



ELEMENTOS BÁSICOS DE UN ALGORITMO

ING. GIANKARIS G. MORENO R., M.SC



ESTRUCTURA DE UN ALGORITMO EN PSEUDOCÓDIGO

- Formato general de escritura de un algoritmo a través de un pseudocódigo.

```
Algoritmo Nombre_Algoritmo
{
    //Bloque Declarativas de Constantes – si aplica
    declaración de constantes;

    //Bloque Declarativas de Variables
    declaración de variables;

    // Bloque de Instrucciones
    sentenciaI;
    :
    sentenciaN;
}
```

REGLAS DE ESCRITURA DE UN ALGORITMO EN PSEUDOCÓDIGO

- El nombre del algoritmo debe iniciar con la palabra Algoritmo y el nombre dado al algoritmo
- Para definir bloques de instrucciones, utilizar la llave de apertura ({) para indicar el inicio y la llave de cierre (}) para finalizar
- Cada instrucción debe terminar con punto y coma (;).
- Los comentarios se realizarán de dos formas:
 - // Comentario de una sola línea
 - /* */ para comentario de bloques.
- Comentar o Identificar las áreas básicas de la estructura de un algoritmo.
 - Bloque Declarativas de Constantes
 - Bloque Declarativas de Variables
 - Bloque de Instrucciones
- Las instrucciones de leer, imprimir, estructuras de alternativas y repetitivas se escribirán en minúscula.

REGLAS DE ESCRITURA DE UN ALGORITMO EN PSEUDOCÓDIGO (BUENAS PRÁCTICAS)

- El nombre del algoritmo inicia con letra en mayúscula y si está compuesta por dos o más palabras, la primera letra de cada palabra debe ser en mayúscula y sin espacios en blanco. *Estilo PascalCase*.
- Utilizar minúscula para los nombres de variables y mayúscula cerrada para las constantes.
- Los nombres de las funciones deben iniciar en minúscula y si está compuesta por dos o más palabras, la primera letra de la primera palabra ira en minúscula y luego la primera letra de cada palabra debe ser en mayúscula y sin espacios en blanco. *Estilo camelCase*.
- Los nombres de funciones deben ser iniciar con un verbo para indicar la acción de la función.
- Indentar las instrucciones.

ELEMENTOS BÁSICOS

Identificadores

Tipos de datos

Operadores aritméticos, expresiones y jerarquía

Asignación

Entrada/Salida de datos en Pseudocódigo



IDENTIFICADORES

- Se definen como los nombres dados a cualquier elemento a crear por el usuario dentro de un programa. Elementos tales como:
 - Variables
 - Constantes
 - Arreglos
 - Archivos
 - nombres de funciones o de subprogramas
- Los nombres de identificadores se forman utilizando las letras mayúsculas y minúsculas, los números y los caracteres especiales \$ y trazo (_).

IDENTIFICADORES

- Reglas
 - Debe comenzar siempre con una letra.
 - A partir del segundo carácter puede ser una letra, un carácter especial o un número.
 - Como máximo debe manejar 10 caracteres sus nombres de identificadores
 - Las palabras reservadas no deben utilizarse como nombre de identificador.
- Ejemplos válidos de nombres de identificadores:
campo, salario, peso, Pobl_a_99, cuentaB, raices22
- Ejemplos Inválidos:
GANANCIA 3452 , “rangoFinal”, 2334Epsilo

VARIABLES

- Son elementos cuyo valor puede ser cambiado a lo largo de la ejecución de un programa o algoritmo. Toda variable antes de ser utilizada debe ser declarada y la declaración involucra un nombre de identificador, tipo de dato y valor.

Formato:

tipo_dato nombre_identificador;

Ejemplo:

entero a, b, C;

flotante tarifa, itbm, radioA;

caracter car, var, signo, resp3;

cadena nombre;

CONSTANTES

- Son objetos cuyo valor permanece invariable a lo largo de la ejecución de un algoritmo o programa, Se recomienda escribir las constantes en cierre de mayúscula para diferenciarla de una variable en el programa o algoritmo.

