

BASES DE PROGRAMACIÓN

ING. FRANCISCO E. MORÁN PEÑA MSC.
DR. FRANCISCO L. MORÁN PEÑA MSC.

BASES DE PROGRAMACIÓN

PRIMERA EDICIÓN

Bases de programación

Autores

Ing. Francisco E. Morán Peña MSc.
Dr. Francisco L. Morán Peña MSc.

Primera edición, abril 2017



Libro sometido a revisión de pares académicos.

Edición
Diagramación
Diseño
Publicación

Maquetación.

Grupo Compás
Cámara Ecuatoriana del Libro - ISBN-E: 978-9942-750-75-4
Guayaquil - Ecuador

Contenido

Capitulo #1.....	9
Conceptos básicos.....	10
Informática	10
La Computadora	10
Unidades de almacenamiento del computador	10
División de los componentes principales del computador	12
Sistemas operativos	12
Programas de aplicación.....	12
Programas comerciales	13
Lenguajes de programación.....	13
Actividad # 1.....	18
Capítulo# 2	19
Programa	19
Programa fuente	21
Traductores	21
Técnicas para resolver un programa	23
Planteamiento del problema.....	24
Algoritmo	24
Información y datos.....	27
Datos simples.....	28
Datos simples Numérico	28
Datos simples carácter	29
Datos simples lógico	29
Datos compuestos.....	29
Datos definidos	30

Normas para diseñar algoritmos.....	30
Actividad # 2.....	31
Capítulo 3.....	33
Elementos fundamentales para diseñar un programa	33
Constantes	34
Nombre de la constante.....	34
Valor de la constante.-	35
Variables	36
Nombre de la variable.-	37
Valor de la variable	38
Instrucción.....	38
Actividad # 3.....	45
Capítulo 4.....	47
Expresiones	47
Expresiones aritméticas.....	48
Operadores aritméticos.....	49
Reglas prácticas para la transformación de las expresiones.....	51
Expresiones lógicas.....	57
Operador relacional	57
Operador lógico.....	58
Representación gráfica de las expresiones lógicas y relacionales	61
Funciones.....	68
Actividad # 4.....	69
Capitulo# 5	73
Diagrama de flujo	73
Definición.....	73
Ventajas de los diagramas	75

Los símbolos	75
Símbolos de inicio o fin.....	76
Símbolo de entrada por teclado.....	77
Símbolo de proceso	78
Símbolo de entrada/ salida por periférico.....	83
Símbolo conectores internos en una página.....	84
Símbolo conectores externos en una página	85
Símbolo de decisión con bifurcación doble	86
Símbolo de líneas de flujo	87
Principios básicos para la resolución de diagramas de flujo	89
Reglas generales para la elaboración de Diagramas	90
Bifurcación	90
Decisiones con bucles o lazo	92
Actividad # 5.....	94
Capitulo #6	98
Clasificación de los diagramas de flujo según su estructura	100
Diagramas de estructuras simples	100
Prueba de escritorio.....	102
Actividad # 6.....	105
Capitulo #7	107
Diagramas de estructura complejas sin realimentación	108
Bucles no iterados	108
Clasificación de las estructuras complejas sin realimentación.....	109
Decisiones simples.-	109
Bucles no iterados	110
Decisiones dobles.-	113
Decisiones Do Case.-	119

Actividad #7.....	120
Capítulo# 8	123
Diagramas de estructura complejas con realimentación (repetitivas).....	124
Decisiones con bucles iterativos.....	125
Decisión.-	125
Cuerpo del bucle-	125
Salida del bucle.-	125
Bucles iterativos de control	126
Cuerpo de los bucles iterativos de control.-	126
Estructura repetitiva REPETIR – HASTA QUE	135
Características estructura repetitiva REPETIR-HASTA QUE	135
Estructura repetitiva MIENTRAS -HACER	136
Características Estructura repetitiva MIENTRAS -HACER	137
Actividades # 8	138
Capitulo #9	142
Diagramas de estructuras complejas con realimentación.....	143
Contadores	143
Acumuladores	150
Ciclos FOR	154
Actividad # 9.....	160
Capítulo# 10	170
Estructura de datos : Arreglos	171
Arreglos con un subíndice (vectores)	171
Operaciones con arreglos	173
Arreglos con dos subíndices (matrices)	174
Matriz X filas , Z columnas	174
Pseudocódigo	179

Enunciados de decisiones	180
Estructura selectiva SI - SINO (IF-ELSE-ENDIF).....	181
Estructura repetitiva REPETIR - HASTA (REPEAT-UNTIL)	184
Estructura repetitiva MIENTRA – FINMIENTRA (WHILE DO - END WHILE).....	187
Estructura COMIENZA-HASTA-INCREMENTO (FOR-TO--STEP-NEXT).190	
Bibliografía	194

Capítulo 1

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Revisar conceptos básicos.
2. Entender a la programación como un medio de comprensión del funcionamiento real del computador.
3. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.

Conceptos básicos

Informática

La real academia de la lengua (2014) define a la informática como “conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras”. En este sentido Morán y Moran (2017) expresan La Informática “es la ciencia que estudia los procedimientos, que tiende a reducir o suprimir la intervención humana de las diferentes actividades de la información, de una manera sistematizada y veloz, a través de las computadoras u otros medios”(p.7).

La Computadora

Es una máquina electrónica, que recibe datos, los procesa, almacena y obtiene y produce resultados. Entre sus partes más importantes se tienen: monitor, teclado, Unidad Central de Sistema (UCS) y mouse (ratón)

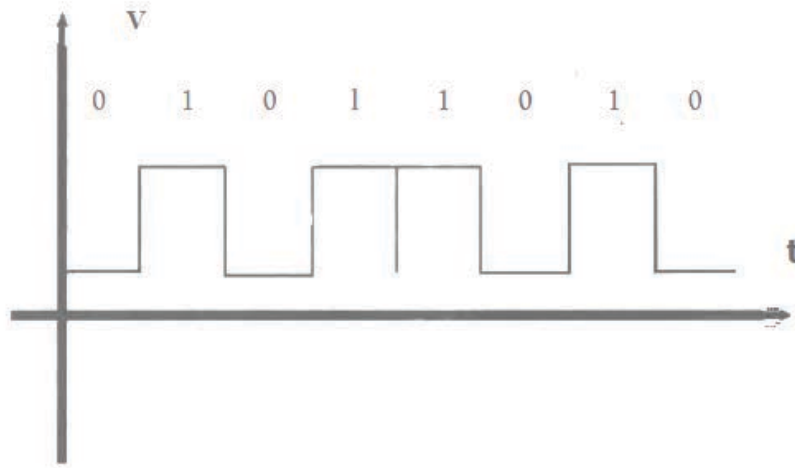
Unidades de almacenamiento del computador

Las unidades que sirven para medir la capacidad de almacenamiento de datos guardados en unidades de almacenamiento como discos duro, flexible, etc, se basan en el BIT (dígito binario), que es la mínima unidad que puede ser representada mediante el código binario: (0 ó 1). Por otra parte la gran cantidad de información que maneja un computador origino crear otra unidad denominada el Byte, representa la unión de 8 bits, pero esta unidad resulta muy pequeña en la actualidad, creándose el kilobyte, megabyte, gigabyte y el terabyte. Las equivalencias de estas unidades se detallan a continuación:

1 Bit	= 1 ó 0
1 Byte	= 8 bit
1 Kilobyte (KB)	= 1024 Byte
1 Megabyte(MB)	= 1024 Kilobytes
1 Gigabyte (GB)	= 1024 Megabytes
1 Terabyte (TB)	= 1024 Gigabytes

1 Pentabyte(PB) = 1024 Terabytes
 1 Exabyte (EB) = 1024 Pentabytes

Figura 1: Representación de la letra Z mediante señales digitales

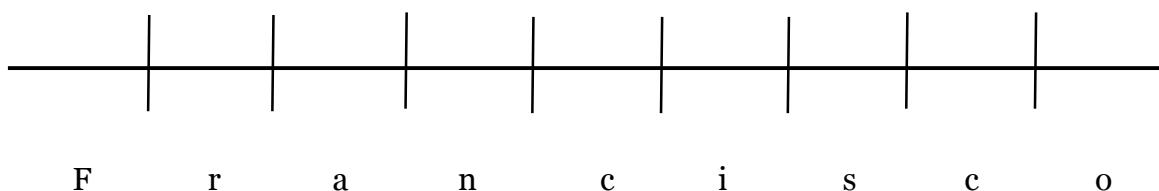


Lo importante para este estudio es saber que al igual que el ser humano posee una forma de comunicarse (el idioma, español, francés etc.), el computador trabaja a base de combinaciones de ceros y uno (código binario) de esa forma interpreta las instrucciones y es su forma de comunicarse por ejemplo:

El código 01011010 equivale a la letra Z, 00111100 equivale al signo <, el 10101011 equivale 1/2.

Una vez ingresada la información al computador es transformada automáticamente a código binario para realizar una tarea deseada. Por ejemplo: Si tipiamos en el teclado la palabra Francisco el computador busca automáticamente mediante una tabla de códigos llamada ASCII (Código Estándar Americano para Intercambio de Información) .Donde se encuentra el valor de todos los símbolos que utiliza el computador en código binario ,entonces el procesamiento de datos lo realiza de una manera sumamente veloz. La figura 2 muestra el código binario de la palabra Francisco

Figura 2: Código binario de palabra Francisco



01000110 01110010 01100001 01101110 01100011 01101001 11110011 01100011 01101111

División de los componentes principales del computador

La computadora para su estudio se divide en: Hardware, Firmware y Software.

Hardware es el conjunto de componentes físicos del computador, es decir, toda la parte tangible y dura ejemplo: monitor, teclado.

Firmware realiza la función de enlace entre el hardware y software por ejemplo: Memoria ROM, PROM etc.

Software es la parte lógica que indica al computador como resolver una tarea encomendada por el ser humano. Generalmente se la conoce como la parte intangible, también es llamada programa. Este es el punto de partida para el estudio de la programación básica. El software se clasifica según su utilidad en: sistema operativo, programas de aplicación, programas comerciales y lenguajes de programación

Sistemas operativos

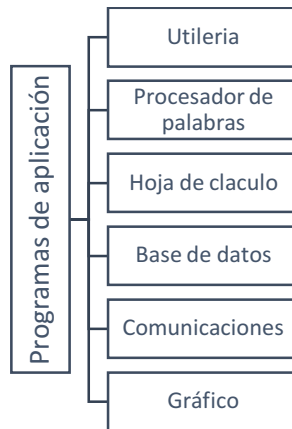
Los sistemas operativos son programas que ayudan a preparar al computador (teclado, unidad central de procesos, sistema de videos, etc.) para su normal funcionamiento y utilización por ser humano, por ejemplo: en automóvil la única manera para preparar su funcionamiento es mediante el ingreso de la llave que permite que todos los dispositivos estén listos para que el ser humano lo utilice. Los sistemas operativos más conocidos son: D.O.S. UNIX, OS/, Windows, Linux

Programas de aplicación

Son programas que simulan las actividades más comunes que realizan el ser humano. Por ejemplo:

El programa que simula la función que realizado por una máquina de escribir se llama WORD. Así mismo el programa que simula la función realizada en un lienzo de pintura con su respectiva paleta de colores se llama PAINT.

Figura 3: Clasificación de los programas de aplicación



Programas comerciales

Son diferentes programas que se encargan de resolver un problema específico según el tipo de aplicación, dependiendo al área en que se lo utilice ejemplo: sistema bancario, diagnóstico Médico, etc.

Lenguajes de programación










Es un elemento fundamental para la creación de los programas y sirven para poder comunicamos con el computador, compuesto de un conjunto de símbolos, caracteres y reglas que utilizadas sistematizadamente nos ayudan a dar órdenes que entienda el computador para que realice una tarea en forma inmediata. Además Joyanes (2008) considera:

El algoritmo se ha de expresar en un formato que se denomina programa, ya que el pseudocódigo o el diagrama de flujo no son comprensibles por la computadora, aunque pueda entenderlos cualquier programador. Un programa se escribe en un lenguaje de programación y las operaciones que conducen a expresar un algoritmo en forma de programa se llaman programación. Así pues, los lenguajes utilizados para escribir programas de computadoras son los

lenguajes de programación y programadores son los escritores y diseñadores de programas. El proceso de traducir un algoritmo en pseudocódigo a un lenguaje de programación se denomina codificación, y el algoritmo escrito en un lenguaje de programación se denomina código fuente (p.36).

Los lenguajes de programación más comunes se pueden observar en la tabla 1

Tabla 1: Lenguajes de programación más utilizados 2016

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

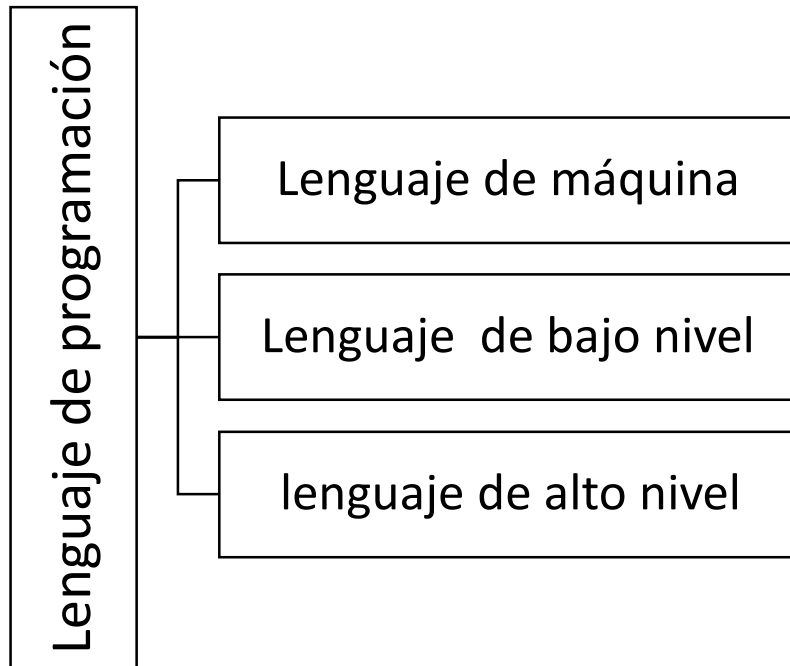
Fuente: IEEE Spectrum , 2017 <http://spectrum.ieee.org>

Los lenguajes de programación se clasifican en: Lenguaje de máquina, de bajo nivel y de alto nivel

El lenguaje de máquina es ingresado directamente al computador, es decir , el programador debía conocer los diferentes códigos binarios, comandos instrucciones y posiciones donde almacenar datos ejemplo: Si un programador quería sumar dos valores tendría que saber que el código binario de la instrucción suma es:

000100000000000000000000000000001011000

Figura 4: Clasificación de los lenguajes de programación



Su programación es muy complicada, fue creado para controlar los circuitos electrónicos del procesador por ejemplo: lenguaje assembler. La ventaja es que no necesita de un programa traductor debido a que comunica sus órdenes al computador directamente, logrando que el tiempo de proceso (si no hay errores) sea rápido en comparación con los otros lenguajes.

Lenguaje de bajo nivel llamados también ensambladores son lenguajes que utilizan palabras nemotécnica en lugar de cadenas de bit o códigos binarios. Toma las instrucciones de las personas y los ensambla a lenguaje de máquina para que pueda realizar instrucciones de alto nivel.

Tienen ventajas sobre los lenguajes de máquina debido a que ahorran tiempo y requieren menos atención y detalle. Se logra tener menos errores y la localización de estos es sumamente sencilla, aunque el procesamiento de datos es lento y el lenguaje para cada máquina es diferente es decir se tendría que codificar los programas nuevamente para cada computadora de diversa máquina, ejemplo:

Nombre Del Comando	Símbolo Nemetécnico
Multiplicar	m
Pensión	ps
Meses	ms

Código nemotécnico

00111	0011101	110101
m	ps	ms

Esto significa multiplicar (m) la pensión (ps) por los meses (ms). Para la programación de este lenguaje se necesita saber programar y tener los conocimientos sobre los componentes electrónicos del computador.

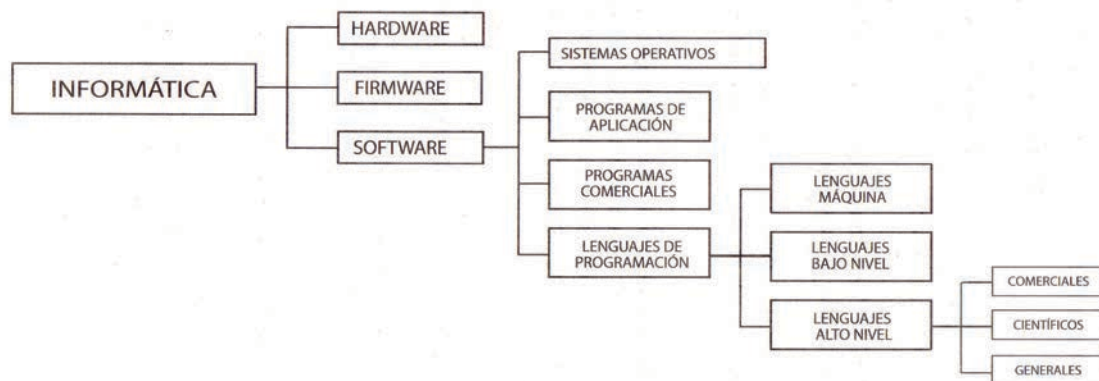
Lenguaje de alto nivel son sencillos de programar ya que le proporciona al usuario o al programador instrucciones fáciles de entender y memorizar. Logran independencia de la máquina permitiendo utilizar un mismo lenguaje diferentes computadoras de distintas marcas con la única condición de disponer de un programa traductor. En estos lenguajes sus instrucciones, comandos son más fáciles de aprender, se puede escribir de una manera rápida, sus errores son mínimos y el procesamiento de datos es más lento en comparación con el lenguaje de máquina. Entre los lenguajes más conocidos tenemos: BASIC, Pascal, COBOL. Se clasifican según su utilidad en: comerciales, científicos y generales.

Autores como Deitel y Deitel (2014) consideran:

Para agilizar el proceso de programación se desarrollaron los lenguajes de alto nivel en donde podían escribirse instrucciones individuales para realizar tareas importantes. Los lenguajes de alto nivel, como C++, Java, C# y Visual Basic nos permiten escribir instrucciones que son muy similares al inglés y contienen expresiones matemáticas de uso común. Los programas traductores llamados compiladores convierten los

programas que se encuentran en lenguaje de alto nivel a programas de lenguajes de máquina (p.10) (Deitel & Deitel, 2014)

Figura 5: Mapa conceptual del lenguaje de alto nivel



Lenguajes comerciales son aquellos que se utilizan en el campo de los negocios para aplicaciones de las empresas. Los más conocidos son RPG, Cobol, Visual Basic, UNIX.

Lenguajes científicos son aquellos que se utilizan para problemas de cálculos matemático, así como en el campo de la ingeniería. Entre los más conocidos se tienen: Fortran, Pascal, C.

Lenguajes generales son aquellos de fácil manejo y se los utiliza comúnmente en el campo de la educación y por programadores novatos. Entre los más conocidos se tienen: BASIC, LOGO

Actividad # 1

1.- Conteste las siguientes preguntas.

¿Qué es un Bit?

¿Qué es un Byte?

¿Cómo entiende la Información el computador?

¿Qué es un Software?

2. - Escriba (F) falso o (V) verdadero

El código Binario es la combinación de los números: 0, 1, 2 ()

3.- Escriba el nombre de 5 Lenguajes de Programación

4.- Investigación

Transforme en mediante el Código ASCII su Apellido en minúscula

¿Cuál es la diferencia entre un lenguaje de alto nivel y el lenguaje de máquina?

Capítulo 2

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Conocer el campo en que desarrolla la materia programación, así como su importancia.
2. Conocer los pasos para resolver un programa
3. Diferenciar entre dato e información
4. Desarrollar un algoritmo básico.
5. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.

Programa

Conjunto de instrucciones en secuencia lógica escritas en un lenguaje de programación regidas por normas, creados con la finalidad de ser un vínculo entre el usuario o el programador y el computador. En este sentido Joyanes (2008) expresa que Un programa de computadora “es un conjunto de instrucciones —órdenes dadas a la máquina— que producirán la ejecución de una determinada tarea. En esencia, un programa es un medio para conseguir un fin. El fin será probablemente definido como la información necesaria para solucionar un problema”(p.84).

Programación.- Conjunto de técnicas, Métodos (diseño, escritura y prueba) para desarrollar un programa.

Programador.- Persona encargada de crear y escribir programas mediante lenguajes de programación, que cumplan las necesidades del usuario.

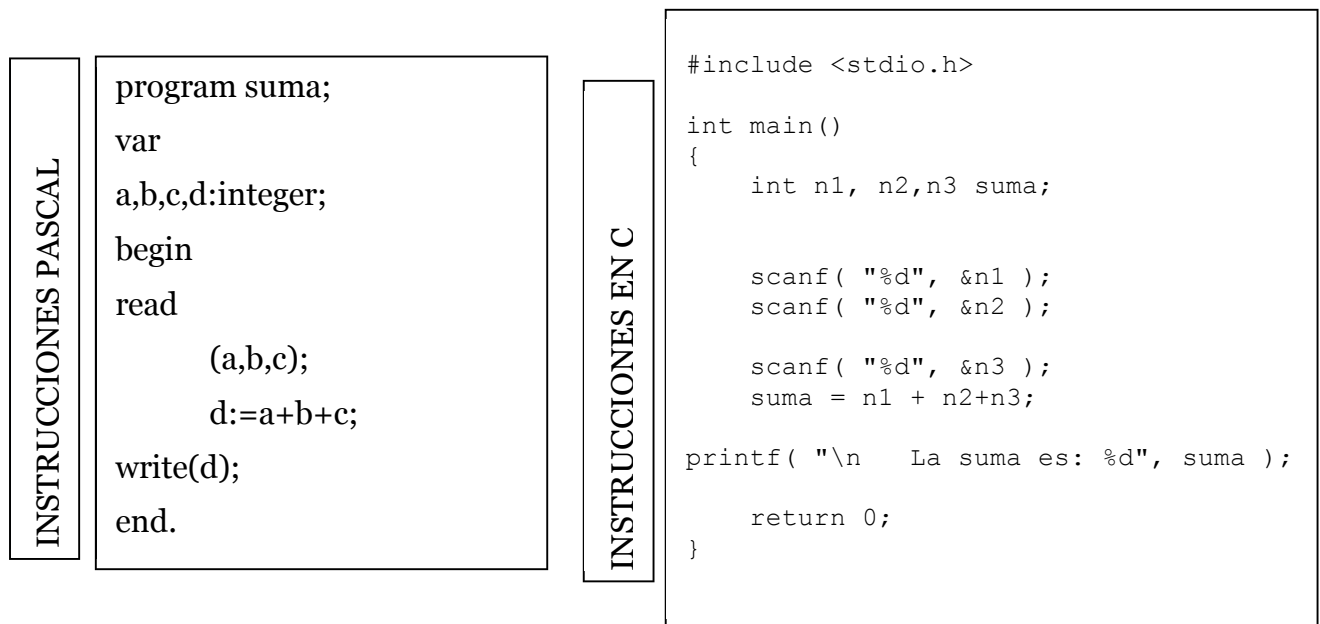
Instrucción.- es la formulación de una orden escrita en los programas que se manifiesta en la expresión del conjunto de operaciones que debe ejecutar el computador. Para nuestro estudio instrucción y sentencia tienen el mismo significado por ejemplo: Instrucciones en lenguaje de bajo nivel SUM, RES, DIV. SUM A,B,P significa sumar A mas B y almacenar el resultado en P. Instrucción en lenguaje de alto nivel: +, -, í, donde $P = A + B$ significa sumar A mas B y almacenar el resultado en P .

Los programas se clasifican en: programa objeto y programa fuente

Programa objeto

Conjunto de instrucciones escrito en lenguaje de máquina de codificación complicada que origina la utilización de mucho tiempo para la programación y se incurre en mayor cantidad de errores .Además el programador comunica directamente las instrucciones al computador sin intermediario.

Figura 6: Programas para calcular la suma de tres números enteros en lenguaje Pascal y C



Programa fuente

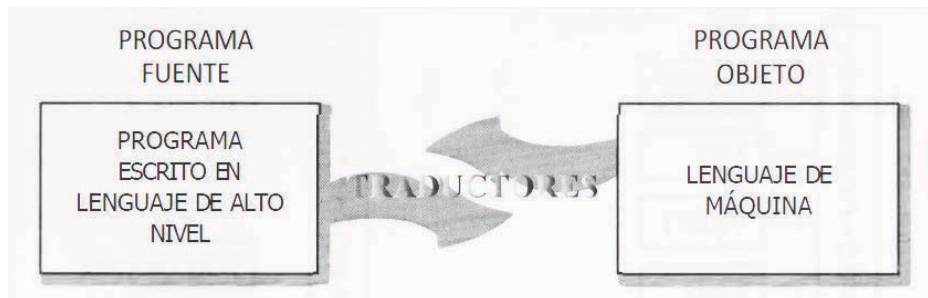
Conjunto de instrucciones que se encuentran escritas en lenguaje de alto nivel o bajo nivel, ahorran tiempo y requiere menos atención a los detalles y tiene la ventaja de ser fáciles de modificar pero su procedimiento es lento. Son los más utilizados para la programación por su sencillez.

Los programas fuentes no se comunican directamente con el computador, necesitan un nexo para que el computador los entienda por lo que se crearon los traductores.

Traductores

Es un programa que ayuda a establecer una transformación de los símbolos del programa fuente a código binario y viceversa, para que el computador pueda entender las instrucciones que se mandan a ejecutar. Se clasifican en: programas ensambladores, programas compiladores y programas intérpretes

Figura 7: transformación de los programas fuentes objeto



Programas ensambladores son programas encargados de transformar los programas fuente escritos en lenguaje de bajo nivel a programa objeto (lenguaje de máquina) para que el computador pueda ejecutar las instrucciones.

Programas compiladores se encarga de traducir el programa fuente a programa objeto, produce un listado de errores al terminar de ejecutar el programa, que tienen que ser corregidos. Su traducción es rápida. Entre los más conocidos tenemos: Pascal, FORTRAN, C, COBOL, etc.

Programas interpretes se encargan de Transforma el programa fuente a programa objeto pero a diferencia del compilador corrige los errores inmediatamente instrucción por instrucción obligando a iniciar de nuevo por cada error ,hasta que se corrija totalmente y así poder ejecutar el programa. Su traducción requiere mayor tiempo Entre los más conocidos tenemos: BASIC, LOGO, etc.

Clasificación de los programas fuentes escritos en lenguajes de alto nivel

Los Lenguajes de Alto Nivel según su estructura se clasifican en: programas no estructurados, programas estructurados, programas orientados a objeto y programación modular

Programas no estructurados soluciona pequeños y medianos problemas en programación. Son fáciles de aprender debido a su programación intuitiva y de aspecto personal por ejemplo: BASIC.

Programas estructurados apareció aproximadamente en la década de los 70 aplicando técnicas que permiten la escritura de programas de una manera fácil de leer y modificar, además requieren de conocimiento técnico básico para escribirlos. Entre los lenguajes estructurados tenemos: Pascal, "C", LOGO, ALGOL.

Haciendo una analogía con la construcción de un casa en relación con los programas no estructurados y estructurados, para construir de una manera sencilla una casa, no se tiene necesidad de un plan, se trabaja de una manera natural (lenguajes no estructurados) pero para construir un edificio tendríamos que elaborar buenas bases mediante un plan y seguir normas de construcción, para tener éxito en nuestro objetivo (lenguajes estructurados).

Muchos de los lenguajes que conocemos en la actualidad nacieron como no Estructurados y luego se estandarizaron y mejoraron, logrando convertirse en lenguajes estructurados

Programas orientados a objeto presentaron nuevos avances en la programación, en lugar de escribir numerosas líneas de instrucciones para describir la apariencia y la ubicación de los elementos de la interfaz (modo gráfico de la presentación de un programa), simplemente puede elegir de un grupo de opciones y colocar el objeto prefabricado. La programación se reduce significativamente por ejemplo: Visual Basic, Visual Fox, Visual C, etc.

Programación modular es una técnica usadas para dividir por partes el programa con fin de desarrollar subprogramas que se realizan independientemente para luego unirlos y alcanzar nuestro objetivo

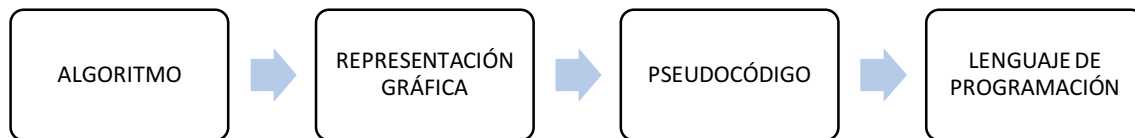
Técnicas para resolver un programa

La resolución de un problema exige por lo menos los siguientes pasos:

- 1.- Planteamiento del Problema.
- 2.- Algoritmo.

- 3.- Representación gráfica del algoritmo.
- 4.- Pseudocódigo.
- 5.- Lenguaje de programación.
- 6.- Depuración y verificación (corrección de errores).
- 7.- Documentación.
8. -Mantenimiento.

Figura 8: Paso básicos para construir un programa



Planteamiento del problema

Consiste en visualizar en la mente la idea la forma de cómo se resuelve el problema, sabiendo que elemento van a intervenir en el ingreso, en el proceso y en la salida, para luego poder expresarlo en el papel en la forma de algoritmo.

Algoritmo

Un algoritmo es una técnica para resolver un problema mediante la descripción no formal de cada uno de los pasos que se tienen que realizar de una manera secuencial y lógica, expresado en forma escrita, para luego poder transformarlo en una representación gráfica, pseudocódigo o lenguaje de programación. Entre las características que debe tener un algoritmo se pueden mencionar:

Debe ser preciso e indicar el orden de realización de cada paso

Después de analizarlo, debemos definirlo.

Debe tener fin (no quedarse en un ciclo).

Debe ser secuencial y coherente.

Solo puede ejecutar una operación a la vez.

Con respecto a la clasificación de los algoritmos, según su utilidad pueden ser: lógicos, aritméticos y se usó general

Algoritmos lógicos.- Son aquellos que utilizan lógica natural del ser humano para lograr resolver problemas.

Ejemplo #1

Un hombre desea transportar un zorro, un pollo y un maíz de una orilla a otra de un río Se dispone de una canoa, que solo soporta el peso del hombre y de alguno de los tres elementos mencionados. ¿Cómo transportarlos ?Si jamás en ninguna de las dos orillas deben estar solos ni el zorro con el pollo, ni el pollo con el maíz

Resolución:

INICIO

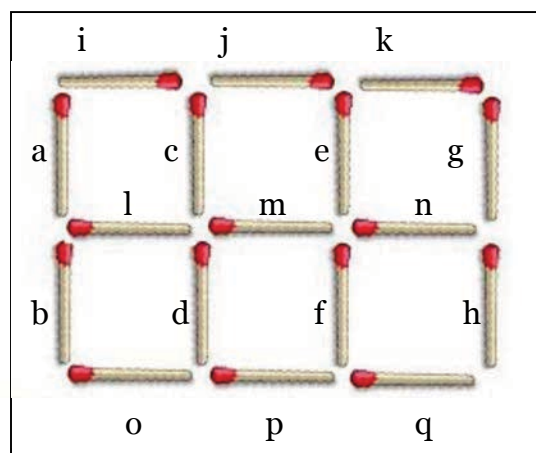
- 1.-Pasa el pollo, regresa solo el hombre.
- 2.-Pasa el zorro y regresa con el pollo.
- 3.-Pasa con el maíz y queda el pollo.
- 4.-Regreso por el pollo
- 5.-Pasa el pollo.

FIN

Ejemplo #2

De la figura 8 quitar 6 de los 17 palillos de modo que queden dos cuadrados

Figura 8: ejemplo 2



Resolución:

INICIO

Borramos el palito i

2.-Borramos el palito a

3.-Borramos el palito m

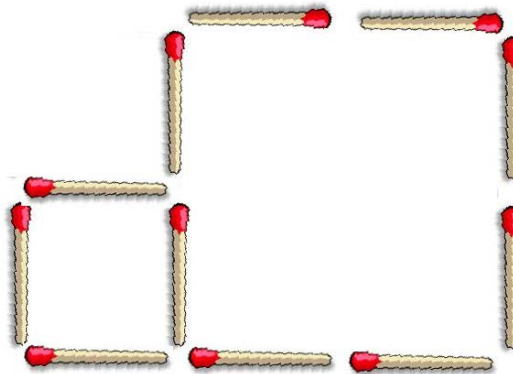
4.-Borramos el palito e

5.-Borramos el palito n

6.-Borramos el palito f

FIN

Figura 9: Solución del ejercicio 2



Algoritmos aritméticos.- Son aquellos que expresan matemáticamente un problema.

Ejemplo

Realizar un algoritmo para sumar dos números y presentar su resultado y finalizar.

Resolución:

1.-Inicio.

1. -Ingreso dos números 1,2.

3.-Proceso $C=1+2$.

4. -Presentar $C=3$.

5.-Fin.

Algoritmos de uso general.- Son aquellos que se emplean para resolver problemas que se presentan a diario en la vida por ejemplo: Realizar un algoritmo

que describa como se realiza el proceso de llenar el tanque de gasolina de un carro en una gasolinera

Resolución:

- 1.- Llegar a la gasolinera
- 2.- Elegir el tipo y la cantidad de gasolina
- 3.- Sacar la tapa del tanque de gasolina
4. -Apagar el carro
- 5.- Colocar la manguera en la boquilla del tanque
- 6.-Llenar tanque
7. -Sacar la manguera
- 8.-Poner la tapa
- 9.-Pagar
- 10.- Prender el carro

Información y datos

La información es el conjunto organizado y procesado de datos y los datos son la mínima parte de la información que se puede procesar en diferentes formas. En forma general las computadoras trabajan en base de datos que son ingresados por el ser humano de una forma manual o automática mediante diferentes tipos de periféricos de entrada y (teclado, lectora de barras, etc.).

Ejemplo:

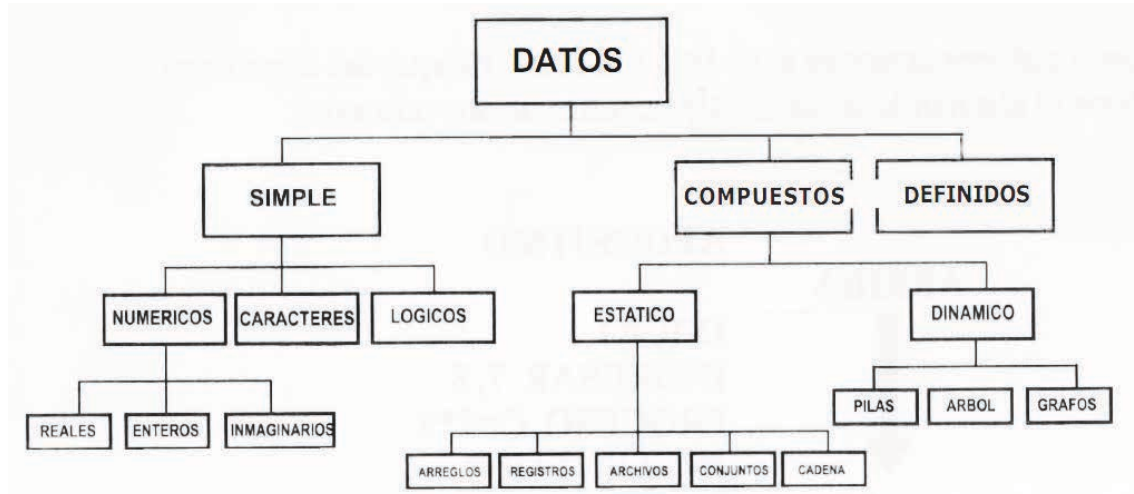
En el comisariato antiguamente la cajera tenía que ingresar por teclado todos los valores correspondientes a las ventas, era un poco tedioso y demoraba la atención al usuario pero en la actualidad se está utilizando la lectora de barra que realiza el ingreso de estos datos de una forma automática y veloz.

En programación básica dependiendo el problema que se presenta se ingresara el tipo de dato más adecuado por ejemplo: la suma de dos números da la idea de datos numéricos ejemplo: 5 , 7 . Un programa para ingresar los nombres

de los alumnos de un colegio da la idea de datos tipo carácter por ejemplo: BILL GATES

Para un mejor estudio los datos se dividen de la siguiente manera: datos simples, compuestos y definidos.

Figura 10: Clasificación de los datos



Datos simples

Son aquellos datos que carecen de estructura convirtiéndose la base para los datos estructurados (compuestos). Se pueden dividir en datos simples: numéricos, carácter y lógicos.

Datos simples Numérico

Es el conjunto de números que se pueden expresar de tres formas: entero, real e imaginario

Datos numéricos tipo entero.-Son números que pueden tener signo negativo y positivo, su única forma fraccionaria es tener en el denominador (parte inferior de la fracción) el número uno y no poseen puntos decimales ejemplo:

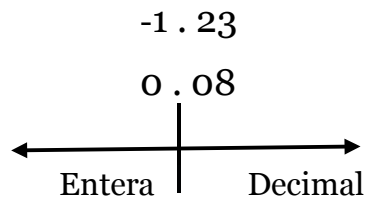
5.1, $\frac{1}{4}$, $\frac{1}{2}$, 2.5

Enteros incorrectos

$\frac{8}{1}$, 5, 7

enteros correctos

Datos numéricos real.-Son números que tiene punto decimal y pueden ser negativos y positivos. Consta de una parte entera y decimal por ejemplo:



Datos numéricos imaginativo.-Son datos que tienen un comportamiento especial debido a que no se encuentran en la recta numérica por ejemplo:

$\sqrt{-2}$ no se podría resolver mediante los números reales, por lo tanto recurrimos a los imaginarios

$$\sqrt{-2} = \sqrt{2}i$$

Datos simples carácter

Es el tratamiento de datos en forma individual o en conjunto pudiendo ser Números, letras o símbolos especiales por ejemplo:

Números

1, 2, 3, 4, 5, 6, 7, 8,9

Letras

A,B,C,D,E,F,G,H,.....,Z

Especiales

1 * _-+, «

Datos simples lógico

Se utilizan para representar alternativas de verdadero falso, si o no

Datos compuestos

Su labor es representar mediante una variable múltiples datos individuales a diferencia de los datos simples que una variable representa un elemento.

Se clasifican en datos compuesto estático y dinámico.

Datos definidos

Son definidos por el usuario, usualmente ubicados en la cabecera de los programas

Normas para diseñar algoritmos

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan, así como de la computadora que los ejecutó. Para este estudio los algoritmos aritméticos son de mayor utilidad. Para diseñar los algoritmos se adoptan ciertas normas.

El algoritmo tiene un inicio y tiene un fin.

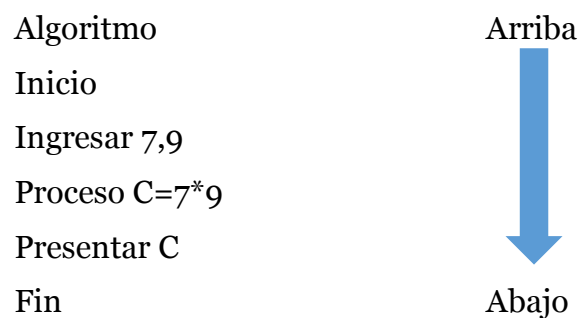
Al comenzar un algoritmo se escribirá la palabra INICIO.

Al terminar el algoritmo se escribirá la palabra FIN.

Tratar de escribirlo de arriba hacia abajo y cuando se presenten decisiones de izquierda a derecha.

El algoritmo se puede dividir en tres partes importantes: ingreso, resolución y salida. Algunos autores conocen esta división como bloque del algoritmo

Realizar el algoritmo de la multiplicación de dos números.



Actividad # 2

1. Conteste las siguientes preguntas

Establecer la diferencia entre Programa, programación y lenguaje de Programación

¿Qué es un programa estructurado?

¿Qué es un lenguaje de alto nivel?

Diferencia entre algoritmo lógico y aritmético





2. Escriba (F) falso o (V) verdadero

Un Programa se puede escribir con solo un Lenguaje de programación ()

El compilador es un máquina ()

3. Utilizando la figura 9 describa los paso mediante un algoritmo lógico para solucionar el problema para desplazarse sin distracciones desde el trabajo hasta su hogar.

Figura 9: ejercicio 3 de la actividad 3

			
			
INICIO 			

4. Conteste las siguientes preguntas

¿Qué ingresamos por el teclado de un computador datos o Información?

Explique su respuesta

¿Qué es un dato Simple?

5. Subraye la respuesta correcta

¿Cuál es datos de tipo Entero?

10, 10.1 9.1/1 7/2 7 -7

¿Cuál es datos de tipo Real?

10.1 10.1 9.1/1 7/2 7 -7

Capítulo 3

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Analizar los diferentes elementos que intervienen en al estructurar un programa
2. Tener un criterio claro sobre variable, constante e instrucción.
3. Manejar los diferentes tipos elementos analizados en la unidad.
4. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad

Elementos fundamentales para diseñar un programa

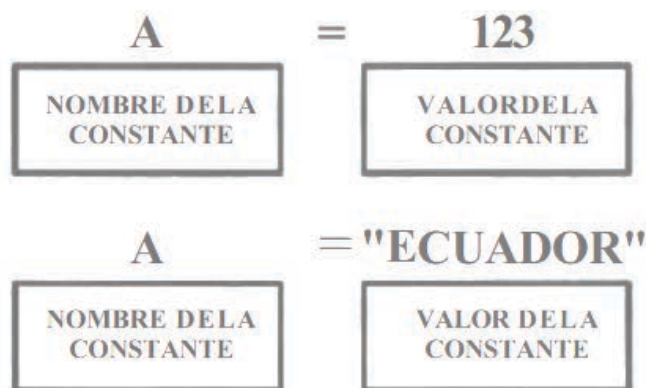
Los elementos fundamentales para diseñar un programa son: caracteres, constantes, variables, instrucciones y expresiones.

