

Cap.II Elementos Básicos de un Algoritmo

Prof. Mitzi Murillo de Velásquez

CONTENIDO

1. Estructura de un algoritmo en Seudocódigo

1.1 Reglas de escritura de un Algoritmo en Seudocódigo

2. Elementos Básicos

2.1 Identificadores

2.2 Variables / Constantes

2.2.1 Tipos de Datos

2.2.2. Operadores Aritméticos, Expresiones y Jerarquía

2.2.3 Asignación

2.2.4 Entrada / Salida de datos en pseudocódigo

ALGORITMO

Es una técnica de solución de problemas, en la cual se escribe una serie de instrucciones paso a paso y que produce resultados a un problema determinado.

- Cuando se plantean programas de computadoras, se utilizan los algoritmos para esquematizar los pasos de solución usando un lenguaje de especificaciones de algoritmos llamado **seudocódigo**, que requieren menos precisión que un lenguaje de programación formal.

Algoritmo

En general, cualquier actividad de la vida cotidiana se puede describir mediante algoritmos. Para empezar a familiarizarnos con ellos, se desarrolla a continuación en ejemplo que pone de manifiesto la necesidad de las características antes mencionadas.

algoritmo Sumar

variables

entero a, b, c

inicio

escribir("Introduzca el primer número (entero): ")

leer(a)

escribir("Introduzca el segundo número (entero): ")

leer(b)

$c \leftarrow a + b$

escribir("La suma es: ", c)

fin

Ejemplo

Si a un experimentado conductor se le preguntase como pone en movimiento su automóvil, el contestaría: “Se pone en marcha el motor y se mete la primera”.

Efectivamente es fácil. Pero ¿Qué ocurriría si el individuo nunca condujo un automóvil?. El resultado no sería muy efectivo debido a que según las instrucciones o pasos impartidos anteriormente, si estuviera en algún cambio, al poner en marcha se hubiese estrellado con lo primero que estuviese adelante o atrás.

La conclusión que se debe sacar de este ejemplo es que el conductor con experiencia no tuvo en cuenta todas las posibilidades que se pueden presentar para obtener el resultado de poner el automóvil en movimiento. Por lo tanto, su algoritmo sería:

Poner en marcha el motor
Meter la primera



Ejemplo

En cambio, el algoritmo más correcto podría ser:

Pisar el embrague con el pie izquierdo

Poner en punto muerto

Dar a la llave de contacto

Pisar el embrague

Meter la primera

Quitar el freno de mano si lo tuviese puesto.

Levantarse lentamente el pie del embrague a la vez que pisa el pedal del acelerador con el pie derecho

¿Por qué es más correcto este algoritmo?

- 1) desglosa el problema en instrucciones simples y concretas, comprensibles para cualquier individuo.**
- 2) indica claramente el orden en que deben ejecutarse dichas instrucciones.**

CARACTERÍSTICAS DE UN ALGORITMO

- El número de operaciones que realiza un algoritmo es finito siempre y cuando sus datos sean adecuados. Por consiguiente, el número de operaciones que necesitamos realizar al ejecutar un algoritmo dependerá de los datos del problema y solamente se conocerá al ejecutar este.

Sus tres características más importantes son:

- a. Debe estar compuesto de acciones bien definidas (ausencia de ambigüedad)
- b. Constar de una secuencia lógica de operaciones,
- c. Debe ser finito.

TODO algoritmo tiene tres partes: entrada, proceso, salida.

a. ALGORITMO

ENTRADA

Ingredientes

la piel de 1 limón
zumo de medio limón
4 huevos
200 g de azúcar
200 g de aceite de girasol
220 g de harina de repostería
1 pizca de sal
1 sobre de levadura química
azúcar
Ingredientes para la crema de limón
2 yemas de huevo
50g de azúcar
30g de harina
300 g de leche
1 limón

Prof. Mitzi Murillo de Velásquez Msc.

PROCESO

¿Cómo hacer magdalenas de limón ?

Incorporamos la mariposa en las cuchillas y rallamos la piel de limón (solo la amarilla para no amargar mucho la receta y sacar solo el sabor del limón, reservamos)

Añadimos las yemas con la leche, la harina y el azúcar. Entonces programaremos el robot en velocidad numero 3 durante 15 minutos. Ponemos después parte de la piel de limón rallada y exprimimos un poco el zumo del limón. Ahora programaremos 11 minutos a 90° y velocidad 2. Con esto terminaremos la crema de limón. Reservamos esta crema para utilizarla más adelante.

