



TECNOLÓGICO
NACIONAL DE MÉXICO®
CAMPUS CERRO AZUL

Guía Para Curso PROPEDÉUTICO 2022

Docente: Claudia Elizabeth Blanco Villanueva



SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

CURSO DE INDUCCIÓN A LAS CIENCIAS COMPUTACIONALES

Carretera Tuxpan-Tampico Km. 60, Cerro Azul, Ver.
C.P. 92519, Apartado Postal No. 118
Tels. +52(785) 85-89-100 Ext. 3220

No.	Temas	Página
	INTRODUCCIÓN	4
1	CONCEPTOS BÁSICOS	5
	1.1 Computadora	5
	1.1.1. Concepto	6
	1.1.2. Clasificación	6
	1.2 Componentes	9
	1.2.1 Hardware	10
	1.2.2 Software	10
	1.3 Dispositivos periféricos	11
	1.3.1 Entrada	11
	1.3.2 Salida	12
	1.3.3 Entrada/Salida	13
	1.3.4 Almacenamiento	13
	1.4 Caja negra	14
	1.4.1 Conceptos: Dato, proceso, información	14
2	LÓGICA FUNCIONAL	16
	2.1 Pensamiento lógico	16
	2.2 Resolución de problemas	17
	2.3 Evaluación de expresiones	18
	2.4 Operadores lógicos, aritméticos y relacionales	19
	2.5 Jerarquías	22
	2.6 Aplicaciones	23
3	ALGORÍTMOS	24
	3.1 Concepto	24
	3.2 Tipos	26
	3.3 Algoritmo pseudocódigo	26
	3.4 Ejemplos	31
	3.5 Ejercicios	35
4	DIAGRAMACIÓN	36
	4.1 Concepto	37
	4.2 Tipos	37
	4.3 Diagramas de flujo	38
	4.3.1 Simbología	41
	4.3.2 Herramientas	42
	4.3.3 Ejemplos	43
	4.3.4 Ejercicios	49

INTRODUCCIÓN

Es fundamental, desde la inducción, introducir al estudiante en el tema de las ciencias computacionales. Este curso le brindará las bases necesarias para desarrollar, a lo largo de su formación profesional, capacidades propias de las carreras del área de sistemas y computación.

Los temas contenidos en este curso permiten propiciar habilidades y competencias que potencien el pensamiento lógico, analítico, creativo y estructurado, para la construcción de soluciones algorítmicas que serán implementadas a través de las computadoras.

Al término del curso el estudiante contará con las siguientes competencias:

- Conocer e identificar los conceptos básicos computacionales
- Conocer y aplicar operadores lógicos, aritméticos y relacionales en la solución de problemas mediante algoritmos computacionales.
- Conocer y aplicar diagramas de flujo en el diseño gráfico de soluciones de problemas computacionales.

Hacer de las computadoras y la tecnología una herramienta básica en nuestros días va más allá de una moda o un estilo de vida, es aplicar conocimiento y destrezas para la solución de problemas en cualquier área del conocimiento.

1. CONCEPTOS BÁSICOS

1.1 COMPUTADORA

Uno de los primeros dispositivos mecánicos para contar fue el ábaco, cuya historia se remonta a las antiguas civilizaciones griega y romana., aunque no se le puede llamar computadora por carecer de programa. Así desde al ábaco hasta los trabajos del ingeniero y matemático húngaro John von Neumann (1903 - 1957) considerado el padre de las computadoras, las computadoras han evolucionado, pasando de analógicas a digitales y convirtiéndose en herramientas indispensables en cada área de conocimiento.

1.1.1 CONCEPTO

Una **computadora** es un dispositivo informático que es capaz de recibir datos, almacenar y procesar información de una forma útil. Una computadora está programada para realizar operaciones lógicas, aritméticas o relacionales de forma automática.

Computadora: Dispositivo electrónico programable, capaz de procesar mucha información a gran velocidad y con gran exactitud.

¿Que es una computadora?



Dispositivo electrónico compuesto de distintos elementos capaz de realizar cálculos y procesar información.

1.1.2 CLASIFICACIÓN

Las computadoras pueden ser clasificadas de diferentes maneras, por su uso (de propósito general o específico), por su manera de procesar la información (analógica, digital, o híbrida) y por su capacidad de procesamiento, potencia y tamaño. Esta última clasificación es la general.

<https://youtu.be/LE448Xbb8uI>

Clasificación general



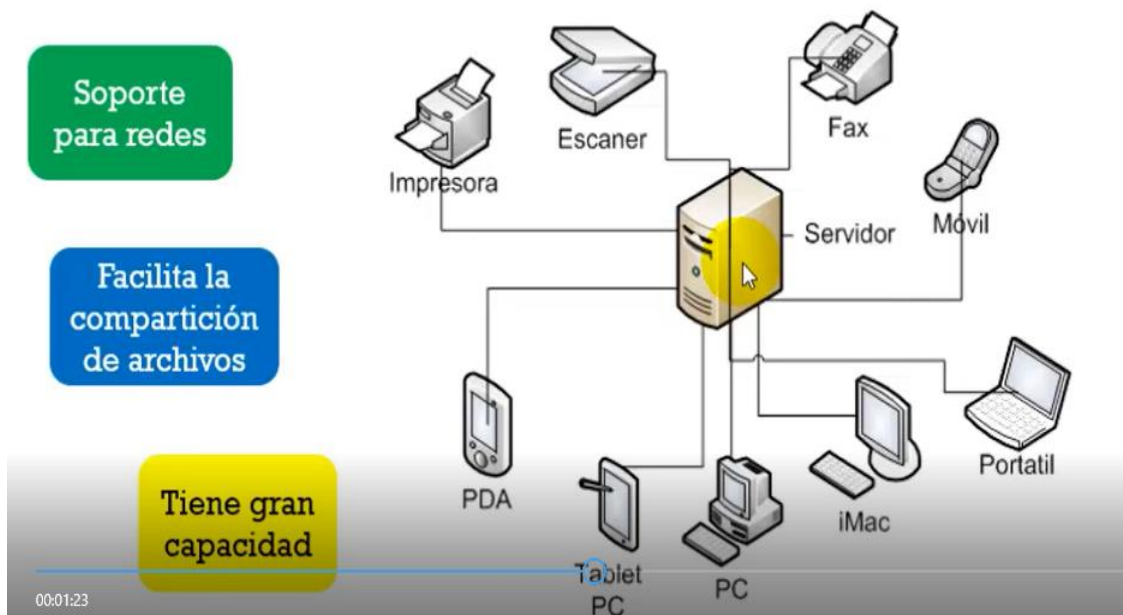
Computadoras personales (microcomputadoras)

Equipos de cómputo con características comunes destinados principalmente a labores que no requieren de grandes cantidades de procesamiento y cálculo, están presentes en las casas, escuelas, oficinas, etc.



En esta categoría entran las computadoras de escritorio y portátiles

SERVIDORES



Principalmente se usan para alojar páginas web y para servicios y administración de redes.

ESTACIONES DE TRABAJO

(WORKSTATION)



Equipos de altas capacidades orientados a la gestión de redes de ordenadores. Utiliza la arquitectura RISC (procesador de cómputo con un conjunto reducido de instrucciones).

Minicomputadoras



Equipos con características moderadas que permiten procesar grandes volúmenes de datos, se les encuentra normalmente destinadas al uso de servidores.

Macrocomputadoras (mainframes)

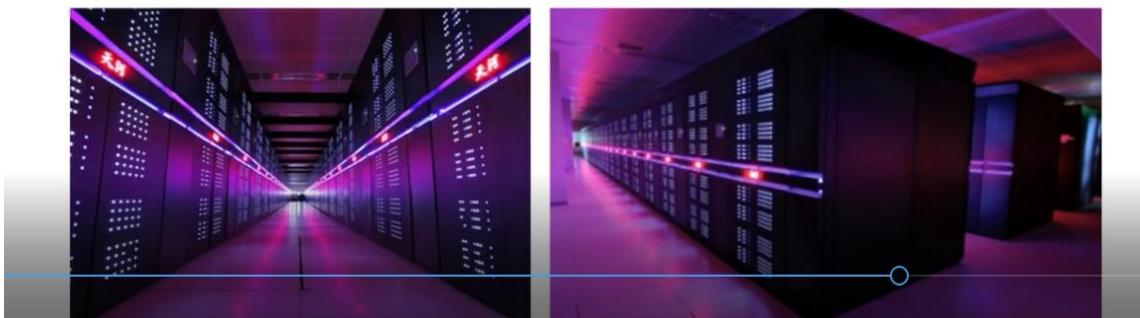


Computadoras destinadas a procesar y almacenar enormes cantidades de información además de soportar conexiones de miles de terminales o usuarios

Usadas en tráfico aéreo, bancos, bolsas de valores, casas de cambio, consorcios, etc.

Supercomputadoras

Son las computadoras más potentes, se le puede llamar supercomputadora a el ordenador más potente de cierta época a nivel mundial, estas bestias de la tecnología están orientadas principalmente a aplicaciones científicas y militares.



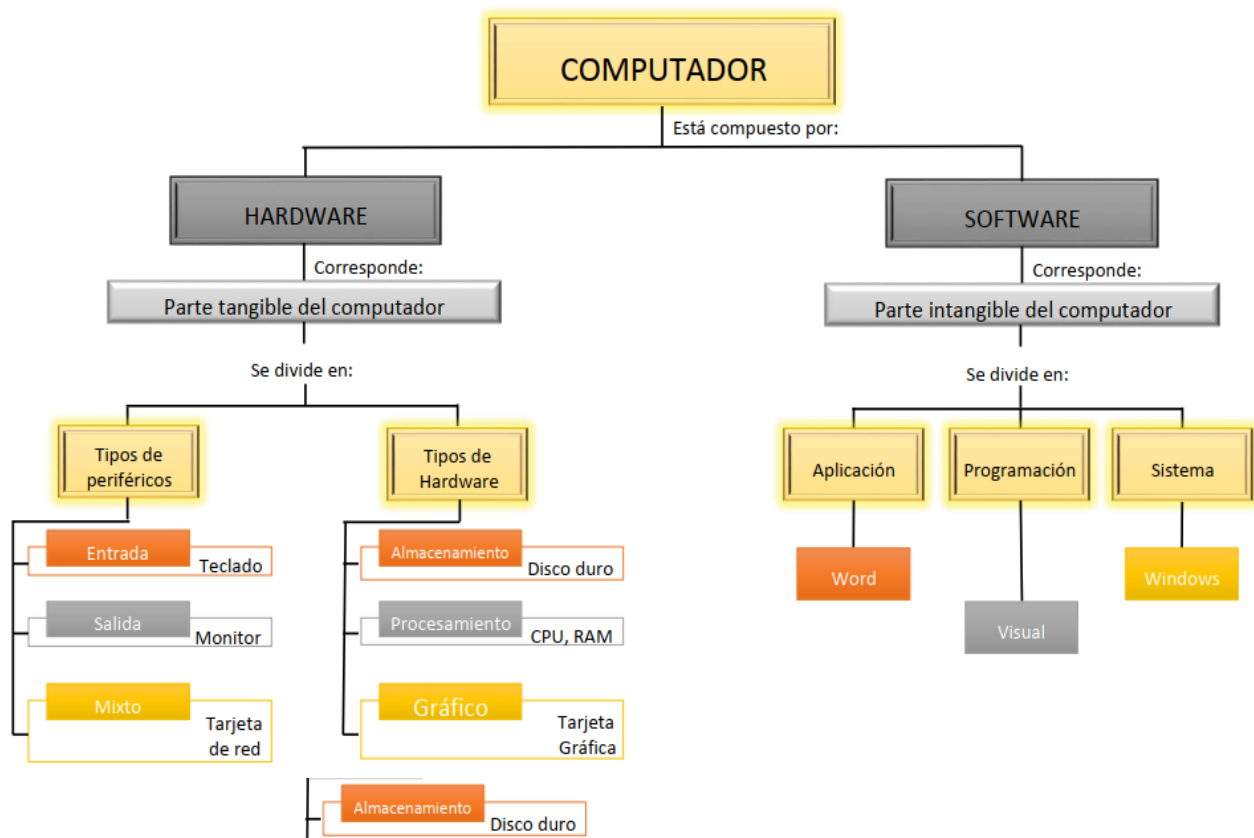
1.2 COMPONENTES

Los componentes físicos principales de una computadora son el núcleo básico y los dispositivos (periféricos).

