ÍNDICE

PRÓLOGO

El libro ha sido escrito para servir de texto a los estudiantes de Técnico Medio de los Institutos Politécnicos en la especialidad de Informática en nuestro país.

El texto fue concebido por profesores de Lógica de Programación de los Institutos Politécnicos “Juan Manuel Castiñeiras” del Mariel y “Mártires de Girón” de Playa, que desde la creación de esta nueva especialidad, recopilamos información mediante texto que se adaptaban a los temas, y de conferencias recibidas.

Pretendemos que los estudiantes encuentren la información necesaria de cada uno de los temas y tengan la posibilidad de estudiar de forma independiente.

Aspiramos a que los profesores encuentren en el libro los contenidos básicos del programa.

Agradecemos nos hagan llegar los criterios o sugerencias que estiman convenientes.

Los autores.

CAPITULO No 1: SISTEMA DE NUMERACIÓN

Introducción:

El hombre desde épocas muy primitivas, trató de representar cantidades de manera simbólica; primeramente, haciendo analogías muy elementales, como fue la utilización de los dedos de la mano y cuando estos no alcanzaron ideó hacer marcas sobre una madera, o simplemente cogiendo objetos equivalentes a la cantidad que quería representar. De esta forma muy elemental surgió la necesidad de contar primeramente y posteriormente la necesidad de realizar operaciones muy elementales como por ejemplo sumar o restar dos cantidades.

Las civilizaciones antiguas alcanzaron determinado grado de desarrollo, como fue el caso de los romanos, idearon un sistema de numeración para representar las cantidades y poderlas contar. Este sistema, conocido como el **Sistema romano**, actualmente se usa muy poco por las limitaciones que presenta; no es tan primitivo como el usado en épocas remotas, pero tiene grandes deficiencias; ya que mismo asigna cantidades a determinados símbolos, como por ejemplo: el número, lo representa con el símbolo **I**; el 5, por **V**; etc. Este sistema presenta dos grandes dificultades, puesto que a medida que las cantidades a representar aumentan, tiene que ir aumentado el número de símbolos. Además, una dificultad fundamental que en el campo de las matemáticas lo condenó a desaparecer, es que no es posible realizar con este sistema de símbolos operaciones matemáticas.

Fueron los árabes los que idearon el sistema de numeración utilizado en la actualidad; todo el mundo conoce este sistema por su utilidad diaria.

Este sistema lo conocemos actualmente como **Sistema decimal,** y tiene la característica de tener una cantidad fija de símbolos par a representar todos los números; a cada una de estos símbolos se les llama dígito. Además, tiene otra propiedad y es que en cada uno de los dígitos, en un numero cualquiera está modificado su valor por la base del sistema elevado a una potencia progresiva, dependiendo de la posición que este dígito ocupa en el número dado.

Aunque todos, incluso los niños más pequeños estamos familiarizados con el sistema decimal para realizar operaciones aritméticas, al estudiar otros sistemas de numeración quedamos sorprendidos por los aspectos que estas operaciones incluyen y que realmente hacemos de manera mecánica.

Por tal razón, para el estudio de los sistemas de numeración nos basaremos en el sistema decimal, que ya conocemos, ya que cuando estudiamos y entendemos las leyes del sistema decimal estos facilitan su comprensión de otros sistemas que tienen las mismas propiedades pero que no son tan evidentes por no estar familiarizados con ellos.

Pasemos pues, al estudio de los sistemas de numeración que tienen importancia particular en el estudio de la técnica digital.

Un sistema de numeración, no es más que un conjunto de símbolos que sirve para representar cantidades y que deben cumplir determinados requisitos o leyes.

Ejemplo 1:

El sistema decimal tiene 10 dígitos o símbolos:

0, 1, 2, 3, 4, 5,6, 7, 8 y 9.

1.1 SISTEMA DECIMAL:

Cuando estamos en presencia de un número decimal, raramente nos detenemos a pensar como está constituido.

Tomemos por ejemplo: número 342; sabemos inmediatamente que cantidad representa; pero, no como está formado el mismo.

Veremos que el número 342 podemos interpretarlo de la forma siguiente:

$$
\begin{aligned}
342 &= 3 \cdot 100 + 4 \cdot 10 + 2 \\
&= 3 \cdot 10^2 + 4 \cdot 10^1 + 2 \cdot 10^0
\end{aligned}
$$

Podemos observar como se aplica la idea del sistema decimal en el cual el número 10, que es la base del sistema, elevado a la potencia cero y multiplicado por el dígito situado más a la derecha, da el valor que representa dicho dígito ($2 \cdot 10^0$). La misma base del sistema, elevado a la primera potencia y multiplicado por el dígito que le sigue en orden hacia la izquierda: ($4 \cdot 10^1$) da el valor que representa este segundo dígito; por último, la base del sistema, elevado a la segunda potencia y multiplicado por el tercer dígito (dígito que sigue en orden hacia la izquierda), da el valor que representa dicho dígito.

Del análisis anterior podemos generalizar una propiedad que se cumple para todos los sistemas de numeración de base fija, y es que para obtener el valor que representa un número que pertenece a cualquier base **b** es necesario multiplicar el primer dígito por la base del sistema elevada a la potencia cero, y sumarlo con el segundo dígito multiplicado por la base del sistema elevada a la primera potencia y este resultado sumarlo con el tercer dígito multiplicado por la base elevada a la segunda potencia, y así sucesivamente.

Generalmente, podemos escribir la ecuación que expresa el valor de un número entero en cualquier sistema de base fija:

$$\mathbf{N = d_{n-1}b^{n-1} + d_{n-2}b^{n-2} + \cdots + d_2b^2 + d_1b^1 + d_0b^0} \qquad (1.1)$$

donde:

N:----------------------- número al que queremos hallar su valor.

n:------------------------ cantidad de dígitos de dicho número.

$d_{n-1}, d_{n-2,\ldots,} d_{2,} d_{1,} d_0$:---- dígitos del número

b:------------------------ base a la que pertenece el número.

b ≥1:-------------------- donde b es cualquier número entero, mayor que 1.

$0 \leq d \leq b-1$:------------ los dígitos son cualquier número entero dado en el rango especificado.

<u>Ejemplo 2:</u>

El numero N = 2 541, en base diez, podemos expresarlo de la siguiente forma:

N = 2 541

$N = 2 \cdot 10^3 + 5 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0$

Aplicando a este ejemplo la fórmula (1.1) tenemos:

N = 2 541
n = 4
$d_3 = 2, d_2 = 5, d_1 = 4, d_0 = 1$
b = 10

En cualquier sistema de base fija, la base del sistema es a la cantidad de dígitos con que cuenta el sistema, pero nunca es un dígito del mismo, así tenemos que:

Base	Dígitos del sistema	Sistema
2	0, 1	binario
8	0, 1, 2, 3, 4, 5, 6, 7	octal
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	decimal
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	hexadecimal

Los sistemas expresados anteriormente son los más utilizados prácticamente en la técnica de computación.

1.2 SISTEMA BINARIO:

Como su nombre lo indica, la base del sistema binario es dos (b = 2) y por tal; razón tiene dos dígitos el cero (0) y el uno (1).

Tomando la expresión general de un número entero en un sistema de base fija, tenemos:

$$\mathbf{N = d_{n-1}b^{n-1} + d_{n-2}b^{n-2} + \cdots + d_2b^2 + d_1b^1 + d_0b^0}$$

Para el sistema binario: b = 2.

luego: d_{n-1}, d_2, d_1, d_0 son: 1 o 0

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

Por tal razón la fórmula general para el sistema binario podemos escribirla en la forma siguiente:

$$N = d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \cdots + d_22^2 + d_12^1 + d_02^0$$

$$N = d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \cdots + d_24 + d_12 + d_0$$

Ejemplo 3:

$N = 1101$

$N = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

$N = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$

$N = 8 + 4 + 0 + 1$

$N = 13$

A continuación damos una tabla con las equivalencias de los dígitos decimales y el binario:

Decimal	**Binario**
0	0 ($0 \bullet 2^0$)
1	1 ($1 \bullet 2^0$)
2	10 ($1 \bullet 2^1 + 0 \bullet 2^0$)
3	11 ($1 \bullet 2^1 + 1 \bullet 2^0$)
4	100 ($1 \bullet 2^2 + 0 \bullet 2^1 + 0 \bullet 2^0$)
5	101 ($1 \bullet 2^2 + 0 \bullet 2^1 + 1 \bullet 2^0$)
6	110 ($1 \bullet 2^2 + 1 \bullet 2^1 + 0 \bullet 2^0$)
7	111 ($1 \bullet 2^2 + 1 \bullet 2^1 + 1 \bullet 2^0$)
8	1 000 ($1 \bullet 2^3 + 0 \bullet 2^2 + 0 \bullet 2^1 + 0 \bullet 2^0$)
9	1 001 ($1 \bullet 2^3 + 0 \bullet 2^2 + 0 \bullet 2^1 + 1 \bullet 2^0$)

Claramente se ve que el sistema binario es engorroso para operar en él, pues en ocasiones, como es el caso del dígito decimal 9, son necesarios 4 dígitos en base dos. Sin embargo, el gran uso del sistema binario, fundamentalmente en los sistemas de computación, es debido a la facilidad con que se puede representar los dos dígitos del sistema binario, 0 y 1. Puesto que sólo tenemos que representar dos dígitos, esto lo podemos lograr con cualquier dispositivo que solamente pueda adoptar dos estados, como por ejemplo una lámpara encendida o apagada, un interruptor abierto o cerrado, por un transistor saturado o en corte, o simplemente por un nivel de tensión dado y otro nivel de tensión diferente del primero.

1.3 CONVERSIONES:

La conversión de números a diferentes sistemas, es necesaria por múltiples motivos:

- El sistema decimal es el sistema con que el hombre está familiarizado para resolver problemas matemáticos de cualquier grado de complejidad.
- Los sistemas de computación solamente pueden operar con el sistema binario.

Lo que hace necesaria la conversión del sistema decimal al binario y viceversa.

1.3.1 **Conversión de números de diferentes bases:**

Como hemos visto, la representación del dígito decimal que en el sistema binario tiene cuatro dígitos (1001); por tanto, el sistema binario es extremadamente engorroso para representar cantidades aún pequeñas. Sin embargo, estas cantidades binarias se pueden convertir con relativa facilidad a otros sistemas que no son tan engorrosos como el sistema binario, y que son el sistema octal y el sistema hexadecimal.

Esta última razón hace que surja la necesidad de la conversión entre los sistemas binario y octal, y viceversa; así como la conversión entre el sistema binario y el sistema hexadecimal y viceversa.

Estas últimas conversiones son particularmente útiles en los sistemas de computación.

Para facilitar la conversión de números entre las bases diferentes y para evitar ambigüedades, se acostumbra a escribir como subíndice la base del sistema a que pertenece el número.

Ejemplo 4:

110_2 en base 2

25_{10} en base 10

16_8 en base 8

34_{16} en base 16

1.3.1.1 **Conversión de un número entero de cualquier base a su equivalente decimal:**

Para la conversión de un número entero de cualquier base, a su equivalente en el sistema decimal, se expande el número entero como un polinomio, en potencias progresivas de la base a que pertenece el número y se suman los términos (aplicación de la fórmula general 1.1).

Ejemplo 5:

a) Convertir el número 10011_2 a su equivalente en el sistema decimal.

$$10011_2 = 1 \bullet 2^4 + 0 \bullet 2^3 + 0 \bullet 2^2 + 1 \bullet 2^1 + 1 \bullet 2^0$$

$$10011_2 = 1 \bullet 16 + 0 \bullet 8 + 0 \bullet 4 + 1 \bullet 2 + 1 \bullet 1$$

$$10011_2 = 16 + 2 + 1$$

$10011_2 = 19_{10}$

b) Convertir el número 637_8 a su equivalente decimal.

$637_8 = 6 \bullet 8^2 + 3 \bullet 8^1 + 7 \bullet 8^0$

$637_8 = 6 \bullet 64 + 3 \bullet 8 + 7 \bullet 1$

$637_8 = 384 + 24 + 7$

$637_8 = 415_{10}$

c) Convertir el número $3A4_{16}$ a su equivalente decimal.

$3A4_{16} = 3 \bullet (16)^2 + A \bullet (16)^1 + 4 \bullet (16)^0$

En el sistema hexadecimal A = 10.

$3A4_{16} = 3 \bullet 256 + 10 \bullet 16 + 4 \bullet 1$

$3A4_{16} = 768 + 160 + 4$

$3A4_{16} = 932_{10}$

1.3.1.2 **Conversión de un número decimal entero a su equivalente en cualquier base:**

Para la conversión de un número decimal entero a su equivalente en cualquier base se divide dicho número sucesivamente por la base a que se quiere convertir el número decimal, y los residuos obtenidos de las divisiones sucesivas, son los dígitos del número en la nueva base, tomando el último residuo como el primer dígito del nuevo número, o sea, como el dígito más significativo.

Ejemplo 6:

a) Convertir el 19_{10} a su equivalente en base 2.

19 : 2 = 9 residuo 1 (dígito menos significativo)

9 : 2 = 4 residuo 1

4 : 2 = 2 residuo 0

$\boxed{19_{10} = 10011_2}$

2 : 2 = 1 residuo 0

1 : 2 = 0 residuo 1 (dígito más significativo)

b) Convertir el número 415_{10} a su equivalente en base 8.

415 : 8 = 51 residuo 7

51 : 8 = 6 residuo 3

6 : 8 = 0 residuo 6

$$\boxed{415_{10} = 637_8}$$

c) Convertir el número 932_{10} a su equivalente en base 16.

932 : 16 = 58 residuo 4

58 : 16 = 3 residuo 10 = A

3 : 16 = 0 residuo 3

$$\boxed{932_{10} = 3\ A\ 4_{16}}$$

Hasta aquí hemos estudiado en primer lugar la conversión de los números de base 2, 8 y 16 al sistema decimal y aunque estas tres conversiones obedecen a la misma ley, queremos destacar que la más utilizada es la conversión **binario – decimal**.

Como segundo aspecto vimos la conversión de los números decimales a su equivalente en cualquiera de las bases 2, 8 y 16, y en este caso ocurre igual que la más utilizada es la conversión **decimal – binario.**

1.3.2 **Conversión del sistema binario al sistema octal y viceversa:**

Estas conversiones tienen una importancia particular para los estudiantes que, de una forma directa o indirecta, están relacionados con la computación.

La conversión entre los sistemas binarios y octal, es en extremo fácil y es esta característica precisamente la que le da importancia al sistema octal, constituyendo una herramienta muy útil para la computación.

1.3.2.1 **<u>Conversión del sistema octal al sistema binario:</u>**

La conversión del sistema octal al sistema binario, podríamos hacerla con divisiones sucesivas del número octal por la base del sistema binario, y los residuos obtenidos de estas divisiones, son los dígitos del número del sistema en la nueva base.

<u>Ejemplo 7:</u>

Convertir el número 432_8 a su equivalente en el sistema binario.

432 : 2 = 215	residuo 0
215 : 2 = 106	residuo 1
106 : 2 = 43	residuo 0
43 : 2 = 21	residuo 1
21 : 2 = 10	residuo 1
10 : 2 = 4	residuo 0
4 : 2 = 2	residuo 0
2 : 2 = 1	residuo 0
1 : 2 = 0	residuo 1

$432_8 = 100011010$

A primera vista puede parecer que el procedimiento es incorrecto, aunque el resultado si lo es, si analizamos detenidamente nos damos cuenta que tanto el resultado como el procedimiento son correctos, pero debemos notar que el número 215 está expresado en notación octal. Analicemos como fue el proceso de la primera división 432/2 = 215 y como residuo obtuvimos 0, al efectuar esta división procedimos de la siguiente forma: 4 entre 2 = 2, 3 entre 2 = 1 y el residuo es 1; entonces, efectuamos la última división: 12 entre 2 = 5 y residuo cero, y finaliza la primera división que a primera vista no parece correcta. Pero, se debe notar que 12 es un número en base octal y su equivalente decimal es 10 ($12_8 = 10_{10}$) por tanto $12_{8/2} = 10_{10/2} = 5$ y el residuo es 0. Por tal razón 432: 2 = 215 y el residuo es cero.

Si continuamos todo el proceso teniendo en cuenta los mismos factores que hemos tenido en la división precedente, nos daremos cuanta que todo el proceso es correcto.

Sin embargo, anteriormente hemos dicho que entre las ventajas del sistema octal estaba, que la conversión entre dicho sistema y el binario es muy sencillo lo que permite representar cantidades binarias que son muy engorrosas, de una forma más práctica. Por otra parte hemos visto en el ejemplo anterior, que la conversión entre el sistema y el binario es muy laborioso.

Para vencer esta dificultad existe un procedimiento práctico, que nos conduce al mismo resultado, lo que permite representar los números binarios que son muy engorrosos en su equivalente en el sistema octal.

<u>Ejemplo 8</u>:

Veamos el procedimiento en el ejemplo anterior:

$$432_8 = \begin{array}{ccc} 4 & 3 & 2 \\ \downarrow & \downarrow & \downarrow \end{array}$$

$432_8 =$ 100 011 010

Como vemos, hemos obtenido el mismo resultado en un solo paso.

El procedimiento que hemos seguido para ello consiste en que, por cada dígito octal se escribe su equivalente binario con tres dígitos, y por último agrupamos estos dígitos binarios en un solo número.

Esto es posible porque ocho es potencia de 2 (2^3), y por tal razón cada número octal tiene su correspondiente binario de forma biunívoca. A continuación mostramos esta correspondencia:

OCTAL	BINARIO
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Veamos otro ejemplo que nos ayude a aclarar el método práctico.

Ejemplo 9:

$73452_8 =$

7 = 111
3 = 011
4 = 100
5 = 101
2 = 010

$73452_8 = 111011100101010_2$

1.3.2.2 **<u>Conversión del sistema binario al sistema octal:</u>**

La conversión del sistema binario al sistema octal es tan fácil como la que acabamos de ver del sistema octal al binario.

Para la conversión binario octal, tomamos el número binario y lo dividimos en grupos de tres dígitos, pero teniendo en cuenta que debemos comenzar del extremo derecho hacia el izquierdo. Por cada grupo binario de tres dígitos escribimos su equivalente octal de acuerdo a la tabla de equivalencias entre los sistemas que hemos dado anteriormente. Si el último grupo de dígitos es de menos de tres, se completa hasta tres, con ceros.

<u>Ejemplo 10:</u>

Convertir el número 101110110_2 a su equivalente octal.

$$\begin{array}{lccccc} 101110110_2 = & 101 & , & 110 & , & 110 \\ & \downarrow & & \downarrow & & \downarrow \\ & 5 & & 6 & & 6 \end{array} \qquad 101110110_2 = 566_2$$

Veamos un segundo ejemplo para insistir en un aspecto que a veces ocasiona errores.

<u>Ejemplo 11:</u>

Convertir al sistema octal el número 1010011101_2.

Si empezamos a separar los grupos de tres, comenzando por la izquierda y hacia la derecha, obtendríamos 101, 001, 110 y 1; escribiendo el equivalente en cada grupo de esta forma, llegaríamos al siguiente resultado erróneo: 5161. Lo correcto sería separar los grupos comenzando por la derecha, hacia la izquierda.

$$\begin{array}{lcccc} 1010011101_2 = & 1 & 010 & 011 & 101 \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ & 001 & 2 & 3 & 5 \\ & \downarrow & & & \\ & 1 & & & \end{array}$$

y obtendremos: $1010011101_2 = 1235_8$

Cuando leemos un número binario de cierta cantidad de dígitos, éste es difícil de reconocer y más aún recordar, como por ejemplo, el número binario 1011111001101; sin embargo es más fácil de leer y recordar 27455_8.

Algunas computadoras dan su resultado en octal, o sea, la computadora realiza sus operaciones en binario y al ofrecer los resultados lo hacen en sistema octal, de forma tal que la persona que use la computadora puede interpretar esto de una forma más sencilla.

Claramente, si la computadora brindara la respuesta en el sistema decimal, esto sería mucho más fácil de interpretar; pero, para realizar esta conversión la computadora requiere de circuitos apropiados que realicen dicha conversión del sistema binario (sistema con que opera la computadora internamente), al sistema decimal (sistema con que está familiarizado el hombre).

Muy pocas computadoras dan su respuesta en el sistema binario, otras usan la conversión al sistema octal, que es muy sencilla; y la mayoría entregan la respuesta a la salida en el sistema decimal.

1.3.3 **Conversión del sistema binario al hexadecimal y viceversa:**

Paras hacer la conversión del sistema octal al binario y viceversa, explicamos un método práctico basado en la tabla de equivalencias entre los dos sistemas; en el presente epígrafe haremos la conversión entre el sistema hexadecimal y el binario, y viceversa, utilizando la tabla de equivalencias.

HEXADECIMAL	BINARIO
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A - 10	1 0 1 0
B - 11	1 0 1 1
C - 12	1 1 0 0
D - 13	1 1 0 1
E - 14	1 1 1 0
F - 15	1 1 1 1

En el sistema hexadecimal, la base del sistema es 16, que también es una potencia de dos ($2^4 = 16$). Esto implica una correspondencia biunívoca entre cada dígito del sistema hexadecimal con un grupo de cuatro dígitos del sistema binario, como refleja la tabla anterior. Por esta razón la conversión entre los sistemas hexadecimal y binario es muy sencilla y similar a la conversión entre los sistemas octal y binario.

1.3.3.1 **<u>Conversión del sistema hexadecimal al sistema binario:</u>**

Para la conversión del sistema hexadecimal al binario, escribimos por cada dígito hexadecimal su equivalente en binario de acuerdo a la tabla, como ilustramos a continuación:

<u>Ejemplo 12:</u>

Convertir el número hexadecimal 9 4 A 3 a su equivalente en el sistema binario.

$$
\begin{array}{lcccc}
9\,4\,A\,3_{16} = & 9 & 4 & A & 3 \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 1001 & 0100 & 1010 & 0011
\end{array}
$$

$9\,4\,A\,3_{16} = 1001010010100011_2$

1.3.3.2 **<u>Conversión del sistema binario al sistema hexadecimal:</u>**

Para la conversión del sistema binario al sistema hexadecimal, separamos el número binario en grupos de cuatro dígitos binarios, comenzando por la derecha y hacia la izquierda, y posteriormente escribimos el equivalente hexadecimal por cada grupo de cuatro dígitos binarios, de acuerdo con la tabla de equivalencias.

<u>Ejemplo 13:</u>

Convertir el siguiente número binario a su equivalente en el sistema hexadecimal: 11010001011_2.

$$
\begin{array}{lccc}
11010001011_2 = & 110 & 1000 & 1011 \\
 & \downarrow & \downarrow & \downarrow \\
 & 6 & 8 & B
\end{array}
$$

$11010001011_2 = 6\,8\,B_{16}$

Al final del capítulo proponemos una serie de ejercicios (III) de conversión entre los diferentes sistemas, pues es necesario que el alumno domine estos aspectos de forma práctica y pueda realizar estas conversiones de forma rápida. Queremos destacar que las conversiones más utilizadas son las de los sistemas octal, decimal y hexadecimal a binario y viceversa.

1.4 <u>SUMA Y RESTA EN EL SISTEMA BINARIO:</u>

En un proceso de suma aritmética, siempre que sumemos dos dígitos y su resultado sea mayor que la base, se produce un acarreo de una unidad que se suma con la posición siguiente. Así en cualquier sistema de numeración de base b, el mayor dígito que aparece en cualquier número es b-1, si la operación de suma de dos dígitos excede este valor, entonces se produce el acarreo de una unidad, la cual se suma a la siguiente posición.

Veamos un ejemplo de lo que planteamos en el sistema decimal:

Ejemplo 14:

a)
$$\begin{array}{r} 5 \\ +\ 7 \\ \hline 12 \end{array}$$

1 acarreo que se suma a la siguiente posición

b)
$$\begin{array}{r} 14 \\ +\ 8 \\ \hline 22 \end{array}$$

1.4.1 **Suma en el sistema binario:**

El proceso de suma en el sistema binario, es más simple que en el sistema decimal, si tenemos en cuenta que sólo vamos a tener cuatro combinaciones posibles cuando vayamos a sumar dos dígitos binarios. Esto lo ilustramos en la tabla 1.1, en la que nos planteamos la suma de dos dígitos binarios A y B, y en la cual quedan definidas las reglas para la suma en el sistema binario.

Tabla 1.1 Suma binaria de A y B

A	B	Suma	Acarreo
0	0	0 + 0 = 0	0
0	1	0 + 1 = 1	0
1	0	1 + 0 = 1	0
1	1	1 + 1 = 0	1

Como hemos visto en el sistema binario, el mayor dígito que podemos tener es 1; así como, en el sistema decimal el mayor dígito es el 9. Por esta razón cuando en el sistema binario sumamos 1 + 1 el resultado es cero y se suma un acarreo a la siguiente posición.

Veamos a continuación dos ejemplos de suma en el sistema binario y su similitud con los equivalentes decimales de los números:

Ejemplo 15:

a)	111	← acarreo →	1
	100101		37
	+ 1101		+ 13
	110010		50

b)	111	← acarreo →	1
	1011011		91
	+ 1011010		+ 90
	10110101		181

Inicialmente, quien no este familiarizado con la suma binaria podrá parecerle difícil; pero, nos podemos dar cuenta en breve tiempo, que en realidad resulta mucho más sencilla que la suma en el sistema decimal. Esto puede notarse a priori si se da cuenta, que como resultado de la suma de dos dígitos binarios solamente pueden ser dos los resultados, 1 o 0 y que con un poco de práctica nos podemos puede familiarizar rápidamente con la suma en el sistema binario.

1.4.2 **Sustracción en el sistema binario:**

La resta en el sistema binario tiene un grado de dificultad mayor que la suma; pero, en realidad también es muy sencillo teniendo en cuenta que son cuatro las diferentes posibilidades al restar dos dígitos binarios.

Cuando vamos a restar dos dígitos, de forma tal que el minuendo es mayor que el sustraendo, la diferencia se obtiene normalmente restando los dos dígitos y el resultado es también un dígito que puede ser 0 ó 1. Sin embargo, cuando el sustraendo es mayor que el minuendo, es necesario pedir prestada una unidad al dígito que ocupa la posición siguiente; esta unidad se refleja o representa para el dígito que pidió préstamo, como la base del sistema b que en este caso es dos (10), y la posición que prestó la unidad, en caso de tenerla, se quedó con cero.

De la misma forma que para la suma, a continuación veremos la resta de dos dígitos binarios A y B, que también tiene cuatro posibilidades (Tabla 1.2).

Tabla 1.2: Resta binaria de A y B

A	B	Diferencia	Préstamo
0	0	0 - 0 = 0	0
* 0	1	0 - 1 = 1	1
1	0	1 - 0 = 1	0
1	1	1 - 1 = 0	0

Como en el caso de la resta decimal se ve, que cuando un dígito es menor se resta de uno mayor, no hay ningún tipo de dificultad, y se efectúa la diferencia; pero, cuando el dígito el mayor se resta del menor, es necesario pedir prestada una unidad al dígito de al lado, como es el caso marcado con un (*) en la tabla 1.2, en la cual 0 – 1 no se puede efectuar directamente; por lo que el cero pide prestada una unidad, que al prestársela, se refleja en él como base del sistema, o sea, dos (10); de forma tal, que o – 1 se convierte en 10 – 1 = 1 (2 – 1) y el préstamo de una unidad a la posición siguiente:

Ejemplo 16:

a)
$$\begin{array}{r} 11110 \\ -\ 1010 \\ \hline 10100 \end{array} \qquad \begin{array}{r} 30 \\ -\ 10 \\ \hline 20 \end{array}$$

b)
$$\begin{array}{r} 1011011 \\ -\ 10101 \\ \hline 1000110 \end{array} \qquad \begin{array}{r} 91 \\ -\ 21 \\ \hline 70 \end{array}$$

1.5 Notación complementaria en el sistema binario:

Cualquier sistema de numeración de base fija permite representar sus números negativos en la forma conocida como **complementación de la base** o verdadera complementación. En el sistema binario nos referiremos a ésta, como la notación en **dos complemento**; además, en el sistema binario existe otra notación que se conoce con el nombre de **uno complemento**.

1.5.1 Uno complemento:

El uno complemento de un número N, se define: como el resultado de restar dicho número de $b^n - 1$, donde b es la base del sistema y n es el número de dígitos que tiene el número.

En el sistema binario b = 2, podríamos expresar el uno complemento de un número N como:

$$\text{Uno complemento de } N = 2^n - 1 - N \qquad (1.2)$$

donde n es el número de bits del número N.

Ejemplo 17:

Halle el uno complemento del número binario 1011101.

El número N tiene 7 bits $n = 7$

Aplicando la fórmula (1.2) tendremos:

El uno complemento de $N = 2^n - 1 - N$

$= 2^7 - 1 - 1011101$

$= 10000000 - 1 - 1011101$

$= 1111111 - 1011101$

$N = 0100010$

Si comparamos el número N = 1011101 y el uno complemento 0100010, nos daremos cuenta que el uno complemento podemos obtenerlo, cambiando todos los unos por ceros y todos los ceros por unos, en un número dado.

1.5.2 **Dos complemento:**

El dos complemento de un número N, se define: como el número obtenido de sustraer dicho número de 2^n; como se ve, el dos complemento es igual al uno complemento, más uno.

Así, el dos complemento de $N = 2^n - N$,   (1.3)
donde n sigue siendo la cantidad de bit del número N.

Si comparamos esta ecuación (1.3) con la (1.2) que define el uno complemento de un número, llegamos a la conclusión de que el dos complemento es igual al uno complemento, más uno.

Comparando las ecuaciones (1.2) y (1.3)

Uno complemento de N = $2^n - 1 - N$ (1.2)

Dos complemento de N = $2^n - N$ (1.3)

$$2^n - N = 2^n - 1 - N + 1$$

Ejemplo 18:

Halle el dos complemento del número binario N = 1011101.

Como N = 1011101 tiene 7 bits n = 7

El dos complemento de N = $2^n - N$

$= 2^7 - 1011101$

$= 10000000 - 1011101$

N = 0100011

De forma mucho más práctica, podríamos obtener el dos complemento, esto es: hallamos el uno complemento del número y le sumamos uno, y llegamos al mismo resultado sin necesidad de realizar tantas operaciones:

Ejemplo 19:

N = 1011101

Uno complemento de N = 0100010

Dos complemento de N = 0100010 + 1

Dos complemento de N = 0100011

Como vemos hemos llegado al mismo resulta \do más fácil y rápidamente.

Como dijimos anteriormente, el dos complemento de un número no es más que la representación negativa de dicho número; por tanto, para realizar una sustracción utilizando el dos complemento, se convierte el proceso de sustracción en suma.

1.5.3 **Resta por el método del dos complemento:**

Para hallar la diferencia por el método del dos complemento, es necesario primeramente completar la cantidad de bits del sustraendo, de forma tal que tenga la misma cantidad de bits que tiene el minuendo y posteriormente hallar la diferencia.

Para hallar la diferencia de dos números binarios por el método de dos complemento, una vez que el sustraendo tiene la misma cantidad de bits que el minuendo, procedemos a hallar el dos complemento del sustraendo y éste lo sumamos al minuendo. Si al realizar la suma no se produce acarreo que sobrepase la cantidad de bits de ambos entonces, el resultado de la suma es la diferencia de los números dados inicialmente. Si se produce un acarreo que sobrepase la cantidad de bits de la suma, éste se desprecia, y lo que quede es el resultado de la diferencia planteada.

<u>Ejemplo 20:</u>

Efectúe la resta 1011011 – 1010

Siguiendo los pasos dados:

1ro: Completo la cantidad de bits del sustraendo de forma que sea igual a la cantidad de bits del minuendo, el sustraendo quedará como: 0001010

2do: Hallo el dos complemento del sustraendo:

El uno complemento de 0001010 = 1110101
El dos complemento de 0001010 = 1110101 + 1

= 1110110

3er: Sumo el dos complemento del sustraendo con el minuendo:

```
  1011011
+ 1110110
---------
1 1010001
```

se desprecia

Entonces:

1011011 – 1010 = 1010001

Una forma de verificar lo anterior es efectuando la resta directamente; si los dos resultados son iguales, queda demostrado que el procedimiento anterior es correcto.

Efectuamos directamente:

$$\begin{array}{r} 1011011 \\ -\quad 1010 \\ \hline 1010001 \end{array}$$

Si comparamos este resultado con el obtenido anteriormente, vemos que son iguales, por lo que podemos concluir que el procedimiento anterior es correcto.

Ejemplo 21:

Restar 100110 – 1110, por el método del dos complemento:

1ro: Completamos el sustraendo con la cantidad de dígitos del minuendo: 001110.

2do: Hallamos el dos complemento de este:

Uno complemento de 001110 = 110001

Dos complemento de 001110 = 110001 + 1
= 110010

3ro: Efectuamos la suma del minuendo más el dos complemento del sustraendo:

$$\begin{array}{r} 100110 \\ +\ 110010 \\ \hline 1011000 \end{array}$$

se desprecia

Por tanto el resultado es: 011000.

Sin embargo, los números utilizados anteriormente (dos complemento) no incluyen el signo y esto hace que se produzca una ambigüedad entre los números positivos más grandes y los números negativos, de forma tal que es necesario que además de expresar las cantidades en su forma del dos complemento, se incluya el signo para evitar las ambigüedades en el momento de expresar cantidades positivas y negativas. Para ello designaremos el bit más significativo como bit signo, siendo cero para los positivos y uno para los negativos.

Este bit en las operaciones aritméticas es tratado de la misma forma que el resto de los bits; sin embargo la magnitud de los números expresados de esta forma (dos complemento con signo) se reduce, pues es necesario dejar el dígito más significativo para el signo.

1.6 MULTIPLICACIÓN BINARIA:

Las reglas para la multiplicación binaria, son verdaderamente simples como se puede observar en la tabla 1.3, en la cual se ve que el producto es uno, cuando los dos dígitos multiplicados son iguales a uno, de lo contrario el producto es cero.

