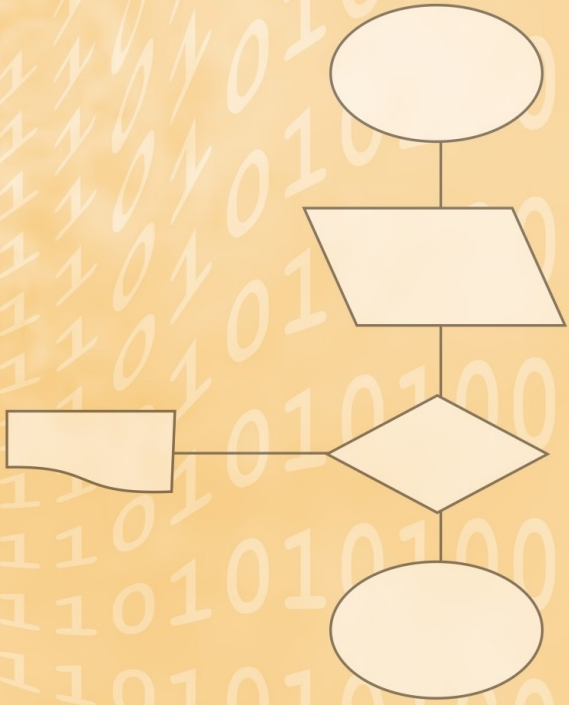
Textos Académicos

Roberto Carlos Guevara Calume

Sentencias básicas usadas en la programación de computadores



ITM

Sentencias básicas usadas en la programación de computadores

Roberto Carlos Guevara Calume

INSTITUTO TECNOLÓGICO METROPOLITANO
Institución Universitaria

SENTENCIAS BÁSICAS USADAS EN LA PROGRAMACIÓN DE COMPUTADORES

1a. Edición: diciembre de 2008

ISBN: 978-958-8351-57-5

Dirección editorial
Fondo Editorial ITM

Corrección de textos
Lucía Inés Valencia

Diagramación y montaje
L. Vieco e Hijas Ltda.

Impreso y hecho en Medellín, Colombia

Instituto Tecnológico Metropolitano
Calle 73 No. 76A 354
Tel.: (574) 440 51 00
Fax: 440 51 01
www.itm.edu.co
Medellín - Colombia

Contenido

Parte I

Conceptualización

Objetivos

Hacer una introducción general a los diferentes tipos de lenguajes de programación, y los elementos comunes a éstos, como son las constantes, las variables, la asignación y demás estructuras relacionadas con la programación.

1. Los lenguajes de programación

Lenguaje: es en general un sistema de comunicación en el cual se utilizan símbolos. Un **lenguaje de programación** es un sistema de comunicación entre el programador y la máquina (computador), creado con el único fin de programar computadoras, esto se hace partiendo del uso de instrucciones preestablecidas. Estas instrucciones las podemos asociar al idioma usado por los humanos para comunicarse.

Un computador funciona siguiendo un conjunto de instrucciones llamadas programas, estos son los pasos que uno a uno sigue un computador para completar una tarea.

En general los programas son escritos por personas que conocen algún lenguaje de programación, como son el Visual Basic, Delphi, C++, Java, entre otros.

Para entender el concepto de lenguaje de programación y programa, intentemos resolver un problema real, el cual implique realizar un programa de computador.

Problema: Se tienen dos números, los cuales serán ingresados por teclado, crear un programa que tome estos números e imprima la suma de ellos.

```
Program ABC;
Var
Numero1, Numero2, Resultado: integer;
Begin
Write('DIGITE DOS NUMEROS');
Read (Numero1);
Read (Numero2);
Resultado := Número1 + Numero2;
Write('La suma es igual a ', resultado)
End.
```

El anterior es un programa codificado en un lenguaje de programación llamado pascal, y cada línea es una instrucción.

