

40  
2e)



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

TEXTO PARA EL CURSO  
"INTRODUCCION A LA INFORMATICA"  
EN EL BACHILLERATO

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I A

P R E S E N T A

MARTHA LETICIA GODINEZ GALICIA



DIRECTORA DE ESTUDIOS ESCOLARES ELISA VISO GUROVICH



MEXICO, D. F.

FACULTAD DE CIENCIAS  
SECCION ESCOLAR

1997

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

M. en C. Virginia Abrin Batule  
Jefe de la División de Estudios Profesionales de la  
Facultad de Ciencias  
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:

TEXTO PARA EL CURSO "INTRODUCCION A LA INFORMATICA"  
EN EL BACHILLERATO  
realizado por MARTHA LETICIA GODINEZ GALICIA  
con número de cuenta 8016407-8 , pasante de la carrera de ACTUARIO  
Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis  
Propietario

M. en C. ELISA VISO GUROVICH

Propietario

DR. JAVIER PAEZ CARDENAS

Propietario

M. en I. MA. DE LUZ GASCA SOTO

Suplente

M. en C. VIRGINIA ABRIN BATULE

Suplente

MAT. CESAR GUEVARA BRAVO

Consejo Departamental de Matemáticas

ACT. AGUSTIN ROMAN AGUILAR

Al recuerdo y memoria de mi abuelita Rosita.

A mis padres, mi eterno agradecimiento por todo el apoyo que me han dado.

A mis niños:

Rosita, gracias por tu apoyo y ejemplo.

Luisito, gracias por tu alegría.

A mi hermano, por los buenos tiempos.

A mi madrina Celsita por su ayuda para la impresión de esta tesis.

A la Universidad Nacional Autónoma de México, por la oportunidad y orgullo de ser parte de ella.

A mi directora de tesis, M. en C. Elisa Viso Gurovich, por toda la paciencia que me tuvo.

A todos mis "cuates" de Prepa 8.

## Objetivo

El procesamiento y obtención de información a través del uso de computadoras a tenido un gran auge en los últimos 30 años y, debido a su efectividad, rapidez y desarrollo no se avizora en un futuro cercano sea creada una máquina que las reemplace.

Debido a esta situación, en la Escuela Nacional Preparatoria de la Universidad Nacional Autónoma de México y como parte de la Restructuración del Plan de Estudios, para el cuarto año de bachillerato se implementó desde 1990, la materia piloto de Introducción a la Informática. Dicha materia tiene como antecedente la ya aprobada Opción Técnica de Computación la cual es impartida dentro de los planteles de la ENP desde 1985. Con esto se busca iniciar y preparar al futuro bachiller en el conocimiento y uso de la computadora, tanto en su aspecto técnico como práctico. Este primer acercamiento, - en muchos casos - del joven con la computadora debe ser lo más convincente y motivante posible y evitando que más adelante exista una aversión o el rechazo total a ésta, también se busca que el alumno tenga una preparación acorde con la situación actual.

Dado que la mayoría de la bibliografía, manuales de programa de uso de las computadoras se encuentra escrito en el idioma inglés, a niveles especializados y presentándose en ellos un gran número de tecnicismos propios del tema, el objetivo del presente trabajo es crear un material de apoyo accesible al alumno que cursa en el bachillerato la materia de Introducción a la Informática considerando todos los temas propuestos en el programa.

## INDICE

### 0. INTRODUCCIÓN.

### I. ANTECEDENTES DE LA INFORMÁTICA.

- 1.1. Evolución histórica de las diversas formas de procesamiento de la información.
  - 1.1.1. Primeras culturas. Formas de conteo.
  - 1.1.2. Primeras máquinas y herramientas de conteo.
  - 1.1.3. Primeros calculadores electrónicos.  
Características.
  - 1.1.4. Tipos de computadoras.
  - 1.1.5. Generaciones de computadoras.
- 1.2. Terminología básica usada en informática y computación.
- 1.3. Implicaciones sociales de las computadoras
- 1.4. Aplicaciones
  - 1.4.1. Aplicaciones militares.
  - 1.4.2. Aplicaciones científicas.
  - 1.4.3. Aplicaciones administrativas.
  - 1.4.4. Aplicaciones industriales.
  - 1.4.5. Aplicaciones en la investigación.
  - 1.4.6. Aplicaciones educativas.
  - 1.4.7. Aplicaciones a la medicina.
  - 1.4.8. Vida diaria.

### II. ESTRUCTURA FÍSICA DE UNA COMPUTADORA.

- 2.1. Hardware.
  - 2.1.1. Definición.
  - 2.1.2. Equipo Central de una computadora.
  - 2.1.3. Unidad de entrada y/o salida.
  - 2.1.4. Unidad de memoria principal.
  - 2.1.5. Unidad central de proceso.
  - 2.1.6. Equipo Periférico.
- 2.2. Teclado.
  - 2.2.1. Teclas de funciones.

### III. PROCESAMIENTO DE TEXTOS.

- 3.1. Introducción.
- 3.2. Programa de aplicación procesador de texto.
- 3.3. Comandos básicos para el manejo del procesador de textos.
  - 3.3.1. Entrada a WORDSTAR.
  - 3.3.2. Menú de apertura.
  - 3.3.3. Cambio de unidad de disco (drive) y/o subdirectorio.
  - 3.3.4. Apertura o edición de un documento.
  - 3.3.5. Descripción de la pantalla de edición.
  - 3.3.6. Movimientos del cursor.
  - 3.3.7. Menú de ayuda.
  - 3.3.8. Comandos de edición.
  - 3.3.9. Guardar ó abandonar un texto.
  - 3.3.10. Modificar un documento.
  - 3.3.11. Modificación de la pantalla de edición.
  - 3.3.12. Manejo de bloques en un texto.
  - 3.3.13. Búsquedas y reemplazo de texto.
  - 3.3.14. Colocación de acentos y caracteres especiales.
  - 3.3.15. Impresión de un documento.
  - 3.3.16. Comandos de punto.

### IV. ESTRUCTURA LÓGICA DE UNA COMPUTADORA

- 4.1. Definición
- 4.2. Sistemas numéricos y representación interna de la información.
  - 4.2.1. Sistemas numéricos posicionales.
  - 4.2.2. Aritmética base 2.
- 4.3. Códigos de comunicación y almacenamiento.
  - 4.3.1. Lenguajes de programación.
    - 4.3.1.1. Lenguaje de máquina.
    - 4.3.1.2. Lenguaje ensamblador.
    - 4.3.1.3. Lenguaje de Alto Nivel.
  - 4.3.2. Modalidades de Traducción.
  - 4.3.3. Funcionamiento general de la computadora.
- 4.4. Sistemas Operativos.
  - 4.4.1. Sistema operativo MS-DOS.
    - 4.4.1.1. Comandos para el manejo y mantenimiento de discos.



- 4.4.1.2. Comandos para el manejo de directorios.
- 4.4.1.3. Comandos para el manejo de archivos.
- 4.4.1.4. Comandos de configuración y operatividad.

## V. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS Y PROGRAMACIÓN.

5.1. Método de solución de problemas.

5.2. Fundamentos de programación en un lenguaje estructurado.

- 5.2.1. Antecedentes.
- 5.2.2. Elementos del lenguaje PASCAL.
  - 5.2.2.1. Identificador.
  - 5.2.2.2. Separador de instrucciones.
  - 5.2.2.3. Comentarios.
  - 5.2.2.4. Tipos simples.
- 5.2.3. Partes de un programa en PASCAL.
  - 5.2.3.1. Encabezado.
  - 5.2.3.2. Sección de declaraciones.
    - 5.2.3.2.1. Etiquetas.
    - 5.2.3.2.2. Constantes.
    - 5.2.3.2.3. Tipos.
    - 5.2.3.2.4. Variables.
    - 5.2.3.2.5. Procedimientos.
    - 5.2.3.2.6. Funciones.
    - 5.2.3.2.7. Criterios para el uso de funciones y procedimientos.
  - 5.2.3.3. Cuerpo del programa.
- 5.2.4. Programación modular.
  - 5.2.4.1. Metodología de programación.
  - 5.2.4.2. Criterios de diseño.
  - 5.2.4.3. Rutinas.
  - 5.2.4.4. Funciones.
- 5.2.5. Operadores y funciones estándar.
  - 5.2.5.1. Operadores.
    - 5.2.5.1.1. Operador de asignación.
    - 5.2.5.1.2. Operadores aritméticos.
    - 5.2.5.1.3. Operadores de comparación.
    - 5.2.5.1.4. Operadores booleanos.
  - 5.2.5.2. Funciones estándar.
  - 5.2.5.3. Precedencia.
- 5.2.6. Rutinas de entrada y salida.
  - 5.2.6.1. Rutinas de entrada.
    - 5.2.6.1.1. Rutina READ.
    - 5.2.6.1.2. Rutina READLN.
  - 5.2.6.2. Rutinas de salida.

- 5.2.6.2.1. Rutina WRITE.
- 5.2.6.2.2. Rutina WRITELN.
- 5.2.6.3. Formato de impresión.
- 5.2.6.4. Consideraciones importantes.

- 5.2.7. Bases de programación estructurada.
  - 5.2.7.1. Estructuras lógicas de programación.
    - 5.2.7.1.1. Secuencia o bloque.
    - 5.2.7.1.2. Estructuras de selección.
    - 5.2.7.1.3. Estructuras de iteración.
  - 5.2.7.2. Tipos estructurados.
    - 5.2.7.2.1. Arreglos.
    - 5.2.7.2.2. Registros.

### 5.3. Virus computacionales

## VI. SOFTWARE DE APLICACIÓN Y SERVICIOS DE RED

- 6.1. Ambientes gráficos. Características.
- 6.2. Hojas de cálculo. Características.
- 6.3. Manejadores de bases de datos. Características.
- 6.4. Editores gráficos. Características.
- 6.5. Servicios de red. Características.

APÉNDICE. PROGRAMA DE LA MATERIA "INTRODUCCIÓN A LA INFORMÁTICA".

## Introducción

Las dificultades económicas, causa que generalmente impide que los jóvenes no puedan continuar una carrera a nivel licenciatura, originó que la Universidad Nacional Autónoma de México consciente de su papel dentro del sistema educativo y su influencia en la formación de recursos humanos implementara dentro de su plan de estudios a nivel medio superior a partir del ciclo escolar 1985-1986, las llamadas opciones técnicas con la finalidad de que los jóvenes que se encuentren en esta situación puedan hacer frente a sus necesidades económicas, pudiendo obtener así una fuente de ingresos.

Debido al gran interés que la comunidad preparatoriana mostró, y a la oportunidad que ello representa en la transmisión de conocimientos que les pueden ser útiles a corto plazo, dicho entusiasmo debe ser canalizado de la mejor manera posible haciendo que el alumno no pierda el interés por la Opción Técnica elegida y haciendo que esta transmisión de conocimientos sea lo más provechosa y duradera posible.

En 1985, se conformó una Comisión del Consejo de Planeación de la UNAM, la cual elaboró diversos estudios relativos a las necesidades de recursos humanos calificados, a la situación de la educación media superior a nivel nacional y en particular en la UNAM, así como la situación que guarda la educación técnica en México y en otros países.

Esta comisión recomendó, entre otras cosas que las opciones técnicas que ofrezca la Escuela Nacional Preparatoria debieran tener un carácter flexible y dinámico en función a la satisfacción de necesidades sociales de recursos humanos calificados, lo que determinará tanto su vigencia como la implementación de nuevas opciones.

Las opciones técnicas deben ser vistas como parte reforzadora de la formación académica obtenida durante el bachillerato y no como parte independiente a él, ya que una complementa a la otra.

Están consideradas dentro del bachillerato, ya que el trabajo académico dentro del mismo da la oportunidad de alternar ambos estudios. Estos le serán útiles al alumno en caso de no poder continuar sus estudios a nivel licenciatura, o ya sea que por obligaciones laborales y la carga académica de la licenciatura se vea orillado a abandonar sus estudios.

La Escuela Nacional Preparatoria ofreció de manera inicial las opciones técnicas en:

- Técnico en Computación.
- Técnico Auxiliar en Contabilidad.
- Técnico en Histopatología.

actualmente se imparten además las opciones de:

- Técnico en Fotografía.
- Técnico Restaurador.
- Técnico en Dibujo arquitectónico.

A partir de febrero de 1990, se implementó el plan piloto de la materia de Introducción a la Informática en el cuarto año del bachillerato y aprobada por el Consejo Técnico en septiembre de 1994.

El presente trabajo es una propuesta de texto dirigida a los estudiantes del cuarto año del bachillerato que cursan la materia de Introducción a la Informática ya que está elaborado de acuerdo al programa de la asignatura. Pensando también en las personas que requieran de un texto accesible para su iniciación en el campo de la informática.

## I. ANTECEDENTES DE LA INFORMÁTICA

## L. ANTECEDENTES DE LA INFORMATICA

El propósito de la presente unidad es que el alumno conozca la forma en que han evolucionado las herramientas utilizadas por el hombre en la tarea de procesar información.

### 1.1. Evolución histórica de las diversas formas de procesamiento de la información.

En su origen el hombre necesitó registrar y llevar un conteo y control de sus propiedades, con el paso del tiempo su entorno, su cúmulo de conocimientos, las tareas que ha adquirido han aumentado en forma considerable y a requerido inventar o diseñar máquinas que faciliten o le auxilien en su trabajo.

#### 1.1.1. Primeras culturas. Formas de conteo.

Primeras Culturas	-	Representaciones y sistemas numéricos. Surge la aritmética como necesidad de contar y agrupar elementos.
Caldeos, babilonios y egipcios	-	Utilizaron cotidianamente números, cuentas y presentaciones aritméticas.
Mayas	-	Representación y utilización del cero logrando la construcción de un sistema numérico poderoso.
Arabes	-	Introducción del sistema decimal. Manejo del álgebra, desarrollándose posteriormente la trigonometría y el cálculo diferencial e integral.

#### 1.1.2. Primeras máquinas y herramientas de conteo.

Primeros Instrumentos	-	Ábaco chino (primera máquina auxiliar de cálculo ó calculadora), tablas con resultados de operaciones más utilizadas. Aunque las primeras máquinas que se utilizaron no tenían las características de una computadora actual, su finalidad era la misma. El ábaco representa la primera
-----------------------	---	---

- calculadora mecánica, que consistía en una tablilla dividida en varias columnas, la columna que se localiza más a la derecha representa a las unidades, la anterior a las decenas y así sucesivamente. El ábaco utiliza un sistema posicional decimal.
- 1642 Blaise Pascal, matemático y filósofo francés; construye una máquina capaz de llevar la cuenta automática de sumas y restas elementales cuyo funcionamiento es por medio de engranes. Manejo del Principio de Acarreo.
- 1671 Leibnitz, G.W. (alemán). Intentó la construcción de una máquina para realizar multiplicaciones y divisiones mediante sumas y restas sucesivas, usando el Principio de Acarreo de Pascal, pero no se pudieron construir las piezas requeridas.
- 1804 Joseph-Marie Jacquard (francés). Automatizó las fases del funcionamiento de las máquinas de tejido textil, intentado un siglo antes por Falcón; las telas mostrarían un dibujo correspondiente a un patrón registrado en una tarjeta perforada que la máquina podía identificar.
- 1822 Charles Babbage, matemático inglés. Ideó una máquina de diferencias capaz de realizar automáticamente operaciones aritméticas en secuencias diferentes. La máquina estaba dividida funcionalmente en dos partes: una que mandaba órdenes y otra que ejecutaba esas ordenes. La que ejecutaba las ordenes era una versión muy ampliada de la máquina de Pascal, mientras que la otra era la parte clave. La innovación consistía en que el usuario podía, cambiando las especificaciones del control, lograr que la misma máquina ejecutara operaciones complejas, diferentes de las que había hecho antes.
- 1887 Herman Hollerith (E.U.). Censo de 1880, representación de datos de la encuesta en tarjetas perforadas que podían ser contadas (procesadas) por una máquina. El uso de ésta máquina de tarjetas perforadas permitió contabilizar y

organizar rápida y económicamente grandes cantidades de datos.

1890 - 1940

Las máquinas de tarjetas perforadas (contables, de registro unitario) fueron desarrolladas y dotadas de otras funciones:

- Recepción y reproducción de datos;
- Clasificación y subdivisión de datos;
- Sumas, restas, multiplicaciones, divisiones;
- Comparación de datos;
- Realizar búsquedas;
- Producir reportes;
- Perforar tarjetas con resultados de operaciones;
- Aceptar y emitir datos numéricos y alfabéticos;

### 1.1.3. Primeras calculadoras electrónicas: Características.

1936

Conrad Zuse (alemán). Construyó una calculadora electromecánica (Z1), introduciendo los siguientes avances:

- Representación binaria de los números
- Dejó de ser la máquina cien por ciento mecánica
- Uso de relevadores o dispositivos electromecánicos

1944

Howard H. Aiken, U. Harvard. Desarrolló una calculadora automática MARK I, usando las ideas de Babbage y la tarjeta perforada de Jacquard:

- Lecturas de instrucciones por cintas de papel perforadas.
- Emisión de resultados de los cálculos, perforándolos en tarjetas o imprimiéndolos por medio de máquinas de escribir adaptadas.
- Conjunto de calculadoras conectadas entre sí.
- Sumar dos números de 23 dígitos en 3 décimas de segundo.
- Multiplicar dos números de 23 dígitos en aproximadamente 6 segundos.

1945-1946

J.P. Eckert, J.W. Mauchly y H.H. Goldstine (E.U.). Primera calculadora electrónica: ENIAC, Electronic Numerical Integrator and Calculator, contaba con



dieciocho mil bulbos, medio millón de piezas diferentes, doscientos metros cuadrados de superficie. Construida en la Universidad de Pensilvania. Finalidad: Cálculos relativos a las trayectorias balísticas y posteriormente trabajos científicos relacionados con la física nuclear. Manejo: modificación manual de interruptores y conexiones de cables eléctricos, empleando mucho personal y horas por el cambios de problema. Aportaciones: se eliminaron las partes mecánicas por bulbos activados por impulsos eléctricos, permitiendo mayor rapidez.

#### 1.1.4. Tipos de computadoras.

Se caracterizan por su tamaño, precio, velocidad de operación, capacidades de memoria y procesamiento y pueden dividirse en :

- **Microcomputadoras.** Son las computadoras más pequeñas. Algunos de sus modelos rivalizan con las minicomputadoras y mainframes. Pueden funcionar como unidades independientes o conectadas otras microcomputadoras o mainframes para ampliar sus capacidades. Tiene a su disponibilidad programas a bajo costo y de fácil operación.
- **Minicomputadoras.** Desarrolladas en la década de los 60's para realizar tareas como el manejo de comunicación de datos. Actualmente se utilizan en el procesamiento de textos e información y en aplicaciones de multiusuarios. Son pequeñas, menos costosas y con mayor facilidad de instalación y mantenimiento que las mainframes.
- **Mainframes.** Sistemas grandes y rápidos, capaces de controlar dispositivos de entrada y salida por varios cientos. Poseen inmensas capacidades de procesamiento. Usadas como depósitos de grandes bancos de datos que pueden ser accedidos directamente por los usuarios de mainframes o conectados desde una microcomputadora a una mainframe.
- **Maxicomputadoras.** Requieren de sistemas especiales de regulación de temperatura y humedad, y de un considerable soporte técnico-humano para su mantenimiento.

- Supercomputadoras. Son las más rápidas pero las más costosas computadoras. Trabajan a una altísima velocidad de procesamiento. Se utilizan para trabajos científicos creando modelos matemáticos del mundo real, simulación.

También pueden ser clasificadas de acuerdo a su forma de manejo de los datos.

Computadora Digital - Aquella en que todas las magnitudes se traducen en números, con los cuales opera para realizar los cálculos.

Computadora Analógica - Aparato computador cuyos componentes se ajustan de modo que sus leyes físicas de funcionamiento sean análogas a las leyes matemáticas de proceso que se trata de estudiar.

#### 1.1.5. Generaciones de computadoras.

Por los avances tecnológicos, aumento en la velocidad de procesamiento, disminución en su tamaño, aumentó en su número de aplicaciones, disminución de costos. La evolución de las computadoras se han dividido en generaciones:

##### Primera generación

1951

John Von Neumann (Hungría). En la Universidad de Princeton, proyecto EDVAC, Electronic Discrete Variable Automatic Calculator. Basado en el concepto de "PROGRAMA ALMACENADO": la máquina almacena, además de los datos, las instrucciones que regirán su funcionamiento. No sólo calculaba, sino que trabajaba millones de datos, organizándolos y transformándolos, creando nueva información. CAPACIDAD DE TOMAR DECISIONES LOGICAS O ALTERAR SU FUNCIONAMIENTO DEPENDIENDO DEL RESULTADO DE SUS OPERACIONES Y DE LOS DATOS MANEJADOS. Es un prototipo de las computadoras modernas: UNIVAC, EDSAC, SEAC. Diferencias: circuitos de tubos de vacío; programación en lenguaje de máquina; grandes y costosas.

La Compañía Remington Rand desarrolló UNIVAC I, Universal Automatic Computer. Primera de tipo comercial. Características: Uso de cinta magnética para suministro de datos; procesamiento de datos alfanuméricos; programa de traducción de lenguaje particular a lenguaje de máquina. Diferencias: Uso de bulbos que consumían mucha energía y producían

demasiado calor lo que obligó a implementar equipos de refrigeración.

### Segunda generación

Aprox 1960

Características:

- Los transistores sustituyen a bulbos.
- Reducción de las deficiencias con modelos anteriores.
- Memorias de ferritas.
- Reducción de tamaños.
- Programación de sistemas.
- Programación en lenguajes de alto nivel.
- Ejemplos: IBM 1401; BURROUGHS 5500; BENDIX 6-20.

### Tercera generación

1964

Características:

- Circuitos integrados monolíticos.
- Aumento de la velocidad de operación.
- Mayor confiabilidad.
- Disminución de costo y tamaño.
- Compatibilidad de componentes, permitiendo modificación o expansión de sistemas sin alterar los sistemas básicos.
- Fines científicos y comerciales.
- Personal especializado.
- Ejemplos: IBM-360; BURROUGHS 6700.

### Cuarta generación

70's

Características:

- Circuitos integrados
- Enorme reducción de espacio: integración a gran escala de circuitos, Large Scale Integration.
- Aparecen los microprocesadores, dispositivo electrónico LSI que realiza las funciones de la UCP (cerebro) de la computadora.
- Ejemplos: IBM 4330; BURROUGHS 6900 y 7900.

### Últimas novedades.

Se piensa ya en una quinta generación de computadoras (primeros años del siglo XXI) con las siguientes características estructurales:

- Estarán hechas con microcircuitos de alta integración, cuyo funcionamiento se dará con un alto grado de paralelismo y emulando algunas características de las redes neuronales, como funciona el cerebro humano.

- Operarán con base en la inteligencia artificial, es decir, tratando de aplicar los procesos del razonamiento humano utilizados para la resolución de problemas a una computadora, pensando en la posibilidad de comunicarse con ella de manera similar a como nos comunicamos con otras personas, en lo que conocemos como lenguaje natural.

## 1.2. Terminología básica usada en informática y computación

Computadora u Ordenador	Algo que computa; especificando, un dispositivo electrónico programable que puede almacenar, recuperar y procesar datos.
Hardware	Los componentes físicos de una computadora.
Software	Los programas de una computadora; el conjunto de todos los programas utilizables en una computadora. Componente lógico.
Informática	Disciplina que abarca las diversas técnicas y actividades relacionadas con el tratamiento lógico y automático de la información, en cuanto ésta es soporte de conocimientos y comunicación humana. Se apoya fundamentalmente en la utilización de los ordenadores o calculadores electrónicos y también en un tipo peculiar de análisis conceptual, que implica una actitud mental característica en la manera de afrontar los problemas de la información, que la vincula con la cibernética.
Computación	Conjunto de disciplinas y técnicas desarrolladas para el tratamiento automático de la información.
Cibernética	Ciencia que estudia el funcionamiento de las conexiones nerviosas en los seres vivos. Arte de construir y manejar aparatos y máquinas que mediante procedimientos electrónicos efectúan automáticamente cálculos complicados y otras operaciones similares.
Bit	Mínima unidad de información equivalente a la elección entre dos posibilidades (0 y 1) igualmente probables.
Byte	Número de bits que la computadora necesita para representar un solo carácter o palabra.
Palabra (WORD)	Es la menor unidad direccionable de almacenamiento en un sistema de cómputo.

Dato	Representación de cualquier información de una manera adecuada para ser procesada por la computadora.
Información	Son datos que ya han sido procesados o manipulados, en este caso por una computadora.
Archivo	Forma de organización de datos en la computadora.
Comando (Orden)	Instrucción en lenguaje de máquina ejecutado por un canal de datos.
Programa	Conjunto detallado y explícito de instrucciones de computadora para realizar algún trabajo. Un programa se escribe en un lenguaje adecuado para la entrada a la computadora; es completo e independiente; es divisible en procesos para comodidad del programador.
Software de Aplicación	Programa diseñado para realizar una función específica y resolver un tipo de problema.