Ahora comenzaremos de nuevo haciendo la **masa de las magdalenas**. Añadimos al vaso limpio y seco la otra parte del limón rallado. Incorporamos el zumo de medio limón, el azúcar y los huevos para después programar el robot de cocina a velocidad 3 en frio durante 3 minutos.

Ponemos ahora el aceite en el vaso y mezclamos a velocidad 5 durante 15 minutos.

Añadiremos más tarde la levadura, la harina y mezclaremos nosotros mismos un poco, después incorporamos la sal (ya sabemos que la levadura y la sal se llevan muy mal).

U. T. P

SALIDA

Plato Terminado



Algoritmo

- Qué hacer para ver la película AQUAMAN?

ESTRUCTURA DE UN ALGORITMO EN SEUDOCÓDIGO

Algoritmo Nombre del Algoritmo (PascalCase)

```
{  
  /* BLOQUE DE DECLARATIVAS */  
  ( DECLARACIÓN DE VARIABLES )  
  
  /* BLOQUE DE INSTRUCCIONES */  
  sentencia 1;  
  Sentencia2;  
  Sentencia3;  
  .  
  .  
  sentencia n;  
}
```


REGLAS DE ESCRITURA DE ALGORITMOS

1. El nombre del algoritmo debe iniciar con la palabra Algoritmo y el nombre dado al algoritmo.
2. El nombre del algoritmo utilizará el estilo de escritura Pascal Case
3. Para definir bloques de instrucciones, utilizar la llave de apertura { para indicar el inicio y la llave de cierre } para finalizar.
4. Cada instrucción debe terminar con punto y coma (;).
5. Los comentarios se realizarán de dos formas: // Comentario de una sola línea ó /* */ para comentario de bloques.
6. Comentar o Identificar las áreas básicas de la estructura de un algoritmo.
 1. Bloque Declarativas de Constantes
 2. Bloque Declarativas de Variables ó Declaración de variables
 3. Bloque de Instrucciones ó Cuerpo del algoritmo
7. Apegarse a los formatos establecidos para cada instrucción.

ALGORITMO

Algoritmo SumarEnteros

```
{  
  /* Bloque de Declarativas    */  
  entero a, b, c;  
  /* Bloque de Instrucciones  */  
  imprimir("introduzca el primer número entero: ");  
  leer (a);  
  imprimir ("introduzca el segundo número entero: ");  
  leer (b);  
  c = a + b;  
  imprimir ("La suma es: ", c);  
}.
```


ELEMENTOS BÁSICOS

Los programas constan de una serie de elementos que se combinan entre sí , de acuerdo a las reglas de sintaxis del lenguaje de programación.

Los elementos básicos de un programa son los siguientes :

- Identificadores
- Tipos de datos
- Variables
- Constantes
- Caracteres del lenguaje.
- Operadores
- Expresiones

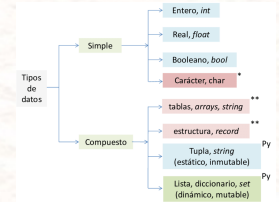
A continuación veremos cada uno de estos elementos desde el punto de vista del **seudocódigo**.

TIPOS DE DATOS

- **Es un atributo que permite que un dato pueda ser creado. Define su capacidad de almacenamiento, el contenido que puede guardar y las operaciones que puede realizar.**
- Todos los elementos con los que requerimos realizar operaciones aritméticas , relacionales o lógicas tiene asociado un tipo de dato que determina las cosas que puede realizar.
- **SE CLASIFICAN EN DOS GRUPOS:**
- **SIMPLES O PRIMITIVOS Y COMPUESTOS O ESTRUCTURADOS.**
-



CLASIFICACIÓN DE LOS TIPOS DE DATOS



Simples o Primitivos

Son aquellos de tipo elemental en los lenguajes de programación esto es, aquellos que nos proporciona el lenguaje y con los que podemos construir estructuras más complejas.

Numéricos

Sólo permiten almacenar números

No Numéricos

Permite almacenar cualquier valor

Compuestos o estructurados

Combinan tipos de datos primitivos en estructuras complejas.

