

36
29

**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**



FACULTAD DE INGENIERIA

**"DISEÑO E IMPLEMENTACION DE UN SISTEMA DE
MONITOREO DE CLAVES DE ACCESO A INTERNET
PARA LA VALIDACION DEL PROPIETARIO
A TRAVES DE RED UNAM"**

T E S I S

**QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION**

**P R E S E N T A N ,
DIAZ ZUÑIGA LILIANA
GUZMAN BAZAN XOCHITL VERONICA
LORIA CASTELLANOS OSCAR
SANCHEZ CRUZ LUIS GONZALO
VEGA HERNANDEZ MARCELA**

**TESIS CON
FALLA DE ORIGEN**



DIRECTOR DE LA TESIS: M.I. JUAN CARLOS ROA BEIZA.

MEXICO, D. F.

**TESIS CON
FALLA DE ORIGEN**

JULIO DE 1998.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



ÍNDICE

ÍNDICE	I
PREFACIO	V
PRÓLOGO	VIII
AGRADECIMIENTOS	X
OBJETIVO	XI

CAPITULO I.

SERVICIOS DE RED.

1.1 Funciones generales de la Dirección de Telecomunicaciones Digitales (DTD)	1
1.2 Ubicación Organizacional de la Dirección dentro de la DGSCA	7
1.2.1 Estándares y definiciones manejadas por esta Dirección	9
1.3 Análisis y selección del software para generar el Front-End del sistema	11



1.4 Diseño de Bases de Datos (Características principales y funcionamiento de bases de datos relacionales)	32
1.5 Sistemas manejadores de Bases de Datos para Pc's y elección de alguna (características, ventajas y desventajas)	62
1.6 Análisis y selección del software para ligar aplicaciones de Pc's con servidores	93

CAPITULO II.

PLANTEAMIENTO DE LA PROBLEMÁTICA Y PROPUESTA DE SOLUCIÓN.

2.1 Problemática actual del uso y servicio que presenta Internet a los usuarios	116
2.2 Requerimientos del usuario	153
2.2.1 Plan de Trabajo	163
2.2.2 Recopilación de la información	166
2.2.3 Clasificación de la información	172
2.3 Análisis (Diagrama de descomposición funcional)	174
2.4 Requerimientos mínimos del sistema de hardware y software en que deberá correr la aplicación	189
2.5 Propuestas de solución y elección de la óptima	197



CAPITULO III.**DESARROLLO DEL SISTEMA**

3.1 Especificaciones del diseño	210
3.1.1 Criterio de monitoreo y verificación	229
3.1.2 Manejo de datos	235
3.1.3 Generación aleatoria de claves de prueba	239
3.2 Diseño y construcción de bases de datos	241
3.2.1 Diagrama de bloques para desarrollo de los módulos principales	243
3.2.2 Diagrama de flujo de datos	251
3.2.3 Diccionario de Datos	253
3.2.4 Normalización de tablas	257
3.2.5 Diagrama entidad relación	263
3.2.6 Desarrollo de la base de Datos	265
3.3 Diseño y construcción del Front-End	282
3.3.1 Desarrollo de cada una de las pantallas para monitoreo, verificación y toma de decisiones	288
3.3.2 Validación de cada uno de los campos para cada una de las pantallas anteriores	304
3.4 Implementación del software para ligar la aplicación con el servidor	306
3.4.1 Desarrollo de la rutina para verificación e inhabilitación posterior de la clave	309
3.5 Integración del sistema	313
3.5.1 Pruebas de verificación de cada uno de los bloques	318
CONCLUSIONES	327



MANUAL DE USUARIO	330
MANUAL TÉCNICO	351
APÉNDICE A (Estructura Organizacional de la UNAM)	A1 - A5
APÉNDICE B (Visual Basic)	B1 - B25
APÉNDICE C (Cuarta y Quinta forma Normal)	C1 - C5
APÉNDICE D (Políticas de RedUNAM)	D1 - D7
APÉNDICE E (Formas de Inscripción)	E1 - E8
APÉNDICE F (Cuestionarios)	F1 - F4
APÉNDICE G (Ayuda en hipertexto)	G1 - G15
APÉNDICE H (Arquitectura Cliente/Servidor)	H1 - H17
APÉNDICE I (Interface Access 2.0-Visual Basic 3.0)	I1 - I6
APÉNDICE J (Terminología CISCO)	J1 - J5
APÉNDICE K (Listado del Programa)	K1 - K24
GLOSARIO	GL1 - GL12
BIBLIOGRAFÍA	BL1 - BL3



PREFACIO

El objetivo principal de esta tesis es dejar plasmado en un trabajo escrito, el desarrollo llevado a cabo con el fin de modernizar el actual esquema de acceso a Internet con el que la Universidad cuenta para brindar a los usuarios sus servicios, a través de RedUNAM; empleando herramientas de cómputo actuales, equipos y software de aplicación específico en el área de comunicaciones en red, como lo son servidores de comunicaciones, conmutadores telefónicos, estaciones de trabajo, computadoras personales, software para el manejo de la información, etc.

Asimismo, se pretende dejar sentadas las bases que motiven e impulsen a nuevos egresados a mejorar el esquema de acceso remoto a red con el que cuenta nuestra "Alma Mater".

La tesis se presenta en tres capítulos principales. El primero de los cuales es una breve introducción acerca de las funciones principales que son desarrolladas por la Subdirección de Telecomunicaciones Digitales de la DGSCA, así como su nivel organizacional; con el fin de ambientar al lector, proseguimos con el análisis y selección de las herramientas de cómputo que serán empleadas en la implementación del sistema piloto (BD, comunicaciones y Front-End).

El segundo capítulo es la parte medular de la tesis, en él se establece y analiza la problemática que enfrenta la UNAM al brindar el acceso a Internet, se explican sus principales servicios, como lo son: transferencia de archivos, correo electrónico, sesión remota, búsqueda de información y por su puesto la video conferencia entre otros tantos. Es aquí donde se definen los requerimientos del usuario, iniciando por los que debe contemplar el software, pasando por una breve descripción de la infraestructura de comunicaciones con que cuenta la DGSCA, para llegar al establecimiento de los requerimientos de hardware que se necesitarán para la implementación del sistema; se concluye este capítulo planteando diversas propuestas de solución y eligiendo la óptima.



Finalmente, en el capítulo III se dan las especificaciones para el monitoreo, manejo y verificación de los datos que constituirán las claves de pruebas del sistema, una vez terminado esto, se procede al diseño y construcción de la Base de Datos, del Front-End y la liga con el servidor, para concluir con la integración del sistema y la pruebas pertinentes de verificación modular y de integración.

Como testigo presencial y parte activa de este proyecto, es mi deber resaltar el arduo esfuerzo de este grupo de empeñosas personas, tanto en el desarrollo del sistema; como en todos los días y noches en vela invertidas para la recopilación y conclusión de este escrito; pero sobre todo en esos momentos "intensos" que se suelen pasar en situaciones de mucho convivio.

Los medios aquí desarrollados se presentan como plataforma principal en el control de usuarios remotos, debido a su gran flexibilidad, durabilidad, capacidad al cambio, bajo costo y alta compatibilidad con el esquema actual.

Al final de esta tesis se encuentran los manuales de usuario y técnico, con lo que esperamos hacer lo más sencillo posible la labor de investigación a los interesados en leer este trabajo, esperando les sirva de apoyo en su actividad profesional o como material de consulta en el desarrollo de su trabajo de tesis.

Claro esta que un proyecto de tal magnitud, que es parte central de la modernización en cómputo en nuestra máxima casa de estudios, y tan requerido por los servicios que brinda, se debe apegar a una serie de lineamientos tales como la consevación del actual esquema de conexión en estado óptimo, propiciar el desarrollo de software nacional; por lo tanto, este trabajo de tesis refleja el estado actual de una de los muchos proyectos de esta Universidad, el cual seguirá desarrollándose y poniéndose a punto.



Realmente resulta imposible describir con palabras la emoción, la inmensa alegría, pero por sobre todo el espíritu de satisfacción que embargó a este equipo de trabajo al ver alcanzada la meta tan anhelada de realizar las pruebas finales y la liberación del sistema, misma que estamos seguros será de utilidad para futuros ingenieros.

Ciudad Universitaria, Junio de 1996

Ing. Octavio López Chávez
 Subdirector de Operaciones.
 DGSCA-DCAA



PRÓLOGO

El vertiginoso desarrollo del que hemos sido testigos en cuanto a materia de cómputo se refiere y obliga a las instituciones que la emplean a esforzarse día a día a mantenerse en vanguardia, lo que involucra a personal altamente capacitado que revise, actualice y desarrolle nuevas y modernas herramientas tecnológicas, para hacer que su labor sea más eficiente.

La historia del desarrollo tecnológico a lo largo de la humanidad nos demuestra una y otra vez que su avance debe ser para acercar a los pueblos, y este trabajo de tesis constituye un ejemplo de lo anterior, pero claro esta, que de igual manera que crece la humanidad debe desarrollarse la computación y las comunicaciones .

Sin embargo el llevar a cabo las investigaciones para la adopción de nuevas tecnologías no es una tarea fácil, se requiere de tiempo y amplios conocimientos de las tecnologías existentes, así como de las necesidades de la institución; en el caso de la Universidad este aspecto se agudiza dado que la mayoría de su personal técnico esta saturado de trabajo por lo que es aquí, donde se resalta la importancia de los becarios y tesistas quienes de manera entusiasta la llevan a cabo con el único fin de beneficiar a su "ALMA MATER" .

Todos estos aspectos han sido superados por la Dirección General de Servicios de Cómputo Académico, con la única finalidad de ir abriendo camino, y a través de la experiencia tenida y brindando el mejor servicio en todo momento a todos y cada uno de los usuarios de esta Universidad.

Dado que el nivel en la calidad de estudios que ofrece la UNAM día a día se fortalece a los ojos del mundo entero, es necesario mantener en un estado óptimo su red de computadoras para que gente de otras instituciones nacionales y extranjeras, puedan visitarnos y observen la elevada tecnología que se puede desarrollar en México.



El presente trabajo es el resultado del meticuloso y exhaustivo proyecto de mejoramiento del sistema de acceso remoto a Internet a través de RedUNAM, desarrollado por estos jóvenes entusiastas, recién egresados de la carrera de Ingeniero en Computación y en su gran mayoría también del programa de becas-DCAA, con el único fin de servir a su Universidad y a su país apoyándose en los conocimientos adquiridos a través de 5 años de constante adquisición de conocimientos en la Facultad de Ingeniería de la UNAM.

Ciudad Universitaria, julio de 1996

Ing. Octavio López Chávez
Subdirector de Operaciones
DGSCA-DCAA



AGRADECIMIENTOS

En este breve espacio hacemos patente nuestro más sincero agradecimiento a la Máxima Casa de Estudios, la Universidad Nacional Autónoma de México, nuestra *Alma Mater*, por habernos brindado generosamente sus conocimientos que la han engrandecido durante toda su historia, y que nos ha transmitido incondicionalmente para apoyarnos en el sendero de nuestra vida profesional como ingenieros. De la misma forma queremos aunarnos a todo ese valioso grupo de tesis que ha sido apoyado por la Dirección General de Servicios de Cómputo Académico, que a través de la Dirección de Cómputo para la Administración Académica nos dio bases sólidas para nuestro desempeño como profesionistas de alta calidad, por medio de su plan de becas que ha dado reconocimiento a la UNAM a nivel nacional . .

Es muy satisfactorio haber culminado una etapa tan importante en nuestra vida, por lo que estamos comprometidos ha no quedarnos sólo aquí, sino a ver siempre más allá de las metas alcanzadas, para podernos superar como profesionistas y así lograr por medio de nuestro constante esfuerzo, dedicación y trabajo, la excelencia.

POR SIEMPRE, GRACIAS

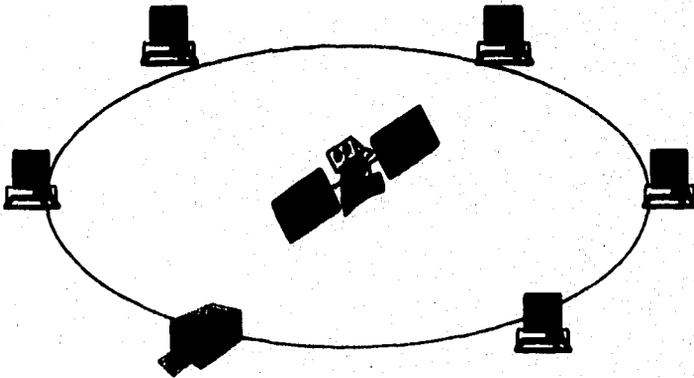


OBJETIVO

Diseñar un sistema que realice el monitoreo de claves de acceso a Internet, mejorando y dando un uso adecuado de las mismas, optimizando el servicio proporcionado por la UNAM. Para realizar este sistema, se utilizará un manejador de Bases de Datos de cuarta generación como back-end y como front-end, una herramienta gráfica que permita una interface visual amigable al usuario.

CAPITULO I

SERVICIOS DE RED





1.1 FUNCIONES GENERALES DE LA SUBDIRECCIÓN DE TELECOMUNICACIONES DIGITALES

La Dirección General de Servicios de Cómputo Académico¹, es la dependencia de la Universidad Nacional Autónoma de México (UNAM) encargada de mantener un alto nivel de tecnología en lo que a materia de cómputo se refiere, para posteriormente aplicarla en toda la diversidad de actividades que se desarrollan en cualquier Universidad de la altura de la UNAM.

Para tener un control más específico y adecuado sobre el manejo e instalación de esta tecnología, la DGSCA ha tenido bien deslindar responsabilidades en diversos departamentos y direcciones. Uno de los principales y más relevantes por la actividad que desarrolla, es la Dirección de Telecomunicaciones Digitales.

Esta subdirección es la encargada de mantener el buen funcionamiento de todo el sistema de comunicación con que cuenta la UNAM, esto incluye tanto al sistema telefónico (digital y analógico), así como la RED de computadoras (tanto en su infraestructura, expansión de computadoras, paquetería y servicios al usuario). Para ello cuenta con tres departamentos especializados.

El primero de estos departamentos es la Subdirección de Redes. En ella se cuenta con los servicios (funciones) de:

- Centro de Operación de la Red-NIC²
- Servidor de Gopher-NOC³
- Servidor de FTP⁴

En el servidor de Gopher se tiene información diversa, relacionada a la RedUNAM⁵

¹DGSCA

²Centro de Información de la Red (network Información Center).

³Centro de operaciones de la Red(Network Operation Center).

⁴Protocolo de Transporte de archivos (File Transport Protocol).



En el servidor de FTP, se provee software de dominio público para MS-DOS y para WINDOWS, para poder realizar la conectividad a los servicios de Gopher, correo electrónico, WWW, etc.

El objetivo de este departamento es facilitar el acceso a la información de la Red, así como promover todos los servicios que proporciona RedUNAM, la UNAM y otras Universidades e Instituciones en el mundo.

Algunos temas que aborde principalmente son:

- **Que es Gopher?**
Información en forma general de Gopher: qué es, cómo se usa, donde obtener información, creación de un servidor, etc.
- **Administradores de Redes Locales.**
Información acerca de los servicios que proporciona RedUNAM a los administradores de redes locales: soporte técnico, software, formas de conexión, etc.
- **Difusión sobre Internet.**
Información acerca de Internet: qué es Internet, Zen and art of Internet, library list, News. Bibliografía recomendada, etc.
- **Servicio de la UNAM.**
NIC, NOC, DNS, Solicitud de cuenta maestra, Supercomputo, laboratorio de visualización, cursos en la DGSCA, como anunciar en este Gopher, etc.
- **Otros Gopher.**

⁵Red perteneciente a la UNAM e interconectada a INTERNET.



Conexión a otros Gopher dentro de la UNAM y hacia el resto del mundo.

- **Redes Mexicanas.**

Instituciones anunciadas para Internet a través de la UNAM, Servicios de conectividad que ofrece la RedUNAM, contactos de red en la DGSCA.

- **Medios de Acceso a la RedUNAM.**

Conexión directa, Red X25 y Telepac, conexión vía módem (emulación de terminal y slip).

- **Reporte de Fallas.**

Procedimiento para reporte de fallas de la red

- **Nodos en la RedUNAM**

Lista de todas las dependencias conectadas a RedUNAM, dentro y fuera del campus universitario (En actualización).

- **Políticas de Uso Aceptable de la RedUNAM.**

Los lineamientos establecidos para el uso correcto de la RedUNAM.

- **Boletín de Información.**

Información de interés, Anuncios de cursos, congresos, avisos a los usuarios de Gopher Noc, Bolsa de Trabajo, etc.

El segundo departamento de esta Dirección, es la Coordinación de Servicios de Cómputo cuyo objetivo es proporcionar servicios de soporte informático y administrativo a los usuarios tanto de la red de la Universidad Nacional Autónoma de México (RedUNAM), como de la red Académica Nacional y Mundial.



Pone a disposición de los usuarios los servicios de :

- Servicios de información NIC-UNAM.
- Servicios de directorios y bases de datos de NIC-UNAM.
 - * Servicios de Búsqueda de Personas y Organizaciones.
 - * Información Organizada por Temas.
 - * Anuncios sobre Eventos.
 - * Servicios de Noticia en Internet.
 - * Información sobre seguridad en Cómputo.

Todo lo anterior con el objeto de promover el intercambio de ideas, pensamientos y opiniones que enriquezcan a los pueblos, Instituciones e individuos, en base a apoyar el crecimiento de la UNAM y de México, brindando una opinión tangible para el libre tránsito de información entre las diversas instituciones generadoras y transformadoras de conocimientos del país y del mundo para de esta forma, acercar los bancos de información y otras fuentes de conocimiento a todo aquel que así lo requiera, dentro y fuera de esta "Alma Mater".

Además en la Dirección de Redes y Comunicaciones se esta buscando constantemente nuevos servicios que ofrecer, siempre con la finalidad de brindarle al usuario una mejor opción para la adquisición de información nacional e internacional. La finalidad de tener un departamento encargado de los servicios de información (NIC) es el de proporcionar servicios de información a sus usuarios sobre diversos tópicos tanto de Internet como de la misma RedUNAM.

- Información Sobre Internet. (secciones)

- * Acerca de Internet



- ♦ Como Conectarse a Internet
 - ♦ Información para principiantes
 - ♦ Servicios Básicos
 - ♦ Sistemas para la localización de información especificada.
 - ♦ Páginas Interesantes.
-
- Información sobre RedUNAM.
(secciones).
 - ♦ Acceso a RedUNAM vía módem
 - ♦ Costos
 - ♦ Requerimientos
 - ♦ Números Telefónicos.
 - ♦ Emulación de Terminales.
 - ♦ Slip
 - ♦ Servicios que proporciona RedUNAM.
 - ♦ RedUNAM cuenta con un servidor de correo electrónico destinado tanto para la comunidad universitaria como para el público en general.
 - ♦ FTP anónimo.
 - ♦ Sesión Remota.
 - ♦ INFO-UNAM
 - ♦ Servidores WWW
 - ♦ Servicios de Cómputo que proporciona la UNAM.
 - ♦ Cómputo para la Docencia.
 - ♦ Centro de Extensión en Cómputo y Telecomunicaciones Nuevo León.
 - ♦ Desarrollo de Proyectos para Empresas e Instituciones.



- Dirección de Cómputo para la Administración Académica
- Laboratorio de Multimedia.
- Redes Mexicanas.
 - Red Metropolitana UNAM
 - R.D.I.
 - Red de la Región Noroeste.
 - Red Tecnológica Nacional (R. T. N.).
 - Red de Cómputo Centro-Occidente (REDCCO).
 - Red Institucional de información.

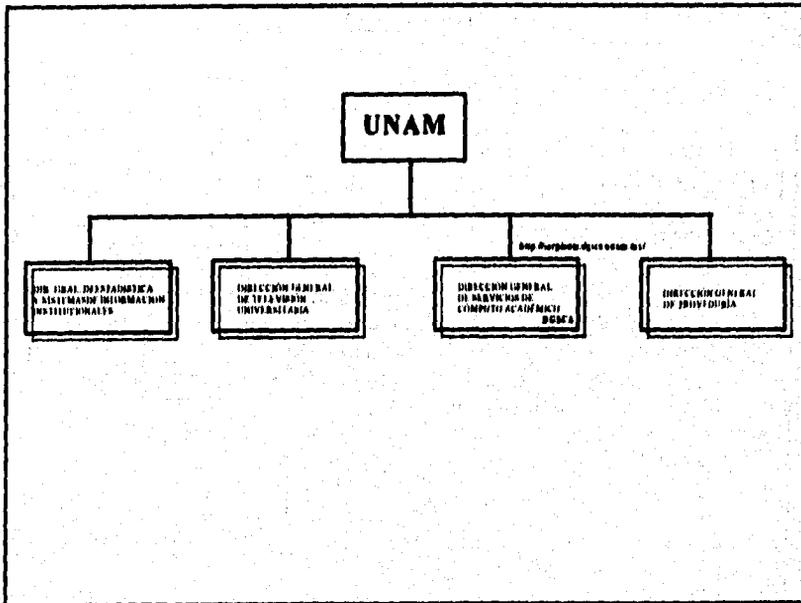
Si los usuarios no cuentan con dispositivos suficientes para conservar grandes cantidades de información pueden utilizar el servicio que brinda este departamento, el cual es el de almacenamiento masivo de información. Se tiene además un equipo de impresión láser disponible para la comunidad universitaria.

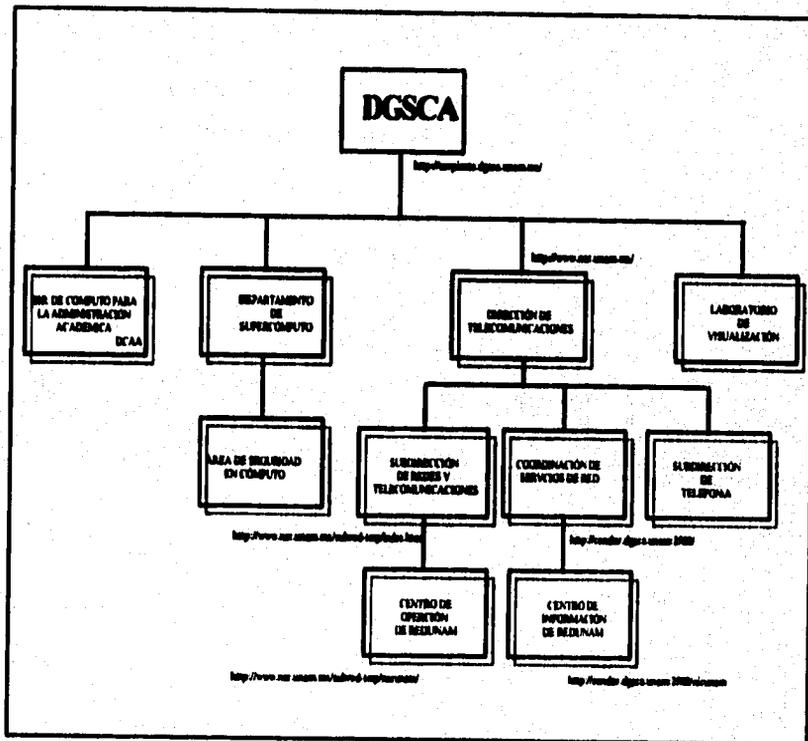
Para llevar a cabo una investigación científica y desarrollo tecnológico de una manera competitiva se requieren de servicios de cómputo modernos. Con la finalidad de proporcionar este tipo de servicio a la comunidad universitaria y productiva del país, la Universidad Nacional Autónoma de México a través de la DGSCA puso en operación en 1991 una supercomputadora CRAY Y-MP/432.

Finalmente, cabe hacer mención de que este departamento, desarrolla actividades paralelas de atención a usuarios en cuanto a posibles fallas tanto de hardware como de software, como lo pueden ser fallas de conexión entre las diversas plataformas (PC's, Terminales, Mainframes y toda la gama de estaciones de trabajo con las que cuenta la UNAM) y la red, dudas sobre el funcionamiento de software de aplicación, líneas telefónicas dentro de esta universidad, infraestructura de la misma RedUNAM. Auxiliando este servicio con registro de visitantes y buzones de dudas/sugerencias en la propia red (A través de Netscape, Mosaic o cualquier otro visualizador de información en Internet).



1.2 UBICACIÓN ORGANIZACIONAL DE LA DIRECCIÓN DENTRO DE LA DGSCA.







1.2.1 ESTÁNDARES Y DEFINICIONES MANEJADAS POR ESTA DIRECCIÓN.

Dentro de esta Dirección, como es de esperarse dadas sus dimensiones y alcances, se manejan una gran cantidad de términos y estándares relacionados con las actividades desempeñadas.

Sin embargo, debido a que son de uso frecuente, no se ha tenido la precaución de plasmar en un manual, todos estos conceptos manejados por el personal de esta dependencia; por lo que se procedió a indagar cuales eran los principales, para posteriormente poder implementar el manual de términos y estándares de la Dirección de Telecomunicaciones Digitales (en particular el departamento encargado de las claves de módem), con el fin de que sirva como referencia rápida tanto a los usuarios miembros de esta organización como a cualquier usuario externo que así lo requiera.

Simultáneamente se decidió integrar ciertos estándares que serán implementados en este Departamento, para el buen funcionamiento del sistema de monitoreo de claves vía remota, los cuales son indispensables.

En seguida se muestra una lista parcial de los términos y estándares que se presentaron para su aceptación a la Dirección General de Servicios de Cómputo Académico⁶; (para una referencia más completa sobre estos tópicos, pasar al glosario al final del tomo).

ESTÁNDARES.

Generación de claves monitoreadas:

⁶ DGSCA



Se entenderá por login o clave de usuario a una serie de 9 caracteres, la cual estará conformada por 4 caracteres alfabéticos y los restantes 5 serán numéricos, con las especificaciones mostradas a continuación:

ILCO00000 LOGIN

Letras Enteros

El significado de las letras según su posición es la siguiente:

Posición (De izq. a der.)	Significado
1	Primer letra del apellido paterno
2	Primer letra del apellido materno
3	Primer letra del primer nombre de pila
4	Tipo de usuario (I=interno, E=externo, etc.)

Los cinco últimos caracteres, son proporcionados automáticamente por el sistema. Aclaramos que este número no está duplicado, dadas las características del tipo de dato que se maneja.

TÉRMINOS MANEJADOS MAS COMÚNMENTE.

MODEM....

SLIP....

TACACS...

ESTACIÓN DE TRABAJO...

PROCOMM

LOGIN

PASSWORD....

etc....



1.3. ANÁLISIS Y SELECCIÓN DEL SOFTWARE PARA GENERAR EL FRONT-END DEL SISTEMA

DEFINICIÓN

El Front-End de un sistema es un programa de computadora que oculta detalles del acceso a los datos de otra computadora. En un sentido, todo programa es un front-end que evita que los usuarios conozcan los detalles reales del intrincado funcionamiento de la computadora. La mayor parte del tiempo, un front-end significa el uso de la computadora en mayor medida que muchos de los programas. Un ejemplo de procesador de éste tipo, es el que se utiliza como enlace de comunicaciones para un mainframe. En otras palabras, un front-end es un procesador de formatos, es decir, un procesador de entrada/salida utilizado para realizar el formato de los datos de entrada y/o procesarlos. A veces, se utiliza el término para referirse a un programa de comunicación.

PROGRAMACIÓN VISUAL

El éxito de Windows en el entorno PC ha obligado a los programadores a reciclarse para poder desarrollar aplicaciones que respeten el "look", y las necesidades de esta interface gráfica. La llamada "programación visual" supone un paso más en la búsqueda de nuevas herramientas que faciliten el desarrollo de software para Windows.

Empresas como Microsoft y Borland se han apresurado a lanzar al mercado versiones de sus productos con las palabras "visual", pero todavía no está muy claro el significado de este término. Se podría decir que "programación visual" se entiende como el proceso de desarrollo de forma interactiva del entorno del usuario para sus aplicaciones. La idea es que la "programación visual" permite que la mayor parte del proceso necesario para crear cualquier aplicación se pueda realizar en forma automática.



Así por ejemplo, si la mayoría de las aplicaciones de Windows disponen de menús desplegables, parece razonable desarrollar una herramienta para crearlos sin tener que recurrir a páginas de código indescifrable. Pero los beneficios no acaban aquí, porque todo lo mencionado hace tiempo que estaba disponible en los editores de recursos, ahora se cuenta con mayor integración entre el desarrollo de aplicaciones propiamente dicha y el diseño de interface de usuario.

Los editores de recursos que actualmente podemos encontrar en el mercado (en los compiladores de Pascal o C para Windows siempre viene alguno) siguen estas pautas:

Primero se crea el recurso deseado -que puede ser una ventana, un menú, un icono, algún tipo de letra, un botón, un campo de texto o cualquier otro elemento típico de Windows- después se genera el código correspondiente al recurso y, por último, con más o menos pasos intermedios, se acaba uniéndolo al resto del código fuente de nuestro programa como se ve en la figura 1.3.1.

Con la "Programación Visual" se pretende simplificar al máximo todo el proceso, evitando cuantos pasos intermedios sea posible. Lo normal es que se parta de la creación de una interface de usuario y se llegue a la escritura del código (si es necesario) en algún lenguaje de programación.

Por el momento, la "Programación Visual" ha dado un paso gigante en la tarea de facilitar la vida a los informáticos: sin embargo, todavía no ha llegado el día en el que cualquier persona sea capaz de decirle a la computadora mediante un micrófono lo que quiere y que ésta lo haga. Hasta entonces tendremos que seguir trabajando con las herramientas disponibles, muchas de las cuales ya han comenzado su transición a la "Programación Visual".

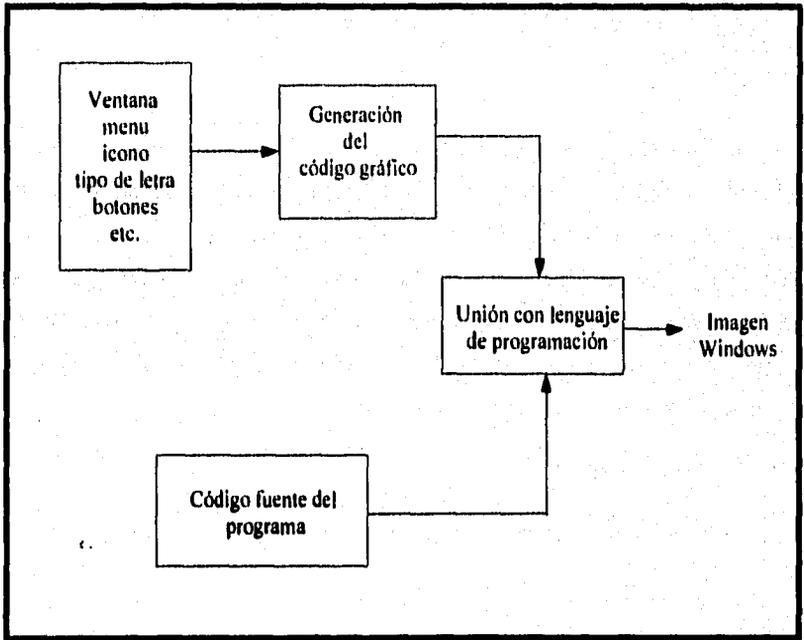


FIG. 1.3.1. Proceso para realizar una aplicación con imagen Windows.

Desde que la informática abandonó las tarjetas perforadas, la mayor preocupación de los programadores ha sido crear aplicaciones que "entrasen por los ojos"; es decir, que no sólo fueran fáciles de usar, sino también atractivos. En pocos años hemos vivido una verdadera revolución en la manera de entender las aplicaciones; esta revolución ha venido de la mano de las nuevas y poderosas PCs, de los sistemas operativos y de periféricos tales como módems, impresoras o scanners.

Además, la proliferación de las interfaces gráficas de usuario -los llamados entornos de ventanas o GUI- no ha hecho más que acentuar lo que existía entre las técnicas de programación de los años ochenta y las actuales. Muchos han sido los desarrolladores que no han sabido adaptarse a los nuevos tiempos y se han quedado "fuera de juego", sin embargo, aquellos que han



actualizado sus conocimientos reconocen que ahora cuentan con mejores y más potentes herramientas para realizar su trabajo.

Al mismo tiempo, las herramientas de programación han evolucionado hasta convertirse en verdaderos entornos de desarrollo integrados que se basan en la misma filosofía de sencillez que gobiernan las interfaces gráficas. En el campo de la integración de herramientas Borland fue pionera con su famoso Turbo Pascal, cuyas primeras versiones aparecieron para el sistema operativo CPM de la famosa familia de procesadores Z80 de Zilog. Por otra parte, Microsoft ha sido pionera con el concepto de "Programación Visual" con su popular Visual Basic, que ahora en su versión para Windows se ha convertido en un best-seller.

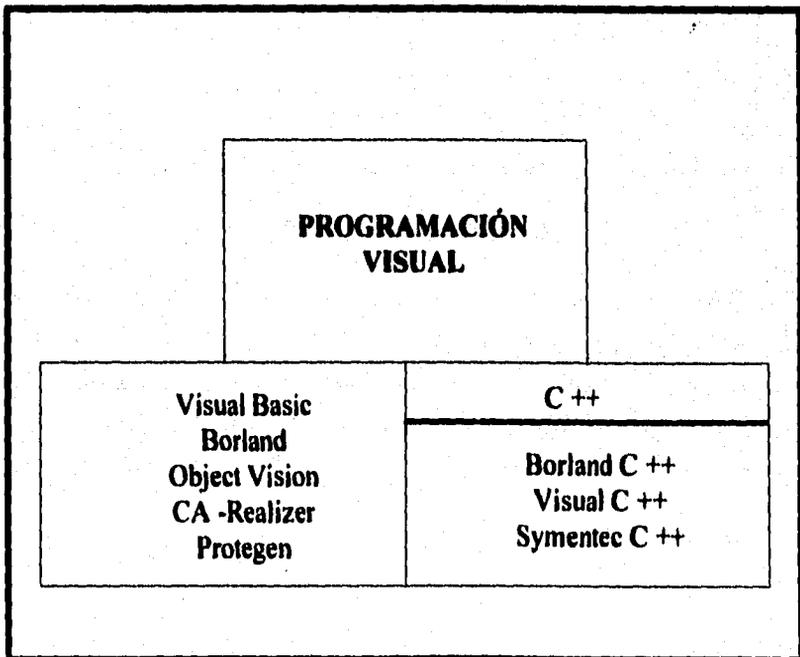


FIG. 1.3.2. Paquetes y lenguajes que manejan la programación "Visual".



Definición

Anteriormente mencionamos que el término en sí era demasiado genérico y que procedía principalmente del nombre que las propias casas de software daban a sus IDE (*Integrated Development Environment*, entornos integrados de desarrollo) para lanzarlos publicitariamente con la idea de ser algo intuitivo y fácil de usar.

Sin embargo, con la aparición de los últimos productos de desarrollo de aplicaciones bajo entornos gráficos se va haciendo necesaria una terminología adecuada que los clasifique de alguna manera.

Un entorno integrado de programación es aquel que permite desde un único programa realizar las tareas más habituales de desarrollo y depuración de una aplicación (incluido en algunos casos la optimización y realización automática de pruebas).

Si el entorno integrado es de "Programación Visual", estamos hablando de un producto que integra una serie de herramientas adicionales a las habituales y que son necesarias para el desarrollo de programas bajo una interface gráfica de usuario, todo ello con la mínima intervención del programador en las tareas más repetitivas y, por tanto, más automatizadas.

Podemos decir que la "Programación Visual" se sitúa directamente por debajo de las llamadas herramientas CASE (*Computer Aided Software Engineering*).

Los programas CASE permiten realizar el sueño (o pesadilla) de todo programador; pasar directamente del diseño de una aplicación (diagramas de flujo, estructuras de datos...) a producir la aplicación propiamente dicha. La pesadilla se puede hacer realidad para los programadores cuando llegue el día en que cualquier usuario normal pueda crear, gracias a estas herramientas, su propia aplicación, prescindiendo, por lo tanto, de los servicios de un profesional.



Programar la Interface

Con estas premisas, está claro que la "Programación Visual" ha pasado de ser una curiosa novedad metodológica, a constituirse como un conjunto de herramientas imprescindibles para el desarrollo de las complejas aplicaciones basadas en GUIs que se comercializan actualmente.

La complejidad de programar bajo una interface gráfica proviene del propio diseño de los entornos. De hecho, desarrollar el programa más sencillo para un entorno gráfico puede convertirse en una tarea desquiciante si no se dispone de las herramientas adecuadas.

Para afrontar el desarrollo de una aplicación para una interface gráfica (Windows u OS/2 en el mundo PC) podemos seguir dos caminos. El primero consiste en armarnos de paciencia y enfrentarnos con los conceptos básicos del entorno, lo cual resulta una tarea tediosa y poco gratificante porque a veces no es efectiva ante las funcionalidades ya existentes (OLE, *Object Linking Embeding*) o los nuevos sistemas (por ejemplo, *Windows 95*). El segundo camino se simplifica al utilizar las mencionadas herramientas de "Programación Visual".

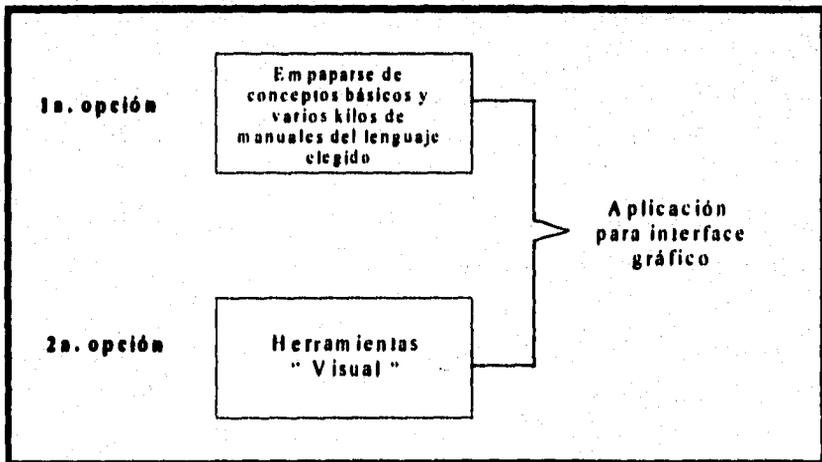


FIG. 1.3.3. Desarrollo de una aplicación para una interface gráfica.



Estos compiladores simplifican el diseño al proporcionarnos formatos, tipo o patrones que podemos aplicar a nuestras propias aplicaciones, al mismo tiempo que ofrecen una elevada confiabilidad porque se tratan de patrones probados en multitud de otros desarrollos. Todo ello redundará en una mayor eficiencia a la hora de programar y en un gran ahorro de tiempo de codificación.

Visual Basic

Visual Basic para Windows es un sistema de programación computarizado excitante. Visual Basic ha sido tremendamente aceptado, y Microsoft puso un poco de esfuerzo para integrarlo a la versión 3.0 de Windows. El poder, flexibilidad y velocidad de Visual Basic está ahora a la par con el lenguaje C únicamente, y en cuanto a productividad, Visual Basic es claramente superior.

Un programador medio necesita meses para tener una velocidad de programación para Windows usando el lenguaje C, pero con Visual Basic se pueden desarrollar aplicaciones para Windows con solo algunas horas de familiarizarse con el lenguaje tanto del paquete como de programación Basic. Se podrá estar satisfecho al ver la primera aplicación desarrollada para Windows con un trabajo fácil y tan rápido como sea posible.

Visual Basic es altamente interactivo, manejando un lenguaje de programación, con el cual permite aprender de ambas formas agradable y productivamente.

Visual Basic utiliza varias formas para sus presentaciones visuales de sistemas (En terminología de Visual Basic, una forma es una ventana asociada a controles, iconos, gráficos y código); la aplicación COLORBAR permite trazado de gráficas. La forma GETFILE permite al usuario seleccionar un archivo desde cualquier directorio en cualquier drive.

Mediante Visual Basic se puede generar un sistema que soporte la instalación y uso del ratón, para facilidad del propio sistema y del usuario; genera Cambios de Datos Dinámicos (DDE) para



aplicaciones basadas en ventanas: trabaja con las bases de datos en una interacción; y Objetos Ligados y Empotrados (OLE).

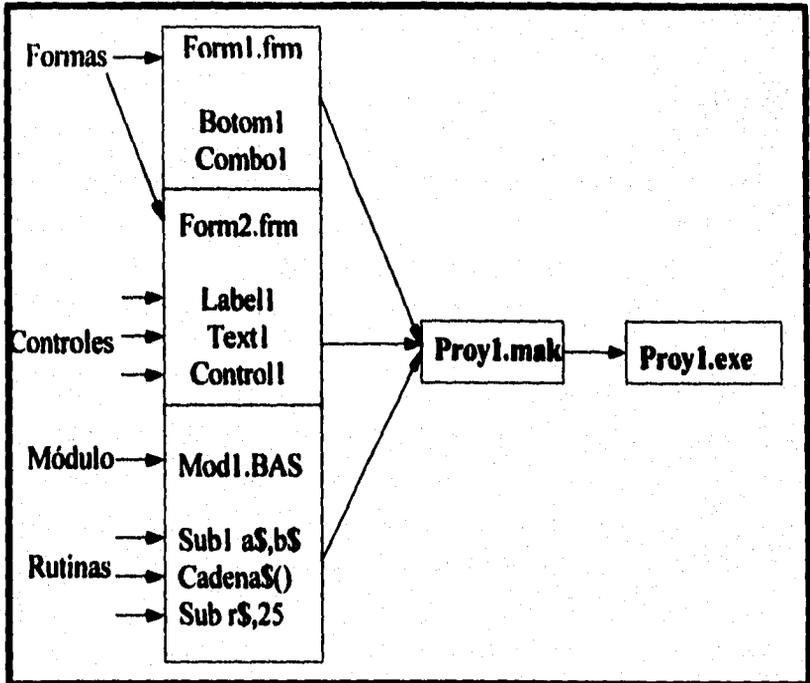


FIG. 1.3.4. Creación de una aplicación en Visual Basic V 3.0

Este lenguaje está disponible para programadores de MS-DOS y Windows. Visual Basic cuenta con una corta curva de aprendizaje, requiere de pocos conocimientos de programación y tiene una gran facilidad para una rápida integración y rehusos de componentes.

Tiene una gran flexibilidad para llevar a cabo llamadas a archivos .DLL y agregar una gran cantidad de controles al gusto del cliente.



Visual Basic constituye un camino fácil para crear interfaces de usuario utilizando componentes preconstruidas, por lo que regresamos a un lenguaje flexible que cuenta con un editor verificador de sintaxis y herramientas de *debugging* (depurado).

VISUAL	BASIC
VERSIÓN:	Visual Basic 3.0 para Windows
REQUERIMIENTOS:	IBM COMPATIBLE 386 13MB de HD Mouse Microsoft Windows 3.0 2MB de RAM (Preferible 4MB)
COMENTARIOS:	Visual Basic 3.0 proporciona una manera simple de controlar las más poderosas Bases de Datos disponibles. Además proporciona soporte para crear cajas de diálogo comunes y utiliza una comunicación más sencilla con otras aplicaciones de Windows.

FIG. 1.3.5. Ficha Técnica de Visual Basic.

Ventajas

- Es un sistema productivo para crear soluciones en Windows.
- Controles Visuales preconstruidos por terceros.
- Acceso a Bases de Datos.
- Manejo de Multimedia.



- Permite ensamblar fácil y rápidamente una interface de usuario con componentes prefabricados.
- Ofrece una gran capacidad y velocidad en su sofisticado *debugger*.
- Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excell, Project. etc..), siempre y cuando dichas aplicaciones soporten OLE automation.
- Soporta diversos tipos de Manejador de Bases de Datos.
- Interface al usuario amigable.
- Manejo de ayudas en línea robustas.
- Interprete poderoso que permite la detección de errores al momento de edición de programa.
- Aplicaciones escritas que toman ventaja de grandes arreglos y un espacio limitado en strings (cadenas).
- Maneja aplicaciones Cliente-Servidor.
- Visual Basic incluye la base de Datos Acces, la cuál provee acceso simultáneo con Foxpro, Dbase, Paradox, SQL, etc.
- Un usuario inexperto puede ser productivo con algunas semanas de utilización de Visual Basic.
- Puede ser utilizado como un servidor de OLE Automation.

Desventajas.

- Visual Basic es poco flexible es decir, se tiene que auxiliar de lenguajes más robustos como el C ó Pascal para hacer por ejemplo manipulación de memoria, accesos directos al Sistema Operativo, etc.
- No cuenta con herencia, característica muy importante dentro de la programación orientada a objetos.
- Su interprete es lento en la creación de ejecutables.
- Su código no es compatible con plataformas no Windows.



DELPHI

Delphi es una herramienta para construir aplicaciones que utiliza el lenguaje de programación *Object Pascal* de Borland - sin lugar a dudas se le pudo haber llamado Visual Pascal. Realizar una aplicación en Delphi es como crear una aplicación en Visual Basic, uno crea las formas y selecciona componentes de una caja de herramientas (*Toolbox*), para ponerlas dentro de las formas, entonces se escribe un procedimiento de eventos para manejar las entradas del usuario y varios otros eventos de Windows.

De otra forma, Delphi es una aplicación basada en componentes bajo un ambiente de desarrollo que da soporte a una implementación rápida y muy eficiente, de programas basadas en Windows con un mínimo de código. Muchos de los requerimientos tradicionales de la programación para Windows son manejados dentro de una librería de clases en Delphi, eliminando los problemas de tareas repetitivas de programación.

Delphi proporciona herramientas de diseño como son formas para aplicaciones, de tal manera que se pueden crear y probar los prototipos de la aplicación. Después, mediante el uso de una gran cantidad de componentes y la generación intuitiva de código, por lo que los prototipos de las aplicaciones se pueden modificar generando programas completos y propios del usuario.

Las herramientas de Bases de Datos de Delphi permiten el desarrollo de poderosas Bases de Datos de escritorio y aplicaciones cliente-servidor con reportes. Se puede ver la información en tiempo real de tal manera que se conocen los resultados de las consultas inmediatamente.

Ventajas

- Presenta programación estructurada y programación orientada a objetos.
- Requiere de una sintaxis precisa.
- Su compilador realiza el programa ejecutable automáticamente.



- Tiene edición multiarchivos.
- Su curva de aprendizaje presenta un nivel medio.
- Su editor de texto es amigable.
- Delphi es un lenguaje muy poderoso.
- Es un lenguaje completamente Orientado a Objetos
- Provee encapsulamiento para proyectos grandes o en grupo, lo cual significa que se tiene control del acceso a variables de una manera mucho más fácil que el esquema procedimiento/módulo/global utilizada en Visual Basic.
- Presenta un módulo separado de encapsulamiento donde las declaraciones en un archivo pueden o no formar parte de la interfaz.

Desventajas

- El lenguaje esta poco difundido
- Su límite de cadenas es de 255 caracteres, existen cadenas terminación cero que pueden ser de 65535 caracteres de extensión, pero las operaciones que se pueden realizar con ellas no son tan poderosas como las de las cadenas de Visual Basic.
- El límite de 255 caracteres también se aplica a algunas propiedades de control.
- No existen arreglos redimensionables. Sin embargo, Delphi tiene colecciones que se pueden cambiar de tamaño, que son mucho más poderosas que los arreglos.

VISUAL EN C

Hasta hace poco los únicos que podían disfrutar de este tipo de ayuda en su trabajo eran los programadores de Basic y Pascal gracias al Visual Basic de Microsoft y al Turbo Pascal para Windows de Borland.

Sin embargo, sucede que la mayoría de los programadores profesionales prefieren el lenguaje C antes que el Basic o el Pascal. Su sencillez y poca rigidez sintáctica hacen que la construcción



de compiladores sea muy sencilla, lo que facilita la portabilidad y la eficiencia de éstos, que llegan incluso a optimizar el código introducido por el programador.

Se puede decir sin temor a equivocarse que existen compiladores de C para TODAS las combinaciones posibles de microprocesadores, arquitecturas de hardware y sistemas operativos.

Si el C++ ha sido una evolución natural del C hacia la programación moderna, lo que aquí llamamos "Programación Visual" podría ser la evolución de los entornos de programación integrados para el lenguaje C que hasta hace poco no habían abandonado el modo texto.

Quede claro que no estamos hablando de una innovación en el campo del lenguaje propiamente dicho, sino de una inevitable evolución hacia formas más cómodas, y a la vez más seguras, de programar, gracias a la facilidad de uso de las interfaces gráficas.

MICROSOFT VISUAL C++

Aunque el lenguaje favorito de Microsoft es el Visual Basic para aplicaciones, son muchos los programadores que por muchos motivos prefieren el C o Pascal para sus aplicaciones. Para estos usuarios existe una herramienta muy poderoso "Microsoft Visual C++".

Hasta hace unos años, Microsoft no había adoptado la filosofía de los entornos integrados de programación en sus productos. Sus compiladores se invocaban desde la línea de comandos, teniendo que incluir multitud de parámetros para conseguir los resultados deseados. Sin embargo, su entrada en este campo ha sido espectacular.

El nacimiento de Windows supuso un paso decisivo en general hacia el desarrollo de herramientas totalmente integradas e intuitivas, pero el más beneficiado en este aspecto ha sido Microsoft. Casi nadie esperaba que alcanzaría a la mismísima Borland -pionera en ambientes bajo DOS- en cuanto a potencia y facilidad de uso en sus entornos integrados de programación.



Visual C++ supone la competencia que Microsoft opone a Borland en este lenguaje que cuenta con tantos adeptos entre los programadores. Se supone que el inventor de Windows debe disponer de los mejores lenguajes de programación para las aplicaciones de este entorno.

El lenguaje Visual C++ equivale a dos lenguajes Windows completos, en un solo paquete, los modelos de programación en Windows y los componentes de C++ trabajan juntos para realizar mejor las aplicaciones en un ambiente Windows.

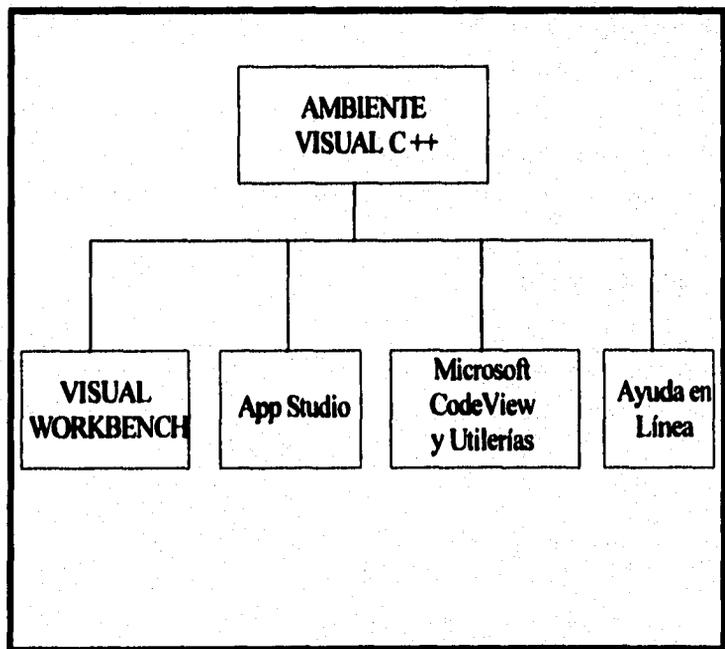


FIG. 1.3.6. Elementos del Ambiente Visual C++

Una gran diferencia entre los programas elaborados en C a los programas elaborados bajo Windows en C++, es que los programas en lenguaje C llaman al sistema de operaciones para utilizar una salida, en cambio Windows utiliza las salidas a través de un mensaje.

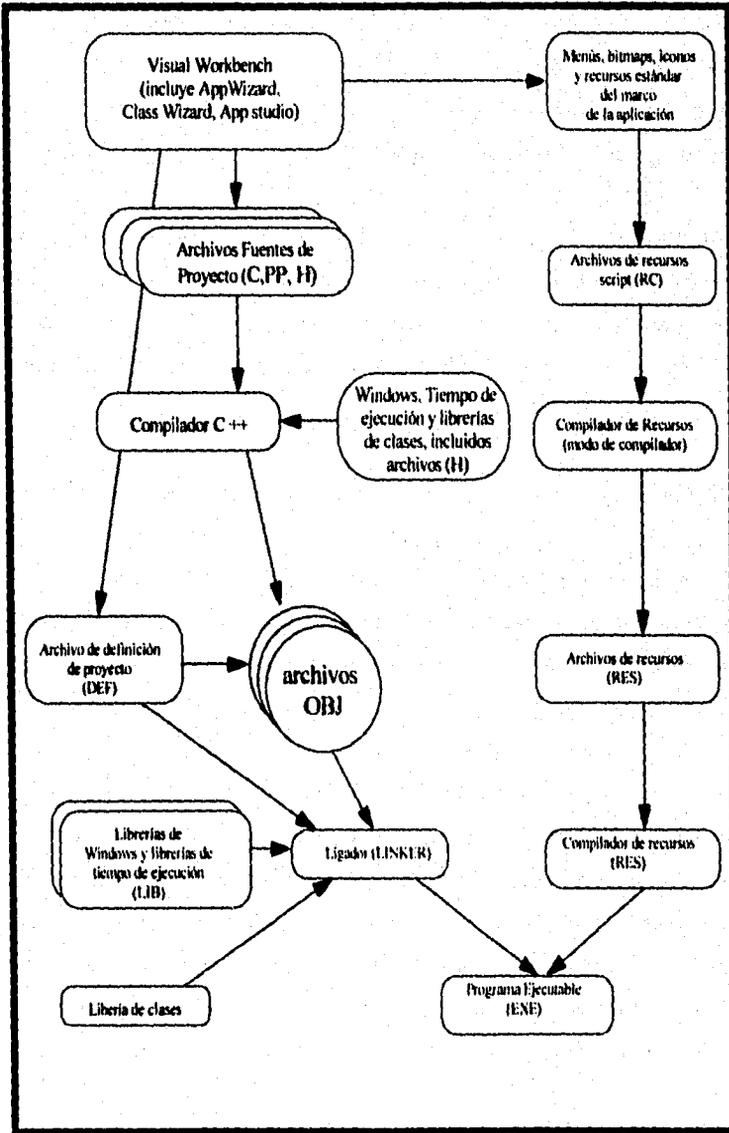


FIG. 1.3.7. Proceso de creación de una aplicación en Visual C++.



Los programas elaborados en C son escritos directamente a la memoria de video y al puerto de impresión, la desventaja de esta técnica es que necesita un "driver" de software para la tarjeta de *display* y para la impresora. Windows y C++ introdujeron un nivel de abstracción llamado interface de dispositivos gráficos, para que así el programa no necesite conocer el tipo de tarjeta de *display* y que tipo de dispositivo de impresión se tiene.

En lugar de localizar el hardware el programa llama a la función GDI la cual hace referencia a una estructura de dato llamada dispositivo de contexto, Windows convierte el dispositivo de contexto a un dispositivo físico y establece las instrucciones apropiadas de entrada/salida. El GDI es tan rápido como el acceso directo a video y permite el uso a diferentes aplicaciones mientras se escribe.

Para programar los datos en lenguaje C se tienen que codificar los datos inicializándolos como constantes, o se deben dar datos separados para que pueda leer el programa. Cuando se programa en Windows y C++, se colocan los datos en un archivo fuente dentro de un programa mediante un proceso llamado "binding". En el archivo fuente pueden estar incluidos mapas de bits, iconos, definiciones del menú. Con el editor de programas de Visual C++ se pueden editar la mayoría de los formatos.

La memoria convencional en el MS-DOS estaba limitada a 640 kbytes limitando el tamaño de los programas, se pueden utilizar técnicas para expandir memoria y así permitir programas más largos pero ha la larga va a ser insuficiente. Windows junto con C++ ofrece un manejador de memoria principal, el resultado es que la memoria ya no es un problema. Simplemente se coloca la memoria que se necesita y Windows se encarga de lo demás; ejecuta el programa, administra los recursos, automáticamente lo cambia al disco y después lo coloca en memoria física, los cambios son tan buenos que la computadora va a tener mucha memoria disponible.

En los programas comunes todos los módulos de objetos en los programas se mantienen estáticos mientras se lleva a cabo el proceso de construcción. Windows y C++ permiten una



interacción dinámica, lo cual quiere decir que librerías especialmente construidas pueden ser cargadas e intercaladas mientras se está corriendo un programa. La interacción dinámica incrementa la modularidad del programa debido a que se puede compilar y probar DLLs separadamente. Algunas de las ventajas que estas librerías tienen es que se puede crear una propia, además de utilizar tanto iteraciones dinámicas como estáticas.

MICROSOFT	VISUAL C ++
VERSIÓN:	V. 1.0
REQUERIMIENTOS:	8 MB de RAM de 50MB a 70MB en HD Windows 3.1 en modo mejorado
COMENTARIOS:	El Visual C++ requiere experiencia en la programación para Windows, pero para el programador profesional, su juego de herramientas visuales facilita la programación.

FIG. 1.3.8. Ficha Técnica de Visual C++ V. 1.0.

Ventajas.

- Incluye librerías preconstruidas.
- Controles Visuales.
- Permite la reutilización de Código total.
- Manejo completo del esquema de la programación orientada a objetos (Herencia. Polimorfismo, etc.).



- Maneja controles VBX (Visual Basic eXtension) que se refiere básicamente a componentes preconstruidas por lo que aceleran el desarrollo del software, compatibles con la Ver. 1.0. de Visual Basic.
- Todo el manejo de dispositivos de Entrada/Salida se hacen a través de librerías.
- Cuenta con las características de un Visual Workbench (integra accesos a herramientas como librerías, compiladores, editores, etc.; Control de proyectos, etc.).
- Cuenta con compilador no con interprete.
- A través de el se pueden crear clientes de OLE.
- Es el único lenguaje que al salir una nueva versión de Windows, ya cuenta con las herramientas necesarias para trabajar en él.
- Manejo de Multitarea.
- Lenguaje poderoso y flexible ya que permite crear aplicaciones núcleo.
- Rapidez en ejecución de programas.
- Manejo de variables locales y globales.
- Permite obtener funcionalidades de otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excell, Project, etc.).
- Manejo de controles OLE.
- Total libertad para manejo de memoria y accesos a Sistema Operativo.
- Permite la creación de servidores de OLE.

Desventajas.

- Curva de aprendizaje alta.
- Tiempos de compilación largos.

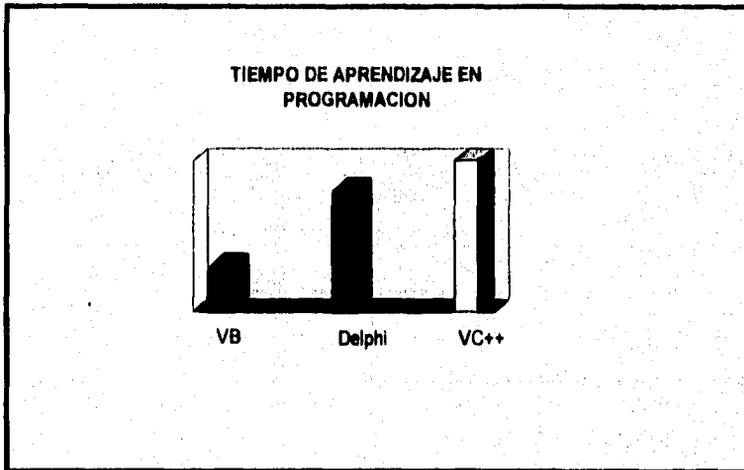


FIG. 1.3.9. Gráfica comparativa de la curva de aprendizaje del software analizado.

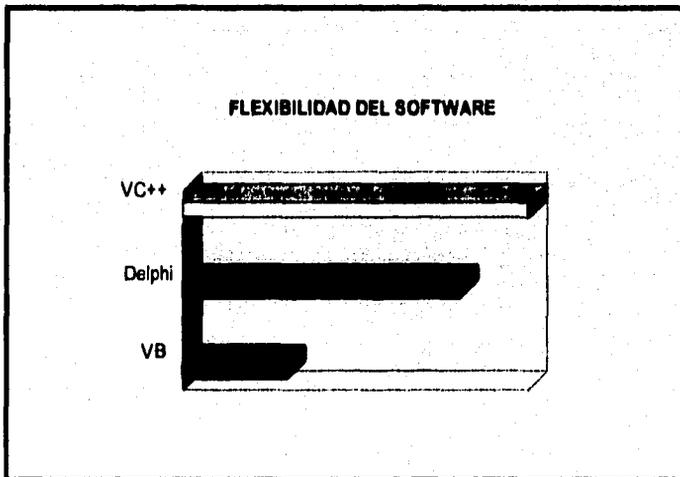


FIG. 1.3.10 Gráfica que muestra comparativamente la flexibilidad del software analizado.



Características	Visual Basic	DELPHI	Visual C++
GDI	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DDE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DLL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Componentes preconstruídos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Verificador de sintaxis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Debugging	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Acceso a B.D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amigable	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ayuda en línea	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Compatibilidad para plataformas no windows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Manejador de memoria principal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Si <input type="checkbox"/> No			

FIG. 1.3. 11. Sumario de Características del Software.



CONCLUSIÓN

En cuanto a la selección del software para la generación del *Front-End* del sistema, se utilizará VISUAL BASIC; debido a todas las características que observamos en la fig. 1.3.13 y también que se mencionaron anteriormente como son:

- Su curva de aprendizaje es baja (fig. 1.3.11.).
- Su poder, flexibilidad y velocidad son adecuados (fig. 1.3.12).
- Su productividad es alta.
- Es altamente interactivo.
- La programación es agradable y accesible.
- Proporciona una facilidad en el uso propio del sistema.
- Genera Intercambios de Datos Dinámicos (DDE).
- Trabajo con bases de datos en una interacción.
- Manejo del OLE.
- Tiene una gran capacidad y velocidad en su sofisticado *debugger*.
- Incluye la Base de Datos Acces.

No hay duda que Visual Basic es una herramienta de programación muy potente y amigable que permite desarrollar aplicaciones sofisticadas con demasiada facilidad para cualquier tipo de sistemas y en este caso como interface gráfica para acceder a la base de datos.

Lo más sobresaliente del producto es la posibilidad de incluir las especificaciones OLE dentro de nuestras propias aplicaciones. OLE se está convirtiendo en una opción casi indispensable para cualquier aplicación que necesite un intercambio de datos (como es nuestro caso) con otras aplicaciones sin perder la transparencia del característico "tomar y arrastrar" de Windows.



1.4 DISEÑO DE BASES DE DATOS (CARACTERÍSTICAS PRINCIPALES Y FUNCIONAMIENTO DE UNA BASE DE DATOS RELACIONAL).

DESARROLLO DE LA BASE DE DATOS

El desarrollo de la Base de Datos colectiva es sin duda una de las actividades más importantes en el campo de la informática en lo que va del presente decenio. Los datos se observarán con mayor interés como un recurso vital para la empresa que merece ser debidamente organizada con el fin de maximizar su valía. Aparte de las bases de datos propias de empresas y diversos organismos, día a día se va acrecentado la demanda por los servicios de información para el público en general, capacitados para el acopio, la organización y la venta de datos.

Actualmente se observa como crecen en volumen e importancia los archivos de datos que utilizan las computadoras. En todas las esferas de la vida y en todas las áreas de la industria y el comercio, las bases de datos ampliarán en gran medida las posibilidades de acción al hombre. En los próximos años, los historiadores contemplarán el surgimiento de los bancos de datos y las facilidades que ofrecen como una etapa de la evolución de la sociedad.

La industria de la computación está aumentando su capacidad para almacenar digitalmente dibujos lineales, objetos, fotografías, voces humanas, etc. De hecho, es posible en nuestros días almacenar y transmitir digitalmente cualquier tipo de información, excepto las comunicaciones más íntimas entre los seres humanos. La disminución del costo de almacenamiento de datos por bit está estrechamente vinculada con el hecho de que cada vez es mayor la cantidad de datos que se almacenan.

El adecuado aprovechamiento de las grandes cantidades de datos almacenados exige el desarrollo de otros dos recursos sistemáticos, aparte del almacén entre sí. Estos son la *transmisión de datos*, es decir, la capacidad de acceso a la Base de Datos desde localidades remotas, y el *diálogo entre el hombre y la computadora*, el que permite al usuario hacer averiguaciones, buscar en los archivos, modificar los datos almacenados o agregar otros, y resolver problemas aprovechando los

datos almacenados. Ambos temas son tan complejos en sí mismos como el diseño de la Base de Datos.

En el diseño de la Base de Datos, es de particular importancia que los datos se almacenen de manera que se les pueda utilizar indiferentemente para una amplia variedad de aplicaciones y que a la vez pueda cambiarse fácil y rápidamente la manera de usarlos. Antes, en el comienzo de la Base de Datos, en la mayoría de los sistemas de computación resultaba muy difícil modificar el modo de utilizar los datos. Los programadores ven los datos cada uno a su modo y quieren siempre modificarlos a medida que varían sus necesidades. Pero cualquier modificación es capaz de desatar una verdadera reacción en cadena de cambios en los programas existentes y resultar así inadmisiblemente difícil. En el diseño además, se deben tener en cuenta dos aspectos importantes con fines a lograr la flexibilidad de uso que es esencial en la mayoría de las aplicaciones comerciales. Primero, los datos deben ser independientes de los programas que los utilizan, de modo que se puedan enriquecer y reestructurar sin que resulte necesario modificar los programas existentes. Segundo, debe ser posible interrogar y explorar la Base de Datos sin necesidad de recurrir a la tediosa operación de escribir programas utilizando los lenguajes convencionales de programación. En lugar de éstos se utilizarán lenguajes especiales para averiguación.

El diseñar una Base de Datos es cada vez más difícil, en particular cuando se pretende alcanzar soluciones óptimas. El software es cada vez más elaborado y a menudo se comprenden mal sus aptitudes, se le emplea erróneamente o no se le aprovecha a fondo. Hay muchas maneras de estructurar los datos y cada una de ellas ofrecen ventajas y desventajas. Cada tipo de datos tiene características propias que afectan el modo de organizarlos y a menudo los usuarios tienen exigencias muy diversas. Los requisitos que se le impone al diseñador son tan variados que a menudo resulta imposible satisfacerlos con una organización única, por lo menos con el hardware actualmente disponible. El diseñador, así, no tiene más remedio que elegir cuidadosamente sus limitaciones entre tantas exigencias contrapuestas.

En la actualidad existe mucha redundancia en los datos utilizados por las organizaciones y es frecuente que el mismo ítem de datos sea definido diferentemente por los departamentos dentro



de la organización. En muchas ocasiones, solo después de instalada la computadora se advierte que eran erróneas algunas de las ideas que sirvieron de base para su diseño o cuan imprecisos eran los métodos previstos. La eliminación de toda imprecisión en la definición y el uso de los datos constituye un objetivo primordial que debe tenerse a la vista en todas las etapas del diseño y la integración de la Base de Datos. En varios casos, los métodos directos de diseño no proveen las respuestas adecuadas porque no toman en cuenta las intrincadas concertaciones a que hay que llegar entre los diversos aspectos de diseño.

En el transcurso del desarrollo de la Base de Datos se observa que, pronto tendremos nuevos dispositivos de almacenamiento, nuevas técnicas de software y nuevos tipos de bases de datos. Los detalles cambiarán, pero los principios fundamentales seguirán siendo los mismos. Por lo tanto, el lector debería concentrarse en esos principios.

SISTEMA DE BASE DE DATOS

Un sistema de procesamiento de datos tiene cuatro componentes básicamente: Los datos, hardware (máquina donde corre), software (los programas) y usuarios. En la figura 1.4.1 se muestra una representación muy simplificada de un sistema de Base de Datos.

Un sistema de Base de Datos, no es más que un sistema de mantenimiento de registros basado en computadoras, es decir, un sistema cuyo propósito general es registrar y mantener información (datos). Una definición de Base de Datos es: *Una colección de datos, integrada y generalizada, con su descripción, que se gestiona de tal forma que pueda satisfacer las distintas necesidades de sus usuarios.*

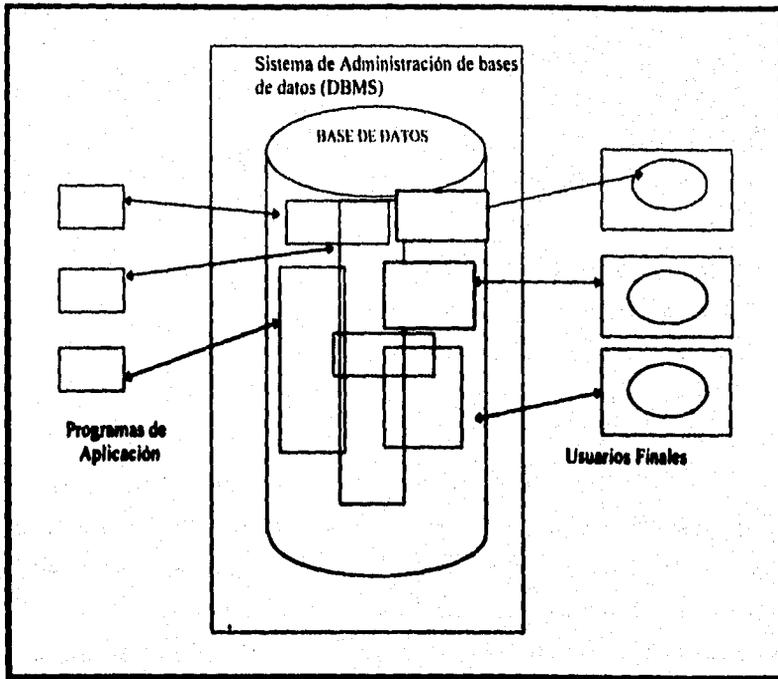


FIG. 1.4.1 Representación simplificada de un sistema de bases de datos

Una definición más elaborada y completa de Base de Datos es: *Una colección de datos imerrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Dicese que un sistema comprende una colección de bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural.*



El enfoque de la Base de Datos nos permite:

- Controlar la redundancia.
- Mantener la consistencia.
- Lograr la integración de los datos.
- Compartir los datos entre las diferentes aplicaciones.
- Cumplir con los estándares.
- Tener facilidad en el desarrollo de aplicaciones.
- Uniformar los controles de seguridad, privacidad e integridad.
- Independencia entre los datos y los programas.
- Reducir el mantenimiento a los programas.

Los sistemas de Base de Datos se diseñan para manejar grandes cantidades de información. El manejo de los datos implica tanto la definición de estructuras para el almacenamiento como la creación de mecanismos para manejar la información. Además, el sistema de Base de Datos debe cuidar la seguridad de la información almacenada en la Base de Datos, previniendo caídas del sistema o intentos de acceso no autorizado. Si se va a compartir la información entre varios usuarios, el sistema debe evitar posibles resultados anómalos. Debido a la importancia que tiene la información en casi todas las organizaciones, la Base de Datos es un recurso valioso. Esto condujo al desarrollo de un gran número de conceptos y técnicas para manejar los datos en forma eficiente.

Uno de los objetivos principales de una Base de Datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema oculta ciertos detalles relativos a la forma en que se almacenan y mantienen los datos. Esto se logra definiendo tres niveles de abstracción en los que puede considerarse la Base de Datos: físico, conceptual y de visión.

Para describir la naturaleza de una Base de Datos, se define el concepto de *modelo de datos*, que es un conjunto de herramientas conceptuales para describir los datos, las relaciones entre ellos, su semántica y sus limitantes. Se han propuesto varios modelos diferentes, los cuales se dividen en tres grupos: *lógicos basados en objetos*, *lógicos basados en registros* y los *modelos físicos de datos*.



INSTANCIAS Y ESQUEMAS

El conjunto de información almacenada en la Base de Datos en determinado momento se denomina *instancia* de la Base de Datos. El diseño general de dicha base se conoce como *esquema de la Base de Datos*. Los esquemas se alteran muy raras veces, o nunca. El concepto de un *esquema* de la Base de Datos corresponde a la idea definida en el lenguaje de programación. Una variable de un determinado tipo tiene un valor específico en un momento dado. Así, el concepto del valor de una variable en los lenguajes de programación corresponde al concepto de una instancia del esquema de la Base de Datos. Existen varios esquemas en la Base de Datos, y éstos se dividen de acuerdo con los niveles de abstracción.

INDEPENDENCIA DE LOS DATOS

La capacidad para modificar una definición de esquema en un nivel sin afectar la definición del esquema en el nivel inmediato superior se denomina *independencia de los datos*. Existen dos niveles de ésta: independencia física e independencia lógica de los datos.

Independencia física.- es la capacidad de modificar el esquema físico sin obligar a que se vuelvan a escribir los programas de aplicaciones. En algunas ocasiones son necesarias modificaciones en el nivel físico para mejorar el rendimiento.

Independencia lógica.- es la capacidad de modificar el esquema conceptual sin obligar a que se vuelvan a escribir los programas de aplicaciones. Las modificaciones en el nivel conceptual son necesarias siempre que se altere la estructura lógica de la Base de Datos.

La independencia lógica de los datos es más difícil de lograr que la independencia física, esto es debido a que los programas de aplicación dependen en alto grado de la estructura lógica de los datos a los que tienen acceso.



El concepto de *tipos abstractos de datos* es similar en varios aspectos a el concepto de independencia de los datos en los lenguajes de programación actuales. Ambos ocultan los detalles de la puesta en acción, lo cual permite a los usuarios concentrarse en la estructura general, en lugar de los detalles de la implantación en el nivel más bajo.

LENGUAJE DE DEFINICIÓN DE DATOS

Un esquema de Base de Datos se especifica por medio de una serie de definiciones que se expresa en un *lenguaje de definición de datos* (DDL). El resultado de la compilación de las proposiciones en DDL es un conjunto de tablas que se almacenan en un archivo especial llamado *diccionario de datos* que contiene *metadatos*, es decir, "datos acerca de los datos".

DATOS

Los datos del sistema pueden estar almacenados en una o más Bases de Datos, es entonces, un repositorio de datos en general, ya que esta es tanto integrada como compartida.

Por integrada se entiende que la Base de Datos puede considerarse como una unificación de varios archivos de datos independientes, donde se eliminan, parcial o totalmente cualquier redundancia entre los mismos.

Por compartida se entiende que partes individuales de la Base de Datos pueden compartirse entre varios usuarios distintos en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la Base de Datos y utilizarla con propósitos diferentes.

La palabra compartida a menudo se amplía para abarcar no solo el comportamiento antes descrito, sino también el comportamiento concurrente, es decir, la oportunidad de que diversos usuarios accedan en realidad la Base de Datos al mismo tiempo.



HARDWARE

El hardware se compone de los volúmenes de almacenamiento secundarios (discos, tambores magnéticos, etc.), donde reside la Base de Datos, junto con dispositivos asociados como las unidades de control, los canales, etc.

SOFTWARE

Entre el almacenamiento real de los datos y los usuarios del sistema existe un nivel de software, que a menudo recibe el nombre de sistema de almacenamiento de Base de Datos. Este maneja todas las solicitudes de acceso de la Base de Datos formulados por los usuarios. Una función general de un sistema manejador de Base de Datos (DBMS), por lo tanto, es proteger a los usuarios de la Base de Datos contra los detalles a nivel de hardware. En otras palabras, el DBMS ofrece una vista de la Base de Datos que está por encima del nivel de hardware y apoya las operaciones del usuario que se expresa en términos de esa vista de nivel superior.

USUARIOS

Para la definición de este tema se consideran tres clases generales de usuario:

La primera la representa el *programador de aplicaciones*, el cual es el encargado de escribir programas de aplicación que se utilicen en la Base de Datos. Estos programas de aplicación operan con los datos de todas las maneras usuales: recuperan información, crear información nueva, suprimen o cambian información existente, etc. Los programas en sí pueden ser aplicaciones convencionales de procesamiento por lotes o programas en línea diseñados para apoyar a un usuario final, que interactúa con el sistema desde una terminal en línea.

La segunda es, entonces, el *usuario final* que accesa la Base de Datos desde una terminal. Un usuario final puede emplear un lenguaje de consulta proporcionado como parte integral del sistema o recurrir a un programa de aplicación escrito por un usuario programador que acepte



órdenes desde la terminal y a su vez formule solicitudes al **DBMS** en nombre del usuario final. Este puede realizar, en general, todas las funciones de recuperación, creación, suspensión y modificación.

La tercera clase de usuario la representa el *administrador de Base de Datos* o **DBA**. Es la persona que tiene el control centralizado sobre el sistema. Las funciones del administrador de la Base de Datos son, entre otras: Definición del esquema, Definición de la estructura de almacenamiento y del método de acceso, Modificación del esquema y de la organización física, Concesión de autorización para el acceso a los datos y la especificación de las limitantes de integridad.

TIPO DE ENFOQUE DE BASE DE DATOS

Existen enfoques alternativos para visualizar y manejar datos a un nivel lógico independiente de cualquier estructura física de soporte en que se basen,

Los modelos de la base de datos que existen son:

- Modelo Jerárquico
- Modelo de Red
- Modelo Relacional

El único modelo que vamos a detallar es el último, ya que es el que nos interesa

MODELO DE UNA BASE DE DATOS RELACIONAL

La estructura de una Base de Datos relacional está basada en la representación de entidades mediante tablas, las cuales constan de columnas (campos) y renglones (registros). Las relaciones entre tablas se llevan a cabo a través de un conjunto de columnas que se tengan en común, logrando una conexión dinámica entre un número ilimitado de ellas a través del contenido de esas columnas (figura 1.4.2).



La ventaja de los sistemas relacionales es el poder modificar la información sin la preocupación de especificar las combinaciones entre registros.

COLUMNA 1	COLUMNA 2
REGLÓN 1	
REGLÓN 2	
REGLÓN 3	

FIG. 1.4.2 Enfoque relacional.

Como ejemplo podemos tener 2 entidades: *entidades cuentasaldo* y *cuentahabiente*, relacionadas entre sí con una sola relación muchos a muchos, es decir, un *cuentahabiente* puede tener varias cuentas, y una cuenta puede pertenecer a varios *cuentahabientes* (figura 1.4.3).

La tabla *cuentahabiente* consiste en tres columnas: nombre, calle y ciudad. La tabla *cuenta* tiene dos columnas: número y saldo.

A finales de los 60's cierto número de personas jugaban con la idea de construir Bases de Datos relacionales como bloques constructores básicos, pero el gran avance llegó en 1970, cuando el Dr. Edgar F.Codd de IBM propuso un modelo relacional generalizado orientado a conseguir independencia de datos; el cual fue mejorado y ampliado por el mismo poco tiempo después, y muy rápidamente llegó a ser el centro fundamental de todas las actividades de investigación sobre Bases de Datos. El modelo básico sólo especifica una estructura de datos en forma de relaciones para el esquema conceptual, y diversas versiones de lenguaje de alto nivel para manipular las relaciones.

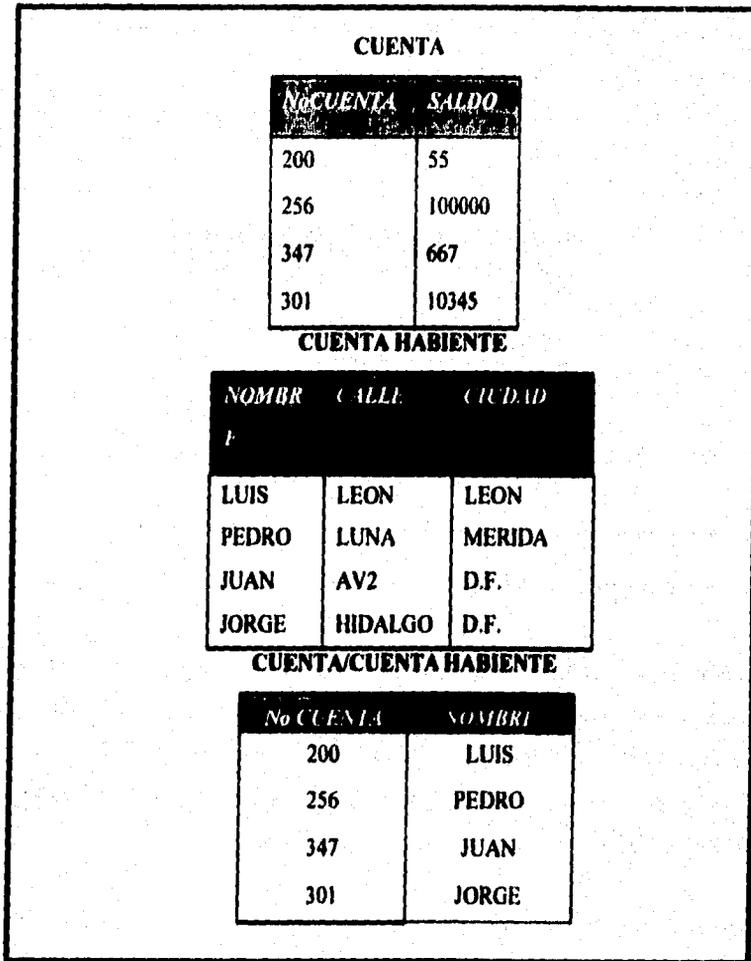


FIG. 1.4.3 Modelo Relacional

Desde una perspectiva histórica, el modelo de datos relacional es relativamente reciente. Los primeros sistemas de Bases de Datos estaban basados en el modelo jerárquico o en el de red. Estos dos primeros modelos están íntimamente ligados a la implementación física de la Base de Datos relacional. El modelo de Base de Datos relacional representa la Base de Datos como un conjunto de



tablas. Existe una correspondencia directa entre el concepto de una tabla y el concepto matemático de una relación.

Se puede definir una *Base de Datos Relacional* como la Base de Datos que el usuario percibe como un conjunto de relaciones normalizadas que varían con el tiempo, de diversos grados (es decir las extensiones de las relaciones cambian con el tiempo). En otras palabras, el término "Base de Datos relacional" significa una Base de Datos para la cual los operadores a disposición del usuario operan sobre estructuras relacionales. En resumen, una Base de Datos relacional consiste en un conjunto de tablas, que tienen asignado un nombre único. A todos los atributos se les ha dado el mismo nombre que al dominio subyacente. En términos tradicionales, una relación se asemeja a un *archivo*, una tupla a un *registro* y un atributo a un *campo*. Es decir las relaciones pueden considerarse archivos; altamente disciplinados.

Los diferentes usuarios de la misma Base de Datos recibirán diferentes conjuntos de datos y diferentes relaciones entre ellos. Es por lo tanto necesario extraer de las columnas de las tablas los subconjuntos pedidos por algunos usuarios, creando así tablas de menor grado, o por el contrario creando una de mayor grado, al fundir dos o más tablas. Estas operaciones se describen en el lenguaje de Codd. Las operaciones que se conocen como "tijera y engrudo" proporcionan un grado de flexibilidad que no es posible lograr con la mayor parte de las estructuras de árbol o *plex*. En las palabras de Codd " Tanto el programador de aplicaciones como el usuario interactivo ven la Base de Datos como una colección variable en el tiempo de relaciones normalizadas de diferentes grados".

Una relación es una tabla de datos, y puede constar sólo de una fila y una columna, proporcionando así la estructura de datos más sencilla posible que puede utilizarse como un común denominador de todas las estructuras de los datos. Esto simplifica el diseño del esquema conceptual, puesto que existe una sola estructura lógica de datos, esto es, la relación a considerar. También el modelo relacional proporciona una libertad sin precedentes al programador de aplicaciones, permitiéndole acceder directamente a cualquier valor de atributo de la Base de Datos; el mecanismo de acceso asociativo o direccionable por el contenido como atributo es aparentemente por su valor, en vez de su posición o por un puntero.



La sencillez y libertad de acceso se consigue, en primer término, expresando las relaciones en lo que Codd define como la primera forma normal, y, en segundo lugar, utilizando un lenguaje de manipulación de datos llamado álgebra relacional. Las operaciones básicas sobre relaciones, tales como las requeridas para extraer un valor de atributo o para combinar partes de dos relaciones para formar una tercera, se llevan a cabo con álgebra relacional. Codd también ha desarrollado un cálculo de predicados conocidos como cálculo relacional, que luego ha utilizado para construir un lenguaje de manipulación de datos de alto nivel llamado Sublenguaje de datos (DSL) Alfa.

Dentro del modelo de Base de Datos jerárquico y el modelo de datos de red se manejan apuntadores físicos ocultos en los registros de los datos, lo que originaba que no se asegurara la integridad de los datos, por lo tanto resultaban fácilmente inestables. Errores dentro del sistema originaban modificaciones en los apuntadores, esto ocasionaba que las relaciones entre los registros se perdieran.

Las relaciones dentro del modelo de datos relacional se representan mediante valores de datos en los registros. Un registro de departamento contiene el identificador de la división a la cual pertenece. Al elemento de datos que funciona como identificador de la división dentro de un registro de departamento es el elemento de datos denominado llave secundaria, mediante la cual es posible obtener el conjunto de registros de departamentos que se identifican en un registro de división en particular.

La esperanza en el modelo relacional fue muy grande al principio, creyendo la mayoría que era la última palabra sobre modelos de datos. Hubo dudas acerca de los méritos de los modelos relacional y de red, cada uno con sus fieles adeptos, normalmente los investigadores favoreciendo el modelo relacional. Se creyó que habían pasado los días de punteros, índices y claves poco claros. Cuando se comenzó con la implementación de los prototipos relacionales, aparecieron las primeras desventajas, representadas por aquellas claves desordenadas, punteros e índices que todavía preocupaban al DBA como en otros modelos.



El usuario del sistema de Base de Datos puede consultar estas tablas, insertar tuplas nuevas, eliminarlas y actualizarlas (modificarlas). Existen varios lenguajes para expresar estas operaciones, como el cálculo relacional de tuplas y el cálculo relacional de dominio los cuales son lenguajes sin procedimientos que representan la capacidad básica requerida de un lenguaje de consulta relacional. El álgebra relacional es un lenguaje de procedimientos que es equivalente, en capacidad, a las dos formas de cálculo relacional; define las operaciones básicas empleadas en los lenguajes de consulta relacionales. El álgebra y los cálculos relacionales son lenguajes concisos y formales que no son apropiados para usuarios casuales del sistema de Base de Datos. Es por eso que se crearon lenguajes más comerciales, para facilitar la actualización, inserción y eliminación de información.

Dentro del modelo de Base de Datos relacional, las relaciones entre los archivos son representados por valores de datos dentro de los registros a diferencia de los apuntadores físicos.

Una característica del modelo relacional es su sencillez. Como modelo de datos resulta ser menos que perfecto, es todavía un modelo que está surgiendo; sus contribuciones especialmente como iniciador de nuevas ideas y como portador de nuevas investigaciones, son incomparables; algunas de ellas son:

- Por primera vez se mostró que el modelado de datos puede basarse en un marco teórico sólido en vez de procesamientos complejos
- Dió ímpetu para investigar en toda el área de modelado de datos, incluyendo algo sobre restricciones de integridad y privacidad.
- Suministró una herramienta importante para el análisis de datos en la fórmula de la tercera y subsiguientes formas normales.
- Abrió una amplio campo de investigación sobre lenguajes no procedurables y facilidades de "vistas".



- Enfatizó la necesidad de la independencia de datos separando la descripción lógica de los datos de los conceptos relacionados con la eficiencia de acceso.
- Ha proporcionado una base para una estructura de datos y un lenguaje estándar, como se necesita en la Base de Datos distribuida para la conversión de datos y la comunicación entre modelos.

Los archivos dentro de una Base de Datos relacional presentan una longitud fija, y el formato de los registros es fijo, los que forman tablas de renglones y columnas. Cada registro en un archivo tiene un elemento de datos llamado llave primaria, que representa un identificador único del registro.

Las características principales de los "archivos" relacionales que los distinguen de los archivos tradicionales indisciplinados son:

1. - Cada "archivo" sólo contiene un tipo de registro.
2. - Cada ocurrencia de registro en un "archivo" específico tiene el mismo número de campos.
3. - Cada ocurrencia de registro tiene un identificador único.
4. - Dentro de un "archivo", las ocurrencias de registro tienen un orden inespecificado o se ordenan según valores contenidos dentro de esas ocurrencias.

MODELO ENTIDAD-RELACIÓN

El modelo entidad-relación es una técnica para definir las necesidades de información de cualquier empresa. Esta técnica involucra conceptos que se identifican con varios objetos de importancia para la empresa, a los cuales se les denomina *ENTIDADES*, a las características de dichos objetos se les denomina *ATRIBUTOS* y a como se relacionan estos objetos entre sí se les denomina *RELACIONES*.



Todos estos conceptos se modelan a través de cierto tipo de esquemas gráficos, los cuales muestran a los usuarios una manera más sencilla y práctica de visualizar sus necesidades de información.

Una entidad es una persona, cosa o lugar, que cae dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información. La entidad se representa por medio de una caja con las esquinas redondeadas y dentro de ésta se escribe el nombre de la entidad, el cual debe estar en singular. Cada entidad debe tener un nombre único dentro del sistema, lo que indica que no pueden existir dos entidades dentro del mismo sistema con el mismo nombre.

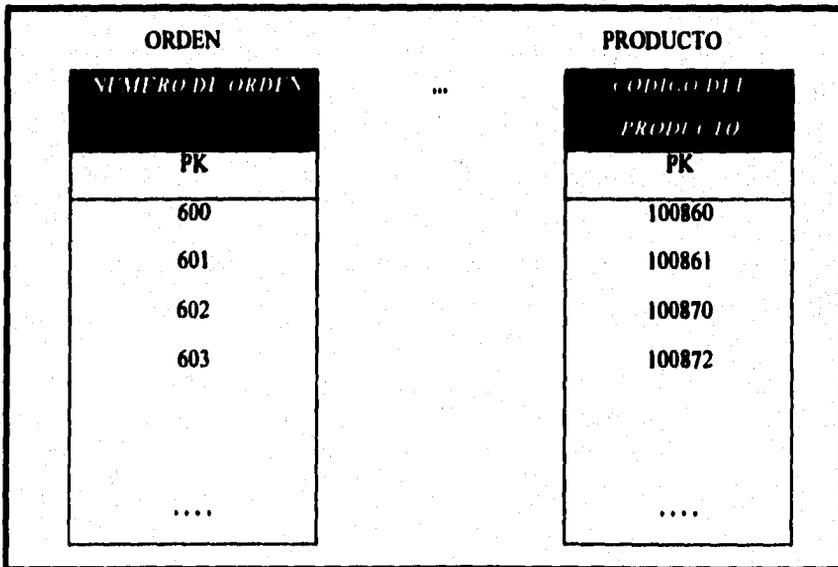


FIG. 1.4.5 Entidad

La entidad dentro del modelo relacional se representa por medio de una tabla, donde el nombre de la tabla debe contener una columna que identifique de forma única a cada renglón de ésta. Esta columna recibe el nombre de llave primaria (PK), la cual no puede contener valores nulos, ni duplicados. Ejemplo figura 1.4.5.



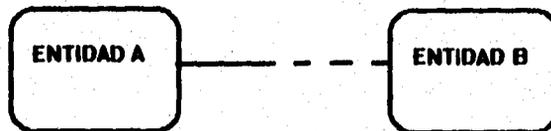
Una relación requiere de una o más entidades, la cual debe caer dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información

Una relación es un término matemático para representar una tabla bidimensional, como una colección de registros. Se define mediante filas y columnas, siendo cada entrada o elemento un valor de atributo. Se llama esto una relación en vez de una matriz porque falta homogeneidad en sus entradas.

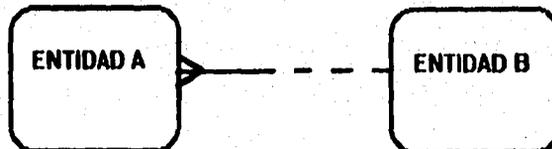
Cada columna de una relación contiene los valores de un atributo con nombre, y cada fila se llama *tupla*. La palabra tupla se toma de la definición de grupos tales como quintuplas, séxtuplas, etc. Así pues, un grupo de n elementos es una n -tupla. La extensión de una relación es el conjunto de las tuplas de la relación.

Las relaciones se presentan en varias modalidades:

Relaciones Uno a Uno con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.

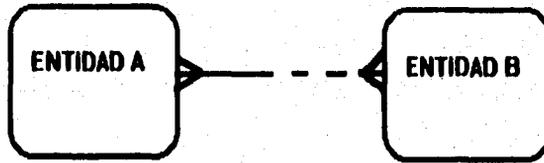


Relaciones Uno a Muchos con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.





Relaciones Muchos a Muchos con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.



Las *RELACIONES* dentro del modelo relacional se presentan de la siguiente forma:

Relación Uno a Uno entre dos entidades se modela dibujando la llave primaria (PK) de la tabla en más renglones como llave foránea (FK) en la otra tabla figura 1.4.6.

PRODUCTO		PRECIO	
CÓDIGO DE PRODUCTO	PK	CLAVE DE PRECIO	NUMERO DE PRODUCTO
	PK	PK	PK
100860		1	100870
100861		2	100876
100870		3	100101
100872		4	100860
....		

Figura 1.4.6 Relación Uno a Uno



Una llave foránea es una o más columnas que son llave primaria en otra tabla, una llave foránea no permite nulos y valores duplicados.

Relación Uno a Muchos o Muchos a Uno entre dos entidades se modela dibujando la llave primaria de la tabla que tiene la correspondencia de uno como llave foránea en la otra tabla figura 1.4.7.

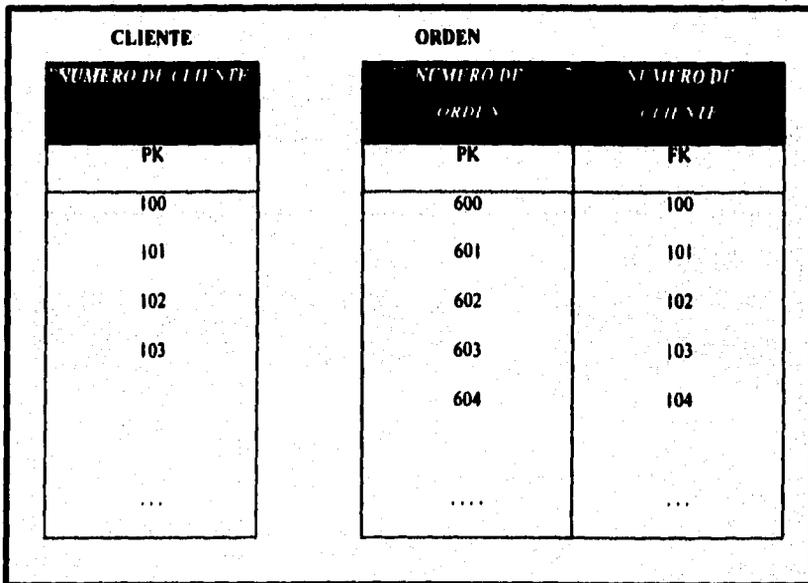


Figura 1.4.7 Relación Uno a Muchos o Muchos a Uno

Relación Muchos a Muchos entre dos entidades se modelan dibujando una tercer tabla, la cual se compone de una llave primaria compuesta de dos columnas, las cuales son llaves primarias y foráneas a la vez Figura 1.4.8.

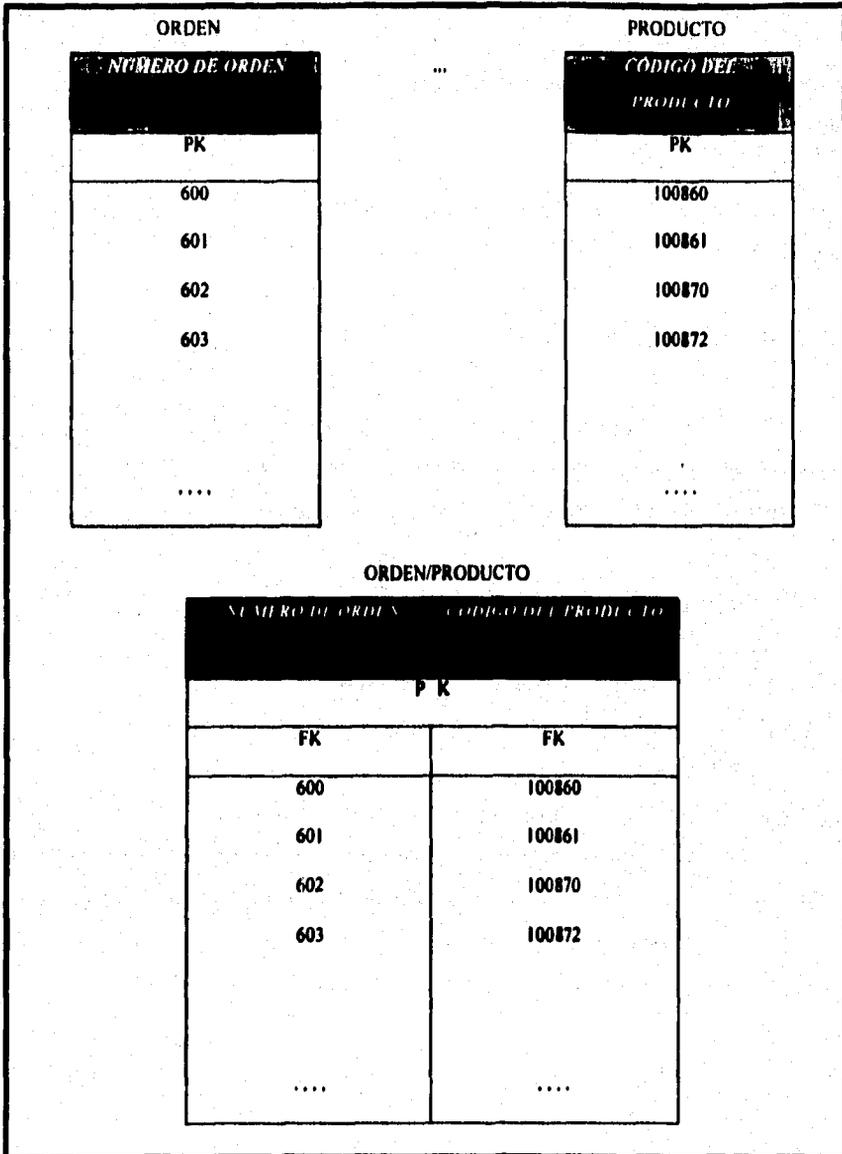


FIGURA 1.4.8 Relación Muchos a Muchos

Un atributo es una característica o cualidad de una entidad o relación, que cae dentro del alcance del sistema. acerca del cual el sistema debe mantener, correlacionar y desplegar información. Para representar uno o varios atributos, se escribe el nombre del atributo dentro de la entidad.

Los atributos dentro del modelo relacional se presentan por medio de columnas dentro de una tabla

PROCESO DE NORMALIZACIÓN

El enfoque relacional posee bases matemáticas rigurosas que respaldan su teoría relacional, proporcionando simplicidad en las estructuras de datos utilizadas, facilitando su uso y modificaciones. Para poder obtener estas facilidades, el proceso de normalización es la clave.

Los objetivos del proceso de normalización son:

- Eliminar en lo posible todos los datos que mantengan anomalías.
- Conservar toda la información.
- Maximizar la flexibilidad.
- La estructura debe ser tal que haya lugar para todos los datos requeridos.
- La redundancia que puede existir deberá ser usada por los elementos que son identificadores o llaves. Por lo que se debe tener cuidado de elegir aquellos que no estén sujetos a actualizaciones.
- Los defectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.

Anomalía de Inserción: No se debe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.

Anomalía de eliminación: La eliminación de un solo registro puede ocasionar la eliminación de toda una ocurrencia de una entidad.



Anomalía de actualización: si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentre definido.

DEPENDENCIA FUNCIONAL

El poder definir si una relación se encuentra en algunas de las formas normales, se recurre a las dependencias funcionales que existan entre los atributos y los dominios particulares a esa relación; las dependencias funcionales las determina directamente el significado de la semántica del contenido de la base de datos según la interpretación del diseñador de la base de datos.

