7.- *Explique los siguientes* COMANDOS DE BASIC :

NEW _____

LIST _____

DELETE _____

RUN _____

SAVE _____

OLD _____

RENAME _____

CAPITULO 3

DISEÑO DE UN PROGRAMA

OBJETIVO GENERAL

Enseñar y orientar al estudiante en la metodología para el desarrollo de un programa BASIC.

OBJETIVO ESPECIFICO

En este capítulo, se orienta al alumno en los pasos necesarios para iniciarse en el diseño de un programa, con los símbolos más importantes para el arreglo de un diagrama de flujo, así como la forma de construir las instrucciones con las que se alimentarán al computador.

3.1 PASOS PARA EL DISEÑO DE UN PROGRAMA

Antes de mencionar los pasos, debemos conocer qué es un **programa ?**

Un **PROGRAMA** *es una serie ordenada de instrucciones en un lenguaje de alto nivel (***BASIC, COBOL, PASCAL,** *etc.).*

Existen varios tipos de lenguajes y cada uno está enfocado a distintas áreas de trabajo.

Los tipos de lenguajes de programación son : **BASIC, COBOL, FORTRAN, PASCAL, RPG, PL1, ADA,** *etc.*

Los pasos para el diseño de un programa, son básicamente cuatro :

1.- *Conocer el* **OBJETIVO** *del problema a resolver.*
2.- *Hacer un* **ALGORITMO.**
3.- *Diseñar un* **DIAGRAMA DE FLUJO.**
4.- **CODIFICAR** *el programa.*

3.2 PASO 1 .- OBJETIVO DEL PROBLEMA

Para poder diseñar un programa, debemos de saber como llegar al objetivo.

Un ejemplo sencillo es el siguiente :

> *Si me encuentro en mi carro, en el centro de la ciudad y mi objetivo es llegar a Saltillo, lógicamente tomaría la carretera a Saltillo ; pero en caso de tomar la carretera a Reynosa, tal vez llegue a Saltillo, pero con ciertas desventajas, tales como : cansancio, gasto de gasolina, mayor recorrido, etc. Si yo sé como llegar al objetivo, no tendré problemas para resolverlo.*

Esto mismo ocurre para programar una computadora, si sé como obtener una nómina, lo puedo programar; serían los mismos pasos manuales que las instrucciones a utilizar por el computador. Por lo tanto, debemos conocer el procedimiento para llegar al objetivo.

3.3 PASO 2.- CREACION DE UN ALGORITMO

3.3.1 *Un* **ALGORITMO** *es una serie de pasos lógicos que sirven para poder llegar a un objetivo. Es un procedimiento para resolver un problema.*

Los algoritmos pueden ser generales o detallados.

Ejemplo de un algoritmo general :

1.- *Inicio.*
2.- *Capturar información.*
3.- *Calcular deducciones.*
4.- *Calcular percepciones.*
5.- *Quién realizó horas extras ?.*
6.- *Calcular sueldo neto.*
7.- *Dar resultado del sueldo neto.*
8.- *Termina el algoritmo*

En el caso del ejemplo anterior, sabemos que en cada punto existen muchos pasos más, todos esos pasos faltantes van a conformar lo que es un algoritmo detallado.

Cuando llegue a diseñar un programa en la computadora, se recomienda hacer un algoritmo detallado, puesto que el computador no sabrá lo siguiente :

a) .- *Qué porcentaje de impuesto quitará ?.*
b) .- *Si el empleado trabajó horas extras ?.*
c) .- *Cuánto le asignará ?.*
d) .- *Qué datos va a preguntar ?.*
e) .- *Y muchas cuestionamientos más.*

Para poder diseñar un buen programa, siempre le daremos el detalle más mínimo que llegue a necesitar la computadora.

El punto que estamos tratando es el más importante para poder programar, ya que es pura **LOGICA** *y la práctica de éste, nos sirve para pensar en los pasos necesarios que emplearemos en la solución de nuestro problema que queremos computarizar.*

3.3.2 *A continuación se encuentran algunos ejemplos de algoritmos :*

Ejemplo 1.-

Algoritmo para obtener el promedio de un alumno.