Constantes

Una constante es la asignación de un valor de datos numéricos o no numéricos que se mantiene sin cambio (no varía) durante el desarrollo y la ejecución de un programa. Para escribir de forma correcta la expresión de una constante debe estructurarse de la siguiente manera:

Nombre de la constante
Valor de la constante
Operador

Figura 10: Estructura de la expresión constante



Nombre de la constante

Son datos tipo numérico, carácter o combinados en los cuales se almacena valor de la constante. Su ubicación se encuentra en la parte izquierda de la expresión. El nombre de la constante puede ser: alfabética o alfanumérica

Nombre de la Constante alfabética.- Representada por los caracteres alfabéticos desde la letra A hasta la Z en mayúscula minúscula o combinaciones. Por ejemplo:

A= 12 DE8E= 23323 a= 12 Debe= 34344
gasto= 1455 Debe = "MOROSO"

Nombre de las Constantes alfanuméricas.- Representadas por las letras del alfabeto en combinación con números o caracteres especiales. Los nombres de las constantes alfanuméricas siempre comienzan con una letra
Por ejemplo:

A\$= "Francisco" Al = "TOTAL" A8= 12 QWER= 12

Algunas sugerencias para escribir los nombre de constantes:

No se permiten utilizar signos aritméticos por ejemplo:

A-8= 0 A+8= 3

No se permite espacios entre las letras por ejemplo:

A 8= 12 ORO SOL= 34

No se permite que el primer carácter sea numérico por ejemplo:

8= 23 28N= 45 123V8= 34

No se permite solo números

23= 234 56= 344

Valor de la constante.-

Representa un valor que se almacena en el nombre de la constante se mantiene invariable en el desarrollo del algoritmo o programa se clasifica el valor de la constante en: numéricas y no numéricas

Las constantes numéricas son la asignación de datos numéricos que se mantienen fijos en el programa y que pueden tener signo o no por ejemplo:

A= 56 B= 9.9 C=-34 D= +34

También se pueden expresarse en notación científica

R= 2.45E4 es decir R= 2.45* 10⁴ o R= 24500

Constantes no numéricas son la asignación de datos tipo carácter. Los datos tipos carácter se encierran entre apóstrofes o dobles comillas, por ejemplo:

A= "R" B= "UNWERSIDAD" C= "Año 2000" D= "34343"

Los datos tipo lógico se pueden representar mediante las palabras

A=verdadero F=falso

Variables

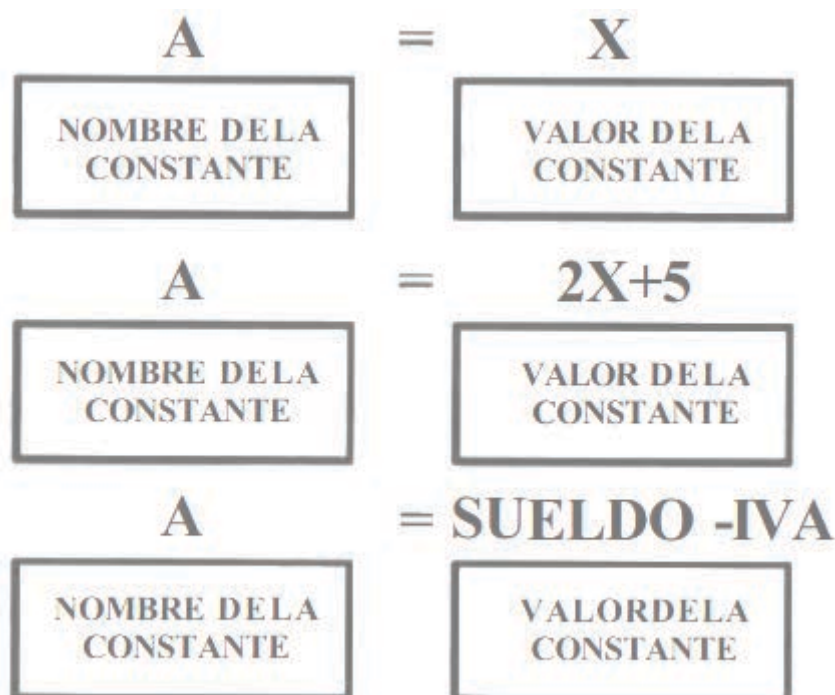
Las variables son valores que cambian a medida que se desarrolla automáticamente nuevos datos en el algoritmo o programa. Según Mata y Cushman (2001) conceptualizan a las variables como “una ubicación de memoria cuyo contenido puede ser rellenado o cambiado durante su ejecución de un programa” (p.75). Para escribir de una forma correcta la expresión de la variable debe estructurarse de la siguiente manera:

Nombre de la variable

Valor de la variable

Operador

Figura 11: Estructura de la expresión variable



Nombre de la variable.-

Son datos tipo numérico, carácter en los cuales almacenamos el valor de la variables .El nombre de la variable puede ser:

Alfabética

Alfanumérica

Con subíndice

Nombre de la variable alfabética.-Representada por los caracteres alfabéticos desde la letra A hasta la Z en mayúscula, minúscula o combinaciones.

Por ejemplo:

A=12 a=12 DEBE=23323 Debe34344
gasto=1455

Nombre de la variable alfanuméricas.-Representadas por las letras del alfabeto en combinación con números o caracteres especiales. Permiten almacenar valores variables numéricos y cadenas de caracteres, por ejemplo:

A\$Francisco Al=45 A C=3434 LEONF=23

Nombre de la variable con subíndice.-Son datos compuestos que se almacenan en la memoria del computador. Se utiliza una variable y el subíndice que se ubica entre paréntesis .Se clasifican en tres tipos de nombre de variables:

Unidimensional	A(i)
Bidimensional	A(i, j)
Tridimensional	A(i,j,k)

Algunas sugerencias para escribir los nombre de variables:

Los nombres de la variable alfanumérica siempre comienzan con una letra

No se permiten utilizar signos aritméticos por ejemplo:

A/BO A*B=3

No se permite espacios entre las letras por ejemplo:

AB C=12 RICK MARTIN=34

No se permite que el primer carácter sea numérico por ejemplo:

1B=23 2BN=45 123 VB=34

Valor de la variable

Es la cantidad que representa la variable en un determinado momento en el transcurso del programa. Se escribe en la parte derecha de la expresión. Su estructura tiene mínimo un dato alfabético o alfanumérico que puede combinarse con números, letras y operadores aritméticos, etc.

Ejemplo: A=A+1 A=2B A=X+2R X=Y

Instrucción

Es la formulación de una orden al computador que se manifiesta en la expresión del conjunto de operaciones que debe ejecutar el computador. Para nuestro estudio instrucción y sentencia tienen el mismo significado. Por ejemplo:

Instrucción en lenguaje de alto nivel: +, -, i

P=A +B significa sumar A mas B y almacenar el resultado en P

Los algoritmos pueden tener básicamente las siguientes tipos de instrucción:

INICIO

ENTRADA

PROCESO

PRESENTAR

DECISIÓN

FIN

Instrucción inicio/fin.- Norma adoptada para dar a conocer donde comienza el algoritmo mediante la palabra INICIO y FIN al terminar.

Instrucción de entrada.- Indica todos los datos que van a intervenir en el algoritmo, ingresados mediante periféricos de entrada (teclado, mouse, etc.). Es representada por la palabra INGRESAR seguido del nombre de variable o constante que se necesita ingresar en el algoritmo.

FORMATO: INGRESAR

Por ejemplo: INGRESAR A

En el caso de ingresar más de dos datos entonces las variables deberán ir separadas con coma ejemplo:

Por ejemplo: INGRESAR A,B,C,D

Ejercicio #1

Se desea ingresar el número de estudiantes y bancas de un colegio entonces:

Resolución:

Número de estudiantes=A

Número de bancas =B

ALGORITMO

1.-INICIO

2.-INGRESO A, B

3.-FIN

Ejercicio #2

Realizar un algoritmo para ingresar tres números

Resolución

Primer número =A

Segundo número =B

Tercer número =C



ALGORITMO

- 1.-INICIO
- 2.-INGRESAR A,B,C
- 3 –FIN

Instrucción de proceso.-Es la encargada de realizar todas las operaciones aritméticas y lógicas. Es representada por la palabra PROCESO seguido de la expresión variable o constante que se necesita procesar en el algoritmo.

FORMATO:

PROCESO

Ejemplo : PROCESO $A=B+C$

Ejercicio 1

Realizar un algoritmo para sumar tres números

Resolución:

Primer número =A

Segundo número =B

Tercer número =C

ALGORITMO

- 1.-INICIO
- 2.-INGRESAR A,B,C
- 3-PROCESO $D=A+B+C$
- 4-FIN

Instrucción de presentar.-Visualiza mediante algún periférico de salida (monitor, impresora, etc.) el resultado parcial o total del algoritmo.

Es representada por la palabra PRESENTAR seguido de la expresión variable o constante que se necesita mostrar en el algoritmo.

FORMATO:

PRESENTAR

Si son varios datos que se necesiten presentar, estos serán separados con coma
por ejemplo: PRESENTAR A,B,C,D

Ejercicio#1:

Realizar un algoritmo para sumar tres números.

Resolución

PRIMER NÚMERO =A

SEGUNDO NÚMERO=B

TERCER NÚMERO=C



ALGORITMO

Si A=2 , B=3, C=6

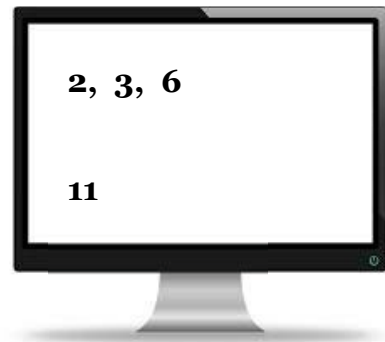
1.-INICION

2.-INGRESO A,B,C

3.-PROCESO $D=A+B+C$

4.-PRESENTARD

5.-FIN



Ejercicios #2

Realizar algoritmos para presentar los resultados de la tabla de multiplicar del número 4 hasta el número 5 con constantes (sin utilizar variables).

ALGORITMO

1-INICIO

2.-PROCESO $C=4*1$

3.- PROCESO $D=4*2$

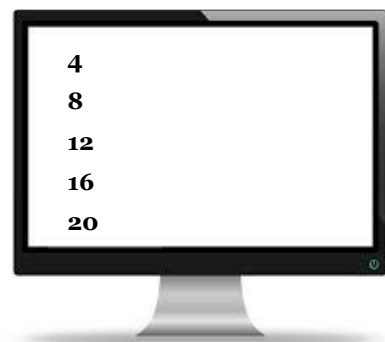
4.- PROCESO $E=4*3$

5.- PROCESO $F=4*4$

6.- PROCESO $G=4*5$

7.- PRESENTAR C,D,E,F,G

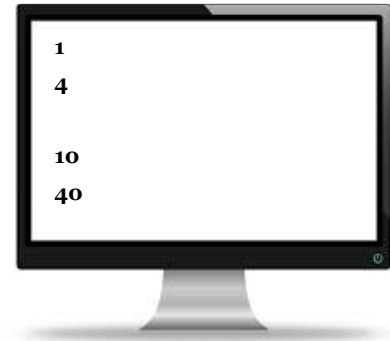
8.-FIN



ALGORITMO

- 1.-INICIO
- 2.-PROCESO $C=4*5$
- 3.-PRESENTAR C
- 4.-FIN

- 1.-INICIO
- 2.-INGRESAR A
- 3.-PROCESO $C=4* A$
- 4.-PRESENTAR C
- 5.-FINALIZAR



Este algoritmo trabajará de la siguiente manera. Se ingresará un número (1) en la variable A, luego lo procesará en el interior del computador ($4*1$) luego de tener el resultado lo mostrará (4) y finalizará y se borrará los primeros datos ingresado, quedando el algoritmo listo para poder ingresar los siguientes datos en este caso el número (2) luego lo procesa ($4*2$) presenta el resultado (8) y finaliza y se borrará los segundos datos ingresados, quedando el algoritmo listo para poder ingresar los siguientes datos y así sucesivamente.

Ejercicio #3

Un profesor de matemáticas desea que realizar un algoritmo con constantes para multiplicar las siguientes cantidades constantes:

$$12 * 134555$$

$$150089 * 3456$$

$$0.251 * 0.3564$$

$$12345 * 12345$$

Por obvias razones la multiplicación manual resulta muy tediosa y se puede producir algún error así que realizaremos los correspondientes algoritmos para resolverlo de manera ágil, veloz y correcta

Resolución

ALGORITMO

1.-INICIO

2.-PROCESO $C=12 * 134555$.

PROCESO $D= 150089 * 3456$

PROCESO $E=0.2516546 * 0,3456 3$

PROCESO $F=12345 * 12345$

3 -PRESENTAR C ,D,E,F

4.-FIN

Realizar un algoritmo por cada cálculo es un gasto de tiempo, espacio de memoria. Imagínese si le manda a resolver mil operaciones tendría que realizar mil algoritmos con constantes. Para evitar esto vamos a utilizar en el ingreso variables y vamos a observar cuantas líneas de algoritmo se reduce el problema.

Resolución:

ALGORITMO

1.-INICIO

2.-INGRESA A,B

3.-PROCESO $C=A * B$

4 -PRESENTAR C

5-FIN

Este algoritmo resuelve de manera corta el problema.

12 * 134555



Se ingresan los dos datos y la variable C almacenaría el valor total de la multiplicación de $A * B$ entonces $C=1614660$, no es presentada hasta que se pide mediante una instrucción, al momento de ejecutar la instrucción PRESENTAR se mostrará el resultado en la pantalla o por la impresora del computador de la siguiente manera: 1614660 .A continuación se finaliza el algoritmo, luego se borrará automáticamente todos los valores almacenados y se pueden ingresar nuevos datos.

Actividad # 3

1.- Conteste a las siguientes preguntas

¿En qué tipo de Algoritmo se basará nuestro estudio?

¿Cuáles son los elementos fundamentales para diseñar su programa?

¿Cuáles la diferencia entre Variables y Constantes?

2.- Subraye las respuestas correctas.

¿Cuáles son constantes? Analizar el nombre y el valor de la constante

Constante = 1

A= 2B

12=34

Z= 2,3

A= "2B"

L= +2

¿Cuáles son nombres de variables? Analizar el nombre y el valor de la variable

A= A+ 1

2A= A+ 1

WORD.DoC= 34E

A\$ = Jo SE

A1 = 2

A(1,2) = 45

RES-RT= 34N

3 .- Realizar los siguientes ejercicios

Elabore un Algoritmo para sacar el 12% o del IVA de un valor ingresado, presentar el resultado y luego finalice.

FORMULA $B=A * 12/100$

Elabore un algoritmo para sacar el promedio de cuatro notas, presentar el resultado, luego finalice.

Realizar un Algoritmo para calcular automáticamente la siguiente formula, los valores numéricos son ingresados por teclado presente el resultado y finalice

$$\sqrt{\frac{m^2 + m + 1}{35 - s + s^2}}$$

Realizar un Algoritmo para calcular los valores de la tabla de sumar del 1 al 12 de cualquier número ingresado presentar los valores y finalizar

Capítulo 4

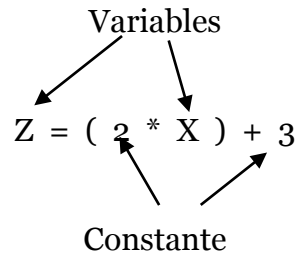
Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Analizar los diferentes tipos de expresiones
2. Elegir la expresión más apropiada para la resolución de su problema.
3. El alumno estará en la capacidad de desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad

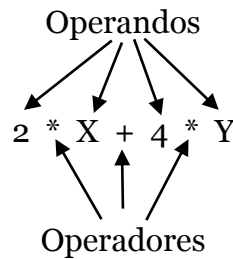
Expresiones

Son combinaciones de constantes variables, operadores, paréntesis o funciones especiales por ejemplo:



A las constantes y variables se denominan operandos

Los operadores son los símbolos que representan una operación que puede ser matemática, lógica, de relación y de carácter por ejemplo:



Pueden encontrar dos o más expresiones en una expresión general por ejemplo:

$$Z = \underline{(S+3X)} + \underline{(4K-6)} * \underline{(S+4X)}$$

EXP1 EXP2 EXP3

EXPRESIÓN GENERAL

Las expresiones se pueden clasificar en:

Expresiones aritméticas

Expresiones lógicas

Expresiones de carácter.

Expresiones aritméticas.

Es la combinación de variables, constantes, paréntesis o nombres de funciones especiales unidas mediante operadores Aritméticos. En esta operación el resultado será algún dato entero o real, según Corona y Ancona (2011) “un

operador es un símbolo que permite relacionar dos datos en una expresión y evaluar el resultado de la operación”(p.14)

Operadores aritméticos

Son símbolos o conjuntos de letras que representan una acción matemática dentro de una expresión aritmética. Además sirven para hacer operaciones aritméticas básicas (Corona y Ancona, 2011).

Tabla2: Operadores aritméticos

Símbolo tradicional	Símbolo del computador	Formato	Significado
+	+	A+B	Suma
-	-	C=A-B	Resta
x	*	C=A*B	Multiplicación
÷	/	C=A/B	División
A ^B	^, **	C=A ^ B	Exponenciación
	DIV	C= A DIV B	División entera
	MOD	C=A MOD B	Residuo
$\sqrt{\quad}$	SQR()	C=SQR(A)	Raíz cuadrada

Según el lenguaje de programación las expresiones pueden variar y no utilizar todos los operadores matemáticos numéricos.

Operador aritmético (/) Representa la operación real de la división y nos presenta el valor completo del cociente en la división por ejemplo: C=A/B

ejercicio: SI A=7 , B=3 entonces C=7 /3

$$7 \overline{) 3 \quad}$$

10 2.33 Cociente completo

10

1

$$\text{Respuesta } C = 7 / 3$$

$$C = 2.33$$

Operador aritmético (DIV) Representa la división entera, y nos presenta el valor entero del cociente por ejemplo:

$$C = A \text{ DIV } B$$

Ejercicio# 1

$$\text{Si } A=7, B=3$$

$$\text{Entonces } C = 7 \text{ DIV } 3$$

$$\begin{array}{r} 7 \overline{) 3} \\ 1 \quad 2 \end{array} \quad \text{Cociente entero}$$

$$\text{Respuesta } C = 7 \text{ DIV } 3 = 2$$

Ejercicio #2

$$\text{Si } A=23, B=12$$

$$\text{Entonces } C = 23 \text{ DIV } 12$$

$$\begin{array}{r} 23 \overline{) 12} \\ 11 \quad 1 \end{array} \quad \text{Cociente entero}$$

$$\text{Respuesta } C = 23 \text{ DIV } 12 = 1$$

Ejercicio #3

$$\text{Si } A = 0.5, B = 5 \text{ entonces } C = 0.5 \text{ DIV } 5$$

$$\begin{array}{r} 0.5 \overline{) 5} \\ 0 \quad 0 \end{array}$$

$$\text{Respuesta } C = 0.5 \text{ MOD } 5 = 0$$

Operador aritmético MOD Es la representación del residuo de la división entera por ejemplo: $C = A \text{ MOD } B$

Ejercicio# 1

$$\text{Si } A = 7, B = 3$$

$$\text{Entonces } C = 7 \text{ MOD } 3$$

$$7 \overline{) 3}$$

Residuo \rightarrow 1 2

Respuesta $C=7 \text{ MOD } 3 = 1$

Ejercicio #2

Si $A=23$, $B= 12$

Entonces $23 \text{ MOD } 12$

$$\begin{array}{r} 23 \overline{) 12} \\ 11 \quad 1 \end{array}$$

Respuesta $C=23 \text{ MOD } 12=11$

Ejercicio #3

SI $A= 0.5$, $B=5$ entonces $0.5 \text{ MOD } 5$

$$\begin{array}{r} 0.5 \overline{) 5} \\ 0 \quad 0 \end{array}$$

Respuesta $C= 0.5 \text{ MOD } 5 = 0$

Operador aritmético $^$ ****** Es la representación del símbolo de la exponenciación en las operaciones matemáticas por ejemplo: $C=A^B$

Donde A: Número base

B: Potencia o exponente

Consiste en multiplicar el número base, tantas veces nos indique la potencia para luego ser almacenado en una variable.

Ejercicio #1

SI $A=2$, $B=3$

Entonces $C=2^3=8$ o $2*2*2=8$

Reglas prácticas para la transformación de las expresiones

Permiten la conversión de una expresión algebraica (tradicional) en expresión aritméticas (símbolos que entiende el computador)

$$X = A^2 + \frac{2B}{C}$$

$$X = A^2 + (2 * B / C)$$

$$X = \frac{A * C + \sqrt{B}}{B}$$

$$X = ((A * C) + \text{SQR}(B)) / B$$

$$X = (a + b)^2$$

$$X = (a + b) ^ 2$$

$$X = \frac{-B^2 + \sqrt{(B^2 - 4AC)}}{2A}$$

$$X = [B + \text{SQR}(B^2 - (4 * A * C))] / 2 * A$$

Cuando se tienen un conjunto de operadores y operandos interactuando, se necesita seguir unos pasos para que la correcta resolución de la expresión aritmética mediante reglas matemáticas que dicen:

Regla 1. Las expresiones más internas entre paréntesis se resuelven primero, una vez resuelta seguimos descomponiendo la expresión hasta llegar al paréntesis externo.

Regla 2. Las expresiones se resuelven manteniendo el siguiente orden de jerarquía del computador:

- a.- Operadores exponenciales
- b. - Operadores de la multiplicación y la división
- c.- Operadores DIV y MOD
- d.- Operadores suma y resta

Regla3. Si hay dos o más operadores del mismo nivel se resuelven el primero que encontremos de izquierda a derecha.

Ejemplo#1

Encontrar el resultado de la siguiente expresión aritmética:

Si A=1 ,B=4, C=3

$$X = (-B + \text{SQR}(B^2 - 4 * A * C)) / 2 * A$$

Expresión 1



Expresión 2



Expresión 3



Expresión general



Se ingresa los valores dados

$$X = (-4 + \text{SQR}(4^2 - (4 * 1 * 3))) / 2 * 1$$

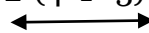
Se aplica las reglas para la resolución de expresiones algebraicas.

Expresión 1

Regla 1

Resolver las expresiones más internas entre paréntesis.

$$X = (-4 + \text{SQR}(4^2 - (4 * 1 * 3))) / 2 * 1$$



Regla 2

Exponenciación no

Multiplicación si

Resolver

$$X = (-4 + \text{SQR}(4^2 - 12)) / 2 * 1$$

Resuelta la primera expresión interna se aplica nuevamente las reglas

Expresión 2

Regla 1

$$X = (-4 + \text{SQR}(4^2 - 12)) / 2 * 1$$



Exponenciación si

Resolver

$$X = (-4 + \text{SQR}(16 - 12)) / 2 * 1$$

Luego

Regla2

Exponenciación no

Multiplicación no

Mod y div no

Suma y resta si

Resolver

$$X=(-4+\text{SQR}(4))/2*1$$

Resuelta la segunda expresión interna se aplica nuevamente las reglas

Expresión 3

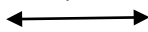
$$X=(-4+\text{SQR}(4))/2*1$$



Regla2

Exponenciación si

$$X=(-4+2)/2*1$$



Luego

Regla

Multiplicación no

Mod y div no

Suma y resta si

Resolver

$$X=(-2)/2$$

Resuelta la tercera expresión interna se aplica nuevamente las reglas

Expresión general

$$X=(-2)/2$$

Regla 2

Exponenciación no

Multiplicación y div si

Resolver

$$X = -1$$

Ejemplo 2

Transformar la siguiente expresión algebraica en expresiones aritmética y encontrar el resultado: Si A=140, B=4, C=3, K=100

$$X = \frac{\frac{A}{B+C}}{\frac{8+\sqrt{K}}{C^2}} * MOD B$$

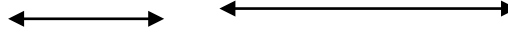
Transformando

$$X = ((A/(B+c)) / ((8+SQR(K)) / C ^ 2)) MOD B$$

Expresión 1



Expresión 2



Expresión 3

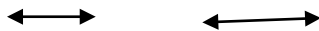


Expresión general



Se ingresa los valores dados

$$X = (140/(4+3) / ((8+SQR(100)) / 3 ^ 2)) MOD 4$$



Se aplica las reglas en las dos partes de la expresión 1

Expresión 1 parte 1

Regla 2

Exponenciación no

Multiplicación no

Mod y div no

Suma y resta si

Resolver

$$X = (140/7 / ((8+SQR(100)) / 3 ^ 2)) MOD 4$$

Expresión 1 parte 2

Regla2

Exponenciación si

resolver

$$X=(140/7/ ((8+10)/3^2))\text{MOD}4$$

Luego

Regla en la parte 2

Exponenciación no

Multiplicación no

Mod y div no

Suma y resta si

Resolver

$$X=((140/7)/(18/3^2))\text{MOD}4$$

Resuelta la segunda expresión interna se aplica nuevamente las reglas

Expresión 3

Regla 1

$$X=((140/7)/(18/3^2))\text{MOD}4$$



Se resuelven las dos partes de la expresión 3

Parte 1

Exponenciación no

Multiplicación y división si

Resolver

$$X=[(20)/(18/3^2)]\text{MOD}4$$

Luego parte 2

Exponenciación si

Resolver

$$X=((20)/(18/9))\text{MOD}4$$

Continuando con la parte 2

Exponenciación no

Multiplicación y división si

Resolver

$$X = ((20) / (2)) \text{MOD} 4$$

Resuelta la tercera expresión interna se aplica nuevamente las reglas

Exponenciación no

Multiplicación y división si

Resolver

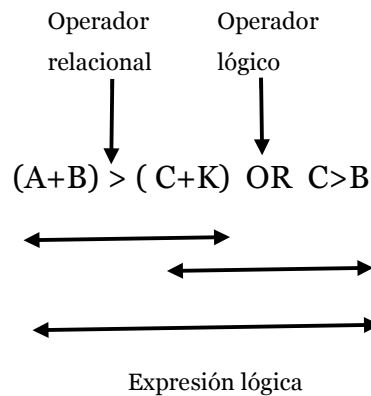
$$X = 10 \text{MOD} 4$$

Luego

$$X = 2$$

Expresiones lógicas

Son expresiones que pueden tomar dos valores verdadero y falso. Utiliza variables, constantes o expresiones aritméticas en combinación con los operadores relacionales y lógico, por ejemplo:



Formato: expresión operador expresión

Se puede combinar más de dos operadores

Formato: expresión operador expresión operador expresión

Por ejemplo: $(A = B + C) > (B = C * A) \text{ AND } (A = B - C)$

Operador relacional

Símbolos que establecen una relación como verdadera o falsa entre expresiones

Formato:

Expresión, variable o constante operador Expresión, variable o constante

Tabla 3: Operadores relacionales

OPERADORES	FORMATO	SIGNIFICADO
>	A>B	A MAYOR QUE B
<	A<B	A MENOR QUE B
<=	A<=B	A MENOR O IGUAL QUE B
>=	A>=B	AMAYOR O IGUAL QUE B
<>	A<>B	A DISTINTO QUE B

Por ejemplo:

Si A=1
B=2

A>B	B>A	A=A	A<=	A>=B
1>2	2>1	1=1	1<=2	1>=2
FALSO	VERDADERO	VERDADERO	VERDADERO	FALSO

Operador lógico

Conjunto de letras que representan una orden que entiende el computador, para dar un resultado de verdadero o falso.

Formato:

Expresión, variable o constante operador Expresión, variable o constante

Tabla 4: Operadores lógicos

Operador	Formato	Significado
NOT	NOT A	Negación
AND	A AND B	Que se cumpla A y B
OR	A or B	Que se cumpla A o B

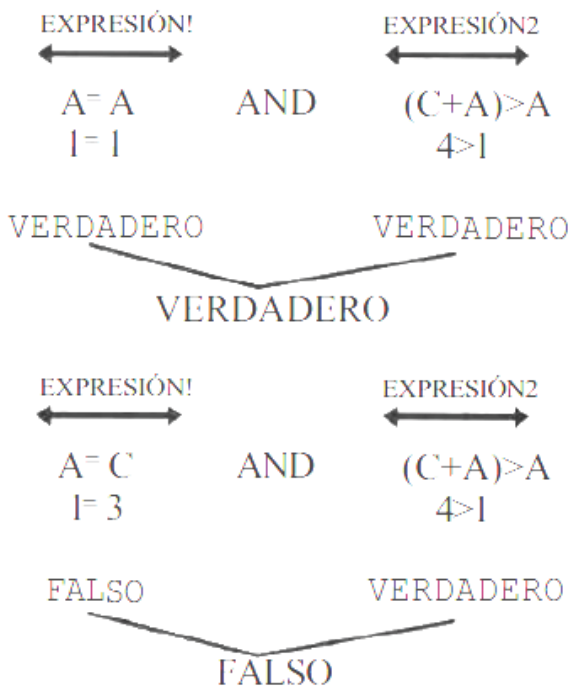
Operador lógico AND (Y) Logran establecer la unión entre las expresiones.

Si todas las expresiones utilizadas son verdaderas el resultado global es verdadero, caso contrario si una o varias son falsa el resultado es falso

Tabla 45: Operador lógico AND

Expresión 1	Expresión 2	Expresión 1 and expresión 2
Falso	Falso	Falso
Falso	Verdadero	Falso
Verdadero	Falso	Falso
Verdadero	Verdadero	Verdadero

Por ejemplo, si $A=1, C=4$



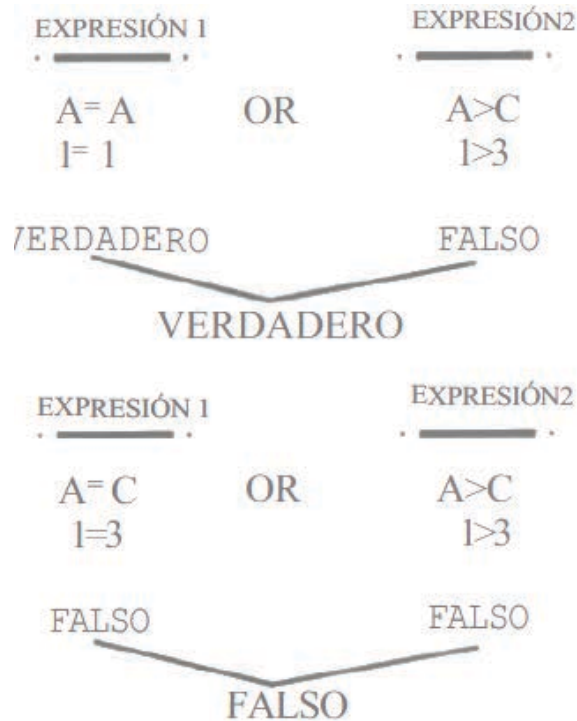
Operador lógico OR (O)-Logra establecer la unión entre expresiones. Si una de las expresiones es verdadera el resultado global es verdadero, caso contrario si todas son falsas el resultado es falso.

Tabla 6: Operador lógico OR

Expresión 1	Expresión 2	Expresión 1 OR expresión 2
-------------	-------------	----------------------------

Falso	Falso	Falso
Falso	Verdadero	Verdadero
Verdadero	Falso	Verdadero
Verdadero	Verdadero	Verdadero

Por ejemplo , si $A=1$, $B=2$

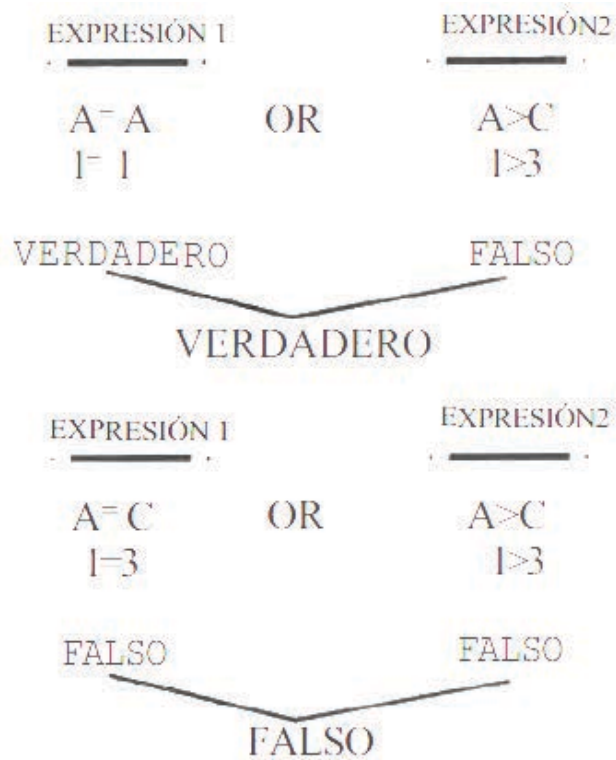


Operador lógico NOT.-Palabra que antecede a la expresión logrando invertir su valor lógico

Tabla 7: Operador lógico NOT

Expresión 1	NOT(Expresión)
Verdadero	Falso
Falso	Verdadero

Por ejemplo, si $A=4$, $B=2$



Representación gráfica de las expresiones lógicas y relacionales

Un modo práctico para entender como funcionan las expresiones es mediante de la recta numérica, observando la ubicación de los valores numéricos comprenderemos que datos son correctos e incorrectos

Figura 12 : Recta numérica

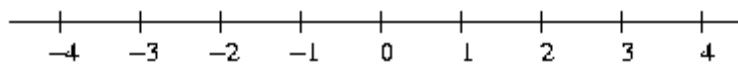
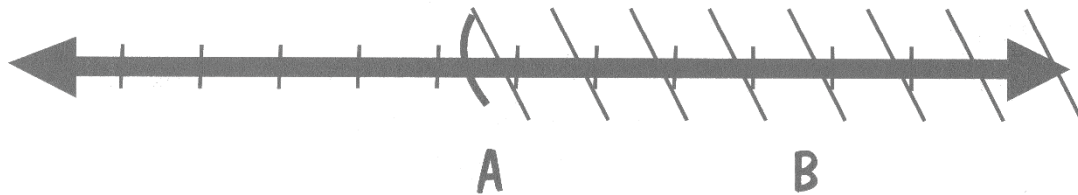


Tabla 8: Símbolos para la representación de la recta numérica

SIMBOLO EQUIVALENTES		SIGNIFICADO
\geq	[PARENTESIS RECTO INICIAL-Indica el punto donde comienza el rango solución, incluyendo el valor inicial
\leq]	PARENTESIS RECTO FINAL-Indica el punto donde termina el rango solución, incluyendo el valor final
$>$)	PARENTESIS CURVO INICIAL- indica el punto donde comienza el rango solución, no incluyendo el valor inicial
$<$)	PARENTESIS CURVO FINAL-Indica el punto donde termina el rango solución, no incluyendo el valor

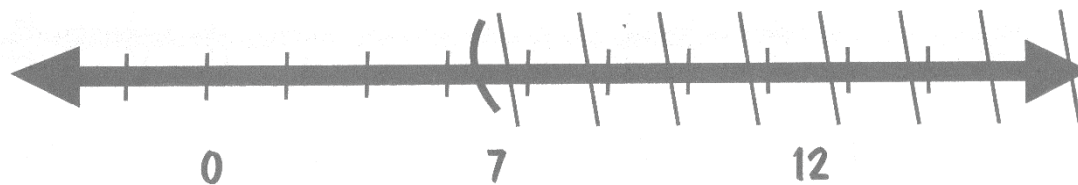
Ejemplo 1 : Si $A > B$ se expresa gráficamente



La interpretación de esta representación es la siguiente:

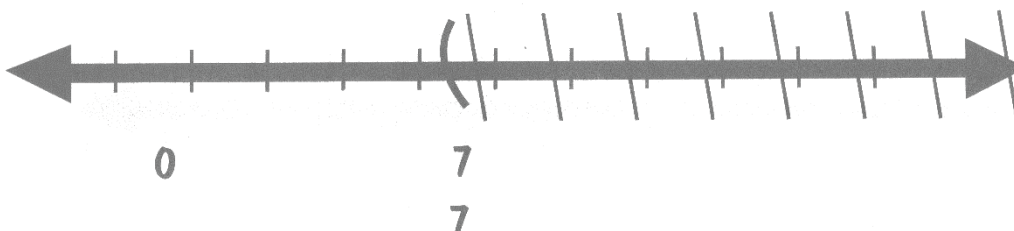
Todos los datos ingresados en la variable A serán verdaderos si se encuentran dentro del rango (zona rayada) que comienza a partir desde B sin incluir su valor, es decir, A es mayor y no igual que B.

Ejemplo 2: Si $A = 12$, $B = 7$, $A > B$ entonces la condición es verdadera

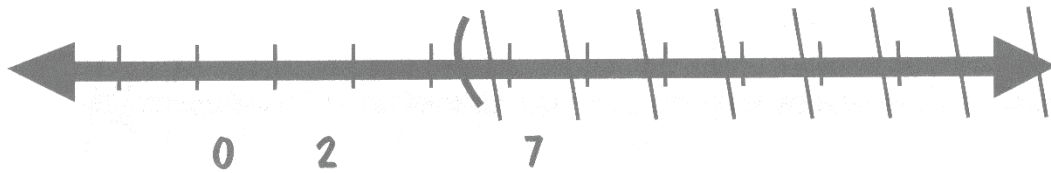


Los valores verdaderos para este ejemplo serán 8,9, 10, 11,12,13, 14

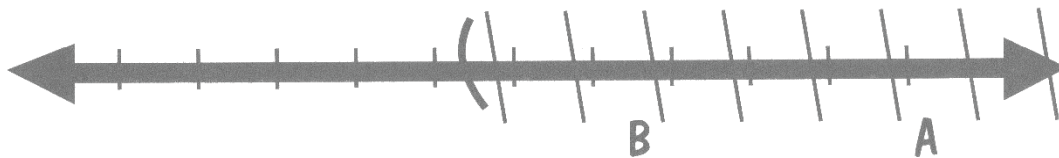
Ejemplo 3 :Si $A = 7$, $B = 7$, $A > B$ entonces la condición es falsa



Ejemplo 4: Si $A= 2 ,B= 7 , A>B$ entonces la condición es falsa



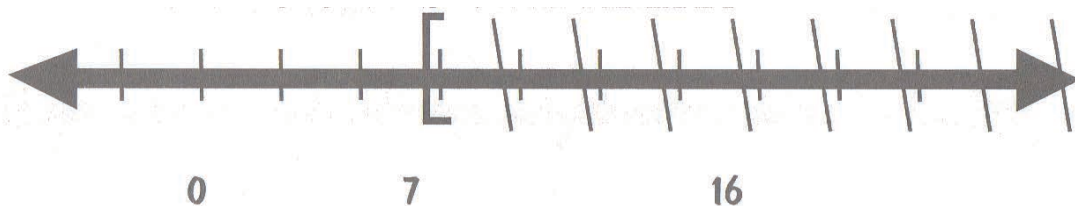
Si $A>B$ se puede expresar gráficamente de esta forma distinta:



La interpretación de esta representación es la siguiente:

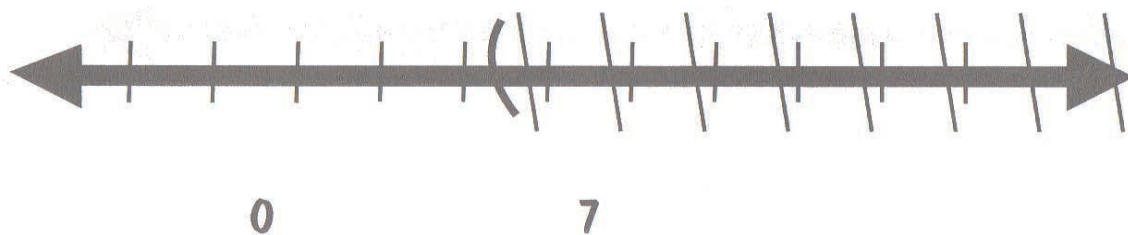
Todos los datos ingresados en la variable A serán verdaderos si se encuentran dentro del rango (zona rayada) que comienza a partir desde B incluyendo su valor, es decir, A es mayor e igual que B.

Ejemplo 5: Si $A=6, B=7 , A \geq B$ entonces la condición es verdadera

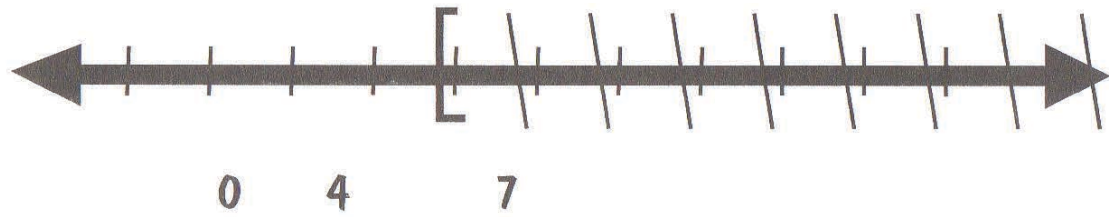


Los valores verdaderos para este ejemplo serán 7,8,9, 10,11,12,13, 14 ..

