

Algoritmos Primeros Pasos

CAPITULO 1. FUNDAMENTOS DE PROGRAMACIÓN

- 1.1 Concepto de Algoritmo
- 1.2 Partes de los Algoritmos
- 1.3 Elementos fundamentales de los Algoritmos
- 1.4 Literales
- 1.5 Datos
- 1.6 Variables
- 1.7 Identificadores
- 1.8 Palabras reservadas
- 1.9 Expresiones
- 1.10 Metodología para la construcción de algoritmos
- 1.11 Herramienta de Pseudo-código
- 1.12 Comentarios

CAPITULO 2. PRIMITIVAS DE PSEUDO-CÓDIGO

- 2.1 Primitiva de entrada
- 2.2 Primitiva de salida
- 2.3 Algoritmos secuenciales
- 2.4 Algoritmos no secuenciales
- 2.5 Primitivas de selección
- 2.6 Primitivas de repetición

CAPITULO 3. PROGRAMACIÓN ESTRUCTURADA

- 3.1 Método TOP DOWN
- 3.2 Funciones
- 3.3 Procedimientos
- 3.4 Librerías

CAPITULO 4. INTRODUCCIÓN A LAS ESTRUCTURAS DE DATOS

- 4.1 Estructura registros
- 4.2 Estructura arreglos
- 4.3 Métodos de ordenación
- 4.4 Métodos de búsqueda

Capítulo 1. FUNDAMENTOS DE PROGRAMACIÓN

En este capítulo se establecerán las bases para la construcción de algoritmos brindando lo básico y esencial que un

Concepto de Algoritmo programador principiante debe conocer para elaborar dichos algoritmos de manera metodológica, además proporcionará una herramienta estándar para su escritura.

1.1 *Concepto de Algoritmo*

Hace años cuando comencé a programar computadoras en el lenguaje BASIC, intuitivamente y sin alguna referencia, casi se puede decir que naturalmente, hacia programas que comenzaban y tenían un final, básicamente eran programas para cálculo de ecuaciones elementales para física, química y matemáticas y que le introducía como insumos valores numéricos de entrada y el programa a través de formulas arrojaba resultados.

Más tarde después de estructurarme en la universidad comprendí que existía el concepto de algoritmo y lo asocie a recetas culinarias, si usted toma la receta para fabricar una torta y tiene los ingredientes a la mano y sigue las instrucciones y conoce el objetivo final muy seguramente obtendrá el mismo producto siempre que se den los mismos requisitos, a estos algoritmos se les conoce como “algoritmos no computacionales”. El concepto de algoritmo es el de una receta o un conjunto de pasos para hacer una labor o tarea. Cuando hablamos no de pasos sino de instrucciones o primitivas para resolver un problema estamos refiriéndonos a “algoritmos computacionales”; estos últimos son convertidos en código fuente en cualquier lenguaje de programación (PASCAL, BASIC, C, JAVA, y muchos mas) para ser ejecutados en el computador, aquí dejan de ser algoritmos y se convierten en programas.

Formalmente encontramos diversas definiciones de algoritmo una de ellas “procedimiento paso a paso para solucionar un problema dado”¹, otra bastante buena es “grupo limitado de pasos expresados claramente y sin ambigüedad que, de acuerdo a un grupo de condiciones iniciales, al ejecutarse produce unos resultados en un tiempo limitado”². una definición personal “Conjunto ordenado, finito, preciso y definido de pasos para resolver un problema”; ordenado porque los pasos deben obedecer a unas reglas y también porque el orden de los pasos influye en el resultado final, finito porque el algoritmo debe tener un principio y un fin, preciso porque se deben evitar las inconsistencias y ambigüedades en un algoritmo y definido porque el algoritmo cada vez que se recorra producirá los mismos resultados con las mismas condiciones como el la receta de la torta (Ejemplo 1).

¹ JOYANES Aguilar, Luis, Problemas de la metodología de la Programación. Pp. 1

² BECERRA Santamaría, Cesar, Algoritmos: Conceptos Básicos. Pp. 5

Ejemplo 1. Algoritmo no Computacional

Nombre: Torta de manzanas

Objetivo: Elaboración de una Torta de manzanas

Análisis del problema: Dados los ingredientes (huevos, harinas, manzanas, leche y polvo para hornear), mezclarlos adecuadamente y en la proporción justa, moldearlo y hornearlo para obtener una torta de manzanas

Pasos.

1. Inicio
2. Mezclar la harina con una taza de leche y batirla
3. Batir los huevos en otro recipiente
4. Agregar los huevos batidos a la harina batida en el paso 2.
5. Pelar las manzanas en tajadas
6. Agregar las manzanas a la mezcla del paso 4.
7. Agregar proporcionalmente una cucharadita de polvo de hornear por cada libra de harina
8. Verter en un molde que tiene mantequilla
9. Colocar en el horno a una temperatura de 250 °C durante 90 minutos
10. Sacar la torta del horno
11. Fin

Podemos sacar varias conclusiones del ejemplo anterior: Existe ambigüedad en mucho de los pasos por ejemplo en el paso 7. una cucharadita que es para una persona y que es para otra, en el paso 9. si a los 90 minutos no esta lista la torta cuantos minutos mas tengo que darle, ¿será que la altura a nivel del mar de la ciudad donde se prepare influirá en algo?, ahora pelar las manzanas es un algoritmo en sí. Es casi imposible evitar imprecisiones.

Los algoritmos computacionales no tienen esos problemas como lo veremos a continuación:

Ejemplo 2. Algoritmo Computacional

Nombre: Sumar dos números

Objetivo: Construir un numero que sume dos números dados

Análisis del problema: El usuario digita los números por el teclado, luego mediante la operación matemática + el computador los suma y por último se muestra el resultado al usuario

Pasos.

1. Inicio
2. Leer Numero1
3. Leer Numero2
4. Asignarle a resultado Numero1 + Numero2
5. Escribir Resultado
6. Fin

No hay ambigüedad, ni imprecisión, hay un orden lógico y cada vez que recorra el algoritmo se obtendrá el mismo resultado.

1.2 Partes de los Algoritmos

Todo algoritmo puede ser descompuesto en tres partes:

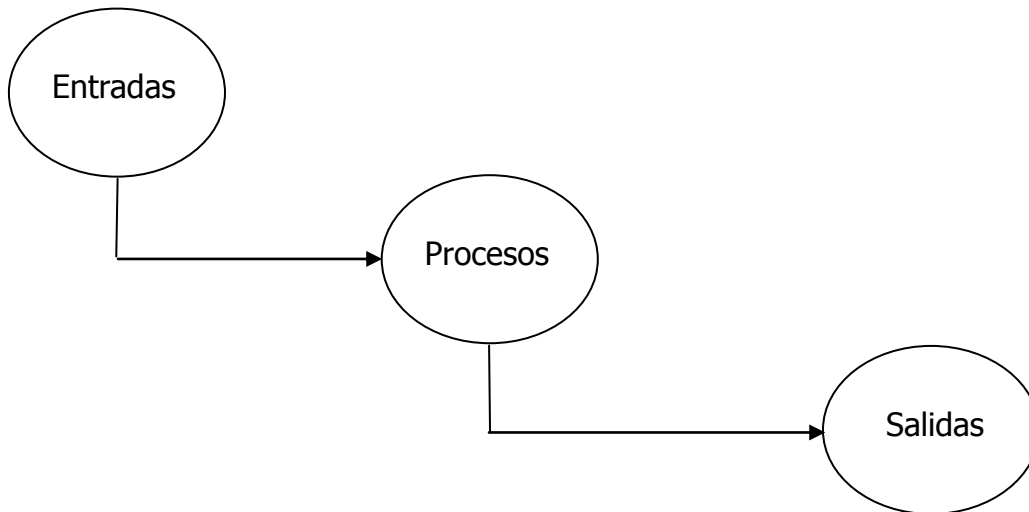


Figura 1. Partes de los algoritmos

Las entradas son todos los insumos o información de entrada; en los algoritmos computacionales las entradas son los datos suministrado al algoritmo mediante los dispositivos de entrada (Ej. teclado, mouse, lector óptico, etc.), en los no computacionales como el de la torta serían los ingredientes de ésta.

Los procesos se refieren a la transformación de ese resultado de manera que se puedan alcanzar los objetivos del algoritmo en el caso de la torta sería de que manera mezclar los ingredientes y en los algoritmos computacionales que orden deben tener las instrucciones y/o expresiones para buscar el objetivo del algoritmo.

Las salidas son los resultados obtenidos por esa transformación y entregados al usuario a través de los dispositivos de salida (Ej. Monitor, impresora), en los algoritmos no computacionales equivaldría a la torta horneada.

