



## SEMANA UNO

### INTRODUCCIÓN A LA PROGRAMACIÓN

#### Contenidos

1. Conceptos generales de programación
  - 1.1. Que es un computador
    - 1.1.2 El Software
    - 1.2.1 El Hardware
2. Clasificación de las computadores
3. Organización física de la computadora
  - 3.1 Dispositivos de Entrada/ Salida
  - 3.2 Memoria principal
  - 3.3 Unidad Central de Proceso (UPC)
  - 3.4 Memoria auxiliar externa
4. Algoritmos y programas
  - 4.1 Algoritmos
    - 4.1.1. Características de un algoritmo
  - 4.2 Programas
  - 4.3 Clasificación de los lenguajes de programación
5. Herramientas de programación
  - 5.1 Diagramas de Flujo
    - 5.1.1 Importancia de los diagramas de flujo
    - 5.1.2 Reglas para dibujar diagramas
6. Los Datos
  - 6.1 Datos simples
  - 6.2 Identificadores, Constantes y variables
  - 6.3 Constantes
  - 6.4 Variables
    - 6.4.1 Clasificación de las variables
  - 6.5 Expresión Algebraica
  - 6.6 Jerarquía de las operaciones

#### Conceptos generales de programación



Antes de iniciar el tema, es conveniente, hacer un breve repaso de conceptos fundamentales que se han de tomar en cuenta cuando se desarrollan programas que se utilizarán en computadoras. Se incluye la metodología que se requiere para diseñar algoritmos de manera sencilla y funcional.

**1.1 ¿Qué es una computadora?:** Es un dispositivo electrónico utilizado para procesar información y obtener resultados. Podemos hacer trabajos de oficina con ella, guardar datos, imágenes, escribir cartas, leer el periódico, comunicarnos con familiares o amigos a través de correos electrónicos, ver videos, dibujar, hacer informes, crear programas de computadoras que llevan a cabo diversas funciones e incluso nos permite hacer presentaciones que pueden ver otros usuarios de computadoras alrededor del mundo, el hecho de que usted este leyendo este material, es evidencia de ello.

Los datos y la información se pueden introducir en la computadora por la entrada y a continuación se procesan para producir una salida (resultados). Los datos de entrada y los datos de salida pueden ser, realmente, cualquier cosa, texto, dibujos, o sonidos.

Hay dos partes básicas que explicar para entender la computadora, estas partes son: el software y el hardware.

**1.1.1 El software:** Conjunto de programas escritos para una computadora, es un término genérico para los programas que funcionan en el interior de una computadora. En este caso posiblemente sea Windows el sistema operativo o programa de funcionamiento que le da la vida a su computadora, es así como usted puede ver ahora mismo esta información en su pantalla.

**1.1.2 El hardware:** Componentes físicos que constituyen la computadora, junto con los dispositivos que realizan las tareas de entrada y salida. corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente al soporte lógico e intangible que es llamado *software*. El término es propio del idioma inglés (literalmente traducido: partes duras), su traducción al español no tiene un significado acorde, por tal motivo se la ha adoptado tal cual es y suena; la Real Academia Española lo define como «Conjunto de los componentes que integran la parte material de una computadora». El término, aunque es lo más común, no necesariamente se aplica a una computadora tal como se la conoce, así por ejemplo, un robot también posee *hardware* (y *software*).

## 1. Clasificación de las computadoras

Las computadoras modernas se pueden clasificar en computadoras personales, servidores, minicomputadoras, grandes computadoras (mainframes) y supercomputadoras.

Las **computadoras personales (PC)** son los más populares y abarcan desde computadoras portátiles (laptops o notebooks, en inglés) hasta computadoras de escritorio (desktop) que se suelen utilizar como herramientas en los puestos de trabajo, en oficinas, laboratorios de enseñanza e investigación, empresas, etc. Los **servidores** son computadoras personales profesionales y de gran potencia que se utilizan para gestionar y administrar las redes internas de las empresas o departamentos y muy especialmente para administrar sitios Web de Internet. Las computadoras tipo servidor son optimizadas específicamente para soportar una red de computadoras, facilitar a los usuarios la compartición de archivos, de *software* o de periféricos como impresoras y otros recursos de red. Los servidores tienen memorias grandes, altas capacidades de memoria en disco e incluso unidades de almacenamiento masivo como unidades de cinta magnética u ópticas, así como capacidades de comunicaciones de alta velocidad y potentes CPUS, normalmente específicas para sus cometidos.