Núcleo básico: la memoria (RAM y ROM), la tarjeta madre, el procesador, microprocesador o CPU (siglas en inglés para el concepto de 'unidad central de procesamiento', compuesta a su vez por la unidad de control y la unidad aritmético lógica).

Dispositivos o periféricos de entrada, salida, E/S y almacenamiento. Estos dispositivos auxiliares o periféricos son variados. Algunos de ellos son: el disco duro, el monitor, el ratón, el teclado, la impresora o los altavoces. Para que estos componentes físicos se intercomunicen y la computadora realice sus funciones se requieren de otros componentes lógicos que permiten que esto se lleve a cabo., es así como los componentes se dividen en hardware y software.

<https://youtu.be/zWRZ07RyeaQ>



Componentes hardware y software de una computadora

1.2.1 HARDWARE

- El hardware es la parte tangible de la computadora.
- Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

El hardware, equipo o soporte físico en informática se refiere a las partes físicas, tangibles, de un <u>sistema informático</u>, sus componentes eléctricos, electrónicos, electromecánicos y mecánicos. Los cables, así como los gabinetes o cajas, los <u>periféricos</u> de todo tipo, y cualquier otro elemento físico involucrado, componen el hardware o soporte físico; contrariamente, el soporte lógico e intangible es el llamado <u>software</u>.	1. Monitor. 2. Placa principal. 3. Microprocesador (CPU) y zócalo. 4. Un módulo de RAM y tres ranuras. 5. Dos tarjetas de expansión y tres ranuras. 6. Fuente de alimentación. 7. Unidad de disco óptico (CD, DVD, BD). 8. Unidad de disco duro o unidad de estado sólido. 9. Teclado. 10. Ratón.
---	--

Algunos componentes hardware del núcleo principal o básico de la computadora son:

MEMORIA RAM: (RANDOM ACCESS MEMORY), memoria de acceso aleatorio cuyo contenido permanecerá presente mientras la computadora permanezca encendida.

MEMORIA ROM: Memoria de solo lectura. Chip de memoria que solo almacena permanentemente instrucciones y datos de los fabricantes.

MICROPROCESADOR: Es el cerebro del ordenador. Es un chip, componente electrónico en cuyo interior existen miles (o millones) de elementos llamados transistores, cuya combinación permite realizar el trabajo que tenga encomendado el chip.

TARJETA MADRE (mainboard), o placa madre (motherboard), es el elemento principal de todo ordenador, al que se conectan todos los demás aparatos y dispositivos.

1.2.2 SOFTWARE

El software es el conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadora, es decir, la parte intangible.

Se conoce como software, logicial o soporte lógico al sistema formal de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.	1. Microsoft Office (Word, Excel, PowerPoint, Outlook) 2. Adobe Reader 3. Adobe Photoshop 4. Adobe FrontPage 5. AutoCAD 6. Mozilla Firefox 7. Google Chrome 7. Internet Explorer 8. Real Player 9. Microsoft Visual Basic 10. Microsoft Windows 11. Linux 12. Unix
---	--

1.3 DISPOSITIVOS Y PERIFÉRICOS.

PERIFERICOS: cualquier dispositivo de hardware E/S conectado a una computadora

- **Un periférico es un dispositivo físico que se conecta a la computadora, pero no forma parte del núcleo básico (CPU, memoria, placa base, puertos, etc.).**
- **Los periféricos sirven para comunicar la computadora con el exterior o para almacenamiento de información.**
- **Los periféricos suelen poder conectarse a los distintos puertos de la computadora.**
- **Estos pueden conectarse o desconectarse de la computadora, pero ella seguirá funcionando, aun sin ellos.**

INTERFAZ: Se denomina así a todo aquel medio físico que conecta un dispositivo periférico con la computadora; también se le conoce así a todo el software que comunica al usuario con la misma.

<https://youtu.be/T1abc6Qtzvw>

1.3.1 DISPOSITIVOS PERIFÉRICOS DE ENTRADA

Dispositivos de Entrada

- **Los dispositivos de entrada sirven para introducir datos y órdenes a la computadora, estos datos son convertidos a un formato digital que la computadora pueda interpretar.**

Principales dispositivos de Entrada.



Dispositivos más representativos de entrada:

TECLADO: Es un periférico de entrada, que convierte la acción mecánica de pulsar una serie de pulsos eléctricos codificados que permiten identificarla. Las teclas que lo constituyen sirven para entrar caracteres alfanuméricos y comandos a una computadora.

MOUSE: El ratón o mouse informático es un dispositivo señalador de entrada.

1.3.2 DISPOSITIVOS PERIFÉRICOS DE SALIDA

Dispositivos de Salida

- Los dispositivos de salida sirven para que la computadora envíe datos al exterior, es decir fuera de ella, estos datos pasan del formato digital a un formato que el usuario pueda entender, como texto, imágenes o sonidos.

Principales dispositivos de Salida.



Dispositivo más representativo de salida:

MONITOR: Es el periférico más utilizado en la actualidad para obtener la salida de las operaciones realizadas por la computadora, las pantallas de los sistemas informáticos muestran una imagen del resultado de la información procesada por la computadora

1.3.3 DISPOSITIVOS ENTRADA/SALIDA

Estos dispositivos integran características de los dispositivos de entrada y se combinan con los de salida, permitiendo que en uno sólo se disponga de las ambas funciones, facilitando la comunicación con el usuario.



1.3.4 DISPOSITIVOS DE ALMACENAMIENTO

Dispositivos de almacenamiento secundario: Discos de estado sólido, dispositivos de respaldo, CD, discos ópticos, D.D. externo, tarjetas flash (usb), microSD, SD.



1.4 CAJA NEGRA

1.4.1 CONCEPTOS

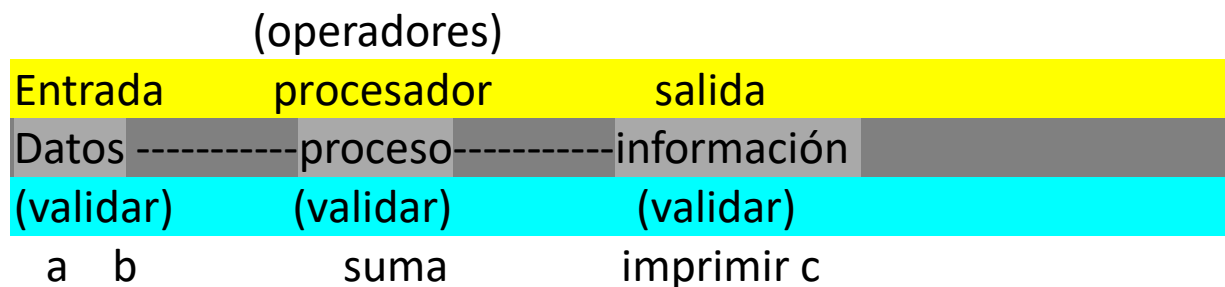
DATO: El término que usamos para describir las señales con las cuales trabaja la computadora es dato; Aunque las palabras dato e información muchas veces son usada indistintamente, si existe una diferencia importante entre ellas. En un sentido estricto, los datos son las señales individuales en bruto y sin ningún significado que manipulan las computadoras para producir información.

PROCESO: Es la forma con la que vamos a obtener la solución al problema. Es gestionar los datos con ayuda de un procesador informático. Son las operaciones para transformar datos en información.

INFORMACION: Es lo que se obtiene del procesamiento de datos, es el resultado final.

ASCII: Código americano estándar para intercambio de información, se utiliza en la mayor parte de las comunicaciones y se encuentra en el código de caracteres incorporado en la mayoría de los microcomputadores y en todos los computadores personales.

Diagrama: caja negra



ACTIVIDAD

Contestar las siguientes preguntas:

1. Mencione 3 ejemplos de dispositivos de entrada.
2. Mencione 3 ejemplos de dispositivos de salida.
3. ¿A qué se le considera hardware?
4. ¿Qué es el software?
5. Mencione 3 ejemplos de unidades de almacenamiento.
6. ¿Qué son los puertos?
7. Mencione 3 ejemplos de dispositivos entrada/salida.
8. Mencione la clasificación general de las computadoras.
9. ¿Qué es dato?
10. ¿Qué es información?

2. LÓGICA FUNCIONAL

Lógica es la técnica utilizada para desarrollar instrucciones en una secuencia para lograr determinado objetivo. La lógica se puede definir en términos de informática como la acción de pensar en forma estructurada y coherente sobre las soluciones a conflictos que se deseen automatizar por medio de programas o sistemas de cómputo. Es la organización y planificación de instrucciones en un algoritmo, con el objetivo de tornar visible la implementación de un programa o software.

El razonamiento o pensamiento lógico es la facultad que tenemos los seres humanos de resolver problemas, evaluar opciones y tomar la mejor opción.



Pensamiento lógico

2.1 PENSAMIENTO LÓGICO

El pensamiento lógico es:

- Deductivo: Parte de ideas generales para llegar a conclusiones particulares.
- Analítico: Desglosa las ideas para facilitar su entendimiento.
- Organizado: Ordena las ideas para analizarlas una detrás de otra.

El pensamiento lógico facilita la abstracción del problema, es decir, la identificación de datos(insumos) relevantes de una situación o problema a resolver por medio de algoritmos.