Miremos el mismo programa codificado en C++

```
#include <stdio.h>
#include <conioh>
void main(void)
{
int Numero1, Numero2, resultado;

printf("DIGITE DOS NUMEROS");
scanf("%d",&Numero1);
scanf("%d",&Numero2);
resultado = Numero1 + Numero2;
printf("La suma es igual a %d" , resultado);
}
```

Como podemos ver, la estructura general es parecida, pero las instrucciones cambian, la lógica de programación se conserva y es aplicable a varios lenguajes de programación.

1.1 Tipos de instrucciones

En forma general, los lenguajes de programación manejan cinco grupos de instrucciones; las dividimos así para facilitar la explicación, además de ayudarnos a estructurar las instrucciones para una mejor comprensión, por lo que usted podrá usarlas sin detenerse a pensar de qué tipo serán:

Instrucciones de entrada/salida: Permiten mostrar e introducir información desde y hacia el PC, o sea sirven para controlar el flujo de información entre la CPU y los periféricos como el teclado o el monitor.

Instrucciones aritméticas lógicas: Son las instrucciones que ejecutan operaciones, tales como suma, resta, multiplicación, división, y otras cuya respuesta no es un número, sino una respuesta de falso o verdadero, como or, xor, not, and*.

Instrucciones selectivas: Son instrucciones que permiten la bifurcación; en otras palabras, la bifurcación permite que el programa no siempre tome el *mismo "camino";* como ejemplo, si la variable edad tiene un valor de 18 o más, un programa podrá imprimir *"mayor de edad",* pero si tiene menos de 18 el programa podrá imprimir *"menor de edad"*; es decir, el programa dará resultados diferentes según el contenido de la variable edad.

Instrucciones repetitivas: Permiten la repetición de partes del programa; es decir, si tengo 50 alumnos y a todos debo hacerles el mismo procedimiento, es deseable usar algo que permita repetir las mismas líneas de código para cada alumno, y no escribir 50 veces una tras otra las mismas instrucciones.

Instrucciones de almacenamiento/consulta: Son instrucciones que permiten llevar valores a la memoria del computador o a otros dispositivos, además de recuperar el valor que contienen estos dispositivos.

1.2 Clasificación de los lenguajes de programación

Aunque todos los lenguajes de programación tienen básicamente los mismos tipos de instrucciones, existen diferencias notables entre los símbolos usados, además del nivel de abstracción y la forma de expresar estas instrucciones.

---

* Se recomienda repasar los conceptos básicos de la lógica booleana.

1.2.1 Lenguaje de máquina

El lenguaje de máquina es un lenguaje complejo, que está al nivel de lo que puede entender directamente un computador, se dice que es de bajo nivel por que se requiere que el programador conozca en detalle el procesador al cual se está programando. Sin embargo como las instrucciones son escritas en el lenguaje nativo de la máquina, se ejecutan más rápido, la razón es que estas instrucciones son entendidas inmediatamente por el computador en forma nativa sin necesidad de ningún proceso extra. Un ejemplo de una línea de código para este tipo de lenguaje es:

00010011111101111111000001111100

Tabla 1. Pros y contras lenguajes de máquinas

Pros	Contras
Rápido	Difícil programación
Natural del computador	Difícil seguimiento de un error
Puede hacer uso de todo el poder de la CPU	El código es único para cada máquina

1.2.2 Lenguajes ensambladores

La programación en lenguaje de máquina "unos y ceros" como vimos puede ser pesada para el programador, por lo que se pensó en un lenguaje que ayude a aliviar el proceso de programación.

El lenguaje ensamblador usa códigos mnemotécnicos* para las instrucciones y direcciones de memoria. Los lenguajes ensambladores se diseñan para un modelo específico de CPU al igual que los lenguajes de bajo nivel, por lo que los programas necesitan ser reescritos si se

* La palabra mnemotecnia es una técnica o procedimiento de asociación mental para recordar fácilmente las cosas.

intentan ponerlos a funcionar con otro modelo de CPU. Un ejemplo de una línea de código en este lenguaje es

MOV **AX**,(F8)

Donde **MOV** es "Guardar en o mover" en código mnemotécnico; **AX** es una parte interna de la CPU llamada registro, donde se puede conservar información; y F8 entre paréntesis **(F8)**, es llevar el número F8, en formato hexadecimal (igual a 248 en decimal), al registro AX. Es de entender que esta misma instrucción puede ser diferente para cada CPU.