El modelo relacional se basa en este concepto para establecer sus relaciones funcionales entre atributos. De esta forma, una definición formal de DEPENDENCIA FUNCIONAL en el modelo relacional sería la siguiente:

Dada la relación R se dice que el atributo Y de R es funcionalmente dependiente del atributo X de R si y sólo si cada valor de X en R tiene asociado a él exactamente un valor de Y en R (en cualquier instante). El indicar que Y es funcionalmente dependiente de X es equivalente a indicar que X identifica o determina a Y, lo cual se denota como $X \rightarrow Y$. Esto último concuerda con la lógica matemática en la que $X \rightarrow Y$ significa que X identifica a Y, es decir, que si X tiene un cierto valor "x", entonces Y debe tener un valor "b".

PRIMERA FORMA NORMAL

Uno de los objetivos del enfoque relacional es representar las bases de datos mediante relaciones planas o tablas. Por lo tanto cada identificador en una relación debe poseer un solo valor de cada uno de los atributos y no múltiples valores de estos, si posee múltiples valores se dice que existen grupos repetitivos.

Por ejemplo consideremos la relación INF_TUR que posee la siguiente estructura y se muestra en la tabla 1.4.1.



En esta relación se puede observar que existen múltiples valores para el atributo CVE_GIRO que es el identificador, por lo tanto no es una relación plana.

Se dice que una relación está en **Primera Forma Normal** si para cada valor específico de un identificador existe uno y sólo un valor de cada atributo. Es decir no hay "grupos repetitivos". Pero se mantiene un alto grado de redundancia.

Para transformar la relación de la tabla 1.4.1 a la primera forma normal aplicamos la conversión de la 1FN.

CONVERSIÓN DE LA PRIMERA FORMA NORMAL (1FN):

La redundancia puede reducirse separándola en otro grupo aparte de la entidad, debiendo contener el identificador de la entidad original para mantenerse relacionadas.

Por lo tanto la relación INF_TUR normalizada en la 1FN quedará de la siguiente manera:

Separando la redundancia en otro grupo aparte, obtendremos un catálogo denominado GIROS (tabla 1.4.2).



INF_TUR

CVE_CBO	CBO	TP	NUM_	NUM_CBO	CALLE	NUM	COLUMA	CP.	TEL.	R.F.C.	RASPN_SBT	CATEDO	PAI
54	HOTEL	5	130	HOTEL BORLAND PUNTO WIND	PASEO ETAPA	500	ETAPA	8000	2200	0000000000	GRUP HOTELER BORLAND	NESTRELLAS	GER COM
55	HOTEL	5	20	ETAPA PALAC DEPORT	PASEO DE LAS CARREAS	500	DEFINENCIA L.	8000	2100	0000110007	GRUP PUNTO S.A DE C.V.	NESTRELLAS	GER COM
56	HOTEL	5	120	GRUP HOTEL	PASEO DE LA FALDNAR	500	ZONA HOTEL	8000	2000	0000000000	PUNTO S.A DE C.V.	NESTRELLAS	GER COM
19	RESTAURANT ES	5	000	PELZEMA MARETA MARETA	PASEO ETAPA	1000	ETAPA	8000	2074	0000000000	GRUP HOTELER MARETA	NET	NET
10	RESTAURANT ES	5	00	LAS MARIANAS	PASEO ETAPA	500	CENTRO	8000	2000	0000000000	ALMA FRANCO	NET	NET
15	RESTAURANT ES	5	00	GRUP HOTEL	PASEO ETAPA	500	ETAPA	8000	2100	0000000000	GRUP HOTELER S.A.	NET	NET
73	DISCOTECA S	5	20	GRUP HOTEL	PASEO ETAPA	500	ETAPA	8000	2000	0000000000	GRUP HOTELER S.A.	NET	NET
87	Tienda	0	700	ACA DE	PASEO	500	ETAPA	8000	2000	0000000000	ACA DE S.A.	NET	NET

TABLA 1.4.1



GIROS

CVE_GIRO	GIRO	TIPO_GIRO	CATEGORIA	PRIN_SER
56	HOTEL	S	5	SERV.
56	HOTEL	S	5	SERV.
56	HOTEL	S	5	SERV.
15	RESTAURANTE	S	N/C	N/C
15	RESTAURANTE	S	N/C	N/C
15	RESTAURANTE	S	N/C	N/C
73	DISCOTEQUES	S	N/C	N/C
87	TIENDA DE	B	N/C	N/C

TABLA 1.4.2

El identificador de la relación GIROS es CVE_GIRO.

La relación de la tabla 1.4.1 quedará tal y como se muestra en la tabla 1.4.3 con la conversión de la 1FN.



conversión de la IFN.

CVE-GRUPO	NOM.	NOM_COM	CALLE	MUN	COL.	C.P.	TEL.	R.F.C.	RAZON_SOC
88	12H	HOTEL BORRERO PASCOCO	PASEO ETAPA	SN	ETAPA	4899	3283	ONG888888	OPER HOTELER BORRERO
86	8H	ETAPA PALAC RESORT	PASEO DE LAS GARDIAS	SN	RECIDENC ML	4899	3139	APT81118H17	ABE PROS TUR SA DE CV.
84	23H	ORIN HOTEL	PASEO BEL PALCOBR	SN	ZONA HOTELER	4899	3883	P88785288LJ	PROP ETA SUR SA DE CV
15	16R	FIZONA MARRA HORN	PASEO ETAPA	LOC 6	ETAPA	4899	3874	DNE48922D72	DELMETO ENRIQUE
15	4R	LAS SERRANITAS	PASEO ETAPA	SN	CENTRO	4899	3816	AGC4898533	ALLIA GIZLEZ FCO
15	6R	SA-SAFFONE	PASEO ETAPA	SN	ETAPA	4899	3132	RD8858888C	REST SA-SAFFONE SA
73	3D	MAGIC CIRCUS	PASEO ETAPA	SN	ETAPA	4899	3189	ONG888888LJA	OPRETAPA MAG SA. DE CV
87	7TR	ACA JOE	PASEO ETAPA	SN	ETAPA	4899	3882	AB88878LJA	ACA ROPA SA

TABLA1.4.3



SEGUNDA FORMA NORMAL

La prueba para determinar si una entidad está en Segunda Forma Normal es: que el valor de cualquier atributo que no es llave dependa de todos los atributos que forman la llave, y que además cumpla con la 1FN.

CONVERSIÓN DE LA SEGUNDA FORMA NORMAL (2FN):

Para normalizar una entidad en la Segunda Forma Normal, se crea una nueva entidad de los atributos que dependen parcialmente de una llave, siendo parte del identificador de esta nueva entidad el atributo del cual depende para mantenerla relacionada con la original.

Siguiendo con el ejemplo anterior y aplicando la Segunda Forma Normal, ahora obtendremos un segundo catálogo denominado FOLIOS como se muestra en la tabla 1.4.4 la cual contiene todos los atributos que dependen parcialmente de una llave.



CVE_GIRO	NOM_COB	CALLE	NUMERO	COLONIA	C.P.	TELEFO	R.F.C	RAZON_SOC
12H	MOTEL PACIFICO	DORADO PASO ETAPA	5H	ETAPA	49000	33025	OND990405U90	OPER MOTELERA DORADO
5H	ETAPA RESORT	PALAC PASO DE LAS GARZAS	5H	RECREACIONAL III	49000	31300	APT951116R17	ASE PROM TUR S.A DE C.V.
23H	OSHI HOTEL	PASO DEL PALOMAR	5H	ZONA MOTELERA	49000	30003	PS9706306KJ	PROP ETA SUR S.A DE C.V
18R	PIZZERIA MORRIS	PASO ETAPA	LOC 8	ETAPA	49000	30374	ONE4302230Y2	REILDOR NIETO ENRIQUE
4R	LAS MARGARITAS	PASO ETAPA	5H	CENTRO	49000	30010	AQ940905K38	ALLIA GONZALEZ FRANCISCO
6R	DA-BAFFONE	PASO ETAPA	5H	ETAPA	49000	31122	RD9403050K5	RESTAURANTE DA-BAFFONTE
3D	MAGIC CIRCUS	PASO ETAPA	5H	ETAPA	49000	31000	OD990603L8A	OPETAPA MAG S.A. DE C.V
77R	ACA JOE	PASO ETAPA	5H	ETAPA	49000	30302	ARB916717L9A	ACA ROPA S.A DE

TABLA I.4.4



En la tabla 1.4.5 se muestra como queda al final ya normalizada (con las dos primeras normas) la relación INF_TUR.

INF_TUR

CVE GIRO	NUM FOLIO
56	12H
56	5H
56	23H
15	10R
15	4R
15	8R
73	3D
87	7TR

TABLA 1.4.5

TERCERA FORMA NORMAL.

Se dice que una entidad se encuentra en Tercera Forma Normal si el valor de cada atributo depende de toda la llave y no de cualquier otro que no lo sea.

En esta forma normal, se buscan los atributos que están dependiendo de otro que no es una llave.

CONVERSIÓN DE LA TERCERA FORMA NORMAL.

Para una entidad en Tercera Forma Normal, se crea una entidad con los atributos que no dependan de ningún atributo que forma la llave, siendo el identificador de la nueva entidad el atributo del cual era independiente. Es decir: Una relación R háyase en la tercera forma normal si se encuentra en la segunda forma normal y cada uno de los atributos no primos son dependientes no transitivos de cada llave candidata de R .

En el ejemplo, se observa que no existe este tipo de dependencia, por lo que cumple también con la Tercera Forma Normal.

1.5 SISTEMAS MANEJADORES DE BASE DE DATOS PARA PC'S Y ELECCIÓN DE ALGUNA (CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS)

El objetivo de un sistema de base de datos es simplificar y facilitar el acceso a los datos. Las vistas de alto nivel ayudan a lograrlo. Si el tiempo de respuesta para una consulta es demasiado largo, el valor del sistema se reduce. El funcionamiento del sistema depende de la eficiencia de las estructuras de datos utilizados para representar los datos de la base de datos y de qué tan eficientemente pueda operar el sistema con esas estructuras. Como sucede en muchos otros aspectos de los sistemas de cómputo, deben hacerse concesiones, no sólo entre el espacio y el tiempo, sino también entre la eficiencia de un tipo de operación y la de otro.

La aparición de la computadora personal en el mercado, y su rápida evolución tecnológica, así como el abatimiento del costo, ha propiciado su uso en gran escala. En consecuencia ha surgido una gran cantidad de software de Base de Datos.

Por la facilidad de implantación del modelo relacional, el desarrollo de los manejadores de Base de Datos se ha orientado a éste. El poder de la facilidad de uso son los aspectos fundamentales para los usuarios de bases de datos relacionales.

El manejador de base de datos es un módulo de programas que constituye la interface entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicaciones y las consultas hechas al sistema. El manejador de base de datos es responsable de las siguientes tareas:

- ◆ **Interacción con el manejador de archivos.** Los datos sin procesar se almacenan en el disco mediante el sistema de archivo proporcionado normalmente por un sistema operativo convencional. El manejador de base de datos traduce las diferentes proposiciones en DML, a comandos de sistema de archivos de bajo nivel. Así, el manejador de base de datos se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base de datos.



- **Implantación de la integridad.** Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de limitantes de consistencia. El administrador de la base de datos debe especificar estas limitantes en forma explícita. Si se especifican estas limitantes, entonces el manejador de la base de datos puede verificar si las actualizaciones a la base de datos resultan en la violación de cualquiera de estas limitantes, y si es así, se podrá realizar la acción apropiada.
- **Puesta en práctica de la seguridad.** Es preciso que todos los usuarios de la base de datos tengan acceso a todo su contenido. Es labor del manejador de la base de datos hacer que se cumplan estos requisitos de seguridad.
- **Respaldo y recuperación.** Un sistema de cómputo, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallas. Es responsabilidad del manejador de la base de datos detectar estas fallas y restaurar la base de datos al estado que existía antes de presentarse la falla.
- **Control de concurrencia.** Cuando varios usuarios actualizan la base de datos en forma concurrente, es posible que no se conserve la consistencia de los datos. Es necesario que el sistema controle la interacción entre los usuarios concurrentes; lograr dicho control es una de las tareas del manejador de la base de datos.

Los sistemas manejadores de base de datos relacionales (RDBMS) han sido altamente aceptados por la forma en que se manejan los datos, los sistemas relacionales ofrecen los siguientes beneficios:

- Permiten un sencillo acceso a los datos .
- Flexibilidad en el modelado de los datos.
- Disminuye la redundancia de datos y el nivel de almacenaje.
- Existe independencia entre el almacenaje físico y el diseño lógico de datos.
- Nivel alto en la manipulación de datos.

CLIPPER

Este paquete para el desarrollo de bases de datos es, definitivamente para los programadores. Si bien carece de algunas de las excelentes funciones de generación de código que tienen los demás paquetes, Clipper ofrece una riqueza de armas y capacidad que los programadores necesitan.

Dos de sus armas son un generador de reportes (RL) y una función (DBU) para la creación y manejo de los archivos de la base de datos, escritos en el propio lenguaje de programación Clipper, y se incluye el código fuente, el cual puede utilizarse como referencia, o modificarlo para añadirlo a las aplicaciones de la base de datos.

La pantalla de DBU enlista las opciones a través de su parte superior, junto con las teclas de funciones asignadas. El resto de la pantalla está dedicado a una representación visual del panorama de la base de datos activa, la cual consiste en una columna partida en tres secciones. El primer grupo exhibe el nombre de la base de datos activa. Si al invocar el programa DBU se invoca un argumento, aparecerá la base de datos o la vista especificada. En la siguiente sección aparecerán los índices activos relacionados con la base de datos activa, y el grupo al fondo muestra los nombres de los campos para la base de datos activa.

El generador de reportes, RL, no es tan flexible, no soporta un bosquejo WYSIWYG del reporte y tampoco permite ver la salida en forma preliminar conforme se vaya trabajando. La verdadera potencia de Clipper se basa en su codificación.

El compilador del programa es muy rápido además de tener algunas funciones muy interesantes, dentro de estas se destacan las siguientes: soporta los llamados bloques de código, pedacitos de código ejecutable que se pueden almacenar como variables, o pasar a otros programas como argumentos, para ejecutar un bloque de código, se utiliza una función EVAL(); otra función útil es el uso hecho por Clipper de los archivos cerrados para el seguimiento, a fin de

determinar cuales archivos en un programa dependen de otros para operar correctamente y quedar al corriente.

Estando instalado este sistema, se puede invocar la función MiMake para llevar a cabo aquellas operaciones de compilación y enlace que se necesiten y mantener todos los archivos sincronizados. Clipper también soporta las funciones para leer los archivos binarios de DOS y escribir en ellos. También se aprecia el depurador de Clipper, el cual permite analizar el funcionamiento de código del programa, ejecutar comandos y revisar el estado de algunas variables, nombres de campos o expresiones en particular. Una ventanilla de estado enlista las bases de datos abiertas en todas las áreas del trabajo activas, así como los valores de todos los comandos Set.

FOXPRO

La interface de FoxPro incluye menús presentados junto con una ventanilla de comandos conveniente, para utilizar los menús o escribir los comandos sin que ninguno de los elementos de la interface interfieran con otro. Su soporte al ratón es el mejor de todos los paquetes basados en caracteres. Así mismo, la ventanilla de comandos mantienen una historia corrida de las instrucciones (al igual que Dbase III plus), esto facilita la repetición de los comandos utilizados a través de una sesión. También se puede seleccionar parte de la historia de los comandos, y anexarla a sus aplicaciones.

La ventana presenta gráficamente todas las áreas de trabajo disponible; se puede seleccionar un área de trabajo disponible, y abrir en ella una base de datos, accionando un botón de comando. FoxPro ofrece una caja de diálogo con una lista de los campos principales.

El constructor de pantalla comienza como pantalla en blanco, en la cual se puede capturar el texto y colocar los campos, también se pueden crear botones de comando, casillas, marcar con "palomas", botones de "radio" y lista de extraer. Se pueden agregar pedacitos de código a cualquier objeto, incluso a los campos.

Desde la pantalla de establecimiento, se puede ejecutar código antes y después del programa de la pantalla de captura, cuando se genere el código para la pantalla diseñada, se pueden anexas otras pantallas a ella, lo cual ahorra tiempo una vez formulada una biblioteca de pantallas genéricas.

El Funcionamiento de FoxPro es magnífico gracias, en parte a su tecnología exclusiva de *Rushmore*, siendo su único inconveniente su voluminosa documentación, ya que es un problema localizar la información rápidamente, lo cual no obstante disminuye gracias a su fuerte función de ayuda en línea.

INFORMIX-SQL

Informix SQL es cien por ciento una base de datos con lenguaje de consulta estructurado (SQL), el aspecto del programa es muy austero: no se encontrará con pantallas de colores múltiples con menús y ventanillas a la vista. Las pantallas del paquete, estilo Lotus, impulsadas por menús, automatizan las operaciones más significativas de la base de datos, tales como la creación de tablas, la definición y modificaciones de los campos, etc. Informix también incluye un generador de reportes, y un sistema para la ejecución de archivos de la definición de formas diseñadas.

En Informix se construye una forma, no moviendo un cursor sobre la pantalla con un ratón o teclas de fecha, sino escribiendo en la forma. un tipo de definición de pantalla acompañada con instrucciones ejecutables.

Dicho archivo comprende cinco partes: una sección de tablas serán accedidas por la forma; una tabla de atributos, la cual describe cada campo exhibido por la forma, y una sección opcional de instrucciones, que define las operaciones que habrán de ser llevadas a cabo sobre los campos dentro de la forma.



Las secciones y tablas informan al sistema que debe presentar; la sección de pantalla indica en donde presentarlo, la sección de atribuciones indica como presentarlo y la sección de instrucciones le dice al sistema que hace antes, mientras y después de presentarlo.

Se incluye con Infromix-SQL varios paquetes de servicios: BECHECK verifica la integridad de los índices, si encuentra una discrepancia entre un archivo de datos y uno de sus índices, le permite reformar el índice; DRLINK y DBLOAD son de utilidad para el traslado de los datos entre Infromix y el mundo exterior de archivos de Lotus 1-2-3, Dbase, o ASCII; con DBSCHEMA, se pueden elaborar las instrucciones de SQL requeridas para crear una tabla o una base de datos.

PARADOX

Una de las características más interesantes de PARADOX es su velocidad, destacándose su rapidísimo tiempo de respuesta para la lectura y edición de las tablas.

Paradox soporta los formatos de importación y exportación más importantes, pero existen problemas en la importación almacenada en el formato ASCII. Con Paradox, es necesario planear con cuidado antes de precipitarse a la construcción de una forma, aplicando esto principalmente con las formas que accesan a múltiples tablas. El diseño de formas con tablas múltiples exige la creación de una forma maestra, después, existe también la introducción, en dicha forma maestra de las formas incluidas en las otras tablas. Así que para crear la forma maestra completa, deben diseñarse las formas que se incluirán primero.

El lenguaje para el manejo de la base de datos PARADOX es PAL, aunque es erróneo representar a PAL como simplemente un DML. Lo que dificulta la programación con PAL es que maneja la transmisión de comandos como si el programador fuera un robot sentado en un teclado operando PARADOX; sin embargo, si simplemente no se quiere tener nada que ver con PAL, podrá instalarse el programador personal. Este programa es, esencialmente un constructor



de aplicaciones que a través de la creación de una aplicación completa con menús y formas, ofreciendo el código PAL como su producto final.

ACCESS

Access ha llegado a ser uno de los más populares manejadores de bases de datos relacional lanzados en el medio informático, debido a todas las posibilidades que integra. Es el producto más escalable en su clase, implanta un impresionante grupo de características nuevas y mejoras a la facilidad de uso, incluyendo varios *Wizards* nuevos.

Tiene el mejor rendimiento de Búsqueda, impone reglas de integridad de datos a nivel de base de datos que le permiten determinar como los registros en tablas asociadas se ven afectados cuando se modifica una matriz, reduciendo así la cantidad de código necesario para construir aplicaciones.

Cuando se crea una base de datos en Access todos sus componentes se guardan en un archivo único .MDB. Pulsando sobre el botón New se comienza el proceso de crear un objeto del tipo seleccionado y es ahí donde los *Wizard* entran en acción, facilitando la creación de diseños funcionales. De los asistentes más potentes está el *Table Wizard*, que provee más de cuarenta modelos de tablas personales y de negocios, cada una con sus propios juegos de campos con formato para usar en su propia base de datos.

Los *Wizard* de botones de comandos frecuentemente determinan de antemano las respuestas de los eventos del ratón. Contiene características que permiten generar rápidamente reportes y formularios. Posee apoyo para OLE 2.0, que se puede incluir en formularios y reportes, así como para BOLB's.

Los *Wizard* son módulos orientados a ciertas tareas, que hacen una serie de preguntas y luego generan los objetos apropiados, según las respuestas. También posee un lenguaje de programación *ObjetPAL*.



Tiene una ventana gráfica de *Relationship* que muestra todas las tablas en la base de datos con líneas entre ellas para representar las relaciones activas. Simplemente se arrastra de un campo en una tabla a otro en otra tabla para definir una relación y luego se personaliza el enlace.

Puede usar datos de varias bases de datos como: dBase III y IV , Foxpro 2.0 y 2.5, Paradox 3.x y 4.x así como Microsoft, Oracle, Sybase y el lenguaje SQL.

Access es una aplicación que se desarrolló bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Access nos ayuda a gestionar nuestra base de datos proporcionando una infraestructura eficiente para almacenar y recuperar la información, además genera automáticamente un formulario para complementar todos nuestros datos. Después de que hayamos introducido algunos datos, podemos pedir a Access que visualice una parte seleccionada de información utilizando unos procedimientos: búsqueda, ordenación y consulta. También nos ayuda a imprimir solamente la parte de la información que deseamos ver en un informe. En cualquier momento, podemos personalizar la apariencia o editar el contenido de cualquier parte de nuestra base de datos.

CARACTERÍSTICAS DE ACCESS

- Cambiar el nombre de la columna para que se ajuste mejor.
- Cortar el texto fácilmente utilizando un menú abreviado.
- Crear automáticamente una nueva tabla con el asistente para tablas.
- Filtrar una tabla para presentar solamente los registros que deseamos ver.
- Exportar solo de una tabla a una hoja de cálculo de Excel.
- Esquematizar el diseño de nuestra base de datos a ver todas las relaciones a la vez utilizando la ventana de relaciones.
- Establecer la propiedad de un campo fácilmente utilizando el botón general que nos ayuda a seleccionar las opciones.



- Ajustar la posición de los controles de un formulario desplazándolo un solo punto en la cuadrícula en un momento.
- Pero su mayor desventaja es que no puede crear archivos ejecutables (hasta el momento).

ELECCIÓN DE UN MANEJADOR DE BASE DE DATOS PARA PC's

Uno de los aspectos importantes de los manejadores de bases de datos más recientes es que logran que entre la característica de poder que ofrecen la mayoría de los fabricantes se cuenta la posibilidad integrada de enlace en redes, la posibilidad de colocar un programa nuevo de software en la red, para ofrecer acceso compartido a datos y lenguajes de programación que se puedan utilizar para desarrollar aplicaciones adecuadas a las necesidades de cada uno de los usuarios, incluyendo menús y reportes y que ese poder sea más accesible con una serie de menús, editores de toda la pantalla y características automáticas que vuelvan relativamente sencilla la realización de operaciones relacionales. Productos pioneros como Dbase por ejemplo, eran predominantemente controlados por líneas de comandos. Para aprovechar este software se tenía que aprender un gran número de comandos y procedimientos. Esto ya no sucede con los productos nuevos, tal es el caso de Access. La demanda del usuario de acceso a base de datos ha apresurado a los productores de software a diseñar interfaces realistas. Recientes adelantos técnicos (implantación de gráficos en Bases de Datos), como se mencionó anteriormente con Access se puede trabajar con la implementación de los avances OLE (*Object Link Embedd*) en la versión 2.0, incorpora datos de Word 6.0, Excel 5.0, Power Point 3.0 y Microsoft Mail. Access es una aplicación que se desarrolló bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Para poder elegir la base de datos que se acerca a nuestras necesidades, fué necesario realizar un criterio de selección el cual se muestra en la Tabla 1.5.1, Tabla 1.5.2, Tabla 1.5.3, Tabla 1.5.4, Tabla 1.5.5, Tabla 1.5.6, Tabla 1.5.7, Tabla 1.5.8, Tabla 1.5.9, Tabla 1.5.10, Tabla 1.5.11 y Tabla 1.5.12



Clipper	Nantucket
FoxPro	Fox Software, Inc
Informix-SQL	Informix
Paradox	Borland International
Access	Microsoft Corp.

Para fines practicos. se abreviarán con las siguientes letras a los productos

Clipper = C

FoxPro = F

Informix-SQL = I

Paradox = P

Access = A

bueno	X				
regular		X		X	X
malo			X		

Tabla 1.5.1. Evaluación de Tiempo de Procesamiento en Batch



Y generación de índices					
bueno		X		X	
regular	X				X
malo			X		

Tabla 1.5.2. Evaluación de Tiempo Importación y generación de índices

bueno			X		
regular	X	X		X	X
malo					

Tabla 1.5.3 Evaluación de Soporte de Núm Máximo de archivos abiertos

bueno		X		X	X
regular	X		X		
malo					

Tabla 1.5.4 Evaluación de Facilidad de Uso

bueno		X	X	X	X
regular	X				
malo					

Tabla 1.5.5 Evaluación que Proporciona Seguridad



Dato					
bueno	X			X	X
regular		X	X		
malo					

Tabla 1.5.6 Evaluación de la Integridad de los Datos

bueno					X
regular	X	X	X	X	
malo					

Tabla 1.5.7 Evaluación de la Presentación de Calidad en los Reportes y Formas



Facilidad de Programación	C	CA	CP	DP	DP
bueno					X
regular	X	X		X	
mallo			X		

Tabla 1.5.8 Evaluación en la Facilidad de Programación

	C	CA	CP	DP	DP
bueno					X
regular	X	X	X	X	
mallo					

Tabla 1.5.9 Evaluación en la Compártición de los datos con otras bases de datos



Diversidad en Tipos de Datos	C	F	S	P	A
bueno					X
regular	X	X		X	
malo			X		

Tabla 1.5.10 Evaluación en la Diversidad en Tipos de Datos

	C	F	S	P	A
bueno					
regular					X
malo	X	X	X	X	

Tabla 1.5.11 Evaluación en el Manejo de Graficos para formulario



Para establecer la puntuación que obtuvo cada producto se determinó un valor para cada categoría determinada de la siguiente manera:

bueno = 2 puntos

regular = 1 punto

malo = 0 puntos

Obteniendo finalmente la siguiente tabla de puntuaciones : (Tabla I.5.12)

CARACTERÍSTICA	C	P	I	P	A
Tiempo de procesamiento en Batch	2	1	0	1	1
Tiempo de importación y generación de índices	1	2	0	2	1
Soporte de Núm Máximos de archivos abiertos	1	1	2	1	1
Facilidad de Uso	1	2	1	2	2
Proporciona Seguridad	1	2	2	2	2
Integridad en los datos	2	1	1	2	2
Presentación de Calidad en los Reportes y Formas	1	1	1	1	2
Facilidad de Programación	1	1	0	1	2
Compartición de Datos con otras base de datos	1	1	1	1	2
Diversidad de tipos de datos	1	1	0	1	2
Manejo de Gráficos Para Formularios	0	0	0	0	1
TOTAL	12	13	8	14	18

Tabla I.5.12 Evaluación para la Puntuación final

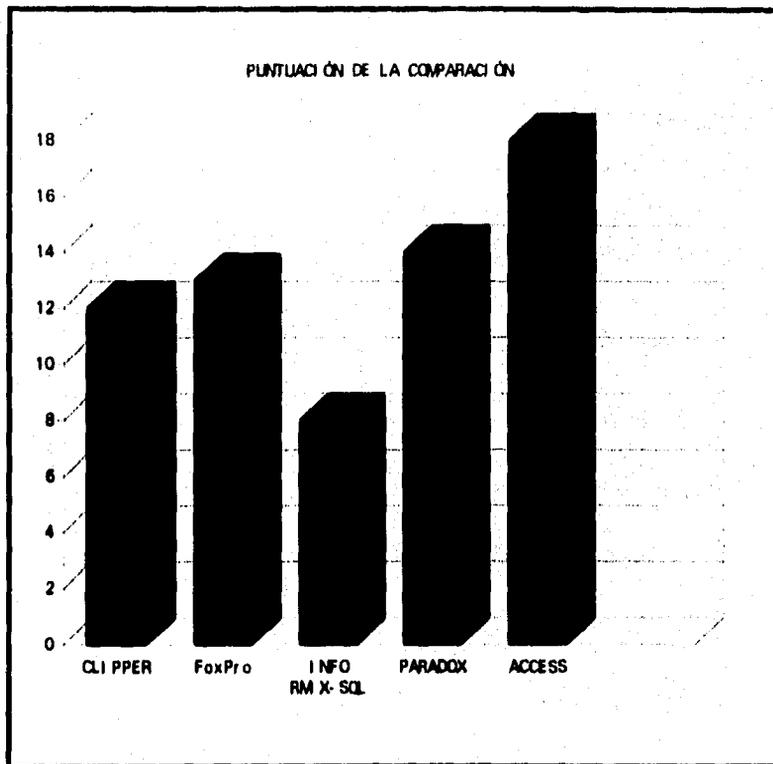


Tabla 1.5.13. Puntuación de la Comparación

La elección de la base de datos se basó, además de todo lo anterior, en la disponibilidad comercial. Por lo antes mencionado y en relación a las capacidades de manejo, programación y de conocimiento, entre otras, asociadas, se determinó el utilizar Microsoft Access para el desarrollo del sistema.



INFORMACIÓN ADICIONAL DEL MANEJADOR DE LA BASE DE DATOS SELECCIONADA (características y funcionamiento).

Microsoft Access, es un sistema de administración de bases de datos relacionales para Microsoft Windows. El diseño de Microsoft Access está orientado a ofrecer una insuperable potencia de acceso a los datos, que se combina con la extrema facilidad de uso que permite Windows. Access cuenta con sencillas herramientas, que proporciona la potencia de una base de datos de características completas

Para poder sacar el máximo provecho de las herramientas con que cuenta Access, resulta fundamental saber diseñar correctamente una base de datos. La clave para comprender el funcionamiento de Access estriba en tener bases sólidas acerca de la organización de las bases de datos.

Pasos para el diseño de una base de datos:

1. **Determinar el propósito de la base de datos**
2. **Precisar las tablas (entidades)**
3. **Distinguir los campos(atributos).**
4. **Definir las relaciones entre tablas.**
5. **Depurar el diseño.**

En la figura 1.5.1 se muestra la ventana de la base de Datos con sus respectivos elementos que la forman.



FIG 1.5.1 Muestra la ventana de Una Base de Datos de Access

ARQUITECTURA DE ACCESS

Cualquier herramienta que forme parte de Access es llamada objeto, los principales objetos que permiten el manejo integral de la información son:

Objeto	Descripción
Tablas	Se definen y usan para almacenar información. Cada tabla contiene información acerca de un tema en particular. Las tablas están representadas por columnas y registros.
Consultas	Responden a una serie de preguntas acerca de datos almacenados en tablas.
Formas	Utilizadas para dar mejor presentación a información proveniente de tablas o consultas. Es un objeto que se usa para ver y editar información en la base de datos, o trabajar registro a registro.
Reportes	Diseñadas para dar formato, calcular, imprimir y agrupar información de la base de datos.
Macro	Es un conjunto de acciones usadas para automatizar tareas comunes; como abrir una forma.
Modulo de Programación	Objeto que contiene una colección de declaraciones, instrucciones y procedimientos de Access Basic.

Access es un paquete que trabaja dentro del ambiente windows, brindando así un modulo de trabajo más amigable y sencillo. La ejecución de las tareas es a base de seleccionar iconos que se asocian con ellas .

El manejo de access se realiza por medio de ventanas, controles del usuario y ayuda en línea.

CARACTERISTICAS DE LA TABLA

Antes de crear cualquier otro objeto de una base de datos como lo son consultas, formas o reportes, se deben de crear primero las tablas.

Una tabla es una colección de datos acerca de un tema en particular, los datos en una tabla son presentados en un formato matricial con columnas llamadas **campos** y renglones llamados **registros**.

En una base de datos puede haber muchas tablas, cada una de las cuales almacena información sobre un tema distinto. Cada tabla puede contener datos de muchos tipos.

En el modo de presentación diseño de una tabla siempre aparece una ventana, donde la parte superior se utiliza para la declaración de campos y la parte inferior para especificar las propiedades correspondientes a cada campo

La declaración de los campos consiste en determinar el nombre y el tipo de datos correspondiente.

Se agrega un campo a la tabla por cada categoría de información que se desea almacenar. Por ejemplo, a una tabla que almacene registros de empleados tal vez se le quieran agregar campos como nombre, dirección y el número telefónico de los empleados.



Cuando se declaran campos es necesario especificar:

- 1.- Nombre del Campo.
- 2.- Tipo de Datos.
- 3.- Descripción (es opcional).

El primer paso es teclear el nombre del campo en la columna Nombre campo. Puede contener como máximo 64 caracteres (letras o números), incluyendo espacios.

Después de teclear el nombre del campo, se debe escoger el tipo de dato que tendrá el campo. En la figura 1.5.2 se muestra una tabla que diseña Access.

Nombre campo	Tipo de datos	Descripción
PubID	Número	Publisher's Index
Name	Texto	Publisher's Short name
Company Name	Texto	Publisher's Company Name
Address	Texto	Publisher's Address
City	Texto	Publisher's City
State	Texto	Publisher's State

Propiedades del campo	
Clave primaria	Si
Máscara de entrada	
Tipo	
Vista preliminar	
Regla de validación	
Tipo de validación	
Requerido	No
Permite longitud cero	No
Inmóvil	No

FIG 1.5.2 Figura que presenta una Tabla de Access



Los tipos de Datos que maneja Access son:

Tipo de Dato	Almacenamiento	Tamaño
Texto	Caracteres Alfanuméricos	1 byte por caracter. Acepta hasta 255
Memo	Caracteres alfanumericos (usualmente grandes párrafos)	Acepta hasta 64,000 bytes.
Numérico	Valores numéricos (enteros o valores fraccionarios)	1,2,4 u 8 bytes
Fecha/hora	Fechas y horarios	8 bytes
Moneda	Valores monetarios	4 bytes
Contador	Valor que aumenta progresivamente	4 bytes
Si/No	Valores Booleanos	1 bit
Objeto OLE	Objetos OLE, gráfico, u otra información binaria	Acepta hasta 1 gigabyte (limitado por espacio en disco duro)

Las propiedades de un Campo.- Cuando se abre una tabla en la parte del diseño, Access despliega las propiedades del campo que se está abriendo en la parte inferior de la ventana. Las propiedades se especifican una por una para cada campo.

Propiedades para los diferentes tipos de datos:

- Tamaño de campo.
- Formato.
- Máscara de entrada.
- Lugares decimales.
- Título.
- Valor predeterminado.
- Regla de validación.
- Texto de validación.
- Requerido.
- Permitir longitud cero.
- Indexado.

CONSULTAS

Una consulta es una respuesta a una serie de preguntas acerca de los datos almacenados en una o más tablas.

Microsoft Access, cuenta con el asistente para consultas, que le ayuda a crear consultas complejas para tareas frecuentes de administración de datos, tales como ver una presentación en tabla de referencias cruzadas de sus datos, buscar todos los registros duplicados de una tabla o consulta, o buscar todos los registros que sean distintos de los registros de la tabla.



Las consultas tienen cuatro tipos de modo de presentación, que son :

Diseño.- Crear y modificar la estructura de una consulta.

Hoja de datos .- Mostrar y en algunos casos agregar o editar datos.

Presentación Preliminar.- Mostrar la apariencia que tendrá el objeto una vez impreso

SQL .- Mostrar y editar la instrucción SQL de la consulta.

Las consultas sirven para:

- Hacer cambios a información contenida en tablas
- Como recurso para elaborar formas, reportes o incluso otras consultas
- Obtener información de diferentes tablas, que a su vez están relacionadas.
- Realizar cálculos

La creación de consultas se puede realizar con un Asistente de Consulta o sin él .El Asistente le ayudará a crear consultas avanzadas, además de acelerar el diseño. En la figura 1.5.3 se muestra un ejemplo de un tipo de consulta que presenta Access.

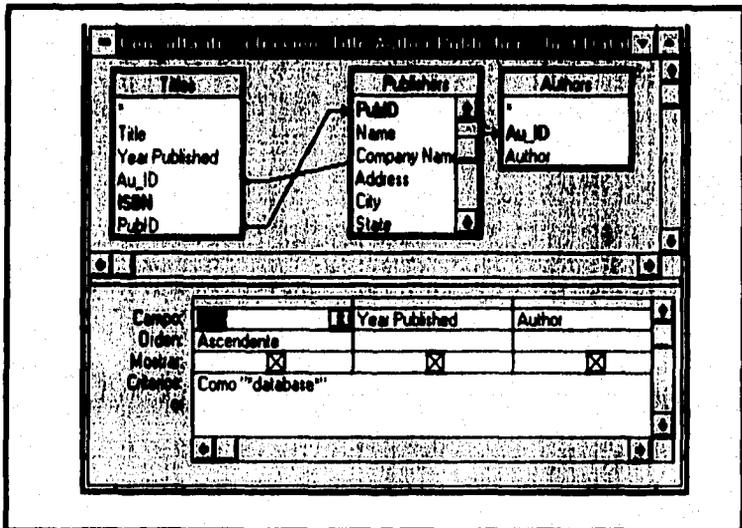


FIG 1.5.3 Presentación de una Consulta



Los tipos de consulta con el Asistente son:

TIPO DE CONSULTAS	Explicación
Selección	Son las más comunes, aquí se plantean preguntas sobre datos almacenados en las tablas y la respuesta es una hoja de datos, en donde se puede ver y cambiar los datos de las tablas subyacentes
Referencias Cruzadas	Presenta los datos con títulos en las filas y columnas, como en una hoja de cálculo. Con una consulta de tabla de referencia cruzada se puede resumir gran cantidad de información en un formato de fácil lectura
Consultas de Unión	Combina campos coincidentes de dos o más tablas
Consultas de definición de datos	Creación, modifica o elimina tablas de una Base de Datos de Access utilizando instrucciones de SQL
Acción	Modifican muchos registros en una sola operación. Se utiliza para crear una nueva tabla, eliminar registros de una tabla o modificarlos. A partir de la selección actualizan, agregan y eliminan datos.



Saber como especificar criterios en una consulta es lo más importante para diseñarla. Únicamente con ellos podrá extraer exactamente la información que requiere.

Los criterios indican a Access la información que presentará en la hoja de respuesta dinámica.

Entre los criterios que pueden incluirse en una consulta están:

- Selección de un rango de registros.
- Patrón de Caracteres.
- Registros que no tienen un valor asociado.
- Registros que tengan múltiples criterios.
- Registros que contengan o no valores en determinados campos.
- Registros para una fecha en específico.
- Registros basados en cálculos.
- Criterios para grupos de campos.
- Criterios para todos los campos.
- Desplegar parte de un campo texto.
- Desplegar parte de un campo fecha.
- Selección de valores únicos.

CREACIÓN DE FORMULARIOS

El corazón de toda aplicación para una base de datos es su interface, definida por sus formularios de pantalla. Mediante el uso del Wizard de Formularios, es bastante simple lograr un conjunto de formularios básicos partiendo de los cuatro tipos disponibles. Una columna, Tabulado, Gráfico y Principal/Subformulario, y asignarle uno de los cinco estilos posibles. Además, existen un nuevo formulario Autoform que crea los mejores formularios posibles



basándose en la tabla o la consulta seleccionada. También es posible generar formularios que usen las capacidades de macro completa que ofrece Access para organizar su aplicación y procesamiento de control, eliminando gran parte de la codificación que normalmente se requiere para tal fin.

Una forma es una herramienta de Access que permite introducir, cambiar, ver o imprimir información. ver figura 1.5.4

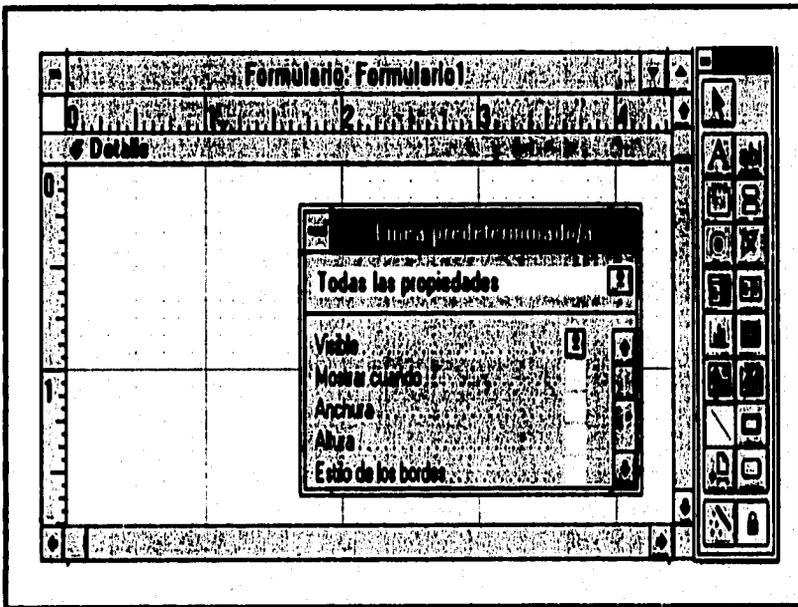


FIG 1.5.4 Presenta el Diseño de un Formulario de Access

Una forma ayuda a ver fácilmente información, en tanto se ahorra tiempo y previene errores al introducir datos. Toda la información en una forma está contenida en controles, objetos que despliegan la información, optimizan acciones o decoran la forma



Con Access podrá diseñar formularios fáciles de utilizar y que presenten la información de la forma deseada. podrá utilizar diversos elementos de diseño (texto, datos, imágenes, líneas, y color (figura 1.5.5)) para crear exactamente el formulario deseado, se debe de elegir los elementos que desea utilizar y determina la forma que desea disponerlos en el formulario.

	Texto
	Imágenes
	Líneas
	Color

FIG 1.5.5 Tabla de Elementos de Formas

Los tipos de formas con que cuenta Access son:

- **Columna simple.-** Despliega los valores de un registro en una columna, además de que cada valor es separado por una línea. Muestra solo un registro.
- **Tabular.-** Muestra todos los valores de un registro en un renglón, haciendo así un formato de renglones y columnas. Especifica varios registros a la vez.



- **Principal/subforma.** Una subforma dentro de una forma enseña la relación uno a muchos entre la información de la forma principal y la información de la subforma. La Forma principal exhibe la información en un formato de hoja de datos.
- **Gráfico.** Una forma gráfica muestra información con un formato de gráfica, ya sea de barras, de columna o de pastel

Los modos de presentación de una Forma son:

- **Diseño.**- Facilita la creación o modificación del diseño de una forma.
- **Formulario.**- Permite introducir, cambiar y visualizar la información. En este modo de presentación, usualmente se ven todos los campos de un registro a la vez.
- **Hoja de datos.** donde introduce, cambia y visualiza la información. En este modo de presentación se pueden ver muchos registros al mismo tiempo, no así todos los campos de un registro sin que haga el *scroll*, además no despliega imágenes, objetos OLE o botones como lo muestra el modo de presentación formulario.
- **Presentación Preliminar.**- Indica el aspecto que tendrá una forma al ser impresa.

CARACTERÍSTICAS DE LOS INFORMES.

Un informe es un conjunto de información que usted organiza según criterios y al que aplica formato de acuerdo con sus especificaciones.

Un informe constituye una forma de recuperar y presentar los datos como información significativa que se puede usar y distribuir.



Con Access podrá diseñar informes que presenten la información de la forma deseada. Podrá utilizar numerosos elementos de diseño (texto, datos, líneas, cuadros y gráficos) para crear exactamente el informe que necesitará, es necesario elegir los elementos que desea utilizar y determinar la forma en que desea disponer en el informe.

Crear un informe es un método muy eficaz de presentar los datos en forma de documento impreso. Si bien es posible imprimir los formularios y las hojas de datos, los informes brindan mayor flexibilidad a la hora de presentar información de resumen.

En Access podrá crear informes para:

- Organizar y presentar los datos en grupo
- Calcular totales y parciales, totales de grupo, sumas de totales y porcentaje de totales.
- Incluir subformularios, subinformes y gráficos.
- Presentar los datos en una formato atractivo, con imágenes, líneas y fuentes especiales

CONTROLES

Access ofrece una variedad de controles que se pueden agregar a la forma o informe. Para la creación de controles se utiliza la caja de herramientas, la cual se despliega automáticamente cuando se abre una forma o informe en el modo de presentación diseño.

Los tipos de controles que proporciona Access son:

Dependientes. La información del control proviene de tablas de consultas. Se usan para desplegar, introducir y actualizar valores de los campos de una base de datos. Los valores pueden ser texto, número, imágenes, gráficas, etcétera. El cuadro de texto es el control dependiente más común.



CARACTERÍSTICAS DE LAS MACROS

Una macro realiza automáticamente una tarea o una serie de tareas. Cada tarea que usted desee realizar con Access se denomina acción.

Access ofrece una lista de acciones, entre las que se selecciona una o más para crear una macro. Cuando se ejecute la macro, Microsoft Access realizará las acciones en el mismo orden en que figurarán dentro de la macro, utilizando los objetos o los datos especificados.

Mediante uso de macros puede conseguir que los formularios, los informes, y los demás objetos de la base de datos funcionen conjuntamente de forma aun más inteligente.

Los macros también son útiles para automatizar tareas rutinarias o repetitivas, como la impresión de informes semanales. Automatizando sus tareas rutinarias conseguirá mayor eficacia y precisión en la base de datos, ya que una macro siempre realizará la tarea de la misma forma.

CARACTERÍSTICAS DE LOS MÓDULOS

Los módulos almacenan código de Access Basic que puede escribirse para personalizar, mejorar y extender las posibilidades de la base de datos.

El contenido de la información aquí presentada, sólo fué un panorama general de todo lo que posee Microsoft Access, para obtener mayor información, consultar el manual de Usuario de Microsoft Access, que constituye una completa guía para crear y trabajar con una base de datos de Access. Además del manual de Usuario, existe La ayuda en pantalla, que proporciona información de referencia e instrucciones prácticas para todas las tareas que pueda realizar Access.

Además existen las Fichas-Guía que son instrucciones en pantalla que proporcionan instrucciones paso a paso para ayudarle a conocer Microsoft Access mientras crea y utiliza su propia base de datos.



1.6 ANÁLISIS Y SELECCIÓN DEL SOFTWARE PARA LIGAR APLICACIONES DE PC'S CON SERVIDORES.

SLIP

El software PC-NFS sobre un puerto serial incluye una vista superficial de *Serial Line Internet Protocol* (SLIP).

El software PC-NFS permite a los usuarios de Pc obtener el máximo de un ambiente en red, como es compartir archivos, impresiones, correo electrónico y otros recursos.

El software PC-NFS utiliza protocolos de comunicación llamados en su conjunto *Protocolo Internet (IP)*. IP codifica la información de tal forma que pueda ser transmitida a través de distintos enlaces de comunicaciones. Si se conecta la PC a un servidor, utilizando una interconexión serial, el PC_NFS utiliza el SLIP para realizarla.

Un enlace serial entre una PC y un servidor puede realizarse de dos maneras:

- Una conexión remota, con un modem y línea telefónica.
- Una conexión directa con un puerto serial RS-232.

Una conexión SLIP bajo una PC corriendo el software de PC-NFS conecta un servidor Sun corriendo una implementación SLIP compatible. El servidor proporciona al usuario de PC las mismas aplicaciones y ambientes de red que se encuentran disponibles, al usuario de PC usando una red *Ethernet* u otra red de área local.

Todos los servicios y aplicaciones del software PC-NFS trabajan sobre una conexión SLIP con algunas limitaciones de velocidad.



Cada red SLIP debe tener su propia dirección y nombre de Internet; cada Pc y cada servidor de una red SLIP deben de tener su nombre y dirección en el ambiente, los nombres y direcciones deben de ser diferentes a los de la red.

Las figuras 1.6.1 y 1.6.2. ilustran los conceptos de una red *Ethernet* y una red SLIP:

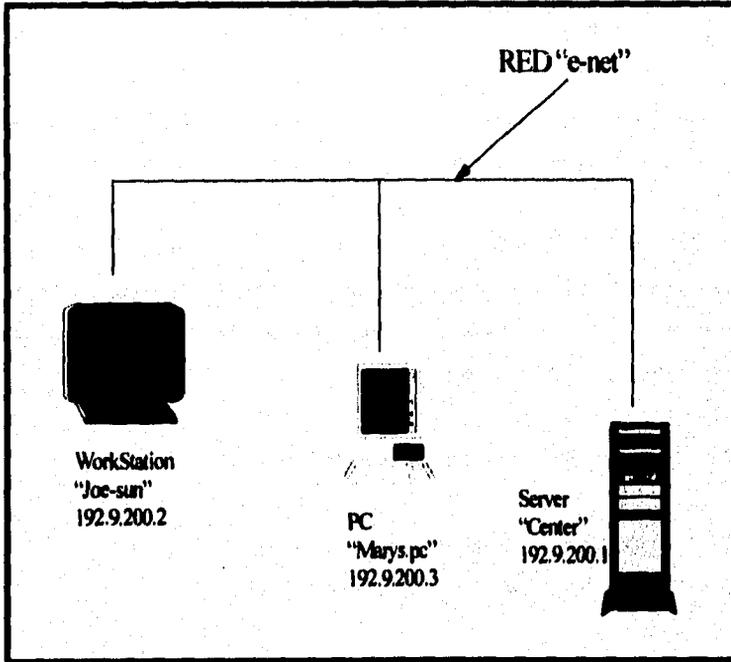


FIG.1.6.1. Red Ethernet

La figura 1.6.1 muestra una red *Ethernet* con un servidor y dos clientes (una PC y una *Workstation*), el nombre de la red es "e-net" y su dirección es 192.9.200, cada máquina en la red tiene su propio nombre y dirección.

Una *gateway* (puerta) es una computadora que conecta dos o mas redes. La figura 1.6.2 muestra la misma red de la figura 1.6.1, con una red SLIP conectada mediante una *gateway*. El

servidor de la red *Ethernet* ha sido configurado como la puerta SLIP y una PC ha sido añadida en una conexión SLIP. La conexión SLIP puede ser directa o remota.

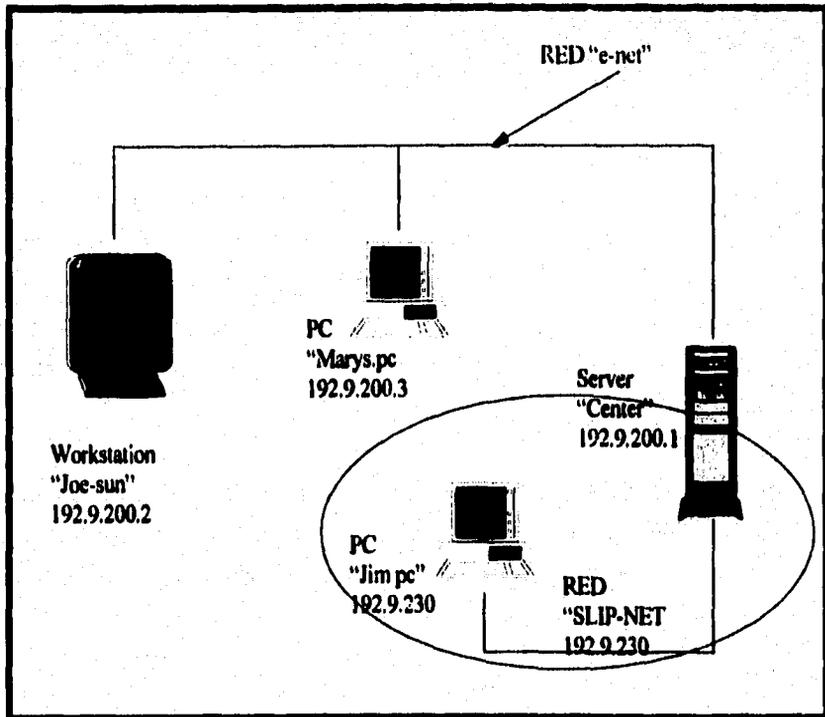


FIG. 1.6.2 Red SLIP.

Tenemos ahora dos redes, cada una con su propio nombre y dirección. El servidor ahora tiene dos nombres y dos direcciones, dado que se está usando para dos propósitos como servidor y puerta.

¿PORQUE USAR SLIP?

Una conexión remota SLIP proporciona la flexibilidad y conveniencia de utilizar PC-NFS lejos del servidor. Si la Pc se encuentra conectada normalmente a una LAN, quizá se quiera conectar al servidor desde otro lugar y la conexión remota SLIP permite realizarlo.



Además, un cable serial es muchas veces la única manera práctica de conectar una PC a la red y una conexión SLIP permite realizarla mediante este tipo de conexión.

- Sistema Sun

El servidor PC-NFS SLIP esta diseñado solamente para máquinas que utilizan el sistema operativo SunOS. Este software proporciona soporte para Sun-3, Sun-4, y Sparc Station, utilizando el sistema operativo Solaris 2.x. Se debe asegurar que el sistema Sun se utilice como GATEWAY (puerta) tenga 1 megabyte o más espacio libre en disco para en el colocar el software de distribución, se pueden instalar más puertos seriales en el sistema Sun para instalar la tercera parte de tarjetas seriales.

- En equipo Pc

Una Pc necesita MS-DOS 3.3 o posterior, y el software PC-NFS versión 3.5 o posterior. Por lo menos una Pc en la red deberá usar el PC-NFS usando una tarjeta Ethernet para poder transferir los archivos de la Pc al servidor. La Pc requiere un cable RS-232 para conexión directa, o un modem en caso de que sea un acceso remoto.

FIG. 1.6.3. Características técnicas de SLIP.

MODEMS Y CABLES

Para usar NFS y compartir archivos se requiere un modem que sea capaz de corregir errores, si el modem es de menor velocidad de 9600 esta acción es deshabilitada.

Las conexiones de cable y modem utilizan las siguientes velocidades: 1200, 2400, 4800 y 9600 baudios; la velocidad de 9600 proporciona la mejor conexión.



PROGRAMA EN C

Debido a sus características C se considera muy eficaz para realizar un programa que ligue nuestra PC con el servidor que lleva el control de las claves que accesan a la Red Internet.

C es un lenguaje de programación de propósito general que ha sido estrechamente asociado con el sistema UNIX en donde fué desarrollado puesto que tanto el sistema como los programas que corren en él están escritos en lenguaje C. Sin embargo, este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina, por lo que es muy utilizado para escribir programas de diversas disciplinas.

C proporciona una variedad de tipos de datos, además existe una jerarquía de tipos de datos derivados, creados con apuntadores, arreglos, estructuras y uniones. Las expresiones se forman a partir de operadores y operandos; también proporciona un adecuado control de flujo; las funciones pueden regresar valores de tipos básicos, estructuras, uniones o apuntadores. Cualquier función puede ser llamada recursivamente. Las variables locales son normalmente "automáticas", o creadas de nuevo con cada invocación. Las variables pueden ser internas a una función, externas pero conocidas sólo dentro de un archivo fuente, o visibles al programa completo.

Como cualquier otro lenguaje, C tiene sus defectos. Algunos de los operadores tienen la precedencia equivocada; algunos elementos de la sintaxis pueden ser mejores. A pesar de todo, C ha probado ser un lenguaje extremadamente efectivo y expresivo para una amplia variedad de programas de aplicación.

C es conocido como un *lenguaje de nivel medio*. Nivel medio no tiene un significado negativo. No significa que C sea menos poderoso, manejable o menos desarrollado que un lenguaje de alto nivel. Es considerado de nivel medio, ya que combina elementos de lenguajes de alto nivel con la funcionalidad del ensamblador.



El código C es muy transportable. La portabilidad significa que se puede adaptar el software escrito para un tipo de computadora a otro tipo. El soporte C del estándar ANSI (*American National Standards Institute*) asegura la transportabilidad del código a otros entornos.

Todos los lenguajes de programación de alto nivel soportan el concepto de tipos de datos. Un *tipo de dato* define un conjunto de valores que una variable puede almacenar con un conjunto de operaciones que la computadora pueda realizar sobre esa variable. Los tipos de datos comunes incluyen enteros, caracteres y reales. Aunque C tiene cinco tipos de datos incorporados, no es un lenguaje fuertemente tipado como Pascal o Ada. C siempre permitirá todas las conversiones de tipo. Por ejemplo, en la mayoría de las expresiones se pueden mezclar libremente caracteres y enteros. En general, los compiladores de C realizan una ligera comprobación de error en tiempo de ejecución, como la comprobación de los límites de array o la compatibilidad del tipo de argumento. Estas comprobaciones son responsabilidad del programador. La razón para esta carencia de comprobación en tiempo de ejecución se realiza de una forma más lenta que la ejecución del programa.

Como lenguaje de tipo medio, C permite la manipulación de bits, bytes y direcciones. Esta posibilidad hace de C un lenguaje adecuado para la programación a nivel de sistema donde son comunes estas operaciones.

Software:	Programa en "C"
Requiere:	Pc 386, Servidor y conexión serial..
Comentarios:	Programa desarrollado para ligar un servidor y una Pc, con el propósito de monitorear las claves de acceso a Internet (las cuales se encuentran en el servidor) comprobando a través de la verificación de datos proporcionados por el usuario con la Base de Datos que se encuentra en la Pc.

FIG. 1.6.4. Característica técnicas de C.

**PROCOMM.**

Para un adecuado aprovechamiento de la unidad de comunicaciones, es necesario seleccionar un programa que se adapte de una parte al modem y de otra a nuestros intereses. Entre éstos, uno de los más completos y de propósito general es PROCOMM. Permite llamar a otro modem; transferir ficheros con diversos protocolos; emular diversas terminales; dar facilidades al usuario para registrar lo que ocurre durante la comunicación; la posibilidad de responder a la llamada de otro modem; ajustar los parámetros de la línea; programar un proceso de conexión; etc. Es evidentemente un programa muy versátil.

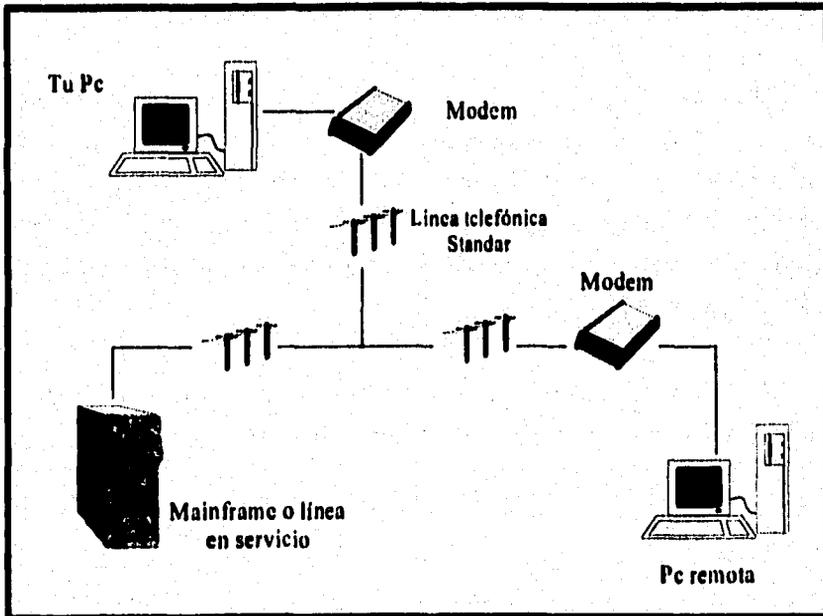


FIG. 1.6.5. Procomm coordina las actividades de los componentes involucrados en las telecomunicaciones.



El ingenioso proceso de instalación de ProComm le da una variedad de opciones. Se puede escoger su configuración de modem de una lista de casi 100 cadenas de configuración y escoger protocolos prefijados para recepción de un menú de más de 10. Los protocolos de transferencia de archivos incluyen ASCII, *CompuServe B Plus*, *Kermit*, *XModem*, *ZModem* y más. Más tarde, puede cambiar más de 100 parámetros de la sesión, tales como el ajuste de la velocidad en bits por segundo dentro de un rango de 300 a 115.200 o la selección de uno de ocho puertos seriales definidos por el usuario.

Los menús expandibles de ProComm añaden otra dimensión al programa, llamando el directorio de llamadas, el menú de emulación de terminal y el servicio de ayuda dependiente del contexto. El menú de comandos le da una lista de las opciones de comunicación, que incluyen el modo de charla y el archivo de registro. Para ahorrar tiempo, puede evitar el menú de comando y ejecutar los comandos en cualquier momento usando las combinaciones de ALT-tecla.

El directorio de llamadas le deja añadir hasta 200 nombres, números y ajustes de sesión para sistemas a los que se llame frecuentemente. Puede crear tantos directorios de llamada como su disco pueda contener. ProComm le deja fijar una lista de números a los que llamará hasta que uno conteste, o puede hacer que siga llamando al mismo número hasta que conecte. El programa emula unas 33 terminales, que incluyen DEC VT52, VT102, VT220 y VT320, así como el IBM 3270 y 3101, ANSI y TTY.

ProComm incluye algunas opciones de seguridad. Aunque no tienen una contraseña para protección contra el acceso local, puede fijar contraseñas cuando use el programa como anfitrión remoto.

El lenguaje de guiones del programa ASpect, ofrece 208 verbos. Se puede usar la opción de aprendizaje del programa para construir un guión a partir de sus golpes de tecla. El editor que se incluye en ProComm puede manejar múltiples archivos sin problema y el mismo incluye una opción en búsqueda y reemplazo.



Además de sus protocolos incluidos, ProComm trabaja con tres protocolos externos definidos por el usuario. Una ventana que muestra el tiempo estimado, el conteo de bytes y el promedio de caracteres por segundo, junto con un gráfico de barra dejan seguir el progreso de la transferencia de archivos. ProComm emite un sonido cuando completa una transferencia.

Como podemos ver PROCOMM es un programa muy completo, que presenta varias opciones para realizar una comunicación adecuada, además nos damos cuenta que éste nos sirve para el propósito de éste tema, es decir, para ligar un servidor a una Pc; esta conexión se puede realizar ya que PROCOMM cuenta con un lenguaje de programación el cuál permite automatizar el proceso de comunicación y ajustarlo a las necesidades requeridas.

Software:	ProComm Plus, versión 2.0
Requiere:	192k de RAM, DOS 2.0 o posterior.
Comentarios:	Un producto poderoso e intuitivo que presenta menús desplegables. Es un placer usar ProComm Plus. El mismo incluye apoyo para más de 100 modems, un modo de charla, y un directorio de llamadas que contienen hasta 200 nombres, números y ajustes de sesión. Por desgracia, no puede conectarse a las velocidades más altas en pruebas de ZModem basado en caracteres.

FIG. 1.6.6. Características técnicas de Procomm Plus.



DYNACOMM

El DynaComm es un poderoso producto de comunicaciones basado en Windows, puede ayudar en cualquier cosa, desde conectarse a un boletín electrónico (BBS) hasta emular una terminal 3270 para tener acceso a un modem de red. La curva de aprendizaje es baja comparada con cualquier otro producto de Windows. El programa Terminal que se incluye con Windows 3.0 es realmente un subconjunto de DynaComm, también permite emular múltiples sesiones de *mainframe* con una sola ventana.

DynaComm incluye el apoyo para DDEs y una larga lista de emulación de terminales, junto con un completo lenguaje de guiones que permite hacer cosas como la creación de cajas de diálogo y menús para sus aplicaciones. *Future Soft* recientemente añadió *ZModem* al producto.

La instalación no es complicada, pero debido a la compleja variedad de opciones de DynaComm, se debe correr primero el tutor. Las cadenas predefinidas de configuración de modem son limitadas, ya que sólo hay cuatro.

DynaComm muestra un icono de ayuda difícil de ignorar. Todos los iconos de DynaComm están bien diseñados y dan mucha información a simple vista. Esta versión no solamente añade ayuda dependiente del contexto sino también la capacidad para ver archivos del tipo GIF y RLE.

Permite modificar sus parámetros para la sesión seleccionando las opciones de menú apropiadas. Estas le dejan fijar la razón de transmisión desde 110 hasta 19.200 bps, escoger COM1 o COM4 o una conexión a módem en red, ajustar el tamaño de palabra o desactivar la verificación de paridad o la detección de portador. El intercambio de señales maneja tanto control de flujo en software como en hardware, para enviar la información más rápido a los modems que tienen posibilidad de compresión de información.

DynaComm apoya 14 tipos de emulación de terminal. El trabajar con estos estándares comunes para terminales asegura la compatibilidad con cualquier servicio de comunicaciones.



DynaComm permite usar o modificar guiones preconfigurados de muestra y los mismos son útiles para aprender el robusto y complejo lenguaje de guiones de DynaComm. Este tiene su propio editor que permanece activo cuando una ventana de edición que contiene un guión está activa. El lenguaje permite personalizar completamente las comunicaciones, y controlar sesiones de anfitrión entre PCs o entre PCs y mainframes 3270.

Podemos concluir, que DynaComm también nos proporciona varias opciones para poder realizar la liga entre un servidor y una PC, a través de su lenguaje de programación y utilizando sus características de programa de comunicaciones.

Software: DynaComm Asynchronous Edition, Versión 3.00.08
Requiere: 512 de RAM (se recomienda 640k), disco duro, Microsoft Windows 2.x o 3.0.
Comentarios: DynaComm incluye muchas opciones valiosas tales como el acceso a los modems de red y la emulación de terminal 3270, pero el dominarlo lleva tiempo. El versátil lenguaje de guiones puede crear una variedad de aplicaciones, incluyendo juegos.

FIG. 1.6.7. Características técnicas de DynaComm.

BTRIEVE

Netware Btrieve es un sistema de teclas indexadas para la administración, diseñado para manejo de datos de alto nivel y para una mejor productividad en la programación. Btrieve permite a las aplicaciones insertar, llamar, actualizar o borrar datos por el valor de una tecla o mediante la utilización de métodos de acceso secuencial o aleatorio. Btrieve ha sido implementado como un programa ejecutable, contenido en sí mismo, que las aplicaciones pueden acceder mediante una función de llamada.



Btrieve puede correr en una workstation (versión cliente-servidor) o en un servidor (versión servidor-base). La versión de cliente-servidor ejecuta todos los procesos en la workstation. Accesa a todos los archivos mediante funciones de llamadas al sistema operativo. Estas llamadas al sistema operativo son ejecutadas ya sean localmente (para archivos locales) o redirigidas al servidor (para archivos en el servidor).

En la versión Servidor-base, el Administrador de Grabación Btrieve se ejecuta en el servidor y un programa residente es ejecutado en la workstation. Existen programas residentes para cada tipo de plataforma que ejecutan generalmente: DOS, OS/2, y para Windows. El programa ejecutable maneja datos de entrada salida I/O entre la workstation y el servidor de la red. El Administrador de Grabación Btrieve que se ejecuta en el servidor maneja los datos I/O con el sistema de archivos.

La versión de Btrieve que corre en un servidor se ejecuta en un servidor Novell Netware. La versión cliente-servidor puede ejecutarse sobre varias plataformas y distintos sistemas operativos como son Windows, DOS y OS/2. Las aplicaciones programadas para la versión de cliente-servidor de Btrieve podrán ser ejecutadas en la versión de servidor y viceversa.

Además de proveer un sistema de administración de grabaciones, en su versión basada en servidor, para las workstation, Btrieve también acepta llamadas de otras aplicaciones basadas en servidor.

En la figura 1.6.8. se muestra una red multiambiente en la cual se utiliza Btrieve. Cada uno de los ambientes de las tres distintas workstation pueden tener acceso a cualquiera de los servidores de archivos de la red. Sus Btrives respectivos se comunican con el Btrieve del servidor que se encuentra ya instalado.

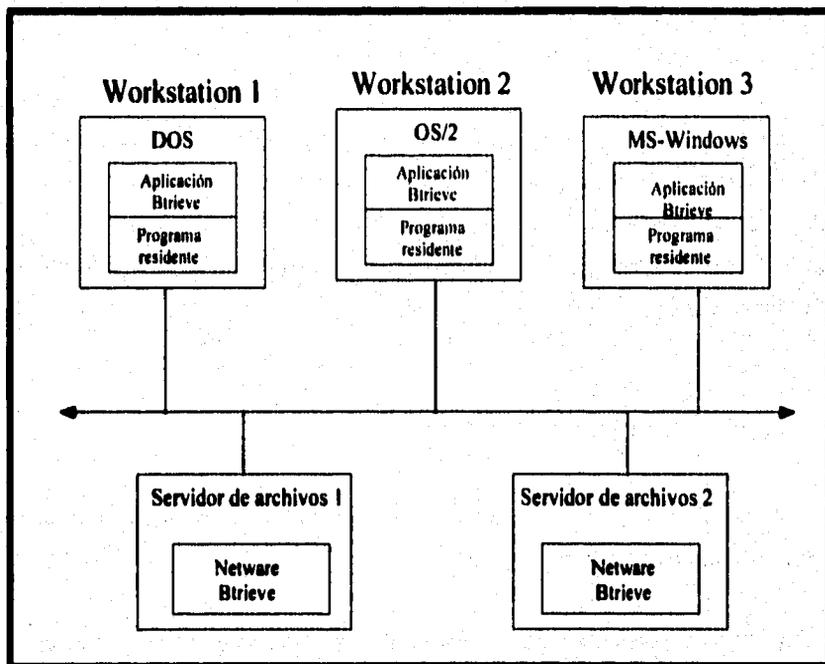


FIG. 1.6.8. Btrieve multiamiente.

Existen cuatro archivos básicos en el paquete de red de Btrieve, estos programas son:

- **BSPXCOM.NLM**
- **BTRIEVE.NLM**
- **BTRIEVE RESIDENTE**
- **BROUNTER.NLM**

Btrieve.nlm es una librería de las funciones de Btrieve en red, reside en el servidor de archivos. Ambas versiones de Btrieve pueden acceder a Btrieve en red.



La versión Basada en cliente se comunica con la versión de Btrieve en servidor haciendo uso del programa residente de Btrieve y de BSPXCOM.

Las Aplicaciones basadas en servidor pueden mandar llamadas a Btrieve al servidor local o a servidores distintos. Llamadas de Btrieve en el mismo servidor donde se originó la petición pueden usar Btrieve directamente o hacer uso de BROUTER. Las llamadas a Btrieve hechas desde otro servidor pueden acceder al Btrieve remoto haciendo uso de BROUTER y el BSPXCOM remoto.

BTRIEVE.NLM

Btrieve.nlm es una librería de funciones de Btrieve, debe de ser cargado en todos los servidores de archivos que almacenan archivos.

BSPXCOM realiza llamadas a Btrieve desde una *workstation* cliente o mediante un BROUTER remoto. Especialmente, estas llamadas realizan las siguientes tareas:

- Realiza todas las I/O de disco para los archivos de Btrieve almacenados en el servidor donde las copias accesibles de Btrieve se encuentran
- Implementa o retira todos los candados de elemento y de archivo, en el servidor donde la copia de Btrieve accesada se encuentra
- Graba las peticiones a Btrieve que resulten en cambios de un archivo.



BSPXCOM.NLM

Para hacer uso de BSPXCOM.NLM, debe de estar cargado en el servidor de archivos que se accese. BSPXCOM permite que las llamadas de Btrieve que se originen fuera del servidor se comuniquen con Btrieve.

Cuando Btrieve es accedido, la llamada puede originarse en: una copia distinta del Btrieve del servidor o bien en una versión diferente de Btrieve, como puede ser la versión basada en cliente de una workstation.

Cuando la petición proviene de otro servidor, BSPXCOM recibe la petición de la otra copia de BROUTER en el otro servidor y realiza las llamadas de función apropiadas en Btrieve para procesar la petición.

Cuando la llamada proviene de una workstation, BSPXCOM recibe la petición del programa residente en la workstation y ejecuta las funciones apropiadas para que Btrieve pueda realizar el proceso.

Después de que una petición ha sido procesada, BSPXCOM comprime la petición para que pueda ser transmitida a una copia del programa residente en una workstation o a una copia de BROUTER en otro servidor.

BROUTER.NLM

BROUTER.NLM debe de ser llamada en el servidor. Es una aplicación basada en servidor que permite que cualquier otra aplicación cargada en el servidor se comunique con Btrieve haciendo uso del programa BSPXCOM. BROUTER realiza las siguientes tareas:



- Provee acceso convencional basado en SPX para poder llamar archivos Btrieve de cualquier otro servidor en la red
- Mantiene un sistema único de identificación de código para cada aplicación para que los candados, transacciones y otros mecanismos de control de acceso puedan trabajar sin conflictos en toda la red

PROGRAMAS DE UTILERIA

Se proporcionan dos programas mas: BCONSOLE.NLM, una utileria de monitoreo de consolas, y BSETUP.NLM, una utileria que permite la instalación y configuración .

ACCESANDO A BTRIEVE

El programa de Btrieve en red funciona como si fuera una subrutina de una aplicación. Btrieve Netware (que es el nombre del programa en red) da apoyo a los siguientes métodos de acceso a Btrieve:

- Una aplicación en una workstation puede acceder Btrieve mediante un programa residente
- Una aplicación de un servidor puede acceder a Btrieve directamente en el mismo servidor al llamar el punto de acceso
- Una aplicación puede acceder Btrieve en otro servidor via el programa BROUTER

Estos métodos de acceso son explicados a continuación:

APLICACIÓN CLIENTE : Aplicaciones basadas en cliente corren en una aplicación y se comunican con Btrieve mediante un programa residente. Los siguientes pasos ilustran los



flujos de control cuando una aplicación basada en cliente accesa a Btrieve mediante el programa residente:

1. La aplicación llama a Btrieve mediante una función de llamada
2. El código de interfaz de la aplicación permite la liga con el programa residente
3. El programa residente compacta la llamada como un mensaje en la red, determina que servidor deberá de recibir la llamada, y manda la llamada al programa
4. BPSXCOM que se encuentra residente en el servidor
5. BPSXCOM recibe el mensaje de la red, valida los parámetros, y ejecuta la petición al realizar llamadas de función a Btrieve. Dependiendo de la naturaleza de la petición, esto puede incluir una operación de memoria o una operación I/O a un dispositivo de almacenamiento. BPSXCOM proporciona el resultado de la operación al programa residente en la *workstation*.
6. El programa residente proporciona los datos adecuados y código de status a las variables residentes en memoria de la aplicación que llamo a Btrieve, después regresa el control a la aplicación.

La siguiente figura ilustra un ejemplo de configuración para una red simple Novell que utiliza Btrieve Network:

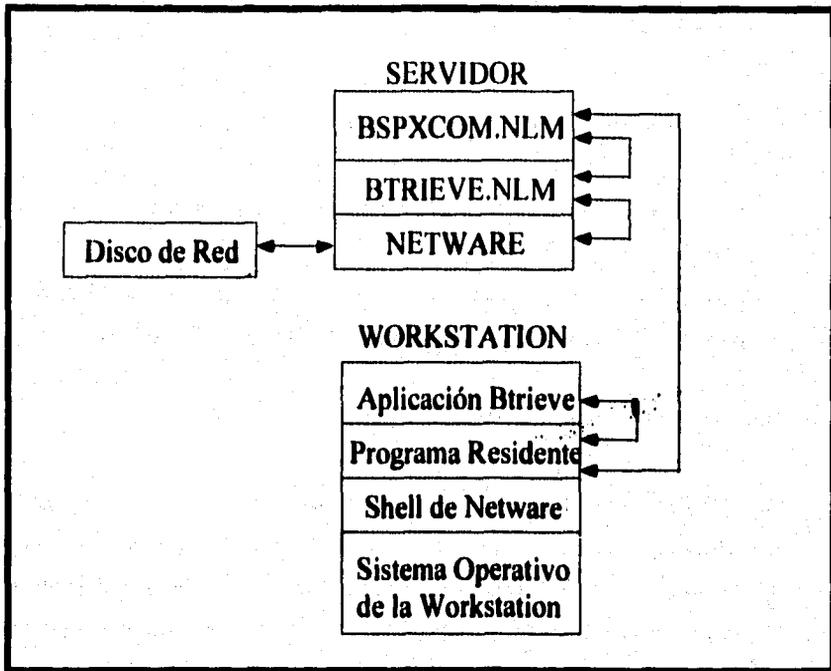


FIG. 1.6.9. Configuración de una red usando BSPXCOM.NLM Y BTRIEVE.NLM.

Aplicaciones basadas en un servidor

Una aplicación basada en un servidor creada en un ambiente NETWARE386 es llamado *Módulo de Llamada Netware* (NLM). Un NLM puede hacer dos tipos de llamadas para Btrieve: *Local* y *Remota*. Una llamada *Local* es aquella que es procesada únicamente en el servidor local, que es donde la llamada a Btrieve se origina.

Una aplicación puede realizar llamadas locales y remotas. Ya que se ha ejecutado una llamada en particular, ya sea local o remota, y es transparente para la aplicación si se tienen los programas de comunicaciones apropiados. Para que una llamada local sea ejecutada eficientemente, BTRIEVE.NLM debe de ser ejecutado en el servidor local. Si BROUTER.NLM



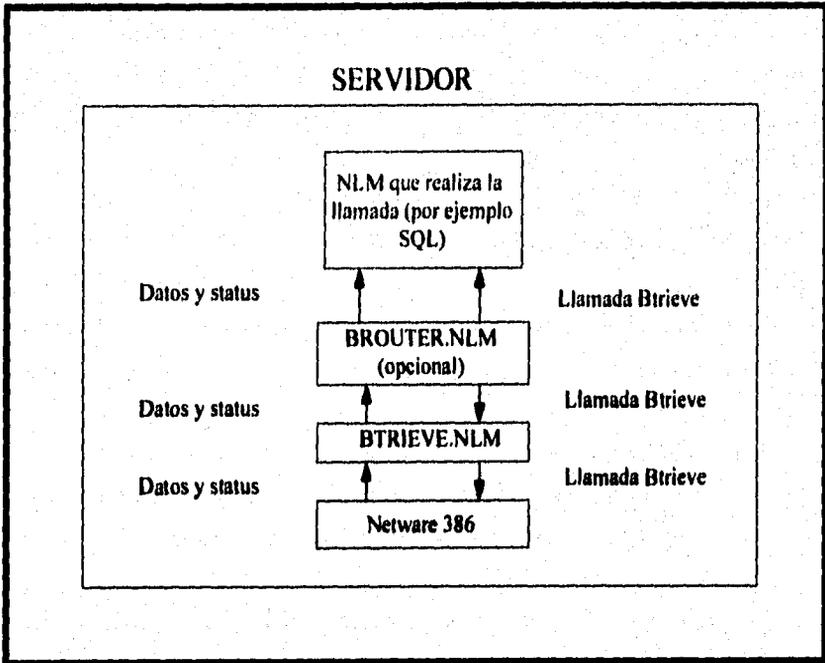
es ejecutado en el servidor local cuando la llamada se da, la función de llamada de Btrieve será tomada por BROUTER. Si no es así, las llamadas se llevan al punto de exportación. Para que una llamada remota sea ejecutada exitosamente, BTRIEVE.NLM y BROUTER.NLM deben de ser cargadas en el servidor local. BPSXCOM.NLM y BTRIEVE.NLM deben de ser llamadas en el servidor remoto.

Flujos de control de BTRIEVE

Los siguientes pasos describen el flujo de control cuando una aplicación NLM utiliza una llamada de Btrieve local:

1. El NLM hace una llamada local a Btrieve usando una función de llamada. Si el programa Brouter no ha sido cargado, la llamada se hace directamente en el punto de entrada de Btrieve. Si *Brouter* ha sido cargado, canaliza la llamada de Btrieve.
2. El proceso de llamada utiliza la librería de Btrieve para usar las funciones de llamada.
3. Los datos apropiados y los códigos de status son regresados directamente al NLM si Brouter no han sido llamado. Los datos y códigos de status son regresados al NLM que realizó la llamada via *Brouter* si han sido cargados.

La figura 1.6.10 muestra los componentes necesarios del servidor para procesar una llamada local a Btrieve.



Control de Flujos para una llamada Btrieve remota

Los siguientes pasos son los que se siguen para realizar el flujo de control cuando una NLM realiza una llamada remota a Btrieve. La figura siguiente muestra los componentes requeridos en cada servidor para procesar una llamada remota:

1. La aplicación NLM local realiza una llamada a Btrieve (usando una función de llamada) que es llevada por el *BROUTER*
2. *BROUTER* lleva la petición de Btrieve usando el *BSPXCOM* al servidor remoto
3. *BSPXCOM* realiza la petición a Btrieve ejecutando una función de llamada



4. El Btrieve en el servidor remoto regresa los datos apropiados, así como su código status al programa BSPXCOM
5. BSPXCOM regresa los datos y el código de status al *BROUTER* en el servidor local donde la llamada se originó.
6. *BROUTER* proporciona estos resultados al NLM que realizó la llamada. Los resultados son depositados en la memoria del NLM designada por los parámetros suministrados por Btrieve. El control es regresado a la aplicación.

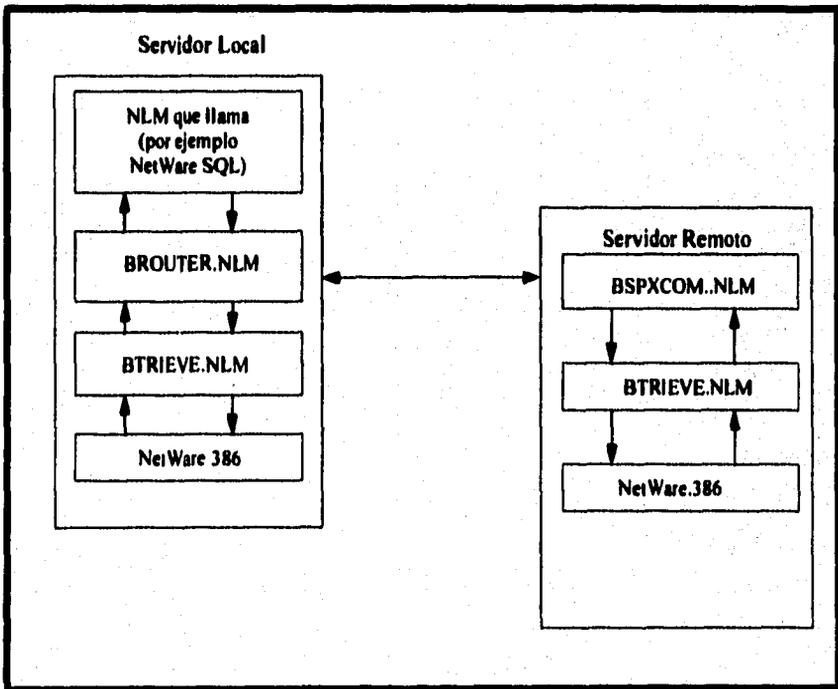


FIG. 1.6.11. Componentes de servidor para el procesamiento de llamadas remotas



Característica	SLIP	PROCOMM	DYNACOM	BTRIEVE	C
IP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CONEXION REMOTA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CONEXIÓN SERIAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SOFTWARE ADICIONAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FLEXIBILIDAD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PROPOSITO ESPECIFICO PARA EL SISTEMA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CONEXIÓN CON MAINFRAMES	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EMULACIÓN DE TERMINALES	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OTROS PROTOCOLOS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input checked="" type="checkbox"/> Si		<input type="checkbox"/> No	

FIG. 1.6.12. Sumario de Características del Software de Comunicación.

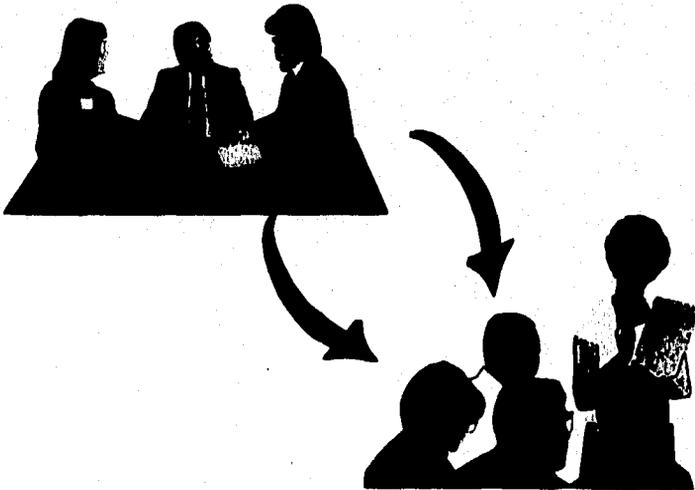


CONCLUSIÓN.

A pesar de todas las ventajas y características que nos proporcionan los programas de comunicación como SLIP, PROCOMM, BTRIEVE y DYNACOMM, observamos que es conveniente realizar un programa en C, el cuál será desarrollado específicamente con el fin que deseamos, es decir, que sea específico para realizar la comunicación entre el servidor y una Pc, la cual se encargará de llevar el monitoreo de las claves de acceso a Internet. Este programa realizado en C se encargará de conectar el programa de monitoreo con el servidor el cuál a su vez se encuentra conectado al conmutador y éste a la línea telefónica; de tal manera que se pueda enviar un cuestionario que será el encargado de validar dicha conexión, es decir, si éste cuestionario se contesta correctamente la clave será validada y continuara la sesión, en caso contrario se cancelará la sesión.

CAPITULO II

PLANTEAMIENTO DE LA PROBLEMÁTICA Y PROPUESTA DE SOLUCIÓN





2.1 PROBLEMÁTICA ACTUAL DEL USO Y SERVICIO QUE PRESTA INTERNET A LOS USUARIOS.

El desarrollo que ha tenido la computación gracias al uso de las redes de computadoras puede verse ampliamente reflejado en RedUNAM, red universitaria de propósitos académicos y de investigación, que nos da acceso a la red INTERNET. Por medio de RedUNAM nos podemos conectar a gran cantidad de computadoras localizadas en casi cualquier parte del mundo, sin embargo la diferencia en las formas de comunicación entre máquinas (protocolos) representa un problema. Para brindar conectividad de una PC que no forma parte de RedUNAM con las computadoras que están dentro de ella, se requiere de un MODEM (MODulador-DEModulador) y utilizar el protocolo TCP-IP¹ utilizando el software PROCOM o PROCOMPLUS para establecer una sesión, o si se desea además realizar transferencia de archivos se utiliza el protocolo SLIP, sin olvidar el uso de TCP-IP.

El sistema enfoca la problemática que enfrenta actualmente la Universidad Nacional Autónoma de México y en especial la DGSCA en cuanto a la saturación de sus líneas telefónicas dedicadas a brindar el servicio de conexión a Internet, esto debido al mal uso que le dan los usuarios a sus claves (principalmente al prestarlas) convirtiéndolas prácticamente en claves multiusuario, repercutiendo directamente en la economía de esta institución y en la calidad del servicio proporcionado.

Por ejemplo, si una persona en circunstancias actuales pretendiera conectarse desde su casa a través de su modem a Internet por medio de la salida de la UNAM, vería que es casi imposible lograrlo debido a la saturación de las líneas, lo cual es consecuencia en gran medida del mal uso antes mencionado aunado a esto que los equipos de comunicaciones han sido rebasados por las necesidades de los usuarios.

¹ Transport Control Protocol-Internet Protocol



Cabe hacer mención que en lo que respecta a este último punto, ya se han tomado las medidas pertinentes y se ha hecho la proyección de crecimiento con planeación a un año a partir de la fecha de este trabajo para cubrir en un 90% los requerimientos de los usuarios.

RedUNAM

Tal vez uno de los ejemplos más complejos de la forma en que se puede llegar a enlazar los equipos de cómputo utilizados en las universidades mexicanas lo brinda RedUNAM. Red de la Universidad Nacional Autónoma de México.

En octubre de 1985, la UNAM e IBM de México suscribieron un convenio con el cual se puso en marcha un proyecto conjunto de investigación y desarrollo en el que se contemplaba: La instalación de una red universitaria de cómputo de apoyo a la docencia que permitiera el acceso remoto a los sistemas de procesamiento de datos actuales y futuros en las dependencias de la UNAM y la creación de un laboratorio para el diseño y la manufactura apoyado por computadoras. Dicho convenio fue firmado por el rector de la UNAM, el presidente y director general de IBM y por el secretario de Comercio y Fomento Industrial. A raíz del convenio se integraron dos grupos de especialistas de la UNAM e IBM con objeto de definir detalladamente el plan de trabajo y así cumplir con los objetivos propuestos.

En el caso específico de la red, el planteamiento tecnológico original sufrió modificaciones a medida que se analizaron problemas similares en otras universidades y al conocimiento que se fue adquiriendo sobre comunicaciones, así como el avance tecnológico en el campo de las redes de computadoras. El esfuerzo de cuatro años concluyó finalmente en septiembre de 1989 con la inauguración oficial de RedUNAM por parte del rector José Sarukhán Kermez.

Una red homogénea de computadoras es aquella compuesta por máquinas del mismo tipo y con características similares. RedUNAM, por el contrario es una red heterogénea de cómputo integrada por equipos de distintos proveedores y con características variadas. La diversidad de la red dificulta la labor de enlazar los sistemas, pero permite utilizar de manera transparente todos



los recursos de cómputo ya existentes. Para lograr la comunicación del equipo se utilizan varios métodos de transmisión como *Token Ring*, *Ethernet* y *x.25*.

El corazón de la red es un sistema medular (*backbone*) de fibra óptica que transmite la información a una velocidad de 16 Mbps en un anillo (*token-ring*) que enlaza cuatro subanillos. El backbone tiene una longitud de 10 kms y se comunica con los subanillos mediante puentes (*bridges*) que sirven de enlace entre el anillo principal y los subanillos satélite. La información fluye a una velocidad de 4 Mbps, combinando señales ópticas de edificio a edificio y señales eléctricas dentro de ellos.

Como se mencionó anteriormente, la comunicación con los anillos satélite se realiza mediante puentes lógicos, los cuales son físicamente microcomputadoras con dos tarjetas de expansión para token-ring y en donde se ejecuta el programa IBM *Token-ring Network Bridge*.

La red *token-ring* cubre la zona de centros e institutos de investigación de la comunidad científica y la zona de las ingenierías como se observa en la figura. 2.1.1. Dentro de los subanillos, los diferentes edificios están conectados por fibra óptica a una velocidad de 4 Mbs. Se utilizó una fibra óptica para exteriores, para poder cablear conjuntamente la *token-ring* y la *Ethernet*.

Parecería que para el volumen de datos de una Universidad, el utilizar fibra óptica es un dispendio innecesario, mas no es así debido primordialmente a dos factores:

- La propiedad que posee la fibra óptica de no verse afectada por descargas eléctricas, lo que constituye en caso de la Ciudad Universitaria, un problema crítico, dadas las particularidades físicas del terreno. Este se encuentra en una zona volcánica que en algunos lugares supera los 20 metros de profundidad, lo que dificulta el logro de una conexión eléctrica adecuada, y
- La velocidad de transmisión que se requiere es alta, ya que se estimaba que para 1990 habría 1500 estaciones de trabajo conectadas a los diferentes subanillos.



El primer anillo satélite de la red está formado por la Dirección de Cómputo Académico, la Facultad de Ciencias, el Instituto de Investigaciones Nucleares y la División de Estudios de Posgrado de la Facultad de Contaduría y Administración. El puente de este anillo hacia el *backbone* se encuentra en la Dirección General de Servicios de Cómputo Académico.

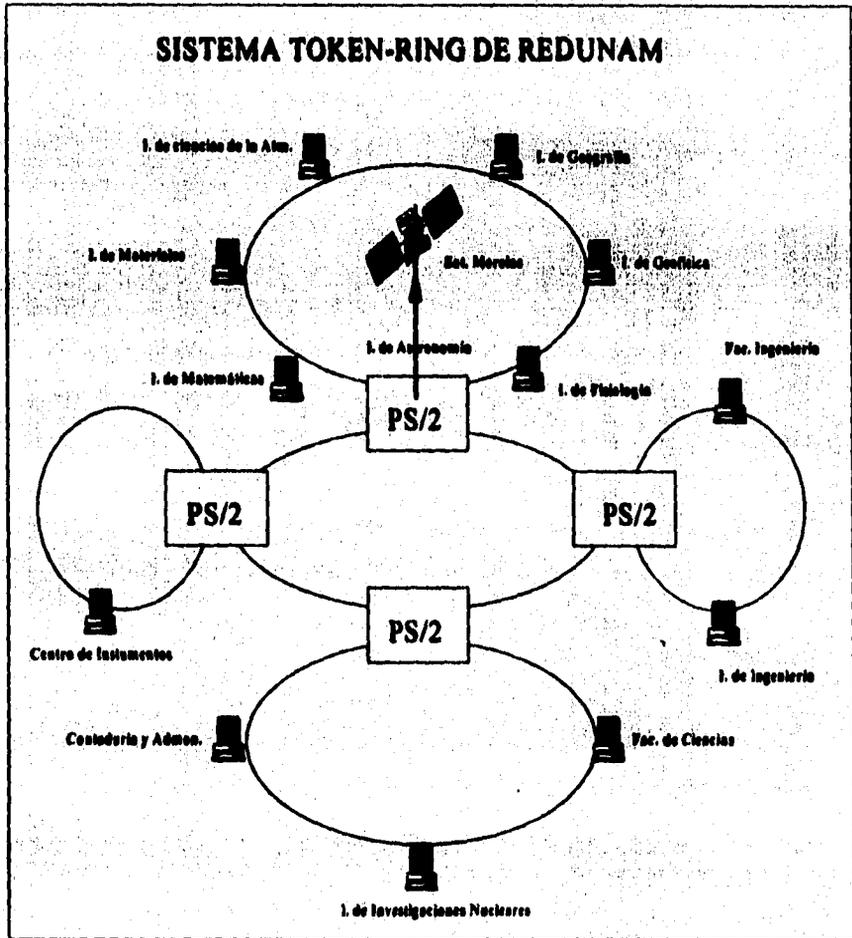


FIG. 2.1.1 Arquitectura de RedUNAM



El segundo anillo satélite está integrado por los institutos de Astronomía, de Matemáticas, de Materiales, de Ciencias de la Atmósfera, de Geografía, de Geofísica y de Fisiología Celular. La compuerta de este anillo se encuentra en el primero de estos.

El tercer anillo satélite está integrado por el Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas (IIMAS); la División de Estudios de Posgrado de la Facultad de Ingeniería, la Unidad de Servicios De Cómputo Académico de la Facultad de Ingeniería (UNICA) y el Instituto de Ingeniería. El puente de este anillo se encuentra en el IIMAS. Además del anillo y subanillos con base en el protocolo token-ring, se decidió instalar en la UNAM una red que utiliza el protocolo conocido como *Ethernet* y cuya principal función es enlazar los sistemas existentes del tipo digital, así como otras computadoras.

En RedUNAM también se utiliza el protocolo x.25, el cual permite conectar a la red a los equipos Hewlett Packard. Usuarios externos al campus de Ciudad Universitaria tienen acceso a la red a través de Telepac (servicios de transmisión de datos públicos de la Secretaría de Comunicaciones y Transportes).

La complejidad de RedUNAM es bastante grande. El número de usuarios aumenta día con día. En el caso de los usuarios de BITNET en la UNAM, éstos suman más de 700 personas.

La Red Internet

Esta es una red de área amplia (mundial), de conmutación de paquetes, que emplea el protocolo de comunicaciones TCP/IP (*Transmisión Control Protocol/Internet Protocol*).

A principios de los 70's, el Departamento de Defensa de los Estados Unidos (DOD) encargó a la Agencia de Proyectos e Investigación Avanzada (ARPA) el desarrollo de un conjunto de protocolos de comunicación, denominado TCP/IP. Inicialmente se conectaron 3 universidades y un centro de investigaciones (Universidad de California en Los Ángeles y Santa



Bárbara, la Universidad de Utah y el Instituto de Investigaciones de Stanford) empleando dicho protocolo para conformar la Red ARPAnet.

ARPAnet era una red experimental que apoyaba a la investigación militar, en particular la investigación sobre cómo construir redes que pudieran soportar fallas parciales (como las producidas por bombardeos) y aún así funcionar (piense en esto cuando se describa como trabaja la red; puede dar una buena idea del porqué del diseño de Internet). En el modelo ARPAnet, la comunicación ocurre entre una computadora fuente y una destino. La red asume por sí misma que es falible; cualquier parte de la red puede desaparecer en cualquier momento (por ejemplo los cortes de cables). La red fue diseñada para requerir un mínimo de información de las computadoras que forman parte de ellas. Para enviar un mensaje en la red, una computadora sólo tiene que poner la información en un *sobre*, llamado paquete de protocolo Internet (*IP: Internet Protocol*) y le asigna el domicilio destino en forma correcta. Las computadoras que se comunican -no la red- tienen la responsabilidad de asegurar que la comunicación se lleve a cabo. La filosofía era que cada computadora en la red se pudiese comunicar, como un elemento individual, con cualquier otra computadora.

Esta red (la antecesora de la actual Internet) creció en forma acelerada debido a que los programas fuentes del protocolo de comunicaciones fueron de dominio público, por lo que los diferentes fabricantes incorporaron TCP/IP a sus equipos. Uno de los principales objetivos de TCP/IP fue el poder interconectar equipo heterogéneo eliminando de esta manera la barrera impuesta por los protocolos propietarios tales como el de IBM (SNA), Digital (DECnet), Xerox (XNS), etc.

Los desarrolladores de Internet en Estados Unidos, el Reino Unido y Escandinavia, en respuesta a las presiones del mercado, empezaron a poner el Software de IP en todo tipo de computadoras: se llegó a convertir en el único método práctico para comunicar computadoras de diferentes fabricantes. Esto resultó muy práctico para el gobierno y las universidades, quienes no tenían políticas que especificaban la compra de una determinada marca de computadoras. Todos



compraron la computadora que mejor les pareció y esperaban poder comunicarse con otras en la red.

Al mismo tiempo que Internet se consolidaba, las redes locales *Ethernet* eran desarrolladas. La tecnología de redes locales maduró hasta 1983, cuando aparecieron las primeras estaciones de trabajo para escritorio y las redes locales se multiplicaron. La mayor parte de las estaciones de trabajo tenían el sistema UNIX de Berkeley instalado, que incluía el software de red IP. Esto creó una nueva demanda: en lugar de conectar una computadora de tiempo compartido en un centro de cómputo, las organizaciones requerían conectar toda su red local a ARPANet, lo cual permitiría que todas las computadoras que estuviesen en la red usaran los servicios de ARPANet. Al mismo tiempo, muchas compañías y otras organizaciones empezaron a construir redes privadas usando los mismos protocolos de comunicación de ARPANet, es decir, IP y sus protocolos asociados. Parecía obvio que si estas redes podían comunicarse entre sí, los usuarios de una red podrían comunicarse con usuarios de otra y todo el mundo sería beneficiado.

La Red Internet está compuesta actualmente por entidades autónomas que van desde universidades, centros de investigación, oficinas gubernamentales, grandes corporativos, etc., por lo tanto, la administración y las políticas de tráfico difieren grandemente entre dichas redes; algunas permiten el libre paso del tráfico mientras que otras lo prohíben proporcionando servicios de red restringidos a sus usuarios.

La demanda sigue creciendo ahora que la mayoría de las universidades están conectadas a la Internet, se está tratando de incluir también a primarias y secundarias, al igual que a las bibliotecas locales. Las personas que recientemente se graduaron en una Universidad saben para qué sirve Internet y hablan con sus compañeros de trabajo acerca de conectar la empresa en que laboran a dicha red.

Toda esta actividad apunta a un crecimiento continuo, a la solución de los problemas de conectividad, a la evolución de las tecnologías y a la seguridad en el empleo de los expertos en



comunicaciones. Mucha gente va más allá; pues una vez que se cuenta con una conexión de red en el trabajo, el siguiente paso lógico es querer conectarse directamente desde la casa.

-Cómo trabaja Internet?

El sistema de redes moderno está constituido sobre el concepto de "niveles o capas de servicio". Usted comienza tratando de mover bits de un lugar a otro, perdiendo algunos en el trayecto. Este nivel se compone de cables y *hardware*, y no necesariamente de cables muy contables. Después agrega una capa de *software* básico que permite aislar los problemas de hardware. Incorpora otra capa de software para darle al básico algunas características deseadas. Continúa agregando funcionalidad e inteligencia a la red, capa por capa, hasta que se obtiene algo amigable y útil.

