

1. Algoritmo, Pseudocódigo, Diagramas de flujo.

Algoritmo: es un método para resolver un problema mediante una serie de pasos definidos, precisos y finitos.

En matemáticas, ciencias de la computación y disciplinas relacionadas, un **algoritmo** (del latín, *dixit algorithmus* y éste a su vez del matemático persa kenvorrio) es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. Los algoritmos son objeto de estudio de la **algoritmia**.

En la vida cotidiana se emplean algoritmos en multitud de ocasiones para resolver diversos problemas. Algunos ejemplos se encuentran en los instructivos (manuales de usuario), los cuales muestran algoritmos para usar el aparato en cuestión o inclusive en las instrucciones que recibe un trabajador por parte de su patrón. También existen ejemplos de índole matemática, como el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para calcular el máximo común divisor de dos enteros positivos, o el método de Gauss para resolver un Sistema lineal de ecuaciones.

Características

- Preciso: implica el orden de realización de cada uno de los pasos
- Definido: si se sigue dos veces, se obtiene el mismo resultado.
- Finito: Tiene un número determinado de pasos, implica que tiene un fin.
- No Ambiguo

Tipos :

Método	Descripción	Ejemplos
Algorítmico	Utiliza un algoritmo y puede ser implementado en una computadora	<ul style="list-style-type: none">• Instrucciones para manejar un vehículo• Instrucciones para secar grano a granel• Instrucciones para resolver ecuación de segundo grado
Heurística:	Se apoya en el resultado obtenido en un análisis de alternativas de experiencias anteriores similares. De las mismas, a se deducen una serie de reglas empíricas o heurísticas que de ser seguidas, conducen a la selección de la mejor alternativa en todas o la mayoría de las veces.	

		Ejemplos
Los algoritmos se pueden expresar por:	Formulas	$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
	Diagramas de flujo Norte-Sur, Top-Down	<pre> // Diagrama de flujo que evalua la edad Inicio edad <- entera "cuanto años tienes?" edad Edad < 50 F "Eres Viejo" V "Eres Joven" Fin </pre>
	Pseudo código	<pre> inicio leer a,b,c calcular <i>perimetro</i> = a + b + c escribir perimetro fin </pre>

Pasos para resolver un problema utilizando una computadora.

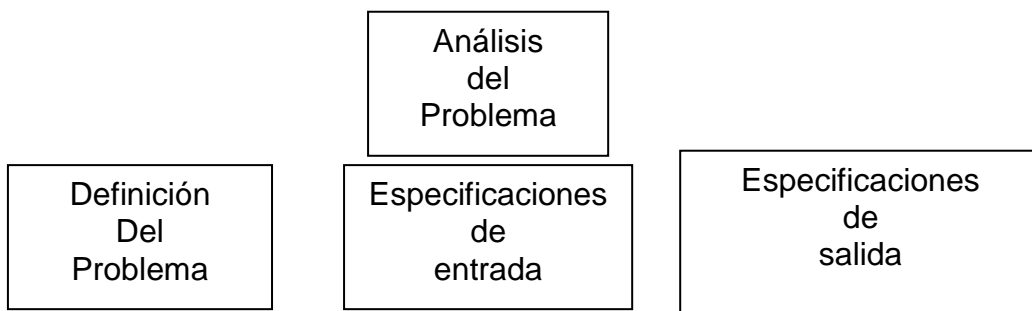
Pasos	Etapa	Descripción
A	Análisis del problema	Conducen al diseño detallado por medio un código escrito en forma de un algoritmo
B	Diseño de algoritmo	
C	Codificación	Se implementa el algoritmo en un código escrito en un lenguaje de programación. Refleja las ideas desarrolladas en las etapas de análisis y diseño
D	Compilación y ejecución	Traduce el programa fuente a programa en código de maquina y lo ejecuta.
E	Verificación	Busca errores en las etapas anteriores y los elimina.
F	Depuración	
G	Documentación	Son comentarios, etiquetas de texto, que facilitan la comprensión del programa

A. Análisis del problema:

Requiere la clara definición del problema donde se indique que va hacer el programa y cual va a ser el resultado.