Tabla 1.3: Producto binario de A y B

A	B	Producto (A • B)
0	0	0 • 0 = 0
0	1	0 • 1 = 0
1	0	1 • 0 = 0
1	1	1 • 1 = 1

Para ilustrar lo planteado, vemos un ejemplo.

Ejemplo 22:

Halle el producto de las cantidades binarias 100110 y 1110

```
    100110
      1110
    ------
    000000
   100110
  100110
 100110
----------
1000010100
```

Como se puede observar en este ejemplo, el proceso de multiplicación incluye la adición de un número de productos parciales, cada uno de estos productos puede ser igual al multiplicando o igual a cero.

Inicialmente la adición puede parecer algo complicada por la generación de múltiples acarreos, pero en todo caso un poco de práctica salvará esta dificultad.

En el caso de multiplicar dos números dados en la forma dos complemento con signo, la magnitud se halla como procedimos en el ejemplo anterior y el signo usando la siguiente regla:

“Si los signos de las cantidades multiplicadas son iguales, el producto es positivo; pero si las cantidades son de signo contrario el producto es negativo”.

Esta regla también puede ser aplicada para los signos en el caso de la división.

1.7 DIVISIÓN BINARIA:

El proceso de división binaria, es esencialmente un proceso de sustracción repetida. Las reglas para la división son mostradas en la tabla 1.4 en la cual la división por cero es indefinida, tal como ocurre en la división decimal.

Tabla 1.4: División binaria de A y B

A	B	Cociente (A : B)
0	0	0 : 0 = indefinido
0	1	0 : 1 = 0
1	0	1 : 0 = indefinido
1	1	1 : 1 = 1

Realicemos el siguiente ejemplo para ilustrar como se procedería en un caso práctico.

Ejemplo 23:

Hallemos el resultado obtenido de dividir 100110 por 1110

```
100110     | 1110
 1110      |--------
-----------  10.1011
0010100
   1110
-----------
0011000
   1110
-----------
 010100
   1110
-----------
  00110
```

RESUMEN:

En este capítulo hemos hecho un estudio de los sistemas binario, octal y hexadecimal, así como conversiones entre estos sistemas y entre estos sistemas y el decimal, enfatizando en los más utilizados en la práctica. Los sistemas octal y hexadecimal son importantes fundamentalmente por su utilización en gran cantidad de computadoras, minicomputadoras y microcomputadoras.

Todos los sistemas digitales operan con información que sólo puede tener dos estados; por tal razón el sistema binario es el adecuado para la representación de la información por todos los circuitos y sistemas digitales.

El uso del dos complemento para restar números binarios, es importante ya que la mayoría de las mini y microcomputadoras utilizan esta forma de restar.

La multiplicación y división binaria, se vieron como forma de ilustrar la forma de realizar dichas operaciones en el sistema binario.

EJERCICIOS PROPUESTOS:

I- Efectúe las conversiones correspondientes entre los sistemas según la base indicada:

	Base		Base
a) 45_{10}	2	b) 17_{8}	10
c) 49_{10}	2	d) 110101_{2}	10
e) 167_{16}	10	f) 575_{10}	16
g) 1101101_{2}	10	h) 975_{8}	10
i) 378_{10}	2	j) 132_{10}	2
k) 1101011_{2}	10	l) 274_{10}	8
m) 54_{10}	16		

II- Una vez hecha la conversión a la base indicada en el ejercicio anterior, vuelva a efectuar la conversión del número encontrado a su base original.

III- Efectúe las siguientes conversiones a los sistemas indicados:

a) 11010101_{2} base 10

b) 376_{10} base 2

c) 376_{8} base 2

d) 376_{16} base 2

e) 1110011101_{2} base 8

f) 1110011101_2 base 16

g) $3\ A\ 4_{16}$ base 2

h) 576_{10} base 2

i) 10101111_2 base 8

j) 10101111_2 base 10

k) 10101111_2 base 16

l) 576_{10} base 8

m) 576_8 base 10

n) 576_{10} base 16

o) 378_{16} base 10

IV- Haga la conversión en sentido contrario, en el ejercicio anterior, para comprobar que no cometió errores en él.

V- En cada uno de los siguientes ejercicios:

i) Efectúe la suma binaria.

ii) Halle la equivalencia de cada operación en el sistema decimal y efectúe la suma para comprobar el resultado según el inciso anterior.

a) 110000
+ 11010

b) 100100
+ 11100

c) 10010101
+ 0110111

d) 10100111
+ 1101011

e) 1001010111
+ 1101011

f) 10011100111
+ 1011111101

VI- Efectúe las operaciones de resta binaria indicadas:

a) 1001011
- 100101

b) 1101010
- 10110

c) 1001010
- 11001

d) 10010001
- 101111

e) 100010001
- 110110

VII- Realice las operaciones indicadas:

a) 110101
+ 11111

b) 10001111
+ 1111

VIII- Realice las operaciones siguientes por el método del dos complemento y verifíquelas por el método directo:

a) 110110101 - 110101
b) 11111010011 - 10101
c) 11101010111 - 110101
d) 10110111011 - 101101

CAPITULO No 2: ALGORITMO

Introducción

El siglo XX proporcionó a la humanidad muchos logros importantes en el campo de la ciencia y la técnica: la radio, el cine sonoro, la televisión, la energía atómica, los vuelos espaciales y los **ORDENADORES.** Seguramente no son menos conocidas la cibernética y la genética entre otros.

Pero no todos sabemos que uno de los alcances más importantes de la ciencia del siglo XX fue la teoría de los ALGORITMOS, que se convirtió en una disciplina matemática. La teoría de los ordenadores y la teoría y la práctica de la programación no puede prescindir de ella. No obstante, ella es una ciencia independiente que esta dispuesta a servir a todas las ciencias, pero posee su individualidad.

El propio nombre, teoría de los algoritmos, nos indica que su disciplina es el algoritmo. Pero ¿qué es un algoritmo?. El concepto de algoritmo es a la vez muy sencillo y muy complejo. Su sencillez está en la gran cantidad de procesos de la vida diaria, que tienen forma de algoritmos, con que tenemos que tratar y su calidad de habituales. Pero estas mismas circunstancias lo hacen impreciso, confuso, difícilmente sometido a una definición científica estricta.

La palabra **ALGORITMO** procede del nombre del matemático árabe (uzbeco) **Alheresmi** que en el siglo IX de nuestra era elaboró las reglas de las cuatro operaciones aritméticas sobre los números en el sistema decimal de numeración. El conjunto de estas reglas recibió en Europa el nombre de **Algoritmia.** Posteriormente esta palabra se transformó en **"ALGORITMO"** convirtiéndose en la **denominación colectiva de reglas aisladas de un tipo determinado** (y no solo de las reglas de operaciones aritméticas). Durante un largo tiempo la palabra **ALGORITMO** la emplearon solo los matemáticos, designando las reglas de solución de diversos problemas.

En los años 30 del siglo XX el concepto de algoritmo se hizo objeto del estudio matemático (antes solo se usaba) y al aparecer los ordenadores adquirió gran popularidad. El desarrollo de los medios electrónicos de cómputo y de los métodos de programación contribuyó a la comprensión de que la elaboración de los algoritmos es una etapa indispensable de la automatización.

En la actualidad la palabra algoritmo rebasa los márgenes matemáticos. Se comenzó a aplicar en las más diversas ramas, comprendiendo por ella una **regla** enunciada con exactitud, cuya destinación es ser guía para alcanzar el resultado necesario.

ALGORITMO (definición): Es un claro y exacto sistema de reglas que determinan una secuencia de acciones a realizar sobre determinado objeto, que después de un **número finito** de pasos nos permite lograr el objetivo de resolver la tarea planteada.

Como características fundamentales de un algoritmo deben tenerse en cuenta que tienen que ser**: finitos, ordenados y lógicos**.

El sistema de reglas se convierte en un algoritmo cuando puede darse en forma de instrucción a diferentes personas, desconocedoras de la esencia del problema y estas pueden, según las instrucciones, actuar de forma similar y alcanzar la misma solución.

2.1- Introducción a la técnica de algoritmos:

En esta unidad daremos tratamiento a ejercicios que nos permitirán acercarnos a las técnicas de la computación, ya que es muy importante la elaboración del algoritmo para dar solución a un problema determinado.

Para tener la posibilidad de automatizar un problema real cualquiera, aunque sea de la vida diaria, es necesario que se sigan los siguientes pasos:

- ► Hacer la interpretación correcta del problema, para encontrar la vía de solución, que muchas veces nos puede llevar a encontrar un problema matemático que nos describa el problema real.

- ► Escribir el algoritmo que permite dar solución al problema.

- ► Traducir el algoritmo a lenguaje de programación para que pueda ser interpretado por la computadora.

En esta unidad trabajaremos sólo las dos primeras etapas. La tercera etapa la trabajaras cuando aprendas a programar, aunque en esta unidad utilizaremos algunas palabras del lenguaje que nos permitirán identificarnos con el trabajo en la computadora como son: **impresión**, **localizaciones**, **corrida de algoritmos**, etc.

2.2 - Noción de algoritmo.

Ejemplo 1:

Hallar la imagen para x = 2 en la función lineal y = 3x + 4.

Teniendo en cuenta los conocimientos matemáticos adquiridos en grados anteriores, debes recordar que el proceso a seguir se puede describir según la siguiente sucesión de pasos, para obtener el resultado esperado:

(1) Sustituir el valor de x en la expresión.
(2) Multiplicar 2 por 3. El resultado es 6.

(3) Sumar el resultado con 4. El resultado es 10.
(4) Resultado: y = 10.

En el ejemplo anterior hemos realizado el proceso para evaluar una función lineal para un valor dado de x. Pero es conveniente generalizar el proceso para calcular la imagen para cualquiera sea una función de la forma $y = a \cdot x + b$, para cualquier valor de x, ya que al realizar el programa no lo debemos hacer para un caso particular, pues serian necesarios infinitos programas para cada uno de las ecuaciones de funciones lineales, por lo que podemos hacer un programa que determine las soluciones de todas las funciones lineales de la forma $y = a \cdot x + b$. En este caso utilizaremos un lenguaje de mayor precisión y con la simbología conveniente que nos acerque a los lenguajes de programación que posteriormente aprenderás.

Ejemplo 2:

Hacer la descripción del proceso para calcular la imagen por la función $y = a \cdot x + b$ de un número x dado.

Respuesta:

(1) Datos: a, b, x
(2) C **:=** a **·** x
(3) Y **:=** C + b
(4) Imprimir: “y =”; y.
(5) Fin.

En este ejemplo observemos que:

Cada paso de la descripción del proceso lo hemos numerado de forma consecutiva. A cada una de las acciones que se realizan en ellos les llamaremos **instrucciones**.

En el primer paso se tendrán como datos los valores de a y b para tener la función particular con que estamos trabajando, en el ejemplo 1, **a = 3** y **b = 4**, y además el valor de x que es al que le determinaremos su imagen por esta función, en el ejemplo 1, **x = 2**.

En el segundo paso estamos calculando el producto de a por x y lo estamos igualando a la variable C, que en este caso hace la función de una localización, donde vamos a guardar el resultado de la operación realizada. Esto al nivel de la memoria de la computadora significa que C es la dirección de memoria donde tendremos como contenido el resultado de la operación indicada en el miembro derecho de la expresión. Los dos puntos seguidos por el signo de igualdad significan que a C le estamos asignando el resultado de a **·** x, si en operaciones posteriores nos tenemos que referir al resultado de esta operación lo haremos utilizando C, ya que en esa localización está guardado el resultado.

En el tercer paso, como solo queda una operación utilizamos como localización a Y, que es a quien le queremos determinar el valor numérico.

En el paso (4) hemos usado la instrucción **Imprimir,** que será usada siempre que queramos indicar la salida de un resultado parcial o final; observa además que para cuando se imprima el resultado se sepa a cual nos referimos, hemos escrito entre comillas la variable que identifica este resultado, que además se usan siempre que se quiera imprimir un texto. Si en la instrucción **Imprimir** escribes una letra o un texto sin utilizar las comillas, se asume automáticamente que son variables y se exigirá el valor que se les asignara**.**

En el paso (6) usamos la instrucción **Fin** que nos indica que hemos llegado al final del proceso y, también hemos utilizado la variable C para nombrar un resultado parcial, esto hace que el proceso quede escrito en un lenguaje más preciso. A este proceso lo llamaremos en lo adelante **ALGORITMO**.

Debemos tener en cuenta que las variables que utilizamos en este algoritmo y en los sucesivos las llamaremos localizaciones, donde se asigna (se guarda) el valor que tendrá ya sea por entrada de datos o por asignación mediante una operación. Esto, visto en la programación, no es más que una dirección de memoria donde se guarda un valor para esa variable, que cada vez que la computadora la encuentra en la corrida del programa, va a esa dirección a buscar el contenido de esa localización, o sea, el valor guardado en ella, para usarlo en la resolución del problema.

Ejemplo 3:

Hagamos la descripción algorítmica para determinar la imagen de una función cuadrática: $\mathbf{y = a \cdot x^2 + b \cdot x + c}$ ($x \in R$.)

Vamos a detallar los pasos que debemos seguir para hallar la imagen:

1. Como datos tenemos los coeficientes a, b, c y, además el valor de x al que se le calculará la imagen.
2. Multiplicamos x por x. El resultado lo llamamos p_1.
3. Multiplicamos el resultado p_1 por a. Llamemos p_2 al resultado.
4. Multiplicamos b por x. Llamemos p_3 al resultado.
5. Sumamos p_2 y p_3. El resultado lo llamamos p_4.
6. Sumamos p_4 y c. El resultado es y.
7. Resultado y.
8. Fin.

Hagamos la descripción algorítmica de manera simbólica correspondiente a este problema:

(1) Datos: a, b, c, x

(2) $p_1 := x \cdot x$
(3) $p_2 := a \cdot p$
(4) $p_3 := b \cdot x$
(5) $p_4 := p_2 + p_3$
(6) $y := p_4 + c$
(7) Imprimir: “y =”; y
(8) Fin.

Debemos destacar que la descripción algorítmica de un problema no es, por lo general, única, sobre todo para las que dependen de una expresión matemática. Esto se debe a que puede estar en función de las operaciones que nos permita la computadora realizar en un solo paso.

Veamos el ejemplo anterior, aplicando transformaciones matemáticas permisibles a la expresión, suponiendo que la computadora nos brinda esas facilidades de trabajo.

Si extraemos factor común **x** en los términos de la expresión donde sea posible, obtendremos para la función la expresión:

$$\mathbf{y = [(a \cdot x) + b] \cdot x + c}$$

El algoritmo correspondiente a la nueva forma de la expresión sería el siguiente:

(1) Datos: a, b, c, x
(2) $p_1 := a \cdot x$
(3) $p_2 := p_1 + b$
(4) $p_3 := p_2 \cdot x$
(5) $y := p_3 + c$
(6) Imprimir: “y =”; y
(7) Fin.

Observa que esta nueva descomposición se realiza en menos pasos, por lo que se dice que es más eficiente que la anterior.

También, si por las facilidades que nos da la computadora, pudiéramos realizar de una sola vez la operación **(a · x) + b**, el algoritmo quedaría en la siguiente forma:

(1) Datos: a, b, c, x
(2) $p_1 := (a \cdot x) + b$
(3) $y := (p_1 . x) + c$
(4) Imprimir: “y =”; y
(5) Fin.

En este caso a la expresión **(a · x) + b** se le llama operador y en las ordenes de los ejercicios nos referiremos a este tipo de expresiones como **Op**, lo cual te indicará que si el algoritmo es lineal podrás hacer las operaciones indicadas en un solo paso, para lo cual no debes fijarte en que las variables usadas para la expresión del operador no sean las mismas que las de la expresión que estas describiendo, sino que debes fijarte que coincidan las formas de operación.

Observa que al aplicar el operador indicado en la expresión, la primera vez:

$$\mathbf{y = [(\underbrace{a \cdot x) + b] \cdot x}_{p_1} + c}$$

Luego la expresión quedaría ahora en la forma:

$$\mathbf{y = p_1 \cdot x + c}$$

En esta nueva expresión, observemos que tiene la misma forma del operador, en cuanto al orden de las operaciones, por lo que se puede también calcular en un solo paso.

En todos los ejemplos anteriores, hablaremos de **ALGORITMOS LINEALES,** ya que desde la **entrada de datos** hasta el **fin** se ejecutan secuencialmente todas las instrucciones, sin dejar de realizar ninguna, por lo que también se les puede denominar **ALGORITMOS SECUENCIALES.**

Es lógico pensar que, de las tres formas en que hemos trabajado con la misma expresión, aunque transformada, es más eficiente esta última, ya que la realizamos en menos pasos.

Pero recuerda que esto es sólo posible teniendo en cuenta que la computadora dé las facilidades de realizar, en un solo paso más de una operación o que tenga otras operaciones que no sean **suma, resta, multiplicación y división**. Es por ello que asumiremos que, siempre que resolvamos ejercicios donde el algoritmo sea lineal, solo realizaremos una operación por paso y en caso que se permita realizar más de una será indicado utilizando el llamado operador **(Op)**, donde se indique que operaciones se pueden realizar en un solo paso, teniendo en cuenta que debes respetar el orden de las operaciones que él te indica; observa en el ejemplo anterior que el operador sería **(a . x) + b** y se tuvo en cuenta la forma de la expresión y no las variables que la conforman y, fue posible usarla en los pasos (2) y (3).

El operador es también utilizado para operaciones como las raíces con cualquier índice, las potencias, los logaritmos, etc. En el caso de las potencias, si no aparece el operador, que te permita realizarlas en un solo paso, las realizarás por multiplicaciones sucesivas.

Ejemplo 4:

Haga la descripción algorítmica para calcular el valor numérico de M si:

$$\textbf{M} = \sqrt{x^2 - y} \quad \textbf{; Op:} \ \sqrt{a}$$

Respuesta:

Fíjate que el operador permite calcular la raíz cuadrada, pero no el cuadrado, por lo que lo calculamos como x · x. Además, en el operador aparece la variable **a** y en el algoritmo la variable usada es $\mathbf{a_2}$.

(1) Datos: x, y
(2) $a_1 := x \cdot x$
(3) $a_2 := a_1 - y$
(4) M := $\sqrt{a_2}$
(5) Imprimir: "M ="; M
(6) Fin.

Ejemplo 5:

Haga la descripción algorítmica para calcular el valor numérico de T si:

$$\textbf{T} = \sqrt{x^2 + y^2} \quad \textbf{; Op:} \ \sqrt{a + b}$$

Respuesta:

En este ejemplo debes estar claro que se resuelven primero $\mathbf{x^2}$ y $\mathbf{y^2}$, y luego calculamos la raíz cuadrada de la suma. Por tanto la descripción algorítmica seria:

(1) Datos: x, y
(2) $p_1 := x \cdot x$
(3) $p_2 := y \cdot y$
(4) T:= $\sqrt{p_1 + p_2}$
(5) Imprimir: "T ="; T

(6) Fin.

Ejemplo 6:

Haga la descripción algorítmica para determinar el área y el perímetro de un rectángulo, conociendo que las longitudes de sus lados son a y b.

Respuesta: Recordemos que:

1. El área del rectángulo se calcula como:

$$A = a \cdot b$$

2. El perímetro se calcula:

$$p = 2\,(a + b) \quad \text{ó} \quad p = 2 \cdot a + 2 \cdot b$$

Luego la descripción algorítmica sería:

(1) Datos: a, b
(2) A:= a · b
(3) x:= a + b
(4) p:= 2 · x
(5) Imprimir: “El área del rectángulo es =”; A; “cm^2 y el perímetro”; p;”cm.”
(6) Fin.

En este ejemplo observa que aunque es un ejemplo de calculo matemático, hemos dado la respuesta en la instrucción **Imprimir** con un texto que le permite saber al usuario del programa a que se refieren los resultados del mismo, teniendo en cuenta que el usuario no tiene por que conocer que problema resuelve el algoritmo o programa; él solo se limitara a introducir los datos que se le solicitan y esperar los resultados correspondientes.

3- Generalidad de un algoritmo.

Otra de las características de un algoritmo y una de las más importantes es que debe ser **General**, o sea, que un algoritmo no debe resolver un problema particular, como vimos en el **ejemplo 1**, sino que debe resolver un conjunto de problemas del mismo tipo.

El algoritmo realizado en el **ejemplo 2** nos permite calcular las imágenes para cualquier **x є R**, pero siendo **y = a · x + b** cualquier función también, en dependencia de los valores de **a** y **b**. Pero si quisiéramos generalizar el problema del cálculo de una imagen en una función potencial cualquiera, podemos tener en cuenta que las funciones lineales y las cuadráticas son casos particulares de las potenciales, por lo que si hacemos la descripción algorítmica para la expresión:

$$\mathbf{y = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_3 x^3 + a_2 x^2 + a_1 x + a_0}$$

donde en dependencia de que se hagan cero o no los coeficientes $\mathbf{a_n, a_{n-1}, \cdots, a_3, a_2, a_1, a_0}$, obtendremos los distintos tipos de funciones potenciales.

Este principio de la generalidad nos permite utilizar el algoritmo para resolver todos los problemas del mismo tipo.

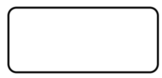
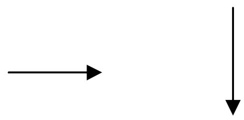
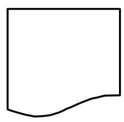
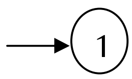
En la medida en que un algoritmo resuelva más problemas es más potente, por lo que debes tratar de hacerlo lo más general posible.

4- **Simbolismo para organigramas.**

En el análisis de la vía de solución de un problema, es siempre importante hacer la representación gráfica del problema para tener una idea mas clara de cómo obtener el resultado.

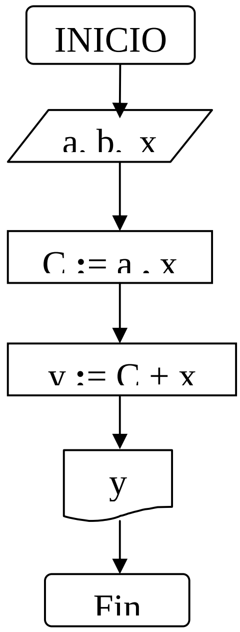
La representación gráfica de los algoritmos se denomina **Organigrama**, que es uno de los instrumentos más importantes para los algoritmos, ya que se puede visualizar la interrelación de los pasos que describen el algoritmo y facilita, por tanto el realizar la descripción algorítmica, restándole complejidad al trabajo.

En su confección usaremos un conjunto de símbolos que por su forma indican el tipo de operación descrita en su interior, unidos a las líneas y flechas que indican el flujo de las operaciones (por ello se le conoce también como diagrama de flujo).

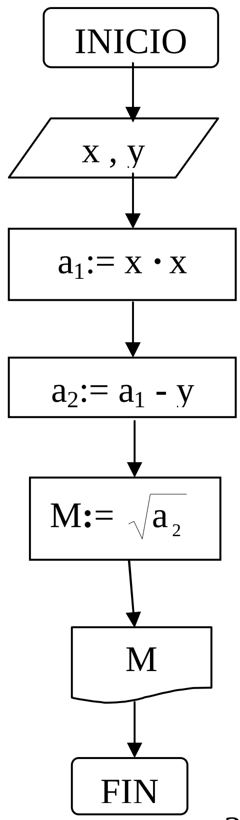
Se utiliza para	Símbolo
Indicar el **Inicio** y el **Fin** del algoritmo	
Indicar los **Datos** de entrada	
Indicar las **operaciones** a efectuar	
Indicar el **sentido de recorrido** del algoritmo	
Imprimir un resultado (parcial o final)	
	

Veamos los organigramas para los problemas de los ejemplos 2 y 4.

Ejemplo 7: (Organigrama del ejemplo 2)



Ejemplo 8: (Organigrama del ejemplo 4)



5- Rastreo de un algoritmo.

Es conveniente que, al confeccionar una descripción algorítmica, para resolver un determinado problema, seas capaz de verificar que funciona correctamente, o sea, que realiza todas las operaciones y se obtienen todos los resultados esperados. Esto es fundamental para que cuando lo traduzcas a un lenguaje de programación, sea un programa que no tenga la posibilidad de errores y además sea óptimo.

Para ello se realiza la operación de **RASTREO** que es cuando, utilizando un juego de datos, se hace la **corrida del algoritmo** realizando las instrucciones que en el aparecen de la misma forma en que lo haría la computadora, desde la entrada de los datos hasta obtener los resultados finales. Debemos tener siempre presente que esta operación se debe realizar con varios juegos de datos para poder comprobar que funciona correctamente con diferentes datos. Veamos un ejemplo, donde debes observar además que se va siguiendo el algoritmo paso a paso, lo que nos permitirá detectar si hay algún error en el proceso y en que paso se encuentra. Para trabajar de forma conveniente se elabora una tabla cuyo encabezamiento tiene las localizaciones y las instrucciones fundamentales del algoritmo, para mayor comodidad en la realización del rastreo.

Ejemplo 9:

Hagamos el rastreo del algoritmo del ejemplo 2.

	PASOS	a	b	x	C := a · x	y := C + b	IMPRESIÓN
1er Caso	(1) (2) (3) (4) (5)	3	2	2	6 = 3 · 2	8 = 6 + 2	8 Fin
2do Caso	(1) (2) (3) (4) (5)	3	2	- 1	- 3 = 3 · (- 1)	- 1 = - 3 + 2	- 1 Fin
3er Caso	(1) (2) (3) (4) (5)	- 2	4	3	- 6 = - 2 · 3	- 2 = - 6 + 4	- 2 Fin

En el ejemplo podemos ver que realizamos tres rastreos para juegos de datos diferentes. En los casos 1 y 2 debes ver que los valores de a y b no variaron, o sea, que estamos trabajando en ellos con la misma función lineal, **y = 3x +2**, para dos valores diferentes de x; ya en el tercer caso variamos también la función, o sea, con **y = - 2x + 4**.

Observa que en cada operación realizada se escribe el resultado a la izquierda y la operación a la derecha, eso es porque la localización donde se guarda el resultado aparece en el algoritmo en el miembro izquierdo de la expresión, por lo que siempre debes tener presente trabajar en esta forma en el rastreo.

Ejemplo 10:

Hagamos el rastreo del ejemplo 4.

	PASOS	x	y	$a_1 = x \cdot x$	$a_2 = a_1 - y$	$M = \sqrt{a_2}$	IMPRESION
1er Caso	(1) (2) (3) (4) (5) (6)	4	7	$16 = 4 \cdot 4$	$9 = 16 - 7$	$3 = \sqrt{9}$	3 Fin
2do Caso	(1) (2) (3) (4) (5) (6)	3	9	$9 = 3 \cdot 3$	$0 = 9 - 9$	$0 = \sqrt{0}$	0 Fin
3er Caso	(1) (2) (3) (4) (5) (6)	2	5	$4 = 2 \cdot 2$	$-1 = 4 - 5$	$\sqrt{-1}$	Error

En este ejemplo, en el 1er y 2do casos, hemos podido recorrer completamente el algoritmo, aún cuando en el 2do caso tuvimos que hallar la raíz cuadrada de cero (0); pero en el 3er caso como el valor al que le debemos hallar la raíz es negativo y, esta no está definida para los números reales negativos, la corrida del algoritmo se interrumpe y nos da error. Tendremos por tanto que tener en cuenta, al hacer la descomposición algorítmica, que estamos trabajando con raíz cuadrada y que tiene posibilidades de indefinición en el conjunto de los números reales.

Es por ello que estudiaremos en el epígrafe siguiente una nueva forma de descripción algorítmica donde podemos prever en el algoritmo cuando hay posibilidades de error en casos similares al anterior, evitando que nos de error y no ocurra una interrupción de la corrida.

Ejemplo 11:

Hagamos el rastreo del algoritmo del ejemplo 6:

	PASOS	a	b	A := x · x	x := a + b	P := 2 · x	IMPRESIÓN
1er Caso	(1)	5	3				
	(2)			15 = 5 · 3			
	(3)				8 = 5 + 3		
	(4)					16 = 2 · 8	
	(5)						El área del rectángulo es 15cm^2 y el perímetro 16cm.
	(6)						Fin
2do Caso	(1)	3	9				
	(2)			9 = 3 · 3			
	(3)				12 = 3 + 9		
	(4)					24 = 2 · 12	
	(5)						El área del rectángulo es 15cm^2 y el perímetro 16cm.
	(6)						Fin

Ejercicios Propuestos:

1. Haga la descripción algorítmica para calcular el valor numérico de la variable del miembro izquierdo en cada una de las expresiones siguientes:

a) $X = 5^a + 4$

b) $Y = 7k - h$

c) $Z = ab + c$

d) $K = 4x^2 - y$

e) $D = xy^2 + 3z$

f) $Q = \dfrac{3a - 8}{x}$

g) $R = \dfrac{3b + c}{a + 2}$

h) $S = \dfrac{4b}{a - 2}$

i) $P = \dfrac{4x^3}{x - 3}$

j) $T = \sqrt{x^2 - y^2}$; Op: $\sqrt{a}$

k) $A = \dfrac{\sqrt[3]{x^3 - y}}{y - 4}$; Op: $\sqrt[3]{a}$

l) $M = \dfrac{\sqrt[4]{c^2 - d}}{b - a}$; Op: $\sqrt[4]{x - y}$

2. Construya el organigrama correspondiente a cada uno de los algoritmos del ejercicio anterior.

3. Haga el rastreo de cada uno de los algoritmos del ejercicio 1 para los valores siguientes, teniendo en cuenta la coincidencia de los incisos correspondientes.

a) a = 4
5
a = 0
1
a = -3
0

b) k = 3 ; h = -2
k = 0; h = 4
k = -1; h = 0,5

c) a = 2; b = 0; c =
a = -4; b = 2; c = -
a = 3; b = -3; c =

d) x = 0 y = -2,5

e) x = 4; y = -2 z = -1

4. Hacer la descripción algorítmica de los problemas siguientes:

a) Determine los coeficientes en el polinomio producto $P \cdot Q$, donde:

$$P = a \cdot x + b \text{ y } Q = c \cdot x + d.$$

b) Determine el área y el perímetro de un paralelogramo de lados a y b.

c) Calcular el área total de una esfera cuyo volumen es x cm^3.

d) Calcular la longitud de la arista y el volumen de un cubo que tiene x cm^2 de área total.

e) Calcular la altura de un cilindro que tiene x dm^3 volumen, si el radio de la base es de *y* cm.

5. Dada la descripción algorítmica siguiente, diga:

(1) Datos: a, b, x
(2) $p = a \cdot x$
(3) $q = p \cdot x$
(4) $r = q \cdot x$
(5) $y = r + b$
(6) Imprimir: y
(7) Fin.

a) ¿Qué problema resuelve?
b) Si dispusiera de la operación elevar al cubo directamente. ¿Cómo haría la descripción algorítmica de este problema?

6. Analice si las descripciones algorítmicas siguientes son dos descomposiciones de un mismo algoritmo o no:

(1) Datos: b, x, y	(1) Datos: b, x, y
(2) $a_1 = b \cdot y$	(2) $a_1 = b \cdot y$
(3) $a_2 = a_1 + x$	(3) $a_2 = a_1 \cdot y$
(4) $a_3 = a_2 \cdot y$	(4) $a_3 = x \cdot y$
(5) Imprimir: a_3	(5) $a_4 = a_2 + a_3$
(6) Fin.	(6) Imprimir: a_4
	(7) Fin

a) ¿Qué problema resuelve?
b) Realice el organigrama para cada uno.
c) Haga el rastreo de cada algoritmo, con los mismos juegos de datos (como mínimo dos juegos), y verifique que los resultados son iguales.

6. Algoritmos con Alternativas.

Ejemplo 12:

Hacer la descripción algorítmica para calcular el valor numérico de T si:

$$\mathbf{T} = \frac{a^2}{b}$$

Si realizamos la descripción algorítmica utilizando un algoritmo lineal para resolver este problema obtendríamos:

(1) Datos: a, b
(2) X:= a · a
(3) T:= $\frac{x}{b}$
(4) Imprimir: " El valor de T es"; T
(5) Fin.

Hagamos el rastreo de este algoritmo para a = 3 y b = 0:

PASOS	a	b	x:= a · a	T := $\frac{x^2}{b}$	IMPRESIÓN
(1)	3	0			
(2)			9 = 3 · 3		
(3)				Indef. = $\frac{3}{0}$	
(4)					error
(5)					

Fíjate que como la división por cero no está definida, al dividir 3 por 0 se interrumpe la ejecución y nos dá error. Como vimos tambien en el ejemplo 4 del epígrafe anterior hay problemas en los que no podemos realizar la descripción utilizando un algoritmo lineal, ya que las respuestas que demos pueden depender de condiciones diferentes; para ello en la descripción algorítmica debemos prever esta situación, para que se envíe un mensaje desde él y no esperar que se interrumpa la ejecución y se envíe un mensaje de error.

Lo anterior nos hace ver la necesidad de introducir un nuevo tipo de algoritmo en el tengamos en cuenta las posibles respuestas del problema según los datos que procesemos o los resultados de operaciones intermedias en el algoritmo.

Estudiaremos, por tanto en este epígrafe, los **ALGORITMOS CON ALTERNATIVAS**, que son aquellos que permiten realizar la descripción algorítmica de problemas en los que podemos encontrarnos con distintas opciones de entrada de datos o de resultados de determinadas operaciones que tengamos que resolver para llegar a la solución del problema.

En el ejemplo 12 para evitar el error debemos tomar una decisión a partir del valor de **b**, ya que no puede tomar el valor de 0, Lugo debemos preguntarnos:

¿b ≥ 0?

Si — No

$T = \frac{a^2}{b}$ — la expresión se indefine

Es por ello necesario que introduzcamos la instrucción **Si … entonces … sino** que nos permitirá realizar la descripción algorítmica correspondiente. Esta instrucción permite que se tome la decisión de ejecutar las instrucciones correspondientes sólo para una de las decisiones, según la respuesta a la pregunta realizada.

En algunos lenguajes de programación esta instrucción es la que se conoce como **IF…THEN…ELSE**, y por su complejidad e importancia dentro del trabajo de programación, debemos prestarle mucha atención ya que también es la base del trabajo con otros problemas más complejos.