En el ejemplo 2. se observa claramente que las estradas están conformadas por los pasos 2 y 3, los procesos por el paso 4 y la salida por el paso 5.

1.3 Elementos Fundamentales de los Algoritmos

Los algoritmos en general cuentan con unos elementos fundamentales que se relacionan entre sí, la manera en que estos se ordenen y/o se dispongan

cumplirán con el objetivo o el propósito de dicho algoritmo. Básicamente los elementos fundamentales son: Datos, variables, expresiones y primitivas. Todos estos elementos son comunes a los paradigmas de programación y a los lenguajes de programación y por supuesto a la metodología de pseudo-código contemplada en este libro.

Teniendo en cuenta lo anterior los elementos en mención son los cimientos y las bases para aprender la lógica de programación, además de ser necesarios para convertirse en un programador experto. Es entonces casi obligatorio dominar muy bien dichos elementos y el papel que juegan en la construcción de algoritmos, a continuación se observa de forma gráfica la importancia de los elementos fundamentales de los algoritmos:

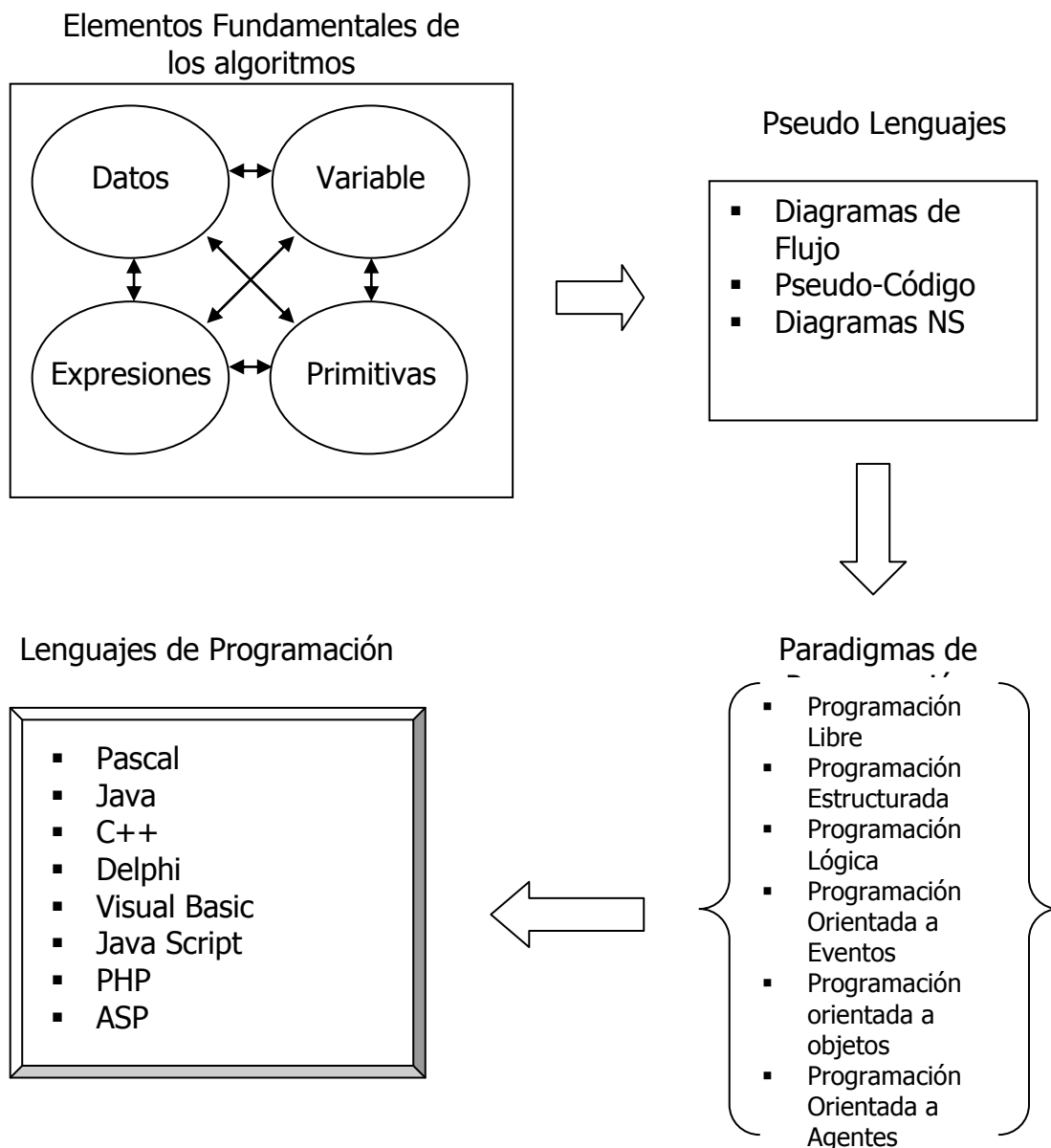


Figura 2. Elementos fundamentales de los algoritmos

Una vez se ha logrado utilizar bien estos elementos, la sintaxis para escribir los algoritmos es brindada por los pseudo lenguajes que son una herramienta fundamental para simular programas en cualquiera de los paradigmas o estilos de programación y estos a su vez son utilizados por la mayoría de los lenguajes de programación. En última instancia los elementos fundamentales de los algoritmos son usados por todos los lenguajes de programación

1.4 Literales

Se denominan literales a todos aquellos valores que se pueden almacenar, procesar y/o transportar a través de un dispositivo computacional y pueden ser:

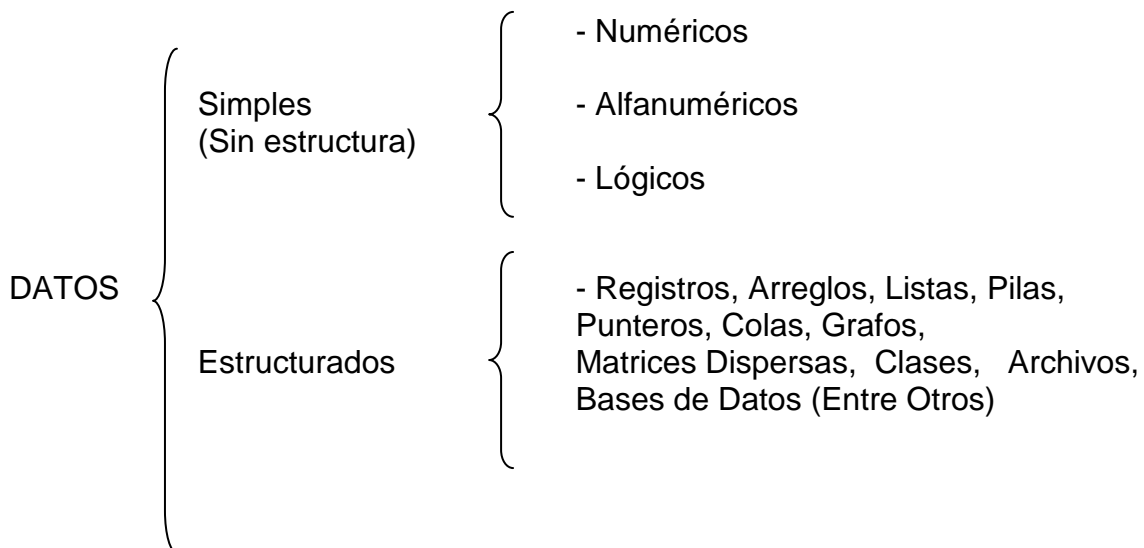
Tipo de Literal	Valor que toma
Literales enteros	22, 0, -17, etc.
Literales reales	3.1416, -5582.11, 2.6E+5 etc.
Literales de carácter	'a', 'B', '@;', 'N', '&', etc.
Literales de cadena	"UNICORDOBA", "Algoritmos Estructurados", etc.
Literales lógicos	verdadero, falso

Tabla 1. Tipos de literales

Los literales pertenecen a un tipo de dato en especial, por ejemplo los literales enteros tienen su dominio en los números enteros.

1.5 Tipos de Datos

Los dispositivos computacionales manejan unidades mínimas de información llamadas datos, estos se clasifican de la siguiente manera:



Definidos por el usuario

Figura 3. Clasificación de los Datos

Los datos simples o sin estructura son utilizados por todos los lenguajes de programación, inicialmente en los albores de la computación eran los únicos que existían, pero a medida que los problemas se fueron haciendo mas complejos aparecieron otros tipos de datos.

Los datos simples numéricos pueden ser enteros o reales tomando literales enteros o reales respectivamente, los datos alfanuméricos por su parte toman literales carácter o de tipo cadena y los lógicos también conocidos como boléanos toman literales lógicos que viene a ser un estado (verdadero o falso) más no un valor.