Aunque no existen parámetros o fórmulas que permitan al programador aprender a desarrollar una lógica correcta, existen ciertos consejos para guiar la lógica del programador:

1. Entender correctamente el problema: Es necesario entender el objetivo principal del algoritmo, diagrama o programa que se va a desarrollar.
2. Identificar todas las variables que afectan al problema: Se deben identificar todos los datos que alimentarán al proceso para obtener los resultados correctos.
3. Conocer la forma de presentar los resultados: Es importante establecer la forma en que los resultados arrojados se van a presentar, desde el periférico hasta la interfaz.
4. Optimizar recursos: Se debe optimizar los recursos con que cuenta la computadora en donde se va a ejecutar el programa para que sea eficiente y productivo.

2.2 RESOLUCIÓN DE PROBLEMAS.

Para resolver problemas a través de la lógica se requieren los siguientes pasos.

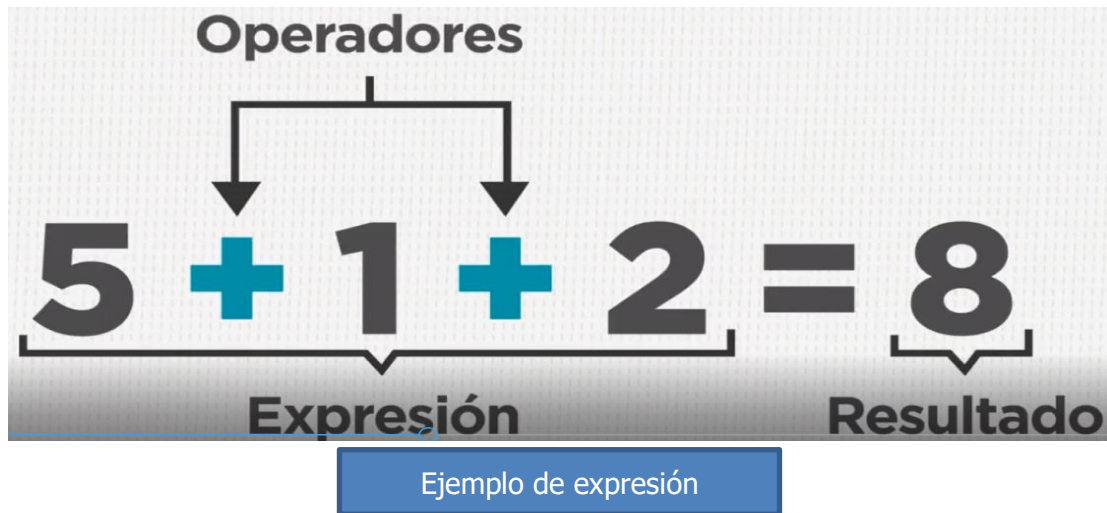
1. Identificar las necesidades, el problema o situación.
2. Abstracter los elementos, datos o insumos.
3. Identificar condiciones o restricciones.
4. Solucionar el problema a través de algoritmos: pseudocódigo y diagrama de flujo.
5. Evaluar la solución o comprobar si la solución es efectiva.



2.3 EVALUACIÓN DE EXPRESIONES.

En el área de las matemáticas y en computación, una expresión es una combinación de valores y operaciones que, al ser evaluados entregan un valor.

Una expresión es una combinación de valores literales, constantes, variables, funciones, palabras reservadas y operadores, que es interpretada de acuerdo con las normas particulares de precedencia y asociación para un lenguaje de programación en particular.



Toda expresión regresa un valor. Si hay más de un operador, se evalúan primero los operadores de mayor precedencia, en caso de empate, se aplica la regla asociativa.

Existen tres reglas de prioridad a seguir para evaluar una expresión:

1. Primero se evalúan los paréntesis (en caso de tener)
2. Después se sigue el orden de prioridad de los operadores presentes en la expresión
3. Por último, si aparecen dos o más operadores iguales, se evalúan de izquierda a derecha.

Se puede construir una expresión válida por medio de:

- Una sola constante o variable, la cual puede estar precedida por un signo + ó -.
- Una secuencia de términos (constantes, variables, funciones, etc.) separados por operadores.

Se debe considerar que toda variable utilizada en una expresión debe tener un valor almacenado para que la expresión dé como resultado un valor.

2.4 OPERADORES ARITMÉTICOS, RELACIONALES Y LÓGICOS.

Un operador es un símbolo en una expresión que representa una operación aplicada a los valores sobre los que actúa. Los valores sobre los que actúa un operador se llaman operandos. Un operador binario es el que tiene dos operandos, mientras que un operador unario es el que tiene sólo uno.

Por ejemplo, en la expresión $2.0 + x$ el operador $+$ es un operador binario que en este contexto representa la operación de adición. Sus operandos son 2.0 y x . Las operaciones más comunes se pueden clasificar en: aritméticas, relacionales, lógicas y de texto.

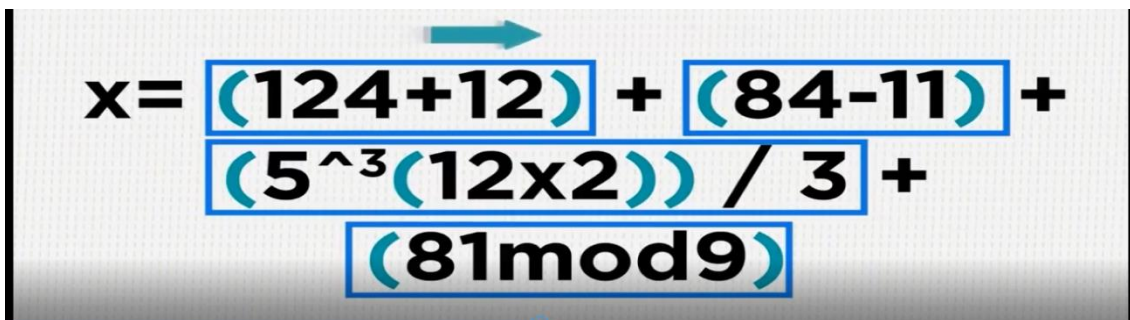
Precedencia(Jerarquía).

La precedencia o jerarquía de operadores es un conjunto de reglas que especifica en qué orden deben ser evaluadas las operaciones de una expresión. La precedencia está dada por la siguiente lista, en que los operadores han sido listados en orden de menor a mayor precedencia.

- or
- and
- not
- $<, <=, >, >=, !=, ==$
- $+, -$ (suma, resta)
- $*, /, \%$
- $+, -$ (positivo, negativo)
- $**$

La función de los paréntesis en una expresión.

Los paréntesis sirven para agrupar elementos, expresiones u operaciones dentro de una expresión. Con el uso de los paréntesis se altera la precedencia normal de los operadores y su asociatividad, permitiendo aislar expresiones condicionales; e indicar llamadas a funciones, además de señalar los parámetros requeridos en dichas funciones.



$$x = (124+12) + (84-11) + ((5^3(12x2))/3) + (81\text{mod}9)$$

Uso de paréntesis en una expresión

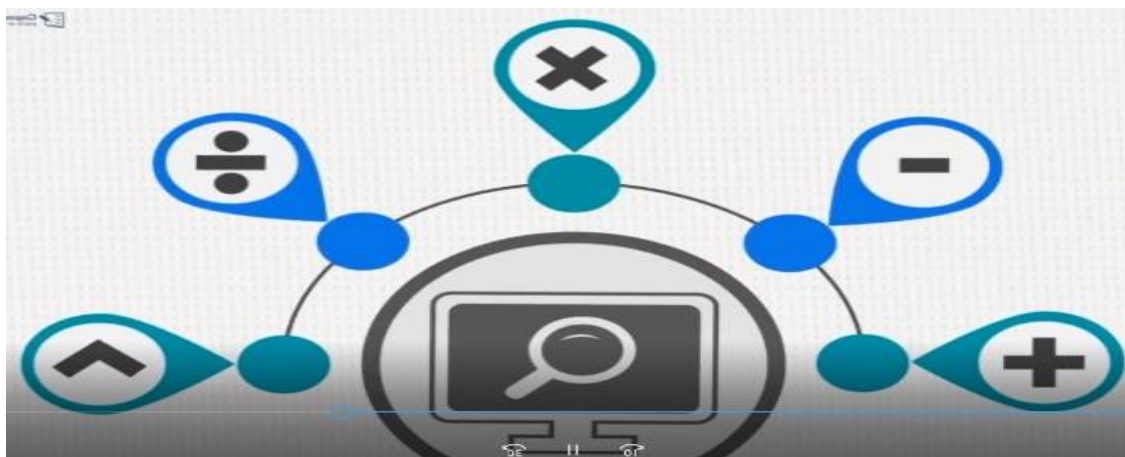
OPERADORES ARITMÉTICOS

Básicamente los operadores aritméticos corresponden a las operaciones aritméticas básicas: suma, resta, multiplicación y división; sin embargo, otras operaciones que se incluyen son la división modular y la potencia.

Existen dos tipos de operadores matemáticos: unarios y binarios.

- Los operadores unarios son operadores aritméticos que realizan una acción sobre un solo operando.
- Los operadores binarios requieren de dos operandos para ejercer acción sobre ellos.

Los operadores aritméticos deben seguir un orden para su ejecución, en este caso de mayor a menor prioridad:



Operadores aritméticos

Orden de prioridad(mayor a menor) de los operadores aritméticos.

- () Los paréntesis
- ^ Potenciación, $\sqrt{\quad}$ radicación, MOD residuo de la división.
- * Multiplicación, / división, % división entera (RES)
- + Suma, - resta

OPERADORES RELACIONALES

Los operadores relacionales o de comparación, son símbolos que se usan para comparar dos valores. Si el resultado de la comparación es correcto, la expresión considerada es correcta de lo contrario es falsa.

A continuación, se muestra la prioridad de los operadores relacionales:

Menor que (<)

Mayor que (>)

Menor o igual que (<=)

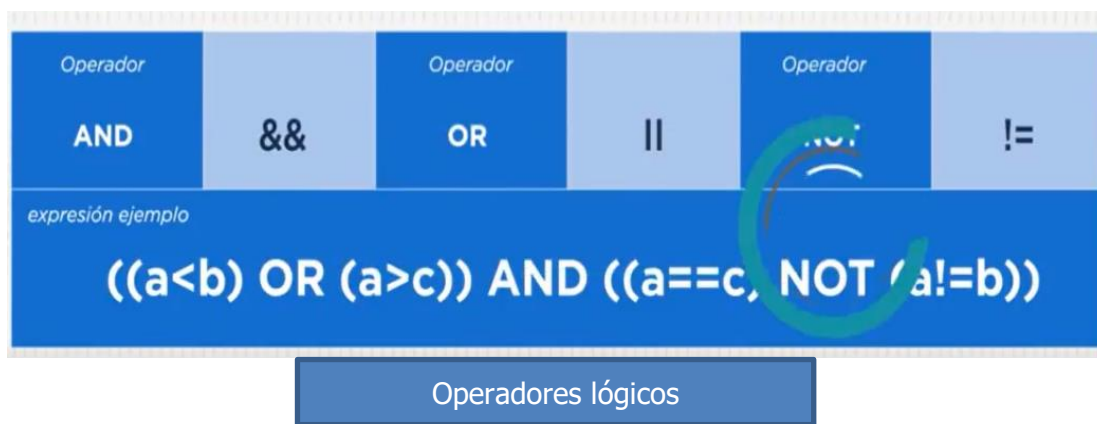
Mayor que o igual que (>=)

Igual que (==)

Desigual que (!=)

OPERADORES LÓGICOS

Los **operadores lógicos** son los que tienen operandos y resultado de tipo lógico.



