

REFORMA ACADÉMICA DEL NIVEL MEDIO SUPERIOR

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Secretaría Académica

M1

Texto
y
Guía del Alumno

COMPUTACIÓN, SEGUNDA EDICIÓN 1996

Computación

C

6
0
6

QA76
.5
U530
1998
v.1

AA76
.5
U530
1996
V.1

0120-24860



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



ÍNDICE

	página
PRESENTACIÓN	v
PRÓLOGO	vii
INTRODUCCIÓN	ix
I INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS	
1.- RESUMEN HISTÓRICO DE LAS COMPUTADORAS	3
A.- Antecedentes	3
B.- Generaciones	5
2.- LA COMPUTADORA	6
A.- Definición	6
B.- Clasificación	7
C.- Componentes básicos	8
a.- Dispositivos de entrada	8
b.- Dispositivos de salida	11
c.- La unidad central de proceso	13
d.- Dispositivos de almacenamiento secundario	16
EJERCICIO GENERAL	18
II SISTEMA OPERATIVO	
1.- INTRODUCCIÓN	23
A.- Sistema operativo y sus funciones	23
B.- Diferentes sistemas operativos	24
C.- Versiones del sistema operativo MS-DOS	25
2.- INICIALIZACIÓN DEL SISTEMA	25
3.- FUNCIONES DE LAS TECLAS EN EL MS-DOS	27
A.- Teclas de control	27
B.- Teclas de edición	28

4.- IMPORTANCIA DE LOS ARCHIVOS EN EL MS-DOS	29
A.- Concepto de archivo	29
B.- Nombre de los archivos	30
5.- USO DEL MS-DOS	31
A.- Proceso de carga	31
B.- Comandos	33
C.- Caracteres comodines	34
D.- Usos de comandos	35
6.- LOS DIRECTORIOS EN EL MS-DOS	49
A.- Estructura de un directorio	49
B.- Comandos para manipular directorios	51
EJERCICIO GENERAL	56
III INTRODUCCIÓN A LA PROGRAMACIÓN	
1.- INTRODUCCIÓN	59
A.- Definición de programa	59
B.- Programas del sistema operativo	60
C.- Programas en lenguaje de programación	60
D.- Programas de aplicación	60
2.- METODOLOGÍA DE LA PROGRAMACIÓN	66
A.- Análisis del problema o tarea	67
B.- Algoritmos	71
C.- Diagrama de flujos	71
D.- Prueba de escritorio	74
G.- Codificación	81
H.- Depuración	81
EJERCICIO GENERAL	83

PRESENTACIÓN

IV PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN	95
A.- Conceptos de BASIC	95
2.- ESTRUCTURA DE UN PROGRAMA	96
A.- Número de línea	96
B.- Instrucciones	97
C.- Documentación del programa	100
D.- Usos de constantes	101
E.- Asignación de variables	102
F.- Terminación de línea	103
3.- COMO CARGAR BASIC	104
4.- PROGRAMACIÓN BASIC	106
A.- Modos de trabajar en BASIC	106
B.- Instrucciones elementales	108
C.- Programación secuencial	111
Ejercicios	114
D.- Programación condicional	119
Ejercicios	123
E.- Programación con lazos o programas repetitivos	128
Ejercicios	131
F.- Programación cíclica	136
Ejercicios	140
EJERCICIO GENERAL	145
Para terminar	146
BIBLIOGRAFÍA	147

PRESENTACIÓN

La Universidad Autónoma de Nuevo León se ha caracterizado, en el curso de su historia, por pretender alcanzar siempre la excelencia académica en todos los niveles. En la época presente y, debido a las exigencias motivadas por los trascendentales cambios en el ámbito mundial, resulta necesario redoblar esfuerzos para lograr la máxima superación; es por ello que ha iniciado una nueva etapa en la educación media superior.

La reforma educativa, cuya meta es preparar estudiantes con un alto grado de calidad cultural, ha propiciado la elaboración de una serie de textos que despierten el interés y la creatividad de los preparatorianos.

Este libro de computación ha sido diseñado tratando de que sirva como una herramienta útil en la introducción al mundo de las computadoras; así como el obtener la gran meta estudiantil fijada por nuestra máxima casa de estudios.

Cabe resaltar la gran ayuda proporcionada por todos los maestros de esta área del saber humano, ya que gracias a sus sugerencias se logró llevar a feliz término este texto, pretendiendo sea de utilidad para el desarrollo del programa.

COMITÉ DE COMPUTACIÓN

Ing. Felipe Rojas Patlán	Preparatoria N° 2
Prof. Ricardo H. Álvarez Charles	Preparatoria N° 7
Ing. Raúl Gallegos Cerda	Preparatoria N° 15
Ing. Araceli Gpe. Álvarez Carvajal	Preparatoria N° 16

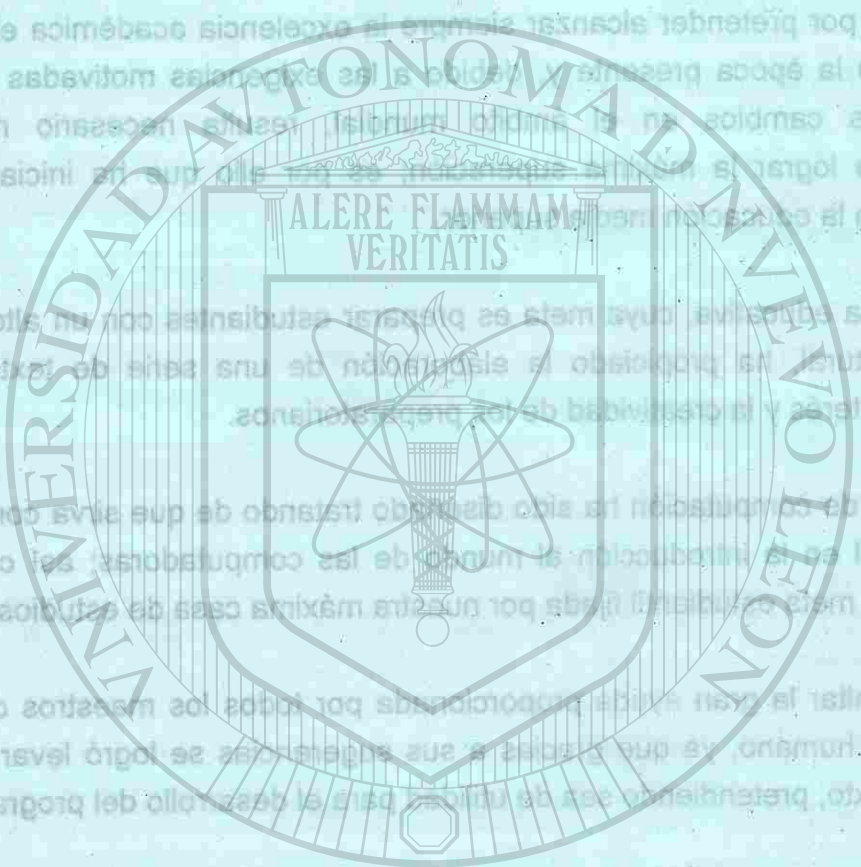


La Universidad Autónoma de Nuevo León se ha caracterizado, en el curso de su historia, por pretender alcanzar siempre la excelencia académica en todos los niveles. En la época presente, debido a las exigencias motivadas por los trascendentales cambios en el mundo, resulta necesario redoblar esfuerzos para lograrlos, a fin de que se inicie una nueva etapa en la educación superior.

La reforma educativa que se está realizando en el país, con un alto grado de calidad cultural, ha propiciado la elaboración de una serie de textos que despiertan el interés y la creatividad de los estudiantes.

Este libro de computación ha sido elaborado tratando de que sirva como una herramienta útil en la introducción al mundo de las computadoras, así como el obtener la gran meta de obtener el título por nuestra máxima casa de estudios.

Debe resaltar la gran responsabilidad por todos los maestros de esta área del saber humano, ya que a través de sus enseñanzas se logró llevar a feliz término este texto, pretendiendo ser de utilidad para el desarrollo del programa.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

Ing. Felipe Rojas Estela
 Ing. Ricardo A. Álvarez Cisneros
 Ing. Paul Collopy
 Ing. Ansel Gde. Álvarez Cervantes

Preparatoria N.º 2
 Preparatoria N.º 7
 Preparatoria N.º 15
 Preparatoria N.º 16

DIRECCIÓN GENERAL DE BIBLIOTECAS

Comité de Computación



Los integrantes del comité de computación, atendiendo al llamado de la secretaría académica de la Universidad Autónoma de Nuevo León, hemos elaborado el siguiente libro, el cual será utilizado como texto por los alumnos de preparatoria.

Dicho libro es producto, tanto de la investigación documental, como de la experiencia docente de cada uno de los miembros del comité. Este trabajo lo realizamos tomando en cuenta las recomendaciones de todos los compañeros maestros que desempeñan su labor magisterial dentro del área de computación.

Queremos agradecer la valiosa cooperación brindada por el Ing. Sergio Eduardo González González en la elaboración de este texto.

Deseamos que con la experiencia y dedicación de cada uno de los maestros del área de computación en la Universidad, alcancemos la meta fundamental marcada en la reforma académica: Preparar a nuestros alumnos para la competitividad en la vida.

ATENTAMENTE

Los integrantes del comité de computación, atendiendo al llamado de la Secretaría Académica de la Universidad Autónoma de Nuevo León, hemos elaborado el siguiente libro, el cual será utilizado como texto por los alumnos de preparatoria.

Dicho libro es producto tanto de la experiencia docente de cada uno de los miembros del comité, como de la experiencia tomada en cuenta las recomendaciones de todos los compañeros maestros que desempeñan su labor magisteral en el área de computación.

Queremos agradecer la valiosa colaboración brindada por el Sr. Sergio Eduardo González González en la elaboración de este texto.

Desearnos que con la experiencia y dedicación de cada uno de los maestros del área de computación en la Universidad, alcancemos la meta fundamental marcada en la reforma académica: Preparar a nuestros alumnos para la competitividad en la vida.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

Para el alumno:

Como tú sabes, las computadoras tienen una gran importancia en la época contemporánea, pues el desarrollo científico y tecnológico en las últimas décadas ha sido vertiginoso.

Con el presente texto pretendemos que conozcas lo básico del mundo de las computadoras; para ello primeramente te ofreceremos una breve reseña histórica sobre los cambios que ha experimentado esta herramienta de trabajo; enseguida te llevaremos al manejo de la máquina mediante una serie de programas que te permitirán interactuar con ella; en otro apartado conocerás la lógica y procedimientos para la solución de problemas y, por último, aprenderás el uso del lenguaje de programación BASIC, que te facilitará la solución de dichos problemas en otras materias: Matemáticas, Física, Química, etc.

Cada una de las cuatro unidades que conforman el texto, cuenta con una serie de ejercicios que te permitirán recrearte y reafirmar tus conocimientos.

Desearnos que este libro aunado a otros que te recomendará tu maestro, sean para ti el inicio de un conocimiento que poco a poco irás profundizando, pero recuerda: la computadora sólo es una herramienta de trabajo, siempre necesitará de un ser humano que la maneje.

Te deseamos éxito en el curso que ahora comienzas. ¡Adelante!

Los autores

INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS

Para el alumno:

Como tú sabes, las computadoras han revolucionado en la época contemporánea, pues el mundo actual vive en las últimas décadas de la vida tecnológica.

Con el presente libro pretendemos acercar al mundo de las computadoras, pero principalmente al mundo de las ciencias exactas y naturales, a través de los experimentos de laboratorio. En este libro se pretende que el alumno pueda comprender los fundamentos de la programación en lenguaje de programación BASIC, para la solución de algunos problemas en otros lenguajes como Pascal, Fortran, etc.

Cada una de las cuatro unidades que componen el libro, termina con una serie de ejercicios que te permitirán repasar y afianzar los conocimientos.

Después de este libro cuando quieras recomendar tu maestro, sean para el inicio de un conocimiento que poco a poco vas profundizando, pero recuerda que la computadora sólo es una herramienta de trabajo, siempre necesitarás de un ser humano que la maneje.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS

RESUMEN HISTÓRICO DE LAS COMPUTADORAS

1. ANTECEDENTES

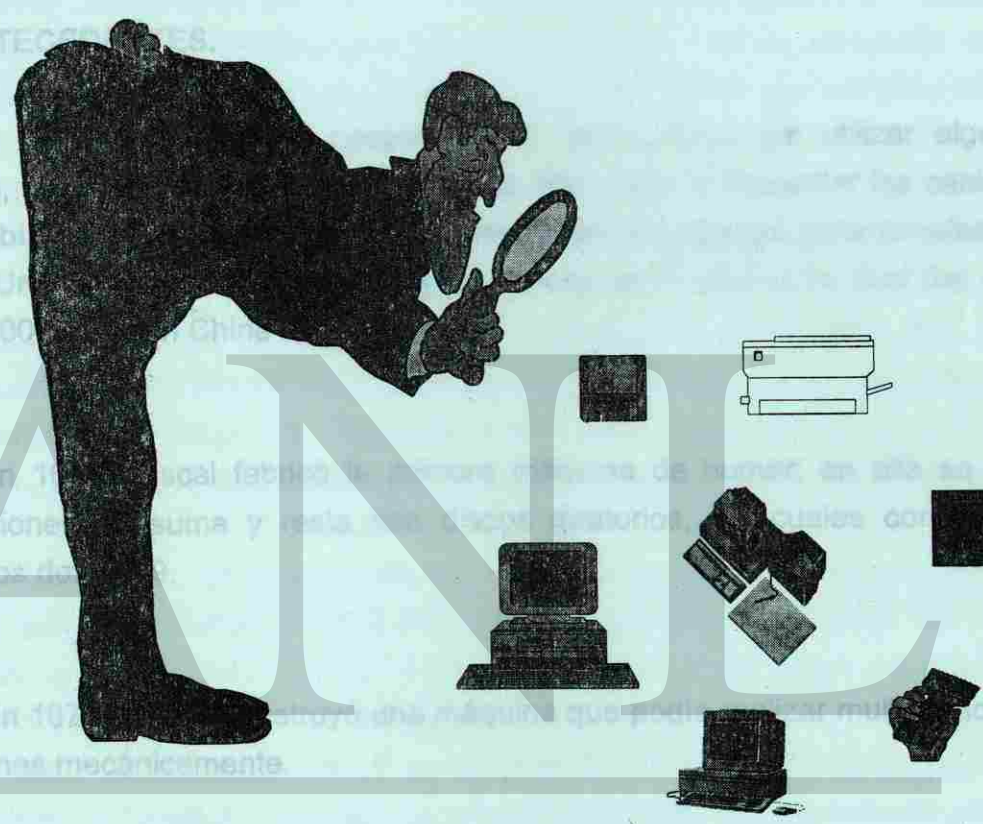
Al hacer historia de las computadoras, debemos hacer un recorrido por los siglos que han precedido a la aparición de las computadoras modernas. En el año 1642, Blaise Pascal inventó la primera calculadora mecánica, que permitía hacer operaciones aritméticas de suma y resta. En 1822, Charles Babbage diseñó la máquina analítica, que era una máquina mecánica que podía realizar cualquier operación aritmética y lógica.

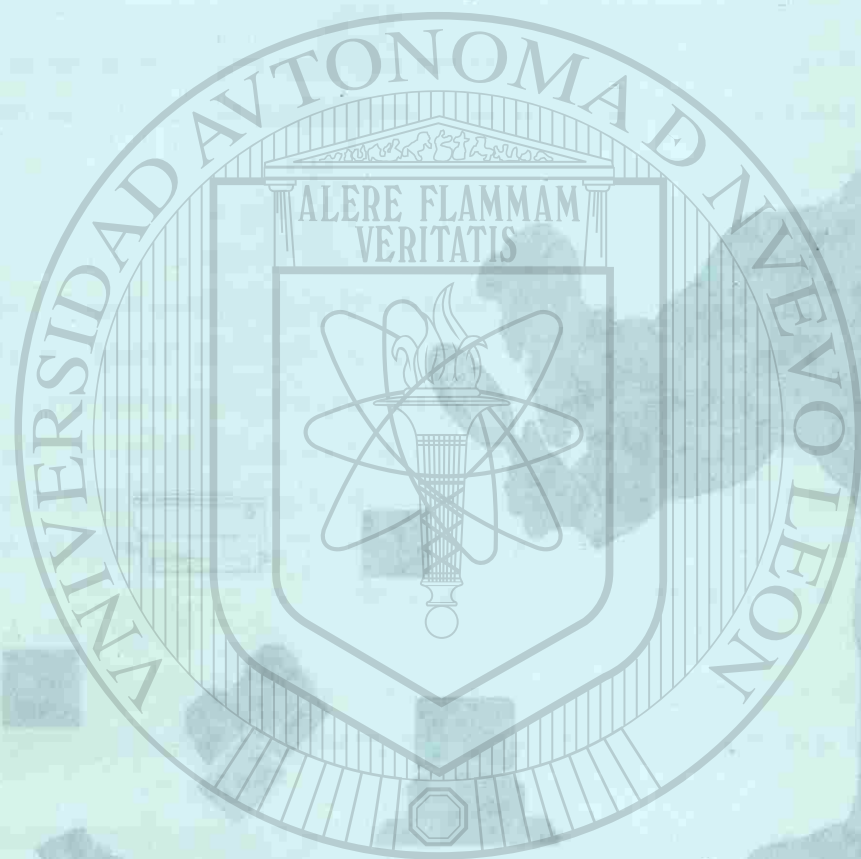
En 1842, Charles Babbage diseñó la máquina analítica, que era una máquina mecánica que podía realizar cualquier operación aritmética y lógica. En 1890, Herman Holerith inventó la tarjeta perforada, que era una forma de representar datos de manera que una máquina mecánica pudiera leerlos.

En 1904, John V. Atanasoff inventó el primer computador electrónico, que era un dispositivo que podía realizar operaciones aritméticas y lógicas. En 1941, Colossus fue el primer computador electrónico de propósito general, que era una máquina que podía realizar operaciones aritméticas y lógicas.

INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS

El computador es un dispositivo que puede realizar operaciones aritméticas y lógicas. El computador moderno es un dispositivo que puede realizar operaciones aritméticas y lógicas. El computador moderno es un dispositivo que puede realizar operaciones aritméticas y lógicas.





UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS
INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS

UNIDAD I

INTRODUCCIÓN AL MUNDO DE LAS COMPUTADORAS

1.- RESUMEN HISTÓRICO DE LAS COMPUTADORAS.

A.- ANTECEDENTES.

Al tener el hombre la necesidad de contar, tuvo que utilizar algo para hacerlo, y comenzó con los dedos, piedras, etc.; pero al aumentar las cantidades que debía contabilizar inventó instrumentos más sofisticados para satisfacer esa tarea. Uno de los más antiguos que se conoce es el **ábaco**, el cual fue creado entre 3000 A.C., en China o Egipto.

En 1642 Pascal fabricó la primera máquina de sumar; en ella se hacían operaciones de suma y resta con discos giratorios, los cuales contaban con números del 0 al 9.

En 1673 Leibniz construyó una máquina que podía realizar multiplicaciones y divisiones mecánicamente.

En 1777 Charles Mahon inventó la primera máquina lógica; era un aparato que solucionaba los silogismos y preguntas elementales de probabilidad. La máquina de Mahon es el principio de los componentes lógicos en las modernas computadoras.

En 1804 Joseph-Marie Jacquard, inspirado en un instrumento musical que tocaba mediante un rollo de papel con perforaciones, adaptó la idea a tarjetas que usó en un telar, en el cual se utilizaban dichas tarjetas a modo de memoria para tejer las figuras en las telas.

En 1822 Charles Babbage termina su **máquina diferencial**, la cual podía transformar cálculos a simples tablas. Este aparato, construido con especificaciones muy exactas, era un complejo ensamble de ruedas, engranajes y demás piezas. Babbage marcó la pauta al diseñar su **máquina analítica**, que sería un dispositivo de propósito general capaz de ejecutar cualquier cálculo matemático. Los planos de este aparato fueron los primeros en dar el concepto de una máquina que puede ejecutar el tipo de cálculos que ahora se consideran el corazón de la computadora. Babbage nunca realizó su proyecto, pero su diseño tuvo influencia en las computadoras digitales modernas. La máquina analítica finalmente fue construida por un equipo de ingenieros en 1989, más de cien años después de la muerte de Babbage en 1871. Por las aportaciones de este científico se le conoce como el "Padre de las computadoras modernas".

En 1854 el inglés George S. Boole publicó su "Álgebra booleana", cuyo sistema reducía los argumentos lógicos a tres operadores básicos (and, or y not) y el manejo de los dígitos 1 y 0. Por el desarrollo del álgebra booleana, Boole es considerado como el "Padre de la teoría de la información".

En 1886 la primera máquina que usó tarjetas perforadas como sistema de entrada de datos fue inventada por el Dr. Herman Hollerith. Esta máquina fue desarrollada para usarse en el censo de los Estados Unidos en 1890, siendo censada una población de 62'979,766 personas; después del censo, Hollerith fundó la Tabulating Machines Company, la cual fusionó y adquirió otras compañías formando la International Business Machines, la actual IBM.

En 1919 el primer circuito con relevadores fue desarrollado por los americanos W.H. Eccles y F.W. Jordan. Era un circuito que tenía sólo dos estados, los cuales eran intercambiados; éstos constituyen la base del formato binario de almacenamiento de las computadoras actuales.

B.- GENERACIONES.

1a. generación.- La primera computadora comercial, llamada UNIVAC I, se fabricó en 1951 para la oficina del censo de los Estados Unidos. Esta máquina utilizaba válvulas de vacío (bulbos) y podía ejecutar unas mil instrucciones por segundo; su campo de aplicación era exclusivamente científico y militar. Algunos fabricantes de esta primera generación de computadoras fueron IBM, HONEYWELL, BURROUGHS y RCA.

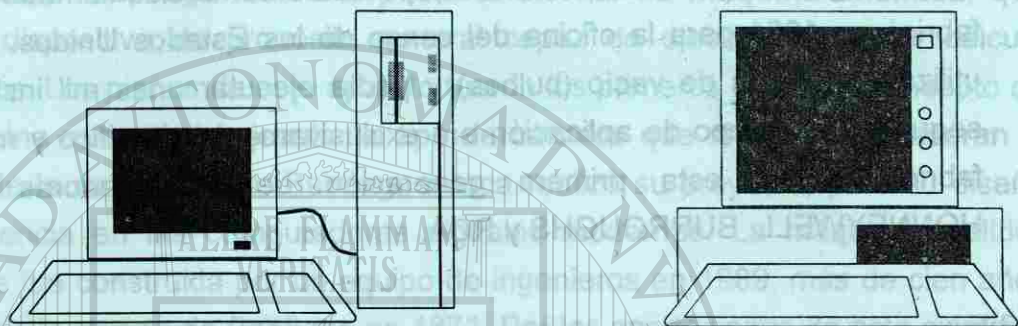
2da. generación.- Se creó con la aparición del transistor en la electrónica, el cual sustituyó la válvula de vacío. Los fabricantes de estas computadoras fueron: IBM, MINNEAPOLIS-HONEYWELL, UNIVAC y CONTROL DATA CORPORATION.

3a. generación.- Llega al mercado después de 1965; se caracteriza por tener circuitos monolíticos integrados, terminales de tiempo compartido, multiprogramación, procesamiento de tiempo real y mayor miniaturización del equipo.

4ta. generación.- Comienza cuando en 1970 nace el microprocesador, que viene a ser en realidad la unidad central de proceso de una computadora, que tiene como características principales la miniaturización de los equipos, gran capacidad de entrada y salida de información, mayor velocidad de proceso, más facilidad para programar, más cantidad de software específico (paquetes), etc.

Con el microprocesador nació la revolución de las microcomputadoras, ya que se comenzaron a usar en gran cantidad y, al final de la década de los setentas, numerosas compañías se lanzaron a conquistar el mercado de dichas máquinas.

2.- LA COMPUTADORA.



A.- DEFINICIÓN.

Dentro del contexto de la tecnología moderna, es un aparato electrónico capaz de interpretar y ejecutar órdenes de entrada, salida, procesamiento y operaciones lógicas.

La computadora, llamada también ordenador o procesador, es la "inteligencia" de un sistema de computación. Las computadoras personales son llamadas PC y tan pronto te sientes frente a ella, te convertirás en parte de un sistema de información, que consiste en personas (liveware o humanware), equipo (hardware), programas (software),

El **hardware** o equipo comprende todos los dispositivos físicos que conforman una computadora; pueden incorporarse en una sola unidad o estar separados. El hardware realiza cuatro tareas importantes en el sistema de computación: **entrada, procesamiento, salida y almacenamiento secundario.**



El **software** o programas son las instrucciones codificadas electrónicamente que dirigen a la computadora para realizar ciertas tareas; sin el software, la computadora sería un conjunto de medios sin utilizar; al cargar los programas en una computadora, la máquina actuará como si recibiera una educación instantánea: de pronto "sabe" cómo pensar y cómo manejar la información. Existen tres categorías generales de software, de las cuales se hablará mas ampliamente en unidades posteriores: lenguajes de programación, sistemas operativos y programas de aplicación (paquetes).

B.- CLASIFICACIÓN DE LAS COMPUTADORAS.

Analógicas : Son aquellas que manejan señales eléctricas, se aplican en problemas de simulación, su programación está cableada en los circuitos que la integran; operan con datos representados en una forma continua, en términos de una cantidad física medible (voltaje, temperatura, revoluciones por minuto, etc.).

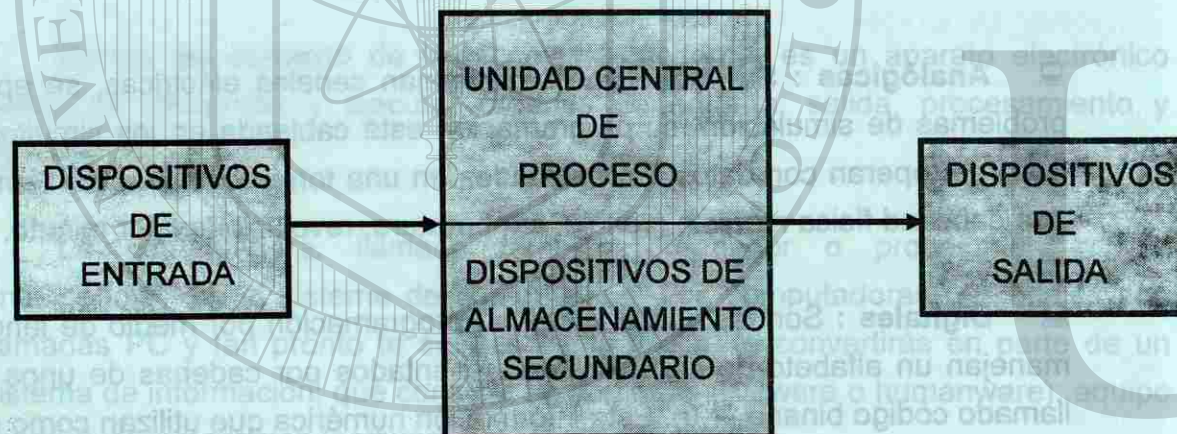
Digitales : Son las que admiten su programación por medio de lenguajes y manejan un alfabeto de caracteres representados por cadenas de unos y ceros llamado código binario (1,0). Esta información numérica que utilizan como entrada, se procesa a través de operaciones aritméticas para producir salidas.

Híbridas: Éstas tienen las características de los dos tipos de computadoras anteriores. La entrada de datos está controlada por un convertidor analógico-digital; la información es procesada por un ordenador digital y la salida se canaliza por un convertidor digital-analógico.

C.- COMPONENTES BÁSICOS.

Un sistema de computación se compone básicamente de unidades de entrada, de salida, unidad central de proceso y dispositivos de almacenamiento secundario y en éste último es donde la computadora coloca la información para que pueda recuperarse más tarde por el usuario.

Los dispositivos de entrada, salida y almacenamiento secundario se conocen como **periféricos**, porque están fuera del procesador central.



a.- Dispositivos de entrada

En los dispositivos de **entrada**, el usuario ingresa los datos e instrucciones en la computadora, utilizando algunos como el teclado, el lápiz óptico, el lector de código de barras, dispositivos de entrada de voz, pantallas sensibles al tacto, scanners, etc.



El **teclado** es muy semejante al de una máquina de escribir, sólo que éste cuenta, además, con algunas teclas especiales para el uso de la computadora; también tiene un teclado numérico que se asemeja al de una calculadora y las que se conocen como teclas de función.

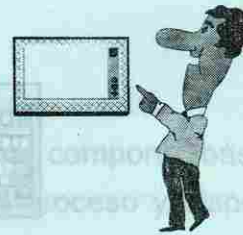


El **lápiz óptico** (pluma óptica) se asemeja a una pluma grande conectada a un cordón eléctrico y requiere de software especial; trabaja como una pluma ordinaria, utiliza una luz donde el usuario toca la pantalla del monitor con ella y puede seleccionar comandos de los programas o trazar imágenes. Se usa para registrar operaciones de puntos de venta en tiendas departamentales, en diseño gráfico auxiliado por computadora, etc.

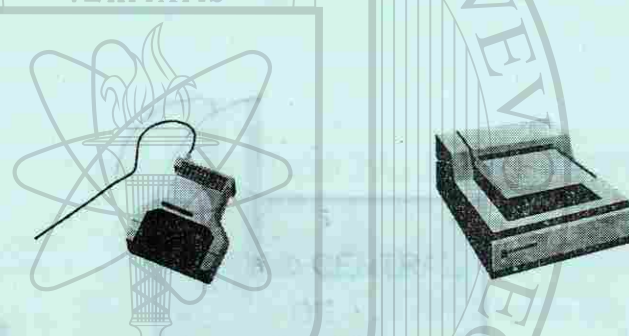


Los **lectores de código de barras** se utilizan como parte de lo que se conoce como terminal punto de venta. Son dispositivos fotoeléctricos que leen las barras verticales que conforman un código y ayudan a la computadora a identificar el código del producto (su descripción, precio, etc.), y así como también controlar el inventario entre otras funciones.

C.- COMPONENTES BÁSICOS.



Las **pantallas sensibles al tacto** permiten dar comandos a la computadora tocando ciertas partes de la pantalla y trabajan a base de ventanas; sólo que muy pocos programas trabajan con ellas; por lo regular se usan en tiendas para indicar a los clientes sobre la ubicación de algunas de las partes en que se encuentran los departamentos de un centro comercial.



Los **scanners o rastreadores** (también se les conoce como digitalizadores) convierten textos, fotografías y gráficas a color o blanco y negro a una forma que puede leer una computadora, transfiriéndola a la misma; la mayoría viene con software que permite ajustar el tamaño de la imagen y los niveles de contraste y brillo. Una vez que la imagen se digitaliza, puede ser modificada o guardada en un dispositivo de almacenamiento secundario.



El **mouse** (ratón), con este dispositivo el usuario desplaza el mouse a lo largo y ancho de una superficie rugosa para mover un apuntador en la pantalla del monitor y poder seleccionar algunas funciones como textos, gráficos, etc.; el mouse no sustituye por completo el teclado, aunque en algunas ocasiones es más rápido; no puede usarse con todos los tipos de programas.

b.- Dispositivos de salida

Los dispositivos de salida comunican los resultados al usuario, desplegando la información en un monitor o mediante una impresora. Estos dispositivos convierten la información que sale de una computadora en imágenes en la pantalla y en diversos tipos de impresiones. Una amplia variedad de tecnologías se utiliza en las computadoras para darnos la salida.



El **monitor** más utilizado en la mayoría de las computadoras es CRT (tubo de rayos catódicos); es una pantalla que tiene fósforo y hace que los electrones, pasados a través de un conjunto de electromagnetos llamado yugo (o cañón), sean desviados hacia la pantalla y, al golpear contra ella, enciendan un punto llamado " **pixel** " (contracción del inglés de **picture element**), que al unir varios de ellos dan forma a una letra o un dibujo. Existen dos tipos de monitores: monocromáticos (blanco y negro) y cromáticos (de color).



