palabras, frases o textos completos (ej: "Hola", "Esto es un texto", "123", "").

Cada tipo de variable está diseñado para almacenar un tipo específico de información, lo que permite a la computadora organizar y manipular los datos de manera eficiente. El tipo de dato de una variable determina qué clase de valores puede contener y **qué operaciones son válidas para ella**.

Expresiones:

Una expresión es una combinación de constantes, variables (que pueden cambiar su valor), operadores y a veces llamadas a funciones, que al ser evaluada produce un resultado.

Expresiones Lógicas:

Estas expresiones manipulan constantes y variables lógicas utilizando operadores lógicos. El resultado de una expresión lógica siempre es un valor booleano: verdadero o falso.

Operadores Lógicos Fundamentales:

- ❖ **AND (Y)**: and o &&. La expresión es verdadera solo si ambas partes de la expresión son verdaderas. (Ejemplo: **(5 > 3) and (2 < 4)** es **verdadero** porque ambas condiciones son verdaderas).
- ❖ **OR (O)**: or o ||. La expresión es verdadera si al menos una de las partes de la expresión es verdadera. (Ejemplo: **(5 < 3) or (2 < 4)** es **verdadero** porque la segunda condición es verdadera).
- ❖ **NOT (NO)**: not o !. Invierte el valor lógico de la expresión que le sigue. Si la expresión es verdadera, not la hace falsa, y viceversa. (Ejemplo: **not (5 < 3)** es **verdadero** porque $5 < 3$ es falso).

- ❖ Operadores Relacionales (que a menudo se usan en expresiones lógicas): Estos operadores comparan valores y producen un resultado booleano.

- Mayor que: $>$ (ejemplo: **5 > 3** es **verdadero**)
- Menor que: $<$ (ejemplo: **2 < 7** es **verdadero**)
- Mayor o igual que: \geq (ejemplo: **5 \geq 5** es **verdadero**)
- Menor o igual que: \leq (ejemplo: **3 \leq 4** es **verdadero**)
- Igual a: \equiv (ejemplo: **10 \equiv 10** es **verdadero**)
- Diferente de: \neq (dependiendo del lenguaje) (ejemplo: **5 \neq 2** es **verdadero**)

Ejemplos de Expresiones Lógicas:

- ❖ **(edad \geq 18) and (tiene_licencia \equiv verdadero)** (verifica si una persona es mayor de edad y tiene licencia)
- ❖ **(temperatura > 30) or (hay_humedad \equiv verdadero)** (verifica si hace mucho calor o hay humedad)
- ❖ **not (llueve \equiv falso)** (es equivalente a llueve \equiv verdadero)
- ❖ **(nombre \equiv "Juan") and (apellido \equiv "Pérez")** (verifica si el nombre y el apellido coinciden)

Expresiones Aritméticas:

Estas expresiones manipulan constantes y variables **numéricas** utilizando operadores aritméticos. El resultado de una expresión aritmética siempre es un valor numérico.

- ❖ Operadores Aritméticos Básicos:
 - Suma: $+$ (ejemplo: **5 + 3** da como resultado **8**)
 - Resta: $-$ (ejemplo: **10 - 4** da como resultado **6**)
 - Multiplicación: $*$ (ejemplo: **2 * 7** da como resultado **14**)
 - División: $/$ (ejemplo: **15 / 3** da como resultado **5**)
 - Módulo (resto de la división entera): $\%$ o **mod** (ejemplo: **17 % 5** da como resultado **2**, porque 17 dividido por 5 es 3 con un resto de 2)
- ❖ Otros Operadores Aritméticos (dependiendo del lenguaje o contexto):

- Potenciación: \wedge o $**$ (ejemplo: $2 \wedge 3$ o $2 ** 3$ da como resultado **8**)
- División entera: $//$ (ejemplo: $17 // 5$ da como resultado **3**, se queda con la parte entera del resultado)

Otros ejemplos de expresiones aritméticas:

- ❖ **$3 + 7 * 2$** (El orden de las operaciones es importante: primero la multiplicación, luego la suma, resultando en **$3 + 14 = 17$**)
- ❖ **$(10 - 5) / 2$** (Los paréntesis alteran el orden de las operaciones: primero la resta, luego la división, resultando en **$5 / 2 = 2.5$**)
- ❖ **$x + 5$** (donde **x** es una variable que contiene un valor numérico)

Expresiones Combinadas:

Estas expresiones combinan expresiones aritméticas y lógicas utilizando tanto operadores aritméticos como lógicos (y a veces operadores relacionales). El resultado final de una expresión combinada suele ser un valor lógico (verdadero o falso), pero las partes internas pueden involucrar cálculos numéricos.

La clave para entender las expresiones combinadas es recordar la precedencia de los operadores. Generalmente, los operadores aritméticos se evalúan antes que los operadores relacionales, y los operadores relacionales se evalúan antes que los operadores lógicos. Sin embargo, los paréntesis se pueden usar para alterar este orden.

Ejemplos de Expresiones Combinadas:

- ❖ **$(x + 5 > 10) \text{ and } (y * 2 < 20)$**
 - Primero se evalúan las expresiones aritméticas: $x + 5$ y $y * 2$.
 - Luego se evalúan las expresiones relacionales: el resultado de $x + 5$ se compara con 10, y el resultado de $y * 2$ se compara con 20.
 - Finalmente, los dos resultados booleanos se combinan con el operador **and**. La expresión completa será verdadero solo si ambas comparaciones son verdaderas.

-
- ❖ **((a * b) - c >= 0) or (d / 2 == 5)**
 - Se realizan las operaciones aritméticas dentro de los paréntesis: $a * b$ y luego se resta c , y por otro lado se calcula $d / 2$.
 - Luego se realizan las comparaciones relacionales: el resultado de la primera parte se compara con 0, y el resultado de la segunda parte se compara con 5.
 - Finalmente, los dos resultados booleanos se combinan con el operador or. La expresión completa será verdadero si al menos una de las comparaciones es verdadera.

 - ❖ **not (contador < 10 and promedio < 7)**
 - Primero se evalúa la expresión lógica dentro del paréntesis: $\text{contador} < 10$ y $\text{promedio} < 7$ se combinan con and.
 - Finalmente, el resultado booleano de la expresión dentro del paréntesis se invierte con el operador not.