En lo general existen tres operadores lógicos básicos:

- la conjunción lógica **and** («y»)
- la disyunción lógica **or** («o»)
- la negación lógica **not** («no»)

p	q	p and q	p or q	not p
True	True	True	True	False
True	False	False	True	
False	True	False	True	True
False	False	False	False	

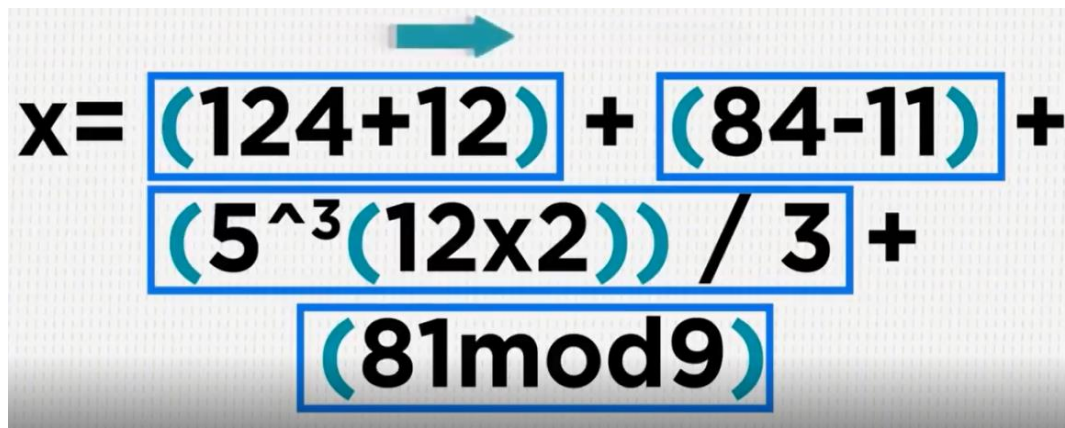
2.5 JERARQUÍAS

La jerarquía o precedencia de operadores es una regla que estipula el orden en que deben ser ejecutadas las operaciones en una expresión numérica con operaciones múltiples.

Las 3 reglas de prioridad

- Primero se deben ejecutar las operaciones agrupadas en paréntesis
- Luego las potencias y raíces, en tercer lugar, las multiplicaciones y divisiones en orden de aparición, y finalmente
- las sumas y restas en orden de aparición de izquierda a derecha.

La jerarquía de operadores determina el orden en el que se resuelven las expresiones cuando se involucran operaciones aritméticas como la suma, resta, multiplicación, división, potencia, raíz y módulo de la división. Este orden es el que permite que una expresión aritmética cualquiera siempre tenga la misma interpretación ya sea resolviéndola en papel, en calculadora o en el computador.



$$x = (124+12) + (84-11) + ((5^3(12x2)) / 3) + (81\text{mod}9)$$

Jerarquía de operadores

La siguiente lista, en que los operadores han sido listados en orden de menor a mayor precedencia o jerarquía.

2.6 APLICACIONES

Prioridad	Operador	Significado	Ejemplo
1	()	Paréntesis	$(2+3)*5 = 25$
2	$\wedge$	Exponenciación	$4^2 = 16$
	$\sqrt{\quad}$	Radicación	$\sqrt{9} = 3$
	MOD	Residuo de la división	$5 \text{ MOD } 2 = 1$
3	*	Multiplicación	$2*4 = 8$
	/	División	$5/2 = 2.5$
4	+	Suma	$3 + 4 = 7$
	-	Resta	$8 - 5 = 3$

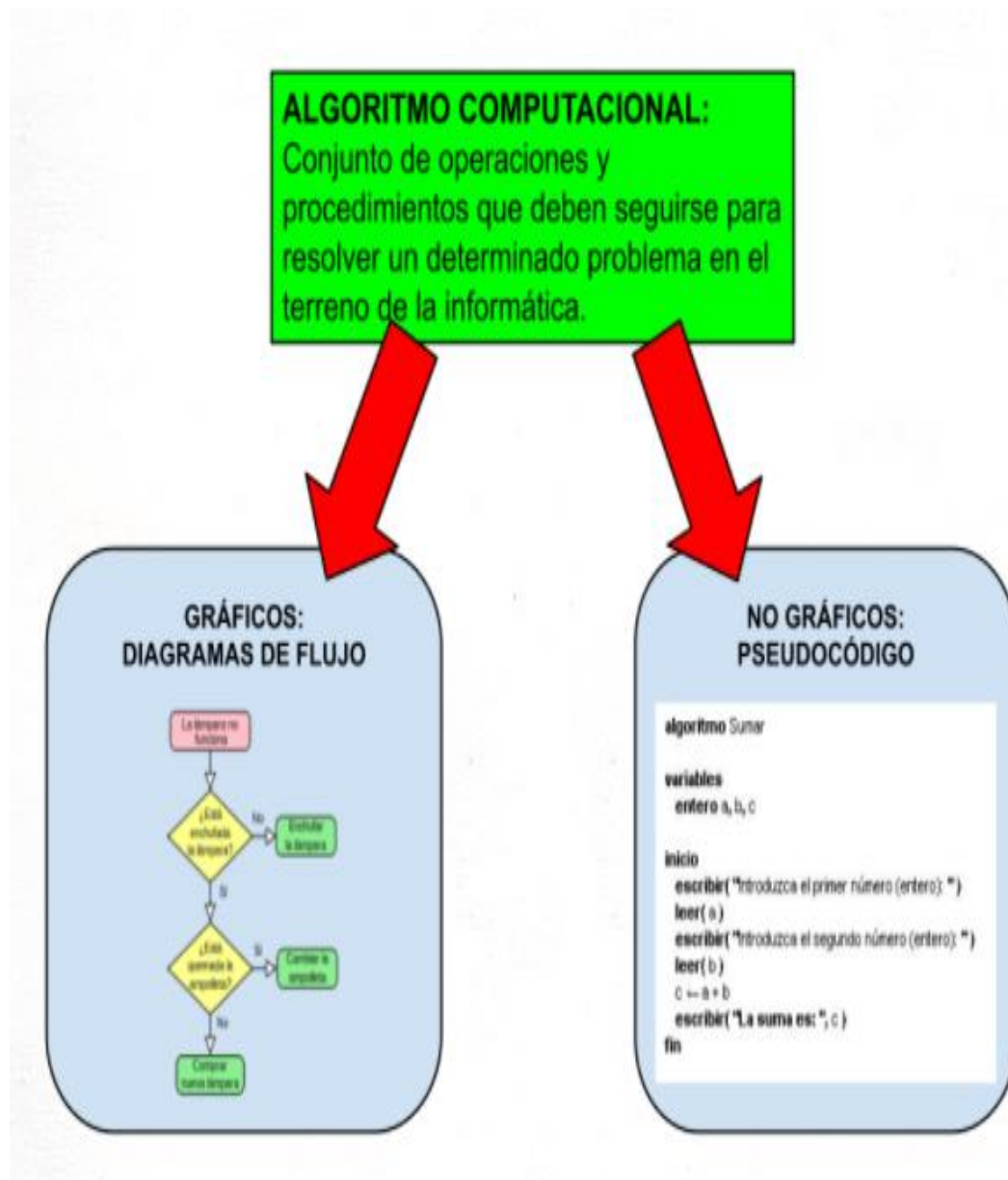
ACTIVIDAD

Evaluación de expresiones: Resolver las siguientes expresiones, usando jerarquía de símbolos matemáticos y exclusión de la misma mediante el uso de paréntesis.

1. $A=4, A*(3+5)=32$
2. $A=4, A3+5 =17$
3. $7 \times 10 - 15 \text{ mod } 3 \times 4 + 9 =$
4. $7 \times 10 - 15 \text{ div } 3 \times 4 + 9 =$
5. $(7 \times 10) - 15 \text{ mod } (3 \times (4 + 9)) =$
6. $7 \times (10 - 15) \text{ mod } (3 \times 4) + 9 =$
7. $A=8, B=3, B + A \times B - 6 =$
8. $A=8, B=3, (B + A) \times (B - 6) =$
9. $A=8, B=3, B + (A \times B) - 6 =$
10. $A=2, B=3, (5(A + B) + 3B) \text{ div } A =$
11. $A=2, B=3; 5(A + B) + (3B \text{ div } A) =$

3. ALGORITMOS

En la vida cotidiana, se emplean algoritmos frecuentemente para resolver problemas determinados. Algunos ejemplos son los manuales de usuario, que muestran algoritmos para usar un aparato, o las instrucciones que recibe un trabajador de su patrón. Algunos ejemplos en matemática son el algoritmo de multiplicación, para calcular el producto, el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para obtener el máximo común divisor de dos enteros positivos, o el método de Gauss para resolver un sistema de ecuaciones lineales.



3.1 Concepto

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

En términos de programación, un algoritmo es una secuencia lógica de pasos que permiten solucionar un problema o cumplir con un objetivo.

<https://youtu.be/akQtuSrr8jq>

Los algoritmos deben ser precisos e indicar el orden lógico de realización de cada uno de los pasos, debe ser definido y esto quiere decir que, si se ejecuta un algoritmo varias veces, se debe obtener siempre el mismo resultado, también debe ser finito o sea debe iniciar con una acción y terminar con un resultado o solución de un problema.

Cuando se elabora un algoritmo se debe tener en cuenta los siguiente:

1. Tener claro cuál es el problema o necesidad a solucionar.
2. Establecer un objetivo que permita medir la solución del problema.
3. Elaborar un algoritmo que solucione el problema
4. Realizar pruebas que verifique los resultados.

Los elementos de la estructura de un algoritmo son:

- Entrada (cabecera, Inicio o punto de partida y entrada de datos)
- Proceso (Instrucciones, Cuerpo)
- Salida (instrucciones puntuales de solución)

Características de los algoritmos.

- **Precisión:** Los pasos e instrucciones deben ser precisos, sin ambigüedad.
- **Legibilidad:** Es sumamente importante porque si no se puede entender su contenido, entonces las instrucciones no podrán ser realizadas. Esto influye en que debe tener una redacción clara, directa y concisa del contenido de las instrucciones.
- **Límite o finitud:** Los algoritmos son finitos, esto significa que deben tener un fin y ofrecer un resultado final.
- **Representación:** Los algoritmos se representan a través del lenguaje verbal, diagrama de flujo, código, lenguaje de programación, diagrama visual y operaciones matemáticas

3.2 Tipos

Tipos de algoritmos según el sistema de signos con el que se describen los pasos a seguir.

1.-Algoritmos cualitativos:

Son instrucciones paso a paso que se dan de forma oral o escrita, como las recetas de cocina.

