

83
Zy



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE COMPUTO PARA CONSULTA DE LA
NORMA OFICIAL MEXICANA NOM-001-SEMP-1994
PARA INSTALACIONES ELECTRICAS (SISPCNOM)

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
ALEJANDRO MONTERROSA ESCOBAR
MIGUEL ANGEL LOPEZ MONROY
RUBEN AGUILAR MONDRAGON

DIRECTOR DE TESIS:

ING. ARTURO MORALES COLLANTES



TESIS CON
FALLA DE ORIGEN

Ciudad UNIVERSITARIA, MEXICO, D.F.

1996

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.



INGENIERÍA
EN
COMPUTACIÓN
UNAM

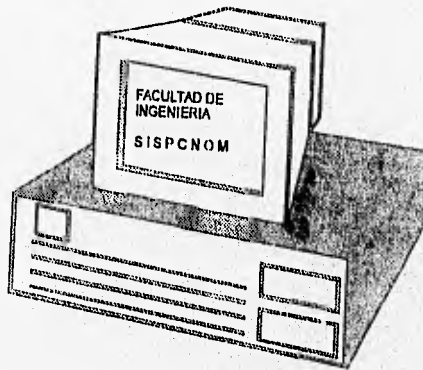


UNAM

FACULTAD DE INGENIERÍA

SISPCNOM

**SISTEMA PARA CONSULTA DE LA
NORMA OFICIAL MEXICANA
NOM-001-SEMP-1994**



S I S P C N O M

AGRADECIMIENTOS.

A la *Universidad Nacional Autónoma de México*, por habernos permitido formar parte de esta gran institución, por su enseñanza y formación tanto académica como personal. Siempre la llevaremos en nuestros corazones.

A la *Facultad de Ingeniería*, por todas las instalaciones y equipo de trabajo proporcionados. que nos permitieron obtener un buen aprendizaje durante estos años en sus aulas y nos hace sentir orgullosos de haber estudiado en ella.

A los *Profesores*, por ser nuestro ejemplo a seguir. por todos los conocimientos y experiencias que depositaron en cada uno de nosotros; esas aportaciones nos han forjado como personas íntegras y productivas a la sociedad.

Al *Ing. Arturo Morales Collantes* por su asesoría, sugerencias y todas las facilidades proporcionadas a lo largo de este trabajo.

Al *Laboratorio de Computación* de la Facultad de Ingeniería. por el material de apoyo proporcionado.

A la familia *Aguilar Mondragón* por aguantarnos toda la lata que dimos y por las facilidades brindadas en la culminación de este trabajo.

Agradecemos por último, a todas aquellas personas que de alguna forma hicieron posible la realización de esta tesis.

A DIOS:

El todo poderoso, que me ha dado tantas satisfacciones a lo largo de mi vida, gracias por permitirme la vida hasta hoy, por tener una familia hermosa y personas muy valiosas a mi alrededor; gracias por todas y cada una de las bendiciones que has tenido conmigo. Todo te lo debo a tí y sin tí no sería lo que ahora soy.

A mi Padre:

Por ese ejemplo de lucha, sacrificio y superación que a diario me demuestras, por ese esfuerzo y apoyo que todos los días me brindas y sobre todo por ese amor que siempre me acompaña. Por lo que en mi sembraste algún día te rindo gratitud, ya que todos mis éxitos son tuyos.

A mi Madre:

Por todos esos consejos que han sido de gran utilidad a lo largo de mi vida, por ese cariño, estímulo y comprensión que has tenido conmigo, y porque has sabido ser la mejor madre con todos tus hijos. Esta satisfacción es toda tuya.

A mis Hermanos:

Gabriel: Por compartir conmigo tan gratos momentos y por ser para mí un ejemplo a seguir en la vida.

Elia: Por todo el amor que siempre me demuestras y por ser una persona tan inteligente y madura, que haces que cada día aprenda algo nuevo.

A Sara:

Por todo este tiempo de felicidad, que me has permitido pasar contigo, por que en todo momento llenas mi vida de luz y de amor. Tu apoyo y confianza me han ayudado a superar todo obstáculo en mi camino. Por que parte de lo que hoy soy, es gracias a tu ayuda y cariño.

A mi Abuelita y mi Tío Alfredo:

Por ese ejemplo de trabajo y lucha, por su aliento y apoyo a lo largo de mi vida.

A todos mis familiares y Amigos que de alguna manera me han ayudado a lograr mis metas.

RUBÉN.

A mis Padres:

Gracias por ese apoyo en los momentos más difíciles durante el transcurso de mi carrera, por el apoyo económico y moral que me brindaron, quiero compartir este logro con ustedes.

A mi Hermano Alejandro:

Por ayudarme en algunas ocasiones y tenerme paciencia cuando las cosas no me salían del todo bien.

A mi Tía Juana:

Por su cariño y comprensión .

A mis Tías que viven en el Estado de Hidalgo:

Por ser tan amables conmigo durante mi estancia en su casa y por su apoyo moral.

A mis Primas y mis Primos:

Por ayudarme en ocasiones oportunas cuando inicie mi carrera, por su sinceridad y amistad en cualquier momento.

A mi Abuelito:

Por ser tan bueno, cariñoso conmigo y por sus palabras de aliento.

A mi Abuelita:

Por estar en la casa esperándome para felicitarme, por su bondad y su amor.

Gracias a todos mis **amigos** de la Facultad de Ingeniería y del Instituto José Vasconcelos por su apoyo, sus comentarios y sugerencias.

MIGUEL ANGEL .

A Dios:

Por llenar mi vida de bendiciones y permitirme realizar este gran sueño.

A mi Padre:

Por todo el amor, comprensión y apoyo que me has brindado durante toda mi vida. Agradezco tus consejos, esfuerzos y sacrificios que representan un ejemplo de superación para mí y hoy hacen posible mi realización profesional. Soy lo que tú hiciste de mí. Eres un padre excepcional.

A mi Madre:

Por el amor y esmero con el que siempre me educaste, desde que era apenas un niño hasta hoy. Me has enseñado a ser mejor y conseguir las cosas a base de esfuerzo y tenacidad. Eres la mejor de las madres.

A mis Hermanos:

Amílcar: Por que siempre me has alentado y representas para mí un ejemplo como hermano y profesionista. Nunca cambies.

José Alberto: Por enseñarme a superar todos los obstáculos y por la ayuda que siempre me brindas, la cual me ha permitido obtener esta meta. Cuenta siempre conmigo.

Marina Fabiola: Por todo el cariño y amor que siempre me demuestras llenando mi vida de alegría y felicidad. Eres una gran hermana.

A Mónica:

Por todos estos años de apoyo incondicional, en los que has llenado mi vida de paz, amor y felicidad. Gracias por esa confianza que nos ha permitido salir adelante y conseguir todas las metas propuestas. Cuenta siempre conmigo. Eres una gran mujer.

A mi Cuñada:

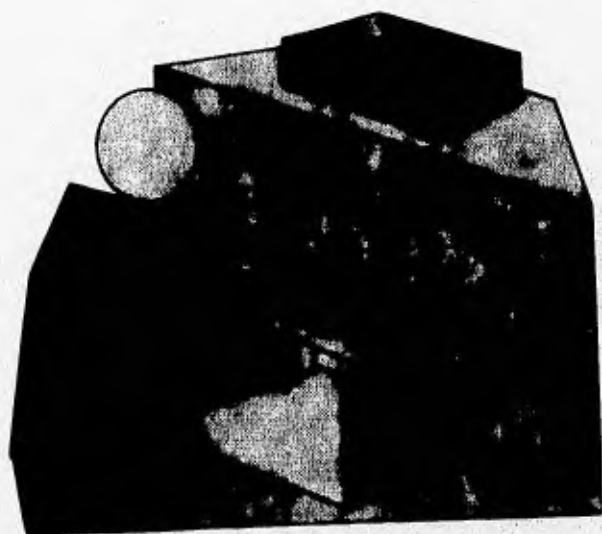
MA. DE JESUS: Por todo el apoyo y amistad que me has brindado.

A mis Familiares y Amigos por confiar en mí y ayudarme a salir siempre victorioso.

ESTE LOGRO ES DE TODOS...

ALEJANDRO .

INDICE



ÍNDICE**INTRODUCCIÓN****CAPÍTULO 1. CONCEPTOS BÁSICOS**

- | | |
|--|-----------|
| 1.1 Definiciones de normalización y consideraciones sobre este concepto | 2 |
| 1.2 Origen y evolución de las normas | 7 |
| 1.3 Norma Oficial Mexicana (NOM) | 15 |

CAPÍTULO 2. NORMA NOM-001-SEMP-1994

- | | |
|--------------------------------|-----------|
| 2.1 Antecedentes | 21 |
| 2.2 Expedición | 26 |
| 2.3 Objetivos | 27 |
| 2.4 Campo de aplicación | 27 |
| 2.5 Referencias | 28 |
| 2.6 Generalidades | 29 |

CAPÍTULO 3. ANTECEDENTES DEL SISTEMA

- | | |
|--|-----------|
| 3.1 Síntesis evolutiva del software | 32 |
| 3.2 Cualidades del software | 35 |
| 3.3 Diseño de sistemas | 38 |
| 3.4 Introducción a las bases de datos | 40 |
| 3.5 El compilador Clipper | 49 |

**CAPÍTULO 4. SITUACIÓN ACTUAL, PLANIFICACIÓN Y
PUNTOS A DESARROLLAR DEL SISTEMA**

| | |
|---|-----------|
| 4.1 Problemática actual | 58 |
| 4.2 Planeación | 61 |
| 4.3 Estrategias para el desarrollo de sistemas | 67 |

CAPÍTULO 5. ANÁLISIS Y DISEÑO DEL SISTEMA

| | |
|--|-----------|
| 5.1 ¿Qué es el análisis y diseño de sistemas? | 74 |
| 5.2 Análisis del sistema | 75 |
| 5.3 Métodos de análisis de requerimientos | 78 |
| 5.4 Seguridad del sistema | 85 |
| 5.5 Diseño del sistema | 87 |

**CAPÍTULO 6. IMPLANTACIÓN Y MANTENIMIENTO DEL
SISTEMA**

| | |
|---------------------------------------|------------|
| 6.1 Introducción | 133 |
| 6.2 Implantación del sistema | 134 |
| 6.3 Depuración del sistema | 138 |
| 6.4 Mantenimiento del software | 140 |
| 6.5 Integridad de los datos | 148 |

| | |
|---------------------|------------|
| CONCLUSIONES | 150 |
|---------------------|------------|

| | |
|-----------------|------------|
| GLOSARIO | 153 |
|-----------------|------------|

| | |
|---------------------|------------|
| BIBLIOGRAFÍA | 162 |
|---------------------|------------|

| | |
|----------------|--------------------------|
| ANEXO 1 | MANUAL DE USUARIO |
|----------------|--------------------------|

| | |
|----------------|-----------------------------|
| ANEXO 2 | LISTADO DE PROGRAMAS |
|----------------|-----------------------------|

INTRODUCCIÓN



INTRODUCCIÓN.

En esta época en la que ya no se considera a la sustitución de importaciones como el motor de nuestro crecimiento industrial y que se piensa en otros factores, tales como la integración Latinoamericana, el fortalecimiento del mercado interno, el incremento del comercio exterior mediante la utilización plena de las capacidades instaladas y otros más; la Industria Eléctrica Nacional, consciente de que el proceso de normalización resulta indispensable no sólo para realizar y promover el comercio tanto interior como exterior, sino también para incrementar el nivel de productividad, ha venido trabajando tanto en la elaboración de normas, como en la difusión necesaria para su aplicación.

La normalización dentro del proceso de comercialización, se presenta como una actividad necesaria, ya que entre los beneficios que ofrece se encuentran: la unificación de criterios, homogenización de calidades de los productos, así como orientar la producción (sobre cuándo y qué producto se debe producir).

Por lo cual, resulta importante que las personas vinculadas a la comercialización, conozcan además de los beneficios, los conceptos, principios, clasificación, importancia, fundamentos, ventajas, procedimientos y estructuración de la normalización.

La Normalización, es y ha sido tarea que los Ingenieros de todo el mundo han venido realizando, principalmente en la rama Industrial y en forma muy especial en la Industria Eléctrica, que es sin lugar a dudas, donde la Normalización ya sea nacional o internacional es un factor determinante para el progreso.

Las normas de instalaciones eléctricas son un conjunto de disposiciones que tienen como propósito, regular por parte del Estado, los proyectos y la ejecución de las obras e instalaciones eléctricas; con el objetivo primordial de proteger la vida y las propiedades de las personas, al mismo tiempo que sirven de elemento normativo en las relaciones entre contratista y usuario, porque representan un nivel de calidad a satisfacer.

Por todo lo antes mencionado, se puede constatar la gran importancia que revisten dichas normas, y por ello es necesario automatizar su consulta; por una parte, para evitar las fallas humanas que siempre se presentan y más cuando se trata de ediciones muy extensas; por otra parte, para simplificar el trabajo que es más de oficina que ingenieril. La consulta por sí misma es una parte importante pero sencilla en el proceso de diseño de instalaciones eléctricas.

El análisis y diseño de sistemas es un proceso en el cual se aplican muchas herramientas para mejorar y simplificar el trabajo en las empresas de todos los sectores (científico, económico, social y hasta político) de un país. Todo esto mediante la implementación o actualización de los sistemas de información. Esta es la causa por la que los analistas de sistemas están obligados a entender al usuario potencial y a las computadoras, para lograr que la información sea útil en la toma de decisiones.

Sin duda alguna, vivimos en una sociedad en la que la mayoría de las personas se ocupan de manipular y transmittir la información. Así, además de producir bienes en masa, se produce ahora información en masa, y estos conocimientos son una de las fuerzas que impulsan la economía, medicina, educación, fabricación y gobierno.

En el presente trabajo, se analizarán tópicos relacionados con la normalización (como marco introductorio); además del análisis, diseño e implantación y mantenimiento de un sistema de consulta automatizado de la norma NOM-001-SEMP-1994 de Instalaciones Eléctricas.

PLANTEAMIENTO DEL PROBLEMA :

En la actualidad, el Departamento de Ingeniería Eléctrica de la Universidad Nacional Autónoma de México, necesita consultar continuamente la Norma Oficial Mexicana NOM-001-SEMP-1994 para Instalaciones Eléctricas, la cual tiene como objeto establecer las especificaciones de carácter técnico que deben satisfacer las instalaciones destinadas al suministro y uso de energía eléctrica, a fin de que ofrezcan condiciones adecuadas de servicio y seguridad para las personas y su patrimonio.

Esta consulta se lleva a cabo en un libro de gran volumen y con letras pequeñas, lo que origina que la consulta resulte muy lenta y tediosa.

Todo esto ocasiona que no se dedique una plena concentración a las tareas de ingeniería de instalaciones eléctricas, ya que se ocupa parte del tiempo en hacer una consulta que resulta sencilla pero muy tardada.

DEFINICIÓN DEL PROBLEMA :

Con base en lo anterior y debido al avance tecnológico en lo que a computación se refiere, en el Departamento de Ingeniería Eléctrica se hace indispensable implantar un sistema de cómputo para la consulta de esta norma.

HIPÓTESIS :

La sistematización de grandes volúmenes de información, es de gran importancia en esta época, ya que si se logra realizar un sistema de consulta de calidad, rápido, fácil de manejar y eficiente, se optimizará de gran manera el tiempo.

OBJETIVO GENERAL :

Investigar y realizar un sistema de cómputo para consulta de la norma NOM-001-SEMP-1994 para el Departamento de Ingeniería Eléctrica de la Facultad de Ingeniería de la UNAM. Esta sistematización tendrá como principal necesidad, optimizar el trabajo, así como la disminución del tiempo de consulta de dicha norma, con lo que se consigue aumentar la productividad del personal que opere el sistema.

OBJETIVO ESPECÍFICO :

Crear un sistema que pueda ser utilizado como apoyo didáctico en las materias de la carrera de Ingeniería Eléctrica.

JUSTIFICACIÓN :

La gran cantidad de información referente a normas eléctricas se encuentra en libros que son, en muchas ocasiones muy voluminosos y con letras no muy visibles. Esto lleva al Departamento de Ingeniería Eléctrica a buscar la manera de resolver este problema, y aunque a simple vista parezca de no mucha relevancia, este hace que se tenga una gran pérdida de tiempo, que pudiera ser aprovechado en otras labores ingenieriles.

Por lo cual se necesita de un sistema de consulta rápido, claro y eficiente, el cual pueda resolver los problemas de tiempo y logre una mayor productividad en dicho departamento.

PLAN DE TRABAJO :

El análisis para realizar un sistema de cómputo , necesariamente conlleva a un plan de trabajo.

El plan que se siguió fue el siguiente :

1.- Definir el problema.

2.- Fase de investigación : Investigar todo lo referente a la parte de historia de la normalización, antecedentes del sistema, evolución del software, diseño de sistemas y bases de datos .

Investigar a través de las diferentes fuentes de información, en este caso :

DOCUMENTAL : Basándose en libros , normas, manuales. folletos y publicaciones.

DE CAMPO : Realizando entrevistas al usuario final.

3.- Con base en la situación actual del problema realizar una planeación de tiempos de realización del sistema.

4.- Analizar el sistema y recurrir a métodos de análisis de requerimientos, para lograr una seguridad total del sistema y diseñar finalmente el sistema. Evaluar los diferentes lenguajes y paquetes de cómputo que podrían ser utilizados para la realización de este sistema.

5.- Captura de la norma y programación del sistema.

6.- Una vez diseñado el sistema, estudiar minuciosamente la manera de implantar dicho sistema y analizar todas las opciones de mantenimiento que se necesiten para que el sistema rinda al máximo.

7.- Partiendo de que el sistema funciona cubriendo todos los objetivos previstos por el usuario, proceder a una capacitación para el usuario y elaborar un manual de operaciones del sistema.

CAPÍTULO 1

CONCEPTOS
BÁSICOS



**CAPÍTULO 1
CONCEPTOS BÁSICOS.****1.1 DEFINICIONES DE NORMALIZACIÓN Y
CONSIDERACIONES SOBRE ESTE CONCEPTO.**

Aún cuando desde hace algún tiempo se habla y se escribe con cierta insistencia sobre el término normalización, en pocas ocasiones se ha tratado sobre su definición aún a pesar de que uno de los propósitos fundamentales de la normalización es precisamente el de definir.

En las siguientes líneas se indican brevemente algunas definiciones sobre los términos "normalización" y "norma" y algunos comentarios adicionales, con el propósito no de analizarlas, sino más bien, de hacer hincapié en su valor circunstancial.

A) CONCEPTO DE NORMALIZACIÓN.**DEFINICIÓN ADOPTADA POR LA ISO:**

La Organización Internacional de Normalización (ISO), organismo constituido por la mayor parte de los países que tienen una institución encargada del proceso de normalización a nivel nacional, tiene varios Comités Especiales que dependen directamente de su Consejo Directivo. Uno de estos comités es el Comité para el Estudio de los Principios Científicos de la Normalización (STACO), el cual está formado por expertos en materia de normalización que pertenecen a varios países e instituciones. Entre los primeros documentos que preparó este comité para someterlos a la aprobación del consejo de la ISO, incluyó las definiciones de normalización y de norma.



De acuerdo con la STACO el concepto de normalización es primero y luego el de norma; para STACO tiene primer lugar la acción, el proceso y luego su reconocimiento por una determinada institución, debido a esto, aprobó primeramente en 1961 la definición de normalización, y en 1962 la de norma.

La definición de normalización según la STACO es la siguiente:

"La normalización es el proceso de formular y aplicar reglas con el propósito de realizar un orden en una actividad específica, para el beneficio y con la cooperación de todos los intereses, y en particular para la obtención de una economía de conjunto óptima, teniendo en cuenta las características funcionales y los requisitos de seguridad".

La normalización se basa en los resultados consolidados de la ciencia, la técnica y la experiencia. Determina no solamente la base para el presente sino también para el desarrollo futuro y debe mantener su paso acorde con el progreso.

Algunas aplicaciones particulares son:

- 1) Unidades de medida.
- 2) Terminología y representación simbólica.
- 3) Productos y procesos (definición y selección de las características de productos, métodos de prueba y de medición, especificación de las características de los productos para definir su calidad, regulación de variedades, intercambiabilidad, etc.).
- 4) Seguridad de las personas y de los bienes.

El diccionario de la *Real Academia Española* presenta la siguiente definición:



NORMALIZACIÓN:

"Regularizar o poner en buen orden lo que no lo estaba".

El diccionario *LAROUSSE* lo define de la siguiente manera:

NORMALIZACIÓN:

"Reglamentación de las dimensiones y calidades de los productos industriales con el fin de simplificar y reducir los gastos de fabricación de los mismos".

Una definición de normalización que en general se acepta es:

"Elaboración y aplicación de normas"; siendo desde luego necesario conocer los conceptos de norma, elaboración de normas y aplicación de normas, mismos que más adelante se definirán.

Se puede decir que la normalización es el establecimiento de referencias para juzgar la calidad de los diversos productos o procesos que se encuentran ante la humanidad.

La normalización puede ser básica o industrial. La normalización básica es aquella que resulta de la investigación científica. La normalización industrial es el resultado de la concurrencia de las experiencias de todos los sectores que pudieran estar afectados por ella, o bien de todos los sectores que aparte de los primeros se interesaran en ella, para lograr la satisfacción de una necesidad común, tomando en consideración los medios con que se cuenta.

La normalización es un factor determinante para el progreso, sin duda porque la norma es la representación del nivel de desarrollo, y que conociéndola se puede optimizar definiendo técnicas mejores y logrando el uso más conveniente de materiales, productos, etc.

La normalización es la captación de la realidad, es imposible basarla en principios rígidos, establecidos a priori, que le quiten la necesaria flexibilidad para adaptarse a las necesidades, a la técnica y a la idiosincrasia nacional. La normalización debe ser considerada como una



gestión paralela al proceso de producción y al proceso de desarrollo de los productos.

B) CONCEPTO DE NORMA.

La definición de norma según la STACO es la siguiente:

*"Una **norma** es el resultado de una gestión particular de normalización, aprobada por una autoridad reconocida".*

Puede tomar la forma de:

- 1) Un documento que contiene un conjunto de condiciones por ser cumplidas.
- 2) Una unidad fundamental o una constante física, ejemplos: amperes, cero absoluto (kelvin).
- 3) Un objeto para comparación física.
Ejemplo: metro.

La definición que ofrece el diccionario de la Real Academia Española es la siguiente:

NORMA:

"Regla que se debe seguir o a que se deben ajustar las operaciones".

El diccionario LAROUSSE presenta la siguiente definición:

NORMA:

"Modelo a que se ajusta una fabricación".



Desde el punto de vista industrial, una norma es el resultado de la combinación de las experiencias para la resolución de una necesidad común.

El principio que sigue la norma es el que dice "las mismas necesidades humanas son resueltas en forma diversa según el grado de adelanto tecnológico y los materiales al alcance del hombre".

A continuación se agrega una definición de norma nacional que fue tomada de las normas que elabora el Comité Consultivo Nacional de Normalización de la Industria Eléctrica (CONNIE) establecido en México:

"La norma nacional debe ser la representación escrita del nivel de desarrollo tecnológico y científico alcanzado en el país, del producto, proceso o función, etc.; que ampara y es la solución o conjunto de soluciones que se han logrado para satisfacer cada una de las necesidades que se plantean".



1.2 ORIGEN Y EVOLUCIÓN DE LAS NORMAS.

1.2.1 ORIGEN.

Aún cuando existe la tendencia a creer que la normalización es una consecuencia del avance tecnológico y que se ha venido aplicando después de la revolución industrial, debe notarse que esta disciplina realmente empieza a manifestarse en los albores mismos de la civilización, cuando la convivencia humana necesitó de normas de vida, como un método para subsistir y conservar la especie; normas de comportamiento para evitar el peligro, de acción para la adquisición de alimentos y más adelante, el advenimiento de normas más evolucionadas de existencia, creó la necesidad de establecer reglas comunes de transacción, de valuación, de acción y de expresión.

Más adelante, las civilizaciones más antiguas de que se tiene conocimiento aplicaron la normalización en sus primitivas tecnologías y en sus intercambios comerciales.

Aunque la normalización industrial propiamente dicha, tuvo su origen hace tan solo algunas décadas como adelante se describirá:

1.2.2 HISTORIA DE LA NORMALIZACIÓN EN DIVERSAS NACIONES.

En 1901 nace en Inglaterra un organismo de normalización, que fue el que dió el primer paso en lo que a normalización nacional se refiere. aún cuando este organismo, funcionó como tal hasta 1918. Lo crearon varios institutos como el de Ingenieros Electricistas, Ingenieros Mecánicos, Ingenieros Civiles, etc. debido a que se percataron de la conveniencia de su establecimiento por el cambio del artesanado al proceso industrial, provocado por la revolución industrial y así en convenio con el gobierno quedó establecido el organismo que en 1931 adoptó el nombre de British



Standard Institution (BSI). Y que sería el que lograría la concurrencia de la experiencia lograda tanto en la fabricación y uso de los artículos producidos por los diferentes procesos industriales.

Siguieron a Inglaterra en lo que respecta al establecimiento de organismos dedicados a la normalización, Suiza, Bélgica, Austria, Finlandia, Francia y Dinamarca.

En Alemania se creó en 1917 un Comité de Normas para la Ingeniería Mecánica que se denominó Comité de Normas de la Industria Alemana. En 1926 adoptó el nombre con el que actualmente se conoce Deutschen Normenausschuss (DNA), que es hasta la fecha el organismo encargado de la normalización en este país.

En los Estados Unidos de Norteamérica se creó en 1918 la "American Standard Association" (ASA) conocidas hasta 1967 con este nombre y actualmente conocida como "United States American Standard Institute" (USASI); esta organización aún no era un organismo central de normalización sino uno de varios dedicados a este fin, pero a partir de que adoptó el nuevo nombre es el organismo oficial de normalización en este país.

En Francia, la Normalización Nacional se inició por un decreto gubernamental expedido en 1918, el que estableció la "Comisión Permanente de Normalización". Posteriormente se creó en 1926 la Asociación Francesa de Normalización (AFNOR), convirtiéndose en una asociación privada y en 1953 amplió sus estatutos para convertirla en un servicio público.

A nivel nacional latinoamericano la normalización se inició en Argentina, que en 1935 creó el Instituto Argentino de Normalización de Materiales (IRAM). Este instituto es un organismo privado con reconocimiento oficial del gobierno como organismo centralizador para el estudio de normas técnicas.

A la formación de la IRAM prosiguieron: en 1937 la Asociación Brasileña de Normas Técnicas (BNT), en 1939 el Instituto Uruguayo de Normas Técnicas (UNIT), en 1943 la Dirección General de Normas de México



(DGN), en 1958 la Comisión Venezolana de Normas Técnicas (COVENIN), en 1961 el Instituto Nacional de Normas Técnicas Industriales y Certificación del Perú (INANTIC), en 1962 el Departamento de Normas Técnicas, Dirección de Normas y Metalurgia del viceministerio para Desarrollo Técnico del ministerio de industrias de Cuba, en 1962 el Departamento de Normalización del Instituto Centroamericano de Investigación y Tecnología Industrial (ICAITI) que actúa en representación de todos los países centroamericanos, en 1964 el Instituto Colombiano de Normas Técnicas (ICOMTEC) y en 1965 el Instituto de Investigaciones Tecnológicas y Normalización del Paraguay (INTECNOR).

En resumen, la normalización nacional es llevada a cabo en casi todos los países actualmente, ya sea en una forma u otra, existen organismos nacionales que en mayor o menor grado han dado impulso a la normalización industrial a nivel nacional del país que se trate.

1.2.3 HISTORIA DE LA NORMALIZACIÓN INTERNACIONAL.

En 1926 se estableció la Federación Internacional de Asociaciones de Normalización (ISA), en donde las asociaciones nacionales de diecinueve países se agruparon.

La Federación Internacional tuvo como base el establecimiento de la cooperación a nivel internacional en el campo de la normalización y principalmente la unificación de las normas nacionales de los miembros que la componían.

En 1939, algunos de los miembros de la ISA se separaron y en 1942 oficialmente desapareció.

En 1944 renace la ISA por la intervención del Comité Coordinador de Normas de las Naciones Unidas (UNSCC) que contenía las organizaciones nacionales de dieciocho países aliados.



En 1946 los representantes de organismos nacionales de normalización y los representantes de la UNSCC se reunieron con el fin de discutir y aprobar la formación de una nueva organización cuyo objeto sería el de facilitar la coordinación internacional así como la unificación de normas industriales, a la vez que emitir recomendaciones concernientes a trabajos técnicos que esta organización hiciera.

Fue en esa reunión, donde veinticinco países participaron, que se creó la Organización Internacional de Normalización (ISO) que tomaría el lugar de la ISA creada anteriormente.

El 24 de octubre de 1946 se aprobó en Londres, Inglaterra, la constitución y el reglamento de la ISO y se acordó que podría funcionar oficialmente hasta que quince organizaciones miembro lo ratificaran, lo que sucedió hasta el 23 de febrero de 1947.

Las funciones de la ISO son las de procurar el desarrollo de las normas en el mundo, para facilitar el intercambio internacional de mercancías y servicios, y desarrollar la mutua cooperación en las actividades intelectuales, tecnológicas, científicas y económicas. Para lograr estos fines, usa entre otros los siguientes medios:

- 1.- Facilitar la unificación y coordinación de normas nacionales y la emisión de las recomendaciones necesarias a los comités miembros para este propósito.
- 2.- Establecer normas internacionales, a condición de que ninguno de los comités miembros presente oposición en cada caso particular.
- 3.- Fomentar y facilitar, cuando la ocasión así lo demande, la creación de nuevas normas que tengan requisitos comunes susceptibles de ser utilizados dentro del dominio nacional e internacional.
- 4.- Organizar el intercambio de información relacionado con los trabajos de las organizaciones miembro y de los comités técnicos.



5.- Cooperar con otras organizaciones internacionales interesadas en materias conexas, particularmente los trabajos de normalización susceptibles de facilitar su tarea.

La Comisión Electrotécnica Internacional (IEC), fue integrada en el año de 1906, durante el Congreso Internacional de Electrotécnica celebrado en EUA. Esta organización creada para examinar el problema de la unificación de la nomenclatura de la clasificación de aparatos y máquinas eléctricas, fue incluida en la ISO en 1947 y aunque hasta la fecha conserva su autonomía, trabaja como la división eléctrica de la ISO.

En forma similar a la ISO, la IEC fundamenta sus trabajos técnicos en comités técnicos, tratando cada uno una determinada materia.

Tanto la ISO como la IEC elaboran recomendaciones como proyectos de normas internacionales; pero no elaboran hasta la fecha normas definitivas, debido entre otras a las siguientes causas:

- 1.- Existen condiciones naturales que impiden, en muchas ocasiones el poder aceptar las recomendaciones como normas, mencionándose la altitud, temperatura y condiciones climatológicas.
- 2.- El desarrollo industrial de un país no permite llegar al nivel técnico-industrial propuesto por países altamente desarrollados.
- 3.- La maquinaria y equipo que se tiene permite elaborar productos en sistemas de medida diferentes a los especificados por las recomendaciones internacionales, y
- 4.- Falta de participación efectiva dentro de la ISO e IEC.



1.2.4 HISTORIA DE LA NORMALIZACIÓN EN MÉXICO.

La necesidad de contar, pesar, medir, la ha satisfecho el hombre de diversas maneras desde los tiempos prehistóricos y también desde entonces enfrentó las dificultades surgidas de la variedad de tipos de medida o de la desigual medición aún usada la misma unidad.

En México, desde la época prehispánica hasta nuestros días, pesar y medir ha sido siempre función del Estado. Entre los aztecas fueron los jueces quienes la ejercieron en los mercados; en la Colonia esta labor correspondió al "Fiel Contraste". Ahora compete a la Secretaría de Industria y Comercio realizar estas funciones.

El 15 de marzo de 1857, el presidente Ignacio Comonfort expidió el decreto que estableció definitivamente la adopción del Sistema Métrico Decimal, sistema aceptado por la mayoría de los países del mundo gracias a las excelencias de sus fundamentos científicos y a la facilidad de su manejo.

En enero de 1943, el antiguo Departamento de Pesas y Medidas fue transformado en la actual Dirección General de Normas, con el propósito de elaborar en el menor tiempo posible las normas industriales destinadas a reglamentar la producción.

En la misma fecha en que se fundó la nueva dirección, se creó el Departamento de Normalización, encargado de estudiar, discutir, formular y aprobar las normas que rigen la calidad, el funcionamiento y el lenguaje técnico industrial a que deben sujetarse los productos industriales.

Las actividades de este departamento comprendieron la realización de trabajos técnicos e investigaciones económicas y sociales, así como el establecimiento inmediato de relaciones con productores y consumidores a fin de mantener acorde su acción con la realidad de ambos sectores. En igual forma buscó coordinarse con organismos nacionales e internacionales dedicados al estudio de los problemas relativos a la normalización y la metrología.



El Gobierno Mexicano, consciente de que el comercio exterior constituye un factor de primordial importancia para el desarrollo de nuestra economía, un factor que nos ofrece enormes posibilidades de mejoramiento, instituyó la Dirección General de Normas como único organismo capacitado para diseñar una política adecuada en materia de normalización industrial. Sus objetivos básicos primarios consistieron en el establecimiento de contactos con la iniciativa privada para unificar criterios en la elaboración de especificaciones sobre productos industriales, así como para fijar patrones que garantizaran los requisitos exigidos por los sectores de amplio consumo nacional.

Así fue como surgieron los primeros proyectos de normas industriales, proyectos que se vieron reforzados, desde el punto de vista legal, con expedición de la Ley de Normas Industriales (31 de diciembre de 1945). Esta ley sentó las bases para el establecimiento de los contactos necesarios con los organismos nacionales de normalización y fue publicada el 11 de febrero de 1946 en el Diario Oficial de la Federación.

En diciembre de 1958, un Decreto Presidencial transformó a la antigua Secretaría de Economía en la actual Secretaría de Industria y Comercio, y el 7 de abril de 1961 se publicó en el Diario Oficial de la Federación la Ley de Normas Pesas y Medidas, actualmente en vigor.

La Dirección General de Normas es miembro de la Organización Internacional de Normalización (ISO) y de la Comisión Panamericana de Normas Técnicas (COPANT); representa al Gobierno de los Estados Unidos Mexicanos en la Comisión del CODEX Alimentarius, creada en 1962 por la organización de las Naciones Unidas; también representa a México en organismos afines de otros países.

Así mismo, representa a México, en los organismos Nacionales e Internacionales de Normalización, manteniendo con ellos intercambio permanente y recíproco de normas, recomendaciones y publicaciones, y está autorizada por los mismos para actuar como intermediaria en la venta de sus publicaciones.

Con plena consciencia de que no debe quedar al margen ni a la zaga de la intensa labor que realizan los diferentes organismos internacionales de



normalización. La Dirección General de Normas envía a sus directivos y técnicos a las reuniones especializadas que organizan los países miembros, con instrucciones de participar activa y firmemente. Considerando que, en un momento dado, pudierase adoptar decisiones lesivas a los intereses de México.

FUNCIONES DE LA DIRECCIÓN GENERAL DE NORMAS.

Para lograr las finalidades que le han sido asignadas, la Dirección General de Normas realiza las siguientes funciones:

- Normaliza los productos industriales a fin de garantizar una calidad sostenida en la que confíen el consumidor nacional y el extranjero.
- Otorga la autorización, para el uso del Sello Oficial de Garantía, a productos sujetos a Norma Oficial de Calidad.
- Revisa sistemáticamente los instrumentos de pesar y medir para asegurar que su uso sea correcto y para evitar posibles fraudes en grande y pequeña escala.
- Estudia e investiga permanentemente la realidad nacional para hacer factible los patrones que establecen las normas y para lograr su renovación. Bien por exigencias del consumidor o por avances técnicos o industriales.
- Vigila la fabricación y uso de los instrumentos de pesar y medir.
- Protege los intereses del público consumidor mediante una labor permanente de orientación y supervisión.
- Opina sobre la aplicación de sanciones y multa en caso de infracciones o falta de cumplimiento a los reglamentos establecidos.



- Establece soluciones o acuerdos con instituciones internacionales afines, asiste y participa en las reuniones convocadas por dichos organismos.

1.3 NORMA OFICIAL MEXICANA (NOM).

DEFINICIÓN:

La Norma Oficial Mexicana (**NOM**) es la regulación obligatoria que contiene características que deben cumplir aquellos productos y procesos cuando estos puedan constituir un riesgo para la seguridad de las personas o dañar la salud humana, animal o vegetal; el medio ambiente o causar daños en la preservación de nuestros recursos naturales.

La Ley Federal sobre Metrología y Normalización, publicada en el Diario Oficial de la Federación el 1 de julio de 1992, es el marco jurídico que reglamenta la expedición y cumplimiento de la Norma Oficial Mexicana.

EMISIÓN DE LA NORMA OFICIAL MEXICANA (NOM).

La Norma Oficial Mexicana, es emitida por las dependencias de la Administración Pública Federal, según su ámbito de competencia, a través de un procedimiento claro, confiable y coordinado. En su elaboración participan los sectores público y privado: dependencias de la Administración Pública Federal, industriales, comerciantes, investigadores y consumidores.

Para fortalecer la estructura nacional de expedición de la NOM se creó la Comisión Nacional de Normalización que se reúne periódicamente para coordinar los esfuerzos de las dependencias y aprobar el Programa Nacional de Normalización.



Para obtener información veraz sobre las reglamentaciones obligatorias que se proponen establecer, es necesario consultar el Programa Nacional de Normalización y recurrir, constantemente, a este medio impreso, para tener conocimiento de los proyectos de NOM, su vigencia, texto y fecha de publicación definitiva.

PROCEDIMIENTO PARA PUBLICAR LA NORMA OFICIAL MEXICANA (NOM).

Para publicar la Norma Oficial Mexicana se lleva a cabo el siguiente procedimiento:

- 1.- Se elabora el anteproyecto de NOM de acuerdo al Programa Nacional de Normalización.
- 2.- Los anteproyectos correspondientes los elabora la dependencia que le compete según su área.
- 3.- El documento obtenido se presenta al Comité Consultivo Nacional de Normalización (C.C.N.N.), para que en un plazo no mayor a los 75 días naturales, formule observaciones.
- 4.- La dependencia que elaboró el anteproyecto contestará fundadamente las observaciones presentadas por el C.C.N.N. en un plazo no mayor de 30 días naturales.
- 5.- Se publicará íntegramente en el Diario Oficial de la Federación a efecto que dentro de los siguientes 90 días naturales, los interesados presenten sus comentarios al C.C.N.N. correspondiente.
- 6.- Al término del plazo el C.C.N.N. correspondiente estudiará los comentarios recibidos, en su caso modificará el proyecto en un plazo que no exceda los 45 días naturales.
- 7.- Las respuestas y comentarios se publicarán con anterioridad a la NOM.



8.- Una vez aprobadas por la C.C.N.N. respectiva, las NOM serán expedidas por la dependencia competente y publicadas en el Diario Oficial de la Federación.

VENTAJAS QUE OFRECE LA NOM AL PRODUCTOR.

- ♦ Mayor transparencia al comercio en territorio nacional.
- ♦ Mayor competitividad.
- ♦ Disminuir el rechazo de productos.
- ♦ Favorecer a prácticas equitativas de comercio.
- ♦ Facilitar la venta de productos.
- ♦ Impedir importar productos de mala calidad.

VENTAJAS QUE OFRECE LA NOM AL CONSUMIDOR.

- ❖ Garantiza la calidad requerida en los productos que ostenten el sello NOM.
- ❖ Ayuda en la elección de mejores productos.
- ❖ Aumenta la seguridad y el buen funcionamiento de los productos.



CUMPLIMIENTO DE LA NOM.

Corresponde a las dependencias de la Administración Pública Federal, según su ámbito de competencia, certificar y verificar que los productos, procesos, métodos, instalaciones, servicios o actividades cumplan con la regulación obligatoria. Para efectuar una observancia ordenada y eficaz, las dependencias competentes ajustarán, conjuntamente, sus procedimientos de certificación y verificación a los lineamientos que dicte la Comisión Nacional de Normalización, buscando coordinarse para el desarrollo de dichas actividades.

CLASIFICACIÓN DE LAS NORMAS.

Las normas, por decreto de ley, se clasifican en opcionales y obligatorias.

A) NORMAS OBLIGATORIAS.

Son aquellas que rigen el sistema de Pesas y Medidas; las industriales que a criterio de la Secretaría de Comercio y Fomento Industrial (SECOFI) fije a los materiales, procedimientos o productos que ponen en peligro la vida, la seguridad o la integridad corporal de las personas; del mismo modo se aplicará la obligatoriedad de las normas, a juicio de SECOFI, a las mercancías objeto de exportación, así como aquellas de consumo nacional que específicamente señale la Secretaría cuando lo requiera la economía del país o el interés público.



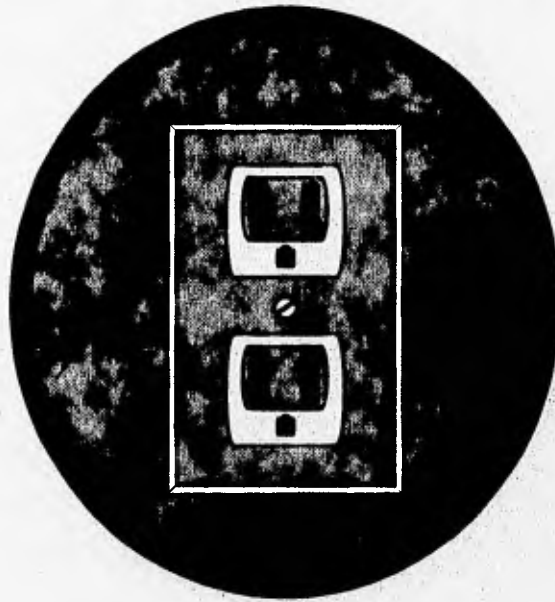
B) NORMAS OPCIONALES.

Serán aquellas que satisfagan los requisitos que establece SECOFI, para que los solicitantes obtengan la autorización de colocar en sus productos, el Sello Oficial de Garantía. Estas normas son las que se utilizan en la fabricación de productos o materiales que no ponen en peligro la vida o salud de las personas e igualmente su empleo queda a criterio del fabricante.



CAPÍTULO 2

NORMA OFICIAL
MEXICANA
NOM-001-SEMP-1994



CAPÍTULO 2**NORMA NOM-001-SEMP-1994.****2.1 ANTECEDENTES:****2.1.1 NORMALIZACIÓN DE LA INDUSTRIA ELÉCTRICA EN MÉXICO.**

El Comité Consultivo Nacional de Normalización de la Industria Eléctrica (CCONNIE) es en México el organismo encargado de la normalización de productos o procesos relacionados con la Industria Eléctrica. Este organismo que se constituyó el 4 de octubre de 1965 con el reconocimiento de la Dirección General de Normas empezó a funcionar hasta el 1 de febrero de 1966, fecha en que se terminó de estudiar la organización que debería adoptarse para su funcionamiento.

Los fines para los que el CCONNIE se estableció se pueden resumir en los siguientes:

- 1.- Planeación de la normalización nacional de la industria eléctrica.
- 2.- Coordinación de la normalización nacional de la industria eléctrica.
- 3.- Difusión de la normalización nacional de la industria eléctrica.
- 4.- Normalización internacional de la industria eléctrica.

El Consejo Directivo del CCONNIE está integrado por Representantes de los Sectores Fabricante, Consumidor, Colegiado y Oficial, así como con representantes asesores de Asociaciones o Institutos reconocidos, establecidos en México.



El Sector Fabricante está representado por la Cámara Nacional de Manufacturas Eléctricas; el Sector Consumidor por la Comisión Federal de Electricidad y la Compañía de Luz y Fuerza del Centro, S.A.; el Sector Colegiado por el Colegio de Ingenieros Mecánicos Electricistas; el Sector Oficial por la Dirección General de Normas y la Dirección General de Electricidad, ambas de la Secretaría de Industria y Comercio.

2.1.2 LEGISLACIÓN EN INSTALACIONES ELÉCTRICAS.

En nuestro país, el área industrial es uno de los sectores básicos de la economía nacional, y en la actualidad la calidad resulta ser un elemento imprescindible para competir con productos y servicios a nivel mundial.

Es responsabilidad del gremio electromecánico, vigilar que se garanticen las condiciones necesarias para lograr esa calidad, entre esas condiciones podemos distinguir dos de las principales que son, disponibilidad y continuidad del servicio de energía eléctrica.

Para lograr los objetivos anteriores, se requiere contar con instalaciones eléctricas adecuadas que aseguren un máximo grado de eficiencia tanto de estas, como de la maquinaria y equipo a los que sirven.

Es de suma importancia que los ingenieros y técnicos que se dedican a las actividades de diseño, proyecto, montaje, operación, conservación y mantenimiento de instalaciones eléctricas, conozcan y apliquen los requisitos mínimos de seguridad que establece la legislación vigente en esta materia.

Como una reseña histórica de la legislación en materia de obras e instalaciones eléctricas, se presentará un resumen de los cambios y modificaciones ocurridos hasta la fecha así como los beneficios obtenidos en el campo de la ingeniería eléctrica.

La Ley del Servicio Público de Energía Eléctrica (LSPEE) fue creada por decreto presidencial en el año de 1975, derogando a la entonces Ley de la Industria Eléctrica de 1945, en dicha ley se hace referencia al Reglamento de Obras e Instalaciones Eléctricas (ROIE) que se publicó el 31 de marzo de 1950; los cambios y avances en esta materia obligaron a efectuar una actualización de dicha legislación, por lo que el 22 de junio de 1981 se publica el nuevo Reglamento de Instalaciones Eléctricas (RIE), derogando al ROIE; con la publicación de este nuevo reglamento y basado en el Código Nacional Eléctrico de los Estados Unidos (NEC) de 1978, aparecen las Normas Técnicas para Instalaciones Eléctricas (NTIE), mismas que no han sufrido actualización alguna, únicamente se han elaborado tres acuerdos y un adendum que aparecieron en el Diario Oficial de la Federación en las siguientes fechas: 18 de junio de 1984, adendum del 22 de abril de 1985, 30 de junio de 1986 y 9 de mayo de 1988.

Actualmente y con la modificación a la Ley del Servicio Público de Energía Eléctrica, corresponde a la Secretaría de Energía Minas e Industria Paraestatal (SEMIP), vigilar que dicha legislación se cumpla, atribución que anteriormente le correspondía a la SECOFI.

A continuación se analizarán algunas legislaciones eléctricas de relevancia:

LEY DEL SERVICIO PÚBLICO DE ENERGÍA ELÉCTRICA (LSPEE).

Esta legislación se aplica para dar cumplimiento a los requisitos que las instalaciones eléctricas deben cumplir para brindar seguridad y el servicio adecuado. Esta ley fue modificada el 23 de diciembre de 1992, derogando a la publicada en el Diario Oficial de la Federación el 27 de diciembre de 1983. Cabe señalar que era urgente la necesidad de llevar a cabo dichas modificaciones ya que los cambios y avances en materia de Instalaciones eléctricas lo requieren en estos momentos en que la vida industrial de nuestro país está llegando a un punto de gran importancia para su avance.



REGLAMENTO DE LA LEY DEL SERVICIO PÚBLICO DE ENERGÍA ELÉCTRICA.

El Reglamento de la Ley de la Industria Eléctrica fue publicado en el Diario Oficial de la Federación el 4 de octubre de 1945, mismo que abrogó el del 28 de agosto de 1940.

El Reglamento de la Ley del Servicio Público de Energía Eléctrica, de las fechas citadas, fue reformado y publicado en el Diario Oficial de la Federación el 31 de mayo de 1993.

El objetivo de este ordenamiento es reglamentar la Ley del Servicio Público de Energía Eléctrica en lo que se refiere a la prestación de dicho servicio y a las actividades previstas en la propia ley que no constituyen servicio público.

LEY FEDERAL SOBRE METROLOGÍA Y NORMALIZACIÓN (LFMN).