Tabla 2. Pros y contras lenguajes ensambladores

Pros	Contras
Se usan instrucciones mnemotécnicas	Sigue siendo una programación tediosa
Se facilita más la programación	El programador aun debe saber en qué parte de la memoria del PC están sus datos

1.2.3 Lenguajes de alto nivel

Son lenguajes donde una sola instrucción puede realizar varias instrucciones en lenguaje ensamblador. Estos lenguajes pueden ser usados en diferentes tipos de máquina y de diferentes fabricantes. Se basan en reglas de sintaxis*.

Están orientados a una clase específica de problemas, por esto existen lenguajes con orientaciones particulares, como: procesamiento de problemas de cálculos científicos, manejo de archivos, inteligencia artificial, etc.

* Son reglas para poder relacionar y enlazar las instrucciones entre sí. Es la forma correcta en la que se puede escribir instrucciones.

Un ejemplo de una instrucción en un lenguaje de alto nivel es

A:= B*5

Donde a una variable llamada **A** se le lleva el contenido de la variable **B** multiplicada por **5**, note que el contenido de **B** no cambia, pero **A** queda con cinco veces el valor de **B**

Tabla 3. Pros y contras lenguajes de alto nivel

Pros	Contras
Más fáciles de entender y de corregir que los otros tipos de lenguajes	Son más lentos, pues éstos deben ser decodificados
Independientes a la máquina	No se hace uso de todo el potencial de la máquina

Algunos ejemplos de lenguajes de alto nivel son: PASCAL, Basic, Visual Basic, C, C++, Java, Cobol, FoxPro, Delphi, ente otros.

1.3 Interpretadores y compiladores

El usar lenguajes de alto nivel implica alejarse del lenguaje que el computador entiende, o sea el lenguaje binario* o de máquina, por lo que cualquier programa escrito para un lenguaje de alto nivel debe ser decodificado de alguna forma para que el computador pueda entenderlo.

Para hacer la labor de decodificación se cuenta con dos tipos de programas, que son los interpretadores y los compiladores,

1.3.1 Interpretadores

Son programas de terceros que toman nuestro código, lo leen instrucción por instrucción y lo traducen, línea a línea, a lenguaje de máquina, al mismo tiempo que se va ejecutando.

* Lenguaje en el cual sólo existen los símbolos 0 y 1.

Ilustración i. Interpretadores

1.3.2 Compiladores

Son programas de terceros que toman nuestro código fuente, y lo traduce completamente, convirtiéndolo en lenguaje de máquina.

El compilador deja un programa en lenguaje de máquina completo y ejecutable, es decir, un archivo que luego puede ser ejecutado. Lo anterior permite que un programa compilado sea más rápido que un programa interpretado, al momento de ejecutarlo.

Ilustración ii. Compiladores

1.4 Elementos de los lenguajes de programacion

En un lenguaje de programación podemos encontrar diferentes elementos. Nuevamente aquí estructuramos los lenguajes en elementos, porque nos facilitarán la explicación, además de ayudarnos a tener una mejor comprensión:

1.4.1 Las constantes

Son elementos que, como su nombre lo indica, se mantienen constantes y no cambian durante la ejecución del programa. Usaremos el ejemplo de una constante muy conocida Π (pi), el valor de Π es 3.141592. Si el programa realiza operaciones con circunferencias o círculos, vemos que Π no cambia, este concepto lo podemos aplicar a todos los valores que no cambien durante la ejecución, incluso si estos valores no provienen de constantes matemáticas.