## 2. Organización física de la computadora

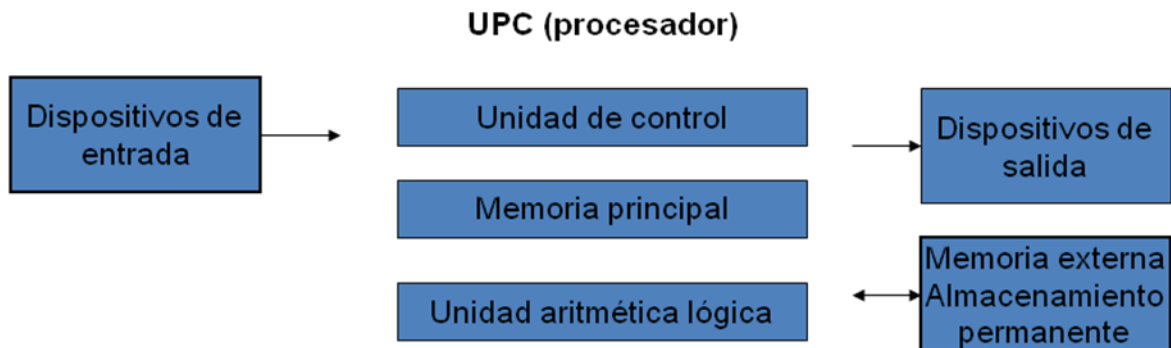


Imagen1. Entrada, proceso salida

### 3.1 Dispositivos de Entrada/Salida



Los dispositivos de Entrada/Salida permiten la comunicación entre la computadora y el usuario para introducir datos (información) para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria principal, convierten la información de entrada en señales eléctricas que se almacenan en la memoria principal. Los dispositivos de Entrada/Salida sirven a la computadora para obtener información del mundo exterior y/o comunicar los resultados generados por el computador al exterior. Hay una gama muy extensa de dispositivos E/S como teclados, monitores, unidades de disco flexible o cámaras web. Los dispositivos de *Entrada/Salida* (E/S) [*Input/Output (I/O)* en inglés] permiten la comunicación entre la computadora y el usuario. Los *dispositivos de entrada*, como su nombre indica, sirven para introducir datos (información) en la computadora para su proceso. Los datos se *leen* de los dispositivos de entrada y se almacenan en la memoria central o interna. Los dispositivos de entrada convierten la información de entrada en señales eléctricas que se almacenan en la memoria central. Dispositivos de entrada típicos son los teclados; otros son: lectores de tarjetas ya en desuso—, lápices ópticos, palancas de mando (*joystick*), lectores de códigos de barras, escáneres, micrófonos, etc. Hoy día tal vez el dispositivo de entrada más popular es el ratón (*mouse*) que mueve un puntero electrónico sobre la pantalla que facilita la interacción usuario-máquina.

Los dispositivos de salida permiten representar los resultados (salida) del proceso de los datos. El dispositivo de salida típico es la pantalla (CRT) o monitor. Otros dispositivos de salida son: impresoras (imprimen resultados en papel), trazadores gráficos (*plotters*), reconocedores de voz, altavoces, etc.

El teclado y la pantalla constituyen en muchas ocasiones un único dispositivo, denominado **terminal**. Un teclado de terminal es similar al teclado de una máquina de escribir moderna con la diferencia de algunas teclas extras que tiene el terminal para funciones especiales. Si está utilizando una computadora personal, el teclado y el monitor son dispositivos independientes conectados a la computadora por cables. En ocasiones, la impresora se conoce como **dispositivo de copia dura** (*hard copy*), debido a que la escritura en la impresora es una copia permanente (dura) de la salida, y en contraste a la pantalla se la denomina **dispositivo de copia blanda** (*soft copy*). Los dispositivos de entrada/salida y los dispositivos de almacenamiento secundario o auxiliar (memoria externa) se conocen también con el nombre de *dispositivos periféricos* o simplemente **periféricos** ya que, normalmente, son externos a la computadora. Estos dispositivos son unidades de discos (disquetes (ya en desuso), CD-ROM, DVD, USB, videocámaras, teléfonos celulares (móviles), etc.



### 3.2 Memoria principal

Conocida como RAM (Random Access Memory), almacena información (instrucciones de un programa y los datos con los que operan las instrucciones) de manera transitoria, es decir, durante la ejecución de un programa.

La **memoria principal** o **primaria** (MP), también llamada **memoria central**, es una unidad dividida en celdas que se identifican mediante una dirección. Está formada por bloques de circuitos integrados o chips capaces de almacenar, retener o "memorizar" información digital, es decir, valores binarios; a dichos bloques tiene acceso el microprocesador de la computadora. La MP se comunica con el microprocesador de la CPU mediante el bus de direcciones. El ancho de este bus determina la capacidad que posea el microprocesador para el direccionamiento de direcciones en memoria. En algunas oportunidades suele llamarse "memoria interna" a la MP, porque a diferencia de los dispositivos de memoria secundaria, la MP no puede extraerse tan fácilmente por usuarios no técnicos. La MP es el núcleo del sub-sistema de memoria de un computador, y posee una menor capacidad de almacenamiento que la memoria secundaria, pero una velocidad millones de veces superior.