Las **impresoras de impacto** son de matriz de puntos, la más común, tiene una cabeza de impresión móvil con varias puntillas cuyo número varía entre 7 y 24, que golpean una cinta entintada para formar caracteres por medio de puntos (de ahí su nombre "matriz de puntos") en el papel. Entre más puntillas tenga la cabeza mejor será la calidad de impresión. La velocidad varía entre los 80 y 450 caracteres por segundo (cps). Las impresoras con dispositivos de impacto pueden utilizar papel carbón y generar varias copias a la vez; mientras que las que no utilizan esta tecnología generan sólo una copia.



Las impresoras láser crean las letras mediante un proceso de fotocopiado, un rayo láser traza los caracteres en un tambor fotosensible, después fija el tóner (solución parecida a la tinta) al papel utilizando calor y sale la impresión; su resolución es muy alta (cerca de 300 puntos por pulgada cuadrada), su velocidad es de 4 a 16 páginas por minuto.



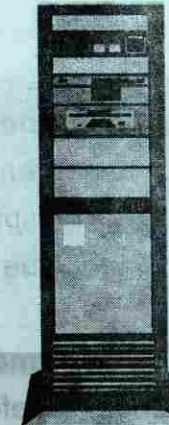
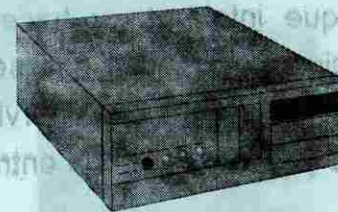
Las impresoras de inyección de tinta cuentan con un cartucho desechable que emite chorros de tinta hacia el papel formando los caracteres; las mejores tienen impresión de calidad de carta; su velocidad oscila en un rango de 80 a 300 cps.



Las graficadoras utilizan una pluma que se mueve (en algunas se desplaza también el papel) y crean gráficos de alta calidad, como los utilizados en dibujos arquitectónicos. Existen diseños de tambor giratorio o de placa plana; las más modernas pueden graficar en colores.

c.- La unidad central de proceso

El procesamiento de datos se realiza en la unidad central de proceso, más conocida por sus siglas en inglés CPU (Central Processing Unit), ésta unidad viene a ser el "cerebro" de la computadora.



El es el responsable de controlar el flujo de datos y la ejecución de las instrucciones de los programas sobre los mismos; realiza operaciones aritméticas, además compara valores numéricos o secuencias de caracteres, se divide en tres partes:

1a.- La memoria primaria es la memoria de la computadora que contiene el sistema operativo, las instrucciones para manejar datos y los datos mismos, así como cierta información que necesita la computadora. Esta memoria primaria puede ser de dos tipos:

- ☑ ROM (Read Only Memory).- Memoria sólo de lectura, en la cual se almacenan cierto tipo de programas que necesita la computadora; no puede ser cambiada por el usuario, ya que viene desde el fabricante y es una memoria permanente.

- ☑ **RAM (Random Access Memory).**- Memoria de acceso aleatorio, es donde se almacenan ciertos programas que son indispensables para la computadora; sin embargo, el usuario puede cambiar la información. La capacidad de la RAM afecta la forma en que se desarrollan los programas y la cantidad de datos que pueden procesarse. Es una memoria volátil y a menos que la información se guarde en un dispositivo de almacenamiento secundario como discos, ésta se perderá al desconectarse la energía eléctrica de la computadora.

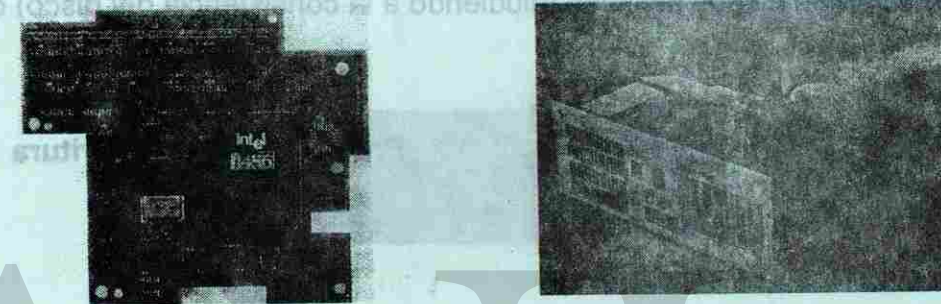
2da.- La Unidad de Control es la que interpreta y hace ejecutar las instrucciones que están en la memoria principal y garantiza que se ejecuten de acuerdo a la secuencia especificada en el programa; también supervisa el proceso de las demás unidades que componen el CPU y las unidades de entrada y salida.

3a.- La Unidad Aritmética-Lógica es donde ocurre el procesamiento real de los datos; es donde se efectúan todas las operaciones aritméticas como son: suma, resta, multiplicación y división, al igual que las operaciones lógicas como: igual (=), menor que (<), mayor que (>), menor o igual que (<=), mayor o igual que (>=), diferente que (< >), etc.

- ☑ Las computadoras digitales realizan las operaciones en notación binaria, en la cual **1** significa activado y **0** significa desactivado; esto es similar a encendido o apagado en un aparato eléctrico. El sistema binario permite a la computadora representar números y realizar operaciones aritméticas; así mismo también puede representar letras del alfabeto y otros símbolos. La unidad básica en este sistema es el **byte**, que está compuesto por ocho **bits** (binary digit, dígito binario 1 ó 0); un byte representa un carácter, cada carácter tecleado se traduce a una combinación de dígitos binarios.

- ☑ La mayoría de las computadoras utilizan el **American Standard Code for Information Interchange (ASCII)** para representar datos. Hay que hacer notar que las letras mayúsculas y las minúsculas tienen diferente código, ya que para la computadora **A (01000001)** no significa lo mismo que **a (01100001)**.

- ☑ Los microprocesadores varían considerablemente su rapidez de operación dependiendo del modelo del procesador. La velocidad se incrementa con su tamaño en bits, ya que por ejemplo una computadora de 16 bits será más rápida que una de 8 bits. La velocidad es medida en millones de ciclos por segundo o MegaHertz (MHz), indica qué tan rápidamente una computadora puede procesar la información, a continuación mencionaremos los diferentes modelos del procesador:

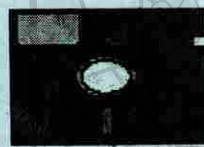


- ☑ En 1981 la IBM-PC se desarrolla con el chip 8088 de Intel y las unidades de disco, los componentes electrónicos y el software son proporcionados también por otras compañías.
- ☑ En 1984 se produce el chip Intel 80286 y se presenta al mercado; también la IBM AT, que usa el citado chip y es mucho más rápida que la PC original. La Apple lanza al mercado la Macintosh con discos de 3.5 pulgadas, que se comercializa como una computadora para la "demás gente".
- ☑ En 1986 el chip de Intel 80386 sale al mercado y es todavía más rápido. En 1987 se introduce la familia de las computadoras IBM PS/2. En 1988 aparece el chip de Intel 80386SX.
- ☑ En 1989 aparece el chip 80486 con un coprocesador matemático incluido, que ejecuta más instrucciones al doble de la velocidad del chip 80386.
- ☑ A principios de los 90's se introduce el procesador Pentium, que trabaja mucho más rápido que el 80486.

d.- Dispositivos de almacenamiento secundario

Existen dispositivos de almacenamiento que, a diferencia de la memoria RAM, pueden conservar información y quedar permanentemente guardados; se les llama dispositivos de almacenamiento secundario.

Los **disquetes** (diskettes) o discos son actualmente el medio de almacenamiento secundario más popular usado en las computadoras; son llamados también discos flexibles (aludiendo a la consistencia del disco) o floppy.

**Protector contra escritura**

Disco flexible de 5 ¼

Baja Densidad (DD) capacidad: 360 Kbytes

Alta Densidad (HD) capacidad: 1.2 Mbytes

**Protector contra escritura**

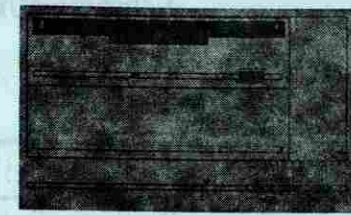
Disco flexible de 3 ½

Baja Densidad (DD) capacidad: 720 Kbytes

Alta Densidad (HD) capacidad: 1.44 Mbytes

Nota: 1 Kbyte es igual a 1024 Bytes

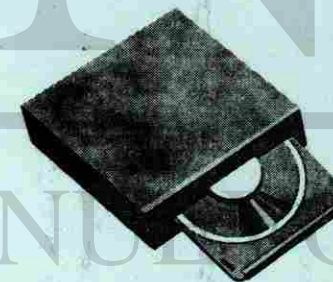
1 Mbyte es igual a 1024 Kbyte



Los **Drives** son dispositivos donde se insertan los discos flexibles (disquetes). Existen también para los discos compactos conocidos como CD-ROM, los cuales cada vez tienen mayor aceptación.



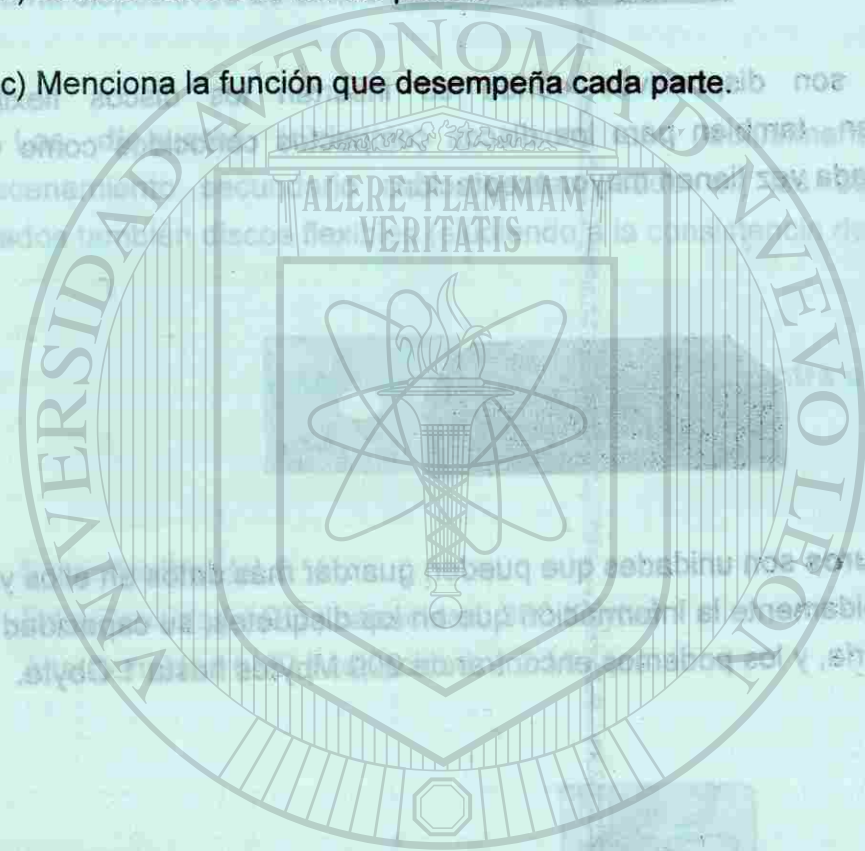
Los **discos duros** son unidades que pueden guardar más datos en ellos y se puede leer más rápidamente la información que en los disquetes, su capacidad de almacenamiento varía, y los podemos encontrar de 200 Mbytes hasta 1 Gbyte.



Un **CD-ROM** tiene una capacidad de almacenar hasta 5 Gbytes, aproximadamente 5 mil millones de caracteres, la información al igual que las grabaciones musicales en CD; es por medio de un láser, que graba la información en los discos como burbujas, que pueden ser leídas por otro láser más tarde, la información grabada no puede ser alterada o borrada, por esto se conocen como CD-ROM (Compact Disk - Read Only Memory). La gran capacidad de almacenamiento y la seguridad que ofrece este sistema hacen de los CD-ROM un medio eficaz para aplicaciones de archivos, respaldo y multimedia.

☺ EJERCICIO GENERAL:

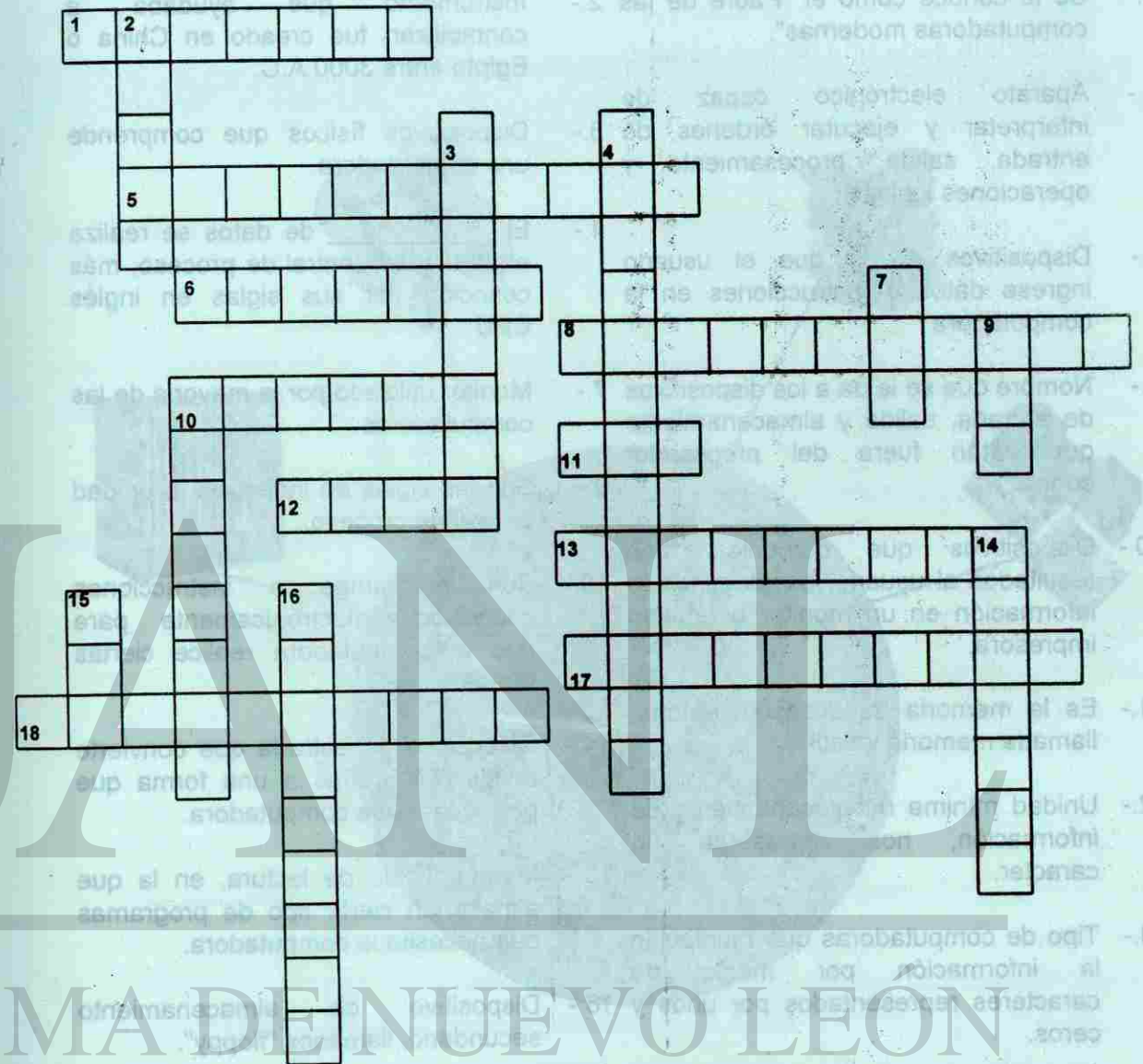
- 1.- a) Realiza el dibujo de una computadora personal.
- b) Señala cada una de sus partes.
- c) Menciona la función que desempeña cada parte.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

2.- Resuelve el siguiente crucigrama:



Instrucciones en la siguiente página.

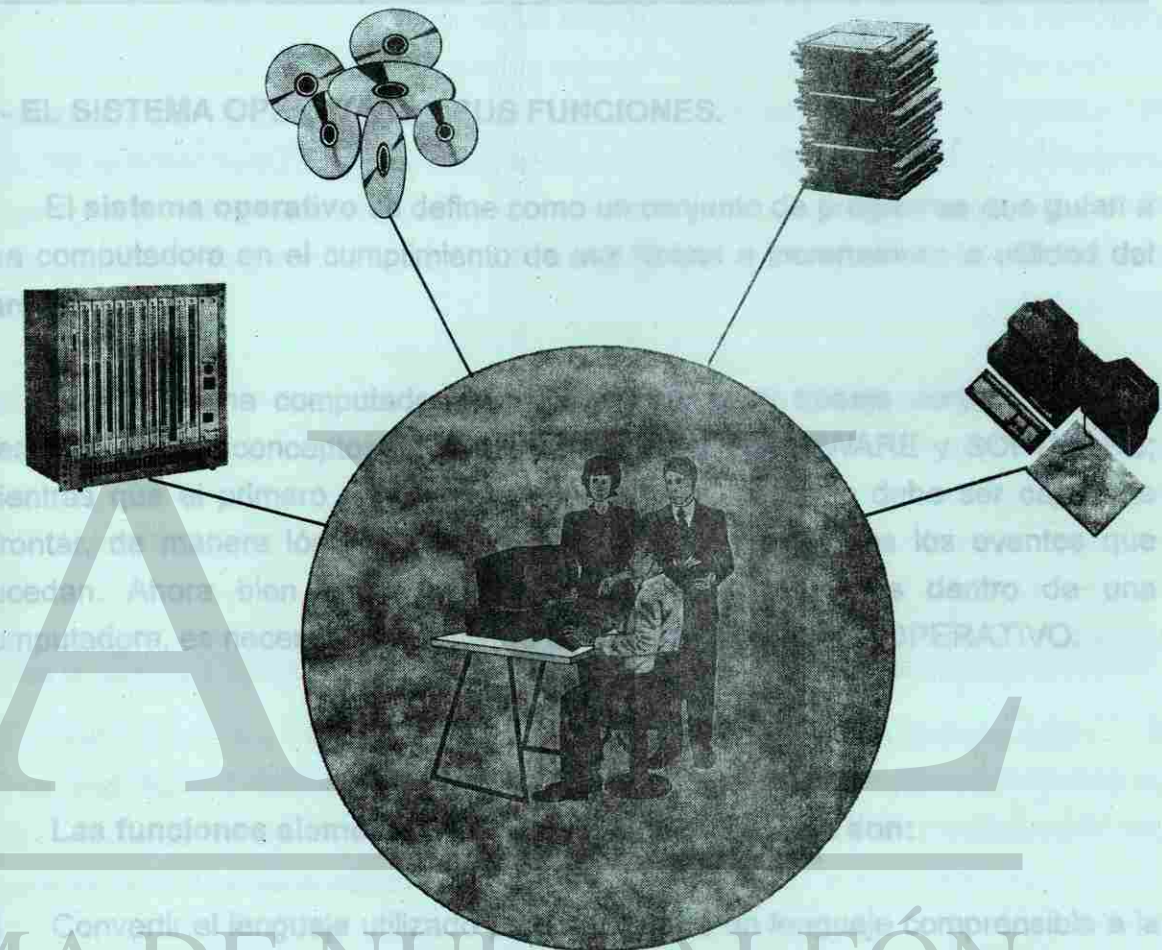
EJERCICIO CRUCIGRAMA DE LA PRIMERA UNIDAD

HORIZONTALES

- 1.- Se le conoce como el "Padre de las computadoras modernas".
- 5.- Aparato electrónico capaz de interpretar y ejecutar órdenes de entrada, salida, procesamiento y operaciones lógicas.
- 6.- Dispositivos en la que el usuario ingresa datos e instrucciones en la computadora.
- 8.- Nombre que se le da a los dispositivos de entrada, salida y almacenamiento que están fuera del procesador central.
- 10.- Dispositivos que comunican los resultados al usuario, desplegando la información en un monitor o en una impresora.
- 11.- Es la memoria de acceso aleatoria, llamada memoria volátil.
- 12.- Unidad mínima de almacenamiento de información, nos representa un carácter.
- 13.- Tipo de computadoras que manipulan la información por medio de caracteres representados por unos y ceros.
- 17.- Tipo de computadoras que manipulan la información por medio de señales eléctricas, se aplican en problemas de simulación.
- 18.- Dispositivos que utilizan papel para dar salida a la información.

VERTICALES

- 2.- Instrumento que ayudaba a contabilizar, fue creado en China o Egipto entre 3000 A.C.
- 3.- Dispositivos físicos que comprende una computadora.
- 4.- El _____ de datos se realiza en la unidad central de proceso, más conocida por sus siglas en inglés CPU.
- 7.- Monitor utilizado por la mayoría de las computadoras.
- 9.- Son las siglas en inglés de la unidad central de proceso.
- 10.- Son programas o instrucciones codificadas electrónicamente para que la computadora realice ciertas tareas.
- 14.- Dispositivo de entrada que convierte textos, fotografías a una forma que puede leer una computadora.
- 15.- Memoria sólo de lectura, en la que almacenan cierto tipo de programas que necesita la computadora.
- 16.- Dispositivo de almacenamiento secundario, llamados "floppy".

UNIDAD II
SISTEMA OPERATIVO

SISTEMA OPERATIVO

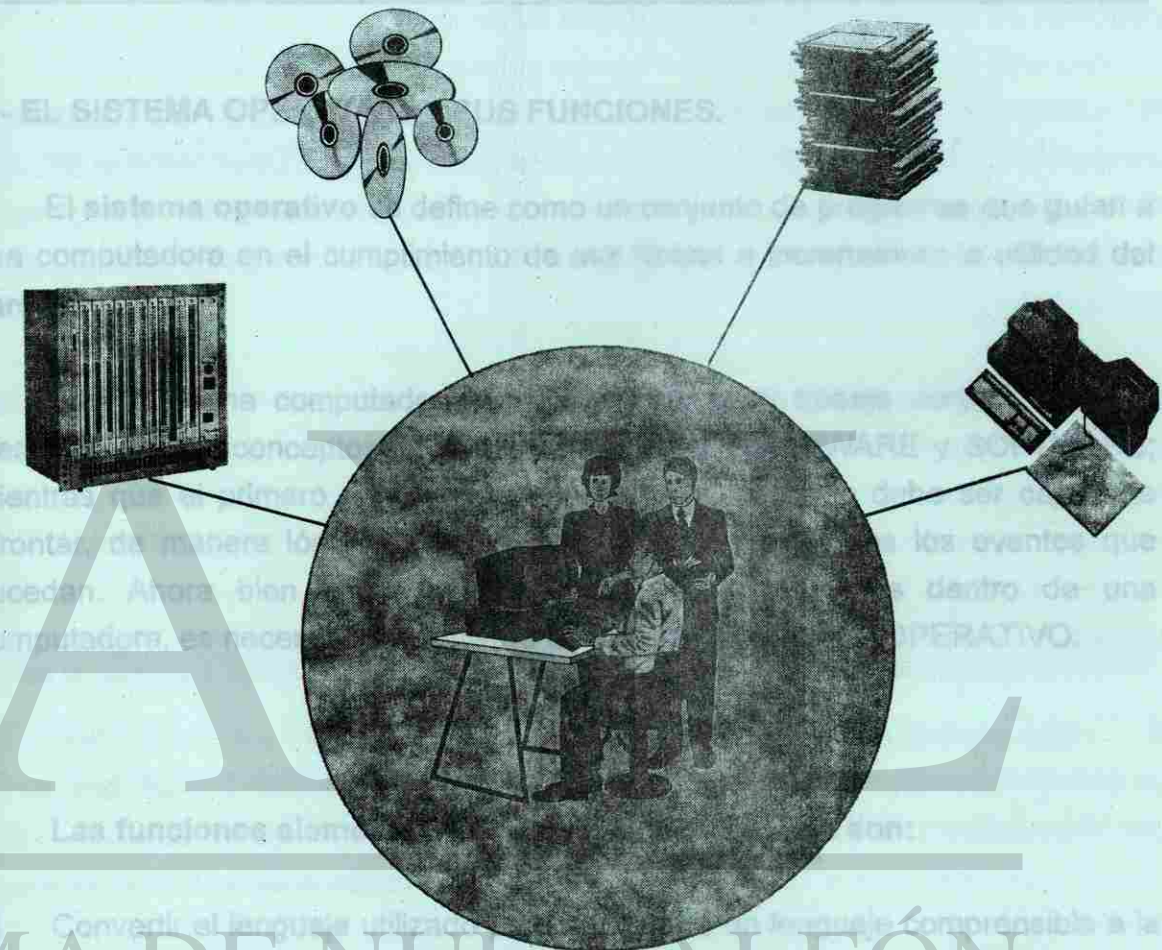
EJERCICIO CRUCIGRAMA DE LA PRIMERA UNIDAD

HORIZONTALES

- 1.- Se le conoce como el "Padre de las computadoras modernas".
- 5.- Aparato electrónico capaz de interpretar y ejecutar órdenes de entrada, salida, procesamiento y operaciones lógicas.
- 6.- Dispositivos en la que el usuario ingresa datos e instrucciones en la computadora.
- 8.- Nombre que se le da a los dispositivos de entrada, salida y almacenamiento que están fuera del procesador central.
- 10.- Dispositivos que comunican los resultados al usuario, desplegando la información en un monitor o en una impresora.
- 11.- Es la memoria de acceso aleatoria, llamada memoria volátil.
- 12.- Unidad mínima de almacenamiento de información, nos representa un carácter.
- 13.- Tipo de computadoras que manipulan la información por medio de caracteres representados por unos y ceros.
- 17.- Tipo de computadoras que manipulan la información por medio de señales eléctricas, se aplican en problemas de simulación.
- 18.- Dispositivos que utilizan papel para dar salida a la información.

VERTICALES

- 2.- Instrumento que ayudaba a contabilizar, fue creado en China o Egipto entre 3000 A.C.
- 3.- Dispositivos físicos que comprende una computadora.
- 4.- El de datos se realiza en la unidad central de proceso, más conocida por sus siglas en inglés CPU.
- 7.- Monitor utilizado por la mayoría de las computadoras.
- 9.- Son las siglas en inglés de la unidad central de proceso.
- 10.- Son programas o instrucciones codificadas electrónicamente para que la computadora realice ciertas tareas.
- 14.- Dispositivo de entrada que convierte textos, fotografías a una forma que puede leer una computadora.
- 15.- Memoria sólo de lectura, en la que almacenan cierto tipo de programas que necesita la computadora.
- 16.- Dispositivo de almacenamiento secundario, llamados "floppy".

UNIDAD II
SISTEMA OPERATIVO

SISTEMA OPERATIVO

CRUCIGRAMA DE LA PRIMERA UNIDAD

HORIZONTALES

- 1.- Se le conoce como el "Padre de las computadoras modernas".
- 5.- Aparato electrónico que interpreta y ejecuta las operaciones de entrada y salida.
- 8.- Dispositivo que permite el ingreso de datos a la computadora.
- 8.- Unidad mínima de información, formada por 8 bits.
- 12.- Unidad mínima de información, formada por 8 bits.
- 12.- Tipo de computadoras que manejan la información por medio de señales eléctricas.
- 17.- Tipo de computadoras que manejan la información por medio de señales eléctricas.
- 18.- Dispositivo que permite el ingreso de datos a la computadora.

VERTICALES

Instrumento que ayudaba a contabilizar; fue creado en China o Egipto entre 3000 A.C.

Dispositivo físico que comprime y almacena los datos de forma digital.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

Unidad de procesamiento de datos que realiza las operaciones de entrada y salida.

UNIDAD II

SISTEMA OPERATIVO

1.- INTRODUCCIÓN.

A.- EL SISTEMA OPERATIVO Y SUS FUNCIONES.

El sistema operativo se define como un conjunto de programas que guían a una computadora en el cumplimiento de sus tareas e incrementan la utilidad del hardware.

Para que una computadora funcione necesita el trabajo conjunto de dos elementos cuyos conceptos ya han sido definidos: HARDWARE y SOFTWARE; mientras que el primero sirve de soporte físico, el segundo debe ser capaz de afrontar, de manera lógica y en un tiempo razonable, todos los eventos que sucedan. Ahora bien, para controlar estas dos situaciones dentro de una computadora, es necesario un software denominado SISTEMA OPERATIVO.

Las funciones elementales de un sistema operativo son:

- ▣ Convertir el lenguaje utilizado por el usuario a un lenguaje comprensible a la computadora (lenguaje máquina 1 ó 0).
- ▣ Interpretar las ordenes (comandos) del operador, los cuales describen el trabajo que debe realizarse.
- ▣ Supervisar la utilización de los dispositivos de entrada y/o salida.
- ▣ Proporcionar servicios de diagnóstico de error de las partes básicas de la computadora.

B.- DIFERENTES SISTEMAS OPERATIVOS.

Existen diferentes sistemas operativos, según sus propósitos: El primer sistema operativo de disco que se empleó para las computadoras personales (PC), fue el **CP/M** (control de programas de microcomputadora), pero debido a lo complejo de su manejo, no alcanzó mayor difusión.

Otro sistema operativo es el **D.O.S.** (disk operating system); se fundamenta en el sistema operativo CP/M. Fue desarrollado por MicroSoft en colaboración con IBM, por lo que es conocido con el nombre de **MS-DOS**. Este sistema fue creado con la idea de que sólo una persona trabajara con la microcomputadora como una unidad independiente.

Debido al avance en la tecnología computacional, que ha dejado atrás los conceptos monousuario, monotarea y monoprogramación, aparece un nuevo sistema operativo llamado **UNIX**, que fue desarrollado por AT&T y trabaja como un sistema multiusuario, multitarea y multiprogramación. MicroSoft hizo una versión del **UNIX** llamándole **XENIX**. El MS-DOS, a partir de la versión 2.0, realiza un esfuerzo por acercarse al **UNIX**.

IBM sacó al mercado, junto con su PC de la serie PS/2 (personal system/2), un sistema operativo multitarea, pero monousuario, llamado **OS/2**, también creado por Microsoft.

De los sistemas operativos anteriormente mencionados, el más común y amigable es el MS-DOS, por lo que éste será el objeto de estudio en la presente unidad. Cabe mencionar que en nuestro estudio en el sistema operativo lo podemos llamar **SO**, **DOS** o **MS-DOS**.

C.- VERSIONES DEL SISTEMA OPERATIVO MS-DOS.

Debido al avance acelerado del hardware en los sistemas computacionales, se hace una continua revisión y actualización del software, por lo que el sistema operativo también tiene innovaciones; éstas se miden por medio de nuevas versiones.

Una versión del DOS se representa con un número que consta de dos partes separadas por un punto: la primera, es un número que nos indica si el DOS sufrió cambios trascendentales y la segunda, si dichos cambios fueron de poca relevancia. El MS-DOS, desde que apareció en 1981, ha tenido las siguientes versiones: **ver 1.00**, **ver 1.10**, **ver 2.00**, **ver 2.10**, hasta llegar a **ver 6.22** (Junio de 1996).

2.- INICIALIZACIÓN DEL SISTEMA.

Inicializar el sistema significa ejecutar un programa que reside en la memoria ROM llamado "programa de arranque", cuya función, entre otras cosas, es cargar el DOS a la memoria principal RAM.

El "programa de arranque" se divide en tres partes:

- ☐ **Test de fiabilidad (POST):** hace un chequeo general de la computadora para comprobar su buen estado.
- ☐ **Proceso de inicialización:** se determina el equipo que está conectado y si existen ampliaciones a la ROM-BIOS; es decir, si existen nuevos periféricos incorporados, también se reconocen.
- ☐ **Cargador (boot-strap):** esta parte es la encargada de arrancar el DOS o cualquier otro sistema operativo, carga el DOS en la memoria RAM.

