



## Estructura de Datos

En el capítulo anterior aprendimos a reconocer un problema y a especificar todos los elementos del mismo. Seguramente estarás recordando que, dentro de esos elementos que identificamos, se encontraban **los Datos de Entrada y los Datos de Salida**.

### Operadores Aritméticos

Consta de los operadores fundamentales de la aritmética con el agregado de la potencia y el módulo o residuo, que devuelve el resto entero que se produce al realizar un cociente entre dos números enteros.

Operadores Aritméticos				
Operador	Nombre	Ejemplo	Resultado	Descripción
+	Suma	12 + 3	15	Devuelve la suma de dos expresiones
-	Resta	12 - 3	9	Devuelve la resta de dos expresiones
*	Multiplicación	12 * 3	36	Devuelve el producto de dos expresiones
/	División	12 / 3	4	Devuelve el cociente de dos expresiones
^	Potenciación	12 ^ 3	1728	Devuelve la potencia entre dos expresiones
%	Módulo o Residuo	12 % 3	0	Devuelve el resto del cociente entre dos enteros

Siguiendo los fundamentos de la aritmética, los operadores de multiplicación, división y módulo tienen mayor prioridad que la suma y resta. Así, por ejemplo, la expresión  $2 + 3 * 4$  devuelve como resultado 14, pues primero se evalúa el término  $3 * 4$  que resulta 12, y luego se realiza la suma entre 2 y 12.

### Operador de asignación

Hasta ahora hemos visto como declarar variables, las reglas de nombrado y la convención más difundida. Pero recordemos que hasta ahora las variables no están inicializadas, es decir, su valor es indefinido.

Hay dos maneras de establecer un valor a una variable. La primera que veremos es a través del operador de asignación.

El operador de asignación de PSeInt es el  $=$ , aunque por defecto también puede usarse el  $<-$ , que es compuesto, dado que se conforma de dos caracteres. Es muy importante que no dejes un espacio entre ellos.

La sintaxis para asignar un valor a una variable es la siguiente:



**<variable>** = <expresión>;

La palabra **<variable>** indica que en ese lugar debe ir el identificador de una variable definida.

= es el operador de asignación.

La palabra **<expresión>** indica que en ese lugar debe ir una expresión válida, de un tipo de dato compatible con lo que espera guardar la variable.

El ; es obligatorio e indica que finalizó la instrucción.

## Operadores relacionales

Son aquellos que permiten comparar expresiones. Si la evaluación es correcta, la máquina retorna VERDADERO, de lo contrario, retorna FALSO.

Operadores relacionales			
Operador	Nombre	Ejemplo	Resultado
<	Menor que	4 < 5	V
<=	Menor o igual que	5 <= 5	V
>	Mayor que	4 > 5	F
>=	Mayor o igual que	5 >= 5	V
==	Igual que	4 == 5	F
!=	Distinto que (para números)	4 != 5	V
<>	Distinto que (para cadenas)	"PEZ" <> "pez"	V

Es muy común al principio confundir el operador de asignación (=) con el de comparación (==). Puede dilucidar que, al poner, por ejemplo:



Escribir `2 = 2;`

en realidad, se quiso comparar dos números entre sí, por lo que devolverá un valor lógico. Pero en los lenguajes formales eso sería un error. Estaríamos queriendo asignar un 2 a un 2. No tiene sentido. Tenemos que prestar atención desde un comienzo y notar el contexto del código, para saber si se trata de una asignación (con `=`) o una comparación (con `==`).