Debe detallarse las especificaciones de entrada y salida,

Los requisitos que definen el análisis son :



Esta parte es para darnos cuenta de la manera como esta expresado el problema, ¿Es fácil de entender cuando se lee? ¿Quedan claros los requerimientos? ¿Esta delimitado de forma adecuada?, todo esto debe quedar absolutamente claro para poder empezar con el siguiente paso.

Determinando un conjunto de preguntas abordaremos aun más el problema :

¿Con que cuento?

¿Que hago con los datos?

¿Qué espero obtener?

B.Diseño del algoritmo.

Se hace un bosquejo de la posible solución, considerando los elementos del análisis, esto implica :

- Como se hace o realiza la tarea (o problema) solicitado
- El todo es la sumatoria de las partes.
- Divide el todo en varias partes.

En la resolución de un problema complejo, se divide en varios sub problemas y seguidamente se vuelven a dividir los sub problemas en otros mas sencillos, hasta que puedan implementarse en el computador.

Esta característica define lo que se entiende como *diseño descendente*(Top-Down / Norte-Sur) o *diseño modular*.

El proceso de ruptura del problema en cada etapa se llama *refinamiento sucesivo*.

- Cada problema se resuelve mediante un *modulo (subprograma)* y tiene un solo punto de entrada y un solo punto de salida.
- Un programa bien diseñado consta de un *programa principal* (modulo de nivel mas alto) que llama a *subprogramas* (módulos de nivel mas bajo), que a su vez pueden llamar otros sub programas.

Los programas que se estructuran de esta forma, se dicen que tienen *diseño modular* y el método de romper el programa en modos pequeños se llama *programación modular*.

Los módulos pueden ser planificados, codificados, compilados y depurados independientemente pueden ser intercambiados entre si.

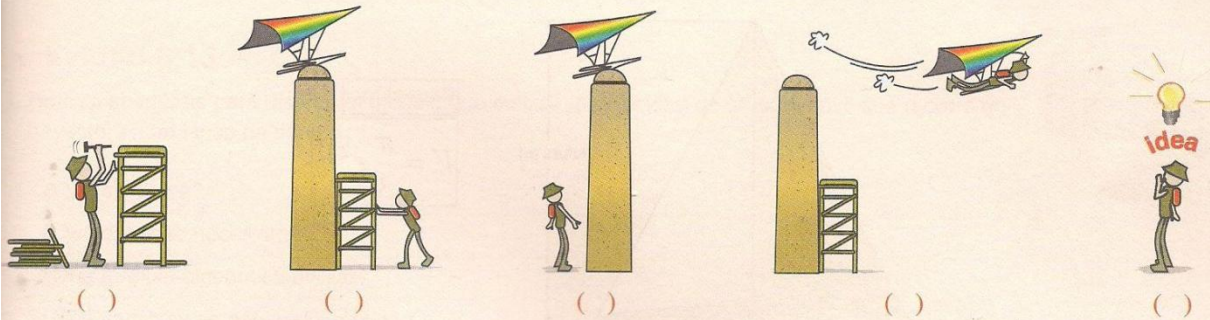
Este proceso implica la ejecución de los siguientes pasos:

1	programar un modulo
2	comprobar un modulo
3	depurar el modulo
4	combinar el modulo con módulos anteriores

este proceso convierte el resultado del análisis del problema en un diseño modular con refinamientos sucesivos que permiten una traducción a un lenguaje que se denomina diseño del algoritmo.

Cómo enfrentar un reto.

1. Analiza los dibujos y relaciona cada viñeta con las palabras de abajo.



1. Identifica el problema (reto)
2. Analiza el problema
3. Diseña la solución
4. Implementa la solución
5. Logra tu objetivo

Ejercicio que muestra las diversas formas de interpretar un problema, se pretende ser objetivos en la solución.