La Ley Federal sobre Metrología y Normalización (publicada en el Diario Oficial de la Federación el 1 de julio de 1992), fija los requisitos e indicaciones para formar parte de las unidades de verificación.

REGLAMENTO DE INSTALACIONES ELÉCTRICAS (RIE).

Considerando la avanzada evolución tecnológica en la industria eléctrica del país, se hizo necesario contar con las disposiciones legales en materia de instalaciones eléctricas, por lo que, siendo imperiosa la necesidad del momento en precisar los requisitos generales referentes a las instalaciones eléctricas para salvaguardar la seguridad de los usuarios y sus



pertenenias, así como de que se establecieran las bases que fijaran la formulación y expedición de las Normas Técnicas que regularan dichas instalaciones; normas que, por otra parte, deberán estar sujetas a una revisión continua, a fin de mantenerse actualizadas y acordes con los logros alcanzados por la técnica moderna en esta área, se expidió el Reglamento de Instalaciones Eléctricas y que apareció publicado en el Diario Oficial de la Federación el 22 de junio de 1981.

Este reglamento tiene por objeto establecer los requisitos que deben satisfacer las instalaciones destinadas al suministro y uso de la energía eléctrica, a fin de que ofrezcan condiciones adecuadas de servicio y seguridad para las personas y su patrimonio.

NORMAS TÉCNICAS PARA INSTALACIONES ELÉCTRICAS (NTIE81).

Para llevar a cabo el proyecto y construcción de instalaciones eléctricas, es importante contar con los requisitos mínimos de seguridad para tal efecto, por lo que el Reglamento de Instalaciones Eléctricas hace referencia al uso de las Normas Técnicas para Instalaciones Eléctricas (NTIE), publicadas en el Diario Oficial de la Federación en el año de 1981, siendo emitidas por la Dirección General de Normas (DGN) de la Secretaría de Patrimonio y Fomento Industrial (actualmente Secretaría de Comercio y Fomento Industrial SECOFI).

El objetivo primordial de las Normas Técnicas es la protección de la vida y las propiedades de las personas contra los riesgos que representan el uso y el suministro de la energía eléctrica. Sus requisitos deben considerarse como requisitos mínimos de seguridad y, en el caso general, su cumplimiento permite obtener un servicio satisfactorio; pero estos requisitos no necesariamente representan las condiciones óptimas de servicio; con frecuencia es recomendable usar valores y diseños más amplios para tener una mejor calidad de servicio y prever aumentos de carga.



CÓDIGO NACIONAL ELÉCTRICO (NEC-93).

Todas las recomendaciones acerca del uso y seguimiento de la legislación en materia de instalaciones eléctricas son simplemente los requisitos mínimos de seguridad para el uso y manejo de la energía eléctrica, sin embargo existe la posibilidad de que dichos requisitos mínimos puedan ser superados, en cuanto a seguridad se refiere, basándonos también en otros tipos de legislación internacional o reglamentación existentes, como por ejemplo el Código Nacional Eléctrico (NEC) de los Estados Unidos de Norteamérica, en su edición más reciente publicada en 1993. Desafortunadamente, este Código es aplicable a las condiciones y necesidades de instalación que requieren en este país por sus adelantados procesos y desarrollos tecnológicos, por lo que su aplicación en México se verá muy limitada.

El propósito de este Código es la salvaguarda práctica de la integridad física de las personas y propiedades contra los peligros y daños que puedan ser provocadas por el uso de la energía eléctrica.

2.2 EXPEDICIÓN DE LA NORMA NOM-001-SEMP-1994.

El 15 de octubre de 1993 se expidió la Norma Oficial Mexicana de Emergencia NOM-EM-001-SEMP-1993, que regula las instalaciones destinadas al suministro y uso de la energía eléctrica.

Con fundamento en lo dispuesto por el artículo 48 de la Ley Federal sobre la Metrología y Normalización con fecha 15 de abril de 1994 se publicó el Acuerdo por el cual se reexpidió la Norma Oficial Mexicana de Emergencia.

El 9 de mayo de 1994 se publicó el Proyecto de Norma Oficial Mexicana NOM-001-SEMP-1994, relativa a instalaciones destinadas al suministro y uso de la energía eléctrica, y de conformidad con el artículo 47 de la Ley Federal sobre Metrología y Normalización inició el plazo de consulta pública, para que en los 90 días naturales siguientes, los interesados presentaran sus comentarios ante el Comité Consultivo Nacional de Normalización de Instalaciones Eléctricas.



El 30 de septiembre de 1994, como resultado del análisis efectuado por el Comité Consultivo Nacional de Normalización de Instalaciones Eléctricas, fueron publicados los comentarios y respuestas al Proyecto de Norma Oficial Mexicana, NOM-001-SEMP-1994, relativa a instalaciones destinadas al suministro y uso de energía eléctrica, y en consecuencia se hicieron las modificaciones procedentes a dicho proyecto.

De esta forma, en atención de contar con el instrumento normativo que regule las instalaciones eléctricas en forma permanente para salvaguardar la seguridad de los usuarios y sus pertenencias; se expidió la Norma Oficial Mexicana NOM-001-SEMP-1994, relativa a instalaciones destinadas al suministro y uso de la energía eléctrica.

2.3 OBJETIVOS DE LA NOM-001-SEMP-1994.

La presente Norma Oficial Mexicana tiene por objeto establecer las especificaciones de carácter técnico que deben satisfacer las instalaciones destinadas al suministro y uso de energía eléctrica, a fin de que ofrezcan condiciones adecuadas de servicio y seguridad para las personas y su patrimonio.

Suplir a la Norma Oficial Mexicana de Emergencia NOM-EM-001-SEMP-1993, cuya prórroga concluye el 15 de octubre de 1994.

2.4 CAMPO DE APLICACIÓN DE LA NOM-001-SEMP-1994.

El campo de aplicación de la presente Norma Oficial Mexicana para Instalaciones Eléctricas será:

a) Las instalaciones que se emplean para la utilización de la energía eléctrica, en cualquiera de las tensiones usuales de operación, incluyendo la instalación del equipo conectado a las mismas por los usuarios.



- b) Las subestaciones y las plantas generadoras de emergencia propiedad de los usuarios.
- c) Las líneas eléctricas y su equipo. Dentro del término "líneas eléctricas" quedan comprendidas las aéreas y las subterráneas conductoras de energía eléctrica, ya sea que formen parte de sistemas de servicio público o bien correspondan a otro tipo de instalaciones.
- d) Cualesquiera otras instalaciones que tengan por finalidad el suministro y uso de la energía eléctrica.

2.5 REFERENCIAS.

Para la correcta aplicación de esta norma es necesario consultar las siguientes Normas Oficiales Mexicanas y Normas Mexicanas vigentes:

- NOM-EM-002-SCFI: Productos eléctricos - conductores, alambres y cables - Especificaciones de seguridad y métodos de prueba.
- NOM-003-SCFI: Requisitos de seguridad en aparatos electrodomésticos y similares.
- NOM-008-SCFI: Sistema general de unidades.
- NOM-024-SCFI: Información comercial -aparatos electrónicos, eléctricos y electrodomésticos- instructivos y garantías para los productos de fabricación nacional e importada.
- NOM-050-SCFI: Información comercial -información comercial del envase o su etiqueta que deberán ostentar los productos de fabricación nacional y extranjera.



- NMX-B-208: Industria Siderúrgica -tubos de aceros para la protección de conductores eléctricos (tubo conduit) tipo pesado.
- NMX-B-209: Tubos de acero para la protección de conductores eléctricos (tubo conduit) tipo semipesado.
- NMX-B-210: Tubos de acero para la protección de conductores eléctricos (tubo conduit) tipo ligero y extraligero.
- NMX-J-10: Productos eléctricos - conductores - con aislamiento termoplástico a base de policloruro de vinilo, para instalaciones hasta de 600 v.
- NMX-J-98: Tensiones normalizadas.
- NMX-J-294: Productos eléctricos -conductores- resistencia de aislamiento -método de prueba-.

2.6 GENERALIDADES.

Esta Norma Oficial Mexicana, consta de una introducción y 14 capítulos subdivididos en dos partes, en la primera, se establecen disposiciones técnicas que deben observarse en las instalaciones eléctricas, de aplicación general, para locales, equipos y condiciones especiales, en sistemas de comunicación y de alumbrado público, incluyéndose un capítulo de tablas.

En la segunda parte, se incluyen las disposiciones técnicas que se deben aplicar a la instalación de subestaciones, de líneas eléctricas de suministro público, transportes eléctricos y otras líneas eléctricas y de comunicación ubicadas en la vía pública, así como a instalaciones similares propiedad de los usuarios, lo cual se ha establecido, considerando, en principio que dichas líneas estarán operadas y mantenidas por personal idóneo.



En el desarrollo de todos los capítulos de esta Norma Oficial Mexicana para instalaciones eléctricas, se ha contado con la valiosa colaboración de personal técnico especializado en cada uno de los temas que se tocan en la misma, que con su esfuerzo han contribuido al logro de la presente edición. Dicho personal continuará colaborando en los grupos de trabajo que conforman el COMITÉ CONSULTIVO NACIONAL DE NORMALIZACIÓN PARA INSTALACIONES ELÉCTRICAS (CCNNIE), para que se de vigencia tecnológica a esta Norma.

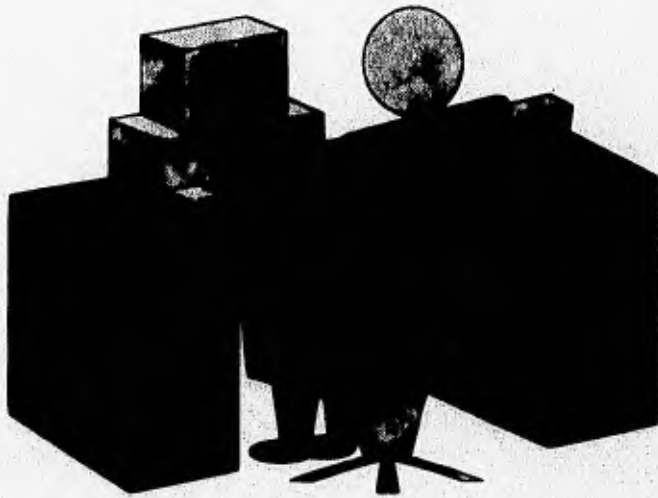
Es importante hacer notar que debido a la magnitud del trabajo, es posible que esta versión carezca de algunos conceptos importantes o que en su contenido exista información que se pueda considerar innecesaria.

Se hace mención especial al capítulo 9 de la norma referente al tema de ALUMBRADO PÚBLICO, el cual es un aspecto que por primera ocasión se logra incluir en una Norma Oficial. Debido a lo especializado del tema se elaboró complementándolo con información básica necesaria para su mayor comprensión, tratando de fijar los parámetros que se considera deben cumplirse en este campo para proporcionar la seguridad requerida.



CAPÍTULO 3

ANTECEDENTES
DEL
SISTEMA

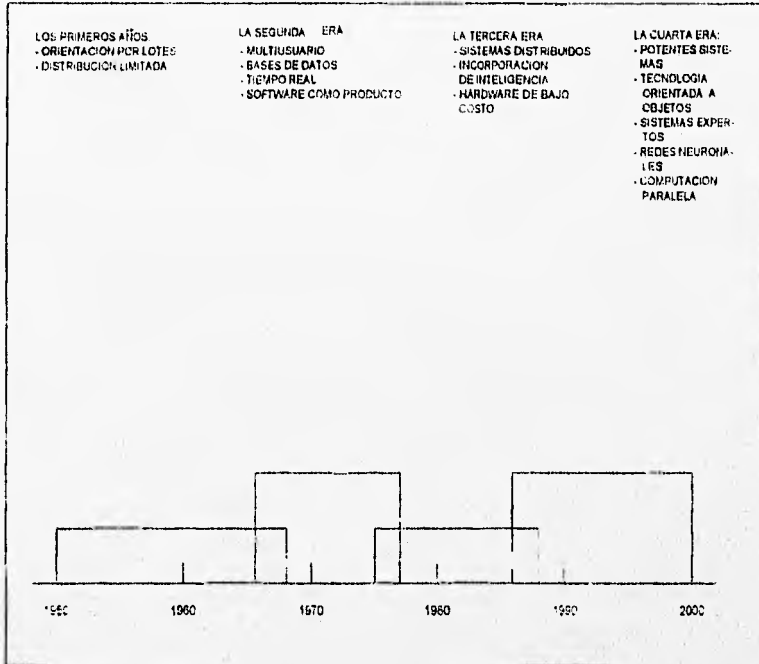


CAPÍTULO 3
ANTECEDENTES DEL SISTEMA.**3.1 SÍNTESIS EVOLUTIVA DEL SOFTWARE.****INTRODUCCIÓN:**

Durante las tres primeras décadas de la informática, el principal desafío era el desarrollo de hardware de las computadoras, de modo que se redujera el costo del procesamiento y almacenamiento de datos. A lo largo de la década de los ochentas los avances de la microeléctrica han dado como resultado una mayor potencia de cálculo y una reducción de costo. Ahora el desafío es mejorar la calidad de las soluciones basadas en las computadoras (soluciones que se implementan con el software). El software es el mecanismo que nos facilita utilizar y explotar el potencial que genera al hardware tanto en el almacenamiento como en la capacidad de procesamiento.

Un mejor rendimiento del hardware, una reducción de tamaño y un costo más bajo, han dado lugar a sistemas informáticos más sofisticados. La figura siguiente describe la evolución del software dentro del contexto de las áreas de aplicación de los sistemas basados en computadoras.





Durante los primeros años de desarrollo de las computadoras el hardware sufrió continuos cambios mientras que el software era considerado un complemento. El desarrollo de software se desarrollaba virtualmente sin ninguna planificación. hasta que los costos comenzaron a crecer, en este período se utilizaban en la mayoría de los sistemas una orientación por lotes, la mayor parte del hardware se dedicaba a la ejecución de un único programa que a su vez se dedicaba a una aplicación específica.

La mayoría del software se desarrollaba y era utilizado por la misma persona u organización; debido a este entorno personalizado del software la documentación normalmente existía.



La segunda era en la evolución de los sistemas de computadoras se extiende desde la mitad de la década de los 60's hasta finales de los 70's. La multiprogramación y los sistemas multiusuarios introdujeron nuevos conceptos de interacción hombre-máquina abriéndose así un nuevo mundo de aplicaciones y nuevos niveles de sofisticación del hardware y software. Los sistemas de tiempo real podrán recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en lugar de minutos. Los avances en los dispositivos de almacenamiento, condujeron a la primera generación de BASES DE DATOS; el software se desarrollaba para tener amplia distribución en un mercado multidisciplinar. Los programas se distribuían para computadoras grandes y minicomputadoras a miles de usuarios, conforme crecían el número de sistemas informáticos comenzaron a extenderse las bibliotecas de software por computadora.

Los productos de software comprados al exterior incorporaban varios miles de líneas de programación, todos estos programas fuente tenían que ser corregidos cuando se detectaban fallos modificados cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos de hardware que se hubiera adquirido. Estas actividades se llamaron colectivamente mantenimiento de software.

La tercera era de la evolución de los sistemas de computadoras comenzó a mediados de los 70's y terminó en los 80's; con el procesamiento distribuido, múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose una con otra, incremento la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y creciente demanda de acceso instantáneo a los datos, ejercieron una fuerte presión sobre los desarrolladores de software. Esta era se caracteriza también por la llegada y el amplio uso de microprocesadores y las computadoras personales. El hardware de las computadoras personales se ha convertido rápidamente en un producto estándar mientras que el software que se suministra con ese hardware es lo que marca la diferencia.



La cuarta era del software de computadoras comenzó en los 80's, con la programación orientada a objetos se esta desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación. los sistemas expertos en la inteligencia artificial se han trasladado de laboratorio a las aplicaciones prácticas para un amplio rango de problemas del mundo real.

3.2 CUALIDADES DEL SOFTWARE.

El software es un sistema lógico, en lugar de físico, por lo tanto, tiene unas características considerablemente distintas a las del hardware.

1. El software se desarrolla, no se fabrica. En sentido clásico, en el desarrollo del software la buena calidad se adquiere mediante un buen diseño. En la fase de construcción de hardware se pueden introducir problemas de calidad que no existen en el software. Los costos de software se encuentran en la ingeniería. esto significa que los proyectos de software no se pueden gestionar como si fueran problemas de fabricación.

2. El software no se estropea. Cuando el tiempo pasa, las fallas se presentan a medida que los componentes de hardware sufren los efectos acumulativos del polvo, la vibración, los malos tratos, las temperaturas extremas y otros males externos.

El software no es susceptible a los males del entorno que hacen que el hardware se estropee, los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen, suponiendo que no existan nuevos errores, este ya no fallará. Pero cuando el software sufre cambios (mantenimiento) es bastante probable que se introduzcan nuevos defectos y lentamente el nivel mínimo de fallas comienza a crecer, así que el software se va deteriorando debido a los cambios.



La diferencia es que cuando un componente de hardware se estropea, se sustituye por una pieza repuesto; pero para el software no existen tales piezas de repuesto. Cada falla en el software indica un error en el diseño o en el proceso mediante el que se tradujo el diseño al código máquina ejecutable. Por lo tanto el mantenimiento de software es mucho más complejo que el de hardware.

3. La mayoría de software se construye a la medida, en vez de ensamblar componentes existentes como en hardware. No existen catálogos de componentes de software. Se puede comprar software ya desarrollado pero solo como una unidad completa, no como componentes que puedan reensamblarse en nuevos programas, excepto por las primeras implementaciones de programación orientada a objetos.

Los componentes de software se crean mediante una serie de traducciones que hacen corresponder los requisitos del cliente con un código ejecutable en la máquina. Se traduce el modelo de requisitos a un diseño, se traduce el diseño del software a una forma en un lenguaje que especifique las estructuras de datos. La forma en lenguaje es procesada por un traductor que la convierte en instrucciones ejecutables en la máquina.

Los componentes de software se construyen mediante un lenguaje de programación que tiene un vocabulario limitado, una gramática definida explícitamente y unas reglas bien formadas de sintaxis y semántica. Estos atributos son esenciales para la traducción por la máquina. Las clases de lenguajes que se utilizan actualmente son : los Lenguajes Máquina, los Lenguajes de alto Nivel y los Lenguajes Procedimentales.

- Lenguajes Máquina : Son una representación simbólica del conjunto de instrucciones de la CPU.

- Lenguajes de Alto nivel : Permiten al programador y al programa independizarse de la máquina, los compiladores e intérpretes de los lenguajes de alto nivel producen lenguaje máquina como salida. El código máquina, los lenguajes ensambladores y los lenguajes de programación de alto nivel son considerados como las tres primeras generaciones.



Con cualquiera de estos lenguajes , el programador a de preocuparse tanto de la especificación de la estructura de la información como de la de control del propio programa. Por ello los lenguajes de las tres primeras generaciones se denominan Lenguajes Procedimentales, entonces el grupo de lenguajes de cuarta generación son los no Procedimentales .

SOFTWARE DE SISTEMAS.

El software de sistemas es un conjunto de programas que han sido escritos para servir a otros programas. Algunos programas de sistemas como compiladores, editores, etc. procesan estructuras de información complejas pero determinadas. Otras aplicaciones de sistemas como ciertos componentes del sistema operativo y utilidades de manejo de periféricos, procesan datos en gran medida indeterminados. Así que el área de software de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora, una gran utilización por múltiples usuarios, unas estructuras de datos complejas, una compartición de recursos. etc.

SOFTWARE DE GESTIÓN.

El proceso de información comercial, como nóminas, inventarios, cuentas de haberes/débitos, es un software que accede a una o más bases de datos grandes que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes en el orden a facilitar las operaciones comerciales.



SOFTWARE DE COMPUTADORAS PERSONALES.

El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, entretenimientos, gestión de bases de datos, aplicaciones financieras de negocios y personales, redes o acceso de base de datos externas son algunas de los cientos de aplicaciones, de hecho el software de las computadoras personales continua representando uno de los diseños de software más innovadores en el campo.

3.3 DISEÑO DE SISTEMAS.

DISEÑO DEL SOFTWARE.

El diseño traduce los requisitos del software a un conjunto de representaciones (algunas gráficas y otras tabulares o basadas en lenguajes) que describen la estructura de datos, la arquitectura, el procedimiento algorítmico y las características de la interfaz.

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería puede definirse como : *"El Proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física".*

El objetivo del diseñador es producir un modelo o representación de una entidad que se construirá más adelante, el proceso por el cual se desarrolla el modelo combina : La Intuición y los criterios con base en la experiencia de construir entidades similares. Dichos criterios que permiten discernir sobre la calidad y un proceso de interacción que conduce finalmente a una representación del diseño final.



Una vez que se han establecido los requisitos del software, el Diseño del Software es la primera de las tres actividades técnicas, Diseño, Codificación y Prueba. Mediante las metodologías de diseño, se realiza el diseño de datos, el diseño Arquitectónico y el diseño Procedimental.

El diseño de datos transforma el modelo del campo de información creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software. El diseño arquitectónico define las relaciones entre los principales elementos estructurales del programa. El diseño Procedimental transforma los elementos estructurales en una descripción procedimental del software. Se genera el código fuente y para integrar y validar el software se llevan a cabo las pruebas.

La importancia del Diseño de Software es la calidad. El Diseño produce las representaciones del software en las que puede evaluarse su calidad.

El diseño es la única forma mediante la cual podemos traducir con precisión los requisitos del cliente a un producto o sistema acabado, el diseño de software sirve como base de todas las posteriores etapas del desarrollo y la fase de mantenimiento.

Desde el punto de vista de la gestión del proyecto, el diseño de software se realiza en dos pasos. El Diseño Preliminar el cual se centra en la transformación de los requisitos en los datos de arquitectura del software. El Diseño Detallado, el cual se ocupa del refinamiento de la representación arquitectónica que lleva a una estructura de datos detallada de la representación algorítmica del software.

Los criterios de calidad del Diseño son :

- 1.- Un diseño debe exhibir una organización jerárquica
- 2.- Un diseño debe ser modular, esto es, el software debe estar dividido de forma lógica en elementos que realicen funciones y subfunciones específicas.



- 3.- Debe contener representaciones distintas y separado de los datos y procedimentales.
- 4.- Debe llevar a módulos (subrutinas o procedimientos) que exhiban características funcionales independientes.
- 5.- Debe llevar a interfases que reduzcan la complejidad de las conexiones entre módulos y el entorno exterior.

3.4 INTRODUCCIÓN A LAS BASES DE DATOS:

DEFINICIONES.

El procesamiento de la información es esencial para la administración de los gobiernos, de los negocios y la educación.

La invención de las computadoras revolucionó los métodos tradicionales de procesamiento de la información. A principios de los años 60's, varias firmas comerciales empezaron a computarizar sus sistemas de información; los datos se guardaban en medios electrónicos en lugar de guardarlos en papel, se usaban lenguajes de alto nivel para recuperar y manejar los datos en aditamentos de almacenamiento.

Al final de los años 60's surgió el sistema de bases de datos para superar los problemas asociados con los sistemas de información tradicionales. Archivos individuales se integraban en una sola base de datos para ser compartidos por todos los usuarios de una empresa.

A continuación se dan algunas definiciones importantes referentes a las bases de datos:

La palabra **DATOS** (del latín data, plural de datum) significa simplemente "hechos", entidades independientes sin evaluar. Los datos pueden ser numéricos o no numéricos (por ejemplo, alfabéticos o simbólicos).



Por otro lado **INFORMACIÓN** es un conjunto ordenado de datos los cuales pueden recuperarse de acuerdo con la necesidad del usuario. Para que un conjunto arbitrario de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe organizar lógicamente en archivos.

Los siguientes son términos básicos usados en la organización lógica de los datos en archivos computacionales :

CAMPO, REGISTRO, ARCHIVO.

Un **CAMPO** es la unidad más pequeña a la cual uno puede referirse en un programa de computadora. Un conjunto de campos con relación entre sí se agrupa como un **REGISTRO**, y una colección de registros del mismo tipo se llama **ARCHIVO**.

REGISTROS LÓGICOS Y REGISTROS FÍSICOS.

Un **REGISTRO LÓGICO** representa la percepción del programador de lo que es un registro de datos.

Un **REGISTRO FÍSICO** puede consistir de varios registros lógicos además de un control del sistema donde guarda información sobre el almacenamiento de los datos para facilitar la búsqueda. Esta parte se llama **INFORMACIÓN SOBRE EL SISTEMA**. Un **REGISTRO FÍSICO** es una unidad de transferencia de datos entre el dispositivo de almacenamiento de datos y la memoria principal.



3.4.1 ¿ QUE ES UN SISTEMA DE BASES DE DATOS ?.

Un sistema de Bases de Datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de Software llamados Sistemas de Manejo de Bases de Datos (DBMS). Los tres componentes principales de un sistema de bases de datos son :

- ◊ El Hardware.
- ◊ El Software DBMS.
- ◊ Los datos por Manejar.

3.4.2 BASES DE DATOS.

Una **BASE DE DATOS** es una colección de archivos interrelacionados creados con un DBMS. El contenido de una base se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios, y los DATOS REDUNDANTES puedan eliminarse o al menos minimizarse.

En la siguiente figura se muestra la base de datos como un recipiente de datos a ser compartidos por varios programas. El usuario podrá recobrar datos de varias partes de la base ya que los archivos ahí almacenados, están conectados directa o indirectamente.





3.4.3 SISTEMAS DE MANEJO DE BASES DE DATOS.

El sistema de manejo de bases de datos es la porción más importante del software de un sistema de base de datos.

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son:

- a) Crear y organizar la base de datos



- b) Establecer y mantener las trayectorias de acceso a la base de datos, de tal manera que los datos en cualquier parte de la base se puedan acceder rápidamente
- c) Manejar los datos de acuerdo con las peticiones de los usuarios
- d) Mantener la integridad y seguridad de los datos
- e) Registrar el uso de las bases de datos.

El DBMS interpreta y procesa las peticiones del usuario para recobrar información de la base. Es decir, el DBMS sirve de interfase entre las peticiones del usuario y la base de datos. Las preguntas a la base pueden tener distintas formas, pueden teclearse directamente desde la terminal, o codificarse como programas en lenguajes de alto nivel y presentarse para procesamiento interactivo o por lotes. En la mayoría de los casos, una petición de consulta tendrá que atravesar varias capas de software en el DBMS y en el sistema operativo antes de que se pueda acceder la base de datos física. El DBMS responde a una pregunta llamando a los subprogramas apropiados, cada uno de los cuales realizará su función especial para interpretar la petición o localizar los datos deseados en la base y presentarlos en el orden solicitado. Así el DBMS protege a los usuarios de la tediosa programación que tendrían que hacer para organizar el almacenamiento de los datos, o accederlos una vez almacenados.

3.4.4 HARDWARE PARA SISTEMAS DE BASES DE DATOS.

El componente de hardware de un sistema de base de datos consiste en el dispositivo de almacenamiento de acceso directo (Direct-Access Storage Device, DASD), la computadora central y sus aditamentos asociados, tales como la memoria principal, el procesador de Entrada/Salida y la unidad de control.



Los dos aditamentos principales para el almacenamiento de datos de una computadora son la memoria principal y los dispositivos de almacenamiento externo (algunas veces llamados auxiliares o secundarios) tales como cinta magnética, disco, tambor y memoria burbuja. La unidad central de procesamiento CPU proporciona el acceso a la memoria principal, con una rapidez de acceso de menos de un microsegundo. Por lo tanto, la memoria principal se usa principalmente como almacén temporal para programas y datos a ser ejecutados por el CPU.

La rapidez de acceso a la memoria de un dispositivo de almacenamiento externo, es mucho más lenta debido al movimiento electromecánico, ésta se encuentra casi siempre en la magnitud de los milisegundos. Sin embargo, el costo por una capacidad igual de almacenamiento en un aditamento externo, es mucho menor que el de la memoria principal. Consecuentemente, el almacenamiento externo se usa para almacenar datos permanente, a pesar de la lentitud de acceso a la memoria.

El problema de usar los lentos aditamentos de almacenamiento externo, disponibles hoy en día, se compensa con la enorme diferencia entre la velocidad de ejecución del CPU (en nanosegundos) y la lenta Entrada/Salida en la transferencia de datos entre el almacenamiento externo y la memoria principal. El CPU tiene que esperar el equivalente de un millón de ciclos entre el momento en que se manda una petición de E/S y el momento en que los datos se transfieren del almacenamiento externo a la memoria principal. Por lo tanto, es de vital importancia el que se organicen eficazmente los archivos para el acceso de datos en dispositivos externos para que así, los registros deseados puedan localizarse con un mínimo número de operaciones de E/S.

Estos factores (limitaciones de rapidez y capacidad de almacenamiento en los dispositivos externos) tienen un gran impacto en el diseño de sistemas de manejo de bases de datos. Juzgando por el estado actual de la tecnología de almacenamiento, los dispositivos y sistemas administrativos actuales seguirán usándose en un futuro previsible.



3.4.5 ADMINISTRADOR DE BASES DE DATOS.

En la mayoría de las empresas los usuarios finales no tienen control directo sobre la forma en la que se guarda la información en la base de datos. En vez de ello, se asignan uno o más individuos (o un equipo de profesionales) para controlar y manejar la base. Este individuo o equipo es llamado administrador de la base de datos (database administrator, DBA). El personal en el equipo DBA generalmente tiene experiencia en DBMS, diseño de bases, sistemas operativos huésped, comunicación de datos, hardware y programación. Un DBA también debe poseer buena comunicación verbal y conocimiento de la empresa para que el sistema pueda servir a sus usuarios de la mejor manera posible.

3.4.6 ¿ QUÉ ES EL MANEJO DE DATOS ?.

El manejo de datos es el uso de procedimientos computarizados para la captura de datos, almacenamiento y procesamiento de la información.

Desde el principio de la civilización se ha reconocido la importancia de organizar y almacenar grandes cantidades de información para después recuperarla en forma eficaz.

Desde la primera aplicación de las computadoras al manejo de datos en los 60's, la cantidad de datos recolectados así como la generación de información ha crecido a pasos acelerados. El uso de computadoras en el manejo de datos se ha generalizado, desde las grandes corporaciones a los pequeños negocios y aún en el manejo del hogar. Esto se debe principalmente a la reducción del costo de las computadoras y la disponibilidad de software fácil de usar, en donde se puede ver la información en forma de gráficas, mapas y otras imágenes, además del despliegue tradicional de texto y tablas.



EVOLUCIÓN DE LA TECNOLOGÍA DEL MANEJO DE DATOS.

En las dos últimas décadas, el software para el manejo de datos evolucionó gradualmente desde el procesamiento secuencial hasta los sistemas actuales de manejo de bases de datos. Esta evolución se desarrolló paralelamente con el hardware. La piedra angular en la evolución del software de manejo de bases de datos fue el desarrollo de los lenguajes de alto nivel a principios de los sesentas. Es mucho más sencillo desarrollar un sistema de manejo de datos en lenguaje de alto nivel que en algún lenguaje ensamblador; sin embargo, una de las complicaciones que aparecieron en ese período durante el diseño de un sistema de información, fue que el almacenamiento de datos quedaba limitado a dispositivos de almacenamiento secuencial. Cuando aparecieron, a mediados de los sesentas, los dispositivos de almacenamiento de acceso directo como los discos magnéticos, fue ya posible el acceso aleatorio de registros en los archivos. Esto aceleró el desarrollo de métodos de acceso para crear y procesar archivos con estructuras más complicadas.

Cuando un método de acceso soporta las organizaciones de archivos para acceso aleatorio, no es necesario que un programador escriba códigos dependientes de dispositivos para el manejo de datos en donde se encuentran almacenados físicamente. En la actualidad, los fabricantes de computadoras incluyen como parte del sistema operativo, métodos básicos de acceso secuencial o aleatorio.

Al continuar el desarrollo en el uso de la información, aparecieron al final de los sesentas los sistemas de manejo de archivos, como solución temporal a la creciente demanda de paquetes sencillos de software. El sistema de manejo de archivos (File Management System FMS) puede usarse para establecer y procesar un sistema de información sin necesidad de escribir programas en lenguajes de alto nivel.

El FMS depende totalmente de los métodos de acceso en los que se basa el sistema operativo para crear y procesar sus archivos de datos. En consecuencia, sólo se puede lograr el acceso aleatorio de archivos de datos FMS en un archivo a la vez, usando las claves primarias. Siguió la



búsqueda de organizaciones de archivos usando claves de búsqueda múltiple. Por fin, el avance vino a finales de los sesentas y principios de los setentas, cuando se desarrollaron los sistemas de manejo de bases de datos (DBMS).

El DBMS crea organizaciones de archivos de alto nivel por encima de los métodos básicos de acceso del sistema operativo central huésped para interconectar registros relacionados. Como resultado, se pueden recuperar datos de distintos archivos en la base de datos con una sola consulta.

3.4.7 SISTEMA DE MANEJO DE ARCHIVOS.

Un sistema de manejo de archivos se asemeja a uno de manejo de bases de datos (DBMS) en que se puedan desarrollar aplicaciones sin necesidad de escribir programas en lenguajes de alto nivel. De hecho, los FMS son anteriores a los DBMS y usan los métodos básicos de acceso como su principal característica en el manejo de datos. Los FMS se desarrollaron a finales de los años sesentas como resultado de la creciente demanda de un software eficiente y sencillo para el procesamiento de información.

Aunque se puede establecer un sistema de información usando un lenguaje de alto nivel, hay algunas desventajas al hacerlo así:

1) Altos costos del software. Pueden requerirse meses y a veces años de esfuerzo de programadores expertos antes de desarrollar un sistema convencional de información; por lo tanto, su costo puede ser muy elevado.

2) Poca eficiencia en las consultas. El acceso aleatorio en un archivo sólo puede efectuarse usando valores de claves únicas. La programación de consultas en línea usando claves múltiples es muy tediosa.



¿ QUÉ ES UN FMS ?.

Un FMS es un paquete de software escrito en lenguaje de alto nivel y/o en lenguaje ensamblador. Tiene posibilidades de organizar, combinar archivos y escribir reportes. El FMS interpreta a los comandos del usuario. No proporciona más utilidades para el manejo de datos que las de los métodos básicos de acceso del sistema operativo.

El objetivo de un FMS no es sólo incrementar la productividad de los programadores sino también proporcionar a los usuarios finales, sin experiencia en programación, la capacidad de desarrollar consultas con un lenguaje especial.

3.5 EL COMPILADOR CLIPPER.

3.5.1 HISTORIA:

Clipper surgió a la sombra de dBase III, tratando de superar las deficiencias que aquel presentaba. Aportaba un compilador que permitía la obtención de programas directamente ejecutables a nivel de DOS y que, por tanto, podían distribuirse a clientes sin necesidad de que tuvieran que comprar el gestor de bases de datos. En esto radica su popularidad. Por un lado permitía trabajar con el cómodo y superconocido lenguaje dBase y por otro facultaba para distribuir, sin costos adicionales para los clientes las aplicaciones desarrolladas. Con el paso del tiempo vinieron nuevas versiones tanto de dBase como de Clipper, y con ellas nuevas funcionalidades. dBase fue aumentando su conjunto de mandatos, Clipper le siguió y permitió además el interfaz con otros lenguajes de más bajo nivel como C y ensamblador. Con ello el programador de Clipper se fue haciendo más profesional. Su herramienta le daba más recursos y parecía sensato aprovecharlos.



Nantucket (la empresa desarrolladora de Clipper), considera que dBase IV está evolucionando por un camino equivocado y que, por tanto, ya no es necesario mantener la compatibilidad. Ello les lleva a apostar, en cambio, por lo que denominan NFT (La tecnología de Nantucket para el futuro - Nantucket Future Technologie). La NFT se caracteriza por el intento de estandarizar un nuevo entorno de desarrollo para sistemas de bases de datos, con la característica esencial de estar enfocado hacia cualquier plataforma de las existentes en el mercado. Dentro de este orden de cosas Clipper aporta un driver DBFNTX para ficheros del tipo dBase, pero también aporta otros drivers para plataformas: SQL, Paradox, Oracle, Informix o cualquier producto de base de datos existente. Lo mismo sucede con las plataformas hardware. Anteriormente, Clipper escribía directamente sobre la pantalla, ahora lo hace sobre un driver que podrá cambiarse por otro distinto cuando camblemos de sistema. Toda esta evolución posiciona a Clipper en un punto óptimo para abordar la nueva filosofía cliente servidor que caracteriza a los sistemas que hoy se comercializan.

VERSIONES DE CLIPPER.

- o Clipper Autumn 86
- o Clipper Summer 87
- o Clipper 5.0
- o Clipper 5.01
- o Clipper 5.2.

3.5.2 REQUERIMIENTOS DE HARDWARE Y SOFTWARE.

Clipper 5.2 puede funcionar en un equipo IBM, PS/2, AT, XT, PC o 100% compatible. La capacidad mínima adecuada de memoria en RAM es de 512 KB, el disco duro es necesario para desarrollar aplicaciones, aunque estas una vez desarrolladas puede funcionar sobre disquettes.



La versión del Sistema Operativo necesaria es la 3.1 o superior tanto para las versiones monousuario, como multiusuarios.

El compilador y el ligador utilizan la RAM que le sea provista (arriba de 640 KB) ambos utilizan bloques de memoria, los cuales crean líneas de información, tablas de símbolos, etc. Los programadores utilizan utilerías como:

- * Norton Guide para documentación en línea.
- * Utilerías residentes en memoria como Side Kick, el Doskey, PcTools, etc.
- * Dispositivos de propósito general declarados en el config.sys para discos duros, tarjetas de red, etc.

MEMORIA EXPANDIDA.

Clipper no hace uso de la memoria expandida en el ciclo de desarrollo, el manejo de memoria virtual usa LIM 3.2 o superior para proveer un almacenamiento de datos en las variables, buffer de índices, código de overlays. Cuando la memoria convencional se termina, la memoria virtual utiliza la memoria expandida hasta que esta se termina, entonces Clipper empieza a intercambiar segmentos de memoria a disco. Sin embargo, la aplicación correrá con o sin memoria expandida.

MEMORIA EXTENDIDA.

Si se tiene una AT o con procesador 386 para desarrollo de aplicaciones, se tendrá oportunidad de tener algo de memoria extendida, esta puede ser usada de dos formas: memoria caché o RAM, estas proveen una mayor rapidez en los procesos de compilación y enlace.



CAPACIDADES DEL SISTEMA.

| | |
|--|-------------------------|
| No. máximo de registros por BD | 1 billón |
| No. máximo de caracteres por registro | RAM disponible |
| No. máximo de campos por registro | 1024 |
| No. máximo de caracteres por campo | 32 KB |
| Dígitos máximo en un campo numérico | 30 |
| Campo memo (longitud variables) | 64 KB |
| Precisión en operaciones de cálculo | 16 dígitos |
| No. máximo de caracteres en un clave de indexación | 256 |
| No. máximo de índices por área de trabajo | 15 |
| No. de variables de memoria públicas o privadas | 2048 |
| No. de variables de memoria locales o estáticas | RAM disponible |
| Tamaño máximo de una variable de memoria (cadena) | 64 KB |
| No. máximo de dígitos en una variable numérica | 30 dígitos |
| Rango válido de fechas | 01/01/100 31/12/9999 |
| No. máximo de arrays | 2048 |
| No. de elementos por dimensión de array | 4096 |
| No. de dimensiones por array | RAM disponible |
| No. de ficheros abiertos (DOS 3.3) | 250 |

MANEJO DE MEMORIA.

- * Poseer un sistema automático para manejo de memoria virtual que hace que los overlays sean dinámicos siendo el propio sistema quien decide cómo y cuándo cargue y descargue los módulos de memoria
- * Usa la memoria expandida para aprovechar mejor los recursos que las máquinas modernas ponen en nuestras manos
- * Los overlays reparten la memoria disponible en dos zonas:

1) Un área principal donde se deposita el programa fundamental y los recursos comunes de la aplicación.



2) Varias áreas de overlays donde se iban cargando y descargando las diferentes partes de la aplicación.

Una aplicación Clipper funciona del siguiente modo: en primer lugar, se determina la cantidad de memoria expandida de que consta el sistema. Si con ella es suficiente la aplicación funciona entre la memoria convencional y la expandida, sin usar otros recursos. Clipper puede direccionar en el disco hasta 64 MB de memoria virtual.

3.5.3 USO DEL COMPILADOR.

La compilación es una traducción del fichero fuente (.PRG) para obtener un fichero objeto (.OBJ). La traducción mencionada lo que realmente hace es traducir cada instrucción desde el lenguaje simbólico en que está escrito (en este caso Clipper) a código entendible por el enlazador del sistema operativo con que se está trabajando.

La sintaxis es:

```
CLIPPER [ <Fprg> / @ <Fclp> ] [ / <opción> { / <opción> } ]
```

Donde:

<Fprg> es el nombre del fichero .PRG que contiene el programa, la extensión no es preciso indicársela ya que la toma por defecto. No obstante, si damos a nuestros programas una extensión diferente a .PRG entonces si hemos de indicarla en el momento de la compilación.

<Fclp> es el nombre de un fichero de compilación que contiene los nombres de varios ficheros fuente.

El programa puede contener así mismo diversas llamadas a otros módulos (DO) .PRG o a procedimientos del mismo programa si no le especificamos lo contrario, Clipper compilará de forma automática el fichero llamado por DO.



Cuando se produce una labor de compilación aparecen en pantalla mensajes que nos indican el bloque de 100 líneas que se están compilando, así como los errores de sintaxis si es que existen, por último se nos informa el número de símbolos, constantes y el tamaño del código. Si queremos que esto no aparezca en pantalla sino que lo haga en un fichero de texto o en la impresora solo se tendrá que desviar la salida hacia dicho fichero o hacia PRN usando el carácter de direccionamiento del DOS ">".

3.5.4 USO DEL ENLAZADOR.

El fin de un enlazador es el de asociar los módulos objeto, obtenidos mediante el compilador, con las librerías donde se encuentran las traducciones máquina de cada una de las sentencias llamadas que aparecen en el módulo objeto.

El producto proporcionado por el enlazador es el módulo ejecutable (.EXE) que es el que ya podemos hacer funcionar con solo teclear su nombre.

- Ⓞ El enlazador permite overlays dinámicos, estos hacen que el programador se olvide de las dimensiones que va a tener su aplicación. El enlazador se encarga de preparar el programa ejecutable para que vaya cargándose en memoria según las necesidades del sistema.
- Ⓞ El programador puede decidir que partes del código cargar y descargar en la memoria según las necesidades de espacio que se generan.

FORMAS DE USO DEL ENLAZADOR.

Existen tres modos diferentes de usar el enlazador RTLINK: En formato libre (FREEFORMAT), en formato posicional (POSITIONAL) o de forma Interactiva.



El primero nos permite situar las cláusulas en cualquier posición, siempre que vayan precedidas por el correspondiente identificador, la sintaxis es:

```
RTLINK [FILE <lista de objetos>[OUTPUT <fichero de salida>]
        [LIBRARY <lista librerías>][<lista de opciones>]] | [ @
        <Fink>]
```

FILE y LIBRARY son dos mandatos del enlazador que sirven para relacionar los módulos objeto y las librerías respectivamente, pueden escribirse también en forma abreviada como FILL o LIB OUTPUT sirve para indicar el nombre y vía del fichero ejecutable cuando queremos que sea diferente al primer objeto relacionado.

La diferencia entre el posicional (positional) y el libre formato (freeformat) es la sintaxis. En el primero, las opciones de enlace son ordenadas en una lista de argumentos delimitados por comas. El libre formato usa etiquetas para identificar las partes de la aplicación para enlazarlas, los argumentos pueden estar en cualquier orden.

El tercer tipo, es decir la forma interactiva, funciona tecleando solo RTLINK y contestando las diferentes preguntas que el enlazador va haciendo.

Clipper tiene la habilidad de crear overlays haciendo que el programador sea capaz de crear aplicaciones más grandes que la memoria disponible. El sistema aloja fragmentos de memoria en varios componentes de un archivo ejecutable : Tablas de símbolos, Tablas de Variables, Códigos Ejecutables, Memoria para Almacenamiento, cuando todo esto no tiene espacio para colocarse en memoria, es entonces cuando son necesarios los overlays.

Overlays = Programación por capas, segmentación, espacio reservado en memoria para el código que ocupe más de la memoria disponible.

RTLINK también permite integrar módulos objeto de código no Clipper (como lenguaje C o ensamblador) usando overlays estáticos. cuando programamos usando esta técnica lo que se hace es dividir la memoria



RAM en dos o más áreas de trabajo. En el área principal se carga el módulo ejecutable y en las áreas de solape se cargan y descargan, conforme se van usando los diferentes módulos overlays que se hayan definido.

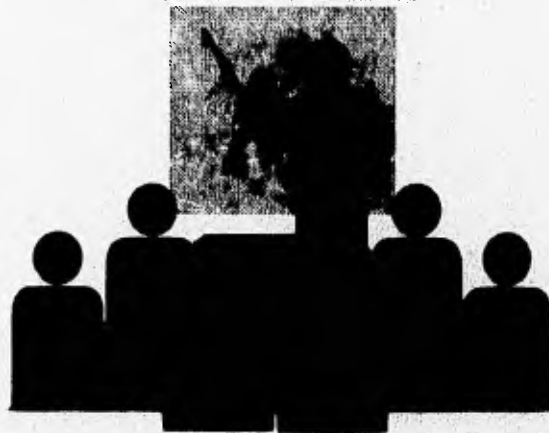
Los Overlays Estáticos en Clipper permiten diversas configuraciones. La más básica es aquella que divide la memoria de trabajo en una área principal y una de solape. Una segunda forma es la que trabaja en una área principal y varias de solape. La tercera y última forma, es la que trabaja con un área principal y varias de solape teniendo algunas de estas otros segmentos anidados. La creación de un área de overlays estático la producen los mandatos "BEGINAREA" y "ENDAREA", los distintos mandatos SECTION FILE sirven para crear cada uno de los módulos que se cargarán en el área definida.

Los Overlays Dinámicos son mucho más versátiles que los anteriores, por default Clipper utiliza el overlay dinámico, este consiste en dos partes : Un módulo raíz y un módulo overlay . el primero contiene las tablas y el código de comienzo, el segundo contiene todo el demás código. El módulo raíz se coloca en la memoria del sistema, el código restante puede intercambiarse dentro y fuera de la memoria. Los overlays dinámicos leen solo la cantidad de código necesario.



CAPÍTULO 4

SITUACIÓN ACTUAL, PLANIFICACIÓN Y DESARROLLO DEL SISTEMA



CAPÍTULO 4
SITUACIÓN ACTUAL, PLANIFICACIÓN Y
PUNTOS A DESARROLLAR DEL SISTEMA

4.1 PROBLEMÁTICA ACTUAL.

Como se mencionó al principio de este trabajo, el Departamento de Ingeniería Eléctrica de la Universidad Nacional Autónoma de México, se ve en la necesidad de consultar cotidianamente la Norma Oficial Mexicana NOM-001-SEMP-1994. misma que tiene por objeto establecer las especificaciones de carácter técnico que deben satisfacer las instalaciones destinadas al suministro y uso de energía eléctrica, a fin de que ofrezcan condiciones adecuadas de servicio y seguridad para las personas y su patrimonio.

Actualmente, esta consulta se lleva a cabo en un libro de gran volumen y con leiras pequeñas y por consiguiente la consulta resulta muy lenta y tediosa.

Todo esto ocasiona que no se dedique una plena concentración a las tareas ingenieriles de instalaciones eléctricas, ya que se dedica parte del tiempo en hacer una consulta que resulta sencilla pero muy tardada.

Por todo lo anterior y debido al avance tecnológico en lo que a computación se refiere, en dicho departamento se hace indispensable un sistema de cómputo para la consulta de esta norma.

META S.

