



# Programación (Fundamentos y Logica)

# Lección 2:

- \* VARIABLES Y CONSTANTES
- \* TIPOS DE DATOS EN PSEINT
- \* ¿CÓMO CREO UNA VARIABLE?
- \* EJERCICIO CREAR VARIABLE
- \* TIPOS DE INSTRUCCIONES
- \* INSTRUCCIONES DE ESCRITURA O SALIDA
- \* EJERCICIO ESCRIBIR
- \* INSTRUCCIONES DE LECTURA
- \* EJERCICIO LECTURA Y ESCRITURA
- \* ¿CÓMO ASIGNAMOS VALORES A LAS VARIABLES?
- \* OPERADORES
- \* OPERADORES ALGEBRAICOS
- \* EJERCICIO VARIABLES
- \* EJERCICIO DETECCIÓN DE ERRORES 1

## ## VARIABLES Y CONSTANTES

Los programas de computadora necesitan información para la resolución de problemas. Esta información puede ser un número, un nombre, etc.

Para nosotros poder guardar esta información en algún lugar y que no esté “suelta”, para no perderla o poder acceder a ella cuando lo necesitemos es crucial. Para solucionar esto, vamos a guardar la información en algo llamado, variables y constantes. Las variables y constantes vendrían a ser como pequeñas cajas, que guardan algo en su interior, en este caso información.

Estas, van a contar como previamente habíamos mencionado, con un identificador, un nombre que facilitará distinguir unas de otras y nos ayudará a saber que variable o constante es la que contiene la información que necesitamos.

Dentro de toda la información que vamos a manejar, a veces, necesitaremos información que no cambie. Tales valores son constantes. De igual forma, existen otros valores que necesitaremos que cambien durante la ejecución del programa; esas van a ser nuestras variables.

Una variable es un objeto o tipo de datos cuyo valor puede cambiar durante el desarrollo del algoritmo o ejecución del programa. Dependiendo del lenguaje, hay diferentes tipos de variables, tales como enteras, reales, carácter, lógicas y de cadena. Una variable que es de un cierto tipo puede tomar únicamente valores de ese tipo.

Una variable de carácter, por ejemplo, puede tomar como valor sólo caracteres, mientras que una variable entera puede tomar sólo valores enteros.

Ejemplo: una variable que guardará el resultado de un cálculo, este valor puede cambiar, en alguna parte de nuestro programa.

Una constante es un dato, que al igual que la variable, puede ser de diferentes tipos como enteras, reales, carácter, lógicas y de cadena.

Estas, también guardan solo valores de ese tipo, pero, permanecen sin cambios durante todo el desarrollo del algoritmo o durante la ejecución del programa. Ejemplo: el valor de Pi

## ## TIPOS DE DATOS EN PSEINT

Las variables y constantes como previamente habíamos mencionado, van a guardar información dependiendo del tipo de dato que le digamos que guarde esa variable. Por ejemplo, si digo que mi variable va a guardar números enteros, significa que el tipo de dato de esa variable es entero.

Los tipos de datos que podemos usar son:

✓ Entero: solo números enteros.

✓ Real: números con cifras decimales. Para separar decimales se utiliza el punto. Ejemplo:  
3.14

✓ Carácter: cuando queremos guardar un carácter. Los Caracteres se encierran entre comillas simples. un carácter (unidimensional): 'a', 'A'.

✓ Lógico: cuando necesitamos guardar una expresión lógica (verdadero o falso)

✓ Cadena: cuando queremos guardar cadenas de caracteres. Las Cadenas se encierran entre comillas dobles. una cadena (multidimensional): "esto es una cadena", "hola mundo"

Notas:

✓ Cadena y Carácter son términos equivalentes, no genera error que las escribamos indistintamente.

✓ El plural de Carácter es Caracteres o Cadena.

## ## ¿CÓMO CREO UNA VARIABLE?

A la hora de crear nuestra variable, vamos a tener que darle un identificador, por lo que usaremos las reglas vistas en el apartado de identificadores, y el tipo de dato que necesitamos que guarde. Para esto vamos a utilizar la palabra reservada Definir. La instrucción definir permite explicitar el tipo de una o más variables.