En las computadoras son utilizados dos tipos:

1. ROM o **memoria de sólo lectura** (*Read Only Memory*). Viene grabada de fábrica con una serie de programas. El software de la ROM se divide en dos partes:

- Rutina de **arranque** o POST (*Power On Self Test*, auto diagnóstico de encendido): Realiza el chequeo de los componentes de la computadora; por ejemplo, circuitos controladores de video, de acceso a memoria, el teclado, unidades de disco, etc. Se encarga de determinar cuál es el hardware que está presente y de la puesta a punto de la computadora. Mediante un programa de configuración, el SETUP, lee una memoria llamada CMOS RAM (RAM de Semiconductor de óxido metálico). Ésta puede mantener su contenido durante varios años, aunque la computadora está apagada, con muy poca energía eléctrica suministrada por una batería, guarda la fecha, hora, la memoria disponible, capacidad de disco rígido, si tiene disquetera o no. Se encarga en el siguiente paso de realizar el arranque (*booteo*): lee un registro de arranque 'BR' (*Boot Record*) del disco duro o de otra unidad (como CD, USB, etc.), donde hay un programa que carga el sistema operativo a la RAM. A continuación cede el control a dicho sistema operativo y el computador queda listo para trabajar.





- Rutina BIOS o Sistema Básico de Entrada-Salida (*Basic Input-Output System*): permanece activa mientras se está usando el computador. Permite la activación de los periféricos de entrada/salida: teclado, monitor, ratón, etc.

2. RAM o **memoria de acceso aleatorio** (*Random Access Memory*). Es la memoria del usuario que contiene de forma temporal el programa, los datos y los resultados que están siendo usados por el usuario del computador. En general es volátil, pierde su contenido cuando se apaga el computador, es decir que mantiene los datos y resultados en tanto el bloque reciba alimentación eléctrica, a excepción de la CMOS RAM.

Tanto la RAM como la ROM son circuitos integrados, llamados comúnmente chips. El chip o circuito integrado es una pequeña pastilla de material semiconductor (silicio) que contiene múltiples circuitos integrados, tales como transistores, entre otros dispositivos electrónicos, con los que se realizan numerosas funciones en computadoras y dispositivos electrónicos; que permiten, interrumpen o aumentan el paso de la corriente. Estos chips están sobre una tarjeta o placa.

El contenido de las memorias no es otra cosa que dígitos binarios o bits (*binary digits*), que se corresponden con dos estados lógicos: el 0 (cero) sin carga eléctrica y el 1 (uno) con carga eléctrica. A cada uno de estos estados se le llama bit, que es la unidad mínima de almacenamiento de datos.

### 3.3 Unidad Central de Proceso (UPC)

La unidad central de proceso (**UCP** o **CPU**, **Central Process Unit**) es la encargada de interpretar ordenadamente las instrucciones almacenadas en la memoria para poder ser ejecutadas. La unidad central de proceso actúa como el cerebro de la unidad central multiusuario, y se encarga del control general y del envío de información a todos los demás elementos de la máquina (memoria principal y periféricos).

La unidad central de proceso está formada por la **Unidad de Control(UC)**, incluyendo los registros en los que se almacena temporalmente la información manejada por la unidad central de proceso y la **Unidad Operativa** o **Unidad Aritmético-Lógica (ALU)**.

### 3.4 Memoria auxiliar (externa)



**Memoria Auxiliar (Externa):** Es donde se almacenan todos los programas o datos que el usuario desee. El **almacenamiento secundario (memoria secundaria, memoria auxiliar o memoria externa)** es el conjunto de dispositivos (aparatos) y medios (soportes) de almacenamiento, que conforman el subsistema de memoria de una computadora, junto a la memoria principal. También llamado periférico de almacenamiento.

La **memoria secundaria** es un tipo de almacenamiento masivo y permanente (no volátil), a diferencia de la memoria RAM que es volátil; pero posee mayor capacidad de memoria que la memoria principal, aunque es más lenta que ésta. El proceso de transferencia de datos a un equipo de cómputo se le llama "procedimiento de lectura". El proceso de transferencia de datos desde la computadora hacia el almacenamiento se denomina "procedimiento de escritura".

En la actualidad para almacenar información se usan principalmente tres 'tecnologías':

1. Magnética (ej. disco duro, disquete, cintas magnéticas);
  1. Óptica (ej. CD, DVD, etc.) Algunos dispositivos combinan ambas tecnologías, es decir, son dispositivos de almacenamiento híbridos, por ej., discos Zip.
2. Memoria Flash (Tarjetas de Memorias Flash y Unidades de Estado sólido SSD)

#### 4 Algoritmos y programas

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 normalmente definido como la información necesaria para solucionar un problema.