El Modelo TCP-IP

Una red de cómputo está formada por un conjunto de computadoras interconectadas para compartir recursos e intercambiar información. Con el fin de permitir que distintas arquitecturas diseñadas por diferentes fabricantes puedan interactuar entre sí, la Organización de Estándares Internacionales (ISO, por sus siglas en inglés) inició en 1977 la definición de un modelo de referencia llamado el *Modelo de Interconexión de Sistemas Abiertos* (o modelo OSI), que divide la arquitectura de red en 7 capas, cada una con funciones específicas independientes de las demás.

En este modelo, la capa N en una computadora realiza sus funciones comunicándose con la capa del mismo nivel en la otra computadora. Esta comunicación se lleva a cabo a través de reglas bien definidas *-protocolos-* para una arquitectura de red en particular.

Las funciones de la capa N sirven para que ésta pueda ofrecer un *servicio* a la capa inmediatamente superior en la misma computadora. De esta manera se van agregando servicios conforme se asciende por las capas de la arquitectura hasta llegar a la capa de aplicación. El



modelo OSI no define protocolos ni servicios, éstos dependen de la implantación específica a cada arquitectura de red. Lo que el modelo define es la función que debe realizar cada capa.

El modelo de red definido por el conjunto de protocolos TCP/IP es mucho más simple que el definido por otras arquitecturas de red.

En particular TCP/IP sólo define 3 capas del modelo OSI:



Llama la atención la ausencia de protocolos en las capas de sesión y presentación. Lo que ocurre es que TCP/IP se definió antes que el modelo OSI, y en un ambiente donde la necesidad de los servicios ofrecidos por estas dos capas no era evidente.

Redes de conmutación de paquetes

Cuando uno trata de imaginar qué es Internet y cómo opera, es normal pensar en un sistema telefónico. Después de todo, ambos son electrónicos y permiten abrir una conexión y transferir información, e Internet está compuesta principalmente por líneas telefónicas permanentemente dedicadas a este uso. Desafortunadamente, esto crea una idea errónea y provoca mucha confusión sobre la forma en que opera Internet. La red telefónica es una red de conmutación de circuitos. Cuando usted habla por teléfono, se separa una parte de la red para dedicarla a atender su llamada. Aún cuando no esté utilizando su parte de la red (por ejemplo cuando la línea está en espera), ésta es inaccesible para otras personas; lo que provoca una subutilización de un recurso muy costoso: la red.

Un mejor modelo para comparar Internet es el Servicio Postal. El Servicio Postal es una red de conmutación de paquetes. Usted no cuenta con una parte de la red dedicada a sus



actividades. Lo que se quiere enviar se mezcla con los mensajes de otras personas, se pone en un conducto, se transfiere a otra oficina postal y se clasifica todo nuevamente. Aunque las tecnologías son completamente diferentes, el Servicio Postal muestra una analogía sorprendentemente similar.

El Protocolo Internet, IP

Un cable puede llevar información de un lugar a otro. Sin embargo, ya se sabe que Internet puede hacer que la información llegue a distintos lugares distribuidos en todo el mundo. ¿Cómo es esto?

Las diferentes partes de Internet están conectadas por un conjunto de computadoras llamadas *enrutadores*, que interconectan las redes. Estas redes pueden ser *Ethernet*, *token ring* o en ocasiones líneas telefónicas.

Las líneas telefónicas y las redes *Ethernet* son equivalentes a los camiones y aviones del Servicio Postal. Son el medio a través del cual el correo va de un lugar a otro. Los enrutadores son sucursales postales; estos equipos deciden cómo dirigir la información (paquetes), de la misma forma que una oficina postal decide cómo distribuir los sobres por correo. No toda subestación o todo enrutador cuentan con una conexión a cada uno de los otros enrutadores de la red. Si se envía un sobre de correo desde Dixville Notch, New Hampshire, con destino a Boonville, California, la Oficina Postal no reserva un avión de New Hampshire a California para llevarlo, sino que envía el sobre a la sucursal de correo y ésta a su vez lo envía a otra, y así sucesivamente hasta alcanzar su destino final.

Esto significa que cada subestación solo necesita conocer las conexiones con las que cuenta y cuál es el mejor "siguiente salto" para acercar el paquete a su destino. Internet trabaja de manera similar: un ruteador se fija en el destino de la información y decide a dónde enviarla. El enrutador elige cuál es el enlace más apropiado para enviar la información.



¿Cómo sabe la red a dónde se dirige la información? Si se quiere enviar una carta, no basta con poner el papel escrito en el buzón y esperar a que sea entregado. Es necesario poner el papel con la información en un sobre, escribir el domicilio del destinatario y pegar los timbres postales. De la misma manera que la Oficina Postal tiene reglas que definen la operación de su red, también existen reglas que definen la operación de Internet. Las reglas son llamadas protocolos. El Protocolo Internet (IP) se hace cargo de establecer domicilios o se asegura de que los enrutadores sepan qué hacer con la información que les llega. Continuando con la analogía de la Oficina Postal, el Protocolo Internet trabaja justo como un sobre, ver figura 2.1.2.

Una parte de la información del domicilio va al principio del mensaje; estos datos dan a la red información suficiente para hacer la entrega del paquete.

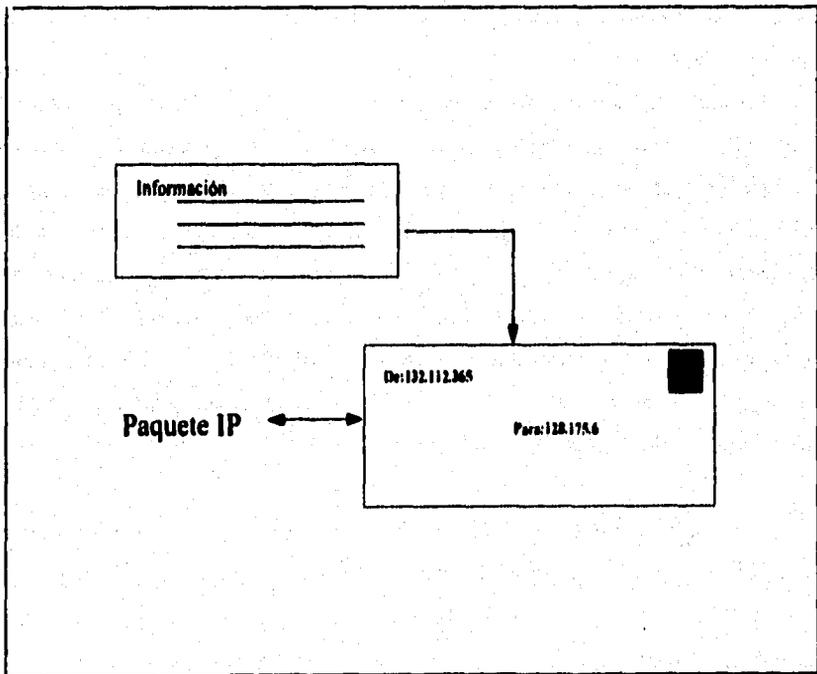


FIG. 2.1.2 Sobres IP



Los domicilios de Internet constan de cuatro números, cada uno menor de 256. Cuando dichos números se escriben se separan por puntos, como se muestra a continuación:

132.248.27.10

128.150.69.90

No debemos preocuparnos por memorizar estos números para poder utilizar la red, en efecto, el domicilio está compuesto por varias partes. Como Internet es una red de redes, los primeros números del domicilio indican a los enrutadores cuál es la red a la que usted pertenece. Los últimos números indican qué computadora personal o equipo anfitrión de la red debe recibir el paquete, bajo este esquema cada computadora de Internet tiene un domicilio único. De nuevo el Servicio Postal es una buena analogía. Considere el domicilio "Del cabo Num 21, Naucalpan, Edo.Mex."

La parte "Naucalpan, Edo. Mex." es como la parte del domicilio correspondiente a la red, el cual permite al sobre llegar a la Oficina correcta, que es la que tiene la información acerca de las calles en un área determinada. La parte "Del cabo Num.21" es como el domicilio del equipo anfitrión; éste identifica a un buzón particular en el área de servicio de la Oficina Postal. La Oficina Postal concluye su trabajo cuando entrega el correo a la oficina local correcta y ésta lo pone en el buzón correcto. De la misma forma, Internet concluye su trabajo cuando los enrutadores llevan la información a la red local correcta y ésta entrega dicha información a la computadora personal o equipo anfitrión correctos, localizados en dicha red.

Por muchas razones prácticas (sobre todo por limitaciones de *hardware*), la información enviada a través de las redes IP se divide en pedazos de tamaño distinto, llamados paquetes. La cantidad de información en un paquete normalmente se encuentra entre 1 y aproximadamente 1500 caracteres de largo. Esto previene que cualquier usuario monopolice la red, permitiendo que todos tengan un acceso equitativo. También significa que cuando la red se sobrecarga, su comportamiento sólo desmerece un poco para todos los usuarios: la red no se inutiliza cuando algunos usuarios la monopolizan.



Una de las propiedades más impresionantes de Internet es que, en un nivel básico, el protocolo IP es todo lo que se necesita para participar en la red.

No obstante, habrá que resolver varios problemas:

- La mayoría de las transferencias de información es mayor que 1500 caracteres.
- En ocasiones se presentan errores: ocasionalmente los paquetes pueden ser extraviados, pero esto se resuelve exitosamente.
- Los paquetes pueden llegar en desorden.

Para evitar esto, la siguiente capa de la red nos permitirá enviar grandes cantidades de información y corregirá las alteraciones que puedan ser causadas por la red.

El protocolo de control de transmisión (TCP)

TCP es el protocolo que se menciona frecuentemente junto con el IP y que se utiliza para resolver los problemas mencionados. El protocolo TCP toma la información que se desea enviar y la divide en segmentos. Además, enumera cada segmento para que el receptor pueda verificar la información y ponerla en el orden adecuado. Para que el protocolo TCP pueda enviar esta secuencia de números a través de la red, cuenta con su propio sobre que le permite "escribir" en ella información requerida para su reordenamiento. Un segmento de la información a transmitir se coloca en el sobre del protocolo TCP. Este sobre es puesto, a su vez, dentro del sobre del protocolo IP y posteriormente es transmitido a la red. Una vez que se pone algo en un sobre IP, la red lo puede transmitir.

Del lado del destinatario, una parte del *software* de TCP reúne los sobres, extrae la información de ellos y la pone en el orden adecuado. Si algún sobre se pierde en la transmisión, el receptor solicita su retransmisión al emisor. Una vez que el protocolo TCP tiene toda la información en el orden adecuado, la pasa a la aplicación del programa que esté utilizando sus servicios.



La descripción anterior del funcionamiento del protocolo TCP es ligeramente utópica. En la realidad, los paquetes no sólo se pierden, además de esto pueden ser modificados por el mal funcionamiento durante la transmisión a través del medio. TCP también resuelve este tipo de problemas. Así como coloca la información en un sobre, el protocolo calcula algo llamado número de verificación (*checksum*). Ese número permite que el receptor TCP detecte errores en el paquete transmitido.

Cuando un paquete llega a su destino, el receptor calcula el número de verificación y lo compara con el enviado por el transmisor. Si no coinciden, significa que ocurrió un error en la transmisión. El receptor deshecha el paquete y solicita la retransmisión.

Servicios disponibles

En Internet existen muchos recursos disponibles, pero no hay una lista oficial. Cualquiera que tenga una conexión a Internet puede poner en línea un recurso nuevo en cualquier momento sin avisar a nadie, y de hecho así ocurre. De manera que el truco es encontrar qué está disponible. En esta sección se tratarán algunos recursos conocidos que son considerados muy especiales. Las listas de los mismos son muy útiles, pero no dicen la última palabra. Nunca están al día, siempre les faltan los avances más recientes y novedosos de la red.

Otro problema es que la calidad de recursos de la red varía enormemente. Mientras que las listas tratan de enfocarse sobre todo a los "mejores" recursos que se pueden encontrar, también es cierto que lo que es bueno para unos no lo es para otros. Para poder llegar a emplear correctamente la Red, el usuario debe aprender a escoger lo que realmente es útil entre todo lo que está disponible. Entre las herramientas más importantes con las que cuenta Internet se encuentran:

telnet: Se utiliza para establecer sesiones de trabajo en las computadoras de Internet y para tener acceso a muchos servicios públicos, que incluyen catálogos de bibliotecas y otros tipos de bases de datos.



ftp: Sirve para transferir archivos de un lugar a otro. Su mayor uso está en la recuperación de archivos de depósitos públicos localizados en toda Internet. A estos depósitos se les llama FTP anónimos debido a que no es necesario contar con una clave de usuario para poder tener acceso a ellos.

Correo electrónico. Permite enviar mensajes desde y a cualquier parte de la Red.

USENET News: Le permite leer y colocar mensajes que han sido enviados a los "grupos de interés" públicos. Esto puede sonar no muy claro, pero realmente se refiere a los tableros de foros o grupos de discusión. USENET es el servicio de tableros de discusión más grande del mundo.

TELNET

Telnet es el protocolo de sesión de trabajo remota de Internet. Le permite estar frente al teclado de una computadora y establecer una sesión en una computadora remota en la red (ver figura.2.1.3) *Telnet* ofrece sus servicios sobre el protocolo de transporte TCP. La sesión puede ser en una máquina en la misma oficina, en la misma universidad, o al otro lado del mundo. Cuando usted se conecta es como si su teclado estuviese conectado a esa computadora remota. Podrá tener acceso a todos los servicios que esa máquina provee a sus terminales locales. Puede realizar una sesión interactiva normal (conectándose y tecleando comandos) o tal vez tener acceso a muchos servicios especiales, como buscar en algún catálogo de biblioteca, saber que está pasando en teoría, tener acceso al texto del periódico *USA Today* y aprovechar muchos de los servicios disponibles en los diferentes equipos de la red. La forma más sencilla de usar telnet es teclear:

```
telnet domicilio-internet-de-computadora remota
```

en el nivel de comandos. Cuando termina la sesión en el sistema remoto, telnet termina su ejecución. Cualquier comando que se dé será ejecutado por el sistema local.



¿Que ocurre realmente?

Se hará una revisión más profunda para ver que es lo que pasa cuando se inicia una sesión de TELNET. Una aplicación consta de dos partes de *software* que colaboran entre sí: el *cliente*, que corre en la computadora que solicita el servicio, y el *servidor* que corre en la computadora que provee el servicio. La red es el medio que permite la comunicación entre ambos.

```

*****
De igual manera si tiene cualquier problema de acceso a su clave
puede comunicarse a los mismos telefonos
*****
Desearnos tenga una agradable sesion!!!
*****

login: escar
Password:
Your password will expire in 4 days
Last login: Tue Feb 28 13:33:29 from opl.dcaa.unam.mx
Sun Microsystems Inc. SunOS 5.4 Generic July 1994

*****

Se los comunica a los usuarios que utilizan el directorio
"/transferencias" que por el momento no podran accederlo porque no
esta en servicio. De manera temporal estaran accedendo a otro disco.

*****
Tienes nuevos mensajes en tu correo.
tzetzal 1:>
tzetzal 1:>
tzetzal 1:>

```

FIG.2.1.3. Sesión de telnet desde una PC.

El cliente, que es el programa que corre en su computadora cuando se tecléa el comando telnet, deberá hacer lo siguiente:

- Crear una conexión de red TCP con un servidor.



- Aceptar lo que se escriba en el teclado proveniente del usuario, de manera conveniente.
- Reformatear la entrada de datos a un formato estándar y enviarla al servidor.
- Aceptar la respuesta del servidor en un formato estándar.

Reformatear esa salida para desplegarla en la pantalla del usuario.

El software del servidor corre en la máquina que provee el servicio; si el servidor no está funcionando, el servicio no está disponible. En los sistemas UNIX, los servidores son conocidos como *daemons*, procesos que corren residentes en memoria todo el tiempo. Estos *auxiliares silenciosos* esperan a que sus servicios sean requeridos y cuando esto sucede comienzan a trabajar. Cuando un servidor típico está listo para aceptar solicitudes de servicio, realiza lo siguiente:

- Informa al software de red que está listo para aceptar conexiones.
- Espera una solicitud en un formato estándar.
- Atiende la solicitud.
- Envía los resultados de regreso al cliente en un formato estándar.
- Vuelve a entrar en el proceso de espera.

Transferencia de archivos: FTP

Muy comúnmente será posible encontrar información en Internet que no querrá examinar mientras se encuentra conectado a algún sistema remoto: sería muy bueno contar con una copia para usted. En muchos de los casos, es necesario trasladar una copia del archivo a su computadora para poder trabajar con ella ahí. La herramienta para lograr esto es *ftp* (ver figura .2.1.4).

Este comando recibe el nombre de *ftp* debido al protocolo de aplicación que usa: el Protocolo de Transferencia de Archivos (FTP: *File transfer Protocol*). Como su nombre lo



indica, la función del protocolo es transferir archivos de una computadora a otra, no importando dónde se localicen, cómo estén conectadas o si tienen o no el mismo sistema operativo. Dado que ambas computadoras "hablan" el protocolo FTP y tienen acceso a Internet, es posible utilizar el comando `ftp` para transferir archivos. Algunas de las características de su uso cambian con cada sistema operativo, pero la estructura básica de comandos es la misma en cualquier máquina.

```
tztetzal 9:> ftp ftp.ncafes.com
Connected to ftphost.ncafes.com.
220 ftphost FTP server (Version wu-2.4(1) Mon May 29 18:00:56 PDT 1995) ready.
Name (ftp.ncafes.com:oscar): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-The response 'ocartztetzal.dcaa.unam.mx' is not valid
230-Next time please use your e-mail address as your password
230- for example: joe@tztetzal.dcaa.unam.mx
230 Guest login ok, access restrictions apply.
ftp>
```

FIG. 2.1.4 Transferencia de Archivos FTP

Al igual que *telnet*, *ftp* ha provocado la proliferación de una vasta gama de bases de datos y servicios. De hecho, usted puede además encontrar cualquier cosa, desde opiniones legales y recetas, hasta software *gratis* en una gran cantidad de bases de datos en líneas disponibles, o depósitos, a los que usted puede tener acceso mediante *ftp*.

Ftp es un programa complejo porque tiene muchas formas diferentes de manejar archivos a estructuras de archivos. Las diversas formas de guardar archivos (en binario o ASCII, comprimidos o sin comprimir, etc.) introducen algunas complicaciones y se requiere de



conocimientos adicionales para hacer las cosas correctamente. En seguida se discutirán algunos aspectos referentes al tipo de transferencia y el FTP anónimo, el cual es un servicio especial que permite el acceso a bases de datos públicas sin necesidad de obtener una cuenta.

Transferencias ASCII y Binarias

El comando FTP tiene dos formas comunes de transferir datos, llamadas *binaria* y *ASCII*. En una transferencia binaria se preserva la secuencia de bits del archivo, de tal forma que el original y la copia del archivo sean idénticas bit por bit, aunque el archivo contenga una secuencia de bits que no tenga ningún significado en la máquina receptora.

El concepto del modo ASCII realmente no es apropiado: debería llamarse modo *texto*. En el modo ASCII las transferencias son tratadas como un conjunto de caracteres, el cliente y el servidor tratan de asegurar que los caracteres que se transfieren tengan el mismo significado tanto en la máquina receptora como en la emisora.

La gran ayuda de transferir en ASCII es porque se desconoce de que tipo de máquina se están tomando y realmente no importa, de lo único que hay que asegurarse es de poder leer los archivos en la máquina local. Para hacer que FTP trabaje en modo ASCII, se debe teclear el comando *ascii*. Para cambiar a modo binario, se debe teclear el comando *binary*.

Es importante saber el tipo de información que se desea transferir. La tabla 2 proporciona algunos tipos comunes de archivos. Los archivos ejecutables generalmente son binarios, sin embargo, existen algunas excepciones. Los programas que son compilados y ejecutados por el mismo procesador siempre son binarios.



UNIX provee varios lenguajes (incluyendo al Shell) con los cuales es muy común escribir programas con comandos básicos del sistema operativo, regularmente estos programas se almacenan en un archivo siendo de tipo texto.

En UNIX, es posible usar el comando *file* para saber el tipo de la mayoría de los archivos. Esta utilidad no fue escrita tomando en cuenta a FTP, por lo que proporciona más información de la que realmente necesita.

Archivo de texto	ASCII por definición
Hoja de Cálculo	Probablemente binario
Archivo de base de datos	Probablemente binario, Posiblemente ASCII
Archivo de procesador de palabras	Probablemente binario, posiblemente ASCII
Programa en código fuente	ASCII
Mensajes de correo electrónico.	ASCII
Archivos encapsulados UNIX	ASCII
Archivos tar de UNIX	Binario
Archivo de respaldo	Binario
Archivo comprimido	Binario
Archivo en formato <i>unecode</i> ó <i>binhexed</i>	ASCII
Archivo ejecutable	Binario
archivo PostScript (impresora láser)	ASCII
Documento en Hipertexto (HTML)	ASCII
Archivos de imagen (GIF, JPEG, MPEG)	Binario

Tabla 1



FTP Anónimo

Con FTP anónimo no se tiene la necesidad de tener una clave de usuario o una contraseña para poder tener acceso a archivos en una máquina. Obviamente existen algunas restricciones: normalmente los usuarios anónimos sólo pueden ejecutar el comando *get* de los archivos (esto es, copiarlos), no podrán instalar archivos nuevos o modificar archivos que ya existen, lo que sería equivalente a ejecutar el comando *put*. Y existen límites estrictos sobre que archivos se pueden copiar.

Cuando se habilita un servidor FTP anónimo, se crea una clave de usuario especial llamada *anonymous* (ver figura.2.1.5). Si usted inicia FTP, establece una sesión en una computadora remota y da la clave de usuario *anonymous*, FTP aceptará cualquier conjunto de caracteres como contraseña. Se considera de buen gusto usar como contraseña su domicilio de correo electrónico, para que los administradores del servidor puedan comunicarse con usted en caso necesario (de hecho algunos sistemas empiezan a demandar que se proporcione un domicilio de correo electrónico válido para permitir el acceso).

```
tzetzal 9:~> ftp ftp.mcafee.com
Connected to ftp.mcafee.com.
220 ftp.mcafee.com FTP server (Version wu-2.4(1) Mon May 29 18:00:56 PDT 1995) ready.
Name (ftp.mcafee.com:oscar): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-The response 'oscartzetzal.dcaa.unam.mx' is not valid
230-Next time please use your e-mail address as your password
230-      for example: joe@tzetzal.dcaa.unam.mx
230 Guest login ok, access restrictions apply.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
pub
incoming
bin
etc
licensed
226 Transfer complete.
35 bytes received in 0.0042 seconds (8.1 Kbytes/s)
ftp>
```

INUI228 Page 11 Tue 11/29/95 12:22:10 C:\ Rep 12:15 N

FIG.2.1.5 FTP anónimo



Después de haber establecido la conexión mediante la clave de usuario *anonymous*, podrá obtener con el comando *get* aquellos archivos que estén permitidos expresamente a los usuarios de los servidores FTP anónimo. Una vez que se ha inicializado la sesión de FTP es posible cambiarse a otros subdirectorios dando el nombre o moverse al directorio "padre" usando el argumento *(..)*, y el uso de varios comandos más.

Correo Electrónico

El correo electrónico difiere de las otras aplicaciones porque no es un servicio de usuario a usuario: no es necesario que las máquinas emisora y receptora del correo electrónico se comuniquen directamente entre sí. Al correo electrónico se le conoce como un *servicio de almacenaje y reenvío* (ver figura 2.1.6 y bis.). El correo pasa de una máquina a otra hasta que llega a su destino final.

El Servicio Postal opera como una red de *almacenaje y reenvío*. Se escribe el domicilio en el sobre y se deposita en el buzón. La carta es recogida por un camión para llevarla a otro lugar y almacenarla. Ahí se clasifica y se reenvía a otro lugar.

Este paso se repite hasta que llega a su buzón destino. Si el buzón destino no está en el área de cobertura del Servicio Postal, el mensaje se envía al servicio postal del país destino.

A partir de esta analogía es posible inferir un par de cosas sobre Internet. Primero, si pone el domicilio correctamente a un mensaje, la red se hará cargo de entregarlo. No se necesita estar muy al tanto de lo que pasa. Es posible inferir también que el correo puede viajar entre Internet y otras redes de correo. Esto es cierto, pero el domicilio se hace más complejo si se requiere que un mensaje vaya de alguna red externa a Internet y viceversa.

Justo como en el Servicio Postal, si el emisor y el receptor no están conectados en la misma red, es necesario colocar el mensaje en algún lugar en donde se concentre todo el correo



que vaya a una determinada red. Los puntos de conexión entre redes de correo electrónico se denominan *puertas de aplicación*. Se les llama *puertas* porque pueden ser vistos como puertas mágicas entre dos mundos; son *puertas de aplicación* porque conocen las aplicaciones de correo electrónico de ambos lados, para que los mensajes sean reformateados a una forma congruente al pasar de una red a otra. Para enviar correo a través de una puerta, casi siempre es necesario proporcionar en el domicilio del mensaje información sobre cómo llegar a la puerta e información sobre como entregar el correo en la otra red.

```

TNVT220

?  HELP          - Get help using Pine
C  COMPOSE MESSAGE - Compose and send a message
I  FOLDER INDEX  - View messages in current folder

A  ADDRESS BOOK  - Update address book
S  SETUP         - Configure or update Pine
Q  QUIT         - Exit the Pine program

Copyright 1989-1994. PINE is a trademark of the University of Washington.

Help      PreuCmd  RelNotes
OTHER CMDS [ListFldrs] NextCmd  KILock

```

FIG 2.1.6 Ventana de MAIL (Correo Electrónico, a través del visualizador PINE).

Listas de distribución de correo

Con el correo electrónico, existe la misma facilidad de enviar un mensaje tanto a un grupo como a una persona. La herramienta para poder hacer eso se llama *lista de distribución de correo*. Permite que un alias o un sobrenombre represente a un grupo de destinatarios; por ejemplo, el



alias *staff* puede definir a un grupo al que pertenecen *todas los empleados* de un negocio. Cuando se envía un mensaje a *staff*, el correo se distribuye a todos los que pertenecen al grupo.

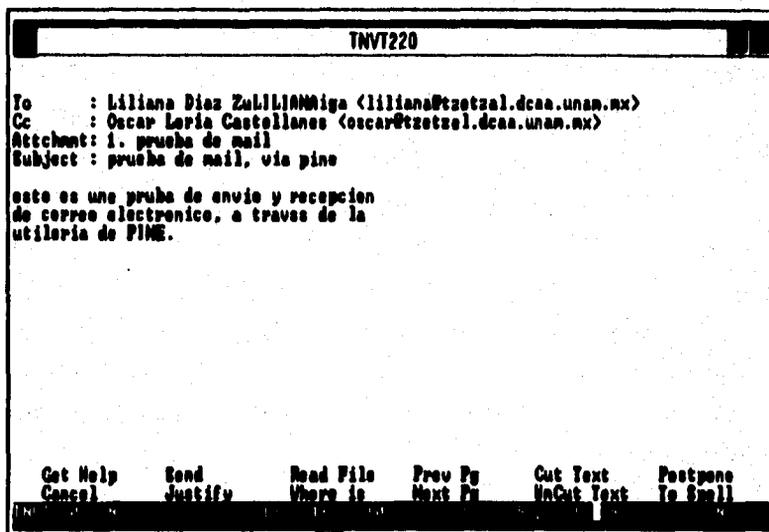


FIG. 2.1.6 bis Ventana de trabajo de PINE.

Una vez que el alias es creado, es posible enviar un mensaje al grupo el cual será entregado a sus miembros. Este es un medio para implantar grupos de discusión a través del correo electrónico. Funciona bien para grupos pequeños o para grupos personales que solo son usados por un usuario en particular. Al crecer el grupo y cuando existen otras personas que quieren usar la misma definición de grupo, la administración del grupo se convierte en algo muy complicado.

Cada vez que se incorpora o se elimina a alguien del grupo, todos los miembros de éste tendrán que modificar su alias personal. Esto no se lleva a cabo por todos, así que casi siempre alguien queda fuera del grupo y no recibe el mensaje y aquí es donde empiezan los problemas.

Lo que realmente se necesita es mantener una lista de correo en forma centralizada, para que cuando se realicen cambios surtan efecto en todos los miembros. Es común implantar esto



con un *reflector de correo*. Un reflector de correo es un domicilio especial de correo electrónico configurado para que, cualquier mensaje enviado a él, automáticamente sea reenviado a todos los participantes de la lista. El reflector de correo funciona bien para un grupo privado, que puede ser grande. Pero ¿Qué pasa si en lugar de ser una lista privada, existiera la posibilidad de que cualquier persona pudiera ingresar a la lista de discusión?. Si éste es el caso; el usuario que desea participar en la lista de discusión debe de hacer una petición, donde él indica lo siguiente: *Por favor, incliyame en la lista*. Enviar un mensaje como éste a la lista no es una solución razonable, porque la lista enviará el mensaje a todos los integrantes. Hacer esto puede funcionar, pero se considera de mal gusto, porque hará que el usuario sea considerado impertinente entre los integrantes de la lista, y peor aún, el administrador de la lista podría no recibir su mensaje. La persona que mantiene la lista puede ser al administrador de algún sistema de correo al que no le importa el tema de discusión. Por desgracia, la forma correcta de suscribirse a una lista depende de cómo sea administrada. Por tradición, Internet utiliza comúnmente domicilios especiales para solicitudes de este tipo.

Siempre que se crea un reflector de correo público, también se crea un segundo buzón de correo en la misma máquina. Este buzón de correo especial es *cerrado*, nada de lo que se recibe ahí es distribuido sino que se envía a la persona que mantiene la lista.

NetworkNews

Network News brinda servicio a través de servidores de noticias, es el equivalente en Internet a los grupos de discusión. *Network News* representa un medio para tomar parte en muchas pláticas más, pero manteniéndolas organizadas y separadas de su correo electrónico. Los artículos de *News* tienen otra ventaja: son ideales para visualizar y navegar entre ellos y no es necesario que se comprometa profundamente. Para el usuario, dentro de *Network News* se organizan pláticas bajo un conjunto de amplios apartados que son conocidos como grupos de interés (*newsgroups*). Un programa despachador de artículos se encarga de presentar tales pláticas de manera ordenada. Mantiene el rastreo sobre los temas que ya vio y solo muestra los temas nuevos que llegaron desde la última sesión. Una vez que el despachador de artículos ha



mostrado que artículos están disponibles sobre determinado tema, puede seleccionar y leer los temas que son de interés.

Archie

Históricamente, uno de los grandes problemas de Internet ha sido encontrar información en la red. Los servidores FTP anónimo surgieron rápidamente para brindar la oportunidad de buscar archivos en la Red. Dentro de ella podemos encontrar cualquier cantidad de información e inclusive software o programas de dominio público (recuerde que se denomina "de dominio público" al software que es ofrecido a la comunidad en general para su uso libre no comercial, respetando la autoría del mismo), pero es un poco difícil. Suena como un trabajo solo para computadoras, pero se puede resolver con *Archie*, un sistema que permite explorar índices disponibles en los servidores públicos especiales. *Archie* es un servicio que surgió en la Universidad McGill, primeramente de la manera más sencilla. Se fue buscando por toda la red y preguntando a quienes estuvieran ejecutando programas servidores FTP anónimo para catalogarlos. Los perpetradores ejecutan su programa una vez al mes, el cual se conecta con los servidores mencionados a través de FTP. Cuando se enlaza con los servidores, construye un directorio para listar todos los archivos que se encuentran en cada uno de ellos, usando comandos de FTP estándar. Al realizar una búsqueda, *Archie* explora los directorios catalogados y devuelve los nombres de los archivos que concuerdan con la cadena proporcionada, junto con los nombres de los servidores que contienen disponibles tales archivos. Así fue creado el servicio básico de *Archie*.

Un problema al que se enfrentaba este servicio es que había ocasiones en que se realizaban búsquedas de cadenas muy largas o palabras muy extrañas, es por eso que los creadores de *Archie* pidieron a las personas responsables de cada servidor que les enviaran la información sobre los paquetes más importantes que manejaran, y utilizaron esta información para crear un servicio llamado *whatts*. Se trata de un conjunto de palabras clave alternativas y catalogadas para archivos de la red, que pueden ser utilizadas para localizar *software* o archivos



de datos, incluso si el nombre del archivo no guarda ninguna semejanza con su contenido. Cuando aumentó el uso de *Archie*, el servicio se transformó para satisfacer la demanda que iba en aumento. En la actualidad, existen muchos servidores *Archie* repartidos por toda la Internet. Cada servidor construye un índice de archivos FTP cerca de él y después los diversos servidores comparten información. Esto permite que las actualizaciones sean más oportunas, sin sobrecargar la red. Con esto se debe empezar si se está buscando programas, datos o archivos de texto. Actualmente, *Archie* cataloga cerca de 1200 servidores y 2.5 millones de archivos. Como usuario, se pueden solicitar que se encuentren nombres de archivo que correspondan a ciertos criterios de búsqueda o que se muestren archivos que contengan ciertas palabras. *Archie* devuelve los nombres de archivo que concuerdan con el criterio de búsqueda y el nombre de los servidores que contienen esos archivos. Una vez que se decida cuál de esos archivos satisface mejor las necesidades, puede ser trasladado a la computadora local con FTP anónimo.

Gopher

Gopher es un servicio que permite buscar recursos utilizando menús (ver figura .2.1.7). El nombre "*Gopher*" (tuza) comenzó como un servicio de distribución interna en el campus de la Universidad de Minnesota, hogar de los "*Golden Gophers*". El nombre *gopher* fue acuñado debido a que, al igual que las tuzas, su principal función consiste en ir por ("*go per*") cosas. El servicio fue diseñado para que cada parte de la burocracia pudiera tener control sobre su propio servidor y sus propios datos.

Es decir, que la administración de la escuela pudiera contar con una computadora en su edificio administrativo para distribuir información del personal administrativo; que el departamento deportivo pudiera tener un servidor para la agenda deportiva en sus oficinas; que cada departamento académico contara con un servidor para proporcionar los horarios de clases, etc. Podrían existir tantos servidores como grupos que quisieran proporcionarlos.

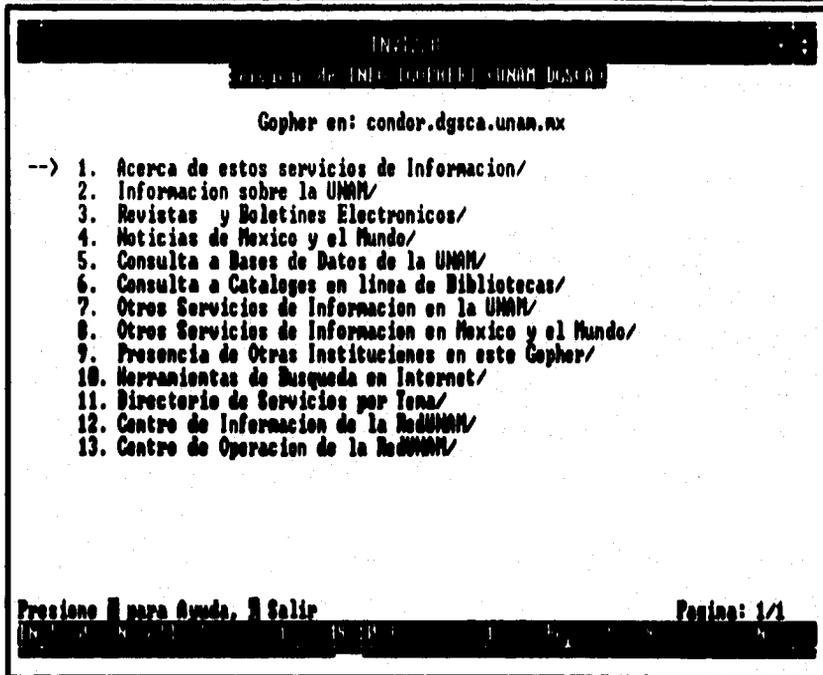


FIG. 2.1.7 Ventana de Gopher

Los inventores de *Gopher* crearon una aplicación especial que pudiera guiar a los estudiantes a la información que requerían, sin necesidad de una capacitación. Para lograr esto, organizaron el sistema por temas de manera que es posible verlo como una enorme base de datos, en lugar de cientos de bases de datos pequeñas interconectadas. Se puede tener acceso a archivos FTP, catálogos de biblioteca y otros servicios de bases de datos con propósitos especiales (basados en *Telnet*); Pero sólo *gopher* sabe dónde se localizan realmente los datos, la forma de tener acceso a ellos y que existen muchos servidores que los proporcionan.

No tomó mucho tiempo darse cuenta de que si el sistema funcionaba bien para muchos servidores colocados en diferentes departamentos, también podía funcionar para servidores colocados en diferentes partes del mundo. Todo lo que se requería era que Internet enlazara a



todos. En cuatro años, más o menos, la utilización del sistema *Gopher* creció de un servidor a más de 1300 servidores.

Al utilizar *Gopher* y cuando se ha encontrado algo de interés, esta información puede ser accedida, y no es necesario conocer la dirección IP ni cambiar de programas que permitan realizar la transferencia o la conexión a otros servidores (ver figura.2.1.8). Todo es transparente para el usuario. La gran ventaja que ofrece *Gopher* no es tanto que ahorre la búsqueda de domicilios o nombres de recursos, o que no se tenga la necesidad de utilizar varios comandos para obtener lo que se desea. la ventaja real consiste en que permite navegar a través de los recursos de Internet, sin importar su tipo (texto, sonido, imagen, etc.). *Gopher* también cuenta con un servicio de consulta en línea a bibliotecas que ayudan a buscar títulos, temas o autores.

Gopher no permite el acceso a cosas que no estén disponibles por otros medios. No existe un formato especial de "recursos *Gopher*" para que se pueda tener acceso, por lo menos no lo hay en el mismo sentido que en los servidores FTP o los directorios donde algunos archivos puedan estar disponibles sólo a través de *Gopher*, pero es parte de la política de seguridad. Si se tiene acceso a esos archivos a través de *Gopher*, llegarán al sistema local a través de FTP. Una vez que se ha encontrado la información que se quiera revisar, *Gopher* sabe que aplicación debe utilizar para obtener un elemento en particular (telnet, ftp, etc). Cada tipo de recurso se maneja de manera diferente. Sin embargo, todos son manejados de manera intuitiva, dependiendo del tipo de cliente *Gopher* que se esté utilizando. *Gopher* es lo suficientemente inteligente para imponer restricciones de licencias. Algún software o algunos recursos (por ejemplo periódicos en línea) pueden tener licencia para ser utilizados únicamente dentro de una ciudad o campus en particular.

Para tener acceso al sistema *Gopher*, se necesita un programa cliente *gopher*. El programa cliente especial debe estar instalado en una computadora que se encuentre en Internet. Cada cliente tiene la "forma y las maneras" del sistema en que se está ejecutando. Casi todas las tareas que puede realizar con un cliente *gopher* dado, se pueden realizar con otras versiones del



mismo. Cualquier cliente que se encuentre instalado, estará configurado con el domicilio de Internet del servidor del cual se extrajo.

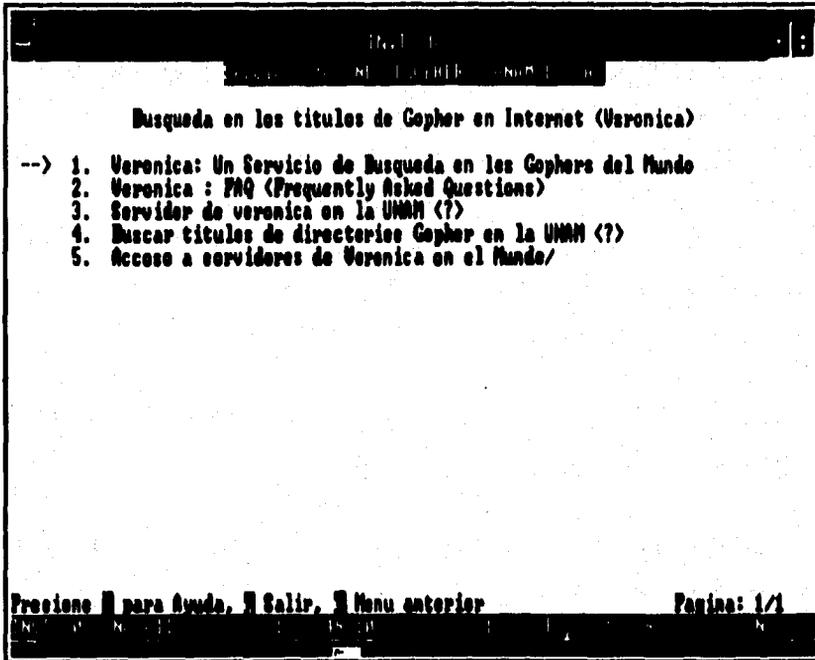


FIG.2.1.8 Ventana del servicio INFO de la UNAM (GOPHER).

WAIS

WAIS (*Wide Area Information Service*) es otro de los servicios de Internet. Sirve para realizar búsquedas a través de material indexado y localizar artículos basándose en su contenido. WAIS permite realizar explotaciones a través de archivos de Internet en busca de documentos que contengan ciertos grupos de palabras.

WAIS es una herramienta para trabajar con conjuntos de datos o con bases de datos y no examina los datos en el proceso de la búsqueda, examina el índice, puede seleccionar la



información y presentarla sin importar el formato que tenga. Como *Gopher*, WAIS permite encontrar y tener acceso a recursos de información de la red sin importar el lugar exacto de residencia.

En *Gopher* se encuentran los recursos informativos buscando a través de una secuencia de menús hasta hallar algo apropiado. WAIS lleva a cabo la misma tarea, sólo que realiza la búsqueda completa. Un comando WAIS puede ser descrito de la siguiente manera: el usuario ordena "encontrar documentos acerca de algo en alguna biblioteca". Entonces WAIS revisa los documentos que existen en esa biblioteca (o bibliotecas) que le han sido indicados e informa cuáles son los documentos con los que probablemente contengan la información requerida. Después si así lo desea el usuario, WAIS despliega los documentos en pantalla.

En la red actualmente existen más de 500 bibliotecas WAIS gratuitas, y en este servicio se puede preguntar "¿Existe una biblioteca para este tema?".

Así se puede averiguar con facilidad si WAIS cuenta con los recursos informativos de utilidad para cualquier usuario.

WWW (World Wide Web)

La *World Wide Web*, también conocida como WWW es el servicio de información de Internet más reciente. *Web* está basada en una tecnología llamada hipertexto. La mayor parte del desarrollo ha sido realizada en el CERN, el Laboratorio Europeo de Física de Partículas. Aunque fue realizado en este laboratorio, no es una herramienta diseñada por y para físicos, a pesar de que estos científicos fueron quienes lo desarrollaron inicialmente, *Web* es una de las herramietas más flexibles para navegar por Internet (ver figura.2.1.9), aún continúa desarrollándose por lo que no debe sorprenderse si no trabaja como se espera.

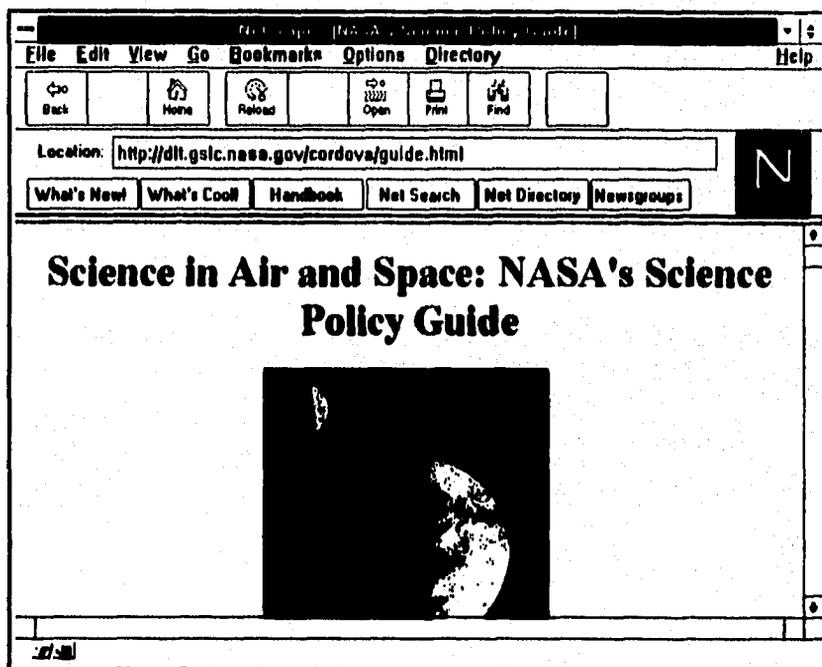


FIG.2.1.9. Pantalla de NETSCAPE.

Para hacer uso de *Web* el primer paso es conectarse vía telnet con info.cem.ch. Se trata de un examinador orientado a texto que funciona en cualquier terminal. Existen muchos examinadores disponibles, si se decide instalar uno en otro sistema (lo cual es altamente recomendable si se planea utilizar *Web* con frecuencia) se puede escoger el que más convenga entre examinadores orientados a texto para el sistema X Window y estaciones de trabajo, así como computadoras *Macintosh* y *PC*. es la base para *Mosaic* (ver figura.2.1.10), examinador que funciona en *UNIX* bajo el sistema X Windows, lo mismo que en *Macintosh* y *Microsoft Windows*.

WWW es un intento de organizar toda la información en Internet, además de cualquier otra aplicación local que se necesite, a manera de un conjunto de documentos de hipertexto. la cantidad de hipertexto que se encuentra en la Red, se ha incrementado considerablemente en los



últimos años. Muchas exhibiciones de museo, revistas y otras presentaciones en hipertexto están disponibles, incluyendo a *Global Network Navigator* (GNN), de *O'Reilly & Associates* (ORA). El problema radica en la escasez de herramientas para construir la estructura de vínculos. La mayoría de los documentos de hipertexto actualmente disponibles se construyeron manualmente. Los editores de hipertexto son muy recientes y tienen mejores herramientas para crearlos. Todos los elementos que se encuentran en hipertexto pueden ser modificados en cualquier momento.

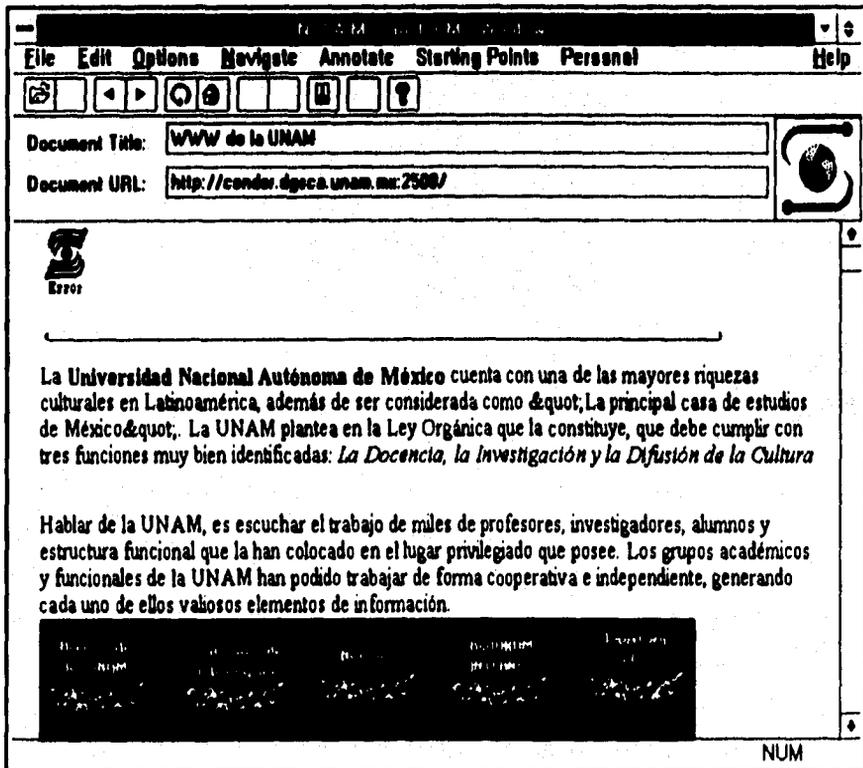


FIG.2.1.10 Pantalla del visualizador de documentos MOSAIC.



VIDEOCONFERENCIA

La Universidad Nacional Autónoma de México a través de la Dirección General de Servicios de Cómputo Académico (DGSCA), realiza entre sus funciones, análisis continuos sobre los desarrollos de nuevas tecnologías que existen en el mercado para resolver, o en su caso optimizar, el quehacer de las actividades académicas que nuestra Universidad lleva a cabo.

Uno de los retos académicos que la Universidad se ha planteado es ampliar las alternativas para la educación a distancia, así como la colaboración en proyectos de investigación e intercambio académico y cultural a nivel nacional e internacional. Para ello se ha venido utilizando distintas tecnologías que permiten la interacción parcial entre puntos remotos (radio, televisión, audioconferencia, audiográficos y comunicación medida por computadora, etc.) , a partir de la evolución de las telecomunicaciones (fibra óptica y enlaces satelitales) y la disponibilidad de nuevos sistemas de comunicación interactiva (videoconferencias) que hoy facilitan la interacción total en la realización de las actividades antes mencionadas.

La *Videoconferencia Interactiva* es una forma de comunicación audiovisual bidireccional y de transmisión de datos en tiempo real, mediante el uso de tecnología que permite enlazar dos o mas sitios geográficamente distantes.

Este medio es una herramienta invaluable de comunicación ya que constituye la forma de interactuar a distancia que más se aproxima a la realidad.

Por lo anterior, las posibilidades que la Videoconferencia ofrece son innumerables, entre las que destacan :

Educación Continua

Capacitación

Telemedicina



Actualización profesional

Videoconferencia punto a punto o multipunto

Mesas redondas virtuales

entre otras.

Un sistema de videoconferencia consiste en varios componentes integrados : monitores de vídeo, cámaras, micrófonos y principalmente el CODEC, que es el componente que captura las señales en vivo, las comprime y las convierte en señales digitales para transmitir las a los sitios remotos, a través de líneas dedicadas.

Para realizar actividades utilizando la videoconferencia se requiere de al menos dos sitios para la interacción; es indispensable que cada sitio cuente con un sistema de videoconferencias así como con el circuito de telecomunicación que los enlace.

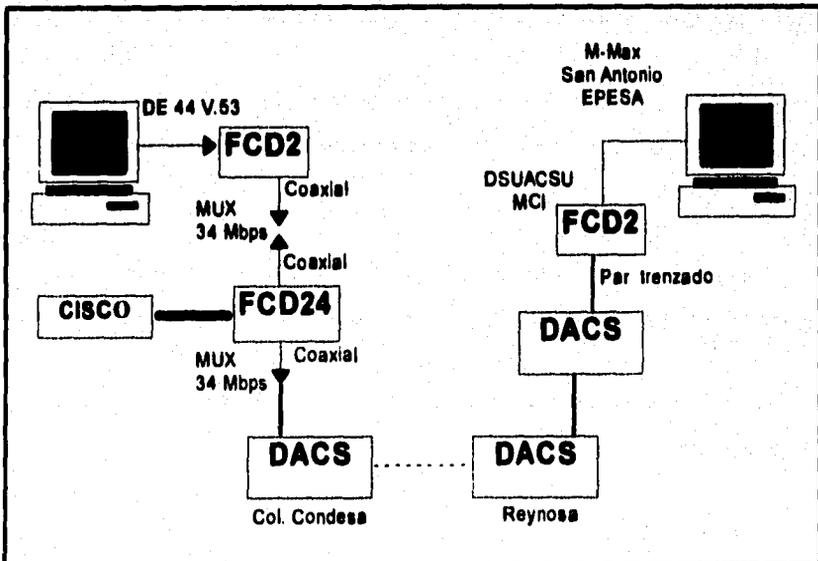


FIG.2.1.11. Enlace para Videoconferencias.



Actualmente, la DGSCA tiene la capacidad de operar a velocidades de transmisión o ancho de banda de hasta 1.28 Mbps (Mega bits por segundo). Cabe mencionar que mientras mayor sea el ancho de banda, es mejor la calidad de la transmisión.

Es indispensable que cada sitio remoto cuente, al menos con una persona con los conocimientos necesarios de telecomunicaciones y operación técnica del equipo de videoconferencia , en tanto se encarga de establecer el enlace con los demás nodos para cada una de las transmisiones que se llevan a cabo.

Para lograr que la transmisión de un curso, seminario, taller, etc., sea exitosa en función del impacto didáctico, es recomendable, y en algunos casos indispensable, preparar material de apoyo a la exposición del contenido.

En virtud de que es posible conectar una gran variedad de equipos periféricos a los sistemas de videoconferencia, las posibilidades de generar material de apoyo son mayores de las que se pueden utilizar en actividades presenciales. Por lo anterior, se recomienda que una persona que conozca las alternativas mencionadas y las técnicas de diseño y elaboración de material trabaje en colaboración con el (los) expositor (es), desde la fase de planeación, hasta el manejo dinámico del grupo durante la sesión.

El uso de la videoconferencia e Internet en educación a distancia hoy en día, permiten una interacción de los participantes con el docente y entre sí, que va más allá de las posibilidades de educación presencial.

Internet es el complemento idóneo para las actividades a distancia ya que nos brinda, entre otras cosas, las siguientes posibilidades :

- Alta velocidad. Los mensajes se transfieren en cuestión de segundos
- Independencia del tiempo. Los mensajes pueden ser escritos, enviados y leídos en cualquier momento sin tener que ajustarse a un horario.



- Independencia de espacio. Los mensajes pueden enviarse y ser consultados a nivel mundial desde el hogar, centro de estudios o trabajo.
- Comunicación síncrona. Los usuarios pueden comunicarse simultáneamente.
- Comunicación asíncrona. No es necesario que el experto y el alumno estén enlazados al mismo tiempo. El correo electrónico es un ejemplo de ello.
- Aprendizaje no lineal. El estudiante puede profundizar en sus investigaciones tanto como lo desee, al consultar la información que está disponible en Internet.

Es de considerar, que la videoconferencia nos brinda, además de las ventajas ya mencionadas, un mejor aprovechamiento del tiempo debido al costo de las telecomunicaciones e infraestructura necesaria del sitio.

Como resultado de las experiencias que DGSCA ha obtenido realizando actividades de videoconferencia, hemos encontrado que los estudiantes adquieren una actitud más participativa en su aprendizaje, tanto durante la sesión como fuera de ella, consideramos que esto constituye un gran avance en lo que se refiere al proceso de enseñanza-aprendizaje.



2.2 REQUERIMIENTOS DEL USUARIO.

Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esto puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad de la empresa o brindar soporte a la gerencia. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

Ciertos tipos de requerimientos son tan fundamentales que son comunes en casi todas las situaciones. Dar respuesta a un grupo específico de preguntas, será de gran ayuda para comprender los requerimientos básicos.

También existe otra clase de requerimientos que depende de si el sistema está orientado hacia transacciones, toma de decisiones o se extiende por varios departamentos. La identificación de las necesidades es el punto de partida para el desarrollo de sistemas, este proceso se lleva a cabo cuando el analista se entrevista con el cliente, definiendo los objetivos del sistema:

Por ejemplo, la información que se va a producir o suministrar, las funciones y el rendimiento requerido. El analista se asegura de distinguir entre lo que necesita el cliente (elementos críticos para la realización) y lo que el cliente quiere (elementos deseables pero no esenciales). (Ver figura. 2.2.1).

Para realizar correctamente el desarrollo del software es necesario realizar una especificación completa de los requerimientos del mismo.

El análisis de requerimientos es el primer paso técnico en el proceso de la ingeniería de software y es cuando se establece de forma general el ámbito del programa y este se da en una especificación concreta que se convierte en la base de la fase de desarrollo.



FIG. 2.2.1 Distinguir requerimientos

Para llegar a un óptimo establecimiento el analista y el cliente tienen un papel activo en la especificación de requerimientos ya que existe una gran comunicación entre ellos y en ocasiones resulta difícil porque frecuentemente se presentan cambios por mala interpretación o falta de información. (Ver figura.2.2.2)

El análisis de requerimientos plantea la asignación de *software* a nivel sistema y el diseño de programas, además facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interface con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa.



FIG .2.2.2 Falta de Información

El análisis de requerimientos permite al ingeniero refinar la asignación de *software* y representar el dominio de la información que será tratada por el programa. Da también al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y procedimientos de diseño. Finalmente suministra al analista y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.

La fase de requerimientos puede dividirse en cuatro áreas:

1. Reconocimiento del problema: Inicialmente el analista estudia la especificación del sistema y el plan de proyecto. Es importante conocer el contexto del sistema y el ámbito de los programas que se utilizaron en la generación de las estimaciones de la planificación. Posteriormente, se debe



establecer la comunicación para el análisis, de forma que asegure el reconocimiento del problema. Las formas de comunicación requeridas para el análisis se muestra en la figura. 2.2.3

2. *Evaluación y síntesis*: La evaluación del problema y la síntesis de la solución es la siguiente área principal del trabajo del análisis. En esta etapa el analista debe evaluar el flujo y la estructura de la información, refinar minuciosamente las funciones del programa a detalle, establecer las características de la interface del sistema.

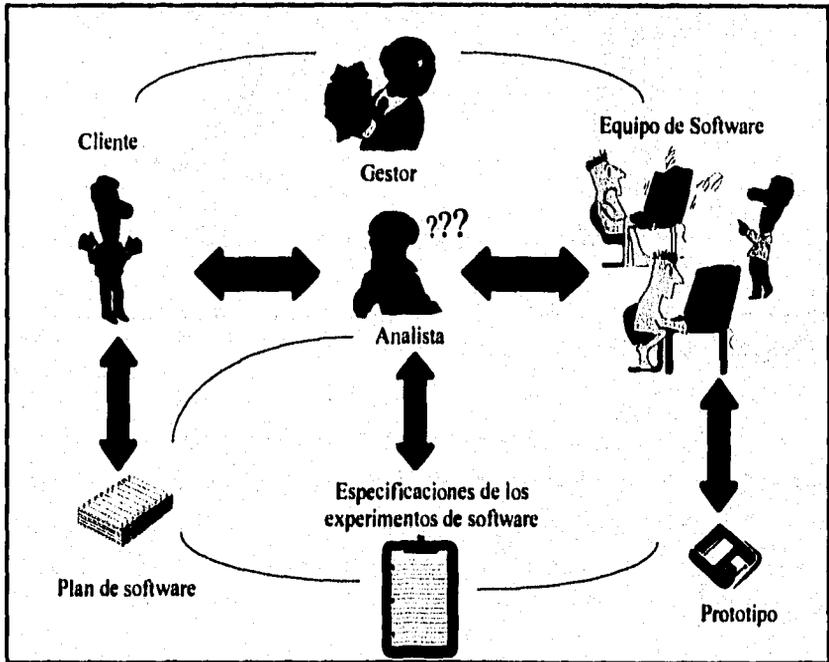


FIG. 2.2.3 Formas de comunicación

3. *Especificación*: Las tareas asociadas con el análisis y especificación existen para dar una representación del programa que pueda ser revisada y aprobada por el cliente. Una vez que se hayan descrito las funciones básicas, comportamiento, interface, información, se especifican los criterios



de validación para demostrar una comprensión de la implementación de los programas. Estos criterios sirven como base para hacer la fase de pruebas durante el desarrollo de los programas.

4. *Revisión*: La revisión de los requerimientos sirve como base para una verificación conducida por el cliente y el técnico. Casi siempre produce modificaciones en la función, comportamiento, representación o criterios de validación.

Entre los problemas que pueden encontrarse durante el análisis de los requerimientos están las necesidades asociadas con la adquisición de la información pertinente, el manejo de la complejidad del problema y el acomodar los cambios que puedan ocurrir durante y después del análisis.

Los problemas que surgen durante la fase de requerimientos son los siguientes:

1. Pobre comunicación que hace difícil la adquisición de la información.
2. Técnicas y herramientas inadecuadas que producen especificaciones imprecisas.
3. Tendencia a acortar el análisis de requerimientos, conduciendo a un análisis inestable.

Las técnicas de análisis de requerimientos combinan procedimientos sistemáticos con una notación única para analizar los dominios de la información funcional de un problema de *software*.

Uno de los primeros puntos a tratar es el desarrollo de un sistema amigable, significando que no deberá causar problemas el uso del sistema para el usuario.

Para tomar este punto podemos decir que el sistema deberá estar planteado para usuarios con conocimientos básicos en computación, al tener esta aseveración como premisa del sistema, podremos garantizar que el uso del sistema no causará ningún problema para el usuario y tendremos la seguridad de que será una herramienta útil para el mismo.

Para desarrollar nuestro sistema requerimos de información de la DGSCA a través de su departamento de redes, en el cual nos proporcionaron información básica del proyecto.

El sistema podrá ser empleado en todos aquellos lugares en donde existan servidores de claves de internet (vía modem), y que requieran de un servicio de seguridad y control del uso de sus claves.

Es útil ver la determinación de requerimientos a través de tres grandes actividades: anticipación, investigación y especificación de requerimientos.

Anticipación de requerimientos. Prever las características del sistema con base en la experiencia previa. Esto puede llevar a investigar áreas y aspectos que de otra forma no serían tomados en cuenta.

Investigación de requerimientos. Estudio y documentación del sistema actual utilizando para ello técnicas para hallar hechos, análisis de flujo de datos y análisis de decisión.

Especificación de requerimientos. Análisis de los datos que describen el sistema para determinar qué tan bueno es su desempeño, qué requerimientos se deben satisfacer y las estrategias para alcanzarlos.

En este caso se debe tomar especial atención en el sentido de que debe de ser un sistema muy amigable, ya que, la mayoría de los usuarios serán personas que muy pocas veces traten con una computadora. Así que las instrucciones y comandos deben ser sencillos de usar.

En este módulo de requerimientos del usuario, se plantean e identifican las necesidades que el usuario tiene y las cuales se pretende satisfacer mediante el desarrollo del presente sistema. A continuación se presentan algunos de los principales requerimientos para el sistema de monitoreo de claves de acceso a Internet para la validación del propietario a través de RedUNAM.

**a) Interface.**

- La interface hacia el usuario debe ser funcional, elegante y debe evitar la confusión. Para producir una interface funcional y visualmente confortable, se necesita entender los requerimientos del departamento, de la aplicación y una percepción de la estética.
- La aplicación debe ser fácil de leer, observar y de usar en comportamiento y organización.
- Debe compartir los datos generados en el sistema con otras aplicaciones de Windows.
- Debe tener la facilidad de enviar los datos del monitoreo a: access.
- Presentar catálogos y poder seleccionar la opción deseada sin tener que teclearla.
- Debe presentar la característica de ser ágil en la captura de información.

b) Operación del sistema*Requerimientos generales.*

- El manejo del sistema debe ser rápido de aprender.
- Debe manejar un número ilimitado de claves.
- Debe ser de fácil instalación, de manera que las personas que no se encuentren muy familiarizadas con el manejo de la computadora puedan instalar y utilizar el sistema con facilidad.
- Contar con un panel de control, desde el cual se pueda editar información diferente relacionada con la clave.



- Generar un catálogo maestro de claves.
- Debe ser capaz de generar informes impresos con diseños mejorados y con formatos propios y adecuados a las necesidades de la Dirección de Cómputo.

El generador de reportes debe brindar las siguientes facilidades:

- Seleccionar los datos que se desea aparezcan en el reporte.
- Incluir el logotipo de la dirección de cómputo.
- Manejar una base de datos.
- Incluir seguridad. (Manejar claves de acceso a la base de datos) en el manejo del sistema.
- Que posea un front-end amigable.
- Que requiera un mínimo mantenimiento.

Requerimiento de Envío y Recepción de Datos.

- Detección de la línea de conexión al servidor de comunicaciones.
- Envío de la pantalla de validación de datos al usuario, con un front-end visual.
- Que cuente con un módulo de recepción de los campos validados por el usuario.
- Que cuente con un sistema de desconexión de la línea del usuario al servidor de comunicaciones.



Requerimientos de monitoreo.

Que cuente con un sistema "aleatorio" de selección de clave.

Requerimientos de procesamiento de la información.

Que cuente con un sistema de comparación entre los campos validados por el usuario y los almacenados en la B.D.

La finalidad de la presentación de los puntos anteriormente mencionados es obtener un panorama general de las necesidades que el usuario tiene del sistema.

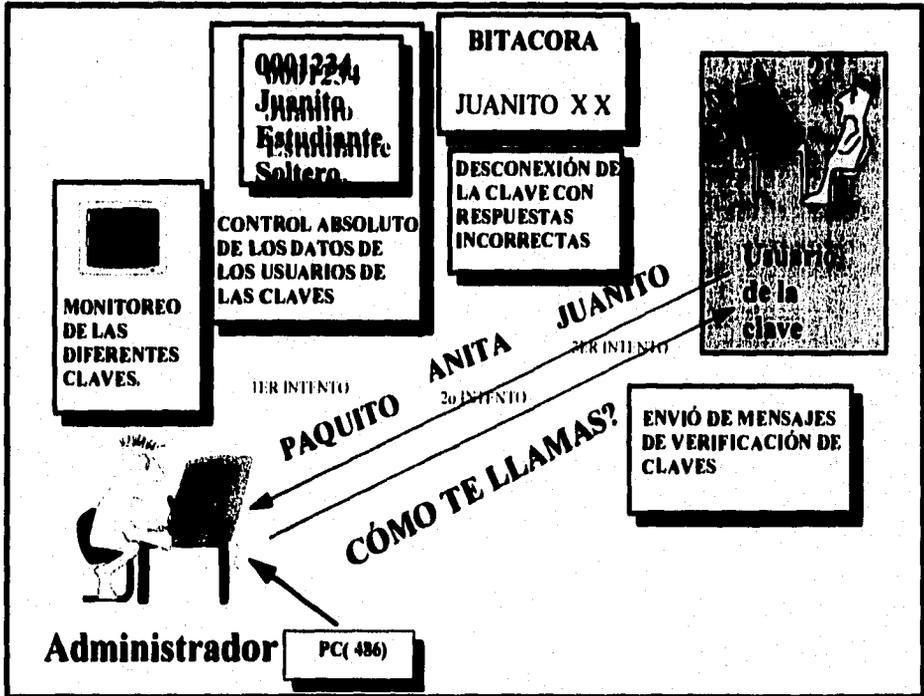


FIG. 2.2.4 *Requerimientos Especificos del Usuario*



Los requerimientos específicos del usuario para el desarrollo del sistema son:

(Ver figura. 2.2.4)

- Uso de un menú gráfico para el acceso a otros menús, utilizando para ello dos dispositivos, el mouse, y el teclado.
- Monitoreo de las diferentes claves que se encuentran por acceder en el momento en que el administrador lo desee.
- Control absoluto de los datos de las personas que tiene claves de acceso.
- Posibilidad de enviar mensajes de verificación a los usuarios de las claves en forma aleatoria.
- Posibilidad de desconectar sesiones de claves repetidas.
- Posibilidad de desconexión de claves que no contesten correctamente los cuestionamientos a los que serán sometidos.
- Obtener un sistema confiable y amigable
- Que el sistema funcione correctamente en cualquier PC(486 en adelante)



2.2.1 PLAN DE TRABAJO

Los planes describen los mecanismos que se ocuparan para lograr las metas y requisitos. Por ejemplo, el objetivo de entregar los productos finales a tiempo se puede expresar en términos de alcanzar logros importantes del proyecto a tiempo. Un logro es un hecho significativo en el ciclo de vida del desarrollo del producto de programación; ejemplos de logros son terminación del análisis de requisitos, terminación del diseño, e integración y prueba con éxito de los componentes del sistema.

Para planear el modo de alcanzar cada logro a tiempo, se deben responder preguntas como las siguientes:

1. ¿Cuántos logros son apropiados ?
2. ¿Cuándo ocurren ?
3. ¿Qué recursos se necesitan para alcanzar cada logro ?
4. ¿Quién será el responsable de alcanzarlos ?
5. ¿Qué se debe cumplir en cada logro ?
6. ¿Exactamente qué significa alcanzar cada logro ?

La consideración de estas preguntas lleva a aspectos como modelos de ciclo de vida, estimación de costos, y cantidad de personal para el proyecto.

El ciclo de vida del producto incluye todas las actividades requeridas para definirlo, desarrollarlo, probarlo, entregarlo, operarlo y mantenerlo. Es esencial definir un modelo de ciclo de vida para cada proyecto de programación, puesto que permite clasificar y controlar las diferentes actividades necesarias para el desarrollo y mantenimiento del producto. Un modelo de ciclo de vida entendido y aceptado por las partes interesadas en el proyecto, mejora la comunicación, permitiendo así una mejor administración, asignación de recursos, control de costos y calidad del producto.(ver figura 2.2.1.1)

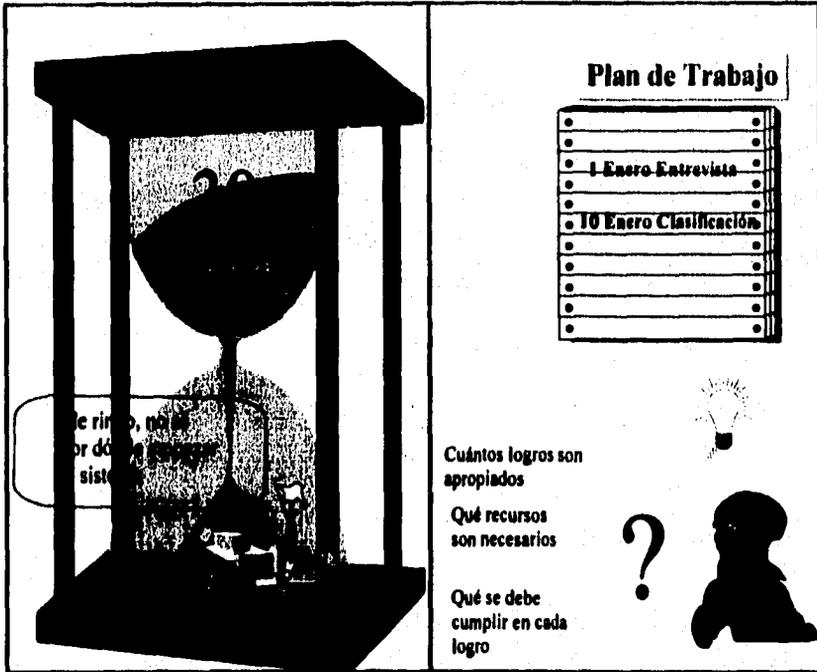


FIG. 2.2.1.1 Plan de Trabajo

Para elaborar el plan de trabajo primeramente se penso, en los aspectos mas importantes que afectan a las líneas de acceso vía *MODEM* para usar internet, sobre todo en la sobre carga y saturación que provoca el mal uso de tales claves.

Para obtener información sobre ello, se recurrió primeramente al departamento de la coordinación de redes, posteriormente se entregaron cuestionarios a los coordinadores de redes de la DGSCA y de teleinformática, así como a usuarios de uso frecuente de la DCAA; seguido se procedió a platicar con los encargados de la contabilidad de las claves de acceso (administrador de *TACACS* para estación de trabajo) y del administrador de *SLIP*, para plantear las metas de este proyecto y que se nos proporcionara la mayor información posible de todo el sistema de acceso vía *MODEM*.



Paralelamente se orientó nuestro esfuerzo a localizar en Internet (por medio de NETSCAPE) el software de contabilidad "Xtacacs", a sabiendas de que es un software de dominio publico, para su posterior modificación de ser posible, para nuestros fines, se desarrollo el diseño e implementación tanto de la base de datos como del Front-End , hasta culminar con su enlace via incrustación de objeto (OLE).



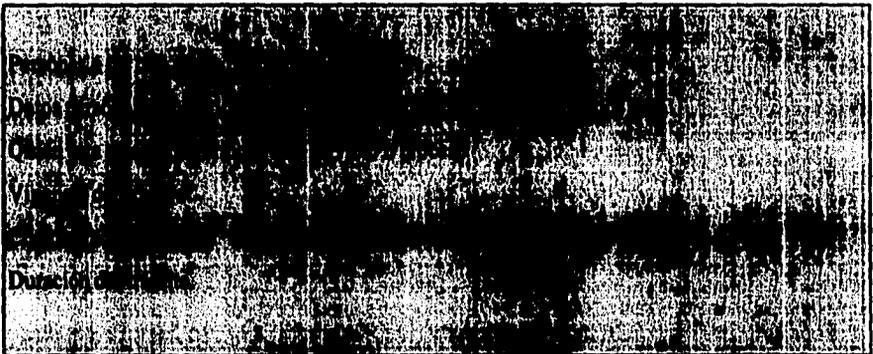
2.2.2. RECOPILACIÓN DE INFORMACIÓN.

Esta etapa nos permitió visualizar la problemática existente para el control de las claves de acceso a Internet que se maneja en la UNAM, así como las actividades que se realizarán para brindar este servicio a los usuarios, y de esta manera poder definir así el alcance del sistema.

Para la recopilación de información se utilizaron básicamente las técnicas de:

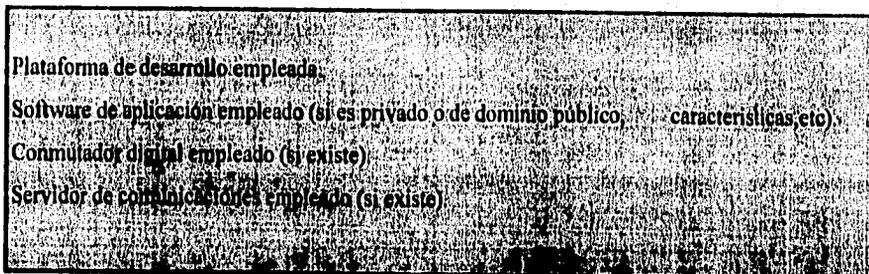
- Entrevistas apoyadas en cuestionario.
- Sesiones periódicas de retroalimentación (Entrevistas personales).
- Revisión de documentos a través de consultas en Netscape y DGSCA
- Depuración de la información obtenida

Para realizar la recopilación de la información se determinaron los elementos sobre los que se iba a efectuar la búsqueda. En primer término se necesita información sobre el sistema actual que se ha estado empleando en la DGSCA para sesiones vía MODEM, así como la información sobre como se lleva el proceso actual de asignación de claves. Para que la búsqueda fuera más específica se convino que de los puntos anteriores se iba a buscar la siguiente información sobre el proceso de asignación:





Sobre el sistema actual se decidió buscar la siguiente información:



Además de la información del sistema actual y de los usuarios que lo emplean (proceso de asignación), se debe recopilar información sobre: como se puede llevar a cabo, un front-end agradable al usuario en ambiente Windows, el como realizar el almacenamiento de las claves para su posterior validación, los programas de aplicación de manejo de datos en líneas telefónicas, para que en un momento determinado se pueda realizar la selección de cual se va a utilizar o de ser necesario implementar).

Es necesaria información de cómo trabaja el software que hasta ahora se ha empleado para la administración de las claves de Internet vía MODEM, la cual será la principal ayuda de comunicación entre el sistema y usuario.

Una vez determinados los puntos a buscar se procedió a la recopilación de la información.

En la Unidad Administrativa perteneciente a la Dirección General de Servicios de Cómputo Académico(DGSCA) se cuenta con formatos de registro (como se muestra en la figura 2.2.2.1) que los usuarios llenan como trámite para solicitar su clave de Internet. En estas formas de registro se contemplan una diversidad de datos como nombre completo del usuario, su dirección, teléfono, etc.


DIRECCIÓN GENERAL DE SERVICIOS DE CÓMPUTO ACADÉMICO.


Departamento de Atención a Usuarios

Solicitud de clave de acceso de REDUNAM (VÍA MODEM)

Datos del usuario:		
Nombre	Apellido paterno	Apellido materno
Dependencia:		
Dirección:		
(Calle, Número, Colonia, Código Postal)		
Teléfonos:		Fax:
Domicilio particular:		
(Calle, Número, Colonia, Código Postal)		
Teléfono particular:		
Función o cargo:		
Jefe inmediato nombre y cargo:		

Vo. Bo. Director Dependencia

(Certifico que el usuario es colaborador académico de esta Dependencia)

firma de compromiso

POLÍTICAS GENERALES DE USO DE REDUNAM

Las claves son personales e intransferibles. Es responsabilidad del poseedor de la clave mantener la confidencialidad de la contraseña correspondiente. El poseedor se obliga a respetar las políticas de uso aceptable impresas al reverso.

FIG. 2.2.2.1 Forma de registro.



Dentro de esta misma dependencia se encuentra el departamento de redes que es el encargado principal de brindar soporte a este sistema y que cuenta con manuales de información técnica sobre la plataforma en la que esta corriendo actualmente la aplicación, así mismo cuentan con la información pertinente relacionada con el servidor de comunicaciones y el conmutador telefónico.

Dado que la plataforma empleada es una estación de trabajo SparcSun, y que esta se esta implantando como un estándar dentro de la universidad, cabe hacer mención de que la mayoría de las dependencias cuentan con estos manuales, tal es el caso del instituto de Ingeniería y de la Dirección de Cómputo para la Administración Académica (DCAA) ubicada actualmente en la planta baja del edificio IIMAS².

Por lo que respecta al software de aplicación y al(los) protocolo(s) utilizados, dado que son de dominio público, toda la información referente a ellos se bajo (recibió) de Internet navegando en esta a través de Netscape. Cabe hacer mención de que a pesar de que el software de aplicación es de dominio público solo lo es en su versión ejecutable por lo que se tuvo la necesidad de contactar a los propietarios del mismo con el fin de que nos proporcionarán una copia del código fuente.

La DGSCA proporciona los teléfonos de RedUNAM(atención a usuarios), por medio de los cuales se cuenta con gente que puede resolver o en su caso canalizar las cuestiones afines con el tema antes mencionado.

La DGSCA proporciona a los usuarios con la adquisición de su clave un software de aplicación, que contiene Mosaic, FTP, Talk, el protocolo SLIP, PROCOMM, por lo que se procedió a conseguir una copia de este software.

Por lo que respecta a la información precisada para el Front-End y el sistemas de almacenamiento de información, debido a su gran auge y demanda de estos últimos años se cuenta con una amplia bibliografía que es de fácil acceso a su consulta. Si se desea obtener más

² Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas



información sobre algún tema en especial la opción es comunicarse directamente a la compañía y pedir información. Ya que cada compañía cuenta con manuales y especialistas altamente calificados.

Una opción en cuanto al *software* de comunicación es el que proporciona Microsoft Windows 3.11 que cuenta con una utilería *terminal* que permite configurar nuestro MODEM y realizar la comunicación vía línea telefónica sin necesidad de adquirir alguna licencia o software especial. Para trabajar con esta herramienta se necesita un modem ya sea interno o externo, puertos serial (según el caso) y el software proporcionado por la Universidad.

También se requirió recopilar información sobre diversos tipos de computadoras personales, para lo cual se contactaron a varios distribuidores de equipos de cómputo que nos facilitaron información sobre sus productos.

Existen varios factores críticos para la selección de una computadora personal de acuerdo a la aplicación que en ella vaya a correr, como lo son: (ver figura 2.2.2.2)

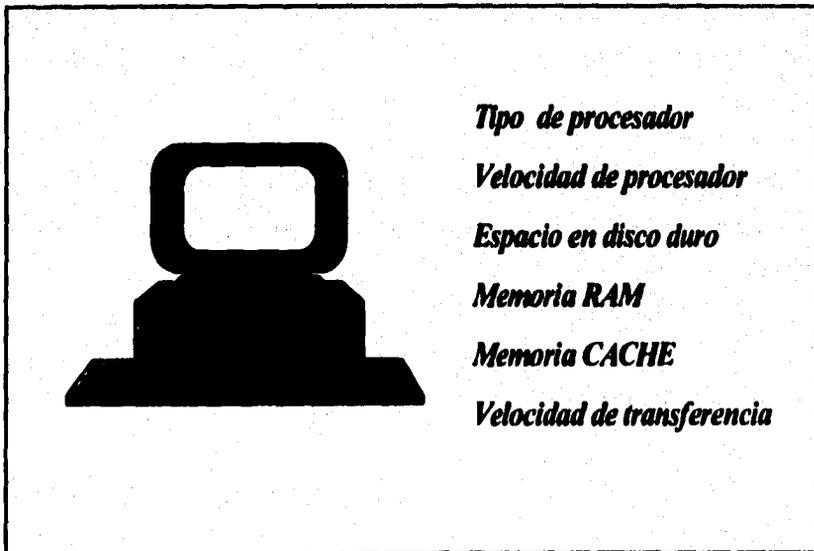


FIG. 2.2.2.2 Factores Críticos.



A continuación se muestran los cuestionarios aplicados :

Coordinadores y personal de redes.

1. ¿De qué privilegios gozan los acreedores de una clave de acceso a internet ?
2. ¿Cuántas claves se encuentran hasta la fecha registradas ?
3. ¿Existen diferentes tipos de claves de acceso a Internet.?
4. ¿Consideran que el numero de usuarios se incrementará considerablemente los próximos años?
5. ¿Existe algun control para evitar que una clave sea utilizada simultaneamente por dos personas?
6. ¿Actualmente cuentan con algún sistema auxiliar en el control de claves de acceso a Internet ?
7. ¿Qué funciones realiza el sistema que actualmente manejan?
8. ¿Considera necesario controlar el pirataje de las claves ?
9. ¿En su opinión cual es el porcentaje en el que repercute este descontrol en el aspecto económico, para la UNAM?

Usuarios de las claves de Internet

1. ¿Qué tan eficiente considera que es el servicio via modem de acceso a Internet?
2. ¿Considera que el precio de las claves que brinda la UNAM es justo al servicio?
3. ¿Qué limitantes encuentra usted en los servicios prestados. ?
4. ¿Cuántas personas son encargadas del manejo de este sistema?



2.2.3 CLASIFICACIÓN DE LA INFORMACIÓN

La información que se obtuvo del punto anterior será utilizada para crear la base de datos, obtener un formato de registro para la solicitud de claves, las pantallas necesarios del sistema, la información que será enviada y aquella que será recibida para su validación, el tipo de monitorco a realizarse y la plataforma en la que deberá correr la aplicación, así como los niveles de seguridad imprescindibles en un sistema de estas características.

El sistema en forma global enviará y recibirá información estratégica, interactuando con los equipos de comunicaciones, la plataforma actual y auxiliar Pc, el programa de liga (de ser necesario) y el usuario final, soportado por la base de datos:

Usuarios.

EXTERNOS.
INTERNOS (ACADEMICOS).
INTERNOS (ESTUDIANTES).
INTERNOS (PROFESORES).

Referente a los protocolos.

Se consigue información a través de los manuales SLIP y PROCOMM-PLUS que se encuentran en el departamento de redes de la DGSCA, consultando a los administradores del actual sistema, o bien navegando con NetScape en Internet, haciendo búsquedas del tema de interés con la ayuda de hipertexto.

**Referente al conmutador.**

Se espera que el departamento de redes cuente con algunos manuales que nos sean de interés sobre los conmutadores telefónicos y de ser el caso ponerse en contacto con la compañía Teléfonos de México (o con la compañía precisa).

Referente al proceso de asignación de claves:

Se puede realizar a través de la Unidad Administrativa de DGSCA.

Referente a la estructura de base de datos.

La base de datos se puede estructurar tomando en cuenta la clasificación anterior de los usuarios, y considerando que parte de la información de cada uno de los elementos anteriores será para acceder la base de datos y validar los datos.

Referente al sistema actual.

Se puede obtener a través de la DGSCA por medio del departamento de redes o, consultando a los administradores del actual sistema, o bien navegando con NetScape en Internet, haciendo búsquedas del tema de interés con la ayuda de hipertexto.

En lo referente de manejo de datos (liga).

Consultando los diferentes manuales de programas comerciales para analizar su funcionamiento y ver si es posible adaptarlos a nuestras necesidades; de otra forma debemos realizar un software propio.



2.3 ANÁLISIS (DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL)

Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos detectados. Las fuentes que originan las ideas de productos de programación incluyen las necesidades del cliente generada externamente, las necesidades internas de la organización, planes de mercadotecnia, y los planes o misiones organizacionales. La mayor parte de las organizaciones que desarrollan productos de programación son muy selectivas al decidir qué productos desarrollarán; no se explotan todas las oportunidades. La decisión de llevar a cabo el proyecto se basa, generalmente, en el resultado de un estudio de factibilidad.

A continuación se listan los pasos que se recomienda seguir para la planeación de un proyecto de programación, los cuales serán explicados brevemente.

1. Desarrollar un enunciado definitivo del problema por resolver. Incluir una descripción de la situación actual, restricciones del problema y de las metas que se lograrán. El enunciado del problema debe realizarse empleando terminología que sea entendida por el cliente.
2. Justificar una estrategia de solución computarizada para el problema.
3. Identificar las funciones por realizar, las restricciones, el subsistema del equipo electrónico, el subsistema del producto de programación, y el del personal.
4. Determinar los objetivos y requisitos en el nivel del sistema para el proceso de desarrollo y los productos finales.

Establecer criterios de alto nivel para la aceptación del sistema.



Para el desarrollo de una estrategia de solución es deseable:

1. Esbozar varias estrategias de solución, sin considerar las restricciones.
2. Realizar un estudio de factibilidad para cada estrategia.
3. Recomendar una estrategia de solución, indicando porque se rechazan las otras.
4. Desarrollar una lista de prioridades para las características del producto.

El primer paso en la planeación de un proyecto de programación es preparar, en la terminología del cliente un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado definitivo del problema debe de incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

Hay que tomar en cuenta que el problema del cliente, desde de un punto de vista, tal vez sea un problema de nómina, de inventario, o de control de tráfico en red, y no un problema de algoritmos de clasificación, o de bases de datos relacionales.

La definición del problema requiere de un entendimiento cabal del dominio del problema y del entorno de este. Las técnicas para obtener este conocimiento, por parte de la persona de planeación, son entrevistas con el cliente, observación de las tareas problemáticas, y de desarrollo de las reales. Esta persona debe ser muy hábil en las técnicas de definiciones del problema, ya que distintos representantes del cliente tendrán diferentes puntos de vista, sesgos, y prejuicios que influirán en su percepción del área del problema. Además, los representantes del cliente quizás no estén familiarizados con las posibilidades que una computadora pueda ofrecer en su situación, rara vez son capaces de formular problemas de modo que sea factible un análisis lógico y algorítmico.



Algunas veces los sistemas computacionales se construyen para aliviar un síntoma y no la causa primaria de un problema. Esto ocurre cuando el problema real se entiende, pero no puede resolverse debido a circunstancias económicas, políticas o sociales, cuando el cliente no es capaz de comunicar el problema real o cuando la persona de planeación no entiende la explicación del cliente sobre el problema.

El segundo paso en la planeación de un proyecto de programación es determinar lo apropiado de una solución computacional. Además de ser eficaz en términos de costo, un sistema computacional debe aceptarse social y políticamente. Para ser eficiente en costo, un nuevo producto de programación debe proporcionar los mismos servicios e información que el sistema antiguo, usando menos tiempo y personal, o proporcionar servicios e información que antes eran inaccesibles. Un sistema que desplace muchos trabajadores puede ser económica y técnicamente posible, pero inaceptable social o políticamente para el usuario.

Habiendo determinado, por lo menos en una forma preliminar, que es apropiada una solución computarizada para el problema, la atención se centra en las funciones de los principales subsistemas del sistema computacional. Un sistema computacional esta formado por los subsistemas de personal, equipo y de productos de programación, más las interconexiones entre ellos. El primer subsistema incluye operadores, personal de mantenimiento y usuarios finales. El segundo comprende el equipo de cómputo y los dispositivos periféricos, y puede tener otros dispositivos como sensores y accionadores para control de procesos, o antenas rastreadoras y radares. El tercer subsistema contiene programas que pueden desarrollarse, mas programas que ya existen y que pueden emplearse (adaptandolos al nuestro)

Las funciones que debe realizar cada subsistema principal se deben identificar; se deben establecer las interacciones entre subsistemas y determinar las restricciones en el desarrollo y operación para cada subsistema principal. Las limitaciones especifican número y tipo de equipos, cantidad y habilidades del personal, y características del producto de programación con funcionamiento, precisión y nivel de confiabilidad.



La asignación precisa de funciones entre equipo, programación y personal puede dificultarse durante la planeación preliminar, tal vez sea necesario desarrollar primero un análisis detallado. No obstante, debe intentarse la definición preliminar de las funciones de los subsistemas principales.

El objetivo primario del análisis de datos es la parte medular para el diseño de la base de datos, y un enfoque disciplinado al catalogar los datos existentes en términos de las entidades y las relaciones que presentan. Sin tal entendimiento de la parte de la organización que esta siendo analizada, es mas difícil establecer si la base de datos será instalada eficientemente. El análisis de datos proporciona un medio efectivo para comunicarse con usuarios que no son profesionales en el mundo de la computación, ya que se dedica solamente a aquellos en que el usuario esta familiarizado.

Dado que los analistas de sistemas no trabajan como gerentes o empleados en los departamentos de usuarios (como mercadotecnia, compras, producción o contabilidad), no tienen los mismos conocimientos, hechos y detalles que los usuarios y gerentes de esas áreas.

Por consiguiente, el primer paso del analista es comprender la situación.

El análisis de sistemas es una actividad que engloba la mayoría de las tareas que integran la ingeniería de sistemas en una computadora. Se centra en todos los elementos del sistema (*hardware* y *software*) y se realiza de acuerdo a los siguientes objetivos:

1. Identificación de las necesidades
2. Estudio de viabilidad
3. Realizar un estudio técnico y económico
4. Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
5. Establecer restricciones de costo y de tiempo
6. Creación de una definición del sistema que sea la base para todo el trabajo de ingeniería subsecuente.



La fase de análisis es referida algunas veces como la formalización y definición de requerimientos, lo cual involucra el establecimiento de los objetivos y la documentación de estos requerimientos.

El análisis de datos debe ser realizado por un equipo que contenga a los usuarios, el departamento de desarrollo de sistemas, y el grupo de administración de datos.

El equipo de análisis de datos puede no intervenir en la fase de análisis de requerimientos, si esta fase esta limitada a entrevistas personales con diferentes niveles de gerencia y empleados claves en el procesamiento de bienes, servicios y organización de datos. El resultado de las entrevistas deben ser los diagramas de proceso. Los objetivos específicos y los requerimientos de la base de datos deben obtenerse de un nivel más alto de la organización.

El equipo de análisis de datos debe identificar las entidades que son necesarias para resolver el problema definido por el usuario. Durante las etapas iniciales del análisis de datos es posible que no se conozcan todos los atributos de todas las entidades. Sin embargo, a medida que estos se determinen, el equipo debe documentar la definición del atributo y su papel en un diccionario de datos.

Técnicas de análisis de requerimientos.

Las técnicas de análisis de requerimientos combinan procedimientos sistemáticos con una notación única para analizar los dominios de información funcional de un problema.

Se ha observado que el dominio de la información se caracteriza por los tres atributos siguientes:



- 1.- Flujo de datos
- 2.- Contenido de los datos
- 3.- Estructura de datos.

Los cuales son comunes en todas las técnicas de análisis de requerimientos.

Características comunes.

Aunque cada técnica y/o herramienta introduce una nueva notación y heurística de análisis, todas las técnicas pueden ser evaluadas en el contexto de las siguientes características comunes.

- 1.- Mecanismos para el análisis del dominio de la información.
- 2.- Método de representación funcional.
- 3.- Definición de interfaces.
- 4.- Mecanismos para subdividir el problema.
- 5.- Soporte de la abstracción.
- 6.- Representación de las visiones físicas y lógicas

Las diferentes técnicas y/o herramientas que se utilizan para el análisis del dominio de la información se conduce de forma diferente en cada uno de los mismos siendo factible, reconocer las siguientes guías comunes:

- Todas las técnicas se enfocan (directa o indirectamente) hacia el flujo de datos y al contenido o estructura de estos.
- El flujo se caracteriza generalmente por el contexto de las transformaciones (funciones) que se aplican para cambiar la entrada en la salida.
- El contenido de los datos puede representarse explícitamente utilizando un mecanismo de diccionarios.



- Las funciones se describen normalmente como procesos o transformaciones de la información. Cada función puede ser representada usando una notación específica (ejem. el diagrama de flujo de datos [D.F.D]).

La mayoría de las técnicas de análisis permiten al analista evaluar la representación física de un problema antes de derivar a la solución lógica. En general la notación se utiliza para representar ambos aspectos.

Técnicas orientadas al flujo de datos.

Para estudiar un sistema de información es recomendable que los analistas se cuestionen por lo general para determinar los requerimientos de los sistemas, lo siguiente:

- 1.- Que procesos integran al sistema.
- 2.- Que datos se utilizan en cada proceso.
- 3.- Que datos se almacenan.
- 4.- Cuales son los datos que entran y cuales los que salen del sistema.

La jerarquía de cada una de las preguntas se determina con las técnicas orientadas al flujo de datos.

Con estas técnicas, el analista de sistemas reconoce el papel central e importancia de los datos en una organización con los cuales al realizar un seguimiento de flujo de los mismos en un sistema obtiene una gran cantidad de información con la que se puede identificar, como se están llevando a cabo los objetivos de una determinada organización o empresa, siendo este el propósito del análisis por flujo de datos.

Lo anterior proporciona que al manejar las transacciones y completar las tareas los datos de entrada sean procesados, almacenados, consultados, utilizados, modificados y emitidos.



realizándose así procesos específicos, que forman parte del objetivo de una investigación de sistemas.

El análisis por flujo de datos es el encargado de estudiar el uso de estos en cada actividad, documentando esta información en las dos herramientas siguientes:

Diagrama de flujo de datos.

Muestra la relación entre los procesos, los datos y los almacenes.

El trabajo básico que hacen los sistemas es la transformación de entrada a salidas, como se muestra en la siguiente figura. 2.3.1.

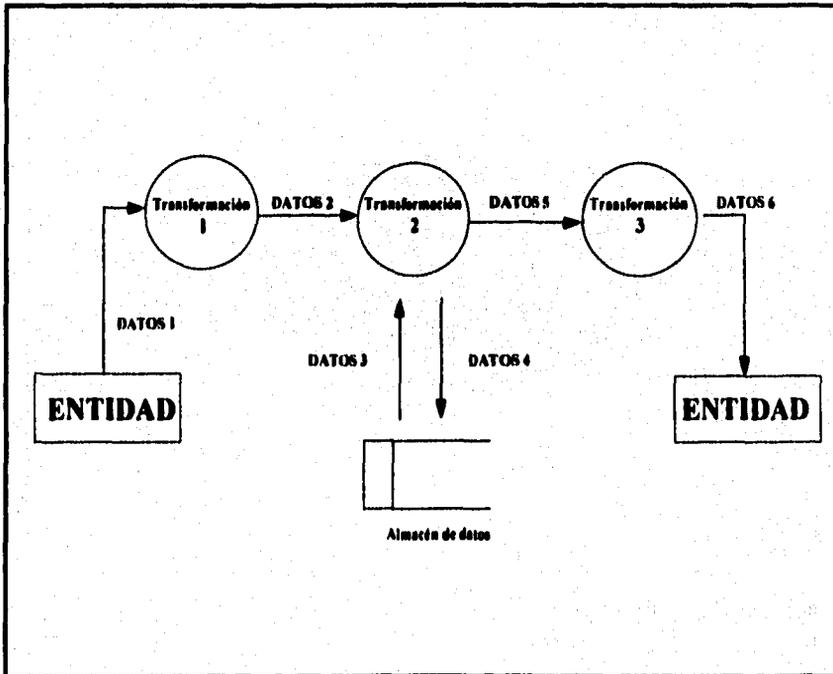


FIG. 2.3.1 Diagrama de Flujo de Datos



El D.F.D es una técnica gráfica que muestra a un sistema como una red de procesos conectados a cualquier otro por la trayectoria de datos, puede describir:

Como se relaciona el sistema con el medio ambiente.(Terminadores)

Como las partes de un sistema se relacionan unas con otras.

El flujo de información y las transformaciones que se aplican a los datos conforme se mueven de la entrada a la salida, obteniéndose de ellos el conocimiento claro para saber como entran, fluyen y salen del sistema.

La forma básica de un D.F.D se muestra en la figura 2.3.2.

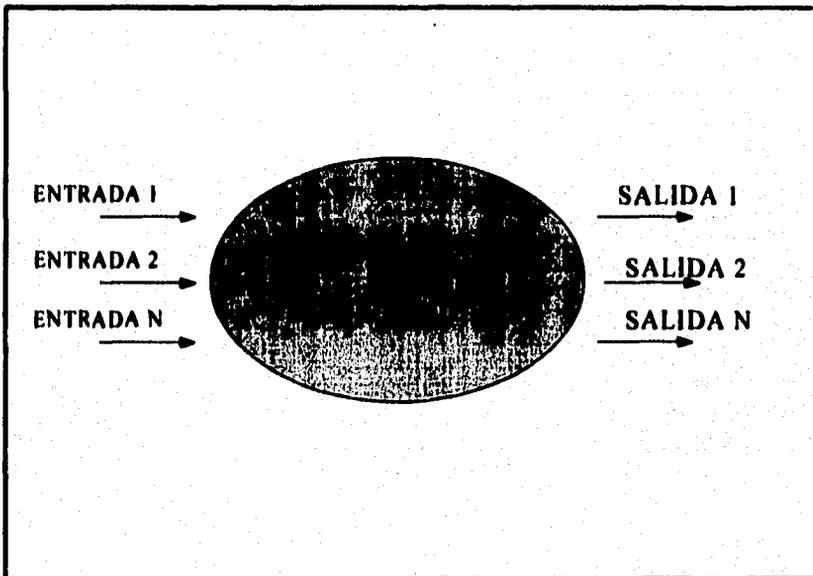


FIG. 2.3.2. Forma Básica del DFD

Los D.F.D son usados principalmente para:

- Representar un sistema o software a cualquier nivel de la abstracción.
- Subdividir un sistema.
- Analizar interfaces.
- Desarrollar una descomposición funcional.
- Comunicar a usuarios y diseñadores.
- Desarrollar especificaciones.
- Particionarse en niveles que representan el flujo incremental de información y detalle funcional.
- Consistencia entre procesos.

Su funcionamiento radica en no introducir en el "nivel inferior" nuevas entradas o salidas en el siguiente proceso, que no se hayan identificado en los diagramas de mayor nivel, sin embargo, si es posible identificar nuevos flujos de datos y almacenamiento de los mismos.

Ventajas

- a) Las actividades paralelas son útiles en diversos organismos, cuyas actividades sean múltiples y concurrentes; mediante el uso de actividades paralelas el D.F.D. permite que los analistas puedan representar diversas actividades que se dan en forma simultánea, mostrándolas con mayor exactitud, según vayan apareciendo, por lo anterior - el D.F.D - aporta una mayor visión del sistema para lograr un entendimiento entre el usuario y el analista.
- b) La concentración de los datos en los D.F.D.'s permite que los analistas tengan un gran conocimiento, de todo el sistema, permitiéndoles la plena identificación y descripción de los datos del área de aplicación, aparte de definir cuales datos son útiles para que sean almacenados.
- c) El análisis del flujo de datos permite a los analistas aislar las áreas de interés, estudiarlas y examinar los datos que entran al sistema, así como observar como se modifican cuando salen del mismo.



- d) Utilizan una notación bastante clara sencilla y accesible para las personas que no están familiarizadas con la terminología de computación.
- e) Permite utilizar una notación sencilla para describir en forma mas exacta la actividad de los procesos del sistema de información de manera que al examinar los diagramas puedan señalarles los problemas mas rápidamente de manera que se puedan corregir, antes que se inicie otro trabajo de diseño.
- f) Los D.F.D.'s se desarrollan y utilizan de manera progresiva, que va de lo general a lo particular para el sistema de interés; esto es, generamos D.F.D's a nivel general, también llamados de nivel 1, a partir de los cuales se generarán D.F.D's de un nivel de mayor detalle para tener la posibilidad de encontrar mas información sobre los procesos internos.
- g) Muestran las características lógicas de las aplicaciones, es decir, señalan que ocurre y cuando.
- h) Proporciona elementos para el análisis y desarrollo de especificaciones.
- i) Al descomponer un proceso en diversos niveles hasta llegar a las funciones simples y sus interconexiones nos permite ver que conjunto de procesos tiene menor numero de conexiones.
- j) La meta es crear un modelo del sistema en el cual todos los procesos, a todos los niveles, sean funciones simples.