B.- La carga del DOS en la memoria RAM supone la carga de tres programas principales y dos opcionales, a saber:

Programas principales:

- ▣ **IO.SYS o IBMBIO.COM:** maneja cada caracter mostrado en la pantalla, teclado, recibido o enviado a través de algún adaptador. Este es un archivo oculto.
- ▣ **MSDOS.SYS o IBMDOS.COM:** administra los archivos que se van a almacenar en un disco. Este también es un archivo oculto.
- ▣ **COMMAND.COM:** es el único archivo visible de la carga del DOS y sirve como interpretador de comandos.

Estos programas se cargan al arrancar la computadora, guardando el orden en que están enunciados.

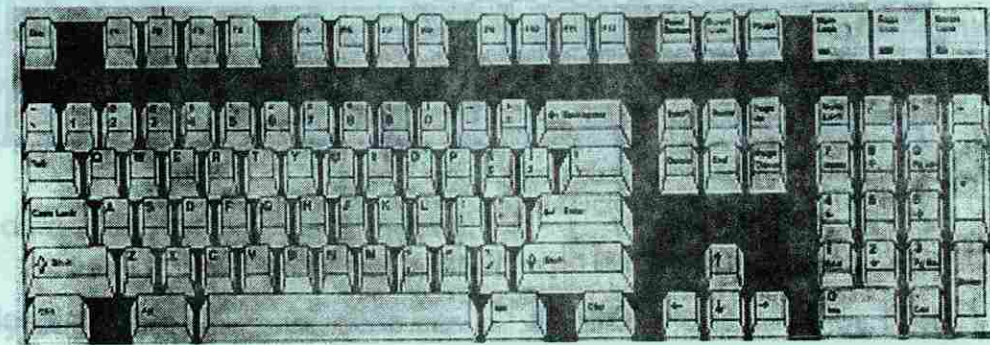
Programas opcionales:

- ▣ **CONFIG.SYS:** es un programa elaborado por el usuario para determinar la configuración de la computadora.
- ▣ **AUTOEXEC.BAT:** es diseñado por el operador para acelerar su trabajo, ya que mediante este programa la computadora ejecuta comandos del DOS uno atrás de otro, sin tener que teclearlos.

Una de las virtudes del MS-DOS es la posibilidad de ejecutar comandos; esta tarea la realiza el COMMAND.COM, que es el único archivo que no está oculto; algunas de las funciones que realiza son:

- ▣ Ubicarse en la memoria RAM y gestionar todos los errores y mensajes.
- ▣ Otra consiste en procesar el archivo AUTOEXEC.BAT, para determinar en qué lugar pueden cargarse los programas.
- ▣ Finalmente se encarga de interpretar los comandos del DOS.

3.- FUNCIONES DE LAS TECLAS EN EL MS-DOS.



Teclado Normal

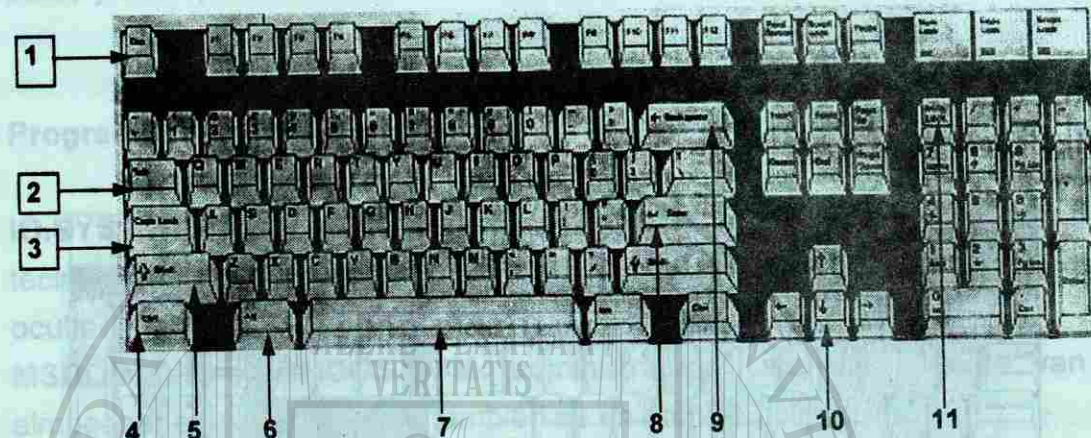
Teclas de control de cursor

Teclado Numérico

A.- TECLAS DE CONTROL

Se usan para introducir comandos o líneas de entrada. Cuando son dos las teclas de control, la primera debe permanecer oprimida mientras se pulsa la segunda; esto se representa con el signo +.

- ▣ **ENTER o INTRO o RETURN** sirve para dar fin a la línea de entrada o para mandar ejecutar un comando.
- ▣ **ESC** cancela la línea que se está escribiendo, o bien se usa para salir de alguna dificultad.
- ▣ **CTRL+BREAK o CTRL+C** aborta la ejecución de un programa.
- ▣ **NUM LOCK** activa o desactiva el teclado numérico y las teclas de direccionamiento del cursor y las teclas especiales de función.



- | | | |
|---------------|---------------------------------|---------------|
| 1.- Esc | 6.- Alt | 11.- Num Lock |
| 2.- Tab | 7.- Barra Espaciadora | |
| 3.- Caps Lock | 8.- Enter | |
| 4.- Ctrl | 9.- Back Space | |
| 5.- Shift | 10.- Teclas de direccionamiento | |

B.- TECLAS DE EDICIÓN

Permiten al operador hacer cambios al último comando del DOS que se haya usado.

- ☞ **DEL** borra el carácter en el que esté ubicado el cursor.
- ☞ **INS** inserta el carácter en el espacio señalado por el cursor.
- ☞ **F1** cada vez que se pulsa, escribe carácter por carácter del último comando utilizado.
- ☞ **F3** escribe todos los caracteres del último comando utilizado.

Estas son, entre otras, algunas teclas de edición; cabe señalar que van ligadas con la versión del DOS.

4.- IMPORTANCIA DE LOS ARCHIVOS EN EL MS-DOS.

A.- CONCEPTO DE ARCHIVO.

Antes de entrar al tema de cómo funciona el sistema operativo, es necesario tener una visión sobre algunos conceptos básicos para un mejor entendimiento de los tópicos a tratar.

Los conceptos elementales son:

- ☞ **Byte:** es un conjunto de ocho bits.
- ☞ **Campo:** es un conjunto de byte(s) o carácter(es).
- ☞ **Registro:** es un conjunto de campos.
- ☞ **Archivo:** es un conjunto de registros.

Tipos de archivos considerando su contenido:

- ☞ Programas de aplicación.
- ☞ Documentos.
- ☞ Gráficas.
- ☞ Hojas de cálculo.
- ☞ Programas en BASIC, COBOL, PASCAL, etc. (en cualquier superlenguaje).
- ☞ Comandos del sistema operativo.

Cabe aclarar que al sistema operativo no le importa el contenido de los archivos, por lo que todos son tratados casi idénticamente.

B.- NOMBRE DE LOS ARCHIVOS.

Ahora mencionaremos las características para asignar el nombre a los archivos y en base al equipo utilizado en las Aulas activas (PC compatibles), tendremos que considerar los siguientes factores:

- ▣ Dar un solo nombre a un archivo.
- ▣ El nombre se compone de dos partes separadas por un punto.
- ▣ La primera (nombre del archivo) puede tener hasta ocho caracteres.
- ▣ La segunda (extensión) puede tener hasta tres caracteres.

FILENAME.EXT

ALUMNOS.TXT

- ▣ El nombre del archivo debe ser lo más descriptivo posible y con referencia al contenido del mismo.
- ▣ No debe haber espacios en blanco, para evitar problemas en su manejo.
- ▣ En el nombre o extensión no utilizar caracteres especiales como:
? , . * : ; " \ | +
- ▣ No emplear palabras reservadas como:
CLOCKS, CON, AUX, COM, LPT, NUL, PRN.
- ▣ Es conveniente que la extensión indique el tipo de información del archivo.

Estos son algunos ejemplos de extensiones:

.BAS	extensión de un archivo de programa BASIC.
.BAT	extensión de un archivo por lotes (BATCH).
.COM	extensión de un archivo de comando ejecutable.
.DBF	extensión de un archivo de base de datos.
.DOC	extensión de un archivo de documento.
.EXE	extensión de un archivo de programa ejecutable.
.PAS	extensión de un archivo de programa en PASCAL.
.TXT	extensión de un archivo de texto sin formatear.
.WKS	extensión de un archivo en Works, en la hoja de cálculo.

Es importante señalar que el nombre adecuado de los archivos beneficia la organización de los datos.

5.- USO DEL MS-DOS.**A.- PROCESO DE CARGA**

Existen dos maneras para inicializar una computadora, el **inicio en frío** que consiste en encender la computadora y cargar el sistema y el **inicio en caliente** que consiste en reinicializar el sistema.

Procura seguir paso a paso las siguientes indicaciones, hay que tomar en cuenta la forma en que se encuentran conectadas las computadoras en el laboratorio de computación (Aula Activa).

COMPUTADORA CONECTADA EN RED

- ▣ **Inicio en frío:**
 - 1° El maestro activará la fuente de alimentación de la aula activa sin encender el servidor.
 - 2° Enciende el CPU y el monitor de tu estación de trabajo.
 - 3° Observa en el monitor y en el CPU de tu estación de trabajo el proceso de carga.
 - 4° Como tu estación de trabajo se encuentra conectada en red, marcará un aviso de que no hay enlace con el servidor y aparecerá en tu monitor el prompt **A:\>**
 - 5° El letrero **A:\>** indica que te encuentras en el drive A, y el disco que contiene se le llama disco de arranque.
 - 6° En el disco de arranque se encuentran algunos archivos entre ellos el MS-DOS
- ▣ **Inicio en caliente:**
 - 1° Espera a que el maestro encienda el servidor y de alta la Red.
 - 2° Oprime las teclas CTRL + ALT + DEL simultáneamente, o bien oprimiendo el botón RESET.

- 3° Observa en el monitor y en el CPU de tu estación de trabajo el proceso de carga.
- 4° Al terminar el proceso escucharas un sonido que te indicará que tu computadora se encuentra conectada en Red.
- 5° En la pantalla de tu monitor apareceran una serie de letreros y por último el letrero (Path) **G: \ALUMNOS\ EST XX >**
- 6° Escribe **A:** y pulsa Enter, con esta instrucción te cambiaras del drive G al drive A (Observa como se enciende la luz roja del drive A)
- 7° Escribe **G:** y pulsa Enter, con esta instrucción te cambiaras del drive A al drive G.
- 8° Deja activado el prompt en el drive A.

COMPUTADORA SIN ESTAR CONECTADA EN RED

Si tu computadora no se encuentra conectada en red el inicio en frío se llevaría acabo con los primeros tres pasos de inicio en frío y el inicio en caliente con el segundo paso de inicio en caliente de una computadora conectada en Red.

Las unidades de almacenamiento secundario (drives, disco duro, CD-ROM, etc.) que contiene una computadora, se encuentran designadas por letras por ejemplo:

drive A	A:\>
drive B	B:\>
disco duro	C:\>
CD-ROM	D:\>
drive de la Red	F:\>, G:\>

Nota: Tu estación de trabajo (nombre que recibe la computadora por estar conectada en Red) pueden tener uno o dos drives, para los discos de 5 ¼ ó 3 ½ , si deseas cambiar de drive solo tienes que poner la letra, dos puntos y pulsar Enter, ejemplos: **A: , B: , C: , D: , F: , G: , etc.**

B.- COMANDOS

Un comando del sistema operativo es una instrucción que puede ejecutar de inmediato la computadora. Los comandos se clasifican en dos tipos:

Comandos internos: Son comandos ocultos del MS-DOS que, cuando se inicializa el sistema, se cargan en la memoria y no se pueden ver en el directorio, pero siempre se pueden utilizar. Como ejemplos de estos comandos podemos citar:

DIR	COPY
CLS	VER
REN	MD
PROMPT	RD
DEL	CD
VOL	

Comandos externos: Son comandos ejecutables, por lo que deben leerse antes de usarse; es decir, se encuentran en el disco del DOS y éste debe estar en el drive A, ya que de lo contrario la computadora no los procesará. Cualquier archivo de extensión BAT, EXE o COM, se considera comando externo, aunque al usarlo no se teclee su extensión. Algunos ejemplos de comandos externos son:

FORMAT
LABEL
CHKDSK
DISKCOPY
TREE

Quando se utiliza un comando del DOS, se puede escribir con letras mayúsculas o minúsculas, porque éste las interpreta de la misma manera. Es importante hacer notar que siempre que se haya terminado de escribir el comando, para ejecutarlo hay que pulsar la tecla Enter.

C.- CARACTERES COMODINES

Los caracteres comodines, como su nombre lo indica, son caracteres utilizados para acomodarlos en el lugar necesario y en el momento oportuno dentro de los comandos, para cumplir una determinada tarea. En computación los comodines más utilizados son:

Caracter	Representa
*	Grupo de caracteres.
?	Un solo caracter.

Por ejemplo si los aplicamos en los nombre de archivos:

ALU*.BAS Toma en cuenta todos los archivo que inicie con alu y con la extensión .BAS ejem: ALUMNOS.BAS, ALUMNA.BAS, ALUDIDO.BAS, etc.

***DO.COM** Toma en cuenta todos los archivos que inicien con cualquier nombre pero que terminen con DO.COM ejem: PERDIDO.COM, SERVIDO.COM, ACEPTADO.COM.

****** Toma en cuenta todos los archivos con todo tipo de extensión ejem: EXAMEN.TXT, ARCHIVO1.BAS, GRUPO.WDB, etc.

GRU?O.EXE Toma en cuenta el nombre del archivo con cualquier caracter en ? ejem: GRUPO.EXE, GRUTO.EXE, GRUEO.EXE, etc.

D.- USO DE COMANDOS

Para iniciar la práctica de los comandos del MS-DOS es necesario que tu cursor se encuentre en el drive G (drive de la Red) y que en el drive A se encuentre el disco de arranque de la estación (con su protector contra escritura), con los archivos MS-DOS y los comandos externos. Por último que tu disco de trabajo se encuentre en el drive B

Nota: Si tu Aula Activa no está conectada en Red, realiza los pasos que el maestro te indicará para ajustarse a la siguiente práctica.

Comando Interno DIR: Enlista los nombres y extensiones de los archivos contenidos en un disco, así como su tamaño, fecha y hora en que fueron creados o modificados por última vez.

DIR	Muestra los archivos de la unidad de disco actual
DIR A:	Muestra los archivos del disco del drive A.
DIR/P	Es un directorio con pausa, muestra 23 archivos por pantalla, proporcionando el nombre y extensión, así como su tamaño, la fecha y hora en que fueron creados o modificados por última vez.
DIR/W	Es un directorio que se presenta a lo ancho de la pantalla, mostrando bloques de cinco archivos en cada línea, y solamente proporciona su nombre y la extensión.
DIR *.EXE	Muestra todos los archivos de extensión .EXE
DIR *.COM	Muestra todos los archivos de extensión .COM
DIR *.*	Muestra todos los archivos con sus extensiones.

EJERCICIOS:

Ubicado el cursor en el drive G.

Teclea DIR y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea DIR A: y pulsa la tecla Enter.

¿Qué aparece en la pantalla A?

Teclea DIR/P y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea DIR/W y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea DIR *.EXE y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea DIR *.COM y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea DIR *. y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

EJERCICIOS:

Nota: Los archivos que aparecieron con extensión <DIR> son subdirectorios que más adelante explicaremos.

Comando externo FORMAT: Este comando se usa para inicializar un disco, es decir, prepara la superficie magnética para asignarle un patrón que la computadora utilizada pueda comprender; el formateo de un disco detecta los sectores defectuosos, borra cualquier información grabada previamente, prepara el espacio para la tabla de asignación de archivos (FAT) y deja preparado el disco para que el DOS. pueda leer y escribir en él.

FORMAT A: Formatea el disco del drive A.

FORMAT B: Formatea el disco del drive B.

EJERCICIOS:

Es necesario que cuentes con el MS-DOS en el disco de arranque de la unidad A y también algunos de los comandos externos (LABEL, FORMAT, TREE), y que tu disco de trabajo se encuentre en el drive B.

Ubicado el cursor en el drive G.

Teclea **FORMAT B:** y pulsa Enter.
¿Qué aparece en la pantalla?

Presiona la tecla Enter. Aparece lo siguiente:

Verificando el formato existente en disco
Guardando la información de RECONSTRUIR
Verificando **XXXX** (tamaño del disco)
XX por ciento completado (espera alrededor de un minuto)
(Formato completado)

Etiqueta de volumen (11 caracteres, presione ENTER si se ignora?) _

Escribe tu primer nombre después del signo ? y pulsa Enter. Aparece lo siguiente:

XXXXXX bytes de espacio total en disco
XXXXXX bytes disponibles en disco
XXXX bytes en cada unidad de asignación.
XXX unidades de asignación disponibles en disco

Número de serie de volumen es **XXXX - XXXX**

¿ Desea dar formato a otro disquete (S/N)? _

Si se presiona "S" la computadora se prepara para dar formato a otro disco, teclea "N" y pulsa Enter para terminar con este comando.

☑ **Comando interno COPY:** Copia uno o más archivos de un disco a otro, o bien puede copiar dos o más archivos en uno.

COPY TREE.COM B: Copia en el disco B el archivo TREE.COM desde el disco de la unidad actual.

COPY *.COM B: Copia todos los archivos de extensión .COM del disco de la unidad actual.

COPY *.* B: Copia todos los archivos del disco de la unidad actual al disco B.

EJERCICIOS:

Ubicado el cursor en el drive A.

Teclea **COPY TREE.COM B:** y pulsa la tecla Enter.
Teclea **DIR B:** y pulsa la tecla Enter.
¿Qué aparece en la pantalla?

Teclea **COPY *.COM B:** y pulsa la tecla Enter.

Teclea **DIR B:** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea **COPY *.EXE B:** y pulsa la tecla Enter.

Teclea **DIR B:** y pulsa Enter.

¿Qué aparece en la pantalla?

Teclea **COPY *.* B:** y pulsa Enter

Teclea **DIR B:** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Comando interno REN: Renombra o modifica el nombre de un archivo.

REN ARBOL.TXT DIAGRAMA.TXT Cambia el nombre del archivo ARBOL a DIAGRAMA.

REN DIAGRAMA.TXT ARBOL.TXT Cambia el nombre del archivo DIAGRAMA a ARBOL.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **DIR** y pulsa la tecla Enter.

Teclea **REN nombre de un archivo del drive B ARBOL.TXT** y pulsa la tecla Enter.

Teclea **DIR** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Cambia el nombre de varios archivo del drive B con los nombres de ARBOL1.TXT, ARBOL2.TXT, etc.

Comando interno DEL: Borra, remueve o elimina archivos.

DEL B:ARBOL.TXT Borra el archivo ARBOL.TXT del disco del drive B.

DEL ARBOL.TXT Borra el archivo ARBOL.TXT del disco de la unidad actual.

DEL *.TXT Borra todos los archivos de extensión TXT.

DEL ARBOL.* Borra los archivos ARBOL de cualquier extensión del disco de la unidad actual.

DEL ARBOL?.TXT Borra los archivos que empiezan con ARBOL y el caracter siguiente, y con extensión TXT.

DEL ARBOL*.TXT Borra los archivos que empiezan con ARBOL y todos los caracteres siguientes, y con extensión TXT.

DEL *.* Borra todos los archivos del disco de la unidad actual.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **DEL ARBOL.TXT** y pulsa la tecla Enter.

Teclea **DIR** y pulsa la tecla Enter.

¿Borró el archivo ARBOL.TXT ?

Teclea **DEL ARBOL?.TXT** y pulsa la tecla Enter.

Teclea **DIR** y pulsa la tecla Enter.

¿Borró todos los archivos que empiezan con ARBOL

¿Por qué?

Teclea **DEL ARBOL*.TXT** y pulsa la tecla Enter.

Teclea **DIR** y pulsa la tecla Enter.

¿Borró todos los archivos que empiezan con **ARBOL** ?

¿Por qué?

Comando externo LABEL: Crea, modifica o borra una etiqueta de volumen en un disco.

LABEL Pide el nombre de la etiqueta del disco de la unidad actual

LABEL B: Pide el nombre de la etiqueta el disco en el drive B.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **LABEL** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea **LABEL A: JPEREZ** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea **LABEL** y pulsa la tecla Enter.

¿Cambió el nombre de la etiqueta?

Cuando aparezca:

ETIQUETA DE VOLUMEN (11 CARACTERES ENTRAR PARA IGNORAR)

Teclea tu primer nombre (no más de 11 caracteres) y pulsa Enter.

Comando interno VOL: Sólo muestra o despliega la etiqueta de volumen de un disco.

VOL Muestra la etiqueta de volumen del disco de la unidad actual

VOL A: Muestra la etiqueta de volumen del disco del drive A.

VOL B: Muestra la etiqueta de volumen del disco del drive B.

EJERCICIOS:

Ubicado el cursor en el drive G.

Teclea **VOL A:** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

Teclea **VOL B:** y pulsa la tecla Enter.

¿Qué aparece en la pantalla?

¿Aparece la etiqueta de volumen que teclaste cuando se formateó tu disco?

Comando interno VER: Muestra o despliega la versión del sistema operativo.

VER

Despliega la versión del sistema operativo que se está usando.

EJERCICIOS:

Ubicado el cursor en el drive G.

Teclea **VER** y pulsa Enter.

¿Qué aparece en la pantalla?

Pulsa la tecla Enter.

Teclea **B:** y pulsa Enter.

Teclea **VER** y pulsa Enter.

¿Qué aparece en la pantalla?

Teclea **N** y pulsa Enter.

Comando interno CLS: Limpia la pantalla, colocando el cursor en la parte superior izquierda de la misma.

CLS

Limpia la pantalla.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **CLS** y pulsa Enter.

¿Qué sucedió en la pantalla?

Comando externo KEYB y KEYBOARD: Cambia el teclado a cualquier idioma. El keyb no funciona sin el archivo keyboard en el disco de la unidad actual.

KEYB SP

Cambia el teclado al español (SP) Spanish.

EJERCICIOS:

Ubicado el cursor en el drive B.

Sólo en caso necesario se utiliza este comando para cambiar el teclado de tu estación de trabajo de Inglés a Español.

Teclea **KEYB SP** y pulsa Enter

¿Qué sucedió?

Comando externo CHKDSK: Crea un reporte del estado del disco, por ejemplo su tamaño en bytes, espacio ocupado, espacio libre, sectores dañados, número de archivos ocultos, directorios y el estado de la memoria RAM.

CHKDSK Checa el disco del drive de la unidad actual.

CHKDSK B: Checa el disco del drive B, desde el disco del drive de la unidad actual

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **CHKDSK** y pulsa Enter.

¿Qué sucedió?

Teclea **CHKDSK A:** y pulsa Enter.

¿Qué aparece en la pantalla?

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Teclea **VER** y pulsa Enter.

Comando externo DISKCOPY: Copia toda la información contenida de un disco a otro; este comando sólo puede ser utilizado con discos flexibles o semiflexibles, y siempre de la misma densidad. Si el disco destino (nuevo) no está formateado, el comando diskcopy lo formateará automáticamente y si tiene alguna información, también la borrará.

DISKCOPY A: B: Copia toda la información del disco A al disco B.

DISKCOPY A: A: Copia el disco fuente del drive A al disco destino en el mismo drive. El sistema operativo indica cuándo deben de cambiarse los discos.

EJERCICIOS:

Una vez que teclaste todos los comandos, tu compañero hará un diskcopy, para que ambos tengan la misma información referente al sistema operativo.

Ubicado el cursor en el drive A.

Coloca el disco fuente (el que tiene la información) en el drive A.

Coloca el disco destino (nuevo) en el drive B.

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

¿Qué sucedió?

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Teclea **DISKCOPY A: B:** y pulsa la tecla Enter

Comando interno **PROMPT**: Es un indicador de petición de entrada, automáticamente despliega el nombre de la unidad actual, por ejemplo A:\> , B:\>, C:\> pero puede ser modificado de acuerdo al gusto del operador.

PROMPT \$T	Aparece la hora actual.
PROMPT \$D	Aparece la fecha actual.
PROMPT \$P	Aparece el drive asignado y el directorio actual.
PROMPT \$G	Aparece el signo >
PROMPT \$L	Aparece el signo <

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **PROMPT P\$G\$** y pulsa Enter.

¿Cambió el PROMPT?

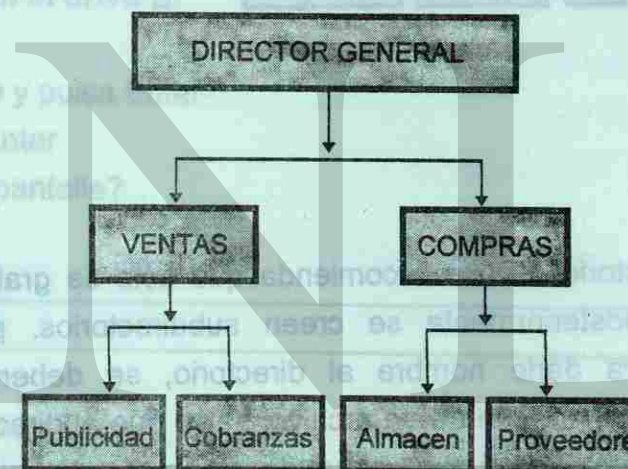
Nota: Se debe dejar el PROMPT \$P\$G para el siguiente tema.

DIRECCIÓN GENERAL DE BIBLIOTECAS

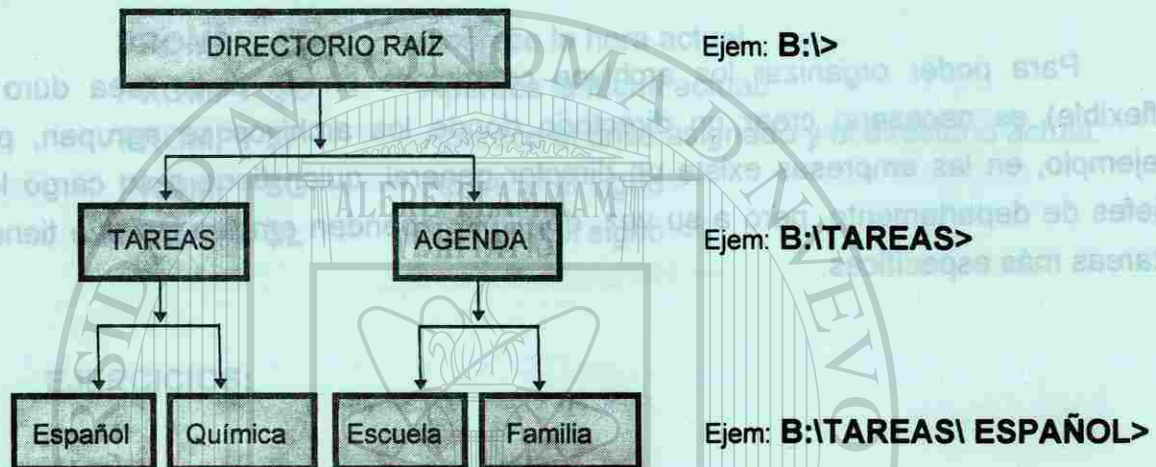
6.- LOS DIRECTORIOS EN EL MS-DOS.

A.- ESTRUCTURA DE UN DIRECTORIO

Para poder organizar los archivos contenidos en un disco (sea duro o flexible) es necesario crear un directorio donde los archivos se agrupen, por ejemplo, en las empresas existe un director general, quien tiene a su cargo los jefes de departamento, pero a su vez, de ellos dependen empleados que tienen tareas más específicas.



Trasladando este ejemplo a la organización de nuestros archivos, observamos que en el primer nivel tendríamos el **directorio raíz** o **directorio principal**; en el segundo nivel aparecerían los **subdirectorios**, que se convertirían en directorios, si existieran más niveles. Cabe aclarar que fuera del directorio raíz todos son subdirectorios; sin embargo, comúnmente se utiliza el término directorio.



En el directorio raíz se recomienda que sólo se graben los archivos más esenciales, y posteriormente se creen subdirectorios, para áreas o temas específicos. Para darle nombre al directorio, se deben seguir las mismas condiciones que el nombre de un archivo, salvo que al directorio no se le pone la extensión. el directorio actual es aquel donde se está trabajando, cuando el DOS inicia, el directorio actual es el directorio raíz.

B.- COMANDOS PARA MANIPULAR DIRECTORIOS

¿A qué directorio cambié?

Para iniciar la práctica de los comandos del MS-DOS es necesario que tu cursor se encuentre en el drive G, que en el drive A se encuentre el disco de arranque y por último que tu disco de trabajo se encuentre en el drive B.

Teclea CD.. y pulsa Enter.

Comando interno MKDIR o MD: Crea un directorio.

MD ALUMNO Crea un directorio raíz llamado ALUMNO.

EJERCICIOS:

Teclea CD.. y pulsa Enter.

Ubicado el cursor en el drive B.

Teclea MD ALUMNO y pulsa Enter

Teclea DIR y pulsa Enter

¿Qué aparece en la pantalla?



Comando interno CHDIR o CD: Cambia de un directorio a otro.

CD nombre del directorio Cambia del directorio raíz al directorio especificado.

CD Cambia al directorio raíz directamente, no importa en el nivel en que te encuentres. Al caracter \ se le conoce con el nombre de diagonal invertida o "backslash".

CD.. Cambia al directorio siguiente. En caso de ser tres los niveles y estar en el último, se utiliza este comando para pasar al segundo nivel y finalmente al primero.


1020124221

EJERCICIOS:

Ubicado en el drive B

Teclea **CD ALUMNO** y pulsa Enter.

¿Observaste algún cambio en el PROMPT? 

¿Cuál? 

Teclea **DIR** y pulsa Enter.

¿Qué aparece en la pantalla?



Teclea **MD CALIFIC** y pulsa Enter.

Teclea **DIR** y pulsa Enter.

¿Qué aparece en la pantalla?



Teclea **CD CALIFIC** y pulsa Enter.

¿Cuál cambio observaste en el PROMPT?



Teclea **MD MATERIA** y pulsa Enter.

Teclea **CD MATERIA** y pulsa Enter.

¿Qué cambio observaste en el PROMPT?



Teclea **CD..** y pulsa Enter.

¿A qué directorio cambió?



Teclea **CD..** y pulsa Enter.

¿A qué directorio cambió?



Teclea **CD..** y pulsa Enter.

¿A qué directorio cambió?



Para ir a un directorio de cualquier nivel en forma directa, debemos teclear el camino que va desde el directorio raíz, al directorio deseado.

Teclea **CDALUMNO\CALIFIC\MATERIA** y pulsa Enter.

Teclea **DIR** y pulsa Enter.

¿En cuál directorio te encuentras?



Teclea **CD** y pulsa Enter.

¿A qué directorio cambió directamente?



Comando externo TREE: Despliega en pantalla la estructura de subdirectorios en un disco.

TREE Muestra la estructura de subdirectorios del disco en el drive de la unidad actual.

TREE B: Muestra desde la unidad actual, la estructura de subdirectorios del disco en el drive B.

TREE A:/F Muestra la estructura de subdirectorios del disco A, y con /F, muestra la lista de archivos contenidos en los subdirectorios.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **TREE** y pulsa Enter.

¿Qué aparece en la pantalla?

Teclea **TREE /F** y pulsa Enter.

¿Observaste algún cambio?

_____ ¿ Por qué ?

Comando interno RMDIR o RD: elimina o remueve directorios, siempre y cuando el directorio a borrar no contenga directorios o archivos.

RD MATERIA Borra o elimina el subdirectorio MATERIA.