Los datos estructurados son la combinación de datos simples y se utilizan para resolver problemas complejos de almacenamiento, procesamiento y transporte de información.

Por último los datos definidos por el usuario son todos aquellos datos nuevos que no existen como por ejemplo el tipo color ("Amarillo", "Verde", "Azul", "Rojo"), a partir de su definición podrá usarse para determinar datos de tipo color.

Ejemplo 3. Tipos de datos simples

Tipo de Dato	Ejemplos
Enteros	60 0 -5 8 -30000
Reales	8.56 3.1416 -5.68 2.5e+5
Caracteres	'A' '1' '@' 'N' '\$'
Cadenas	"COLOMBIA" "UNICORDOBA"

Lógicos	Verdadero Falso
---------	--------------------

Tabla 2. Tipos de datos

En el ejemplo anterior se muestran los tipos de datos simples y algunos valores que podrían tomar, cabe resaltar que el tipo de dato real puede tomar también el valor en notación científica 2.5e+5 (significa 2.5 x 10 a la 5).

1.6 Variables

El concepto de variable esta muy asociado al concepto de dato, porque la vía por la cual los datos serán almacenados temporalmente en un sistema de computo son las variables. Una variable tiene varias características:

- 1) Espacio en memoria, como consecuencia de esto una variable es reservada por el administrador de memoria del sistema operativo, lo que quiere decir es que una vez reservada otra variable no podrá ocupar el espacio de ella.
- 2) Nombre, el cual es un identificador y por ende debe obedecer a las reglas de los identificadores que veremos en el ítem 1.6.
- 3) Un tipo de datos, por lo tanto si una variable es numérica entera solo podrá almacenar datos numéricos enteros no de otro tipo (en los lenguajes modernos existe el casting³).

Las variables se declararán según la metodología de este documento con la palabra reservada (ver ítem 1.7) **Variables**.

Ejemplo 4. Declaración de Variables

Variables x:entero

quiere decir que estamos declarando una variable x de tipo entero

Identificadores

Los identificadores son las palabra con que nombramos variable, funciones, procedimientos, módulos y programas entre otros. En el ejemplo 3 “x” corresponde al identificador.

Los identificadores tienen las siguientes reglas:

- 1) Deben comenzar por una letra
- 2) No deben tener espacios en blanco

³ “Casting”: Es un recurso que manejan ciertos lenguajes de programación, cuando una variable se asigna otra de un tipo distinto la primera transforma al tipo de la última.

- 3) No deben contener caracteres extraños o especiales por ejemplo +, -, *, #, %, @, Ñ etc, excepto el subrayado _
- 4) No deben ser palabras reservadas (ver ítem 1.7)
- 5) Preferiblemente, el nombre debe ser nemotécnico, es decir; el nombre que se escoja para declarar una variable debe ser coherente con el significado de dicha variable.

1.7 Palabras Reservadas

Existen palabras propias de los lenguajes de programación que no pueden ser utilizadas como variables debido a que se confundiría el compilador, estas palabras se conocen como palabras reservadas y que para efecto de este libro cuando se defina la sintaxis de la herramienta Pseudo-código serán escritas en negrita.

Hasta el momento definiremos las siguientes palabras que serán utilizadas como reservadas de aquí en adelante:

PALABRA	SIGNIFICADO
Entero	Servirá para declarar datos de tipo entero
Real	Se usará para declarar datos de tipo real
Carácter	Será utilizada para declarar datos de tipo carácter
Cadena	Será usada para declarar datos de tipo cadena
Lógico	Se utilizará para declarar datos de tipo lógico
Variables	Se empleará para la declaración de variables

Tabla 3. Palabras reservadas

1.8 Expresiones

Las expresiones son uno de los elementos fundamentales en la programación y son utilizadas para alcanzar el objetivo del algoritmo, estas pueden validar un valor devolviendo un literal numérico o alfanumérico, también pueden devolver un estado falso o verdadero. A continuación estudiaremos los tipos de expresiones.

1.8.1 Expresiones aritméticas

Las expresiones aritméticas se construyen utilizando literales numéricos ya sean enteros o reales con los operadores aritméticos(+,-,*,/). Los operadores aritméticos

según el lenguaje BASIC⁴ y que es común a varios de los lenguajes de programación son los siguientes:

Nombre del Operador	Símbolo	Capítulo 2. Jerarquía
Paréntesis	()	<div style="text-align: center;"> Mayor ↑ ↓ Menor </div>
Multiplicación, División	*, /	
Suma, Resta	+,-	
Cociente, Residuo	Div, Mod	

Tabla 4. Jerarquía de los operadores aritméticos según BASIC

Ejemplo 5. Jerarquía en expresión aritmética

$$5 * 6 + 4$$

La jerarquía de los operadores significa que primero ejecuta la operación que tenga mas jerarquía, por lo tanto pareciera que la expresión anterior tuviera dos respuestas $(30+4) = 34$ y $(5 * 10) = 50$ pero en realidad la jerarquía indica que la multiplicación tiene mayor jerarquía que la suma, entonces primero se hace la operación multiplicación. En vista de esto la respuesta correcta es 34.

Ejemplo 6. Utilización del Mod y el Div

$$\begin{array}{r} 10 \overline{) 3} \\ 1 \\ \hline 3 \end{array}$$

La división entre el dividendo 10 y el divisor 10 tiene 3 respuestas:

- La primera respuesta es la división absoluta: $10/3 = 3.333$
- Otra respuesta es utilizando en operador Div (cociente): $10 \text{ Div } 3 = 3$
- Y la última es utilizando el operador Mod (residuo): $10 \text{ Mod } 3 = 1$

1.8.2 Expresiones relacionales o de relación

Las expresiones relacionales se construyen al comparar dos expresiones aritméticas o estas con literales numéricos, al contrario de las expresiones aritméticas no devuelven un valor sino un estado verdadero o falso, son utilizadas mas adelante en las primitivas de selección y las de repetición.

Los operadores relacionales tienen igual jerarquía y son los siguientes:

⁴ Lenguaje BASIC: lenguaje de programación que viene el la memoria ROM de los computadores con tecnología INTEL

Nombre del operador	Símbolo
Mayor que	>
Menor que	<
Igual	=
Diferente	<>
Mayor o Igual	>=
Menor o Igual	<=

Tabla 5. Operadores relacionales

Ejemplo 7. Uso de expresiones relacionales comparando expresiones aritméticas

$$3 + 4 <= 50 \text{ div } 3$$

Como se puede observar del lado izquierdo del operador hay una expresión aritmética al igual que del lado derecho y la respuesta es la siguiente: $7 <= 16$ **verdadero**.

Ejemplo 8. Comparación de un literal numérico con una expresión aritmética

$$8 + 6 - 3 > 7$$

En este ejemplo se compara una expresión aritmética del lado izquierdo con un literal numérico del lado derecho y la respuesta es $11 > 7$ y la respuesta es **falso**.

1.8.3 Expresiones Lógicas

Estas expresiones comparan estrictamente dos expresiones relacionales a través de los operadores lógicos (**And**, **Or** y **Not**) aunque existen otros operadores lógicos a partir de estos tres se pueden alcanzar usando estos tres, al igual que los relacionales estos devuelven un estado falso o verdadero determinado por las tablas de la verdad.

- Operador **And**: este operador lógico se basa en la tabla de la verdad descrita a continuación :

Expresión 1	Expresión 2	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Falso

Falso	Verdadero	Falso
Falso	Falso	Falso

Tabla 6. Tabla del operador And

Es fácil de aprender, en el operador **And** el resultado de la comparación es verdadero cuando ambas expresiones a comparar son verdaderas

Ejemplo 9. Expresión lógica con operador And

$$8 + 5 > 6 \text{ And } 4 <> 2+3$$

Para conocer el resultado de la siguiente expresión debemos resolver primero la expresión del lado izquierdo del operador $13 > 6$, la respuesta es verdadero, ahora resolvemos la expresión del lado derecho del algoritmo $4 <> 5$, la respuesta es verdadero, nos remitimos a la tabla del operador **And**, la cual indica que si la expresión 1 es verdadero y la expresión 2 es verdadero toda la expresión es verdadero, por lo tanto la respuesta es **verdadero**.

Ejemplo 10. Expresión lógica con operador And

$$5 + 5 > 4 \text{ And } 4 <> 2+3$$

Resolvemos igualmente la expresión del lado izquierdo del operador $10 > 4$, la respuesta es falso, quiere decir que con una de las expresiones relacionales que sea falso la expresión lógica es falso y la respuesta por ende es **falso**.