El algoritmo se puede *representar* por medio de dos formas :

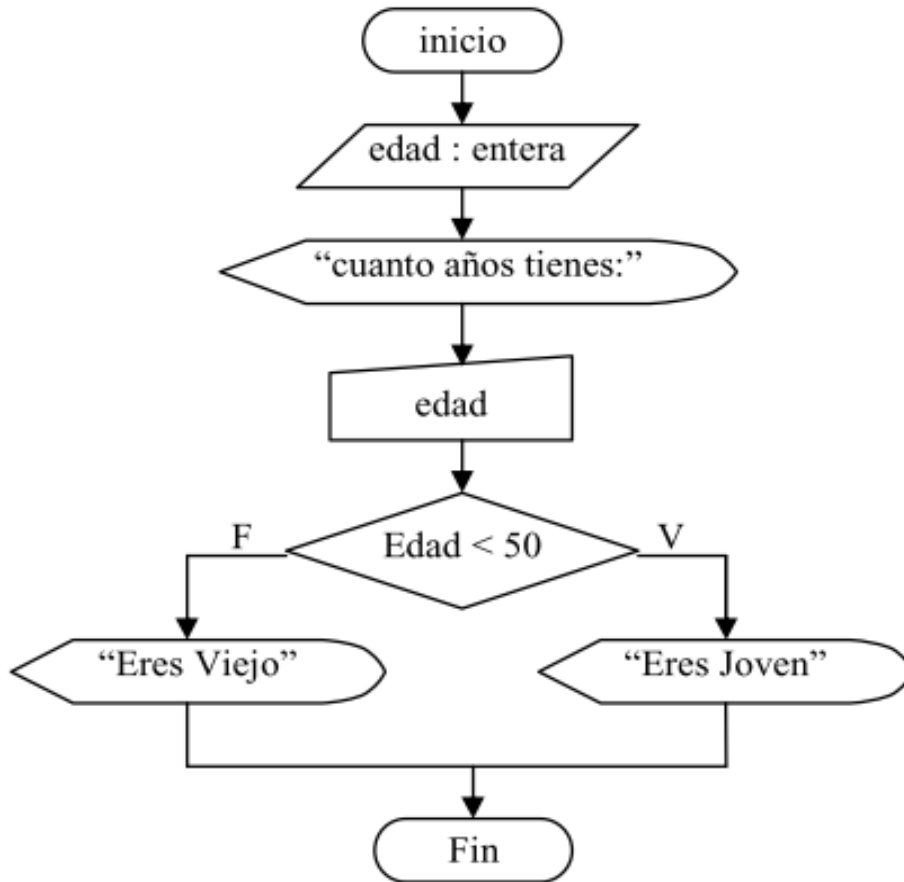
- **Pseudo código:** es el lenguaje de especificación de algoritmos y tiene una estructura: Las instrucciones se escriben en ingles o en palabras similares al ingles o español que facilitan la escritura de programación.

Ejemplo :


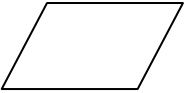
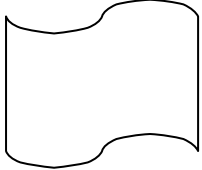

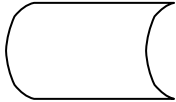
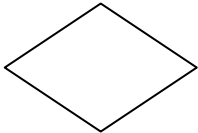
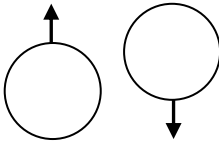
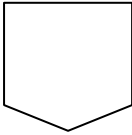

1. Inicio
2. Escribir "Da tu calificación"
3. Leer Cal
4. Si $Cal > 5$ entonces
 Escribir "Aprobado" si no
 Escribir "Reprobado"
- Fin Si
5. Fin

- **Diagramas de flujo (flows charts):** Es la representación grafica del algoritmo. Esta representación se da cuando varios símbolos (que indican diferentes procesos en la computadora), se relacionan entre si mediante líneas que indican el orden en que se deben ejecutar los procesos. Según la ANSI consta de una simbología, que tiene diversos significados. Ejemplo