RD CALIFIC Borra o elimina el subdirectorio CALIFIC.

RD ALUMNO Borra o elimina el subdirectorio ALUMNO.

EJERCICIOS:

Ubicado el cursor en el drive B.

Teclea **RDALUMNO\CALIFIC\MATERIA** y pulsa Enter.

Teclea **TREE** y pulsa Enter.

Menciona la diferencia que observaste:

Teclea **RDALUMNO\CALIFIC** y pulsa Enter.

Teclea **TREE** y pulsa Enter.

Menciona la diferencia que observaste:

Teclea **RDALUMNO** y pulsa Enter.

Teclea **TREE** y pulsa Enter.

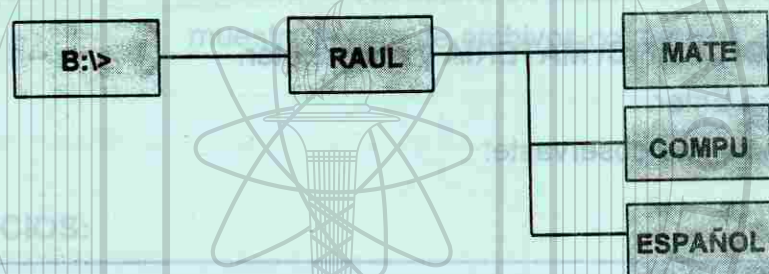
Anota tus observaciones:

Teclea **DIR** y pulsa Enter.

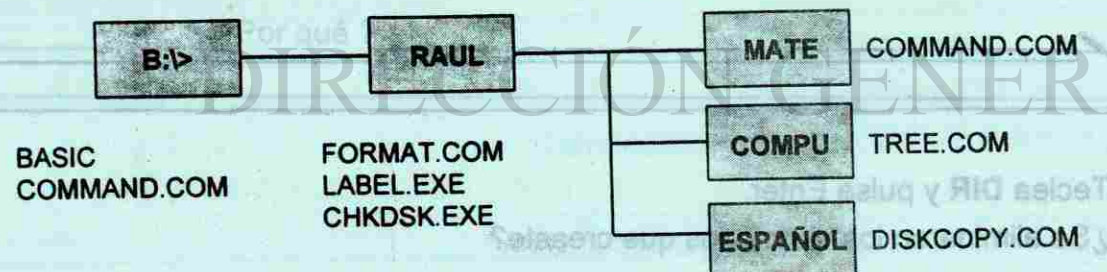
¿Se eliminaron los directorios que creaste?

☺ EJERCICIO GENERAL :

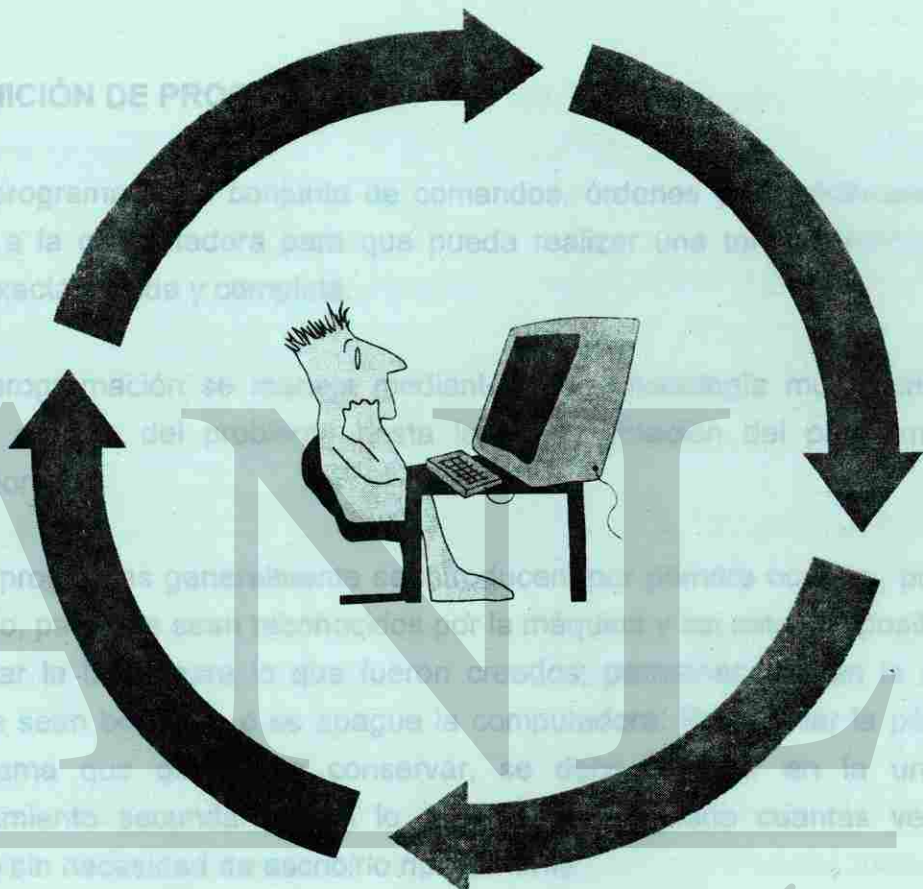
- 1.- Realiza los pasos necesarios para dejar tu disco de trabajo sin ningún archivo o subdirectorio.
- 2.- Crea en el directorio raíz un subdirectorio con tu primer nombre, a partir de ese subdirectorio crea otros tres con el nombre de las materias que estas cursando actualmente.



- 3.- En el directorio raíz copia **BASIC** y el **COMMAND.COM**
- 4.- Copia los comandos externos **FORMAT**, **LABEL** y **CHKDSK** en el subdirectorio de tu **nombre**.
- 5.- Copia el comando externo **COMMAND.COM** al subdirectorio **MATE**.
- 6.- Copia el comando externo **TREE.COM** al subdirectorio **COMPU**.
- 7.- Copia el comando externo **DISKCOPY.COM** al subdirectorio **ESPAÑOL**.
- 8.- Ubicado el cursor en el subdirectorio raíz ejecuta el **TREE** y el **TREE /F**



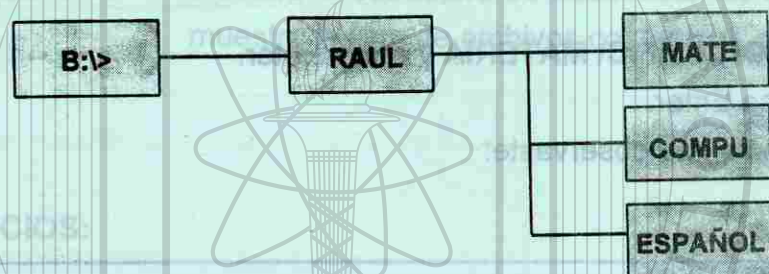
INTRODUCCIÓN A LA PROGRAMACIÓN



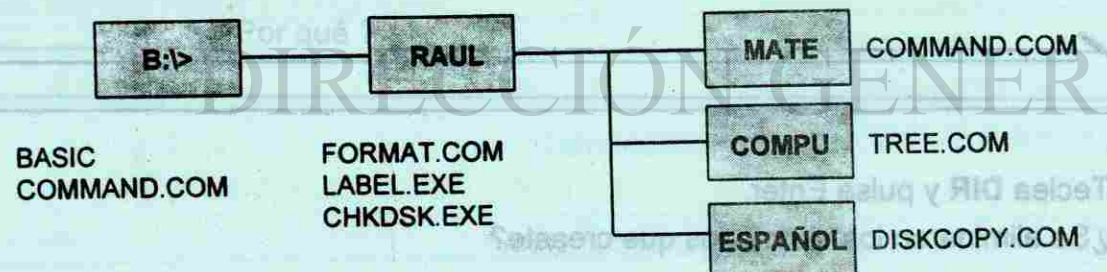
INTRODUCCIÓN A LA PROGRAMACIÓN

☺ EJERCICIO GENERAL :

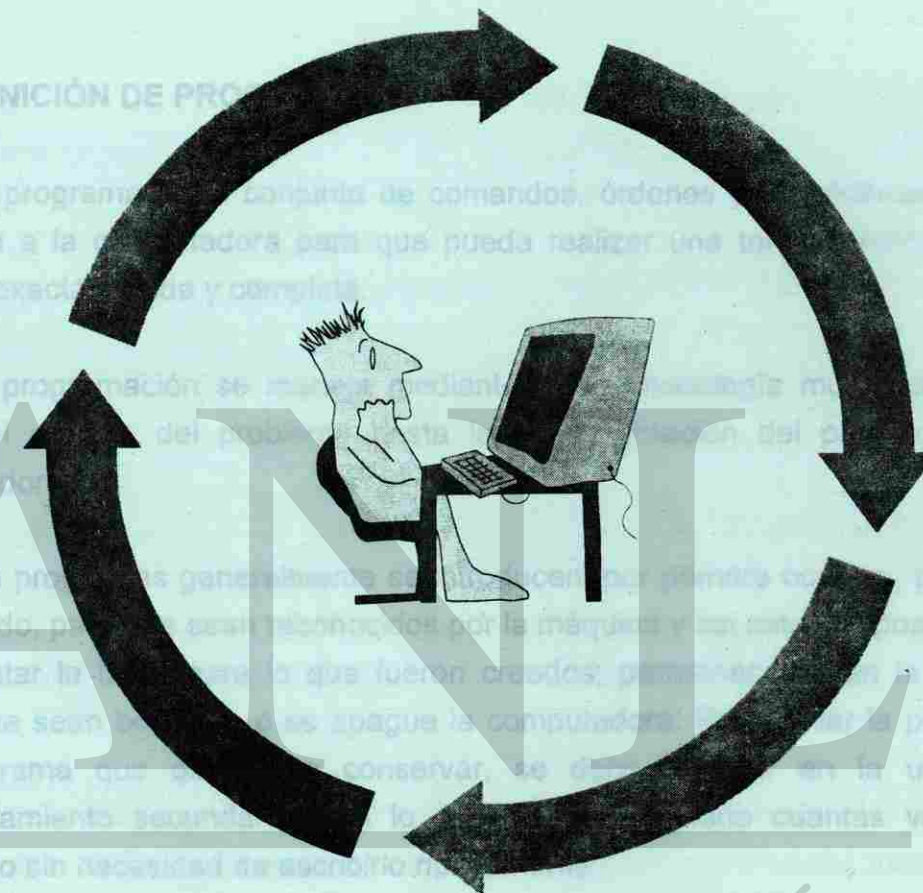
- 1.- Realiza los pasos necesarios para dejar tu disco de trabajo sin ningún archivo o subdirectorio.
- 2.- Crea en el directorio raíz un subdirectorio con tu primer nombre, a partir de ese subdirectorio crea otros tres con el nombre de las materias que estas cursando actualmente.



- 3.- En el directorio raíz copia **BASIC** y el **COMMAND.COM**
- 4.- Copia los comandos externos **FORMAT**, **LABEL** y **CHKDSK** en el subdirectorio de tu nombre.
- 5.- Copia el comando externo **COMMAND.COM** al subdirectorio **MATE**.
- 6.- Copia el comando externo **TREE.COM** al subdirectorio **COMPU**.
- 7.- Copia el comando externo **DISKCOPY.COM** al subdirectorio **ESPAÑOL**.
- 8.- Ubicado el cursor en el subdirectorio raíz ejecuta el **TREE** y el **TREE /F**



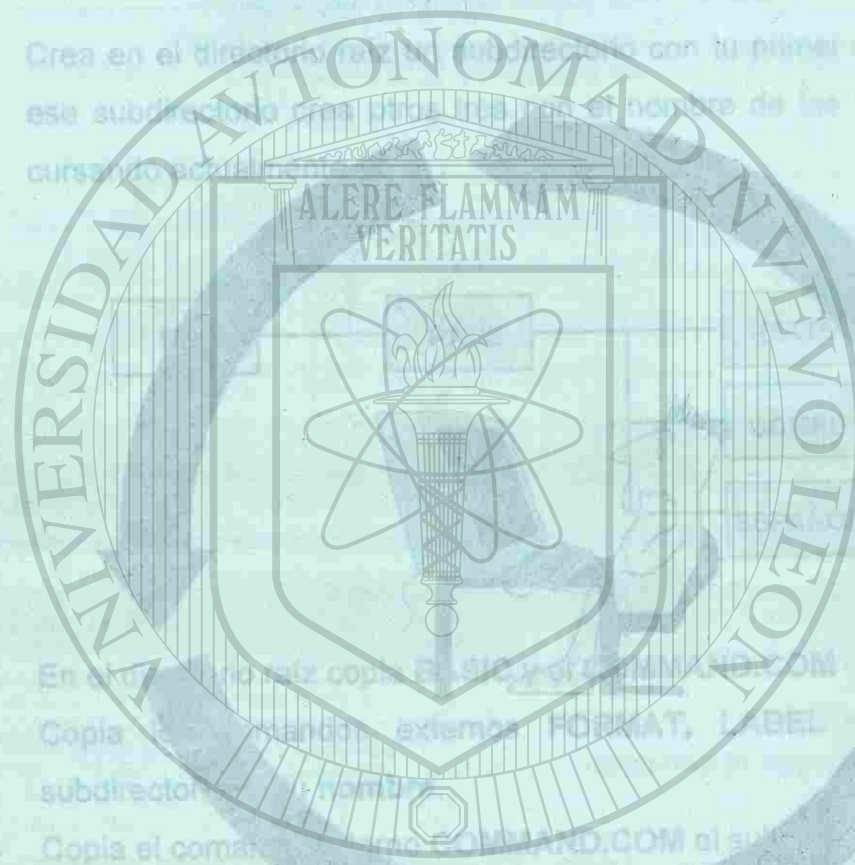
INTRODUCCIÓN A LA PROGRAMACIÓN



INTRODUCCIÓN A LA PROGRAMACIÓN

EJERCICIO GENERAL

- 1.- Realiza los pasos necesarios para dejar tu disco de trabajo sin ningún archivo o subdirectorio.
- 2.- Crea en el directorio raíz un subdirectorio con tu primer nombre, a partir de ese subdirectorio crea un archivo llamado "PROGRAMA" que contenga los comandos siguientes:
 - 3.- En el directorio raíz copia el comando FORMAT.
 - 4.- Copia el comando externo FORMAT, LABEL.
 - 5.- Copia el comando interno COPY.
 - 6.- Copia el comando externo COPY.
 - 7.- Copia el comando externo DISKCOPY.
 - 8.- Ubica el comando en el directorio raíz.



LA DIRECCIÓN GENERAL DE BIBLIOTECAS

BASIC
COMMAND

PROGRAMACIÓN

TRIE.COM

DISKCOPY.COM

Hoja electrónica

UNIDAD III PROGRAMAS DEL SISTEMA

INTRODUCCIÓN A LA PROGRAMACIÓN

1.- INTRODUCCIÓN.

A.- DEFINICIÓN DE PROGRAMA

Un programa es el conjunto de comandos, órdenes y procedimientos, que instruyen a la computadora para que pueda realizar una tarea determinada de manera exacta, rápida y completa.

La programación se maneja mediante una metodología muy técnica, que parte del análisis del problema hasta la implementación del programa en la computadora.

Los programas generalmente se introducen, por primera ocasión, por medio del teclado, para que sean reconocidos por la máquina y así estar en posibilidades de ejecutar la tarea para lo que fueron creados; permanecerán en la memoria hasta que sean borrados o se apague la computadora. Para evitar la pérdida de un programa que deseamos conservar, se debe guardar en la unidad de almacenamiento secundario, con lo cual podemos usarlo cuantas veces sea necesario sin necesidad de escribirlo nuevamente.

Los programas que manejan las computadoras pueden dividirse en tres tipos:

- ☞ Programas del sistema operativo.
- ☞ Programas de lenguaje de programación.
- ☞ Programas de aplicación.



B.- PROGRAMAS DEL SISTEMA OPERATIVO

Como ya se dijo anteriormente, un sistema operativo es una colección de programas y utilerías diseñados para facilitar al usuario la creación y el manejo de archivos; además se encargan de ejecutar programas, de la operación y control de los dispositivos periféricos conectados con la computadora.

C.- PROGRAMAS DE LENGUAJE DE PROGRAMACIÓN

En el lenguaje de programación el usuario puede elaborar, mediante un conjunto de instrucciones que se le van dictando a la computadora, su propio programa para resolver un problema en particular. Los lenguajes de programación más comunes son: BASIC, QUICK BASIC, FORTRAN, PASCAL, "C" y otros más.

D.- PROGRAMAS DE APLICACIÓN

Los programas de aplicación están formados por un conjunto de instrucciones generales codificadas para resolver cierto tipo de problemas con las mismas características.

Las ventajas para los usuarios en este tipo de programas o paquetes, es que no se necesita saber programación, ya que es relativamente fácil su uso, puesto que el equipo que generó el paquete lo hizo pensando en dar solución a todo tipo de problemas que se encuentre relacionado con su aplicación.

Estos programas se clasifican en:

- ☐ HOJA ELECTRÓNICA
- ☐ PROCESADORES DE PALABRAS
- ☐ BASES DE DATOS
- ☐ COMUNICACIÓN
- ☐ GRÁFICOS

Hoja electrónica

Una hoja electrónica es básicamente una hoja donde se manejan renglones marcados por números y columnas señaladas con letras, a la intersección de un renglón y una columna se le llama CELDA.

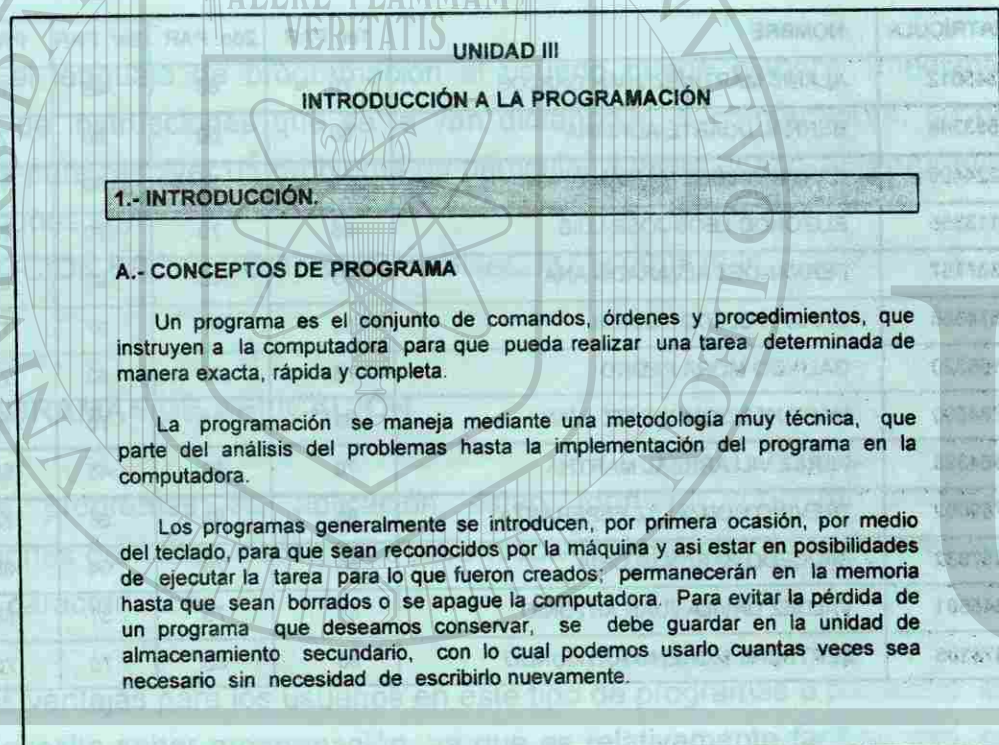
	A	B	C	D	E	F
1	MATRÍCULA	NOMBRE	1er. PAR	2do. PAR	3er. PAR	PROM
2	345612	ALANIS MARTINEZ JUAN CARLOS	70	88	65	76.5
3	553344	BERNAL DUARTE ADRIANA	80	50	90	73.3
4	224499	CANTÚ VALDEZ MARIANA	70	75	80	75
5	113355	ELIZONDO LEOS JOSE LUIS	85	70	70	75
6	337787	FERNANDEZ ALVARADO ANA	95	90	93	93
7	874356	GARZA TOLEDO YUDILIA	56	67	97	73.3
8	155320	GALINDO MORA PEDRO	78	82	53	71
9	784590	MARTINEZ DOMINGUEZ JUAN	69	71	76	72
10	564398	PEREZ VILLARREAL MARTHA	70	53	45	58
11	789062	TREVIÑO GONZALEZ ESPERANZA	85	93	56	72
12	457832	SAUCEDO BARON GLORIA	56	78	64	66
13	345691	VALDEZ DAVILA JUAN ANTONIO	62	75	31	56
14	674395	ZERTUCHE MORENO RAYMUNDO	80	60	70	70

Ejemplo de una hoja de Cálculo

El primer software de hoja electrónica fué el Visi Cal (VC), desarrollándose posteriormente otras como: Multiplan, Proyect, Lotus 1-2-3, Works, Excel, etc.

Procesadores de palabras

Los paquetes de **procesadores de palabras o textos** se crearon con el fin de facilitar y simplificar la redacción, corrección, modificación e impresión de un documento, mejorando la eficacia y productividad personal.



Ejemplo de una hoja en el procesador de texto **WORD**

Algunos de los procesadores de palabras son: WordStar, Professional Write, Chi Write, Word Perfect, Word, etc.

Base de datos

Los programas de aplicación de **base de datos** nos permiten manejar y manipular gran cantidad de información de una manera rápida. Por ejemplo, si deseamos localizar un alumno de esta preparatoria y no supieramos en que semestre o grupo toma clase, para resolver esta cuestión usamos la computadora y le proporcionamos el nombre del alumno para que lo busque en la base de datos que tiene y en seguida nos dirá no solo el grupo y/o semestre en que está, sino, además, todos los datos que se tienen acerca de él.

MATRÍCULA: 358793	
NOMBRE: RAMIREZ RENDÓN JUAN JOSE	
NOM. DEL PADRE: RAMIREZ GUTIERREZ JUAN ANTONIO	
NOM. DE LA MADRE: MARGARITA RENDÓN CHAVIRA	
DOMICILIO: CALLEJÓN DE SALSIPUEDES 2135	COL: METRORREY
CIUDAD: HIDALGO	MUNICIPIO: RIO BRAVO.
TELÉFONO: 905-3587	
SEMESTRE: TERCERO	GRUPO: 3A
MTO. DE PLANTA: RODOMIRO CAMPOSECO	
MATERIAS PENDIENTES:	MATEMATICAS 2da. OPORTUNIDAD COMPUTACIÓN 3a. OPORTUNIDAD
MATERIAS ACREDITADAS:	ORATORIA 1er. LUGAR MARATÓN 3er. LUGAR

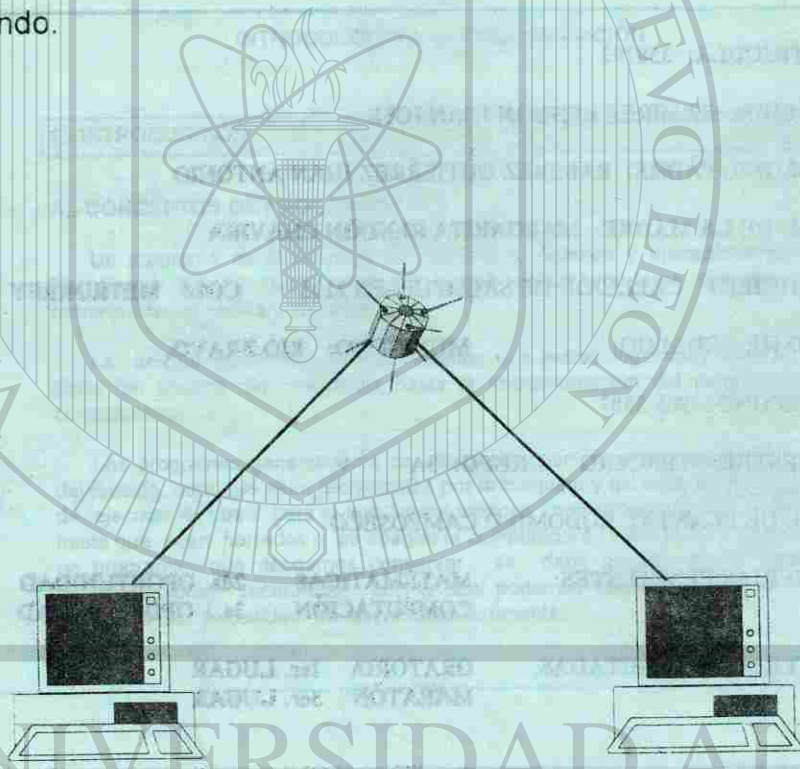
Ejemplo de un registro en una **BASE DE DATOS**

Los programas más utilizados en la administración de bases de datos son: dBASEIII, FOX, dBASE III plus, FOX BASE, FOX BASE plus, FOX PRO, FOX PRO Line, Progress, Oracle y otros más.

Comunicación

Los programas de aplicación en **comunicación** nos permiten transmitir mensajes de nuestra computadora a otra distante, por vía telefónica o por satélite, para lo cual se necesita un dispositivo llamado MODEM que significa MOdulador-DEModulador.

Las tareas que se pueden realizar mediante este tipo de software son ilimitadas y están al alcance de nuestras manos, pues por medio de nuestra computadora, podemos acceder la información de la base de datos de cualquier parte del mundo.



Ejemplo de una comunicación entre dos computadoras

Se requiere de un software que permita la comunicación entre dos o más computadoras, actualmente un ejemplo de esta aplicación sería el enlace por medio de INTERNET.

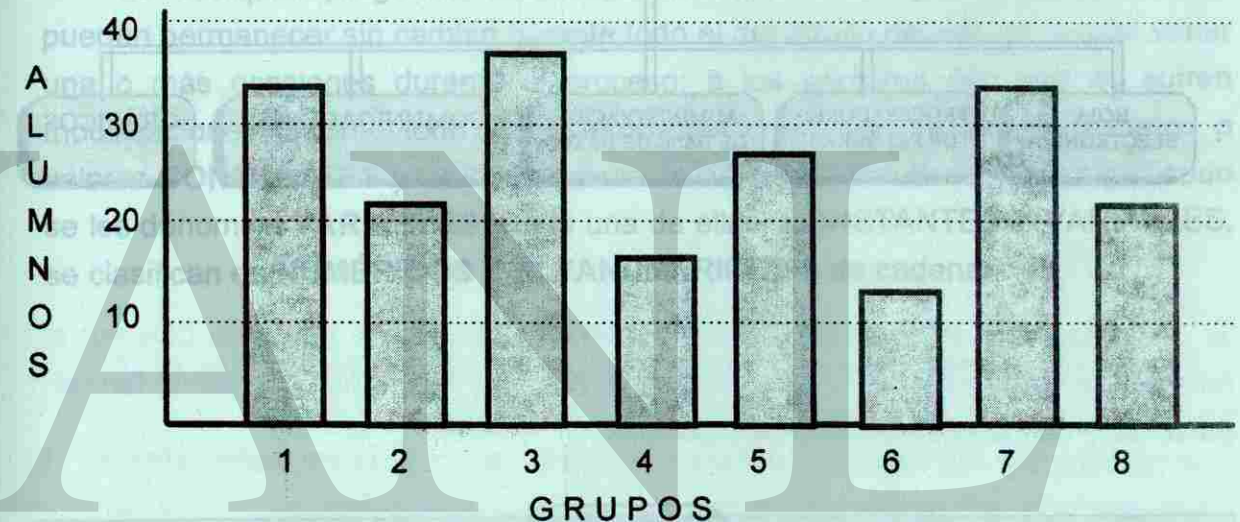
Gráficos

La posibilidad de convertir datos numéricos en **gráficos** es que sirva para comparar rápidamente unos datos contra otros y que expliquen una información demasiado larga de leer en una más sencilla y clara de entender para poder obtener algunas conclusiones sobre ellos.

la realidad.

Dentro de un programa, la computadora maneja dos tipos de datos: constantes y variables.

ALUMNOS APROBADOS EN FÍSICA



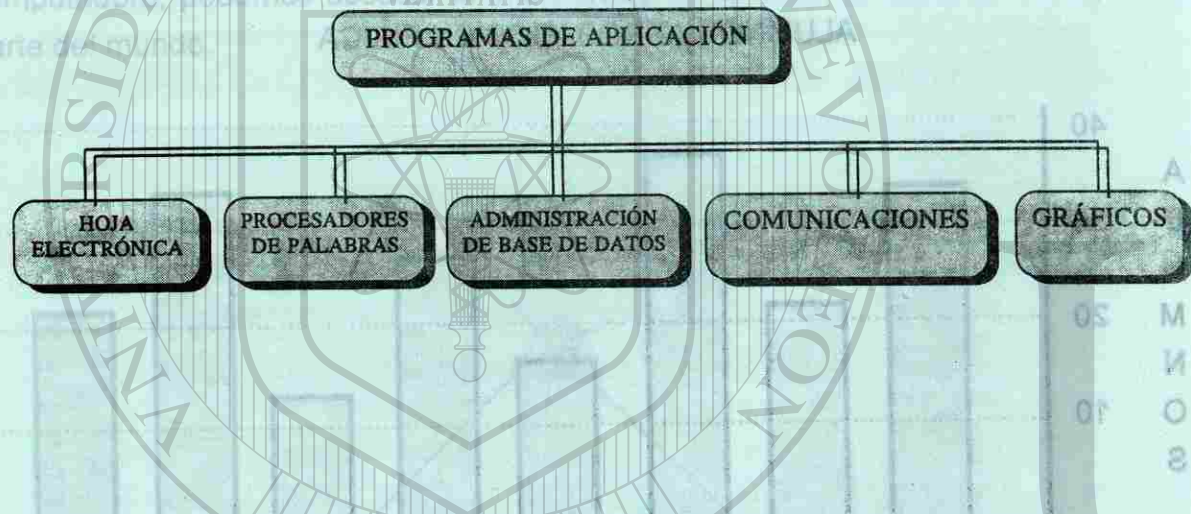
Ejemplo de un gráfico

En el mercado se pueden encontrar en los paquetes de graficadores tales como el Chart-Master, Harvard Graphics, Power Point, Applause II, Corel Draw y otros.

Algunas hojas electrónicas incluyen la posibilidad de reportar los datos contenidos en ella mediante gráfica de barras, poligonal, pastel y otras.

Paquetes Mixtos

Existen también algunos paquetes de programas de aplicación como el SYMPHONY o el WORKS que integran en el mismo software, una hoja electrónica, un procesador de palabras y una base de datos, o como el WORKS que además de los anteriores también incluye los programas para comunicación.



2.- METODOLOGÍA DE LA PROGRAMACIÓN

Al desarrollar un programa es conveniente tener una metodología, te recomendamos desarrollar los siguientes pasos:

- ① Análisis del problema o tarea.
- ② Algoritmo.
- ③ Diagrama de flujo.
- ④ Prueba de escritorio.
- ⑤ Codificación.
- ⑥ Depuración.

A.- ANÁLISIS DEL PROBLEMA O TAREA.

En el proceso de desarrollo de un programa debemos de efectuar, como primer paso, un **análisis del problema o tarea** que se requiere sea realizado por la computadora, pensar en las operaciones que se necesitan para resolverlo, los datos que se van a manejar dentro del programa y las implicaciones que tiene con la realidad.

Dentro de un programa, la computadora maneja dos tipos de datos: **constantes y variables.**

En cualquier programa se utilizan diferentes datos y valores, los cuales pueden permanecer sin cambio durante todo el desarrollo del mismo, o bien variar una o más ocasiones durante el proceso; a los primeros, los que no sufren modificaciones durante todo el desarrollo del programa, se les llama datos o valores **CONSTANTES** y los que sí cambian o modifican sus valores o significado se les denomina **VARIABLES**. Cada una de ellas, **CONSTANTES Y VARIABLES**, se clasifican en **NUMÉRICOS Y ALFANUMÉRICOS** o de cadena.