Formato:

NOMBRE_IDENTIFICADOR = valor;

Ejemplos:

EXPO = 8998

OPC = '@'

SW = CIERTO

CAMBIAR = "2-344-566"



TIPOS DE DATOS

- Son los valores que pueden tomar los objetos definidos por el usuario.
- Estos tipos de datos pueden ser simples o compuestos. Los tipos de datos simples solo pueden almacenar un valor, mientras que los compuestos pueden almacenar más de un valor a la vez.
- Los tipos de datos son:
 - entero
 - flotante
 - carácter
 - cadena

TIPOS DE DATOS

entero: es un valor numérico que puede tener tomar como máximo 5 dígitos, no excederse de un valor predefinido de 32,627.

- Ejemplo: 345, 978, 8000, 1021

flotante: son valores numéricos que representan una parte entera y otra fraccionaria. Existen dos maneras de representarlos, punto fijo y punto flotante.

- Ejemplo: 69.745, -0.1542, 1.32, 3.1416

TIPOS DE DATOS

caracter: son aquellos que solo pueden representar un solo caracter, de los 255 caracteres de UNICODE. Los tipos de datos caracter deben estar encerrados entre apóstrofes (').

- Ejemplo: '*', 'm', 'F', '@'

cadena: Son aquellos que están compuestos por valores alfanuméricos. Se expresan encerrados entre comillas (").

- Ejemplo: "Marcos", "8-777-2121", "Axel_1423"



OPERADORES ARITMÉTICOS, EXPRESIONES Y JERARQUÍA

- Operadores Aritméticos: son los que indican el tipo de operación que debe ser llevada a cabo con los operandos.
- Los operadores aritméticos son:

Operadores	Representación
+	Suma
-	Resta
*	Multiplicación o producto
/	División
**	Potenciación
%	Residuo

OPERADORES ARITMÉTICOS, EXPRESIONES Y JERARQUÍA

- Expresiones: se define como un valor, una constante, una variable, una función especial, la llamada a una función o el conjunto de todos los elementos antes mencionados unidos a través de operadores.

Ejemplos:

- Un valor: 315, "Juan José"
- Una constante: TANGENTE=0.9999 , LIMITE = 25
- Una variable: puntos, perímetro, área
- Una función especial : Truncar(x), redondear (x), sqrt(x)
- Un conjunto de elementos unidos por operadores aritméticos, lógicos o de relación:
 $349 + 345 / 45 * \text{sqrt}(m)$

OPERADORES ARITMÉTICOS, EXPRESIONES Y JERARQUÍA

■ Jerarquía de los operadores

Prioridad	Operador	Descripción
1	()	paréntesis, comenzando por los más internos
2	**	Potenciación
3	/ * %	división, producto y residuo
4	+ -	suma, resta

NOTA: Cuando operadores de igual prioridad aparezcan en una misma expresión deben ser evaluados de izquierda a derecha.

OPERADORES ARITMÉTICOS, EXPRESIONES Y JERARQUÍA

■ Jerarquía de los operadores

Prioridad	Operador	Descripción
1	()	paréntesis, comenzando por los más internos
2	**	Potenciación
3	/ * %	división, producto y residuo
4	+ -	suma, resta

OPERADORES ARITMÉTICOS, EXPRESIONES Y JERARQUÍA

- Evaluar una expresión de acuerdo a la jerarquía de los operadores

$$\frac{b}{c + d}$$

$$y^3 + 4x$$

Primero se debe reescribir la expresión, en expresión de computadoras

$$\underbrace{b / (c + d)}_2$$

Diagram illustrating the evaluation of the expression $b / (c + d)$ using operator precedence. A bracket labeled '1' is placed under the sub-expression $(c + d)$, indicating it is evaluated first. A second bracket labeled '2' is placed under the entire expression $b / (c + d)$, indicating the final division operation.

$$\underbrace{y ** 3}_1 + \underbrace{4 * x}_2$$

Diagram illustrating the evaluation of the expression $y ** 3 + 4 * x$ using operator precedence. A bracket labeled '1' is placed under $y ** 3$, and a bracket labeled '2' is placed under $4 * x$, indicating these two sub-expressions are evaluated first. A third bracket labeled '3' is placed under the entire expression $y ** 3 + 4 * x$, indicating the final addition operation.





¿PREGUNTAS?