// Diagrama de flujo que evalua la edad



3. Símbolos de diagramas de flujo

Nombre	Símbolo	Función
Proceso		Cualquier tipo de operación que origine cambio de valor, formato, posición de la información almacenada en memoria, operaciones aritméticas, de transferencia, etc.
Entrada/Salida		Cualquier tipo de introducción de datos en la memoria desde los periféricos "entrada" o registros de información procesada en un periférico de salida
Impresora		Se utiliza en ocasiones en lugar del símbolo de entrada/salida
Subrutina		Llamada a subrutina, función o procedimiento; este es un módulo independiente del programa principal, que recibe una entrada procedente de dicho programa, realiza una tarea determinada y regresa al terminar al programa principal
Monitor		Pantalla; en ocasiones se utiliza en lugar de símbolo de entrada / salida
Proceso: Decisión		Indica operaciones lógicas o de comparación entre datos - normalmente entre dos y en función del resultado, determina cuál de los dos caminos alternativos se debe seguir; normalmente tiene dos salidas - respuestas si o no.
Conector		Sirve para enlazar dos partes cualesquiera de un organigrama a través de un conector de salida y otro en la entrada. En la misma página del diagrama
Conector		Conexión entre dos puntos del organigrama situados en páginas diferentes
Teclado		En ocasiones se utiliza en lugar del símbolo de entrada/salida

4. Operaciones con los datos

La importancia de este tema es tan grande debido a que todo el sistema de información realiza cálculos con datos para entregar resultados a la empresa, por lo cual debemos saber que los datos que maneja la empresa solamente pueden ser números, letras y una respuesta afirmativa; y los cálculos que el sistema puede realizar sobre estos datos son operaciones como suma, resta, multiplicación y división, además de comparaciones entre los datos para saber si uno es mayor que otro, si es menor, si son iguales o diferentes, y establecer un grado de satisfacción entre dos datos en base a las tablas de la verdad (AND, OR Y NOT).

Sabiendo todo lo anterior, debemos aprender a expresar los cálculos a realizar por el sistema de una manera que la computadora pueda comprenderlos y arrojar los resultados correctos mediante una expresión o fórmula que se rige por un conjunto de reglas.

Además de que debemos aprender a crear los espacios temporales de almacenamiento donde se guardaran tanto los datos como los resultados.

4.1 Tipos de datos simples.

DATO.- “es una pequeña parte de la información que por sí sola no dice nada, pero que en conjunto forma información”

INFORMACION.- “es un conjunto de datos estructurados o procesados”

Clasificación de los datos

Simples	➤ Numérico	▪ Entero ▪ Reales
	➤ Lógicos	
	➤ Alfanuméricos	
Complejos	➤ Arreglos	▪ Unidimensionales ▪ Multidimensionales
	➤ Estructuras	

A) Datos simples

- **Datos numéricos.**- permiten representar valores escalares de forma numérica, esto incluye a los números enteros y los reales.

- **Enteros.**- son los números que no tienen parte decimal, pueden ser positivos o negativos, por ejemplo: 10, 0, 1358, -456.
- **Reales.**- son los números que contienen una fracción, es decir, punto decimal y estos al igual que los enteros pueden ser positivos o negativos, por ejemplo: 12.45, 7.0 -157.0001.
- **Datos lógicos.**- son aquellos que solo pueden tener uno de los dos valores posibles (cierto o falso) ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).
- **Datos alfanuméricos.**- es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombres de personas, direcciones, et. Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir no es posible hacer operaciones con aquellos. Este tipo de datos se representan encerrados entre comillas.

4.2 Tipos de operadores

Debemos de aprender a utilizar los datos y operadores, pues somos nosotros quien le indicara a la computadora los cálculos a realizar a ciertos datos.

Al conjunto de todos los operadores, los podemos dividir en tres grupos:

- A) Operadores aritméticos
- B) Operadores relacionales
- C) Operadores lógicos

A) Operadores aritméticos.- son aquellos con los que podemos realizar operaciones como suma, resta, multiplicación, división, modulo y asignación.


Las expresiones aritméticas se deben escribir en una línea continua y bajo unas **reglas de precedencia de operadores**. Las cuales son guías de acción que permiten calcular las expresiones en el orden correcto.

