

1.1 Introducción

El computador no solamente es una maquina que puede realizar procesos para darnos resultados, sin que tengamos la noción exacta de las operaciones que realiza para llegar a esos resultados. Con el computador además de lo anterior también podemos diseñar soluciones a la medida, de problemas específicos que se nos presenten. Mas aun, si estos involucran operaciones matemáticas complejas y/o repetitivas, o requieren del manejo de un volumen muy grande de datos.

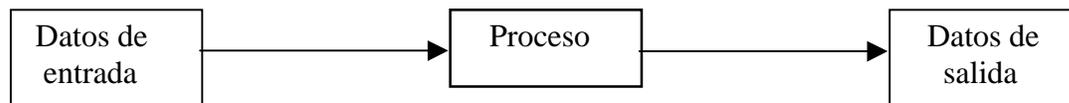
El diseño de soluciones a la medida de nuestros problemas, requiere como en otras disciplinas una metodología que nos enseñe de manera gradual, la forma de llegar a estas soluciones.

A las soluciones creadas por computador se les conoce como **programas** y no son mas que una serie de operaciones que realiza el computador para llegar a un resultado, con un grupo de datos específicos. Lo anterior nos lleva al razonamiento de que un **programa** nos sirve para solucionar un problema específico.

Para poder realizar **programas**, además de conocer la metodología mencionada, también debemos de conocer, de manera específica las funciones que puede realizar el computador y las formas en que se pueden manejar los elementos que hay en la misma.

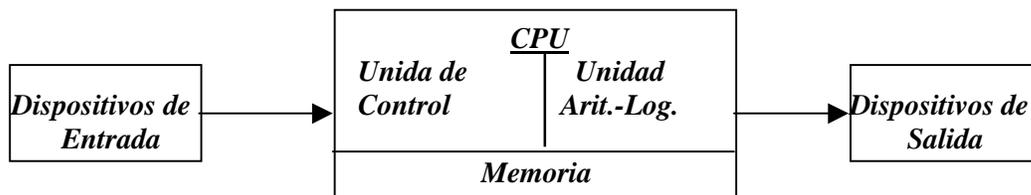
Computador: Es un dispositivo electrónico utilizado para procesar datos y obtener resultados. Los datos y la información se pueden introducir en el computador como entrada (input) y a continuación se procesan para producir una salida (output).

Proceso de información en el computador



Programa: Es el conjunto de instrucciones escritas de algún lenguaje de programación y que ejecutadas secuencialmente resuelven un problema específico.

Organización física de una computador



Dispositivos de Entrada: Como su nombre lo indica, sirven para introducir datos (información) en el computador para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria central o interna. Ejemplos: teclado, scanners (digitalizadores de rastreo), mouse (ratón), trackball (bola de ratón estacionario), joystick (palancas de juego), lápiz óptico.

Dispositivos de Salida: Regresan los datos procesados que sirven de información al usuario. Ejemplo: monitor, impresora.

La Unidad Central de Procesamiento (C.P.U) se divide en dos:

- Unidad de control
- Unidad Aritmético - Lógica

Unidad de Control: Coordina las actividades del computador y determina que operaciones se deben realizar y en que orden; así mismo controla todo el proceso del computador.

Unidad Aritmético - Lógica: Realiza operaciones aritméticas y lógicas, tales como suma, resta, multiplicación, división y comparaciones.

La Memoria del computador se divide en dos:

- Memoria Principal o Interna
- Memoria Auxiliar o Externa

Memoria Principal (interna): La CPU utiliza la memoria del computador para guardar información mientras trabaja con ella; mientras esta información permanezca en memoria, el computador puede tener acceso a ella en forma directa. Esta memoria construida internamente se llama memoria de acceso aleatorio (RAM).

La **memoria interna** consta de dos áreas de memoria:

La memoria **RAM (Random Access Memory):** Recibe el nombre de memoria principal o memoria del usuario, en ella se almacena información sólo mientras el computador esta encendido. Cuando se apaga o arranca nuevamente el computador, la información se pierde, por lo que se dice que la memoria RAM es una memoria volátil.