1.4.2 Las variables

A diferencia de las constantes estos elementos pueden tomar diferentes valores durante la ejecución de un programa. Para "bautizar" o escoger los nombres de las variables debemos usar palabras mnemotécnicas, que indiquen la información que contiene, por ejemplo, la variable sueldo indica un valor numérico que puede cambiar según los cálculos de liquidación de cada empleado.

1.4.3 Los datos simples

Son datos que contienen un solo valor que puede ser un número, letras o datos lógicos.

Tabla 4. Tipos de datos simples

Tipo dato simple	Contiene	Ejemplo
Numéricos	Enteros, reales	10, 3.2154, 2.35E18
Lógicos	Booleanos	Verdadero, falso
Carácter	Son todos los caracteres individuales	A,B,9,0,+,%, etc.
Cadena (String)	Se forman agrupando varios caracteres	"hola", "Juan Pérez", "2h34", etc.

Ejemplos:

Sueldo: Real
Nombre: String
Vive: Boolean

1.4.4 Los datos compuestos

Estos tipos de datos son creados por el programador, usando tipos de datos simples, creando así nuevos tipos de datos.

Por ejemplo, se puede crear un nuevo tipo de datos compuesto, que se llamará empleado y podría definirse así:

```
Empleado:
        Nombre : String
        Telefono: String
        Sueldo: Real
        Pensionado: Boolean
Fin
```

1.4.5 Las expresiones

Las expresiones son combinaciones de constantes, variables, símbolos, y operadores. Algunos ejemplos:

$$y = 2x^3 + 3x^2 + 4x + 5 \qquad Total = Suma1 + Suma2$$

$$y = \frac{b^2 + \sqrt{4ac}}{2a} \qquad Circunferencia = \prod * r^2$$

$$\text{Sumatoria\_calculada} = \sum_{k=0}^{n} \left(x_k a^{n-k}\right)$$

Podemos agrupar las expresiones en expresiones matemáticas y expresiones lógicas.

Expresiones matemáticas

Son como las fórmulas matemáticas, y sus valores de respuesta son valores numéricos. Éstos, a su vez, tienen operadores.

Tabla 5. Expresiones matemáticas

Operador	Significado	Ejemplo
^	Se usa para elevar un número	5 ^ 3 = 125
+	Suma	5 + 3 = 8
-	Resta	5 - 3 = 2
*	Multiplicación	5 * 3 = 5
/	División	5/3 = 1.666666666
Div	Cociente de la división	5 Div 3 = 1
Mod	Residuo de la división	5 Mod 3 = 2

Expresiones lógicas

También llamadas booleanas*. Son expresiones cuyo valor sólo es verdadero o falso; éstas, a su vez, están ligadas a otras expresiones llamadas relacionales o de comparación.

Tabla 6. Expresiones lógicas

Operador	Significado	Ejemplo	Tipo
>	Mayor que	5 > 3 Verdadero	Comparación
<	Menor que	5 < 3 Falso	Comparación
=	Igual	5 = 3 Falso	Comparación
≥	Mayor o igual que	5 ≥ 3 Verdadero	Comparación
≤	Menor o igual que	5 ≤ 3 Falso	Comparación
<>	Diferente a	5<>3 Verdadero	Comparación
No	Negación	No (5<>3) Falso	Lógica
∧	"Y" verdadero si de dos expresiones ambas son verdaderas	[(5>3) ∧ (5=5)] Verdad [(5>3) ∧ (5<>5)] Falso	Lógica
∨	"O" verdadero si de dos expresiones una de ellas o ambas son verdaderas	[(5>3) ∨ (5=5)] Verdad [(5>3) ∨ (5>6)] Verdad [(3<>3)∧(5<>5)] Falso	Lógica

1.4.6 Las funciones internas

Las funciones internas son funciones incorporadas en los lenguajes de programación, que nos permiten realizar cálculos usando funciones predefinidas.

* Llamadas así en honor del matemático británico George Boole.

