

Programación Digital

Unidad 3. Elementos básicos de un programa



1. Concepto de Programa

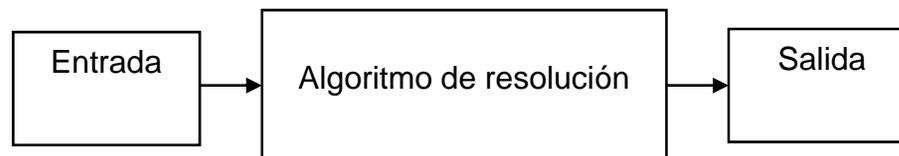
Es un conjunto de instrucciones (órdenes dadas a la computadora), que producirán la ejecución de una determinada tarea.

En esencia, un programa es un medio para conseguir un fin, probablemente dicho fin sea solucionar un problema.

2. Parte de un programa

Las parte de un programa son:

- una cabecera: es una acción simple que comienza con la palabra **program** seguida del nombre asignado al programa completo.
- un bloque algoritmo: es el resto del programa, y a su vez consta de dos componentes o secciones:
 - las acciones de declaración: definen o declaran las variables, constantes y subprogramas.
 - las acciones ejecutables: son las acciones que debe ejecutar la computadora, cuando se codifica el algoritmo.





3. Instrucciones y tipos de instrucciones

Las instrucciones son las acciones que ejecuta un programa, definidas en el proceso de diseño de un algoritmo y su codificación. Estas deben crearse en el mismo orden que serán ejecutadas, es decir, en secuencia.



3. Instrucciones y tipos de instrucciones

Un programa puede ser:

- **Lineal:** si las instrucciones se ejecutan secuencialmente, sin bifurcaciones, decisiones o comparaciones.
- **No lineal:** cuando se interrumpe la secuencia mediante instrucciones de bifurcación.



3. Instrucciones y tipos de instrucciones

Tipos de instrucciones:

- instrucciones de inicio/fin (begin/inicio)/(end/fin).
- instrucciones de lectura (readln/leer).
- instrucciones de escritura (writeln/escribir).
- instrucciones de asignación ($A \leftarrow 5$, $B \leftarrow 7$).
- instrucciones aritméticas y lógicas.
- Instrucciones de decisión o selección.
- Instrucciones de repetición.



3. Instrucciones y tipos de instrucciones

- instrucciones de inicio/fin: indican el inicio y fin del algoritmo de resolución del programa, es decir, donde se inicia y finaliza el conjunto de instrucciones del programa. Palabras reservadas para un algoritmo: inicio y fin.
- Instrucciones de lectura de datos (entrada): captura los datos usando un dispositivo de entrada. Palabra reservada para un algoritmo: *leer* .
 - Ej. Leer(nombre), Leer(horas, precio).



3. Instrucciones y tipos de instrucciones

- Instrucciones de escritura de resultados (salida): escribir los resultados en un dispositivo de salida. Palabra reservada para un algoritmo: *escribir*. Ejemplos:

- escribir(A)
- escribir(B, C)
- escribir('mensaje')
- escribir('mensaje',A)

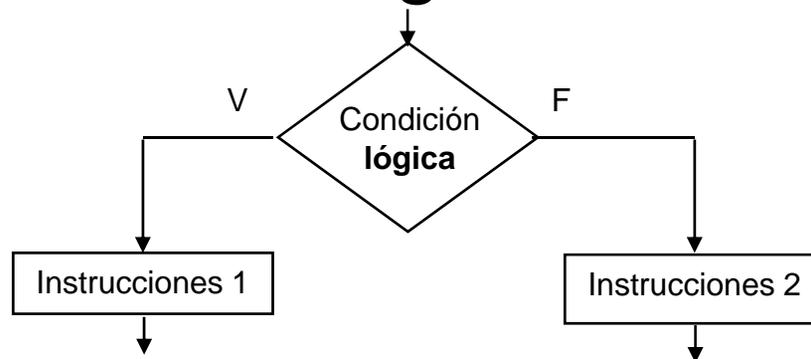


3. Instrucciones y tipos de instrucciones

- Instrucciones de asignación: se emplean para asignar un valor o resultado de una operación a una variable.
- Símbolo empleado: \leftarrow . Ejemplos:
 - $A \leftarrow 5$
 - $B \leftarrow 0$ (inicializar variables)
 - $N \leftarrow N+1$

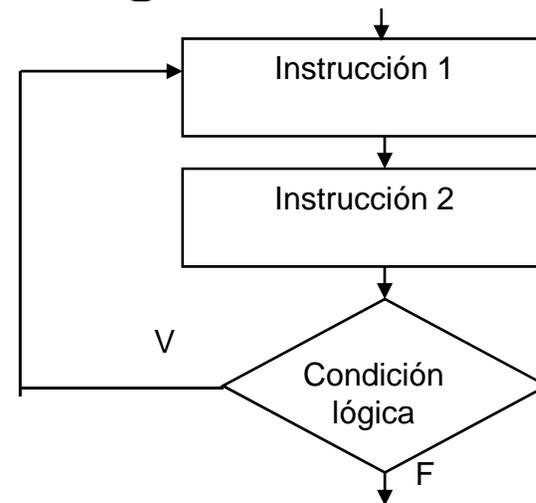
3. Instrucciones y tipos de instrucciones

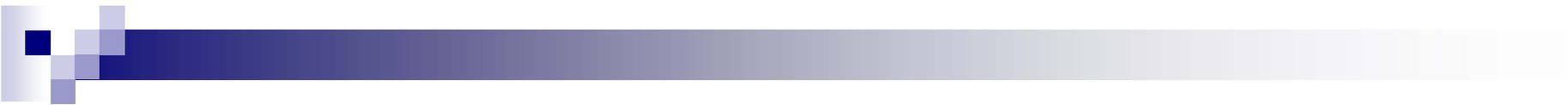
- Instrucciones de decisión o selección: se emplean para decidir qué instrucciones del programa ejecutar, según el cumplimiento de una condición lógica. Palabras reservadas para un algoritmo: *si*, *entonces*, *sino*, *según_sea*.