Diccionario de datos.

El diccionario de datos D.D. es un segundo componente o técnica que consisten en describir el contenido de los elementos de información de los D.F.D's.

El D.D. contiene las definiciones de todos los datos mencionados en el D.F.D. en una especificación del proceso y en el propio diccionario de datos. (ver figura 2.3.3)

El D.D proporciona:



- a) Información adicional sobre el sistema.
- b) Auxilia al analista involucrado en la determinación de los requerimientos dentro del sistema.
- c) Describe la actividad o entidad que se esta estudiando.
- d) Almacena los detalles adicionales y las descripciones de todos los elementos incluidos en el conjunto de los D.F.D's, su flujo dentro del sistema, su estructura, así como los procesos de los sistemas.
- e) Define los elementos de información sin ambigüedad.
- f) Ilustra tanto los datos, como flujos a través del sistema y las transformaciones que se dan en los datos del mismo.

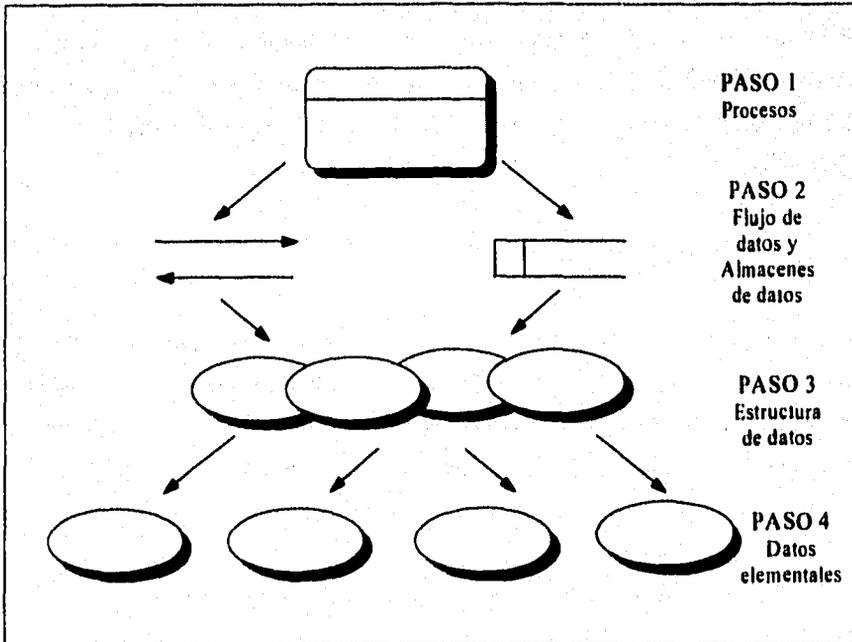


FIG.2.3.3 Diccionario de Datos



Ventajas.

- 1) Proporciona al usuario el tener un conjunto conciso de definiciones para todas las entidades de (los) proceso(s) que conforman al sistema de información que se esta estudiando.
- 2) Frecuentemente es la única fuente común de definiciones para los investigadores y usuarios, para ser utilizado como única fuente de respuestas a todas las preguntas que se relacionan con el formato y el contenido de los conjuntos de datos utilizados en el sistema.
- 3) Permite encontrar cuales son los flujos de datos que faltan, detectar definiciones duplicadas así como descubrir datos que no se utilicen en ningún proceso durante la investigación, así también evitara problemas posteriores en la determinación de los requerimientos en el diseño de sistemas.
- 4) El D.D. por si mismo, puede procesarse para revelar información adicional en : - Listados de elementos datos/estructura de datos. Conjunto completo de todos los datos utilizados en el sistema que se esta investigando y que incluyen nombres, descripción, longitud de campo y nombres alternos (alias).

Diagrama de descomposición funcional.

La etapa de flujo de datos muestra el empleo de estos en forma gráfica. Las herramientas utilizadas al seguir esta estrategia muestran todas las características esenciales del sistema y la forma en que se ajustan entre si. Puede ser difícil comprender en su totalidad un proceso de la empresa si se emplea para ello solo una descripción verbal; las herramientas para el flujo de datos ayudan a ilustrar los componentes esenciales de un sistema junto con sus interacciones.

El diagrama de descomposición funcional muestra las partes fundamentales de la estructura del sistema. En el se muestran las diferentes entidades y como podemos llegar a ellas de acuerdo al flujo de información ya sea manual o automatizada, incluyendo procesos y retroceso de información para el diseño.

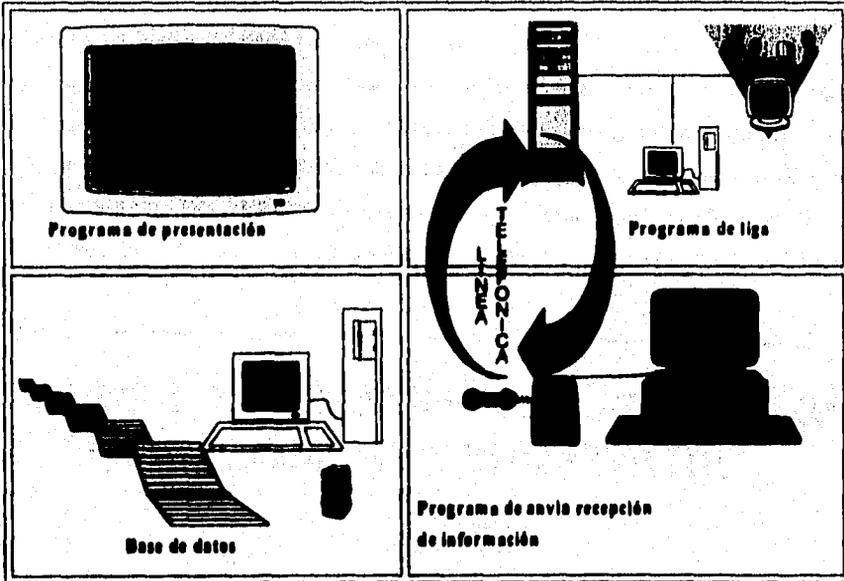


FIG.2.3.4 Diagrama de Descomposición Funcional del Sistema

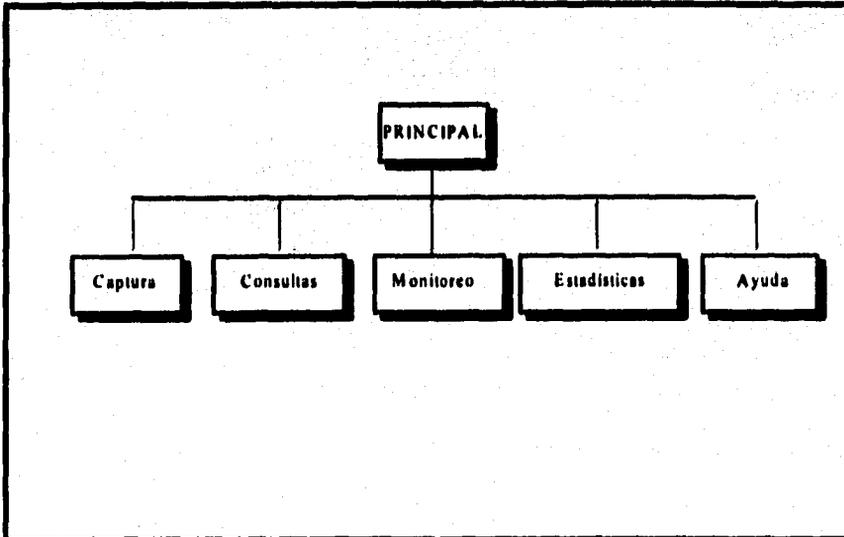


FIG.2.3.5 Módulos del Programa Principal



Los usuarios y otros de la empresa que forman parte del proceso bajo estudio comprenden con facilidad anotaciones sencillas. Por consiguiente, los analistas pueden trabajar con los usuarios y lograr que participen en el estudio de los diagramas de flujo de datos. Los usuarios pueden hacer sugerencias para modificar los diagramas con la finalidad de describir la actividad con mayor exactitud. Así mismo pueden examinar las gráficas y conocer con rapidez problemas, esto permite efectuar las correcciones necesarias antes de que comiencen otras tareas relacionada con el diseño.

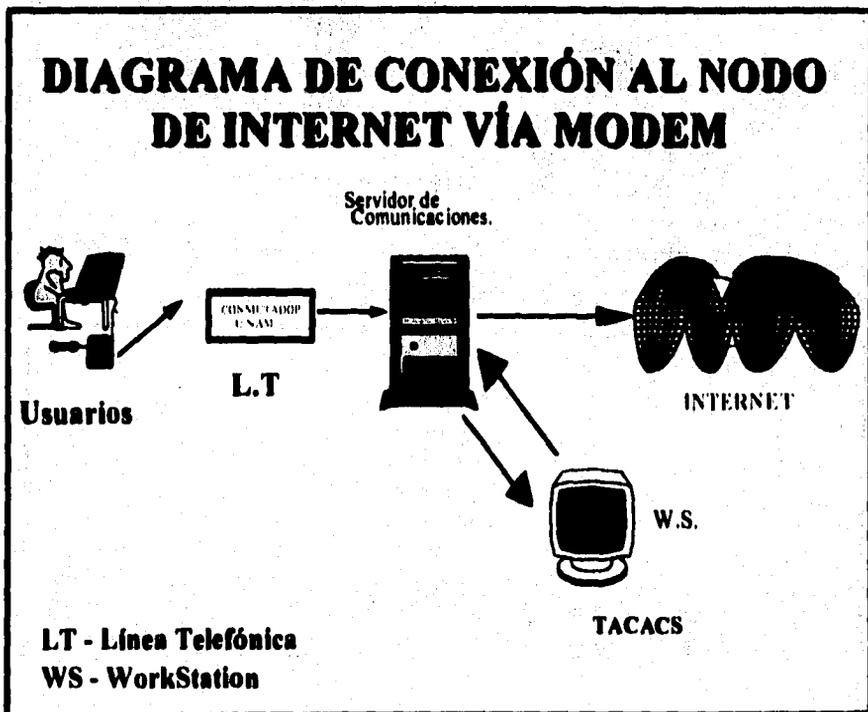


FIG. 2.3.6 Diagrama de Conexión Via Modem



2.4 REQUERIMIENTOS MÍNIMOS DEL SISTEMA DE HARDWARE Y SOFTWARE EN QUE DEBERÁ CORRER LA APLICACIÓN.

La UNAM cuenta con cinco conmutadores digitales o nodos principales, estos se encuentran en: Facultad de Arquitectura, Torre 2 de Humanidades, IIMAS (Instituto de Investigación en Matemáticas Aplicadas a Sistemas), DGSCA (Dirección General de Servicios de Cómputo Académico) y Zona Cultural .

Todos estos servidores están conectados entre sí con fibra óptica formando una malla, además a cada uno de estos conmutadores se le asocian otros nodos secundarios y también tienen la capacidad de permitir conexión de otros conmutadores digitales, con la diferencia de que estos están enlazados en forma de estrella ya sea por medio de fibra óptica, RDI (Red Digital Integrada), enlace microondas, cable coaxial, entre otros; siendo el nodo principal el centro de la misma. (Ver figura 2.4.1).

Los servidores de comunicaciones CS-500 que proporcionan líneas para el acceso remoto a RedUNAM son el de DGSCA (con 54 líneas disponibles), el conmutador del IIMAS (con 40 líneas) y el de la Torre 2 de Humanidades (también con 40 líneas disponibles); la forma como se ubican estas líneas es por medio de grupos llamados UCD (*Uniform Call Distribution*) Distribución Uniforme de Llamadas es decir, las líneas se van otorgando en forma cíclica sin importar si estas están ocupadas o no, en caso de estar ocupadas el acceso remoto tendrá que esperar hasta que se desocupe la línea o en todo caso volver a intentar la llamada.

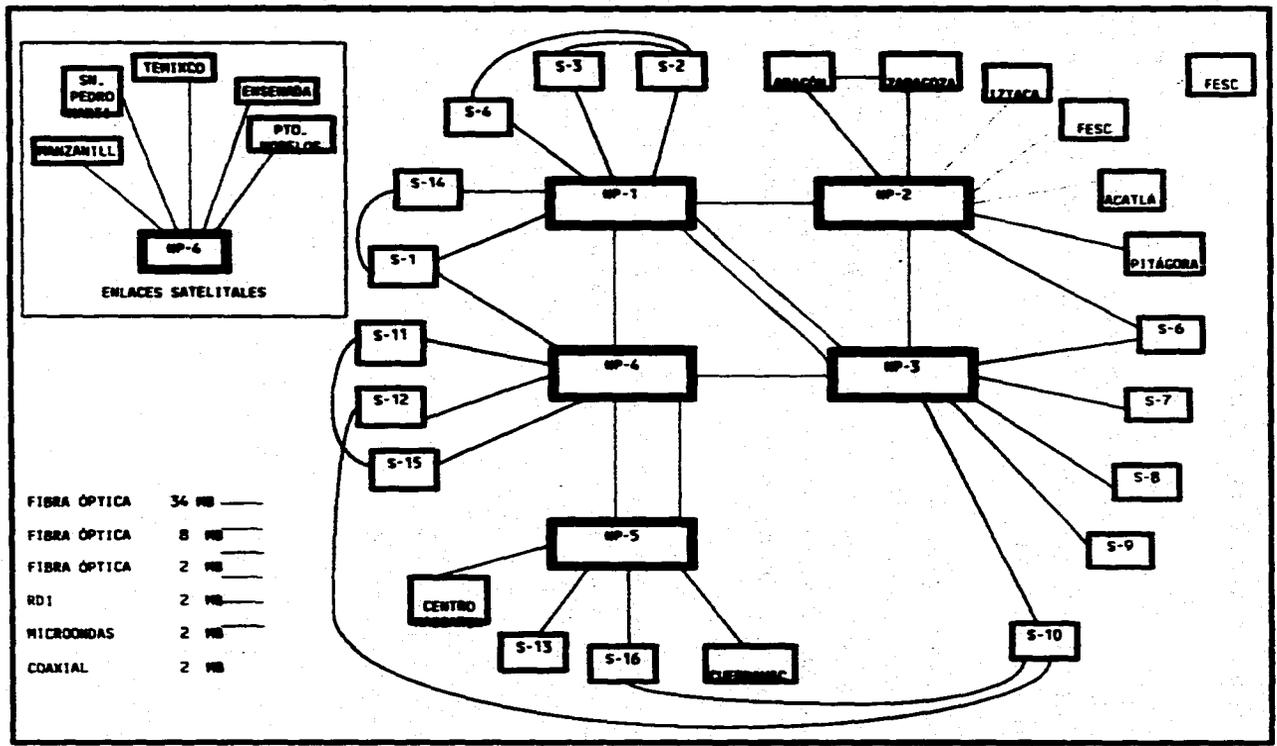


FIG: 2.4.1 Red Telefónica Digital



A continuación se explican las características del conmutador digital de DGSCA (Ver figura 2.4.2).

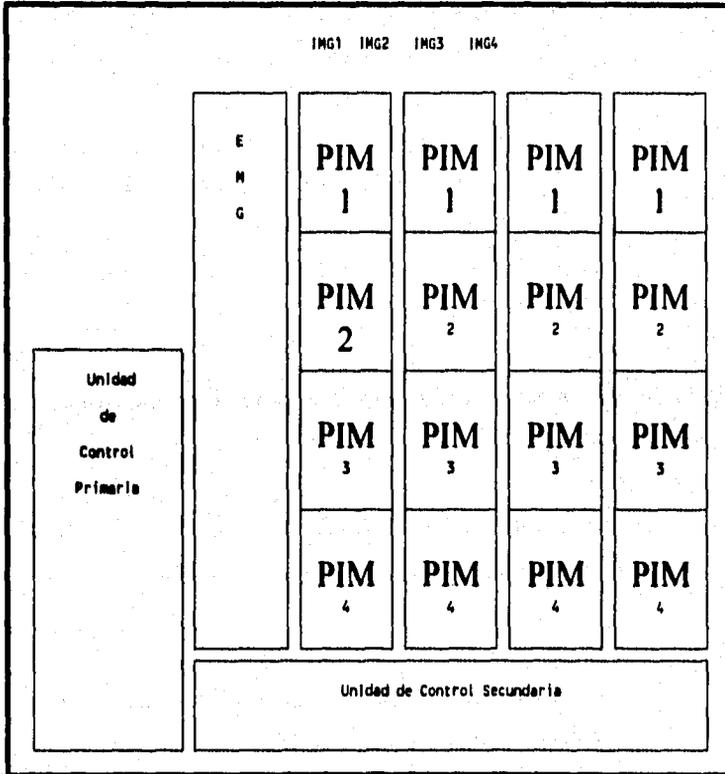


Fig. 2.4.2 Diagrama de Bloques del Conmutador Digital de DGSCA

Como se muestra en el diagrama de bloques de la figura 2.4.2 el conmutador esta fornado por una Unidad de Control General para todo el sistema, un EMG (Expansion Module Group) Grupo Modular de Expansion, cuatro IMG's (Interface Module Group) Grupo Modular de Interface. Cada IMG a su vez se divide en cuatro PIM's (Port Interface Module) Módulo de Interface de Puertos, estos PIM's contienen unidades llamadas slots o grupos a los cuales se



conectan las tarjetas de control que van a otorgar los servicios troncales para enlaces vía satélite o microondas, las líneas de teléfono normales y líneas de teléfono digitales. Además existe una Unidad de Control Secundaria en donde se maneja el número de procesadores, tarjetas de potencia, unidad de ruteo que controla a los PIM's, una fuente de 0-5 V de Corriente Directa para la lógica de control, los voltajes de línea para tono y para timbrado, entre otros.

A la ubicación o dirección de una línea disponible u ocupada se le conoce como **LEN** (*Line Equipment Number*) Numero de Equipo de Línea y esta formado por seis dígitos: los dos primeros indican el número de IMG, el tercer dígito se asocia al número de PIM, el cuarto y quinto corresponde al número de grupo o slot y el último dígito es el que indica el número de la línea.

Los requerimientos mínimos del sistema de hardware en que deberá correr la aplicación son los siguientes:

En cuanto al conmutador digital se utilizará el que actualmente tiene la UNAM, para aprovechar los recursos del mismo; el modelo que se empleará es el siguiente:

- Modelo NEAX 2400 Conmutador Digital
- Flexibilidad para soportar configuraciones como: **FL** (*Full Line Terminal*) Terminal de Línea Completa, **LR** (*Line Repeater*) Repetidor de Línea, **ADM** (*Add/Drop Mux*).
- Salida a X.25 para niveles altos de OSS
- Interface con IEEE 802.3
- Soporte completo para manejo de **GUI** (*Graphical User Interface*) Interface de Uso Gráfico
- Capacidad de transmisión: OC-48 (Combinaciones de DS3, STS-1, OC-3, OC-3c, OC-12, OC-12c)
- Sistema de seguridad para la conmutación a través de la configuración DS3

conectan las tarjetas de control que van a otorgar los servicios troncales para enlaces vía satélite o microondas, las líneas de teléfono normales y líneas de teléfono digitales. Además existe una Unidad de Control Secundaria en donde se maneja el número de procesadores, tarjetas de potencia, unidad de ruteo que controla a los PIM's, una fuente de 0-5 V de Corriente Directa para la lógica de control, los voltajes de línea para tono y para timbrado, entre otros.

A la ubicación o dirección de una línea disponible u ocupada se le conoce como **LEN** (*Line Equipment Number*) Numero de Equipo de Línea y esta formado por seis dígitos: los dos primeros indican el número de IMG, el tercer dígito se asocia al número de PIM, el cuarto y quinto corresponde al número de grupo o slot y el último dígito es el que indica el número de la línea.

Los requerimientos mínimos del sistema de hardware en que deberá correr la aplicación son los siguientes:

En cuanto al conmutador digital se utilizará el que actualmente tiene la UNAM, para aprovechar los recursos del mismo; el modelo que se empleará es el siguiente:

- Modelo NEAX 2400 Conmutador Digital
- Flexibilidad para soportar configuraciones como: **FL** (*Full Line Terminal*) Terminal de Línea Completa, **LR** (*Line Repeater*) Repetidor de Línea, **ADM** (*Add/Drop Mux*).
- Salida a X.25 para niveles altos de OSS
- Interface con IEEE 802.3
- Soporte completo para manejo de **GUI** (*Graphical User Interface*) Interface de Uso Gráfico
- Capacidad de transmisión: OC-48 (Combinaciones de DS3, S1S-1, OC-3, OC-3c, OC-12, OC-12c)
- Sistema de seguridad para la conmutación a través de la configuración DS3



- Sensibilidad en el Receptor de -30 dBm (BER= 10⁻¹¹)
- Máximo número de L.R: 30 nodos
- Requerimientos del ambiente donde se va a ubicar el equipo:
 - Rango para la temperatura máxima: 32 F a 104 F (0 C a 40 C)
 - Rango para la temperatura mínima: 32 F a 122 F (0 C a 50 C)
 - Humedad de 20 a 80% sin condensación
 - Altitud de -61 a 3,660 metros sobre el nivel del mar
- Requerimientos de potencia:
 - Potencia Primaria de -42 a -56 Vdc
 - Consumo de Potencia para L.R aproximadamente de 300 W

La estación de trabajo donde actualmente corre el administrador TACACS es la siguiente:

- Máquina SPARC SUN Station 486 20
- 32 MB de memoria RAM
- Memoria Virtual 106 MB
- 1GB en Disco Duro
- Sistema operativo SunOS Versión 5.4
- X-Windows Versión 3.4

La computadora que se requiere para el buen funcionamiento de nuestro sistema debe tener las siguientes características:

- IBM o compatible con procesador 486DX a 100 MHz
- 8 MB de RAM como mínimo
- Monitor Supe VGA
- Sistema Operativo MS-DOS Versión 6.20
- Windows 3.1 o superior



- Disco duro de 500 MB como mínimo
- Drive de 3.5
- Mouse

Concluyendo con el análisis anterior donde se presentaron las características del hardware que se requiere para que corra el software de aplicación que desarrollamos; procederemos a determinar los requerimientos de software que necesita nuestra aplicación en cuanto a: *Front-End*, Manejador de Base de Datos y el software para ligar aplicaciones de PC's con servidores.

FRONT-END: VISUAL BASIC 3.0

Requisitos para correr Visual Basic:

- computadora personal que contenga Windows versión 3.1 o posterior.
- 4 MB de memoria RAM como mínimo (se recomiendan 8 MB).
- Disco duro con capacidad mínima de 18 MB.
- Monitor VGA o de una mayor resolución.
- Configuración mínima 286.
- Mouse

MANEJADOR DE BASE DE DATOS: ACCESS

Se requiere de:

- Computadora personal que contenga Windows versión 3.1 o posterior.
- 4 MB de memoria RAM como mínimo (se recomiendan 8 MB).
- Espacio en disco duro de 12 MB como mínimo.
- Monitor a color o monocromático con tarjeta gráfica EGA o VGA o superior.
- Mouse



PROGRAMA DE LIGA

Para correr esta aplicación se requiere³ de:

- Computadora personal IBM o compatible 486
- Windows versión 3.1 o posterior.
- 8 MB de memoria RAM disponible como mínimo.
- Espacio en disco duro de 20 MB como mínimo.
- Monitor a color o monocromático con tarjeta gráfica EGA o VGA.
- Mouse
- Tarjeta de red.

En la figura 2.4.3 se muestra la memoria RAM recomendada y el espacio en disco duro que se requiere para trabajar con Visual Basic, Access, el programa de comunicación, la aplicación y además se muestran las características recomendadas para un adecuado funcionamiento de la aplicación.

Ya que el sistema esta sujeto a un crecimiento en función del numero de registros de la Base da Datos, el administrador en un momento dado puede hacer cambios o darle mantenimiento a la misma o al sistema en sí, para ello es recomendable tener instalados los paquetes completos de Visual Basic y Access en la máquina donde va a correr la aplicación. Además se tienen que considerar los requerimientos de hardware para el sistema operativo y para el ambiente de trabajo (en nuestro caso Windows).

³ Basado en los requerimientos que presentan los programas comerciales



	MEMORIA R A M RECOMENDADA	ESPACIO EN DISCO DURO
VISUAL BASIC	8 MB	18 MB
ACCESS	8 MB	12 MB
COMUNICACIÓN ENTRE T_x Y R_x	8 MB	20 MB
APLICACIÓN	15 MB	18 MB
TOTAL	16 MB	500 MB

FIG. 2.4.2. Resumen de requerimientos mínimos de hardware en que deberá correr la aplicación



2.5 PROPUESTAS DE SOLUCIÓN Y ELECCIÓN DE LA ÓPTIMA

Una vez planteada la problemática y analizados los requerimientos del usuario tanto de hardware como de software, mostramos a continuación las diversas propuestas de solución, dando una breve descripción de cada una de ellas para posteriormente seleccionar la óptima.

Propuesta 1.

Esta opción consistía en la modificación del código de TACACS, de tal forma que además de llevar a cabo el control y monitoreo de los logins y passwords, también se realice la verificación de las mismas a Internet, vía modem a través de RedUNAM para hacer la validación de las claves (propietario).

Esta opción implica mucho tiempo para su análisis y desarrollo, además de que el costo de adquisición del código fuente para cada módulo de expansión del sistema es demasiado elevado. La disponibilidad del código fuente es casi nula, ya que aunque se trata de un software de dominio público, este sólo lo es en su versión ejecutable. Una posible desavenencia que tiene esta solución es la de correr el riesgo de cambiar partes del programa original y perder algunas de la aplicaciones que proporciona actualmente.

Propuesta 2.

Esta solución consiste en realizar un sistema comenzando desde cero, que sustituya las funciones que realiza el actual, utilizando para ello nuevas herramientas para su desarrollo, como lo pueden ser Unix, Lenguaje C, Visual Basic o bien cualquier otra que sea manejada en DGSCA.

Esta opción es bastante llamativa, sin embargo nos enfrenta a una serie de problemas, principalmente el factor tiempo, es por esto que se pensó que si la finalidad era crear un software



similar al que opera, tal vez sería mas factible aprovechar lo que se tiene funcionando hasta el momento.

Propuesta 3.

Crear un sistema que corra bajo un ambiente de computadora personal con la finalidad de no saturar con procesos simultáneos al servidor (estación de trabajo donde corre el administrador tacacs), viendo como cajas negras los módulos del sistema actual.(ej. tacacs).

Evidentemente esta es una propuesta bastante interesante, ya que se baja el costo de la aplicación, pero a cambio encontramos que la interface entre los módulos de la aplicación y esencialmente el sistema TACACS no es directa, de hecho es bastante compleja, ya que sus salidas son encriptadas; lo que repercutiría en una inversión mayor de tiempo y personal dedicado a esta tarea.

Propuesta 4.

Dado que esta institución cuenta con la infraestructura básica para hacer conexión a Internet para bajar información a través de los servicios mencionados en el capítulo II, se propone aprovechar la tecnología que sea de dominio público existente en esta red, como parte del sistema rescatando las entradas y salidas, procediendo a diseñar solo interfaces.

Esta propuesta es factible desde el punto de vista de que se cuenta con todos los recursos para traer la información necesaria, sin embargo como la aplicación es muy específica normalmente las compañías que ponen a disposición pública sus aplicaciones sólo lo hacen de una manera parcial, lo que trae como consecuencia una alza en el costo en el caso de que se desee adquirir todos los módulos(ej. VBscript de Microsoft).



Propuesta 5.

Esta propuesta es la más factible para realizar, consiste en que una vez conocido el funcionamiento del servidor de comunicaciones, el sistema taacas, y en fin de todo el equipo empleado, se cree nuestra propia base de datos para almacenar toda la información correspondiente a los usuarios y sus claves como lo hace actualmente taacas, agregándole un *Front-End* que sea gráfico y sumamente amigable al usuario (tanto al administrador del sistema así como al usuario final), cuestión que no es favorable en el sistema actual; que esté basado en una arquitectura cliente servidor⁴, lo que traería como consecuencia que se tuviera que programar tanto la parte del servicio como la del cliente, (sin embargo a pesar de lo que pudiera pensarse el tiempo requerido para este punto es menor que en el resto de las opciones) esto con la finalidad de dejar un sistema compatible con la visión que se tiene en DGSCA⁵ y dentro de la UNAM, todo esto con el objetivo de dejar sentadas las bases para aquellas personas que dentro de algún tiempo deseen realizar mejoras en el sistema y que la aplicación se pueda migrar sin mayor esfuerzo con la ayuda de herramientas que seguramente ya estarán liberadas al público o porque no, tal vez ya sean gratuitas y fáciles de obtener de Internet. Ver figura 2.5.1

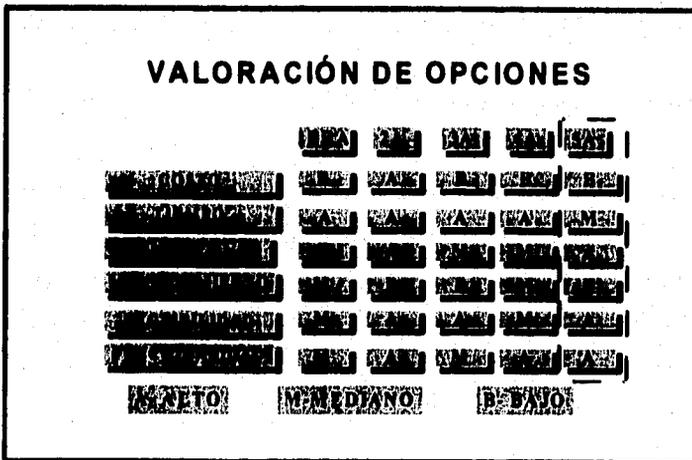


Fig. 2.5.1 Valoración de Opciones

⁴ver apéndice I al final del trabajo.
⁵poner visión



Es muy importante hacer notar que el objetivo es automatizar los procesos que ellos llevan, únicamente debe presentarse un nuevo esquema de hacer lo mismo de manera más eficiente y no cambiar de fondo los procedimientos que deben realizarse.

Además deben considerarse los siguientes puntos relevantes en la selección de la mejor opción, ya que ellos permiten definir de manera más precisa los alcances de una solución :

TECNOLOGÍA

Desde un punto de vista técnico se debe buscar que la solución a la necesidad de elaborar un sistema de monitoreo de claves de acceso a Internet este montado en una infraestructura de nueva tecnología, es decir, es absurdo pensar en ofrecer un sistema desarrollado en ambientes obsoletos o fuera de tiempo, porque al intentar dar mantenimiento al sistema o cuando sea necesario la generación de un nuevo módulo debemos asegurar la existencia de herramientas adecuadas, herramientas nuevas que de alguna manera permitan mantener de forma relativamente fácil al módulo principal del sistema. Es decir, al realizar un proyecto de este tipo se deben evaluar las nuevas opciones disponibles en el mercado para el desarrollo del sistema planteado.

HERRAMIENTAS DE APOYO

Al definir la base en la cual se desarrollará el sistema, es muy importante pensar en el tipo de rutinas, librerías, utilerías que existen para la herramienta base a utilizar, porque no se trata de empezar de cero, debe buscarse la integración de estas ayudas disponibles en el mercado para disminuir los tiempos de desarrollo y a su vez el monto de la inversión a realizar, es mejor utilizar funciones ya elaboradas para el manejo de pantallas, de menús de reportes, que intentar desarrollarlos desde sus inicios.

Debemos asegurar que dichas herramientas de apoyo permitan dar al sistema los niveles de seguridad y calidad que el Centro de Cómputo requiere.



INNOVACIÓN

Las herramientas que se utilicen en el desarrollo de la aplicación deben permitir dar al sistema aspectos innovadores que llamen la atención del usuario, estas innovaciones no deben requerir de mucho tiempo de trabajo o de mucha inversión, lo que se requiere es ofrecer a los usuarios las características del sistema como ventajas ante los demás sistemas que puedan existir en el mercado, es necesario captar la atención de los usuarios con el sistema, y la mejor forma de hacerlo es ofreciendo características que ningún otro producto tenga en el mercado y además que representen un ahorro en tiempo y costo.

INTERFACE AMIGABLE

Uno de los objetivos al ofrecer este sistema es precisamente darle al usuario una herramienta de ambiente gráfico, para llevar a cabo una captura sencilla de la información referente al propietario de la clave correspondiente, de forma tal que no requiera de un largo proceso de aprendizaje para poner en práctica la aplicación.

Es por ello que debe tomarse en cuenta para la selección de la mejor alternativa a aquellas herramientas y plataformas de desarrollo que cumplan con este objetivo.

Desde el inicio de la relación entre el usuario y la aplicación deberá existir una intuición natural en el uso del programa de forma que el usuario sienta que realmente la aplicación lo ayuda en la realización de su trabajo.

RELACIÓN COSTO-BENEFICIO

La plataforma de desarrollo y los recursos mínimos de operación para la aplicación muchas veces dependen de las herramientas en que ha sido desarrollada, es por ello que debemos tomar en cuenta que existe en el Centro de Cómputo una infraestructura de equipos bastante



aceptable y para los cuales se podría desarrollar una aplicación eficiente que consuma bastantes recursos, sin que ello represente un problema para el mismo, pero no debemos olvidar a aquellos usuarios que también realizan los mismos procesos y que no cuentan con equipos tan sofisticados. Tomando en cuenta estos puntos debe definirse una plataforma de desarrollo y herramientas en un punto intermedio, es decir que no requieren de gran cantidad de recursos pero sí con un mínimo que permita ofrecer una aplicación eficiente.

TIEMPO DE RESPUESTA

Los tiempos de respuesta de una aplicación dependen de los lenguajes y herramientas utilizados en su desarrollo, así como de la forma en que se lleve a cabo la programación y las características del equipo en que sea ejecutado. Lo importante aquí es tomar en cuenta que la herramienta para el desarrollo debe permitir dar un tiempo óptimo de respuesta a los usuarios de acuerdo a los equipos con los que cuente cada uno de ellos.

COMPATIBILIDAD

Las aplicaciones deben permitir al usuario la interacción con las más comunes dentro del mismo ambiente de operación. Esto se debe a que existen aplicaciones específicas para cada proceso a realizar dentro del monitoreo, por ejemplo, un sistema de monitoreo deberá poder comunicarse de alguna forma con un programa de validación de datos.

A continuación se muestra en la figura 2.5.2 una tabla con las características, plataforma, herramientas y software para ligar aplicaciones entre PC's y servidor, que se considera cumplen con los puntos anteriores. Más adelante se presenta la justificación de porqué usar cada una de ellas.



PLATAFORMA DE DESARROLLO

Las necesidades de cómputo para esta aplicación pueden cubrirse con Microcomputadoras, buscando cumplir el objetivo de proporcionar una aplicación capaz de solucionar el problema real existente en el acceso a Internet, consideramos que esta es la plataforma más adecuada, desde el punto de vista técnico para desarrollar la aplicación.

CARACTERÍSTICA	SELECCIÓN
Plataforma de Desarrollo	Computadoras Personales
Ambiente Operativo para la aplicación	Microsoft Windows
Software para liga de PC's con Servidores	C ó Unix
Manejador de Base de Datos	Access
Herramientas de Apoyo	Visual Basic

FIG. 2.5.2. Selección del software

AMBIENTE OPERATIVO PARA LA APLICACIÓN

Hasta hace algunas años, el sistema operativo MS-DOS venía siendo el más común instalado en las Microcomputadoras, de hecho, actualmente se encuentra instalado en la mayoría



de ellas, pero ahora existen tendencias a desaparecerlo lo cual ha sido originado por diferentes causas. las aplicaciones desarrolladas en DOS fueron muy exitosas y el usuario aprendió a utilizarlas ya que le permitían realizar su trabajo de una manera más rápida, sin embargo la misma necesidad de contar con aplicaciones extensas para mercados verticales muy específicos como la construcción, administración, contabilidad, etc., trajo consigo que dichas aplicaciones se hicieran más difíciles de entender, por lo que el usuario debía tomarse un tiempo de entrenamiento para saber utilizar cada aplicación, estos entrenamientos representaban un tiempo sin productividad en las empresas así como también implicaban un costo.

Con la aparición en el mercado del ambiente operativo MS-WINDOWS en un principio no se lograron resultados que pudiesen solucionar estos problemas, fue hasta la versión 3.0 donde Microsoft dio un giro completo a la interface buscando que el usuario pudiese utilizar las aplicaciones casi de inmediato, la interface gráfica de MS-WINDOWS considerando la utilización de los iconos (pequeñas imágenes representativas de la aplicación), comandos en forma de botones hicieron más fácil el aprendizaje para los usuarios.

Si a esto agregamos que a partir de Windows versión 3.0 se estableció un estándar para todas las aplicaciones que quisieran ejecutarse bajo este ambiente, buscando así que el usuario al saber operar Windows ya conociera de manera general el funcionamiento de cualquier aplicación Windows.

Actualmente se cuenta en el mercado con la versión 3.1 de Windows y con la nueva versión, Windows 95, en la cual se ofrecen nuevas características de interface que permiten al usuario, utilizar al máximo los recursos de su máquina de una manera mucho más fácil. Debemos tomar en cuenta también que el mismo mercado ha estado desplazando el uso de aplicaciones para MS-DOS por lo cual no sería adecuado a estas alturas pensar en desarrollar una aplicación en modo texto.

Se seleccionó desarrollar la aplicación en Windows versión 3.1 porque nos permitirá desarrollar un producto amigable con una buena interface al usuario.



Si sin importar las herramientas de desarrollo que se utilicen, la programación en Windows es diferente de la forma en que tradicionalmente se venía programando.

Antes de comenzar a hacer programas bajo Windows, es necesario conocer el funcionamiento fundamental de Windows desde el punto de vista técnico.

Normalmente los programas en MS-DOS escriben directamente en la memoria del video y en la memoria de los puertos de impresión, la desventaja de esta técnica es que era necesario sustituir los drivers (archivos de control) utilizados para cada tipo de video y para cada tipo de impresora, lo cual de alguna manera hace no muy eficiente a los programas.

Windows introduce un nuevo concepto GDI Graphics Device Interface (Dispositivos de Interface Gráfica), los cuales son manejados por Windows sin necesidad de que cada programa deba configurar las características de video y/o impresora.

Para los programas en MS-DOS es necesario codificar constantes, inicializaciones y datos o deben manejarse archivos de datos separados, que se leen al momento de ejecución del programa, ahora Windows permite almacenar este tipo de datos en un archivo de recursos con un formato específico, los cuales no se compilan junto con el demás código, sino que únicamente son incluidos en los programas al momento de hacer el ligado.

El uso de memoria por los programas para MS-DOS fue siempre limitada a 640 KB. incluso el tamaño de los programas se encontraba igualmente limitado, se requerían varios manejadores de memoria extendida/expandida (memoria superior a los 640 KB), ahora con Windows se cuenta con herramientas y características que nos permiten obtener un mejor aprovechamiento de esta memoria de una manera confiable y fácil.

En el ambiente de MS-DOS todos los módulos de cada programa son ligados o unidos para generar un solo archivo que cuente con todas las funciones necesarias para su ejecución, en



Windows aparece el concepto de Librerías Dinámicas (DLL) las cuales ayudan a ahorrar memoria y espacio en disco, porque muchas de ellas son utilizadas por diferentes aplicaciones, además permiten dar una mayor modularidad a la programación porque las librerías se compilan y se prueban por separado.

Otras ventajas que ofrece Windows son el concepto de OLE y *TrueType*, la primera le permite al usuario integrar objetos de diferentes aplicaciones en una sola, es decir generar documentos o archivos que incluyan características de diferentes aplicaciones. El *TrueType* es un tipo especial de letras o caracteres que siempre tendrán la misma forma independientemente de la escala en la que se impriman sobre pantalla o impresora.

LENGUAJE DE DESARROLLO

Al pensar en el desarrollo de un producto debe buscarse un lenguaje que permita cubrir todas las necesidades del usuario. Actualmente existen en el mercado una gran variedad de herramientas para la construcción de aplicaciones. En nuestro caso, en una primera instancia se pretende desarrollar el programa liga en Unix.

Unix se ha encontrado como una de las mejores alternativas, no implica que sea la mejor ya que el tiempo que toma desarrollar una aplicación en este lenguaje requiere de mayor capacitación de los desarrolladores y toma mayor tiempo, sin embargo de acuerdo a las características que proporciona, consideramos que bien vale la pena el desarrollo de la aplicación con esta herramienta.

MANEJADOR DE BASE DE DATOS

El hecho de pensar en desarrollar un producto con posibilidades de trabajar con una gran cantidad de información, nos define un camino a seguir en cuanto a la utilización de las librerías de un manejador de base de datos (*engine*), por tanto es lógico pensar en hacer uso de las rutinas de una manera indirecta, esto es, agregar código en una función de la aplicación que haga uso de las funciones de un manejador de base de datos determinado.



Algunas de las características deseables más relevantes en el manejador de bases de datos, son las siguientes:

- Manejo en Red
- Sencillez de uso
- Portabilidad
- Volumen de Información que se pueda manejar
- Documentación
- Soporte Técnico
- Disponibilidad del Código Fuente
- Facilidad para migrar a otros ambientes

De acuerdo a las características anteriores decidimos usar ACCESS, que es un manejador de base de datos poderoso, proporciona la ventaja de manejar volúmenes de información grandes, además de tener las características de manejo en red, sencillez de uso y portabilidad, entre otras.

MANEJO EN RED

Access tiene la posibilidad de funcionar en un ambiente de red, para que en un futuro sea posible implementar su funcionamiento. Este manejador mantiene el control, haciendo su uso transparente y sencillo para el desarrollador.

SENCILLEZ DE USO

En el programa de demostración se puede apreciar de una manera muy sencilla el uso de las librerías de acceso a datos, ya que la base de datos se maneja como una sola entidad y el manejador se encarga automáticamente de intercalar las llaves en una forma muy sencilla, disminuyendo así los tiempos de desarrollo.



DOCUMENTACIÓN

El manual de referencia se encuentra bastante claro y entendible, se encuentra la documentación para códigos de error, acceso a datos, búsquedas, ingreso de registros nuevos y se encuentran ejemplos de cómo utilizarlos.

LENGUAJES SOPORTADOS

Access soporta ser utilizado por C, C++ y Pascal, por lo cual no existe inconveniente alguno para desarrollar la aplicación en alguno de estos lenguajes.

HERRAMIENTAS DE APOYO

El *Front-End* que se usará para el desarrollo de la aplicación es Visual Basic, porque permite el acceso a bases de datos, manejo de multimedia, es una interfaz amigable para el usuario, además de ser el que menos requiere de capacidad en memoria RAM y espacio libre en disco duro.

CONCLUSIÓN

Considerando todas las características anteriores para las herramientas en evaluación fue como se decidió utilizar las siguientes, que de alguna manera permiten obtener grandes beneficios y que fueron consideradas desde el inicio de nuestro análisis de herramientas.

El trabajar bajo estas herramientas y con sus características nos permiten obtener la calidad que es necesaria para el producto a desarrollar, de entre los que destacan:



- Ambiente amigable para el usuario
- Fácil aprendizaje y operación intuitiva que cumple con el estándar de aplicaciones en su tipo en el mercado. (Aplicaciones para Windows).
- Utilización de herramientas de gran poder para mejorar el funcionamiento de la aplicación, con la seguridad de que dichas herramientas son actualmente mejoradas para contar en el futuro con versiones más adelantadas de las mismas, que permitan proporcionar un fácil y adecuado mantenimiento a las aplicaciones.
- Un costo bajo, lo cual representa uno de los puntos más importantes que deben cubrirse.

CAPITULO III

DESARROLLO DEL SISTEMA





3.1. ESPECIFICACIONES DE DISEÑO.

El diseño de sistemas tiene dos etapas: **el diseño lógico y la construcción física del sistema.** Cuando el analista formula el diseño lógico, escribe las especificaciones detalladas del nuevo sistema, es decir aquellas que describen sus características: salidas, entradas, archivos, bases de datos y los gran procedimientos, todo en una forma que satisfaga los requerimientos del proyecto. El conjunto formado por todas estas características recibe el nombre de especificaciones de diseño del sistema.

Ver figura 3.1.1.

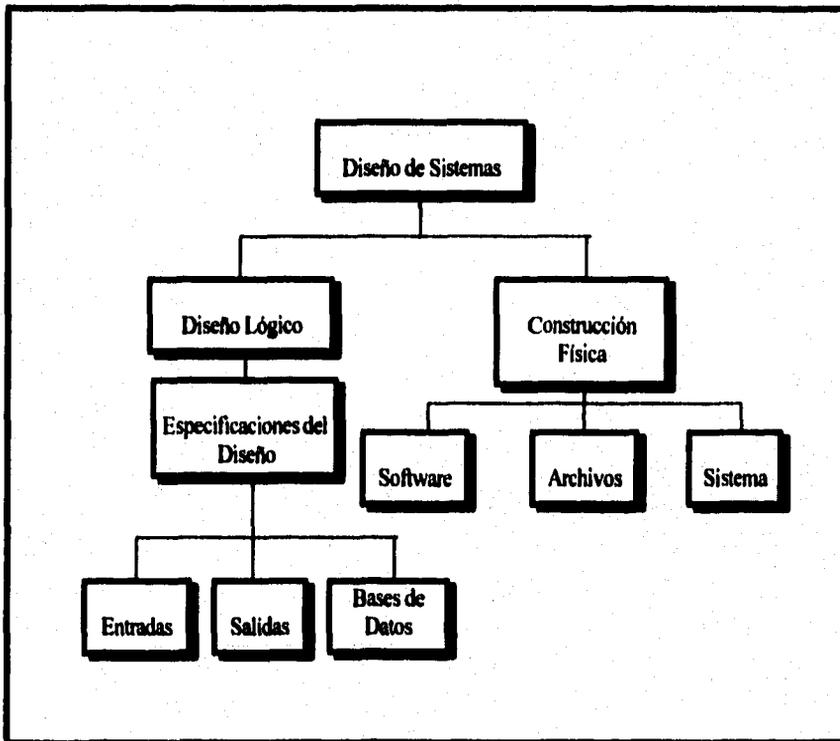


FIG.3.1.1. Elementos del Diseño de Sistema.



El diseño lógico de un sistema de información es similar al proyecto de ingeniería de un automóvil: muestra las características más sobresalientes (como el motor, la transmisión y el espacio para los pasajeros) y la relación que guardan entre sí (dónde se conectan unos a otros o cuál es la separación que existe entre las puertas). Los reportes y salidas generadas por el analista son similares a los componentes de diseño del ingeniero. Los procedimientos y datos se enlazan entre sí para producir un sistema que trabaja.

La construcción física, que es la siguiente actividad después del diseño lógico, produce el *software*, los archivos y un sistema que funciona. Las especificaciones de diseño indican a los programadores lo que el sistema debe hacer. A su vez, los programadores escriben programas que aceptan la entrada proporcionada por los usuarios, procesan los datos, producen los reportes y guardan los datos en los archivos.

El diseño está formado por instrucciones de programa, escritas en un lenguaje de programación. Durante la construcción física, los programadores escriben las instrucciones necesarias del programa.

Un objetivo fundamental en el diseño de un sistema de información es asegurar que éste brinde apoyo a la actividad de la empresa para la que fue desarrollado.

En otras palabras, la tecnología de cómputo y comunicaciones especificada en el diseño siempre debe tener un papel secundario en relación con los resultados que se pretende que el sistema proporcione.

Por ejemplo si es esencial para una organización mover la información con mucha rapidez para mantener su competitividad, entonces las especificaciones de diseño del sistema de información deben girar alrededor de este objetivo esencial de la empresa. En este ambiente, un sistema que procesa la información con lentitud causará daños a la organización. Imagine que un sistema de reservación de vuelos que no dice con prontitud a sus usuarios que asientos quedan disponibles en un determinado vuelo o un sistema de cajero automático en un banco que no permite a los clientes conocer con rapidez el saldo de sus cuentas. Para estas organizaciones, dichas actividades son esenciales y los sistemas de información deben apoyarlas.



De manera similar, el diseño tiene que ajustarse a la forma que trabaja la empresa. Si el sistema de ventas de una compañía es diseñado para trabajar mejor con pedidos que son pagados en efectivo, cuando de hecho la empresa tiene una política de crédito para las ventas, entonces la gerencia no estará muy feliz con el sistema, y lo mismo sucederá con los clientes. Aún a pesar de que el sistema de información trabaje bien desde el punto de vista técnico, no se ajusta a las actividades de la empresa.

Los ejemplos anteriores ilustran la importancia de adecuar el sistema a las necesidades de la organización, un objetivo que debe servir virtualmente de guía a todas las decisiones de diseño.

Durante el diseño, los requerimientos del usuario se trasladan en características del sistema.

Se dice que un sistema de información satisface las necesidades del usuario si cumple con lo siguiente, (ver figura 3.1.2):

- Realiza en forma apropiada los procedimientos correctos.
- Presenta información e instrucciones en una forma aceptable y efectiva.
- Produce resultados exactos.
- Proporciona una interface y métodos de interacción aceptables.
- Es percibido por los usuarios como un sistema confiable.

Especificaciones del Software.

Al igual que con las características de un sistema de información, el *software* también debe ser diseñado con cuidado. El diseño del sistema incluye la formulación de las especificaciones del *software*.

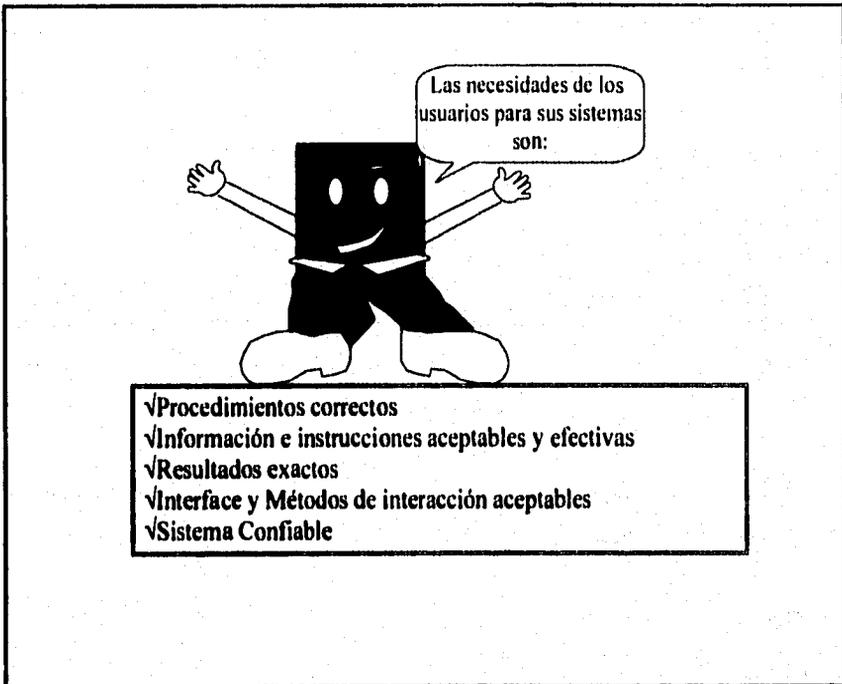


FIG.3.1.2. Necesidades de los usuarios en un Sistema

Estas especificaciones establecen las funciones de entrada, salida y procesamientos así como los algoritmos necesarios para efectuarlas.

Los módulos de *software* junto con las rutinas, se enfocan sobre lo que cada función realiza; así mismo, se especifican los procedimientos necesarios para llevar a cabo dichas funciones. La selección de lenguajes de programación, paquetes de *software* y utilerías se efectúa durante el proceso de diseño lógico y las recomendaciones se incluyen como parte de las especificaciones del *software*.

Como puede observarse, los objetivos del diseño de sistemas son muy amplios y afectan aspectos tanto de la aplicación como de la organización en la que será utilizado el sistema. Por consecuencia, no debe sorprender el hecho de que los grupos de sistemas de información mejor manejados, también



mantengan estándares para el desarrollo de sistemas. Las especificaciones de diseño se establecen dentro del marco fijador de los estándares.

Los siguientes son ejemplos de áreas incluidas en el diseño de estándares:

Estándares para datos

Son los lineamientos para asignar nombres a los datos y especificar su longitud y tipo. Estos lineamientos son utilizados por todas las aplicaciones desarrolladas por el grupo de sistemas de información. Con frecuencia están contenidas en el diccionario de datos.

Estas también pueden ser abreviaturas y designaciones formales para describir actividades y entidades dentro de la organización por ejemplo, clasificación de clientes y tipo de transacciones.

Estándares Estructurales

Son los lineamientos sobre como estructurar el *software* y el sistema. Así como Políticas para dividir el *software* en módulos, para la codificación estructurada de la relación existente entre los componentes del sistema. Pueden incluir estándares sobre la longitud del programa y lineamientos para volver a utilizar los módulos del *software*.

Estándares de Documentación

Descripciones de las características del diseño de sistemas, de la relación entre componentes y de las características de relación que pueden ser revisadas para conocer los detalles de la aplicación.

Con el objeto de garantizar que la aplicación cumpla con los estándares, muchas organizaciones cuentan con un grupo de control de calidad que tiene la responsabilidad de revisar todas las especificaciones del diseño de sistemas de información, así como el propio sistema una vez terminado.



¿Qué características son las que se deben diseñar?

Las especificaciones de diseño describen las características del sistema, los componentes o elementos del sistema y en la forma en que estos aparecerán ante el usuario.

Para muchos usuarios, el éxito de un sistema esta relacionado con la creencia de que tenga las características adecuadas.

Esta sección describe las características que debe diseñar el analista de sistemas. Pero antes de considerarlas, es conveniente primero aclarar los elementos que tienen que tomarse en cuenta en las especificaciones normales de diseño.

Elementos del Diseño

Los componentes de un sistema de información descritos durante el análisis de requerimientos, son el punto focal del diseño de sistemas. Los analistas deben diseñar los siguientes elementos:

Flujos de Datos

Los movimientos de datos hacia, o alrededor del sistema (Ver la figura 3.1.3.a).

Almacenes de Datos

Conjuntos temporales ó permanentes de datos (figura 3.1.3.b).

Procesos

Métodos y rutinas para utilizar el sistema de información y lograr con ello los resultados esperados (figura . 3.1.3.c.)



Controles

Estándares y lineamientos para determinar si las actividades están ocurriendo en la forma anticipada o aceptada, es decir si se encuentran bajo control. Así mismo, debe especificar las acciones que tienen que emprenderse cuando ocurren problemas o se presentan circunstancias inesperadas. Puede incluirse un reporte sobre las excepciones o procedimientos para la corrección de problemas. (figura 3.1.3.d).

Funciones del Personal

Las responsabilidades de todas las personas que tienen que ver con el nuevo sistema, incluyendo los usuarios, operadores de computadora y personal de apoyo. Abarca todo el espectro de componentes del sistema, incluso de la entrada de datos hasta la distribución de salidas o resultados. (figura 3.1.3.e).

A menudo las funciones del personal se establecen en forma de procedimientos.

Estos elementos aparecen una y otra vez en muchas características de los sistemas de información. por consiguiente, todos estos elementos tienen una misma importancia al estructurar el diseño.

Diseño de la salida

El término *salida*, se refiere a los resultados e información generados por el sistema. Para muchos usuarios finales, la salida es la única razón para el desarrollo del sistema y la base sobre la que ellos evaluarán la utilidad de la aplicación. En realidad, muchos usuarios no operan el sistema de información y tampoco ingresan datos en él, pero utilizan la salida generada por el sistema.

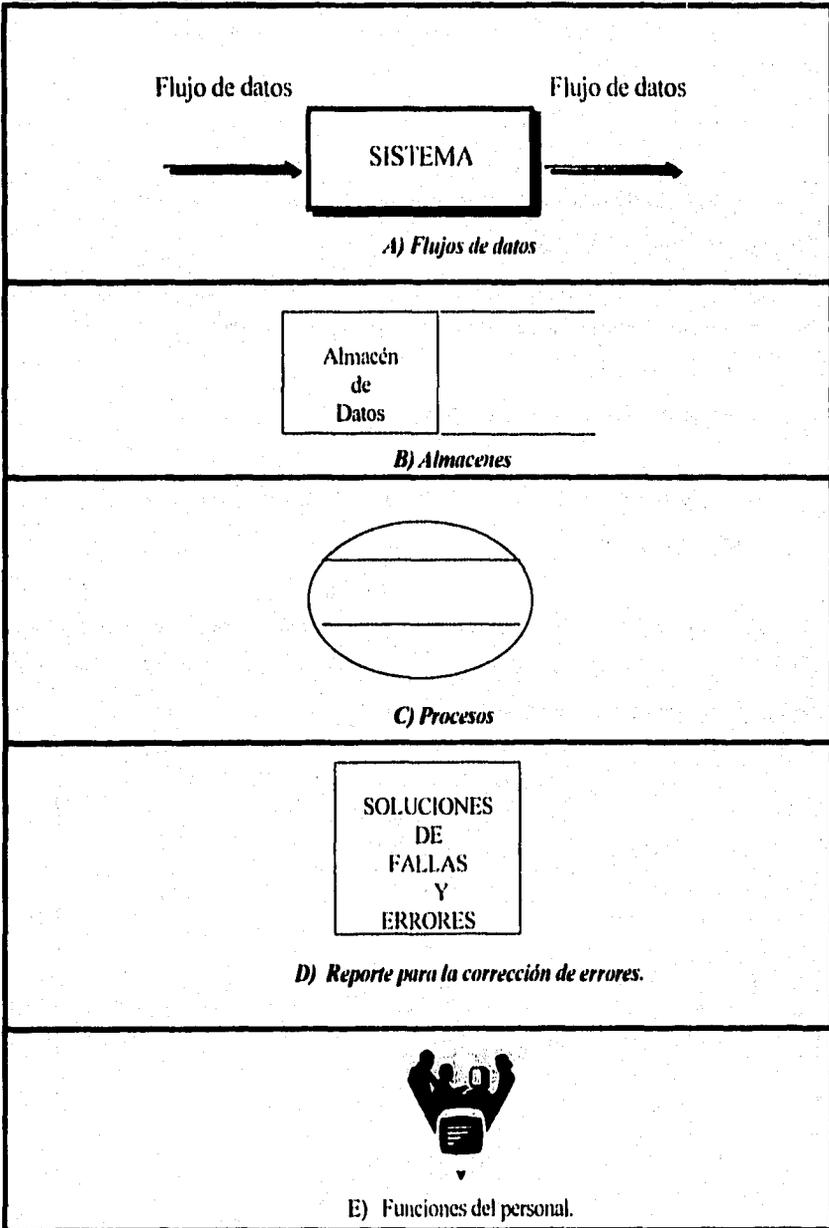


FIG. 3.1.3 Elementos del diseño.



Cuando diseñan la salida (figura 3.1.4), los analistas deben realizar lo siguiente:

- Determinar que información presentar.
- Decidir si la información será presentada en forma visual, verbal ó impresa y seleccionar el medio de salida.
- Disponer la presentación de la información en un formato aceptable.
- Decidir cómo distribuir la salida entre los posibles destinatarios.
- La disposición de la información sobre la pantalla o documento impreso se denomina distribución.

Para llevar a cabo las actividades antes mencionadas, se requieren decisiones específicas tales como el empleo de formatos ya sea impresos cuando se preparan reportes, cuantas líneas planear sobre una página impresa o si se deben emplear gráficas y colores.

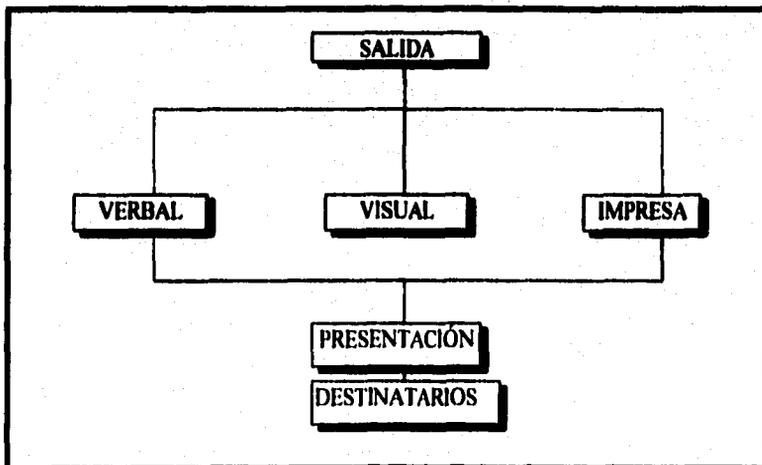


FIG. 3.1.4. *Diseño de la salida.*



Diseño de Archivos

El diseño de archivos incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera emplear para guardar detalles de las transacciones, datos de tipo histórico o información de referencia. Entre las decisiones que se toman durante el diseño de archivos (ver figura 3.1.5), se encuentran las siguientes:

- Los datos que deben incluirse en el formato de los registros contenidos en el archivo.
- La longitud de cada registro, con base en las características de los datos que contiene.
- La secuencia o disposición de los registros dentro, del archivo, la estructura de almacenamiento que puede ser secuencial, indexada o relativa.

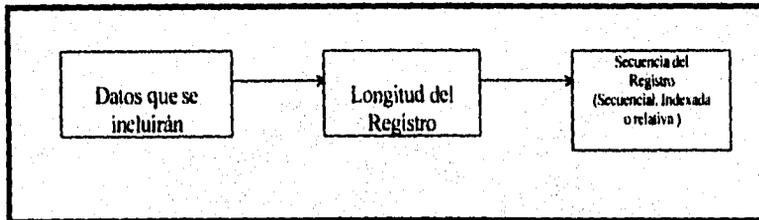


FIG.3.1.5 Diseño de Archivos

Diseño de Interacciones con la base de datos

Muchos sistemas de información, ya sean implantados en sistemas de cómputo grandes o pequeños, interactúan con las bases de datos que abarcan varias aplicaciones.

Dada la importancia que tienen las bases de datos en muchos sistemas, su diseño es establecido y vigilado por un administrador de bases de datos, que es una persona (o grupo de personas) que tienen la responsabilidad de desarrollar y mantener la base de datos. En estos casos el analista de sistemas no



efectúa el diseño de la base de datos sino que consulta al administrador de la base para determinar las interacciones mas apropiadas con la base de datos. El analista proporciona al administrador la descripción de:

- 1.- Los datos que son necesarios de la base de datos.
- 2.- Las acciones que tendrán efecto sobre la propia base (por ejemplo, la recuperación de datos, cambios en los valores de los datos o el ingreso de nuevos datos en la base).

A su vez, el papel del administrador de la base de datos incluye las siguientes responsabilidades (ver figura 3.1.6):

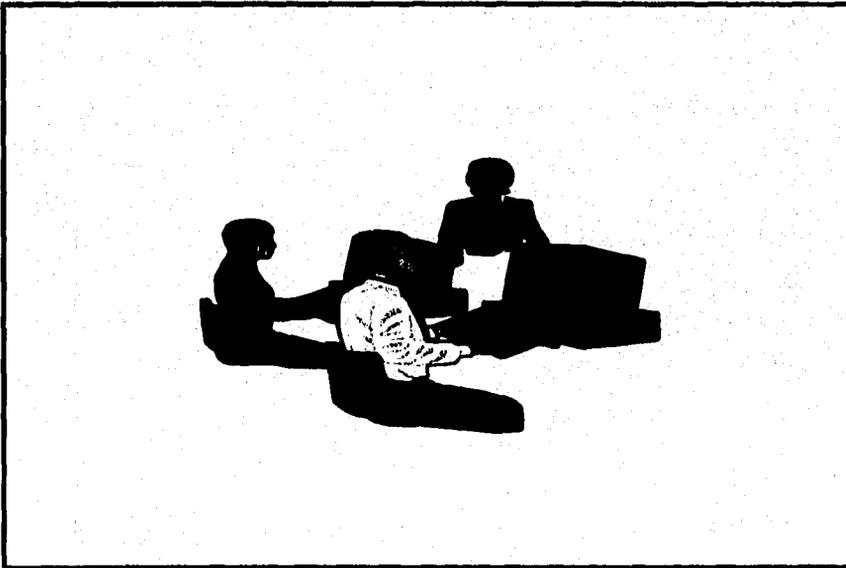


FIG. 3.1.6. Responsabilidades del administrador.



- Evaluar la conveniencia de la solicitud del analista.
- Describir los métodos para interactuar con la base de datos.

Diseño de la Entrada.

Los analistas de sistemas deciden los siguientes detalles del diseño de entradas (ver figura 3.1.7):

- 1.- Que datos ingresan al sistema.
- 2.- Que medios utilizar.
- 3.- La forma en que se deben disponer o codificar los datos.
- 4.- El dialogo que servirá de guía para dar entrada a los datos.
- 5.- Validación necesarias de datos y transacciones para detectar errores.
- 6.- Métodos para llevar a cabo y transacciones para detectar errores.

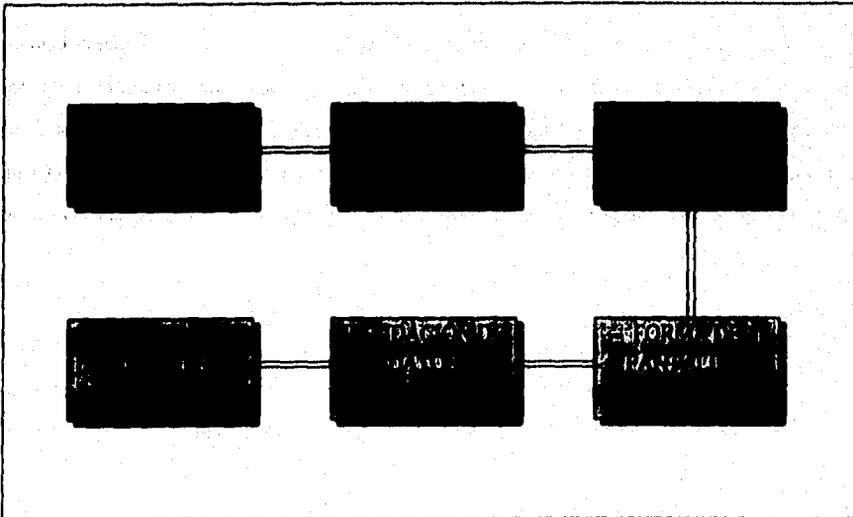


FIG. 3.1.7. Detalles del diseño de entrada.



Las decisiones para el manejo de entradas, especifican la forma en que serán aceptados los datos para su procesamiento por computadora. Los analistas deciden si los datos serán proporcionados directamente, quizás a través de una estación de trabajo, o por el uso de documentos, como talones de ventanas, cheques bancarios o facturas, donde los datos a su vez son transferidos hacia la computadora para su procesamiento.

El diseño de la entrada también incluye la especificación de los medios por lo que tanto los usuarios finales como los operadores darán instrucciones al sistema sobre las acciones que debe emprender.

Por ejemplo, un usuario que interactúa con el sistema por medio de una estación de trabajo, tiene que ser capaz de indicarle al sistema ya sea que acepte una entrada, genere un reporte o termine el procesamiento.

Diseño de controles .

Los analistas de sistemas también deben anticipar los errores que se cometerán al ingresar los datos en el sistema o al solicitar la ejecución de ciertas funciones. Algunos errores no tienen importancia ni consecuencias, pero otros pueden ser tan serios que ocasionarían el borrado de datos o el uso inapropiado del sistema. Aunque exista solo la más mínima probabilidad de cometer un error serio, un buen diseño de sistemas de información ofrecerá los medios para detectar y manejar el error.

Los controles de entrada proporcionan medios para:

- 1.- Asegurar que solo los usuarios autorizados tengan acceso al sistema.
- 2.- Garantizar que las transacciones sean aceptables.
- 3.- Validar los datos para comprobar su exactitud.



4.- Determinar si se han omitido datos que son necesarios.

Diseño de procedimientos.

Los procedimientos especifican que tareas deben efectuarse al utilizar el sistema y quienes son los responsables de llevarlas a cabo entre los procedimientos importantes se encuentran (ver figura 3.1.8):

Procedimientos para entrada de datos.

Métodos para la captura de datos de las transacciones y su ingreso en el sistema de información (ejemplo: secuencia para dar entrada a los datos registrados en los documentos fuentes)

Procedimientos durante la ejecución.

Pasos y acciones emprendidas por los operadores del sistema y en ciertos casos, por los usuarios finales que interactúan con el sistema para alcanzar los resultados deseados Ejemplo: montar paquetes en disco o colocar en las impresoras formas preimpresas.

Procedimiento para manejo de errores.

Acciones a seguir cuando se presentan resultados inesperados (ejemplo: ocurre un error cuando el sistema intenta leer datos de un archivo, la impresora se atasca durante la impresión de una gran cantidad de hojas, etc.).

Procedimiento de seguridad y respaldo.

Acciones para proteger al sistema y sus recursos contra posibles daños (ejemplo: cuándo y cómo hacer copias de los archivos maestros o de partes de la base de datos?).



Estos procedimientos deben formularse por escrito y formar parte de la documentación del sistema.

Diseño de especificaciones para programas.

Las especificaciones para programas son por sí mismas un diseño. Ellas describen como transformar las especificaciones del diseño del sistema - salidas, entradas, archivos, procesamientos y otras- en *software* de computadora.

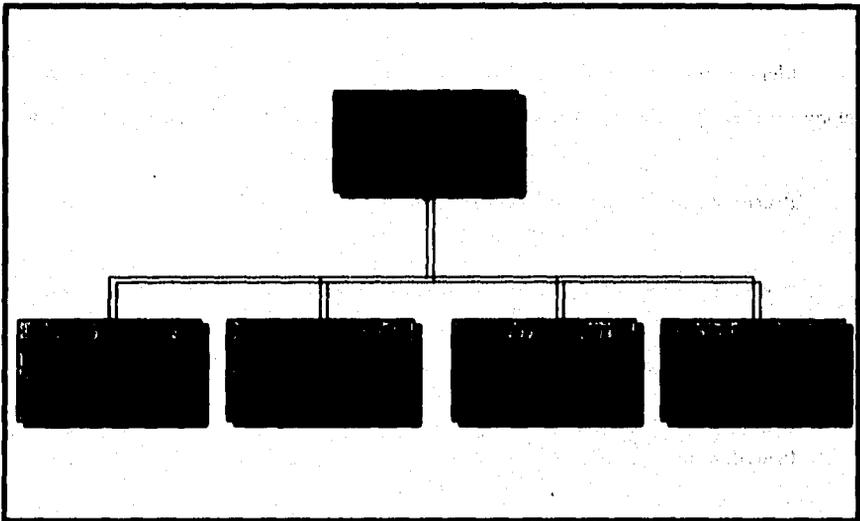


FIG. 3.1.8. Diseño de procedimientos.

El diseño del *software* de computadora es importante para asegurar que:

- Los programas producidos lleven a cabo todas las tareas y lo hagan en la forma establecida.
- La estructuración del *software* en módulos permita su prueba y validación para determinar si los procedimientos son correctos.



- Las modificaciones futuras se pueden realizar en forma eficiente y con un mínimo de interrupción en el diseño del sistema.

Carpeta de descripción del diseño del sistema.

Quando el diseño de un sistema de información está completo, las especificaciones son documentadas en una forma que esboza las características de la aplicación. Los analistas de sistemas denominan a estas especificaciones información liberada o carpeta de diseño. Ningún diseño esta completo sin la carpeta de diseño, ya que esta contiene todos los detalles que deben incluirse en el *software* de la computadora, conjunto de datos y procedimientos que comprenden el sistema de información.

La información liberada incluye los siguientes aspectos:

Cuadro de despliegue

Descripciones de las entradas y salidas donde se muestra la ubicación de todos los detalles que aparecerán en los reportes, documentos y pantallas de terminal.

Estructuras de registro

Descripción de todos los datos contenidos en los archivos maestros y de transacciones así como los diagramas relacionados con la base de datos.

Sistemas de codificación

Descripciones de los códigos que explican o identifican tipos de transacciones, clasificaciones y categorías de eventos o entidades.

Especificaciones de los programas

Cuadros, tablas y descripciones gráficas de los módulos y componentes del *software* de computadora junto con la interacción entre cada uno de ellos; también se incluyen las funciones realizadas y los datos utilizados o producidos por cada una de ellas.

Especificación de Procedimientos

Procedimientos planificados para instalar y operar el sistema cuando esté terminado.

Plan de desarrollo

Cronogramas que indican los tiempos necesarios para el desarrollo de las actividades: análisis de sistemas, programadores y personal necesario para el desarrollo; planes para pruebas preliminares e implantación.

Costo del paquete

Gastos anticipados para el desarrollo, implantación y operación del nuevo sistema, clasificados por categorías tales como personal, equipo, comunicaciones, facilidades y suministros; revisados y actualizados con base en las proyecciones de costo y beneficio de la propuesta original del proyecto.



RESUMEN.

Los objetivos del diseño de sistemas de información son proporcionar las especificaciones del bosquejo del sistema, esto es, las características del sistema que serán trasladadas en *software* para su uso por la organización. Estas especificaciones, denominadas **diseño lógico** del sistema incluyen detalles sobre las salidas, entradas, archivos, interacciones con bases de datos, controles y procedimientos. La **construcción física**, que es la etapa que le sigue al diseño lógico, produce el *software*, los archivos y un sistema que trabaje.

Otros objetivos de diseño incluyen proporcionar especificaciones detalladas para el *software* y asegurar que el sistema proporcione apoyo a las actividades de las empresas, cumpla con los requerimientos de los usuarios finales e incorpore la ingeniería necesaria para hacerlo **orientado hacia los usuarios**. Todas las características de diseño deben adherirse a los estándares establecidos por la organización para sus sistemas de información.

Tomando en cuenta que el sistema será realizado en un lenguaje de programación visual, en donde la interface con el usuarios será gráfica, entonces estamos hablando de que el sistema será manejado bajo ambiente Windows. Así, el usuario podrá obtener información mediante el uso de iconos y de ventanas.

Para elaborar el presente sistema nos basaremos en las características de la Programación visual y de Unix(C-estándar). Una vez definido lo anterior, el trabajo de diseño consistirá primeramente en:

- 1.- Hacer un análisis del problema planteando, definiendo las variables de entradas y de salidas de información.
- 2.- Identificar los módulos que el sistema necesita. Los módulos deben estar basados en las actividades centrales del sistema.



3.- Una vez que se identificaron los módulos , la siguiente tarea es determinar que responsabilidades presentan, a esto se le llama " asignación de atributos y comportamientos". Las responsabilidades de un módulo caen dentro de dos categorías:

- La información que un módulo debe contener.
- Las operaciones que un módulo pueda ejecutar o que pueden ser ejecutados (¿qué puede hacer este módulo?).

1.- Encontrar relaciones entre los módulos, para poder identificar tales relaciones se debe de considerar el comportamiento asignado al módulo.

2.- Recabar y capturar información correspondiente de la base de datos que el sistema manejará. La base de datos contendrá información de los usuarios que serán vaciadas de las solicitudes de clave.



3.1.1 CRITERIOS DE MONITOREO Y VERIFICACIÓN

La información es la parte más importante de una organización, ya que es la que le da el control y le permite tomar decisiones en función de resultados obtenidos de un proceso, por tal motivo, en los siguientes párrafos se procederá a estudiar la información concierne al proceso de monitoreo y verificación de los usuarios, pero antes de iniciar dicho estudio, se presenta la definición de información, para que ésta de la pauta de los puntos a tratar.

Información

Se llama información a los datos o hechos relacionados por una condición, que proporcionan un mensaje y al ser percibidas proporciona conocimiento.

Por consiguiente, para efectuar el estudio de la información, es decir de los datos que necesitamos, se procedió a analizarlos. La forma como se va a verificar a cada uno de los usuarios, esta basada en relación a cierto tipo de información, misma que se obtiene del usuario; primeramente se realizó una investigación detallada de los datos que más frecuentemente se requieren de los usuarios. Por lo que se procederá a realizar un sondeo sobre diferentes tipos de formas y documentos; tales como:

- Forma para Inscripción para los servicios de Correo Electrónico
- Forma para Inscripción a la Biblioteca
- Darse de alta a RedUNAM a través de la DGSCA
- Inscripción a Cursos de Computación
- Inscripción a una Escuela en General
- Inscripción a un Gimnasio
- Inscripción a un Club de Video
- Inscripción para obtener la Credencial de Elector



La información que se proporcione debe ser relacionada con los datos personales, escolares o laborales, mismos que son detallados, ya que en base a estos se genera información para la base de datos.

Para tener un control adecuado de la gran cantidad de información que se esta manejando es necesario ordenarla de acuerdo a una clasificación, que para fines prácticos del sistema se basa en Tipos de usuarios, es decir, se consideran dos tipos de usuarios, Internos (I) que a su vez se clasifican en Alumnos (A), Profesores(P) e Investigadores(In) y Externos (E) que se subdividen en: Instituciones Académicas Públicas (IPU), Instituciones Académicas Privadas (IPR), Instituciones Lucrativas (IL) e Instituciones no Lucrativas (INL). Con respecto a las Instituciones Académicas Públicas así como con las Privadas se tiene la clasificación de Alumnos, Profesores e Investigadores. (Ver figura 3.1.1.1)

Con respecto a la información que proporcione el usuario en las siguientes fichas se muestran en forma general los datos que se requieren para el sistema.

Información que se necesita del usuario interno alumno:

USUARIO INTERNO ALUMNO
NOMBRE
DIRECCIÓN
TELÉFONO
DATOS ESCOLARES

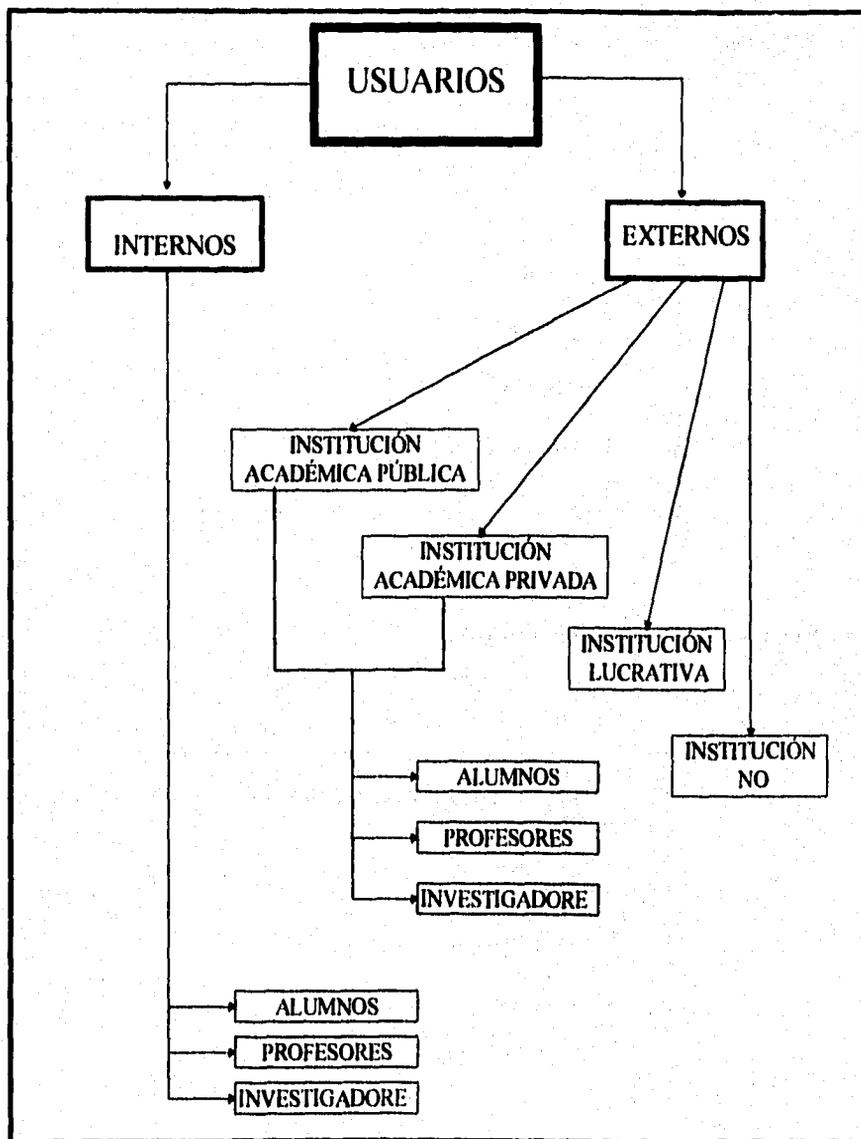


FIG. 3.1.1.1 *Clasificación de usuarios.*



Información requerida del Usuario Interno Profesor:

<p>USUARIO INTERNO PROFESOR</p> <p>NOMBRE</p> <p>DIRECCIÓN</p> <p>TELÉFONO</p> <p>DATOS LABORALES</p>
--

Datos requeridos del Usuario Interno Investigador

<p>USUARIO INTERNO INVESTIGADOR</p> <p>NOMBRE</p> <p>DIRECCIÓN</p> <p>TELÉFONO</p> <p>DATOS DEL PROYECTO</p>



Datos requeridos del usuario externo (alumno, Investigador, Profesor):

<p style="text-align: center;">USUARIO EXTERNO</p> <p>NOMBRE</p> <p>DIRECCIÓN</p> <p>TELÉFONO</p> <p>DATOS ESCOLARES/LABORALES</p>

De los datos que se necesitan de los usuario se obtuvieron , los formatos que deberá llenar el usuario al momento de inscribirse, estos formatos se encuentran en el **Apéndice E**

Monitoreo

Dentro de las funciones principales del sistema a desarrollar se encuentra la de monitorear al usuario. Esta es la parte encargada de leer qué máquinas se encuentran en sesión o activas en la red y se realiza a través de un conmutador digital que proporciona líneas para el servidor que ofrece una conexión a RedUNAM.

Para seleccionar la línea a monitorear se realizará un procedimiento aleatorio el cual determinará la misma. El siguiente paso será identificar al usuario a través de su ID, esto nos permitirá hacer referencia al mismo, en la base de datos que se tiene en el sistema y que contiene los datos del usuario en sesión.

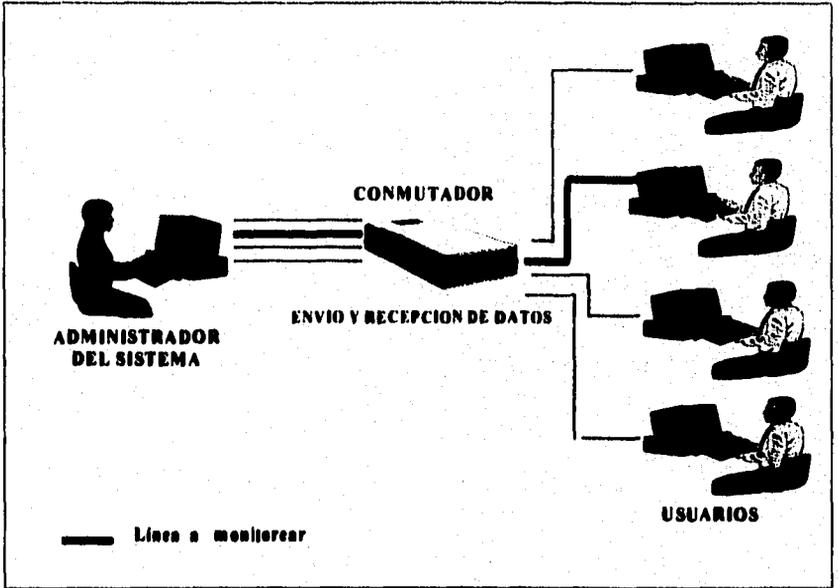


FIG. 3.1.1.2. Proceso de Monitoreo.



3.1.2 MANEJO DE DATOS

Primeramente es necesario dar un panorama general de todos los posibles usuarios que llegan a DGSCA para solicitar una clave de RedUNAM, para lo cual, como se mencionó anteriormente, se tienen en general dos tipos de usuarios: Internos y Externos.

Un usuario Interno es aquella persona que pueda comprobar que es miembro de la Universidad Nacional Autónoma de México y que para solicitar una cuenta de RedUNAM se identifique como Alumno, Académico o Investigador de la misma.

Un usuario Externo es aquella persona que no forma parte de la Universidad Nacional Autónoma de México, pero que sí puede estar integrado a otra Institución Académica ya sea Pública o Privada siendo Alumno, Académico o Investigador o bien formar parte de alguna Institución Lucrativa o No Lucrativa.

La consideración de manejar únicamente dos tipos globales de usuarios Figura 3.1.1.3, en general, nos permite delimitar con claridad cuales serán las tarifas a considerar como pago por el uso de la clave de RedUNAM, además de facilitar el manejo de los datos a través del login.

Después de esta consideración debemos de establecer el formato para el manejo de datos en nuestra aplicación, es decir, como se van a validar los datos de los usuarios en el sistema.

Consideraremos dos tipos de validación: con máscaras en Access, y por comparación y máscaras a través de Visual Basic.

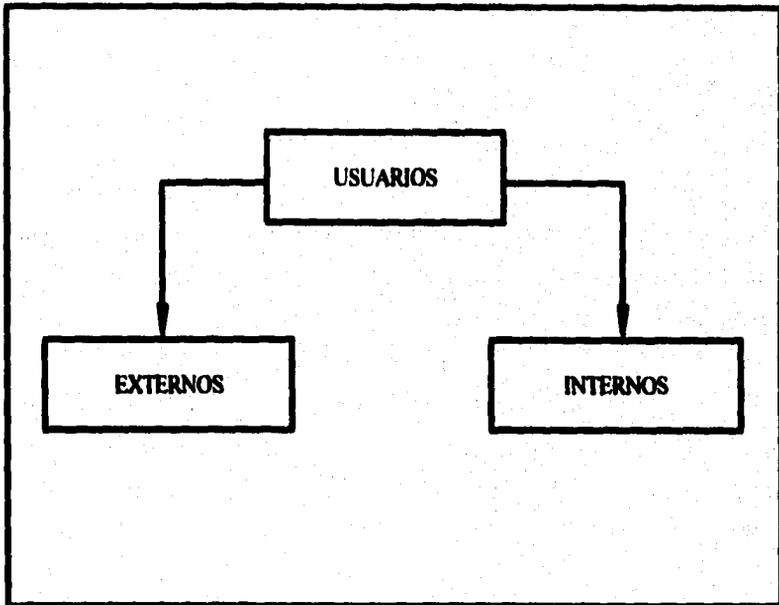


FIG. 3.1.1.3. Clasificación Básica de Usuarios.

Validación por máscaras.

Este tipo de validación se realiza tanto en Access como en Visual Basic (en fase de diseño y por medio de programación) para manejar los datos contenidos en la Base de Datos, así como los que son introducidos por los usuarios. La máscara es un formato que se especifica para ayudar a introducir datos en un control o un campo. Una máscara de entrada consta de caracteres literales de presentación (como paréntesis, puntos o guiones) y caracteres de máscara que especifican donde deben introducirse los datos, que tipos de datos están permitidos y cuántos caracteres se permiten. Los caracteres literales de presentación de una máscara de entrada pueden almacenarse opcionalmente como datos..



Validación por comparación

Al igual que la validación por máscaras, este tipo de validación también es un formato que se especifica para ayudar a introducir datos en un control o un campo, la única diferencia es que este tipo de validación requiere programación, que en nuestro caso se realizó en Visual Basic. Los procedimientos de validación que se realizaron son utilizados en los módulos donde se lleva a cabo la captura de la información para que sólo considere como información válida aquella que cumpla con las especificaciones de cada campo. En este tipo de validación se especificaron entre otras cosas que los datos de entrada cumplan con características de longitud, de combinación de máscaras, etc.

La siguiente tabla especifica la máscara de cada campo en la Base de Datos.(ver figura 3.1.1.4).

CAMPO	MASCARA
ID_US	LLLL0000
TIPO_US	L
TIPO_CLAVE	L
ERS_CONS	###
APE_MAT_US	????????????????
APE_PAT_US	????????????????
NOM_US	????????????????
CALLE_US	
NUM_US	
COL_US	????????????????
DEL_US	????????????????
CP_US	0000
TEL_US	(###)000-0000
RFC	LLLL00000
NAC_US	00/00/00
LUG_NAC_US	????????????????



INS_US	????????????????
CALLE_INS	
NUM_INS	
COL_INS	????????????????????????????
DEL_INS	????????????????????????????
CP_INS	00000
TEL_INS	(##)000-0000
NUM_CTA	000000-0
AÑO_ING	00/0000
SEM_CURSA	
CEP_PROF	000000
PROM_ACT	#0.0
FOTO	
0 = Sólo número L = Letra (a-z)	# = Número, espacio o + o - ? letra (a-z) ¹

FIG. 3.1.1.4 Validación de Datos

¹ Sólo guarda el número de caracteres tecleados y no toma en cuenta los nulos



3.1.3 GENERACIÓN ALEATORIA DE CLAVES DE PRUEBA

Para la generación aleatoria de claves de pruebas se utilizaron dos funciones de Visual Basic las cuales son:

RANDOMIZE: Se encarga de activar el generador de números aleatorios a partir de un número determinado.

RND: Da como resultado un número al azar entre un rango predefinido a partir de un límite inferior y un límite superior conocidos.

En la aplicación se utilizaron ambas funciones (Randomize y Rnd) que generaron números pseudoaleatorios enteros dentro de un rango definido para realizar esta función se utilizó la siguiente función:

$Int((lim_superior - lim_inferior + 1) * Rnd + lim_inferior)$

donde

lim_superior: es el número más alto dentro del rango determinado.

lim_inferior: es el número más chico dentro del rango determinado.

Cabe hacer mención que Rnd da como resultado un **double** comprendido entre 0 y 1, es decir, un número real de doble precisión, y por este motivo es que se utiliza la función **int** para que nos genere entero.

Se determinó utilizar este método de generación de números aleatorios porque era muy apropiado para nuestra aplicación ya que se tuvo que generar números pseudoaleatorios enteros dentro del rango [1..17 para seleccionar las preguntas a enviarle al usuario y un rango [1..15 para seleccionar la línea del conmutador.



Es importante mencionar que el randomize toma la hora del reloj del sistema y esto permite que la secuencia generada de números aleatorios no se repita, de tal modo que nos aseguramos de no enviar las mismas preguntas y no seleccionar la misma línea a monitorear.

El conmutador digital detecta qué líneas están activas en ese momento, se tomará aleatoriamente por medio del sistema sólo una de todas las líneas ocupadas. Una vez que se tenga la línea seleccionada se procede a identificar qué usuario está en ese momento haciendo uso de la red, dicha operación se logra a través del login respectivo de cada usuario, posteriormente se selecciona el número de preguntas a enviar (de una a tres); también se seleccionan las preguntas a realizar, mediante el proceso antes descrito ya que existe un archivo que contiene todas las preguntas posibles a realizarle al usuario. Por último se enviará un cuestionario con las preguntas previamente seleccionadas, para que el usuario las responda y automáticamente retomen para su pronta validación.

Transcurrido un intervalo de tiempo se repite la operación para monitorear otra línea, esto si se encuentra activado el monitoreo automático de lo contrario se interrumpe este proceso para realizar de manera manual la generación de la selección de la clave a probar.



3.2 DISEÑO Y CONSTRUCCIÓN DE LA BASE DE DATOS.

Un sistema de Base 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 Base de Datos son: el *hardware*, el *software* DBMS y los datos por manejar.

Una Base de Datos es un conjunto de información que tiene relación entre sí y es lo suficientemente amplia para satisfacer las necesidades de los usuarios de dicha información.

Las Bases de Datos son creadas con un DBMS. El contenido de una Base de Datos 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, la figura. 3.2.1 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 de Datos ya que los archivos ahí almacenados están conectados directa o indirectamente.

El proceso de diseño involucra los tres procesos siguientes:

- 1) Agrupamiento de los datos globales de una Base de Datos en uno o más registros de tipo conceptual (Entidades).
- 2) Definición de las relaciones conectando entidades relacionadas en un modelo conceptual de Base de Datos.
- 3) Transformación del modelo conceptual de la Base de Datos en un esquema mejorado que se puede describir mediante un DBMS específico para la implantación de la Base de Datos.

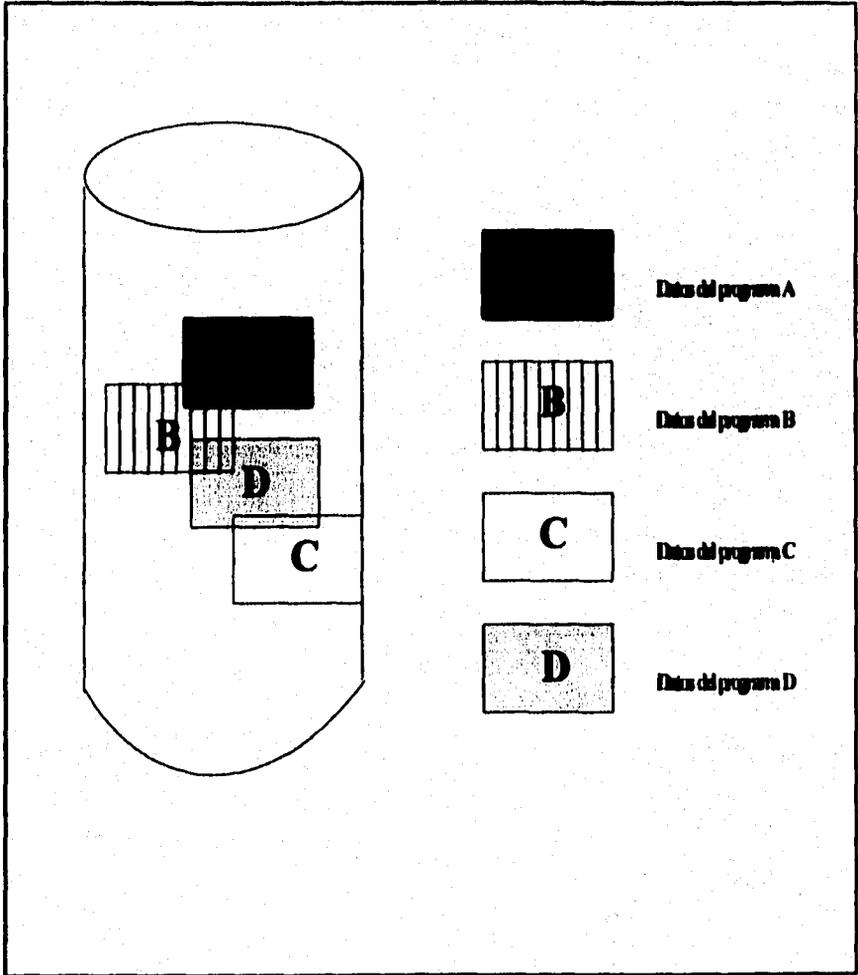


FIG.3.2.1 Base de Datos



3.2.1 DIAGRAMA DE BLOQUES PARA DESARROLLO DE LOS MÓDULOS PRINCIPALES.

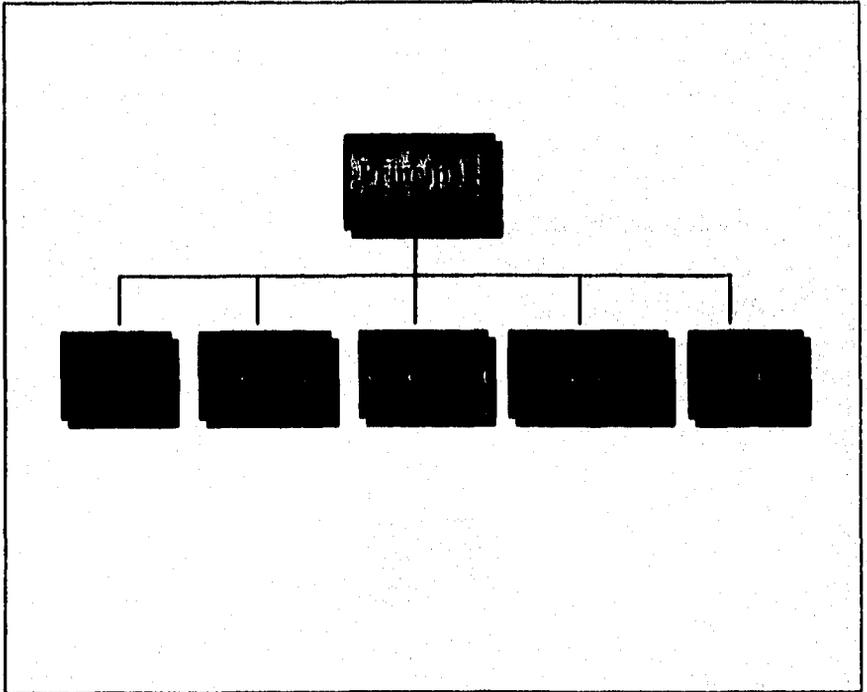


FIG. 3.2.1.1 Módulo Principal

El diagrama que se muestra en la fig. 3.2.1.1 contiene todos los módulos disponibles del sistema. los cuáles son:

Captura

Alias

Bajas

Cambios

Consultas

Monitoreo

Automático

Manual

Estadísticas

Bitacora

Ayuda

Ayuda en línea (ver manual anexo).

CAPTURA.(Ver figura 3.2.1.2)

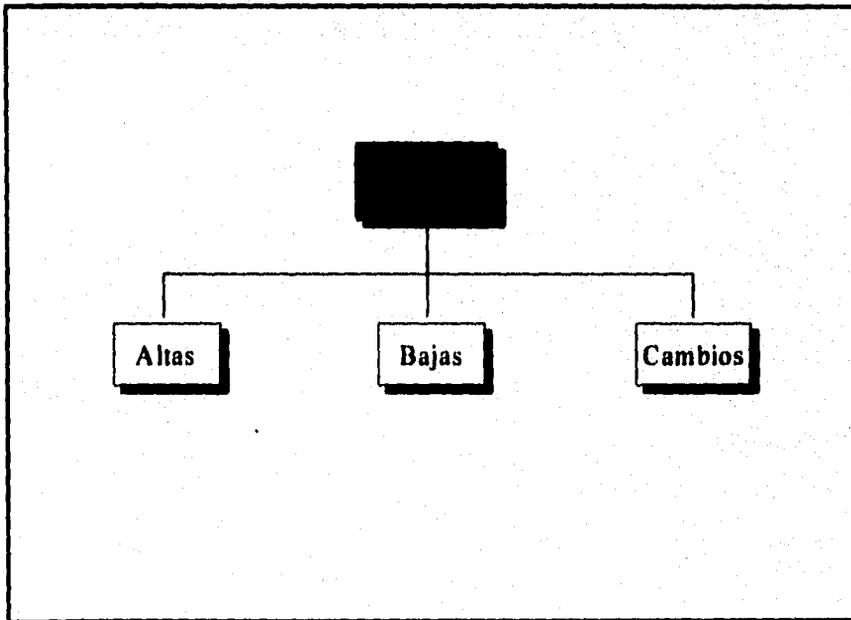


FIG. 3.2.1.2 Módulo de Captura



Este módulo es el encargado de registrar toda la información del usuario de cada clave en base a campos los cuales serán los siguientes:

Nombre:

Apellido paterno

Apellido materno

Nombre de Pila

Domicilio particular:

calle

número

colonia

C.P.

teléfono

Fax

Dependencia, Facultad o Institución:

nombre

teléfono

Dirección

Función o cargo

Tipo de clave

Fecha de nacimiento

Lugar de nacimiento

R.F.C.

Número de materias

Número de cuenta

Año de ingreso (Dependencia, Facultad)

Folio de la Cédula Profesional

Promedio Actual

Estos campos son utilizados por los 3 submódulos que comprenden el módulo de captura, que se describen a continuación.

Altas.

Aquí se crea el registro donde serán almacenados los datos en los diversos campos de información que conforman la Base de Datos. Este módulo es la base del sistema ya que de él depende la correcta validación de la información del usuario de la clave monitoreada en un momento dado; de lo contrario, si los datos son capturados erróneamente proporcionará una incorrecta validación de la clave.

Bajas.

En este submódulo se realiza un proceso de depuración manual en la base de datos. El proceso de depuración se lleva a cabo periódicamente bajo las siguientes circunstancias:

- 1.- Falta de renovación.
- 2.- Término de vigencia.
- 3.- Uso inadecuado de la clave.
- 4.- Incumplimiento de políticas.
- 5.- Baja voluntaria.