Entre las metas a cumplir que también pueden catalogarse como atributos de calidad que todo buen sistema de cómputo debe cumplir se encuentran los siguientes:



- ❖ El sistema ha de cumplir con los requerimientos del cliente para que pueda lograr la eficiencia y la máxima productividad en el Departamento de Ingeniería Eléctrica de la Facultad de Ingeniería, así como su plena satisfacción y autorización.
- ❖ Se desarrollará el producto de programación obteniendo los atributos de: calidad, eficiencia, portabilidad, confiabilidad, solidez, corrección y exactitud.

PORTABILIDAD: Facilidad con la que un producto de programación puede ser transferido de un sistema de cómputo a otro.

CONFIABILIDAD: Capacidad de un programa de realizar una función requerida bajo ciertas condiciones durante un período determinado.

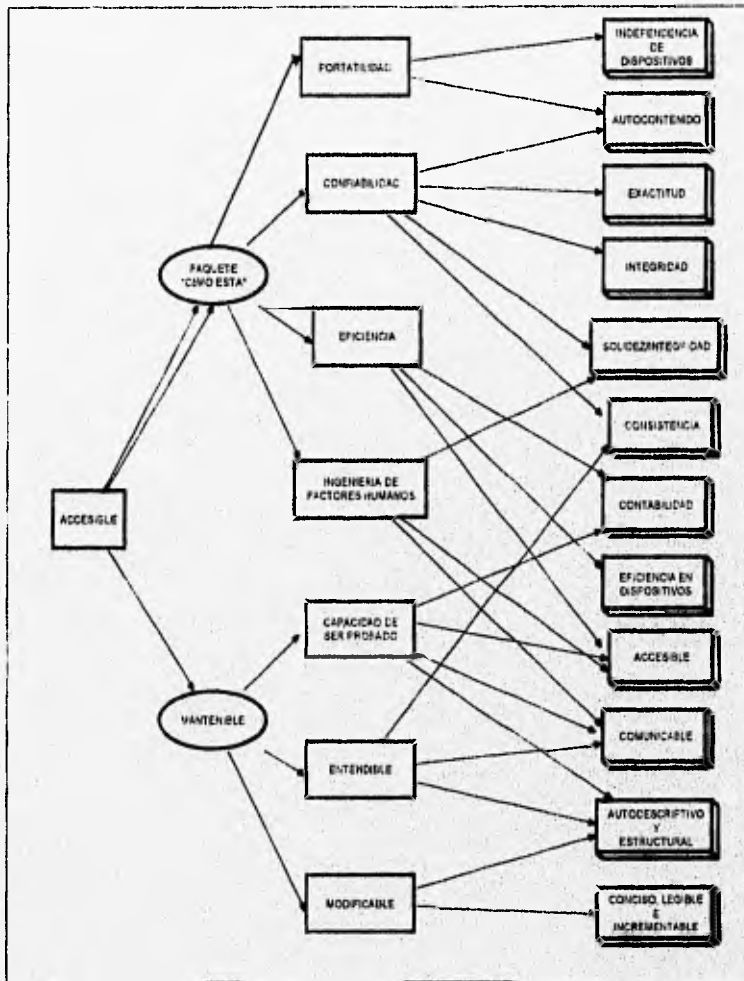
EXACTITUD: Especificación cualitativa de ausencia de error. Medida cuantitativa de la magnitud del error, de preferencia expresada como una función del error relativo.

SOLIDEZ: Grado con el que un producto de programación puede continuar operando correctamente, a pesar de la introducción de datos inválidos.

CORRECCIÓN: Grado en el que un producto de programación está libre de defectos de diseño y de codificación, esto es, libre de fallas. Grado en que un producto de programación cumple con las expectativas del usuario.

- ❖ El sistema debe producir respuestas en tiempo real.
- ❖ El sistema debe emplear con eficiencia la memoria principal.
- ❖ El sistema se diseñará modularmente.
- ❖ El sistema deberá hacer más interesante el trabajo de los usuarios.





4.2 PLANEACIÓN.

4.2.1 INTRODUCCIÓN.

Los planeadores han intentado reducir la incertidumbre asociada a la estimación de las duraciones, proyectando estimaciones pesimistas, optimistas y más probables, y luego, mediante una fórmula de promedios ponderados determinan la duración esperada para una actividad. Este enfoque sólo ofrece cierta seguridad. Tal vez la mejor estrategia para planear un sistema, consista en apegarse a un enfoque estructurado de identificación de actividades, sin dejar de describirlas con suficiente detalle.

El objetivo de la planeación del proyecto de Software es el de suministrar una estructura que permita hacer estimaciones razonables de recursos, costos y agendas.

El seguimiento del software se puede llevar a cabo de diferentes maneras:

- ❖ Realizando reuniones periódicas sobre el estado del proyecto en las cuales cada miembro del equipo da un informe de los progresos y de los problemas.
- ❖ Evaluando los resultados de todas las revisiones realizadas.
- ❖ Determinando si cada paso ha alcanzado la fecha programada.
- ❖ Comparando la fecha de comienzo real con la fecha de comienzo planeada.



4.2.2 ESTIMACIÓN DEL PROYECTO DE SOFTWARE.

La estimación de Proyectos de software es una forma de resolución de problemas, y en la mayoría de los casos, el problema a resolver (desarrollar estimación de costos y de esfuerzo para el proyecto de software), es descomponiendo el problema y recaracterizándolo como un conjunto de pequeños problemas. La estimación es importante para realizar buenas decisiones en el proyecto, este determina también cuanto esfuerzo y tiempo se requiere. La estimación de costos tienen dos usos :

- Durante la etapa de planeación, se decide cuántas personas son necesarias para el desarrollo del proyecto.
- Monitorear el progreso del proyecto ; en el monitoreo se debe saber que trabajos han sido complementados y cuáles faltan por complementar.

La mayoría de los métodos de estimación de Software comienzan con predecir el tamaño de Software y usando las entradas para derivar el total de esfuerzo requerido.

4.2.3 USO DE LOS DIAGRAMAS DE GANTT PARA LA PROGRAMACIÓN DE PROYECTOS.

Una manera sencilla para programar actividades es mediante el diagrama de Gantt. Esencialmente, es un diagrama que contiene barras que representan cada una de las actividades, y cuya longitud representa la duración de la actividad respectiva.

Esta técnica también determina:

- * El tiempo final e inicial en que una actividad puede comenzar.
- * La duración más probable de un proyecto.
- * La colocación de las actividades según su orden cronológico.

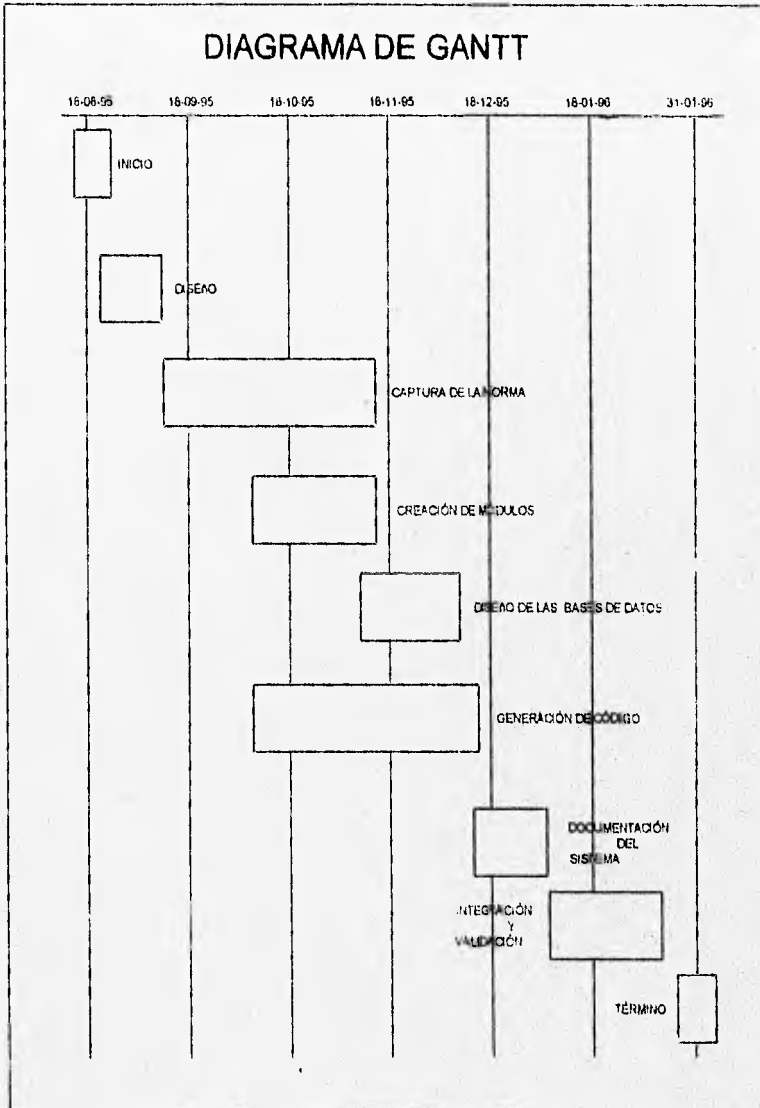


Los diagramas de GANTT (desarrollados por Henry L. Gantt) son una técnica de control de proyectos que pueden usarse para varios propósitos incluyendo calendarización, presupuestos y recursos de planeación. Un diagrama de GANTT es una gráfica de barras, en la que cada barra representa una actividad, estas se dibujan contra una línea de tiempo. La longitud de cada barra es proporcional a la longitud del tiempo planeado para esa actividad.

Un Diagrama de GANTT ayuda a calendarizar las actividades de un proyecto pero no ayuda a identificarlos. Durante la calendarización o la implementación del proyecto se pueden identificar nuevas actividades que no estén dentro del plan inicial. Así que se debe regresar y revisar la estructura para agregar estas nuevas actividades.

La ventaja principal del diagrama de Gantt es su sencillez. El analista de sistemas no sólo encontrará fácil esta metodología, sino que también contará con un excelente instrumento de comunicación con los usuarios finales. Otra ventaja de utilizar los diagramas de Gantt es que las barras que representan las actividades se dibujan a escala; esto es, el tamaño de una barra indica la duración relativa de la actividad.





4.2.4. **DIAGRAMAS PERT.**

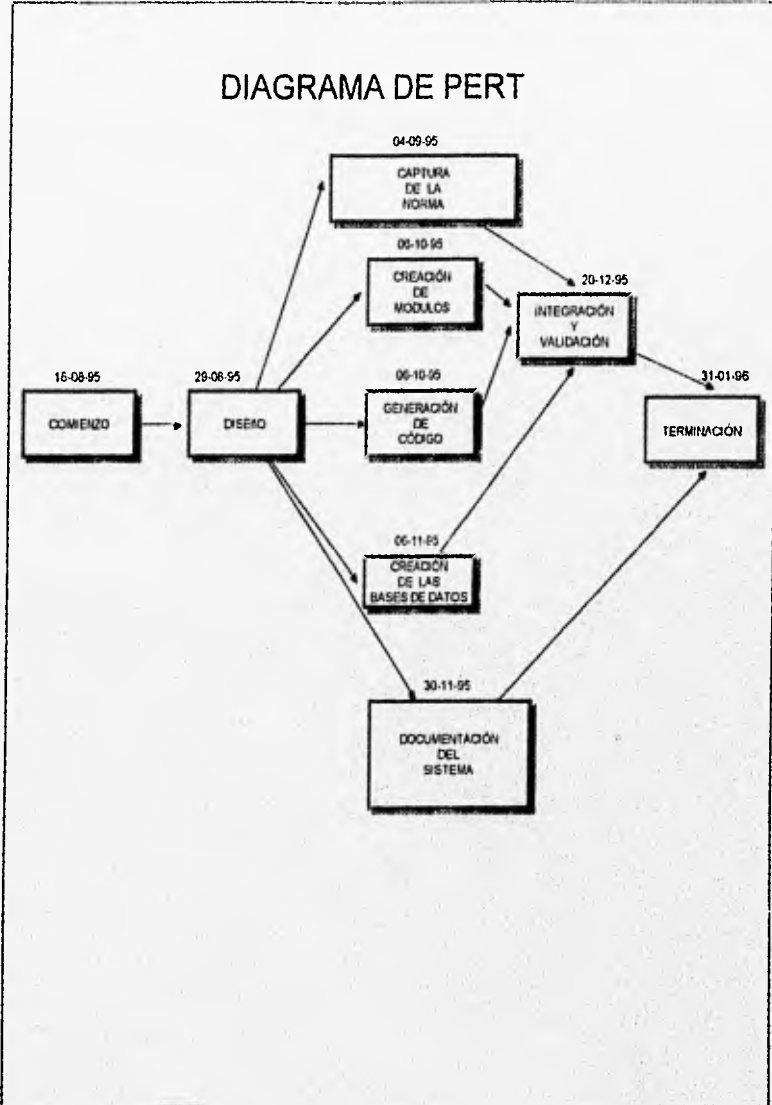
PERT son las siglas de Program Evaluation and Review Techniques (Técnicas de Revisión y Evaluación del Programa). Un programa (sinónimo de proyecto) se representa por una red de nodos y flechas, que luego se evalúa, tanto para determinar cuáles son las actividades críticas y mejorar su programación si fuera necesario, como para revisar el avance del proyecto una vez que se ha iniciado. *PERT* se desarrolló a fines de la década de los 50's para utilizarlo en el proyecto del submarino nuclear Polaris. La armada estadounidense reportó haber logrado ahorros de hasta dos años en el desarrollo del submarino al aplicar esta técnica de planeación.

PERT es útil cuando pueden realizarse varias actividades paralelamente. Los analistas de sistemas pueden beneficiarse del *PERT* al aplicarlo en proyectos de sistemas de menor escala, en especial cuando varios miembros de un equipo trabaja simultáneamente en varias actividades y otras personas de la organización participan en otras tareas.

Las ventajas son :

- ◊ Se genera una planificación .
- ◊ Muestra las relaciones entre las tareas del proyecto, identificando claramente el camino crítico, permitiendo considerar alternativas para solucionar el proyecto.
- ◊ Expone todos los posibles paralelismo entre las actividades.
- ◊ Permite la calendarización del proyecto.

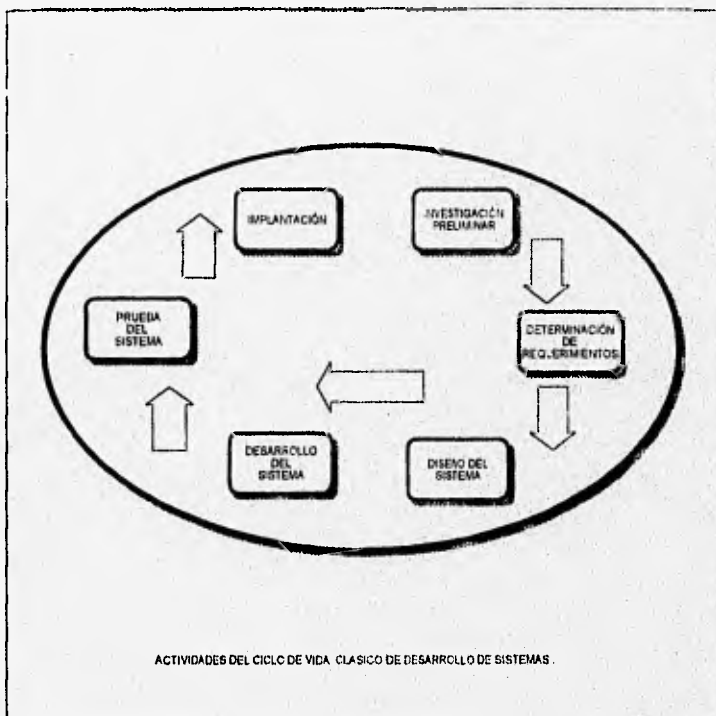




4.3 ESTRATEGIAS PARA EL DESARROLLO DE SISTEMAS.

CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS:

El desarrollo de sistemas, un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un sistema de la empresa que necesita mejorar o emprender un sistema nuevo. El método del ciclo de vida para desarrollo de sistemas (SDLC) como se ve en la siguiente figura, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.



En esta parte se analizará en forma general las 6 actividades que constituyen el ciclo de vida de desarrollo de sistemas. Cabe mencionar que en los dos siguientes capítulos se analizarán a fondo estas 6 actividades.

En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se siguen para efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases del desarrollo; algunos componentes en la fase de análisis mientras que otros en etapas avanzadas de diseño.

El método del ciclo de vida para el desarrollo de sistemas consta de las siguientes actividades:

1. Investigación preliminar.
2. Determinación de los requerimientos del sistema.
3. Diseño del sistema.
4. Desarrollo de software.
5. Prueba del sistema.
6. Implantación y evaluación.

1. INVESTIGACIÓN PRELIMINAR.

Esta actividad tiene dos partes a analizar: estudio de la factibilidad y aprobación de la solicitud.

ESTUDIO DE FACTIBILIDAD: Un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible. En la investigación preliminar existen tres aspectos relacionados con el estudio de factibilidad:

1. Factibilidad técnica. El trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible?. Si se necesita nueva tecnología, ¿cuál es la posibilidad de desarrollarla?.



2. Factibilidad económica. Al crear el sistema ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿los costos asociados con la decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto?.

3. Factibilidad operacional. Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios que dé como resultado una disminución de los posibles beneficios de la aplicación?.

APROBACIÓN DE LA SOLICITUD: No todos los proyectos solicitados son deseables o factibles, sin embargo, aquellos proyectos que son factibles deben incorporarse en los planes. En algunos casos el desarrollo puede comenzar inmediatamente. Después de aprobar la solicitud de un proyecto se estima el tiempo necesario para terminarlo.

2. DETERMINACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. (Es por esta razón que el proceso de adquirir información se denomina, investigación detallada.) Los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

- ¿Qué es lo que se hace?
- ¿Cómo se hace?
- ¿Con qué frecuencia se presenta?
- ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
- ¿Existe algún problema?
- Si existe un problema, ¿qué tan serio es?
- Si existe un problema, ¿cuál es la causa que lo origina?

Para contestar estas preguntas, el analista conversa con varias personas para reunir detalles relacionados con los procesos de la empresa.



Asimismo, las investigaciones detalladas requieren el estudio de manuales y reportes, la observación en condiciones reales de las actividades del trabajo y, en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad.

Conforme se reúnen los detalles, los analistas estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles de procesamiento, tiempo de respuesta y métodos de entrada y salida.

3. DISEÑO DEL SISTEMA.

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la de desarrollo del software, a la que denominan diseño físico.

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema a la vez de que también indican los datos de entrada que serán calculados y los que deben ser almacenados.

Los diseñadores son los responsables de dar a los programadores las especificaciones de software completas y claramente delineadas. Una vez comenzada la fase de programación, los diseñadores contestan preguntas, aclaran dudas y manejan los problemas que enfrentan los programadores cuando utilizan las especificaciones de diseño.



4. DESARROLLO DE SOFTWARE.

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitando. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores.

Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

5. PRUEBA DE SISTEMAS.

Durante la fase de prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas. es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

6. IMPLANTACIÓN Y EVALUACIÓN.

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla.



Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones; realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de las siguientes dimensiones:

➔ **Evaluación operacional**

Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

➔ **Impacto organizacional**

Identificación y medición de los beneficios para la organización en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.

➔ **Opinión de los administradores**

Evaluación de las actitudes de directivos y administradores dentro de la organización así como de los usuarios finales

➔ **Desempeño del desarrollo**

La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo.

Desafortunadamente la evaluación del sistema no siempre recibe la atención que merece. Sin embargo, cuando se conduce en forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.



CAPÍTULO 5

ANÁLISIS Y DISEÑO
DEL SISTEMA





5.1 ¿QUÉ ES EL ANÁLISIS Y DISEÑO DE SISTEMAS?

Dentro de las organizaciones, el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados. A continuación se presenta un panorama del análisis y diseño del sistema y describe el trabajo de los analistas de sistemas así como los diferentes tipos de usuarios que participan en el proceso de desarrollo.

5.1.1 PANORAMA DEL ANÁLISIS Y DISEÑO DE SISTEMAS.

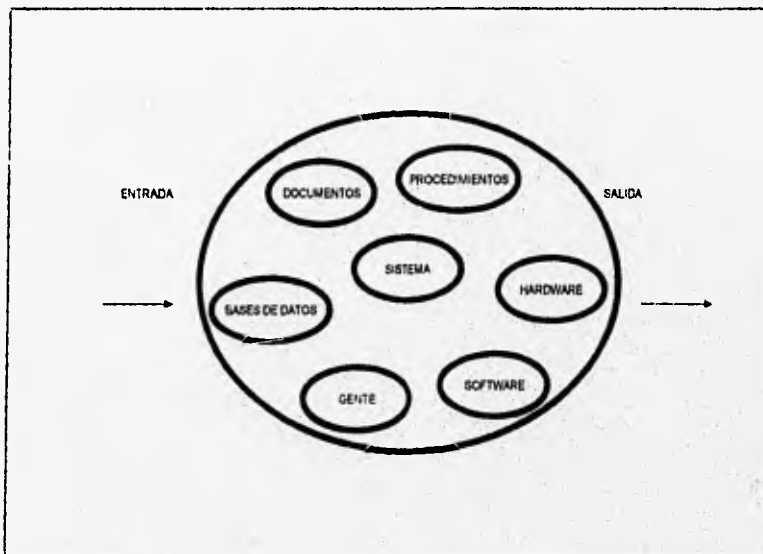
El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: el análisis de sistemas y el diseño de sistemas. El **diseño de sistemas** es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. Pero antes de llevar a cabo esta planeación es necesario comprender en su totalidad el viejo sistema y determinar la mejor forma en que se pueden, si es posible, utilizar las computadoras para hacer la operación más eficiente. El **análisis de sistemas**, por consiguiente, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de información para recomendar mejoras al sistema. Este es el trabajo del analista de sistemas.



5.2 ANÁLISIS DEL SISTEMA.

INTRODUCCIÓN:

Un sistema basado en computadora es un conjunto u ordenación de elementos organizados, para llevar a cabo un método, procedimiento o control mediante el proceso de información. Los elementos de un sistema son :



Software : Los programas de computadora, las estructuras de datos, la documentación asociada que sirve para realizar el control requerido.

Hardware : Los dispositivos electrónicos que proporcionan el procesamiento de datos.

Gente : Usuarios u operadores de hardware y software.



Bases de Datos : Colección grande y organizada de información que accede mediante el software. Es una parte integral del funcionamiento del sistema.

Documentos : Los manuales, los impresos e información descriptiva que explica el uso y la operación del sistema .

Procedimientos : Los pasos que definen el uso específico de cada elemento del sistema .

Todos estos elementos se combinan en diversas formas para transformar la información.

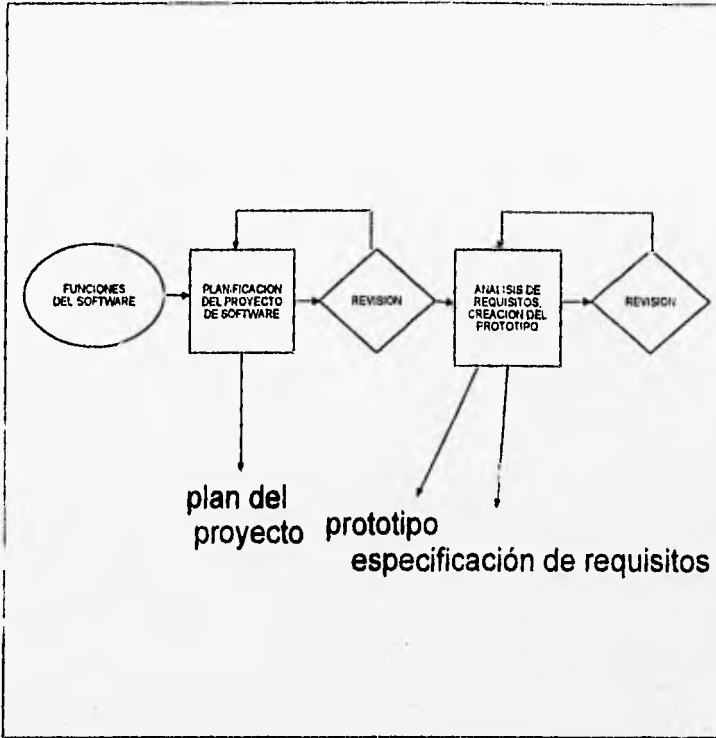
La ingeniería de sistemas parte de los objetivos y de las restricciones definidas por el usuario y desarrolla una representación de la función, del rendimiento, de las interfases y de la estructura de la información. El ingeniero en sistemas debe delimitar el sistema identificando el ámbito del funcionamiento y el rendimiento deseado.

El Software se usa para adquirir información que puede ser suministrada por alguna fuente externa o por otro elemento del sistema. Cuando el sistema basado en computadora requiere una interfaz interactiva entre hombre y máquina, el software implementa la conversión de entrada-salida. En el software se implementan los mecanismos de petición y de entrada de datos; Con el software se generan las pantallas y los gráficos y mediante el se lleva acabo la lógica que conduce al usuario a través de la secuencia de pasos interactivos. También se usa para establecer interfases con las bases de datos permitiendo a un programa acceder a fuentes de datos existentes.

El Software implementa los algoritmos de procesamiento requeridos para realizar las funciones del sistema. En general un algoritmo de procesamiento transforma datos de entrada y produce información o control como salida a otro elemento del sistema. Para producir la información de salida, debe dar un formato a los datos que resulte apropiado para el medio de salida y saber cómo comunicarse con el dispositivo de salida.



5.2.1 FASE DE DEFINICIÓN.



La fase de definición representada por la figura anterior comienza con la etapa de planificación, donde se establecen las estimaciones de tiempo, las cuales ya se establecieron en el capítulo anterior. El propósito de esta etapa es proporcionar una indicación preliminar de viabilidad del proyecto de acuerdo con el coste y con la agenda que se haya establecido. El paso siguiente es el análisis y la definición de los requisitos del software, en este paso se define con detalle el elemento del sistema asignado al software. Los requisitos se analizan y se definen haciendo un análisis formal del ámbito de información para establecer módulos de flujo y la estructura de la información, luego se amplían esos módulos para convertirlos en una especificación de software.



El análisis y definición de los requisitos de software es un esfuerzo conjunto llevado a cabo por el desarrollador de software y el cliente.

5.3 MÉTODOS DE ANÁLISIS DE REQUERIMIENTOS.

1. DIAGRAMAS DE FLUJO DE DATOS.

La información que se maneja en un software, pasa por ciertos procesos, en los cuales la mayoría de las veces es modificada. El diagrama de flujo de datos (DFD), es una representación gráfica de los flujos de datos y de las funciones o procesos de un sistema, permitiendo al ingeniero de software desarrollar los módulos del ámbito de información y del ámbito funcional.

La notación básica que se usa para crear diagramas de flujo de datos consta de 4 elementos que son:



Los flujos de datos son canales, en los cuales fluyen unidades de información cuya estructura conocemos.



Las transformaciones o procesos, son acciones que transforman uno o varios flujos de datos entrantes ó varios flujos de datos salientes.



Un almacén de datos, es un dispositivo para los datos al que se accede con retardo temporal. Puede tratarse tanto de ficheros manuales, como de ficheros legibles para la máquina.



Un enlace final, es un enlace del sistema con exterior, con ellos se representan por ejemplo las introducciones que efectúa el usuario el sistema, así como las salidas del sistema destinadas al usuario.



Al diseñar los diagramas de flujo de datos se deben seguir las siguientes reglas:

- ⇒ Se documenta exclusivamente el flujo de datos.
- ⇒ Los flujos de datos así como las transformaciones deben llevar nombres claros y expresivos.
- ⇒ Cada una de las transformaciones pueden generar únicamente aquellas salidas para las que también recibe las entradas relevantes.

El diagrama de flujo de datos (DFD) más sencillo, es aquel en el cual el programa se especifica con una sola burbuja, y se dice que ese DFD es de nivel 0, mientras más específico sea ese diagrama va aumentando de nivel.

A continuación se muestran los diagramas de flujo de datos del sistema SISPCNOM, se observa un diagrama de nivel 0, en que se representa al sistema con una burbuja y las salidas de este dirigidas a los diferentes dispositivos, en los siguientes diagramas aumenta de nivel ya que se especifican más afondo las acciones.

SISPCNOM

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA



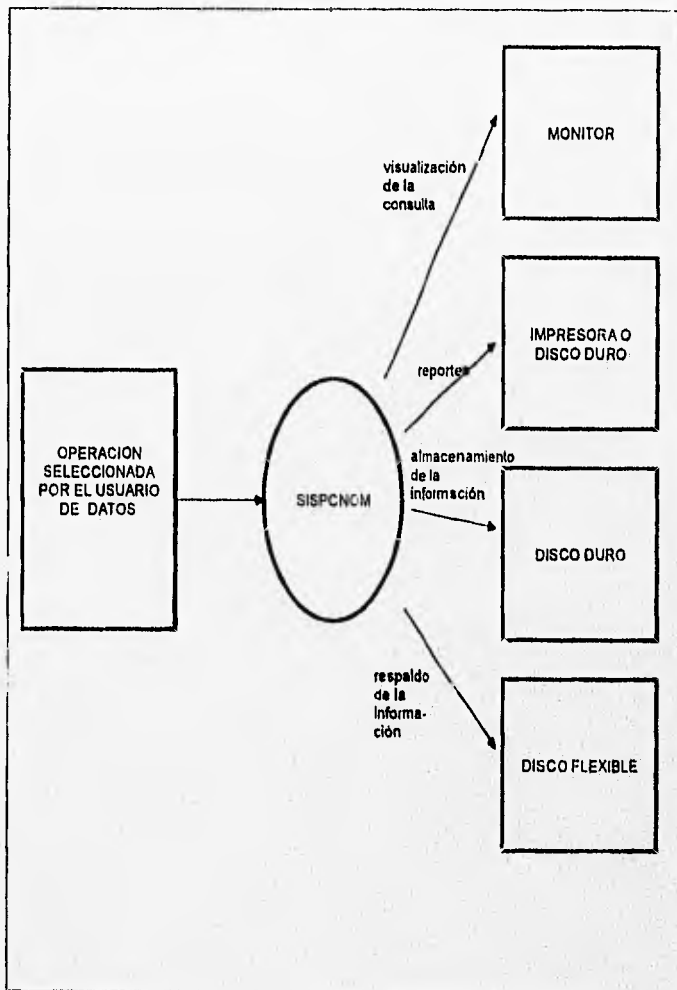


Diagrama de flujo de datos del sistema SISPCNOM nivel 0



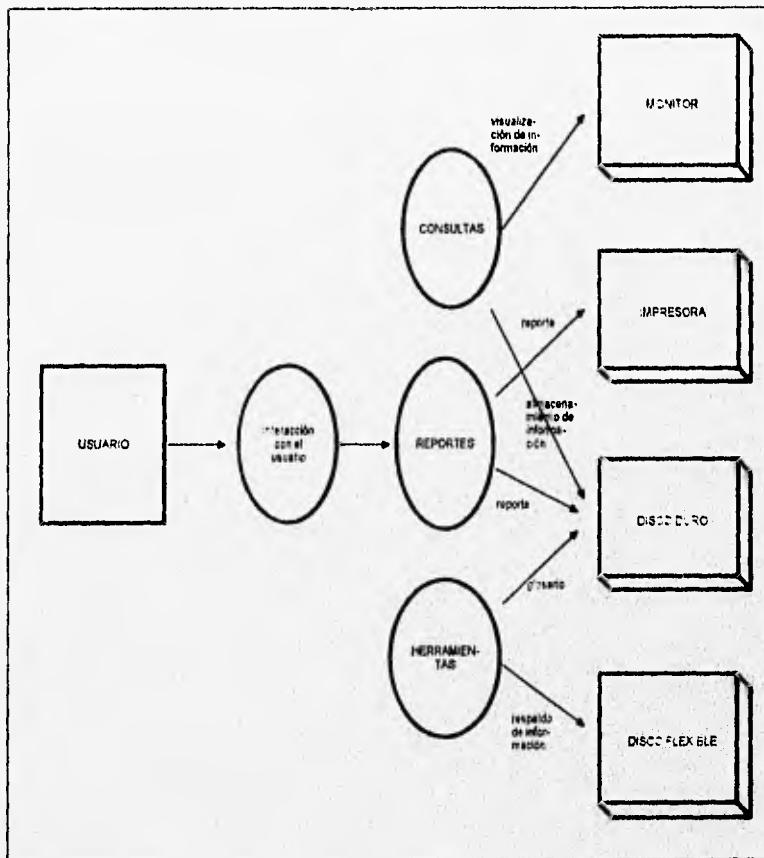


Diagrama de flujo de datos del sistema SISPCNOM nivel 1



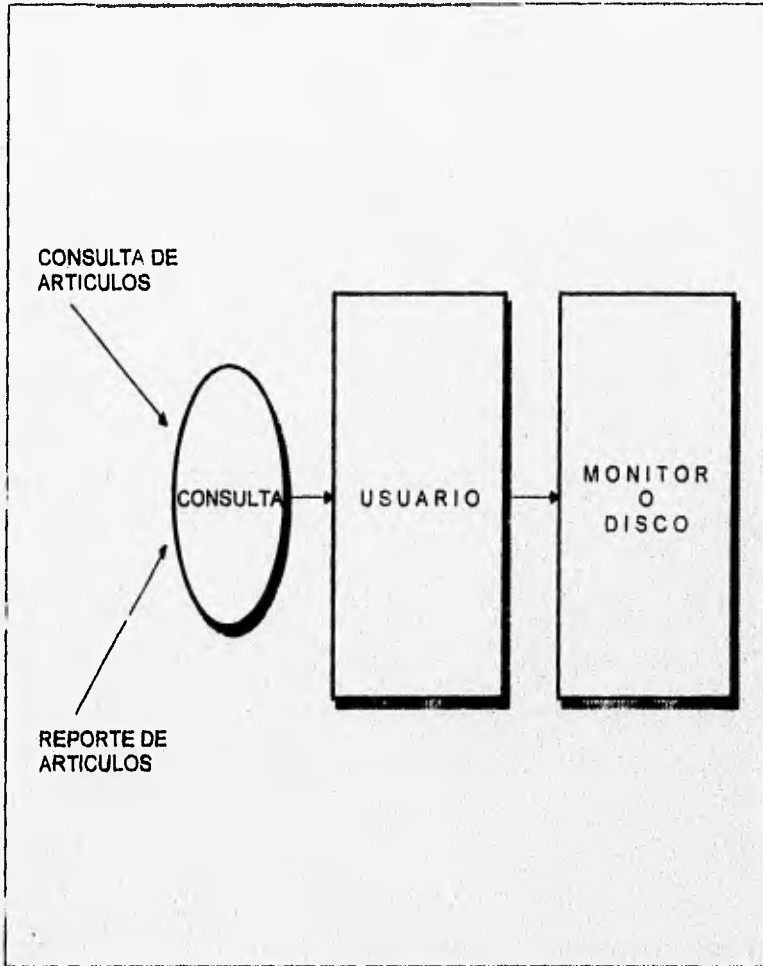


Diagrama de flujo de datos del registro de información



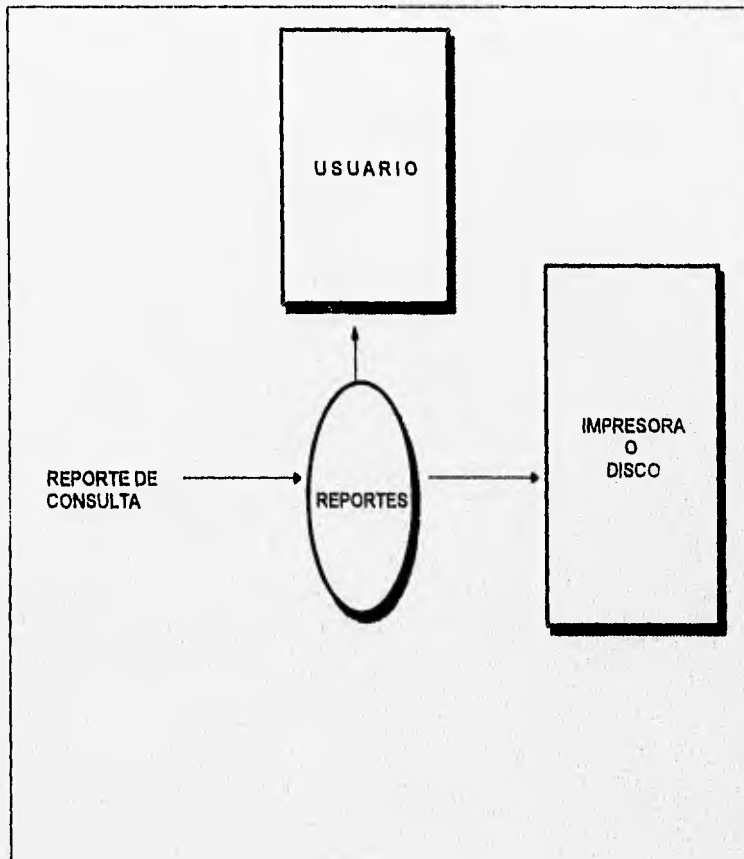


Diagrama de flujo de datos del módulo de reportes



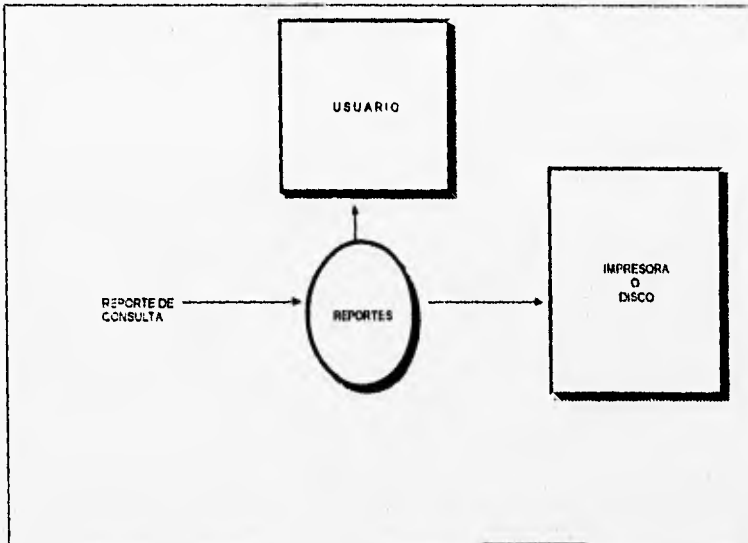


Diagrama de flujo de datos del módulo de reportes

2. DESCRIPCIONES FUNCIONALES.

Las descripciones funcionales de un sistema, son aquellos servicios que el usuario espera del sistema. En principio, las descripciones funcionales de un sistema deben ser completas y consistentes, esto es que ninguna definición debe contradecir a otra. A medida que se descubren los problemas el documento debe modificarse consecuentemente.

Para obtener las descripciones funcionales del sistema, solicitamos una entrevista con el solicitante del sistema para así conocer las necesidades que este tenía, así como de qué forma le gustaría que se solucionarían. Y lo que se obtuvo fue lo siguiente:



- Se requiere de un sistema el cual elimina tiempos de búsqueda en los manuales de dicha norma.
- Un sistema de fácil manejo y amigable para el usuario.
- Que el sistema sea a base de menús.
- Que el sistema realice principalmente una búsqueda por palabras.
- Que al momento de proporcionar la palabra a buscar, el sistema presente una pantalla con todos los artículos en los cuales se hace referencia a la palabra buscada.
- Que posteriormente se pueda acceder a los artículos en los cuales se encontró la palabra.
- Que el sistema emita reportes de un formato específico de los artículos requeridos.

Estos fueron los primeros requerimientos que surgieron de esta visita. Los demás detalles se dejaron a nuestro criterio.

5.4 SEGURIDAD DEL SISTEMA.

La seguridad del software es una actividad de garantía de calidad, se centra en la identificación y evaluación de los riesgos potenciales que puede producir un impacto negativo en el software y hacer que falle el sistema completo. Si se pueden identificar pronto los riesgos en el proceso de ingeniería del software, se podrán identificar las características del diseño de software.

Como parte de la seguridad, se puede dirigir un proceso de análisis y modelización. Inicialmente se identifican los riesgos y se clasifican por su importancia y su grado de riesgo.



Aunque la fiabilidad y la seguridad están muy relacionadas, es importante entender la diferencia entre ellas. La fiabilidad utiliza el análisis estadístico para determinar la probabilidad de que pueda ocurrir un fallo. Sin embargo, la ocurrencia de este no lleva necesariamente a un riesgo. La seguridad del software examina los modos según los cuales los fallos producen condiciones que puedan llevar a errores.

En un sistema de base de datos, la seguridad juega un papel muy importante, ya que se debe garantizar que la información que en ella se almacene sea correcta y sin fallas; en caso de un sistema bancario, este es muy importante ya que la información que maneja es muy delicada, y se debe asegurar la integridad de la información.

La seguridad en el SISPCNOM es buena, la información que se maneja es confiable y no hay posibilidad de un almacenamiento incorrecto.

Otro de los aspectos de seguridad que se deben tomar en cuenta al desarrollar software en particular base de datos, es la existencia de contraseñas y permisos de acceso en distintos niveles para cada tipo de usuario, esto es, limitar según el usuario, la información que tiene derecho a consultar o a modificar.

Por último un aspecto importante en cuanto a la seguridad del software es la relación de copias de seguridad y recuperación de las mismas así como el almacenamiento de grupos de información determinados. por ejemplo: por proyectos o aplicación.

En relación a este último punto el SISPCNOM cuenta con una opción, la cual realiza copias de seguridad (Backup) de las bases de datos que utiliza. Contando así con un respaldo de la información en caso de un accidente.

5.5 DISEÑO DEL SISTEMA.

5.5.1 INTRODUCCIÓN.

El diseño de software es un proceso creativo que requiere del diseñador ciertas cualidades y el diseño final, suele ser una repetición de varios diseños preliminares. Dada una definición de requisitos, el programador debe utilizarla para desarrollar el diseño de un sistema de programación que satisfaga estos requisitos.

Una parte esencial en el proceso de diseño es tener una especificación precisa de lo que se pretende alcanzar con el sistema.

El diseño de software, se divide en dos actividades que son:

DISEÑO EXTERNO:

En esta actividad se conciben, planean y se especifican las características del producto de programación. Estas características incluyen: definición de despliegues de pantalla, formatos de los reportes, definición de las entradas y salidas de datos, así como las características funcionales, los requerimientos de desempeño y la estructura general del producto.

Se puede decir que el diseño externo empieza durante la fase de análisis y continúa ya entrada la fase de diseño general. Además, el diseño externo se encarga del refinamiento de los requisitos planteados en el análisis de requerimientos y de la definición de la estructura del sistema.

DISEÑO INTERNO:

En el diseño interno se incluye la concepción, planeación y especificación de la estructura interna, así como los detalles de proceso del producto de programación.



En resumen, los objetivos del diseño interno son:

- ➔ Especificar la estructura interna
- ➔ Detalles de procesamiento
- ➔ Guardar las decisiones tomadas en el diseño (externo) e indicar el por qué ciertas alternativas y acuerdos fueron aceptados
- ➔ Elaborar los planes de prueba y proporcionar una guía para su instrumentación
- ➔ Pruebas y mantenimiento.

5.5.2 ARQUITECTURA DEL SOFTWARE.

En esta parte del diseño, se busca un refinamiento de la vista conceptual del sistema, identificando funciones internas del proceso, descomposición de funciones de alto nivel en subfunciones y la definición de las cadenas de datos locales y su almacenamiento.

En la arquitectura del sistema se tienen que definir dos características muy importantes del sistema que son:

- ☆ La estructura jerárquica de los componentes procedimentales (módulos).

- ☆ Estructura de datos.

La arquitectura del sistema se obtiene mediante el proceso de partición, que relaciona los elementos de una solución de software con partes de un problema del mundo real definido implícitamente durante el análisis de requerimientos.

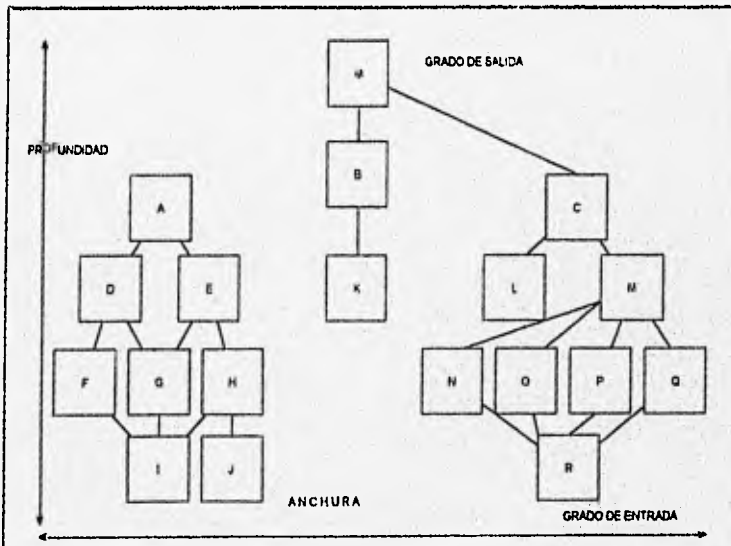


5.5.3 ESTRUCTURA DEL PROGRAMA.

La estructura es una característica fundamental de los productos de programación. El uso de una estructuración permite que un sistema grande sea definido en términos de unidades más pequeñas y manejables con una clara definición de las relaciones entre las diferentes partes del sistema.

Representa una organización (frecuentemente jerárquica) de los componentes del programa (módulos) e implica una jerarquía de control. No presenta aspectos procedimentales del software, tales como la secuencia de procesos.

Para representar la estructura del programa se utiliza en diferentes notaciones. La más común es el diagrama en forma de árbol, sin embargo se pueden utilizar otras notaciones como diagramas de Wamier-Orr y de Jackson. Definiendo los términos de la siguiente figura tenemos:



La profundidad y la anchura son una condición del número de niveles de control y de la amplitud global del control, respectivamente. El grado de salida es una medida del número de módulos que están directamente controlados por otros módulos. El grado de entrada indica cuántos módulos controlan directamente a un módulo dado.

La estructura del programa representa también la visibilidad y la conectividad. La visibilidad indica el conjunto de componentes del programa que pueden ser invocados o utilizados por otro componente dado, la conectividad indica el conjunto de componentes a los que directamente se invoca o utiliza un determinado módulo.

La estructura del sistema "SISPCNOM" es básicamente la siguiente:

- **Consulta de la información.**
- **Reportes.**
- **Herramientas.**

A su vez estos tres procesos se subdividieron en:

• **CONSULTA DE LA INFORMACIÓN:**

- A) Consulta por artículo.
- B) Consulta por palabra.
- C) Consulta por índice temático.

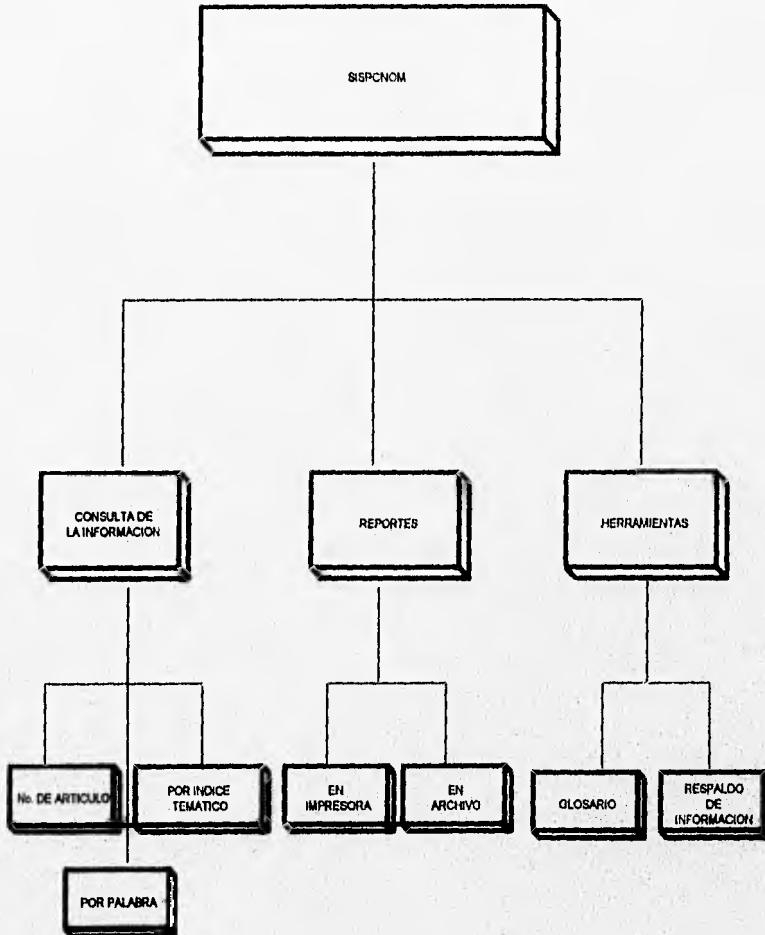
• **REPORTES:**

- A) Reportes en impresora.
- B) Reportes en archivo.