3. Instrucciones y tipos de instrucciones

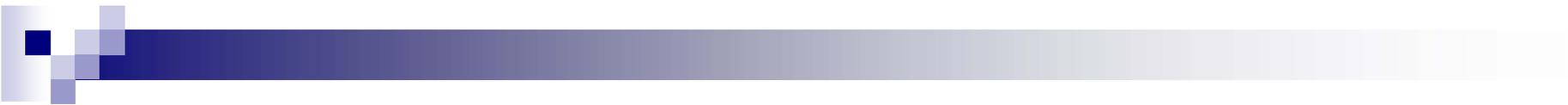
- Instrucciones de repetición: permite, como su nombre lo indica, repetir un conjunto de instrucciones una cantidad dada de veces. Palabra reservada para un algoritmo: *mientras, repetir, desde*.





4. Elementos básicos de un programa

- Los lenguajes de programación tienen elementos básicos que se utilizan como bloques constructivos, así como reglas utilizar dichos elementos. Estas reglas se denominan *sintaxis* del lenguaje.
- Solamente las instrucciones sintácticamente correctas, pueden ser interpretadas por la computadora. Los programas que contengan errores de sintaxis, son rechazados por la máquina.



4. Elementos básicos de un programa

Los elementos básicos constitutivos de un programa o algoritmo son:

- Palabras reservadas (inicio, fin, si-entonces, ...).
- Identificadores (nombre del programa, de las variables, procedimientos, funciones, etc.)
- Caracteres especiales (, ' * / : ; = <> > <).
- Constantes.
- Variables.
- Expresiones.
- Instrucciones.



Otros elementos de un programa

Además de estos elementos básicos, existen otros elementos que forman parte de los programas, cuya comprensión y funcionamiento será vital para el correcto diseño de un algoritmo y su codificación. Estos elementos son:

- bucles: en programación es una sentencia que se realiza repetidas veces. Un bucle se utiliza para hacer una acción repetida sin tener que repetir varias veces el mismo código, lo que ahorra tiempo, deja el código más claro y facilita su modificación en el futuro.



Otros elementos de un programa

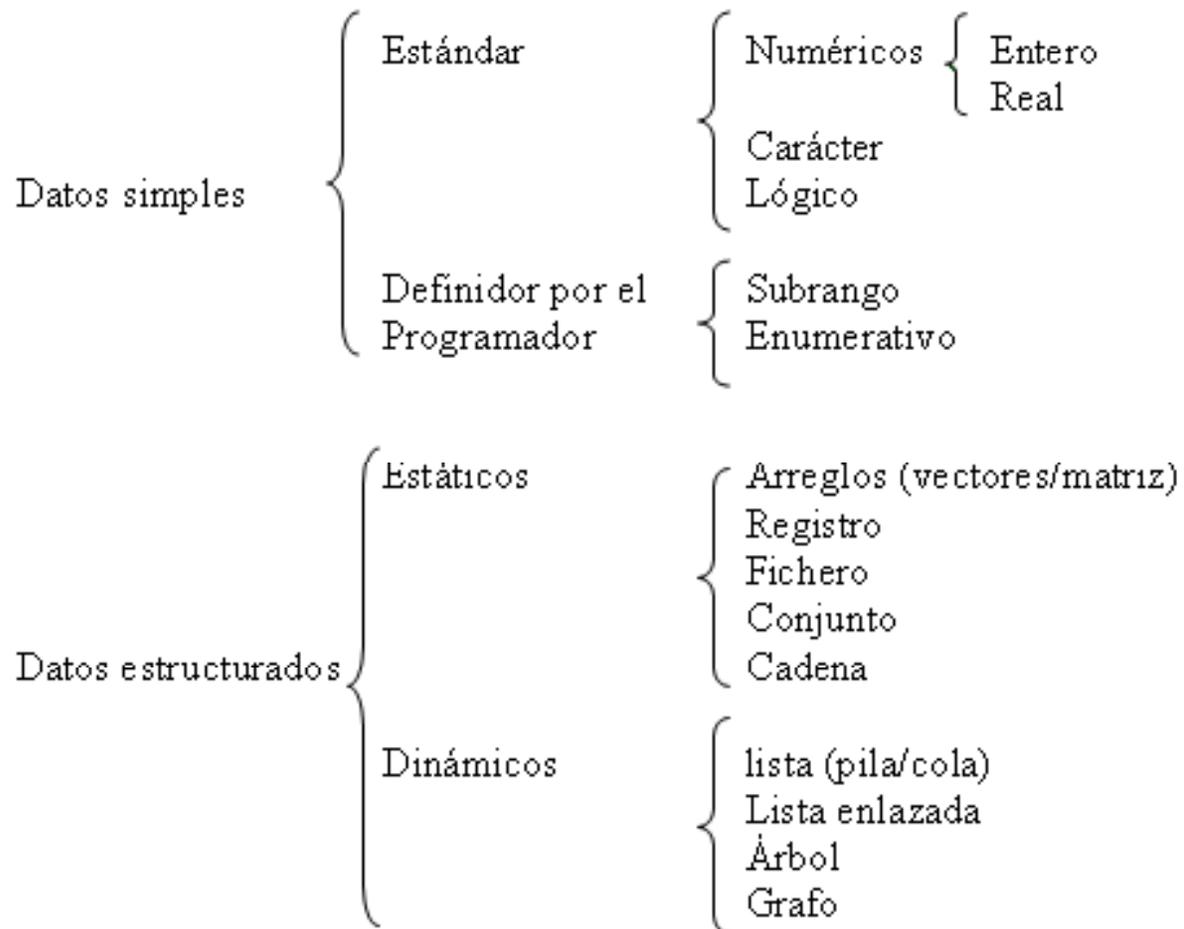
- **contadores:** es una variable que cuenta la cantidad de veces que se ejecuta. Se suele denominar contador cuando el incremento de la variable es de 1 en 1, pero no tiene por que ser así obligatoriamente. La sintaxis genérica sería: $\text{variable} = \text{variable} + 1$.
- **acumuladores:** La diferencia entre acumuladores y contadores es que los acumuladores se incrementan con cualquier cantidad, como por ejemplo el total de una factura, mientras que los contadores se incrementan siempre en la misma cantidad, normalmente 1.
- **estructuras:** secuenciales, selección y repetición.