## ## EJERCICIO CREAR VARIABLE

Vamos a poner en práctica lo que acabamos de ver, vamos a crear en PseInt, una variable de cada tipo de dato posible.

### Palabras Reservadas

Palabras que dentro del lenguaje significan la ejecución de una instrucción

determinada, por lo que no pueden ser utilizadas con otro fin. En Pseudocódigo, las palabras reservadas aparecen de color azul.

Por ejemplo, la palabra Algoritmo y FinAlgoritmo.

## ## TIPOS DE INSTRUCCIONES

Las instrucciones –acciones– básicas que se pueden implementar de modo general en un algoritmo y que esencialmente soportan todos los lenguajes son las siguientes:

### INSTRUCCIONES DE INICIO/FIN

Son utilizadas para delimitar bloques de código. Por ejemplo, Algoritmo y FinAlgoritmo.

## ## INSTRUCCIONES DE ESCRITURA O SALIDA

Ayer estuvimos trabajando un poco con esto, ahora sí, te explicaremos cómo funcionan estas instrucciones.

Se utilizan para escribir o mostrar mensajes o contenidos de las variables en un dispositivo de salida.

La salida puede aparecer en un dispositivo de salida (pantalla, impresora, etc.). La operación de salida se denomina escritura (escribir).

En la escritura de algoritmos la acción escritura se representa con el siguiente formato: pEscritura o salida. La instrucción Escribir permite mostrar información y valores de variables en la interfaz gráfica de usuario.

**Escribir** <expr1> , <expr2> , ... , <exprN>

Esta instrucción imprime en la interfaz gráfica de usuario (en este caso en la pantalla) los valores obtenidos al evaluar N expresiones, el valor de

un variable o de u mensaje. Dado que pued incluir una o más expresiones, mostrará uno o más valores.

Ejemplo:

Algoritmo EjemploEscribir

Definir num Como Entero

Escribir "Hola Mundo"

Escribir 2+2

num=10

Escribir num

FinAlgoritmo

En este ejemplo de escribir, vemos que nuestro primer escribir muestra un mensaje o cadena, que va entre comillas dobles, después nuestro segundo escribir muestra el resultado de una suma de dos números y nuestro último escribir, muestra el valor de una variable de tipo entero a la que se le asignó un valor previo.

Con el escribir también podemos mostrar variables o valores con un mensaje previo, para esto vamos a concatenar nuestra variable, usando una coma o un espacio en blanco, con un mensaje entre comillas.

Escribir "Mensaje entre comillas" variable

Escribir "La suma de los números es:" suma

```

## ## EJERCICIO ESCRIBIR

Escribir un algoritmo en el cual se muestre nuestro nombre completo en la interfaz gráfica de PseInt.

## ## INSTRUCCIONES DE LECTURA

Los cálculos que realizan las computadoras requieren, para ser útiles la entrada de los datos necesarios para ejecutar las operaciones que posteriormente se convertirán en resultados, es decir, salida.

Las operaciones de entrada permiten leer datos de un dispositivo de entrada y asignarlos a determinadas variables.

Esta entrada se conoce como operación de lectura (leer). Los datos de entrada se introducen al procesador mediante dispositivos de entrada (teclado, tarjetas perforadas, unidades de disco, etc.).

Lectura o entrada

La instrucción Leer permite ingresar información por teclado al usuario a través de la interfaz gráfica o ambiente de Pseint. Que se mostrará al correr nuestro algoritmo

Leer <variable1>, <variable2>, ..., <variableN>

Ejemplo:

Algoritmo EjemploLeer

```
    Definir num Como Entero
    leer num
FinAlgoritmo
```
```

En este ejemplo definimos una variable de tipo entero llamada num y le asignamos un valor a través de la instrucción Leer.

## ## EJERCICIO LECTURA Y ESCRITURA

Escribir un algoritmo en el cual el programa nos pida nuestro nombre completo, lo aloje en una variable de tipo Cadena llamada nombreCompleto y después mostrarlo usando la instrucción Escribir

## ## ¿CÓMO ASIGNAMOS VALORES A LAS VARIABLES?

La instrucción de asignación permite almacenar un valor en una variable (previamente definida).