CONSTANTES

☞ **Constantes numéricas** son las figuras o símbolos que representan los mismos valores o números, para todas las personas que los lean, por ejemplo:

SÍMBOLOS	REPRESENTA
8	ocho unidades
15	quince unidades
π	3.141592....

Con estas figuras o símbolos se pueden realizar operaciones aritméticas (suma, resta, multiplicación, división, etc.).

☞ **Constantes alfanuméricas** o de cadena son figuras o símbolos que por si mismos, generalmente no representan algo, pero al unirse unas con otras, forman palabras y dan la idea de algo, por ejemplo:

SÍMBOLOS

árbol
pelota
libreta

REPRESENTAN



Con estas constantes no se pueden realizar operaciones aritméticas.

VARIABLES

☞ Las **variables numéricas** son las que guardan datos numéricos.

Por ejemplo en lenguaje BASIC tenemos:

VARIABLE	EJEMPLO DE VALOR ALMACENADO
A	243
PROM	56
C2	-2.45

La variable y el valor alacenoado para cada ejemplo quedaría:

A = 243
PROM = 56
C2 = -2.45

Por la función que realizan las variables numéricas se pueden clasificar en:

Contadoras, las cuales las utilizamos para contar algún tipo de suceso; se incrementan en un rango fijo, por ejemplo: cuando contamos los alumnos que están sentados en clase lo podemos hacer de uno en uno, 1, 2, 3, 4, 5, ..., la variable que guarda la cantidad de alumnos varía de uno en uno; o si lo hacemos de dos en dos, 2, 4, 6, 8, ..., la variable se modifica en un rango fijo de dos.

Por ejemplo en lenguaje BASIC tenemos:

$C = C + 1$
 $A = A + 2$
 $PROM = PROM - 1$
 $PR = PR - 5$

☑ **Acumuladoras** son las que están cambiando su valor durante el proceso, debido al resultado de una operación o de una condición, pero su incremento no es fijo; por ejemplo las horas trabajadas por una persona durante una semana: el Lunes 6 hrs, Martes 8 hrs, Miércoles 4 hrs, Jueves 7 hrs, Viernes 9 hrs, Sábado 4 hrs; la variable que lleva la cuenta se incrementa primero en 6, después en 8, luego en 4, etc. la variable va acumulando los valores obtenidos o dictados.

Por ejemplo en lenguaje BASIC tenemos:

```
HORAS = HORAS + N
SUM = SUM + CAL
B = B + C
```

Cada lenguaje de programación tiene sus propias reglas en el uso de los tipos de variables tanto para valores numéricos como valores alfanuméricos.

☒ Las **variables alfanuméricas** son las que guardan datos de cadena o alfanuméricos también se les conoce como string.

Por ejemplo en lenguaje BASIC tenemos:

VARIABLE	EJEMPLO DE VALOR ALMACENADO
A\$	" UNIVERSIDAD "
NOM\$	" Denisse "
B\$	" TIGRES "

La variable y el valor almacenado para cada ejemplo quedaría:

A\$ = " UNIVERSIDAD "

NOM\$ = " Denisse "

B\$ = " TIGRES "

B.- ALGORITMOS

Al conjunto de pasos que, siguiendo un orden lógico, describe la solución de un problema o el desarrollo de un proceso se le llama **algoritmo**.

El algoritmo, al describir los pasos para resolver un programa, nos permite diseñar como **enseñar a la computadora la realización del proceso de solución**. La máquina no sabe como realizar las tareas, quien le enseña es el programador.

Los algoritmos están escritos en lenguaje materno, en nuestro caso el Español. Se considera que los algoritmos se escriben en un pseudocódigo.

Las características de un algoritmo son:

- ☒ Tiene principio y fin.
- ☒ Tiene una cantidad finita de pasos.
- ☒ Los pasos están secuencialmente ordenados.
- ☒ Cada paso está concatenado con el anterior, menos el primero.
- ☒ Cada paso está concatenado con el posterior, menos el último.
- ☒ Todos los pasos buscan solucionar el problema.

C.- DIAGRAMA DE FLUJO

A la representación gráfica del algoritmo se le llama **DIAGRAMA DE FLUJO** el cual sirve para que mediante dibujos y pequeñas notas, se vea el desarrollo de la solución del problema.

Las herramientas utilizadas para éste propósito se llaman bloques o plantillas, las más usuales son las que se muestran en la siguiente página.

FIGURA

FORMA

FUNCIÓN o SIGNIFICADO

ELIPSE



PRINCIPIO o FIN

Se usa para indicar en donde comienza o termina el programa.

RECTÁNGULO



OPERACIÓN o PROCESO

Con esta figura se indica cualquier proceso interno de la computadora, una transferencia de datos u operación aritmética, la operación específica se indica dentro del rectángulo.

RECTÁNGULO (sin esquina)



ENTRADA o LECTURA

Representa una operación de entrada o lectura de datos.

ROMBO

o

TRIÁNGULO



DECISIÓN (falso o verdadero)

Se utiliza para representar una verificación o comparación lógica. Se efectúa una pregunta con respuesta de falso o verdadero.



FIGURA

FORMA

FUNCIÓN o SIGNIFICADO

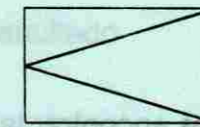
CÍRCULO



CONECTOR

Este símbolo nos indica hacia donde debe continuar el flujo del programa.

RECTÁNGULO (dividido en tres triángulos)



PROCESO REPETITIVO

Se utiliza para contar las veces que se repite un proceso determinado.

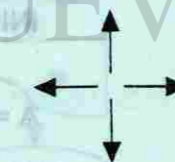
TRAPEZIO (deformado)



SALIDA DE RESULTADOS

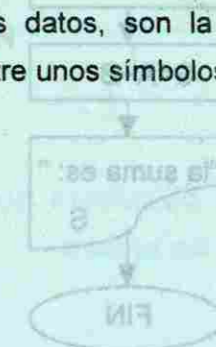
Imprime los resultados del proceso.

FLECHAS



FLUJO DE DATOS

Con esta figura se indica como fluyen los datos, son la conexión lógica entre unos símbolos y otros.



D.- PRUEBA DE ESCRITORIO.

La prueba de escritorio consiste en dar valor a las variables marcadas en el diagrama de flujo y comprobar si el resultado obtenido es el esperado.

El algoritmo, el diagrama de flujo y la prueba de escritorio los utilizaremos al mismo tiempo para explicarlos con ejemplos los tres tipos de programas.

PROGRAMAS SECUENCIALES.- Los problemas que se resuelven de una manera continua, sin hacer brincos o cambios de dirección, les denominaremos programas **SECUENCIALES**.

Ejemplo No.1.- Dados dos números encontrar la suma de ellos.

ALGORITMO

- 1.- Inicio
- 2.- Pedir dos números
- 3.- Efectuar la suma
- 4.- Imprimir el resultado de la suma
- 5.- Fin

DIAGRAMA DE FLUJO



PRUEBA DE ESCRITORIO

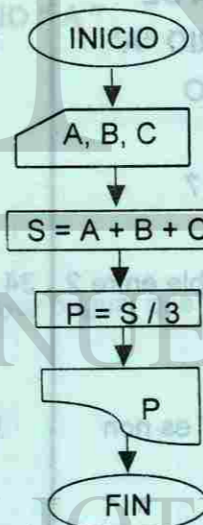
INICIO
 A = 5, B = 12
 S = 5 + 12 = 17
 La suma es 17
 FIN

Ejemplo No.2.- A partir de tres evaluaciones parciales, obtener el promedio de ellas.

ALGORITMO

- 1.- Inicio.
- 2.- Pedir las tres calificaciones.
- 3.- Efectuar la suma de las calificaciones.
- 4.- Obtener el promedio.
- 5.- Imprimir el resultado
- 6.- Fin.

DIAGRAMA DE FLUJO



PRUEBA DE ESCRITORIO

INICIO
 A = 70, B = 90, C = 50
 S = 70 + 90 + 50 = 210
 P = 210 / 3 = 70
 73.33
 FIN

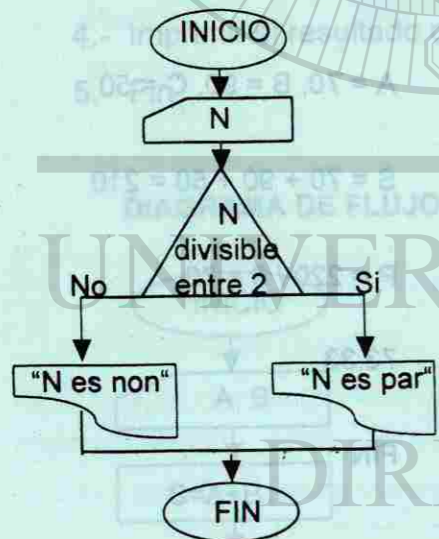
PROGRAMAS CONDICIONALES.- Los problemas en donde, en alguno de sus pasos se presenta la necesidad de seleccionar entre dos caminos (bifurcación), de los cuales sólo se toma uno de ellos, les denominaremos programas **CONDICIONALES**; esto se presenta mediante una pregunta o cuestionamiento que da como respuesta un verdadero (si) o un falso (no) para seguir uno de los caminos.

Ejemplo No. 1.- Determinar si un número es par o non.

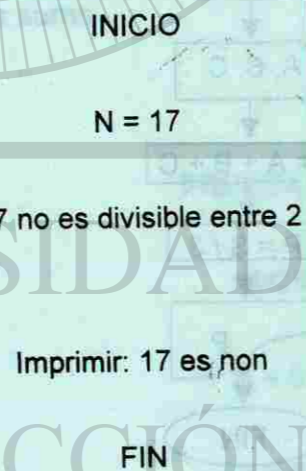
ALGORITMO

- 1.- Inicio.
- 2.- Pedir un número (N)
- 3.- **SI** el número es divisible entre 2 y al paso 4 y si **NO** ir al paso 5
- 4.- Imprimir N es par y después ir al paso 6
- 5.- Imprimir N es non
- 6.- Fin

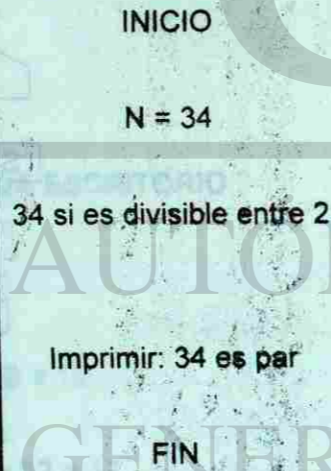
DIAGRAMA DE FLUJO:



PRUEBA DE ESCRITORIO "A"



PRUEBA DE ESCRITORIO "B"



Nota: Se considera que un número es divisible entre otro, si la respuesta es un entero.

Ejemplo No.2.- Programa que sabiendo la calificación de un alumno nos indica si está aprobado o reprobado (La calificación mínima para aprobar es 70).

ALGORITMO

- 1.- Inicio.
- 2.- Pedir la calificación.
- 3.- **SI** es mayor o igual que 70 ir al paso 6 y si **NO** ir al paso 4.
- 4.- Imprimir REPROBÓ e ir al paso 7.
- 5.- Ir al paso 7.
- 6.- Imprimir APROBÓ.
- 7.- FIN

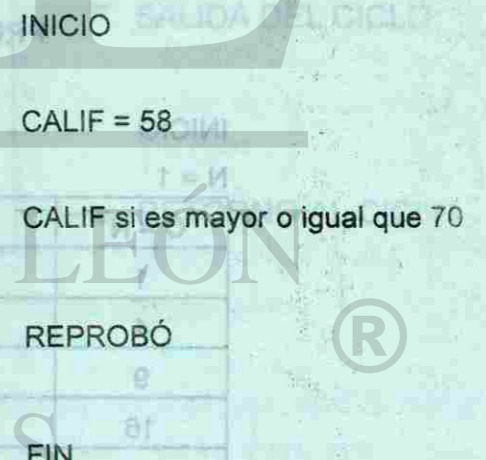
DIAGRAMA DE FLUJO:



PRUEBA DE ESCRITORIO "A"



PRUEBA DE ESCRITORIO "B"



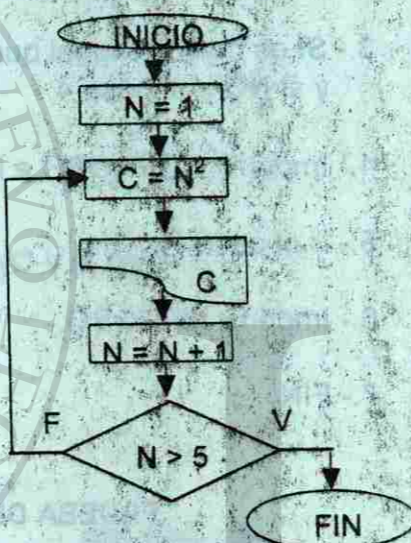
PROGRAMAS CÍCLICOS.- En éstos existen una o varias acciones o pasos que se repiten, mientras se cumpla una condición dada. A éste tipo de problemas de procesos repetitivos lo llamaremos programas **CÍCLICOS**.

Ejemplo No.1.- Programa que encuentra e imprime los cuadrados de los 5 primeros números naturales.

ALGORITMO

- 1.- Inicio.
- 2.- El primer número es 1.
- 3.- Encuentro el cuadrado del número.
- 4.- Imprimo el cuadrado del número.
- 5.- Número se incrementa en 1.
- 6.- Si el número es mayor que 5 ir al paso 8 y si NO ir al paso 7.
- 7.- Ir al paso 3.
- 8.- Fin

DIAGRAMA DE FLUJO



PRUEBA DE ESCRITORIO

INICIO

N = 1

C = N ²	C	N = N + 1	N > 5
1	1	2	2 > 5
4	4	3	3 > 5
9	9	4	4 > 5
16	16	5	5 > 5
25	25	6	6 > 5

FIN

Nota: Se considera que un número es divisible entre otro cuando el cociente es un entero.

En el siguiente ejercicio utilizaremos, en el diagrama de flujo, un block o plantilla que representa tres acciones:

Inicio del contador

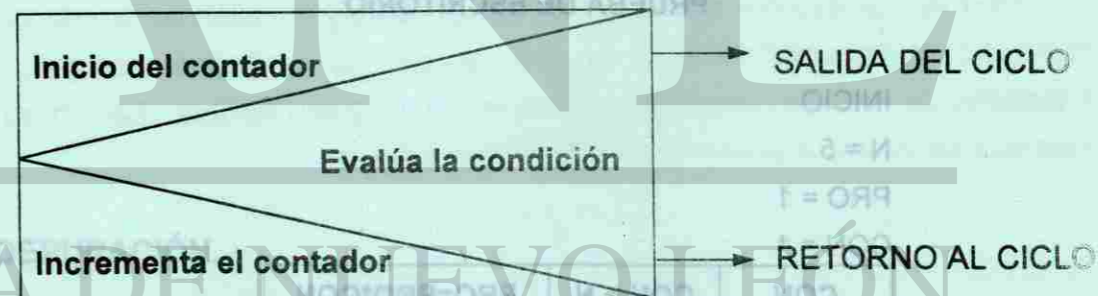
Se utiliza una variable o expresión para iniciar el proceso de contar.

Evalúa la condición

Se utiliza una o varias expresiones para determinar si se continúa con el proceso o terminar la ejecución del mismo.

Incrementa el contador en el rango determinado.

Se utiliza para incrementar el contador del proceso, puede incrementarse de uno en uno, de dos en dos, etc.

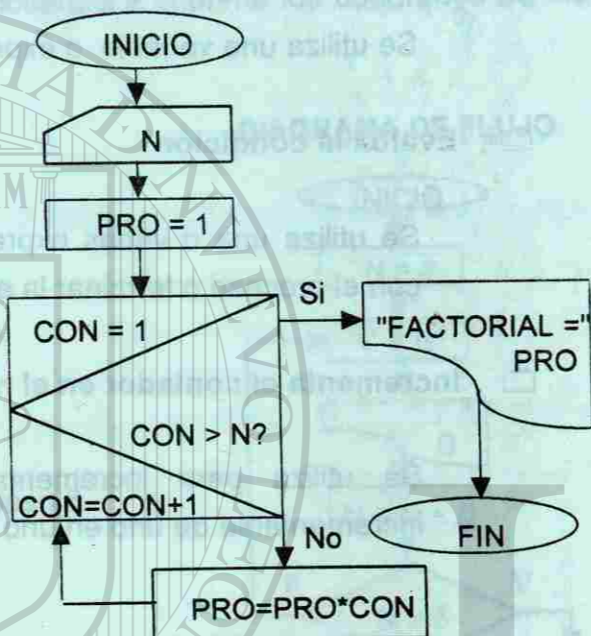


Ejemplo No. 2.- Determina el factorial de un número dado.

ALGORITMO

- 1.- Inicio
- 2.- Pedir el número
- 3.- Inicializar Producto = 1
- 4.- Inicia contador en 1
- 5.- **SI** el contador es mayor que N ir al paso 9 y si **NO** ir al paso 6
- 6.- Producto igual a producto por contador
- 7.- Incrementar el contador
- 8.- Ira al paso 5
- 9.- Imprimir el resultado
- 10.- Fin

DIAGRAMA DE FLUJO



PRUEBA DE ESCRITORIO

INICIO

N = 5

PRO = 1

CON = 1

CON	CON > N	PRO = PRO * CON
1	1 > 5	1 * 1 = 1
2	2 > 5	1 * 2 = 2
3	3 > 5	2 * 3 = 6
4	4 > 5	6 * 4 = 24
5	5 > 5	24 * 5 = 120
6	6 > 5	

PRO = 120

FIN

E.- CODIFICACIÓN DEL PROGRAMA.

Los cuatro primeros pasos: análisis, algoritmos, diagramas de flujo y la prueba de escritorio en cualquier problema de programación, son básicos sin importar el lenguaje en que se programe; **la codificación** consiste en pasar las instrucciones del lenguaje común al lenguaje máquina, y esto depende de las reglas del propio lenguaje que estemos utilizando.

Al codificar un programa, lo que estamos haciendo es enseñar a la computadora a realizar un problema o tarea; para que la máquina pueda aprenderlo necesita que sea en un lenguaje que ella entienda: BASIC, COBOL, FORTRAN, PASCAL, "C", VISUAL BASIC o cualquier otro.

En la siguiente unidad veremos cómo codificar en el lenguaje BASIC, explicando los comandos, ordenes o instrucciones que en este lenguaje se utilizan, así como la sintaxis correcta para su uso.

F.- DEPURACIÓN.

Al proceso de corrección del programa se le llama **depuración**. ®

Después de haber codificado un programa debemos verificar, que lo hemos tecleado correctamente, que la codificación fue realizada de manera adecuada y que no cometimos errores en los pasos anteriores; en fin, que con las instrucciones que hemos escrito, la computadora realizará adecuadamente su tarea y para esto lo que haremos es ejecutar o correr el programa.

Existen dos tipos de errores: los de sintaxis y los de lógica.

El **error de sintaxis** se detecta al correr el programa; la máquina empieza a leer cada una de las órdenes dictadas y si las entiende, las realiza; si no es así, se interrumpe la ejecución y nos despliega un mensaje de **ERROR**, indicándonos en dónde y de que tipo es; esto nos permite regresar al programa para hacer los cambios pertinentes en las instrucciones donde se cometió el error y, posteriormente, ejecutar de nuevo el programa para detectar algún otro error.

El **error de tipo lógico** se detecta cuando el programa se ejecuta totalmente, sin interrupciones, pero el resultado obtenido no es el esperado; esto nos indica que fue codificado en forma adecuada pero el uso inadecuado de las operaciones, variables o asignaciones que se utilizaron en la solución del problema provocaron el error; para corregir este tipo de errores debemos revisar nuestro algoritmo y nuestro diagrama de flujo y comprobar si la lógica que utilizamos fue adecuada, si las operaciones están bien definidas, si la asignación de datos es la correcta o si el tipo de variables utilizado corresponden a los datos asignados.

Una vez concluidos estos seis pasos de la programación podemos, como se mencionó al principio de esta unidad, guardar el programa realizado en un dispositivo de almacenamiento secundario (en un disco flexible, disco duro, etc.) para que se vuelva a utilizar cuando sea conveniente.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

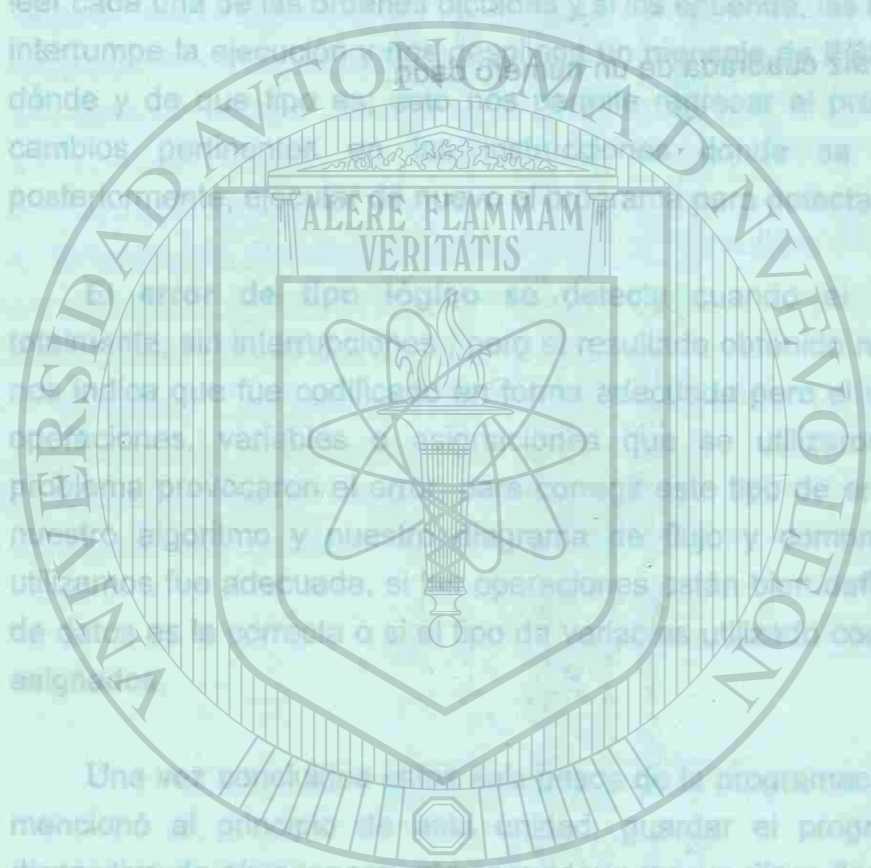
DIRECCIÓN GENERAL DE BIBLIOTECAS

☺ **EJERCICIO GENERAL:**

Desarrolla el algoritmo, el diagrama de flujo y la prueba de escritorio para los siguientes programas:

- 1.- Determina la raíz cuadrada de un número dado.

2.- Encuentra el área de un rectángulo, dada la longitud de la base y la altura.



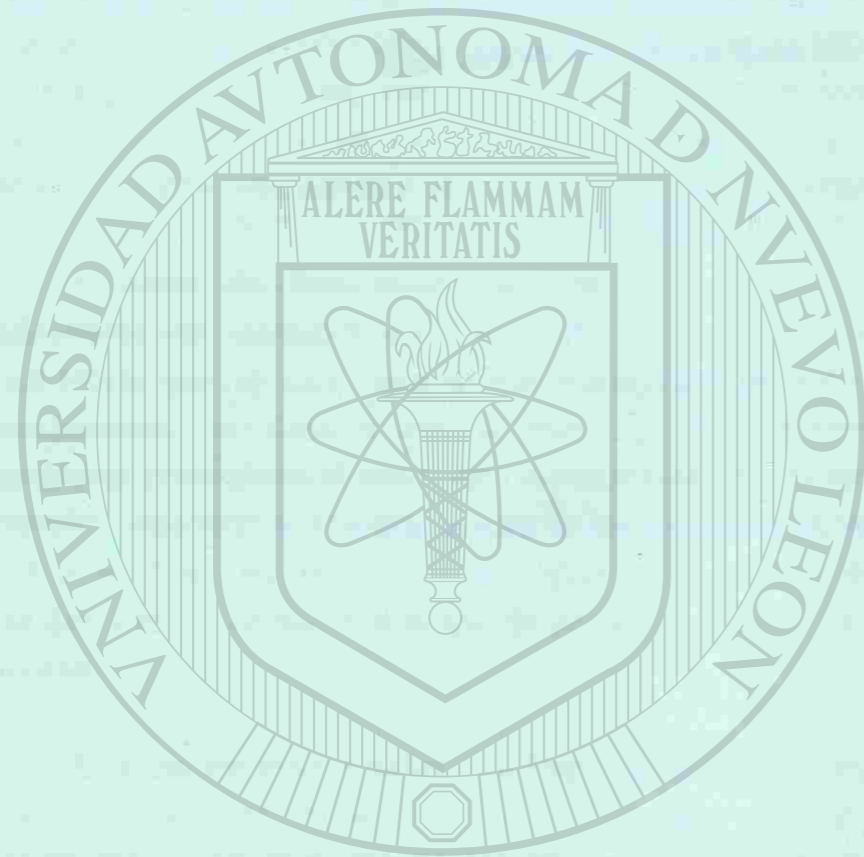
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

3.- En una librería se le hace el 10% de descuento a los estudiantes, determina a partir del pago de libros y el tipo de persona (estudiante o no estudiante) la cantidad total a pagar por persona.

U A N L

4.- Al pedir la edad de una persona que imprima "es una persona madura", si es de 30 años o más, de lo contrario que imprima "es una persona joven".



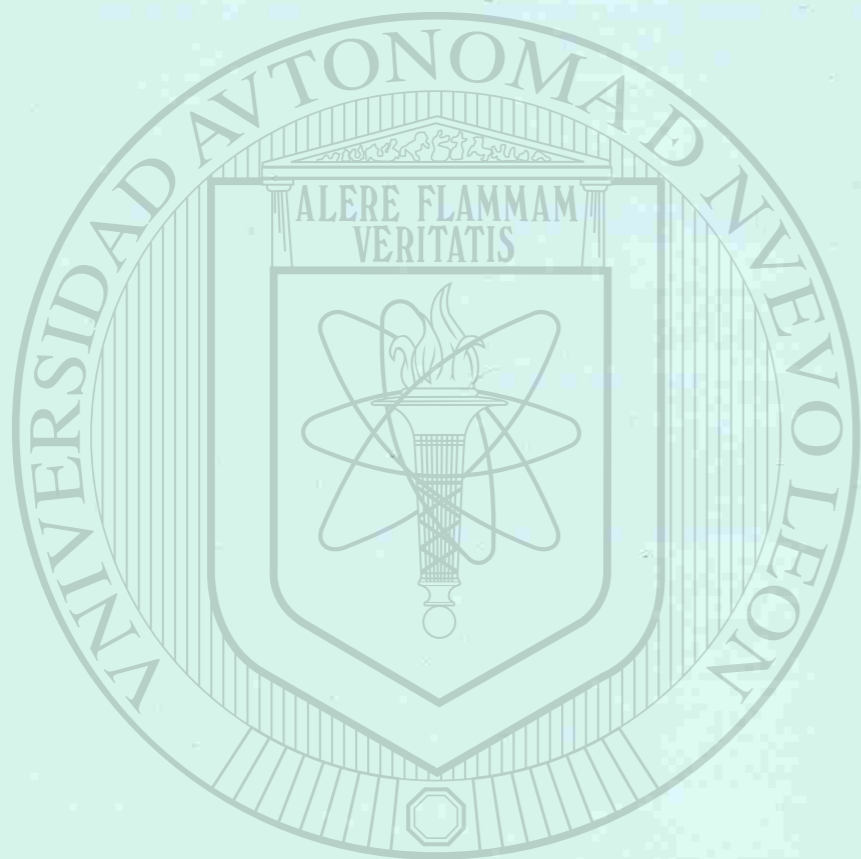
U A N L

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

6.- Desarrollar la tabla de multiplicar del número cinco.



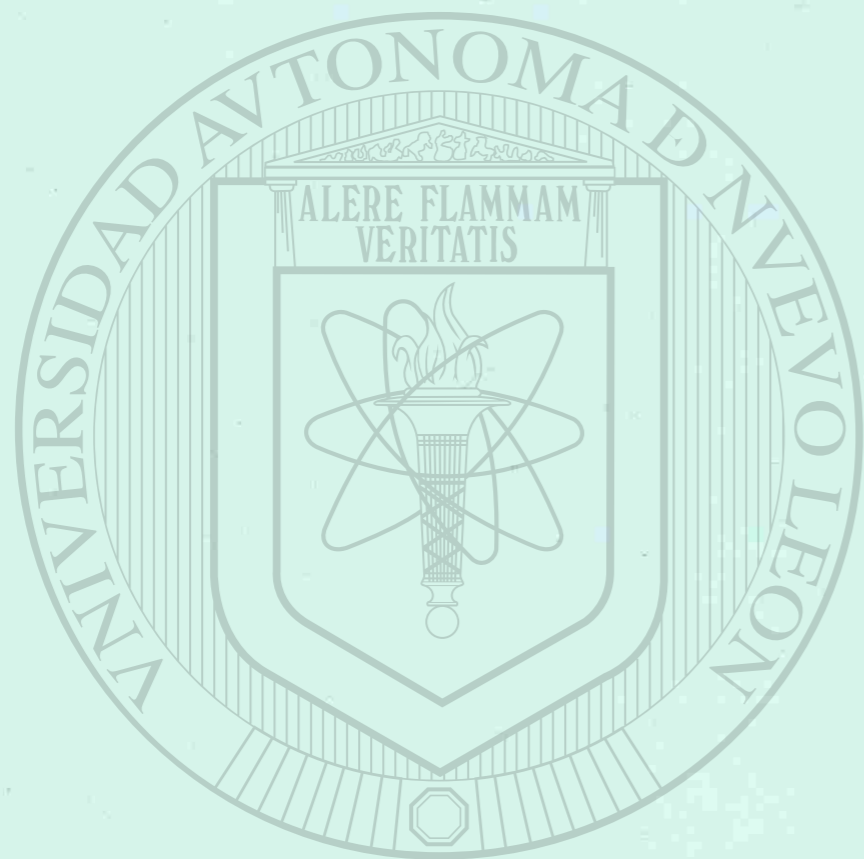
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

7.- Convertir un número de pulgadas en centímetros.

1 pulgada = 2.54 cm.

8.- En una tienda, los vendedores trabajan por sueldo base más una comisión. El sueldo base es de \$ 250.00 por semana, más el 3% de comisión sobre la venta si ésta es más de \$5,000.00 . Determina el salario del empleado en base a la venta realizada por semana.