### 1.3. Implicaciones sociales de las computadoras

En la actualidad, con los avances tecnológicos logrados en este siglo, la sociedad ha aprendido a convivir con dos procesos que han influido en su vida diaria que son la automatización y la computarización.

**Automatización** .- Es el diseño de instrucciones que den solución a tareas rutinarias, a través de la utilización de dispositivos electrónicos, sustituyendo el esfuerzo del ser humano y permitiéndole la realización de otras actividades.

**Computarización** .- Es el uso de la computadora para resolver problemas o procesos relativos al intelecto humano, su función es la de almacenar datos, procesarlos y realizar complejos análisis e interpretaciones de los mismos para finalmente poder hacer uso de la información generada.

Pero ésto ha ocasionado problemas y discusiones en varios sentidos:

- Violación del derecho a la privacidad.
- Deshumanización de las actividades.
- Propiciar el desempleo en niveles técnicamente bajos.
- Centralización del poder en quienes saben manejar bancos de información de datos automatizados.

Estos problemas son temas de discusión y obligan a los educadores a tomar conciencia y a concientizar a los estudiantes sobre los valores éticos que como profesionales deben tener presentes cuando su trabajo este relacionado con las computadoras.

#### 1.4. Aplicaciones

En el inicio del desarrollo de las computadoras fueron creadas con la finalidad de ser auxiliares en el procesamiento numérico, es decir, realizar una gran cantidad de operaciones numéricas exactas en el menor tiempo posible.

En su evolución, las computadoras han cambiado su función a procesadores de datos, permitiendo el almacenamiento, procesamiento y recuperación de datos en forma rápida y eficiente.

En la actualidad se desarrollan como procesadores del conocimiento, siendo así una extensión del cerebro humano.

##### 1.4.1. Aplicaciones militares.

Durante la Segunda Guerra Mundial, con el diseño, construcción y producción de nuevos tipos de armamento como proyectiles, cohetes anti-aéreos y antitanques, se requirió a su vez la elaboración de tablas que indicaban las trayectorias que éstos seguirían al ser empleados bajo ciertas condiciones como eran: dirección del viento, medidas de casquillos y municiones, inclinación del cañón, etcétera. Estas tablas de trayectorias con todas sus posibles combinaciones de condiciones debían ser enviadas junto con el armamento a los campos de batalla pero su elaboración de manera manual requería en promedio 7 horas-hombre para el cálculo de una sola trayectoria.

Para auxiliar en la elaboración de dichas tablas de trayectorias balísticas, se construyó la computadora ENIAC, pero cuando estuvo terminada la guerra había finalizado dos meses antes.

Posteriormente, las computadoras han sido utilizadas por los militares en el diseño de cohetes y proyectiles, en el control de proyectiles teledirigidos, en pruebas de funcionamiento de cohetes balísticos, en la simulación de tácticas militares, en vuelos espaciales, entre otras aplicaciones.

#### 1.4.2. Aplicaciones científicas.

Diseñadas inicialmente por los científicos matemáticos como herramienta de ayuda en los cálculos numéricos, las computadoras ofrecen una alternativa de solución a problemas planteados con grandes y laboriosos sistemas de ecuaciones, permitiendo a su vez verificar las mejores alternativas entre varias opciones de solución de un problema.

En campos como la meteorología, sismología, física nuclear, química cuántica permite dar solución a grandes sistemas de ecuaciones que representan algún fenómeno físico.

En los laboratorios de Ingeniería, Física y Química se utilizan para controlar el desarrollo de experimentos, de los dispositivos de adquisición de datos y del análisis de los datos generados sustituyendo así algunos o todos los instrumentos científicos. Permite también que el científico verifique sus propios datos y pueda hacer uso de la nueva información generando un nuevo ambiente experimental (retroalimentación).

Realizar lecturas constantes de sucesos o fenómenos en lugares distantes o remotos, por ejemplo, permite desde otro planeta el envío de imágenes, lecturas de temperatura, de componentes ambientales, entre otras y verificar la validez de la información.

#### 1.4.3. Aplicaciones administrativas.

El manejo de computadoras ha permitido la automatización de procedimientos rutinarios propios de una empresa :

- Elaboración de nóminas,
- Procesos contables y financieros,
- Control de adquisiciones, inventarios y almacenes.

Los paquetes de apoyo gerencial, además, han permitido la proyección a futuro en distintos rubros del mercado económico a fin de mejorar la planeación organizacional de la empresa.

Otra área importante de aplicación es la del manejo de información almacenada con cierta organización en sus datos, en la cual sea posible llevar a cabo movimientos (altas, bajas y cambios) sin necesidad de perder la estructura original de la información, permitiendo la recuperación y consulta de la información, así como la generación de reportes.

Esto es posible mediante el uso de bases de datos cuyas aplicaciones no son sólo administrativas sino también de otras áreas como economía, demografía, mercadotecnia, entre otras.

#### 1.4.4. Aplicaciones industriales.

El uso de computadoras en la industria en general ha permitido el diseñar y construir instrumentos que realicen además de mediciones, análisis y toma de decisiones específicas. Estos instrumentos deben contar con la capacidad de medir constantemente (varias en un segundo o en lapsos cortos de tiempo) un sinnúmero de fenómenos con exactitud, velocidad y confiabilidad. Ya que del análisis, procesamiento de los datos y su correlación con otros fenómenos dependerá la nueva información que sea utilizada en la retroalimentación del proceso.

El uso de la computación permite además el seguimiento y control de la maquinaria. Un sistema de control funciona bajo el principio de un ciclo de retroalimentación, que provee información que es utilizada como guía en la toma de decisiones que modifiquen las propias cantidades medidas. A este tipo de aplicación se le conoce como "aplicación en tiempo real".

El diseño asistido por computadora permite a su vez, a través del uso de un catálogo de descripción de elementos requeridos, la elaboración de un determinado producto para someterlo a diversas condiciones de operación.

Maquinaria de control numérico, que no es forzosamente de tiempo real. La aplicación de estos sistemas son: Ingeniería (mecánica, química, industrial, eléctrica y electrónica), aeronáutica y arquitectura.

#### 1.4.5. Aplicaciones en la investigación.

Participa en la mayoría de los campos de investigación ya sea como herramientas de apoyo en dichas investigaciones o bien como el objeto mismo de la investigación.

Entre los temas que se desarrollan o trabajan en ella están:

- Procesamiento y creación de imágenes.
- Administración. Análisis de costos y de sistemas.
- Simulación.
- Impactos sociales de la computación.
- Arquitectura de computadoras.
- Técnicas de programación.
- Inteligencia artificial.

#### 1.4.6. Aplicaciones educativas.

La computadora se utiliza en la educación ya sea utilizando simuladores, sistemas deductivos o asistencia por computadora.

Los simuladores son programas hechas para la computadora con el fin de imitar o demostrar parte o todo el comportamiento.



por lo general, de un fenómeno físico, económico, etcétera. Por ejemplo, programar a la computadora para comportarse como una nave espacial o como una planta química de procesamiento.

La simulación permite al usuario estudiar el comportamiento de una situación específica bajo ciertas condiciones y en periodos definidos de tiempo sin el riesgo de errores reales o cuando la experimentación directa resulta imposible, no es práctica, es costosa, o demasiado lenta.

La simulación puede ser determinística o probabilística.

**Simulación Determinística** .- Cuando el estado actual del evento o fenómeno depende directamente del estado anterior así como de las condiciones iniciales o variables introducidas, ya durante el proceso. Es determinístico si el estado del sistema en el momento  $t+1$  depende unívocamente del estado del sistema en el tiempo  $t$  y del estímulo en el tiempo  $t$ . Desde un mismo estado y frente al mismo estímulo, la respuesta es siempre la misma y es única. Por ejemplo en: Física, Química, Matemáticas aplicadas, Biología, Sociología, Demografía, Sismografía, Meteorología, entre otras.

**Simulación Probabilística** .- El estado del evento o fenómeno no depende de las condiciones iniciales, sino que durante el proceso ha tomado opciones de manera aleatoria de acuerdo a una situación particular. Es probabilística si se utiliza alguna función probabilística para determinar el siguiente estado, dado un estado actual y un cierto estímulo. Aunque sea probabilístico, las condiciones iniciales pueden "sugerir" ó inducir determinadas elecciones. Utilizada en: Lingüística, Jurisprudencia, Diagnósticos clínicos, entre otros.

Los sistemas deductivos obtienen sus propias conclusiones de acuerdo con ciertas premisas dadas o por ciertas reglas de lógica deductivas introducidas en las instrucciones que conforman el programa.

#### 1.4.7. Aplicaciones a la medicina.

Utilizada por lo general en monitoreo en línea de pacientes de cuidados intensivos, vigilando constantemente (varias veces por segundo) sus signos vitales; la elaboración de dietas, el control de fármacos, el análisis de electrocardiogramas y encefalogramas, la investigación médica y la administración hospitalaria.

#### 1.4.8. Vida diaria.

Hoy día no hay actividad humana en la cual no intervenga la computadora. Entre ellas :

- Caricatura electrónica.
- Elaboración de banderines o carteles para eventos artísticos, sociales o culturales.
- Video juegos.
- Horóscopos.
- Dietas.
- Control de presupuesto familiar.
- Robótica.
- Exploración por satélites.
- Diseño para la construcción.
- Diseño gráfico.
- Análisis y composición musical.
- Escritura de guiones para televisión.
- Enseñanza asistida por computadora.
- Control de tráfico aéreo, terrestre, ferroviario, marítimo.
- Fichas policíacas.
- Control y acoplamiento de vehiculos espaciales.
- Diseño y control de armamento.
- Defensa aérea.
- Sistemas de alarma.
- Control de cuentas bancarias.
- Cajeros automáticos.
- Pintura.
- Administración escolar.
- Telefonía y telecomunicaciones, entre otras.

## II. ESTRUCTURA FÍSICA DE UNA COMPUTADORA

## II. ESTRUCTURA FÍSICA DE UNA COMPUTADORA

Esta unidad tiene como objetivo enseñar al alumno a identificar las partes esenciales que componen un equipo de cómputo y la función específica de cada una de esas partes.

### 2.1. Hardware.



Un conjunto de instrucciones serán la mínima información necesaria para que la computadora pueda comenzar a realizar cualquier trabajo.

#### 2.1.1. Definición.

El hardware corresponde al conjunto de componentes físicos de la computadora, es decir, los circuitos electrónicos y dispositivos electromecánicos que constituyen el sistema de computación.

#### 2.1.2. Equipo Central de una computadora (Arquitectura Von Neuman).

El modelo de computadora básico, está constituido por tres unidades básicas:

- Unidades de Entrada y/o Salida (U E/S).
- Unidad de Memoria Principal.
- Unidad Central de Proceso (C P U).

#### 2.1.3. Unidades de entrada y/o salida.

Dispositivos utilizados por una computadora para recibir o detectar en forma sensible los datos provenientes del ambiente externo y también para transferir datos desde su medio interno al ambiente.

Por medio de la unidad de entrada, el usuario le proporciona a la computadora las instrucciones y los datos necesarias.

Algunos de los dispositivos periféricos de entrada más comunes son:

- Lectora de cinta de papel.
- Lectora óptica.
- Lectora de cinta magnética.
- Teclados alfanuméricos.
- Micrófonos.
- Cámaras de televisión.
- Sensores térmicos, eléctricos, entre otros.
- Discos
- Discos compactos, CD.
- Ratón (Mouse)

Los resultados que se generen de dicha comunicación son enviados por la computadora hacia el exterior a través de las unidades de salida. Algunos de los dispositivos periféricos de salida más comunes son:

- Impresoras de papel
- Graficadores
- Perforadora de cinta de papel
- Pantalla de video
- Autoparlantes
- Microfichas
- Discos
- Campanas de alarmas

Si las unidades permiten efectuar tanto la introducción como la exposición de información se les llama unidades de entrada y salida, entre ellas están:

- Fax
- Modem
- Unidades de disco
- Unidades de cinta

#### 2.1.4. Unidad de memoria principal.

El concepto de memoria se aplica a todo dispositivo electrónico que pueda almacenar información. La memoria es una sucesión ordenada de celdas de almacenamiento, cada una conteniendo un fragmento de información. La unidad de memoria principal de la computadora permite, almacenar tanto instrucciones como datos, y además resultados parciales y finales que se generen. Se considera integral, interno o primario para el sistema de computación y es la principal área de trabajo a partir de la cual pueden ejecutarse instrucciones o recuperarse operandos para la manipulación de datos. Las instrucciones a ejecutarse en la computadora deben estar localizadas en la memoria principal cuando se están ejecutando. Los datos sobre los que se opera la instrucción ejecutándose también deben estar en ella, los datos pueden estar en dispositivos de almacenamiento

externo, pero para utilizarlos se deben colocar en la memoria, lo que hace que el tiempo de ejecución sea muy prolongado.

Existen dos tipos de memorias de acuerdo a sus características:

- **Memoria RAM, Random Access Memory .-** Memoria de Acceso Directo.
  - Memoria de lectura y escritura
  - Pierde información con la falta de energía eléctrica, volátil.
  - Se pueden direccionar sus celdas de manera directa.
  - En esta memoria residen los programas y datos del usuario, y resultados temporales que se utilizarán en cálculos posteriores.
  
- **Memoria ROM, Read Only Memory .-** Memoria de Sólo Lectura.
  - Memoria de lectura.
  - Instrucciones permanentes grabadas con anterioridad.
  - No se ve afectada por la falta de corriente eléctrica.
  - El usuario podrá utilizar las instrucciones que contiene, mas no podrá modificarlas.

Existe otro tipo de memoria más económica que permite almacenar enormes cantidades de información, se trata de la memoria secundaria o externa de la computadora. Este tipo de memoria es de lectura y escritura, y la información no se ve afectada por la falta de corriente eléctrica ya que los datos son almacenados en medios magnéticos. Aun cuando tiene un tiempo más lento de acceso, es menos costoso que la memoria interna. La memoria externa está constituida por: discos, tambores magnéticos y cintas magnéticas para equipos grandes, diskettes, cassettes para equipos pequeños.

A las actividades de almacenar y extraer información de la memoria se les conoce como escritura y lectura respectivamente.

#### 2.1.5. Unidad central de proceso.

Las funciones de la Unidad Central de Proceso de la computadora son análogas, en cierto sentido, a las de un cerebro humano. Esta se encarga de coordinar y organizar cada una de las funciones de las demás partes de la computadora.

También se encarga de ejecutar, al pie de la letra, cada una de las instrucciones que el usuario le solicita.

Para tales acciones la Unidad Central de Proceso cuenta con dos elementos fundamentales:

- Unidad de Control.
- Unidad Aritmética y Lógica.

### Unidad de Control

Sus funciones consisten en leer y escribir contenidos en las celdas de la memoria; el CPU (Unidad Central de Proceso) a través de la Unidad de Control, ejecuta cuatro pasos a gran velocidad: extrae de memoria, decodifica, ejecuta y ajusta el contador del programa.

### Unidad Aritmética y Lógica

Como su nombre lo indica es la unidad encargada de realizar todas las operaciones aritméticas y lógicas aplicables a los datos almacenados. Esta unidad puede realizar un número reducido de operaciones elementales a gran velocidad. Las operaciones que puede efectuar son:

- Suma y resta de dos números.
- Multiplicación y división de dos números, en computadoras muy poderosas.
- Operaciones lógicas AND, OR, NOT.
- Comparación entre dos valores.

Por último, el CPU además cuenta con un conjunto de celdas especiales, con características similares a las de memoria, pero siendo éstas más poderosas, más rápidas y muchas de ellas tienen funciones específicas, como el acumulador que está alambrado para las operaciones básicas; sólo que las utiliza tan frecuentemente que es necesario que no formen parte de la memoria. A estas celdas se les conoce con el nombre de Registros Especiales. El de mayor uso se llama acumulador y es ahí donde se ejecutan las operaciones de la UAL.

#### 2.1.6. Equipo Periférico.

Son los dispositivos que sirven para alimentar y recibir los datos que procesa la computadora. El equipo periférico se conecta al equipo central (CPU y Memoria Principal) a través de los subsistemas de entrada y salida. El equipo periférico comprende los dispositivos de entrada, salida y la memoria secundaria.

En general cada dispositivo periférico entrega información a la computadora o recibe información de la computadora. En el primer caso se trata de equipo periférico de entrada, en el segundo, equipo periférico de salida y el que cumple ambas funciones (entregar y recibir información) se llaman equipo periférico de entrada y salida.

Los verbos almacenar, recuperar y procesar implican a las cinco componentes básicas de la mayoría de las computadoras: Unidad de Memoria, Unidad Aritmética y Lógica, Unidad de Control y Dispositivos de Entrada y/o Salida.

## 2.2. Teclado

Dispositivo de entrada del tipo de máquina de escribir que se emplea para introducir en forma manual información y datos a la computadora. Al oprimir una tecla se genera una señal específica en código que representa el carácter o símbolo. Además de su disposición estándar de teclas QWERTY, común en casi todas las máquinas de escribir. Además de estas teclas estándar, las computadoras utilizan teclas adicionales que desarrollan una variedad de tareas (Fig 2.1).

Un teclado de la computadora se puede dividir en tres partes:

- Teclas alfanuméricas. Ubicadas en la parte central del teclado y sirven para teclear caracteres alfabéticos, numéricos y símbolos de puntuación.
- Teclado numérico. Ubicado en la parte derecha. Permite ingresar rápidamente números y símbolos aritméticos para efectuar cálculos matemáticos y también para mover el cursor.
- Teclas de funciones (F1, F2, ... Fn). Ubicadas en la parte superior o izquierda del teclado, son teclas de funciones especiales que permiten al usuario responder preguntas, dar instrucciones o elegir entre las opciones de un menú todo ello dependiendo del programa que se use.

### 2.2.1. Teclas de funciones.

- a. Teclas de función (F1, F2, ... Fn) - Realizan una función o actividad especial, esto dependerá del programa en uso y podrá cambiar, al cambiar de programa.
- b. Tecla ENTER - Necesaria para indicarle a la computadora que acepte un comando, instrucción u orden y ésta pueda responder a la petición hecha.
- c. Tecla de Escape (ESC) - Utilizada para interrumpir o cancelar un comando previo.
- d. Tecla de control (CTRL) - Se utiliza para enviar un comando o pieza de programa a la computadora. Se mantiene oprimida mientras se pulsan la tecla o teclas restantes. Se le representa con un acento circunflejo (^) para indicar su uso.



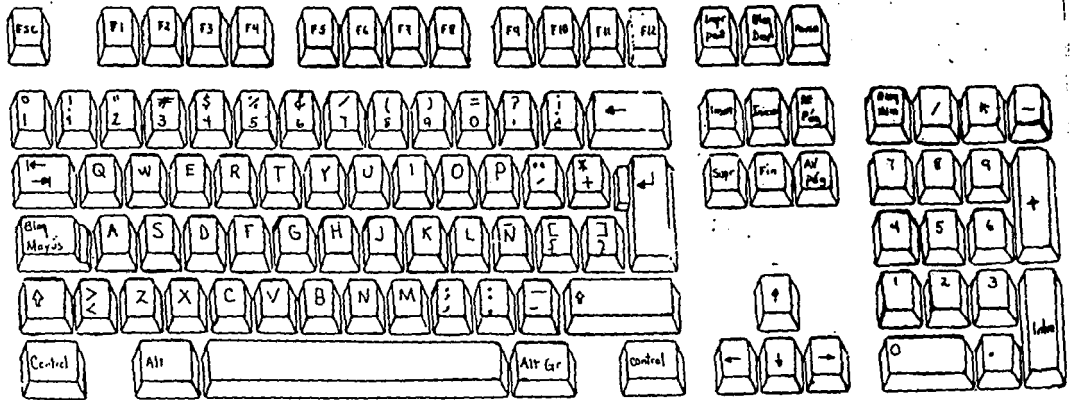


Figura 2.1 Teclado de Computadora

- e. Tecla ALT - Similar a la tecla de control. Necesita oprimirse y combinarse con otras teclas para realizar cierta acción o proceso.
- f. Tecla de mayúsculas (SHIFT) - Permite escribir caracteres en mayúsculas estando en modo de minúsculas y viceversa.
- g. Tecla de bloque de mayúsculas (CAPS LOCK) - Parecida a la tecla SHIFT. Se usa cuando se necesitan todas las letras mayúsculas pero no causa el mismo efecto en las teclas numéricas del teclado alfanumérico.
- h. Tecla de inserción (INSERT) - Permite agregar texto en cualquier posición de un archivo.
- i. Tecla de borrado (DELETE) - Elimina el caracter ubicado en la posición donde se encuentra el cursor.
- j. Tecla de retroceso (BACKSPACE) - Elimina el caracter ubicado a la izquierda del cursor.
- k. Tecla de inicio (HOME) - En general, coloca el cursor en la esquina superior izquierda o al inicio de un renglón.
- l. Tecla de final (END) - En general, coloca el cursor en la posición ya utilizada más baja de la pantalla ó al final de un renglón.
- m. Tecla para subir de página (PAGE UP) - En una tarea de enrollamiento muestra por lo general los 24 renglones anteriores a los expuestos de un archivo.
- n. Tecla para bajar de página (PAGE DOWN) - En una tarea de enrollamiento muestra por lo general los 24 renglones posteriores a los expuestos de un archivo.
- o. Tecla de impresión de pantalla (PRTSCR) - Usada conjuntamente con la tecla SHIFT permite imprimir casi cualquier cosa que se este visualizando en la pantalla de la computadora, con las gráficas no siempre es posible.
- p. Tecla de pausa (PAUSE) - Permite detener momentáneamente la ejecución de una instrucción o programa.

Aquí se analizó el funcionamiento de cada una de las teclas adicionales del teclado, pero la función que cada una de ellas desarrolle dependerá directamente del sistema operativo con que se éste trabajando y de la aplicación específica que se esté ejecutando.

### III. PROCESAMIENTO DE TEXTOS

### III. PROCESADORES DE TEXTO.

El propósito de la unidad es que el alumno compruebe el beneficio de conocer el manejo de las herramientas de uso de una computadora, como son los procesadores de texto, elaborando sus tareas escolares.

#### 3.1. Introducción

La popularidad alcanzada por las computadoras personales para mediados de la década de 1980, la gran variedad de usuarios no expertos ni en ingeniería, ni en computación y que demandaban la utilización sencilla de los equipos costados ya por millones, hizo que la industria de la microcomputación desarrollara programas fáciles de manejar.

Este tipo de programas recibe el nombre genérico de "paquetes" ya que son soluciones "empaquetadas" para problemas específicos y no piden del operador más allá de un conocimiento mínimo sobre el manejo de la computadora, y casi todos disponen de manuales y textos de ayuda para auxiliar al usuario en el uso del sistema.

El conocimiento en el uso de uno o varios paquetes no implica de ninguna manera el conocimiento sólido sobre la ciencia de la computación, incluyendo de la programación. Cabe advertir que toda institución educativa a nivel medio superior y superior debe fijar entre sus objetivos el formar creadores y diseñadores de sistemas y paquetes, sin conformarse con meros usuarios y consumidores de la tecnología de los "paquetes".

Los paquetes tienen un uso específico. De manera general existen cinco áreas de aplicación:

- Hojas electrónicas
- Procesadores de palabra o texto
- Comunicación entre datos
- Manejo de bases de datos
- Gráficos

En nuestro caso particular, haremos referencia a los procesadores de palabra.

#### 3.2. Programa de aplicación procesador de texto

Los procesadores de palabra han influido en los métodos y hábitos de trabajo. Sin importar el contenido o tipo de texto a trabajar, para que pueda ser manejado por la computadora, éste debe ser colocado en un archivo, que define a una colección de caracteres grabados en algún dispositivo de almacenamiento.

Para entender como realiza su trabajo un procesador de palabra, primero explicaremos la forma de trabajo de su antecesor: "el editor".

Editor- Nombre genérico que se le da a un programa cuya versatilidad permite manipular texto. Entre las funciones que realiza un editor están:

- Encontrar una cadena de caracteres dentro de un texto.
- Reemplazar una cadena de caracteres por otra dentro del texto.
- Hacer búsquedas y reemplazos globales.
- Insertar cadenas en cualquier posición del texto.
- Eliminar cualquier subcadena de texto.

El uso o manejo de éstas funciones debe ser claro, sencillo y que si en un momento dado deseamos deshacer el efecto de una función ya ejecutada, exista forma de hacerlo, es decir, exista también una función de recuperación. Para esto muchos editores cuentan con "menús" de ayuda que están presentes durante todo el uso del editor o que son invocados por teclas especiales.

Los editores trabajan con una copia del archivo a editar, siendo en esta copia donde se trabaja de esta forma si el usuario desea abandonar el trabajo podrá volver a su texto original. En caso contrario, el editor sustituye al archivo original por el que se terminó de editar. La mayoría de los editores guarda el texto antes de editarse en un respaldo (backup) asignándole a dicha copia una identificación final (.bak).

Los editores pueden ser de dos tipos: editores en línea o editores en pantalla.