2.-Algoritmos cuantitativos

Al contrario del anterior, este se basa en instrucciones numéricas, especialmente para encontrar el resultado de algún cálculo o ecuación.

3.-Algoritmos computacionales

Son algoritmos complejos, deben ser realizados a través de una computadora y dan como resultado un algoritmo cuantitativo optimizado.

De estos tipos de algoritmos se pueden generar:

Algoritmos pseudocódigos, Diagramas de flujo y Tablas de decisión.

3.3 ALGORITMO PSEUDOCÓDIGO.

Los algoritmos pseudocódigos son la representación de los pasos a seguir para resolver un problema, necesidad o situación mediante la utilización de frases (sentencias, instrucciones, oraciones) en lenguaje común, integrando palabras en lenguaje de programación y algunas palabras claves.

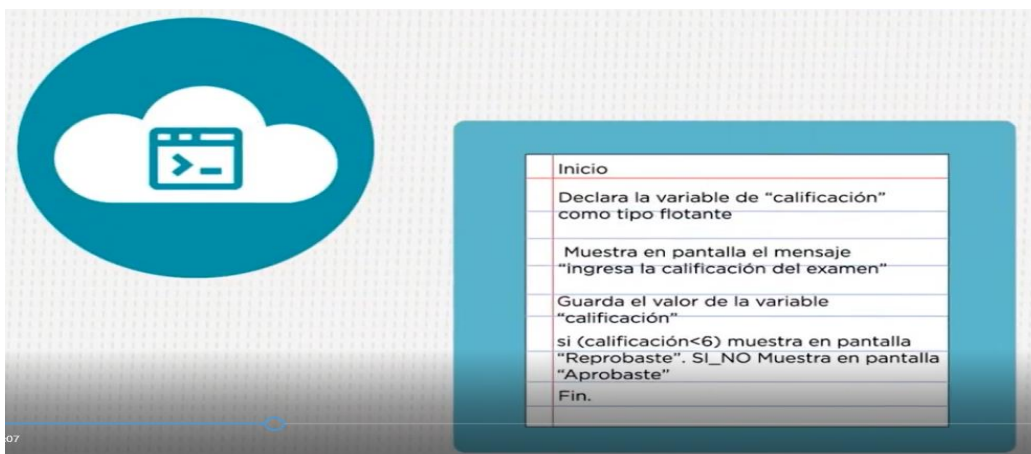


Ventajas.

- Permite elaborar algoritmos estructurados
- Es sencillo de diseñar y fácil su programación.
- Es ideal para representar situaciones repetitivas.

Desventajas.

- Requiere de una lógica bien fundamentada.
- Se requiere conocer la sintaxis del lenguaje de programación bajo el cual se va a elaborar el pseudocódigo.



Estructura de los algoritmos pseudocódigos.

Por lo general, todo pseudocódigo puede contener tres partes las cuales van a depender del problema específico.

1. Inicio

Esta parte consta de:

- Inicializar y declarar variables a usar.
- Abrir archivos
- Asignar valores o introducirlos por teclado

2. Instrucciones simples e instrucciones repetitivas o ciclos

- Esta parte es el cuerpo del programa, ya sea que se repita o se realiza una sola vez.

3. Final

- Esta parte consta de impresión de resultados, cierres de archivos y salidas de información.

Elementos de un algoritmo pseudocódigo.

- Bucle o ciclo (instrucciones iterativas)
- Contadores
- Sumadores o acumuladores
- Tomas de decisión (instrucciones de bifurcación)
- Palabras reservadas(inicio, fin, etc)
- Caracteres especiales(i#\$&', etc)
- Variables, constantes e identificadores(datos, funciones,etc.)

Bucles o ciclos.

Se le conoce como bucle o ciclo a todo aquel proceso que se repite un número determinado de veces dentro de un algoritmo o programa.

Este tipo de instrucciones lleva condiciones. La condición del ciclo o bucle no tiene que ser única, puede haber más de una siempre y cuando estén unidas por los operadores lógicos (OR, AND y NOT)

Si la condición se cumple se ejecutan todas las instrucciones del ciclo.

Si la condición no se cumple, no entra al ciclo y se ejecutan todas las instrucciones fuera de él.

Contadores

Un contador es una variable destinada a contener valores que se pueden incrementar o decrementar. Este incremento o decremento siempre será constante.

Para usar un contador la variable debe ser previamente inicializada

La variable puede ser incrementada o decrementada por una constante o por otra variable siempre y cuando intervenga el valor anterior de la variable contador.

Sumadores

Un sumador o un acumulador es una variable que nos permite guardar un valor que se incrementa o decrementa de forma no constante durante el proceso.

La variable que funge como sumador debe ser inicializada.

El sumador puede ser incrementado o decrementado por una variable o por una operación algebraica que involucra simultáneamente varias variables en la instrucción.

Tomas de decisión.

Estas instrucciones se consideran instrucciones compuestas y son estructuras que permiten a los algoritmos ofrecer acciones alternativas o selectivas por lo tanto, la ejecución de una línea o grupo de líneas dependen de si se cumple o no una o varias condiciones.

Si la condición se cumple, se ejecutan todas las instrucciones que hay dentro de la sentencia o estructura de decisión (SI-entonces IF), y después todas las que está, fuera.

Si la condición no se cumple, se ejecutan sólo las instrucciones siguientes a la sentencia (IF).

Se puede tener más de una condición siempre y cuando están unidas por los operadores lógicos.

Existen tres tipos de instrucción de decisión IF

1. Simple. Sólo cuenta con una alternativa al cumplirse la condición.
2. Compuesta. Cuanta con dos alternativas o caminos. Cuando se cumple la condición y cuando no se cumple.
3. Anidada. Son más de dos IF dentro de cualquiera de los caminos o alternativas

Palabras reservadas

Las palabras reservadas son identificadores reservados predefinidos que tienen un significado especial y no se pueden utilizar como identificadores en sus programas

En los lenguajes y algoritmos informáticos, una palabra reservada es una palabra que tiene un significado gramatical especial para ese lenguaje y no puede ser utilizada como un identificador de objetos en códigos del mismo, como pueden ser las variables, constantes, funciones o tipos de datos.

Caracteres especiales

Los caracteres especiales definen la forma de ejecución de las sentencias de código de computadora o los comandos interpretados. Pueden ser un único carácter. Algunos representan un formato de entrada o salida (validar), y otros indican un comando directo al compilador

Identificadores, variables, constantes.

Identificadores: En general los identificadores son los nombres que se le dan a los datos, objetos, archivos, rutinas, datos, variables o elementos que vamos a manipular en un proceso, con ayuda de un algoritmo o un programa.

Variable: Es una zona de memoria, la cual es referenciada por medio de un nombre, en ella se pueden almacenar valores y se pueden cambiar cuando se deseen.

Constante: Es una zona de memoria, la cual es referenciada por medio de un nombre, en ella se pueden almacenar valores que no se pueden cambiar o permanece invariable a lo largo del proceso.

Tipo de dato: Especifica el tipo de valor que se puede almacenar en las variables o constantes. Estos valores pueden ser:

- Numéricas: Sólo pueden almacenar datos numéricos.
- Alfanuméricas: Almacena datos numéricos y alfanuméricos.
- Booleanas: Almacena datos lógicos (cierto /falso)

3.4 EJEMPLOS

EJEMPLOS DE ALGORITMOS (PSEUDOCÓDIGO)

1. Elabore un algoritmo que permita leer dos valores numéricos enteros distintos, en base a los valores, determinar cuál de los dos números es el mayor y escribirlo.

Pseudocódigo

```
1. INICIO
2. Inicializar variables
   a. a=0
   b. b=0
3. Leer los números desde teclado
   a. Leer a
   b. Leer b
4. Comparar  SI (a=b) entonces
               regresar al paso 3
   SINO son iguales (si la condición es falsa)
   SI (a<b) entonces
               imprime "EL NÚMERO MENOR ES : " a
               imprime "HOLA"
   SINO a no es mayor que b
               imprime "El número menor es: " b
   FIN_SI
FIN_SI
```

5. FIN

2. Realizar un algoritmo que permita leer tres valores distintos y almacenarlos en las variables A, B, y C respectivamente. El algoritmo debe imprimir cuál es el mayor.

1. INICIO
2. Inicializar Var numéricas
 - a. $A=0$
 - b. $B=0$
 - c. $C=0$
3. Leer valores A, B, C
4. Comparar IF (A=B) and (B=C) entonces
Imprimir "NO se pueden comparar los números porque son iguales"
ELSE
IF (A>B and A>C) ENTONCES
Imprime "El mayor es " A
ELSE
IF (B>A AND B>C) ENTONCES
Imprime "El mayor es " B
SINO
Imprime "El mayor es " C
FIN_IF
FIN_IF
FIN_IF
5. FIN

3. Elabore un algoritmo que realice la sumatoria de los números enteros comprendidos entre el 1 y el 10, es decir , $1+2+3+4+5+.....+10$. Al final imprimir la suma.

Pseudocódigo

1. INICIO
2. Var enteras $x=0$, $\text{suma}=0$
3. $x=x+1$
4. $\text{suma}=\text{suma}+x$
~~~~~
5. IF ( $x=10$ ) ENTONCES  
    Imprimir "la sumatoria de 1 al 10 es:" suma  
ELSE  
    Regresar al paso 3  
FIN\_IF
6. FIN

4. Desarrolle un algoritmo que permita determinar el área y volumen de un cilindro dado su radio (R) y altura (H).

**Pseudocódigo**

1.INICIO

2.Inicializar variables

a.  $r = 0$

b.  $h = 0$

c.  $v = 0$

d.  $a = 0$

3.Leer los datos de r y h desde teclado

a.Leer r

b.Leer h

4.  $\pi = 3.1416$

5.Obtener área

a.  $a = (2 * (\pi) * (r * h)) + (2 * (\pi * (r * r)))$

6.Obtener volumen

a.  $v = \pi * ((r * r) * h)$

7.imprimir "El área del cilindro es: " a

8.imprimir "El volumen del cilindro es: " v

9.FIN