UNANL

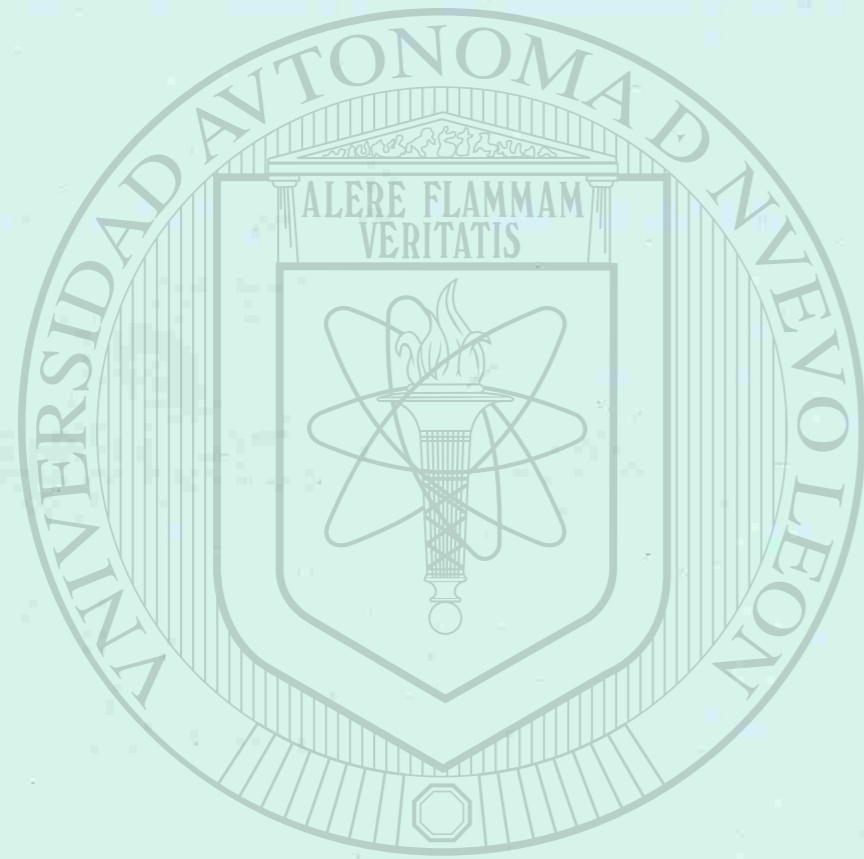
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

PROGRAMACIÓN BÁSICA

9.- Desarrollar una tabla de conversiones de pesos a dólares, a partir de \$1.00 hasta \$100.00

8.- En una tienda, los vendedores trabajan por sueldo base más una comisión
 10.- Dados dos números, si ambos son mayores que 10, que imprima la suma de ellos; si uno de ellos no es mayor que 10 que imprima el producto.
 base a la venta realizada por semana.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

PROGRAMACIÓN BASIC

UNIDAD IV

PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN

A.- CONCEPTOS DE BASIC

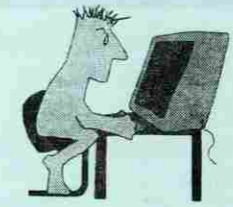
En este curso se aplicará de nuestro texto, emplearemos exclusivamente al lenguaje BASIC. Este lenguaje es fundamental por la que decidimos utilizarlo como lenguaje de enseñanza debido a su sencillez y facilidad para la comunicación. Debido a la sencillez de su lenguaje, ya que las instrucciones están en palabras técnicas del inglés, que permite al alumno una asimilación inmediata.

BASIC fue desarrollado en 1963 por John Kemeny y Thomas Kurtz en Estados Unidos, su característica principal es la facilidad que ofrece al programador.

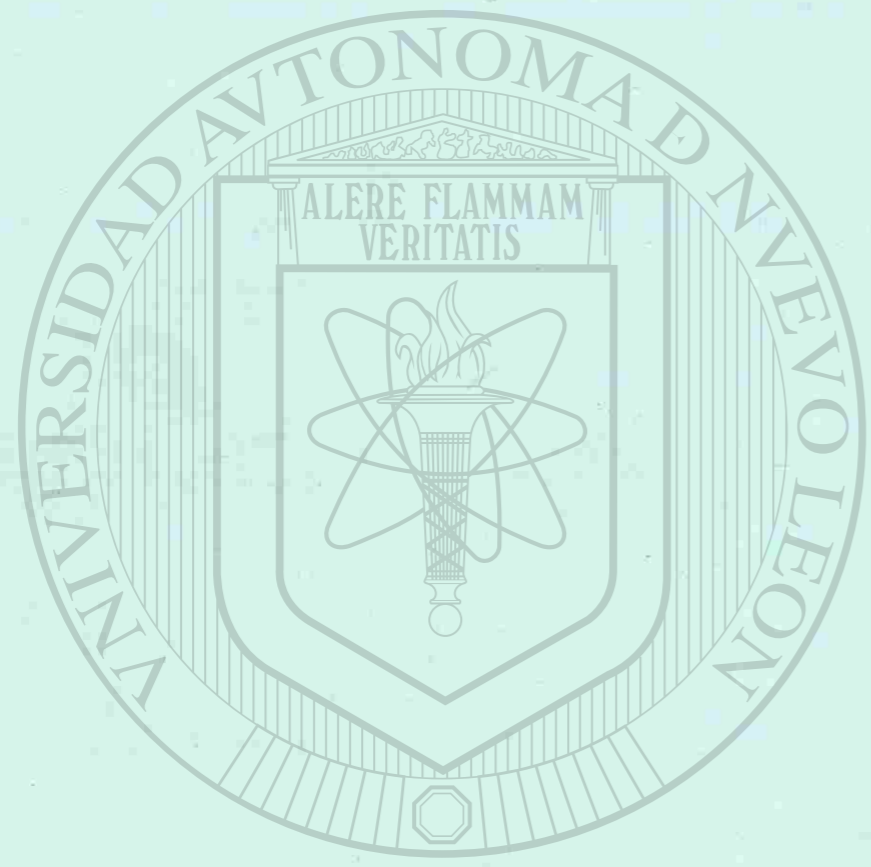
Las siglas de BASIC significan:

- Beginner (de principiante)
- All purpose (todopropósito)
- Simple (sencillo)
- Instruction (instrucción)
- Code (código)

propósitos para principiantes.



10.- Dados dos números, si ambos son mayores que 10, que imprima la suma de ellos; si uno de ellos no es mayor que 10 que imprima el producto.



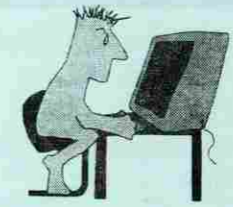
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN

A.- CONCEPTOS DE BASIC



En este curso... En este curso se emplearán exclusivamente el lenguaje BASIC... debido a la sencillez de su lenguaje, ya que las instrucciones... técnicas del Inglés, que permite al alumno un aprendizaje inmediato.

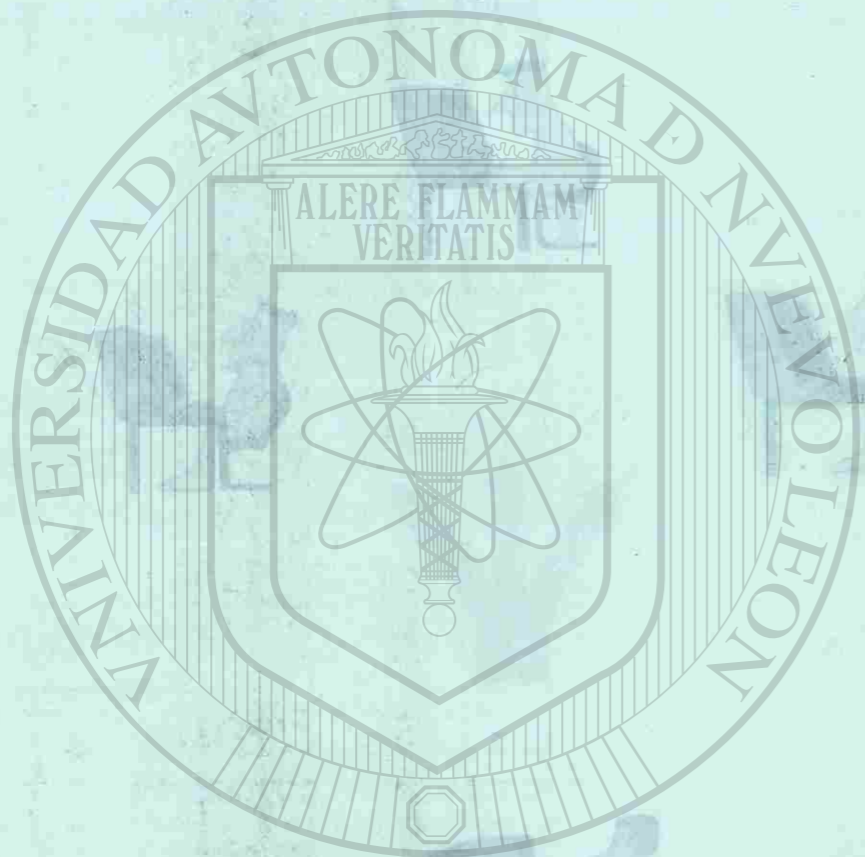
BASIC fue desarrollado en 1963 por John Kemeny y Thomas Kurtz en Estados Unidos, su características principales es la facilidad que ofrece al programar.

Las siglas de BASIC significan:

- Beginner (de primer nivel)
- All purpose (de propósito general)
- Simple (sencillo)
- Instruction (instrucción)
- Code (código)



10.- Dados dos números, si ambos son mayores que 10, que imprima la suma de ellos, si uno de ellos no es mayor que 10 que imprima el producto.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

PROGRAMACIÓN BASIC

UNIDAD IV

PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN.

A.- CONCEPTOS DE BASIC

En este cuarto capítulo de nuestro texto, emplearemos exclusivamente el lenguaje BASIC. La razón fundamental por la que decidimos utilizar este tipo de lenguaje es su operatividad, pues facilita la comunicación entre hombre-máquina, debido a la sencillez de su manejo, ya que las instrucciones se dan en palabras técnicas del inglés, que permite al alumno un acercamiento inmediato.

BASIC fue desarrollado en 1963 por John Kemeny y Thomas Kurtz en Estados Unidos, su característica principal es la facilidad que ofrece al programar.

Las siglas de BASIC significan:

- Beginner's (de principiante)
- All-purpose (todo propósito)
- Symbolic (simbólico)
- Instruction (instrucción)
- Code (código)

Traducida resultaría : Código de instrucciones simbólicas para todos los propósitos para principiantes.

2.- ESTRUCTURA DE UN PROGRAMA

Un programa en BASIC está formado por:

- A.- Número de línea.
- B.- Instrucciones.
- C.- Documentación del programa.
- D.- Uso de constantes.
- E.- Asignación de variables.
- F.- Terminación de línea.

A.- NÚMERO DE LÍNEA.

Son números naturales usados como referencia para la secuencia del programa, generalmente marcados de diez en diez para dejar la opción de intercalar instrucciones posteriores, sirven además como etiqueta informativa al hacer transferencia de control.

Si después de haber realizar el programa se requiere introducir otra instrucción **12 instrucción**, se puede teclear al final del programa y BASIC lo reacomodará de acuerdo a su número de línea.

Ejemplo

10 instrucción
20 instrucción
30 instrucción
40 instrucción
... ..

Ejemplo modificado

10 instrucción
12 instrucción
20 instrucción
30 instrucción
40 instrucción
... ..

B.- INSTRUCCIONES.

Las instrucciones para realizar un programa en lenguaje BASIC pueden ser de dos tipos: **Operaciones y Estatutos.**

OPERACIONES.- Pueden ser **Aritméticas y Lógicas.**

Operaciones aritméticas.- Parte del programa donde se evalúan todas las expresiones matemáticas que requieren una operación. Los operadores aritméticos son:

Operador	Operación
+	suma
-	resta
*	multiplicación
/	división
\	división entera
^	exponenciación
MOD	el residuo de la división

La división entera se representa por la diagonal invertida \, al contrario de la barra inclinada / que representa la división de dos números. Esta operación convierte al dividendo y divisor en enteros, redondeándolos si fuera necesario, antes de efectuar la división propiamente dicha y, tras su ejecución, el cociente se redondea de igual manera.

Ejemplos: $323.89 \setminus 9 = 36$
 $427.70 \setminus 52 = 8$

El operador MOD proporciona el residuo de una división, pero antes de realizar la división, tanto el dividendo como el divisor se redondean como en la división entera.

Formato : Dividendo MOD Divisor

- Ejemplos:
- 7 MOD 4 La respuesta es: 3
 - 18.6 MOD 3 La respuesta es: 1
 - 75 MOD 8.9 La respuesta es: 3
 - 8 MOD 2 La respuesta es: 0
 - 750 MOD 25 La respuesta es: 0

Los operadores aritméticos son ejecutados bajo la siguiente jerarquía:

Jerarquía	Operación
1 ^a ^	exponenciación
2 ^a * /	multiplicación división
\	división entera
MOD	residuo de la división
3 ^a + -	suma y resta

Los operadores de igual jerarquía se evalúan de izquierda a derecha. En una expresión que incluya dos o más operadores aritméticos, los de jerarquía más alta se ejecutan antes de que los de menor jerarquía. Este orden de evaluación se puede anular por el uso de paréntesis, de modo que cualquier operador dentro de ellos se ejecuta primero.

Si la expresión tiene paréntesis, las operaciones contenidas en ella se calcula en primer lugar; por lo tanto, se puede cambiar la secuencia de evaluación o el valor de la expresión aritmética, encerrando parte de las expresiones en paréntesis.

- Ejemplos:
- a) $2 + 5 * 3 + 7 - 12 / 4 + 3 * 3 ^ 2$
 $2 + 5 * 3 + 7 - 12 / 4 + 3 * 9$
 $2 + 15 + 7 - 3 + 27$
 48
 - b) $(2 + 5) * 3 + (7 - 12) / 4 + (3 * 3) ^ 2$
 $7 * 3 - 5 / 4 + 9 ^ 2$
 $7 * 3 - 5 / 4 + 81$
 $21 - 1.25 + 81$
 100.75

Nota: La existencia de paréntesis anidados implica iniciar la solución de los más internos hacia fuera.

Operadores Lógicos .- Son los relacionales y los booleanos.

Los operadores relacionales son:

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

- Ejemplos:
- A < 100
 - NUM >= 70
 - AB1 <> 500

Los operadores booleanos son:

- NOT complemento lógico
- AND conjunción, "Y"
- OR disyunción, "O"

Ejemplos:

A > 100 AND NUM >= 70

AB1 <> 500 OR CD2 <= 800

Los operadores lógicos, tanto los relacionales como los booleanos, tienen igual prioridad; si varios de ellos están presentes en la misma expresión, se evalúan de izquierda a derecha. En el caso de que en una expresión existan operadores aritméticos y lógicos, los operadores lógicos se ejecutan después de los aritméticos.

ESTATUTOS. - Son palabras reservadas en el lenguaje BASIC que indican una orden específica. Los estatutos o sentencias los mencionaremos más adelante, daremos su sintaxis y las formas de como utilizarlos.

Ejemplos de estatutos: REM, PRINT, INPUT, LET, CLS, GOTO, etc.

C.- DOCUMENTACIÓN DEL PROGRAMA.

Este aspecto no es ejecutado por la computadora; sólo permite al programador introducir comentarios al programa tales como: especificar objetivos, quien lo elaboró, inicio de un proceso determinado, etc.

Ejemplo: **REM** Este programa determina como aprobar el curso.

D.- USO DE CONSTANTES.

Las constantes, como ya lo tratamos en el capítulo anterior, son valores declarados en los programas y que no cambian durante el proceso. Existen dos tipos: **Alfanuméricas** y **Numéricas**.

Constantes alfanuméricas o strings. - En el lenguaje BASIC éstas constantes pueden contener hasta 255 caracteres: alfabéticos, combinación de letras y números o números solamente. Deberán encerrarse entre comillas y no es posible realizar operaciones aritméticas o cálculos con ellas.

Ejemplos: "UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN"

"Col. Roble."

"Madero 2315 Pte."

"Sierra de Santa. Clara"

Constantes numéricas. - Representan valores numéricos que pueden ser negativas o positivas; a las constantes numéricas positivas, se les puede suprimir el signo, pero la negativas siempre deben especificarse.

Ejemplos: 10 - 60.25

- 22 37.12

1563 - 53.295

- 135 874.395

E.- ASIGNACIÓN DE VARIABLES.

Nombre simbólico que permite almacenar información en un espacio de la memoria, la cual puede representar valores alfanuméricos o numéricos.

Las reglas para la formación de variables son:

- ☐ Longitud máxima de 40 caracteres.
- ☐ Siempre debe empezar con un caracter alfabético.
- ☐ Sólo puede almacenar valores de su mismo tipo.
- ☐ No pueden emplearse palabras reservadas como: RUN, TO, IF, etc.

En el lenguaje BASIC, utilizaremos tres tipos de variables: **Reales, Enteras y Alfanuméricas (Strings o Cadenas)**

☞ **Variables reales.**- Su valor puede contener valores enteros y decimales.

Ejemplos: B = -587.734
DOM = 41416794456
AB2 = 9

☞ **Variables enteras.**- Son aquellas que utilizamos para guardar números enteros y deben ir acompañadas del signo %.

Ejemplos: B% = -699
N% = 12
R% = 167

☞ **Variables alfanuméricas o strings.**- Son todas aquellas variables que contengan una cadena de caracteres, pueden ser letras, números o ambos; aunque contengan un número, éste no representa un valor numérico. Las variables strings deberán acompañarse al final con un signo \$ y su valor tendrá que encerrarse entre comillas.

Ejemplo: DOM\$ = " Secretaría Académica "

A\$ = "34"

B\$ = "50"

SUM\$ = A\$ + B\$

En el último ejemplo, una variable strings puede estar en función de otra variable alfanumérica; y tal como se mencionó en el párrafo anterior, el valor de SUM\$ sería igual a "3450". El espacio en blanco dentro de las comillas también se toma como un caracter.

Nota: BASIC respeta todo aquello escrito entre comillas, dejándolo tal como se escribió.

F.- TERMINACIÓN DE LÍNEA.

En este lenguaje todos los estatutos, órdenes y nombres de los programas pueden escribirse con mayúsculas o minúsculas, pero BASIC los cambia a mayúsculas. Al realizar una corrección y al finalizar una línea del programa se debe pulsar **Intro, Return o Enter**.

3.- CÓMO CARGAR BASIC.

Para cargar BASIC, lo primero que necesitamos es estar dentro de la red, una vez que el PROPMT de tu estación de trabajo se encuentra en **G:\ALUMNOS\ESTxx>** tecllea **BASIC**, con mayúsculas o con minúsculas, y pulsa la tecla Enter (↵), con ésta instrucción lo llamará y aparecerá una pantalla semejante a la siguiente figura, si por algún motivo te marca un error o no puedes cargar el lenguaje BASIC pregúntale a tu maestro en que lugar del SERVIDOR se encuentra.

GW-BASIC X.xx
© Copyright Microsoft 1983,1984
60131 Bytes free

Ok

1 LIST 2 RUN 3 LOAD 4 SAVE 5 CONT 6 "LPT" 7 TRON 8 TROFF 9 KEY 0 SCREEN

En la parte superior de la pantalla, aparece la información sobre el lenguaje y la versión del mismo; en la parte inferior se destacan diez cuadros que nos indican las órdenes que se pueden ejecutar con las teclas de las funciones F1, F2, F3, ..., F10. Explicaremos algunas de ellas en la siguiente página.

Lista de acciones que realiza la computadora con el lenguaje BASIC

Tecla	Instrucción	Acción a realizar
F1	List	Lista el programa que está en memoria.
F2	Run	Ejecuta (corre) el programa que está en memoria.
F3	Load	Carga un programa que está en la unidad de disco.
F4	Save	Guarda en la unidad de disco un programa.
	New	Borra el programa que está en memoria.
	Files	Muestra los archivos de la unidad de disco actual.
	System	Salir de BASIC

Con las teclas F1, F3, y F4, después de pulsarlas hay que pulsar Enter para que se lleven a cabo, solo con la tecla F2 se ejecuta la acción sin tener que pulsar Enter.

La forma de utilizar cada una de las instrucciones, las veremos en el siguiente tema.

Nota: Antes de iniciar con la práctica del siguiente tema, es necesario que les des formato a tu disco de trabajo y lo dejes sin ningún archivo.

4.- PROGRAMACIÓN BASIC.

A.- MODOS DE TRABAJAR EN BASIC

En BASIC existen dos maneras de trabajar: **MODOS DIRECTO** y **MODOS PROGRAMA**.

☐ **MODOS DIRECTO**.- Se teclean las instrucciones sin número de línea y la computadora ejecuta la orden de manera inmediata, por ejemplo tecléa la siguiente instrucción y escribe sobre la línea lo que aparece en la pantalla:

PRINT "Buenos días" (↵) _____

PRINT "Estamos en clase de computación" (↵) _____

PRINT "MI nombre es" (↵) _____

☐ **MODOS PROGRAMA**: En este modo las instrucciones se inician con un número entero positivo (natural), las instrucciones no se ejecutan, ya que el BASIC las guarda para ser ejecutadas posteriormente, por ejemplo tecléa:

10 PRINT "Buenos días" (↵)

20 PRINT "Estamos en clase de computación" (↵)

30 PRINT "MI nombre es" (↵)

Como te diste cuenta al teclear ENTER la computadora no realizó ninguna acción ya que ahora las instrucciones son guardadas.

Pulsa la tecla F2 y ¿ Qué aparece en la pantalla?

Pulsa la tecla F1 y pulsa Enter, ¿ Qué aparece en la pantalla?

Teclea: 15 PRINT " o Buenas tardes " (↵), Pulsa la tecla F1y después F2

¿ Qué aparece en la pantalla?

Realiza los siguientes pasos, para guardar el programa en tu disco de trabajo.

- ① Pulsa la tecla F4
- ② Después de la instrucción **SAVE** escribe **B:PAG106** y pulsa Enter.

Observa como se enciende la luz del drive B.

- ③ Teclea **FILES** y pulsa Enter.
- ④ Teclea **FILES"B:** y pulsa Enter

Como observastes la instrucción **FILES** nos da un listado de los archivos contenido en la unidad de disco especificada (**FILES"B:**) o en la unidad de disco actual (**FILES**)

- ⑤ Teclea **NEW** y pulsa Enter
- ⑥ Pulsa la tecla F1 y pulsa Enter

Con la instrucción **NEW** el programa realizado es borrado de la memoria, por lo tanto con la instrucción **LIST** no aparece ninguna instrucción.

- ⑦ Pulsa la tecla F3, después del letrero **LOAD** escribe **B:PAG106** y pulsa Enter.

- ⑧ Pulsa la tecla F1 y pulsa Enter. ¿ Qué aparece en la pantalla?

Ejemplo: END

Para salir de BASIC teclea **SYSTEM** y pulsa Enter.

B.- INSTRUCCIONES ELEMENTALES

Antes de iniciar con la programación veremos algunas instrucciones o estatutos que se pueden aplicar en BASIC.

LET.- Se utiliza para asignar un valor a una variable o para realizar operaciones aritméticas dentro del programa.

Formato: LET variable = valor, constante o expresión.

Ejemplos:
 LET A= 3 Asignación de valor.
 LET P = 6 Asignación de valor.
 LET K= A + B Operación aritmética.

Otro formato equivalente que se puede utilizar es omitiendo la instrucción

LET .

Ejemplos:
 A= 3 Asignación de valor.
 P = 6 Asignación de valor.
 K= A + B Operación aritmética.

PRINT.- Permite dar salida a expresiones desde nuestro programa al monitor de la computadora y nos muestra los valores de las variables, constantes y resultados de las operaciones aritméticas.

Formato para mandar imprimir mensajes:

PRINT " Mensaje a exhibir "

Ejemplo:
 \$\$ = "Hola,"
 I\$ = " como estas?"
 PRINT \$\$ + I\$

Formato para mandar imprimir variables con mensajes:

PRINT " Mensaje a exhibir " , variable
PRINT " Mensaje a exhibir " ; variable

PRINT seguido de una coma (,), deja un espacio en la impresión del mensaje y la variable; utilizando el punto y coma (;) imprime inmediatamente dicha variable después del mensaje.

Si los resultados se requieren que sean directamente a la impresora, la instrucción de PRINT deberá cambiarse por LPRINT, respetando su formato.

CLS.- Se utiliza para limpiar la pantalla borrando la información que en ella se encuentra.

Formato: CLS

Ejemplo: CLS

END.- Indica la terminación del programa, ésta será la última instrucción en ejecutarse.

Formato: END

Ejemplo: END

INPUT. Otra manera de asignar valores a las variables es por medio de la instrucción INPUT; ésta acepta datos a través del teclado que se guardan en la variable que va después del mensaje de la instrucción INPUT, y cada vez que el programa sea procesado el valor asignado puede ser distinto.

Formato: INPUT " Mensaje de pantalla " ; variable

INPUT " Mensaje de pantalla " , variable

Ejemplo: INPUT " Dame el tiempo " , A

INPUT " Dame tu nombre " ; AS

El INPUT, seguido de un punto y coma (;), muestra un signo de interrogación (?) después del mensaje para indicar al operador que el programa está esperando a que introduzca los datos. Cuando el formato está dado con coma (,) no aparece el signo de interrogación, sólo espera la información del dato.

C.- PROGRAMACIÓN SECUENCIAL.

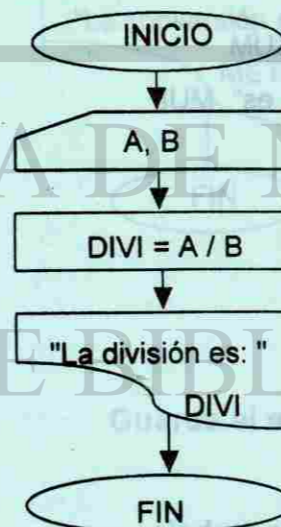
A continuación se presentan ejemplos de programación secuencial en BASIC, en donde se utilizan las instrucciones estudiadas. En cada uno de ellos estará elaborado el algoritmo, el diagrama de flujo y la codificación BASIC. Realiza los pasos necesarios para entrar a BASIC y teclea los siguientes ejemplos.

Ejemplo No. 1.- Programa que divide dos números cualesquiera e imprime su resultado.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Pedir el valor del primer número y guardarlo en una variable.
- 3.- Pedir el valor del segundo número y guardarlo en otra variable.
- 4.- Dividir ambas variables y guardar el valor en una tercera.
- 5.- Imprimir el valor de la tercera variable.
- 6.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Dame el valor del primer número " , A
30 INPUT "Dame el valor del segundo número" , B
40 LET DIVI = A / B
50 PRINT "La división es : " , DIVI
60 END
    
```

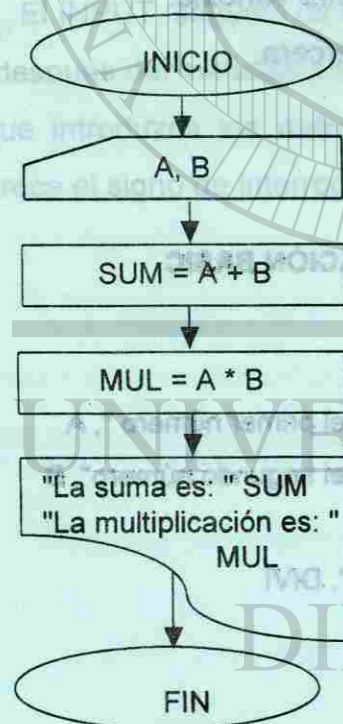
Guarda el ejemplo que teclasteas en tu disco de trabajo

Ejemplo No. 2.- Programa que suma dos números cualesquiera, los multiplica e imprime sus resultados.

ALGORITMO

- 1.- Inicio del Algoritmo.
- 2.- Pedir el valor del primer número y guardarlo en una variable.
- 3.- Pedir el valor del segundo número y guardarlo en otra variable.
- 4.- Sumar ambas variables y guardar su valor en una tercera.
- 5.- Multiplicar ambas variables y guarda su valor en una cuarta.
- 6.- Imprimir el valor de la tercera y cuarta variable.
- 7.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Dame el valor del primer número ", A
30 INPUT "Dame el valor del segundo número", B
40 LET SUM = A + B
50 LET MUL = A * B
60 PRINT "La suma es : ", SUM
70 PRINT "La multiplicación es", MUL
80 END
    
```

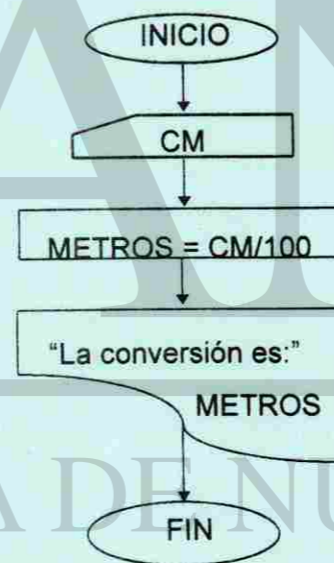
Guarda el ejemplo que tecleaste en tu disco de trabajo

Ejemplo No. 3.- Programa que permite cambiar de centímetros a metros
 METROS = CM /100

ALGORITMO

- 1.- Inicio del Algoritmo.
- 2.- Pedir la cantidad en centímetros y guardarlo en una variable.
- 3.- Calcular la conversión de centímetros a metros.
- 4.- Imprimir el valor del resultado.
- 5.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Dame los centímetros a cambiar: "; CM
30 LET METROS = CM /100
40 PRINT "Los metros son: "; METROS
50 END
    