Los editores en línea trabajan con cada una de las líneas o renglones presentes en el texto, ya que fueron pensados para trabajar únicamente con programas de computadora. Las modificaciones que se realizan en estos editores no siempre son visibles al momento y la verificación de la posición del cursor requiere mucho tiempo. En él, las líneas vienen numeradas y las acciones se efectúan sobre una línea particular identificada por un número de línea o en su defecto, la línea sobre la cual se encuentra el cursor. El cursor no es visible sólo que en un momento dado pidamos esa información, entonces el sistema imprime la línea frente a la cual está el cursor, por lo tanto, no puede brincar o retroceder a una posición arbitraria deseada.

Los editores en pantalla, trabaja sobre toda una pantalla, que siempre se encuentra llena con una porción de archivo. Muestra en la pantalla un pequeño cuadro blanco o el símbolo subraya ( \_ ) parpadeando que todo el tiempo esta frente a la posición, caracter, disponible en ese momento. Permite el movimiento en cualquier dirección dentro de la pantalla. Las modificaciones que se efectúan al texto son mostradas en la pantalla inmediatamente. Con el uso de este tipo de editor, es conveniente conocer la resolución de la pantalla con que se trabaja, ésta esta dada por el número de puntos que son

utilizados para "construir" o mostrar cada uno de los caracteres, dichos puntos se encuentran organizados en una pequeña matriz. Cada caracter se forma con las diferentes combinaciones de puntos. Es importante saber que entre más alta o mayor sea la resolución de una pantalla, más detalle y mejor calidad de imagen se tendrá.

Para lograr la presentación final que requiere el texto se puede hacer uso de uno de los dos métodos siguientes:

- Manipulador de formatos, o
- Editor / manipulador de formatos.

En general, un manipulador de formatos tiene entre otras las siguientes funciones:

- Justificación. El editor acomoda el texto entre los márgenes señalados, insertando de ser necesario espacios en blanco entre las palabras hasta ajustar el texto a la línea con palabras completas a los márgenes.
- Posibilita que el usuario decida sobre el número de columnas y renglones con que contará la página, así como el valor inicial en la numeración de las páginas.
- Permite el uso de tabuladores.
- Permite el uso de la gama de caracteres con que cuenta la impresora.

Con el texto se mezclan secuencias de caracteres especiales con un significado de uso de una función especial o comando dirigidas al editor/manipulador de formatos. La instrucción o mensaje que genera esta secuencia de caracteres se conoce como comando y la marca que queda en la pantalla de edición para indicar la introducción o uso de un comando se llama caracteres de control de impresión. Con esto, el texto quedará entremezclado con las marcas de comandos (identificables por el efecto visual que presentan, diferente al del texto) haciendo uso del manipulador de formatos podremos ver cual será el resultado final del texto editado.

Algunos editores permiten realizar un preanálisis de la impresión final que tiene el texto editado antes de la impresión, esto hace posible verificar el conveniente uso que se hizo de los comandos y en caso de ser necesario modificarlos.

De esta sección se pueden desprender las siguientes recomendaciones o sugerencias:

- Por su facilidad de trabajo son preferibles los editores de pantalla que los editores de línea.
- Es preferible un editor que muestra su menú de ayuda de manera permanente (sobre todo para los principiantes) que uno que lo oculta.
- Es recomendable guardar los archivos de respaldo (BACKUP) por si se presenta un daño accidental o se requiere recuperar información anterior, esto siempre y cuando se cuente con espacio en el disco.

- Tiene una gran ventaja el poder visualizar el texto tal cual va a quedar antes de la impresión, ya que evitará el consumo excesivo de papel en caso de necesitar correcciones.

Los editores/manipulador de formatos reciben el nombre genérico de procesadores de palabra. Los procesadores tienen un ambiente de trabajo creado, es decir, presentan plantillas o pantallas diseñadas de trabajo con instrucciones de manejo (menús) y ubicación de la información, entre otros. Los procesadores por construcción cuenta con una pantalla de edición que tiene valores preestablecidos de tamaño de página, sangrías, tipo de letra, ancho de renglón, entre otros. El texto capturado se irá acomodando dentro de éstos parámetros y los cuales pueden ser modificados por el usuario según sus necesidades de trabajo.

Son preferibles los procesadores que manejan sus opciones de trabajo tanto por ambiente como por comandos, ya que facilitan la edición. Con excepción de los caracteres especiales, estos procesadores muestran la presentación final que tendrá el texto. El procesador que se manejará en este trabajo (WORDSTAR) es de este tipo. Los que además muestran caracteres especiales y diferentes tamaños requieren el uso de pantalla de buena resolución.

Los editores que permiten ir armando sobre la marcha el texto reciben el nombre genérico de WYSIWYG (What You See Is What You Get) que traducido significa: Lo Que Ves Es Lo Que Obtienes (LQVELQO). De este tipo son los procesadores siguientes:

- El procesador WORDSTAR debe su aceptación a que cuenta con un menú de ayuda permanente que permite el uso de todos los comandos.
- El procesador WORD cuenta además con la facilidad de numeración de capítulos, verificación automática de índice y un menú compacto siempre presente.
- El procesador CHI WRITER permite el manejo de simbología matemática y la edición de caracteres especiales mostrándolos siempre si son requeridos en la construcción del texto final.

Entre mayores sean las funciones permitidas por el editor/manipulador de formatos el paquete debe ser más inteligente y más grandes sus propios programas, estos a su vez deben poder ser utilizados en subconjuntos, no con todas sus funciones pero si las principales y así poder ser utilizados en computadoras de unidad de disco flexible ya que el conjunto completo sólo puede ser utilizado si se cuenta con un disco duro por su capacidad.

Finalmente se debe elegir el procesador de acuerdo a las siguientes observaciones:

- Características del texto que se desea elaborar.
- Resolución de la pantalla con que se trabaja.
- Cantidad y tipo de disco con que se cuente.
- Tipo de impresora en la que se imprime.

Lanzado al mercado en 1978, WORDSTAR ha pasado por varias versiones; emplea una amplia combinación de comandos y teclas de control; despliega el texto página por página, ofreciendo el enrollamiento del mismo; despliega fielmente el texto como deberá salir en la impresión.

Es posible definir bloques de texto, con el fin de efectuar algún proceso con ellos, como copiar, mover, borrar o escribirlos en otro archivo. Muestra de manera permanente un menú de comandos básicos en la parte superior de la pantalla y a través de éste hacer consultas de uso de comandos especiales.

Al trabajar con WORDSTAR no es necesario hacer cambio de línea al final de cada renglón, ya que realiza un ajuste de palabras automático al tamaño del renglón indicado por los márgenes. Permite centrar textos, efectuar tabulaciones, marcar márgenes, espaciado entre líneas, entre otras.

El WORDSTAR requiere para su funcionamiento del siguiente equipo:

- Procesador 386 en adelante
- 4 Megabytes de memoria.
- Monitor VGA.

### 3.3. Comandos básicos para el manejo del procesador de textos

En esta sección se presentan las instrucciones y comandos básicos para el manejo del procesador de textos WORDSTAR.

#### 3.3.1. Entrada a WORDSTAR

1. Cargar el sistema operativo y esperar la identificación de la unidad de disco ( A > o B >) en la pantalla.
2. Colocar el disco de Wordstar en el drive A y el disco de trabajo (donde se grabará el texto editado) en el drive B.
3. Teclar WS y oprimir la tecla de <ENTER>.

#### 3.3.2. Menú de apertura

Una vez dentro de WORDSTAR aparecerá un menú inicial llamado Menú de Apertura (Opening menu), mostrando las siguientes opciones de trabajo.



### Comandos Preliminares

- L - (Change logged drive/directory) Permite cambiar o seleccionar la unidad de disco y/o directorio de trabajo.
- F - (Turn directory on/off) Da opción de observar o no el directorio del disco de trabajo.
- J - (Help) Permite desplegar niveles de ayuda. Los menús de ayuda se despliegan en pantalla pero pueden eliminarse para tener mayor visibilidad del texto en edición, esto se hace modificando el nivel de ayuda.

### Comandos para creación o edición de un archivo

- D - (Open a document) Abrir un archivo de tipo texto o documento.
- N - (Open a non-document) Abrir un archivo no texto, en él se podrá editar un archivo de datos o un programa en cualquier lenguaje de programación.

### Comandos de manejo de archivos

- P - (Print a file) Imprimir un archivo.
- E - (Rename a file) Cambiar el nombre a un archivo.
- O - (Copy a file) Copiar un archivo.
- Y - (Delete a file) Borrar un archivo.

### Comandos del sistema

- R - (Run a DOS command) Ejecutar un comando desde el sistema operativo
- X - (Exit WordStar) Salir de WordStar

### Operaciones del WORDSTAR

- M - (Merge print a file) Llamar al MailMerge o manejador de correspondencia.

### 3.3.3. Cambio de unidad de disco (drive) y/o subdirectorio

1. Estar en el Menú de Apertura (Opening Menu).
2. Teclar L y la tecla del drive al cual nos queremos cambiar.

En nuestro caso teclear B y oprimir <ENTER>. Pero en general teclear la letra de la unidad y/o nombre del directorio a la cual se desea cambiar.

### 3.3.4. Apertura o edición de un documento

1. Estar en el Menú de Apertura (Opening Menu).
2. Teclear D (Open a document file).
3. Teclear el nombre del archivo. Este estará formado por:  
**nombre-de-documento.extensión**
  - a. El nombre-de-documento debe ser: no mayor de ocho caracteres, no tener espacios en blanco, no contener caracteres especiales - el único caracter especial permitido es la subraya (\_). Es recomendable que el nombre del archivo tenga relación con el contenido del mismo.
  - b. En general, la extensión no debe ser mayor de tres caracteres y sirve para identificar el tipo de información que contiene el archivo. Ejemplo: **TAREA.TXT** o **HISTORIA.TXT**.

Una vez capturado el nombre del archivo oprimir la tecla <ENTER>. La computadora verificará si el archivo existe con anterioridad o no. En caso de ya existir lo abrirá y leerá su contenido mostrándolo en la pantalla; sino, enviará un mensaje pidiendo la confirmación de la apertura de un documento con el nombre teclado. La respuesta debe ser Y para confirmar y N para negar.

### 3.3.5. Descripción de la pantalla de edición

Al darle el nombre aparecerá un menú denominado Menú de Edición (Edit Menu) y dos líneas mostrando ciertos parámetros en la parte inmediata superior e inferior del menú, una área en blanco y el cursor posicionado en la esquina superior izquierda de dicha área.

#### Línea de estado y línea maestra

La línea superior al marco del menú de edición es la línea de estado y nos indica:

- \* El archivo que estamos editando, unidad de disco y nombre del archivo.
- \* La página, línea y columna en que se encuentra ubicado el cursor.
- \* Si se está o no en modo de inserción.

La línea inferior al marco del menú de edición es la línea maestra, indica donde está colocado el margen izquierdo con una letra L, el margen derecho con una letra R, y las paradas o altos del tabulador con signos de admiración, !.

La captura del texto se realiza de manera similar que con una máquina de escribir. Aquí, conforme se captura el texto, éste ira apareciendo en el área de edición y será acomodado dentro del espacio indicado por los márgenes (letras L y R) de la línea maestra, en caso de que una palabra rebase el límite del margen derecho (R), ésta es colocada en el siguiente renglón automáticamente y sin necesidad de solicitar el cambio de línea manual con la tecla <ENTER>. En este caso el último carácter de la última palabra que se capturó completa en el renglón se coloca a la altura de la marca del margen derecho (R) y se agregan espacios en blanco suficientes entre las palabras hasta ajustar al tamaño de la línea. Únicamente cuando se termina el párrafo, se desean dejar líneas en blanco se oprime la tecla de <ENTER>.

#### Teclas de edición de uso frecuente

Cuando se cometa algún error, usa la tecla <BACKSPACE> para corregirlo. En caso de que el modo inserción (INSERT) esté habilitado, oprimir la tecla <INSERT> para deshabilitarlo y entrar automáticamente en modo de sobrescritura (OVERWRITE).

Se usa la tecla <TAB> para genera una sangría en la primera línea de cada párrafo en caso de así desearse.

Conforme nos acercamos a la parte baja de la pantalla el texto subirá automáticamente y se desplegará, de tal forma del que el cursor siempre se encuentra visible.

#### Significado de la columna de la extrema derecha

Esta columna proporciona información acerca de las características que guarda cada línea de texto que se va capturando.

Si en la pantalla aparecen una serie de "gorritos" (") significa que aún no se tiene texto en esa área y el cursor no se moverá, a menos que se oprima la tecla de RETURN o la barra de espacios.

El símbolo < significa fin de línea o fin de bloque de texto y se indica tecleando <ENTER>. Se marca también al dejar líneas libres.

Cuando en la columna de la extrema derecha no aparece marca alguna, significa que no se dio <ENTER> en la línea y que se dejó que WORDSTAR acomodara automáticamente el texto en los márgenes indicados.

El símbolo "+" significa que aunque no aparezca en la pantalla, hay texto o espacios en blanco mas allá del borde derecho de la pantalla.

Una línea punteada con una P al final, significa cambio de página.

Una vez abierto el documento se pueden modificar las características de la hoja de edición.

### 3.3.6. Movimientos del cursor

Quando en el texto aparezca el caracter "" antecediendo a una letra, quiere decir que se debe oprimir y sostener la tecla de <CONTROL> mientras se teclean las demas letras.

Podemos colocar el cursor en la posición que deseemos, auxiliandonos de las siguientes opciones:

#### Caracter a caracter

- ^S <-- Un caracter a la izquierda
- ^D --> Un caracter a la derecha

#### Palabra por palabra

- ^A A la primera letra de la palabra a la izquierda
- ^F A la primera letra de la palabra a la derecha

#### Línea por línea

- ^E Coloca el cursor en la línea inmediata superior
- ^X Coloca el cursor en la línea inmediata inferior

#### A los extremos de la pantalla

- ^QS Mueve el cursor al extremo izquierdo de la línea
- ^QD Mueve el cursor al extremo derecho de la línea en que se encuentre
- ^QE El cursor se colocará en la parte superior izquierda de la pantalla actual
- ^QX El cursor se colocará en la última línea de la pantalla que estemos visualizando

^QR Colocaremos el cursor en el principio del archivo  
 ^QC Colocaremos el cursor en el final del archivo  
 <HOME> El cursor se coloca en la esquina superior izquierda  
 <END> El cursor se coloca en la parte inferior de la pantalla

### Mover la pantalla

^Z Desplaza la pantalla una línea hacia abajo  
 ^W Desplaza la pantalla una línea hacia arriba  
 ^R <Pg Up> Sustituye el contenido de la pantalla por las 20 líneas anteriores del documento, la página anterior.  
 ^C Pg Dn Sustituye el contenido de la pantalla por las 20 líneas posteriores del documento, la página posterior.

### 3.3.7. Menú de Ayuda

En caso de que "nos encontremos en apuros" podemos recurrir a las siguientes acciones:

- Visualizar los diferentes menús de ayuda que aparecen en la parte superior de la pantalla. En el menú principal encontramos ayuda de: movimiento del cursor, enrollamiento, borrar, varios y los que nos permiten visualizar otros menús a partir del actual. Así podemos acceder el menú de ayuda, que a su vez nos permite entrar a un nivel de ayuda más detallado en los siguientes aspectos:

- \* Desplegar y colocar el nivel de ayuda
- \* Reformar párrafos
- \* Explicación de los caracteres "bandera" de la parte de más a la derecha de cada línea de edición
- \* Los Comandos de punto y controles de impresión
- \* El índice de comandos
- \* La línea de Estados
- \* La línea maestra
- \* Los márgenes y tabuladores
- \* Colocar marcas de bloque
- \* Mover parte del texto

La última línea de la pantalla nos permite habilitar ciertos comandos cuando oprimamos las teclas de función. La acción que se va a realizar cuando oprimamos cada una de las teclas es la

siguiente:

- F1 Permite visualizar los diferentes niveles de ayuda.
- F2 Permite recuperar la última cadena de caracteres que fue eliminada
- F3 Marca el texto para que al imprimirse aparezca subrayado.
- F4 Marca el texto para que al imprimirse aparezca remarcado.
- F5 Permite eliminar la línea en la cual se encuentra ubicado el cursor.
- F6 Permite eliminar la palabra en la cual se encuentra ubicado el cursor.
- F7 Permite colocar el cursor en la posición previa que tenía.
- F8 Justificar el texto desde donde se localiza actualmente el cursor y hasta el fin del archivo.
- F9 Permite guardar el documento en el dispositivo de almacenamiento y regresar a edición.
- F10 Permite guardar el documento en el dispositivo de almacenamiento y salir al menú de apertura.

- En el caso de que nos encontremos en una acción en la cual parece imposible recuperarnos, la mejor opción puede ser reinicializar la computadora. Esto es, oprimir las teclas: CTRL - ALT - DEL simultáneamente, y tendremos que reiniciar la sesión. Por ejemplo, que accidentalmente borramos la mitad de nuestro documento.

### 3.3.8. Comandos de edición

La tecla <CONTROL> tiene la característica especial de que cuando se usa en combinación con otras teclas produce una variedad de opciones que le indican a WORDSTAR que realice ciertas funciones especiales como son: borrar caracteres, palabras o líneas, movimiento del cursor, entre otras.

#### Centrar un texto

Para centrar un texto, es necesario teclearlo primeramente, al final de éste y sin bajar al siguiente renglón oprimir "O" (<CONTROL> conjuntamente con la letra O) y con esto entraremos al menú de formatos de pantalla (Onscreen Format). Una vez en este menú teclear C que es la opción para centrar.

#### Remarcar un texto

En general, se remarca un texto dando énfasis a las letras, poniéndolas en negritas (letra tipo Bold), para remarcar un texto es necesario indicar el inicio del remarcado. Esto se hace oprimiendo la tecla <CONTROL> junto con la letra P, teniendo

acceso al menú de Controles de impresión (Print Controls), una vez ahí teclar la letra B de Bold. Después hay que teclar el texto que se desea remarcar, cuando ya no deseamos que el texto salga remarcado hay que volver a teclar CONTROL conjuntamente con P y la B.

#### Subbravar un texto

Al igual que en el remarcado hay que indicar el inicio y el fin del subrayado. Esto se hace con CONTROL, la letra P y la letra S.

### 3.3.9. Guardar o abandonar un documento

Existen varias formas de guardar un documento, la diferencia entre ellas estriba en la acción que se realiza después de guardar el documento. Las posibilidades son:

- \*KS Guarda el archivo y regresa a editar el documento actual
- \*KD Guarda el archivo y regresa al menú principal
- \*KX Guarda el archivo y regresa al sistema operativo
- \*KQ Se ignoran todas las correcciones. Se abandona la sesión sin guardar el documento. Esto pudiera ser deseable si no se desean conservar los cambios hechos a partir de que se guardó el documento por última vez. También se puede declinar la opción de abandono del documento.

### 3.3.10. Modificar un documento

Comúnmente es necesario modificar un texto por equivocaciones, por omisiones o por existir nuevo texto a agregar, todas éstas tareas son permitidas por el procesador.

#### Eliminar texto

En muchos casos en nuestro documento hemos cometido errores. En WORDSTAR es posible eliminar partes del texto de las siguientes formas:

- \*G Elimina el caracter en el cual está posicionado el cursor.
- [Del] Elimina el caracter que se encuentra a la izquierda del cursor.
- \*T Elimina la palabra que se encuentra a la derecha del

cursor.

- ~QY Elimina la parte derecha de la línea a partir de la posición del cursor.
- ~Q[Del] Elimina la parte izquierda de la línea a partir de la posición del cursor.
- ~Y Elimina la línea donde se encuentra el cursor.

### Insertar texto

Cuando iniciamos la edición de un documento, en la línea de estado visualizamos la leyenda "Insert". Esto nos indica que nos encontramos en modo de inserción. Implica que todo lo que nosotros digitemos se insertará a la izquierda de la posición que guarde el cursor.

El modo de inserción se puede deshabilitar por medio de cualquiera de las siguientes opciones:

~V o bien la tecla <Ins>

de este modo colocaremos la pantalla en modo de sobrescritura (overwrite), es decir, que todo lo que nosotros digitemos eliminará todo lo que se encuentre en la posición del cursor, "sobrescribiéndolo."

### 3.3.11. Modificación de la pantalla de edición

A veces es necesario modificar las características de la hoja de edición cambiando los márgenes, el espaciado entre líneas, la colocación de los tabuladores, el número de líneas por hoja, etcétera. Existen dentro del procesador comandos que nos permiten dichos cambios.

#### Eliminación de márgenes

Recordemos que la línea inferior del menú de ayuda, a la cual se la llamó línea maestra, se encuentran en ambos sus extremos las letras L y R; dichas letras nos indican la posición de los márgenes izquierdo y derecho respectivamente. En un momento dado podemos modificar dichos márgenes. El procedimiento para lograrlo es el siguiente:

Margen izquierdo - Utilizando la instrucción ~OL (las teclas de <CONTROL>, O y L) al cual ubicará el margen izquierdo en la columna que nosotros indiquemos.



Margen derecho - Utilizando la instrucción ^OR (las teclas de <CONTROL>, O y R) el cual ubicará el margen derecho en la columna que nosotros indiquemos.

### Volver a dar formato a los párrafos

Una vez que hayamos hecho alguna inserción, o bien, eliminamos una parte de nuestro documento, o redeclaramos la ubicación de los márgenes, es necesario volver a dar formato al párrafo o los párrafos que hayamos modificado, ya que éstos actualmente están como cuando los creamos. Para realizar dicha acción utilizamos el comando: ^B.

La mecánica a seguir es la siguiente: colocamos el cursor al principio del párrafo a modificar, digitamos el comando ^B y automáticamente ajusta el párrafo a las condiciones imperantes.

### Ayuda de guión

Normalmente WORDSTAR tiene habilitada la ayuda de guión, que consiste en que el usuario puede decidir colocar el guión en las palabras largas que no caben en el renglón. Para ello es necesario colocar el guión en el lugar deseado de la palabra, o bien si no se desea el guión se debe digitar nuevamente el comando ^B, con el cual el procesador asume que rechazamos esa ayuda.

### Espacio entre líneas

El comando ^OS nos permite indicarle a WORDSTAR el número de espacios entre líneas. Los valores válidos se encuentran en el rango de 1 a 9.

### Colocación de tabuladores

El comando ^OI permite colocar una parada de tabulador en la columna que nosotros deseamos.

También podemos eliminar una parada de tabulador específica, o bien, todas las que estén marcadas; el comando que nos permite esto es: ^ON.

### Tabuladores decimales

Una facilidad de WORDSTAR, es colocar tabuladores decimales, para lo cual procedemos como cuando colocamos los tabuladores (!), sólo que en esta ocasión en lugar de utilizar signos de admiración ocupamos el símbolo "gato" (#).

### Tabular párrafos

El comando ^OG, nos permite meter temporalmente el margen izquierdo, con el fin de indentar el párrafo en que nos encontremos. El "nuevo" margen izquierdo estará en la columna en la que se haya teclado el ^OG. La línea maestra volverá a su forma original una vez que encuentre <ENTER>.

### **3.3.12. Manejo de bloques en un texto**

Un bloque es un conjunto marcado de líneas de texto. En ocasiones es necesario manejar bloques o secciones de nuestro texto en lugar de hablar de líneas, caracteres o palabras, como hasta ahora lo hemos hecho; en esta sección vamos a tratar de describir los comandos que nos permitirán manejar dichos bloques de texto.

#### Delimitación de un bloque

Esto es limitar el área de texto que será tomada como una sola entidad; los comandos para realizar esta función son los siguientes:

^KB Coloca la marca al principio del bloque

^KK Coloca la marca al fin del bloque

notar que el área bloqueada se visualiza en color inverso. Un bloque permanece definido hasta que se define otro o se quita la marca de bloque.

#### Movimientos de un bloque

Se puede realizar desplazamientos de bloques dentro de nuestro documento:

^KV Mueve el bloque que se encuentra marcado a la posición actual del cursor. Se debe tener cuidado con la posición del cursor y con la forma de trabajo de la pantalla, ya que si colocamos el cursor donde existe material y además estamos en modo de sobrescritura, lo que sucederá es que el material existente desaparecerá, ocupando su lugar el nuevo bloque. Existen límites en la cantidad de material a desplazar, en caso de que se intente mover una gran cantidad de material, WORDSTAR emitirá un mensaje de error; en esos casos debemos dividir el material en dos o tres secciones.

^KC Copia el bloque al lugar donde se ubique el cursor, sin eliminarlo de su área original.

\*KH Deja de mostrar el bloque en video inverso, pero las marcas seguirán existiendo; en caso de querer visualizarlas nuevamente, utilizar este comando otra vez.