Cambios.

Existe un submódulo de cambios como una herramienta al administrador que le permite manejar los registros de la base de datos de la manera que lo crea más conveniente, es necesario, tener precauciones para sólo modificar los registros y campos indicados para su correcta validación. Desde el punto de vista de la información el submódulo de cambios es una tabla virtual de referencia que guarda las correcciones aplicables a todos y cada uno de los registros que conforman la base de datos.



La información que interviene en éste nivel es la que se encuentra en la Base de Datos, y que en forma estricta es de uso confidencial, para lo cual solo los administradores indicados tienen acceso a este módulo, que será controlado por medio de una fecha de modificación asignada a la bitácora de cada administrador.

CONSULTAS

Las consultas permiten al encargado del sistema visualizar agrupando e identificando a un conjunto de claves con características similares, es decir, es una relación de uno a muchos con el módulo de altas, porque cada consulta puede estar asociada a "n" claves. Sin embargo, también se cuenta con la posibilidad de consultar en forma individual, lo que conlleva a tener una relación uno a uno, siendo flexibles en éste aspecto. Por otro lado es importante notar, que al ser únicamente una vista de los elementos de cada registro no se requiere de cálculo alguno, por lo que no existen procedimientos que calculan resultados, únicamente existen rutinas para la selección de registros de características particulares o en su caso por selección de conjuntos.

MONITOREO. (Ver figura. 3.2.1.3)

La parte medular de nuestro sistema consiste en el correcto monitoreo de las claves de Internet. Este proceso consiste en realizar la validación adecuada de los usuarios, mediante los siguientes pasos:

- 1.- Seleccionar clave.
- 2.- Identificar línea telefónica.
- 3.- Seleccionar línea.
- 4.- Enviar cuestionario
- 5.- Recibir respuesta.
- 6.- Comparar respuestas con la base de datos.
- 7.- De no ser correctas, terminar sesión.
- 8.- De ser correctas, continuar sesión.



Se presenta en dos submódulos los cuales son: Automático y Manual; quedando por omisión el automático.

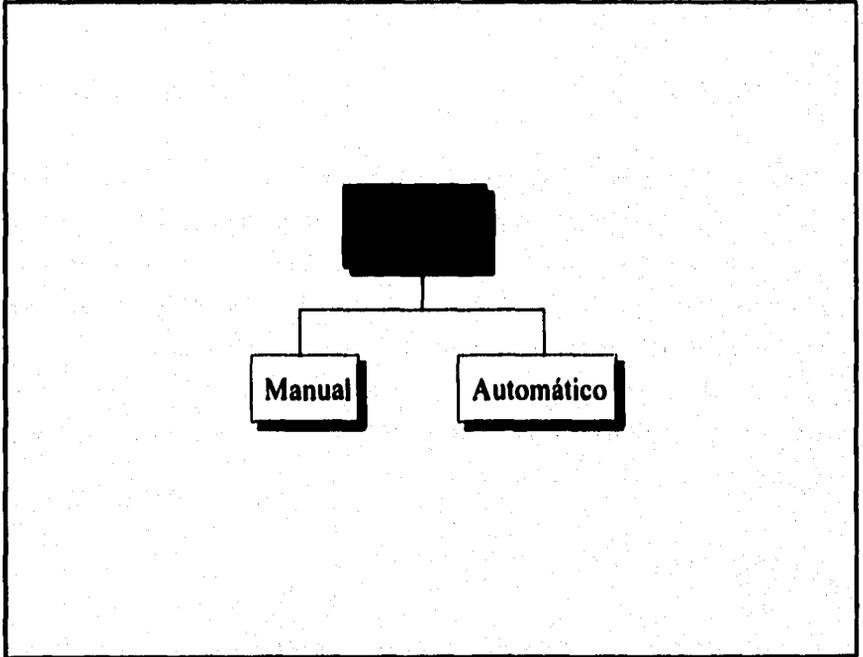


FIG.3.2.1.3. Módulo de Monitoreo

Automático.

Se encarga de realizar el proceso de monitoreo seleccionando aleatoriamente la clave a rastrear, generando los resultados que a cada caso corresponden. Este monitoreo se realiza periódicamente, el administrador del sistema es el encargado de definir el periodo de monitoreo.

Manual



Este submódulo inhabilita el monitoreo automático, de tal forma que el administrador especifica la clave a monitorear y realiza el proceso de monitoreo paso a paso.

ESTADÍSTICAS.

Al invocar éste módulo nos presenta de forma gráfica diversas funciones desarrolladas por el sistema como son:

- ◆ Porcentaje de Tipos de Usuarios,
- ◆ Control de la bitácora de los administradores,
- ◆ Porcentaje de tipos de claves,
- ◆ Porcentaje de uso de claves,
- ◆ Porcentaje de desconexión,

El análisis lo realiza con diferentes tipos de gráficos, los cuales pueden ser cambiados a gusto del usuario. Estos tipos de gráficos van desde mostrar el análisis en un gráfico sencillo, de barras, de líneas y de sectores, hasta mostrarlo en gráfica de tercera dimensión de líneas, barras y sectores, si se selecciona esta última, se tiene la opción de separar o unir las diferentes divisiones.

Además en este módulo se cuenta con la opción de imprimir el gráfico, el que imprime es el que tiene en ese instante desplegado en la pantalla al momento de realizar un clic en el botón de "IMPRIMIR".



AYUDA.

Este módulo nos proporciona información sobre el manejo y uso adecuado del sistema, apoyándose en ligas de hipertexto con el fin de facilitar y asesorar a los usuarios (administradores) que así lo requieran.

El módulo de ayuda cuenta con cuatro submódulos los cuales se listan a continuación:

En línea.

La pantalla en línea aparece cuando el usuario pulsa F1 o selecciona la opción contenido de menú de ayuda e informa al usuario del contenido de ayuda con que cuenta el sistema o del control que tiene la atención del sistema (enfoque).

Desde esta pantalla, el usuario elige un tema y recibe ayuda directamente.

Buscar acerca de.

Despliega una forma de ayuda sobre los diversos tópicos que se programaron previamente, presenta una caja de texto para que el usuario introduzca el tópico que sea de su interés y el sistema regresa diversos temas relacionados con el tópico

A cerca de.

Al activar este submodulo nos despliega en la pantalla un ventana, la cual contiene información de los derechos de autor, versión, identificador y datos de donde conseguir este sofisticado sistema.

Índice.

Aquí se muestra una lista en orden alfabético de todos los tópicos que el sistema requiera para asegurar su buen funcionamiento y brindar asesoría al usuario al estar navegando en el sistema. Es decir el usuario puede buscar un tema específico utilizando palabras clave a las que se esta haciendo referencia y existe una segunda lista del tema seleccionado para realizar una consulta del tema más detallada.

Además realiza una breve introducción de lo que hace el sistema, de su manejo, de la instalación.



3.2.2 DIAGRAMA DE FLUJO DE DATOS

En las figuras 3.2.2.1 y 3.2.2.2 se muestra el esquema general del funcionamiento del sistema.

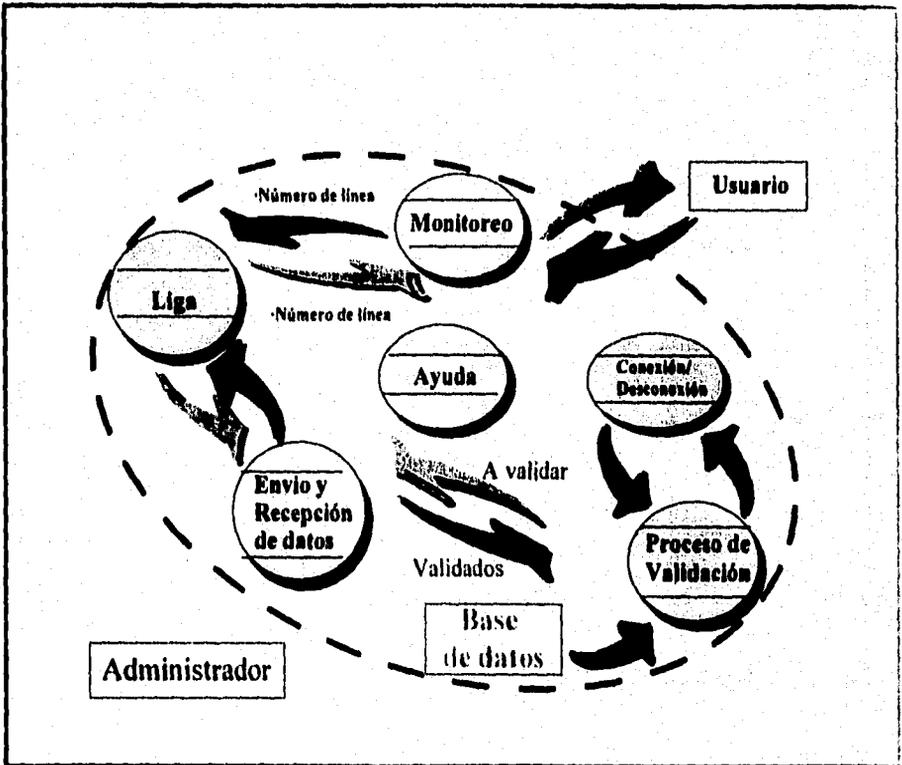


FIG. 3.2.2.1 Diagrama de Flujo del Sistema

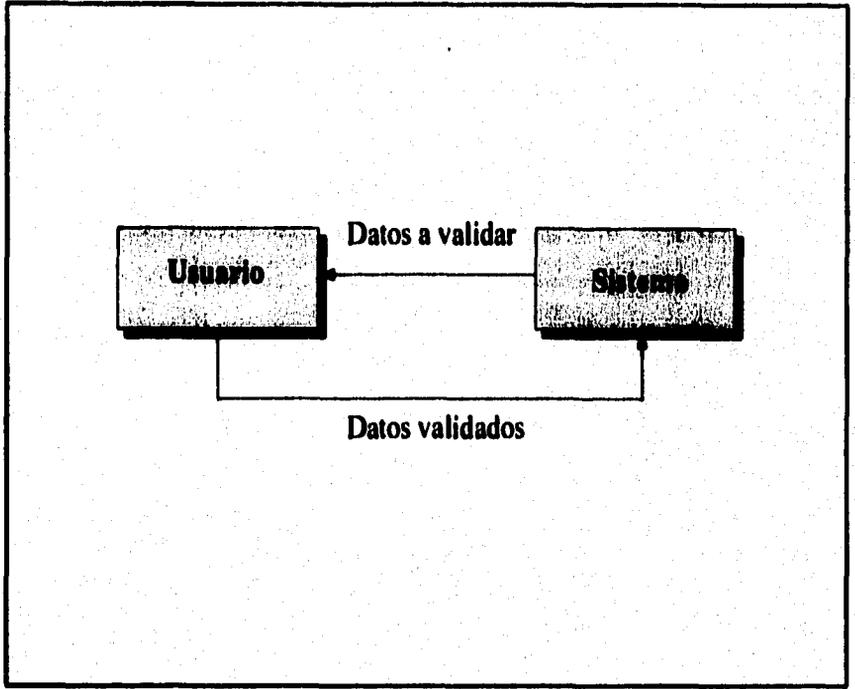


FIG. 3.2.2.2 Diagrama de Flujo del Datos



3.2.3 DICCIONARIO DE DATOS.

El Diccionario de Datos es un listado organizado de todos los datos pertinentes al Sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas y componentes de almacenamiento.

El empleo del Diccionario de Datos:

Reduce ambigüedades

Especifica detalles.

La siguiente figura muestra las tablas con sus campos, (así como sus características) que conforman la Base de Datos del sistema.

TABLA DE USUARIO

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
ANO_ING	Númerico	Entero largo	No	Año de Ingreso a la Facultad
APE_MAT_US	Char	20	No	Apellido Materno
APE_PAT_US	Char	20	No	Apellido Paterno
CALLE_US	Char	40	No	Calle del Domicilio
CED_PROF	Númerico	Entero Largo	No	Cédula Profesional
COL_US	Char	30	No	Colonia
CP_US	Númerico	Entero Largo	No	Código Postal
FOTO	Imagen	.	No	Foto del Usuario
HRS_CONS	Númerico	Double	No	Tiempo Usado
ID_CLAVE	Númerico	Entero Largo	No	Tipo de Clave
ID_DEL	Númerico	Entero largo	No	Identificador de la Delegación
ID_INST	Númerico	Entero Largo	No	Identificador de Institución



ID_LUGAR	Númérico	Entero Largo	No	Lugar de Nacimiento
ID_TIPO	Texto	1	No	Identificador del tipo de Usuario
ID_US	Contador	8	Si	Clave del Usuario
NAC_US	Date	Fecha Corta	No	Fecha de Nacimiento
NOM_US	Char	30	No	Nombre
NUM_CTA	Char	9	No	Número de Cuenta
NUM_MAT	Númérico	Entero Largo	No	Número de materias que cursa
NUM_US	Char	10	No	Número
PROM_ACT	Númérico	Doble	No	Promedio Actual
RFC	Char	10	No	Registro Federal de Causantes
SEM_CURSA	Char	10	No	Semestre que Cursa
TEL_US	Char	10	No	Teléfono

TABLA BITÁCORA.

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
FECHA1	Date	Fecha Corta	No	Fecha del primer fallo en validación
FECHA2	Date	Fecha Corta	No	Fecha del 2do fallo en validación
FECHA3	Date	Fecha Corta	No	Fecha del 3er fallo en validación
HORA1	Date	Hora Corta	No	Hora del 1er fallo en validación
HORA2	Date	Hora Corta	No	Hora del 2o fallo en validación



HORA3	Date	Hora Corta	No	Hora del 3er fallo en validación
ID_US	Númérico	Entero Largo	Si	Identificador del usuario

TABLA CLAVE

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
HORAS_CLAVE	Númérico	Double	No	Horas asignadas
ID_CVE	Contador	.	Si	Identificador de la clave

TABLA DELEGACIÓN

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
ID_DEL	Contador	.	Si	Identificador de la Delegación
NOMBRE_DEL	Char	50	No	Nombre de la Delegación

TABLA INSTITUCIÓN

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
CALLE_INS	Char	20	No	Calle de la Institución
COL_INS	Char	30	No	Colonia de la Institución
CP_INS	Númérico	Entero largo	No	Código Postal de la Institución
ID_DEL	Númérico	Entero Largo	No	Identificador de la Delegación



ID_INST	Contador	-	Si	Identificador de la Institución
NOM_INS	Char	35	No	Nombre de la Institución
NUM_INS	Char	10	No	Número de la Institución
TEL_INS	Char	15	No	Teléfono de la Institución

TABLA LUGAR DE NACIMIENTO

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
ID_LUGAR	Contador	-	Si	Identificador del Lugar
LUG_DE_NAC	Char	50	No	Lugar de Nacimiento del Usuario

TABLA TIPO-USUARIO

CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCIÓN
DESCRIPCIÓN	Char	30	No	Descripción del tipo de usuario
ID_TIPO	Char	1	Si	Identificador del tipo de Usuario



3.2.4. NORMALIZACIÓN DE TABLAS.

En la normalización se elimina en lo posible todos los datos que tengan anomalías, debe de conservarse toda la información y además evitar la redundancia.

En la siguiente tabla (tabla 3.2.4.1) se muestran los datos que son capturados de la solicitud de cada usuario que desea adquirir una cuenta en RedUNAM e información propia del sistema.

LISTA DE CAMPOS	
CAMPO	DESCRIPCIÓN
APE_MAT_US	Apellido Materno
APE_PAT_US	Apellido Paterno
NOM_US	Nombre
CALLE_US	Calle del Domicilio
NUM_US	Número
COL_US	Colonia
DEL_US	Delegación
CP_US	Código Postal
TEL_US	Teléfono
RFC	Registro Federal de Causantes
NAC_US	Fecha de Nacimiento
LUG_NAC_US	Lugar de Nacimiento
TIPO_USUARIO	Tipo de Usuario
TIPO_CLAVE	Tipo de Clave
HRS_CONS	Tiempo Usado
INS_US	Institución donde trabaja
CALLE_INS	Calle de la Institución
NUM_INS	Número de la Institución
COL_INS	Colonia de la Institución



DEL_INS	Delegación de la Institución
CP_INS	Código Postal de la Institución
TEL_INS	Teléfono de la Institución
NUM_CTA	Número de Cuenta
NUM_MAT	Número de materias que cursa
AÑO_ING	Año de Ingreso a la Facultad
SEM_CURSO	Semestre que Cursa
CED_PROF	Cédula Profesional
PROM_ACT	Promedio Actual
FOTO	Foto del Usuario
FECHA1	Fecha del primer intento fallido
HORA1	Hora del primer intento fallido
FECHA2	Fecha del segundo intento fallido
HORA2	Hora del segundo intento fallido
FECHA3	Fecha del tercer intento fallido
HORA3	Hora del tercer intento fallido

Figura 3.2.4.1 Tabla USUARIO

De la tabla anterior se observa que no se encuentra normalizada, ya que no cumple con la primera forma normal (1FN), la cual en resumen establece lo siguiente :

Si para cada valor específico de un identificador existe uno y solo un valor de cada atributo. Es decir no hay grupos repetitivos. Pero se mantiene un alto grado de redundancia.

En el tema 1.4 se explicó detalladamente la normalización de las tablas, basándonos en la teoría descrita y en lo anterior, se tiene que de la Tabla 3.2.4.1 se pueden identificar 6 grupos repetitivos, a saber, el grupo que corresponde a los datos del tipo de institución al cual pertenece el usuario, otro grupo el cual identifica la bitácora del mismo, es decir, en donde se almacenan los intentos fallidos que este



usuario pueda tener, un tercer grupo que consiste en el lugar de nacimiento del usuario, otro el tipo de clave que tiene y por último el tipo de usuario que es.

Hecho lo anterior, se derivan las siguientes tablas:

Tabla INSTITUCION	
Campo	Descripción
NOM_INST	Institución donde trabaja
CALLE_INS	Calle de la Institución
NUM_INS	Número de la Institución
COL_INS	Colonia de la Institución
CP_INS	Código Postal de la Institución
TEL_INS	Teléfono de la Institución
DEL_INST	Delegación de la Institución

Figura 3.2.4.2 Tabla INSTITUCIÓN

Tabla TIPO_USUARIO	
Campo	Descripción
ID_TIPO	Identificador del tipo de usuario
DESCRIPCION	Describe el tipo de usuario

Figura 3.2.4.3 Tabla TIPO_USUARIO

Tabla LUGAR DE NACIMIENTO	
Campo	Descripción
ID_LUGAR	Identificador del lugar de nacimiento
LUG_DE_NAC	Describe el lugar de nacimiento

Figura 3.2.4.4 Tabla LUGAR DE NACIMIENTO



Tabla CLAVE	
Campo	Descripción
HORAS_CLAVE	Especifica el No. de horas disponibles de uso

Figura 3.2.4.5 Tabla CLAVE

Tabla BITÁCORA	
Campo	Descripción
FECHA1	Fecha del primer intento fallido
HORA1	Hora del primer intento fallido
FECHA2	Fecha del segundo intento fallido
HORA2	Hora del segundo intento fallido
FECHA3	Fecha del tercer intento fallido
HORA3	Hora del tercer intento fallido

Figura 3.2.4.6 Tabla BITÁCORA

Podemos observar que existe una entidad que esta relacionada tanto con la entidad USUARIO como con la de INSTITUCIÓN, ésta es la entidad DELEGACIÓN y esta representada en la siguiente tabla



TABLA DELEGACION	
CAMPO	DESCRIPCION
NOMBRE_DEL	Nombre de la Delegación

Figura 3.2.4.7 Tabla DELEGACION

En cada una de las nuevas tablas se observa que se ha agregado un nuevo campo el cual identifica de forma única a cada uno de los nuevos registros, ésta es la llave primaria. Para identificarlas, se muestra en tono gris.

Para no perder las Relaciones con la Tabla de USUARIO, estos subgrupos se relacionan con ella a través de las llaves primarias de cada subgrupo. Dichas llaves en la tabla de USUARIO se denominan llaves foráneas. Entonces, la tabla de USUARIOS se convierte en la siguiente tabla, después de la normalización

TABLA USUARIO	
CAMPO	DESCRIPCION
NOM_US	Nombre
APE_PAT_US	Apellido Paterno
APE_MAT_US	Apellido Materno
FOTO	Foto del Usuario
CALLE_US	Calle del Domicilio
NUM_US	Número
COL_US	Colonia
CP_US	Código Postal
ID_DEL	Identificador de la Delegación
TEL_US	Teléfono
RFC	Registro Federal de Causantes
NAC_US	Fecha de Nacimiento



ID_LUGAR	Identificador de Lugar de Nacimiento
ID_CVE	Identificador de Clave
ID_INST	Identificador de Institución
NUM_CTA	Número de Cuenta
SEM_CURSO	Semestre que Cursa
CED_PROF	Cédula Profesional
PROM_ACT	Promedio Actual
NUM_MAT	Número de materias que cursa
AÑO_ING	Año de Ingreso a la Facultad
HRS_CONS	No. de horas consumidas

Figura 3.2.4.8 Tabla USUARIO

Por lo que sabemos una *relación* esta en 2FN si esta en 1FN y además que el valor de cualquier atributo que no es llave depende de todos los atributos que forman la llave.

La 3FN o forma Normal de Boyce/Codd (FNBC) nos dice: *una relación esta en FNBC si y solo si cada determinante² es una llave candidata³*; en las tablas anteriores podemos darnos cuenta que cumple con la premisa de FNBC ya que todos los atributos dependen funcionalmente del ID_US. Es decir, que nuestra relación cumple con la segunda y tercera forma normal. Un caso similar ocurre con las tablas de: DELEGACIÓN en la que los atributos dependen funcionalmente del ID_DEL, para INSTITUCIÓN sus atributos dependen funcionalmente del ID_INST, en la tabla TIPO_USUARIO sus atributos dependen funcionalmente del ID_TIPO, en LUGAR DE NACIMIENTO sus atributos dependen funcionalmente del ID_LUGAR y finalmente en la tabla CLAVE los atributos dependen funcionalmente del ID_CVE.

²Se acuerda llamar *determinante* (funcional) a un atributo tal vez compuesto del cual depende funcionalmente en forma completa algún otro atributo.

³*Llave candidata* es cualquier atributo que pueda convertirse en llave primaria.

3.2.5 DIAGRAMA DE ENTIDAD RELACION

El diagrama que se muestra en la figura 3.2.5.1 muestra que la entidad USUARIO esta formada por 26 valores atómicos (atributos) correspondientes a los datos personales del usuario.

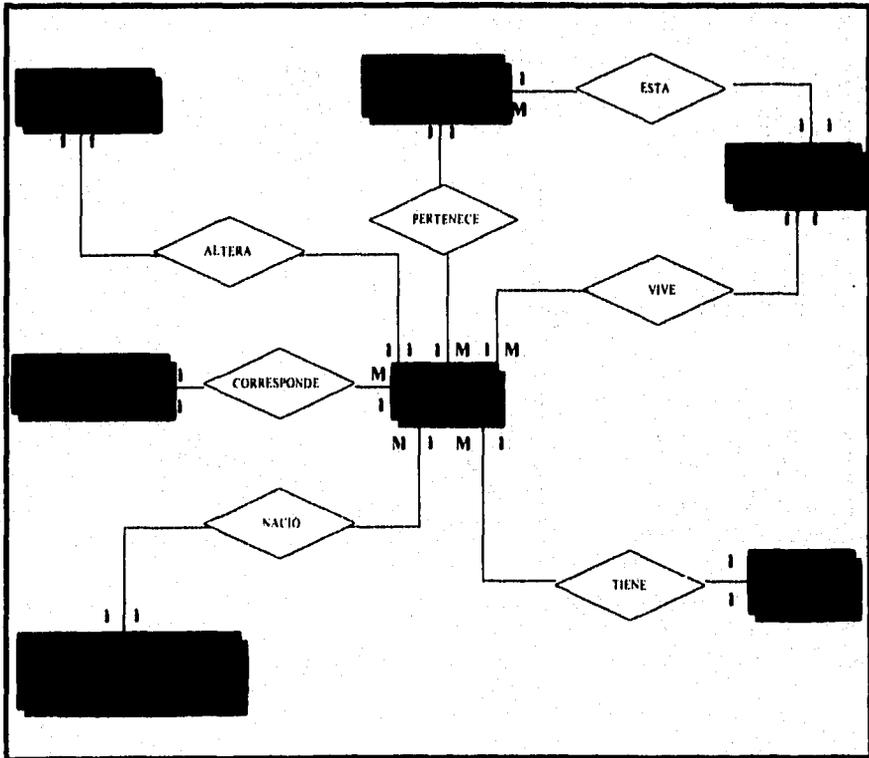


FIG. 3.2.5.1 Diagrama Entidad-Relación



El diagrama de entidad-relación anterior es un modelo conceptual, el cual se implementará físicamente, para nuestro caso en Microsoft Access ver. 2.0 el cual cuenta con un manejo gráfico para facilitar el proceso de relacionar las tablas de nuestro modelo, en la siguiente figura 3.2.5.2 se muestra como quedó el modelo entidad relación para Microsoft Access.

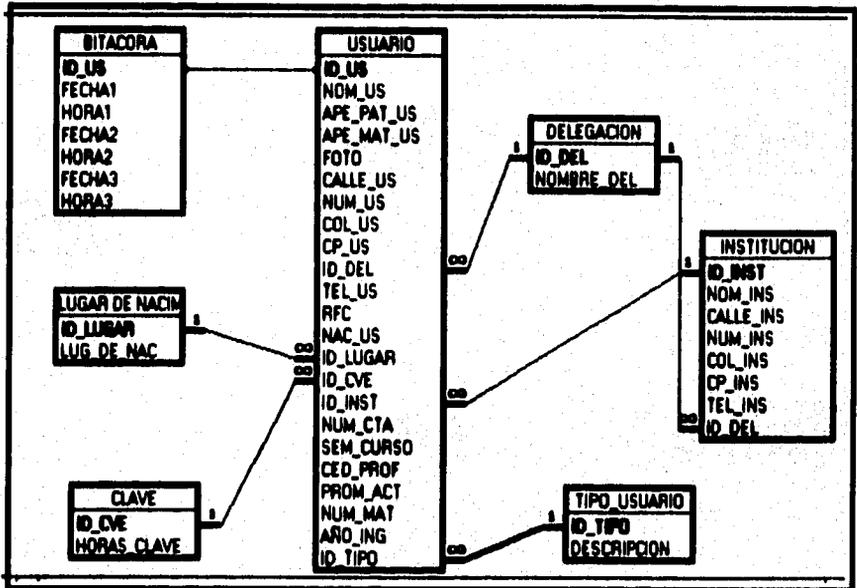


FIG. 3.2.5.2 Diagrama de entidad relación en access ver. 2.0



3.2.6 DESARROLLO DE LA BASE DE DATOS

Como se determinó previamente la base de datos se desarrollo en Microsoft Access ver 2.0, además es necesario capturar la información que vamos a requerir para el sistema. ésta puede ser capturada con las herramientas que Microsoft Access nos proporciona, sin embargo, considerando que a futuro la información puede ser modificada por personas que no estén familiarizadas con las herramientas de desarrollo se optó por generar un sistema en Access que permita dar de Altas, Bajas, Cambios y Consultas de datos en forma amigable.

Con la información obtenida en el análisis, se obtuvo que el sistema esta formado por siete tablas, por lo cual, se requerirán ciertas formas de capturar los datos. La manera de acceder a éstas será a través de pantallas que muestren alguna de las tablas, sobre las cuales se va a trabajar, en esta forma el usuario podrá dar altas, bajas y cambios de la información lo cual le da una mayor flexibilidad al sistema.

La implementación de lo antes mencionado en Access se llevó a cabo de la siguiente manera:

- A) Crear una base de datos en Microsoft Access
- B) Crear las tablas que componen a la base de datos
- C) Hacer las relaciones necesarias
- D) Realizar las formas necesarias, basándonos en las tablas, de ser necesario utilizando macros, e Integración de las formas en cada una de las pantallas, para la implementación de nuestro sistema de Manejo de Datos

A) Creación de la Base de Datos en Microsoft Acces

Primeramente al entrar a Access aparece un menú como el que se muestra en la figura 3.2.6.1, seleccionamos la opción de Archivos y elegimos crear una nueva base de datos.

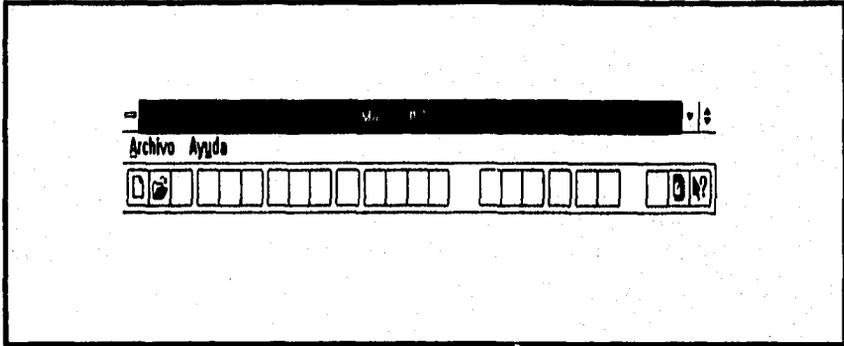


FIG 3.2.6.1 Menù inicial de Microsoft Access

Esta opción nos va a pedir el nombre que le vamos a dar a la base de datos, el cual puede ser de 8 caracteres como máximo. Una vez dado éste, damos aceptar. (Observar la figura 3.2.6.2

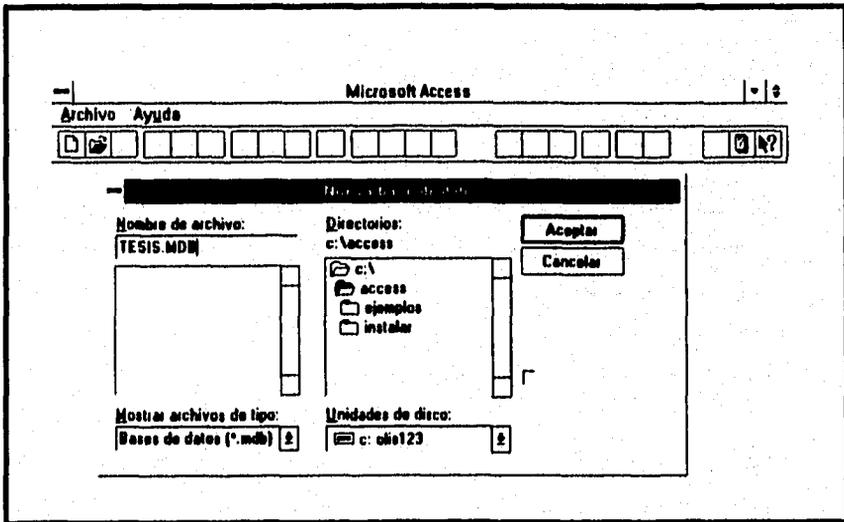


FIG. 3.2.6.2. Generación de una nueva Base de Datos



En la figura 3.2.6.3 se muestra la ventana de la Base de Datos, con los respectivos elementos que la forman.

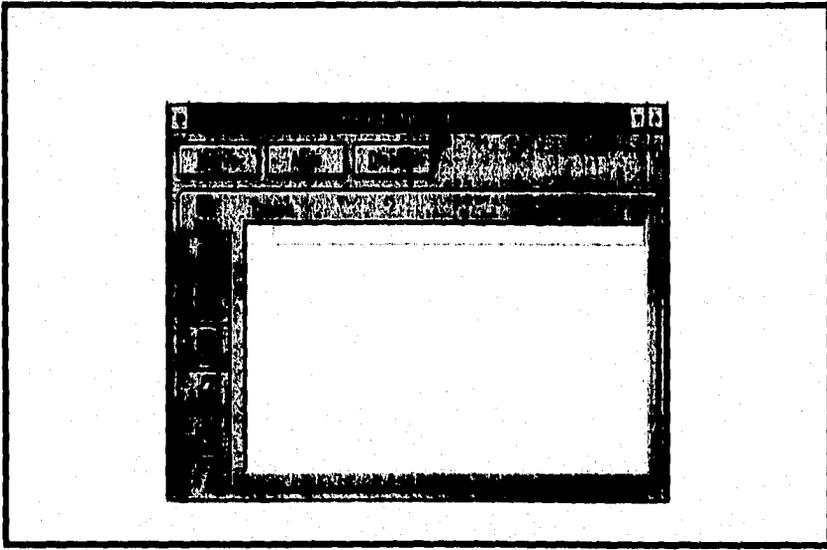


Fig. 3.2.6.3. Ventana de Base de Datos

Microsoft Access nos presenta una ventana en la que nos muestra diferentes opciones: tablas, consultas, formas, reportes, macros y módulos. Todas las opciones nos permiten crear un nuevo objeto o abrir uno ya existente.

Para utilizar una de las opciones de Microsoft Access se debe:

1. Elegir la opción deseada
2. Seleccionar uno de los tres botones siguientes:

Nuevo

para generar un nuevo objeto de la opción seleccionada

Abrir

para ver la información de un objeto

Diseñar

para editar el objeto



Al entrar en esta opción también contamos con una barra de herramientas de la cual, utilizaremos principalmente  el icono de especificar la llave primaria que deberemos presionar después de seleccionar el o los campos que deseamos sea la llave de nuestra tabla. Asimismo, vamos a utilizar el icono de guardar el objeto activo, con el cual salvaremos con un nombre nuestra tabla o bien las modificaciones realizadas a una ya existente.

La figura 3.2.6.5 muestra como creamos la tabla Usuarios que forma parte de nuestro sistema. En esta tabla se muestran los campos que la forman con sus respectivas características.

Nombre de campo	Tipo de datos	Descripción
<input checked="" type="checkbox"/> ID_US	Contador	
<input type="checkbox"/> NOM_US	Texto	NOMBRE(S) DEL USUARIO
<input type="checkbox"/> APE_PAT_US	Texto	APELLIDO PATERNO DEL USUARIO
<input type="checkbox"/> APE_MAT_US	Texto	APELLIDO MATERNO DEL USUARIO
<input type="checkbox"/> FOTO	Objeto OLE	FOTOGRAFIA DEL USUARIO
<input type="checkbox"/> CALLE_US	Texto	CALLE DEL DOMICILIO DEL USUARIO
<input type="checkbox"/> NUM_US	Texto	NUMERO DEL DOMICILIO DEL USUARIO
<input type="checkbox"/> COL_US	Texto	COLONIA DEL DOMICILIO DEL USUARIO
<input type="checkbox"/> CP_US	Númerico	CODIGO POSTAL DEL DOMICILIO DEL USUARIO
<input type="checkbox"/> ID_DEL	Númerico	IDENTIFICADOR DE LA DELEGACIÓN
<input type="checkbox"/> TEL_US	Texto	TELEFONO EN EL DOMICILIO DEL USUARIO
<input type="checkbox"/> RFC	Texto	REGISTRO FEDERAL DE CAUSANTES DEL USUARIO
<input type="checkbox"/> NAC_US	Fecha/Hora	FECHA DE NACIMIENTO DEL USUARIO (da/mes/año)
<input type="checkbox"/> ID_LUGAR	Númerico	IDENTIFICADOR DEL LUGAR DE NACIMIENTO

Propiedades del campo

Tamaño del campo 10

Formato

Máscara de entrada \#####0000-0000

Título

Valor predeterminado

Regla de validación

Texto de validación

Requerido No

Permitir longitud cero Sí

Indexado No

Un nombre de campo puede tener hasta 64 caracteres de longitud, incluyendo espacios. Presione F1 para buscar ayuda sobre los nombres de campo.

Presentación Diseño. F6 = Cambiar paneles. F1 = Ayuda

MAYÚ

FIG. 3.2.6.5. Tabla de Usuario de la Base de Datos



Como se ve en la figura anterior, en la parte superior de la ventana se muestran tres columnas que son: Nombre del campo, tipo de datos y descripción, y en la parte inferior de la ventana se muestran las propiedades del campo.

Así por ejemplo, para el campo de teléfono de usuario, el cual, para fines prácticos en nuestra base de datos se llama Tel_Us, tiene en la columna de tipo de dato texto y no numérico, esto con el fin de poderle poner una máscara de entrada y realizar la verificación conveniente, para que no acepte otros tipos de datos que no sean los requeridos, Microsoft Access tiene la ventaja de seleccionar una máscara a los campos, pero solo realiza esto en campos de tipo texto o fecha.

Una vez introducidos los campos de la tabla y validados todos los diferentes tipos de datos, se procede a cerrar la tabla en la cual aparece la pantalla de la figura 3.2.6.6 en donde se proporciona el nombre con el cual se quiere guardar.

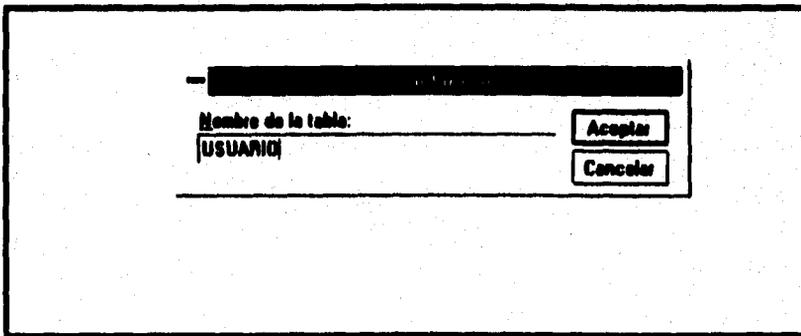


FIG. 3.2.6.6. Pantalla que muestra Access al salvar una tabla

Una vez explicado la manera en como se creó la tabla de USUARIO se procedió a realizar de manera similar las otras tablas que conforman a la base de datos, en la figura 3.2.6.7 se muestra el nombre de todas las tablas creadas.

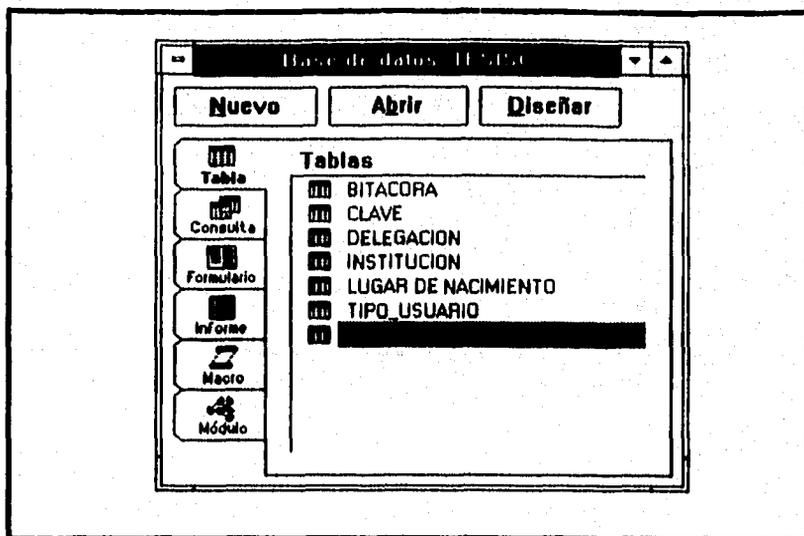


FIG. 3.2.6.7. Pantalla que muestra todas las tablas de la Base de Datos

C) Creación de las relaciones

En el menú de Microsoft Access se tiene la opción de relaciones,  al presionar este comando nos aparece en la pantalla que se muestra en la figura 3.2.6.8, en la cual existe la posibilidad de ir agregando las tablas que sean necesarias para determinar las relaciones que deseamos.

Las tablas relacionadas se representan mediante una línea que conecta los campos coincidentes. Microsoft Access utiliza las relaciones para crear automáticamente combinaciones en las nuevas consultas y para relacionar los registros mostrados en los subformularios. Las tablas deben estar relacionadas por campos equivalentes. No es necesario que los campos equivalentes tengan el mismo nombre, pero sí deben contener el mismo tipo de datos. Normalmente, las tablas se relacionan vinculando el campo de clave principal de una tabla con un campo coincidente de otra tabla. Las relaciones de las tablas de nuestra base de datos se muestran en la figura 3.2.6.9.

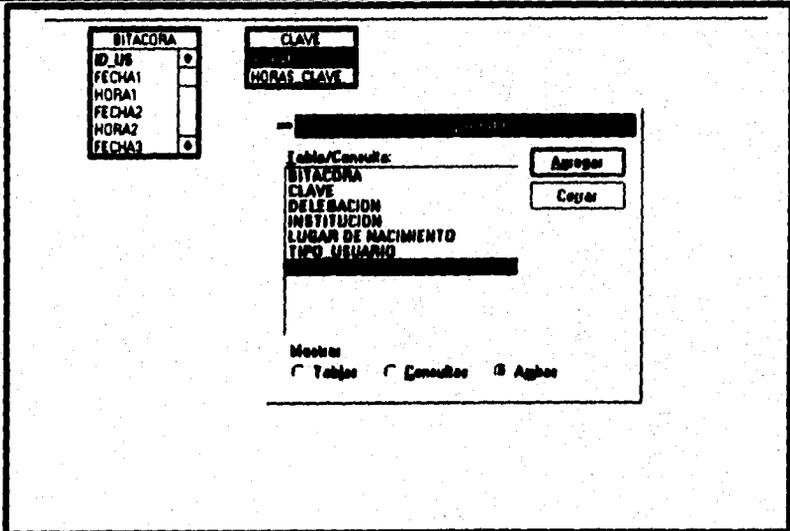


FIG. 3.2.6.8. Pantalla que muestra el diseño de las relaciones

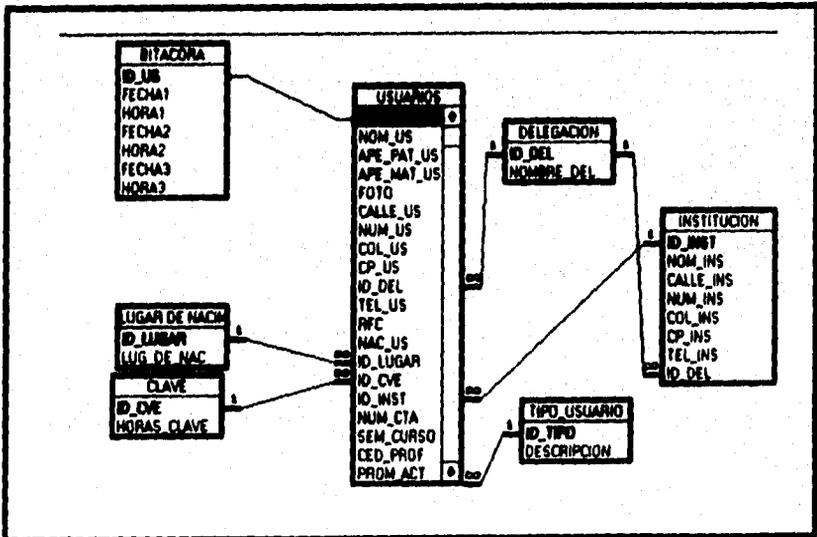


FIG. 3.2.6.9. Relaciones de la Base de Datos



D) Creación de la forma

Las formas se utilizan para ver, editar e imprimir información de una base de datos. Estas obtienen datos de una tabla o una consulta, para crear una forma se elige la opción de formularios y damos un clic en el botón de nuevo. Para esta opción, si nos vamos a auxiliar del asistente(figura 3.2.6.10) por lo cual, una vez seleccionada la tabla, o en nuestro caso la consulta en función de la cual se va a realizar el formulario, damos un clic en el botón de asistente para formularios.

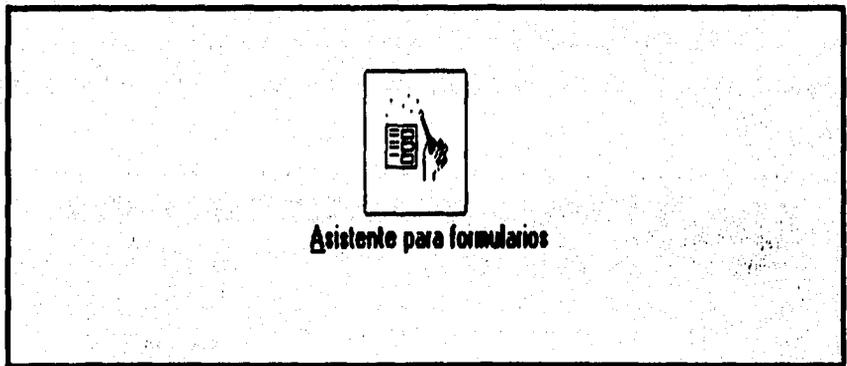


FIG. 3.2.6.10. Botón de Asistente para Formularios

Aparece una ventana donde nos muestra los diferentes tipos de formularios que podemos crear con ayuda del asistente: Tenemos en ella las siguientes opciones: columna simple que crea un formulario y que presenta los campos en una columna simple, tabular que presenta cada registro como una fila de campos, gráfico que crea un formulario que presenta un gráfico de los datos, principal/subformulario que es un formulario que contiene otro formulario y formulario automático que crea un formulario simple en forma automática sin pedimos ninguna información para su diseño.

Como se planteó anteriormente, se van a generar formas en función de las tablas que ya creamos. De las opciones, consideramos que formulario en blanco es la que nos permite diseñar los formularios en forma más amigable al usuario y una vez creados los mismos se editan y se le da un formato amigable.

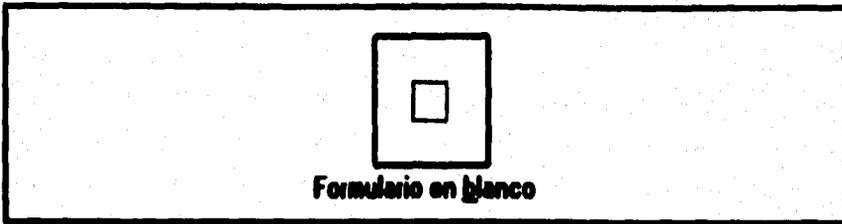


FIG. 3.2.6.11. Botón para un Formulario en Blanco

Una vez seleccionada la opción de formulario automático se presenta una pantalla con los campos que contiene la tabla seleccionada; en presentación de diseño de formulario se muestra en la figura 3.2.6.12

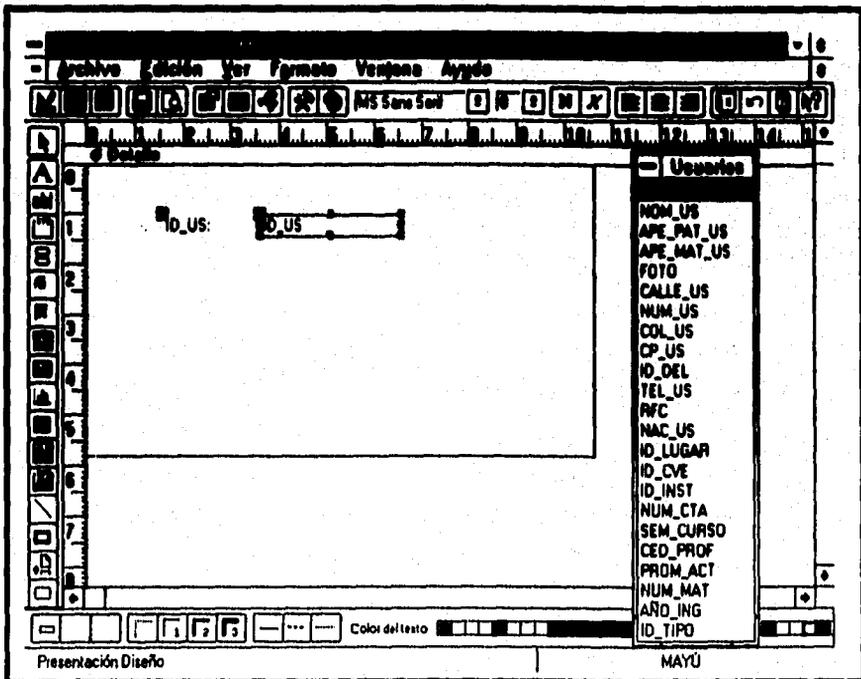


FIG. 3.2.6.12. Presentación de un Formulario en la Etapa de Diseño



Microsoft Access ajusta automáticamente el tamaño de la ventana a diseñar, la primera vez que usted abre un formulario. Sin embargo, no guarda esta configuración de ventana si posteriormente cambia a otro modo de presentación del formulario y cambia el tamaño de la ventana. Cuando vuelva al modo de presentación Formulario, tendrá que volver a elegir el comando *Ajustar* al formulario si desea mostrar registros completos. El tamaño de la ventana y el número de registros mostrados depende del valor de la propiedad *Presentación Predeterminada* del formulario.

Para auxiliarnos en el diseño de los formularios Microsoft Access cuenta con herramientas las cuales se muestran en la figura 3.2.6.13, estas herramientas son para el formulario en general además cuenta con herramientas para la creación de botones de comando, como se muestra en la figura 3.2.6.14

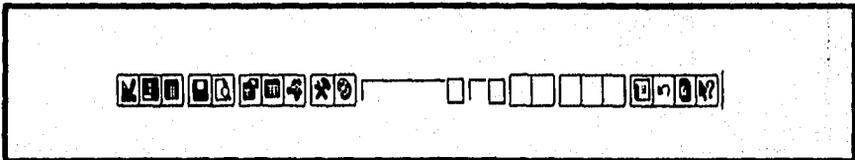


FIG. 3.2.6.13. Barra de Herramientas para el formulario

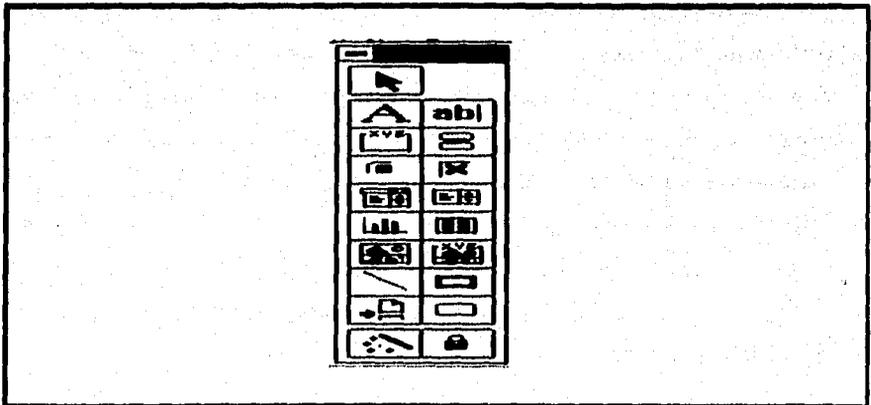


FIG. 3.2.6.14. Caja de herramientas para controles



Cada control, sección, formulario o informe cuenta con configuraciones de propiedades que sirven para alterar la apariencia o comportamiento de ese elemento determinado. Las propiedades se examinan y cambian en la hoja de propiedades.(ver figura 3.2.6.15)

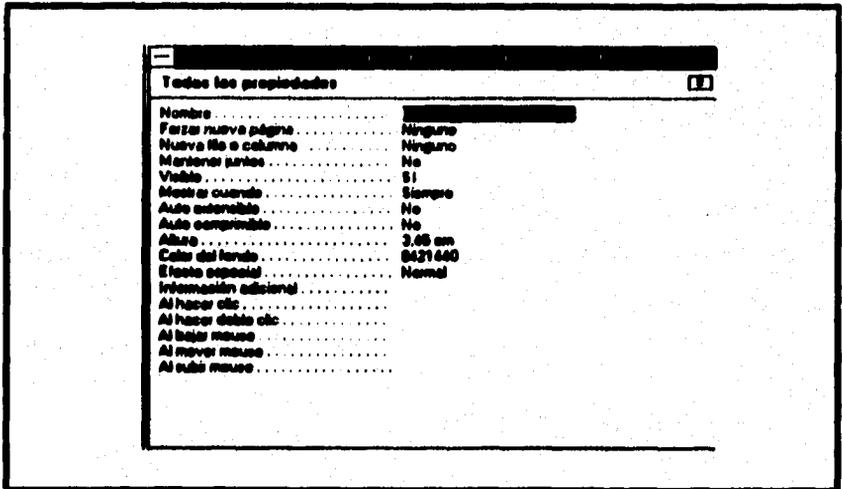


FIG. 3.2.6.15. Propiedades del formulario

Algunas propiedades se pueden establecer escribiendo una configuración de propiedad en el control mismo, lo que podría ser más fácil que escribirla en la hoja de propiedades; un control calculado en un formulario o informe y Crear etiquetas en un formulario o informe. Si la hoja de propiedades está visible y hace clic en un control que ya está seleccionado, la hoja de propiedades se despejará por completo. Si está introduciendo una configuración de propiedad en un control, presione la tecla ENTRAR para establecer el valor y volver a mostrar la hoja de propiedades. Si no está introduciendo una configuración, presione la tecla ESC para volver a mostrar la hoja de propiedades. Si lo desea puede cambiar las configuraciones predeterminadas de las propiedades de manera que los controles que cree tengan las nuevas configuraciones predeterminadas.

Una vez determinadas cuales son las herramientas a utilizar, se procedió a crear los formularios que nos van a ser de gran utilidad para la integración del sistema en Microsoft Access.



Empezaremos explicando la creación del formulario de Altas, como se dijo en párrafos anteriores una vez seleccionado el botón de formulario en blanco y seleccionado nuestra tabla de Usuarios, nos aparece una pantalla como la que se muestra en la figura 3.2.6.11, en la cual se muestran los campos de nuestra tabla, nos colocamos en cada uno de los campos y los arrastramos a nuestro formulario para posteriormente darle forma y acomodarlos de tal manera que aparezcan como nosotros deseamos y de la barra de herramientas seleccionamos una caja de texto, esto con el fin de asignarle un nombre a nuestra forma y que aparezca en tiempo de corrida (ver figura 3.2.6.16).

The screenshot shows a software interface for designing a form. At the top, there is a menu bar with options: Archivo, Edición, Ver, Formato, Ventana, Ayuda. Below the menu is a toolbar with various icons for editing and formatting. The main workspace is divided into two sections:

- Encabezado del formulario:** This section contains a single text box with the text "ALTAS DE USUARIO".
- Detalle:** This section contains several text boxes arranged vertically:
 - LOGIN (ID_US)
 - APELLIDO PATERNO (APE_PAT_US)
 - APELLIDO MATERNO (APE_MAT_US)
 - NOMBRE(s) (NOM_US)
 - DIRECCIÓN: CALLE (CALLE US)
 To the right of these text boxes is a large empty rectangular box labeled "FOTOGRAFÍA".

FIG. 3.2.6.16. Presentación del formulario de Altas en tiempo de Diseño.

Cada campo cuenta con un conjunto de propiedades que se usan para especificar cómo se desea almacenar, manejar y mostrar los datos. Las propiedades que pueden establecerse para cada campo dependen del tipo de datos que se seleccionen para el mismo. Para establecer una propiedad de campo:



- 1.- En el modo de presentación *Diseño de la tabla*, seleccione el campo cuyas propiedades desee establecer. (Figura 3.2.6.17)
- 2.- De las propiedades mostradas en la parte inferior de la ventana, haga clic en la que desee establecer.
- 3.- Establezca la propiedad.

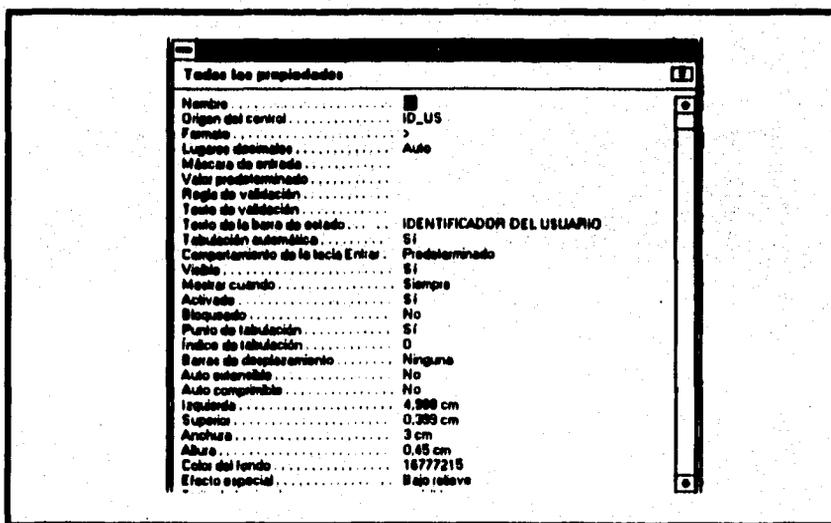


FIG. 3.2.6.17 Propiedades de los Campos.

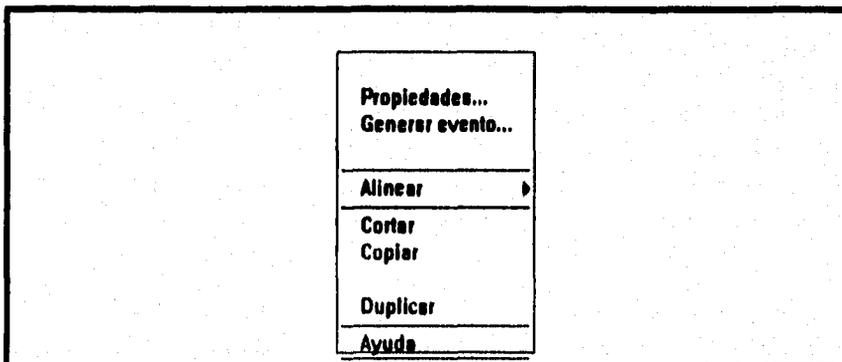


FIG. 3.2.6.18 Recuadro de Opciones



De esta manera se colocaron todas las propiedades a cada uno de los campos. Además para el campo de Delegación se tenían que implementar una macro. El Generador de macros es una herramienta de Microsoft Access que permite crear una macro desde una propiedad de evento de un formulario, informe, sección o control. Para utilizar el Generador de macros se coloca en el campo y se le dá un clic, el cual nos muestra el recuadro de la figura 3.2.6.18, después se coloca en la opción de propiedades y nos aparece el recuadro de la figura 3.2.6.17, para que posteriormente dependiendo de en cual evento se requiera se efectue la macro al seleccionara éste. Se procedió a la creación de la macro (ver figura 3.2.6.19) y diseñar que es lo que se desea que la macro realice, en nuestro caso se requiere abrir el formulario de altas y dar de alta a las delegaciones faltantes, es decir aquellas Delegaciones que no se encuentran registradas en el sistema. y lo mismo sucede para los siguientes datos: Lugar de Nacimiento, Tipo de Clave, Tipo de Usuario, Institución.

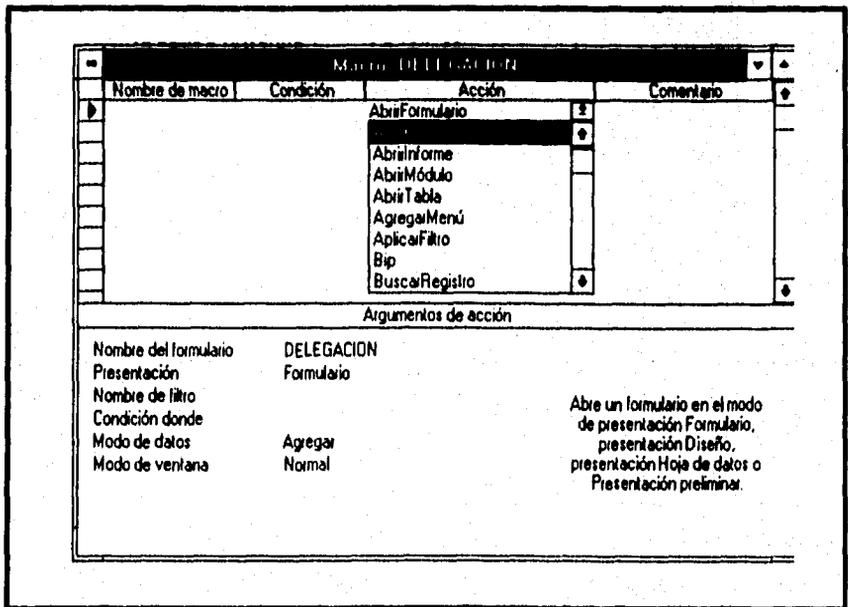


FIG. 3.2.6.19 Presentación de una macro



Una vez que se tuvieron todos los campos evaluados y las posibles macros a utilizar en cada uno de los campos requeridos, se procedió a darle una presentación amigable con la ayuda de botones de comandos. En el formulario de Altas se cuentan con tres botones de comandos, los cuales se agregan arrastrándolos de la caja de herramienta y colocándolos en la posición que se desee, enseguida se le agrega un código, para determinar su tarea a realizar, quedando finalmente el formulario de la figura 3.2.6.20 en la presentación de diseño.

ALTAS DE USUARIO

DESHACER REGISTRO ANTES DE SALVAR SALVAR EL REGISTRO SALIR

LOGIN ID_US FOTOGRAFÍA

APELLIDO PATERNO APE_PAT_US

APELLIDO MATERNO APE_MAT_US

NOMBRE(s) NOM_US

DIRECCIÓN : CALLE CALLE US

FIG. 3.2.6.20 Presentación de parte de la pantalla de Altas en la etapa de Diseño

Una vez terminado el formulario de altas se procedió a realizar de igual forma todos los formularios necesarios como son: Bajas, Cambios, Consultas, Delegación, menú de la pantalla principal(ver figura 3.2.6.21), entre otros. Y se fueron integrando uno a uno a través de comandos y macros, para quedar finalmente el sistema de Microsoft Access.

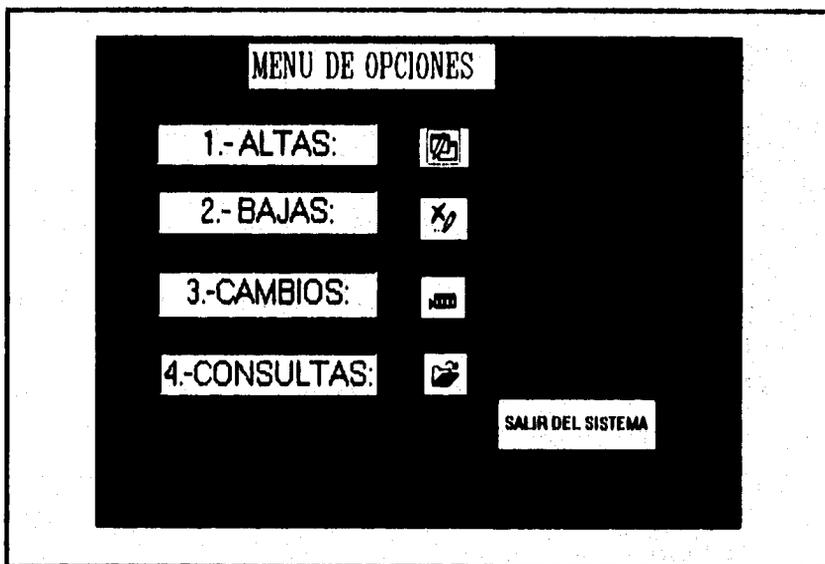


FIG. 3.2.6.21 Pantalla de Menù Principal en la Etapa de Corrida

3.3 DISEÑO Y CONSTRUCCIÓN DEL FRONT-END.

Como ya se ha mencionado el *Front-End* del "Sistema de monitoreo de claves de acceso a Internet para la validación del propietario a través de REDUNAM", se desarrolla en Visual Basic v. 3.0. y es precisamente en este tema donde se realizará y explicará el procedimiento de desarrollo del mismo.

Para crear una aplicación en Visual Basic podemos considerar los siguientes pasos:

- 1.- Crear un nuevo proyecto para organizar todas las partes que componen su aplicación. Un proyecto es un conjunto de pantallas y módulos de una aplicación. Para crear un proyecto nuevo seleccione la opción *New Project* del menú *File* (Archivo).
- 2.- Crear una forma para cada ventana de la aplicación. Recordamos que una forma es una interfaz gráfica entre su aplicación y el usuario. Para crear una nueva forma se selecciona la opción *New Form* del menú *File*.
- 3.- Diseñar los objetos para cada forma. Sabemos que un objeto gráfico es creado por el Caja de Herramientas. Cada objeto posee su propio conjunto de sucesos, propiedades y métodos. Para diseñar un objeto seleccione la herramienta deseada de la Caja de Herramientas. A continuación se pulsa el botón del ratón en la pantalla en el lugar donde se desea insertar ese objeto. Se fija el objeto en la posición que se cree conveniente dentro de la forma.
- 4.- Se crea una barra de menús (si se desea) para la pantalla principal..
- 5.- Asociar formas y propiedades de los objetos.
- 6.- Escribir los procedimientos de sucesos y los procedimientos genéricos, si los hubiera.
- 7.- Almacenar los archivos del proyecto.
- 8.- Crear un archivo ejecutable para transformar su proyecto en una aplicación independiente bajo ambiente Windows.



Diseño del Front-End del Sistema

Partiendo de que debe ser visualmente agradable y de fácil manejo para el usuario, se determinó desarrollar un diseño modular en la que la presentación de los datos quedara agrupada conforme a características similares para su rápido acceso.

Por otra parte, es en esta etapa donde se realiza un análisis de color, para determinar las propiedades de todos los controles involucrados en la pantalla y de la forma misma; concluyendo que los colores más adecuados para la presentación del sistema son: la combinación de grises, blancos, pasteles y los colores primarios para el despliegue de gráficos, esto conforme a lo determinado por Microsoft.

De lo antes mencionado, se procede a la implementación de las pantallas:

Pantalla de Altas

Para el desarrollo de esta pantalla, se realizaron los siguientes pasos:

1. Abrir Visual Basic desde el administrador de programas de Windows
2. Seleccionar del Menú de *File* la opción de *New Form* o hacer un clic en el icono de forma de la Barra de Menú.
3. Presione F4 para desplegar las propiedades de la forma donde se definieron las siguientes propiedades:

Clip Control=false

Control Box = False

Max Button =False

Min Button = False

Window State= 2 Maximized

Caption = "Sistema de Monitoreo y Verificación OLIS"

Border Style = 1 Single

BackColor = &H008008&



4. Para crear la Barra de Menú del Sistema seleccione del Menú *Window*, la opción *Menu Design*; en este momento se despliega en pantalla una ventana flotante, en la cual especificamos las opciones siguientes:

Caption = Captura

Name = mCaptura

Para poder formar hijos en este Menú es necesario hacer un clic en la flecha que indica hacia la derecha, se observa que se colocan cuatro puntos en la ventana, para regresar a un nivel anterior es necesario presionar la flecha que apunta hacia la derecha; posteriormente se coloca el cursor en la parte superior de la ventana para definir las propiedades de *Caption* y *Name* para cada Menú y cada opción deseada quedando de la siguiente manera:

CAPTION	NAME	NIVEL
Captura	mCaptura	1
Altas	smAltas	2
Bajas	smBajas	2
Cambios	smCambios	2
Consulta	mConsulta	1
Monitoreo	mMonitoreo	1
Manual	smManual	2
Automático	smAutomático	2
Estadísticas	mEstadísticas	1
Rendimiento	smRendimiento	2
Procesamiento	smProcesamiento	2
Bitácora	smBitácora	2
Salir	mSalir	1
Ayuda	mAyuda	1
Contenido	smContenido	2
Buscar Topico	smBuscar	2
Acerca de....	smAcerca	2

5. Para desarrollar la Barra de Herramientas seleccionamos de la Caja de Herramientas el control *Frame*, ajustando sus propiedades de la siguiente manera:

```
Top = 0
Left = 0
Width = Form4.Width
Height = 1000
Caption = ""
Border = 1
Align = Top
```

6. Para incrustar los botones de la Barra de Herramientas se selecciona un control de imagen arrastrándolo hacia el panel de control, ajustando sus propiedades como:

```
Top = Frame1.Top + 10
Left = Frame1.Left + 10
Width = 400
Height = Form1.Height - 20
Picture = LoadPicture("C:\VB\ICONS\HERR.ICO")
```

Este proceso se repite para cada botón de la barra de herramientas, asignando en su propiedad de *picture* lo siguiente además de sumarle 10 unidades en cada propiedad *Left*.

```
Picture = LoadPicture("C:\VB\ICONS\HERR2.ICO")
Picture = LoadPicture("C:\VB\ICONS\HERR.3.ICO")
Picture = LoadPicture("C:\VB\ICONS\HERR4.ICO")
Picture = LoadPicture("C:\VB\ICONS\HERR4.ICO")
Picture = LoadPicture("C:\VB\ICONS\HERR5.ICO")
Picture = LoadPicture("C:\VB\ICONS\HERR6.ICO")
```



Pantalla de Password

1. Seleccionar un control de forma de la Caja de Herramientas, dimensionarlo automáticamente con el arrastre del mouse y definir las siguientes propiedades:

Caption = Password
Border Style = Fixed
Clip Control = False
Box Controls = False
MinButton = False
MaxButton = False
Window State = 2 Maimized
BackColor = &H098000&

2. Seleccione y arrastre una Caja de Texto con máscara o en su defecto una Caja de Texto y defina sus propiedades:

Char Password = *
BackColor = &H00876&
Color = &H09800&

3. Seleccione un control de etiqueta de tercera dimensión (Panel) y definalo como:

Visible = False
ForeColor = &H098000&
Autosize = False

Pantalla de Gráficas

1. Seleccione una nueva forma cuyas propiedades serán:

Caption = Password



Border Style = Fixed
Clip Control = False
Box Controls = False
MinButton = False
MaxButton = False
Window State = 2 Maimized
BackColor = &H098000&

2. Seleccione y arrastre un control *Graph* dimensiónelo con el arrastre del Mouse según sus necesidades, deje las propiedades en default ya que estas se definen mediante programación.

3. Coloque en la forma un control de Comando de tercera dimensión y dimensiónelo.

Caption = &Imprimir

4. Elija un control *Frame* y dimensiónelo y defina sus propiedades:

Caption = Tipo de Gráfico

5. Seleccione y coloque dentro del *Frame* seis controles de opciones y para cada uno defina su propiedad

Caption con las siguientes leyendas:

Pie 2D

Pie 3D

Barras

Superficie

Lineas

Cruces

6. Seleccione y coloque un cuadro de examinación en el *Frame* asignando su propiedad *Caption* como separador.

Este proceso se aplica a todas y cada una de las pantallas que conforman nuestro sistema; para profundizar la construcción de pantallas en Visual Basic favor de referirse al apéndice B.



3.3.1 DESARROLLO DE CADA UNA DE LAS PANTALLAS PARA MONITOREO, VERIFICACIÓN Y TOMA DE DECISIONES.

Visual Basic es un sistema de desarrollo dirigido para la creación de aplicaciones bajo el ambiente Windows, o en otras palabras, aplicaciones esencialmente gráficas. De esta forma, las pantallas, ventanas o *formas*; son tremendamente fáciles de crear y no necesitan prácticamente ninguna experiencia sobre la lista de mandatos del lenguaje de programación. Las formas a su vez contienen *objetos*, que son denominados *controles*. Algunos de estos controles son las opciones del usuario dentro de las ventanas. A continuación, el programa, a través de las pantallas y controles, deben responder a las acciones de los usuarios, denominadas *eventos*. Es decir, un evento es una acción reconocida por Visual Basic.

Para nuestro sistema se crearon 5 *formas principales*:

FORMA 1	Presenta el escudo de la UNAM.
FORMA 2	Presenta créditos y agradecimientos.
FORMA 3	Pide Passwords de administradores del sistema.
FORMA 4	Permite realizar los movimientos necesarios en la base de Datos: Captura, Consultas, Monitoreo, Estadística y Bitácora del Sistema.
FORMA 5	Realiza y envía los cuestionarios al usuario monitoreado.

FIG. 3.3.1.1 Formas del Front-end del Sistema.

Creando la forma 1.

Para el diseño de esta pantalla solo se necesitaron 4 objetos: la misma forma, dos botones de comando y un cuadro de imagen; estos objetos se toman del *cuadro de herramientas*.



Cuadro de imagen: Puede ser un icono u otro elemento gráfico, como un *bitmap* o *metafile* creado en otra aplicación. En nuestro sistema se cargo el archivo **escudo.wmf** que contiene el escudo de la UNAM en tercera dimensión..

Botón de comando: Controla cuando el usuario pulsa e inmediatamente ejecuta una operación específica o selecciona una opción de una aplicación. Un botón de mandato ejecuta una tarea específica seleccionada por el usuario. Para mostrar el texto en el botón de comando, se ajusto la propiedad **Caption**. Como se puede ver en la figura 3.3.1.2, esta propiedad se identificó como "Salir" y "Continuar" respectivamente; otra propiedad que se especifico fue **BackColor**, esta propiedad se aplica a la pantalla, botón de mandato y otros más, se encarga de ajustar o retornar el color de fondo de un objeto. La propiedad **ForeColor** ajusta o retorna el color del primer plano (tinta) de un objeto.

En Visual Basic, cada objeto tiene propiedades, es decir, a cada objeto le damos una serie de características como son: color, tamaño, leyenda, etc.; para proporcionarles éstas, solo presionamos F4 previamente seleccionado el objeto, y así seleccionamos las características deseadas del cuadro de propiedades.

OBJETO	PROPIEDAD	VALOR
FORMA 1	<i>Name</i> <i>Caption</i> <i>WindowState</i>	Presentación PRESENTACIÓ N 2
BOTON	<i>Name</i> <i>Caption</i>	Command3D1 &Continuar
BOTÓN	<i>Name</i> <i>Caption</i>	Command3D2 &Salir
CAJA IMAGEN	<i>Name</i> <i>Picture</i>	Picture1 Escudo.bmp

FIG. 3.3.1.2 Propiedades de los objetos de la Forma 1.



En esta forma al presionar el Botón de Comandos *Command1*, llama la forma 2; al presionar el botón de comandos *Command2*, sale del programa. También en esta forma se presenta una rutina la cuál nos presenta una animación al dar un click en la forma.

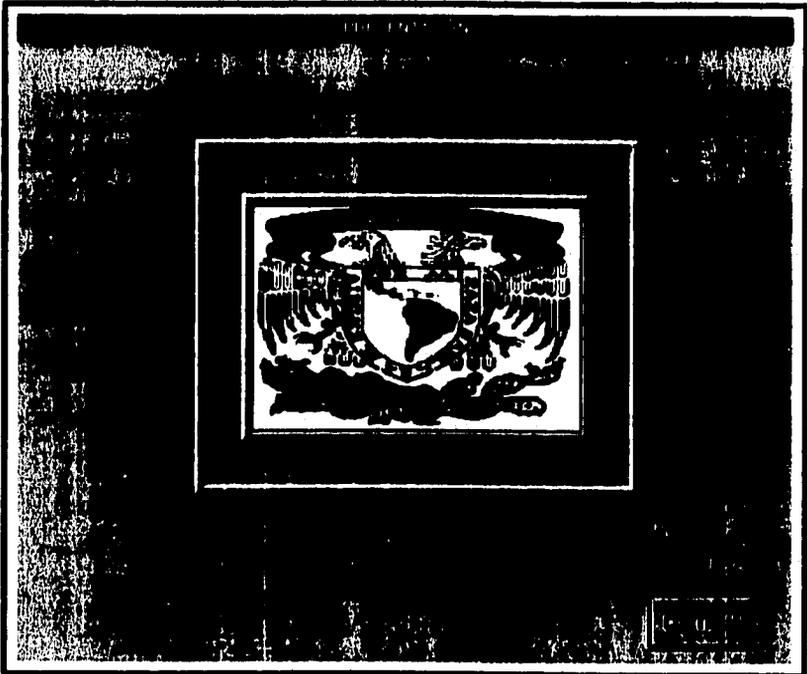


FIG. 3.3.1..3. Forma Presentación del sistema.

Creando la forma 2.

En esta forma se incrustaron 21 objetos, como son: líneas, cuadros de imágenes donde se despliegan los escudos de las dependencias DCAA y DGSCA, también para las fotos de los integrantes del equipo de desarrollo del Sistema, botones de comando, etc.



OBJETO	PROPIEDAD	VALOR
FORMA	<i>Name</i> <i>Caption</i> <i>WindowState</i>	Agradecimiento AGRADECIMIENTOS 2
BOTÓN	<i>Name</i> <i>Caption</i>	Command3D1 &Continuar
PANEL	<i>Name</i> <i>Caption</i>	Panel3D1 Hacemos patente nuestro agradecimiento a todas aquellas personas que directamente o indirectamente han contribuido a la consecución de este proyecto y a nuestra formación profesional en general
PANEL	<i>Name</i> <i>Caption</i>	Panel3D2 Atentamente:
PANEL	<i>Name</i> <i>Caption</i>	Panel3D3 Dirección General de Servicios de Cómputo Académico
PANEL	<i>Name</i> <i>Caption</i>	Panel3D4 Dirección de Cómputo para la Administración Académica
ETIQUETA	<i>Name</i> <i>Caption</i>	Label2 A través de:
CAJA IMAGEN	<i>Name</i> <i>Picture</i>	Picture1 Escdcaa.bmp
CAJA IMAGEN	<i>Name</i> <i>Picture</i>	Picture2 Escdgsca.bmp



CAJA IMAGEN	Name Picture	Picture3 foto1.bmp
CAJA IMAGEN	Name Picture	Picture4 foto2.bmp
CAJA IMAGEN	Name Picture	Picture5 foto3.bmp
CAJA IMAGEN	Name Picture	Picture6 foto4.bmp
CAJA IMAGEN	Name Picture	Picture7 foto5.bmp
LINEA	Name BorderWidth	Line 1 1
LINEA	Name BorderWidth	Line 2 1
LINEA	Name BorderWidth	Line 3 1
LINEA	Name BorderWidth	Line 4 1

FIG. 3.3.1.4. Propiedades de los objetos de la Forma 2.

En la figura 3..3.1.4. observamos otros tipos de objetos como son:

Etiqueta: Identifica un elemento de un cuadro de diálogo. Puede ser utilizado para mostrar textos extensos, que no necesitan modificaciones durante la ejecución de la aplicación. Se puede ajustar solamente una línea de texto con la propiedad *Caption*, por ejemplo en la fig. 3.3.1.3 podemos ver que en esta propiedad de las etiquetas se utilizó para desplegar mensajes de texto como: "Dirección General de Servicios de Cómputo Académico , etc. En este objeto también encontramos otras propiedades como: *FontBold*, *FontItalic*, *FontStrikethru*, *FontTransparent* y

FontUnderline, estas propiedades se aplican a las formas, cuadro de verificación, botón de comando, marco, cuadro de listas, botón de opción, cuadro de imagen y cuadro de textos; y determinan el estilo de los tipos de letra que serán utilizados en los objetos.

El procedimiento que realiza esta forma es el siguiente: al cargar esta forma otro procedimiento descarga la forma 1; al dar un click al botón de comandos se carga la forma 3.

Observando las figura 3.3.1.1 y la figura 3.3.1.2. nos damos cuenta que el número de objetos en cada forma varía según los requerimientos del sistema o por el mismo diseño de cada una de ellas. También es importante notar que las propiedades de cada objeto son diferentes e independientes; además la mayoría de las propiedades son asignadas por el sistema, por ejemplo las posiciones de las líneas, las da el sistema ya que nosotros solo indicamos con el mouse donde queremos ponerla.

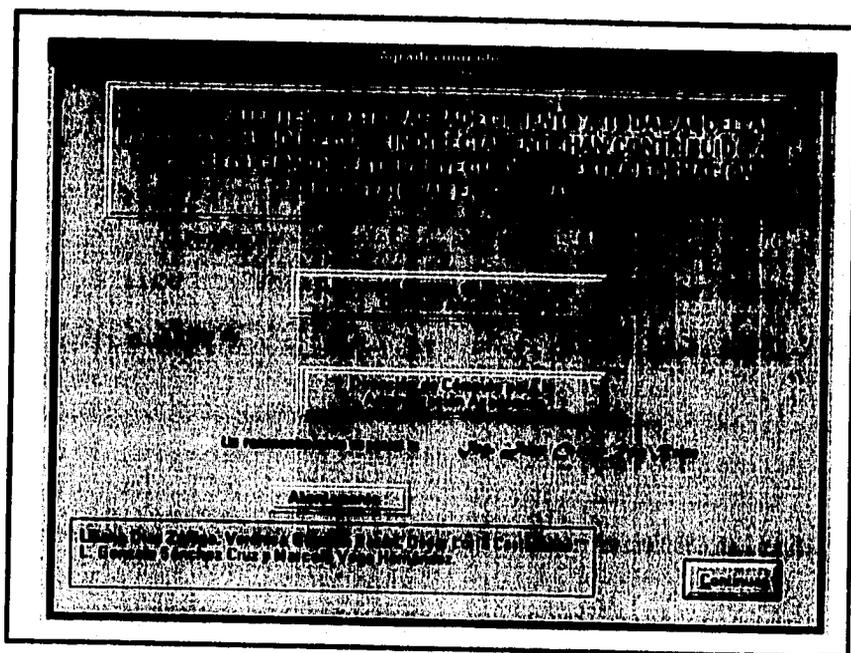


FIG. 3.3.1.5. Forma Agradecimiento del Sistema.



Creando la forma 3.

Esta forma es la encargada de autorizar la entrada al sistema, mediante el uso de *passwords* los cuales serán proporcionados a los administradores del sistema.

Esta forma esta constituida por 4 objetos: la misma forma, un panel y dos imágenes; además contiene dos procedimientos: uno encargado de descargar la forma 2 y el otro encargado de comprobar el password de entrada y registrar en la bitácora la hora y fecha de entrada al sistema así como el nombre del administrador.

OBJETO	PROPIEDAD	VALOR
FORMA 1	<i>Name</i> <i>Caption</i> <i>WindowState</i>	Presentación PRESENTACION 2
BOTÓN	<i>Name</i> <i>Caption</i>	Command3D1 &Continuar
BOTÓN	<i>Name</i> <i>Caption</i>	Command3D2 &Salir
CAJA IMAGEN	<i>Name</i> <i>Picture</i>	Picture1 Escudo.bmp

FIG. 3.3.1.6. Propiedades de los objetos de la forma 3.

En esta forma se tienen códigos de seguridad para evitar que cualquier otra persona ajena al sistema pueda acceder al mismo. Se presenta una Caja de diálogo, en la cuál se escribe el password y si este es correcto continua la ejecución del sistema, si no es correcto, envía el mensaje "PASSWORD INCORRECTO" y continua en la misma forma sin poder salir del programa.

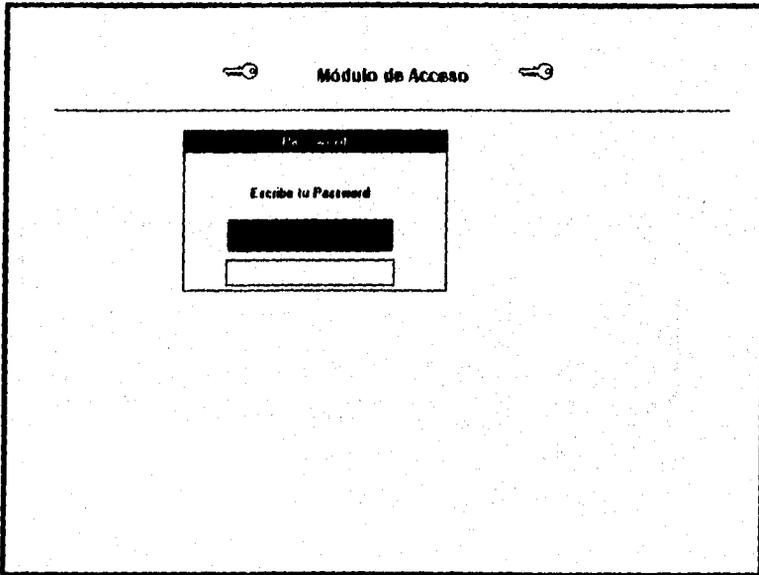


FIG. 3.3.1.7. Forma Acceso del Sistema.

Creando la forma 4.

Esta forma es la interface entre el administrador y el propio sistema, es decir, le permite al administrador monitorear el funcionamiento y rendimiento del sistema como son: estadísticas, monitoreo de usuarios, consultas, y actualización de usuarios; debido a estas características esta forma solo contiene un menú que contiene las siguientes características.

OBJETO	PROPIEDAD	VALOR
FORMA	<i>Name</i>	sistema
	<i>Caption</i>	Programa de Monitoreo y verificación de claves.
	<i>Control Bar</i>	False
	<i>MaxButton</i>	False
	<i>MinButton</i>	False



MENU CAPTURA	Name Caption	mcap &Captura
CAPTURA: orden 0	Name Caption	smalt Altas
CAPTURA: orden 1	Name Caption	smbaj Bajas
CAPTURA: orden 2	Name Caption	smcam Cambios
MENU CONSULTAS	Name Caption	mcon C&onsultas
MENU MONITOREO	Name Caption	mmon &Monitoreo
MONITOREO: orden 0	Name Caption	smman Manual
MONITOREO: orden 1	Name Caption	smaut Automático
MENU ESTADÍSTICAS	Name Caption	mest &Estadísticas
ESTADÍSTICAS : orden 0	Name Caption	smren Rendimiento
ESTADÍSTICAS : orden 1	Name Caption	smpro Procesamiento
ESTADÍSTICAS : orden 2	Name Caption	smbit Bitácora
MENU SALIR	Name Caption	msal &Salir
MENU AYUDA	Name Caption	may &Ayuda

FIG. 3.3.1.8. Propiedades de los objetos de la forma 4.

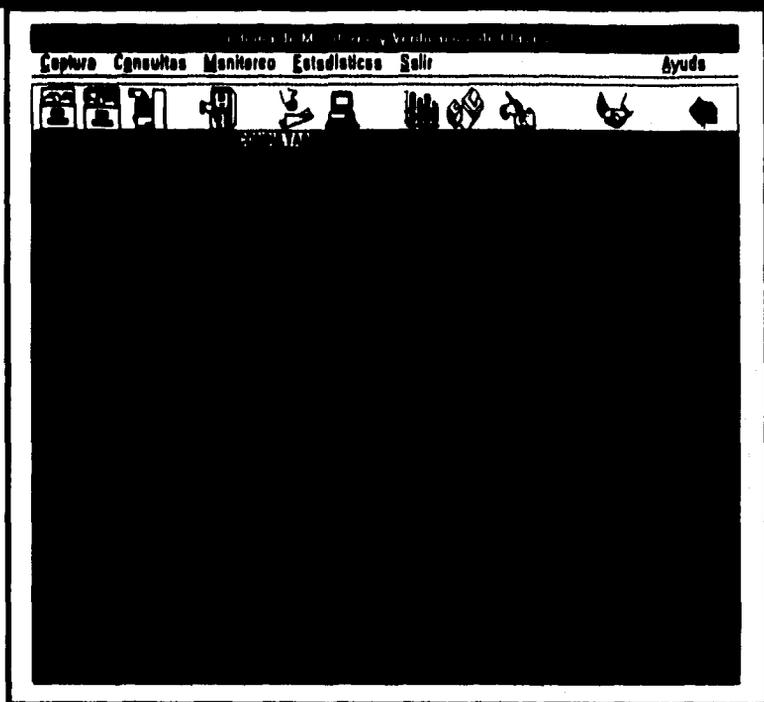


FIG. 3.3.1.9. Forma Sistema.

En las siguientes formas solo mencionaremos los objetos que se utilizan ya que las propiedades de cada forma se presentarán en el listado del sistema.

Lo que a continuación se describe es la manera en como quedan las pantallas de los submódulos del menú principal.

Empezamos por " Captura ", al seleccionar esta opción en el menú principal se despliegan tres opciones las cuales son: Altas, Bajas y Cambios. Si seleccionamos altas se despliega la pantalla que se muestra en la Figura 3.3.1.10, en la cual se capturan los datos del usuario, además esta forma



cuenta con 3 comandos de: salvar(introduce los datos ya capturados), nuevo(capturamos un nuevo registro) y salida(finaliza con la opción de altas).

Alta de Usuarios

Login

Nombre (N)
 Apellido Paterno
 Apellido Materno
 Fotografía

Dirección
 Calle
 Número
 Colonia
 Código Postal
 Delegación o Municipio
 Teléfono

Otros Datos
 Institución
 Lugar Nacimiento
 No Cuenta
 Fec Nacimiento
 Semestre Actual
 RFC
 Promedio Actual
 Cédula Prof.
 No Materias
 Tipo de Clave
 Año de Ingreso
 Tipo de Usuario

FIG. 3.3.1.10 Muestra la pantalla de Altas de Usuarios

En la opción de bajas, se despliega la pantalla de la figura 3.3.1.11, en la cual su funcionamiento principal; es la de eliminar los registros de los usuarios que por diferentes causas ya no pertenecen a nuestra base de datos.

En la opción de Cambios, se tiene la posibilidad de realizar cambios en los datos de los usuarios, si es que se requiere y es necesario.



La opción "Consulta" del menú principal, cuenta con la opción de Tipos, al seleccionarla despliega un menú en el cual se puede analizar 3 diferentes tipos de consultas. (ver figura 3.3.1.12) y una vez seleccionada la consulta específica, por ejemplo "Consulta por RFC", aparece la pantalla en la cual se van a tener los datos del usuario a consultar. (Ver figura 3.3.1.13)

The diagram illustrates the 'MÓDULO DE BAJAS' (User Deletion Module) interface. It is enclosed in a thick black border. At the top center, the title 'MÓDULO DE BAJAS' is displayed. Below the title, the interface is divided into two main sections: 'CONTROL' and 'SALIDA'. The 'CONTROL' section is a large rectangular area containing a sub-section with the title 'CONTROL' at its top center. This sub-section includes a label 'IDENTIFICADOR DE USUARIO (ID)' next to an empty rectangular input field, and a button labeled 'VERIFICAR USUARIO' below it. To the right of the input field is a small square icon. The 'SALIDA' section is a smaller rectangular area located to the right and below the 'CONTROL' section, containing a button labeled 'ACTIVAR'.

FIG. 3.3.1.11 Muestra la pantalla de Bajas de Usuario



MENU DE CONSULTAS

FILTRO DE BÚSQUEDA

APELLIDO PATERNO

LOGIN

TIPO DE USUARIO

DEPENDENCIA

SALIDA

FIG. 3.3.1.12 Pantalla de los tres tipos de Búsqueda

Login	<input type="text" value="RFLAS001"/>		
	<input type="text" value="LUIS ALFONSO"/>	<input type="text" value="BLAZ"/>	<input type="text" value="FURUSA"/>
	Nombre (s)	Apellido Paterno	Apellido Materno
	<input type="button" value="Fotografía"/>		
Dirección	<input type="text" value="BIENES NACIONALES"/>	<input type="text" value="50"/>	<input type="text" value="CUATRO AGOSTOS"/>
	Calle	Número	Colonia
	<input type="text" value="10700"/>	<input type="text" value="Escam"/>	<input type="text" value="7822001"/>
	Código Postal	Delegación o Municipio	Teléfono
Otros Datos		Lug. Nacimiento	<input type="text" value="MEXICO, DF"/>
Institución	<input type="text" value="FACULTAD DE INGENIERIA"/>	Fec. Nacimiento	<input type="text" value="09/01/79"/>
No. Cuenta	<input type="text"/>	RFC	<input type="text" value="0001700100"/>
Semestre Actual	<input type="text"/>	Cédula Prof.	<input type="text" value="0"/>
Promedio Actual	<input type="text" value="0"/>	Tipo de Clave	<input type="text" value="100 MENS"/>
No. Materias	<input type="text" value="0"/>	Tipo de Usuario	<input type="text" value="EXTERNO"/>
Año de Ingreso	<input type="text"/>		
<input type="button" value="Imprimir"/>		<input type="button" value="Aceptar"/>	

FIG. 3.3.1.13 Pantalla de Consultas Especifica



En la opción de "Monitoreo", existen dos maneras de monitorear al usuario : monitoreo manual y Monitoreo Automático.

En el menú general aparece la opción de "Estadísticas" la cual contiene lo siguiente: Rendimiento (ver figura 3.3.1.14), Procesamiento y Bitácora (ver figura 3.3.1.15)

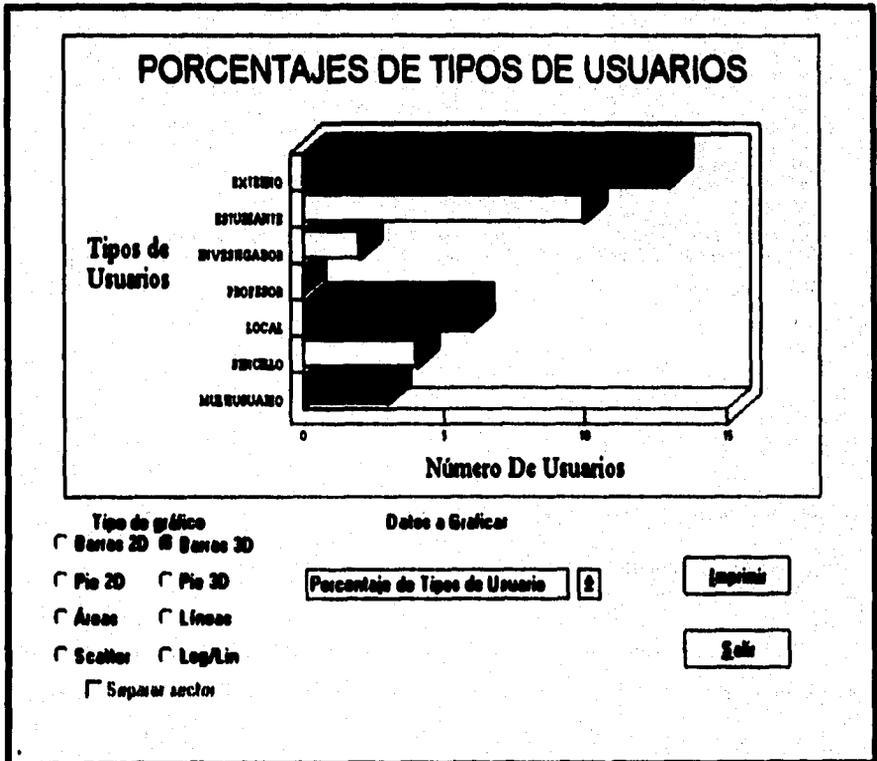


FIG. 3.3.1.14. Muestra las Estadísticas



Búsquedas Salir					
Nº Acceso	Administrador	Fecha de Entrada		Fecha de Salida	
Nº 1	SUPER USUARIO	30/05/1996	20:23:04	30/05/1996	20:23:20
Nº 2	ADMINISTRADOR B	30/05/1996	20:23:23	30/05/1996	20:24:16
Nº 3	SUPER USUARIO	30/05/1996	20:24:19	30/05/1996	20:24:21
Nº 4	ADMINISTRADOR A	30/05/1996	20:31:50	30/05/1996	20:44:34
Nº 5	SUPER USUARIO	30/05/1996	20:44:36	30/05/1996	20:44:38
Nº 6	ADMINISTRADOR B	30/05/1996	20:47:03	30/05/1996	20:49:10
Nº 7	SUPER USUARIO	30/05/1996	20:49:11	30/05/1996	20:49:15
Nº 8	ADMINISTRADOR B	30/05/1996	20:52:23	30/05/1996	20:52:56
Nº 9	SUPER USUARIO	30/05/1996	20:52:57	30/05/1996	21:00:01
Nº 10	SUPER USUARIO	30/05/1996	21:00:34	30/05/1996	21:00:55
Nº 11	SUPER USUARIO	30/05/1996	21:01:10	30/05/1996	21:01:53
Nº 12	SUPER USUARIO	30/05/1996	21:09:05	30/05/1996	21:09:54
Nº 13	SUPER USUARIO	30/05/1996	21:11:52	30/05/1996	21:12:20
Nº 14	SUPER USUARIO	30/05/1996	21:13:52	30/05/1996	21:14:24
Nº 15	SUPER USUARIO	30/05/1996	21:15:04	30/05/1996	21:15:33

FIG. 3.3.1.15 *Búsqueda del Sistema*

La opción de ayuda como en todos los sistemas, debe de contemplarse, y aquí no es la excepción, en este sistema la ayuda cuenta con tres opciones ver figura (3.3.1.16) como son: Contenido, Buscar Tópico (ver figura 3.3.1.17) y Acerca de. Y por ultimo la opción de "Salir".

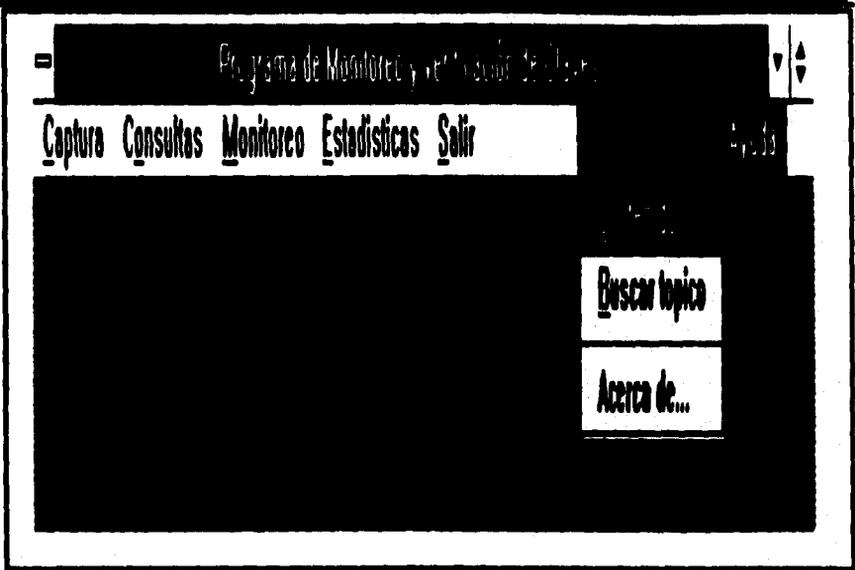


FIG. 3.3.1.16 Menú de Ayuda General

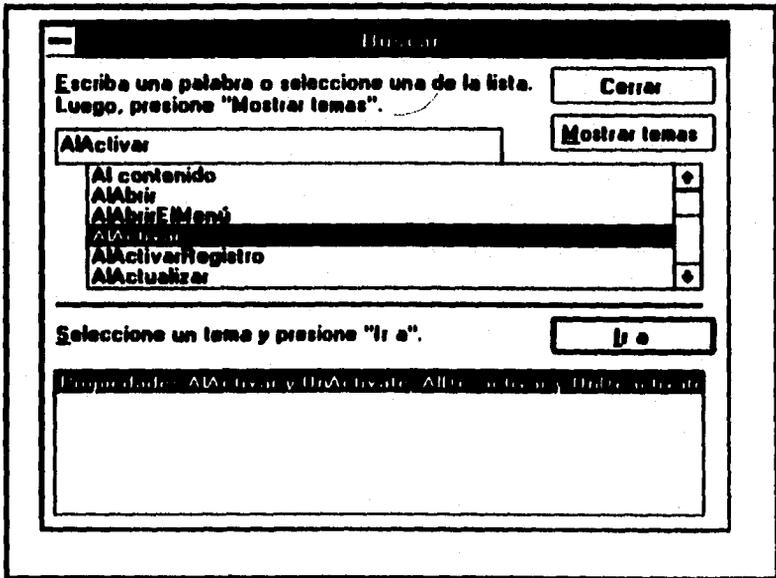


FIG.3.3.1.17 Pantalla de Ayuda de Buscar Tópico



3.3.2. VALIDACIÓN DE CADA UNO DE LOS CAMPOS PARA CADA UNA DE LAS PANTALLAS ANTERIORES.

Para la validación de cada uno de los campos para las pantallas del *Front-End* del sistema de monitoreo de claves de acceso a Internet se utilizó el control *Masked Edit*.

Este control de Visual Basic, provee una máscara en los datos de entrada así como un formato predeterminado de salida en los mismos, además proporciona una señal sobre el tipo de datos que se están introduciendo o accedando.

Este tiene un comportamiento similar a un control de caja de texto pero con otras características, para un manejo óptimo de los datos de entrada y salida con un formato específico. Si no se utiliza una máscara de entrada, el control *Masked Edit* se emplea como una caja estándar de texto, con la excepción de su capacidad de manejo de intercambio dinámico de datos (DDE).