Sintaxis (Forma en que se utiliza en un algoritmo):

SI <Exp. Arit. > **ENTONCES** <Lista de Inst.> [**SINO** <Lista de Inst.>]

<Exp. Arit.> **: : =** Puede ser una expresión matemática o simplemente una comparación que represente a la condición planteada en el problema.

<Lista de Inst.> **: : =** Puede ser una instrucción o varias instrucciones una en cada paso. En el caso del SINO se escribe entre corchetes por que es opcional que se utilice según si el problema que estemos resolviendo lo necesita o no.

La expresión aritmética debe estar en correspondencia con la condición que tenga el problema para la decisión y se analiza si esta expresión se cumple o no. En ocasiones, si la condición no puede ser escrita como expresión aritmética, entonces se plantea la condición correspondiente y se ejecutan las instrucciones de cada bloque según se cumpla o no.

En las expresiones aritméticas debes tener en cuenta que puedes usar los operadores de relación:

>	Mayor	**=**	Igual
>=	Mayor o Igual	**<=**	Menor o igual
<	Menor	**< >** o **><**	Diferentes

Es importante señalar que debemos diferenciar los símbolos **:=** y = ya que el primero lo usaras para indicar la asignación de un valor a una variable y el segundo para realizar comparaciones entre valores.

Ejemplo 13:

Hacer la descripción algorítmica para dada la nota (Nt) de un alumno decir su condición en la asignatura analizada.

Respuesta:

Debes tener en cuenta, para resolver este problema, que un alumno está aprobado en una asignatura si tiene 60 puntos o más, o sea, si la nota (Nt) es mayor o igual que 60 está "APROBADO" y si es menor que 60 está "SUSPENSO".

Luego la pregunta a realizar está en la relación entre Nt, que es la nota del alumno que estamos analizando y la nota de aprobado que es 60 puntos. En dependencia de la respuesta a esta pregunta se dará como resultado la condición del alumno en la asignatura.

Por tanto la descripción algorítmica en este caso sería:

(1) Datos: Nt. // se entra la nota del alumno.
(2) Si Nt >= 60 entonces
(3) Imprimir: "APROBADO"
(4) sino Imprimir: "SUSPENSO"
(5) Fin.

En este ejemplo al comparar la nota Nt del alumno con 60, si se cumple que es mayor o igual entonces se imprime "APROBADO" y automáticamente iría al Fin, ya que es la única instrucción correspondiente a cuando se cumple la condición, de lo contrario se ejecuta la instrucción que está en el SINO, es decir, se imprime "SUSPENSO", y como es lógico se iría al Fin en este caso por que es la instrucción que le sigue.

Para lo anterior debes tener en cuenta que la instrucción **Si ... entonces ... sino ...**, aunque contenga, en cada bloque de instrucciones para cada alternativa varias instrucciones, desde el **Si** hasta el final de la última instrucción que esté dentro del bloque del **sino**, constituyen una sola instrucción. Esto debes tenerlo en cuenta siempre para poder saber cual es la instrucción que se ejecutaría a continuación de la instrucción **Si ... entonces ... sino ...**

Ejemplo 14:

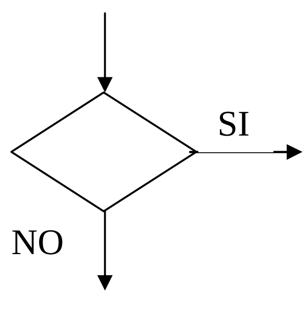
Analicemos el siguiente esquema de un algoritmo cualquiera:

(1) **...**
(2) **...**
(3) Si <exp.> entonces
(4) Instrucción 1
(5) Instrucción 2
(6) sino Instrucción 4
(7) Instrucción 5
(8) Instrucción 6
(9) Instrucción 7
(10) Fin.

Al ejecutarse este algoritmo se realizarían los pasos **(1), (2)** y **(3),** que para cualquier juego de datos son comunes y obligatorios de realizar, Cuando se ejecute el paso (3) se evalúa la expresión **<exp.>**, si es **verdadera**, o sea, si se cumple la expresión para el juego de datos con que se está trabajando, entonces se ejecutarían las instrucciones de los pasos **(4)** y **(5)** que son las que pertenecen al bloque del **entonces** y a continuación se ejecutarían las instrucciones de los pasos **(8), (9)** y **(10)**. Los pasos **(6)** y **(7)** no se ejecutan en este caso porque pertenecen al bloque de instrucciones del **sino**. Por el contrario si la expresión fuera **falsa**, o sea, si no se cumpliera, después del paso **(3)** se ejecutarían las instrucciones de los pasos **(6)** y **(7)**, seguidos de las de los pasos **(8), (9)** y **(10)**.

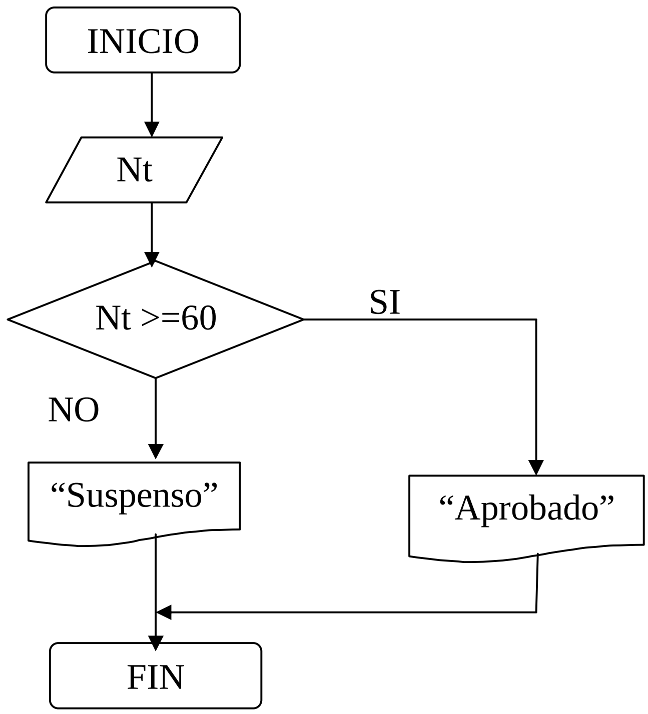
O sea, cuando se utiliza una instrucción de alternativa tanto para el **entonces** como para el **sino** se ejecutará el bloque de instrucciones correspondiente teniendo en cuenta si se cumple (bloque del **entonces**) o si no se cumple (bloque del **sino**) la condición del problema. Decimos en este caso que hay un **Salto condicionado**, ya que el control de ejecución salta al bloque correspondiente a la respuesta a la pregunta de la condición del problema.

Para construir el organigrama correspondiente introduciremos un nuevo símbolo:



Este símbolo se utiliza cuando se debe tomar una decisión lógica, por lo que tendrá una entrada y dos salidas, que al responder a la pregunta planteada dentro de este símbolo puede tener dos respuestas **SI** o **NO**, en dependencia de la cual se ejecutarán las instrucciones correspondientes.

Por tanto para problema del ejemplo 13, el organigrama quedaría en la forma siguiente:



Veamos el rastreo correspondiente a este algoritmo:

PASOS	Nt	Nt >= 60	IMPRESIÓN
(1) (2) (3) (5)	93	 SI	 Aprobado Fin
(1) (2) (3) (5)	60	 SI	 Aprobado Fin
(1) (2) (4) (5)	52	 NO	 Suspenso Fin

Observa que en este ejemplo hemos corrido 3 veces el algoritmo, utilizando diferentes datos, para comprobar que corre para todos los casos posibles según la condición planteada en el problema.

En este tipo de problemas es necesario para que no conlleve a errores, como ocurrió en los ejemplos 10 y 12, que se analice en la búsqueda de la vía de solución, todos los casos de posibles de respuestas a la pregunta formulada a partir de la condición planteada en el problema.

En el ejemplo 12 procederíamos de la siguiente forma:

(1) Datos: a, b
(2) Si b <> 0 entonces
(2) T := $\frac{a^2}{b}$
(3) Imprimir: " El valor numérico de T es"; T
(4) sino Imprimir: " La expresión se indefine"
(5) Fin.

Debes fijarte que en este caso no se orientó el uso de ningún operador y sin embargo trabajamos la expresión completa en un solo paso, te avisamos que a partir de ahora, que ya dominas como trabajar en un algoritmo el orden de las operaciones, se te está permitido que en un solo paso puedas realizar varias operaciones, siempre y cuando no ocurra una indefinición o halla mas de una opción o posibilidad de trabajo

Veamos como quedaría el algoritmo del cálculo del valor de M descrito en el ejemplo 4.

Ejemplo 15:

Hagamos la descripción algorítmica para calcular el valor de M. Considere las indefiniciones.

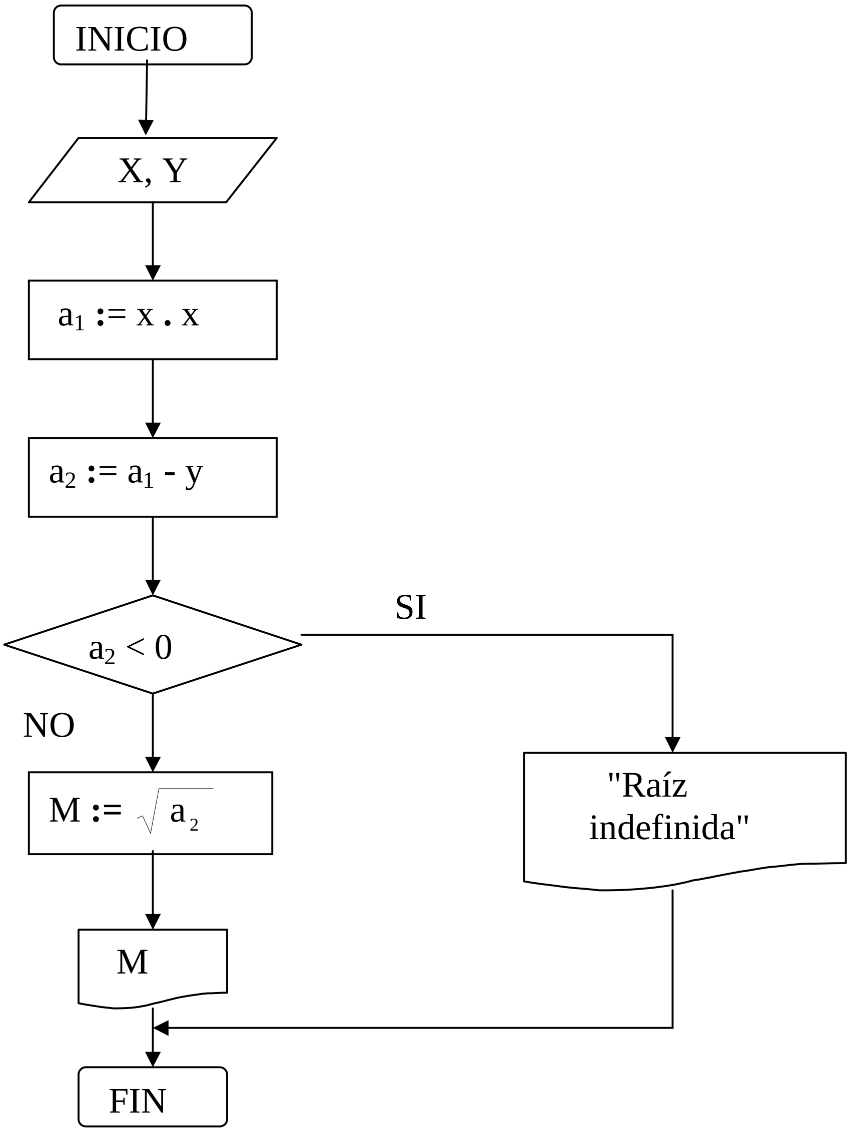
$$M = \sqrt{x^2 - y}$$

Respuesta:

De forma conveniente hagamos primero el organigrama correspondiente como análisis de la vía de solución. Esto lo puedes hacer en ese orden en lo adelante, ya que el confeccionar primero el organigrama te dará una idea de cómo escribir la descripción algorítmica para resolver el problema.

Teniendo en cuenta el orden de las operaciones calcularíamos primero el cuadrado de **x** y al resultado le restaríamos el valor de **y**. Debemos entonces ver que ocurre con este resultado para poder continuar, ya que debes recordar que la raíz cuadrada no se le puede calcular a todos los números reales, o sea, aquí debemos realizar la pregunta correspondiente a las alternativas que se pueden dar según el resultado del calculo anterior, la cual sería si el valor obtenido es positivo o cero (mayor o igual que cero) o negativo (menor que cero)

Luego el organigrama sería:



Fíjate que luego de haber calculado $\mathbf{a_2}$ se debe comparar con cero para tomar la decisión, en este caso si es negativo ($\mathbf{a_2 < 0}$) no se puede calcular la raíz y por tanto se imprime **“raíz indefinida”**, de lo contrario se calcula la raíz y se imprime su valor como resultado del algoritmo. En esta forma no debe ocurrir lo que antes de conocer este tipo de algoritmos, o sea nunca nos dará error en la ejecución por no poder calcular, ya que siempre tendrá una respuesta que darnos.

Por tanto el algoritmo nos quedaría de la siguiente forma:

(1) Datos: x, y.
(2) $a_1 := x \cdot x$
(3) $a_2 := a_1 - y$
(4) Si $a_2 >= 0$ entonces
(5) $\qquad M := \sqrt{a_2}$
(6) $\qquad$ Imprimir: M.
(7) sino $\quad$ Imprimir: " raíz indefinida".
(8) Fin.

Hagamos el rastreo correspondiente a esta nueva descomposición.

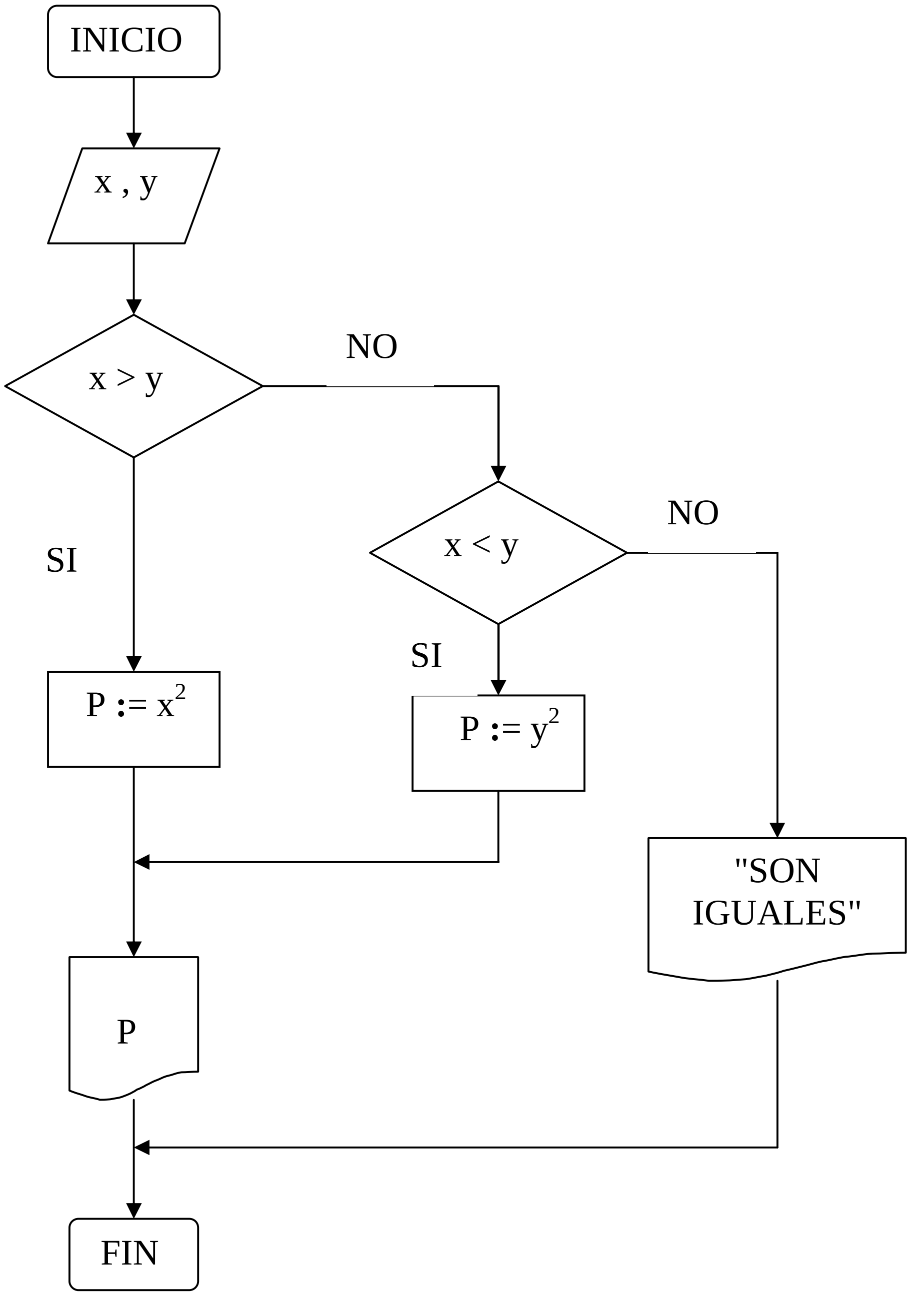
	PASOS	x	y	$a_1 = x \cdot x$	$a_2 = a_1 - y$	? $a_2 \geq 0$	$M = \sqrt{a_2}$	IMPRESION
1er CASO	(1) (2) (3) (4) (5) (6) (8)	4	7	$16 = 4 \cdot 4$	$9 = 16 - 7$	SI	$3 = \sqrt{9}$	3 Fin
2do CASO	(1) (2) (3) (4) (5) (6) (8)	3	9	$9 = 3 \cdot 3$	$0 = 9 - 9$	SI	$0 = \sqrt{0}$	0 Fin
3er CASO	(1) (2) (3) (4) (7) (8)	2	5	$4 = 2 \cdot 2$	$-1 = 4 - 5$	NO		Raíz indefinida Fin

Ejemplo 16:

Hacer la descripción algorítmica para:“Dados dos números X e Y se imprima el cuadrado del mayor, y si son iguales imprima un texto adecuado.”

Respuesta:

El organigrama correspondiente sería:



Luego el algoritmo correspondiente sería:

(1) Datos: X, Y
(2) Si X > Y entonces
(3) $P := X^2$
(4) Imprimir: P.
(5) sino Si X < Y entonces
(6) $P := Y^2$
(7) Imprimir: P
(8) sino Imprimir: " Son iguales"
(9) Fin.

En este ejemplo puedes observar que se usó la instrucción de alternativa dos veces esto ocurre porque hay tres posibilidades en la comparación entre el valor de X y el valor de Y (X > Y, X = Y o X < Y), luego asumiremos siempre que esta instrucción se utilizará una vez menos que la cantidad de posibilidades de casos que tenga la condición planteada en el problema. Pero ten en cuenta que las preguntas que hagas y el orden en que las hagas no es fijo, o sea puede ser como desees o a veces como sea más factible para obtener los resultados convenientes.

Además, debemos aclarar que para guardar el valor de X^2 y el de Y^2 , utilizamos la misma localización P, esto es posible porque como sólo se ejecuta una de las posibilidades de la alternativa entonces la localización sólo tendrá un valor cada vez, el correspondiente al calculo según la parte ejecutada. Es por ello que este mismo algoritmo es posible escribirlo en la siguiente forma:

(1) Datos: X, Y
(2) Si X = Y entonces
(3) Imprimir: "son iguales".
(4) sino Si X > Y entonces
(5) $P = X^2$
(6) sino $P = Y^2$
(7) Imprimir: P
(8) Fin.

En esta forma observemos que se determinan los valores de los cuadrados y según cual se haya calculado es el contenido de la localización P y sería el que se imprimiría al ejecutarse la instrucción del paso (7).

Ejercicios Propuestos.

I. Haga la descripción algorítmica para determinar los valores numéricos de las expresiones del ejercicio 1, del epígrafe anterior incisos del f al l, considerando en ellos las indefiniciones si todas las variables están definidas en los reales.

II. Haga la descripción algorítmica para resolver los siguientes problemas:

1. - Dado un número si es posible calcule su raíz cuadrada, en caso contrario imprima un texto.

2. - Dados dos números hallar el cuadrado del mayor y el cubo del menor, de ser iguales imprimir un texto adecuado.

3.- Dados dos números A y B, si su suma es positiva calcular su raíz cuadrada y si es negativa halle su opuesto.

4. - Dados tres números determinar si pueden ser los valores de los ángulos interiores de un mismo triángulo.

5. - Dados dos números A y B, calcule el cubo del mayor y de ser iguales calcule:

$$y = A \cdot B - A - B$$

6. - Dados cuatro números multiplicar el primero por el cuarto, sumar los intermedios e imprimir solamente el mayor de los resultados.

7. - Dado un punto (x; y) determine si pertenece o no a la recta $y = ax + b$.

8. - Dada la concentración de una disolución determinar si es base, ácido o neutra. Recuerde que:

$$pH = - \lg C$$

- Si $pH > 7$ es base
- Si $pH < 7$ ácida
- Si $pH = 7$ neutra.

9. – Evaluar la función:

$$R = \begin{cases} x^2 & \text{pa } x > 5 \\ \\ 2x + 1 & \text{para } x \le 5 \end{cases}$$

10. - Determinar el cumplimiento del plan de producción de una embotelladora, que es de 1 000 cajas de refresco, si se trabaja a un ritmo de CH cajas por hora y se trabajó H horas.

11. – Calcule los valores de X e Y, y siga las indicaciones:

$X = A + B – C^2$ $Y = A^2 \cdot B – B \cdot D$

Si Y = X imprimir su producto.

Si Y > X imprimir el módulo de su diferencia

Si Y < X imprimir su promedio.

12. - Conociendo, de una casa su número (NC) y su gasto eléctrico (GE) en el mes, calcular cuanto debe pagar (P) , si conoces además que:

- Si GE ≤ 100 kWh se calcula como P = GE • 0,09
- Si 100 kWh ≤ GE ≤ 300 se calcula como P = GE • 0,20 – 11
- Si GE > 300 se calcula como P = GE • 0,30 – 41

a) Si el gasto eléctrico es mayor que 300kWh, envíele un mensaje adecuado referido al trabajo que se realiza en el país con el PAE.

7. ALGORITMO CON LAZO O CICLO.

Existen problemas de la vida cotidiana que conllevan a repetir una o más acciones para lograr los resultados necesarios.

Por ejemplo, cuando se quiere calcular un promedio de notas, sumas de varios valores, conteos y de manera general cuando se quiere procesar listas de datos. En estos casos utilizaremos los **ALGORITMOS CON CICLOS O LAZOS**.

Al trabajar con listas de datos se pueden utilizar dos tipos de ciclos: **Determinados** e **Indeterminados**.

► **Ciclos Determinados**

Son aquellos en que se conoce la cantidad de juegos de datos que se van a procesar, o sea, cuando se conoce la cantidad de veces que se repetirá el ciclo.

Para este tipo de ciclos utilizaremos la instrucción:

Sintaxis:

Desde <var. = valor inic.> **hasta** < valor final> **hacer** <lista de inst.>

< var. = valor inic. > **::=** valor inicial, o sea, valor que se asigna en el orden al primer elemento de la lista, este valor se guarda en la localización indicada por la variable **var** que se use, la cual recibe el nombre de **contador**, que se encarga de contar las veces que se ha ejecutado el ciclo.

< valor final> **::=** valor que indica la cantidad de veces que se debe repetir el ciclo.

< lista de inst.> **::=** lista de instrucciones que se repiten.

Esta instrucción es similar a la instrucción **For … to … do** que aparece en algunos lenguajes de programación. Es posible usarla en dos formas diferentes:

1era Forma: Si el número de repeticiones es fijo:

Desde **i** := 1 hasta 10 hacer

Lista de instrucciones
que se repiten

Próximo **i**

En este caso el valor final nos indica que se harán 10 repeticiones del ciclo solamente, y por tanto cuando se hagan se pasara automáticamente a la instrucción que esta a continuación del ciclo.

2da Forma: Si el número de repeticiones es variable:

Datos: N
Desde **i :=** 1 hasta N hacer

Lista de instrucciones
que se repiten

Próximo **i**

En esta forma se realizaran N repeticiones del ciclo y luego se ejecuta la siguiente instrucción.

Te llamaremos la atención acerca de que aunque decimos que N es variable, fijate que ese valor se pide como dato antes de entrar en el ciclo, por lo que una vez que entre ya se le asigno un valor con el cual se controlará el número de repeticiones del ciclo, por ello se dice que es un ciclo determinado.

En ambos casos el cierre del ciclo se indica con la instrucción **Próximo i**, que en este caso hace que el valor de i (contador **general** en el ciclo) se incremente cada vez en uno y que se repita el ciclo si no se cumple que **i > N**. Cada vez que se llega a esta instrucción se produce un salto automático hacia la instrucción donde comienza el ciclo, para chequear si ya se produjeron las N repeticiones y salir del ciclo. Cuando esto ocurra, se produce un salto automático a la próxima instrucción después del ciclo.

Ejemplo 17:

Observemos como funciona la ejecución de un algoritmo con ciclo en el siguiente algoritmo:

(1) Datos: N
(2) Desde **i :=** 1 hasta N hacer
(3) Instrucción 1
(4) Instrucción 2
(5) Instrucción 3
(6) Próximo **i**
(7) Instrucción 4
(8) Instrucción 5
(9) Fin.

En este algoritmo se ejecutaran la primera vez las instrucciones de los pasos (1) y (2), si se cumple que **i ≤ N** se entra al ciclo y se ejecutan las instrucciones de los pasos (3), (4), (5) y (6), de ahí saltaría a ejecutar nuevamente el paso (2) donde se verifica nuevamente si **i ≤ N**, de cumplirse se volverán a repetir todas las instrucciones de los pasos del (3) al (6) hasta que se cumpla que **i > N** y en ese caso se produce un salto al paso (7), que es donde está la primera instrucción después del ciclo y continua con las de los pasos (8) y (9).

Ejemplo 18:

Haga la descripción algorítmica para dadas las notas (Nt) de Física de los 40 alumnos de nombre (Nb) de un grupo, indique la condición de cada uno en la asignatura.

Respuesta:

Es necesario que tengas en cuenta que en este algoritmo se repiten 40 veces las instrucciones del algoritmo del **ejemplo 13** del epígrafe anterior, en el ciclo ya que se procesaran las notas de todos los alumnos del grupo.

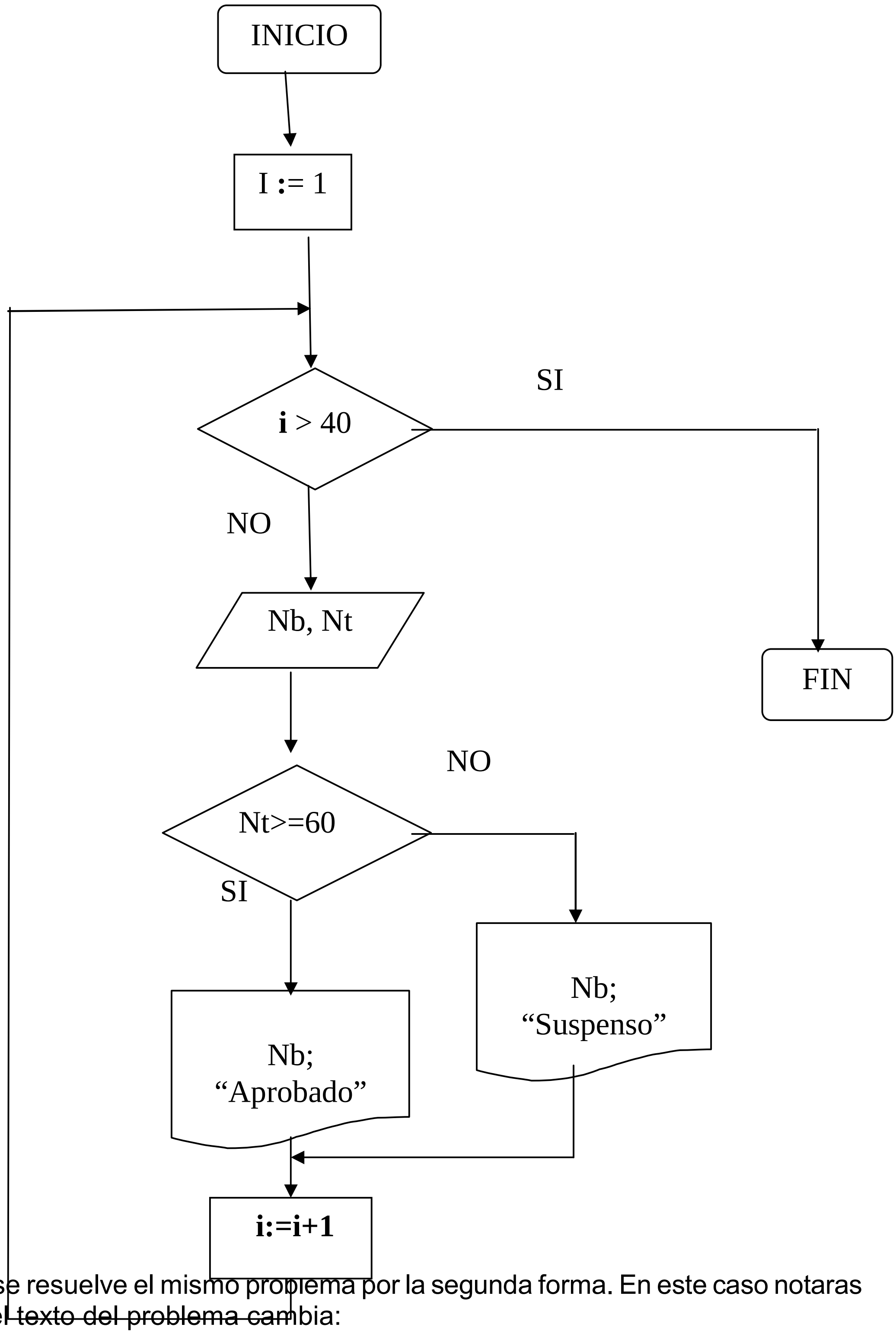
Luego el algoritmo quedaría como sigue:

(1) Desde **i** := 1 hasta 40 hacer

(2) Datos: Nb, Nt.
(3) Si Nt >= 60 entonces
(4) Imprimir: Nb; " aprobado "
(5) sino Imprimir: Nb; " suspenso "
(6) Próximo **i**
(7) Fin.

Leyenda:
i: contador de los alumnos que se van procesando.
Nb: nombre del alumno
Nt: nota del alumno

Veamos el organigrama correspondiente:



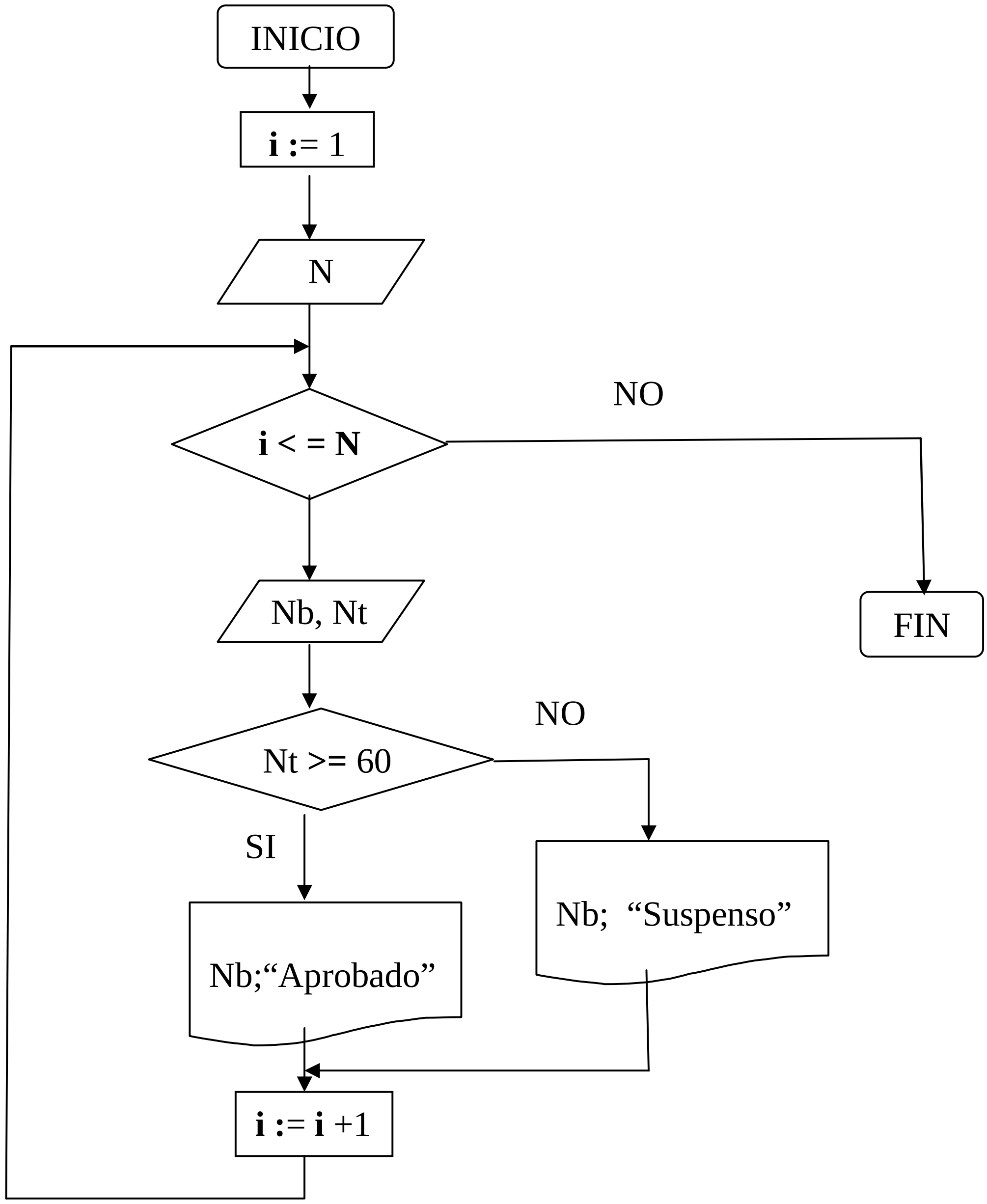
Veamos como se resuelve el mismo problema por la segunda forma. En este caso notaras que la forma del texto del problema cambia:

Ejemplo 19:

Haga la descripción algorítmica para dadas las notas (Nt) de los N alumnos de nombre (Nb) de un grupo se obtenga el listado con la condición de cada uno en la asignatura.