5. Datos, tipos de datos y operaciones primitivas

- Un dato es la expresión general que describe los objetos con los cuales opera una computadora. La mayoría de las computadoras pueden trabajar con varios tipos de datos.
- En el proceso de resolución de problemas el diseño de la estructura de datos es tan importante como el diseño del algoritmo y del programa.
- A nivel de máquina, un dato es una secuencia de bits (ceros y unos).

Tipos de datos





5.1 Tipos de datos

Existen dos tipos de datos: simples (sin estructura) y compuestos (estructurados).

- **Datos Simples:** son datos que se representan con un único valor.
- **Datos Estructurados:** son datos que pueden almacenar uno, dos o n valores, bajo el mismo identificador.



5.1 Tipos de datos

- Tipos de datos simples:
 - Numéricos: entero (integer) y real (real).
 - Lógicos (boolean).
 - Carácter (char).
 - Cadena de caracteres (string).



Datos Numéricos

Es el conjunto de los valores numéricos y pueden representarse en dos formas distintas:

- Tipo numérico entero (integer): es un subconjunto finito de los números enteros. Los enteros son números completos, no tiene componentes fraccionarios o decimales y pueden ser negativos o positivos. Los números enteros máximos y mínimos de una computadora suelen ser -32768 a +32767. Los números enteros fuera de este rango no se suelen representar como entero, sino como reales, aunque en algunos lenguajes como Fortran o Turbo Pascal existe el tipo de dato entero largo cuyo rango es -2.147.483.648 a 2.147.483.647.



Datos Numéricos

- Tipo numérico real (real): el tipo real consiste en un subconjunto de los números reales y siempre tiene un punto decimal y pueden ser positivos o negativos.

Rango 2.9×10^{-39} a 1.7×10^{38} .

En aplicaciones científicas se requiere de la notación exponencial o científica para números muy grandes o muy pequeños. Ejemplos:

367.520.100.000.000.000.000 se utiliza

3.675201×10^{20}

0.00000000000302579 se utiliza 3.02579×10^{-11}



Datos lógicos: (booleano)

Es aquel dato que sólo puede tomar uno de dos posibles valores: verdadero (true) o falso (false).

Se utiliza para representar las alternativas si/no a determinadas condiciones. Por ejemplo, cuando se pide si un valor entero es par, la respuesta será verdadera o falsa según sea el caso.



Datos tipo carácter y tipo cadena

El tipo carácter es el conjunto finito y ordenado de caracteres que la computadora reconoce. Un dato tipo carácter contiene un solo carácter. Los caracteres que reconoce una computadora no es estándar, sin embargo la mayoría reconoce los siguientes caracteres alfabéticos y numéricos:

- Caracteres alfabéticos: (A,B,C,...Z) (a,b,c,...,z)
- Caracteres numéricos: (0,1,2,3,.....,9)
- Caracteres especiales: (+ , - , * , / , ^ , . , ; , < , > , \$, ...)



Datos tipo carácter y tipo cadena

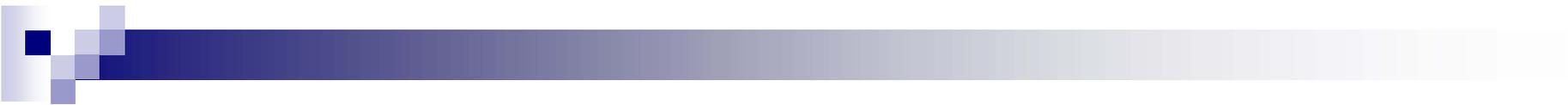
Una cadena de caracteres (string) es una sucesión de caracteres que se encuentran delimitados por una comilla.

La longitud de una cadena de caracteres es el número de ellos comprendidos entre los limitadores. Ej. 'Esta es una clase de programación digital'.



6. Constantes y Variables

- Constantes: son los valores que no cambian durante la ejecución del programa.
- Variables: son los valores que si cambian en la ejecución del programa.
- Tipos de constantes y variables: numéricas (enteros y reales), caracteres, lógicas y de cadenas.



6. Constantes y Variables

- Una variable que es de cierto tipo puede tomar únicamente valores de ese tipo. Si se intenta asignar un valor que no corresponda al tipo de dato de la variable, se producirá un error de tipo.
- Una variable se identifica por los siguientes atributos: *nombre* que se le asigna y *tipo* que describe el uso de la variable.



7. Expresiones

Las expresiones con combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Constan de operandos y operadores.

Según sea el tipo de objetos que manipula, las expresiones se clasifican en:

- aritméticas: resultado es de tipo numérico.
- lógicas: resultado de tipo lógico.
- de relación: resultado de tipo lógico.



7.1 Expresiones aritméticas

Son análogas a las fórmulas matemáticas.

Las variables y constantes son numéricas (tipo real o entero) y las operaciones son las aritméticas:

+ suma, - resta, * multiplicación, / división,

^ elevar a una potencia, DIV cociente de la división entera, MOD módulo o resto de la división entera.



7.1 Expresiones aritméticas

- Operadores aritméticos: + - * / ^ y las palabras reservadas DIV y MOD.
- Los operadores se utilizan de igual forma que en matemáticas: $A*B$, $C/8$.
- Al igual que en matemáticas el signo menos juega un doble papel, como resta en $A-B$ y como negación $-A$.