```
1.Inicio
2.Var enteras  x= 10, s=0
3.s=s+x
4.x=x+2
5.IF (x=16) ENTONCES
    Imprimir "la suma de 10+12+14 es" s
ELSE
    Regresar al paso 3
FIN_IF
6.FIN
```

### 3.5 EJERCICIOS



## **ACTIVIDAD**

Resolver los siguientes algoritmos propuestos utilizando pseudocódigo

Ejercicios:

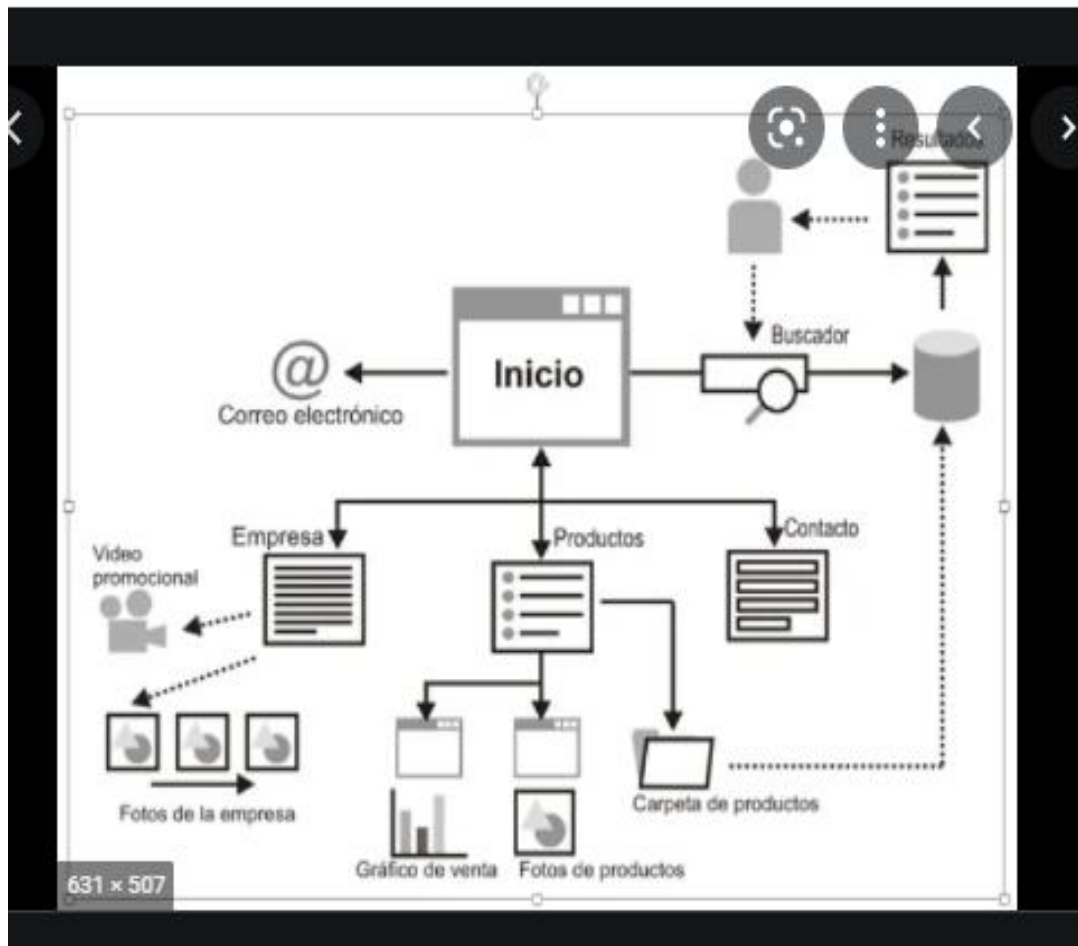
- a) Realizar un algoritmo permita leer dos valores numéricos enteros, determinar el menor y escribirlo.
- b) Realizar un algoritmo que sume dos números enteros positivos e imprima la suma en una tercera variable.
- c) Escribir un algoritmo para indicar si tres números enteros positivos leídos son iguales.
- d) Escribir un algoritmo para indicar cuál es el menor de tres números leídos.
- e) Modificar el algoritmo anterior para que imprima la suma de los tres valores
- f) Escribir un algoritmo que realice la sumatoria de los números enteros comprendidos entre el 1 y el 10, es decir,  $1+2+3+4+5+\dots+10$ . Sin considerar el número 10. Al final imprimir la suma.

## **4. DIAGRAMACIÓN**

La diagramación es un método de solución de problemas mediante la utilización de diagramas.

Este método requiere que los elementos estén ordenados y que permitan de manera visual seguir una secuencia y entenderla.

La diagramación permite representar soluciones gráficas a problemas o situaciones que requieran solución por esquematización.



Ejemplo de diagramación

### Tipos de diagramación:

- **Diagramación lineal:** Son aquellos diagramas que ilustran soluciones a

problemas sin la utilización de módulos o subrutinas.

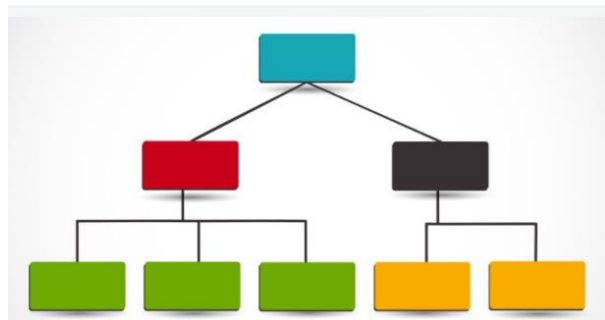
- **Ventajas:** Son más sencillos de elaborar y en el diagrama se muestra el proceso completo.
  - **Desventaja:** Mientras más complejo es el problema, más extenso se hace el diagrama, además no permite el uso de subrutinas o módulos.
- **Diagramación estructurada:** Son aquellos diagramas que ilustran soluciones a problemas a través de módulos o subrutinas que facilitan el desarrollo y entendimiento de un proceso.
    - **Ventajas:** Permite dar solución a problemas de cualquier índole, además de incluir la modulación, esta técnica es más sencilla de entender y de elaborar los procesos.
    - **Desventaja:** Requiere de mucho orden y lógica por parte del programador.

## 4.1 CONCEPTO

**Diagrama.** Es un conjunto de figuras esquemáticas que muestran paso a paso el procedimiento que se debe seguir en una determinada situación o problema.

Un diagrama es un diseño geométrico, cuya función es representar gráficamente procedimientos, procesos, ideas, soluciones, mecanismos o fenómenos, de tal modo que el "lector" pueda comprender de manera clara y rápida una información, y comprender también cómo actuar o qué esperar ante determinadas situaciones.

La palabra diagrama proviene del latín *diagramma*, y este del griego *διάγραμμα*, que quiere decir 'esquema'. Por lo que es un esquema que resume los elementos fundamentales de una información a través de elementos gráficos.



## 4.2 TIPOS

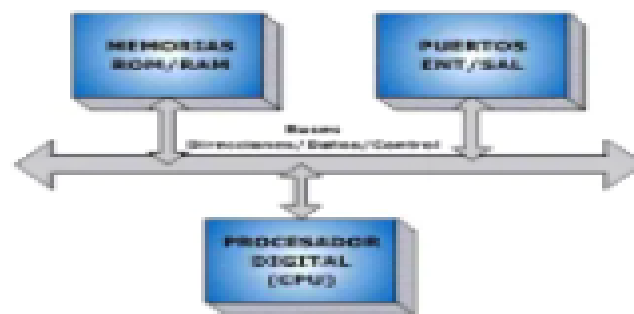
Los más representativos en las ciencias computacionales son:

1. Diagramas de sistema o de bloque
2. Diagramas de flujo.

En el diagrama de sistema o bloque se representan todos los dispositivos y/o máquinas que se emplearán durante un determinado proceso. En este tipo de diagramas las flechas indican cuáles dispositivos son entrada, de salida o e entrada y salida.

Las ventajas que tiene este diagrama es que permite identificar todos los dispositivos que se encuentran conectados a la computadora. A demás no requiere de diagrama externo y es sencillo de entender y elaborar.

Su desventaja es que es un diagrama muy general y no muestra la solución del problema, a diferencia de los diagramas de flujo que grafican el flujo de los datos, el camino que recorren hasta encontrar la solución.



Ejemplo de diagrama de bloque

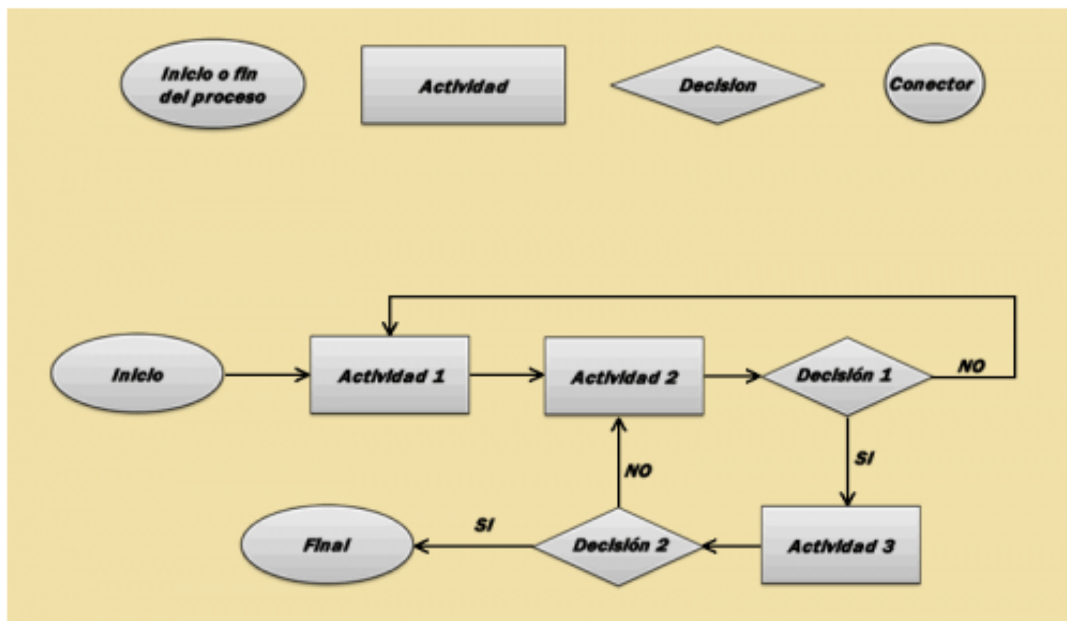
### 4.3 DIAGRAMAS DE FLUJO

El diagrama de flujo, también conocido como flujograma o diagrama de actividades, es muy común en la informática y en el mundo empresarial.