Respuesta:

Debes ver que en este caso es necesario trabajar con N notas, o sea, emplearemos necesariamente un ciclo donde se conoce el número de veces que se repiten las instrucciones, aunque esta cantidad es variable.



El algoritmo quedaría por tanto en la siguiente forma:

(1) Datos: N.
(2) Desde **i** := 1 hasta N hacer
(3) Datos: Nb, Nt
(4) Si Nt >= 60 entonces
(5) Imprimir: Nb; “aprobado”
(6) sino Imprimir: Nb; “suspenso”
(7) Próximo **i**
(8) Fin.

<u>Leyenda</u>:
N: cantidad de alumnos del grupo
i: controla la cantidad de alumno que se va analizando.
Nt: nota del alumno.
Nb: nombre del alumno.

Esta segunda forma tiene la ventaja de que podemos procesar grupos que tengan diferentes matriculas, ya que a N se le puede asignar como valor cualquier número y por tanto esta forma de trabajo es más general y útil; en la primera forma sólo se podrán procesar grupos cuya matrícula sea 40 alumnos.

Ejemplo 20:

Haga la descripción algorítmica para dadas las notas (Nt) de los N alumnos de nombre (Nb) de un grupo, se obtenga:

a) **La condición de cada uno en la asignatura analizada.**

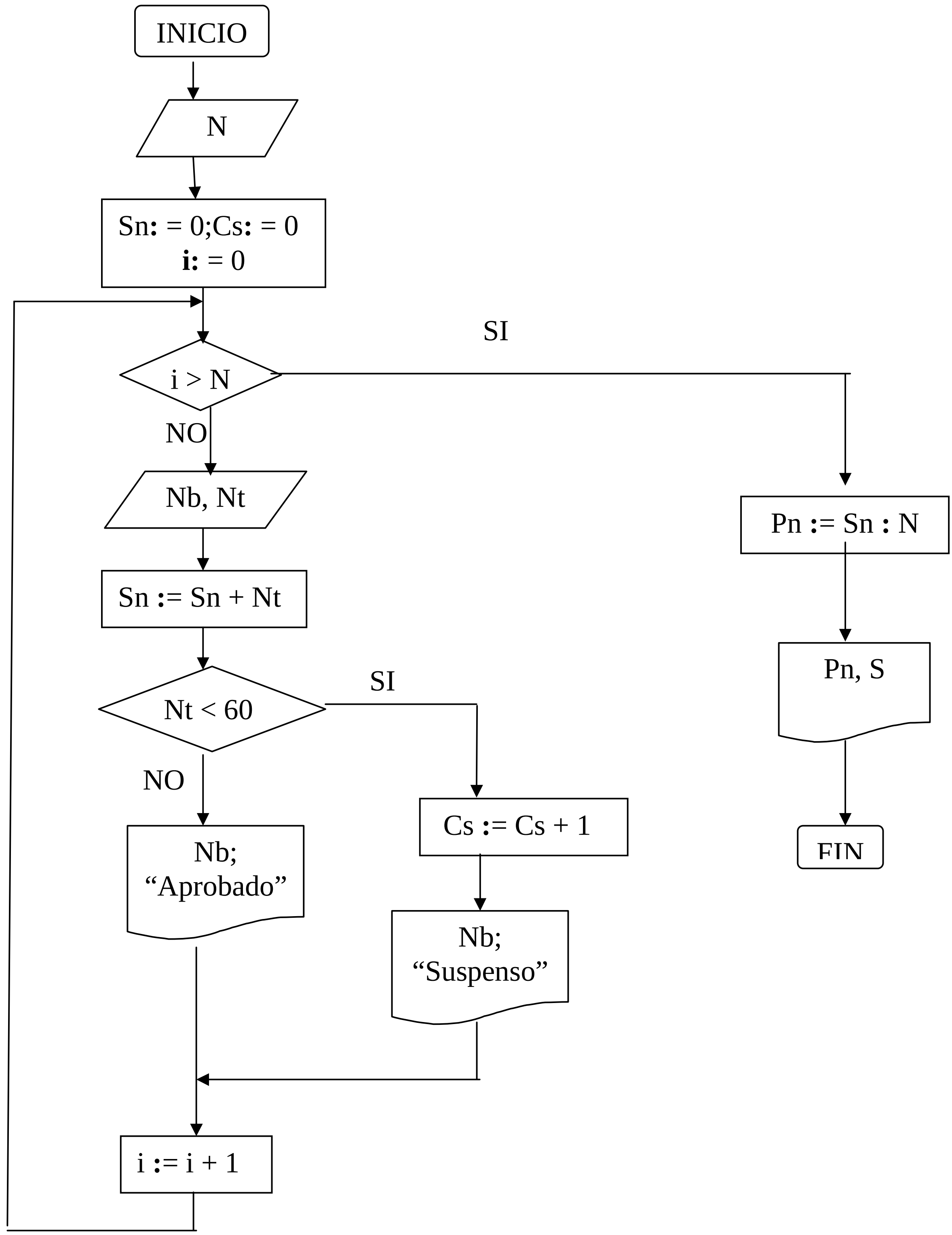
b) **La cantidad de alumnos suspensos.**

c) **El promedio de notas del grupo en la asignatura.**

Respuesta:

Debes fijarte que en este ejemplo se tiene que dar como resultado el promedio de notas del grupo, por lo tanto debes recordar que se calcula el promedio sumando las notas de todos los alumnos y dividiendo esa suma entre la cantidad de alumnos del grupo, además se debe dar la cantidad de alumnos que suspendieron la asignatura por lo que se deben ir contando los alumnos que cumplen esa condición.

Veras en este ejemplo la necesidad que tenemos de usar nuevas herramientas para poder llegar a las respuestas que debemos dar como conclusión de la resolución del problema. Luego el organigrama sería:


INICIO
N
Sn: = 0;Cs: = 0
i: = 0
SI
i > N
NO
Nb, Nt
Pn := Sn : N
Sn := Sn + Nt
Pn, S
SI
Nt < 60
NO
Cs := Cs + 1
FIN
Nb;
“Aprobado”
Nb;
“Suspenso”
i := i + 1

Luego el algoritmo sería:

```
(1)  Datos: N.

(2)  Sn := 0;  Cs := 0

(3)  Desde  i := 1  hasta N hacer
(4)           Datos: Nb, Nt
(5)           Sn := Sn + Nt
(6)           Si Nt >= 60 entonces
(7)                     Imprimir: Nb; " aprobado"
(8)           sino     Imprimir: Nb; " suspenso"
(9)                     Cs := Cs + 1
(10)          Próximo i
(11)  Pn := Sn: N
(12)  Imprimir: "Promedio de notas del grupo"; Pn
(13) Fin
```

Leyenda:

N: cantidad de alumnos del grupo.
Pn: promedio de notas del grupo.
Sn: Suma de notas.
Cs: Cantidad de suspensos.
Nt: Nota del alumno.
Nb: nombre del alumno.

Como puedes observar en el paso (5) hemos usado una nueva instrucción: **Sn = Sn + Nt**, donde se fue guardando la suma de las notas de cada uno de los alumnos que se va procesando, o sea se van acumulando una a una las notas; es por ello que **Sn** recibe el nombre de **sumador**. En ese caso observa que en el paso (2) esta la instrucción **Sn = 0**, esto es porque para ir acumulando en esa variable tiene que contener inicialmente un valor al cual se le va a sumar la primera nota que se procese, luego se dice que en este paso se está **inicializando el sumador**. Además en el paso (9) tenemos la instrucción **Cs = Cs + 1**, en este caso está utilizada la variable **Cs** para poner el conteo de la cantidad de alumnos suspensos que voy procesando, por lo que se le da el nombre de **contador**, muy usado en ciclos para contar de uno en uno elementos que se vayan procesando o los que cumplen una determinada condición.

Veamos un ejemplo de cómo funcionaría cada uno:

Ejemplo 21:

Se tienen las notas de Matemáticas de 4 alumnos de un grupo:

Pablo	87
Lissette	45
Luis	52
Omar	91

En este caso antes de entrar al ciclo Sn = 0 y Cs = 0, paso (2).

Al llegar al paso (5) como ya entró la nota de Pablo, o sea 87, entonces en Sn ocurre que:

Sn = 0 + 87

luego en Sn se guardará ahora el resultado de esa suma, o sea, Sn = 87.

Al concluir el procesamiento de Pablo como no he procesado la cantidad de alumnos indicada, al entrar la nota de Lissette en Sn ocurre que:

Sn = 87 + 45

Luego: Sn = 132

pero en este caso como Lissette está suspensa en Cs ocurre que:

Cs = 0 + 1

O sea se incrementaría este contador en uno, conté un suspenso, por tanto

Cs = 1

Cuando procese a Luis ocurriría:

Sn = 132 + 52 Cs = 1 + 1

Sn = 184 Cs = 2

Al procesar a Omar:

Sn = 184 + 91 Cs = 2 (no se actualiza porque Omar no está suspenso)

Sn = 275

Por tanto, como ya analicé los 4 alumnos que me pedían, concluye el trabajo en el ciclo y por tanto automáticamente se calcularía el promedio:

Pn = 275: 4

Pn = 68,7

Además daría la cantidad de suspensos en este caso 2, que es el ultimo valor que se guardó en la localización Cs.

Veamos como realizare el rastreo para este tipo de algoritmo. Observa que de ahora en lo adelante obviaremos el referirnos a los pasos, lo que evitaría una tabla muy grande, ya que podemos en una misma línea trabajar todos los pasos que no se interfieran en la corrida del algoritmo. Recuerda además ir guiándote por el algoritmo y realizándolo paso a paso:

N	i	? i<=N	Nb	Nt	Sn	? Nt<=60	Cs	Pn	IMPRESIÓN
					0		0		
4	1	si	Pablo	87	87=0 + 87	no			Pablo, aprobado
	2	si	Lissette	45	132 = 87+ 45	si	1 = 0 + 1		Lissette, suspenso
	3	si	Luis	52	184 = 132 + 52	si	2 = 1 + 1		Luis, suspenso
	4	si	Omar	91	275 = 184 + 91	no			Omar, aprobado
	5	no						$68,7 = \frac{275}{4}$	Promedio de notas del grupo: 68,7 Cantidad de suspensos: 2
									Fin

► **Ciclos Indeterminados:**

En estos no se conoce la cantidad de veces que se repetirá el ciclo, o sea se desconoce la cantidad de juegos de datos que se va a procesar. En este caso utilizaremos la instrucción:

Sintaxis:

Mientras <condición > **hacer** <lista de inst.>

Donde:

<condición > : se pondrá una condición que permitirá que se ejecute el ciclo mientras se esté cumpliendo, pero en cuanto se deje de cumplir, Automáticamente se ejecutará la primera instrucción que está a continuación del bloque de instrucciones que forma el ciclo.

<lista de inst.> : una o varias instrucciones que se repetirán mientras se este cumpliendo la condición.

Cuando la usemos en una descripción algorítmica será en la forma siguiente:

Mientras <condición> **hacer**

Lista de instrucciones
que se repiten

Cuando esta instrucción es usada en un algoritmo, mientras se cumpla la condición, se repetirán las instrucciones de la lista que se encuentra dentro del ciclo; cuando se deje de cumplir automáticamente se pasa a ejecutar la instrucción que sigue al ciclo.

Ejemplo 22:

Observemos el siguiente algoritmo:

(1) …
(2) …
(3) Mientras < condición > hacer
(4) Instrucción 1
(5) Instrucción 2
(6) Instrucción 3
(7) Instrucción 4
(8) Instrucción 5
(9) Fin.

En este ejemplo debes tener en cuenta que se ejecutaran las instrucciones de los pasos (1), (2) y (3) y, si la condición se cumpla se ejecutarán las de los pasos (4) hasta el (6);

luego se ejecutaran los pasos del (4) al (6) repetidas veces mientras se este cumpliendo la condición. Cuando deje de cumplirse la condición, se ejecutarán las instrucciones de los pasos (7) al (9), ya que la instrucción del paso (7) es la que sigue al ciclo; produciéndose hacia ella un salto automático, al salir del ciclo.

Ejemplo 23:

Haga la descripción algorítmica para dadas las notas (Nt) de los alumnos de nombre (Nb) de un grupo, se obtenga la condición de cada uno en la asignatura analizada. Cuando la lista del grupo concluya se indicará con un (“ * ”) como valor de la variable Nb.

Respuesta:

Primero te proponemos que te detengas en ver que la primera diferencia que existe entre el ciclo **Determinado** y el **Indeterminado** es que en el texto del problema aparecerá en este último la indicación de cómo saber cuando se dejarán de repetir las instrucciones del ciclo, lo cual se hará siempre utilizando una de las variables de entrada de datos, por donde se entrara un valor imposible para ese dato cuando se quiera indicar que ya no se procesara ningún otro juego de datos, que en el primer tipo se sabe porque se conoce la cantidad de juegos de datos que se va a procesar.

El algoritmo quedaría por tanto como sigue:

```
(1) Datos: Nb
(2) Mientras Nb < > “ * ” hacer
(3)             Datos: Nt
(4)             Si  Nt >= 60  entonces
(5)                             Imprimir: Nb; “ Aprobado”
(6)                       sino  Imprimir: Nb; “Suspenso”
(7)             Datos: Nb
(8) Fin.
```

Donde: Nb: nombre de cada uno de los alumnos.
Nt: nota del alumno Nb que se esta analizando en ese momento.

Fíjate que el ciclo comienza desde el paso (2), pero como la condición de terminar el ciclo que esta en este paso es **Nb < > “ * ”**, o sea, depende del valor que tenga Nb, en un paso anterior hacemos la entrada del nombre del primer alumno para chequear si la lista esta vacía o no. Si se cumple la condición entramos al ciclo y se ejecutan las instrucciones de los pasos (3) al (6). En el paso (7) observa que se vuelve a utilizar la instrucción de entrada del dato **Nb**, esto es porque, cuando automáticamente vamos al paso (2) por el ciclo, debemos tener un nuevo nombre en la dirección **Nb**, que indicaría el siguiente alumno de la lista.

Luego el terminador (valor que toma la variable para terminar el ciclo) será cuando Nb = “ * ”, que se escribe entre comillas ya que no es ni un valor numérico ni una variable que se usará como localización, para guardar valores, sino que es un caractér que se introduce como valor de Nb ya que como no es un nombre de persona nos permite que cuando entre ese valor se salga del ciclo y se realicen el resto de las instrucciones que están fuera del ciclo, siempre saltando automáticamente a la primera instrucción que está a continuación del ciclo.

Ejemplo 24:

Haga la descripción algorítmica para dadas las notas (nt) de los alumnos (Nb) de un grupo se obtenga:

a) **La lista_donde se indique la condición de cada alumno en la asignatura analizada.**

b) **La cantidad de alumnos suspensos.**

c) **El promedio de las notas del grupo.**

Utilice como terminador del ciclo (*) como en el ejemplo anterior.

Respuesta:

En este ejemplo es necesario que te des cuenta que para resolver este problema debes utilizar un contador para los suspensos y otro general ya que no conocemos la cantidad de alumnos que tiene el grupo, y necesitamos saberla para poder hallar el promedio. Además utilizaremos un sumador para acumular las notas de los alumnos.

```
 (1)   C :=0; Sn :=0; Cs :=0

 (2)   Datos: Nb

 (3)   Mientras Nb < > “ * ”  hacer
 (4)             Datos: Nt
 (5)             Sn := Sn + Nt.
 (6)             C := C + 1
 (7)              Si  Nt >= 60  entonces
 (8)                     Imprimir: Nb; “Aprobado"
 (9)             sino    Imprimir: Nb; “Suspenso”
(10)                     Cs := Cs + 1.
(11)             Datos:Nb
(12)  Imprimir: “Hay”; c; “ alumnos suspensos ”
(13)  Pn := Sn : C
(14)  Imprimir: “Promedio de notas del grupo:” ; Pn
(15)  Fin
```

Leyenda:

C: contador de alumnos del grupo
Cs: contador de alumnos suspensos.
Nb: nombre de cada alumno.
Nt: nota de cada alumno.
Sn: sumador de notas.
Pn: promedio de notas

En este epígrafe hemos visto como trabajar con ciclos, aunque no son las únicas formas, ya que estas dependen de los lenguajes de programación que estés usando. Cuando los estudies deberás hacer adaptaciones de lo que aquí hemos explicado ya que cada uno tiene sus reglas específicas que deberás respetar. Aprenderás también que hay ciclos en que, según el problema que estés resolviendo, el terminador se chequea al final del ciclo y no como lo vimos aquí en el principio. En ese momento aprenderás como reconocer cuando lo debes hacer de una forma y cuando de otra, lo cual te permitirá trabajar con mayor libertad y más posibilidades.

Ejemplo 25:

Haga la descripción algorítmica para resolver el siguiente problema:

Dadas las estaturas de los N alumnos de un grupo, imprimir:

a) Estatura promedio del grupo.

b) Cantidad alumnos que miden más de 1,60m de altura.

Respuesta:

(1) Dato: N
(2) **i :=** 0; S **:=** 0
(3) Desde c **:=** 1 hasta N hacer
(4) Dato: Es
(5) S **:=** S + Es
(6) Sí Es > 1,60 entonces
(7) **i :=** **i** + 1
(8) Próximo c.
(9) Pr = S **:** N
(10) Imprimir: “El promedio de estatura del grupo es de”; Pr ; “m”.
(11) Imprimir: “Miden más de 1,60m”; i; “alumnos”.
(12) Fin.

<u>Leyenda</u>

N: cantidad de alumnos del grupo.
i: contador de los que miden mas de 1,60m
S: sumador de estaturas.
c: contador general.
Es: estatura de cada alumno.
Pr: promedio de estatura en el grupo.

Ejercicios Propuestos:

I. - Haga la descripción algorítmica para calcular los N valores de la variable que aparecen en el miembro izquierdo de las expresiones siguientes. Considere en los casos necesarios las indefiniciones.

g) $X = 5^a + 4$ g) $R = \dfrac{3b + c}{a + 2}$

h) $Y = 7k - h$ h) $S = \dfrac{4b}{a - 2}$

i) $Z = ab + c$ i) $P = \dfrac{4x^3}{x - 3}$

j) $K = 4x^2 - y$ j) $T = \sqrt{x^2 - y^2}$; Op: $\sqrt{a}$

k) $D = xy^2 + 3z$ k) $A = \dfrac{\sqrt[3]{x^3 - y}}{y - 4}$; Op: $\sqrt[3]{a}$

I) $Q = \frac{3a - 8}{x}$ I) $M = \frac{\sqrt[4]{c^2 - d}}{b - a}$; Op: $\sqrt[4]{x - y}$

II. - Haga la descripción algorítmica de los siguientes problemas. Realice en cada caso el organigrama correspondiente. Haga el rastreo para una cantidad suficiente de juegos de datos.

1. - Imprimir los números del 1 al 10, sus cuadrados y sus cubos.

2. - Dado un valor de N, imprimir los inversos de los números de 1 hasta N.

3. - Dada una lista de N números, si es posible calcule la raíz cuadrada de cada uno, en caso contrario imprima un texto adecuado.

4. - Dados N pares de números p y q, hallar el cuadrado del mayor y de ser iguales imprimir un texto adecuado.

5. - Dados N pares de números A y B, si la suma es positiva calcular su raíz cuadrada y si es negativa indicar su opuesto.

6. - Dados N pares de números A y B, calcule el cubo del mayor y de ser iguales calcule:

$$\mathbf{y = A \bullet B - A - B}$$

7. - Dadas las concentraciones de N disoluciones cuya formula es Fr, determinar para cada una si es base, ácida o neutra. Recuerde que:

$$\mathbf{pH = - lg\ C}$$

$$\text{si pH} \begin{cases} > 7 & \text{es base} \\ = 7 & \text{es neutra} \\ < 7 & \text{es ácida} \end{cases}$$

8. - Construya el algoritmo para calcular los N primeros múltiplos de un número A.

9. - Escribir la tabla de multiplicar del 5.

10.- Escribir la tabla de multiplicar de cualquier número del 1 al 10.

11. - Dada una lista de números imprimir el cubo de los números hasta la mitad de la lista y cuadrado de los restantes.

12. - En el Instituto de Meteorología se procesan las informaciones acerca de la presión que se observó en un grupo de estaciones a una hora determinada. El final de los datos estará indicado cuando la presión sea cero. Se desea determinar la presión media y la cantidad de estaciones con presión por encima de la normal (1 013 hPa).

13. - En el Instituto de Meteorología se procesan las informaciones acerca de la temperatura y la presión que se observaron en un grupo de estaciones a una hora determinada. El final de los datos estará indicado cuando la presión sea cero. Se desea determinar:

a) Temperatura y presión medias.

b) Cantidad de estaciones con temperatura menor que 25 grados Celcius y presión por encima de la normal (1013 hPa).

14.- En una cafetería se venden 3 productos que cuestan:

producto 1: refresco ------------------- $1.00

producto 2: pan con croquetas ------- $2.00

producto 3: bocadito de jamón ------ $3.50

Se quiere determinar:

a) La recaudación del día.

b) Lo vendido por cada producto.

El algoritmo debe terminar cuando se indique un producto 0.

15. - Calcular el promedio de las notas de un alumno. Considere que el algoritmo imprima los resultados cuando la nota sea –1, lo que indicara que ya se dieron todas las notas del alumno.

16. - Dadas las notas de los N alumnos de un grupo hacer el listado donde diga la condición de cada uno en la asignatura analizada.

a) Halle el por ciento de aprobados del grupo.

b) Diga el por ciento de calidad (alumnos con más de 85 puntos en el examen).

III. - Convierta los problemas anteriores a ciclo indeterminado o a determinado y escriba la descripción algorítmica correspondiente.

8. EJERCICIO DEL CAPITULO.

I.- Dadas las expresiones siguientes:

a) $C = 2a + 5$

b) $P = \dfrac{2x}{y + z}$

c) $R = \dfrac{\sqrt[3]{a^2 + 1}}{2b}$

d) $T = x^2 - \dfrac{5}{4x}$

e) $S = \dfrac{\sqrt{x - y}}{x^3 + 2y}$

f) $A = \dfrac{x + y}{c + \dfrac{x}{e + f}}$

g) $B = z + \dfrac{4x}{c + d}$

h) $W\left[\dfrac{m^3 + b}{c + d}\right]^2 + x^2$

1.1- Haga el algoritmo lineal correspondiente a cada una.

1.2- Haga el algoritmo con saltos de las expresiones en que haya divisiones, considerando las indefiniciones.

1.3- Haga el algoritmo para calcular N valores numéricos de la expresión.

II. - Realice la descripción algorítmica de los problemas siguientes:

1. - Calcule la edad promedio de cada una de los grupos de una escuela. Determine además el promedio de edad en toda la escuela.

1. - Dados N tríos de valores x, y, z se quiere determinar la cantidad de valores de z que cumplen la condición de ser menores que x + 2y.

3. - Se quiere saber en cuanto se venderá un lote de N naranjas si se conoce que:

a) Si pesan menos de 75g, valen 5 centavos.
b) Las que pesan entre 75g y 100g valen 7 centavos.
c) Cuestan 10 centavos las que pesan más de 100g.

4. - En una fábrica se elaboran N piezas y de ellas se consideran buenas aquellas cuyo diámetro oscila entre 20 cm. y 23 cm., determine el promedio del diámetro de las piezas buenas y el por ciento de las piezas en mal estado.

5. - De una caja de N tomates se desea saber cuantos de ellos pesan entre 8 y 11 onzas; cuántos, menos de 8 onzas y cuántos, más de 11 onzas y además indicar el peso total en libras.

6. - En una carrera de 100m planos los corredores clasifican para cada grupo de la final teniendo en cuenta que:

- Si el tiempo es menor que 10,10s va a la final A.
- Si está entre 10,10s y 10,25s va a la final B.
- Si es mayor que 10,25 va la final C.

Haga la descripción algorítmica para, dado el tiempo (t) realizado por el corredor en la semifinal, le diga en que grupo corre en la final.

7. - En una carrera de 100m planos participaron N corredores, de los cuales clasificaron los que hicieron un tiempo menor o igual que10,4s. Se quiere saber:

a) Número total de los que no clasificaron.
b) Promedio de tiempo de todos los corredores.

8. - Dado el siguiente algoritmo:

```
(1) Datos: N
(2) Desde i = 1 hasta N hacer
(3)         Datos: Nb, P
(4)         Si P >= 95 entonces
(5)                 Imprimir: Nb; “Titulo de Oro”.
(6)         Sino    Imprimir: Nb; “No opta”.
(7)         Próximo i
(8) Fin.
```

Leyenda

N: Cantidad de alumnos.
P: Promedio general del alumno
Nb: Nombre del alumno

a) Haga el rastreo del algoritmo para por lo menos 3 alumnos.
b) ¿Qué problema resuelve?.
c) Elabore un enunciado para dicho problema.
d) ¿Para qué se usa la instrucción del paso (4)?

9. - En un CDR se desea hacer el listado de todos los cederistas. Haga la descripción algorítmica para dados los N nombres (Nb) y edades (E) de los vecinos de la cuadra:

a) Se obtenga el listado de los que son cederistas.

b) Indicar cuantos no son cederistas.

Nota: Recuerde que una persona es cederista a partir de que cumple 14 años.

10. - Conociendo, de las N casas de una cuadra su número (NC) y su gasto eléctrico (GE) en el mes, calcular cuanto debe pagar (P) cada casa, si conoces que:

- ♦ Si GE ≤ 100 kWh se calcula como P = GE • 0,09
- ♦ Si 100 kWh ≤ GE ≤ 300 se calcula como P = GE • 0,20 – 11
- ♦ Si GE > 300 se calcula como P = GE • 0,30 – 41

b) Si el gasto eléctrico es mayor que 300kWh, envíele un mensaje adecuado referido al trabajo que se realiza en el país con el PAE.

Nota: Recuerde que para que su listado se pueda entender no sólo debe imprimir lo que paga cada casa sino también el número de la misma.

11. - Dados N tríos de valores A, B y C se desea calcular el promedio de cada uno de ellos y además imprimir el mayor promedio obtenido.

12.- Un encuestador realiza un censo de población en un reparto, y de cada vivienda obtiene: Total de mujeres, total de hombres y total de niños. Se quiere obtener el total censado en cada categoría. Escoja un terminador apropiado.

13.- Se desea investigar de un lote de varillas cuántas son defectuosas, si se sabe que las varillas deben tener su diámetro entre 10mm y 12mm y una longitud entre 60cm y 62cm. El último dato ésta dado por el diámetro 0.

14.- En un torno, dedicado a la confección de anillos, su producción se clasifica en: piezas buenas si el radio interior es de 4,9cm a 5,1cm, recuperables cuando de llevarse de nuevo al torno puede lograrse una medida correcta y desechable el resto. Se desea:

- Determinar la cantidad de anillos en cada categoría.
- Si el por ciento de anillos buenos es mayor de 90 imprimir un texto indicando que es un buen operario.

15. - En una carrera de 100m planos participaron N corredores, de los cuales clasificaron los que hicieron un tiempo menor o igual que 10,4s. Haga la descripción algorítmica para saber:

a) El número de clasificados.
b) Promedio de tiempo de todos los corredores.
c) Si el número de corredores clasificados es mayor que la mitad, imprimir “GRUPO DE CORREDORES BUENOS”.

16. - Se tiene información de los N alumnos de una escuela en la forma siguiente: EDAD, SEXO, GRADO. Haga la descripción algorítmica para obtener:

- la media general de edad de los alumnos.
- la media de edad por sexo.
- la cantidad de alumnos por grado y sexo.

17. - Haga la descripción algorítmica para calcular el valor de un inventario de productos donde:

C_i = Cantidad de productos, en unidades físicas.
P_i = Precio por unidades físicas del producto.

18. - Dados N valores de X, hacer el algoritmo para obtener la suma de sus cuadrados.

19. - Dados dos números positivos, calcular todas las potencias naturales del primero que sean menores que el segundo.

20. - Hacer la descripción algorítmica para determinar a partir de los pesos de la carga, en arrobas, de un grupo de camiones:

a) ¿Cuántas cargas exceden las 600@?
b) Si el total es mayor que 10 000@, imprimir un texto adecuado.
c) ¿Cuántos camiones tienen menos de 480@?

21. - Dada una lista de las calificaciones de estudiantes en las pruebas, se desea conocer la cantidad de estudiantes que no se presentaron, el por ciento de aprobados y la nota promedio obtenida (a los que no se presentaron se les pone NE como valor de la nota). Haga la descripción algorítmica para resolver este problema. Escoja un terminador apropiado.

22. - Se tiene el listado de un grupo de alumnos con el tamaño y el peso de cada uno. Haga la descripción algorítmica para obtener:

a) El promedio de tamaño y el promedio de peso.
b) Cantidad que pesan más de 200 libras.
c) ¿Cuántos miden más de 1,60 m?
d) Si no hay alumnos que pesen más de 200 libras, imprimir “NO HAY GORDOS.”

23. - En una mesa de control de taxis se desea realizar una investigación sobre el aprovechamiento de estos equipos y para ello se tiene por cada viaje realizado los datos

referentes a la cantidad de pasajeros y kilómetros recorridos. Haga la descripción algorítmica para conocer:

a) Promedio de pasajeros por viaje.

b) Total de kilómetros recorridos.

c) ¿Cuántos viajes se efectuaron con 4 pasajeros?

d) ¿Cuántos viajes recorrieron de 3 a 5 kilómetros?

24. - En un taller dedicado a la producción de piezas de repuesto se lleva diariamente el control total de piezas producidas por cada obrero, así como cuántas defectuosas produjeron. Se desea determinar:

a) Total de piezas confeccionadas en el día.

b) Cantidad de piezas defectuosas.

c) Promedio de piezas producidas por trabajador (buenas y defectuosas).

d) Imprimir un texto si no hay piezas defectuosas en el día.

25.-En la preparación de un atleta que compite en 3 000 m con obstáculos, este se entrena corriendo diariamente. Se tiene el control de los kilómetros recorridos en cada día de una etapa de su entrenamiento y se desea conocer:

a) Total de kilómetros recorridos.

b) Promedio de kilómetros recorridos por día.

c) Número de días en que no corrió.

d) Si el promedio de diario fue mayor que 12km y corrió todos los días, expresar mediante un texto que el entrenamiento ha sido bueno.

26.- Se tiene la producción de cada día de una semana de un central azucarero y se quiere saber:

a) Cantidad de veces que cumplió la norma potencial de 200 mil @.

b) Total producido en la semana.

c) Promedio dio de molida.

d) Imprimir un texto si se cumplió la norma potencial en cada uno de los 7 días.

27. - Para calcular, lo que se paga en una casa por gasto eléctrico en un mes, se procede de la siguiente forma:

Hasta 100kWh --------------- pago = GE · 0,09.
De 100kWh a 300kWh ----- pago = GE · 0,20 – 11.
Más de 300kWh ------------- pago = GE · 0,30 – 41.

Si, de las casas de una cuadra, se conocen el número de la casa (NC) y el gasto eléctrico del mes (GE). Obtener:

a) Cantidad de casas con gastos por encima de 300 kWh.

b) Promedio de gasto eléctrico de la cuadra.

c) Si el promedio de la cuadra es mayor que 300kWh, imprimir un texto adecuado. El proceso de entrada de datos termina cuando el número de la casa sea cero (NC = 0).

28. - Haga el rastreo de los problemas anteriores para no menos de 4 juegos de datos.

CAPITULO No 3: LÓGICA PROPOSICIONAL.

3.1 Objetivo de la Lógica Matemática.

En el uso común del lenguaje frecuentemente encontramos la frase ***esto es lógico*,** con lo cual se expresa que, por ejemplo, una afirmación es clara y consecuente y puede ser considerada como razonable. Por el contrario, cuando una asociación de ideas se nos manifiesta como algo sin sentido, o inconsecuente, la clasificamos de ilógica, o sea, que no corresponde a la lógica. Naturalmente que estas explicaciones dependen frecuentemente de los sentimientos y son deducidas de la experiencia sobre un concepto por lo cual no bastan para cumplir las exigencias de un planteamiento científico. Sin embargo a través de la relación de la vida diaria reconocemos la importancia que se le da a esta disciplina, así como el objeto de estudio de la lógica, que es el pensamiento, la forma superior de la actividad psíquica del hombre.

La palabra ***lógica*** proviene del griego logos que significa idea, palabra, razón, razonamiento, por lo que aquí nos ocupamos de problemas que desde hace muchos siglos son objeto de investigación por parte de los científicos.

Naturalmente que otras ciencias se ocupan también de la cuestión relacionada con el razonamiento. A causa del constante desarrollo y perfeccionamiento de la sociedad humana y a la ampliación cuantitativa y cualitativa del documento a que esta da lugar, surgieron y surgen en la actualidad, nuevas disciplinas que están relacionadas con los procesos de pensamiento.

El pensamiento lógico se mantendrá mientras exista la humanidad. La lógica como ciencia existe desde hace más de 2000 años. Se tiene indicios de que ya en el siglo V a.n.e los chinos, los hindúes y los hebreos se ocuparon de las cuestiones de la lógica y en la antigua Grecia, Xenón, Sócrates y Platón hicieron grandes aportes, pero como creador de la lógica formal se considera al filósofo griego Aristóteles (384-322 a.n.e).