Supongamos que a la variable RESPUESTA quisiéramos llevar la variable **A,** pero elevada al cuadrado y a la respuesta de lo anterior, sacarle raíz cuadrada.

$$Respuesta = \sqrt{A^2}$$

Si mi lenguaje de programación no tiene incorporada una función para elevar al cuadrado ni una función para sacar raíz cuadrada, tendría que valerme de *"complejos cálculos"* donde sólo podría usar las sumas, restas, divisiones y multiplicaciones, pero gracias a estas funciones podría simplificar el cómputo, algo parecido a lo utilizado en las hojas de cálculo, de la siguiente manera.

```
RESPUESTA = sqrt(sqr(A));
```

No siempre todos los lenguajes incorporan las mismas funciones. En los manuales de cada lenguaje se especifican cuáles funciones vienen incorporadas. Las más comunes a todos los lenguajes son:

Tabla 7. Funciones internas

Función	**Significado**	**Ejemplo**
Abs(x)	Valor absoluto de X	Abs(-5) = 5
Sen(x)	Seno de x	Sen(90) = 1
Cos(x)	Coseno de x	Cos(90) = 0
Tan(x)	Tangente de x	Tan(180)= 0
Ln(x)	Logaritmo natural de x	Ln (1) = 0
Log(x)	Logaritmo en base 10	Log(10) = 1
Truncar(x)	Elimina la parte decimal de x	Truncar (7.823) = 7
Redondeo(x)	Lleva x al entero más cercano	Redondeo(7.82) = 8
Sqr(x)	Eleva x al cuadrado	Sqr (2) = 4
Sqrt(x)	Saca raíz cuadrada a x	Sqrt(4) = 2

1.4.7 Las asignaciones

Es la operación en la cual un valor es *"pasado"* o asignado a una variable, y se denota así:

Nombre de Variable ← Expresión o Variable o Valor

Ejemplos

Tabla 8. Asignaciones

Instrucción	Efecto
A ← 5	La variable A recibe el valor de 5
A ← B	A toma el valor que tenga B en ese momento
A ← sen(B)+5	A recibe la suma de seno(B) y luego le suma 5
Total ←suma1+suma2	La variable total toma el valor de la suma, las variables suma1 y suma2
Nombre←"Carmen"	La variable nombre toma el valor de Carmen
$y \leftarrow \frac{b^2 + \sqrt{4a}}{2a}$	La variable "y" recibe el valor de la expresión matemática
y ← (b^2+(4*a*c)^(1/2))/2*a	Es similar a la anterior

1.4.8 Entrada datos

La entrada de datos se realiza generalmente por teclado, mediante instrucciones que permiten capturar datos.

Para leer datos se usan los siguientes formatos

LEA(X);	LEA(A,B,C);

Ejemplos:

Tabla 9. entrada de datos

Instrucción	Efecto
Lea(A)	La variable A recibe el valor dado por teclado
Lea(A, B)	Lee dos valores por teclado y el primero lo asigna a la variable A y el segundo a la variable B
Lea(Sueldo)	La variable Sueldo recibe el valor introducido por teclado
Lea(Nombre, Sueldo)	Las variables Nombre y Sueldo toman valores dados por teclado

1.4.9 Salida de datos

Son realizadas generalmente por pantalla o impresora, y permiten visualizar datos. Se usan los siguientes formatos*:

IMP(X);	IMP("el Sueldo es ",Sueldo);

Ejemplos

Tabla 10. Salida de datos

Instrucción	Efecto
Imp(A)	El contenido de la variable A es impresa
Imp (A, B)	Las variables A y B son impresas
Imp("Sueldo")	La palabra Sueldo es impresa en pantalla
Imp("el sueldo es ", S)	La frase "el sueldo es "se imprime y a continuación se imprime el contenido de la variables S