```

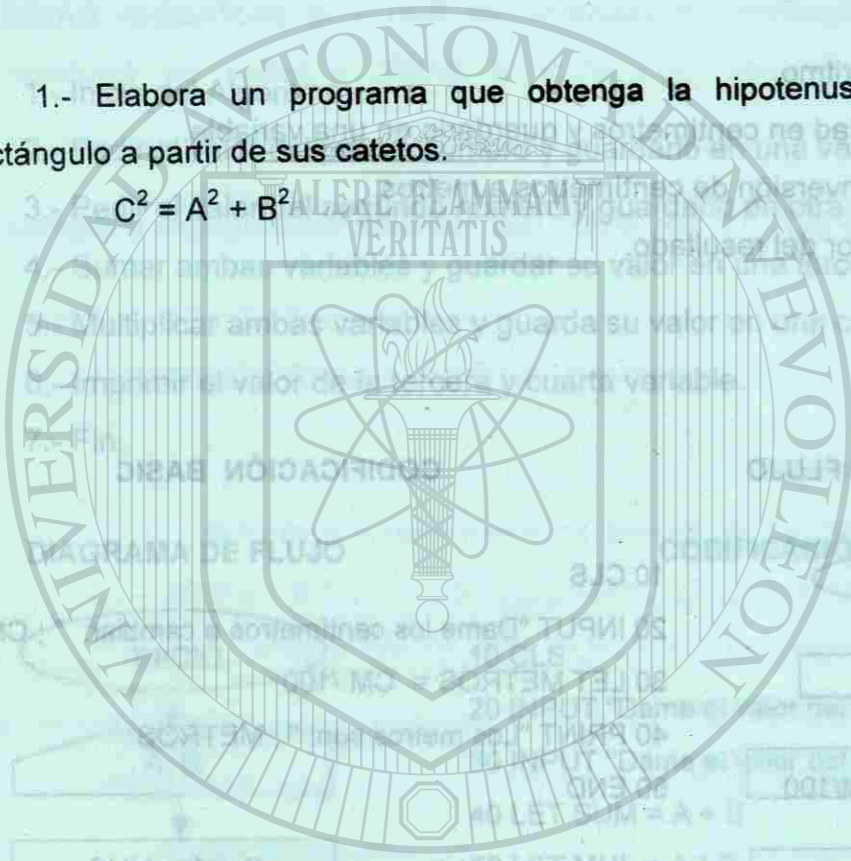
Guarda el ejemplo que tecleaste en tu disco de trabajo

☺ EJERCICIOS

Desarrolla el algoritmo, el diagrama de flujo y la codificación BASIC para los siguientes programas.

1.- Elabora un programa que obtenga la hipotenusa de un triángulo rectángulo a partir de sus catetos.

$$C^2 = A^2 + B^2$$



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

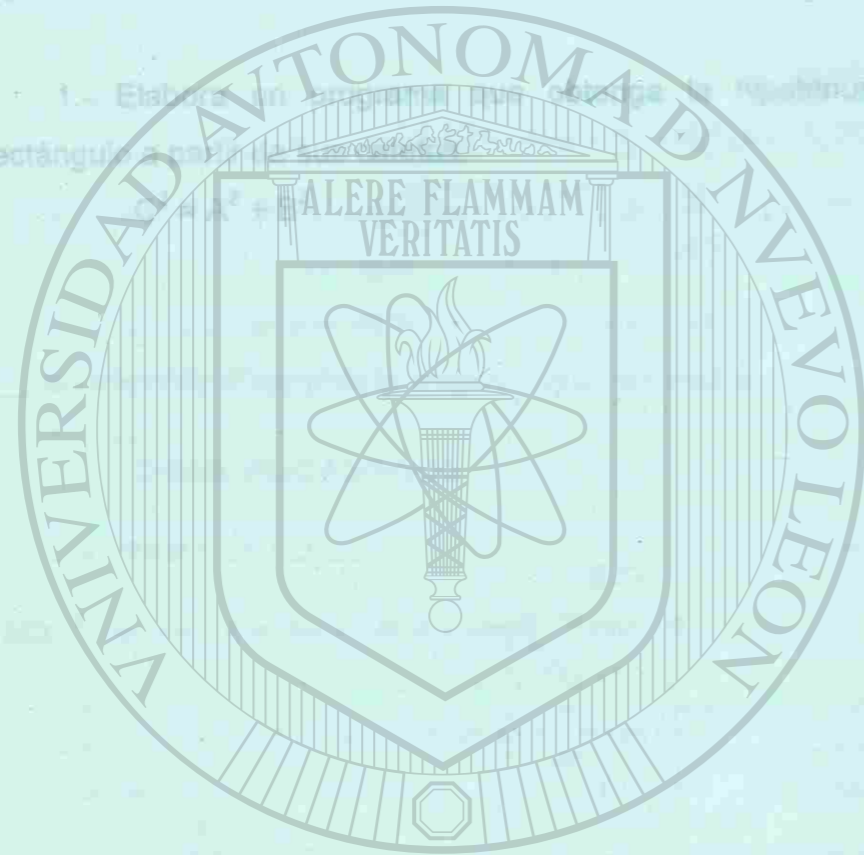


2.- Elabora un programa que permita cambiar de grados centígrados a grados fahrenheit, considerando la siguiente fórmula.

$$F = (C * 9/5) + 32$$

3.- Realiza un programa que permita cambiar de grados Fahrenheit a grados Centígrados.

$$C = (F - 32) * 5/9$$



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

4.- Elabora un programa que permita cambiar de kilogramos a libras.

$$LIBRAS = KG / .454$$

⊙ GOTO.- Interrumpe la ejecución de un programa y lo transfiere a una línea cuyo número está indicado.

Formato: GOTO número de línea

Ejemplo: GOTO 150

Al llegar a leer esta instrucción el computador transfiere al programa a la línea 150.

⚠ IF-THEN-ELSE.- En algunos programas es necesario establecer condiciones o premisas que permitan obtener una verdadera y otra falsa; esta instrucción decide el curso de la ejecución.

Formato: IF condición THEN instrucciones ELSE instrucciones

Ejemplo:

IF B > 100 THEN 20 ELSE 30

Si B es mayor que 100 entonces va a la instrucción 20 y si NO va a 30.

IF B <= 100 THEN 20 ELSE 30

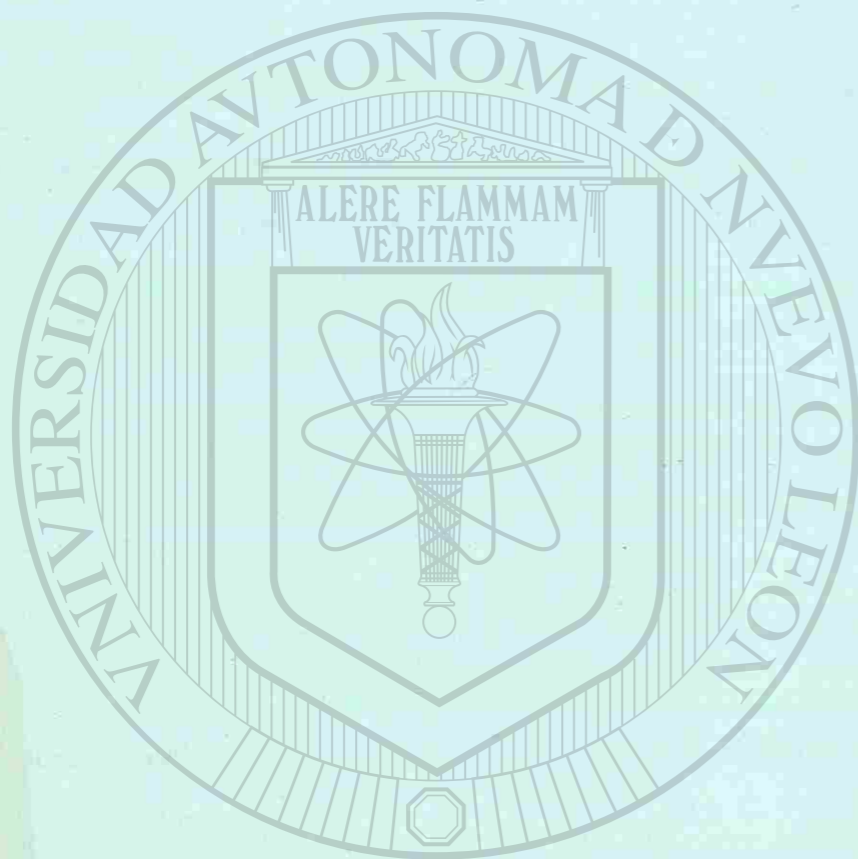
Si B es igual a 100 entonces va a la instrucción 20 y si NO a 30.

IF A > 70 OR A < 30 THEN 1000 ELSE 1200

Si A es mayor que 70 o menor que 30 entonces va a 1000 y si NO a 1200.

5.- Elabora un programa que calcule, con dos números cualesquiera, de las operaciones básicas aritméticas (Suma, Resta, Multiplicación y División).

$$C = (F - 32) \cdot 5/9$$



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

D.- PROGRAMACIÓN CONDICIONAL

A continuación se presentan ejemplos de programación condicional, en cada uno de ellos estará elaborado el algoritmo, el diagrama de flujo y la codificación BASIC. Antes de iniciar con la programación condicional veremos dos instrucciones.

GOTO.- Interrumpe la ejecución secuencial del programa y lo transfiere a una línea cuyo número está indicado después de GOTO.

Formato: GOTO número de línea

Ejemplo: GOTO 150

Al llegar a leer ésta instrucción la computadora transfiere el programa a la línea 150.

△ IF-THEN-ELSE.- En algunos programas es necesario establecer condiciones o premisas que permitan obtener dos salidas, una verdadera y otra falsa; ésta instrucción decide el curso de la información.

Formato: IF condición THEN instrucciones ELSE instrucciones

Ejemplos:

IF S > 100 THEN 60 ELSE 90

SI S es mayor que 100 entonces ve a la instrucción 60 y si NO ve a 90.

IF N\$ = "RAUL" THEN PRINT "MAESTRO" ELSE N\$=B\$

SI N\$ es igual a RAUL entonces imprime MAESTRO y si NO N\$ igual a B\$

IF A > 70 OR A > 80 THEN 1500 ELSE 1200

SI A es mayor que 70 o menor que 80 entonces ve a 1500 y si NO a 1200

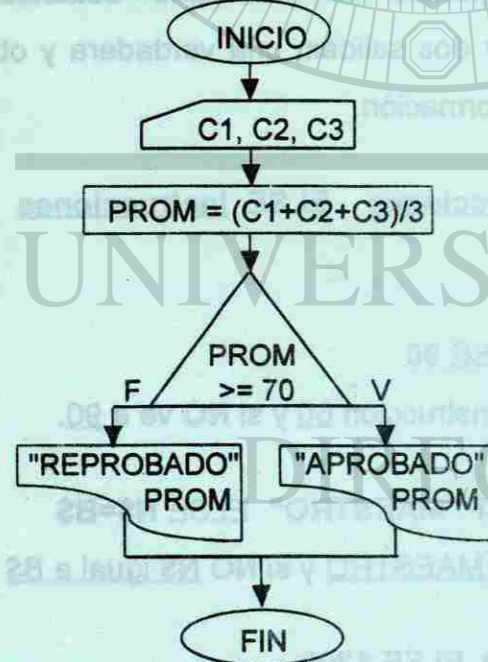
Realiza los pasos necesarios para entrar a BASIC y teclea los siguientes ejemplos.

Ejemplo No. 1.- Programa que calcula el promedio de un alumno, que a partir de 3 calificaciones nos imprime si es aprobado o reprobado y su promedio.
(Alumno aprobado si su promedio es mayor o igual que 70).

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Pedir la calificación del primer parcial.
- 3.- Pedir la calificación del segundo parcial.
- 4.- Pedir la calificación del tercer parcial.
- 5.- Calcular el promedio.
- 6.- **SI** el Promedio es mayor o igual que 70 ir al paso 7 y si **NO** ir al paso 8
- 7.- Imprimir aprobado e ir al paso 9.
- 8.- Imprimir reprobado.
- 8.- Imprimir su promedio.
- 9.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Dame la calificación del 1er. parcial: ",C1
30 INPUT "Dame la calificación del 2do. parcial: ",C2
40 INPUT "Dame la calificación del 3er. parcial: ",C3
50 PROM = (C1 + C2 + C3)/3
60 IF PROM >=70 THEN 70 ELSE 90
70 PRINT "APROBADO"
80 PRINT "Promedio" , PROM : GOTO 110
90 PRINT "REPROBADO"
100 PRINT "Promedio" , PROM
110 END
    
```

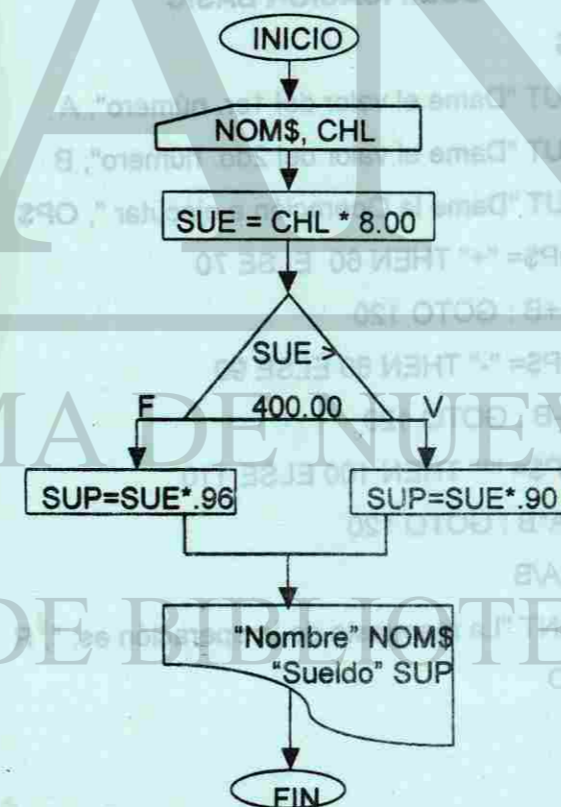
Guarda el ejemplo que tecleastes en tu disco de trabajo

Ejemplo No. 2.- El siguiente programa es el de un trabajador que recibe \$ 8.00 por hora y que a partir del nombre y las horas trabajadas obtiene el sueldo, ya que a éste se le deduce 10% si su sueldo es superior de \$ 400.00 y si el sueldo no es mayor su deducción es de 4%, e imprime el nombre y el sueldo.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Pedir el nombre del empleado.
- 3.- Pedir la cantidad de horas laboradas.
- 4.- Calcular el sueldo.
- 5.- **SI** el sueldo es mayor que \$ 400.00 ir al paso 6 y si **NO** ir al paso 7.
- 6.- Calcular el sueldo con una deducción del 10% e ir al paso 8.
- 7.- Calcular el sueldo con una deducción del 4%.
- 8.- Imprimir el nombre del empleado y el sueldo final.
- 9.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Nombre del empleado: ",NOM$
30 INPUT "Horas laboradas: ",CHL
40 SUE = CHL * 8.00
50 IF SUE > 400 THEN 60 ELSE 70
60 SUP = SUE*.90 : GOTO 80
70 SUP = SUE*.96
80 PRINT "Nombre : ", NOM$
90 PRINT "Sueldo : ", SUP
100 END
    
```

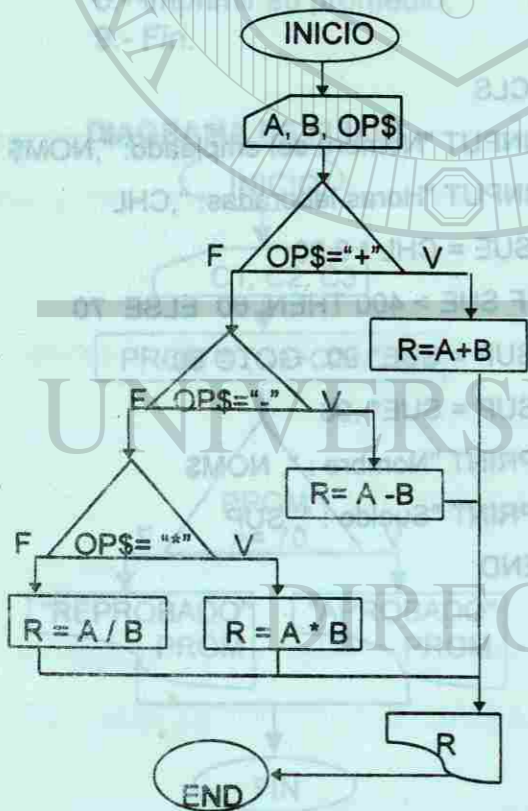
Guarda el ejemplo que tecleastes en tu disco de trabajo

Ejemplo No. 3.- Programa que con dos números cualesquiera puede realizar una de las operaciones básicas aritméticas.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Pedir el valor del 1er. número.
- 3.- Pedir el valor del 2do. número.
- 4.- Pedir la opción a realizar (+, -, *, /).
- 5.- **SI** la operación a ejecutar es igual a "+" ir al paso 6 y si **NO** ir al paso 7
- 6.- Realizar la operación de suma e ir al paso 12
- 7.- **SI** la operación a ejecutar es igual a "-" ir al paso 8 y si **NO** ir al paso 9
- 8.- Realizar la operación de resta e ir al paso 12
- 9.- **SI** la operación a ejecutar es igual a "*" ir al paso 10 y si **NO** ir al paso 11
- 10.- Realizar la operación de multiplicación e ir al paso 12
- 11.- Realizar la operación de división.
- 12.- Imprimir el resultado.
- 13.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 INPUT "Dame el valor del 1er. número", A
30 INPUT "Dame el valor del 2do. número", B
40 INPUT "Dame la Operación a ejecutar ", OP$
50 IF OP$ = "+" THEN 60 ELSE 70
60 R=A+B : GOTO 120
70 IF OP$ = "-" THEN 80 ELSE 90
80 R=A-B : GOTO 120
90 IF OP$ = "*" THEN 100 ELSE 110
100 R=A*B : GOTO 120
110 R= A/B
120 PRINT "La respuesta de la operación es: ", R
130 END
    
```

Guarda el ejemplo que teclasteas en tu disco de trabajo

EJERCICIOS

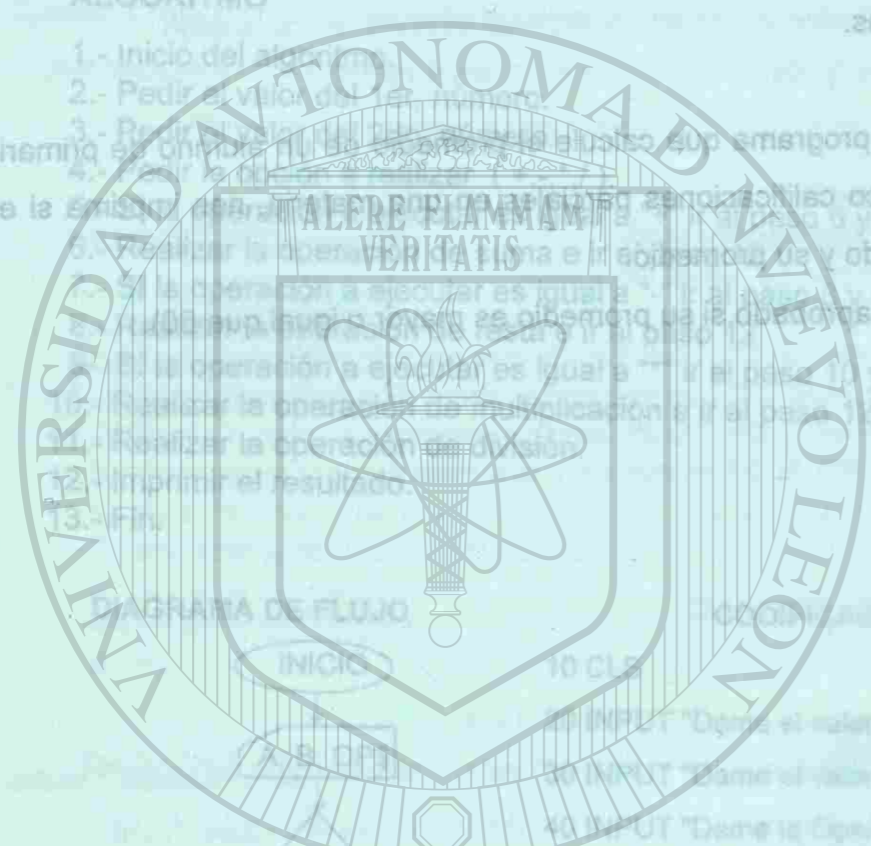
Desarrolla el algoritmo, el diagrama de flujo y la codificación BASIC para los siguientes programas.

- 1.- Realiza un programa que calcule el promedio de un alumno de primaria que cuenta con cinco calificaciones parciales en una materia, nos imprima si es aprobado o reprobado y su promedio.

(Alumno aprobado si su promedio es mayor o igual que 60).

Ejemplo 2.- Elabora un programa para determinar el tipo de raíces de una ecuación cuadrática $ax^2 + bx + c = 0$ e imprima si son Reales o Imaginarias.

$D = b^2 - 4ac$



```
10 INPUT "Dame el coeficiente de x^2, número", A
20 INPUT "Dame el coeficiente de x, número", B
30 INPUT "Dame la constante a ejecutar", C
40 IF D < 0 THEN 60 ELSE 70
50 R = (-B + SQR(D)) / (2*A)
60 R = (-B - SQR(D)) / (2*A)
70 IF OPS = "+" THEN 90 ELSE 90
80 R = A + B
90 IF OPS = "-" THEN 100 ELSE 110
100 R = A - B
110 R = A / B
120 PRINT "El resultado de la operación es", R
130 END
```

Guarda el ejemplo que pedísteles en tu disco de trabajo

3.- Un trabajador recibe \$ 7.00 por una hora normal de trabajo y \$ 12.00 la hora extra. A partir del nombre, las horas normales trabajadas y las horas extras del empleado, calcula el sueldo semanal, ya que se le deduce 10% de su salario si excede de \$ 420.00 y si el sueldo no es superior a esta cantidad su pago es íntegro, es necesario imprimir el nombre y el salario final.

JUANIL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

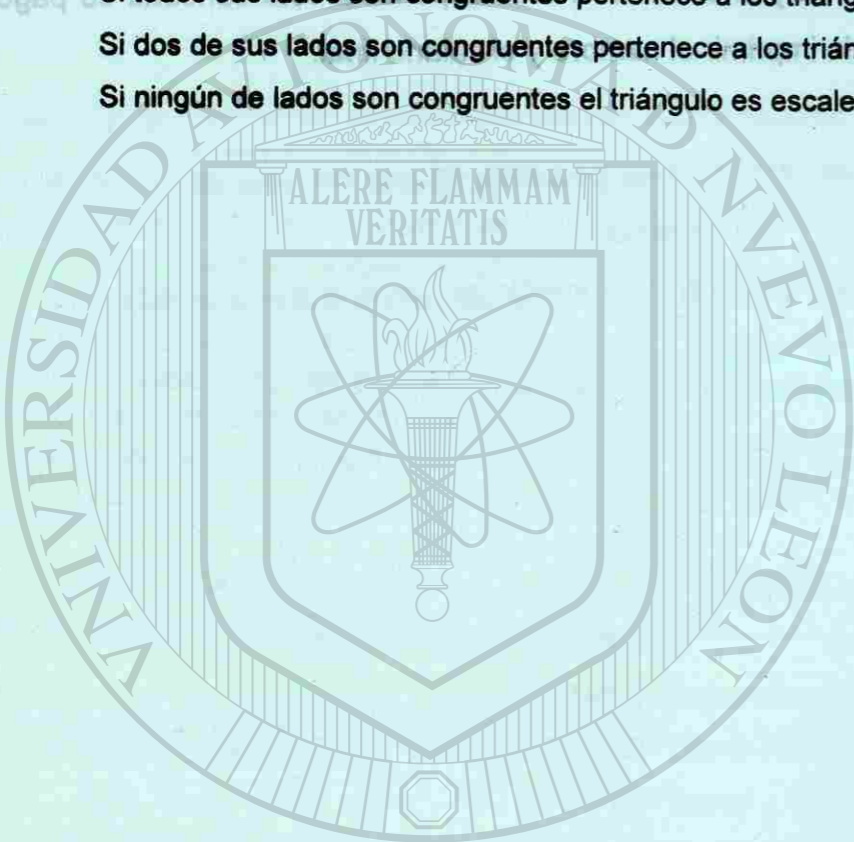


4.- Realiza un programa que permita clasificar a que familia pertenece un triángulo, con respecto a la longitud de sus lados, si por definición se dice que:

Si todos sus lados son congruentes pertenece a los triángulos equiláteros.

Si dos de sus lados son congruentes pertenece a los triángulos isósceles.

Si ningún de lados son congruentes el triángulo es escaleno.

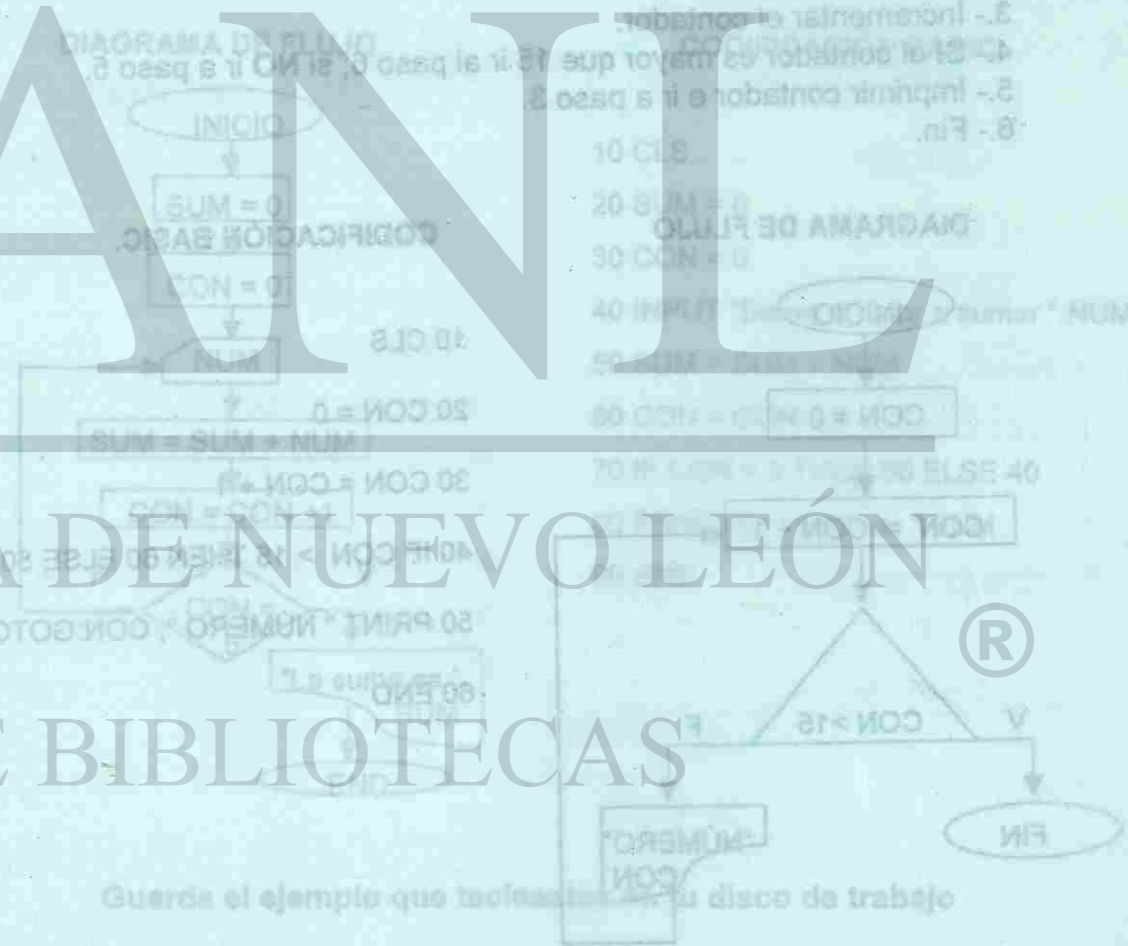


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

5.- Realiza un programa que determine, utilizando el operador MOD, si un número cualquiera es par o non.

Algoritmo
1.- Inicio del algoritmo.
2.- Inicial el contador.
3.- Incrementar el contador.
4.- Si el contador es mayor que 10 ir al paso 8.
5.- Imprimir contador e ir a paso 8.
6.- Fin.



E.- PROGRAMACIÓN CON LAZOS O PROGRAMAS REPETITIVOS.

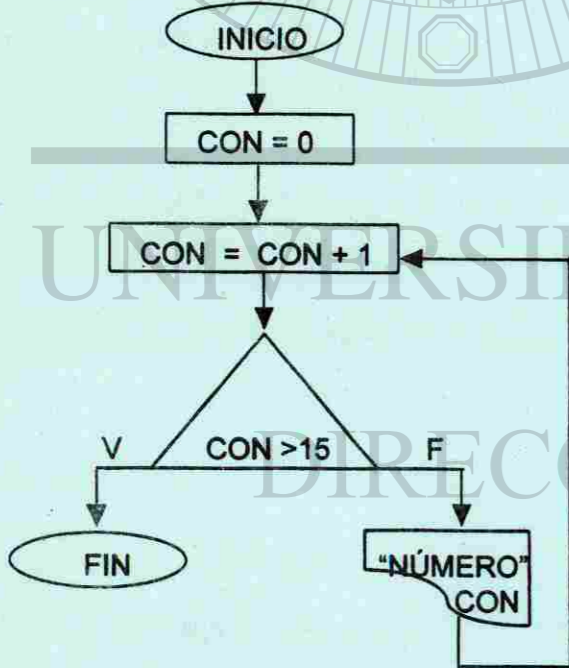
En los siguientes ejemplos de programación con lazos, están elaborados el algoritmo, el diagrama de flujo y la codificación BASIC, verás la aplicación de las instrucciones estudiadas hasta el momento. Realiza los pasos necesarios para entrar a BASIC y teclea los siguientes ejemplos.

Ejemplo No.1.- Programa que imprime los números enteros del 1 al 15.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Iniciar el contador.
- 3.- Incrementar el contador.
- 4.- **SI** el contador es mayor que 15 ir al paso 6; si **NO** ir a paso 5.
- 5.- Imprimir contador e ir a paso 3.
- 6.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC.

```

10 CLS
20 CON = 0
30 CON = CON + 1
40 IF CON > 15 THEN 60 ELSE 50
50 PRINT " NÚMERO ", CON:GOTO 30
60 END
  
```

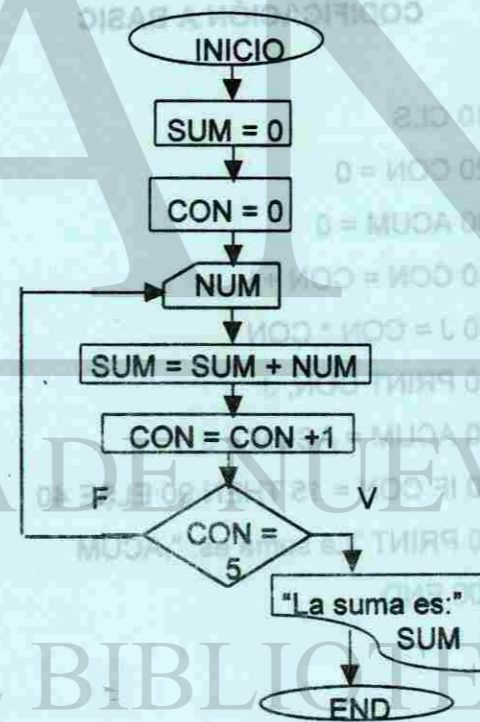
Guarda el ejemplo que tecleastes en tu disco de trabajo

Ejemplo No. 2.- Realiza un programa que sume cinco números cualesquiera e imprima su resultado.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Inicializar el acumulador para la suma de los 5 números.
- 3.- Inicializar el contador.
- 4.- Pedir el número.
- 5.- Incrementar la suma más el número.
- 6.- Incremento del contador.
- 7.- **SI** el contador es igual a 5, ir al paso 8; si **NO** ir al paso 4.
- 8.- Imprimir la suma.
- 9.- Fin

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 SUM = 0
30 CON = 0
40 INPUT "Dame el valor a sumar ", NUM
50 SUM = SUM + NUM
60 CON = CON + 1
70 IF CON = 5 THEN 80 ELSE 40
80 PRINT "La suma es: ", SUM
90 END
  
```

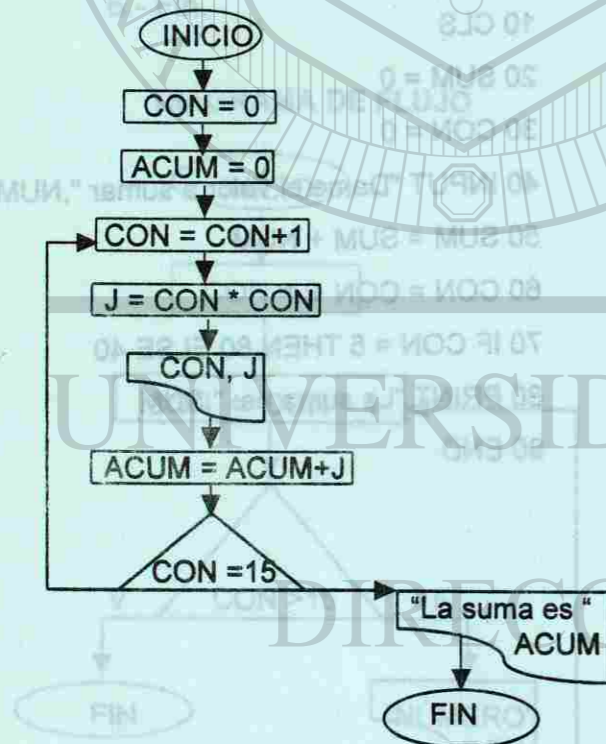
Guarda el ejemplo que tecleastes en tu disco de trabajo

Ejemplo No.3.- Programa que calcula los cuadrados de los números del 1 al 15 e imprime la suma de dichos cuadrados.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Inicialización del contador.
- 3.- Inicialización del acumulador.
- 4.- Incremento del contador.
- 5.- Encontrar el cuadrado del contador.
- 6.- Imprimir el contador y su cuadrado.
- 7.- El acumulador se le incrementa el cuadrado del contador.
- 8.- **SI** el contador es igual a 15, entonces ir a paso 9, si **NO** ir a paso 4
- 9.- Imprimir el acumulador.
- 10.- Fin

DIAGRAMA DE FLUJO



CODIFICACIÓN A BASIC