- 1- Se calculan primero las operaciones de multiplicación, división y modulo, los cuales tienen el mismo nivel de precedencia, por lo cual si existen varios de estos en una expresión se comienzan a calcular de izquierda a derecha.
- 2- Se calculan las operaciones de suma y de resta, los cuales tienen el mismo nivel de precedencia. Si la expresión la expresión contiene varias de esta se realiza de izquierda a derecha.
- 3- Si en la expresión se encuentran paréntesis, esto indica que lo que está dentro de ellos se debe resolver antes que cualquier cosa siguiendo las reglas de precedencia antes mencionados, por lo cual los paréntesis son utilizados para

obligar a la computadora a evaluar primero ciertas expresiones. En caso de existir paréntesis anidados se evalúa el par mas interno.

- 4- Por último se realiza la asignación, la cual significa que el valor de la derecha es obligado al **identificador** de la izquierda.

PRECEDENCIA DE LOS OPERADORES ARITMETICOS

OPERADOR	PRECEDENCIA
()	Mayor  Menor
*, /, %	
+, -	
=	

Ejemplo 1 : $y = 2 * 5 * 5 + 3 * 5 + 7$

1. Realiza la multiplicación más a la izquierda $y = 2 * 5 * 5 + 3 * 5 + 7$ $y = 10 * 5 + 3 * 5 + 7$
2. Realiza la multiplicación más a la izquierda $y = 10 * 5 + 3 * 5 + 7$ $y = 50 + 3 * 5 + 7$
3. Realiza la multiplicación más a la izquierda $y = 50 + 3 * 5 + 7$ $y = 50 + 15 + 7$
4. Realiza suma más a la izquierda $y = 50 + 15 + 7$ $y = 65 + 7$
5. Realiza la suma $y = 65 + 7$ $y = 72$

Ejemplo 2 : $Z = 4 * ((2 + 6) * (8 - 10))$

a. Realiza el paréntesis más interno de la izquierda $Z = 4 * ((2 + 6) * (8 - 10))$ $Z = 4 * (8 * (8 - 10))$
2. Realiza el paréntesis más interno $Z = 4 * (8 * (8 - 10))$ $Z = 4 * (8 * -2)$
3. Realiza el paréntesis $Z = 4 * (8 * -2)$ $Z = 4 * -16$
4. Realiza la multiplicación $Z = 4 * -16$ $Z = -64$

Ejercicios:

- 1.- Resuelve las siguientes operaciones utilizando las reglas de precedencia, donde:

$$W=5, X=7, Y=3, Z=9$$

$$A = Y - Z * X + W / 3$$

$$A = Z + W \% Y$$

$$A = X * (Z - Y) / W$$

$$A = (4 * Y + Z \% W) * X$$

$$A = Z * W - X + Y / Z$$

2.- Expresa las siguientes formulas para que las entienda la computadora:

- Calcular el perímetro de un círculo
- calcular el área de un rectángulo
- calcular el área de un círculo
- $X = Z^3$
- $X = 4 AC + (A + C) - 2 AB$
- $X = 5 (Y^2 + Z^3 - 5 ZW + 3)$


Recordar que dos números enteros dan como resultado un valor entero y un entero y un real dan como resultado un valor real.

B) Operadores relacionales.- Los operadores relacionales se usan para determinar la relación de la expresión de la izquierda con la de la derecha (binarios). El resultado de esta evaluación regresa el valor de falso o verdadero.

Conjunto de operadores relacionales

OPERADOR	RELACION
= =	Igual
! =	Diferente
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

Precedencia de los operadores aritméticos y relacionales

OPERADOR	PRECEDENCIA
()	Mayor  Menor
*, /, %	
+, -	
<, >, <=, >=	
==, !=	
=	

Ejemplo 1.-

Expresión $Z = 4 \leq 2 == 6 != 8 > 10$ Resultado : $z = 0$

Ejemplo 2.-

Expresión $Z = 8 == (9 + (1 != 0)) > 3 * 5$ Resultado : $z = 0$

Ejercicios: Realiza las siguientes operaciones siguiendo las reglas de precedencia, donde :