- Operador **Or**: La tabla de este operador es la siguiente:

Expresión 1	Expresión 2	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Verdadero
Falso	Verdadero	Verdadero
Falso	Falso	Falso

Tabla 7. Tabla Operador Or

Es fácil de aprender, en el operador **Or** el resultado de la comparación es falso cuando ambas expresiones a comparar son falsas

Ejemplo 11. Expresión lógica con operador Or

$$3 * 2 \leq 6 \text{ Or } 4 + (6 * 8) \text{ Mod } 3$$

Examinemos este ejemplo calculando la expresión de la izquierda del operador **Or** $6 \leq 6$, la respuesta es verdadero, si nos remitimos a la tabla de la verdad para cualquier estado de la expresión a la derecha el resultado final de la expresión completa es **verdadero**.

Ejemplo 12. Expresión lógica con operador **Or**

$$8 * 8 + 3 < 5 + 2 \text{ Or } 3 + 6 * 5 = 33$$

Si examinamos la respuesta de la expresión relacional a la izquierda $67 < 7$, es falso, esto nos obliga a examinar la respuesta del lado derecho para poder sacar la respuesta de la expresión lógica $33 = 33$, es verdadero por lo tanto remitiéndonos a la tabla del operador **Or** la respuesta es **verdadero**.

- Operador **Not**: el operador compara una sola expresión relacional como se muestra a continuación:

Expresión	Resultado
Verdadero	Falso
Falso	Verdadero

Tabla 8. Tabla operador Not

El operador **Not**, cambia el estado de la expresión que está afectando de falso a verdadero y viceversa

Ejemplo 12. Expresión lógica con operador Not

$$\text{Not } (24 + 5 < 7 + 3)$$

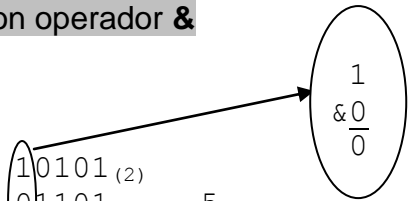
Primero debemos conocer el estado de la expresión relacional que es influida por el operador Not ($29 < 10$), la respuesta es falso, aplicando la tabla de dicho operador cambia el valor de la expresión, por ello la respuesta de la expresión lógica es **verdadero**.

1.8.4 Expresiones de bits

La mayoría de lenguajes de programación pueden manipular información a nivel de bits empleando operadores de bits que para no confundirlos con los operadores lógicos los vamos a llamar (**&**, **||** y **!**) que simbolizan (**And**, **Or** y **Not**) respectivamente y se comportan igual que los operadores lógicos y se utilizan con las tablas de la verdad de estos operadores para pero a nivel de bits. Consideremos unos ejemplos y miremos su utilización.

Ejemplo 13. Expresión de bits con operador &

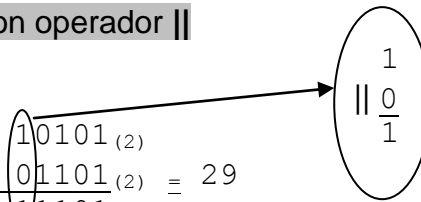
21 & 13 equivale a:

$$\begin{array}{r} 10101_{(2)} \\ \& 01101_{(2)} \\ \hline 00101 \end{array} = 5$$


Antes de hacer la operación se cambia a base 2 el entero 21 y esto nos da (10101) luego se convierte a base 2 el entero 13 y esto equivale a (01101), luego se compara bit a bit como lo podemos observar en el área oval, utilizando la tabla de verdad del operador lógico **And** 1 & 0 es igual a 0, aplicando dicha tabla columna a columna nos da en binario el resultado de 00101 que transformado a decimal corresponde a 5.

Ejemplo 14. Expresión de bits con operador ||

21 || 13 equivale a:

$$\begin{array}{r} 10101_{(2)} \\ || 01101_{(2)} \\ \hline 11101 \end{array} = 29$$


De igual manera se transforma 21 a binario (10101) y 13 (01101) luego mediante la tabla de verdad de el operador lógico **Or** se comparan los bits, como se puede observar en el área ovoide 1 || 0 nos da como resultado 1; así sucesivamente vamos tomando columna por columna y el resultado es 11101₍₂₎ que equivale en base 10 a 29.

Ejemplo 15. Expresión de bits con operador !

$$\begin{array}{r} 10101_{(2)} \\ || 01101_{(2)} \\ \hline 11101 \\ \hline !(11101) \\ \hline 00010 \end{array} = 4$$

Se había concluido en el ejemplo anterior que 21 || 13 era igual a 29 o 11101₍₂₎, este valor le aplicamos la tabla de verdad del operador lógico **Not** lo que nos da como resultado 00010₍₂₎ es decir 4₍₁₀₎.

1.8.5 Expresiones de Asignación

Aunque existen lenguajes que manejan diversos tipos de expresiones de asignación, hay una expresión de asignación utilizando el operador \leftarrow y así diferenciarlo del operador relacional $=$ (igual a).

Las expresiones de asignación sirven para modificar el valor a una variable, lo que indica que del lado izquierdo de la expresión estará una variable, mientras que del lado derecho podrá estar un literal, una expresión de cualquier tipo u otra variable.

Ejemplo 16. Diversas expresiones de asignación

- a) $X \leftarrow 10$
- b) $X \leftarrow \text{"UNICORDOBA"}$
- c) $X \leftarrow 5 * 6 + 4$
- d) $X \leftarrow (2 + 5 > 3)$
- e) $X \leftarrow Y$

Analizando los ejemplos anteriores podemos observar: a) le asignamos a una variable X un literal entero, b) le asignamos a la variable X un literal de tipo cadena, c) le asignamos a la variable X una expresión aritmética, d) le asignamos a la variable X una expresión relacional y e) se le asigna a la variable X otra variable Y.

1.8.6 Expresiones algorítmicas

Para los sistemas de cómputo actuales no reconocen expresiones algebraicas; por ello hay que convertir dichas expresiones a expresiones lineales utilizando los operadores aritméticos, estas expresiones lineales son conocidas de igual manera como expresiones algorítmicas.

Ejemplo 17. De expresiones algebraicas a expresiones algorítmicas

$$\frac{(3+5)}{2} + \frac{1}{4} = ((3+5)/2) + (1/4)$$

La expresión a la izquierda del igual es algebraica y es convertida a forma lineal del lado derecho del igual a expresión algorítmica.

1.9 Metodología para la construcción de algoritmos

Para la construcción de algoritmos hay aproximaciones metodológicas pero si se requiere hacerlos, necesitamos lógica matemática, lógica formal y no formal (sentido común). Sin embargo, se expondrá una metodología genérica para la creación de los mismos.

Como primer paso se necesita analizar el algoritmo, este análisis consiste en tener claro el objetivo del mismo luego se definen las entradas que son los datos que el usuario proporciona al algoritmo a través de los dispositivos de entrada, se reconocen a continuación las variables intermedias que son variables auxiliares que ayudarán a almacenar provisionalmente datos o expresiones, se identifican inmediatamente los procesos que son la transformación de las entradas a través de expresiones y/o instrucciones y por último las salidas que son los resultados que serán presentados al usuario por medio de los dispositivos de salida.

Ejemplo 18. Aplicación de la metodología

Se requiere construir un algoritmo que calcule el área de un triángulo dadas la base y la altura:

Objetivos: El objetivo de este algoritmo es calcular el área del triángulo para cualquier base y altura digitada por el usuario
Entradas: Base y Altura
Variables Intermedias: No hay
Procesos: $\text{Área} = \text{Base} * \text{Altura} / 2$
Salidas: Área

Primero se determina el objetivo claramente, después con base al discurso del problema se establecen las entradas a continuación los procesos que pueden o no involucrar variables intermedias y luego las salidas.

1.10 Herramienta de pseudo-código

Hay varias Herramientas para la elaboración de algoritmos como son los diagramas de flujo, los diagramas N-S entre otras, pero para nuestro libro usaremos el pseudo-código.

El pseudo-código se aproxima mucho al código de los lenguajes de programación pero no es tan riguroso en cuanto a sintaxis, aunque tiene una estructura que es la siguiente:

Bloque de declaraciones: aquí se declaran las variables, funciones y procedimientos.

Bloque principal: es donde se escriben las instrucciones y expresiones de asignación que lógicamente dispuestas alcanzarán el objetivo del algoritmo.

Variables Declaración de variables
Función y Procedimiento Definición de funciones y procedimientos
Inicio ... Instrucciones ... Fin

Cabe anotar que las palabras **Función**, **Procedimiento**, **Inicio** y **Fin** se convierten automáticamente en palabras reservadas

Ejemplo 19. Muestra de aplicación del pseudo-código

Se requiere construir un algoritmo que calcule el área de un triángulo dadas la base y la altura:

Análisis:

Objetivos: El objetivo de este algoritmo es calcular el área del triángulo para cualquier base y altura digitada por el usuario
Entradas: Base y Altura
Variables Intermedias: No hay
Procesos: $\text{Área} = \text{Base} * \text{Altura} / 2$
Salidas: Área

Pseudo-código:

Variables

Entero Base, Altura, Area

Inicio

Leer (Base, Altura)

Area \leftarrow Base * Altura / 2

Escribir (Area)

Fin

Podemos notar que en el ejemplo se utilizan dos palabras reservadas **Leer** y **Escribir** que serán estudiadas en el próximo capítulo, solo se pretende mostrar como se escribe un algoritmo en pseudo-código, como también como el análisis es transformado en pseudo código.