• **HERRAMIENTAS:**

- A) Glosario.
- B) Respaldo de información en disco.





5.5.4 ESTRUCTURA DE DATOS.

El impacto de la estructura de datos sobre la estructura del programa, hace que la estructura de datos tenga una gran influencia para el desarrollo de un software de calidad.

La estructura de datos es una representación lógica existente entre los documentos individuales de datos, debido a que la estructura de información afectará invariablemente el diseño final.

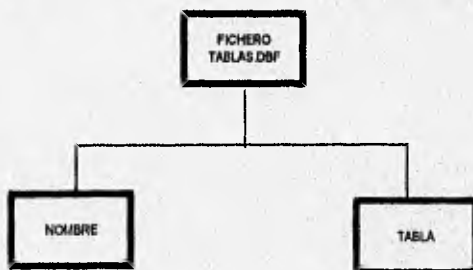
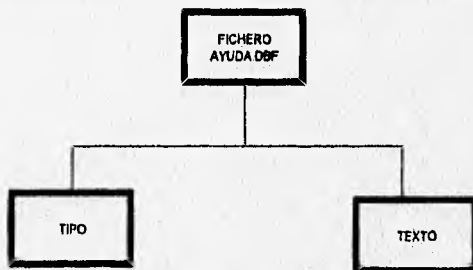
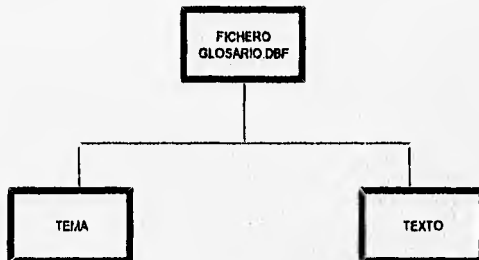
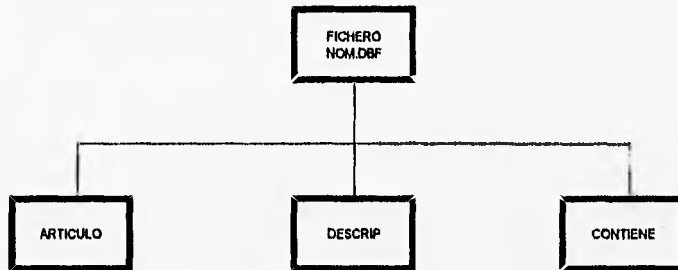
La estructura de datos dicta la organización, los métodos de acceso, el grado de asociatividad y las alternativas de procesamiento para la información. Son las relaciones físicas y lógicas entre los registros de los archivos del ordenador. Una estructura física de los datos es sencillamente el medio de encontrar los datos en un dispositivo de almacenamiento, habitualmente están relacionados de manera lógica, de acuerdo con una secuencia basada en algún campo de claves.

Los diagramas estructurales describen el orden en que aparecen (en un fichero serie) los elementos de datos (requisitos) y los grupos de elementos de datos, hasta donde este orden sea de interés para el programa. Los elementos de datos y los grupos de elementos de los distintos ficheros pueden aparecer en algún orden significativo para el proceso que se requiera.

En la siguiente figura se muestran los diagramas estructurales del sistema SISPCNOM.



DIAGRAMAS ESTRUCTURALES



| CAMPO | DESCRIPCIÓN | FICHERO | TIPO | LONG. |
|----------|---------------------------------------|--------------|----------|-------|
| ARTICULO | NUMERO. DEL ARTICULO PARA LA NOM. | NOM.DBF | CARÁCTER | 4 |
| DESCRIP | TEMA O TITULO DEL ARTICULO DE LA NOM. | NOM.DBF | CARÁCTER | 100 |
| CONTIENE | CONTENIDO DEL ARTICULO DE LA NOM | NOM.DBF | MEMO | |
| TEMA | TEMA DEL GLOSARIO | GLOSARIO.DBF | CARÁCTER | 11 |
| TEXTO | CONTENIDO DEL TEMA DE GLOSARIO | GLOSARIO.DBF | MEMO | |
| TIPO | TEMA DE AYUDA. | AYUDA.DBF | CARÁCTER | 34 |
| TEXTO | CONTENIDO DEL TEMA DE AYUDA. | AYUDA.DBF | MEMO | |
| NOMBRE | NUMERO DE LA TABLA EN NOM | TABLAS.DBF | CARÁCTER | 15 |
| TABLA | CONTENIDO DE LA TABLA | TABLAS.DBF | MEMO | |

5.5.5 MODULARIDAD.

Los sistemas modulares consisten en unidades claramente definidas y manejables con las interfases claramente definidas entre los diversos módulos. Las propiedades deseadas de un sistema modular cumplen con los siguientes criterios:

- ⊗ Cada abstracción de un proceso es un sistema claramente definido y con el potencial de ser útil para otras aplicaciones.
- ⊗ Cada función en cada abstracción tiene un propósito específico, claramente definido.
- ⊗ Cada función maneja no más de una estructura de datos principal del sistema.



- ✦ Las funciones comparten datos globales en forma selectiva. Son ciertamente fácil de identificar todas las rutinas que comparten una estructura de datos principal.
- ✦ Las funciones que manejan las instancias de un tipo abstracto de datos que quedan encapsulados con la estructura de datos en cuestión.

La modulación ciertamente mejora la calidad del diseño, que a su vez facilita la instrumentación y el mantenimiento de un producto de programación.

Un módulo debe procurar cumplir con ciertas características que son:

- ✦ Los módulos pueden ser compilados aparte y almacenados en una biblioteca.
- ✦ Los módulos contienen instrucciones lógicas de procesos y estructuras de datos.
- ✦ Los módulos pueden quedar incluidos dentro de un programa.
- ✦ Los elementos de un módulo pueden ser utilizados por medio de invocar un nombre con algunos parámetros.

Como ejemplo de módulos se incluyen los procedimientos, subrutinas y funciones; así como los grupos funcionales de procedimientos, subrutinas y funciones relacionadas; los grupos de abstracción de datos: los grupos de programas de apoyo y los procesos concurrentes.

La modularización permite al diseñador descomponer un sistema en sus unidades funcionales con el fin de imponer un orden jerárquico en el uso de funciones; igualmente permite la instrumentación de abstracciones de datos y el desarrollo independiente de subsistemas útiles.



Además, la modularización puede ser utilizada para aislar a las dependencias funcionales; para mejorar el desempeño de un producto de programación o para facilitar la depuración, las pruebas, la integración, el ajuste y la modificación de un sistema.

Los módulos convencionales tienen una única entrada y una única salida y ejecutan secuencialmente una tarea.

5.5.6 INDEPENDENCIA FUNCIONAL.

La independencia funcional, se deriva de la modularidad. La independencia funcional se adquiere desarrollando módulos con una "clara" función y una "aversión" a una excesiva interacción con otros módulos. Lo que se persigue con esto, es diseñar software de forma que cada módulo se centre en una subfunción específica de los requisitos y cuente con una interfaz sencilla, cuando se ve desde otras partes de la estructura de software.

Desarrollar la independencia funcional (módulos independientes) es fácil, ya que su función puede ser partida y se simplifican las interfaces. Los módulos independientes son más fáciles de mantener debido a que se limitan los efectos secundarios producidos por las modificaciones en el diseño/código, se reduce la propagación de errores y se fomenta la reutilización de los módulos.

La independencia se mide con dos criterios cualitativos: la cohesión y el acoplamiento. La cohesión es una medida de fortaleza funcional relativa de un módulo. El acoplamiento es una medida de independencia relativa entre módulos.



COHESIÓN.

Un módulo cohesivo ejecuta una tarea sencilla de un procedimiento de software y requiere poca interacción con procedimientos que ejecutan otras partes del programa. En la práctica, el diseñador no tiene que preocuparse por el grado preciso de cohesión de un módulo específico sino que debe comprender el concepto general y evitar los niveles bajos de cohesión en el diseño de los módulos.

Los mejores módulos son aquellos que son funcionalmente cohesivos (módulos en los cuales cada introducción es necesaria para llevar a cabo una tarea), los peores módulos son aquellos que son cohesivos (cuyas instrucciones no tienen relación entre uno y otro).

ACOPLAMIENTO.

La comunicación entre módulos incluye el pasaje de datos, de elementos de control (bandera, interrupciones, etiquetas y nombres de procedimientos), así como de las modificaciones del código de un módulo a otro. El grado de acoplamiento es menor para la comunicación de datos, mayor para la de control y mucho mayor en el caso de módulos que modifican el código de otros módulos. El Acoplamiento es una medida de interconexión entre los módulos de una estructura de programa. Este depende de la complejidad de las interfases entre los módulos, del punto en el que se hace una entrada o referencia a un módulo y de los datos que pasan a través de la interfaz.



MODULARIZACIÓN EN EL SISTEMA SISPCNOM.

En el sistema se definieron siete módulos que a continuación se describen:

| MÓDULO | DESCRIPCIÓN |
|-----------------|--|
| SISPCNOM | <p>Este módulo tiene el programa principal que se encarga de realizar la pantalla principal, las consultas de la información y los tipos de consulta de la información como son:</p> <ul style="list-style-type: none">- Consulta por artículo- Consulta por palabra- Consulta por índice. <p>También contiene todas las funciones utilizadas para mejorar la apariencia y facilitar el uso del sistema.</p> <p>La base de datos que utiliza es NOM.DBF.</p> |
| AYUDA | <p>Este módulo realiza el menú de ayuda, el cual es activado al momento de presionar la tecla F1 y proporciona ayuda en cualquier momento.</p> <p>La base de datos que maneja es AYUDA.DBF.</p> |
| GLOSARIO | <p>Este módulo realiza el menú de glosario.</p> <p>La base de datos que maneja es GLOSARIO.DBF.</p> |



INFORME

Este módulo contiene los procedimientos adecuados para generar un informe del artículo de la NOM de dos formas distintas:

- En impresora
- En archivo.

La base de datos utilizada es NOM.DBF.

RESPALDA

Este módulo contiene los procedimientos para respaldar las bases de datos en disco flexible .

La base de datos empleada es RESPALDA.DBF.

PANTALLA

Este módulo contiene funciones para salvar la pantalla y recuperarla con efectos de movimiento. Cuenta con una función para aumentar a 80 columnas, las letras de un mensaje previamente introducido.

TABLAS

Este módulo contiene un procedimiento para desplegar en pantalla un índice con las tablas contenidas en los artículos de la NOM para seleccionar la que se desea, esto funciona a través de una tecla de atajo, que en este caso es F5.

La base de datos manejada es TABLAS.DBF.



PROCEDIMIENTOS.

| NOMBRE | DESCRIPCIÓN |
|------------------------|--|
| sub_menú | Crea submenús para cada opción del menú principal. |
| flecha_izq | Envía al buffer del teclado el control de flecha izquierda. |
| flecha_der | Envía al buffer del teclado el control de flecha derecha. |
| ligar_flecha | Permite utilizar la tecla de flecha derecha o flecha izquierda para navegar en las ventanas de las diferentes opciones del menú principal. |
| desligar_flecha | Libera las teclas de flecha derecha y flecha izquierda para su uso común. |
| ARTICULO | Realiza la búsqueda de un artículo especificado por el usuario. Base de datos utilizada: NOM.DBF. |
| SALIR | Regresa al ambiente de sistema operativo MS-DOS. |
| BUSCA | Realiza la búsqueda de una palabra en el contenido de la base de datos. Base de datos que utiliza: NOM.DBF. |



- INDICE** Despliega el índice por temas de la norma NOM-001-SEMP-1994 y es posible seleccionarlos.
- Base de datos que utiliza: NOM.DBF.
- HELP** Despliega el índice temático para la ayuda del sistema.
- Base de datos que utiliza: AYUDA.DBF.
- INFORME_A** Crea un archivo que contiene el informe de un artículo determinado.
- Base de datos que utiliza: NOM.DBF.
- INFORME_I** Imprime el informe generado del artículo especificado.
- Base de datos que utiliza: NOM.DBF.
- RESP** Realiza el respaldo de las bases de datos del sistema.
- Base de datos que utiliza: NOM.DBF, AYUDA.DBF, GLOSARIO.DBF.
- VER** Despliega el índice de tablas de la NOM-001-SEMP-1994.
- Base de datos que utiliza: TABLAS.DBF.
- TERMINOS** Despliega el índice de los temas utilizados en el glosario.



FUNCIONES.

| NOMBRE | DESCRIPCIÓN |
|-------------------|---|
| sombra_2 | Crea aun efecto de sombra alrededor de una ventana ya creada. |
| cuadro | Crea una ventana con las coordenadas especificadas, con un color de fondo y algún carácter de relleno en el caso de especificarlo. |
| moldura | Crea una moldura para la ventana de texto. |
| inicio | Crea la pantalla principal del sistema, especificando la hora y fecha actual. |
| mensaje | Despliega un mensaje de error durante un tiempo determinado. |
| TERMOH | Despliega un termómetro que indica un porcentaje de avance en la búsqueda por palabra en la base de datos. Base de datos que utiliza: NOM.DBF. |
| MUESTRAREC | Despliega un cuadro en el que aparece el número de registros procesados durante la búsqueda por palabra en la base de datos. Base de datos que utiliza: NOM.DBF. |
| LOGO | Aumenta a 80 columnas el tamaño de un carácter introducido anteriormente. Pueden aparecer hasta 8 caracteres. |



- SAVHOR2** Guarda la pantalla actual para su uso posterior por la función RECHOR2.
- RECHOR2** Devuelve una pantalla guardada por la función SAVHOR2, dividida horizontalmente en dos mitades.
- SAVVER2** Guarda la pantalla actual para su uso posterior por la función RECVER2.
- RECVER2** Devuelve una pantalla guardada por la función SAVVER2, dividida verticalmente en dos mitades.

5.5.7 DIAGRAMAS DE FLUJO.

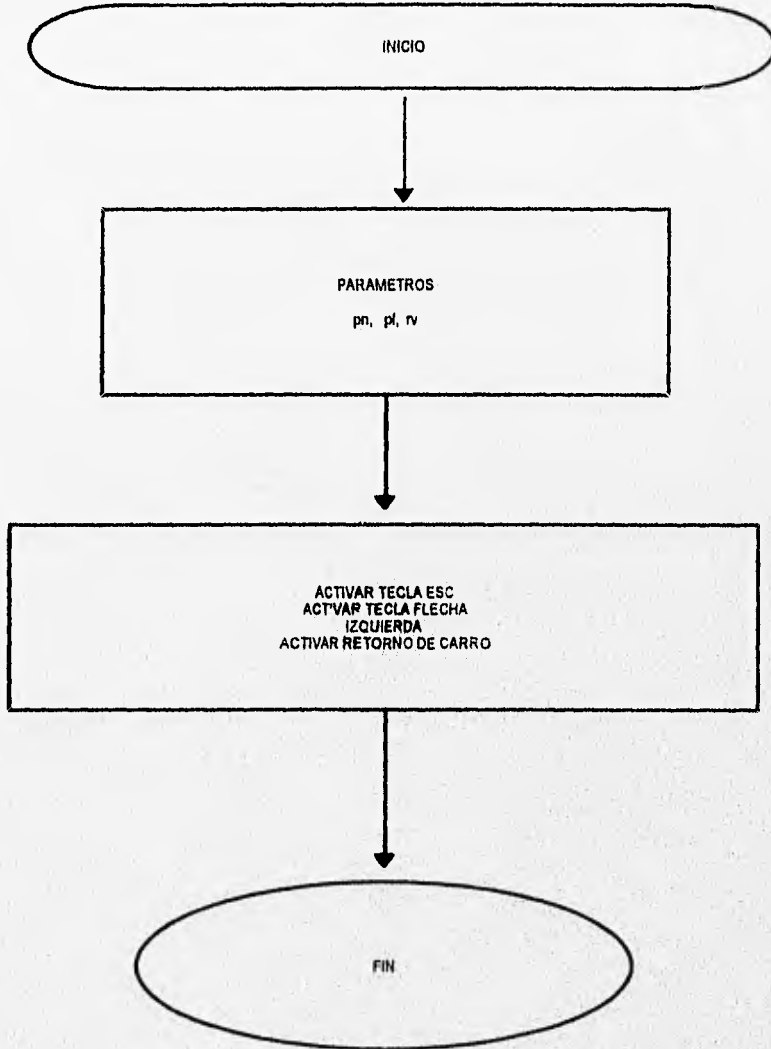
Los diagramas de flujo son la representación gráfica más ampliamente usada para el diseño modular, los cuales representan los detalles algorítmicos de un programa. El diagrama de flujo es un gráfico muy sencillo; estos diagramas utilizan cajas rectangulares para especificar las acciones, cajas en forma de rombos para las proposiciones de decisión, arcos dirigidos para las interconexiones entre las diversas cajas, así como una variedad de formas especiales para denotar las entradas, las salidas, los almacenamientos, etcétera.

Las formas básicas se caracterizan por una entrada única y una salida única para cada una de las formas.

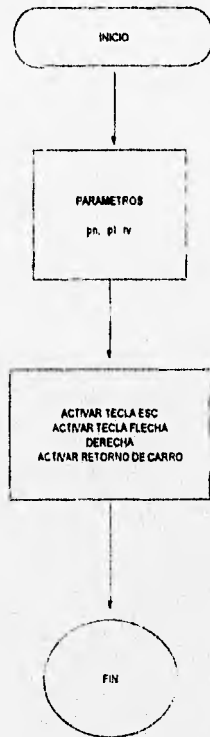
Los diagramas de flujo de los programas principales del sistema SISPCNOM son los siguientes :



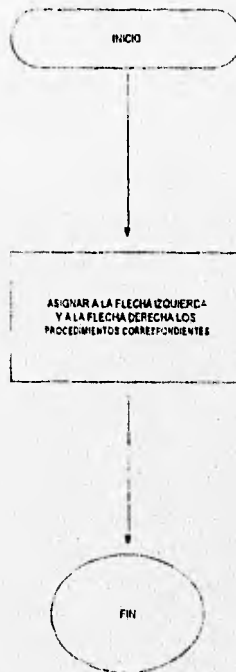
PROCEDIMIENTO FLECHA_IZQ.



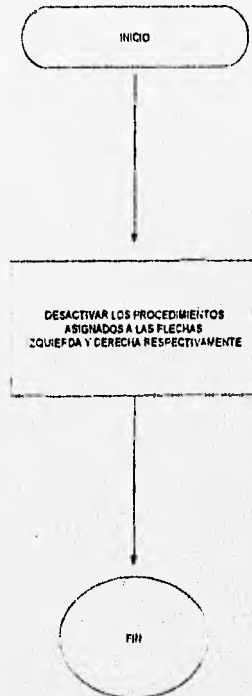
PROCEDIMIENTO FLECHA_DER



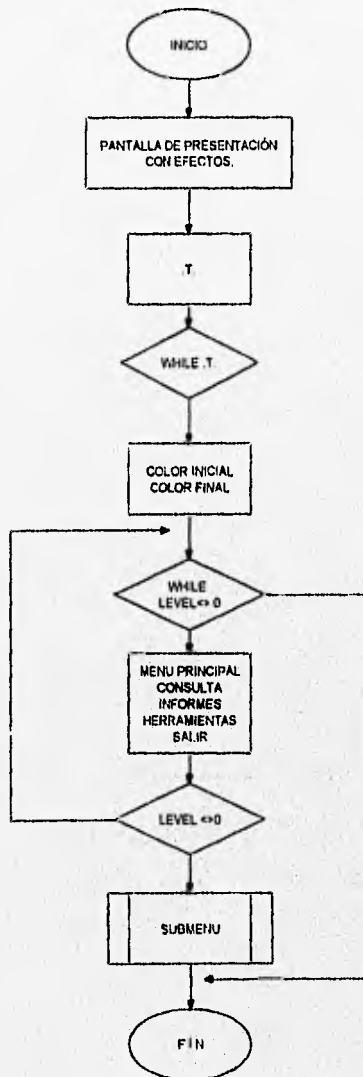
PROCEDIMIENTO LIGAR FLECHA



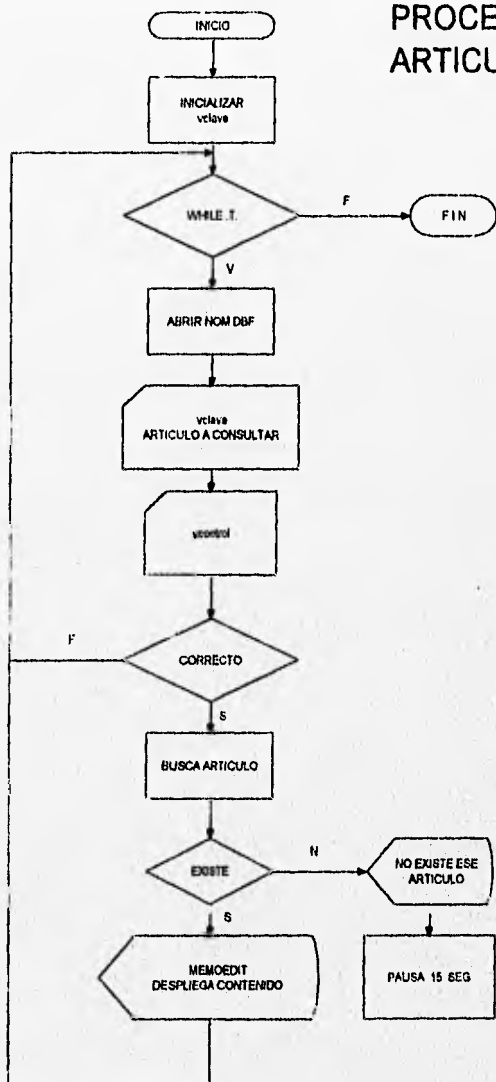
PROCEDIMIENTO DESLIGAR FLECHA



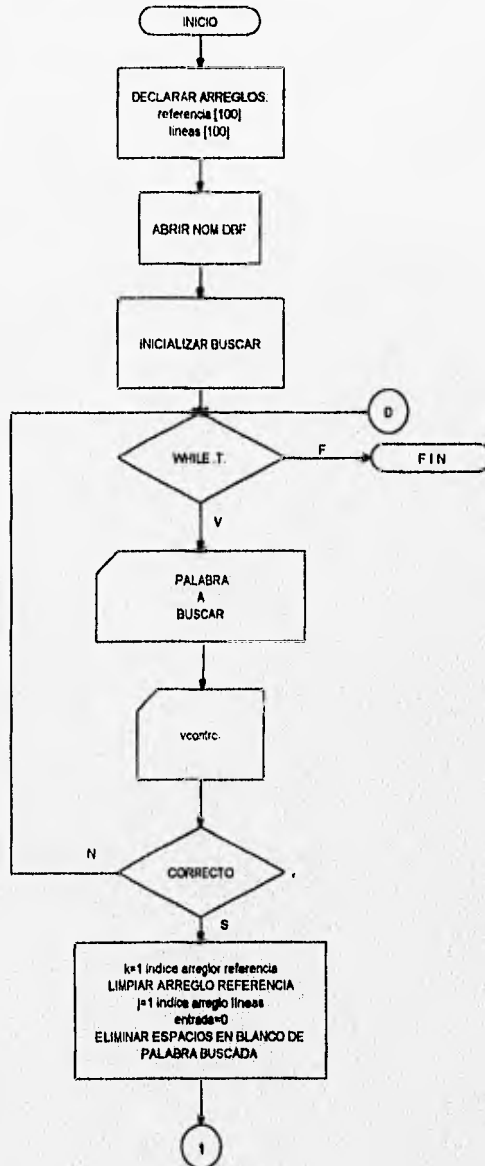
MODULO PRINCIPAL

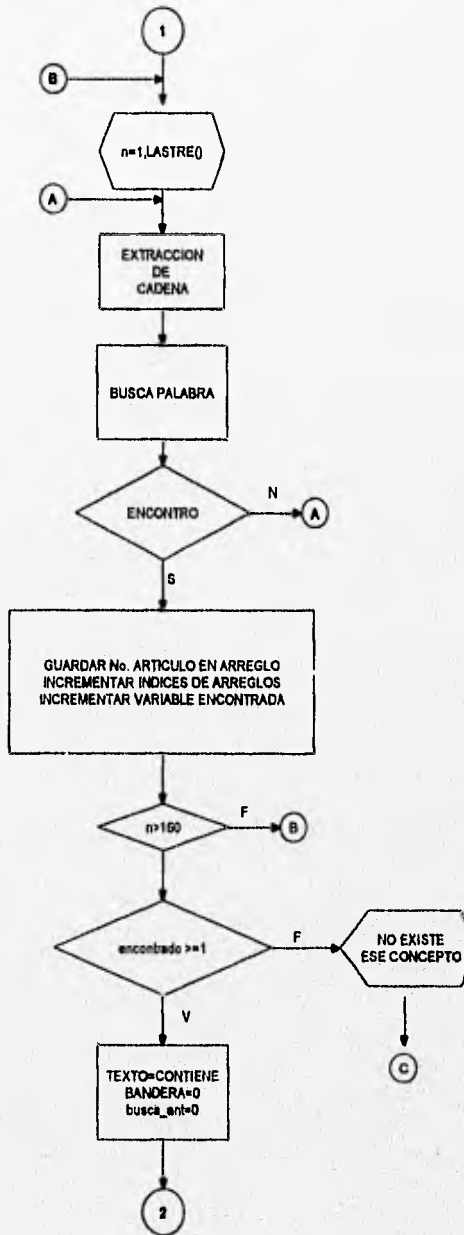


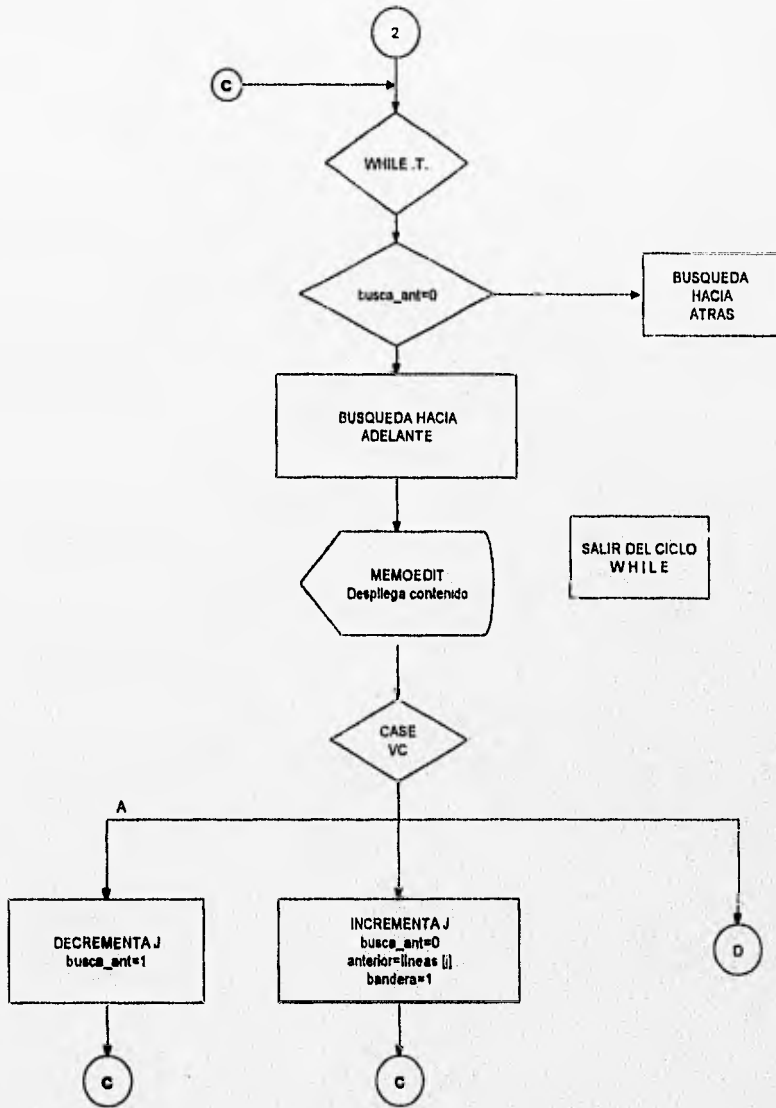
PROCEDIMIENTO
ARTICULO.



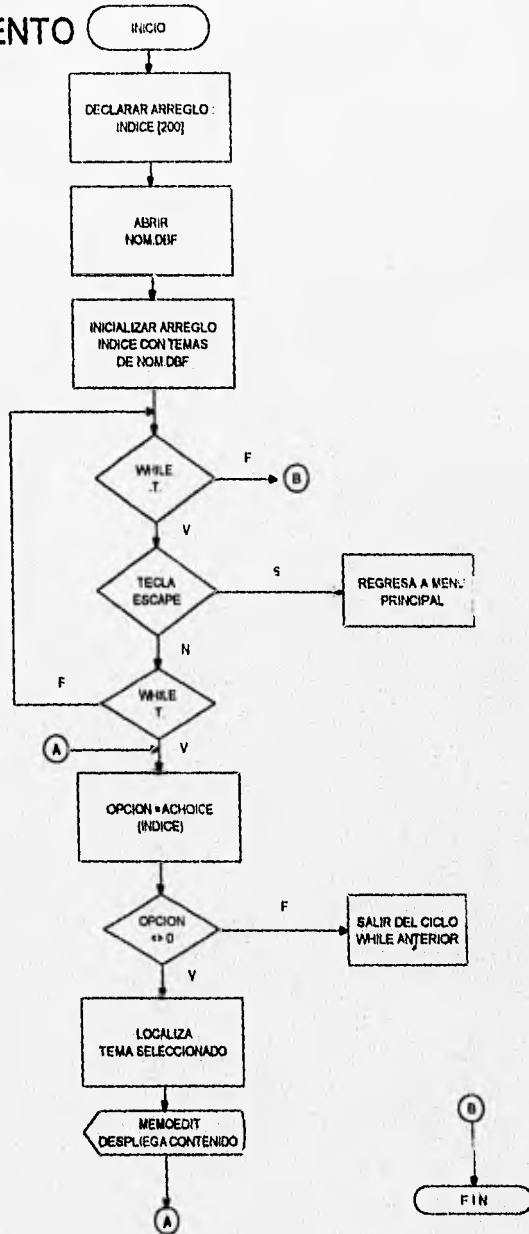
PROCEDIMIENTO BUSCA



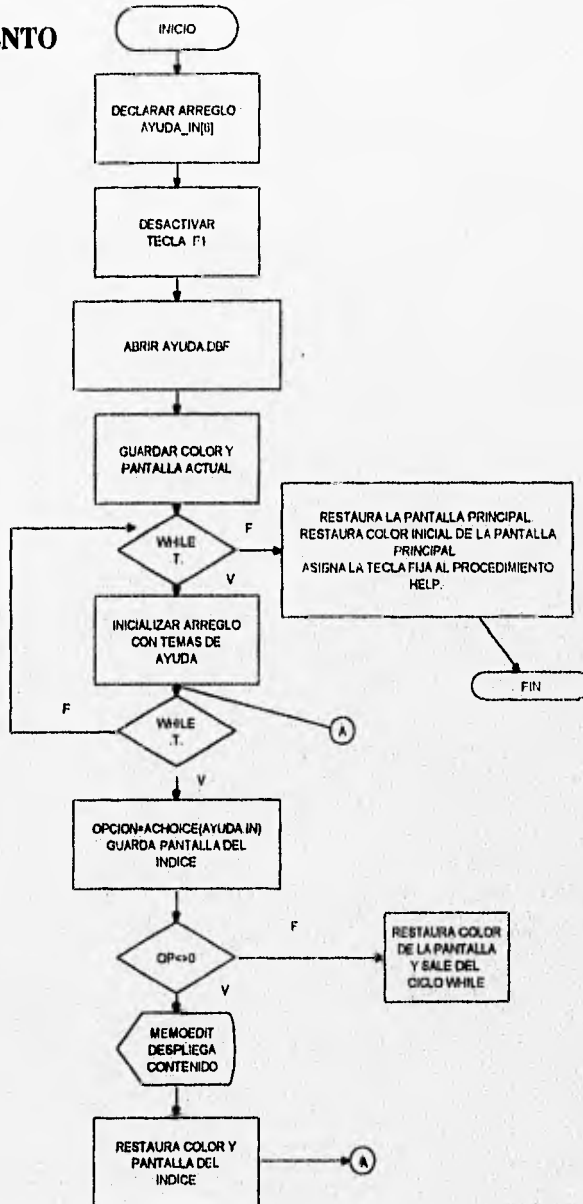




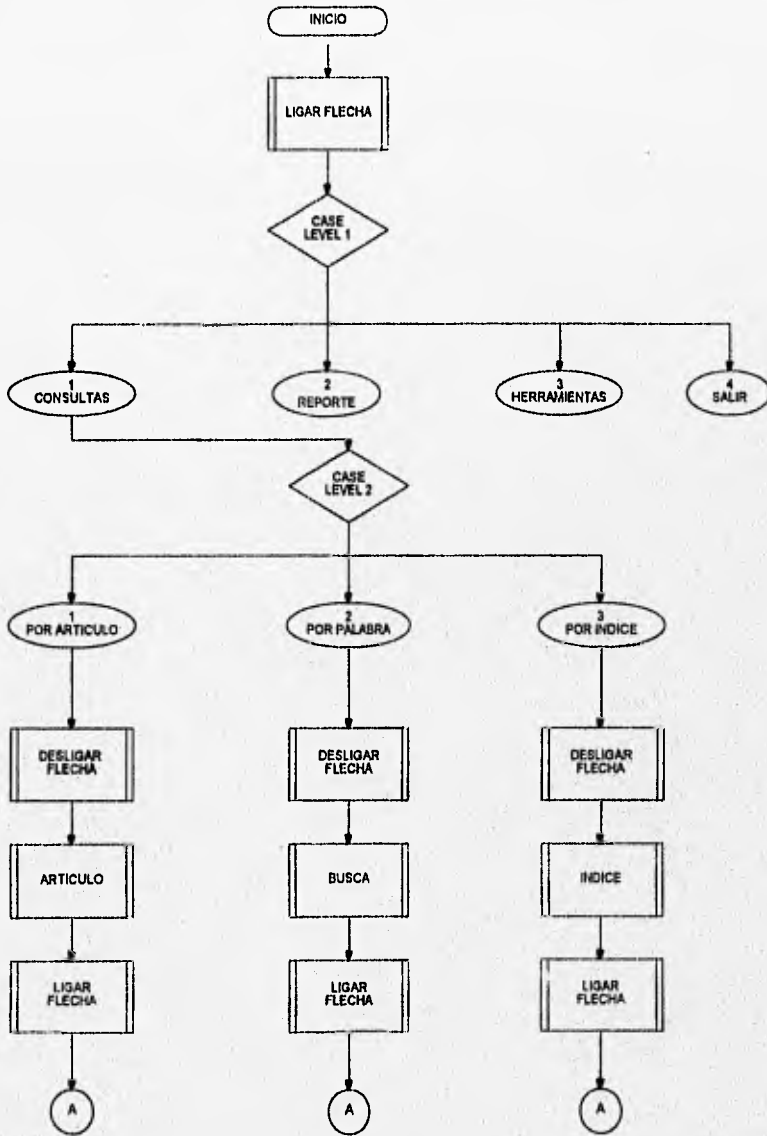
PROCEDIMIENTO
INDICE.

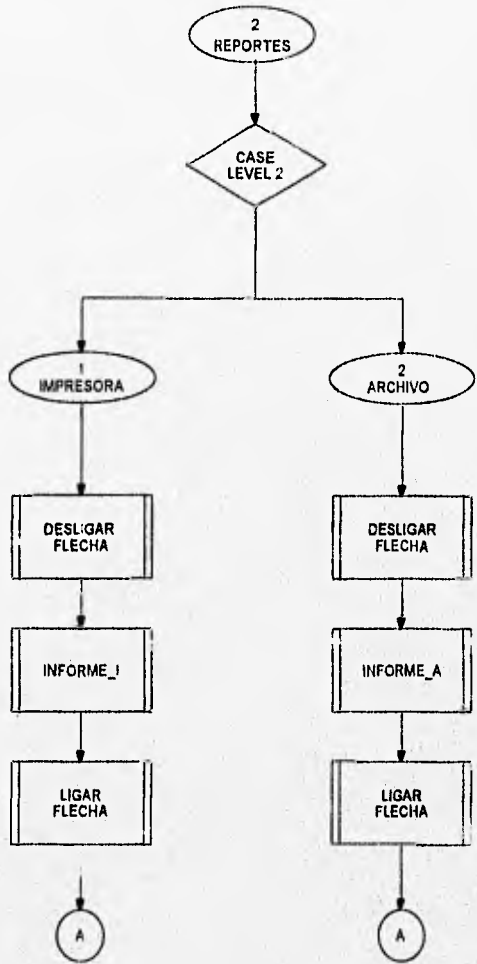


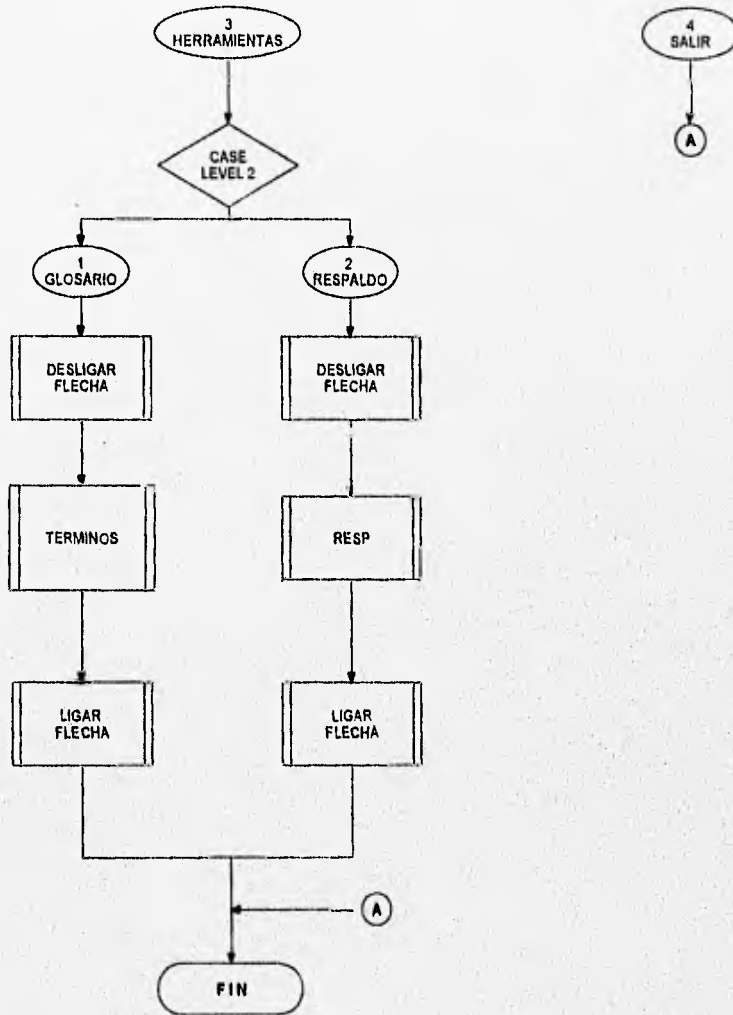
**PROCEDIMIENTO
HELP.**



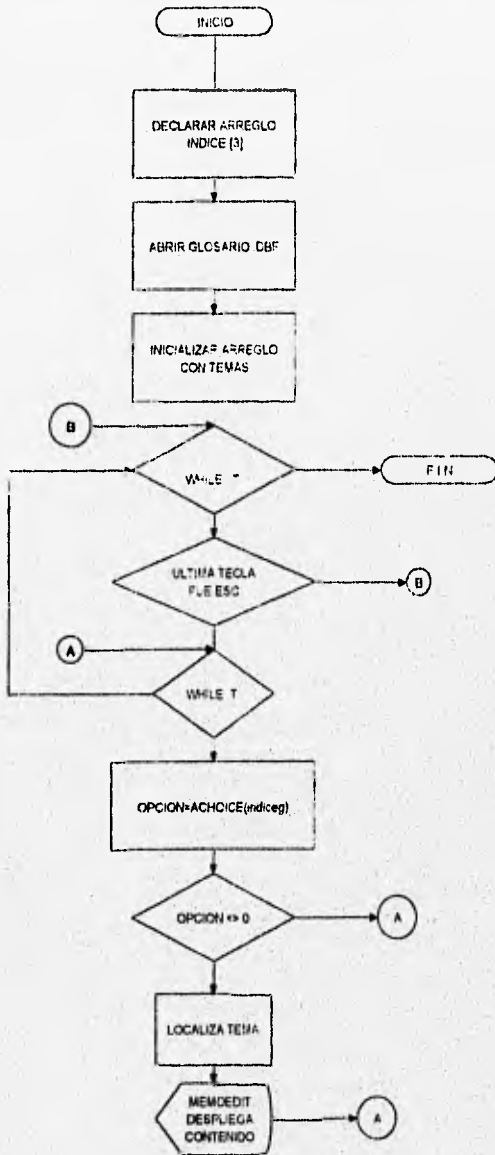
PROCEDIMIENTO SUBMENU



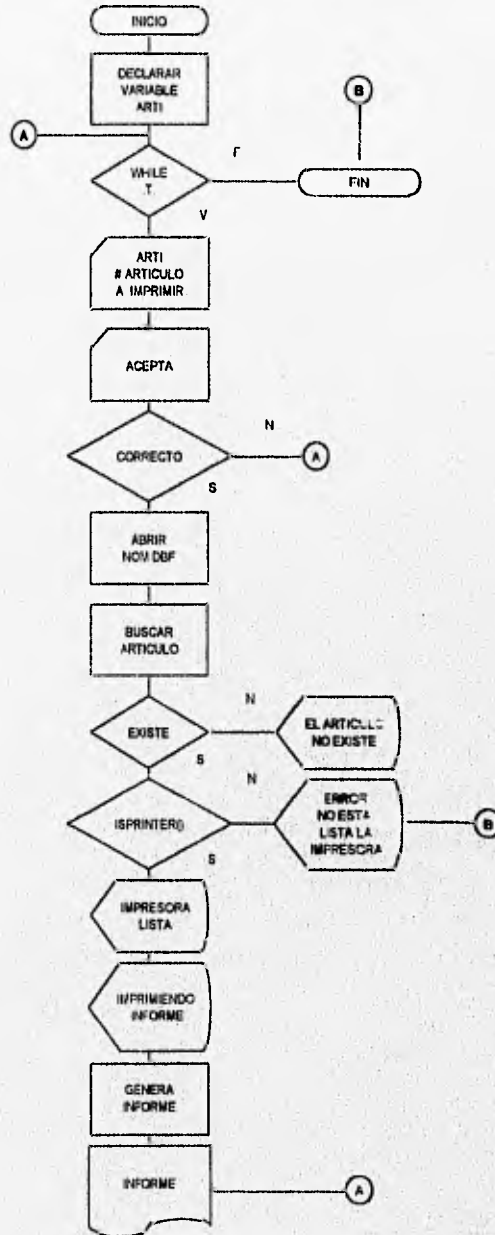




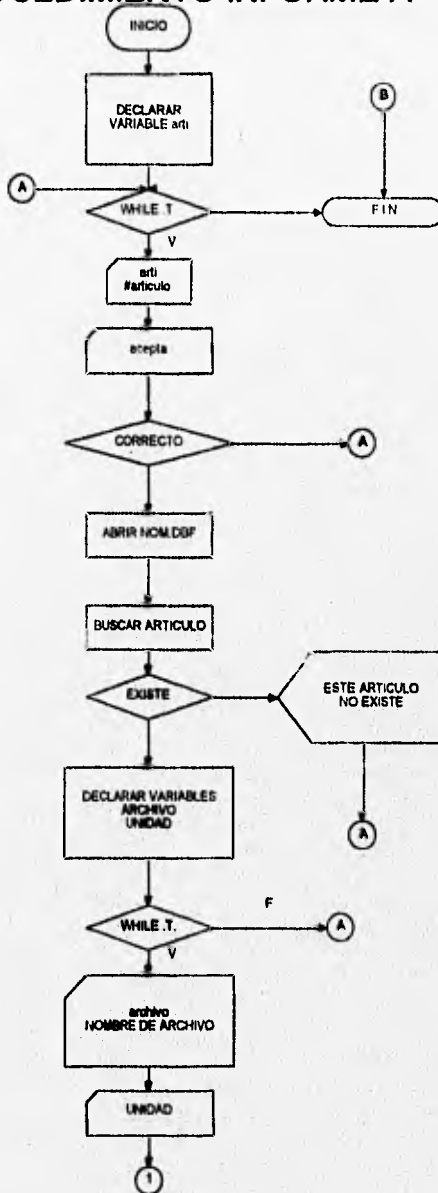
PROCEDIMIENTO TERMINO

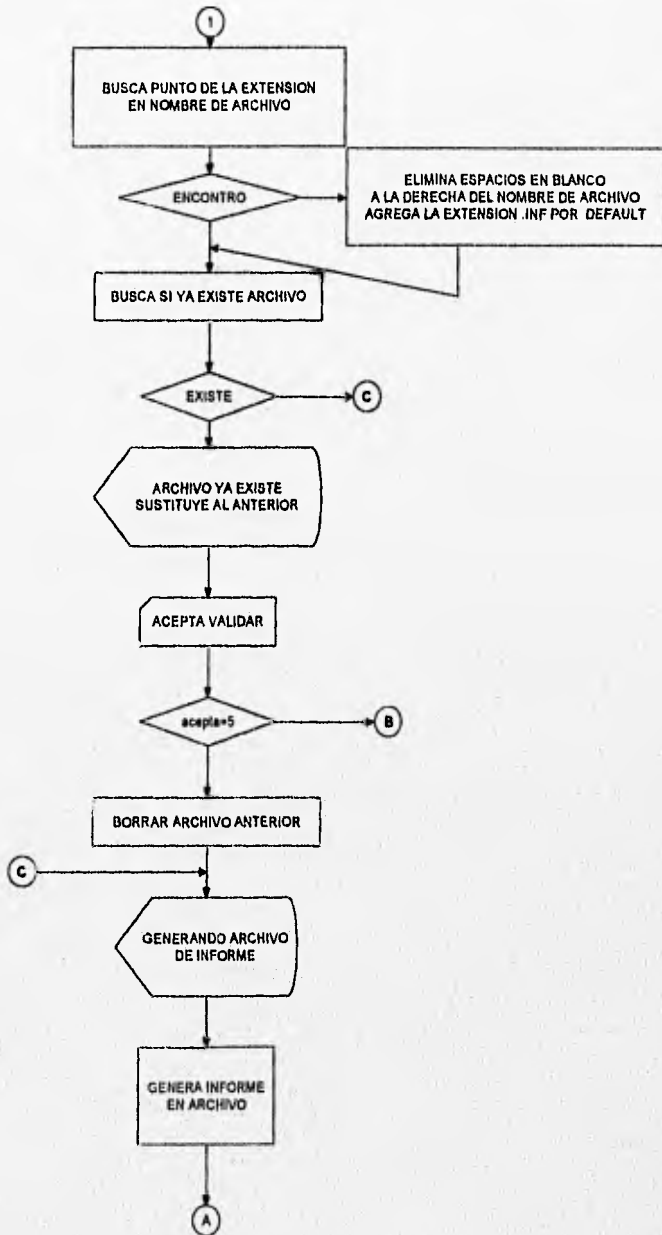


**PROCEDIMIENTO
INFORME_I**



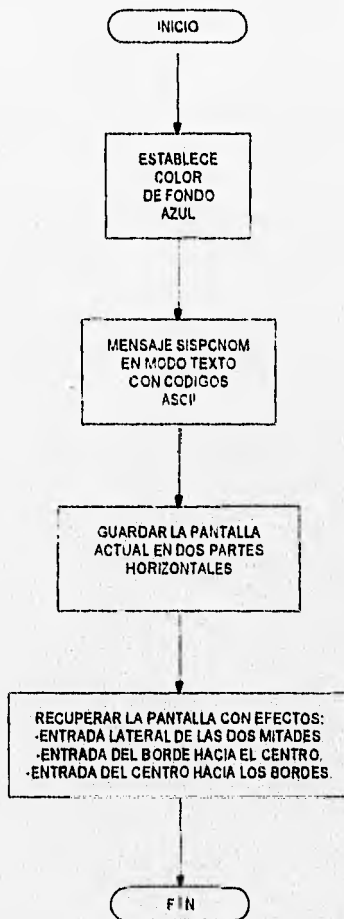
PROCEDIMIENTO INFORME-A



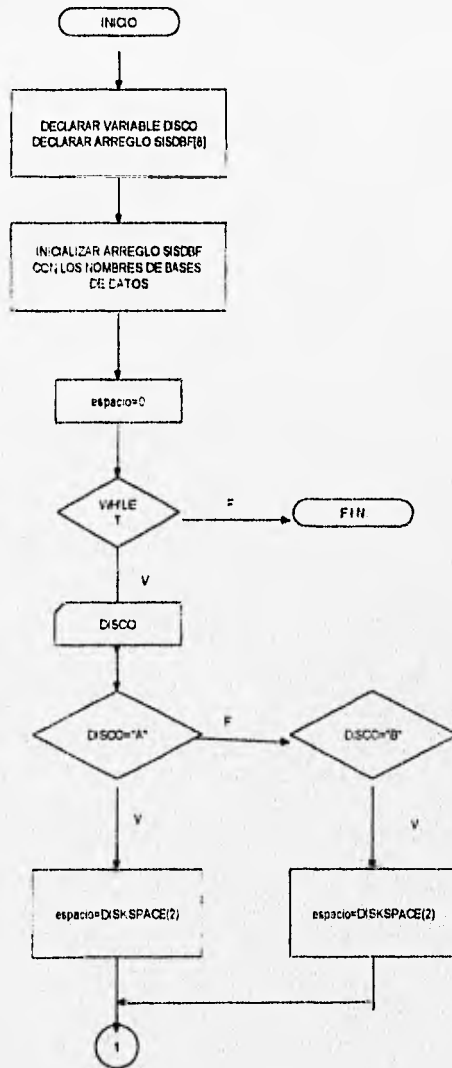


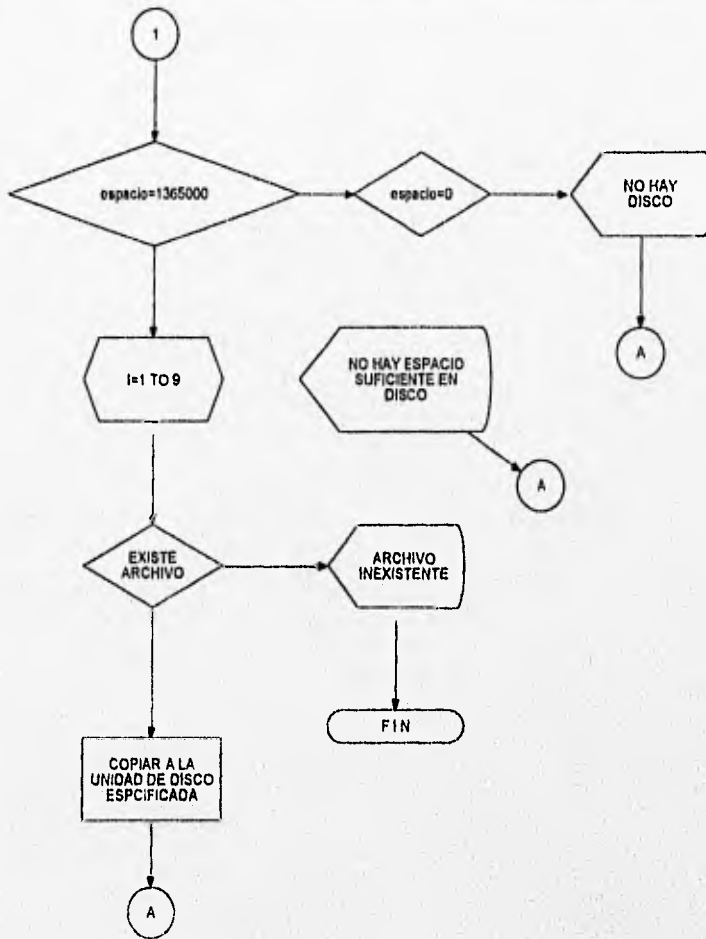
PROCEDIMIENTO SALIR



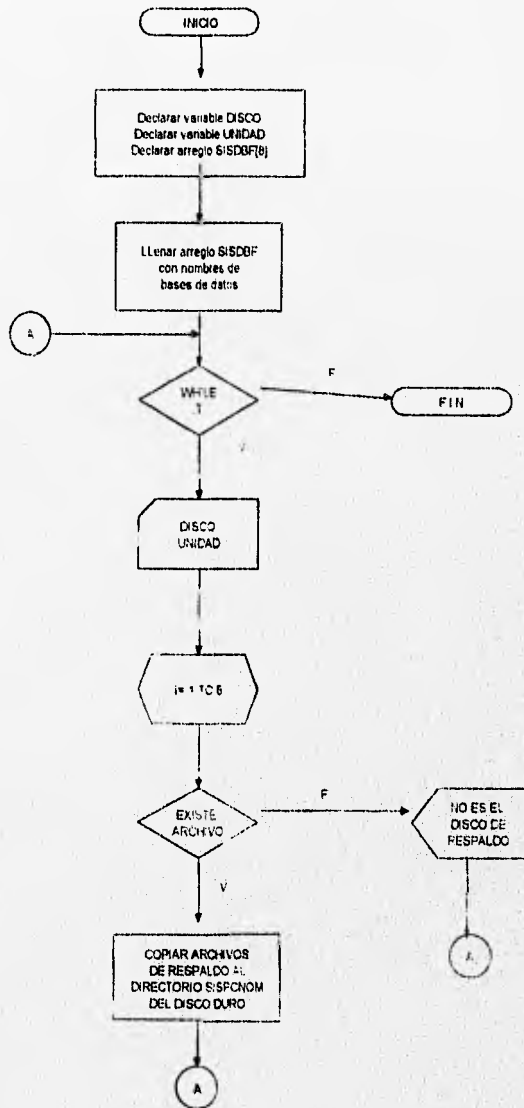
MODULO PANTALLA.PRG

MODULO RESPALDA.PRG. PROCEDURE RESP.