Si se define una máscara de entrada usando la propiedad *Mask*, cada posición de carácter en la máscara será mapeado ya sea a un contenedor del tipo especificado por la máscara o a un carácter literal. Los caracteres literales proporcionan señales visuales acerca de los tipos de datos que se están empleando. Por ejemplo, los paréntesis que rodean el código de área de un número telefónico son literales: (800).

Si se trata de introducir un dato que no sea del formato de la máscara, el control generará un evento de error de validación. La máscara de entrada previene que se introduzcan caracteres inválidos en el control.

Cuando se define una máscara de entrada, el punto de inserción brinca los literales, permitiendo la inserción nada más de los datos necesarios.

Cuando se inserta o se borra un carácter todos los caracteres no literales que se encuentren a la derecha del punto de inserción son movidos como sea necesario. Si este movimiento de



caracteres provoca un error de validación, la inserción o borrado es prevenida, y un evento de error de validación es invocado.

Como podemos ver todas las características mencionadas anteriormente, son útiles en nuestra aplicación para un adecuado manejo de nuestra Base de Datos, ya que se le da un formato de entrada específico a cada uno de los campos de la misma. De esta forma se tiene un control minucioso de la información almacenada en nuestro sistema.

Las máscaras de carácter que se utilizan son las siguientes:

<i>Máscara</i>	<i>Descripción</i>
#	Contiene un dígito: 0-9
?	Contiene una letra: a-z o A-Z

A continuación se presentan algunas de las máscaras que se utilizaron en el sistema:

MÁSCARA	CAMPO
????#####	ID_USUARIO
????????????????	NOMBRE
????????????????	AP_PATERNO
????????????????	AP_MATERNO
## / ## / ##	FECHA DE NAC.
????????????????	LUGAR DE NAC.
????#####	R.F.C.
(###) / ### - ####	TELÉFONO
##### - #	No. CUENTA
## . #	PROMEDIO



3.4 IMPLEMENTACIÓN DEL SOFTWARE PARA LIGAR LA APLICACIÓN CON EL SERVIDOR.

Para poder realizar la rutina que implementará la conexión permanente entre nuestro sistema y el servidor de comunicaciones, se procedió primeramente a verificar la marca de este, para poder conseguir información sobre su manejo de datos y su lenguaje de programación; una vez conseguido y analizado los manuales (auxiliándonos de Internet) necesarios se realizaron análisis simultáneos en relación a como mandarle información en un formato que pudiera comprender y por otra parte programarlo adecuadamente para que por medio de esta combinación de procesos se obtuviera la solución óptima.

Por lo que respecta a la programación del Cisco (servidor de comunicaciones), se configuró con un conjunto de instrucciones propias de la compañía Cisco, las cuales son a nivel de programación administrativa, y que están a disposición de la universidad a través de manuales (también en ayuda en línea en el mismo cisco) e incluso en cd-ROM¹, una vez definido este módulo (ver apéndice k) y ya a sabiendas de que la información se baja al servidor de comunicaciones vía uno de sus puertos series (normalmente el primero) los cuales son del tipo CAB-500DCM o CAB-500DCF (EIA/TIA-232) para un DCE y CAB-500DTM o CAB-500DTF (EIA/TIA-232) para un DTE y en formato texto (Ascii). El equipo necesario para llevar a cabo esta transferencia es el siguiente:

Hardware (mínimo)

Pc con un microprocesador Intel 80386 sx o superior

Dos puertos seriales o un puerto serial y un puerto para mouse

4 Mb de memoria RAM como mínimo.

Mouse Microsoft o compatible con Windows

Software

MS DOS 3.1 o posterior; se recomienda versión 5.0 o posterior

¹ Para ver una lista más completa, referirse al apéndice j.



MS Windows 3.1

Posteriormente pasamos al desarrollo del módulo que interconectará al cisco 500-cs con nuestra aplicación y a través del cual se le enviarán los datos a validar al usuario, para lo cual se diseñaron bajo la arquitectura cliente/servidor los servicios soportados por Internet, los cuales para fines de desarrollo fueron un mail, un ftp y un news mail, los tres siendo diseñados en Visual Basic ver 3.0, esto debido a que cuenta con la capacidad de explotar el ambiente windows, además de que ofrece un socket de conexión para internet con los principales servicios, este conector es el *winsocket* de Microsoft.

Ya contando con este módulo continuaremos explicando las herramientas para administrar el trabajo de Internet.

El software ruteador del Cisco, combinado con una línea completa de Hardware para redes ha hecho que el Cisco sea un sistema líder de multimedia, así como productos de trabajo en red con multiprotocolo.

Cisco ha aplicado su conocimiento y las aplicaciones obtenidas de su experiencia para el diseño de las herramientas administradoras de rutas únicas. Por ejemplo, cada ruteador que Cisco construye incorpora características de manejadores preconstruídos que forman parte del sistema Operativo *Internetwork (IOS)*. Un *framework* provee una sola plataforma integrada para todos los usuarios de red y sus aplicaciones. Además Cisco ofrece a los administradores un conjunto de aplicaciones, que son poderosas y fáciles de usar, para el trabajo de Internet.

Este incluye al constructor de configuraciones Cisco, un producto para simplificar las configuraciones iniciales de red, y *CiscoWorks* que es una serie de herramientas que provee a los usuarios con capacidades de administración centralizada.



CONSTRUCTOR DE CONFIGURACIÓN CISCO

- Configuraciones sincronizadas para grupos de ruteadores.

CARACTERÍSTICAS DE ADMINISTRACIÓN IOS CISCOWORKS

- Monitor de interface: Estado del ruteador, utilización del protocolo y la misma interface.
- Herramienta de direcciones: Visualización y análisis gráfico de rutas.
- Series de Administración: Análisis de patrones de tráfico de red.
- Administrador de archivos de contexto: Administrador de configuración y verificación.
- Comandos de visualización: Estadísticas gráficas de ruta.
- Administrador de acceso: Base de Datos histórica para eventos SNMP y SYSLOG.
- Administrador de Ambiente: Visualización gráfica de temperatura y voltaje.
- Facilidad de comandos globales: Herramienta de actualización automática de configuraciones, para la administración central.
- Administrador de autoinstalación: Herramienta para la instalación automática de rutas, para el control central.
- Servidor TACACS: Control de acceso a red mediante el uso de *USERNAME* y *PASSWORD*.
- Administrador de seguridad: 5 niveles de acceso dentro de los dominios administrativos para asegurar la administración de los datos en la red.
- Administrador de Software: Administrador de sistema y microcódigo.

El administrador CISCO de trabajo en Internet posee la característica de tener un ambiente de rutas tipo "*plug and play*", donde la instalación, configuración y administración de los ruteadores es bastante simple.

Complementándose con una inteligencia nativa del ruteador, las aplicaciones *CISCOWORKS* aumentan la productividad de un control central.



3.4.1 DESARROLLO DE LA RUTINA PARA VERIFICACIÓN E INHABILITACIÓN POSTERIOR DE LA CLAVE

Para llevar a cabo esta rutina, se tuvieron que tomar en cuenta diferentes factores que determinaron los criterios para permitir el acceso a la cuenta en Internet.

Primeramente, el usuario introduce su *login* y *password* para su acceso al servidor de comunicaciones. Una vez que el servidor se encarga de validar el *login* o pasárselo al sistema para que el lo valide, con su respectivo *password*, el sistema genera internamente un número aleatorio entre 1 y 15. Si este número corresponde al de la línea en la cual el usuario estableció su comunicación, entonces se ejecuta un procedimiento que toma de la memoria del cisco el *login* del usuario y realiza una búsqueda en la base de datos para verificar que esa clave exista. En el caso de las claves privilegiadas en las que por motivos ajenos a este trabajo no se les aplica el sistema, el cisco maneja una tabla de usuarios de este tipo lo que hace posible dichos usuarios no pasen por el sistema de monitoreo, lo que concluye en un acceso limpio² a Internet. Mientras que en el caso de los demás usuarios, si fueron seleccionados para ser monitoreados, se les envía una orden de activación de la forma para su validación. Esta forma consiste en una serie de preguntas que varía desde una hasta 3 preguntas, también de forma aleatoria, las cuales el usuario debe contestar de forma correcta para poder permitirle el acceso a Internet.

Estas preguntas son relacionadas a sus datos personales laborales o escolares. Las respuestas son verificadas de acuerdo al contenido del registro del usuario en cuestión en la base de datos. Los datos son tomados de las formas de suscripción que fueron llenadas por los usuarios al momento de adquirir su cuenta.

De acuerdo a las respuestas que de este cuestionario se obtengan el acceso a Internet es permitido o denegado.

² Acceso limpio es el que presentan las claves, a las que no se les aplica la verificación y el monitoreo.



El diagrama esquemático de la rutina para verificación e inhabilitación de la clave se muestra en la figura 3.4.1.1.

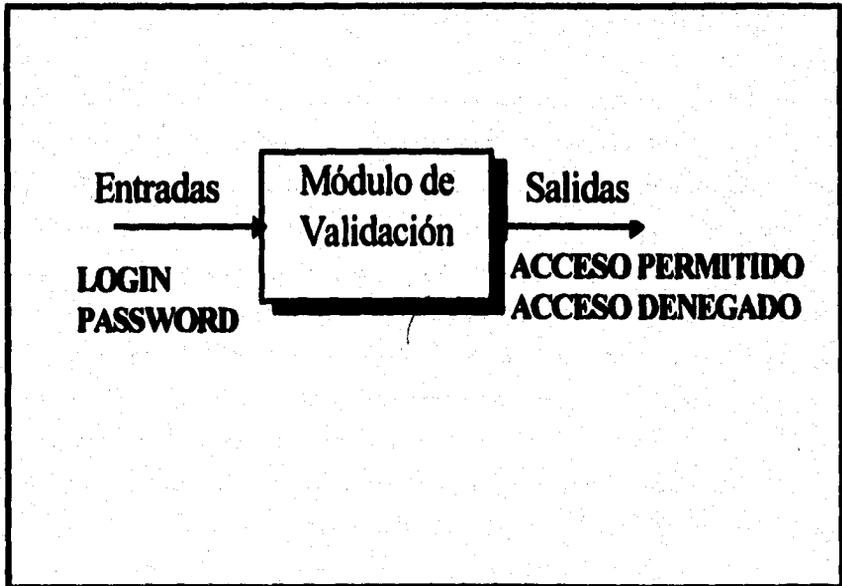


Figura 3.4.1.1 Módulo de la rutina para verificación e inhabilitación de la clave

A continuación se muestra parte de la rutina que realiza este proceso. Fig 3.4.1.2



```

Sub Text1_KeyPress (KeyAscii As Integer)
  Dim MSG As String, NumCars, var As Integer

  If ((KeyAscii >= 48) And (KeyAscii <= 57)) Or ((KeyAscii >= 64) And (KeyAscii
  If KeyAscii = 13 Then

    If Len(text1.Text) = 9 Then
      CommX.Output = buffersal
      buffersal = Mid$(buffersal, NumCars + 1)
      text1.Text = ""
      buffersal = ""
      ciclo = 1
      text2.SetFocus
    Else
      MsgBox " Clave incorrecta"
      buffersal = ""
      text1.Text = ""
      text3.SetFocus
    End If
  Else
    buffersal = buffersal + Chr$(KeyAscii)
  End If
Else
  text1.Text = ""
  buffersal = ""

  MsgBox "Caracter Inválido"

```

Fig. 3.4.1.2 Impresión parcial del código

Como se observa en el código, este deberá residir en el router, esto con la finalidad de tratar de aprovechar sus capacidades programables y que solo reciba una orden de conexión desconexión proveniente del sistema de monitoreo¹. El proceso de conexión desconexión es el que recibe la orden del módulo de validación una vez que este concluya con el proceso de certificación de los datos del usuario.

Referente a la inhabilitación de la clave se creó una tabla especial en la base de datos llamada BITÁCORA la cual liga un solo registro de datos a cada usuario, los datos que contiene son en relación a los intentos que los usuarios fallan al contestar la forma de preguntas que se les visualiza inmediatamente al serles validados correctamente su *login* y *password* del cisco, por cada intento fallido en la validación el sistema va asignado al registro correspondiente de ese usuario la fecha y hora de su intento, por lo tanto el sistema al detectar la entrada de cada usuario

¹ Par amayor información, referirse al apéndice j.



(*login y password* en el cisco) hará una petición a la base de datos y en especial a la bitácora para determinar si el campo que almacena la fecha o la hora del intento tres son o no cadenas vacías, de no serlo entonces el sistema entenderá que el usuario ya no tiene derecho a conexión mandándole un mensaje de que pase a la DGSCA a aclarar su caso, en el caso contrario , el sistema deberá buscar en los campos donde se encuentra la fecha y/o la hora del intento dos repitiendo el mismo ciclo explicado anteriormente, para el primer intento.y como ya se menciono el sistema sólo soportará tres intentos de validación antes de proceder a la inhabilitación permanente de la clave.

Como se mencionó en el punto anterior el problema de entradas simultáneas se resuelve con programación directa del servidor de comunicaciones, para lo cual primeramente se implementa todo el código fuente a ejecutar en código ASCII, esto es en formato texto sin caracteres especiales (archivo plano) para posteriormente hacer una sesión remota al cisco desde la pc en la que estemos editando el programa y una vez conectados se procede a bajar la información al troutter.



3.5 INTEGRACIÓN DEL SISTEMA

En este punto ya se implementó la Base de Datos en Microsoft Access. Por separado tenemos los archivos .bnp que corresponden a las fotografías digitalizadas.

Al realizar la integración de los datos tuvimos que incorporar utilerías adicionales para poder acceder la Base de Datos desarrollada en Microsoft Access 2.0 con el *Front-End Visual Basic. 3.0* ya que estas versiones son incompatibles; Las utilerías que nos permiten la comunicación se llaman "*Compatibility Layer*".

Nos basamos en una integración ascendente para realizar la integración del sistema. Esta consiste en pruebas de unidad, seguidas por pruebas de subsistemas y luego por pruebas del sistema completo.

Las pruebas de unidad tienen como objetivo descubrir errores en los módulos individuales del sistema. Estos módulos se prueban aislados unos de otros en un ambiente artificial formado por los programas conductores y los datos necesarios para ejecutar los módulos. Las pruebas de unidad deben ser tan exhaustivas como sea posible, para garantizar que sea probado cada caso representativo empleado en cada módulo. Dichas pruebas son más fáciles, si las estructuras se componen de módulos pequeños y débilmente acoplados.

En el caso de las pruebas a los subsistemas, su propósito principal es verificar la operación de las interfaces entre los módulos en el subsistema. Se deben probar tanto las interfaces de control como las de datos.

Las pruebas del sistema se relacionan con las interfaces, la lógica de decisión, el flujo de control, los procedimientos de recuperación, la eficiencia global, la capacidad y las características de tiempo del sistema integrado en su totalidad.



El objetivo a lograr, siguiendo esta secuencia de pruebas, es dar un seguimiento, orden y por tanto estabilidad al proceso de desarrollo del software, permitiéndonos obtener un producto de mejor calidad. Asimismo, para lograr lo anterior, nos auxiliamos de las denominadas pruebas de demostración que tienen como regla los siguientes tres puntos:

1. Objetivo práctico del sistema
2. Metodología empleada en el sistema
3. Comprobación de la metodología

Las pruebas ascendentes suelen revelar que es más fácil crear casos de prueba y observar las entradas de está. Sin embargo, tiene el inconveniente de que si existen errores de diseño en los módulos de alto nivel se detectan hasta una etapa avanzada de la prueba del sistema.

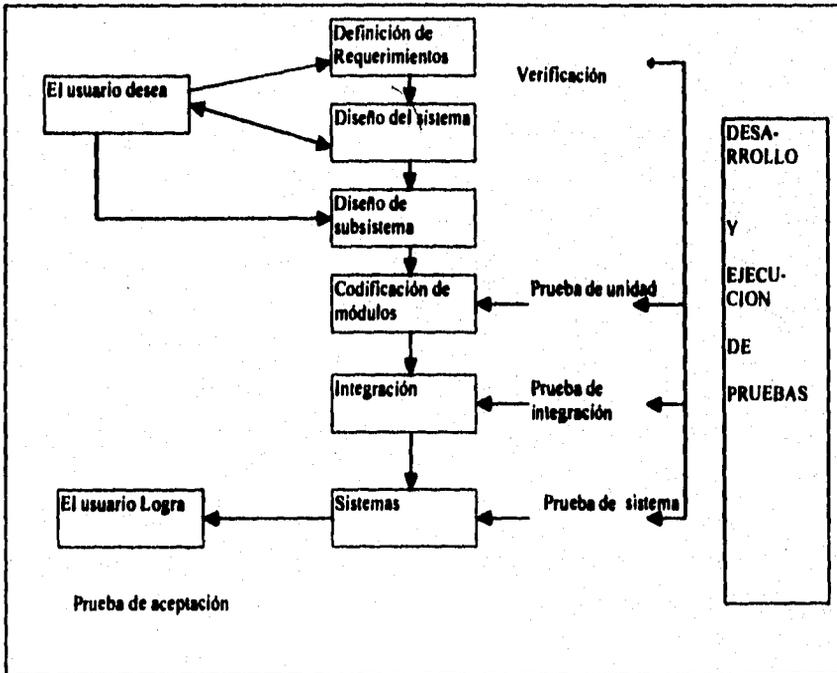


FIG. 3.5.1. Principales Operaciones de Prueba que se deben durante el desarrollo



1.- Integración y pruebas a la Base de Datos en ACCESS

Consideremos necesario crear un sistema que nos permitiera manipular fácilmente los datos que se iban a utilizar en el proyecto, que fuera amigable para aquellas personas que tengan que actualizar la Base de Datos y las cuales no requieran tener conocimiento del manejador de bases de datos Microsoft Access para poder realizar los cambios.

Una vez implementado el sistema, se procedió a cargar la información de diferentes datos, con el cual es más sencillo el detectar fallas en la programación y corregir las mismas. Asimismo, nos auxiliamos de personas que desconocían el sistema para comprobar la facilidad de uso del mismo y su sencilla manipulación.

2.- Integración y pruebas al Front-End

Una vez que se cuenta con la Base de Datos y las utilerías necesarias para leer la información de la Base de Datos desde Visual Basic, se procedió a conjuntar el Front-End.

Una vez probado los módulos se anexaron uno a uno en el proyecto final. Cada vez que un nuevo módulo se incorporaba se ejecutaba la aplicación para verificar su correcto funcionamiento conviviendo con los otros módulos. Si se detectaba un fallo en el funcionamiento del sistema se depuraba el código y sólo cuando éste funcionaba sin problemas se procedía a agregar uno nuevo.

3.- Integración y Pruebas Finales

Al conjuntar los módulos del *Front-End* se obtuvo la integración de todos los elementos del sistema. Para cumplir el objetivo de que el sistema fuera amigable al usuario (administrador), durante el desarrollo nos auxiliamos de diferentes usuarios para observar como se comportaban ante la aplicación y así saber si se cumplía con el objetivo del sistema, de no ser así se realizaron los cambios necesarios.



Una vez que contamos con la aplicación final se implementaron una serie de pruebas del sistema ya integrado con lo cual se dio por concluido el proceso de integración ascendente.

Finalmente para asegurarnos del buen funcionamiento del sistema, se efectuaron pruebas de tipo manuales y de adhoc (al azar).

Prueba Manual

Esta prueba involucra un conjunto de casos de pruebas los cuales, al ser ejecutados, son teclados en la computadora en forma manual.

Ventajas

Un sistema de ejecución automatizada es esencial para asegurar resultados confiables. Como los casos de pruebas que pueden ser construidas y almacenadas para pruebas posteriores de regresión.

Desventajas

Este tipo de pruebas se implemento al momento de capturar la información en la Base de Datos y en probar como se comportaba el sistema al establecer diferentes datos, con lo que se comprobaba que los datos requeridos eran los proporcionados por el usuario, y se validarán conforme a la reglas preestablecidas.

Pruebas AD HOC

Esto involucra al programador o a un probador, intentando hacer fallar el software bajo prueba. Puede ser en la unidad o a nivel de módulo, pero es casi siempre empleado en las pruebas a nivel de sistema.



Ventajas

El uso de captura/repetición de herramientas puede ayudar a superar algunos de los defectos inherentes del método al introducir una medida de repetición

Desventajas

El proceso entero es unisistemático.

Si se encuentra una sospecha de error, la naturaleza al azar de las entradas, la falta de documentación y el posible uso de caracteres no impresos significan que el efecto tal vez no puede ser duplicado.



3.5.1 PRUEBAS DE VERIFICACIÓN DE CADA UNO DE LOS BLOQUES

Prueba de caja blanca

Este método es empleado en el diseño de casos de prueba que usa la estructura de control del diseño de procedimientos para derivar los casos de prueba. Mediante esta prueba se puede obtener casos de prueba con los que se garantice que: (1) por lo menos una vez se han empleado todos los *caminos independientes* de cada módulo; (2) que se ponen en práctica todas las decisiones lógicas en sus dos posibilidades *verdadera y falsa*; (3) se ejecuten todos los bucles en sus límites propuestos y con sus límites de operación y (4) se ejerciten las estructuras internas de datos para que con ello sea segura su validez.

Esto se realiza debido a las siguientes razones en el desarrollo de software:

- *Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa*, lo que nos quiere decir es que los errores tienden a presentarse en los casos que se encuentren fuera de lo normal, el procedimiento habitual tiende a ser mucho más comprensible, mientras que el procesamiento de casos especiales por lo general tiende a caer en el caos.

- *A menudo se cree que un camino lógico tiene pocas posibilidades de ejecutarse cuando en realidad se puede ejecutar en una forma regular*, esto no siempre implica que el flujo lógico de un programa sea intuitivo, lo que implica que las suposiciones intuitivas de flujos de control y los datos nos pueden ocasionar errores de diseño que sólo las podemos detectar cuando se emplean las pruebas del camino, que de cualquier otra forma serían pasadas por alto y no se podrían corregir a tiempo en nuestro software.



- *Los errores tipográficos son aleatorios*, muchas veces cuando se traduce un programa diseñado a código fuente es muy probable que se presenten errores de escritura, muchos serán descubiertos por los mecanismos de comprobación de sintaxis, pero muchos otros no serán descubiertos, por lo que es igual de probable que se presente un error en un camino lógico poco empleado a uno dentro de un camino principal.

Cada una de estas razones presentadas nos da un argumento para emplear las pruebas de caja blanca, con lo que se descubren de una manera mucho mas sencilla.

La *prueba del camino principal* es un método que se emplea para poder obtener una medida de la complejidad lógica en un diseño de procedimientos de nuestro sistema, con lo que podemos emplear esa medida como una guía para la creación de un *conjunto básico* de caminos de ejecución. Esto nos permite garantizar que los casos de prueba que sean derivados del conjunto básico por lo menos serán ejecutados una vez cada sentencia del programa.

Prueba del camino básico.

La prueba del *camino básico* es una técnica de prueba de la caja blanca propuesta originalmente por Tom McCabe. El método del camino básico permite, al diseñador de casos de prueba, obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un *conjunto básico* de caminos de ejecución. Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Prueba de la caja (Grafos causa-efecto)

Los métodos de prueba de caja negra se centran en los requisitos funcionales del software, por lo que permite al desarrollador obtener conjuntos de condiciones de entrada que sean empleados para utilizar completamente todos los requisitos funcionales de un programa.



La prueba de la caja negra *no* es una alternativa para las pruebas de la caja blanca, mas bien es un enfoque complementario de dicha prueba que intenta descubrir errores que las pruebas de la caja blanca no puede detectar y que por lo mismo son distintos tipos de errores.

Los métodos de la caja negra intentan encontrar errores de las siguientes categorías: (1) funciones incorrectas o ausentes; (2) errores de interfaz; (3) errores de estructuras de datos o en acceso a bases de datos externas; (4) errores de rendimiento y (5) errores de inicialización y de terminación.

La diferencia de la prueba de la caja blanca y la de la caja negra radica en que esta última tiende a aplicarse durante posteriores fases de prueba, ya que intencionalmente ignora la estructura de control y centra su atención fundamentalmente en el campo de la información, las pruebas empleadas están diseñadas para responder de una manera pronta y confiable a las siguientes preguntas fundamentales:

- ¿Cómo se prueba la validez funcional?
- ¿Qué clases de entradas compondrán unos buenos casos de prueba?
- ¿Es el sistema particularmente sensible a ciertos valores de entrada?
- ¿De qué forma están aislados los límites de una clase de datos?
- ¿Qué volúmenes y niveles de datos tolerará el sistema?
- ¿Qué efectos sobre la operación del sistema tendrán combinaciones específicas de datos?

Mediante el uso de los métodos de prueba de la caja negra se obtiene un conjunto de casos de prueba que satisfacen los criterios siguientes: (1) casos de prueba que reducen el número de casos de prueba que se deben diseñar para poder obtener una prueba *razonable*; (2) casos de prueba que nos proporcionan información sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba que se encuentre disponible.



Grafos de Causa-Efecto

Los *grafos de causa-efecto* son una técnica de diseño de casos de prueba que proporciona una concisa representación de las condiciones lógicas y sus correspondientes acciones. La técnica sigue cuatro pasos principales:

1. Se listan para un módulo las *causas* (condiciones de entrada) y los *efectos* (acciones), asignando un identificador a cada uno de ellos.
2. Se desarrolla un grafo de causa-efecto.
3. Se convierte el grafo en una tabla de decisión.
4. Se convierten las reglas de la tabla de decisión a casos de prueba.

En la siguiente figura (fig 3.5.1.1) se muestra una versión simplificada de la simbología de los grafos de causa-efecto. La columna a la izquierda de la figura ilustra varias relaciones lógicas entre causas c_i y efectos e_j . La notación de líneas punteadas de la columna derecha indica relaciones potencialmente restrictivas que se pueden aplicar tanto a las causas como a los efectos

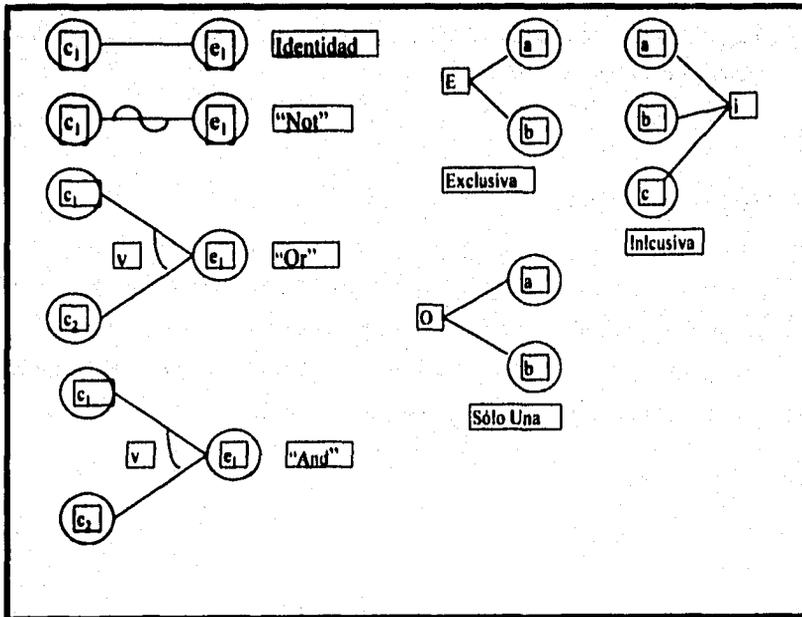


FIG: 3.5.1.1. Simbología de Grafos Causa-Efecto



En la fig 3.5.1.2. se muestra un grafo causa-efecto, en el extremo izquierdo se muestran las causas 1,2,3 y 4, y los efectos finales 101, 102 y 103, en el extremo derecho. En la parte central del grafo aparecen identificadas causas secundarias (p. ej.: causas 11, 12, 13, 14, etc.). A partir del grafo causa-efecto se puede desarrollar una Tabla de decisión. Los datos de los casos de prueba se seleccionan de tal manera que se ejercite cada *regla* de la tabla. Obviamente, si se ha usado una tabla de decisión como herramienta de diseño, los grafos de causa-efecto ya no son necesarios.

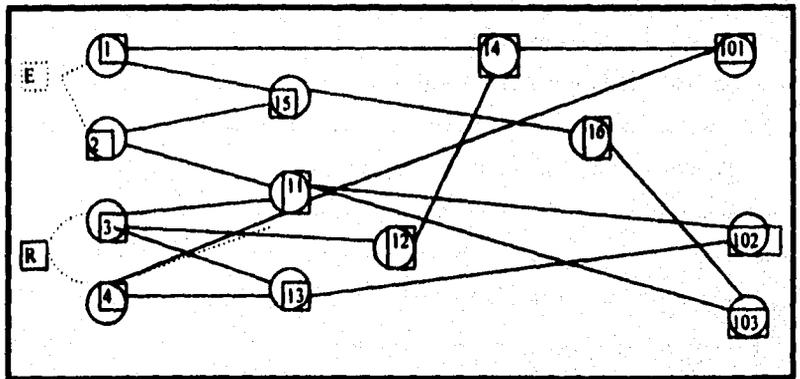


FIG..3.5.1.2. Grafo Causa-Efecto

Prueba del software

La prueba del 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.

Estas pruebas están diseñadas para ser una prueba final del sistema ya terminado y permite descubrir errores en el mismo, las pruebas son las siguientes:



Pruebas de integración

Esta prueba se dirige a todos los aspectos asociados con el doble problema de verificación y de construcción del programa. Durante la integración, las técnicas que más son empleadas son las de diseño de casos de prueba de la caja negra, aunque también se pueden emplear algunas pocas pruebas de la caja blanca con el objetivo de asegurar que se cumplen con los principales caminos de control, después de que el software se a ensamblado se pasa al siguiente conjunto de pruebas que son las de alto nivel.

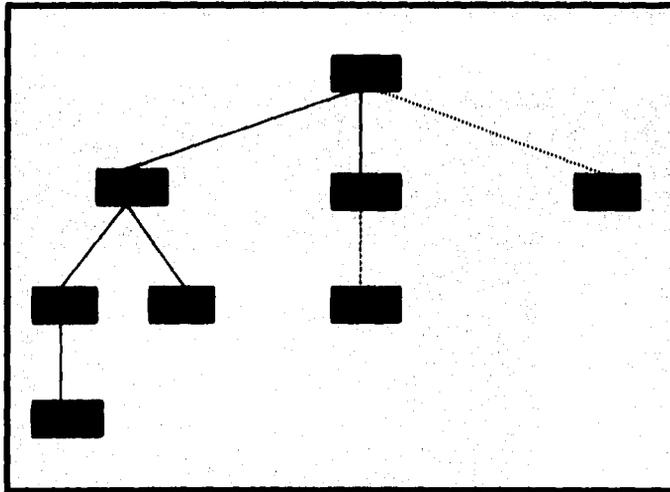


FIG. 3.5.1.4.-Integración descendente.

Pruebas de validación

Dado que se deben de comprobar los criterios de validación, en esta prueba se obtiene una comprobación final de que el software satisface todos los requerimientos funcionales, de comportamiento y de rendimiento; durante esta prueba se utilizan exclusivamente métodos de la caja negra. Se emplean dos métodos importantes que son las pruebas *alfa* y *beta*.



comportamiento y de rendimiento; durante esta prueba se utilizan exclusivamente métodos de la caja negra. Se emplean dos métodos importantes que son las pruebas *alfa* y *beta*.

Prueba Alfa y Beta

Es virtualmente imposible que el encargado del desarrollo del software pueda prever cómo un cliente usará *realmente* un programa. Se pueden mal interpretar las instrucciones de un comando, se pueden emplear extrañas combinaciones de datos y los resultados pueden ser muy claros y precisos para el desarrollador pero no así para el usuario normal.

Cuando se desarrolla un software a medida para un cliente, se lleva a cabo una serie de pruebas de aceptación para permitir que el cliente valide o rechace todos los requisitos. Llevado a cabo por el usuario final en lugar del equipo de desarrollo, una prueba de aceptación puede ser desde un informal "paso de prueba" hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, la prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores que pueden ir degradando el sistema.

Si el paquete se desarrolla como un producto que será empleado por muchos usuarios, no resulta práctico el utilizar pruebas de aceptación formales para cada uno de los usuarios. La mayoría de los desarrolladores de software llevan a cabo un proceso denominado *prueba alfa* y *beta* para poder encontrar los errores que parezca que sólo el usuario final puede descubrir.

La *prueba alfa* es llevada a cabo por un cliente en el lugar de desarrollo del software. Se utiliza el paquete de una forma normal, con el encargado del desarrollo del software presente, observando que es lo que el cliente esta realizando, registrando errores y problemas de uso. Las pruebas alfa se llevan acabo en un entorno controlado.

La *prueba beta* se lleva acabo en uno o más lugares y es realizada por los mismos usuarios finales del software. A diferencia de la prueba alfa, el programador no se encuentra presente. Así, la prueba beta es una aplicación "en vivo" del software en un entorno que no es



controlado por el equipo de desarrollo. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la duración de la prueba e informa en tiempos regulares a los desarrolladores del software, esto da como resultado que el equipo encargado del desarrollo del software puede implementar modificaciones y realizar una versión "corregida" del software para toda la base de clientes.

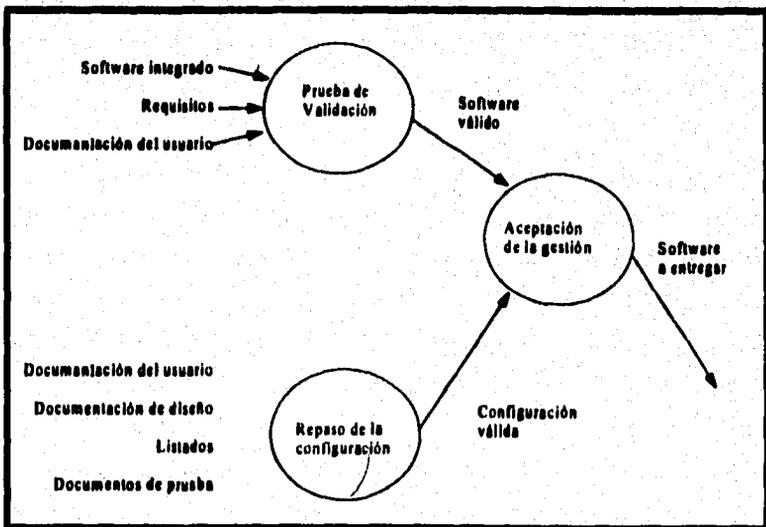
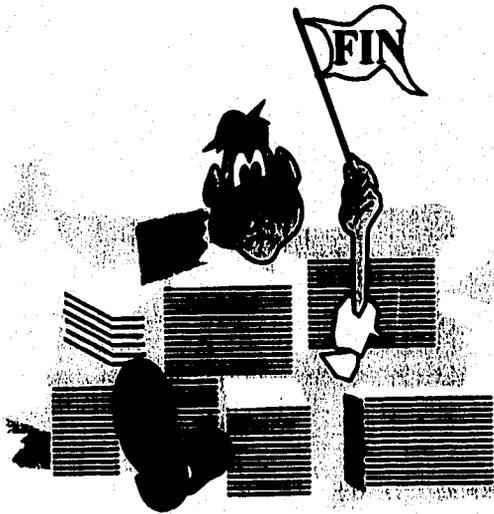


Fig. 3.S.1.S Repaso de la configuración.

Prueba del sistema

En esta última prueba del software se debe de combinar con otros elementos del sistema (p.ej.: hardware, usuarios, Bases de Datos, etc.), en esta prueba final se verifica que cada elemento encaja de forma adecuada y según lo planeado y que con esto se alcanza la funcionalidad requerida y que el rendimiento total del sistema es el deseado.

CONCLUSIONES





CONCLUSIONES

Nuestra tesis conjunta una gran variedad de conocimientos que forman parte importante de la informática contemporánea, tal es el caso de los lenguajes visuales que han adquirido gran importancia en el desarrollo de aplicaciones en general y específicamente en aquellas en que la comunicación entre el usuario y la aplicación se debe realizar en forma interactiva, lo cual implica que la interface sea lo mas apegado a sus requerimientos. Uno de los lenguajes visuales mas representativos es *Visual Basic*.

Visual Basic es uno de los primeros lenguajes de programación que admite el concepto de programación orientada a eventos; un estilo de programación adaptada a las interfaces gráficas del usuario.

Por esta y por muchas otras razones fue de gran utilidad en el desarrollo de este proyecto.

Consideramos que el desarrollo de las bases de datos es una de las actividades más importantes en el campo de la informática actualmente. Los datos se consideran como un recurso vital para la empresa que inerece ser debidamente organizada con el fin de maximizar su valía.

Uno de los objetivos de la automatización del procesamiento de datos de nuestra aplicación involucró utilizar potentes herramientas de manejo de datos que permitan el control absoluto, confiabilidad e integridad de los mismos.

Parte del código empleado en el desarrollo de este tema fue ligado de otras aplicaciones generadas en *Visual Basic* versión 3.0 y el manejador de Bases de Datos *Access*, debido a que la programación en ese lenguaje es fácil, el código generado puede ser reutilizable proporcionando flexibilidad y un gran potencial.

Debido al crecimiento diáfano e inusitado y al explosivo desarrollo de la tecnología en los aspectos de redes y comunicaciones, así como el uso cada día mas frecuente e importante de la



red de redes Internet por parte de toda clase de usuarios que requieren de servicios de información tan oportunos como es el servicio prestado por RedUNAM, así como su aprovechamiento óptimo de los mismos, este trabajo permitió la oportunidad de brindar este tipo de servicios bajo un sistema de control de datos de la misma red, validando datos que los mismos usuarios proporcionan, con lo que el sistema se vuelve eficiente.

Durante la elaboración de este proyecto, se ratificó la gran importancia que ha adquirido el uso de la Red de Redes INTERNET, ya que gracias a ella se puede obtener una gran cantidad de información de diferentes tópicos. Nos enteramos de las existencia de variadas herramientas las cuales podemos explotar, permitiéndonos con ello descubrir nuevos elementos y métodos para realizar mejoras, correcciones y adaptaciones a los sistemas de información.

De este trabajo se puede concluir que sí se cumplió el objetivo del mismo, que era obtener un sistema funcional para el control de claves de acceso a Internet vía modem a través de RedUNAM.

Cabe mencionar que el sistema cumplió con otro objetivo: ser un sistema de monitoreo de las mismas claves, proporcionándole al administrador, una vista real del funcionamiento y estado del mismo. Este aspecto es fundamental para la UNAM, ya que involucra aspectos financieros y contables.

La elaboración de la tesis nos dió mas experiencia en el desarrollo de trabajos de investigación, obviamente mas organizados y estructurados; además nos permitió ampliar nuestros conocimientos en temas tales como Bases de Datos, redes de Computadoras y programación orientada a eventos.

La aplicación resultó ampliamente eficiente y amigable.

Al concluir este trabajo se resuelve la problemática para el servicio proporcionado por la UNAM para el acceso a Internet. Con el presente sistema se mejoró el control de las claves de



acceso a Internet, además que para su implementación se aprovecharon medios con los que ya se contaba y que no se utilizaban adecuadamente.

Una vez más, al igual que otros compañeros egresados de la **Facultad de Ingeniería** y del programa de becas de la **Dirección de Cómputo para la Administración Académica** aplicamos los conocimientos que nos transmitieron para cristalizarlos en un proyecto de relevancia para engrandecer a nuestra *Alma Mater*, la **Universidad Nacional Autónoma de México** y a nuestro país.

MANUAL DE USUARIO

COMO REALIZO UNA
CONSULTA?





ACERCA DEL MANUAL

Este manual fue creado para que usted tenga una guía acerca de la instalación y uso del sistema LIOS. Aquí encontrará las diferentes opciones que proporciona el sistema con una explicación de su uso.

Este manual fue creado en el mes de julio de 1996, por los desarrolladores del software:

- Díaz Zúñiga Liliana
- Guzmán Bazán Xóchitl Verónica
- Loria Castellanos Oscar
- Sánchez Cruz Luis Gonzalo
- Vega Hernández Marcela

REQUERIMIENTOS DE SOFTWARE Y HARDWARE

• Computadora Personal

IBM 486DX (o superior) o compatible.

8Mb de RAM.

Monitor SVGA color

Disco Duro de 500 Mb.

Velocidad : 100Mhz.

Impresora.

Escaner, se recomienda uno manual a color.



• Paquetería.

Sistema Operativo 5.0 o posterior (se recomienda 6.2).

Windows 3.1 o posterior

Manejador de bases de datos Acces ver. 2.0

Software de monitoreo y verificación de claves LIOS.

Software para el escáner seleccionado (que soporte preferentemente los formatos bmp,pcx,pic.).

INSTALACIÓN.

Usted debe introducir el disco de instalación numerado como 1 y desde el administrador de programas de windows en el menú de archivo seleccione la opción de ejecutar, al desplegarle la ventana de datos seleccione el botón de examinar y con la ayuda de mouse seleccione la unidad a:, haga click en el botón de aceptar y al cerrarse presione el botón ejecutar.

Después de unos segundos usted visualizará la pantalla que se muestra en la figura MU.1, debiendo seguir las instrucciones que se mostrarán en pantalla.

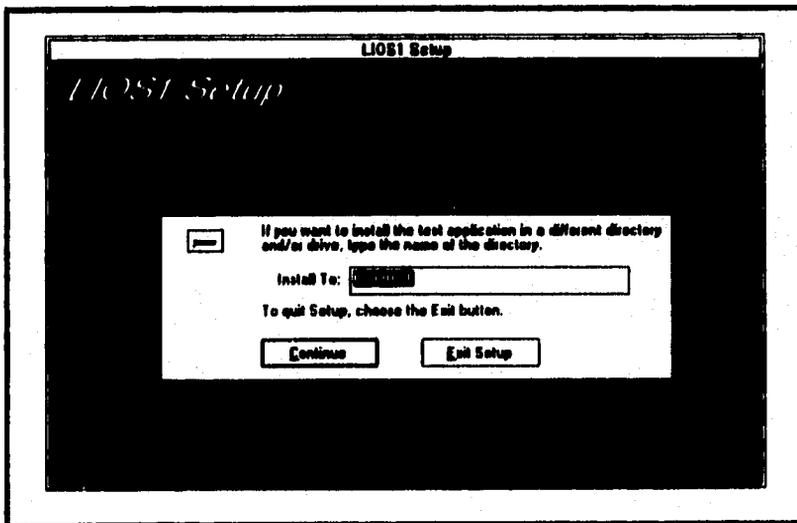


Figura MU.1 Pantalla de Instalación del Sistema LIOS



FUNCIONAMIENTO DEL SISTEMA.

Para ejecutar el sistema LIOS basta con dar dos clicks en el icono asociado a la aplicación. Ver figura MU.2.

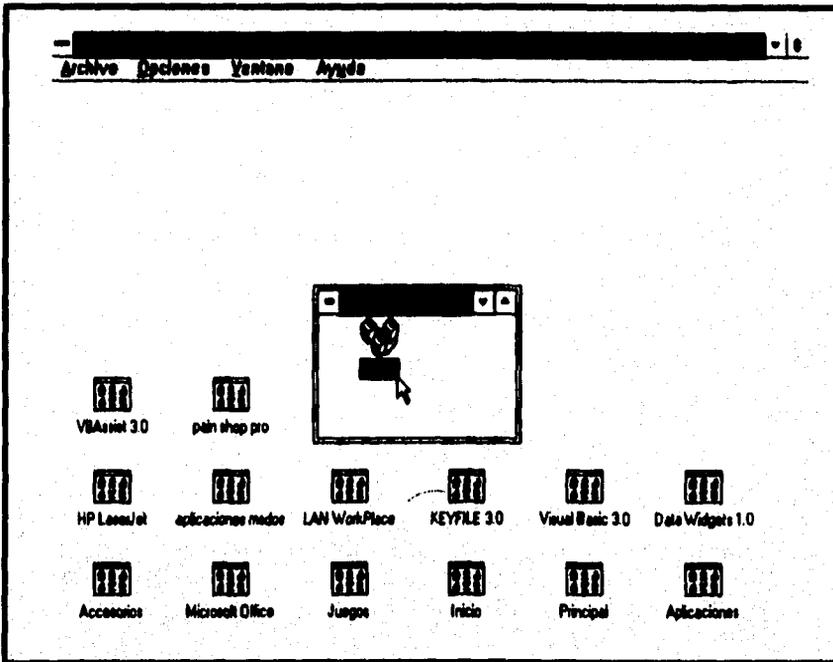


Figura MU.2 La aplicación en el Administrador de Programas

La primer pantalla que aparece en su monitor es la de presentación del sistema, haga click en el botón de continuar, es en este momento que aparecerá la ventana de seguridad del sistema en la cual tendrá que validar su *login* y *password*. Ver figura MU.3.

Existen tres usuarios distintos que pueden accesar el sistema, cada uno de ellos posee distintos privilegios.



🔑 **Módulo de Acceso** 🔑

Escribe su Password

Figura MU.3 Forma de Verificación de Entrada

A continuación se muestran los logins disponibles del sistema en orden de seguridad y una breve descripción de lo que el sistema pondrá a disposición de este tipo de usuario:



NIVEL DE PRIVILEGIOS	
SUPERA	EJECUTA TODAS LAS OPCIONES DENTRO DEL SISTEMA, ALTAS, BAJAS, CAMBIOS, REPORTES, ESTADÍSTICAS, ETC. ADEMÁS DE PODER DESACTIVAR POR COMPLETO DEL SISTEMA.
USUARIOA	LAS TAREAS QUE PUEDE REALIZAR ESTE TIPO DE USUARIO SON SIMILARES A LAS QUE EL SUPERADMINISTRADOR REALIZA A DIFERENCIA DE QUE ÉSTE NO PUEDE DESACTIVAR EL SISTEMA, NI TAMPOCO PUEDE CONSULTAR LA BITÁCORA DE LOS ADMINISTRADORES DEL SISTEMA.
USUARIOB	SOLO PUEDE REALIZAR CONSULTAS Y ESTADÍSTICAS

De acuerdo al privilegio de acceso que se tenga, se presentará la ventana principal del sistema, la cual dependiendo de éste privilegio, habilitará los menús que están disponibles en el sistema LIOS.

Una vez validado el login y password correspondiente, se accesa a la pantalla general del sistema. Ver figura MU.4.

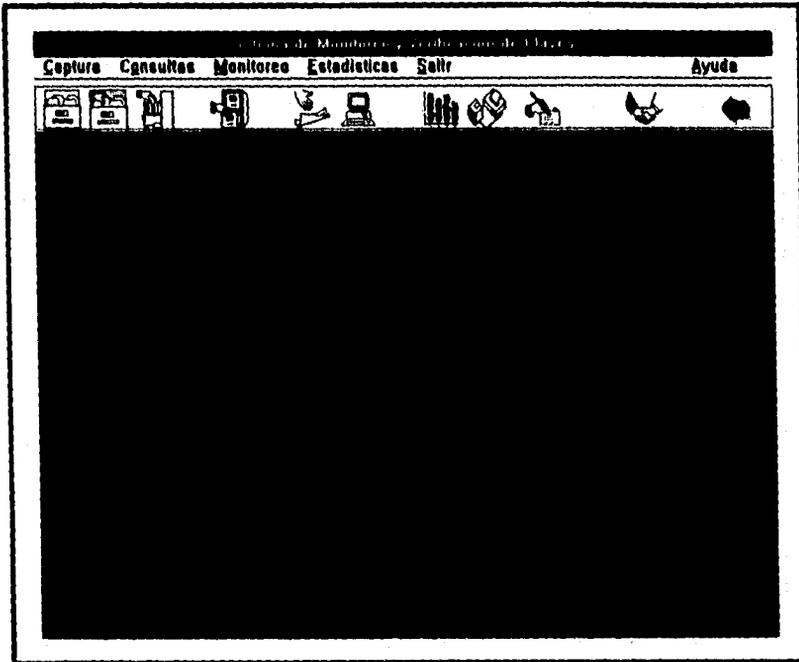


Figura MU.4 Pantalla General del Sistema

De acuerdo a la figura MU.4, podemos observar que se dispone de una barra de opciones en la parte superior de la pantalla. Esta barra de opciones contiene las siguientes etiquetas :

- Captura
- Consultas
- Monitoreo
- Estadísticas
- Salir
- Ayuda

Al pulsar algunas opciones contenidas en la barra de menú, se tendrá a su vez, un menú desplegable asociado, con otro conjunto de opciones a elegir.



En la figura MU.5 se muestra en forma esquemática esta jerarquía de opciones.

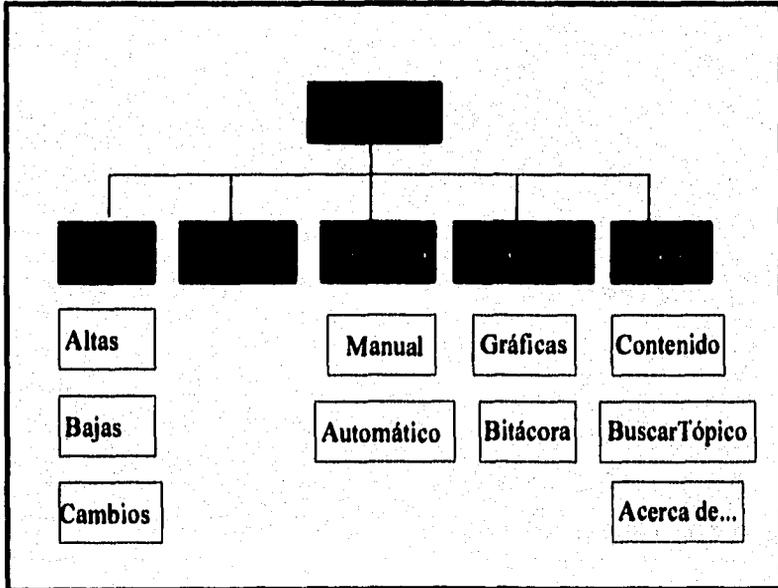


Figura MU.5 Jerarquía de opciones en el conjunto de Menu del Sistema

Bajo esta barra de menú, se distingue además, una barra de iconos. En esta barra de iconos al pulsar alguno de ellos, se dispone de forma directa a algunas de las acciones que se realizan con mayor frecuencia y están relacionadas directamente con las opciones asociadas a la barra de menú descrita anteriormente.

Al pasar el puntero del mouse bajo cada uno de los iconos de esta barra, se despliega un pequeño letrero, indicando que acción se tomara, si éste es pulsado. Esto se puede apreciar en la figura MU.6.

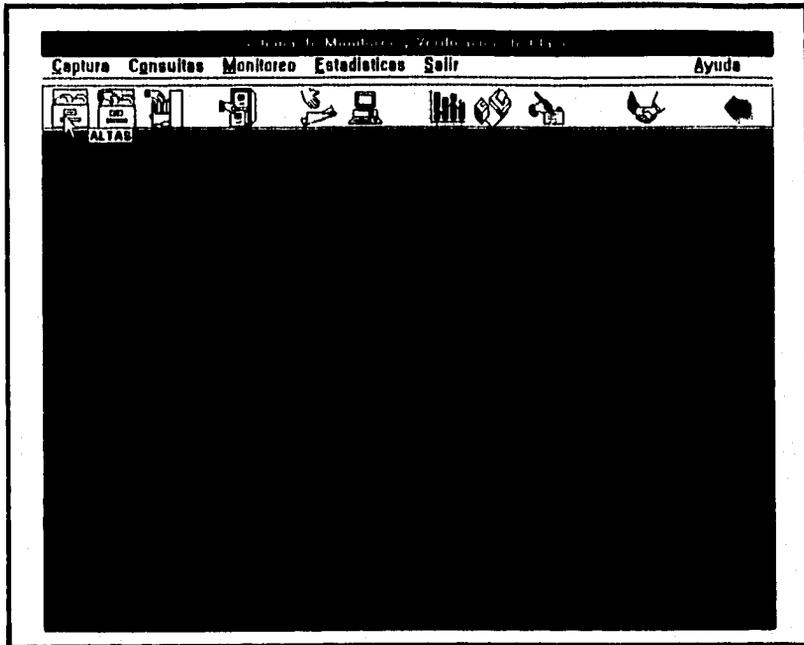


Figura MU.6 Íconos con etiquetas de acción asociadas.

MANEJO DE LOS MENÚS.

Si usted desea manejar los datos de los usuarios entonces tendrá que desarrollar todas estas operaciones a partir del menú de **CAPTURA**, que tiene disponible las opciones de:

- ALTAS
- BAJAS
- CAMBIOS



ALTAS.

Al ser seleccionada esta opción se muestra la pantalla en donde se capturarán los datos procedentes de las hojas de registros que previamente llenaron los usuarios, aclaramos que al llegar a este punto ya se debió de haber escaneado la fotografía del usuario y haberla colocado en el directorio de imágenes (c:\tesis\imagenes) para poder cargarla al control que contendrá la fotografía.

El sistema puede ser usado en línea, esto es al momento en que el usuario solicite su clave se llenará la forma de altas exceptuando el campo foto, el cual se podrá llenar posteriormente, esto con el fin de proporcionar una respuesta casi inmediata al usuario, lo que repercutirá en una disminución en el tiempo total del trámite así como en la espera que el usuario pasa para poderse conectar. En la figura MU.7 se puede apreciar la pantalla de captura de los datos de un usuario. También se aprecia que se disponen de botones para agregar un nuevo usuario (*nuevo*), *guardar* los datos capturados, *limpiar* los datos contenidos en la forma y *salir* de esa opción.

Esta tarea puede ser realizada por los siguientes usuarios : SUPERA y USUARIOA.

The screenshot shows a web-based form for user registration. It is organized into several sections:

- Top Section:** A 'Login' field, a 'Fotografía' field (represented by a square box), and three fields for 'Nombre (n)', 'Apellido Paterno', and 'Apellido Materno'.
- Address Section:** Labeled 'Dirección', it contains fields for 'Calle', 'Número', 'Colonia', 'Código Postal', 'Delegación o Municipio', and 'Teléfono'.
- Other Data Section:** Labeled 'Otros Datos', it includes fields for 'Institución', 'Lug. Nacimiento', 'Fac. Nacimiento', 'RFC', 'Semestre Actual', 'Promedio Actual', 'Cédula Prof.', 'No Materias', 'Año de Ingreso', 'Tipo de Clave', and 'Tipo de Usuario'.
- Bottom Section:** Four buttons: 'Nuevo', 'Guardar', 'Limpiar', and 'Salir'.

Figura MU.7 Pantalla de Captura de Datos del Usuario.



BAJAS

El proceso consiste en entrar al menú de captura y escoger la opción de bajas, en la pantalla se muestran los datos correspondientes para realizar la baja, proporcionando una opción de verificación de los datos para su posterior baja en caso de que sea el registro correcto. Esta opción sólo esta disponible para los usuarios : SUPERA y USUARIOA. Ver figura MU.8.

MÓDULO DE BAJAS

CONTROL

IDENTIFICADOR DE USUARIO (ID)

VERIFICAR USUARIO

SALIDA

ACTIVAR

Figura MU.8 Bajas de usuarios en el Sistema



CAMBIOS

Dentro de esta opción existe disponibilidad de uso para los siguientes usuarios : SUPERA. USUARIOA porque al igual que en las bajas es indispensable llevar un control estricto de las personas encargadas de modificar los cambios.

Dentro del menú de captura existe como última opción la de cambios, en la cual se despliega una pantalla igual que la de altas, pero con la diferencia de que aquí se puede sobrescribir o editar sobre la información del usuario de la clave seleccionada previamente, y después de haber seleccionado el botón de aceptar, los datos serán actualizados dentro de la base de datos, con los cambios que se realizaron previamente. Al pulsar esta opción, se preguntará por el login del usuario al que se desea hacer algún cambio. Ver figura MU.9. Inmediatamente después, se buscará aquel usuario que tenga asociado dicho login. Si la búsqueda fue exitosa, entonces se mostrarán los datos de ese usuario en forma de edición, es decir, se podrán modificar sus datos. Ver figura MU.10. Si la búsqueda no tuvo éxito, se mostrará un mensaje de error.

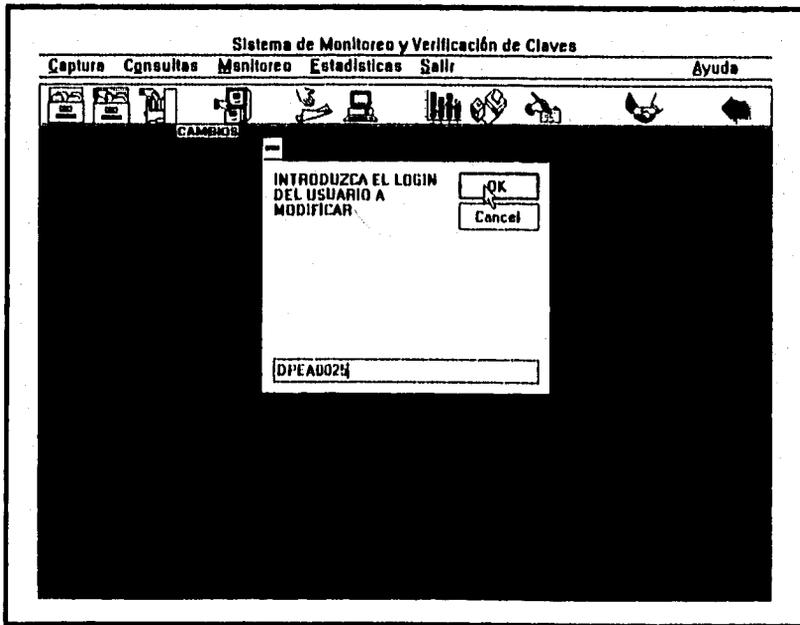


Figura MU.9 Solicitud del login del usuario a modificar



Captura de los Datos de los Usuarios

Login

Nombre (s)
 Apellido Paterno
 Apellido Materno
 Fotografía

Dirección

Calle
 Número
 Colonia

Código Postal
 Delegación o Municipio
 Teléfono

Otros Datos

Institución
 Lug. Nacimiento

No Cuenta
 Fec. Nacimiento

Semestre Actual
 RFC

Promedio Actual
 Cédula Prof.

No Materias
 Tipo de Clave

Año de Ingreso
 Tipo de Usuario

Figura MU.10 Datos del usuario a modificar, en forma de edición

CONSULTAS

Las consultas pueden ser realizadas por todos los tipos de usuarios disponibles del sistema, pueden llevarse a cabo de cuatro formas distintas : por *login*, por apellido paterno, por dependencia, y por tipo de usuario. Ver figura MU.II. En el caso de consulta por *login*, ésta es de forma individual ya que para un *login* sólo corresponde un usuario, por lo que al momento de que usted elige la opción de *consultas por login*, el resultado será el registro que tenga el *login* asociado. Para las otras formas de consultas, siempre se parte de lo general a lo particular, es decir si se quiere hacer una consulta de todos los usuarios que tengan como primer apellido X



eligiendo la opción de *consultas por apellido paterno*, se mostrarán todos los usuarios que correspondan al apellido asociado, para posteriormente y si así se desea, escoger un registro de todos los que se mostraron, y hacer una consulta particular.

El mismo caso se presenta en las consultas por dependencia y por tipo de usuario. Todas estas opciones tienen un botón que permite imprimir la consulta correspondiente. Ver figura MUI2.

MENU DE CONSULTAS

FILTRO DE BUSQUEDA

APELLIDO PATERNO C

LOGIN C

TIPO DE USUARIO C

DEPENDENCIA C

SALIDA
Salir

Figura 11. Tipos de consultas



Consulta de Usuario

Login

Nombre (s) Apellido Paterno Apellido Materno

Fotografía

Dirección

Calle Número Colonia

Código Postal Delegación o Municipio Teléfono

Otros Datos

Institución	<input type="text" value="FACULTAD DE INGENIERIA"/>	Lug. Nacimiento	<input type="text" value="MEXICO, DF"/>
No. Cuenta	<input type="text"/>	Fac. Nacimiento	<input type="text" value="22/01/70"/>
Semestre Actual	<input type="text"/>	RFC	<input type="text" value="DZL700122"/>
Promedio Actual	<input type="text" value="0"/>	Cédula Prof.	<input type="text" value="0"/>
No. Materias	<input type="text" value="0"/>	Tipo de Clave	<input type="text" value="100 Hora"/>
Año de Ingreso	<input type="text"/>	Tipo de Usuario	<input type="text" value="EXTERNO"/>

Figura MU.12 Pantalla de Datos del usuario en Consultas

MONITOREO

Dentro de este menú se presentan dos opciones, MONITOREO MANUAL y MONITOREO AUTOMÁTICO.

MONITOREO MANUAL.

Esta opción habilita al administrador para realizar paso a paso el proceso de monitoreo, desde seleccionar la línea hasta proceder a su desconexión en caso de ser necesario.



Para el monitoreo manual abrir el menú de monitoreo y seleccionar MANUAL o ALT+M. En este momento se despliega la forma para seleccionar la línea a ser monitoreada posteriormente se deben seleccionar las preguntas a ser enviadas mediante un clic donde éstas se marcarán con una cruz, una vez seleccionada la línea y las preguntas se deberá presionar el botón de *Enviar* para que el cuestionario se active en el monitor del usuario, en este momento el sistema se inactiva hasta recibir las respuestas enviadas por el usuario y procede a la validación de las mismas; en caso de ser correctas realiza la conexión de la línea y en caso contrario procede a realizar la desconexión de la misma, mediante el botón de *desconexión de línea*; para continuar con el proceso de monitoreo de otra línea. Ver figura MU.13.

MONITOREO MANUAL

LINEA SERVIDOR

Datos Personales

Nombre

Apellido Paterno

Apellido Materno

Fecha de Nacimiento

Lugar de Nacimiento

RFC

Calle

Número

Colonia

Delegación

Código Postal

Datos Escolares/Laborales

Nombre de la Institución

Teléfono de

Céd. Prof.

Núm. de Cue

Semestre en curso

Promedio Actual

Núm. de Materias

Año de Ingreso

ACTIVAR MONITOREO MANUAL (sí/no)

Verificar

No. de Intento No. de Respuestas Correctas

Activa

 No. de Preguntas Enviadas

Figura MU.13 Pantalla de Monitoreo Manual



MONITOREO AUTOMÁTICO

Este proceso es el encargado de seleccionar de manera automática la línea a ser monitoreada por el sistema; el proceso de monitoreo en éste módulo es transparente para el administrador ya que el sistema es el encargado de seleccionar la línea; elegir, enviar y validar preguntas así como realizar conexión o desconexión de la línea según sea el caso.

Este módulo se activa al arrancar la aplicación ya que se diseñó para ser la opción por omisión del sistema. Esta opción se mantiene activada hasta que el administrador cambia al modo de monitoreo manual o se sale del sistema.

Para seleccionar éste módulo del sistema abra el menú de monitoreo del mismo y haga un clic en la opción de monitoreo automático o presione ALT + A.

ESTADÍSTICAS

Bajo esta etiqueta se disponen de dos opciones las cuales su acceso esta restringido en una de ellas. Esta opción es la de las *bitácora del sistema*. Como se mencionó anteriormente, sólo la puede acceder el superusuario. Entrando en ella, se muestran todos los accesos que se han realizado de cada uno de los usuarios del sistema. Para cada acceso se despliega la siguiente información : Número de sesión, usuario que abrió esa sesión, fecha de entrada, hora de entrada, fecha de salida y hora de salida, todo bajo una forma tabular. Ver figura MU.14.

Número de Sesión	Usuario	Fecha de entrada	Hora de entrada	Fecha de salida	Hora de salida



También se puede particularizar esta información especificando a que tipo de usuario se desea visualizar los accesos que ha realizado. Ver figura MU15. De esta forma se tiene una bitácora de todos los usuarios que accesan el sistema y por consiguiente es posible rastrear usos indebidos que se pudieran presentar.

Bitácora				
Búsquedas Salir				
Nº Acceso	Administrador	Fecha de Entrada		Fecha de Salida
Nº 1	SUPER USUARIO	30/05/1996	20:23:04	30/05/1996 20:23:20
Nº 2	ADMINISTRADOR B	30/05/1996	20:23:23	30/05/1996 20:24:16
Nº 3	SUPER USUARIO	30/05/1996	20:24:18	30/05/1996 20:24:21
Nº 4	ADMINISTRADOR A	30/05/1996	20:31:50	30/05/1996 20:44:34
Nº 5	SUPER USUARIO	30/05/1996	20:44:36	30/05/1996 20:44:38
Nº 6	ADMINISTRADOR B	30/05/1996	20:47:03	30/05/1996 20:49:10
Nº 7	SUPER USUARIO	30/05/1996	20:49:11	30/05/1996 20:49:15
Nº 8	ADMINISTRADOR B	30/05/1996	20:59:23	30/05/1996 20:59:56
Nº 9	SUPER USUARIO	30/05/1996	20:59:57	30/05/1996 21:00:01
Nº 10	SUPER USUARIO	30/05/1996	21:00:34	30/05/1996 21:00:55
Nº 11	SUPER USUARIO	30/05/1996	21:01:10	30/05/1996 21:01:53
Nº 12	SUPER USUARIO	30/05/1996	21:09:06	30/05/1996 21:09:54
Nº 13	SUPER USUARIO	30/05/1996	21:11:52	30/05/1996 21:12:20
Nº 14	SUPER USUARIO	30/05/1996	21:13:52	30/05/1996 21:14:24
Nº 15	SUPER USUARIO	30/05/1996	21:15:04	30/05/1996 21:15:33

Figura MU.14. Datos tabulares de la bitácora



Búsquedas **BW**

Usuario

Total de Accesos

2

Sesion Nº	Datos de Entrada		Datos de Salida		
	Fecha	Hora	Fecha	Hora	
Nº 1	30/05/1996	20:23:04	30/05/1996	20:23:20	+
Nº 3	30/05/1996	20:24:10	30/05/1996	20:24:21	
Nº 5	30/05/1996	20:44:36	30/05/1996	20:44:30	
Nº 7	30/05/1996	20:49:11	30/05/1996	20:49:15	
Nº 9	30/05/1996	20:59:57	30/05/1996	21:00:01	
Nº 10	30/05/1996	21:00:34	30/05/1996	21:00:55	
Nº 11	30/05/1996	21:01:10	30/05/1996	21:01:53	
Nº 12	30/05/1996	21:09:05	30/05/1996	21:09:54	
Nº 13	30/05/1996	21:11:52	30/05/1996	21:12:20	+

Aceptar

Figura MU.15. Particularización de los datos de la bitácora.

La segunda opción que se dispone bajo la etiqueta de *Estadísticas* es *Gráficas*

Dentro de esta opción se visualizan las diferentes gráficas que se pueden crear con los datos almacenados en la base de datos del sistema, permitiendo tener un panorama general de los mismos. Por ejemplo, es posible obtener una gráfica de las estadísticas de usuarios con el mayor número de horas consumidas, por número de usuarios en cada dependencia, usuarios dados de baja temporal (por haber agotado sus tres intentos), frecuencia de uso del sistema por parte de los administradores, etc.



Para realizar las gráficas basta con seleccionar el conjunto de datos a los cuales se desea obtener un gráfico. Así mismo se tiene la posibilidad de escoger el tipo de gráfico que se desea obtener, pudiendo ser éste gráfica de pie, barras en dos y tres dimensiones y áreas, entre otras.

En la figura MU.16 se puede observar una muestra de éste tipo de gráficas.

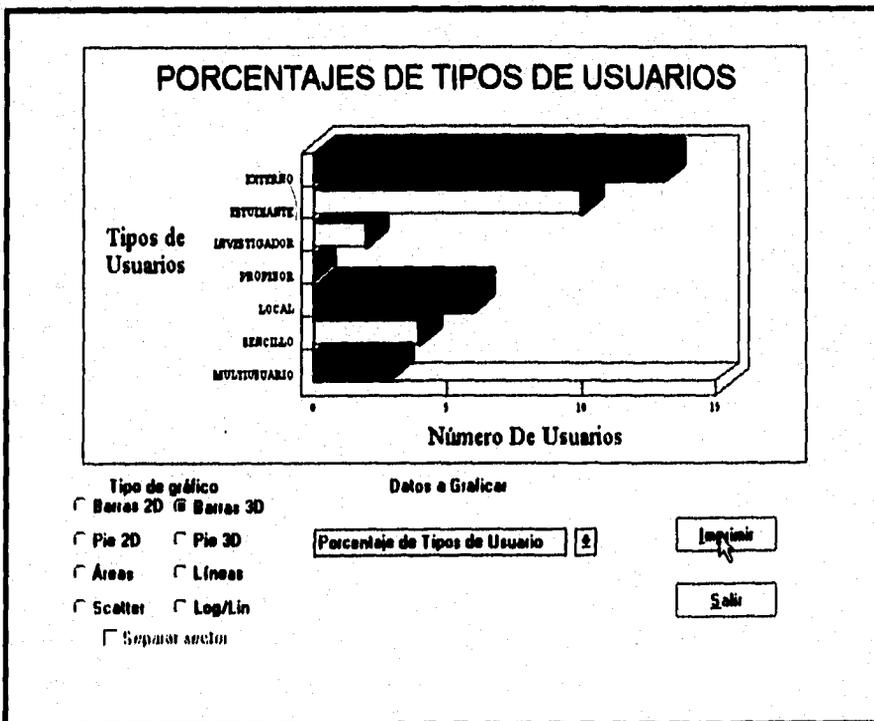


Figura MU.16. Gráficas del Sistema



AYUDA

Dentro del menú de ayuda podemos apreciar tres opciones que son : *A cerca de*, *Buscar* y *Ayuda del programa*. El diseño y uso de esta interface explicativa está basada en la misma que proporciona el ambiente gráfico Windows, es decir, se aprovechan las bondades del uso de hipertexto, búsqueda por palabras y búsquedas por tópicos, entre otras. Ver figura MU.17.

A esta opción tienen acceso todos los diferentes tipos de administradores, porque es una referencia rápida de las funciones que se pueden realizar dentro de los menús.

Basta con dar el nombre de lo que se quiere consultar, y automáticamente la ayuda desplegará la información correspondiente al tópico seleccionado, si no hay ayuda disponible se indicará mediante un mensaje.

En la opción de ayuda, se dispone de una nota informativa acerca del sistema. Ésta se puede desplegar en *acerca de...* como lo muestra en la figura MU.18.

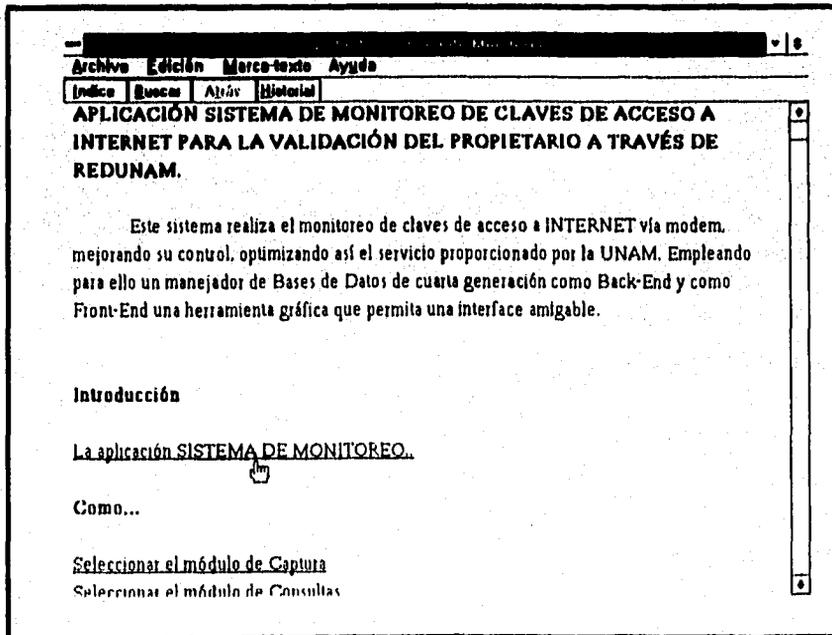


Figura MU.17. Ayuda del Sistema.

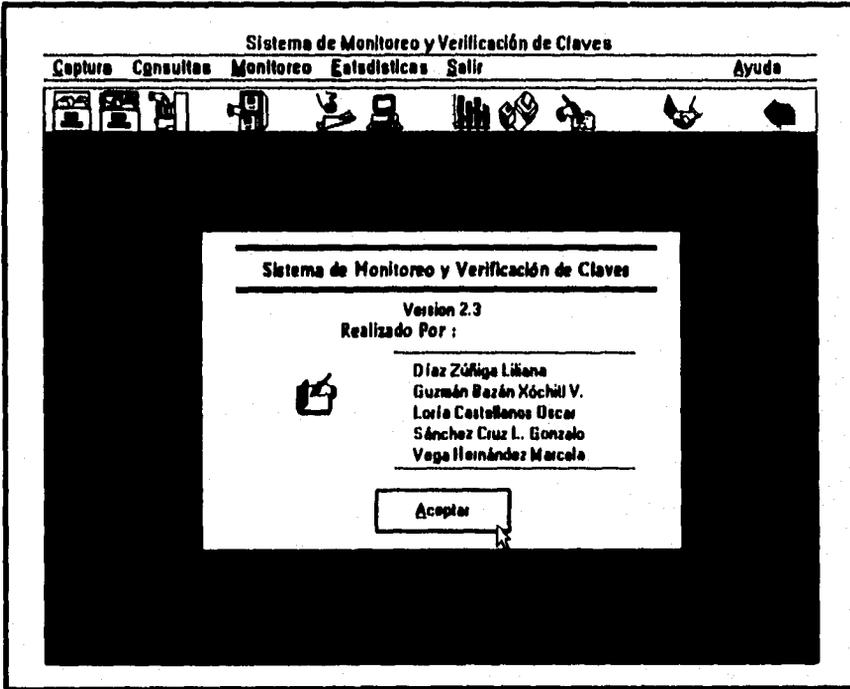
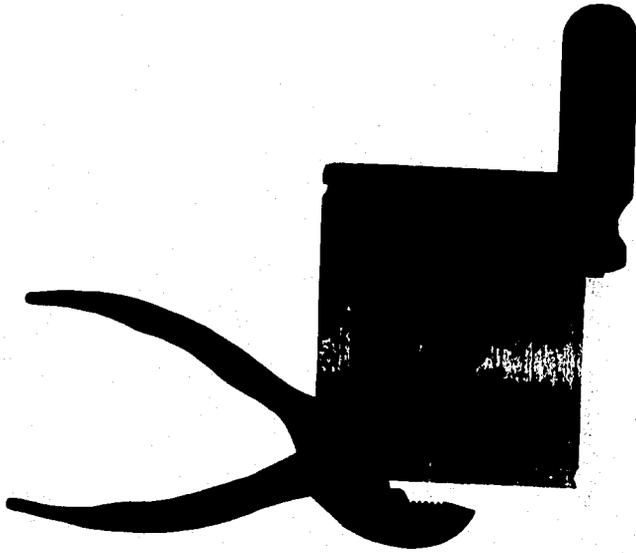


Figura MU.18. Acerca de... del Sistema.

**MANUAL
TÉCNICO**





MANUAL TÉCNICO

Debido a que el mantenimiento de la Base de Datos se realiza con el manejador de Bases de Datos Microsoft ACCESS. El manual técnico del sistema solo hace mención a este manejador. Es decir, no mencionamos en el manual técnico a VISUAL BASIC debido a que éste aspecto es tratado en el Manual de Usuario.

Primeramente accedamos a Windows, se selecciona del menú de opciones a Microsoft Office, entramos a él y localizamos el icono de ACCESS, se le da un clic con el mouse o <Enter> con el teclado. Ver figura 1.

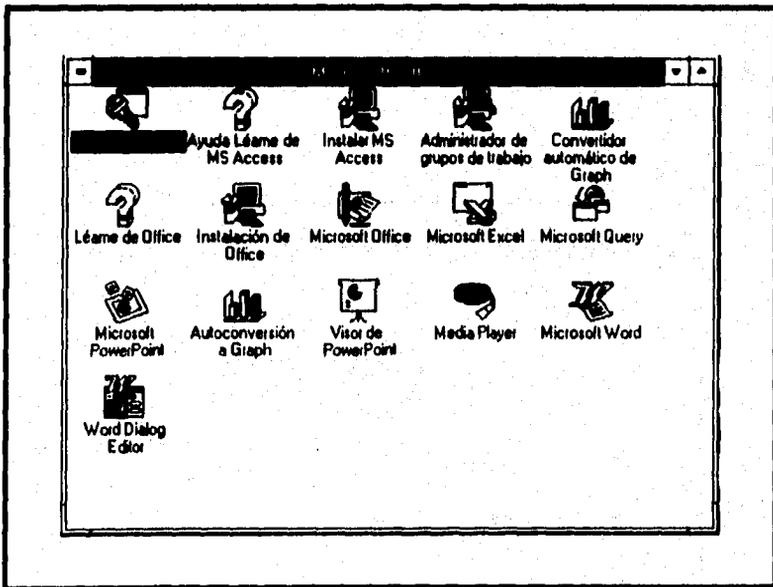


FIG. 1 Ventana de Microsoft Office



Una vez cargado Microsoft ACCESS, se procede a abrir la Base de Datos mediante los siguientes pasos:

- A) Oprimir <Alt> para tener acceso al menú principal y directamente entramos a la opción de archivo mediante un <Enter> o, seleccionamos con el apuntador del mouse esta opción y dar un clic.
- B) Inmediatamente aparecerá una lista de opciones, entonces se escoge la que dice "Abrir Base de Datos" y dar <Enter> o clic con el mouse.
- C) En este momento aparece una ventana pidiendo la base de datos que se desea abrir y directamente tecleando el nombre de la Base de Datos y <Enter> se tiene acceso a ella. Si la Base de Datos se encuentra en una ruta (*Path*) diferente a la que estamos ubicados entonces es necesario dar la ruta correcta a la computadora para que pueda localizar la Base de Datos que se desea abrir ver figura 2.

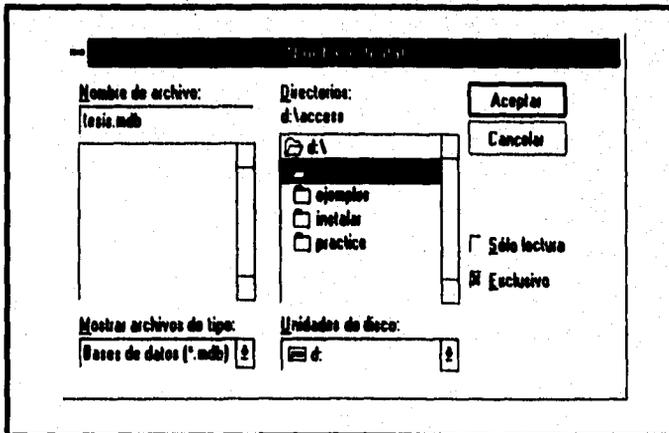


FIG. 2 Ventana de Acceso a una Base de Datos en ACCESS



Una vez abierta la Base de Datos aparece una ventana con el nombre de la Base con la que se va trabajar y con los diferentes componentes que conforman a la misma como: tablas, consultas, formularios, informes, macros y módulos. En nuestro caso los componentes que se utilizaron fueron: las tablas (USUARIOS, CLAVE, LUGAR DE NACIMIENTO, BITÁCORA, DELEGACIÓN, INSTITUCIÓN, TIPO_USUARIO). Ver figura 3. Y los formularios: INSTITUCIÓN, CLAVE, USUARIOS, ALTAS, BAJAS, CAMBIOS, CONSULTAS DELEGACIÓN.

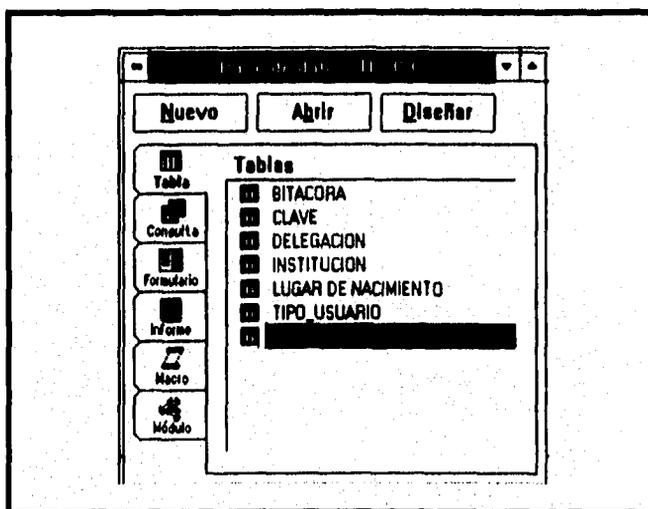


FIG. 3 Ventana de las diferentes Tablas que conforman la Base de Datos

Además se tuvo la necesidad de utilizar otra herramienta de Microsoft ACCESS que son los Macros, se realizaron macros con los siguientes nombres: Altas, Bajas, Delegación, Claves, Borrar, entre otras. Apuntando directamente a los iconos de estos componentes y seleccionando al que se desee se tiene acceso a él.

La forma principal de la Base de Datos es la forma llamada "Altas, Bajas, Cambios, Consultas", ya que mediante ella se pueden tener acceso a las formas restantes y a las tablas que



integran la Base de Datos, con esto se permite la manipulación de la información (altas, bajas, cambios, consultas) de la Base de Datos.

Entonces se procede a abrir esta forma mediante los siguientes pasos :

- A) Apuntar al icono de formulario y dar clic
- B) Aparecen las diferentes formas, entonces con las teclas de flechas o directamente con el mouse seleccionar la que se llama Altas, Bajas, Cambios, Consultas.
- C) Oprimir la tecla <Enter> o dar doble clic con el mouse.
- D) La forma esta abierta. Ver figura 4

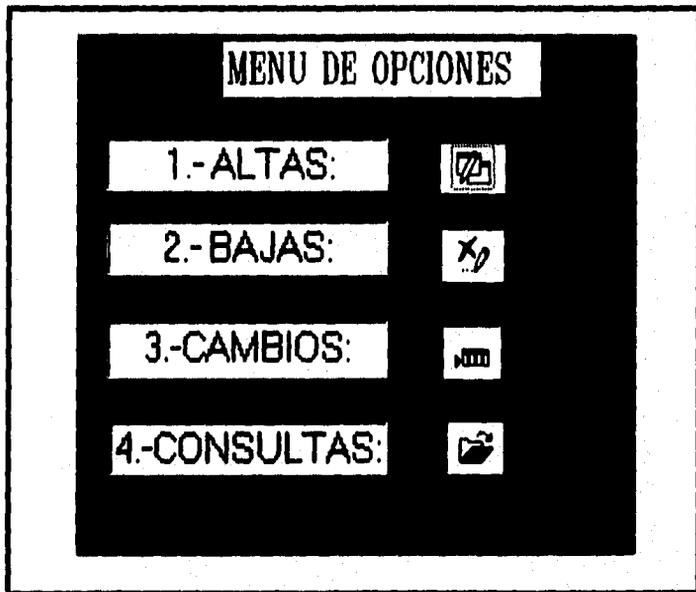


FIG. 1 Ventana del Menú de Opciones

Dependiendo de la información que se desea manipular se selecciona alguno de los comandos que aparecen en el menú.



El manejo de las formas restantes es muy similar en todas, pero como la principal realiza el llamado de las otras, nos enfocaremos a esta forma, en particular sin profundizar en las restantes:

Para dar de Alta un registro en la tabla de USUARIO mediante esta forma, es muy sencillo, solo basta seguir los siguientes pasos:

- Se selecciona con las flechas o con el apuntador del mouse el icono que dice Altas.
- Oprimir <Enter> o dar clic
- La forma de Altas esta abierta y con esto se pueden introducir los datos
- En la forma de Altas una vez abierta se encuentran con tres comandos, el primero es para borrar el registro antes de salvar, el segundo es para salvar los registros una vez llenado todos los campos y el tercero es para salirte de la forma de altas.

En esta forma como en las restantes cuenta con acceso a la información de las diferentes tablas. Esto es por ejemplo, si se llega al campo en el cual la información este predefinida con un combo¹ (como es el caso de Delegación) y no está dada de alta la Delegación a la cual pertenece, con darle doble clic, automáticamente se abre otro formulario, en la cual se da de alta a la Delegación requerida y la información se guarda en otra tabla diferente a la de USUARIO, que es la de DELEGACIÓN, un caso similar ocurre para los campos de: Lugar de Nacimiento, Tipo de Clave, Institución y Tipo de Usuario

Si deseamos borrar o modificar información contenida en algún registro de la Base de Datos, el procedimiento es fácil desde el menú principal se selecciona el icono de Bajas o Cambios, al entrar a la forma de Bajas o Cambios aparecen tres botones de comandos, el primero es para encontrar a la persona que desea realizar la baja o en el caso de cambios al usuario que deseas modificar, el segundo comando, en el caso de bajas, borra al usuario seleccionado, y un tercer comando que es de salir, el cual se sale de la forma en la que estas en ese momento.

¹ Comando que despliega una lista de elementos



Cuando estamos posicionados en un campo determinado de las diferentes tablas nos damos cuenta de que en el último renglón de la pantalla nos aparece un mensaje determinado con una leyenda en particular y no es más que una ayuda visual para conocer en que campo estamos ubicados y que es lo que el campo contiene. Esto quiere decir que mientras estamos cambiando de campo de información, el mensaje de ayuda también esta variando.

En caso de que se desee realizar modificaciones en los campos o agregar otro, es necesario seleccionar de Microsoft Access el módulo de Tablas y en la etapa de Diseño se agregan o se anulan los campos que desee. Solo hay que tener presente que si se desea anular un campo y esté contiene información, se debe de realizar una tabla temporal, en la que se tengan los campos que si se quieran guardar y una vez realizada la tabla temporal se anulan los campos, quedando como tabla original, la tabla temporal (Además existe la ayuda de Microsoft Access para cualquier aclaración).

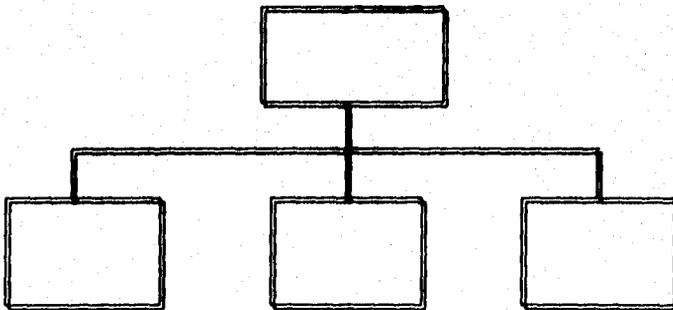
Existen dos maneras de Indexar los datos de nuestra base de datos:

Primero.- Crear desde el inicio en nuestra nueva tabla un campo tipo contador, el cual si se anula un campo el contador sigue su conteo sin importarle que se anulo un campo, es decir que una vez asignándole un número a un registro este le acompaña todo el tiempo.

Segundo.- Si es que se tiene una tabla con registros dados de alta. Se realiza lo siguiente: se crea una tabla temporal con todos los campos excepto el de tipo contador, después en la tabla temporal se agrega este campo, y finalmente la tabla que era temporal ya no lo es, es decir, la que era la tabla temporal, ya es la tabla original.

APÉNDICE A

ESTRUCTURA ORGANIZACIONAL



**ESTRUCTURA ORGANIZACIONAL DE LA UNAM**

- ◆ Facultad de Arquitectura
- ◆ Facultad de Ciencias
 - ◆ Laboratorio de Cómputo
 - ◆ Laboratorio de Visualización
- ◆ Facultad de Ciencias Políticas
- ◆ Facultad de Contaduría y Administración
- ◆ Facultad de Ingeniería
 - ◆ Departamento de Control
 - ◆ Departamento de Ingeniería Eléctrica
 - ◆ Laboratorio Cómputo Avanzado
- ◆ Facultad de Medicina
- ◆ Facultad de Química



◆ **COORDINACIÓN DE LA INVESTIGACIÓN CIENTÍFICA**

- ◆ Instituto de Astronomía
- ◆ Instituto de Biología
- ◆ Instituto de Biotecnología
- ◆ Instituto de Ciencias Nucleares
- ◆ Instituto de Fisiología Celular
- ◆ Instituto de Física
- ◆ Instituto de Geofísica
- ◆ Instituto de Geografía
- ◆ Instituto de Geología
- ◆ Instituto de Ingeniería
 - ◆ Laboratorio de Prospección Arqueológica
- ◆ Instituto de Investigaciones Biomédicas
- ◆ Instituto de Investigaciones Jurídicas
- ◆ Instituto de Investigaciones en Materiales
- ◆ Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
 - ◆ Laboratorio de Procesamiento Paralelo
- ◆ Instituto de Matemáticas



- ◆ Instituto de Química

- ◆ Centro de Ciencias de la Atmósfera
- ◆ Centro de Ecología
- ◆ Centro de Investigación sobre Fijación de Nitrógeno
- ◆ Centro de Investigaciones Interdisciplinarias en Ciencias y Humanidades
- ◆ Centro de Instrumentos
- ◆ Centro de Neurobiología

- ◆ Programa Universitario de Alimentos
- ◆ Programa Universitario de Investigación en Salud
- ◆ Programa Universitario de Energía
- ◆ Programa Universitario de Investigación y Desarrollo Espacial
- ◆ Programa Universitario del Medio Ambiente

- ◆ Centro de Información Científica y Humanística
- ◆ Centro para la Innovación Tecnológica
- ◆ Centro Universitario de Comunicación de la Ciencia



- ◆ E.N.E.P Aragón
- ◆ E.N.E.P Acatlán
- ◆ E.N.E.P Iztaacala
- ◆ F.E.S. Cuautitlan
- ◆ F.E.S. Zaragoza

- ◆ Dirección General de Estadística y Sist. de Información Institucionales
- ◆ Dirección General de Televisión Universitaria
- ◆ Dirección General de Proveduría
 - ◆ Dirección de Cómputo para la Administración Académica
 - ◆ Subdirección de Redes y Telecomunicaciones
 - ◆ Centro de Operación de RedUNAM
 - ◆ Centro de Información de RedUNAM
 - ◆ Departamento de Supercómputo
 - ◆ Área de Seguridad en Cómputo
 - ◆ Laboratorio de Visualización
- ◆ Secretaría de Asuntos Estudiantiles
 - ◆ Sistema Universidad Abierta SUA



◆ COLEGIO DE CIENCIAS Y HUMANIDADES

◆ UNIVERSUM Museo de las Ciencias

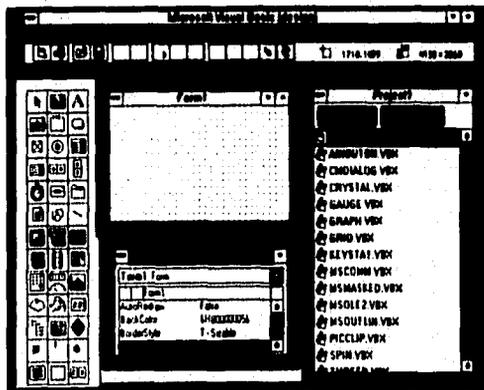
◆ Taller de Desarrollo Empresarial

◆ AIESEC México, Comité local UNAM

◆ IEEE en la UNAM

APÉNDICE B

VISUAL BASIC CARACTERÍSTICAS Y FUNCIONAMIENTO





VISUAL BASIC, CARACTERÍSTICAS Y FUNCIONAMIENTO

Introducción a Visual Basic

Visual es un poderoso sistema de programación gráfica que le permite crear aplicaciones Windows reales con código BASIC. Este sistema es un extraordinario avance para desarrollo de programas, porque combina la sintáxis simplificada de BASICA y GWBASIC con la estructura de programación QBASIC y QuickBasic.

El sistema de programación Visual Basic le permite crear *objetos*, establecer y cambiar sus propiedades y, posteriormente, asignarles un código BASIC funcional.

Visual Basic está centrado en dos tipos de objetos, ventanas y controles, que permiten diseñar sin programar, una interface gráfica (mecanismos de comunicación entre el usuario y la aplicación) para una aplicación.

La filosofía de la programación Visual Basic consiste, primero, en crear objetos como ventanas, iconos y menús y después elaborar procedimientos que sean llamados por cada uno de estos objetos. Esto es diferente del método tradicional de elaboración de un programa el cual exige estructuras para controlar el flujo del programa de un procedimiento a otro de manera lógica hasta que el programa termina. La programación con objetos es un método flexible y conveniente para escribir programas en Windows.

Por ejemplo, usted puede elaborar el código para un objeto que ha creado y después hacer copias del mismo con el código asignado; no tiene que escribir el código otra vez.

Ahora bien, definir Visual Basic como un sistema *de programación orientada a objetos* sería erróneo. Porque usted crea objetos, llamados formas y controles, que hacen que su aplicación funcione. Sin profundizar en el tema estos objetos carecen de las propiedades de herencia y polimorfismo, las cuales tienen que estar presentes en un verdadero ambiente



orientado a objetos. Además, usted está creando aplicaciones que se correrán en Windows, el cual es un ambiente orientado a lo visual que proporciona librerías de herramientas y objetos para aplicaciones de programación. La mayor ventaja de utilizar Windows es que proporciona una interfaz consistente y manejable para desplegar información en muchas aplicaciones diferentes que comparten controles similares.

La filosofía de la interfaz Windows se basa en la utilización de metáforas visuales para realizar acciones y tareas.

El paradigma es simple: después de que usted ha utilizado una aplicación Windows, aprende fácilmente a utilizar otra ya que está familiarizado con la interfaz.

Al utilizar controles Windows como menús, iconos, barras de desplazamiento y cajas de diálogo, la información es presentada a los usuarios en un formato dinámico y visualmente interesante.

Visual Basic reduce drásticamente las dificultades y la cantidad de tiempo de la programación en Windows, usted puede construir una aplicación Windows dinámica en pocos días. Ya que este lenguaje *es interpretado* más que compilado, las aplicaciones están listadas para correr tan pronto como se termine de escribir el código. En contraste, con un lenguaje compilado, un programa debe compilarse antes de que pueda ser ejecutado. Esto significa que un compilador convierte los programas al lenguaje de máquina. Esta conversión se realiza solamente una vez, cada vez que se corre el programa, el lenguaje de máquina ya está disponible y puede ser ejecutado de manera directa por la computadora.

Elaboración de programas orientado a eventos

Cuando se inicia una aplicación durante la corrida sólo las formas que usted creó y programó serán visibles. Una forma es una ventana que usted crea y adapta a sus necesidades para cada aplicación. Esta es la base de cualquier aplicación Visual Basic que eventualmente



correrá como un programa independiente de Windows. Un control es el nombre preciso para cualquier objeto que usted dibuje en una forma, pero también se refiere a la forma misma. Los botones de comando, menús, cajas de diálogo y barras de desplazamiento, son ejemplos de controles. Cuando un usuario activa un control, el código BASIC que tiene asignado el control, es invocado y realiza una tarea específica. Un módulo es una estructura para escribir el código que usted le asigna a una forma y sus controles. Después de dibujar los controles que usted desea que aparezcan en una forma, usted escribe el código funcional para cada control que los usuarios pueden invocar. Visual Basic utiliza la metáfora del "evento" para describir su paradigma de programación, siempre utilizará este enfoque orientado a los eventos cuando cree una aplicación. *Orientado a eventos*, significa que todos los 4589 controles que usted dibuje en una forma especifican cómo se comportará la interface. Esto es, los controles de Visual Basic esperan que sucedan eventos particulares antes de que respondan. Un evento es una acción que es reconocida por un control, es código escrito para cada control a fin de que algo suceda cuando los usuarios interactúan con la forma.

Mientras se está en Visual Basic y corre una aplicación, cada línea de código es interpretada y ejecutada por separado cada vez que aparece. Es esta ejecución individual de cada línea la que puede hacer que un programa interpretado corra con mayor lentitud que un programa compilado. Por consiguiente el archivo terminado también es más grande. Por supuesto, un archivo compilado es un archivo ejecutable, que ya no es interpretado.

Terminología de visual

La necesidad de familiarizarse con la terminología de Visual Basic tiene su singular enfoque orientado a eventos, para programar aplicaciones. Por consiguiente es útil para describir los elementos de una aplicación Visual Basic:

- **Control:** un término general utilizado para describir cualquier forma o elemento gráfico que usted dibuje sobre una forma, incluyendo cajas de texto, cajas de listado, botones de comandos, cajas de imágenes, barras de desplazamiento e iconos. Una definición



más precisa de un control es la de datos acoplados con una serie de rutinas, conocida como métodos. Las únicas operaciones que pueden desarrollarse sobre un control son aquellas definidas como métodos para el.

- **Evento:** acción reconocida por un control Visual Basic.

- **Forma:** una ventana que usted crea y adapta a sus necesidades para su aplicación.

- **Método:** una palabra de código Visual Basic que es similar a una función o instrucción, pero que siempre actúa sobre un control particular. Para cada control, Visual Basic predefine una serie de métodos que usted puede utilizar.

- **Procedimiento:** un término que se refiere tanto a procedimientos Sub y Function (Función). Un procedimiento es sencillamente una secuencia de instrucciones Visual Basic que son ejecutadas en grupo durante la corrida. Existen dos tipos de procedimientos:
 - *procedimientos de eventos y procedimientos generales.* Los procedimientos de eventos se limitan a las formas y los controles, mientras que los *procedimientos generales* se utilizan durante toda la aplicación y pueden ser requeridos por procedimientos de eventos.

- **Proyecto:** conjunto de todos los archivos que forman la aplicación.

- **Propiedad:** una característica o atributo de un control para cada tipo de control, Visual Basic define una serie de propiedades que se aplican solamente a ese control.

- **Definición:** el valor de una propiedad, se puede cambiar la definición de la mayoría de las propiedades mientras está construyendo una aplicación. El código de una aplicación que se corre también puede cambiar las definiciones.



Acceso al menú de barras

El ambiente de programación Visual Basic está formado por cinco elementos a los que usted accesa a fin de crear aplicaciones propias:

- Menú de barra
- Barra de propiedades
- Ventana Form
- Ventana Project
- Caja de Herramientas

Estas ventanas es lo primero que se observa cada vez que inicia Visual Basic como se ilustra en la figura 1.

Menú de barra

El Menú de barra hace aparecer los nombres de los menús que están disponibles para utilizar en la ventana activa. Encima del menú de barra esta la barra de titulo, los botones de maximización y minimización y la caja de control los cuales se muestran en la figura 2. Para entrar al menú hay varias maneras: puede hacer clic sobre un nombre del menú para abrirlo y hacer que aparezcan sus comandos o también puede oprimir una tecla aceleradora de nombre de menú, reconocidas porque la letra esta subrayada. Por ejemplo, la tecla aceleradora para el menú File (Archivo) es **r**. Ver figura 2.