Un Diagrama de Flujo representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Su correcta construcción es sumamente importante porque, a partir del mismo se escribe un programa en algún Lenguaje de Programación.

Hay cuatro tipos de diagrama de flujo en base al modo de su representación:

- Horizontal. Va de derecha a izquierda, según el orden de la lectura.
- Vertical. Va de arriba hacia abajo, como una lista ordenada.
- Panorámico. Permiten ver el proceso entero en una sola hoja, usando el modelo vertical y el horizontal.
- Arquitectónico. Representa un itinerario de trabajo o un área de trabajo. – **Fuente:** <https://concepto.de/diagrama-de-flujo/>



Ejemplo de diagrama de flujo horizontal

Los diagramas de flujo son un mecanismo de control y descripción de procesos, que permiten una mayor organización, evaluación o replanteamiento de secuencias de actividades y procesos de distinta índole, dado que son versátiles y sencillos. Son empleados a menudo en disciplinas como la programación, la informática, la economía, las finanzas, los procesos industriales e incluso la psicología cognitiva. - **Fuente:** <https://concepto.de/diagrama-de-flujo/>

Un diagrama de flujo representa al programador, una panorámica más amplia y completa de las operaciones lógicas y/o aritméticas que se han de ejecutar durante un proceso.

Las principales ventajas son:

- Muestra cada uno de los pasos que se deben seguir para solucionar el problema
- Con la elaboración del diagrama de flujo es más sencillo elaborar la codificación del programa.

Desventajas:

- Mientras más grande sea el diagrama de flujo más complicada será su utilización.
- Cualquier modificación o actualización por mínima que sea, afecta a todo el diagrama.

### **Consejos antes de comenzar a la diagramación.**

Como se mencionó con anterioridad, un diagrama de flujo es la representación gráfica de lo que se desea realizar en la computadora.

Todo diagrama es en sí una obra de arte, por esta razón, ningún diagrama mostrará la misma solución para otro problema, aún cuando sean muy similares.

Antes de elaborar un diagrama es recomendable realizar los siguientes pasos:

1. Entender perfectamente el problema a automatizar.
2. Identificar todas las variables que afectan en forma directa la solución del problema.
3. Definir los formatos de entrada de datos (captura) y de salida de la información.
4. Elaborar el diagrama de bloque o sistema.

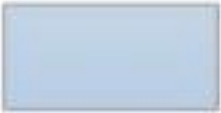
Habiendo reunido y verificado la información anterior, podemos iniciar al diseño del diagrama de flujo. Para ello se pueden considerar los siguientes puntos:

1. Ser creativo: No se debe dar por hecho ninguna situación, se deben considerar todas las posibles alternativas de solución.
2. Ser elocuente: El diagrama debe llevar una secuencia y flujo lógico.
3. Ser claro, el diagrama no debe presentar ambigüedades o secuencias confusas.
4. Ser organizado: Planear que el flujo lógico del diagrama comience de arriba hacia abajo y de izquierda a derecha de la hoja(según el caso horizontal o vertical)
5. Ser meticuloso: Diseñar o dibujar correctamente cada símbolo y flecha de flujo para evitar confusiones. Si es necesario anotar comentarios que faciliten la comprensión del diagrama.

Teniendo en cuenta estos pasos y consejos, lograremos que otras personas comprendan mejor nuestro diagrama de flujo.



### 4.3.1 SIMBOLOGÍA

| Símbolo                                                                             | Nombre           | Función                                                                                        |
|-------------------------------------------------------------------------------------|------------------|------------------------------------------------------------------------------------------------|
|    | Inicio / Final   | Representa el inicio y el final de un proceso                                                  |
|    | Línea de Flujo   | Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción. |
|    | Entrada / Salida | Representa la lectura de datos en la entrada y la impresión de datos en la salida              |