Producto de las investigaciones de los fundamentos de la Matemática y de otras disciplinas, la lógica formal se siguió desarrollando hacia la lógica Matemática actual. Cada ciencia desarrolla, mediante definiciones, un determinado vocabulario específico o terminología propia, en la que cada palabra tiene un significado exacto. El lenguaje natural y la terminología de cada ciencia en conjunto, constituyen entonces el lenguaje científico de esta disciplina.

En el uso científico del lenguaje es muy importante hacer una clara diferenciación entre el objeto, lo que observo, sus imágenes en el pensamiento, lo que me imagino y los símbolos lingüísticos, lo que percibo a través de la vida y la lectura. Entre ellas existen las siguientes relaciones:

El ***símbolo*** es la forma de existencia de una imagen en el pensamiento. El símbolo es el nombre, la denominación de un objeto.

La ***imagen en el pensamiento*** representa la significación de un símbolo.
El ***objeto*** se caracteriza por un símbolo.

3.2 Aplicaciones e importancia de la lógica proposicional o Matemática.

La lógica se ocupa del análisis de las proposiciones y demostraciones, proporciona ideas claras y procesos sobre la naturaleza de la conclusión deductiva, desarrolla el pensamiento funcional y hace una contribución esencial al desarrollo del pensamiento científico y creador.

La lógica matemática también tiene aplicación práctica en las ciencias técnicas. El álgebra lógica posibilitó el acceso a la cibernética y especialmente a la técnica de medición, control y regulación industrial de gran importancia para la automatización. También en el procesamiento electrónico de datos y en la ciencia de la información (Informática). La tarea del álgebra lógica es emprender el análisis y la síntesis de estas conexiones, empleando los medios de la lógica proposicional.

La Matemática no tiene una lógica propia, aunque se hable de *lógica matemática*, pero si tiene un estilo propio de razonamiento. La brevedad en la expresión, el proceso de reflexión estructurado con exactitud, la ausencia de saltos lógicos y la exactitud en la simbología son característicos de este estilo de pensar. La matemática se aspira a la concordancia óptima con un esquema lógico-formal. El estilo matemático de pensar posibilita en grado sumo, a causa de su concordancia, controlar la exactitud en el proceso de pensamiento. Con esto se pueden evitar los errores.

3.3 Formas de determinación de los conceptos.

Se hace necesario conocer los procedimientos para explicar lo nuevo. Como se trata de divisiones o clasificaciones, estableceremos primero lo que se entiende por concepto de *clase.*

Definición 1*.* Clase de individuos.
Una totalidad de individuos con una característica común, como mínimo.
Al tomar como base una o varias características se hace una división en un conjunto previamente dado de individuos. Pero, como las clases pueden tener características comunes, existen también las clases de clases o las clases de nivel superior.

Ejemplo 1.1
Una organización de base de la Unión de Jóvenes Comunistas (UJC) abarca todos los grupos de la UJC de un Politécnico. Pero estos son a su vez clases de miembros de la

UJC. Con la adición de otra característica un individuo puede pasar con otros elementos a otra clase.

Ejemplo 1.2
Si se señala la militancia en la UJC como una característica de clasificación, con ello se divide la población de nuestro país, tomada en este caso como dominio básico, en dos clases a saber, los miembros de la UJC y los que los son. Si, por el contrario se toma como base el hecho de las BPD, resulta entonces otra división. Ahora hay ciudadanos que son militantes de la UJC y de las BPD, otros que solo son una de las dos cosas y otros que no son ni de la UJC, ni de las BPD.

Del ejemplo anterior se tiene que el dominio básico dado se compone en subconjuntos por medio de la división en clases. Al mismo tiempo los subconjuntos abarcan todos los elementos del dominio básico, y cada elemento pertenece exactamente a un subconjunto.

Definición 2. Concepto.
El reflejo mental de una clase de individuos o de una clase de clases sobre la base de sus características inalterables.

Ejemplo 2.1
Mediante el concepto *maestro* se caracteriza una clase de hombres que forma y educa a los demás hombres.

Ejemplo 2.2
Los conceptos, naturalmente, pueden ser observados en el aspecto cuantitativo o cualitativo. En este sentido se habla de la *extensión del concepto* o del *contenido del concepto.*

La extensión designa todos los elementos de la clase observada, mientras que el contenido se caracteriza por las características comunes.

Ambas se delimitan mutuamente, pero constituyen una unidad, puesto que señalando las características se establece la extensión.

Ejemplo 2.3
Maestro primario, es el que educa y forma solo a alumnos de primero a sexto grado.

La ampliación de la extensión generalmente ocasiona una disminución del contenido y viceversa.

Definición 3. Definición.
Una definición es:

a) Una proposición que establece lo que es un objeto, como surge o como se le reconoce;
b) Una regla que fija cómo se debe emplear un signo lingüístico ;
c) Una proposición o regla que establece o fija lo que significa o debe significar un signo lingüístico.

Cuando se determinan los conceptos pueden cometerse fácilmente errores.

Señalaremos algunos errores que se cometen, aunque el hecho de detectar los errores supone naturalmente el conocimiento de la definición correcta.

Ejemplo 3.1

a) Se llama cuadrado al paralelogramo rectangular (falta la característica esencial de que los lados son iguales)
b) La FMC es la organización de masas cuyos miembros son mujeres trabajadoras, estudiantes y amas de casas, que son militante de la UJC.
c) Lo expresado se refiere, a una parte de los miembros de la FMC

Estos errores se pueden descubrir fácilmente. La inversión de la oración y la colocación de *todos* y *cada* o al añadir *solo* son distintas posibilidades.

Las definiciones no pueden contener directa, ni indirectamente el concepto por definir.

Ejemplo 3.2
La psicología es la ciencia que se ocupa del estudio de la psiquis.

- La definición debe ser lo menos negativo posible, aunque en algunos casos es inevitable.

Ejemplo 3.3
Los números irracionales son todos los números reales que no son racionales.

- Una definición no puede confundirse con la enumeración de los tipos de conceptos.

Ejemplo 3.4
Cuadriláteros son el cuadrado, el rectángulo, el rombo, el trapecio, etc.

Definición 4. Constante.
Entre las constantes tenemos los siguientes signos:
π, 0, 1, $\frac{2}{3}$ (Constantes numéricos)
λ, c, κ (constantes físicas)

=, < (constantes de relación)

Definición 5 Variable.
Una variable es un signo que representa cualquier elemento de un dominio básico previamente establecido.

Una variable se puede sustituir por el signo de cualquier elemento del domino básico. Entonces se habla de la sustitución de la variable, o de la interpretación de la variable.

Definición 6 Término.
Son las constantes, las variables y sus combinaciones con los signos de las operaciones y los signos técnicos.

Ejemplo 6.1

$$2; a; 3b; \frac{a+3b}{c}$$

Definición 7: Proposiciones
Las proposiciones son estructuras lingüísticas con sentido, verdaderas o falsas, (sin ambigüedad).

Ejemplo 7.1 Proposición verdadera
"Cuba es la mayor de las Antillas"
$5-2=3$

Ejemplo 7.2 Proposición falsa
"La playa de Varadero se encuentra en Cienfuegos".
$5*4=10$

Ejemplo 7.3 No son proposiciones.
- "El domingo es redondo " (oración sin sentido)
- "La tierra es el único planeta en el universo que tiene vida" (La afirmación pudiera ser verdadera o falsa, pero nadie podría decidir esto en este momento)
- $x \in N$; $8+x<12$ (No puede decirse si es verdadera o falsa mientras no se conozca el valor de x).

Definición 8: Forma proposicional.

Es una estructura lingüística que contiene por lo menos una variable y se convierte en una proposición cuando se sustituyen todas las variables por símbolos, que denotan objetos de dominio básico.

Ejemplo 8.1

$8 - x > 5$ Con $x \varepsilon N$ Se convierte en una proposición verdadera si se sustituye por $0,1,2$, para cualquier otro valor será falsa.

Las formas proposicionales se pueden clasificar en la forma siguiente: Aquellas formas proposicionales que mediante una sustitución por lo menos, se pueden transformar en una proposición verdadera, se denominan ***interpretables.*** Todas las demás se denominan ***no interpretables.***

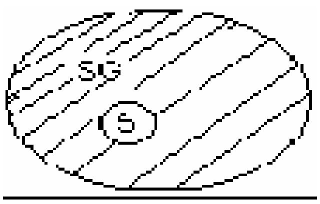
Entre las interpretables se destacan las formas proposicionales de ***validez general,*** que son aquellas que al hacer cualquier sustitución por los elementos del dominio básico se convierte en una proposición verdadera.

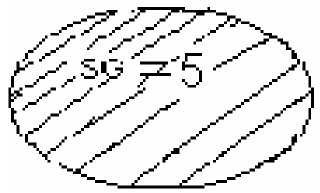
Ejemplo 8.2

a)8+x<12 Representa una forma proposicional interpretable(Conjunto solucion no vacío)

b) $x^2 = -1$ No es interpretable en el dominio de los números reales

c)a+b=b+a Es de validez general

<table>
<tr><th colspan="3"><u>FORMA PROPOSICIONAL.</u></th></tr>
<tr><th><u>No interpretable</u></th><th colspan="2"><u>Interpretable</u></th></tr>
<tr><th colspan="2"><u>Sin Validez general</u></th><th><u>De validez general</u></th></tr>
<tr>
<td><u>No interpretable:</u>
Cuando el conjunto solución no contiene elemento alguno

</td>
<td><u>Interpretable:</u>
Pero no tiene validez general cuando el conjunto solución es un subconjunto propio del conjunto base de solución.

</td>
<td>De validez general:
Cuando cada elemento del conjunto base de solución es también un elemento del conjunto solución.
</td>
</tr>
<tr><td><u>Contradicción</u></td><td><u>Neutralidad</u></td><td><u>Identidad</u></td></tr>
</table>

Ejemplo:

a) Validez general: $(a+b)^2 = a^2 + 2ab + b^2$ con a, b $\in$ R

b) Neutralidad $(a+b)^2 = a^2 + b^2$ con a, b $\in$ R

a	B	$(a+b)^2$	$a^2 + b^2$	Valor de Verdad
0	0	0	0	v
0	3	9	9	v
6	0	36	36	v
1	2	9	5	f
13	7	400	218	f

c) Contradicción: $x^2 - 5x + 10 = 0$ con x $\in$ R.

Ejemplo

Analizar las expresiones matematicas y determinar si son terminos, formas proposicionales o proposiciones

a) $4*3+5$
b) $3 < x < 7$
c) $\frac{a}{3} - 2b$
d) $\frac{14}{6} = 3$
e) $\sqrt{x^2 + y^2}$
f) sen de 45°<1

g) para cada xeR existe exactamente una yeR con la propiedad y=2x-1
h) 4 es un número par y 11 es un número primo
i) Si a < b entonces $a^2 < b^2$

Respuesta:

a) Un término cuyo valor es 17
b) Un término
c) Forma proposicional interpretable S={4,5,6}
d) Una proposición falsa
e) Forma proposicional interpretable.Conjunto solución "trio de números pitogóricos"
f) Proposición verdadera
g) Proposición verdadera
h) Proposición verdadera
i) Forma proposicional interpretable y de validez general

Ejercicios Propuestos:
Diga si las siguientes expresiones son términos, formas proposicionales o proposiciones.

a) $(y+3)^2 = y^2 + 6y + 9 \, con \, y \in R$
b) Los angulos opuestos son iguales
c) $6 = 2*2+2$
d) $a^2 + 7a - 5 = 0$
e) $6m + 5 = -1 \ ; m \in Z$
f) $10^2 + 3*4$
g) El ortocentro es un punto interior del triángulo rectángulo.
h) $sen^2 x + \cos^2 x = 1$

OPERACIONES LÓGICAS.

A partir de proposiciones se forman enlaces de proposiciones al igual que en el lenguaje común de oraciones sencillas se forman oraciones más complejas, luego mediante el enlace de las proposiciones se obtiene en todos los casos una proposición, cuyo valor depende tanto del tipo de enlace, como de los valores de verdad de las proposiciones enlazadas y en otros depende del valor de verdad de la proposición negada.

El proceso de negación se denomina **Operación Lógica de un lugar.**

Los enlaces de dos proposiciones se denominan: **Operaciones Lógicas de dos lugares.** Existen también operaciones lógicas de 3 o más lugares.

FUNCIÓN PROPOSICIONAL.

Cuando a n-uplo de proposiciones se le hace corresponder una proposición.
Las funciones proposicionales de mayor importancia son las **funciones proposicionales clásicas.**

	Negación	**Conjunción**	**Alternativa**	**Implicación**	**Equivalencia**
Función proposicional	"No **p**"	"**p** y **q**"	"**p** o **q**"	"**p,**entonces **q**"	"**p** exactamente cuando **q**"
Representación simbólica	**~ p**	**p** $\wedge$ **q**	**p**$\vee$ **q**	**p**$\rightarrow$**q**	**P**$\leftrightarrow$ **q**

NEGACIÓN.

La negación corresponde a la función proposicional de un lugar "no p"
Dado una proposición p, se puede formar otra proposición, que se llaman negación de p, cuyo valor de verdad es opuesto al valor de verdad de p.
Se denota **~ p**

Ejemplo
Dadas las proposiciones.
a) Varadero está en Matanzas.
b) Es falso que Varadero está en Matanzas.
c) Varadero no está en Matanzas.
Entonces (b) y (c) son cada uno una negación de (a).

Ejemplo.
a) $2+2=5$
b) Es falso que $2+2=5$
c) $2+2\neq 5$
Entonces (b) y (c) son cada uno negación de (a).

El valor de verdad de la negación depende de la siguiente condición.

Si **p** es verdadero, entonces **~ p** es falso, si **p** es falso **~ p** es verdadero.

El valor de la verdad de la proposición **~ p** se define mediante la tabla de verdad. Donde se demuestran todas las combinaciones posibles.

p	**~ p**
v	f
f	v

CONJUNCIÓN.

Dos proposiciones cualesquiera se pueden cambiar por medio de la palabra "y" para formar una proposición compuesta. Se denota $p \wedge q$.

Ejemplo.

Si **p** "Está lloviendo" y sea **q** "El sol brilla". Entonces $p \wedge q$ denota el enunciado. "Está lloviendo y el sol brilla".

Ejemplo.

$A \cap B = \{x / x \in A \quad \wedge \quad x \in B\}$.

El valor de verdad de la proposición compuesta $p \wedge q$ queda definido mediante la tabla de verdad.

p	**q**	**p ∧ q**
v	v	v
v	f	f
f	v	f
f	f	f

La conjunción **p ∧ q** es verdadera si **p y q** son ambas verdaderas; en cualquier otro caso **p ∧ q** es falsa.

ALTERNATIVA O DISYUNCIÓN no excluyente.

Dos proposiciones cualesquiera se pueden cambiar por medio de la palabra "o" (en el sentido "y/o") para formar una nueva proposición.

Se denota **p ∧ q.**

Es necesario advertir que la conectiva "o" tiene en nuestro idioma dos significados distintos:

La alternativa excluyente, como cuando decimos “escógelo a él o a mí” y la alternativa no excluyente, cuando decimos “Al sonar el despertador Pedro y/o Juan se despierta”. Es en este sentido no excluyente es en el que estamos considerando la alternativa

Ejemplo
Sea **p** “El estudió francés en la universidad” y sea **q** “ El vivió en Francia” Entonces $p \vee q$ es la proposición “ Él estudió francés en la universidad o él vivió en Francia.

Ejemplo
$A \cup B = \{x / x \in A \;\vee\; x \in B\}$

El valor de verdad de la proposición compuesta $p \vee q$ se define mediante la tabla de verdad.
La alternativa es falsa sólo cuando ambas proposiciones son falsas.

p	**q**	**p$\vee$q**
v	v	v
V	f	v
f	v	v
f	f	f

IMPLICACIÓN.

Cuando unimos dos proposiciones (simples y/o compuesta)mediante la conectiva “ Si , ..., entonces “ formamos una proposición compuesta conocida por **implicación ,condicional o implicación material.**
Las proposiciones particulares a partir de las cuales se forma la proposición compuesta se denominan antecedentes y consecuentes .

Se denota $p \to q$ se puede leer
p implica **q**
p solamente si **q**
p es suficiente para q
q es necesario para p

Ejemplo
a) $2*3n = (5+1)n$ implica $2*3 = 5*1$

b) $\sqrt{\frac{2}{3}\in R}$ implica $\sqrt{\frac{2}{3}}\in Q$

c) Si 5 < 3, 3 > 5 entonces sumadas ordenadamente 8 = 8

d) Si 2 es impar entonces 3 es par

Sólo (b) es falso, los otros son verdaderos

La implicación es falsa solo cuando una proposición verdadera implique una falsa.

El valor de verdad de la proposición $p \rightarrow q$ queda determinado en la siguiente tabla de verdad.

p	Q	p→q
v	v	v
v	f	f
f	v	v
f	f	v

EQUIVALENCIA.

Dada las proposiciones **p** y **q** se denomina equivalencia de estas proposiciones a la prosicion compuesta "**p** equivale a q" o " **p** si y solo si **q**"
Se denota $p \leftrightarrow q$.

Ejemplo
a) "Un entero es múltiplo de 5 si solo si su ultima cifra es cero o es 5"
b) "La vega es un planeta del sistema solar , cuando 2*2=5"
Ambos enlaces de proposiciones son verdaderas,ya que sus proposiciones parciales en a) son verdaderas y en b) ambas son falsas

La tabla de valores de verdad de la equivalencia es la siguiente.

p	Q	p↔q
v	v	v
v	f	f
f	v	f
f	f	v

La equivalencia es verdadera cuando ambas proposiciones son verdaderas o cuando son falsas

Ejemplo

Sean **p** “Hace frío” y **q** “Está lloviendo”
Describir con una proposición verbal las siguientes aserciones.

	Respuesta.
a) $\sim p$	a) No hace frío.
b) $p \wedge q$	b) Hace frío y está lloviendo.
c) $p \vee q$	c) Hace frío ó está lloviendo.
d) $q \leftrightarrow p$	d) Está lloviendo si y solo si hace frío.
e) $p \rightarrow \sim q$	e) Si hace frío entonces no está lloviendo.
f) $q \vee \sim p$	f) Esta lloviendo o no hace frío.
g) $\sim p \wedge \sim q$	g) No hace frío y no está lloviendo.
h) $p \leftrightarrow \sim q$	h) Hace frío si y solo si no está lloviendo.
i) $\sim (\sim q)$	i) No es verdad que no está lloviendo.
j) $(p \wedge \sim q) \rightarrow p$	j) Sí hace frío y no está lloviendo entonces hace frío.

Ejercicios Propuestos.

1- Determinar el valor de verdad de cada una de las proposiciones compuestas

a) Si $3+2=7$, entonces $4+4=8$
b) No es verdad que 2+2=5, si y solo si ,4+4=10
c) Marianao está en Matanza o el Cerro está en Cienfuegos.
d) No es verdad que $1+1=3$ o que $2+1=3$
e) Es el falso que si Marianao está en Matanzas, entonces el Cerro está en Cienfuegos

2- Lea P “Ella es inteligente” y sea q “Ella es estudiosa”
Expresa por enunciados verbales los siguientes enunciados simbólicos

a) $p \vee q$
b) $p \wedge q$
c) $q \rightarrow p$
d) $p \vee \sim q$
e) $q \leftrightarrow p$
f) $\sim p \sim q$
g) $\sim (\sim q)$
h) $(\sim p \wedge q) \rightarrow p$

3- Determina el valor de los siguientes enunciados.

a) Si 5<3, entonces –3<-5
b) No es verdad que $2+2=4$ o $3+5=6$
c) Es verdad de $2+2 \neq 4$ y $3+3=6$
d) Si 3<5 entonces –3<-5
e) Si Polo Montañés escribió la Guantanamera, entonces el era Habanero.
f) Si entre 13 y 15 hay números primos, entonces, entre 5 y 15 también hay números primos.
g) Si entre 5y15 hay números primos, entonces entre 13 y 15 hay números primos.
h) 7 es un divisor de 1001 y 3 es divisor de 24.
i) 7 es un número primo o 6 es un número primo.
j) no es 4=5
k) 1346 es divisible por 2 si y solo si 6 es un número primo.

4-Dada las siguientes formas proposicionales. Escriba la conjunción y disyunción de las funciones proposicionales de las funciones dada.

a)	$x+1=0$	$x^2-1=0$
b)	$(x+2^2)=0$	$x+2=0$
c)	$x \rangle 3$	$x \rangle 5$
d)	$x-7=0$	$x \rangle 5$
e)	$x^2-16=(x+4)(x-4)$	

5-dada las siguientes proposiciones compuestas, escríbela en forma simbólica designando las proporciones simple por p, q, r...

a) Estudiaré mucho y atenderé a clase.
b) 5 es un número primo o es compuesto.
c) No iré al cine.

6-Dado los vértice A, B, C, D de un paralelogramo y las proporciones, p: AB//CD, q, AD//BC, traduce a expresión verbal las siguientes expresiones simbólicas.

¿Cuál es el valor de cada uno'?

a) $p \vee q$

b) $p \wedge q$

c) ~ $p \wedge q$

<u>Expresiones de la lógica proposicional</u>

En el estudio de las funciones veritativas hemos designados las variables con p, q, etc. Su dominio básico. Su domino básico esta formado solamente por dos elementos, los valores de verdad V y F. Las palabras "y";"o",etc son según sus características constante que, entre otras cosas ,sirven para caracterizar las operaciones de la lógica proposicional. También las constantes en este caso la constituyen los caracteres (~,$\wedge$,$\vee$,$\rightarrow$,$\leftrightarrow$).Mediante el enlace lineal de las variables con los valores de verdad **p**, **q**, etc. y conectores, así mediante la aplicación de los signos técnicos ( ), podemos formar series de signos.

Ejemplo.

Z1= ($p \wedge q$) $\leftrightarrow$ ($q \wedge p$)

Z2= (~$\vee$ p_0 p_1, p_2)

<u>Definición</u>**<u>. Expresiones de la lógica proposicional.</u>**

<u>1- **Las variables p_j son expresiones**</u>

2- Si z es una expresión entonces –z es también una expresión

a) Si z_1 y z_2 son expresiones entonces (z_1 z_2), (z_1 z_2) ($z_1 \rightarrow z_2$) y ($z_1 \leftrightarrow z_2$) son también expresiones.

3-Una serie de signos Z es una expresión solo cuando se trata de los casos de 1y2

Luego del ejemplo anterior Z_1 si una expresión, no así la serie de signos Z_2 porque su estructura contradice la definición, los signos no pueden sucederse de forma directa (~ $\vee$) ni (p_0, p_1, p_2).

Ejemplo

Calcula los valores de verdad de la expresión.

H= $(\sim p \vee q) \leftrightarrow (r \rightarrow q)$

p	**q**	**r**	**$\sim p$**	**$\sim p \vee q$**	**$r \rightarrow q$**	**$(\sim p \vee q) \leftrightarrow (r \rightarrow q)$**
(1)	(2)	(3)	(4)	(5)	(6)	(7)
v	v	v	f	v	v	v
v	v	f	f	v	v	v
v	f	v	f	f	f	v
v	f	f	f	f	v	f
f	v	v	v	v	v	v
f	v	f	v	v	v	v
f	f	v	v	v	f	f
f	f	f	v	v	v	v

Del desarrollo de la columna 7 podemos deducir que la expresión dada H toma tanto el valor V como el F.

A este tipo de expresión la llamamos ***neutralidad*** de la lógica proposicional.

La expresión dada es satisfactible, porque al hacer, como mínimo una sustitución de la variable, toma el valor V.

Reconocemos entonces que las expresiones son aquellas series de signos que en lenguaje formalizado corresponden las formas proposicionales.

La tabla utilizada para calcular los valores de H es una tabla completa de valores de verdad. Esta contiene todos los valores de H que se han determinados en todas las interpretaciones posibles de la variables con los valores de verdad.

Ejemplo.

La expresión H= $(p \rightarrow q) \leftrightarrow (\sim q \rightarrow \sim p)$, en cualquier interpretación de las variables, toma el valor de verdad v. La tabla completa de valores de verdad demuestra esto.

p	**q**	**$p \rightarrow q$**	**$\sim q$**	**$\sim p$**	**$\sim q \rightarrow \sim p$**	**$(p \rightarrow q) \leftrightarrow (\sim q \rightarrow \sim p)$**
v	v	v	f	f	v	v
v	f	f	v	f	f	v
f	v	v	f	v	v	v
f	f	v	v	v	v	v

Es una expresión interpretable de validez general, luego es una ***identidad*** de la lógica proposicional llamada también **tautología.**

También existen expresiones no interpretables

Ejemplo.
H= **~** ((p $\vee$ (p $\vee$ q)) $\rightarrow$ (p $\vee$ q))

p	**q**	**p $\vee$ q**	**p $\vee$ (p $\vee$ q)**	**p $\vee$ (p $\vee$ q) $\rightarrow$ (p $\vee$ q)**	**~ ((p $\vee$ (p $\vee$ q)) $\rightarrow$ (p $\vee$ q))**
v	v	v	v	v	f
v	f	v	v	v	f
f	v	v	v	v	f
f	f	f	f	v	f

Una expresión de este tipo, donde los valores de verdad son existe una **contradicción** de la lógica proposicional.

Como una tautología es siempre verdadera, la negación de una tautología será siempre falsa ó sea que se trata de una contradicción y viceversa.

Igualdad de las tablas de valores de verdad.

Dada las expresiones
H_1 = p $\leftrightarrow$ q   y H_2 = (p $\rightarrow$ q) $\wedge$ (q $\rightarrow$ p)
Confeccionemos las correspondientes tablas completas de valores de verdad. Como en ambas expresiones se presentan las mismas variables, podemos utilizar la misma tabla.

p	**q**	H_1	**p $\rightarrow$ q**	**q $\rightarrow$ p**	H_2
v	v	v	v	v	v
v	f	f	f	v	f
f	v	f	v	f	f
f	f	v	v	v	v

Ambas expresiones H_1 y H_2 tienen la misma tabla de valores de verdad y se dicen **lógicamente equivalente**
$H_3 = H_1 \leftrightarrow H_2$
Para esta expresión confeccionamos la tabla completa de valores de verdad.
$H_3 = (p \leftrightarrow q) \leftrightarrow (p \rightarrow q) \wedge (q \wedge p)$

p	**q**	H_1	**p $\rightarrow$ q**	**q $\rightarrow$ p**	H_2	H_3
v	v	v	v	v	v	v
v	f	f	f	v	f	v
f	v	f	v	f	f	v
f	f	v	v	v	v	v

H_3 es una identidad de la lógica preposicional.

Las expresiones H_1 y H_2 reciben el nombre de *semánticamente equivalente* o simplemente *equivalentes*, cuando la equivalencia formada a partir de ellas es de validez general.

Ejercicios.

Hallar la tabla de valores de verdad de las siguientes expresiones. Clasifíquelas.

a) $H_1 = \sim(p \wedge q)$
b) $H_2 = \sim p \vee \sim q$
c) $H_3 = (p \wedge q) \vee (p \vee q)$
d) $H_4 = \sim(p \wedge q)$
e) $H_5 = (p \wedge q) \rightarrow (p \vee q)$
f) $H_6 = (\sim(p \wedge q)) \vee (\sim(p \leftrightarrow q))$
g) $H_7 = [p \rightarrow (\sim q \vee r)] \wedge \sim[q \vee (p \leftrightarrow \sim r)]$

Algunas leyes de la Lógica proposicional.

Teorema 1.

Para la conjunción, las alternativas y la equivalencia se cumple la ley conmutativa y la ley asociativa con respecto a la igualdad de las tablas de valores de verdad.

Para la implicación no se cumple ni la ley asociativa, ni la ley conmutativa.

Asociativa

a) $p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$
b) $p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$
c) $p \leftrightarrow (q \leftrightarrow r) \leftrightarrow (p \leftrightarrow q) \leftrightarrow r$

Teorema 2

La conjunción es con respecto a la alternativa, a ambos lados, **distributiva** y viceversa, o sea, que las siguientes expresiones son identidades de la lógica proposicional.

a) $p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$
b) $(p \vee r) \wedge p \leftrightarrow (q \wedge p) \vee (r \wedge p)$
c) $p \wedge (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$
d) $(q \wedge r) \vee p \leftrightarrow (q \vee p) \wedge (r \vee p)$

Conjuntamente con la distributividad, se cumple que la implicación, con respecto a las demás funciones veritativas, es **distributiva** a la derecha, pero no distributiva a la izquierda.

e) $p \rightarrow (q \wedge r) \leftrightarrow (p \rightarrow q) \wedge (p \rightarrow r)$
f) $p \rightarrow (q \vee r) \leftrightarrow (p \rightarrow q) \vee (p \rightarrow r)$
g) $p \rightarrow (q \rightarrow r) \leftrightarrow (p \rightarrow q) \rightarrow (p \rightarrow r)$
h) $p \rightarrow (q \leftrightarrow r) \leftrightarrow (p \rightarrow q) \leftrightarrow (p \rightarrow r)$

Teorema 3 **Unión de Premisa.**

Si la conclusión, segundo miembro de una implicación es igualmente una implicación, entonces las dos premisas, (primeros miembros), se pueden unir formando una sola premisa y recíprocamente. Luego es una identidad de la lógica proposicional.

$$p \rightarrow (q \rightarrow r) \leftrightarrow (p \wedge q) \rightarrow r$$

Una conjunción premisa de una implicación, se puede descomponer en varias premisas simples.

$$(p \wedge q) \rightarrow r \leftrightarrow (p \rightarrow r) \vee (q \wedge r)$$

Teorema 4 **Negación doble.**

Una expresión doblemente negada tiene la misma tablado valores de verdad que la correspondiente expresión dada, o sea.

$\sim \sim p \leftrightarrow p$

Teorema 5 **Contrarrecíproco**

a) $(p \rightarrow q) \leftrightarrow (\sim q \rightarrow \sim p)$
b) $(\sim p \rightarrow q) \leftrightarrow (\sim q \rightarrow p)$
c) $(p \rightarrow \sim q) \leftrightarrow (q \rightarrow \sim p)$
d) $(\sim p \rightarrow \sim q) \leftrightarrow (q \rightarrow p)$

OTRAS LEYES DE LA LÓGICA.

Leyes de Idempotencia.

$p \vee p \leftrightarrow p$
$p \wedge p \leftrightarrow p$

Leyes de la Identidad.

$p \vee F \leftrightarrow p$
$p \vee V \leftrightarrow V$
$p \wedge V \leftrightarrow p$
$p \wedge F \leftrightarrow F$

Leyes del Complemento

$\sim V \leftrightarrow F$
$\sim F \leftrightarrow V$
$p \wedge \sim p \leftrightarrow F$
$p \vee \sim p \leftrightarrow V$

Leyes de De Morgan.

$\sim (p \vee q) \leftrightarrow \sim p \wedge \sim q$
$\sim (p \wedge q) \leftrightarrow \sim p \vee \sim q$
$\sim (p \rightarrow q) \leftrightarrow p \wedge \sim q$
$\sim (p \leftrightarrow q) \leftrightarrow p \leftrightarrow \sim q \leftrightarrow \sim p \leftrightarrow q$

Transformación de las expresiones de la lógica proposicional.

Teorema 6 **Regla de la sustitución.**
Si en una expresión de validez general, o sea, en una identidad de la lógica proposicional, se sustituye una variable proposicional por una expresión cualquiera en todos los lugares donde se presenta la expresión correspondiente, entonces se obtiene nuevamente una expresión de validez general.

Ejemplo.
En la expresión $(p \rightarrow q) \leftrightarrow (\sim p \vee q)$, sustituimos la variable **q** por la expresión $(r \rightarrow s)$ se obtiene una expresión de validez general.
$(p \rightarrow (r \rightarrow s)) \leftrightarrow (\sim p \vee (r \rightarrow s))$.

En las expresiones semánticamente equivalentes tiene interés mencionar la siguiente regla.

Teorema 7 **Posibilidades de la Sustitución.**
Cuando en una expresión H_1, se sustituye en una cierta subexpresión $H^{\prime}$ por una expresión $H^{\prime\prime}$, toma los mismos valores de verdad que $H^{\prime}$. La expresión obtenida se llama H_2 y tiene los mismos valores de verdad que la expresión H_1.

Si $H^{´}$ está presente en varios lugares de H_1, debe sustituirse por $H^{´´}$ en todos los lugares donde esté presente.

Ejemplo.
Sea H_1 = (p ↔ q) ↔ (p → q) ∧ (q → p)

$H^{´}$ = (p → q) (subexpresión de H_1).

Sustituyendo $H^{´}$ por $H^{´´}$ = ~ q → ~ p

se obtiene

H_2 = (p ↔ q) → (~ q → ~ p) ∧ (q → p)

H_1 y H_2 tienen los mismos valores de verdad.

Ejemplo.
Simplificar la siguiente expresión.
(p ∧ ~ q) → p

(p ∧ ~ q) → p	Definición Implicación
~ (p ∧ ~ q) ∨ p	Leyes de Morgan
(~ p ∧ ~ ~ q) ∨ p	Negación doble
(~ p ∨ q) ∨ p	Ley conmutativa
(q ∨ ~ p) ∨ p	Ley asociativa
q ∨ (~ p ∨ p)	Complemento
q ∨ V	Identidad
V	Tautología

Ejercicios propuestos.

1) Simplifique aplicando las leyes de transformación.
   a) (p → q) → (q ∨ r)
   b) ((p ∧ ~ q) ∨ q) ∨ (q ∧ ~ p)
   c) [l → (m ∨ l)] ↔ n
   d) [(M ∨ ~ N) ∨ N] [(M ∧ ~ N) → N]
   e) [(p ∧ ~ q) ∨ q] → [(p ∧ ~ q) → p
   f) ~ { [p ∨ (p ∨ q)] → (p ∨ q) }

Aplicaciones

El cálculo de proposiciones no solo es utilizado para el tratamiento de problemas lógicos. Se conoce que toda proposición o enlace de proposiciones toman exactamente una de los dos valores de verdad posible, resulta la posibilidad de establecer la correspondencia entre los enlaces de proposiciones por una parte y dos “estados” relacionados entre sí por otra.