Ejemplo 6: Si $A=7 ,B=7 , A \geq B$ la condición es verdadera

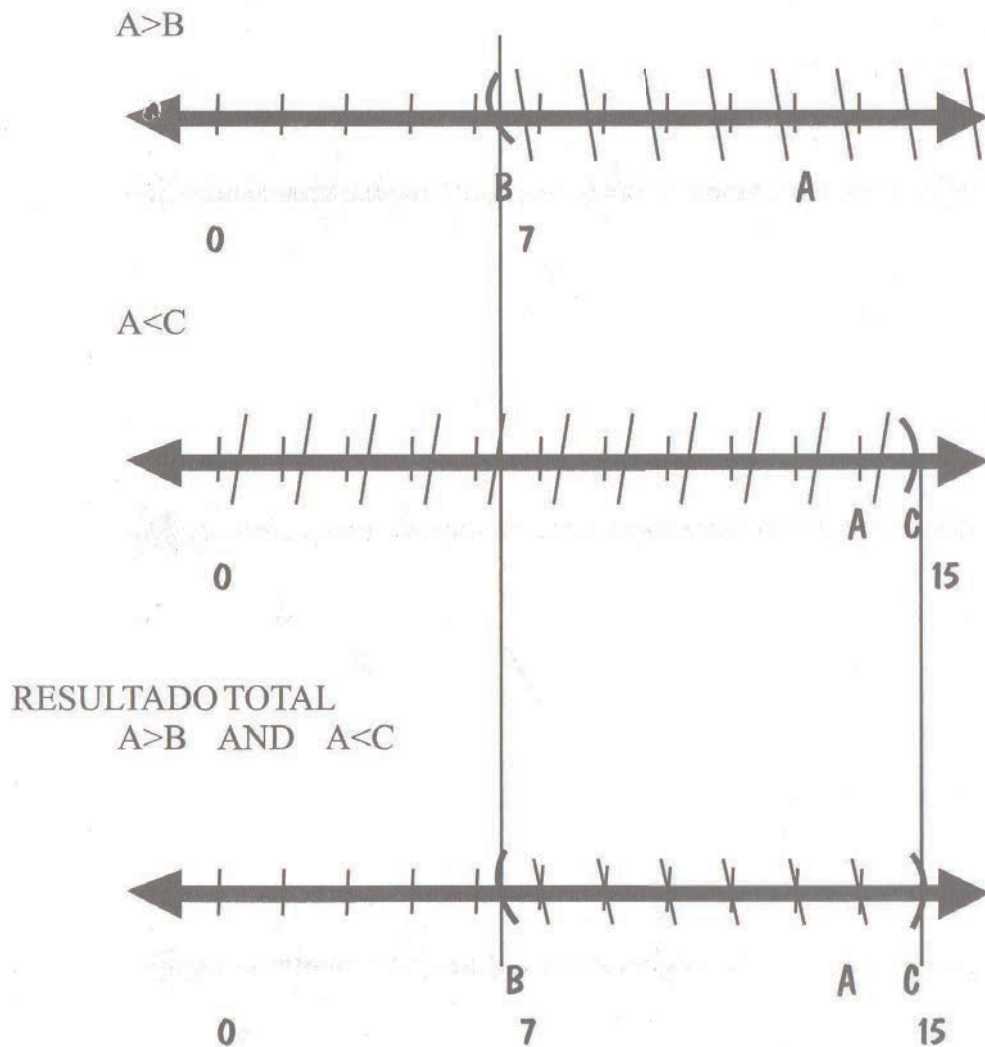


Ejemplo 7: Si $A=4$, $B=7$, $A \geq B$ entonces la condición es falsa



Ejemplo 8: $(A=B) \text{ AND } (A < C)$ Si $B=7$, $C=15$

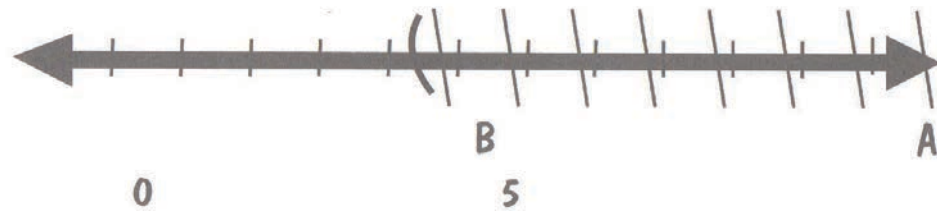
Para resolver esta expresión y saber cuáles son los datos verdaderos, se divide la expresión por partes y si todas las condiciones se cumplen el resultado total es verdadero como observamos en la tabla AND .Se procede a interceptar las condiciones y en los lugares con doble rayado se obtendrá el rango de valores validos que se pueden ingresar por medio de la variable A.



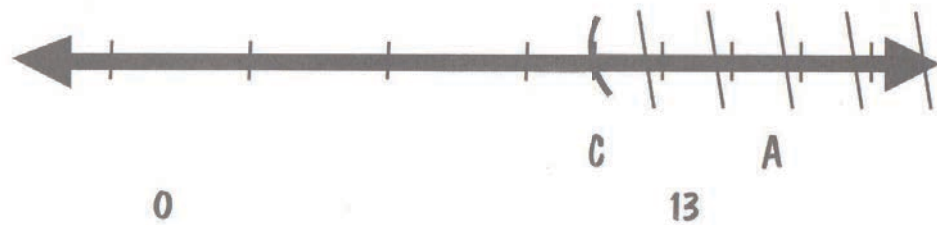
Los valores verdaderos de la variable A deben encontrarse dentro de este rango entre B y C, es decir, los valores válidos son: 8,9, 10, 11, 12, 13, 14. Si revisamos la tabla AND tendrían que cumplirse las dos condiciones para que el resultado total sea verdadero.

Ejercicio #8

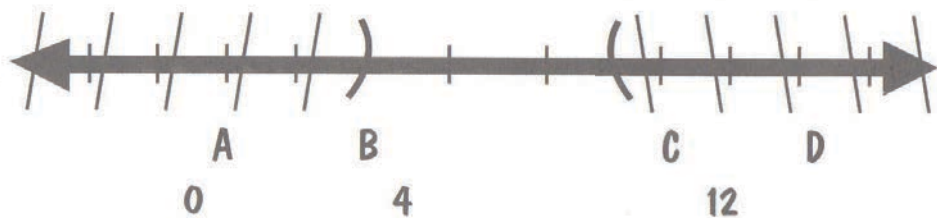
$A > B$ AND $A > C$ Si, $B=5$, $C=13$



$C > B$



RESULTADO TOTAL
 $A > B$ AND $A > C$

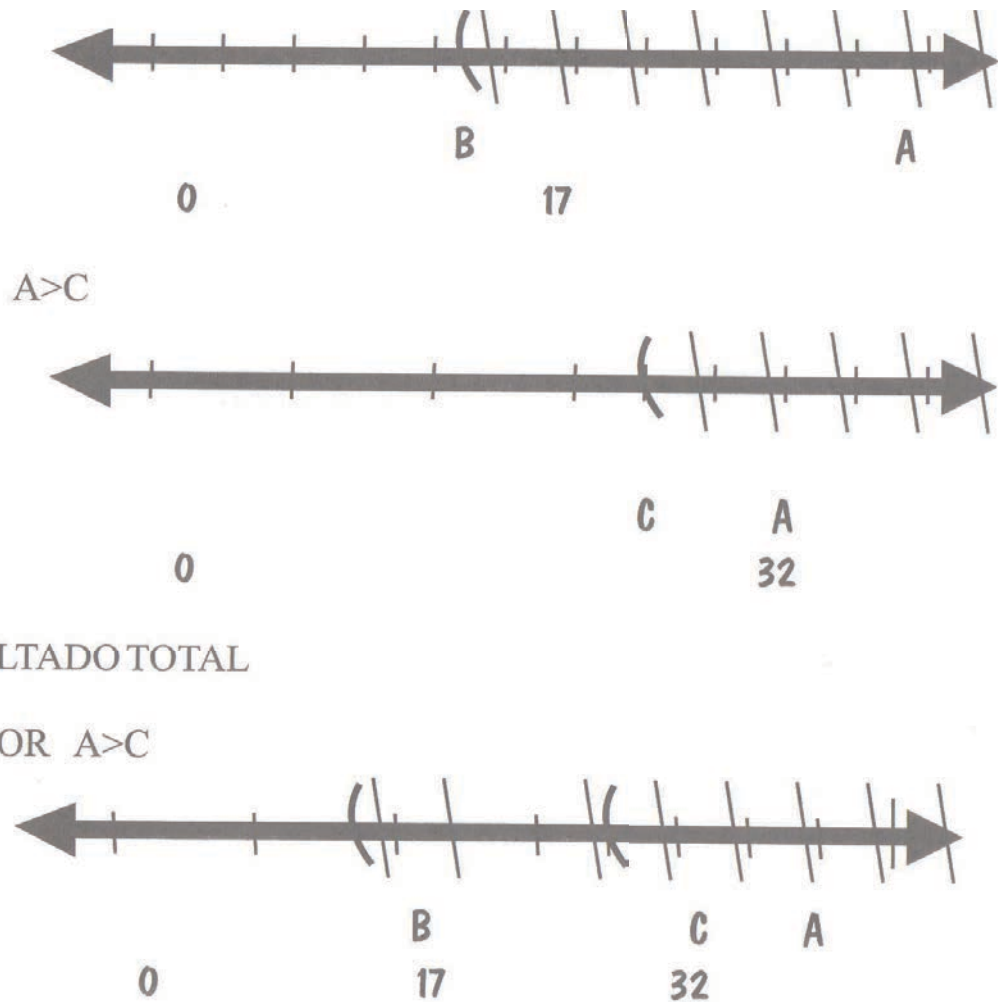


Como no se interceptan los rangos se los tratan por separado y observamos si las dos condiciones son verdaderas, guiándonos por la tabla AND, Si la primera y segunda condición es verdadera el resultado total es verdadero

Ejercicio #9

$A > B$ OR $A > C$ Si $B=17$, $C=32$

Para resolver expresiones con operadores OR si una de las condiciones se cumple el resultado total es verdadero o utilizamos el método de la recta numérica interceptando las condiciones y todas zonas rayadas son solución



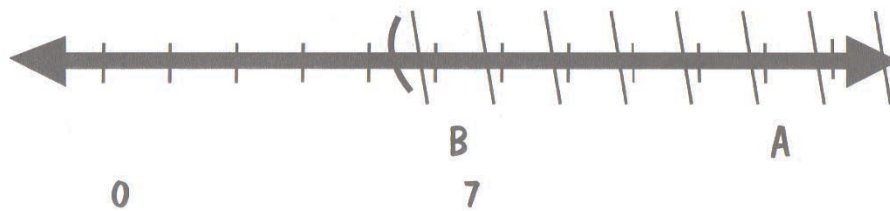
Cualquier valor ingresado en la variable A a partir del número 18 es un resultado válido.

Ejercicio #10

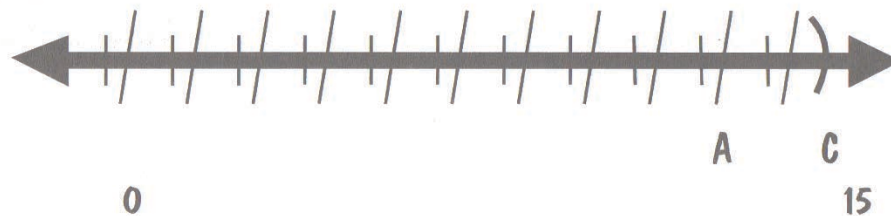
$$A > B \text{ OR } A < C \quad \text{Si } B=7, C=15$$

Para resolver esta expresión y saber cuáles son los datos verdaderos, dividimos la expresión por partes y si una de las condiciones se cumple el resultado total es verdadero como observamos en la tabla OR o todo valor que se encuentre en zona rayada es solución.

$A > B$

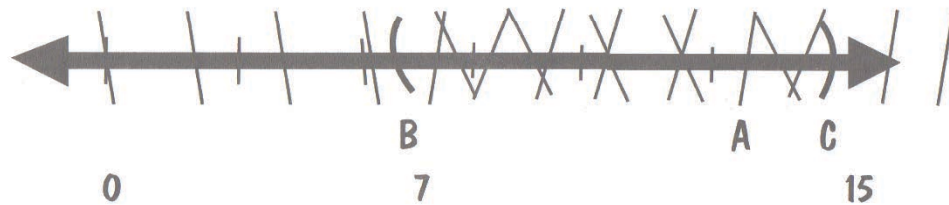


$A < C$



RESULTADO TOTAL

$A > B \text{ OR } A < C$



Cualquier valor regresado en A es válido como resultado total debido a que la zona rayada abarca toda la recta numérica. Pueden existir mas formas de expresar las condiciones en conjunto con los operadores dependiendo de las necesidades.

Funciones

Son expresiones almacenadas en el computador para desempeñar una función específica con el fin de resolver de una manera sencilla operaciones complicadas.

Las principales Funciones son:

Tabla 9: Funciones

DESCRIPCIÓN	FUNCIÓN	EXPLICACIÓN
Valor Absoluto de x	Abs (x)	Devuelve el valor absoluto de la variable.
Función Exponencial	Exp (x)	Calcula la función exponencial de la variable
Valor Entero	Int (x)	Determina el mayor entero menor que la variable
Logaritmo Natural	Ln(x)	Calcula el Logaritmo natural, base e de la variable
Redondeo de x	Round(x)	Redondea el valor aproximado
Seno de x	Sen (x)	Extrae el seno de la variable
Coseno de x	Cos (x)	Extrae el coseno de la variable
Logaritmo Natural	Log10 (x)	Calcula el Logaritmo Decimal de la variable

Ejemplos:

Abs (-12)	12	Log (15)	1.1760
Abs (-134)	134	Log (25)	1.397
Abs (9)	9	Log (78)	1.892
Round (3.89)	4	Sen (60)	0.866
Round (15 .444)	15	Sen (47)	0.731
Round (13.56)	14	Cos (60)	0.5

Actividad # 4

1.-Conteste las siguientes preguntas

¿Cuál es la diferencia entre Expresión y Operador Aritmético?

¿Cuáles son las reglas prácticas para la resolución de una expresión aritmética?

2.-Realice los siguientes ejercicios

Transforme las expresiones Algebraicas en Aritméticas

$$E = \frac{\frac{1}{A}}{\frac{\sqrt{K}}{C^{10}}}$$

$$X = \frac{Z^Z(B + A)}{Z^Z + Y^Z + Z * A}$$

$$J = \frac{A * 5 + 2}{A}$$

3.-Resolver

$K=15 \text{ DIV } 6$	$K=15 \text{ MOD } 6$	$K=0 \text{ DIV } 6$
$K=10,3 \text{ DIV } 3$	$K=10,3 \text{ MOD } 3$	$K=23 \text{ MOD } 4$
$K=60 \text{ DIV } 86$	$K=60 \text{ MOD } 8$	$K=0 \text{ MOD } 6$

4.-Resolver las siguientes expresiones

$L=330/5/11 \text{ mod } 3$ respuesta $L=0$

$M= \lceil \{2+6^2 \text{ mod } 2 + 5\} / 3^2 \rceil / 2$ respuesta $M=0.38$

$M=7/3+4 \text{ div } 2+4^2+7 \text{ mod } 3$ respuesta $Z=21.3$

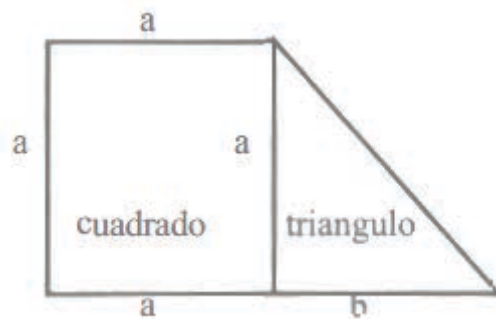
$X=15/5*3/3-5+7^2$ respuesta $X=47$

$S=28/2/2^2+18 \text{ mod } 15/3 \text{ div } 3$ respuesta $S=4.5$

$S=2^2 \wedge 2^2 \wedge 2 \wedge 2 \text{ div } 2$ respuesta $S=32768$

$M=2^5 * [2^2 - 2 * 2] / (3^2 + 3^2)$ respuesta $M=0$

5. Realice un algoritmo para calcular el área de la siguiente figura, presentar el resultado y luego finalizar



6.-Conteste Falso (F) o Verdadero (V)

Ir en buseta es más económico que el taxi. No tengo dinero suficiente para ir en taxi; por lo tanto:

Puedo ir en taxi o en buseta ()

Puedo ir en taxi y en buseta ()

Ninguna casa de este barrio tienen balcones, pero todas tienen ascensor por lo tanto

Las casas tienen balcones o ascensor ()

Las casas tienen balcones y ascensor ()

7. Resolver

Si $A=10$, $B=12$, $Z=C$, $C=A+B$

$X=Z$

$Y=X+A$

¿Cuál es el valor de Z , X , Y ?

Si $X=2$, $Y=12$ $Z=X^2$

$Z = Y+Z$

$A= Z+X$

$Y=X+Y+Z$

¿Cuál es el valor de Z , X , Y ?

Si SALARIO= 300, S=4000, Q=2

$$Z=S+Q$$

$$DEUDA=Z*3$$

$$SALARIO=SALARIO+Z$$

$$T= SALARIO-DEUDA$$

¿Cuáles el valor de SALARIO, Q, T?

Si $X=2$

$Y=12$

$$Z=X+Y^X$$

$$Z=Y+Z$$

$$A=Z+X$$

$$R=Z+A^2$$

¿Cuál es el valor de Z, X, Y?

Capítulo 5

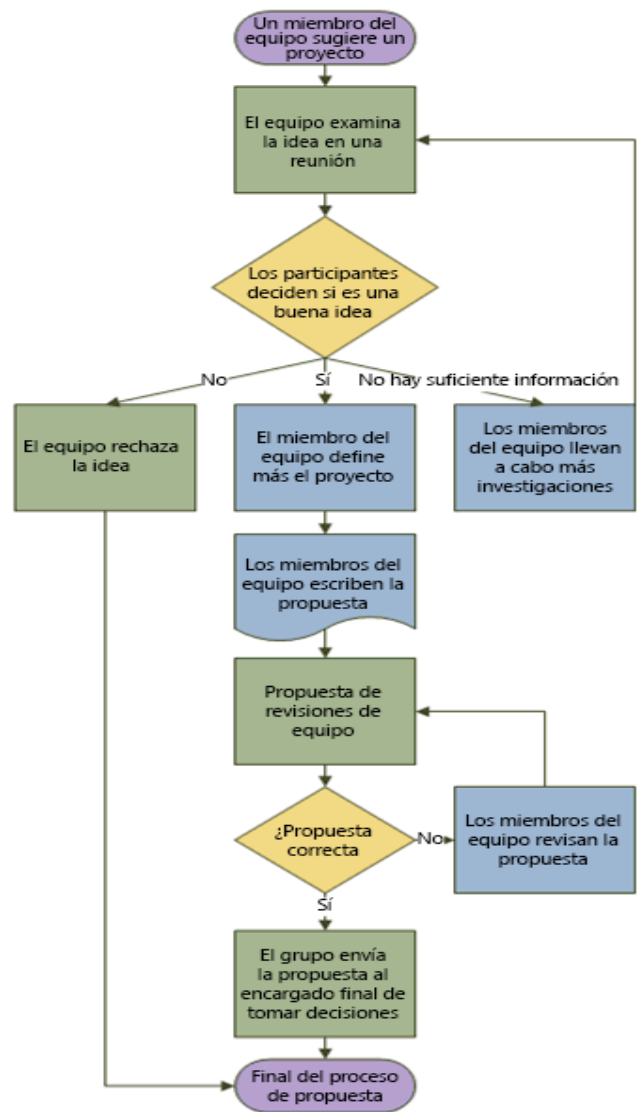
Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Tener conocimiento de los conceptos básicos para el desarrollo de un diagrama de flujo
2. Tener un criterio claro sobre diagramas de flujo
3. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad

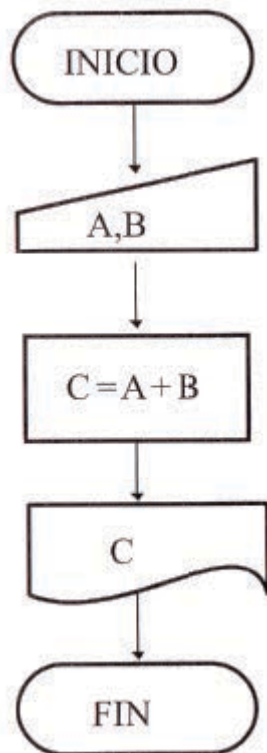
Diagrama de flujo
Definición

Representación gráfica de un algoritmo que permite un mejor entendimiento de problemas complejos mediante símbolos conectados entre sí. Además Cairo (2005) dice que los diagramas de flujo” representan la esquematización grafica de los algoritmos, en realidad muestran de forma gráfica los pasos o procesos a seguir para alcanzar la solución de un problema” (p.4), por ejemplo: Desarrollo de una propuesta de proyecto.



Fuente: Microsoft, 2017, <https://support.office.com/es-es/article/Crear-un-diagrama-de-flujo-b%3%A1sico-e207d975-4a51-4bfa-a356-eeec314bd276>

Realizar un diagrama de flujo de la suma de dos números ingresados por teclado, luego presentar el resultado.













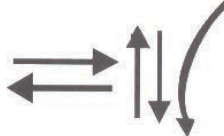
Ventajas de los diagramas

El diagrama de flujo clarifica y simplifica la resolución de un programa, se puede realizar modificaciones presente y futuras, además debido a su diseño se lo realiza por módulos sencillos de entender y es la base para programar en diversos lenguajes.

Los símbolos

Son formas geométricas estandarizadas que indican una orden a ejecutarse. En la figura 12 se presentan los símbolos del diagrama de flujo

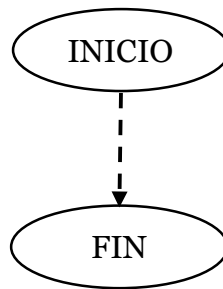
Figura 12: Símbolos de los diagramas de flujo

	Instrucción de inicio ó fin
	Instrucción ingreso por teclado
	Instrucción presentación por impresora
	Instrucción entrada o salida por pantalla
	Instrucción entrada o salida por periféricos
	Instrucción de proceso
	Conector dentro de una página
	Conector de diferentes páginas
	Instrucción de decisión
	Instrucción de decisión de rutinas
	Líneas conectoras

Símbolos de inicio o fin

Muestra donde comienza y termina la resolución del problema. En la figura 13 se presenta el símbolo de inicio y fin.

Figura 13: Símbolo de inicio y fin



Símbolo de entrada por teclado

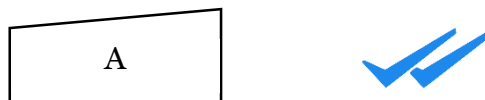
Representa el ingreso de datos mediante el teclado. En la figura 14 se presenta el símbolo de inicio y fin.

Figura 14: símbolo ingreso por teclado.



El ingreso de los datos se los realiza en el interior del símbolo mediante nombre de variables de una expresión que se desea resolver por ejemplo:

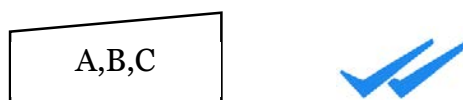
Ingresa las variable de la siguiente expresión $B=A+7$



Ingresa las variables de la siguiente expresión: $M=Y/G*8$



Cuando se tiene varias variables el ingreso se realiza separándolas con una coma por ejemplo: ingresar la expresión $F=A/B*C$



Ingresar la expresión $X = [(Y * G) + \text{SQR}(Y)] / Y$

Y,G,Y,Y



Y, G



Siempre se debe ingresar una sola vez la variable aunque este repetida

Ingresar las variables de la siguiente expresión aritmética $X = U^2 + \text{SQR}(R)$

$L^2, \text{SQR}(R)$



L, R



No se recomienda ingresar expresiones dentro de los símbolos de ingreso solo las variables que intervienen por ejemplo: $A > B$

$A > B$



A, B



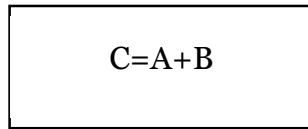
Símbolo de proceso

Realiza cualquier operación matemática interna del computador. Dentro del símbolo se escriben las expresiones aritméticas que se necesita resolver a través de computador.

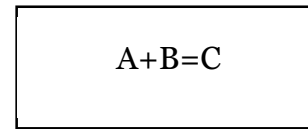
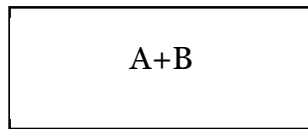
Figura 15: Símbolo de proceso



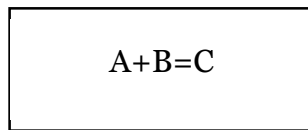
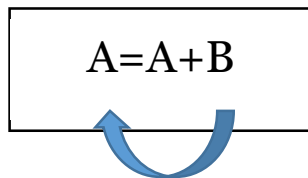
Por ejemplo: Suma dos números



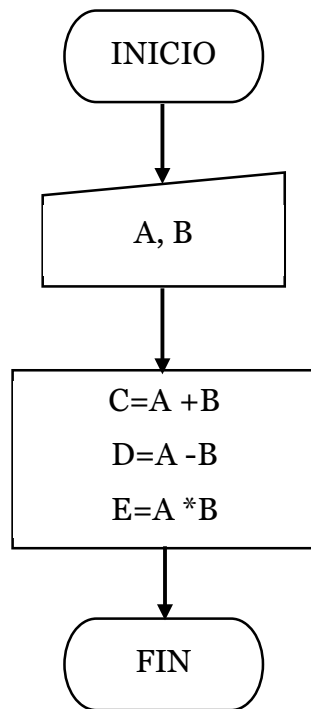
Siempre se debe guardar el resultado de la operación en otra variable para poder utilizarla cuando el computador la necesite.



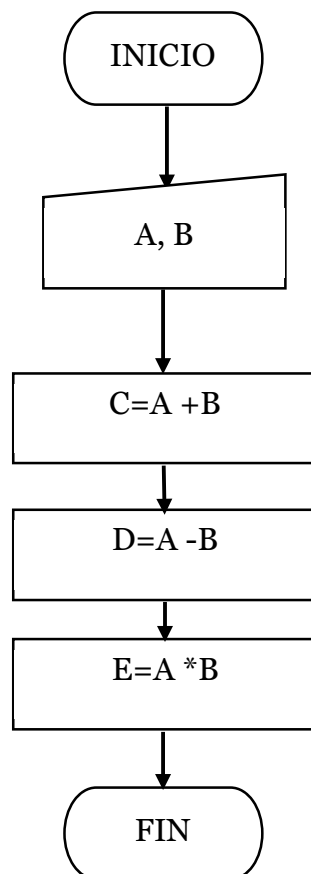
Este almacenamiento siempre se realiza de derecha a izquierda



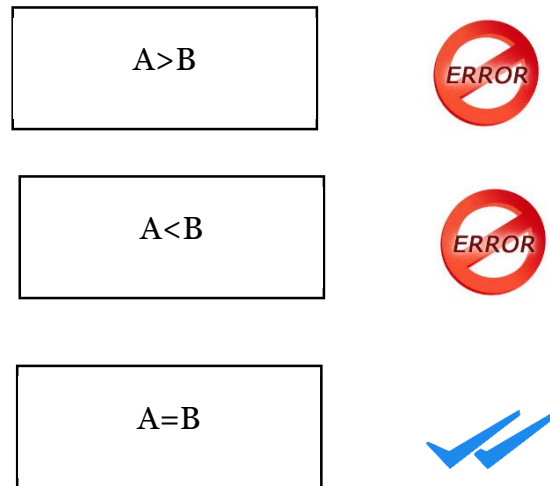
Cuando se tienen varias expresiones que realizar, se pueden escribirlas en el mismo símbolo o por separado ejemplo: ingrese dos números, súmelos , réstelos y multiplíquelos.



También resulta realizarlo por separado



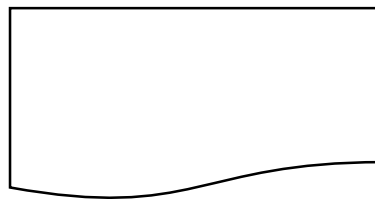
No se recomienda representar expresiones lógicas en el símbolo de proceso con excepción del operador igual(=)



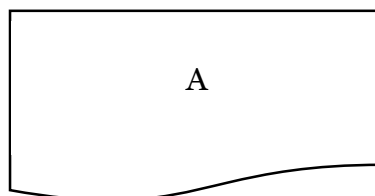
Símbolo de presentación por impresora

Muestran mensajes, resultados almacenados en alguna variable mediante la impresora. En la figura 16 se presenta el símbolo presentar por impresora

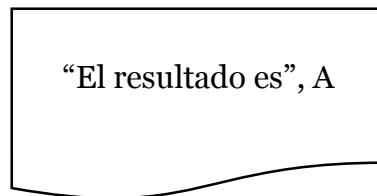
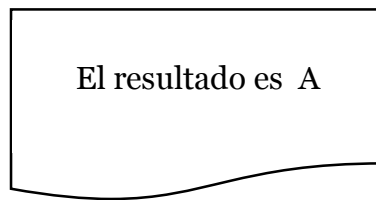
Figura 16: Símbolo presentar por impresora.



Dentro del símbolo se escribe las variables a presentarse.

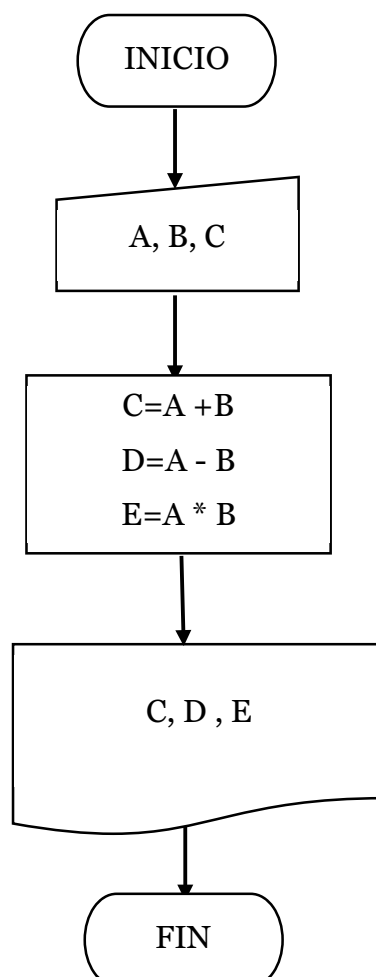


Si a la respuesta se le desea agregar un mensaje se deberá escribir dentro del símbolo de presentar por impresora y los mensajes deberán ir entre comillas y separados por una coma de la variable

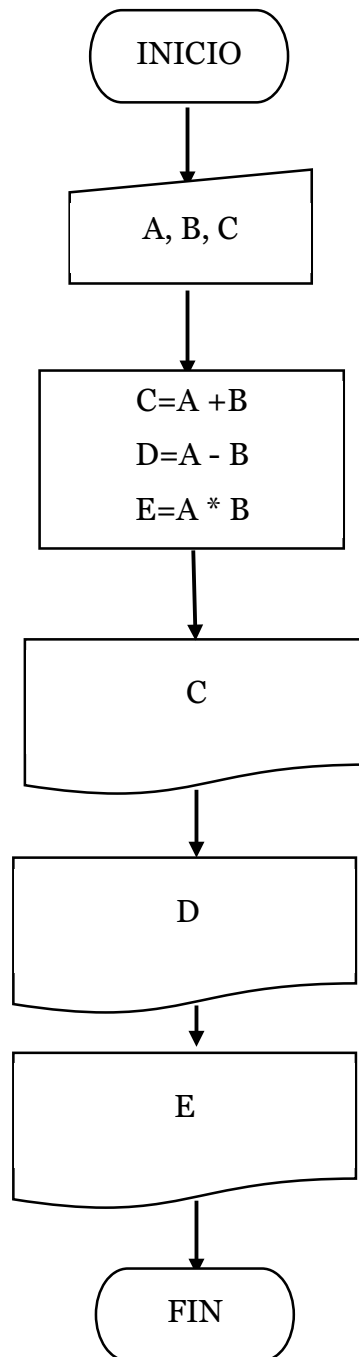


Si se tiene que presentar varios resultados se puede realizar en el mismo símbolo pero con la condición de separarlo con coma o se realizan individualmente

Por ejemplo: Realizar un diagrama de flujo para ingresar dos números, súmelos, réstelos y multiplíquelos, Luego presentarlos los resultados por impresora



El anterior diagrama podría ser diseñado con el símbolo de presentación por separado



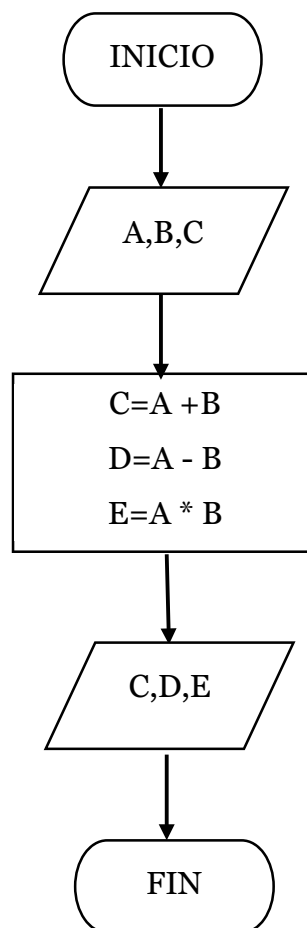
Símbolo de entrada/ salida por periférico

Puede realizar cualquiera de las dos acciones de entrada o salida

Figura 17: Símbolo de entrada /salida por periférico



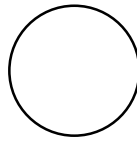
Por ejemplo: Realizar un diagrama de flujo para ingresar dos números, súmelos, réstelos y multiplíquelos, Luego presentarlos los resultados por impresora



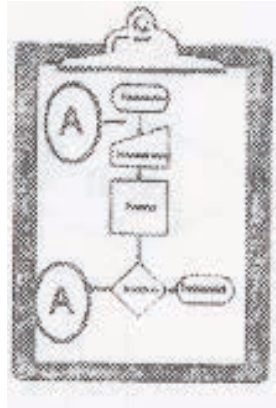
Símbolo conectores internos en una página

Mantiene el orden de un diagrama de flujo en una página mediante un enlace representado por un círculo en su interior se escribe un número o letras que indicará de donde viene o donde va.

Figura 18: Símbolo de conector interno de página



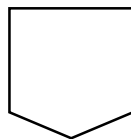
Por ejemplo:



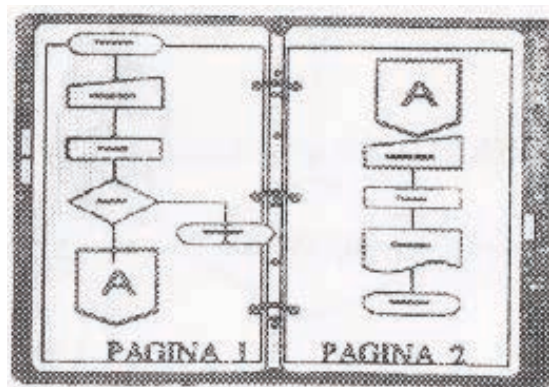
Símbolo conectores externos en una página

Mantiene el orden en un diagrama de flujo cuando se tiene que diseñar en varias páginas mediante un enlace. En su interior se escribe un número o letras que indicará de donde viene o donde va.

Figura 19: Símbolo conectores externos en una página

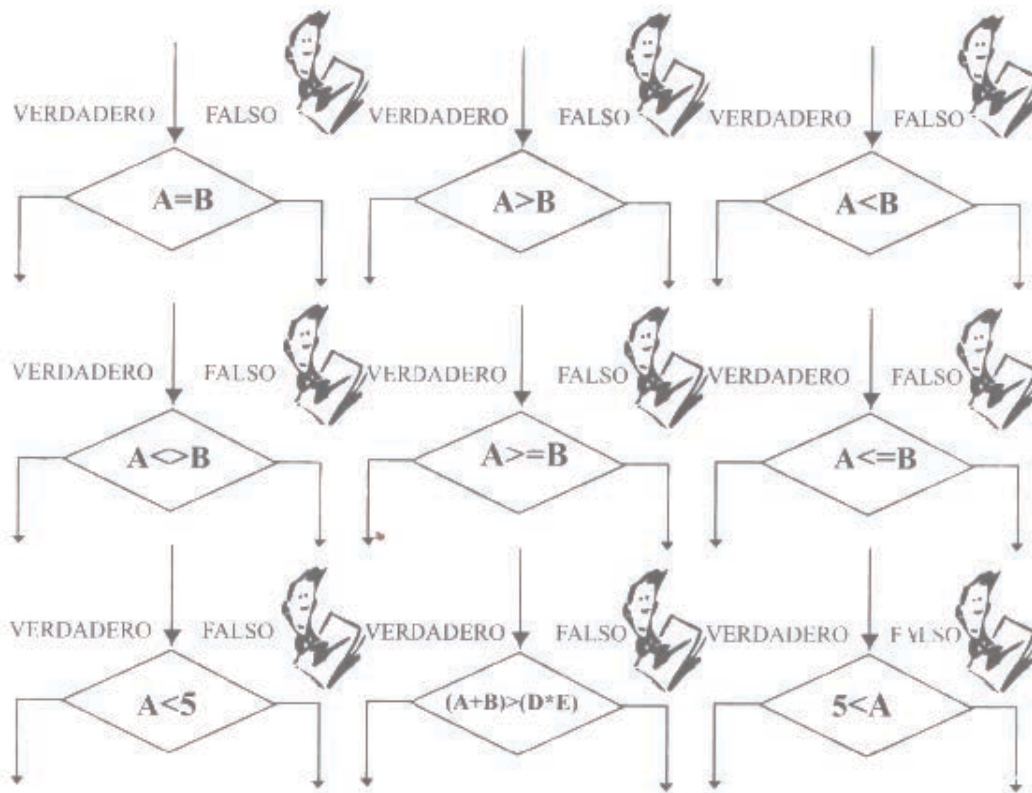


Por ejemplo:

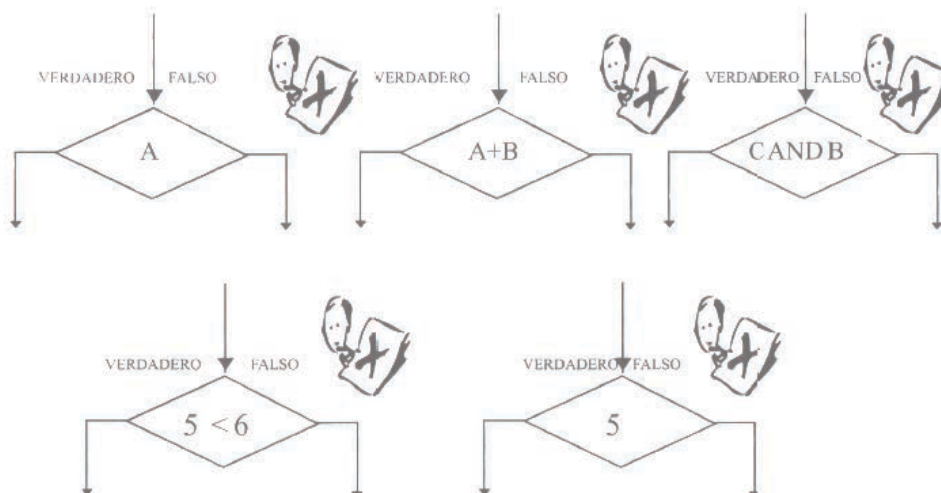


Símbolo de decisión con bifurcación doble

Mejor conocida como condición que realiza la pregunta de cuál es el camino a seguir, permitiendo la opción de elegir entre dos caminos denominados verdadero o falso, en otros casos si o no. La decisión posee en su interior expresiones lógicas utilizando operadores relacionales. Ejemplo de expresiones en la decisión:



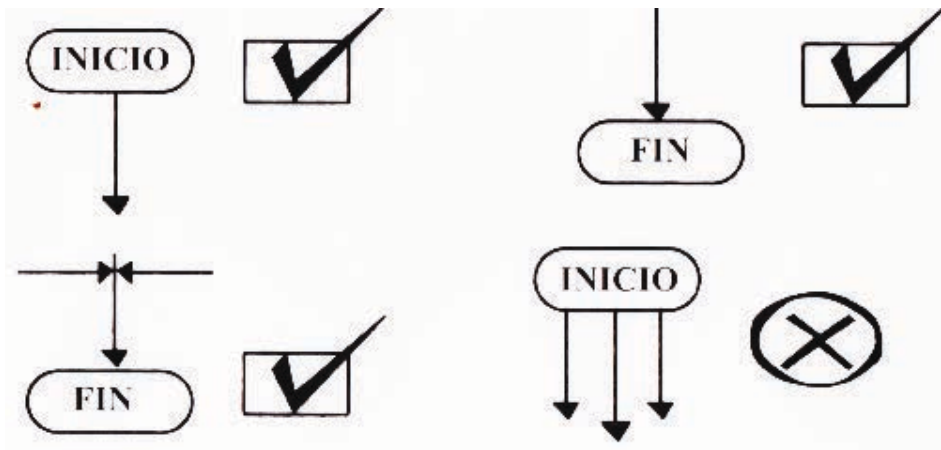
Ejemplo de la decisión con una condición incorrecta:



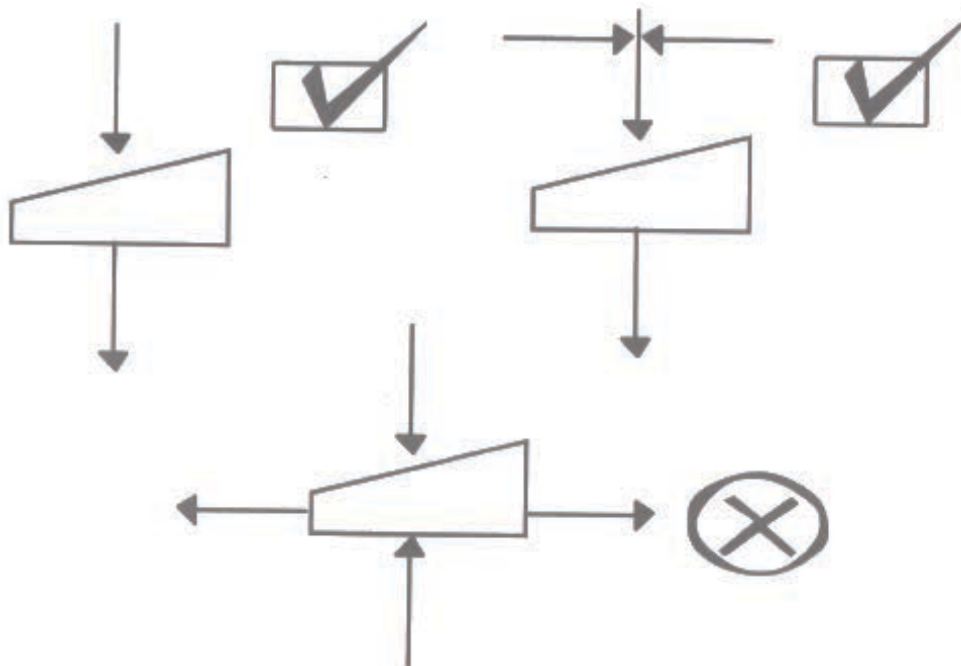
Símbolo de líneas de flujo

Permiten la conexión entre símbolos, la dirección del flujo de datos se determina por la cabeza de flecha que indican la ejecución de las operaciones. Las líneas pueden ser rectas o curvas

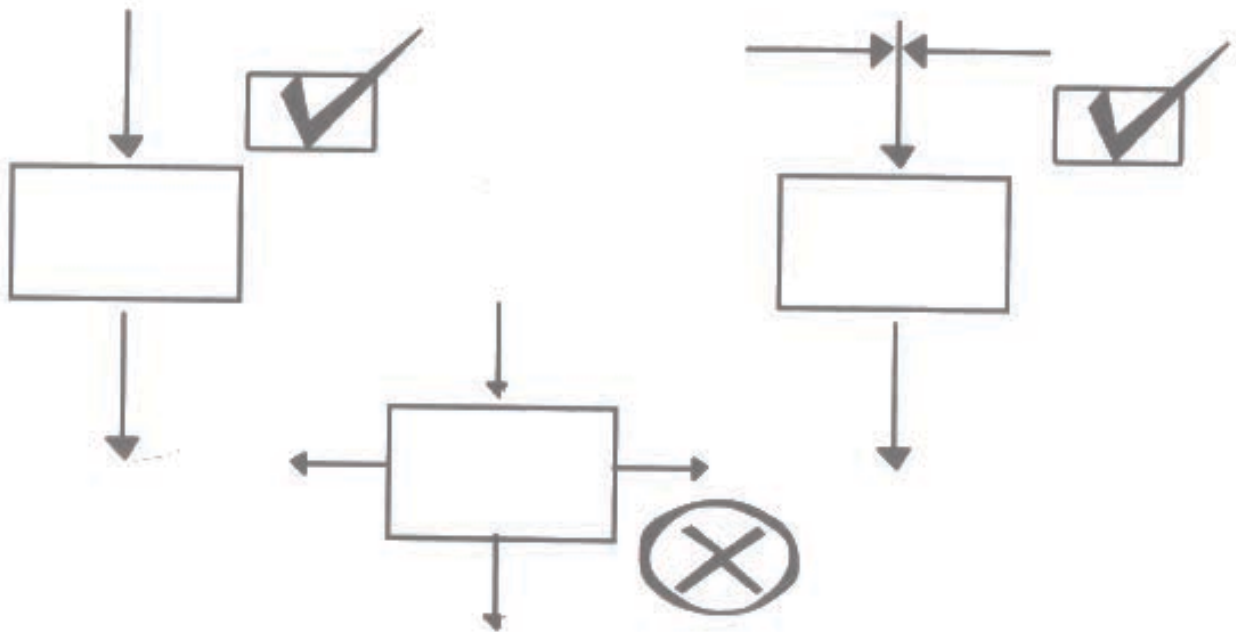
Las líneas de flujo en el símbolo inicio o fin .- Con lo que respecta al símbolo de inicio solamente sale una línea, en el fin pueden llegar una o más líneas pero ninguna sale , por ejemplo :



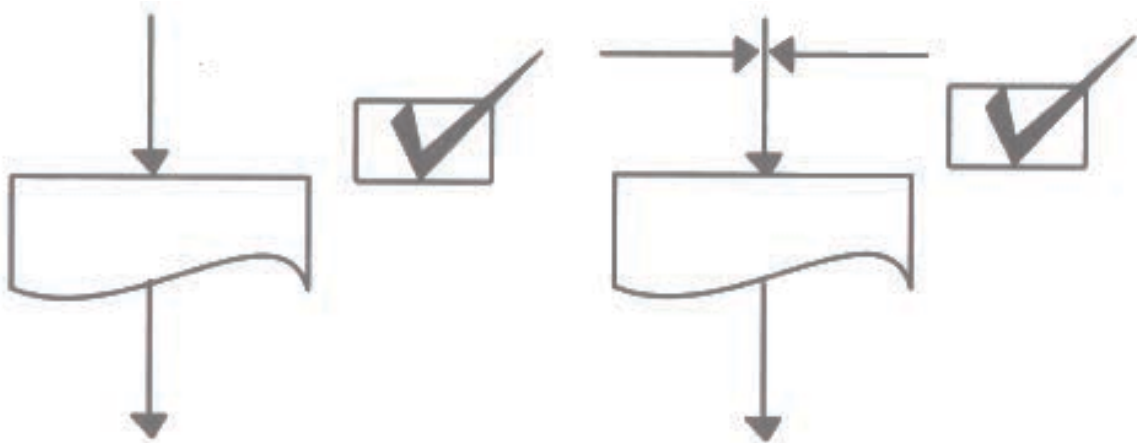
Las líneas de flujo en el símbolo de ingreso por teclado.- Ingresan una o más líneas y sale una sola, por ejemplo:



Líneas de flujo en el símbolo de proceso.- Ingresa una o varias líneas y solamente sale una, por ejemplo:

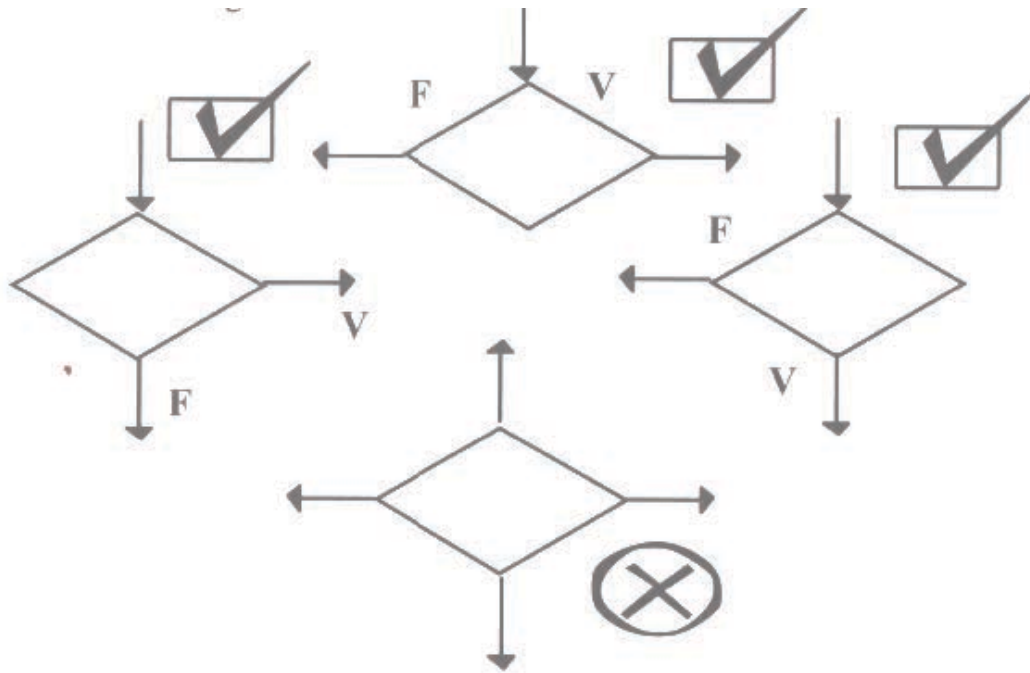


Líneas de flujo en el símbolo de presentación por impresora.- Ingresa una o varias líneas y solamente sale una, por ejemplo:



Líneas de flujo en el símbolo ingreso/salida por periférico.- Se adoptan las mismas normas que los símbolos de ingreso por teclado y presentación por impresora

Líneas de flujo en el símbolo de la decisión.- Ingresan una o varias líneas y salen dos. El camino verdadero o falso lo determina la persona que diseñe el diagrama, por ejemplo:



Principios básicos para la resolución de diagramas de flujo

Se deben tener presente los datos que van usar el programa.