Esta es nuestra manera de guardar información en una variable, para utilizar ese valor en otro momento. Se puede realizar de dos maneras, con el signo igual o una flecha:

<variable> <- <expresión>

<variable> = <expresión>

expresión es igual a una expresión matemática o lógica, a una variable o constante.

Al ejecutarse la asignación, primero se evalúa la expresión de la derecha y luego se asigna el resultado a la variable de la izquierda. El tipo de la variable y el de la expresión deben coincidir

## ## EJERCICIO DEFINIR

Escribe un algoritmo definiendo la variable nombre como cadena, asigna allí tu información y luego muéstralo por pantalla escribiendo la variable.

## ## OPERADORES

Este pseudolenguaje dispone de un conjunto básico de operadores que pueden ser utilizados para la construcción de expresiones más o menos complejas.

## ## OPERADORES ALGEBRAICOS

Los operadores algebraicos o también conocidos como operadores aritméticos. Realizan operaciones aritméticas básicas: suma, resta, multiplicación, división, potenciación y modulo para datos de tipo numérico tanto enteros como reales. Estas operaciones son binarias porque admiten dos operandos.

Operador	Significado	Resultado
Algebraicos		
+	Suma	suma = 2 + 2
-	Resta	resta = 10 - 4
*	Multiplicación	multiplicación = 10 * 2
/	División	división = 9 / 3
^	Potenciación	potencia = 10 ^ 2
% o MOD	Módulo (resto de la división entera)	resto = 4 % 2

Reglas de prioridad:

Las expresiones que tienen dos o más operadores requieren unas reglas matemáticas que permitan determinar el orden de las operaciones, se denominan reglas de prioridad y son:

1. Las operaciones que están encerradas entre paréntesis se evalúan primero. Si existen diferentes paréntesis anidados (interiores unos a otros), las expresiones más internas se evalúan primero.
2. Las operaciones aritméticas dentro de una expresión suelen seguir el siguiente orden de prioridad:

✓ operador ( )  
✓ operadores unitarios (potenciación),  
✓ operadores \*, /, % (producto, división, módulo) ✓ operadores +, - (suma y resta).

En caso de coincidir varios operadores de igual prioridad en una expresión o sub expresión encerrada entre paréntesis, el orden de prioridad en este caso es de izquierda a derecha, y a esta propiedad se denomina asociatividad.

## ## EJERCICIO VARIABLES

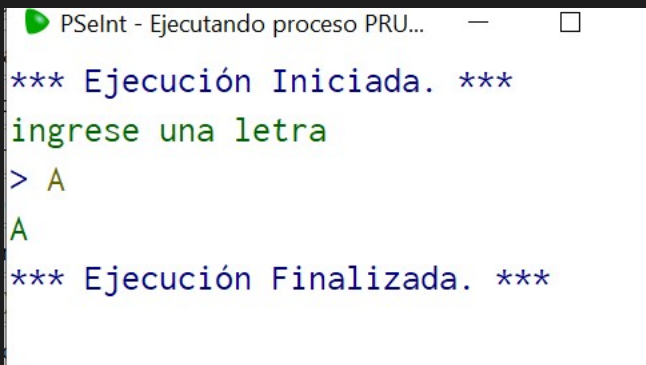
Define dos variables que guarden números enteros, defina una tercera variable donde aloje el resultado se sumarán ambas variables. Comente su código indicando qué finalidad tiene cada línea.

### ## EJERCICIO DETECCIÓN DE ERRORES 1

¿Puedes corregir esta porción de código para que cumpla el resultado esperado?

```
Algoritmo Prueba
Definir letra Como Entero
Escribir ingrese una letra
letra
FinAlgoritmo
````
```

¿CUÁL ES EL RESULTADO A LOGRAR?



```
PSeInt - Ejecutando proceso PRU...
*** Ejecución Iniciada. ***
ingrese una letra
> A
A
*** Ejecución Finalizada. ***
```