|   | Proceso          | Representa cualquier tipo de operación                                                         |
|  | Decisión         | Nos permite analizar una situación, con base en los valores verdadero y falso                  |

- Óvalo: tiene por función abrir o cerrar el diagrama.
- Rectángulo: representa actividad/Proceso.
- Romboide: representa entrada de datos/salida de información.
- Rombo: representa un proceso de decisión en forma de pregunta.
- Círculo: representa conexiones y enlaces.
- Triángulo boca arriba: representa un archivo temporal.
- Triángulo boca abajo: representa un archivo definitivo.

| SÍMBOLO                                                                           | FUNCIÓN                            |
|-----------------------------------------------------------------------------------|------------------------------------|
|  | ABRE Y CIERRA UN DIAGRAMA          |
|  | ENTRADA DE DATOS                   |
|  | EJECUTA UNA OPERACIÓN O UN PROCESO |
|  | SALIDA DE DATOS                    |
|  | DECISIÓN                           |
|  | DIRECCIÓN                          |

### 4.3.2 HERRAMIENTAS

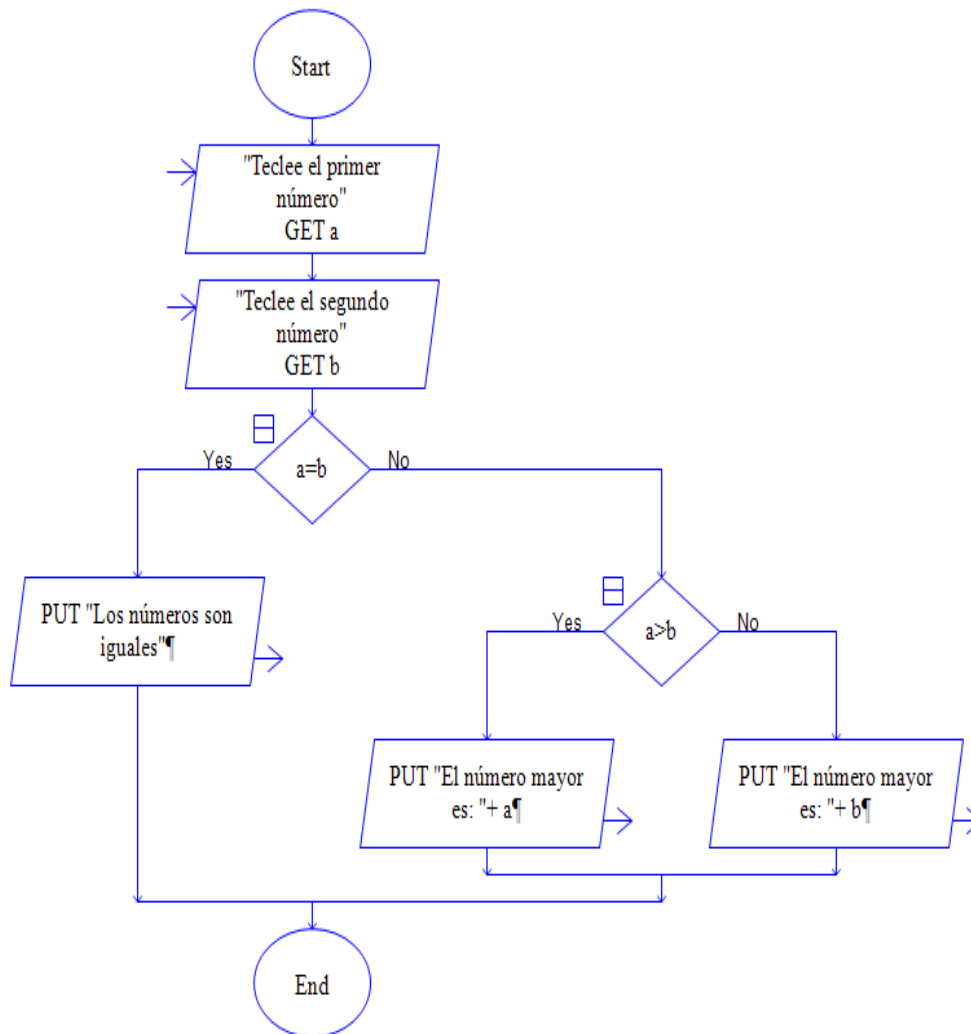
Las herramientas para diseñar diagramas de flujo son:

- **Iteraciones:** También conocidos como ciclos, son aquellos procesos de un diagrama, donde el flujo lógico retorna a ellos a través del cumplimiento de ciertas condiciones.
- **Transferencias de control:** Son muy útiles porque permiten alterar el curso de acción dentro de un diagrama; para ello se utiliza el símbolo de decisión (rombo). Existen tres tipos de transferencia de control:
  - **Por comparación:** La transferencia de control se realiza a través de la decisión de comparar una variable contra una constante o contra otra variable.
  - **Por comparación de contadores:** La transferencia de control se realiza a través de la decisión de comparar una variable contra una constante.
  - **Por comparación de banderas:** La transferencia de control se realiza a través de la decisión de comparar una bandera. Esta bandera es una variable que sólo puede tomar los valores de cierto o falso (1-0), por lo general una bandera cambia de valor cuando se cumple una o varias condiciones.
- **Índices:** Son muy útiles porque permiten diseñar diagramas en los cuales se van a emplear estructuras de datos como listas, colas, matrices, etc.

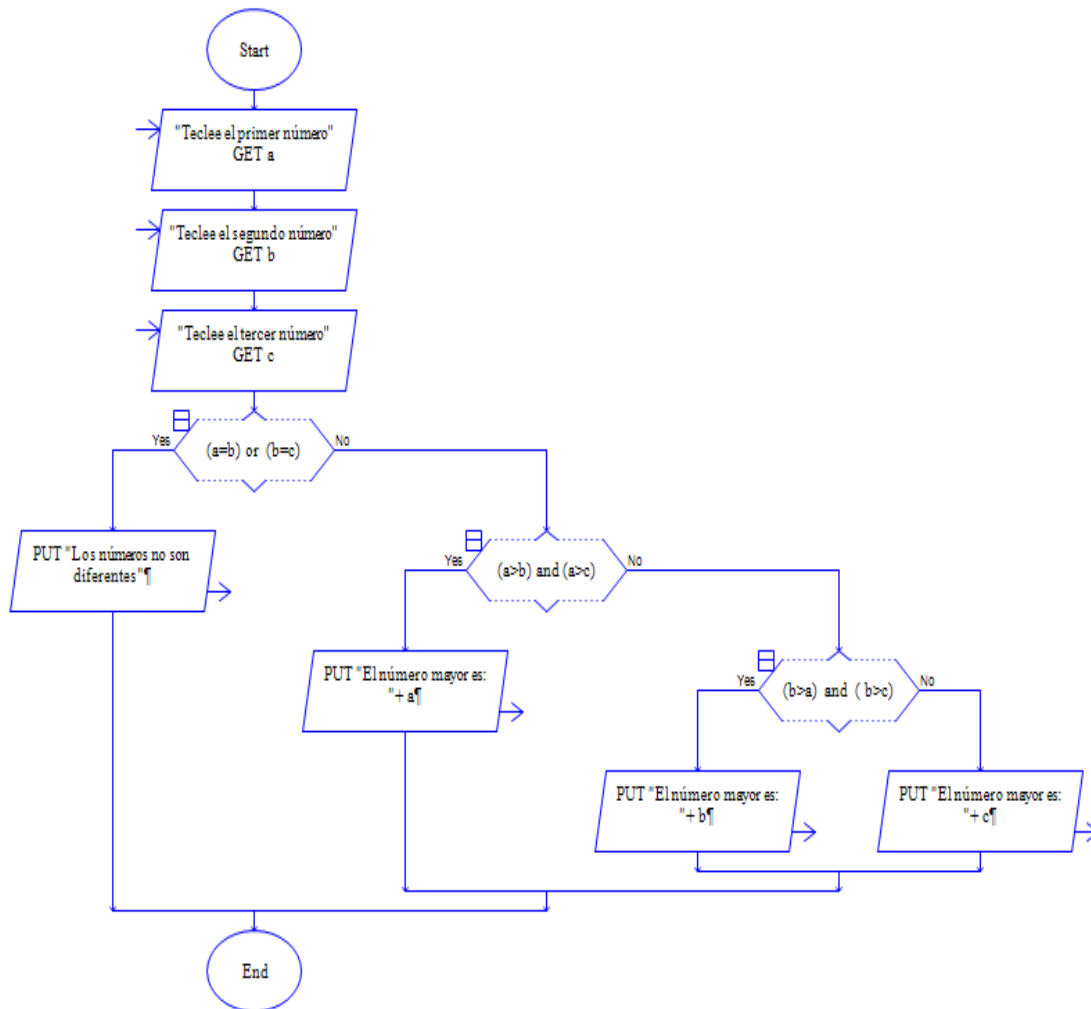
### 4.3.3 EJEMPLOS

1. Elabore un algoritmo que permita leer dos valores distintos, en base a valores, determinar cuál de los dos es el mayor y escribirlo.

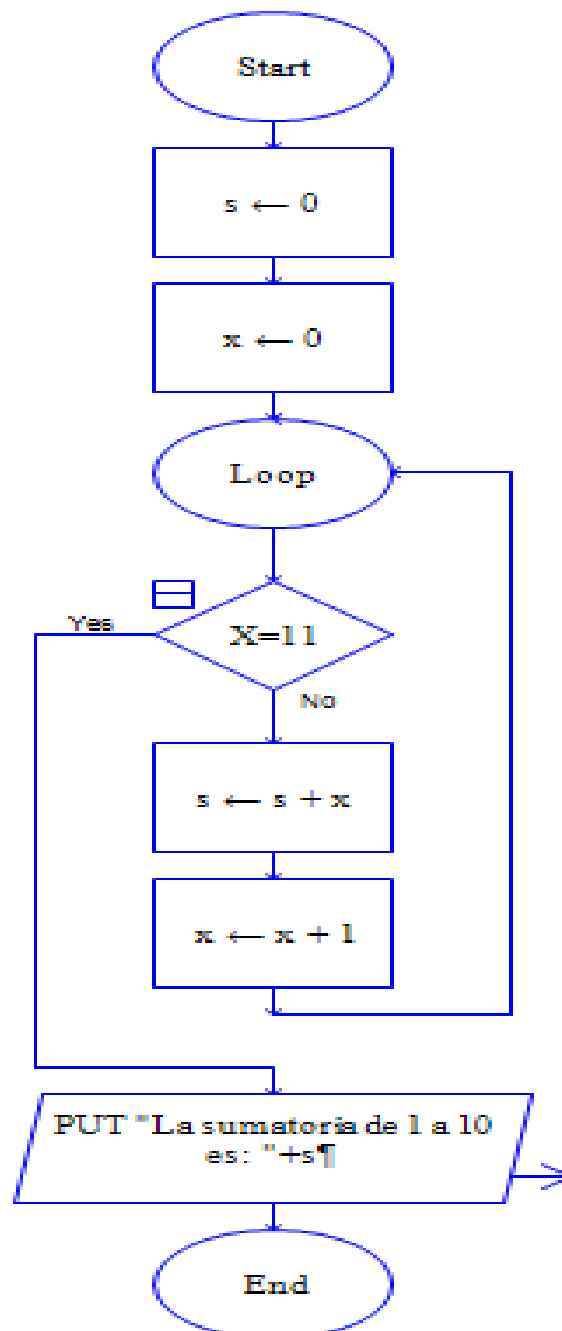
#### Diagrama de Flujo



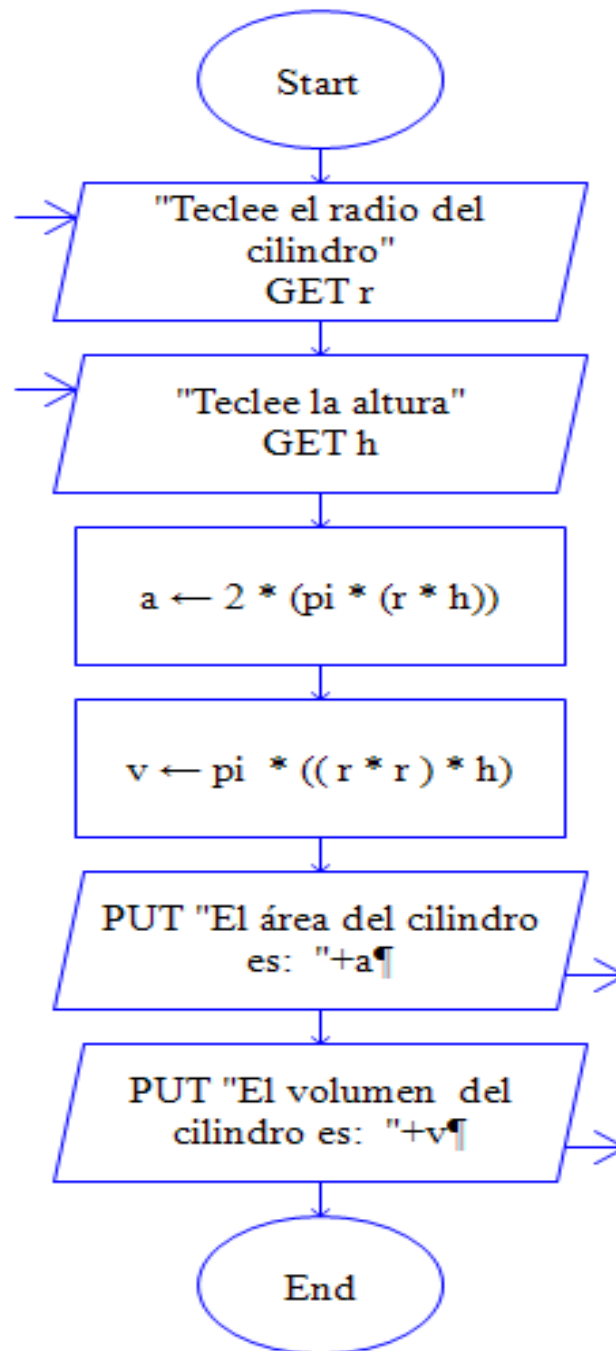
- Realizar un algoritmo que permita leer tres valores distintos y almacenarlos en las variables A, B, y C respectivamente. El algoritmo debe imprimir cuál es el mayor.



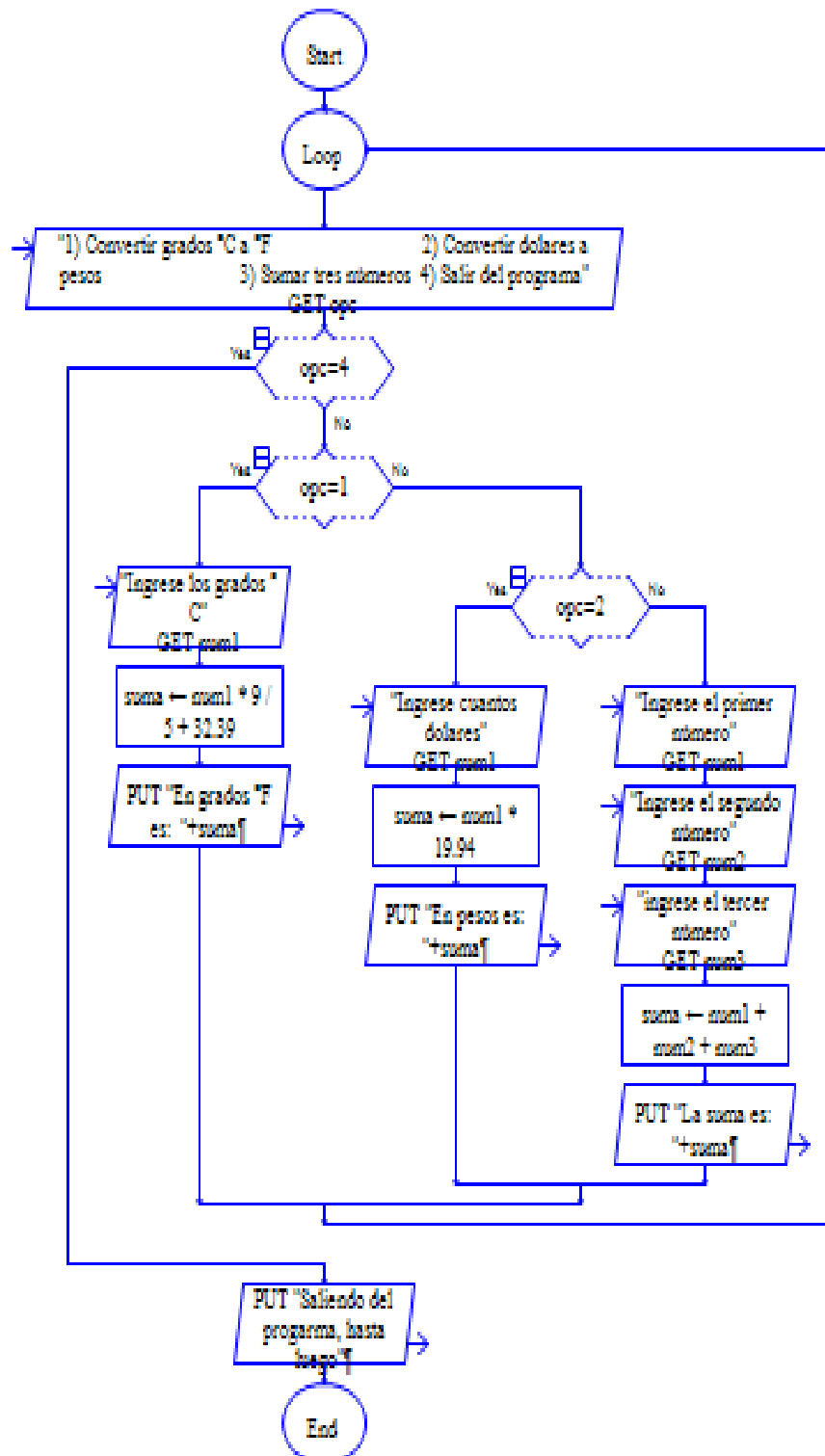
3. Elabore un algoritmo que realice la sumatoria de los números enteros comprendidos entre el 1 y el 10, es decir ,  $1+2+3+4+5+.....+10$ .



4. Desarrolle un algoritmo que permita determinar el área y volumen de un cilindro dado su radio (R) y altura (H).



## 5. Uso de ciclos





## 4.3.4 EJERCICIOS

### ACTIVIDAD

Resolver los siguientes algoritmos propuestos utilizando diagramas de flujo en el entorno de desarrollo integrado RAPTOR

Ejercicios:

- a) Realizar un algoritmo permita leer dos valores numéricos enteros, determinar el menor y escribirlo.
- b) Realizar un algoritmo que sume dos números enteros positivos e imprima la suma en una tercera variable.
- c) Escribir un algoritmo para indicar si tres números enteros positivos leídos son iguales.
- d) Escribir un algoritmo para indicar cuál es el menor de tres números leídos.
- e) Modificar el algoritmo anterior para que imprima la suma de los tres valores
- F) Escribir un algoritmo que realice la sumatoria de los números enteros comprendidos entre el 1 y el 10, es decir,  $1+2+3+4+5+.....+10$ . Sin considerar el número 10. Al final imprimir la suma.