Los datos que van generar el programa

Entender el problema que se quiere resolver

Luego, nos planteamos un mecanismo o proceso básico, que permite la resolución del problema.

Una vez que tengamos claras nuestras ideas de la solución del problema, lo representamos, por medio de los diagramas de flujos

Finalmente, realizamos las pruebas necesarias para comprobar que no existen errores.

Reglas generales para la elaboración de Diagramas

Todo diagrama tiene un inicio y un fin.

No deben cruzarse los bucles.

El diagrama se dibuja de arriba hacia abajo y de derecha a izquierda.

Dos líneas flujos no deben cruzarse entre sí, utilice conectores o líneas arqueadas de flujo.

Utilizar conectores dentro y fuera de la página cuando sean necesarios

Un diagrama de flujo se puede realizar de más de una forma, la variación depende del mecanismo para la resolución del problema.

Para representar la dirección del flujo se utilizan líneas rectas horizontales y verticales. El sentido está dado por una flecha en sus extremos.

Las líneas de flujo deben ir a parar siempre a un símbolo o línea. no deben existir líneas desconectadas.

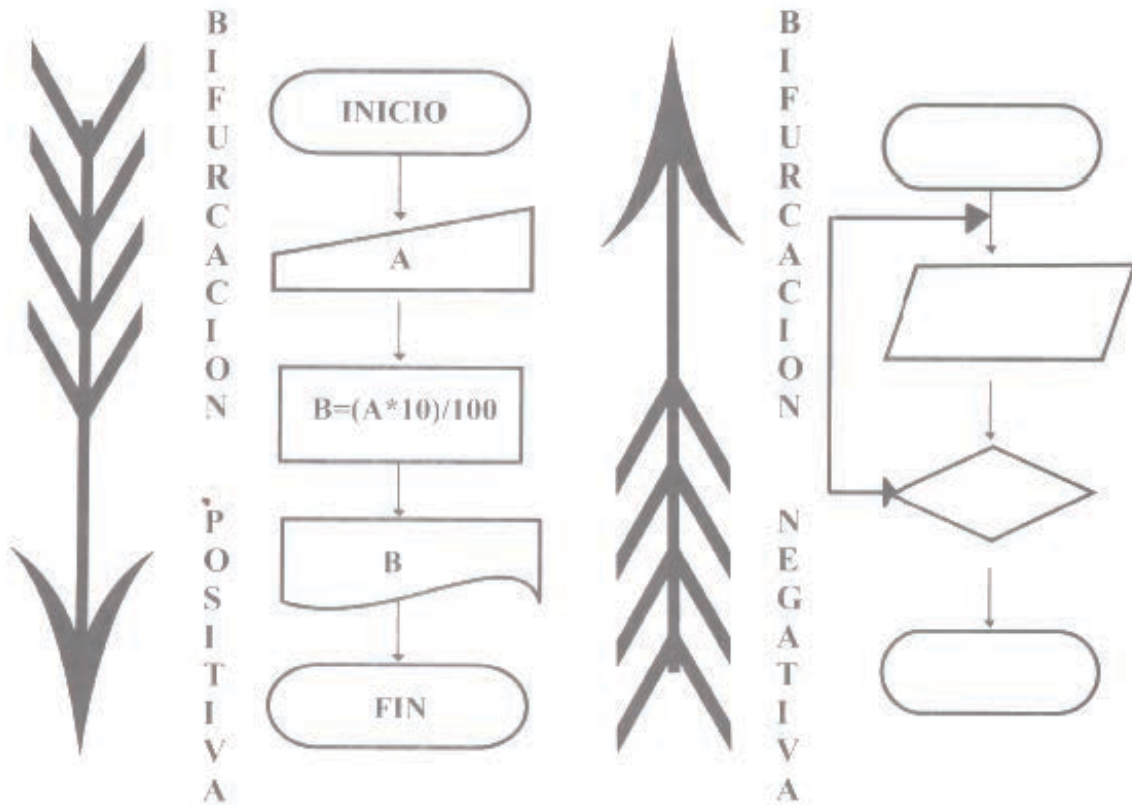
Bifurcación

Son los caminos que siguen las líneas de flujo y el número de caminos que se dividen las líneas de flujo, según la dirección se clasifican en: bifurcaciones positiva y bifurcaciones negativas

Bifurcaciones positiva.-Cuando el camino que sigue es de arriba hacia abajo

Bifurcaciones negativas.-Cuando el camino que sigue es de abajo hacia arriba

Figura 20: Bifurcaciones positivas y negativas



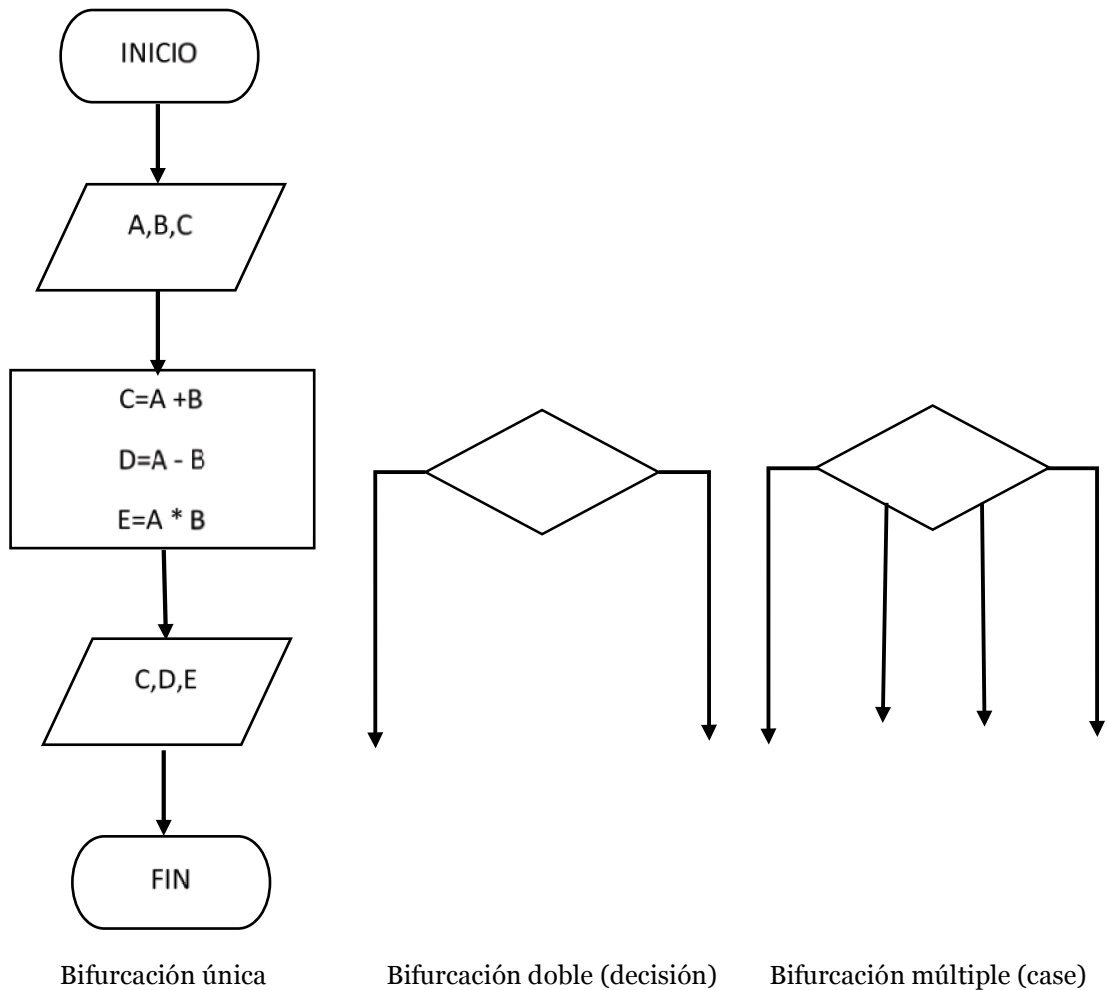
Según el número de caminos en que dividen las líneas de flujo se clasifica en: Bifurcación única, doble y múltiple

Bifurcación única realiza cuando las líneas de flujo siguen un solo camino

Bifurcación doble se presenta cuando las líneas de flujo siguen dos caminos

Bifurcación múltiple se presentan cuando las líneas de flujo siguen más de dos caminos

Figura 21: Bifurcación única, doble y múltiple



Decisiones con bucles o lazo

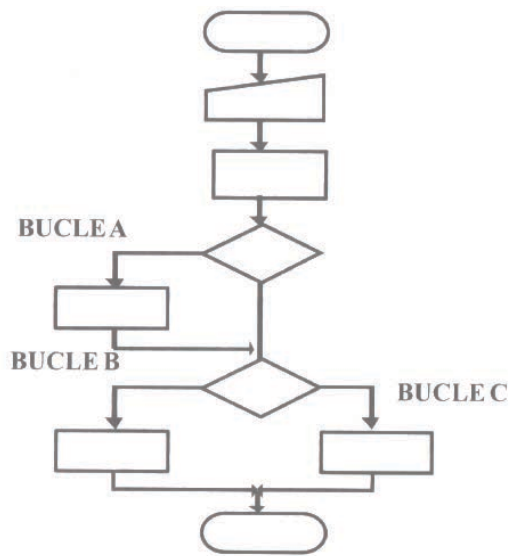
Es una parte del diagrama de flujo, que logra repetir un número determinado de veces las órdenes expresadas por los símbolos, mientras se cumpla una condición que puede ser verdadera o falsa.

Los bucles se clasifican según su forma y su iteración. Según su forma se clasifican en: decisiones con bucles anidados y bucles independientes

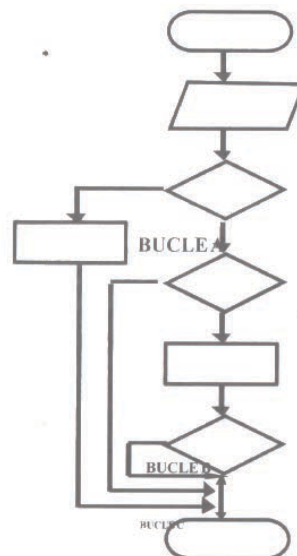
Bucles independientes.-Son independientes cuando son externos uno a otros.

Bucles anidados.-Son anidados cuando existen varios lazos contenidos en bucles independientes

Figura 22: Bucle independiente y anidado



Bucle independiente



Bucle anidado

Según su iteración se clasifican en: decisiones con bucles no iterados y bucles iterados

Decisiones con bucles no iterados se los utiliza en los diagramas simples y complejo sin realimentación.

Decisiones con bucles iterados se los utiliza para los diagramas complejos con realimentación.

Actividad # 5

¿Qué es una Bifurcación?

¿Cuál diferencia entre Bucle y Bifurcación?

¿Cuál es la diferencia entre Decisión y Bifurcación?

Escriba sus ideas sobre las líneas de flujo

¿Qué símbolos desempeñan más de una función?

Realizar el diagrama de los siguientes enunciados

a) Presentar el 14% del IVA de un valor ingresado, luego finalice

Fórmula $B = A * 12/100$

b) Presentar el promedio de cuatro notas, luego finalice

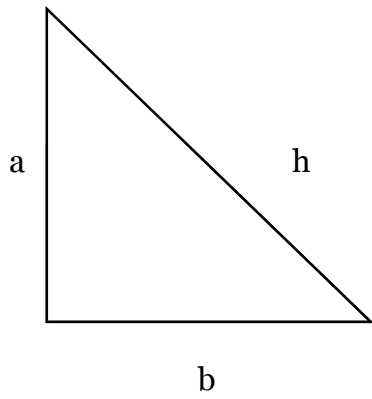
c) Un diagrama para calcular automáticamente la siguiente fórmula,

$$\text{fórmula} = \frac{m^2 + m + e * e}{35 - s + s^2 \sqrt{c + 1}} / g + 32x$$

Los números son ingresados por teclado, presente el resultado y finalice

d) Realizar un diagrama para calcular los valores de la tabla de multiplicar del 1 al 12 de cualquier número ingresado presentar los valores

e) Para obtener la Hipotenusa (h) en un triángulo rectángulo, utilice el teorema de Pitágoras, presente el resultado y finalice



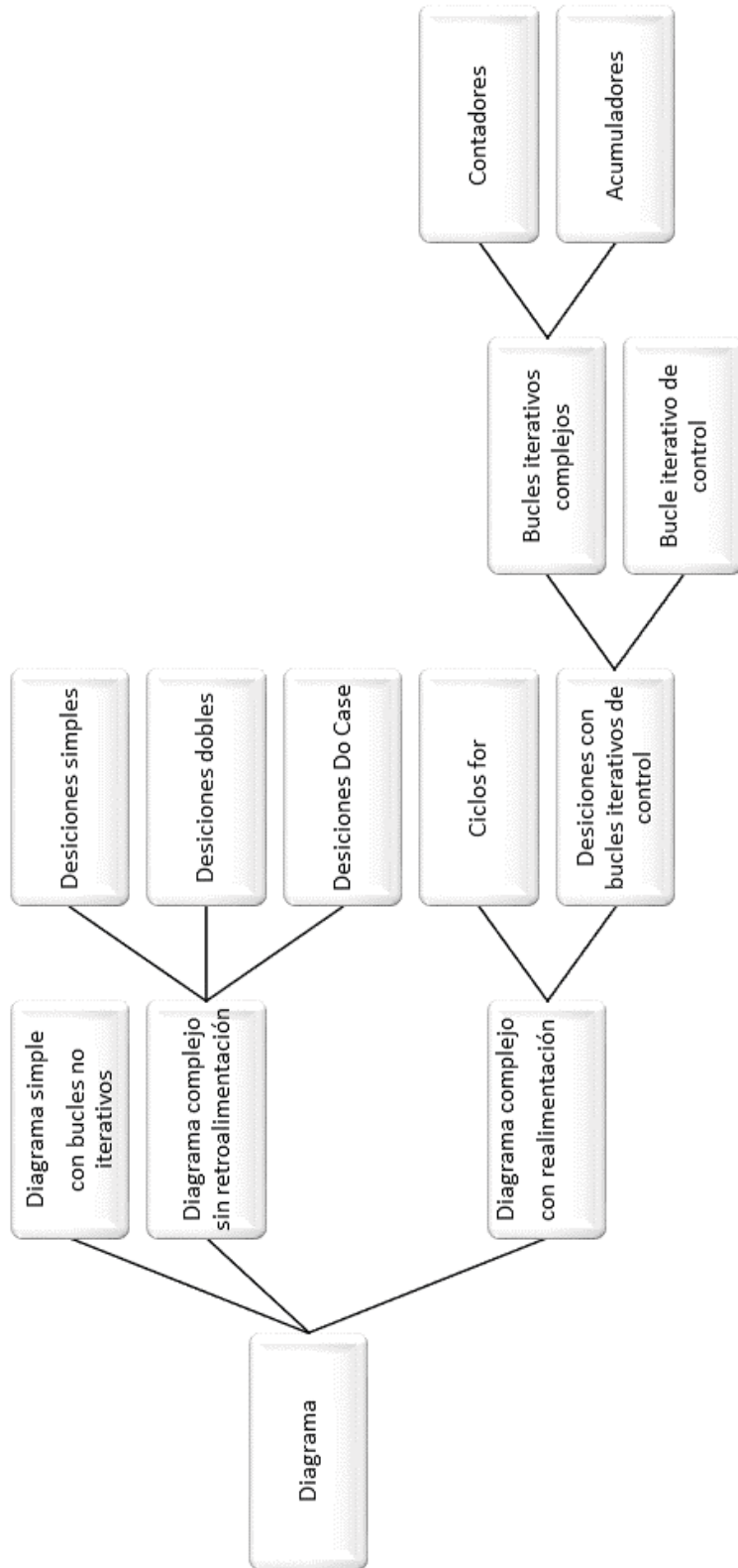
Capítulo 6

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Obtener conocimiento teórico, para desarrollar de una forma práctica, los diagrama de flujo simple.
2. Entender a la programación como un medio de comprensión del funcionamiento real del computador.
3. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.

Figura 23: Clasificación de los diagramas de flujo

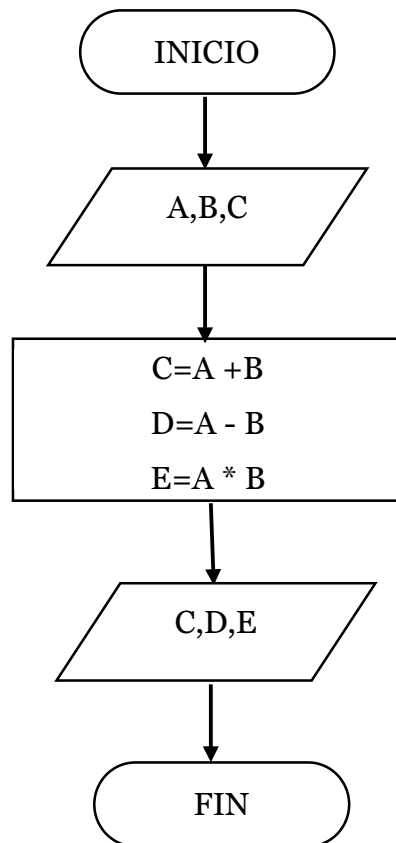


Clasificación de los diagramas de flujo según su estructura

Se puede clasificar a los diagramas de flujo en: diagramas de estructura simple, compleja sin realimentación y compleja con realimentación (repetitivas)

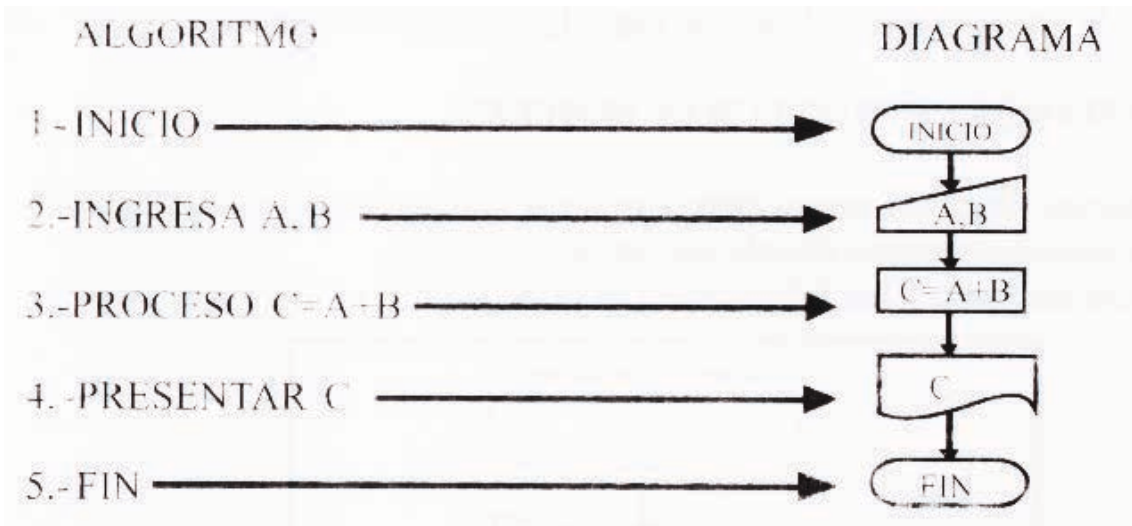
Diagramas de estructuras simples

Son diagramas sencillos con una estructura de arriba hacia abajo, solo sigue secuencialmente un solo camino hasta el final. No se utiliza decisiones, ni inicializaciones de rutinas, por ejemplo:

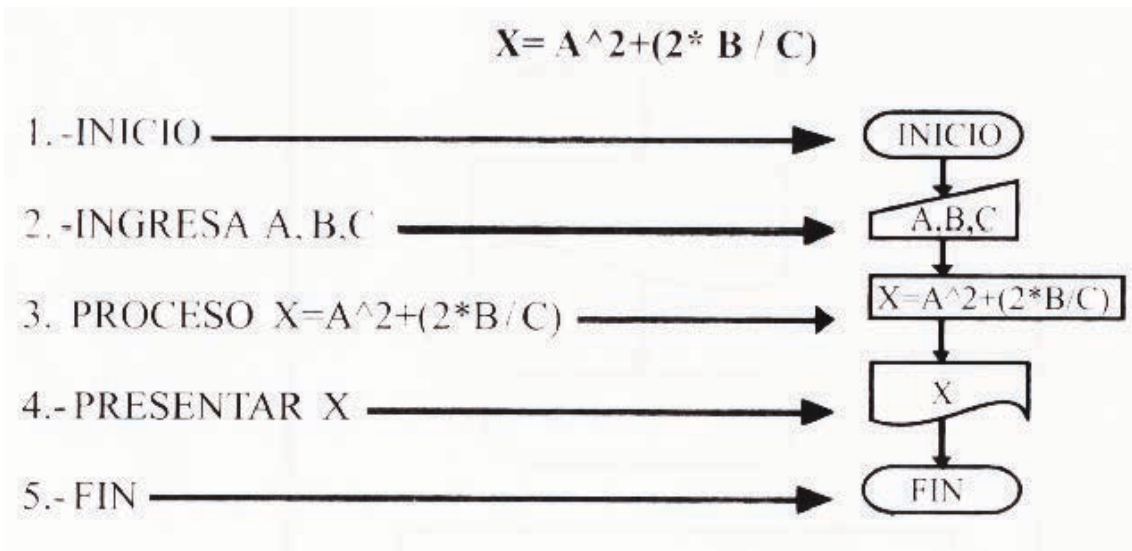


Para comenzar a diagramar tenemos que acordamos de las primeras clase sobre algoritmo

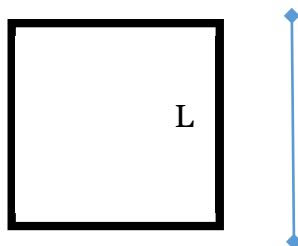
Realizar un diagrama de flujo de la suma de dos números cualquiera ingresados por teclado presentar el resultado, luego finalice.



Realizar un diagrama de flujo para resolver la siguiente expresión aritmética para cualquier dato ingresado, presentar el resultado y finalizar

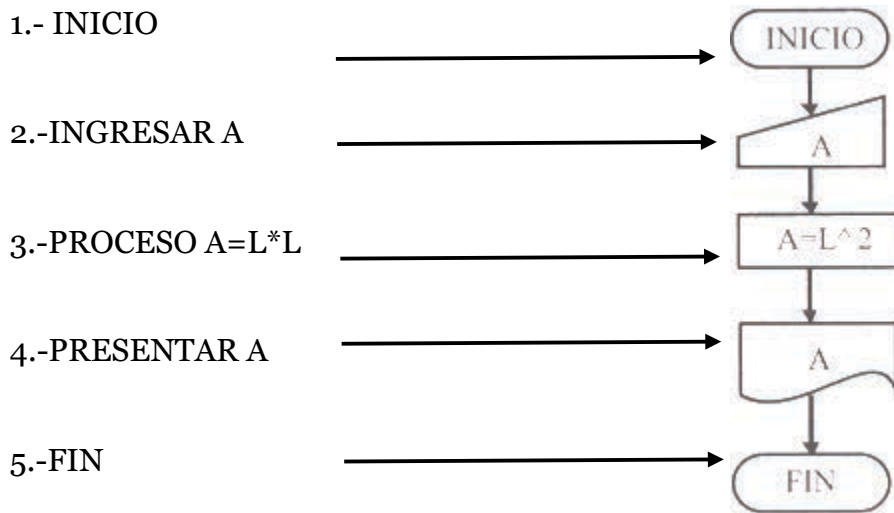


Realizar un diagrama de flujo que calcule el área de un cuadrado, presentar el resultado y finalizar



Área del cuadrado = lado al cuadrado

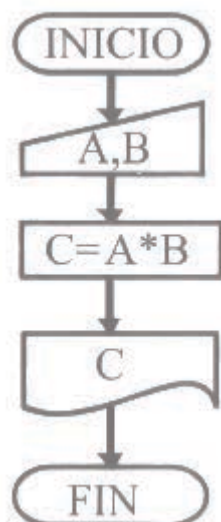
$$A=L^2$$



Prueba de escritorio

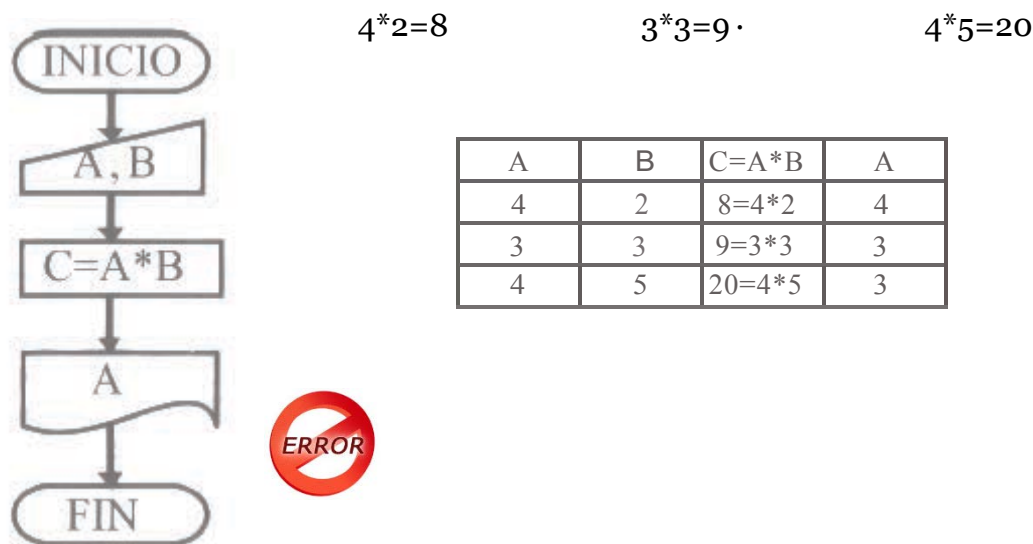
La prueba de escritorio es la comprobación escrita que se realiza al diagrama de flujo para detectar errores. Se utiliza una tabla donde intervienen todos los elementos del diagrama, ubicado de la misma manera como están estructurados pero de izquierda a derecha por ejemplo:

Realizar un diagrama de flujo para obtener la multiplicación de dos números, ingresarlos por teclado, presentarlos y finalizar. Luego mediante La prueba de escritorio compruebe si existen errores



Prueba de escritorio			
A	B	C=A*B	c
5	6	30=5*6	30
2	-5	-10=2*(-5)	-10

El diagrama siempre se lo lee de arriba hacia abajo siguiendo el flujo que indican las flechas. Además algo muy importante al realizar la prueba de escritorio es poder ejecutarla por lo menos tres veces y comparar en lo posible los resultados generados mentalmente o por otros medios. Qué pasaría si nuestro diagrama está incorrecto. En el siguiente ejemplo se ilustra esta posibilidad



Si comparamos los resultados de la tabla con los datos generados mentalmente observaremos que no son iguales, existe un error debido a que no se presenta el resultado de la multiplicación que requiere el problema. La manera correcta, es cambiar el símbolo de presentación por impresora la variable A por la C.

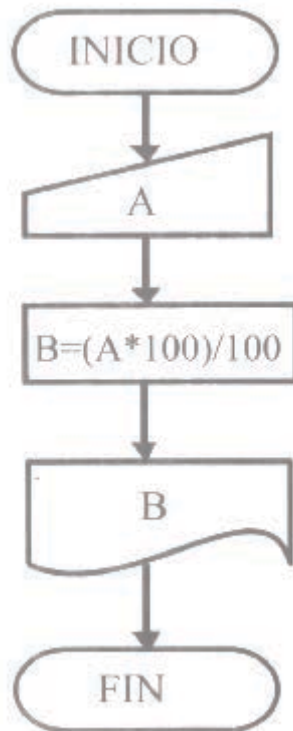
Ejercicio

En el trabajo de José Banderas se cobra el 10% por cada mercadería que venda, pero siempre se equivoca al calcular éste valor. Por tal motivo él necesita que se diseñe un diagrama de flujo para solucionar su problema

Primero determinemos que ingresamos. Ingresamos la mercadería que se representa con la variable A.

Segundo crear una fórmula para resolver el 10% $B = (A * 10) / 100$

Que resultado presentamos. Presentamos la variable B que es el resultado del 10%



Mercadería	Total (A)
100 Sodas	5000
100 Pollos	1200
100 Docena de huevos	300

A	$B=(A * 10) / 100$	B
5000	$500=(5000*10)/100$	500
1200	$120=(1200*10/100$	120
300	$30=300*10/100$	30

Actividad # 6

¿Qué es un Diagrama de Flujo?

Dibuje 3 símbolos usados en los diagramas de flujo

Realizar un diagrama de flujo para presentar su nombre y apellido

Ingrese las variables de las siguientes expresiones en el símbolo

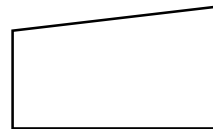
$$X = \frac{Z^2(B+A)}{Z^2 - Y^2 + Z * A}$$



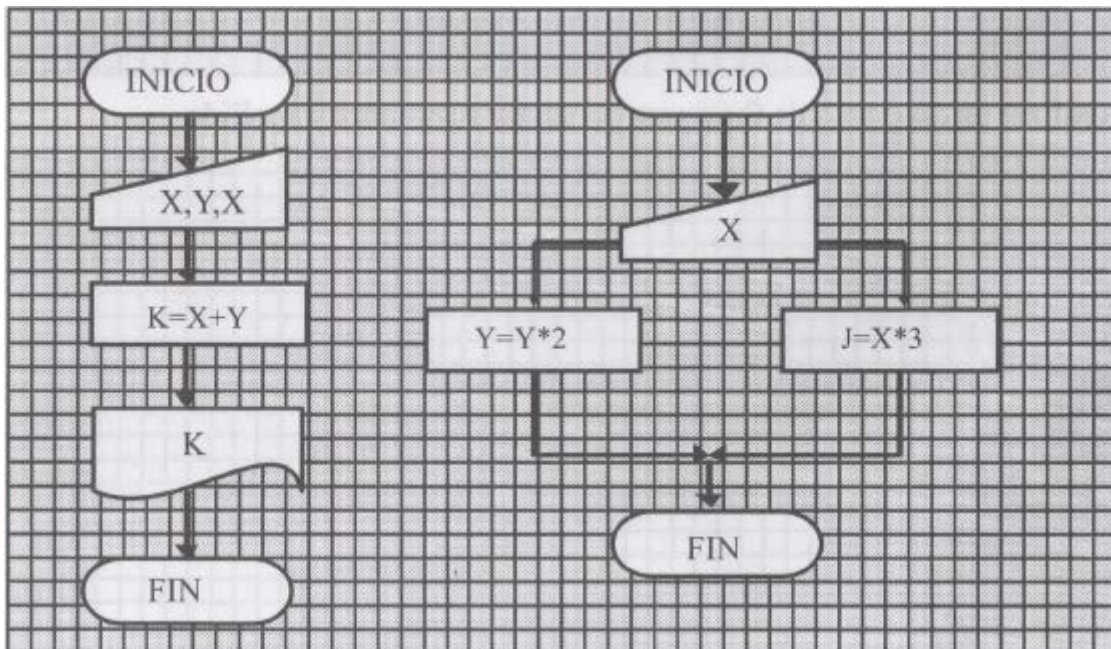
$$E = \frac{1/A}{\frac{\sqrt{K}}{C^{10}}}$$



$$C = (A+B+d + E) / (E+B)$$



Corrija el error del diagrama. Explique su razonamiento



Capítulo 7

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

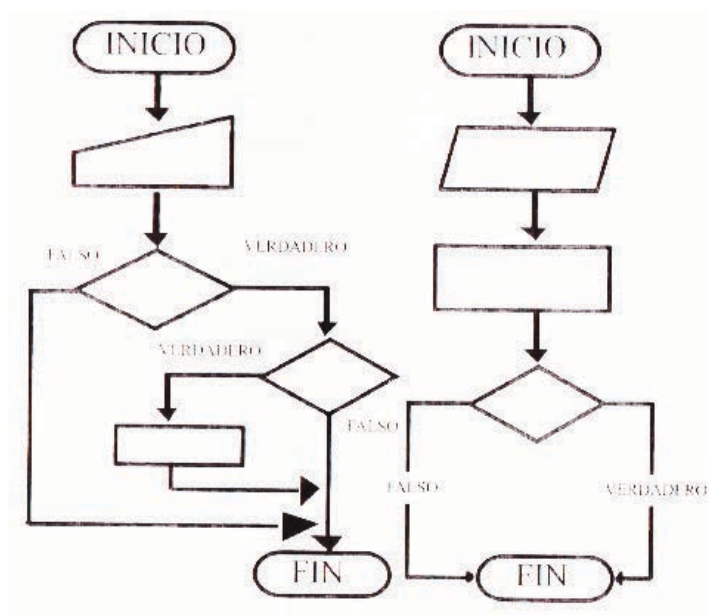
1. Obtener conocimiento teóricos de los diagrama de flujo complejos sin realimentación
2. Desarrollar de una forma práctica los diagrama de flujo complejos sin realimentación
3. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.

Diagramas de estructura complejas sin realimentación

Son diagramas que necesitan mayor tiempo para su elaboración, sus características principales es la utilización siempre del símbolo de la decisión y la dirección del flujo de datos de arriba hacia abajo.

La estructura del diagrama varía según la necesidad, existiendo muchas posibilidades para su diseño a diferencia que los diagramas simples que su estructura es estándar, además un diagrama complejo puede contener a uno o más diagramas simples.

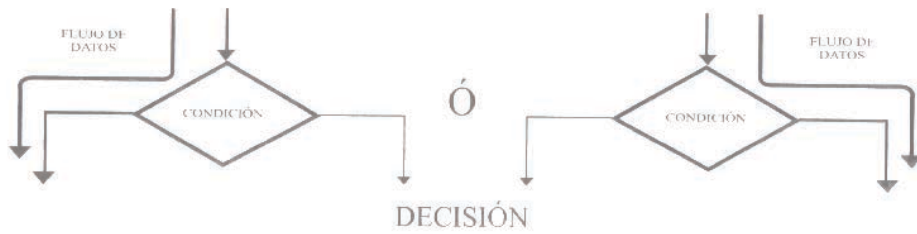
Figura 24: Diagramas de estructura complejas sin realimentación



Bucles no iterados

Se grafican de arriba hacia abajo y no realizan repeticiones, solamente sigue el camino o ubicación que determina la condición de la decisión.

Figura 25: Bucles no iterados



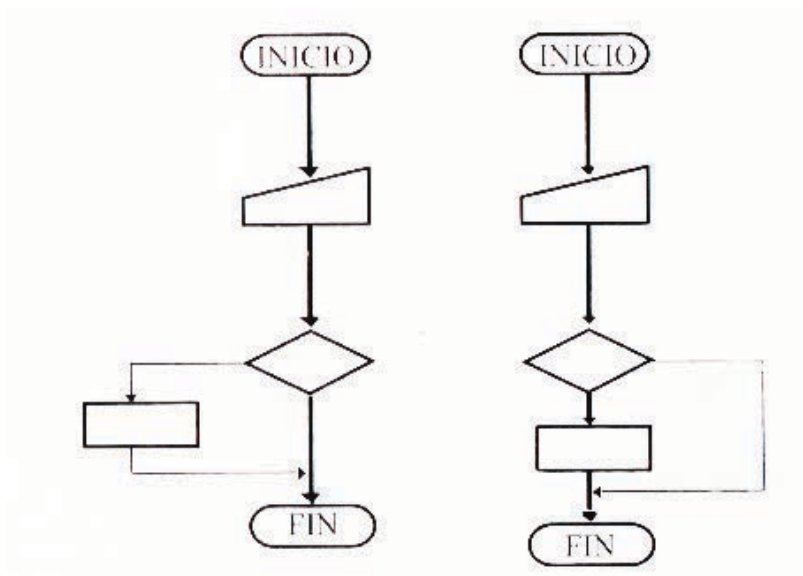
Clasificación de las estructuras complejas sin realimentación

Podemos clasificarlos en: decisiones simples, dobles y do case

Decisiones simples.-

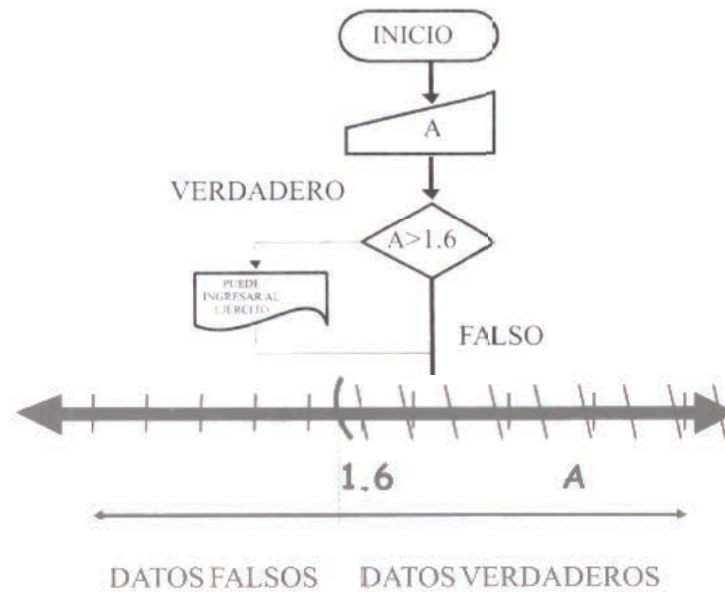
Es una decisión simple, si solamente por una de sus bifurcaciones se utilizan los símbolos.