1.- *Inicio del algoritmo.*

2.- *Cuál es el nombre del alumno ?.*

3.- *Qué calificaciones obtuvo en sus parciales ?.*

4.- *Sumar las calificaciones.*

5.- *Divida el resultado de la suma de calificaciones entre el número de calificaciones y nos dará el promedio del alumno.*

6.- *Presentar los resultados (Quién es el alumno y el promedio que obtuvo).*

7.- *Finaliza el algoritmo.*

Con el algoritmo anterior nos enseña lo siguiente :

QUE LOS PASOS LLEVAN ORDEN LOGICO

a).- *No puedo obtener el promedio si no tengo la suma.*

b).- *No puedo hacer la suma si no tengo las calificaciones.*

c).- *De dónde obtengo las calificaciones ?, preguntamos por ellas, de una lista !, de los examenes !, etc.*

Observe el ejemplo paso por paso y se dará cuenta, que lo haría en forma manual, con ésto le quiero demostrar que cuando se quiera diseñar un programa, son los mismos pasos manuales que los computarizados.

Lo importante es que entienda como llegar al objetivo que desea.

Ejemplo 2 .-

Algoritmo para obtener el sueldo neto de " N " empleados.

1.- *Inicio del algoritmo.*
2.- *Cuál es nombre del trabajador ?.*
3.- *Si hay trabajador, vé al paso 4, si no hay trabajador vé al paso 9.*
4.- *Cuántas horas trabajó ?.*
5.- *Cuanto le pagan por hora ?.*
6.- *Multiplicar las horas trabajadas por el sueldo hora y obtenemos el sueldo neto.*
7.- *Presentar los resultados (nombre del empleado y su sueldo neto).*
8.- *Ir por otro empleado, continuar con el paso 2.*
9.- *Termina el algoritmo.*

En este ejemplo, no estamos incluyendo deducciones, percepciones, total del sueldo neto, total de trabajadores, etc., pero se pueden agregar nadamás incluyendo los pasos en el lugar donde correspondan.

En el paso 3, se está tomando una decisión, si existe trabajador continuamos preguntando, en caso contrario, daremos por terminado nuestro problema.

Ejemplo 3 .-

Algoritmo para obtener los datos generales de " N " personas.

1.- *Inicio del algoritmo.*
2.- *Cuál es el nombre de la persona ?.*
3.- *Si hay persona vé al paso 4, si no al paso 10.*
4.- *Cuál es su dirección ?.*
5.- *Cuál es su teléfono ?.*
6.- *Qué edad tiene ?.*
7.- *En dónde trabaja ?.*
8.- *Dar los resultados (nombre, dirección, teléfono edad, lugar de trabajo).*
9.- *Ir por otra persona, continuar con el paso 2.*
10.- *Termina el algoritmo.*

Si observamos, en este algorítmo no estamos realizando ninguna operación aritmética, porque no existe la necesidad de hacerlo, solamente nos concretamos a preguntar.

Otra vez ocurre lo mismo que en ejemplos anteriores, el procedimiento manual sería el mismo para el computarizado.

Ejemplo 4.-

Algorítmo para obtener la comisión de cada vendedor que labora en la compañía **" XYZ "** *y obtener el total de la comisiones que tendrá que pagar dicha compañía. Son cinco (5) vendedores.*

Porcentaje de comisión :

Si la venta es menor a **$** *50,000.00, le asignaremos el 10% de comisión, pero si su venta fué igual o mayor a* **$** *50,000.00, le daremos el 20% de comisión.*

1. - *Inicio del algorítmo.*
2. - *Vendedor* 1. *viene el primer vendedor.*
3. - *Ya son más de 5 vendedores vé al paso* 13, *si no lo son vé al paso* 4.
4. - *Nombre del vendedor ?.*
5. - *Cuál fué su venta ?.*
6. - *La venta es menor a* **$** *50,000.00, vé al paso* 7, *si no, vé al paso* 8.
7. - *Comisión es igual a la venta por el porcentaje del* 10%, *Comisión = venta * .10 y despues vé al paso* 9.
8. - *Comisión es igual a la venta por el porcentaje de* 20%, *Comisión = venta * .20.*
9. - *Dar los resultados (nombre del vendedor, la venta que realizó y su comisión).*
10. - *Acumular la comisión del vendedor para el total de comisiones.*
    total de comisiones = total de comisiones + comisión.
11. - *Sumamos* 1 *al número de vendedores.*
12. - *Continuar con el paso* 3.
13. - *Dar el resultado del total de comisiones.*
14. - *Termina el algorítmo.*