La memoria **ROM (Read Only Memory):** Es una memoria estática que no puede cambiar, el computador puede leer los datos almacenados en la memoria ROM, pero no se pueden introducir datos en ella, o cambiar los datos que ahí se encuentran; por lo que se dice que esta memoria es de solo lectura. Los datos de la memoria ROM están grabados en forma permanente y son introducidos por el fabricante del computador.

Memoria Auxiliar (Externa): Es donde se almacenan todos los programas o datos que el usuario desee. Los dispositivos de almacenamiento o memorias auxiliares (externas o secundarias) mas comúnmente utilizados son: cintas magnéticas y discos magnéticos.

1.2 Definición de Lenguaje

Lenguaje: Es una serie de símbolos que sirven para transmitir uno o más mensajes (ideas) entre dos entidades diferentes. A la transmisión de mensajes se le conoce comúnmente como **comunicación**.

La **comunicación** es un proceso complejo que requiere una serie de reglas simples, pero indispensables para poderse llevar a cabo. Las dos principales son las siguientes:

- Los mensajes deben correr en un sentido a la vez.
- Debe forzosamente existir 4 elementos: Emisor, Receptor, Medio de Comunicación y Mensaje.

Lenguajes de Programación

Es un conjunto de símbolos, caracteres y reglas (programas) que le permiten a las personas comunicarse con el computador.

Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, cálculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.

Los lenguajes de programación se clasifican en:

- **Lenguaje Máquina:** Son aquellos cuyas instrucciones son directamente entendibles por el computador y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje máquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 o 1).
- **Lenguaje de Bajo Nivel (Ensamblador):** En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Los lenguajes de programación de alto nivel (BASIC, Pascal, COBOL, FORTRAN, etc.) son aquellos en los que las instrucciones o sentencias al computador son escritas con palabras similares a los lenguajes humanos (en general en inglés), lo que facilita la escritura y comprensión del programa.

1.3 Definición de Algoritmo

La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhwarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

1.4 Tipos de Algoritmos

- ***Cualitativos:*** Son aquellos en los que se describen los pasos utilizando palabras.
- ***Cuantitativos:*** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

1.5 Lenguajes Algorítmicos

Es una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Tipos de Lenguajes Algorítmicos

- ***Gráficos:*** Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).
- ***No Gráficos:*** Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudolenguaje).

1.6 Metodología para la solución de problemas por medio de computador

1.7 Definición del Problema

Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice el computador; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.

1.8 Análisis del Problema

Una vez que se ha comprendido lo que se desea del computador, es necesario definir:

Los datos de entrada.

Cual es la información que se desea producir (salida)

Los métodos y fórmulas que se necesitan para procesar los datos.

Una recomendación muy practica es el que nos pongamos en el lugar del computador y analicemos que es lo que necesitamos que nos ordenen y en que secuencia para producir los resultados esperados.

1.9 Diseño del Algoritmo

Las características de un buen algoritmo son:

Debe tener un punto particular de inicio.

Debe ser definido, no debe permitir dobles interpretaciones.

Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.

Debe ser finito en tamaño y tiempo de ejecución.

1.10 Codificación

La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudolenguaje), en una serie de instrucciones detalladas, en un código reconocible por el computador, la serie de instrucciones detalladas se le conoce como código fuente, el cual se escribe en un lenguaje de programación o lenguaje de alto nivel.

1.11 Prueba y Depuración

Los errores humanos dentro de la programación de computadores son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama ***depuración***.

La ***depuración o prueba*** resulta una tarea tan creativa como el mismo desarrollo de la solución, por ello se debe considerar con el mismo interés y entusiasmo.

Resulta conveniente observar los siguientes principios al realizar una depuración, ya que de este trabajo depende el éxito de nuestra solución.

1.12 Documentación

Es la guía o comunicación escrita es sus variadas formas, ya sea en enunciados, procedimientos, dibujos o diagramas.