Figura 26: Decisiones simples



Ejercicio# 1

En el ejército se necesita detectar de una forma automática a las personas que pueden ingresar al servicio militar según la siguiente condición, si tienen una estatura mayor a 1.60 metro se deberá presentar por impresora “PUEDE INGRESAR AL EJERCITO” caso contrario finalizar.

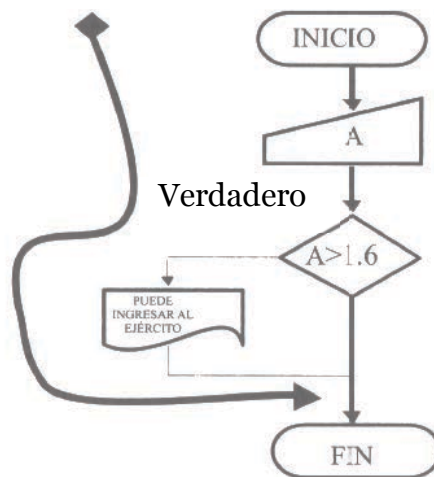


Cualquier dato ingresado se lo puede expresar mediante la variable A. Ingresamos la primera estatura $A=1.4$ este dato se desplaza a la decisión siguiendo la línea de flujo ,establecemos la condición que nos pide el problema, cualquier dato que ingrese puede tomar un camino determinado si cumple o no la condición. Si $A > 1.6$, es decir $1.4 > 1.6$ es falso y seguirá éste camino como lo demuestra la figura 27.Si deseamos ver que datos son verdaderos y falso nos fijamos en la recta numérica

Bucles no iterados

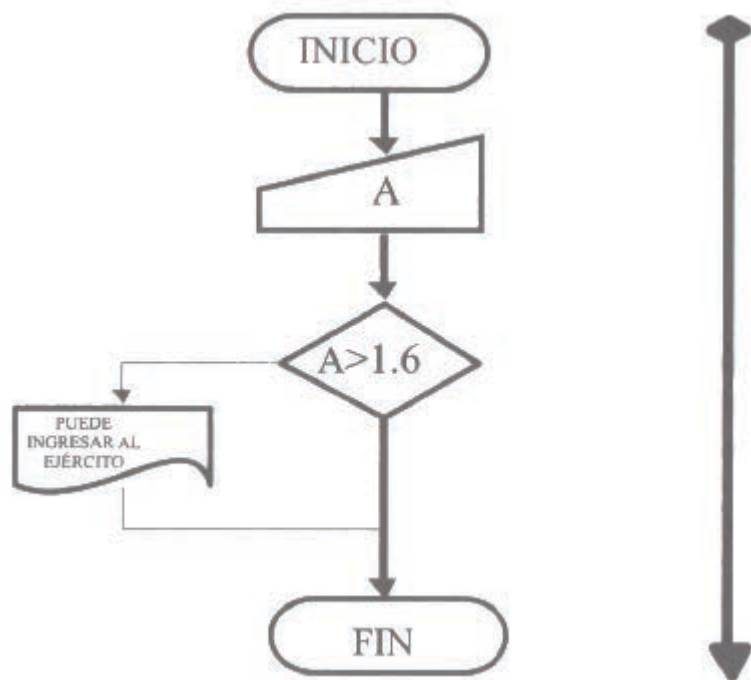
Se grafican de arriba hacia abajo y no realizan repeticiones, solamente sigue el camino o ubicación que determina la condición de la decisión.

Figura 27: Condición por el camino verdadero ejercicio #1



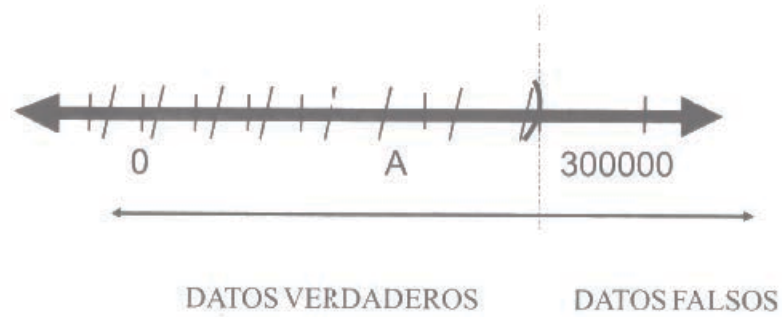
Se ingresa la primera estatura $A=2$ este dato se traslada automáticamente a la decisión siguiendo la línea de flujo se establece la condición que pide el problema, cualquier dato que se ingrese puede tomar un camino determinado si cumple o no la condición. Si $A > 1.6$, es decir $2 > 1.6$ es verdadero y seguirá éste camino como lo demuestra la Figura 27, cada vez que un dato elige un camino decidido por la condición, el otro se elimina hasta que finalice el diagrama de flujo.

Figura 28: Condición por el camino falso ejercicio #1

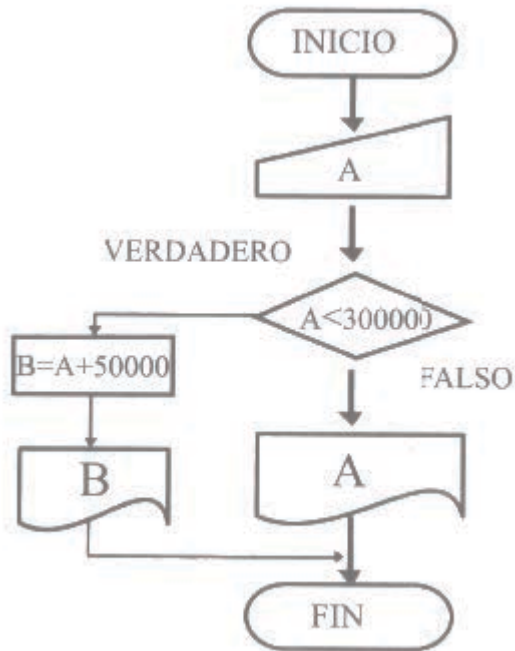


Ejercicio #2

En una empresa de automóviles a los trabajadores que ganan menos de 300000 dólares reciben un bono de 50000. Realice un diagrama para ingresar por teclado el sueldo del trabajador y verificar si necesita el bono, presentar el resultado por impresora y finalizar.



El sueldo del trabajador se representa con la variable A

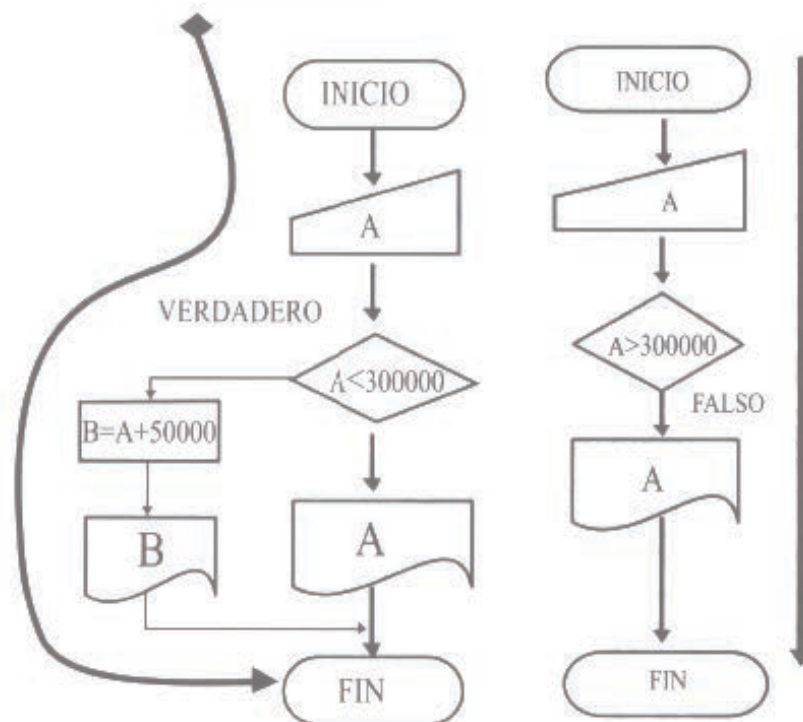


PRUEBA DE ESCRITORIO

A	$A < 300000$	VERDADERO $B = A + 50000$	F	PRESENT
450000	$450000 < 300000$		Si	A
200000	$200000 < 300000$	$250000 = 200000 + 50000$		B
100000	$100000 < 300000$	$150000 = 100000 + 50000$		B

DATOS MENORES QUE A

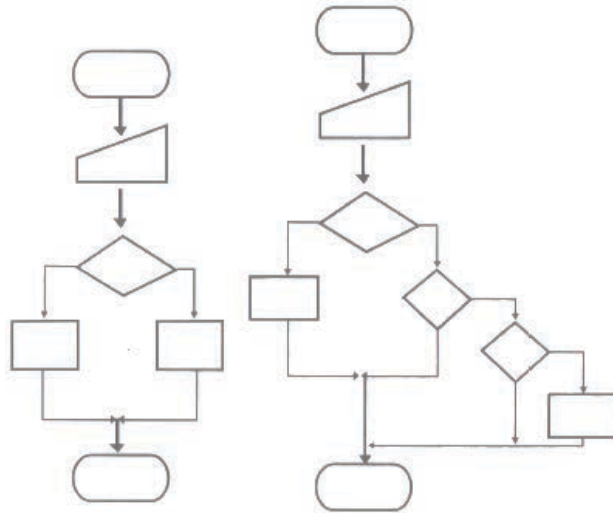
DATOS MAYORES QUE A



Decisiones dobles.-

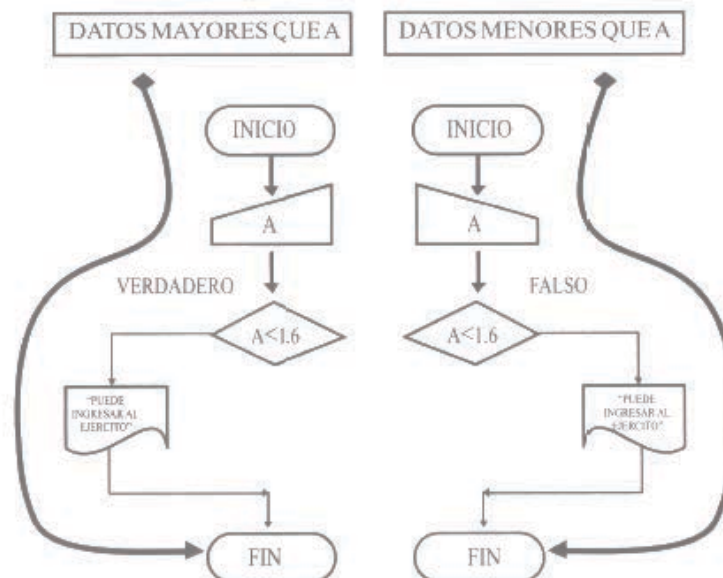
Es una decisión doble cuando se realiza una bifurcación doble y cada camino utiliza los símbolos. Una decisión doble puede contener una o varias decisiones simples.

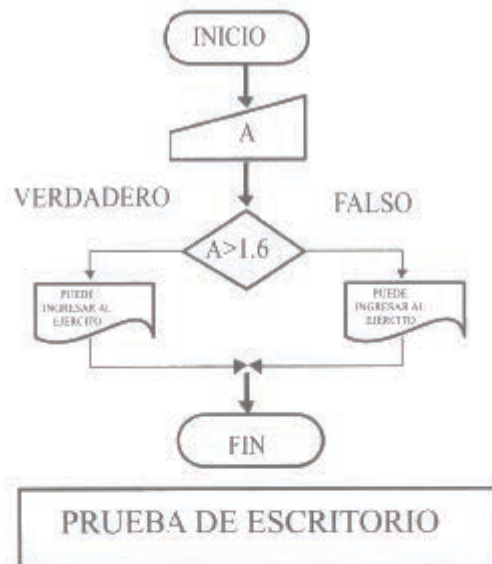
Figura 29: Estructuras con decisiones dobles



Ejercicio# 1

En el ejército se necesita detectar automáticamente a las personas que pueden ingresar al servicio militar si tienen una estatura mayor a 1.60 metros. Se deberá ingresar por teclados cada uno de los datos, si es mayor a 1.60 se deberá presentar por impresora PUEDE INGRESAR AL EJERCICIO caso contrario presentar NO PUEDE INGRESAR AL EJERCITO luego finalice





A	$A > 1.6$	VERDADERO PRESENTA	FALSO PRESENTA
1.4	$1.4 > 1.6$		NO INGRESA AL EJERCITO
2	$2 > 1.6$	INGRESA AL EJERCITO	
1.78	$1.78 > 1.6$	INGRESA AL EJERCITO	



Ejercicio #2

Realizar un diagrama de flujo para detectar automáticamente cual de dos números ingresados por teclado es el mayor presentar por impresora el mensaje MAYOR, a continuación la variable y finalice

Nota: no considere ingresar valores iguales para este ejercicio.

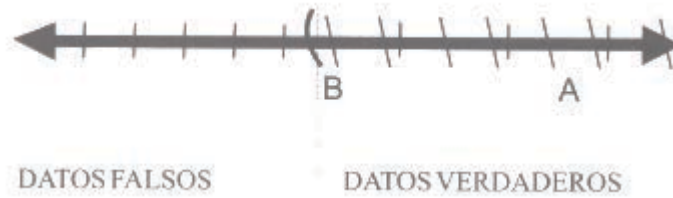
Establezcamos las variables que intervienen:

Primer dato ingresado por teclado: A

Segundo dato ingresado por teclado: B

Como podemos establecer cual es el mayor Debido hay dos posibilidades que el dato ingresado resultase mayor que B o menor que B entonces utilizamos una decisión.

Mediante una condición $A > B$



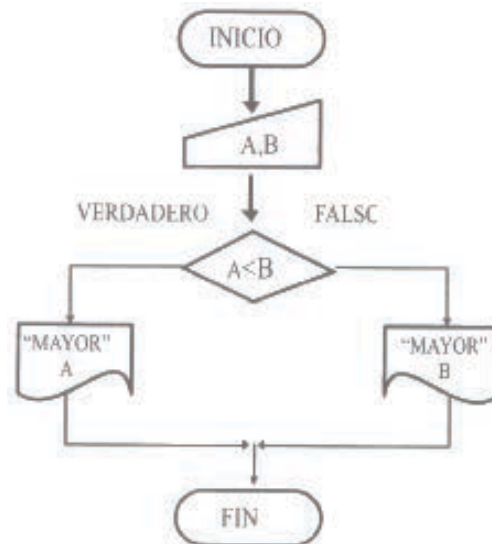
Todos los valores de A que caen dentro del área rayada serán mayores que B por ejemplo

$A=19$ y $B=10$

$A > B$ $19 > 10$

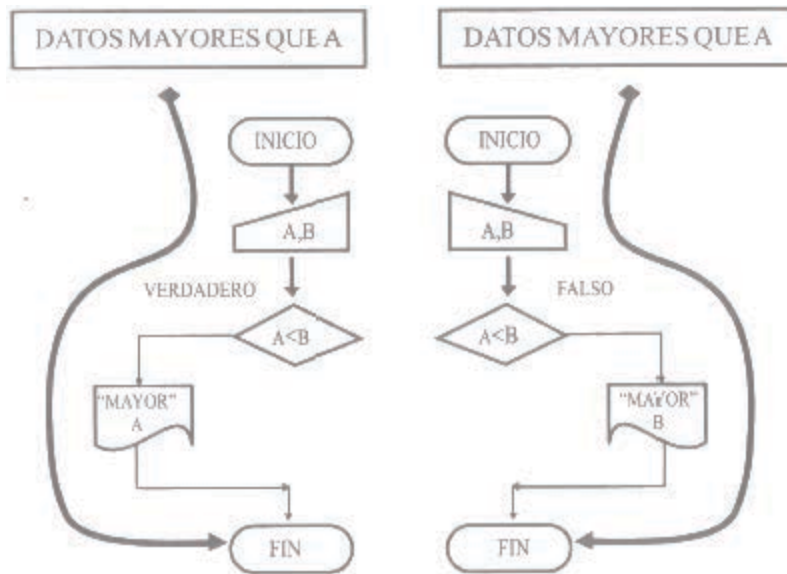


Una vez analizado el problema



PRUEBA DE ESCRITORIO

A	B	$A > B$	VERDADERO PRESENTA	FALSO PRESENTA
19	10	$19 > 10$	MAYOR 19	
5	15	$5 > 15$		MAYOR 15
34	2	$34 > 2$	MAYOR 34	



EJERCICIO #3

Realizar un diagrama de flujo para detectar automáticamente cual de dos números ingresados por teclado es el mayor , presentar por impresora el mensaje MAYOR a continuación la variable, si los números son iguales presentar IGUALES y finalice.

Establezcamos las variables que intervienen:

Primer dato ingresado por teclado: A

Primer dato ingresado por teclado: B

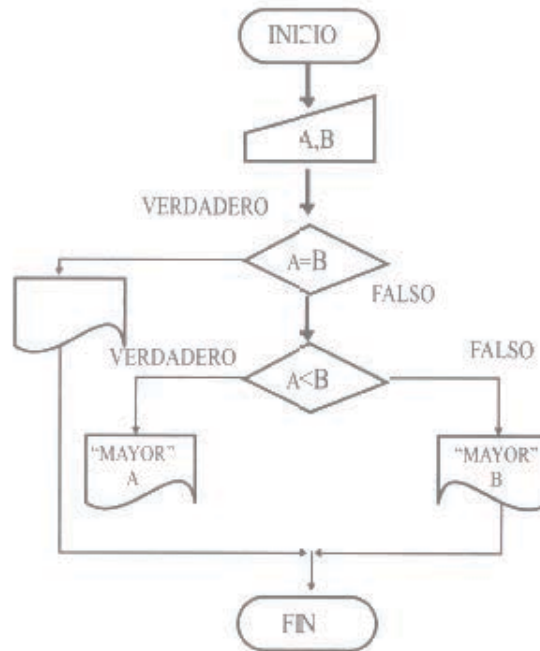
En este ejercicio tenemos tres posibilidades: mayor que B, menor que B e igual que B.

Como las decisiones con la que trabajaremos solo tienen una bifurcación doble faltaría un camino, por lo tanto utilizamos otra decisión para cumplir la resolución de nuestro problema.

Las condiciones dentro de las decisiones serán:

Primera condición $A > B$

Segunda condición $A = B$

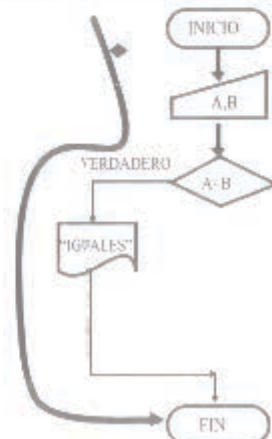


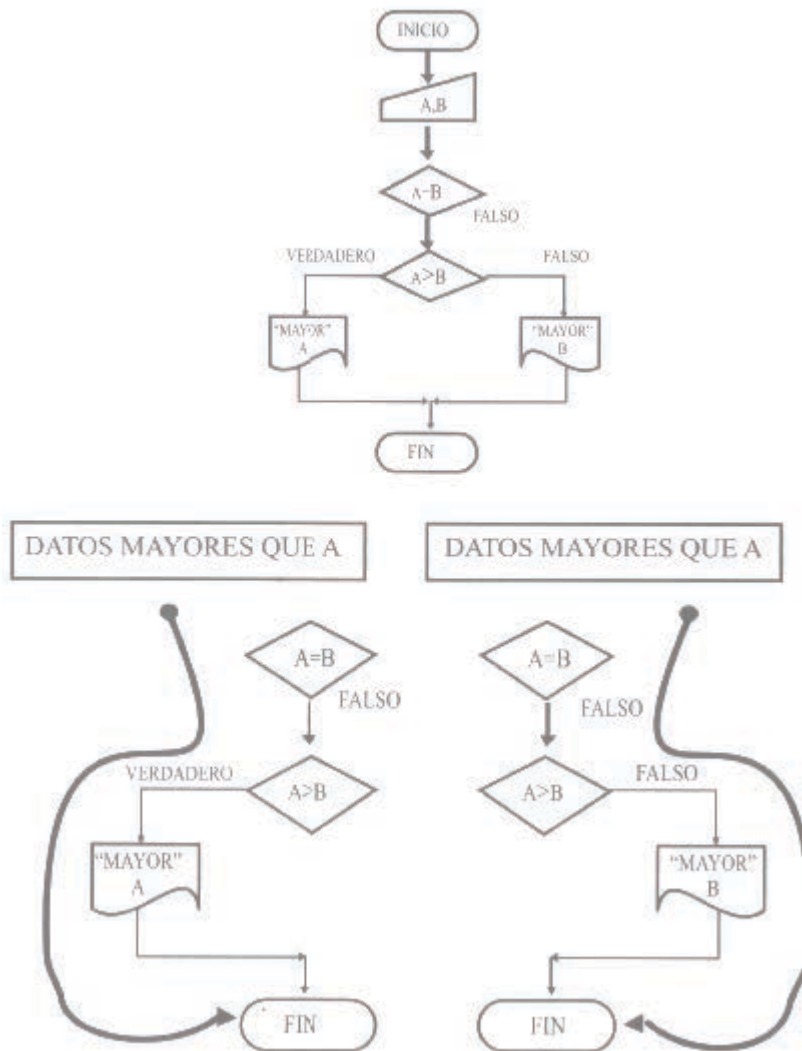
PRUEBA DE ESCRITORIO

A	B	A=B	VERDADERO PRESENTA	FALSO PRESENTA
5	5	5=5	IGUAL	
12	24	12=24		IR A LA OTRA DECISIÓN
78	59	78=59		IR A LA OTRA DECISIÓN

A>B	VERDADERO PRESENTA	FALSO PRESENTA
12> 24		MAYOR 24
78> 24	MAYOR 78	

DATOS IGUALES

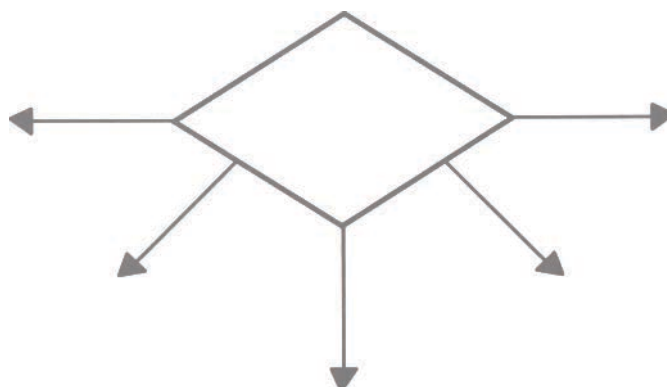




Decisiones Do Case.-

Es una decisión DO CASE cuando consta de bifurcaciones múltiples, es decir, más de dos caminos posibles por ejemplo:

Figura 29: Decisiones Do Case



Actividad #7

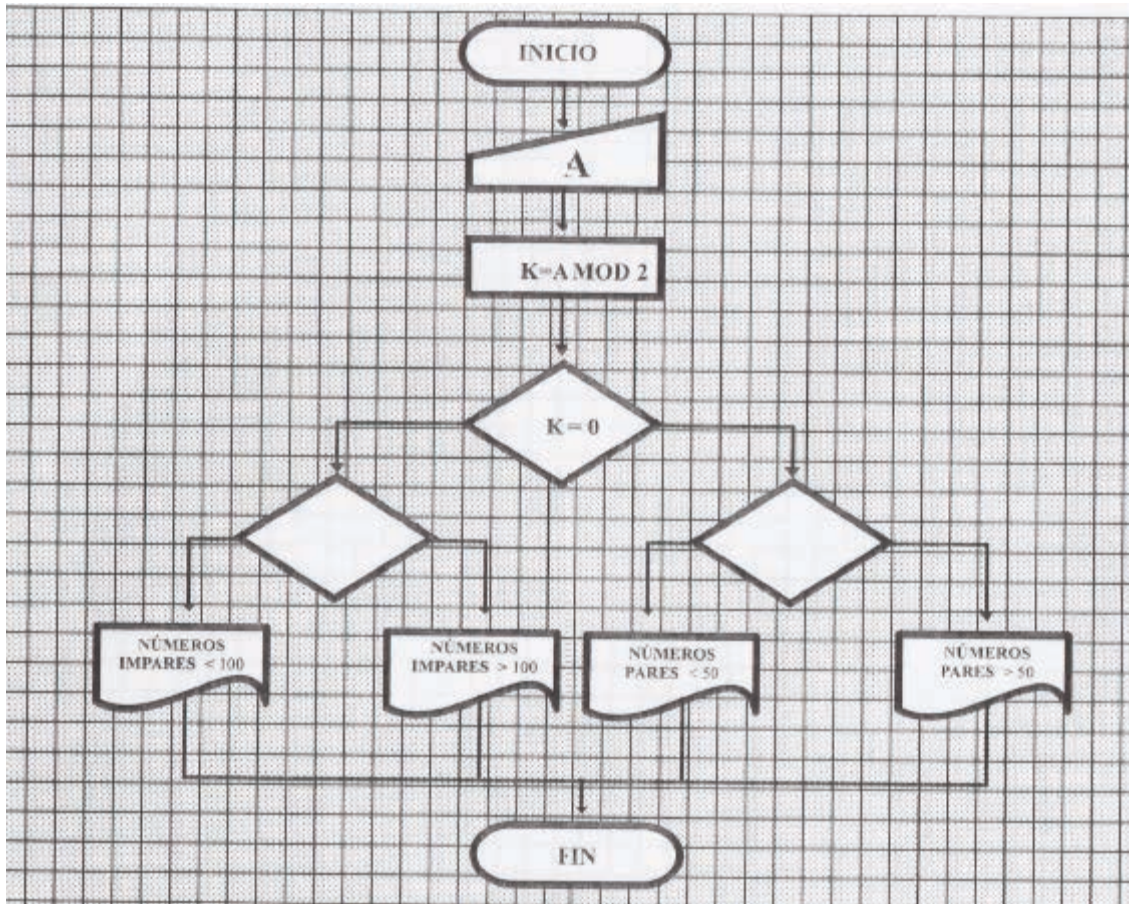
¿Qué es un diagrama de estructura Compleja sin retroalimentación?

Realizar un diagrama de Flujo para ingresar 3 notas, si la suma de las tres notas es mayor o igual a 40 presentar pasa de año. Luego finalizar. Comprobar utilizando la prueba de escritorio

Realizar un Diagrama de flujo para presentar el resultado de la suma de dos números, si el resultado es mayor a 30 por verdadero multiplique los números caso contrario divida el primer número para el segundo número, presente el resultado y finalice. Comprobar el resultado utilizando la prueba de escritorio

Completar el siguiente diagrama de Flujo para que presente si un número es par o impar, si es par y mayor a 50 mostrar NUMERO PAR MAYOR A 50 , caso contrario presentar NÚMERO PAR MENOR A 50.

Si es par y mayor a 109 presentar NÚMERO IMPAR MAYOR A 100, caso contrario presentar NÚMERO IMPAR MENOR A 100. Compruebe utilizando la prueba de escritorio



Diseñe un diagrama de Flujo para ingresar la edad de una persona y detectar si es mayor o igual a 40 presentar "persona madura", si está entre 18 y 40 presente "Adulto", si es menor o igual a 18 presente "Joven" luego finalice. Realizar prueba de escritorio

Comprobar si el diseño del diagrama de flujo es correcto.

Un hacendado desea calcular el área de sus haciendas, pero tiene un pequeño problema que sus formas son diversas, cuadradas, rectangulares y circulares. Realice la prueba de escritorio

C	C=1	FALSO IR C=2	VERDAD M	VERDAD $K=M^2$	PRESENTAR K	FIN

C=2	FALSO IR C=3	VERDAD X	VERDAD Y	VERDAD $Z=X*Y$	VERDAD Z	FIN

C=3	S	VERDAD R	VERDAD $Q=S*R/2$	VERDAD Q	FIN

Capítulo 8

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

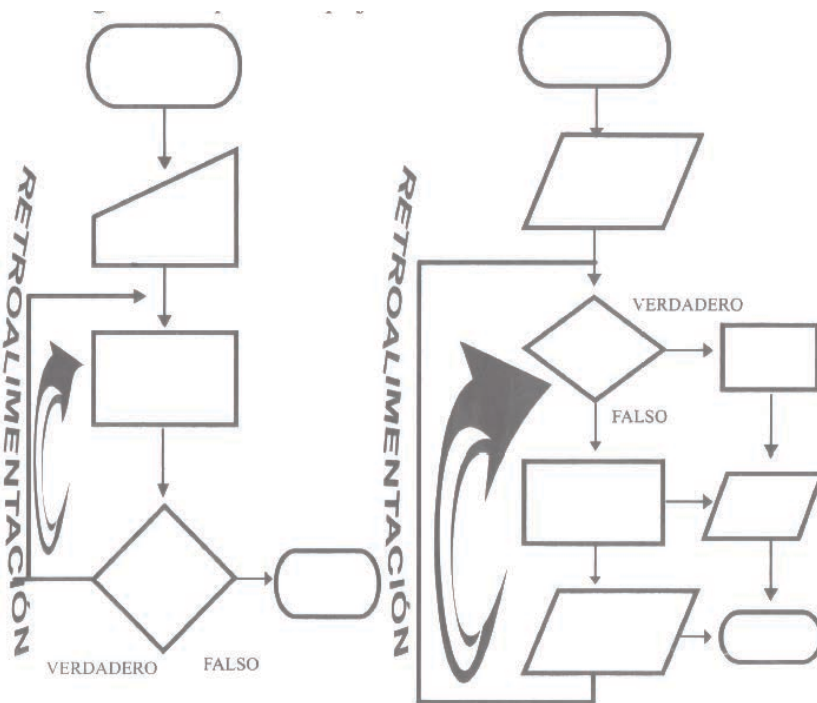
1. Obtener conocimiento teóricos, para desarrollar de una forma práctica, los diagrama de flujo complejo con realimentación
2. Al finalizar la unidad el lector será capaz de tener un criterio claro sobre diagramas de flujo complejos con realimentación
3. Desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.

Diagramas de estructura complejas con realimentación (repetitivas)

El computador es una máquina electrónica que realiza las operaciones con una exactitud y velocidad admirables. En la mayoría de los programas se utilizan las repeticiones y estructurar estos diagramas son un poco más laboriosos, la ventaja es que su estructura puede realizarse primero de arriba hacia abajo y luego de abajo hacia arriba para las repeticiones, es decir, la dirección de las líneas de flujo es variada.

Un diagrama de estructura compleja con realimentación puede tener uno o varios diagramas simples o complejos sin realimentación

Figura 30: Estructuras de diagramas complejos con realimentación



Podemos dividir a las estructuras de diagramas complejo con realimentación en: decisiones con bucles iterativos y ciclo For

Decisiones con bucles iterativos

Ayuda a representar en forma gráfica las repeticiones o iteraciones consecutivas en el diagrama de flujo. Entre las partes importantes tenemos: decisión, cuerpo del bucle y salida del bucle

Decisión.-

Dentro del diagrama de flujo debe existir el símbolo de decisión caso contrario, el bucle será infinita, es decir, se repetirá una y otra vez sin poder salir al final. Por un camino realizará las iteraciones y una vez cumplidas las repeticiones saldrá por la otra bifurcación por ejemplo

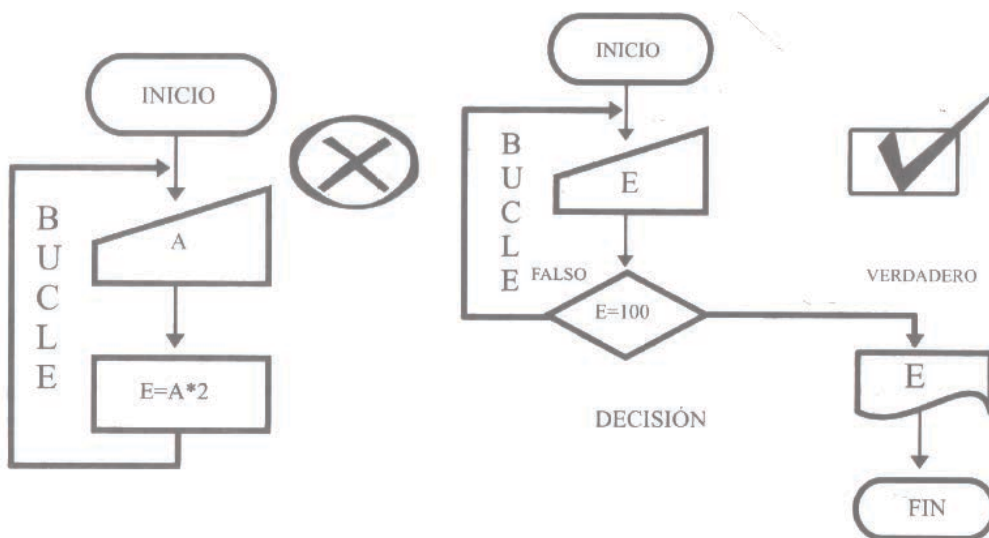


Figura 31: Bucle infinito

Figura 32: Bucle correcto

Cuerpo del bucle-

Se tratará independientemente según el tipo bucle de iterativo.

Salida del bucle.-

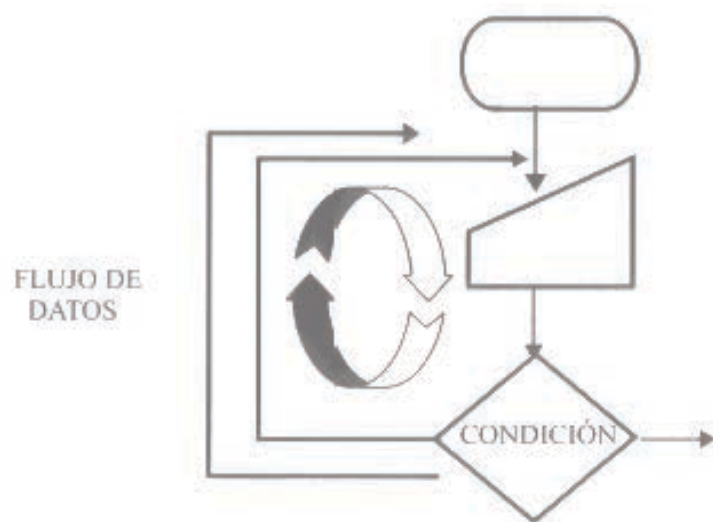
Nos determina la manera para salir del lazo caso contrario volvería a tener un bucle infinito. Se realiza por una de las salidas de la decisión

Los bucles iterativos se clasifican en: bucles iterativos de control e iterativos complejos.

Bucles iterativos de control

Ayudan a controlar el ingreso de datos, que no están considerados en el diagrama de flujo, para que funcione normalmente

Figura 33: Estructura de los bucles iterativos de control



Cuerpo de los bucles iterativos de control.-

Realiza la detección de datos no deseados, teniendo como elemento básico la decisión logrando repetir el ingreso hasta tener datos correctos que funcionen en el diagrama.

Ejercicio# 1

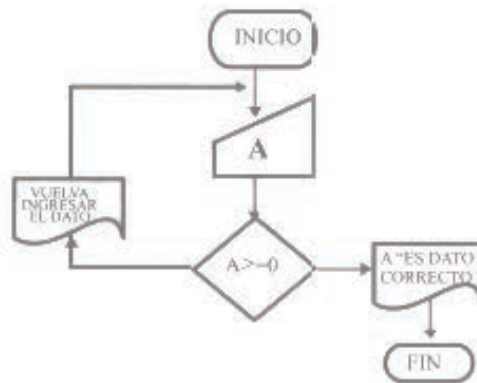
Realizar un diagrama de flujo para ingresar por teclado al diagrama solo números positivos y presentar un mensaje de DATOS CORRECTOS, caso contrario presentar VUELVA A INGRESAR EL DATO, luego finalice.

Se ingresan valores numéricos representador por cualquier variable, en este caso se utilizara A.



En este caso se asume que el número cero es positivo. Observando la recta numérica concluimos que tenemos dos posibilidades que pueden ser válidas como condición de la decisión: $A \geq 0$ o $A < 0$

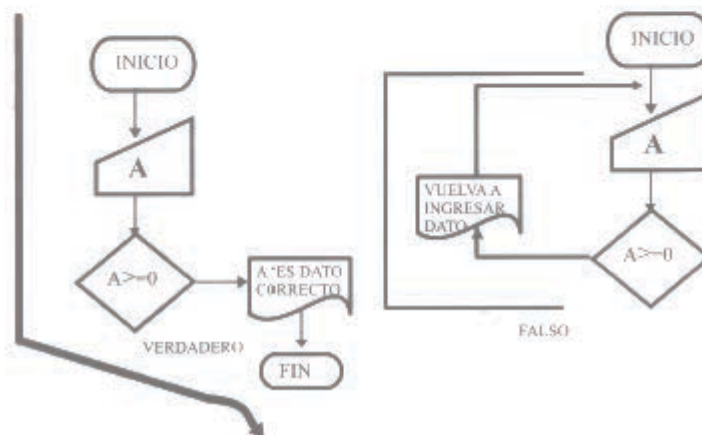
Resolución de la primera forma #1 $A \geq 0$



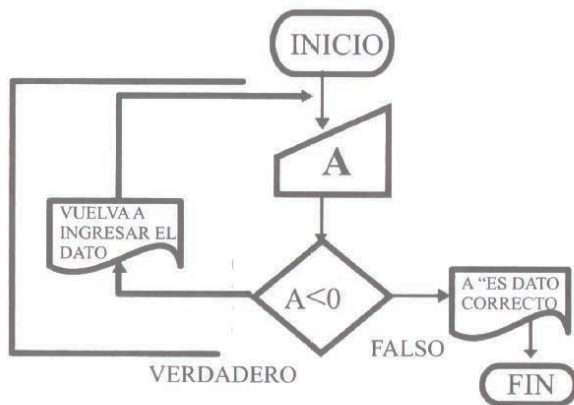
A	$A \geq 0$	VERDADERO PRESENTA	FALSO PRESENTA	FIN
3	$3 \geq 0$	A ES DATO CORRECTO		SI
0	$0 \geq 0$	A ES DATO CORRECTO		SI
-3	$-3 \geq 0$		VUELVA A INGRESAR EL DATO	NO
24	$24 \geq 0$	A ES DATOS CORRECTO		SI

DATOS CORRECTOS

DATOS INCORRECTOS



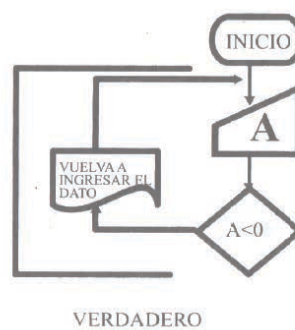
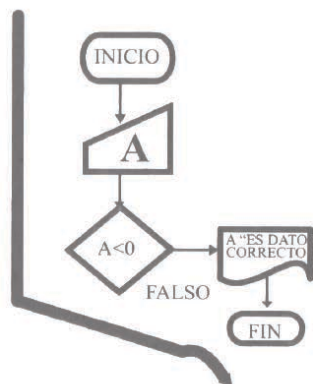
Para salir del bucle iterado de control de los datos incorrectos debe cumplirse con la condición de la decisión de $A \geq 0$.



A	$A \leq 0$	VERDADERO PRESENTA	FALSO PRESENTA	FIN
3	$3 \leq 0$		A ES DATO CORRECTO	SI
0	$0 \geq 0$		A ES DATO CORRECTO	SI
-3	$-3 \geq 0$	VUELVA A INGRESAR EL DATO		NO
24	$24 > 0$		A ES DATOS CORRECTO	SI

DATOS CORRECTOS

DATOS INCORRECTOS



Ejercicio#2

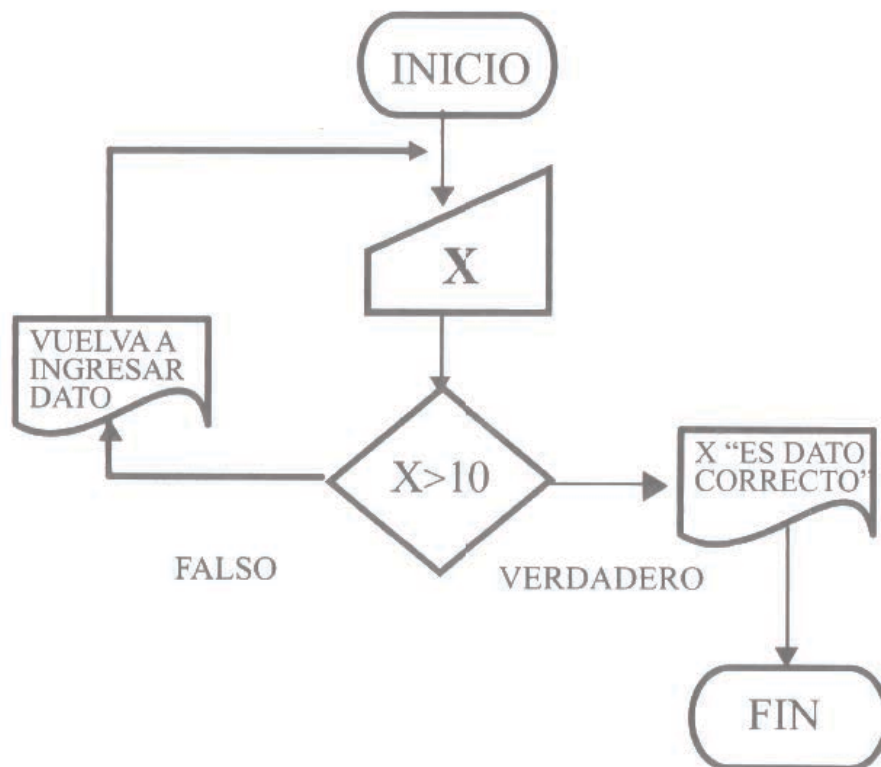
Realizar un diagrama de flujo para ingresar números mayores a 10, si el número es mayor al 10 presentar MAYOR A 10 caso contrario presentar MENOR A 10, luego finalizar.