Estáticos

Su tamaño no puede modificarse

Dinámicos

Su tamaño puede ser cambiado

TIPOS DE DATOS NUMÉRICOS

Enteros: Son números que no tienen parte decimal, los expresaremos con la palabra **entero**.

FORMATO:

entero nombre_var ;

**VALORES QUE PUEDE
CONTENER:**

5677, 344 , 8, +45

Flotante : Son números con parte entera y parte decimal.

FORMATO:

flotante nombre_var ;

**VALORES QUE PUEDE
CONTENER :**

456.343 , -56.00 , 0.788

Se pueden expresar en punto fijo y punto flotante

Punto fijo: 3.456879

Punto Flotante: $4.866678 * 10^7$

sería 4.866678E7

Si desea representar valores muy grandes o muy pequeños puede expresarlos en punto flotante nosotros trabajaremos punto fijo.

TIPOS DE DATOS NO NUMÉRICOS

Tipo de datos carácter :

Es un elemento pequeño utilizado en el tratamiento de la información.
Solo pueden almacenar **un** carácter.

FORMATO:

caracter nombre_var ;

Pueden contener cualquier caracter:

- ☐ **Alfabéticos:** las letras minúsculas y mayúsculas (A,B....Z, a,b...z)
- ☐ **Numéricos:** los números del 0..9
- ☐ **Caracteres especiales:** (+, -, *, /, < , >, \$......)

TIPOS DE DATOS NO NUMÉRICOS

Tipo de dato Cadena :

Permite que se el dato pueda contener una combinación de caracteres tales como letras, números y caracteres especiales.

Sirve para guardar información de tipo texto.

FORMATO:

cadena nombre_var ;

TIPOS DE DATOS NO NUMÉRICOS

Tipo de datos Booleano:

Los datos lógicos o booleanos son aquellos que solo pueden tomar dos valores: **cierto** o **falso**.

Se utilizan para realizar preguntas y obtener respuesta de cierto o falso.

FORMATO:

booleano nombre_var ;

Este tipo de dato lo veremos trabajando en las sentencias condicionales.

EJEMPLO:

si(edad >= 18) esta pregunta puede generar dependiendo del valor de la variable **edad**, dos posibles respuestas **verdadero** si es cierto o **falso** si la respuesta es falsa. De esta manera trabajaremos a los tipos de datos booleanos.

IDENTIFICADORES

Le dan nombre a los elementos con los trabajamos en nuestro algoritmo: programas, funciones, clases, objetos, variables y constantes entre otros.

Son contruidos de acuerdo a las reglas de sintaxis definidas por cada lenguaje.

Cuenta
costo3
produccion
Calcular_itbm
seguro_social



IDENTIFICADORES

REGLAS DE CONSTRUCCIÓN DE UN NOMBRE DE IDENTIFICADOR.

1. El primer carácter debe ser una letra.
2. A partir del segundo carácter puede combinar letras , números y como único carácter especial , el subrayado (_).
3. Utilizar **minúscula para los nombres de variables** y **mayúscula cerrada** para las **constantes**.

EJEMPLO : **costo3** **producción** **seguro_social** → **estilo small_caps**

CONT → **estilo ALL_CAPS**

4. 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 irá en minúscula y luego la primera letra de cada palabra debe ser en mayúscula y sin espacios en blanco.

Prof. Mitzi M. **calcularDerivada** → **estilo camelCase.**



IDENTIFICADORES

REGLAS DE CONSTRUCCIÓN DE UN NOMBRE DE IDENTIFICADOR.

5. Los **nombres de funciones** deben ser iniciar con un verbo para indicar la acción de la función.

Ejemplo : mostrarSalarios → **Estilo CamelCase**

6. No pueden ser nombres de identificadores las palabras reservadas, por ejemplo:

imprimir **X** leer **X** entero **X**

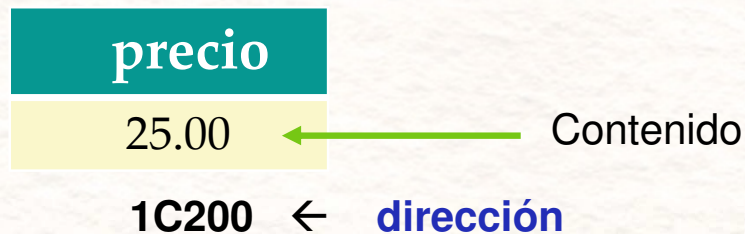
7. Use un nombre autodocumentativo.