En el procesamiento electrónico de datos, esta conversión ha conducido a importantes resultados. Por tanto la lógica proposicional tiene aplicación directa en la dirección y conducción científica de la producción.

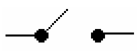
La aplicación de la lógica proposicional en el análisis y síntesis de circuitos eléctricos es de gran importancia para la técnica .estrechamente ligado a esta encontramos una ciencia joven: la cibernética, A*utómatas inteligente* que son capaces de manejar sistemas mecánicos complejos según los deseos del hombre.

Aplicación a los circuitos.

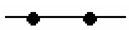
Se denominan circuitos eléctricos a la trayectoria conductora completa por la cuál circula o puede circular una corriente eléctrica.
Una RED. De cambio es una unión de alambres e interruptores que se conectan entre si a dos terminales.

Ejemplo
T_1 —— p—— T_2

Un interruptor puede estar:

Abierto 
p

Entra el flujo de la corriente, valor de verdad F

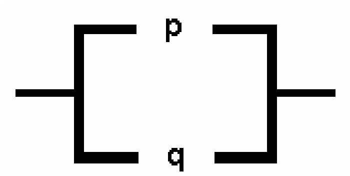
Cerrado 
~ p

Permite el flujo de la

Dos interruptores "p", "q" se pueden conectar por un alambre en

serie —— **p** —— **q** —— **(p$\wedge$q)**

La corriente fluye si y solo si, ambas son verdadera, es decir si p$\wedge$q es verdadera.

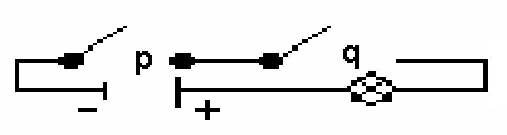
Paralel 

p $\vee$ q

Si se indica por **1** y **0** respectivamente, que un interruptor o circuito está cerrado o abierto, se tendrá que : ~ p V $\leftrightarrow$ 1 cerrado
p F $\leftrightarrow$ 0 abierto

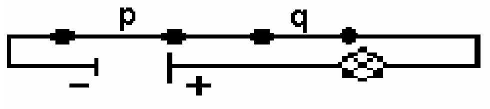
La corriente fluye si cualquiera de los dos interruptores está cerrado.
Ejemplos de conexiones sencillas y su relación con los enlaces proposicionales.

a)



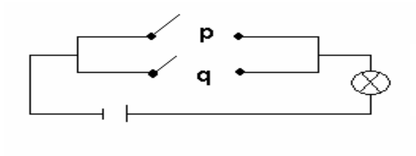
$\mathbf{p \wedge q}$ No fluye la

b)



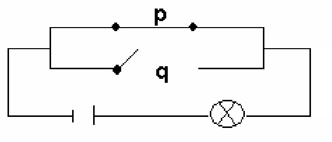
$\sim \mathbf{p} \wedge \sim \mathbf{q}$ Circula la corriente

c)



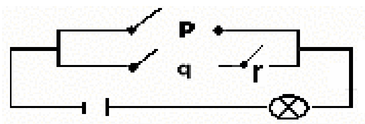
$\mathbf{p \vee q}$ No circula la

d)



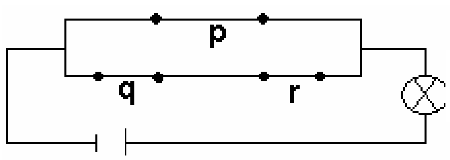
~ No circula la

e)



$\mathbf{p} \vee (\mathbf{q} \wedge \mathbf{r}$ No circula la

f)

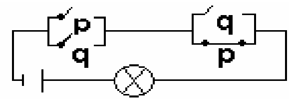


$\sim \mathbf{p} \vee (\sim \mathbf{q} \wedge \sim r)$

Fluye la corriente

Ejercicios propuestos:

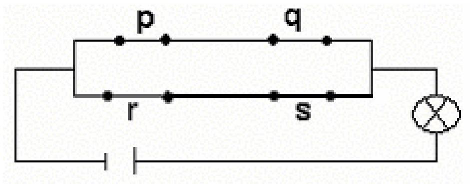
1. 1.1 ¿Qué enunciado corresponde al siguiente circuito?
   1.2 Construya una tabla de verdad para el enunciado anterior



   1.3 ¿Qué nos dice la tabla acerca de circuito?

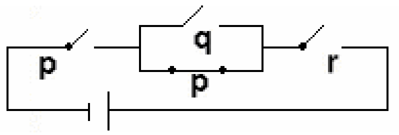
2. 2.1 Construya una red que corresponda a: $p \wedge (q \wedge \sim p)$
   2.2 Construya la tabla de verdad y diga si fluye la corriente

3. Determine el enunciado que corresponde al circuito.

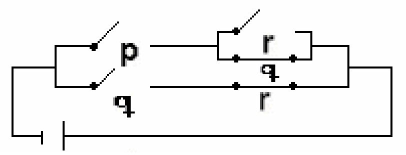


4. Determina los enunciados con enlaces proposicionales de los siguientes circuitos.

a)



b)



5. Construir un circuito por cada una de las expresiones siguientes.

a) $(p \wedge q) \vee [\sim p \wedge (\sim q \vee p \vee q)]$

b) $(p \wedge q) \wedge r \wedge (\sim p \vee \sim q \vee \sim r)$

CAPITULO No 4: LÓGICA DE PROGRAMACIÓN.

INTRODUCCION A LA LÓGICA DE PROGRAMACIÓN.

En la actualidad es prácticamente imposible encontrar un solo campo de las ciencias, las letras, las artes o negocios donde no se utilicen las computadoras y esto ha sido posible gracias a los avances de la electrónica.

El origen de la computación se remonta al propio origen de la civilización humana, pero tan lento como fue el desarrollo de la humanidad, en aquellos tiempos fueron los primeros pasos que se dieron en esa actividad.

Surgen diferentes instrumentos, primeros manuales, luego mecánicos hasta que casi a mediados del presente siglo aparece el primer equipo automático de cómputo, la primera computadora electrónica.

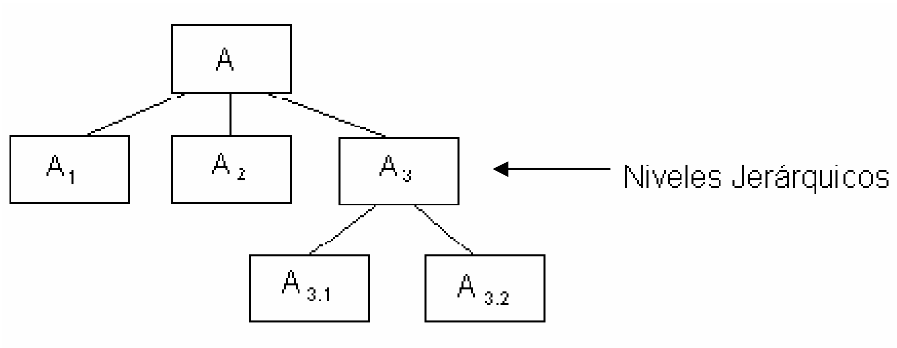
La computadora es un equipo automático capaz de recibir un conjunto de datos de entrada, procesarlos de acuerdo a un “Programa” quien entregará como salida de este proceso un determinado resultado.

Un ***Programa*** es un conjunto de órdenes o instrucciones que la computadora ejecuta secuencialmente para solucionar un problema. La labor del **programador** es confeccionar estos programas que serán luego interpretados y ejecutados por la máquina con vista a la solución de un cierto problema.

A partir del surgimiento de la compactación, comienza entonces el desarrollo de la programación para las mismas. Los primeros programas fueron un tanto caótico, se diseñaban sin una estructura determinada, las instrucciones se iban colocando en correspondencia con lo que se pretendía que la computadora ejecutara para resolver un cierto problema, de acuerdo a las concepciones del programador .Por esta razón los programas no eran eficientes y el proceso de rectificación o corrección eran difícil.

Cuando se analiza globalmente el problema haciendo un enfoque general del programa y se divide el mismo en otros subprogramas donde cada uno de ellos realiza una determinada tarea, el proceso de programación es más organizado y comprensible. Se hace un análisis descendente del programa, partiendo de lo general a lo particular. De esta manera, un problema se va dividiendo en problemas más pequeños hasta llegar a la unidad mínima divisible.

Estructura: Esquema que permite representar de manera simplificada alguna idea y que bajo condiciones normales es constante.



Esta filosofía para el diseño de los programas impone una nueva metodología de programación, una programación dividida en niveles y bloques o módulos, donde cada uno de ellos realiza una función específica. Esa estructura jerárquica de módulos de programas se ha denominado:

Programación estructurada.

De acuerdo a la tecnología de las computadoras las instrucciones se ejecutan una a continuación de la otra, es decir, en secuencia, por lo que tradicionalmente los lenguajes de programación empleados para escribir y construir los programas han exigido que las órdenes se coloquen manteniendo ese orden continuo, y así, para interrumpir esa secuencia con el propósito de ejecutar una instrucción que se encuentra en otra parte del programa, se diseñó una instrucción de"salto ";(GO TO)

Cuando se programa de forma estructurada o modular, esta instrucción de salto puede y debe evitarse tanto y como sea posible .Existen diferentes métodos de Programación Estructurada entre los cuales se pueden mencionar la Metodología Warner, la de Jackson, ADESA .En cualquier caso el objetivo de métodos de Programación Modular es enseñar a programar sin asociarse a un lenguaje de programación específico, sino a través de un seudolenguaje o lenguaje genérico.

En este curso estudiaremos el Método Jackson, el cual surgió en la década del 60, precisamente porque ya en esa época se sentía la necesidad de encontrar herramientas que facilitara el diseño de los programas.

Ideas básicas de la programación estructurada.

1. Los problemas deben descomponerse en estructuras jerárquicas con una disección de cada problema en sus correspondientes estructuras y partes. El método Jackson realiza esta descomposición de forma gráfica con una estructura arbórea.

2. Cada nivel de descomposición debe limitarse al uso de tres formas estructurales:
   .Secuencial
   .Selectiva
   .Iterativa
   (Estructuras básicas de control de la Programación Estructurada)
3. Siempre que sea posible se debe prescindir del salto incondicional (GO TO)

La primera forma que se empleo para representar gráficamente a los programas fueron los Diagramas de Bloque, los cuales se construían como una secuencia de acciones que no daba idea de jerarquía.

Con la aplicación de la Programación Estructurada, el diseño de un programa es más comprensible, ya que este se divide en subprogramas o módulos, en cada módulo se representa una tarea o función específica y para su representación solo se precisa de las tres estructuras básicas de control.

ESTRUCTURAS BÁSICAS DE CONTROL

SECUENCIAL.

Consiste en ejecutar un conjunto de acciones o instrucciones, una a continuación de otra. Es la estructura más simple. Para ejecutar una cierta instrucción, primero debe ser ejecutada todas las que le anteceden.

SELECTIVA.

Es una estructura que presenta caminos alternativos. La computadora tendrá que ser una selección de acuerdo con determinadas condiciones que se planteen en el programa y escoger un camino u otro en cada ejecución del mismo.
Existen dos tipos de selección:

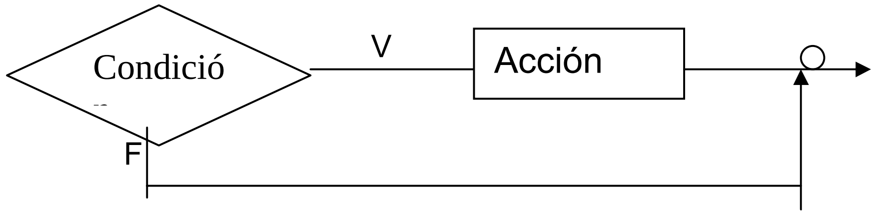
A. Selección simple
B. Selección múltiple

A. Selección o condicional simple.
Para este tipo de selección existen dos variantes:

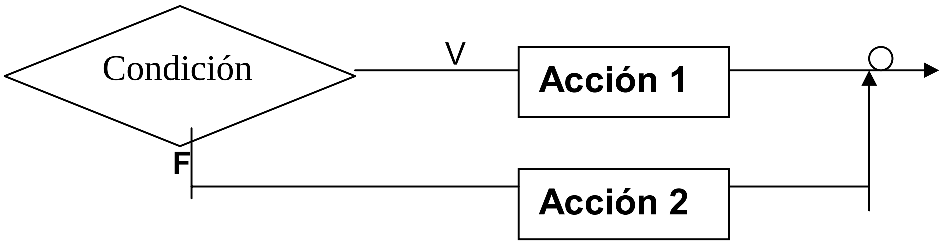
1- La acción será ejecutada si se cumple la condición planteada. La condición es una preparación con dos posibles respuestas, verdadera o falsa. En ese orden.

Ejemplo.

Si esta lloviendo, **abro** la sombrilla.
(Condición) (Acción)



En esta variante, (IF…THEN) si la **condición** que se esta evaluando es verdadera se ejecuta la **acción** que se indica, en caso contrario continua la secuencia normal del programa.
2- Una acción será ejecutada si se cumple la condición que se plantea, es decir, si se hace verdadera, en caso contrario se ejecuta otra acción diferente**.**

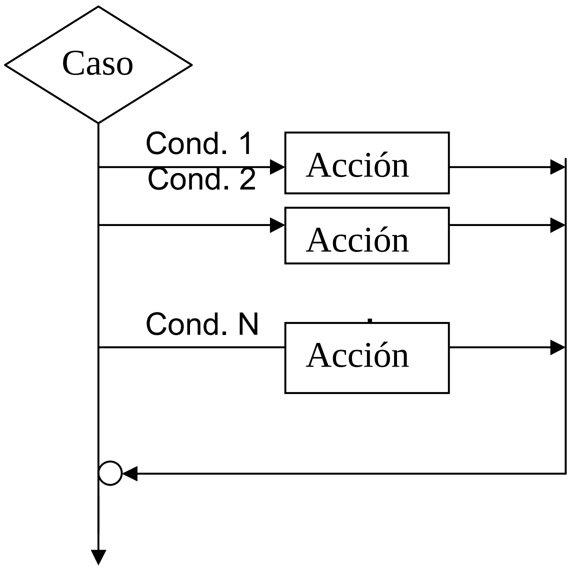


En esta variante, (IF…THEN…ELSE) si la **condición** es verdadera se ejecuta la **acción 1,** en caso contrario (falso) se ejecuta **acción 2.**

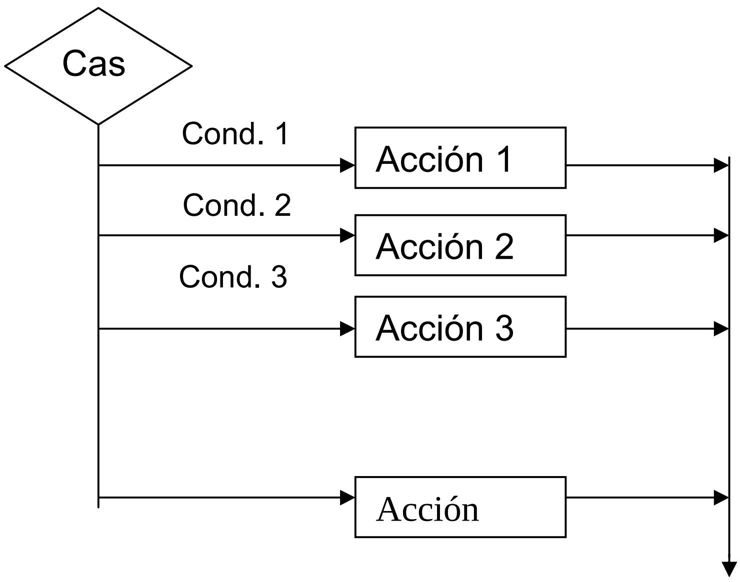
B. Selección Múltiple.

Este modelo de selección suele presentar varias alternativas, permitiendo hacer una elección entre las diferentes proposiciones. Dentro de este tipo tenemos también dos variantes.

1- Se evalúan varias condiciones y solo se ejecuta la acción que se corresponda con la primera condición que se hagan verdaderas. Si ninguna toma valor verdadero, entonces no se ejecuta acción alguna.



2- En esta variante también se evalúan varias condiciones y solo se ejecuta la acción que se corresponda con la primera condición que será la verdadera, pero si ninguna toma un valor cierto, entonces se ejecuta una acción que no esta sujeta a condición alguna.

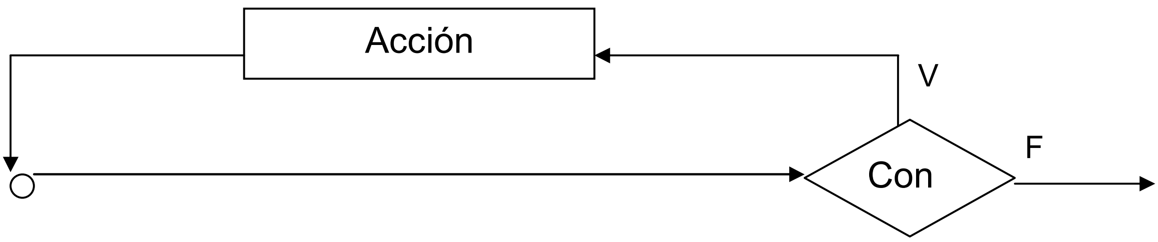


ITERATIVA.

La estructura iterativa consiste en repetir varias veces la ejecución de una determinada acción o grupo de estas. De forma general los ciclos pueden ser de dos tipos: Lógicos o Aritméticos.
Los ciclos de tipo aritmético tienen ciertas limitaciones ya que solo pueden ser usadas cuando se conocen el número de repeticiones que debe hacerse.

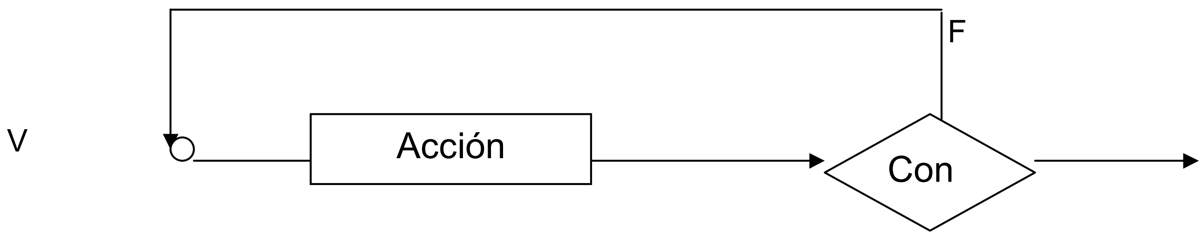
En el caso de los ciclos lógicos, lo que determina su final o salida, es el resultado de una expresión lógica (una condición que se evalúa como verdadera o falsa)

Existen dos variantes de ciclos lógicos: Do While y Do Until.
- Do While (Hacer… mientras)
Se evalúa una condición y **mientras** esta sea Verdadera, se **repite la acción** o grupo de acciones comprendidas dentro del ciclo.



Mientras la *condición* sea <u>*verdadera*</u> **repite** la ejecución de la *acción.*

-Do Until (Hacer… hasta que)
Se ejecuta una acción o grupo de estas, se evalúa una cierta condición y **hasta que** se haga verdadera, se **repite** la ejecución de la acción.



Repite la ejecución de la *acción,* **hasta que** la *condición* sea <u>*Verdadera.*</u>
El Método Jackson utiliza el ciclo **Do While** para representar las estructuras repetitivas o iteraciones.

PARTICULARIDADES DEL METODO DE JACKSON.

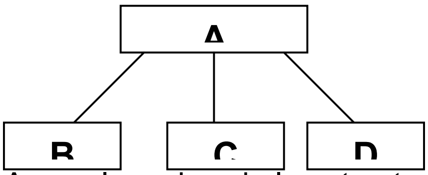
1- La estructura del programa debe corresponderse con la estructura de los datos de Entrada y de los resultados (salidas). Esto significa que se debe confeccionar el diagrama de las salidas, el de las entradas, y el del programa y entre ellos debe existir coherencia.
2- El sistema de notaciones empleado por el Método Jackson en la solución y desarrollo de un problema esta formado por:
   . Diagrama de estructuras (Representación Gráfica)
   . Lógica esquemática (Acciones o instrucciones del programa escrito.

3- Análisis Descendente: Debe partirse siempre de lo más general hacia lo más general hacia lo más particular. Un problema se va dividiendo en problemas más pequeños hasta llegar a la unidad mínima divisible: la acción **primitiva.**
4- Los programas están compuestos solamente por las tres estructuras básicas de control y por las acciones primitivas.

REPRESENTACION DE LAS ESTRUCTURAS DE CONTROL EN EL METODO DE JACKSON.

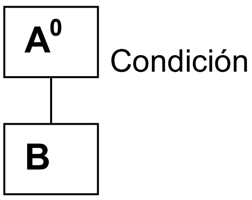
Diagrama de estructura.

1- Secuencial



A: es el nombre de la estructura secuencial
B, C y D: Son las acciones que se ejecutan de izquierda a derecha dentro de la secuencia A

2- Selección
. **Selección simple** (variante)
(IF… THEN)



A: es el nombre de la estructura selectiva.
B: es la acción que se ejecuta si la condición que se evalúa es *verdadera.*

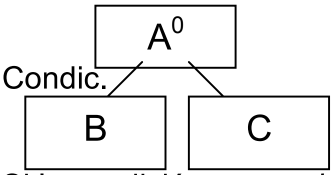
.

Lógica esquemática.

A sec
B
C
D
A fin

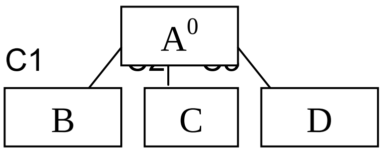
A selec condición
B
A fin

Selección simple (variante
IF…THEN…ELSE**)**



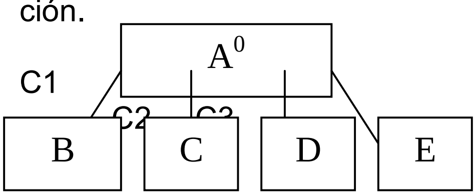
Si la condición es *verdadera* se ejecuta la acción **B,** de ser falsa se ejecuta **C.**

. Selección Multiple
donde todas las alternativas tienen asociada una condición.



La acción **B** se ejecuta si la condición **1** es verdadera.
La acción **C** se ejecuta si la condición **2** es verdadera.
La acción **D** se ejecuta si la condición **3** es verdadera.
Si todas las condiciones son falsas no se ejecuta ninguna acción.

. Selección Multiple
Donde aparece una alternativa que no tiene asociado ninguna condición.



Se ejecuta B si la condición 1 es verdadera.
Se ejecuta C si la condición 2 es verdadera.
Se ejecuta D si la condición 3 es verdadera.

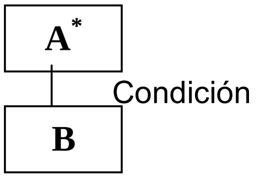
A selec condición
B
A o
C
A fin

A selec C1
B
A o C2
C
A o C3
D
A fin

A selec C1
B
A o C2
C
A o C3
D
A o
E
A fin

La acción E se ejecuta si no se cumple ninguna de las anteriores condiciones.

3- Iteración (Do while)



La iteración tiene una sola condición. La estructura A es una iteración que ocurrirá tantas veces como ocurra B, que es su parte iterada.

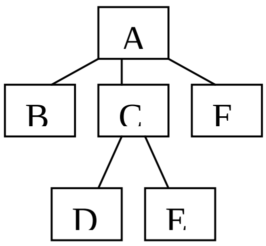
```
A iter mientras condición
        B
A fin
```

El sistema de notaciones será empleado en la confección de un programa, haciendo primer diseño del mismo a través del diagrama de estructura y luego la codificación de las acciones por medio de la lógica esquemática.

Ejemplos.

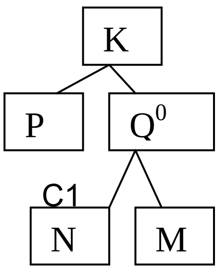
Construya la lógica esquemática a partir de los siguientes diagramas de estructura.

a)



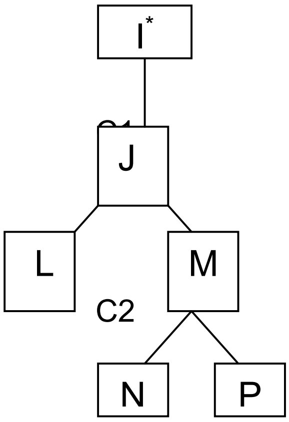

```
A sec
   B
   C  sec
        D
        E
   C  fin
   F
A fin
```

b)




```
K sec
    P
    Q selec C1
          N
    Q
          M
    Q fin
K fin
```

c)

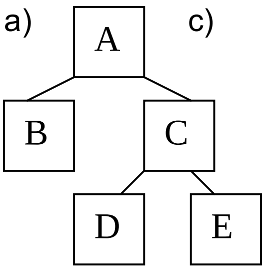



```
I iter mientras C1
   J sec
      L
     M selec  C2
          N
      M   o
          P
      M fin
   J  fin
I fin
```

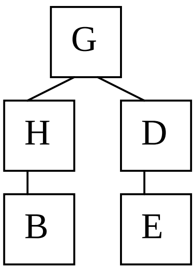
EJERCICIOS PROPUESTOS

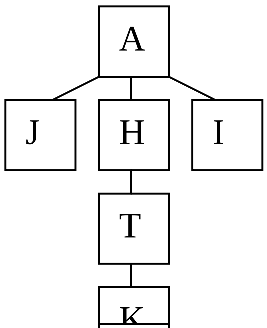
1- Realice la lógica esquemática a partir de los siguientes diagramas de estructura.

a)

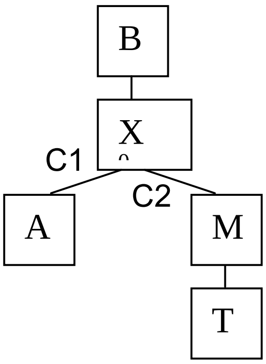


c)

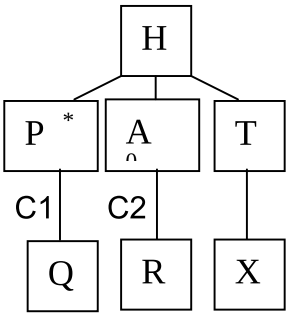




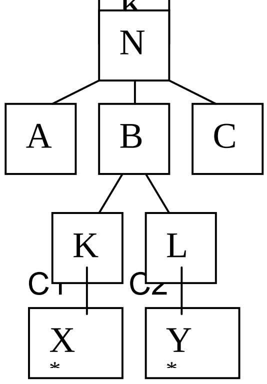
d)



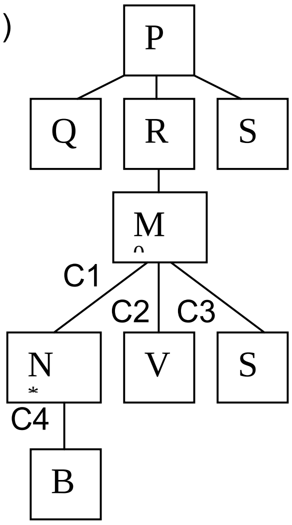
e)



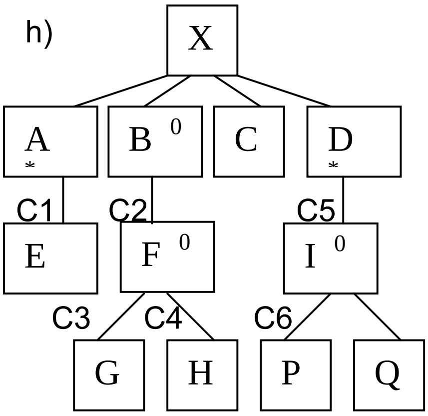
f)



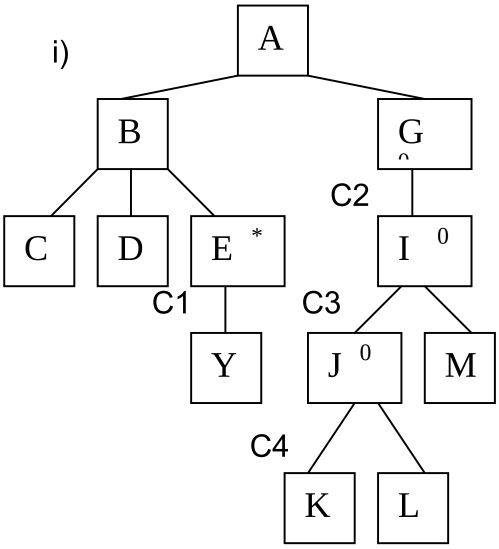
g)


P
Q
R
S
M
C1
C2
C3
N
V
S
C4
B

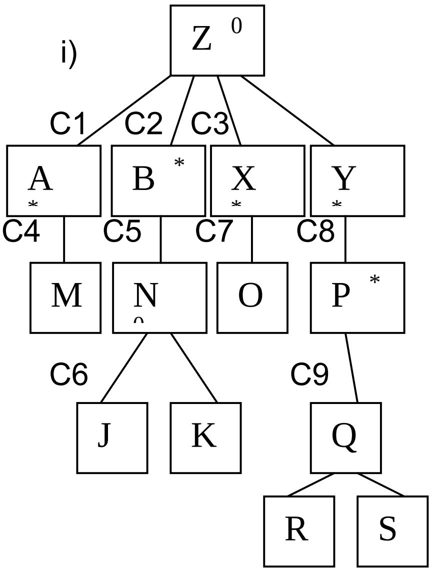
h)


X
A
B 0
C
D
C1
C2
C5
E
F 0
I 0
C3
C4
C6
G
H
P
Q

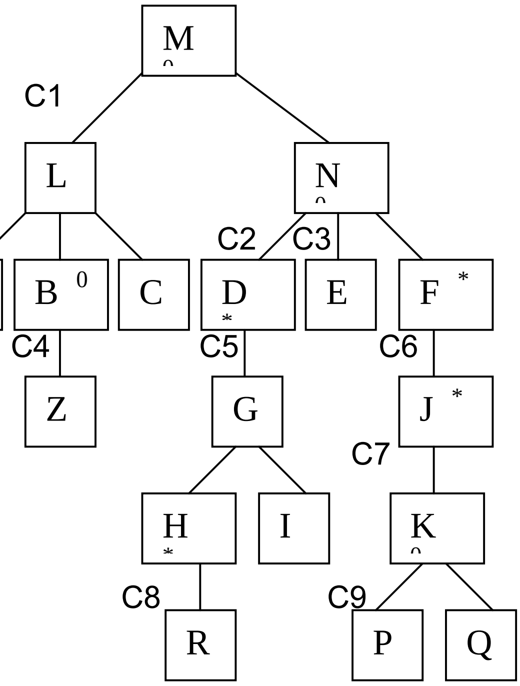
i)


A
B
G
C2
C
D
E *
I 0
C1
C3
Y
J 0
M
C4
K
L

i)


Z 0
C1
C2
C3
A
B *
X
Y
C4
C5
C7
C8
M
N
O
P *
C6
C9
J
K
Q
R
S

k)



2-Construya el diagrama de estructura dada la lógica esquemática.

```
a) N  sec
        P sec
            Q sec
                A
                 B sec
                     Y
                 B fin
           Q fin
       P fin
   N fin
```

C

```
b)   N sec
        A
        B sec
            X selec  C1
                 K1
            X    o    C2
                 K2
            X    fin
            Y iter mientras C3
                  L
           Y fin
        B  fin

     N  fin
```

```
c)   A sec
          B
          C selec  C1
               D iter mientras  C2
                        E
               D fin
          C  o
               F
          C fin
     A fin
```

```
d)  A  selec  C1
          B
    A   o
          C iter mientras C2
              D iter mientras C3
                  E iter mientras C4
                      F selec C5
                           G
                      F  o
                           H
                      F fin
                  E fin
              D fin
          C fin
    A fin
```

ESQUEMAS DE PROGRAMAS CON LA APLICACIÓN DEL METODO.
Aplicación del sistema de notaciones del Método Jackson. Diagramas de estructuras y Lógica Esquemática en el despeje de enigmas reales.

Ante cualquier problema de la vida nos encontramos que en la concepción del mismo surgen diferentes formas de resolverlos, pero no siempre el planteamiento es incorrecto y se dificulta la solución. Por lo que hace necesario prever la mayor cantidad de posibles soluciones de hacerse se corre el riesgo que avanzado el desarrollo de solución surjan estas y se tenga que regresar al punto de partida, con la correspondiente perdida de tiempo y recursos materiales.

Un diagrama de estructura aplicado a cualquier problema nos representa el recorrido de la solución en todas sus etapas, permitiéndonos conocer todos los pormenores.

Ejemplo:

De cómo se puede simplificar la solución de un problema a partir de la aplicación de un diagrama lógico del mismo.

La confección (no elaboración) de un cake es representativa de la eficiencia del Método Jackson.

Estructura del cake.

1- Base <circular o rectangular>
2- Panetelas <cantidad de panetelas>
3- Revestimiento <crema o merengue>

Pasos de la secuencia.

1- Confección de la base. <selección>
Condición: Circular o rectangular
2- Confección de las N panetelas <iteración>
Condición: Mientras sea menor que la cantidad de panetelas deseadas.
3- Montaje de las panetelas.
4- Confección de revestimiento <selección>
Condición: Crema o merengue.

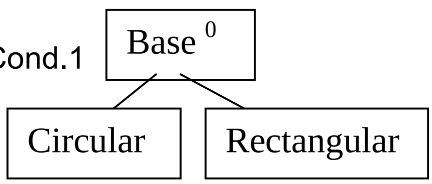
ANOTACIONES.

. La secuencia se representa al mismo nivel de jerarquía. (Puntos 1,2 y 3)
. Lo que se coloca debajo de cada elemento indica que pertenece a esta.
. Las condiciones tienen que tener respuestas de verdadero o falsa.

Análisis de cada paso (Modulo) de la secuencia cake.