### Leer, guardar o eliminar un bloque

Las áreas marcadas como bloques también pueden ser sujetas a guardar; esto es, si una parte de material se utiliza en varios documentos, bueno será que los tengamos en un archivo diferente para llamarlo las ocasiones que tengamos que insertarlo en los demás documentos. A continuación veamos algunos comandos que nos permiten trabajar a ese nivel con los bloques:

- \*KW Sirve para escribir en el disco un bloque de material. WORDSTAR preguntará por el nombre del archivo en donde se guardará el bloque marcado.
- \*KR Leer el Archivo. Este comando nos permitirá traer material que se encuentre en forma de archivo dentro de nuestro disco; Lo utilizaremos cuando queramos incluir material ya una vez guardado. El material se insertará a partir de la ubicación actual del cursor.
- \*KY Este comando se utiliza para eliminar un bloque.
- \*KJ Borra un archivo. Para eliminar los archivos necesarios utilizaremos este comando; aunque recuerde que del menú de apertura también podemos utilizar una opción para realizar la misma función.

### 3.3.13. Búsqueda y reemplazo de texto

Una de las características que hacen muy potente a cualquier procesador de textos es su facilidad para buscar determinada palabra o cadena de caracteres a lo largo de un documento, WORDSTAR permite realizar esta tarea con cierta facilidad. Para ello se describen los siguientes comandos.

#### Búsqueda

- \*QF Permite buscar una cadena de caracteres dentro de nuestro documento. WORDSTAR, al momento de dígitar este comando preguntará por la cadena de caracteres a buscar con Find what? nuestra contestación debe ser la cadena de caracteres que deseamos buscar, la longitud de dicha cadena puede ser de 30 caracteres. Enseguida el procesador nos cuestionará sobre las opciones de búsqueda (Option(s)?). Por ahora responderemos oprimiendo la tecla de RETURN, las opciones las analizaremos más adelante. En este momento WORDSTAR inicia la búsqueda de lo que pedimos, deteniéndose en la primera ocurrencia que

encuentre; una vez ahí nosotros podemos insertar, eliminar o modificar el documento. Si oprimimos ^L el procesador buscará la siguiente ocurrencia de la cadena que buscamos; pero si ya no existen más ocurrencias, entonces mandará un mensaje de que ya no encontró la cadena.

### Búsqueda y reemplazo

**QA** Buscar y sustituir una cadena. Nos permite, como su nombre lo indica, sustituir rápidamente una cadena de caracteres en todo nuestro documento. Después de digitar el comando, nos preguntará la cadena que la sustituirá y por último nos preguntará las opciones. Cuando encuentre la primera ocurrencia nos cuestionará si deseamos o no la sustitución (Replace (Y/N)?). En caso de que queramos buscar las demás ocurrencias, avanzamos con ^L

### Opciones de búsqueda y reemplazo

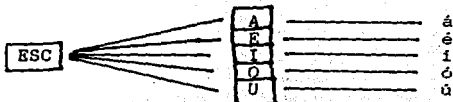
Las opciones que tenemos para los comandos de búsqueda son las siguientes:

- n** Número: Ejecutará el comando el número de veces establecido, esto es buscar las primeras n ocurrencias de la cadena. El valor por omisión es 1.
- g** Global. Busca en todo el archivo a partir del principio. En caso de que el comando utilizado sea reemplazar se detiene en cada ocurrencia para solicitar la confirmación del cambio. Si se usó el comando de encontrar, se detiene en la última ocurrencia. El valor por omisión es a partir del punto en que se encuentra el cursor.
- v** Sustituye la palabra buscada preguntando o pidiendo la confirmación. Si no aparece, que es el valor de omisión, no verifica.
- b** Busca hacia atrás: a partir de la posición del cursor busca moviéndose hacia el principio del archivo. El valor por omisión es hacia el final del archivo.
- u** Ignora la diferencia entre las mayúsculas y las minúsculas. El valor por omisión es que no ignore las diferencias.
- w** Encuentra la cadena completa; esto es si en el documento existen cadenas parecidas pero con otro caracter en los extremos, no las menciona como ocurrencias. El valor por omisión es que reporte subcadenas.

Por último cabe hacer notar que las opciones aquí presentadas se pueden plantear en conjunto, es decir una no excluye a la otra. Ejemplo: **OPTIONS? BOW**: Buscará hacia atrás a partir de la posición del cursor, no distinguirá las letras mayúsculas de las minúsculas, y solamente buscará palabras completas.

### 3.3.14. Colocación de acentos y caracteres especiales

Para acentuar en WS se oprime la tecla <ESC> y la vocal que se desea acentuar:



La tecla <ESC> da acceso a un menú de "atajos" (Shorthand menú) que permite la impresión de otros caracteres especiales como la ñ, Ñ, ¡, ¿, Ü, U, entre otras.

### 3.3.15. Impresión de un documento

Para imprimir un documento podemos seleccionar la opción a partir de dos menús diferentes:

- Del menú de apertura, seleccionando la opción P
  - Del menú principal, oprimiendo el comando "KP".
- WORDSTAR hará una serie de preguntas, que son las siguientes:

- \* **Document to print?** (Documento a imprimir?) Dar el nombre del archivo a imprimir. El valor por omisión es el del último archivo utilizado.
- \* **Number of copies?** (Número de copias?) Número de juegos de copias del documento. El valor por omisión es de 1 pero se puede modificar al valor requerido.
- \* **Pause between pages (Y/N)?** (Pausa entre páginas (S/N)?). Pausa entre impresión de las páginas. El paquete por construcción trae el valor de "No", si se desea mantener este valor basta con oprimir la tecla <ENTER>. Si se desea la pausa habrá que teclear "Y".
- \* **Use forms feeds (Y/N)?** (Uso de formas u hojas sueltas (S/N)?) Por omisión el paquete supone el uso de formas continuas de impresión, por tanto el valor de la respuesta es "N" pero sino se puede modificar con "Y". Si se elige este último valor, la impresora parará al terminar con la

página dando tiempo a que el usuario coloque la siguiente hoja.

- \* **Starting page?** (Página inicial de impresión?) Indicar el número de página a partir de la cual se inicia la impresión. Por omisión el paquete supone el inicio de impresión a partir de la página 1.
- \* **Ending page?** (Página final de impresión?) Indicar el número de página hasta la cual se hará la impresión. Por omisión el paquete supone la impresión hasta encontrar la marca de fin del archivo (EOF).
- \* **Nondocument (Y/N)?** (Archivo no documento (S/N)?) Pregunta si el archivo a imprimir es de tipo documento o no documento. Por omisión supone un archivo de tipo documento y así respetará los formatos de impresión. El archivo de no documento imprimirá los caracteres ASCII sin los formatos de edición como el cambio de página, la numeración, etc.
- \* **Name of printer?** (Nombre de la impresora?) Se indicará el nombre de la impresora para la salida de información, en nuestro caso es IBMPRO o ENTER para que tome el valor por omisión.

#### Funciones de impresión

Para lograr diferentes efectos de impresión podemos utilizar varios comandos. Todos los comandos que mencionaremos tienen como factor común, el hecho de que deben digitarse al principio y al final de la parte del texto en el que se desea tenga dicho efecto.

COMANDO	EFEECTO
^PB	F6 <b>Negritas</b>
^PS	F5 <b>Subraya</b>
^PD	Doble Golpe
^PX	Tachar
^PV	Subíndice ( $L_n$ )
^PT	Exponente ( $a^2$ )

#### Visualizar impresión

Si nuestro documento tiene ya varios efectos de impresión, es difícil imaginarse como se verá ya impreso; para visualizar el texto eliminando los caracteres de control de impresión

utilizaremos el comando: ^OD.

### 3.3.16. Comandos de punto

Los comandos con punto controlan diferentes especificaciones de impresión. Cada uno de estos comandos debe colocarse en una sola línea, el punto debe colocarse en la columna 1. Aquí veremos tres diferente grupos de comandos, que son los siguientes:

#### Márgenes superior e inferior de la página

- .MT Especificación de margen superior. El valor por omisión del margen superior es de 3 líneas, si se desea modificar se utiliza este comando y después de un espacio el número de líneas que se desea especificar. Ejemplo: **.MT 5**
- .MB Especificación de margen inferior. Es la distancia entre el final del texto y el final de la hoja de papel. Su valor por omisión es 10. En caso de desear modificarlo proceder en forma similar a la especificación de margen superior. Ejemplo: **.MB 10**
- .HE Encabezado. Sirve para colocar un encabezado en todas las páginas del documento. Sólo se usa una línea de encabezado. Para especificar un texto, se debe digitar después de un espacio en blanco. Ejemplo: **.HE ESTE ES EL ENCABEZADO**
- .FO Pié de Página. Se utiliza para colocar un pié de página en todas las páginas del documento, se puede usar sólo una línea para especificarlo. Por omisión contiene únicamente el número de página. Ejemplo: **.FO Y ESTE ES EL PIE DE PAGINA**
- .FM Margen del pié de página. Es la distancia existente entre el final del texto y el pié de página. Se usa en forma similar al margen inferior. Ejemplo: **.FM 3**
- .HM Margen de Encabezado. Es la distancia que hay entre el encabezado y el inicio del texto. Su valor por omisión es 2; se especifica en forma similar al margen del pié de página.

#### Paginación

- .OP Omitir el número de páginas. Regularmente WORDSTAR coloca el número de página en todas las páginas del documento. Este comando permite omitir el número de página hasta que se encuentre activado el comando de numeración de página.
- .PN Numeración de páginas. Da valor inicial a la numeración de

páginas, permitiendo la paginación a partir del número que se desee. Ejemplo: .PN 5

- .PC **Columna de colocación del número de página.** Los números de página se encuentran normalmente en la parte inferior de la misma, en la columna 33. Si se desea modificar la ubicación del número se especifica con este comando. Ejemplo: .PC 60
- .PA **Nueva página.** Este comando se utiliza cuando se desea forzar el inicio de una nueva página.

#### Ancho y longitud de impresión

Antes de explicar los comandos de este grupo, daremos a conocer las especificaciones por omisión de impresión:

- El tamaño de la hoja de papel es de 8.5 x 11 pulgadas.
  - Las impresoras imprimen 10 o 12 caracteres por pulgada en sentido horizontal; lo más común es 10; verticalmente se imprimen 8 líneas por pulgada.
  - En la pantalla se trabaja con 80 columnas. Regularmente no todas las columnas se utilizan como texto.
- .LH **Altura de línea.** Permite especificar el número de líneas por pulgada que se imprimen en una página, siendo el valor por omisión de 8. El valor del parámetro se dará en 48-avos de pulgada, así si deseamos que imprima 8 líneas por pulgada debemos dar el número 6, ya que  $6/48 = 1/8$ .
  - .CW **Anchura de carácter.** La unidad base de anchura de carácter es de 1/20 pulgadas. El valor implícito es de 12 puntos ( $12/120 = 1/10$ ) o 10 caracteres por pulgada.
  - .PL **Longitud del Papel.** En caso de tener papel de menor dimensión de 11 pulgadas, o bien si se modifica la altura de la línea se recomienda utilizar este comando que permitirá especificar la longitud de la página. El valor por omisión es de 66.
  - .PO **Desplazamiento de Página.** El valor por omisión es 8, sirve para indentar en el momento de impresión nuestro documento; esto es para evitar que nuestro documento aparezca en el margen izquierdo del papel.

#### Colocación de marcas en el documento

En el documento que contenga varias páginas es útil colocar marcas en intervalos adecuados para que se pueda ubicar rápidamente en cualquier parte del documento. Se pueden colocar



hasta 10 marcas que se identificarán por los dígitos 0 (cero) al 9 (nueve). El comando para colocar las marcas es:

**^K[digito]** En la pantalla aparecerá el símbolo de dígito que utilizamos en el comando.

Para moverse rápidamente a alguna de las marcas utilizaremos el comando:

**^Q[digito]** Con este comando ubicaremos el cursor en el lugar donde hayamos colocado la marca que esté identificada por el dígito que usemos en el comando.

También podemos movernos al principio o fin de un bloque marcado con:

**^QB** Colocarnos en la esquina superior o inicio de un bloque

**^QK** Colocarnos en la esquina inferior o fin de un bloque

independientemente de si el bloque se muestra en video inverso o no.

#### IV. ESTRUCTURA LÓGICA DE UNA COMPUTADORA

## IV. ESTRUCTURA LÓGICA DE UNA COMPUTADORA

Un sistema de computación está conformado por dos estructuras: **software** y **hardware**.

### 4.1. Definición.

El **software** comprende el conjunto de programas y datos necesarios para el funcionamiento del sistema. Dentro de este conjunto de programas se encuentran:

- Sistemas Operativos.
- Ensambladores.
- Compiladores e Intérpretes.
- Paquetes.
- Utilerías.
- Programas de aplicación.

### 4.2. Sistemas numéricos y representación interna de la información.

El sistema numérico de más uso en general es el decimal. Con él se puede representar cualquier cantidad por grande o pequeña que sea, se pueden llevar cuentas de eventos y sucesos, efectuar operaciones aritméticas básicas o calcular expresiones algebraicas complejas.

Existen otros sistemas numéricos con los cuales es posible obtener los mismos resultados, que con el sistema decimal.

La existencia de estos sistemas numéricos, no es una mera curiosidad del campo de las matemáticas. De hecho, las computadoras que se diseñan y fabrican en la actualidad están basadas en el sistema de numeración binaria o sistema de numeración de dos dígitos: el cero y el uno.

En una computadora, la información es representada, almacenada y procesada en forma binaria (combinaciones de unos y ceros).

#### 4.2.1. Sistemas numéricos posicionales.

El sistema numérico decimal utiliza diez dígitos o símbolos que, combinados permiten representar cualquier cantidad.

El dígito 8 cuando ocupa la tercera posición antes del punto decimal representa ochocientas unidades, muy diferentes a las ochenta unidades que representaría si ocupara la segunda

posición. Esta propiedad de asignar valor a los dígitos de acuerdo a su posición, hace de nuestro sistema numérico decimal un sistema posicional.

Dado cualquier número en el sistema decimal, éste podrá ser representado en forma polinomial. Ejemplo:

$$\begin{aligned} 753 &= 700 + 50 + 3 \\ &= (7 \cdot 10^2) + (5 \cdot 10^1) + (3 \cdot 10^0) \\ &= (7 \cdot 10^2) + (5 \cdot 10^1) + (3 \cdot 10^0) \end{aligned}$$

En forma general un número formado de  $n$  dígitos  $(d_n d_{n-1} \dots d_2 d_1)$ , en el sistema numérico decimal tendrá el valor:

$$d_n d_{n-1} \dots d_2 d_1 = (d_n \cdot 10^{n-1}) + (d_{n-1} \cdot 10^{n-2}) + \dots + (d_2 \cdot 10^1) + (d_1 \cdot 10^0)$$

El número diez aparece en cada paréntesis ya que es la base del sistema.

Los sistemas numéricos posicionales ofrecen entre sus muchas ventajas la fácil realización de operaciones aritméticas a través de reglas sencillas, las cuales comienzan por indicar la forma de escribir y colocar los números, efectuar operaciones teniendo cuidado de los "acarreos".

#### 4.2.2. Aritmética base 2.

El sistema binario tiene como elementos al 0 y al 1 y su número base es el 2. Las reglas que rigen la numeración decimal son aplicables al sistema binario.

En este sistema, toda la numeración se genera únicamente a través de las combinaciones de ceros y unos. Así un número escrito en sistema binario también se podrá representar en forma polinomial tomando en cuenta su número base y así conocer su valor en sistema decimal. Ejemplo:

$$\begin{aligned} 43210 & \\ 10100_2 &= (1 \cdot 2^4) + (0 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (0 \cdot 2^0) \end{aligned}$$

Las potencias a las cuales será elevado el número base se numeran a partir de cero el cual será colocado en el primer número a la izquierda del punto decimal: El conteo a partir de cero permite generar la unidad (1) ya que  $n^0 = 1$ , donde  $n$  es cualquier número.

## TABLA DE POTENCIA DE 2

2 <sup>0</sup>	=	1
2 <sup>1</sup>	=	2
2 <sup>2</sup>	=	4
2 <sup>3</sup>	=	8
2 <sup>4</sup>	=	16
2 <sup>5</sup>	=	32
2 <sup>6</sup>	=	64
2 <sup>7</sup>	=	128
2 <sup>8</sup>	=	256

$$\begin{aligned}
 &= (1 * 16) + (0 * 8) + (1 * 4) + (0 * 2) + (0 * 1) \\
 &= 16 + 0 + 4 + 0 + 0 \\
 &= 20
 \end{aligned}$$

Esta será la forma de encontrar el valor en sistema decimal de un número expresado en sistema binario.

El proceso contrario, es decir, pasar un número de decimal a binario, se mostrará resolviendo un ejercicio.

**EJERCICIO:** Pasar el número 25 en base 10 a base 2.

- 1) El número a convertir se divide entre 2, base del sistema (se puede dividir entre la base del sistema al que queramos cambiar, v.gr., dividir entre 8 significa cambiar a base octal). Se considera únicamente la parte entera del resultado, cociente entero. Cuando el número a convertir es par tendremos residuo igual a 0 y en caso contrario, el residuo será 1.

	12	resultado parte entera
2	25	
	5	
	1	residuo

Esto será colocado en una tabla de la siguiente manera:

	COCIENTE ENTERO	RESIDUOS
Cociente entero 1a. div.	25 12	1      Residuo 1a. división

- 2) Se repetirán las divisiones del cociente entero resultante de la división anterior (12) entre el número base (2) mientras el resultado de la división sea mayor de uno, anotando siempre a la parte entera del resultado y los residuos.

	PARTE ENTERA	RESIDUOS	
	25	1	Residuo 1a. división
Parte entera 1a. div.	12	0	Residuo 2a. división
Parte entera 2a. div.	6	0	Residuo 3a. división
Parte entera 3a. div.	3	1	Residuo 4a. división
Parte entera 4a. div.	1		

3) Cuando el resultado de la división es uno, éste resultado se recorrerá a la columna de los residuos.

	PARTE ENTERA	RESIDUOS	
	25	1	Residuo 1a. división
Parte entera 1a. div.	12	0	Residuo 2a. división
Parte entera 2a. div.	6	0	Residuo 3a. división
Parte entera 3a. div.	3	1	Residuo 4a. división
Parte entera 4a. div.	1	1	Se recorre el 1 de la parte entera

4) El número convertido, será la lectura de los residuos de las divisiones leídos de abajo hacia arriba.

PARTE ENTERA	RESIDUOS
25	1
12	0
6	0
3	1
1	1

Por lo tanto  $25_{10}$  en base 2 es igual al número 11001.

### Suma binaria

Siguiendo el método descrito anteriormente se podrá generar la tabla de equivalencia numérica que a continuación se presenta, del sistema decimal y el sistema binario.

**SISTEMA  
DECIMAL**

**SISTEMA  
BINARIO**

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Es fácil observar que entre el número 4 y el número 5 hay una unidad de diferencia; lo mismo sucede entre el número 8 y el número 9, eso quiere decir, que entre dos números consecutivos la distancia o diferencia es la unidad.

Eso querrá decir que si se le suma 1 a un número me dará su consecutivo.

Ejemplo:  $6 + 1 = 7$

7 es el consecutivo de 6.

Ya que nuestra tabla es de equivalencia de sistemas numéricos, será fácil de observar que: Si del lado decimal a 3 se le suma 1 da 4, y por lo tanto querrá decir que si a  $(11)_2$  del sistema binario, 3 del sistema decimal, se le suma 1 da  $(100)_2$ , 4 del sistema decimal, y esto debido a la equivalencia que se presenta de un sistema numérico a otro. por lo tanto, en sistema binario:

$$\begin{aligned} 0_2 + 1_2 &= 1_2 \\ 1_2 + 1_2 &= 10_2 \\ 10_2 + 1_2 &= 11_2 \\ 11_2 + 1_2 &= 100_2 \end{aligned}$$

Es importante tomar en cuenta esto para fines de acarreo.

Ejemplo: Realizar la siguiente suma binaria.

$$(10101)_2 + (11011)_2 + (1010)_2 =$$

1) Es importante alinear los sumandos como se acostumbra en el sistema decimal, pues como ya dijimos, estamos trabajando en un sistema POSICIONAL.

$$\begin{array}{r} 10101_2 \\ + 11011_2 \\ \hline 1010_2 \end{array}$$

2) La suma se efectúa iniciando por el lado derecho y avanzando hacia la izquierda, al igual que en el sistema decimal.

Por lo tanto:

- 1a. columna  $1_2 + 1_2 + 0_2 = 10_2$  colocamos el 0 y "llevamos" 1  
2a. columna  $0_2 + 1_2 + 1_2 = 10_2$  más 1 que "llevamos"  
 $10_2 + 1_2 = 11_2$  colocamos el 1 y "llevamos" 1  
3a. columna  $1_2 + 0_2 + 0_2 = 1_2$  más 1 que "llevamos"  
 $1_2 + 1_2 = 10_2$  colocamos el 0 y "llevamos" 1  
4a. columna  $0_2 + 1_2 + 1_2 = 10_2$  más 1 que "llevamos"  
 $10_2 + 1_2 = 11_2$  colocamos el 1 y "llevamos" 1  
5a. columna  $1_2 + 1_2 = 10_2$  más 1 que "llevamos"  
 $10_2 + 1_2 = 11_2$  el cual se coloca completo ya que  
no hay más cifras a la izquierda.

Por lo tanto:

$$(10101)_2 + (11011)_2 + (1010)_2 = (111010)_2$$

Al convertir cada uno de los sumandos y la suma, se puede observar que la suma de los sumandos convertidos es igual al resultado convertido de la suma.

$$\begin{aligned} 10101_2 &= (1 \cdot 2^4) + (0 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0) \\ &= (1 \cdot 16) + (0 \cdot 8) + (1 \cdot 4) + (0 \cdot 2) + (1 \cdot 1) \\ &= 16 + 0 + 4 + 0 + 1 = 21 \end{aligned}$$

$$\begin{aligned} 11011_2 &= (1 \cdot 2^4) + (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) \\ &= (1 \cdot 16) + (1 \cdot 8) + (0 \cdot 4) + (1 \cdot 2) + (1 \cdot 1) \\ &= 16 + 8 + 0 + 2 + 1 = 27 \end{aligned}$$

$$\begin{aligned} 1010_2 &= (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0) \\ &= (1 \cdot 8) + (0 \cdot 4) + (1 \cdot 2) + (0 \cdot 1) \\ &= 8 + 0 + 2 + 0 = 10 \end{aligned}$$

$$\begin{array}{r} 10101_2 = 21_{10} \\ + 11011_2 = + 27_{10} \\ \hline 1010_2 = 10_{10} \\ 111010_2 = 58_{10} \end{array}$$

$$\begin{aligned} 111010_2 &= (1 \cdot 2^5) + (1 \cdot 2^4) + (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0) \\ &= (1 \cdot 32) + (1 \cdot 16) + (1 \cdot 8) + (0 \cdot 4) + (1 \cdot 2) + (0 \cdot 1) \\ &= 32 + 16 + 8 + 0 + 2 + 0 = 58 \end{aligned}$$

El resultado de la conversión de la suma coincide con la suma de los sumandos convertidos. Por lo tanto, no se pierde exactitud en la suma aritmética de un sistema numérico a otro, cuando se está trabajando con enteros.



## Resta binaria

La resta binaria se efectua de igual modo que una resta en sistema decimal cuidando siempre únicamente las cantidades que "se llevan". Ejemplo:

- Efectuar la siguiente resta binaria.

$$\begin{array}{r} 1101011_2 \\ - 110110_2 \\ \hline 110101 \end{array}$$

- Columna 1 De igual manera que con la resta decimal se inicia de decha a izquierda. Entonces se tiene en ésta columna que uno menos cero es igual a uno.
- Columna 2 Seria uno menos uno es igual a cero.
- Columna 3 Aquí se diria cero menos uno, pero como "no se puede" efectuar se piensa como en la resta decimal, diez (que en el sistema binario es un 2 decimal) menos uno, que en éste caso binario es igual a uno y "llevamos" uno.
- Columna 4 Seria uno menos cero, pero por el "uno que llevábamós" de la columna anterior, éste se suma al sustraendo y entonces queda uno menos uno que es igual a cero y ya "no llevamos nada".
- Columna 5 Aquí, seria como en el caso de la columna 3.
- Columna 6 En esta columna seria uno menos uno, pero cómo "llevábamós uno" de la columna anterior, éste lo sumamos al uno del sustraendo que se convierte en diez y entonces se dice "once menos diez" que es igual a uno y "llevamos uno"; y, finalmente
- Columna 7 Aquí como ya no hay cifra en el sustraendo y supone un cero y más el uno que "llevábamós" queda un uno que restado al uno del minuendo es igual a cero y se termina la resta.

Otro método para efectuar la resta binaria seria el siguiente:

- 1) Alinear el minuendo y el sustraendo como se acostumbra en el sistema decimal.
- 2) Agregar al sustraendo ceros por la izquierda a igualarlo al número de posiciones del minuendo.
- 3) Invertir en el sustraendo los ceros por unos y viceversa, para cada posición.

- 4) Sumar una unidad a éste número invertido.
- 5) Sumar éste nuevo número al minuendo de la resta original.
- 6) Se cuenta de derecha a izquierda en el resultado de ésta suma el número de posiciones con que cuenta el minuendo y ese es el resultado de la resta.