UTILERIA RECUPERA.PRG.



5.5.8 SELECCIÓN DE SOFTWARE.

El paso de codificación traduce una representación del software, dada por un diseño detallado, a una realización en el lenguaje de programación. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina más tarde la salida del compilador es traducida a código máquina.

El proceso inicial de traducción es un punto fundamental. La interpretación equivocada de las especificaciones del diseño detallado puede conducir a un código fuente erróneo. La complejidad o las restricciones de un lenguaje de programación darán como resultado un código difícil de mantener y probar.

Las características de los lenguajes de programación se centra en las necesidades que puede tener un proyecto en específico de desarrollo de software, se puede establecer un conjunto general de características :

- ♦ Facilidad de traducción del diseño al código
- ♦ Eficiencia del compilador
- ♦ Portabilidad del código fuente
- ♦ Disponibilidad de herramientas de desarrollo
- ♦ Facilidad de mantenimiento.

El paso de codificación comienza tras haber definido, revisado y modificado el diseño detallado, las críticas actuales a los compiladores de lenguaje de alto nivel van dirigidas a la capacidad de producir código ejecutable rápido y ajustado.



La portabilidad del código fuente es una característica de los lenguajes de programación, que puede interpretarse de tres formas:

- El código fuente puede ser transportado de un procesador a otro y de un compilador a otro sin ninguna o muy pocas modificaciones
- El código fuente permanece inalterado cuando cambia su entorno de funcionamiento
- El código fuente puede ser inalterado en diversos paquetes de software sin que prácticamente se requieran modificaciones debidas a las características propias del lenguaje de programación.

La disponibilidad de herramientas de desarrollo puede acortar el tiempo requerido para la generación del código fuente y puede generar la calidad del código, muchos lenguajes de programación pueden ser adquiridos con un conjunto de herramientas que incluyen :

- + Compiladores con depuradores
- + Ayuda de formato para el código fuente
- + Facilidades de edición incorporadas
- + Herramientas para el control de código fuente
- + Correctores , etcétera.

La elección de un lenguaje de programación para un proyecto específico debe tener en cuenta las características anteriores, el problema asociado con la elección puede desaparecer si solo se dispone de un lenguaje o si el cliente demanda uno en particular.

Entre los criterios que se aplican durante la evaluación de los lenguajes disponibles están :



- o Área de aplicación general
- o Complejidad computacional
- o Entorno en el que se ejecutará el software
- o Consideraciones de rendimiento
- o Complejidad de las estructuras de datos
- o Disponibilidad de un buen compilador.

El lenguaje de programación tendrá impacto en la planificación, el análisis, el diseño, la codificación, la prueba y el mantenimiento de un proyecto. El papel de un lenguaje de programación ha de tenerse presente en todo momento, ya que estos proporcionan los medios de traducción hombre-máquina.

ANÁLISIS DE OPCIONES DE SOFTWARE PARA EL DESARROLLO DEL SISTEMA SISPCNOM.

Inicialmente se plantearon todas las posibles soluciones. Surgió entonces la primera disyuntiva, hacerlo en Windows o en DOS. El sistema SISPCNOM, se pudo haber desarrollado en el ambiente gráfico de Windows, sin embargo, los requerimientos básicos para ser ejecutados serían mucho mayores que si se realizaba en el entorno de DOS. Después de analizar el usuario las ventajas y desventajas que pudieran existir al emplear cualquiera de los dos ambientes a desarrollar, solicitó que el sistema se hiciera bajo DOS, ya que se necesitaba que el sistema tuviera la mayor portabilidad posible y un mínimo de requerimientos debido a que lo iba a utilizar en diversas PC's.

Los lenguajes de DOS que se analizaron fueron: FoxPro, SQL, Visual C++, Visual Basic, Pascal y Clipper.



Un programa intérprete del tipo dBase III, dBase IV, FoxBase, FoxPro, se caracteriza porque traduce las instrucciones a la computadora línea a línea, y aunque poseen gran versatilidad para realizar programas, la velocidad de ejecución suele ser muy lenta al estar obligado a traducir simultáneamente dichas instrucciones. Se pensó también en SQL, pero existe la desventaja de que requiere un lenguaje huésped (C, Pascal, Cobol, etcétera) y además tener el paquete (SQL) instalado para que pueda ejecutar el sistema.

Los lenguajes de programación como Visual C++ , Visual Basic y Pascal, aunque pueden ser útiles para implementar sistemas de bases de datos, no tienen instrucciones propias para crear y manejar bases de datos.

Con base en lo anterior, el software que se eligió para desarrollar el sistema SISPCNOM fue el compilador CLIPPER 5.2 ya que es un lenguaje de fácil manejo, el cual genera archivos ejecutables (.EXE). Con este tipo de archivos puede operar un sistema sin necesidad de tener el código fuente y el paquete instalado.

Clipper como es un compilador checa la sintaxis de cada línea; si encuentra un error, despliega un mensaje asociado a este sin ejecutar el programa a diferencia de un intérprete, que ejecuta el programa y se detienen en cada línea donde exista un error el cual es corregido, pero teniendo que volver ejecutar el programa hasta totalizarlo, repitiendo el proceso las veces que sea necesario.

Clipper es un sistema completo que contienen todas las facilidades requeridas para el desarrollo de aplicaciones, como las que se llevan a cabo en el lenguaje de programación dBASE III Plus, conteniendo casi todos los comandos propios de este, así como otros extras que permiten algunos beneficios adicionales como:

- Manejo rápido y eficiente de las bases de datos.
- Rapidez de ejecución en el programa compilado (ejecutable).
- Privacidad en los programas, ya que quedan en código máquina y por lo tanto no pueden ser alterados por el usuario final.



- Ejecución desde el Sistema Operativo.

Otros de los motivos principales por el cual se eligió es que permite el desarrollo de sistemas complejos y posibilita el uso óptimo de memoria RAM a través de Solapamiento Dinámico.

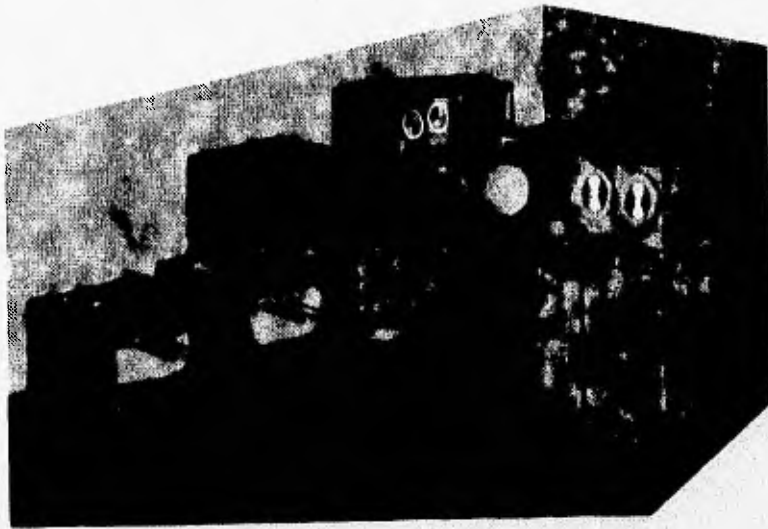
El Solapamiento Dinámico permite ejecutar programas mayores que el tamaño de la memoria disponible, metiéndolos y sacándolos de memoria a medida que se va necesitando. Cuando se realiza una llamada a una función, el administrador de solapamiento de Clipper (Overlays) examina si dicha función ya esta en memoria. Si no lo esta, la carga desde el disco duro y reemplaza parte del código que no se esta utilizando en ese momento.

En el capítulo 3 se dió una introducción sobre este lenguaje de programación, concluyendo de acuerdo con lo descrito anteriormente que este lenguaje es el que llena todos los requisitos para desarrollar el sistema SISPCNOM.



CAPÍTULO 6

IMPLANTACIÓN Y MANTENIMIENTO DEL SISTEMA



CAPITULO 6 IMPLANTACIÓN Y MANTENIMIENTO DEL SISTEMA

6.1 INTRODUCCIÓN.

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son enormes. Los errores pueden darse desde el primer momento del proceso.

La prueba de software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

Las pruebas del sistema implican dos clases de actividades : Pruebas de Integración y de Aceptación. Las estrategias para integrar los componentes del software en un producto que funcione incluye las estrategias ascendentes, descendentes y de emparedado. La estrategia de integración dicta el orden en que los módulos deben de estar disponibles.

Las pruebas de aceptación implican la planeación y ejecución de pruebas funcionales, de desempeño y de tensión para verificar que el sistema realizado satisfaga sus requisitos.

Otra de las actividades importantes en el desarrollo de software es el mantenimiento, este término se usa para describir las actividades de la ingeniería de software que ocurren después de entregar un producto al cliente.

Las actividades de mantenimiento implican mejorar los productos del software, adaptarlos a nuevos ambientes y corregir problemas.



6.2 IMPLANTACIÓN DEL SISTEMA.

Una vez concluidas las etapas de análisis y diseño del sistema, se procede a la implantación del sistema. Esto nos lleva a realizar las pruebas para verificar que las necesidades y requerimientos del cliente sean cubiertos por el software así como descubrir fallas del mismo.

Los objetivos a los que se orientan las pruebas se pueden definir en tres formas:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta posibilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

La prueba de integración es un técnica sistemática para construir la estructura de un programa mientras que, al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la integración.

Para las pruebas de integridad existen tres estrategias que son :

■) La prueba de integridad ascendente, es una de las más empleadas para la integración de los módulos de un sistema, y así formar un todo. esta consiste en pruebas de unidad, cuyo objetivo es encontrar errores en los módulos individuales del sistema, estos módulos se prueban aislándolos, proporcionándoles los datos necesarios para su ejecución. Después de las pruebas de unidad se realizan las pruebas de subsistemas (un subsistema es un conjunto de módulos). El propósito de estas pruebas es la verificación de las operaciones entre los módulos del sistema. Y por último se realizan las pruebas del sistema completo, estas pruebas están relacionadas con: la lógica de decisión, el flujo de control, los procedimientos, recuperación, así como la capacidad y las características de tiempo del sistema.



b) La integración descendente empieza con la rutina principal y una o dos rutinas inmediatamente subordinadas en la estructura del sistema. Después de que este "esqueleto" de alto nivel ha sido probado con detenimiento, se convierte en el arreo de prueba para sus subrutinas inmediatamente subordinadas. La integración de alto nivel requiere del uso de troncos de programas para simular el efecto de las rutinas de más bajo nivel que son llamadas por las rutinas de prueba.

Este tipo de Integración ofrece varias ventajas:

- La integración del sistema se distribuye en todas las fases de implementación. Los módulos se integran a medida que se desarrollan.
- Las interfases del nivel más alto se prueban primero con más frecuencia.
- Las rutinas del nivel más alto proporcionan un arreo de pruebas natural para las rutinas inferiores.
- Los errores se localizan en los nuevos módulos e interfaces que se estén añadiendo.

También este tipo de integración tiene desventajas:

- Puede ser difícil encontrar datos de entrada del nivel más alto que ejercite a un módulo de más bajo nivel en la forma particular deseada.
- Puede ser costoso tanto en tiempo, como en dinero la compilación y el ligar el sistema cada vez que se le agrega una línea. Este defecto se puede eliminar si cada rutina es eliminada antes de integrarla al sistema.

c) La integración enparejado es predominantemente descendente, pero las técnicas ascendentes se usan en algunos módulos y subsistemas. Esta mezcla mitiga muchos de los problemas encontrados en las pruebas descendentes puras y además refina las ventajas de la integración descendente al nivel de subsistemas y de sistemas.



En la segunda ocasión, propusimos que se agregara al sistema una búsqueda por número de artículo y hacer un módulo que pudiera respaldar toda la información que contiene el sistema. Ambas propuestas fueron aceptadas por el usuario final.

En la tercera entrevista, el usuario nos propuso una nueva forma de búsqueda, la cual consistía en una búsqueda por índice temático y se sugirió poner una ayuda para facilitar la utilización del sistema.

En la cuarta ocasión, se sugirió crear el módulo de herramientas, el cual consistía en respaldar la información en disco flexible y crear un glosario de términos para que el usuario consultará algunos términos de computación, instalaciones eléctricas y normas.

En la quinta ocasión, se realizó una prueba con el sistema terminado. Con esta prueba, el usuario quedó satisfecho de la presentación, funcionalidad y eficiencia del sistema.

La última prueba de aceptación se llevó a cabo con varias personas que desconocían el sistema y se les pidió que realizaran diferentes tipos de consulta (sin manual) y que utilizarán si creían necesarios todos los módulos del sistema. Esta prueba fue exitosa, ya que no se presentaron problemas en cuanto a la dificultad del manejo del sistema.

6.2.2 CAPACITACIÓN.

Otras de las actividades que forma el desarrollo del sistema, en la etapa de implantación es la capacitación del personal que usará el sistema.

El sistema SISPCNOM se diseñó de tal forma que este fuera amigable al usuario, contando con mensajes informativos.

Aún cuando el sistema es fácil de manejar, al entregar el sistema, se dio capacitación al usuario principal del sistema, esta capacitación abarcó:



- ❖ Cómo consultar artículos (por número de artículo, por palabra y por índice temático).
- ❖ Cómo emitir reportes (en impresora o en disco).
- ❖ Manejo de herramientas (glosario y respaldo de información).
- ❖ Uso de la ayuda y teclas de atajo.

Dentro de la consulta de artículos se explicó la forma en que se debía seleccionar la opción de consulta deseada, ya sea por número de artículo, por palabra y por índice temático.

En cuanto a la emisión de reportes, se capacitó al usuario para poder enviar reportes a la impresora o bien si se deseaba mandar algún artículo por medio de un archivo a un disco flexible o a disco duro.

Referente al manejo de las herramientas se enseñó como realizar el respaldo de información de todo el sistema a discos flexibles y cómo poder utilizar los glosarios de ayuda del sistema.

Finalmente, se capacitó en cómo, cuándo y dónde se podía hacer uso de la ayuda y de las teclas de atajo que se crearon para facilitar el uso del sistema.

6.3 *DEPURACIÓN DEL SISTEMA.*

La depuración aparece como una consecuencia de una prueba efectiva, es decir, cuando se descubre un error, la depuración es el proceso que resulta en la eliminación de este.

La depuración no es una tarea de prueba, aunque siempre se da como una consecuencia de la misma. El proceso inicia con la ejecución de pruebas, se evalúan los resultados y aparece una falta de correspondencia entre los esperados y los reales.



La función de este proceso es hacer corresponder el sistema con la causa, llevando así a la corrección de error.

Pudiéndose obtener cualquiera de los siguientes resultados:

- 1.- Se encuentra la causa, se corrige y se elimina.
- 2.- No se encuentra la causa.

En este último caso, la persona que realiza la depuración debe sospechar la causa y diseñar pruebas que ayuden a validar sus sospechas, haciendo que el trabajo vuelva hacia atrás a la corrección del error en una forma iterativa.

El objetivo primordial en el proceso es encontrar y corregir la causa de un error en el software, este se consigue mediante una combinación de una evaluación sistemática, de intuición y de suerte.

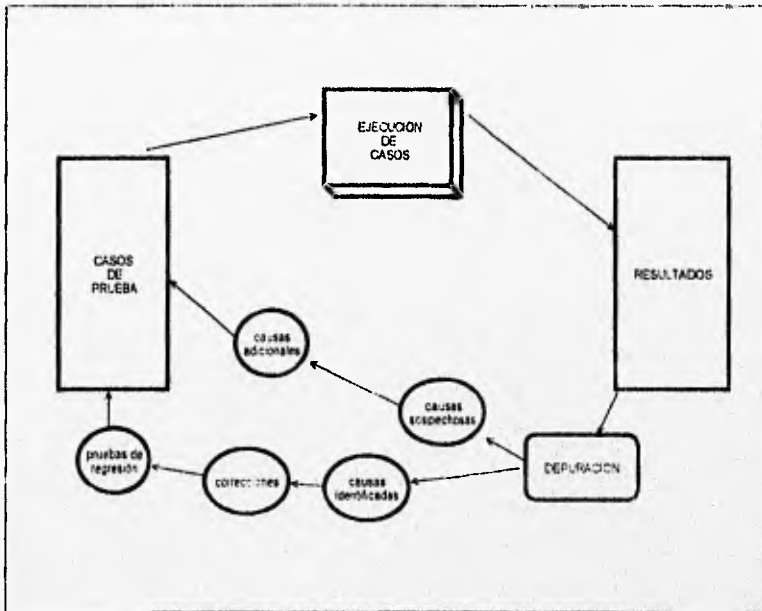
La vuelta atrás es el mejor enfoque para la depuración pudiéndose usar con éxito para pequeños programas, partiendo del lugar donde se descubre el posible error se recorre el código fuente hacia atrás (manualmente) hasta llegar a la posición del error, pero a medida que aumenta el número de líneas de código, el número de posibles caminos de vuelta se hace inmanejablemente grande.

Otro enfoque para la depuración - eliminación de causas - se manifiesta mediante inducción o deducción. los datos relacionados con la ocurrencia del error se organizan para aislar las posibles causas, alternativamente se desarrolla una lista de todas las posibles causas y se llevan a cabo pruebas para eliminar cada una. Si alguna prueba inicial indica que determinada hipótesis de causa en particular parece prometedora, se refinan los datos con el fin de intentar aislar el error.

Como ya se mencionó, en las pruebas realizadas al sistema surgieron algunos errores, la mayoría lógicos o triviales, los cuales fueron depurados encontrando las causas del error.



La mayor dificultad que se presentó fue el tiempo de ejecución del sistema, por lo que se estuvo trabajando en la reducción del tiempo de ejecución de cada uno de los módulos.



6.4 MANTENIMIENTO DE SOFTWARE.

El mantenimiento del software puede llevarse hasta el 70% del esfuerzo gastado por el desarrollo, ¿por qué es necesario el mantenimiento?. Cuando se crean los sistemas se utilizan las mejores técnicas de diseño y codificación existentes en su momento, dichos sistemas son creados teniendo como principales requisitos el tamaño del programa y el espacio de almacenamiento, luego, migraron a plataformas nuevas, fueron ajustados a cambios en las máquinas, los sistemas operativos y mejorados



para satisfacer las necesidades de nuevos usuarios. El resultado es la existencia de sistemas de software con estructuras, codificación, lógica y documentación pobre, concluyendo así que el cambio es inevitable en la construcción de sistemas basados en computadora.

El mantenimiento del software es mucho más que una "corrección de errores", este se describe definiendo las cuatro actividades que se llevan a cabo tras distribuir un programa.

1.- La primera actividad se debe a que no es razonable asumir que la prueba de software haya descubierto todos los errores latentes del sistema. Durante el uso del mismo, se encontrarán errores, siendo informado el equipo de desarrollo. El proceso que incluye el diagnóstico y la corrección de uno o más errores se denomina **mantenimiento correctivo**.

2.- La segunda actividad se produce por el avance inherente en cualquier aspecto de la informática, regularmente aparece nuevo hardware, sistemas operativos o nuevas versiones de los antiguos, se mejoran o modifican los equipos periféricos y otros elementos de los sistemas. Por otro lado la vida útil del software de aplicación puede superar los 10 años, sobreviviendo al entorno del sistema para el que fue originalmente desarrollado. Por tanto, el mantenimiento adaptativo -una actividad que modifica el software para que interaccione adecuadamente en su entorno cambiante- es muy necesario.

3.- La tercera actividad se produce cuando un paquete de software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades, sobre modificaciones de funciones ya existentes y sobre mejoras en general.

4.- La cuarta actividad se da cuando se cambia el software para mejorar una futura facilidad de mantenimiento o para proporcionar una buena base para futuras mejoras, a menudo se denomina **mantenimiento perfectivo**. Este normalmente se utiliza en el mantenimiento de hardware y otros sistemas físicos.



Después de las pruebas y depuración del sistema, se entregó al usuario con el fin de que se detectaran errores para realizar el mantenimiento correctivo, pero solo se realizaron algunas modificaciones ya que no se encontró ningún error.

Para el mantenimiento perfectivo no se ha recibido ninguna nueva recomendación o mejora al sistema por parte del usuario.

CARACTERÍSTICAS DEL MANTENIMIENTO DE SOFTWARE.

- ◉ La tarea del mantenimiento comienza con una evaluación de la documentación del diseño. Se determinan las importantes características estructurales, de rendimiento y de interfaz del software, se analizan las correcciones o modificaciones, se modifica el diseño y se revisa. Se desarrolla nuevo código fuente, se realizan las pruebas y se vuelve a lanzar el software; esta secuencia de sucesos constituye el mantenimiento estructurado.
- ◉ Los costes de mantenimiento en dinero es lo que obviamente más nos interesa, sin embargo, existen otros costes menos tangibles, un coste intangible del mantenimiento de software se encuentra en una oportunidad de desarrollo que se pospone o que se pierde, disminución de la calidad global del software debido a los errores latentes que introducen los cambios en el software mantenido.
- ◉ La mayoría de los problemas asociados con el mantenimiento de software se debe a las deficiencias de la forma en que el software ha sido definido y desarrollado.

Entre los muchos problemas clásicos asociados con el mantenimiento de software se encuentran los siguientes:

- ⊞ A menudo es difícil o imposible seguir la evolución de software a través de varias versiones. Los cambios no están adecuadamente documentados.



- ⊞ A menudo es demasiado complejo seguir el proceso por el que se construyó el software.
- ⊞ Es difícil comprender un programa "ajeno".
- ⊞ A medida que existen menos elementos de configuración de software, mayor es la dificultad.
- ⊞ No podemos esperar una explicación personal del software por el que lo ha desarrollado, una vez que se requiere el mantenimiento.
- ⊞ No existe una documentación apropiada o está mal preparada.
- ⊞ La revisión de mantenimiento más formal se da al terminar las pruebas y se denomina revisión de la configuración, esta asegura que todos los elementos de la configuración del software son completos, comprensibles y dispuestos para un control de modificación.

El sistema "SISPCNOM" ha sido documentado con la mayor exactitud posible tratando de evitar así los problemas asociados con el mantenimiento; los programas fuente se comentaron lo más posible para tener un buen entendimiento de cada uno.

TAREAS DE MANTENIMIENTO.

Las tareas asociadas con el mantenimiento del software comienzan mucho antes de que se haga una petición de mantenimiento. Inicialmente se debe establecer una organización de mantenimiento, se deben escribir procedimientos de evaluación y de información así como una secuencia estandarizada de sucesos para cada petición de mantenimiento.



La organización sugerida reduce la confusión y mejora el flujo de las actividades de mantenimiento. Cada uno de los anteriores puestos de trabajo sirve para establecer un área de responsabilidad en el mantenimiento. El controlador y la autoridad de control de cambios puede ser una misma persona. El supervisor del sistema puede tener otras obligaciones sin dejar de constituir un "contacto" con los paquetes de software.

EFFECTOS SECUNDARIOS DEL MANTENIMIENTO.

La modificación de software es peligrosa, desgraciadamente, cada vez que se introduce un cambio en un complejo procedimiento lógico, la probabilidad del error aumenta. La documentación del diseño ayuda a eliminar errores.

EFFECTOS SECUNDARIOS SOBRE EL CÓDIGO.

Nos comunicamos con la máquina mediante código fuente en un lenguaje de programación. Las posibilidades de efectos secundarios abundan aunque cada modificación del código puede conducir a un error, los siguientes cambios tienen mayor probabilidad de inducir error que otros.

- 1.- Un subprograma eliminado o cambiado.
- 2.- Eliminación o modificación de una sentencia de etiqueta.
- 3.- Eliminación o modificación de un identificador.
- 4.- Modificación de apertura o cierre de archivos.
- 5.- Modificación de operadores lógicos.



EFFECTOS SECUNDARIOS SOBRE LOS DATOS.

Durante el mantenimiento a menudo se hacen cambios sobre determinados elementos de una estructura de datos, cuando cambian los datos, el diseño de software puede no cuadrar con los datos y aparecer errores.

Los siguientes cambios en los datos a menudo, producen efectos secundarios:

- ✦ Redefinición de constantes locales o globales.
- ✦ Redefinición de formatos de registro o archivos.
- ✦ Modificación de datos globales.
- ✦ Reorganización de argumentos de E/S o de subprogramas.

EFFECTOS SECUNDARIOS SOBRE LA DOCUMENTACIÓN.

El mantenimiento se debe centrar en la configuración completa del software y no solo en las modificaciones del código fuente. Estos aspectos se dan cuando no se reflejan los cambios del código fuente en la documentación del diseño y en los manuales orientados al usuario.

Siempre que se haga un cambio sobre el flujo de datos, la arquitectura del diseño y en los procedimientos, se debe actualizar la documentación de soporte. Si no se reflejan los cambios del software ejecutable en el manual del usuario, los efectos secundarios están garantizados.



MANTENIMIENTO PREVENTIVO.

Un programa con módulos de 2000 líneas, con tres comentarios significativos y sin ninguna otra documentación, debe ser modificado para acomodar los cambios en los requisitos de los usuarios. Se tienen las siguientes opciones:

1.- Intentar comprender a grandes rasgos el funcionamiento interno del programa, en un esfuerzo de llevar a cabo las modificaciones en forma más efectiva.

2.- Rediseñar, recodificar y probar las partes del software que requieran modificaciones.

En lugar de esperar a recibir una petición de mantenimiento, la organización de desarrollo de mantenimiento selecciona un programa que:

1.- Vaya a estar en un uso durante una determinada serie de años.

2.- Este siendo usado correctamente.

3.- Pueda necesitar en un futuro cercano modificaciones o mejoras importantes.

El mantenimiento preventivo será, la aplicación de las metodologías actuales a sistemas de ayer para facilitar requisitos del mañana.

Cuando una organización de desarrollo de software vende productos de software, el mantenimiento preventivo se ve como "nuevas versiones" del programa. Se mejora la futura facilidad de mantenimiento y la fiabilidad como base para las futuras mejoras.



6.5 INTEGRIDAD DE LOS DATOS.

La integridad de los datos es un factor muy importante en el desarrollo de sistemas, ya que el usuario debe respaldar su información continuamente cuando el volumen de esta se va incrementando constantemente. Al no contar con la integridad de los datos en el sistema, es casi segura la pérdida de nuestra información, sabemos que pueden ocurrir errores ocasionados por los usuarios (eliminar una base de datos, eliminar registros, etc.).

Es por esto que el sistema "SISPCNOM" cuenta con una herramienta de respaldo de información.

RESULTADOS.

Se desarrolló un sistema que realiza la consulta automatizada de la norma NOM-001-SEMP-1994. Dicho sistema realiza las siguientes tareas:

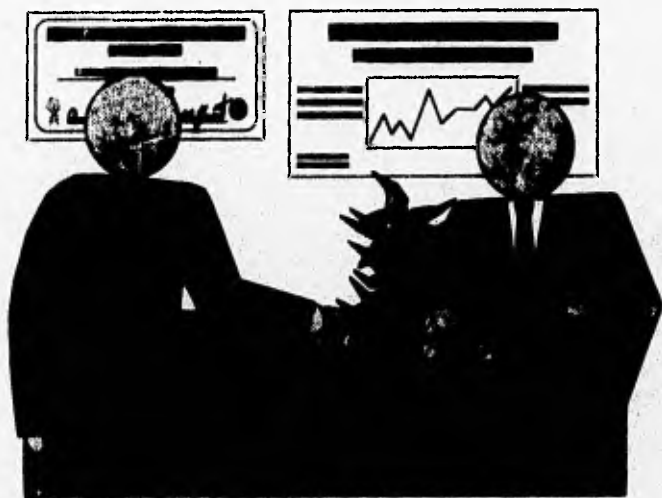
- ◆ Consulta por palabra.
- ◆ Consulta por artículo.
- ◆ Consulta por índice temático.
- ◆ Elaboración de reportes en impresora y en disco.
- ◆ Despliegue de glosario de términos.
- ◆ Herramientas de ayuda para manejar el sistema.
- ◆ Respaldo de la información.

Todas estas operaciones cumplieron con el objetivo de optimizar el tiempo de consulta de la norma y concentrar el mayor tiempo posible al trabajo ingenieril de instalaciones eléctricas.

Los resultados obtenidos cumplieron con las expectativas del usuario final y en algunos casos se mejoraron con algunas herramientas extras que contiene el sistema.



CONCLUSIONES



CONCLUSIONES.

Al término del presente trabajo podemos concluir lo siguiente:

En la actualidad, la aplicación adecuada de sistemas de cómputo constituye una herramienta valiosa en el manejo de información, ya que dichos sistemas son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones.

Por esta razón el ingeniero está obligado a entender al usuario final y a las computadoras, y lograr que la información sea realmente útil en la toma de decisiones.

Como consecuencia del avance tecnológico, los sistemas de cómputo se han perfeccionado para que su interacción con el usuario resulte más productiva y agradable, en otras palabras, más amigable.

Las dos principales ventajas obtenidas por el sistema SISPCNOM al automatizar la información correspondiente a la norma *NOM-001-SEMP-1994* para Instalaciones Eléctricas en el Departamento de Ingeniería Eléctrica de la Facultad de Ingeniería, fueron:

- ① Agilizar el trabajo en el diseño de instalaciones eléctricas, al proporcionar varios tipos de consulta, y elaboración de reportes del contenido de dicha norma.
- ② Proporcionar una herramienta de asesoría en todas las materias de Ingeniería Eléctrica y otros cursos que se impartirán en la D.I.M.E. debido a que se utilizará como material didáctico de apoyo a los profesores y alumnos; por lo que se pondrá inmediatamente en funcionamiento en la Facultad de Ingeniería.

Finalmente fue indispensable examinar si el sistema cumplió cabalmente con las expectativas planteadas al principio. Por lo que después de una evaluación exhaustiva se concluyó lo siguiente:

SISPCNOM



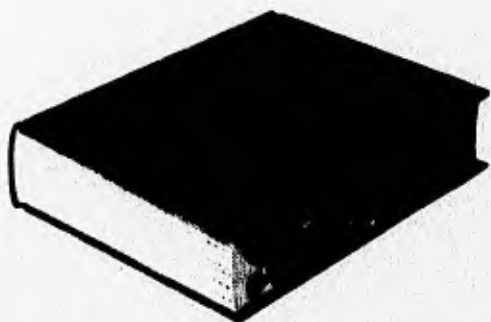
- ✓ El sistema SISPCNOM resulta de gran utilidad ya que satisface plenamente todas las necesidades del personal que lo maneja.
- ✓ Puede ser fácilmente actualizado en caso de que surjan nuevos elementos en el diseño de instalaciones eléctricas.
- ✓ Se le puede dar mantenimiento sin tanta dificultad, ya que está desarrollado en módulos muy comprensibles.
- ✓ Resulta de bajo costo, debido a que sólo se requiere una computadora PC compatible e impresora.
- ✓ Presenta una gran seguridad ya que cuenta con un módulo de respaldo de información (BACKUP).

Es importante mencionar que el presente trabajo fue posible gracias a un esfuerzo conjunto entre los departamentos de Ingeniería Eléctrica e Ingeniería en Computación, sentando un precedente en las actividades coordinadas entre estos dos departamentos.

Por todos los aspectos antes citados se concluye que los resultados son satisfactorios ya que se cumplieron todos los objetivos propuestos por este trabajo.



G L O S A R I O



G L O S A R I O**DEFINICIONES :**

ADMINISTRADOR DE LA BASE DE DATOS : (database administrator, DBA) uno o más individuos responsables del control y manejo del sistema de la base de datos.

ALGORITMO : Programa lógico con instrucciones escritas en algún lenguaje de programación para ejecutar alguna tarea.

ANÁLISIS : Método de investigación que comienza por separar una situación o problema en las partes que los componen y luego trata de entender cómo se afectan las partes entre sí.

ANÁLISIS DE SISTEMAS: Es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

ARCHIVO : Documento o aplicación a los cuales se ha asignado un nombre.

ARCHIVO FUENTE (PRG) : Es el archivo (.PRG) que usted escribe en CLIPPER y pretende compilar.

ARREGLO : Una de las estructuras de datos fundamentales en programación; tabla sencilla o multidimensional que el programa trata como un solo dato. Cualquier elemento del arreglo puede llamarse mediante el nombre del arreglo y la localidad del elemento dentro del arreglo.

AYUDA EN LÍNEA : Utilería de ayuda disponible en pantalla mientras usa un programa.

BACKUP : Ver respaldo.

SISPCNOM



BASES DE DATOS : Conjunto de archivos interrelacionados, creados y manejados por un DBMS (Sistema de Administración de Bases de Datos).

BUFFER O DEPÓSITO DE TRANSFERENCIA : Espacio temporal de memoria reservado por el sistema operativo para el almacenamiento físico de registros. Su contenido puede transferirse entre el almacén externo y la memoria del programa.

CAMPO : Unidad direccionable más pequeña que puede ser referida por un programa.

CLIPPER : Compilador desarrollado por Nantucket Systems, Inc. para el lenguaje de comandos de dBase. Muchos creadores de aplicaciones lo consideran superior que el que ofrecen los editores dBase.

COMPATIBILIDAD : Capacidad de un periférico, un programa o un adaptador para funcionar con o sustituir a un modelo determinado de computadora. Asimismo, capacidad de una computadora para correr el software de otra marca de computadora.

COMPILADOR : Programa que lee instrucciones escritas en un lenguaje de programación legible para los humanos. Como Pascal o Clipper, y que traduce las instrucciones en un programa ejecutable, legible para la máquina. Los programas compilados corren mucho más rápido que los interpretados, ya que todo el programa se traduce a lenguaje de máquina y no necesita compartir espacio de memoria con la interpretadora. Ver interpretadora y lenguaje máquina.

CONSULTA : En administración de bases de datos, interrogante de búsqueda que le indica al programa que tipo de información debe recuperar de la base de datos. El aspecto sobresaliente de un sistema administrativo de base de datos efectivo consiste en no presentar toda la información que contiene el sistema, sino en mostrar únicamente la que necesita usted para un propósito específico.

Una consulta determina el criterio que se sigue para extraer información de la base de datos. La consulta guía a la computadora en la recuperación de la información requerida y la eliminación de la que no se necesita.



COPIAR : Colocar una copia del texto para poder transferirla a otro lugar.

COPIA DE RESPALDO : Cinta magnética que contiene copia de archivos de otras fuentes de almacenamiento. Rutinariamente se copian archivos de discos a cintas para prevenir la pérdida de información por si ocurrieran accidentes o fallas.

CPU (Central Processing Unit) : Unidad de Proceso Central; parte de una computadora que se encarga de controlar todas las actividades que se llevan a cabo. También se le llama microprocesador.

CURSOR DE SELECCIÓN : Dispositivo para marcar que indica la posición actual en una ventana o menú. El cursor de selección puede tener el aspecto de una zona resaltada.

DATOS : Información real que puede usar usted para generar cálculos o tomar decisiones.

DBASE : Sistema de manejo de bases de datos para microcomputadora.

DIAGRAMA DE FLUJO : Diagrama que contiene símbolos referentes a operaciones de cómputo que describe cómo se realiza el programa.

DIRECTORIO : Parte de la estructura de organización de archivos en el disco. Un directorio puede contener archivos y otros directorios (denominados subdirectorios). La estructura de directorios y subdirectorios de un disco se denomina árbol de directorios.

DISCO : Medio utilizado para el almacenamiento de información. Cuando se almacena información en un disco esta se conserva incluso después de apagar la computadora.

DISCO DURO : Disco que se encuentra montado de forma permanente en el interior de su unidad. también denominado Disco Fijo.

DISEÑO DE SISTEMAS : Es el proceso de planificar, reemplazar o complementar un sistema organizacional existente.



DISQUETE : Disco que puede introducirse y retirarse de la unidad de disco de la computadora. También denominado Disco Flexible.

DOCUMENTACIÓN : Instrucciones, tutorías y fuente de consulta que le proporcionan la información requerida para usar con eficacia un programa de computadora o un sistema de cómputo.

DOCUMENTO : Archivo que contiene trabajo creado por usted, como un informe de negocios, un memorándum o una hoja de trabajo.

DOS : Referente a MS-DOS (Microsoft Disk Operating System) Sistema Operativo en disco de Microsoft , el sistema operativo más utilizado para las computadoras personales. Se puede decir que un sistema operativo es un conjunto de programas que se encargan de administrar los recursos de hardware y software de un sistema de cómputo.

FMS : (File Management System) sistema administrador de archivos.

GLOSARIO : Catálogo o vocabulario de palabras.

HARDWARE : Conjunto de dispositivos de los que consiste un sistema. Comprende componentes tales como el teclado, mouse, las unidades de disco y el monitor.

IMPRESORA : Una impresora imprime la información procesada por la computadora. Las impresoras varían en la velocidad de impresión y en las capacidades de información de salida.

INFORMACIÓN : Datos organizados y ordenados extraídos de un conjunto sin evaluación previa.

INTEGRIDAD DE LOS DATOS : Mantenimiento de los datos correctos en la base de datos todo el tiempo.

INTERFASE : Aditamento que puede ser tanto software como hardware y que permite establecer comunicación entre dos entidades diferentes.



INTERFAZ : Circuito electrónico que gobierna la conexión entre dos dispositivos de hardware y los ayuda a intercambiar información de manera confiable. Es sinónimo de puerto.

INTERPRETADORA: Traductor de un lenguaje de programación de alto nivel que no crea una versión ejecutable de un programa: en vez de ello, una interpretadora traduce y ejecuta el programa al mismo tiempo.

LENGUAJE DE MÁQUINA : Lenguaje que reconoce y ejecuta la unidad central de procesamiento (CPU) de una computadora. Esta simbolizado por ceros (0) y unos (1) y su uso y lectura son sumamente difíciles.

MEMORIA : Área de almacenamiento para información y aplicaciones.

MEMORIA EXPANDIDA : Memoria que utilizan algunas aplicaciones además de la memoria convencional. La memoria expandida es un estándar antiguo que se está sustituyendo hoy por la memoria extendida.

MEMORIA EXTENDIDA : Memoria más allá de un megabyte (MB) en las computadoras basadas en los procesadores 80286, 80386 y 80486.

MENÚ : Lista de opciones disponibles en la ventana del sistema. Los nombres de los menús aparecen en la barra de menús situada en la parte superior de la ventana principal.

MICROPROCESADOR : Circuito integrado que contiene la unidad lógico-aritmética (ALU) y la unidad de control de la unidad central de procesamiento (CPU) de la computadora.

MÓDULO : Es un programa de computación, unidad o sección capaz de funcionar por sí misma. Por ejemplo, en un programa integrado, usted puede usar el módulo de procesamiento de texto como si fuera un programa independiente y único.

MONITOR : El monitor tiene una pantalla que presenta información. es decir, las instrucciones que el usuario envía a la computadora y el resultado e interpretación de esas instrucciones. La Pantalla podrá presentar información en un solo color o en varios.



NOM : Norma Oficial Mexicana.

NORMALIZACIÓN : Técnica en el diseño de las bases de datos que empieza agrupando todos los atributos en una relación universal, la cual después es descompuesta en relaciones más pequeñas hasta que todas las relaciones divididas pertenezcan a la cuarta forma normal.

OPCIÓN : Alternativa que se puede elegir en un menú.

PANTALLA : Ver monitor.

PROCESO : Ejecución de instrucciones de un programa, realizada por la unidad central de procesos (CPU), que transforma de cierta manera la información, esto es, la ordena y selecciona parte de ésta según un criterio especificado o practica cálculos matemáticos sobre ella.

PROGRAMA : Lista de instrucciones escritas en lenguaje de programación que ejecuta una computadora para que la máquina actúe de una forma determinada. El término es sinónimo de software.

RAM : (Random Acces Memory) Memoria de Acceso Aleatorio. La memoria RAM es también conocida como Memoria Principal debido a que el usuario está en contacto directo con ella. Esta memoria carga (extrae del disco duro) la información con la que el usuario vaya a trabajar o esté trabajando.

REGISTRO (Record) : Entidad representada por cierto número fijo de campos.

REGISTRO FÍSICO : Registro físico que consta de registros lógicos almacenados juntos como una unidad de transferencia de datos entre la memoria principal y el dispositivo de almacenamiento.

REGISTRO LÓGICO : Percepción del programador de un registro definido en un programa.



RESPALDO : Proceso de copiado de un programa o de un archivo de datos con fines de seguridad. Asimismo, acto de copiar un archivo de datos o de programa a un dispositivo de almacenamiento secundario para conservar la información en lugar seguro.

RESPALDO DE INFORMACIÓN : Guarda la información contenida en un archivo a un disco flexible o bien a un disco duro.

ROM : (Random Only Memory) Memoria de solo lectura.

SELECCIONAR : Marcar un elemento para que se aplique sobre él cualquier acción que se realice subsecuentemente. Por lo general , se selecciona una ventana utilizando el Mouse o presionando una tecla. Se elige luego la acción que se debe aplicar. Vea también Elegir.

SISPCNOM : Sistema Para Consulta de la Norma Oficial Mexicana.