7.1 Expresiones aritméticas

Operadores DIV y MOD

- Solo se pueden usar cuando los operandos son expresiones enteras.
 - MOD: determina el resto de una división entera.
 - DIV: determina el cociente de una división entera.
- Ejemplo:
- 15 MOD 6 toma el valor 3.
- 15 DIV 6 toma el valor 2.

15 | 6
resto → 3 2 ← cociente



Reglas de prioridad

- Las expresiones que tiene dos o más operandos, requieren unas reglas matemáticas que permitan determinar el **orden** de las operaciones, dichas reglas se denominan reglas de prioridad o precedencia y son:
 1. las operaciones que están encerradas entre paréntesis se evalúan primero. Si existen diferentes paréntesis anidados, las expresiones más internas se evalúan primero.



Reglas de prioridad

2. Las operaciones aritméticas dentro de una expresión, suelen seguir el siguiente orden de prioridad:
 - operador de potencia (^)
 - operadores * , /
 - operadores DIV y MOD
 - operadores + , -



Reglas de prioridad

3. En caso de coincidir varios operadores de igual prioridad en una expresión o subexpresión encerrada entre paréntesis, el orden de prioridad en este caso es de izquierda a derecha.

Reglas de prioridad

Ejemplos: Obtener el resultado de las siguientes expresiones:

$$\begin{array}{l} \text{a) } 4 + 9 * 12 \\ \quad \underbrace{} \\ 4 + 108 \\ \quad \underbrace{} \\ 112 \end{array}$$

$$\begin{array}{l} \text{b) } 2 + 5 * 3 + 10 * 7 \\ \quad \underbrace{} \\ 2 + 15 + 10 * 7 \\ \quad \underbrace{} \\ 2 + 15 + 70 \\ \quad \underbrace{} \\ 17 + 70 \\ \quad \underbrace{} \\ 87 \end{array}$$

$$\begin{array}{l} \text{c) } 4 * 7 + 2^3 / 4 - 5 \\ \quad \underbrace{} \\ -4 * 7 + 8 / 4 - 5 \\ \quad \underbrace{} \\ -28 + 8 / 4 - 5 \\ \quad \underbrace{} \\ -28 + 2 - 5 \\ \quad \underbrace{} \\ -26 - 5 \\ \quad \underbrace{} \\ -31 \end{array}$$



Reglas de prioridad

Convertir en expresiones aritméticas algorítmicas las siguientes expresiones algebraicas:

a) $5 \cdot (x + y) \rightarrow 5 * (x + y)$

b) $a^2 + b^2 \rightarrow a^{\wedge}2 + b^{\wedge}2$

c) $\frac{x+y}{\frac{u+w}{a}} \rightarrow (x+y) / (u + w/a)$

d) $\frac{x}{y} \cdot (z + w) \rightarrow x / y * (z + w)$



7.2 Expresiones lógicas (booleanas)

- Su valor siempre es verdadero o falso. Recuerde que existen dos constantes lógicas: verdadero (true) y falso (false) y que las variables lógicas pueden tomar sólo estos dos valores.
- En esencia una expresión lógica es una expresión que solo puede tomar dos valores: verdadero y falso.



7.2 Expresiones lógicas (booleanas)

- Las expresiones lógicas se forman combinando constantes lógicas, variables lógicas y otras expresiones lógicas, Se utilizan los operadores lógicos **not**, **and** y **or**, y los operadores relacionales (de relación o comparación) =, <, >, <=, >=, <>.



7.2 Expresiones lógicas (booleanas)

- Operadores lógicos:

Los operadores lógicos o booleanos básicos son **not** (no), **and** (y) y **or** (o).

En la siguiente tabla se resume el funcionamiento de dichos operadores:



7.2 Expresiones lógicas (booleanas)

| Operador lógico | Expresión lógica | Significado |
|-----------------|------------------|---------------------|
| no (not) | no p (not p) | Negación de p |
| y (and) | p y q (p and q) | Conjunción de p y q |
| o (or) | p o q (p or q) | Disyunción de p y q |



7.2 Expresiones lógicas (booleanas)

- Las definiciones de las operaciones no, y, o se resumen en unas tablas conocidas como tablas de la verdad.
- **NOT**: $\text{not } (3 > 8)$ se evalúa como verdadero, ya que $(3 > 8)$ se evalúa como falso.

| a | not a |
|-----------|--------------|
| Verdadero | Falso |
| Falso | Verdadero |



7.2 Expresiones lógicas (booleanas)

- **AND:** (a y b) es verdadera sólo si a y b son verdaderas.

| a | b | a and b |
|-----------|-----------|-----------|
| Verdadero | Verdadero | Verdadero |
| Verdadero | Falso | Falso |
| Falso | Verdadero | Falso |
| Falso | Falso | Falso |

7.2 Expresiones lógicas (booleanas)

- **OR:** $(a \text{ o } b)$ es verdadera cuando a , b o ambas son verdaderas

| a | b | a or b |
|-----------|-----------|-----------|
| Verdadero | Verdadero | Verdadero |
| Verdadero | Falso | Verdadero |
| Falso | Verdadero | Verdadero |
| Falso | Falso | Falso |



7.2 Expresiones lógicas (booleanas)

- En las expresiones lógicas se pueden mezclar operadores de relación y lógicos.

Por ejemplo:

- $(1 < 5)$ y $(5 < 10)$ es verdadera
- $(5 > 10)$ o $(\text{'A'} < \text{'B'})$ es verdadera

7.2 Expresiones lógicas (booleanas)

■ Prioridad de los Operadores Lógicos

Lenguaje Pascal:

| Operador lógico | Expresión lógica |
|---------------------|-----------------------------|
| no (not) | Más alta (1ra. Ejecutada) |
| <hr/> | |
| ^ potencia | |
| /, *, y (and) | ↓ |
| div, mod | |
| +, -, o (or) | |
| <hr/> | |
| <, >, =, <=, >=, <> | Más baja (última ejecutada) |

Al igual que en las expresiones aritméticas, los paréntesis se pueden utilizar y tendrán prioridad sobre cualquier operación.