El proceso de programación es, por consiguiente, un proceso de solución de problemas y el desarrollo de un programa requiere las siguientes fases:

1. Definición y análisis del problema.
2. Diseño de algoritmos. (diagrama de flujo, pseudocódigo)
3. Codificación del programa.
4. Depuración y verificación del programa.
5. Documentación.



## 6. Mantenimiento.

**4.1 Algoritmo:** Un algoritmo es un método para resolver un problema, debe presentarse como una secuencia ordenada de instrucciones que siempre se ejecutan en un tiempo finito y con una cantidad de esfuerzo también finito. En un algoritmo siempre debe haber un punto de inicio y un punto de terminación, estos deben ser únicos y deben ser fácilmente identificables.

### 4.1.2 CARACTERÍSTICAS DE UN ALGORITMO

Todo algoritmo debe cumplir las siguientes características:

- A. Debe ser *Preciso*; Esto es, debe especificar sin ambigüedad el orden en que se deben ejecutar las instrucciones.
- B. Debe estar *Definido*; Esto es, cada vez que se ejecute bajo las mismas condiciones, la secuencia de ejecución deberá ser la misma proporcionándonos el mismo resultado.
- C. Debe ser *Finito*; Esto es, siempre que sea adecuado se realizarán un número finito de instrucciones, en un tiempo finito y requiriendo una cantidad finita de esfuerzo.

**4.2 Programa:** Un programa de computadoras, también llamado software, es un conjunto de códigos o instrucciones secuenciales que describen, definen o caracterizan la realización de una acción en la computadora. Si bien en los inicios de la historia de la programación los programas dictaban instrucciones directamente a las máquinas, hoy los programas se escriben usando "lenguajes de programación", que son a su vez otros programas que para simplificar la tarea ya traen construidas secuencias y utilidades para facilitar la tarea del programador. También podemos decir que un **Lenguaje de Programación** es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis.

Los **lenguajes de programación** se utilizan para escribir programas. Los programas de las computadoras modernas constan de secuencias de instrucciones que se codifican como secuencias de dígitos numéricos, que podrán entender dichas computadoras.

El sistema de codificación se conoce como **lenguaje máquina** que es el lenguaje nativo de una computadora, pero la escritura de programas en lenguaje máquina es una tarea tediosa y difícil ya que sus instrucciones son secuencias de 0 y 1





(*patrones de bit*, tales como 11110000, 01110011,...) que son muy difíciles de recordar y manipular por las personas. En consecuencia, se necesitan lenguajes de programación “amigables con el programador” que permitan escribir los programas para poder “charlar” con facilidad con las computadoras.

Sin embargo, las computadoras sólo entienden las instrucciones en lenguaje máquina, por lo que será preciso traducir los programas resultantes a lenguajes de máquina antes de que puedan ser ejecutadas por ellas. Cada lenguaje de programación tiene un conjunto o “juego” de instrucciones (acciones u operaciones que debe realizar la máquina) que la computadora podrá entender directamente en su código máquina o bien se traducirán a dicho código máquina. Las instrucciones básicas y comunes en casi todos los lenguajes de programación son:

- *Instrucciones de entrada/salida.* Instrucciones de transferencia de información entre dispositivos periféricos y la memoria central, tales como "leer de..." o bien "escribir en...".
- *Instrucciones de cálculo.* Instrucciones para que la computadora pueda realizar operaciones aritméticas.
- *Instrucciones de control.* Instrucciones que modifican la secuencia de la ejecución del programa.

#### 4.3 Clasificación de los Lenguajes de Programación

Los programadores escriben instrucciones en diversos lenguajes de programación. La computadora puede entender directamente algunos de ellos, pero otros requieren pasos de traducción intermedios. Hoy día se utilizan cientos de lenguajes de computadora.

Los Lenguajes de Programación pueden clasificarse **de acuerdo con su uso** en:

1. Lenguajes desarrollados para el cálculo numérico. Tales como FORTRAN, Mathematica y Matlab.
2. Lenguajes para **sistemas**. Como C, C++ y ensamblador.
3. Lenguajes para aplicaciones de **Inteligencia Artificial**. Tales como Prolog, y Lisp. También se pueden clasificar de acuerdo con el **tipo de instrucciones de que constan**. En esta clasificación se tiene al **lenguaje máquina**, al **lenguaje**



**ensamblador** y al **lenguaje de alto nivel**. Se presenta a continuación una descripción de cada uno de ellos.

**Lenguaje máquina (Binario)** Una computadora sólo puede entender el lenguaje máquina. El lenguaje de máquina ordena a la computadora realizar sus operaciones fundamentales una por una. Dicho lenguaje es difícil de usar para las persona porque trabajar con números. **Lenguajes de bajo nivel (ensamblador)** Para facilitar y agilizar su labor a los programadores, se buscaron nuevos lenguajes. Cuando abstraemos los opcodes y los sustituimos por una palabra que sea una clave de su significado, se tiene el concepto de Lenguaje Ensamblador.