SISTEMA : Conjunto de procedimientos interrelacionados entre sí que son enfocados a realizar un objetivo específico.

SISTEMA DE ADMINISTRACIÓN DE BASES DE DATOS (DBMS) : En computación con macrocomputadoras, sistema de cómputo organizado para la administración sistemática de una enorme cantidad de información. En computación personal (microcomputadoras), programa como dBase con capacidad análoga de organización, almacenamiento y recuperación de la información.

SISTEMA OPERATIVO : Programa de control maestro para computadora que maneja las funciones internas de la máquina y le proporciona los medios para controlar la operatividad de la misma.

SOFTWARE : Conjunto de instrucciones mediante las cuales la computadora puede realizar tareas. Los programas, los sistemas operativos y las aplicaciones son ejemplos de software.

SQL : (Structure Query Language) lenguaje de consulta estructurado: lenguaje de manejo relacional de bases de datos.



TECLAS DE DIRECCIÓN : Teclas marcadas con una flecha en el teclado de la computadora. Cada flecha recibe el nombre correspondiente a la dirección hacia la cual apunta la flecha: **FLECHA ARRIBA**, **FLECHA ABAJO**, **FLECHA IZQUIERDA** y **FLECHA DERECHA**.

TECLA DE MÉTODO ABREVIADO : Tecla o combinación de teclas que pueden presionarse para ejecutar un determinado comando o acción.

TIEMPO DE ESPERA : Cantidad de tiempo que deberá esperar la computadora cuando un dispositivo no ejecute una determinada tarea antes de presentar un mensaje de error.

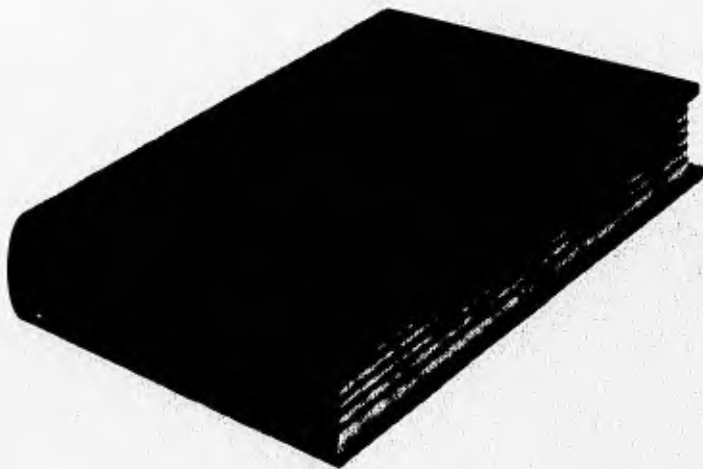
UNIDAD DE DISCO : Dispositivo utilizado para almacenar y recuperar información guardada en discos.

VELOCIDAD DE TRANSMISIÓN : Velocidad a la cual puede transferirse la información a través de un puerto.











VENTANA : Área rectangular de la pantalla en la cual se presenta una aplicación.



BIBLIOGRAFÍA



BIBLIOGRAFÍA.

-  Análisis y Diseño de Sistemas
Kenneth E. Kendall, Julie E. Kendall
Prentice Hall Hispanoamericana
-  Análisis y Diseño de Sistemas de Información
James A. Senn
Mc Graw-Hill 2a. edición
-  Clipper 5.0
Francisco Marín, Antonio Quirós, Antonio Torres.
Macrobot 1991
-  Clipper 5.0
José Javier García-Badell
Mc Graw-Hill
-  Clipper 5.01 Avanzado
José A. Ramalho
Mc Graw-Hill
-  111 Funciones en Clipper
José A. Ramalho
Mc Graw-Hill
-  Diagramación y Programación Estructurada
Letvin Lozano
Mc Graw-Hill
-  Diccionario para Usuarios de Computadoras
Bryan Pfaffenberger
Prentice Hall Hispanoamericana 3a. edición
-  Diseño de Sistemas
Enid Squire
Fondo Educativo Interamericano
-  Diseño Modular
Alan Cohen
Mc Graw-Hill



- ☐ Ingeniería de Software
Ian Sommerville
Adison-Wesley Iberoamericana
- ☐ Ingeniería de Software (Metodología de Estructura de Datos)
Michael G. Schneider
Anaya Multimedia
- ☐ Ingeniería de Software
Richard Fairley
Mc Graw- Hill
- ☐ Ingeniería de Software (Un Enfoque Práctico)
Roger S. Pressman
Mc Graw-Hill
- ☐ Norma NOM-001-SEMP-1994 Relativa a las Instalaciones
destinadas al Suministro y Uso de la Energía Eléctrica
Diario Oficial de la Federación
Secretaría de Energía, Minas e Industria Paraestatal
Octubre de 1994
- ☐ Planeación y Control de la Producción
Robert H. Bock, William K. Holstein
Limusa
- ☐ Sistemas de Bases de Datos (Administración y Uso)
Alice Y. H. Tsai
Prentice Hall



A N E X O 1

M A N U A L
D E
U S U A R I O



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE INGENIERÍA

MANUAL DEL USUARIO

**SISTEMA PARA CONSULTA DE LA NORMA OFICIAL
MEXICANA
NOM-001-SEMP-1994
PARA INSTALACIONES ELÉCTRICAS.**

(SISPCNOM)

**SISTEMA PARA CONSULTA DE LA
NORMA OFICIAL MEXICANA
NOM-001-SEMP-1994**



S I S P C N O M

BIENVENIDO

Bienvenido a **SISPCNOM**, el Sistema Para Consulta de la Norma Oficial Mexicana **NOM-001-SEMP-1994**.

Este manual está organizado de manera que constituya una guía de referencia útil y comprensible. Está dividido por menús e incluye información para usuarios que tengan poca experiencia en el uso de PC's.

No será necesario leer el manual completo sino que, consultando la tabla de índice, se podrán elegir los temas que se necesiten.

ÍNDICE

I.- REFERENCIA BÁSICA

| | | |
|-----|-----------|---|
| 1. | OBJETIVO | 1 |
| 1.1 | OPERACIÓN | 1 |

II.- INSTRUCTIVO DE OPERACIÓN

| | | |
|-----|-------------------------|---|
| 2. | OBJETIVO | 2 |
| 2.1 | DESCRIPCIÓN GENERAL | 2 |
| 2.2 | INSTALACIÓN DEL SISTEMA | 2 |
| 2.3 | ACCESO AL SISTEMA | 3 |

III.- SELECCIÓN

3. MENÚ DE CONSULTAS

| | | |
|--|-----------------------|----|
| | CONSULTA POR ARTÍCULO | 5 |
| | CONSULTA POR PALABRA | 7 |
| | CONSULTA POR ÍNDICE | 10 |

3.1 MENÚ DE REPORTES

| | | |
|--|-----------------------|----|
| | REPORTES EN IMPRESORA | 12 |
| | REPORTES EN ARCHIVO | 14 |

3.2 MENÚ DE HERRAMIENTAS

| | | |
|--|-------------------|----|
| | GLOSARIO | 16 |
| | RESPALDO DE DISCO | 18 |

3.3 SALIR

| | | |
|--|---------------|----|
| | VOLVER AL DOS | 19 |
|--|---------------|----|

3.4 TECLAS AUXILIARES

| | | |
|-------|---|----|
| F1 | AYUDA | 20 |
| | OPCIONES DEL MENÚ DE CONSULTAS | 21 |
| | OPCIONES DEL MENÚ DE REPORTE | 21 |
| | OPCIONES DEL MENÚ DE HERRAMIENTAS | 21 |
| | TECLAS DE AYUDA | 22 |
| | TECLAS PARA EL MOVIMIENTO DEL CURSOR | 22 |
| | TECLAS DEL MENÚ | 22 |
| TECLA | ESC | 22 |
| TECLA | F5 | 23 |
| TECLA | F10 | 24 |

I. REFERENCIA BÁSICA

1. OBJETIVO :

Contribuir a automatizar los procedimientos de consulta de la norma oficial mexicana NOM-001-SEMP-1994 para Instalaciones Eléctricas, para su uso en el Departamento de Ingeniería Eléctrica de la Facultad de Ingeniería de la UNAM y otras instituciones.

1.1 OPERACIÓN :

Este sistema podrá instalarse en cualquier equipo de cómputo compatible con PC.

Los requerimientos mínimos en cuanto a hardware son:

- Una computadora PC compatible con 10 MB disponibles en disco duro.
- Una unidad de disco flexible de 5 1/4" o 3 1/2".
- Memoria principal de 2 MB.
- Sistema Operativo MS-DOS versión 5.0 o superior.
- Monitor a color o blanco y negro.
- Impresora de matriz de puntos.

II. INSTRUCTIVO DE OPERACIÓN

2. OBJETIVO :

Se recomienda leer este manual antes de operar el sistema, así como consultarlo en el momento en que surja alguna duda al estarlo utilizando.

La finalidad del presente manual, es orientar al usuario sobre la operación del SISPCNOM en la consulta de la norma oficial mexicana NOM-001-SEMP-1994 para Instalaciones Eléctricas, así como los pasos que se deben seguir para obtener la máxima eficiencia en el manejo del sistema.

2.1 DESCRIPCIÓN GENERAL :

El sistema esta compuesto de una serie de programas que trabajan con base en menús, los cuales son desplegados según se vaya seleccionando la información que se desea consultar.

2.2 INSTALACIÓN DEL SISTEMA :

El disco que acompaña este trabajo debe de instalarse en el disco duro de la PC de la siguiente manera :

Primero debe introducirse el disco flexible en el drive de la máquina ya sea la unidad A o B y teclear instalar y dar enter.

```
A:\>instalar <enter>
```

El acceso al sistema se verá en el siguiente punto.

2.3 ACCESO AL SISTEMA :

Una vez situado en la unidad de disco duro (C:\>) colocarse en el directorio de SISPCNOM de la siguiente manera:

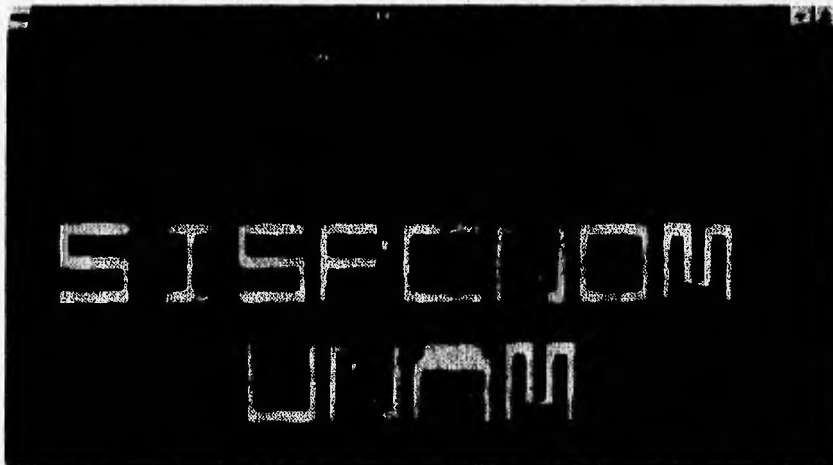
```
C:\> CD SISPCNOM      <enter>
```

Una vez que ya se esta en el directorio SISPCNOM se deberá volver a teclear SISPCNOM para poder entrar al archivo ejecutable y poder ingresar al sistema.

```
C:\SISPCNOM>SISPCNOM  <enter>
```

Después de estos pasos aparecerá la pantalla de presentación del sistema (ver pantalla No.1). posteriormente aparecerá la pantalla principal del sistema SISPCNOM (ver pantalla No. 2).

PANTALLA No. 1.



PANTALLA No. 2.



MENÚ DE CONSULTA :

- Consulta por artículo
- Consulta índice temático
- Consulta por palabra

MENÚ DE REPORTE

- Imprimir artículo
- Crear un archivo que contenga un artículo

MENÚ DE HERRAMIENTAS

- Hacer respaldo de la información
- Glosario de términos

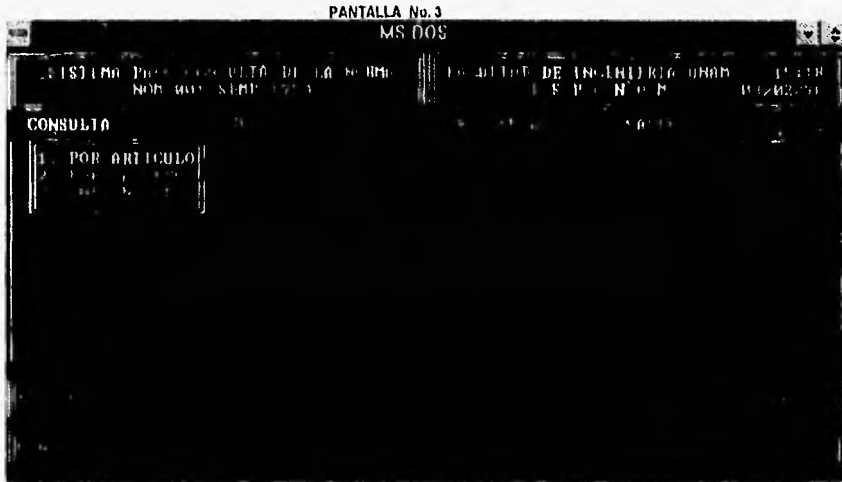
MENÚ DE SALIDA

- Retornar al sistema operativo

III SELECCIÓN

3. MENÚ DE CONSULTAS.

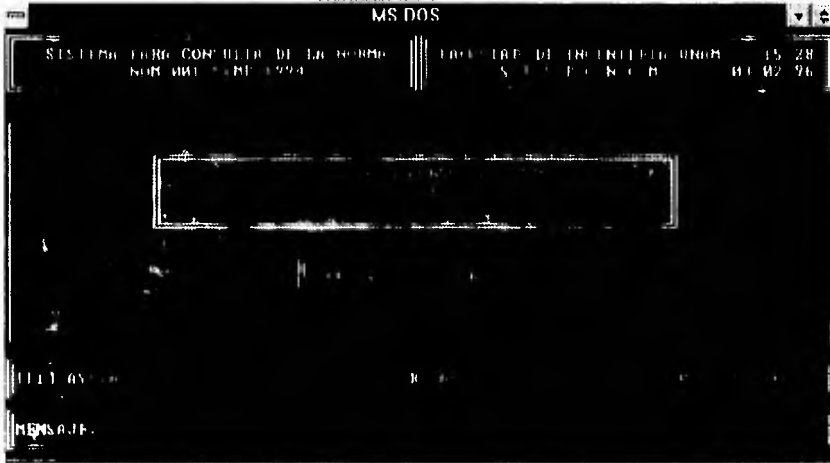
CONSULTA POR ARTÍCULO:



Si su selección fue consulta por artículo aparecerá otra pantalla con la siguiente frase: **ARTÍCULO A CONSULTAR**___. Aquí se pondrá el número de artículo que se quiera consultar y posteriormente presionar enter.

A continuación aparecerá una pantalla en la que se preguntará : **CORRECTO (S/N) ?**___. Si ha tecleado el número correcto de artículo a consultar , presione "S", con lo cual el sistema buscará el artículo y en pocos segundos aparecerá toda la información del mismo. En caso de equivocación al teclear el número de artículo a consultar, deberá presionar la letra "N" y volverá a desplegarse la primer pantalla para que usted pueda volver a teclear el número de artículo de manera correcta (ver pantalla 4 y 5).

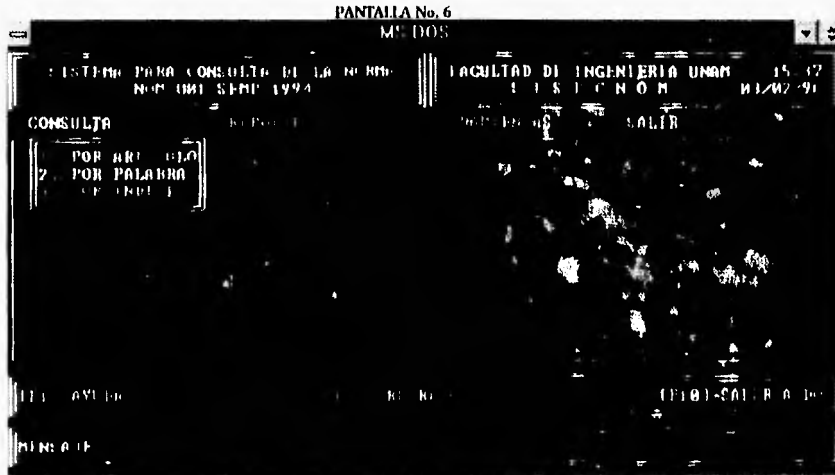
PANTALLA No. 4
MS DOS



PANTALLA No. 5
MS-DOS



CONSULTA POR PALABRA :



Si su selección fue **consulta por palabra**, aparecerá otra pantalla con la siguiente frase: **PALABRA A BUSCAR**__. Aquí deberá escribirse la palabra a consultar y posteriormente presionar enter. A continuación aparecerá una pantalla en la que se preguntará : **CORRECTO (S/N) ?**__. En caso de que exista algún error presione "N" y volverá a solicitarle la escritura de la palabra a buscar (ver pantalla No. 7).

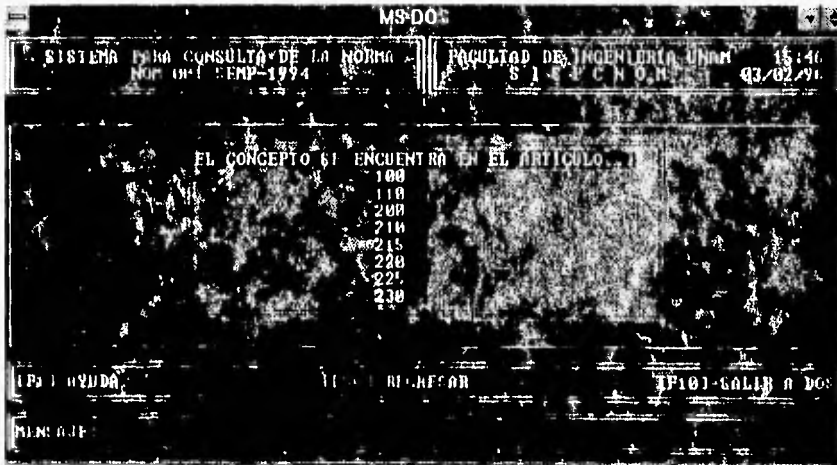


Si se ha teclado correctamente la palabra, presionamos "S", con esto aparecerá en la pantalla un indicador de porcentaje de búsqueda similar a un termómetro y al concluir la búsqueda desplegará una lista con todos los artículos que contienen dicha palabra, con objeto de que se pueda revisar cada uno de ellos. Al entrar a cada artículo, nos posicionará en el renglón en donde se encuentre la palabra solicitada. Si presionamos ESC, aparecerá en la ventana de mensajes la siguiente frase: BUSCAR PALABRA: Siguiendo Anterior Finalizar (S/A/F)____. Si se desea localizar el siguiente renglón donde se localiza la palabra presionamos "S", en caso de volver a visualizar el anterior presionamos "A" y si ya no se desea seguir buscando en ese artículo presionamos "F". Al presionar esta última opción, podemos seguir consultando en los archivos restantes (ver pantalla 8.9,10 y 11).

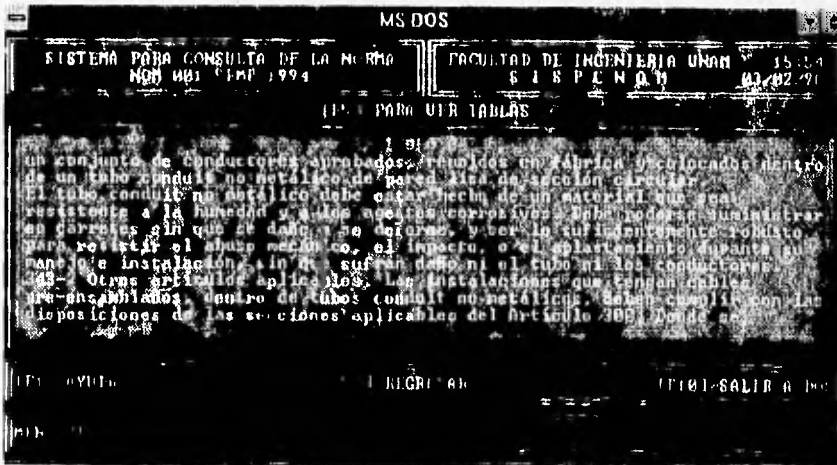
PANTALLA No. 8

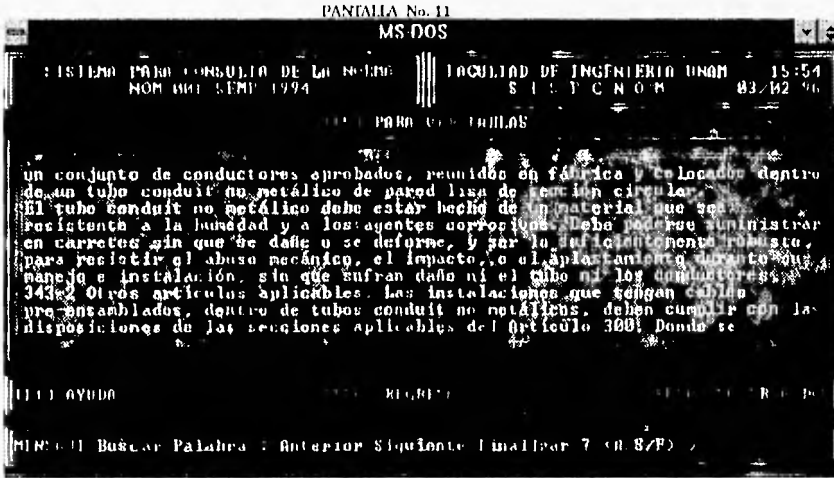


PANTALLA No. 9

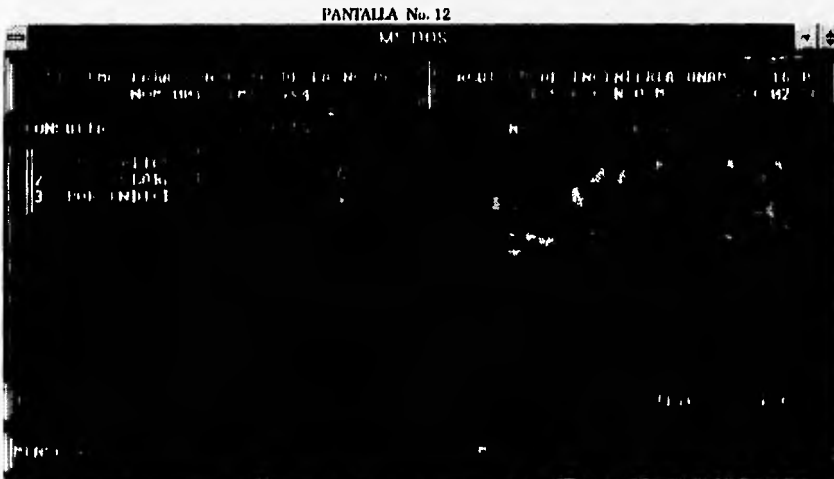


PANTALLA No. 10

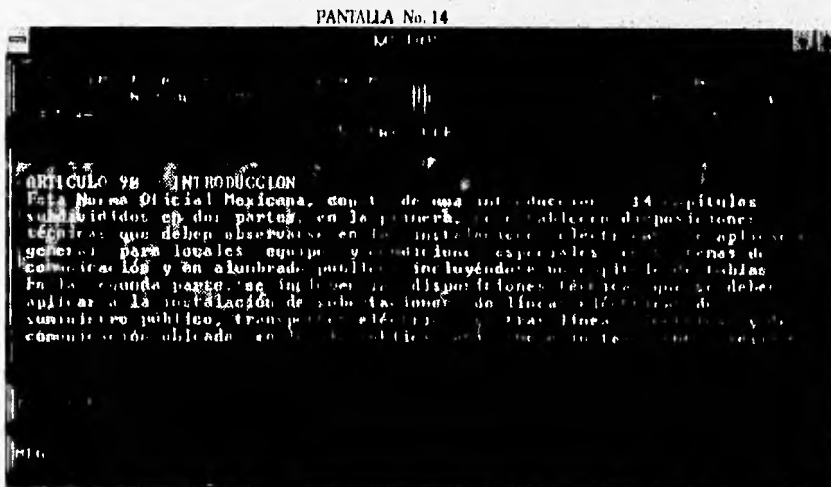
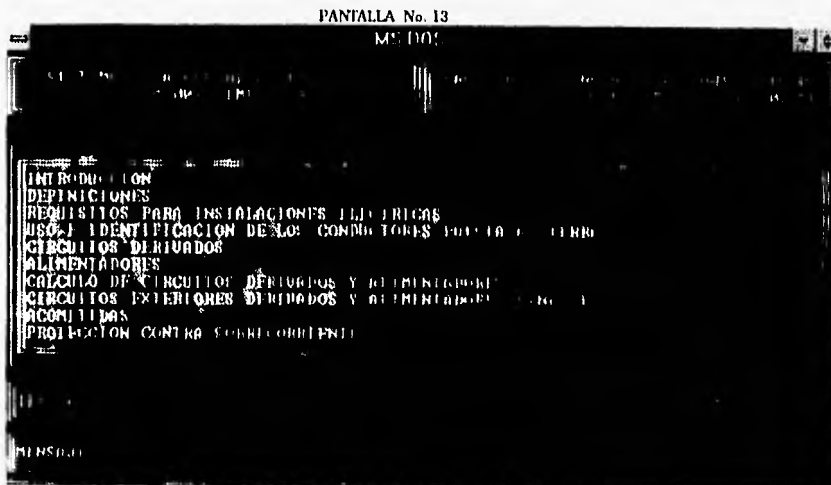




CONSULTA POR ÍNDICE TEMÁTICO:

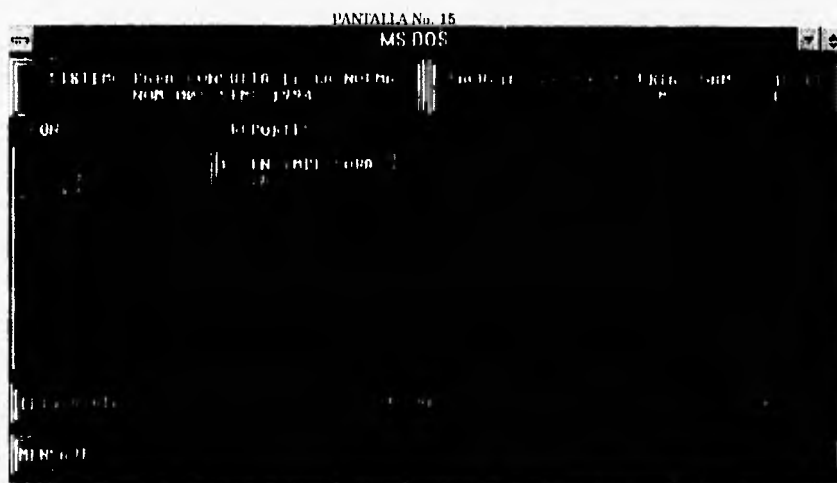


Si su selección fue consulta por índice temático aparecerá otra pantalla que contiene el índice temático de toda la norma. A continuación podrá seleccionar algún tema específico y en pocos segundos aparecerá el artículo del tema seleccionado. Para salir del menú basta con presionar la tecla ESC (ver pantalla 13 y 14).

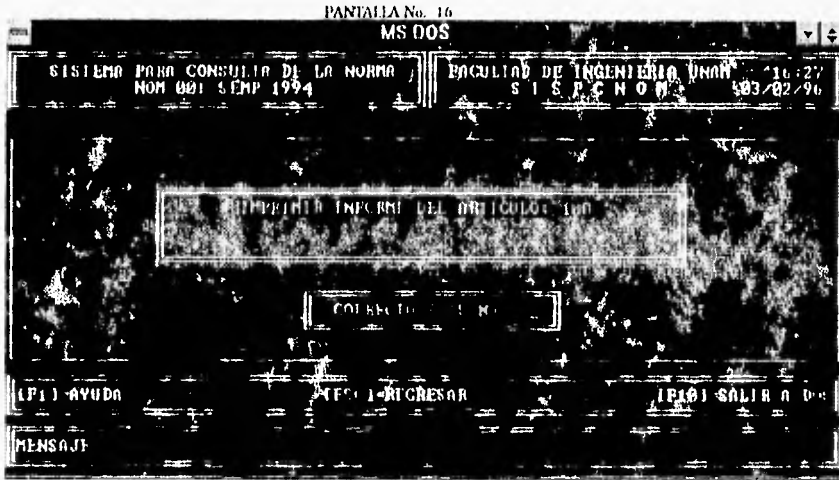


3.1 MENÚ DE REPORTES:

REPORTE EN IMPRESORA



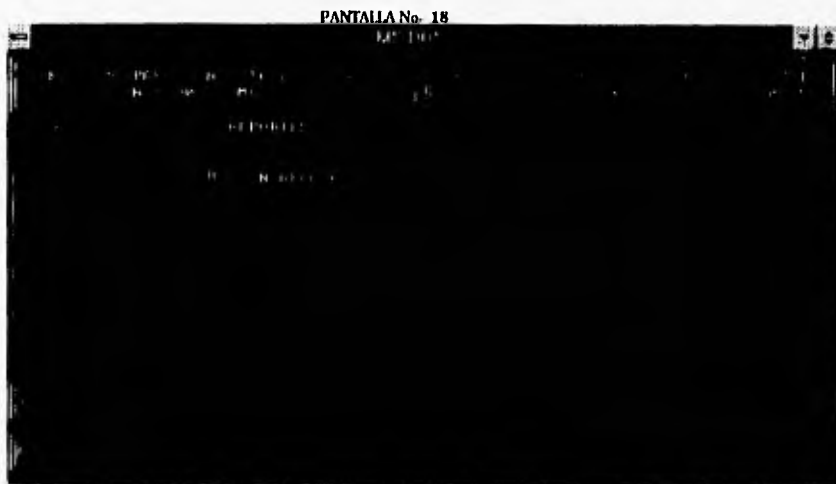
Si usted se posiciona en el menú de reportes aparecerá una ventana con la opción "**imprimir**", si selecciona esta opción, aparecerá una pantalla en la que se preguntará: IMPRIMIR INFORME DEL ARTÍCULO: _____. Aquí deberá teclear el número del artículo que desee imprimir. Posteriormente aparecerá una pantalla que preguntará : CORRECTO (S/N)___ en donde con una "S" se afirmará la pregunta y con una "N" la negará y tendrá que poner el número correcto del artículo que desee mandar imprimir. (ver pantalla 16 y 17).



PANTALLA No. 17



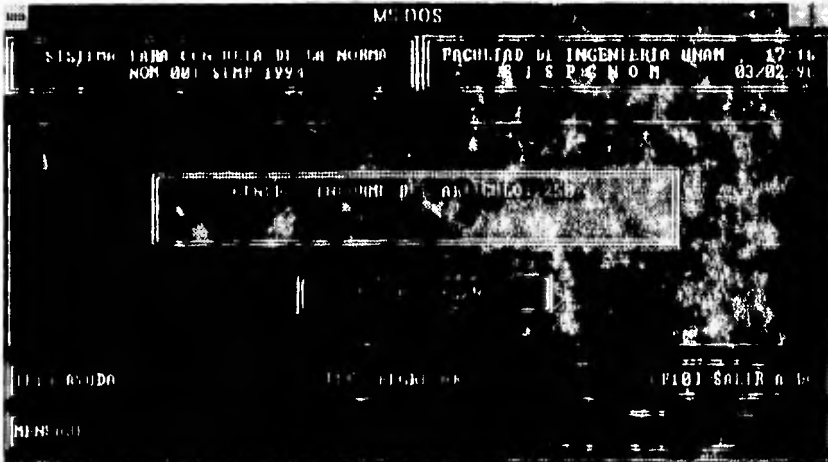
REPORTES EN ARCHIVO



Si usted se posiciona en el menú de reportes aparecerá un ventana con la opción de crear el informe en un archivo, si selecciona esta opción, aparecerá una pantalla en la que se preguntará: **GENERAR INFORME DEL ARTÍCULO:** ____ . Aquí se debe teclear el número del artículo que se desee imprimir. Posteriormente aparecerá una pantalla que preguntará : **CORRECTO (S/N)**__ . En la siguiente pantalla aparecerán las siguientes frases: **NOMBRE DEL ARCHIVO :** ____ . Aquí se pondrá el nombre que se le asignará al archivo (no rebasar de 8 caracteres), también se visualiza otra sentencia; **COPIAR EL ARCHIVO EN LA UNIDAD:** __ . Aquí se deberá poner la unidad en donde será grabado el archivo, esta puede ser A , B o C.

En caso de que se dé un nombre de archivo inválido o una unidad no válida aparecerá un mensaje de error, que desaparecerá en unos segundos. (ver pantalla 19 y 20)

PANTALLA No. 19
MÉTODOS



PANTALLA No. 20



3.2 MENÚ DE HERRAMIENTAS :

GLOSARIO :

PANTALLA No. 21

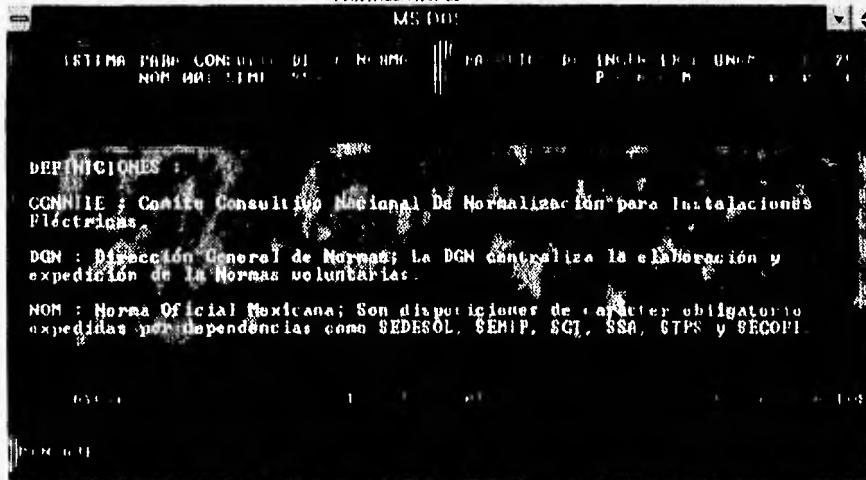


Si usted selecciona el menú de herramientas, encontrará la opción **"GLOSARIO"**; al presionar enter en esta opción , aparecerá un menú con 4 opciones: Normas, Computación, Eléctrica y Sinónimos, en cada una de ellas podrá presionar "enter" para ver el contenido de cada opción. Si se presiona "enter" en la opción de Normas, aparecerán varios términos referentes a la normalización con sus respectivas definiciones. Así mismo si se escoge la opción "Computación", se encontrarán términos de computación usados a lo largo de la norma y del sistema en general. Si escogemos la opción "Eléctrica", se desplegarán términos referentes a las instalaciones eléctricas con sus respectivos significados y por último si se elige la opción "Sinónimos", se presentará una lista de sinónimos de palabras referentes a términos de instalaciones eléctricas que se utilizaron en este trabajo. (pantalla 22 y 23).

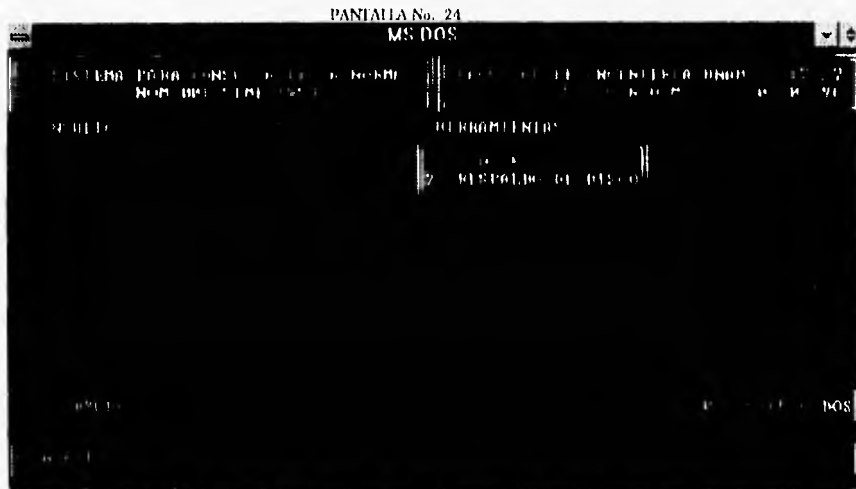
PANTALLA No. 22



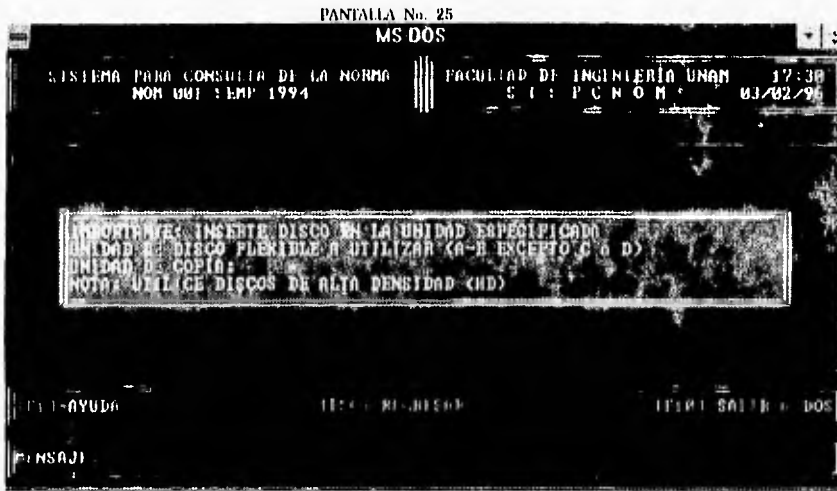
PANTALLA No. 23



RESPALDO DE DISCO :



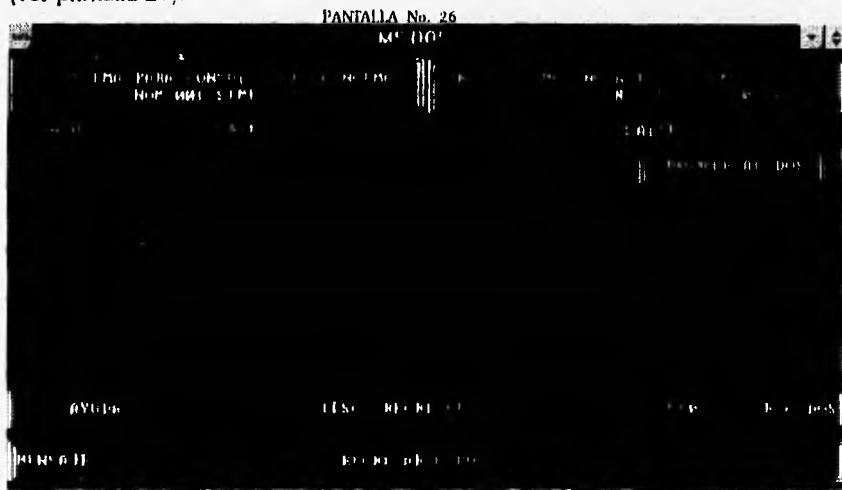
Si se elige la opción de Respaldar Información, se verá un mensaje que dice PREPARANDO ARCHIVOS PARA EL PROCESO, a continuación se verá una pantalla con instrucciones para llevar a cabo el respaldo de información, el cual dice : IMPORTANTE! INSERTE DISCO EN LA UNIDAD ESPECÍFICA, UNIDAD DE DE DISCO FLEXIBLE A UTILIZAR (A O B EXCEPTO C O D), UNIDAD ESPECÍFICA__ . Aquí habrá que poner la unidad en la que se desea que se respalde la información y meter el disco en dicha unidad. Cabe mencionar que para esto se deben utilizar discos de alta densidad, esto con el fin de que se almacene toda la información del sistema sin ningún inconveniente (ver pantalla 25).



3.3 MENÚ SALIR :

VOLVER AL DOS

Si se elige esta opción el sistema saldrá automáticamente al sistema operativo. Al momento de salirse despliega la pantalla de salida del sistema. Para volver a entrar consulte la parte de ACCESO AL SISTEMA (ver pantalla 26).



3.4 TECLAS AUXILIARES :

AYUDA DEL SISTEMA F1

Si usted oprime la tecla F1, automáticamente se desplegará una pantalla que contiene un menú o índice de ayudas (la ayuda se activará desde cualquier menú en que sea oprimido la tecla F1), para poder acceder a cualquier ayuda bastará posicionarse en el renglón correspondiente (ver pantalla 27). El índice de ayuda contiene lo siguiente.

- Opciones del menú de consultas
- Opciones del menú de reportes
- Opciones del menú de herramientas
- Teclas de ayuda
- Teclas para el movimiento del cursor
- Teclas del menú

PANTALLA No. 27



OPCIONES DEL MENÚ DE CONSULTAS

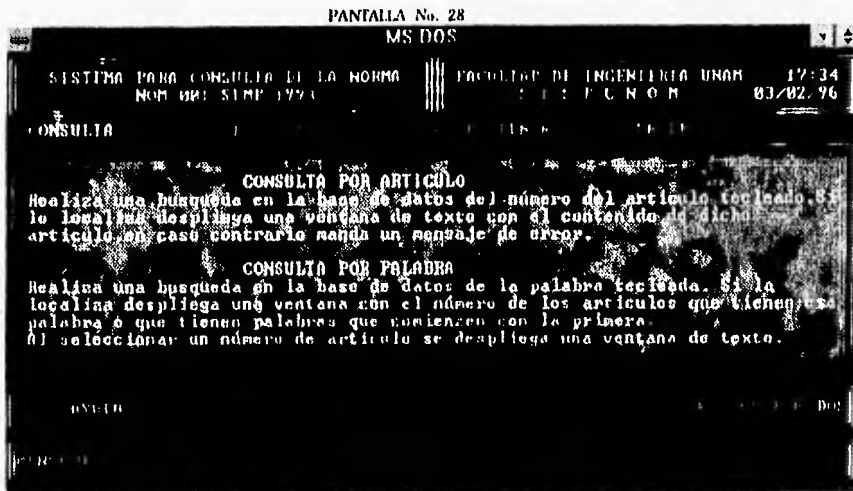
Si se elige esta opción aparecerá una pantalla que contiene una amplia descripción de como realizar las diferentes consultas que el sistema ofrece. Es decir, explica como hacer una consulta por artículo, una consulta por palabra o bien una consulta por índice; bastará teclear la tecla ESC para salir de esa pantalla y regresar al menú de ayuda. Si ya no se desea consultar otra opción del menú de ayuda, tecleando nuevamente ESC se regresa a la pantalla original de trabajo (ver pantalla 28).

OPCIONES DEL MENÚ DE REPORTES

Si se elige esta opción aparecerá una pantalla que contiene una amplia descripción de como realizar las diferentes reportes que el sistema ofrece, es decir, explica como poder enviar el reporte de un artículo a impresora y también enseña los pasos que se deben seguir para crear un reporte de artículo en un archivo. Este último caso es de gran utilidad para poder pasar un artículo que sea de su interés a un disco flexible almacenándolo en un archivo, y así poderlo transportar a otra máquina o bien imprimirlo desde otro lugar de trabajo. Bastará presionar la tecla ESC para salir de esa ayuda y regresar al menú de ayuda. Si ya no se desea consultar otra opción del menú de ayuda, presionando nuevamente la tecla ESC se regresa a la pantalla original de trabajo.

OPCIONES DEL MENÚ DE HERRAMIENTAS

Si se elige esta opción aparecerá una pantalla que contiene una amplia descripción de como utilizar las diferentes herramientas que el sistema ofrece, es decir, explica como poder acceder a la opción de glosario y como poder entrar al menú que este nos ofrece. A su vez proporciona una amplia información sobre cómo realizar el respaldo de la información de la norma a un disco. Bastará presionar la tecla ESC para salir de esa pantalla y regresar al menú de ayuda. Si ya no se desea consultar otra opción del menú de ayuda, presionando nuevamente ESC regresará a la pantalla original de trabajo.



TECLAS DE AYUDA

Si se elige esta opción, se proporcionarán las teclas de ayuda con que consta este sistema, en este caso en particular la única tecla de ayuda que se creó para el SISPCNOM es la tecla **F1**.

TECLAS PARA EL MOVIMIENTO DEL CURSOR

Si se elige esta opción, se proporcionará una descripción sobre las teclas que se necesitan para poder realizar los movimientos del cursor.

TECLAS DEL MENÚ

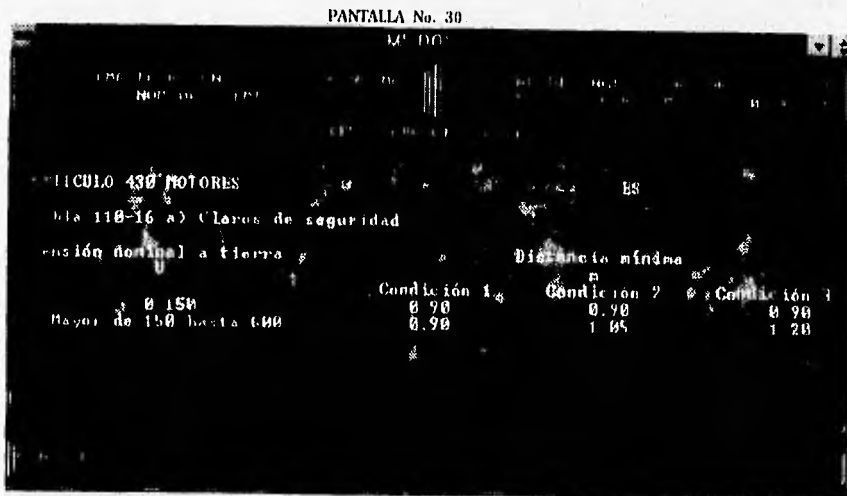
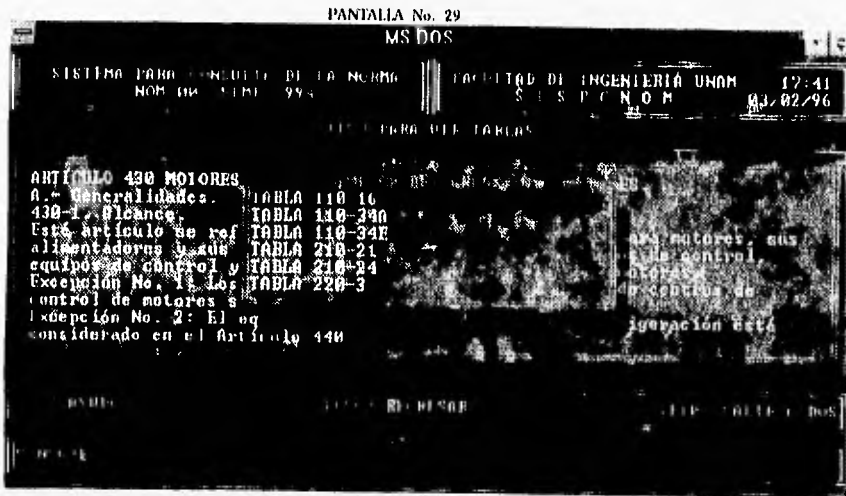
Si se elige esta opción, se proporcionará información correspondiente a las teclas que se necesitarán para poder hacer uso de los menús del sistema.

TECLA ESC

Esta tecla se utilizará cada vez que se desee regresar a la pantalla anterior, o bien puede ser utilizada cada que se haya cometido un error y se quiera regresar a corregirlo.