1.11 Comentarios

Los comentarios se usan para explicar una línea o un bloque de pseudo-código, son muy útiles porque marcan indicaciones para el entendimiento del algoritmo por otro usuario o por el propio creador. No influyen en el resultado del algoritmo. En nuestro caso usaremos // doble barra para comentar líneas y /* barra asterisco */ asterisco barra para comentar bloques.

Ejemplo 20. Comentarios en pseudo-código

Se requiere construir un algoritmo que calcule el área de un triángulo dadas la base y la altura:

Análisis:

Objetivos: El objetivo de este algoritmo es calcular el área del triángulo para cualquier base y altura digitada por el usuario
Entradas: Base y Altura
Variables Intermedias: No hay
Procesos: $\text{Área} = \text{Base} * \text{Altura} / 2$
Salidas: Área

Pseudo-código:

```
/* Algoritmo para calcular el
   área de un triángulo */
```

Variables

```
Entero Base, Altura, Area    // declaración de variables
```

Inicio

```
Leer (Base, Altura)          // entradas del algoritmo
```

```
Area ← Base * Altura / 2     // procesos
```

```
Escribir (Area)              // salidas del algoritmo
```

Fin

En este ejemplo se observan los comentarios por bloque /* */ y por líneas //, el algoritmo es el mismo pero ahora esta descrito y será mas claro y entendible.

Problemas Resueltos

Problema 1.1 Diga cual de los siguientes nombres pueden usarse como nombre de variables o identificadores válidos:

- a) Unicor b) Licenciatura en Informática
- c) 40Grados d) Año_escolar
- e) Suma+Total f) calle#1

Rtas. a) Si, b) no, c) no, d) no, e) no, f) no

Problema 1.2 Cual es el valor resultante de las siguientes expresiones aritméticas

- a) $((48/79) - 2) + (9 \text{DIV} (9*3+9))$
 $(0,60-13) + (9 \text{DIV} 35)$
 $-12.3 + 0.25$
 -12.05
- b) $(3+5*(10-(2+4)))$
 $3+5*(10-6)$
 $3+5*6$
 $3+20$
 23
- c) $((12*(2+4))14)-((82 \text{MOD} 4+8) \text{DIV} (5*12)18))$
 $((12*15)-((82 \text{MOD} 12) \text{DIV} 7-5))$
 $18-10 \text{DIV} 7.5$
 $8 \text{DIV} 7.5$
 1.0
- d) $(13*2) + 15 \text{DIV} 2+4$
 $26+15 \text{DIV} 6$
 $41 \text{DIV} 6$
 6
- e) $18+25*9 - 54/6$
 $18+255 - 9$
 234
- f) $18 - 12 + ((37/8 - 3) * 23 \text{DIV} 7)$
 $18 - 12 + ((4.6 - 3) * 23 \text{DIV} 7)$
 $18 - 12 + 1.6 * 23 \text{DIV} 7$
 $18-12+36.8 \text{DIV} 7$
 $42.8 \text{DIV} 7$
 $42.8 \text{DIV} 7$
 6.1
- g) $135+6*9 \text{DIV} (6*9) + 144/2* (5+7-4) \text{MOD} 3$
 $135+6*9 \text{DIV} 54 + 144/2* 8 \text{MOD} 3$
 $135 + 54 \text{DIV} 54 + 144/16 \text{MOD} 3$
 $135 + 54 \text{DIV} 54 + 9 \text{MOD} 3$
 $189 \text{DIV} 63 \text{MOD} 3$
 $3 \text{MOD} 3$
 $0.$
- h) $8+6*5-214$
 $8+30 - 0.5$
 $38 - 05$
 37.5
- i) $45 \text{DIV} 9*5+1 + (3*7 \text{MOD} 6) / 2$
 $45 \text{DIV} 9*5+1 + (21 \text{MOD} 6) / 2$
 $45 \text{DIV} 9*5+1 1.5$
 $45 \text{DIV} 9*45+1+1.5$
 $45 \text{DIV} 47.5$
 0.9
- j) $3*8 (90 \text{DIV} 2) \text{DIV} 8*2+7$
 $3*8*45 \text{DIV} 16+7$
 $1080 \text{DIV} 23$
 46

$$\begin{aligned} \text{k)} \quad & 8+9*4/7*9 \\ & 8+36/7*9 \\ & 8+45.9 \\ & 53.9 \end{aligned}$$

$$\begin{aligned} \text{l)} \quad & 7*8*2*(27 \text{ MOD } 4) \text{ DIV } 3*2-5 \\ & 7*8*2*3 \text{ DIV } 3*2-5 \\ & 336 \text{ DIV } 6-5 \\ & 336 \text{ DIV } 1 \\ & 336 \end{aligned}$$

$$\begin{aligned} \text{m)} \quad & 36-12*2+8(7 \text{ MOD } 2) \\ & 36-12*2+8*1 \\ & 36-24+8 \\ & 20 \end{aligned}$$

$$\begin{aligned} \text{n)} \quad & (40 \text{ MOD } (9+7*8)/15) \text{ DIV } 7-5+1 \\ & (40 \text{ MOD } (65)/5) \text{ DIV } 7-6 \\ & (40 \text{ MOD } 13) \text{ DIV } 1 \\ & 1 \text{ DIV } 1 \\ & 1 \end{aligned}$$

$$\begin{aligned} \text{o)} \quad & 3+12/4*(81 \text{ MOD } 2*7)/11 \\ & 3+12/4*(81 \text{ MOD } 14)/11 \\ & 3+3*11/11 \\ & 3+33/11 \\ & 6 \end{aligned}$$

$$\begin{aligned} \text{p)} \quad & 9*2/2+(20-8) \text{ MOD } 2 \\ & 9*2/2+12 \text{ MOD } 2 \\ & 9+12 \text{ MOD } 2 \\ & 21 \text{ MOD } 2 \\ & 1 \end{aligned}$$

$$\begin{aligned} \text{q)} \quad & 10+5*2-40/5 \\ & 10+10-8 \\ & 20-8 \\ & 12 \end{aligned}$$

$$\begin{aligned} \text{r)} \quad & 5+10*15-35/2 \\ & 5+150-17.5 \\ & 155-17.5 \\ & 137.5 \end{aligned}$$

$$\begin{aligned} \text{s)} \quad & 25*2/2+(8+(40-3*7)/3)-9 \\ & 25*2/2+(8+(40-21)/3)-9 \\ & 25*2/2+(8+19/3)-9 \\ & 25*2/2+8+6.3-9 \\ & 50/2+8+6.3-9 \\ & 25+8+6.3-9 \\ & 30.3 \end{aligned}$$

$$\begin{aligned} \text{t)} \quad & 7*8*2*(27 \text{ MOD } 4) \text{ DIV } 3*2-5 \\ & 7*8*2*1 \text{ DIV } 3*2-5 \\ & 112 \text{ DIV } 6-5 \\ & 112 \text{ DIV } 1 \\ & 112 \end{aligned}$$

$$\begin{aligned} \text{u)} \quad & 40-5+15-9 \text{ MOD } 3*2+2 \\ & 8+15-9 \text{ MOD } 6+2 \\ & 23-9 \text{ MOD } 8 \\ & 14 \text{ MOD } 8 \\ & 6 \end{aligned}$$