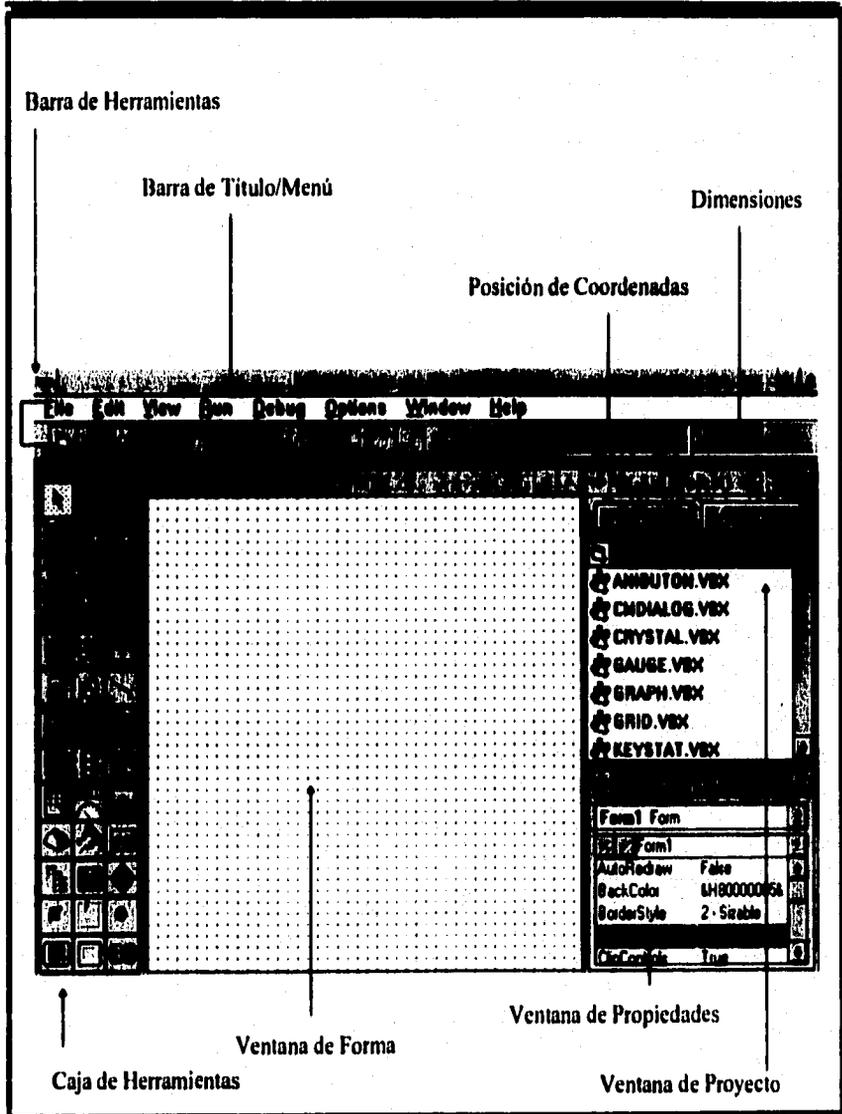


FIG. 1. El ambiente de Programación Visual Basic

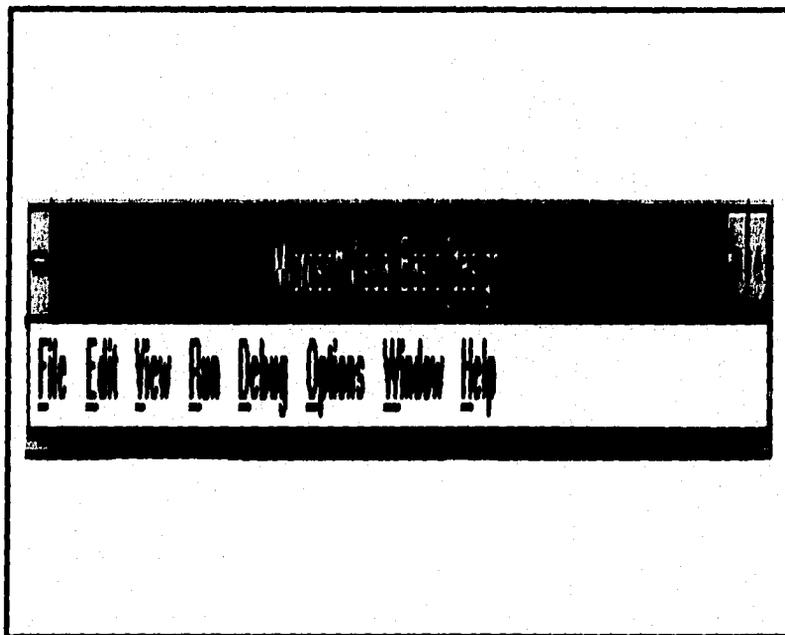


FIG. 2. Menú de barra y los títulos de la barra.

Características

El sistema dinámico de ventanas, menús y herramientas le permite crear formas y controles, manipular archivos de proyectos, manipular texto, correr programas y depurar el código. Entre las características más interesantes de Visual Basic se incluyen las siguientes:

Cuando se comienza un programa durante la corrida, puede seleccionar comandos del menú **File** (Archivo) figura 2, para poner/quitar puntos de interrupción en el código y para ejecutar un programa a fin de evaluar cómo se ejecutan las instrucciones.

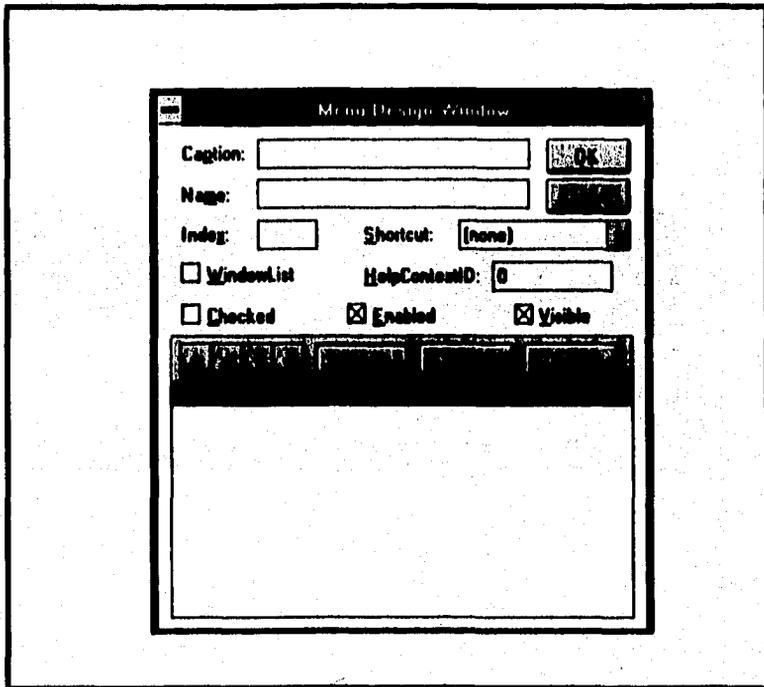


FIG. 3. Ventana de diseño de menú

- La ventana **Inmediate** (inmediato) le permite observar como se ejecuta una instrucción en particular mientras usted corre un programa.
- La ventana **Menú Design** (diseño de menú) está disponible para crear menús con teclas aceleradoras, la figura 4 muestra las características de esta ventana.
- Se puede crear un sistema Help completo, un tutorial y un índice por medio del menú Help.



- La ventana de propiedades le ofrece una manera conveniente de escoger propiedades de control y cambiar definiciones durante el diseño, la figura 4 muestra los elementos de esta barra.
- La caja de herramientas le permite crear 15 tipos básicos diferentes de controles sobre una forma.

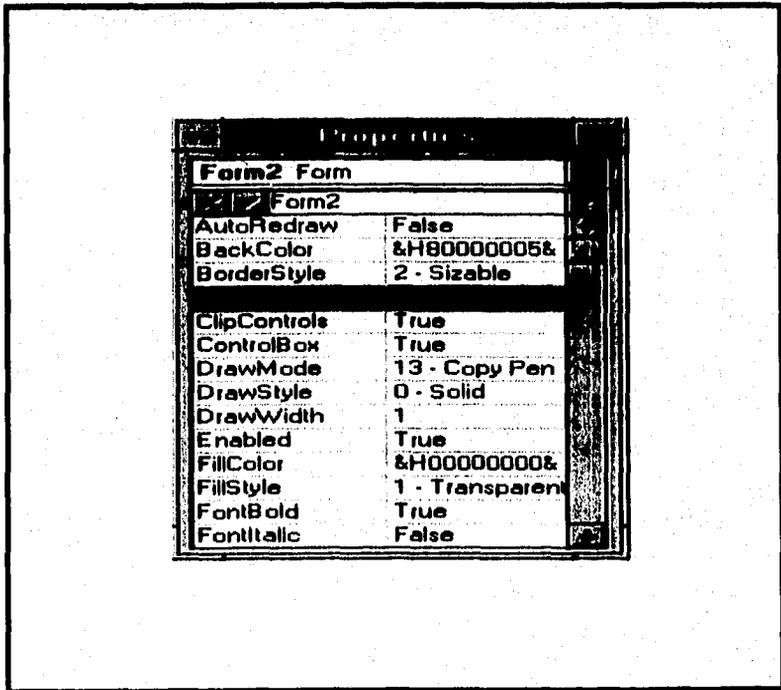


FIG. 4. Ventana de Propiedades.

- Una ventana de forma, como muestra la figura 5 por omisión aparecerá automáticamente cuando usted inicie Visual Basic. Cuando usted hace clic con el botón izquierdo del mouse (ratón, dispositivo de señalización o apuntador óptico) sobre una forma o



control, aparece una ventana Code en la cual usted escribe el código funcional para una aplicación.

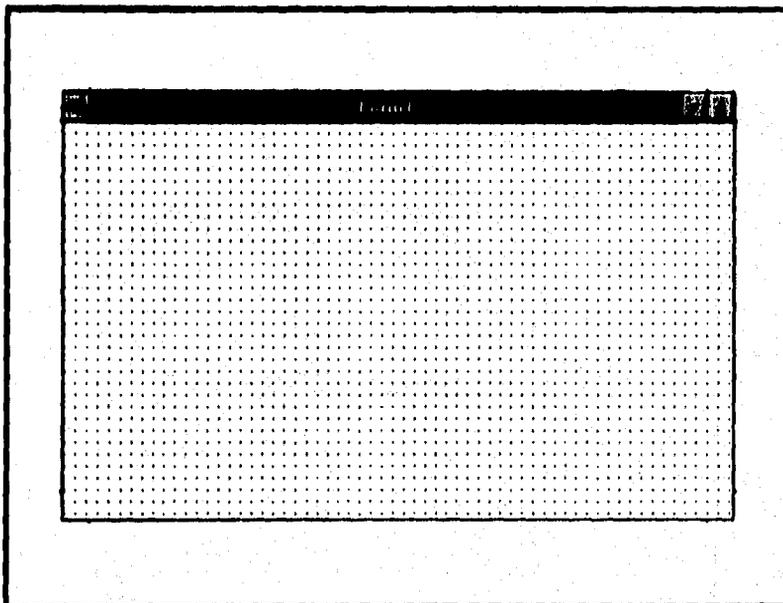


FIG. 5. Ventana Forma de muestra.

La ventana code (código) figura 6 muestra las cajas del listado de las cuales usted selecciona los nombres de controles, eventos y los nombres de procedimientos generales.

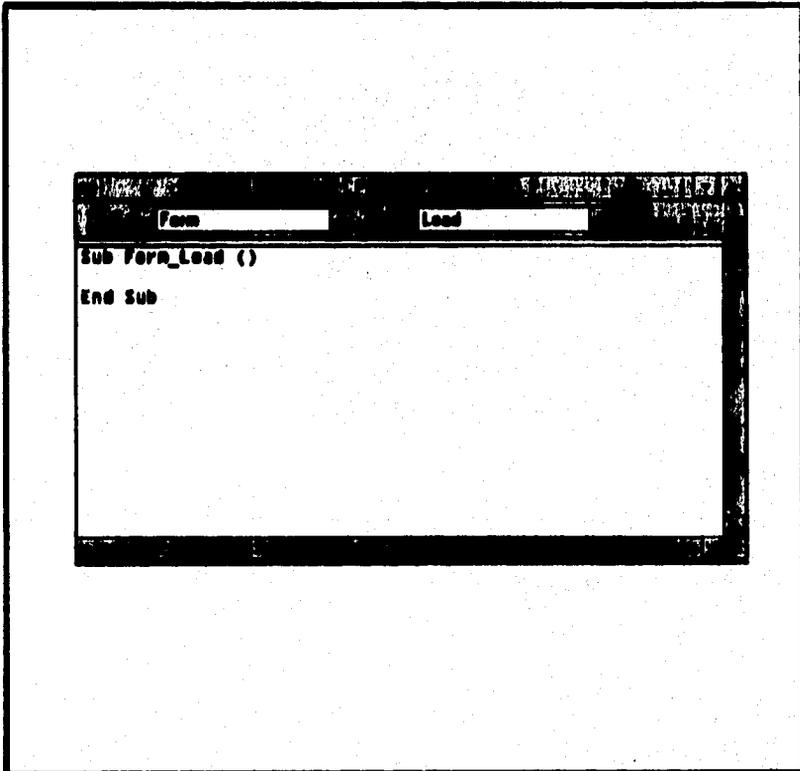


FIG. 6. Ventana de Código.

- La ventana **Project** (proyecto) organiza y muestra los nombres de todas las formas y módulos que son parte de una aplicación particular. Usted puede mostrar el código para una forma destacando el nombre del archivo en esta ventana y haciendo clic en el botón **View Code** (vista del código); no tiene que mostrar primero la forma (o control) real para observar el código. La figura 7 muestra la ventana **project**.

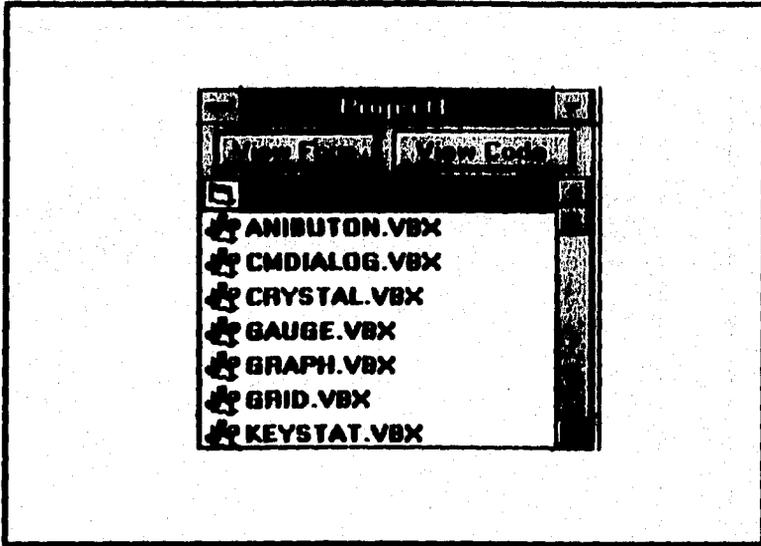


FIG. 7. Ventana Project.

Creación de controles con la caja de herramientas

La caja de herramientas Visual Basic contiene un conjunto de 24 herramientas (las cuales muestra la Figura 8) que puede utilizar para dibujar, mover o redimensionar los controles de su forma y los cuales posteriormente les asignará el código BASIC funcional.

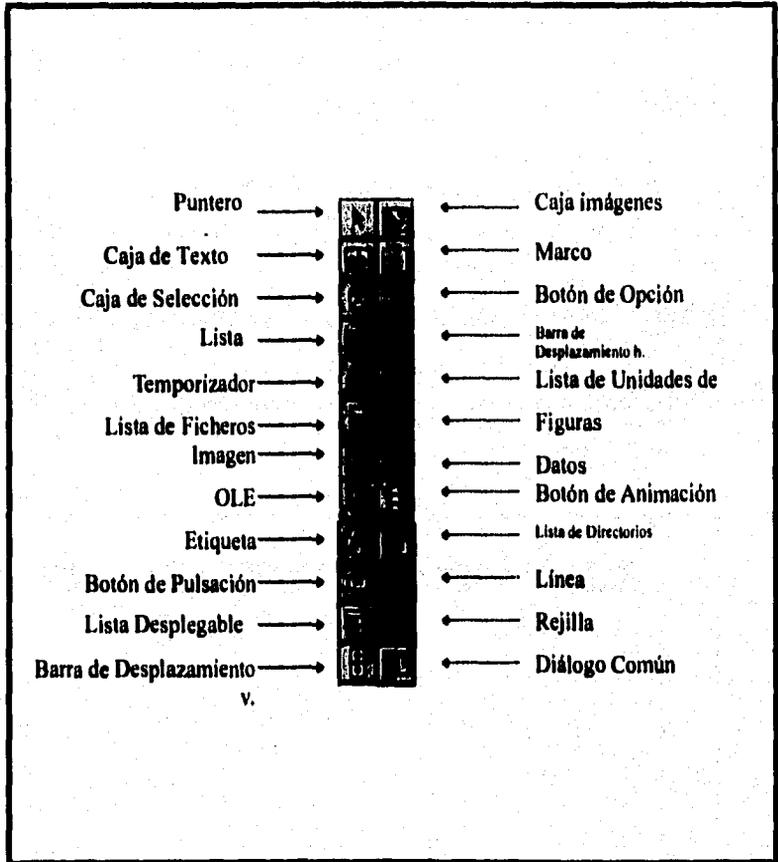


FIG. 8. Caja de Herramientas de Visual Basic

Creación de un control

Para crear cualquier control, siga estos pasos:

1. Haga clic sobre la herramienta Text Box para seleccionarla. El apuntador del ratón se convertirá en una cruz.



2. Coloque la cruz sobre la forma en la posición deseada.
3. Mantenga apretado el botón izquierdo del ratón y arrastre la cruz hacia abajo diagonalmente hacia la derecha hasta que el control tenga el tamaño deseado.
4. Suelte el botón del ratón. El encabezado `Text1` aparecerá sobre la caja de texto y *selection handles* aparece alrededor de la caja para indicar que esta seleccionada.
5. Centre la caja de texto sobre la forma, como se muestra en la figura 9. arrastrándola.
6. Usted también puede arrastrar las manijas de selección para aumentar o disminuir el tamaño de la caja de texto.

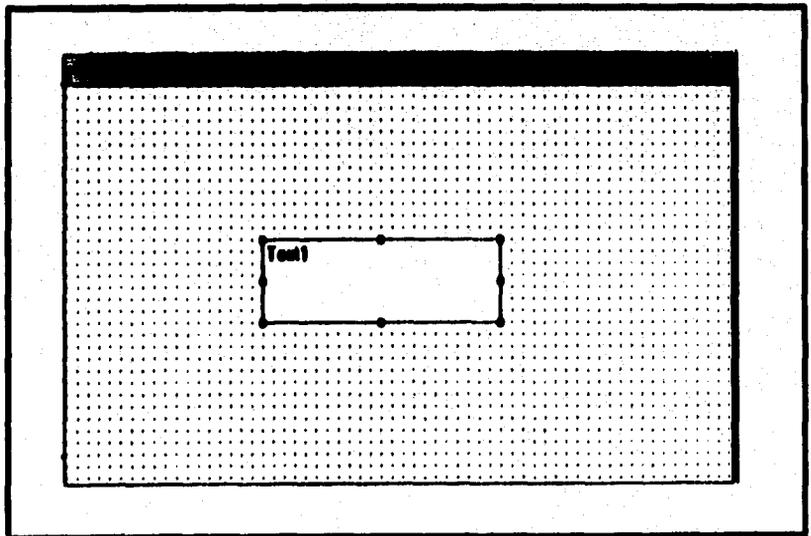


FIG. 9. Caja de texto centrada sobre la forma.



Modificación de las propiedades de un control

Ahora que usted ha creado un control de caja de texto, puede establecer las propiedades tanto para la forma como para la caja de texto para que aparezcan como usted desea.

Cambie **Caption** que aparece en la barra de título de la forma:

1. Seleccione la forma haciendo clic sobre ella (a menos que este ya seleccionada). El nombre **Form1** aparece en la caja de listado **Setting** para la propiedad **Caption**. (La caja de listados **Settings** esta a la mitad de la barra de propiedades, debajo de la barra de menú de **Visual Basic**).
2. Mueva el apuntador del ratón a la caja de listados **Settings**. Se convertirá en un cursor en forma de **I** al que se le llama rayo **I**.
3. Haga resaltar el título haciendo doble clic sobre él o colocado el rayo **I** al principio de la palabra y arrastrándolo a la derecha.
4. Oprima la tecla **Delete** (**Borrar**) para quitar **Caption**.
5. Teclee **Welcome** (**Bienvenido**) y presione **enter** para confirmar la nueva definición, o haga clic sobre el botón **enter** (botón izquierdo del ratón).

Ahora cambiemos la propiedad **Name** (**Nombre**) de la caja de texto para que usted pueda referirse a ella en un procedimiento:

1. Con la caja de texto sobre la forma seleccionada, haga clic sobre la caja de listado **properties** (a la izquierda de la caja de listado **settings**).



2. Seleccione la propiedad **Name**, sino ha sido seleccionada. El nombre **Text1** aparece en la caja de listado **Settings**.

3. Teclee **Display** en la caja de listado **Settings** y oprima **Enter** para confirmar el cambio.

Ahora cambie la propiedad **Text** (Texto) del control:

1. Seleccione la propiedad **Text** en la caja de listado **Properties**. El nombre de **Text1** aparece en la caja de listado **Settings**.

2. Haga un doble clic dentro de la caja de listado **Settings** para resaltar el texto.

3. Oprima la tecla **Delete** para quitar el texto.

4. Oprima **Enter** (Introducir) para confirmar el cambio.

Finalmente, cambie la propiedad **BorderStyle** (Estilo de Borde) del control:

1. Haga clic sobre la caja del listado de **Properties** que esta abierta, mueva consecutivamente la lista, y seleccione la propiedad **BorderStyle**. La definición **1** aparece por omisión en la caja **Settings**, el cual es el valor por omisión de una sola línea compuesta.

2. Haga clic sobre la flecha del lado derecho de la caja de listado **Settings** (la flecha indica que existen más definiciones para la propiedad). Aparece una lista.

3. Seleccione la definición **Settings**, la cual es el valor para no tener límites.

4. Haga clic sobre el botón **Enter** para confirmar el cambio. La caja de texto deberá aparecer como la que se muestra en la figura 10.

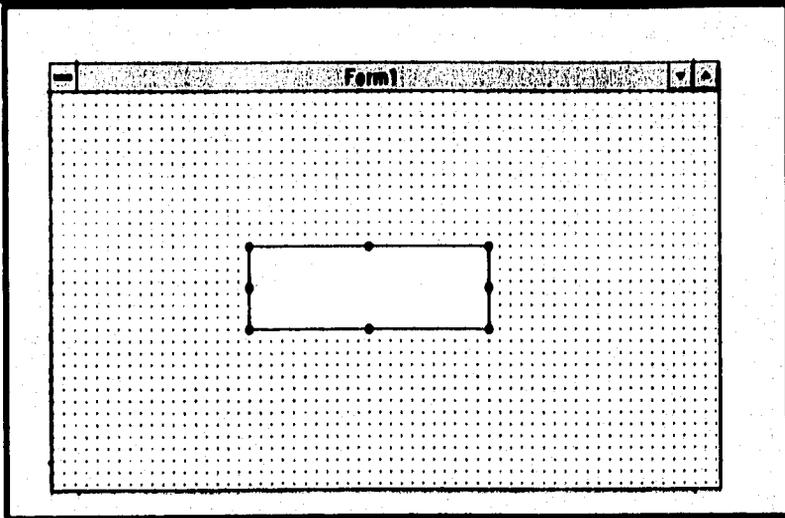


FIG. 10 *Modificando sus propiedades.*

Escritura del código funcional

Para escribir el código para un evento Load, siga estos pasos:

1. Haga doble clic en cualquier parte de la forma. La ventana de código aparecerá, como se muestra en la figura 11. Una plantilla de código aparece automáticamente, mostrando la primera y la última línea del procedimiento de evento. En este ejemplo, el procedimiento de evento es identificado con el nombre `Form_click()`.
2. Haga clic sobre la flecha hacia abajo del lado derecho de la caja de listado Procedures.

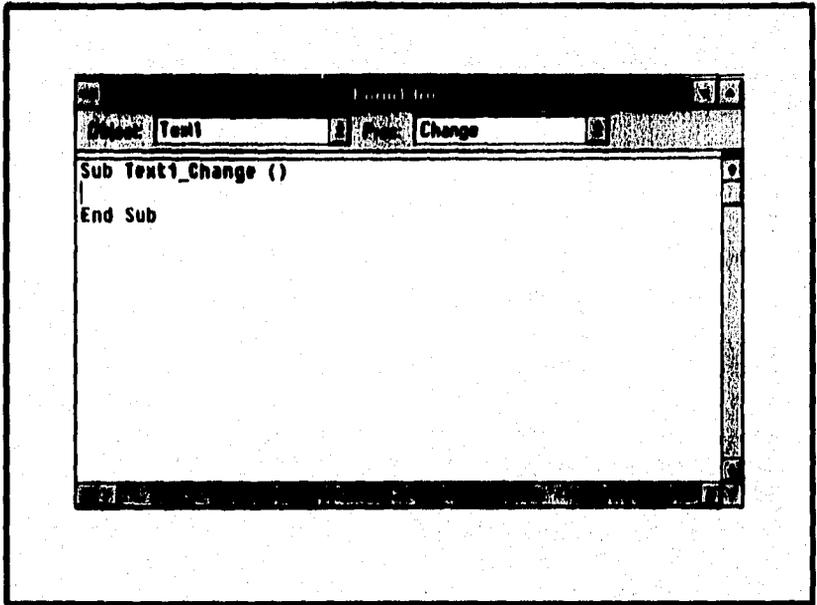


FIG. 11. Ventana de código con el evento clic mostrado en la forma.

3. Seleccione Load (Cargar). El nuevo evento aparecerá en la plantilla de código automáticamente.
4. Coloque la barra destellante de inserción entre las instrucciones Sub y End Sub.
5. Presione la tecla Tab para darle sangría a esta línea (es un buen estilo de programación escribir las instrucciones con sangría).
6. Teclee: Display.Text=Welcome to Visual Basic
7. Haga doble clic sobre la caja de control para cerrar la ventana de código.



8. Abra el menú **Run** (correr) y seleccione el comando **Star** (Ejecutar) para correr el programa. La aplicación hará aparecer el mensaje **Welcome to Visual Basic**, como se muestra en la figura 12.

9. Abra el menú **Run** de nuevo y seleccione el comando **End** (Terminar) para detener la ejecución del programa.

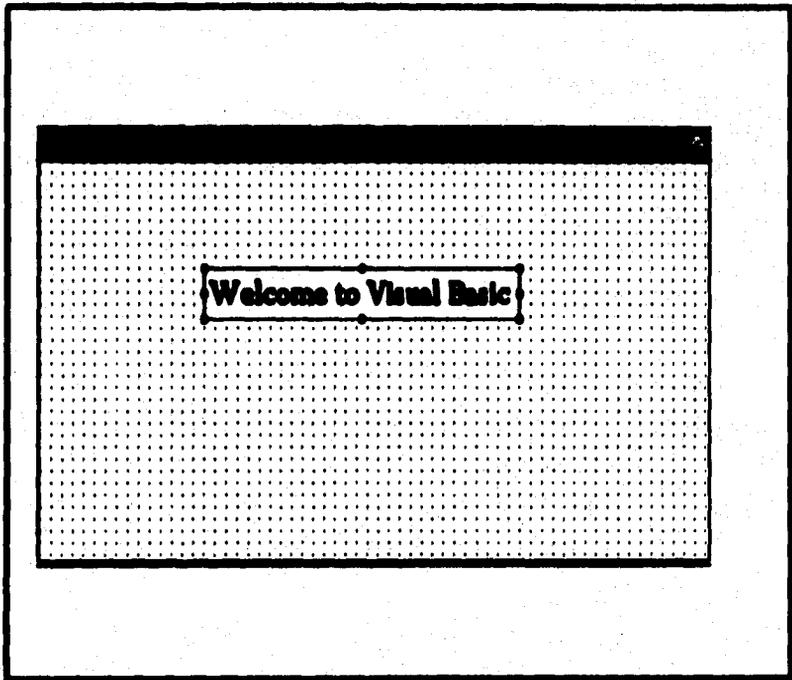


FIG. 12 Forma con mensaje "Welcome to Visual Basic"

Para crear cualquier control automáticamente, solo hacer doble clic sobre la herramienta en particular, aparecerá un control de tamaño por omisión en el centro de la forma. Para



redimensionar un control, arrastre sus manijas de selección. Para reposicionar un control, sólo arrástrelo a otra posición dentro de la forma.

UTILIZACIÓN DEL AMBIENTE DE TRABAJO VISUAL BASIC

¿Que es una interface?

El término interface para el usuario es difícil de definir. Si usted pregunta a varios programadores cómo lo definirían, es probable que obtenga respuestas muy diferentes porque no hay interface para el usuario estándar para aplicaciones basadas en DOS. Aunque las aplicaciones Windows suelen reflejar especificaciones que los programadores pueden seguir para mantener una consistencia funcional entre las aplicaciones. Esto no quiere decir que las aplicaciones Windows no se apeguen a las convenciones generales de diseño. Pero se puede aplicar una cierta libertad, que con frecuencia resulta en aplicaciones Windows tan diferentes como los programadores que las crearon.

El área de las interfaces gráficas para el usuario abarca todo desde los aspectos filosóficos hasta psicológicos, y se ha convertido en una importante disciplina, captando la atención de programadores e investigadores en las áreas de la computación, ingeniería y psicología cognoscitiva. Lo que ha surgido es una visión del usuario, una persona, tal como usted, que utiliza una computadora para realizar muchas tareas y cumplir con su trabajo.

Generalmente, la interface entre usted y la computadora consiste en una pantalla, un teclado y un ratón. Eso es lo que presenta información a usted y acepta información suya.

Usted interactúa con su computadora respondiendo con lo que ve en la pantalla, escribiendo comandos en el teclado o apuntando y haciendo clic con el ratón.

Mediante una aplicación, usted se comunica o más precisamente, tiene una interface con su computadora. De hecho, una aplicación Visual Basic como todas las aplicaciones Windows,



se accesa mediante una interface gráfica para el usuario, de Microsoft. Una interface es más que una visión de objetos, texto, gráficas y patrones. Una interface gráfica para el usuario (o interface computadora-humano), tal como una aplicación, es el "diálogo" que ocurre entre una computadora y sus usuarios. En este caso, el usuario interactúa con la computadora dándole comandos; la computadora, a su vez, le responde al usuario realizando una tarea. En otras palabras, la interface es aquel componente de una aplicación que traduce una acción del usuario con las consecuencias de sus acciones.

Visual Basic permite diseñar la interface de una aplicación utilizando interface Windows para dar a los usuarios un método consistente de interacción con la computadora. Probablemente, la consistencia sea la mayor ventaja de una interface. Todas las aplicaciones utilizan comandos y controles cuando usted aprende una aplicación Visual Basic. Una manera como Visual Basic mantiene la consistencia de una aplicación con otra, es mediante el uso de objetos y metáforas de la vida real. La metáfora de las ventanas es utilizada porque representa una experiencia de la vida real que la mayoría de nosotros comprendemos.

La creación de una aplicación efectiva comienza con un buen diseño y una visión amplia del contenido y propósito de la aplicación. La manera en que usted organiza una aplicación depende de la información que ésta contiene y de cómo espera que los usuarios la utilicen, y no sencillamente para que la reserven.

Finalmente, una persona correrá la aplicación que usted ha creado fuera de Visual Basic, simplemente como cualquier otra aplicación Windows. Por tanto, es importante que se detenga y se haga usted mismo varias preguntas fundamentales que afectarán el diseño:

- ¿Cuál es el propósito de la aplicación?
- ¿Qué razón (o razones) tiene el usuario para utilizarla?
- ¿Qué información se presenta en la aplicación?



- ¿Cómo se da a entender esta información en la aplicación?

- ¿Cómo encuentra el usuario la información en la aplicación?

- ¿Qué indicaciones le da la aplicación al usuario?

Contestando estas preguntas, estará en mucho mejor posición para empezar a construir una aplicación efectiva. Si toma en consideración el diseño de una aplicación antes de comenzar, verá mejores resultados y se ahorrará mucho tiempo (y dolores de cabeza).

Mediante el proceso de hacer claras, las necesidades de los usuarios de su aplicación, usted la creará para que la utilicen todas las personas. Los usuarios serán capaces de transitar fácilmente por ella, siguiendo una trayectoria lógica, intuitiva y no lineal, marcada con condiciones claras. Una de las maneras de dar instrucciones es representarlás mediante controles. Por tanto una aplicación bien diseñada debe guiar a los usuarios de un punto a otro, permitiéndole salir en cualquier momento, convirtiéndose, los controles que usted coloque en su aplicación como las señales del camino que los usuarios seguirán, para no perderse.

GUÍAS PARA EL DISEÑO DE UNA INTERFACE

Existen diferentes maneras de enfocar la tarea de diseñar una interface y construir una aplicación. No obstante, usted puede seguir estos pasos generales:

1. Cree un nuevo proyecto (o utilice el proyecto elaborado cuando usted inicio Visual Basic) para organizar las partes de su aplicación.

2. Elabore una forma para cada ventana en su aplicación.

3. Dibuje los controles para cada forma.



4. Cree una barra de menús para la forma principal.
5. Establezca propiedades de forma y control.
6. Escriba procedimientos de evento y procedimientos generales.
7. Guarde su trabajo.
8. Depure su código.
9. Cree un archivo ejecutable para convertir el proyecto en una aplicación.
10. Cree un ícono sobre el que pueda hacer clic para abrir la aplicación desde el programa Manager. (Usted puede crear su propio ícono con la aplicación Icon Works que se incluye en Visual Basic, o puede utilizar cualquiera de los cientos de íconos que se incluyen en la biblioteca de íconos de Visual Basic.)

Para analizar el diseño de una interface, observemos nuevamente el ambiente de trabajo Visual Basic. Cuando usted abre Visual Basic, ¿qué es lo primero que nota? ¿qué piensa de las diferentes ventanas conforme las explora? ¿son fáciles de acceder los menús y comandos? ¿son útiles las cajas de herramientas, la barra de propiedades, la forma por omisión y las ventanas project? ¿todos estos elementos hacen atractivo o limitativa la utilización de Visual Basic? Lo que es más importante, ¿le ayudan a usted?

Sería ideal que el contenido de una aplicación y su diseño se complementaran uno al otro, en donde uno reflejara la intención del otro. Es decir, cualquier control que usted cree debe servir para una función obvia y debe ayudar a los usuarios a transitar por toda la aplicación.



Alcance de las variables locales y globales

Todos los lenguajes de programación de alto nivel funcionan sobre el principio de establecer un alcance de variables locales y globales. El alcance de variables se refiere a cómo una aplicación reconoce cada una de las variables que se han declarado en la aplicación. En Visual Basic, el alcance está determinado sobre la base de dónde declara usted las variables: en el módulo global o en un módulo, en la forma misma.

Las variables y constantes con frecuencia son declaradas en: el módulo global de la aplicación. (Usted no puede ubicar procedimientos o funciones en el módulo global, únicamente puede declarar variables, constantes y tipos.) El colocar las declaraciones aquí le brinda a las variables su más amplio alcance, por eso el nombre de "global". Su colocación también las pone en el nivel más lejano de alcance en una aplicación. La ventaja de declarar variables en el nivel global, es que siempre mantienen su valor en toda la corrida de la aplicación. En otras palabras, las variables son persistentes.

Las variables a nivel modular se incluyen en la sección de declaraciones en un módulo separado, no en la sección de una forma. La ventaja de declarar las variables en un módulo, es que todos los procedimientos en la aplicación pueden reconocer y compartir las variables. Esto es importante cuando usted tiene más de una forma en una aplicación refiriéndose a los mismos procedimientos generales y funciones.

Por ejemplo, usted solamente tiene que declarar la variable una vez en el procedimiento general de un módulo. Cualquier procedimiento de evento dentro de toda la aplicación puede entonces llamar al procedimiento general.

Las variables a nivel forma se incluyen en la sección de declaraciones de un procedimiento general de forma. Usted declara siempre las variables a nivel de forma cuando desea que varios procedimientos dentro de la forma compartan datos; una variable a nivel de forma es reconocida por todos los procedimientos asignados a la forma. Tanto los niveles de



forma como las de nivel modular se consideran también persistentes; mantienen sus valores después de que un procedimiento ha sido ejecutado, pero sólo dentro de la forma.

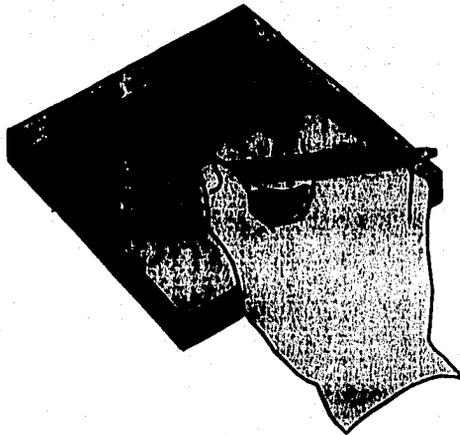
Las variables locales se reconocen solamente por el procedimiento en el cual aparecen (de ahí el nombre de local). También tiene el nivel más cercano de alcance. Utilizar una variable local es una buena manera de declarar cualquier variable que es temporal, porque el valor de la variable puede con frecuencia cambiar en un procedimiento. Además, cada procedimiento puede declarar una variable con el mismo nombre; usted puede nombrar muchas variables locales con el mismo nombre, en tanto las coloque en procedimientos separados. Si usted cambia un nombre de variable, las otras variables con el mismo nombre no serán afectadas.

Es necesario tener en mente que pueden ocurrir conflictos cuando usted declara una variable local con el mismo nombre que le ha dado a una variable en otro nivel.

Como una variable local tiene el más cercano nivel de alcance, cualquier referencia a la variable será accesada primero. Si una variable con el mismo nombre existe a un nivel más alto, la instrucción en la cual se encuentra la variable podría no ejecutarse adecuadamente, dando como resultado un error. Cuando depura un programa, no hay nada más frustrante que descubrir que el culpable del conflicto es un nombre de variable. Cuando le sea posible, declare una variable en un sólo lugar.

APÉNDICE C

CUARTA Y QUINTA FORMAS NORMALES





CUARTA FORMA NORMAL.

Para fines prácticos en la Cuarta y Quinta Forma Normal se tomará un ejemplo diferente a el que hemos tomado como referencia. Supóngase que se da una relación no normalizada que contiene información acerca de cursos, profesores y textos. Cada registro en la relación se compone de un nombre de curso, más un grupo de repetición de nombre de profesores, más un grupo de repetición de nombres de textos. La tabla 1. muestra dos de esos registros.

CTX

FISICA	PROF. VERDE PROF. PARDO	MECANICA BASICA PRINCIPIOS DE OPTICA
MATEMATICAS	PROF. BLANCO	ALGEBRA MODERNA

Tabla 1. Tabulación de muestra de CTX (no normalizada)

Ahora se convierte esta estructura en forma normalizada equivalente. Obsérvese primero que en los datos no existe dependencia funcional en absoluto. Por lo tanto; la única operación es la de "aplanar" la estructura, la cual, para los datos de la tabla 1. produce la tabulación mostrada en la tabla 1.2. Esta claro que la relación CTX contiene mucha redundancia, lo que ocasiona problemas con las operaciones de actualización. Sin embargo , CTX está en FNBC (forma Normal de Boyce/Codd), porque es "toda llave" y no hay otros determinantes funcionales. Es fácil advertir que los problemas se reducirían si CTX se reemplazara por sus dos proyecciones CT(CURSO,



PROFESOR) y CX(CURSO, TEXTO). Figura 3 (CT y CX son "sólo llave" y, por lo tanto, ambas están en FNBC).

CTX

FISICA	PROF. VERDE	MECANICA BASICA
FISICA	PROF. VERDE	PRINCIPIOS DE OPTICA
FISICA	PROF. PARDO	MECANICA BASICA
FISICA	PROF. PARDO	PRINCIPIOS DE OPTICA
FISICA	PROF. NEGRO	MECANICA BASICA
FISICA	PROF. NEGRO	PRINCIPIOS DE OPTICA
MATEMATI CAS	PROF. BLANCO	ALGEBRA MODERNA
MATEMATI CAS	PROF. BLANCO	GEOMETRIA PROYECTIVA

Tabla 1.2 Tabulación de muestra de CTX(normalizada)



Dada una relación R con atributos A,B y C, la *dependencia multivaluada* (DMV)

$$R, A \twoheadrightarrow R, B$$

se cumple en R si y sólo si el conjunto de valores de B que corresponden a un par (valor de A, valor de C) dado en R depende tan sólo del valor de A y es independiente del valor de C. Como siempre, A, B y C pueden ser compuestos.

CT	<table border="1" style="margin: auto;"> <tr style="background-color: black; height: 20px;"> <td colspan="2"></td> </tr> <tr> <td style="padding: 5px;">FISICA</td> <td style="padding: 5px;">PROF. VERDE</td> </tr> <tr> <td style="padding: 5px;">FISICA</td> <td style="padding: 5px;">PROF. PARDO</td> </tr> <tr> <td style="padding: 5px;">FISICA</td> <td style="padding: 5px;">PROF. NEGRO</td> </tr> <tr> <td style="padding: 5px;">MATEMATICAS</td> <td style="padding: 5px;">PROF. BLANCO</td> </tr> </table>			FISICA	PROF. VERDE	FISICA	PROF. PARDO	FISICA	PROF. NEGRO	MATEMATICAS	PROF. BLANCO
FISICA	PROF. VERDE										
FISICA	PROF. PARDO										
FISICA	PROF. NEGRO										
MATEMATICAS	PROF. BLANCO										
CX	<table border="1" style="margin: auto;"> <tr style="background-color: black; height: 20px;"> <td colspan="2"></td> </tr> <tr> <td style="padding: 5px;">FISICA</td> <td style="padding: 5px;">MECANICA BASICA</td> </tr> <tr> <td style="padding: 5px;">FISICA</td> <td style="padding: 5px;">PRINCIPIOS DE OPTICA</td> </tr> <tr> <td style="padding: 5px;">MATEMATICAS</td> <td style="padding: 5px;">ALGEBRA MODERNA</td> </tr> <tr> <td style="padding: 5px;">MATEMATICAS</td> <td style="padding: 5px;">GEOMETRIA MODERNA</td> </tr> </table>			FISICA	MECANICA BASICA	FISICA	PRINCIPIOS DE OPTICA	MATEMATICAS	ALGEBRA MODERNA	MATEMATICAS	GEOMETRIA MODERNA
FISICA	MECANICA BASICA										
FISICA	PRINCIPIOS DE OPTICA										
MATEMATICAS	ALGEBRA MODERNA										
MATEMATICAS	GEOMETRIA MODERNA										

FIG 3 Tabulación de Muestra de CT y CX



Ahora se puede definir la Cuarta Forma Normal (4FN): Una relación R está en Cuarta Forma Normal (4FN) si y sólo si (siempre que exista una DMV en R , por ejemplo $A \twoheadrightarrow B$) todos los atributos de R también son funcionalmente dependientes de A (es decir, $A \twoheadrightarrow X$ para todos los atributos X de R).

En otras palabras, las únicas dependencias (DF's o DMV's) en R son de la forma $K \twoheadrightarrow X$ (esto es una dependencia funcional de una llave candidata K a algún otro atributo X).

Ahora se puede advertir que la relación CTX no está en 4FN, porque comprende una DMV que no es en absoluto una DF, sin embargo las dos proyecciones CT y CX están en 4FN. De esta manera la 4FN es un mejoramiento sobre la FNBC, en el sentido de que elimina otra forma de estructura indeseable.

QUINTA FORMA NORMAL.

Existen relaciones que no se pueden descomponer sin pérdidas en dos proyecciones (como en la 4FN), pero que se pueden descomponer sin pérdida en tres (o más). Considérese la relación SPJ (figura 4). Esta relación es "toda llave" y no comprende DF's ni DMV no triviales, y por lo tanto está en 4FN. La figura 4 también muestra las tres proyecciones SP , PJ y JS de SPJ sobre $(J\#, S\#)$. Como se satisface si y sólo si la relación que se estudia es la de algunas de sus proyecciones, tal restricción se llama dependencia de reunión (DR) «*(SP,PJ,JS)». En general, la relación R satisface la $DR^*(X,Y,\dots,Z)$ si y sólo si es la reunión de sus proyecciones sobre X,Y,\dots,Z , donde X,Y,\dots,Z , son subconjuntos del conjunto de atributos de R . El proceso de descomposición se puede repetir hasta que todas las proyecciones estén en Quinta Forma Normal (5FN).

Una relación R está en quinta forma normal (5FN) - también llamada forma normal de proyección-reunión (FN/PR) si y sólo si toda dependencia de reunión en R está implicada por las llaves candidatas de R .



La relación SPJ no está en 5FN; su sola llave candidata, la combinación (S#, P#, J#), en verdad no implica que la relación se pueda descomponer sin pérdidas en sus proyecciones SP, PJ y JS. Las proyecciones SP, PJ y JS están en 5FN porque no contienen ninguna DR en absoluto.

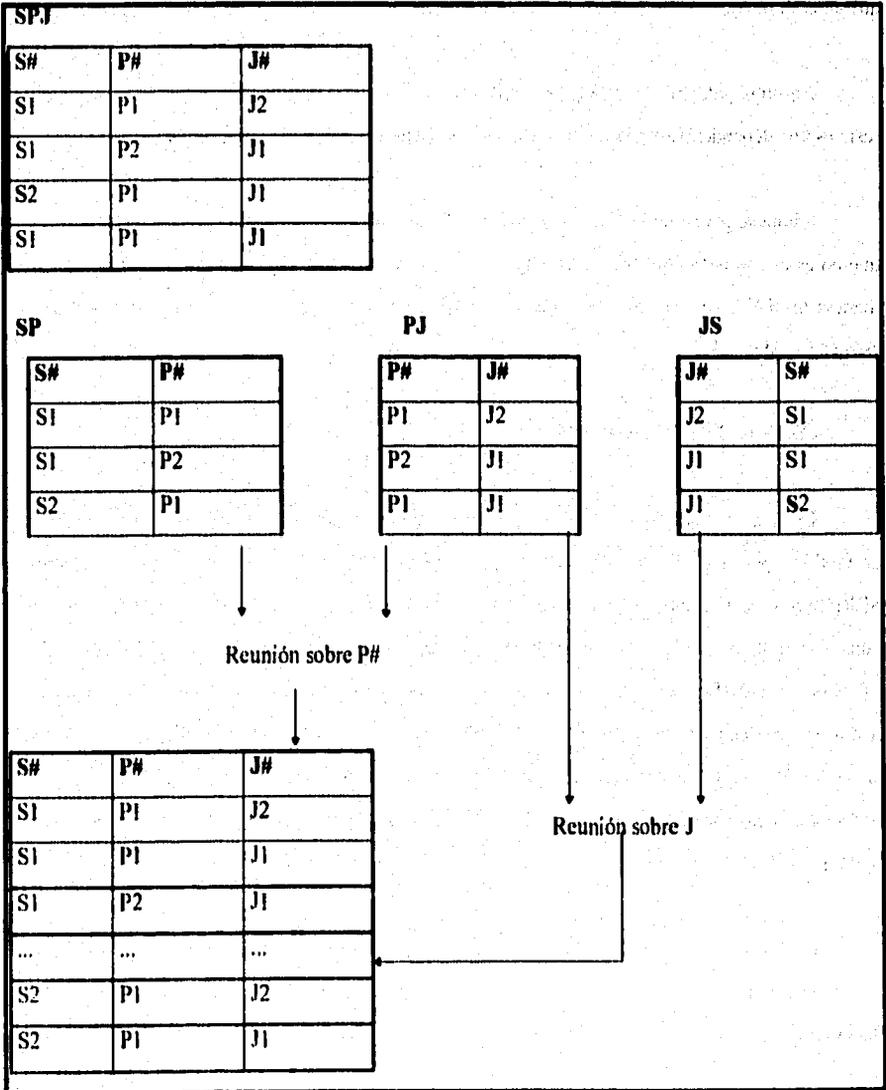
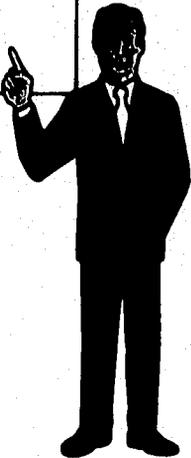


FIG 4 SPJ es la reunión de tres de sus proyecciones, pero no de dos cualesquiera

APÉNDICE D

POLÍTICAS DE REDUNAM



POLÍTICAS
•SANCIONES
•CONEXIONES



POLÍTICA DE USO ACEPTABLE PARA REDUNAM

Introducción

La política de uso es una guía para definir el uso aceptable de **RedUNAM**. **RedUNAM** es una red de comunicación de datos que tiene como propósito principal el de servir en la transmisión e intercambio de información entre organizaciones académicas y de investigación a servicios locales, nacionales e internacionales a través de conexiones con otras redes fuera de **RedUNAM**. En general, es la responsabilidad de las redes externas el indicar sus propias políticas.

La comunicación de tráfico no necesita conformarse con la política de uso aceptable de **NSFNET** (Red de la *National Science Foundation*) y no existe la prohibición del tráfico comercial. Pero, el usuario deberá observar la política de uso aceptable de cualquier otra red que acceda a través de la conexión de **RedUNAM**.

El tráfico sobre la **NSFNET** es primordialmente para investigación, desarrollo y de academia. El uso de la **NSFNET** para fines comerciales como regla general no es permitido, excepto si da soporte a la investigación y desarrollo de la academia.

RedUNAM no acepta responsabilidad por tráfico que origina del que viola la política de uso aceptable a redes directamente o indirectamente conectados mas que informar al usuario de su violación si la red conectada informa a **RedUNAM**.



POLÍTICAS

1. La intención de la política de uso aceptable es clarificar que el uso de la red sea consistente con los propósitos de **RedUNAM**, no de enunciar todos los posibles usos.
2. Todo tráfico originado por dependencias u organizaciones conectadas tendrían el propósito de servir o dar soporte a:
 - la investigación;
 - academia;
 - asuntos de gobierno local, estatal o nacional;
 - desarrollo económico o de servicio público;
3. Se considera que la información y los recursos disponibles a través de **RedUNAM** son privadas y los individuos y organizaciones de quienes son dueños tienen los derechos sobre estos recursos o información a menos que sea explícitamente expresado por los dueños o tenedores de derechos sobre estos recursos. Por lo tanto no es uso aceptable de **RedUNAM** acceder información o recursos sin autorización de los dueños o tenedores de derechos sobre recursos o información.
4. No es permitido el uso de **RedUNAM** para fines ilícitos.
5. No es permitido interferir o disturbar a usuarios, equipo o servicios en la red. Los disturbios incluyen, pero no limitan a la distribución de anuncios no solicitados, propagación de virus o gusanos (worm) informáticos, o cualquier otro uso que cause el congestionamiento de la red.
6. Si el uso es consistente con los propósitos de **RedUNAM**, entonces las actividades que soportan directamente a este uso ser no consideradas como consistentes con el propósito de

**RedUNAM.**

Por ejemplo, comunicaciones administrativas que soportan actividades aceptables son permitidas.

7. Conexiones que crean patrones de ruteo que no son consistentes con la efectividad del uso compartido de la red no son permitidos.

8. El uso de la red para juegos no son permisibles cuando su uso provoca una carga excesiva sobre recursos escasos (por ejemplo, líneas conmutadas o privadas).

ACCIONES

1. El centro de operación podrá en cualquier momento determinar si un uso en particular es consistente o no con el propósito de *RedUNAM*.

2. Cuando el centro de operaciones es enterado de un uso no aceptable, su personal notificará al administrador responsable de la dependencia o de la red externa para que se tome acción para remediarlo inmediatamente. En una emergencia, se podrá desconectar temporalmente al usuario o red involucrada. Si se considera necesario, un esfuerzo por parte del personal del centro de operaciones informar al responsable de la red o sistema antes de dicha desconexión, y se hará todo lo posible por restablecer la reconexión en cuanto se considere seguro.

CONEXIONES DE RedUNAM A INTERNET

RedUNAM tiene tres conexiones permanentes a Internet a través de los siguientes proveedores:

- *Advanced network & Systems, Inc (Houston, Texas).*
- *Rice University (Houston, Texas)*



- *National Center for Atmospheric Research (Boulder, Colorado).*

Actualmente están incluidas mas de 8,000 computadoras.

Sistemas de tiempo compartido: Mainframes, servidores y estaciones de trabajo de diferentes marcas: *Sun, HP, IBM, Silicon Graphics., Netx, DEC*, y una super computadora *CRAY Y-MP/464*.

Aproximadamente 4500 microcomputadoras.

A ellas se encuentran conectadas otras instituciones educativas, de investigación y comerciales, como son:

Banco de MÉXICO,

Televisa.

Banco Nacional de México (BANAMEX).

Universidad Iberoamericana.

Universidad La SALLE.

Instituto Tecnológico Autónomo de México. (ITAM).

Universidad Autónoma Metropolitana (UAM).

Secretaría de Comunicaciones y Transportes. (SCT).

Colegio de México.

Consejo Nacional de Ciencia y Tecnología (CONACYT).

Centro Nacional para la Prevención de desastres (CENAPRED).

Instituto Mexicano del Petróleo.

Consortio RedUno-Universidad Tecnológica de Nezahualcoyotl.

Instituto de Investigaciones Eléctricas (Cuernavaca, Morelos).

CICESE (Ensenada, Baja California).

Universidad de Guadalajara.

Universidad de Guanajuato.

**OBJETIVOS:**

Los objetivos que persigue esta gran red son:

- a) Promover el intercambio de ideas, pensamientos y opiniones que enriquezcan a los pueblos, instituciones e individuos.
- b) Apoyar el crecimiento de la UNAM y de México, brindando una opción tangible para el libre tránsito de información entre las diversas instituciones generadoras y transformadoras de conocimientos del país y del mundo.
- c) Acercar los bancos de información y otras fuentes de conocimientos a todo estudiante, personal académico y administrativo, y en general, a todo aquel que así lo quiera.

Se comunican con RedUNAM a través de algunos de los siguientes medios:

1.- Dentro del área metropolitana.

- Radio modem.
- Líneas conmutadas o privadas.
- RDI.

2.- Resto de la república Mexicana.

- RDI.
- Enlaces satelitales.



En las redes propias de la UNAM las topologías más empleadas son variantes de Ethernet: en primer lugar se tienen las redes tipo estrella, (conocida también como red par trenzado pues este es el medio físico con el que se construye). Es posible encontrar este tipo de redes complementado con verticales de coaxial grueso en coaxial delgado, aunque su uso empieza a decaer debido a sus desventajas frente al par trenzado.

Las redes de Token Ring se encuentran en franca desaparición.

Que servicios ofrece RedUNAM al usuario ?.

En primer lugar, aquellos que permiten la comunicación entre los diversos host (entendiéndose por estos las máquinas que funcionan como servidores y como clientes):

- Ruteo sin el cual no sería posible la comunicación con los hosts de otras redes. Se ha fijado cuantas veces accesa o se comunica con máquinas que no están en su red local?.

El ruteo se encarga de que sus archivos no se extravíen en alguna parte del mundo o le lleguen a la persona equivocada. También es posible limitar el acceso a su host consentido de los usuarios indeseables.

- DNS (*Domain Name Service*) que resuelve la conversión entre las direcciones lógicas y los nombres de las máquinas, de forma que el usuario pueda utilizar el nombre pata.estanque.granja para comunicarse con su servidor de correo en lugar de la dirección 219.98.174.189

- NOC/NIC (*Network Operation Center y Network Information Center*) que se encargan de atender el monitoreo y mantenimiento de la red, así como de las asesorías requeridas por los usuarios, respectivamente.

En base a estos servicios el usuario puede hacer uso de otros, tales como:



- *Servidores de correo.* Uno de los servicios de mayor demanda es el correo electrónico, el cual permite la comunicación entre usuarios de cualquier parte del mundo a través de la colocación de mensajes de texto en el buzón electrónico del destinatario. RedUNAM cuenta con varios servidores de correo, dos de los cuales están destinados al público universitario y académico en general: UNAMVML y REDVAXI.

- *Gophers.* Menús jerárquicos que permiten buscar información en RedUNAM e Internet a través de conexiones transparentes al usuario. En la UNAM contamos por el momento con dos gophers: Condor y NOC que contienen información de interés general y concerniente a la UNAM.

- *Archie.* Búsqueda electrónica en los servidores anunciados para tal fin. Basta con definir el tema a buscar y en pocos minutos se obtendrá una lista que contiene los nombres y direcciones de las máquinas que contienen archivos referentes al tema y que son de dominio público. Dichos archivos pueden ser copiados por el usuario vía FTP Anónimo.

- *Servicios de telnet.* Sesiones remotas a grandes computadoras para aprovechar sus altas capacidades de cálculo y otros recursos que difícilmente se encuentran en máquinas menores. La UNAM cuenta con sistemas Unysis y Hewlett Packard para sus procesos administrativos y de atención a usuarios, varios sistemas UNIX para los cientos de proyectos de investigación y recopilación de información que se desarrollan en los distintos y centros de difusión, y diversos mainframes para el desarrollo de la cultura informática entre la población estudiantil.

La máquina más conocida de la UNAM es la supercomputadora CRAY Y-MP/464 en la que se desarrollan interesantes proyectos de Astronomía Latina y entre sus usuarios hallamos investigadores de varias instituciones nacionales y extranjeras.

Además, de buscar continuamente nuevos servicios que ofrecer, siempre con la finalidad de brindarle al usuario una mejor opción para la adquisición de información nacional e internacional.

APÉNDICE E

FORMAS DE INSCRIPCIÓN





**Dirección General de Servicios de Cómputo Académico
Departamento de Atención a Usuarios**

USUARIO INTERNO ALUMNO

Solicitud de claves de acceso de RedUNAM

Nombre	Apellido Paterno	Apellido Materno
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	Teléfono Particular
Fecha de Nacimiento (m/d/a)	Lugar de Nacimiento	
Nombre de la Escuela/Facultad		
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	
Semestre o año	Núm. De Cuenta	
Materias que cursas (actualmente)		
Año de Ingreso a Facultad/Escuela	RFC	
Promedio Actual:	Horario de Clases	

Vo Bo Dir. de la Dependencia

Certificó que el usuario es colaborador de esta Dependencia

Firma de Compromiso

Cantidad de Horas a utilizar	<input type="checkbox"/>	10	<input type="checkbox"/>	30	<input type="checkbox"/>	50	<input type="checkbox"/>	100
Tarifa por hora para Usuarios Internos Alumnos \$6.00. La Tarifa se paga por adelantado en cantidades mínimas de 10 horas. Al agotarse la cuota se suspenderá la clave. Todo servicio extra se paga aparte.								
POLÍTICAS GENERALES DE USO DE RedUNAM LAS CLAVES SON PERSONALES E INTRANSFERIBLES. ES RESPONSABILIDAD DEL POSEEDOR DE LA CLAVE MANTENER LA CONFIDENCIALIDAD DE LA CONTRASEÑA CORRESPONDIENTE. EL POSEEDOR SE OBLIGA A RESPETAR LAS POLÍTICAS DE USO ACEPTABLE IMPRESAS AL REVERSO.								



POLÍTICAS DE USO ACEPTABLE PARA RedUNAM

RedUNAM es una red de comunicación de datos que tiene como propósito principal servir en la transmisión e intercambio de información entre organizaciones académicas y de investigación, a servicios locales, nacionales e internacionales a través de conexiones con otras redes. Además puede dar servicio de apoyo a asuntos de gobierno locales, estatales, nacionales o internacionales; al desarrollo económico, al servicio pública empresas comerciales. La intención de esta política de uso aceptable es fomentar que el uso de la red sea consistente con los propósitos de RedUNAM, no de enunciar todos los posibles usos.

Se considera que la información y los recursos disponibles a través de RedUNAM son privados y sus dueños tienen derechos sobre ellos, a menos que renuncien explícitamente a ellos. Por lo tanto no es aceptable emplear RedUNAM para acceder información o recursos sin autorización de los dueños o tenedores de derechos sobre recursos o información.

No se permite el uso de RedUNAM para fines ilícitos.

No se permite molestar a usuarios, o interferir con el equipo o servicios de la red. Las molestias incluyen entre otras, la distribución de anuncios no solicitados, propagación de virus o gusanos (worm) informáticos, o cualquier otro uso que cause congestión en la red.

No se permite conexiones que creen patrones de ruteo que afecten la efectividad del uso compartido de la red.

No se permite el uso de la red para juegos cuando provoquen una carga excesiva sobre recursos escasos (por ejemplo, líneas conmutadas o privadas).

ACCIONES

1.- El centro de operación podrá en cualquier momento determinar si un uso particular es consistente o no con el propósito de RedUNAM.

2.- Cuando el centro de operación detecte un uso no aceptable, su personal notificará al usuario, al administrador responsable de la dependencia o de la red externa para que se tome acción para remediarlo inmediatamente. En una emergencia, se podrá desconectar temporalmente al usuario o red involucradas. El personal del centro de operaciones hará un esfuerzo para informar al responsable de la red o sistema antes de dicha desconexión, y se hará todo lo posible por restablecer la conexión en cuanto se considere que el uso no aceptable se ha suspendido.

En general, corresponde a las otras redes externas fijar sus políticas de uso aceptable y los usuarios deberán conocerlas. En particular, el tráfico debe obedecer al uso aceptable de NSFNET (Red de la *National Science Foundation*). El tráfico sobre la NSFNET es primordialmente para investigación, desarrollo y docencia. El uso de la NSFNET para fines comerciales, como regla general, no es permitido, excepto si da soporte a la investigación, desarrollo o docencia. Sin embargo el tráfico comercial de RedUNAM no circula por NSFNET, y por lo tanto es aceptable.

RedUNAM no tiene responsabilidad por tráfico que envíe y que viole la política de uso aceptable a redes directamente o indirectamente conectadas. El responsable es el usuario que origine ese tráfico.



**Dirección General de Servicios de Computo Académico
Departamento de Atención a Usuarios**

USUARIO INTERNO PROFESOR

Solicitud de claves de acceso de RedUNAM

Nombre	Apellido Paterno	Apellido Materno
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	Teléfono Particular
Fecha de Nacimiento	Lugar de Nacimiento	

Nombre de la Escuela/Facultad		
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	
Núm. de Folio de la Cédula Profesional		
Materias que imparte (actualmente)		
RFC		
Horario de Trabajo		
Hora de entrada/salida		

Vo Bo Dir. de la Dependencia
Certificó que el usuario es colaborador de esta Dependencia

Firma de Compromiso

Cantidad de Horas a utilizar 10 30 50 100

Tarifa por hora para Usuarios Internos Profesor \$6.00. La Tarifa se paga por adelantado en cantidades mínimas de 10 horas. Al agotarse la cuota se suspenderá la clave. Todo servicio extra se paga aparte.

***POLÍTICAS GENERALES DE USO DE RedUNAM**
LAS CLAVES SON PERSONALES E INTRANSFERIBLES. ES RESPONSABILIDAD DEL POSEEDOR DE LA CLAVE MANTENER LA CONFIDENCIALIDAD DE LA CONTRASEÑA CORRESPONDIENTE. EL POSEEDOR SE OBLIGA A RESPETAR LAS POLÍTICAS DE USO ACEPTABLE IMPRESAS AL REVERSO.



POLÍTICAS DE USO ACEPTABLE PARA RedUNAM

RedUNAM es una red de comunicación de datos que tiene como propósito principal servir en la transmisión e intercambio de información entre organizaciones académicas y de investigación, a servicios locales, nacionales e internacionales a través de conexiones con otras redes. Además puede dar servicio de apoyo a asuntos de gobierno locales, estatales, nacionales o internacionales; al desarrollo económico, al servicio pública empresas comerciales. La intención de esta política de uso aceptable es fomentar que el uso de la red sea consistente con los propósitos de RedUNAM. no de enunciar todos los posibles usos.

Se considera que la información y los recursos disponibles a través de RedUNAM son privados y sus dueños tienen derechos sobre ellos, a menos que renuncien explícitamente a ellos. Por lo tanto no es aceptable emplear RedUNAM para acceder información o recursos sin autorización de los dueños o tenedores de derechos sobre recursos o información.

No se permite el uso de RedUNAM para fines ilícitos.

No se permite molestar a usuarios, o interferir con el equipo o servicios de la red. Las molestias incluyen entre otras, la distribución de anuncios no solicitados, propagación de virus o gusanos (worm) informáticos, o cualquier otro uso que cause congestión en la red.

No se permite conexiones que creen patrones de ruteo que afecten la efectividad del uso compartido de la red.

No se permite el uso de la red para juegos cuando provoquen una carga excesiva sobre recursos escasos (por ejemplo, líneas conmutadas o privadas).

ACCIONES

1.- El centro de operación podrá en cualquier momento determinar si un uso particular es consistente o no con el propósito de RedUNAM.

2.- Cuando el centro de operación detecte un uso no aceptable, su personal notificará al usuario, al administrador responsable de la dependencia o de la red externa para que se tome acción para remediarlo inmediatamente. En una emergencia, se podrá desconectar temporalmente al usuario o red involucradas. El personal del centro de operaciones hará un esfuerzo para informar al responsable de la red o sistema antes de dicha desconexión, y se hará todo lo posible por restablecer la conexión en cuanto se considere que el uso no aceptable se ha suspendido.

En general, corresponde a las otras redes externas fijar sus políticas de uso aceptable y los usuarios deberán conocerlas. En particular, el tráfico debe obedecer al uso aceptable de NSFNET (Red de la *National Science Foundation*). El tráfico sobre la NSFNET es primordialmente para investigación, desarrollo y docencia. El uso de la NSFNET para fines comerciales, como regla general, no es permitido, excepto si da soporte a la investigación, desarrollo o docencia. Sin embargo el tráfico comercial de RedUNAM no circula por NSFNET, y por lo tanto es aceptable.

RedUNAM no tiene responsabilidad por tráfico que envíe y que viole la política de uso aceptable a redes directamente o indirectamente conectadas. El responsable es el usuario que origine ese tráfico.



**Dirección General de Servicios de Computo Académico
Departamento de Atención a Usuarios**

USUARIO INTERNO INVESTIGADOR

Solicitud de claves de acceso de RedUNAM

Nombre	Apellido Paterno	Apellido Materno
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	Teléfono Particular
Fecha de Nacimiento	Lugar de Nacimiento	

Escuela/Facultad/Dependencia		
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	Núm. Telefonico
Nombre del Proyecto a Cargo		
Fecha de Inicio del Proyecto	Nombre del Jefe Inmediato	
Núm. de Folio de la Cédula Profesional	RFC	
Horario de Trabajo		
Hora de entrada/salida		

Vo Bo Dir. de la Dependencia

Certificó que el usuario es colaborador de esta Dependencia

Firma de Compromiso

Cantidad de Horas a utilizar 10 30 50 100

Tarifa por hora para Usuarios Internos Investigadores \$6.00. La Tarifa se paga por adelantado en cantidades mínimas de 10 horas. Al agotarse la cuota se suspenderá la clave. Todo servicio extra se paga aparte.



POLÍTICAS DE USO ACEPTABLE PARA RedUNAM

RedUNAM es una red de comunicación de datos que tiene como propósito principal servir en la transmisión e intercambio de información entre organizaciones académicas y de investigación, a servicios locales, nacionales e internacionales a través de conexiones con otras redes. Además puede dar servicio de apoyo a asuntos de gobierno locales, estatales, nacionales o internacionales; al desarrollo económico, al servicio pública empresas comerciales. La intención de esta política de uso aceptable es fomentar que el uso de la red sea consistente con los propósitos de RedUNAM. no de enunciar todos los posibles usos.

Se considera que la información y los recursos disponibles a través de RedUNAM son privados y sus dueños tienen derechos sobre ellos, a menos que renuncien explícitamente a ellos. Por lo tanto no es aceptable emplear RedUNAM para acceder información o recursos sin autorización de los dueños o tenedores de derechos sobre recursos o información.

No se permite el uso de RedUNAM para fines ilícitos.

No se permite molestar a usuarios, o interferir con el equipo o servicios de la red. Las molestias incluyen entre otras, la distribución de anuncios no solicitados, propagación de virus o gusanos (worm) informáticos, o cualquier otro uso que cause congestión en la red.

No se permite conexiones que creen patrones de ruteo que afecten la efectividad del uso compartido de la red.

No se permite el uso de la red para juegos cuando provoquen una carga excesiva sobre recursos escasos (por ejemplo, líneas conmutadas o privadas).

ACCIONES

- 1.- El centro de operación podrá en cualquier momento determinar si un uso particular es consistente o no con el propósito de RedUNAM.
- 2.- Cuando el centro de operación detecte un uso no aceptable, su personal notificará al usuario, al administrador responsable de la dependencia o de la red externa para que se tome acción para remediarlo inmediatamente. En una emergencia, se podrá desconectar temporalmente al usuario o red involucradas. El personal del centro de operaciones hará un esfuerzo para informar al responsable de la red o sistema antes de dicha desconexión, y se hará todo lo posible por restablecer la conexión en cuanto se considere que el uso no aceptable se ha suspendido

En general, corresponde a las otras redes externas fijar sus políticas de uso aceptable y los usuarios deberán conocerlas. En particular, el tráfico debe obedecer al uso aceptable de NSFNET (Red de la National Science Foundation). El tráfico sobre la NSFNET es primordialmente para investigación, desarrollo y docencia. El uso de la NSFNET para fines comerciales, como regla general, no es permitido, excepto si da soporte a la investigación, desarrollo o docencia. Sin embargo el tráfico comercial de RedUNAM no circula por NSFNET, y por lo tanto es aceptable.

RedUNAM no tiene responsabilidad por tráfico que envíe y que viole la política de uso aceptable a redes directamente o indirectamente conectadas. El responsable es el usuario que origine ese tráfico



**Dirección General de Servicios de Computo Académico
Departamento de Atención a Usuarios**

USUARIO EXTERNO

Solicitud de claves de acceso de RedUNAM

Datos del usuario		
Nombre	Apellido Paterno	Apellido Materno
Domicilio Particular		
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	Teléfono Particular
Fecha de Nacimiento	Lugar de Nacimiento	
mes/día/año		
Datos Laborales/Escolares		
Nombre de la Esc/Fac/Dependencia		
Domicilio de La		
Esc/Fac/Dependencia		
Calle	Núm.	Colonia
Delegación/Municipio	Código Postal	
Núm Telefonico	Fax	
Núm. de Folio de Cédula Profesional	RFC	
Nombre del Jefe Inmediato		
Horario de Clases/trabajo		
Hora de entrada/salida		

Vo Bo Dir. de la Dependencia

 Certificó que el usuario es colaborador de esta Dependencia

Firma de Compromiso

Cantidad de Horas a utilizar 10 30 50 100

--



POLÍTICAS DE USO ACEPTABLE PARA RedUNAM

RedUNAM es una red de comunicación de datos que tiene como propósito principal servir en la transmisión e intercambio de información entre organizaciones académicas y de investigación, a servicios locales, nacionales e internacionales a través de conexiones con otras redes. Además puede dar servicio de apoyo a asuntos de gobierno locales, estatales, nacionales o internacionales; al desarrollo económico, al servicio pública empresas comerciales. La intención de esta política de uso aceptable es fomentar que el uso de la red sea consistente con los propósitos de RedUNAM. no de enunciar todos los posibles usos.

Se considera que la información y los recursos disponibles a través de RedUNAM son privados y sus dueños tienen derechos sobre ellos, a menos que renuncien explícitamente a ellos. Por lo tanto no es aceptable emplear RedUNAM para acceder información o recursos sin autorización de los dueños o tenedores de derechos sobre recursos o información.

No se permite el uso de RedUNAM para fines ilícitos.

No se permite molestar a usuarios, o interferir con el equipo o servicios de la red. Las molestias incluyen entre otras, la distribución de anuncios no solicitados, propagación de virus o gusanos (worm) informáticos, o cualquier otro uso que cause congestión en la red.

No se permite conexiones que creen patrones de ruteo que afecten la efectividad del uso compartido de la red.

No se permite el uso de la red para juegos cuando provoquen una carga excesiva sobre recursos escasos (por ejemplo, líneas conmutadas o privadas).

ACCIONES

1.- El centro de operación podrá en cualquier momento determinar si un uso particular es consistente o no con el propósito de RedUNAM.

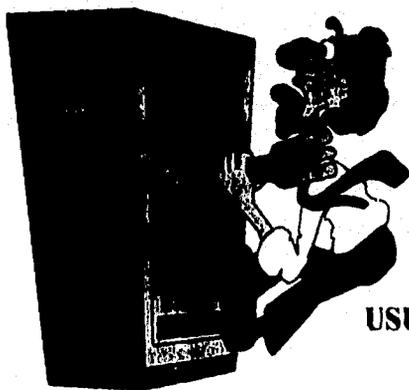
2.- Cuando el centro de operación detecte un uso no aceptable, su personal notificará al usuario, al administrador responsable de la dependencia o de la red externa para que se tome acción para remediarlo inmediatamente. En una emergencia, se podrá desconectar temporalmente al usuario o red involucradas. El personal del centro de operaciones hará un esfuerzo para informar al responsable de la red o sistema antes de dicha desconexión, y se hará todo lo posible por restablecer la conexión en cuanto se considere que el uso no aceptable se ha suspendido

En general, corresponde a las otras redes externas fijar sus políticas de uso aceptable y los usuarios deberán conocerlas. En particular, el tráfico debe obedecer al uso aceptable de NSFNET (Red de la *National Science Foundation*). El tráfico sobre la NSFNET es primordialmente para investigación, desarrollo y docencia. El uso de la NSFNET para fines comerciales, como regla general, no es permitido, excepto si da soporte a la investigación, desarrollo o docencia. Sin embargo el tráfico comercial de RedUNAM no circula por NSFNET, y por lo tanto es aceptable.

RedUNAM no tiene responsabilidad por tráfico que envíe y que viole la política de uso aceptable a redes directamente o indirectamente conectadas. El responsable es el usuario que origine ese tráfico

APÉNDICE F

PREGUNTAS AL USUARIO



USUARIO

**Preguntas de Usuario Interno Alumno**

- 1) ¿Cuál es tu nombre?
- 2) ¿Cuál es tu apellido paterno?
- 3) ¿Cuál es tu apellido materno?
- 4) ¿Cuál es tu fecha de nacimiento (mes/día/año)?
- 5) ¿Cuál es tu lugar de nacimiento?
- 6) Proporciona el nombre de la calle donde habitas
- 7) Proporciona el número de la casa donde habitas
- 8) ¿Cuál es el nombre de la colonia donde habitas?
- 9) Indica el Código Postal de la colonia donde vives
- 10) ¿Cuál es el número telefónico de tu domicilio?
- 11) Delegación o Municipio de donde vives.
- 12) Proporciona el nombre de la Escuela o Facultad donde estudias
- 13) Proporciona la Dirección completa de la Escuela o Facultad donde estudias
- 14) Indica que semestre o año que cursas actualmente
- 15) ¿Cuál es tu número de cuenta?
- 16) ¿Cuántas son las materias que cursas actualmente?
- 17) ¿Cuál es el año de ingreso a tu Facultad o Escuela?
- 18) ¿Cuál es tu promedio actual?

**Preguntas del Usuario Interno Profesor**

- 1) ¿Cuál es su nombre?
- 2) ¿Cuál es su apellido paterno?
- 3) ¿Cuál es su apellido materno?
- 4) ¿Cuál es su fecha de nacimiento (mes/día/año)?
- 5) ¿Cuál es su lugar de nacimiento?
- 6) Proporcione el nombre de la calle donde habita
- 7) Proporcione el número de la casa donde habita
- 8) ¿Cuál es el nombre de la colonia donde habita?
- 9) Delegación o Municipio donde vive.
- 10) Indique el Código Postal de la colonia donde vive
- 11) ¿Cuál es el número telefónico de su domicilio?
- 12) Proporcione el nombre de la Escuela o Facultad donde imparte clases
- 13) Proporcione la Dirección completa de la Escuela o Facultad donde imparte clases
- 14) ¿Cuál es su horario de trabajo (mencionando la hora de entrada y la de salida)?
- 15) ¿Cuál es el número de folio de su cédula profesional?
- 16) ¿Cuántas son las materias que imparte actualmente?

**Preguntas de Usuario Interno Investigador**

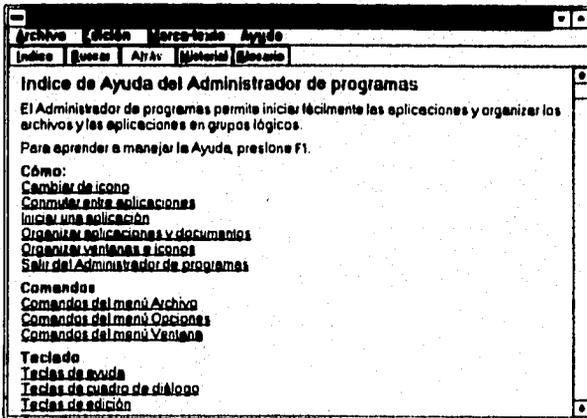
- 1) ¿Cuál es su nombre?
- 2) ¿Cuál es su apellido paterno?
- 3) ¿Cuál es su apellido materno?
- 4) ¿Cuál es su fecha de nacimiento (mes/día/año)?
- 5) ¿Cuál es su lugar de nacimiento?
- 6) Proporcione el nombre de la calle donde habita
- 7) Proporcione el número de la casa donde habita
- 8) ¿Cuál es el nombre de la colonia donde habita?
- 9) Delegación o Municipio donde vive.
- 10) Indique el Código Postal de la colonia donde vive
- 11) ¿Cuál es el número telefónico de su domicilio?
- 12) Proporcione el nombre de la Escuela, Facultad o Dependencia donde participa en algún proyecto.
- 13) ¿Cuál es el número de folio de su cédula profesional?
- 14) ¿Cuál es su horario de trabajo (mencionar únicamente hora de entrada y de salida)?
- 15) ¿Cuál es el nombre de su Jefe inmediato?

**Preguntas de Usuario Externo**

- 1) ¿Cuál es su nombre?
- 2) ¿Cuál es su apellido paterno?
- 3) ¿Cuál es su apellido materno?
- 4) ¿Cuál es su fecha de nacimiento (mes/día/año)?
- 5) ¿Cuál es su lugar de nacimiento?
- 6) Proporcione el nombre de la calle donde habita
- 7) Proporcione el número de la casa donde habita
- 8) ¿Cuál es el nombre de la colonia donde habita?
- 9) Delegación o Municipio de donde vive.
- 10) Indique el Código Postal de la colonia donde vive
- 11) ¿Cuál es el número telefónico de su domicilio?
- 12) Proporcione el nombre de la Escuela, Facultad, Dependencia o Empresa donde estudia o trabaja
- 13) Proporcione la dirección completa de la Escuela, Facultad, Dependencia o Empresa donde estudia o trabaja
- 14) En caso de trabajar, proporcione el número telefónico y de fax de la Empresa
- 15) En caso de trabajar: ¿Cuál es su horario de trabajo (mencionar únicamente hora de entrada y de salida)?

APÉNDICE G

AYUDA EN HIPERTEXTO





AÑADIENDO EL SISTEMA DE AYUDA A UNA APLICACIÓN.

Cuando las aplicaciones son sencillas, se puede proporcionar una ayuda en línea utilizando algunas cajas de diálogo para visualizar mensajes. Ahora cuando las aplicaciones empiezan a ser complicadas, el usuario espera tener asistencia desde un menú de ayuda o pulsando F1, al igual que en Windows o en aplicaciones comerciales para Windows.

Windows incluye una aplicación denominada *WinHelp* (WINHELP.EXE) como motor de ayuda Windows. Soporta múltiples fuentes, búsqueda por palabras clave y muchas otras características comunes en un sistema de ayuda.

Una aplicación Windows tiene que utilizar el motor de ayuda Windows para activar su sistema de ayuda. Esto requiere que la información de ayuda de una aplicación determinada sea compilada en un formato especial. Para ello, la versión profesional de Windows incluye un compilador de ayuda *Help Compiler para Windows 3.1* (HC31.EXE) que crea ficheros de ayuda en el formato requerido por WinHelp.

Creando un sistema de ayuda

Para crear un sistema de ayuda, es necesario tener instalada la versión profesional de Visual Basic, que proporciona el compilador de ayuda y adicionalmente necesita un procesador de textos que soporte ficheros *.rtf* (*Rich Text Format*). Después es necesario ejecutar un número de tareas que se especifican a continuación:

1. Reunir la información que agrupa los temas de ayuda.
2. Planificar el sistema de ayuda creando un esquema de su alcance y una lista de temas.
3. Escribir el texto correspondiente a los temas.
4. Introducir los códigos requeridos en los ficheros de texto para moverse por el sistema de ayuda.
5. Crear los gráficos si es necesario.



6. Crear los macros si es necesario.
7. Crear el fichero del proyecto de ayuda (*.hpi*) y compilar los ficheros fuente.
8. Depurar el sistema de ayuda.
9. Programar la aplicación para que acceda al sistema de ayuda.

A continuación se creará un sistema de ayuda para una aplicación.

Diseñando el sistema de ayuda.

En primer lugar, trataremos de reunir la información y planificar el sistema de ayuda. Para ello, determine:

- ¿ Quién va a utilizar la aplicación ? Personas sin experiencia, con alguna experiencia, con experiencia o expertos.
- ¿Cuál va a ser el contenido del sistema de ayuda ? Un sistema de ayuda contiene varias componentes: el menú de ayuda y los tópicos que aparecen en la pantalla de contenido.

Un menú de ayuda estándar puede estar compuesto por: *Contenido*, *Buscar ayuda sobre...* y *Acercas de...* Opcionalmente pueden añadirse un *Índice alfabético*, un *Índice de palabras clave* y *Cómo utilizar la ayuda*.

La *pantalla de contenido* aparece cuando el usuario pulsa F1 o selecciona la opción *Contenido* del menú de ayuda, e informa al usuario del contenido del sistema de ayuda. Desde esta pantalla, el usuario elige un tema y recibe ayuda directamente. Como normas generales esta pantalla debe agrupar sus entradas por categorías, evitar incluir más de 15 entradas, y limitar a tres el número de subniveles a los que el usuario debe acceder para visualizar ayuda sobre un tema. (ver figura 1.)

A continuación planificamos la estructura de los temas de ayuda. La tabla siguiente muestra en su columna *Título* el esquema del sistema de ayuda que pretendemos crear. En nuestro



caso, esta será la primera página del sistema de ayuda, desde la cual saltaremos al resto de las páginas que componen la ayuda total. Los sangrados indican la dependencia de las líneas. Una frase subrayada supone un salto a otra página y una frase sin subrayar visualizará una ventana flotante. Cada frase, tema, coincide con una página del documento que va a formar la ayuda. Cada página está identificada por los nombres de la columna *HelpContextID* del objeto de la aplicación que esté relacionado.

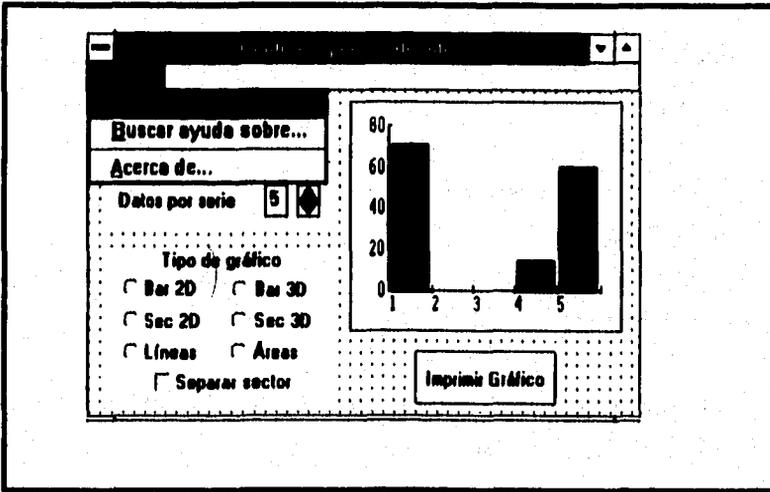


FIGURA 1. Menú Ayuda de la Aplicación ejemplo.

Según la tabla siguiente, la primera página del documento que contiene toda la información del sistema de ayuda está identificada por *ID_INDICE*, la siguiente página por *ID_APP*, y así sucesivamente hasta llegar a la última página, que está identificada por *ID_TFIN*. Según esto, cada tema del sistema de ayuda, que en el documento agrupa la información de ayuda se corresponde con una página, se identifica por una *cadena de contexto*, que debe ser única, incluyendo la primera página que visualiza el índice o contenido del sistema de ayuda.



Título	ID	Cod	Función
(Principio de la primera página)	ID_INDICE	101	Indice
Introducción		0	
La aplicación "controles personalizados"	ID_APP		Aplicación
	ID_CGRAFICO	110	"Ctrls. pers."
control gráfico	ID_MARCO	0	Control gráfico
marco	ID_CCONTEO	111	Control marco
controles de		0	Control de
conteo		112	conteo
		0	
		113	
		0	
Como....			
Seleccionar la cantidad de datos	ID_DATOS	120	Datos a
Número de series de datos	ID_NSERIES	0	representar
Número de datos por serie	ID_NDATOS	121	Datos a
Seleccionar el tipo de gráfico	ID_TIPO	0	representar
Actualizar el gráfico	ID_ACTUALI	122	Datos a
Imprimir el gráfico	ID_IMPRIMIR	0	representar
		125	Tipo de gráfico
		0	Actualizar el
		130	gráfico
		0	Imprimir el
		135	gráfico.
		0	
Órdenes			
Orden para finalizar la aplicación	ID_OFIN	140	Finalizar la
		0	aplicación



Teclado			
Teclas para finalizar la aplicación	ID_TFIN	145 0	Finalizar la aplicación.

La columna de *Palabra clave* es la que aparecerá, según muestra la figura siguiente, en la ventana visualizada por el botón *Buscar* del sistema de ayuda.

Para asignar una cadena de contexto a un tema de ayuda, suponiendo que ha iniciado un nuevo documento en su procesador de palabras (Word, por ejemplo).

1. Coloque el punto de inserción en la primera posición de la página y ejecute la orden de insertar notas al pie. Con Microsoft Word al ejecutar la orden de *Notas al pie...* del menú *insertar* se visualizará la ventana de la figura 2.

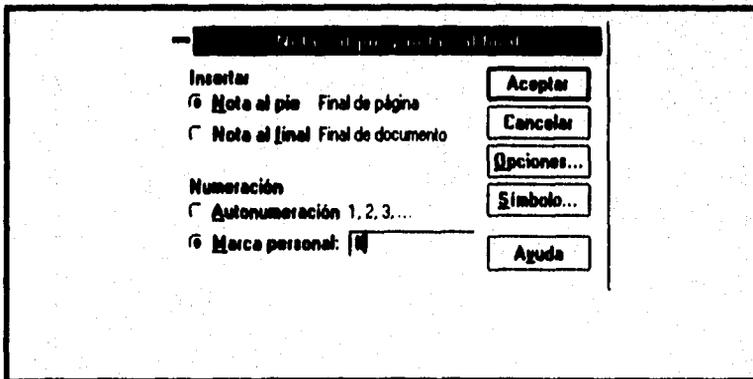


FIGURA 2. Personalizar Notas al Pie.

2. Inserte el carácter # (sup) como marca de notas al pie. Observe que aparece como superíndice.



3. Escriba la cadena de *contexto* (el identificador del tema) como nota al pie. El resultado que obtenemos con Microsoft Word, después de haber introducido dos notas al pie más que ahora explicaremos, se muestra a continuación. (Ver figura 3)

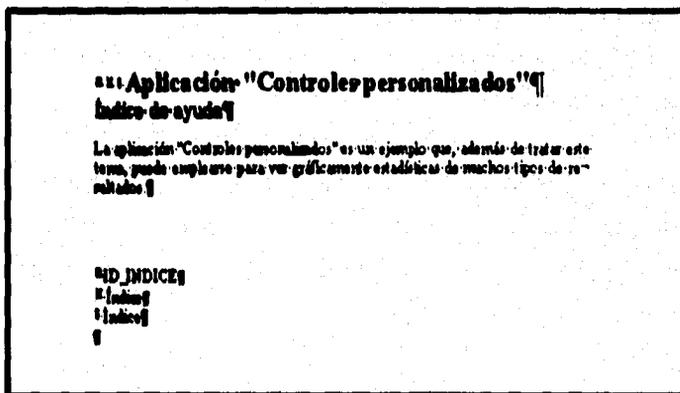


FIGURA 3. Archivo .RTF

Asegúrese de que sólo haya un espacio en blanco entre el carácter # y la cadena de contexto.

Con la aplicación *WinHelp*, el usuario puede buscar un tema específico utilizando palabras clave. Si observa la ventana *Buscar* mostrada anteriormente, hay una primera lista correspondiente a las palabras clave a las que estamos haciendo referencia, y hay una segunda lista que se corresponde con los temas relacionados con esa palabra clave. La primera lista, la de las palabras clave, se forma a partir de las notas al pie marcadas con K^(h) y la segunda lista de temas marcados, con las notas al pie marcadas con S^(h).

Para añadir una nota al pie marcada con K o con S, proceda de la misma forma que se indicó para añadir una nota al pie marcada con #.

La tabla siguiente presenta dos páginas cualesquiera del documento que contiene la información del sistema de ayuda. En ellas se observa:



- El encabezado formado por los caracteres # (cadena de contexto), K (palabra clave) y \$ (tema relacionado con la palabra clave), cuyos valores asignados están especificados como notas al pie.
- Identificadores en mayúsculas, por ejemplo ID_NSERIES. Son las cadenas de contexto (identificadores de temas) y son *caracteres ocultos* en mayúsculas o minúsculas (cuando se visualiza el texto, no aparecen).
- Palabras con subrayado doble. Indican un salto al tema (página) identificado por la cadena de contexto escrito en texto oculto justamente a continuación. Estas palabras aparecerán en el sistema de ayuda en otro color y subrayadas con una línea continua. Si se hace clic sobre ellas, se produce un salto al tema especificado.
- Palabras con subrayado simple. Indican un salto al tema (página) identificado por la cadena de contexto escrita en texto oculto justamente a continuación. Estas palabras aparecerán en el sistema de ayuda en otro color y subrayadas con una línea discontinua. Si se hace clic sobre ellas, se visualizará una ventana flotante conteniendo la información correspondiente al tema especificado.
- Notas al pie correspondientes a los caracteres especiales que encabezan cada una de las páginas.

#k\$

Seleccionando la cantidad de datos

Un conjunto de datos puede estar formado por una o más series de datos, cada una de las cuales agrupa un número determinado de datos.

Número de series de datosID_NSERIES

Número de datos por serieID_NDATOS

FIGURA 4a. Documento Word con Información del Sistema de Ayuda

**#k\$****Número de series de datos**

Permite especificar cuántas series de datos se van a representar. El mínimo número de series es 1.

Para especificar el número de series:

1. Utilizando el ratón, haga clic en el **control de conteo** ID_CONTEO hasta visualizar en la caja de texto asociada al mismo número deseado.
2. Situése en la caja de texto y escriba directamente el número deseado.

Tema relacionado.

Notas al pie correspondientes a las dos páginas anteriores**# ID_DATOS**

K Datos a representar

\$ Series y datos por serie

ID_NSERIES

K Datos a representar

FIGURA 4b. Documento Word con Información del Sistema de Ayuda

Una vez que ha escrito toda la información correspondiente al sistema de ayuda de su aplicación, guarde el fichero con formato RTF. En nuestro caso, el fichero ha sido guardado con el nombre *ctrlsper.rtf*. Para hacer esto en Microsoft Word ejecute la orden de *Guardar como...* del menú Archivo, y en la ventana que se visualiza, elija en la lista Guardar archivo como: la opción Texto con formato (*.rtf).



Compilando el fichero de ayuda

Para controlar cómo el compilador de ayuda (HC31.EXE) crea el fichero de ayuda (.hlp) a partir de los ficheros .rtf, podemos crear un fichero proyecto de ayuda con extensión .hpj (*help project*). Para crear este fichero, se puede utilizar cualquier editor, pero el fichero se debe guardar en ASCII (sólo texto sin formato). Por ejemplo, para nuestra aplicación Controles personalizados hemos creado el proyecto de ayuda *ctrlsper.hpj* que se muestra a continuación :

```
; Sistema de ayuda para Controles personalizados
; Opciones utilizadas
[OPTIONS]
```

```
COMPRESS = NO
TITLE = Controles Personalizados
CONTENTS = ID_INDICE
```

```
; Identificadores de contexto y números de contexto asociados
```

```
[MAP]
```

```
ID_INDICE      1010
ID_APP         1100
ID_CGRAFICO   1110
ID_MARCO      1120
ID_CCONTEO    1130
ID_DATOS      1200
ID_NSERIES    1210
ID_NDATOS     1220
ID_TIPO       1250
ID_ACTUALI    1300
ID_IMPRIMIR   1350
ID_OFIN       1400
ID_TFIN       1450
```

```
; Ficheros utilizados para construir el sistema de ayuda
```

```
[FILES]
```

```
C'CTRLSPER.RTF
```



Observe que un fichero de este tipo está dividido en secciones, en nuestro caso en tres: [OPTIONS], [MAP] y [FILES]. Además de éstas existen otras secciones, como [BUILDTAGS], [CONFIG], [BITMAPS], [ALIAS], [WINDOWS] y [BAGGAGE].

La sección [OPTIONS] especifica las opciones que controlan cómo se construye el fichero de ayuda. Por ejemplo, COMPRESS indica el tipo de compresión a aplicar durante la construcción (NO, MEDIUM o HIGH), TITLE indica el título de la ventana de ayuda y CONTENTS identifica la cadena de contexto que referencia el tema (normalmente una tabla de contenido o un índice) que se visualiza cuando se hace clic en el botón índice de la ventana de ayuda. Si no se especifica, se asume como tema primero del primer fichero especificado en la sección [FILES]. La sección [OPTIONS] es opcional, pero cuando se especifica tiene que ser la primera.

La sección [MAP] asocia las cadenas de contexto con números de contexto. Esta sección es opcional.

La sección [FILES] especifica los ficheros que proporcionan la información para construir el fichero de ayuda. Esta sección es necesaria.

Una vez construido el proyecto de ayuda, el fichero de ayuda se construye invocando al compilador de ayuda de la forma siguiente:

HC31 CTRLSPEC.HPJ

Si es necesario especificar las rutas de acceso para cada uno de los ficheros anteriores, hágalo.



Añadiendo el sistema de ayuda a la aplicación

Para especificar el fichero de ayuda que una aplicación invocará cuando el usuario pulse *F1*, se puede proceder de dos formas:

- Durante el diseño, ejecutando la orden **Project** del menú **Options** y asignando el nombre del fichero de ayuda a la opción **Help File**.
- Durante la ejecución, asignando el nombre del fichero de ayuda a la propiedad **Help File** del objeto **App**.

Siguiendo este último criterio, modifique el procedimiento *Form_load* de la forma que se indica a continuación:

Sub Form_Load ()

```
'Hacer que el directorio donde se ejecuta la
'aplicación sea el directorio de trabajo
ChDir App.Path
'Especificar el fichero de ayuda
App.HelpFile = ctrlspcr.hlp
'Inicializar el generador de números aleatorios
Randomize
'Visualizar el primer diagrama
Graph1_click
```

End Sub

Este procedimiento espera que el fichero de ayuda (*.hlp*) se encuentre en el directorio donde se ejecuta la aplicación.

Para casos donde el fichero de ayuda se ubique en un determinado directorio y la aplicación en cualquier otro directorio. La solución sería construir un fichero de inicialización (*.ini*) que especifique la ruta y el nombre del fichero de ayuda, para que la aplicación pueda leerlo cuando arranque. Para este tipo de operaciones la **API** de Windows provee de dos funciones, denominadas *WritePrivateProfileString* y *GetProfileString*.



La propiedad `helpcontextid`

La forma más fácil de hacer que una aplicación Visual Basic utilice la ayuda es asignando a la propiedad `HelpContextID` (que tiene los formularios, los menús y sus ordenes, y casi todos los controles) un valor de los especificados en la sección [MAP] del fichero `.hlp` se encuentra relacionado. Por ejemplo, en nuestra aplicación, utilizando la ventana de propiedades, hemos asignado a la propiedad `HelpContextID` de:

<i>Graph1</i> (control gráfico)	1300
<i>Nseries</i> (caja de texto)	1210
<i>Ndatos</i> (caja de texto)	1220
botones de opción	1250
<i>ImprimirGráfico</i> (botón de pulsación)	1350
<i>Contenido y</i>	
<i>Buscar ayuda sobre</i> (menú Ayuda)	1010

Esto hace que el pulsar *F1* (ayuda) se visualice el tema referenciado por el valor de `HelpContextID` del objeto que esté enfocado. Por esta razón, no utilice en un menú o en una de sus ordenes *F1* como valor para la propiedad `ShortCut`.

Función `winhelp`

Además de `HelpContextID`, otra forma de hacer que una aplicación utilice la ayuda es invocando la función `WinHelp` de la API de Windows. Esta función puede ser invocada especialmente para solicitar ayuda a través de un fichero diferente al especificado por la propiedad `HelpFile`.

La declaración de la función `WinHelp` es la siguiente:

```
Declare Function WinHelp Lib User (
    ByVal hWnd As Integer,
    ByVal lpHelpFile As String
    ByVal wCommand As Integer,
    dwData As Any) As Integer
```



El parámetro *hWnd* identifica la ventana desde la que se solicita la ayuda. En Visual Basic se corresponde con el valor de la propiedad *hWnd* del formulario.

El parámetro *lpHelpFile* representa la ruta y el nombre del fichero de ayuda.

El parámetro *wCommand* especifica la acción a tomar por WinHelp. Puede ser uno de los siguientes valores (Especificados en *CONSTANT.TXT*):

El parámetro *dwData* representa el tema sobre el que la aplicación solicita ayuda. Su valor *Long* indicando un valor de contexto o *String* indicando una palabra clave, depende del valor del parámetro *wCommand*.

Valor	Significado
HELP_CONTEXT = &H1	Ayuda para un tema identificado por un número de contexto.
HELP_QUIT = &H2	Salir de la ayuda
HELP_CONTENTS = &H3 o HELP_INDEX = &H3	Visualizar el tema de ayuda especificado como índice (CONTENTS).
HELP_HELPOHELP = &H4	Carga <i>winhelp.hlp</i> y visualiza el tema "Utilizando la Ayuda".
HELP_SETCONTENTS = &H5 o HELP_SETINDEX = &H5	Establece el índice de ayuda para un fichero que contiene más de uno. El índice es identificado colocando el número de contexto en <i>dwData</i> . A continuación de una llamada con esta opción, tiene que hacerse otra con <i>HELP_CONTEXT</i> .
HELP_CONTEXTPOPUP = &H8 HELP_FORCEFILE = &H9 HELP_KEY = &H101	Ayuda para la palabra clave especificada por <i>dwData</i> .
HELP_COMMAND = &H102 HELP_PARTIALKEY = &H105	Visualizar la ventana <i>Buscar</i> .



Menú ayuda

Aunque no es necesario programar *F1*, si hay que añadir código para acceder al sistema de ayuda a través del menú *Ayuda*.

En primer lugar, escriba las siguientes declaraciones en un módulo que vamos a denominar *ctrlsper.bas*:

```
Global Const HELP_CONTEXT = &H1
Global Const HELP_QUIT = &H2
Global Const HELP_INDEX = &H3
Global Const HELP_HELPOHELP = &H4
Global Const HELP_SETINDEX = &H5
Global Const HELP_CONTEXTPOPUP = &H8
Global Const HELP_FORCEFILE = &H9
Global Const HELP_COMMAND = &H102
Global Const HELP_PARTIALKEY = &H105
Declare Function WinHelp User (
    ByVal hWnd As Integer,
    ByVal lpHelpFile As Integer,
    ByVal wCommand As Integer,
    dwData As Any) As Integer
```

Cuando el usuario haga clic en la orden *Contenido...* del menú *Ayuda*, tiene que visualizarse el tema referenciado por *ID_INDICE*. Para ello, escriba el siguiente procedimiento:

```
Sub AyudaContenido_Click ()
    Dim As Integer
    vr = WinHelp(hWnd, (App.HelpFile), HELP_INDEX, CInt(0))
End Sub
```

Cuando *wCommand* es *HELP_INDEX*, el valor *dwData* se ignora

Cuando el usuario haga clic en la orden *Buscar ayuda sobre...* del menú *Ayuda*, tiene que visualizarse la ventana *Buscar* del sistema de ayuda. Para ello, escriba el siguiente procedimiento:



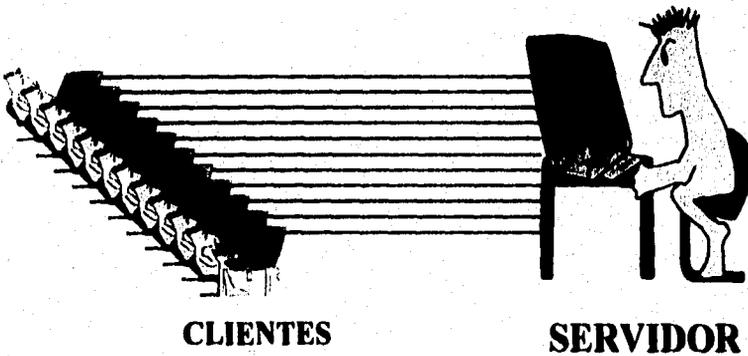
```
Sub AyudaBuscar_Click ()  
    Dim vr As Integer  
    vr = WinHelp (hWnd, (App.HelpFile), HELP_INDEX, Clng(0))  
End Sub
```

Cuando el usuario haga clic en la orden *Acerca de...* del menú *Ayuda*, tiene que visualizarse, por ejemplo, la ventana *información del sistema* vista en la siguiente imagen. Para ello añade a la aplicación el fichero *recurdlg.frm* correspondiente a la ventana mencionada y escriba el siguiente procedimiento para visualizarla:

```
Sub AyudaAcercaDe_Click ()  
    Infos.show  
End Sub
```

APÉNDICE H

ARQUITECTURA CLIENTE-SERVIDOR



ARQUITECTURA CLIENTE/SERVIDOR

Introducción

En este módulo es necesario para cualquier empresa tomar ventaja de cada oportunidad para reducir costos, aumentar la calidad y ofrecer un mejor servicio. Resulta indispensable contar con medios que permiten tener y generar a tiempo toda la información necesaria para tomar la mejor decisión. El uso de las computadoras en las diferentes ramas de la industria fue un hecho que incrementó de manera extraordinaria la eficiencia y rapidez en diversas actividades.