TECLA F5

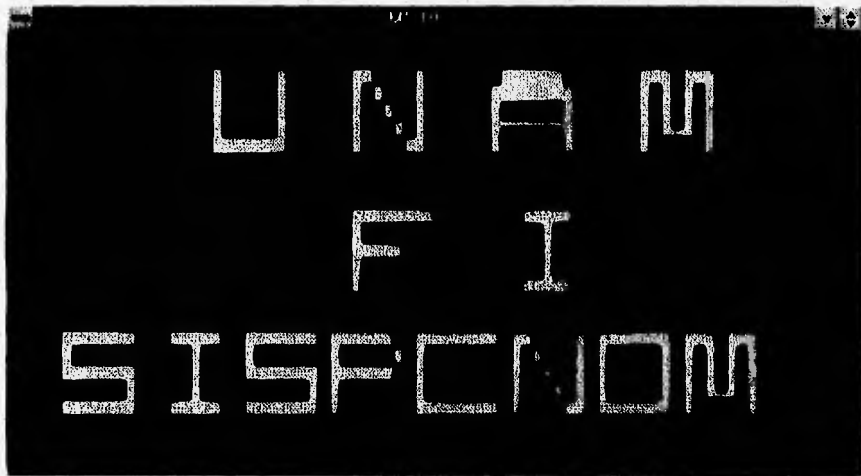
Despliega el índice de tablas siempre y cuando se encuentre dentro de un artículo realizando la búsqueda por palabra (ver pantalla 29 y 30).



TECLA F10

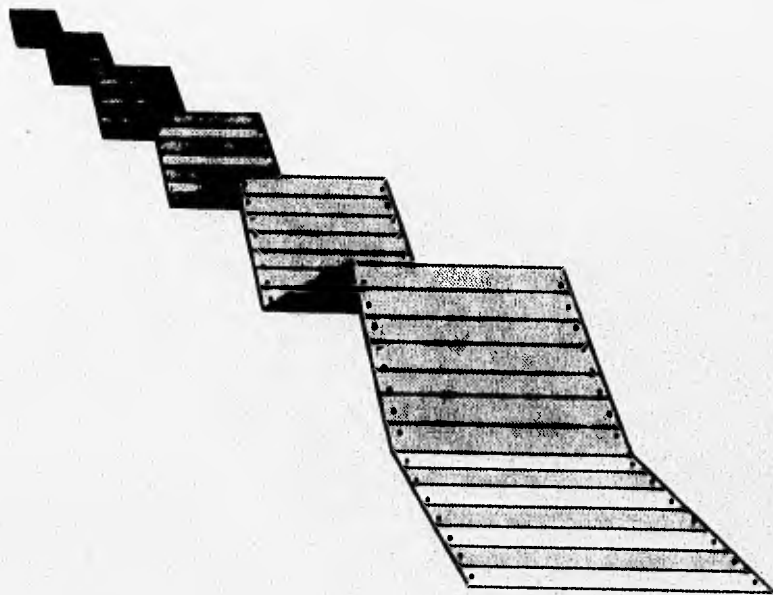
Esta tecla facilitará la salida al sistema operativo, ya que al hacer uso de ella, bloquea toda aplicación en que se encuentre el sistema SISPCNOM y regresará al sistema operativo después de desplegar una pantalla de salida (ver pantalla 31). Si desea volver a entrar al sistema consulte la parte de ACCESO AL SISTEMA.

PANTALLA No. 31



A N E X O 2

LISTADO DE PROGRAMAS



***** MODULO PRINCIPAL (SISPCNOM.PRG) *****

```
SET PROCEDURE TO AYUDA.PRG
SET PROCEDURE TO INFORME.PRG
SET PROCEDURE TO RESPALDA.PRG
SET PROCEDURE TO GLOSARIO.PRG
SET PROCEDURE TO PANTALLA.PRG
SET PROCEDURE TO TABLAS.PRG
CLEAR
PUBLIC level1, level2, level3, DEFINICAO:=.F.
DO WHILE .T.
level1 = 1          && memvar para primer menu
level2 = 1          && memvar para menus anidados
level3 = 1
aplicacion="FACULTAD DE INGENIERIA UNAM"
empresa1="SISTEMA PARA CONSULTA DE LA NORMA "
empresa2="NOM-001-SEMP-1994"
COLINICIO="GR+/B+" //Color de inicio para la pantalla de entrada
BOXCOL= "GR+/B,B/R+" //Color para la ventana
SETCOLOR(COLINICIO)
@1,15 SAY "SISTEMA PARA CONSULTA DE LA NORMA OFICIAL MEXICANA"
@4,32 SAY "NOM-001-SEMP-1994"
@7,29 SAY "FACULTAD DE INGENIERIA"
LOGO("SISPCNOM",11,4)
LOGO("UNAM",18,22)
INKEY(3)
SAVHOR2()
INICIO("S I S P C N O M")
INKEY(.05)
RECHOR2(1)
INKEY(2)
INICIO("S I S P C N O M")
INKEY(.05)
RECHOR2(2)
INKEY(2)
INICIO("S I S P C N O M")
INKEY(.05)
RECHOR2(3)
INKEY(2)
INICIO("S I S P C N O M")
INKEY(.05)
SELECT 1
USE NOM
SET WRAP ON
SET DATE BRITISH //formato de fecha
SET DELETED ON
SET SCORE OFF
SET MESSAGE TO 23 CENTER // visualiza MENSAJES en la linea 24
SET KEY -9 TO SALIR
KEYBOARD CHR(13)
DO WHILE (level1 <> 0)
SET KEY -4 TO
INICIO("S I S P C N O M")
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
@ 04,02 PROMPT "CONSULTA"
```

```

@ 04,21 PROMPT "REPORTES" //Menu Principal
@ 04,40 PROMPT "HERRAMIENTAS"
@ 04,59 PROMPT "SALIR"
MENU TO level1
  IF !FILE("NOM.DBF") //Si el archivo no esta manda un mensaje de error
    SOMBRA2(12,31,15,57)
    CUADRO(12,31,15,57,1,"W+/R+",.T.)
    SET COLOR TO W+/R+
    @ 13,32 SAY "NO SE ENCONTRO EL ARCHIVO"
    @ 14,41 SAY "NOM.DBF"
    ? REPLICATE(CHR(7),3) //Suena una alarma
    //SET CURSOR OFF //Hace invisible al cursor
    INKEY(0) //Espera a que se presione una tecla
    SET COLOR TO
    CLEAR
    QUIT
  ENDIF
  IF level1 <> 0
    DO sub_menu
  ENDIF
ENDDO
IF LASTKEY()=27 //Si se presiona la tecla ESC no se sale del sistema
  KEYBOARD CHR(13) //Envia al buffer del teclado un ENTER
  LOOP
ENDIF
ENDDO
RETURN

```

```

*****
* procedimiento sub_menu *
* Objetivo: Crea submenus para cada opcion del menu principal *
*****

```

```

PROCEDURE sub_menu && SUB MENU PROCEDURE
ligarflecha()
DO CASE
CASE level1 = 1
  SOMBRA2(5,2,9,18)
  CUADRO(5,2,9,18,1,BOXCOL,.F.)
  @ 06,03 PROMPT "1. POR ARTICULO" ;
  MESSAGE "BUSQUEDA POR ARTICULO ESPECIFICO" //Submenu de la opcion CONSULTA
  @ 07,03 PROMPT "2. POR PALABRA " ;
  MESSAGE "BUSQUEDA POR CONCEPTO"
  @ 08,03 PROMPT "3. POR INDICE " ;
  MESSAGE "SELECCIONA TEMA"
  MENU TO level2

  DO CASE
  CASE level2 = 1
    desligarflecha()
    ARTICULO() // llama al procedimiento ARTICULO
    ligarflecha()
  CASE level2 = 2
    desligarflecha()
    BUSCA() // llama al procedimiento BUSCA
    ligarflecha()

```

```
CASE level2 = 3
  desligarflecha()
  INDICE() // llama al procedimiento INDICE
  ligarflecha()
ENDCASE
CASE level1 = 2
  SOMBRA2(5,19,8,36)
  CUADRO(5,19,8,36,1,BOXCOL,.F.) // Crea una ventana con las coordenadas
  @ 06,20 PROMPT "1. EN IMPRESORA "; //especificadas.
  MESSAGE "IMPRIME REPORTE" //Submenu de la opcion REPORTES
  @ 07,20 PROMPT "2. EN ARCHIVO ";
  MESSAGE "GENERA ARCHIVO DE REPORTE"
  MENU TO level2
DO CASE
  CASE level2 = 1
    desligarflecha()
    INFORME_I() // llama al procedimiento INFORME_I
    ligarflecha()
  CASE level2 = 2
    desligarflecha()
    INFORME_A() // llama al procedimiento INFORME_A
    ligarflecha()
ENDCASE
CASE level1 = 3
  SOMBRA2(5,38,8,59) //Crea una sombra alrededor de una ventana
  CUADRO(5,38,8,59,1,BOXCOL,.F.)
  @ 6,39 PROMPT "1. GLOSARIO ";
  MESSAGE "DEFINICIONES DE TERMINOS"
  @ 7,39 PROMPT "2. RESPALDO DE DISCO"; // Submenu de la opcion HERRAMIENTAS
  MESSAGE "RESPALDA INFORMACION EN DISCO"
  MENU TO level2
DO CASE
  CASE level2 = 1
    desligarflecha()
    TERMINOS() //llama al procedimiento TERMINOS
    ligarflecha()
  CASE level2 = 2
    desligarflecha()
    RESP() //llama al procedimiento RESP
    ligarflecha()
ENDCASE
CASE level1 = 4
  SOMBRA2(5,60,7,77)
  CUADRO(5,60,7,77,1,BOXCOL,.F.)
  @ 06,61 PROMPT " VOLVER AL DOS "; //Sale al Sistema Operativo
  MESSAGE "REGRESAR A DOS"
  MENU TO level2
DO CASE
  CASE level2 = 1
    CLOSE ALL
    SALIR()
ENDCASE
ENDCASE
```

```
IF level2 = 0
  level2 = 1
ELSE
  KEYBOARD CHR(13)
ENDIF
SET KEY 19 TO
SET KEY 4 TO
RETURN
```

```
*****
*Procedimiento flecha_izq
*Objetivo: Envia al buffer del teclado el control de flecha izquierda
*****
```

```
PROCEDURE flecha_izq
PARAMETER pn,pl,rv
KEYBOARD CHR(27)+CHR(19)+CHR(13)
RETURN
```

```
*****
* Procedimiento flecha_der
*Objetivo: Envia al buffer del teclado el control de flecha derecha
*****
```

```
PROCEDURE flecha_der
PARAMETER pn,pl,rv
KEYBOARD CHR(27)+CHR(4)+CHR(13)
RETURN
```

```
*****
* Procedimiento ligarflecha
* Objetivo: Permite utilizar la tecla de flecha derecha o flecha izquierda para
* navegar en las ventanas de las diferentes opciones del Menu Principal.
*****
```

```
PROCEDURE ligarflecha
SET KEY 19 TO flecha_izq
SET KEY 4 TO flecha_der
```

```
*****
* Procedimiento desligarflecha
* Objetivo: Libera las teclas de flecha derecha y flecha izquierda para su
* uso comun.
*****
```

```
PROCEDURE desligarflecha
SET KEY 19 TO
SET KEY 4 TO
```



```
*****
* Funcion Sombra2
* Objetivo: Crea un efecto de sombra alrededor de una ventana ya creada
*****
```

```
FUNCTION SOMBRA2(LS,CS,LI,CI)
COL=SAVESCREEN(LS+1,CI+1,LI+1,CI+1)
LIN=SAVESCREEN(LI+1,CS+1,LI+1,CI+1)
```

```
FOR I = 2 TO LEN(COL) STEP 2
COL=STUFF(COL,I,1,CHR(8))
NEXT
FOR I = 2 TO LEN(LIN) STEP 2
LIN=STUFF(LIN,I,1,CHR(8))
NEXT
RESTSCREEN(LS+1,CI+1,LI+1,CI+1,COL)
RESTSCREEN(LI+1,CS+1,LI+1,CI+1,LIN)
RETURN(.T.)
```

```
*****
* Funcion CUADRO
* Objetivo: Crea una ventana con las coordenadas especificadas , con un color*
* de fondo y algun carácter de relleno en el caso de especificarlo.
*****
```

```
FUNCTION CUADRO(LS,cs,li,ci,tipo,COL,RELLENA)
local q[9]
COLBOX=COL
IF TIPO=NIL
TIPO=1
ENDIF
IF RELLENA=NIL
RELLENA=.F.
ENDIF
q[1]=CHR(201)+CHR(205)+CHR(187)+CHR(186)+;
CHR(188)+CHR(205)+CHR(200)+CHR(186)
q[2]=CHR(218)+CHR(196)+CHR(191)+CHR(179)+;
CHR(217)+CHR(196)+CHR(192)+CHR(179)
q[3]=CHR(213)+CHR(205)+CHR(184)+CHR(179)+;
CHR(190)+CHR(205)+CHR(212)+CHR(179)
q[4]=CHR(214)+CHR(196)+CHR(183)+CHR(186)+;
CHR(189)+CHR(196)+CHR(211)+CHR(186)
q[5]=CHR(220)+CHR(220)+CHR(220)+CHR(219)+;
CHR(219)+CHR(220)+CHR(219)+CHR(219)
q[6]=CHR(218)+CHR(196)+CHR(183)+CHR(186)+;
CHR(188)+CHR(205)+CHR(212)+CHR(179)
q[7]=CHR(201)+CHR(205)+CHR(184)+CHR(179)+;
CHR(217)+CHR(196)+CHR(211)+CHR(186)
```

```

IF VALTYPE(TIPO) = "C"
  Q[9]=REPL(TIPO,8)
  TIPO=9
ENDIF
IF (tipo >=176 .AND.tipo <=176) .OR.tipo=219
  IF RELLENA
    q[9]=REPLICATE(CHR(tipo),9)
  ELSE
    q[8]=REPLICATE(CHR(tipo),8)
  ENDIF
  tipo=8
ENDIF
IF TIPO<>8 .AND. RELLENA // LIMPIA EL FONDO DE LA VENTANA
  Q[TIPO]=Q[TIPO]+CHR(32)
ENDIF
IF COL<>NIL
  @ls,cs,li,ci box q[tipo] COLOR COLBOX
ELSE
  @LS,CS,LI,CI BOX Q[TIPO]
ENDIF

*****
* Funcion moldura
* Objetivo: Crea una moldura para la ventana de texto.
*****
function moldura(topo,esq,dir ,baixo,duplo,texto)
if pcount()<6
  @23,10 say "NUMERO DE PARAMETROS NO VALIDOS"
  return (nil)
endif
if duplo
  @topo,esq clear to dir ,baixo //identifica si se quiere la moldura
  @topo,esq to dir ,baixo //con caracter sencillo o caracter doble
else
  @topo,esq clear to dir ,baixo
  @topo,esq to dir ,baixo
endif
if len(texto)>0
  tam=baixo-esq
  novapos=(tam-len(texto))/2 //identifica si hay texto en el centro
  @topo,esq+novapos say texto //de la moldura
endif
return (nil)

*****
* Funcion INICIO
* Objetivo: Crea la pantalla principal del sistema,especificando la hora
* y la fecha actual.
*****
function INICIO(programa)
LOCAL COLACTUAL:=SETCOLOR()
SETCOLOR("GR+/B")
clear
@0,0 to 3,39 double
@0,40 to 3,79 double

```

```

@5,0,18,79 box replicate(chr(176),9) COLOR [G/B]
@5,0 to 18,79
@19,0 to 21,79 double
@22,0 to 24,79 double
@1,4 say empresa1
@2,12 say empresa2
@1,42 say aplicacion
@1,73 say left(time(),5)
@2,48 say programa
@2,70 say date()
@23,1 say "MENSAJE:"
@20,1 say "[F1]=AYUDA          [ESC]=REGRESAR          [F10]=SALIR A DOS"
SETCOLOR(COLACTUAL)
return nil

```

```

*****
* Funcion mensaje *
* Objetivo: Despliega un mensaje de error durante un tiempo determinado. *
*****

```

```
function mensaje(ls,cs,texto,tempo)
```

```
IF CS>19
```

```
CS=19
```

```
ENDIF
```

```
PANTM=SAVESCREEN(LS,CS,LS,CS+60) //Salva la pantalla actual en una variable
```

```
@ls,cs say space(LEN(TEXTO)) //Crea un espacio en blanco de la longitud
```

```
@ls,cs say texto COLOR [R/W] //escribe el texto
```

```
if tempo<>nil
```

```
inkey(tempo)
```

```
else //Si no se especifico un tiempo espera que
```

```
inkey(0) //se presione una tecla
```

```
endif
```

```
RESTSCREEN(ls,cs,LS,CS+60,PANTM) //Restaura la pantalla anterior
```

```
return nil
```

```

*****
* Procedimiento ARTICULO *
* Objetivo:Realiza la busqueda de un articulo especificado por el usuario. *
* Base de datos utilizada: NDM.DBF *
*****

```

```
PROCEDURE ARTICULO()
```

```
vclave=SPACE(4)
```

```
DO WHILE .T.
```

```
SELECT 1
```

```
USE NOM
```

```
INICIO("S I S P C N O M")
```

```
SOMBRA2(7,14,11,64)
```

```
CUADRO(7,14,11,64,1,"W+/BG+,W+/B+",.T.)
```

```
@ 8,28 SAY "ARTICULO A CONSULTAR " GET vclave; //Pregunta por articulo
```

```
PICTURE "@|" VALID vclave<>SPACE(4) //a consultar
```

```
READ
```

```
IF LASTKEY()=27 //Si se presiona la tecla ESC se sale del procedimiento
```

```
EXIT
```

```
ENDIF
```

```
SOMBRA2(13,28,15,52)
```

```
CUADRO(13,28,15,52,1,"W/B",.T.)
```

```
vcontrol=SPACE(1)
```

```

@ 14,31 SAY "CORRECTO ? (S/N)-"; //valida lo escrito
GET vcontrol PICTURE "[" VALID(vcontrol$SN")
READ
IF vcontrol$N"
  LOOP
ENDIF
IF vcontrol$S"
  CLEAR
  INICIO("S I S P C N O M")
  SET KEY -4 TO VER
  SET COLOR TO GR+/B+
  @4,30 SAY "[F5] PARA VER TABLAS" //Manda mensaje
  SELECT 1
  LOCATE FOR ARTICULO=vclave
  IF FOUND()
    SOMBRA2(6,1,17,76)
    SET COLOR TO N+/BG+
    MOLDURA(6,1,17,76,.T.,"ARTICULO" + " " + vclave)
    SETCOLOR("W+/BG+")
    MEMOEDIT(CONTIENE,7,2,16,77,.F.,"",76) //despliega el contenido del ar-
    ELSE //título si no encuentra artículo manda mensaje
    SOMBRA2(10,20,13,60) //de error.
    CUADRO(10,20,13,60,1,"W+/R+",.T.)
    SET COLOR TO W+/R+
    @ 11,29 SAY "NO EXISTE ESE ARTICULO !"
    @ 12,21 SAY "PRESIONE CUALQUIER TECLA PARA CONTINUAR"
    ?REPLICATE(CHR(7),3)
    INKEY(15) //Pausa por 15 segundos
    SET COLOR TO BG+/B+,W+/R+,R+,.G+/GR+
    INICIO("S I S P C N O M")
    LOOP
  ENDIF
  vclave=SPACE(4)
ENDIF
ENDIF
SET COLOR TO BG+/B+,W+/R+,R+,.G+/GR+
ENDDO
RETURN

```

```

*****
* Procedimiento Salir
* Objetivo:Regresa al ambiente del Sistema operativo MS-DOS
*****
PROCEDURE SALIR
SET COLOR TO GR+/N+
CLEAR
LOGO("U N A M",2,18)
LOGO("F I",10,31)
LOGO("SISPCNOM",17,4)
INKEY(2)
SAVVER2()
INICIO("S I S P C N O M")
INKEY(.05)
RECV2(1)
INKEY(2)
INICIO("S I S P C N O M")
INKEY(.05)

```

```

RECVER2(2)
INKEY(2)
INICIO("S I S P C N O M")
INKEY(.05)
  SET COLOR TO
  CLEAR
  CLOSE ALL
  QUIT

```

```

*****
* Procedimiento Busca
* Objetivo:Realiza la busqueda de una palabra en el contenido de la base de *
* datos.
* Base de datos utilizada: NOM.DBF
*****
PROCEDURE BUSCA()
DECLARE referencia[200]//Arreglo p/guardar #s de articulos con la palabra busc.
DECLARE lineas[200]//Arr. p/guardar #s lins. del campo memo de la palabra busc.
DECLARE columnas[200]//Arr. guarda #s de col. del campo memo de la palab. busc.
SELECT 1
USE NOM
buscar:= SPACE(40)
DO WHILE .T.
  INICIO("S I S P C N O M")
  SOMBRA2(9,10,11,69)
  CUADRO(9,10,11,69,1,"W+/BG+,W+/B+",.T.)
  @ 10,11 SAY "PALABRA A BUSCAR " GET buscar; //Pide una palabra para buscar
  PICTURE "@N" VALID buscar <> SPACE(40)
  READ
  IF ISDIGIT(buscar) //Si es digito no lo acepta
    LOOP
  ENDIF
  IF LASTKEY()=27
    EXIT
  ENDIF
  SOMBRA2(14,24,16,56)
  CUADRO(14,24,16,56,1,"W/B",.T.)
  vcontrol=SPACE(1)
  @ 15,25 SAY "PALABRA CORRECTA ? (S/N)->";
  GET vcontrol PICTURE "!" VALID(vcontrol$"SN")
  READ
  IF vcontrol$"N"
    LOOP
  ENDIF
  IF vcontrol$"S"
    INICIO("S I S P C N O M")
    SELECT 1
    SET COLOR TO W+*/B+
    @23,30 SAY "ESPERE UN MOMENTO..."
    k=1
    FOR indice:=1 TO 200
      ADEL(referencia,indice) //Inicializa el arreglo referencia con ceros
    NEXT
    encontrado=0 //Variable que controla el numero articulos en los
    // que se encontro la palabra.

```

```

j=1 //Subindice inicial de arreglos
n=2 //Apuntador al registro de la base de datos
buscar=RTRIM(buscar) //Elimina espacios en blanco a la derecha de la palabra
//Ciclo de busqueda en toda la BD, a excepcion de los apendices y tablas
VALOR=0 //Valor de avance del termometro
ULTIMO=LASTREC()-5 //Numero de registros a procesar.
TERMOH(VALOR) //Llama a la funcion que hace el termometro.
TOTPROC=0 //Numero actual de registros procesados.
TOTMOSTRA=0 //Numero de registros actual.
DO WHILE n<=LASTREC()-5
  SET COLOR TO W+/B+
  GO n //Localiza el registro n
  cad=CONTIENE //cadena de caracteres extraida del campo memo.
  IF ISLOWER(cad) //Si la cadena extraida esta en minusculas y la pa-
  IF ISLOWER(buscar) //labra buscada tambien, entonces convierte la
  buscar=UPPER(buscar) //la palabra a mayusculas.
  cad=UPPER(cad)
ENDIF
ENDIF
IF ISUPPER(cad) //Si la cadena extraida esta en mayusculas.
IF ISUPPER(buscar) .OR. ISLOWER(buscar) //y la palabra esta en mayus-
  buscar=LOWER(buscar) //culas o en minusculas, entonces
  cad=LOWER(cad) //convierte la palabra y la cadena en minusculas.
ENDIF
ENDIF
posicion=AT(buscar,cad)
IF posicion<>0 //Si encuentra la palabra
referencia[k]= ARTICULO //Guarda el # del ART. en el arreglo referencia.
k++ //Incrementa el indice del arreglo.
encontrado++ //Incrementa la variable de control de ocurrencias.
j++ //Incrementa el indice del arreglo lineas.
n++ //Incrementa apuntador a registro siguiente.
VALOR=(++TOTPROC)*100/ULTIMO//Calcula el val. que sera mostrado en el termometro
TERMOH(VALOR) //Llama al termometro.
MUESTRAREC(6,23) //Llama a la funcion para mostrar el numero de registros
ELSE //procesados.
  n++
  VALOR=(++TOTPROC)*100/ULTIMO
  TERMOH(VALOR)
  MUESTRAREC(6,23)
ENDIF
ENDDO
SET COLOR TO W+/B+
@23,9 CLEAR TO 23,70
IF encontrado>=1 //Si encontro la palabra en algun articulo inicia ciclo.
opc=1
DO WHILE .T.
  SET COLOR TO W+/B+
  INICIO("S I S P C N O M")
  SOMBRA2(6,17,16,62)
  CUADRO(6,17,16,62,1,"B+/BG+",.T.)
  @ 7,18 SAY "EL CONCEPTO SE ENCUENTRA EN EL ARTICULO..."
  opc=ACHOICE(8,35,15,39,referencia,,,opc) //Despliega el # de articulos en
  //los que se encontro la palabra.

```

```

IF opco=1 //Elegge una opcion
  LOCATE FOR ARTICULO= referencia[opc] //Localiza el # del articulo.
  IF FOUND() //Si lo encuentra despliega el contenido del articulo.
    CLEAR
    PRINTO( " I S P C N O M")
    SET COLOR TO GR+/B+
    @4,30 SAY " (F5) PARA VER TABLAS"
    SET KEY -4 TO VER
    COMBRAT(6,1,17,78)
    SET COLOR TO W+/BG+
    MOLDURA(6,1,17,78,.T.,"ARTICULO" + " " + referencia[opc])
    SET COLOR TO W+/BG+
  ENDF
  j=1 //Inicializa el subindice de arreglos para filas y columnas.
  inicio=1 //Variable que indica el inicio de la subcadena.
  texto=CONTIENE //El campo memo es asignado a la variable.
  bandera=0 //Controla la repeticion de la busqueda en el campo memo.
  busca ant=0
  DO WHILE .T.
    CLEAR 1
    IF busca ant=0 //Detecta si la busqueda no es hacia atras.
      SET COLOR TO W+/B+ //Despliega mensaje mientras busca la palabra
      @23,30 SAY "ESPERE UN MOMENTO..."
      IF bandera=1 //Repite la busqueda en la subcadena texto2.
        texto2=SUBSTR(texto,inicio)
        texto=texto2
      ENDF
      IF ISLOWER(texto) //Si la cadena extraida esta en minusculas y la pala-
      IF ISLOWER(buscar) //bra buscada tambien, entonces convierte la pala-
        buscar=UPPER(buscar) //bra a mayusculas.
        texto=UPPER(texto)
      ENDF
      ENDF
      IF ISUPPER(texto) //Si la cadena extraida esta en mayusculas y la pala-
      IF ISUPPER(buscar) .OR. ISLOWER(buscar) //bra esta en mayusculas o en
        buscar=LOWER(buscar) //minusculas, entonces convierte la palabra y la
        texto=LOWER(texto) //cadena en minusculas.
      ENDF
      ENDF
      fila=AT(buscar,texto)
      IF fila<>0 //Si encontro palabra obtiene la fila y la col. donde se encuentra
        tx=SUBSTR(texto,1,fila) //Extrae una subcadena desde el primer caracter
        IF inicio <>1 //hasta el ultimo en donde se encontro la palabra.
          lineas[j]=anterior+MLCOUNT(tx,76,4,.T.) //Para la repeticion de busqueda
        ELSE //se suma el # de linea de la busqueda anterior.
          lineas[j]=MLCOUNT(tx,76,4,.T.)//En caso contrario se indica en que linea esta.
        ENDF
        IF MOD(fila,76)<>0 //Determina en que columna se encontro la palabra.
          columnas[j]= MOD(fila,76)
        ELSE
          columnas[j]=76
        ENDF
      ELSE //Si no encuentra la palabra al repetir la busqueda se sale del ciclo.
        SET ESCAPE ON
        @23,9 CLEAR TO 23,78 //Borra mensaje parpadeante.
        MENSAJE(23,10,"NO HAY MAS PALABRAS EN TEXTO...",5)
    
```

```

EXIT
ENDIF
SET COLOR TO W+/B+
@23,9 CLEAR TO 23,78 //Borra mensaje parpadeante.
SET COLOR TO W+/BG+ //Despliega el contenido del campo memo.
MEMOEDIT (CONTIENE,7,2,16,77..F., "",76,"",lineas[j],columnas[j], "", "")
ELSE
SET COLOR TO W+/B+
@23,9 CLEAR TO 23,78 //Borra mensaje parpadeante.
SET COLOR TO W+/BG+ //Despliega el contenido del campo memo.
MEMOEDIT (CONTIENE,7,2,16,77..F., "",76,"",lineas[j],columnas[j], "", "")
ENDIF
vc=SPACE(1) //Variable para indicar si la busqueda se repite o finaliza.
@ 23,9 SAY "Buscar Palabra : Anterior Siguiente Palabra ? (A/S/F)->";
SET vc PICTURE "1" VALID(vc$"ASF")
SET ESCAPE OFF
READ
DO CASE
CASE vc$"A" //Si se escoge anterior repite la busqueda hacia atras.
j-- //Decrementa el subindice de los arreglos.
IF j<=0 //Si la busqueda apenas se va a repetir y se escoge esta opcion,
EXIT //se sale del ciclo puesto que no hay palabra antes.
j=1
ENDIF
busca_ant=1
SET KEY -4 TO VER //Activa la tecla F5 para tablas.
SET COLOR TO W+/B+
@23,9 CLEAR TO 23,78 //Borra mensaje parpadeante.
LOOP
CASE vc$"S" //Si se escoge siguiente repite la busqueda hacia adelante.
IF j>=200 //Si se agoto el tamano de los arreglos se sale
j=200
EXIT
ENDIF
busca_ant=0 //Indica que no es busqueda hacia atras.
bandera=1 //Indica que la busqueda se repite hacia adelante.
inicio= fila+LEN(buscar) //Inicializa el inicio de la subcadena con el valor
//de la posicion anterior de la palabra encontrada mas la longitud de la
//palabra buscada.
anterior=lineas[j] //Almacena el valor anterior de la fila en donde encon-
//tro la palabra.
j++ //Incrementa el subindice de los arreglos.
SET COLOR TO W+/B+
@23,9 CLEAR TO 23,78 //Borra mensaje parpadeante.
SET KEY -4 TO VER //Activa la tecla F5 para tablas
LOOP //Continua ciclo
CASE vc$"F" //Si se escoge finaliza termina la busqueda en ese articulo
SET ESCAPE ON //y se sale.
SET KEY -4 TO VER
EXIT //salir del ciclo.
OTHERWISE
SET ESCAPE OFF
SET KEY -4 TO
ENDCASE
ENDDO
SET ESCAPE ON

```



```

CLEAR //Borra pantalla.
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
INICIO("S I S P C N O M")
SET COLOR TO GR+/B+
@4,30 SAY "[F5] PARA VER TABLAS"
ELSE //Si no se selecciona nada se sale.
IF LASTKEY()=27
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
INICIO("S I S P C N O M")
EXIT
ENDIF
ENDIF
ENDDO
ELSE //Si no existe esa palabra manda un mensaje de error.
SOMBRA2(10,20,13,60)
CUADRO(10,20,13,60,1,"W+/R+,,.T.)
SET COLOR TO W+*/R+
@ 11,29 SAY "NO EXISTE ESE CONCEPTO!"
@ 12,21 SAY "PRESIONE CUALQUIER TECLA PARA CONTINUAR"
? CHR(7) //Suena alarma.
? CHR(7)
? CHR(7)
INKEY(20) //Pausa de 20 segundos.
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
INICIO("S I S P C N O M")
buscar:=SPACE(40) //Inicializa variable para buscar otra palabra
LOOP
ENDIF
buscar:= SPACE(40)
ENDIF
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
ENDDO
RETURN

```

```

*****
* Procedimiento Indice
* Objetivo:Despliega el indice tematico de la NOM, y es posible seleccionar
* los.
* Base de datos que utiliza: NOM.DBF
*****

```

```

PROCEDURE INDICE()
DECLARE indice[200] //Arreglo que guarda todos los temas de la NOM.
SELECT 1
USE NOM
FOR I=1 TO LASTREC()
GO I
indice[I]=DESCRIP //Inicializa el arreglo con los temas de la base de datos.
NEXT
DO WHILE .T.
CLEAR
INICIO("S I S P C N O M")
IF LASTKEY()=27
EXIT
ENDIF

```



```
SET COLOR TO G+ //Color verde para el caracter espacio.
@15,13 SAY REPLICATE(CHR(219),VALOR/2) //Muestra el caracter espacio avanzando
ENDIF //segun el proceso.
RETURN NIL
```

```
*****
* Funcion MUESTRAREC *
* Objetivo: Despliega un cuadro en el que aparece el numero de registros *
* procesados durante la busqueda por palabra en la base de datos. *
* Base de datos que utiliza: NOM.DBF *
*****
```

```
FUNCTION MUESTRAREC(LIN1,COL1)
PUBLIC TESTA:=0
IF TESTA=0
SET COLOR TO W+/BG+
@ LIN1,COL1 CLEAR TO LIN1+2,COL1+34//Borra parte de la pantalla p/el rectangulo
@ LIN1,COL1 TO LIN1+2,COL1+34 DOUBLE //Crea un rectangulo.
@ LIN1+1,COL1+3 SAY "REGISTROS : "
@ LIN1+1,COL()+1 SAY LASTREC()-5 PICTURE "@B 999"//Despliega el numero
@ LIN1+1,COL1+19 SAY "ACTUAL : " //de registros a consultar.
TESTA=1
ENDIF
@ LIN1+1,COL1+29 SAY STRZERO(++TOTMOSTRA,3) //Despliega el numero de registros
RETURN NIL //procesados actualmente.
```

```

***** MODULO RESPALDA.PRG *****
*****
* Procedimiento RESP
* Objetivo: Realiza el respaldo de las bases de datos del sistema.
* Bases de datos que utiliza: NOM.DBF, AYUDA.DBF, GLOSARIO.DBF.
*****
PROCEDURE RESP()
SET CURSOR ON
DISCO=SPACE(1) //Variable que guarda la UNIDAD.
DECLARE SISDBF[9]
SISDBF[1]="NOM.DBF"
SISDBF[2]="TABLAS.DBF"
SISDBF[3]="AYUDA.DBF"
SISDBF[4]="GLOSARIO.DBF"
SISDBF[5]="GLOSARIO.DBT"
SISDBF[6]="AYUDA.DBT"
SISDBF[7]="SISPCNOM.PRG"
SISDBF[8]="RECUPER.A.PRG"
INICIO("S I S P C N O M")
MENSAJE(23,10,"PREPARANDO ARCHIVOS PARA EL PROCESO...",2)
espacio=0
DO WHILE .T.
  SOMBRA2(9,5,14,74)
  CUADRO(9,5,14,74,1,"W+/BG+,W+/B+",.T.)
  SET COLOR TO W+/BG+
  @ 10,6 SAY "IMPORTANTE: INSERTE DISCO EN LA UNIDAD ESPECIFICADA"
  @ 11,6 SAY "UNIDAD DE DISCO FLEXIBLE A UTILIZAR (A-B EXCEPTO C O D)"
  @ 12,6 SAY "UNIDAD DE COPIA: "GET DISCO PICT "!" VALID DISCO<>SPACE(1)
  @ 13,6 SAY "NOTA: UTILICE DISCOS DE ALTA DENSIDAD (HD)"
  READ
  IF LASTKEY()=27
    EXIT
  ENDIF
  IF DISCO="A"
    espacio=DISKSPACE(1)
  ELSEIF DISCO="B"
    espacio=DISKSPACE(2)
  ENDIF
  IF espacio< 1440000
    IF espacio=0
      MENSAJE(23,10,"NO HAY DISCO EN LA UNIDAD!",5)
      LOOP
    ENDIF
    MENSAJE(23,10,"NO HAY ESPACIO SUFICIENTE EN EL DISCO",5)
    LOOP
  ENDIF
  ruta=DISCO+";" //Variable que guarda la ruta y la unidad.
  IF DISCO="C" .OR. DISCO="D"
    MENSAJE(23,10,"NO SE PUEDE COPIAR UN ARCHIVO SOBRE SI MISMO...",2)
    RETURN
  ENDIF
  DISCO1=ASC(DISCO)
  IF DISCO1 < 65 .OR. DISCO1 >66
    MENSAJE(23,10,"UNIDAD NO VALIDA",2)

```

```
SET ESCAPE OFF
@ 7,23.5 SAY "EL SISTEMA SISPCNOM SERA INSTALADO!"
@ 8,19 SAY "POR FAVOR ESCRIBA LA UNIDAD DE SU DISCO DURO"
@ 9,19 SAY "UNIDAD DE DISCO DURO: * GET DISCO ;
?GET *1* VALID DISCO<>SPACE(1)
READ
DISCO1=ASC(DISCO)
IF DISCO1 < 67 .OR. DISCO1 > 68
    MENSAJE(23,10,"UNIDAD NO VALIDA PARA DISCO DURO",4)
    SALIR()
ENDIF
IF DISCO="C"
    unidad=3
ELSEIF DISCO="D"
    unidad=4
ENDIF
SOMBRA2(12,21,15,60)
CUADRO(12,21,15,60,1,"W+/BG+",.T.)
SET COLOR TO W+/BG+
@ 13,22 SAY "ESPACIO DISPONIBLE:"
@ 13,COL() SAY DISKSPACE(unidad)
@ 13,COL()+1 SAY "BYTES"
@ 14,22 SAY "ESPACIO REQUERIDO:"
@ 14,COL()+1 SAY REQUIERE
@ 14,COL()+1 SAY "BYTES"
INKEY(5)
IF DISKSPACE(unidad)<9000000
    SOMBRA2(6,14,10,66)
    CUADRO(6,14,10,66,1,"W+/R+",.T.)
    SET COLOR TO W+/R+*
    @ 7,19 SAY "EL SISTEMA SISPCNOM NO PUEDE SER INSTALADO!"
    @ 8,20.5 SAY "POR FAVOR BORRE INFORMACION QUE NO OCUPE"
    @ 9,19 SAY "EN SU DISCO DURO Y REINTENTE LA INSTALACION"
    ? REPLICATE(CHR(7),5)
    INKEY(0)
    SALIR()
ENDIF
IF DISCO="C"
    MENSAJE(23,10,"CREANDO EL DIRECTORIO SISPCNOM EN UNIDAD C ...",3)
    RUN MD C:\SISPCNOM
ELSEIF DISCO="D"
    MENSAJE(23,10,"CREANDO EL DIRECTORIO SISPCNOM EN UNIDAD D ...",3)
    RUN MD D:\SISPCNOM
ENDIF
ruta=DISCO + ":"
N=1
VALOR=0 //Valor de avance del termometro.
ULTIMO=4 //Numero de archivos a copiar.
TOTPROC=0 //Numero actual de archivos copiados.
PRESENTA("I N S T A L A R")
detectado="DISCO1"
DO WHILE N<=1
    SOMBRA2(5,14,8,66)
    CUADRO(5,14,8,66,1,"W+/BG+",.T.)
    SET COLOR TO W+/BG+
    @ 6,20.5 SAY "INSERTE EL DISCO DE INSTALACION"
```

***** MODULO INSTALAR.PRG *****

```
*****
* Programa Instalar *
* Objetivo: Instalar el sistema SISPCNOM en un disco duro y configurarlo *
* para el buen desempeño del sistema. *
*****
aplicacion="FACULTAD DE INGENIERIA UNAM"
empresa1="PROGRAMA DE INSTALACION"
empresa2="DEL SISTEMA SISPCNOM"
SET DATE BRITISH
REQUIERE=9000000
DISCO=SPACE(1)
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
PRESENTA("I N S T A L A R")
COPIAR()
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
SET ESCAPE ON
PRESENTA("I N S T A L A R")
CONFIGURAR()
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
PRESENTA("I N S T A L A R")
AUTO()
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
PRESENTA("I N S T A L A R")
SOMBRA2(6,14,11,66)
CUADRO(6,14,11,66,1,"W+/R+",.T.)
SET COLOR TO W+/R+
SET cursor OFF
@ 7,28 SAY "INSTALACION FINALIZADA!"
@ 8,23.5 SAY "La computadora debe ser apagada y"
@ 9,16 SAY "encendida de nuevo para que las modificaciones"
@ 10,33 SAY "tengan efecto"
inkey(0)
SET COLOR TO
SET cursor ON
CLEAR
RETURN
```

```
*****
* Procedimiento Copiar *
* Objetivo:Crea el directorio SISPCNOM, copia los archivos al directorio y *
* descompacta la NORMA OFICIAL MEXICANA que se encuentra en la *
* base de datos NOM.DBF Y NOM.DBT *
*****
PROCEDURE COPIAR()
PUBLIC tabl1,tabla2
SET DEFAULT TO A:
SOMBRA2(6,14,10,66)
CUADRO(6,14,10,66,1,"W+/BG+,W+/B+",.T.)
SET COLOR TO W+/BG+
```

```
RETURN
ENDIF
FOR I=1 TO 8
  LAST=SEGREF(I) //Guarda en la variable el nombre del archivo.
  IF FILE ("%BASE.%") //Busca el archivo.
    IF LASTKEY()=27
      GOTO 1000
    ENDIF
    SAVE SCREEN TO P1 //Salva la pantalla actual en una variable.
    CLOSE DATABASES //Cierra todas las bases de datos
    MESSAGE(3,10,"COPIANDO ARCHIVOS PARA RESPALDO...",2)
    COPY FILE &BASE TO &ruta&BASE //Copia los archivos en la unidad especificada.
  ELSE
    MESSAGE(23,10,"ARCHIVO INEXISTENTE...",2)
  RETURN
ENDIF
NEXT
RESTORE SCREEN FROM P1 //Restaura la pantalla anterior.
GOTO 1000
RETURN
```

```

@ 7,15 SAY "PRESIONE CUALQUIER TECLA PARA CONTINUAR..."
INKEY(0)
IF !FILE("&detectado.")
  MENSAJE(23,10,"NO ES EL DISCO DE INSTALACION!",2)
  ? REPLICATE(CHR(7),5)
  LOOP
ENDIF
tabla1:=DIRECTORY("A:\*.*)
longitud=LEN(tabla1)
destino="\SISPCNOM\"
origen="A:\"
j=1
FOR I=1 TO longitud
  TERMOH(VALOR) //Llama al termometro.
  base=tabla1[I,1]
  @ 20,1 CLEAR TO 20,78
  @ 20,40 SAY base
  IF base<>"TODO.ZIP"
  COPY FILE &origen&base TO &ruta&destino&base
  VALOR=(+TOTPROC)*100/ULTIMO //Calcula el valor que sera mostrado en el ter-
  ENDIF //mometro.
  IF base="TODO.ZIP"
  SAVE SCREEN TO P1
  SET COLOR TO
  CLEAR
  IF DISCO="C"
  RUN PKUNZIP A:\TODO.ZIP C:\SISPCNOM
  ELSEIF DISCO="D"
  RUN PKUNZIP A:\TODO.ZIP D:\SISPCNOM
  ENDIF
  RESTORE SCREEN FROM P1
  ENDIF
  j++
  NEXT
N=N+1
ENDDO
RETURN

```

```

*****
* Procedimiento Configurar
* Objetivo: Realiza modificaciones o crea el archivo CONFIG.SYS
*****
PROCEDURE CONFIGURAR()
CR=CHR(13)+CHR(10)
BUFFERDEF="16"
FILESDEF="25"
alterado=.F.
config=GETENV("COMSPEC")
drconfig=LEFT(config,3)+"CONFIG.SYS"
NCONFIG=LEFT(config,3)+"CONFIG.001"
IF FILE("&DRCONFIG,")
COPY FILE &DRCONFIG. TO &NCONFIG.
***COMPROBANDO BUFFERS
pes=UPPER(MEMOREAD("&DRCONFIG,")
okb=AT("BUFFERS",pes)
arqhandle=FOFEN("&DRCONFIG",2)

```



```
IF arqhandle ==-1
  MENSAJE(23,10,"NO FUE POSIBLE LA APERTURA DEL ARCHIVO &DRCONFIG.",3)
  SALIR()
ENDIF
IF okb=0
  texto="BUFFERS="+BUFFERDEF+CR+CHR(26)
  fseek(arqhandle,-1,2)
  e=bin2i(freadstr(arqhandle,1))
  IF e=26
    fseek(arqhandle,-1,2)
  ENDIF
  ok=FWRITE(arqhandle,texto)
  IF ok < LEN(texto)
    MENSAJE(23,10,"ERROR DE ESCRITURA (FWRITE) : ",FERROR(),3)
    SALIR()
  ENDIF
  alterado=.t.
ELSE
  CHECAB=SUBS(PES,OKB,20)
  IGUALB=AT("=",CHECAB)
  FIMB=AT(CHR(10),CHECAB)
  VALBUF=VAL(SUBS(CHECAB,IGUALB+1,10))
  IF VALBUF<VAL(BUFFERDEF)
    FSEEK(ARQHANDLE,OKB+IGUALB-1,0)
    FWRITE(ARQHANDLE,BUFFERDEF+CR+SUBS(PES,OKB+FIMB))
    ALTERADO=.T.
  ENDIF
ENDIF
FCLOSE(ARQHANDLE)
*****COMPROBANDO FILES
pes=UPPER(MEMOREAD("&DRCONFIG."))
okf=AT("FILES",pes)
arqhandle=FOPEN("&DRCONFIG",2)
IF arqhandle ==-1
  MENSAJE(23,10,"NO FUE POSIBLE LA APERTURA DEL ARCHIVO &DRCONFIG.",3)
  SALIR()
ENDIF
IF okf=0
  texto="FILES="+FILESDEF+CR+CHR(26)
  fseek(arqhandle,-1,2)
  e=bin2i(freadstr(arqhandle,1))
  IF e=26
    fseek(arqhandle,-1,2)
  ENDIF
  ok=FWRITE(arqhandle,texto)
  IF ok < LEN(texto)
    MENSAJE(23,10,"ERROR DE ESCRITURA (FWRITE) : ",FERROR(),3)
    SALIR()
  ENDIF
  alterado=.t.
ELSE
  CHECAF=SUBS(PES,OKF,20)
  IGUALF=AT("=",CHECAF)
  FIMF=AT(CHR(10),CHECAF)
  VALFIL=VAL(SUBS(CHECAF,IGUALF+1,10))
  IF VALFIL<VAL(FILESDEF)
```

```

FSEEK (ARQHANDLE, OKF+IGUALF-1, 0)
FWRITE (ARQHANDLE, FILESDEF+CR+SUBS (PES, OKF+FIMF))
ALTERADO=.T.
ENDIF
ENDIF
FCLOSE (ARQHANDLE)
ELSE && ARCHIVO NO ENCONTRADO
  arqhandle = FCREATE("&DRCONFIG.")
  IF arqhandle =-1
    MENSAJE(23,10,"NO FUE POSIBLE LA CREACION DEL ARCHIVO &DRCONFIG.",3)
    SALIR ()
  ENDIF
  texto="DEVICE=HIMEM.SYS"+CR+"FILES="+FILESDEF+CR+"BUFFERS="+BUFFERDEF+CR
  texto=texto+"DEVICE=EMM386.EXE"+CR+CHR(26)
  FSEEK(arqhandle, 0, 2)
  FWRITE(arqhandle, texto)
  FCLOSE(arqhandle)
  alterado=.T.
  ENDIF
  IF alterado
    SOMBRA2(6,14,18,66)
    SET COLOR TO W+/GR+
    @ 6,14 CLEAR TO 18,66
    @ 6,14 TO 18,66 double
    @ 6,25 SAY "CONTENIDO ACTUAL DE CONFIG.SYS"
    @ 19,14 SAY "Pulse <ESC> para continuar.."
    SET COLOR TO W+/BG+
    MEMOEDIT(MEMOREAD("&DRCONFIG."),7,15,17,65,.f.)
  ELSE
    SOMBRA2(6,14,18,66)
    SET COLOR TO W+/GR+
    @ 6,14 CLEAR TO 18,66
    @ 6,14 TO 18,66 double
    @ 6,25 SAY "CONTENIDO ACTUAL DE CONFIG.SYS"
    @ 19,14 SAY "Pulse <ESC> para continuar.."
    SET COLOR TO W+/BG+
    MEMOEDIT(MEMOREAD("&DRCONFIG."),7,15,17,65,.f.)
    MENSAJE(23,10,"NO HUBO MODIFICACIONES EN CONFIG.SYS",2)
    MENSAJE(23,10,"CONFIG.SYS CHECADO!",3)
  ENDIF
RETURN

```

```

*****
* Procedimiento Auto
* Objetivo: Realiza modificaciones en archivo AUTOEXEC.BAT o lo crea
*****
PROCEDURE AUTO ()
CR=CHR(13)+CHR(10)
CLIPPERDEF="25"
EXPANDEF="256"
alterado=.F.
AUTOEXEC=GETENV("COMSPEC")
DRAUTOEXEC=LEFT(AUTOEXEC,3)+"AUTOEXEC.BAT"
pathdef=LEFT(AUTOEXEC,3)+"DOS"
NAUTOEXEC= LEFT(AUTOEXEC,3)+"AUTOEXEC.001"
IF FILE("&DRAUTOEXEC.")