A menudo un programa escrito por una persona, es usado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).

La **documentación** se divide en tres partes:

Documentación Interna
Documentación Externa
Manual del Usuario

➤ Documentación Interna: Son los comentarios o mensaje que se añaden al código fuente para hacer mas claro el entendimiento de un proceso.

➤ Documentación Externa: Se define en un documento escrito los siguientes puntos:

 Descripción del Problema
 Nombre del Autor
Algoritmo (diagrama de flujo o pseudolenguaje)
 Diccionario de Datos
 Código Fuente (programa)

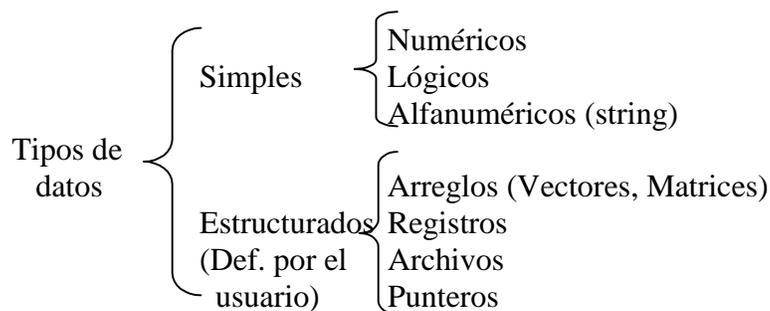
➤ Manual del Usuario: Describe paso a paso la manera como funciona el programa, con el fin de que el usuario obtenga el resultado deseado.

1.13 Mantenimiento

Se lleva acabo después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa este correctamente documentado.

2.1 Tipos De Datos

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.



Tipos de Datos Simples

- **Datos Numéricos:** Permiten representar valores escalares de forma numérica, esto incluye a los números enteros y los reales. Este tipo de datos permiten realizar operaciones aritméticas comunes.
- **Datos Lógicos:** Son aquellos que solo pueden tener dos valores (cierto o falso) ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).
- **Datos Alfanuméricos (String):** Es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombres de personas, direcciones, etc. Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir no es posible hacer operaciones con ellos. Este tipo de datos se representan encerrados entre comillas.

Ejemplo:
"Región del Maule"
"1997"

2.2 Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales. Por ejemplo:

$$a+(b + 3)/c$$

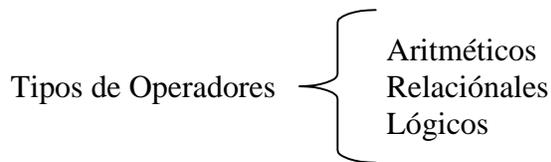
Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.

Una expresión consta de operadores y operandos. Según sea el tipo de datos que manipulan, se clasifican las expresiones en:

- Aritméticas
- Relacionales
- Lógicas

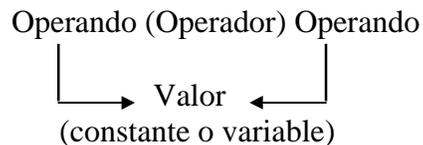
2.3 Operadores y Operandos

➤ **Operadores:** Son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes. Es decir, los operadores nos permiten manipular valores.



➤ **Operadores Aritméticos:** Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes).

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real.



Operadores Aritméticos

- + Suma
- Resta
- * Multiplicación
- / División
- Mod Modulo (residuo de la división entera)

Ejemplos:

Expresión	Resultado
$7 / 2$	3.5
$12 \bmod 7$	5
$4 + 2 * 5$	14

Prioridad de los Operadores Aritméticos

✱ Todas las expresiones entre paréntesis se evalúan primero. Las expresiones con paréntesis anidados se evalúan de dentro hacia fuera, el paréntesis más interno se evalúa primero.

✱ Dentro de una misma expresión los operadores se evalúan en el siguiente orden.

- 1.- ^ Exponenciación
- 2.- *, /, mod Multiplicación, división, modulo.
- 3.- +, - Suma y resta.