Ejemplo: Efectuar la siguiente resta binaria.

$$\begin{array}{r}
 \text{(1)} \quad - \quad \begin{array}{r} \underline{76543210} \\ 11011010_2 \end{array} \quad (\text{Minuendo}) \\
 \quad \quad \quad \begin{array}{r} \underline{110101} \\ \quad \quad \quad \end{array} \quad (\text{Sustraendo}) \\
 \text{(2)} \quad \quad \quad \underline{00110101} \\
 \text{(3)} \quad \quad \quad 11001010 \\
 \text{(4)} \quad \quad \quad + \quad \quad \quad \begin{array}{r} 1 \\ 11001011 \end{array} \\
 \text{(5)} \quad \quad \quad + \quad \quad \quad \begin{array}{r} \underline{11011010} \\ \underline{110100101} \end{array}
 \end{array}$$

$$R = 10100101_2$$

Transformando el minuendo, el sustraendo y el resultado a sistema decimal, queda:

$$\begin{array}{r}
 \text{Minuendo} \quad = \quad 2+8+16+64+128 \quad = \quad 218 \\
 \text{Sustraendo} \quad = \quad 1+4+16+32 \quad = \quad 53 \\
 \text{Resultado} \quad = \quad 1+4+32+128 \quad = \quad 165
 \end{array}$$

No se pierde exactitud en la resta aritmética de un sistema numérico a otro.

### 4.3. Códigos de comunicación y almacenamiento

El ser humano posee órganos que le permiten comunicarse en forma oral (labios, dientes, lengua, laringe, y el resto del aparato bucal) para transmitir palabras y sonidos; y los oídos para recibirlos. Sin embargo, este tipo de comunicación solo es posible cuando dos personas conocen el mismo lenguaje o la misma manera de codificar la información.

De igual manera que el habla sería imposible sin lenguajes comunes, la comunicación entre computadoras sería imposible sin el acuerdo de códigos comunes, entonces es necesario la existencia de un conjunto de reglas que permitan la conversión de datos de una representación simbólica a otra.

ASCII (se pronuncia ASQUI) es un código moderno de 7 bits.

Con él se representan 128 caracteres. Dentro del conjunto de caracteres que se representan están las letras del alfabeto (mayúsculas y minúsculas), los 10 numerales (0,1,...,9), y los signos de puntuación, además se utilizan códigos de control no imprimibles para regular la operación del dispositivo receptor; por ejemplo, el cambio de línea o alimentación de línea de una impresora. Como hoy en día la unidad para almacenar caracteres es el byte, el código ASCII ha sido adaptado a 8 bits, dando la posibilidad de tener tipos especiales de letra, itálicas o negrillas, o una mayor cantidad de caracteres especiales como (símbolos, recuadros, etcétera).

#### 4.3.1. Lenguajes de programación.

El lenguaje de programación permite especificar instrucciones de computadora. Incluye vocabulario, reglas o convenciones que rigen la forma y secuencia en que se escriben las instrucciones para su ejecución en la computadora; se han ido simplificando de acuerdo con las necesidades del hombre y la importancia que las computadoras han adquirido.

##### 4.3.1.1. Lenguaje de máquina.

El lenguaje de máquina es una representación particular de instrucciones y datos, los cuales son interpretados INMEDIATAMENTE por el hardware de la computadora. Esta representación está dada en bits, por lo que puede presentarse en términos de ceros y unos.

Un programa escrito en lenguaje de máquina no requiere de una traducción para que la computadora lo ejecute, por ello al lenguaje de máquina también se le denomina CODIGO DE MAQUINA.

Para que cualquier programa pueda ser ejecutado por la computadora es necesario que se encuentre escrito en lenguaje de máquina, pues es la única manera en que la computadora puede entender las órdenes que se le dan. Por lo que todo programa elaborado en otro lenguaje de programación debe ser traducido a código de máquina para que pueda ser ejecutado.

##### 4.3.1.2. Lenguaje ensamblador.

A diferencia del lenguaje de máquina, el lenguaje ensamblador consta de un conjunto de instrucciones representadas con caracteres alfanuméricos, que son nemotécnicos de palabras que indican instrucciones lo cual le da una mayor comprensión.

Una instrucción en lenguaje ensamblador se convierte a una o más instrucciones al representarse en código de máquina. Al igual

que el lenguaje de máquina, el lenguaje ensamblador es específico de un modelo dado de computadora.

Para que un programa en lenguaje ensamblador pueda ejecutarse es necesario que se tenga su representación en código de máquina, a través de una traducción. Este proceso lo lleva a cabo el PROGRAMA ENSAMBLADOR.

#### 4.3.1.3. Lenguaje de alto nivel.

El uso del lenguaje ensamblador implica un conocimiento profundo de la máquina con que se esté trabajando. Por ello, el aprender a programar en lenguaje ensamblador resulta complejo, al igual que en el caso del lenguaje de máquina.

Se dice que tanto el lenguaje de máquina como el lenguaje ensamblador son de BAJO NIVEL respecto a la comunicación del hombre con la máquina y a su dependencia con una computadora específica, y a que cada instrucción tiene un "bajo nivel" de información: la traducción de un enunciado en lenguaje ensamblador es casi uno a uno respecto al lenguaje de máquina.

A lo largo del desarrollo de la computación se vió la necesidad de desarrollar otro tipo de lenguaje más natural para expresar soluciones en forma de programas. Aparecieron entonces los lenguajes de ALTO NIVEL, los cuales presentan las siguientes ventajas:

- Fáciles de aprender.
- Permiten elaborar programas fáciles de entender y, por lo tanto, de corregir.
- Son independientes de la computadora. Pueden utilizarse en cualquier máquina, dado su carácter universal.
- Utilizan un conjunto de palabras, generalmente en inglés, que permiten comprender con mayor facilidad y claridad el razonamiento que postula el programador para la solución de su problema.

Algunos Lenguajes de Alto Nivel:

**BASIC.** El lenguaje BASIC (Beginner's All-Purpose Symbolic Instruction Code) fue desarrollado a mediados de la década de los sesentas. El propósito de su elaboración fue el de tener un lenguaje fácil de aprender. Desarrollado inicialmente para estudiantes y poco a poco su campo de acción se ha extendido a aplicaciones administrativas e industriales.

**FORTRAN.** El primer lenguaje de alto nivel ampliamente usado fue FORTRAN (FORMula TRANslation) el cual fue desarrollado a mediados de los cincuenta por la compañía IBM. Fue diseñado para la solución de problemas de naturaleza matemática.

**COBOL.** Diseñado para la elaboración de sistemas administrativos, COBOL (Common Business Oriented Language) fue desarrollado a fines de los 50's por un grupo de usuarios y fabricantes, apoyados por el gobierno de los Estados Unidos.

**ALGOL.** Desarrollado por un comité internacional, también a fines de los años cincuentas, ALGOL (ALGOrithmic Language) se diseñó con el fin de contar con un lenguaje para una fácil traducción de los algoritmos (razonamiento propuesto para la solución de problemas).

**PASCAL.** El lenguaje PASCAL fue desarrollado por Nicklaus Wirth en 1968. Fue diseñado para la enseñanza de la programación como una disciplina sistemática, desarrollado con base en las características de ALGOL, diferenciándose de éste en la facilidad de manejo en los datos, en no ser tan extenso y por ello, de fácil adaptación e implementación. Está estructurado por bloques, siendo cada bloque una rutina independiente. La estructura modular obliga al programador a ser organizado.

Todos estos lenguajes son de propósito general. Ha habido otros de propósito más específico, como LISP (para inteligencia artificial), PROLOG (para programación lógica), SNOBOL (para manejo de cadenas).

#### 4.3.2. Modalidades de traducción.

##### Programa Fuente y Programa Objeto

Se le denomina PROGRAMA OBJETO a aquél que se encuentra en código de máquina y que, por lo tanto, está listo para ser ejecutado.

Se le designa PROGRAMA FUENTE al que está escrito en lenguaje ensamblador o en lenguaje de alto nivel cuyo programa objeto debe ser obtenido antes de ser ejecutado por la computadora.

##### Compiladores e intérpretes

Los compiladores son programas que, teniendo como entrada un programa fuente en lenguaje de alto nivel, producen un programa objeto listo para ser ejecutado.

El trabajo de los compiladores puede dividirse en varias fases:

a) ANALISIS LEXICOGRAFICO - Verificación de que las palabras utilizadas en el programa fuente son válidas en el lenguaje.

b) ANALISIS DE SINTAXIS - Verificación y determinación de la estructura de las frases y organización de cada instrucción del programa fuente.

c) ANALISIS SEMANTICO Y GENERACION DE CODIGO INTERMEDIO - Determinar el "significado" de los enunciados y producción de un conjunto de instrucciones similares a las de un lenguaje ensamblador que corresponden al programa que se está compilando.

d) OPTIMIZACION DE CODIGO - Simplificación del código intermedio obtenido, con el fin de producir un programa objeto más compacto y con uso óptimo de la memoria.

e) GENERACION DE CODIGO DE MAQUINA - Producción final del programa objeto.

Si un programa fuente tiene errores sintácticos, el COMPILADOR los detecta y no se produce el programa objeto correspondiente, hasta que dichos errores sean corregidos por el programador. PASCAL funciona como compilador.

Un INTERPRETE es un programa que traduce y ejecuta una por una las instrucciones de un programa fuente escrito en lenguaje de alto nivel. A diferencia de un COMPILADOR el INTERPRETE no produce un programa objeto, sino que analiza la sintáxis y semántica de cada línea del programa y, si son correctas, las ejecuta, de lo contrario manda un mensaje del error correspondiente y se suspende la ejecución. BASIC originalmente funciona como intérprete.

**CODIFICACION** Es la traducción llevada a cabo por una persona de un conjunto de instrucciones a un lenguaje de programación.

#### 4.3.3. Funcionamiento general de la computadora.

Una vez conocidas las partes que forman una computadora, explicaremos de que manera se integran en la ejecución de las instrucciones y funcionamiento de sus componentes.

1. Es necesario contar con una lista finita de instrucciones donde se indiquen las operaciones necesarias para resolver un problema dado (algoritmo).
2. Una vez definidos los pasos o algoritmo, se debe codificar cada instrucción en términos del lenguaje de programación elegido. Esto en base al conocimiento de la estructura y sintaxis del mismo.
3. Ya con el programa codificado y utilizando el dispositivo de entrada conectado a la computadora, se introduce dicho programa para que sea procesado. Entonces el CPU al recibir la petición de "entrada de

programa" a través de un comando, alista la unidad de memoria para que ahí sean almacenados tanto el programa como los datos. Como en las celdas de memoria únicamente se almacenan valores numéricos, entonces el programa es transformado por orden de la unidad de control y a través de un programa editor residente en la memoria principal, en un código numérico que lo representa plenamente. El programa editor permite a su vez realizar operaciones sobre las instrucciones de memoria, como son: insertar, borrar, modificar, mover, etc. Al programa almacenado en código numérico se le llama programa fuente.

4. El programa fuente es verificado y traducido por el CPU a fin de poder ser ejecutado por la computadora. La verificación de sintaxis del programa fuente es llevado a cabo por un programa especial que está estrechamente relacionado al lenguaje de programación utilizado y cuya tarea será en caso de encontrar errores en la sintaxis en la etapa de verificación, detener la ejecución y suspender la traducción. A ésta nueva traducción se le llama código de máquina ó programa objeto, y al programa encargado de llevar a cabo la verificación y traducción se le conoce como: compilador, intérprete, traductor ó ensamblador. Una vez generado el código objeto, el cual también es almacenado en la memoria, ya es posible ejecutar el programa.
5. En la ejecución del programa, la computadora a través de la unidad de control realizará, en general de manera repetitiva, la siguiente serie de pasos y hasta terminar con la última instrucción del programa.
  - a. Extraer de la memoria la instrucción a ejecutar.
  - b. Decodificar la instrucción leída (entender de que instrucción se trata).
  - c. Ejecutar la instrucción.
  - d. Prepararse para ir a la siguiente instrucción en la memoria y repetir el procedimiento, a menos de que se trate de la instrucción terminar.

#### 4.4. Sistemas operativos.

Un Sistema Operativo es un programa que controla y administra los recursos de la computadora, tales como el procesador, la memoria principal, los dispositivos de almacenamiento, los dispositivos de entrada y salida, así como el manejo de archivos. Un archivo es un conjunto de datos de la misma naturaleza, organizados bajo un esquema bien definido.

Le quita al usuario una gran carga de responsabilidades de programación y de índole operacional, dando margen a un aprovechamiento más eficiente de la computadora.

El sistema operativo consta de un conjunto de programas residentes en el almacenamiento de discos de la computadora, que están diseñados para maximizar y simplificar el uso de los recursos con que ésta cuenta.

De manera general, son cinco módulos los que forman un sistema operativo:

- 1) Administrador de memoria.
- 2) Administrador del procesador.
- 3) Administrador de dispositivos.
- 4) Administrador de información.
- 5) Activador de tareas.

#### Administrador de memoria

Este programa tiene a su cargo las siguientes funciones: llevar un registro de las localidades libres y las localidades ocupadas en la memoria principal de la máquina; determinar a que programa se le permite utilizar la memoria, las localidades que puede usar para almacenar sus datos y en que momento lo hará; y, asignar la memoria a un proceso.

#### Administrador del procesador.

Las funciones de éste son: llevar un registro y control del procesador (o de los procesadores), supervisando el estado del programa que se esté ejecutando, determinar qué programa hará uso del procesador y por cuanto tiempo; asignar el procesador al programa correspondiente; designar el procesador para que lo utilice otro programa.

#### Administrador de dispositivos.

Sus funciones básicas son las siguientes: llevar un control del estado de todos los dispositivos del sistema de computación (impresoras, lectoras, unidades de disco, pantallas, etcétera.); determinar a que programa se le permite uso de un dispositivo específico; asignar físicamente el dispositivo requerido al programa que lo solicita.

#### Administrador de información.

Este módulo realiza las siguientes funciones: llevar un



control de los archivos del sistema (DIRECTORIO); determinar el acceso a los archivos por los programas; asignar el archivo específico al programa que lo va a utilizar.

#### Activador de tareas.

Esta sección se encarga de programar cronológicamente el uso del CPU. En el uso de sistemas de computadoras de proceso único sólo se puede ejecutar una tarea a la vez, pero en los sistemas de tareas múltiples, la computadora parece estar ejecutando en forma simultánea varios procesos o tareas. Esto se hace repartiendo dichas tareas. Aunque se crea o parezca un proceso simultáneo, el procesador ésta alternando a velocidades muy rápidas con las tareas. Así el manejador de tareas determina a que tareas se dedicará el CPU y el orden de trabajo.

#### 4.4.1. Sistema Operativo MS-DOS.

MS-DOS son las siglas de MicroSoft Disk Operating System, diseñado en 1980 y patentado por Microsoft Corporation. Utilizado por computadoras que usan como CPU un chip 8086 u 8088.

MS-DOS trabaja a dos niveles:

1. Sistema de gestión de hardware. Coordinando la CPU de la computadora con el resto del hardware de la misma
2. Función de utilidad. Ejecuta órdenes que permiten interactuarse directamente con la computadora.

Desde que se carga el sistema operativo, éste se guarda en la memoria de acceso aleatorio, RAM. Actúa como intermediario entre el CPU y los programas de aplicación, realizando dos tareas: (a.) localiza el programa a ejecutar, colocándolo en la memoria RAM, y (b.) ayuda a que el programa realice tareas de entrada y salida (INPUT/OUTPUT).

Una vez cargado el sistema operativo MS-DOS, se visualiza una señal, petición de orden o PROMPT en la pantalla que indica que el programa espera se teclee algo.

El prompt de MS-DOS (usualmente A>, B> ó C>) le indica al intérprete de órdenes que lea e identifique el comando que se escribió para MS-DOS, pudiendo ser, encontrar un programa ó instrucción que se desea ejecutar y ejecutarla. Así el sistema operativo localiza el programa correspondiente al comando.

El MS-DOS cuenta con dos caracteres especiales, que se pueden usar en el nombre del archivo, cuando se quieren especificar más de un archivo: el signo de interrogación "?" y el asterisco "\*". Estos caracteres se utilizan como especificadores de archivos cuando se desea que el MS-DOS actúe sobre un grupo de archivos en

vez de hacerlo sobre un archivo individual. Se emparejan los nombres de los archivos letra por letra, y estos caracteres sirven para representar cualquier caracter "?" o cadena de caracteres "\*".

El signo de interrogación (?) se utiliza para sustituir a cualquier caracter en la posición en que se encuentra, dentro del nombre del archivo.

El asterisco "\*" sustituye cualquier número de caracteres en la posición en la que aparece en el nombre del archivo. Sin importar el número de letras.

Por su ubicación en la memoria un comando de MS-DOS puede ser:

**Interno** - Cuando está incorporado al MS-DOS; es decir, se instala en la memoria RAM junto con el MS-DOS.

**Externo** - Es necesario leerlos del disco de MS-DOS antes de que los comandos pueda ser ejecutados.

#### 4.4.1.1. Comandos para el manejo y mantenimiento de discos.

**Comando DIR** - Se utiliza para listar los archivos contenidos en un disco, su tamaño y la fecha de su última actualización.

##### Sintaxis general

DIR [d:] [archiespeci] [/P] [/W]

donde:

d: - Mostrará el directorio del disco ubicado en la unidad de discos d:

archiespeci - Se listarán los archivos que contengan el nombre definido por archiespeci, el cual podrá contener caracteres comodines, interrogación y asterisco.

/P - Listará los archivos contenidos en el disco efectuando una pausa al final de cada página.

/W - Muestra los archivos acomodados en cinco columnas pero no muestra ni el tamaño del archivo, ni la fecha de su última actualización.

**Comando FORMAT** - Prepara un disco (duro o flexible) para ser usado con el MS-DOS.  
(Externo)

### Sintaxis general

**FORMAT [d:] [/S] [/1] [/8] [/V] [/B]**

donde:

**d:** - Indicará la unidad de disco al que se le dará formato, si ésta se omite se toma la unidad de disco por defecto.

**/S** - Las pistas de carga inicial y los archivos del sistema se escriben en el disco formateado.

**/1** - "Formatear" discos para unidades de una sola cara.

**/8** - "Formatear" el disco con 8 sectores por pista.

**/V** - Colocar un rótulo de volumen al disco.

**/B** - "Formatear" el disco con 8 sectores por pista y espacio reservado para archivos ocultos del sistema.

**Comando DISKCOPY** - Copia un disco completo (disco fuente) a otro disco (disco destino). Borrando el contenido previo del disco destino.  
(Externo)

### Sintaxis general

**DISKCOPY [d1:] [d2:] [/1]**

donde:

**d1:** - Indica al disco fuente u origen.

**d2:** - Indica al disco destino.

**/1** - Indica la copia de una sola cara del disco independientemente de si el disco es de una o dos caras.

Otras instrucciones para el manejo y mantenimiento de discos son:

**DISKCOM** - Sirve para comparar dos discos.

**CHKDSK** - Util para examinar y reparar discos.

**RECOVER** - Utilizada para reparar archivos con sectores en mal estado.

**SYS** - Se utiliza para grabar los archivos del sistema y las pistas de carga inicial en un disco.

#### 4.4.1.2. Comandos para el manejo de directorios.

Un subdirectorio es la subdivisión que se hace de un directorio en grupos de archivos con ciertas características.

Comando MKDIR o MD - Se usa para crear un subdirectorio.  
(Interno)

##### Sintaxis general

MKDIR [d:] camino ó MD [d:] camino

donde:

d: - Indicará la unidad de disco donde se creará el nuevo subdirectorio, de no indicarse, se creará en la unidad por defecto.

camino - Indicará el nombre del nuevo subdirectorio así como su ubicación dentro del disco, que puede ser en el directorio general ó raíz o dentro de otro subdirectorio.

Comando CHDIR ó CD - Utilizado para cambiar el subdirectorio actual o para visualizar su nombre.  
(Interno)

##### Sintaxis general

CHDIR [[d:] camino] ó CD [d:]

cambiar el subdirectorio actual dentro de una unidad      visualizar el subdirectorio actual

y donde:

d: - Indicará la unidad de disco sobre la cual trabajará el comando, si no se especifica trabajará en la unidad por defecto.

camino - Ubicación en el disco del subdirectorio al que se quiere cambiar o pasar.

Comando RMDIR ó RD - Utilizado para eliminar un subdirectorio que no contiene archivos de un disco.  
(Interno)

##### Sintaxis general

RMDIR [[d:] camino]

donde:

d: - Unidad de disco donde se encuentra el subdirectorío vacío a eliminar.

camino - Ubicación o ruta en el disco del subdirectorío a eliminar.

Comando **TREE** - Visualiza todas las rutas o caminos de los (Externo) subdirectoríos de un disco.

Sintaxis general

**TREE** [d:] [/F]

donde:

d: - Unidad de disco de la cual se quieren conocer sus rutas o caminos.

/F - Permite el listado de archivos contenidos en un subdirectorío.

#### 4.4.1.3. Comandos para el manejo de archivos.

Comando **ERASE** ó **DEL** - Se usa para borrar un archivo o un grupo de (Interno) archivos de un disco.

Sintaxis general

**ERASE** archiespeci o **DEL** archiespeci

donde:

archiespeci - Se eliminarán los archivos que contengan el nombre definido por archiespeci, el cual podrá contener caracteres comodines.

Comando **RENAME** ó **REN** - Cambia el nombre de un archivo sin cambiar (Interno) su contenido.

Sintaxis general

**RENAME** archiespeci1 archiespeci2

donde:

archiespeci1 - Es el nombre del archivo que se quiere cambiar.

archiespeci2 - Es el nuevo nombre que se dará al

archivo.

Comando COPY - Permite copiar un archivo o varios archivos con las mismas características de nombre y/o extensión a otro disco. Donde la copia podrá conservar o cambiar el nombre y/o extensión de archivo.

Sintaxis general

COPY [/A/B] arch1.ext1 [/A] [/B] [d:] [arch2.ext2] [/A] [/B]

donde:

d: - Unidad de disco en la cual se realizará la copia.

arch1 - Es el nombre del archivo fuente u origen.

ext1 - Es la extensión del archivo fuente u origen.

arch2 - Es el nombre del archivo destino.

ext2 - Es la extensión del archivo destino.

4.4.1.4. Comandos de configuración y operatividad.

Comando CLS - Borra la pantalla y visualiza el prompt del MS-DOS.  
(Interno)

Sintaxis general

CLS

Comando DATE - Pone la fecha que el MS-DOS utiliza cuando actualiza archivos.

Sintaxis general

DATE [mm-dd-aa]

donde:

mm-dd-aa - Corresponden al mes, día y año respectivamente, se pueden separar los datos con guiones (-) o diagonal (/).

Comando TIME - Pone la hora que el MS-DOS utiliza cuando actualiza archivos.

Sintaxis general

**TIME** [hh:mm:ss.ss]

donde:

hh:mm:ss.xx - Corresponden a las horas, minutos y segundos, se deben utilizar los dos puntos (:) para separar las cantidades.

Comando **PROMPT** - Sustituye la petición de orden del MS-DOS por cualquier otra cadena de caracteres. la petición de orden común es la letra de la unidad por defecto, seguida por un signo de mayor que (>).

(Interno)

Sintaxis general

**PROMPT** [cadena-petición-orden]

donde:

cadena-petición-orden - Corresponden a una cadena de caracteres que contiene la información que se quiere que aparezca en la petición de orden del MS-DOS.

## V. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS Y PROGRAMACIÓN



## V. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS Y PROGRAMACIÓN

El propósito de la presente unidad es que el estudiante aprenda un método para solucionar problemas y que posteriormente puedan ser resueltos haciendo uso de la computadora.

### **5.1. Método de solución de problemas.**

Se cuenta con un problema en general, es cuando lo que se "quiere" es distinto a lo que se "tiene" y en particular, es una cuestión a resolver.

La técnica de solución de problemas permite en muchos casos proponer la metodología necesaria para solucionar un problema. Ello requiere estudio y práctica, lo cual necesita creatividad, imaginación y habilidad de cada persona. Por ello, la solución que se dé a un problema planteado dependerá de la persona que desarrolle el método de solución. Y aunque el resultado deba ser el mismo, la forma de solución variará de persona a persona. Por lo tanto, deberá quedar claro que el método de solución a un problema no es único.

Para nuestro propósito se distinguirán dos tipos de problemas: computables y no computables.

En particular nos enfocaremos a los problemas computables o aquellos en que se puede hacer uso de la computadora y que deben contar con las siguientes características:

- Se puede reducir su tiempo de solución considerablemente.
- Problemas que se presentan repetidamente (recurrentes) con pequeñas modificaciones o distintos datos.
- Requieren una solución precisa.
- Problemas perfectamente definidos, en el que se elige entre varios métodos de solución, el más conocido por la persona o el que considere más adecuado.

### Etapas de solución de problemas

Se proponen cuatro etapas para la solución de problemas:

1. Definición del problema.
2. Análisis del problema.
3. Diseño de solución del problema.
4. Desarrollo de la solución del problema.

