Pseudocódigo

Y

Tipos de datos

Los algoritmos expresados en pseudocódigo son una **herramienta** muy útil en el diseño de soluciones de cualquier grado de complejidad. Y particularmente didácticos, cuando se está haciendo una inducción en el apasionante mundo de la programación.

> **Pseudo:** es una palabra proviene de un vocablo griego que significa falso. La palabra código, se derivó de un vocablo latín que significa, entre otras acepciones, una combinación de signos que tiene un determinado valor de un sistema establecido.

Usualmente, se hace referencia a los algoritmos en pseudocódigo como programación en papel, ya que típicamente las herramientas utilizadas son el papel y el lápiz; y por su puesto el borrador.

Tiene asociado un **léxico** (conjunto de palabras), una **sintaxis** (reglas gramaticales) y una **semántica** precisa (significado), de manera análoga a un lenguaje natural como el castellano.

El uso de tal lenguaje hace relativamente fácil el paso de codificación final del algoritmo (esto es, la traducción a un lenguaje de programación). La ventaja del pseudocódigo es que le permite al programador **concentrarse en la lógica** y en las estructuras de control del algoritmo que quiere diseñar, sin preocuparse de las reglas de un lenguaje específico de programación, que normalmente incluyen infinidad de detalles.

Es también fácil modificar el pseudocódigo si se descubren errores o anomalías en la lógica del programa; además de lo anterior, es fácil su traducción a lenguajes de programación. El pseudocódigo utiliza **palabras reservadas** (similares a sus homónimos en los lenguajes de programación) para representar las acciones y estructuras de control.

El formato general de un algoritmo expresado mediante pseudocódigo es el siguiente:

- Definición de constantes
- Declaración de variables
- Cuerpo del programa
    - Inicio
    - Instrucciones
    - Fin
- Procedimientos/Funciones adicionales

Tipos de datos

Al elaborar un algoritmo utilizamos su simbología y reglas de construcción, pero también existe una parte fundamental e indispensable que son los tipos de datos que maneja, estos pueden ser **simples o estructurados** y están agrupados así:

	Concepto	Ejemplo
SIMPLES	Hace referencia a un **único** valor, ocupa un espacio de memoria.	Numéricos, carácter, lógicos, enumerados y subrango.
ESTRUCTURADOS	Tiene varios componentes, los cuales pueden ser un dato simple ó estructurado, ocupa un **grupo** de espacios de memoria.	Arreglos, cadena de caracteres, registros, conjuntos y archivos.

Para el manejo de datos en un programa es necesario establecer si los valores van a cambiar o no, con la intención de poder declararlos o definirlos de la forma más adecuada. Si el dato no va a cambiar durante el programa, entonces el valor podemos declararlo como **constante**. Si los datos van a cambiar, entonces es necesario definir una **variable** que permita la captura de información y procesamiento dentro del programa realizado.

Expresiones y operadores

Cada expresión toma un valor, que se determina cuando los valores de las constantes y las variables implicadas, son evaluadas en la expresión. Una expresión consta de operandos y operadores. Para realizar operaciones necesitamos de operadores aritméticos y lógicos:

Operaciones Aritméticas:

Nos permiten elaborar operaciones aritméticas entre operandos. El resultado de la expresión es de tipo numérico.

OPERACIONES ARITMÉTICAS			
Operador Aritmético	**Operación**	Ejemplo	**Resultados Obtenidos**
**	Exponenciación	2**3	8
*	Multiplicación	8.45*6	50.7
/	División	17/6	2.83
+	Suma	25.36+11.098	36.458
-	Resta	78.09-13.2	64.89
mod	Módulo (residuo)	13 mod 2	1
div	División entera	19 div 3	6

Para evaluar las expresiones que tienen operadores aritméticos, es indispensable que se lleven a cabo algunas jerarquías de orden de aplicación o reglas de prioridad, primero se ejecuta el operador de jerarquía mayor prioridad al de menor.

Por lo anterior, es necesario conocer la **jerarquía de operadores** aritméticos:

Operación	Operador	Jerarquía
Exponenciación	**	del mayor
Multiplicación, división, módulo, división entera	*,/,mod,div	al
Suma, resta	+,-	menor

Para resolver las expresiones aritméticas, cuando tienen uno o más operandos se establecen algunas **reglas**, las cuales nos permitan determinar el orden de las operaciones:

1. Las operaciones que están encerradas entre paréntesis se elaboran primero. Si existieran otros paréntesis interiores (anidados: uno dentro de otro) las expresiones más internas se evalúan primero.

2. Los operadores aritméticos se aplican tomando en cuenta las jerarquías y de izquierda a derecha.

3. Si coincidieran varios operadores de igual prioridad en una expresión el orden de prioridad será de izquierda a derecha.

Expresiones Booleanas o lógicas:

Se forman combinando constantes lógicas variables lógicas y otras expresiones lógicas, utiliza los operadores lógicos (not, and y or) y los operadores relaciónales. El valor que pueden tomar estas expresiones es el de Verdadero o Falso, se ocupa en las estructuras algorítmica selectivas dependiendo del resultado de la evaluación se toma por un determinado camino.

Operadores Relaciónales:

Son operadores que nos permiten **comparar** dos operandos, de tipo numérico o carácter. El resultado de una expresión con operadores relaciónales es verdadero o falso.

Operador Relacional	Operación
=	Igual que
<>	Diferente a
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que

Operadores Lógicos:

Nos permiten formular condiciones complejas a partir de condiciones simples. Los operadores son:

Operador Lógico	Operación
No	Negación
Y	Conjunción
O	Disyunción

Asignación:

La asignación se utiliza para destinar **valores o expresiones** a una variable, si antes de la asignación la variable tenía un valor almacenado, éste se pierde y conserva el nuevo valor.

Para asignar valores a una variable se debe tomar en cuenta la siguiente **sintaxis**:

Variable = valor o expresión

La asignación se puede aplicar para expresiones aritméticas o lógicas, una variable o una constante.

EDUFUTURO
765 palabras

Referencias

http://dis.unal.edu.co/~programacion/book/modulo1.pdf
http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9621/1/Introduccion_a_la_Programacion.pdf
https://www.editorial-club-universitario.es/pdf/405.pdf