**Lenguajes de alto nivel** Para acelerar aun más el proceso de programación se desarrollaron los lenguajes de alto nivel, en los que se puede escribir un sólo enunciado para realizar tareas sustanciales. Los lenguajes de alto nivel permiten a los programadores escribir instrucciones que asemejan al inglés cotidiano y contiene notaciones matemáticas de uso común.

### EJEMPLO DE TIPOS DE LENGUAJES

#### Lenguaje Máquina

```
100001010101010  
100100101010100  
100011100101110
```

#### Lenguaje de Nivel Bajo (Ensamblador)

```
LOAD R1, (B)  
LOAD R2, (C)  
ADD R1, R2  
STORE (A), R1
```

#### Lenguajes de Alto Nivel

```
A = B + C;
```

Partes de un programa

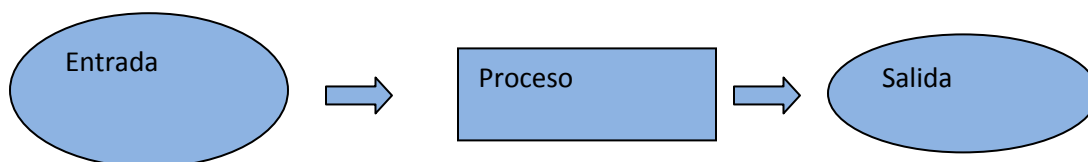


Imagen2.partes de un programa

### UN ALGORITMO ES



Un algoritmo es una secuencia de pasos lógicos y ordenados con las cuales le damos solución a un problema determinado. En la vida diaria cada uno de nosotros diseña y realiza algoritmos para solucionar los problemas cotidianos, es así que al levantarnos de la cama ya tenemos en la mente una serie de pasos que debemos seguir para llegar al sitio de estudio o al sitio de trabajo. Una vez en el sitio de estudio, tenemos en nuestra mente una serie de tareas que debemos realizar en unos horarios ya definidos. Si quisiéramos realizar una comida especial, en nuestra mente construimos un algoritmo o serie de pasos que debemos seguir en un orden específico para que todo nos salga como queremos. Si quisiéramos transcribir estos pasos en una hoja de papel, para que otra persona realizará las mismas tareas y obtenga el mismo resultado que nosotros, debemos seguir una serie de normas para que esta otra persona nos entienda. Por ejemplo debe estar escrito en el idioma que ella comprende, se deben enumerar los pasos etc.

Las normas que se deben seguir al momento de transcribir el algoritmo dependen de quién será el encargado de ejecutarlo. Muchas veces aplicamos el algoritmo de forma inconsciente, inadvertidamente o automáticamente usamos los algoritmos. Esto sucede cuando el problema que tenemos lo hemos realizado en repetidas veces, por ejemplo supongamos que tenemos que salir de nuestra casa a trabajar, lo hemos hecho tantas veces que nunca enumeramos pasos que hacemos para alcanzar el objetivo. Lo mismo sucede en cualquier actividad diaria que realizamos.

### ALGUNAS DE LAS CARACTERÍSTICAS DE LOS ALGORITMOS

1. Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
2. Un algoritmo debe estar definido. Si se sigue el algoritmo dos veces, se debe obtener el mismo resultado cada vez.
3. Un algoritmo debe ser finito. Si se sigue el algoritmo. Se debe terminar en algún momento, o sea debe tener un número finito de pasos y producir un resultado.

**Todo Algoritmo: Acciones** + orden + datos = solución. Independencia del lenguaje de programación. Características: Conciso y detallado. Flexibilidad. Finito, limitado. Exacto, preciso Claro y sencillo. Existen dos grandes clases de grupos así:

Algoritmos cualitativos: son todos aquellos pasos o instrucciones descritos por medio de palabras o variables ordenados y lógicos que nos llevan a la



realización de una actividad o tarea o a la solución de un problema, normalmente no involucran cálculos numéricos. Ejemplo: Un cliente realizar un pedido en una fabrica. La fabrica examina en su banco de datos la ficha del cliente, si es solvente entonces la empresa autoriza le pedido, en caso contrario lo rechaza. El algoritmo seria.

1. Inicio
2. Leer el pedido
3. Examinar la ficha de cliente
4. Si el cliente es solvente aceptar pedido en caso contrario rechazar pedido.
5. Fin.