7.3 Expresiones de relación

- Los operadores relacionales o de relación se pueden aplicar a cualquiera de los cuatro tipos de datos estándar: enteros, real, lógico, carácter.
- Los operadores de relación sirven para expresar las condiciones lógicas en los algoritmos.



7.3 Expresiones de relación

- El formato general para las comparaciones es:

expresión 1 (operador de relación) expresión 2

y el resultado de la operación será verdadero o falso.

Así por ejemplo, si $A=10$ y $B = 7$, entonces $A > B$ es verdadero, mientras que $(A-1) < (B-4)$ es falso.

7.3 Expresiones de relación

- Los operadores de relación son:

| Operador | Significado |
|----------|-------------------|
| < | Menor que |
| > | Mayor que |
| = | Igual que |
| <= | Menor o igual que |
| >= | Mayor o igual que |
| <> | Diferente que |



7.3 Expresiones de relación

La aplicación a valores numéricos es evidente:

| Expresión lógica | Resultado |
|------------------|-----------|
| ■ $3 < 6$ | verdadero |
| ■ $0 > 1$ | falso |
| ■ $4 = 2$ | falso |
| ■ $9 \geq 9$ | verdadero |



7.3 Expresiones de relación

- Si se utilizan los operadores relacionales $=$ y $<>$ para comparar cantidades numéricas, es importante recordar que la mayoría de los valores reales no pueden ser almacenados exactamente. Ej: $(1.0/3.0)*3.0 = 1.0$ teóricamente es verdadera y, sin embargo, al realizar el cálculo en una computadora se puede obtener 0.99999 y en consecuencia el resultado es falso, esto se debe a la precisión limitada de la aritmética real en las computadoras.



7.3 Expresiones de relación

- Comparaciones con datos tipo carácter: se requiere una secuencia de ordenación de los caracteres similar al orden creciente o decreciente. Esta ordenación suele ser alfabética, tanto para mayúsculas como minúsculas, y numéricas, considerándolas de modo independiente. Pero si se consideran caracteres mixtos, se debe recurrir a un código normalizado como es el ASCII.

| ASCII Hex Simbolo | | | ASCII Hex Simbolo | | | ASCII Hex Simbolo | | | ASCII Hex Simbolo | | |
|-------------------|---|-----|-------------------|----|-----|-------------------|----|-----------|-------------------|----|---|
| 0 | 0 | NUL | 16 | 10 | DLE | 32 | 20 | (espacio) | 48 | 30 | 0 |
| 1 | 1 | SOH | 17 | 11 | DC1 | 33 | 21 | ! | 49 | 31 | 1 |
| 2 | 2 | STX | 18 | 12 | DC2 | 34 | 22 | " | 50 | 32 | 2 |
| 3 | 3 | ETX | 19 | 13 | DC3 | 35 | 23 | # | 51 | 33 | 3 |
| 4 | 4 | EOT | 20 | 14 | DC4 | 36 | 24 | \$ | 52 | 34 | 4 |
| 5 | 5 | ENQ | 21 | 15 | NAK | 37 | 25 | % | 53 | 35 | 5 |
| 6 | 6 | ACK | 22 | 16 | SYN | 38 | 26 | & | 54 | 36 | 6 |
| 7 | 7 | BEL | 23 | 17 | ETB | 39 | 27 | ' | 55 | 37 | 7 |
| 8 | 8 | BS | 24 | 18 | CAN | 40 | 28 | (| 56 | 38 | 8 |
| 9 | 9 | TAB | 25 | 19 | EM | 41 | 29 |) | 57 | 39 | 9 |
| 10 | A | LF | 26 | 1A | SUB | 42 | 2A | * | 58 | 3A | : |
| 11 | B | VT | 27 | 1B | ESC | 43 | 2B | + | 59 | 3B | ; |
| 12 | C | FF | 28 | 1C | FS | 44 | 2C | , | 60 | 3C | < |
| 13 | D | CR | 29 | 1D | GS | 45 | 2D | - | 61 | 3D | = |
| 14 | E | SO | 30 | 1E | RS | 46 | 2E | . | 62 | 3E | > |
| 15 | F | SI | 31 | 1F | US | 47 | 2F | / | 63 | 3F | ? |

| ASCII Hex Simbolo | | | ASCII Hex Simbolo | | | ASCII Hex Simbolo | | | ASCII Hex Simbolo | | |
|-------------------|----|---|-------------------|----|---|-------------------|----|---|-------------------|----|---|
| 64 | 40 | @ | 80 | 50 | P | 96 | 60 | ` | 112 | 70 | p |
| 65 | 41 | A | 81 | 51 | Q | 97 | 61 | a | 113 | 71 | q |
| 66 | 42 | B | 82 | 52 | R | 98 | 62 | b | 114 | 72 | r |
| 67 | 43 | C | 83 | 53 | S | 99 | 63 | c | 115 | 73 | s |
| 68 | 44 | D | 84 | 54 | T | 100 | 64 | d | 116 | 74 | t |
| 69 | 45 | E | 85 | 55 | U | 101 | 65 | e | 117 | 75 | u |
| 70 | 46 | F | 86 | 56 | V | 102 | 66 | f | 118 | 76 | v |
| 71 | 47 | G | 87 | 57 | W | 103 | 67 | g | 119 | 77 | w |
| 72 | 48 | H | 88 | 58 | X | 104 | 68 | h | 120 | 78 | x |
| 73 | 49 | I | 89 | 59 | Y | 105 | 69 | i | 121 | 79 | y |
| 74 | 4A | J | 90 | 5A | Z | 106 | 6A | j | 122 | 7A | z |
| 75 | 4B | K | 91 | 5B | [| 107 | 6B | k | 123 | 7B | { |
| 76 | 4C | L | 92 | 5C | \ | 108 | 6C | l | 124 | 7C | |
| 77 | 4D | M | 93 | 5D |] | 109 | 6D | m | 125 | 7D | } |
| 78 | 4E | N | 94 | 5E | ^ | 110 | 6E | n | 126 | 7E | ~ |
| 79 | 4F | O | 95 | 5F | _ | 111 | 6F | o | 127 | 7F | |