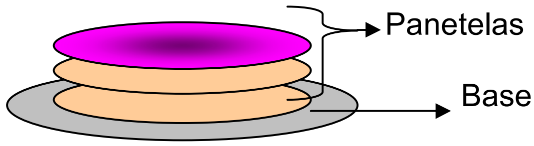
1- Confección de la base.

Cond.1



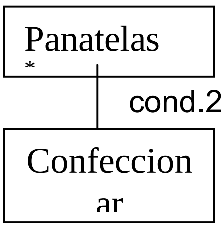
Cond.1 Se desea circular Se confecciona la base circular si cond.1 es verdadera. En caso contrario se confecciona base rectangular.

2- Confección de las N panetelas.
Análisis de la iteración
Vamos a asumir que se quiere un cake de **tres** panetelas.



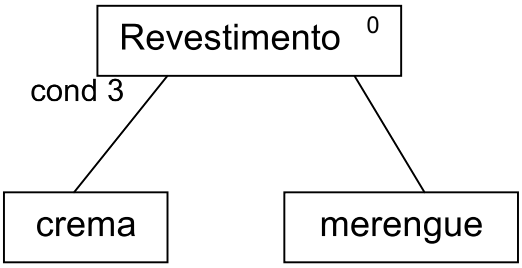
CAKE.

Número actual de panetelas		*Número deseado de panetelas*	*Respuestas*		*Acción*
0	<	3	Verdadero	→	confeccionar panetelas
1	<	3	Verdadero	→	confeccionar panetelas
2	<	3	Verdadero	→	confeccionar panetelas
3	<	3	Falso	→	terminar confección



Cond.2: Numero actual de panetelas < N
Siendo N: Cantidad de panetelas deseadas.
Si cond.2 es *Verdadera*, o sea, si el número actual de panetelas es menor que la cantidad de deseada.

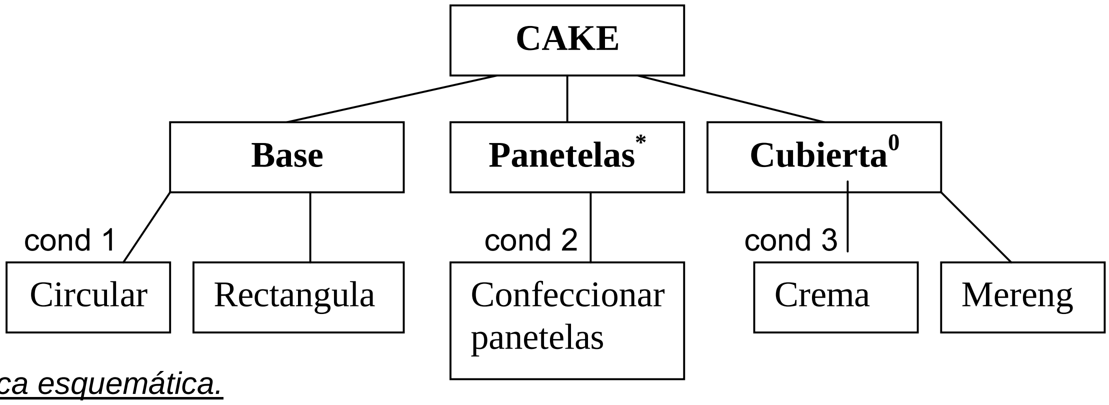
3.-Confección del revestimiento



cond 3. Se desea crema.

Se cubre con crema si la condición 3 es verdadera.
En caso contrario se cubre con merengue

Diagrama estructurado.



<u>*Lógica esquemática.*</u>
CAKE Secuencia
    Base selección condición 1 (*se desea circular*)
        Confeccionar base circular
    Base o
        Confeccionar base rectangular
    Base fin
    Panetelas iteración mientras cond 2 (numero actual de panetelas < n)
        Confeccionar panetelas
    Panetelas fin
    Cubierta selección cond 3 (se desea crema)
        Cubrir con crema
    Cubierta o
        Cubrir con merenge
    Cubierta fin
CAKE   fin.

Ejercicio resuelto.
. Hacer un programa que represente la construcción de un barco aplicando el Método Jackson y teniendo en cuenta los aspectos que relacionan a continuación.

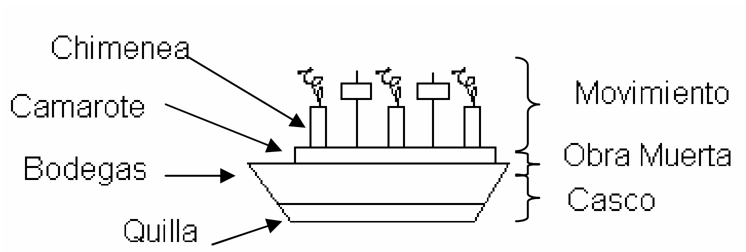
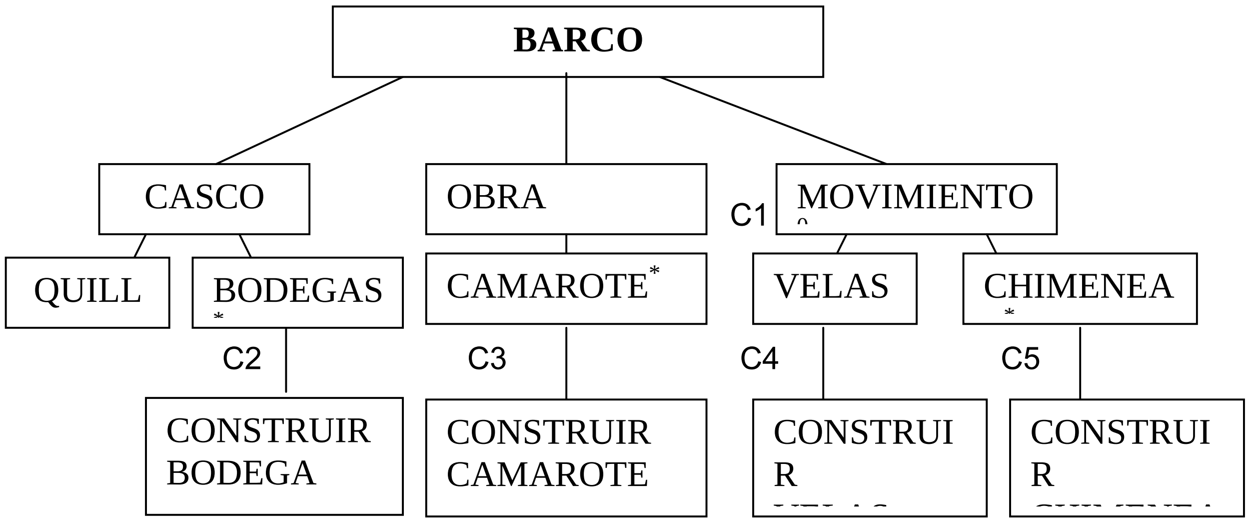


DIAGRAMA ESTRUCTURADO.



C_1: Se desea velas.

C_2: Cantidad de bodegas menor que las deseadas.

C_3: Cantidad de semestres menor que la deseada.

C_4: Cantidad de velas menor que la deseada.

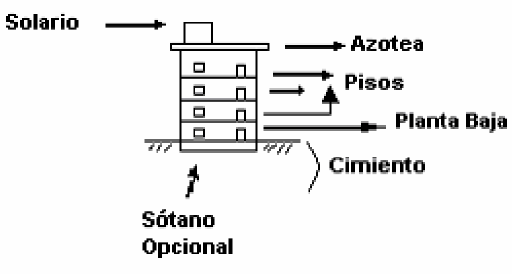
C_5: Cantidad de chimenea menor que la deseada.

Lógica esquemática.

Barco sec
Casco sec
Quilla
Bodegas inter mientras C_2
Construir bodega
Bodega fin
Casco fin
Obra Muerta sec
Camarote inter Mientas C_3
Construir camarote
Camarote fin
Obra Muerta fin
Movimiento selec C_1
Velas inter mientras C_4
Construir velas
Velas fin
Movimiento o
Chimenea inter mientras C_5
Construir chimenea
Chimenea fin
Movimiento fin
Barco fin.

Ejercicios Propuestos.

Hacer un programa que represente el proceso de construcción de un edificio.



HERRAMIENTAS NECESARIAS PARA LA PROGRAMACIÓN.
Constante:
Valor o magnitud conocida antes de la ejecución del programa y que no varía durante el mismo.

Ejemplo.
Constantes numéricas ($7, -55, \quad 3.1416\ldots$).
Constantes lógicas (verdaderas, falsas).
Constantes de cadena de caracteres (“A”, “B”, “ALFA”, “Ángel”).

Variable.
Elemento de los programas que representan a una zona de la memoria interna, en la cual se guarda una información que puede cambiar durante la ejecución de los mismos.

Las variables sirven para almacenar valores durante la ejecución, es decir son un recipiente de información.

Ejemplo.
Variables Numéricas: num y tiempo, (tomarán durante la ejecución del programa valores enteros o reales).
Variables Lógicas: Fin, error (tomarán dentro de la ejecución del programa valores verdaderos o falsos).

Operadores aritméticos.
Se usan para construir las expresiones aritméticas.

+	suma	a + b
-	resta	a - b
*	multiplicación	a * b
/	división	a / b

Expresiones Aritméticas.
Expresiones cuyo resultado es un valor numérico. Se forma por la combinación de constantes, variables y operadores aritméticos.

Ejemplo.
$5 + a / b$
Para la ejecución de las expresiones aritméticas se tendrá en cuenta el orden de prioridad de los operadores.

ASIGNACIÓN.
Es la acción mediante la cual se le da valor a una variable. El valor que se le asigna a una variable puede ser una constante u otra variable o el resultado de una expresión aritmética o el resultado de una expresión lógica o una cadena de caracteres.

Ejemplo.

Variable:= valor   variable ← valor

a: =1
b:= a
z:= "Luis"
s:= s + 1   (a la variable s se le asigna el contenido que poseía hasta ese momento).
Error:= falso.
Nombre:= "Samuel"

Si a una variable se le asigna un nuevo valor pierde el anterior, es decir en la variable siempre quedará almacenado el último valor que le sea asignado.

Operadores lógicos.

Son los que permiten hacer operaciones de tipo lógico o sea V o F. Sirven para limitar y controlar la circulación del programa.

<u>Negación:</u> **(not)**   **I**nvierte el valor lógico.

<u>O</u> **(or)**   Este operador debe colocarse entre dos operadores lógicos y el resultado de la operación será verdadera si al menos uno de sus operando también lo es.

<u>Y</u> **(and)**   Debe colocarse entre dos operando lógicos y el resultado de la operación será verdadero si todos sus operando también lo son.

Expresiones Lógicas.

Se forman por la combinación de valores lógicos con los **operadores** lógicos.

Ejemplo.

a:= V	(a o b) $\wedge$ c	
b:= F	(V o F) $\wedge$ V	
c:= V	V $\wedge$ V	El Resultado es Verdadero

Operadores de relación.
Devuelven un resultado lógico y permiten la comparación de dos operandos los cuales pueden ser, variables o expresiones aritméticas del mismo tipo de datos.
= Igual
< > Desigual
< **=** Menor o igual
> Mayor
> = Mayor o igual

Prioridad de los operadores.

+
-
*, 1, Y
+,-,0
<, >, >=, <=, =, < >

Las condiciones simples: Se forman a partir de expresiones lógicas simples que como máximo incluyen dos operandos. Las **condiciones compuestas** se forman a partir de expresiones lógicas que combinan operadores lógicos y de relación, o sea varias condiciones simples son integradas y evaluadas en una misma expresión que retorna siempre como resultado un valor verdadero o falso.

Las expresiones lógicas ayudan a construir las expresiones (simples o compuestas). Para las estructuras **selectivas** e **Iterativas.**

Sistema de Entrada / Salida.

En todo sistema de procesamiento de Datos, independientemente de su tamaño, tipo, costo, etc, existen entre ellos consideraciones básicas:

1. Datos de entrada al sistema.

2. Proceso planificado en el sistema.

3. Información resultante o salida del sistema.

En los Sistemas Electrónicos de Procesamiento de Datos o Sistema de cómputo, los datos se introducen a través de determinados dispositivos (denominados **periféricos** de entrada) tales como el teclado, scanner, mouse, mientras que la información resultante se envía al exterior por medio de otros dispositivos (conocidos como **periféricos de salida**) entre los que se encuentra el impresor, la pantalla o display, etc.

Los periféricos de entrada y de salida posibilitan la ejecución de las operaciones de **entrada y salida**. Estas operaciones permiten introducir los datos con los que va a operar el procesador central de la máquina y devolver los resultados de su trabajo, facilitando las interfaces entre el usuario y la computadora, simplificando así la comunicación hombre-máquina.

La entrada de datos acarrea una operación de **Lectura**; del mismo modo, la salida de la información resultante requiere de una operación de **Escritura.** Debido a que las unidades lectoras-grabadoras de discos ofrecen la posibilidad de ejecutar ambas operaciones, son consideradas dispositivos de entrada y de salida.

1- Operaciones de Entrada: Permiten la inserción de los datos que proviene del exterior. La entrada de los datos hacia el interior de la computadora se realiza a través de la operación **Lectura.**

- Lectura desde teclado; el teclado es el dispositivo estándar de entrada de un sistema de computo. La representación sintáctica de toda acción es :

Leer < variable >
Ejemplo:
Leer a (El valor introducido se asigna a la variable **a**).
Leer Beta (El valor introducido se asigna a la variable **beta**).
Cuando un programa durante su ejecución encuentra una orden de lectura desde teclado, este se detiene en espera de que se deprima cualquier tecla y no continúa ejecutándose un valor y este se deporta en una variable al deprimir la tecla ENTER
La lectura desde teclado provoca un eco por la pantalla del carácter o caracteres tecleados. Si deseamos que a continuación de la lectura se produzca un cambio de línea lo indicaremos de la siguiente forma:

Leer < variable> $/_{CL}$
La acción de *lectura* provoca una acción de *asignación,* solo que esta se confirma luego de pulsar la tecla ENTER.

- Lectura desde fichero en disco:

Leer **<** fichero**>, <** variable>

2- Operaciones de salida: Permiten entregar los resultados de un trabajo realizado por la computadora, así como los mensajes del sistema y los que le programador desee enviar. Esta información puede visualizarse en la pantalla o display, (dispositivo de salida estándar del sistema), el impresor o grabarse sobre un fichero en disco.

- Salida por el display.

Puede denotarse.
Escribir < variable> o Imprimir < variable>

Ejemplo:

Display

numer: =15
Escribir: numer

13_

beta: = 7
Escribir: beta $/_{CL}$

19
_

Escribir < texto>

Ejemplo:
Si se desea título:
Escribir: “Curso de lógica” $/_{CL}$
Si necesita solicitar algún dato
Escribir: “Teclee la edad:”
Si quiere enviar un mensaje de error:
Escribir: “Error. Debe teclear un valor distinto de cero”.

Escribir < texto>, < variable>, < texto>,…
Ejemplo:
a: = 1
b: = 4
S: = a + b
Escribir: “El resultado es =”, S $/_{CL}$ o
Escribir: “El resultado es =”, a + b $/_{CL}$
Salida a través del impresor.
Escribir <PRN>, <texto>, <variable>, <texto>,…
Escribir PRN, “Listado de estudiantes”
.Salida hacia ficheros en disco.
Escribir <fichero>, <variable>

PASOS PARA APLICAR EL MÉTODO JACKSON.

1-*Hacer el diagrama general donde se presenten todos los datos que se necesitan para resolver el problema, así como los resultados que se desean obtener a través de la pantalla o el impresor.*

2-*Hacer el diagrama de los resultados.* (Salida)
(Basándose en el diagrama general, hacer el diagrama de estructura de salida, en este no se incluye las condiciones lógicas)

3- *Hacer el diagrama del programa.*
(Basándose en el diagrama de estructura de la salida, hacer el diagrama de estructura del programa, este último tendrá las mismas estructuras de control que el de la salida y la misma cantidad de niveles, si bien no importa que sus secuencias tengan la misma cantidad de ramas o pasos)

4- *Escribir la lista de acciones primitivas,* las cuales se clasifican en los siguientes grupos.

I- *Aperturas, cierres y lecturas.*
(Se incluye todas las acciones que tienen que ver con abrir ficheros, cerrar ficheros y leer datos)

II-*Preparación de decisiones y decisiones.*
(Se escriben las asignaciones a variables de valores lógicos o de condiciones)

III- *Preparación de cálculos y cálculos.*
(Se tiene en cuenta todas las operaciones numéricas o darles valores numéricos a variables)

IV- *Preparación de salida y salida.*
(Se incluyen todas las acciones que tengan que ver con la entrega de resultados del programa)

Cada acción se escribe con un número consecutivo en la parte que corresponda. No importa en el orden que se ponen las acciones dentro de cada grupo, pero si es importante que las acciones queden clasificadas en el grupo que le corresponde.

V- *Distribuir las acciones primitivas.*
(De la lista en el diagrama de estructura del programa de forma lógica, situándolas en los nodos terminales)

Nota: Nodo terminar es el que se encuentra en el extremo de cada rama.

VI- *Escribir la lógica esquemática.*
(A partir de la distribución de las acciones primitivas)

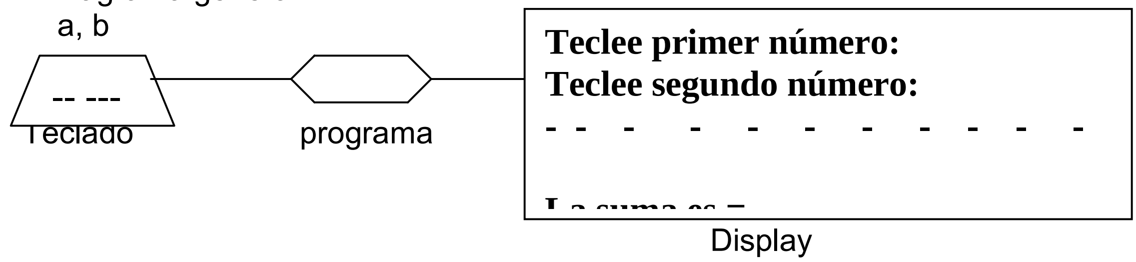
El desarrollo de todos los ejercicios se aplicaran los pasos establecidos en el Método de Jackson.

EJEMPLOS

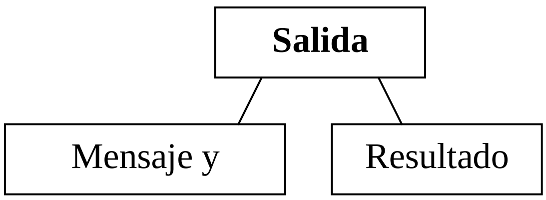
ESTRUCTURA SECUENCIAL.

1-Desarrollar un programa que lea por el teclado dos números y visualiza como resultado la suma de ambos.

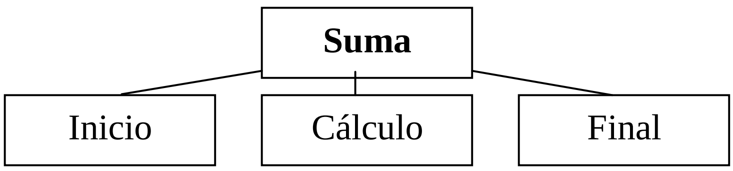
1- Diagrama general.



2- Diagrama de salida.



3- Diagrama de programa.



4-Lista de acciones primitivas.

I- Aperturas, cierres y lecturas.

1. Abrir teclado
2. Abrir display
3. Cerrar teclado
4. Cerrar display
5. Leer a
6. Leer b

II- Preparación de decisiones. (no hay)

III- Preparación de cálculos.

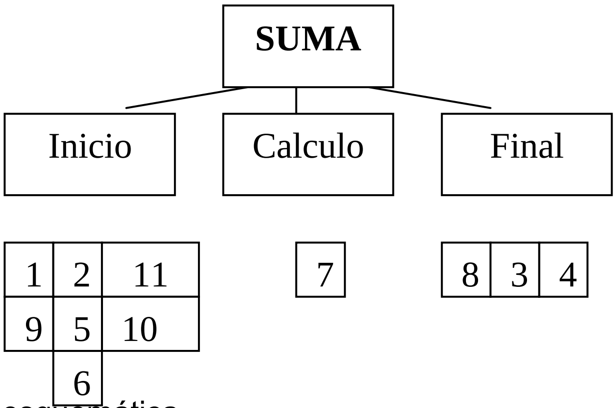
7. S:= a + b

IV- Preparación de salidas.

8. Escribir:"La suma es = "S / cl
9. Escribir:" Teclee primer número"
10. Escribir:" Teclee segundo número"

11. Limpiar pantalla.

5- Colocar las acciones dentro del programa según corresponda



6- Lógica esquemática.

```
SUMA   sec
          Inicio sec
                 Abrir teclado
                 Abrir display
                 Limpiar pantalla
                 Escribir:” Teclee primer número”
                 Leer a
                 Escribir:” Teclee segundo número”
                 Leer b
           Inicio fin
           Calculo sec
                 S:= a + b
           Calculo fin
           Final  sec
                  Escribir:”La suma es = “S  / cl
                  Cerrar teclado
                  Cerrar display
            Final fin
SUMA fin
```

Ejercicios propuestos:

1. Elabore un programa que lea como datos el largo y ancho de un rectángulo, escribir como resultado el área y el perímetro del mismo.
2. Diseñe un programa en el cual se lean como datos dos números. Imprimir como resultado que % es el primer número del segundo.

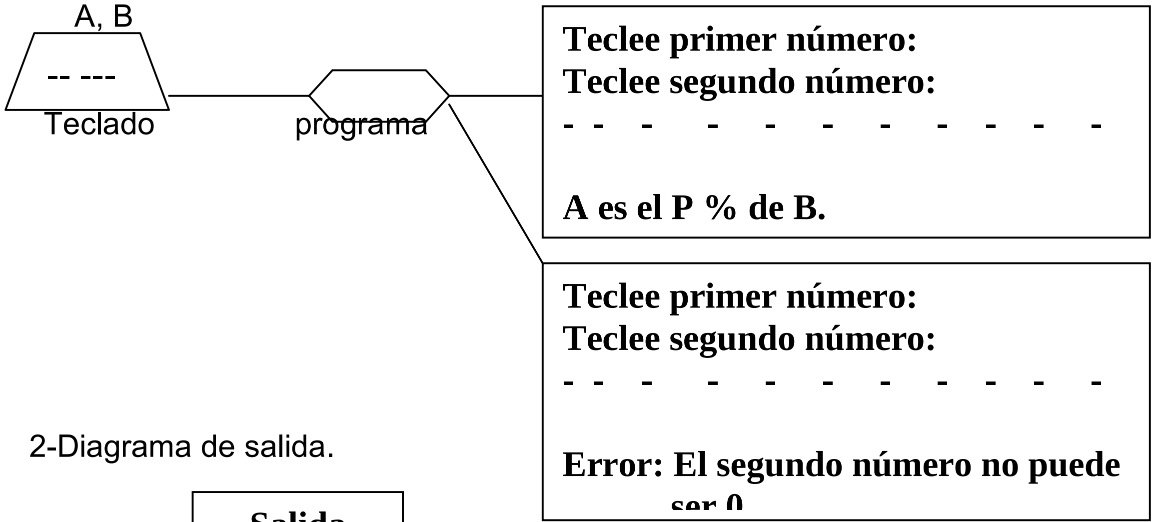
3. Confeccione un programa que permita calcular la Potencia Eléctrica (P) que consume una computadora dado el voltaje (V) y la corriente (I), si P = V * I.
4. Haga un programa que posibilite el cálculo del área de un trapecio.
5. Diseñe un programa para calcular el volumen de una pirámide de base cuadrada si $V=(l^2 * h) / 3$

ESTRUCTURA SELECTIVA SIMPLE.

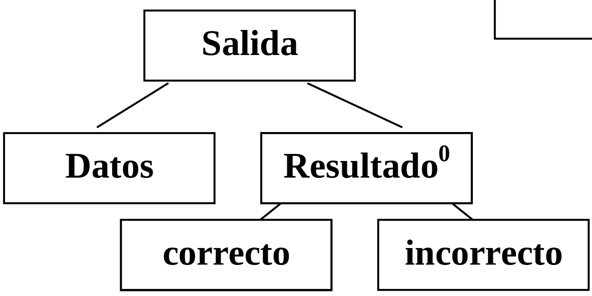
Ejemplo 1

Hacer una versión mejorada del programa que se calcula que % es un número de otro, en el cual se tenga en cuenta la posibilidad de que el segundo número sea 0 y en ese caso se imprimirá el mensaje correspondiente.

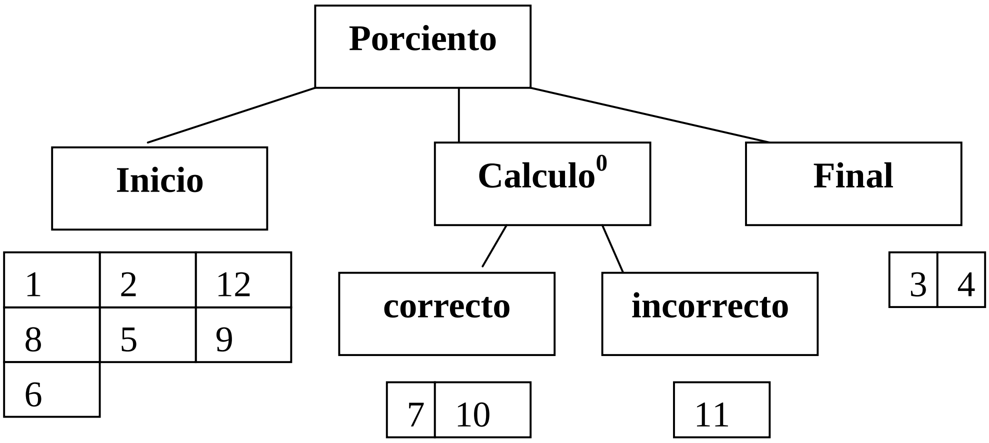
1- Diagrama general.



2-Diagrama de salida.



3- Diagrama de programa.



4-Lista de acciones primitivas.

I- Aperturas, cierres y lecturas.

1. Abrir teclado
2. Abrir display
3. Cerrar teclado
4. Cerrar display
5. Leer A
6. Leer B

II- Preparación de decisiones. (No se escribe, porque se situó en el diagrama de programa)

III- Preparación de cálculos.

7. P:= (A * 100) / B

IV- Preparación de salidas.

8. Escribir:” Teclee primer número”
9. Escribir:” Teclee segundo número”
10. Escribir: A,”es el“, P,”% de”, B / cl
11. Escribir:” Error: El segundo número no puede ser 0.”
12. Limpiar pantalla

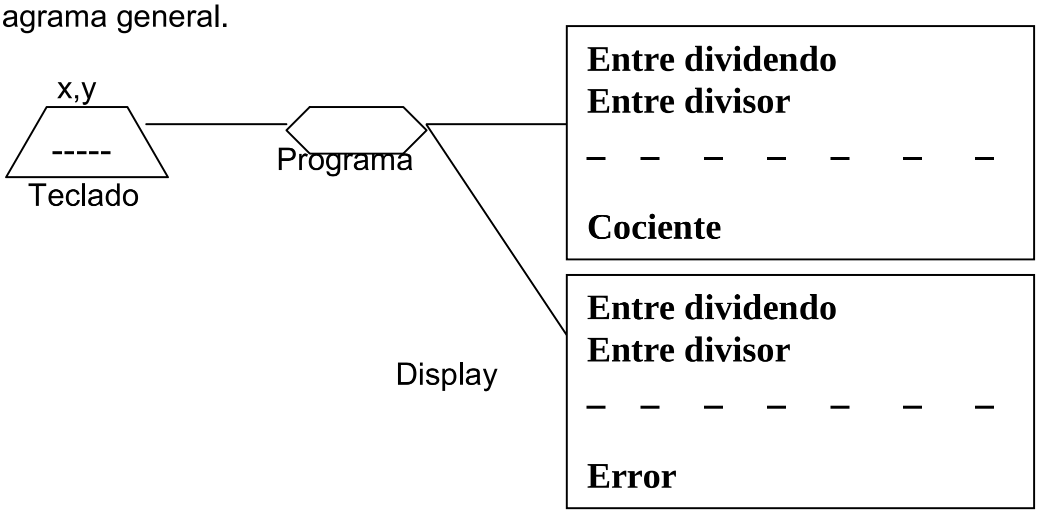
6. Lógica esquemática.

```
PORCIENTO  sec
            Inicio sec
                  Abrir teclado
                  Abrir display
                  Limpiar pantalla
                  Escribir:” Teclee primer número”
                  Leer A
                  Escribir:” Teclee segundo número”
                  Leer B
            Inicio fin
            Cálculo  selec
                  Si B<>0 entonces
                        Correcto  sec
                              P:= (A *  100) / B
                              Escribir: A,”es el“, P,”% de”, B  / cl
                        Correcto fin
                  Sino
                        Incorrecto  sec
                              Escribir:” Error: El segundo número no puede ser 0.”
                        Incorrecto fin
            Cálculo fin
            Final sec
                  Cerrar teclado
                  Cerrar display
            Final  fin
PORCIENTO fin
```

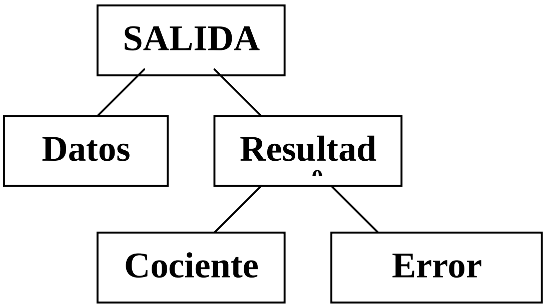
Ejemplo 2.

Elabore un programa en el que se lean como datos dos valores numéricos. Escribir a través de la pantalla su cociente.

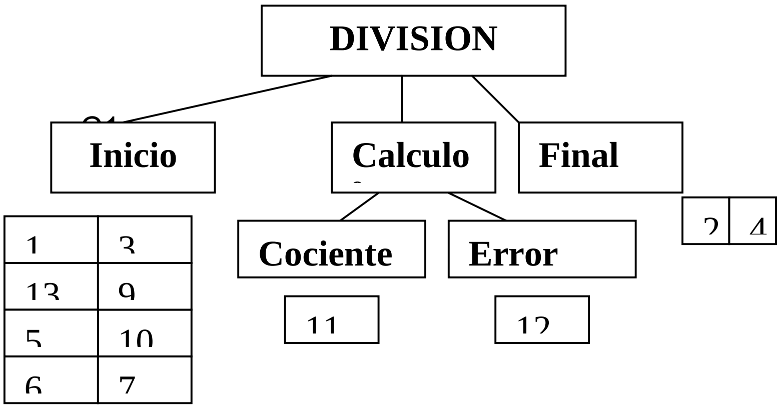
1- Diagrama general.



2- Diagrama de salida.



3- Diagrama del programa.



4- Lista de acciones.

I- Aperturas, cierres y lecturas.

1- Abrir teclado
2- Cerrar teclado
3- Abrir display
4- Cerrar display
5- Leer X
6- Leer Y

II- Preparación de decisiones.
7- C1:= (y <> 0)
III- Preparación de cálculos.
8- D:= X / Y
IV- Preparación de salida
9- Escribir:”Entre el valor del dividendo”
10- Escribir:”Entre el valor del divisor”
11- Escribir:” El cociente es =”, D /cl
12- Escribir:”Error el divisor no puede ser cero” /cl
13- Limpiar pantalla

6- Lógica esquemática.

```
DIVISION  sec
        Inicio  sec
            Abrir teclado
            Abrir display
            Limpiar pantalla
            Escribir:”Entre el valor del dividendo”
            Leer X
            Escribir:”Entre el valor del divisor”
            Leer Y
            C1:= (y <> 0)
        Inicio fin
        Calculo selec C1
            Cociente  sec
                D:= X / Y
                Escribir:” El cociente es =”, D /cl
            Cociente fin
        Calculo  o
            Error sec
                Escribir:”Error el divisor no puede ser cero” /cl
            Error fin
        Calculo fin
        Final sec
            Cerrar teclado
            Cerrar display
        Final fin
DIVISION fin
```

Ejercicios propuestos.

1- Haga un programa para dada la nota de un alumno poder determinar su condición (aprobado o desaprobado) en una asignatura.

2- Confeccione un programa para calcular la frecuencia eléctrica siendo la formula $f=\dfrac{1}{2\Pi\sqrt{L*C}}$

Donde f: frecuencia
L: inductancia
C: capacitancia
Π: 3, 14

3- Diseñe un programa que permita conocer si un numero dado esta en el intervalo [a;b]

4- Elabore un programa para calcular el logaritmo de un número, conocido la base y el argumento.

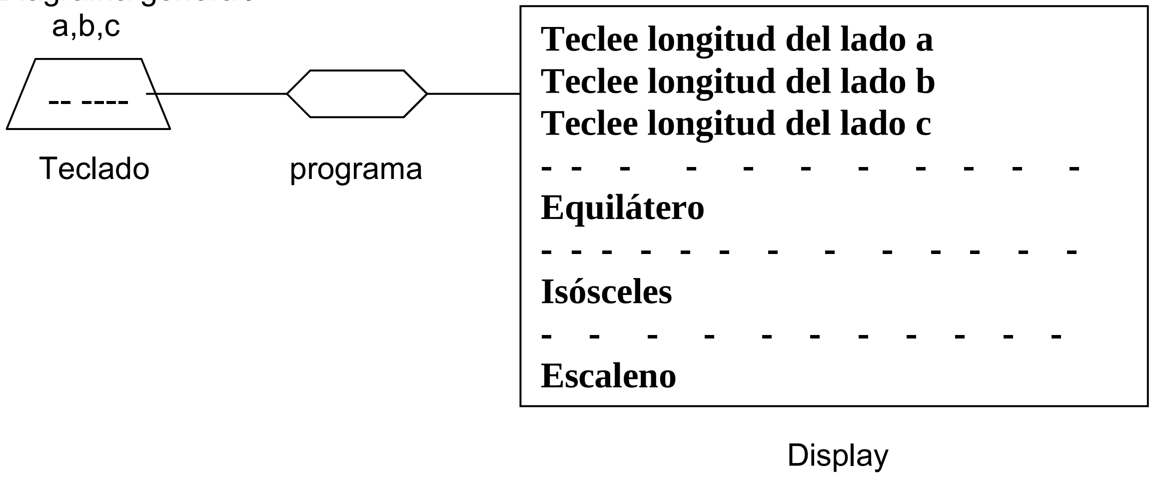
5- Haga un programa que permita evaluar la función f si $f(x)=\dfrac{3x+4}{x}$.