```

10 CLS
20 CON = 0
30 ACUM = 0
40 CON = CON + 1
50 J = CON * CON
60 PRINT CON, J
70 ACUM = ACUM + J
80 IF CON = 15 THEN 90 ELSE 40
90 PRINT "La suma es: ",ACUM
100 END
  
```

Guarda el ejemplo que tecleaste en tu disco de trabajo

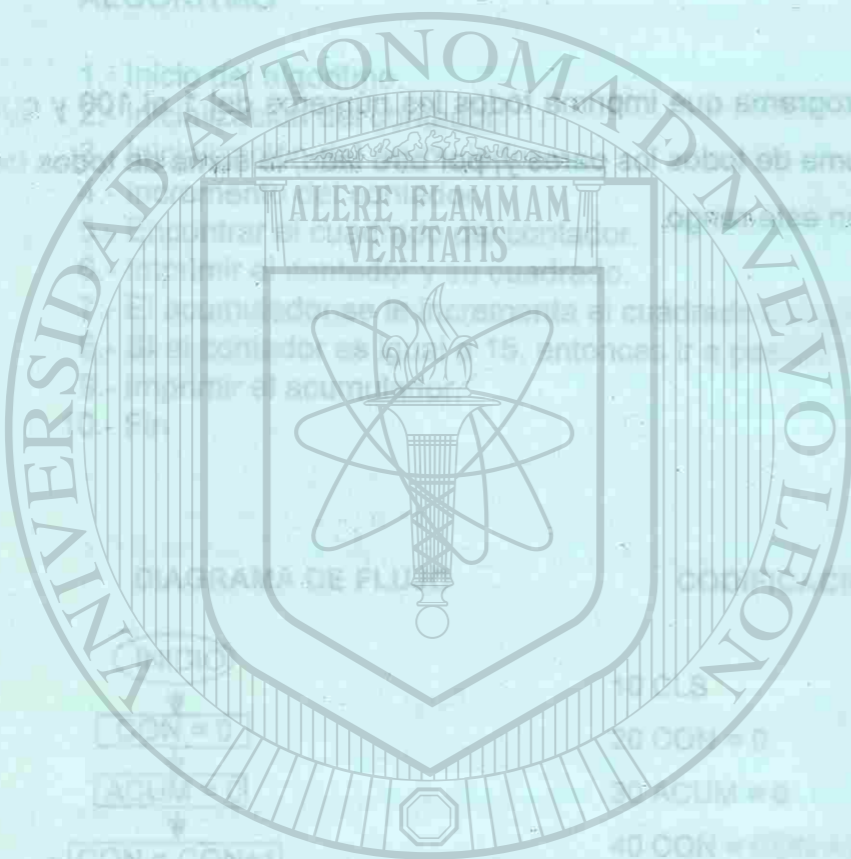
© EJERCICIOS

Desarrolla el algoritmo, el diagrama de flujo y la codificación BASIC para los siguientes programas.

- 1.- Realiza un programa que imprima todos los números del 1 al 100 y que calcule e imprima la suma de todos los pares y, por otro lado, la suma de todos los nones comprendidos en este rango.

2.- Elabora un programa que obtenenga el promedio de temperatura de un día (24 hrs), si ésta se registra cada hora.

ALGORITMO



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

Guarda el ejemplo que hicierdes en tu disco de trabajo

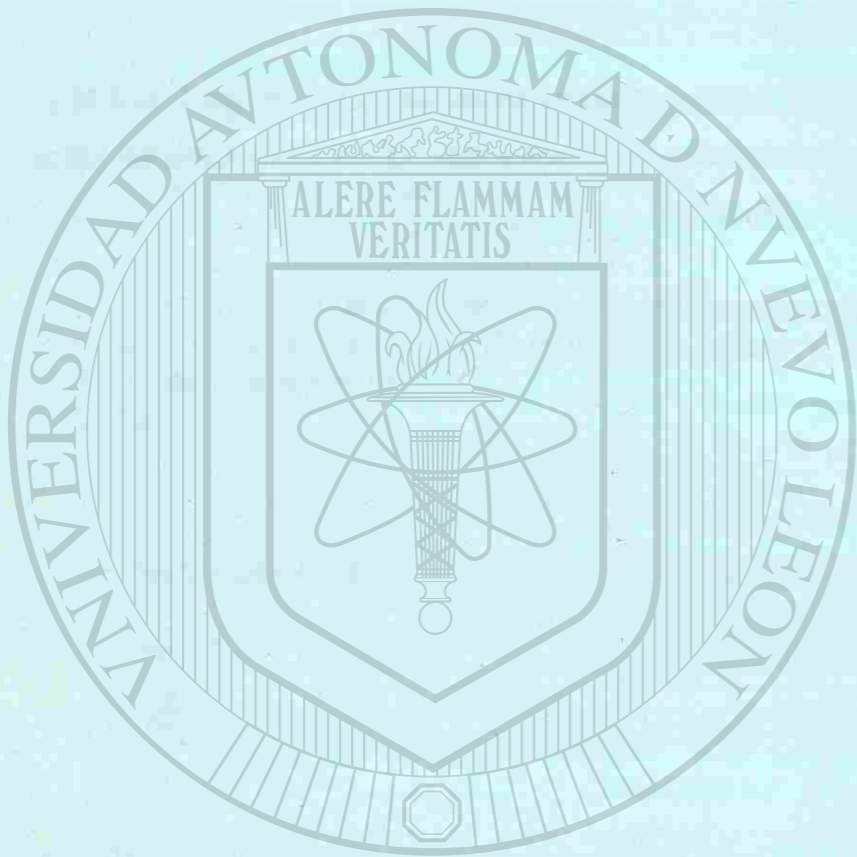
3.- Realiza un programa que calcule una tabla de dos columnas: en la primera se imprimen los números del 1 al 10 y en la segunda, sus cuadrados.

U A N L

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

4.- Elabora un programa que imprima una tabla de multiplicar. Tú le indicas que tabla deseas.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

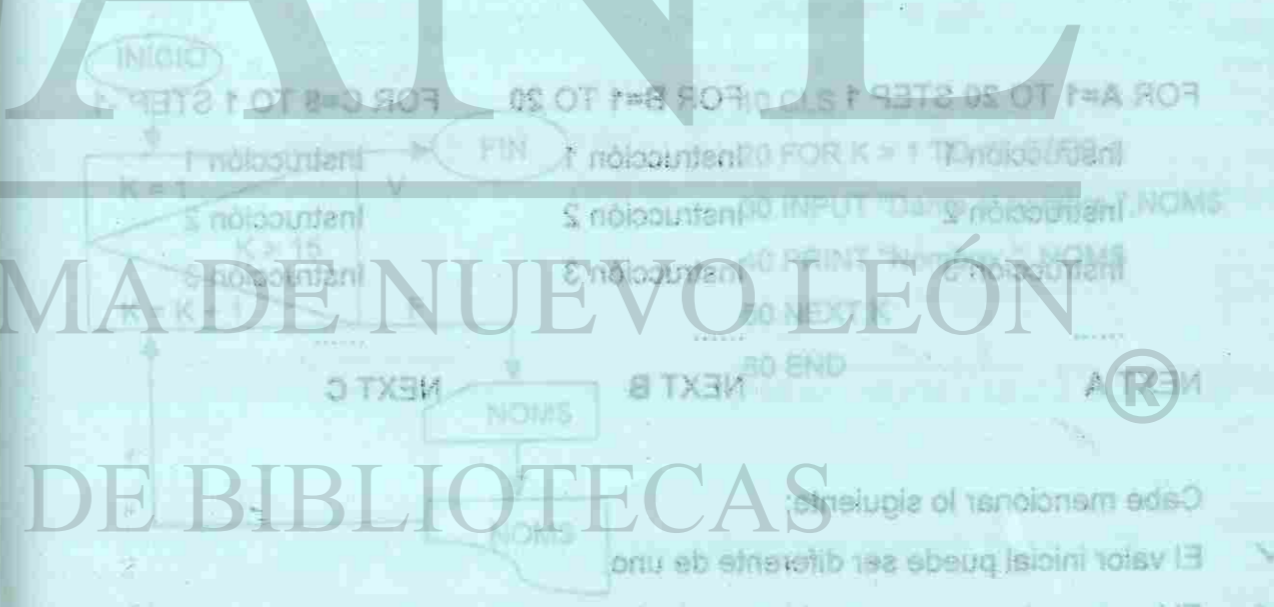
5.- Realiza un programa que imprima y sume los múltiplos de 3, comprendidos entre 1 y 200, además que imprima la suma total de ellos.

El tratamiento de bucles o ciclos de ejecución se utiliza para repetir un conjunto de instrucciones que se ejecutan en programación. Un bucle es el conjunto de instrucciones que se ejecutan repetidas ocasiones. Antes de iniciar con la programación de bucles veremos los siguientes ejemplos.

Ejemplo No. 1.- Programa que permite capturar 15 nombres de estudiantes e imprimirlos.

```
FOR-NEXT - Es una instrucción que se utiliza para ejecutar una o varias instrucciones en forma repetitiva.
Formato: FOR variable = valor inicial TO valor final STEP incremento
1.- Inicio del bucle.
2.- Incrementar el contador en 1.
3.- Si el contador es mayor que 15, se pasa al paso 4.
4.- Pedir el nombre del estudiante y almacenarlo en una variable.
5.- Imprimir el nombre.
6.- Incrementar el contador y pasar al paso 3.
7.- Fin.
```

Diagrama de flujo y codificación básica para el ejemplo No. 1.



Si el incremento es diferente de uno, entonces debemos especificar su valor. El incremento es uno cuando se omite la sentencia STEP. El valor inicial puede ser diferente de uno. Cabe mencionar lo siguiente:

F.- PROGRAMACIÓN CÍCLICA

El tratamiento de bucles, ciclos o lazos es una de las técnicas más utilizadas en programación. Un bucle es el conjunto de instrucciones que se ejecutan en repetidas ocasiones. Antes de iniciar con la **programación cíclica** veremos la siguiente instrucción.

FOR-NEXT. Es una instrucción que se utiliza para ejecutar una o varias instrucciones en forma repetitiva.

Formato: FOR variable = valor inicial TO valor final STEP incremento.

Instrucciones a realizarse dentro del ciclo

NEXT variable

Ejemplos:

FOR A=1 TO 20 STEP 1	FOR B=1 TO 20	FOR C=8 TO 1 STEP -1
Instrucción 1	Instrucción 1	Instrucción 1
Instrucción 2	Instrucción 2	Instrucción 2
Instrucción 3	Instrucción 3	Instrucción 3
.....
NEXT A	NEXT B	NEXT C

Cabe mencionar lo siguiente:

- ✓ El valor inicial puede ser diferente de uno.
- ✓ El incremento es uno cuando se omite la sentencia **STEP**.
- ✓ Si el incremento es diferente de uno, entonces debemos especificar su valor.

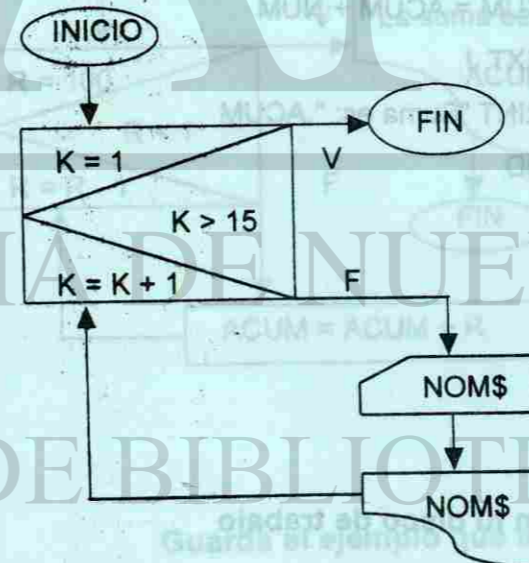
En los siguientes ejemplos de programación cíclica, están elaborados el algoritmo, el diagrama de flujo y la codificación BASIC, verás la aplicación de las instrucciones estudiadas hasta el momento. Realiza los pasos necesarios para entrar a BASIC y teclea los siguientes ejemplos.

Ejemplo No.1.- Programa que permite capturar 15 nombres de personas y los imprime.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Iniciar el contador en 1.
- 3.- **SI** el contador es mayor que 15, ir al paso 7, y si **NO** ir al paso 4
- 4.- Pedir el nombre.
- 5.- Imprimir el nombre.
- 6.- Incrementar el contador, ir al paso 3.
- 7.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN A BASIC

```

10 CLS
20 FOR K = 1 TO 15 STEP 1
30 INPUT "Dame el nombre ", NOM$
40 PRINT "Nombre: ", NOM$
50 NEXT K
60 END
    
```

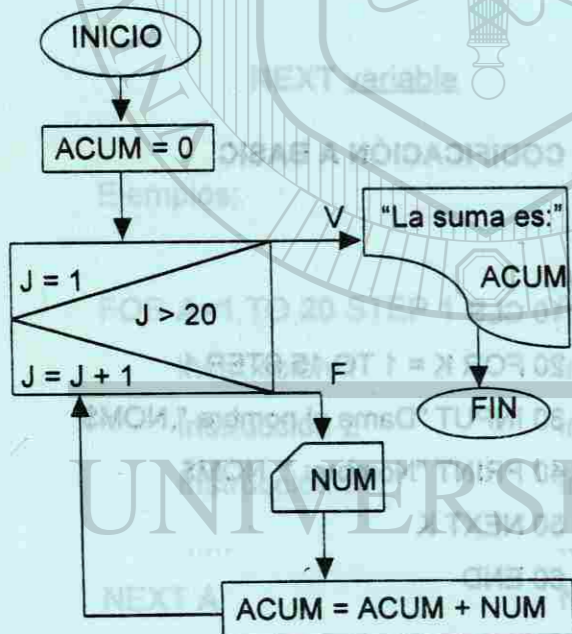
Guarda el ejemplo que tecleaste en tu disco de trabajo

Ejemplo No.2.- Programa para capturar 20 números y obtener la sumatoria de ellos.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Inicializar el Acumulador = 0
- 3.- Inicializar el Contador = 1
- 4.- **SI** contador es mayor que 20, ir al paso 8, si **NO** ir al paso 5.
- 5.- Pedir un número.
- 6.- Sumar el número al acumulador.
- 7.- Incrementar el contador, ir al paso 4.
- 8.- Imprime la suma.
- 9.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 ACUM = 0
30 FOR J = 1 TO 20
40 INPUT " Dame el numero a sumar ", NUM
50 ACUM = ACUM + NUM
60 NEXT J
70 PRINT "Suma es: ", ACUM
80 END
    
```

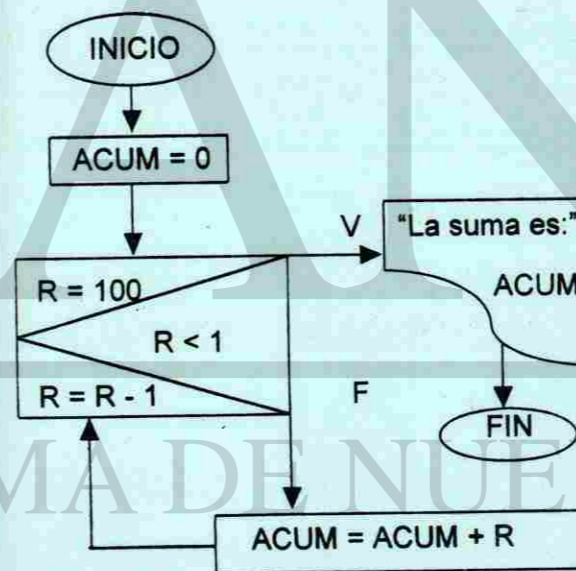
Guarda el ejemplo que tealeastes en tu disco de trabajo

Ejemplo No.3.- Programa que imprime la serie del 100 al 1, calcula e imprime la suma de ellos.

ALGORITMO

- 1.- Inicio del algoritmo.
- 2.- Inicializar el Acumulador en 0
- 3.- Inicializar el Contador en 100
- 4.- **SI** contador es menor que 1, ir al paso 7, si **NO** ir al paso 5.
- 5.- Sumar el contador al acumulador.
- 6.- Decrementar el contador, ir al paso 4.
- 7.- Imprime la suma.
- 8.- Fin.

DIAGRAMA DE FLUJO



CODIFICACIÓN BASIC

```

10 CLS
20 ACUM = 0
30 FOR R = 100 TO 1 STEP -1
40 ACUM = ACUM + R
60 NEXT R
70 PRINT "Suma es: ", ACUM
80 END
    
```

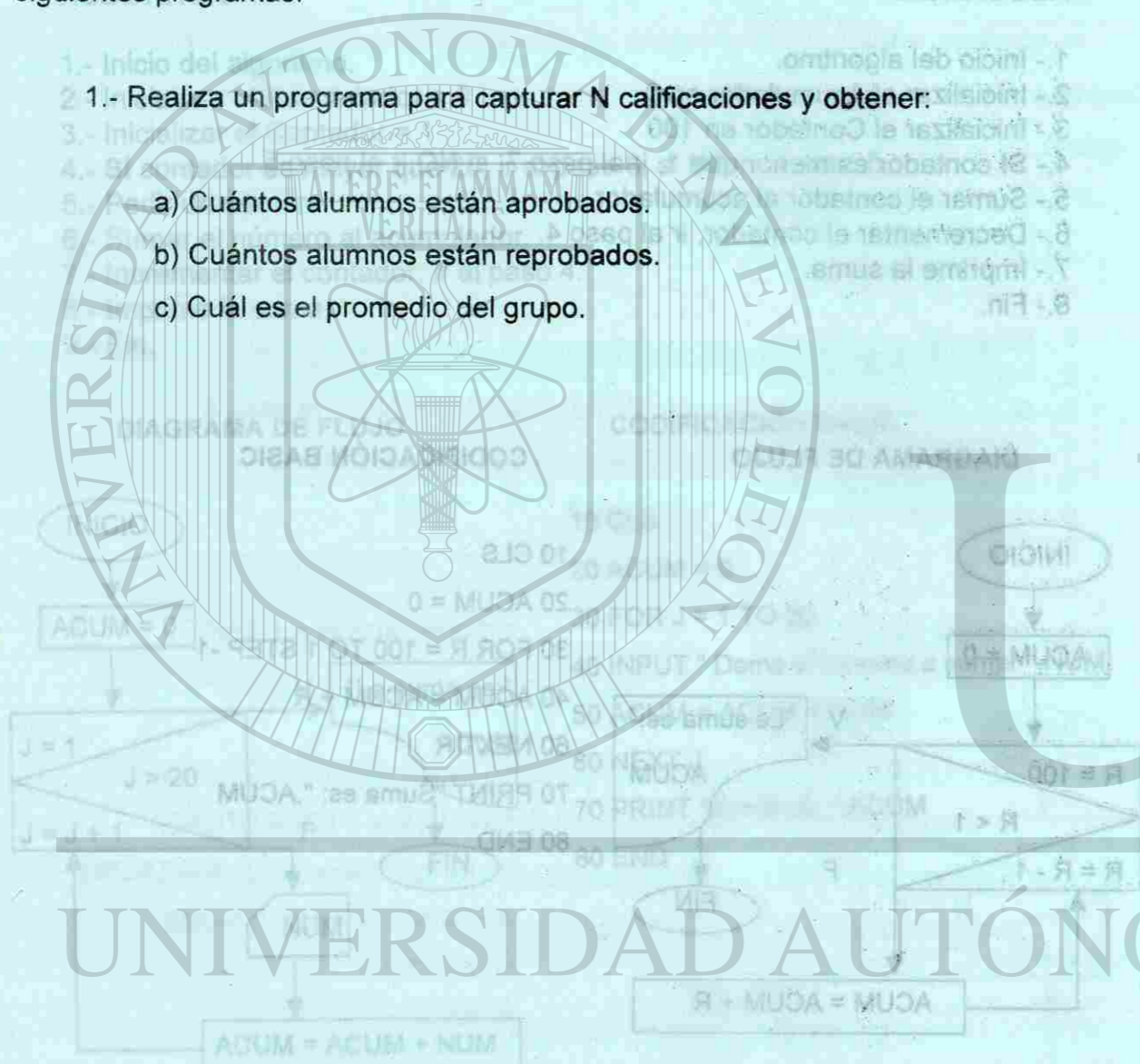
Guarda el ejemplo que tealeastes en tu disco de trabajo

EJERCICIOS

Desarrolla el algoritmo, el diagrama de flujo y la codificación BASIC para los siguientes programas.

1.- Realiza un programa para capturar N calificaciones y obtener:

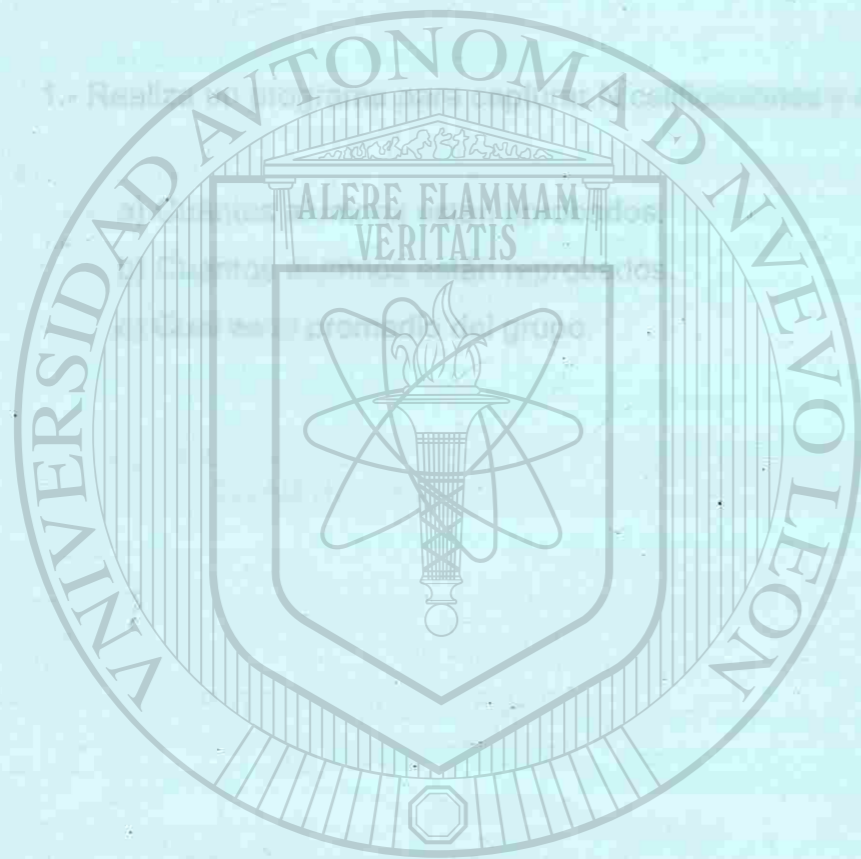
- Cuántos alumnos están aprobados.
- Cuántos alumnos están reprobados.
- Cuál es el promedio del grupo.



2.- Realiza un programa que elabore una tabla de temperaturas en grados centígrados y su equivalente en grados Fahrenheit, desde cero hasta cincuenta grados centígrados.

3.- Realiza un programa que calcule una tabla de dos columnas: En la primera se imprimen los números del 1 al 10 y en la segunda, sus cuadrados.

(Utiliza la instrucción FOR / NEXT)



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

4.- Elabora un programa que imprima una tabla de multiplicar. Tú le indicas que tabla deseas.

(Utiliza la instrucción FOR / NEXT)

1.- Dentro de BASIC realiza los pasos necesarios para observar la lista de archivos o programas de tu disco de trabajo.

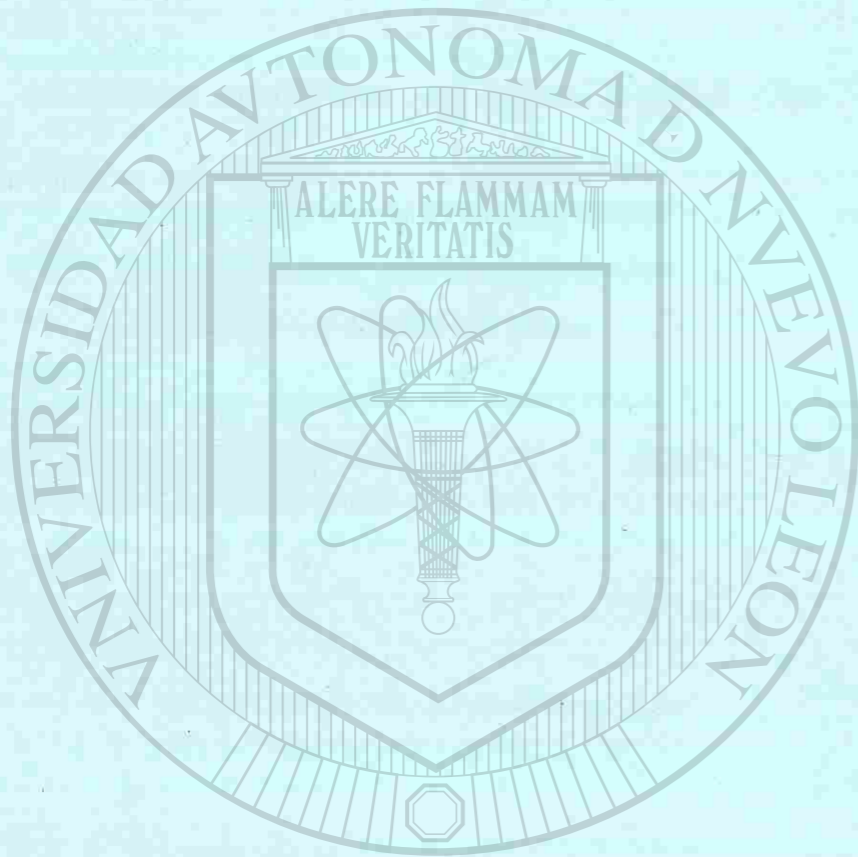
2.- Llévate a cabo los pasos necesarios para abrir de tu disco de trabajo, el ejemplo No. 1 de la programación secuencial y efectúa las modificaciones necesarias para que al final del programa imprima tu nombre, grupo y Preparación donde estás.

3.- Realiza los pasos necesarios para grabar y cerrar el programa del ejercicio anterior, además sale de BASIC.

4.- ¡ FELICIDADES ! has terminado tu curso de Computación.

5.- Realiza un programa que imprima y sume los múltiplos de 3, comprendidos entre 1 y 200, además que imprima la suma total de ellos.

(Utiliza la instrucción FOR / NEXT)



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

☺ EJERCICIO GENERAL:

- 1.- Dentro de BASIC realiza los pasos necesarios para observar la lista de archivos o programas de tu disco de trabajo.
- 2.- Lleva a cabo los pasos necesarios para abrir de tu disco de trabajo, el ejemplo No.1 de la programación secuencial y efectúa las modificaciones necesarias para que al final del programa, imprima tu nombre, grupo y Preparatoria donde estás.
- 3.- Realiza los pasos necesarios para grabar y cerrar el programa del ejercicio anterior, además salte de BASIC.
- 4.- ¡ FELICIDADES ! has terminado tu curso de Computación I.

JOYANES AGUILAR, LUIS. *Programación de la Programación*, Mc. Graw-Hill Segunda edición, México, 1987.

LONG, LARRY. *Programación de la Programación*, Mc. Graw-Hill Hispanoamericana México, 1987.

PARA TERMINAR ...

Has concluido este curso de computación, lo cual te permite acercarte, de manera directa, al mundo de la tecnología moderna.

Te aclaramos que hay mucho más por conocer; lo que has aprendido hasta el momento te servirá de base fundamental para ampliar tus conocimientos en cursos posteriores.

Esperamos haber despertado tu interés en esta rama del saber humano y deseamos que lo asimilado sea de utilidad en tu vida inmediata de estudiante y, más tarde, de profesionista; así como también haber motivado tu inquietud y afán de superación.

ATENTAMENTE

Los autores.

BIBLIOGRAFÍA

ADAMIS, EDDIE. Diccionario BASIC. Mc Graw Hill. México, 1987.

BLACK, ULYSES. Redes de Computadoras. Macrobit Editores, S. A. de C. V. México, 1990.

DUFFY, TIM. Cuatro herramientas de Software. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

DUFFY, TIM. Introducción a la Informática. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

FERSTER, PAUL A. Álgebra I. Addison-Wesley Publishing Company. Segunda edición. U.S.A., 1990.

FRIEDMAN, KOFFMAN. BASIC introducción al lenguaje y resolución de problemas con programación estructurada. Addison-Wesley Iberoamericana. México, 1987.

GOLDSTEIN, LARRY JOEL. I.B.M. PC. Brady Books. Tercera edición. U.S.A., 1986.

HENNEFELD, JULIEN. BASIC con Aplicaciones. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

JOYANES AGUILAR, LUIS. Programación BASIC para microcomputadoras. Mc Graw Hill. Segunda edición. México, 1987.

JOYANES AGUILAR, LUIS. Metodología de la Programación. Mc Graw Hill. Segunda edición. México, 1992.

LONG, LARRY. Programación en BASIC. Prentice Hall Hispanoamericana. México, 1987.

PARA TERMINAR ...

Has concluido este curso de computación, lo cual te permite acercarte, de manera directa, al mundo de la tecnología moderna.

Te aclaramos que hay mucho más por conocer; lo que has aprendido hasta el momento te servirá de base fundamental para ampliar tus conocimientos en cursos posteriores.

Esperamos haber despertado tu interés en esta rama del saber humano y deseamos que lo asimilado sea de utilidad en tu vida inmediata de estudiante y, más tarde, de profesionista; así como también haber motivado tu inquietud y afán de superación.

ATENTAMENTE

Los autores.

BIBLIOGRAFÍA

ADAMIS, EDDIE. Diccionario BASIC. Mc Graw Hill. México, 1987.

BLACK, ULYSES. Redes de Computadoras. Macrobit Editores, S. A. de C. V. México, 1990.

DUFFY, TIM. Cuatro herramientas de Software. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

DUFFY, TIM. Introducción a la Informática. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

FERSTER, PAUL A. Álgebra I. Addison-Wesley Publishing Company. Segunda edición. U.S.A., 1990.

FRIEDMAN, KOFFMAN. BASIC introducción al lenguaje y resolución de problemas con programación estructurada. Addison-Wesley Iberoamericana. México, 1987.

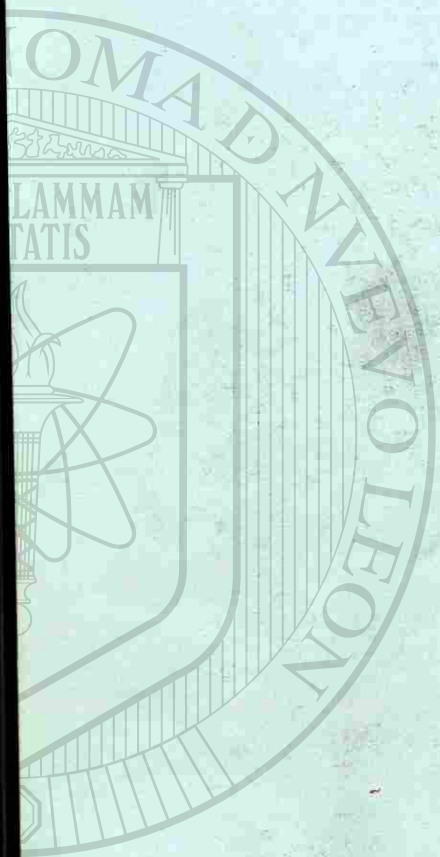
GOLDSTEIN, LARRY JOEL. I.B.M. PC. Brady Books. Tercera edición. U.S.A., 1986.

HENNEFELD, JULIEN. BASIC con Aplicaciones. Grupo Editorial Iberoamérica, S. A. de C. V. México, 1993.

JOYANES AGUILAR, LUIS. Programación BASIC para microcomputadoras. Mc Graw Hill. Segunda edición. México, 1987.

JOYANES AGUILAR, LUIS. Metodología de la Programación. Mc Graw Hill. Segunda edición. México, 1992.

LONG, LARRY. Programación en BASIC. Prentice Hall Hispanoamericana. México, 1987.



U A N

SIDAD AUTÓNOMA DE NUEVO

CCIÓN GENERAL DE BIBLIOTE