En este ejemplo, vemos que trabajamos acumuladores y sumadores como lo demuestran los pasos 10 *y* 11 *respectivamente.*

Se toma una decisión en el paso 3, *para saber si ya pasaron los* 5 *vendedores para calcularle su comisión dependiendo de su venta, en la cual también se toma la decisión en el paso* 7.

3.4 PASO 3 .- DIAGRAMA DE FLUJO

3.4.1 *El* **DIAGRAMA DE FLUJO** *es una representación gráfica de todos los pasos (operaciones) a realizarse para poder llegar al objetivo.*

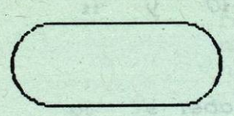
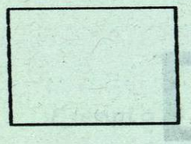
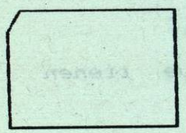
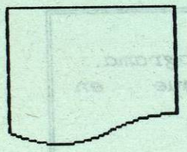
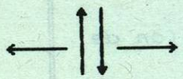
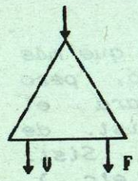
El diagrama de flujo es representado por símbolos que tienen un significado especial.

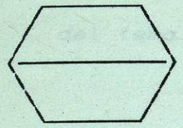
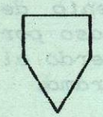
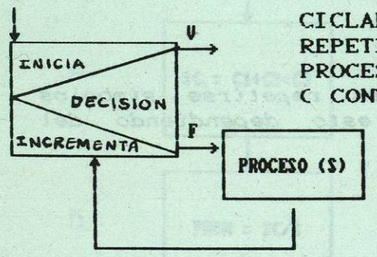
3.4.2 VENTAJAS *del diagrama de flujo :*

1. - *Visualizamos mejor la lógica del programa. Es más entendible en símbolos que en instrucciones.*
2. - *Detectamos más rápido los errores.*
3. - *Es muy utilizado para la documentación del programa y de un sistema de información.*
4. - *Nos ayuda a obtener nuestra codificación de instrucciones.*

3.4.3 SIMBOLOS MAS IMPORTANTES

A continuación, vamos a mostrar los símbolos que más se utilizan en el diseño de los diagramas de flujo, pero tambien hay que mencionar, que existen símbolos para el diseño de todo un sistema de información. (Sist. de contabilidad, Sist. de nómina, Sist. de inventarios, Sist. de tablas de amortización, Sist. de cuentas por pagar, etc.)

SIMBOLO	NOMBRE	SIGNIFICADO
	**INICIO o FIN**	*Para indicar el inicio o el fin del diagrama de flujo.*
	**PROCESO u OPERACION**	*Para especificar una operación aritmética o la asignación de un valor.*
	**LECTURA o ENTRADA**	*Espera la entrada de datos que va alimentar.*
	**IMPRESION**	*Nos indica los resultados que vamos a presentar.*
	**FLECHAS DE SEGUIMIENTO**	*Es el flujo del proceso.*
	**DECISION**	*Decide mediante una condición si es verdadera o falsa.*

SIMBOLO	NOMBRE	SIGNIFICADO
	**SUBRUTINA o SUBPROGRAMA**	*Especifica una serie de instrucciones que tendrá que procesar, dentro del mismo programa.*
	CONECTOR	*Para conectar diferentes símbolos del diagrama en una misma página.*
	**CONECTOR DE PAGINA**	*Para conectar diferentes símbolos del diagrama, que se encuentran en distintas páginas.*
	**CICLADOR o REPETIDOR DE PROCESOS (CONTADOR)**	*Sirve para repetir uno o varios procesos, dependiendo de una condición.*

Los símbolos anteriores son los utilizables para el diagrama de flujo, más adelante, estudiaremos que instrucciones **BASIC** *les corresponde a cada uno de ellos.*