W = 3, x = 5, Y = 7 , Z = 9

- 1) $A = X == Z$
- 2) $A = W >= Y$
- 3) $A = W == X < Y < Z$
- 4) $A = (W == X) == (Y > Z)$
- 5) $A = X \text{ } i = (W < Z < Y) == 1$
- 6) $A = W * Y >= W * Z$
- 7) $A = Y + W * Z / W \text{ } i = Z + W - Y * X$
- 8) $(Y + W) * Z / W == Y * X - 20 / 4$
- 9) $W * Y >= W * Z == (Y + W) * Z > 0$
- 10) $X > Z * (W + Y) \text{ } i = W <= X$

C) Operadores Lógicos : Los operadores lógicos, se usan para soportar las operaciones básicas lógicas AND, OR y NOT. Las tablas se describen a continuación :

$$2^2 = 4$$


a	b	a and b
0	0	0
0	1	0
1	0	0
1	1	1

A	b	a or b
0	0	0
0	1	1
1	0	1
1	1	1

a	Not
0	1
1	0

Los operadores lógicos No están al mismo nivel de precedencia entre ellos. El operador NOT es el de mayor, posteriormente se encuentra el AND y por último el OR.

Tabla de precedencia de todos los operadores.

Operador	Precedencia
()	Mayor  Menor
NOT i	
* , /, % (residuo)	
+ , -	
< , > , <= , >=	
== , i=	
AND &&	
OR	
=	

5. Identificadores

Para que la computadora procese los datos se pueden guardar temporalmente en una pequeña parte de la memoria de la computadora, a este espacio se le debe decir que tipo de datos puede almacenar (enteros, reales, alfanuméricos, etc.) y como queremos que se le llame para poder localizarlo posteriormente. A este espacio de memoria con un nombre y tipo específico, se le conoce como **identificador**.

Si nosotros no creamos un identificador, el dato que deseamos guardar se almacenaría en una posición de memoria la cual esta identificada por un número hexadecimal, y para recuperarla tendríamos que saber esta dirección, por lo cual es más fácil asignarle un nombre. Además, si nosotros no le indicamos un tipo para los datos que se van a almacenar, la computadora no sabrá como tratar a esta información, recordemos que en la computadora solo están almacenados ceros y unos.

Los identificadores se dividen en dos:

- **Constantes.** Es aquel en el cual, el dato que tiene dentro es el mismo desde que comienza el programa hasta que termina, y bajo ninguna circunstancia ni procedimiento puede cambiar. Por ejemplo: Pi, ya que siempre es 3.1416.
- **Variables.** Es aquel en el cual, el dato que tiene dentro puede cambiar todas las veces necesarias por otro en cualquier parte del programa siempre y cuando sean del tipo especificado anteriormente. Por ejemplo: edad, ya que puede almacenar en determinado momento mi edad, en otro la tuya, etc. A su vez, las variables se pueden clasificar por su uso en:

- **Variables de Trabajo:** Son aquellas que reciben el resultado de una operación matemática compleja y que se usan normalmente dentro de un programa, pero si es del tipo alfanumérico solo se utiliza para almacenar información.
Ejemplo: promedio = $(9 + 8 + 7) / 3$
- **Contadores:** Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno. Podríamos utilizarlos cuando necesitamos llevar el conteo del número de personas que votaron por el PAN. Son exclusivamente del tipo entero.
- **Acumuladores:** Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente. Una variable de este tipo podríamos utilizarla para ir sumando poco a poco el monto total de nuestra compra en un supermercado.
- **Variable indicador o de bandera:** Es aquella que recibe uno de dos posibles valores. Se les conoce también como BANDERAS y generalmente son del tipo **booleano**.

Reglas para formar un identificador

1. Debe comenzar con una letra (A-Z, mayúsculas o minúsculas)
2. No deben contener espacios en blanco.
3. Dígitos y caracteres especiales están permitidos después del primer carácter.
4. La longitud de identificadores puede ser de hasta 256 caracteres.
5. El nombre del identificador debe ser significativo.
6. Indicar su tipo (entero, real, alfanumérico, booleano).
7. Si se desea, se puede indicar su uso, el cual como ya sabemos solo es para las variables.
8. Si se desea, asignarles un valor de inicio. En los constantes es forzoso este punto.