Algoritmos cuantitativos: son una serie de pasos o instrucciones ordenados y lógicos que nos llevan a la solución de un problema o a la obtención de una respuesta, normalmente involucran cálculos numéricos de manera explicita. Ejemplo: resolver la siguiente ecuación matemática  $F = M * 7 X$

Una solución es:

1. Inicio
2. Asignar un valor a M
3. Asignar valor a X
4. Realizar las operaciones  $M*7$  y al resultado restarle el valor de X, luego guardar el valor resultante en la variable F es decir  $F = M * Z - X$
5. Mostrar el valor que guarda
6. Fin.

**5.- Herramientas de programación.** Las herramientas de programación empleadas como lenguajes algorítmicos son:

1. Diagramas de flujo
2. Pseudocódigo
3. Diagramas de Nassi-Shneiderman (N-S)



## 5.1 DIAGRAMA DE FLUJO

Es un esquema para representar gráficamente un algoritmo. Se basan en la utilización de diversos símbolos para representar operaciones específicas, es decir, es la representación gráfica de las distintas operaciones que se tienen que realizar para resolver un problema, con indicación expresa el orden lógico en que deben realizarse. 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. Si el Diagrama de Flujo está completo y correcto, el paso del mismo a un Lenguaje de Programación es relativamente simple y directo. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación. Para hacer comprensibles los diagramas a todas las personas, los símbolos se someten a una normalización; es decir, se hicieron símbolos casi universales, ya que, en un principio cada usuario podría tener sus propios símbolos para representar sus procesos en forma de Diagrama de flujo. Esto trajo como consecuencia que sólo aquel que conocía sus símbolos, los podía interpretar. La simbología utilizada para la elaboración de diagramas de flujo es variable y debe ajustarse a un patrón definido previamente.

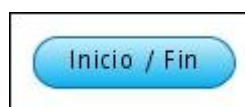
El diagrama de flujo representa la forma más tradicional y duradera para especificar los detalles algorítmicos de un proceso. Se utiliza principalmente en programación, economía y procesos industriales.

### 5.1.1 IMPORTANCIA DE LOS DIAGRAMAS DE FLUJO

Los diagramas de flujo son importantes porque nos facilita la manera de representar visualmente el flujo de datos por medio de un sistema de tratamiento de información, en este realizamos un análisis de los procesos o procedimientos que requerimos para realizar un programa o un objetivo.

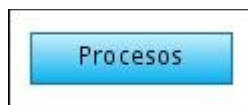
### 5.1.2 Reglas para dibujar diagramas de flujo.

Los Diagramas de flujo se dibujan generalmente usando algunos símbolos estándares; sin embargo, algunos símbolos especiales pueden también ser desarrollados cuando sean requeridos. Algunos símbolos estándares, que se requieren con frecuencia para diagramar programas de computadora se muestran a continuación:



Inicio o fin del programa





Pasos, procesos o líneas de  
instrucción de programa de  
computo



Operaciones de entrada y  
salida



Toma de decisiones y  
Ramificación



Conector para unir el flujo a  
otra parte del diagrama



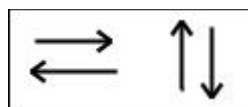
Cinta magnética



Disco magnético



Conector de pagina



Líneas de flujo



Anotación



Display, para mostrar datos



Envía datos a la impresora  
imagen3.diagramas de flujo

**Observación:** Para elaborar los diagramas con los símbolos respectivos y obtener la correcta elaboración, existen varias formas, por ejemplo: para hacerlos manualmente se puede usar una regleta creada para este fin, la cual se consigue en cualquier papelería. También se encuentran diversos programas para realizar estos diagramas. Algunos de ellos son:

SmartDraw, DFD, Microsoft Office Visio Professional 2010, GPL. Recuerde que se deben adquirir las licencias respectivas, para evitar la piratería de Software.

## 6. Los datos

Los datos son objetos sobre los que opera una computadora. Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como b, un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.



Imagen 4 tipos de datos

Los datos son objetos sobre los que opera una computadora y se pueden clasificar en:

- Simples
- Estructurados



- Definidos por el usuario: son propios de lenguajes modernos.

Los datos simples se caracterizan porque ocupan sólo una casilla de memoria, por lo tanto una variable simple hace referencia a un único valor a la vez. Dentro de este grupo de datos encontramos:

- Numéricos
  - Lógicos
  - De Caracteres
- Los datos estructurados están constituidos por datos simple:
    - Cadenas (string)
    - Vectores y matrices (arrays)
    - Registros (record)
    - Archivos o ficheros (file)
    - Punteros (pointer)

Datos numéricos: Los datos numéricos son los relativos a las diferentes clases de números. Estos se dividen en: enteros y reales.