Se utiliza una variable para representar cualquier número: X

Visualicemos los números mayores a 10, mediante la recta numérica.



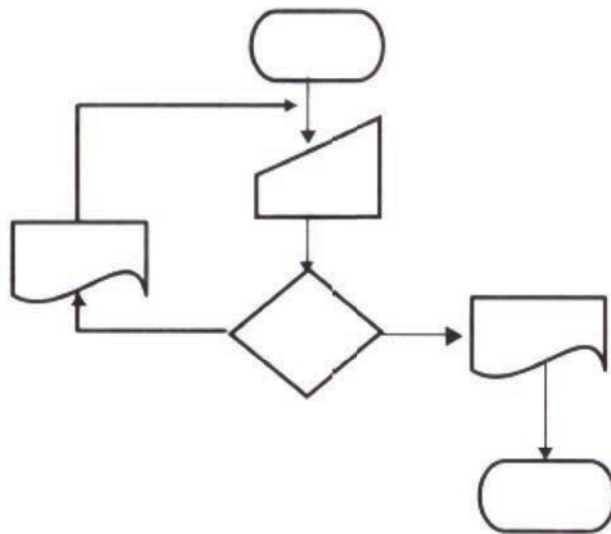
Observando la recta numérica concluimos que tenemos dos posibilidades que se pueden ser validas como condición de la decisión: $X > 10$ o $X \leq 10$ Resolución de la primera forma $X > 10$



A	$X \geq 10$	VERDADERO PRESENTA	FALSO PRESENTA	FIN
3	$3 > 0$	X ES DATO CORRECTO		SI
10	$10 > 10$		VUELVA A INGRESAR EL DATO	SI
-3	$-3 > 0$		VUELVA A INGRESAR EL DATO	NO
24	$24 > 0$	X ES DATOS CORRECTO		SI

Actividad

Tomando con guía el ejercicio #2 de los bucles iterativos de control, Realice el cambio de las condiciones

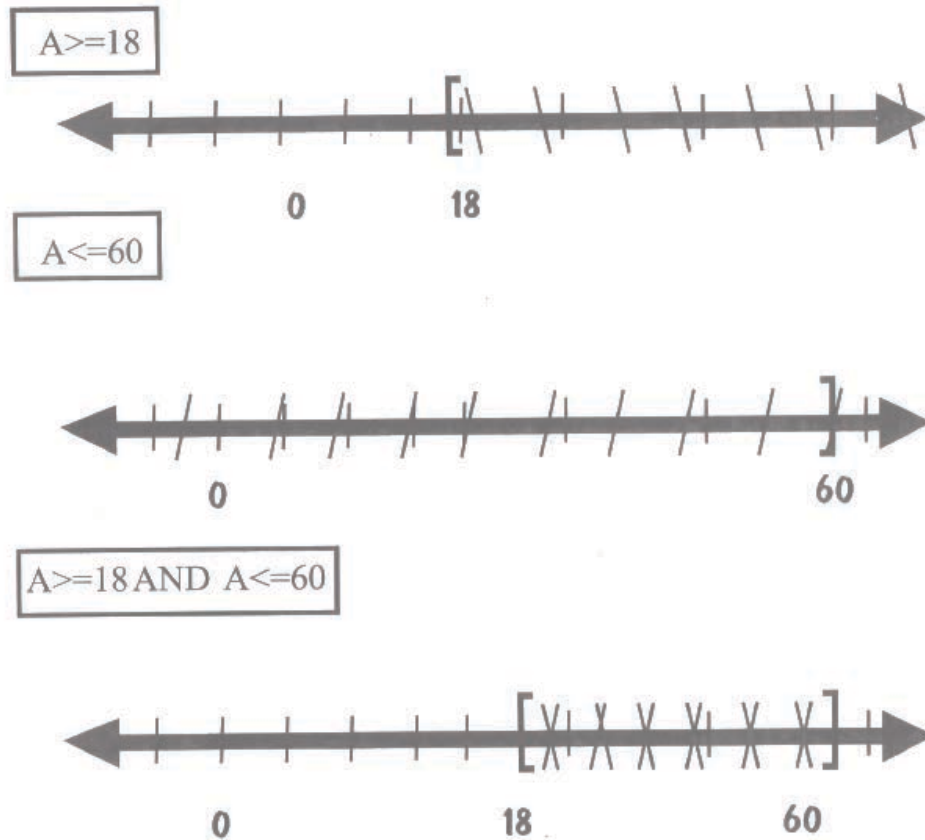


Ejercicio#3

En el tribunal electoral se desea saber si la edad de las personas ingresadas a la computadora por periférico que votan, están dentro del rango obligatorio, es decir, mayor o igual a 18 y menor o igual a 60, si cumple con estas dos condiciones presentar PUEDE VOTAR, caso contrario presentar NO PUEDE VOTAR, luego finalice.

Determinemos las variables que intervienen en el problema Ingreso de la edad:

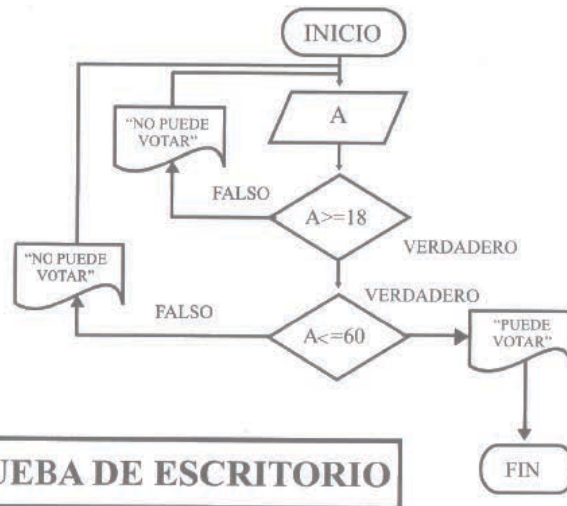
A



La edad debe estar dentro del rango establecido para votar. Observando la recta numérica concluimos que tenemos cuatro posibilidades entonces utilizaremos dos

Las condición de la decisión que pueden ser validas son primera condición $A \geq 18$ o $A < 18$ segunda condición $A \leq 60$ o $A > 60$

Primera forma



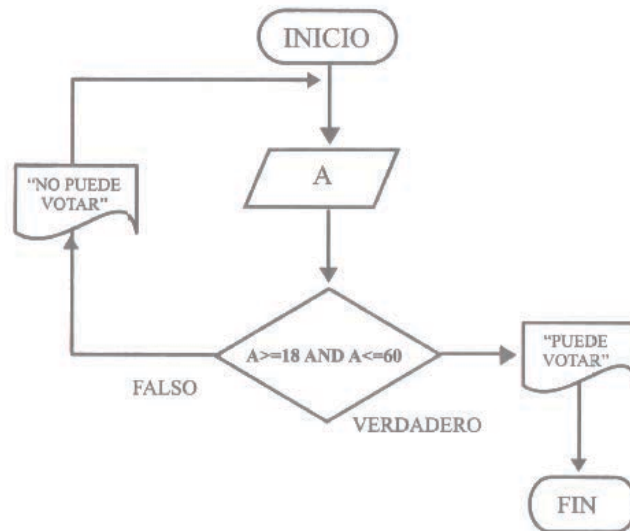
PRUEBA DE ESCRITORIO

A	A >= 18	VERDADERO IR	FALSO PRESENTA	FIN
23	23 >= 18	DECISIÓN A <= 60		NO
10	10 >= 18		NO PUEDE VOTAR	NO
70	70 >= 18	DECISIÓN A <= 60		NO
54	54 >= 60	DECISIÓN A <= 60		

DECISIÓN A <= 60	VERDADERO PRESENTA	FALSO PRESENTAR	FIN
23 <= 60	PUEDE VOTAR		SI
70 <= 60		NO PUEDE VOTAR	NO

Otra alternativa de realizar el ejercicio #3 de los bucles iterativos de control es utilizar operadores AND. Cuando se trata de los operadores AND se tienen cuatro posibilidades y se utiliza una sola decisión como observamos en la recta numérica. En el capítulo de expresiones lógicas se puede observar la tabla AND

Se debe tener el cuidado al diferenciar entre operadores AND y OR para no cometer errores. En el operador AND se deben cumplir todas condiciones que intervienen en la decisión

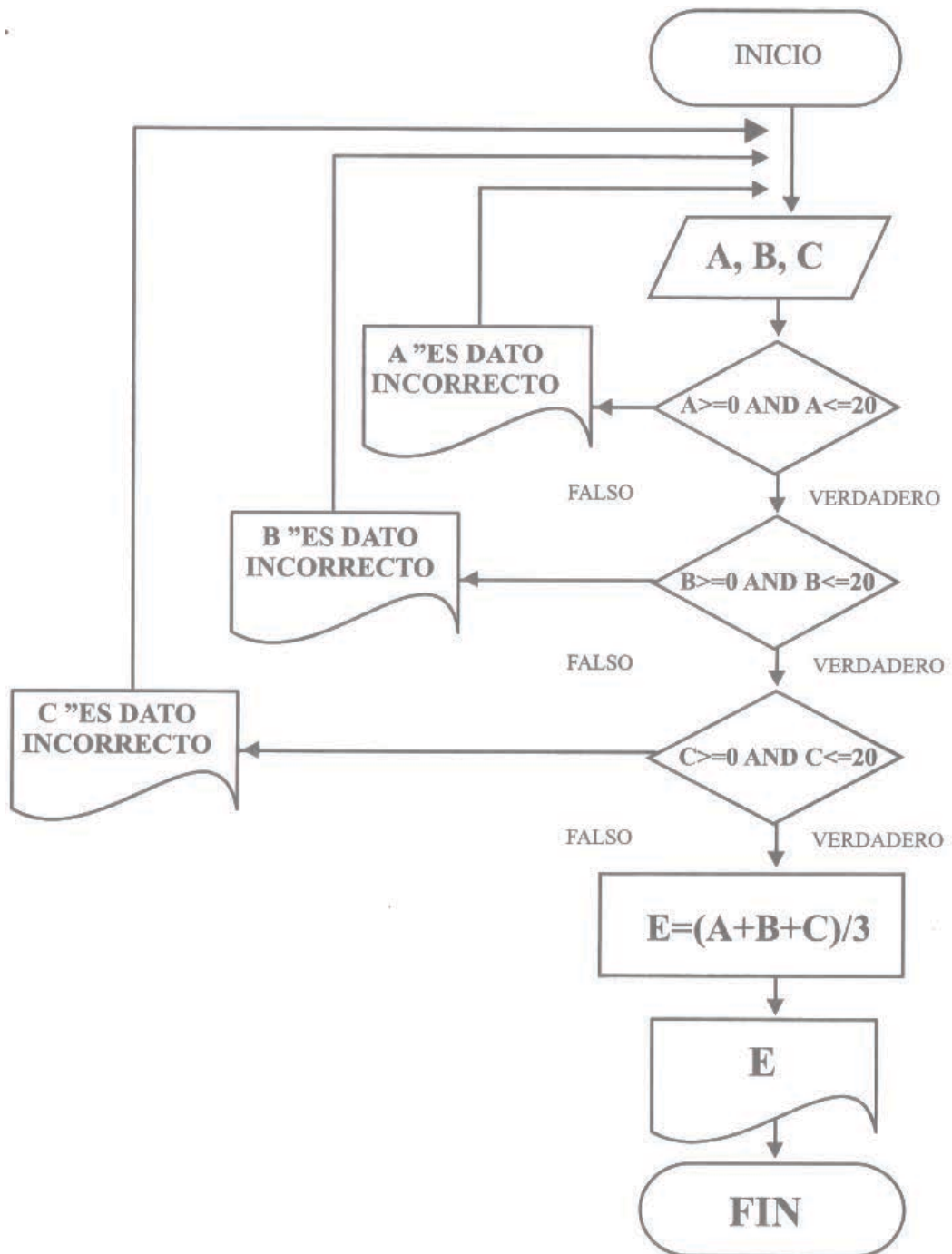


PRUEBA DE ESCRITORIO

A	A >= 18 AND A <= 60	VERDADERO	FALSO	FIN
23	23 >= 18 AND 23 <= 60 V V	PUEDE VOTAR		SI
10	10 >= 18 AND 10 <= 60 F V		NO PUEDE VOTAR	NO
70	70 >= 18 AND 70 <= 60 V V		NO PUEDE VOTAR	NO
54	54 >= 60 AND 54 <= 60 V V	PUEDE VOTAR		SI

Ejercicio#4

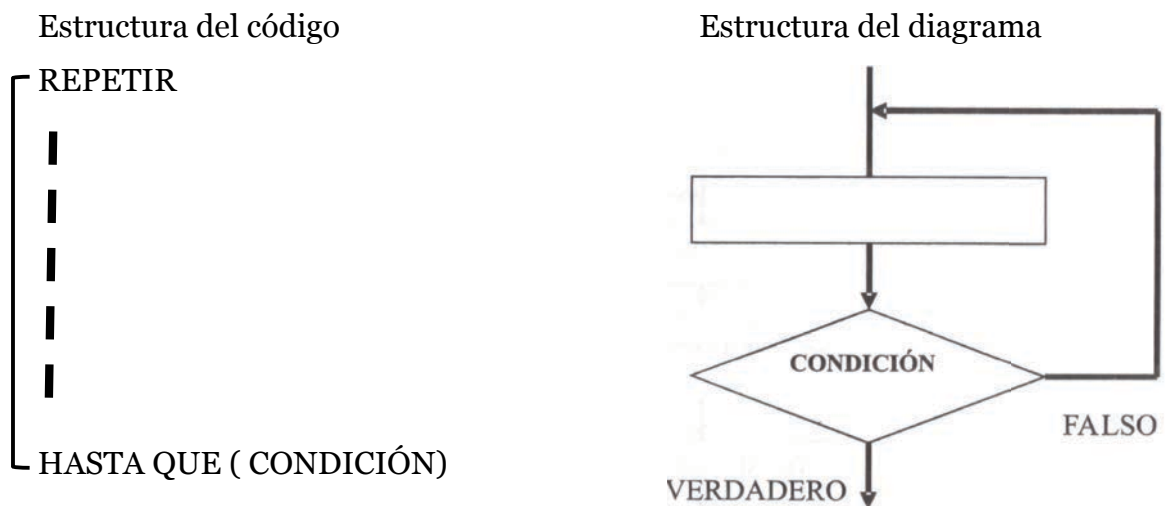
Construir un diagrama de flujo para presentar el promedio de tres notas conociendo que el máximo de la calificación es 20 y la mínima 0. Ingrese los datos por teclado, si cumple con la condición mostrar el promedio y el resultado, caso contrario presentar DATO INCORRECTO y vuelva a ingresar datos.



Es recomendable al establecer la estructura del diagrama con bucle poder diferenciar la más conveniente para la solución del problema, su clasificación varía pero entre las más importantes se tiene: estructura repetir y mientras hacer

Estructura repetitiva REPETIR – HASTA QUE

Figura 34: Estructura REPETIR-HASTA QUE



Características estructura repetitiva REPETIR-HASTA QUE

La condición posee operadores relacionales y lógicos contrarios a los de la estructura mientras

El bucle se ejecuta por falso

Los procesos se desarrollan en el ramal principal

Se sale por verdadero

Se ejecuta una vez y luego comprueba la condición para poder ingresar de nuevo al bucle

Características Estructura repetitiva MIENTRAS -HACER

La condición posee operadores relacionales y lógicos contrarios a los de la estructura repetir

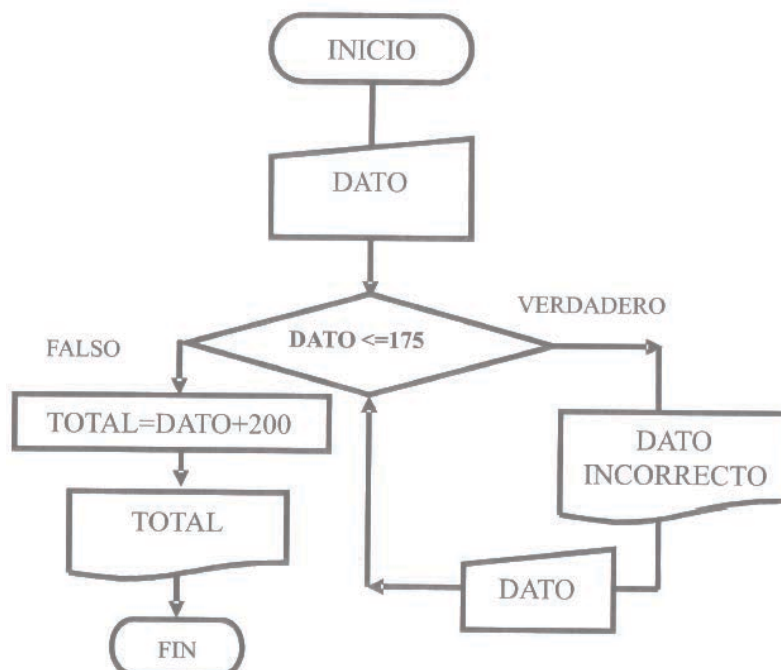
El bucle se ejecuta por verdadero. Los procesos se desarrollan en el ramal verdadero. Se sale por falso.

Comprueba la condición para poder ingresar al bucle.

La estructura mientras repetir verifica si la condición es falsa o verdadera, si es verdadera ejecuta los procesos y regresaran los datos para que ingresen de nuevo al ciclo de bucle hasta que la condición sea falsa y pueda salir del ciclo y continuar con el siguiente proceso

Se necesita inicializar la decisión, es decir colocarle un valor inicial manualmente o automáticamente para que pueda ingresar al bucle

Por ejemplo.: Realizar el diagrama de flujo para ingresar valores mayores a 175, si se cumple sume 2000 al dato, presente y finalice.



Actividades # 8

Conteste

¿Que es un diagrama de estructura compleja con realimentación?

¿Que es un bucle iterarlo?

Resolver

Se ingresan por teclado 2 números que tienen como máximo 2 dígitos, diseñe el diagrama de flujo presente los números correctos y finalice.

Realizar un diagrama de flujo para ingresar por teclado números de un mínimo de tres dígitos y máximo de cinco dígitos presentar y finalizar. Comprobar mediante la prueba de escritorio.

Realizar un diagrama de flujo que permita para ingresar solamente el número cuatro presentar y finalizar

Realizar el diagrama para ingresar números mayores e iguales a -10 y menores iguales a 10, presentar el dato correcto y finalizar. Comprobar mediante la prueba de escritorio

Realizar un diagrama de flujo que permita ingresar solamente el número cero o el uno, presentar y finalizar. Comprobar mediante la prueba de escritorio.

Realizar un diagrama de flujo que permita ingresar solamente el número cero y el uno, presentar y finalizar. Comprobar mediante la prueba de escritorio.

Realizar el diagrama de flujo que permita ingresar valores mayores e iguales a 15 y menores e iguales a 18 o mayores e iguales a 30 y menores e iguales a 50 presentar el dato correcto, luego finalice

Capítulo 9

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Obtener conocimiento sobre las técnicas, para desarrollar de una forma práctica, los diagramas de flujo complejos con realimentación
2. Al finalizar la unidad el lector será capaz de tener un criterio claro sobre diagramas de flujo complejos con realimentación
3. El lector estará en la capacidad de desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad

Diagramas de estructuras complejas con realimentación

Establecen el número de iteraciones que se producen en un bucle. Además dentro del cuerpo del bucle se puede estructurar los contadores y acumuladores.

Contadores

Se utiliza como elemento básico y una de sus funciones básicas es controlar a la decisión por intermedio de un contador que es una expresión aritmética que se incrementa o decrementa en un valor constante. Así como también representa el número de iteraciones.

Siempre a la variable del contador hay que inicializarla con un valor numérico constante para que funcione la expresión y por lo menos una decisión que detenga las iteraciones. Los contadores son positivos y negativos

Contador positivo.- Se incrementa en un valor constante

Formato Inicialización : Variable=Valor Numérico

Contador: Variable= Variable +Constante

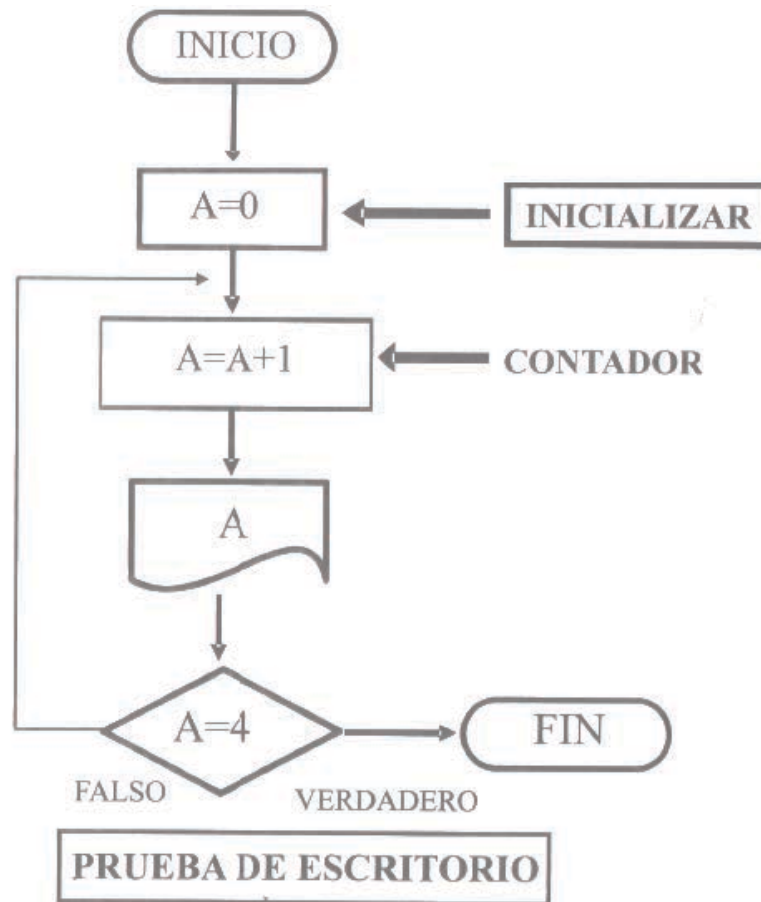
Ejemplo:

Inicialización A=0

Contador A=A+1

Ejercicio #1

Construya un diagrama de flujo que presente el número de iteraciones o ciclos
Necesitamos que realice 4 iteraciones, luego finalizar



A	A=A+1	A	PRESENTA	A=4	VERDADERO FIN	FALSO REGRESA
0	1=0+1	1	1	1=4		SI
	2=1+1	2	2	2=4		SI
	3=2+1	3	3	3=4		SI
	4=3+1	4	4	4=4	SI	

Explicación

En el ejercicio#1 de contadores en la figura 37a se inicia el diagrama con valor de cero que se almacena en la variable A, luego se ingresa automáticamente a la expresión del contador $A=0+1$ el nuevo valor es $A=1$ comparamos este valor con la condición de la decisión $1=4$ es falsa, en este momento el contador comienza a iterar.

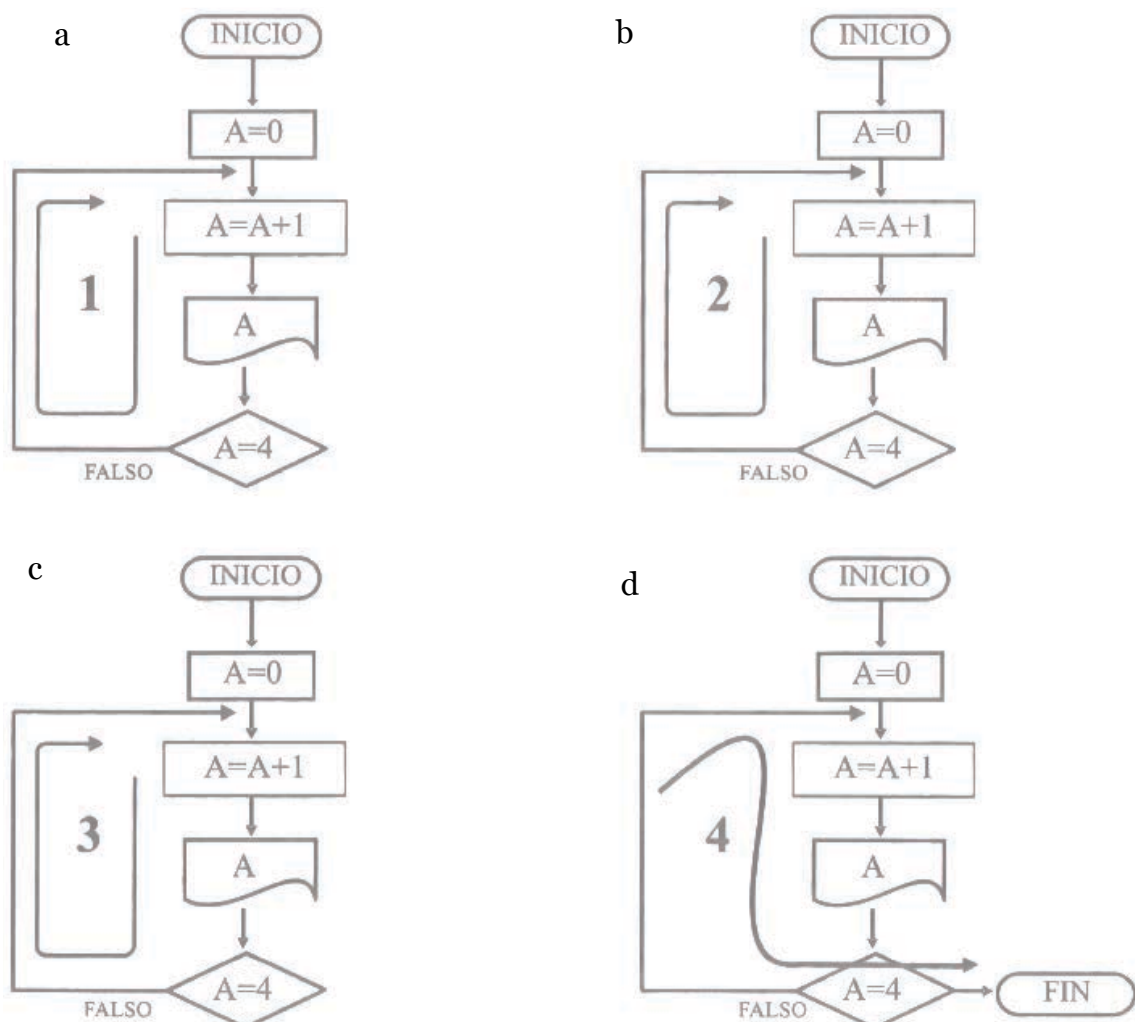
En la figura 37 b el ultimo valor almacenado es $A=1$ ingresa a la expresión del contador $A=1+1$ entonces el nuevo valor de $A=2$ comparamos este valor con

la condición de la decisión $2=4$ es falsa, en este momento el contador iterar su segunda vuelta

En la figura 37 c el ultimo valor almacena es $A=2$ ingresa a la expresión del contador $A=2+1$ entonces el nuevo valor $A=3$ comparamos este valor con la condición de la decisión $3=4$ es falsa, en este momento el contador iterar su tercer vuelta.

En la figura 37 d el ultimo valor almacena es $A=3$ ingresa a la expresión del contador $A=3+1$ entonces el nuevo valor de $A=4$ comparamos este valor con la condición de la decisión $4=4$ es verdadera, en este momento el contador termina de iterar y su salida la realiza por la bifurcación verdadera y finalizar.

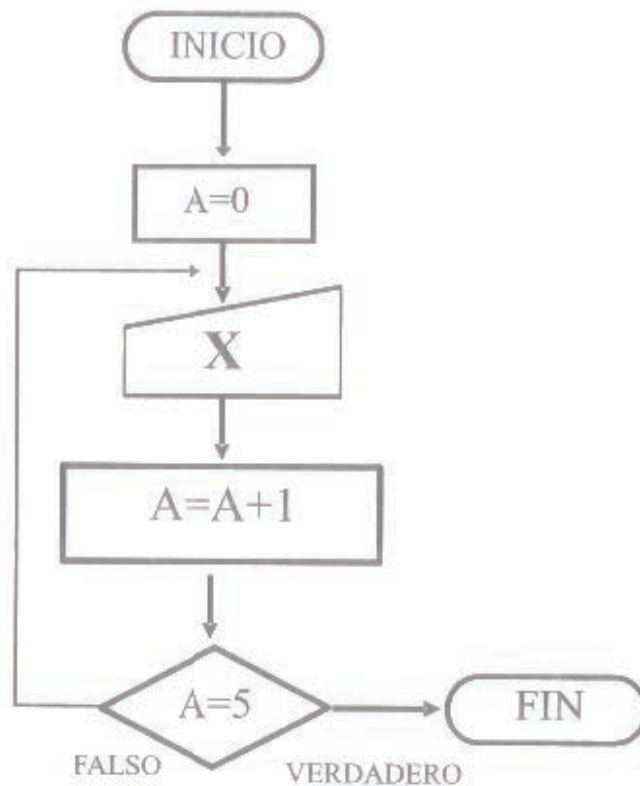
Figura 37: Explicación de contadores ejercicio#1



Ejercicio#2

Construya un diagrama de flujo que realice el ingreso por teclado de 5 salarios luego finalice

La variable X representa los salarios que se ingresan



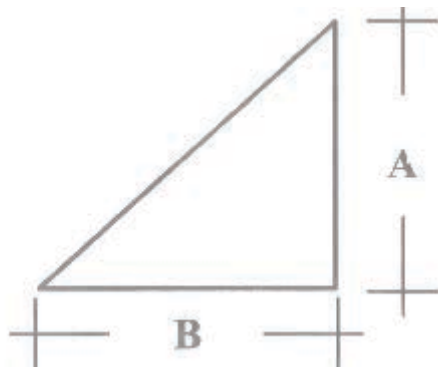
PRUEBA DE ESCRITORIO

A	X	A=A+1	A	A=4	VERDADERO FIN	FALSO REGRESA
0	120000	1=0+1	1	1=4		SI
	232323	2=1+1	2	2=4		SI
	400000	3=2+1	3	3=4		SI
	100000	4=3+1	4	4=4		SI
	100000	5=4+1	5	5=5	SI	

Ejercicto#3

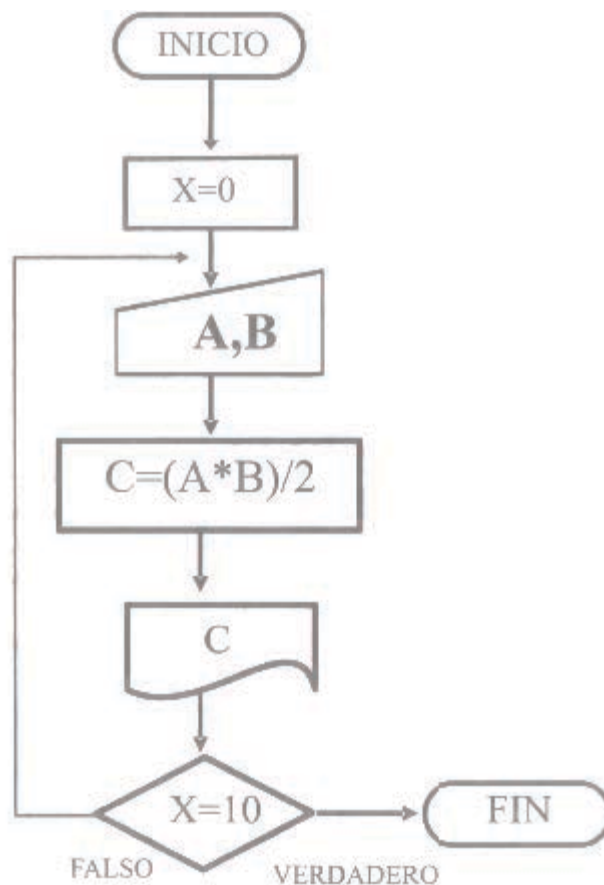
Un hacendado desea saber cuál es el área de sus diez haciendas que tienen la forma de un triángulo rectángulo. Ingrese los datos por periférico y presentar cada uno de los resultados. Realizar el ejercicio con contadores para el ingreso de los datos.

NOTA: Solo se deben ingresar los valores positivos



Área del triángulo rectángulo

$$C=(A*B)/2$$

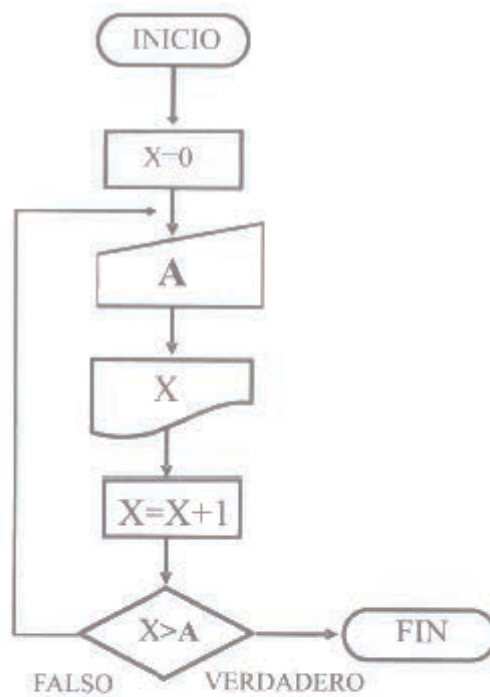


Ejercicio #4

Construya un diagrama de flujo que presente de una forma automática todos los números enteros positivos desde el cero hasta el dato ingresado por teclado. Por ejemplo

Si ingresamos el 5 se debe presentar automáticamente 0, 1, 2, 3, 4 5

Si ingresamos el 9 se debe presentar automáticamente 0, 1, 2, 3, 4 5, 6, 7, 8, 9



PRUEBA DE ESCRITORIO

X	A	PRESENTAR X	X=X+1	X>A	VERDADERO FIN	FALSO REGRESA
0	4					

Contador negativo.- Se decrementa en un valor constante

Formato Inicialización: Variable=Valor numérico

Contador: Variable= Variable - Constante

Ejemplo:

Inicialización A=0

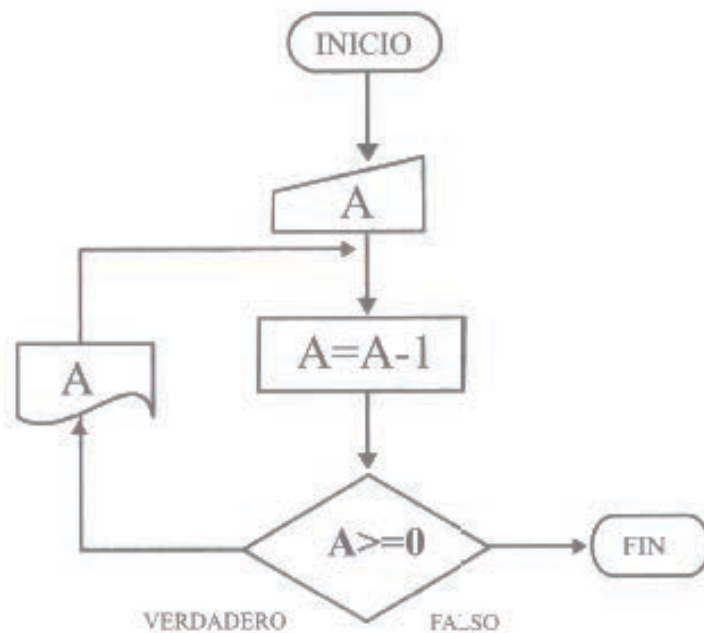
Contador A=A-1

Ejercicio

Construya un diagrama de flujo que realice la cuenta regresiva de un número ingresado por teclado hasta el cero uno en uno presentar todos los datos, luego finalice

Si ingresamos el 5 el resultado es 4, 3, 2, 1,0

Si ingresamos el 7 el resultado es 6, 5, 4, 3, 2, 1,0



Acumuladores

Al igual que el contador se incrementa o decrementa al realizar una iteración, pero el incremento es variable, se utiliza para realizar sumas sucesivas dentro de un bucle. Este incremento o decremento se obtiene mediante el operador de la suma, resta, multiplicación y división. Dependiendo el problema que deseamos resolver se utilizara el operador Aritmético correspondientes. La inicialización de la variable, se realiza mediante una constante o el ingreso de un dato representado por una variable.

Formato:

SUMA =SUMA operador VARIABLE

Importante es la repetición de la variable suma para lograr la acumulación en cada iteración, aunque su formato puede variar

EJEMPLO:

$S=S+A$

$S=S-A$

$S=S*A$

$S=S/A$

Se necesita saber el total de los sueldo pagados en una empresa, en este caso utilizaríamos un acumulador con operador +.

Ejercicio#1

Construya un diagrama de flujo que presente el resultado de la suma total de tres notas. Utilizando contadores y acumuladores:

NOTA: Supongamos que todos los datos ingresados estarán entre 0 y 20.

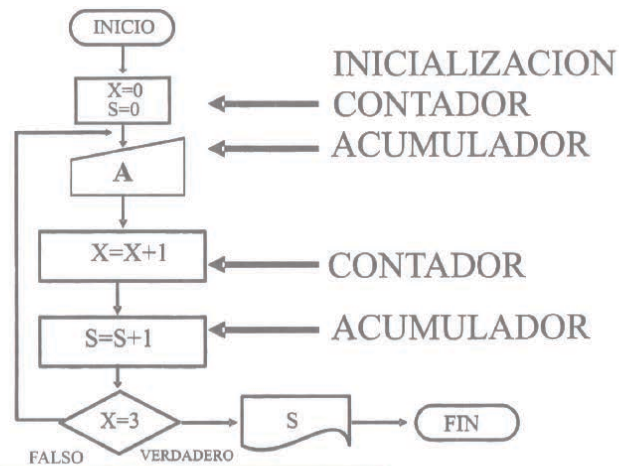
Para el ingreso utilizaremos la variable A

Se necesita un contador para controlar la decisión y que realice 3 iteraciones

Se necesita un acumulador para sumar las tres notas Por ejemplo:

Si ingresamos 20, 3, 12 el resultado es 35

Si ingresamos 20, 19, 18 el resultado es 57



PRUEBA DE ESCRITORIO

X	S	A	X=X+1	S=S+A	X=3	VERDADERO PRESENTA	FALSO REGRESA
0	0	20	1=0+1	20=0+20	1=3		SI
		3	2=1+1	23=20+3	2=3		SI
		12	3=2+1	35=23+12	3=3	35	NO

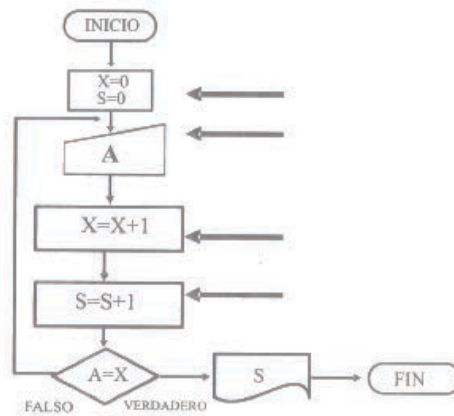
X	S	A	X=X+1	S=S+A	X=3	VERDADERO PRESENTA	FALSO REGRESA
0	0	20	1=0+1	20=0+20	1=3		SI
		19	2=1+1	49=20+19	2=3		SI
		18	3=2+1	57=49+18	3=3	57	NO

Ejercicio #2

Diseñe un diagrama de flujo que nos permita sumar automáticamente de una forma sucesiva desde el cero hasta un número que determine el usuario, presente el resultado y finalice, por ejemplo:

Si ingresamos 3 el resultado es 1+2+3=6

Si ingresamos 6 el resultado es 1+2+3+4+5+6=21



PRUEBA DE ESCRITORIO

X	S	A	X=X+1	S=S+X	A=X	VERDADERO PRESENTA	FALSO REGRESA
0	0	3	1=0+1	10=0+1	3=1		SI
			2=1+1	3=1+2	3=2		SI
			3=2+1	6=3+3	3=3	6	NO
X	S	A	X=X+1	S=S+X	A=X	VERDADERO PRESENTA	FALSO REGRESA
0	0	6	1=0+1	1=0+1	6=1		SI
			2=1+1	3=1+2	6=2		SI
			3=2+1	6=3+3	6=3		SI
			4=3+1	10=6+4	6=4		SI
			5=4+1	15=10+5	6=5		SI
			6=5+1	21=15+6	6=6	21	NO

Ejercicio #3

Diseñe un diagrama de flujo que sume 10 veces un número ingresado presente el resultado y finalice, por ejemplo:

Si ingresamos 2 el resultado es $2+2+2+2+2+2+2+2+2+2=20$

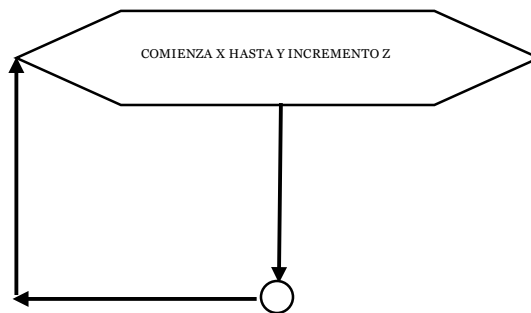
Si ingresamos 10 el resultado es $10+10+10+10+10+10+10+10+10+10=100$

Ciclos FOR

Ejecuta las iteraciones mediante el ingreso de parámetros establecidos por el programador o el usuario. Puede ser utilizado como un contador o acumulador pero no necesita inicialización presenta mayores ventajas .Es un contador pero con mayores ventajas, no necesita inicialización

Formato: COMIENZA X HASTA Y INCREMENTO Z

Figura 38: Estructura Ciclo FOR



La expresión comienza a contar en un número X que se almacena en A hasta finalizar en un número Y incrementándose Z números. La variable donde se almacenaran todos estos valores es A según la estructura de los ciclos FOR.

Las variables X, Y, Z deben ser números enteros, reales, positivos, negativos, ceros o una expresión aritmética o variables.