ESTRUCTURA SELECTIVA MULTIPLE.

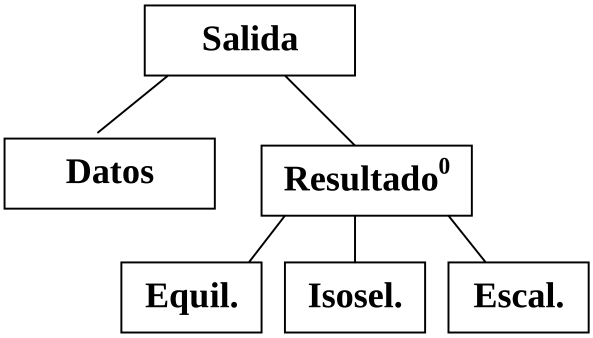
Ejemplo

Elabore un programa que dada las longitudes de los lados de un triángulo, lo clasifique en equilátero, isósceles o escaleno.

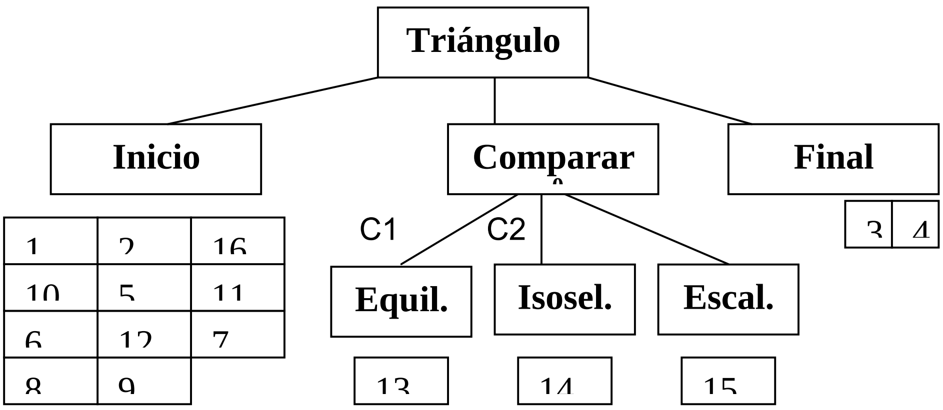
1- Diagrama general.



2- Diagrama de salida.



3- Diagrama de programa.



4-Lista de acciones primitivas.

I- Aperturas, cierres y lecturas.

1. Abrir teclado
2. Abrir display
3. Cerrar teclado
4. Cerrar display
5. Leer a
6. Leer b
7. Leer c

II- Preparación de decisiones.

8. C1:=(a=b) $\wedge$ (b=c)
9. C2:=(a=b) $\vee$ (b=c) $\vee$ (a=c)

III- Preparación de cálculos. (no hay)

IV- Preparación de salidas.

10. Escribir:” Teclee longitud del lado a “
11. Escribir:” Teclee longitud del lado b”
12. Escribir:” Teclee longitud del lado c”
13. Escribir:” El triángulo es equilátero” / cl
14. Escribir:” El triángulo es isósceles” / cl

15. Escribir:” El triángulo es escaleno” / cl
16. Limpiar pantalla.

6- Lógica esquemática.

```
TRIANGULO  sec
        Inicio  sec
            Abrir teclado
            Abrir display
            Limpiar pantalla
            Escribir:” Teclee longitud del lado a “
            Leer a
            Escribir:” Teclee longitud del lado b”
            Leer b
            Escribir:” Teclee longitud del lado c”
            Leer c
            C1 :=( a=b) ∧ (b=c)
            C2 :=( a=b) ∨ (b=c) ∨  (a=c)
        Inicio fin
        Comparar selec  C1
            Equilátero sec
                Escribir:” El triángulo es equilátero” / cl
            Equilátero fin
        Comparar  o   C2
            Isósceles sec
                Escribir:” El triángulo es isósceles” / cl
            Isósceles  fin
        Comparar  o
            Escaleno sec
                Escribir:” El triángulo es escaleno” / cl
            Escaleno  fin
        Comparar  final
         Final  sec
             Cerrar teclado
             Cerrar display
         Final  fin
TRIANGULO   fin
```

EJERCICIOS PROPUESTOS

1- Haga un programa que lea 3 números por el teclado. Imprimir como resultado el mayor de los 3 (siendo los 3 diferentes).

2- Elabore un programa que permita determinar a los compañeros del Registro de Población la condición de portar tarjeta del menor o carné de Identidad del personal. Si la persona es mayor de 60 años imprima que la persona pertenece a la 3^{ra} edad.

3- Haga un programa que dado dos números permita calcular el cuadrado del mayor y de ser iguales, imprimir un texto adecuado.

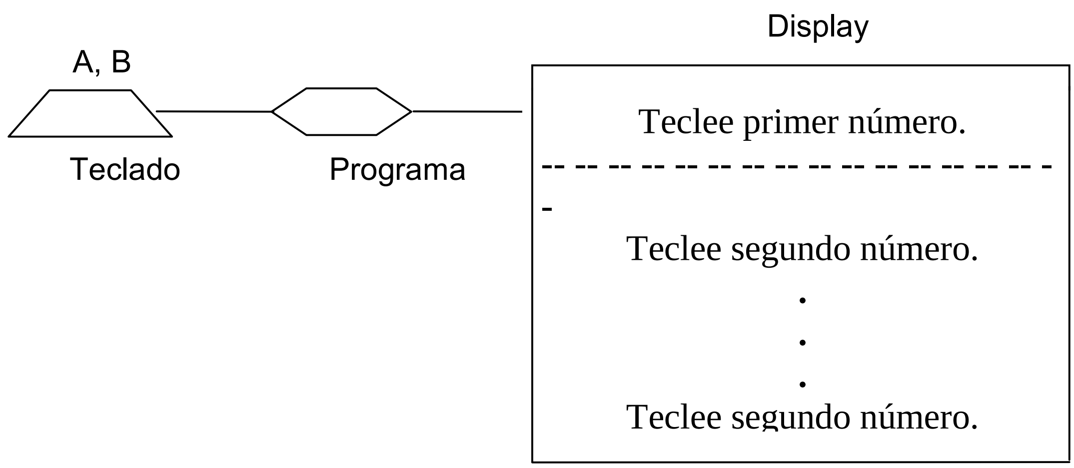
4- Elabore un programa que dado el peso en libros de una persona. Imprimir si es delgada, normal o gorda. (siendo de estatura pequeña).
Menos de 100 lb. – delgada.
Entre 100 y 160 lb. – normal.
Mayor o igual a 160 lb. – gorda.

5- Confeccione un programa que dado dos números hallar el cuadrado del mayor y de ser iguales, imprimir un texto adecuado.

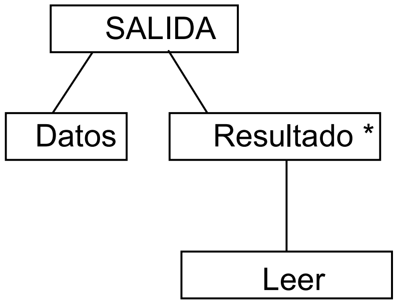
ESTRUCTURA ITERATIVA.

Ejemplo.
Hacer un programa en el cual se lean como datos dos números, imprimir como resultado el cociente de la división del primero por el segundo, si el segundo es cero repetir su lectura.

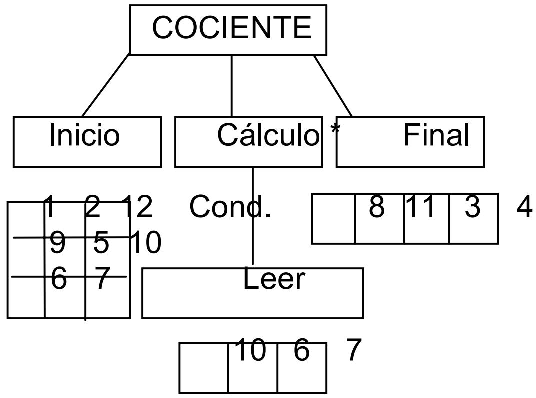
1- Diagrama General



2- Diagrama de Salida.



3- Diagrama del Programa.



4- Lista de acciones.

I: **1.** Abrir Teclado
2. Abrir display
3. Cerrar teclado
4. Cerrar display
5. Leer A
6. Leer B

II: **7.** Cond: = (B= 0)

III: **8.** F: = (A/B)

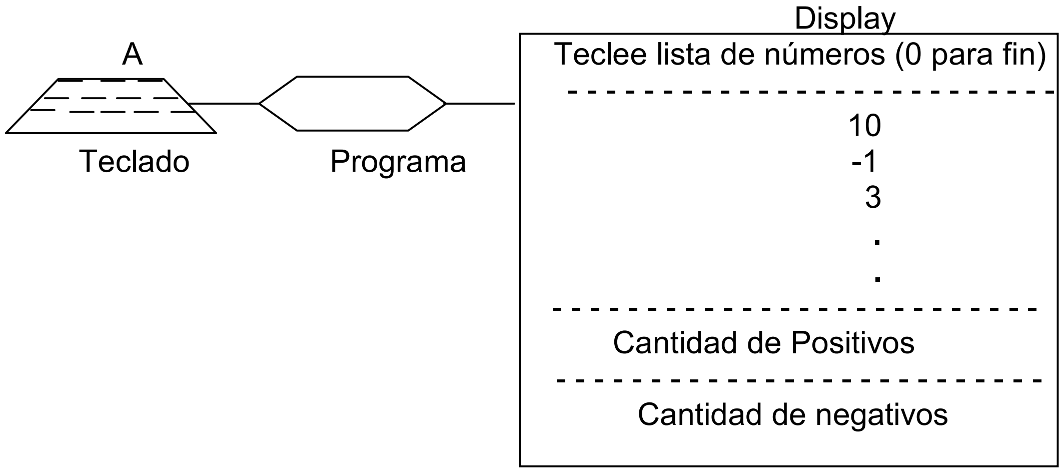
IV: **9.** Escribir: “Introduzca el primer número”.
10. Escribir: “Introduzca el Segundo número”
11. Escribir: “Cociente =”, F $/_{CL}$
12. Limpiar pantalla.

6- Lógica Esquemática.

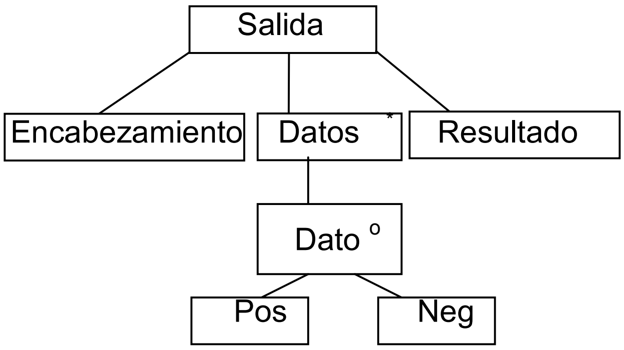
```
COCIENTE sec
        Inicio sec
           Abrir teclado
           Abrir display
          Limpiar pantalla
          Imprimir: “Introduzca el primer número”.
          Leer A
           Imprimir: “Introduzca el Segundo número”
          Leer B
           Cond: = (B = 0)
       Inicio fin
       Cálculo iter mientras Cond.
          Leer sec
              Escribir: “Introduzca el Segundo número”
               Leer B
               Cond: = (B = 0)
          Leer fin.
        Cálculo fin
       Final sec
           F: = (A/B)
            Escribir: “Cociente =”, F /CL
            Cerrar teclado.
            Cerrar display.
        Final fin
COCIENTE fin.
```

Ejemplo.
Hacer un programa en el cual se lea como dato un grupo de números, el fin de lectura viene dado porque se teclee un cero. Imprimir como resultado cantidad de positivos y de negativos.

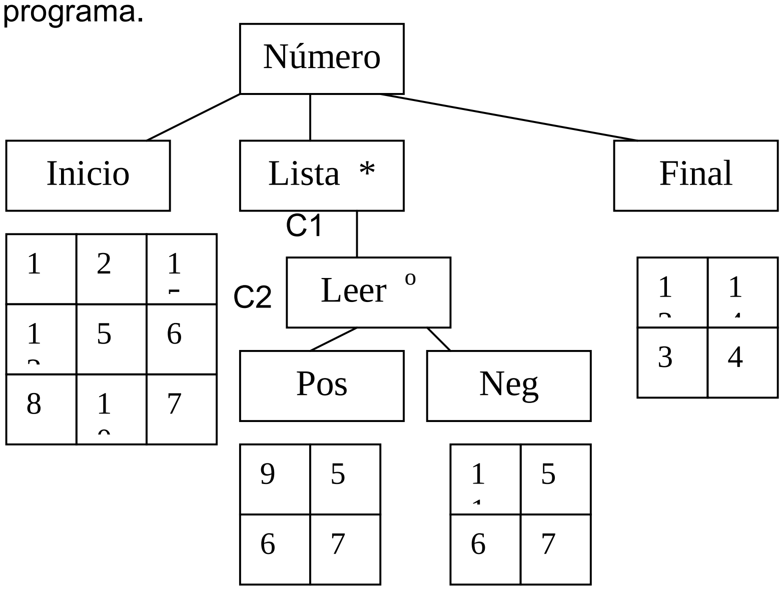
1. Diagrama del programa



2. Diagrama de Salida.



3. Diagrama del programa.



4. Lista de acciones

I- **1.** Abrir teclado.
2. Abrir display
3. Cerrar teclado.
4. Cerrar display.
5. Leer A

II- **6.** C_1 **:=** (A<> 0)
7. C_2 **:=** (A>0)

III- **8.** C_p **:=** 0
9. C_p **:=** C_p **+** 1
10. C_n **:=** 0
11. C_n **:=** C_n **+** 1
12. Escribir: “Entre lista de números (0 para fin)”.
13. Escribir: “Cantidad de positivos =”, $C_p \ /_{CL}$
14. Escribir: “Cantidad de negativos =”, $C_n \ /_{CL}$
15. Limpiar pantalla.

6. Lógica esquemática.

```
NÚMERO sec
    Inicio sec
        Abrir teclado.
        Abrir display
        Limpiar pantalla
        Escribir: “Entre lista de números (0 para fin)”
        Leer A
        C1: = (A<> 0)
        Cp: = 0
        Cn: = 0
        C2: = (A > 0)
    Inicio fin
    Lista iter mientras C1
        Leer selec C2
            Pos sec
                Cp: = Cp + 1
                Leer A
                C1: = A<> 0
                C2: = A > 0
            Pos fin
        Leer  o
            Neg sec
                Cn: = Cn + 1
                Leer A
                C1: = A<> 0
                C2: = A > 0
            Neg fin
        Leer fin
    Lista fin
    Final sec
        Imprimir: “Cantidad de positivos es  =”, Cp /CL
        Imprimir: “Cantidad de negativos es =”, Cn /CL
        Cerrar teclado
        Cerrar display
    Final fin
NÚMERO fin
```

EJERCICIOS PROPUESTOS.

1. Elabore un programa que imprima los 10 primeros números pares.
2. Elabore un programa en el que se lean como datos las notas de los alumnos de una escuela, el fin de lectura es cuando se teclee una nota negativa. Imprimir como resultado la cantidad de aprobados, cantidad de suspensos y promedio de notas de los aprobados.
3. Confeccionar un programa que dada la entrada de un número, imprima los productos del 1 al 100.
4. Hacer un programa que lea por el teclado un grupo de precios de productos de un almacén. Imprimir como resultado el precio total de los productos cuyos precios estén comprendidos entre $ 2000 y $ 5000 (incluidos ambos).el programa debe leer primero la cantidad de precios.
5. Hacer un programa que imprima por le display un listado que contenga todos los números divisibles por 3 o por 5 comprendido entre 1 y 500.

TIPO DE DATOS ESTRUCTURADOS

Hasta aquí se ha trabajado solo con variables que almacenan datos de tipo simple, y se caracterizan precisamente por guardar únicamente un dato.

Ejemplo:

Numer:= 15
Car:=”a”
Nombre:= “Odalys”

Si intentamos almacenar otro dato diferente en alguna de las variables relacionadas antes, este nuevo valor sustituye el anterior, es decir, que el dato que ella contenía se pierde. En múltiples ocasiones se necesita guardar en un mismo elemento u objeto, una agrupación de datos relacionados, como por ejemplo, los nombres de los estudiantes de un curso. Un tipo simple de dato, no me permite hacerlo (observe el ejemplo de la variable Nombre que solo acepta un nombre en cada asignación).
Sin embargo existen estructuras de datos capaces de almacenar un conjunto de valores, estos son los llamados *datos estructurados*.

.Estructura lineal
Estructura lineal de cadenas de caracteres.

K	Alumnos
1	“Ángel”
2	“Luis”
3	“María”
4	“Pedro”
.	
.	
n	

Estructura lineal de valores numéricos.

H	Promedios
1	97
2	93
3	100
4	95
.	.
.	.
.	.
m	

Para localizar un elemento de la lista debe tenerse en cuenta la posición que ocupa dicho elemento dentro de ella.

Ejemplo**:**

Alumno [4]:=’’Pedro’’

Promedio [2]:=93

Para recorrer todos los componentes incluidos en una estructura de este tipo se puede iniciar una variable simple con valor igual a 1 (que se corresponde con la primera posición) y luego incrementar su valor hasta llegar a la cantidad total de componentes. Observe estos, en el segmento del programa que se ofrece a continuación aplicado al ejemplo de la estructura alumno, donde se ha utilizado una variable simple (k) que inicialmente se le asigna el valor 1, y posteriormente se va *incrementando* su valor dentro de una iteración, en la que se mantiene, muestra dicha variable no alcance ‘’n’’, que se refiere a la posición del ultimo componente, o sea, ''n'' representa la cantidad de elementos de la estructura *Alumnos*, que se quiere reconocer.

.
.
```
K:=1
Guardar iter mientras k<= n
            Escribir: ''Tecle el nombre del alumno''
            Leer nombre
           Alumno [k]:= nombre
           K:= k+1
Guardar fin
```
Es necesario que el valor de ''n'' halla sido registrado previamente en la memoria de la máquina, bien mediante una acción de lectura o por una simple asignación pero debe ser conocido al llegar a este segmento de programa para que pueda evaluarse la expresión k<= n.

Estructura bidimensional

Ejemplo de estructura bidimensional de valores numéricos.

	L = 1	2	3	4	5
K = 1	97	100	100	92	95
2	91	88	96	85	93
3	100	98	94	99	95
4	90	100	92	100	96
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Para recorrer este tipo este tipo de estructura se necesita del auxilio de dos variables simples, una se usara para borrar las filas (k) y la otra para la columna (l).
La referencia a un componente se hace indicando la posición de la misma dentro de la estructura: Notas [k, l]

Ejemplo.

Notas [2,4]:= 85

Los tipos de estructuras estudiados se conocen por *arreglos.* Los arreglos exigen que sus componentes sean del mismo tipo de dato.

- *Registros*

En numerosas ocasiones surge la necesidad de guardar en un solo objeto, datos de tipos diferentes pero relativo a una misma entidad y por lo tanto, relacionados entre si, para lo cual no pueden utilizarse los arreglos, pero existen otros tipos de datos estructurados que si ofrecen esta posibilidad.

Ellos son los *Registros*.
Un registro es una estructura capaz de agrupar un conjunto de datos de diferentes tipos.

Ejemplo
Nombre del trabajador
Ocupación
Salario
Edad

Las diferentes partes que componen un registro se denominan **campos**.
El registro es un tipo de datos muy útil, ya que puede contener a otros tipos de datos en sus campos. Es de destacar que una vez declarado los tipos de datos que estarán en cada campo estos serán invariables.

Los registros a su vez pueden formar parte de otra estructura, de tal manera que podemos tener una distribución de datos cuyos componentes sean Registros y que los campos de este Registro pueda definirse por nosotros,

Permitiéndonos utilizar diferentes tipos de datos en una misma estructura, así entonces, podremos definir un **arreglo** formado por elementos del tipo **registro**.

Ejemplo del Arreglo Trabajadores cuyos componentes son Registros.

Nombre del trabajador
Ocupación
Salario
Edad
Estudia

Ángel Martínez
Ingeniero
375.00
57
V

Pilar Díaz
Profesora

505.00
55
F

La referencia a un campo de un registro se hace escribiendo el nombre de la variable del registro segundo de un punto y a continuación el nombre del campo.

<Variable Registro>.<campo>

En este ejemplo:
Trabajadores [1].nombre:= Ángel Martínez
Trabajadores [2].ocupación:= Profesora

- ***Ficheros***

Dentro de la computadora, la información se almacena en el disco duro (de forma permanente) o en la memoria (de forma temporal para realizar las operaciones). Pero esta información debe estar organizada de alguna forma, ya que sino nos resultaría imposible acceder a ella (se unirían textos con imágenes, con música, con números y como todo al final son ceros (0) y unos (1), no se podría saber donde empieza y terminan las cosas).
Para evitar esto la información se guarda en una estructura llamada archivo o fichero. Así, una imagen seria archivo, una carta escrita con un procesador de texto, seria otro archivo distinto, una tabla generada con un tabulador electrónico seria otro tercer archivo, y una agrupación de datos organizada podría ser el cuarto archivo y así sucesivamente.
Los archivos tienen un nombre y una extensión para diferenciarlos uno de otros, la extensión nos indica que tipo de archivo son (DOC. TXT, texto, TIF, BMP, o PCX, imágenes, WAN, sonido, DBF, base de datos…)

Los componentes de los archivos de Base de Datos son registros. Los archivos cuyos componentes son registros, sus campos no pueden ser accedidos por separado, sino a través del Registro al que pertenecen ya que en este caso el registro es la menor cantidad de información que se puede transferir en una operación de lectura o escritura a los ficheros.

Operaciones de entrada /salida (E/S) con ficheros.
Abrir <nombre archivo>
Acción que permite inicializar un archivo y dejar todas las condiciones preparadas para poder realizar operaciones de lectura o escritura sobre el.
Ej. Abrir trabajadores.

.Cerrar <nombre archivo>

Ej. cerrar trabajadores

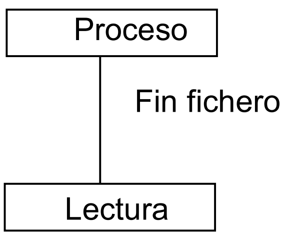
Acción que permite finalizar la sesión de trabajo con el archivo y dejar todas las condiciones preparadas para operaciones futuras con el mismo.
.Leer <nombre archivo. Nombre variable>
Ej. Leer trabajadores, T.
Acción que transfiere desde el archivo, el componente apuntado, hacia la memoria .El puntero del archivo se mueve automáticamente hasta la próxima componente.

.Escribir <nombre- archivo, nombre – variable >

Ej. Escribir trabajadores,'
Acción que trasfiere el contenido de la variable de memoria hacia el archivo, en la posición donde se encuentre el puntero del archivo. Este se mueve automáticamente hasta la próxima componente.

Funciones
Fichero: Función lógica que retoma verdadero si el puntero del fichero esta al final del mismo y falso en cualquier otro caso.

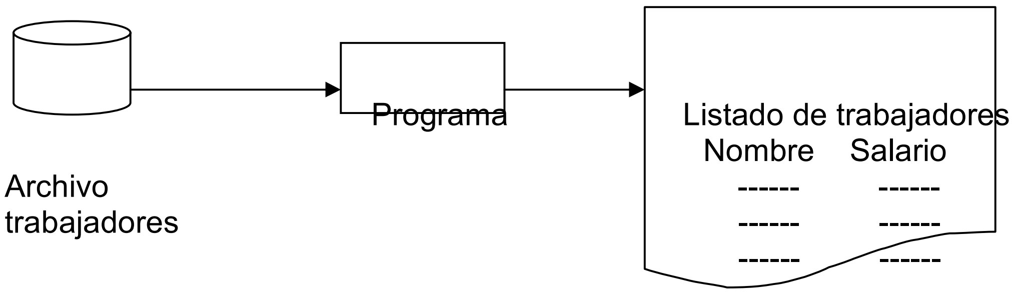
Ejemplo del proceso que representa la lectura secuencial de un archivo hasta que se encuentre su fin.



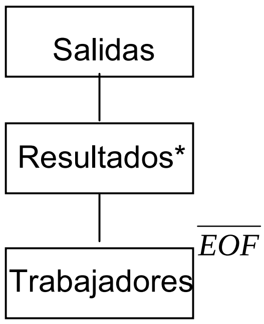
Programa que lee un archivo de trabajadores sobre disco, y emite a través del impresor un listado con el nombre y salario de todos los trabajadores. La estructura de las componentes del archivo es la siguiente:
Nombre del trabajador
Ocupación
Salario
Edad

Ejemplo

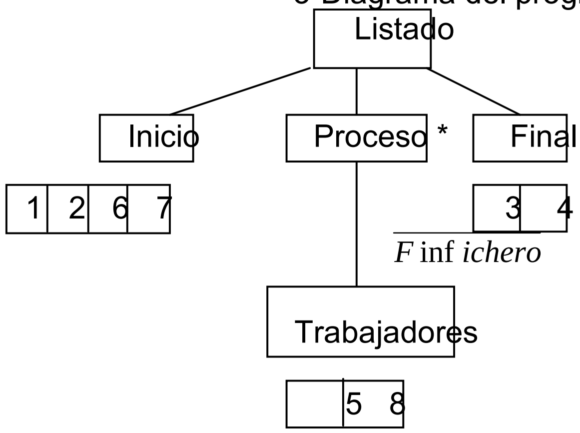
1-Diagrama general



2-Diagrama de salida



3-Diagrama del programa



4-Lista de acciones

1-Abrir trabajadores
2-Abrir P.R.N
3-Cerrar trabajadores
4-Cerrar P.R.N
5-Leer trabajadores , T
6-Encender P.R.N, ‘’Listado de trabajadores’’/cl
7-Escribir P.R.N, ‘’Nombre   Salario’’ /cl
8-Escribir P.R.N, T.   Nombre, ‘’   ’’, T. Salario /cl

Lógica esquemática

```
Listado    6- sec
              Inicio sec
                    Abrir trabajadores
                    Abrir PRN
                    Escribir PRN. ''Listado de trabajadores'' /cl
                    Escribir PRN. ''Nombre        Salario'' /cl
             Inicio fin
             Proceso iter mientras no finfichero
                       Trabajadores sec
                                      Leer Trabajadores, T
                                      Escribir PRN, T.Nombre."      ", T.Salario   /cl
                       Trabajadores Fin
             Proceso Fin
             Final sec
                       Cerrar Trabajadores
                       Cerrar PRN
             Final Fin
Listado Fin
```

Ejercicios propuestos

1- Confeccione un programa que cree el archivo de productos (Almacén) sobre discos, cuyos componentes deben tener la siguiente estructura.

CODIGO
DESCRIPCION
PRECIO UITARIO
CANTIDAD
CLAVE

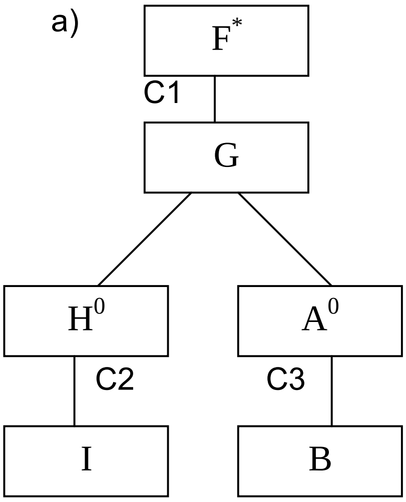
Los datos necesarios para crear dicho archivo, serán leídos desde teclado. La CLAVE puede ser solo un carácter "I" o "N".
Donde "I" indica que el producto es de importación.
"N" indica que el producto es de producción nacional.

2- Elabore un programa que lea el archivo de productos creados en el ejercicio anterior y emita a través del impresor un listado con los productos de importación, y cuyo precio unitario esta por encima de los $5.00.Al final del listado, debe imprimir el porciento que estos producto s representan el total.

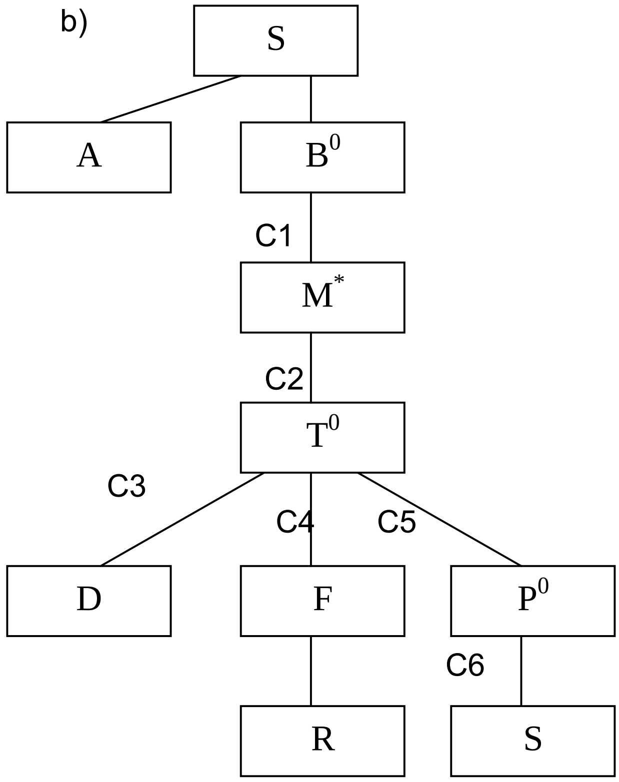
EJERCICIOS DEL CAPITULO.

1- Dado los siguientes diagramas de estructura, haga la lógica esquemática.

a)



b)



2- Haga un programa que permita determinar el voltaje con que trabaja un equipo que consume en la red (I) amp., teniendo una resistencia (R) determina si : V= I * R.

3- Para las elecciones del Poder Popular se necesita que usted elabore un programa que permita conocida la edad de un cederista de la circunscripción determinar si es o no elector.

4- Haga un programa que permita evaluar la función que

$$g(x) = \begin{cases} 2x + 3 & \text{si} \quad x > 5 \\ x^2 & \text{si} \quad x \leq 5 \end{cases}$$

5- Se proyectara una película apta para mayores. Haga un programa que imprima si una persona puede o no entrar al cine, si se conoce su edad.
6- Elabore un programa que lea como dato un Password, comparar dicha palabra clave con una constante de caracteres que se encuentra dentro del programa que contiene el verdadero Password, si coincide imprimir un mensaje que diga “Bienvenido” de lo contrario imprimir un mensaje que diga “acceso denegado”.

7- Para la Evaluación de la FEEM los grupos se clasifican de acuerdo a los puntos obtenidos.
Vanguardia, mayor o igual a 90 puntos
Destacado, entre 50 y 90 puntos
Rezagado, menor de 50 puntos
Confeccione un programa que dado los puntos obtenidos por un grupo. Imprima su clasificación.
8- Haga un programa que pueda clasificar a los estudiantes en las diferentes organizaciones estudiantiles conociendo que:
Pionero, menor de 15 años
FEEM, entre 15 y 18 años
FEU, mayor o igual de 18 años

9- Diseñe un programa que pueda calcular el impuesto a pagar por las empresas a partir de sus ingresos conociendo que:

Ingresos	*Impuestos*
<= $ 2000	No paga impuestos
Entre $2000 y $5000	2% de la cantidad excedida a $2000
> $5000	$60 + 5% de la cantidad excedida a $ 5000

10- Elabore un programa que imprima todos los números pares comprendidos ente 1 y 100.

11- Haga un programa que lea como dato la cantidad de alumnos de una escuela, a continuación leer las de aprobados, la cantidad de sobrevivientes (90 o mas puntos) y promedio de los sobrevivientes.

12- Confeccione un programa en que lea un número por el teclado. Imprimir como resultado el factorial de dicho numero.

13-) Desde una terminal salieron N cantidad de ómnibus, conociéndose la cantidad de pasajeros que viajan en cada uno.
Haga el programa que permita determinar la cantidad total de pasajeros que viajaban en el ómnibus y cuantos ómnibus salieron con menos de 40 pasajeros.

14-) Dada la lectura con el espacio “x” que ocupa en disco cada una de las carpetas. Elabore un programa que posibilite: calcular e imprimir el total de información contenida entre todos los sectores, así como determinar la cantidad de carpetas que se pueden guardar en el disquete que solo le queda 45kb de memoria libre. Si la ultima carpeta esta vacía.

15-) Hacer un programa en el cual se lea como dato la cantidad de atletas que compiten en los 100m planos masculinos. Leer el tiempo realizado por cada atleta. Imprimir como resultado la cantidad de atletas cuyas marcas realizadas estén entre 9.7seg y 10.3seg y el total de tiempo realizado por los atletas en menos de 9.7seg o mas de 10.3seg, así como el promedio de este último.

BIBLIOGRAFÍA

Apelt, Harry. *Introducción a la Lógica Matemática*. Editorial Pueblo y Educación, 1979.

Bello Corvo, Higinio. *Un trabajo con Algoritmo.* Revista Giga 3/2001.

Díaz Llorca, Carlos. *Introducción a la Computación.* La Habana. Editorial Pueblo y Educación, 1980.

Johnsonbaugh, Richard. *Matemáticas Discretas.* Editorial Pearson.

Lipschutz Seymour. *Teoría de Conjuntos y temas afines.* Editorial Pueblo y Educación 1975.

List, G. *Lógica Matemática, Teoría de conjunto y Dominios numéricos*. Editorial Pueblo y Educación 1978.