1. Definición del problema. En esta etapa se debe definir y entender el problema, reconocer las necesidades y definir los objetivos a alcanzar con la solución del problema; todo ello por

escrito en lenguaje claro y preciso, evitar los tecnicismos. Nunca pensar que las soluciones son obvias. Definir y comprender perfectamente el problema que se presenta, concediendo para ello el tiempo necesario, reconociendo primeramente cuál es el problema y luego cómo solucionarlo.

Modificar o reescribir la definición del problema las veces necesarias con el fin de lograr la mayor claridad posible. Al definir un problema se deben de considerar:

- a. **Los datos de entrada o información inicial.** Saber que datos se necesitan y en que momento serán utilizados. A estos se les denomina parámetros del problema.
- b. **Procesamiento.** Es la manipulación o transformación de los datos iniciales en entidades significativas y resolutivas del problema planteado.
- c. **Información de salida.** Es traspasar la información generada en el procesamiento a un almacenamiento externo o cualquier dispositivo de salida. La información se presentará a través de resúmenes, listas, reportes o archivos.
- d. **Control y retroalimentación.** Verificar el buen manejo de la información tratando de eliminar los errores, y así, la información o datos generados sean confiables y puedan volver a ser utilizados.

2. **Análisis del problema.** Es la etapa más importante, en ella debe darse una excelente comunicación entre quien elabora o realiza el programa y quien lo utilizará.

Se debe iniciar dividiendo el problema en partes. Cada parte deberá tener una función o actividad perfectamente definida y delimitada de cada parte del todo.

Se pretende definir en un lenguaje claro, el conjunto de actividades necesarias para que el programa en su totalidad funcione adecuadamente.

3. **Diseño de solución del problema.** En esta etapa se intenta describir una solución aceptable al problema en forma sistemática, sin dejar brechas en la lógica.

#### Características del diseño.

- a. **Concepto general de diseño.** Se debe proporcionar un panorama general de todo el programa en una combinación de textos y diagramas.
- b. **Normas y estándares.** Debe incluir normas de codificación y estándares que se deben observar durante la fase de programación.
- c. **Diseño del programa.** Descripción de la estructura real del programa, jerarquización y descomposición del mismo en fracciones más pequeñas.

- d. Diseño de archivos. Definir detalladamente los archivos a ser utilizados.
- e. Flujo de datos. Indica la ruta que seguirán los datos durante el proceso.

#### Guías de diseño.

- a. Modularidad. Implica la división del trabajo tomando en cuenta las tareas que se realizarán, tratando de que éstas puedan ser tratadas como procesos independientes entre sí.
- b. Definición de interfases. Consiste en definir y documentar las formas de comunicación entre los diferentes módulos del programa.
- c. Simplicidad. Expresar el diseño en un lenguaje claro, sencillo y comprensible.
- d. Regla de las líneas negras. Definir las entradas y salidas sin tomar en cuenta la estructura interna del programa.

#### Herramientas de diseño.

Es necesario auxiliarse de las herramientas de diseño, el uso de ellas, dependerá de la complejidad del problema a resolver.

- a. Algoritmos. Secuencia ordenada y finita de pasos, exenta de ambigüedades, que lleva a la solución de un problema.
- b. Diagramas de flujo. Es la representación gráfica de la lógica y secuencia de un algoritmo, proporcionando un lenguaje común entre la gente dedicada a la computación. La representación gráfica se realiza haciendo uso de figuras geométricas conectadas por líneas, cada figura representa una etapa en la solución del problema, ejecutándose en forma secuencial, de tal manera que no se pueden procesar dos etapas al mismo tiempo, así las figuras están conectadas entre sí con líneas que tienen una flecha que indica la dirección o sentido en que los procesos deben ejecutarse. Son herramientas útiles y poderosas que constituyen un medio efectivo de comunicación compensando las limitaciones de los textos escritos, su claridad no deberá requerir de una explicación complementaria. Tienen las siguientes características:
  - Son universales. Permiten la comunicación entre los programadores de computadoras.
  - Son gráficos. Son la representación en símbolos de un algoritmo.
  - Emplea un conjunto de símbolos normalizados. Sus símbolos y función son conocidos universalmente.
  - La estructura del diagrama no depende de la cantidad de datos. Los diagramas de flujo sirven para un número indeterminado de datos.
  - Debe ser lo más general posible. Se debe pensar en su funcionalidad para un campo amplio.
  - Son independientes del lenguaje de programación.

## Simbología

Es la serie de figuras geométricas utilizadas y la definición de la función o tarea que representan. A continuación se presentan algunas de ellas:

# SIMBOLOGIA DE DIAGRAMAS DE FLUJO



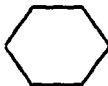
Inicio o fin de un  
diagrama de flujo



Desplegado



Entrada o salida  
de dispositivos



Ciclo iterativo



Tarjeta perforada



Liga de comunicacion



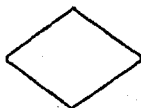
Cinta magnetica



Conector de pagina



Cinta de papel  
perforada



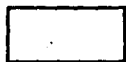
Decision



Impresion en papel



Entrada optica



Proceso



Conector de rutina

4. **Desarrollo de solución del problema.** Una vez obtenido el mejor método de solución del problema, se debe traducir el programa a un lenguaje de programación, tomando en cuenta para ello la codificación y las pruebas al programa.

**Codificación o programación.** Con el diagrama de flujo listo, el programador elegirá de acuerdo a sus conocimientos y experiencia el lenguaje de programación más apropiado para la traducción y obtención del programa para computadora, respetando la lógica exacta y especificando las operaciones lógicas y aritméticas que debe realizar el programa durante el proceso.

Se debe pretender finalmente la obtención de programas correctos, confiables, fáciles de mantener, modificables, legibles y comprensibles.

**Prueba del programa.** Una vez codificado el programa se procederá a compilarlo y ejecutarlo, aquí el programador deberá verificar que el programa realmente hace lo que se requiere; acepte los datos necesarios, y proporcione finalmente la información deseada. En caso de que esto no suceda se procederá a su revisión, corrección y prueba hasta obtener la solución requerida.

## 5.2. Fundamentos de programación en un lenguaje estructurado

### 5.2.1. Antecedentes.

El lenguaje PASCAL, fue desarrollado en Zurich (Suiza) por Niklaus Wirth en 1968, y debe su nombre al filósofo y matemático francés Blaise Pascal.

Diseñado a ser una herramienta de enseñanza para los conceptos de programación. Por ser un lenguaje altamente estructurado y modular se adapta a las modernas técnicas de programación. Su forma simple y legible lo hacen uno de los mejores lenguajes para enseñar los conceptos de programación y adquirir buenos hábitos en esta disciplina.

### 5.2.2. Elementos del lenguaje PASCAL.

Siendo uno de los lenguajes más utilizados en la enseñanza de la programación cuenta con la ventaja de ser un lenguaje de "formato", esto es, que la instrucción puede aparecer en cualquier columna, en varias líneas; o, más de una instrucción en una sola línea. Para ello cuenta con algunos elementos.

### 5.2.2.1. Identificador.

Es una sucesión de letras, números y el caracter de subrayado, bajo la cual el programador identifica constantes, tipos de variables, rutinas, funciones y el nombre mismo del programa. El identificador debe comenzar forzosamente con una letra, y puede estar constituido con cualquier combinación de letras y números. Un identificador puede ser el nombre de una variable, posición de memoria cuyo contenido cambia o varía, o de una constante posición de memoria cuyo contenido nunca cambia.

### 5.2.2.2. Separador de instrucciones.

El compilador del lenguaje de Pascal hace un análisis de la sintaxis y semántica del programa, checando que no existan errores e indicando, si es que los hay, la naturaleza de éstos y su posición en el programa. Esto se hace instrucción por instrucción, por ello, es necesario indicar de alguna manera el término de éstas. Para ello se coloca un punto y coma (;) para separar cada instrucción o grupo de instrucciones, de las siguientes. Un grupo de instrucciones que se desea funcionen juntas se delimitan por las palabras **BEGIN** y **END**.

### 5.2.2.3. Comentarios

Los comentarios se utilizan para indicar:

- El nombre y los datos personales del programador.
- La función que realiza cada módulo.
- La fecha en que se elaboró el programa, así como la fecha de la última versión.
- El objetivo de algún módulo o de algún proceso.
- Explicar pasos del programa que no se expliquen a si mismos.

En Pascal, un comentario es cualquier mensaje del tamaño que se desee, delimitado por los símbolos:

(\*                    \*)                    o                    {                    }

que no son intercambiables como parejas.

### 5.2.2.4. Tipos simples

El conjunto de valores que pueden tomar están ordenados linealmente. Los tipos escalares son cuatro:

	Enteros (INTEGER)
Númericos	
	Reales (REAL)
Caracteres	(CHAR)
Lógicos	(BOOLEAN)

### 5.2.3. Partes de un programa en PASCAL.

Todo programa consta de tres partes:

- Encabezado
- Sección de Declaraciones
- Cuerpo

#### 5.2.3.1. Encabezado.

A través del encabezado se indica:

- Que el siguiente conjunto de instrucciones se refieren a un programa en Pascal.
- El nombre del programa.
- Los archivos a los cuales el programa hará referencia.

La sintaxis es:

```
PROGRAM <nombre del programa> (<lista de archivos>);
```

PROGRAM : Indica que se trata de un programa en PASCAL.

<nombre del programa>: Es un identificador escogido por el programador. Se aconseja que el programa tenga un nombre acorde al proceso que efectuará.

<lista de archivos>: Son los nombres de los archivos que va a utilizar el programa, separados por comas.

Debemos hacer notar que un archivo tiene dos nombres:

- Nombre Lógico
- Nombre Físico

El nombre lógico es aquel al cual hacemos referencia en nuestro programa. El nombre físico, del mismo archivo, es el que lo identifica en el dispositivo de almacenamiento.

Un archivo puede tener varios nombres lógicos, tantos como programas hagan uso de él, pero sólo tiene un nombre físico, que es el que lo identifica de todos los demás contenidos en el dispositivo.

Existen dos archivos para texto, que son referidos automáticamente a la entrada y salida estándar de datos en la computadora que se esté utilizando. Son:

- INPUT para la entrada y que físicamente corresponde al teclado.
- OUTPUT para la salida y que físicamente corresponde a la pantalla.



### 5.2.3.2. Sección de declaraciones.

En ella se definirán y declararán las etiquetas, constantes, tipos, variables, procedimientos y/o funciones. Cada una de estas declaraciones son indicadas con una palabra reservada específica. Estas son, en orden:

- Etiquetas (LABEL)
- Constantes (CONST)
- Tipos (TYPE)
- Variables (VAR)
- Procedimientos (PROCEDURE)
- Funciones (FUNCTION)

Un programa puede o no tener cualquiera de estas declaraciones. Puede tener todas, algunas o ninguna, según lo requiera. En las nuevas versiones de TurboPascal ya no es necesario respetar el orden, pero resulta conveniente conservar este hábito.

#### 5.2.3.2.1. Etiquetas.

La sintaxis es:

```
LABEL <lista de etiquetas>;
```

Donde:

<lista de etiquetas> es una serie de uno o más números enteros, separados por comas. Las etiquetas serán números enteros positivos no mayores a 9999. En la mayoría de las versiones de compiladores del lenguaje Pascal las etiquetas ya son consideradas identificadores.

Una etiqueta indica una posición en el programa, a la cual puede mandarse la ejecución del programa a través de la instrucción GO TO:

```
GO TO <etiqueta>.
```

Al ejecutarse la instrucción GO TO, el programa irá a ejecutar, a continuación, la instrucción indicada por la etiqueta correspondiente y continuará su ejecución partiendo de ese lugar en el programa.

#### 5.2.3.2.2. Constantes

La sintaxis es:

```
CONST  
  <identificador> = <valor de la constante>;
```

Donde:

<identificador> lo determina el programador.  
<valor de la constante> es el valor asignado al identificador, y puede ser un: número entero, número real, valor lógico, caracter, cadena de caracteres. Una posición de memoria, referenciada por un nombre de constante (identificador), donde se almacena un valor (este valor no puede cambiarse). Las definiciones de constantes (si las hay) van antes de la declaración de variable.  
Pueden declararse varias constantes separándolas con ";".

A lo largo del programa estas constantes no podrán cambiar su valor. No se pueden usar como variables. Las variables permiten modificar sus valores; las constantes NO.

#### 5.2.3.2.3. Tipos

La sintaxis es:

TYPE

<tipo definido> = <estructura>;

Donde:

<tipo definido> será cualquier identificador.  
<estructura> será la especificación de algún tipo simple o de un tipo estructurado.

Es muy importante no confundir un tipo con una variable. Un tipo NO puede tomar valores; una variable SI

#### 5.2.3.2.4. Variables

La sintaxis es:

VAR

<lista de variables> :<tipo>;

<lista de variables> es un conjunto de uno o más identificadores separados por comas.  
<tipo> es un tipo primitivo o definido por el programador.

Las variables son definidas indicando cuál es el tipo de valores que podrán tomar. Puede observarse que es válido tener más de una variable asociada a un tipo. Es una posición de memoria, referenciada por un nombre de variable (identificador), donde se puede almacenar un valor del dato, este valor puede cambiarse.

### 5.2.3.2.5. Procedimientos

Las rutinas son subprogramas de los cuales consta el programa mismo. Cada una de ellas tendrá su encabezado, su sección de declaraciones y su cuerpo. Un procedimiento se define como un encabezado y un bloque. La sintaxis general es:

```
PROCEDURE <identificador>(<parámetros>);
```

A un procedimiento se le llama (o se le invoca) usando el identificador del procedimiento como una sentencia en el programa principal o en el cuerpo de otro procedimiento, con los parámetros que se van a usar entre paréntesis a continuación del nombre. Es importante señalar que el uso de parámetros no es obligatorio.

### 5.2.3.2.6. Funciones

Son procesos programados que dan como resultado un valor, dependiendo de los parámetros que se le proporcionen. Existen las funciones intrínsecas o estándar y las elaboradas por el programador, que son también subprogramas.

```
FUNCTION <identificador>(<lista de parámetros>):<tipo>;
```

Observar que el identificador usado como nombre de función tiene dos papeles. Identifica a la función que puede así llamarse, y actúa como variable a través del cual se devuelve un valor a quien hizo la llamada.

### 5.2.3.2.7. Criterios para el uso de Procedimientos y Funciones

1. Si entrega más de un resultado del subprograma, no usar función.
2. Si sólo hay una salida del subprograma y ese valor se va a usar inmediatamente, se recomienda usar una función.
3. Cuando haya duda, usar un procedimiento. Cualquier función puede redefinirse como un procedimiento en el que el nombre de la función sería uno de los parámetros que se utiliza para entregar resultados.
4. Si ambas cosas son aceptables, use aquélla con la que se sienta más seguro.

La salida de un subprograma se refiere al punto en que la ejecución regresa se dicho subprograma al programa principal.

PASCAL permite que una función o procedimiento se llame a sí mismo. A esta propiedad se le llama recursividad. La recursividad es una poderosa característica de PASCAL.

### 5.2.3.3. Cuerpo del programa

Es el conjunto de instrucciones que indican el procedimiento o camino a seguir. Este conjunto de instrucciones es delimitado por las palabras **BEGIN** (inicio) y **END** (fin).

La sintaxis es:

**BEGIN**

-----  
-----  
<instrucciones>  
-----  
-----

**END;**

Sólo en el caso del cuerpo del programa principal, el **END** tendrá asociado un punto ".", y no un punto y coma ";":

**BEGIN**

-----  
-----  
<instrucciones del programa principal>  
-----  
-----

**END.**

### 5.2.4. Programación modular.

Al proceso de descomponer un programa en varias partes de acuerdo a los objetivos de cada parte se le llama **PROGRAMACION MODULAR**. A cada una de estas partes las llamaremos módulos.

#### 5.2.4.1. Metodología de programación.

El tener codificado el programa en pequeñas rutinas o módulos nos brinda la posibilidad de localizar y resolver los cambios que se tengan que hacer con mayor eficiencia y rapidez. Cada módulo será una rutina o función que realiza un trabajo específico.

Los módulos deben tener las siguientes características:

1. Deben poder ser llamados o invocados por su nombre.
2. El control del programa debe regresar a donde fue llamado el módulo, al terminar la ejecución de este último.
3. Deben poder llamar o invocar a otro módulo, si es necesario.
4. Deben contener **UNICAMENTE** un punto de entrada y un punto de salida, en lo que se refiere a su ejecución.

5. Debe ser de un tamaño relativamente pequeño.
6. Debe desempeñar una sola función.
7. Deben ser independientes de otros módulos y del sistema mismo.
8. Debe ser fácil su eliminación o inserción en otros sistemas.

#### 5.2.4.2. Criterios de Diseño.

Estos diseños permiten la organización del trabajo a efectuar en la programación y al programador le proporcionan una visión más exacta de la tarea que realiza cada módulo, su conexión con otros módulos y con el programa principal.

**Diseño descendente (TOP-DOWN)** .- Es un criterio de diseño que consiste en ir programando de lo general a lo particular, es decir, de los módulos principales a los módulos que son llamados por ellos. Esta técnica es la que se debe utilizar con lenguajes imperativos como PASCAL.

**Diseño ascendente (BOTTOM-UP)** .- La metodología de diseño bottom up, implica comenzar de lo particular a lo general. Se debe ir diseñando y programando los módulos por separado y posteriormente, ir agrupándolos hasta conseguir la integración total del sistema. Esta técnica se utiliza mucho con lenguajes de los llamados "orientados a objetos".

#### 5.2.4.3. Rutinas (Procedures).

Las rutinas se encuentran casi al final de la zona de declaraciones de los módulos, inmediatamente después de ser declaradas las variables (VAR). Una rutina consta, como todo subprograma de:

- Encabezado
- Sección de declaraciones
- Cuerpo del subprograma

#### Variables globales, locales y parámetros

Las variables globales, que son declaradas en el VAR del programa principal, se encuentran presentes a lo largo de TODO EL PROGRAMA. Se puede hacer referencia a ellas en cualquier momento y dentro de cualquier subprograma.

Las variables locales son declaradas en el subprograma, y su dominio de acción es SOLO DENTRO DEL MISMO SUBPROGRAMA.

Las variables manejadas como parámetros permiten el FLUJO CONTROLADO de la información, de tal manera que se indican:

- Las variables que deben entrar de la parte externa hacia el subprograma, para que éste se ejecute de acuerdo a esos valores se les denomina PARAMETROS DE ENTRADA.
- Las variables que deben de salir del subprograma hacia el resto del sistema, para que éste siga ejecutándose tomando en cuenta las modificaciones que hizo el procedure a las variables. Se le denomina PARAMETROS DE SALIDA.

Los parámetros son la conexión del módulo con el resto del sistema y se indican en el encabezado del subprograma.

La sintaxis es:

```
PROCEDURE <identificador> (<lista de parámetros>);
```

Donde:

<identificador> Es el nombre que se le asignará a la rutina.  
<lista de parámetros> Es la definición de todos y cada uno de los parámetros con los que se comunica la rutina al sistema. Son opcionales, de usarse consta del nombre de las variables (parámetros) y del tipo de cada uno de ellas.

La sintaxis de la lista de parámetros es:

```
indicador de variación lista de variables : tipo ;
```

Donde:

indicador de variación es la palabra VAR que señala si el parámetro es de salida (ó entrada y salida), es opcional y la ausencia de la palabra VAR indica que el parámetro es únicamente de entrada. De acuerdo a los propósitos que se requieran:

-Si una lista de variables asociadas a un tipo tienen este indicador de variación, entonces el valor de las variables será modificado por el subprograma. Se dice entonces que el parámetro es pasado por referencia.

-De lo contrario, el valor de las variables, después de la ejecución del subprograma, será el mismo que tenían antes de la ejecución de éste. Se dice que el parámetro es pasado por valor.

Para que sean ejecutadas las rutinas, basta con que sean llamadas dentro del cuerpo de un módulo, siempre que estas rutinas estén dentro de la zona de declaraciones del mismo módulo.

En el paso de los parámetros, nosotros le indicamos a la máquina qué variables se van a transmitir de un módulo a otro para ser utilizados y/o modificadas. Además, tenemos la ventaja de indicar qué variables deseamos que cambien su valor y cuáles no.

#### 5.2.4.4. Funciones

Al igual que en las rutinas, las funciones constan de:

- Encabezado
- Sección de declaraciones
- Cuerpo

A diferencia de las rutinas, las funciones tienen asignado un valor por sí mismas. El uso de parámetros, al igual que en los procedimientos, es opcional.

La sintaxis es:

```
FUNCTION <identificador> (<lista de parámetros>) : <tipo> ;
```

Donde:

<lista de parámetros> Es la misma sintaxis que para las rutinas; con los parámetros asociados a su tipo respectivo. el manejo de parámetros será igual que en los procedimientos. <tipo>. Es cualquier tipo ya definido. Indica el tipo de valor que la función va a proporcionar.

Es importante recalcar que la función proporciona un valor bajo su mismo nombre. Por ello, dentro del cuerpo de la función, se deberá tener a la expresión correspondiente asignada al identificador de la función.

#### 5.2.5. Operadores y funciones estándar

##### 5.2.5.1. Operadores

Nos permiten efectuar operaciones al elaborar expresiones relacionando con ellos variables, constantes y funciones. Los cuatros grupos de operadores básicos son:

- De asignación.
- Aritméticos.
- Lógicos.
- De comparación.

#### 5.2.5.1.1. Operador de Asignación.

La instrucción fundamental, es la asignación. A través de instrucciones de asignación, el programador guarda el valor de una constante o de una expresión en una variable. La sintaxis es:

< variable > := < expresión > ;

Donde:

< variable > - Es cualquier identificador asociado a algún tipo.  
:= - Es el operador de asignación.  
< expresión > - Es una o más variables, constantes, funciones o expresiones, relacionadas a través de operadores.

#### 5.2.5.1.2. Operadores aritméticos.

<u>Operador</u>	<u>Operación</u>	<u>Tipos de los operandos</u>	<u>Tipo del resultado</u>
+	Adición	Entero, entero Entero, real Real, real	ENTERO REAL REAL
-	Substracción	Entero, entero Entero, real Real, real	ENTERO REAL REAL
*	Multiplicación	Entero, entero Entero, Real Real, real	ENTERO REAL REAL
/	División	Entero, entero Entero, real Real, real	REAL REAL REAL
DIV	División entera con resultado entero (truncado)	Entero, entero	ENTERO
MOD	Módulo (residuo de división entera)	Entero, entero	ENTERO

#### 5.2.5.1.3. Operadores de Comparación.

Dos elementos o datos pueden compararse. El resultado de tal comparación será un valor lógico o valor de verdad: verdadero (TRUE) ó falso (FALSE).



Las comparaciones pueden hacerse entre variables, constantes, expresiones y funciones que entreguen valores simples, siempre y cuando correspondan las expresiones que se están comparando al mismo tipo.

Llamaremos **CONDICION** a:

- Una variable lógica.
- La relación entre dos expresiones, variables o constantes a través de este tipo de operadores.

Una **CONDICION** es utilizada:

- En las figuras lógicas de programación.
- Para asignar un valor a una variable lógica.

<u>Operador</u>	<u>Operación</u>	<u>Tipos de los operandos</u>	<u>Tipo del resultado</u>
=	Igual	Cualquiera	BOOLEANO
<>	Diferente	Cualquiera	BOOLEANO
<	Menor a	Cualquiera	BOOLEANO
>	Mayor a	Cualquiera	BOOLEANO
<=	Menor o igual a	Cualquiera	BOOLEANO
>=	Mayor o igual a	Cualquiera	BOOLEANO

#### 5.2.5.1.4. Operadores booleanos.

Se utilizan para relacionar dos o más **CONDICIONES**, construyendo una **EXPRESION CONDICIONAL**. los operadores booleanos son:

NOT Negación "no"  
 AND Conjunción "y"  
 OR Disyuntiva "o"

Las tablas de verdad determinan el posible resultado de una expresión condicional.

- Tabla de verdad del **NOT**

NOT FALSE ..... TRUE  
 NOT TRUE ..... FALSE

- Tabla de verdad del AND

FALSE AND FALSE	.....	FALSE
FALSE AND TRUE	.....	FALSE
TRUE AND FALSE	.....	FALSE
TRUE AND TRUE	.....	TRUE

- Tabla de verdad del OR

FALSE OR FALSE	.....	FALSE
FALSE OR TRUE	.....	TRUE
TRUE OR FALSE	.....	TRUE
TRUE OR TRUE	.....	TRUE

### 5.2.5.2. Funciones estándar

Las funciones estándar, también llamadas intrínsecas, están implementadas en el lenguaje. El programador no necesita elaborarlas, sino simplemente hacer uso de ellas.

<u>Función</u>	<u>Operación</u>
ABS(X)	Función valor absoluto de X. Opera con valores enteros y reales, devolviendo un resultado entero y real respectivamente.
ARCTAN(X)	Angulo en radianes cuya tangente es X. Opera con valores enteros y reales, devolviendo un resultado real.
CHR(X)	Da el caracter correspondiente a la posición X en el código ASCII. Opera con valores enteros y devuelve un resultado de tipo caracter.
COS(X)	Coseno de X radianes. Opera con valores enteros y reales, devolviendo un resultado real.
EOF(A) o EOF	Indica si el apuntador del archivo A se encuentra en el final de éste.