Por ejemplo: COMIENZA A=25 HASTA 100 INCREMENTO 20

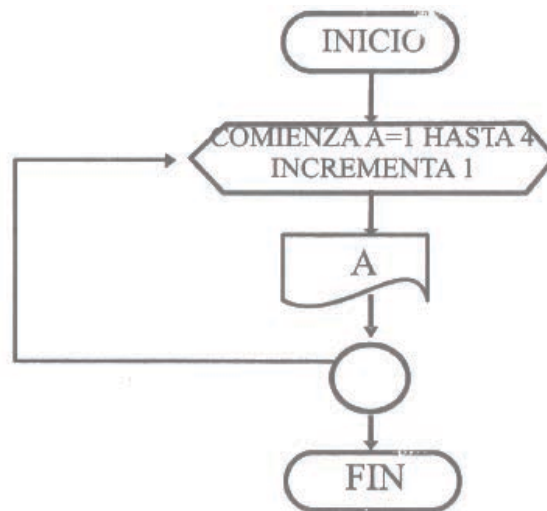
El resultado es 1, 3, 5, 7, 9, 11, 13, 15, 17,19

COMIENZA A=25 HASTA 100 INCREMENTO 20

El resultado es 25, 45, 65, 85

Ejercicio# 1

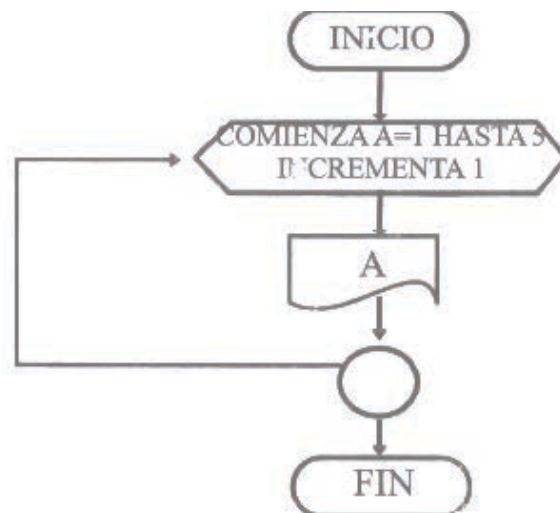
Construya un diagrama de flujo que visualice el número de iteraciones ciclos que se ejecutan. Realice 4 iteraciones, luego finalizar



COMIENZA A=1 HASTA 4 INCREMENTA 1	PRESENTA A	FIN
1	1	
2	2	
3	3	
4	4	SI

Ejercicio# 2

Construya un diagrama de flujo que visualice por teclado de 5 salarios, luego finalice

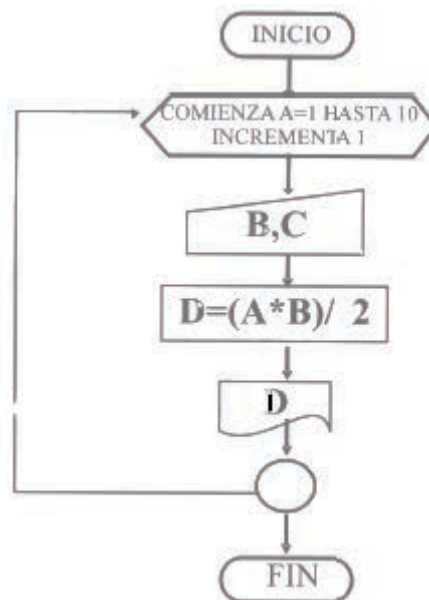


COMIENZA A=1 HASTA 4 INCREMENTA 1	INGRESAR S	FIN
1=0+1	2100000	
2=1+1	5000000	
3=2+1	3959999	
4=3+1	3764676	
5=4+1	45345345	SI

Ejercicio #3

Un hacendado desea saber cuál es el área de cada una de sus diez haciendas que tienen la forma de un triángulo rectángulo. Ingrese los datos por periférico y presentar cada uno de los resultados.

Realizar el ejercicio con ciclos FOR para el ingreso NOTA: No se puede ingresar valores negativos

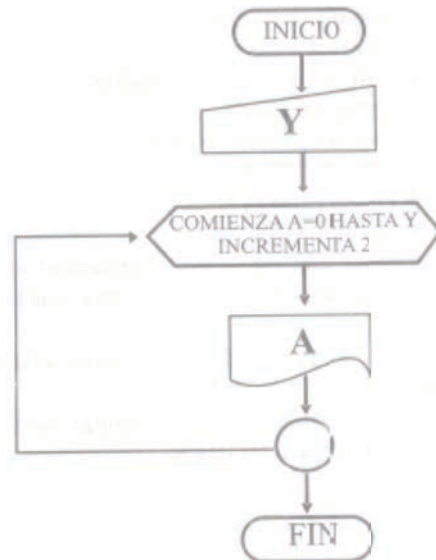


Ejercicio #4

Realice un diagrama de flujo para generar todos números pares desde el cero hasta un número ingresados por teclado y presentar cada uno de los valores, luego finalizar, por ejemplo:

Ingresamos 10 el resultado es 0, 2, 4, 6, 8, 10

Ingresamos 15 el resultado es 0, 2, 4,6, 8, 10, 12, 14



Y	COMIENZA A=1 HASTA 4 INCREMENTA 1	PRESENTA A	FIN
10	2=0+2	2	
	4=2+2	4	
	6=4+2	6	
	8=6+2	8	
	10=8+2	10	SI

Ejercicios resueltos

1.-Diseñe un diagrama de flujo para resolver el problema del pago la matrícula de los estudiantes de los colegios de Guayaquil. Se establecen tres valores según el nivel socioeconómico de la familia.

Primer nivel 300000
 Segundo nivel 200000
 Tercer nivel 150000

A este valor tenemos que sumarle 15000 .La variable que representa el nivel socioeconómico es A

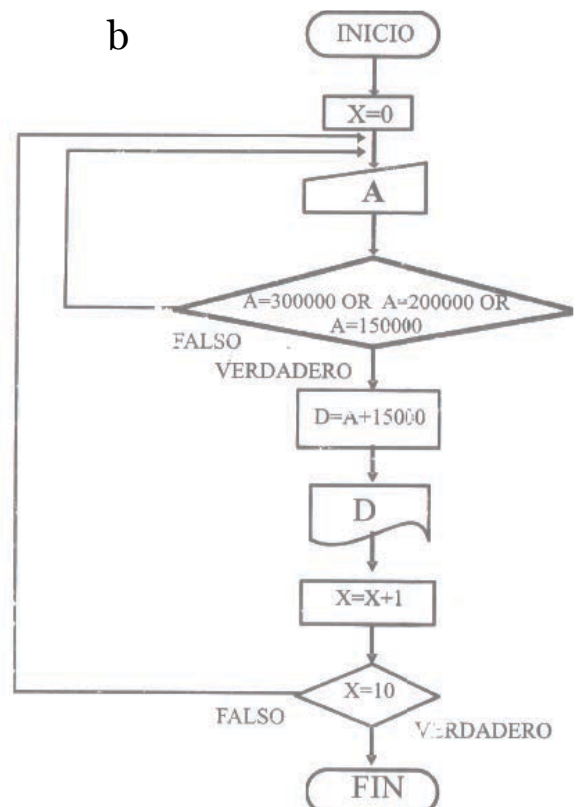
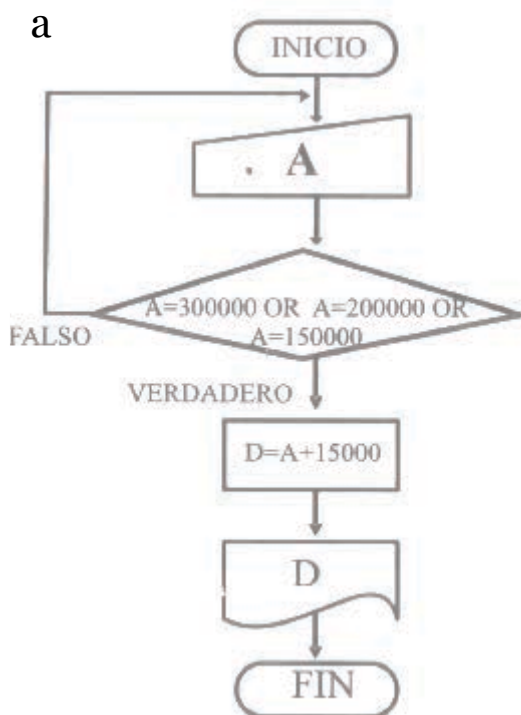
$$D=A+15000$$

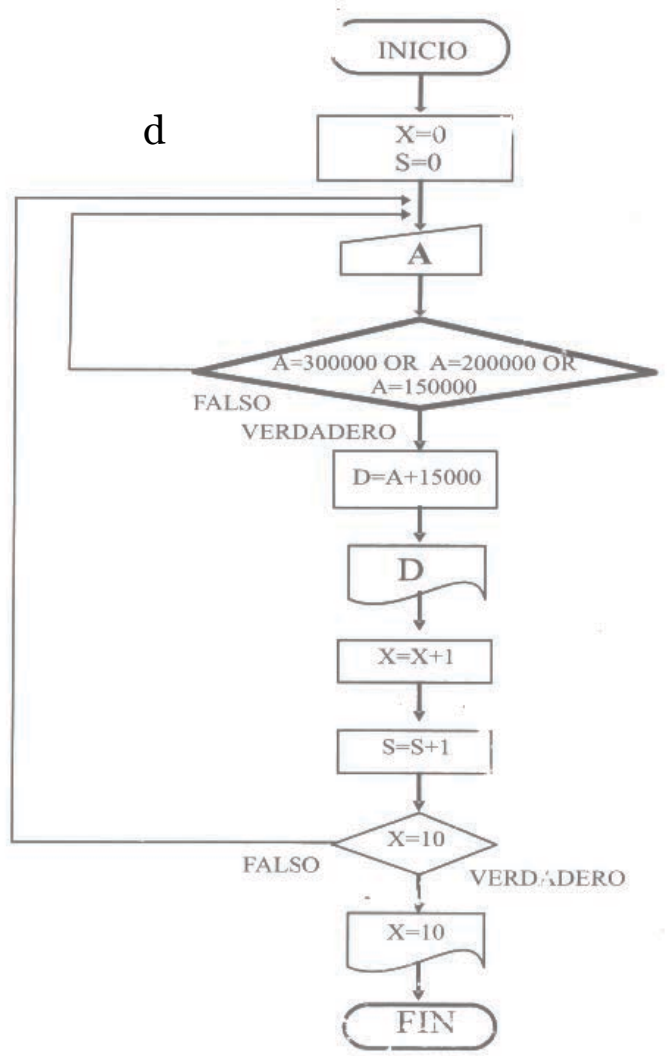
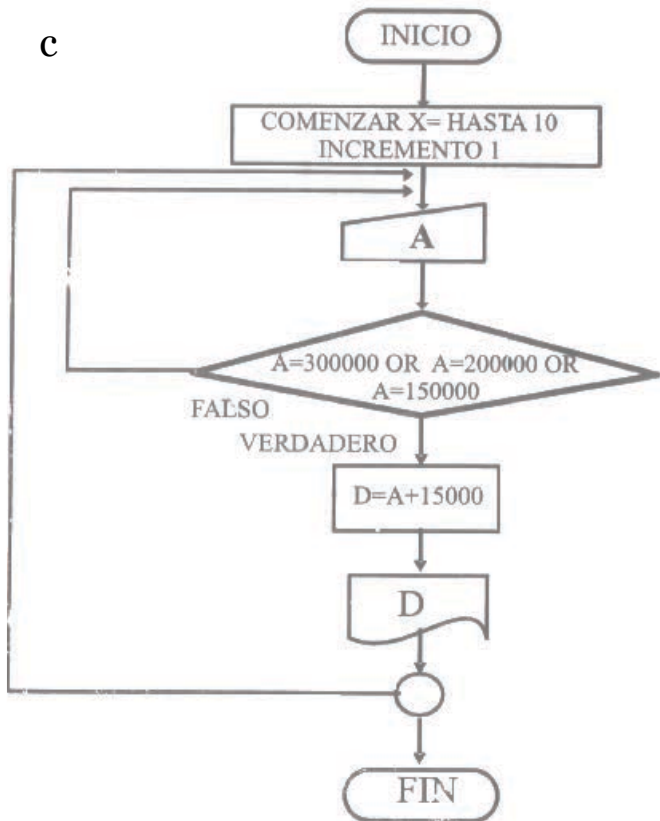
- a) Realizar el diagrama para ingresar una matrícula y presentar el resultado

b) Realizar el diagrama para ingresar 10 matrículas y presentar cada uno de los resultados, utilice contadores

c) Realizar el diagrama para ingresar 10 matrículas y presentar cada uno de los resultados, utilice Ciclos FOR

d) Realizar el diagrama para ingresar 10 matrículas y presentar cada uno de los resultados y su suma total, utilice acumuladores





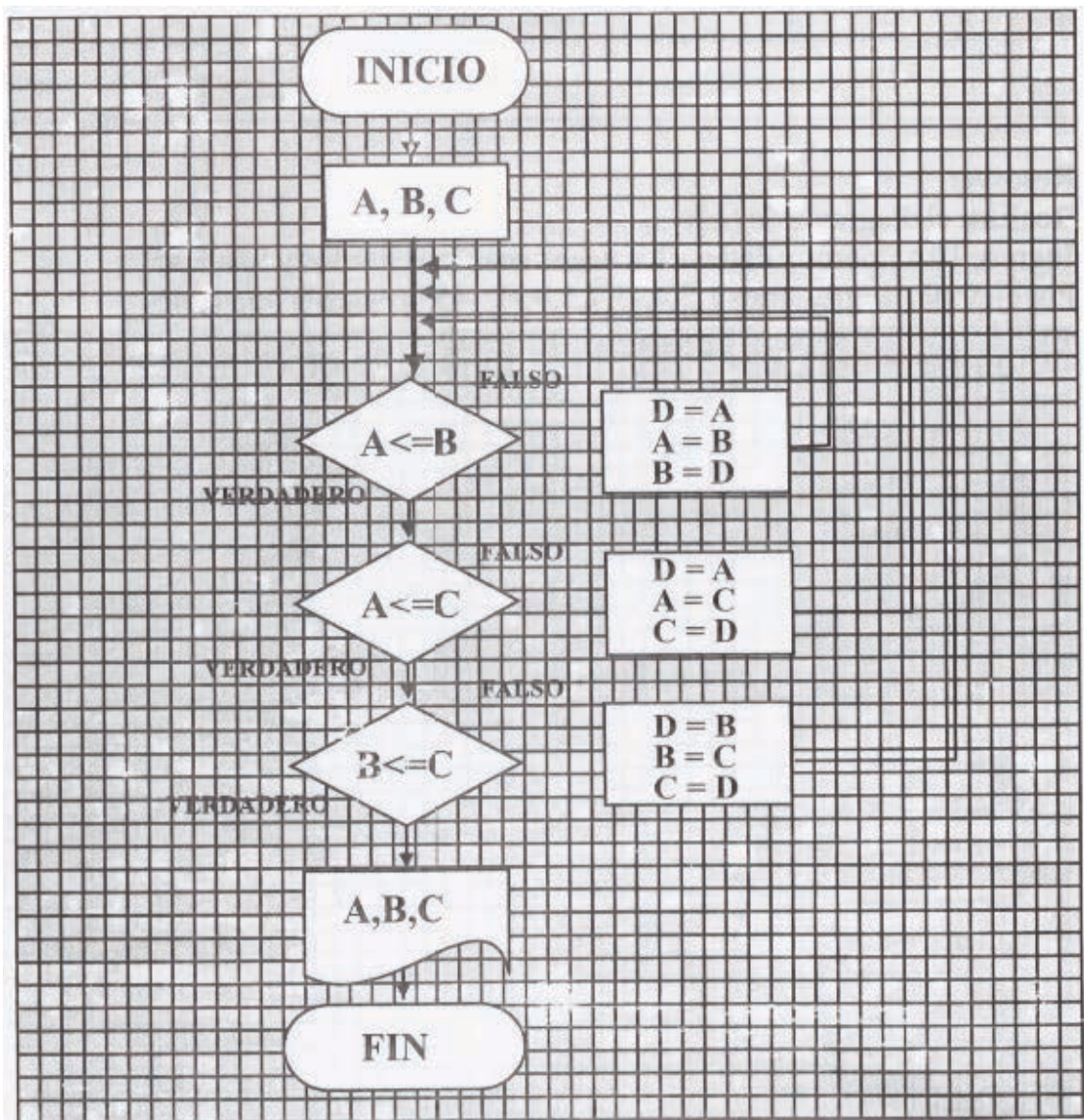
Actividad # 9

Resolver

Utilizando la prueba de escrito verifique si el diagrama de flujo está diseñado correctamente. Se ingresan 3 números por teclado, nuestro ejercicio consiste en presentar los datos ordenados de menor a mayor, luego finalizar por ejemplo:

Ingresamos 3,5, 4 presenta 3, 4, 5

Ingresamos 4,1, 0 presenta 0, 1,4



Conteste

¿Cuáles la diferencia entre contador y acumulador?

¿Cuáles la diferencia entre contador y ciclo FOR?

Resolver

Realizar un diagrama de flujo para ingresar 10 números y detecte los negativos, presentarlos y finalizar Por ejemplo: ingresamos 1, 3, 4,-5, 7, 5, 4,-1,-45,3presenta --5,-1,-45

Sumar los 12 primeros números enteros positivos, presentar el resultado y finalizar, por ejemplo $1+2+3+4+5+6+7+8+9+10+11+12=78$

Realizar un diagrama para ingresar un número y luego sumar los 4 números enteros positivos que le siguen, presentar el resultado y finalizar, por ejemplo:

Ingresamos el 5 entonces $6+7+8+9$ presentamos 30

Ingresamos el 20 entonces $21+22+23+24$ presentamos 90

Realizar un diagrama de flujo para presentar los siguientes resultados.
(Utilice ciclos FOR) 4, 8,16 ,32, 64, 128

Realizar un diagrama de flujo para Presentar los siguientes resultados.
(Utilice contadores) 4, 8, 16, 32, 64, 128

Realizar un diagrama de flujo para Presentar los siguientes resultados.
(Utilice ciclos FOR) 5, 3, 1,-1,-3,-5

Realizar un diagrama de flujo para presentar los siguientes resultados.
(Utilice contadores) 5,3, 1,-1,-3,-5

Realizar un diagrama de flujo para realizar la tabla de multiplicar del 1 hasta el 12 de cualquier número ingresado por teclado, presentar cada uno de los resultados, luego finalizar

Realizar un diagrama de flujo para presentar los números de 5 en 5 iniciando desde el 1 hasta un número ingresado, luego finalizar, por ejemplo:
Ingresamos el 20 presentamos 1,5, 10, 15,20.
Ingresamos el 32 presentamos 1, 5, 10, 15, 20, 25, 30

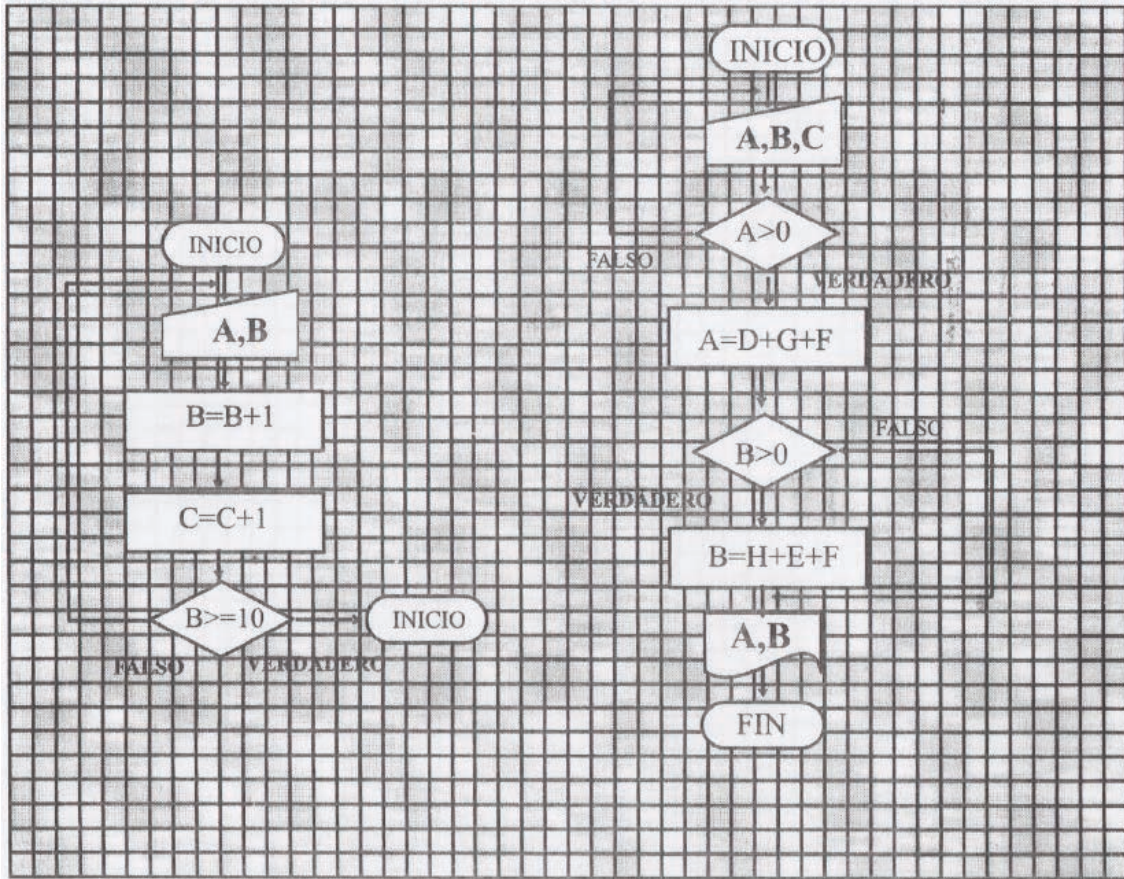
Realizar un diagrama de flujo para presentar el promedio de un número indeterminados de datos ingresados, luego finalice

En un banco para ganar clientes que inviertan realiza la siguiente propuesta. Si invierten su dinero en una póliza de 15 días su dinero se triplicará 100 veces caso contrario 10 veces presente el resultado y finalice. (Utilizar contadores y acumulares)

Realizar un diagrama de flujo para realizar la tabla de sumar de un número ingresado y calcule la suma desde el 1 hasta el 12, luego presentar cada uno de los resultados, luego finalizar

- a) Sin contadores, acumuladores o ciclos FOR
- b) Utilizando contadores
- c) Se quiere sumar todos los resultados de la tabla presentar la respuesta total (utilizar contadores y acumuladores)
- d) Utilizando ciclos FOR y acumuladores

Corregir el siguiente diagrama. Explique su razonamiento



Actividades (opcionales)

Realizar el diagrama de flujo

1) Ingresar un número N de tres dígitos y presentarlo en forma invertida Por
Ejemplo: Ingreso 456 presento 654

Ingreso 123 presento 321

2) Presentar el cuarto dígito de un número N ingresado Ejemplo:

Ingresamos 1263756 presentamos 3

Ingresamos 22376 presentamos 7

Formula $D = \text{MOD} (\text{INT} (N_i / 10^{(P-1)}), 10)$

D= dígito encontrado posición del dígito que deseamos encontrar

INT= función valor entero

N= número ingresado

3) Presentar el séptimo dígito de un número N

4) Encontrar cuantos dígitos tiene un número N ingresado

Formula $K = 1 + \text{INT}(\text{LOG}_{10} (N))$

N= números ingresado

K =números de dígitos

5)Descomponer una cantidad de dólares ingresada por teclado en su equivalente en monedas de 500, 100, 50, 20, 10, 5, para utilizarlas en la compra de bebidas gaseosas en una máquina surtidora, por ejemplo:

Ingresamos 25 presenta 1 moneda de 20

1 moneda de 5

Ingresamos 1025 2 monedas de 500

1 moneda de 20

1 moneda de 5

Capítulo 10

Objetivos

Con el estudio de este capítulo se desea que el lector alcance los siguientes objetivos:

1. Conocer otro método que nos ayudara a resolver de una forma sencilla los diagramas de flujos
2. Realizar el pseudocódigo para programar en un lenguaje de programación.
3. Al finalizar la unidad el lector será capaz de tener un criterio claro sobre arreglos.
4. El lector estará en la capacidad de desarrollar una actitud crítica y autocrítica sobre los temas tratados en la unidad.
5. Decidir el método más adecuado para la resolución de los problemas.

Estructura de datos : Arreglos

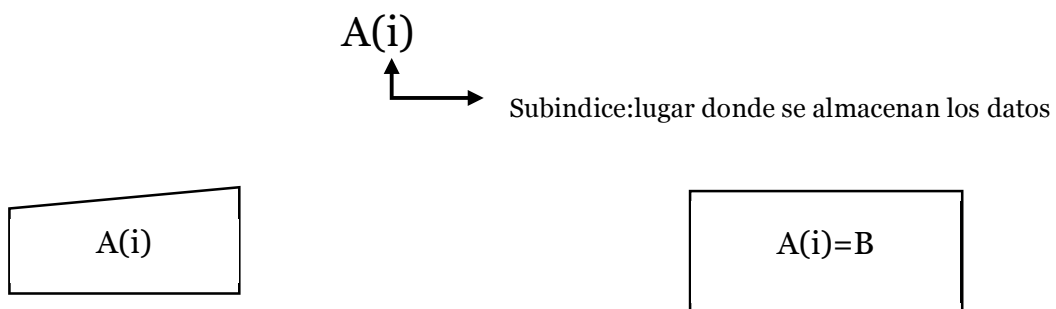
En los capítulos anteriores se utilizaron datos simples para diseñar los diagramas de flujos. Su desventaja en el momento de almacenar y relacionar los datos es notable, se utilizan muchas variables, por tal motivo, para resolver estos problemas de una forma sencilla se crearon los arreglos (arrays).

Los arreglos son un conjunto de valores del mismo tipo que se almacenan de forma secuencial en la memoria central del computador mediante la utilización de subíndices que indican la posición en que se encuentra ubicado el valor almacenado en la memoria. Podemos clasificar los arreglos en: arreglos con un subíndice, con dos subíndices y varios subíndices

Arreglos con un subíndice (vectores)

Son valores finitos almacenados con el mismo tipo de dato, constituidos en una sola dimensión de tipo vector columna o vector fila. El subíndice es generado por un contador, ciclo FOR, o constante.

Formato de ingreso de los arreglos con un subíndice



Por ejemplo: Ingrese las edades las edades de los jugadores de un equipo de indoor

Dimensión tipo vector fila

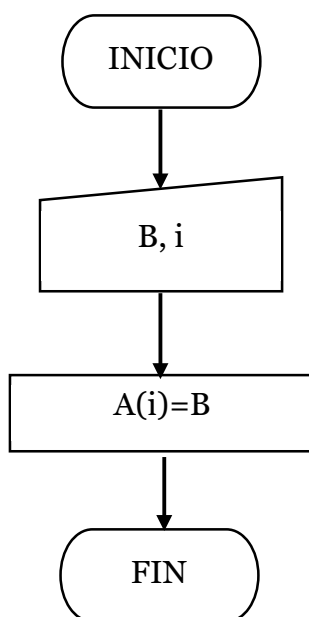
19	27	34	38	23	19	20
----	----	----	----	----	----	----

Dimensión tipo vector columna

19
27
34
38
23
19
20

Ejercicio#1

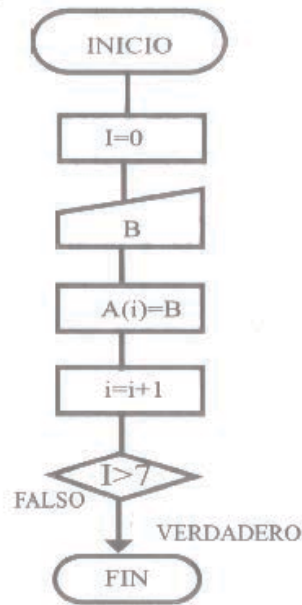
Realizar un diagrama de flujo para almacenar valores



i	B	A(i)=B
1	425	A(1)=425
2	888	A(2)=888
3	5000	A(3)=5000
4	100	A(4)=100

Ejercicio#2

Realizar el diagrama de flujo que permita almacenar en un vector las edades de los siete jugadores de un equipo de indoor



A(0)	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)
19	27	34	38	23	19	20

i	B	A(i)=19	1=0+1	i > 7	VERDADERO FIN	FALSO REGRESA
0	19	A(0)=19	1=0+1	1 > 6		SI
	27	A(1)=27	2=1+1	2 > 6		SI
	34	A(2)=34	3=2+1	3 > 6		SI
	38	A(3)=38	4=3+1	4 > 6		SI
	23	A(4)=23	5=4+1	5 > 6		SI
	19	A(5)=19	6=5+1	6 > 6		SI
	20	A(6)=20	7=6+1	7 > 6	SI	

Operaciones con arreglos

Se utiliza los operadores aritméticos tanto externamente como internamente Se combinarse para intervenir en una expresión

Por ejemplo:

$B=A(1)+2$ EXTERNO

$C=A(1+2)$ INTERNO

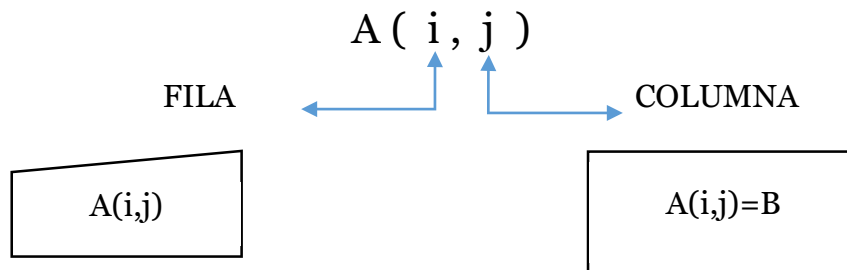
A(0)	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)
19	27	34	38	23	19	20

PRESENTAR	A(3)	<input type="text" value="38"/>
PROCESO	B=A(3)+2 B=38+2	<input type="text" value="40"/>
PROCESO	C=A(3)+A(6) C=38+20	<input type="text" value="58"/>
PRESENTAR	A(3+3) A(6)	<input type="text" value="20"/>
PRESENTAR	A(4+3)	<input type="text" value="NO VALIDO"/>
PROCESO	D=A(2+1)*4+A(5)-A(4)+A(0) 38*4+19-23+19	<input type="text" value="167"/>

Arreglos con dos subíndices (matrices)

Son valores finitos almacenados del mismo tipo de datos, en dos dimensiones, es decir trabajan conjunto vector columna y vector fila.

Formato de ingreso de los arreglos con dos subíndices



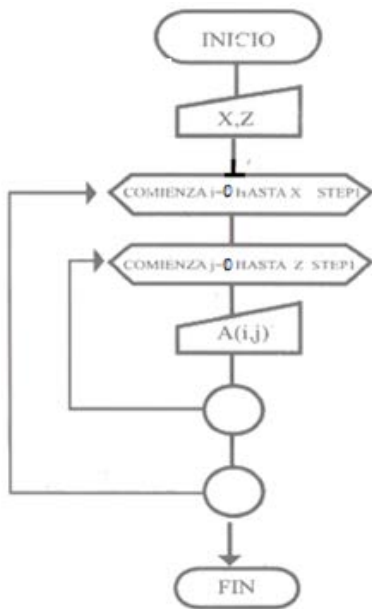
Matriz X filas , Z columnas

Z columnas

X filas

primer dato	Segundo dato	Tercer dato
Cuarto dato	Quinto dato	Sexto dato
Séptimo dato	Octavo dato	Último dato

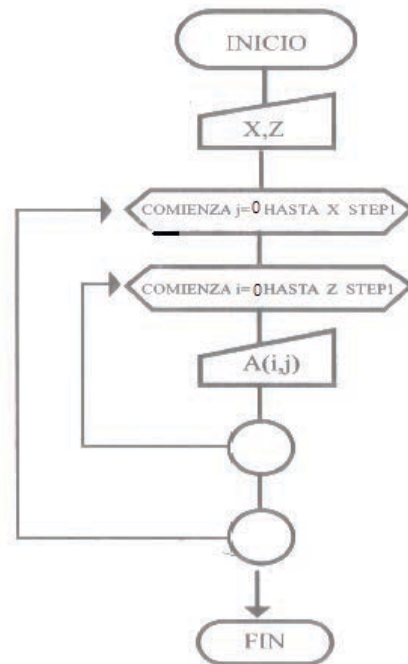
Estructura para generar X filas , Z columna primer método: almacenamiento por fila



Arreglo 3 x 3

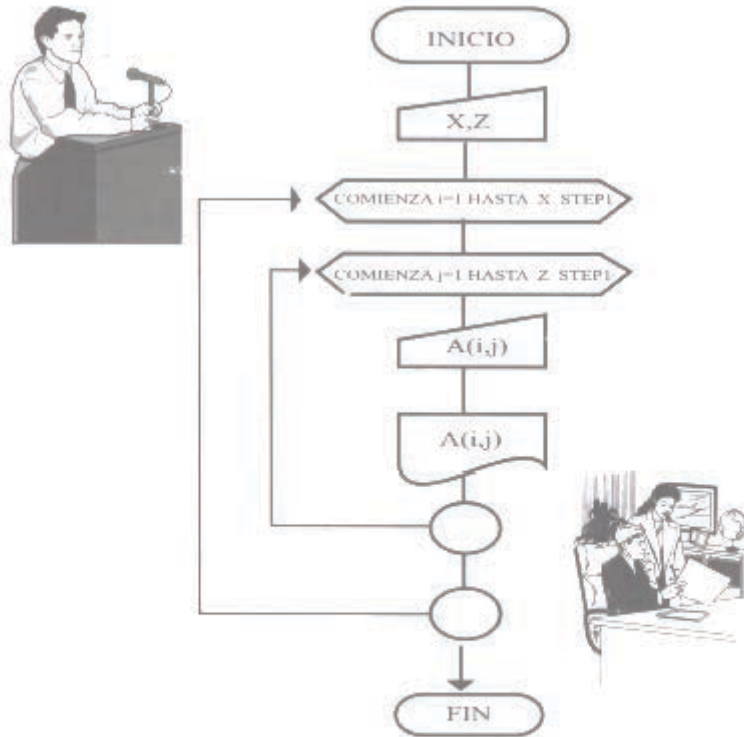
A(i,j)	A(0,1)	A(0,2)
A(1,0)	A(1,1)	A(1,2)
A(2,0)	A(2,1)	A(2,2)
A(3,0)	A(3,1)	A(X,Z)

Estructura para generar X, filas Z columna segundo método: almacenamiento por columna



Ejercicio#1

En una matriz de 2×3 (dos filas y tres columnas) ingrese los nombres de presidentes del Ecuador, presentar, luego finalice. Realice el ejercicio utilizando arreglos con el método de almacenamiento columna

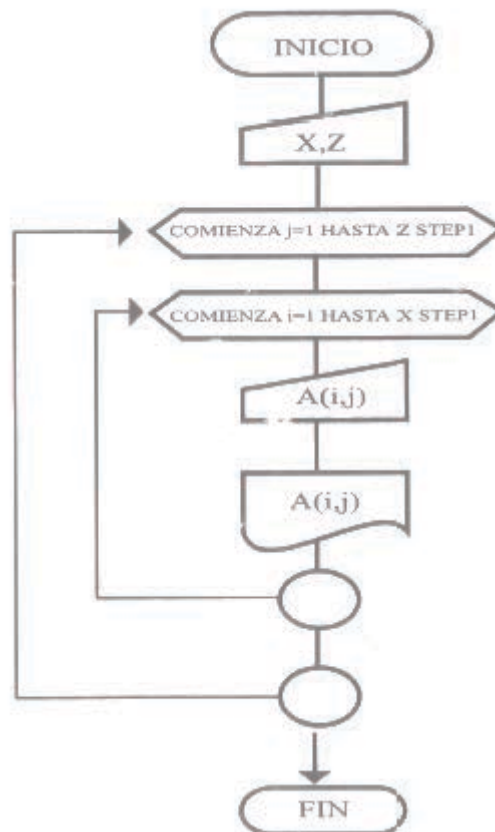


ROCAFUERTE	ROLDOS	LEÓN
VELASCOIBARRA	BORJA	SIXTO

X	Z	COMENZAR i=1 HASTA X	COMENZAR j=1 HASTA Z	INGRESAR A(i,j)	PRESENTAR A(i,j)	FIN
2	3	1	1	A(1,1)=ROCAFUERTE	ROCAFUERTE	NO
		1	2	A(1,2)=ROLDOS	ROLDOS	NO
		1	3	A(1,3)=LEON	LEON	NO
		2	1	A(2,1)=VELASCO IBARRA	VELASCO IBARRA	NO
		2	2	A(2,2)=BORJA	BORJA	NO
		2	3	A(2,3)=SIXTO	SIXTO	SI

Ejercicio# 2

En una matriz de 2*3 (dos filas y tres columnas) ingrese los nombres de presidentes del Ecuador, presentar, luego finalice .Realice el ejercicio utilizando arreglos con el método de almacenamiento por filas



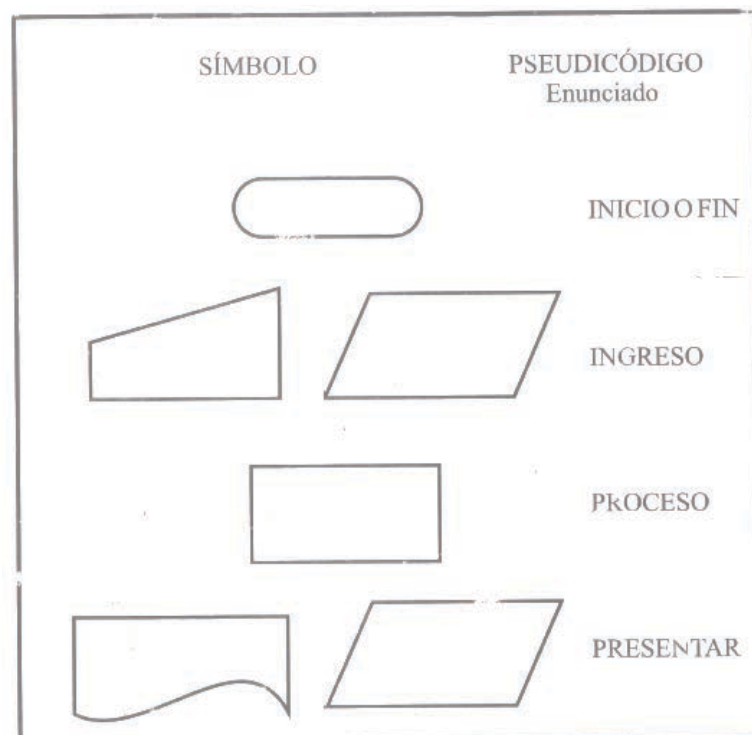
x	z	COMENZAR j=1 HASTA Z	COMENZAR i=1 HASTA X	INGRESAR A(i,j)	PRESENTAR A(i,j)	FIN
2	3	1	1	A(1,1)=ROCAFUERTE	ROCAFUERTE	NO
		1	2	A(2,1)=ROLDOS	ROLDOS	NO
		2	1	A(1,2)=LEON	LEON	NO
		2	2	A(2,2)=VELASCO IBARRA	VELASCO IBARRA	NO
		3	1	A(1,3)=BORJA	BORJA	NO
		3	2	A(2,3)=SIXTO	SIXTO	SI

Pseudocódigo

Después de realizar el diagrama de flujo el paso siguiente para resolver un problema por computadoras es escribir el pseudocódigo. Es una forma escrita de y representar un diagrama de flujo, su estructura sirve como base para la codificación de los lenguajes de programación, debido a que no se preocupa de las reglas de estos. En este sentido Joyane y Zahonero (2005) expresan que el pseudocódigo “es una herramienta de programación en la que las instrucciones se escriben en palabras similares al inglés o español, que facilitan tanto la escritura como la lectura de programas. En esencia, el pseudocódigo se puede definir como un lenguaje de algoritmos”(p.33). Además El pseudocódigo no puede ser ejecutado en un computador.

Adoptaremos las mismas instrucciones que se mencionaron en de algoritmos que son una forma básica de los pseudocódigos. El símbolo de ingreso/salida por periférico desempeñando cualquiera de las dos funciones

Figura 39: Instrucciones básicas del pseudocódigo



Enunciados de decisiones

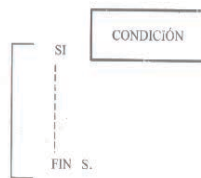
Son las estructuras de las decisiones simples de los diagramas complejos sin realimentación.

Figura 40: Estructura SI (IF-ENDIF)

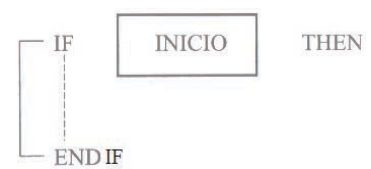
Símbolo



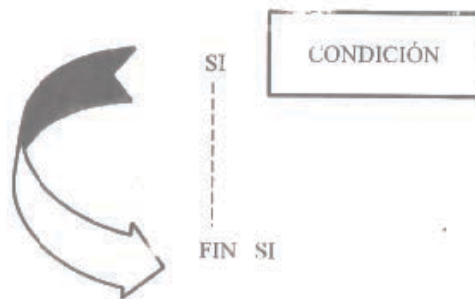
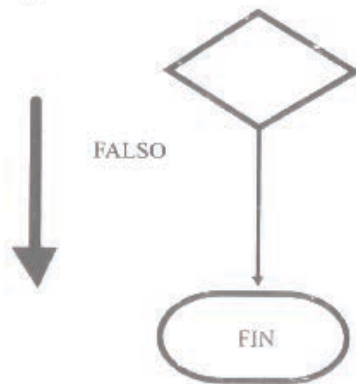
Pseudocódigo



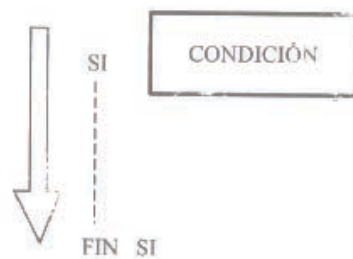
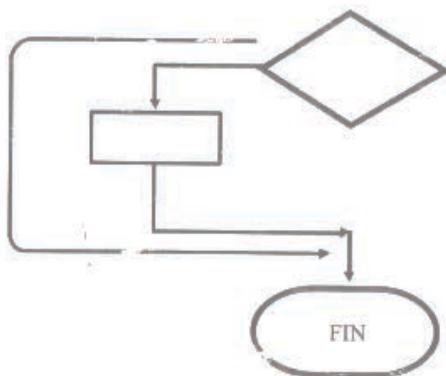
Pseudocódigo en ingles



Si no cumple con la condición saltara al FIN SI

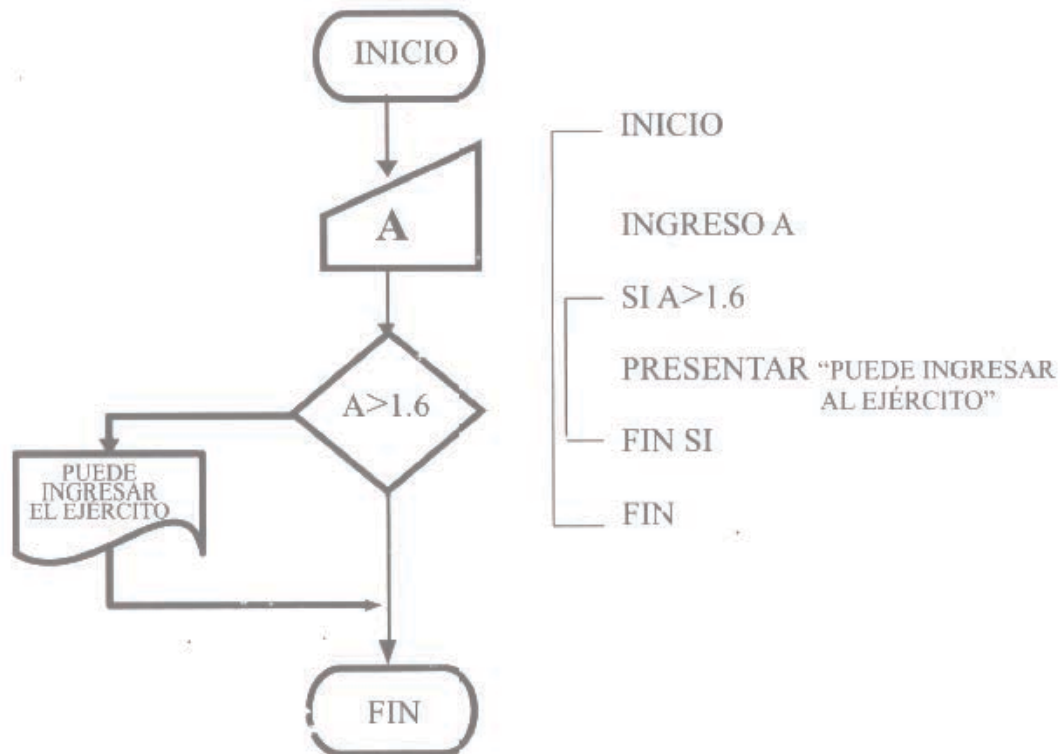


Si cumple con la condición seguirá su flujo por el camino verdadero



Ejemplos#1

En el ejército se necesita detectar automáticamente a las persona que pueden ingresar al servicio militar si tienen una estatura mayor a 1.60 metros. Realizar el pseudicódigo para ingresar las estaturas, si es mayor a 1.60 se presentar PUEDE INGRESAR AL EJERCITO caso contrario finaliza.