Variable

Es un área de almacenamiento temporal con un nombre simbólico cuyo valor puede ser modificado a lo largo de la ejecución de un programa.

El contenido de una variable es el valor almacenado(contenido) en la posición particular de memoria (dirección) en ese momento.

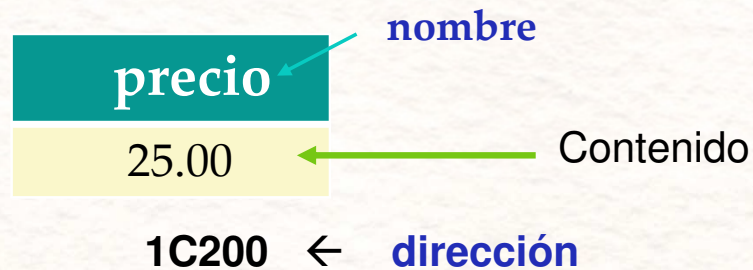


Variable

Características :

1. Tienen un tipo de dato,
2. Tienen un nombre,
3. Un área de almacenamiento
4. Una dirección de memoria donde se guarda su valor

CUÁL SERÍA EL TIPO DE DATO
Para la variable precio ?



Variable

Las variables pueden almacenar cualquiera de los tipos de datos explicados anteriormente. **Para poder usar una variable debemos declararla.**

Declaración de Variables:

FORMATO:

tipo de dato lista de variables;

EJEMPLOS

entero i;

caracter sexo, estado;

flotante tarifa, monto;

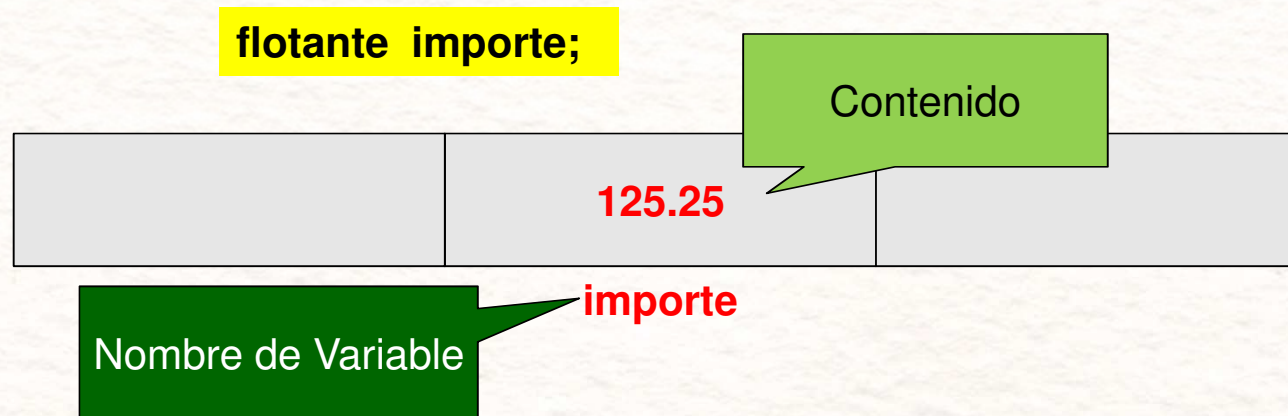
cadena direc;

De acuerdo a las reglas de estilo de escritura las variables se escriben usando small-caps

Variable

Hay que diferenciar entre nombre de la variable y su contenido. El **nombre** es una **identificación** que se le da a un conjunto de posiciones contiguas de memoria, mientras que el **contenido** de una variable es el valor que está almacenado en dichas posiciones.

Ejemplo:



CONSTANTES

Es un valor que no puede ser alterado durante la ejecución del programa.

Existen tantos tipos de constantes como tipos de datos.

Cuando las constantes son muy frecuentes pueden recibir un nombre de identificador con el que se les reconocerá a lo largo de todo el programa.

- Las constantes numéricas pueden ser enteras o reales.
100 , 26, +56.3 , 1000.78
- Las constantes de carácter deben escribirse entre apóstrofes
'q' '# '5'
- Las constantes de cadena deben ir entre **comillas**.
"Ave. Manuel Espinosa B.", "TEATRO BALBOA" "12"