### 5.2.5.3. Precedencia

La elaboración de las operaciones, en PASCAL y en los demás lenguajes, sigue ciertas normas de orden, ya que existen operaciones que se ejecutan primero en una expresión y otras que se ejecutan posteriormente. Se dice entonces que existen operadores con mayor precedencia que otros.

**Nivel de precedencia****Operadores**

Primero			NOT	
Segundo	*	/	AND DIV	MOD
Tercero			OR	
			+ -	
Cuarto	<	>	<>	=
			<= >=	

La ejecución de las operaciones, en una expresión, se lleva a cabo comenzando con las operaciones de niveles superiores y terminando con las operaciones de niveles inferiores.

Por otro lado, los PARENTESIS, (), rompen con estas normas de precedencia.

Frente a operadores de la misma precedencia, el orden es de izquierda a derecha.

**5.2.6. Rutinas de entrada y salida**

Para PASCAL, la entrada y salida de datos siempre se hace a través de archivos. Los archivos pueden encontrarse almacenados en un disco o en una cinta, pero también el teclado, la pantalla, la lectora de tarjetas y a la impresora, PASCAL los considera como archivos dado que, gracias a ellos, se obtiene y se deposita información (se interacciona con el "exterior").

**5.2.6.1. Rutinas de entrada****5.2.6.1.1. Rutina READ**

Es utilizada para leer valores y asignarlos a las variables. Las variables pueden ser de tipo: INTEGER, REAL, CHAR. La sintaxis es:

```
READ(<archivo>,<lista de variables>);
```

Donde:

<archivo> Archivo del cual se obtendrán los valores correspondientes para las variables. Use INPUT u omite esta indicación si los valores se proporcionarán a través del teclado.

<lista de variables> Conjunto de variables de los

tipos antes indicados, en el orden correspondiente para la asignación de los valores que se leerán.

Al ejecutarse esta instrucción, usando el archivo INPUT en el READ, el programa se detendrá hasta que se hayan introducido, a través del teclado los datos correspondientes a las variables enunciadas en la instrucción.

#### 5.2.6.1.2. Rutina READLN

La instrucción READLN tiene el mismo propósito que READ, pero difiere de ella en que para indicar el fin de la introducción de los datos por el teclado se realiza siempre con la tecla de <ENTER> o caracter de fin de línea.

El READLN implica "leer hasta el fin de línea"; por ello, hay que indicar a la computadora el fin de la línea con la tecla correspondiente (RETURN, NEWLINE, ENTER, etcétera). Cada uno de los datos debe estar separado entre sí por espacios en blanco o caracteres de fin de línea, al menos uno.

La sintaxis es:

```
READLN(<archivo>,<lista de variables>);
```

donde:

<archivo> y <lista de variables> son los mismos que en la sintaxis del READ.

#### 5.2.6.2. Rutinas de salida

##### 5.2.6.2.1. Rutina WRITE

La rutina WRITE, sirve para "escribir", ya sea en la pantalla, la impresora o en un archivo en disco, constantes o valores. Después de la instrucción WRITE, la siguiente "escritura" se presentará INMEDIATAMENTE DESPUES del último caracter escrito. La sintaxis es:

```
WRITE(<archivo>,<lista de elementos>);
```

Donde:

<archivo> Es el archivo en el que se escribirán los valores de los elementos indicados. Use OUTPUT u omite esta indicación si los valores de los elementos se desean desplegar en la pantalla.

<lista de elementos> Conjunto de posibles valores o constantes cuyos valores se desean presentar. También pueden ser expresiones o funciones.

Deben estar separados por comas.

El elemento puede ser: entero, caracter, lógico o cadena.

#### 5.2.6.2.2. Rutina WRITELN

Tiene el mismo propósito que la rutina WRITE, pero difiere con ésta en que la siguiente escritura se hará en la primera posición de la LINEA SIGUIENTE, tanto en la pantalla como en la impresora o en el archivo correspondiente. La sintaxis es:

**WRITELN(<archivo>,<lista de elementos>);**

Donde:

<archivo> y <lista de elementos> son los mismos que en la rutina WRITE.

#### 5.2.6.3. Formatos de impresión

PASCAL permite que el programador indique el número de columnas o lugares que un valor dado puede ocupar al ser desplegado.

Esto se logra indicando el número de lugares así:

**<elemento>: <número de lugares>**

dentro de la lista de elementos a desplegar. Todos los compiladores de PASCAL permiten un formateo en la impresión de los números REALES, indicando:

- El número total de lugares usados.
- El número de cifras a la derecha del punto decimal.

La sintaxis es:

**<elemento real> : <total de lugares>:<lugares a la derecha del punto>**

#### 5.2.6.4. Sugerencias.

1. Antes de cada READ o READLN tener un WRITE o WRITELN que le indique al usuario el tipo de dato que debe teclear y a qué se refiere.
2. Presentar los resultados, preferentemente formateados y con mensajes que comenten a qué se refiere lo que se está imprimiendo.

## 5.2.7. Bases de programación estructurada

La programación estructurada es una metodología de organización y programación de sistemas que mantiene una lógica ordenada, simple y directa, para facilitar su entendimiento y modificación.

A la Programación Estructurada se le denomina así porque en ello se hace uso de ESTRUCTURAS LÓGICAS DE PROGRAMACIÓN, las cuales son simplemente, un conjunto de instrucciones iterativas y de decisión, que indican la manera lógica en que se comporta el programa. PASCAL es un lenguaje ideal para aprender y aplicar la Programación Estructurada.

### 5.2.7.1. Estructuras lógicas de programación.

Existe un postulado denominado Teorema de la estructura que indica:

"Cualquier programa puede ser elaborado utilizando, únicamente, las estructuras lógicas de programación".

Las estructuras lógicas de programación sirven para CONTROLAR las acciones que realiza un programa. Se clasifican en tres grupos:

- a) Secuencia o Bloque. Proceso.
- b) Selección. SI - ENTONCES - DE LO CONTRARIO.  
EN CASO DE.
- c) Iteración. MIENTRAS - HACER.  
HACER - HASTA QUE.  
DESDE QUE - HASTA QUE - HACER.

Cada una de las estructuras representa una forma natural de razonamiento para la solución de algún problema o elaboración de un conjunto de acciones. Cada una de las seis aseveraciones, a las que llamamos INSTRUCCIONES, consta de:

- Estructuras Lógicas de Programación.
- Secuencias.
- Condiciones.

#### 5.2.7.1.1. Secuencia o bloque (PROCESO)

Una secuencia es la indicación de un proceso, el cual puede involucrar una o más acciones. Una secuencia es un conjunto de una o más secuencias.

Una secuencia (BLOQUE) puede ser muy simple e implicar una sola acción o ser muy compleja e implicar un conjunto ordenado de acciones. Una secuencia implica una o más estructuras lógicas de

programación. Esto es debido a que la función de las estructuras lógicas es controlar la ejecución de las secuencias en el programa. Para representar una secuencia utilizaremos un rectángulo. Si una secuencia consta de más de un conjunto de acciones, éstas se delimitan con las palabras de BEGIN y END.

Un proceso, secuencia o bloque en PASCAL, puede ser:

- Una instrucción, que puede ser:
  - a) Una asignación, o
  - b) Una indicación de ejecución a una rutina.
- Un conjunto de instrucciones delimitadas por las palabras BEGIN y END.
- Una estructura lógica de programación.

#### 5.2.7.1.2. Estructuras de Selección.

Cuestiona al usuario para decidir el valor de una condición, permitiendo según su respuesta la ejecución de un proceso o no, o elegir entre dos o varios procesos.

#### SI - ENTONCES - DE LO CONTRARIO (IF - THEN - ELSE)

Esta estructura indica la elaboración de procesos de acuerdo a una EXPRESION CONDICIONAL.

Esta estructura de selección tiene dos formas: a) SI - ENTONCES; b) SI - ENTONCES - DE LO CONTRARIO.

a) SI - ENTONCES. La sintaxis es:

```
IF (expresión condicional) THEN
-----
<secuencia>
-----
(* end if *)
```

(fig. 5.1)

b) SI - ENTONCES - DE LO CONTRARIO. La sintaxis es:

```
IF (expresión condicional) THEN
-----
<secuencial>
-----
ELSE
-----
<secuencia2>
-----
(* end if *)
```

# SI - ENTONCES

(IF - THEN)

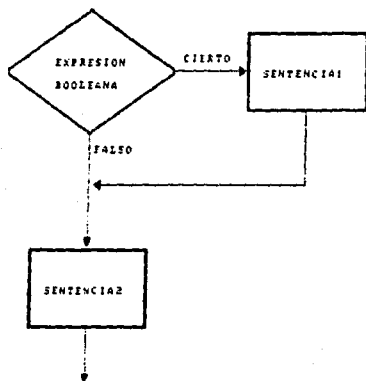


figura 5.1



Nótese que en este caso la <secuencial> no debe llevar punto y coma. Esto es sumamente importante, pues es la forma de indicar que no se trata de un SI - ENTONCES, sino de un SI - ENTONCES - DE LO CONTRARIO.

Nótese que la instrucción anterior a un END no requiere llevar el separador ";" (fig. 5.2).

#### EN CASO DE (CASE)

Esta estructura de selección sirve para indicar la ejecución de una secuencia específica dentro de un grupo de secuencias, respecto al valor de una expresión. La sintaxis es:

```
CASE <expresión escalar> OF
  <constante1>: <secuencia1>;
  <constante2> : <secuencia2>;
  :
  <constanteN> : <secuenciaN>
END;
(* end case *)
```

donde:

<expresión escalar>, es una variable de tipo caracter, entera, lógica, enumerada o subrango.

<constantes>, es una o más constantes, del mismo tipo que la variable, separadas por comas.

Debe tenerse cuidado en asegurar que el valor que tendrá la variable se encuentre en las listas de constantes, de lo contrario existen dos posibilidades: se trunca la ejecución del programa o se sigue de frente en la ejecución, pero nunca se equivoca de caso (fig. 5.3).

# SI - ENTONCES - DE LO CONTRARIO

(IF - THEN - ELSE)

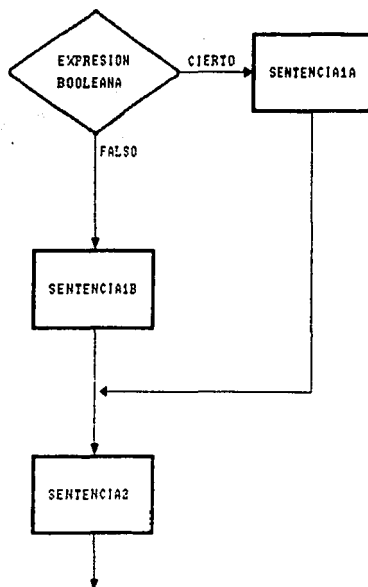


figura 5.2

# EN CASO DE - HACER

(EN CASO DE)

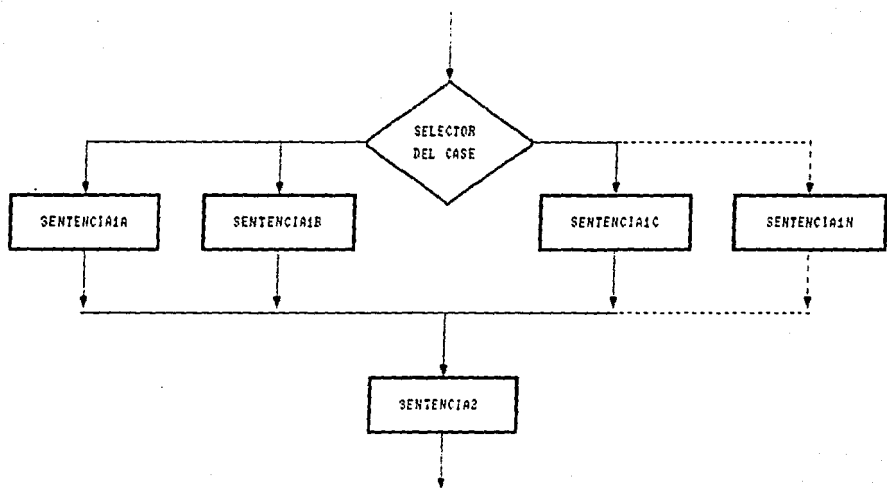


figura 5.3

### 5.2.7.1.3. Estructuras de Iteración.

Estas estructuras permiten la creación de ciclos de trabajo mientras sea válida una condición que el usuario será el encargado de modificar según sea el caso.

#### MIENTRAS - HACER (WHILE ... DO)

Con esta estructura se indica la acción repetitiva de un bloque o secuencia mientras una condición sea cumplida. Debe tomarse en cuenta que la secuencia se ejecuta *unicamente* si la expresión condicional es verdadera. Primero se pregunta por la condición y luego, si ésta es verdadera, se ejecuta la sentencia (fig. 5.4).

La sintaxis es:

```
WHILE (expresión condicional) DO
-----
<secuencia>
-----
(* end while *)
```

Donde:

(expresión condicional) que puede ser desde una simple variable booleana hasta una expresión compuesta por igualdades, relacionadas a través de operadores lógicos.

<secuencia> conjunto de instrucciones ejecutables.

# MIENTRAS - HACER

(WHILE - DO)

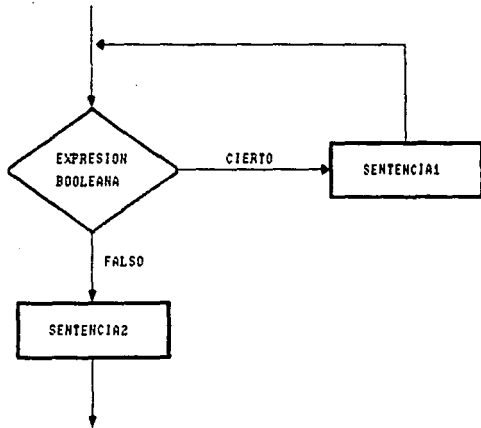


figura 5.4

## HACER - HASTA (REPEAT ... UNTIL)

Con ella se indica la elaboración repetitiva de un bloque o secuencia hasta que una condición se cumpla.

Esta estructura de repetición de acciones tiene la característica de que primero se ejecuta la secuencia y luego se pregunta por la expresión condicional. De tal manera que la secuencia indicada siempre se ejecutará por lo menos una vez, antes de preguntar por la condición (fig. 5.5).

A diferencia del HACER - HASTA el MIENTRAS - HACER primero, evalúa la condición por lo que la secuencia podría no ejecutarse nunca, mientras que en el HACER - HASTA siempre se ejecuta por lo menos en una ocasión.

La sintaxis es:

```
REPEAT
  <secuencia>
UNTIL (expresión condicional);
(→ end repeat →)
```

donde:

<secuencia> conjunto de instrucciones ejecutables.  
(expresión condicional) que puede ser desde una simple variable lógica hasta una expresión compuesta por igualdades, relacionadas a través de operadores lógicos.

Algo importante es el observar que la secuencia del REPEAT puede o no estar limitada por las palabras BEGIN y END. Podría decirse que el REPEAT y el UNTIL contiene un BEGIN y un END respectivamente, de manera implícita.

# HACER - HASTA

(REPEAT - UNTIL)

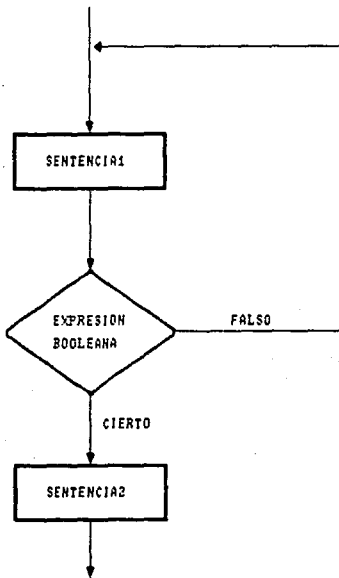


figura 5.5

## DESDE QUE - HASTA QUE - HACER (FOR ... DO)

Esta estructura de iteración se diferencia de las anteriores por su rigidez. Una secuencia será repetida tantas veces como sea propuesta (Fig. 5.6).

La diferencia entre la estructura DESDE QUE - HASTA QUE - HACER respecto a las dos anteriores (Mientras - Hacer y Repite - Hasta) es la poca flexibilidad en el control y número de ciclos.

Con esta estructura, el número de repeticiones de la secuencia será fijada antes de iniciarse la ejecución de la misma, y ningún acontecimiento en la secuencia hará cambiar el número de iteraciones, como sucede en el MIENTRAS - HACER y en REPITE - HASTA.

La sintaxis es:

```
FOR <variable> := <valor inicial> TO <valor final> DO
  <secuencia>
(* end for *)
```

Donde:

<variable> debe ser cualquier tipo escalar, excepto REAL.

<valor inicial> y <valor final> deben ser constantes o variables, con un valor determinado, que debe ser del mismo tipo que <variable>.

<secuencia> conjunto de instrucciones ejecutables.

En este caso (valor inicial)  $\leq$  (valor final) y se trata de incrementos en una unidad.

Por otro lado, existen casos en los que resulta necesario realizar decrementos. Esto se lleva a cabo con la palabra reservada DOWNTO. La sintaxis es:

```
FOR <variable> := <valor inicial> DOWNTO <valor final> DO
  <secuencia>
(* end for *)
```

En este caso, (valor inicial)  $>$  (valor final) y se trata de DECREMENTOS UNITARIOS.



# DESDE QUE -HASTA QUE - HACER

(FOR - DO)

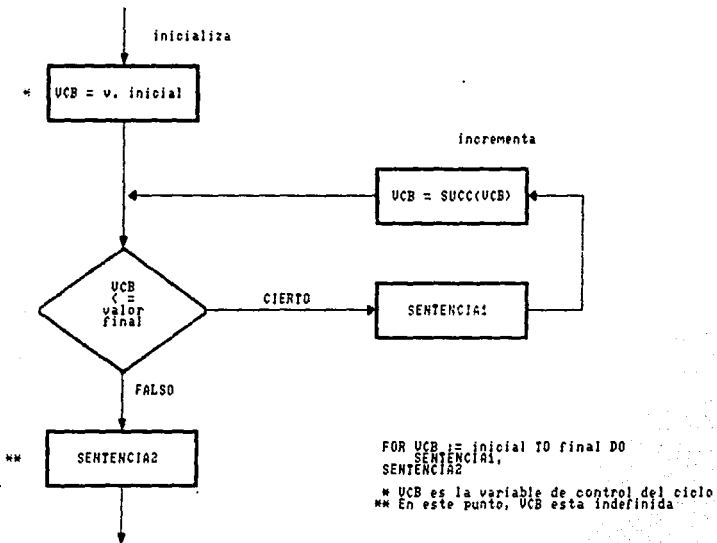


figura 5.6

### 5.2.7.2. Tipos estructurados

En el manejo de los datos, para la solución de un problema a través de un programa, se hace una abstracción de la realidad para representar la información de una manera ordenada.

El lenguaje PASCAL permite implementar un conjunto de abstracciones que son comunes en la mayoría de los problemas de procesamiento de datos. Una de las ventajas que tiene PASCAL, con respecto a otros lenguajes, es la facilidad de crear un nuevo tipo de variable, que en la mayoría de los casos es definido en términos de un tipo primitivo (INTEGER, REAL, CHAR, BOOLEAN).

Una estructura de datos es una conformación ordenada de los datos dentro de la memoria de la computadora, que permite un manejo fácil y rápido de la información.

En PASCAL los tipos estructurados implementados son:

- Arreglos.
- Registros.
- Conjuntos.
- Archivos.

#### 5.2.7.2.1. Arreglos

El arreglo es una estructura de datos fija y homogénea, es decir, el número de elementos de los que consta no variará a lo largo de la ejecución del programa; y, los elementos que constituyen la estructura son del mismo tipo.

La sintaxis es:

```
TYPE
  <tipo de arreglo> = ARRAY [subrango] OF <tipo> ;
VAR
  <variable arreglo> : <tipo arreglo> ;
```

Donde:

<tipo arreglo> es un identificador cualquiera.

<tipo> es cualquier tipo, primitivo o determinado ANTERIORMENTE.

<variable arreglo> es un identificador cualquiera.

<subrango> puede ser de tipo caracter, entero, lógico pero no real. Posee dos límites, uno inferior y otro superior, indicando con ellos el número de elementos que tendrá el arreglo. Ambos límites deben ser de tipo entero, caracter o enumerado. El <tipo del índice> puede indicarse de tres maneras:

1. Con un tipo simple (no real, ni de tipo entero. Este último por limitaciones de memoria y direccionamiento).
2. Con un tipo subrango definido anteriormente.
3. Con los límites, superior e inferior.

Uso y asignación de los elementos de un arreglo - La manera de seleccionar los elementos de un arreglo, ya sea para obtener su contenido o para asignarle un valor, es a través de un índice. Este índice indica la posición del elemento en el arreglo.

La asignación de un valor a un elemento del arreglo se hace de la siguiente manera:

```
<variable arreglo>[<índice>] := <valor>;
```

En donde:

<variable arreglo> es un identificador cualquiera declarado como un arreglo.

<índice> puede ser una constante, una variable o una expresión de cualquier tipo simple, a excepción del tipo real.

<valor> puede ser una variable, una constante o una expresión del tipo correspondiente a los elementos del arreglo.

También es posible asignar un valor a un arreglo completo, siempre y cuando este valor sea del mismo tipo del arreglo.

#### CARACTERES := SIMBOLOS ;

Los dos son arreglos del mismo tamaño y tipo con el mismo subrango declarado como índice.

Para la utilización de un elemento de arreglo, simplemente se hace referencia a él a través del índice. De manera general tendríamos:

```
<variable> := <variable arreglo>[<índice>];
```

Donde:

<variable> es del mismo tipo que los elementos del arreglo.

<índice> como <variable arreglo> se explicaron anteriormente.

### 5.2.7.2.2. Registros

Un Registro (RECORD) es una estructura de datos heterogénea, de configuración variable y su tamaño será establecido por la estructura que se defina.

Un registro es una estructura formada por una o más campos, los cuales pueden ser de diferente tipo. La sintaxis es:

```
TYPE
<tipo record> = RECORD
    <campo1> : <tipo1>;
    <campo2> : <tipo2>;
    :
    :
    <campoN> : <tipoN>;
    END;
(* end record *)
VAR
<variable record> : <tipo record>;
```

Donde:

<tipo record> es el identificador del tipo declarado.  
<campo1> es el identificador del campo correspondiente.  
<variable record> es el identificador de la variable de tipo registro.  
<tipo1> es cualquier tipo primitivo o declarado anteriormente.

Asignación y Selección de valores - La sintaxis para asignar un valor al campo de un registro es:

```
<variable record>.<campo1>:= <valor>;
```

Donde:

<valor> debe ser del mismo tipo que <campo1>.  
<campo1> y <variable record> fueron definidas anteriormente.

Para la selección de un campo se tiene la sintaxis:

```
<variable> := <variable record>.<campo1>;
```

Donde:

<variable> debe ser del mismo tipo que <campo1>.  
<variable record> y <campo1> fueron definidas anteriormente.

El indicador WITH - Con el indicador WITH se tiene una forma más simple y abreviada para la selección de campos. La sintaxis de la indicación WITH es:

```

WITH <variables record> DO
  <secuencia>;
(* end WITH *)

```

Donde:

<variables record> es una o más variables separadas por comas y que son de tipo registro.  
 <secuencia> es el conjunto de instrucciones que son afectadas por el indicador WITH. Los campos de las variables de registro involucradas se manejarán sin necesidad de hacer referencia a las mismas variables de tipo registro.

Registro con campos variantes - Una característica importante de la estructura de registro es la de su posible VARIACION de forma, de acuerdo a ciertas condiciones en la ejecución del programa. La sintaxis del registro, con campos variantes, es:

```

TYPE
<tipo record> = RECORD
  <campo1> : <tipo1>;
  <campo2> : <tipo2>;
  :
  <campoN> : <tipoN>;
CASE <campo de cambio> OF
  <valor1> : (<lista de campos
             variantes1>);
  <valor2> : (<lista de campos
             variantes2>);
  :
  <valorM> : (<lista de campos
             variantesM>);
END;
(* END RECORD *)

```

### 5.3: Virus computacionales

Los virus computacionales son programas de computadora que reciben este nombre por su similitud con los virus biológicos en cuanto a su capacidad de reproducirse, es decir, formar copias de sí mismos, causar daños en el sistema u organismo que los aloja y difícil detección y erradicación. En una computadora pueden causar graves daños como la pérdida irreparable de los datos almacenados o simplemente pequeñas molestias, pueden ser infectadas por los servicios en línea, tableros de boletines electrónicos y discos de usuarios, y los cuales a su vez pueden infectar programas de aplicación, archivos de datos y sistemas operativos en discos.