Problema 1.2 Cual es el estado de las siguientes expresiones relacionales

1. $((240 \text{ DIV } 2) \leq (12 * 5 + 6 \text{ DIV } 3))$
 $120 \leq (60 + 12 / 4 \text{ DIV } 3)$
 $120 \leq 60 + 3 \text{ DIV } 3$
 $120 \leq 63 \text{ DIV } 3$
 $120 \leq 21 = \textbf{(Falso)}$
2. $3 * 5 - 18 > 27 / 3 + 1$
 $15 - 18 > 9 + 1$
 $-3 > 10 = \textbf{(Falso)}$
3. $A + 5 < B + 7$ $A=10, B=5$
 $10 + 5 < 5 + 7$
 $15 < 12 = \textbf{(Falso)}$
4. $43 \text{ MODE } 15 < (30 * 6) / 18$
 $13 < 180 / 18$
 $13 < 10 = \textbf{(Falso)}$
5. $12 \text{ MOD } 3 > (2 * 4) / 5$
 $0 > 8 / 5$
 $0 > 1.6 = \textbf{(Falso)}$
6. $(14 \text{ MOD } 3) * 6 > 12 / 4 + 9 \text{ DIV } 2 + 4$
 $2 * 6 > 3 + 9 \text{ DIV } 2 + 4$
 $12 > 12 \text{ DIV } 6$
 $12 > 2 = \textbf{(Verdadero)}$
7. $43 * 5 - 8 \text{ DIV } 6 > 93 - 6 \text{ MOD } 3 + 9$
 $215 - 8 \text{ DIV } 6 > 87 \text{ MOD } 12$
 $207 \text{ DIV } 6 > \text{MOD } 12$
 $34 > 3 = \textbf{(Verdadero)}$
8. $90 \text{ DIV } 3 + F + A > = F \text{ MOD } 6 + (A * 2)$ $F=26, A=19$
 $90 \text{ DIV } 3 + 26 + 19 > = 26 \text{ MOD } 6 + (19 * 2)$
 $90 \text{ DIV } 48 > = 26 \text{ MOD } 6 + 38$
 $90 \text{ DIV } 48 > = 26 \text{ MOD } 44$
 $1 > = 26 = \textbf{(Falso)}$
9. $A * 7 < J + (3 * 2) =$ $A=9; J=10$

$$9*7 < 10 + (3*2)$$

$$63 < 10+6$$

$$63 < 16 = \textbf{(Falso)}$$

10. $Z \ 15 < N3 = \quad Z=5; \ N=24$

$$5*15 < 24*3 =$$

$$75 < 72 = \textbf{(Verdadero)}$$

11. $25+7 - (16*8) < (712 \text{ MOD } 5) / 3$

$$25+7-128 < 2/3$$

$$96 < 0.6 \textbf{(Falso)}$$

12. $X*Y \ Z+4 =$

$$7*3 > 8+4$$

$$21 > 12 \textbf{(Verdadero)}$$

13. $24/15+7 \text{ DIV } 3+9 < (6*8) - 38$

$$1.6+7 \text{ DIV } 12 < 48-38$$

$$8.6 \text{ DIV } 12 < 10$$

$$0 < 10 \textbf{(Verdadero)}$$

14. $45 > = A*C \text{ DIV } 3*3 \quad A=18; \ C=15$

$$45 > = 18*15 \text{ DIV } 3*3$$

$$45 > = 270 \text{ DIV } 9$$

$$45 > = 30 \textbf{(Verdadero)}$$

15. $(33*5) + 52 < (13* 10 \text{ MOD } 4)$

$$165 + 52 < (130 \text{ MOD } 4)$$

$$217 < 2 \textbf{(Falso)}$$

16. $50 \text{ MOD } 20 > (9/20) + 15$

$$50 \text{ MOD } 20 > 0.45+15$$

$$10 > 15.45 \textbf{(Falso)}$$

17. $B* 5+C > A* 2+C \quad A=10; \ B=20; \ C=30$

$$20*5+30 > 10*2+30$$

$$100 + 30 > 20+30$$

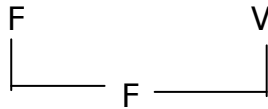
$$130 > 50 = \textbf{(Verdadero)}$$

Problema 1.3 Calcule es el estado de las siguientes expresiones lógicas:

1. $(20 > 8*3) \text{ AND } (36 > 4*6 \text{ Div } 3)$

$20 > 24 \text{ AND } 36 > 24 \text{ Div } 3$

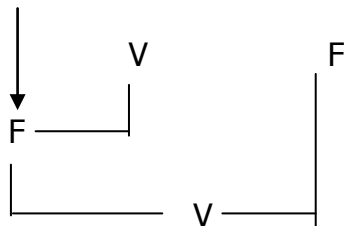
$20 > 24 \text{ AND } 36 > 8$



2. $\text{NOT } (6 > 5) \text{ OR } (8*4 \text{ MOD } 4 > 5)$

$\text{NOT } (6 > 5) \text{ OR } (32 \text{ MOD } 4 > 5)$

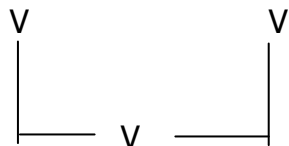
$\text{NOT } (6 > 5) \text{ OR } (0 > 5)$



3. $(11 = < (2*7)*5) \text{ OR } (19 < (4*6 + 15) / 12))$

$(11 = < 70) \text{ OR } (19 < 360 / 12)$

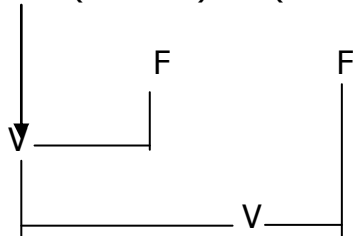
$(11 = < 70) \text{ OR } (19 < 30)$



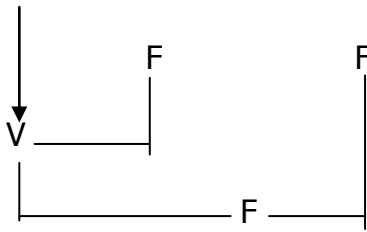
4. $\text{NOT } (1612*3 < 15) \text{ OR } (18 - 4 \text{ div } 7 > (15/3*2))$

$\text{NOT } (24 < 15) \text{ OR } (14 \text{ Div } 7 > 10)$

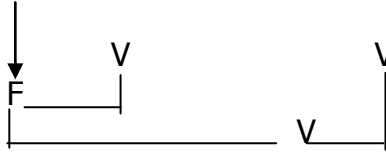
$\text{NOT } (24 < 15) \text{ OR } (2 > 10)$



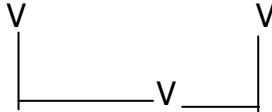
5. NOT $(7*9 < 14/7)$ AND $(73 - 37 > = 18 + 10 *2 - 20)$
 NOT $(63 < 2)$ AND $36 > = 18 + 20 - 20$
 NOT $(63 < 20)$ AND $36 > = 18$



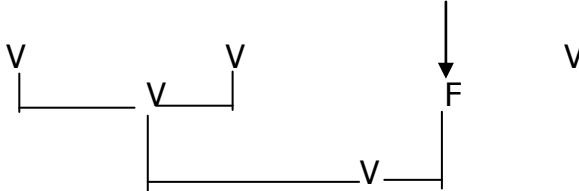
6. NOT $(16 > = 8*2)$ OR $(4*6 > 6*3 + 1)$
 NOT $(16 > = 16)$ OR $(24 > 19)$



7. $(25 < 9*3)$ AND $(52 + 5 - 12) > 14*5 \text{ MOD } 7)$
 $(25 < 27)$ AND $(45 > 70 \text{ MOD } 7)$
 $(25 < 27)$ AND $(45 > 0)$



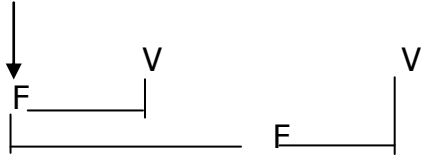
8. $(16 > 12 \text{ AND } 3 < > 7)$ OR NOT $(20 \text{ DIV } 2 = < 17)$
 $(16 > 12 \text{ AND } 3 < > 7)$ OR NOT $(10 = < 17)$



9. NOT $(39 = < 4*9)$ AND $(56 + 3*9 \text{ DIV } 6 < > 17 - 12 \text{ DIV } 4)$
 NOT $(39 = < 36)$ AND $(56 + 27 \text{ DIV } 6 < > 5 \text{ DIV } 4)$
 NOT $(39 = < 36)$ AND $(83 \text{ DIV } 6 < > 5 \text{ DIV } 4)$
 NOT $(39 = < 36)$ AND $(13 < > 1)$



15. NOT (20 > = 5*3) AND (25 - 3*4 DIV 2 < > 5*2)
 NOT (22 > 15) AND (13 DIV 2 < > 10)
 NOT (20 > = 15) AND (6 < > 10)



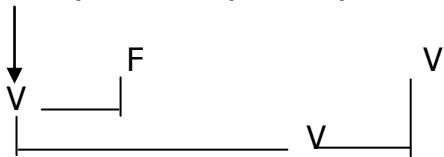
16. (5+2 > = 4) AND (3-2 < 7*4)
 (7 > = 4) AND (1 < 28)



17. (14 + 2 < 3) AND (20 + 1 = 6*6)
 (16 < 3) AND (21 = 36)



18. NOT (13 > = 8*2) AND (24 - 6*3 DIV 5 < > 3*4 DIV 3)
 NOT (13 > = 16) AND (6 DIV 5 < > 12 DIV 3)
 NOT (13 > = 16) AND (1 < > 4)