En los últimos años, la implementación y mejoramiento de los sistemas de comunicación entre los equipos de cómputo, como es el caso del desarrollo de las redes de computadoras, han permitido que se tenga disponible una gran cantidad de información el cual es una característica esencial en el proceso de toma de decisiones, por lo que se desarrollan implementaciones para controlar y permitir el acceso a la información.

Uno de los modelos de cómputo que más revuelo han producido en estos últimos años es el denominado *modelo de cómputo cliente/servidor* el cual es aplicable a los más diversos campos como bases de datos, redes, sistemas y hardware.

El *Cliente* es algo que requiere que se lleve a cabo una tarea, y un *Servidor* por el otro lado, es aquello encargado de realizar la tarea requerida. En términos de cómputo, un servidor es un programa que responde a las peticiones de otra máquina a través de lo que se conoce como *SRPI (Server Requester Programming Interface)*. A diferencia del Cliente, el Servidor debe existir y debe ser buscado. Este primer concepto de cliente y servidor no difiere mucho del teleproceso, sin embargo como se desarrollará más adelante, las características de la arquitectura Cliente/Servidor se oponen a las del modelo de teleproceso o modelo de cómputo basado en mainframes.



Desarrollo

Durante muchos años, el procesamiento de datos consistió en aplicaciones relativamente sencillas orientadas a transacciones, diseñadas para automatizar aplicaciones repetitivas en los negocios tales como manejo del libro de contabilidad, envío de estados de cuenta, etc. Programas en modo batch o en línea corrían en mainframes y manejaban este tipo de tareas fácilmente.

Este modelo de cómputo basado en la utilización de mainframes presenta las siguientes características:

- Procesamiento centralizado y compartido por todos los usuarios.
- Interacción con terminales. La terminal solo transporta las entradas hacia el host y despliega lo que envía como respuesta.
- Administración y mantenimiento complicado realizado por un pequeño grupo de expertos.

Es importante destacar que lo que denominamos procesamiento en mainframes no está necesariamente ligado a una plataforma de hardware en especial. Este puede ser realizado en cualquier computadora, lo que importa aquí es la manera en la cual se realizó el diseño de la aplicación.

Sin embargo a principios de 1980 la manera en que la gente empezó a usar computadoras cambio dramáticamente. Cada día fue mayor el número de usuarios que, por la naturaleza de sus actividades, eran más bien considerados como el usuario final, estamos hablando de analistas financieros, de mercado, gerentes, vendedores, etc.

Para este nuevo tipo de usuarios se creó un nuevo mercado en el que se desarrollaron equipos como la PC y aplicaciones sumamente amigables y funcionales. Para aquellos que requerían trabajar con los sistemas en mainframes les resultaba difícil el trabajo en terminales, por lo que se demandó el tener mejores interfaces de trabajo. Además, el aumento de complejidad e interacción con el usuario de los sistemas de transacciones tradicionales también



generaron la necesidad de un sistema que, además de mejorar el ambiente de trabajo, hiciera las transacciones de manera más eficiente y rápida.

La arquitectura Cliente/Servidor es un modelo de cómputo en red en el cual ciertas funciones requeridas por clientes, generalmente estaciones de trabajo individuales dentro de una LAN, son atendidas por los procesadores más adecuados dentro de la red, a los cuales denominamos **servidores**. El término **servidor** aparece en muchos lugares y bajo varios contextos; podemos hablar de:

- **Servidores de Bases de Datos** (ver figura 1).

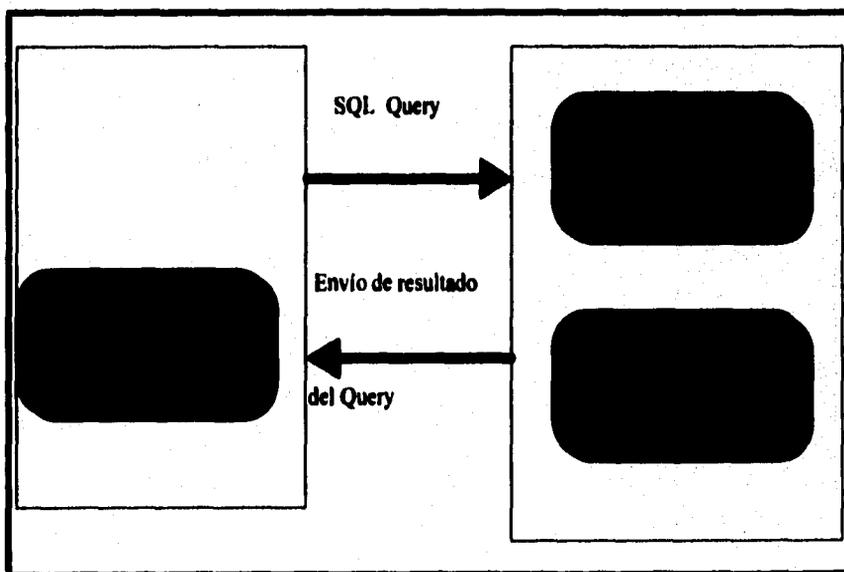


FIG.1 Servidores de Bases de Datos



- Servidores de archivos.(ver figura 2).

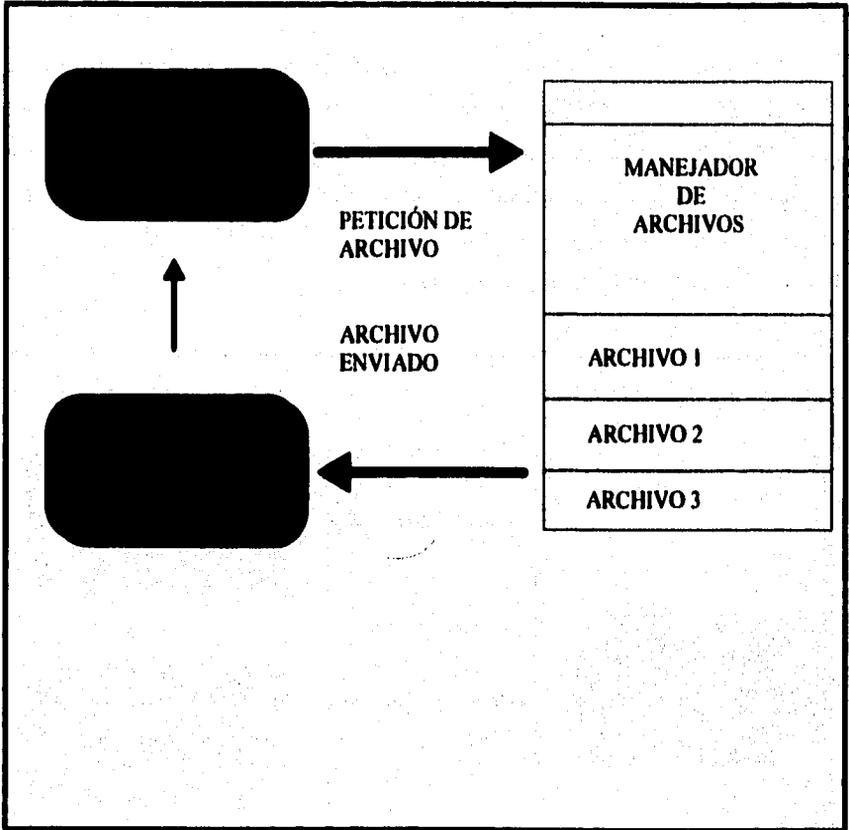


Figura 2. Servidor de Archivos

- Servidores de impresión.(ver figura 3)

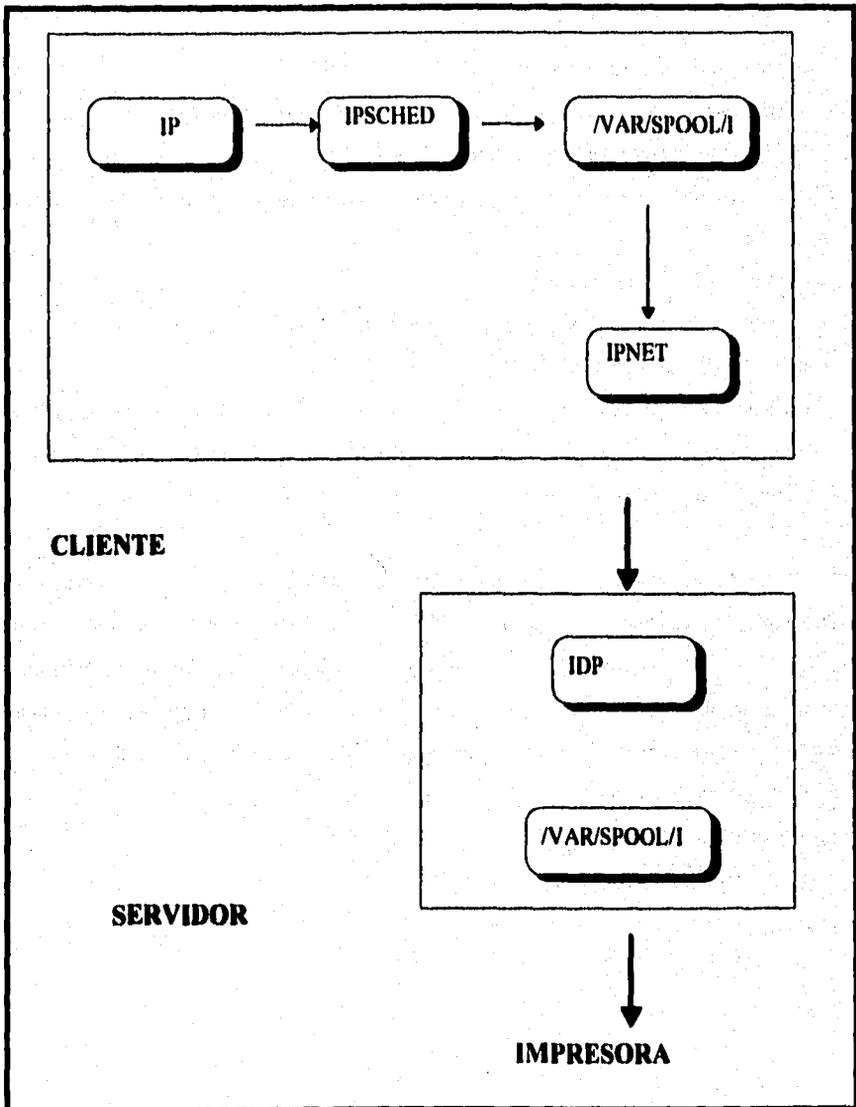


FIG 2 Servidores de Impresión



Como habíamos mencionado, esta arquitectura no sólo se refiere al hecho de tener clientes y servidores, para poder llevar a cabo una implementación cliente/servidor son necesarias una serie de consideraciones las cuales se enumeran a continuación:

- Una **interface gráfica amigable** residente en el lado del cliente
- La **división adecuada de las actividades de la aplicación entre el cliente y el servidor**
- Que el sistema cliente/servidor tenga **capacidades de red**
- Cliente y servidor pueden trabajar en plataformas separadas, pero podrían hacerlo en una sola plataforma.
- El cliente es quien inicia las acciones.
- El servidor de la base de datos es quien provee la protección de datos, seguridad, respaldo, recuperación y capacidad de SQL.
- Si el cliente o servidor necesitan ser actualizados, no debe ser necesario actualizar ambos. Si la plataforma de hardware del cliente llega a ser muy poderosa, el desempeño del servidor puede convertirse en un cuello de botella.

Uno de los puntos fundamentales de esta arquitectura y el que lo hace diferente del teleproceso es la ejecución de ciertas partes de la aplicación en el cliente y otras en el servidor. Una buena implementación de ésta depende del tipo de actividades que se decida que se lleven a cabo en el mainframe y servidores de archivos, así como las variedades de arquitecturas cliente/servidor vamos a ver las partes en que podemos dividir una aplicación (ver figura 4):

- **Servicios de Presentación**
- **Lógica de Presentación**
- **Lógica de trabajo o aplicaciones**
- **Lógica de datos.**
- **Servicio de datos.**
- **Servicio de archivos.**

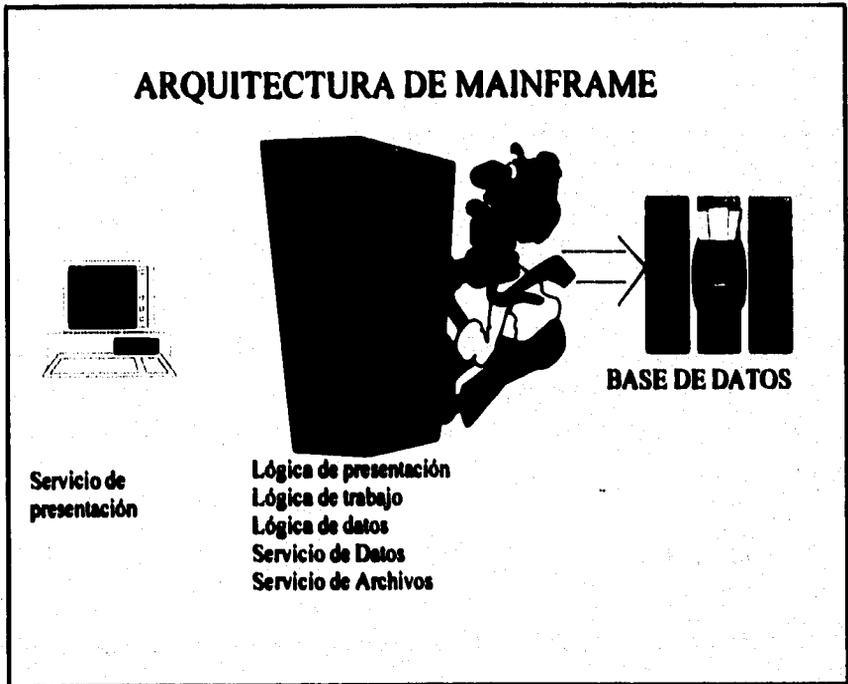


FIG. 4 *Arquitectura de Mainframe*

Servicio de Presentación

Es provisto por un dispositivo que acepta entradas del usuario y despliega aquello que la lógica de presentación le indique.

Lógica de Presentación

Es la encargada de controlar actividades como la navegación entre formas, validación preliminar de datos, etc.; podemos decir que si controla la interacción entre el usuario y la computadora. Se encarga también del manejo de objetos en las ventanas y de los errores de presentación

Lógica de trabajo o aplicaciones

Es un conjunto de decisiones, cálculos y operaciones que la aplicación debe llevar a cabo. Maneja cosas tales como la decisión de aceptar una orden, la evaluación de un préstamo, etc.

Lógica de datos.

Es la expresión de las operaciones a ser realizadas en las bases de datos que son necesarias para llevar a cabo lo indicado por la lógica de aplicación. Por ejemplo para una base de datos relacional, estas expresiones son instrucciones SQL tales como "select", "update", "insert", etc.

Servicio de datos.

Son las acciones que el DBMS o una aplicación cualquiera toma al llevar a cabo la lógica de datos incluyendo manipulación de datos, definición de datos, etc.

Servicio de archivos.

Accesa el disco y recupera la información buscada por el DBMS o la aplicación. El servicio de archivos es llamado a cabo por funciones del S.O.

A continuación vemos una perspectiva de los diferentes modelos que han surgido con base en las divisiones que se han hecho de las tareas anteriormente descritas. Esta división se refiere a que tipo de actividades se van a efectuar en cada uno de los lados del modelo cliente/servidor.

En ambientes que utilizan mainframes, la terminal provee los servicios de presentación mientras que el host provee las funciones restantes. En este esquema existen dos serios



problemas. El primero de ellos es que la interface de usuario esta limitada al estilo de interacción de una terminal. Aunque no es imposible, si es bastante difícil tener una interface gráfica eficiente debido a que todo el procesamiento reside en un *host* central y la gran carga que implicaría así como su transporte por la red son puntos a considerar.



FIG. 5 Servidor de Archivos

En segundo lugar, por cada usuario adicional y por cada aplicación se genera una carga substancial al mainframe. Esto resulta en una pérdida de escalabilidad, lo que significa que el *host* está limitado en el número de usuarios y aplicaciones que puede manejar. Inclusive la carga de red es mayor porque no sólo envía datos sino también las instrucciones que permiten desplegar los datos en la pantalla.



La **arquitectura de servidor de archivos** (ver figura 5) soluciona estos problemas pero introduce sus propias fallas. Al dividir los componentes de la aplicación, pasa todo a excepción de servicio de archivos a la PC. Los clientes manejan toda la presentación, aplicación y funciones de Bases de Datos. El servidor de archivos sólo recupera los archivos necesarios y los pasa hacia el cliente.

Sin ningún retraso, ocasionado por el transporte a través de la red, entre la lógica y los servicios de presentación, es más fácil desarrollar un buen GUI para una aplicación. Y debido a que el servidor tan sólo recupera archivos el hecho de agregar usuarios adicionales y aplicaciones sólo ocasiona un pequeño incremento en la carga del CPU. Además como cada usuario tiene su propio procesador resulta en un mayor poder de procesamiento de la red en general.

Sin embargo la arquitectura tiene dos problemas: la **demandada de procesamiento** a la computadora local son muy altas y en segundo lugar es el **gran incremento de la carga de red**. Por ejemplo supongamos que tengo una base con las siguientes tablas: clientes, ordenes, parte y pedido. Una petición razonable sería la lista de cada cliente con cada parte que el cliente ordenó y el valor total de cada parte ordenada. Esto requiere el movimiento del contenido de las cuatro tablas completas sin importar cuantos datos estén involucrados en realidad.

La **arquitectura cliente/servidor** está diseñada para encarar estos problemas al separar los componentes de una aplicación y colocarlos donde éstos sean más eficientes. Debido a que existen diferentes formas de realizar esta división es necesario entender las diferentes arquitecturas y para qué clase de aplicación es apropiada.

***Two-tiered Client/Server* o Arquitectura de dos enlaces.**

La gran mayoría de configuraciones cliente/servidor usan un modelo *two-tiered* o de dos enlaces. Como mínimo una arquitectura Cliente/servidor supone que ambos, los servicios de presentación y la lógica de presentación residen en el cliente. El resto de los servicios se colocan



del lado del cliente o del servidor de acuerdo al diseño del modelo. La decisión depende del tipo de aplicación y sus requerimientos.

La primera posibilidad es colocar sólo los servicios de datos y de archivos en el servidor y el servicio de presentación la lógica de presentación y aplicación y la lógica de datos en el cliente. Este camino es llamado remote DBMS.

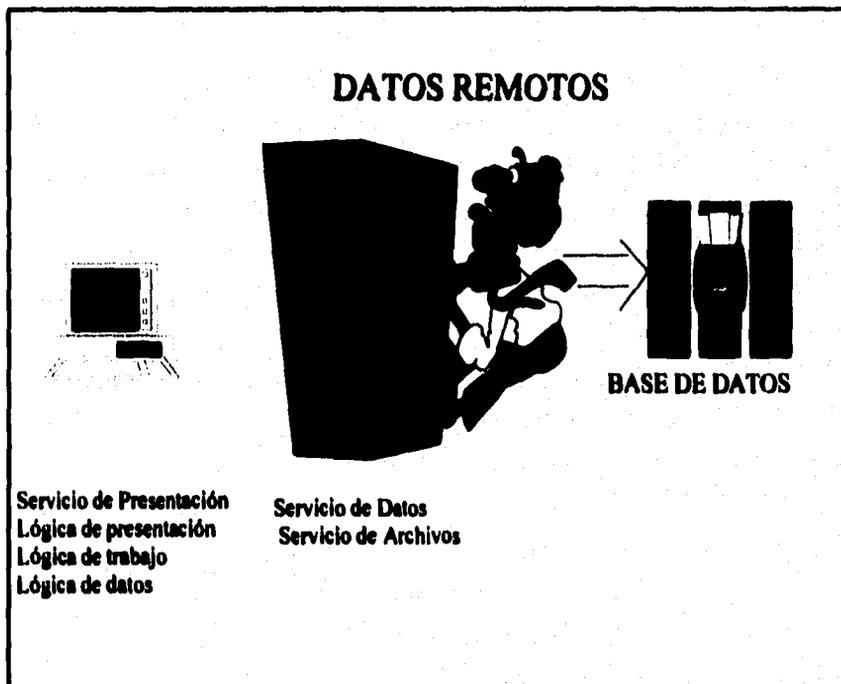


FIG.6 Datos Remotos

Como coloca el menor número de demandas en el servidor, el modelo de DBMS remote provee la mejor escalabilidad. Sin embargo, para aplicaciones complejas que involucren gran interacción con la base de datos, puede llevarnos una gran carga en el procesamiento del cliente y en el ancho de banda de la red. El resultado completo de las instrucciones SQL deben regresar al cliente para su procesamiento porque la lógica de decisión se encuentra en el cliente. Este modelo



complica la administración porque todo el código de la aplicación reside en cada cliente. Cuando se modifica el código éste se debe modificar en todos los clientes.

Se puede reducir la carga en el cliente y en la red al mover parte de la lógica de trabajo al servidor (**el modelo de división de lógica**). Dando un paso adelante, se puede mover toda la parte de lógica de trabajo al servidor, dejando sólo los servicios de presentación y lógica (**modelo de presentación remota**)

El decidir entre estos tres modelos, el diseñador del sistema debe determinar la mejor localización para la lógica de trabajo. Generalmente no es una buena idea permitir que las restricciones de hardware dominen la planeación. El ahorro en el desarrollo, operación y administración de un diseño apropiado son mayores que el costo de mudarse hacia servidores y clientes más poderosos.

Fallas del modelo *Two-Tiered*

Esta arquitectura presenta algunos problemas en aplicaciones complejas con muchos clientes, lógica intrincada, bases de datos heterogéneas y entradas heterogéneas en las bases. Los problemas que se presentan en esta arquitectura son:

- La carga de administración aumenta con un mayor número de usuarios.
- El uso de procedimiento almacenado en una lógica compleja.
- Problemas con la utilización de programas *batch*.
- Problemas con la integridad de la base bajo diferentes DBMS.

Arquitectura de Tres Enlaces (*Three-Tiered* Arquitectura)

En esta arquitectura (ver figura 7) el cliente se dedica a la lógica de presentación y los servicios y tiene un **API** (*Application Program Interface*) para invocar la aplicación en una capa intermedia. El servidor es dedicado a los servicios de datos y de archivos (los cuales se pueden optimizar sin riesgo de usar procedimientos almacenados). La capa intermedia es un servidor de



aplicaciones en el cual la lógica de trabajo se ejecuta y desde la cual la lógica de datos invoca los servicios de datos.

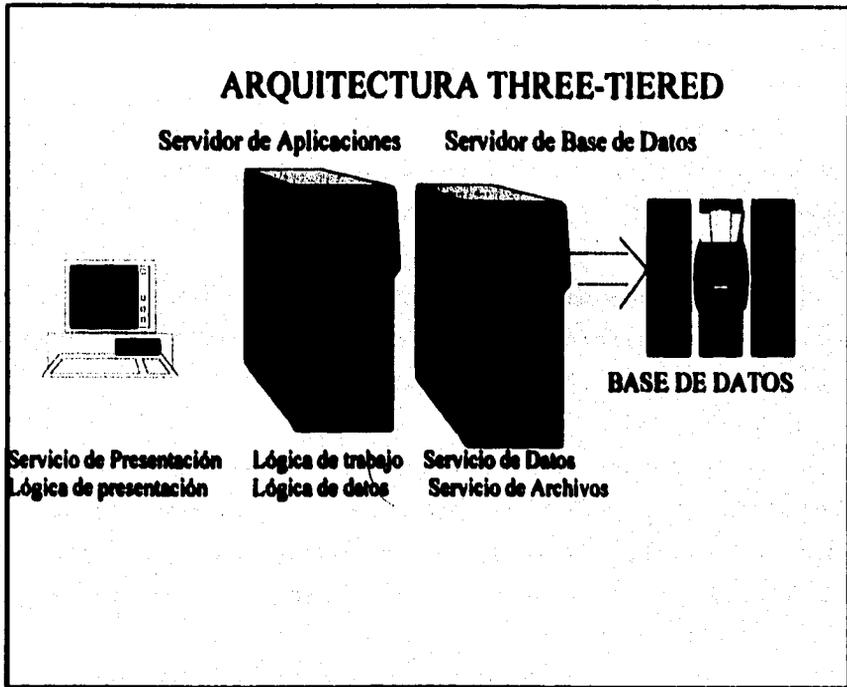


FIG. 7 Arquitectura Three-Tiered

El servidor de aplicaciones puede manejar las transacciones y asegurar la integridad de la base de datos distribuida al utilizar un proceso denominado *two phase commit*. El servidor de aplicaciones también puede manejar colas asíncronas para asegurar la confiabilidad de transacciones completas. También centraliza la lógica de la aplicación para facilitar la administración y proveer recursos por nombres y no por localización.



COMPONENTES DE LAS APLICACIONES CLIENTE/SERVIDOR

El Cliente es la estación de trabajo personal, es decir, una estación de trabajo como una Apple, IBM PS/2, NCD X-Terminal, SparcStation Sun, etc., pueden ser en algún momento clientes. No existe una tecnología específica para el cliente.

Durante los últimos 10 años, el desempeño de las estaciones ha crecido dramáticamente. Este crecimiento en poder permite que las aplicaciones más sofisticadas puedan correr desde el equipo personal.

Las comunicaciones y la velocidad de las redes han crecido de igual manera en los últimos tiempos. Con el advenimiento de redes de áreas LAN y WAN de alta velocidad y la fibra óptica, el desempeño y la confiabilidad han crecido substancialmente.

Las estaciones cliente pueden utilizar ambientes como DOS, Windows, Windows NT, OS/2, o UNIX. El Cliente por lo general provee funciones para producir localmente aplicaciones, cuando está conectado en red puede acceder los servicios provistos por el sistema operativo de red. El cliente puede ser usado como terminal para acceder aplicaciones residentes en otro *host*.

IMPLEMENTACIÓN DE LA ARQUITECTURA CLIENTE/SERVIDOR.

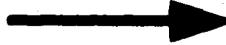
Un ambiente cliente/servidor tiene varios componentes. Los principales servicios son presentación, cliente distribución, servidor y base de datos. Las tareas realizadas por cada uno se presentan a continuación:



COMPONENTES DE ARQUITECTURA

FUNCIÓN

SERVICIOS DE PRESENTACIÓN



CLIENTE

SERVICIOS DE INTERACCIÓN

SERVICIOS DE MANEJO DE PANTALLA

SERVICIOS DE MANEJO DE DATOS

- EXTENSIONES.
 - Validación numérica
 - Fechas
- INTERACCIÓN CON EL USUARIO.
 - Teclado
 - Manipulación directa.
 - Animación
- MODELO DE MEMORIA.
 - Estructura de ventana
 - Facilidad de uso

COMPONENTES DE ARQUITECTURA

FUNCIÓN

SERVICIOS DE REGISTRO

CLIENTE

SERVICIOS DE DISTRIBUCIÓN

SERVIDOR

SERVICIOS DE BASE DE DATOS



- CAPTURA DE DATOS Y VALIDACIÓN SIMPLE
- ACCESO A TABLAS DE CÓDIGO.
- FRONT END DEL CLIENTE.
 - Recepción/Envío de eventos de la Interfaz de usuario.
 - Recepción/Envío de eventos asíncronos.
 - Manejador de errores..
- MANEJADOR DE MEMORIA COMPARTIDA
- TRANSCRIPCIÓN DE SESIÓN
 - Log de Mensajes.
 - Debugger.
- SEGURIDAD

COMPONENTES DE ARQUITECTURA

SERVICIOS DE PRESENTACIÓN

CLIENTE

SERVICIOS DE DISTRIBUCIÓN

SERVIDOR

SERVICIOS DE BASE DE DATOS



FUNCIÓN

- COMUNICACIÓN PROGRAMA A PROGRAMAS.
- MANEJO DE MENSAJES
 - Síncrono/Asíncrono
 - Ruteo transparente
 - Concordancia de petición/respuesta.
 - Traducción de datos.
 - Manejo de timeout.
 - Garantía de entrega
- INDEPENDENCIA DE PROTOCOLO

COMPONENTES DE ARQUITECTURA

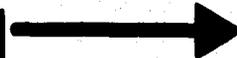
SERVICIOS DE PRESENTACIÓN

CLIENTE

SERVICIOS DE DISTRIBUCIÓN

SERVIDOR

SERVICIOS DE BASE DE DATOS



FUNCIÓN

- FUNCIÓN DE TRABAJO REUSABLE
- FUNCIÓN DE TRABAJO PORTABLE
- ACCESO A TABLAS DE CÓDIGO.
- FRONT END DEL SERVIDOR
 - Recepción/Envío de mensajes de servicio.
 - Respuesta a mensajes.
 - Recuperación
 - Manejador de



COMPONENTES DE ARQUITECTURA

FUNCIÓN

SERVICIOS DE PRESENTACIÓN

CLIENTE

SERVICIOS DE DISTRIBUCIÓN

SERVIDOR

SERVICIOS DE BASE DE DATOS

- IMPLEMENTACIÓN CLIENTE/SERVIDOR
- *TWO PHASE COMMIT*
- INTEGRIDAD DE DATOS Y RECUPERACIÓN
- SEGURIDAD
 - Recepción/Envío de mensajes de servicio.
 - Respuesta a mensajes.
 - Recuperación
 - Manejador de errores
- SEGURIDAD.





APÉNDICE I

CAPA DE COMPATIBILIDAD VISUAL BASIC-ACCESS



**VISUAL BASIC
COMMLAYER
ACCESS**





INTERFACE VB 3.0-ACCESS 2.0

Cuando se utiliza una aplicación de Visual Basic versión 3.0 para acceder a datos de alguna Base de Datos, se debe actualizar la aplicación de Visual Basic antes de que ésta pueda utilizar datos almacenados en una base de datos como la de Microsoft Access versión 2.0. Para actualizar una aplicación de Visual Basic se utiliza el *Compatibility Layer (COMLYR.EXE)*.

El Visual Basic versión 3.0 realizado en 1993 incluía soporte para el manejador de bases de datos Microsoft Access versión 1.1 que también se conoce como el *Microsoft Jet Database Engine (Jet)*.

Microsoft Access versión 2.0 contiene una nueva versión del manejador de bases de datos, Jet 2.0, mismo que puede acceder a aplicaciones de Visual Basic versión 3.0.

Para habilitar las aplicaciones de Visual Basic versión 3.0 con el uso del manejador de Bases de Datos Jet 2.0, Microsoft creó el Microsoft Jet 2.0/Visual Basic 3.0 *Compatibility Layer*.

El *Compatibility Layer* se compone de lo siguiente:

- ◆ Una actualización a la versión del Visual Basic con las interfaces del Jet (VBDB300.DLL).
- ◆ Un nuevo archivo (MSAJT112.DLL) que en conjunto con el archivo VBDB300.DLL interactúan con el manejador Microsoft Jet 2.0.
- ◆ Archivos que modifican los reportes de Visual Basic para trabajar con el Jet 2.0.
- ◆ Versiones actualizadas del soporte Jet a través de drivers ISAM (*Paradox, FoxPro y Btrieve*).
- ◆ Un archivo *setup* que se instala con el *Compatibility Layer* y actualiza apropiadamente los archivos del sistema.

**Beneficios del Jet 2.0:**

- ◆ Actualización en cascada y eliminación permitida a los usuarios para que borren o actualicen los registros de la base de datos.
- ◆ Validación en tablas iguales otorgando un gran aumento en la integridad y seguridad de los datos.
- ◆ Optimización en las consultas *Rushmore* que se ejecutan con rapidez, la tecnología *Rushmore* optimiza las consultas cuando los criterios de las mismas usan ciertos tipos de expresiones.
- ◆ Virtualmente cualquier perspectiva o forma puede ser actualizada
- ◆ Soporte para archivos con formato Paradox 4.x o Btrieve 6.x.
- ◆ Proporciona más palabras reservadas para las instrucciones SQL.

Limitaciones del Jet 2.0:

- ◆ La actualización y eliminación de registros a través de Visual Basic sólo es permitida cuando se está usando el Microsoft Access versión 2.0
- ◆ La validación de tablas iguales para garantizar la integridad y seguridad de los datos sólo es posible cuando se está usando Microsoft Access versión 2.0
- ◆ El Jet 2.0 no maneja cadenas de longitud cero, por consiguiente algunas actualizaciones pueden fallar porque se infringen las reglas de validación de los datos, en Visual Basic no se puede usar esto.
- ◆ El formato de bases de datos Jet 1.0 no se puede crear por archivos Jet 2.0
- ◆ Para obtener datos de Microsoft Access, se necesita:
- ◆ El controlador de ODBC Microsoft Access



- El administrador de controladores ODBC 1.0 (ODBC.DLL.)

- Una PC con el sistema operativo MS-DOS versión 3.3 o posterior

- Microsoft Windows 3.1 o posterior

- Microsoft Access versión 2.0

ODBC (*Open Database Connectivity*) Conexión de bases de datos abierta. Es un medio potente de acceder a los datos almacenados en un amplio rango de sistemas gestores de bases de datos. Access usa controladores ODBC para gestionar la información de distintas fuentes de datos. Los controladores ODBC que vienen con Access son controladores para Access, FoxPro, dBASE, Paradox y Btrieve.

Para instalar el controlador de ODBC Microsoft Access se siguen los siguientes pasos:

- I. En el grupo Principal de la ventana Administrador de programas, haga doble clic en el icono Panel de Control. En la ventana Panel de Control, haga doble clic en el icono ODBC.

- II. En el cuadro de diálogo Orígenes de datos, elija el botón Controladores

- III. En el cuadro de diálogo controladores, elija el botón Agregar.

- IV. En el cuadro de texto del cuadro diálogo Agregar controlador, escriba el nombre de la unidad y del directorio que contiene el controlador ODBC Microsoft Access. También puede elegir el botón Examinar para seleccionar una unidad y un nombre de directorio.

- V. Elija el botón Aceptar.



VI. En el cuadro de diálogo Instalar controladores, elija Access Data en la lista de Controladores ODBC disponibles

VII. Elija el botón Aceptar para instalar dicho controlador.

Para obtener acceso a datos con el controlador de ODBC Microsoft Access, deberá agregar un *origen de datos* llevando a cabo los siguientes pasos:

I. En el grupo Principal de la ventana Administrador de programas, haga doble clic en el icono Panel de control. En la ventana Panel de control, haga doble clic en el icono ODBC.

II. En el cuadro de diálogo Orígenes de datos, elija el botón Agregar.

III. En el cuadro de diálogo Agregar un origen de datos, elija Access Data en la lista de Controladores ODBC instalados y luego elija el botón Aceptar.

IV. En el cuadro de diálogo Instalación de ODBC Microsoft Access, introduzca la información necesaria para configurar el origen de los datos.

Cuando se conecta a un *origen de datos* de Microsoft Access, es posible que una aplicación le solicite que introduzca el nombre de una Base de Datos de Microsoft Access. De ser así, escriba o seleccione la ruta de acceso completa de la base de datos de Microsoft Access a la que desea acceder. Es posible que una aplicación le pida que introduzca un nombre y una contraseña. El nombre predeterminado es Admin y la contraseña predeterminada de Access para el usuario Admin es una cadena vacía.

Para el uso del controlador ODBC Microsoft Access, la siguiente información puede resultar muy útil:



Columnas

- Los nombres de columna están limitados a 64 caracteres como máximo. Los nombres de columnas que tengan más de 64 caracteres producen un error.
- El controlador permite que los nombres de columnas contengan cualquier carácter válido de Microsoft Access, como por ejemplo espacios. Si los nombres de columna contienen caracteres que no sean letras, números y caracteres de subrayado, éstos deberán ser delimitados. Para delimitar un nombre de columna, ponga el nombre entre apóstrofes (').
- El controlador supone que la primera fila de la base de datos de Microsoft Access contiene los nombres de columna.
- El controlador acepta un máximo de 255 columnas cuando se crea una tabla.

Tablas

- El controlador acepta nombres de tablas de una longitud de hasta 64 caracteres.
- El controlador permite que los nombres de tablas contengan cualquier carácter válido de Microsoft Access, como por ejemplo espacios. Si los nombres de tablas contienen caracteres que no sean letras, números y caracteres de subrayado, éstos deberán ser delimitados. Para delimitar un nombre de tabla, ponga el nombre entre apóstrofes (') o entre corchetes.

Grupo de caracteres

- La longitud máxima de cualquier grupo de caracteres (como por ejemplo una cadena) es de 1000 caracteres.

Una consulta de paso a través es una consulta que envía órdenes SQL directamente a un servidor de bases de datos ODBC. Una consulta de paso a través le permite trabajar con tablas en un servidor ODBC sin unir las tablas a una base de datos Access. Una consulta de paso a través



también puede ejecutar procedimientos almacenados en un servidor ODBC. Este tipo de consulta se puede usar para cambiar datos en la base de datos que se esté editando, crear un objeto de base de datos o hacer una acción parecida a una consulta de acción. Esto se hace introduciendo las órdenes SQL adecuadas para estas acciones, que se efectúan en el servidor ODBC, no en Access.

El controlador de ODBC Microsoft Access permite al usuario de la familia de productos Microsoft Office abrir y realizar una consulta en una base de datos de Microsoft Access mediante la interface ODBC. El uso de este controlador sólo es compatible con productos de la familia *Microsoft Office*.

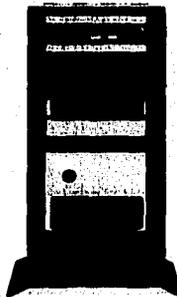
Si a través de una consulta hecha hacia Microsoft Access por medio de la interface ODBC se cambian los datos o se introduce nueva información, estos cambios se salvan automáticamente en la base de datos mientras se mueve el control de datos hacia un nuevo registro. Se puede usar un evento de validación y la propiedad de cambio de datos para realizar esta aplicación. Para esto se utilizan unas estructuras de datos conocidas como *Dynaset* en donde se almacenan grupos de registros que cumplen con cierta condición dada en una consulta, se pueden incluir campos de una o más tablas y su ventaja principal es que almacena los cambios hechos a estos registros y los actualiza en las tablas que conforman la base de datos.

Visual Basic usa la base de datos de Access como un instrumento para un acceso local de los datos y el ODBC para un acceso remoto de los datos. En Visual Basic se incluyen funciones en SQL que se usan cuando se está consultando una Base de datos.

APÉNDICE J

TERMINOLOGÍA CISCO

**SERVIDORES DE
COMUNICACIONES.**



**INSTRUCCIONES DEL SERVIDOR CISCO DE COMUNICACIONES (CS-500)****TROUTER>?****Comandos Exec.**

connect	Abre una conexión de terminal.
disconnect	Desconecta una conexión existente de red.
enable	Activa los comandos especiales.
exit	Salida del EXEC.
help	Descripción del sistema interactivo de ayuda.
lat	Abre una conexión lat.
lock	Asegura una terminal.
login	Accesar el sistema como un usuario determinado.
logout	Salida del EXEC.
name/connection	Da un nombre a una conexión de red existente.
ping	Envía un mensaje de eco.
ppp	Inicia una conexión IETF con el protocolo Punto a Punto (PPP)
resume	Resume de las actividades de la conexión a la red.
rlogin	Inicia una conexión rlogin.
show	Muestra información del sistema.
slip	Inicia una conexión Serial\line IP (SLIP)
sysstat	Muestra información de las líneas de la terminal.
telnet	Inician una conexión telnet.
terminal	Determina los parametros de línea.
tn3270	Inicia una conexión tn3270.
trace	Trazado de una ruta de destino.
where	Lista de las conexiones activas.
xremote	Entra al modo XRemote.

TROUTER>terminal ?



autohangup	Automáticamente cuelga cuando la última conexión se cierra.
data\character\bits	Tamaño de los caracteres que son manejados.
databits	Especifica el número de bits de datos por caracter.
dispatch\character	Define el caracter de envío.
dispatch\timeout	Define el tiempo de envío
download	Especifica la línea que se encuentra en el modo de envío.
editing	Activa la línea de edición de comandos.
escape\character	Cambia el caracter de ESCAPE de la línea actual.
exec\character\bits	Tamaño de caracteres del comando exec.
flowcontrol	Especifica el control de flujo.
full\help	Provee ayuda a usuarios sin privilegios.
help	Describe un sistema de ayuda interactivo.
history	Determina el tamaño del <i>buffer</i> del comando <i>history</i> .
hold\character	Define el caracter para el comando hold.
keymap\type	Especifica una entrada <i>Keymap</i> al usuario.
lat	Especifica la configuración del protocolo DEC Transporte de Area Loca (LAT).
length	Determina el número de líneas en la pantalla.
no	Niega un comando o lo especifica por default.
notify	Informa a los usuarios de los resultados de la sesión actual.
padding	Coloca el caracter de relleno a un caracter de salida.
parity	Determina la paridad de la terminal
rxspeed	Indica la velocidad de recepción.
special\character\bits	Especifica el tamaño de los caracteres especiales.
speed	Determina la velocidad de transmisión y recepción.
start\character	Define el caracter de inicio.
stop\character	Define el caracter de paro.
stopbits	Determina los bits de paro en una línea asíncrona.



telnet	Configuración específica del protocolo Telnet.
telnet\transparent	Envía un CR seguido de un NULL en lugar de un CR simple.
terminal\type	Determina el tipo de terminal.
transport	Define el protocolo de transporte de la línea.
txspeed	Determina la velocidad de transmisión.
width	Especifica el ancho de despliegue de la terminal.

TROUTER>terminal width ?

<0\512> Número de caracteres en una línea de la pantalla.

TROUTER>terminal txspeed ?

<0\4294967295> Velocidad de Transmisión.

TROUTER>terminal transport ?

lat	Protocolo DEC LAT.
mop	Protocolo de consola remota DEC MOP.
none	Sin protocolo.
rlogin	Protocolo rlogin de Unix.
telnet	Protocolo Telnet TCP/IP.

TROUTER>terminal terminal\type ?

WORD Tipo de Terminal.

TROUTER>terminal telnet ?



<code>break\onlip</code>	Envía una señal de ruptura cuando la señal de interrupción es detectada.
<code>refuse\negotiations</code>	Suprime las negociaciones de los ecos remotos de telnet y suprime la opción de adelante.
<code>speed</code>	Especifica la velocidad de la línea.
<code>sync\on\break</code>	Envía una señal sincronizada de telnet después de recibir una señal de rompimiento telnet

TROUTER>terminal telnet speed ?

<1\4294967295> Velocidad de default.

TROUTER>terminal stopbits ?

1 Un bit de paro.
1.5 Uno y medio bits de paro.
2 Dos bits de paroTwo stop bits

TROUTER>terminal stop\character ?

CHAR or <0\255> Caracter de paro o su equivalente decimal.

TROUTER>terminal start\character ?

CHAR or <0\255> Caracter de inicio o su equivalente decimal.

TROUTER>terminal speed ?

<0\4294967295> Velocidades de transmisión y de recepción.

TROUTER>terminal special\character\bits ?

<7\8> Bits por caracter.

TROUTER>terminal rxspeed ?



<0\4294967295>

Velocidad de recepción.

TROUTER>terminal parity ?

even	Paridad impar.
mark	Paridad marcada.
none	No paridad.
odd	Paridad par.
space	Paridad Space.

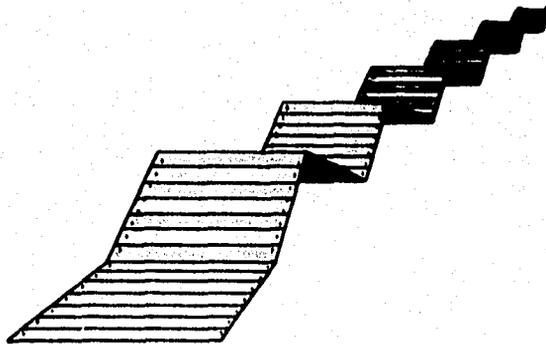
TROUTER>terminal flowcontrol ?

NONE	Determina que no exista control de flujo.
hardware	Control de flujo mediante hardware.
software	Control de flujo mediante software.



APÉNDICE K

LISTADO DEL PROGRAMA




ESPECIFICAR EL FICHERO DE AYUDA

App.HelpFile = "ctrlsper.hlp"

DECLARACIÓN DE VARIABLES.

```

Dim N As Integer
Option Explicit
Const Series = 3 'número máximo de series
Const Datos = 5 'número máximo de datos
Sub Combo1_Click ()
Dim Total As Integer
Dim regis As passtipo
Dim buffersal As String 'buffer de salida
'tamaño del buffer de E y de S
Dim EVMag As String
Dim ERMag As String
Dim VR As Integer
'Saltar de acuerdo con la propiedad CommEvent
Select Case CommX.CommEvent
Dim indice As Integer

```

PANTALLA DEL SISTEMA PRINCIPAL.

```

Sub Form_Load ()
Unload ACCESO
app.HelpFile = "c:\liliana\virusita\tesihel3.hlp"
End Sub
Sub Form_Unload (Cancel As Integer)
DATOS.HORA_SAL = Now
Open "C:\LILIANA\VIRUSITA\BITACORA.DAT" For Random As #1
' Put #1, , DATOS
'Close #1
End Sub
Sub Frame3DI_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
LABEL1.Visible = False
End Sub
Sub Image1_Click ()
smbaj_Click
End Sub
Sub Image1_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
LABEL1.Left = IMAGE1.Left + IMAGE1.Width
LABEL1.Top = IMAGE1.Top + IMAGE1.Height
LABEL1.Caption = "BAJAS"
LABEL1.Visible = True

```



```
End Sub
Sub Image10_Click ()
    msal_Click
End Sub
Sub Image10_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE10.Left
    LABEL1.Top = IMAGE10.Top + IMAGE10.Height
    LABEL1.Caption = "SALIR"
    LABEL1.Visible = True
End Sub
Sub Image11_Click ()
    smalt_Click
End Sub
Sub Image11_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE11.Left + IMAGE11.Width
    LABEL1.Top = IMAGE11.Top + IMAGE11.Height
    LABEL1.Caption = "ALTAS"
    LABEL1.Visible = True
End Sub
Sub Image12_Click ()
    smaut_Click
End Sub
Sub Image12_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE12.Left + IMAGE12.Width
    LABEL1.Top = IMAGE12.Top + IMAGE12.Height
    LABEL1.Caption = "MONITOREO AUTOMÁTICO"
    LABEL1.Visible = True
End Sub
Sub Image2_Click ()
    smcam_Click
End Sub
Sub Image2_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE2.Left + IMAGE2.Width
    LABEL1.Top = IMAGE2.Top + IMAGE2.Height
    LABEL1.Caption = "CAMBIOS"
    LABEL1.Visible = True
End Sub
Sub Image3_Click ()
    sbtipos_Click
End Sub
Sub Image3_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE3.Left + IMAGE3.Width
    LABEL1.Top = IMAGE3.Top + IMAGE3.Height
    LABEL1.Caption = "CONSULTAS"
End Sub
```



```
Sub Image4_Click ()
    smman_Click
End Sub
Sub Image4_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE4.Left + IMAGE4.Width
    LABEL1.Top = IMAGE4.Top + IMAGE4.Height
    LABEL1.Caption = "MONITOREO MANUAL"
    LABEL1.Visible = True
End Sub
Sub Image5_Click ()
    smren_Click
End Sub
Sub Image5_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE5.Left + IMAGE5.Width
    LABEL1.Top = IMAGE5.Top + IMAGE5.Height
    LABEL1.Caption = "Graf. Rendimiento"
    LABEL1.Visible = True
End Sub
Sub Image6_Click ()
    smbnt_Click
End Sub
Sub Image6_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE6.Left + IMAGE6.Width
    LABEL1.Top = IMAGE6.Top + IMAGE6.Height
    LABEL1.Caption = "BITÁCORA"
    LABEL1.Visible = True
End Sub
Sub Image7_Click ()
    smpro_Click
End Sub
Sub Image7_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE7.Left + IMAGE7.Width
    LABEL1.Top = IMAGE7.Top + IMAGE7.Height
    LABEL1.Caption = "Graf. Procesamiento"
    LABEL1.Visible = True
End Sub
Sub Image9_Click ()
    smbunca_Click
End Sub
Sub Image9_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    LABEL1.Left = IMAGE9.Left + IMAGE9.Width
    LABEL1.Top = IMAGE9.Top + IMAGE9.Height
    LABEL1.Caption = "AYUDA"
    LABEL1.Visible = True
End Sub
```



```
Sub mcon_Click ()
    CONSULTA.Show
End Sub
Sub msal_Click ()
    BANDERA = 1
    Load FORM_Password
    FORM_Password.Show
    'Unload sistema
End Sub
Sub smacerca_Click ()
    Screen.MousePointer = 11
    Load AcercaDe
    AcercaDe.Show 1
End Sub
Sub smalt_Click ()
    Screen.MousePointer = 11
    Moomin2.Show 1
End Sub
Sub smaut_Click ()
    MsgBox "Activando Monitoreo Automático"
End Sub
Sub smbaj_Click ()
    BAJAS.Show
End Sub
Sub smbit_Click ()
    Open BITACORAS For Random As #1 Len = 57
    Total = LOF(1) / 57
    Close #1
    If Total > 0 Then
        bitacora.Show 1
    Else
        MsgBox "El archivo de Bitácoras esta vacío", 48, "Notificación"
    End If
End Sub
Sub smbunca_Click ()
    Screen.MousePointer = 11
    R = WinHelp(sistema.hwnd, "c:\lilianavirusita\tesihel3.hlp", HELP_PARTIALKEY,
    CLng(1010))
    Screen.MousePointer = 0
End Sub
Sub smcam_Click ()
    RESP_CONSUL = ""
    RESP_CONSUL = UCase(InputBox("INTRODUZCA EL LOGIN DEL USUARIO A
    MODIFICAR"))
    If RESP_CONSUL <> "" Then
```



```

cambiar2!Data3.Refresh
cambiar2!Data3.Recordset.FindFirst "[LOGIN] LIKE " & RESP_CONSUL & ""
If cambiar2!Data3.Recordset.NoMatch Then
    msg = "No hay algún Usuario con Login" + Chr(13) + Chr(10) + Case(RESP_CONSUL)
    MsgBox msg, 48, "Error"
Else
    Load cambiar2
    cambiar2.Show
End If
Else
End If
End Sub
Sub smcontenido_Click ()
    Screen.MousePointer = 11
    R = WinHelp(sistema.hWnd, "c:\Militana\virusita\tesihel3.hlp", 1, CLng(1010))
    Screen.MousePointer = 0
End Sub
Sub smman_Click ()
    MsgBox "BLOQUE EN CONSTRUCCION"
    cuestionario.Show
End Sub
Sub smpro_Click ()
    Screen.MousePointer = 11
    Load Graphi
    Graphi.Show
End Sub
Sub smren_Click ()
    gráfica.Show
End Sub

```

MODULO DE MONITOREO MONITOREO

```

Sub Cerrar_Com ()
    If CommX.PortOpen Then
        CommX.PortOpen = False
    End If
End Sub
Sub CommX_OnComm ()
    'Mensajes relativos a sucesos
    Case MSCOMM_EV_RECEIVE
        recibireom
    Case MSCOMM_EV_SEND
    Case MSCOMM_EV_CTS
        EVMsg = "Cambio detectado en CTS"
    Case MSCOMM_EV_DSR

```



```
    EVMsg = "Cambio detectado en DSR"
Case MSCOMM_EV_CD
    EVMsg = "Cambio detectado en DCD"
Case MSCOMM_EV_RING
    EVMsg = "El teléfono está sonando"
Case MSCOMM_EV_EOF
    EVMsg = "Fin de fichero"
'Mensajes de error
Case MSCOMM_ER_BREAK
    EVMsg = "Interrupción detectada"
Case MSCOMM_ER_CTSTO
    ERMMsg = "Tiempo para CTS sobrepasado"
Case MSCOMM_ER_DSRTO
    ERMMsg = "Tiempo para DSR sobrepasado"
Case MSCOMM_ER_FRAME
    EVMsg = "Error de transmisión (encuadre)"
Case MSCOMM_ER_OVERRUN
    ERMMsg = "Error de sobreescritura"
Case MSCOMM_ER_CDTO
    ERMMsg = "Tiempo para DCD sobrepasado"
Case MSCOMM_ER_RXOVER
    ERMMsg = "Buffer de recepción lleno"
Case MSCOMM_ER_RXPARITY
    EVMsg = "Error de paridad"
Case MSCOMM_ER_TXFULL
    ERMMsg = "Buffer de transmisión lleno"
Case Else
    ERMMsg = "Error o suceso desconocido"
End Select
If Len(EVMsg) Then
    'Visualizar el mensaje EVMsg
    MsgBox EVMsg
    EVMsg = ""
Elseif Len(ERMMsg) Then
    'Visualizar el mensaje de error
    Beep
    VR = MsgBox(ERMMsg, 1, "Pulse Cancelar para salir, Aceptar para ignorar.")
    ERMMsg = ""
    'Si se pulsó Cancelar
    If VR = 2 Then
        CommX.PortOpen = False 'Cerrar el puerto y salir
    End If
End If
End Sub
Sub Config_Click ()
```

```

' ConfigCom.Show
End Sub
Sub Detener_Click ()
  Cerrar_Com
  formcom.Caption = "Puerto de comunicaciones cerrado"
End Sub
Sub Form_Load ()
  MONITOR = 0
  var = 0
  'Características por defecto del puerto
  DefCom$ = "COM1:300,N,8,1"
  Abrir_Com DefCom$
  If CommX.PortOpen Then
    'formcom.Caption = "Puerto de comunicaciones abierto"
  End If
  'bloque que determina si se hará monitoreo o no
  'si=1, no=0
  Randomize
  MONITOR = Int(2 * Rnd)
  MsgBox "SISTEMA ACTIVADO" + Str$(MONITOR), 4160, "PROGRAMA MONITOR"
End Sub
Sub Salir_Click ()
  Cerrar_Com
  End
End Sub
Sub Text1_KeyPress (KeyAscii As Integer)
  Dim MSG As String, NumCars, var As Integer
  If ((KeyAscii >= 48) And (KeyAscii <= 57)) Or ((KeyAscii >= 64) And (KeyAscii <= 90)) Or
    ((KeyAscii >= 97) And (KeyAscii <= 122)) Or (KeyAscii = 13) Then
    If KeyAscii = 13 Then
      If Len(text1.Text) = 9 Then
        CommX.Output = buffersal
        buffersal = Mid$(buffersal, NumCars + 1)
        text1.Text = ""
        buffersal = ""
        ciclo = 1
        text2.SetFocus
      Else
        MsgBox "Clave incorrecta"
        buffersal = ""
        text1.Text = ""
        text3.SetFocus
      End If
    Else
      buffersal = buffersal + Chr$(KeyAscii)
    End If
  End Sub

```



```

End If
Else
text1.Text = ""
buffersal = ""
MsgBox "Caracter Inválido"
text3.SetFocus
End If
End Sub
Sub Text2_KeyPress (KeyAscii As Integer)
Dim MSG As String, NumCars As Integer
If ((KeyAscii >= 48) And (KeyAscii <= 57)) Or ((KeyAscii >= 64) And (KeyAscii <= 90)) Or
((KeyAscii >= 97) And (KeyAscii <= 122)) Or (KeyAscii = 13) Then
If KeyAscii = 13 Then
If text2.Text = "OSCAR" Or text2.Text = "VEROI" Then
CommX.Output = buffersal
buffersal = Mid$(buffersal, NumCars + 1)
buffersal = ""
ciclo = 2
MOUSEPOINTER = 11
MONITOR = 1
IF MONITOR Then ' RUTINA DE ACTIVACIÓN DE
Load cuestionario ' MONITOREO SI=1,NO=0
cuestionario.Show
Else
Load form1
form1.Show
End If
Else
var = var + 1
If var = 3 Then
MsgBox " desconexión de línea..."
buffersal = ""
text2.Text = ""
End
Else
MsgBox " PASSWORD INCORRECTO"
text4.SetFocus
End If
End If
Else
buffersal = buffersal + Chr$(KeyAscii)
End If
Else
text2.Text = ""
buffersal = ""

```



```

    MsgBox "Caracter Inválido"
    text4.SetFocus
End If
End Sub
Sub Text3_GotFocus ()
    text1.Text = ""
    buffersal = ""
    text1.SetFocus
End Sub
Sub Text4_GotFocus ()
    text2.Text = ""
    buffersal = ""
    text2.SetFocus
End Sub

```

MONITOREO AUTOMATICO (VALIDACION)

```

pos = InStr(combo1.Text, "") 'posición del número
cad = Mid$(combo1.Text, 1, pos + 1)
CLV_USU = Val(cad)
' MsgBox clv_usu
' PRUEBA
If CLV_USU <> "" Then
    BITA1!USUS.Refresh
    BITA1!USUS.Recordset.FindFirst "[ID_US] LIKE '" & CLV_USU & "'"
    If BITA1!USUS.Recordset.NoMatch Then
        msg = "No hay algún Usuario con Login" + Chr(13) + Chr(10) + UCase(LOGIN)
        MsgBox msg, 48, "Error"
        BITA1!Option3D1.Value = True
        BITA1!Option3D1.Visible = True
    Else
        Load BITA1
        BITA1.Show
    End If
Else
    BITA1!Option3D1.Value = True
End If
End Sub

```

MODULO DE BITACORA

```

Sub Form_Load ()
    DESCONEXION = 0
    Data2.Refresh
    Do While Not Data2.Recordset.EOF

```



```

If Data2.Recordset(5) <> "" Then
    combo1.AddItem Data2.Recordset(0) + "      " + "3"
DESCONEXION = 1
Elseif Data2.Recordset(3) <> "" Then
    combo1.AddItem Data2.Recordset(0) + "      " + "2"
Elseif Data2.Recordset(1) <> "" Then
    combo1.AddItem Data2.Recordset(0) + "      " + "1"
Else
    combo1.AddItem Data2.Recordset(0) + "      " + "0"
End If
Data2.Recordset.MoveNext
Loop
End Sub

```

MODULO DE GRAFICOS

```

Sub AyudaAcercaDe_Click ()
    'InfoSis.Show
End Sub
Sub AyudaBuscar_Click ()
    Dim vr As Integer
    vr = WinHelp(hWnd, (App.HelpFile), HELP_PARTIALKEY, CLng(0))
End Sub
Sub AyudaContenido_Click ()
    Dim vr As Integer
    vr = WinHelp(hWnd, (App.HelpFile), HELP_INDEX, CLng(0))
End Sub
Sub Command3D1_Click ()
    Graph1.DrawMode = 5
End Sub
Sub Command3D2_Click ()
    gráfica.Hide
    Unload gráfica
End Sub
Sub Form_Load ()
    'Hacer que el directorio donde se ejecuta la
    'aplicación sea el directorio de trabajo
    ChDir App.Path
    'Inicializar el generador de números aleatorios
    Randomize
    'Visualizar el primer diagrama
    Graph1_Click
End Sub
Sub Graph1_Click ()
    Dim S As Integer, D As Integer, Dato As Integer

```



```

Dim Mayor As Integer, Menor As Integer
Dim IMayor As Integer, IMenor As Integer
'Títulos superior e izquierdo
Graph1.GraphTitle = "ACCESOS"
Graph1.LeftTitle = "%"
'Introduciendo datos y calcular el mayor y el menor
Dato = Int(Rnd * 76) + 25
Menor = Dato: IMenor = 1
For S = 1 To Graph1.NumSets
  For D = 1 To Graph1.NumPoints
    If Dato > Mayor Then
      Mayor = Dato 'dato mayor
      IMayor = D 'índice del dato mayor
    ElseIf Dato < Menor Then
      Menor = Dato 'dato menor
      IMenor = D 'índice del dato menor
    End If
    Graph1.GraphData = Dato 'asignar siguiente dato
    Dato = Int(Rnd * 76) + 25
  Next D
Next S
'Tamaño de la fuente para los títulos
'Detalles para los gráficos
If Graph1.GraphType = 1 Or Graph1.GraphType = 2 Then
'Si es un diagrama de sectores...
Graph1.DataReset = 3 'inicializar separar sectores
Graph1.BottomTitle = "" 'eliminar título inferior
  'Poner leyendas
  For D = 1 To Graph1.NumPoints
    Graph1.LegendText = "CLAVE " & D
  Next
  If SepararSec.Value Then 'si separar sectores...
    Graph1.ThisPoint = IMayor 'elemento actual
    Graph1.ExtraData = 1 'separar sector actual
    Graph1.ThisPoint = IMenor 'elemento actual
    Graph1.ExtraData = 1 'separar sector actual
  End If
Else
'Si es un diagrama de barras, líneas o áreas
  Graph1.DataReset = 5 'eliminar leyendas
  Graph1.BottomTitle = "Aulas" 'título inferior
End If
'Visualizar el gráfico
Graph1.DrawMode = 2
End Sub

```



```
Sub ImprimirGráfico_Click ()
    Graph1.DrawMode = 5
End Sub

Sub NDatos_Change ()
    Dim N As Integer
    N = CInt(NDatos.Text)
    If N >= 2 And N <= Datos Then
        Graph1.NumPoints = N
    Else
        MsgBox "El valor tiene que estar entre 2 y " & Datos
        NDatos.Text = Datos
        Graph1.NumPoints = Datos
    End If
End Sub

Sub NSeries_Change ()
    N = CInt(NSeries.Text)
    If N >= 1 And N <= Series Then
        Graph1.NumSets = N
    Else
        MsgBox "El valor tiene que estar entre 1 y " & Series
        NSeries.Text = 1
        Graph1.NumSets = 1
    End If
End Sub

Sub Opción_Click (Index As Integer, Value As Integer)
    'Deshabilitar y ocultar la opción separar sector
    SepararSec.Value = False
    SepararSec.Visible = False
    'Tipo de gráfico
    Select Case Index
        Case 0 'barras 2D
            Graph1.GraphType = 3
        Case 1 'barras 3D
            Graph1.GraphType = 4
        Case 2 'sectores 2D
            Graph1.GraphType = 1
            NSeries.Text = 1
            SepararSec.Visible = True
        Case 3 'sectores 3D
            Graph1.GraphType = 2
            NSeries.Text = 1
            SepararSec.Visible = True
        Case 4 'líneas
            Graph1.GraphType = 6
        Case 5 'áreas
```



```
Graph1.GraphType = 8
End Select
End Sub
Sub SALIR_Click ()
gráfica.Hide
Unload gráfica
End Sub
Sub Spin1_SpinDown ()
N = Cint(NSeries.Text)
If N = 1 Then
NSeries.Text = Series
Else
NSeries.Text = N - 1
End If
End Sub
Sub Spin1_SpinUp ()
Dim N As Integer
N = Cint(NSeries.Text)
If N = Series Then
NSeries.Text = 1
Else
NSeries.Text = N + 1
End If
End Sub
Sub Spin2_SpinDown ()
Dim N As Integer
N = Cint(NDatos.Text)
If N = 2 Then
NDatos.Text = Datos
Else
NDatos.Text = N - 1
End If
End Sub
Sub Spin2_SpinUp ()
Dim N As Integer
N = Cint(NDatos.Text)
If N = Datos Then
NDatos.Text = 2
Else
NDatos.Text = N + 1
End If
End Sub
Sub Combo1_Click ()
Combo7.ListIndex = Combo1.ListIndex
MaskedEdit39.Text = Val(Combo7.Text)
```



```
MaskedEdit35.SetFocus
End Sub
Sub Combo1_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        MaskedEdit35.SetFocus
    End If
End Sub
Sub Combo2_Click ()
    Combo9.ListIndex = Combo2.ListIndex
    MaskedEdit41.Text = Val(Combo9.Text)
    MaskedEdit36.SetFocus
End Sub
Sub Combo2_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        MaskedEdit36.SetFocus
    End If
End Sub
Sub Combo3_Click ()
    Combo10.ListIndex = Combo3.ListIndex
    MaskedEdit42.Text = Val(Combo10.Text)
    Combo5.SetFocus
End Sub
Sub Combo3_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        Combo5.SetFocus
    End If
End Sub
Sub Combo4_Click ()
    Combo8.ListIndex = Combo4.ListIndex
    MaskedEdit40.Text = Val(Combo8.Text)
    MaskedEdit30.SetFocus
End Sub
Sub Combo4_KeyPress (KeyAscii As Integer)
    If KeyAscii = 13 Then
        MaskedEdit30.SetFocus
    End If
End Sub
Sub Combo5_Click ()
    MaskedEdit43.Text = Combo6.List(Combo5.ListIndex)
    Command3D4.SetFocus
End Sub
Sub Combo5_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13) Then
        Command3D4.SetFocus
    End If
End Sub
```



```
End If
End Sub
Sub Command3D1_Click ()
    Unload cambiar2
End Sub
Sub Command3D2_Click ()
    Data3.Recordset.MoveNext
    Data4.Refresh
    Data4.Recordset.FindFirst "[ID_DEL] like '" & CStr(Data3.Recordset(9)) & ""
    Combo1.Text = Data4.Recordset(1)
    Data5.Refresh
    Data5.Recordset.FindFirst "[ID_INST] like '" & CStr(Data3.Recordset(15)) & ""
    Combo4.Text = Data5.Recordset(1)
    Data1.Refresh
    Data1.Recordset.FindFirst "[ID_LUGAR] like '" & CStr(Data3.Recordset(13)) & ""
    Combo2.Text = Data1.Recordset(1)
    Data2.Refresh
    Data2.Recordset.FindFirst "[ID_CVE] like '" & CStr(Data3.Recordset(14)) & ""
    Combo3.Text = Data2.Recordset(1) + " Horas"
    Data6.Refresh
    Data6.Recordset.FindFirst "[ID_TIPO] like '" & CStr(Data3.Recordset(22)) & ""
    Combo5.Text = Data6.Recordset(1)
End Sub
Sub Command3D4_Click ()
    Dim respu As Integer
    If Data3.ReadOnly = True Or Data3.Recordset.Updatable = False Then
        MsgBox "Registro Sólo de lectura...!", 48, "Error"
    ElseIf MaskedEdit39.Text = "" Then
        MsgBox "Faltó especificar la Delegación", 48, "Error"
    ElseIf MaskedEdit40.Text = "" Then
        MsgBox "Faltó especificar la Institución", 48, "Error"
    ElseIf MaskedEdit41.Text = "" Then
        MsgBox "Faltó especificar el lugar de Nacimiento", 48, "Error"
    ElseIf MaskedEdit42.Text = "" Then
        MsgBox "Faltó especificar el tipo de Clave", 48, "Error"
    ElseIf MaskedEdit43.Text = "" Then
        MsgBox "Faltó especificar el Tipo de Usuario", 48, "Error"
    Else
        'Se actualiza el valor de los campos
        MaskedEdit2.Text = UCase(MaskedEdit23.Text)
        MaskedEdit3.Text = UCase(MaskedEdit24.Text)
        MaskedEdit4.Text = UCase(MaskedEdit25.Text)
        MaskedEdit5.Text = UCase(MaskedEdit26.Text)
        MaskedEdit6.Text = MaskedEdit27.Text
        MsgBox "Numero Domicilio " + MaskedEdit6.Text
```



```

MaskedEdit7.Text = UCase(MaskedEdit29.Text)
MaskedEdit8.Text = MaskedEdit28.Text
MsgBox "CP " + MaskedEdit8.Text
MaskedEdit10.Text = MaskedEdit35.Text
MsgBox "Telefono " + MaskedEdit10.Text
MaskedEdit13.Text = MaskedEdit30.Text
MsgBox "N. Cuenta " + MaskedEdit13.Text
MaskedEdit14.Text = MaskedEdit31.Text
MsgBox "Semestre A " + MaskedEdit14.Text
MaskedEdit16.Text = MaskedEdit32.Text
MsgBox "Promedio " + MaskedEdit16.Text
MaskedEdit17.Text = MaskedEdit33.Text
MsgBox "Numero Materias " + MaskedEdit17.Text
MaskedEdit18.Text = MaskedEdit34.Text
MsgBox "Año ingreso " + MaskedEdit18.Text
MaskedEdit12.Text = MaskedEdit36.Text
MsgBox "Fecha Nac " + MaskedEdit12.Text
MaskedEdit11.Text = UCase(MaskedEdit37.Text)
MsgBox "RFC " + MaskedEdit11.Text
MaskedEdit15.Text = UCase(MaskedEdit38.Text)
MsgBox "Ced PProf " + MaskedEdit15.Text
MaskedEdit9.Text = MaskedEdit39.Text
MaskedEdit19.Text = MaskedEdit40.Text
MaskedEdit20.Text = MaskedEdit41.Text
MaskedEdit21.Text = MaskedEdit42.Text
MaskedEdit22.Text = MaskedEdit43.Text
'Fin de la actualización de los campos
Data3.Recordset.Update
End If
respu = MsgBox("Desea realizar otro cambio?", 36, "Confirmación")
If respu = 7 Then
  Unload cambiar2
Else
  RESP_CONSUL = ""
  RESP_CONSUL = UCase(InputBox("INTRODUCE EL LOGIN..."))
  If RESP_CONSUL <> "" Then
    Data3.Refresh
    Data3.Recordset.FindFirst "[LOGIN] LIKE " & RESP_CONSUL & ""
    If Data3.Recordset.NoMatch Then
      msg = "No hay algún Usuario con Login" + Chr(13) + Chr(10) +
        UCase(RESP_CONSUL)
      MsgBox msg, 48, "Error"
    Else
      Data4.Refresh
      Data4.Recordset.FindFirst "[ID_DEL] like " & CStr(Data3.Recordset(9)) & ""

```



```

Combo1.Text = Data4.Recordset(1)
Data5.Refresh
Data5.Recordset.FindFirst "[ID_INST] like '" & CStr(Data3.Recordset(15)) & ""
Combo4.Text = Data5.Recordset(1)
Data1.Refresh
Data1.Recordset.FindFirst "[ID_LUGAR] like '" & CStr(Data3.Recordset(13)) & ""
Combo2.Text = Data1.Recordset(1)
Data2.Refresh
Data2.Recordset.FindFirst "[ID_CVE] like '" & CStr(Data3.Recordset(14)) & ""
Combo3.Text = Data2.Recordset(1) + " Horas"
Data6.Refresh
Data6.Recordset.FindFirst "[ID_TIPO] like '" & CStr(Data3.Recordset(22)) & ""
Combo5.Text = Data6.Recordset(1)
MaskedEdit23.Text = MaskedEdit2.Text 'Nombre
MaskedEdit24.Text = MaskedEdit3.Text 'Apellido Paterno
MaskedEdit25.Text = MaskedEdit4.Text 'Apellido Materno
MaskedEdit26.Text = MaskedEdit5.Text 'Calle
MaskedEdit27.Mask = ""
MaskedEdit27.Text = MaskedEdit6.Text 'Número del domicilio
MaskedEdit27.Mask = "#####"
MaskedEdit27.Text = MaskedEdit7.Text 'Colonia
MaskedEdit28.Mask = ""
MaskedEdit28.Text = MaskedEdit8.Text 'Codigo Postal
MaskedEdit28.Mask = "#####"
MaskedEdit35.Mask = ""
MaskedEdit35.Text = MaskedEdit10.Text 'Teléfono
MaskedEdit35.Mask = "(###) ###-####"
MaskedEdit37.Mask = ""
MaskedEdit37.Text = MaskedEdit11.Text 'RFC
MaskedEdit37.Mask = "????#####"
MaskedEdit36.Mask = ""
MaskedEdit36.Text = MaskedEdit12.Text 'Fecha de nacimiento
MaskedEdit36.Mask = "###/###/##"
MaskedEdit30.Mask = ""
MaskedEdit30.Text = MaskedEdit13.Text 'Número de Cuenta
MaskedEdit30.Mask = "#####-"
MaskedEdit31.Text = MaskedEdit14.Text
MaskedEdit38.Mask = ""
MaskedEdit38.Text = MaskedEdit15.Text 'Ced Profesional
MaskedEdit38.Mask = "#####"
MaskedEdit32.Text = MaskedEdit19.Text 'Promedio
MaskedEdit33.Mask = ""
MaskedEdit33.Text = MaskedEdit17.Text 'No. Materias
MaskedEdit33.Mask = "##"
MaskedEdit34.Mask = ""

```



```

MaskedEdit34.Text = MaskedEdit18.Text ' Año de Ingreso
MaskedEdit34.Mask = "19##"
MaskedEdit39.Text = MaskedEdit9.Text
MaskedEdit40.Text = MaskedEdit19.Text
MaskedEdit41.Text = MaskedEdit20.Text
MaskedEdit42.Text = MaskedEdit21.Text
MaskedEdit43.Text = MaskedEdit22.Text
End If
End If
End If
End Sub
Sub Form_Load ()
ban_lugarC = 1
ban_claveC = 1
ban_delegacionC = 1
ban_institucionC = 1
ban_tipoC = 1
If ban_lugarC = 1 Then
Combo2.Clear
Combo9.Clear
Data1.Refresh
Do While Not Data1.Recordset.EOF
If Not IsNull(Data1.Recordset(0)) Then
Combo2.AddItem Data1.Recordset(1) 'añadirlo a la lista
Combo9.AddItem Data1.Recordset(0)
End If
Data1.Recordset.MoveNext 'ir al siguiente registro
Loop
End If
' Combo para Tipos de clave
If ban_claveC = 1 Then
Combo3.Clear
Combo10.Clear
Data2.Refresh
Do While Not Data2.Recordset.EOF
If Not IsNull(Data2.Recordset(0)) Then
Combo3.AddItem Data2.Recordset(1) + " Horas" 'añadirlo a la lista
Combo10.AddItem Data2.Recordset(0)
End If
Data2.Recordset.MoveNext 'ir al siguiente registro
Loop
End If
' Combo para las DELEGACIONES
If ban_delegacionC = 1 Then
Combo1.Clear

```



```
Combo7.Clear
Data4.Refresh
Do While Not Data4.Recordset.EOF
  If Not IsNull(Data4.Recordset(0)) Then
    Combo1.AddItem Data4.Recordset(1) 'añadirlo a la lista
    Combo7.AddItem Data4.Recordset(0)
  End If
  Data4.Recordset.MoveNext 'ir al siguiente registro
Loop
End If
' Combo para INSTITUCIONES
If ban_institucionC = 1 Then
  Combo4.Clear
  Combo8.Clear
  Data5.Refresh
  Do While Not Data5.Recordset.EOF
    If Not IsNull(Data5.Recordset(0)) Then
      Combo4.AddItem Data5.Recordset(1) 'añadirlo a la lista
      Combo8.AddItem Data5.Recordset(0)
    End If
    Data5.Recordset.MoveNext 'ir al siguiente registro
  Loop
End If
' Combo para TIPO DE USUARIO
If ban_tipoC = 1 Then
  Combo5.Clear
  Combo6.Clear
  Data6.Refresh
  Do While Not Data6.Recordset.EOF
    If Not IsNull(Data6.Recordset(0)) Then
      Combo5.AddItem Data6.Recordset(1) 'añadirlo a la lista
      Combo6.AddItem Data6.Recordset(0)
    End If
    Data6.Recordset.MoveNext 'ir al siguiente registro
  Loop
End If
ban_lugarC = 0
ban_claveC = 0
ban_delegacionC = 0
ban_institucionC = 0
ban_tipoC = 0
'Form_Activate
Data3.Refresh
Data3.Recordset.FindFirst "[LOGIN] like " & CS$(RESP_CONSUL) & ""
Data4.Refresh
```



```
Data4.Recordset.FindFirst "[ID_DEL] like " & CStr(Data3.Recordset(9)) & ""
Combo1.Text = Data4.Recordset(1)
Data5.Refresh
Data5.Recordset.FindFirst "[ID_INST] like " & CStr(Data3.Recordset(15)) & ""
Combo4.Text = Data5.Recordset(1)
Data1.Refresh
Data1.Recordset.FindFirst "[ID_LUGAR] like " & CStr(Data3.Recordset(13)) & ""
Combo2.Text = Data1.Recordset(1)
Data2.Refresh
Data2.Recordset.FindFirst "[ID_CVE] like " & CStr(Data3.Recordset(14)) & ""
Combo3.Text = Data2.Recordset(1) + " Horas"
Data6.Refresh
Data6.Recordset.FindFirst "[ID_TIPO] like " & CStr(Data3.Recordset(22)) & ""
Combo5.Text = Data6.Recordset(1)
MaskedEdit23.Text = MaskedEdit2.Text 'Nombre
MaskedEdit24.Text = MaskedEdit3.Text 'Apellido Paterno
MaskedEdit25.Text = MaskedEdit4.Text 'Apellido Materno
MaskedEdit26.Text = MaskedEdit5.Text 'Calle
MaskedEdit27.Mask = ""
MaskedEdit27.Text = MaskedEdit6.Text 'Número del domicilio
MaskedEdit27.Mask = "#####"
MaskedEdit27.PromptInclude = False
MaskedEdit29.Text = MaskedEdit7.Text 'Colonia
MaskedEdit28.Mask = ""
MaskedEdit28.Text = MaskedEdit8.Text 'Codigo Postal
MaskedEdit28.Mask = "#####"
MaskedEdit28.PromptInclude = False
MaskedEdit39.Text = MaskedEdit9.Text
MaskedEdit35.Mask = ""
MaskedEdit35.Text = MaskedEdit10.Text ' Teléfono
MaskedEdit35.Mask = "(###) ###-####"
MaskedEdit35.PromptInclude = False
MaskedEdit37.Mask = ""
MaskedEdit37.Text = MaskedEdit11.Text ' RFC
MaskedEdit37.Mask = "???"#####"
MaskedEdit37.PromptInclude = False
MaskedEdit36.Mask = ""
MaskedEdit36.Text = MaskedEdit12.Text ' Fecha de nacimiento
MaskedEdit36.Mask = "###/###/###"
MaskedEdit36.PromptInclude = False
MaskedEdit30.Mask = ""
MaskedEdit30.Text = MaskedEdit13.Text ' Número de Cuenta
MaskedEdit30.Mask = "#####-#"
MaskedEdit30.PromptInclude = False
MaskedEdit31.Text = MaskedEdit14.Text
```



```

MaskedEdit38.Mask = ""
MaskedEdit38.Text = MaskedEdit15.Text ' Ced Profesional
MaskedEdit38.Mask = "#####"
MaskedEdit38.PromptInclude = False
MaskedEdit32.Text = MaskedEdit19.Text ' Promedio
MaskedEdit33.Mask = ""
MaskedEdit33.Text = MaskedEdit17.Text ' No. Materias
MaskedEdit33.Mask = "##"
MaskedEdit33.PromptInclude = False
MaskedEdit34.Mask = ""
MaskedEdit34.Text = MaskedEdit18.Text ' Año de Ingreso
MaskedEdit34.Mask = "19##"
MaskedEdit34.PromptInclude = False
MaskedEdit40.Text = MaskedEdit19.Text
MaskedEdit41.Text = MaskedEdit20.Text
MaskedEdit42.Text = MaskedEdit21.Text
MaskedEdit43.Text = MaskedEdit22.Text
'MaskedEdit23.SetFocus
End Sub
Sub Frame3D1_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    label23.Caption = ""
End Sub
Sub Frame3D2_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    label23.Caption = ""
End Sub
Sub Frame3D3_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    label23.Caption = ""
End Sub
Sub Frame3D4_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    label23.Caption = ""
End Sub
Sub Frame3D5_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    label23.Caption = ""
End Sub
Sub MaskedEdit23_KeyPress (KeyAscii As Integer)
    If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit23.Text = "") Then
        MsgBox "Se requiere especificar el Nombre del Usuario", 48, "Error"
        MaskedEdit23.SetFocus
    Else
        If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit23.Text <> "") Then
            MaskedEdit24.SetFocus
        End If
    End If
End Sub
Sub MaskedEdit24_KeyPress (KeyAscii As Integer)

```



```
If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit24.Text = "") Then
    MsgBox "Se requiere especificar el Apellido Paterno del Usuario", 48, "Error"
    MaskedEdit24.SetFocus
Else
    If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit23.Text <> "") Then
        MaskedEdit25.SetFocus
    End If
End If
End Sub
Sub MaskedEdit25_KeyPress (KeyAscii As Integer)
    If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit25.Text = "") Then
        MsgBox "Se requiere especificar el Apellido Materno del Usuario", 48, "Error"
        MaskedEdit25.SetFocus
    Else
        If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit23.Text <> "") Then
            MaskedEdit26.SetFocus
        End If
    End If
End Sub
Sub MaskedEdit26_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit27.SetFocus
    End If
End Sub
Sub MaskedEdit27_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit29.SetFocus
    End If
End Sub
Sub MaskedEdit28_KeyPress (KeyAscii As Integer)
    If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit28.Text = "") Then
        Combo1.SetFocus
    ElseIf ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit28.Text) = 5) Then
        Combo1.SetFocus
    ElseIf ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit28.Text) > 0) Then
        MsgBox "El Código Postal que especifico, es incorrecto...", 48, "Error"
    End If
End Sub
Sub MaskedEdit29_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit28.SetFocus
    End If
End Sub
Sub MaskedEdit30_KeyPress (KeyAscii As Integer)
    MaskedEdit30.PromptInclude = False
```



```

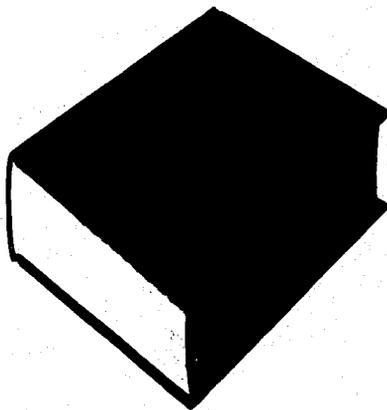
If ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit30.Text) = 0) Then
    MaskedEdit31.PromptInclude = False
    MaskedEdit31.SetFocus
    'MsgBox CStr(Len(MaskedEdit30.Text))
Elseif ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit30.Text) = 8) Then
    MaskedEdit30.PromptInclude = True
    MaskedEdit31.SetFocus
    'MsgBox CStr(Len(MaskedEdit30.Text))
Elseif ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit30.Text) <> 8) Then
    MsgBox "El Número de cuenta que Usted especificó es incorrecto", 48, "Error"
    MaskedEdit30.SetFocus
    'MsgBox CStr(Len(MaskedEdit30.Text))
End If
End Sub
Sub MaskedEdit31_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit32.SetFocus
    End If
End Sub
Sub MaskedEdit32_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit33.SetFocus
    End If
End Sub
Sub MaskedEdit33_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        MaskedEdit34.SetFocus
    End If
End Sub
Sub MaskedEdit34_KeyPress (KeyAscii As Integer)
    If ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit34.Text) = 2) Then
        PromptInclude = True
        Combo2.SetFocus
    Elseif ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit34.Text) = 0) Then
        PromptInclude = False
        Combo2.SetFocus
    Elseif ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit34.Text) < 2) Then
        MsgBox "Usted especificó un año invalido", 48, "Error"
        MaskedEdit34.SetFocus
    End If
End Sub
Sub MaskedEdit35_KeyPress (KeyAscii As Integer)
    If (KeyAscii = 13 Or KeyAscii = 9) Then
        Combo4.SetFocus
    End If

```



```
End Sub
Sub MaskedEdit36_KeyPress (KeyAscii As Integer)
  MsgBox CStr(Len(MaskedEdit36.Text))
  MaskedEdit36.PromptInclude = False
  If ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit36.Text) = 0) Then
    MaskedEdit36.PromptInclude = False
    MaskedEdit37.SetFocus
  Else
    If ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit36.Text) <> 0) Then
      If (Len(MaskedEdit36.Text) <> 6) Then
        MsgBox "La Fecha que especifico, es invalida...", 48, "Error"
        MaskedEdit36.SetFocus
      Else
        MaskedEdit36.PromptInclude = True
        MaskedEdit37.SetFocus
      End If
    End If
  End If
End Sub
Sub MaskedEdit37_KeyPress (KeyAscii As Integer)
  If ((KeyAscii = 13 Or KeyAscii = 9) And MaskedEdit37.Text = "") Then
    MaskedEdit38.SetFocus
  ElseIf ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit37.Text) = 10) Then
    MaskedEdit38.SetFocus
  ElseIf ((KeyAscii = 13 Or KeyAscii = 9) And Len(MaskedEdit37.Text) > 0) Then
    MsgBox "El RFC que Usted especificó es incorrecto", 48, "Error"
    MaskedEdit37.SetFocus
  End If
End Sub
Sub MaskedEdit38_KeyPress (KeyAscii As Integer)
  If (KeyAscii = 13 Or KeyAscii = 9) Then
    Combo3.SetFocus
  End If
End Sub
```

GLOSARIO





GLOSARIO

ANSI

American National Standards Institute. Grupo que define los estándares de EU. para la industria del procesamiento de la información. ANSI participa en la definición de estándares para protocolos de redes.

APLICACIÓN

Programa que es creado para ayudar al usuario en una tarea específica.

BASE DE DATOS

Conjunto de información no redundante (conjunto de datos) relacionada que se encuentra agrupada o estructurada y que es compartida por varios usuarios y/o aplicaciones.

BBS

Bulletin Board Service ó Boletín electrónico, es una computadora preparada para recibir llamadas y actuar como un host. Los BBS permiten comunicarse entre sí a los usuarios mediante mensajes, así como intercambiar archivos.

BUFFER

Área reservada de memoria que en conjunto se comporta como una sola unidad.

CASE (*Computer Aided Software Engineering*)

Tecnología de ayuda en la realización de análisis sistemáticos, diseño, implementación y mantenimiento del software.



CLIENTE

Programa de computadora que utiliza los recursos dados por otra máquina o programa a través de la red.

CUI

Character User Interface. Interface de presentación de datos orientada al despliegue únicamente de caracteres.

DDE (*Dynamic Data Exchange*)

Protocolo de Liga de Microsoft Windows 3.0 que permite la comunicación entre programas de aplicación usando un modelo Cliente-Servidor.

DLL (*Dynamically Linked Library*)

Librerías que permiten ligar aplicaciones cuando son cargadas o corren en vez de la final de compilación. Esto significa que el mismo código puede ser compartido entre varias tareas en vez de que cada tarea contenga copias de las rutinas en uso.

DOMINIO

Parte de la jerarquía de nombres de sistemas. Sintácticamente un nombre de un dominio consiste en una sola secuencia de nombres u otras palabras separadas por puntos.

EBCDIC

Código expandido de intercambio de información codificado en binario. Es de 8 bits por carácter y se utiliza a menudo en mainframes.



FDDI

Fiber Distribution Data Interface. Estándar reciente de comunicaciones basado en fibra óptica que ha sido establecido por ANSI. FDDI especifica una velocidad de señalización de 100 Mbps y una longitud máxima de 200Km.

FIRMWARE.

Hardware programado permanentemente.

FQDN

Full Qualified Domain Name. EL FQDN es el nombre completo de un sistema, no solamente el nombre del host en sí. Por ejemplo, el sistema tzetzal que se encuentra en la DCAA tiene un FQDN de tzetzal.dcaa.unam.mx

FRAMES

Es un paquete transmitido por una línea serial . El término deriva de los protocolos orientados a carácter que añaden caracteres especiales que indican el principio y fin del frame cuando transmite paquetes de información.

FRONT-END

Término que se refiere a la parte de una aplicación que interactúa directamente con el usuario.

FTP

File Transport Protocol. Este es el protocolo de alto nivel estándar de la Internet para la transferencia de archivos de una computadora a otra.



FYI

For You Information. Término dado a los documentos almacenados en varios host de la red Internet con un lenguaje no técnico que describen tópicos relacionados con la misma red.

GATEWAY

Una computadora dedicada a unir dos redes y rutear paquetes de una a otra. Los gateways rutean paquetes a otro gateway hasta que sean entregados a su destino final a través de la red física.

GUI

Graphical user interfaz. Interface de salida que permite la visualización de imágenes, gráficas, etc.

HARDWARE

Elementos físicos que componen a un sistema electrónico.

HIPERLIGA

Medio a través del cual se puede apuntar de forma única a un documento, imagen, video, sonido a través de la red.

HIPERMEDIOS

Conjunción de datos de diferentes formatos, como pueden ser textos, imágenes, audio, video, etc., los cuales pueden ser accesados por una aplicación.



HOST

Cualquier computadora capaz de ejecutar programas de aplicación que se encuentran conectadas a una red.

HTML

HyperText Markup Language Este es un lenguaje de descripción muy simple, pero lo suficientemente complejo para permitir formateo y presentación de documentos de hipertexto.

HTTP

HyperText Transfer Protocol. Protocolo estándar para la transferencia de documentos de hipertexto.

LAN

Local Area Network. cualquier tecnología física de redes que opera a altas velocidades sobre distancias cortas (por ejemplo a pocos cientos de metros).

LISTAS DE CORREO (Mailing List).

Una lista de correo es un grupo de discusión, distribuido por correo electrónico desde una computadora central que mantiene una lista de gente relacionada con algún tema de discusión.

MAINFRAME

Término aplicado a las computadoras de alto rendimiento con características de un alto nivel transaccional, gran capacidad para atender gran cantidad de usuarios y con requerimientos especiales en cuanto a condiciones ambientales para su funcionamiento.



MIPS

Millones de Instrucciones Por Segundo. Término utilizado para medir la capacidad de procesamiento de un equipo de cómputo.

MODEM

MODulador/DEModulador. Dispositivo electrónico que convierte datos seriales a señales de computadora o audio a través de la línea telefónica.

MULTIMEDIA

Conjunción de video, imágenes, sonidos, texto, gráficas, etc. en un programa que permite la mas fácil interacción entre el programa y los usuarios.

ODBC

Open Database Connectivity. Conexión de Bases de Datos abierta, es una interfaz de conexión a Bases de Datos tal como bases de datos SQL Server u Oracle Server, que viene con el programa Access y contiene los controladores apropiados para acceder a datos de Bases de Datos SQL e importar, exportar o adjuntar los datos almacenados en dichas Bases de Datos.

OLE (*Object Linking and Embedding*)

Es un sistema de objetos distribuidos y protocolo de Microsoft. OLE permite a un editor insertar parte de un documento hacia otro editor e importarlos nuevamente.

OSI

Open System Interconnection. Estándar de arquitectura de redes propuesto por ISO.



PÁGINA

Subdivisión natural de la memoria. El número de página se identifica por los bits más significativos de la dirección. Las posiciones dentro de la página las determinan los bits menos significativos.

PROTOCOLO

Conjunto de convenciones y normas para establecer un diálogo a distintos niveles de implementación de una arquitectura de redes.

QBE

Query By Example. Consulta según ejemplo, el uso de QBE permite extraer la información que necesita sin programar. Access convierte el diseño de consulta de la cuadrícula QBE en un orden SQL.

QUERY

Programa o consulta que permite al usuario tener acceso a información contenida en algún repositorio (base de datos o archivos).

RedUNAM

Proyecto de interconexión de cómputo en la comunidad universitaria.

RFC

Request For Comments. Documentos técnicos que se encuentran distribuidos en la red Internet que definen estándares y políticas de ésta.



SCRIPT

Conjunto de instrucciones y/o comandos (generalmente de un sistema operativo como UNIX) cuyo fin es el de automatizar o programar alguna acción.

SCSI

Small Computer System Interface. Definición lógica y física para la interconexión de hosts, unidades de disco, cinta, y otros periféricos.

SEARCH ENGINE

Parte de un programa o programa encargado de realizar búsquedas de información en archivos planos o bases de datos.

SERVIDOR

Programa de computadora o máquina que da recursos a otras computadoras o programas.

SISTEMA OPERATIVO

Software del sistema que controla el funcionamiento de la computadora incluyendo varias tareas, como asignación de posiciones en memoria, procesado de interrupciones y manejo de tareas.

SITE

Centro de cómputo.



SOFTWARE

Conjunto de programas que hacen uso a su vez de los respectivos conjuntos de datos para su funcionamiento.

SQL

Structure Query Language. Lenguaje de consulta estructurado que se utiliza para consultar, actualizar y gestionar bases de datos relacionales.

STACK

Grupo de registros o posiciones de memoria, usados para guardar las direcciones de retorno durante la ejecución de subrutinas.

TCP/IP

(Transmission Control Protocol/Internet Protocol). Este es un conjunto (familia) de protocolos, resultado de los esfuerzos de ARPA, usado por la Internet para mantener servicios como login's remotos (telnet), transferencias de archivos (FTP) y correo electrónico (SMTP).

TELEPROCESO

Forma de comunicación entre terminales y el mainframe, en donde las terminales solo realizan la función de presentación y captura de datos de parte del usuario mientras que el mainframe realiza el procesamiento en sí.



TELNET

Es el protocolo estándar de la Internet del servicio de conexión remota de terminales. Telnet permite a un usuario en una máquina interactuar con un sistema remoto de tiempo compartido en algún otro lugar como si la terminal del usuario estuviera conectada directamente al sistema remoto.

TPS

Tcks Per Second. Término utilizado para medir la rapidez del reloj de una computadora.

UNIX

Sistema Operativo interactivo de tiempo compartido desarrollado en 1969 por Ken Thompson después que los laboratorios Bell abandonaran el proyecto Multics. UNIX existe en varias formas e implementaciones, siendo las más comunes System V y BSD.

URL

Uniform Resource Locator. Se puede pensar el URL como una extensión de los nombres de los archivos, ya que no solamente podemos apuntar al archivo en sí, sino que también en que máquina se encuentra y puede ser accesada por muchos métodos.

VISUAL BASIC

Sistema de Programación para Microsoft Windows en donde fragmentos de código de Basic son incluidos cuando el usuario desarrolla ciertas operaciones con objetos gráficos.

**VPE**

Visual Programming Environments. Son ambientes que proveen elementos o iconos gráficos los cuales pueden ser manipulados por el usuario en forma interactiva de acuerdo con una gramática especial para la construcción de programas.

VPL

Visual Programming Language. Son los lenguajes de programación que permiten al usuario realizar un programa por uno o más caminos. Un VPL permite programar con expresiones visuales como símbolos gráficos y texto.

WORKSTATION

Estación de trabajo. Cualquier combinación de dispositivos de cómputo que provee al usuario de un gran poder de procesamiento y velocidad, gráficas sofisticadas y un medio de almacenamiento permanente y de rápido acceso. Generalmente conectado a red y con un sistema operativo poderoso.

WWW

World Wide Web (W³). Proyecto realizado por el CERN (*Europea Laboratory for Particle Physics, ubicado en Ginebra Suiza*) el cual permite acceder información a través de la red sobre cualquier tópico utilizando para ello hipertexto y técnicas de multimedia.

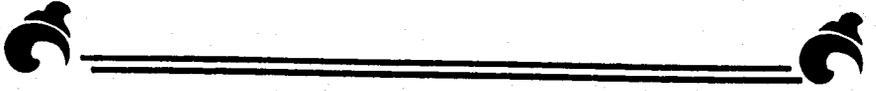
WYSIWYG

What You See Is What You Get. Acrónimo de la frase "lo que ves es lo que obtienes". Esta es una característica de programas de aplicación en el que la presentación en la pantalla será igual a la obtenida por una impresión en papel.

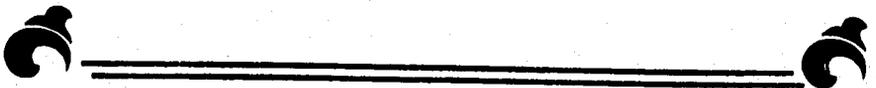
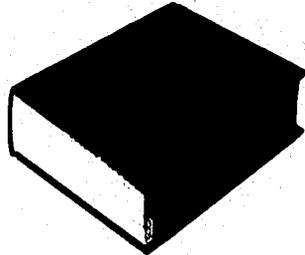
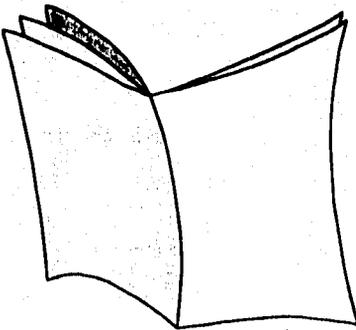


XII

Conjunto estandarizado de rutinas para el manejo gráfico de pantallas, aplicaciones, etc. que son totalmente independientes del hardware utilizado para dicho propósito. Este fue desarrollado por el MIT y es de dominio público.



BIBLIOGRAFIA





BIBLIOGRAFÍA**LIBROS**

Burgad, Michael J. ed alt.
Dos ↔ Unix Networking and Internetworking
Wiley, 1994.

Ceballos, Francisco Javier
Enciclopedia de Visual Basic
Cd, Editorial, Año

Ceballos, Francisco Javier
Manual para Quick Base 4.5 Guía del Programador
Macrobit RA-MA

Date, Chris J.
Introducción a los Sistemas de Bases de Datos Vol.1
Addison-Wesley Iberoamerican, Quinta Edición

Diccionario de Informática
Ediciones Díaz de Santos, S.A. 1990

Gookin, Dan ed alt.
Diccionario Ilustrado de Computación para Inexpertos
Megabyte, Noriega Editores

Joyanes, Aguilar Luis
Turbo Pascaln 6.0 a su alcance
Mc. Graw Hill

Korjil, Henry F. ed alt.
Fundamentos de Bases de Datos
Mc Graw Hill.

Marchuk, Michael
Building Internet Applications with Visual Basic
USA, De. QUE Corporation, 1995.

Martin, James
Organizaciones de Bases de Datos
Prentice Hall



McGee, Margaret D.
Microsoft Access for Windows Step by Step
USA, Microsoft Press, 1993.

Nelson, Ross
Guía Completa de Visual Basic
España, Mc Graw Hill, 1994.

Pe NFS Serial Communications Guide
USA, Sunselect Microsystems, 1993.

Presman, S. Roger
Ingeniería de Software "Un Enfoque Práctico "
México, Mc Graw Hill, 1993

TESIS

- **Diseño e Implementación de Múltiples Servicios Computacionales bajo Ambiente Gráfico en la red de Comisión Nacional de Derechos Humanos.**
- **Diseño e Implementación de una Guía Gráfica de Ciudad Universitaria** Colocación: 32 95 108
- **Implementación de un Sistema de prestadores de Bienes y Servicios en Ixtapa Zihuatanejo**
- **Kioskos de Información** Colocación: 32 95 071
- **Sistemas de Comercialización de Productos automotrices en base a la metodología Case de Oracle**

DIRECCIONES ELECTRÓNICAS

http://noc.noc.unam.mx>70/11/redunam/servi_unam
<http://www.noc.unam.mx/>
<http://noc.noc.unam.mx/dtd/conect.html>
<http://www.microsoft.com/>
<http://www.cisco.com/>
<http://www.yahoo.com/> índice de búsqueda



REVISTAS

- PC Magazine en Español D.F, Vol 5 Num 10
Pag-51 Equilibrio de fuerza
- PC Magazine en Español D.F, Vol 6 Num 6
Pag-8 World Wide Web
- PC Magazine en Español D.F, Vol 4 Num 4
Pag-7 Microsoft Access
- PC Magazine en Español D.F, Vol 6 Num 7
Pag-56 Los modems V.3.4
- PC Magazine en Español D.F, Vol 4 Num 8
Pag-12 Visual C++