✱ Los operadores en una misma expresión con igual nivel de prioridad se evalúan de izquierda a derecha.

Ejemplos:

$4 + 2 * 5 = 14$	
$23 * 2 / 5 = 9.2$	$46 / 5 = 9.2$
$3 + 5 * (10 - (2 + 4)) = 23$	$3 + 5 * (10 - 6) = 3 + 5 * 4 = 3 + 20 = 23$
$3.5 + 5.09 - 14.0 / 40 = 5.09$	$3.5 + 5.09 - 3.5 = 8.59 - 3.5 = 5.09$
$2.1 * (1.5 + 3.0 * 4.1) = 28.98$	$2.1 * (1.5 + 12.3) = 2.1 * 13.8 = 28.98$

➤ ***Operadores Relacionales:***

- ✱ Se utilizan para establecer una relación entre dos valores.
- ✱ Compara estos valores entre si y esta comparación produce un resultado de certeza o falsedad (verdadero o falso).
- ✱ Los operadores relacionales comparan valores del mismo tipo (numéricos o cadenas)
- ✱ Tienen el mismo nivel de prioridad en su evaluación.
- ✱ Los operadores relacionales tiene menor prioridad que los aritméticos.

Operadores Relacionales

>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	Diferente
=	Igual

Ejemplos:

Si $a = 10$ $b = 20$ $c = 30$

$a + b > c$ Falso
 $a - b < c$ Verdadero
 $a - b = c$ Falso
 $a * b < > c$ Verdadero

Ejemplos no lógicos:

$a < b < c$
 $10 < 20 < 30$
True < 30 (no es lógico porque tiene diferentes operandos)

➤ **Operadores Lógicos:**

- ✱ Estos operadores se utilizan para establecer relaciones entre valores lógicos.
- ✱ Estos valores pueden ser resultado de una expresión relacional.

Operadores Lógicos

And Y
Or O
Not Negación

Operador And

Operando1	Operador	Operando2	Resultado
T	AND	T	T
T		F	F
F		T	F
F		F	F

Operador Or

Operando1	Operador	Operando2	Resultado
T	OR	T	T
T		F	T
F		T	T
F		F	F

Operador Not

Operando	Resultado
T	F
F	T

2.4 Identificadores

Los *identificadores* representan los datos de un programa (constantes, variables, tipos de datos). Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria del computador, que nos permite acceder a su contenido.

Ejemplo: Nombre
 Num_hrs
 Calif2

Reglas para formar un Identificador

- ✱ Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no deben contener espacios en blanco.
- ✱ Letras, dígitos y caracteres como la subraya (_) están permitidos después del primer carácter.
- ✱ La longitud de identificadores puede ser de hasta 256 caracteres.

Constantes y Variables

- **Constante:** Una constante es un dato numérico o alfanumérico que no cambia durante la ejecución del programa.

Ejemplo:

pi = 3.1416

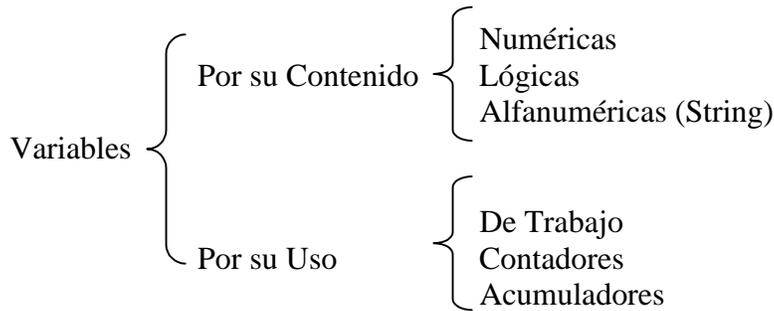
- **Variable:** Es un espacio en la memoria del computador que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa. Para poder reconocer una variable en la memoria del computador, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

Ejemplo:

área = pi * radio ^ 2

Las variables son : el radio, el área y la constante es pi

Clasificación de las Variables



Por su Contenido

➤ **Variable 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:

suma=a+b/c

➤ **Contadores:** Se utilizan para llevar el control del numero 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.