<http://www.ascii.cl/es/>

TABLA ASCII ESTANDAR

<http://www.opcionweb.com>

| Oct | Hex | Dec | Carácter | Oct | Hex | Dec | Carácter | Oct | Hex | Dec | Carácter | Oct | Hex | Dec | Carácter |
|-----|-----|-----|--------------------------------|-----|-----|-----|----------|-----|-----|-----|----------|-----|-----|-----|----------|
| 0 | 00 | 0 | NUL NULI | 40 | 20 | 32 | | 100 | 40 | 64 | @ | 140 | 60 | 96 | ` |
| 1 | 01 | 1 | SOH Start Of Heading | 41 | 21 | 33 | ! | 101 | 41 | 65 | A | 141 | 61 | 97 | a |
| 2 | 02 | 2 | STX Start of TeXt | 42 | 22 | 34 | " | 102 | 42 | 66 | B | 142 | 62 | 98 | b |
| 3 | 03 | 3 | ETX End of TeXt | 43 | 23 | 35 | # | 103 | 43 | 67 | C | 143 | 63 | 99 | c |
| 4 | 04 | 4 | EOT End of Transmission | 44 | 24 | 36 | \$ | 104 | 44 | 68 | D | 144 | 64 | 100 | d |
| 5 | 05 | 5 | ENQ ENquiry | 45 | 25 | 37 | % | 105 | 45 | 69 | E | 145 | 65 | 101 | e |
| 6 | 06 | 6 | ACK ACKnowledge | 46 | 26 | 38 | & | 106 | 46 | 70 | F | 146 | 66 | 102 | f |
| 7 | 07 | 7 | BEL BELI | 47 | 27 | 39 | ' | 107 | 47 | 71 | G | 147 | 67 | 103 | g |
| 10 | 08 | 8 | BS BackSpace | 50 | 28 | 40 | (| 110 | 48 | 72 | H | 150 | 68 | 104 | h |
| 11 | 09 | 9 | TAB horizontal TAB | 51 | 29 | 41 |) | 111 | 49 | 73 | I | 151 | 69 | 105 | i |
| 12 | 0A | 10 | LF new Line Feed | 52 | 2A | 42 | ^ | 112 | 4A | 74 | J | 152 | 6A | 106 | j |
| 13 | 0B | 11 | VT Vertical Tab | 53 | 2B | 43 | + | 113 | 4B | 75 | K | 153 | 6B | 107 | k |
| 14 | 0C | 12 | FF new page From Feed | 54 | 2C | 44 | , | 114 | 4C | 76 | L | 154 | 6C | 108 | l |
| 15 | 0D | 13 | CR Carriage Return | 55 | 2D | 45 | - | 115 | 4D | 77 | M | 155 | 6D | 109 | m |
| 16 | 0E | 14 | SO Shift Out | 56 | 2E | 46 | . | 116 | 4E | 78 | N | 156 | 6E | 110 | n |
| 17 | 0F | 15 | SI Shift In | 57 | 2F | 47 | / | 117 | 4F | 79 | O | 157 | 6F | 111 | o |
| 20 | 10 | 16 | DLE Data Link Escape | 60 | 30 | 48 | 0 | 120 | 50 | 80 | P | 160 | 70 | 112 | p |
| 21 | 11 | 17 | DC1 Device Control 1 | 61 | 31 | 49 | 1 | 121 | 51 | 81 | Q | 161 | 71 | 113 | q |
| 22 | 12 | 18 | DC2 Device Control 2 | 62 | 32 | 50 | 2 | 122 | 52 | 82 | R | 162 | 72 | 114 | r |
| 23 | 13 | 19 | DC3 Device Control 3 | 63 | 33 | 51 | 3 | 123 | 53 | 83 | S | 163 | 73 | 115 | s |
| 24 | 14 | 20 | DC4 Device Control 4 | 64 | 34 | 52 | 4 | 124 | 54 | 84 | T | 164 | 74 | 116 | t |
| 25 | 15 | 21 | NAK negative acknowledge | 65 | 35 | 53 | 5 | 125 | 55 | 85 | U | 165 | 75 | 117 | u |
| 26 | 16 | 22 | SYN SYNchronous idle | 66 | 36 | 54 | 6 | 126 | 56 | 86 | V | 166 | 76 | 118 | v |
| 27 | 17 | 23 | ETB End of Transmission. Block | 67 | 37 | 55 | 7 | 127 | 57 | 87 | W | 167 | 77 | 119 | w |
| 30 | 18 | 24 | CAN CANcel | 70 | 38 | 56 | 8 | 130 | 58 | 88 | X | 170 | 78 | 120 | x |
| 31 | 19 | 25 | EM End of Medium | 71 | 39 | 57 | 9 | 131 | 59 | 89 | Y | 171 | 79 | 121 | y |
| 32 | 1A | 26 | SUB SUBstitute | 72 | 3A | 58 | : | 132 | 5A | 90 | Z | 172 | 7A | 122 | z |
| 33 | 1B | 27 | ESC ESCape | 73 | 3B | 59 | ; | 133 | 5B | 91 | [| 173 | 7B | 123 | { |
| 34 | 1C | 28 | FS File Separator | 74 | 3C | 60 | < | 134 | 5C | 92 | \ | 174 | 7C | 124 | |
| 35 | 1D | 29 | GS Group Separator | 75 | 3D | 61 | = | 135 | 5D | 93 |] | 175 | 7D | 125 | } |
| 36 | 1E | 30 | RS Record Separator | 76 | 3E | 62 | > | 136 | 5E | 94 | ^ | 176 | 7E | 126 | ~ |
| 37 | 1F | 31 | US Unit Separator | 77 | 3F | 63 | ? | 137 | 5F | 95 | _ | 177 | 7F | 127 | DELETE |