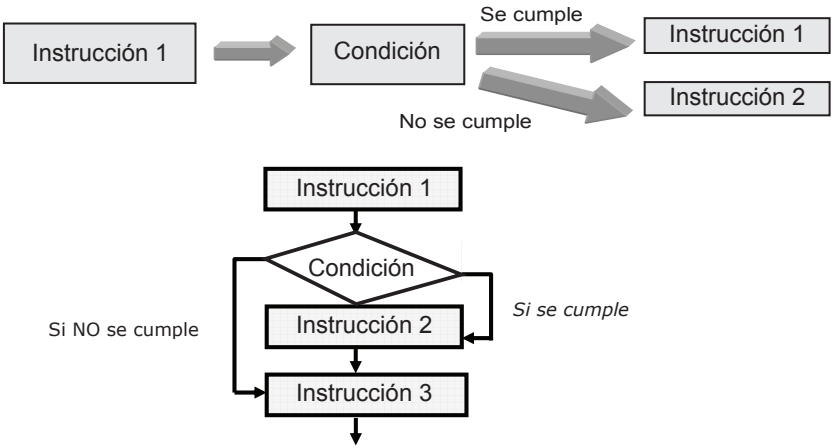
* Se usará la palabra IMP para indicar una salida por pantalla, aunque también puede ser usada la palabra IMPRIMA.

1.4.10 Las bifurcaciones

Estas instrucciones son las que permiten "saltos" dentro de un mismo programa a otras líneas de código no consecutivas, según se cumpla o no una condición.

Por ejemplo, si se cumple cierta condición, entonces el programa continúa en la línea X; si no se cumple la condición, éste "salta" o continúa en la línea Y.

Ilustración iii. Bifurcaciones



1.5 Ejercicios propuestos y resueltos

1. Efectúe los siguientes ejercicios

1 Lea un número por teclado
2 Lea un nombre por teclado
3 Lea Nombre, Cedula y Dirección
4 A la variable total lleve (1000 + sueldo)
5 Imprima su nombre
6 Lea una variable e imprímala
7 Lea tres variables e imprima su promedio

8 Lea tres variables y súmelas, e imprima las tres variables y su suma
9 Lea una variable aumente su valor en 10 y el resultado aumén-telo 10 veces
10 Lea un número y redondéelo al entero más cercano
11 Saque la raíz cuadrada de 30 y almacene el resultado en RC30

Respuestas

1 Lea(N)
2 Lea(nom)
3 Lea(Nombre, Cedula, Direccion)
4. Total ← (1000 + sueldo)
5. Imp("mi nombre")
6 lea (num); Imp(num)
7 Lea(A, B, C) ; Prom ← (A+B+C) / 3
8. Lea(A, B, C) ; Sum←A+B+C ; Imp(A, B, C, Sum)
9. Lea(A); A←(A+10)*10
10. lea(num) ; num←redondeo(num)
11. RC30 ← sqrt(30) ó RC30 ← 30^(0.5) **ó** RC30 ← 30^(1/2)

2. Responda falso o verdadero

1. La ventaja de los lenguajes de máquina es que pueden ser ejecutados en cualquier máquina ___.
2. En un lenguaje de alto nivel, las instrucciones son escritas directamente en lenguaje que entiende el computador ___.
3. El compilador traduce una a una las instrucciones e inmediatamente las ejecuta ___.

Respuestas

1. F
2. F
3 F

4. Responda a las siguientes preguntas:

- ¿Cuál es la diferencia entre interpretador y compilador?
- Describa los pros y los contras de los diferentes lenguajes de programación.
- ¿Cuáles son los elementos que intervienen en un lenguaje de programación?
- Mencione algunas funciones internas de los lenguajes de programación y explique cómo usarlas
- ¿Qué es un programa?

5. **Defina los siguientes conceptos**

Lenguajes de máquina
Ensambladores
Lenguajes de alto nivel
Compilador e interpretador

Sentencias básicas usadas en la programación de computadores

se terminó de imprimir en diciembre de 2008.
Para su elaboración se utilizó papel Bond de 75 g,
en páginas interiores, y cartulina Propalcote 240 g para la carátula.
Las fuentes tipográficas empleadas son Times New Roman 11 puntos,
en texto corrido, y Myriad Pro 14 puntos en títulos.