Estructura selectiva SI - SINO (IF-ELSE-ENDIF)

Son las estructuras de las decisiones dobles de los diagramas complejos sin realimentación. Si la condición es verdadera ejecutara las instrucciones hasta el SINO y después saltará al FINSI. Además Deitol y Deitol (2008) dicen “la instrucción if realiza una acción indicada solamente cuando la condición es verdadera; de no ser así, se evita dicha acción. La instrucción if... else de selección doble permite al programador especificar una acción a realizar cuando la condición es verdadera y otra distinta cuando la condición es falsa” (p.116).

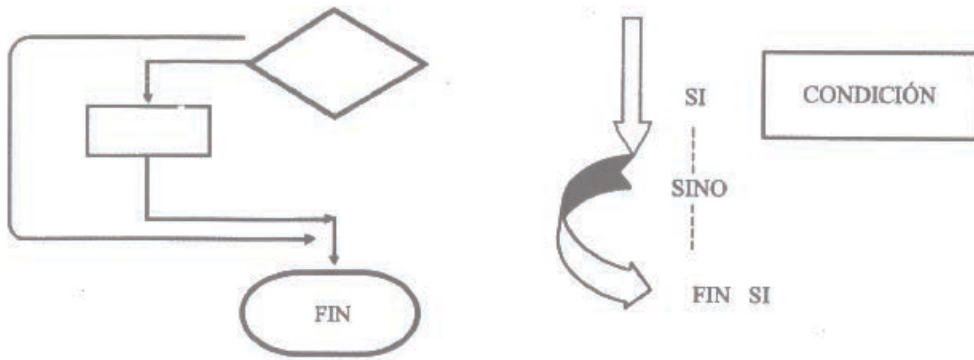
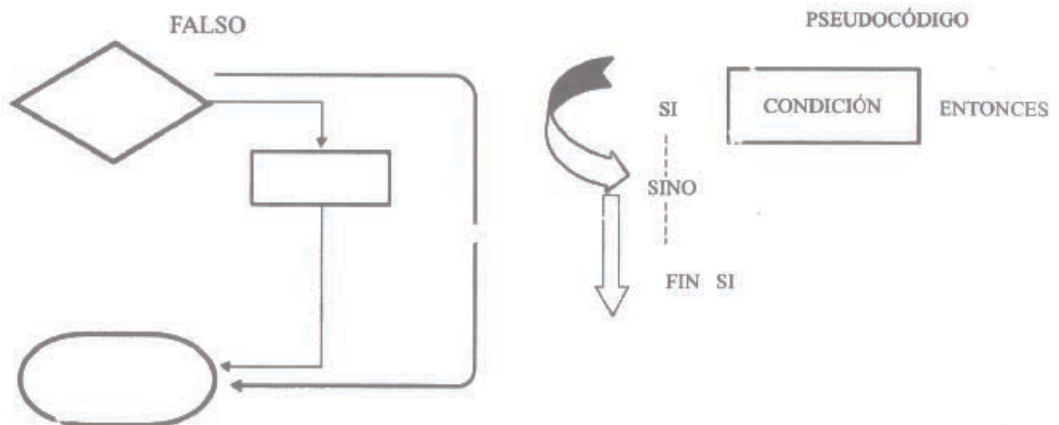


Figura 41: Estructura SI (IF-ELSE-ENDIF)

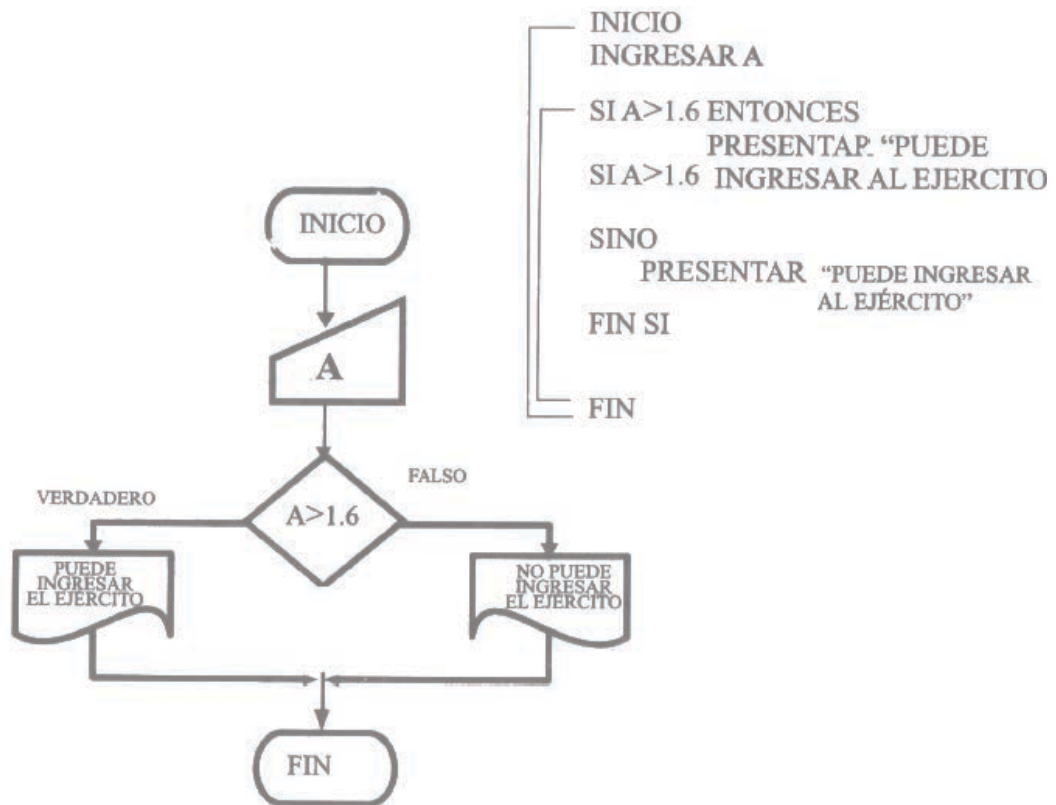


Si la condición no se cumple seguirá el camino falso, es decir, saltará hasta el SINO y desde aquí comenzará a ejecutar las instrucciones.



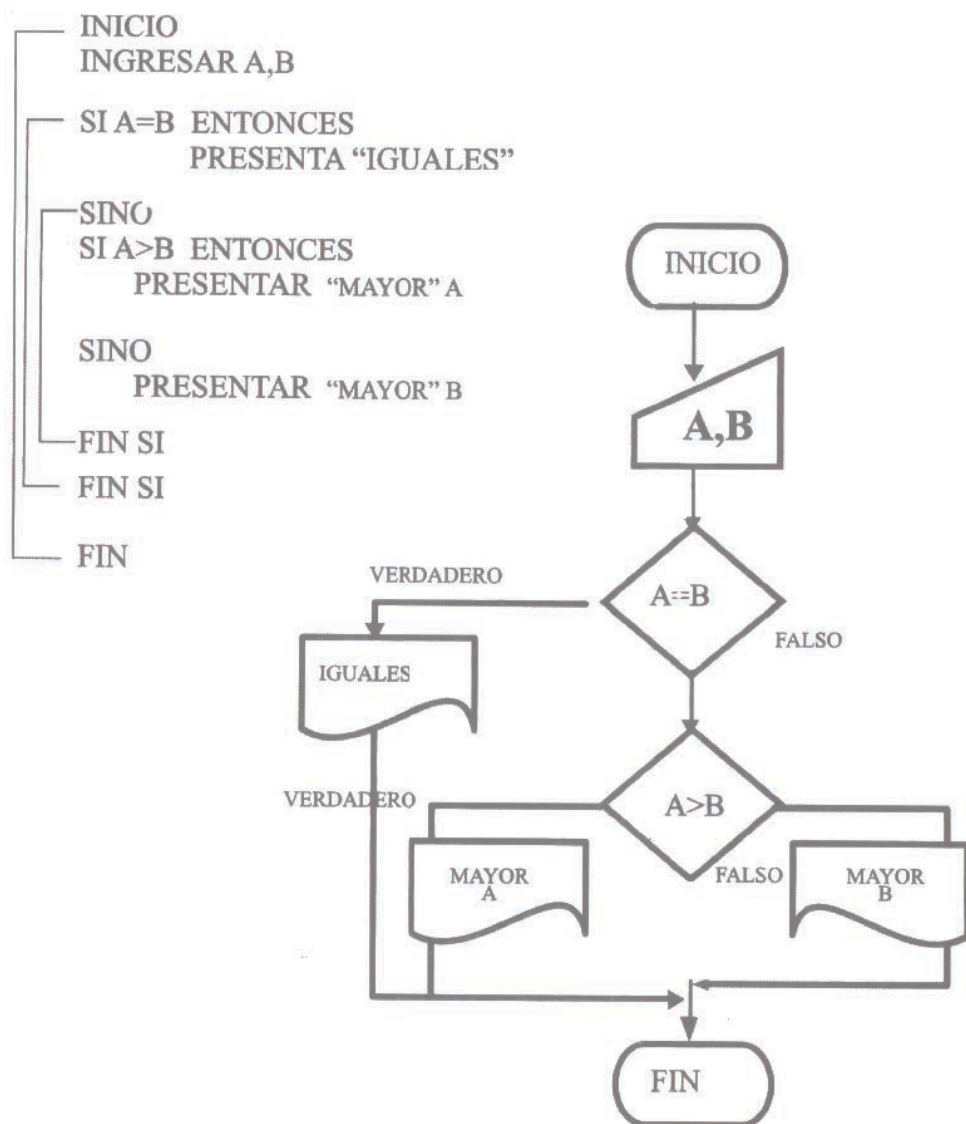
Ejercicio#1

En el ejército se necesita detectar automáticamente a las personas que pueden ingresar al servicio militar si tienen una estatura mayor 1.60 metros. Se deberá ingresar por teclado cada uno de los datos, si es mayor a 1.60 se presentara por impresora PUEDE INGRESAR AL EJERCITO caso contrario presentar NO PUEDE INGRESAR AL EJERCITO, luego finalice.



Ejercicio#2

Realizar un diagrama de flujo para detectar automáticamente cual de dos números ingresados por teclado es el mayor presentar por impresora el mensaje MAYOR y a continuación la variable, si los números son iguales presentar IGUALES y finalice



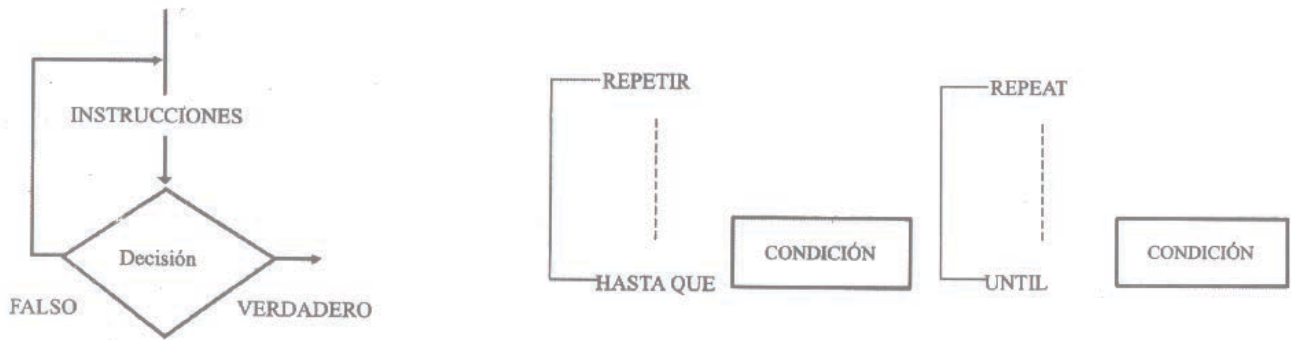
Enunciados de iteración

Son las estructuras de las decisiones de los diagramas complejos con realimentación

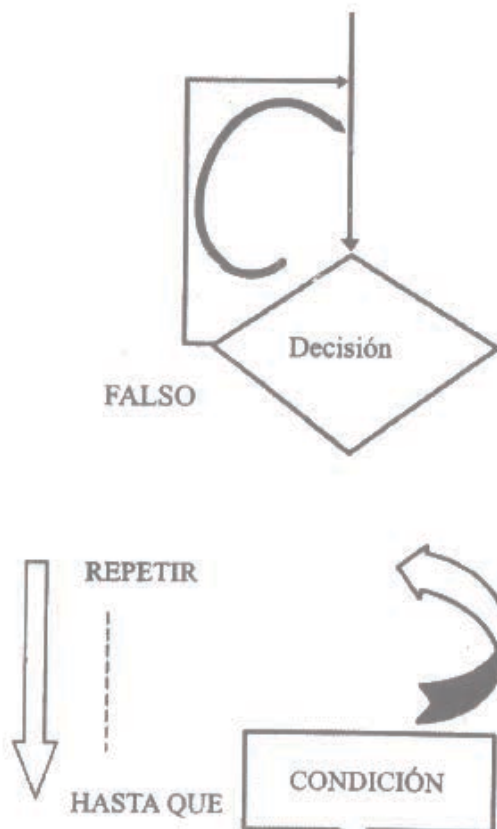
Estructura repetitiva REPETIR - HASTA (REPEAT-UNTIL)

Cuando las iteraciones se realizan por la bifurcación falsa

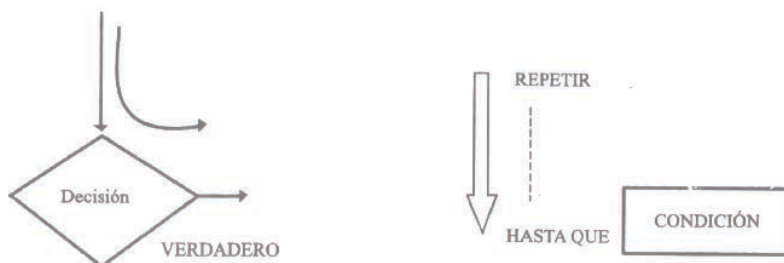
Figura 42: Estructura SI (IF-ELSE-ENDIF)



Si la condición de la estructura es falsa el computador regresara a leer las instrucciones desde REPETIR hasta que se cumpla.

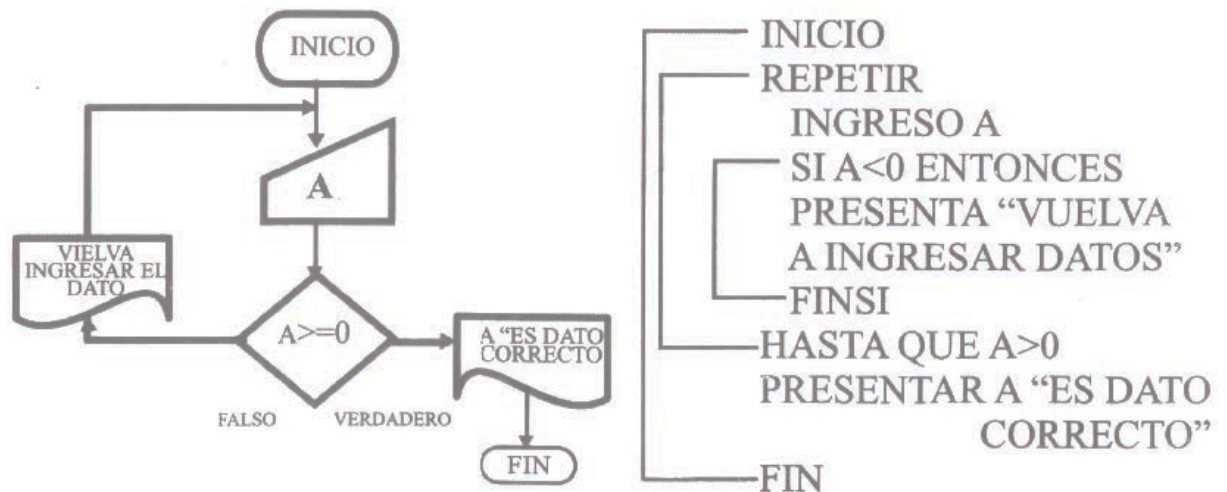


Si la condición es verdadera el computador continuara leyendo las instrucciones después del HASTA QUE



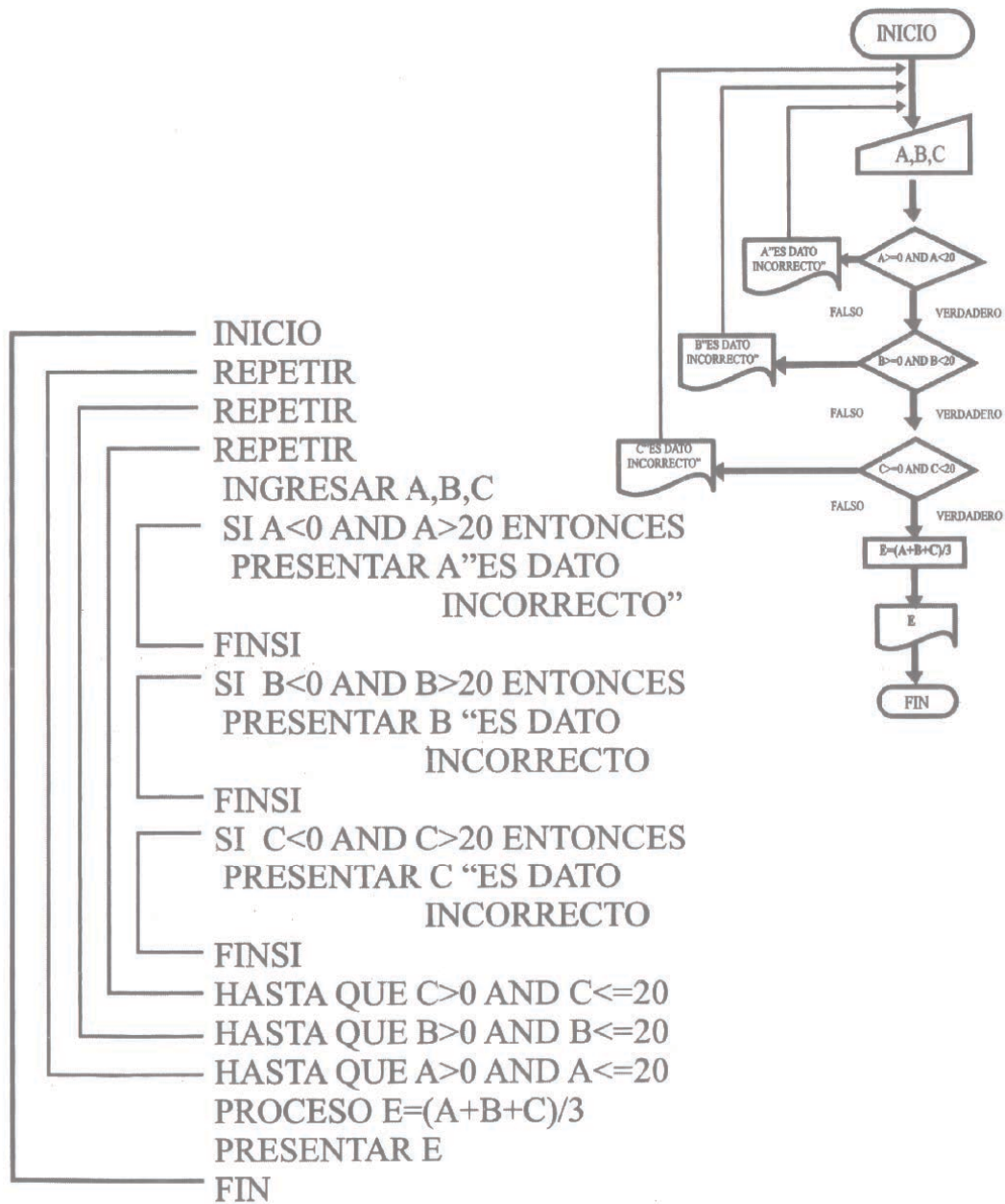
Ejercicio# 1

Realizar un diagrama de flujo para ingresar por teclado solo números positivos y presentar un mensaje de DATOS CORRECTOS caso contrario presentar VUELVA A INGRESAR EL DATO, luego finalice.



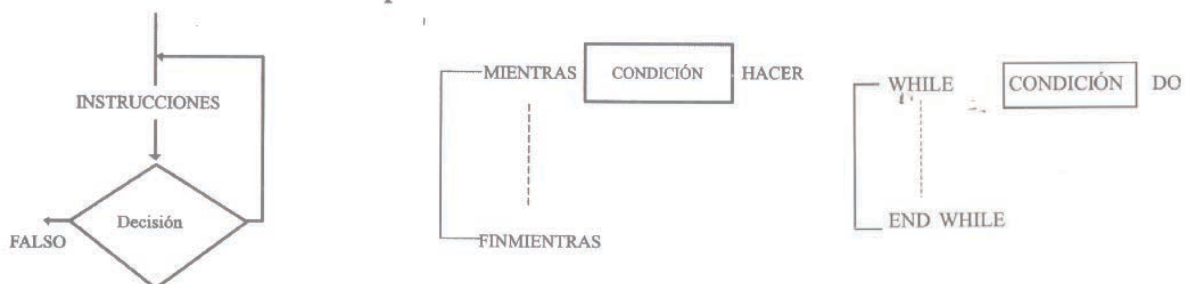
Ejercicio#2

Construir un diagrama de flujo para sacar el promedio de tres notas conociendo que el máximo de la calificación es 20 y la mínima 0, ingrese los datos por teclado, si cumple con la condición sacar el promedio y presentar el resultado, caso contrario presentar DATO INCORRECTO y vuelva a ingresar datos.

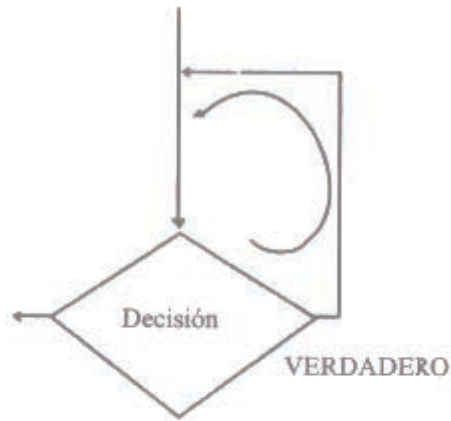


Estructura repetitiva MIENTRA – FINMIENTRA (WHILE DO - END WHILE)

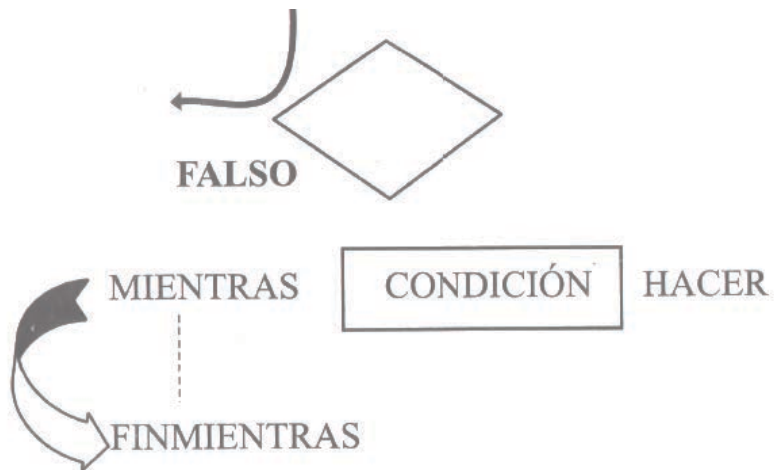
Figura 43: Estructura MIENTRA - FINMIENTRA



Si la condición de la estructura es verdadera, el computador leerá las instrucciones hasta repetitivamente

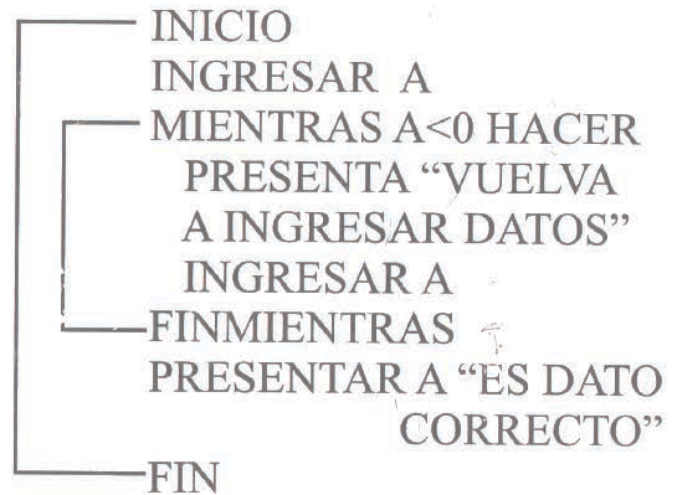
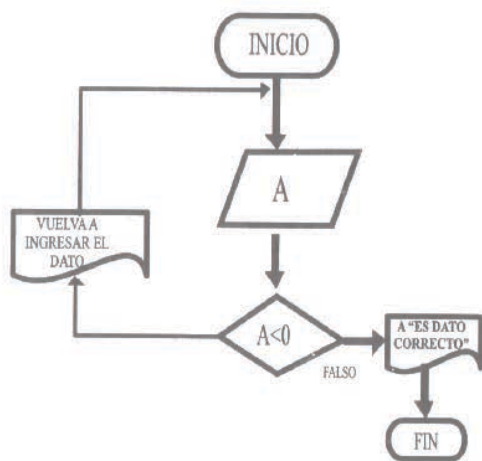


Si la condición es falsa el computador saltara a leer la instrucción FINMIENTRAS



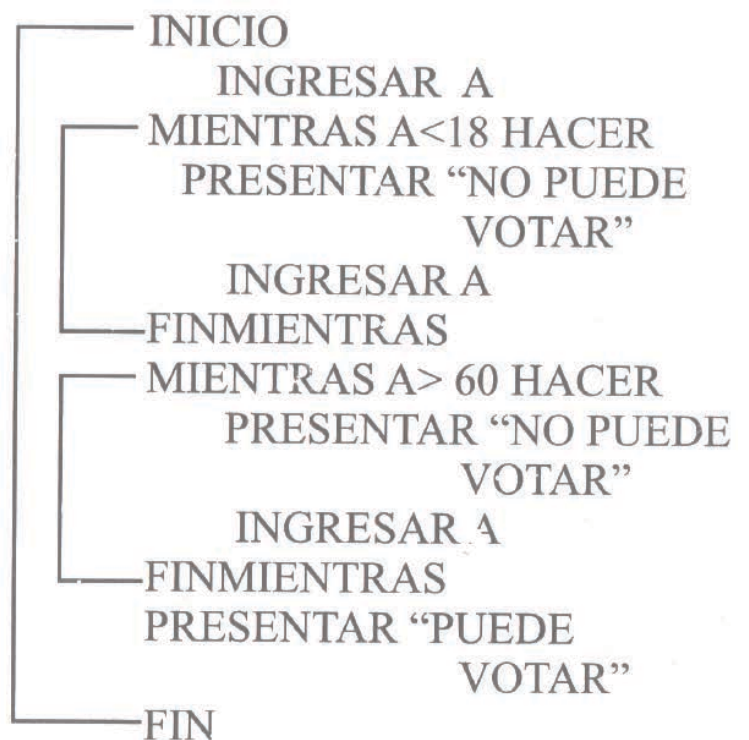
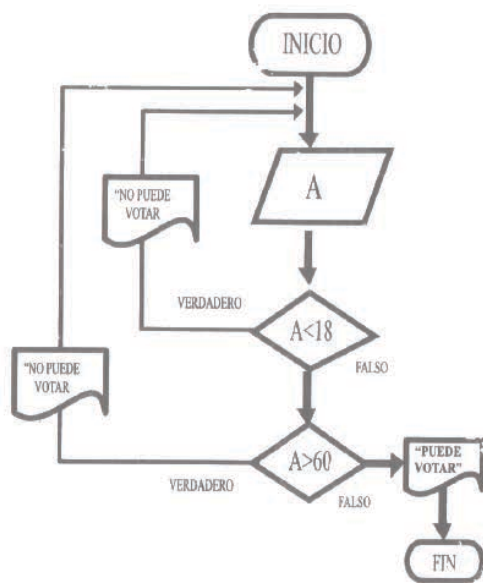
Ejemplo#1

Realizar un diagrama de flujo para ingresar por teclado solo números positivos y presentar un mensaje de DATOS CORRECTOS, caso contrario presentar VUELVA A INGRESAR LOS DATOS, luego finalice.



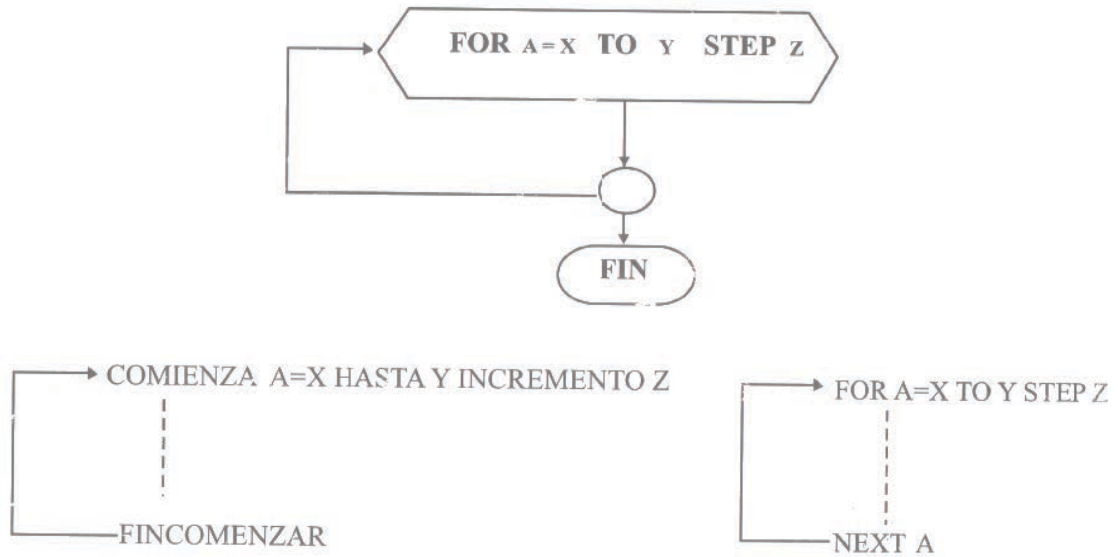
Ejemplo#2

En el Consejo Nacional Electoral se desea saber si la edad de las personas ingresada por periférico que van votar están dentro del rango obligatorio, es decir, mayor o igual a 18 y menor o igual a 60, si cumple con esta dos condiciones presente PUEDE VOTAR, caso contrario NO PUEDE VOTAR, luego finalice



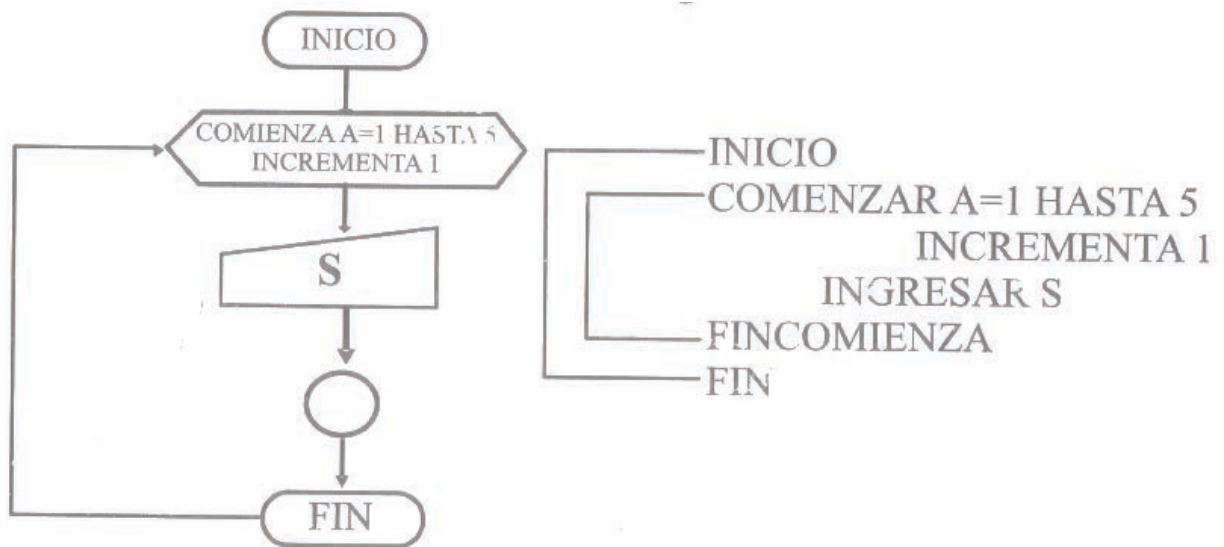
Estructura COMIENZA-HASTA-INCREMENTO (FOR-TO--STEP-NEXT)

Figura 43: Estructura COMIENZA-HASTA-INCREMENTO



Ejercicio#1

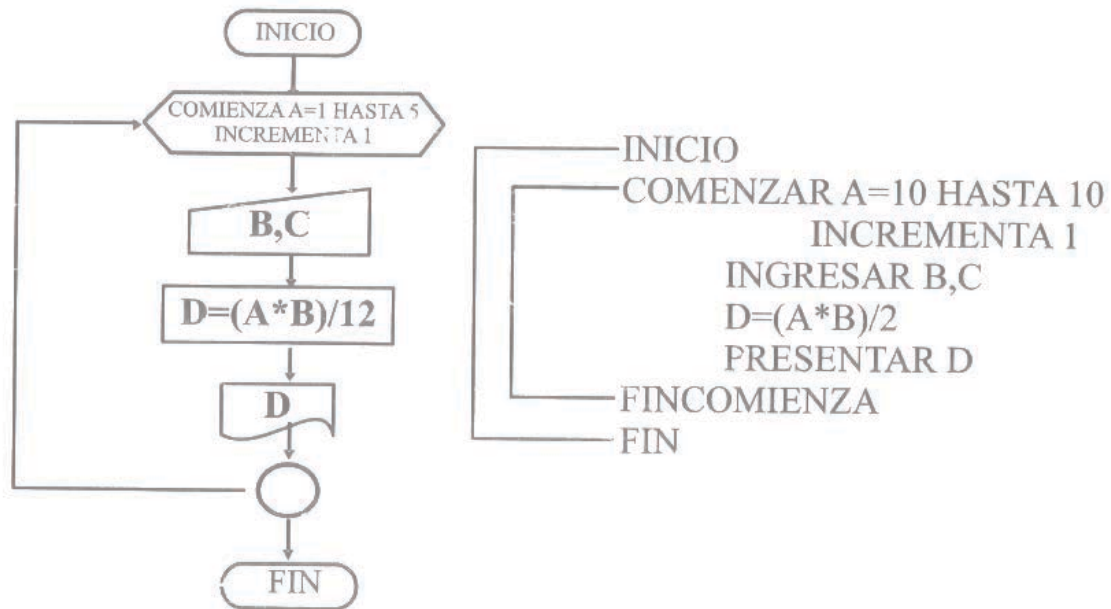
Construya un diagrama de flujo que realice 4 iteraciones, visualice el número de iteracioneo ciclo, luego finalizar



Ejemplo#2

Un hacendado desea saber cuál es el área de cada una de sus 10 haciendas de forma triangular rectangular, ingrese los datos por periférico y presentar cada uno de los resultados. Realizar el ejercicio con ciclos FOR para el ingreso.

NOTA: No podemos ingresaremos valores negativos



EJERCICIOS RESUELTOS

1 -Escriba el pseudocódigo para resolver el problema del pago de la matrícula de los estudiantes de un colegio Para ayudar a los alumnos debido a la crisis económica, se establecen tres valores según el nivel socioeconómico de la familia.

Primer nivel 300000

Segundo nivel 200000

Tercer nivel 150000

A este valor tenemos que sumarle el valor de 15000

La variable que representa el nivel socioeconómico es A

$$D=A+15000$$

- a) Escriba el pseudocódigo para ingresar una matrícula y presentar el resultado.
- b) Escriba el pseudocódigo para ingresar 10 matrículas y presentar cada uno de los resultados utilice Contadores.
- c) Escriba el pseudocódigo para ingresar 10 matrículas y presentar cada uno de los resultados utilice ciclos FOR.
- d) Escriba el pseudocódigo para ingresar 10 matrículas y presentar cada uno de los resultados y su suma total utilice Acumuladores.

Pseudicódigo ítem a

```
INICIO
  REPEAT
    INGRESAR A
  HASTA QUE A=300000 OR A=200000 OR A=150000
  PROCESO D=A+150000
  PRESENTAR D
FIN
```

Pseudicódigo ítem b con contadores

```
INICIO
  PROCESO X=0
  REPETIR
    REPETIR
      INGRESAR A
    HASTA QUE A=300000 OR A=200000 OR A=150000
    PROCESO D=A+15000
    PRESENTAR D
    PROCESO X=X+1
  HASTA QUE X=10
FIN
```


Pseudicódigo ítem c con ciclos FOR

```
INICIO
  COMENZAR X=1 HASTA 10 INCREMENTO 1
    REPETIR
      INGRESAR A
      HASTA QUE A=300000 OR A=200000 OR A=150000
      PROCESO D=A+15000
      PRESENTAR D
    FINCOMENZAR
FIN
```

Pseudicódigo ítem d con acumuladores

```
INICIO
  PROCESO S=0
  PROCESO X=0
    REPETIR
      REPETIR
        INGRESAR A
        HASTA QUE A=300000 OR A=200000 OR A =150000
      PROCESO D=A+15000
      PRESENTAR D
      PROCESO X=X+1
      S=S+D
    HASTA QUE X=10
  PRESENTAR S
FIN
```



Bibliografía

- Cairo, O. (2005). *Metodología de programación* (3 ed.). México: Alfa Omega.
- Corona, M., & Ancona, M. (2011). *Diseño de algoritmos*. México: McGraw-Hill.
- Deitel, H., & Deitel, P. (2014). *Cómo programar en C* (9 ed.). México: Pearson Educación.
- Deitel, H., & Deitel, P. (2008). *C++ cómo programar* (3 ed.). México: Perason Educación.
- IEEE. (2017). *IEEE Spectrum*. Obtenido de <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>
- Joyanes, L. (2008). *Fundamentos de programación* (4 ed.). Madrid: McGraw-Hill.
- Mata, R., & Cushman, P. (2001). *Introducción a la programación*. México: McGraw-Hill.
- Microsoft. (2017). *Office*. Obtenido de https://support.office.com/es-es/article/Crear-un-diagrama-de-flujo-b%C3%A1sico-e207d975-4a51-4bfa-a356-eeec314bd276#OfficeVersion=2013_2016
- Morán, F., & Morán, F. (2017). *Tecnología e Infofrmática*. Guayaquil: Compas.
- Real Academia Española. (2014). *Diccionario de la lengua española* (23 ed.). Madrid: Espasa.
- Zahonero, I., & Joyanes, L. (2005). *Programación en C* (2 ed.). Madrid: McGraw-Hill.

FRANCISCO LENIN MORÁN PEÑA

Docente Titular Principal Tiempo Completo, Director de Carrera Sistemas Multimedia, Director de Laboratorios de Tecnologías, Docente de Grado y Post Grado, Consultor Académico de Grado y Post Grado en la Universidad de Guayaquil. Doctorado en Educación y Sociedad (Cursando) Universidad de Barcelona, Doctor en Ciencias de la Educación, Magíster en Educación Superior, Especialista en Proyectos de Desarrollo Educativos y Sociales

FRANCISCO ENRIQUE MORÁN PEÑA

Universidad de Barcelona doctorando en educación y sociedad Universidad de Guayaquil docente titular principal universidad, posgrado Universidad de Guayaquil, Magister en Educación Superior Posgrado Universidad de Guayaquil especialista en proyectos de desarrollo educativos y sociales Posgrado Universidad de Guayaquil diploma superior en evaluación y acreditación de la educación superior tercer nivel Escuela Superior Politécnica del Litoral ingeniero en electrónica y telecomunicaciones tercer nivel universidad de Guayaquil licenciado en ciencias de la educación especialización informática tercer nivel Universidad de Guayaquil profesor en informática tercer nivel Universidad de Guayaquil tecnólogo pedagógico en informática

ISBN: 978-9942-750-75-4



9 789942 750754