TABLA ASCII EXTENDIDA

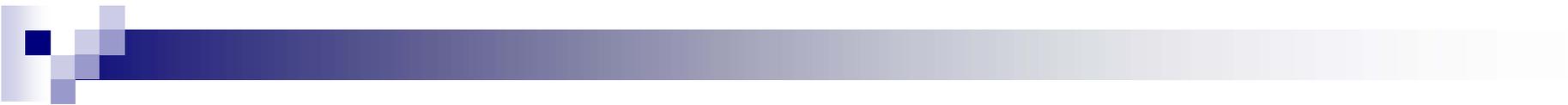
<http://www.opcionweb.com>

| Oct | Hex | Dec | Carácter |
|-----|-----|-----|----------|-----|-----|-----|----------|-----|-----|-----|----------|-----|-----|-----|----------|
| 200 | 80 | 128 | ç | 240 | A0 | 160 | á | 300 | C0 | 192 | Ł | 340 | E0 | 224 | Ó |
| 201 | 81 | 129 | û | 241 | A1 | 161 | í | 301 | C1 | 193 | ł | 341 | E1 | 225 | ó |
| 202 | 82 | 130 | é | 242 | A2 | 162 | ó | 302 | C2 | 194 | Ŧ | 342 | E2 | 226 | Ô |
| 203 | 83 | 131 | â | 243 | A3 | 163 | ú | 303 | C3 | 195 | Ŧ | 343 | E3 | 227 | Õ |
| 204 | 84 | 132 | ä | 244 | A4 | 164 | ñ | 304 | C4 | 196 | — | 344 | E4 | 228 | ö |
| 205 | 85 | 133 | à | 245 | A5 | 165 | Ñ | 305 | C5 | 197 | † | 345 | E5 | 229 | Ö |
| 206 | 86 | 134 | ä | 246 | A6 | 166 | ° | 306 | C6 | 198 | ã | 346 | E6 | 230 | μ |
| 207 | 87 | 135 | ç | 247 | A7 | 167 | ° | 307 | C7 | 199 | Ä | 347 | E7 | 231 | þ |
| 210 | 88 | 136 | è | 250 | A8 | 168 | ¿ | 310 | C8 | 200 | Ł | 350 | E8 | 232 | ƒ |
| 211 | 89 | 137 | é | 251 | A9 | 169 | ® | 311 | C9 | 201 | Ŧ | 351 | E9 | 233 | Ů |
| 212 | 8A | 138 | è | 252 | AA | 170 | ¬ | 312 | CA | 202 | Ŧ | 352 | EA | 234 | Ů |
| 213 | 8B | 139 | ï | 253 | AB | 171 | ½ | 313 | CB | 203 | Ŧ | 353 | EB | 235 | Ů |
| 214 | 8C | 140 | ï | 254 | AC | 172 | ¼ | 314 | CC | 204 | Ŧ | 354 | EC | 236 | ý |
| 215 | 8D | 141 | ì | 255 | AD | 173 | ¡ | 315 | CD | 205 | = | 355 | ED | 237 | Ÿ |
| 216 | 8E | 142 | Ä | 256 | AE | 174 | « | 316 | CE | 206 | † | 356 | EE | 238 | — |
| 217 | 8F | 143 | Ä | 257 | AF | 175 | » | 317 | CF | 207 | ¤ | 357 | EF | 239 | · |
| 220 | 90 | 144 | É | 260 | B0 | 176 | ⋮ | 320 | D0 | 208 | ö | 360 | F0 | 240 | - |
| 221 | 91 | 145 | æ | 261 | B1 | 177 | ⋮ | 321 | D1 | 209 | Đ | 361 | F1 | 241 | ± |
| 222 | 92 | 146 | Æ | 262 | B2 | 178 | ⋮ | 322 | D2 | 210 | È | 362 | F2 | 242 | — |
| 223 | 93 | 147 | ò | 263 | B3 | 179 | ⋮ | 323 | D3 | 211 | È | 363 | F3 | 243 | ‰ |
| 224 | 94 | 148 | ó | 264 | B4 | 180 | — | 324 | D4 | 212 | È | 364 | F4 | 244 | ¶ |
| 225 | 95 | 149 | ò | 265 | B5 | 181 | Δ | 325 | D5 | 213 | ¡ | 365 | F5 | 245 | § |
| 226 | 96 | 150 | û | 266 | B6 | 182 | À | 326 | D6 | 214 | í | 366 | F6 | 246 | ÷ |
| 227 | 97 | 151 | ù | 267 | B7 | 183 | Á | 327 | D7 | 215 | î | 367 | F7 | 247 | · |
| 230 | 98 | 152 | ÿ | 270 | B8 | 184 | © | 330 | D8 | 216 | ï | 370 | F8 | 248 | ° |
| 231 | 99 | 153 | Ö | 271 | B9 | 185 | ª | 331 | D9 | 217 | ĵ | 371 | F9 | 249 | · |
| 232 | 9A | 154 | Û | 272 | BA | 186 | » | 332 | DA | 218 | Ŧ | 372 | FA | 250 | · |
| 233 | 9B | 155 | ø | 273 | BB | 187 | Ŧ | 333 | DB | 219 | ■ | 373 | FB | 251 | ¹ |
| 234 | 9C | 156 | ε | 274 | BC | 188 | Ŧ | 334 | DC | 220 | ■ | 374 | FC | 252 | º |
| 235 | 9D | 157 | ø | 275 | BD | 189 | € | 335 | DD | 221 | — | 375 | FD | 253 | ² |
| 236 | 9E | 158 | × | 276 | BE | 190 | ¥ | 336 | DE | 222 | — | 376 | FE | 254 | ■ |
| 237 | 9F | 159 | f | 277 | BF | 191 | Ŧ | 337 | DF | 223 | ■ | 377 | FF | 255 | — |