Los enteros. Se denominan en ocasiones número de coma o punto fijo. Sin números que no tienen parte decimal y forma parte del conjunto de los enteros.

Los reales son números con parte entera y parte decimal que pertenece al conjunto de los números reales.

Ejemplo 3.5 5.67 -287.32 654.887

## 6.1 Datos simples



**Enteros.-** son números que no contienen componentes fraccionarias y, por lo tanto, no incluyen el punto decimal, pueden estar precedidos del signo + o --. Por ejemplo:

465 -387 9 99 -1578 125550

**Reales (flotantes).-** son números que contienen una parte fraccionaria y, por lo tanto, incluyen el punto decimal, pueden estar precedidos del signo + o --. Por ejemplo:

17725.87 -45128.0 158000.75 -35.58788

**Carácter.-** contiene un solo carácter y va encerrado entre comillas simples. Por ejemplo:

'a' 'A' '\$' '#' '7' '+' 'h' 'F'

**Cadena de caracteres.-** contiene un conjunto de caracteres alfabéticos, numéricos, y especiales, que sirven para representar y manejar datos como: nombres, descripciones de artículos, direcciones, etc. Van encerrados entre comillas dobles, por ejemplo: "xyz" "Juan Manuel" "95-15-24-65" "\$8#4"

**Booleanos.-** toma los valores de verdadero (true) o falso (false), no puede ser leído o escrito, sólo asignado, es útil para plantear cierto tipo de condiciones en el manejo de las estructuras lógicas de control.

Existen otra clase de tipo de datos como:

Datos tipo subserie:

Datos tipo ordinal

Dato

### Para tener en cuenta

- Para que el computador reconozca los datos alfanuméricos como caracteres o cadena carácter, (en la mayoría de lenguaje) deben encerrarse entre comillas sencillas o apostrofes.
- La longitud de una cadena de caracteres depende de los lenguajes de programación.
- Los espacios en blanco que se encuentran dentro de una cadena de caracteres también son contados como caracteres.

## 6.2 Identificadores constantes y Variables.



Cuando se escriben programas de computadoras en un lenguaje de alto nivel se necesita nombres para identificar a los objetos que se desean manipular: variables, constantes, procedimientos, funciones o métodos, procedimientos etc.

Un **identificador**: Son los nombres que se dan a los programas, constantes, variables, subprogramas y otros objetos o entidades de dígito, pero deben comenzar con una letra. Es un nombre que define a una variable, una función o un tipo de datos. Un identificador válido ha de empezar por una letra o por el carácter de subrayado `_`, seguido de cualquier cantidad de letras, dígitos o subrayados. Muy importante si se distinguen mayúsculas de minúsculas. No se pueden utilizar palabras reservadas como **int**, **char** o **while**. Muchos compiladores no permiten letras acentuadas o eñes.

#### Ejemplos válidos:

```
char letra;  
int Letra;  
float CHAR;  
int __variable__;  
int cantidad_envases;  
double precio123;  
int __;
```

#### Ejemplos no válidos:

```
int 123var;           /* Empieza por dígitos */  
char int;             /* Palabra reservada */  
int una sola;         /* Contiene espacios */  
int US$;              /* Contiene $ */  
int var.nueva;        /* Contiene el punto /  
int eñe;              /* Puede no funcionar */
```

### 6.3 Constantes





Es el nombre que se le da a un campo en una memoria, cuyo contenido no cambia o varía a través de un proceso. Son datos que no cambian durante la ejecución de un programa. Existen tantos tipos de constantes como tipos de datos, por lo tanto, puede haber constantes enteras, reales (flotantes), de carácter, de cadenas de caracteres, booleanas, etc.

#### 6.4 Variables.

Es el nombre que se le da a un campo en una memoria, cuyo contenido puede cambiar o variar a través de un proceso. Son datos que no cambian durante la ejecución de un programa. Existen tipos de constantes como tipos de datos, por lo tanto, puede haber constantes enteras, reales (flotantes), de carácter, de cadenas de caracteres, booleanas, etc. Para nombrar variables se utilizan identificadores. La declaración de una variable origina una reserva de una posición en la memoria de la computadora y que sea etiquetada con el correspondiente con el correspondiente identificador. Se utiliza para representar un dato tipo entero, real (flotante), o una constante de carácter. Por ejemplo:

Variable **i** es de tipo entero

Variable **prom** es de tipo real (flotante)