3.1 Top Down

También conocida como de arriba-abajo y consiste en establecer una serie de niveles de mayor a menor complejidad (arriba-abajo) que den solución al problema. Consiste en efectuar una relación entre las etapas de la estructuración de forma que una etapa jerárquica y su inmediato inferior se relacionen mediante entradas y salidas de información.

Este diseño consiste en una serie de descomposiciones sucesivas del problema inicial, que recibe el refinamiento progresivo del repertorio de instrucciones que van a formar parte del programa.

La utilización de la técnica de diseño ***Top-Down*** tiene los siguientes objetivos básicos:

- Simplificación del problema y de los subprogramas de cada descomposición.
- Las diferentes partes del problema pueden ser programadas de modo independiente e incluso por diferentes personas.
- El programa final queda estructurado en forma de bloque o módulos lo que hace más sencilla su lectura y mantenimiento.

3.2 Bottom Up

El diseño ascendente se refiere a la identificación de aquellos procesos que necesitan computarizarse con forme vayan apareciendo, su análisis como sistema y su codificación, o bien, la adquisición de paquetes de software para satisfacer el problema inmediato.

Cuando la programación se realiza internamente y haciendo un enfoque ascendente, es difícil llegar a integrar los subsistemas al grado tal de que el desempeño global, sea fluido. Los problemas de integración entre los subsistemas son sumamente costosos y muchos de ellos no se solucionan hasta que la programación alcanza la fecha límite para la integración total del sistema. En esta fecha, ya se cuenta con muy poco tiempo, presupuesto o paciencia de los usuarios, como para corregir aquellas delicadas interfaces, que en un principio, se ignoran.

Aunque cada subsistema parece ofrecer lo que se requiere, cuando se contempla al sistema como una entidad global, adolece de ciertas limitaciones por haber tomado un enfoque ascendente. Uno de ellos es la duplicación de esfuerzos para acceder el software y más aun al introducir los datos. Otro es, que se introducen al sistema muchos datos carentes de valor. Un tercero y tal vez el mas serio inconveniente del enfoque ascendente, es que los objetivos globales de la organización no fueron considerados y en consecuencia no se satisfacen.

Las dos herramientas utilizadas comúnmente para diseñar algoritmos son:

Diagrama de Flujo

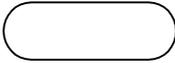
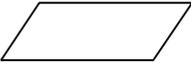
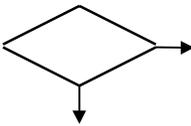
Pseudocódigo

4.1 Diagrama de Flujo

Un diagrama de flujo es la representación gráfica de un algoritmo. También se puede decir que es la representación detallada en forma gráfica de como deben realizarse los pasos en el computador para producir resultados.

Esta representación gráfica se da cuando varios símbolos (que indican diferentes procesos en el computador), se relacionan entre si mediante líneas que indican el orden en que se deben ejecutar los procesos.

Los símbolos utilizados han sido normalizados por el instituto norteamericano de normalización (ANSI).

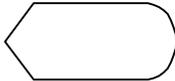
<u>SÍMBOLO</u>	<u>DESCRIPCIÓN</u>
	Indica el inicio y el final de nuestro diagrama de flujo.
	Indica la entrada y salida de datos.
	Símbolo de proceso y nos indica la asignación de un valor en la memoria y/o la ejecución de una operación aritmética.
	Símbolo de decisión indica la realización de una comparación de valores.
	Se utiliza para representar los subprogramas.
	Conector dentro de pagina. Representa la continuidad del diagrama dentro de la misma página.



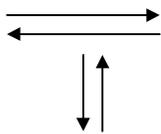
Conector fuera de página. Representa la continuidad del diagrama en otra página.



Indica la salida de información por impresora.



Indica la salida de información en la pantalla o monitor.



Líneas de flujo o dirección. Indican la secuencia en que se realizan las operaciones.