```

```
COPY FILE &DRAUTOEXEC. TO &NAUTOEXEC.
****COMPROBANDO PATH
pes=UPPER(MEMOREAD("&DRAUTOEXEC."))
okb=AT("PATH",pes)
arqhandle=FOPEN("&DRAUTOEXEC",2)
IF arqhandle =-1
  MENSAJE(23,10,"NO FUE POSIBLE LA APERTURA DEL ARCHIVO &DRAUTOEXEC.",3)
  SALIR()
ENDIF
IF okb=0
  texto="PATH="+SUBS(PATHDEF,1)+CR+CHR(26)
  fseek(arqhandle,-1,2)
  e=bin2i(freadstr(arqhandle,1))
  IF e=26
    fseek(arqhandle,-1,2)
  ENDIF
  ok=FWRITE(arqhandle,texto)
  IF ok < LEN(texto)
    MENSAJE(23,10,"ERROR DE ESCRITURA (FWRITE) : ",FERROR(),3)
    SALIR()
  ENDIF
  alterado=.t.
ELSE
  CHECAB=SUBS(PES,OKB,127)
  FIMB=AT(CHR(13),CHECAB)
  EXISTEPATH=AT(SUBS(PATHDEF,2),CHECAB)
  IF EXISTEPATH=0
    FSEEK(ARQHANDLE,OKB+FIMB-2,0)
    FWRITE(ARQHANDLE,PATHDEF+CR+SUBS(PES,OKB+FIMB-1))
    ALTERADO=.T.
  ENDIF
ENDIF
FCLOSE(ARQHANDLE)
***** COMPROBANDO CLIPPER *****
pes=UPPER(MEMOREAD("&DRAUTOEXEC."))
okf=AT("CLIPPER=",pes)
arqhandle=FOPEN("&DRAUTOEXEC",2)
IF arqhandle =-1
  MENSAJE(23,10,"NO FUE POSIBLE LA APERTURA DEL ARCHIVO &DRAUTOEXEC.",3)
  RETURN
ENDIF
IF okf=0
  texto="SET CLIPPER=E"+EXPANDEF+";F"+CLIPPERDEF+CR+CHR(26)
  fseek(arqhandle,-1,2)
  e=bin2i(freadstr(arqhandle,1))
  IF e=26
    fseek(arqhandle,-1,2)
  ENDIF
  ok=FWRITE(arqhandle,texto)
  IF ok < LEN(texto)
    MENSAJE(23,10,"ERROR DE ESCRITURA (FWRITE) : ",FERROR(),3)
  ENDIF
  alterado=.t.
ELSE
  CHECAF=SUBS(PES,OKF,20)
```

```

IGUALF=AT("=",CHECAF)
FIMF=AT(CHR(10),CHECAF)
VALFIL=VAL(SUBS(CHECAF,IGUALF+2,10))
IF VALFIL<VAL(CLIPPERDEF)
  FSEEK(ARQHANDLE,OKF+IGUALF,0)
  FWRITE(ARQHANDLE,CLIPPERDEF+CR+SUBS(PES,OKF+FIMF))
  ALTERADO=.T.
ENDIF
ENDIF
  FCLOSE(ARQHANDLE)
ELSE && ARCHIVO NO ENCONTRADO
  arqhandle = FCREATE("&DRAUTOEXEC.")
  IF arqhandle ==-1
    MENSAJE(23,10,"NO FUE POSIBLE LA CREACION DEL ARCHIVO &DRAUTOEXEC.",3)
  return
ENDIF
  texto="SET CLIPPER=E"+EXPANDEF+";F"+CLIPPERDEF+CR+"PATH="+SUBS(PATHDEF,1)+CR+CH
  FSEEK(arqhandle,0,2)
  FWRITE(arqhandle,texto)
  FCLOSE(arqhandle)
  alterado=.T.
ENDIF
IF alterado
  SOMBRA2(6,14,18,66)
  SET COLOR TO W+/GR+
  @ 6,14 CLEAR TO 18,66
  @ 6,14 TO 18,66 double
  @ 6,25 SAY "CONTENIDO ACTUAL DE AUTOEXEC.BAT"
  @ 19,14 SAY "Pulse <ESC> para continuar.."
  SET COLOR TO W+/BG+
  MEMOEDIT(MEMOREAD("&DRAUTOEXEC."),7,15,17,65,.f.)
ELSE
  SOMBRA2(6,14,18,66)
  SET COLOR TO W+/GR+
  @ 6,14 CLEAR TO 18,66
  @ 6,14 TO 18,66 double
  @ 6,25 SAY "CONTENIDO ACTUAL DE AUTOEXEC.BAT"
  @ 19,14 SAY "Pulse <ESC> para continuar.."
  SET COLOR TO W+/BG+
  MEMOEDIT(MEMOREAD("&DRAUTOEXEC."),7,15,17,65,.f.)
  MENSAJE(23,10,"NO HUBO MODIFICACIONES EN AUTOEXEC.BAT",2)
  MENSAJE(23,10,"AUTOEXEC.BAT CHECADO!",3)
ENDIF
RETURN

```

```

*****
* Procedimiento Salir *
* Objetivo: Salir del instalador *
*****
PROCEDURE SALIR()
SET COLOR TO
CLEAR
QUIT

```

```
*****
* Funcion Sombra2 *
* Objetivo: Crea un efecto de sombra alrededor de una ventana ya creada *
*****
FUNCTION SOMBRA2(LS,CS,LI,CI)
COL=SAVESCREEN(LS+1,CI+1,LI+1,CI+1)
LIN=SAVESCREEN(LI+1,CS+1,LI+1,CI+1)
FOR I = 2 TO LEN(COL) STEP 2
  COL=STUFF(COL,I,1,CHR(8))
NEXT
FOR I = 2 TO LEN(LIN) STEP 2
  LIN=STUFF(LIN,I,1,CHR(8))
NEXT
REBSCREEN(LS+1,CI+1,LI+1,CI+1,COL)
REBSCREEN(LI+1,CS+1,LI+1,CI+1,LIN)
RETURN(.T.)
```

```
*****
* Funcion CUADRO *
* Objetivo: Crea una ventana con las coordenadas especificadas , con un color *
* de fondo y algun caracter de relleno en el caso de especificarlo. *
*****
FUNCTION CUADRO(LS,cs,li,ci,tipo,COL,RELLENA)
local q[9]
COLBOX=COL
IF TIPO=NIL
  TIPO=1
ENDIF
IF RELLENA=NIL
  RELLENA=.F.
ENDIF
q[1]=CHR(201)+CHR(205)+CHR(187)+CHR(186)+;
CHR(188)+CHR(205)+CHR(200)+CHR(186)
q[2]=CHR(218)+CHR(196)+CHR(191)+CHR(179)+;
CHR(217)+CHR(196)+CHR(192)+CHR(179)
q[3]=CHR(213)+CHR(205)+CHR(184)+CHR(179)+;
CHR(190)+CHR(205)+CHR(212)+CHR(179)
q[4]=CHR(214)+CHR(196)+CHR(183)+CHR(186)+;
CHR(189)+CHR(196)+CHR(211)+CHR(186)
q[5]=CHR(220)+CHR(220)+CHR(220)+CHR(219)+;
CHR(219)+CHR(220)+CHR(219)+CHR(219)
q[6]=CHR(218)+CHR(196)+CHR(183)+CHR(186)+;
CHR(188)+CHR(205)+CHR(212)+CHR(179)
q[7]=CHR(201)+CHR(205)+CHR(184)+CHR(179)+;
CHR(217)+CHR(196)+CHR(211)+CHR(186)
IF VALTYPE(TIPO)="C"
  Q[9]=REPL(TIPO,8)
  TIPO=9
ENDIF
IF (tipo >=176 .AND.tipo <=178) .OR.tipo=219
  IF RELLENA
    q[8]=REPLICATE(CHR(tipo),9)
  ELSE
    q[8]=REPLICATE(CHR(tipo),8)
  ENDIF
ENDIF
```

```

tipo=8
ENDIF
IF TIPO<>8 .AND. RELLENA // LIMPIA EL FONDO DE LA VENTANA
  Q[TIPO]=Q[TIPO]+CHR(32)
ENDIF
IF COL<>NIL
  @ls,cs,li,ci box q(tipo) COLOR COLBOX
ELSE
  @LS,CS,LI,CI BOX Q[TIPO]
ENDIF

```

```

*****
* Funcion INICIO *
* Objetivo: Crea la pantalla principal del sistema,especificando la hora *
* y la fecha actual. *
*****

```

```

function PRESENTA(programa)
LOCAL COLACTUAL:=SETCOLOR()
SETCOLOR("GR./B")
clear
@0,0 to 3,39 double
@0,40 to 3,79 double
@5,0,18,79 box replicate(chr(176),9) COLOR [G/B]
@5,0 to 18,79
@19,0 to 21,79 double
@22,0 to 24,79 double
@1,7 say empresa1
@2,14 say empresa2
@1,42 say aplicacion
@1,73 say left(time(),5)
@2,48 say programa
@2,70 say date()
@23,1 say "MENSAJE:"
SETCOLOR(COLACTUAL)
return nil

```

```

*****
* Funcion mensaje *
* Objetivo: Despliega un mensaje de error durante un tiempo determinado. *
*****

```

```

function mensaje(ls,cs,texto,tempo)
IF CS>19
  CS=19
ENDIF
PANTM=SAVESCREEN(LS,CS,LS,CS+60) //Salva la pantalla actual en una variable
@ls,cs say space(LEN(TEXTO)) //Crea un espacio en blanco de la longitud
@ls,cs say texto COLOR [R/W] //Escribe el texto.
if tempo<>nil
  inkey(tempo)
else //Si no se especifico un tiempo espera que se presione una tecla.
  inkey(0)
endif
RESTSCREEN(ls,cs,LS,CS+60,PANTM) //Restaura la pantalla anterior.
return nil

```


***** MODULO PASA.PRG *****

```
DO WHILE .T.
IF LASTKEY()=27
EXIT
ENDIF
CLEAR
SELECT 1
USE TABLAS
R:=SPACE(3)
@ 7,30 SAY " NUMERO DE REGISTRO" GET R ;
VALIDE R <> SPACE(3)
READ
R:=VAL(R)
ARCHIVO:=SPACE(12)
//RUTA="C:\ARCHIVOS"
@9,30 SAY "NOMBRE DEL ARCHIVO" GET ARCHIVO PICT "@X1";
VALID ARCHIVO <> SPACE(12)
READ
IF ARCHIVO="SALIR"
EXIT
ENDIF
LOC=LOC(" ",ARCHIVO)
IF POS=0
ARCHIVO=TRIM(ARCHIVO)
ENDIF
//ARCHIVO="C:\ARCHIVOS"+ARCHIVO
IF FILE("&ARCHIVO.")
@12,30 SAY "REPLAZAR ARCHIVO EN CAMPO..."
GO R
@ 14,30 SAY "REPLAZANDO..."
REPLACE TABLA WITH MEMOREAD("&ARCHIVO.")
ELSE
CLEAR
@ 12,30 SAY "ARCHIVO INEXISTENTE"
ENDIF
@ 16,30 SAY "PRESIONAR CUALQUIER TECLA "
INKEY(0)
ENDDO
CLOSE
```



```
***** MODULO AYUDA.PRG *****
*****
* Procedimiento HELP
* Objetivo: Despliega el indice tematico para la ayuda del sistema.
* Base de datos que utiliza:AYUDA.DBF
*****
PROCEDURE HELP
DECLARE ayuda_in[6]
SET KEY 28 TO //Desactiva la tecla F1 para la ayuda.
IF !FILE("AYUDA.DBF") //Si el archivo no esta manda un mensaje de error.
  SOMBRA2(12,31,15,57)
  CUADRO(12,31,15,57,1,"W+/R+",.T.)
  SET COLOR TO W+/R+
  @ 13,32 SAY "NO SE ENCONTRO EL ARCHIVO"
  @ 14,39 SAY "AYUDA.DBF"
  ? REPLICATE(CHR(7),3) //Suena una alarma.
  INKEY(0) //Espera a que se presione una tecla.
  SET COLOR TO
  CLEAR
  QUIT
ENDIF
SELECT 3
USE AYUDA
P:=SETCOLOR() //Guarda el color actual.
SAVE SCREEN TO PANTALLAH //Guarda la pantalla actual.
DO WHILE .T.
IF LASTKEY()=27
  EXIT
ENDIF
ayuda_in[1]="OPCIONES DEL MENU CONSULTAS" //Inicializa el arreglo.
ayuda_in[2]="OPCIONES DEL MENU REPORTES"
ayuda_in[3]="OPCIONES DEL MENU HERRAMIENTAS"
ayuda_in[4]="TECLAS DE AYUDA"
ayuda_in[5]="TECLAS PARA MOVIMIENTO DEL CURSOR"
ayuda_in[6]="TECLAS DE MENU"
SOMBRA2(7,23,13,57)
SET COLOR TO N+/BG+
MOLDURA(6,22,13,57,.T.,"INDICE DE AYUDA")
SET COLOR TO W+/BG+
op=1
DO WHILE .T.
op=ACHOICE(7,23,12,56,ayuda_in,,op) //Despliega el indice de la ayuda.
Q:=SETCOLOR() //Guarda el color del indice.
SAVE SCREEN TO PANTALLAM //Salva la pantalla del indice.
IF op<>0
  SELECT 3
  LOCATE FOR TIPO = ayuda_in[op]
  IF FOUND()
    TIPOS=RTRIM(TIPO)
    SOMBRA2(6,1,17,78)
    SET COLOR TO N+/BG+
    MOLDURA(6,1,17,78,.T.,TIPOS)
    SETCOLOR("W+/BG+")
```

```
MEMOEDIT(TEXTO,7,2,16,77,.F.,"",76) //Despliega el contenido del tema
ENDIF //seleccionado.
ELSE
  SET COLOR TO P //Restaura el primer color
  EXIT
ENDIF
@ 0,0 CLEAR TO 24,79
SET COLOR TO Q //Restaura el ultimo color del indice.
RESTORE SCREEN FROM PANTALLAM //Restaura la ultima pantalla del indice.
ENDDO
ENDDO
RESTORE SCREEN FROM PANTALLAH //Restaura la primera pantalla del prog. principal
SETCOLOR() //Restaura el color original de la pantalla.
SET KEY 28 TO HELP //Asigna la tecla F1 al procedimiento HELP.
RETURN
```

```
ELSE
  SET COLOR TO P //Restaura el primer color
  EXIT
ENDIF
@1,1 CLEAR TO 24,80
SET COLOR TO Q //Restaura el ultimo color del indice.
RESTORE SCREEN FROM PANTALLAM2 //Restaura la ultima pantalla del indice.
ENDDO
ENDDO
RESTORE SCREEN FROM PANTALLAH2 //Restaura la 1ra. pantalla del prog. principal
SETCOLOR() //Restaura el color original de la pantalla.
SET KEY +4 TO VER //Asigna la tecla F5 al procedimiento VER
RETURN
```

***** MODULO GLOSARIO.PRG *****

 * Procedimiento TERMINOS. *
 * Objetivo: Despiega el indice de los temas utilizados en el Glosario *

```

PROCEDURE TERMINOS()
DECLARE indiceg[4] //Arreglo para indice tematico
IF IFILE("GLOSARIO.DBF") //Si el archivo no esta manda un mensaje de error.
  SOMBRA2(12,31,15,57)
  CUADRO(12,31,15,57,1,"N+/R+",.T.)
  SET COLOR TO W+/R+
  @ 13,32 SAY "NO SE ENCONTRO EL ARCHIVO"
  @ 14,37.5 SAY "GLOSARIO.DBF"
  ? REPLICATE(CHR(7),3) //Suena una alarma.
  INKEY(0) //Espera a que se presione una tecla.
  SET COLOR TO
  CLEAR
  QUIT
ENDIF
SELECT 4
USE GLOSARIO
FOR I=1 TO LASTREC() //Inicializa el arreglo con los temas.
GO I
indiceg[I]=TEMA
NEXT
DO WHILE .T.
  CLEAR
  INICIO("S I S P C N O M")
  IF LASTKEY()=27
    EXIT
  ENDIF
  opcion=1
  DO WHILE .T.
    SOMBRA2(7,29,13,52)
    SET COLOR TO N+/BG+
    MOLDURA(7,29,13,52,.T.,"ESCOGE UNA OPCION")
    opcion=ACHOICE(9,31,14.51,indiceg,,opcion)
    IF opcion<>0
      SELECT 4
      USE GLOSARIO
      GO TOP
      SKIP
      LOCATE FOR TEMA = indiceg[opcion] //Localiza el tema en la base de datos.
      IF FOUND()
        CLEAR
        titulo:=RTRIM(TEMA)
        INICIO("S I S P C N O M")
        SOMBRA2(6,1,17,78)
        SET COLOR TO N+/BG+
        MOLDURA(6,1,17,78,.T.,"TERMINOS DE " + " " , titulo)
        SET COLOR TO W+/BG+
        MEMOEDIT(TEXTO,7,2,16.77,.F.,"",76)
      ENDIF
    ENDIF
  ENDWHILE
ENDWHILE

```

```
CLEAR
SET COLOR TO BG+/B+,W+/R+,R+.,G+/GR+
INICIO "S I S P C N O M"
ELSE
IF LANTREY-1-27
  CLEAR
  INICIO "S I S P C N O M"
  SET COLOR TO BG+/B+,W+/R+,R+.,G+/GR+
  EX
  INICIO
  INICIO
  INICIO
  SET COLOR TO W+/R+,BG+/B+,R+.,G+/GR+
  SET
  RETURN
```

```

***** MODULO INFORME.PRG *****
*****
* Procedimiento INFORME_A
* Objetivo: Crea un archivo que contiene el informe de un articulo determina
* do.
* Base de datos que utiliza: NOM.DBF
*****
PROCEDURE INFORME_A
arti:=SPACE(4)
DO WHILE.T.
IF LASTKEY()=27
EXIT
ENDIF
INICIO("S I S P C N O M")
SOMBRA2(8,14,12,64)
CUADRO(8,14,12,64,1,"W+/BG+,W+/B+",.T.)
@9,22 SAY "GENERAR INFORME DEL ARTICULO:" GET arti; //Pide # de articulo.
PICTURE "@!"
READ
IF arti=SPACE(4) //Si no escribe numero de articulo se sale
RETURN
ENDIF
SOMBRA2(14,28,16,52)
CUADRO(14,28,16,52,1,"W+/B+",.T.)
accepta:=SPACE(1)
@15,31 SAY "CORRECTO ? (S/N)-.";
GET acepta PICTURE "!" VALID(accepta$"SN") //Valida respuesta.
READ
IF acepta$"N"
LOOP
ENDIF
IF acepta$"S"
INICIO("S I S P C N O M")
SELECT 1
USE NOM
LOCATE FOR ARTICULO=arti
IF .NOT. FOUND() //Si no existe el articulo teclado manda mensaje de error.
MENSAJE(23,10,"ESE ARTICULO NO EXISTE",2)
LOOP
ENDIF
SOMBRA2(8,14,12,64)
CUADRO(8,14,12,64,1,"W+/BG+,W+/B+",.T.)
DO WHILE .T.
archivo:=SPACE(12) //Declara variable para nombre de archivo.
UNIDAD:=SPACE(1)
SET COLOR TO W+/BG+
@9,18 SAY "NOMBRE DEL ARCHIVO:" GET archivo;
PICTURE "@X!" VALID archivo <> SPACE(12)
@11,18 SAY "COPIAR ARCHIVO EN LA UNIDAD ?:" GET UNIDAD;
PICTURE "!" VALID UNIDAD<>SPACE(1)
READ
posicion=RAT( ":",archivo) //Detecta nombres invalidos de archivos
IF posicion<>0

```

```

MENSAJE(23,10,"NOMBRE DE ARCHIVO INVALIDO",2)
RETURN
ENDIF
ruta=UNIDAD+:" //Agrega la ruta de copiado
DISCO1=ASC(UNIDAD) //Identifica si la unidad es valida
IF DISCO1 < 65 .OR. DISCO1 >68
MENSAJE(23,10,"UNIDAD NO VALIDA",2)
RETURN
ENDIF
IF archivo=SPACE(15)//Si no se escribe ningun nombre se sale del procedimiento.
RETURN
ENDIF
POS=RAT(".",archivo) //Busca el punto de la extension.
IF POS=0
archivo=TRIM(archivo)+".INF" //Elimina espacios en blanco a la derecha y
ENDIF //agrega la extension .INF por default.
IF FILE("&ruta&archivo.") //Busca si ya existe el archivo.
SOMBRA2(15,13,17,67)
CUADRO(15,13,17,67,1,"W+/B+",.T.)
accepta:=SPACE(1)
@16,14 SAY "ARCHIVO YA EXISTE...SUSTITUYE AL ANTERIOR ? (S/N)->";
GET acepta PICTURE "!" VALID(accepta$"SN") //Valida respuesta.
READ
IF acepta$"S"
DELETE FILE("&ruta&archivo.") //Borra el archivo anterior para ser sustitui-
ELSE //do por el nuevo archivo.
LOOP
ENDIF
ENDIF
MENSAJE(23,10,"GENERANDO ARCHIVO DE INFORME...",2)
SET CONSOLE OFF
salida=CONTIENE //Almacena en la variable el campo memo.
MEMOWRIT("&ruta&archivo.",salida) //escribe el contenido de la variable en el
EXIT //archivo.
ENDDO
ENDIF
arti:=SPACE(4)
ENDDO
SET CONSOLE ON
RETURN

```

```

*****
* Procedimiento INFORME I *
* Objetivo: Imprime el Informe generado del articulo especificado *
* Base de datos que utiliza: NOM.DBF *
*****

```

```

PROCEDURE INFORME I
SET MARGIN TO 5 //Especifica un margen izquierdo de 5 para la impresora.
arti:=SPACE(4)
DO WHILE.T.
IF LASTKEY()=27
EXIT
ENDIF
INICIO("S I S P C N O M")
SOMBRA2(8,14,12,64)

```

```

CUADRO(8,14,12,64,1,"W+/BG+,W+/B+",.T.)
@9,22 SAY "IMPRIMIR INFORME DEL ARTICULO:" GET arti;//Pide el # del articulo.
PICTURE "@1"
READ
IF arti=SPACE(4)
  RETURN
ENDIF
SOMERA2(14,28,16,52)
CUADRO(14,28,16,52,1,"W+/B+",.T.)
accepta:=SPACE(1)
@15,31 SAY "CORRECTO ? (S/N)->";
GET accepta PICTURE "!" VALID(accepta$SN) //Valida la respuesta.
READ
IF accepta$N"
  LOOP
ENDIF
IF accepta$"S"
INICIO("S I S P C N O M")
SELECT 1
USE NOM
LOCATE FOR ARTICULO=arti
IF .NOT. FOUND() //Si no existe el # del articulo manda mensaje de error.
  MENSAJE(23,10,"ESE ARTICULO NO EXISTE",2)
  LOOP
ENDIF
IF ISPRINTER() //Detecta si la impresora esta lista.
  MENSAJE(23,10,"IMPRESORA LISTA",2)
ELSE
  MENSAJE(23,10,"ERROR NO ESTA LISTA LA IMPRESORA",2)
  EXIT
ENDIF
MENSAJE(23,10,"IMPRIMIENDO INFORME...",2)
SET CONSOLE OFF
filas=MLCOUNT(CONTIENE,70) //Obtiene el numero de lineas con longitud de 70
IF filas%55 = 0 //caracteres en el campo memo.
  pagina=INT(filas/55) //Calcula el numero de paginas que seran utilizadas
ELSE //para imprimir.
  pagina=INT(filas/55)+1
ENDIF
SET DEVICE TO PRINT //Indica que la salida sera en la impresora
numero=1 //Variables que controlan el numero de lineas en el campo
cuenta=55 //memo, las que se estan imprimiendo y las que se va a imprimir.
FOR p=1 TO pagina //Ciclo para las paginas que seran impresas.
  IF p=1
    comienza=PROW()+1 //Si es la primer pagina coloca la cabeza de la impresora
  ELSE //en la linea siguiente
    comienza=PROW()+9 //Si no 9 lineas abajo
  ENDIF
  @ comienza,40 SAY "PAGINA"
  @ PROW(),PCOL() SAY p
  @ PROW()+1,5 SAY DATE()
  @ PROW()+1,16 SAY "NORMA NOM-001-SEMP-1994 PARA INSTALACIONES ELECTRICAS"
  @ PROW()+1,22 SAY "INFORME REFERENTE AL ARTICULO SELECCIONADO"

```



```
FOR linea=numero TO cuenta //Ciclo que controla el numero de lineas impresas
  salida=MEMOLINE(CONTIENE,70,linea,6,.t.) //y las que se van a imprimir.
  G PROW()+1,5 SAY salida
NEXT
numero=numero+55 //Incremento de las variables de control de impresion.
cuenta=cuenta+55
NEXT
SET DEVICE TO SCREEN //Regresa al estado normal de pantalla la salida
EXIT
ENDIF
arti:=SPACE(4) //Inicializa la variable para pedir otro articulo para
ENDDO //imprimir.
SET CONSOLE ON
RETURN
```

```

***** MODULO TABLAS.PRG *****
*****
* Procedimiento VER
* Objetivo: Despliega el indice de tablas para la NOM
* Base de datos que utiliza: TABLAS.DBF
*****
PROCEDURE VER
DECLARE tabla_in[110] //Arreglo para almacenar nombres de tablas.
SET KEY -4 TO //Desactiva la tecla F5 para las tablas
IF !FILE("TABLAS.DBF") //Si el archivo no esta, manda un mensaje de error.
  SOMBRA2(12,31,15,57)
  CUADRO(12,31,15,57,1,"W+/R+",.T.)
  SET COLOR TO W+/R+
  @ 13,32 SAY "NO SE ENCONTRO EL ARCHIVO"
  @ 14,38.5 SAY "TABLAS.DBF"
  ? REPLICATE(CHR(7),3) //Suena una alarma.
  INKEY(0) //Espera a que se presione una tecla.
  SET COLOR TO
  CLEAR
  QUIT
ENDIF
SELECT 2
USE TABLAS
FOR I=1 TO 110
  GO I
  tabla_in[I]=NOMBRE
NEXT
GO TOP
P:=SETCOLOR() //Guarda el color actual.
SAVE SCREEN TO PANTALLAH2 //Guarda la pantalla actual.
DO WHILE .T.
  IF LASTKEY()=27
    EXIT
  ENDIF
  SOMBRA2(8,23,14,57)
  SET COLOR TO N+/BG+
  MOLDURA(7,22,14,57,.T.,"INDICE DE TABLAS")
  SET COLOR TO W+/BG+
  op=1
  DO WHILE .T.
    op=ACHOICE(8,23,13,56,tabla_in,,op) //Despliega el indice de las tablas.
    Q:=SETCOLOR() //Guarda el color del indice.
    SAVE SCREEN TO PANTALLAM2 //Salva la pantalla del indice.
    IF op<>0
      SELECT 2
      LOCATE FOR NOMBRE = tabla_in[op]
      IF FOUND()
        NOM=RTNIN(NOMBRE)
        SOMBRA2(9,2,19,78)
        SET COLOR TO N+/BG+
        MOLDURA(8,1,19,78,.T.,NOM)
        SET COLOR TO W+/BG+
        REPLACE TABLA WITH MEMOEDIT(TABLA,9,2,18,77,.T.,180,.1,1) //Despliega el con-
        tenido del tema seleccionado.
      ENDIF
    ENDIF
  ENDIF

```

***** MODULO PANTALLA.PRG *****

* Funcion. LOGO *
 * Objetivo: Aumenta a 80 columnas el tamaño del caracter introducido ante- *
 * riormente. Pueden aparecer hasta 8 caracteres. *

```

FUNCTION LOGO(TEXTO,LIN, COL)
LOCAL L1, L2, L3, L4, L5, G[6,23]
IF LIN=NIL
    LIN=10
ENDIF
IF COL=NIL
    COL=1
ENDIF
    
```

```

G[1,1]="A"
G[2,1]="  ÛÛÛÛÛÛ  "
G[3,1]="  ÛÛÛÛÛÛÛÛ  "
G[4,1]="  Û  Û  "
G[5,1]="  ÛÛÛÛÛÛÛÛ  "
G[6,1]="  Û  Û  "
    
```

```

G[1,2]="B"
G[2,2]="  ÛÛÛÛÛÛÛÛ  "
G[3,2]="  Û  Û  "
G[4,2]="  ÛÛÛÛÛÛÛÛ  "
G[5,2]="  Û  Û  "
G[6,2]="  ÛÛÛÛÛÛÛÛ  "
    
```

```

G[1,3]="C"
G[2,3]="  ÛÛÛÛÛÛÛÛÛÛ  "
G[3,3]="  Û  Û  "
G[4,3]="  Û  Û  "
G[5,3]="  Û  Û  "
G[6,3]="  ÛÛÛÛÛÛÛÛÛÛ  "
    
```

```

G[1,4]="D"
G[2,4]="  ÛÛÛÛÛÛÛÛÛÛ  "
G[3,4]="  Û  Û  "
G[4,4]="  Û  Û  "
G[5,4]="  Û  Û  "
G[6,4]="  ÛÛÛÛÛÛÛÛÛÛ  "
    
```

G[1,5] = "E" "
 G[2,5] = "ÜÜÜÜÜÜÜÜ " "
 G[3,5] = "Ü " "
 G[4,5] = "ÜÜÜÜÜÜ " "
 G[5,5] = "Ü " "
 G[6,5] = "ÜÜÜÜÜÜÜÜ " "

G[1,6] = "F" "
 G[2,6] = "ÜÜÜÜÜÜÜÜ " "
 G[3,6] = "Ü " "
 G[4,6] = "ÜÜÜÜÜÜ " "
 G[5,6] = "Ü " "
 G[6,6] = "Ü " "

G[1,7] = "G" "
 G[2,7] = "ÜÜÜÜÜÜÜÜ " "
 G[3,7] = "Ü Ü " "
 G[4,7] = "Ü ÜÜÜ " "
 G[5,7] = "Ü Ü " "
 G[6,7] = "ÜÜÜÜÜÜÜÜ " "

G[1,8] = "H" "
 G[2,8] = "Ü Ü " "
 G[3,8] = "Ü Ü " "
 G[4,8] = "ÜÜÜÜÜÜÜÜ " "
 G[5,8] = "Ü Ü " "
 G[6,8] = "Ü Ü " "

G[1,9] = "I" "
 G[2,9] = " ÜÜÜÜÜÜ " "
 G[3,9] = " Ü " "
 G[4,9] = " Ü " "
 G[5,9] = " Ü " "
 G[6,9] = " ÜÜÜÜÜÜ " "

G[1,10] = "J" "
 G[2,10] = " Ü " "
 G[3,10] = " Ü " "
 G[4,10] = " Ü " "
 G[5,10] = " Ü " "
 G[6,10] = " ÜÜÜÜÜÜÜÜ " "

G[1,11] = "K" "
 G[2,11] = "Ü Ü " "
 G[3,11] = "Ü ÜÜ " "
 G[4,11] = "ÜÜÜÜ " "
 G[5,11] = "Ü ÜÜ " "
 G[6,11] = "Ü Ü " "

G[1,12] = "L" "
 G[2,12] = "Ü " "
 G[3,12] = "Ü " "
 G[4,12] = "Ü " "
 G[5,12] = "Ü Ü " "
 G[6,12] = "ÜÜÜÜÜÜÜÜ " "

G[1,13]="M"
G[2,13]="ÜÜÜ ÜÜÜ "
G[3,13]="Ü Ü Ü "
G[4,13]="Ü Ü Ü "
G[5,13]="Ü ÜÜÜ Ü "
G[6,13]="Ü Ü "

G[1,14]="N"
G[2,14]="ÜÜ Ü "
G[3,14]="Ü Ü Ü "
G[4,14]="Ü Ü Ü "
G[5,14]="Ü Ü Ü "
G[6,14]="Ü ÜÜ "

G[1,15]="O"
G[2,15]="ÜÜÜÜÜÜÜ "
G[3,15]="Ü Ü "
G[4,15]="Ü Ü "
G[5,15]="Ü Ü "
G[6,15]="ÜÜÜÜÜÜÜ "

G[1,16]="P"
G[2,16]="ÜÜÜÜÜÜÜ "
G[3,16]="Ü Ü "
G[4,16]="ÜÜÜÜÜÜÜ "
G[5,16]="Ü "
G[6,16]="Ü "

G[1,17]="Q"
G[2,17]="ÜÜÜÜÜÜÜÜÜ "
G[3,17]="Ü Ü "
G[4,17]="Ü Ü "
G[5,17]="ÜÜÜÜÜÜÜÜÜ "
G[6,17]=" Ü "

G[1,18]="R"
G[2,18]="ÜÜÜÜÜÜÜÜÜ "
G[3,18]="Ü Ü "
G[4,18]="ÜÜÜÜÜÜÜÜÜ "
G[5,18]="Ü ÜÜÜÜ "
G[6,18]="Ü Ü "

G[1,19]="S"
G[2,19]="ÜÜÜÜÜÜÜÜÜÜÜ "
G[3,19]="ÜÜÜÜÜÜÜÜÜÜÜ "
G[4,19]="ÜÜÜÜÜÜÜÜÜÜÜ "
G[5,19]="ÜÜÜÜÜÜÜÜÜÜÜ "
G[6,19]="ÜÜÜÜÜÜÜÜÜÜÜ "

G[1,20]="T"
G[2,20]="ÜÜÜÜÜÜÜÜÜÜÜÜÜ "
G[3,20]="ÜÜÜÜÜÜÜÜÜÜÜÜÜ "
G[4,20]="ÜÜÜÜÜÜÜÜÜÜÜÜÜ "
G[5,20]="ÜÜÜÜÜÜÜÜÜÜÜÜÜ "
G[6,20]="ÜÜÜÜÜÜÜÜÜÜÜÜÜ "

```
G[1,21]="U"
G[2,21]="Ü  Ü "
G[3,21]="Ü  Ü "
G[4,21]="Ü  Ü "
G[5,21]="Ü  Ü "
G[6,21]="ÜÜÜÜÜÜ "

```

```
G[1,22]="V"
G[2,22]="Ü  Ü "
G[3,22]="Ü  Ü "
G[4,22]="Ü  Ü "
G[5,22]=" Ü  Ü "
G[6,22]=" Ü  Ü "

```

```
G[1,23]="W"
G[2,23]="Ü  Ü "
G[3,23]="Ü  ÜÜÜ Ü "
G[4,23]="Ü  Ü  Ü  Ü "
G[5,23]="Ü  Ü  Ü  Ü "
G[6,23]="ÜÜÜ  ÜÜÜ "

```

```
L1:=L2:=L3:=L4:=L5:=" "

```

```
Z=1

```

```
DO WHILE Z<=LEN(TEXTO)

```

```
  LETRA=UPPER(SUBS(TEXTO,Z,1))

```

```
  IF LETRA=" "

```

```
    L1=L1+SPACE(5)

```

```
    L2=L2+SPACE(5)

```

```
    L3=L3+SPACE(5)

```

```
    L4=L4+SPACE(5)

```

```
    L5=L5+SPACE(5)

```

```
  ELSE

```

```
    indice=ASCAN(G[1],LETRA)

```

```
    IF indice<>0

```

```
      L1+=G[2,indice]

```

```
      L2+=G[3,indice]

```

```
      L3+=G[4,indice]

```

```
      L4+=G[5,indice]

```

```
      L5+=G[6,indice]

```

```
   ENDIF

```

```
  ENDDIF

```

```
  Z=Z+1

```

```
ENDDO

```

```
$ LIN++,COL SAY L1

```

```
@ LIN++,COL SAY L2

```

```
@ LIN++,COL SAY L3

```

```
@ LIN++,COL SAY L4

```

```
@ LIN++,COL SAY L5

```

```
RETURN NIL

```

```
*****
* Funcion SAVHOR2 *
* Objetivo: Guarda la pantalla actual para su uso posterior por la funcion *
* RECHOR2 *
*****
FUNCTION SAVHOR2
PUBLIC MT[2]
MT[1]=SAVESCREEN(0,0,11,79)
MT[2]=SAVESCREEN(12,0,24,79)
RETURN NIL

*****
* Funcion RECHOR2 *
* Objetivo: Devuelve una pantalla guardada por la funcion SAVHOR2,dividida *
* horizontalmente en dos mitades *
*****
FUNCTION RECHOR2(NMOV)
IF NMOV=NIL
    NOMV=1
ENDIF
DO CASE
CASE NMOV=1 //Efecto de entrada de las dos mitades lateralmente.
    FOR I=0 TO 79
        RESTSCREEN(0,0,11,I,MT[1])
        INKEY(.05)
        IF LASTKEY()=13
            RETURN NIL
        ENDIF
    NEXT
    FOR I=79 TO 0 STEP -1
        RESTSCREEN(12,I,24,79,MT[2])
        INKEY(.05)
        IF LASTKEY()=13
            RETURN NIL
        ENDIF
    NEXT
CASE NMOV=2 //Efecto de entrada de las dos mitades desde el borde hacia el
    FOR I=0 TO 11 //centro.
        RESTSCREEN(0,0,I,79,MT[1])
        INKEY(.08)
        IF LASTKEY()=13
            RETURN NIL
        ENDIF
    NEXT
    FOR I=24 TO 12 STEP -1
        RESTSCREEN(I,0,24,79,MT[2])
        INKEY(.08)
        IF LASTKEY()=13
            RETURN NIL
        ENDIF
    NEXT
CASE NMOV=3 //Efecto de entrada de las dos mitades desde el centro hacia los
    FOR I=11 TO 0 STEP -1 //bordes.
        RESTSCREEN(I,0,11,79,MT[1])
        INKEY(.08)
    NEXT
```

```
IF LASTKEY()=13
  RETURN NIL
ENDIF
NEXT
FOR I=12 TO 24
  RESTSCREEN(12,0,I,79,MT[2])
  INKEY(.08)
  IF LASTKEY()=13
    RETURN NIL
  ENDIF
NEXT
ENDCASE
RETURN NIL
```

```
*****
*   Funcion SAVVER2                                     *
*   Objetivo: Guarda la pantalla actual para su uso posterior por la *
*   funcion RECVER2                                     *
*****
```

```
FUNCTION SAVVER2
PUBLIC MT[2]
MT[1]=SAVESCREEN(0,0,24,39)
MT[2]=SAVESCREEN(0,40,24,79)
RETURN NIL
```

```
*****
*   Funcion RECVER2                                     *
*   Objetivo: Devuelve una pantalla guardada por la funcion SAVVER2,dividida*
*   verticalmente en dos mitades                         *
*****
```

```
FUNCTION RECVER2(NMOV)
IF NMOV=NIL
  NOMV=1
ENDIF
DO CASE
CASE NMOV=1 //Efecto de entrada de las dos mitades desde los bordes hacia el
FOR I=0 TO 39 //centro.
  RESTSCREEN(0,0,24,I,MT[1])
  INKEY(.05)
  IF LASTKEY()=13
    RETURN NIL
  ENDIF
NEXT
FOR I=79 TO 40 STEP -1
  RESTSCREEN(0,I,24,79,MT[2])
  INKEY(.05)
  IF LASTKEY()=13
    RETURN NIL
  ENDIF
NEXT
CASE NMOV=2//Efecto de entrada en la que la mitad izquierda aparece surgiendo
FOR I=0 TO 24 //de la parte superior de la pantalla. Mientras tanto, la mitad
  RESTSCREEN(0,0,I,39,MT[1]) //derecha aparece subiendo.
  INKEY(.08)
```



```
IF LASTKEY()=13
  RETURN NIL
ENDIF
NEXT
FOR I=24 TO 0 STEP -1
  RESTSCREEN(I,40,24,79,MT[2])
  INKEY(.08)
  IF LASTKEY()=13
    RETURN NIL
  ENDIF
NEXT
ENDCASE
RETURN NIL
```

```

***** MODULO RECUPERA.PRG *****
*****
* UTILERIA DE RECUPERACION DE INFORMACION EN EL DISCO DE RESPALDO *
* Objetivo:Recupera los archivos del disco de respaldo *
* Bases de datos que utiliza:NOM,AYUDA,GLOSARIO *
*****
CLEAR
aplicacion="FACULTAD DE INGENIERIA UNAM"
empresa1="UTILERIA DE RECUPERACION"
empresa2="DEL DISCO DE RESPALDO"
DISCO=SPACE(1) //Variable que indica la unidad de disco a utilizar.
UNIDAD=SPACE(1) //Variable que indica la unidad a utilizar.
DO WHILE .T.
SET COLOR TO BG+/B+,W+/R+,R+,,G+/GR+
INICIO("R E C U P E R A")
MENSAJE(23,10,"PREPARANDO ARCHIVOS PARA EL PROCESO...",2)
SOMBRA2(9,5,15,74)
CUADRO(9,5,15,74,1,"W+/BG+,W+/B+",.T.)
SET COLOR TO W+/BG+
@ 10,6 SAY "IMPORTANTE: INSERTE DISCO EN LA UNIDAD A O B"
@ 11,6 SAY "UNIDAD DE DISCO FLEXIBLE: "GET UNIDAD PICT "!"
@ 12,6 SAY "UNIDAD DE DISCO DURO A UTILIZAR (C-D EXCEPTO A O B)"
@ 13,6 SAY "UNIDAD DE DISCO DURO: "GET DISCO PICT "!"
@ 14,6 SAY "NOTA: UTILICE LOS DISCOS DE RESPALDO "
READ
rutahd=DISCO+;" //ruta del disco duro
rutaf=UNIDAD+;" //ruta del disco flexible
IF LASTKEY()=27
SET COLOR TO
CLEAR
EXIT
ENDIF
IF DISCO="A" .OR. DISCO="B"
MENSAJE(23,10,"NO SE PUEDE COPIAR UN ARCHIVO SOBRE SI MISMO...",2)
SET COLOR TO
CLEAR
RETURN
ENDIF
DISCO1=ASC(UNIDAD)
IF DISCO1 < 65 .OR. DISCO1 >66
MENSAJE(23,10,"UNIDAD NO VALIDA",2)
SET COLOR TO
CLEAR
RETURN
ENDIF
DISCO2=ASC(DISCO)
IF DISCO2 < 67 .OR. DISCO2 >68
MENSAJE(23,10,"UNIDAD NO VALIDA",2)
SET COLOR TO
CLEAR
RETURN
ENDIF

```

```

SAVESCREEN TO F1
IF DISCO="C"
  CLEAR
  RUN PKUNZIP A:\TODO.ZIP C:\SISPCNOM
ELSEIF DISCO="D"
  RUN PKUNZIP A:\TODO.ZIP C:\SISPCNOM
ENDIF
RESTORE SCREEN FROM P1
ENDDO

```

```

*****
* Funcion INICIO *
* Objetivo: Crea la pantalla principal del sistema,especificando la hora *
* y la fecha actual. *
*****

```

```

function INICIO(programa)
LOCAL COLACTUAL:=SETCOLOR()
SETCOLOR("GR+/B+")
clear
@0,0 to 3,39 double
@0,40 to 3,79 double
@5,0,18,79 box replicate(chr(176),9) COLOR [G/B]
@5,0 to 18,79
@19,0 to 21,79 double
@22,0 to 24,79 double
@1,4 say empresa1
@2,16 say empresa2
@1,42 say aplicacion
@1,73 say left(time(),5)
@2,48 say programa
@2,70 say date()
@23,1 say "MENSAJE:"
@20,32 say " ESC=SALIR A DOS"
SETCOLOR(COLACTUAL)
return nil

```

```

*****
* Funcion mensaje *
* Objetivo: Despliega un mensaje de error durante un tiempo determinado. *
*****

```

```

function mensaje(ls,cs,texto,tempo)
IF CS>19
  CS=19
ENDIF
PANTM=SAVESCREEN(LS,CS,LS,CS+60) //Salva la pantalla actual en una variable.
@ls,cs say space(LEN(TEXTO)) //Crea un espacio en blanco de la longitud
@ls,cs say texto COLOR [R/W] //escribe el texto.
if tempo<>nil
  inkey(tempo)
else //Si no se especifico un tiempo, espera que se presiones una tecla.
  inkey(0)
endif
RESTSCREEN(ls,cs,LS,CS+60,PANTM) //Restaura la pantalla anterior.
return nil

```

```

*****
* Funcion Sombra2
* Objetivo: Crea un efecto de sombra alrededor de una ventana ya creada
*****
FUNCTION SOMBRA2 (LS,CS,LI,CI)
COL=SAVESCREEN (LS+1,CI+1,LI+1,CI+1)
LIN=SAVESCREEN (LI+1,CS+1,LI+1,CI+1)
FOR I = 2 TO LEN(COL) STEP 2
COL=STUFF(COL,I,1,CHR(8))
NEXT
FOR I = 2 TO LEN(LIN) STEP 2
LIN=STUFF(LIN,I,1,CHR(8))
NEXT
RESTSCREEN(LS+1,CI+1,LI+1,CI+1,COL)
RESTSCREEN(LI+1,CS+1,LI+1,CI+1,LIN)
RETURN(.T.)

*****
* Funcion CUADRO
* Objetivo: Crea una ventana con las coordenadas especificadas , con un color*
* de fondo y algun caracter de relleno en el caso de especificarlo.
*****
FUNCTION CUADRO(LS,cs,li,ci,tipo,COL,RELLENA)
local q[9]
COLBOX=COL
IF TIPO=NIL
TIPO=1
ENDIF
IF RELLENA=NIL
RELLENA=.F.
ENDIF
q[1]=CHR(201)+CHR(205)+CHR(187)+CHR(186)+;
CHR(188)+CHR(205)+CHR(200)+CHR(186)
q[2]=CHR(218)+CHR(196)+CHR(191)+CHR(179)+;
CHR(217)+CHR(196)+CHR(192)+CHR(179)
q[3]=CHR(213)+CHR(205)+CHR(184)+CHR(179)+;
CHR(190)+CHR(205)+CHR(212)+CHR(179)
q[4]=CHR(214)+CHR(196)+CHR(183)+CHR(186)+;
CHR(189)+CHR(196)+CHR(211)+CHR(186)
q[5]=CHR(220)+CHR(220)+CHR(220)+CHR(219)+;
CHR(219)+CHR(220)+CHR(219)+CHR(219)
q[6]=CHR(218)+CHR(196)+CHR(183)+CHR(186)+;
CHR(188)+CHR(205)+CHR(212)+CHR(179)
q[7]=CHR(201)+CHR(205)+CHR(184)+CHR(179)+;
CHR(217)+CHR(196)+CHR(211)+CHR(186)
IF VALTYPE(TIPO)="C"
Q[9]=REPL(TIPO,8)
TIPO=9
ENDIF
IF (tipo >=176 .AND.tipo <=178) .OR.tipo=219
IF RELLENA
q[8]=REPLICATE(CHR(tipo),9)
ELSE
q[8]=REPLICATE(CHR(tipo),8)
ENDIF

```

```
tipo=8
ENDIF
IF TIPO<>8 .AND. RELLENA // LIMPIA EL FONDO DE LA VENTANA.
  Q(TIPO)=Q(TIPO)+CHR(32)
ENDIF
IF COL<>NIL
  @ls,cs,li,ci box q(tipo) COLOR COLBOX
ELSE
  @LS,CS,LI,CI BOX Q(TIPO)
ENDIF
```

S I S P C N O M

**Sistema Para Consulta
de la Norma Oficial
Mexicana**

NOM-001-SEMP-1994.