19. (8+2 > = 3) AND (5-1 < 5*6)
 (10 > = 3) AND (4 < 30)



20. (20 > = 8*2) AND (24 - 8 > 8 DIV 4)
 (20 > = 16) AND (16 > 2)



Problema 1.4 Determine el valor de las siguientes expresiones de bits:

1. $15 \& 24 = 8$

$\& 01111 (15)$

$11000 (24)$

$1000 (8)$

2. $!(164 \& 325) = 507$

$\& 010100100$

101000101

$! 000000100$

$111111011 (507)$

3. $!(32 || 22) \& 10 = 8$

$32 = 100000 || 100000 \& 001001$

$22 = 10100 \quad \underline{010110} \quad \underline{001010}$

$\quad \quad \quad ! \quad 110110 \quad \quad 001000$

$\quad \quad \quad ! \quad 001001$

4. $86 || (112 \& 22) \& 26 = 62$

5. $198 \& 240 = 192$

$\& 11000110$

11110000

11000000

6. $70 \& 90 = 66$

$\& 1000110$

1011010

1000010

7. $50 \& 20 = 2$

$\& 110010$

001010

000010

8. $22 \& 14 = 4$

$\& 10110$

01100

00100

9. $(! 5 \parallel (3 \& 4\&)) = 2$

5 = 101	3 = 011	& 011	010
! 5 = 010	4 = 100	<u>100</u>	<u>000</u>
		000 (1)	010 (2)

10. $!1311\ 20 = 22$

11. $!(!12 \& (6 \parallel 9)) \& 14 = 1100$

12. $15 \& 11 = 11$

& 01111 (15)
<u>01011</u> (11)
01011 (11)

13. $(7 \& ((69\ 11\ 16)\ 11\ 34))\ 11\ !(6 \& 9) - 15$

14. $8 \& 6 = 0$

& 001000
<u>000110</u>
000000 (0)

15. $20 \& 19 = 16$

& 010100
<u>010011</u>
010000 (16)

16. $18 \& 14 = 2$

& 010010 (18)
<u>001110</u> (14)
000010 (2)

17. $10 \& 20 = 0$

& 10100 (10)
<u>01010</u> (20)
00000 (0)

18. $110 \mid\mid 24 = 126$

11 1101110 (110)
0011000 (24)
 1111110 (126)

19. $!(4 \& 20) = 20$

& 00100 (4)
10100 (20)
 ! 00100
 10100

20. $14 \& 5 = 4$

001110
000101
 000100 (4)

Problema 1.5 Cual será el valor resultante de las variables después de ejecutadas las siguientes expresiones de asignación:

1. a xx 15 R = a = 5
 b xx a + b b = 5
 c xx a + b c = 20
 a xx c - a
 b xx a

2. C xx 15 R = A = 27
 A xx 7 C = 22
 F xx C F = 35
 A xx F
 C xx A
 F xx 20

3. R xx 8 R = R = 6
 S xx 2 T = 2
 T xx 6 S = 8
 S xx R
 T xx S
 R xx T

4. X ← 5	R = X = 10
Y ← 4	Y = 5
Z ← 10	Z = 4
X ← Z	
Y ← X	
Z ← Y	

5. H ← 5	R = N = 5
L ← 3	H = 3
N ← H	J = 5
4 ← 5	
J ← N	
N ← '5'	

6. Z ← '5'	R = 2 = 1
4 ← '1'	9 = 8
3 ← '8'	3 = 8
2 ← 4	
9 ← 3	
3 ← 'Z'	

7. 15 ← F	R = 15 = 9
12 ← 9	12 = H
9 ← K	9 = F
15 ← 12	
12 ← 9	
9 ← 15	

8. A ← 16	R = A = 4
F ← 4	F = 16
P ← 4	P = 16
A ← F	
F ← P	
P ← 16	

9. D \leftarrow 4 R = D = 6
 I \leftarrow 9 I = 26
 N \leftarrow 14 N = 9
 A \leftarrow 1 A = 3
 D \leftarrow 54 li
 I \leftarrow 16 Div D
 N \leftarrow 3 + 6
 A \leftarrow 6-3

10. X \leftarrow 3 R = X = 21
 Y \leftarrow 18 Y = 39
 Z \leftarrow Y Z = 24
 X \leftarrow Z
 Y \leftarrow X
 Z \leftarrow 6

11. J \leftarrow 'B' R = J = S
 A \leftarrow '?' AA = 6
 G \leftarrow 'i' G = 'i'
 J \leftarrow A
 AA \leftarrow 'G'
 J \leftarrow 'S'

12. Y \leftarrow 50
 ULT \leftarrow Y DIV 3
 Y \leftarrow Y MAS 4
 MES \leftarrow Y DIV 4
 Y \leftarrow Y MOD 4
 PRIM \leftarrow Y DIV 4

13. a \leftarrow 15 R = d = 0
 b \leftarrow 2 b = 15
 c \leftarrow b c = 2
 b \leftarrow a
 a \leftarrow b DIV 10

14. X \leftarrow 3 R = X = 10
 b \leftarrow 10 b = 3
 z \leftarrow 5 z = 5
 b \leftarrow x
 x \leftarrow b

15. $a \leftarrow 25$ $R = a = 50$
 $b \leftarrow 2$ $b = 25$
 $c \leftarrow 50$ $c = 50$
 $a \leftarrow c$
 $b \leftarrow a$
16. $a \leftarrow 5$ $R = a = 3$
 $b \leftarrow 3$ $b = 1$
 $c \leftarrow 1$ $c = 2$
 $a \leftarrow b$
 $b \leftarrow c$
 $c \leftarrow 2$
17. $J \leftarrow 20$ $R = J = 15$
 $K \leftarrow 15$ $= K = 20$
 $L \leftarrow J$ $= L = 20$
 $J \leftarrow K$
 $K \leftarrow L$
 $L \leftarrow 20$
18. $a \leftarrow 8$ $R = a = 9$
 $b \leftarrow a$ $= b = 4$
 $c \leftarrow b$ $= c = 5$
 $a \leftarrow c$
 $b \leftarrow a$
 $c \leftarrow 5$
19. $a \leftarrow 5$ $R = a = 15$
 $b \leftarrow 10$ $R = b = 15$
 $c \leftarrow 15$ $R = c = 15$
 $a \leftarrow c$
 $b \leftarrow a$
 $c \leftarrow b$
20. $B \leftarrow 1$ $R = A = 1$
 $A \leftarrow 4*5*3$ $B = 2$
 $A \leftarrow 4*1$ $C = 12$
 $A \leftarrow A+B$
 $C \leftarrow A+B$
 $C \leftarrow C+A+B$
 $A \leftarrow A*0$
 $A \leftarrow 1$
 $B \leftarrow 1 + 1$

Problema 1.6 Transformar a expresiones algorítmicas las siguientes expresiones algebraicas

$$1. \frac{c+2}{x-5x} = (c + 2) / (x - 5x)$$

$$2. \frac{a}{3} - \frac{5mx}{6b} = (a/3) - ((5*m*n) / (6*b))$$

$$3. \frac{A-B+A}{P*2} = ((A - B + A) / (P*2))$$

$$4. \frac{A-B*C}{B} * \frac{A+B}{B*C} = ((A - (B*C) / B) (A + B) / (BC))$$

$$5. J*Y + \frac{d*n}{h*i} = (J*y) + ((d*n) / (h*i))$$

$$6. \frac{x+y*z}{h-t} = (x*z + x) / h - t)$$

$$7. \frac{CA + B*D}{R+Z} = ((B*D) + CA) / (R + Z)$$

$$8. \frac{d*b-p}{L*t} * c - w = ((db)-p) / (L + z)* c - w$$

$$9. R*H - \frac{L+T}{M} * \frac{B}{5} = ((R*H) - L + T) / M)*(B/S)$$

$$10. F*A + \frac{E}{A} + \frac{F*A}{FA} = (F*A) + (F/A) + ((F/A) / (FA))$$

$$11. \frac{d-i}{n*a} + \frac{m+r*c}{e-L*v} + h-k =$$

$$((d - i) / (h*a)) + (m + r *c)) / (e - L/v) * (h - k)$$

$$12. \frac{P*Z + b - c}{h*x} = ((p*z) + b) - c) / (h*c)$$

$$13. \left\{ \frac{B+2}{10} - \frac{A*B}{2} \right\} = ((b+2) / 10) - ((A*B) / 2)) / 2^a$$

$$10 \quad 2$$

$$14. \frac{Y*Z + P}{0+M} - \frac{h}{k} + i*x = (Y*Z) + P / (0 + M) - (h / k) + (i*x)$$

$$15. \frac{7 + P}{x} = ((7 / x) + p / 4)$$

$$16. \frac{4^a + b - c}{d + e} = (4^a + b - c) / (a + e)$$

$$17. \frac{A}{B} * C + \frac{d*e}{F} * g = ((A / B)* c) + ((d*e) / f)* g)$$

$$18. \frac{\alpha}{b} * \frac{c - d}{d + \alpha} = (a / b) * ((c - d) / (d + \alpha))$$

$$19. \frac{P + A - M}{B - C} * Y + Z = ((P + A - M) / (B + C)) * Y + Z$$

$$20. \frac{A*B}{C} * \frac{b}{e-f} = ((A*B) / C)* (D/ e - F)$$

Ejercicios Suplementarios

Ejercicio 1.1 Cual de los siguientes nombres es valido para identificar una variable:

- | | |
|---------------------|---------------------|
| a) Total | e) Salario Bruto |
| b) Impuestos | f) Ingresos-Egresos |
| c) Valor_En\$ | g) 100Nombres |
| d) Resultado_ventas | |

Ejercicio 1.2 Cual es el valor de las siguientes expresiones aritméticas:

- | | |
|-----------------------|-----------------------------|
| a) 1+6 | f) 35+6-7 MOD (8+2) / 1+3+5 |
| b) * 7 - 18/2 | g) 8*5 - 4+16/7 |
| c) 7+5 * 8-36 DIV 2+3 | h) 14*8 MOD 10/2 |
| d) 9+7*8 - 36/5 | i) 12*5 / (10 DIV 2) +4 |
| e) 7+8 * 3+4*6 | |

- | | |
|--|--|
| j) $10 / 20 + 30 - 2 \text{ DIV } 3 * 3 + 1$ | o) $3 * 9 + 12 * (5/2 + 8) \text{ DIV } 2$ |
| k) $7 * 8 * 2 (27 \text{ MOD } 4) \text{ DIV } 3 * 2 - 5$ | p) $5 - 2 + 9 * 17/8$ |
| l) $8 - 3 * 2 * (90 \text{ MOD } 4 - 2) / 10$ | q) $18 + 5 * 5 * 6/3$ |
| m) $8 + 20/2 * 4 + (3 - 6)$ | r) $10 + 8 * 3 - 49/3$ |
| n) $6 * 3 (140 \text{ MOD } 2 * 2) \text{ DIV } 3 * 10 - 15$ | s) $5 + 2 * 6 - 19$ |

Ejercicio 1.3 Cual es el estado las expresiones relacionales siguientes:

- a) $(A * B) + (7 * 12) / 2 > (7 + B * 4)$, Donde $A=20$; $B=4$
- b) $4 * 9 < X - 4$, Donde $Y=5$; $X=6$
- c) $(X * 3 \text{ DIV } 2) < Y + 5$, Donde $X=6$; $Y=2$
- d) $60 \text{ MOD } 15 < (5 - 30) * 5$
- e) $A * 5 > B * 19$, Donde $A=9$, $B=2$
- f) $A * 4 > B * 6$, Donde $A=8$; $B=14$
- g) $B * A > A + C$, Donde $A=4$; $B=5$; $C=1$
- h) $A * 6 > B * 6$, Donde $A=3$; $B=20$
- i) $(50 * 5) / 5 = 2 + 5 * 4$
- j) $X * 3 < Y - 2$, Donde $X=4$; $Y=3$
- k) $(X * 3 + B * 5/2) = < (X * 3 \text{ DIV } B)$, Donde $X=8$; $B=7$
- l) $(2 * 4 + 4) > (7 + 8 * 8/4)$
- m) $A * 4 < B * 2$, Donde $A=5$; $B=12$
- n) $A * 7 + Y * 9 > = (A * 3 \text{ DIV } Y * 2)$, Donde $A=2$; $Y=5$
- o) $(X * 3 + 16 - 2) > (24 / (Y \text{ DIV } 3) + 4)$, Donde $X=6$; $Y=9$
- p) $X * 50 \text{ DIV } 9 = < (Y - 2)$, Donde $X=7$; $Y=4$
- q) $8 * Y > X * 20$, Donde $X=10$; $Y=15$
- r) $((8 * A + C - 5) / 5) > (3 + 5 \text{ MOD } 2)$, Donde $A=4$; $C=6$
- s) $10 / 2 * 5 + 20 > = 45 \text{ MOD } 18$
- t) $(E * 2 + 3/5) > (A - 9) * 2$, Donde $A=4$; $E=5$
- u) $(500 \text{ DIV } A) > = h * 2 / h * A$, Donde $h=10$; $A=15$

Ejercicio 1.4 Halle el es el estado las siguientes expresiones lógicas:

- a) 22. $\text{NOT} (12 > = 3 * 3) \text{ AND } (50 - 20 * 2 \text{ DIV } 3 < > 20)$
- b) 23. $\text{NOT} (5 > = 6 * 3) \text{ AND } (25 + 8 * 2 \text{ DIV } 3 < > 7 * 3 \text{ DIV } 4)$
- c) 24. $(12 + 6 > 15 - 6) \text{ OR } (12 \text{ DIV } 4 * 6 < 3)$
- d) 25. $((2 + 5) < (12 * 6 \text{ DIV } 2)) \text{ AND } (5/3 = 20)$
- e) 26. $(4 + 5 > = 10 - 2) \text{ OR } (12 \text{ MOD } 3 * 2 = 10)$
- f) 27. $(12 < 8 + 2 * 4) \text{ AND } ((5 + 7 \text{ MOD } 3) > 5)$

- g) 28. $\text{NOT} (20 \geq 8*9) \text{ AND } (32 - 15*4 \text{ DIV } 7 < > 4*48 \text{ DIV } 12)$
- h) 29. $(18 < 32) \text{ AND NOT } (10 \text{ DIV } 2) < 10$
- i) 30. $\text{NOT} (3 = < 6*2) \text{ AND } (30 + 7*1 \text{ DIV } 2 < > 2*4 \text{ DIV } 5)$

Ejercicio 1.5 Determine el es el valor las expresiones de bits a continuación:

- a) $40 \& 2$
- b) $!(8 \& 4)$
- c) $!(15 || 12)$
- d) $!(7 || 12)$
- e) $(25 \& !10) || 5$
- f) $11 \ 17) \& (20 \& 19)$
- g) $|| 15$
- h) $!(22 \& 10) || 15$
- i) $(!12 \& (24 || 30))$
- j) $8 \& 16 \& 20$

Ejercicio 1.6 Cual es valor resultante después de ejecutarse las siguientes expresiones de asignación.

a) $E \leftarrow '?'$
 $M \leftarrow 5$
 $T \leftarrow '@'$
 $T \leftarrow E$
 $E \leftarrow M$
 $M \leftarrow T$

b) $a \leftarrow 'a'$
 $b \leftarrow 'i'$
 $c \leftarrow 5$
 $c \leftarrow b$
 $a \leftarrow c$
 $b \leftarrow a$

c) $A \leftarrow B$
 $b \leftarrow 5$
 $c \leftarrow b \text{ MOD } 4$
 $A \leftarrow b$
 $C \leftarrow 5 \text{ MOD } 4$

d) $b \leftarrow 30$
 $x \leftarrow 8$
 $z \leftarrow b$
 $b \leftarrow x$
 $x \leftarrow z$
 $z \leftarrow 30$

e) $y \leftarrow 9$
 $e \leftarrow 14$
 $y \leftarrow 5$
 $e \leftarrow y$
 $J \leftarrow 1$
 $J \leftarrow e$

f) $S \leftarrow 18$
 $P \leftarrow 19$
 $D \leftarrow 5$
 $P \leftarrow D$
 $S \leftarrow P$

g) $Y \leftarrow Z$
 $X \leftarrow 2*1*3$
 $X \leftarrow 2*1$
 $X \leftarrow X + Y$
 $Z \leftarrow X + Y$
 $Z \leftarrow Z + X + Y$
 $X \leftarrow X*1$
 $X \leftarrow 2$
 $Y \leftarrow 1+1$

h) $L \leftarrow 3$
 $M \leftarrow 4$
 $N \leftarrow L$
 $L \leftarrow M$
 $M \leftarrow N$
 $N \leftarrow 6$

Ejercicio 1.8 Transformar las siguientes expresiones algebraicas a expresiones algorítmicas:

a) $\frac{X + F}{B} + \frac{T*B}{a-B}$

b) $\frac{Y + P*C}{a-m}$

c) $\frac{S + P}{a + \frac{b}{c}}$

d) $\frac{a * b}{c} * d - F$

e) $\frac{a + b * c - d}{e + z}$

f) $\frac{x + y}{a*b} - K - b$

g) $\frac{A * C}{X + Z} + J$

$$h) \frac{t + h * c}{Z + b}$$

$$i) \frac{R + L * h}{m - a}$$