Recomendaciones para el diseño de Diagramas de Flujo

- ✱ Se deben de usar solamente líneas de flujo horizontales y/o verticales.
- ✱ Se debe evitar el cruce de líneas utilizando los conectores.
- ✱ Se deben usar conectores solo cuando sea necesario.
- ✱ No deben quedar líneas de flujo sin conectar.
- ✱ Se deben trazar los símbolos de manera que se puedan leer de arriba hacia abajo y de izquierda a derecha.
- ✱ Todo texto escrito dentro de un símbolo deberá ser escrito claramente, evitando el uso de muchas palabras.

4.2 Pseudolenguaje

Mezcla de lenguaje de programación y español (o inglés o cualquier otro idioma) que se emplea, dentro de la programación estructurada, para realizar el diseño de un programa. En esencial, el pseudolenguaje se puede definir como un lenguaje de especificaciones de algoritmos.

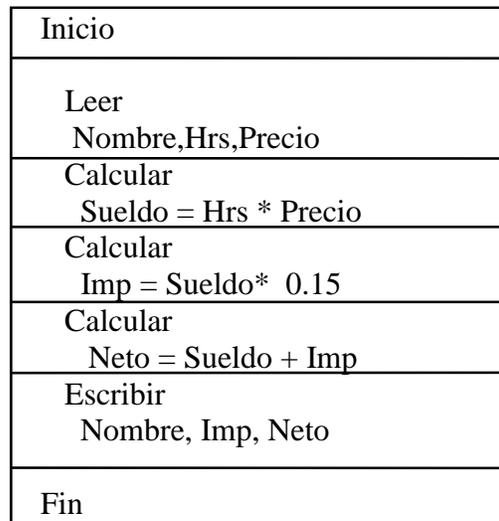
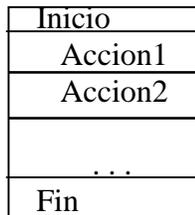
Es la representación narrativa de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. El pseudolenguaje utiliza palabras que indican el proceso a realizar.

Ventajas de utilizar un Pseudolenguaje a un Diagrama de Flujo

- ✱ Ocupa menos espacio en una hoja de papel
- ✱ Permite representar en forma fácil operaciones repetitivas complejas
- ✱ Es muy fácil pasar de pseudolenguaje a un programa en algún lenguaje de programación.
- ✱ Si se siguen las reglas se puede observar claramente los niveles que tiene cada operación.

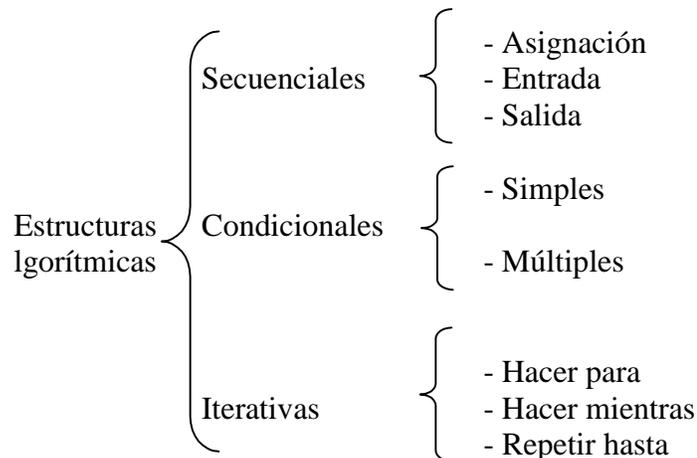
4.3 Diagramas estructurados (Nassi-Schneiderman)

El diagrama estructurado N-S también conocido como diagrama de chapin es como un diagrama de flujo en el que se omiten las flechas de unión y las cajas son contiguas. Las acciones sucesivas se pueden escribir en cajas sucesivas y como en los diagramas de flujo, se pueden escribir diferentes acciones en una caja. Un algoritmo se represente en la sig. forma:



ESTRUCTURAS ALGORITMICAS

Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas estructuras se clasifican de acuerdo con su complejidad en:



5.1. Estructuras Secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:

```
Inicio
Accion1
Accion2
.
.
AccionN
Fin
```

- **Asignación:** La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor. La asignación se puede clasificar de la siguiente forma:

- **Simple:** Consiste en pasar un valor constante a una variable ($a=15$)
- **Contador:** Consiste en usarla como un verificador del numero de veces que se realiza un proceso ($a=a+1$)
- **Acumulador:** Consiste en usarla como un sumador en un proceso ($a=a+b$)

- **De trabajo:** Donde puede recibir el resultado de una operación matemática que involucre muchas variables ($a=c+b*2/4$).

- **Lectura:** La lectura consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor. Esta operación se representa en un pseudolenguaje como sigue:

Leer a, b

Donde “a” y “b” son las variables que recibirán los valores

Escritura: Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje. Este proceso se representa en un pseudolenguaje como sigue:

Escribir “El resultado es:”, R

Donde “El resultado es:” es un mensaje que se desea aparezca y R es una variable que contiene un valor.

Problemas Secuenciales

1) Suponga que un individuo desea invertir su capital en un banco y desea saber cuanto dinero ganará después de un mes si el banco paga a razón de 2% mensual.

Análisis

Datos de entrada

Capital a invertir (cap_invertir)

Interés pagado por el banco = 2% mensual

Datos de salida

Ganancia obtenida en un mes (ganancia)

Inicio

Entero cap_invertir

Real ganancia

Leer cap_invertir

$ganancia = cap_invertir * 0.02$

Escribir ganancia

Fin

2) Un vendedor recibe un sueldo base mas un 10% extra por comisión de sus ventas, el vendedor desea saber cuanto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

Análisis

Datos de entrada

Sueldo base del vendedor (sueldo_base)

Comisión por venta = 10% de las ventas

Realizo 3 ventas: Venta1, Venta2 y Venta3

Datos de salida

Valor de la comisión de las ventas (comision)

Sueldo a recibir (sueldo_recibir)

Consideraciones

Para calcular el valor de la comision, se debe conocer el total de las ventas

Total de las ventas es la suma acumulativa de cada una de las ventas.

La comision corresponde al 10% del total acumulado de las ventas

El sueldo a recibir es la suma del sueldo base más la comision de las ventas.

Algoritmo

Inicio

Entero sueldo_base, venta1, venta2, venta3, total_venta

Real comision, sueldo_recibir

Leer sueldo_base, venta1, venta2, venta3

$total_venta = venta1 + venta2 + venta3$

$comision = total_venta * 0.10$

sueldo_recibir = sueldo_base + comision
Escribir sueldo_recibir, comision

Fin

3) Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuanto deberá pagar finalmente por su compra.

Análisis

Datos de entrada

El total de la compra (total_compra)

Descuento ofrecido por la tienda = 15% sobre el total de la compra

Datos de salida

Total a pagar por la compra (total_pagar)

Consideraciones

El descuento es el 15 del valor total comprado

Descuento = total_compra * 0,15

Algoritmo

Inicio

Entero total_compra

Real descuento, total_pagar

Leer total_compra

descuento = total_compra * 0.15

total_pagar = total_compra - descuento

Escribir total_pagar

Fin

4) Un alumno desea saber cual será su nota final en el curso de Introducción a la programación. Dicha nota se compone de los siguientes porcentajes:

55% del promedio de sus tres notas parciales.

30% de la nota del examen final.

15% de la nota de un trabajo final.