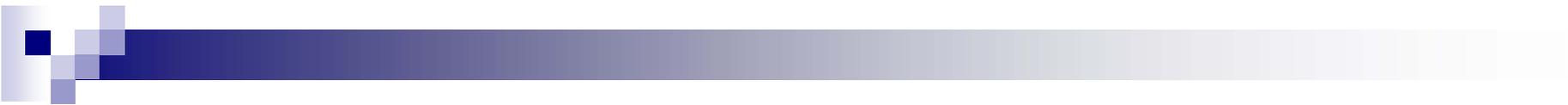
7.3 Expresiones de relación

- Estos códigos normalizados son:
 - Los caracteres especiales #, %, \$, (,), +, -, /, exigen la consulta del código de ordenación de su computadora (normalmente es el ASCII).
 - Los valores de los caracteres que representan a los dígitos están en su orden natural. Esto es $0 < 1$, $1 < 2$, $2 < 3$, ..., $8 < 9$.
 - Las letras mayúsculas A a Z siguen el orden alfabético ($A < B$, $C < F$, etc.).
 - Si existen letras minúsculas siguen el mismo criterio alfabético ($a < b$, $c < h$, etc.).



7.3 Expresiones de relación

- Cuando se utilizan los operadores de relación con valores lógicos, la constante false es menor que la constante true, por definición: `false < true`; `true > false`.



8. Funciones internas

- Las funciones internas son operadores especiales.
- Por ejemplo, la función **ln** se puede utilizar para determinar el logaritmo neperiano de un número y la función **SQRT** calcula la raíz cuadrada de un número positivo.
- Funciones internas más usuales:

8. Funciones internas

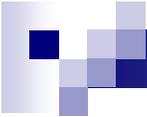
| Función | Descripción | Tipo de argumento | Resultado |
|-----------|--------------------------|-------------------|---------------------|
| abs(x) | Valor absoluto de x | Entero o real | Igual que argumento |
| arctan(x) | Arco tangente de x | Entero o real | Real |
| cos(x) | Coseno de x | Entero o real | Real |
| exp(x) | Exponencial de x | Entero o real | Real |
| ln(x) | Logaritmo neperiano de x | Entero o real | Real |
| log10(x) | Logaritmo decimal de x | Entero o real | Real |
| round(x) | Redondeo de x | Real | Entero |
| sin(x) | Seno de x | Entero o real | Real |
| sqr(x) | Cuadrado de x | Entero o real | Igual que argumento |
| sqrt(x) | Raíz cuadrada de x | Entero o real | Real |
| trunc(x) | Truncamiento de x | Real | Entero |



9. Escritura de algoritmos/programas

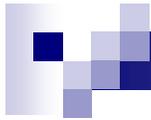
Un algoritmo constará de dos componentes:

- una cabecera: es una instrucción simple que comienza con la palabra algoritmo, seguida del nombre asignado al algoritmo completo.
- un bloque algoritmo: es el resto del algoritmo y a su vez consta de dos componentes o secciones:
 - las acciones de declaración: definen o declaran las variables y constantes, además de los tipos de dato.
 - las acciones ejecutables: son las acciones que debe ejecutar la computadora cuando se convierta en programa al algoritmo.

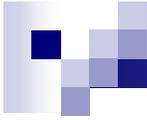


Escriba los siguientes algoritmos que:

1. Calcule la superficie de un círculo y la longitud de su circunferencia.
2. Determine el salario neto semanal de un trabajador, si sabe que los impuestos aplicados equivalen al 25% del salario bruto. Debe visualizar por pantalla el nombre del trabajador, la cantidad de horas que trabajó en la semana, los impuestos a descontar y el salario neto.



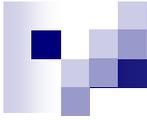
3. Emita una factura correspondiente a la compra de n unidades de un solo tipo de artículo. El IVA a aplicar es del 12%.
4. Intercambie los valores de dos variables reales A y B .



5. Se desea determinar la cantidad de leche que produce una finca mensualmente. Dicha cantidad depende de la edad de las vacas. Si la vaca tiene entre 0 y 2 años produce 6 litros diarios, si tiene entre 3 y 5 años produce 10 litros diarios, pero si tiene más de 5 años produce 4 litros diarios. La cantidad de vacas que tiene la finca de cada tipo es un dato que varía. Escriba un algoritmo (en pseudocódigo y diagrama de flujo), que calcule la cantidad de leche total producida en la finca y discriminada por edad de la vaca. Realice el análisis de entrada, proceso y salida.



6. El dueño de un galpón desea calcular la cantidad de baldosas de 20x20 cm que debe comprar, para cubrir el piso de su galpón rectangular y la cantidad de litros de pintura para pintar las cuatro paredes internas, si un litro de pintura alcanza para cubrir 3 m². Escriba un algoritmo que le permita a este señor hacer estos cálculos. Realice el análisis de entrada, proceso, salida y diseñe el algoritmo correspondiente en pseudocódigo y diagrama de flujo.



7. Debido a la inseguridad, en su casa deciden construir un muro alrededor del terreno que tiene forma rectangular, para instalar una cerca eléctrica en la parte superior del mismo. Escriba un algoritmo que le permita a su familia calcular: a) la longitud total del muro y b) la cantidad de metros de cable que debe comprar, si la cerca consta de tres líneas de cable. Realice el análisis de entrada, proceso, salida y diseñe el algoritmo correspondiente en pseudocódigo y diagrama de flujo.