Una vez ubicados en la memoria RAM de la computadora, los virus pueden infectar a otros programas que sean cargados para su ejecución. Los virus destruyen datos, modifican la información y sabotean las impresiones.

Desarrollados a partir de 1986 en Lahore, Pakistán, por dos hermanos comerciantes de computadoras y desde entonces se han desarrollado diferentes tipos de virus pudiendo ser clasificables en dos categorías:

Tipo "troyano" - Por su semejanza con el caballo de Troya. Invaden los discos tanto duros como flexibles usados en la computadora en el momento menos esperado y sin ningún aviso, esporádica y lentamente comienzan a hacer daños cada vez más intensos y frecuentes. Cuando los daños se hacen evidentes suele ser demasiado tarde para intentar recuperar información.

Tipo "bomba de tiempo" - Este tipo de virus permanece inactivo, latente, esperando solamente la orden adecuada, que puede ser cualquier condición preprogramada en la máquina, para inmediatamente y con frecuencia ponerse en actividad produciendo grandes daños en un tiempo muy breve.

A raíz de todo esto, han surgido un gran número de "vacunas" y "detectores", programas diseñados específicamente para localizar y destruir los virus informáticos.

Las "vacunas" revisan los archivos contenidos en disco o en la memoria RAM en busca de los patrones de almacenaje o configuraciones de bits que asemejan a la de los virus conocidos. Es recomendable revisar los discos constantemente.

Existen en el mercado una gran cantidad de programas antivirus como son: NORTON ANTI-VIRUS, Certus LAN, Data Physician, Virex-PC, Virucide, Vi-Spy, SCAN, entre otros. Es necesario prepararse para adquirir oportunamente lo más reciente en paquetería o sistemas de prevención antiviral.

**VI. SOFTWARE DE APLICACIÓN Y SERVICIOS DE RED**

## VI. SOFTWARE DE APLICACION Y SERVICIOS DE RED

El propósito de esta unidad es que el alumno conozca las características y áreas de uso de los principales programas de aplicación y los conceptos básicos de redes.

### **6.1. Ambientes gráficos**

Son en los cuales la imagen se representa a través de símbolos.

**Iconos** - Es una representación gráfica asociada a una aplicación, programa o grupo de programas dentro de un ambiente gráfico. Dichas imágenes al ser "abiertas" o maximizarse se convierten en ventanas, mientras que al "cerrarse" o minimizarse adoptan la forma de un icono.

**Ventana** - Están compuestas por más elementos que los iconos, y realizan un mayor número de funciones que se ejecutan por elección de opciones en un menú, con un retorno de carro o un "click" del ratón, o por medio de botones, los cuales se activan con un solo "click" del ratón. Las ventanas cuentan con los siguientes elementos:

- Barra de título. Informa que programa se está ejecutando.
- Botón de la caja de control. Permite abrir la caja de control de la ventana.
- Botón de minimizar. Convierte a la ventana en un icono sin cerrar la aplicación.
- Botón de maximizar. Provoca que la ventana abarque toda la pantalla.
- Barra de movimiento vertical. Permite desplazarse hacia arriba y hacia abajo dentro de una ventana si es que esta última no posee el tamaño suficiente para mostrar todo su contenido distribuido verticalmente.
- Barra de movimiento horizontal. Similar ala anterior, pero en este caso en sentido horizontal, si es que la ventana no tiene el tamaño suficiente para mostrar todos sus datos.
- Barra de menú. Contiene las opciones de trabajo de las ventanas, ejecutando comandos o procesos ya definidos.
- Borde. Es el contorno de la ventana, a partir de ella se puede variar el tamaño de la misma.



## 6.2. Hojas de cálculo

Las hojas de cálculo electrónicas son programas que permiten la manipulación de datos acomodados en renglones y columnas. A la intersección de un renglón y una columna se le conoce como celda y en la cual podrá estar contenido un texto, un número o una fórmula que establezca relación con otras columnas y/o celdas.

Al modificar el contenido de una celda se realiza un recálculo automático en las celdas que se relacionen con la celda modificada. Además es posible utilizar instrucciones lógicas avanzadas; importar o intercambiar información con otros paquetes de aplicación; generar e imprimir gráficas de distintos tipos en base a la información contenida en la hoja de cálculo.

Entre las hojas de cálculo más utilizadas están: LOTUS 1-2-3, EXCEL, VISICALC.

## 6.3. Manejadores de bases de datos

Conjunto de equipos y programas que se utilizan en la organización, actualización y almacenamiento de registros y archivos. Permite ahorrar una gran cantidad de tiempo al acceder la información desde un archivo.

Entre los manejadores de bases de datos más utilizados están: DBASE, SYSDBASE, ORACLE.

## 6.4. Editores gráficos

Convierten o simulan la pantalla de la computadora como un tablero o lienzo luminoso sobre el cual se puede crear o modificar imágenes gráficas, haciendo uso de herramientas de arte computarizado. Entre ellas están: HARVARD GRAPHICS, PAINT BRUSH, COREL DRAW, AUTOCAD, FORM TOOLS.

## 6.5. Servicios de red

Una red es un sistema de comunicaciones de datos que enlaza dos o más computadoras y dispositivos periféricos. Entre los servicios que presta se encuentran:

1. Compartición de archivos y programas.
2. Compartición de recursos de red.
3. Compartición de bases de datos.
4. Expansión económica de la base de PC's.

5. Grupos de Trabajo.
6. Correo electrónico.
7. Software de grupos y de flujo de trabajo.
8. Gestión centralizada.
9. Mejora de la estructura corporativa.
10. Transferencia de archivos.

1. **Compartición de archivos y programas.** Los paquetes más populares de software están disponibles en versiones de red produciendo un ahorro considerable de costos. Los archivos de programas y datos se almacenan en un servidor de archivos y acceden a ellos muchos usuarios de red.
2. **Compartición de recursos de red.** Los recursos de red incluyen impresoras, trazadores gráficos y dispositivos de almacenamiento ofreciendo un enlace de comunicación que permite a los usuarios compartir estos dispositivos.
3. **Compartición de bases de datos.** Un sistema gestor de bases de datos es una aplicación ideal para una red. Una característica de red llamada bloqueo de registro permite que múltiples usuarios accedan simultáneamente a un archivo sin corromper los datos. El bloqueo de registro asegura que dos usuarios no editan el mismo registro a la vez.
4. **Expansión económica de la base de computadoras personales.** Las redes proporcionan una forma económica de aumentar el número de computadoras en una organización. Se pueden enlazar económicamente estaciones de trabajo sin disco a una red que utilice la unidad de disco duro del servidor para el arranque y el almacenamiento.
5. **Grupos de Trabajo.** Una red ofrece un medio de crear grupos de usuarios que no se localicen necesariamente en el mismo departamento. Dichos grupos facilitan nuevas estructuras colectivas "planas" donde las personas de diversos y remotos departamentos pertenecen a proyectos de grupos especiales.
6. **Correo electrónico.** Es la aplicación más usada en redes. A través de él los usuarios pueden comunicarse de forma rápida entre ellos. Cuenta con un buzón electrónico que permite almacenar la correspondencia electrónica que no puede ser atendida de forma inmediata. Además proporciona una forma rápida y sencilla de empaquetar distintos tipos de información que se desee transferir a otro usuario. Puede incluir la capacidad de cifrar el contenido de los mensajes o añadir una firma digital para probar la

autenticidad de los mismos y prevenir alteraciones.

7. **Software de grupos y de flujo de trabajo.** Se diseñó específicamente el software de grupos y el software de flujo de trabajo para las redes, y para aprovechar las ventajas de los sistemas de correo electrónico que ayudan a los usuarios a colaborar en proyectos, programas y procesamiento de documentos.
8. **Gestión centralizada.** Una red ofrece un camino a los servidores centralizados y a sus datos, junto con otros recursos. Cuando los dispositivos se localizan en un lugar, las modernizaciones de hardware, las copias de seguridad de software, el mantenimiento y la protección de sistemas son mucho más fáciles de manipular.
9. **Mejora de la estructura corporativa.** Las redes pueden cambiar la estructura y la gestión de una organización. Los usuarios que trabajan en un departamento concreto y para un director específico ya no tienen necesidad de estar en la misma área específica. Pueden tener sus oficinas en áreas donde se necesite más su experiencia. La red los une con sus directores y compañeros de departamento. Esta disposición es útil para proyectos especiales donde individuos de distintos departamentos, como investigación, producción, mercadotecnia, necesitan trabajar estrechamente unos con otros.
10. **Transferencia de archivos.** Las personas que están trabajando conjuntamente en un proyecto normalmente necesitan compartir los archivos. Una manera de lograrlo es teniendo una máquina en donde se conserva el original de cada archivo, mientras se transfieren copias a otras máquinas según se necesiten. Otra manera sería la de conservar "vivo" cada archivo en la máquina donde se creó, y hacer que los usuarios de otras máquinas soliciten copias cuando las necesiten.

## CONCLUSIONES

Este trabajo tuvo como finalidad, el proporcionar a los alumnos que cursan en el bachillerato de la Escuela Nacional Preparatoria, la materia de "Introducción a la Informática", un texto sencillo, completo y realizado de acuerdo con el programa de estudios de la asignatura.

Finalmente cabe señalar que personalmente considero que el plan de estudios debe ser modificado en cuanto al su orden en los contenidos y revisado continuamente en cuanto a los programas de aplicación que se manejan, ya que los avances que se presentan en el área de informática son muy frecuentes.

Cabe mencionar la importancia que la materia tiene en la formación laboral y profesional de los alumnos que egresan de la Escuela Nacional Preparatoria y en general del bachillerato. Por lo tanto considero que debe ser de primordial importancia incluirla dentro del plan de estudios de cualquier institución de nivel medio superior desde su primer año o semestre, sin descuidar las demás materias de formación académica.

**INTRODUCCION A LA INFORMATICA**  
**(Programa de Estudios)**

**UNIDAD I. ANTECEDENTES DE LA INFORMATICA.**

- Evolución histórica de las diversas formas de procesamiento de la información.
  - \* Abaco, dispositivos mecánicos de cálculo, dispositivos electromecánicos de cálculo, primera computadora digital.
  - \* Tipos de computadoras.
  - \* Generaciones de computadoras.
- Terminología básica usada en Informática y Computación.
  - \* Informática, computación, computadora, bit, byte, palabra, archivo, comando, software, hardware, programa, software de aplicación, dato, información.
- Implicaciones sociales de las computadoras.
  - \* Ética, cuestiones sociales y dependencia a los equipos de cómputo.
- Aplicaciones.
  - \* En la educación, administración, ciencia, industria, militar y en el arte.

**UNIDAD II. ESTRUCTURA FISICA DE UNA COMPUTADORA.**

- Equipo central de una computadora.
  - \* Unidades de entrada y salida.
  - \* Memoria.
  - \* Unidad central de proceso.
- Equipo periférico.
  - \* Dispositivos de almacenamiento (discos duros, flexibles, ópticos, cintas, discos compactos CDs).
  - \* Dispositivos de entrada.
  - \* Dispositivos de salida.
  - \* Dispositivos de entrada y salida.
- Teclado.
  - \* Alfanumérico, numérico y teclas propias de una computadora.
  - \* Teclas de funciones y su interdependencia del software.

### UNIDAD III. PROCESAMIENTO DE TEXTOS.

- Programa de aplicación procesador de texto.
  - \* Características y ventajas de la utilización de un procesador de texto.
- Partes básicas de un documento.
  - \* Se reconocerán las partes de un documento como título, página, párrafo, bloque, tipografía, alineamiento y pie de página.
- Comandos básicos para la manipulación del procesador de textos.
  - \* Cómo iniciar y finalizar una sesión en el procesador.
  - \* Captura de datos.
  - \* Almacenamiento de los datos.
  - \* Presentación de la información. Títulos, alineación (izquierda, derecha, centrado, justificación), fuentes (tipo y tamaño de letra), numeración de páginas.
  - \* Modificación de los datos (inserción y borrado).
  - \* Manejo de bloques (copiar, mover y borrar).
  - \* Impresión de la información.

### UNIDAD IV. ESTRUCTURA LOGICA DE UNA COMPUTADORA.

- Sistemas numéricos.
  - \* Sistema base decimal, binario y otras bases numéricas.
  - \* Códigos de comunicación y almacenamiento (ASCII u otros).
- Sistema Operativo.
  - \* Comandos para manejo de discos (format, diskcopy, dir, label).
  - \* Comandos para el manejo de directorios (md, rd, cd, tree).
  - \* Comandos para el manejo de archivos (del, copy, rename).
  - \* Comandos de configuración y operatividad (cls, date, time, prompt).

### UNIDAD V. METODOLOGIA DE SOLUCION DE PROBLEMAS Y PROGRAMACION.

- Método de solución de problemas.
  - \* Definición del problema.
  - \* Análisis del problema.
  - \* Diseño de la solución del problema.
  - \* Desarrollo de la solución del problema.
- Lenguajes de programación.
  - \* Concepto, tipos y aplicaciones de los lenguajes informáticos.
- Fundamentos de programación en un lenguaje estructurado.
  - \* Estructura básica del lenguaje.
  - \* Variables y tipos primitivos.

- \* Operadores.
- \* Secuencia.
- \* Decisión.
- \* Iteración.

- Virus.
  - \* Los virus como programas, características, efectos sobre la información y el hardware, prevención, detección y eliminación.

#### UNIDAD VI. SOFTWARE DE APLICACION Y SERVICIOS DE RED.

- Ambientes gráficos.
  - \* Ventanas e iconos.
- Hojas de cálculo.
  - \* Características y usos.
- Manejadores de bases de datos.
  - \* Importancia del manejo de información con características comunes y su tratamiento con los manejadores de bases de datos.
- Editores gráficos.
  - \* Presentaciones electrónicas, usos.
- Servicios de red.
  - \* Correo electrónico, transferencia de archivos, listas de discusión, búsqueda de información.

## GLOSARIO DE TERMINOS

- ANALOGIA** - Relación de semejanza entre cosas distintas. Relación, proporción. En matemáticas experimentales y en física, correspondencia entre dos magnitudes de naturaleza distinta.
- ALGORITMO** - Lista de instrucciones donde se especifica una sucesión de operaciones necesarias para resolver cualquier problema dado. Características : \* Debe darse en la forma de una lista finita de instrucciones que indiquen el procedimiento a seguir en cada paso del cálculo. \* Como una lista de instrucciones que definen un cálculo que puede efectuarse con cualesquiera datos iniciales y que, en cada caso, proporciona una respuesta final que en el mejor de los casos será la solución al problema o un mensaje indicando el error o imposibilidad de efectuar el cálculo con los datos proporcionados.
- ALGORITMO NUMERICO** - Basado en el uso de las operaciones aritméticas. No se conoce de antemano el número de operaciones que deben efectuarse para resolver un problema particular; depende del problema específico y sólo se descubre en el transcurso de la aplicación del algoritmo.
- ARCHIVO** - Forma de organización de datos en la computadora.
- CAMPO** - Subsección de un registro en un archivo, que contiene un solo tipo de información. Se utiliza para registrar el mismo tipo de información, el contenido de un campo por lo general se trata de una unidad.
- CIBERNETICA** - Ciencia que estudia el funcionamiento de las conexiones nerviosas en los seres vivos. \* Arte de construir y manejar aparatos y máquinas que mediante procedimientos electrónicos efectúan automáticamente cálculos complicados y otras operaciones similares.
- CODIFICACION** - Es la actividad de traducir el método de solución a algún lenguaje de computadora.
- COMPILADOR** - Programa que traduce un programa en lenguaje de alto nivel en código máquina
- COMPUTADORA** - Algo que computa; especificando: un dispositivo electrónico programable que puede almacenar, recuperar y procesar datos.



- COMPUTADORA ANALOGICA - Aparato computador cuyos componentes se ajustan de modo que sus leyes físicas de funcionamiento sean análogas a las leyes matemáticas de proceso que se trata de estudiar.
- COMPUTADORA DIGITAL - Aquel en que todas las magnitudes se traducen en números, con los cuales opera para realizar los cálculos.
- COMPUTADORA HIBRIDA - El compuesto por una parte analógica y otra digital y que aprovecha óptimamente las características de ambas.
- DIAGRAMA DE FLUJO - Representación gráfica para la definición, análisis o solución de un problema, en la que se utilizan símbolos para representar operaciones, datos, flujo y equipo. Puede trazarse con diferentes niveles de detalle para mostrar los pasos de procedimiento amplios o detallados. Da un panorama general y amplio.
- DISPOSITIVO PERIFERICO - Un dispositivo de entrada, salida o de almacenamiento auxiliar de una computadora.
- EDITOR - Ayuda a crear y modificar el texto de un programa.
- ENSAMBLADOR - Sistema que traduce un programa en lenguaje ensamblador a código máquina.
- ENTRADA/SALIDA - Medios y dispositivos usados para lograr la comunicación hombre-máquina.
- HARDWARE - Los componentes físicos de una computadora.
- INFORMATICA - Disciplina que incluye las diversas técnicas y actividades relacionadas con el tratamiento lógico y automático de la información, en cuanto ésta es soporte de conocimientos y comunicación humana. Se apoya fundamentalmente en la utilización de los ordenadores o calculadores electrónicos y también en un tipo peculiar de análisis conceptual, que implica una actitud mental característica en la manera de afrontar los problemas de la información, que la vincula con la cibernética.
- INSUMOS - Instrucciones y datos que la computadora requiere para realizar trabajos. Al conjunto de instrucciones que solucionan un problema se le conoce como programa. Los datos son los valores que un programa toma para transformarlos en información.

- INTERFACE** - Hardware que comunica las computadoras y dispositivos periféricos. La circuitería electrónica que conecta la línea de entrada/salida de la computadora para actuar como intermediario y traducir voltajes, formatos de señal y otros pulsos necesarios para establecer una comunicación entre la computadora y los dispositivos periféricos. Una zona compartida en donde se encuentran los sistemas independientes y actúan o se comunican unos con otros.
- LENGUAJE DE PROGRAMACION** - Un conjunto de reglas, símbolos y palabras especiales utilizadas para construir un programa.
- LENGUAJE MAQUINA** - El lenguaje usado directamente por la computadora y compuesto de instrucciones codificadas en binario.
- PASCAL** - Lenguaje de programación diseñado por Niklaus Wirth en 1968. Su nombre se debe al matemático francés Blaise Pascal. Destinado originalmente a ser una herramienta de enseñanza para los conceptos de programación. Por ser un lenguaje altamente estructurado, se presta bien a las técnicas de programación tradicionales. Su simplicidad y legibilidad le hacen uno de los mejores lenguajes para enseñar los conceptos de programación e impartir buenos hábitos.
- PROCESAMIENTO POR LOTES** - Una técnica para ejecutar programas y datos sin interacción intermedia de usuario con la computadora.
- PROCESAMIENTO, TIEMPO DE** - Tiempo de máquina disponible para manipular datos. Tiempo entre la entrada al CPU y la salida de allí.
- PROGRAMA** - Conjunto detallado y explícito de instrucciones de computadora para realizar algún trabajo. Un programa se escribe en un lenguaje adecuado para la entrada a la computadora; es completo e independiente; es divisible en una rutina principal y subrutinas para comodidad del programador.
- PROGRAMA DE COMPUTADORA** - Una secuencia de instrucciones que indica las acciones que han de ejecutarse.
- PROGRAMACION** - Arte y ciencia de planear la solución de los problemas al reducir el plan a un conjunto de instrucciones razonables para la máquina que dirige las acciones de un sistema de computación o procesamiento de datos. La computadora realiza

operaciones aritméticas y lógicas que dan soluciones a los problemas. La planificación, proyección o ejecución de una tarea o suceso.

- PROGRAMACION DE COMPUTADORA - El proceso de planificar una secuencia de instrucciones que ha de seguir una computadora.
- RED - Grupo de computadoras que se comunican a través de líneas telefónicas o que emplean ondas de radio, infrarrojas, microondas o simples cables.
- REGISTRO - Grupos de elementos de datos, elementos o códigos adyacentes que se manipulan como una unidad.
- ROUTINA - Conjunto de instrucciones de computadora digital en una secuencia diseñada para realizar una función específica. Una rutina puede ser independiente, pero lo más común es que se utilice en interacción con otras.
- SEMANTICA - El conjunto de reglas que da el significado de una construcción del lenguaje.
- SEUDOCODIGO - Descripción con palabras o fórmulas de un algoritmo o programa; resumen general de las funciones de un programa que se utiliza con un diagrama de flujo para diseñar o documentar un programa. También es un código intermedio de lenguaje de computadora (entre el código fuente y el de máquina) diseñado para permitir que los programas sean transportables entre dos o más tipos de computadoras.
- SINTAXIS - Las reglas formales que gobiernan la construcción de secuencias válidas en un lenguaje.
- SISTEMA INTERACTIVO - Comunicación directa entre el usuario y la computadora; una conexión terminal-computadora que permite la entrada directa de programas y datos y suministra realimentación inmediata del usuario.
- SISTEMA OPERATIVO - Conjunto de programas que gestionan todos los recursos de la computadora. Tales como introducir programas, llamar al compilador, ejecutar el programa objeto resultante, etcétera.
- SOFTWARE - Los programas de una computadora; el conjunto de todos los programas utilizables en una computadora.

- SUBROUTINA - Grupo de instrucciones de computadora diseñado para ser usado por otras rutinas a fin de lograr un objetivo específico. Por lo general, una subrutina se emplea más de una vez.
- UNIDAD ARITMETICO-LOGICA - La componente de la computadora que ejecuta las operaciones aritméticas (suma, resta, multiplicación, división) y las operaciones lógicas (comparación de dos valores).
- UNIDAD CENTRAL DE PROCESAMIENTO - El "cerebro" de una computadora, el cual interpreta y ejecuta las instrucciones.
- UNIDAD DE CONTROL - La componente de la computadora que controla las acciones de las otras componentes para que se ejecuten las instrucciones (un programa) en secuencia.
- UNIDAD DE MEMORIA - El almacén interno de datos de una computadora.
- VALORES POR OMISION - Cuando un programa o sistema se carga en la computadora, el software (del fabricante) hace ciertas suposiciones, es decir son valores de inicio que se dan a los parámetros de un programa.

## RIBLIOGRAFIA

- 1.- LIBRO DE LA COMPUTACION  
Bradbeer / De Bono / Laurie  
Edit. Fondo Educativo Interamericano
- 2.- CURSO GENARAL DE INFORMATICA  
Gayan / Sagarra  
Edit. Gustavo Gili
- 3.- INTRODUCCION AL PROCESAMIENTO DE DATOS  
Harris  
Edit. Limusa
- 4.- INFORMATICA  
Price  
Edit. Interamericana, 3a. Edición
- 5.- COMO USAR DISCOS EN COMPUTACION  
Jorge Tamayo  
Edit. Macrobit Editores
- 6.- INTRODUCCION A LOS COMPUTADORES  
Craig Field  
Edit. Alianza Editorial, 2a. Edición, 1982
- 7.- FUNDAMENTOS DE COMPUTACION  
Arechiga G.  
Edit. Noriega Limusa
- 8.- INTRODUCCION A LAS COMPUTADORAS  
Larry Long  
Edit. Prentice-Hall, 2a. Edición.
- 9.- INFORMATICA : LAS COMPUTADORAS EN LA SOCIEDAD  
James Radlow  
Edit. McGraw Hill, 1987
- 10.- INFORMATICA : PRESENTE Y FUTURO  
Sanders, Donald H.  
Edit. McGraw Hill, 1985

- 11.- FUNDAMENTOS DE PROGRAMACION  
 Joyanes  
 Edit. McGraw Hill, 1988
- 12.- INTRODUCCION A LA COMPUTACION Y A LA PROGRAMACION  
 ESTRUCTURADA  
 Levine González  
 Edit. McGraw Hill
- 13.- COMPUTACION  
 Bitter  
 Edit. Addison-Wesley Iberoamericana, 1987
- 14.- INTRODUCCION A LA INFORMATICA  
 Tim Duffy  
 Grupo Editorial Iberoamérica, 1993.
- 15.- WYSIWYG O LO QUE ES LO MISMO LQUELQO  
 Viso Gurovich, Elisa  
 Revista "Ciencias", 1988.
- 16.- PASCAL  
 Nell Dale, David Orshalick  
 McGraw Hill, 1987.
- 17.- INTRODUCCION A UN LENGUAJE DE PROGRAMACION ESTRUCTURADA :  
 PASCAL  
 Editorial Porrúa, 1987
- 18.- LAN TIMES. ENCICLOPEDIA DE REDES (NETWORKING)  
 Tom Sheldod  
 Osborne/McGraw Hill, 1985.
- 19.- REDES DE ORDENADORES  
 Andrew S. Tanenbaum  
 Prentice Hall, 2a. edición, 1991
- 20.- SISTEMA OPERATIVO MS-DOS (Guía de Usuario)  
 Paul Hoffman / Tamara Nicoloff  
 Edit. McGraw Hill.
- 21.- ENCICLOPEDIA DE TERMINOS DE MICROCOMPUTACION  
 Linda Gail Christie / John Christie  
 Edit. Prentice Hall, 1986.