Análisis

Datos de entrada

Tres notas parciales (nota1, nota2 y nota3)

Nota Examen final (examen)

Nota de trabajo final (trabajo)

Datos de salida

Nota final del curso de programación (nota_final)

Consideraciones

Calcular el promedio de sus tres notas parciales

Obtener el 55% del promedio antes calculado

Calcular el 30% de la nota del examen final

Calcular el 15% de la nota del trabajo final

Sumar los valores antes calculados

Algoritmo

Inicio

Real nota1, nota2, nota3, examen, trabajo, nota_final
Real promedio, por_promedio, por_examen, por_trabajo
Leer nota1, nota2, nota3, examen, trabajo
 $\text{promedio} = (\text{nota1} + \text{nota2} + \text{nota3})/3$
 $\text{por_promedio} = \text{promedio} * 0.55$
 $\text{por_examen} = \text{examen} * 0.30$
 $\text{por_trabajo} = \text{trabajo} * 0.15$
 $\text{nota_final} = \text{por_promedio} + \text{por_examen} + \text{por_trabajo}$
Escribir nota_final

Fin

5) Un profesor conoce la cantidad de hombres y mujeres del curso de Algoritmos, desea saber que porcentaje de hombres y que porcentaje de mujeres hay en su grupo de estudiantes.

Análisis

Datos de entrada

Cantidad de hombres (num_hombres)

Cantidad de mujeres (num_mujeres)

Datos de salida

Porcentaje de hombres (por_hombres)

Porcentaje de mujeres (por_mujeres)

Consideraciones

Calcular el total de alumnos del curso (num_hombres + num_mujeres)

A partir de este valor se calculan los porcentajes respectivos

Algoritmo

Inicio

Entero num_hombres, num_mujeres

Real por_hombres, por_mujeres

Entero total_alumnos

Leer num_hombres, num_mujeres

$\text{total_alumnos} = \text{num_hombres} + \text{num_mujeres}$

$\text{por_hombres} = \text{num_hombres} * 100 / \text{total_alumnos}$

$\text{por_mujeres} = \text{num_mujeres} * 100 / \text{total_alumnos}$

Escribir por_hombres, por_mujeres

Fin

Problemas Propuestos

- 1) Dada un cantidad en pesos, obtener la equivalencia en dólares, asumiendo que la unidad cambiaría es un dato desconocido.
- 2) La presión, el volumen y la temperatura de una masa de aire se relacionan por la formula:

$$\text{masa} = (\text{presión} * \text{volumen}) / (0.37 * (\text{temperatura} + 460))$$

Escribir un algoritmo que calcule dicha fórmula.

- 3) Calcular el numero de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, si la formula es:

$$\text{num. pulsaciones} = (220 - \text{edad}) / 10$$

- 4) Calcular el nuevo sueldo de un empleado si obtuvo un incremento del 25% sobre su sueldo anterior.
- 5) En un hospital existen tres áreas: Ginecología, Pediatría, Traumatología. El presupuesto anual del hospital se reparte conforme a la sig. tabla:

Área	Porcentaje del presupuesto
Ginecología	40%
umatología	30%
Pediatría	30%

Obtener la cantidad de dinero que recibirá cada área, para cualquier monto presupuestado.

- 6) El dueño de una tienda compra un artículo a un precio determinado. Obtener el precio en que lo debe vender para obtener una ganancia del 30%.
- 7) Todos los lunes, miércoles y viernes, una persona corre la misma ruta y cronometra los tiempos obtenidos. Determinar el tiempo promedio que la persona tarda en recorrer la ruta en una semana cualquiera.
- 8) Tres personas deciden invertir su dinero para fundar una empresa. Cada una de ellas invierte una cantidad distinta. Obtener el porcentaje que cada cual invierte con respecto a la cantidad total invertida.
- 9) Un alumno desea saber cual será su promedio general en los tres cursos más difíciles que cursa y cual será el promedio que obtendrá en cada una de ellas. Estas materias se evalúan como se muestra a continuación:

La calificación de Matemáticas se obtiene de la sig. manera:

Examen 90%

Promedio de tareas 10%

En esta materia se pidió un total de tres tareas.

La calificación de Física se obtiene de la sig. manera:

Examen 80%

Promedio de tareas 20%

En esta materia se pidió un total de dos tareas.

La calificación de Química se obtiene de la sig. manera:

Examen 85%

Promedio de tareas 15%

En esta materia se pidió un promedio de tres tareas.