Variable **opción** es de tipo carácter

##### 6.4.1 Clasificación de las Variables

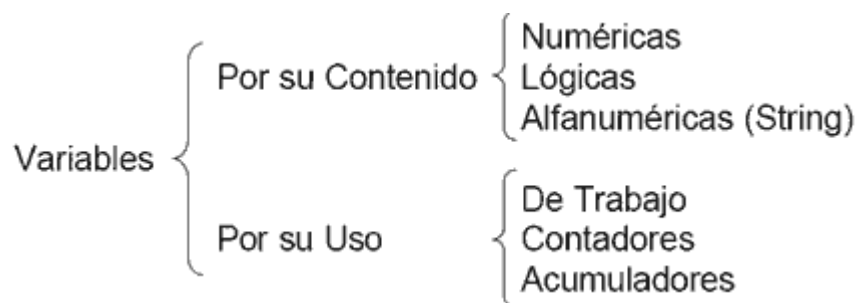


Imagen5.Clasificación de las Variables

#### Por su contenido



**Variables Numéricas:** Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal.

Ejemplo:

iva = 0.15 pi = 3.1416 costo = 2500

- **Variables Lógicas:** Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.
- **Variables Alfanuméricas:** Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales).

Ejemplo:

letra = 'a' apellido = 'lopez' direccion = 'Av. Libertad #190'

### Por su uso

- **Variables de Trabajo:** Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.  
Ejemplo:  
 $\text{Suma} = a + b / c$
- **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.
- **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.

### 6.5. Expresión aritmética o algebraica.



Es el conjunto de variables y/o constantes unidas por operadores aritméticos.  
Ejemplos.

$$Y = X1 + X2 + 6$$

Diagram illustrating the components of the expression  $Y = X1 + X2 + 6$ :

- $X1$  is labeled as a **Variable**.
- $X2$  is labeled as a **variable**.
- $6$  is labeled as a **constante**.

**Operador.** El operador se define como símbolo o palabra que permite formular operaciones, establecer relaciones y hacer comparaciones de tipo lógico y matemático.

Entre los tipos de operadores tenemos:

- Operadores aritméticos
- Operadores de relación (relacionales)
- Operadores lógicos.

**Operadores aritméticos.** Se utilizan

- Operadores de relación (relacionales)
- Operadores lógicos.

Operadores aritméticos



Operación	Operador	Tipo	Operando	Resultado
Adición	+	Binario	Ambos enteros	Entero
			Un operando no es entero	Doble precisión
Sustracción	-	Binario	Ambos enteros	Entero
			Un operando no es entero	Doble precisión
Multiplicación	*	Binario	Ambos enteros	Entero
			Un operando no es entero	Doble precisión
División	/	Binario	Ambos enteros	Entero
			Un operando no es entero	Doble precisión
Módulo	%	Binario	Ambos enteros	Entero
			Un entero	Entero
Negación	-	Unario	Un operando de punto flotante	Doble precisión
			o un operando de doble precisión	Doble precisión

Imagen 6 operadores aritméticos

No todos los operadores aritméticos, existen en todos los lenguajes, por lo tanto al codificar en determinado lenguaje, se debe investigar muy bien cual se utiliza.

## 6.6 Jerarquía de operaciones

1. Operador exponencial: ( \*\* ,)
2. Operadores de multiplicación y división: ( , /)
3. Operadores de suma y resta: ( + , - )
4. Operaciones de división entera y de residuo: (div, mod)

Ejemplos

a)  $4 + 6 * 15$   
 $4 + 90$   
 $94$

b)  $5 + 7 * 3 + 2 * 4$   
 $5 + 21 + 8$   
 $34$

Uso de paréntesis: Cuando una expresión aritmética posee paréntesis, se deben tener en cuenta los siguientes parámetros:

- La computadora ejecuta primero las operaciones que estén dentro del paréntesis ( ).
- Si existen varios pares de paréntesis, se comienza a realizar las operaciones por el más interno hasta llegar al externo.



- Dentro de cada paréntesis, se sigue la jerarquía normal de operaciones

Operadores Lógicos: se utilizan para crear las operaciones lógicas o booleanas.

Operadores Lógicos		
Operador	Descripción	Resultado
A and B	AND	"true" si A y B son <b>ambos</b> verdaderos
A or B	OR	"true" si <b>ambos</b> o <b>al menos uno</b> de A y B son verdaderos
A == B	Igualdad	"true" si A y B son iguales
A <> B	Desigualdad	"true" si A y B <b>NO</b> son iguales
not B	Desigualdad	"true" si B no es verdadero (siendo B un valor o una expresión booleana)

Imagen7 operador lógico

Operador de Relaciones

Operador	Operación
=	Igual a





<>	No igual a (distinto de)
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que

Imagen8.operador de relaciones

Se pueden utilizar en conjunto los tres tipos de operadores, para crear expresiones complejas. Al igual que en las expresiones aritméticas los paréntesis se pueden utilizar y tendrán prioridad durante cualquier operación.